



Welcome to [E-XFL.COM](http://E-XFL.COM)

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

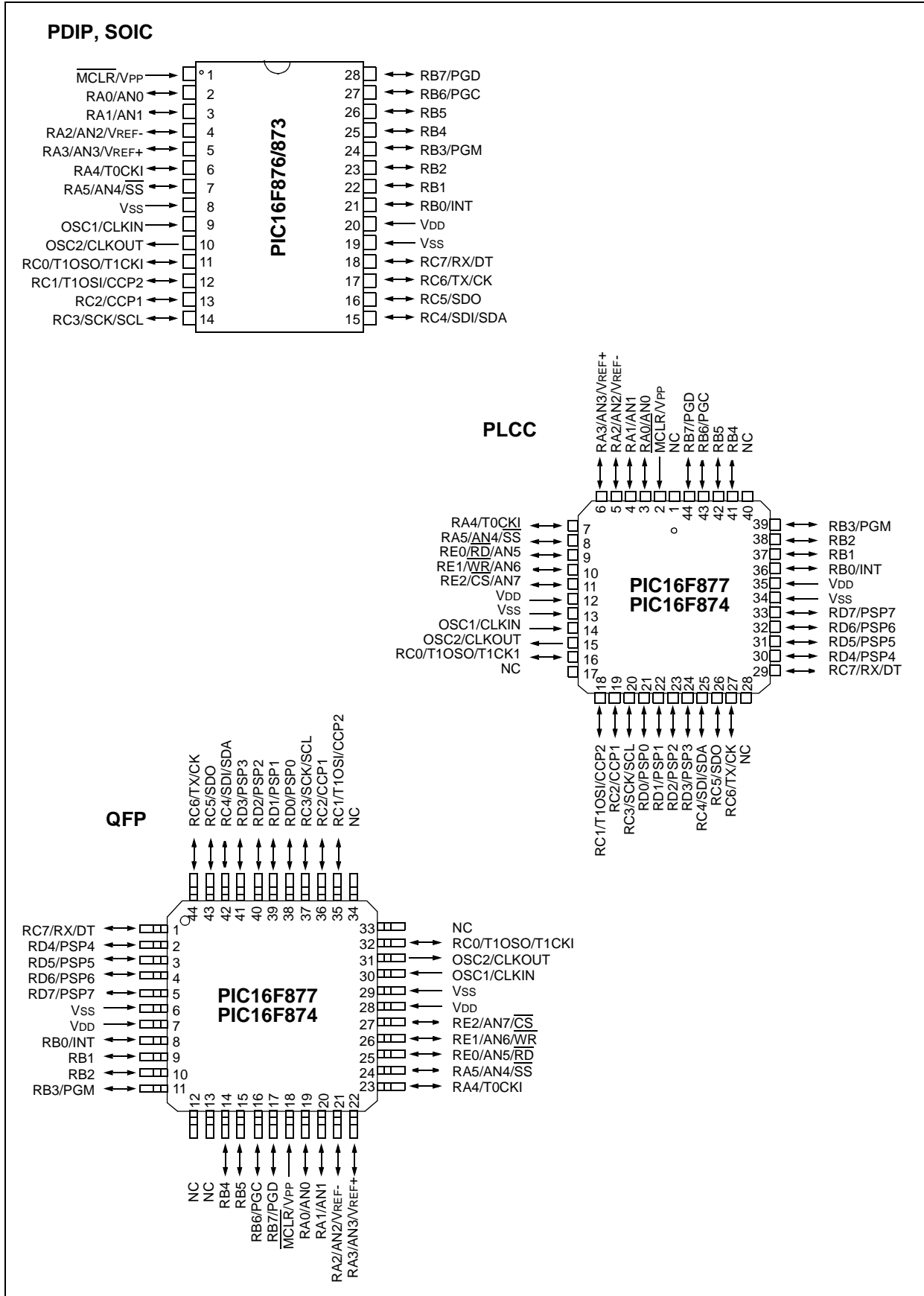
"[Embedded - Microcontrollers](#)" refer to small, integrated circuits designed to perform specific tasks within larger systems. These microcontrollers are essentially compact computers on a single chip, containing a processor core, memory, and programmable input/output peripherals. They are called "embedded" 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	4MHz
Connectivity	I <sup>2</sup> C, SPI, UART/USART
Peripherals	Brown-out Detect/Reset, POR, PWM, WDT
Number of I/O	22
Program Memory Size	7KB (4K x 14)
Program Memory Type	FLASH
EEPROM Size	128 x 8
RAM Size	192 x 8
Voltage - Supply (Vcc/Vdd)	2V ~ 5.5V
Data Converters	A/D 5x10b
Oscillator Type	External
Operating Temperature	0°C ~ 70°C (TA)
Mounting Type	Through Hole
Package / Case	28-DIP (0.300", 7.62mm)
Supplier Device Package	28-SPDIP
Purchase URL	<a href="https://www.e-xfl.com/product-detail/microchip-technology/pic16lf873-04-sp">https://www.e-xfl.com/product-detail/microchip-technology/pic16lf873-04-sp</a>

# PIC16F87X

## Pin Diagrams



# PIC16F87X

---

NOTES:

# PIC16F87X

## 2.2.2.5 PIR1 Register

The PIR1 register contains the individual flag bits for the peripheral interrupts.

**Note:** Interrupt flag bits are set when an interrupt condition occurs, regardless of the state of its corresponding enable bit or the global enable bit, GIE (INTCON<7>). User software should ensure the appropriate interrupt bits are clear prior to enabling an interrupt.

### REGISTER 2-5: PIR1 REGISTER (ADDRESS 0Ch)

R/W-0	R/W-0	R-0	R-0	R/W-0	R/W-0	R/W-0	R/W-0
PSPIF <sup>(1)</sup>	ADIF	RCIF	TXIF	SSPIF	CCP1IF	TMR2IF	TMR1IF
bit 7							bit 0

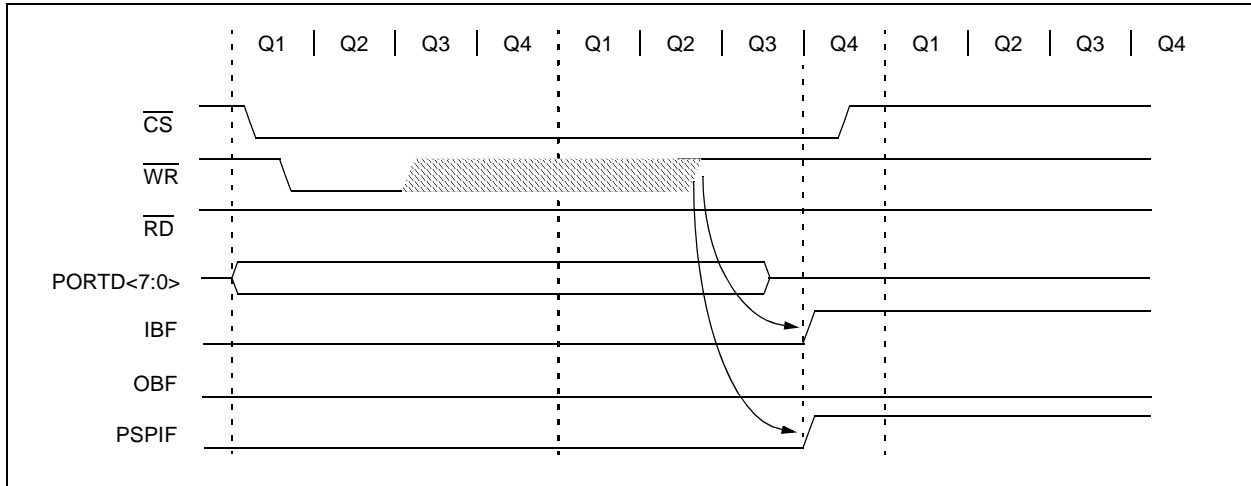
- bit 7 **PSPIF<sup>(1)</sup>**: Parallel Slave Port Read/Write Interrupt Flag bit  
 1 = A read or a write operation has taken place (must be cleared in software)  
 0 = No read or write has occurred
- bit 6 **ADIF**: A/D Converter Interrupt Flag bit  
 1 = An A/D conversion completed  
 0 = The A/D conversion is not complete
- bit 5 **RCIF**: USART Receive Interrupt Flag bit  
 1 = The USART receive buffer is full  
 0 = The USART receive buffer is empty
- bit 4 **TXIF**: USART Transmit Interrupt Flag bit  
 1 = The USART transmit buffer is empty  
 0 = The USART transmit buffer is full
- bit 3 **SSPIF**: Synchronous Serial Port (SSP) Interrupt Flag  
 1 = The SSP interrupt condition has occurred, and must be cleared in software before returning from the Interrupt Service Routine. The conditions that will set this bit are:
- SPI
    - A transmission/reception has taken place.
  - I<sup>2</sup>C Slave
    - A transmission/reception has taken place.
  - I<sup>2</sup>C Master
    - A transmission/reception has taken place.
    - The initiated START condition was completed by the SSP module.
    - The initiated STOP condition was completed by the SSP module.
    - The initiated Restart condition was completed by the SSP module.
    - The initiated Acknowledge condition was completed by the SSP module.
    - A START condition occurred while the SSP module was idle (Multi-Master system).
    - A STOP condition occurred while the SSP module was idle (Multi-Master system).
- 0 = No SSP interrupt condition has occurred.
- bit 2 **CCP1IF**: CCP1 Interrupt Flag bit  
Capture mode:  
 1 = A TMR1 register capture occurred (must be cleared in software)  
 0 = No TMR1 register capture occurred  
Compare mode:  
 1 = A TMR1 register compare match occurred (must be cleared in software)  
 0 = No TMR1 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 in software)  
 0 = No TMR2 to PR2 match occurred
- bit 0 **TMR1IF**: TMR1 Overflow Interrupt Flag bit  
 1 = TMR1 register overflowed (must be cleared in software)  
 0 = TMR1 register did not overflow

**Note 1:** PSPIF is reserved on PIC16F873/876 devices; always maintain this bit clear.

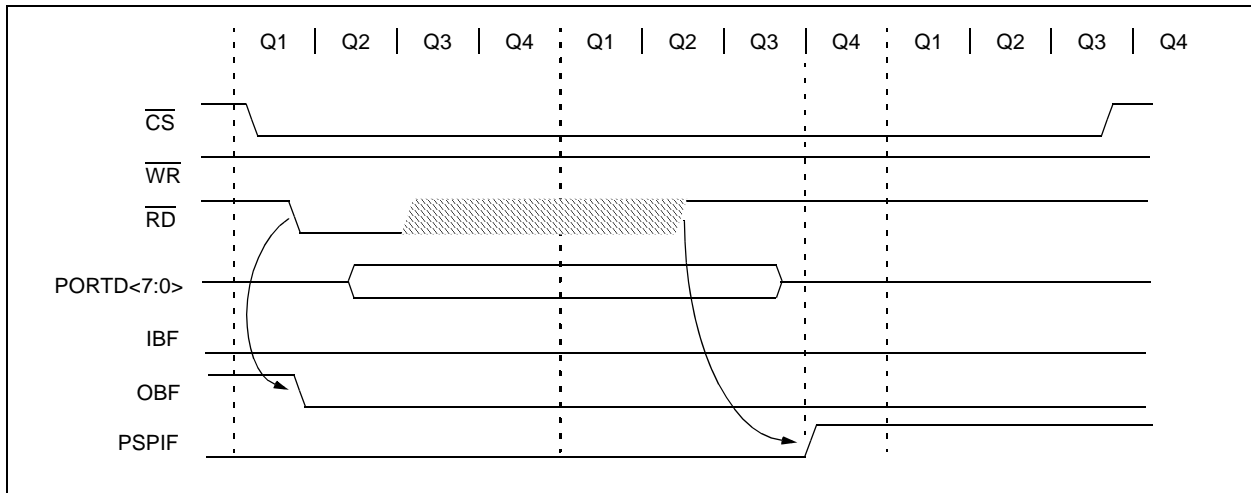
Legend:

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

**FIGURE 3-10: PARALLEL SLAVE PORT WRITE WAVEFORMS**



**FIGURE 3-11: PARALLEL SLAVE PORT READ WAVEFORMS**



**TABLE 3-11: REGISTERS ASSOCIATED WITH PARALLEL SLAVE PORT**

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Value on: POR, BOR	Value on all other RESETS
08h	PORTD	Port Data Latch when written: Port pins when read								xxxx xxxx	uuuu uuuu
09h	PORTE	—	—	—	—	—	RE2	RE1	RE0	---- -xxx	---- -uuu
89h	TRISE	IBF	OBF	IBOV	PSPMODE	—	PORTE Data Direction Bits			0000 -111	0000 -111
0Ch	PIR1	PSPIF <sup>(1)</sup>	ADIF	RCIF	TXIF	SSPIF	CCP1IF	TMR2IF	TMR1IF	0000 0000	0000 0000
8Ch	PIE1	PSPIE <sup>(1)</sup>	ADIE	RCIE	TXIE	SSPIE	CCP1IE	TMR2IE	TMR1IE	0000 0000	0000 0000
9Fh	ADCON1	ADFM	—	—	—	PCFG3	PCFG2	PCFG1	PCFG0	--0- 0000	--0- 0000

Legend: x = unknown, u = unchanged, - = unimplemented, read as '0'. Shaded cells are not used by the Parallel Slave Port.

**Note 1:** Bits PSPIE and PSPIF are reserved on the PIC16F873/876; always maintain these bits clear.

# PIC16F87X

## 4.4 Reading the FLASH Program Memory

Reading FLASH program memory is much like that of EEPROM data memory, only two `NOP` instructions must be inserted after the `RD` bit is set. These two instruction cycles that the `NOP` instructions execute, will be used by the microcontroller to read the data out of program memory and insert the value into the `EEDATH:EEDATA` registers. Data will be available following the second `NOP` instruction. `EEDATH` and `EEDATA` will hold their value until another read operation is initiated, or until they are written by firmware.

The steps to reading the FLASH program memory are:

1. Write the address to `EEADRH:EEADR`. Make sure that the address is not larger than the memory size of the PIC16F87X device.
2. Set the `EEPGD` bit to point to FLASH program memory.
3. Set the `RD` bit to start the read operation.
4. Execute two `NOP` instructions to allow the microcontroller to read out of program memory.
5. Read the data from the `EEDATH:EEDATA` registers.

### EXAMPLE 4-3: FLASH PROGRAM READ

```
BSF    STATUS, RP1    ;
BCF    STATUS, RP0    ;Bank 2
MOVF   ADDR, W        ;Write the
MOVWF  EEADR          ;address bytes
MOVF   ADDRH, W       ;for the desired
MOVWF  EEADRH        ;address to read
BSF    STATUS, RP0    ;Bank 3
BSF    EECON1, EEPGD ;Point to Program memory
BSF    EECON1, RD     ;Start read operation
NOP    ;Required two NOPs
NOP    ;
BCF    STATUS, RP0    ;Bank 2
MOVF   EEDATA, W      ;DATAH = EEDATA
MOVWF  DATAH        ;
MOVF   EEDATH, W      ;DATAH = EEDATH
MOVWF  DATAH        ;
```

## 4.5 Writing to the FLASH Program Memory

Writing to FLASH program memory is unique, in that the microcontroller does not execute instructions while programming is taking place. The oscillator continues to run and all peripherals continue to operate and queue interrupts, if enabled. Once the write operation completes (specification D133), the processor begins executing code from where it left off. The other important difference when writing to FLASH program memory, is that the `WRT` configuration bit, when clear, prevents any writes to program memory (see Table 4-1).

Just like EEPROM data memory, there are many steps in writing to the FLASH program memory. Both address and data values must be written to the SFRs. The `EEPGD` bit must be set, and the `WREN` bit must be set to enable writes. The `WREN` bit should be kept clear at all times, except when writing to the FLASH Program memory. The `WR` bit can only be set if the `WREN` bit was set in a previous operation, i.e., they both cannot be set in the same operation. The `WREN` bit should then be cleared by firmware after the write. Clearing the `WREN` bit before the write actually completes will not terminate the write in progress.

Writes to program memory must also be prefaced with a special sequence of instructions that prevent inadvertent write operations. This is a sequence of five instructions that must be executed without interruption for each byte written. These instructions must then be followed by two `NOP` instructions to allow the microcontroller to setup for the write operation. Once the write is complete, the execution of instructions starts with the instruction after the second `NOP`.

The steps to write to program memory are:

1. Write the address to `EEADRH:EEADR`. Make sure that the address is not larger than the memory size of the PIC16F87X device.
2. Write the 14-bit data value to be programmed in the `EEDATH:EEDATA` registers.
3. Set the `EEPGD` bit to point to FLASH program memory.
4. Set the `WREN` bit to enable program operations.
5. Disable interrupts (if enabled).
6. Execute the special five instruction sequence:
  - Write `55h` to `EECON2` in two steps (first to `W`, then to `EECON2`)
  - Write `AAh` to `EECON2` in two steps (first to `W`, then to `EECON2`)
  - Set the `WR` bit
7. Execute two `NOP` instructions to allow the microcontroller to setup for write operation.
8. Enable interrupts (if using interrupts).
9. Clear the `WREN` bit to disable program operations.

# PIC16F87X

## REGISTER 8-1: CCP1CON REGISTER/CCP2CON REGISTER (ADDRESS: 17h/1Dh)

U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	
—	—	CCPxX	CCPxY	CCPxM3	CCPxM2	CCPxM1	CCPxM0	
bit 7								bit 0

bit 7-6 **Unimplemented:** Read as '0'

bit 5-4 **CCPxX:CCPxY:** PWM Least Significant bits

Capture mode:

Unused

Compare mode:

Unused

PWM mode:

These bits are the two LSbs of the PWM duty cycle. The eight MSbs are found in CCPxL.

bit 3-0 **CCPxM3:CCPxM0:** CCPx Mode Select bits

0000 = Capture/Compare/PWM disabled (resets CCPx module)

0100 = Capture mode, every falling edge

0101 = Capture mode, every rising edge

0110 = Capture mode, every 4th rising edge

0111 = Capture mode, every 16th rising edge

1000 = Compare mode, set output on match (CCPxIF bit is set)

1001 = Compare mode, clear output on match (CCPxIF bit is set)

1010 = Compare mode, generate software interrupt on match (CCPxIF bit is set, CCPx pin is unaffected)

1011 = Compare mode, trigger special event (CCPxIF bit is set, CCPx pin is unaffected); CCP1 resets TMR1; CCP2 resets TMR1 and starts an A/D conversion (if A/D module is enabled)

11xx = PWM mode

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

# PIC16F87X

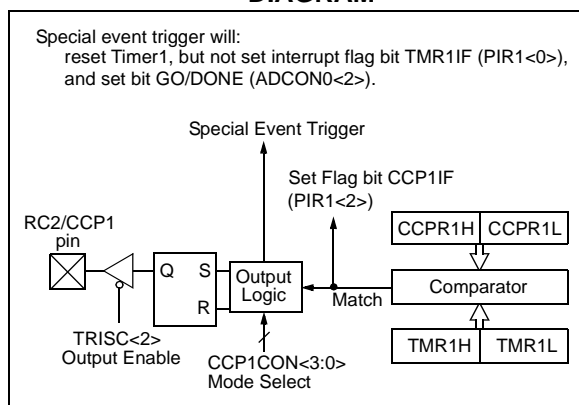
## 8.2 Compare Mode

In Compare mode, the 16-bit CCPR1 register value is constantly compared against the TMR1 register pair value. When a match occurs, the RC2/CCP1 pin is:

- Driven high
- Driven low
- Remains unchanged

The action on the pin is based on the value of control bits CCP1M3:CCP1M0 (CCP1CON<3:0>). At the same time, interrupt flag bit CCP1IF is set.

**FIGURE 8-2: COMPARE MODE OPERATION BLOCK DIAGRAM**



### 8.2.1 CCP PIN CONFIGURATION

The user must configure the RC2/CCP1 pin as an output by clearing the TRISC<2> bit.

**Note:** Clearing the CCP1CON register will force the RC2/CCP1 compare output latch to the default low level. This is not the PORTC I/O data latch.

### 8.2.2 TIMER1 MODE SELECTION

Timer1 must be running in Timer mode, or Synchronized Counter mode, if the CCP module is using the compare feature. In Asynchronous Counter mode, the compare operation may not work.

### 8.2.3 SOFTWARE INTERRUPT MODE

When Generate Software Interrupt mode is chosen, the CCP1 pin is not affected. The CCPIF bit is set, causing a CCP interrupt (if enabled).

### 8.2.4 SPECIAL EVENT TRIGGER

In this mode, an internal hardware trigger is generated, which may be used to initiate an action.

The special event trigger output of CCP1 resets the TMR1 register pair. This allows the CCPR1 register to effectively be a 16-bit programmable period register for Timer1.

The special event trigger output of CCP2 resets the TMR1 register pair and starts an A/D conversion (if the A/D module is enabled).

**Note:** The special event trigger from the CCP1 and CCP2 modules will not set interrupt flag bit TMR1IF (PIR1<0>).



## 9.2 MSSP I<sup>2</sup>C Operation

The MSSP module in I<sup>2</sup>C 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.

Refer to Application Note AN578, "Use of the SSP Module in the I<sup>2</sup>C Multi-Master Environment."

A "glitch" filter is on the SCL and SDA pins when the pin is an input. This filter operates in both the 100 kHz and 400 kHz modes. In the 100 kHz mode, when these pins are an output, there is a slew rate control of the pin that is independent of device frequency.

**FIGURE 9-5: I<sup>2</sup>C SLAVE MODE BLOCK DIAGRAM**



Two pins are used for data transfer. These are the SCL pin, which is the clock, and the SDA pin, which is the data. The SDA and SCL pins are automatically configured when the I<sup>2</sup>C mode is enabled. The SSP module functions are enabled by setting SSP Enable bit SSPEN (SSPCON<5>).

The MSSP module has six registers for I<sup>2</sup>C operation. They are the:

- SSP Control Register (SSPCON)
- SSP Control Register2 (SSPCON2)
- SSP Status Register (SSPSTAT)
- Serial Receive/Transmit Buffer (SSPBUF)
- SSP Shift Register (SSPSR) - Not directly accessible
- SSP Address Register (SSPADD)

The SSPCON register allows control of the I<sup>2</sup>C operation. Four mode selection bits (SSPCON<3:0>) allow one of the following I<sup>2</sup>C modes to be selected:

- I<sup>2</sup>C Slave mode (7-bit address)
- I<sup>2</sup>C Slave mode (10-bit address)
- I<sup>2</sup>C Master mode, clock = OSC/4 (SSPADD +1)
- I<sup>2</sup>C firmware modes (provided for compatibility to other mid-range products)

Before selecting any I<sup>2</sup>C mode, the SCL and SDA pins must be programmed to inputs by setting the appropriate TRIS bits. Selecting an I<sup>2</sup>C mode by setting the SSPEN bit, enables the SCL and SDA pins to be used as the clock and data lines in I<sup>2</sup>C mode. Pull-up resistors must be provided externally to the SCL and SDA pins for the proper operation of the I<sup>2</sup>C module.

The CKE bit (SSPSTAT<6:7>) sets the levels of the SDA and SCL pins in either Master or Slave mode. When CKE = 1, the levels will conform to the SMBus specification. When CKE = 0, the levels will conform to the I<sup>2</sup>C specification.

The SSPSTAT register gives the status of the data transfer. This information includes detection of a START (S) or STOP (P) bit, specifies if the received byte was data or address, if the next byte is the completion of 10-bit address, and if this will be a read or write data transfer.

SSPBUF is the register to which the transfer data is written to, or read from. The SSPSR register shifts the data in or out of the device. In receive operations, the SSPBUF and SSPSR create a doubled buffered receiver. This allows reception of the next byte to begin before reading the last byte of received data. When the complete byte is received, it is transferred to the SSPBUF register and flag bit SSPIF is set. If another complete byte is received before the SSPBUF register is read, a receiver overflow has occurred and bit SSPOV (SSPCON<6>) is set and the byte in the SSPSR is lost.

The SSPADD register holds the slave address. In 10-bit mode, the user needs to write the high byte of the address (1111 0 A9 A8 0). Following the high byte address match, the low byte of the address needs to be loaded (A7:A0).

# PIC16F87X

## 9.2.1 SLAVE MODE

In Slave mode, the SCL and SDA pins must be configured as inputs. The MSSP module will override the input state with the output data, when required (slave-transmitter).

When an address is matched, or the data transfer after an address match is received, the hardware automatically will generate the Acknowledge ( $\overline{ACK}$ ) pulse, and then load the SSPBUF register with the received value currently in the SSPSR register.

There are certain conditions that will cause the MSSP module not to give this  $\overline{ACK}$  pulse. These are if either (or both):

- The buffer full bit BF (SSPSTAT<0>) was set before the transfer was received.
- The overflow bit SSPOV (SSPCON<6>) was set before the transfer was received.

If the BF bit is set, the SSPSR register value is not loaded into the SSPBUF, but bit SSPIF and SSPOV are set. Table 9-2 shows what happens when a data transfer byte is received, given the status of bits BF and SSPOV. The shaded cells show the condition where user software did not properly clear the overflow condition. Flag bit BF is cleared by reading the SSPBUF register, while bit SSPOV is cleared through software.

The SCL clock input must have a minimum high and low time for proper operation. The high and low times of the I<sup>2</sup>C specification, as well as the requirement of the MSSP module, is shown in timing parameter #100 and parameter #101 of the electrical specifications.

### 9.2.1.1 Addressing

Once the MSSP module has been enabled, it waits for a START condition to occur. Following the START condition, the 8-bits are shifted into the SSPSR register. All incoming bits are sampled with the rising edge of the clock (SCL) line. The value of register SSPSR<7:1> is compared to the value of the SSPADD register. The address is compared on the falling edge of the eighth clock (SCL) pulse. If the addresses match, and the BF and SSPOV bits are clear, the following events occur:

- The SSPSR register value is loaded into the SSPBUF register on the falling edge of the 8th SCL pulse.
- The buffer full bit, BF, is set on the falling edge of the 8th SCL pulse.
- An  $\overline{ACK}$  pulse is generated.
- SSP interrupt flag bit, SSPIF (PIR1<3>), is set (interrupt is generated if enabled) on the falling edge of the 9th SCL pulse.

In 10-bit address mode, two address bytes need to be received by the slave. The five Most Significant bits (MSBs) of the first address byte specify if this is a 10-bit address. Bit R/W (SSPSTAT<2>) must specify a write so the slave device will receive the second address byte.

For a 10-bit address, the first byte would equal '1111 0 A9 A8 0', where A9 and A8 are the two MSBs of the address. The sequence of events for a 10-bit address is as follows, with steps 7-9 for slave-transmitter:

- Receive first (high) byte of Address (bits SSPIF, BF and UA (SSPSTAT<1>) are set).
- Update the SSPADD register with the second (low) byte of Address (clears bit UA and releases the SCL line).
- Read the SSPBUF register (clears bit BF) and clear flag bit SSPIF.
- Receive second (low) byte of Address (bits SSPIF, BF and UA are set).
- Update the SSPADD register with the first (high) byte of Address. This will clear bit UA and release the SCL line.
- Read the SSPBUF register (clears bit BF) and clear flag bit SSPIF.
- Receive Repeated Start condition.
- Receive first (high) byte of Address (bits SSPIF and BF are set).
- Read the SSPBUF register (clears bit BF) and clear flag bit SSPIF.

**Note:** Following the Repeated START condition (step 7) in 10-bit mode, the user only needs to match the first 7-bit address. The user does not update the SSPADD for the second half of the address.

### 9.2.1.2 Slave Reception

When the R/W bit of the address byte is clear and an address match occurs, the R/W bit of the SSPSTAT register is cleared. The received address is loaded into the SSPBUF register.

When the address byte overflow condition exists, then no Acknowledge ( $\overline{ACK}$ ) pulse is given. An overflow condition is defined as either bit BF (SSPSTAT<0>) is set, or bit SSPOV (SSPCON<6>) is set. This is an error condition due to user firmware.

An SSP 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 received byte.

**Note:** The SSPBUF will be loaded if the SSPOV bit is set and the BF flag is cleared. If a read of the SSPBUF was performed, but the user did not clear the state of the SSPOV bit before the next receive occurred, the  $\overline{ACK}$  is not sent and the SSPBUF is updated.

## 10.1 USART Baud Rate Generator (BRG)

The BRG supports both the Asynchronous and Synchronous modes of the USART. It is a dedicated 8-bit baud rate generator. The SPBRG register controls the period of a free running 8-bit timer. In Asynchronous mode, bit BRGH (TXSTA<2>) also controls the baud rate. In Synchronous mode, bit BRGH is ignored. Table 10-1 shows the formula for computation of the baud rate for different USART modes which only apply in Master mode (internal clock).

Given the desired baud rate and Fosc, the nearest integer value for the SPBRG register can be calculated using the formula in Table 10-1. From this, the error in baud rate can be determined.

It may be advantageous to use the high baud rate (BRGH = 1), even for slower baud clocks. This is because the  $F_{osc}/(16(X + 1))$  equation can reduce the baud rate error in some cases.

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

### 10.1.1 SAMPLING

The data on the RC7/RX/DT pin is sampled three times by a majority detect circuit to determine if a high or a low level is present at the RX pin.

**TABLE 10-1: BAUD RATE FORMULA**

SYNC	BRGH = 0 (Low Speed)	BRGH = 1 (High Speed)
0	(Asynchronous) Baud Rate = $F_{osc}/(64(X+1))$	Baud Rate = $F_{osc}/(16(X+1))$
1	(Synchronous) Baud Rate = $F_{osc}/(4(X+1))$	N/A

X = value in SPBRG (0 to 255)

**TABLE 10-2: REGISTERS ASSOCIATED WITH BAUD RATE GENERATOR**

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Value on: POR, BOR	Value on all other RESETS	
98h	TXSTA	CSRC	TX9	TXEN	SYNC	—	BRGH	TRMT	TX9D	0000 -010	0000 -010	
18h	RCSTA	SPEN	RX9	SREN	CREN	ADDEN	FERR	OERR	RX9D	0000 000x	0000 000x	
99h	SPBRG	Baud Rate Generator Register									0000 0000	0000 0000

Legend: x = unknown, - = unimplemented, read as '0'. Shaded cells are not used by the BRG.

# PIC16F87X

---

NOTES:

**TABLE 12-4: STATUS BITS AND THEIR SIGNIFICANCE**

$\overline{\text{POR}}$	$\overline{\text{BOR}}$	$\overline{\text{TO}}$	$\overline{\text{PD}}$	
0	x	1	1	Power-on Reset
0	x	0	x	Illegal, $\overline{\text{TO}}$ is set on $\overline{\text{POR}}$
0	x	x	0	Illegal, $\overline{\text{PD}}$ is set on $\overline{\text{POR}}$
1	0	1	1	Brown-out Reset
1	1	0	1	WDT Reset
1	1	0	0	WDT Wake-up
1	1	u	u	MCLR Reset during normal operation
1	1	1	0	MCLR Reset during SLEEP or interrupt wake-up from SLEEP

Legend: x = don't care, u = unchanged

**TABLE 12-5: RESET CONDITION FOR SPECIAL REGISTERS**

Condition	Program Counter	STATUS Register	PCON Register
Power-on Reset	000h	0001 1xxx	---- --0x
MCLR Reset during normal operation	000h	000u uuuu	---- --uu
MCLR Reset during SLEEP	000h	0001 0uuu	---- --uu
WDT Reset	000h	0000 1uuu	---- --uu
WDT Wake-up	PC + 1	uuu0 0uuu	---- --uu
Brown-out Reset	000h	0001 1uuu	---- --u0
Interrupt wake-up from SLEEP	PC + 1 <sup>(1)</sup>	uuu1 0uuu	---- --uu

Legend: u = unchanged, x = unknown, - = unimplemented bit, read as '0'

**Note 1:** When the wake-up is due to an interrupt and the GIE bit is set, the PC is loaded with the interrupt vector (0004h).

## 14.4 MPLINK Object Linker/ MPLIB Object Librarian

The MPLINK object linker combines relocatable objects created by the MPASM assembler and the MPLAB C17 and MPLAB C18 C compilers. It can also link relocatable objects from pre-compiled libraries, using directives from a linker script.

The MPLIB object librarian is a librarian for pre-compiled code to be used with the MPLINK object linker. When a routine from a library is called from another 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 MPLIB object librarian manages the creation and modification of library files.

The MPLINK object linker features include:

- Integration with MPASM assembler and MPLAB C17 and MPLAB C18 C compilers.
- Allows all memory areas to be defined as sections to provide link-time flexibility.

The MPLIB object librarian features include:

- Easier linking because single libraries can be included instead of many smaller files.
- Helps keep code maintainable by grouping related modules together.
- Allows libraries to be created and modules to be added, listed, replaced, deleted or extracted.

## 14.5 MPLAB SIM Software Simulator

The MPLAB SIM software simulator allows code development in a PC-hosted environment by simulating the PIC MCU series microcontrollers on an instruction level. On any given instruction, the data areas can be examined or modified and stimuli can be applied from a file, or user-defined key press, to any of the pins. The execution can be performed in single step, execute until break, or trace mode.

The MPLAB SIM simulator fully supports symbolic debugging using the MPLAB C17 and the MPLAB C18 C compilers and the MPASM assembler. The software simulator offers the flexibility to develop and debug code outside of the laboratory environment, making it an excellent multi-project software development tool.

## 14.6 MPLAB ICE High Performance Universal In-Circuit Emulator with MPLAB IDE

The MPLAB ICE universal in-circuit emulator is intended to provide the product development engineer with a complete microcontroller design tool set for PIC MCU microcontrollers (MCUs). Software control of the MPLAB ICE in-circuit emulator is provided by the MPLAB Integrated Development Environment (IDE), which allows editing, building, downloading and source debugging from a single environment.

The MPLAB ICE 2000 is a full-featured emulator system with enhanced trace, trigger and data monitoring features. Interchangeable processor modules allow the system to be easily reconfigured for emulation of different processors. The universal architecture of the MPLAB ICE in-circuit emulator allows expansion to support new PIC microcontrollers.

The MPLAB ICE in-circuit emulator system has been designed as a real-time emulation system, with advanced features that are generally found on more expensive development tools. The PC platform and Microsoft® Windows environment were chosen to best make these features available to you, the end user.

## 14.7 ICEPIC In-Circuit Emulator

The ICEPIC low cost, in-circuit emulator is a solution for the Microchip Technology PIC16C5X, PIC16C6X, PIC16C7X and PIC16CXXX families of 8-bit One-Time-Programmable (OTP) microcontrollers. The modular system can support different subsets of PIC16C5X or PIC16CXXX products through the use of interchangeable personality modules, or daughter boards. The emulator is capable of emulating without target application circuitry being present.

# PIC16F87X

---

## 14.13 PICDEM 3 Low Cost PIC16CXXX Demonstration Board

The PICDEM 3 demonstration board is a simple demonstration board that supports the PIC16C923 and PIC16C924 in the PLCC package. It will also support future 44-pin PLCC microcontrollers with an LCD Module. All the necessary hardware and software is included to run the basic demonstration programs. The user can program the sample microcontrollers provided with the PICDEM 3 demonstration board on a PRO MATE II device programmer, or a PICSTART Plus development programmer with an adapter socket, and easily test firmware. The MPLAB ICE in-circuit emulator may also be used with the PICDEM 3 demonstration board to test firmware. A prototype area has been provided to the user for adding hardware and connecting it to the microcontroller socket(s). Some of the features include a RS-232 interface, push button switches, a potentiometer for simulated analog input, a thermistor and separate headers for connection to an external LCD module and a keypad. Also provided on the PICDEM 3 demonstration board is a LCD panel, with 4 commons and 12 segments, that is capable of displaying time, temperature and day of the week. The PICDEM 3 demonstration board provides an additional RS-232 interface and Windows software for showing the demultiplexed LCD signals on a PC. A simple serial interface allows the user to construct a hardware demultiplexer for the LCD signals.

## 14.14 PICDEM 17 Demonstration Board

The PICDEM 17 demonstration board is an evaluation board that demonstrates the capabilities of several Microchip microcontrollers, including PIC17C752, PIC17C756A, PIC17C762 and PIC17C766. All necessary hardware is included to run basic demo programs, which are supplied on a 3.5-inch disk. A programmed sample is included and the user may erase it and program it with the other sample programs using the PRO MATE II device programmer, or the PICSTART Plus development programmer, and easily debug and test the sample code. In addition, the PICDEM 17 demonstration board supports downloading of programs to and executing out of external FLASH memory on board. The PICDEM 17 demonstration board is also usable with the MPLAB ICE in-circuit emulator, or the PICMASTER emulator and all of the sample programs can be run and modified using either emulator. Additionally, a generous prototype area is available for user hardware.

## 14.15 KEELOQ Evaluation and Programming Tools

KEELOQ evaluation and programming tools support Microchip's HCS Secure Data Products. The HCS evaluation kit includes a LCD display to show changing codes, a decoder to decode transmissions and a programming interface to program test transmitters.

# PIC16F87X

## 15.4 DC Characteristics: PIC16F873/874/876/877-04 (Extended) PIC16F873/874/876/877-10 (Extended) (Continued)

DC CHARACTERISTICS		Standard Operating Conditions (unless otherwise stated) Operating temperature $-40^{\circ}\text{C} \leq T_A \leq +125^{\circ}\text{C}$ Operating voltage $V_{DD}$ range as described in DC specification (Section 15.1)					
Param No.	Sym	Characteristic	Min	Typ†	Max	Units	Conditions
		<b>Output Low Voltage</b>					
D080A	VOL	I/O ports	—	—	0.6	V	$I_{OL} = 7.0\text{ mA}$ , $V_{DD} = 4.5\text{V}$
D083A		OSC2/CLKOUT (RC osc config)	—	—	0.6	V	$I_{OL} = 1.2\text{ mA}$ , $V_{DD} = 4.5\text{V}$
		<b>Output High Voltage</b>					
D090A	VOH	I/O ports <sup>(3)</sup>	$V_{DD} - 0.7$	—	—	V	$I_{OH} = -2.5\text{ mA}$ , $V_{DD} = 4.5\text{V}$
D092A		OSC2/CLKOUT (RC osc config)	$V_{DD} - 0.7$	—	—	V	$I_{OH} = -1.0\text{ mA}$ , $V_{DD} = 4.5\text{V}$
D150*	VOD	<b>Open Drain High Voltage</b>	—	—	8.5	V	RA4 pin
		<b>Capacitive Loading Specs on Output Pins</b>					
D100	Cosc2	OSC2 pin	—	—	15	pF	In XT, HS and LP modes when external clock is used to drive OSC1
D101	CI0	All I/O pins and OSC2 (RC mode)	—	—	50	pF	
D102	CB	SCL, SDA (I <sup>2</sup> C mode)	—	—	400	pF	
		<b>Data EEPROM Memory</b>					
D120	ED	Endurance	100K	—	—	E/W	25°C at 5V
D121	VDRW	VDD for read/write	V <sub>MIN</sub>	—	5.5	V	Using EECON to read/write $V_{MIN}$ = min. operating voltage
D122	TDEW	Erase/write cycle time	—	4	8	ms	
		<b>Program FLASH Memory</b>					
D130	EP	Endurance	1000	—	—	E/W	25°C at 5V
D131	VPR	VDD for read	V <sub>MIN</sub>	—	5.5	V	$V_{MIN}$ = min operating voltage
D132A		VDD for erase/write	V <sub>MIN</sub>	—	5.5	V	Using EECON to read/write, $V_{MIN}$ = min. operating voltage
D133	TPEW	Erase/Write cycle time	—	4	8	ms	

\* These parameters are characterized but not tested.

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

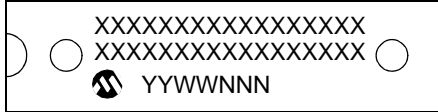
- Note 1:** In RC oscillator configuration, the OSC1/CLKIN pin is a Schmitt Trigger input. It is not recommended that the PIC16F87X be driven with external clock in RC mode.
- 2:** The leakage current on the MCLR pin is strongly dependent on the applied voltage level. The specified levels represent normal operating conditions. Higher leakage current may be measured at different input voltages.
- 3:** Negative current is defined as current sourced by the pin.



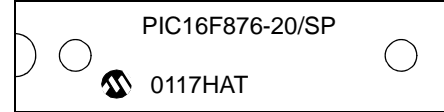
## 17.0 PACKAGING INFORMATION

### 17.1 Package Marking Information

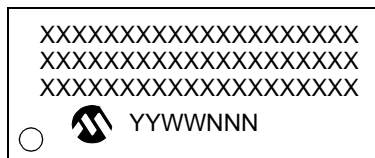
#### 28-Lead PDIP (Skinny DIP)



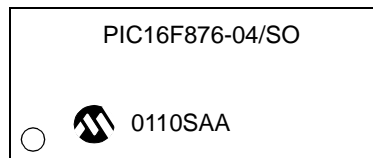
#### Example



#### 28-Lead SOIC



#### Example



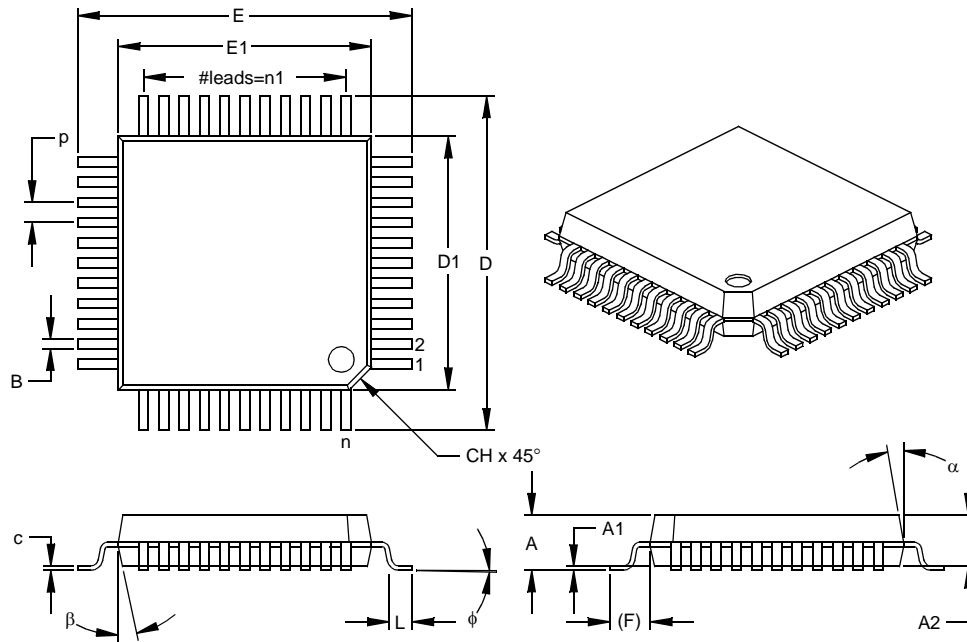
<b>Legend:</b>	XX...X	Customer-specific information
	Y	Year code (last digit of calendar year)
	YY	Year code (last 2 digits of calendar year)
	WW	Week code (week of January 1 is week '01')
	NNN	Alphanumeric traceability code
	(e3)	Pb-free JEDEC designator for Matte Tin (Sn)
	*	This package is Pb-free. The Pb-free JEDEC designator (e3) can be found on the outer packaging for this package.

**Note:** In the event the full Microchip part number cannot be marked on one line, it will be carried over to the next line, thus limiting the number of available characters for customer-specific information.

# PIC16F87X

## 44-Lead Plastic Metric Quad Flatpack (PQ) 10x10x2 mm Body, 1.6/0.15 mm Lead Form (MQFP)

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



Dimension Limits	Units	INCHES			MILLIMETERS*		
		MIN	NOM	MAX	MIN	NOM	MAX
Number of Pins	n		44			44	
Pitch	p		.031			0.80	
Pins per Side	n1		11			11	
Overall Height	A	.079	.086	.093	2.00	2.18	2.35
Molded Package Thickness	A2	.077	.080	.083	1.95	2.03	2.10
Standoff §	A1	.002	.006	.010	0.05	0.15	0.25
Foot Length	L	.029	.035	.041	0.73	0.88	1.03
Footprint (Reference)	(F)		.063			1.60	
Foot Angle	φ	0	3.5	7	0	3.5	7
Overall Width	E	.510	.520	.530	12.95	13.20	13.45
Overall Length	D	.510	.520	.530	12.95	13.20	13.45
Molded Package Width	E1	.390	.394	.398	9.90	10.00	10.10
Molded Package Length	D1	.390	.394	.398	9.90	10.00	10.10
Lead Thickness	c	.005	.007	.009	0.13	0.18	0.23
Lead Width	B	.012	.015	.018	0.30	0.38	0.45
Pin 1 Corner Chamfer	CH	.025	.035	.045	0.64	0.89	1.14
Mold Draft Angle Top	α	5	10	15	5	10	15
Mold Draft Angle Bottom	β	5	10	15	5	10	15

\* Controlling Parameter  
§ Significant Characteristic

**Notes:**

Dimensions D1 and E1 do not include mold flash or protrusions. Mold flash or protrusions shall not exceed .010" (0.254mm) per side.

JEDEC Equivalent: MS-022

Drawing No. C04-071

# PIC16F87X

PWM Mode .....	61
Block Diagram .....	61
Duty Cycle .....	61
Example Frequencies/Resolutions (Table) .....	62
PWM Period .....	61
Special Event Trigger and A/D Conversions .....	60
CCP. See Capture/Compare/PWM	
CCP1CON .....	17
CCP2CON .....	17
CCPR1H Register .....	15, 17, 57
CCPR1L Register .....	17, 57
CCPR2H Register .....	15, 17
CCPR2L Register .....	15, 17
CCPxM0 bit .....	58
CCPxM1 bit .....	58
CCPxM2 bit .....	58
CCPxM3 bit .....	58
CCPxX bit .....	58
CCPxY bit .....	58
CKE .....	66
CKP .....	67
Clock Polarity Select bit, CKP .....	67
Code Examples	
Call of a Subroutine in Page 1 from Page 0 .....	26
EEPROM Data Read .....	43
EEPROM Data Write .....	43
FLASH Program Read .....	44
FLASH Program Write .....	45
Indirect Addressing .....	27
Initializing PORTA .....	29
Saving STATUS, W and PCLATH Registers .....	130
Code Protected Operation	
Data EEPROM and FLASH Program Memory .....	45
Code Protection .....	119, 133
Computed GOTO .....	26
Configuration Bits .....	119
Configuration Word .....	120
Conversion Considerations .....	198
<b>D</b>	
D $\bar{A}$ .....	66
Data EEPROM .....	41
Associated Registers .....	46
Code Protection .....	45
Reading .....	43
Special Functions Registers .....	41
Spurious Write Protection .....	45
Write Verify .....	45
Writing to .....	43
Data Memory .....	12
Bank Select (RP1:RP0 Bits) .....	12, 18
General Purpose Registers .....	12
Register File Map .....	13, 14
Special Function Registers .....	15
Data/Address bit, D $\bar{A}$ .....	66
DC and AC Characteristics Graphs and Tables .....	177
DC Characteristics	
Commercial and Industrial .....	152–156
Extended .....	157–160
Development Support .....	143
Device Differences .....	197
Device Overview .....	5
Direct Addressing .....	27
<b>E</b>	
Electrical Characteristics .....	149
Errata .....	4
External Clock Input (RA4/T0CKI). See Timer0	
External Interrupt Input (RB0/INT). See Interrupt Sources	
<b>F</b>	
Firmware Instructions .....	135
FLASH Program Memory .....	41
Associated Registers .....	46
Code Protection .....	45
Configuration Bits and Read/Write State .....	46
Reading .....	44
Special Function Registers .....	41
Spurious Write Protection .....	45
Write Protection .....	46
Write Verify .....	45
Writing to .....	44
FSR Register .....	15, 16, 17, 27
<b>G</b>	
General Call Address Sequence .....	76
General Call Address Support .....	76
General Call Enable bit .....	68
<b>I</b>	
I/O Ports .....	29
I <sup>2</sup> C .....	73
I <sup>2</sup> C Bus	
Connection Considerations .....	94
Sample Device Configuration .....	94
I <sup>2</sup> C Master Mode Reception .....	84
I <sup>2</sup> C Master Mode Repeated START Condition .....	81
I <sup>2</sup> C Mode Selection .....	73
I <sup>2</sup> C Module	
Acknowledge Sequence Timing .....	86
Addressing .....	74
Associated Registers .....	77
Baud Rate Generator .....	79
Block Diagram .....	78
BRG Block Diagram .....	79
BRG Reset due to SDA Collision .....	91
BRG Timing .....	80
Bus Arbitration .....	89
Bus Collision .....	89
Acknowledge .....	89
Repeated START Condition .....	92
Repeated START Condition Timing	
(Case1) .....	92
Repeated START Condition Timing	
(Case2) .....	92
START Condition .....	90
START Condition Timing .....	90, 91
STOP Condition .....	93
STOP Condition Timing (Case1) .....	93
STOP Condition Timing (Case2) .....	93
Transmit Timing .....	89
Bus Collision Timing .....	89
Clock Arbitration .....	88
Clock Arbitration Timing (Master Transmit) .....	88
Conditions to not give ACK Pulse .....	74
General Call Address Support .....	76
Master Mode .....	78
Master Mode 7-bit Reception Timing .....	85
Master Mode Block Diagram .....	78

# PIC16F87X

## O

On-Line Support .....	207
OPCODE Field Descriptions .....	135
OPTION_REG Register .....	19, 48
INTEDG Bit .....	19
PS2:PS0 Bits .....	19
PSA Bit .....	19
T0CS Bit .....	19
T0SE Bit .....	19
OSC1/CLKIN Pin .....	7, 8
OSC2/CLKOUT Pin .....	7, 8
Oscillator Configuration .....	119
HS .....	121, 124
LP .....	121, 124
RC .....	121, 122, 124
XT .....	121, 124
Oscillator, WDT .....	131
Oscillators	
Capacitor Selection .....	122
Crystal and Ceramic Resonators .....	121
RC .....	122

## P

P (STOP bit) .....	66
Package Marking Information .....	189
Packaging Information .....	189
Paging, Program Memory .....	11, 26
Parallel Slave Port (PSP) .....	9, 35, 38
Associated Registers .....	39
Block Diagram .....	38
RE0/RD/AN5 Pin .....	9, 36, 38
RE1/WR/AN6 Pin .....	9, 36, 38
RE2/CS/AN7 Pin .....	9, 36, 38
Read Waveforms .....	39
Select (PSPMODE Bit) .....	35, 36, 37, 38
Write Waveforms .....	39
PCL Register .....	15, 16, 26
PCLATH Register .....	15, 16, 17, 26
PCON Register .....	25, 124
BOR Bit .....	25
POR Bit .....	25
PIC16F876 Pinout Description .....	7
PIC16F87X Product Identification System .....	209
PICDEM 1 Low Cost PIC MCU	
Demonstration Board .....	145
PICDEM 17 Demonstration Board .....	146
PICDEM 2 Low Cost PIC16CXX	
Demonstration Board .....	145
PICDEM 3 Low Cost PIC16CXXX	
Demonstration Board .....	146
PICSTART Plus Entry Level	
Development Programmer .....	145
PIE1 Register .....	21
PIE2 Register .....	23
Pinout Descriptions	
PIC16F873/PIC16F876 .....	7
PIC16F874/PIC16F877 .....	8
PIR1 Register .....	22
PIR2 Register .....	24
POP .....	26
POR. See Power-on Reset	

PORTA .....	7, 8, 17
Analog Port Pins .....	7, 8
Associated Registers .....	30
Block Diagram	
RA3:RA0 and RA5 Pins .....	29
RA4/T0CKI Pin .....	29
Initialization .....	29
PORTA Register .....	15, 29
RA3	
RA0 and RA5 Port Pins .....	29
RA4/T0CKI Pin .....	7, 8
RA5/SS/AN4 Pin .....	7, 8
TRISA Register .....	29
PORTB .....	7, 8, 17
Associated Registers .....	32
Block Diagram	
RB3:RB0 Port Pins .....	31
RB7:RB4 Port Pins .....	31
PORTB Register .....	15, 31
RB0/INT Edge Select (INTEDG Bit) .....	19
RB0/INT Pin, External .....	7, 8, 130
RB7:RB4 Interrupt on Change .....	130
RB7:RB4 Interrupt on Change Enable	
(RBIE Bit) .....	130
RB7:RB4 Interrupt on Change Flag	
(RBIF Bit) .....	130
RB7:RB4 Interrupt-on-Change Enable	
(RBIE Bit) .....	20
RB7:RB4 Interrupt-on-Change Flag	
(RBIF Bit) .....	20, 31
TRISB Register .....	17, 31
PORTC .....	7, 9, 17
Associated Registers .....	34
Block Diagrams	
Peripheral Output Override	
(RC 0:2, 5:7) .....	33
Peripheral Output Override	
(RC 3:4) .....	33
PORTC Register .....	15, 33
RC0/T1OSO/T1CKI Pin .....	7, 9
RC1/T1OSI/CCP2 Pin .....	7, 9
RC2/CCP1 Pin .....	7, 9
RC3/SCK/SCL Pin .....	7, 9
RC4/SDI/SDA Pin .....	7, 9
RC5/SDO Pin .....	7, 9
RC6/TX/CK Pin .....	7, 9, 96
RC7/RX/DT Pin .....	7, 9, 96, 97
TRISC Register .....	33, 95
PORTD .....	9, 17, 38
Associated Registers .....	35
Block Diagram .....	35
Parallel Slave Port (PSP) Function .....	35
PORTD Register .....	15, 35
TRISD Register .....	35

# PIC16F87X

---

NOTES: