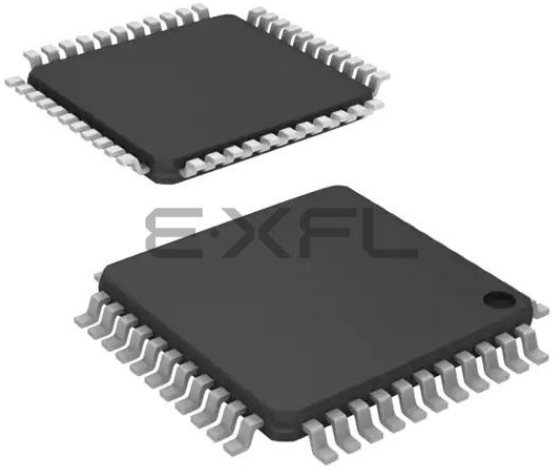


Welcome to E-XFL.COM

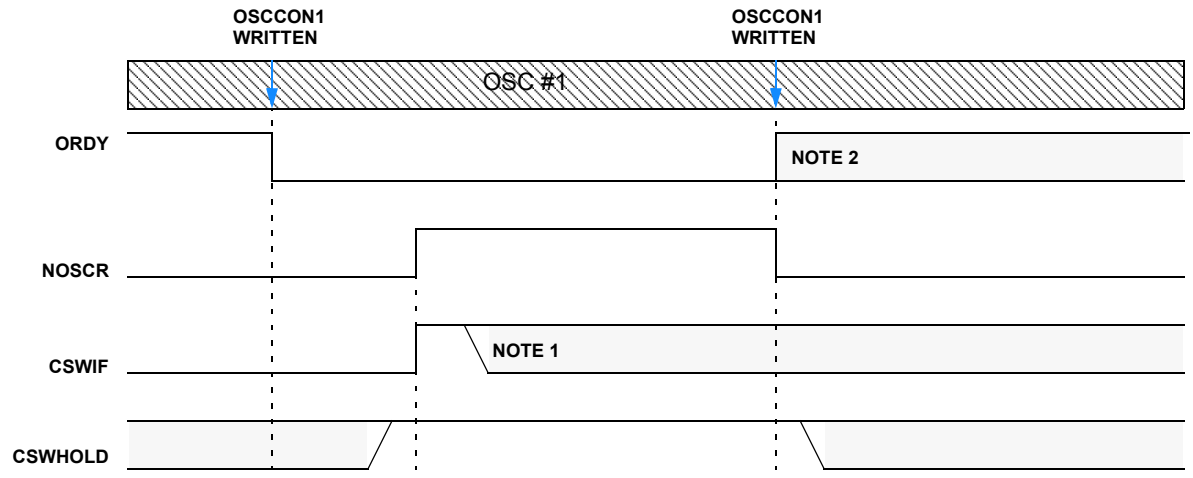
What is "Embedded - Microcontrollers"?

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

Applications of "Embedded - Microcontrollers"

Details	
Product Status	Active
Core Processor	PIC
Core Size	8-Bit
Speed	64MHz
Connectivity	I²C, LINbus, SPI, UART/USART
Peripherals	Brown-out Detect/Reset, DMA, HLVD, POR, PWM, WDT
Number of I/O	36
Program Memory Size	32KB (16K x 16)
Program Memory Type	FLASH
EEPROM Size	256 x 8
RAM Size	2K x 8
Voltage - Supply (Vcc/Vdd)	2.3V ~ 5.5V
Data Converters	A/D 35x12b; D/A 1x5b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 125°C (TA)
Mounting Type	Surface Mount
Package / Case	44-TQFP
Supplier Device Package	44-TQFP (10x10)
Purchase URL	https://www.e-xfl.com/product-detail/microchip-technology/pic18f45k42-e-pt

FIGURE 7-8: CLOCK SWITCH ABANDONED



Note 1: CSWIF may be cleared before or after rewriting OSCCON1; CSWIF is not automatically cleared.
2: ORDY = 0 if OSCCON1 does not match OSCCON2; a new switch will begin.

9.3 Interrupt Priority

The final priority level for any pending source of interrupt is determined first by the user-assigned priority of that source in the IPRx register, then by the natural order priority within the IVT. The sections below detail the operation of Interrupt priorities.

9.3.1 USER (SOFTWARE) PRIORITY

User-assigned interrupt priority is enabled by setting the IPEN bit in the INTCON0 register (Register 9-1). Each peripheral interrupt source can be assigned a high or low priority level by the user. The user-assignable interrupt priority control bits for each interrupt are located in the IPRx registers (Registers 9-25 through 9-35).

The interrupts are serviced based on predefined interrupt priority scheme defined below.

1. Interrupts set by the user as high-priority interrupt have higher precedence of execution. High-priority interrupts will override a low-priority request when:
   a) A low priority interrupt has been requested or its request is already pending.
   b) A low- and high-priority interrupt are triggered concurrently, i.e., on the same instruction cycle**(1)**.
   c) A low-priority interrupt was requested and the corresponding Interrupt Service Routine is currently executing. In this case, the lower priority interrupt routine will complete executing after the high-priority interrupt has been serviced**(2)**.
2. Interrupts set by the user as a low priority have the lower priority of execution and are preempted by any high-priority interrupt.
3. Interrupts defined with the same software priority cannot preempt or interrupt each other. Concurrent pending interrupts with the same user priority are resolved using the natural order priority. (when MVECEN = ON) or in the order the interrupt flag bits are polled in the ISR (when MVECEN = OFF).

Note 1: When a high priority interrupt preempts a concurrent low priority interrupt, the GIEL bit may be cleared in the high priority Interrupt Service Routine. If the GIEL bit is cleared, the low priority interrupt will NOT be serviced even if it was originally requested. The corresponding interrupt flag needs to be cleared in user code.

2: When a high priority interrupt is requested while a low priority Interrupt Service Routine is executing, the GIEL bit may be cleared in the high priority Interrupt Service Routine. The pending low priority interrupt will resume even if the GIEL bit is cleared.

REGISTER 9-26: IPR1: PERIPHERAL INTERRUPT PRIORITY REGISTER 1

R/W-1/1	R/W-1/1	R/W-1/1	R/W-1/1	R/W-1/1	R/W-1/1	R/W-1/1	R/W-1/1
SMT1PWAIP	SMT1PRAIP	SMT1IP	C1IP	ADTIP	ADIP	ZCDIP	INT0IP
bit 7							bit 0

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

bit 7 **SMT1PWAIP:** SMT1 Pulse Width Acquisition Interrupt Priority bit
1 = High priority
0 = Low priority

bit 6 **SMT1PRAIP:** SMT1 Period Acquisition Interrupt Priority bit
1 = High priority
0 = Low priority

bit 5 **SMT1IP:** SMT1 Interrupt Priority bit
1 = High priority
0 = Low priority

bit 4 **C1IP:** C1 Interrupt Priority bit
1 = High priority
0 = Low priority

bit 3 **ADTIP:** ADC Threshold Interrupt Priority bit
1 = High priority
0 = Low priority

bit 2 **ADIP:** ADC Interrupt Priority bit
1 = High priority
0 = Low priority

bit 1 **ZCDIP:** ZCD Interrupt Priority bit
1 = High priority
0 = Low priority

bit 0 **INT0IP:** External Interrupt 0 Interrupt Priority bit
1 = High priority
0 = Low priority

REGISTER 9-28: IPR3: PERIPHERAL INTERRUPT PRIORITY REGISTER 3

R/W-1/1	R/W-1/1	R/W-1/1	R/W-1/1	R/W-1/1	R/W-1/1	R/W-1/1	R/W-1/1
TMR0IP	U1IP	U1EIP	U1TXIP	U1RXIP	I2C1EIP	I2C1IP	I2C1TXIP
bit 7							bit 0

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

bit 7 **TMR0IP:** TMR0 Interrupt Priority bit
1 = High priority
0 = Low priority

bit 6 **U1IP:** UART1 Interrupt Priority bit
1 = High priority
0 = Low priority

bit 5 **U1EIP:** UART1 Framing Error Interrupt Priority bit
1 = High priority
0 = Low priority

bit 4 **U1TXIP:** UART1 Transmit Interrupt Priority bit
1 = High priority
0 = Low priority

bit 3 **U1RXIP:** UART1 Receive Interrupt Priority bit
1 = High priority
0 = Low priority

bit 2 **I2C1EIP:** I^2C1 Error Interrupt Priority bit
1 = High priority
0 = Low priority

bit 1 **I2C1IP:** I^2C1 Interrupt Priority bit
1 = High priority
0 = Low priority

bit 0 **I2C1TXIP:** I^2C1 Transmit Interrupt Priority bit
1 = High priority
0 = Low priority

REGISTER 11-3: WDTPSL: WWDT PRESCALE SELECT LOW BYTE REGISTER (READ-ONLY)

R-0/0	R-0/0	R-0/0	R-0/0	R-0/0	R-0/0	R-0/0	R-0/0
PSCNT<7:0>							
bit 7							bit 0

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

bit 7-0 **PSCNT<7:0>:** Prescale Select Low Byte bits(1)

Note 1: The 18-bit WDT prescale value, PSCNT<17:0> includes the WDTPSL, WDTPSH and the lower bits of the WDTTMR registers. PSCNT<17:0> is intended for debug operations and should not be read during normal operation.

REGISTER 11-4: WDTPSH: WWDT PRESCALE SELECT HIGH BYTE REGISTER (READ-ONLY)

R-0/0	R-0/0	R-0/0	R-0/0	R-0/0	R-0/0	R-0/0	R-0/0
PSCNT<15:8>							
bit 7							bit 0

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

bit 7-0 **PSCNT<15:8>:** Prescale Select High Byte bits(1)

Note 1: The 18-bit WDT prescale value, PSCNT<17:0> includes the WDTPSL, WDTPSH and the lower bits of the WDTTMR registers. PSCNT<17:0> is intended for debug operations and should not be read during normal operation.

TABLE 14-1: SCANNER OPERATING MODES(1)

TRIGEN	BURSTMD	Scanner Operation
0	0	Memory access is requested when the CRC module is ready to accept data; the request is granted if no other higher priority source request is pending.
1	0	Memory access is requested when the CRC module is ready to accept data and trigger selection is true; the request is granted if no other higher priority source request is pending.
x	1	Memory access is always requested, the request is granted if no other higher priority source request is pending.

Note 1: See **Section 3.1 "System Arbitration"** for Priority selection and **Section 3.2 "Memory Access Scheme"** for Memory Access Scheme.

REGISTER 14-12: SCANLADRU: SCAN LOW ADDRESS UPPER BYTE REGISTER

U-0	U-0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0
—	—	LADR<21:16>(1,2)					
bit 7							bit 0

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

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

bit 5-0 **LADR<21:16>:** Scan Start/Current Address bits(1,2)
Upper bits of the current address to be fetched from, value increments on each fetch of memory.

Note 1: Registers SCANLADRU/H/L form a 22-bit value, but are not guarded for atomic or asynchronous access; registers should only be read or written while SGO = 0 (SCANCON0 register).
2: While SGO = 1 (SCANCON0 register), writing to this register is ignored.

REGISTER 14-13: SCANLADRH: SCAN LOW ADDRESS HIGH BYTE REGISTER

R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0
LADR<15:8>(1, 2)							
bit 7							bit 0

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

bit 7-0 **LADR<15:8>**: Scan Start/Current Address bits(1, 2)
Most Significant bits of the current address to be fetched from, value increments on each fetch of memory.

Note 1: Registers SCANLADRU/H/L form a 22-bit value, but are not guarded for atomic or asynchronous access; registers should only be read or written while SGO = 0 (SCANCON0 register).
2: While SGO = 1 (SCANCON0 register), writing to this register is ignored.

17.3 Bidirectional Pins

PPS selections for peripherals with bidirectional signals on a single pin must be made so that the PPS input and PPS output select the same pin. Peripherals that have bidirectional signals include:

- I^2C

Note: Refer to Table 17-1 for pins that are I^2C compatible. Clock and data signals can be routed to any pin, however pins without I^2C compatibility will operate at standard TTL/ST logic levels as selected by the INVLV register.

17.4 PPS Lock

The PPS includes a mode in which all input and output selections can be locked to prevent inadvertent changes. PPS selections are locked by setting the PPSLOCKED bit of the PPSLOCK register. Setting and clearing this bit requires a special sequence as an extra precaution against inadvertent changes. Examples of setting and clearing the PPSLOCKED bit are shown in Example 17-1.

EXAMPLE 17-1: PPS LOCK SEQUENCE

```
; Disable interrupts:
    BCF      INTCON0,GIE

; Bank to PPSLOCK register
    BANKSEL  PPSLOCK
    MOVLB    PPSLOCK
    MOVLW    55h

; Required sequence, next 4 instructions
    MOVWF    PPSLOCK
    MOVLW    AAh
    MOVWF    PPSLOCK

; Set PPSLOCKED bit to disable writes
; Only a BSF instruction will work
    BSF      PPSLOCK,0

; Enable Interrupts
    BSF      INTCON0,GIE
```

EXAMPLE 17-2: PPS UNLOCK SEQUENCE

```
; Disable interrupts:
    BCF      INTCON0,GIE

; Bank to PPSLOCK register
    BANKSEL  PPSLOCK
    MOVLB    PPSLOCK
    MOVLW    55h

; Required sequence, next 4 instructions
    MOVWF    PPSLOCK
    MOVLW    AAh
    MOVWF    PPSLOCK

; Clear PPSLOCKED bit to enable writes
; Only a BCF instruction will work
    BCF      PPSLOCK,0

; Enable Interrupts
    BSF      INTCON0,GIE
```

17.5 PPS One-way Lock

When this bit is set, the PPSLOCKED bit can only be cleared and set one time after a device Reset. This allows for clearing the PPSLOCKED bit so that the input and output selections can be made during initialization. When the PPSLOCKED bit is set after all selections have been made, it will remain set and cannot be cleared until after the next device Reset event.

17.6 Operation During Sleep

PPS input and output selections are unaffected by Sleep.

17.7 Effects of a Reset

A device Power-on-Reset (POR) clears all PPS input and output selections to their default values. All other Resets leave the selections unchanged. Default input selections are shown in pin allocation Table 1. The PPS one-way lock is also removed.

31.2.1.1 Enabling the Transmitter

The UART transmitter is enabled for asynchronous operations by configuring the following control bits:

- TXEN = 1
- MODE<3:0> = 0h through 3h
- UxBRGH:L = desired baud rate
- UxBRGS = desired baud rate multiplier
- RxyPPS = code for desired output pin
- ON = 1

All other UART control bits are assumed to be in their default state.

Setting the TXEN bit in the UxCON0 register enables the transmitter circuitry of the UART. The MODE<3:0> bits in the UxCON0 register select the desired mode. Setting the ON bit in the UxCON1 register enables the UART. When TXEN is set and the transmitter is not idle, the TX pin is automatically configured as an output. When the transmitter is idle, the TX pin drive is relinquished to the port TRIS control. If the TX pin is shared with an analog peripheral, the analog I/O function should be disabled by clearing the corresponding ANSEL bit.

Note:	The UxTXIF Transmitter Interrupt flag is set when the TXEN enable bit is set and the UxTXB register can accept data.

31.2.1.2 Transmitting Data

A transmission is initiated by writing a character to the UxTXB register. If this is the first character, or the previous character has been completely transmitted from the TSR, the data in the UxTXB is immediately transferred to the TSR register. If the TSR still contains all or part of a previous character, the new character data is held in the UxTXB until the previous character transmission is complete. The pending character in the UxTXB is then transferred to the TSR at the beginning of the previous character Stop bit transmission. The transmission of the Start bit, data bits and Stop bit sequence commences immediately following the completion of all of the previous character's Stop bits.

31.2.1.3 Transmit Data Polarity

The polarity of the transmit data is controlled with the TXPOL bit in the UxCON2 register. The default state of this bit is '0' which selects high true transmit idle and data bits. Setting the TXPOL bit to '1' will invert the transmit data, resulting in low true idle and data bits. The TXPOL bit controls transmit data polarity in all modes.

31.2.1.4 Transmit Interrupt Flag

The UxTXIF interrupt flag bit in the PIR register is set whenever the UART transmitter is enabled and no character is being held for transmission in the UxTXB. In other words, the UxTXIF bit is clear only when the TSR is busy with a character and a new character has been queued for transmission in the UxTXB.

The UxTXIF interrupt can be enabled by setting the UxTXIE interrupt enable bit in the PIE register. However, the UxTXIF flag bit will be set whenever the UxTXB is empty, regardless of the state of UxTXIE enable bit.The UxTXIF bit is read-only and cannot be set or cleared by software.

To use interrupts when transmitting data, set the UxTXIE bit only when there is more data to send. Clear the UxTXIE interrupt enable bit upon writing UxTXB with the last character of the transmission.

31.2.1.5 TSR Status

The TXMTIF bit in the UxERRIR register indicates the status of the TSR. This is a read-only bit. The TXMTIF bit is set when the TSR is empty and idle. The TXMTIF bit is cleared when a character is transferred to the TSR from the UxTXB. The TXMTIF bit remains clear until all bits, including the Stop bits, have been shifted out of the TSR and a byte is not waiting in the UxTXB register.

The TXMTIF will generate an interrupt when the TXMTIE bit in the UxERRIE register is set.

Note:	The TSR is not mapped in data memory, so it is not available to the user.

31.2.1.6 Transmitter 7-bit Mode

7-Bit mode is selected when the MODE<3:0> bits are set to '0001'. In 7-bit mode, only the seven Least Significant bits of the data written to UxTXB are transmitted. The Most Significant bit is ignored.

31.2.1.7 Transmitter Parity Modes

When the Odd or even Parity mode is selected, all data is sent as nine bits. The first eight bits are data and the 9th bit is parity. Even and odd parity is selected when the MODE<3:0> bits are set to '0011' and '0010', respectively. Parity is automatically determined by the module and inserted in the serial data stream.

32.6.3 SLAVE MODE SLAVE SELECT

In Slave mode, an external Slave Select Signal can be used to synchronize communication with the Master device. The Slave Select line is held in its inactive state (high by default) until the master device is ready to communicate. When the Slave Select transitions to its active state, the slave knows that a new transmission is starting.

When the Slave Select goes false at the end of the transmission the receive function of the selected SPI Slave device returns to the inactive state. The slave is then ready to receive a new transmission when the Slave Select goes True again.

The Slave Select signal is received on the $\overline{SS}$ input pin. This pin is remappable with the SPIxSSPPS register (see **Section 17.1 "PPS Inputs"**). When the input on this pin is true, transmission and reception are enabled, and the SDO pin is driven. When the input on this pin is false, the SDO pin is either tri-stated (if the TRIS bit associated with the SDO pin is set) or driven to the value of the LAT bit associated with the SDO pin (if the TRIS bit associated with the SDO pin is cleared). In addition, the SCK input is ignored.

If the SS input goes False, while a data transfer is still in progress, it is considered a slave select fault. The SSFLT bit of SPIxCON2 indicates whether such an event has occurred. The transfer counter value determines the number of bits in a valid data transfer (see **Section 32.4 "Transfer Counter"** for more details).

The Slave Select polarity is controlled by the SSP bit of SPIxCON1. When SSP is set (its default state), the Slave Select input is active-low, and when it is cleared, the Slave Select input is active-high.

The Slave Select for the SPI module is controlled by the SSET bit of SPIxCON2. When the bit is cleared (its default state), the slave select will act as described above. When the bit is set, the SPI module will behave as if the SS input was always in its active state.

Note:	When SSET is set, the effective SS(in) signal is always active. Hence, the SSFLT bit may be disregarded.

32.6.4 SLAVE MODE CLOCK CONFIGURATION

In Slave Mode, SCK is an input, and must be configured to the same polarity and clock edge as the master device. As in Master mode, the polarity of the clock input is controlled by the CKP bit of SPIxCON1 and the clock edge used for transmitting data is controlled by the CKE bit of SPIxCON1.

32.6.5 DAISY-CHAIN CONFIGURATION

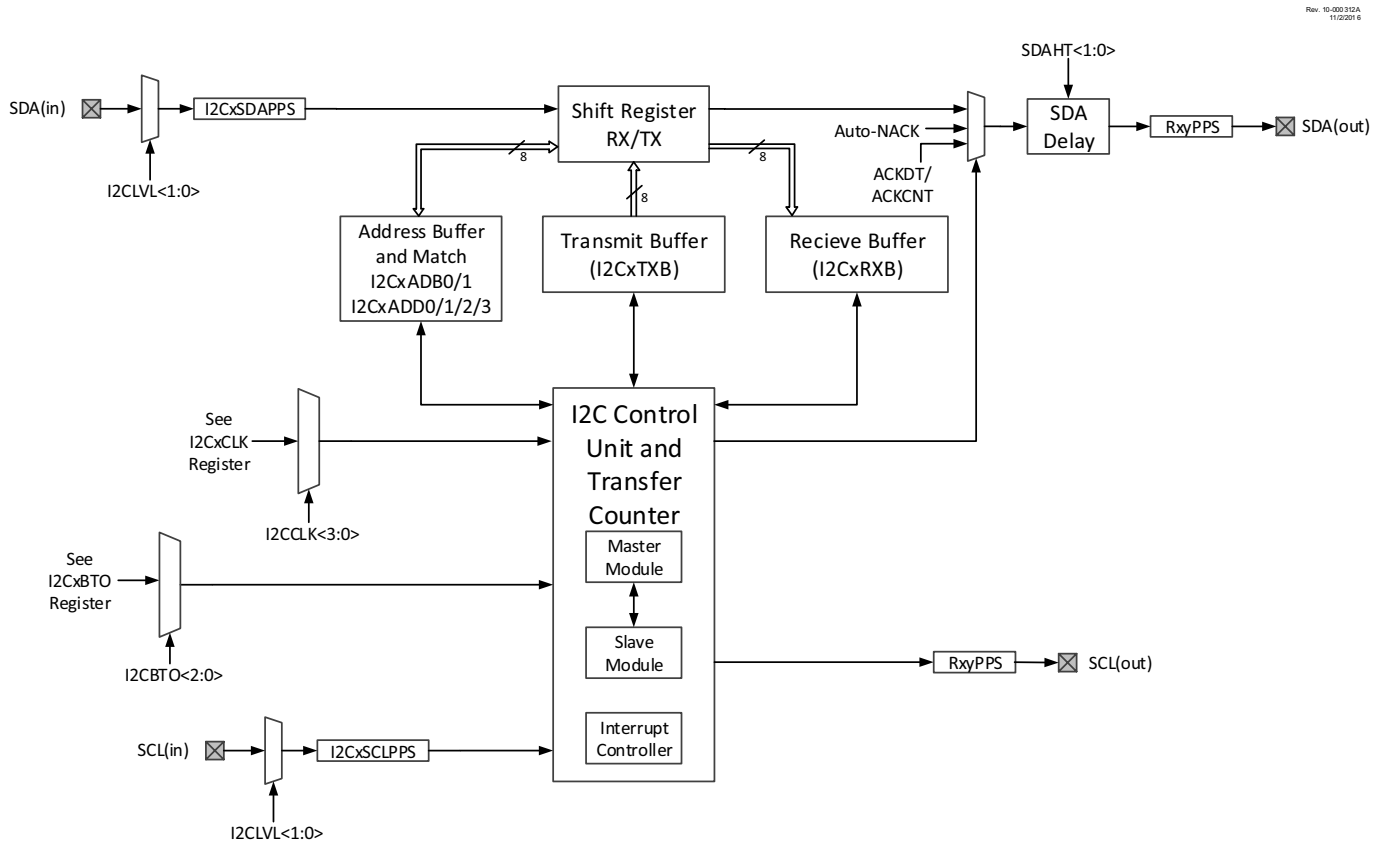
The SPI bus can be connected in a daisy-chain configuration. The first slave output is connected to the second slave input, the second slave output is connected to the third slave input, and so on. The final slave output is connected to the master input. Each slave sends out, during a second group of clock pulses, an exact copy of what was received during the first group of clock pulses. The whole chain acts as one large communication shift register. The daisy-chain feature only requires a single Slave Select line from the master device connected to all slave devices (alternately, the slave devices can be configured to ignore the slave select line by setting the SSET bit). In a typical Daisy-Chain configuration, the SCK signal from the master is connected to each of the slave device SCK inputs. However, the SCK input and output are separate signals selected by the PPS control. When the PPS selection is made to configure the SCK input and SCK output on separate pins then, the SCK output will follow the SCK input, allowing for SCK signals to be daisy-chained like the SDO/SDI signals.

Figure 32-12 shows the block diagram of a typical daisy-chain connection, and Figure 32-13 shows the block diagram of a daisy-chain connection possible using this SPI module.

33.0 I²C MODULE

The device has two dedicated, independent I²C modules. Figure 33-1 is a block diagram of the I²C interface module. The figure shows both the Master and Slave modes together.

FIGURE 33-1: I²C MODULE BLOCK DIAGRAM



There are four main operations based on the direction of the data being shared during I²C communication.

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

To begin any I²C communication, 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 ACK. The master then continues to shift data in or out of the slave until it terminates the message with a Stop.

Further details about the I²C module are discussed in the section below.

33.3 I²C Mode Operation

All I²C communication is 8-bit data and 1-bit acknowledge and shifted out MSb first. The user can control the interaction between the software and the module using several control registers and interrupt flags. Two pins, SDA and SCL, are exercised by the module to communicate with other external I²C devices.

33.3.1 DEFINITION OF I²C TERMINOLOGY

The I²C communication protocol terminologies are defined for reference below in Table 33-1. These terminologies are used throughout this document. Table 33-1 has been adapted from the Phillips I²C specification.

TABLE 33-1: I²C BUS TERMS

TERM	Description
Transmitter	The device which shifts data out onto the bus
Receiver	The device which shifts data in from the bus
Master	The device that initiates a transfer, generates clock signals and terminates a transfer
Slave	The device addressed by the master
Multi-master	A bus with more than one device that can initiate data transfers
Arbitration	Procedure to ensure that only one master at a time controls the bus. Winning arbitration ensures that the message is not corrupted
Synchronization	Procedure to synchronize the clocks of two or more devices on the bus.
Idle	No master is controlling the bus, and both SDA and SCL lines are high
Active	Any time one or more master devices are controlling the bus
Addressed Slave	Slave device that has received a matching address and is actively being clocked by a master
Matching Address	Address byte that is clocked into a slave that matches the value stored in I2CxADR
Write Request	Slave receives a matching address with R/W bit clear and is ready to clock in data
Read Request	Master sends an address byte with the R/W bit set, indicating that it wishes to clock data out of the Slave. This data is the next and all following bytes until a Restart or Stop.
Clock Stretching	When a device on the bus holds SCL low to stall communication
Bus Collision	Any time the SDA line is sampled low by the module while it is outputting and expected high state.
Bus Timeout	Any time the I2CBTOISM input transitions high, the I²C module is reset and the module goes idle.

33.4.3 SLAVE OPERATION IN 7-BIT ADDRESSING MODE

The 8th bit in an address byte transmitted by the master is used to determine if the Master wants to read from or write to the Slave device. If set, it denotes that the Master wants to read from the slave and if cleared it means the master wants to write to the slave device. If there is an address match, the R/W bit is copied to the R/W bit of the I2CxSTAT0 register.

33.4.3.1 Slave Reception (7-bit Addressing Mode)

This section describes the sequence of events for the I^2C module configured as an I^2C slave in 7-bit Addressing mode and is receiving data. Figure 33-6, Figure 33-7, and Figure 33-8 are used as a visual reference for this description.

1. Master asserts Start condition (can also be a restart) on the bus. Start condition Interrupt Flag (SCIF) in I2CxPIR register is set.
2. If Start condition interrupt is enabled (SCIE bit is set), generic interrupt I2CxIF is set.
3. Master transmits eight bits – 7-bit address and R/W = 0.
4. Received address is compared with the values in I2CxADR0/I2CxADR1/I2CxADR2/I2CxADR3 registers. Refer to section **Section 33.4.1 "Slave Addressing Modes"** for slave addressing modes.
5. If address matches; SMA in I2CxSTAT0 register is set, R/W is copied to R/W bit, D/A bit is cleared. If the address does not match; module becomes idle.
6. The matched address data is loaded into I2CxADB0 and ADRIF in I2CxPIR register is set.
7. If Address hold interrupt is enabled (ADRIE = 1), CSTR is set. I2CxIF is set. Slave software can read address from I2CxADB0 and set/clear ACKDT before releasing SCL.
8. If there are any previous error conditions, e.g., Receive buffer overflow or transmit buffer underflow errors, Slave will force a NACK and the module becomes idle.
9. ACKDT value is copied out to SDA for ACK pulse to be read by the Master on the 9th SCL pulse.
10. If the Acknowledge interrupt and hold is enabled (ACKTIE = 1), CSTR is set, I2CxIF is set, then Slave software can read address from I2CxADB0 register and change the value of ACKDT before releasing SCL by clearing CSTR.
11. Master sends first seven SCL pulses of the data byte or a Stop condition (in the case of NACK).
12. If Stop condition; PCIF in I2CxPIR register is set, module becomes idle.
13. If the receive buffer is full from the previous transaction i.e. RXBF = 1 (I2CxRXIF = 1), CSTR is set. Slave software must read data out of I2CxRXB to resume communication.
14. Master sends 8th SCL pulse of the data byte. D/A bit is set, WRIF is set.
15. I2CxRXB is loaded with new data, RXBF bit is set, I2CxRXIF is set.
16. If Data write interrupt and hold is enabled (WRIE = 1), CSTR is set, I2CxIF is set. Slave software can read data from I2CxRXB and set/clear ACKDT before releasing SCL by clearing CSTR.
17. If I2CxCNT = 0, the ACKCNT value is output to the SDA; else, if I2CxCNT!= 0, the ACKDT value is used and the value of I2CxCNT is decremented.
18. The ACK value is copied out to SDA to be read by the Master on the 9th SCL pulse.
19. If I2CxCNT = 0, CNTIF is set.
20. If a NACK was sent, NACKIF is set, module becomes idle.
21. If ACKTIE = 1, CSTR is set, I2CxIF is set. Slave software can read data from I2CxRXB clearing RXBF, before releasing SCL by clearing CSTR.
22. Go to step 11.

33.5.12 MASTER RECEPTION IN 10-BIT ADDRESSING MODE

This section describes the sequence of events for the I^2C module configured as an I^2C master in 10-bit Addressing mode and is receiving data. Figure 33-22 is used as a visual reference for this description.

1. Master software loads high address byte in I2CxADB1 and low address byte in I2CxADB0 for write and sets restart enable (RSTEN) bit.
2. Master software sets START bit.
3. Master hardware waits for BFRE bit to be set; then shifts out start, high address and waits for acknowledge.
4. If slave responds with a NACK, master hardware sends Stop and ends communication.
5. If slave responds with ACK, master hardware shifts out the low address.
6. If the transmit buffer empty flag (TXBE) is set and I2CxCNT! = 0, the clock is stretched on 8th falling SCL edge. Allowing master software writes next data to I2CxTXB.
7. Master hardware sends 9th SCL pulse for ACK from slave and loads the shift register from I2CxTXB.
8. If slave responds with a NACK, master hardware sends Stop and ends communication.
9. If slave responds with an ACK and I2CxCNT = 0, master hardware sets MDR bit, go to Step 11.
10. If slave responds with an ACK and I2CxCNT! = 0, master hardware outputs data in shift register on SDA and waits for ACK from slave. Go to step 4.
11. Master software loads I2CxADB0 for read, and I2CCNT with the number of bytes to be received in the current transaction.
12. Master software sets Start bit.
13. Master hardware shifts out Restart and high address with R/W = 1.
14. Master sends out 9th SCL pulse for ACK from Slave.
15. If slave responds with a NACK, master hardware sends Stop or sets MDR (RSEN bit).
16. If slave responds with an ACK, master hardware shifts 7 bits of data into the shift register from the slave.
17. If the receive buffer full flag (RXBF) is set, clock is stretched on seventh falling SCL edge.
18. Master software can clear clock stretching by reading the previous data in the receive buffer.
19. Master hardware shifts 8th bit of data into the shift register from slave and loads it into I2CxRXB.
20. Master software reads data from I2CxRXB register.
21. If I2CxCNT! = 0, master hardware clocks out ACKDT as ACK value to slave.
22. If I2CxCNT = 0, master hardware clocks out ACKCNT as ACK value to slave
23. Go to step 4.

REGISTER 33-14: I2CxADR2: I²C ADDRESS 2 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
ADR7	ADR6	ADR5	ADR4	ADR3	ADR2	ADR1	ADR0
bit 7							bit 0

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

bit 7-0 **ADR<7-0>:** Address 2 bits

MODE<2:0> = 000 | 110 - 7-bit Slave/Multi-Master Modes
ADR<7:1>:7-bit Slave Address

MODE<2:0> = 001 | 111 - 7-bit Slave/Multi-Master Modes with Masking
ADR<7:1>:7-bit Slave Address

MODE<2:0> = 010 - 10-Bit Slave Mode
ADR<7:0>: Eight Least Significant bits of second 10-bit address

MODE<2:0> = 011 - 10-Bit Slave Mode with Masking
MSK0<7-0>: The received address byte is masked, then compared to I2CxADR0

34.0 FIXED VOLTAGE REFERENCE (FVR)

The Fixed Voltage Reference, or FVR, is a stable voltage reference, independent of VDD, with 1.024V, 2.048V or 4.096V selectable output levels. The output of the FVR can be configured to supply a reference voltage to the following:

- ADC input channel
- ADC positive reference
- Comparator input
- Digital-to-Analog Converter (DAC)

The FVR can be enabled by setting the EN bit of the FVRCON register.

Note:	Fixed Voltage Reference output cannot exceed VDD.

34.1 Independent Gain Amplifiers

The output of the FVR, which is connected to the ADC, Comparators, and DAC, is routed through two independent programmable gain amplifiers. Each amplifier can be programmed for a gain of 1x, 2x or 4x, to produce the three possible voltage levels.

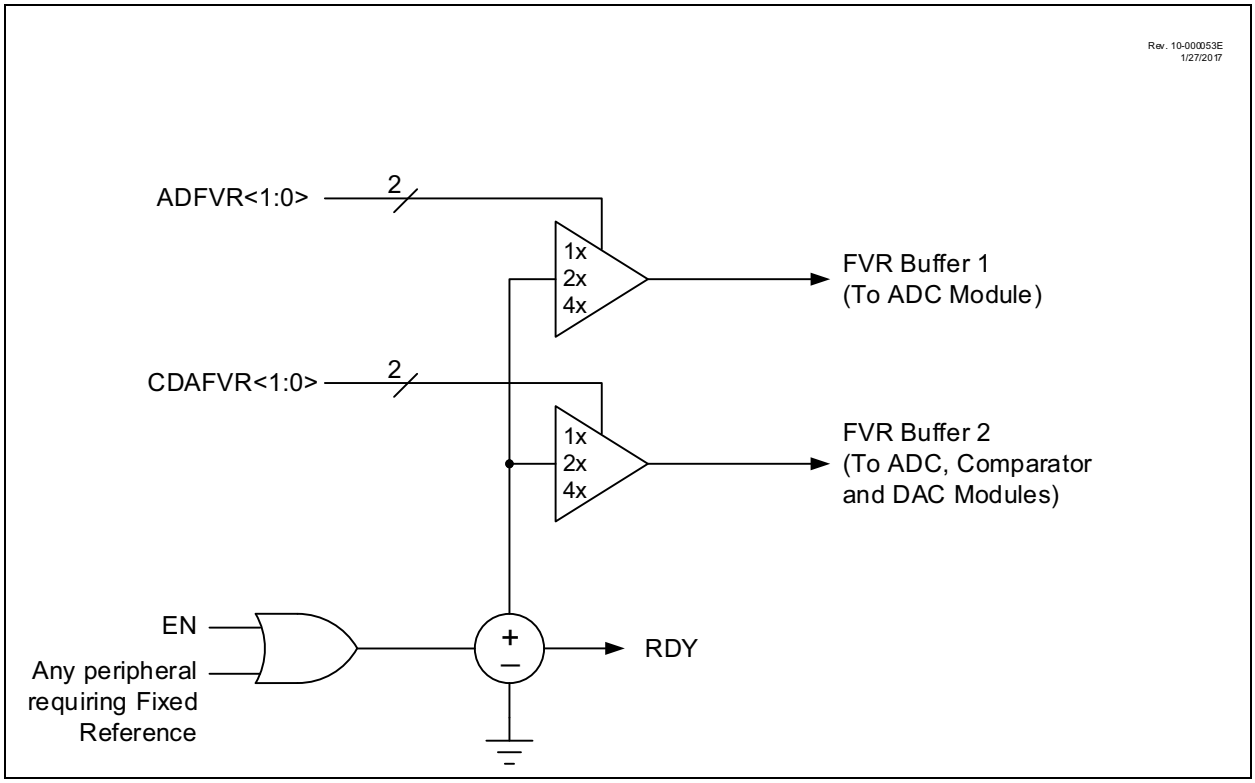
The ADFVR<1:0> bits of the FVRCON register are used to enable and configure the gain amplifier settings for the reference supplied to the ADC module. Reference **Section 36.0 "Analog-to-Digital Converter with Computation (ADC2) Module"** for additional information.

The CDAFVR<1:0> bits of the FVRCON register are used to enable and configure the gain amplifier settings for the reference supplied to the DAC and comparator module. Reference **Section 37.0 "5-Bit Digital-to-Analog Converter (DAC) Module"** and **Section 38.0 "Comparator Module"** for additional information.

34.2 FVR Stabilization Period

When the Fixed Voltage Reference module is enabled, it requires time for the reference and amplifier circuits to stabilize. Once the circuits stabilize and are ready for use, the RDY bit of the FVRCON register will be set.

FIGURE 34-1: VOLTAGE REFERENCE BLOCK DIAGRAM



REGISTER 36-11: ADACQL: ADC ACQUISITION TIME CONTROL REGISTER (LOW BYTE)

R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0
ACQ<7:0>							
bit 7							bit 0

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

bit 7-0 ACQ<7:0>: Acquisition (charge share time) Select bits
See Table 36-5.

REGISTER 36-12: ADACQH: ADC ACQUISITION TIME CONTROL REGISTER (HIGH BYTE)

U-0	U-0	U-0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0
—	—	—	ACQ<12:8>				
bit 7							bit 0

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

bit 7-5 **Unimplemented**: Read as ‘0’

bit 4-0 ACQ<12:8>: Acquisition (charge share time) Select bits
See Table 36-5.

TABLE 36-5: ACQUISITION TIME

ADACQ	Acquisition time
1 1111 1111 1111	8191 clocks of the selected ADC clock
1 1111 1111 1110	8190 clocks of the selected ADC clock
1 1111 1111 1101	8189 clocks of the selected ADC clock
...	...
0 0000 0000 0010	2 clocks of the selected ADC clock
0 0000 0000 0001	1 clock of the selected ADC clock
0 0000 0000 0000	Not included in the data conversion cycle(1)

Note 1: If ADPRE is not equal to ‘0’, then ADACQ = 0b0_0000_0000_0000 means Acquisition time is 8192 clocks of the selected ADC clock.

TABLE 42-1: REGISTER FILE SUMMARY FOR PIC18(L)F26/27/45/46/47/55/56/57K42 DEVICES

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Register on page
3A16h	RC6PPS	—	—	—	RC6PPS4	RC6PPS3	RC6PPS2	RC6PPS1	RC6PPS0	280
3A15h	RC5PPS	—	—	—	RC5PPS4	RC5PPS3	RC5PPS2	RC5PPS1	RC5PPS0	280
3A14h	RC4PPS	—	—	—	RC4PPS4	RC4PPS3	RC4PPS2	RC4PPS1	RC4PPS0	280
3A13h	RC3PPS	—	—	—	RC3PPS4	RC3PPS3	RC3PPS2	RC3PPS1	RC3PPS0	280
3A12h	RC2PPS	—	—	—	RC2PPS4	RC2PPS3	RC2PPS2	RC2PPS1	RC2PPS0	280
3A11h	RC1PPS	—	—	—	RC1PPS4	RC1PPS3	RC1PPS2	RC1PPS1	RC1PPS0	280
3A10h	RC0PPS	—	—	—	RC0PPS4	RC0PPS3	RC0PPS2	RC0PPS1	RC0PPS0	280
3A0Fh	RB7PPS	—	—	—	RB7PPS4	RB7PPS3	RB7PPS2	RB7PPS1	RB7PPS0	280
3A0Eh	RB6PPS	—	—	—	RB6PPS4	RB6PPS3	RB6PPS2	RB6PPS1	RB6PPS0	280
3A0Dh	RB5PPS	—	—	—	RB5PPS4	RB5PPS3	RB5PPS2	RB5PPS1	RB5PPS0	280
3A0Ch	RB4PPS	—	—	—	RB4PPS4	RB4PPS3	RB4PPS2	RB4PPS1	RB4PPS0	280
3A0Bh	RB3PPS	—	—	—	RB3PPS4	RB3PPS3	RB3PPS2	RB3PPS1	RB3PPS0	280
3A0Ah	RB2PPS	—	—	—	RB2PPS4	RB2PPS3	RB2PPS2	RB2PPS1	RB2PPS0	280
3A09h	RB1PPS	—	—	—	RB1PPS4	RB1PPS3	RB1PPS2	RB1PPS1	RB1PPS0	280
3A08h	RB0PPS	—	—	—	RB0PPS4	RB0PPS3	RB0PPS2	RB0PPS1	RB0PPS0	280
3A07h	RA7PPS	—	—	—	RA7PPS4	RA7PPS3	RA7PPS2	RA7PPS1	RA7PPS0	280
3A06h	RA6PPS	—	—	—	RA6PPS4	RA6PPS3	RA6PPS2	RA6PPS1	RA6PPS0	280
3A05h	RA5PPS	—	—	—	RA5PPS4	RA5PPS3	RA5PPS2	RA5PPS1	RA5PPS0	280
3A04h	RA4PPS	—	—	—	RA4PPS4	RA4PPS3	RA4PPS2	RA4PPS1	RA4PPS0	280
3A03h	RA3PPS	—	—	—	RA3PPS4	RA3PPS3	RA3PPS2	RA3PPS1	RA3PPS0	280
3A02h	RA2PPS	—	—	—	RA2PPS4	RA2PPS3	RA2PPS2	RA2PPS1	RA2PPS0	280
3A01h	RA1PPS	—	—	—	RA1PPS4	RA1PPS3	RA1PPS2	RA1PPS1	RA1PPS0	280
3A00h	RA0PPS	—	—	—	RA0PPS4	RA0PPS3	RA0PPS2	RA0PPS1	RA0PPS0	280
39FFh - 39F8h	—	Unimplemented								
39F7h	SCANPR	—	—	—	—	—	PR			31
39F6h - 39F5h	—	Unimplemented								
39F4h	DMA2PR	—	—	—	—	—	PR			31
39F3h	DMA1PR	—	—	—	—	—	PR			30
39F2h	MAINPR	—	—	—	—	—	PR			30
39F1h	ISRPR	—	—	—	—	—	PR			30
39F0h	—	Unimplemented								
39EFh	PRLOCK	—	—	—	—	—	—	—	PRLOCKED	31
39EEh - 39E7h	—	Unimplemented								
39E6h	NVMCON2	NVMCON2								211
39E5h	NVMCON1	REG		—	FREE	WRERR	WREN	WR	RD	210
39E4h	—	Unimplemented								
39E3h	NVMDAT	DAT								212
39E2h	—	Unimplemented								
39E1h	NVMADRH(4)	—	—	—	—	—	—	ADR		211
39E0h	NVMADRL	ADR								211
39DFh	OSCFRQ	—	—	—	—	FRQ				107
39DEh	OSCTUNE	—	—	TUN						108
39DDh	OSCEN	EXTOEN	HFOEN	MFOEN	LFOEN	SOSCEN	ADOEN	—	—	109
39DCh	OSCSTAT	EXTOR	HFOR	MFOR	LFOR	SOR	ADOR	—	PLLR	106
39DBh	OSCCON3	CSWHOLD	SOSCPWR	—	ORDY	NOSCR	—	—	—	105

Legend: x = unknown, u = unchanged, — = unimplemented, q = value depends on condition

Note 1: Unimplemented in LF devices.

2: Unimplemented in PIC18(L)F26/27K42.

3: Unimplemented on PIC18(L)F26/27/45/46/47K42 devices.

4: Unimplemented in PIC18(L)F45/55K42.

TABLE 42-1: REGISTER FILE SUMMARY FOR PIC18(L)F26/27/45/46/47/55/56/57K42 DEVICES

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Register on page
388Dh	FSR2L_SHAD	FSR2L								125
388Ch	FSR1H_SHAD	—	—	FSR1H						125
388Bh	FSR1L_SHAD	FSR1L								125
388Ah	FSR0H_SHAD	—	—	FSR0H						125
3889h	FSR0L_SHAD	FSR0L								125
3888h	PCLATU_SHAD	—	—	—	PCU					125
3887h	PCLATH_SHAD	PCH								125
3886h	BSR_SHAD	—	—	BSR						125
3885h	WREG_SHAD	WREG								125
3884h	STATUS_SHAD	—	$\overline{TO}$	$\overline{PD}$	N	OV	Z	DC	C	125
3883h	SHADCON	—	—	—	—	—	—	—	SHADLO	168
3882h	BSR_CSHAD	—	—	BSR						57
3881h	WREG_CSHAD	WREG								57
3880h	STATUS_C-SHAD	—	$\overline{TO}$	$\overline{PD}$	N	OV	Z	DC	C	57
387Fh - 3800h	—	Unimplementeds								

Legend: x = unknown, u = unchanged, — = unimplemented, q = value depends on condition

Note 1: Unimplemented in LF devices.

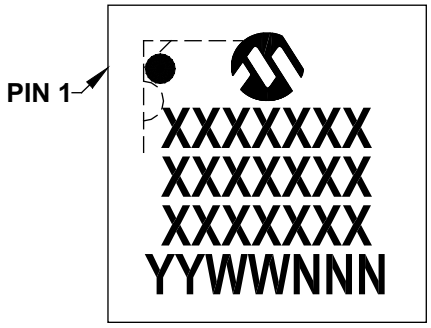
2: Unimplemented in PIC18(L)F26/27K42.

3: Unimplemented on PIC18(L)F26/27/45/46/47K42 devices.

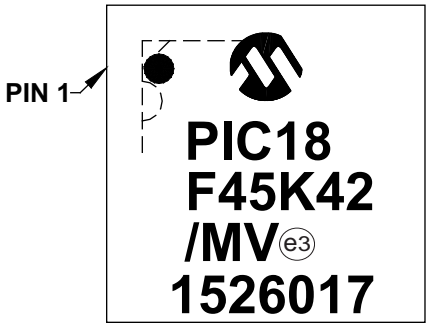
4: Unimplemented in PIC18(L)F45/55K42.

Package Marking Information (Continued)

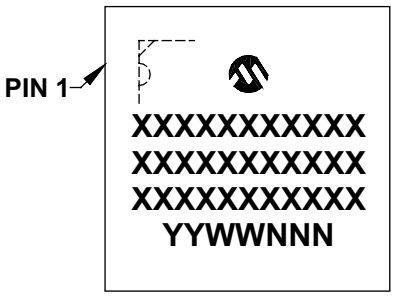
40-Lead UQFN (5x5x0.5 mm)



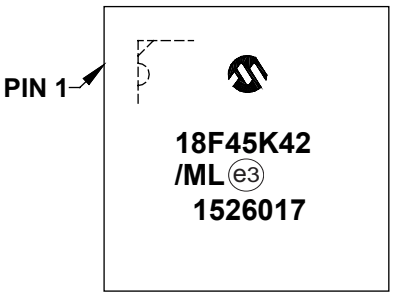
Example



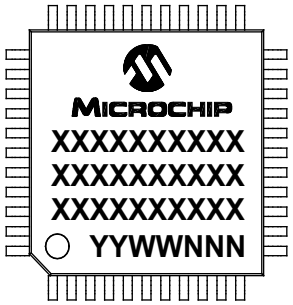
44-Lead QFN (8x8x0.9 mm)



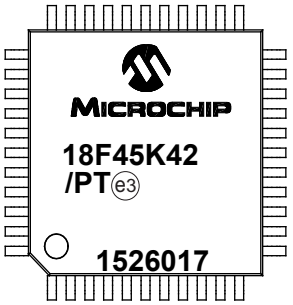
Example



44-Lead TQFP (10x10x1 mm)



Example



Legend:	XX...X	Customer-specific information or Microchip part number
	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.

PRODUCT IDENTIFICATION SYSTEM

To order or obtain information, e.g., on pricing or delivery, refer to the factory or the listed sales office.

PART NO.	[X](2)	-	X	/XX	XXX
Device	Tape and Reel Option		Temperature Range	Package	Pattern

Device:	PIC18F26K42, PIC18LF26K42, PIC18F27K42, PIC18LF27K42 PIC18F45K42, PIC18LF45K42 PIC18F46K42, PIC18LF46K42 PIC18F47K42, PIC18LF47K42 PIC18F55K42, PIC18LF55K42 PIC18F56K42, PIC18LF56K42 PIC18F57K42, PIC18LF57K42
Tape and Reel Option:	Blank = standard packaging (tube or tray) T = Tape and Reel(1), (2)
Temperature Range:	E = -40°C to +125°C (Extended) I = -40°C to +85°C (Industrial)
Package:	ML = 28-lead QFN 6x6mm ML = 44-lead QFN 8x8x0.9mm MX = 28-lead UQFN 6x6x0.5mm MV = 40-lead UQFN 5x5x0.5mm MV = 48-lead UQFN P = 40-lead PDIP PT = 44-lead TQFP (Thin Quad Flatpack) PT = 48-lead TQFP SO = 28-lead SOIC SP = 28-lead Skinny Plastic DIP SS = 28-lead SSOP
Pattern:	QTP, SQTP, Code or Special Requirements (blank otherwise)

Examples:

a) PIC18F26K42-E/P 301 = Extended temp., PDIP package, QTP pattern #301.

b) PIC18F45K42-E/SO = Extended temp., SOIC package.

c) PIC18F46K42T-I/ML = Tape and reel, Industrial temp., QFN package.

Note 1: Tape and Reel option is available for ML, MV, PT, SO and SS packages with industrial Temperature Range only.

2: Tape and Reel identifier only appears in catalog part number description. This identifier is used for ordering purposes and is not printed on the device package.