



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 "<u>Embedded -</u> <u>Microcontrollers</u>"

Details

Product Status	Active
Core Processor	PIC
Core Size	8-Bit
Speed	64MHz
Connectivity	I ² C, LINbus, SPI, UART/USART
Peripherals	Brown-out Detect/Reset, DMA, HLVD, POR, PWM, WDT
Number of I/O	44
Program Memory Size	128KB (64K x 16)
Program Memory Type	FLASH
EEPROM Size	1K x 8
RAM Size	8K x 8
Voltage - Supply (Vcc/Vdd)	2.3V ~ 5.5V
Data Converters	A/D 43x12b; D/A 1x5b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 85°C (TA)
Mounting Type	Surface Mount
Package / Case	48-UFQFN Exposed Pad
Supplier Device Package	48-UQFN (6x6)
Purchase URL	https://www.e-xfl.com/product-detail/microchip-technology/pic18f57k42t-i-mv

Email: info@E-XFL.COM

Address: Room A, 16/F, Full Win Commercial Centre, 573 Nathan Road, Mongkok, Hong Kong

40-nin PDIP		_
40-pin i Dii		0 RB7/ICSPDAT
	RA0 2 3	
	RA1 🛛 3 3	8 RB5
	RA2 4 3	7 RB4
	RA3 5 3	6 RB3
		5 RB2
		4 RB1 2 RB0
	RE2∏10 - 3	 1
	Vss 12 ā 2	9 RD6
	RA7 13 2	8 RD5
	RA6 14 2	7 RD4
	RC0 15 2	6 RC7
	RC1 16 2	5 RC6
		4 RC5
	$\frac{RC3}{RD0} \frac{18}{10} \qquad 2$	2 RC4 2 RD3
	$RD1 \square 20 2$	1 RD2
Note: See Table 2 for loc	ation of all peripheral functions.	
40-pin UQFN (5x5x0.5mm)		
	33 0 7 5 3 7 2 8	8.5
	RC R	2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2
	RC7 40 39 38 37 36 35 34 33 3	32 31
	RD4 2	³⁰ I RC0
	RD5 3	²⁹ RA6
	RD6 4 RD7 5	28 RA7 27 Vss
	Vss 6 PIC18(L)F4XK42	261 VDD
		²⁵ RE2
	RB0 8	24 RE1 23 RE0
	RB2 10	22 RA5
		²¹ RA4
	0 7 9 9 9 9 9 7 10 19 2 0 7 9 9 9 9 9 10 17 10 19 2	<u>0</u>
	RA RB RA RB RA RA RA RB RA RB RA RB RA RB RA RB RA RB RA RB RB RB RB RB RB RB RB RB RB RB RB RB R	A Y
	a SS PVM	
	9 9 4 ×	
Note 1: See Table 2 for location	on of all peripheral functions.	
2: It is recommended that	at the exposed bottom pad be connec	ted to Vss, however it must not be the only
Vss connection to the	device.	

4.3.2 LOOK-UP TABLES IN PROGRAM MEMORY

There may be programming situations that require the creation of data structures, or look-up tables, in program memory. For PIC18 devices, look-up tables can be implemented in two ways:

- Computed GOTO
- Table Reads

4.3.2.1 Computed GOTO

A computed GOTO is accomplished by adding an offset to the program counter. An example is shown in Example 4-2.

A look-up table can be formed with an ADDWF PCL instruction and a group of RETLW nn instructions. The W register is loaded with an offset into the table before executing a call to that table. The first instruction of the called routine is the ADDWF PCL instruction. The next instruction executed will be one of the RETLW nn instructions that returns the value 'nn' to the calling function.

The offset value (in WREG) specifies the number of bytes that the program counter should advance and should be multiples of two (LSb = 0).

In this method, only one data byte may be stored in each instruction location and room on the return address stack is required.

EXAMPLE 4-2: COMPUTED GOTO USING AN OFFSET VALUE

	MOVF	OFFSET,	W
	CALL	TABLE	
ORG	nn00h		
TABLE	ADDWF	PCL	
	RETLW	nnh	
	RETLW	nnh	
	RETLW	nnh	
	•		
	•		
	•		

4.3.2.2 Table Reads and Table Writes

A better method of storing data in program memory allows two bytes of data to be stored in each instruction location.

Look-up table data may be stored two bytes per program word by using table reads and writes. The Table Pointer (TBLPTR) register specifies the byte address and the Table Latch (TABLAT) register contains the data that is read from or written to program memory.

Table read and table write operations are discussed further in Section 13.1.1 "Table Reads and Table Writes".

© 2017 Microchip Technology Inc.

9.5 Context Saving

The Interrupt controller supports a two-level deep context saving (Main routine context and Low ISR context). Refer to state machine shown in Figure 9-6 for details.

The Program Counter (PC) is saved on the dedicated device PC stack. CPU registers saved include STATUS, WREG, BSR, FSR0/1/2, PRODL/H and PCLATH/U.

After WREG has been saved to the context registers, the resolved vector number of the interrupt source to be serviced is copied into WREG. Context save and restore operation is completed by the interrupt controller based on current state of the interrupts and the order in which they were sent to the CPU.

Context save/restore works the same way in both states of MVECEN. When IPEN = 0, there is only one level interrupt active. Hence, only the main context is saved when an interrupt is received.

9.5.1 ACCESSING SHADOW REGISTERS

The Interrupt controller automatically saves the context information in the shadow registers available in Bank 56. Both the saved context values (i.e., main routine and low ISR) can be accessed using the same set of shadow registers. By clearing the SHADLO bit in the SHADCON register (Register 9-43), the CPU register values saved for main routine context can accessed, and by setting the SHADLO bit of the CPU register, values saved for low ISR context can accessed. Low ISR context is automatically restored to the CPU registers upon exiting the high ISR. Similarly, the main context is automatically restored to the CPU registers upon exiting the low ISR.

The Shadow registers in Bank 56 are readable and writable, so if the user desires to modify the context, then the corresponding shadow register should be modified and the value will be restored when exiting the ISR. Depending on the user's application, other registers may also need to be saved.



EXAMPLE 9-4: SETTING UP VECTORED INTERRUPTS USING XC8

```
// NOTE 1: If IVTBASE is changed from its default value of 0x000008, then the
// "base(...)" argument must be provided in the ISR. Otherwise the vector
// table will be placed at 0x0008 by default regardless of the IVTBASE value.
// NOTE 2: When MVECEN=0 and IPEN=1, a separate argument as "high priority"
// or "low priority" can be used to distinguish between the two ISRs.
// If the argument is not provided, the ISR is considered high priority
// by default.
// NOTE 3: Multiple interrupts can be handled by the same ISR if they are
// specified in the "irq(...)" argument. Ex: irq(IRQ TMR0, IRQ CCP1)
void interrupt(irq(IRQ TMR0), base(0x4008)) TMR0 ISR(void)
{
       PIR3bits.TMR0IF = 0;
                                             // Clear the interrupt flag
       LATCbits.LC0 ^= 1;
                                             // ISR code goes here
}
void interrupt(irq(default), base(0x4008)) DEFAULT ISR(void)
{
       // Unhandled interrupts go here
}
void INTERRUPT Initialize (void)
{
                                            // Enable high priority interrupts
       INTCONObits.GIEH = 1;
                                             // Enable low priority interrupts
       INTCONObits.GIEL = 1;
       INTCONObits.IPEN = 1;
                                             // Enable interrupt priority
       PIE3bits.TMR0IE = 1;
                                            // Enable TMR0 interrupt
       PIE4bits.TMR1IE = 1;
                                             // Enable TMR1 interrupt
       IPR3bits.TMR0IP = 0;
                                             // Make TMR0 interrupt low priority
       // Change IVTBASE if required
       IVTBASEU = 0 \times 00;
                                             // Optional
       IVTBASEH = 0 \times 40;
                                             // Default is 0x0008
       IVTBASEL = 0 \times 08;
}
```

14.2 CRC Functional Overview

The CRC module can be used to detect bit errors in the program memory using the built-in memory scanner or through user input RAM memory. The CRC module can accept up to a 16-bit polynomial with up to a 16-bit seed value. A CRC calculated check value (or checksum) will then be generated into the CRCACC<15:0> registers for user storage. The CRC module uses an XOR shift register implementation to perform the polynomial division required for the CRC calculation.

EXAMPLE 14-1: CRC EXAMPLE



REGISTER 15-12: DMAxSCNTL: DMAx SOURCE COUNT LOW REGISTER

R-0	R-0	R-0	R-0	R-0	R-0	R-0	R-0
			SCN	T<7:0>			
bit 7							bit 0

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

bit 7-0 SCNT<7:0>: Current Source Byte Count

REGISTER 15-13: DMAxSCNTH: DMAx SOURCE COUNT HIGH REGISTER

U-0	U-0	U-0	U-0	R-0	R-0	R-0	R-0
—	—	—	—		SCNT<	11:8>	
bit 7				bit 0			

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

bit 7-4 Unimplemented: Read as '0'

bit 3-0 SCNT<11:8>: Current Source Byte Count

REGISTER 15-14: DMAxDSAL: DMAx DESTINATION START ADDRESS LOW REGISTER

| R/W-0/0 |
|---------|---------|---------|---------|---------|---------|---------|---------|
| | | | DSA | \<7:0> | | | |
| bit 7 | | | | | | | bit 0 |
| | | | | | | | |

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

bit 7-0 DSA<7:0>: Destination Start Address bits

© 2017 Microchip Technology Inc.

21.12 Register Definitions: Timer1/3/5

Long bit name prefixes for the Timer1/3/5 are shown below. Refer to **Section 1.3.2.2 "Long Bit Names"** for more information.

Peripheral	Bit Name Prefix
Timer1	T1
Timer3	Т3
Timer5	T5

REGISTER 21-1: TXCON: TIMERx CONTROL REGISTER

U-0	U-0	R/W-0/u	R/W-0/u	U-0	R/W-0/u	R/W-0/0	R/W-0/u
—	—	CKPS	<1:0>	—	SYNC	RD16	ON
bit 7							bit 0

Legend:				
R = Readable bit	Readable bit W = Writable bit U = Unimplemented bit, read as '0'			
-n = Value at POR	'1' = Bit is set	'0' = Bit is cleared	u = unchanged	

bit 7-6	Unimplemented: Read as '0'
bit 5-4	CKPS<1:0>: Timerx 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	Unimplemented: Read as '0'
bit 2	SYNC: Timerx External Clock Input Synchronization Control bit TMRxCLK = Fosc/4 or Fosc: This bit is ignored. Timer1 uses the incoming clock as is. Else: 1 = Do not synchronize external clock input 0 = Synchronize external clock input with system clock
bit 1	 RD16: 16-Bit Read/Write Mode Enable bit 1 = Enables register read/write of Timerx in one 16-bit operation 0 = Enables register read/write of Timerx in two 8-bit operation
bit 0	ON: Timerx On bit 1 = Enables Timerx 0 = Disables Timerx





FIGURE 25-12: GATED WINDOWED MEASURE MODE REPEAT ACQUISITION TIMING DIAGRAM

31.13 Checksum (UART1 only)

This section does not apply to the LIN mode, which handles checksums automatically.

The transmit and receive checksum adders are enabled when the C0EN bit in the UxCON2 register is set. When enabled, the adders accumulate every byte that is transmitted or received. The accumulated sum includes the carry of the addition. Software is responsible for clearing the checksum registers before a transaction and performing the check at the end of the transaction.

The following is an example of how the checksum registers could be used in the asynchronous modes.

31.13.1 TRANSMIT CHECKSUM METHOD

- 1. Clear the UxTXCHK register.
- 2. Set the COEN bit.
- 3. Send all bytes of the transaction output.
- 4. Invert UxTXCHK and send the result as the last byte of the transaction.

31.13.2 RECEIVE CHECKSUM METHOD

- 1. Clear the UxRXCHK register.
- 2. Set the COEN bit.
- 3. Receive all bytes in the transaction including the checksum byte.
- 4. Set MSb of UxRXCHK if 7-bit mode is selected.
- 5. Add 1 to UxRXCHK.
- 6. If the result is '0', the checksum passes, otherwise it fails.

The CERIF checksum interrupt flag is not active in any mode other than LIN.

31.14 Collision Detection

External forces that interfere with the transmit line are detected in all modes of operation with collision detection. Collision detection is always active when RXEN and TXEN are both set.

When the receive input is connected to the transmit output through either the same I/O pin or external circuitry, a character will be received for every character transmitted. The collision detection circuit provides a warning when the word received does not match the word transmitted. The TXCIF flag in the UxERRIR register is used to signal collisions. This signal is only useful when the TX output is looped back to the RX input and everything that is transmitted is expected to be received. If more than one transmitter is active at the same time, it can be assumed that the TX word will not match the RX word. The TXCIF detects this mismatch and flags an interrupt. The TXCIF bit will also be set in DALI mode transmissions when the received bit is missing the expected mid-bit transition.

Collision detection is always active, regardless of whether or not the RX input is connected to the TX output. It is up to the user to disable the TXCIE bit when collision interrupts are not required.

The software overhead of unloading the receive buffer of transmitted data is avoided by setting the RUNOVF bit in UxCON2 and ignoring the receive interrupt and letting the receive buffer overflow. When the transmission is complete, prepare for receiving data by flushing the receive buffer (see Section 31.11.2, FIFO Reset) and clearing the RXFOIF overflow flag in the UxERRIR register.

31.15 RX/TX Activity Timeout

The UART works in conjunction with the HLT timers to monitor activity on the RX and TX lines. Use this feature to determine when there has been no activity on the receive or transmit lines for a user specified period of time.

To use this feature, set the HLT to the desired timeout period by a combination of the HLT clock source, timer prescale value, and timer period registers. Configure the HLT to reset on the UART TX or RX line and start the HLT at the same time the UART is started. UART activity will keep resetting the HLT to prevent a full HLT period from elapsing. When there has been no activity on the selected TX or RX line for longer than the HLT period then an HLT interrupt will occur signaling the timeout event.

For example, the following register settings will configure HLT2 for a 5 ms timeout of no activity on U1RX:

- T2PR = 0x9C (156 prescale periods)
- T2CLKCON = 0x05 (500 kHz internal oscillator)
- T2HLT = 0x04 (free running, reset on rising edge)
- T2RST = 0x15 (reset on U1RX)
- T2CON = 0xC0 (Timer2 on with 1:16 prescale)

REGISTER 32-12: SPIxTxB: SPI TRANSMIT BUFFER REGISTER

W-0	W-0	W-0	W-0	W-0	W-0	W-0	W-0
TXB7	TXB6	TXB5	TXB4	TXB3	TXB2	TXB1	TXB0
bit 7							bit 0

Legend:		
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as '0'

bit 7-0 **TXB<7:0>**: Transmit Buffer bits (write only)

If TXFIFO is not full:

Writing to this register adds the data to the top of the TXFIFO and increases the occupancy of the TXFIFO write pointer $% \left(T_{\rm A}^{\rm A}\right) =0$

If TXFIFO is full:

Writing to this register does not affect the data in the TXFIFO or the write pointer, and the TXWE bit of SPIxSTATUS will be set

REGISTER 32-13: SPIxCLK: SPI CLOCK SELECTION REGISTER

U-0	U-0	U-0	U-0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0
—	—	—	—	CLKSEL3	CLKSEL2	CLKSEL1	CLKSEL0
bit 7							bit 0

Legend:		
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as '0'

bit 7-4 Unimplemented: Read as '0'

bit 3-0 CLKSEL<3:0>: SPI Clock Source Selection bits

1111-1001 = Reserved

- 1000 = SMT_match
- 0111 = TMR6_Postscaled
- 0110 = TMR4_Postscaled
- 0101 = TMR2_Postscaled
- 0100 = TMR0_overflow
- 0011 = CLKREF
- 0010 = MFINTOSC
- 0001 = HFINTOSC
- 0000 **= FOSC**



FIGURE 33-7: $I^{2}C$ SLAVE, 7-BIT ADDRESS, RECEPTION WITH I2CxCNT (ACKTIE = 1, ADRIE = 0, WRIE = 0)

36.5.2 PRECHARGE CONTROL

The precharge stage is an optional period of time that brings the external channel and internal sample and hold capacitor to known voltage levels. Precharge is enabled by writing a non-zero value to the ADPRE register. This stage is initiated when an ADC conversion begins, either from setting the GO bit, a special event trigger, or a conversion restart from the computation functionality. If the ADPRE register is cleared when an ADC conversion begins, this stage is skipped.

During the precharge time, CHOLD is disconnected from the outer portion of the sample path that leads to the external capacitive sensor and is connected to either VDD or VSS, depending on the value of the PPOL bit of ADCON1. At the same time, the port pin logic of the selected analog channel is overridden to drive a digital high or low out, in order to precharge the outer portion of the ADC's sample path, which includes the external sensor. The output polarity of this override is also determined by the PPOL bit of ADCON1. The amount of time that this charging receives is controlled by the ADPRE register.

- **Note 1:** The external charging overrides the TRIS setting of the respective I/O pin.
 - **2:** If there is a device attached to this pin, Precharge should not be used.

36.5.3 ACQUISITION CONTROL

The Acquisition stage is an optional time for the voltage on the internal sample and hold capacitor to charge or discharge from the selected analog channel. This acquisition time is controlled by the ADACQ register. If PRE = 0, acquisition starts at the beginning of conversion. When PRE = 1, the acquisition stage begins when precharge ends.

At the start of the acquisition stage, the port pin logic of the selected analog channel is overridden to turn off the digital high/low output drivers so they do not affect the final result of the charge averaging. Also, the selected ADC channel is connected to CHOLD. This allows charge averaging to proceed between the precharged channel and the CHOLD capacitor.

Note: When PRE! = 0, acquisition time cannot be '0'. In this case, setting ADACQ to '0' will set a maximum acquisition time (8191 ADC clock cycles). When precharge is disabled, setting ADACQ to '0' will disable hardware acquisition time control.

36.5.4 GUARD RING OUTPUTS

Figure 36-8 shows a typical guard ring circuit. CGUARD represents the capacitance of the guard ring trace placed on the PCB board. The user selects values for RA and RB that will create a voltage profile on CGUARD, which will match the selected acquisition channel.

The purpose of the guard ring is to generate a signal in phase with the CVD sensing signal to minimize the effects of the parasitic capacitance on sensing electrodes. It also can be used as a mutual drive for mutual capacitive sensing. For more information about active guard and mutual drive, see Application Note AN1478, "*mTouch*TM Sensing Solution Acquisition Methods Capacitive Voltage Divider" (DS01478).

The ADC has two guard ring drive outputs, ADGRDA and ADGRDB. These outputs can be routed through PPS controls to I/O pins (see Section **17.0 "Peripheral Pin Select (PPS) Module**" for details) and the polarity of these outputs are controlled by the ADGPOL and ADIPEN bits of ADCON1.

At the start of the first precharge stage, both outputs are set to match the ADGPOL bit of ADCON1. Once the acquisition stage begins, ADGRDA changes polarity, while ADGRDB remains unchanged. When performing a double sample conversion, setting the ADIPEN bit of ADCON1 causes both guard ring outputs to transition to the opposite polarity of ADGPOL at the start of the second precharge stage, and ADGRDA toggles again for the second acquisition. For more information on the timing of the guard ring output, refer to Figure 36-8 and Figure 36-9.





REGISTER 36-32: ADLTHL: ADC LOWER THRESHOLD LOW 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	
			LTH<	<7:0>				
bit 7							bit 0	
Legend:								
R = Readable bit W = Writable bit			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 Reset							other Resets	

bit 7-0 LTH<7:0>: ADC Lower Threshold LSB. LTH and UTH are compared with ERR to set the ADUTHR and ADLTHR bits of ADSTAT. Depending on the setting of ADTMD, an interrupt may be triggered by the results of this comparison.

REGISTER 36-33: ADUTHH: ADC UPPER THRESHOLD HIGH BYTE REGISTER

'0' = Bit is cleared

UTH<15:8>											
	UTH<15:8>										
bit 7	bit 0										
Legend:											

Legena.		
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 **UTH<15:8>**: ADC Upper Threshold MSB. LTH and UTH are compared with ERR to set the ADUTHR and ADLTHR bits of ADSTAT. Depending on the setting of ADTMD, an interrupt may be triggered by the results of this comparison.

REGISTER 36-34: ADUTHL: ADC UPPER THRESHOLD LOW 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	
UTH<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 **UTH<7:0>**: ADC Upper Threshold LSB. LTH and UTH are compared with ERR to set the ADUTHR and ADLTHR bits of ADSTAT. Depending on the setting of ADTMD, an interrupt may be triggered by the results of this comparison.

© 2017 Microchip Technology Inc.

= Bit is set

40.0 IN-CIRCUIT SERIAL PROGRAMMING™ (ICSP™)

ICSP[™] programming allows customers to manufacture circuit boards with unprogrammed devices. Programming can be done after the assembly process, allowing the device to be programmed with the most recent firmware or a custom firmware. Five pins are needed for ICSP[™] programming:

- ICSPCLK
- ICSPDAT
- MCLR/VPP
- VDD
- Vss

In Program/Verify mode the program memory, User IDs and the Configuration Words are programmed through serial communications. The ICSPDAT pin is a bidirectional I/O used for transferring the serial data and the ICSPCLK pin is the clock input. For more information on ICSP™ refer to the "PIC18F26/27/45/ 46/47/55/56/57K42 *Memory Programming Specification*" (DS40001886).

40.1 High-Voltage Programming Entry Mode

The device is placed into High-Voltage Programming Entry mode by holding the ICSPCLK and ICSPDAT pins low then raising the voltage on MCLR/VPP to VIHH.

40.2 Low-Voltage Programming Entry Mode

The Low-Voltage Programming Entry mode allows the PIC[®] Flash MCUs to be programmed using VDD only, without high voltage. When the LVP bit of Configuration Words is set to '1', the low-voltage ICSP™ programming entry is enabled. To disable the Low-Voltage ICSP mode, the LVP bit must be programmed to '0'.

Entry into the Low-Voltage Programming Entry mode requires the following steps:

- 1. MCLR is brought to VIL.
- 2. A 32-bit key sequence is presented on ICSPDAT, while clocking ICSPCLK.

Once the key sequence is complete, MCLR must be held at VIL for as long as Program/Verify mode is to be maintained.

If low-voltage programming is enabled (LVP = 1), the MCLR Reset function is automatically enabled and cannot be disabled. See **Section 6.5** "MCLR" for more information.

The LVP bit can only be reprogrammed to '0' by using the High-Voltage Programming mode.

40.3 Common Programming Interfaces

Connection to a target device is typically done through an ICSP[™] header. A commonly found connector on development tools is the RJ-11 in the 6P6C (6-pin, 6connector) configuration. See Figure 40-1.





Another connector often found in use with the PICkit[™] programmers is a standard 6-pin header with 0.1 inch spacing. Refer to Figure 40-2.

For additional interface recommendations, refer to your specific device programmer manual prior to PCB design.

It is recommended that isolation devices be used to separate the programming pins from other circuitry. The type of isolation is highly dependent on the specific application and may include devices such as resistors, diodes, or even jumpers. See Figure 40-3 for more information.

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Register on page
3989h	IPR9	—	—	—	_	CLC3IP	CWG3IP	CCP3IP	TMR6IP	165
3988h	IPR8	TMR5GIP	TMR5IP	_	_	—	—	—		164
3987h	IPR7		_	INT2IP	CLC2IP	CWG2IP	-	CCP2IP	TMR4IP	164
3986h	IPR6	TMR3GIP	TMR3IP	U2IP	U2EIP	U2TXIP	U2RXIP	I2C2EIP	I2C2IP	163
3985h	IPR5	I2C2TXIP	I2C2RXIP	DMA2AIP	DMA2ORIP	DMA2DCN- TIP	DMA2SCN- TIP	C2IP	INT1IP	162
3984h	IPR4	CLC1IP	CWG1IP	NCO1IP	—	CCP1IP	TMR2IP	TMR1GIP	TMR1IP	161
3983h	IPR3	TMR0IP	U1IP	U1EIP	U1TXIP	U1RXIP	I2C1EIP	I2C1IP	I2C1TXIP	160
3982h	IPR2	I2C1RXIP	SPI1IP	SPI1TXIP	SPI1RXIP	DMA1AIP	DMA10RIP	DMA1DCN- TIP	DMA1SCNTIP	159
3981h	IPR1	SMT1PWAIP	SMT1PRAIP	SMT1IP	C1IP	ADTIP	ADIP	ZCDIP	INT0IP	158
3980h	IPR0	IOCIP	CRCIP	SCANIP	NVMIP	CSWIP	OSFIP	HLVDIP	SWIP	157
397Fh - 397Eh	—				Unimple	emented				
397Dh	SCANTRIG	—	—	—	—	TSEL				226
397Ch	SCANCON0	EN	TRIGEN	SGO	_	_	MREG	BURSTMD	BUSY	222
397Bh	SCANHADRU	—	—			F	HADR			
397Ah	SCANHADRH				HA	DR				225
3979h	SCANHADRL				HA	DR				225
3978h	SCANLADRU		—			LADR				
3977h	SCANLADRH			LADR					223	
3976h	SCANLADRL	LADR						224		
3975h - 396Ah	—				Unimple	emented				
3969h	CRCCON1		DLEI	N			P	LEN		218
3968h	CRCCON0	EN	CRCGO	BUSY	ACCM	—	—	SHIFTM	FULL	218
3967h	CRCXORH	X15	X14	X13	X12	X11	X10	X9	X8	221
3966h	CRCXORL	X7	X6	X5	X4	X3	X2	X1	—	221
3965h	CRCSHIFTH	SHFT15	SHFT14	SHFT13	SHFT12	SHFT11	SHFT10	SHFT9	SHFT8	220
3964h	CRCSHIFTL	SHFT7	SHFT6	SHFT5	SHFT4	SHFT3	SHFT2	SHFT1	SHFT0	220
3963h	CRCACCH	ACC15	ACC14	ACC13	ACC12	ACC11	ACC10	ACC9	ACC8	219
3962h	CRCACCL	ACC7	ACC6	ACC5	ACC4	ACC3	ACC2	ACC1	ACC0	220
3961h	CRCDATH	DATA15	DATA14	DATA13	DATA12	DATA11	DATA10	DATA9	DATA8	219
3960h	CRCDATL	DATA7	DATA6	DATA5	DATA4	DATA3	DATA2	DATA1	DATA0	219
395Fh	WDTTMR			WDTTMR			STATE	PS	SCNT	185
395Eh	WDTPSH				PS	CNT				184
395Dh	WDTPSL				PS	CNT				184
395Ch	WDTCON1	—		CS		_		WINDOW		183
395Bh	WDTCON0		—			PS			SEN	182
395Ah - 38A0h	—				Unimple	emented				
389Fh	IVTADU				A	D				167
389Eh	IVTADH				A	D				167
389Dh	IVTADL				A	D				167
389Ch - 3891h	—				Unimple	emented				
3890h	PRODH_SHAD				PRO	DDH				125
388Fh	PRODL_SHAD				PR	DDL				125
388Eh	FSR2H_SHAD	_	_			F	SR2H			125

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

own, u angea, unimpiemente

Note 1: Unimplemented in LF devices.

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

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

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



FIGURE 44-14: SPI MASTER MODE TIMING (CKE = 0, SMP = 0)





46.1 Package Details

The following sections give the technical details of the packages.

28-Lead Skinny Plastic Dual In-Line (SP) – 300 mil Body [SPDIP]

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



	Units	INCHES			
Dimensior	n Limits	MIN	NOM	MAX	
Number of Pins	Ν		28		
Pitch	е		.100 BSC		
Top to Seating Plane	Α	-	-	.200	
Molded Package Thickness	A2	.120	.135	.150	
Base to Seating Plane	A1	.015	-	—	
Shoulder to Shoulder Width	E	.290	.310	.335	
Molded Package Width	E1	.240	.285	.295	
Overall Length	D	1.345	1.365	1.400	
Tip to Seating Plane	L	.110	.130	.150	
Lead Thickness	С	.008	.010	.015	
Upper Lead Width	b1	.040	.050	.070	
Lower Lead Width	b	.014	.018	.022	
Overall Row Spacing §	eB	_	_	.430	

Notes:

- 1. Pin 1 visual index feature may vary, but must be located within the hatched area.
- 2. § Significant Characteristic.
- 3. Dimensions D and E1 do not include mold flash or protrusions. Mold flash or protrusions shall not exceed .010" per side.
- 4. Dimensioning and tolerancing per ASME Y14.5M.

BSC: Basic Dimension. Theoretically exact value shown without tolerances.

Microchip Technology Drawing C04-070B

28-Lead Plastic Quad Flat, No Lead Package (ML) - 6x6 mm Body [QFN] With 0.55 mm Terminal Length





Microchip Technology Drawing C04-105C Sheet 1 of 2