

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

E·XFI

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, LVD, 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)	1.8V ~ 3.6V
Data Converters	A/D 35x10b; D/A 1x5b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 85°C (TA)
Mounting Type	Surface Mount
Package / Case	44-VQFN Exposed Pad
Supplier Device Package	44-QFN (8x8)
Purchase URL	https://www.e-xfl.com/product-detail/microchip-technology/pic18lf45k40t-i-ml

Email: info@E-XFL.COM

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



FIGURE 4-1:

7.5 Register Definitions: Peripheral Module Disable

R/W-0/0	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	
SYSCM	D FVRMD	HLVDMD	CRCMD	SCANMD	NVMMD	CLKRMD	IOCMD	
7						······································	0	
Legend:								
R = Readable bit W = Writable bit U = Unimplemented bit, read as '0'								
u = Bit is u	unchanged	x = Bit is unkn	own	-n/n = Value a	t POR and BO	R/Value at all o	ther Resets	
'1' = Bit is	set	'0' = Bit is clea	ared	q = Value dep	ends on condi	tion		
bit 7	SYSCMD: D See descript 1 = System 0 = System	isable Periphera ion in Section 7 clock network di clock network ei	Il System Cloo .4 "System C sabled (Fosc nabled	ck Network bit ⁽¹⁾ Clock Disable".)				
bit 6	FVRMD: Dis 1 = FVR mo 0 = FVR mo	able Fixed Volta dule disabled dule enabled	ge Reference	e bit				
bit 5	HLVDMD:D 1 = HLVD n 0 = HLVD n	isable Low-Volta nodule disabled nodule enabled	ge Detect bit					
bit 4	CRCMD: Dis 1 = CRC mo 0 = CRC mo	able CRC Engir odule disabled odule enabled	ne bit					
bit 3	bit 3 SCANMD: Disable NVM Memory Scanner bit ⁽²⁾ 1 = NVM Memory Scan module disabled 0 = NVM Memory Scan module enabled							
bit 2	bit 2 NVMMD: NVM Module Disable bit ⁽³⁾ 1 = All Memory reading and writing is disabled; NVMCON registers cannot be written 0 = NVM module enabled							
bit 1	bit 1 CLKRMD: Disable Clock Reference bit 1 = CLKR module disabled 0 = CLKR module enabled							
bit 0	IOCMD: Disa 1 = IOC mod 0 = IOC mod	able Interrupt-on dule(s) disabled dule(s) enabled	-Change bit, <i>i</i>	All Ports				
Note 1:	Clearing the SYS	SCMD bit disable ot affected.	es the system	ı clock (Fosc) to	peripherals, h	owever periphe	rals clocked	

REGISTER 7-1: PMD0: PMD CONTROL REGISTER 0

- 2: Subject to SCANE bit in CONFIG4H.
- **3:** When enabling NVM, a delay of up to 1 µs may be required before accessing data.

10.2.3 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

10.2.3.1 Computed GOTO

A computed GOTO is accomplished by adding an offset to the program counter. An example is shown in Example 10-2.

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 10-2: COMPUTED GOTO USING AN OFFSET VALUE

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

10.2.3.2 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. Data is transferred to or from program memory one byte at a time.

Table read and table write operations are discussed further in Section 11.1.1 "Table Reads and Table Writes".

© 2015-2017 Microchip Technology Inc.

11.1.3 READING THE PROGRAM FLASH MEMORY

The TBLRD instruction retrieves data from program memory and places it into data RAM. Table reads from program memory are performed one byte at a time.

TBLPTR points to a byte address in program space. Executing TBLRD places the byte pointed to into TABLAT. In addition, TBLPTR can be modified automatically for the next table read operation. The CPU operation is suspended during the read, and it resumes immediately after. From the user point of view, TABLAT is valid in the next instruction cycle.

The internal program memory is typically organized by words. The Least Significant bit of the address selects between the high and low bytes of the word. Figure 11-4 shows the interface between the internal program memory and the TABLAT.

FIGURE 11-4: READS FROM PROGRAM FLASH MEMORY



EXAMPLE 11-1: READING A PROGRAM FLASH MEMORY WORD

	NOTITI			The dimptomp of the black being
	MOVLW	CODE_ADDR_UPPER	;	Load TELPTR with the base
	MOVWF	TBLPTRU	;	address of the word
	MOVLW	CODE_ADDR_HIGH		
	MOVWF	TBLPTRH		
	MOVLW	CODE_ADDR_LOW		
	MOVWF	TBLPTRL		
READ_WORD				
	TBLRD*+		;	read into TABLAT and increment
	MOVF	TABLAT, W	;	get data
	MOVWF	WORD_EVEN		
	TBLRD*+		;	read into TABLAT and increment
	MOVFW	TABLAT, W	;	get data
	MOVF	WORD_ODD		

13.2 Register Definitions: CRC and Scanner Control

Long bit name prefixes for the CRC and Scanner peripherals are shown in Table 13-1. Refer to **Section 1.4.2.2 "Long Bit Names"** for more information.

TABLE 13-1:

Peripheral	Bit Name Prefix		
CRC	CRC		

REGISTER 13-1: CRCCON0: CRC CONTROL REGISTER 0

R/W-0/0	R/W-0/0	R-0	R/W-0/0	U-0	U-0	R/W-0/0	R-0
EN	GO	BUSY	ACCM	—	—	SHIFTM	FULL
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	 EN: CRC Enable bit 1 = CRC module is released from Reset 0 = CRC is disabled and consumes no operating current
bit 6	GO: CRC Start bit 1 = Start CRC serial shifter 0 = CRC serial shifter turned off
bit 5	BUSY: CRC Busy bit 1 = Shifting in progress or pending 0 = All valid bits in shifter have been shifted into accumulator and EMPTY = 1
bit 4	ACCM: Accumulator Mode bit 1 = Data is augmented with zeros 0 = Data is not augmented with zeros
bit 3-2	Unimplemented: Read as '0'
bit 1	SHIFTM: Shift Mode bit 1 = Shift right (LSb) 0 = Shift left (MSb)
bit 0	FULL: Data Path Full Indicator bit 1 = CRCDATH/L registers are full 0 = CRCDATH/L registers have shifted their data into the shifter

REGISTER 13-2: CRCCON1: CRC CONTROL REGISTER 1

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
DLEN<3:0>				PLEN<3:0>			
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 7-4	DLEN<3: Denotes t	0>: Data Length bits the length of the data word -1 (S	ee Example 13-1)					

bit 3-0 PLEN<3:0>: Polynomial Length bits

Denotes the length of the polynomial -1 (See Example 13-1)

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		
HADR<15:8> ^(1, 2)									
bit 7 bit 0									
Legend:									
R = Readable b	bit	W = Writable	bit	U = Unimpler	mented bit, read	d as '0'			
u = Bit is unchanged x = Bit is unknown -n/n = Value at POR and BOR/Value at all other Rese						other Resets			
'1' = Bit is set		'0' = Bit is clea	ared						

REGISTER 13-16: SCANHADRH: SCAN HIGH ADDRESS HIGH BYTE REGISTER

bit 7-0 HADR<15:8>: Scan End Address bits^(1, 2)

Most Significant bits of the address at the end of the designated scan

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

REGISTER 13-17: SCANHADRL: SCAN HIGH ADDRESS 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		
HADR<7:0> ^(1, 2)									
bit 7							bit 0		
Legend:									
R = Readable bit W = Writable bit				U = Unimpler	nented bit, read	l as '0'			
u = Bit is uncha	anged	x = Bit is unkr	nown	own -n/n = Value at POR and BOR/Value at all other Res					
'1' = Bit is set		'0' = Bit is clea	ared						

bit 7-0 HADR<7:0>: Scan End Address bits^(1, 2)

Least Significant bits of the address at the end of the designated scan

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

13.3 CRC Functional Overview

The CRC module can be used to detect bit errors in the Flash 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 13-1: CRC EXAMPLE



19.9 Timer1/3/5 Interrupt

The Timer1/3/5 register pair (TMRxH:TMRxL) increments to FFFFh and rolls over to 0000h. When Timer1/3/5 rolls over, the Timer1/3/5 interrupt flag bit of the PIR4 register is set. To enable the interrupt-on-rollover, you must set these bits:

- TMRxON bit of the TxCON register
- TMRxIE bits of the PIE4 register
- PEIE/GIEL bit of the INTCON register
- · GIE/GIEH bit of the INTCON register

The interrupt is cleared by clearing the TMRxIF bit in the Interrupt Service Routine.

For more information on selecting high or low priority status for the Timer1/3/5 Overflow Interrupt, see **Section 14.0 "Interrupts"**.

Note: The TMRxH:TMRxL register pair and the TMRxIF bit should be cleared before enabling interrupts.

19.10 Timer1/3/5 Operation During Sleep

Timer1/3/5 can only operate during Sleep when set up in Asynchronous Counter mode. In this mode, an external crystal or clock source can be used to increment the counter. To set up the timer to wake the device:

- TMRxON bit of the TxCON register must be set
- TMRxIE bit of the PIE4 register must be set
- PEIE/GIEL bit of the INTCON register must be set
- TxSYNC bit of the TxCON register must be set
- Configure the TMRxCLK register for using secondary oscillator as the clock source
- Enable the SOSCEN bit of the OSCEN register (Register 4-7)

The device will wake-up on an overflow and execute the next instruction. If the GIE/GIEH bit of the INTCON register is set, the device will call the Interrupt Service Routine.

The secondary oscillator will continue to operate in Sleep regardless of the $\overline{\text{TxSYNC}}$ bit setting.

19.11 CCP Capture/Compare Time Base

The CCP modules use the TMRxH:TMRxL register pair as the time base when operating in Capture or Compare mode.

In Capture mode, the value in the TMRxH:TMRxL register pair is copied into the CCPRxH:CCPRxL register pair on a configured event.

In Compare mode, an event is triggered when the value in the CCPRxH:CCPRxL register pair matches the value in the TMRxH:TMRxL register pair. This event can be a Special Event Trigger.

For more information, see Section 21.0 "Capture/Compare/PWM Module".

19.12 CCP Special Event Trigger

When any of the CCP's are configured to trigger a special event, the trigger will clear the TMRxH:TMRxL register pair. This special event does not cause a Timer1/3/5 interrupt. The CCP module may still be configured to generate a CCP interrupt.

In this mode of operation, the CCPRxH:CCPRxL register pair becomes the period register for Timer1/3/5.

Timer1/3/5 should be synchronized and FOSC/4 should be selected as the clock source in order to utilize the Special Event Trigger. Asynchronous operation of Timer1/3/5 can cause a Special Event Trigger to be missed.

In the event that a write to TMRxH or TMRxL coincides with a Special Event Trigger from the CCP, the write will take precedence.

20.5.8 LEVEL RESET, EDGE-TRIGGERED HARDWARE LIMIT ONE-SHOT MODES

In Level -Triggered One-Shot mode the timer count is reset on the external signal level and starts counting on the rising/falling edge of the transition from Reset level to the active level while the ON bit is set. Reset levels are selected as follows:

- Low Reset level (MODE<4:0> = 01110)
- High Reset level (MODE<4:0> = 01111)

When the timer count matches the PRx period count, the timer is reset and the ON bit is cleared. When the ON bit is cleared by either a PRx match or by software control a new external signal edge is required after the ON bit is set to start the counter.

When Level-Triggered Reset One-Shot mode is used in conjunction with the CCP PWM operation the PWM drive goes active with the external signal edge that starts the timer. The PWM drive goes inactive when the timer count equals the CCPRx pulse width count. The PWM drive does not go active when the timer count clears at the PRx period count match.



24.9 Dead-Band Jitter

When the rising and falling edges of the input source are asynchronous to the CWG clock, it creates jitter in the dead-band time delay. The maximum jitter is equal to one CWG clock period. Refer to Equation 24-1 for more details.

EQUATION 24-1: DEAD-BAND DELAY TIME CALCULATION

 $T_{DEAD - BAND_MIN} = \frac{1}{F_{CWG} CLOCK} \bullet DBx < 4:0>$ $T_{DEAD - BANDMAX} = \frac{1}{F_{CWG} CLOCK} \bullet DBx < 4:0>+1$ $T_{JITTER} = T_{DEAD - BAND_MAX} - T_{DEAD - BAND_MIN}$ $T_{JITTER} = \frac{1}{F_{CWG_CLOCK}}$ $T_{DEAD - BAND_MAX} = T_{DEAD - BAND_MIN} + T_{JITTER}$ EXAMPLE DBR < 4:0>= 0x0A = 10 $F_{CWG_CLOCK} = 8 MHz$ $T_{JITTER} = \frac{1}{8MHz} = 125 \text{ ns}$ $T_{DEAD - BAND_MIN} = 125 \text{ ns} *10 = 125 \text{ µs}$ $T_{DEAD - BAND_MIN} = 1.25 \text{ µs} + 0.125 \text{ µs} = 1.37 \text{µs}$

PIC18LF26/45/46K40



26.8 I²C Mode Operation

All MSSP I²C communication is byte oriented and shifted out MSb first. Six SFR registers and two interrupt flags interface the module with the PIC[®] microcontroller and user software. Two pins, SDA and SCL, are exercised by the module to communicate with other external I²C devices.

26.8.1 BYTE FORMAT

All communication in I^2C is done in 9-bit segments. A byte is sent from a master to a slave or vice-versa, followed by an Acknowledge bit sent back. After the eighth falling edge of the SCL line, the device outputting data on the SDA changes that pin to an input and reads in an acknowledge value on the next clock pulse.

The clock signal, SCL, is provided by the master. Data is valid to change while the SCL signal is low, and sampled on the rising edge of the clock. Changes on the SDA line while the SCL line is high define special conditions on the bus, explained below.

26.8.2 DEFINITION OF I²C TERMINOLOGY

There is language and terminology in the description of I^2C communication that have definitions specific to I^2C . That word usage is defined below and may be used in the rest of this document without explanation. This table was adapted from the Philips I^2C specification.

26.8.3 SDA AND SCL PINS

Selection of any I²C mode with the SSPEN bit set, forces the SCL and SDA pins to be open-drain. These pins should be set by the user to inputs by setting the appropriate TRIS bits.

- Note 1: Data is tied to output zero when an I²C mode is enabled.
 - 2: Any device pin can be selected for SDA and SCL functions with the PPS peripheral. These functions are bidirectional. The SDA input is selected with the SSPxDATPPS registers. The SCL input is selected with the SSPxCLKPPS registers. Outputs are selected with the RxyPPS registers. It is the user's responsibility to make the selections so that both the input and the output for each function is on the same pin.

26.8.4 SDA HOLD TIME

The hold time of the SDA pin is selected by the SDAHT bit of the SSPxCON3 register. Hold time is the time SDA is held valid after the falling edge of SCL. Setting the SDAHT bit selects a longer 300 ns minimum hold time and may help on buses with large capacitance.

TABLE 26-2: 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 SSPxADD.
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 hold SCL low to stall communication.
Bus Collision	Any time the SDA line is sampled low by the module while it is out- putting and expected high state.

^{© 2015-2017} Microchip Technology Inc.

26.10.13.3 Bus Collision During a Stop Condition

Bus collision occurs during a Stop condition if:

- a) After the SDA pin has been deasserted and allowed to float high, SDA is sampled low after the BRG has timed out (Case 1).
- b) After the SCL pin is deasserted, SCL is sampled low before SDA goes high (Case 2).

The Stop condition begins with SDA asserted low. When SDA is sampled low, the SCL pin is allowed to float. When the pin is sampled high (clock arbitration), the Baud Rate Generator is loaded with SSPxADD and counts down to zero. After the BRG times out, SDA is sampled. If SDA is sampled low, a bus collision has occurred. This is due to another master attempting to drive a data '0' (Figure 26-38). If the SCL pin is sampled low before SDA is allowed to float high, a bus collision occurs. This is another case of another master attempting to drive a data '0' (Figure 26-39).

FIGURE 26-38: BUS COLLISION DURING A STOP CONDITION (CASE 1)



FIGURE 26-39: BUS COLLISION DURING A STOP CONDITION (CASE 2)



27.5.1.5 Synchronous Master Reception

Data is received at the RXx/DTx pin. The RXx/DTx pin output driver is automatically disabled when the EUSART is configured for synchronous master receive operation.

In Synchronous mode, reception is enabled by setting either the Single Receive Enable bit (SREN of the RCxSTA register) or the Continuous Receive Enable bit (CREN of the RCxSTA register).

When SREN is set and CREN is clear, only as many clock cycles are generated as there are data bits in a single character. The SREN bit is automatically cleared at the completion of one character. When CREN is set, clocks are continuously generated until CREN is cleared. If CREN is cleared in the middle of a character the CK clock stops immediately and the partial character is discarded. If SREN and CREN are both set, then SREN is cleared at the completion of the first character and CREN takes precedence.

To initiate reception, set either SREN or CREN. Data is sampled at the RXx/DTx pin on the trailing edge of the TX/CK clock pin and is shifted into the Receive Shift Register (RSR). When a complete character is received into the RSR, the RCxIF bit is set and the character is automatically transferred to the two character receive FIFO. The Least Significant eight bits of the top character in the receive FIFO are available in RCxREG. The RCxIF bit remains set as long as there are unread characters in the receive FIFO.

Note:	If the RX/DT function is on an analog pin,							
	the corresponding ANSEL bit must be							
	cleared for the receiver to function.							

27.5.1.6 Slave Clock

Synchronous data transfers use a separate clock line, which is synchronous with the data. A device configured as a slave receives the clock on the TX/CK line. The TXx/CKx pin output driver is automatically disabled when the device is configured for synchronous slave transmit or receive operation. Serial data bits change on the leading edge to ensure they are valid at the trailing edge of each clock. One data bit is transferred for each clock cycle. Only as many clock cycles should be received as there are data bits.

Note: If the device is configured as a slave and the TX/CK function is on an analog pin, the corresponding ANSEL bit must be cleared.

27.5.1.7 Receive Overrun Error

The receive FIFO buffer can hold two characters. An overrun error will be generated if a third character, in its entirety, is received before RCxREG is read to access the FIFO. When this happens the OERR bit of the RCxSTA register is set. Previous data in the FIFO will not be overwritten. The two characters in the FIFO buffer can be read, however, no additional characters will be received until the error is cleared. The OERR bit can only be cleared by clearing the overrun condition. If the overrun error occurred when the SREN bit is set and CREN is clear then the error is cleared by reading RCxREG. If the overrun occurred when the CREN bit is set then the error condition is cleared by either clearing the CREN bit of the RCxSTA register or by clearing the SPEN bit which resets the EUSART.

27.5.1.8 Receiving 9-Bit Characters

The EUSART supports 9-bit character reception. When the RX9 bit of the RCxSTA register is set the EUSART will shift nine bits into the RSR for each character received. The RX9D bit of the RCxSTA register is the ninth, and Most Significant, data bit of the top unread character in the receive FIFO. When reading 9-bit data from the receive FIFO buffer, the RX9D data bit must be read before reading the eight Least Significant bits from the RCxREG.

27.5.1.9 Synchronous Master Reception Setup:

- Initialize the SPxBRGH:SPxBRGL register pair for the appropriate baud rate. Set or clear the BRGH and BRG16 bits, as required, to achieve the desired baud rate.
- 2. Clear the ANSEL bit for the RXx pin (if applicable).
- 3. Enable the synchronous master serial port by setting bits SYNC, SPEN and CSRC.
- 4. Ensure bits CREN and SREN are clear.
- 5. If interrupts are desired, set the RCxIE bit of the PIE3 register and the GIE and PEIE bits of the INTCON register.
- 6. If 9-bit reception is desired, set bit RX9.
- 7. Start reception by setting the SREN bit or for continuous reception, set the CREN bit.
- Interrupt flag bit RCxIF will be set when reception of a character is complete. An interrupt will be generated if the enable bit RCxIE was set.
- 9. Read the RCxSTA register to get the ninth bit (if enabled) and determine if any error occurred during reception.
- 10. Read the 8-bit received data by reading the RCxREG register.
- 11. If an overrun error occurs, clear the error by either clearing the CREN bit of the RCxSTA register or by clearing the SPEN bit which resets the EUSART.

31.5.5 BURST AVERAGE MODE

The Burst Average mode (ADMD = 011) acts the same as the Average mode in most respects. The one way it differs is that it continuously retriggers ADC sampling until the ADCNT value is greater than or equal to ADRPT, even if Continuous Sampling mode (see **Section 31.5.8 "Continuous Sampling mode"**) is not enabled. This allows for a threshold comparison on the average of a short burst of ADC samples.

31.5.6 LOW-PASS FILTER MODE

The Low-pass Filter mode (ADMD = 100) acts similarly to the Average mode in how it handles samples (accumulates samples until ADCNT value greater than or equal to ADRPT, then triggers threshold comparison), but instead of a simple average, it performs a low-pass filter operation on all of the samples, reducing the effect of high-frequency noise on the average, then performs a threshold comparison on the results. (see Table 31-3 for a more detailed description of the mathematical operation). In this mode, the ADCRS bits determine the cut-off frequency of the low-pass filter (as demonstrated by Table 31-4).

31.5.7 THRESHOLD COMPARISON

At the end of each computation:

- The conversion results are latched and held stable at the end-of-conversion.
- The error is calculated based on a difference calculation which is selected by the ADCALC<2:0> bits in the ADCON3 register. The value can be one of the following calculations (see Register 31-4 for more details):
 - The first derivative of single measurements
 - The CVD result in CVD mode
 - The current result vs. a setpoint
 - The current result vs. the filtered/average result
 - The first derivative of the filtered/average value
 - Filtered/average value vs. a setpoint
- The result of the calculation (ADERR) is compared to the upper and lower thresholds, ADUTH<ADUTHH:ADUTHL> and ADLTH<ADLTHH:ADLTHL> registers, to set the ADUTHR and ADLTHR flag bits. The threshold logic is selected by ADTMD<2:0> bits in the ADCON3 register. The threshold trigger option can be one of the following:
 - Never interrupt
 - Error is less than lower threshold
 - Error is greater than or equal to lower threshold
 - Error is between thresholds (inclusive)
 - Error is outside of thresholds
 - Error is less than or equal to upper threshold
 - Error is greater than upper threshold
 - Always interrupt regardless of threshold test results
 - If the threshold condition is met, the threshold interrupt flag ADTIF is set.

Note 1: The threshold tests are signed operations.2: If ADAOV is set, a threshold interrupt is

 If ADAOV is set, a threshold interrupt is signaled.

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
—	ADCS						
bit 7							bit 0
Legend:							
R = Readable	e bit	W = Writable	bit	U = Unimpler	nented bit, read	d as '0'	
u = Bit is uncl	hanged	x = Bit is unkr	iown	-n/n = Value a	at POR and BC	R/Value at all	other Resets
'1' = Bit is set		'0' = Bit is clea	ared				
bit 7-6	Unimplemer	nted: Read as '	כ'				
bit 5-0	ADCS<5:0>:	ADC Conversi	on Clock Sele	ect bits			
	111111 = F c	osc/128					
111110 = Fosc/126							
111101 = Fosc/124							
	•						
	•						
	•						
000000 = Fosc/2							

REGISTER 31-6: ADCLK: ADC CLOCK SELECTION REGISTER

REGISTER 31-7: ADREF: ADC REFERENCE SELECTION REGISTER

U-0	U-0	U-0	R/W-0/0) U-0 U-0		R/W-0/0	R/W-0/0
		—	ADNREF	—	—	ADPRE	F<1:0>
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 7-5 bit 4	Unimplemented: Read as '0' ADNREF: ADC Negative Voltage Reference Selection bit 1 = VREF- is connected to external VREF- 0 = VREF- is connected to AVss
bit 3-2	Unimplemented: Read as '0'
bit 1-0	ADPREF: ADC Positive Voltage Reference Selection bits 11 = VREF+ is connected to internal Fixed Voltage Reference (FVR) module 10 = VREF+ is connected to external VREF+ 01 = Reserved 00 = VREF+ is connected to VDD

35.2.3 BYTE-ORIENTED AND BIT-ORIENTED INSTRUCTIONS IN INDEXED LITERAL OFFSET MODE

Note:	Enabling	the	PIC18	instruction	set		
	extension	may	cause leg	gacy applicat	tions		
	to behave	errat	tically or fail entirely.				

In addition to eight new commands in the extended set, enabling the extended instruction set also enables Indexed Literal Offset Addressing mode (Section 10.7.1 "Indexed Addressing with Literal Offset"). This has a significant impact on the way that many commands of the standard PIC18 instruction set are interpreted.

When the extended set is disabled, addresses embedded in opcodes are treated as literal memory locations: either as a location in the Access Bank ('a' = 0), or in a GPR bank designated by the BSR ('a' = 1). When the extended instruction set is enabled and 'a' = 0, however, a file register argument of 5Fh or less is interpreted as an offset from the pointer value in FSR2 and not as a literal address. For practical purposes, this means that all instructions that use the Access RAM bit as an argument – that is, all byte-oriented and bitoriented instructions, or almost half of the core PIC18 instructions – may behave differently when the extended instruction set is enabled.

When the content of FSR2 is 00h, the boundaries of the Access RAM are essentially remapped to their original values. This may be useful in creating backward compatible code. If this technique is used, it may be necessary to save the value of FSR2 and restore it when moving back and forth between C and assembly routines in order to preserve the Stack Pointer. Users must also keep in mind the syntax requirements of the extended instruction set (see Section 35.2.3.1 "Extended Instruction Syntax with Standard PIC18 Commands").

Although the Indexed Literal Offset Addressing mode can be very useful for dynamic stack and pointer manipulation, it can also be very annoying if a simple arithmetic operation is carried out on the wrong register. Users who are accustomed to the PIC18 programming must keep in mind that, when the extended instruction set is enabled, register addresses of 5Fh or less are used for Indexed Literal Offset Addressing.

Representative examples of typical byte-oriented and bit-oriented instructions in the Indexed Literal Offset Addressing mode are provided on the following page to show how execution is affected. The operand conditions shown in the examples are applicable to all instructions of these types.

35.2.3.1 Extended Instruction Syntax with Standard PIC18 Commands

When the extended instruction set is enabled, the file register argument, 'f', in the standard byte-oriented and bit-oriented commands is replaced with the literal offset value, 'k'. As already noted, this occurs only when 'f' is less than or equal to 5Fh. When an offset value is used, it must be indicated by square brackets ("[]"). As with the extended instructions, the use of brackets indicates to the compiler that the value is to be interpreted as an index or an offset. Omitting the brackets, or using a value greater than 5Fh within brackets, will generate an error in the MPASM assembler.

If the index argument is properly bracketed for Indexed Literal Offset Addressing, the Access RAM argument is never specified; it will automatically be assumed to be '0'. This is in contrast to standard operation (extended instruction set disabled) when 'a' is set on the basis of the target address. Declaring the Access RAM bit in this mode will also generate an error in the MPASM assembler.

The destination argument, 'd', functions as before.

In the latest versions of the MPASMTM assembler, language support for the extended instruction set must be explicitly invoked. This is done with either the command line option, $/_Y$, or the PE directive in the source listing.

35.2.4 CONSIDERATIONS WHEN ENABLING THE EXTENDED INSTRUCTION SET

It is important to note that the extensions to the instruction set may not be beneficial to all users. In particular, users who are not writing code that uses a software stack may not benefit from using the extensions to the instruction set.

Additionally, the Indexed Literal Offset Addressing mode may create issues with legacy applications written to the PIC18 assembler. This is because instructions in the legacy code may attempt to address registers in the Access Bank below 5Fh. Since these addresses are interpreted as literal offsets to FSR2 when the instruction set extension is enabled, the application may read or write to the wrong data addresses.

When porting an application to the PIC18(L)F2x/ 4xK40, it is very important to consider the type of code. A large, re-entrant application that is written in 'C' and would benefit from efficient compilation will do well when using the instruction set extensions. Legacy applications that heavily use the Access Bank will most likely not benefit from using the extended instruction set.

37.4 AC Characteristics





FIGURE 37-13: CAPTURE/COMPARE/PWM TIMINGS (CCP)



TABLE 37-20: CAPTURE/COMPARE/PWM REQUIREMENTS (CCP)

Standard Operating Conditions (unless otherwise stated)Operating Temperature $-40^{\circ}C \le TA \le +125^{\circ}C$									
Param No. Sym. Characteristic				Min.	Тур†	Max.	Units	Conditions	
CC01*	TccL	CCPx Input Low Time	No Prescaler	0.5Tcy + 20	—	—	ns		
			With Prescaler	20	_	_	ns		
CC02*	ТссН	CCPx Input High Time	No Prescaler	0.5Tcy + 20	_	_	ns		
			With Prescaler	20	_	_	ns		
CC03*	TccP	CCPx Input Period		<u>3Tcy + 40</u> N	—	—	ns	N = prescale value	

* These parameters are characterized but not tested.

† Data in "Typ" column is at 3.0V, 25°C unless otherwise stated. These parameters are for design guidance only and are not tested.