



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	25MHz
Connectivity	I ² C, SPI, UART/USART
Peripherals	Brown-out Detect/Reset, HLVD, POR, PWM, WDT
Number of I/O	36
Program Memory Size	8KB (4K x 16)
Program Memory Type	FLASH
EEPROM Size	256 x 8
RAM Size	512 x 8
Voltage - Supply (Vcc/Vdd)	4.2V ~ 5.5V
Data Converters	A/D 13x10b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 125°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/pic18f4321-e-ml

Email: info@E-XFL.COM

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

3.4 RC Oscillator

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

- supply voltage
- values of the external resistor (REXT) and capacitor (CEXT)
- · operating temperature

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

- normal manufacturing variation
- difference in lead frame capacitance between package types (especially for low CEXT values)
- variations within the tolerance of limits of $\ensuremath{\mathsf{REXT}}$ and $\ensuremath{\mathsf{CEXT}}$

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





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





3.5 PLL Frequency Multiplier

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

3.5.1 HSPLL OSCILLATOR MODE

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

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





3.5.2 PLL AND INTOSC

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

7.3 Reading the Flash Program Memory

The TBLRD instruction is used to retrieve data from program memory and place 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 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 7-4 shows the interface between the internal program memory and the TABLAT.

FIGURE 7-4: READS FROM FLASH PROGRAM MEMORY



EXAMPLE 7-1: READING A FLASH PROGRAM MEMORY WORD

	MOVLW MOVWF MOVLW MOVLW MOVLW	CODE_ADDR_UPPER TBLPTRU CODE_ADDR_HIGH TBLPTRH CODE_ADDR_LOW TBLPTRL	;;	Load TBLPTR with the base address of the word
READ_WORD				
	TBLRD*+		;	read into TABLAT and increment
	MOVF	TABLAT, W	;	get data
	MOVWF	WORD EVEN		
	TBLRD*+	—	;	read into TABLAT and increment
	MOVF	TABLAT, W	;	get data
	MOVWF	WORD_ODD		

9.0 8 x 8 HARDWARE MULTIPLIER

9.1 Introduction

All PIC18 devices include an 8 x 8 hardware multiplier as part of the ALU. The multiplier performs an unsigned operation and yields a 16-bit result that is stored in the product register pair, PRODH:PRODL. The multiplier's operation does not affect any flags in the STATUS register.

Making multiplication a hardware operation allows it to be completed in a single instruction cycle. This has the advantages of higher computational throughput and reduced code size for multiplication algorithms and allows the PIC18 devices to be used in many applications previously reserved for digital signal processors. A comparison of various hardware and software multiply operations, along with the savings in memory and execution time, is shown in Table 9-1.

9.2 Operation

Example 9-1 shows the instruction sequence for an 8×8 unsigned multiplication. Only one instruction is required when one of the arguments is already loaded in the WREG register.

Example 9-2 shows the sequence to do an 8 x 8 signed multiplication. To account for the sign bits of the arguments, each argument's Most Significant bit (MSb) is tested and the appropriate subtractions are done.

EXAMPLE 9-1: 8 x 8 UNSIGNED MULTIPLY ROUTINE

MOVF	ARG1,	w ;	
MULWF	ARG2	;	ARG1 * ARG2 ->
		;	PRODH: PRODL

EXAMPLE 9-2: 8 x 8 SIGNED MULTIPLY

MOVF	ARG1, W									
MULWF	ARG2	; ARG1 * ARG2 ->								
		; PRODH:PRODL								
BTFSC	ARG2, SB	; Test Sign Bit								
SUBWF	PRODH, F	; PRODH = PRODH								
		; – ARG1								
MOVF	ARG2, W									
BTFSC	ARG1, SB	; Test Sign Bit								
SUBWF	PRODH, F	; PRODH = PRODH								
		; – ARG2								

		Program	Cycles	Time			
Routine	Multiply Method	Multiply Method Memory (Words)		@ 40 MHz	@ 10 MHz	@ 4 MHz	
9 x 9 upsigned	Without hardware multiply	13	69	6.9 μs	27.6 μs	69 µs	
8 x 8 unsigned	Hardware multiply	1	1	100 ns	400 ns	1 μs	
	Without hardware multiply	33	91	9.1 μs	36.4 μs	91 μs	
o x o signed	Hardware multiply	6	6	600 ns	2.4 μs	6 μs	
16 x 16 uppigned	Without hardware multiply	21	242	24.2 μs	96.8 μs	242 μs	
To x To unsigned	Hardware multiply	28	28	2.8 μs	11.2 μs	28 μs	
16 x 16 signed	Without hardware multiply	52	254	25.4 μs	102.6 μs	254 μs	
TO X TO SIGNED	Hardware multiply	35	40	4.0 μs	16.0 μs	40 μs	

TABLE 9-1: PERFORMANCE COMPARISON FOR VARIOUS MULTIPLY OPERATIONS

		0.0011			1
Pin	Function	TRIS Setting	I/O	l/O Type	Description
RD0/PSP0	RD0	0	0	DIG	LATD<0> data output.
		1	I	ST	PORTD<0> data input.
	PSP0	х	0	DIG	PSP read data output (LATD<0>); takes priority over port data.
		х	I	TTL	PSP write data input.
RD1/PSP1	RD1	0	0	DIG	LATD<1> data output.
		1		ST	PORTD<1> data input.
	PSP1	х	0	DIG	PSP read data output (LATD<1>); takes priority over port data.
		х	Ι	TTL	PSP write data input.
RD2/PSP2	RD2	0	0	DIG	LATD<2> data output.
		1	Ι	ST	PORTD<2> data input.
	PSP2	х	0	DIG	PSP read data output (LATD<2>); takes priority over port data.
		х	I	TTL	PSP write data input.
RD3/PSP3	RD3	0	0	DIG	LATD<3> data output.
		1	I	ST	PORTD<3> data input.
	PSP3	х	0	DIG	PSP read data output (LATD<3>); takes priority over port data.
		х	Ι	TTL	PSP write data input.
RD4/PSP4	RD4	0	0	DIG	LATD<4> data output.
		1	I	ST	PORTD<4> data input.
	PSP4	х	0	DIG	PSP read data output (LATD<4>); takes priority over port data.
		x	I	TTL	PSP write data input.
RD5/PSP5/P1B	RD5	0	0	DIG	LATD<5> data output.
		1	I	ST	PORTD<5> data input.
	PSP5	х	0	DIG	PSP read data output (LATD<5>); takes priority over port data.
		х	I	TTL	PSP write data input.
	P1B	0	0	DIG	ECCP1 Enhanced PWM output, Channel B; takes priority over port and PSP data. May be configured for tri-state during Enhanced PWM shutdown events.
RD6/PSP6/P1C	RD6	0	0	DIG	LATD<6> data output.
		1	I	ST	PORTD<6> data input.
	PSP6	х	0	DIG	PSP read data output (LATD<6>); takes priority over port data.
		х	I	TTL	PSP write data input.
	P1C	0	0	DIG	ECCP1 Enhanced PWM output, channel C; takes priority over port and PSP data. May be configured for tri-state during Enhanced PWM shutdown events.
RD7/PSP7/P1D	RD7	0	0	DIG	LATD<7> data output.
		1	I	ST	PORTD<7> data input.
	PSP7	x	0	DIG	PSP read data output (LATD<7>); takes priority over port data.
		х	Ι	TTL	PSP write data input.
	P1D	0	0	DIG	ECCP1 Enhanced PWM output, Channel D; takes priority over port and PSP data. May be configured for tri-state during Enhanced PWM shutdown events.

TABLE 11-7: PORTD I/O SUMMARY

Legend: DIG = Digital level output; TTL = TTL input buffer; ST = Schmitt Trigger input buffer; x = Don't care (TRIS bit does not affect port direction or is overridden for this option).

12.0 TIMER0 MODULE

The Timer0 module incorporates the following features:

- Software selectable operation as a timer or counter in both 8-bit or 16-bit modes
- · Readable and writable registers
- Dedicated 8-bit, software programmable
 prescaler
- Selectable clock source (internal or external)
- Edge select for external clock
- · Interrupt-on-overflow

The T0CON register (Register 12-1) controls all aspects of the module's operation, including the prescale selection. It is both readable and writable.

A simplified block diagram of the Timer0 module in 8-bit mode is shown in Figure 12-1. Figure 12-2 shows a simplified block diagram of the Timer0 module in 16-bit mode.

REGISTER 12-1: T0CON: TIMER0 CONTROL 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
TMR0ON	T08BIT	TOCS	T0SE	PSA	T0PS2	T0PS1	T0PS0
bit 7							bit 0

- bit 7 TMR0ON: Timer0 On/Off Control bit
 - 1 = Enables Timer0
 - 0 = Stops Timer0
- bit 6 **T08BIT**: Timer0 8-Bit/16-Bit Control bit
 - 1 = Timer0 is configured as an 8-bit timer/counter
 - 0 = Timer0 is configured as a 16-bit timer/counter
- bit 5 **TOCS**: Timer0 Clock Source Select bit
 - 1 = Transition on T0CKI pin
 - 0 = Internal instruction cycle clock (CLKO)
- bit 4 TOSE: Timer0 Source Edge Select bit
 - 1 = Increment on high-to-low transition on TOCKI pin
 - 0 = Increment on low-to-high transition on T0CKI pin
- bit 3 **PSA**: Timer0 Prescaler Assignment bit
 - 1 = TImer0 prescaler is NOT assigned. Timer0 clock input bypasses prescaler.
 - 0 = Timer0 prescaler is assigned. Timer0 clock input comes from prescaler output.
- bit 2-0 TOPS<2:0>: Timer0 Prescaler Select bits
 - 111 = 1:256 Prescale value
 - 110 = 1:128 Prescale value
 - 101 = 1:64 Prescale value
 - 100 = 1:32 Prescale value
 - 011 = 1:16 Prescale value
 - 010 = 1:8 Prescale value
 - 001 = 1:4 Prescale value
 - 000 = 1:2 Prescale value

Legend:

Legenu.			
R = Readable bit	W = Writable bit	U = Unimplemented b	it, read as '0'
-n = Value at POR	'1' = Bit is set	'0' = Bit is cleared	x = Bit is unknown

13.0 TIMER1 MODULE

The Timer1 timer/counter module incorporates these features:

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

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

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

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

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

IER 13-1.	TICON.												
	R/W-0	R-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0					
	RD16	T1RUN	T1CKPS1	T1CKPS0	T1OSCEN	T1SYNC	TMR1CS	TMR10N					
	bit 7							bit 0					
bit 7	RD16: 16	RD16: 16-Bit Read/Write Mode Enable bit											
	1 = Enab 0 = Enab	les register les register	read/write of read/write of	f Timer1 in or f Timer1 in tw	ie 16-bit opei o 8-bit opera	ration tions							
bit 6	T1RUN: T	ïmer1 Syst	em Clock Sta	atus bit									
	1 = Devic 0 = Devic	e clock is o clock is o	derived from derived	Timer1 oscilla another sourc	ator ce								
bit 5-4	T1CKPS<	: 1:0>: Time	er1 Input Cloc	k Prescale S	elect bits								
	11 = 1:8 F	11 = 1:8 Prescale value											
	10 = 1:4 F 01 = 1:2 F	rescale va rescale va	lue										
	00 = 1:1 F	Prescale va	lue										
bit 3	T1OSCEN	I: Timer1 C	Dscillator Ena	ble bit									
	1 = Timer	1 = Timer1 oscillator is enabled											
	0 = Limer	= Timer1 oscillator is shut off be oscillator inverter and feedback resistor are turned off to eliminate power drain											
bit 2	T1SYNC:	Timer1 Ex	ternal Clock I	nput Synchro	nization Sele	ect bit							
	When TM	When $TMR1CS = 1$:											
	1 = Do no	1 = Do not synchronize external clock input											
	0 = Synch	0 = Synchronize external clock input											
	This bit is	This bit is ignored. Timer1 uses the internal clock when TMR1CS = 0.											
bit 1	TMR1CS:	TMR1CS: Timer1 Clock Source Select bit											
	1 = Exter 0 = Intern	nal clock fr ial clock (F	om pin RC0/ [*] osc/4)	T1OSO/T130	KI (on the ris	ing edge)							
bit 0	TMR10N	TMR1ON: Timer1 On bit											
	1 = Enab 0 = Stops	les Timer1 Timer1											
	Legend:	Legend:											
	R = Read	able bit	VV = V	Vritable bit	U = Unim	plemented b	oit, read as '	D'					
	-n = Value	at POR	'1' = E	Bit is set	'0' = Bit is	cleared	x = Bit is u	nknown					

REGISTER 13-1: T1CON: TIMER1 CONTROL REGISTER

The CCPRxH register and a 2-bit internal latch are used to double-buffer the PWM duty cycle. This double-buffering is essential for glitchless PWM operation.

When the CCPRxH and 2-bit latch match TMR2, concatenated with an internal 2-bit Q clock or 2 bits of the TMR2 prescaler, the CCPx pin is cleared.

The maximum PWM resolution (bits) for a given PWM frequency is given by the equation:

EQUATION 16-3:

PWM Resolution (max) =
$$\frac{\log(\frac{FOSC}{FPWM})}{\log(2)}$$
 bits

Note: If the PWM duty cycle value is longer than the PWM period, the CCPx pin will not be cleared.

TABLE 16-4:	EXAMPLE PWM FREQUENCIES AND RESOLUTIONS AT 40 MHz

PWM Frequency	2.44 kHz	9.77 kHz	39.06 kHz	156.25 kHz	312.50 kHz	416.67 kHz
Timer Prescaler (1, 4, 16)	16	4	1	1	1	1
PR2 Value	FFh	FFh	FFh	3Fh	1Fh	17h
Maximum Resolution (bits)	10	10	10	8	7	6.58

16.4.3 PWM AUTO-SHUTDOWN (CCP1 ONLY)

The PWM auto-shutdown features of the Enhanced CCP module are also available to CCP1 in 28-pin devices. The operation of this feature is discussed in detail in **Section 17.4.7 "Enhanced PWM Auto-Shutdown"**.

Auto-shutdown features are not available for CCP2.

16.4.4 SETUP FOR PWM OPERATION

The following steps should be taken when configuring the CCP module for PWM operation:

- 1. Set the PWM period by writing to the PR2 register.
- 2. Set the PWM duty cycle by writing to the CCPRxL register and CCPxCON<5:4> bits.
- 3. Make the CCPx pin an output by clearing the appropriate TRIS bit.
- 4. Set the TMR2 prescale value, then enable Timer2 by writing to T2CON.
- 5. Configure the CCPx module for PWM operation.

18.4.6 MASTER MODE

Master mode is enabled by setting and clearing the appropriate SSPM bits in SSPCON1 and by setting the SSPEN bit. In Master mode, the SCL and SDA lines are manipulated by the MSSP hardware.

Master mode of operation is supported by interrupt generation on the detection of the Start and Stop conditions. The Stop (P) and Start (S) bits are cleared from a Reset or when the MSSP module is disabled. Control of the I^2C bus may be taken when the P bit is set, or the bus is Idle, with both the S and P bits clear.

In Firmware Controlled Master mode, user code conducts all I^2C bus operations based on Start and Stop bit conditions.

Once Master mode is enabled, the user has six options.

- 1. Assert a Start condition on SDA and SCL.
- 2. Assert a Repeated Start condition on SDA and SCL.
- 3. Write to the SSPBUF register initiating transmission of data/address.
- 4. Configure the I²C port to receive data.
- 5. Generate an Acknowledge condition at the end of a received byte of data.
- 6. Generate a Stop condition on SDA and SCL.

Note: The MSSP module, when configured in I²C Master mode, does not allow queueing of events. For instance, the user is not allowed to initiate a Start condition and immediately write the SSPBUF register to initiate transmission before the Start condition is complete. In this case, the SSPBUF will not be written to and the WCOL bit will be set, indicating that a write to the SSPBUF did not occur.

The following events will cause the MSSP Interrupt Flag bit, SSPIF, to be set (MSSP interrupt, if enabled):

- · Start condition
- Stop condition
- · Data transfer byte transmitted/received
- Acknowledge transmit
- Repeated Start



18.4.12 ACKNOWLEDGE SEQUENCE TIMING

An Acknowledge sequence is enabled by setting the Acknowledge Sequence Enable bit. ACKEN (SSPCON2<4>). When this bit is set, the SCL pin is pulled low and the contents of the Acknowledge data bit are presented on the SDA pin. If the user wishes to generate an Acknowledge, then the ACKDT bit should be cleared. If not, the user should set the ACKDT bit before starting an Acknowledge sequence. The Baud Rate Generator then counts for one rollover period (TBRG) and the SCL pin is deasserted (pulled high). When the SCL pin is sampled high (clock arbitration), the Baud Rate Generator counts for TBRG. The SCL pin is then pulled low. Following this, the ACKEN bit is automatically cleared, the Baud Rate Generator is turned off and the MSSP module then goes into Idle mode (Figure 18-25).

18.4.12.1 WCOL Status Flag

If the user writes the SSPBUF when an Acknowledge sequence is in progress, then WCOL is set and the contents of the buffer are unchanged (the write doesn't occur).

18.4.13 STOP CONDITION TIMING

A Stop bit is asserted on the SDA pin at the end of a receive/transmit by setting the Stop Sequence Enable bit, PEN (SSPCON2<2>). At the end of a receive/ transmit, the SCL line is held low after the falling edge of the ninth clock. When the PEN bit is set, the master will assert the SDA line low. When the SDA line is sampled low, the Baud Rate Generator is reloaded and counts down to 0. When the Baud Rate Generator times out, the SCL pin will be brought high and one TBRG (Baud Rate Generator rollover count) later, the SDA pin will be deasserted. When the SDA pin is sampled high while SCL is high, the P bit (SSPSTAT<4>) is set. A TBRG later, the PEN bit is cleared and the SSPIF bit is set (Figure 18-26).

18.4.13.1 WCOL Status Flag

If the user writes the SSPBUF when a Stop sequence is in progress, then the WCOL bit is set and the contents of the buffer are unchanged (the write doesn't occur).

FIGURE 18-25: ACKNOWLEDGE SEQUENCE WAVEFORM



FIGURE 18-26: STOP CONDITION RECEIVE OR TRANSMIT MODE



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

TABLE 19-3: BAUD RATES FOR ASYNCHRONOUS MODES

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

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

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

© 2009 Microchip Technology Inc.

FIGURE 19-6: EUSART RECEIVE BLOCK DIAGRAM





RX (pin)	Start bit 0 bit 1 5 bit 7/8 St	op Start bit 0 5 bit 7/8 Stop bit	Start bit ////////////////////////////////////
Rcv Shift Reg		↓ Word 1 Word 2	<u></u>
Read Rcv Buffer Reg RCREG	<u></u> \$\$	RCREG RCREG	<u></u>
RCIF (Interrupt Flag)	<u> </u>		<u>_</u>
OERR bit	<u></u>		
CREN	Ś	ζ	<u>_</u>

Note: This timing diagram shows three words appearing on the RX input. The RCREG (receive buffer) is read after the third word causing the OERR (overrun) bit to be set.

Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Reset Values on page
INTCON	GIE/GIEH	PEIE/GIEL	TMR0IE	INT0IE	RBIE	TMR0IF	INT0IF	RBIF	55
PIR1	PSPIF ⁽¹⁾	ADIF	RCIF	TXIF	SSPIF	CCP1IF	TMR2IF	TMR1IF	58
PIE1	PSPIE ⁽¹⁾	ADIE	RCIE	TXIE	SSPIE	CCP1IE	TMR2IE	TMR1IE	58
IPR1	PSPIP ⁽¹⁾	ADIP	RCIP	TXIP	SSPIP	CCP1IP	TMR2IP	TMR1IP	58
RCSTA	SPEN	RX9	SREN	CREN	ADDEN	FERR	OERR	RX9D	57
RCREG	EUSART F	Receive Regis	ster						57
TXSTA	CSRC	TX9	TXEN	SYNC	SENDB	BRGH	TRMT	TX9D	57
BAUDCON	ABDOVF	RCIDL	RXDTP	TXCKP	BRG16	_	WUE	ABDEN	57
SPBRGH	EUSART E	aud Rate Ge	enerator Reg	gister High	Byte				57
SPBRG	EUSART E	aud Rate Ge	enerator Reg	gister Low E	Byte				57

 TABLE 19-6:
 REGISTERS ASSOCIATED WITH ASYNCHRONOUS RECEPTION

Legend: — = unimplemented locations read as '0'. Shaded cells are not used for asynchronous reception.

Note 1: These bits are unimplemented on 28-pin devices and read as '0'.

19.2.5 BREAK CHARACTER SEQUENCE

The EUSART module has the capability of sending the special Break character sequences that are required by the LIN/J2602 bus standard. The Break character transmit consists of a Start bit, followed by twelve '0' bits and a Stop bit. The Frame Break character is sent whenever the SENDB and TXEN bits (TXSTA<3> and TXSTA<5>) are set while the Transmit Shift register is loaded with data. Note that the value of data written to TXREG will be ignored and all '0's will be transmitted.

The SENDB bit is automatically reset by hardware after the corresponding Stop bit is sent. This allows the user to preload the transmit FIFO with the next transmit byte following the Break character (typically, the Sync character in the LIN/J2602 specification).

Note that the data value written to the TXREG for the Break character is ignored. The write simply serves the purpose of initiating the proper sequence.

The TRMT bit indicates when the transmit operation is active or Idle, just as it does during normal transmission. See Figure 19-10 for the timing of the Break character sequence.

19.2.5.1 Break and Sync Transmit Sequence

The following sequence will send a message frame header made up of a Break, followed by an Auto-Baud Sync byte. This sequence is typical of a LIN/J2602 bus master.

- 1. Configure the EUSART for the desired mode.
- 2. Set the TXEN and SENDB bits to set up the Break character.
- 3. Load the TXREG with a dummy character to initiate transmission (the value is ignored).
- 4. Write '55h' to TXREG to load the Sync character into the transmit FIFO buffer.
- 5. After the Break has been sent, the SENDB bit is reset by hardware. The Sync character now transmits in the preconfigured mode.

When the TXREG becomes empty, as indicated by the TXIF, the next data byte can be written to TXREG.

19.2.6 RECEIVING A BREAK CHARACTER

The Enhanced USART module can receive a Break character in two ways.

The first method forces configuration of the baud rate at a frequency of 9/13 the typical speed. This allows for the Stop bit transition to be at the correct sampling location (13 bits for Break versus Start bit and 8 data bits for typical data).

The second method uses the auto-wake-up feature described in **Section 19.2.4 "Auto-Wake-up on Sync Break Character"**. By enabling this feature, the EUSART will sample the next two transitions on RX/DT, cause an RCIF interrupt and receive the next data byte followed by another interrupt.

Note that following a Break character, the user will typically want to enable the Auto-Baud Rate Detect feature. For both methods, the user can set the ABD bit once the TXIF interrupt is observed.

FIGURE 19-10: SEND BREAK CHARACTER SEQUENCE



20.1 A/D Acquisition Requirements

For the A/D converter to meet its specified accuracy, the charge holding capacitor (CHOLD) must be allowed to fully charge to the input channel voltage level. The analog input model is shown in Figure 20-3. The source impedance (Rs) and the internal sampling switch (Rss) impedance directly affect the time required to charge the capacitor CHOLD. The sampling switch (Rss) impedance varies over the device voltage (VDD). The source impedance affects the offset voltage at the analog input (due to pin leakage current). The maximum recommended impedance for analog sources is 2.5 k Ω . After the analog input channel is selected (changed), the channel must be sampled for at least the minimum acquisition time before starting a conversion.

Note:	When	the	conversion	is	started,	the
	holding) capa	acitor is disco	nne	ected from	the
	input p	in.				

To calculate the minimum acquisition time, Equation 20-1 may be used. This equation assumes that 1/2 LSb error is used (1024 steps for the A/D). The 1/2 LSb error is the maximum error allowed for the A/D to meet its specified resolution.

Example 20-3 shows the calculation of the minimum required acquisition time TACQ. This calculation is based on the following application system assumptions:

Chold	=	25 pF
Rs	=	2.5 kΩ
Conversion Error	\leq	1/2 LSb
Vdd	=	$5V \rightarrow Rss = 2 k\Omega$
Temperature	=	85°C (system max.)

EQUATION 20-1: ACQUISITION TIME

TACQ	=	Amplifier Settling Time + Holding Capacitor Charging Time + Temperature Coefficient
	=	TAMP + TC + TCOFF

EQUATION 20-2: A/D MINIMUM CHARGING TIME

VHOLD	=	$(\text{VREF} - (\text{VREF}/2048)) \bullet (1 - e^{(-\text{TC/CHOLD}(\text{RIC} + \text{RSS} + \text{RS}))})$
or		
TC	=	-(CHOLD)(RIC + RSS + RS) $\ln(1/2048)$

EQUATION 20-3: CALCULATING THE MINIMUM REQUIRED ACQUISITION TIME

TACQ	=	TAMP + TC + TCOFF
TAMP	=	0.2 μs
TCOFF	=	(Temp – 25°C)(0.02 μs/°C) (85°C – 25°C)(0.02 μs/°C) 1.2 μs
Tempera	ture c	oefficient is only required for temperatures $> 25^{\circ}$ C. Below 25° C, TCOFF = 0 ms.
ТС	=	-(CHOLD)(RIC + RSS + RS) $\ln(1/2047)$ -(25 pF) (1 k Ω + 2 k Ω + 2.5 k Ω) $\ln(0.0004883)$ 1.05 µs
TACQ	=	0.2 μs + 1 μs + 1.2 μs 2.4 μs

25.0 INSTRUCTION SET SUMMARY

PIC18F2221/2321/4221/4321 family devices incorporate the standard set of 75 PIC18 core instructions, as well as an extended set of 8 new instructions for the optimization of code that is recursive or that utilizes a software stack. The extended set is discussed later in this section.

25.1 Standard Instruction Set

The standard PIC18 instruction set adds many enhancements to the previous PIC[®] MCU instruction sets, while maintaining an easy migration from these PIC MCU instruction sets. Most instructions are a single program memory word (16 bits), but there are four instructions that require two program memory locations.

Each single-word instruction is a 16-bit word divided into an opcode, which specifies the instruction type and one or more operands, which further specify the operation of the instruction.

The instruction set is highly orthogonal and is grouped into four basic categories:

- · Byte-oriented operations
- · Bit-oriented operations
- · Literal operations
- · Control operations

The PIC18 instruction set summary in Table 25-2 lists **byte-oriented**, **bit-oriented**, **literal** and **control** operations. Table 25-1 shows the opcode field descriptions.

Most byte-oriented instructions have three operands:

- 1. The file register (specified by 'f')
- 2. The destination of the result (specified by 'd')
- 3. The accessed memory (specified by 'a')

The file register designator 'f' specifies which file register is to be used by the instruction. The destination designator 'd' specifies where the result of the operation is to be placed. If 'd' is zero, the result is placed in the WREG register. If 'd' is one, the result is placed in the file register specified in the instruction.

All bit-oriented instructions have three operands:

- 1. The file register (specified by 'f')
- 2. The bit in the file register (specified by 'b')
- 3. The accessed memory (specified by 'a')

The bit field designator 'b' selects the number of the bit affected by the operation, while the file register designator 'f' represents the number of the file in which the bit is located.

The **literal** instructions may use some of the following operands:

- A literal value to be loaded into a file register (specified by 'k')
- The desired FSR register to load the literal value into (specified by 'f')
- No operand required (specified by '—')

The **control** instructions may use some of the following operands:

- A program memory address (specified by 'n')
- The mode of the CALL or RETURN instructions (specified by 's')
- The mode of the table read and table write instructions (specified by 'm')
- No operand required (specified by '—')

All instructions are a single word, except for four double-word instructions. These instructions were made double-word to contain the required information in 32 bits. In the second word, the 4 MSbs are '1's. If this second word is executed as an instruction (by itself), it will execute as a NOP.

All single-word instructions are executed in a single instruction cycle, unless a conditional test is true or the program counter is changed as a result of the instruction. In these cases, the execution takes two instruction cycles, with the additional instruction cycle(s) executed as a NOP.

The double-word instructions execute in two instruction cycles.

One instruction cycle consists of four oscillator periods. Thus, for an oscillator frequency of 4 MHz, the normal instruction execution time is 1 μ s. If a conditional test is true, or the program counter is changed as a result of an instruction, the instruction execution time is 2 μ s. Two-word branch instructions (if true) would take 3 μ s.

Figure 25-1 shows the general formats that the instructions can have. All examples use the convention 'nnh' to represent a hexadecimal number.

The Instruction Set Summary, shown in Table 25-2, lists the standard instructions recognized by the Microchip MPASM[™] Assembler.

Section 25.1.1 "Standard Instruction Set" provides a description of each instruction.

ΒZ		Branch if	Branch if Zero							
Synta	ax:	BZ n	BZ n							
Oper	ands:	-128 ≤ n ≤ 1	$-128 \le n \le 127$							
Oper	ation:	lf Zero bit is (PC) + 2 +	If Zero bit is '1', (PC) + 2 + 2n \rightarrow PC							
Statu	s Affected:	None	None							
Enco	ding:	1110	0000	nnn	n	nnnn				
Desc	ription:	If the Zero will branch. The 2's cor added to th have increr instruction, PC + 2 + 2t two-cycle ir	If the Zero bit is '1', then the program will branch. The 2's complement number '2n' is added to the PC. Since the PC will have incremented to fetch the next instruction, the new address will be PC + 2 + 2n. This instruction is then a two-cycle instruction							
Word	ls:	1	1							
Cycle	es:	1(2)	1(2)							
Q C If Ju	ycle Activity:									
	Q1	Q2	Q3 Process Data		Q4					
	Decode	Read literal 'n'			Write to PC	/rite to PC				
	No operation	No operation	No operat	tion	ор	No eration				
lf No	o Jump:									
i	Q1	Q2	Q3			Q4				
	Decode	Read literal 'n'	Proce Data	ess a	ор	No eration				
Example:		HERE	BZ	Jump						
PC After Instruction		tion = ad on	dress (I	HERE)						
	If Zero PC If Zero	= 1; = ad = 0;	dress (Jump)						
	PC	= ad	dress (1	HERE	+ 2)				

CALL	Subroutir	Subroutine Call					
Syntax:	CALL k {,s	5}					
Operands:	0 ≤ k ≤ 104 s ∈ [0,1]	8575					
Operation:	$\begin{array}{l} (PC) + 4 \rightarrow \\ k \rightarrow PC < 20 \\ \text{if s = 1,} \\ (W) \rightarrow WS, \\ (STATUS) \rightarrow \\ (BSR) \rightarrow B \end{array}$	TOS,):1>; → STATU SRS	JSS,				
Status Affected:	None						
Encoding: 1st word (k<7:0>) 2nd word(k<19:8>)	1110 1111	110s k ₁₉ kkk	k ₇ kkk kkkk	kkkk ₀ kkkk ₈			
	(PC + 4) is stack. If 's' BSR register respective s STATUSS a update occ 20-bit value CALL is a t	memory range. First, return address (PC + 4) is pushed onto the return stack. If 's' = 1, the W, STATUS and BSR registers are also pushed into their respective shadow registers, WS, STATUSS and BSRS. If 's' = 0, no update occurs (default). Then, the 20-bit value 'k' is loaded into PC<20:1>.					
Words:	2	2					
Cycles:	2						
Q Cycle Activity:							
Q1	Q2	Q3		Q4			
Decode	Read literal 'k'<7:0>,	PUSH F stac	PCtoR k '	ead literal k'<19:8>, /rite to PC			
No operation	No operation	No opera	tion	No operation			
Example:	HERE	CALL	THERE,	1			
Before Instruct	tion						
PC After Instructio	= address	G (HERE)				
TOS TOS WS BSRS STATUSS	 address address W BSR STATUS 	G (THER G (HERE	些) + 4)				

MUI	LLW	Multiply I	Literal v	ith W		MULV	
Synt	ax:	MULLW	k			Syntax	
Operands:		$0 \le k \le 255$	5			Operar	
Ope	ration:	(W) x k \rightarrow	PRODH:	PRODL			
Statu	us Affected:	None				Operat	
Enco	oding:	0000	1101	kkkk	kkkk	Status	
Description:		An unsigne out betwee 8-bit literal placed in th pair. PROE W is uncha None of the Note that n possible in is possible in	An unsigned multiplication is carried out between the contents of W and the 8-bit literal 'k'. The 16-bit result is placed in the PRODH:PRODL register pair. PRODH contains the high byte. W is unchanged. None of the Status flags are affected. Note that neither Overflow nor Carry is possible in this operation. A Zero result				
Wor	ds:	1					
Cycl	es:	1					
QC	cycle Activity:						
	Q1	Q2	Q3		Q4		
	Decode	Read literal 'k'	Proce Data	SS a i	Write registers PRODH: PRODL		
<u>Exar</u>	<u>mple:</u>	MULLW	0C4h				
	Before Instruct					Words	
	PRODH PRODL After Instructio	= E2 = ? = ? n	2n			Cycles Q Cyc	
	W PRODH PRODL	= E2 = A[= 08	2h Dh ¦h				

MULWF	Multiply	W with f	
Syntax:	MULWF	f {,a}	
Operands:	0 ≤ f ≤ 255 a ∈ [0,1]	5	
Operation:	(W) x (f) –	> PRODH:PR	ODL
Status Affected:	None		
Encoding:	0000	001a ff	ff ffff
Description:	An unsign out betwee register file result is st register pa high byte. unchange None of th Note that r possible ir result is po If 'a' is '0', selected. I to select th If 'a' is '0' a set is enat operates in Addressin $f \le 95$ (5FH "Byte-Orie Instructio Mode" for	ed multiplicati en the content e location 'f'.] ored in the PF air. PRODH co Both W and 'f d. e Status flags neither Overflo n this operatio ossible but no the Access B f 'a' is '1', the ne GPR bank and the extend oled, this instr n Indexed Lite g mode when n). See Sectio ented and Bit ns in Indexed details.	on is carried s of W and the The 16-bit RODH:PRODL ontains the f' are are affected. ow nor Carry is n. A Zero t detected. ank is BSR is used (default). ded instruction uction eral Offset ever on 25.2.3 -Oriented Literal Offset
Words:	1		
Cycles:	1		
Q Cycle Activity:			
Q1	Q2	Q3	Q4
Decode	Read register 'f'	Process Data	Write registers PRODH: PRODL
Example:	MILLINE	DEC 1	
Before Instruct	ion	1.11G , 1	
W REG PRODH PRODL After Instructio	= C4 = B5 = ? = ? n	lh ih	
vv REG PRODH PRODL	= C4 = B5 = 8A = 94	ih h h	

NEGF	Negate f
Syntax:	NEGF f {,a}
Operands:	0 ≤ f ≤ 255 a ∈ [0, 1]
Operation:	$(\overline{f}) + 1 \rightarrow f$
Status Affected:	N, OV, C, DC, Z
Encoding:	0110 110a ffff ffff
Description:	Location 'f' is negated using two's complement. The result is placed in the data memory location 'f'. If 'a' is '0', the Access Bank is selected. If 'a' is '1', the BSR is used to select the GPR bank (default). If 'a' is '0' and the extended instruction set is enabled, this instruction operates in Indexed Literal Offset Addressing mode whenever $f \le 95$ (5Fh). See Section 25.2.3 "Byte-Oriented and Bit-Oriented Instructions in Indexed Literal Offset Mode" for details.
Words:	1
Cycles:	1
Q Cycle Activity:	

NOP			No Operation				
Syntax:			NOP				
Oper	ands:	١	None				
Oper	ation:	١	No operatio	on			
Statu	s Affected:	١	None				
Encoding:			0000	0000	000	0	0000
			1111	XXXX	XXX	xx	XXXX
Description:		١	No operation.				
Words:		1	1				
Cycles:		1	1				
QC	ycle Activity:						
	Q1		Q2	Q	3		Q4
	Decode		No	No)		No
		C	peration	opera	tion	op	peration

Example:

None.

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

Example: NEGF REG, 1

> Before Instruction REG = 0011 1010 [3Ah] After Instruction REG = 1100 0110 [C6h]

© 2009 Microchip Technology Inc.

44-Lead Plastic Quad Flat, No Lead Package (ML) – 8x8 mm Body [QFN]

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







		MILLIMETERS	;	
Dimension	n Limits	MIN	NOM	MAX
Number of Pins	Ν		44	
Pitch	е		0.65 BSC	
Overall Height	Α	0.80	0.90	1.00
Standoff	A1	0.00	0.02	0.05
Contact Thickness	A3		0.20 REF	
Overall Width	Е		8.00 BSC	
Exposed Pad Width	E2	6.30	6.45	6.80
Overall Length	D		8.00 BSC	
Exposed Pad Length	D2	6.30	6.45	6.80
Contact Width	b	0.25	0.30	0.38
Contact Length	L	0.30	0.40	0.50
Contact-to-Exposed Pad	K	0.20	_	_

Notes:

1. Pin 1 visual index feature may vary, but must be located within the hatched area.

2. Package is saw singulated.

3. Dimensioning and tolerancing per ASME Y14.5M.

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

REF: Reference Dimension, usually without tolerance, for information purposes only.

Microchip Technology Drawing C04-103B

Example Frequencies/Resolutions	151
Operation Setup	
Period	150
TMR2 to PR2 Match	150, 155
PWM (ECCP Module)	
CCPR1H:CCPR1L Registers	
Duty Cycle	
Effects of a Reset	
Enhanced PWM Auto-Shutdown	
Example Frequencies/Resolutions	
Full-Bridge Application Example	
Full-Bridge Mode	
Direction Change	
Half-Bridge Mode	
Half-Bridge Output Mode Applications	
Example	
Operation in Power-Managed Modes	
Operation with Fail-Safe Clock Monitor	
Output Configurations	
Output Relationships (Active-High)	
Output Relationships (Active-Low)	
Period	
Programmable Dead-Band Delay	
Setup for PWM Operation	
Start-up Considerations	
Q	
	151 156
	101. 100

R

ĸ	
RAM. See Data Memory.	
RC Oscillator	
RCIO Oscillator Mode	31
RC_IDLE Mode	
RC_RUN Mode	41
RCALL	309
RCON Register	
Bit Status During Initialization	54
Reader Response	400
Register File	67
Register File Summary	69–71
Registers	
ADCON0 (A/D Control 0)	233
ADCON1 (A/D Control 1)	234
ADCON2 (A/D Control 2)	235
BAUDCON (Baud Rate Control)	214
CCP1CON (Enhanced Capture/Compare/PWM	
Control 1)	153
CCPxCON (CCPx Control)	145
CMCON (Comparator Control)	243
CONFIG1H (Configuration 1 High)	260
CONFIG2H (Configuration 2 High)	262
CONFIG2L (Configuration 2 Low)	261
CONFIG3H (Configuration 3 High)	263
CONFIG4L (Configuration 4 Low)	264
CONFIG5H (Configuration 5 High)	265
CONFIG5L (Configuration 5 Low)	265
CONFIG6H (Configuration 6 High)	266
CONFIG6L (Configuration 6 Low)	266
CONFIG7H (Configuration 7 High)	267
CONFIG7L (Configuration 7 Low)	267
CVRCON (Comparator Voltage	
Reference Control)	249
DEVID1 (Device ID 1)	268
DEVID2 (Device ID 2)	268
ECCP1AS (ECCP Auto-Shutdown Control)	163
. ,	

ECCP1DEL (PWM Dead-Band Delay) 162
EECON1 (Data EEPROM Control 1) 81, 90
HLVDCON (High/Low-Voltage Detect Control) 253
INTCON (Interrupt Control)
INTCON2 (Interrupt Control 2) 100
INTCON3 (Interrupt Control 3)
IPR1 (Peripheral Interrupt Priority 1)
IPR2 (Peripheral Interrupt Priority 2) 107
OSCCON (Oscillator Control)
OSCEUNE (Oscillator Curition)
DIE1 (Device and Interrupt Enchical)
PIE2 (Peripheral Interrupt Enable 2)
PIR1 (Peripheral Interrupt Request (Flag) 1) 102
PIR2 (Peripheral Interrupt Request (Flag) 2) 103
RCON (Reset Control) 48, 108
RCSTA (Receive Status and Control) 213
SSPADD(MSSP Address) 180
SSPCON1 (MSSP Control 1, I ² C Mode) 178
SSPCON1 (MSSP Control 1, SPI Mode)
SSPCON2 (MSSP Control 2 I^2 C Mode) 179
SSPSTAT (MSSP Status I^2 C Mode) 177
SSPSTAT (MSSP Status, FC Mode)
STATUS
STRPTR (Stack Pointer)
TUCON (Timeru Control)
11CON (Timer1 Control)
T2CON (Timer2 Control) 139
T3CON (Timer3 Control) 141
TRISE (PORTE/PSP Control) 124
TXSTA (Transmit Status and Control) 212
WDTCON (Watchdog Timer Control) 270
RESET
Reset State of Registers
Resets
Brown-out Reset (BOR)
Oscillator Start-up Timer (OST) 259
Power-on Reset (POR) 259
Power-up Timer (PWRT) 250
RETER
REILW
RETURN
Return Address Stack
Associated Registers 60
Return Stack Pointer (STKPTR) 61
Revision History
RLCF
RLNCF
RRCF
RRNCF
0
3
SCK 167
SDI 167
SDO
SEC IDLE Mode
SEC_RUN Mode

 Slave Select (SS)
 167

 SLEEP
 314

OSC1 and OSC2 Pin States 38

Serial Peripheral Interface. See SPI Mode.

Single-Supply ICSP Programming.

Sleep

Software Simulator (MPLAB SIM)	330
Special Event Trigger, See Compare (CCP Mode)	
Special Event Trigger. See Compare (CCI Mode).	`
Special Event Higger. See Compare (ECCF Module)). 250
Special Function Degisters	
SPI Mode (MSSP)	
Associated Registers	175
Bus Mode Compatibility	175
Effects of a Reset	175
	175
Master Mode	171
Master/Slave Connection	172
Operation	170
Operation in Power-Managed Modes	175
Serial Clock	167
Serial Data In	167
Serial Data Out	167
Slave Mode	
Slave Select	167
Slave Select Synchronization	173
SPI Clock	172
Typical Connection	171
SS	167
SSPOV	201
SSPOV Status Flag	201
SSPSTAT Register	
R/W Bit	181, 183
Stack Full/Underflow Resets	
SUBFSR	325
SUBFWB	314
SUBLW	315
SUBULNK	325
SUBWF	315
SUBWFB	316
SWAPF	316

т

Table Reads/Table Writes	62
TBLRD	317
TBLWT	318
Time-out in Various Situations (table)	51
Timer0	129
Associated Registers	131
Operation	130
Överflow Interrupt	131
Prescaler	131
Prescaler Assignment (PSA Bit)	131
Prescaler Select (T0PS2:T0PS0 Bits)	131
Prescaler. See Prescaler, Timer0.	
Reads and Writes in 16-Bit Mode	130
Source Edge Select (T0SE Bit)	130
Source Select (T0CS Bit)	130
Switching Prescaler Assignment	131
Timer1	133
16-Bit Read/Write Mode	135
Associated Registers	137
Interrupt	136
Operation	134
Oscillator13	3, 135
Layout Considerations	136
Low-Power Option	135
Overflow Interrupt	133
Resetting, Using the CCP Special Event Trigger	136
Special Event Trigger (ECCP)	154
TMR1H Register	133
5	

TMR1L Register	133
Use as a Real-Time Clock	136
Timer2	139
Associated Registers	140
Interrupt	140
Operation	139
Output	140
PR2 Register 15	0 155
TMP2 to PP2 Match Interrupt	155
TMP2 to PP2 Match Interrupt	150
	150
	141
16-Bit Read/Write Mode	143
Associated Registers	143
Operation	142
Oscillator 14	1, 143
Overflow Interrupt 14	1, 143
Special Event Trigger (CCP)	143
TMR3H Register	141
TMR3L Register	141
Timing Diagrams	
A/D Conversion	371
Acknowledge Seguence	204
Asynchronous Reception	225
Asynchronous Transmission	222
Asynchronous Transmission (Back to Back)	222
Automatic Baud Bate Calculation	220
Auto Wake up Bit (WHE) During	220
Normal Operation	226
	220
Auto-Wake-up Bit (WOE) During Sleep	100
Bauu Rale Generator with Clock Arbitration	190
BRG Overnow Sequence	220
BRG Reset Due to SDA Arbitration During	
	207
Brown-out Reset (BOR)	357
Bus Collision During a Repeated Start	
Condition (Case 1)	208
Bus Collision During a Repeated Start	
Condition (Case 2)	208
Bus Collision During a Start Condition	
(SCL = 0)	207
Bus Collision During a Stop Condition (Case 1)	209
Bus Collision During a Stop Condition (Case 2)	209
Bus Collision During Start Condition	
(SDA Only)	206
Bus Collision for Transmit and Acknowledge	205
Capture/Compare/PWM (All CCP Modules)	359
CLKO and I/O	356
Clock Synchronization	191
Clock/Instruction Cycle	63
EUSART Synchronous Receive (Master/Slave)	369
EUSART Synchronous Transmission	
(Master/Slave)	369
Example SPI Master Mode (CKE = 0)	361
Example SPI Master Mode (CKE = 1)	362
Example SPI Slave Mode (CKE = 0)	363
Example SPI Slave Mode (CKE = 1)	364
External Clock (All Modes Except PLL)	354
Eail Safe Clock Monitor	004
Fail-Sale Cluck Wolfillor	100
FIISE State DIL HIIIIIY	199
	159
Half-Bridge PWW Output	158
High/Low-voltage Detect Characteristics	351
High-voltage Detect Operation (VDIRMAG = 1) .	256
	365
120 Due Chart/Otars Dite	000