

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

-XF

Product Status	Active
Core Processor	PIC
Core Size	16-Bit
Speed	32MHz
Connectivity	I ² C, IrDA, SPI, UART/USART, USB OTG
Peripherals	Brown-out Detect/Reset, LVD, POR, PWM, WDT
Number of I/O	19
Program Memory Size	64KB (22K x 24)
Program Memory Type	FLASH
EEPROM Size	-
RAM Size	8K x 8
Voltage - Supply (Vcc/Vdd)	2V ~ 3.6V
Data Converters	A/D 9x10b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 85°C (TA)
Mounting Type	Surface Mount
Package / Case	28-VQFN Exposed Pad
Supplier Device Package	28-QFN (6x6)
Purchase URL	https://www.e-xfl.com/product-detail/microchip-technology/pic24fj64gb002t-i-ml

Email: info@E-XFL.COM

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

2.6 External Oscillator Pins

Many microcontrollers have options for at least two oscillators: a high-frequency primary oscillator and a low-frequency secondary oscillator (refer to **Section 8.0 "Oscillator Configuration"** for details).

The oscillator circuit should be placed on the same side of the board as the device. Place the oscillator circuit close to the respective oscillator pins with no more than 0.5 inch (12 mm) between the circuit components and the pins. The load capacitors should be placed next to the oscillator itself, on the same side of the board.

Use a grounded copper pour around the oscillator circuit to isolate it from surrounding circuits. The grounded copper pour should be routed directly to the MCU ground. Do not run any signal traces or power traces inside the ground pour. Also, if using a two-sided board, avoid any traces on the other side of the board where the crystal is placed.

Layout suggestions are shown in Figure 2-4. In-line packages may be handled with a single-sided layout that completely encompasses the oscillator pins. With fine-pitch packages, it is not always possible to completely surround the pins and components. A suitable solution is to tie the broken guard sections to a mirrored ground layer. In all cases, the guard trace(s) must be returned to ground.

In planning the application's routing and I/O assignments, ensure that adjacent port pins and other signals in close proximity to the oscillator are benign (i.e., free of high frequencies, short rise and fall times and other similar noise).

For additional information and design guidance on oscillator circuits, please refer to these Microchip Application Notes, available at the corporate web site (www.microchip.com):

- AN826, "Crystal Oscillator Basics and Crystal Selection for rfPIC™ and PICmicro[®] Devices"
- AN849, "Basic PICmicro[®] Oscillator Design"
- AN943, "Practical PICmicro[®] Oscillator Analysis and Design"
- AN949, "Making Your Oscillator Work"

FIGURE 2-4: SUGGESTED PLACEMENT OF THE OSCILLATOR CIRCUIT



FIGURE 7-1: PIC24F INTERRUPT VECTOR TABLE

1	Reset – GOTO Instruction	000000h	
	Reset – GOTO Address	000002h	
	Reserved	000004h	
	Oscillator Fail Trap Vector	1	
	Address Error Trap Vector	-	
	Stack Error Tran Vector		
	Math Error Trap Vector	-	
	Reserved	-	
	Reserved	-	
	Reserved	-	
	Interrupt Vector 0	000014h	1
	Interrupt Vector 1		
		-	
		-	
		-	
	Interrupt Vector 52	00007Ch	
rity	Interrupt Vector 53	00007Eh	Interrupt Vector Table (IVT) ⁽¹⁾
Lio	Interrupt Vector 54	000080h	
<u>г</u>			
de		-	
ō			
ra	Interrupt Vector 116	0000FCh	
atu	Interrupt Vector 117	0000FFh	1
Ž	Reserved	000100h	
sing	Reserved	000102h	
sas	Reserved		
SCR	Oscillator Fail Trap Vector		
De	Address Error Trap Vector		
	Stack Error Trap Vector		
	Math Error Trap Vector		
	Reserved	1	
	Reserved	1 –	7
	Reserved		
	Interrupt Vector 0	000114h	
	Interrupt Vector 1	1	
	 	1	
			Alternate Interrupt Vector Table (AIVT) ⁽¹⁾
	Interrupt Vector 52	00017Ch	
	Interrupt Vector 53	00017Eh	
	Interrupt Vector 54	000180h	
	—	1	
		1	
	—	1 –	1
	Interrupt Vector 116	1	
*	Interrupt Vector 117	0001FEh	
	Start of Code	000200h	

TABLE 7-1: TRAP VECTOR DETAILS

Vector Number	IVT Address	AIVT Address	Trap Source
0	000004h	000104h	Reserved
1	000006h	000106h	Oscillator Failure
2	000008h	000108h	Address Error
3	00000Ah	00010Ah	Stack Error
4	00000Ch	00010Ch	Math Error
5	00000Eh	00010Eh	Reserved
6	000010h	000110h	Reserved
7	000012h	000112h	Reserved

11-0	11-0	R/W-0	R/\/_0	R///-0	R/M_0	R///-0	R/\\/_0
	<u> </u>	AD1IF	U1TXIF	U1RXIE	SPI1IF	SPF1IF	T3IF
bit 15		7.811	U TI/	U H U H	01111	0.111	bit 8
R/W-0	R/W-0	R/W-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0
T2IF	OC2IF	IC2IF	—	T1IF	OC1IF	IC1IF	INT0IF
bit 7							bit 0
Logondi							
R = Readable	e bit	W = Writable	bit	U = Unimplem	nented bit read	d as '0'	
-n = Value at	POR	'1' = Bit is set		'0' = Bit is clea	ared	x = Bit is unkr	iown
bit 15-14	Unimplemen	ted: Read as '	כי				
bit 13	AD1IF: A/D C	Conversion Cor	nplete Interrup	t Flag Status bit			
	1 = Interrupt r	request has oc request has no	curred				
bit 12	U1TXIF: UAR	RT1 Transmitter	Interrupt Flag	Status bit			
	1 = Interrupt r	request has oc	curred				
	0 = Interrupt r	request has no	occurred				
bit 11	1 = Interrupt r	RI1 Receiver li	nterrupt Flag S	tatus bit			
	0 = Interrupt r	request has no	occurred				
bit 10	SPI1IF: SPI1	Event Interrup	t Flag Status b	it			
	1 = Interrupt r	request has oc	curred				
bit 9	SPF1IF: SPI1	Equest has no I Fault Interrup	t Flag Status b	it			
Sit 0	1 = Interrupt r	request has oc	curred	it i			
	0 = Interrupt r	request has no	occurred				
bit 8	T3IF: Timer3	Interrupt Flag	Status bit				
	1 = Interrupt r 0 = Interrupt r	request has oc request has no	curred				
bit 7	T2IF: Timer2	Interrupt Flag	Status bit				
	1 = Interrupt r	equest has oc	curred				
h # 0		request has no	coccurred	nt Elen Otatua k	.:.		
DILO	1 = Interrupt r	request has oc	curred	pi riag Status i	אנ		
	0 = Interrupt r	request has no	occurred				
bit 5	IC2IF: Input C	Capture Chann	el 2 Interrupt F	lag Status bit			
	1 = Interrupt r	request has oc request has no	curred				
bit 4	Unimplemen	ted: Read as '	D'				
bit 3	T1IF: Timer1	Interrupt Flag	Status bit				
	1 = Interrupt r	request has oc	curred				
hit 2		request has no	coccurred	nt Elag Status k	t		
	1 = Interrupt r	request has oc	curred	pt Flag Status t	Л		
	0 = Interrupt r	request has no	occurred				
bit 1	IC1IF: Input C	Capture Chann	el 1 Interrupt F	lag Status bit			
	1 = Interrupt r 0 = Interrupt r	equest has oc	curred				
bit 0	INTOIF: Exter	nal Interrupt 0	Flag Status bit				
	1 = Interrupt r	equest has oc	curred				
	0 = Interrupt r	request has no	occurred				

REGISTER 7-5: IFS0: INTERRUPT FLAG STATUS REGISTER 0

REGISTER 7-20: IPC3: INTERRUPT PRIORITY CONTROL REGISTER 3

000 = Interrupt source is disabled

U1TXIP<2:0>: UART1 Transmitter Interrupt Priority bits 111 = Interrupt is Priority 7 (highest priority interrupt)

Unimplemented: Read as '0'

001 = Interrupt is Priority 1 000 = Interrupt source is disabled

U-0	U-0	U-0	U-0	U-0	U-0	U-0	U-0	
_	—	—	_	_	—	—	—	
bit 15							bit 8	
U-0	R/W-1	R/W-0	R/W-0	U-0	R/W-1	R/W-0	R/W-0	
	AD1IP2	AD1IP1	AD1IP0	—	U1TXIP2	U1TXIP1	U1TXIP0	
bit 7			•				bit 0	
							,	
Legend:								
R = Readal	ble bit	W = Writable	bit	U = Unimplemented bit, read as '0'				
-n = Value a	at POR	'1' = Bit is set		'0' = Bit is cleared x = Bit is unkr			nown	
							J	
bit 15-7	Unimplement	ted: Read as 'd)'					
bit 6-4	AD1IP<2:0>:	A/D Conversio	n Complete In	terrupt Priority I	oits			
	111 = Interrug	ot is Priority 7 (hiahest prioritv	interrupt)				
	•							
	•							
	•							
	001 = Interrupt is Priority 1							

bit 3

bit 2-0

© 2010 Microchip Technology Inc.

U-0	U-0	U-0	U-0	U-0	R/W-0	R/W-0	R/W-0	
—	—	—			USB1IP2	USB1IP1	USB1IP0	
bit 15		·			•	•	bit 8	
								
U-0	U-0	U-0	U-0	U-0	U-0	U-0	U-0	
_		—		—	—	—	—	
bit 7						•	bit 0	
Legend:								
R = Readable	e bit	W = Writable	W = Writable bit		U = Unimplemented bit, read as '0'			
-n = Value at	POR	'1' = Bit is set	l' = Bit is set		'0' = Bit is cleared		x = Bit is unknown	
bit 15-11	Unimplemen	ted: Read as '	0'					
bit 10-8	bit 10-8 USB1IP<2:0>: USB Interrupt Priority bits							
111 = Interrupt is Priority 7 (highest priority interrupt)								
	•							
	•							
	•							
001 = Interrupt is Priority 1 000 = Interrupt source is disabled								
bit 7-0	Unimplemen	ted: Read as '	0'					

REGISTER 7-34: IPC21: INTERRUPT PRIORITY CONTROL REGISTER 21

10.4.3.2 Output Mapping

In contrast to inputs, the outputs of the Peripheral Pin Select options are mapped on the basis of the pin. In this case, a control register associated with a particular pin dictates the peripheral output to be mapped. The RPORx registers are used to control output mapping. Each register contains up to two 5-bit fields, with each field being associated with one RPn pin (see Register 10-15 through Register 10-27). The value of the bit field corresponds to one of the peripherals and that peripheral's output is mapped to the pin (see Table 10-3).

Because of the mapping technique, the list of peripherals for output mapping also includes a null value of '000000'. This permits any given pin to remain disconnected from the output of any of the pin-selectable peripherals.

TABLE 10-3:	SELECTABLE OUTPUT SOURCES	(MAPS FUNCTION TO OUTPUT)

Output Function Number ⁽¹⁾	Function	Output Name
0	NULL ⁽²⁾	Null
1	C1OUT	Comparator 1 Output
2	C2OUT	Comparator 2 Output
3	U1TX	UART1 Transmit
4	U1RTS ⁽³⁾	UART1 Request To Send
5	U2TX	UART2 Transmit
6	U2RTS ⁽³⁾	UART2 Request To Send
7	SDO1	SPI1 Data Output
8	SCK1OUT	SPI1 Clock Output
9	SS1OUT	SPI1 Slave Select Output
10	SDO2	SPI2 Data Output
11	SCK2OUT	SPI2 Clock Output
12	SS2OUT	SPI2 Slave Select Output
18	OC1	Output Compare 1
19	OC2	Output Compare 2
20	OC3	Output Compare 3
21	OC4	Output Compare 4
22	OC5	Output Compare 5
23-28	(unused)	NC
29	CTPLS	CTMU Output Pulse
30	C3OUT	Comparator 3 Output
31	(unused)	NC

Note 1: Setting the RPORx register with the listed value assigns that output function to the associated RPn pin.

2: The NULL function is assigned to all RPn outputs at device Reset and disables the RPn output function.

3: IrDA[®] BCLK functionality uses this output.

10.4.3.3 Mapping Limitations

The control schema of the Peripheral Pin Select is extremely flexible. Other than systematic blocks that prevent signal contention caused by two physical pins being configured as the same functional input, or two functional outputs configured as the same pin, there are no hardware enforced lock outs. The flexibility extends to the point of allowing a single input to drive multiple peripherals or a single functional output to drive multiple output pins.

10.4.3.4 PPS Mapping Exceptions for PIC24FJ64GB0 Family Devices

Although the PPS registers allow for up to 32 remappable pins, not all of these are implemented in all devices. Exceptions and unimplemented RPn pins are listed in Table 10-4.

TABLE 10-4: REMAPPABLE PIN EXCEPTIONS FOR PIC24FJ64GB004 FAMILY DEVICES

Device Pin	RP Pins (I/O)			
Count	Total	Unimplemented		
28 Pins	15	RP12, RP16-RP25		
44 Pins	25	RP12		

10.4.4 CONTROLLING CONFIGURATION CHANGES

Because peripheral remapping can be changed during run time, some restrictions on peripheral remapping are needed to prevent accidental configuration changes. PIC24F devices include three features to prevent alterations to the peripheral map:

- Control register lock sequence
- · Continuous state monitoring
- Configuration bit remapping lock

10.4.4.1 Control Register Lock

Under normal operation, writes to the RPINRx and RPORx registers are not allowed. Attempted writes will appear to execute normally, but the contents of the registers will remain unchanged. To change these registers, they must be unlocked in hardware. The register lock is controlled by the IOLOCK bit (OSCCON<6>). Setting IOLOCK prevents writes to the control registers; clearing IOLOCK allows writes.

To set or clear IOLOCK, a specific command sequence must be executed:

- 1. Write 46h to OSCCON<7:0>.
- 2. Write 57h to OSCCON<7:0>.
- 3. Clear (or set) IOLOCK as a single operation.

Unlike the similar sequence with the oscillator's LOCK bit, IOLOCK remains in one state until changed. This allows all of the Peripheral Pin Selects to be configured with a single unlock sequence, followed by an update to all control registers, then locked with a second lock sequence.

10.4.4.2 Continuous State Monitoring

In addition to being protected from direct writes, the contents of the RPINRx and RPORx registers are constantly monitored in hardware by shadow registers. If an unexpected change in any of the registers occurs (such as cell disturbances caused by ESD or other external events), a Configuration Mismatch Reset will be triggered.

10.4.4.3 Configuration Bit Pin Select Lock

As an additional level of safety, the device can be configured to prevent more than one write session to the RPINRx and RPORx registers. The IOL1WAY (CW2<4>) Configuration bit blocks the IOLOCK bit from being cleared after it has been set once. If IOLOCK remains set, the register unlock procedure will not execute and the Peripheral Pin Select Control registers cannot be written to. The only way to clear the bit and re-enable peripheral remapping is to perform a device Reset.

In the default (unprogrammed) state, IOL1WAY is set, restricting users to one write session. Programming IOL1WAY allows users unlimited access (with the proper use of the unlock sequence) to the Peripheral Pin Select registers.

REGISTER 10-5: RPINR7: PERIPHERAL PIN SELECT INPUT REGISTER 7

U-0	U-0	U-0	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1
—	—	—	IC2R4	IC2R3	IC2R2	IC2R1	IC2R0
bit 15							bit 8

U-0	U-0	U-0	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1
—	—	—	IC1R4	IC1R3	IC1R2	IC1R1	IC1R0
bit 7							bit 0

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

bit 15-13	Unimplemented: Read as '0'
bit 12-8	IC2R<4:0>: Assign Input Capture 2 (IC2) to Corresponding RPn or RPIn Pin bits
bit 7-5	Unimplemented: Read as '0'
bit 4-0	IC1R<4:0>: Assign Input Capture 1 (IC1) to Corresponding RPn or RPIn Pin bits

REGISTER 10-6: RPINR8: PERIPHERAL PIN SELECT INPUT REGISTER 8

-n = Value at POR '1' = Bit is set			'0' = Bit is cleared x = Bit is unknown				
R = Readable	e bit	W = Writable b	bit	U = Unimplemented bit, read as '0'			
Legend:							
bit 7				•			bit 0
_	—	—	IC3R4	IC3R3	IC3R2	IC3R1	IC3R0
U-0	U-0	U-0	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1
bit 15				•			bit 8
_	—	—	IC4R4	IC4R3	IC4R2	IC4R1	IC4R0
U-0	U-0	U-0	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1

bit 15-13 **Unimplemented:** Read as '0'

bit 12-8 IC4R<4:0>: Assign Input Capture 4 (IC4) to Corresponding RPn or RPIn Pin bits

bit 7-5 Unimplemented: Read as '0'

bit 4-0 IC3R<4:0>: Assign Input Capture 3 (IC3) to Corresponding RPn or RPIn Pin bits

12.0 TIMER2/3 AND TIMER4/5

Note: This data sheet summarizes the features of this group of PIC24F devices. It is not intended to be a comprehensive reference source. For more information, refer to the "PIC24F Family Reference Manual", Section 14. "Timers" (DS39704).

The Timer2/3 and Timer4/5 modules are 32-bit timers, which can also be configured as four independent 16-bit timers with selectable operating modes.

As 32-bit timers, Timer2/3 and Timer4/5 can each operate in three modes:

- Two Independent 16-Bit Timers with all 16-Bit Operating modes (except Asynchronous Counter mode)
- Single 32-Bit Timer
- Single 32-Bit Synchronous Counter

They also support these features:

- Timer Gate Operation
- Selectable Prescaler Settings
- Timer Operation during Idle and Sleep modes
- · Interrupt on a 32-Bit Period Register Match
- ADC Event Trigger (Timer4/5 only)

Individually, all four of the 16-bit timers can function as synchronous timers or counters. They also offer the features listed above, except for the ADC event trigger; this is implemented only with Timer5. The operating modes and enabled features are determined by setting the appropriate bit(s) in the T2CON, T3CON, T4CON and T5CON registers. T2CON and T4CON are shown in generic form in Register 12-1; T3CON and T5CON are shown in Register 12-2.

For 32-bit timer/counter operation, Timer2 and Timer4 are the least significant word; Timer3 and Timer4 are the most significant word of the 32-bit timers.

Note:	For 32-bit operation, T3CON and T5CON							
	control bits are ignored. Only T2CON and							
	T4CON control bits are used for setup and							
	control. Timer2 and Timer4 clock and gate							
	inputs are utilized for the 32-bit timer							
	modules, but an interrupt is generated with							
	the Timer3 or Timer5 interrupt flags.							

To configure Timer2/3 or Timer4/5 for 32-bit operation:

- 1. Set the T32 bit (T2CON<3> or T4CON<3> = 1).
- 2. Select the prescaler ratio for Timer2 or Timer4 using the TCKPS<1:0> bits.
- 3. Set the Clock and Gating modes using the TCS and TGATE bits. If TCS is set to an external clock, RPINRx (TxCK) must be configured to an available RPn pin. See Section 10.4 "Peripheral Pin Select (PPS)" for more information.
- 4. Load the timer period value. PR3 (or PR5) will contain the most significant word of the value while PR2 (or PR4) contains the least significant word.
- If interrupts are required, set the interrupt enable bit, T3IE or T5IE. Use the priority bits, T3IP<2:0> or T5IP<2:0>, to set the interrupt priority. Note that while Timer2 or Timer4 controls the timer, the interrupt appears as a Timer3 or Timer5 interrupt.
- 6. Set the TON bit (= 1).

The timer value, at any point, is stored in the register pair, TMR3:TMR2 (or TMR5:TMR4). TMR3 (TMR5) always contains the most significant word of the count, while TMR2 (TMR4) contains the least significant word.

To configure any of the timers for individual 16-bit operation:

- Clear the T32 bit corresponding to that timer (T2CON<3> for Timer2 and Timer3 or T4CON<3> for Timer4 and Timer5).
- 2. Select the timer prescaler ratio using the TCKPS<1:0> bits.
- 3. Set the Clock and Gating modes using the TCS and TGATE bits. See Section 10.4 "Peripheral Pin Select (PPS)" for more information.
- 4. Load the timer period value into the PRx register.
- 5. If interrupts are required, set the interrupt enable bit, TxIE; use the priority bits, TxIP<2:0>, to set the interrupt priority.
- 6. Set the TON bit (TxCON<15> = 1).

REGISTER 12-2: TyCON: TIMER3 AND TIMER5 CONTROL REGISTER⁽³⁾

R/W-0	U-0	R/W-0	U-0	U-0	U-0	U-0	U-0
TON ⁽¹⁾		TSIDL ⁽¹⁾	—		—		—
bit 15							bit 8
U-0	R/W-0	R/W-0	R/W-0	U-0	U-0	R/W-0	U-0
_	TGATE ⁽¹⁾	TCKPS1 ⁽¹⁾	TCKPS0 ⁽¹⁾		_	TCS ^(1,2)	—
bit 7							bit 0

Legend:				
R = Read	lable bit	W = Writable bit	U = Unimplemented bit,	, read as '0'
-n = Value	e at POR	'1' = Bit is set	'0' = Bit is cleared	x = Bit is unknown
bit 15	TON: Tim	ery On bit ⁽¹⁾		
	1 = Starts	s 16-bit Timery		
	0 = Stop	s 16-bit Timery		
bit 14	Unimple	mented: Read as '0'		
bit 13	TSIDL: S	top in Idle Mode bit ⁽¹⁾		
	1 = Disco	ntinue module operation whe	en device enters Idle mode	
	0 = Conti	nue module operation in Idle	mode	
bit 12-7	Unimplei	mented: Read as '0'		
bit 6	TGATE:	Fimery Gated Time Accumula	ation Enable bit ⁽¹⁾	
	When TC	<u>S = 1:</u>		
		ignorea.		
	1 = Gate	<u>5 = 0.</u> d time accumulation enabler	4	
	0 = Gate	d time accumulation disable	d	
bit 5-4	TCKPS<	1:0>: Timery Input Clock Pre	escale Select bits ⁽¹⁾	
	11 = 1:25	6		
	10 = 1:64			
	01 = 1:8			
h:4 0 0	00 = 1.1	mented. Deed as (o)		
DIT 3-2			1 2)	
bit 1	TCS: Im	ery Clock Source Select bit		
	1 = Exter	nal clock from pin TyCK (on	the rising edge)	
hit 0		$\mathbf{n} = \mathbf{n} + $		
	Unimplei	nented: Read as 0		
Note 1:	When 32-bit operation; all	operation is enabled (T2CON timer functions are set throu	N<3> or T4CON<3> = 1), these igh T2CON and T4CON.	e bits have no effect on Timery
2:	If TCS = 1, F	RPINRx (TxCK) must be con	figured to an available RPn pir	n. See Section 10.4 "Peripheral

Pin Select (PPS)" for more information.

3: Changing the value of TyCON while the timer is running (TON = 1) causes the timer prescale counter to reset and is not recommended.

13.0 INPUT CAPTURE WITH DEDICATED TIMERS

Note: This data sheet summarizes the features of this group of PIC24F devices. It is not intended to be a comprehensive reference source. For more information, refer to the "PIC24F Family Reference Manual", "Input Capture Section 34. with Dedicated Timer" (DS39722).

Devices in the PIC24FJ64GB004 family all feature 5 independent input capture modules. Each of the modules offers a wide range of configuration and operating options for capturing external pulse events and generating interrupts.

Key features of the input capture module include:

- Hardware-configurable for 32-bit operation in all modes by cascading two adjacent modules
- · Synchronous and Trigger modes of output compare operation, with up to 20 user-selectable trigger/sync sources available
- A 4-level FIFO buffer for capturing and holding timer values for several events
- Configurable interrupt generation
- · Up to 6 clock sources available for each module, driving a separate internal 16-bit counter

The module is controlled through two registers: ICxCON1 (Register 13-1) and ICxCON2 (Register 13-2). A general block diagram of the module is shown in Figure 13-1.

13.1 General Operating Modes

13.1.1 SYNCHRONOUS AND TRIGGER MODES

By default, the input capture module operates in a free-running mode. The internal 16-bit counter, ICxTMR, counts up continuously, wrapping around from FFFFh to 0000h on each overflow, with its period synchronized to the selected external clock source. When a capture event occurs, the current 16-bit value of the internal counter is written to the FIFO buffer.

In Synchronous mode, the module begins capturing events on the ICx pin as soon as its selected clock source is enabled. Whenever an event occurs on the selected sync source, the internal counter is reset. In Trigger mode, the module waits for a Sync event from another internal module to occur before allowing the internal counter to run.

Standard, free-running operation is selected by setting the SYNCSEL bits to '00000' and clearing the ICTRIG bit (ICxCON2<7>). Synchronous and Trigger modes are selected any time the SYNCSEL bits are set to any value except '00000'. The ICTRIG bit selects either Synchronous or Trigger mode; setting the bit selects Trigger mode operation. In both modes, the SYNCSEL bits determine the sync/trigger source.

When the SYNCSEL bits are set to '00000' and ICTRIG is set, the module operates in Software Trigger mode. In this case, capture operations are started by manually setting the TRIGSTAT bit (ICxCON2<6>).



FIGURE 13-1: INPUT CAPTURE BLOCK DIAGRAM

14.3 Pulse-Width Modulation (PWM) Mode

In PWM mode, the output compare module can be configured for edge-aligned or center-aligned pulse waveform generation. All PWM operations are double-buffered (buffer registers are internal to the module and are not mapped into SFR space).

To configure the output compare module for edge-aligned PWM operation:

- 1. Configure the OCx output for one of the available Peripheral Pin Select pins.
- 2. Calculate the desired on-time and load it into the OCxR register.
- 3. Calculate the desired period and load it into the OCxRS register.
- Select the current OCx as the synchronization source by writing 0x1F to SYNCSEL<4:0> (OCxCON2<4:0>) and '0' to OCTRIG (OCxCON2<7>).

- 5. Select a clock source by writing to the OCTSEL2<2:0> (OCxCON1<12:10>) bits.
- Enable interrupts, if required, for the timer and output compare modules. The output compare interrupt is required for PWM Fault pin utilization.
- 7. Select the desired PWM mode in the OCM<2:0> (OCxCON1<2:0>) bits.
- 8. If a timer is selected as a clock source, set the TMRy prescale value and enable the time base by setting the TON (TxCON<15>) bit.
- Note: This peripheral contains input and output functions that may need to be configured by the Peripheral Pin Select. See Section 10.4 "Peripheral Pin Select (PPS)" for more information.



FIGURE 14-2: OUTPUT COMPARE BLOCK DIAGRAM (DOUBLE-BUFFERED, 16-BIT PWM MODE)

EQUATION 15-1: RELATIONSHIP BETWEEN DEVICE AND SPI CLOCK SPEED⁽¹⁾

FCY

FSCK = Primary Prescaler * Secondary Prescaler

Note 1: Based on FCY = FOSC/2; Doze mode and PLL are disabled.

TABLE 15-1: SAMPLE SCK FREQUENCIES^(1,2)

		Secondary Prescaler Settings					
	1:1	2:1	4:1	6:1	8:1		
Primary Prescaler Settings	1:1	Invalid	8000	4000	2667	2000	
	4:1	4000	2000	1000	667	500	
	16:1	1000	500	250	167	125	
64:1		250	125	63	42	31	
Fcy = 5 MHz							
Primary Prescaler Settings	1:1	5000	2500	1250	833	625	
	4:1	1250	625	313	208	156	
	16:1	313	156	78	52	39	
	64:1	78	39	20	13	10	

Note 1: Based on FCY = FOSC/2, Doze mode and PLL are disabled.

2: SCKx frequencies shown in kHz.

20.2 RTCC Module Registers

The RTCC module registers are organized as three categories:

- RTCC Control Registers
- RTCC Value Registers
- · Alarm Value Registers

20.2.1 REGISTER MAPPING

To limit the register interface, the RTCC Timer and Alarm Time registers are accessed through corresponding register pointers. The RTCC Value register window (RTCVALH and RTCVALL) uses the RTCPTR bits (RCFGCAL<9:8>) to select the desired Timer register pair (see Table 20-1).

By writing to the RTCVALH byte, the RTCC Pointer value, RTCPTR<1:0> bits, decrements by one until they reach '00'. After they reach '00', the MINUTES and SECONDS value will be accessible through RTCVALH and RTCVALL until the pointer value is manually changed.

TABLE 20-1: RTCVAL REGISTER MAPPING

	RTCC Value Register Window					
KICFIKSI.02	RTCVAL<15:8>	RTCVAL<7:0>				
00	MINUTES	SECONDS				
01	WEEKDAY	HOURS				
10	MONTH	DAY				
11		YEAR				

The Alarm Value register window (ALRMVALH and ALRMVALL) uses the ALRMPTR bits (ALCFGRPT<9:8>) to select the desired Alarm register pair (see Table 20-2).

By writing to the ALRMVALH byte, the Alarm Pointer value, ALRMPTR<1:0> bits, decrements by one until they reach '00'. Once they reach '00', the ALRMMIN and ALRMSEC value will be accessible through ALRMVALH and ALRMVALL until the pointer value is manually changed.

EXAMPLE 20-1: SETTING THE RTCWREN BIT

```
asm volatile("push w7");
asm volatile("push w8");
asm volatile("disi #5");
asm volatile("mov #0x55, w7");
asm volatile("mov w7, _NVMKEY");
asm volatile("mov w8, _NVMKEY");
asm volatile("mov w8, _NVMKEY");
asm volatile("bset _RCFGCAL, #13"); //set the RTCWREN bit
asm volatile("pop w8");
asm volatile("pop w7");
```

TABLE 20-2: ALRMVAL REGISTER MAPPING

ALRMPTR	Alarm Value Register Window					
<1:0>	ALRMVAL<15:8>	ALRMVAL<7:0>				
00	ALRMMIN	ALRMSEC				
01	ALRMWD	ALRMHR				
10	ALRMMNTH	ALRMDAY				
11	_					

Considering that the 16-bit core does not distinguish between 8-bit and 16-bit read operations, the user must be aware that when reading either the ALRMVALH or ALRMVALL bytes, the ALRMPTR<1:0> value will be decremented. The same applies to the RTCVALH or RTCVALL bytes with the RTCPTR<1:0> being decremented.

Note:	This only applies to read operations and
	not write operations.

20.2.2 WRITE LOCK

In order to perform a write to any of the RTCC Timer registers, the RTCWREN bit (RCFGCAL<13>) must be set (refer to Example 20-1).

Note: To avoid accidental writes to the timer, it is recommended that the RTCWREN bit (RCFGCAL<13>) is kept clear at any other time. For the RTCWREN bit to be set, there is only one instruction cycle time window allowed between the 55h/AA sequence and the setting of RTCWREN; therefore, it is recommended that code follow the procedure in Example 20-1.

20.2.3 SELECTING RTCC CLOCK SOURCE

The clock source for the RTCC module can be selected using the Flash Configuration bit, RTCOSC (CW4<5>). When the bit is set to '1', the Secondary Oscillator (SOSC) is used as the reference clock, and when the bit is '0', LPRC is used as the reference clock.

21.1.3 DATA SHIFT DIRECTION

The LENDIAN bit (CRCCON1<3>) is used to control the shift direction. By default, the CRC will shift data through the engine, MSb first. Setting LENDIAN (= 1) causes the CRC to shift data, LSb first. This setting allows better integration with various communication schemes and removes the overhead of reversing the bit order in software. Note that this only changes the direction of the data that is shifted into the engine. The result of the CRC calculation will still be a normal CRC result, not a reverse CRC result.

21.1.4 INTERRUPT OPERATION

The module generates an interrupt that is configurable by the user for either of two conditions.

If CRCISEL is '0', an interrupt is generated when the VWORD<4:0> bits make a transition from a value of '1' to '0'. If CRCISEL is '1', an interrupt will be generated after the CRC operation finishes and the module sets the CRCGO bit to '0'. Manually setting CRCGO to '0' will not generate an interrupt.

21.1.5 TYPICAL OPERATION

To use the module for a typical CRC calculation:

- 1. Set the CRCEN bit to enable the module.
- 2. Configure the module for the desired operation:
 - a) Program the desired polynomial using the CRCXORL and CRCXORH registers, and the PLEN<4:0> bits
 - b) Configure the data width and shift direction using the DWIDTH and LENDIAN bits
 - c) Select the desired interrupt mode using the CRCISEL bit
- Preload the FIFO by writing to the CRCDATL and CRCDATH registers until the CRCFUL bit is set or no data is left
- Clear old results by writing 00h to CRCWDATL and CRCWDATH. CRCWDAT can also be left unchanged to resume a previously halted calculation.
- 5. Set the CRCGO bit to start calculation.
- 6. Write remaining data into the FIFO as space becomes available.
- When the calculation completes, CRCGO is automatically cleared. An interrupt will be generated if CRCISEL = 1.
- 8. Read CRCWDATL and CRCWDATH for the result of the calculation.

21.2 Registers

There are eight registers associated with the module:

- CRCCON1
- CRCCON2
- CRCXORL
- CRCXORH
- CRCDATL
- CRCDATH
- CRCWDATL
- CRCWDATH

The CRCCON1 and CRCCON2 registers (Register 21-1 and Register 21-2) control the operation of the module and configure the various settings. The CRCXOR registers (Register 21-3 and Register 21-4) select the polynomial terms to be used in the CRC equation. The CRCDAT and CRCWDAT registers are each register pairs that serve as buffers for the double-word, input data and CRC processed output, respectively.

Assembly Mnemonic		Assembly Syntax	Description	# of Words	# of Cycles	Status Flags Affected
BTSS	BTSS	f,#bit4	Bit Test f, Skip if Set	1	1 (2 or 3)	None
	BTSS	Ws,#bit4	Bit Test Ws, Skip if Set	1	1 (2 or 3)	None
BTST	BTST	f,#bit4	Bit Test f	1	1	Z
	BTST.C	Ws,#bit4	Bit Test Ws to C	1	1	С
	BTST.Z	Ws,#bit4	Bit Test Ws to Z	1	1	Z
	BTST.C	Ws,Wb	Bit Test Ws <wb> to C</wb>	1	1	С
	BTST.Z	Ws,Wb	Bit Test Ws <wb> to Z</wb>	1	1	Z
BTSTS	BTSTS	f,#bit4	Bit Test then Set f	1	1	Z
	BTSTS.C	Ws,#bit4	Bit Test Ws to C, then Set	1	1	С
	BTSTS.Z	Ws,#bit4	Bit Test Ws to Z, then Set	1	1	Z
CALL	CALL	lit23	Call Subroutine	2	2	None
	CALL	Wn	Call Indirect Subroutine	1	2	None
CLR	CLR	f	f = 0x0000	1	1	None
	CLR	WREG	WREG = 0x0000	1	1	None
	CLR	Ws	Ws = 0x0000	1	1	None
CLRWDT	CLRWDT		Clear Watchdog Timer	1	1	WDTO, Sleep
COM	COM	f	f = f	1	1	N, Z
	COM	f,WREG	WREG = f	1	1	N, Z
	COM	Ws,Wd	$Wd = \overline{Ws}$	1	1	N, Z
CP	CP	f	Compare f with WREG	1	1	, C, DC, N, OV, Z
	CP	Wb,#lit5	Compare Wb with lit5	1	1	C, DC, N, OV, Z
	CP	Wb.Ws	Compare Wb with Ws (Wb – Ws)	1	1	C. DC. N. OV. Z
CP0	CP0	f	Compare f with 0x0000	1	1	C. DC. N. OV. Z
	CP0	Ws	Compare Ws with 0x0000	1	1	C, DC, N, OV, Z
CPB	CPB	f	Compare f with WREG, with Borrow	1	1	C, DC, N, OV, Z
	CPB	Wb,#lit5	Compare Wb with lit5, with Borrow	1	1	C, DC, N, OV, Z
	CPB	Wb,Ws	Compare Wb with Ws, with Borrow $(Wb - Ws - \overline{C})$	1	1	C, DC, N, OV, Z
CPSEQ	CPSEQ	Wb,Wn	Compare Wb with Wn, Skip if =	1	1 (2 or 3)	None
CPSGT	CPSGT	Wb,Wn	Compare Wb with Wn, Skip if >	1	1 (2 or 3)	None
CPSLT	CPSLT	Wb,Wn	Compare Wb with Wn, Skip if <	1	1 (2 or 3)	None
CPSNE	CPSNE	Wb,Wn	Compare Wb with Wn, Skip if ≠	1	1 (2 or 3)	None
DAW	DAW.B	Wn	Wn = Decimal Adjust Wn	1	1	С
DEC	DEC	f	f = f - 1	1	1	C, DC, N, OV, Z
	DEC	f,WREG	WREG = f – 1	1	1	C, DC, N, OV, Z
	DEC	Ws,Wd	Wd = Ws - 1	1	1	C, DC, N, OV, Z
DEC2	DEC2	f	f = f - 2	1	1	C, DC, N, OV, Z
	DEC2	f,WREG	WREG = f – 2	1	1	C, DC, N, OV, Z
	DEC2	Ws,Wd	Wd = Ws - 2	1	1	C, DC, N, OV, Z
DISI	DISI	#lit14	Disable Interrupts for k Instruction Cycles	1	1	None
DIV	DIV.SW	Wm,Wn	Signed 16/16-bit Integer Divide	1	18	N, Z, C, OV
	DIV.SD	Wm,Wn	Signed 32/16-bit Integer Divide	1	18	N, Z, C, OV
	DIV.UW	Wm,Wn	Unsigned 16/16-bit Integer Divide	1	18	N, Z, C, OV
	DIV.UD	Wm,Wn	Unsigned 32/16-bit Integer Divide	1	18	N, Z, C, OV
EXCH	EXCH	Wns,Wnd	Swap Wns with Wnd	1	1	None
FF1L	FF1L	Ws,Wnd	Find First One from Left (MSb) Side	1	1	С
FF1R	FF1R	Ws,Wnd	Find First One from Right (LSb) Side	1	1	С

TABLE 28-2: INSTRUCTION SET OVERVIEW (CONTINUED)

TABLE 29-22: ADC MODULE SPECIFICATIONS

AC CHARACTERISTICS		$\begin{array}{ll} \mbox{Standard Operating Conditions: } 2.0V \ to \ 3.6V \\ \mbox{(unless otherwise stated)} \\ \mbox{Operating temperature} & -40^{\circ}C \leq TA \leq +85^{\circ}C \ for \ Industrial \\ & -40^{\circ}C \leq TA \leq +125^{\circ}C \ for \ Extended \end{array}$					
Param No.	Symbol	Characteristic	Min.	Тур	Max.	Units	Conditions
			Device S	Supply			
AD01	AVDD	Module VDD Supply	Greater of VDD – 0.3 or 2.0	_	Lesser of VDD + 0.3 or 3.6	V	
AD02	AVss	Module Vss Supply	Vss – 0.3	—	Vss + 0.3	V	
	-		Reference	e Inputs	-		
AD05	VREFH	Reference Voltage High	AVss + 1.7		AVdd	V	
AD06	VREFL	Reference Voltage Low	AVss	—	AVDD - 1.7	V	
AD07	VREF	Absolute Reference Voltage	AVss – 0.3	—	AVDD + 0.3	V	
AD08	IVREF	Reference voltage input current	—	—	1.25	mA	(Note 3)
AD09	ZVREF	Reference input impedance	—	10k	_	Ω	(Note 4)
			Analog	Input			
AD10	VINH-VINL	Full-Scale Input Span	VREFL	—	VREFH	V	(Note 2)
AD11	Vin	Absolute Input Voltage	AVss - 0.3	—	AVDD + 0.3	V	
AD12	VINL	Absolute VINL Input Voltage	AVss – 0.3	_	AVDD/2	V	
AD13	—	Leakage Current	—	±0.001	±0.610	μA	VINL = AVSS = VREFL = 0V, AVDD = VREFH = $3V$, Source Impedance = $2.5 \text{ k}\Omega$
AD17	Rin	Recommended Impedance of Analog Voltage Source	—	—	2.5K	Ω	10-bit
		·	ADC Ac	curacy			
AD20b	NR	Resolution	—	10	—	bits	
AD21b	INL	Integral Nonlinearity	—	±1	<±2	LSb	VINL = AVSS = VREFL = 0V, AVDD = VREFH = 3V
AD22b	DNL	Differential Nonlinearity	—	±0.5	<±1.25	LSb	VINL = AVSS = VREFL = 0V, AVDD = VREFH = 3V
AD23b	GERR	Gain Error	_	±1	±3	LSb	VINL = AVSS = VREFL = 0V, AVDD = VREFH = 3V
AD24b	EOFF	Offset Error	—	±1	±2	LSb	VINL = AVSS = VREFL = 0V, AVDD = VREFH = 3V
AD25b	—	Monotonicity ⁽¹⁾	—	_	—	_	Guaranteed

Note 1: The ADC conversion result never decreases with an increase in the input voltage and has no missing codes.

2: Measurements taken with external VREF+ and VREF- are used as the ADC voltage reference.

3: External reference voltage applied to VREF+/- pins. IVREF is current during conversion at 3.3v, 25C. Parameter is for design guidance only and is not tested.

4: Impedance during sampling at 3.3, 25C. Parameter is for design guidance only and is not tested.

NOTES:

30.2 Package Details

The following sections give the technical details of the packages.

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

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



	Units			MILLIMETERS			
Dimens	MIN	NOM	MAX				
Number of Pins	Ν		28				
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	E		6.00 BSC				
Exposed Pad Width	E2	3.65	3.70	4.20			
Overall Length	D		6.00 BSC				
Exposed Pad Length	D2	3.65	3.70	4.20			
Contact Width	b	0.23	0.30	0.35			
Contact Length	L	0.50	0.55	0.70			
Contact-to-Exposed Pad	К	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 C04-105B

CPU

Arithmetic Logic Unit (ALU)	
Clocking Scheme	
Control Registers	
Core Registers	
Programmer's Model	25
CRC	
Registers	255
Typical Operation	
User Interface	254
Data	
Polynomial	254
CTMU	
Measuring Capacitance	275
Measuring Time	
Pulse Delay and Generation	
Customer Change Notification Service	
Customer Notification Service	
Customer Support	

D

Data Memory	
Address Space	33
Memory Map	33
Near Data Space	34
SFR Space	34
Software Stack	50
Space Organization and Alignment	34
DC Characteristics	
Comparator Specifications	318
Comparator Voltage Reference	318
I/O Pin Input Specifications	316
I/O Pin Output Specifications	317
Idle Current	311
Internal Voltage Regulator	318
Operating Current	309
Power-Down Base Current	313
Power-Down Peripheral Module Current (IPD)	314
Program Memory	317
Temperature and Voltage Specifications	308
Deep Sleep BOR (DSBOR)	67
Deep Sleep Watchdog Timer (DSWDT)	289
Development Support	293
DISVREG Pin	287
Doze Mode	125

Е

Electrical Characteristics	
Absolute Maximum Ratings)5
Thermal Conditions	70
V/F Graphs)6
Equations	
A/D Conversion Clock Period26	37
Baud Rate Reload Calculation 18	33
Calculating the PWM Period16	35
Calculation for Maximum PWM Resolution	35
Estimating USB Transceiver Current	
Consumption20)1
Relationship Between Device and SPI	
Clock Speed18	30
UART Baud Rate with BRGH = 0 19	90
UART Baud Rate with BRGH = 1 19	90
Errata	. 8
Examples	
Baud Rate Error Calculation (BRGH = 0) 19	90

F

Flash Configuration Words	
Flash Program Memory	55
and Table Instructions	55
Enhanced ICSP Operation	56
JTAG Operation	56
Programming Algorithm	58
RTSP Operation	56
Single-Word Programming	61

I

I/O Ports	
Analog Input Voltage Considerations 12	8
Analog Port Pins Configuration 12	8
Input Change Notification 12	9
Open-Drain Configuration 12	8
Parallel (PIO)	7
Peripheral Pin Select 12	9
Pull-ups and Pull-Downs	9
l ² C	
Clock Rates	3
Communicating as Master in a Single	
Master Environment 18	1
Reserved Addresses 18	3
Setting Baud Rate When Operating as	
Bus Master 18	3
Slave Address Masking 18	3
Input Capture	
32-Bit Mode 15	8
Operations	8
Synchronous and Trigger Modes	7
Input Capture with Dedicated Timers	7
Instruction Set	
Overview	9
Summary	7
Symbols Used in Opcode Descriptions	8
Instruction-Based Power-Saving Modes11	7
Deep Sleep 11	8
Idle	8
Sleep	7
Inter-Integrated Circuit. See I ² C	1
Internet Address	8
Interrupt Service Routine (ISR)	6
Interrupt Vector Table (IVT)	9
Interrupts	
and Reset Sequence	9
Control and Status Registers	2
Implemented Vectors 7	1
Setup and Service Procedures	6
Trap Vectors7	0
Vector Table	0
J	
JTAG Interface	1
М	
Microshin Internet Web Site	o
MPLAR ASM30 Assembler Linker Librarian	0
MPLAP Integrated Development Environment Software 20	4
MDI AR DM3 Device Programmer	5
MDLAB DEAL ICE In Circuit Emulator System	U F
WIF LAD REAL ICE III-CIICUIL EIIIUIALOI SYSLEIII	0

MPLINK Object Linker/MPLIB Object Librarian 294