

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	16-Bit
Speed	32MHz
Connectivity	I ² C, IrDA, LINbus, SPI, UART/USART
Peripherals	Brown-out Detect/Reset, LVD, POR, PWM, WDT
Number of I/O	53
Program Memory Size	64KB (22K x 24)
Program Memory Type	FLASH
EEPROM Size	·
RAM Size	16K x 8
Voltage - Supply (Vcc/Vdd)	2V ~ 3.6V
Data Converters	A/D 16x10b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 85°C (TA)
Mounting Type	Surface Mount
Package / Case	64-VFQFN Exposed Pad
Supplier Device Package	64-VQFN (9x9)
Purchase URL	https://www.e-xfl.com/product-detail/microchip-technology/pic24fj64ga106t-i-mr

Email: info@E-XFL.COM

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

1.0 DEVICE OVERVIEW

This document contains device-specific information for the following devices:

- PIC24FJ64GA106 PIC24FJ64GA110
- PIC24FJ128GA106 PIC24F
 - PIC24FJ128GA110
 PIC24FJ192GA110
- PIC24FJ192GA106PIC24FJ256GA106
- PIC24FJ256GA110
- PIC24FJ64GA108
- PIC24FJ128GA108
- PIC24FJ192GA108
- PIC24FJ256GA108

This family expands on the existing line of Microchip's 16-bit general purpose microcontrollers, combining enhanced computational performance with an expanded and highly configurable peripheral feature set. The PIC24FJ256GA110 family provides a new platform for high-performance applications, which have outgrown their 8-bit platforms, but don't require the power of a digital signal processor.

1.1 Core Features

1.1.1 16-BIT ARCHITECTURE

Central to all PIC24F devices is the 16-bit modified Harvard architecture, first introduced with Microchip's dsPIC[®] digital signal controllers. The PIC24F CPU core offers a wide range of enhancements, such as:

- 16-bit data and 24-bit address paths with the ability to move information between data and memory spaces
- Linear addressing of up to 12 Mbytes (program space) and 64 Kbytes (data)
- A 16-element working register array with built-in software stack support
- A 17 x 17 hardware multiplier with support for integer math
- Hardware support for 32 by 16-bit division
- An instruction set that supports multiple addressing modes and is optimized for high-level languages, such as 'C'
- · Operational performance up to 16 MIPS

1.1.2 POWER-SAVING TECHNOLOGY

All of the devices in the PIC24FJ256GA110 family incorporate a range of features that can significantly reduce power consumption during operation. Key items include:

• **On-the-Fly Clock Switching:** The device clock can be changed under software control to the Timer1 source or the internal, low-power RC Oscillator during operation, allowing the user to incorporate power-saving ideas into their software designs.

- **Doze Mode Operation:** When timing-sensitive applications, such as serial communications, require the uninterrupted operation of peripherals, the CPU clock speed can be selectively reduced, allowing incremental power savings without missing a beat.
- Instruction-Based Power-Saving Modes: The microcontroller can suspend all operations, or selectively shut down its core while leaving its peripherals active, with a single instruction in software.

1.1.3 OSCILLATOR OPTIONS AND FEATURES

All of the devices in the PIC24FJ256GA110 family offer five different oscillator options, allowing users a range of choices in developing application hardware. These include:

- Two Crystal modes using crystals or ceramic resonators.
- Two External Clock modes offering the option of a divide-by-2 clock output.
- A Fast Internal Oscillator (FRC) with a nominal 8 MHz output, which can also be divided under software control to provide clock speeds as low as 31 kHz.
- A Phase Lock Loop (PLL) frequency multiplier available to the external oscillator modes and the FRC Oscillator, which allows clock speeds of up to 32 MHz.
- A separate internal RC Oscillator (LPRC) with a fixed 31 kHz output, which provides a low-power option for timing-insensitive applications.

The internal oscillator block also provides a stable reference source for the Fail-Safe Clock Monitor. This option constantly monitors the main clock source against a reference signal provided by the internal oscillator and enables the controller to switch to the internal oscillator, allowing for continued low-speed operation or a safe application shutdown.

1.1.4 EASY MIGRATION

Regardless of the memory size, all devices share the same rich set of peripherals, allowing for a smooth migration path as applications grow and evolve. The consistent pinout scheme used throughout the entire family also aids in migrating from one device to the next larger, or even in jumping from 64-pin to 100-pin devices.

The PIC24F family is pin-compatible with devices in the dsPIC33 and PIC32 families, and shares some compatibility with the pinout schema for PIC18 and dsPIC30 devices. This extends the ability of applications to grow from the relatively simple, to the powerful and complex, yet still selecting a Microchip device.

|--|

Features	PIC24FJ64GA110	PIC24FJ128GA110	PIC24FJ192GA110	PIC24FJ256GA110		
Operating Frequency		DC – 3	32 MHz			
Program Memory (bytes)	64K	128K	192K	256K		
Program Memory (instructions)	22,016	44,032	67,072	87,552		
Data Memory (bytes)			384	·		
Interrupt Sources (soft vectors/NMI traps)		66 (62/4)			
I/O Ports		Ports A, B,	C, D, E, F, G			
Total I/O Pins		8	35			
Remappable Pins		46 (32 I/O, 1	14 input only)			
Timers:		5	(1)			
32 Bit (from paired 16 bit timers)		0	2			
Input Capture Channels		9	(1)			
Output Compare/PWM		9	(1)			
Channels		Ū				
Input Change Notification Interrupt	85					
Serial Communications:						
UART		4	(1)			
SPI (3-wire/4-wire)		3	(1)			
I ² C™		:	3			
Parallel Communications (PMP/PSP)		Y	es			
JTAG Boundary Scan		Y	/es			
10-Bit Analog-to-Digital Module (input channels)		1	16			
Analog Comparators			3			
CTMU Interface		Y	es			
Resets (and delays)	POR, B REPEAT Ins	OR, RESET Instruction struction, Hardware Tra (PWRT, OS	n, MCLR, WDT; Illegal aps, Configuration Wc T, PLL Lock)	Opcode, ord Mismatch		
Instruction Set	76 Bas	e Instructions, Multiple	e Addressing Mode Va	ariations		
Packages		100-Pi	n TQFP			

Note 1: Peripherals are accessible through remappable pins.

	Pin Number					
Function	64-Pin TQFP, QFN	80-Pin TQFP	100-Pin TQFP	I/O	Input Buffer	Description
RA0	_	-	17	I/O	ST	PORTA Digital I/O.
RA1	—	_	38	I/O	ST	
RA2	_	_	58	I/O	ST	
RA3	_	_	59	I/O	ST	
RA4	—	_	60	I/O	ST	
RA5	_	_	61	I/O	ST	
RA6	_	_	91	I/O	ST	
RA7	_		92	I/O	ST	
RA9	_	23	28	I/O	ST	
RA10	—	24	29	I/O	ST	
RA14	—	52	66	I/O	ST	
RA15	_	53	67	I/O	ST	
RB0	16	20	25	I/O	ST	PORTB Digital I/O.
RB1	15	19	24	I/O	ST	
RB2	14	18	23	I/O	ST	
RB3	13	17	22	I/O	ST	
RB4	12	16	21	I/O	ST	
RB5	11	15	20	I/O	ST	
RB6	17	21	26	I/O	ST	
RB7	18	22	27	I/O	ST	
RB8	21	27	32	I/O	ST	
RB9	22	28	33	I/O	ST	
RB10	23	29	34	I/O	ST	
RB11	24	30	35	I/O	ST	
RB12	27	33	41	I/O	ST	
RB13	28	34	42	I/O	ST	-
RB14	29	35	43	I/O	ST	
RB15	30	36	44	I/O	ST	
RC1	—	4	6	I/O	ST	PORTC Digital I/O.
RC2	—	_	7	I/O	ST	
RC3	—	5	8	I/O	ST	
RC4	_		9	I/O	ST	
RC12	39	49	63	I/O	ST	
RC13	47	59	73	I/O	ST	
RC14	48	60	74	I/O	ST	
RC15	40	50	64	I/O	ST	

ΤΔRI F 1-4·	PIC24EJ256GA110 FAMILY PINOUT DESCRIPTIONS (
IADLE 1-4.	FIC24FJ250GATTO FAMILI FINOUT DESCRIFTIONS	CONTINUED)

Legend: TTL = TTL input buffer ANA = Analog level input/output ST = Schmitt Trigger input buffer

I²C[™] = I²C/SMBus input buffer

2.2 Power Supply Pins

2.2.1 DECOUPLING CAPACITORS

The use of decoupling capacitors on every pair of power supply pins, such as VDD, VSS, AVDD and AVSS is required.

Consider the following criteria when using decoupling capacitors:

- Value and type of capacitor: A 0.1 μ F (100 nF), 10-20V capacitor is recommended. The capacitor should be a low-ESR device with a resonance frequency in the range of 200 MHz and higher. Ceramic capacitors are recommended.
- Placement on the printed circuit board: The decoupling capacitors should be placed as close to the pins as possible. It is recommended to place the capacitors on the same side of the board as the device. If space is constricted, the capacitor can be placed on another layer on the PCB using a via; however, ensure that the trace length from the pin to the capacitor is no greater than 0.25 inch (6 mm).
- Handling high-frequency noise: If the board is experiencing high-frequency noise (upward of tens of MHz), add a second ceramic type capacitor in parallel to the above described decoupling capacitor. The value of the second capacitor can be in the range of 0.01 μ F to 0.001 μ F. Place this second capacitor next to each primary decoupling capacitor. In high-speed circuit designs, consider implementing a decade pair of capacitances as close to the power and ground pins as possible (e.g., 0.1 μ F in parallel with 0.001 μ F).
- Maximizing performance: On the board layout from the power supply circuit, run the power and return traces to the decoupling capacitors first, and then to the device pins. This ensures that the decoupling capacitors are first in the power chain. Equally important is to keep the trace length between the capacitor and the power pins to a minimum, thereby reducing PCB trace inductance.

2.2.2 TANK CAPACITORS

On boards with power traces running longer than six inches in length, it is suggested to use a tank capacitor for integrated circuits including microcontrollers to supply a local power source. The value of the tank capacitor should be determined based on the trace resistance that connects the power supply source to the device, and the maximum current drawn by the device in the application. In other words, select the tank capacitor so that it meets the acceptable voltage sag at the device. Typical values range from 4.7 μ F to 47 μ F.

2.3 Master Clear (MCLR) Pin

The MCLR pin provides two specific device functions: device Reset, and device programming and debugging. If programming and debugging are not required in the end application, a direct connection to VDD may be all that is required. The addition of other components, to help increase the application's resistance to spurious Resets from voltage sags, may be beneficial. A typical configuration is shown in Figure 2-1. Other circuit designs may be implemented, depending on the application's requirements.

During programming and debugging, the resistance and capacitance that can be added to the pin must be considered. Device programmers and debuggers drive the $\overline{\text{MCLR}}$ pin. Consequently, specific voltage levels (VIH and VIL) and fast signal transitions must not be adversely affected. Therefore, specific values of R1 and C1 will need to be adjusted based on the application and PCB requirements. For example, it is recommended that the capacitor, C1, be isolated from the $\overline{\text{MCLR}}$ pin during programming and debugging operations by using a jumper (Figure 2-2). The jumper is replaced for normal run-time operations.

Any components associated with the $\overline{\text{MCLR}}$ pin should be placed within 0.25 inch (6 mm) of the pin.

FIGURE 2-2: EXAMPLE OF MCLR PIN CONNECTIONS



Overstress (EOS). Ensure that the MCLR pin

VIH and VIL specifications are met.



REGISTER 3-2: CORCON: CPU CONTROL REGISTER

U-0	U-0	U-0	U-0	U-0	U-0	U-0	U-0
	—	—	—	—	—	—	—
bit 15							bit 8
U-0	U-0	U-0	U-0	R/C-0	R/W-0	U-0	U-0

00	00	00	00	1000	1011 0	00	00
—	_		_	IPL3 ⁽¹⁾	PSV		_
bit 7							bit 0

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

bit 15-4 **Unimplemented:** Read as '0'

- bit 3IPL3: CPU Interrupt Priority Level Status bit⁽¹⁾1 = CPU interrupt priority level is greater than 70 = CPU interrupt priority level is 7 or lessbit 2PSV: Program Space Visibility in Data Space Enable bit1 = Program space visible in data space0 = Program space not visible in data spacebit 1-0Unimplemented: Read as '0'
- **Note 1:** User interrupts are disabled when IPL3 = 1.

EXAMPLE 5-2: ERASING A PROGRAM MEMORY BLOCK (C LANGUAGE CODE)

<pre>// C example using MPLAB C30 unsigned long progAddr = 0xXXXXXX; unsigned int offset;</pre>	// Address of row to write
//Set up pointer to the first memory locatio	n to be written
TBLPAG = progAddr>>16;	// Initialize PM Page Boundary SFR
offset = progAddr & 0xFFFF;	// Initialize lower word of address
<pre>builtin_tblwtl(offset, 0x0000);</pre>	<pre>// Set base address of erase block // with dummy latch write</pre>
NVMCON = $0 \times 4042;$	// Initialize NVMCON
asm("DISI #5");	<pre>// Block all interrupts with priority <7 // for next 5 instructions</pre>
builtin_write_NVM();	// C30 function to perform unlock
	// sequence and set WR

EXAMPLE 5-3: LOADING THE WRITE BUFFERS (ASSEMBLY LANGUAGE CODE)

;	Set up NVMCON	I for row programming operatior	ıs	
	MOV	#0x4001, W0	;	
	MOV	W0, NVMCON	;	Initialize NVMCON
;	Set up a poin	ter to the first program memor	ſУ	location to be written
;	program memor	ry selected, and writes enabled	ł	
	MOV	#0x0000, W0	;	
	MOV	W0, TBLPAG	;	Initialize PM Page Boundary SFR
	MOV	#0x6000, W0	;	An example program memory address
;	Perform the I	BLWT instructions to write the	e 1	latches
;	0th_program_w	ord		
	MOV	#LOW_WORD_0, W2	;	
	MOV	<pre>#HIGH_BYTE_0, W3</pre>	;	
	TBLWTL	W2, [W0]	;	Write PM low word into program latch
	TBLWTH	W3, [W0++]	;	Write PM high byte into program latch
;	lst_program_w	vord		
	MOV	#LOW_WORD_1, W2	;	
	MOV	#HIGH_BYTE_1, W3	;	Weite DM les and international let a
	TBLWTL	W2, [W0]	,	Write PM low word into program latch
	Jud program	W3, [W0++]	i	Write PM nigh byte into program latch
'	Znu_program_	HION NODD 2 N2		
	MOV	HLOW_WORD_2, W2	΄.	
	MOV TRIMTI	#niGn_Biik_2, W5		Write DM low word into program latch
	TBLWTH	W2, [W0] W3 [W0++]	;	Write DM high byte into program latch
	•		'	write in high byte into program raten
	•			
	•			
;	63rd program	word		
	MOV	#LOW_WORD_31, W2	;	
	MOV	#HIGH_BYTE_31, W3	;	
	TBLWTL	W2, [W0]	;	Write PM low word into program latch
	TBLWTH	W3, [W0]	;	Write PM high byte into program latch

7.4 Interrupt Setup Procedures

7.4.1 INITIALIZATION

To configure an interrupt source:

- 1. Set the NSTDIS control bit (INTCON1<15>) if nested interrupts are not desired.
- Select the user-assigned priority level for the interrupt source by writing the control bits in the appropriate IPCx register. The priority level will depend on the specific application and type of interrupt source. If multiple priority levels are not desired, the IPCx register control bits for all enabled interrupt sources may be programmed to the same non-zero value.

Note:	At a device	Rese	et, the	IPC	Cx regi	isters are
	initialized,	such	that	all	user	interrupt
	sources are	e assig	gned f	to pi	iority l	evel 4.

- 3. Clear the interrupt status flag bit associated with the peripheral in the associated IFSx register.
- 4. Enable the interrupt source by setting the interrupt enable control bit associated with the source in the appropriate IECx register.

7.4.2 INTERRUPT SERVICE ROUTINE

The method that is used to declare an ISR and initialize the IVT with the correct vector address will depend on the programming language (i.e., 'C' or assembler) and the language development toolsuite that is used to develop the application. In general, the user must clear the interrupt flag in the appropriate IFSx register for the source of the interrupt that the ISR handles. Otherwise, the ISR will be re-entered immediately after exiting the routine. If the ISR is coded in assembly language, it must be terminated using a RETFIE instruction to unstack the saved PC value, SRL value and old CPU priority level.

7.4.3 TRAP SERVICE ROUTINE

A Trap Service Routine (TSR) is coded like an ISR, except that the appropriate trap status flag in the INTCON1 register must be cleared to avoid re-entry into the TSR.

7.4.4 INTERRUPT DISABLE

All user interrupts can be disabled using the following procedure:

- 1. Push the current SR value onto the software stack using the PUSH instruction.
- 2. Force the CPU to priority level 7 by inclusive ORing the value E0h with SRL.

To enable user interrupts, the POP instruction may be used to restore the previous SR value.

Note that only user interrupts with a priority level of 7 or less can be disabled. Trap sources (level 8-15) cannot be disabled.

The DISI instruction provides a convenient way to disable interrupts of priority levels 1-6 for a fixed period of time. Level 7 interrupt sources are not disabled by the DISI instruction.

10.0 I/O PORTS

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 12. "I/O Ports with Peripheral
	Pin Select (PPS)" (DS39711).

All of the device pins (except VDD, VSS, MCLR and OSCI/CLKI) are shared between the peripherals and the parallel I/O ports. All I/O input ports feature Schmitt Trigger inputs for improved noise immunity.

10.1 Parallel I/O (PIO) Ports

A parallel I/O port that shares a pin with a peripheral is, in general, subservient to the peripheral. The peripheral's output buffer data and control signals are provided to a pair of multiplexers. The multiplexers select whether the peripheral or the associated port has ownership of the output data and control signals of the I/O pin. The logic also prevents "loop through", in which a port's digital output can drive the input of a peripheral that shares the same pin. Figure 10-1 shows how ports are shared with other peripherals and the associated I/O pin to which they are connected. When a peripheral is enabled and the peripheral is actively driving an associated pin, the use of the pin as a general purpose output pin is disabled. The I/O pin may be read, but the output driver for the parallel port bit will be disabled. If a peripheral is enabled, but the peripheral is not actively driving a pin, that pin may be driven by a port.

All port pins have three registers directly associated with their operation as digital I/O. The Data Direction register (TRIS) determines whether the pin is an input or an output. If the data direction bit is a '1', then the pin is an input. All port pins are defined as inputs after a Reset. Reads from the Output Latch register (LAT), read the latch. Writes to the latch, write the latch. Reads from the port (PORT), read the port pins, while writes to the port pins, write the latch.

Any bit and its associated data and control registers that are not valid for a particular device will be disabled. That means the corresponding LAT and TRIS registers and the port pin will read as zeros.

When a pin is shared with another peripheral or function that is defined as an input only, it is regarded as a dedicated port because there is no other competing source of outputs.



FIGURE 10-1: BLOCK DIAGRAM OF A TYPICAL SHARED PORT STRUCTURE

REGISTER 10-9: RPINR10: PERIPHERAL PIN SELECT INPUT REGISTER 10

U-0	U-0	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1
—	—	IC8R5	IC8R4	IC8R3	IC8R2	IC8R1	IC8R0
bit 15							bit 8
U-0	U-0	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1
—	—	IC7R5	IC7R4	IC7R3	IC7R2	IC7R1	IC7R0
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			nown	
							, ,

bit 15-14	Unimplemented: Read as '0'
bit 13-8	IC8R<5:0>: Assign Input Capture 8 (IC8) to Corresponding RPn or RPIn Pin bits
bit 7-6	Unimplemented: Read as '0'
bit 5-0	IC7R<5:0>: Assign Input Capture 7 (IC7) to Corresponding RPn or RPIn Pin bits

REGISTER 10-10: RPINR11: PERIPHERAL PIN SELECT INPUT REGISTER 11

U-0	U-0	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1
—	—	OCFBR5	OCFBR4	OCFBR3	OCFBR2	OCFBR1	OCFBR0
bit 15							bit 8

U-0	U-0	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1
_	—	OCFAR5	OCFAR4	OCFAR3	OCFAR2	OCFAR1	OCFAR0
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 15-14 Unimplemented: Read as '0'

bit 13-8 **OCFBR<5:0>:** Assign Output Compare Fault B (OCFB) to Corresponding RPn or RPIn Pin bits

bit 7-6 Unimplemented: Read as '0'

bit 5-0 OCFAR<5:0>: Assign Output Compare Fault A (OCFA) to Corresponding RPn or RPIn Pin bits

U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	—	RP5R5 ⁽¹⁾	RP5R4 ⁽¹⁾	RP5R3 ⁽¹⁾	RP5R2 ⁽¹⁾	RP5R1 ⁽¹⁾	RP5R0 ⁽¹⁾
bit 15							bit 8

U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	—	RP4R5	RP4R4	RP4R3	RP4R2	RP4R1	RP4R0
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 15-14	Unimplemented: Read as '0'
bit 13-8	RP5R<5:0>: RP5 Output Pin Mapping bits ⁽¹⁾

Peripheral output number n is assigned to pin, RP5 (see Table 10-3 for peripheral function numbers).bit 7-6 Unimplemented: Read as '0'

bit 5-0 **RP4R<5:0>:** RP4 Output Pin Mapping bits Peripheral output number n is assigned to pin, RP4 (see Table 10-3 for peripheral function numbers).

Note 1: Unimplemented in 64-pin devices; read as '0'.

REGISTER 10-25: RPOR3: PERIPHERAL PIN SELECT OUTPUT REGISTER 3

U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	—	RP7R5	RP7R4	RP7R3	RP7R2	RP7R1	RP7R0
bit 15							bit 8
U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
_	_	RP6R5	RP6R4	RP6R3	RP6R2	RP6R1	RP6R0

bit 7

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 15-14 Unimplemented: Read as '0'

bit 13-8 **RP7R<5:0>:** RP7 Output Pin Mapping bits

Peripheral output number n is assigned to pin, RP7 (see Table 10-3 for peripheral function numbers).

bit 7-6 Unimplemented: Read as '0'

bit 5-0 **RP6R<5:0>:** RP6 Output Pin Mapping bits Peripheral output number n is assigned to pin, RP6 (see Table 10-3 for peripheral function numbers).

bit 0

REGISTER 14-1: OCxCON1: OUTPUT COMPARE x CONTROL 1 REGISTER

11.0	11.0						
0-0	0-0					0-0	0-0
 bit 15	_	UCSIDE	UCISEL2	UCISELI	UCISELU		— hit 9
DIC 15							DIL O
R/W-0	U-0	U-0	R/W-0, HCS	R/W-0	R/W-0	R/W-0	R/W-0
ENFLT0	—	_	OCFLT0	TRIGMODE	OCM2 ⁽¹⁾	OCM1 ⁽¹⁾	OCM0 ⁽¹⁾
bit 7	•	•			•	•	bit 0
Legend:		HCS = Hardw	are Clearable/S	Settable bit			
R = Reada	ble bit	W = Writable	bit	U = Unimplem	ented bit, read	as '0'	
-n = Value	at POR	'1' = Bit is set		'0' = Bit is clea	ared	x = Bit is unkno	own
bit 15_1/	Unimplement	tod. Pead as ') '				
bit 13		o Output Comp	, are v in Idle Mc	de Control hit			
bit 15	1 = Output C	ompare x halts	in CPU Idle mo	de contror bit			
	0 = Output C	ompare x conti	nues to operate	e in CPU Idle m	ode		
bit 12-10	OCTSEL<2:0	>: Output Com	pare x Timer S	elect bits			
	111 = Periphe	eral Clock (Fcy)				
	110 = Reserv	red					
	101 = Reserv	red					
	011 = Timer5						
	010 = Timer4						
	001 = Timer3						
	000 = 1 mer2		.,				
DIT 9-8		ted: Read as (
DIL 7	1 = Eoult 0 in	it o input Enabled	e bit				
	1 = Fault 0 ir 0 = Fault 0 ir	nput is disabled					
bit 6-5	Unimplemen	ted: Read as '0)'				
bit 4	OCFLT0: PW	M Fault Conditi	ion Status bit				
	1 = PWM Fau	ult condition ha	s occurred (cle	ared in HW only	/)		
	0 = No PWM	Fault condition	has occurred	(this bit is only ι	used when OCI	M<2:0> = 111)	
bit 3	TRIGMODE:	Trigger Status I	Mode Select bit	İ			
	1 = TRIGSTA	AT (OCxCON2<	6>) is cleared v	when OCxRS =	OCxTMR or in	software	
hit 2 0	0 = TRIGSTA		ed by sollware	t hite(1)			
DIL 2-0	111 = Center			2)			
	110 = Edge-/	Aligned PWM n	node on OCx ⁽²⁾	1			
	101 = Double	e Compare Cor	ntinuous Pulse	mode: initialize	OCx pin low, t	oggle OCx stat	e continuously
	on alte	ernate matches	of OCxR and O				
	100 = 001	e Compare Sing	gie-Snot mode:	initialize OCx p	in iow, toggle C	CX state on ma	icnes of OCxR
	011 = Sinale	Compare Cont	inuous Pulse n	node: compare	events continuo	ously togale OC	x pin
	010 = Single	Compare Sing	le-Shot mode:	initialize OCx pi	n high, compar	e event forces (OCx pin low
	001 = Single	Compare Sing	le-Shot mode:	initialize OCx pi	n low, compare	e event forces O	Cx pin high
	000 = Output	t compare chan	inel is disabled				
Note 1:	The OCx output	must also be c	onfigured to an	available RPn	pin. For more i	nformation, see	Section 10.4
	"Peripheral Pin	Select".					

2: OCFA pin controls OC1-OC4 channels; OCFB pin controls the OC5-OC9 channels. OCxR and OCxRS are double-buffered only in PWM modes.

REGISTER 16-2: I2CxSTAT: I2Cx STATUS REGISTER (CONTINUED)

bit 4	P: Stop bit
	1 = Indicates that a Stop bit has been detected last
	0 = Stop bit was not detected last
	Hardware set or clear when Start, Repeated Start or Stop detected.
bit 3	S: Start bit
	 1 = Indicates that a Start (or Repeated Start) bit has been detected last 0 = Start bit was not detected last
	Hardware set or clear when Start, Repeated Start or Stop detected.
bit 2	R/W : Read/Write Information bit (when operating as I^2C slave)
	 1 = Read – indicates data transfer is output from slave 0 = Write – indicates data transfer is input to slave Hardware set or clear after reception of I²C device address byte.
bit 1	RBF: Receive Buffer Full Status bit
	 1 = Receive complete, I2CxRCV is full 0 = Receive not complete, I2CxRCV is empty Hardware set when I2CxRCV is written with received byte. Hardware clear when software reads I2CxRCV.
bit 0	TBF: Transmit Buffer Full Status bit
	1 = Transmit in progress, I2CxTRN is full 0 = Transmit complete, I2CxTRN is empty Hardware set when software writes I2CxTRN. Hardware clear at completion of data transmission

REGISTER 17-2: UxSTA: UARTx STATUS AND CONTROL REGISTER (CONTINUED)

bit 5	ADDEN: Address Character Detect bit (bit 8 of received data = 1)	
	 1 = Address Detect mode enabled. If 9-bit mode is not selected, this does not take effe 0 = Address Detect mode disabled 	ct.
bit 4	RIDLE: Receiver Idle bit (read-only)	
	1 = Receiver is Idle0 = Receiver is active	
bit 3	PERR: Parity Error Status bit (read-only)	
	 1 = Parity error has been detected for the current character (character at the top of the 0 = Parity error has not been detected 	receive FIFO)
bit 2	FERR: Framing Error Status bit (read-only)	
	 1 = Framing error has been detected for the current character (character at the top of the 0 = Framing error has not been detected 	receive FIFO)
bit 1	OERR: Receive Buffer Overrun Error Status bit (clear/read-only)	
	1 = Receive buffer has overflowed	
	 0 = Receive buffer has not overflowed (clearing a previously set OERR bit (1 → 0 trans the receiver buffer and the RSR to the empty state) 	ition) will reset
bit 0	URXDA: Receive Buffer Data Available bit (read-only)	
	 1 = Receive buffer has data; at least one more character can be read 0 = Receive buffer is empty 	
Note	1: Value of bit only affects the transmit properties of the module when the IrDA [®] encoder is enab	ed (IREN = 1).

2: If UARTEN = 1, the peripheral inputs and outputs must be configured to an available RPn pin. See Section 10.4 "Peripheral Pin Select" for more information.

REGISTER 19-1: RCFGCAL: RTCC CALIBRATION AND CONFIGURATION REGISTER⁽¹⁾ (CONTINUED)

bit 7-0 CAL<7:0>: RTC Drift Calibration bits

...

011111111 = Maximum positive adjustment; adds 508 RTC clock pulses every one minute

00000001 = Minimum positive adjustment; adds 4 RTC clock pulses every one minute

00000000 = No adjustment

11111111 = Minimum negative adjustment; subtracts 4 RTC clock pulses every one minute

10000000 = Maximum negative adjustment; subtracts 512 RTC clock pulses every one minute

- **Note 1:** The RCFGCAL register is only affected by a POR.
 - **2:** A write to the RTCEN bit is only allowed when RTCWREN = 1.
 - **3:** This bit is read-only; it is cleared to '0' on a write to the lower half of the MINSEC register.

REGISTER 19-2: PADCFG1: PAD CONFIGURATION CONTROL REGISTER

U-0	U-0	U-0	U-0	U-0	U-0	U-0	U-0	
		—	—		_	—	—	
bit 15							bit 8	
U-0	U-0	U-0	U-0	U-0	U-0	R/W-0	R/W-0	
—	—	—	—	—	—	RTSECSEL ⁽¹⁾	PMPTTL	
bit 7							bit 0	
Legend:								
R = Readable	e bit	W = Writable I	oit	U = Unimplemented bit, read as '0'				
-n = Value at	POR	'1' = Bit is set		'0' = Bit is cleared x = Bit is unknown			own	
bit 15-2	Unimplemen	ted: Read as 'o)'					
bit 1	RTSECSEL:	RTCC Seconds	Clock Output	Select bit ⁽¹⁾				
	1 = RTCC se	conds clock is	selected for the	e RTCC pin				
	0 = RTCC ala	arm pulse is sel	ected for the F	RTCC pin				
bit 0	PMPTTL: PM	P Module TTL	Input Buffer Se	elect bit				
1 = PMP module inputs (PMDx, PMCS1) use TTL input buffers								

0 = PMP module inputs use Schmitt Trigger input buffers

Note 1: To enable the actual RTCC output, the RTCOE (RCFGCAL<10>) bit must also be set.

REGISTER 21-3: AD1CON3: A/D CONTROL REGISTER 3

R/W-0	r-0	r-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
ADRC	r	r	SAMC4	SAMC3	SAMC2	SAMC1	SAMC0
bit 15							bit 8

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
ADCS7	ADCS6	ADCS5	ADCS4	ADCS3	ADCS2	ADCS1	ADCS0
bit 7							bit 0

Legend: r = Reserved bit					
R = Readable	e 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	ADRC: A/D C	Conversion Clock Source b	oit		
	1 = A/D inter	nal RC clock			
	0 = Clock de	rived from system clock			
bit 14-13	Reserved: M	aintain as '0'			
bit 12-8	SAMC<4:0>:	Auto-Sample Time bits			
	11111 = 31 T	F AD			
	00001 = 1 TA	۱D			
	00000 = 0 TA	D (not recommended)			
bit 7-0	ADCS<7:0>:	A/D Conversion Clock Se	lect bits		
	11111111				
	••••• =	Reserved, do not use			
	0100000				
	00111111 =	64 TCY			
00111110 = 63 T CY					

© 2010 Microchip Technology Inc.

00000001 = 2 * Tcy 00000000 = Tcy



REGISTER 25-1: CW1: FLASH CONFIGURATION WORD 1

R/PO-1	R/PO-1	R/PO-1	R/PO-1	R/PO-1	R/PO-1	R/PO-1	R/PO-1
—	—	—	—	—	—	—	—
bit 23							bit 16

r-x	R/PO-1	R/PO-1	R/PO-1	R/PO-1	r-1	R/PO-1	R/PO-1
r	JTAGEN	GCP	GWRP	DEBUG	r	ICS1	ICS0
bit 15							bit 8

R/PO-1	R/PO-1	R/PO-1	R/PO-1	R/PO-1	R/PO-1	R/PO-1	R/PO-1
FWDTEN	WINDIS	—	FWPSA	WDTPS3	WDTPS2	WDTPS1	WDTPS0
bit 7							bit 0

Legend:	r = Reserved bit		
R = Readable bit	PO = Program Once bit	U = Unimplemented bit, read	l as '0'
-n = Value when device is unprogrammed		'1' = Bit is set	'0' = Bit is cleared

bit 23-16	Reserved
bit 15	Reserved: The value is unknown; program as '0'
bit 14	JTAGEN: JTAG Port Enable bit
	1 = JTAG port is enabled0 = JTAG port is disabled
bit 13	GCP: General Segment Program Memory Code Protection bit
	1 = Code protection is disabled0 = Code protection is enabled for the entire program memory space
bit 12	GWRP: General Segment Code Flash Write Protection bit
	1 = Writes to program memory are allowed0 = Writes to program memory are disabled
bit 11	DEBUG: Background Debugger Enable bit
	1 = Device resets into Operational mode0 = Device resets into Debug mode
bit 10	Reserved: Always maintain as '1'
bit 9-8	ICS<1:0>: Emulator Pin Placement Select bits
	 11 = Emulator functions are shared with PGEC1/PGED1 10 = Emulator functions are shared with PGEC2/PGED2 01 = Emulator functions are shared with PGEC3/PGED3 00 = Reserved; do not use
bit 7	FWDTEN: Watchdog Timer Enable bit
	1 = Watchdog Timer is enabled0 = Watchdog Timer is disabled
bit 6	WINDIS: Windowed Watchdog Timer Disable bit
	1 = Standard Watchdog Timer enabled 0 = Windowed Watchdog Timer enabled; FWDTEN must be '1'
bit 5	Reserved
bit 4	FWPSA: WDT Prescaler Ratio Select bit 1 = Prescaler ratio of 1:128 0 = Prescaler ratio of 1:32

27.0 DEVELOPMENT SUPPORT

The PIC[®] microcontrollers and dsPIC[®] digital signal controllers are supported with a full range of software and hardware development tools:

- Integrated Development Environment
- MPLAB[®] IDE Software
- Compilers/Assemblers/Linkers
 - MPLAB C Compiler for Various Device Families
 - HI-TECH C for Various Device Families
 - MPASM[™] Assembler
 - MPLINK[™] Object Linker/ MPLIB[™] Object Librarian
 - MPLAB Assembler/Linker/Librarian for Various Device Families
- Simulators
 - MPLAB SIM Software Simulator
- Emulators
 - MPLAB REAL ICE™ In-Circuit Emulator
- In-Circuit Debuggers
 - MPLAB ICD 3
 - PICkit™ 3 Debug Express
- Device Programmers
 - PICkit[™] 2 Programmer
 - MPLAB PM3 Device Programmer
- Low-Cost Demonstration/Development Boards, Evaluation Kits, and Starter Kits

27.1 MPLAB Integrated Development Environment Software

The MPLAB IDE software brings an ease of software development previously unseen in the 8/16/32-bit microcontroller market. The MPLAB IDE is a Windows[®] operating system-based application that contains:

- A single graphical interface to all debugging tools
 - Simulator
 - Programmer (sold separately)
 - In-Circuit Emulator (sold separately)
 - In-Circuit Debugger (sold separately)
- A full-featured editor with color-coded context
- A multiple project manager
- Customizable data windows with direct edit of contents
- · High-level source code debugging
- · Mouse over variable inspection
- Drag and drop variables from source to watch windows
- · Extensive on-line help
- Integration of select third party tools, such as IAR C Compilers

The MPLAB IDE allows you to:

- Edit your source files (either C or assembly)
- One-touch compile or assemble, and download to emulator and simulator tools (automatically updates all project information)
- Debug using:
 - Source files (C or assembly)
 - Mixed C and assembly
 - Machine code

MPLAB IDE supports multiple debugging tools in a single development paradigm, from the cost-effective simulators, through low-cost in-circuit debuggers, to full-featured emulators. This eliminates the learning curve when upgrading to tools with increased flexibility and power.

64-Lead Plastic Thin Quad Flatpack (PT) – 10x10x1 mm Body, 2.00 mm [TQFP]

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



Units		MILLIMETERS		
Dimension Limits		MIN	NOM	MAX
Contact Pitch	E	0.50 BSC		
Contact Pad Spacing	C1		11.40	
Contact Pad Spacing	C2		11.40	
Contact Pad Width (X64)	X1			0.30
Contact Pad Length (X64)	Y1			1.50
Distance Between Pads	G	0.20		

Notes:

1. Dimensioning and tolerancing per ASME Y14.5M

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

Microchip Technology Drawing No. C04-2085A