

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	I²C, IrDA, SPI, UART/USART, USB OTG
Peripherals	Brown-out Detect/Reset, GFX, LVD, POR, PWM, WDT
Number of I/O	52
Program Memory Size	256KB (85.5K x 24)
Program Memory Type	FLASH
EEPROM Size	-
RAM Size	96K x 8
Voltage - Supply (Vcc/Vdd)	2.2V ~ 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/pic24fj256da206-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:

- PIC24FJ128DA106 PIC24FJ128DA206
- PIC24FJ256DA106
- PIC24FJ256DA206 • PIC24FJ128DA110 PIC24FJ128DA210
- PIC24FJ256DA110
 - PIC24FJ256DA210

The PIC24FJ256DA210 family enhances on the existing line of Microchip's 16-bit microcontrollers, adding a new Graphics Controller (GFX) module to interface with a graphical LCD display and also adds large data RAM, up to 96 Kbytes. The PIC24FJ256DA210 family allows the CPU to fetch data directly from an external memory device using the EPMP module.

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 (DSCs). 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 32 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 PIC24FJ256DA210 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.

OSCILLATOR OPTIONS AND 1.1.3 FEATURES

All of the devices in the PIC24FJ256DA210 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 Low-Power 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 (FSCM). 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 family, and shares some compatibility with the pinout schema for PIC18 and dsPIC30. This extends the ability of applications to grow from the relatively simple, to the powerful and complex, yet still selecting a Microchip device.

		Pin Number						
Function	ion 64-Pin 100-Pin 121-Pin BGA		Buffer	Description				
TCK	27	38	J6	Ι	ST	JTAG Test Clock Input.		
TDI	28	60	G11	Ι	ST	JTAG Test Data Input.		
TDO	24	61	G9	0	_	JTAG Test Data Output.		
TMS	23	17	G3	Ι	ST	JTAG Test Mode Select Input.		
USBID	33	51	K10	Ι	ST	USB OTG ID (OTG mode only).		
USBOEN	12	21	H2	0	_	USB Output Enable Control (for external transceiver).		
VBUS	34	54	H8	Ι	ANA	USB Voltage, Host mode (5V).		
VBUSCHG	49	76	A11	0	_	External USB VBUS Charge Output.		
VBUSON	11	20	H1	0	_	USB OTG External Charge Pump Control.		
VBUSST	58	87	B6	Ι	ANA	USB OTG Internal Charge Pump Feedback Control.		
VBUSVLD	58	87	B6	Ι	ST	USB VBUS Boost Generator, Comparator Input 1.		
VCAP	56	85	B7	Р	_	External Filter Capacitor Connection (regulator enabled).		
VCMPST1	58	87	B6	Ι	ST	USB VBUS Boost Generator, Comparator Input 1.		
VCMPST2	59	88	A6	Ι	ST	USB VBUS Boost Generator, Comparator Input 2.		
VCPCON	49	76	A11	0	_	USB OTG VBUS PWM/Charge Output.		
Vdd	10, 26, 38	2, 16, 37, 46, 62	C2, C9, F8, G5, H6, K8, H4, E5	Р	_	Positive Supply for Peripheral Digital Logic and I/O Pins.		
VMIO	14	23	J2	Ι	ST	USB Differential Minus Input/Output (external transceiver).		
VPIO	13	22	J1	Ι	ST	USB Differential Plus Input/Output (external transceiver).		
VREF-	15	28, 24 ⁽⁴⁾	L2, K1 ⁽⁴⁾	Ι	ANA	A/D and Comparator Reference Voltage (low) Input.		
VREF+	16	29, 25 ⁽⁴⁾	K3, K2 ⁽⁴⁾	Ι	ANA	A/D and Comparator Reference Voltage (high) Input.		
Vss	9, 25, 41	15, 36, 45, 65, 75	B10, F5, F10, G6, G7, H3, D4, D5	Ρ	_	Ground Reference for Logic and I/O Pins.		
VSYNC	1	96	C3	0	—	Graphics Display Vertical Sync Pulse.		
VUSB	35	55	H9	Р	—	USB Voltage (3.3V).		
Legend:	TTL = TTL inp	ut buffer			ST = Schmitt Trigger input buffer			

TABLE 1-3:	PIC24FJ256DA210 FAMILY PINOUT DESCRIPTIONS ((CONTINUED)

ANA = Analog level input/output $I^2C^{TM} = I^2C/SMBus input buffer$

Note 1: The alternate EPMP pins are selected when the $\overline{\text{ALTPMP}}$ (CW3<12>) bit is programmed to '0'.

2: The PMSC2 signal will replace the PMA15 signal on the 15-pin PMA when CSF<1:0> = 01 or 10.

3: The PMCS1 signal will replace the PMA14 signal on the 14-pin PMA when CSF<1:0> = 10.

4: The alternate VREF pins selected when the ALTVREF (CW1<5>) bit is programmed to '0'.

TABLE 4-5: ICN REGISTER MAP

File Name	Addr	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	All Resets
CNPD1	0056	CN15PDE	CN14PDE	CN13PDE	CN12PDE	CN11PDE	CN10PDE	CN9PDE	CN8PDE	CN7PDE	CN6PDE	CN5PDE	CN4PDE	CN3PDE	CN2PDE	CN1PDE	CN0PDE	0000
CNPD2	0058	CN31PDE	CN30PDE	CN29PDE	CN28PDE	CN27PDE	CN26PDE	CN25PDE	CN24PDE	CN23PDE	CN22PDE	CN21PDE ⁽¹⁾	CN20PDE ⁽¹⁾	CN19PDE ⁽¹⁾	CN18PDE	CN17PDE	CN16PDE	0000
CNPD3	005A	CN47PDE ⁽¹⁾	CN46PDE ⁽¹⁾	CN45PDE ⁽¹⁾	CN44PDE ⁽¹⁾	CN43PDE ⁽¹⁾	CN42PDE ⁽¹⁾	CN41PDE ⁽¹⁾	CN40PDE ⁽¹⁾	CN39PDE ⁽¹⁾	CN38PDE ⁽¹⁾	CN37PDE ⁽¹⁾	CN36PDE ⁽¹⁾	CN35PDE ⁽¹⁾	CN34PDE ⁽¹⁾	CN33PDE ⁽¹⁾	CN32PDE	0000
CNPD4	005C	CN63PDE	CN62PDE	CN61PDE	CN60PDE	CN59PDE	CN58PDE	CN57PDE ⁽¹⁾	CN56PDE	CN55PDE	CN54PDE	CN53PDE	CN52PDE	CN51PDE	CN50PDE	CN49PDE	CN48PDE ⁽¹⁾	0000
CNPD5	005E	CN79PDE ⁽¹⁾	CN78PDE ⁽¹⁾	CN77PDE ⁽¹⁾	CN76PDE ⁽¹⁾	CN75PDE ⁽¹⁾	CN74PDE ⁽¹⁾	CN73PDE ⁽¹⁾	_	CN71PDE	CN70PDE ⁽¹⁾	CN69PDE	CN68PDE	CN67PDE ⁽¹⁾	CN66PDE ⁽¹⁾	CN65PDE	CN64PDE	0000
CNPD6	0060	_	_	_	_	_	_	_	—	—	—	—	CN84PDE	CN83PDE	CN82PDE ⁽¹⁾	CN81PDE ⁽¹⁾	CN80PDE(1)	0000
CNEN1	0062	CN15IE	CN14IE	CN13IE	CN12IE	CN11IE	CN10IE	CN9IE	CN8IE	CN7IE	CN6IE	CN5IE	CN4IE	CN3IE	CN2IE	CN1IE	CN0IE	0000
CNEN2	0064	CN31IE	CN30IE	CN29IE	CN28IE	CN27IE	CN26IE	CN25IE	CN24IE	CN23IE	CN22IE	CN21IE ⁽¹⁾	CN20IE ⁽¹⁾	CN19IE ⁽¹⁾	CN18IE	CN17IE	CN16IE	0000
CNEN3	0066	CN47IE ⁽¹⁾	CN46IE ⁽¹⁾	CN45IE ⁽¹⁾	CN44IE ⁽¹⁾	CN43IE ⁽¹⁾	CN42IE ⁽¹⁾	CN41IE ⁽¹⁾	CN40IE ⁽¹⁾	CN39IE ⁽¹⁾	CN38IE ⁽¹⁾	CN37IE ⁽¹⁾	CN36IE ⁽¹⁾	CN35IE ⁽¹⁾	CN34IE ⁽¹⁾	CN33IE ⁽¹⁾	CN32IE	0000
CNEN4	0068	CN63IE	CN62IE	CN61IE	CN60IE	CN59IE	CN58IE	CN57IE ⁽¹⁾	CN56IE	CN55IE	CN54IE	CN53IE	CN52IE	CN51IE	CN50IE	CN49IE	CN48IE ⁽¹⁾	0000
CNEN5	006A	CN79IE ⁽¹⁾	CN78IE ⁽¹⁾	CN77IE ⁽¹⁾	CN76IE ⁽¹⁾	CN75IE ⁽¹⁾	CN74IE ⁽¹⁾	CN73IE ⁽¹⁾	—	CN71IE	CN70IE ⁽¹⁾	CN69IE	CN68IE	CN67IE ⁽¹⁾	CN66IE ⁽¹⁾	CN65IE	CN64IE	0000
CNEN6	006C	_	_	_	_	_	_	_	—	—	—	—	CN84IE	CN83IE	CN82IE ⁽¹⁾	CN81IE ⁽¹⁾	CN80IE ⁽¹⁾	0000
CNPU1	006E	CN15PUE	CN14PUE	CN13PUE	CN12PUE	CN11PUE	CN10PUE	CN9PUE	CN8PUE	CN7PUE	CN6PUE	CN5PUE	CN4PUE	CN3PUE	CN2PUE	CN1PUE	CN0PUE	0000
CNPU2	0070	CN31PUE	CN30PUE	CN29PUE	CN28PUE	CN27PUE	CN26PUE	CN25PUE	CN24PUE	CN23PUE	CN22PUE	CN21PUE ⁽¹⁾	CN20PUE ⁽¹⁾	CN19PUE ⁽¹⁾	CN18PUE	CN17PUE	CN16PUE	0000
CNPU3	0072	CN47PUE ⁽¹⁾	CN46PUE ⁽¹⁾	CN45PUE ⁽¹⁾	CN44PUE ⁽¹⁾	CN43PUE ⁽¹⁾	CN42PUE ⁽¹⁾	CN41PUE ⁽¹⁾	CN40PUE ⁽¹⁾	CN39PUE ⁽¹⁾	CN38PUE ⁽¹⁾	CN37PUE ⁽¹⁾	CN36PUE ⁽¹⁾	CN35PUE ⁽¹⁾	CN34PUE ⁽¹⁾	CN33PUE ⁽¹⁾	CN32PUE	0000
CNPU4	0074	CN63PUE	CN62PUE	CN61PUE	CN60PUE	CN59PUE	CN58PUE	CN57PUE ⁽¹⁾	CN56PUE	CN55PUE	CN54PUE	CN53PUE	CN52PUE	CN51PUE	CN50PUE	CN49PUE	CN48PUE ⁽¹⁾	0000
CNPU5	0076	CN79PUE ⁽¹⁾	CN78PUE ⁽¹⁾	CN77PUE ⁽¹⁾	CN76PUE ⁽¹⁾	CN75PUE ⁽¹⁾	CN74PUE ⁽¹⁾	CN73PUE ⁽¹⁾	—	CN71PUE	CN70PUE ⁽¹⁾	CN69PUE	CN68PUE	CN67PUE ⁽¹⁾	CN66PUE ⁽¹⁾	CN65PUE	CN64PUE	0000
CNPU6	0078	—	—	—	—	—	—	—	_	—	—	—	CN84PUE	CN83PUE	CN82PUE ⁽¹⁾	CN81PUE ⁽¹⁾	CN80PUE ⁽¹⁾	0000

 Legend:
 — = unimplemented, read as '0'. Reset values are shown in hexadecimal.

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

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

// C example using MPLAB C30	
unsigned long progAddr = 0xXXXXXX;	// Address of row to write
unsigned int offset;	
//Set up pointer to the first memory location to	be written
TBLPAG = progAddr>>16;	// Initialize PM Page Boundary SFR
offset = progAddr & 0xFFFF;	<pre>// Initialize lower word of address</pre>
<pre>builtin_tblwtl(offset, 0x0000);</pre>	// Set base address of erase block
	// with dummy latch write
NVMCON = 0x4042;	// Initialize NVMCON
asm("DISI #5");	// Block all interrupts with priority <7
	// for next 5 instructions
builtin_write_NVM();	// check function to perform unlock
	// sequence and set WR

EXAMPLE 5-3: LOADING THE WRITE BUFFERS

; ; ;	<pre>Set up NVMCON for row programming operations MOV #0x4001, W0 MOV W0, NVMCON Set up a pointer to the first program memory program memory selected, and writes enabled MOV #0x0000, W0</pre>	; ; loc ;	Initialize NVMCON ation to be written
	MOV W0, TBLPAG	;	Initialize PM Page Boundary SFR
	MOV #0x6000, W0	;	An example program memory address
;	Perform the TBLWT instructions to write the 1	latc	hes
;	0th_program_word		
	MOV #LOW_WORD_0, W2	;	
	MOV #HIGH_BYTE_0, W3	;	
	TBLWTL W2, [W0]	;	Write PM low word into program latch
	TBLWTH W3, [W0++]	;	Write PM high byte into program latch
;	lst_program_word		
	MOV #LOW_WORD_1, W2	;	
	MOV #HIGH_BYTE_1, W3	;	
	TBLWTL W2, [W0]	;	Write PM low word into program latch
	TBLWTH W3, [W0++]	;	Write PM high byte into program latch
;	2nd_program_word		
	MOV #LOW_WORD_2, W2	;	
	MOV #HIGH_BYTE_2, W3	;	
	TBLWTL W2, [W0]	;	Write PM low word into program latch
	TBLWTH W3, [W0++]	;	Write PM high byte into program latch
	•		
	62rd program word		
'	MOV #LOW WORD 63 W2		
	MOV #HIGH BYTE 63 W3	;	
	TBLWTL W2. $[W0]$;	Write PM low word into program latch
	TBLWTH W3. [W0]	;	Write PM high byte into program latch
	100, []	'	Miles in migh syst into program factor

EXAMPLE 5-4: INITIATING A PROGRAMMING SEQUENCE

DISI	#5	;	Block all interrupts with priority <7
		;	for next 5 instructions
MOV.B	#0x55, W0		
MOV	W0, NVMKEY	;	Write the 0x55 key
MOV.B	#0xAA, W1	;	
MOV	W1, NVMKEY	;	Write the OxAA key
BSET	NVMCON, #WR	;	Start the programming sequence
NOP		;	Required delays
NOP			
BTSC	NVMCON, #15	;	and wait for it to be
BRA	\$-2	;	completed

REGISTE	R 8-1: OSCCON: OSCILLATOR CONTROL REGISTER (CONTINUED)
bit 10-8	NOSC<2:0>: New Oscillator Selection bits ⁽¹⁾ 111 = Fast RC Oscillator with Postscaler (FRCDIV) 110 = Fast RC/16 Oscillator 101 = Low-Power RC Oscillator (LPRC) 100 = Secondary Oscillator (SOSC) 011 = Primary Oscillator with PLL module (XTPLL, HSPLL, ECPLL) 010 = Primary Oscillator (XT, HS, EC) 001 = Fast RC Oscillator with Postscaler and PLL module (FRCPLL) 000 = Fast RC Oscillator (FRC)
bit 7	CLKLOCK: Clock Selection Lock Enabled bit
	If FSCM is enabled (FCKSM1 = 1): 1 = Clock and PLL selections are locked 0 = Clock and PLL selections are not locked and may be modified by setting the OSWEN bit If FSCM is disabled (FCKSM1 = 0): Clock and PLL selections are never locked and may be modified by setting the OSWEN bit.
bit 6	IOLOCK: I/O Lock Enable bit ⁽²⁾
	 1 = I/O lock is active 0 = I/O lock is not active
bit 5	LOCK: PLL Lock Status bit ⁽³⁾
	 1 = PLL module is in lock or PLL module start-up timer is satisfied 0 = PLL module is out of lock, PLL start-up timer is running or PLL is disabled
bit 4	Unimplemented: Read as '0'
bit 3	CF: Clock Fail Detect bit
	 1 = FSCM has detected a clock failure 0 = No clock failure has been detected
bit 2	POSCEN: Primary Oscillator Sleep Enable bit
	 1 = Primary Oscillator continues to operate during Sleep mode 0 = Primary Oscillator is disabled during Sleep mode
bit 1	SOSCEN: 32 kHz Secondary Oscillator (SOSC) Enable bit
	1 = Enable the Secondary Oscillator
	0 = Disable the Secondary Oscillator
bit 0	OSWEN: Oscillator Switch Enable bit
	 1 = Initiate an oscillator switch to the clock source specified by the NOSC<2:0> bits 0 = Oscillator switch is complete
Note 1:	Reset values for these bits are determined by the FNOSC Configuration bits.
2:	The state of the IOLOCK bit can only be changed once an unlocking sequence has been executed. In addition, if the IOL1WAY Configuration bit is '1', once the IOLOCK bit is set, it cannot be cleared.

3: Also resets to '0' during any valid clock switch or whenever a non PLL Clock mode is selected.

8.4 Clock Switching Operation

With few limitations, applications are free to switch between any of the four clock sources (POSC, SOSC, FRC and LPRC) under software control and at any time. To limit the possible side effects that could result from this flexibility, PIC24F devices have a safeguard lock built into the switching process.

Note: The Primary Oscillator mode has three different submodes (XT, HS and EC) which are determined by the POSCMDx Configuration bits. While an application can switch to and from Primary Oscillator mode in software, it cannot switch between the different primary submodes without reprogramming the device.

8.4.1 ENABLING CLOCK SWITCHING

To enable clock switching, the FCKSM1 Configuration bit in CW2 must be programmed to '0'. (Refer to **Section 27.1 "Configuration Bits"** for further details.) If the FCKSM1 Configuration bit is unprogrammed ('1'), the clock switching function and Fail-Safe Clock Monitor function are disabled. This is the default setting.

The NOSCx (OSCCON<10:8>) control bits do not control the clock selection when clock switching is disabled. However, the COSCx (OSCCON<14:12>) control bits will reflect the clock source selected by the FNOSCx Configuration bits.

The OSWEN (OSCCON<0>) control bit has no effect when clock switching is disabled; It is held at '0' at all times.

8.4.2 OSCILLATOR SWITCHING SEQUENCE

At a minimum, performing a clock switch requires this basic sequence:

- If desired, read the COSCx (OSCCON<14:12>) control bits to determine the current oscillator source.
- 2. Perform the unlock sequence to allow a write to the OSCCON register high byte.
- Write the appropriate value to the NOSCx (OSCCON<10:8>) control bits for the new oscillator source.
- 4. Perform the unlock sequence to allow a write to the OSCCON register low byte.
- 5. Set the OSWEN bit to initiate the oscillator switch.

Once the basic sequence is completed, the system clock hardware responds automatically as follows:

- 1. The clock switching hardware compares the COSCx bits with the new value of the NOSCx bits. If they are the same, then the clock switch is a redundant operation. In this case, the OSWEN bit is cleared automatically and the clock switch is aborted.
- If a valid clock switch has been initiated, the LOCK (OSCCON<5>) and CF (OSCCON<3>) bits are cleared.
- The new oscillator is turned on by the hardware if it is not currently running. If a crystal oscillator must be turned on, the hardware will wait until the OST expires. If the new source is using the PLL, then the hardware waits until a PLL lock is detected (LOCK = 1).
- 4. The hardware waits for 10 clock cycles from the new clock source and then performs the clock switch.
- 5. The hardware clears the OSWEN bit to indicate a successful clock transition. In addition, the NOSCx bit values are transferred to the COSCx bits.
- The old clock source is turned off at this time, with the exception of LPRC (if WDT or FSCM are enabled) or SOSC (if SOSCEN remains set).
 - Note 1: The processor will continue to execute code throughout the clock switching sequence. Timing-sensitive code should not be executed during this time.
 - 2: Direct clock switches between any Primary Oscillator mode with PLL and FRCPLL modes are not permitted. This applies to clock switches in either direction. In these instances, the application must switch to FRC mode as a transition clock source between the two PLL modes.

9.0 POWER-SAVING FEATURES

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 10. "Power-Saving Features" (DS39698). The information in this data sheet supersedes the information in the FRM.

The PIC24FJ256DA210 family of devices provides the ability to manage power consumption by selectively managing clocking to the CPU and the peripherals. In general, a lower clock frequency and a reduction in the number of circuits being clocked constitutes lower consumed power. All PIC24F devices manage power consumption in four different ways:

- Clock Frequency
- Instruction-Based Sleep and Idle modes
- Software Controlled Doze mode
- · Selective Peripheral Control in Software

Combinations of these methods can be used to selectively tailor an application's power consumption, while still maintaining critical application features, such as timing-sensitive communications.

9.1 Clock Frequency and Clock Switching

PIC24F devices allow for a wide range of clock frequencies to be selected under application control. If the system clock configuration is not locked, users can choose low-power or high-precision oscillators by simply changing the NOSC bits. The process of changing a system clock during operation, as well as limitations to the process, are discussed in more detail in **Section 8.0** "Oscillator Configuration".

9.2 Instruction-Based Power-Saving Modes

PIC24F devices have two special power-saving modes that are entered through the execution of a special PWRSAV instruction. Sleep mode stops clock operation and halts all code execution; Idle mode halts the CPU and code execution, but allows peripheral modules to continue operation. The assembly syntax of the PWRSAV instruction is shown in Example 9-1.

Sleep and Idle modes can be exited as a result of an enabled interrupt, WDT time-out or a device Reset. When the device exits these modes, it is said to "wake-up".

9.2.1 SLEEP MODE

Sleep mode has these features:

- The system clock source is shut down. If an on-chip oscillator is used, it is turned off.
- The device current consumption will be reduced to a minimum, provided that no I/O pin is sourcing current.
- The Fail-Safe Clock Monitor (FSCM) does not operate during Sleep mode since the system clock source is disabled.
- The LPRC clock will continue to run in Sleep mode if the WDT is enabled.
- The WDT, if enabled, is automatically cleared prior to entering Sleep mode.
- Some device features or peripherals may continue to operate in Sleep mode. This includes items such as the input change notification on the I/O ports or peripherals that use an external clock input. Any peripheral that requires the system clock source for its operation will be disabled in Sleep mode. Users can opt to make the voltage regulator enter standby mode on entering Sleep mode by clearing the VREGS bit (RCON<8>). This will decrease current consumption but will add a delay, TVREG, to the wake-up time. For this reason, applications that do not use the voltage regulator should set this bit.

The device will wake-up from Sleep mode on any of the these events:

- On any interrupt source that is individually enabled
- · On any form of device Reset
- On a WDT time-out

On wake-up from sleep, the processor will restart with the same clock source that was active when Sleep mode was entered.

EXAMPLE 9-1:	PWRSAV INSTRUCTION		
	SYNTAX		

PWRSAV	#0	;	Put	the	device	into	SLEEF	mode
PWRSAV	#1	;	Put	the	device	into	IDLE	mode

10.1.1 I/O PORT WRITE/READ TIMING

One instruction cycle is required between a port direction change or port write operation and a read operation of the same port. Typically, this instruction would be a NOP.

10.1.2 OPEN-DRAIN CONFIGURATION

In addition to the PORT, LAT and TRIS registers for data control, each port pin can also be individually configured for either a digital or open-drain output. This is controlled by the Open-Drain Control register, ODCx, associated with each port. Setting any of the bits configures the corresponding pin to act as an open-drain output.

The open-drain feature allows the generation of outputs higher than VDD (e.g., 5V) on any desired digital only pins by using external pull-up resistors. The maximum open-drain voltage allowed is the same as the maximum VIH specification.

10.1.3 CONFIGURING D+ AND D- PINS (RG2 AND RG3)

The input buffers of the RG2 and RG3 pins are by default, tri-stated. To use these pins as input pins, the UTRDIS bit (U1CNFG2<0>) should be set which enables the input buffers on these pins.

10.2 Configuring Analog Port Pins (ANSEL)

The ANSx and TRISx registers control the operation of the pins with analog function. Each port pin with analog function is associated with one of the ANS bits (see Register 10-1 through Register 10-7), which decides if the pin function should be analog or digital. Refer to Table 10-1 for detailed behavior of the pin for different ANSx and TRISx bit settings.

When reading the PORT register, all pins configured as analog input channels will read as cleared (a low level).

10.2.1 ANALOG INPUT PINS AND VOLTAGE CONSIDERATIONS

The voltage tolerance of pins used as device inputs is dependent on the pin's input function. Pins that are used as digital only inputs are able to handle DC voltages of up to 5.5V, a level typical for digital logic circuits. In contrast, pins that also have analog input functions of any kind can only tolerate voltages up to VDD. Voltage excursions beyond VDD on these pins should always be avoided. Table 10-2 summarizes the input capabilities. Refer to **Section 30.1 "DC Characteristics"** for more details.

Pin Function	ANSx Setting	TRISx Setting	Comments
Analog Input	1	1	It is recommended to keep ANSx = 1.
Analog Output	1	1	It is recommended to keep ANSx = 1.
Digital Input	0	1	Firmware must wait at least one instruction cycle after configuring a pin as a digital input before a valid input value can be read.
Digital Output	0	0	Make sure to disable the analog output function on the pin if any is present.

TABLE 10-1: CONFIGURING ANALOG/DIGITAL FUNCTION OF AN I/O PIN

TABLE 10-2: INPUT VOLTAGE LEVELS FOR PORT OR PIN TOLERATED DESCRIPTION INPUT

Port or Pin	Tolerated Input	Description
PORTA ⁽¹⁾ <10:9, 7:6>		
PORTB<15:0>		
PORTC ⁽¹⁾ <15:12, 4>		
PORTD<7:6>	VDD	Only VDD input levels are tolerated.
PORTE ⁽¹⁾ <9>		
PORTF<0>		
PORTG<9:6, 3:2>		
PORTA ⁽¹⁾ <15:14, 5:0>		
PORTC ⁽¹⁾ <3:1>		
PORTD ⁽¹⁾ <15:8, 5:0>	E E)/	Tolerates input levels above VDD, useful
PORTE ⁽¹⁾ <8:0>	5.5V	for most standard logic.
PORTF ⁽¹⁾ <13:12, 8:7, 5:1>		
PORTG ⁽¹⁾ <15:12, 1:0>		

Note 1: Not all of the pins of these PORTS are implemented in 64-pin devices (PIC24FJXXXDAX06); refer to the device pinout diagrams for the details.

10.4.6 PERIPHERAL PIN SELECT REGISTERS

The PIC24FJ256DA210 family of devices implements a total of 37 registers for remappable peripheral configuration:

- Input Remappable Peripheral Registers (21)
- Output Remappable Peripheral Registers (16)

Note: Input and output register values can only be changed if IOLOCK (OSCCON<6>) = 0. See Section 10.4.4.1 "Control Register Lock" for a specific command sequence.

REGISTER 10-8: RPINR0: PERIPHERAL PIN SELECT INPUT REGISTER 0

U-0	U-0	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1
_	_	INT1R5	INT1R4	INT1R3	INT1R2	INT1R1	INT1R0
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 bit W = Writable bit			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	INT1R<5:0>: Assign External Interrupt 1 (INT1) to Corresponding RPn or RPIn Pin bits
bit 7-0	Unimplemented: Read as '0'

REGISTER 10-9: RPINR1: PERIPHERAL PIN SELECT INPUT REGISTER 1

U-0	U-0	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1
—	—	INT3R5	INT3R4	INT3R3	INT3R2	INT3R1	INT3R0
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
—	—	INT2R5	INT2R4	INT2R3	INT2R2	INT2R1	INT2R0
bit 7							bit 0
Legend:							
R = Readable bit W = Writable bit		oit	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	INT3R<5:0>: Assign External Interrupt 3 (INT3) to Corresponding RPn or RPIn Pin bits
bit 7-6	Unimplemented: Read as '0'
bit 5-0	INT2R<5:0>: Assign External Interrupt 2 (INT2) to Corresponding RPn or RPIn Pin bits

x = Bit is unknown

REGISTER 10-28: RPINR29: PERIPHERAL PIN SELECT INPUT REGISTER 29

U-0	U-0	U-0	U-0	U-0	U-0	U-0	U-0
	—	—	—	—	—	—	—
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
_	_	SS3R5	SS3R4	SS3R3	SS3R2	SS3R1	SS3R0
bit 7	•						bit 0
Legend:							
R = Readable bit W = Writable bit				U = Unimplen	nented bit, read	d as '0'	

'0' = Bit is cleared

bit 15-6 Unimplemented: Read as '0'

'1' = Bit is set

-n = Value at POR

bit 5-0 SS3R<5:0>: Assign SPI3 Slave Select Input (SS31IN) to Corresponding RPn or RPIn Pin bits

|--|

Legend:							
bit 7							bit 0
—	—	RP24R5	RP24R4	RP24R3	RP24R2	RP24R1	RP24R0
U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
bit 15							bit 8
—	_	RP25R5	RP25R4	RP25R3	RP25R2	RP25R1	RP25R0
U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0

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 15-14 Unimplemented: Read as '0'

- bit 13-8
 RP25R<5:0>: RP25 Output Pin Mapping bits

 Peripheral output number n is assigned to pin, RP25 (see Table 10-4 for peripheral function numbers).

 bit 7-6
 Unimplemented: Read as '0'
- bit 5-0 **RP24R<5:0>:** RP24 Output Pin Mapping bits Peripheral output number n is assigned to pin, RP24 (see Table 10-4 for peripheral function numbers).

REGISTER 10-42: RPOR13: PERIPHERAL PIN SELECT OUTPUT REGISTER 13

-							
U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
_	—	RP27R5	RP27R4	RP27R3	RP27R2	RP27R1	RP27R0
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
—	_	RP26R5	RP26R4	RP26R3	RP26R2	RP26R1	RP26R0
bit 7							bit 0
Legend:							
R = Readable bit W = Writable bit		bit	U = Unimplen	nented bit, read	l as '0'		

bit 15-14 **Unimplemented:** Read as '0'

bit 13-8 **RP27R<5:0>:** RP27 Output Pin Mapping bits

'1' = Bit is set

Peripheral output number n is assigned to pin, RP27 (see Table 10-4 for peripheral function numbers).

'0' = Bit is cleared

bit 7-6 Unimplemented: Read as '0'

bit 5-0 **RP26R<5:0>:** RP26 Output Pin Mapping bits Peripheral output number n is assigned to pin, RP26 (see Table 10-4 for peripheral function numbers).

-n = Value at POR

x = Bit is unknown

REGISTER 16-1: I2CxCON: I2Cx CONTROL REGISTER (CONTINUED)

bit 5	ACKDT: Acknowledge Data bit (when operating as I ² C master. Applicable during master receive.)
	Value that will be transmitted when the software initiates an Acknowledge sequence.
	1 = Sends NACK during Acknowledge
	0 = Sends ACK during Acknowledge
bit 4	ACKEN: Acknowledge Sequence Enable bit (when operating as I ² C master. Applicable during master receive.)
	1 = Initiates Acknowledge sequence on SDAx and SCLx pins and transmits the ACKDT data bit. Hardware is clear at the end of the master Acknowledge sequence.
	0 = Acknowledge sequence is not in progress
bit 3	RCEN: Receive Enable bit (when operating as I ² C master)
	 1 = Enables Receive mode for I²C. Hardware is clear at the end of the eighth bit of the master receive data byte.
	0 = Receive sequence is not in progress
bit 2	PEN: Stop Condition Enable bit (when operating as I ² C master)
	1 = Initiates Stop condition on the SDAx and SCLx pins. Hardware is clear at the end of the master Stop sequence.
	0 = Stop condition is not in progress
bit 1	RSEN: Repeated Start Condition Enabled bit (when operating as I ² C master)
	1 = Initiates Repeated Start condition on the SDAx and SCLx pins. Hardware is clear at the end of the master Repeated Start sequence
	0 = Repeated Start condition is not in progress
bit 0	SEN: Start Condition Enabled bit (when operating as I ² C master)
	1 = Initiates Start condition on SDAx and SCLx pins. Hardware is clear at end of the master Start sequence.
	0 = Start condition is not in progress

REGISTER 17-2: UxSTA: UARTx STATUS AND CONTROL REGISTER

R/W-0	R/W-0	R/W-0	U-0	R/W-0 HC	R/W-0	R-0, HSC	R-1, HSC
UTXISEL1	UTXINV ⁽¹⁾	UTXISEL0	—	UTXBRK	UTXEN ⁽²⁾	UTXBF	TRMT
bit 15							

R/W-0	R/W-0	R/W-0	R-1, HSC	R-0, HSC	R-0, HSC	R/C-0, HS	R-0, HSC
URXISEL1	URXISEL0	ADDEN	RIDLE	PERR	FERR	OERR	URXDA
bit 7							bit 0

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

bit 15,13 UTXISEL<1:0>: Transmission Interrupt Mode Selection bits

- 11 = Reserved; do not use
- 10 = Interrupt when a character is transferred to the Transmit Shift Register (TSR) and as a result, the transmit buffer becomes empty
- 01 = Interrupt when the last character is shifted out of the Transmit Shift Register; all transmit operations are completed
- 00 = Interrupt when a character is transferred to the Transmit Shift Register (this implies there is at least one character open in the transmit buffer)

bit 14 UTXINV: IrDA[®] Encoder Transmit Polarity Inversion bit⁽¹⁾

IREN = 0:

	1 = UxTX is Idle '0'
	0 = UxTX is Idle '1'
	IREN = 1:
	1 = UxTX is Idle '1'
	0 = UxTX is Idle '0'
bit 12	Unimplemented: Read as '0'
bit 11	UTXBRK: Transmit Break bit
	 1 = Send Sync Break on next transmission – Start bit, followed by twelve '0' bits, followed by Stop bit; cleared by hardware upon completion
	0 = Sync Break transmission is disabled or completed
bit 10	UTXEN: Transmit Enable bit ⁽²⁾
	1 = Transmit is enabled, UxTX pin controlled by UARTx
	0 = Transmit is disabled, any pending transmission is aborted and the buffer is reset; UxTX pin is controlled by port.
bit 9	UTXBF: Transmit Buffer Full Status bit (read-only)
	1 = Transmit buffer is full
	0 = Transmit buffer is not full, at least one more character can be written
bit 8	TRMT: Transmit Shift Register Empty bit (read-only)
	1 = Transmit Shift Register is empty and transmit buffer is empty (the last transmission has completed)
	0 = Transmit Shift Register is not empty, a transmission is in progress or queued

Note 1: Value of bit only affects the transmit properties of the module when the $IrDA^{\mathbb{R}}$ encoder is enabled (IREN = 1).

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

20.0 REAL-TIME CLOCK AND CALENDAR (RTCC)

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 29. "Real-Time Clock and Calendar (RTCC)" (DS39696). The information in this data sheet supersedes the information in the FRM.

The Real-Time Clock and Calendar (RTCC) provides a function that can be calibrated.

Key features of the RTCC module are:

- · Operates in Sleep mode
- Provides hours, minutes and seconds using 24-hour format
- FIGURE 20-1: RTCC BLOCK DIAGRAM

- · Visibility of half of one second period
- Provides calendar weekday, date, month and year
- Alarm configurable for half a second, one second, 10 seconds, one minute, 10 minutes, one hour, one day, one week, one month or one year
- · Alarm repeat with decrementing counter
- Alarm with indefinite repeat chime
- Year, 2000 to 2099, leap year correction
- BCD format for smaller software overhead
- · Optimized for long-term battery operation
- User calibration of the 32.768 kHz clock crystal/32K INTRC frequency with periodic auto-adjust
 - Calibration to within ±2.64 seconds error per month
 - Calibrates up to 260 ppm of crystal error



20.1 RTCC Module Registers

The RTCC module registers are organized into three categories:

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

20.1.1 REGISTER MAPPING

To limit the register interface, the RTCC Timer and Alarm Time registers are accessed through the 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 the RTCVALH byte, the RTCC Pointer value, RTCPTR<1:0> bits, decrement by one until they reach '00'. Once 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

RTCPTR	RTCC Value Register Window			
<1:0>	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 the ALRMVALH byte, the Alarm Pointer value bits, ALRMPTR<1:0>, decrement 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.

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, they will decrement the ALRMPTR<1:0> value. 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.

20.1.2 WRITE LOCK

In order to perform a write to any of the RTCC Timer registers, the RTCWREN (RCFGCAL<13>) bit 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 1 instruction cycle time window allowed between the unlock sequence and the setting of RTCWREN; therefore, it is recommended that code follow the procedure in Example 20-1.

For applications written in C, the unlock sequence should be implemented using in-line assembly.

EXAMPLE 20-1: SETTING THE RTCWREN BIT

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

REGISTER 21-2: C	RCCON2: CRC	CONTROL 2	REGISTER
------------------	-------------	------------------	----------

U-0	U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	—	—	DWIDTH4	DWIDTH3	DWIDTH2	DWIDTH1	DWIDTH0
bit 15						•	bit 8
U-0	U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
_	—	—	PLEN4	PLEN3	PLEN2	PLEN1	PLEN0
bit 7				•		•	bit 0
Legend:							
R = Readable bit W = Writable bit U = Unimplemented bit, read as '0'			l as '0'				
-n = Value at	POR	'1' = Bit is set		'0' = Bit is cleared x = Bit is unknown		iown	

bit 15-13	Unimplemented: Read as '0'
bit 12-8	DWIDTH<4:0>: Data Word Width Configuration bits
	Configures the width of the data word (data word width -1).
bit 7-5	Unimplemented: Read as '0'
bit 4-0	PLEN<4:0>: Polynomial Length Configuration bits
	Configures the length of the polynomial (polynomial length – 1).

REGISTER 21-3: CRCXORL: CRC XOR POLYNOMIAL REGISTER, LOW BYTE

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
X15	X14	X13	X12	X11	X10	X9	X8
bit 15			L				bit 8
R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	U-0
X7	X6	X5	X4	X3	X2	X1	—
bit 7							bit 0
Legend:							
R = Readable	e bit	W = Writable I	bit	t U = Unimplemented bit, read as '0'			
-n = Value at	= Value at POR '1' = Bit is set		'0' = Bit is cleared x = Bit is unknown			nown	

bit 15-1 X<15:1>: XOR of Polynomial Term xⁿ Enable bits

bit 0 Unimplemented: Read as '0'

REGISTER 22-18: G1DPH: DISPLAY BUFFER HEIGHT REGISTER

U-0	U-0	U-0	U-0	U-0	R/W-0	R/W-0	R/W-0
—	—	—	—	—	DPH10	DPH9	DPH8
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
DPH7	DPH6	DPH5	DPH4	DPH3	DPH2	DPH1	DPH0
bit 7							bit 0
Legend:							
R = Readable	e bit	W = Writable I	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 15-11 Unimplemented: Read as '0'

bit 10-0 **DPH<10:0>:** Display Frame Height bits (in pixels)

REGISTER 22-19: G1DPWT: DISPLAY TOTAL WIDTH REGISTER

U-0	U-0	U-0	U-0	U-0	R/W-0	R/W-0	R/W-0
—	—	—	—	—	DPWT10	DPWT9	DPWT8
bit 15							bit 8

| R/W-0 |
|-------|-------|-------|-------|-------|-------|-------|-------|
| DPWT7 | DPWT6 | DPWT5 | DPWT4 | DPWT3 | DPWT2 | DPWT1 | DPWT0 |
| bit 7 | | | | | | | bit 0 |

Legend:			
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 15-11 Unimplemented: Read as '0'

bit 10-0 **DPWT<10:0>:** Display Total Width bits (in pixels)

REGISTER 22-20: G1DPHT: DISPLAY TOTAL HEIGHT REGISTER

U-0	U-0	U-0	U-0	U-0	R/W-0	R/W-0	R/W-0	
—	—	_	—	—	DPHT10	DPHT9	DPHT8	
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	
DPHT7	DPHT6	DPHT5	DPHT4	DPHT3	DPHT2	DPHT1	DPHT0	
bit 7 bit 0								
Legend:								
R = Readable bit W = Writable bit		bit	U = Unimplemented bit, read as '0'					
-n = Value at	POR	'1' = Bit is set		'0' = Bit is cleared x = Bit is		x = Bit is unkr	nknown	

bit 15-11 Unimplemented: Read as '0'

bit 10-0 **DPHT<10:0>:** Display Total Height bits (in pixels)

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
01	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
CPO	CPO	f	Compare f with 0x0000	1	1	C DC N OV Z
010	CPO	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
01.2	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 29-2: INSTRUCTION SET OVERVIEW (CONTINUED)

64-Lead Plastic Quad Flat, No Lead Package (MR) – 9x9x0.9 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 Limits		MIN	NOM	MAX
Number of Pins	N	64		
Pitch	е	0.50 BSC		
Overall Height	A	0.80	0.90	1.00
Standoff	A1	0.00	0.02	0.05
Contact Thickness	A3	0.20 REF		
Overall Width	E	9.00 BSC		
Exposed Pad Width	E2	7.05	7.15	7.50
Overall Length	D	9.00 BSC		
Exposed Pad Length	D2	7.05	7.15	7.50
Contact Width	b	0.18	0.25	0.30
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 C04-149B Sheet 2 of 2

PRODUCT IDENTIFICATION SYSTEM

To order or obtain information, e.g., on pricing or delivery, refer to the factory or the listed sales office.

Microchip Trader Architecture — Flash Memory Fa Program Memory Product Group Pin Count — Tape and Reel Fla Temperature Ran Package — Pattern —	PIC 24 FJ 256 DA2 10 T - 1 / PT - XXX nark	Exa a) b) c)	 mples: PIC24FJ128DA206-I/PT: PIC24F device with Graphics Controller and USB On-The-Go, 128-KB program memory, 96-KB data memory, 64-pin, Industrial temp., TQFP package. PIC24FJ256DA110-I/PT: PIC24F device with Graphics Controller and USB On-The-Go, 256-KB program memory, 24-KB data memory, 100-pin, Industrial temp., TQFP package. PIC24FJ256DA210-I/BG: PIC24F device with Graphics Controller and USB On-The-Go, 256-KB program memory, 96-KB data memory, 121-pin, Industrial temp., BGA package. 	
Architecture	24 = 16-bit modified Harvard without DSP			
Flash Memory Family	FJ = Flash program memory			
Product Group	DA2 = General purpose microcontrollers with Graphics Controller and USB On-The-Go			
Pin Count	06 = 64-pin 10 = 100-pin (TQFP)/121-pin (BGA)			
Temperature Range	I = -40° C to $+85^{\circ}$ C (Industrial)			
Package	PT = 100-lead (12x12x1 mm) TQFP (Thin Quad Flatpack) PT = 64-lead, TQFP (Thin Quad Flatpack) MR = 64-lead (9x9x0.9 mm) QFN (Quad Flatpack, No Lead) BG = 121-pin BGA package			
Pattern	Three-digit QTP, SQTP, Code or Special Requirements (blank otherwise) ES = Engineering Sample			