

#### 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·XF

Product Status	Active
Core Processor	PIC
Core Size	16-Bit
Speed	32MHz
Connectivity	I <sup>2</sup> C, IrDA, SPI, UART/USART
Peripherals	Brown-out Detect/Reset, POR, PWM, WDT
Number of I/O	24
Program Memory Size	16KB (5.5K x 24)
Program Memory Type	FLASH
EEPROM Size	512 x 8
RAM Size	1.5K x 8
Voltage - Supply (Vcc/Vdd)	1.8V ~ 3.6V
Data Converters	A/D 9x10b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 85°C (TA)
Mounting Type	Surface Mount
Package / Case	28-SSOP (0.209", 5.30mm Width)
Supplier Device Package	28-SSOP
Purchase URL	https://www.e-xfl.com/product-detail/microchip-technology/pic24f16ka102-i-ss

Email: info@E-XFL.COM

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

## Pin Diagrams (Continued)



## 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  $\mu$ 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  $\mu$ F to 0.001  $\mu$ 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  $\mu$ F in parallel with 0.001  $\mu$ 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  $\mu$ F to 47  $\mu$ 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.

## 3.0 CPU

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 on the CPU, refer to the *"PIC24F Family Reference Manual"*, Section 2. "CPU" (DS39703).

The PIC24F CPU has a 16-bit (data) modified Harvard architecture with an enhanced instruction set and a 24-bit instruction word with a variable length opcode field. The Program Counter (PC) is 23 bits wide and addresses up to 4M instructions of user program memory space. A single-cycle instruction prefetch mechanism is used to help maintain throughput and provides predictable execution. All instructions execute in a single cycle, with the exception of instructions that change the program flow, the double-word move (MOV.D) instruction and the table instructions. Overhead-free program loop constructs are supported using the REPEAT instructions, which are interruptible at any point.

PIC24F devices have sixteen, 16-bit working registers in the programmer's model. Each of the working registers can act as a data, address or address offset register. The 16<sup>th</sup> working register (W15) operates as a Software Stack Pointer (SSP) for interrupts and calls.

The upper 32 Kbytes of the data space memory map can optionally be mapped into program space at any 16K word boundary of either program memory or data EEPROM memory defined by the 8-bit Program Space Visibility Page Address (PSVPAG) register. The program to data space mapping feature lets any instruction access program space as if it were data space.

The Instruction Set Architecture (ISA) has been significantly enhanced beyond that of the PIC18, but maintains an acceptable level of backward compatibility. All PIC18 instructions and addressing modes are supported, either directly, or through simple macros. Many of the ISA enhancements have been driven by compiler efficiency needs.

The core supports Inherent (no operand), Relative, Literal, Memory Direct and three groups of addressing modes. All modes support Register Direct and various Register Indirect modes. Each group offers up to seven addressing modes. Instructions are associated with predefined addressing modes depending upon their functional requirements. For most instructions, the core is capable of executing a data (or program data) memory read, a working register (data) read, a data memory write and a program (instruction) memory read per instruction cycle. As a result, three parameter instructions can be supported, allowing trinary operations (that is, A + B = C) to be executed in a single cycle.

A high-speed, 17-bit by 17-bit multiplier has been included to significantly enhance the core arithmetic capability and throughput. The multiplier supports Signed, Unsigned and Mixed mode, 16-bit by 16-bit or 8-bit by 8-bit integer multiplication. All multiply instructions execute in a single cycle.

The 16-bit ALU has been enhanced with integer divide assist hardware that supports an iterative non-restoring divide algorithm. It operates in conjunction with the REPEAT instruction looping mechanism and a selection of iterative divide instructions to support 32-bit (or 16-bit), divided by 16-bit integer signed and unsigned division. All divide operations require 19 cycles to complete but are interruptible at any cycle boundary.

The PIC24F has a vectored exception scheme with up to eight sources of non-maskable traps and up to 118 interrupt sources. Each interrupt source can be assigned to one of seven priority levels.

A block diagram of the CPU is illustrated in Figure 3-1.

## 3.1 Programmer's Model

Figure 3-2 displays the programmer's model for the PIC24F. All registers in the programmer's model are memory mapped and can be manipulated directly by instructions.

Table 3-1 provides a description of each register. All registers associated with the programmer's model are memory mapped.

## 3.2 CPU Control Registers

### REGISTER 3-1: SR: ALU STATUS REGISTER

U-0	U-0	U-0	U-0	U-0	U-0	U-0	R/W-0, HSC
—	—	—	_	—	—	—	DC
bit 15							bit 8
R/W-0 HSC(1)	R/W-0 HSC(1)	R/W-0 HSC(1)	R-0 HSC	R/W-0 HSC	R/W-0 HSC	R/W-0 HSC	R/W-0 HSC

R/W-0, HSC <sup>(1)</sup>	R/W-0, HSC <sup>(1)</sup>	R/W-0, HSC <sup>(1)</sup>	R-0, HSC	R/W-0, HSC	R/W-0, HSC	R/W-0, HSC	R/W-0, HSC
IPL2 <sup>(2)</sup>	IPL1 <sup>(2)</sup>	IPL0 <sup>(2)</sup>	RA	N	OV	Z	С
bit 7							bit 0

Legend:	HSC = Hardware Settable/Clearable bit					
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 15-9	Unimplemented: Read as '0'
bit 8	DC: ALU Half Carry/Borrow bit
	<ul> <li>1 = A carry-out from the 4<sup>th</sup> low-order bit (for byte-sized data) or 8<sup>th</sup> low-order bit (for word-sized data) of the result occurred</li> </ul>
	0 = No carry-out from the 4 <sup>th</sup> or 8 <sup>th</sup> low-order bit of the result has occurred
bit 7-5	IPL<2:0>: CPU Interrupt Priority Level Status bits <sup>(1,2)</sup>
	111 = CPU interrupt priority level is 7 (15); user interrupts disabled
	110 = CPU interrupt priority level is 5 (14) 101 = CPU Interrupt priority level is 5 (13)
	100 = CPU interrupt priority level is 4 (12)
	011 = CPU interrupt priority level is 3 (11)
	010 = CPU interrupt priority level is 2 (10)
	001 = CPU interrupt priority level is 1 (9)
	000 = CPU interrupt priority level is 0 (8)
bit 4	RA: REPEAT Loop Active bit
	1 = REPEAT loop in progress
	0 = REPEAT loop not in progress
bit 3	N: ALU Negative bit
	1 = Result was negative
	0 = Result was non-negative (zero or positive)
bit 2	OV: ALU Overflow bit
	<ul> <li>1 = Overflow occurred for signed (2's complement) arithmetic in this arithmetic operation</li> <li>0 = No overflow has occurred</li> </ul>
bit 1	Z: ALU Zero bit
	<ul> <li>1 = An operation, which effects the Z bit, has set it at some time in the past</li> <li>0 = The most recent operation, which effects the Z bit, has cleared it (i.e., a non-zero result)</li> </ul>
bit 0	C: ALU Carry/Borrow bit
	1 = A carry-out from the Most Significant bit (MSb) of the result occurred
	0 = No carry-out from the Most Significant bit (MSb) of the result occurred
Note 1:	The IPL Status bits are read-only when NSTDIS (INTCON1<15>) = 1.
2:	The IPL Status bits are concatenated with the IPL3 bit (CORCON<3>) to form the CPU Interrupt Priority Level (IPL). The value in parentheses indicates the IPL when IPL3 = 1.

## 6.3 NVM Address Register

As with Flash program memory, the NVM Address Registers, NVMADRU and NVMADR, form the 24-bit Effective Address (EA) of the selected row or word for data EEPROM operations. The NVMADRU register is used to hold the upper 8 bits of the EA, while the NVMADR register is used to hold the lower 16 bits of the EA. These registers are not mapped into the Special Function Register (SFR) space; instead, they directly capture the EA<23:0> of the last table write instruction that has been executed and selects the data EEPROM row to erase. Figure 6-1 depicts the program memory EA that is formed for programming and erase operations. Like program memory operations, the Least Significant bit (LSb) of NVMADR is restricted to even addresses. This is because any given address in the data EEPROM space consists of only the lower word of the program memory width; the upper word, including the uppermost "phantom byte", are unavailable. This means that the LSb of a data EEPROM address will always be '0'.

Similarly, the Most Significant bit (MSb) of NVMADRU is always '0', since all addresses lie in the user program space.

### FIGURE 6-1: DATA EEPROM ADDRESSING WITH TBLPAG AND NVM ADDRESS REGISTERS



## 6.4 Data EEPROM Operations

The EEPROM block is accessed using table read and write operations, similar to those used for program memory. The TBLWTH and TBLRDH instructions are not required for data EEPROM operations, since the memory is only 16 bits wide (data on the lower address is valid only). The following programming operations can be performed on the data EEPROM:

- Erase one, four or eight words
- Bulk erase the entire data EEPROM
- Write one word
- Read one word

#### Note 1: Unexpected results will be obtained should the user attempt to read the EEPROM while a programming or erase operation is underway.

2: The C30 C compiler includes library procedures to automatically perform the table read and table write operations, manage the Table Pointer and write buffers, and unlock and initiate memory write sequences. This eliminates the need to create assembler macros or time critical routines in C for each application.

The library procedures are used in the code examples detailed in the following sections. General descriptions of each process are provided for users who are not using the C30 compiler libraries.

#### 6.4.1.1 Data EEPROM Bulk Erase

To erase the entire data EEPROM (bulk erase), the address registers do not need to be configured because this operation affects the entire data EEPROM. The following sequence helps in performing bulk erase:

- 1. Configure NVMCON to Bulk Erase mode.
- 2. Clear NVMIF status bit and enable NVM interrupt (optional).
- 3. Write the key sequence to NVMKEY.
- 4. Set the WR bit to begin erase cycle.
- 5. Either poll the WR bit or wait for the NVM interrupt (NVMIF is set).

A typical bulk erase sequence is provided in Example 6-3.

#### 6.4.2 SINGLE-WORD WRITE

To write a single word in the data EEPROM, the following sequence must be followed:

- Erase one data EEPROM word (as mentioned in Section 6.4.1 "Erase Data EEPROM") if the PGMONLY bit (NVMCON<12>) is set to '1'.
- 2. Write the data word into the data EEPROM latch.
- 3. Program the data word into the EEPROM:
  - Configure the NVMCON register to program one EEPROM word (NVMCON<5:0> = 0001xx).
  - Clear NVMIF status bit and enable NVM interrupt (optional).
  - Write the key sequence to NVMKEY.
  - Set the WR bit to begin erase cycle.
  - Either poll the WR bit or wait for the NVM interrupt (NVMIF is set).
  - To get cleared, wait until NVMIF is set.

A typical single-word write sequence is provided in Example 6-4.

### EXAMPLE 6-3: DATA EEPROM BULK ERASE

// Set up NVMCON to bulk erase the data EEPROM NVMCON =  $0 \times 4050;$ 

// Disable Interrupts For 5 Instructions
asm volatile ("disi #5");

// Issue Unlock Sequence and Start Erase Cycle
\_\_builtin\_write\_NVM();

### EXAMPLE 6-4: SINGLE-WORD WRITE TO DATA EEPROM

<pre>intattribute ((space(eedata))) eeData = 0x1234;</pre>	<pre>// Variable located in EEPROM,declared as a global variable.</pre>
int newData;	// New data to write to EEPROM
unsigned int offset;	
// Set up NVMCON to erase one word of data EEPROM	
NVMCON = 0x4004;	
// Set up a pointer to the EEPROM location to be en	rased
<pre>TBLPAG =builtin_tblpage(&amp;eeData);</pre>	// Initialize EE Data page pointer
<pre>offset =builtin_tbloffset(&amp;eeData);</pre>	// Initizlize lower word of address
builtin_tblwtl(offset, newData);	// Write EEPROM data to write latch
asm volatile ("disi #5");	// Disable Interrupts For 5 Instructions
builtin_write_NVM();	// Issue Unlock Sequence & Start Write Cycle

## 9.1 CPU Clocking Scheme

The system clock source can be provided by one of four sources:

- Primary Oscillator (POSC) on the OSCI and OSCO pins
- Secondary Oscillator (SOSC) on the SOSCI and SOSCO pins

The PIC24F16KA102 family devices consist of two types of secondary oscillator:

- High-Power Secondary Oscillator
- Low-Power Secondary Oscillator

These can be selected by using the SOSCSEL (FOSC<5>) bit.

- · Fast Internal RC (FRC) Oscillator
  - 8 MHz FRC Oscillator
  - 500 kHz Lower Power FRC Oscillator
- · Low-Power Internal RC (LPRC) Oscillator

The primary oscillator and 8 MHz FRC sources have the option of using the internal 4x PLL. The frequency of the FRC clock source can optionally be reduced by the programmable clock divider. The selected clock source generates the processor and peripheral clock sources.

The processor clock source is divided by two to produce the internal instruction cycle clock, FcY. In this document, the instruction cycle clock is also denoted by Fosc/2. The internal instruction cycle clock, Fosc/2, can be provided on the OSCO I/O pin for some operating modes of the primary oscillator.

## 9.2 Initial Configuration on POR

The oscillator source (and operating mode) that is used at a device Power-on Reset (POR) event is selected using Configuration bit settings. The Oscillator Configuration bit settings are located in the Configuration registers in the program memory (refer to Section 26.1 "Configuration Bits" for further details). The Primary Oscillator POSCMD<1:0> Configuration bits, (FOSC<1:0>), and the Initial Oscillator Select Configuration bits, FNOSC<2:0> (FOSCSEL<2:0>), select the oscillator source that is used at a POR. The FRC Primary Oscillator with Postscaler (FRCDIV) is the default (unprogrammed) selection. The secondary oscillator, or one of the internal oscillators, may be chosen by programming these bit locations. The EC mode frequency range Configuration bits, POSCFREQ<1:0> (FOSC<4:3>), optimize power consumption when running in EC mode. The default configuration is "frequency range is greater than 8 MHz".

The Configuration bits allow users to choose between the various clock modes, shown in Table 9-1.

#### 9.2.1 CLOCK SWITCHING MODE CONFIGURATION BITS

The FCKSM Configuration bits (FOSC<7:6>) are used jointly to configure device clock switching and the FSCM. Clock switching is enabled only when FCKSM1 is programmed ('0'). The FSCM is enabled only when FCKSM<1:0> are both programmed ('00').

Oscillator Mode	Oscillator Source	POSCMD<1:0>	FNOSC<2:0>	Note
8 MHz FRC Oscillator with Postscaler (FRCDIV)	Internal	11	111	1, 2
500 MHz FRC Oscillator with Postscaler (LPFRCDIV)	Internal	11	110	1
Low-Power RC Oscillator (LPRC)	Internal	11	101	1
Secondary (Timer1) Oscillator (SOSC)	Secondary	00	100	1
Primary Oscillator (HS) with PLL Module (HSPLL)	Primary	10	011	
Primary Oscillator (EC) with PLL Module (ECPLL)	Primary	00	011	
Primary Oscillator (HS)	Primary	10	010	
Primary Oscillator (XT)	Primary	01	010	
Primary Oscillator (EC)	Primary	00	010	
8 MHz FRC Oscillator with PLL Module (FRCPLL)	Internal	11	001	1
8 MHz FRC Oscillator (FRC)	Internal	11	000	1

#### TABLE 9-1: CONFIGURATION BIT VALUES FOR CLOCK SELECTION

Note 1: OSCO pin function is determined by the OSCIOFNC Configuration bit.

2: This is the default oscillator mode for an unprogrammed (erased) device.

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-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
	_	TUN5 <sup>(1)</sup>	TUN4 <sup>(1)</sup>	TUN3 <sup>(1)</sup>	TUN2 <sup>(1)</sup>	TUN1 <sup>(1)</sup>	TUN0 <sup>(1)</sup>
bit 7	·	•			•	•	bit 0
Legend:							
R = Readable	e bit	W = Writable	bit	U = Unimplem	nented bit, read	l as '0'	
-n = Value at	POR	'1' = Bit is set		'0' = Bit is cleared		x = Bit is unkr	iown
bit 15-6	Unimplemen	ted: Read as '	כ'				
bit 5-0	<b>TUN&lt;5:0&gt;:</b> F	RC Oscillator T	uning bits <sup>(1)</sup>				
	011111 <b>= Ma</b>	ximum frequer	icy deviation				
	011110						
	•						
	•						
	000001						
	000000 = Ce	nter frequency,	oscillator is ru	inning at factory	/ calibrated free	quency	
	111111						
	•						
	•						
	100001						
	100000 <b>= M</b> ir	nimum frequen	cy deviation				

#### REGISTER 9-3: OSCTUN: FRC OSCILLATOR TUNE REGISTER

**Note 1:** Increments or decrements of TUN<5:0> may not change the FRC frequency in equal steps over the FRC tuning range and may not be monotonic.

## 12.0 TIMER1

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 on Timers, refer to the "PIC24F Family Reference Manual", Section 14. "Timers" (DS39704).

The Timer1 module is a 16-bit timer which can serve as the time counter for the Real-Time Clock (RTC), or operate as a free-running, interval timer/counter. Timer1 can operate in three modes:

- 16-Bit Timer
- 16-Bit Synchronous Counter
- 16-Bit Asynchronous Counter

Timer1 also supports these features:

- Timer Gate Operation
- Selectable Prescaler Settings
- Timer Operation During CPU Idle and Sleep modes
- Interrupt on 16-Bit Period Register Match or Falling Edge of External Gate Signal

Figure 12-1 presents a block diagram of the 16-bit Timer1 module.

To configure Timer1 for operation:

- 1. Set the TON bit (= 1).
- 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.
- 4. Set or clear the TSYNC bit to configure synchronous or asynchronous operation.
- 5. Load the timer period value into the PR1 register.
- 6. If interrupts are required, set the interrupt enable bit, T1IE. Use the priority bits, T1IP<2:0>, to set the interrupt priority.



## FIGURE 12-1: 16-BIT TIMER1 MODULE BLOCK DIAGRAM

## 13.0 TIMER2/3

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 on Timers, refer to the "PIC24F Family Reference Manual", Section 14. "Timers" (DS39704).

The Timer2/3 module is a 32-bit timer, which can also be configured as two independent 16-bit timers with selectable operating modes.

As a 32-bit timer, Timer2/3 operates in three modes:

- Two independent 16-bit timers (Timer2 and Timer3) 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
- A/D Event Trigger

Individually, both of the 16-bit timers can function as synchronous timers or counters. They also offer the features listed above, except for the A/D event trigger (this is implemented only with Timer3). The operating modes and enabled features are determined by setting the appropriate bit(s) in the T2CON and T3CON registers. T2CON and T3CON are provided in generic form in Register 13-1 and Register 13-2, respectively.

For 32-bit timer/counter operation, Timer2 is the least significant word (lsw) and Timer3 is the most significant word (msw) of the 32-bit timer.

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

To configure Timer2/3 for 32-bit operation:

- 1. Set the T32 bit (T2CON<3> = 1).
- 2. Select the prescaler ratio for Timer2 using the TCKPS<1:0> bits.
- 3. Set the Clock and Gating modes using the TCS and TGATE bits.
- 4. Load the timer period value. PR3 will contain the msw of the value while PR2 contains the lsw.
- 5. If interrupts are required, set the interrupt enable bit, T3IE. Use the priority bits, T3IP<2:0>, to set the interrupt priority.

While Timer2 controls the timer, the interrupt appears as a Timer3 interrupt.

6. Set the TON bit (= 1).

The timer value, at any point, is stored in the register pair, TMR<3:2>. TMR3 always contains the msw of the count, while TMR2 contains the lsw.

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

- 1. Clear the T32 bit in T2CON<3>.
- 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.
- 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 16-2: SPI1CON1: SPI1 CONTROL REGISTER 1 (CONTINUED)

bit 1-0 **PPRE<1:0>:** Primary Prescale bits (Master mode)

- 11 = Primary prescale 1:1
- 10 = Primary prescale 4:1
- 01 = Primary prescale 16:1
- 00 = Primary prescale 64:1
- **Note 1:** The CKE bit is not used in the Framed SPI modes. The user should program this bit to '0' for the Framed SPI modes (FRMEN = 1).

#### REGISTER 16-3: SPI1CON2: SPI1 CONTROL REGISTER 2

R/W-0	R/W-0	R/W-0	U-0	U-0	U-0	U-0	U-0		
FRMEN	SPIFSD	SPIFPOL	_	_	_	_			
bit 15					•		bit 8		
U-0	U-0	U-0	U-0	U-0	U-0	R/W-0	R/W-0		
_	_	_	_	_	_	SPIFE	SPIBEN		
bit 7						·	bit 0		
Legend:									
R = Readable	e bit	W = Writable b	it	U = Unimplem	nented bit, rea	ad as '0'			
-n = Value at	POR	'1' = Bit is set		'0' = Bit is clea	ared	x = Bit is unkı	nown		
bit 15	FRMEN: Frai	med SPI1 Suppo	rt bit						
	1 = Framed S	SPI1 support is e	nabled						
	0 = Framed S	SPI1 support is d	isabled						
bit 14	SPIFSD: Fra	me Sync Pulse D	irection Cor	ntrol on SS1 Pin	bit				
	1 = Frame sy	nc pulse input (s	lave)						
h:+ 40			Delevity bit		.h. A				
DIC 13	SPIFPUL: Fr	ame Sync Pulse	Polarity bit	(Frame mode on	liy)				
	1 = Frame sy 0 = Frame sy	inc pulse is active	e-low						
bit 12-2	Unimplemen	ted: Read as '0'							
bit 1	SPIFE: Fram	e Svnc Pulse Ed	ae Select bi	t					
	1 = Frame sy	nc pulse coincide	es with the f	irst bit clock					
	0 = Frame sy	nc pulse precede	es the first b	it clock					
	SPIBEN: Enhanced Buffer Enable bit								
bit 0	SPIBEN: Enh	nanced Buffer En	able bit						
bit 0	SPIBEN: Enh	nanced Buffer En d Buffer is enable	able bit ed						

## 18.0 UNIVERSAL ASYNCHRONOUS RECEIVER TRANSMITTER (UART)

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 on the Universal Asynchronous Receiver Transmitter, refer to the *"PIC24F Family Reference Manual"*, Section 21. "UART" (DS39708).

The Universal Asynchronous Receiver Transmitter (UART) module is one of the serial I/O modules available in this PIC24F device family. The UART is a full-duplex asynchronous system that can communicate with peripheral devices, such as personal computers, LIN, RS-232 and RS-485 interfaces. This module also supports a hardware flow control option with the UxCTS and UxRTS pins, and also includes an IrDA<sup>®</sup> encoder and decoder.

The primary features of the UART module are:

- Full-Duplex, 8-Bit or 9-Bit Data Transmission through the UxTX and UxRX Pins
- Even, Odd or No Parity Options (for 8-bit data)
- One or Two Stop bits
- Hardware Flow Control Option with UxCTS and UxRTS pins

- Fully Integrated Baud Rate Generator (IBRG) with 16-Bit Prescaler
- Baud Rates Ranging from 1 Mbps to 15 bps at 16 MIPS
- 4-Deep, First-In-First-Out (FIFO) Transmit Data Buffer
- · 4-Deep FIFO Receive Data Buffer
- Parity, Framing and Buffer Overrun Error Detection
- Support for 9-Bit mode with Address Detect (9<sup>th</sup> bit = 1)
- Transmit and Receive Interrupts
- Loopback mode for Diagnostic Support
- Support for Sync and Break Characters
- Supports Automatic Baud Rate Detection
- IrDA Encoder and Decoder Logic
- 16x Baud Clock Output for IrDA Support

A simplified block diagram of the UART is displayed in Figure 18-1. The UART module consists of these important hardware elements:

- · Baud Rate Generator
- Asynchronous Transmitter
- · Asynchronous Receiver

#### FIGURE 18-1: UART SIMPLIFIED BLOCK DIAGRAM



#### 19.2.6 ALRMVAL REGISTER MAPPINGS

## REGISTER 19-8: ALMTHDY: ALARM MONTH AND DAY VALUE REGISTER<sup>(1)</sup>

U-0	U-0	U-0	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x	
_	—	—	MTHTEN0	MTHONE3	MTHONE2	MTHONE1	MTHONE0	
bit 15				•			bit 8	
U-0	U-0	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x	
_	—	DAYTEN1	DAYTEN0	DAYONE3	DAYONE2	DAYONE1	DAYONE0	
bit 7							bit 0	
Legend:								
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			iown	
bit 15-13	Unimplemen	ted: Read as '0'						
bit 12	MTHTENO: E	Binary Coded De	ecimal Value of	f Month's Tens	Digit bit			
	Contains a va	alue of '0' or '1'.						
bit 11-8	MTHONE<3:	:0>: Binary Cod	ed Decimal Va	lue of Month's (	Ones Digit bits			
	Contains a va	alue from 0 to 9			-			
bit 7-6	Unimplemer	nted: Read as 'd	)'					
bit 5-4	DAYTEN<1:	0>: Binary Code	d Decimal Val	ue of Day's Ten	s Digit bits			
	Contains a va	alue from 0 to 3			-			
bit 3-0	DAYONE<3:	0>: Binary Code	ed Decimal Val	ue of Day's On	es Digit bits			
	Contains a va	alue from 0 to 9		-	-			

**Note 1:** A write to this register is only allowed when RTCWREN = 1.

## REGISTER 19-9: ALWDHR: ALARM WEEKDAY AND HOURS VALUE REGISTER<sup>(1)</sup>

U-0	U-0	U-0	U-0	U-0	R/W-x	R/W-x	R/W-x
		—		—	WDAY2	WDAY1	WDAY0
bit 15	•		•				bit 8
U-0	U-0	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x
—	—	HRTEN1	HRTEN0	HRONE3	HRONE2	HRONE1	HRONE0
bit 7	•		•				bit 0
1							

Legend:			
R = Readable bit	W = Writable bit	U = Unimplemented bit, read	d 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 10-8	WDAY<2:0>: Binary Coded Decimal Value of Weekday Digit bits				
	Contains a value from 0 to 6.				
bit 7-6	Unimplemented: Read as '0'				
bit 5-4	HRTEN<1:0>: Binary Coded Decimal Value of Hour's Tens Digit bits				
	Contains a value from 0 to 2.				
bit 3-0	HRONE<3:0>: Binary Coded Decimal Value of Hour's Ones Digit bits				
	Contains a value from 0 to 9.				

**Note 1:** A write to this register is only allowed when RTCWREN = 1.

#### REGISTER 20-2: CRCXOR: CRC XOR POLYNOMIAL REGISTER

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							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	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 15-1 X<15:1>: XOR of Polynomial Term X<sup>n</sup> Enable bits

bit 0 Unimplemented: Read as '0'

### 27.11 PICkit 2 Development Programmer/Debugger and PICkit 2 Debug Express

The PICkit<sup>™</sup> 2 Development Programmer/Debugger is a low-cost development tool with an easy to use interface for programming and debugging Microchip's Flash families of microcontrollers. The full featured Windows<sup>®</sup> programming interface supports baseline (PIC10F, PIC12F5xx, PIC16F5xx), midrange (PIC12F6xx, PIC16F), PIC18F, PIC24, dsPIC30, dsPIC33, and PIC32 families of 8-bit, 16-bit, and 32-bit microcontrollers, and many Microchip Serial EEPROM products. With Microchip's powerful MPLAB Integrated Development Environment (IDE) the PICkit<sup>™</sup> 2 enables in-circuit debugging on most PIC<sup>®</sup> microcontrollers. In-Circuit-Debugging runs, halts and single steps the program while the PIC microcontroller is embedded in the application. When halted at a breakpoint, the file registers can be examined and modified.

The PICkit 2 Debug Express include the PICkit 2, demo board and microcontroller, hookup cables and CDROM with user's guide, lessons, tutorial, compiler and MPLAB IDE software.

### 27.12 MPLAB PM3 Device Programmer

The MPLAB PM3 Device Programmer is a universal, CE compliant device programmer with programmable voltage verification at VDDMIN and VDDMAX for maximum reliability. It features a large LCD display (128 x 64) for menus and error messages and a modular, detachable socket assembly to support various package types. The ICSP™ cable assembly is included as a standard item. In Stand-Alone mode, the MPLAB PM3 Device Programmer can read, verify and program PIC devices without a PC connection. It can also set code protection in this mode. The MPLAB PM3 connects to the host PC via an RS-232 or USB cable. The MPLAB PM3 has high-speed communications and optimized algorithms for quick programming of large memory devices and incorporates an MMC card for file storage and data applications.

### 27.13 Demonstration/Development Boards, Evaluation Kits, and Starter Kits

A wide variety of demonstration, development and evaluation boards for various PIC MCUs and dsPIC DSCs allows quick application development on fully functional systems. Most boards include prototyping areas for adding custom circuitry and provide application firmware and source code for examination and modification.

The boards support a variety of features, including LEDs, temperature sensors, switches, speakers, RS-232 interfaces, LCD displays, potentiometers and additional EEPROM memory.

The demonstration and development boards can be used in teaching environments, for prototyping custom circuits and for learning about various microcontroller applications.

In addition to the PICDEM<sup>™</sup> and dsPICDEM<sup>™</sup> demonstration/development board series of circuits, Microchip has a line of evaluation kits and demonstration software for analog filter design, KEELOQ<sup>®</sup> security ICs, CAN, IrDA<sup>®</sup>, PowerSmart battery management, SEEVAL<sup>®</sup> evaluation system, Sigma-Delta ADC, flow rate sensing, plus many more.

Also available are starter kits that contain everything needed to experience the specified device. This usually includes a single application and debug capability, all on one board.

Check the Microchip web page (www.microchip.com) for the complete list of demonstration, development and evaluation kits.

Assembly Mnemonic	Assembly Syntax		Description	# of Words	# of Cycles	Status Flags Affected
ADD	ADD f		f = f + WREG	1	1	C, DC, N, OV, Z
	ADD f,WREG		WREG = f + WREG	1	1	C, DC, N, OV, Z
	ADD #lit10,Wn		Wd = lit10 + Wd	1	1	C, DC, N, OV, Z
	ADD Wb,Ws,Wd		Wd = Wb + Ws	1	1	C, DC, N, OV, Z
	ADD	Wb,#lit5,Wd	Wd = Wb + lit5	1	1	C, DC, N, OV, Z
ADDC	ADDC	£	f = f + WREG + (C)	1	1	C, DC, N, OV, Z
	ADDC	f,WREG	WREG = f + WREG + (C)	1	1	C, DC, N, OV, Z
	ADDC	#lit10,Wn	Wd = lit10 + Wd + (C)	1	1	C, DC, N, OV, Z
	ADDC	Wb,Ws,Wd	Wd = Wb + Ws + (C)	1	1	C, DC, N, OV, Z
	ADDC	Wb,#lit5,Wd	Wd = Wb + lit5 + (C)	1	1	C, DC, N, OV, Z
AND	AND	£	f = f .AND. WREG	1	1	N, Z
	AND	f,WREG	WREG = f .AND. WREG	1	1	N, Z
	AND	#lit10,Wn	Wd = lit10 .AND. Wd	1	1	N, Z
	AND	Wb,Ws,Wd	Wd = Wb .AND. Ws	1	1	N, Z
	AND	Wb,#lit5,Wd	Wd = Wb .AND. lit5	1	1	N, Z
ASR	ASR	£	f = Arithmetic Right Shift f	1	1	C, N, OV, Z
	ASR	f,WREG	WREG = Arithmetic Right Shift f	1	1	C, N, OV, Z
	ASR	Ws,Wd	Wd = Arithmetic Right Shift Ws	1	1	C, N, OV, Z
	ASR	Wb,Wns,Wnd	Wnd = Arithmetic Right Shift Wb by Wns	1	1	N, Z
	ASR	Wb,#lit5,Wnd	Wnd = Arithmetic Right Shift Wb by lit5	1	1	N, Z
BCLR	BCLR	f,#bit4	Bit Clear f	1	1	None
	BCLR	Ws,#bit4	Bit Clear Ws	1	1	None
BRA	BRA	C,Expr	Branch if Carry	1	1 (2)	None
	BRA	GE, Expr	Branch if Greater than or Equal	1	1 (2)	None
	BRA	GEU, Expr	Branch if Unsigned Greater than or Equal	1	1 (2)	None
	BRA	GT, Expr	Branch if Greater than	1	1 (2)	None
	BRA	GTU, Expr	Branch if Unsigned Greater than	1	1 (2)	None
	BRA	LE, Expr	Branch if Less than or Equal	1	1 (2)	None
	BRA	LEU, Expr	Branch if Unsigned Less than or Equal	1	1 (2)	None
	BRA	LT, Expr	Branch if Less than	1	1 (2)	None
	BRA	LTU, Expr	Branch if Unsigned Less than	1	1 (2)	None
	BRA	N,Expr	Branch if Negative	1	1 (2)	None
	BRA	NC, Expr	Branch if Not Carry	1	1 (2)	None
	BRA	NN,Expr	Branch if Not Negative	1	1 (2)	None
	BRA	NOV, Expr	Branch if Not Overflow	1	1 (2)	None
	BRA	NZ,Expr	Branch if Not Zero	1	1 (2)	None
	BRA	OV,Expr	Branch if Overflow	1	1 (2)	None
	BRA	Expr	Branch Unconditionally	1	2	None
	BRA	Z,Expr	Branch if Zero	1	1 (2)	None
	BRA	Wn	Computed Branch	1	2	None
BSET	BSET	f,#bit4	Bit Set f	1	1	None
	BSET	Ws,#bit4	Bit Set Ws	1	1	None
BSW	BSW.C	Ws,Wb	Write C bit to Ws <wb></wb>	1	1	None
	BSW.Z	Ws,Wb	Write Z bit to Ws <wb></wb>	1	1	None
BTG	BTG	f,#bit4	Bit Toggle f	1	1	None
	BTG	Ws,#bit4	Bit Toggle Ws	1	1	None
BTSC	BTSC	f,#bit4	Bit Test f, Skip if Clear	1	1 (2 or 3)	None
	BTSC	Ws,#bit4	Bit Test Ws, Skip if Clear	1	1 (2 or 3)	None

TABLE 28-2:	INSTRUCTION SET	OVERVIEW

DC CHARACTERISTICS		$ \begin{array}{llllllllllllllllllllllllllllllllllll$				
Parameter No.	Typical <sup>(1)</sup>	Мах	Units	Conditions		
Power-Down C	Current (IPD): F	PMD Bits are	Set, PMSLP	Bit is '0' <sup>(2)</sup>		
DC62		0.650		-40°C		
DC62a		0.650		+25°C		
DC62b	0.450	0.650	μA	+60°C	1.8V	
DC62c		0.650		+85°C		
DC62d				+125°C		Timer1 w/32 kHz Crystal: T132
DC62e		0.980		-40°C		(SOSC – LP) <sup>(3)</sup>
DC62f		0.980		+25°C		
DC62g	0.730	0.980	μΑ	+60°C	3.3V	
DC62h		0.980		+85°C		
DC62i				+125°C		
DC64		7.10		-40°C		
DC64a		7.10		+25°C		
DC64b	5.5	7.80	μA	+60°C	1.8V	
DC64c		8.30		+85°C		
DC64d		10.00		+125°C		
DC64e		7.10		-40°C		
DC64f		7.10		+25°C		
DC64g	6.2	7.80	μA	+60°C	3.3V	
DC64h		8.30		+85°C		
DC64i		9.00		+125°C		
DC63		6.60		-40°C		
DC63a		6.60		+25°C		
DC63b	4.5	6.60	μA	+60°C	3.3V	BOR <sup>(3,4)</sup>
DC63c		6.60		+85°C		
DC63d		9.00		+125°C		
DC62	_	0.65		-40°C		
DC62a		0.65		+25°C		
DC62b	0.49	0.65	μΑ	+60°C	1.8V	
DC62c	_	0.65		+85°C		
DC62d		0.98		+125°C		BTCC(3,5)
DC62e		0.98		-40°C		
DC62f		0.98	4	+25°C	_	
DC62g	0.80	0.98	μA	+60°C	3.3V	
DC62h		0.98	4	+85°C	_	
DC62i		0.98		+125°C		

#### TABLE 29-8: DC CHARACTERISTICS: POWER-DOWN CURRENT (IPD) (CONTINUED)

Note 1: Data in the "Typical" column is at 3.3V, 25°C unless otherwise stated. Parameters are for design guidance only and are not tested.

2: Base IPD is measured with all peripherals and clocks shut down. All I/Os are configured as outputs and set low. WDT, etc., are all switched off.

3: The ∆ current is the additional current consumed when the module is enabled. This current should be added to the base IPD current.

4: Current applies to Sleep only.

5: Current applies to Sleep and Deep Sleep.

**6:** Current applies to Deep Sleep only.

## APPENDIX A: REVISION HISTORY

## **Revision A (November 2008)**

Original data sheet for the PIC24F16KA102 family of devices.

## **Revision B (March 2009)**

Section 29.0 "Electrical Characteristics" was revised and minor text edits were made throughout the document.

## **Revision C (October 2011)**

- · Changed all instances of DSWSRC to DSWAKE.
- Corrected Example 5-2.
- Corrected Example 5-4.
- Corrected Example 6-1.
- Corrected Example 6-3.
- Added a comment to Example 6-5.
- Corrected Figure 9-1 to connect the SOSCI and SOSCO pins to the Schmitt trigger correctly.
- Added register descriptions for PMD1, PMD2, PMD3 and PMD4.
- Added note that RTCC will run in Reset.
- Corrected time values of ADCS (AD1CON3<5:0>).
- Corrected CH0SB and CH0SA (AD1CHS<11:8> and AD1CHS<3:0>) to correctly reference AVDD and AN3.
- Added description of PGCF15 and PGCF14 (AD1PCFG<15:14>).
- Edited Figure 22-2 to correctly reference RIC and the A/D capacitance.
- Changed all references from CTEDG1 to CTED1.
- Changed all references from CTEDG2 to CTED2.
- Changed description of CMIDL: it used to say it disables all comparators in Idle, now only disables interrupts in Idle mode.
- Changed all references of RTCCKSEL to RTCOSC.
- Changed all references of DSLPBOR to DSBOREN.
- Changed all references of DSWCKSEL to DSWDTOSC
- Imported Figure 40-9 from PIC24F FRM, Section 40.
- Added spec for BOR hysteresis.
- Edited Note 1 for Table 29-5 to further describe LPBOR.
- Edited max values of DC20d and DC20e on Table 29-6.
- Edited typical value for DC61-DC61c in Table 29-8.
- Edited Note 2 of Table 29-8.
- Added Note 5 to Table 29-9.
- Added Table 29-15.
- Added AD08 and AD09 in Table 29-26.
- Added Note 3 to Table 29-26.

- Imported Figure 40.10 from PIC24F FRM, Section 40.
- Deleted TVREG spec.
- Imported Figure 15-5 from PIC24F FRM, Section 15.
- Imported Table 15-4 from PIC24F FRM, Section 15.
- Imported Figure 16-22 from PIC24F FRM, Section 16.
- Imported Table 16-9 from PIC24F FRM, Section 16.
- Imported Figure 16-23 from PIC24F FRM, Section 16.
- Imported Table 16-10 from PIC24F FRM, Section 16.
- Imported Figure 21-24 from PIC24F FRM. Section 21.
- Imported Figure 21-25 from PIC24F FRM, Section 21.
- Imported Table 21-5 from PIC24F FRM, Section 21.
- Imported Figure 23-17 from PIC24F FRM, Section 23.
- Imported Table 23-3 from PIC24F FRM, Section 23.
- Imported Figure 23-18 from PIC24F FRM, Section 23.
- Imported Table 23-4 from PIC24F FRM, Section 23.
- Imported Figure 23-19 from PIC24F FRM, Section 23.
- Imported Table 23-5 from PIC24F FRM, Section 23.
- Imported Figure 23-20 from PIC24F FRM, Section 23.
- Imported Table 23-6 from PIC24F FRM, Section 23.
- Imported Figure 24-33 from PIC24F FRM, Section 24.
- Imported Table 24-6 from PIC24F FRM, Section 24.
- Imported Figure 24-34 from PIC24F FRM, Section 24.
- Imported Table 24-7 from PIC24F FRM, Section 24.
- Imported Figure 24-35 from PIC24F FRM, Section 24.
- Imported Table 24-8 from PIC24F FRM, Section 24.
- Imported Figure 24-36 from PIC24F FRM, Section 24.
- Imported Table 24-9 from PIC24F FRM, Section 24.

## **Worldwide Sales and Service**

#### AMERICAS

Corporate Office 2355 West Chandler Blvd. Chandler, AZ 85224-6199 Tel: 480-792-7200 Fax: 480-792-7277 Technical Support: http://www.microchip.com/ support Web Address: www.microchip.com

Atlanta Duluth, GA Tel: 678-957-9614 Fax: 678-957-1455

Boston Westborough, MA Tel: 774-760-0087 Fax: 774-760-0088

Chicago Itasca, IL Tel: 630-285-0071 Fax: 630-285-0075

**Cleveland** Independence, OH Tel: 216-447-0464 Fax: 216-447-0643

**Dallas** Addison, TX Tel: 972-818-7423 Fax: 972-818-2924

Detroit Farmington Hills, MI Tel: 248-538-2250 Fax: 248-538-2260

Indianapolis Noblesville, IN Tel: 317-773-8323 Fax: 317-773-5453

Los Angeles Mission Viejo, CA Tel: 949-462-9523 Fax: 949-462-9608

Santa Clara Santa Clara, CA Tel: 408-961-6444 Fax: 408-961-6445

Toronto Mississauga, Ontario, Canada Tel: 905-673-0699 Fax: 905-673-6509

#### ASIA/PACIFIC

Asia Pacific Office Suites 3707-14, 37th Floor Tower 6, The Gateway Harbour City, Kowloon Hong Kong Tel: 852-2401-1200 Fax: 852-2401-3431 Australia - Sygney

Tel: 61-2-9868-6733 Fax: 61-2-9868-6755

**China - Beijing** Tel: 86-10-8569-7000 Fax: 86-10-8528-2104

**China - Chengdu** Tel: 86-28-8665-5511 Fax: 86-28-8665-7889

**China - Chongqing** Tel: 86-23-8980-9588 Fax: 86-23-8980-9500

**China - Hangzhou** Tel: 86-571-2819-3187 Fax: 86-571-2819-3189

**China - Hong Kong SAR** Tel: 852-2401-1200 Fax: 852-2401-3431

**China - Nanjing** Tel: 86-25-8473-2460 Fax: 86-25-8473-2470

**China - Qingdao** Tel: 86-532-8502-7355 Fax: 86-532-8502-7205

**China - Shanghai** Tel: 86-21-5407-5533 Fax: 86-21-5407-5066

China - Shenyang Tel: 86-24-2334-2829 Fax: 86-24-2334-2393

**China - Shenzhen** Tel: 86-755-8203-2660 Fax: 86-755-8203-1760

**China - Wuhan** Tel: 86-27-5980-5300 Fax: 86-27-5980-5118

**China - Xian** Tel: 86-29-8833-7252 Fax: 86-29-8833-7256

**China - Xiamen** Tel: 86-592-2388138 Fax: 86-592-2388130

**China - Zhuhai** Tel: 86-756-3210040 Fax: 86-756-3210049

#### ASIA/PACIFIC

India - Bangalore Tel: 91-80-3090-4444 Fax: 91-80-3090-4123

**India - New Delhi** Tel: 91-11-4160-8631 Fax: 91-11-4160-8632

India - Pune Tel: 91-20-2566-1512 Fax: 91-20-2566-1513

**Japan - Yokohama** Tel: 81-45-471- 6166 Fax: 81-45-471-6122

**Korea - Daegu** Tel: 82-53-744-4301 Fax: 82-53-744-4302

Korea - Seoul Tel: 82-2-554-7200 Fax: 82-2-558-5932 or 82-2-558-5934

Malaysia - Kuala Lumpur Tel: 60-3-6201-9857 Fax: 60-3-6201-9859

**Malaysia - Penang** Tel: 60-4-227-8870 Fax: 60-4-227-4068

Philippines - Manila Tel: 63-2-634-9065 Fax: 63-2-634-9069

Singapore Tel: 65-6334-8870 Fax: 65-6334-8850

**Taiwan - Hsin Chu** Tel: 886-3-5778-366 Fax: 886-3-5770-955

**Taiwan - Kaohsiung** Tel: 886-7-536-4818 Fax: 886-7-330-9305

Taiwan - Taipei Tel: 886-2-2500-6610 Fax: 886-2-2508-0102

**Thailand - Bangkok** Tel: 66-2-694-1351 Fax: 66-2-694-1350

#### EUROPE

Austria - Wels Tel: 43-7242-2244-39 Fax: 43-7242-2244-393 Denmark - Copenhagen Tel: 45-4450-2828 Fax: 45-4485-2829

France - Paris Tel: 33-1-69-53-63-20 Fax: 33-1-69-30-90-79

**Germany - Munich** Tel: 49-89-627-144-0 Fax: 49-89-627-144-44

Italy - Milan Tel: 39-0331-742611 Fax: 39-0331-466781

Netherlands - Drunen Tel: 31-416-690399 Fax: 31-416-690340

**Spain - Madrid** Tel: 34-91-708-08-90 Fax: 34-91-708-08-91

**UK - Wokingham** Tel: 44-118-921-5869 Fax: 44-118-921-5820

08/02/11