Section 10. Power-Saving Modes

HIGHLIGHTS

This section of the manual contains the following major topics:

10.1 Introduction ................................................................. 10-2
10.2 Power-Saving Modes Control Registers ........................................ 10-3
10.3 Operation of Power-Saving Modes ........................................... 10-3
10.4 Interrupts ........................................................................... 10-8
10.5 I/O Pins Associated with Power-Saving Modes ............................. 10-9
10.6 Operation in Debug Mode .................................................. 10-9
10.7 Related Application Notes .................................................. 10-10
10.8 Revision History .............................................................. 10-11
10.1 INTRODUCTION

This section describes the power-saving modes of operation for the PIC32 device family. PIC32 devices have nine low-power modes in two categories, that allow the user to balance power consumption along with device performance. In all of the modes discussed in this section, the device can select the desired power-saving mode through software.

10.1.1 CPU Running Modes

In the CPU running modes, the CPU is running and peripherals can optionally be switched ON or OFF.

- Fast RC (FRC) Run mode: the CPU is clocked from the FRC clock source with or without postscalers
- Low-Power RC (LPRC) Run mode: the CPU is clocked from the LPRC clock source
- Secondary Oscillator (SOSC) Run mode: the CPU is clocked from the Sosc clock source
- Peripheral Bus Scaling mode: Peripherals are clocked at programmable fraction of the system clock (SYSCLK)

10.1.2 CPU Halted Modes

In the CPU halted modes, the CPU is halted. Depending on the mode, peripherals can continue to operate or be halted.

- Primary Oscillator (Posc) Idle mode: the SYSCLK is derived from the Posc. SYSCLK source continues to operate. Peripherals continue to operate, but can optionally be individually disabled.
- FRC Idle mode: the SYSCLK is derived from the FRC with or without postscalers. Peripherals continue to operate, but can optionally be individually disabled.
- Sosc Idle mode: the SYSCLK is derived from the Sosc. Peripherals continue to operate, but can optionally be individually disabled.
- LPRC Idle mode: the SYSCLK is derived from the LPRC. Peripherals continue to operate, but can optionally be individually disabled. This is the lowest power mode for the device with a clock running.
- Sleep mode: the CPU, the SYSCLK source and any peripherals that operate from the SYSCLK source are halted. Some peripherals can operate while in Sleep mode using specific clock sources. This is the lowest power mode for the device.
Power-Saving Modes

10.2 POWER-SAVING Modes Control Registers

The Special Function Registers (SFRs) that can be used in support of power-saving modes are:
- Oscillator Control Register (OSCCON)
- Watchdog Timer Control Register (WDTCON)
- Resets Control Register (RCON)

Refer to the following family reference manual sections and the related chapter in the specific device data sheet for descriptions of these registers and their available bits:
- Section 6. “Oscillators” (DS61112)
- Section 7. “Resets” (DS61118)
- Section 9. “Watchdog Timer (WDT) and Power-up Timer (PWRT)” (DS61114)

10.3 OPERATION OF Power-SAVING Modes

The PIC32 device family consists of nine power-saving modes, the purpose of which is to reduce power consumption by reducing the device clock frequency. To achieve this, multiple low-frequency clock sources can be selected. In addition, the peripherals and CPU can be halted or disabled to further reduce power consumption.

10.3.1 Sleep Mode

Sleep mode has the lowest power consumption of the device power-saving operating modes. When a device is placed in Sleep mode, the CPU and most peripherals are halted. However, selected peripherals can continue to operate and can be used to wake-up the device from Sleep mode. Refer to the individual peripheral module family reference manual sections for descriptions of behavior in Sleep mode.

The characteristics of Sleep mode are as follows:
- The CPU is halted
- The SYSCLK source is typically shut down. For more information, refer to 10.3.1.1 “Oscillator Shutdown in Sleep Mode”.
- There can be a wake-up delay based on the oscillator selection (see Table 10-1)
- The Fail-Safe Clock Monitor (FSCM) does not operate during Sleep mode
- The Brown-out Reset (BOR) circuit remains operative during Sleep mode
- The Watchdog Timer (WDT), if enabled, is not automatically cleared prior to entering Sleep mode
- Some peripherals can continue to operate in Sleep mode. These peripherals include I/O pins that detect a change in the input signal, WDT, RTCC, ADC, and UART modules, and the peripherals that use an external clock input or the internal LPRC oscillator.
- I/O pins continue to sink or source current in the same manner when the device is not in Sleep mode
- The Universal Serial Bus (USB) module can override the disabling of the Posc or FRC. For more information, refer to Section 27. “USB On-The-Go” (DS61126).

The processor will exit or wake-up from Sleep mode in one of the following events:
- On any interrupt from an enabled source that is operating in Sleep mode. The interrupt priority must be greater than the current CPU priority.
- On any form of device Reset
- On a WDT time-out (see 10.4.2 “Wake-up from Sleep Mode or Idle Mode on Watchdog Time-out (NMI)"

If the interrupt priority is lower than or equal to the current priority, the CPU will remain halted, but the Peripheral Bus Clock (PBCLK) will start running and the device will enter Idle mode. Example 10-1 shows an example code for placing the device in Sleep mode and using the WDT to wake the device.
Example 10-1: Placing the Device in Sleep mode and Using the WDT to Wake the Device

```c
// Code example to put the Device in sleep and then wake the device using the WDT

OSCCONSET = 0x10; // set Power-Saving mode to Sleep

WDTCONCLR = 0x0002; // Disable WDT window mode
WDTCONSET = 0x8000; // Enable WDT

// WDT time-out period is set in the device configuration

... user code ...

WDTCONSET = 0x01; // service the WDT
asm volatile(“wait”); // put device in selected power-saving mode
// code execution will resume after wake

... user code ...

// The following code fragment is at the beginning of the ‘C’ start-up code
// to find whether the wake from Sleep is due to the WDT

if ( RCON & 0x18 ) // The WDT caused a wake from Sleep
{
    asm volatile(“eret”); // return from interrupt
}
```

10.3.1.1 Oscillator Shutdown in Sleep Mode

The oscillator behavior in Sleep mode is as follows:

- If the CPU clock source is Posc, the oscillator is turned OFF in Sleep mode
- If the CPU clock source is FRC, the oscillator is turned OFF in Sleep mode
- If the CPU clock source is Sosc, the oscillator will be turned OFF if the SOSCEN bit is not set
- If the CPU clock source is LPRC, the oscillator will be turned OFF if the clock source is not being used by a peripheral that will be operating in Sleep mode, such as the WDT

**Note:** Refer to Table 10-1 for applicable delays when waking from Sleep mode. The USB module can override the disabling of the Posc or FRC. For more information, refer to Section 27. “USB On-The-Go” (DS61126).

10.3.1.2 Clock Selection on Wake-up from Sleep mode

The processor will resume code execution and use the same clock source that was active when Sleep mode was entered. The device is subject to a start-up delay, if a crystal oscillator and/or Phase-Locked Loop (PLL) is used as a clock source when the device exits Sleep mode.

10.3.1.3 Delay on Wake-up from Sleep mode

The Oscillator Start-up Timer (OST) and FSCM delays, if enabled, associated with wake-up from Sleep mode, are shown in Table 10-1.

**Table 10-1:** Delay Times for Exit from Sleep Mode

<table>
<thead>
<tr>
<th>Clock Source</th>
<th>Oscillator Delay</th>
<th>FSCM Delay</th>
</tr>
</thead>
<tbody>
<tr>
<td>EC, EXTRC</td>
<td>—</td>
<td>—</td>
</tr>
<tr>
<td>EC + PLL</td>
<td>TLOCK</td>
<td>TFSCM</td>
</tr>
<tr>
<td>XT + PLL</td>
<td>TOST + TLOCK</td>
<td>TFSCM</td>
</tr>
<tr>
<td>XT, HS, XTL</td>
<td>TOST</td>
<td>TFSCM</td>
</tr>
<tr>
<td>LP (OFF during Sleep)</td>
<td>TOST</td>
<td>TFSCM</td>
</tr>
<tr>
<td>LP (ON during Sleep)</td>
<td>—</td>
<td>—</td>
</tr>
<tr>
<td>FRC, LPRC</td>
<td>—</td>
<td>—</td>
</tr>
</tbody>
</table>

**Note:** Refer to the “Electrical Characteristics” section in the specific device data sheet for the TPOR, TFSCM and TLOCK specification values.
10.3.1.4 WAKE-UP FROM SLEEP MODE WITH CRYSTAL OSCILLATOR OR PLL

If the SYSCLK source is derived from a crystal oscillator and/or the PLL, the OST and/or PLL lock times will be applied before the SYSCLK source is made available to the device. As an exception to this rule, no oscillator delays are applied if the SYSCLK source is the POSC oscillator, and it was running while in Sleep mode.

Note: Regardless of the various delays applied, the crystal oscillator (and PLL) may not be up and running at the end of the TOST or TLOCK delays. For proper operation, the user must design the external oscillator circuit such that reliable oscillation will occur within the delay period.

10.3.1.5 Fail-Safe Clock Monitor Delay and Sleep Mode

The FSCM does not operate while the device is in Sleep mode. If the FSCM is enabled, it will resume operation when the device wakes from Sleep mode. A delay of TFSCM is applied to allow the oscillator source to stabilize before the FSCM resumes monitoring.

When the following conditions are true, a delay of TFSCM will be applied when waking from Sleep mode:

- The oscillator was shut down while in Sleep mode
- The SYSCLK is derived from a crystal oscillator source and/or the PLL

In most cases, the TFSCM delay provides time for the OST to expire and the PLL to stabilize before device execution resumes. If the FSCM is enabled, it will begin to monitor the SYSCLK source after the TFSCM delay expires.

10.3.1.6 Slow Oscillator Start-up

When an oscillator starts slowly, the OST and PLL lock times may not expire before the FSCM time-out.

If the FSCM is enabled, the device will detect this condition as a clock failure and a clock fail trap will occur. The device will switch to the FRC oscillator and the user can re-enable the crystal oscillator source in the clock failure Interrupt Service Routine (ISR).

If the FSCM is not enabled, the device will simply not start executing the code until the clock is stable. From the user perspective, the device will appear to be in Sleep mode until the oscillator clock has started.

10.3.1.7 USB Peripheral Control of Oscillators in Sleep Mode

The USB module, when active, will prevent the clock source it is using from being disabled when the device enters Sleep mode. Although the oscillator remains active, the CPU and peripherals will remain halted.
10.3.2 Peripheral Bus Scaling

Most of the peripherals on the device are clocked using the PBCLK. The peripheral bus can be scaled relative to the SYSCLK to minimize the dynamic power consumed by the peripherals. The PBCLK divisor is controlled by the PBDIV<1:0> bits (OSCCON<20:19>), allowing SYSCLK-to-PBCLK ratios of 1:1, 1:2, 1:4 and 1:8. All peripherals using the PBCLK are affected when the divisor is changed. Peripherals such as the Interrupt Controller, Direct Memory Access (DMA), Bus Matrix and Prefetch Cache are clocked directly from the SYSCLK and as a result, they are not affected by the PBCLK divisor changes.

Most of the peripherals on the device are clocked using the PBCLK. The peripheral bus can be scaled relative to the SYSCLK to minimize the dynamic power consumed by the peripherals. The PBCLK divisor is controlled by the PBDIV<1:0> bits (OSCCON<20:19>), allowing SYSCLK-to-PBCLK ratios of 1:1, 1:2, 1:4 and 1:8. All peripherals using PBCLK are affected when the divisor is changed. Peripherals such as the USB, Interrupt Controller, DMA, Bus Matrix and Prefetch Cache are clocked directly from the SYSCLK and as a result, they are not affected by PBCLK divisor changes.

Changing the PBCLK divisor affects:

• The CPU-to-peripheral access latency. The CPU has to wait for the next PBCLK edge for a read to complete. In 1:8 mode, this results in a latency of one to seven SYSCLKs.
• The power consumption of the peripherals. Power consumption is directly proportional to the frequency at which the peripherals are clocked. The greater the divisor, the lower the power consumed by the peripherals.

To minimize dynamic power, the PBCLK divisor should be selected to run the peripherals at the lowest frequency that provides acceptable system performance. When selecting a PBCLK divisor, peripheral clock requirements such as baud rate accuracy should be considered. For example, the UART peripheral may not be able to achieve all baud rate values at some PBCLK divisor depending on the SYSCLK value.

10.3.2.1 Dynamic Peripheral Bus Scaling

The PBCLK can be scaled dynamically, in software, to save additional power when the device is in a low activity mode. The following issues need to be considered when scaling the PBCLK:

• All the peripherals clocked from the PBCLK will scale at the same ratio and at the same time. This needs to be accounted for in peripherals that need to maintain a constant baud rate, or pulse period while in low-power modes.
• Any communication through a peripheral, on the peripheral bus that is in progress when the PBCLK changes, may cause a data or protocol error due to a frequency change during transmission or reception.

If the user intends to scale the PBCLK divisor dynamically, the following steps are recommended:

1. Disable all communication peripherals whose baud rate will be affected. Ensure that no communication is currently in progress before disabling the peripherals as it may result in protocol errors.
2. Update the Baud Rate Generator (BRG) settings for peripherals as required for operation at the new PBCLK frequency.
3. Change the peripheral bus ratio to the desired value.
4. Enable all communication peripherals whose baud rate were affected.

**Note:** Modifying the peripheral baud rate is done by writing to the associated peripheral SFRs. To minimize latency, the peripherals should be modified in the mode where the PBCLK is running at its highest frequency.
10.3.3 Idle Modes

In an idle mode, the CPU is halted but the SYSCLK source is still enabled. This allows peripherals to continue to operate when the CPU is halted. Peripherals can be individually configured to halt when entering an idle mode by setting their respective SIDL bit. Latency when exiting an idle mode is very low due to the CPU oscillator source remaining active.

There are four idle modes of operation:

- **Posc Idle mode**: The SYSCLK is derived from the Posc. The CPU is halted, but the SYSCLK source continues to operate. Peripherals continue to operate, but can optionally be individually disabled. If the PLL is used, the multiplier value, which is controlled by the PLLMULT<2:0> bits (OSCCON<18:16>), can also be lowered to reduce power consumption by peripherals.

- **FRC Idle mode**: The SYSCLK is derived from the FRC. The CPU is halted. Peripherals continue to operate, but can optionally be individually disabled. If the PLL is used, the multiplier value, which is controlled using the PLLMULT<2:0> bits (OSCCON<18:16>), can also be lowered to reduce power consumption by peripherals. The FRC clock can be further divided by a postscaler using the RCDIV<2:0> bits (OSCCON<26:24>).

- **Sosc Idle mode**: The SYSCLK is derived from the Sosc and the CPU is halted. Peripherals continue to operate, but can optionally be individually disabled.

- **LPRC Idle mode**: The SYSCLK is derived from the LPRC and the CPU is halted. Peripherals continue to operate, but can optionally be individually disabled.

The device enters an idle mode when the SLPEN bit (OSCCON<4>) is clear and a WAIT instruction is executed. The processor will wake-up or exit from an idle mode on the following events:

- On any interrupt event for which the interrupt source is enabled. The priority of the interrupt event must be greater than the current priority of CPU. If the priority of the interrupt event is lower than or equal to current priority of CPU, the CPU will remain halted and the device will remain in the selected idle mode.

- On any source of device Reset

- On a WDT time-out interrupt. Refer to 10.4.2 “Wake-up from Sleep Mode or Idle Mode on Watchdog Time-out (NMI)” and Section 9. “Watchdog Timer and Power-up Timer” (DS61114).

Example 10-2 provides the example code for placing the device in an idle mode and wake-up by the ADC event.

---

**Note 1:** Changing the PBCLK divisor ratio requires recalculation of peripheral timing. For example, assume the UART is configured for 9600 baud with a PBCLK ratio of 1:1 and a Posc of 8 MHz. When the PBCLK divisor of 1:2 is used, the input frequency to the baud clock is divided into half; therefore, the baud rate is reduced to half its former value. Due to numeric truncation in calculations (such as the baud rate divisor), the actual baud rate may be a very small percentage and different than expected. For this reason, any timing calculation required for a peripheral should be performed with the new PBCLK frequency instead of scaling the previous value based on a change in the PBCLK divisor ratio.

**Note 2:** Oscillator start-up and PLL lock delays are applied when switching to a clock source that was disabled and that uses a crystal and/or the PLL. For example, assume the clock source is switched from the Posc to the LPRC before entering Sleep mode in order to save power. No oscillator start-up delay would be applied when exiting Idle mode. However, when switching back to the Posc, the appropriate PLL and or oscillator start-up/lock delays would be applied.
Example 10-2: Placing the Device in an Idle Mode and Wake-up by ADC Event

```c
// Code example to put the Device in Idle and then
// wake the device when the ADC completes a conversion
SYSKEY = 0x0;  // Write invalid key to force lock
SYSKEY = 0xAA996655;  // Write Key1 to SYSKEY
SYSKEY = 0x556699AA;  // Write Key2 to SYSKEY
OSCCONCLR = 0x10;  // Set the power-saving mode to an idle mode
SYSKEY = 0x0;  // Write invalid key to force lock

asm volatile ("wait");  // Put device in selected power-saving mode

// Code execution will resume here after wake and
// the ISR is complete
... user code ...

// interrupt handler
void __ISR(_ADC_VECTOR, ipl7) ADC_HANDLER(void)
{
    unsigned long int result;
    result = ADC1BUF0;  // Read the result
    IFS1CLR = 2;  // Clear ADC conversion interrupt flag
}
```

10.4 INTERRUPTS

There are two sources of interrupts that will wake the device from a power-saving mode: peripheral interrupts and a Non-maskable Interrupt (NMI) generated by the WDT while in a power-saving mode.

10.4.1 Wake-up from Sleep mode or Idle mode on Peripheral Interrupt

Any source of interrupt that is individually enabled using the corresponding Interrupt Enabled (IE) control bit in the IECx register, and is operational in the current power-saving mode, will be able to wake the processor from Sleep mode or an idle mode. When the device wakes up, one of the following events will occur based on the interrupt priority:

- If the assigned priority for the interrupt is less than or equal to the current CPU priority, the CPU will remain halted and the device enters or remains in an idle mode.
- If the assigned priority level for the interrupt source is greater than the current CPU priority, the device will wake up and the CPU will jump to the corresponding interrupt vector. Upon completion of the ISR, CPU will start executing the next instruction after the `WAIT` instruction.

The Idle Status bit (RCON<2>) is set upon wake-up from an idle mode. The Sleep Status bit (RCON<3>) is set upon wake-up from Sleep mode.

**Note 1:** The Idle Status bit (RCON<2>) will also become set upon wake-up from Sleep mode.

**Note 2:** A peripheral with an interrupt priority setting of Zero cannot wake up the device.

**Note 3:** Any applicable oscillator start-up delays are applied before the CPU resumes code execution.
10.4.2  Wake-up from Sleep Mode or Idle Mode on Watchdog Time-out (NMI)

When the WDT times out in Sleep mode or an idle mode, an NMI is generated. The NMI causes the CPU code execution to jump to the device reset vector. Although the CPU executes the reset vector, it is not a device Reset – peripherals and most CPU registers do not change their states.

Note: Any applicable oscillator start-up delays are applied before the CPU resumes code execution.

To detect a wake-up from a power-saving mode caused by WDT expiration, the WDTO bit (RCON<4>), the SLEEP bit (RCON<3>) and the IDLE bit (RCON<2>) must be tested. If the WDTO bit (RCON<4>) is ‘1’, the event was due to a WDT time-out. The SLEEP and IDLE bits can then be tested to determine if the WDT event occurred in Sleep mode or an idle mode.

To use a WDT time-out during Sleep mode as a wake-up interrupt, a return from interrupt (ERET) instruction must be used in the start-up code after the event was determined to be a WDT wake-up. This will cause code execution to continue from the instruction following the WAIT instruction that put the device in a power-saving mode.

Note: If a peripheral interrupt and WDT event occur simultaneously, or in close proximity, the NMI may not occur, due to the device wake-up by the peripheral interrupt. To avoid an unexpected WDT reset, the WDT is automatically cleared when the device wakes up.

For more details on WDT operation, refer to Section 9. “Watchdog Timer and Power-up Timer” (DS61114).

10.4.3  Interrupts Coincident with Power-Saving Instruction

Any peripheral interrupt that coincides with the execution of a WAIT instruction will be held until entry into Sleep mode or an idle mode has completed. The device will then wake up from Sleep mode or the selected idle mode.

10.5  I/O PINS ASSOCIATED WITH POWER-SAVING MODES

No device pins are associated with power-saving modes.

10.6  OPERATION IN DEBUG MODE

The user cannot change clock modes when the debugger is active. Clock source changes due to the FSCM will still occur when the debugger is active.
10.7 RELATED APPLICATION NOTES

This section lists application notes that are related to this section of the manual. These application notes may not be written specifically for the PIC32 device family, but the concepts are pertinent and could be used with modification and possible limitations. The current application notes related to Power-Saving Modes include the following:

<table>
<thead>
<tr>
<th>Title</th>
<th>Application Note #</th>
</tr>
</thead>
<tbody>
<tr>
<td>Low-Power Design using PIC® Microcontrollers</td>
<td>AN606</td>
</tr>
</tbody>
</table>

Note: Please visit the Microchip web site (www.microchip.com) for additional application notes and code examples for the PIC32 family of devices.
10.8 REVISION HISTORY

Revision A (October 2007)
This is the initial released version of this document.

Revision B (October 2007)
Updated document to remove Confidential status.

Revision C (April 2008)
Revised status to Preliminary; Revised U-0 to r-x.

Revision D (July 2008)
Revised Example 10-1 and 10-3; Revised Table 10-1; Revised Register 10-5 and 10-9; Revised Section 10.3.2 (2nd para.); Change Reserved bits from “Maintain as” to “Write”; Added Note to ON bit (WDTCN Register).

Revision E (July 2008)
Revised Examples 10-2, 10-3.

Revision F (August 2011)
• Examples:
  - Updated Example 10-1
  - Removed Example 10-2: Changing the PBCLK Divisor
  - Updated Example 10-2
• Notes:
  - Removed a note in the first paragraph, in 10.3 “Operation of Power-Saving Modes”
  - Removed a note on Freeze mode in 10.3.1 “Sleep Mode”
  - Added Note 1 in 10.4.1 “Wake-up from Sleep mode or Idle mode on Peripheral Interrupt”
• Registers:
  - Removed Register 10-1 through Register 10-12
• Sections:
  - Updated 10.2 “Power-Saving Modes Control Registers”
  - Updated the characteristics of Sleep mode, in 10.3.1 “Sleep Mode”
  - Updated 10.3.1.1 “Oscillator Shutdown in Sleep Mode”
  - Removed Section 10.7 “Resets”
  - Removed Section 10.8 “Design Tips”
• Tables:
  - Removed Table 10-1: Power-Saving Modes SFR Summary
• All references to PIC32MX were updated to PIC32
• Removed the Preliminary document status
• Additional minor corrections such as text and formatting updates were incorporated throughout the document
Note the following details of the code protection feature on Microchip devices:

- Microchip products meet the specification contained in their particular Microchip Data Sheet.
- Microchip believes that its family of products is one of the most secure families of its kind on the market today, when used in the intended manner and under normal conditions.
- There are dishonest and possibly illegal methods used to breach the code protection feature. All of these methods, to our knowledge, require using the Microchip products in a manner outside the operating specifications contained in Microchip’s Data Sheets. Most likely, the person doing so is engaged in theft of intellectual property.
- Microchip is willing to work with the customer who is concerned about the integrity of their code.
- Neither Microchip nor any other semiconductor manufacturer can guarantee the security of their code. Code protection does not mean that we are guaranteeing the product as “unbreakable.”

Code protection is constantly evolving. We at Microchip are committed to continuously improving the code protection features of our products. Attempts to break Microchip’s code protection feature may be a violation of the Digital Millennium Copyright Act. If such acts allow unauthorized access to your software or other copyrighted work, you may have a right to sue for relief under that Act.

Trademarks

The Microchip name and logo, the Microchip logo, dsPIC, Keeloo, Keeloo logo, MPLAB, PIC, PICmicro, PICSTART, PIC32 logo, rfPIC and UNI/O are registered trademarks of Microchip Technology Incorporated in the U.S.A. and other countries.

FilterLab, Hampshire, Hi-Tech C, Linear Active Thermistor, MXDEV, MXLAB, SEEVAL and The Embedded Control Solutions Company are registered trademarks of Microchip Technology Incorporated in the U.S.A.

Analog-for-the-Digital Age, Application Maestro, chipKIT, chipKIT logo, CodeGuard, dsPICDEM, dsPICDEM.net, dsPICworks, dsPEAK, ECAN, ECONOMONITOR, FanSense, Hi-TIDE, In-Circuit Serial Programming, ICSP, Mindi, MIWI, MPASM, MPLAB Certified logo, MPLIB, MPLINK, mTouch, Omniscient Code Generation, PICC, PICC-18, PICDEM, PICDEM.net, PICkit, PICtail, REAL ICE, rLAB, Select Mode, Total Endurance, TSHARC, UniWinDriver, WiperLock and ZENA are trademarks of Microchip Technology Incorporated in the U.S.A. and other countries.

SQTP is a service mark of Microchip Technology Incorporated in the U.S.A.

All other trademarks mentioned herein are property of their respective companies.


Printed on recycled paper.

### 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

**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: 216-447-0466  
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 - Sydney**  
Tel: 61-2-9888-6733  
Fax: 61-2-9888-6755

**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: 86-25-6473-2460  
Fax: 86-25-6473-2470

**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

© 2011 Microchip Technology Inc.