INTRODUCTION

Accessing the exact peripheral set, utilizing a smaller package and reducing the board layout complexity are some of the benefits when a microcontroller has the capability to map its digital peripherals to a wide range of I/O pins. The Peripheral Pin Select (PPS) module in Microchip’s 8-bit microcontroller provides this capability statically after code initialization or dynamically throughout during code execution. This technical brief describes the PPS features, method of configuration and mapping flexibility of PPS for both PIC16F and PIC18F microcontroller devices.

PPS IN PIC16F DEVICES

The PPS module selects which digital input or output of a peripheral goes in and out to a microcontroller pin. Figure 1 shows the PPS block diagram for the PIC16F devices. It is basically composed of multiplexers and control registers. In the input section, the pins are multiplexed to the input of a certain peripheral. In the output section, the different peripheral outputs are multiplexed to the pins of the microcontroller. The PPS control registers map the different peripheral functions and pins.

The PPS varies for each microcontroller device. Some devices have full PPS capability, where every remappable peripheral input or output functions can be placed on any I/O pins of the microcontroller. Other devices have a reduced version of PPS where the remappable peripheral functions can only be mapped to specific pins. Refer to the respective device data sheets for more information.
PIC16F PPS INPUT SELECTION

In Figure 2, the Peripheral xxx Input Selection (xxxPPS) register controls which pin would be connected to the input of the peripheral. The notation “xxx” describes the type of peripheral input. By default, the register is preloaded with a 5-bit field value, which means that each digital input is initially tied to a specific pin. Modifying this register changes the connection of the peripheral to another pin.

FIGURE 2: INPUT SELECTION DIAGRAM

For example, the RC4 pin can be assigned to the COGIN input of the Complementary Output Generator peripheral. See code in Example 1.

EXAMPLE 1: ASSIGNING THE RC4 PIN TO THE COGIN INPUT

/*RC4 is selected as COGIN of PIC16F1716*/
COGINPPS=0b10100

PIC16F PPS OUTPUT SELECTION

In Figure 3, the pin Rxy Output Source Selection (RxyPPS) register controls which peripheral output would be connected to a particular pin. The notation “Rxy” refers to any pin of the microcontroller. By default, this register holds a 5-bit field value of 00h, which ties the Rxy pin to LATxy. Modifying this register disconnects Rxy from LATxy and connects it to the output of other peripherals.

FIGURE 3: OUTPUT SELECTION DIAGRAM

To prevent unwanted changes to the PPS registers, the PPS1WAY fuse in the CONFIG2 register can be set to lock the PPS registers permanently. When the PPS1WAY is enabled, the PPSLOCKED bit can only be set once in software, thus preventing any more changes on the PPS registers. The code in Example 4 shows how PPS1WAY can be enabled.

EXAMPLE 4: ENABLING THE PPS1WAY FUSE

#pragma config PPS1WAY = ON

The PWM output of the Capture Compare PWM (CCP1) peripheral can be assigned to RB1. See code in Example 2.

EXAMPLE 2: ASSIGNING THE CCP1 PWM OUTPUT TO THE RB1 PIN

/*RB1 is selected as CCP1PWM of PIC16F1716*/
RB1PPS=0b01100;

PIC16F PPS UNLOCK/LOCK MECHANISM

It is necessary that the PPS registers must be unlocked first before modifications can be made on the registers. Unlocking or locking of PPS registers are done through a special code sequence that clears or sets the PPSLOCKED bit. See the code in Example 3.

EXAMPLE 3: CODE SEQUENCE FOR UNLOCKING/LOCKING THE PIC16F PPS REGISTERS

//disable interrupts
//unlock the PPS registers for changes
GIE = 0;
PPSLOCK = 0x55;
PPSLOCK = 0xAA;
PPSLOCK = 0;

/*Place code here for Peripheral Pin Selection, write to PPS registers*/

//lock PPS registers
//re-enable interrupts
PPSLOCK = 0x55;
PPSLOCK = 0xAA;
PPSLOCK = 1;
GIE = 1;

/*RC4 is selected as COGIN of PIC16F1716*/
COGINPPS=0b10100

/*RB1 is selected as CCP1PWM of PIC16F1716*/
RB1PPS=0b01100;
PPS IN PIC18F DEVICES

The PPS for PIC18F devices has a different implementation than that of the PIC16F devices. The control registers and the pin have different naming, as shown in Figure 4. The peripheral pin mapping can be done through reconfigurable pins labeled as RPn. The PPS input register RPINRx selects which pins goes to a peripheral input while the PPS output register RPORx selects which peripheral output function goes to an RPn pin.

FIGURE 4: PPS BLOCK DIAGRAM FOR PIC18F DEVICES

PIC18F PPS INPUT SELECTION

The RPINRx register controls which pin can be mapped to specific peripheral input. A value must be written on the RPINRx to select the RPn pin that can be connected to the input of the peripheral. The RPINRx holds a default 5-bit field value of 0b11111. This ties all the input of remappable peripherals to initially RP31.

The code in Example 5 shows how to assign RP2 as the FLT0 PWM Fault Input pin of the Enhanced Capture Compare PWM (ECCP1) peripheral. This can be done by making RP2 a digital input pin and loading RPINR24 (the register associated with FLT0) with a decimal value of 2.

EXAMPLE 5: ASSIGNING THE RP2 PIN TO THE FLT0 INPUT

/*RP2(RA5) goes to FLT0 of ECCP1 PWM of PIC18F47J13*/
RPINR24 = 2;
PIC18F PPS OUTPUT SELECTION

The RPORx register controls which peripheral output can be mapped to a particular RPn pin. A value must be written on the RPORx to select the peripheral output that can be connected to that pin. RPORx holds a default 5-bit value of 0b00000 or null, which means that the pin is disconnected to any of the remappable peripheral outputs.

The RP15 pin can be assigned as the CCP1 PWM output. This can be done by configuring RP15 as digital output and writing the RPOR15 (the register associated with RP15) with a decimal value that corresponds to CCP1/P1A output function (see the code in Example 6).

EXAMPLE 6: ASSIGNING THE CCP1/P1A TO THE RP15 PIN

```c
/*CCP1/P1A (Output Function 14) is remapped to pin RP15 of PIC18F47J13*/
RPOR15 = 14;
```

TABLE 1: RECONFIGURABLE PIN GROUPING FOR PIC18F97J94

<table>
<thead>
<tr>
<th>n</th>
<th>Group 4n</th>
<th>Group 4n+1</th>
<th>Group 4n+2</th>
<th>Group 4n+3</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>RP0</td>
<td>RP1</td>
<td>RP2</td>
<td>RP3</td>
</tr>
<tr>
<td>1</td>
<td>RP4</td>
<td>RP5</td>
<td>RP6</td>
<td>RP7</td>
</tr>
<tr>
<td>2</td>
<td>RP8</td>
<td>RP9</td>
<td>RP10</td>
<td>RP11</td>
</tr>
<tr>
<td>3</td>
<td>RP12</td>
<td>RP13</td>
<td>RP14</td>
<td>RP15</td>
</tr>
<tr>
<td>4</td>
<td>RP16</td>
<td>RP17</td>
<td>RP18</td>
<td>RP19</td>
</tr>
<tr>
<td>5</td>
<td>RP20</td>
<td>RP21</td>
<td>RP22</td>
<td>RP23</td>
</tr>
<tr>
<td>6</td>
<td>RP24</td>
<td>RP25</td>
<td>RP26</td>
<td>RP27</td>
</tr>
<tr>
<td>7</td>
<td>RP28</td>
<td>RP29</td>
<td>RP30</td>
<td>RP31</td>
</tr>
<tr>
<td>8</td>
<td>RP32</td>
<td>RP33</td>
<td>RP34</td>
<td>RP35</td>
</tr>
<tr>
<td>9</td>
<td>RP36</td>
<td>RP37</td>
<td>RP38</td>
<td>RP39</td>
</tr>
<tr>
<td>10</td>
<td>RP40</td>
<td>RP41</td>
<td>RP42</td>
<td>RP43</td>
</tr>
<tr>
<td>11</td>
<td>RP44</td>
<td>RP45</td>
<td>RP46</td>
<td></td>
</tr>
</tbody>
</table>

The control registers in PPS-Lite contain two values. The RPINRx input register holds two 4-bit fields. Each bit field is associated with a peripheral input for the pin selection. For example, in the PIC18F97J94, the RPINRx register controls both the ECCP1 Capture input and the FLT0 input. RPINRx<7:4> maps ECCP1 Capture to pins from “Group 4n+3” and RPINRx<3:0> maps FLT0 input to pins from “Group 4n.”

Loading RPINRx<7:4> with 0h assigns ECCP1 Capture input to RP0. Writing RPINRx<3:0> with 0h assigns FLT0 input to RP3. ECCP1 peripheral is from Group 4n and FLT0 belongs to Group 4n+3. This is shown in the code in Example 7. Refer to the PIC18F97J94 data sheet for the complete list of RPINRx registers and available functions.

EXAMPLE 7: MAPPING PINS TO INPUT FUNCTIONS IN PIC18F DEVICES HAVING PPS-LITE

```c
//ECCP1 Capture to RP0
RPINRx<7:4> = 0x0;

//FLT0 to RP3
RPINRx<3:0> = 0x0;
```

The RPORx output register also contains two 4-bit fields that correspond to the two RPn pins. For example, the RPORx<7:4> is associated to RP0 of “Group 4n” while RPORx<3:0> to RP1 of “Group 4n+1.” The value of these bit fields selects the corresponding peripheral output that is in the group. The code in Example 8 illustrates the selection for RP0, RP1, RP2, and RP3 as the P1D, P1C, P1B and...
EXAMPLE 8: MAPPING PINS TO INPUT FUNCTIONS IN PIC18F DEVICES HAVING PPS-LITE

```c
/*P1D (Function 5) output is remapped to pin RP0*/
RPOR0_1bits.RPOR0R = 0x5;

/*P1C (Function 5) output is remapped to pin RP1*/
RPOR0_1bits.RPOR1R = 0x5;

/*P1B (Function 6) output is remapped to pin RP2*/
RPOR2_3bits.RPOR2R = 0x6;

/*P1A (Function 4) output is remapped to pin RP3*/
RPOR2_3bits.RPOR3R = 0x4;
```

### PIC18F PPS UNLOCK/LOCK MECHANISM

The PPS registers can only be modified by first clearing the IOLOCK bit. IOLOCK can be cleared by writing 55h and AAh to the EECON2 register then applying the changes to the PPS registers. To lock the PPS registers, IOLOCK must be set through the same code sequence as shown in the code in Example 9.

### EXAMPLE 9: CODE SEQUENCE FOR UNLOCKING/LOCKING THE PIC18F PS REGISTERS

```c
GIE = 0;
EECON2 = 0x55;
EECON2 = 0xAA;
IOLOCK = 0;

/*Place code here for PPS*/

EECON2 = 0x55;
EECON2 = 0xAA;
IOLOCK = 1;
GIE = 1;
```

An IOL1WAY fuse allows the IOLOCK bit to be set only once in software. It protects the PPS registers from unintentional changes that can happen during software revisions or software debug. The code in Example 10 shows how IOL1WAY can be enabled.

### EXAMPLE 10: ENABLING THE IOL1WAY FUSE

```c
#pragma config IOL1WAY = ON
```

### PPS FLEXIBLE MAPPING

When using PPS, mapping flexibility can be achieved. PPS can expand the range of functions of a digital input pin and can also output a single digital function on multiple pins.

Two or more input functions can be defined on a single pin through the routing of the digital signal to different peripheral inputs. A single event can trigger several peripheral functions like shutting down a PWM signal while capturing a timed event through the use of only one input pin.

The PPS also allows the same digital output signal of a peripheral to be made available on multiple pins simultaneously. For example, an SPI, COG or PWM peripheral output signal may be replicated without the use of external buffer or added code. This functionality can be used for applications that require an output signal to be routed to different sections of a circuit or a system.

For example, the code in Example 11 illustrates the mapping flexibility of the PIC16F's PPS module. The code sets the pin RB0 to serve both as COGIN shutdown and CCP2 capture input. Then, it outputs COG1A signals on RC0 and RC7, COG1B signals on RC1 and RC6, COG1C signals on RC2 and RC5, and COG1D signals on RC3 and RC4. An active-low input on RB0 will trigger shut down on the eight COG signals while at the same time, capturing the values of a Timer register. Figure 5 shows the internal peripheral-pin connections.
EXAMPLE 11: MAPPING FLEXIBILITY OF PERIPHERALS AND PINS ON A PIC16F DEVICE

/* RB0 becomes an input pin for both COGIN and CCP2 Capture Input of PIC16F1716 */
TRISB0 = 1; // RB0 becomes an input pin
ANSB0 = 0; // RB0 is a digital input pin
COGINPPS = 0b01000; // RB0 is selected as COGIN
CCP2PPS = 0b01000; // RB0 is selected as CCP2 Capture Input

// Remappable COG output signals overrides TRIS control
RC0PPS = 0b01000; // RC0 is selected as COG1A
RC7PPS = 0b01000; // RC7 is selected as COG1A
RC1PPS = 0b01001; // RC1 is selected as COG1B
RC6PPS = 0b01001; // RC6 is selected as COG1B
RC2PPS = 0b01010; // RC2 is selected as COG1C
RC5PPS = 0b01010; // RC5 is selected as COG1C
RC3PPS = 0b01101; // RC3 is selected as COG1D
RC4PPS = 0b01111; // RC4 is selected as COG1D

FIGURE 5: MULTIPLE INPUT FUNCTIONS AND MULTIPLE OUTPUT PINS ON A PIC16F DEVICE

![Diagram of multiple input functions and output pins on a PIC16F device]
The code in Example 12 shows the PPS mapping flexibility for the PIC18F devices. It sets the pin RP2 to serve two functions, as FLT0 fault input of ECCP1 and as CCP1 Capture input. It creates three ECCP1/P1A PWM signals on RP16, RP17, and RP18. An active-low signal on RP2 will shut down the three replicated CCP1/P1A PWM signals and capture the contents of the internal Timer. Figure 6 illustrates the internal peripheral-pin connections.

**EXAMPLE 12: MAPPING FLEXIBILITY OF PERIPHERALS AND PINS ON A PIC18F DEVICE**

```
TRISAbits.TRISA5 = 1; // make RP2 (RA5) an input
ANCON0bits.PCFG4 = 1; // make RP2 (RA5) a digital input, disable analog input AN4
RPINR24 = 2; // RP2 (RA5) goes to FLT0 of ECCP1 PWM
RPINR8 = 2; // RP2 (RA5) goes to CCP2 Capture Input
TRISC = 0; // make PORTC as output
LATC = 0; // clear PORTC first
RPOR16 = 14; // ECCP1/P1A (Function 14) output is remapped to pin RP16 (RC5)
RPOR17 = 14; // ECCP1/P1A (Function 14) output is remapped to pin RP17 (RC6)
RPOR18 = 14; // ECCP1/P1A (Function 14) output is remapped to pin RP18 (RC7)
```

**FIGURE 6: MULTIPLE INPUT FUNCTIONS AND MULTIPLE OUTPUT PINS ON A PIC18F DEVICE**
USING THE MPLAB® CODE CONFIGURATOR FOR PERIPHERAL PIN SELECTION

The MPLAB® Code Configurator (MCC) provides a quick and easy method of configuring and initializing different peripherals, including the PPS. It includes a visual approach on selecting the pin for peripheral input/output. A click on a blue lock symbol on a pin from the Pin Manager window of the MCC selects that particular pin. Clicking the Generate Code button automatically creates the needed code. The pin_manager.c file contains all the information regarding the pins.

The following procedure shows how to use the MPLAB Code Configurator for assigning pin functions in the PIC16F1716 device. In this example, the pin functions for the PIC16F1716’s COG peripheral will be assigned to the following pins: COG1A on RC0, COG1B on RC1, COG1C on RC2, COG1D on RC3, CCP1 PWM out on RC5 and COGIN on RC4.

1. On the main MPLABX project, go to Tools- >Embedded->MPLAB Code Configurator, see Figure 7.

2. After selecting the COG and CCP1 PWM from the Device Resources area, go into the Pin Manager window. There is a list of possible pin assignments for each peripheral (see Figure 8). Note that an available pin for the peripheral is displayed as an open blue lock.
3. Select the pins, COG1D on RC3, COG1C on RC2, COG1B on RC1, COG1A on RC0, CCP1 output on RC5, COGIN on RC4, by just clicking on the respective lock for that pin. Once selected, the lock closes and turns green. Observe the change in name on the pin diagrams from the Pin Manager window. This is shown in Figure 9.

4. Click the **Generate Code** Button on the top left corner of the Composer Area (See Figure 10). This will add a `pin_manager.c` file to the MPLAB X project. The code in Example 13 shows the generated `pin_manager.c` file, in which the PPS is configured automatically.
FIGURE 10: GENERATE CODE BUTTON ON THE COMPOSER AREA

EXAMPLE 13: PIN MANAGER.C FILE GENERATED BY MPLAB® CODE CONFIGURATOR

GIE = 0;
PSPLOCK = 0x55;
PSPLOCK = 0xAA;
PSPLOCK = 0x00; // unlock PPS
RC0PPSbits.RC0PPS = 0x08; // RC0->COG:COG1A
RC1PPSbits.RC1PPS = 0x09; // RC1->COG:COG1B
RC2PPSbits.RC2PPS = 0x0A; // RC2->COG:COG1C
RC3PPSbits.RC3PPS = 0x0B; // RC3->COG:COG1D
COGINPPSbits.COGINPPS = 0x14; // RC4->COG:COGIN
RC5PPSbits.RC5PPS = 0x0C; // RC5->CCP1:CCP1
PSPLOCK = 0x55;
PSPLOCK = 0xAA;
PSPLOCK = 0x01; // lock PPS
GIE = state;

CONCLUSION

The Peripheral Pin Select (PPS) increases the flexibility of the Microchip’s 8-bit microcontrollers by providing an option for the developer to customize pin layouts of the device. This technical brief highlights the advantage of PPS and gives an overview on how PPS is configured and implemented for 8-bit MCUs.
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.

Information contained in this publication regarding device applications and the like is provided only for your convenience and may be superseded by updates. It is your responsibility to ensure that your application meets with your specifications. MICROCHIP MAKES NO REPRESENTATIONS OR WARRANTIES OF ANY KIND WHETHER EXPRESS OR IMPLIED, WRITTEN OR ORAL, STATUTORY OR OTHERWISE, RELATED TO THE INFORMATION, INCLUDING BUT NOT LIMITED TO ITS CONDITION, QUALITY, PERFORMANCE, MERCHANTABILITY OR FITNESS FOR PURPOSE. Microchip disclaims all liability arising from this information and its use. Use of Microchip devices in life support and/or safety applications is entirely at the buyer’s risk, and the buyer agrees to defend, indemnify and hold harmless Microchip from any and all damages, claims, suits, or expenses resulting from such use. No licenses are conveyed, implicitly or otherwise, under any Microchip intellectual property rights.

**Trademarks**

The Microchip name and logo, the Microchip logo, dsPIC, FlashFlex, flexPWR, JukeBox, KEELOG, KEELOG logo, Kleer, LANCheck, MediaLB, MOST, MOST logo, MPLAB, OptoLyzer, PIC, PICSTART, PIC18® logo, RightTouch, SpyNIC, SST, SST Logo, SuperFlash and UNI/O are registered trademarks of Microchip Technology Incorporated in the U.S.A. and other countries.

The Embedded Control Solutions Company and mTouch are registered trademarks of Microchip Technology Incorporated in the U.S.A.

Analog-for-the-Digital Age, BodyCom, chipKIT, chipKIT logo, CodeGuard, dsPICDEM, dsPICDEM.net, ECAN, In-Circuit Serial Programming, ICSP, Inter-Chip Connectivity, KleerNet, KleerNet logo, MiWi, MPASM, MFP, MPLAB Certified logo, MPLIB, MPLINK, MultiTRAK, NetDetach, Omniscient Code Generation, PICDEM, PICDEM.net, PICkit, PICtail, RightTouch logo, REAL ICE, SQI, Serial Quad I/O, Total Endurance, TSHARC, USBCheck, VariSense, ViewSpan, WiperLock, Wireless DNA, 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.

Silicon Storage Technology is a registered trademark of Microchip Technology Inc. in other countries.

GestIC is a registered trademarks of Microchip Technology Germany GmbH & Co. KG, a subsidiary of Microchip Technology Inc., in other countries.

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

© 2015, Microchip Technology Incorporated, Printed in the U.S.A., All Rights Reserved.
ISBN: 978-1-63277-105-6
# Worldwide Sales and Service

**AMERICAS**

<table>
<thead>
<tr>
<th>City</th>
<th>Address</th>
<th>Telephone</th>
<th>Fax</th>
</tr>
</thead>
<tbody>
<tr>
<td>American</td>
<td>2355 West Chandler Blvd.</td>
<td>480-792-7200</td>
<td>480-792-7277</td>
</tr>
<tr>
<td>Chandler, AZ</td>
<td>85224-6199</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Technical</td>
<td>Support</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Web Address</td>
<td><a href="http://www.microchip.com">www.microchip.com</a></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**EUROPE**

<table>
<thead>
<tr>
<th>City</th>
<th>Address</th>
<th>Telephone</th>
<th>Fax</th>
</tr>
</thead>
<tbody>
<tr>
<td>Austria</td>
<td>Wels</td>
<td>43-7242-2244-39</td>
<td>43-7242-2244-39</td>
</tr>
<tr>
<td>Denmark</td>
<td>Copenhagen</td>
<td>45-4450-2828</td>
<td>45-4485-2829</td>
</tr>
<tr>
<td>Germany</td>
<td>Dusseldorf</td>
<td>49-2129-3766400</td>
<td></td>
</tr>
<tr>
<td>Germany</td>
<td>Munich</td>
<td>49-89-627-144-0</td>
<td>49-89-627-144-44</td>
</tr>
<tr>
<td>Germany</td>
<td>Pforzheim</td>
<td>49-7231-424750</td>
<td></td>
</tr>
<tr>
<td>Italy</td>
<td>Milan</td>
<td>39-0331-742611</td>
<td>39-0331-466781</td>
</tr>
<tr>
<td>Italy</td>
<td>Venice</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Netherlands</td>
<td>Drunen</td>
<td>31-416-690399</td>
<td>31-416-690340</td>
</tr>
<tr>
<td>Poland</td>
<td>Warsaw</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Spain</td>
<td>Madrid</td>
<td>34-91-708-08-90</td>
<td>34-91-708-08-91</td>
</tr>
<tr>
<td>Sweden</td>
<td>Stockholm</td>
<td>46-8-5090-4654</td>
<td></td>
</tr>
<tr>
<td>UK</td>
<td>Wokingham</td>
<td>44-118-921-5800</td>
<td>44-118-921-5820</td>
</tr>
</tbody>
</table>

**ASIA/PACIFIC**

<table>
<thead>
<tr>
<th>City</th>
<th>Address</th>
<th>Telephone</th>
<th>Fax</th>
</tr>
</thead>
<tbody>
<tr>
<td>Asia Pacific</td>
<td>Office</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Hong Kong</td>
<td>3707-14, 37th Floor, Tower 6, The Gateway, Harbour City, Kowloon</td>
<td>852-2943-5100</td>
<td>852-2401-3431</td>
</tr>
<tr>
<td>China</td>
<td>Xiamen</td>
<td>86-592-2388138</td>
<td>86-592-2388130</td>
</tr>
<tr>
<td>China</td>
<td>Zhuhai</td>
<td>86-756-3210040</td>
<td>86-756-3210049</td>
</tr>
<tr>
<td>India</td>
<td>Bangalore</td>
<td>91-80-3090-4444</td>
<td>91-80-3090-4443</td>
</tr>
<tr>
<td>India</td>
<td>New Delhi</td>
<td>91-11-4160-8631</td>
<td>91-11-4160-8632</td>
</tr>
<tr>
<td>Japan</td>
<td>Osaka</td>
<td>81-6-6152-7160</td>
<td>81-6-6152-9310</td>
</tr>
<tr>
<td>Japan</td>
<td>Tokyo</td>
<td>81-3-6880-3770</td>
<td>81-3-6880-3771</td>
</tr>
<tr>
<td>Korea</td>
<td>Daegu</td>
<td>82-53-744-4301</td>
<td>82-53-744-4302</td>
</tr>
<tr>
<td>Korea</td>
<td>Seoul</td>
<td>82-2-554-7200</td>
<td>82-2-558-5932</td>
</tr>
<tr>
<td>Malaysia</td>
<td>Kuala Lumpur</td>
<td>60-3-6201-9857</td>
<td>60-3-6201-9859</td>
</tr>
<tr>
<td>Malaysia</td>
<td>Penang</td>
<td>60-4-227-8870</td>
<td>60-4-227-4068</td>
</tr>
<tr>
<td>Philippines</td>
<td>Manila</td>
<td>63-2-634-9065</td>
<td>63-2-634-9069</td>
</tr>
<tr>
<td>Singapore</td>
<td></td>
<td>65-6334-8870</td>
<td>65-6334-8850</td>
</tr>
<tr>
<td>Taiwan</td>
<td>Hsin Chu</td>
<td>886-3-5778-366</td>
<td>886-3-5770-955</td>
</tr>
<tr>
<td>Taiwan</td>
<td>Kaohsiung</td>
<td>886-7-213-7828</td>
<td></td>
</tr>
<tr>
<td>Thailand</td>
<td>Bangkok</td>
<td>66-2-694-1351</td>
<td>66-2-694-1350</td>
</tr>
</tbody>
</table>

**ASIA/PACIFIC**

<table>
<thead>
<tr>
<th>City</th>
<th>Address</th>
<th>Telephone</th>
<th>Fax</th>
</tr>
</thead>
<tbody>
<tr>
<td>Atlanta</td>
<td>Duluth, GA</td>
<td>678-957-9614</td>
<td>678-957-1455</td>
</tr>
<tr>
<td>Austin, TX</td>
<td>512-257-3370</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Boston</td>
<td>774-760-0007</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Chicago</td>
<td>Itasca, IL</td>
<td>630-285-0071</td>
<td>630-285-0075</td>
</tr>
<tr>
<td>Cleveland</td>
<td>Independence, OH</td>
<td>216-447-0464</td>
<td>216-447-0643</td>
</tr>
<tr>
<td>Dallas</td>
<td>Addison, TX</td>
<td>972-818-7423</td>
<td>972-818-2924</td>
</tr>
<tr>
<td>Detroit</td>
<td>Novi, MI</td>
<td>248-848-4000</td>
<td></td>
</tr>
<tr>
<td>Houston, TX</td>
<td>281-894-5983</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Indianapolis</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Noblesville, IN</td>
<td>Tel: 317-773-8323</td>
<td>Fax: 317-777-5453</td>
<td></td>
</tr>
<tr>
<td>Los Angeles</td>
<td>Mission Viejo, CA</td>
<td>949-462-9523</td>
<td>949-462-9608</td>
</tr>
<tr>
<td>New York, NY</td>
<td>631-435-6000</td>
<td></td>
<td></td>
</tr>
<tr>
<td>San Jose, CA</td>
<td>408-735-9110</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Canada - Toronto</td>
<td>Tel: 905-673-0699</td>
<td>Fax: 905-673-6509</td>
<td></td>
</tr>
</tbody>
</table>

© 2015 Microchip Technology Inc.