INTRODUCTION

Most of the embedded applications require real-time communications to support their application environment. Wired standards of communication, such as RS232, RS422, RS485 or Ethernet are not easily implemented due to the infrastructure support required in the end user application. Wi-Fi® and Bluetooth® have emerged as the standards of choice for connecting local embedded applications to the cloud through a router, smartphone, or tablet.

Wi-Fi, based on the IEEE 802.11 b/g/n/ac, is a standard feature in laptops, smart phones, smart machines and many other applications. Wi-Fi provides substantial bandwidth for data transfer; however, it is heavy in protocol stack and power consumption.

Bluetooth is characterized by easy, temporary connectivity to smartphones and tablets, and is currently supported in many Android® and iOS® applications. It provides a convenient cable replacement for applications, such as audio streaming and data synchronization between devices. Initially Bluetooth supported 1 Mbps data transfer (Bluetooth 1.2), and later the data transfer rate has increased to 3 Mbps with the Enhanced Data Rate version (Bluetooth 2.1 + EDR), and further advanced to a high-speed version (Bluetooth 3.0 + HS) to support large file transfers.

Bluetooth Classic and Bluetooth Low Energy (BLE) technology are different, hence users must consider the technology which meets the requirements of the applications. However, both Bluetooth Classic and BLE have found their presence with IoT that requires ease of network connectivity by enabling physical objects or devices to connect and exchange data. Many applications only need simple command and control, or a quick status from a sensor. Therefore, BLE has evolved to support these low-duty cycle applications.

The primary purpose of this application note is to help users or application developers to have a quick understanding of the interface requirements and the process of communication between the RN4020 module and PIC24 (16-bit) microcontroller over the UART using the ASCII command interface.

It essentially supports BLE applications and application developers with an interface framework in using the Microchip Bluetooth module and MCU which are suitable for IoT and related applications.

This application note also showcases the low data rate transfer when a BLE private service known as MLDP (Microchip Low-energy Data Profile) is used to transfer data.

BLUETOOTH SMART COMMUNICATION

Bluetooth Low Energy, also known as Bluetooth Smart, is a hallmark in the Bluetooth 4.0 specification. BLE is intended for energy-constrained applications, such as sensors or disposable devices. BLE is intended for low-duty cycle devices that support low-data throughput and can operate for a longer duration compared to other protocols from a coin cell battery. The key benefits in implementing the technology are inexpensive silicon, light MCU processing requirements, and reduced memory. It also suitable for applications that are related to the body area network (BAN), which represents a ‘connectivity bubble’ that moves along with the individual’s network.

BLE operates in the same spectrum band (i.e 2.400 GHz to 2.4835 GHz ISM band) as Classic Bluetooth technology, but uses a different set of channels. Classic Bluetooth has 79 channels, each of 1 MHz wide, whereas BLE has 40 channels, each of 2 MHz wide. Within a channel, data is transmitted using Gaussian Frequency Shift Keying (GFSK) modulation technique, which is similar to Classic Bluetooth's FSK modulation. The maximum over-the-air bit rate is 1 Mbps, and the maximum transmit power is 10 mW. For additional information on Bluetooth and its specifications, refer to the “Bluetooth Core Specification V4.0” document which is available or download from the following website: http://www.bluetooth.org.

The Microchip RN4020 module is a fully-certified Bluetooth version 4.1 module for adding a low-power wireless capability to applications and products. The surface mounted module has a complete Bluetooth stack on-board and is controlled through simple ASCII commands over the UART interface.
The RN4020 also includes most of the Bluetooth SIG standard profiles and Microchip Low-energy Data Profile (MLDP) for user specific serial data transfer over BLE. Developers can utilize the scripting features to enable standalone operation without a host microcontroller or processor. The RN4020 can be remotely controlled or updated over-the-air by another module over a secure connection, or can be controlled or updated through its UART interface.

Figure 1 illustrates the RN4020 module mounted on a PICtail™ board.

FIGURE 1: RN4020 MODULE MOUNTED ON A PICTAIL™ BOARD

This application note provides the following information:

- Framework for any of the user application platform using the RN4020 module and PIC24FJ series of microcontrollers.
- Specifically an interface of the RN4020 module with PIC24FJ128GA010 microcontroller.
- Start up or reference code to command the RN4020 module through PIC® microcontroller.
- Initial procedures or techniques for interfacing a PIC microcontroller and BLE module.

Note: This application note is not intended to provide a complete understanding of the Bluetooth technology principles or using the commands related to the RN4020 module.
RN4020 MODULE AND PIC24 MCU INTERFACE FRAMEWORK

The primary purpose of this application note is to help users or developers to have a quick understanding of the interface requirements and process of communication using commands between the RN4020 modules and PIC24 (16-bit) microcontroller over UART interface.

Figure 2 illustrates the PIC24 MCU interface with the RN4020 module. The inputs or commands are given through the switches available on the Explorer 16 Development board, and the status is monitored through on-board LCDs and LEDs.

FIGURE 2: SMART COMMUNICATION APPLICATION DIAGRAM

Application Demo Requirements

This section describes the hardware and software utilities required for the demo setup.

HARDWARE REQUIREMENTS

- Two BLE RN4020 PICtail™/PICtail Plus daughter boards
- Two Explorer 16 Development Boards
- Any one of the following Microchip development tools for programming and debugging: MPLAB® REAL ICE™, MPLAB® ICD 3, or PICkit™ 3
- Two power supplies: 9V/0.75A or equivalent battery packs

Note: The Explorer 16/32 Development Board can also be used in place of the Explorer 16 Development Board.

SOFTWARE/UTILITY REQUIREMENTS

The demo application which is based on the RN4020 module is used to showcase communication between the BLE-based embedded nodes. In the demo application, nodes can emulate a sensor, actuator system, fitness device, healthcare gadget and so on.

Alternatively terminal emulator programs (for example, TeraTerm for Windows OS and CoolTerm for MAC OS) can be used to control and monitor the RN4020 module. This is to do some independent tests using the RN4020 PICtail card.

The application or demo source code, comes along with this application note, is available as MPLAB workspace which is available for download from the Microchip web site: http://www.microchip.com. The source code is compiled using the Microchip XC16 compiler through MPLABX IDE. The demo uses commands which are initiated by the switches on one of the Explorer 16 Development board and the result is displayed on another Explorer 16 Development board’s LCD.
HARDWARE DEMO SETUP
The RN4020 demo requires two wireless nodes. The demo setup is done by using two Explorer 16 Development boards with identical RN4020 module mounted on the PICtail board. Thus, two identical RN4020 module-based wireless nodes are used for this application demonstration. For more information on the RN4020 module, go to the Microchip web site: http://www.microchip.com.

Explorer 16 Development Board and RN4020 Module Connections
The RN4020 module-based PICtail daughter board's 30-pin PCB-edge connector (J3) is used to connect the Explorer 16 Development board's PICtail plus connector. This connection supplies 3.3V power, two/four wire UART, Reset, wake and interrupt connections to the RN4020 module from the Microcontroller. Figure 4 illustrates the plug-in arrangement between the Explorer 16 Development board and the RN4020 module. For more information on using and programming the Explorer 16 Development board with the RN4020 module, refer to the “RN4020 PICtail™/PICtail Plus Daughter Board User’s Guide” (DS50002265).

**Note:** Use the Explorer 16/32 Development Board for the PIC24 platform development boards for new designs or advanced feature requirements. The board offers compatibility for application code or firmware developed using the Explorer 16 Development Board and related PICtail/PICtail Plus boards. For additional information, go to the Microchip web site: http://www.microchip.com/explorer16.
Figure 4 illustrates the RN4020 module mounted on the Explorer 16 Development board.

**FIGURE 4: RN4020 MODULE MOUNTED ON EXPLORER 16 DEVELOPMENT BOARD**

---

**GETTING STARTED**

To set up the RN4020 BLE module as a wireless node, perform the following actions:

1. Insert the RN4020 PICtail card into the Explorer 16 Development board's 30-pin card edge connector. The Explorer 16 Development board can be a base board for the RN4020 PICtail Daughter card.
2. Display on the Explorer 16 Development board’s LCD can be used for configuring or monitoring the wireless terminals.
3. Plug-in the 9V power supply to the base board.
4. Connect the programmer or debugger (MPLAB ICD3/PICKIT3) to the Explorer 16 Development board.
6. In Release mode, compile two source codes which are related to the application demo.
7. The generated .hex files can be programmed into two wireless nodes, node A and node B, using the MPLAB ICD3 or any other programmer available with the user.

**Note:** Users can use the generated .hex files which are available in the source code WinZip file.

For additional information on programming and debugging with MPLAB ICD 3, refer to the “MPLAB® ICD 3 User’s Guide for MPLAB X” (DS50002081) and for Explorer 16 Development Board, refer to the “Explorer 16 Development Board User’s Guide” (DS51589).

**Appendix A:** provides details of the source code, related files with description, and call graphs of the main() functions associated to Central and Peripheral nodes. However, users can generate call graphs related to specific functions of the source code for their understanding.
RN4020 Demo Application

The execution of the RN4020 demo application involves these steps:

1. Configuring one node as Central and another node as Peripheral.
2. Switch ON or OFF to toggle LEDs (D7 and D8) on the Node B when switches (S3 and S6) on the Node A are pressed, and vice versa.

Configuring the Node A as Central

Using the demo code, the first RN4020 module (referred to as module A) is configured as Central (Node A). The following commands are used to configure the device:

1. Pull WAKE_SW high to enter command mode. (i.e., default state on the daughter board).
2. Open a terminal emulator that connects the serial port of the module A with the following parameters:
   - Baud rate: 115200
   - Data bits: 8
   - Parity: none
   - Stop bits: 1
   - Flow control: hardware
3. SF,1 // factory reset.
4. SR,92000000 // set device as central, support MLDP and enable UART flow control.
5. R,1 // reboot to make the changes effective.

Note: Flow control is required when MLDP is enabled. The UART flow control (with SR,02000000) setting is used to control the RTS/CTS hardware flow control on the UART port of the RN4020 module. If set, flow control is enabled and the host needs to support the UART hardware flow control feature.
Figure 6 and Figure 7 illustrate the configuration and scanning of Central (Node A).

**FIGURE 6:** BLE CENTRAL NODE IN CONFIGURATION MODE

**FIGURE 7:** BLE CENTRAL NODE IN SCANNING MODE
Figure 8 illustrates the BLE Central node demo showing switches for scroll and select. Figure 9 illustrates the BLE Central node in the connected mode.

**FIGURE 8: BLE CENTRAL NODE IN SCROLL AND SELECT MODE**

**FIGURE 9: BLE CENTRAL NODE IN CONNECTED MODE**
Configuring the Node B as Peripheral

Using the demo code, the second RN4020 module (referred to as module B) is configured as Peripheral (Node B). Use the following commands to configure the device:

1. Pull WAKE SW high to enter Command mode. (i.e., default state on the daughter board).
2. Open a terminal emulator that connects the serial port of the module B with the following parameters:
   - Baud rate: 115200
   - Data bits: 8
   - Parity: none
   - Stop bits: 1
   - Flow control: hardware
3. SF,1 // factory reset.
4. SR,32000000 // set device as peripheral with automatic advertisement, and support for MLDP and flow control features.
5. R,1 // Reboot the device to make the changes effective.

Figure 10 illustrates the BLE Peripheral (Node B) flowchart.

**Note:** Flow control is required when MLDP is enabled. The UART flow control (with SR,02000000) setting is used to control the RTS/CTS hardware flow control on the UART port of the RN4020 module. If set, flow control is enabled and the host needs to support the UART hardware flow control feature.
Figure 11 and Figure 12 illustrate the configuration and auto advertising of the BLE Peripheral (Node B).

FIGURE 11: BLE PERIPHERAL NODE IN CONFIGURATION MODE

FIGURE 12: BLE PERIPHERAL NODE IN AUTO ADVERTISING MODE
Figure 13 illustrates the BLE Peripheral node when connected.

FIGURE 13: BLE PERIPHERAL NODE IN CONNECTED MODE
Connecting Two Devices

When the module B is powered up, it automatically starts advertising because the auto advertisement feature is enabled using the "SR" command, and then the module A discovers the module B using the "F" command:

F // Start scan

The scan result will display three elements: MAC address, MAC address type, and device name (for example, 00035B0358E6,0,MCHP-LE,-50).

Input the "X" command followed by an "E" command to stop scanning and establishing the connection:

X // Stop scanning

E,0,00035B0358E6 // Try to establish connection with the device of public MAC address, 0x00035B0358E6.

Figure 14 illustrates the application demo diagram using the terminal emulator interface.

FIGURE 14: APPLICATION DEMO DIAGRAM WITH LCD SWITCH INTERFACE
MLDP Mode Details

MLDP mode is entered by setting the CMD/MLDP pin high, all data from the UART is sent to the peer device as a data stream. To exit MLDP mode, the CMD/MLDP pin must be set low so that the RN4020 module is returned to Command mode by outputting “CMD” to the UART. The CMD/MLDP pin (pin 8) is used to control the RN4020 module when an MLDP serial data service is used. For more information on MLDP Commands, refer to the “RN4020 Bluetooth Low Energy Module User’s Guide” (DS70005191).

After the access and characteristics in public services are verified, the MLDP service can be started. The MLDP service is built on top of the private service, but acts transparently by routing binary data read and write to a private characteristic through the UART. To use the MLDP service between two RN4020 devices, both the devices must enable MLDP with the “SR” command. The MLDP mode can only be started when two RN4020 modules are MLDP-enabled and are connected to each other.

To start MLDP mode, perform the following actions:

1. Assert the CMD/MLDP pin to be high. The RN4020 module will acknowledge with “MLDP” string to indicate the start of the MLDP mode.
2. Once in the MLDP mode, any data from the UART will be sent to the peer device.
3. While receiving the MLDP data from the peer, if the AUTO_MLDP_DISABLE feature is disabled, the RN4020 module will automatically enter the MLDP mode; otherwise, all data will be ignored until the CMD/MLDP pin is set high to enter MLDP mode.
4. From the module A, assert the CMD/MLDP pin to be high and wait until “MLDP” is output to the UART. Provided module B shows “MLDP”, anything entered on the UART of the module A will be displayed on the UART of the module B, and vice versa.

Note: To exit MLDP mode, set the CMD/MLDP pin to be low and the module will acknowledge with the “CMD” string, which appears on the UART indicating that the RN4020 module is back in command mode. A host microcontroller is required to interpret the command and responses available on the UART of the modules.

5. Set the CMD/MLDP pin to be low on module B (the WAKE_HW and CMD/MLDP pins have weak pull-down resistors hence they will stay low if not pulled high).
6. On module A, the status change will be notified to the host. However, module A is currently in the MLDP mode and only output MLDP data is sent to the UART. Instead, PIO2 will be set high (the red LED (MLDP_EV) illuminates on the RN4020 PICtail Board) to indicate the pending status message to be sent over the UART to the host microcontroller.
7. Once the CMD/MLDP pin is set low to enter command mode, the status message will be the output to the UART. The maximum status message that can be buffered is 256 bytes.
Running the Demo

As mentioned earlier, configure one node as Peripheral and another node as Central as required by the BLE platform for communication. Use the following procedure to program, connect, and test two wireless BLE nodes.

1. Program one board with the BLE_Peripheral.X.production.hex file and another board with the BLE_Central.X.production.hex file.
2. After two boards are programmed, the Peripheral device will be in the Auto-Advertising mode and the Central device will be in the Scan mode.
3. The Central node has an LCD for monitoring and two switches for scrolling and selecting.
4. Scroll for the specific Peripheral node from the Central node using the switch SW3.
5. Select the Peripheral node using the switch SW6.
6. Upon selection, the Peripheral node will connect to the Central node and the green (CONN) LED on the RN4020 PICtail board turns ON, which indicates the connection.
7. The modules then switch to the MLDP mode.
8. Press the switch S3 or S6 from the Peripheral node or Central node. The LED D9 or D10 toggle on the Central or Peripheral node.

Figure 15 illustrates the Central and Peripheral nodes which are connected in the MLDP mode.

FIGURE 15: CENTRAL AND PERIPHERAL NODES CONNECTED IN MLDP MODE
Conclusion
This application note is designed to help Microchip Bluetooth customers to acquire a basic understanding of the Bluetooth Low Energy (BLE) and hence provide communication between the two Microchip BLE RN4020 modules. This application note is supported by the sample code or demo source code for enabling the RN4020 modules, as Central node and Peripheral node through 16-bit PIC microcontroller, using the ASCII commands supported over the UART. The interface hardware details and source code can be used further as a framework for any of the user applications or custom projects.

The RN4020 module supports only BLE as the communication mode. However, users who intend to work with dual-mode Microchip Bluetooth devices or modules which include Bluetooth Classic and Low Energy for application purposes, refer to the application note AN2244, “Bluetooth Data Transfer over SPP and Transparent UART Service Using Microchip RN4677 Dual-Mode Module”.

References
- Bluetooth 4.1 GATT definitions browser: https://developer.bluetooth.org/gatt/Pages/Definition-Browser.aspx
- "RN4020 Bluetooth Low Energy Module Data Sheet" (DS50002279)
- "RN4020 Bluetooth Low Energy Module User's Guide" (DS70005191)
- "RN4020 PICtail™/PICtail Plus Board User's Guide" (DS50002265)
- "Explorer 16 Development Board User's Guide" (DS50001589)
- "PIC24FJ128GA010 Family Data Sheet" (DS39747)
- "MPLAB ICD 3 USER'S GUIDE FOR MPLAB X IDE" (DS50002081)
- "Bluetooth® Data Transfer over SPP and Transparent UART Service Using Microchip RN4677 Dual-Mode Module" (DS00002244)

REVISION HISTORY

Revision A Document (2014)
This is the initial released version of the document.

Revision B Document (April 2017)
Updated Section “Introduction”, Section “Bluetooth smart communication”, Section “RN4020 module and PIC24 MCU Interface Framework”, and Section “Conclusion”.

Additional minor corrections, such as language and formatting updates were incorporated throughout the document.
APPENDIX A:

This section provides details of the source code, related source files with description, and call graphs of the `main()` function associated to Central and Peripheral nodes.

Source Code

**Software License Agreement**

The software supplied herewith by Microchip Technology Incorporated (the “Company”) is intended and supplied to you, the Company's customer, for use solely and exclusively with products manufactured by the Company. The software is owned by the Company and/or its supplier, and is protected under applicable copyright laws. All rights are reserved. Any use in violation of the foregoing restrictions may subject the user to criminal sanctions under applicable laws, as well as to civil liability for the breach of the terms and conditions of this license.

THIS SOFTWARE IS PROVIDED IN AN “AS IS” CONDITION. NO WARRANTIES, WHETHER EXPRESS, IMPLIED OR STATUTORY, INCLUDING, BUT NOT LIMITED TO, IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE APPLY TO THIS SOFTWARE. THE COMPANY SHALL NOT, IN ANY CIRCUMSTANCES, BE LIABLE FOR SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES, FOR ANY REASON WHATSOEVER.

All of the software details covered in this application note is available for download as a WinZip archive file from the Microchip web site: www.microchip.com.

**Source code file list**

Table 1 provides the source files that are used as a part of the Central node (Node A).

**TABLE 1: CENTRAL NODE SOURCE FILES**

<table>
<thead>
<tr>
<th>File Name</th>
<th>File Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Central</td>
<td>.c and .h files</td>
<td>Central command state machine</td>
</tr>
<tr>
<td>Central_string</td>
<td>.c and .h files</td>
<td>Command and response strings required for Central node</td>
</tr>
<tr>
<td>Config</td>
<td>.h file</td>
<td>PIC24F device configurations</td>
</tr>
<tr>
<td>Config_fuse</td>
<td>.c file</td>
<td>Config fuses used to configure PIC24F</td>
</tr>
<tr>
<td>LCD</td>
<td>.c and .h files</td>
<td>LCD interface</td>
</tr>
<tr>
<td>Main</td>
<td>.c and .h files</td>
<td>Initialization of the PIC24F device</td>
</tr>
<tr>
<td>UART</td>
<td>.c and .h files</td>
<td>UART driver interface for RN4020</td>
</tr>
</tbody>
</table>

Table 2 provides the source files that are used as part of the Peripheral node (Node B).

**TABLE 2: PERIPHERAL NODE SOURCE FILES**

<table>
<thead>
<tr>
<th>File Name</th>
<th>File Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ADC</td>
<td>.c and .h files</td>
<td>Not used for this demo application</td>
</tr>
<tr>
<td>Config</td>
<td>.h file</td>
<td>PIC24F device configurations</td>
</tr>
<tr>
<td>Config_fuse</td>
<td>.c file</td>
<td>Config fuses used to configure PIC24F</td>
</tr>
<tr>
<td>LCD</td>
<td>.c and .h files</td>
<td>LCD interface</td>
</tr>
<tr>
<td>Main</td>
<td>.c and .h files</td>
<td>Initialization of the PIC24F device</td>
</tr>
<tr>
<td>Peripheral</td>
<td>.c and .h files</td>
<td>Peripheral command state machine</td>
</tr>
<tr>
<td>Peripheral_string</td>
<td>.c and .h files</td>
<td>Command and response strings required for Peripheral node</td>
</tr>
<tr>
<td>UART</td>
<td>.c and .h files</td>
<td>UART driver interface for RN4020</td>
</tr>
</tbody>
</table>
Source code call graphs

Figure A-1 illustrates the functions used by the Central `main()` program in the application code.

**FIGURE A-1: CENTRAL NODE CALL GRAPH**
Figure A-2 illustrates the functions used by the Peripheral main() program in the application code.

**FIGURE A-2: PERIPHERAL NODE CALL GRAPH**
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 unless otherwise stated.

Trademarks
The Microchip name and logo, the Microchip logo, AnyRate, AVR, AVR logo, AVR Freaks, BeaconThings, BitCloud, CryptoMemory, CryptoRF, dsPIC, FlashFlex, flexPWR, Helda, JukeBloxx, KElLOO, KElLOO logo, Kleer, LANCheck, LINK MD, maXStylus, maXTouch, MediaLB, megaAVR, MOST, MOST logo, MPLAB, OptoLyzer, PIC, picoPower, PICSTART, PIC32 logo, Prochip Designer, QTouch, RightTouch, SAM-BA, SpyNIC, SST, SST Logo, SuperFlash, tinyAVR, UNI/O, and XMEGA are registered trademarks of Microchip Technology Incorporated in the U.S.A. and other countries.

Microchip received ISO/TS-16949:2009 certification for its worldwide headquarters, design and wafer fabrication facilities in Chandler and Tempe, Arizona; Gresham, Oregon and design centers in California and India. The Company's quality system processes and procedures are for its PIC® MCUs and dsPIC® DSCs, KEELOQ® code hopping devices, Serial EEPROMs, microperipherals, nonvolatile memory and analog products. In addition, Microchip's quality system for the design and manufacture of development systems is ISO 9001:2000 certified.

QUALITY MANAGEMENT SYSTEM CERTIFIED BY DNV

ISO/TS 16949

© 2014-2017 Microchip Technology Inc.
## 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

**Austin, TX**
Tel: 512-257-3370

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

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

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

**Detroit**
Novi, MI
Tel: 248-848-4000

**Houston, TX**
Tel: 713-686-6867
Fax: 713-686-6868

**Indianapolis**
Noblesville, IN
Tel: 317-536-2380
Fax: 317-536-2381

**Los Angeles**
Mission Viejo, CA
Tel: 949-462-9523
Fax: 949-462-9608
Tel: 951-273-7800

**Raleigh, NC**
Tel: 919-844-7510

**New York, NY**
Tel: 631-435-6000

**San Jose, CA**
Tel: 408-735-9110
Fax: 408-436-4270

**Canada - Toronto**
Tel: 905-695-1980
Fax: 905-695-2078

### ASIA/PACIFIC

**Asia Pacific Office**
Suites 3707-14, 37th Floor
Tower 6, The Gateway
Harbour City, Kowloon

**Hong Kong**
Tel: 852-2943-5100
Fax: 852-2401-3431

**Australia - Sydney**
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 - Hangzhou**
Tel: 86-20-8755-8029
Fax: 86-20-8755-8030

**China - Hong Kong SAR**
Tel: 852-6-6152-7160
Fax: 86-21-3326-8000

**China - Shenyang**
Tel: 86-24-2334-2829
Fax: 86-24-2334-2830

**China - Shenzhen**
Tel: 86-755-8864-2200
Fax: 86-755-8203-1760

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

**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: 91-20-3019-1500

**Japan - Osaka**
Tel: 81-6-6152-7160
Fax: 81-6-6152-9310

**Japan - Tokyo**
Tel: 81-3-6880-3770
Fax: 81-3-6880-3771

**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: 886-7-213-7830

**Taiwan - Taipei**
Tel: 886-2-2508-6800
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

**Finland - Espoo**
Tel: 358-9-4520-820

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

**France - Saint Cloud**
Tel: 33-1-30-60-70-00

**Germany - Garching**
Tel: 49-8931-9700

**Germany - Haan**
Tel: 49-2129-3766400

**Germany - Heilbronn**
Tel: 49-71-31-673636

**Germany - Karlsruhe**
Tel: 49-721-625370

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

**Germany - Rosenheim**
Tel: 49-80-31-354-560

**Israel - Ra'anana**
Tel: 972-9-744-7705

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

**Italy - Padova**
Tel: 39-049-7625286

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

**Norway - Trondheim**
Tel: 47-7289-7561

**Poland - Warsaw**
Tel: 48-22-3325737

**Romania - Bucharest**
Tel: 40-21-407-87-50

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

**Sweden - Gothenberg**
Tel: 46-31-704-60-40

**Sweden - Stockholm**
Tel: 46-6-5090-4654

**UK - Wokingham**
Tel: 44-118-921-5800
Fax: 44-118-921-5820