Bluetooth® Audio Module Command Reference User's Guide

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in this publication regarding device applications and the like is provided only for your convenience and may be superseded by updates.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.

QUALITY MANAGEMENT SYSTEM CERTIFIED BY DNV

═ ISO/TS 16949 ═

Trademarks

Printed on recycled paper.

ISBN: 978-1-63276-243-6

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.

Table of Contents

Preface 5

Chapter 1. Overview and Getting Started

1.1 Overview 11
1.2 Audio and Data Bluetooth Profiles 12
1.3 Audio Settings 13
1.4 Making a Bluetooth Connection 13
1.5 Command Mode vs. Data Mode 15
1.6 GPIO Pins 19
1.7 Connecting with a Microcontroller 20
1.8 Device Firmware Upgrade for Firmware Versions 1.10 and Lower 21
1.9 Device Firmware Upgrade for Firmware Versions 1.16 and Higher 21
1.10 Status LEDs 22
1.11 Evaluation Boards and Reference Designs 22

Chapter 2. Command Reference

2.1 Command Syntax 25
2.2 Set Commands 25
2.3 Get Commands 30
2.4 Action Commands 32

Appendix A. Command Quick Reference Guide

Appendix B. Firmware Revision History

Worldwide Sales and Service 40

NOTES:

Preface

NOTICE TO CUSTOMERS

All documentation becomes dated, and this manual is no exception. Microchip tools and documentation are constantly evolving to meet customer needs, so some actual dialogs and/or tool descriptions may differ from those in this document. Please refer to our web site (www.microchip.com) to obtain the latest documentation available.

Documents are identified with a "DS" number. This number is located on the bottom of each page, in front of the page number. The numbering convention for the DS number is "DSXXXXXA", where "XXXXX" is the document number and "A" is the revision level of the document.

For the most up-to-date information on development tools, see the MPLAB® IDE on-line help. Select the Help menu, and then Topics to open a list of available on-line help files.

INTRODUCTION

This chapter contains general information that will be useful to know before using the Bluetooth® Audio Module Command Reference. Items discussed in this chapter include:

- **Document Layout**
- **Conventions Used in this Guide**
- **Warranty Registration**
- **Recommended Reading**
- **The Microchip Web Site**
- **Development Systems Customer Change Notification Service**
- **Customer Support**
- **Document Revision History**

DOCUMENT LAYOUT

This user's guide describes how to use the Bluetooth Audio Module Command Reference. The document is organized as follows:

- **Chapter 1. "Overview and Getting Started"** – This chapter introduces the Bluetooth Audio Module Command Reference and provides an overview of various features.
- **Chapter 2. "Command Reference"** – This chapter describes the command reference and syntax of the Bluetooth Audio Module Command Reference.
- **Appendix A. "Command Quick Reference Guide"** – This appendix displays tables of firmware commands and factory defaults for the Bluetooth Audio Module Command Reference.
- **Appendix B. "Firmware Revision History"** – This appendix is a revision history of the firmware.

CONVENTIONS USED IN THIS GUIDE

This manual uses the following documentation conventions:

DOCUMENTATION CONVENTIONS

Description	Represents	Examples
Arial font:		
Italic characters	Referenced books	*MPLAB® IDE User’s Guide*
	Emphasized text	...is the *only* compiler...
Initial caps	A window	the Output window
	A dialog	the Settings dialog
	A menu selection	select Enable Programmer
Quotes	A field name in a window or dialog	“Save project before build”
Underlined, italic text with right angle bracket	A menu path	*File>Save*
Bold characters	A dialog button	Click **OK**
	A tab	Click the **Power** tab
N‘Rnnnn	A number in verilog format, where N is the total number of digits, R is the radix and n is a digit	4‘b0010, 2‘hF1
Text in angle brackets < >	A key on the keyboard	Press <Enter>, <F1>
Courier New font:		
Plain Courier New	Sample source code	`#define START`
	Filenames	`autoexec.bat`
	File paths	`c:\mcc18\h`
	Keywords	`_asm, _endasm, static`
	Command-line options	`-Opa+, -Opa-`
	Bit values	`0, 1`
	Constants	`0xFF, ‘A’`
Italic Courier New	A variable argument	*file*`.o`, where *file* can be any valid filename
Square brackets [ ]	Optional arguments	`mcc18 [options]` *file* `[options]`
Curly brackets and pipe character: { | }	Choice of mutually exclusive arguments; an OR selection	`errorlevel {0|1}`
Ellipses...	Replaces repeated text	`var_name [, var_name...]`

	Represents code supplied by user	`void main (void)`
		`{ ...`
		`}`

WARRANTY REGISTRATION

Please complete the enclosed Warranty Registration Card and mail it promptly. Sending in the Warranty Registration Card entitles users to receive new product updates. Interim software releases are available at the Microchip web site.

RECOMMENDED READING

This user's guide describes how to configure Microchip Bluetooth modules. The module-specific data sheets contain current information on the module specifications. Other useful documents are listed below. The following Microchip documents are available at www.microchip.com and recommended as supplemental reference resources:

RN52 Bluetooth Audio Module Data Sheet (DS70005120)

This document provides the technical specifications for the RN52 module.

RN52 Evaluation Kit User's Guide (DS50002153)

This document describes how to use the RN52 Evaluation Kit and provides an audio demonstration.

THE MICROCHIP WEB SITE

Microchip provides online support via our web site at **www.microchip.com**. This web site is used as a means to make files and information easily available to customers. Accessible by using your favorite Internet browser, the web site contains the following information:

- **Product Support** – Data sheets and errata, application notes and sample programs, design resources, user's guides and hardware support documents, latest software releases and archived software
- **General Technical Support** – Frequently Asked Questions (FAQs), technical support requests, online discussion groups, Microchip consultant program member listing
- **Business of Microchip** – Product selector and ordering guides, latest Microchip press releases, listing of seminars and events, listings of Microchip sales offices, distributors and factory representatives

DEVELOPMENT SYSTEMS CUSTOMER CHANGE NOTIFICATION SERVICE

Microchip's customer notification service helps keep customers current on Microchip products. Subscribers will receive e-mail notification whenever there are changes, updates, revisions or errata related to a specified product family or development tool of interest.

To register, access the Microchip web site at **www.microchip.com**, click on Customer Change Notification and follow the registration instructions.

The Development Systems product group categories are:

- **Compilers** – The latest information on Microchip C compilers and other language tools. These include the MPLAB® C compiler; MPASM™ and MPLAB 16-bit assemblers; MPLINK™ and MPLAB 16-bit object linkers; and MPLIB™ and MPLAB 16-bit object librarians.
- **Emulators** – The latest information on the Microchip MPLAB REAL ICE™ in-circuit emulator.
- **In-Circuit Debuggers** – The latest information on the Microchip in-circuit debugger, MPLAB ICD 3.
- **MPLAB IDE** – The latest information on Microchip MPLAB IDE, the Windows® Integrated Development Environment for development systems tools. This list is focused on the MPLAB IDE, MPLAB SIM simulator, MPLAB IDE Project Manager and general editing and debugging features.
- **Programmers** – The latest information on Microchip programmers. These include the MPLAB PM3 device programmer and the PICkit™ 3 development programmers.

CUSTOMER SUPPORT

Users of Microchip products can receive assistance through several channels:

- Distributor or Representative
- Local Sales Office
- Field Application Engineer (FAE)
- Technical Support

Customers should contact their distributor, representative or Field Application Engineer (FAE) for support. Local sales offices are also available to help customers. A listing of sales offices and locations is included in the back of this document.

Technical support is available through the web site at: **http://support.microchip.com**

DOCUMENT REVISION HISTORY

Revision A (May 2014)

This is the initial released version of this document.

NOTES:

Chapter 1. Overview and Getting Started

1.1 OVERVIEW

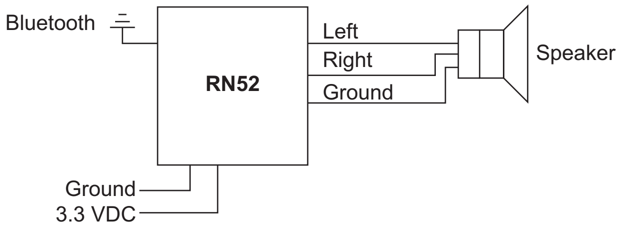
This document contains the software command reference and advanced configuration settings for Microchip’s Bluetooth audio modules. Commands and settings that are specific to a single product or product family are identified as such in the document.

> **NOTICE TO CUSTOMERS**
>
> **The commands and applications described in this document apply to Microchip’s Bluetooth *audio modules*, e.g., the RN52. They do not apply to Microchip’s Bluetooth data modules such as the RN41 or RN42. For data module configuration information, refer to the *Bluetooth Data Module Command Reference and Advanced Information User’s Guide*.**

Microchip's RN52 Bluetooth audio module offers multiple Bluetooth services in a compact form factor that can be controlled via a simple ASCII command set. The RN52 provides the following functions: A2DP sink, HFP/HSP, SPP and AVRCP. The A2DP sink function receives high-quality stereo audio from a Bluetooth audio source, such as a smartphone or MP3 media player. In addition to audio, the RN52's HFP/HSP service accepts voice calls from an Audio Gateway (AG), such as a smartphone or VOIP host software. The AVRCP profile allows the RN52 to control the volume level and to track the A2DP audio’s information. The SPP profile allows asynchronous serial data be transferred from the RN52 to a remote Bluetooth device.

An example RN52 use case is provided below in Figure 1-1. This example shows an implementation of Bluetooth wireless speakers using the RN52.

FIGURE 1-1: EXAMPLE OF BLUETOOTH® WIRELESS SPEAKERS USING THE RN52 MODULE



This document assumes that you have a working knowledge of Bluetooth operation and communications. To configure Microchip’s modules, you need a Bluetooth enabled smartphone or PC (either built-in or using a USB Bluetooth dongle). You can only configure one device at a time. Once configured, device settings are saved (independent of power down) until they are explicitly changed or the factory defaults are restored.

1.2 AUDIO AND DATA BLUETOOTH PROFILES

The audio module supports several Bluetooth profiles, as described in Table 1-1. Upon power-up, the module is configured as a slave and is ready to pair and connect. The A2DP/AVRCP profile is enabled and connected to the negotiated codec and analog I/O.

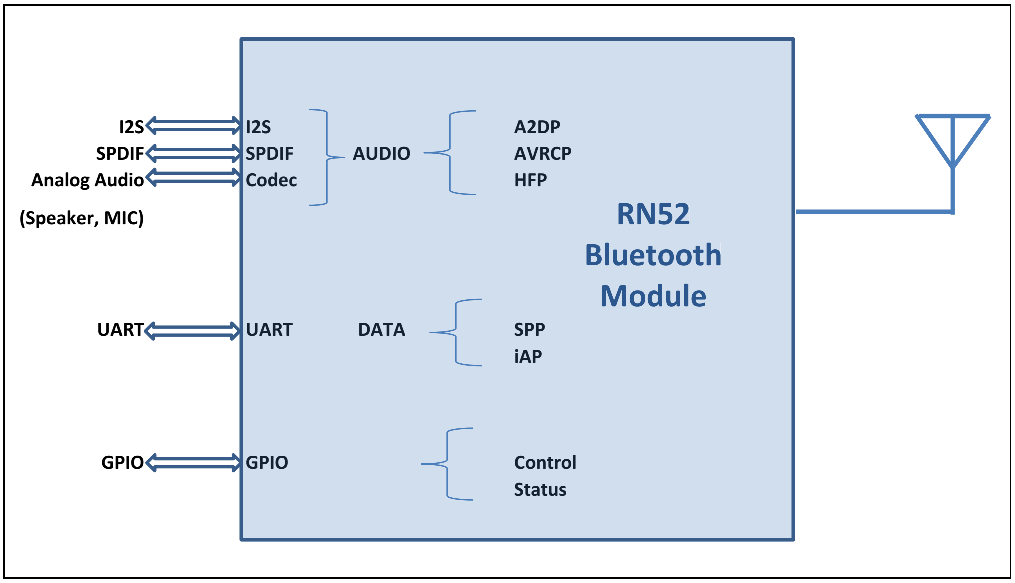
Bluetooth master devices can discover and use the profiles listed in Table 1-1. You configure each profile to be discoverable by using ASCII commands in Command mode over the data interface. The module can enable multiple profile connections simultaneously, and broadcast the profiles it has available. When you pair the module with a smartphone, the smartphone decides which profile connection(s) to use.

TABLE 1-1: SUPPORTED BLUETOOTH® PROFILES

Profile	Type	Comments
A2DP	Audio	The advanced audio distribution profile (A2DP) defines how high-quality audio (stereo or mono) can be streamed from one device to another over a Bluetooth® connection.
AVRCP	Audio	The audio-video remote control profile provides a standard interface to control audio/video equipment such as TVs and high-fidelity equipment. This profile is dependent on and used with the A2DP profile.
HFP	Audio	The Hands-Free Profile (HFP) is commonly used in car hands-free kits to communicate with mobile phones in the vehicle.
HSP	Audio	The headset profile provides support for using Bluetooth headsets with mobile phones. This profile is dependent on and used with the HFP profile.
SPP	Data	SPP defines a virtual serial port between two Bluetooth-enabled devices. SPP emulates a bidirectional serial link.
iAP	Data	The RN52 supports Wireless iAP discovery by MFi devices and provides a low level RFCOMM connection to the MFi device. An external host controller with an MFi library is required to perform authentication and iAP protocol implementation.

Figure 1-2 shows a block diagram of the RN52 with Bluetooth audio and data profiles, and GPIO pins.

FIGURE 1-2: RN52 PROFILE CONNECTION BLOCK DIAGRAM



1.3 AUDIO SETTINGS

The module’s settings can be controlled through the UART interface using a simple ASCII command language. For example, you can change the audio routing and/or profile. Set commands configure the module and get commands echo the current configuration. Because the module reads the configuration into RAM once at boot time, configuration settings modified with the set command do not take effect until the module has been rebooted (unless otherwise noted), even though the get command may show otherwise. Some example commands are:

```
SK,0008  // Set the connection profile to HFP
GA       // Display the authentication mode
GP       // Display the pin code
+        // Turn on local echo
V        // Display the firmware version
D        // Display the current settings
Q        // Show the connection status
```

1.4 MAKING A BLUETOOTH CONNECTION

By default, the Bluetooth module acts as a Bluetooth slave and the PC or smartphone is the Bluetooth master. You connect to the Bluetooth module using the Bluetooth device manager, which varies depending on your smartphone or computer’s operating system. In all cases, the process is the same:

- *Discovery* – In the discovery phase, the Bluetooth module broadcasts its name, profile support, and Bluetooth address. It is ready for other devices to pair with it.
- *Pairing* – During pairing, the Bluetooth module and the Bluetooth master validate the pin code. If the pin code validates successfully, they exchange security keys and a channel hopping pseudo-random sequence. Successful pairing results in the module and master establishing link keys.
- *Connecting* – Before connecting, the Bluetooth devices must have paired successfully. The master initiates a connection, the master and slave validate the link keys, and a Bluetooth link is established.

The following sections describe these processes in detail.

1.4.1 Discovery

Upon power up, the module can be set to enter discoverable mode. See Figure 1-3 for Bluetooth device scanning examples. Bluetooth devices, such as smartphones and PCs, can discover the module.

- *Discovery using a smartphone* – Open the Bluetooth settings on your smartphone and scan for devices available to pair and connect. The device discovers the module and displays it as **RN52-*XXXX***, where *XXXX* is the last 4 digits of the module’s MAC address. If not already paired, the module displays in the available devices list as **Not Paired.**
- *Discovery using a PC* – Open your PC’s Bluetooth device manager and choose to add a new device. The Bluetooth device manager’s icon is generally located in the bottom right corner of your screen in the taskbar for Windows and in the upper right corner for Mac OS-X. The Bluetooth device manager displays a list of discoverable Bluetooth devices. The module displays as **RN52-*XXXX***, where *XXXX* is the last 4 digits of the module’s MAC address.

The module's label also shows its MAC address. The display active configuration command also displays the MAC address of the module and the MAC address of the connected device.

1.4.2 Pairing

The module supports the following pairing methods:

- Legacy Pairing using PIN code
- Secure Simple Pairing (SSP)
  - Just Works
  - Numeric Comparison using Yes/No Keyboard IO

The RN52 module can be configured through the UART command console to use either of the above pairing methods.

To pair with the module, click its name in the Bluetooth device scan list. The default authentication mode is Just Works, which requires no user interaction. The link key is generated, stored and a connection gets established. For subsequent connections the stored link keys are compared and the connection is established. The module's firmware automatically stores up to 8 pairings from remote hosts in a first in, first out fashion.

If the remote Bluetooth device does not require authentication, a connection can occur without the pairing process. However, the Bluetooth specification requires that if either device involved in the pairing process requires authentication, the other device must participate to ensure a secure link. Microchip's modules default to Just Works mode.

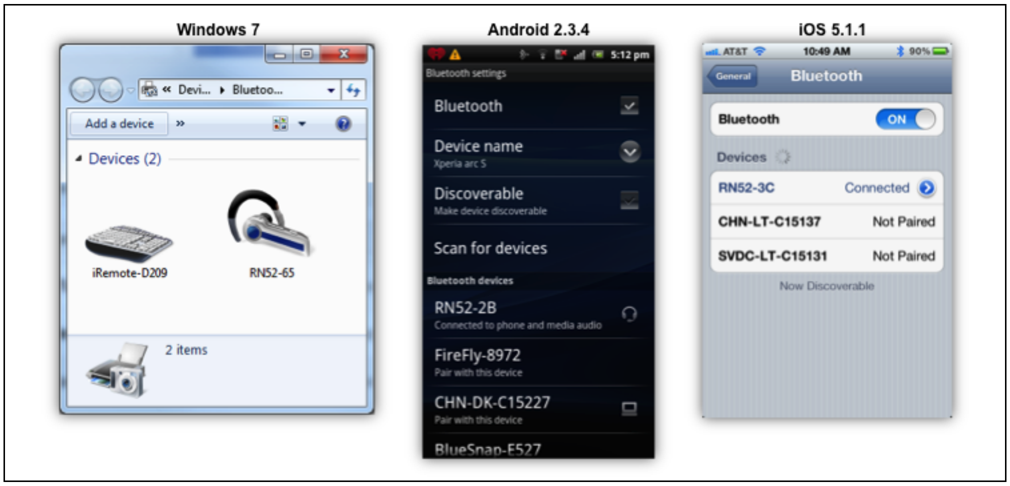
In numeric comparison using yes/no keyboard IO mode, the user is shown a 6-digit number on both the module console and the pairing device display. The user has to verify that the numbers match or don't match on both devices by entering a yes or no response. A command is provided to enter a yes or no response on the module command console. The module can also be configured to send an accept response automatically.

In legacy pairing mode, the pairing device requests a 4-digit fixed PIN for authentication. In this case, use the module's default PIN code, 1234, as the pass key. After you enter the PIN code, the Bluetooth devices compare them. If they match, a link key is generated and stored. Usually, but not always, the remote device stores the link key. For subsequent connections, the devices compare link keys. If they are correct, you do not need to re-enter the pin code. The PIN code can be set as desired using the command console.

To remove the stored link key on the remote device, you typically "unpair" or remove the device from the Bluetooth manager. The pairing information can be reset by executing the reset Paired Device List (PDL) command on the command console. You can change the pin code to remove the link key on the Bluetooth adapter, forcing a new pin code exchange to occur upon subsequent connection attempts.

Figure 1-3 shows examples of Bluetooth devices discovered list and pair/connect interfaces.

FIGURE 1-3: PAIRING/CONNECTING WITH THE BLUETOOTH® MODULE



1.4.3 Connecting

After pairing with the device is successful, the module can now connect to the device. Click connect using the pair/connect interface to establish a connection.

Once connected, the module enables the Advanced Audio Distribution Profile (A2DP) for audio playback and the Audio Video Remote Control Profile (AVRCP) for player control. Additionally, a Serial Port Profile (SPP) link can be established by opening the virtual COM port available on the RN52 module from a PC over its Bluetooth device.

When you connect to a PC for using SPP, after the Bluetooth device manager completes pairing/connecting, it issues a message that the Bluetooth device is installed on COM port X, where the specific COM port is unique to your computer. After connecting successfully, the data can be transmitted and received on the Bluetooth COM port link in Data mode. In some cases, the Bluetooth device manager creates two COM ports. In this situation, use the incoming port to wait for the module to initiate a connection. Open the outgoing port to establish a connection to the module.

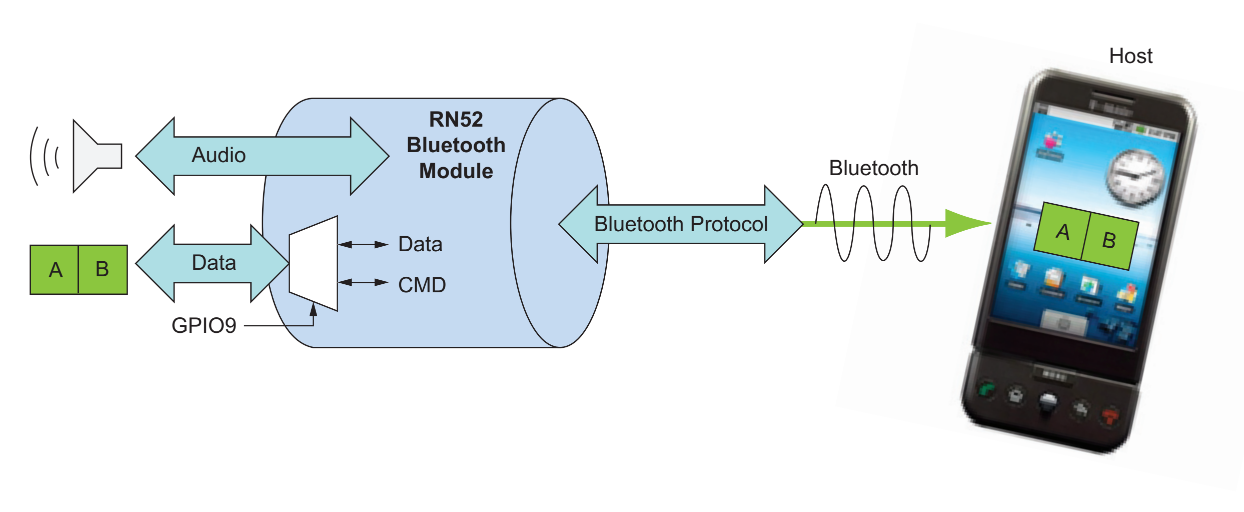
Note: Only one Bluetooth master can connect to an RN52 Bluetooth slave device at a time.

1.5 COMMAND MODE VS. DATA MODE

The RN52 UART has two modes: Data mode (default) and Command mode. While in Data mode and connected over Bluetooth (SPP or iAP), the module is essentially a data pipe. When the module receives data over the wireless Bluetooth connection, it strips the Bluetooth headers and trailers and passes the user data to the UART. When data is written to the UART, the module constructs the Bluetooth packet protocol and sends it out over the Bluetooth connection. Thus, the entire process of sending/receiving data to the host is transparent to the end microprocessor. See Figure 1-4.

Note: While in Command mode, the module can be configured for specific features using a serial terminal emulator over UART. The terminal emulator behaves as a command console in Command mode.

FIGURE 1-4: DATA AND COMMAND MODES



> **Note:** You can only configure the Bluetooth audio module locally using your computer's serial port using the RN52 Command mode. You cannot configure the module remotely over the Bluetooth link.

1.5.1 Default Configuration and Serial Port Settings

Table 1-2 shows the default configuration for the Bluetooth module.

TABLE 1-2: DEFAULT CONFIGURATION AND SERIAL PORT SETTINGS

Option	Setting
Bluetooth® mode	Slave
Bluetooth pin code (for legacy pairing mode, can be changed)	1234
Baud rate	115,200 Kbps
Bits	8
Parity	None
Stop bits	1
Flow control	Enabled

1.5.2 Configuring the Module over the UART

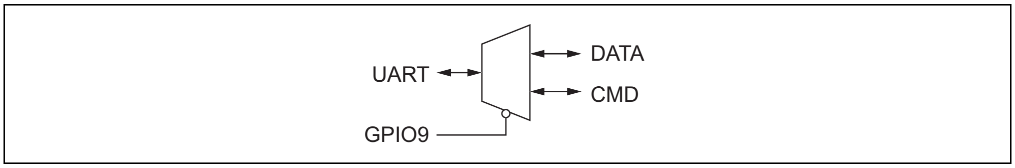
Connect the module to your computer. For example, if you are using the RN-52-EK evaluation board, connect the UART-to-USB connector on the RN52-EK to your computer using a mini-B USB cable. With the Bluetooth module connected and powered on, run a terminal emulator and open the COM port to which the cable is connected. The terminal emulator's communication settings should match the Bluetooth module's default serial port settings. Once configuration is complete, reset the device.

1.5.3 Command Mode and GPIO9

Launch a terminal emulator and specify the module's default settings (see Table 1-2).

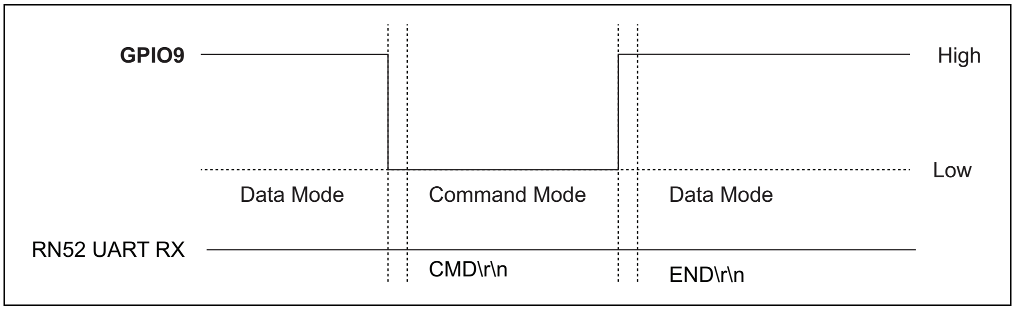
The module monitors GPIO9 as an input to determine whether data traversing the UART should be routed to the active Bluetooth SPP connection or to the command console. When you hold GPIO9 low, the UART enters Command mode. The module returns the string `CMD` to the UART console to indicate that the module is in Command mode. Figure 1-5 shows a logical diagram of the GPIO9 function.

FIGURE 1-5: GPIO9 CONTROLLING COMMAND MODE



In Command mode, the module routes all data entering the UART to the command console. In the command console, you configure the module and query its status using ASCII commands. See Figure 1-6.

FIGURE 1-6: PULLING GPIO9 LOW ROUTES UART TRAFFIC TO THE COMMAND CONSOLE

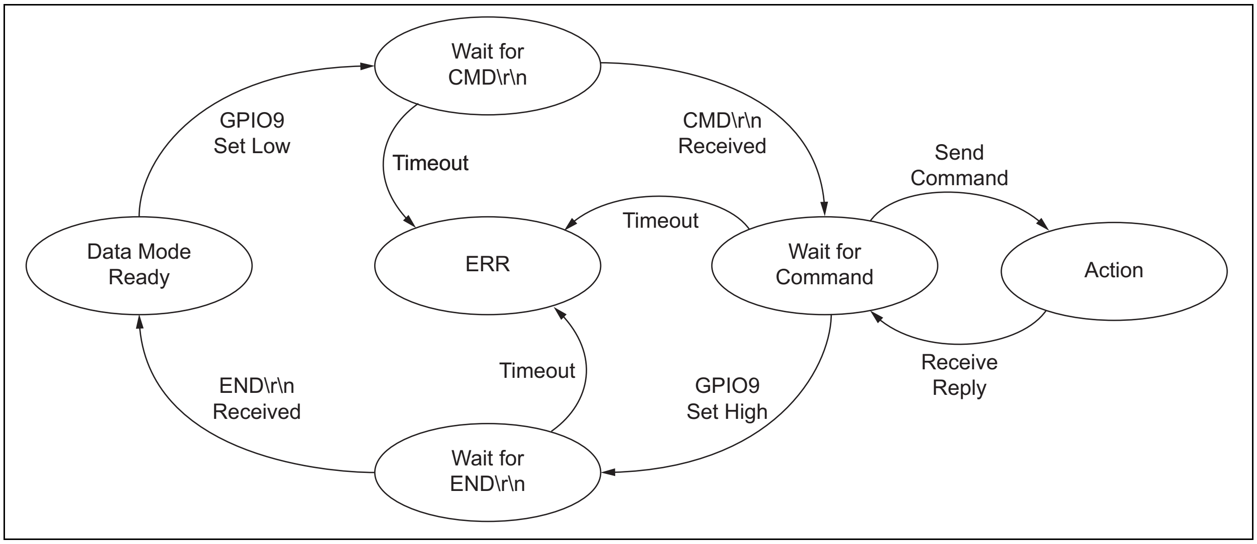


For applications in which a microcontroller controls GPIO9 and sends commands over the UART, the microcontroller should monitor the UART RX line for the following strings immediately after driving GPIO9:

- `CMD\r\n`
- `END\r\n`

Use the state diagram shown in Figure 1-7 as a guideline when designing code to monitor GPIO9 and command states.

FIGURE 1-7: STATE DIAGRAM FOR ENTERING AND EXITING COMMAND MODE



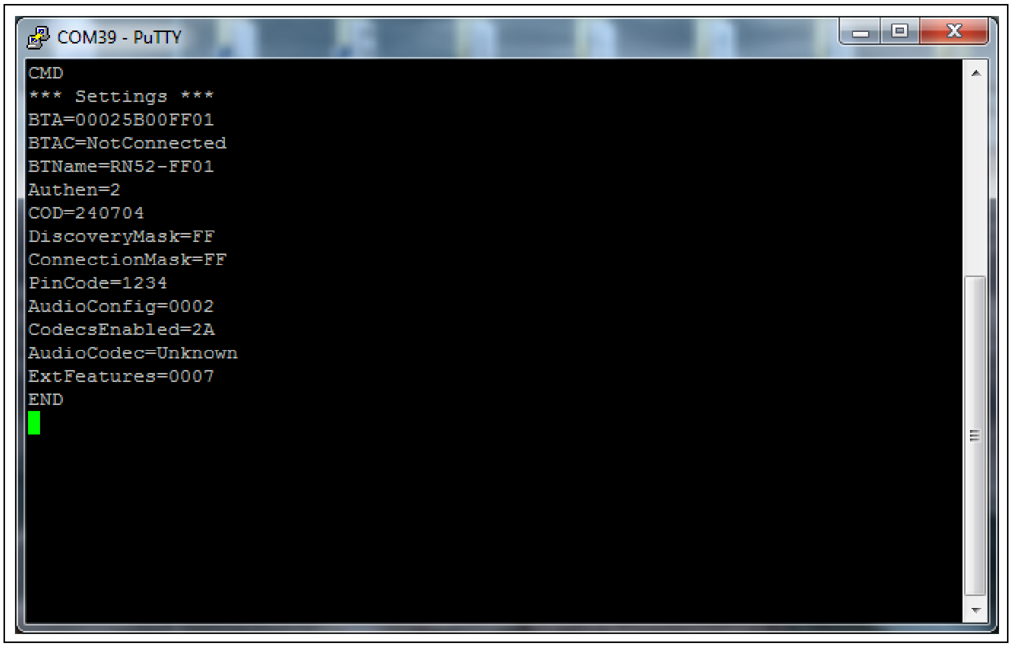
While in Command mode, the device accepts ASCII bytes as commands. When you enter a valid command, the module returns `AOK`. It returns `ERR` for an invalid command and `?` for unrecognized commands. Type `h` <cr> to see a list of commands.

A quick check to confirm that you are in Command mode is to type the `D` <cr> command after entering Command mode. This command shows the a summary of the module’s current settings, such as the Bluetooth name, device class, and serial port settings.

See Figure 1-8.

To return to Data mode, drive GPIO9 high. When leaving Command mode, the module sends END to the UART.

FIGURE 1-8: VIEW CURRENT SETTINGS



```
CMD
*** Settings ***
BTA=00025B00FF01
BTAC=NotConnected
BTName=RN52-FF01
Authen=2
COD=240704
DiscoveryMask=FF
ConnectionMask=FF
PinCode=1234
AudioConfig=0002
CodecsEnabled=2A
AudioCodec=Unknown
ExtFeatures=0007
END
```

1.6 GPIO PINS

The Bluetooth audio module has 11 GPIO pins. Several of these pins are reserved for specific functions during bootup and runtime. As a demonstration, the RN-52-EK board uses various GPIO pins to control the board's audio playback. Table 1-3 describes the RN-52-EK GPIO pin demo function.

TABLE 1-3: RN52 PIN ASSIGNMENTS AND FUNCTIONS

GPIO Pin	RN-52-EK Demo Function	Use	Direction	Default
GPIO2	Toggles from high to low for 100 ms to indicate that the module's state has changed. A microcontroller can enter Command mode and poll the state register using the Q action command. Alternately, the pin can also be configured to remain latched low after it gets asserted low for each new event until the "Q" command unlatches it back to high. Reserved. Not available for reconfiguring as a general purpose IO pin.	Runtime, Reserved	Output	High
GPIO3	Driving the pin high at boot time will put the module in DFU MODE. The device will enter DFU mode in three seconds. The PIO3 should be asserted high only before the device enters DFU mode and not after. (For USB DFU, the USB device powers VBUS. PIO3 requires 47K to ground and 22K to USB VBUS if USB VBUS is supplying power to main board.) During runtime, driving the pin high will reset and reboot the device. Reserved. Not available for reconfiguring as a general purpose IO pin. **Note:** During runtime, the device will reset within 500 ms after the IO is driven high. A reset pulse of 100 ms is recommended. If the IO continues to be driven high after reset, the device will naturally enter DFU mode. The device will enter DFU mode in three seconds. The PIO3 should be asserted high only before the device enters DFU mode and not after.	Bootup, Runtime, Reserved	Input	Low
GPIO4	Driving this pin high during bootup will put the module in Factory Reset mode. To reset the module to the factory defaults, this pin should be high on power-up and then toggle low, high, low, high with a 1 second wait between the transitions. During runtime, the module will enter voice command mode if this pin is driven low. Reserved. Not available for reconfiguring as a general purpose IO pin.	Bootup, Runtime, Reserved	Input	Low
GPIO5	The RN52-EK uses this pin for the volume up button. This pin is active-low.	Runtime, Configuration	Input	High
GPIO6	Available for input or output configuration at runtime.	Runtime, Configuration	I/O	High
GPIO7	Driving this pin low sets the UART baud rate to 9,600 bps. By default the pin is high and the UART has a baud rate of 115200 bps.	Bootup, Configuration	Input	High
GPIO9	Driving this pin low puts the module's UART into Command mode. Driving this pin high puts the module's UART into Data mode. Reserved. Not available for use at runtime.	Runtime, Reserved	Input	High
GPIO10	The RN52-EK uses this pin for the volume down button. This pin is active-low.	Runtime, Configuration	Input	High
GPIO11	The RN52-EK uses this pin for the previous track button. This pin is active-low.	Runtime, Configuration	Input	High
GPIO12	The RN52-EK uses this pin for the previous track button. This pin is active-low.	Runtime, Configuration	Input	High
GPIO13	The RN52-EK uses this pin for the previous track button. This pin is active-low.	Runtime, Configuration	Input	High

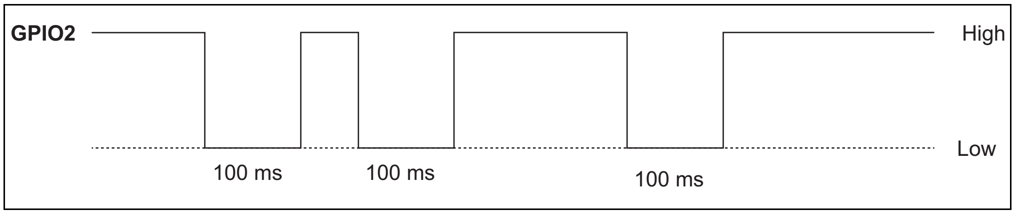
1.6.1 Using GPIO2 to Monitor the Event or Status Register

The module contains an event or status register, which you can read to determine status changes. Status changes include changes to profile connections or voice call connections. You access the register with the `Q` command. When you issue this command, the module responds with an encoded 2-byte stream of ASCII hex data terminated by the `\r\n` characters. See **Section 2.4.26 "Q"** for more information on using the `Q` command.

The module drives GPIO2 as an output to notify an external microcontroller of an event or status change. The microcontroller should enter Command mode and issue the `Q` command and parse the response to retrieve the current status.

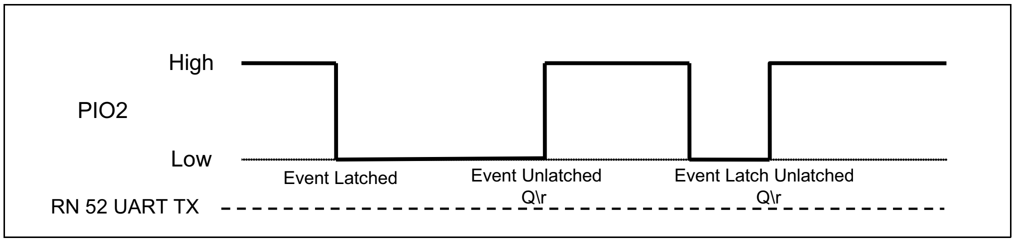
The module holds GPIO2 low for 100 ms to indicate a change or event has occurred. If a new event occurs, the register's data is overwritten with the new event information. Some bits are cleared when you read the data. See **Section 2.4.26 "Q"** for details. Figure 1-9 illustrates three event notifications.

FIGURE 1-9: GPIO2 EVENT NOTIFICATION



Starting with firmware 1.16 and later, a new option was added to use a latched event notifications. PI02 can also be configured to be held LOW for each new event and to remain latched LOW until the "`Q`" command unlatches PIO2 back to HIGH. If this notification scheme is not desired, then the module can be reconfigured to 100 ms low pulse notification on an event.

FIGURE 1-10: GPIO2 LATCHED EVENT NOTIFICATION



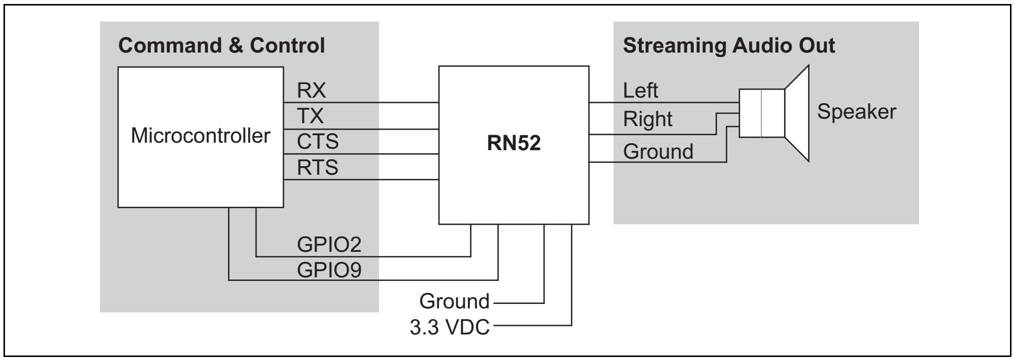
1.7 CONNECTING WITH A MICROCONTROLLER

Figure 1-11 shows a simple schematic in which a microcontroller controls the RN52 module. This schematic shows the bare minimum configuration required for the microcontroller to perform the following functions:

- Configure and control the module
- Get module's status information (see **Section 2.4.26 "Q"** for more details)
- Stream audio over the Bluetooth link

For a more complex example of how to control the RN52, refer to the RN52 Evaluation Kit schematic in the "*RN52 Evaluation Kit User's Guide*" available online at Microchip (http://www.microchip.com) web site.

FIGURE 1-11: CONNECTING THE RN52 TO A MICROCONTROLLER

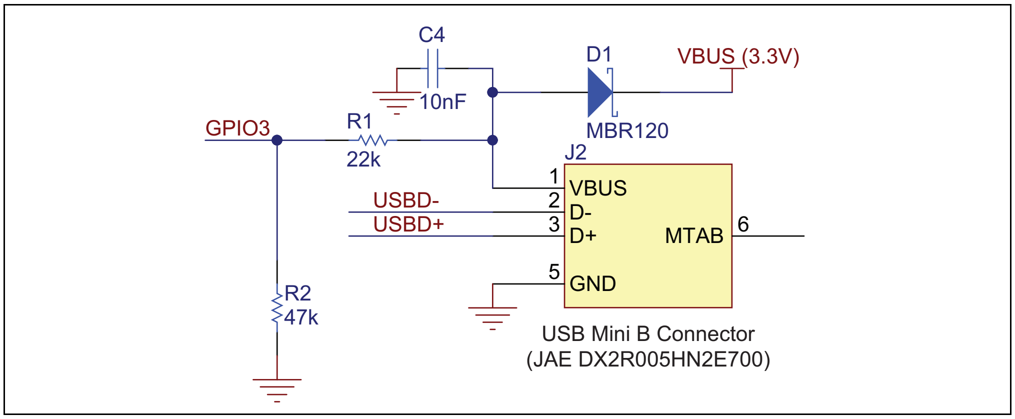


1.8 DEVICE FIRMWARE UPGRADE FOR FIRMWARE VERSIONS 1.10 AND LOWER

The module supports the Device Firmware Update (DFU) mode in which you use the module's USB interface to update the firmware. Implementing the DFU feature is recommended highly because firmware updates offer new features and enhance the module's functionality. Follow the reference design shown in Figure 1-12 to support this mode.

> **Note:** A 47 KΩ pull-down resistor (R2 in Figure 1-12) is required on GPIO3 even if you do not use the USB for DFU.

FIGURE 1-12: USB DFU PORT AND GPIO3 SCHEMATIC

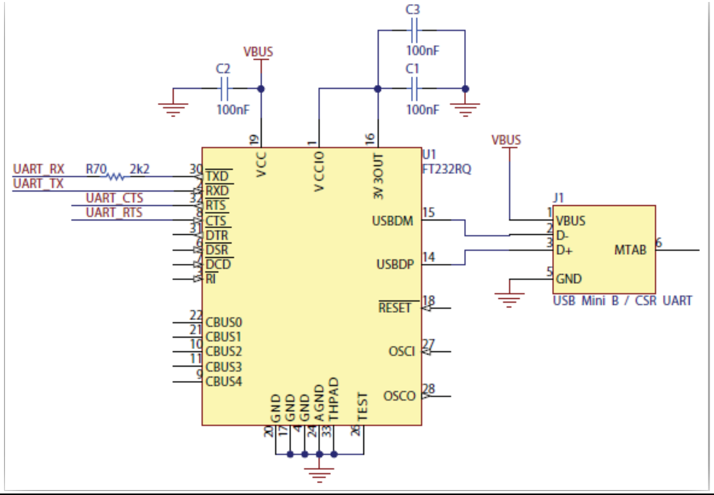


When you connect an external USB host into the DFU port and power `VBUS`, the module enters DFU mode when you reboot or power it. If your board is not powered by `VBUS`, you do not need to include the C4 and D1 to `VBUS` inputs on the voltage regulator.

1.9 DEVICE FIRMWARE UPGRADE FOR FIRMWARE VERSIONS 1.16 AND HIGHER

Device Firmware Update (DFU) uses the UART port to update the firmware in the RN52. Implementing the DFU feature is highly encouraged as firmware updates offer new features which enhance the functions of the RN52 module. Follow the reference design in Figure 1-13. When an external USB host is plugged into UART-USB bridge DFU port, the bridge enumerates as a virtual COM port. Then, at the command window, execute the ‘$’ command and the RN52 enters DFU mode. Use the DFU wizard to then perform DFU on the RN52 module.

FIGURE 1-13: UART DFU PORT USING UART-USB BRIDGE



1.10 STATUS LEDS

The module can drive status LEDs that give you a visual confirmation that the module or board is powered up and operating. Table 1-4 describes the status LEDs.

TABLE 1-4: STATUS LED FUNCTIONS

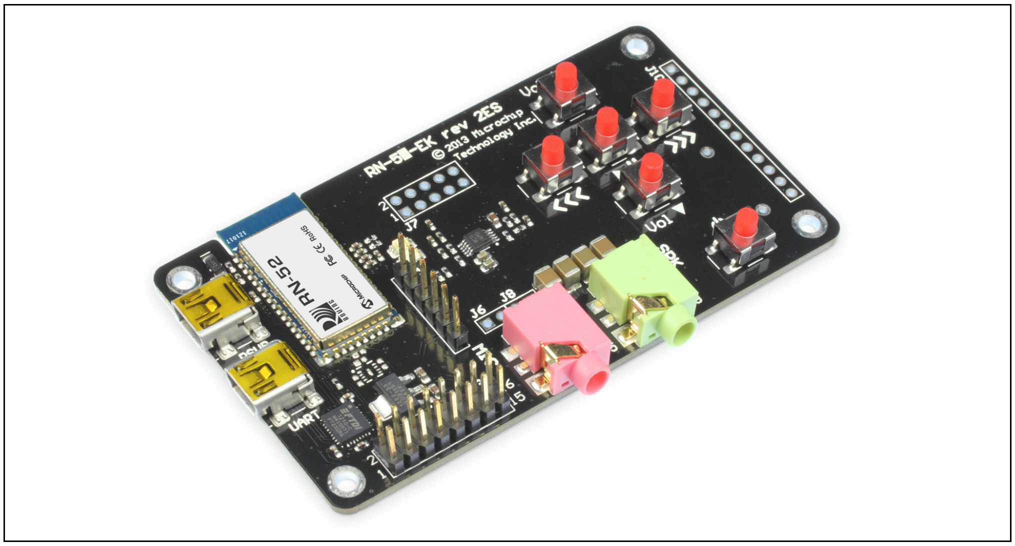
Blue LED	Red LED	Description
Flashing	Flashing	The RN52 module is discoverable
Off	Flashing	The module is connected
Flashing	Off	The module is connectable

1.11 EVALUATION BOARDS AND REFERENCE DESIGNS

Microchip’s provides a variety of evaluation kits and reference designs for evaluation and prototyping. The RN-52-EK evaluation kit is a prototyping platform for the RN52 module. The board contains buttons to control audio playback (volume up/down, next/previous track, pause/play) and connections for plugging in external speakers.

The board has the flexibility to connect directly to PCs via a standard USB interface or to embedded processors through the TTL UART interface. The status LEDs, switches, and signal headers enable rapid prototyping and integration into existing systems. Figure 1-14 shows the board.

FIGURE 1-14: RN-52-EK EVALUATION KIT



For more information on available evaluation boards and reference designs, refer to the Microchip (http://www.microchip.com) web site.

NOTES:

Chapter 2. Command Reference

Microchip's Bluetooth modules support a variety of commands for configuration. This section describes these commands in detail and provides examples.

NOTICE TO CUSTOMERS

The commands and applications described in this document apply to Microchip's Bluetooth *audio modules*, e.g., the RN52. They do not apply to Microchip's Bluetooth data modules such as the RN41 or RN42. For data module configuration information, refer to the *Bluetooth Data Module Command Reference and Advanced Information User's Guide*.

2.1 COMMAND SYNTAX

To issue commands to the module, you send a keyword followed by optional parameters via the UART.

- All commands are one or two characters and can be upper or lower case
- Delimit command arguments with a comma
- Commands use decimal input, except where noted
- Text data, such as the Bluetooth name and pin code, is case sensitive

There are three general command categories, as shown in Table 2-1.

TABLE 2-1: COMMAND TYPES

Command Type	Description
Set commands	Store information to Flash memory. Changes take effect after a power cycle or reboot
Get commands	Retrieve and display the stored information
Action commands	Perform actions such as controlling the audio playback, performing inquiries, connecting, etc.

Set commands only take effect AFTER reboot, except where noted. Get and action commands take effect immediately.

Each command terminates with the carriage return (\r).

2.2 SET COMMANDS

Set commands specify configuration settings and take effect after power cycling or rebooting. Commands are not case sensitive. All commands respond with `AOK`,<*optional message*>\r\n for success or `ERR`,<*optional message*>\r\n if the command fails.

2.2.1 S|,<*hex16*>

This command sets the routing for the A2DP audio output, where <*value*> is a value shown in Table 2-2. HSP/HFP voice calls are routed through analog speaker and microphone. Sets the route for audio output (AU), the audio bit resolution (W) and sample rate (R). Values are: "|,<AUWR>"

TABLE 2-2: A2DP AUDIO ROUTING VALUES

AU Value	Description
00	Analog output (default)
01	Set the output for I^{2}S. See Table 2-3 for the I^{2}S settings the RN52 interface supports
02	Set the output for S/PDIF
03	Intercom DAC mode

TABLE 2-3: RN52 I^{2}S SETTINGS

Parameter	Value
Configuration	Master mode
Supported sample rates	Variable
Sample width	24 bits
Synchronous data	Data is right channel with word select (WS) high. The SD data MSB occurs in the second SCLK period
Justification	Left justified

WR: Sample Rate and Sample Width

W – Bits Per Sample

0-24
1-32

R – Sample Rate (Samples Per Second)

0-8K
1-32K
2-44K1
3-48K

Default: 0002

Example: `S|,0102` // Set the audio output to I2S

2.2.2 S-,<*string*>

This command sets the module’s normalized name where <*string*> is a prefix of up to 15 alphanumeric characters. The module’s name is set to <*string*>-*XXXX*, where *XXXX* is the last four digits of the module’s MAC address.

This setting is useful for situations in which you want to set up multiple modules with similar but unique identifiers.

Default: RN52

Example: `S-,MCHIP` // Set module’s name to MCHIP-*XXXX*

2.2.3 S^,<*dec*>

The command sets automatic shutdown on idle time in seconds.

0 – Feature disable, on forever

1-600 seconds of idle time (not-connected)

Default: 0

Example: `S^,30`

2.2.4 S%,*<16-bit hex value>*

This command sets the extended features. The bit mapping for the extended features is:

Bit 0 – Enable AVRCP buttons for EK
Bit 1 – Enable reconnect on power-on
Bit 2 – Discoverable on start up
Bit 3 – Codec indicators PIO7 (AAC) and PIO6 (aptX)
Bit 4 – Reboot after disconnect
Bit 5 – Mute volume up/down tones
Bit 6 – Enable voice command button on PIO4
Bit 7 – Disable system tones
Bit 8 – Power off after pairing timeout
Bit 9 – Reset after power off
Bit 10 – Enable list reconnect after panic
Bit 11 – Enable latch event indicator PIO2
Bit 12 – Enable track change event
Bit 13 – Enable tones playback at fixed volume
Bit 14 – Enable auto-accept passkey in Keyboard I/O Authentication mode

Default: 0

Example: `S%,0087`

2.2.5 SA,*<dec>*

The set authentication command forces authentication when a remote device attempts to connect, where <*value*> is a decimal value shown in Table 2-4. Regardless of this setting, if a remote device forces authentication, this device responds with the stored pin code. Once a remote device has exchanged pin codes with this device, a link key is stored for future use. The device stores up to 8 keys in Flash memory in a first in, first out fashion.

TABLE 2-4: SET AUTHENTICATION VALUES

Value	Description
0	Open. Authentication is not required. The device accepts pin code mode
1	SSP Keyboard I/O mode is an association model of Secure Simple Pairing (SSP) (default). This mode is for scenarios where both devices are capable of displaying a passkey and are capable of providing an accept/reject response on a keyboard. If this option is set, the remote host receives a prompt; reply yes to pair. The host posts a message asking for confirmation; the module can respond yes or no. Using console command, the module can be configured to automatically respond yes, or can be configured for the user to issue a yes or no confirmation.
2	SSP "just works" mode is an association model of SSP. This mode is for end applications where at least one of the devices does not have a display capable of displaying the passkey information or keyboard capability to provide a response.
4	Pin Code mode is part of legacy pairing. Forces pin code mode, which requires the host device to enter a pin code that matches the stored pin code

> **Note:** Modes 0 and 4 are legacy modes that do not support Secure Simple Pairing (SSP) (Bluetooth version 2.0).

Default: 1 // Keyboard mode

Example: `SA,4` // Set to pin code mode

2.2.6 SC,<*24-bit hex value*>

This command sets the service class field in the Class of Device (COD), where <*24-bit hex value*> represents the COD. Bluetooth master devices use the COD when scanning for available devices to determine whether a device in a given class offers the type of service that they want.

The service class consists of the most significant 11 bits in the COD. This command sets the MSW to create the 24-bit device class number. The inquiring device interprets the service class to determine the service. A complete listing of available Bluetooth service classes is available on the Bluetooth SIG web site: https://www.bluetooth.org/apps/content

The default COD, 240704, represents the following device:

- Service class: rendering, audio
- Major device class: wearable
- Minor device class: wrist watch

Default: 240704

Example:
```
SC,240710
```
// Service class: rendering, audio
// Major device class: wearable
// Minor device class: helmet

2.2.7 SD,<*8-bit hex value*>

This command sets the discovery mask, where <*8-bit hex value*> represents the profiles enabled for discovery. The Bluetooth profiles are represented by an 8-bit hex value as shown in Table 2-5.

TABLE 2-5: BLUETOOTH PROFILE BITMASK VALUES

Bit Position	Value	Profile
0	01	iAP
1	02	SPP
2	04	A2DP
3	08	HFP

Default: FF

Example:
```
SD,01
```
// Set the discovery profile to iAP

2.2.8 SF,1

This command sets all module parameters to the factory defaults. The changes do not take effect until you reboot the module.

> **Note:** When performing a factory reset using a microcontroller, send the `SF,1` command, wait for 20 ms, send the `R,1` command, and wait for the module to return `reboot`. Then, drive GPIO9 high to exit Command mode.

Example:
```
SF,1
R,1
```
// Invoke factory defaults
// Reboot

2.2.9 SK,<*8-bit hex value*>

This command sets the connection mask where <*8-bit hex value*> represents the profiles enabled for connection. The Bluetooth profiles are represented by an 8-bit hex value as shown in Table 2-6.

TABLE 2-6: BLUETOOTH PROFILE BITMASK VALUES

Bit Position	Value	Profile
0	01	iAP
1	02	SPP
2	04	A2DP
3	08	HFP

Default: FF

Example: `SK,08` // Set the connection profile to HFP

2.2.10 SM,<*32-bit hex value*>

The command sets microphone and line-in gain levels. The speaker level can be set from 00 through 1F. Format of the command parameter is <resresMBMA>.

MA - gain of microphone A
MB - gain of microphone B

Default: 00000909

Example: `SM,00000A0A`

2.2.11 SN,<*string*>

This command sets the device name, where <*string*> is up to 20 alphanumeric characters.

Default: RN52-*XXXX*, where *XXXX* is the last 4 digits of the module’s MAC address. When you set the name, the -*XXXX* is not appended to <*string*>.

Example: `SN,MyDevice` // Set the device name to “MyDevice”

2.2.12 SP,<*string*>

This command sets the security pin code, where <*string*> is up to 20 alphanumeric characters. Each time the device pairs successfully, it saves the Bluetooth address. The device can store up to eight addresses on a first in, first out basis. Using this command also erases all stored pairings. You can use the same value that is already set.

You cannot erase the pin code, however, you can overwrite the default pin code.

Default: 1234

Example: `SP,0123` // Set pin code to 0123

2.2.13 SS,<*8-bit hex value*>

The command sets speaker gain levels. The speaker level can be set from 00 through 0F.

Default: 0A

Example: `SS,0A`

2.2.14 ST,*<8-bit hex value>*

The command sets tone gain levels. The tone level can be set from 00 through 1F.

Default: 16

Example: `ST,10`

2.2.15 STA,*<dec>*

The command sets the delay between connection tries in milliseconds.

Default: 100

Example: `STA,1000`

2.2.16 STP,*<dec>*

The command sets the pairing timeout in seconds.

Default: 0

Example: `STP,30`

2.2.17 SU,*<8-bit hex value>*

The command sets the UART baudrate.

01 – 9600 bps
02 – 19200 bps
03 – 38400 bps
04 – 57600 bps
05 – 115200 bps
06 – 230400 bps
07 – 460800 bps

Default: 05

Example: `SU,01`

2.3 GET COMMANDS

Get commands retrieve and display the device’s stored information. These commands do not have a keyword or character and do not take any parameters, except as noted. Commands are not case sensitive. All commands respond with `AOK,`<*optional message*>\r\n for success, or `ERR,`<*optional message*>\r\n if the command fails.

2.3.1 D

This command displays basic settings or the active configuration. Figure 2-1 shows an example of the output.

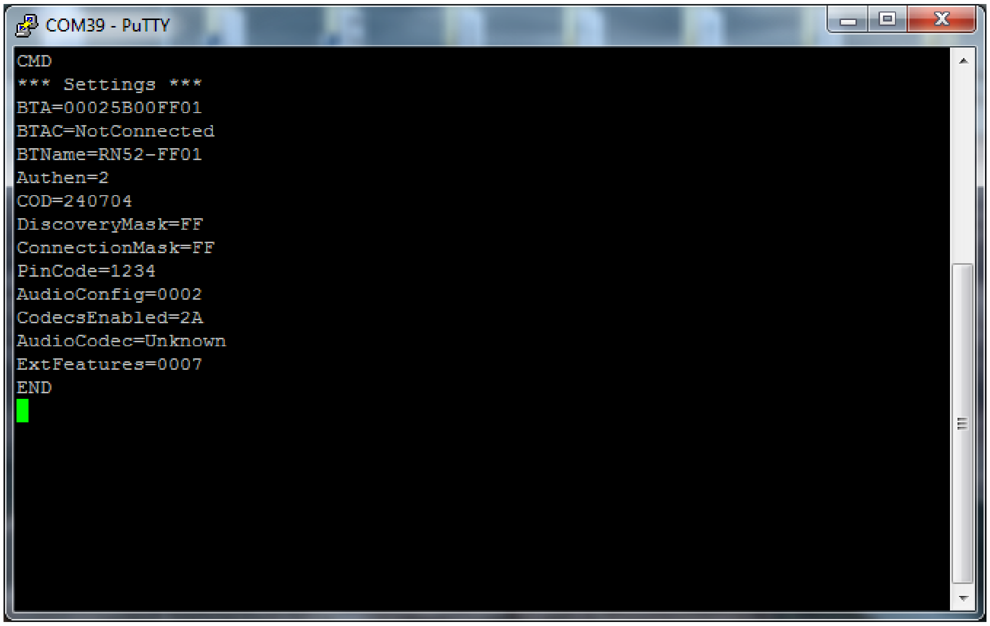
Example: `D` // Display basic settings

Active Configuration Legend:

BTA – Bluetooth address of the module
BTAC – Bluetooth address of the connected device
BTName – Friendly Bluetooth name of the module
Authen – Active authentication mode
COD – Class of device
DiscoveryMask – Active discovery mask
ConnectionMask – Active connection mask
PinCode – Assigned pin code for legacy pairing
AudioRoute – Active audio route

AudioConfig – Current audio configuration
CodecsEnabled – Active codecs
ExtFeatures – Extended features selected

FIGURE 2-1: DISPLAY COMMAND EXAMPLE OUTPUT



```
CMD
*** Settings ***
BTA=00025B00FF01
BTAC=NotConnected
BTName=RN52-FF01
Authen=2
COD=240704
DiscoveryMask=FF
ConnectionMask=FF
PinCode=1234
AudioConfig=0002
CodecsEnabled=2A
AudioCodec=Unknown
ExtFeatures=0007
END
```

2.3.2 G<*command*>

This command displays the stored settings for a set command, where <*command*> is the second character of a set command.

Example: `GA` // Display the authentication mode

`GP` // Display the pin code

2.3.3 GB

This command retrieves and displays the battery status level of the Audio Gateway (AG).

Example: `GB`

AGBatteryLevel = 80%

2.3.4 H

The help command displays a list of commands and their basic syntax.

Example: `H` // Display help

2.3.5 V

This command displays the firmware version.

Example: `V` // Show the firmware version

2.4 ACTION COMMANDS

Action commands perform actions such as audio playback, inquiries, connecting, and entering/exiting Command mode. Commands are not case sensitive.

2.4.1 +

This command toggles the local echo on and off. If you send the `+` command in Command mode, all typed characters are echoed to the output. Typing `+` a second time turns local echo off.

Default: Off

Example: `+` // Turn local echo on

2.4.2 @,<*flag*>

This command toggles whether the module is discoverable, where <*flag*> is ‘`1`’ (discoverable) or ‘`0`’ (not discoverable).

Example: `@,1` // Make the module discoverable

2.4.3 #,<*0,1*>

This command accepts/rejects pairing in keyboard IO authentication mode. <y,n> or <Y,N> also work.

Example: `#,1` // Accept pairing pass key

2.4.4 $

This command will put the RN52 module into Device Firmware Upgrade (DFU) mode.

Example: `$`

2.4.5 A,<*telephone number*>

This command initiates a voice call to a telephone, where <*telephone number*> is a decimal telephone number up to 25 digits. The module returns an error (`ERR`) if the call status is not idle.

Example: `A,14083955300` // Call 1 (408) 395-5300

2.4.6 AD

This command shows track metadata information which includes Track Title, Artist, Album, Genre, Number, Total Count, Duration (ms).

2.4.7 AR

This command will redial last dialed number.

Example: `AR` // Redial last dialed number

2.4.8 AV+

This command increases the volume.

2.4.9 AV-

This command reduces the volume.

2.4.10 AT+

This command plays the next track by sending an AVRCP volume previous track command to the host.

2.4.11 AT-

This command plays the previous track by sending an AVRCP volume previous track command to the host.

2.4.12 AP

This command pauses or starts playback by sending an AVRCP volume pause/play command to the host.

2.4.13 B

The module attempts to reconnect the Bluetooth profiles specified in the connection mask to the most recently paired device. **Section 2.2.9 “SK,<8-bit hex value>”**

Use the `Q` command to retrieve the Bluetooth profile connection status in byte 0 (bits 0-3). The module returns an error if it has not been previously connected or if the connection mask is set to 00 (meaning the module is not connectable).

2.4.14 C

This command instructs the module to accept an incoming voice call. You use the `Q` command to retrieve the call status (bits 8-10) value. The module returns an error (`ERR`) if the call status is not set to incoming calls.

Example: `C` // Accept incoming call

2.4.15 E

This command terminates an active call or rejects an incoming call. The module returns an error (`ERR`) if the call status is not an incoming call or active call.

Example: `E` // Terminate call

2.4.16 F

This command releases/terminates all held calls or sets user busy for waiting call.

Example: `F`

2.4.17 J

This command releases/terminates all active calls and accepts other held or waiting calls.

Example: `J`

2.4.18 L

This command places all active calls on hold and accepts other held or waiting calls.

Example: `L`

2.4.19 N

This command adds a held call to the active conversation.

Example: `N`

2.4.20 O

This command connects the two calls and disconnects the subscriber from both calls.

Example: `O`

2.4.21 X, <0,1>

This command transfers the active call between HF and AG.

Example: `X,1` // Transfer active call from AG to HF

2.4.22 P

This command activates the voice command application on the media device if available and allows sending voice commands over Bluetooth.

Example: `P`

2.4.23 I@

This command configures GPIO direction. A 16-bit mask is used to set GPIO direction: bit on for output, and bit off for input. If no argument is specified, the current configuration is returned.

2.4.24 I&

This command configures GPIO level and input pull-up configuration. Bit on is logical high, bit off logical low. If no argument is specified, the current levels are returned. The 16-bit <hex16> value is PIO bit map (bit4 = PIO4). When PIO is configured as an output, I&,<hex16> drives the level high for PIO specified in <hex16>. When PIO is configured as an input, I& configures the PIO to use a pull-up (bit set) or pull-down (bit clear).

2.4.25 K,<*8-bit hex value*>

This command disconnects the currently active connection, where <*8-bit hex value*> represents the profile to disconnect. The characters `KILL`<cr><lf> are echoed to the local UART once the connection is broken. The Bluetooth profiles are represented by an 8-bit hex value as shown in Table 2-7.

TABLE 2-7: BLUETOOTH PROFILE BITMASK VALUES

Bit Position	Value	Profile
0	01	iAP
1	02	SPP
2	04	A2DP
3	08	HFP

Example: `K,01` // Disconnect the iAP profile

2.4.26 M,<*flag*>

This command controls the hold/mute function for the current telephone call, where <*flag*> is '`0`' or '`1`'. If <*flag*> is '`1`', the module mutes the call; if <*flag*> is '`0`', the call is unmuted.

Example: `M,1` // Mute the call

2.4.27 Q

This command queries the current connection status in the event/status register. It returns an encoded byte stream of ASCII hex values terminated by \r\n to describe the status of the currently connected profile(s). The module drives GPIO2 low for 100 ms to notify attached equipment that the event/status register has been changed. See **Section 1.6.1 "Using GPIO2 to Monitor the Event or Status Register"** for more information.

Byte 0 of the byte stream indicates which profiles are connected as shown Table 2-8.

TABLE 2-8: BYTE 0 BIT FORMAT

Bit	Description
0	iAP wireless active connection to remote device
1	SPP active connection to remote device
2	A2DP active connection to remote device
3	HFP/HSP active connection to remote device
4	Caller ID notification event from audio gateway
5	Track change event notification
6-7	Reserved

Table 2-9 describes the bits in byte 1. Note that:

- The connection state values (bits 0-3) only change when the module’s status changes
- The event bits (4-6) in byte 0 are cleared when you issue the `Q` command

TABLE 2-9: BYTE 1 BIT FORMAT

Bit	Description
0-3	These bits indicate the connection state. See Table 2-10 for a listing of the possible states
4	HFP audio volume level change from audio gateway (phone). Use the `Y,0` command to retrieve the volume level. The module clears this bit when you read the event/status register with the `Q` command
5	HFP audio microphone level change from audio gateway (phone). Use the `Y,1` command to retrieve the volume level. The module clears this bit when you read the event/status register with the `Q` command
6-7	Reserved

Table 2-3 describes the connection states in bits 0-3 of byte 1.

TABLE 2-10: BYTE 1 CONNECTION STATES

Value	State	Description
0	Limbo	Logically off, but physically on
1	Connectable	The module is connectable, page scanning
2	Connectable and discoverable	The module is connectable and discoverable, page and inquiry scanning
3	Connected	The module is connected to an audio gateway
4	Outgoing call established	The connected audio gateway has an outgoing call in progress
5	Incoming call established	The connected audio gateway has an active call in progress and the audio is in the headset
6	Active call	The connected audio gateway has an active call in progress and the audio is in the headset
7	Test mode	The headset is in Test mode
8	Three-way call waiting	The connected audio gateway has an active call and a second call on hold
9	Three-way call on hold	The connected audio gateway has an active call and a second call on hold
10	Three-way call multi-call	The connected audio gateway has an active call and a second call on hold
11	Incoming call on hold	The connected audio gateway has an incoming call on hold
12	Active call	The connected audio gateway has an active call and the audio is in the handset
13	Audio streaming	The headset is streaming A2DP audio
14	Low battery	The system has a low battery

```
Example: 0C16\r\n                    \\ Indicates that the A2DP and HFP
                                     \\ profiles are connected, the call status is
                                     \\ active, and that an audio mute/hold
                                     \\ event was received from the phone.
```

2.4.28 R,1

This command forces a complete device reboot (similar to a power cycle). After you send this command, the module responds `reboot`.

> **Note:** When performing a factory reset using a microcontroller, send the `R,1` command and then wait for the module to return `reboot`. Then, drive GPIO9 high to exit Command mode.

```
Example: R,1                         // Reboot the device
```

2.4.29 T

The command displays the caller ID information, which includes the name and number of the calling line.

Use "`Q`" to retrieve the caller ID notification event in byte 0 (bit 4) value.

Example: `T`

2.4.30 U

This command resets and clears all the previously paired devices in the Paired Device List.

Example: `U`

Appendix A. Command Quick Reference Guide

This section provides a quick reference of the firmware commands as well as the factory defaults. Table A-1 provides an overview of the set commands, Table A-2 describes the get (or display) commands and Table A-3 describes the action commands.

TABLE A-1: SET COMMANDS

Command	Description	Factory Settings
`S|,<hex16>`	Audio output routing	0002
`S-,<string>`	Sets the normalized name	RN52
`S^,<dec>`	Automatic Shutdown on Idle	0
`S%,<hex16>`	Extended Features	0000 for RN52 modules 0007 for RN52-EK
`SA,<0,1,2,4>`	Authentication enable/disable	2
`SC,<hex24>`	Service class	240704
`SD,<hex8>`	Discovery profile mask	FF
`SF,1`	Factory defaults	N/A
`SK,<hex8>`	Connection profile mask	FF
`SM,<hex32>`	Microphone/LINEIN gain	00000909
`SN,<string>`	Device name	RN52-*XXXX*
`SS,<hex8>`	Speaker Level	0A
`ST,<hex8>`	Tone Level	16
`STA,<dec>`	Connection Delay	100
`STP,<dec>`	Pairing Timeout	0
`SU,<hex8>`	UART Baudrate	05

TABLE A-2: GET (DISPLAY) COMMANDS

Command	Description
`D`	Basic settings
`G`*<command>*	Displays setting for the set command indicated by *<command>*
`GB`	AG Battery Status
`H`	Display help
`V`	Display the firmware version

TABLE A-3: ACTION COMMANDS

Command	Description
`+`	Toggle the local echo of RX characters in Command mode
`@,`*<flag>*	Toggle whether the module is discoverable
`#,<0,1>`	Accept/reject pairing
`$`	Put the module into DFU mode
`A,`*<telephone number>*	Initial a voice call to *<telephone number>*
`AD`	Retrieve track metadata information
`AR`	Redial last dialed number
`AV+`	Increase the volume (AVRCP command)
`AV-`	Decrease the volume (AVRCP command)
`AT+`	Play the next track (AVRCP command)
`AT-`	Play the previous track (AVRCP command)
`AP`	Pause or start playback (AVRCP command)
`B`	Reconnect Bluetooth® profiles to the most recently paired and connected device
`C`	Accept an incoming voice call
`E`	Terminate an active call or reject an incoming call
`F`	Release all held calls
`I@`	Read GPIO configuration
`I@,<hex16>`	Set GPIO configuration
`I&`	Reads current GPIO levels for input
`I&,<hex16>`	Set GPIO levels for output
`J`	Accept waiting calls and release active calls
`K,`*<hex8>*	Kill the currently active connection
`L`	Accept waiting calls and hold active calls
`M,`*<flag>*	Toggle the on hold/mute function
`N`	Add held call
`O`	Connect two calls and disconnect the subscriber
`P`	Activate Voice Command
`Q`	Query the current connection status
`R,1`	Reboot
`U`	Reset Paired Device List (PDL)
`T`	Retrieves caller ID information
`X,<0,1>`	Transfer call between HF and AG

Appendix B. Firmware Revision History

The following table provides the firmware revision history.

TABLE B-1: FIRMWARE RELEASE REVISION

Version	Description	Date
1.08	First release on RN52-EK	March 2013
1.09	Added new commands: I@ action command to configure GPIO I& action command to set/get GPIO levels S%, to configure extended features	April 2013
1.10	Modified S% extended features to avoid factory default via SF,1 command. Extended features must be set with S%,<byte> command. This was done to support RN52EK configuration on production line.	April 2013
1.11	Fixed an issue with B command not reconnection iAP profile.	April 2013
1.12	Changed GPIO2 from pulse to latched/unlatch mechanism.	April 2013
1.13	Hardened K,FF and B,a	May 2013
1.16	Added support for optional decoders Added UART DFU support. HFP wideband support. Added HFP three-way calling and caller ID support. Added audio track meta data support.	May 2014

MICROCHIP

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

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

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

Detroit
Novi, MI
Tel: 248-848-4000

Houston, TX
Tel: 281-894-5983

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

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

New York, NY
Tel: 631-435-6000

San Jose, CA
Tel: 408-735-9110

Canada - Toronto
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: 852-2943-5100
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 - Chongqing
Tel: 86-23-8980-9588
Fax: 86-23-8980-9500

China - Hangzhou
Tel: 86-571-8792-8115
Fax: 86-571-8792-8116

China - Hong Kong SAR
Tel: 852-2943-5100
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: 86-755-8864-2200
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: 91-20-3019-1500

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: 886-2-2508-8600
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 - Dusseldorf
Tel: 49-2129-3766400

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

Germany - Pforzheim
Tel: 49-7231-424750

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

Italy - Venice
Tel: 39-049-7625286

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

Poland - Warsaw
Tel: 48-22-3325737

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

Sweden - Stockholm
Tel: 46-8-5090-4654

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

03/25/14