AN2160

Classic Bluetooth® Communication Using Microchip RN41/42 Module and 8-bit PIC® Microcontroller

Authors: *Pradeep Shamanna*
Raghuraj Tarikere
Microchip Technology Inc.

INTRODUCTION

Most of the embedded applications require real-time communications to support their ecosystem. Standard wired communication, such as RS232, RN422, RS485 or Ethernet are *not* easily used due to the infrastructure support required by the end application. Wi-Fi® and Bluetooth® have emerged as the standards of choice for connecting embedded applications to the cloud through a router, smartphone or tablet.

Bluetooth is known to provide easy, temporary connectivity to smartphones/tablets, and is supported by Android™ and iOS® applications. It provides a convenient cable replacement communication medium for applications involving audio streaming and data synchronization between devices. Bluetooth 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 Low Energy (BLE) technology is introduced through Bluetooth version 4.0 from Special Interest Group (SIG) and with this, there has been a considerable interest in various application possibilities in different market segments. BLE works with extremely low-power, unique features and also supports new services/profiles.

Bluetooth Classic and LE technology are quite different from one another, thus, user has to consider the technology which meets the applications requirements. However, both Classic Bluetooth and BLE have found presence with the Internet of Things (IoT) that requires ease of network connectivity by enabling physical objects or devices to connect and exchange data.

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 Classic Bluetooth Microchip RN41/42 module and the PIC18 (8-bit) microcontroller over the UART using the ASCII command interface. It essentially supports the application developers with an interface framework in using the Microchip Bluetooth Module and the MCU which are suitable for IoT and related applications.

CLASSIC BLUETOOTH COMMUNICATION

Wireless technology, like Bluetooth, has become the standard for exchanging data over short distances from fixed and mobile devices, and for building Personal Area Networks (PANs) and Body Area Networks (BANs). Bluetooth technology is initially designed for continuous data and voice streaming applications. It successfully eliminated wires in many consumers, industrial, and medical applications. Classic Bluetooth technology continues to provide a robust wireless connection between devices ranging from infotainment in cars to industrial controllers and medical sensors.

The Classic Bluetooth uses short-wavelength UHF radio waves, which are part of the globally unlicensed Industrial, Scientific and Medical (ISM) 2.4 GHz radio frequency band. Bluetooth uses frequency-hopping spread spectrum. Bluetooth operates at frequencies between 2400 MHz to 2483.5 MHz, includes guard bands of 2 MHz at the bottom and 3.5 MHz at the top. Each channel has a bandwidth of 1 MHz. The first channel starts at 2402 MHz and continues up to 2480 MHz in 1 MHz steps. Bluetooth divides transmitted data into packets, and transmits each packet on one of the 79 designated channels. It usually performs 1600 hops per second, with Adaptive Frequency-Hopping (AFH) enabled. The maximum transmit power in a band is limited to 10 mW by ISM standards.

Initially, Gaussian Frequency-Shift Keying (GFSK) modulation is the only modulation scheme adopted. Since the introduction of Bluetooth 2.0+EDR, the $\pi/4$-Differential Quadrature Phase Shift Keying (DQPSK) and 8-DPSK modulation are also used between compatible devices. Devices functioning with GFSK are operating in Basic Rate (BR) mode where an instantaneous data rate of 1 Mbps is possible, whereas the Enhanced Data Rate (EDR) is used to support the $\pi/4$-DQPSK and 8-DPSK schemes, each giving 2 and 3 Mbps, respectively.

Bluetooth protocol supports Master-Slave network architecture. One master can communicate with up to seven slaves in a Piconet. All devices share the master's clock. Packet exchange is based on the basic clock, defined by the master which ticks at 312.5 µs intervals. Two clock ticks make up a slot of 625 µs, and two slots make up a slot pair of 1250 µs. In single-slot packets, the master transmits in even slots and receives in odd slots.

The slave, conversely, receives in even slots and transmits in odd slots. Packets may be 1, 3, or 5 slots long, but in all cases the master's transmission begins in even slots and the slave's in odd slots.

For additional information related to Bluetooth and its specifications, refer to *“Classic Bluetooth Specification”* by SIG from the following website: http://www.bluetooth.org.

SERIAL PORT PROFILE (SPP)

Bluetooth profiles are additional protocol formats that are based on the Bluetooth standard to define the kind of data transmitted by the Bluetooth module. Bluetooth specifications define how the technology works while the profiles define how it is used.

The Serial Port Profile defines the specific protocol format and procedures for devices using Bluetooth usually for RS232 serial cable emulation. SPP is one of the most fundamental Bluetooth profiles to replace RS232 cables as it enables sending bursts of data between two devices. Using SPP, each connected device can send and receive data such as these devices are connected by RX and TX lines. There are no fixed Master/Slave roles in this profile. The transport layer of Bluetooth, Radio Frequency Communication (RFCOMM), is used to transport the user data, modem control signals, and configuration commands.

For the execution of the SPP profile, use of security implementation features such as authorization, authentication, and encryption is optional. Support for authentication and encryption is mandatory if the device has to take part in the security procedures requested from a peer device. The two devices are paired during the connection establishment phase that makes the connections secure. Bonding is *not* explicitly used in SPP profile, therefore support for this is optional.

MICROCHIP RN41/42 CLASSIC BLUETOOTH MODULES

The Microchip RN41/42 module is a small form factor, low-power, Class 1/Class 2 Bluetooth radio ideal for designers who want to add wireless capability to their products without spending significant time and money developing Bluetooth-specific hardware and software. The RN41/42 module is fully certified, easy to design-in and supports multiple interface protocols, making it a complete embedded Bluetooth solution.

With its high-performance capability, available options for PCB trace antenna, chip antenna or external antenna, and support for Bluetooth EDR, the RN41/42 module delivers up to 3 Mbps data rate for distances up to 100/10 meters. The surface-mounted RN41/42 module has the complete Bluetooth stack on board and is controlled through simple ASCII commands over the UART and Port Input/Output (PIO) signals interface.

A Microcontroller Unit (MCU) or host processor sends commands to configure module features, read status, and manage Bluetooth data connections. The UART TX and RX lines are required to communicate with the module and transfer data through the Bluetooth SPP connection. Connecting the hardware flow control lines, CTS and RTS, is also highly recommended for applications that transmit a continuous stream of data. The RN41/42 module can also be used in Master/Slave modes.

Note:	Bluetooth devices can be configured over the Bluetooth link or through the module’s UART using a simple ASCII command language by entering the Command mode. The Set commands configure the module while the Get commands echo the configuration.

Figure 1 and Figure 2 illustrate the RN41 and RN42 modules mounted on the RN-41-EK and RN-42-EK Development boards, respectively.

FIGURE 1: RN-41-EK DEVELOPMENT BOARD

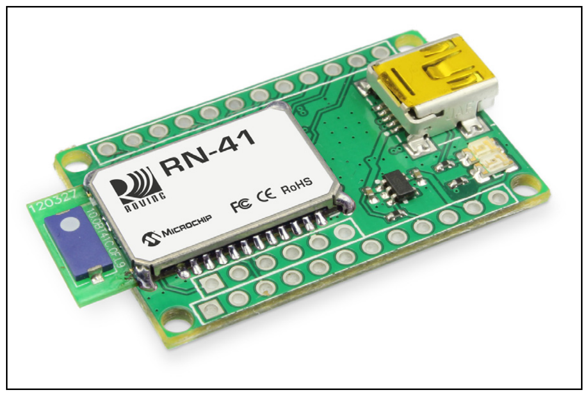
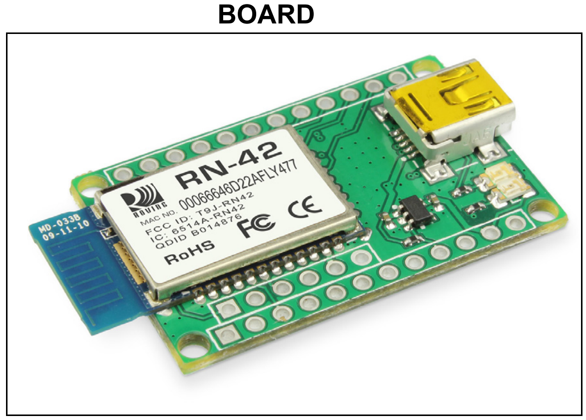


FIGURE 2: RN-42-EK DEVELOPMENT BOARD



RN41/42 MODULE AND PIC18 MCU INTERFACE FRAMEWORK

The demo application uses required ASCII commands, issued by the PIC18F87J11 microcontroller, to configure and setup the wireless BT nodes. User input is given through the switches on the PIC18 Explorer Development board. Status messages are displayed on the LCD of PIC18 Explorer Development board. After successfully establishing a Bluetooth connection between two nodes, data in the form of strings/ characters are transferred between these nodes, showcasing the SPP profile which emulates the serial RS232 type of connection.

This application note provides the users with the following functionalities:

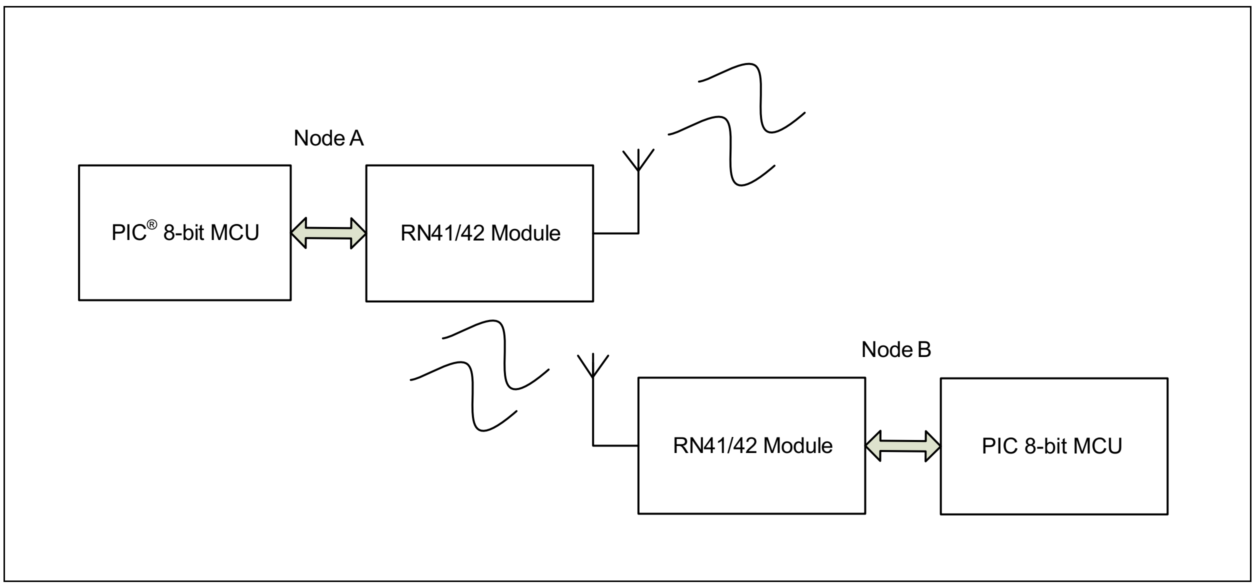
- Framework for any user application platform using the RN41/42 Bluetooth module and PIC18F series of microcontrollers
- An interface between the RN41/42 Bluetooth module and the PIC18F87J11 microcontroller
- Reference code to manage connections of the RN41/42 module through PIC® microcontroller
- Procedures or techniques for interfacing a PIC microcontroller and Classic Bluetooth module such as the RN41/42 module
- Demonstration of the Classic Bluetooth SPP for emulation of serial data connections

Figure 3 illustrates the PIC18 MCU interface with the RN41/42 module. User inputs are obtained through the switches available on the PIC18 Explorer Development board, and the status can be monitored through on-board LCDs and LEDs.

The hardware interface of the RN41/42 module with MCU makes it a Bluetooth wireless node as shown in Figure 3. In this application note, the wireless node refers to the interface between the RN-42-EK, RN42 Evaluation board, and the PIC18 Explorer Development board. The RN-41-EK board can also be used instead of the RN-42-EK board.

Note:	This application note is *not* intended to provide a complete understanding of the Bluetooth technology principles or usage of all the ASCII commands that are related to the RN41/42 module. Instead, it uses commands relevant for running the application demo code.

FIGURE 3: RN41/42 CLASSIC BLUETOOTH COMMUNICATION APPLICATION DIAGRAM



Application Demo Requirements

This section describes the hardware, software, and related utility tools required for the demo setup.

HARDWARE REQUIREMENTS

Use the following hardware for the demo application:

- Two Microchip RN-41/42-EK boards mounted with any RN41/42 modules
- Two PIC18 Explorer Development boards with PIC18F87J11 PIMs mounted
- One of the following Microchip development tools for programming and debugging: MPLAB® REAL ICE™ In-Circuit Emulator, MPLAB ICD 3 or PICkit™ 3
- Two power supplies: 9V/0.75A batteries
- 12 multi-strand (gauge>20) wires with soldered female Berg connectors on either end

> **Note 1:** Solder the J1 (12-pin) and J2 (12-pin) connector slots using conventional male Berg sticks/pins with appropriate pitch size. Refer to Table 1.
>
> **2:** The application note demo uses the RN42 module mounted on the RN-42-EK boards.

SOFTWARE/UTILITY REQUIREMENTS

This demo application intends to showcase communication between two Classic Bluetooth wireless nodes. The application demo source code related to this application note is available as MPLAB X workspace project file and is available for download from the Microchip website. The code is compiled using the Microchip XC8 compiler v1.34 and MPLAB X IDE v3.05.

> **Note:** The RN41/42 modules must have firmware version 6.30 and above for the demo code to work.

Demo source code is available for download from the *Documentation* and *Software* link section of the RN41 or RN42 product page at www.microchip.com/RN41 or www.microchip.com/RN42. Use the precompiled `RN42_MCU18_Interface.X.production.hex` file of the demo or compile the RN42_MCU18_Interface project code if required. Make sure the compilation is successful. For additional information on the source code, related files with description, and call graph, refer to **Appendix C: "Source Code"**. From MPLAB X, user can generate call graphs related to specific functions of the demo code.

HARDWARE DEMO SETUP

The RN42-based communication demo requires two wireless nodes. The demo setup consists of two PIC18 Explorer Development boards with two identical RN-42-EK boards interfaced to it as shown in Figure 3. For more information on the RN41/42 module, refer to the Microchip website: http://www.microchip.com.

> **Note:** Instead of using two identical RN-41-EK/ RN-42-EK boards, a combination of the boards can also be used.

PIC18 Explorer Development Board and RN42 Module Connections

GPIO Signal Headers 1 and 2 (J1 and J2) from RN-42-EK Development board are connected to the PIC18 Explorer Development board's J9 and J3 connectors. This connection supplies 3.3V power, two/four wire UART, Reset, GPIO7 (Baud rate control) to the RN42 module from the Microcontroller.

TABLE 1: CONNECTION DETAILS BETWEEN THE RN-42-EK BOARD AND THE PIC18 EXPLORER DEVELOPMENT BOARD

RN41/42-EK Board(2)			PIC18 Explorer Development Board		
Signal Header J1	**Signal Header J2**	**Pin No.**	**Connector J3**	**Connector J9**	**Pin No.**
—	TXD	18	RC7/RX1/DT1	—	38
—	RXD	19	RC6/TX1/CK1	—	37
GPIO7	—	2	RA3/AN3/VREF+	—	27
$\overline{\text{RESET}}$_N	—	3	RA4/PMD5/T0CKI	—	34
VDD (3.3V)	—	11	—	+3.3V	5 (3.3V)
GND	—	12	—	GND	6 (GND)
—	CTS(1)	16	—	—	—
—	RTS(1)	17	—	—	—

Note 1: Short CTS to GND (any of the common ground available on the board) and leave RTS unconnected on the Signal Header J2 on the RN41/42 EK board.

2: For details on the connectors/signal headers of the RN-42-EK board, refer to **Appendix A: "RN-42-EK Board Signal Headers"**.

Figure 4 illustrates pin to pin connections used in the application demo between the PIC18F87J11 MCU PIM residing on the PIC18 Explorer Development board and the RN42 Classic Bluetooth Module mounted on the RN-42-EK board. Figure 5 shows the RN-42-EK board is connected to the PIC18 Explorer Development board.

FIGURE 4: CLASSIC BLUETOOTH RN42 MODULE TO MCU - INTERFACE DIAGRAM

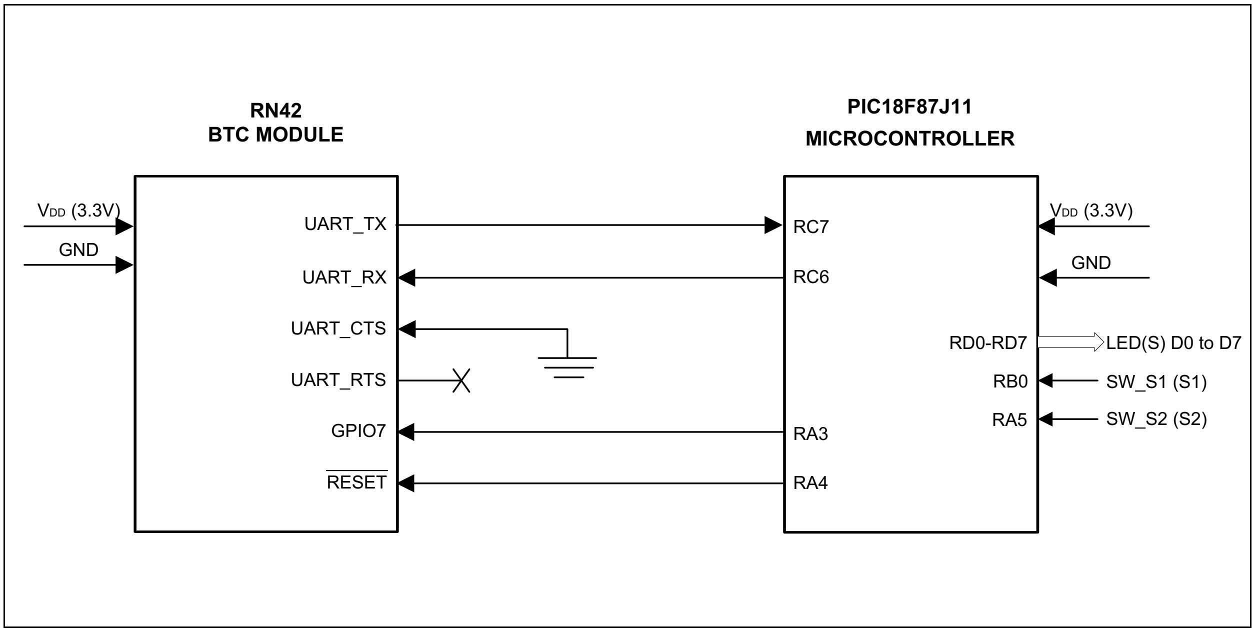
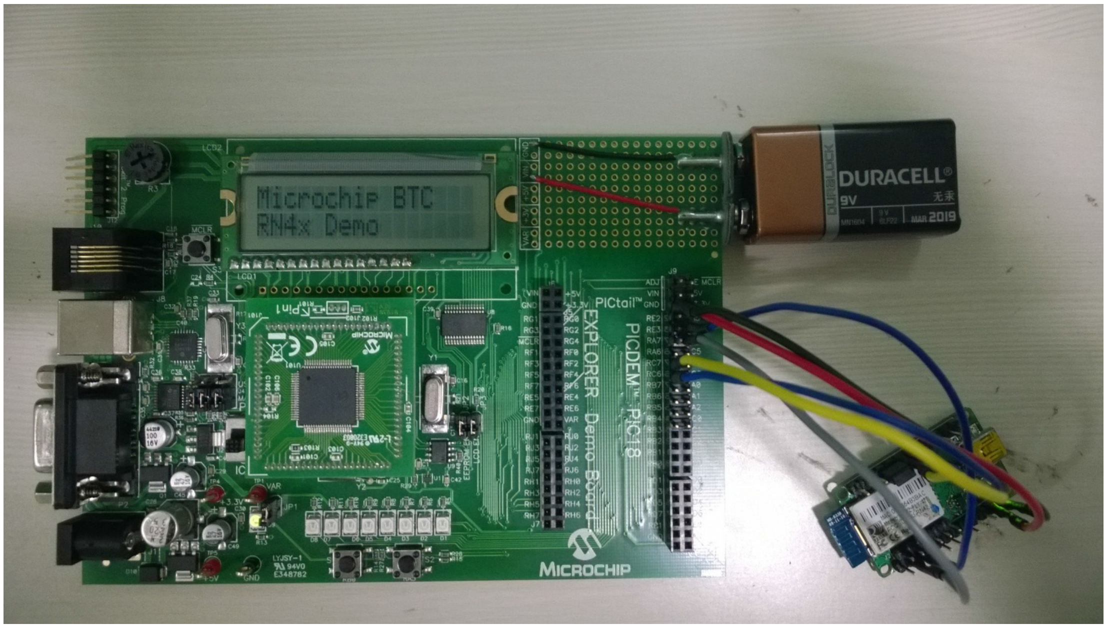


FIGURE 5: RN-42-EK BOARD CONNECTED TO PIC18 EXPLORER DEVELOPMENT BOARD



GETTING STARTED

Setting Up the Bluetooth Nodes

To setup a wireless Bluetooth node, perform the following instructions:

1. Connect the RN-42-EK Development board's GPIO Signal Headers 1 and 2 (J1 and J2) to the PIC18 Explorer Development board's J9 and J3 connectors using wires (external wiring).
2. Set the jumpers/positions on PIC18 Explorer Development board as follows:
    - JP3 to enable LCD
    - JP1 to enable LEDs
    - J13 to ensure that communication is routed through the RS-232 socket
    - J4 to ensure the Main PIC is programmed
    - Switch S4 to enable Processor In Module (PIM) (pointing towards MPLAB REAL ICE In-Circuit Emulator when ON).

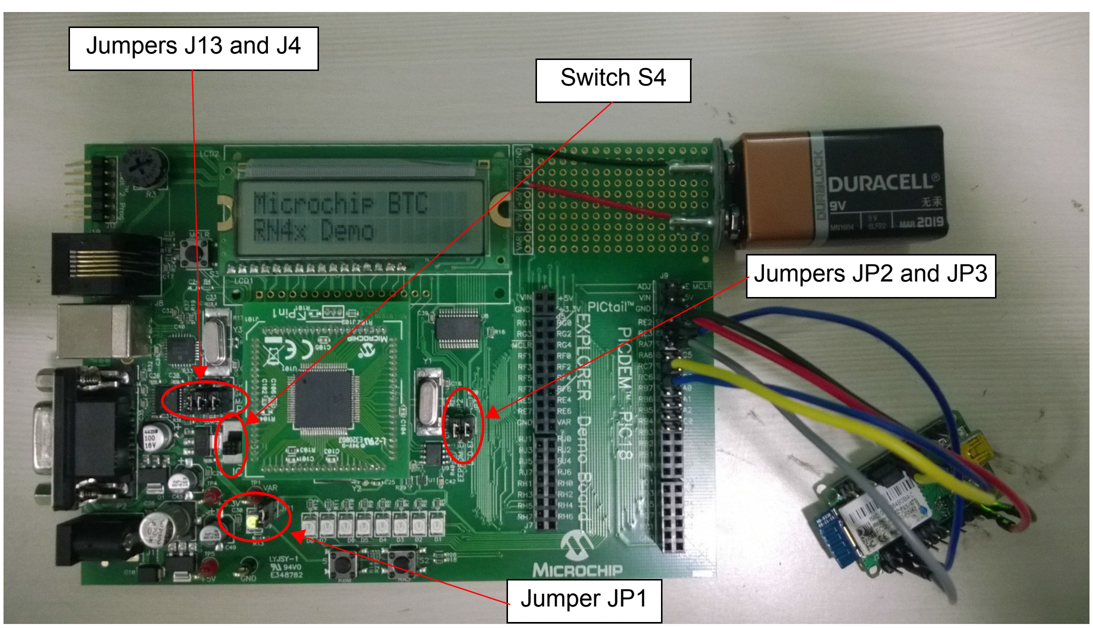
Figure 6 shows the position of jumpers and switches on the PIC18 Explorer board.

3. Switches (S1 and S2) and LCD (LCD1) on the PIC18 Explorer Development board are used for configuring and monitoring the wireless terminals.
4. Connect the programmer or debugger (MPLAB REAL ICE In-Circuit Emulator, MPLAB ICD 3 or PICkit 3) to the PIC18 Explorer Development board.
5. Plug-in the 9V power supply to the PIC18 Explorer Development board through the 9V adapter (wall power) or through 9V Battery as shown in Figure 6.
6. Open the downloaded application demo source code and compile it. Alternatively, use the precompiled `RN42_MCU18_Interface.X.production.hex` file available in the downloaded folder.
7. The generated or precompiled `RN42_MCU18_Interface.X.production.hex` file can then be programmed into the two wireless nodes, A and B, using any of the Microchip devices supporting the PIC18F87J11.
8. The boards are now ready to run the demo. There are times when the user performs a Hardware Reset to run the code, specifically in case of PIC18 Explorer Development board.

Note 1:	9V battery supply to the PIC18 board is connected only if a battery socket provision is made as shown in Figure 6.
2:	For the application demo, a 9VMN1604 (9V-6LF22, alkaline manganese dioxide) battery is used.

For additional information on programming/debugging with MPLAB ICD 3, refer to *"MPLAB® ICD 3 In-Circuit Debugger User's Guide for MPLAB X IDE"* (DS50002081), and for PIC18 Explorer Development board, refer to *"PICDEM™ PIC18 Explorer Demonstration Board User's Guide"* (DS50001721) which are available for download from the Microchip website at www.microchip.com.

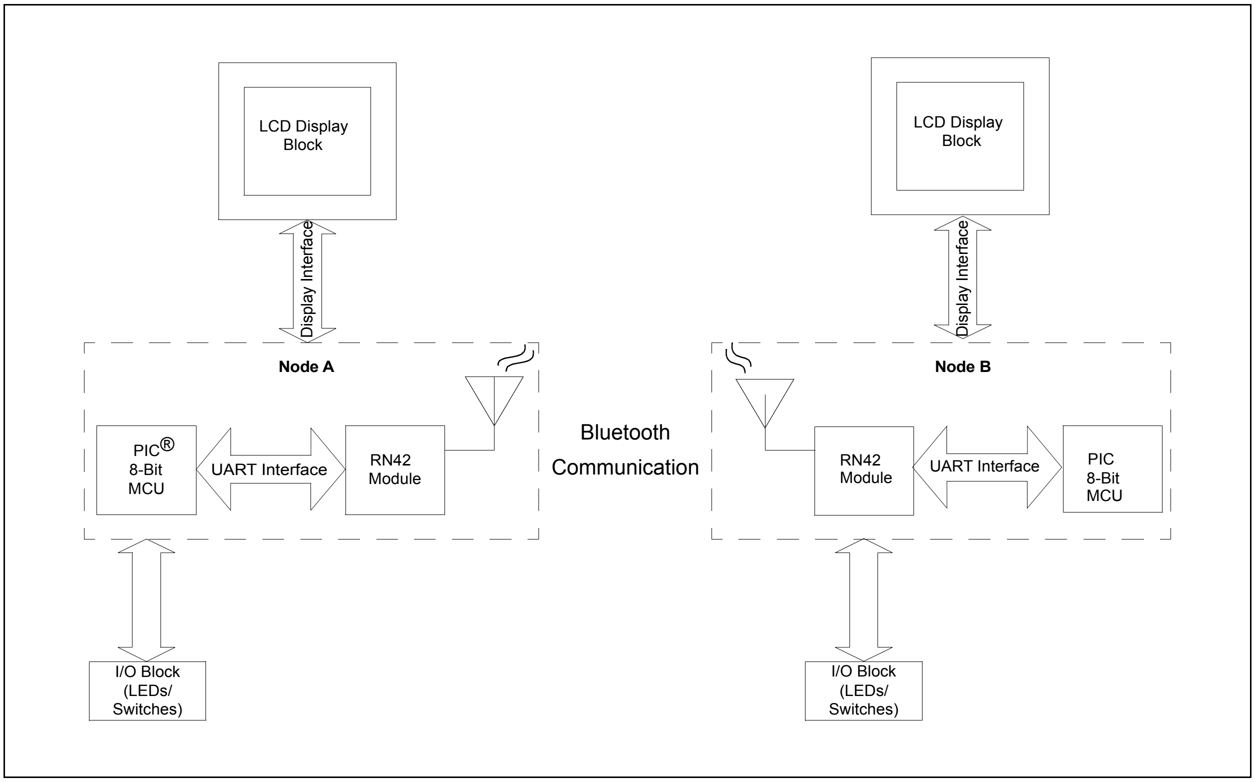
FIGURE 6: SWITCH AND JUMPER POSITIONS ON THE PIC18 EXPLORER BOARD



Application Block Diagram and Flow Chart

Figure 7 shows a Bluetooth-based wireless application demo interface.

FIGURE 7: APPLICATION BLOCK DIAGRAM WITH LCD SWITCHES INTERFACE



Running the Demo Application

Running the RN42 demo application involves the following steps:

1. Configure the two wireless nodes such that one of the wireless nodes (for example, Node A) initiates a connection and the other node (for example, Node B) waits for the connection request.
2. Connect two BT wireless nodes, Node A and Node B.
3. Send and receive data strings between two nodes over Bluetooth.

The LCD on the PIC18 Explorer board displays the sequence of events happening in the background such as initializing, scanning/inquiring/discovering of nodes, connecting and so on, and then enable the user to operate using the interactive messages. The user must operate using the hardware switches S1 and S2 to provide either a *Yes* or *No* responses as inputs to configure and control the demo.

Note:	The application demo requires one BT wireless node (Node A) to be configured to initiate/inquire a connection and the other wireless node (Node B) to wait for the connection request in Discoverable mode.

Figure 8 and Figure 9 illustrate the complete cycle of the application demo.

FIGURE 8: APPLICATION DEMO FLOW CHART

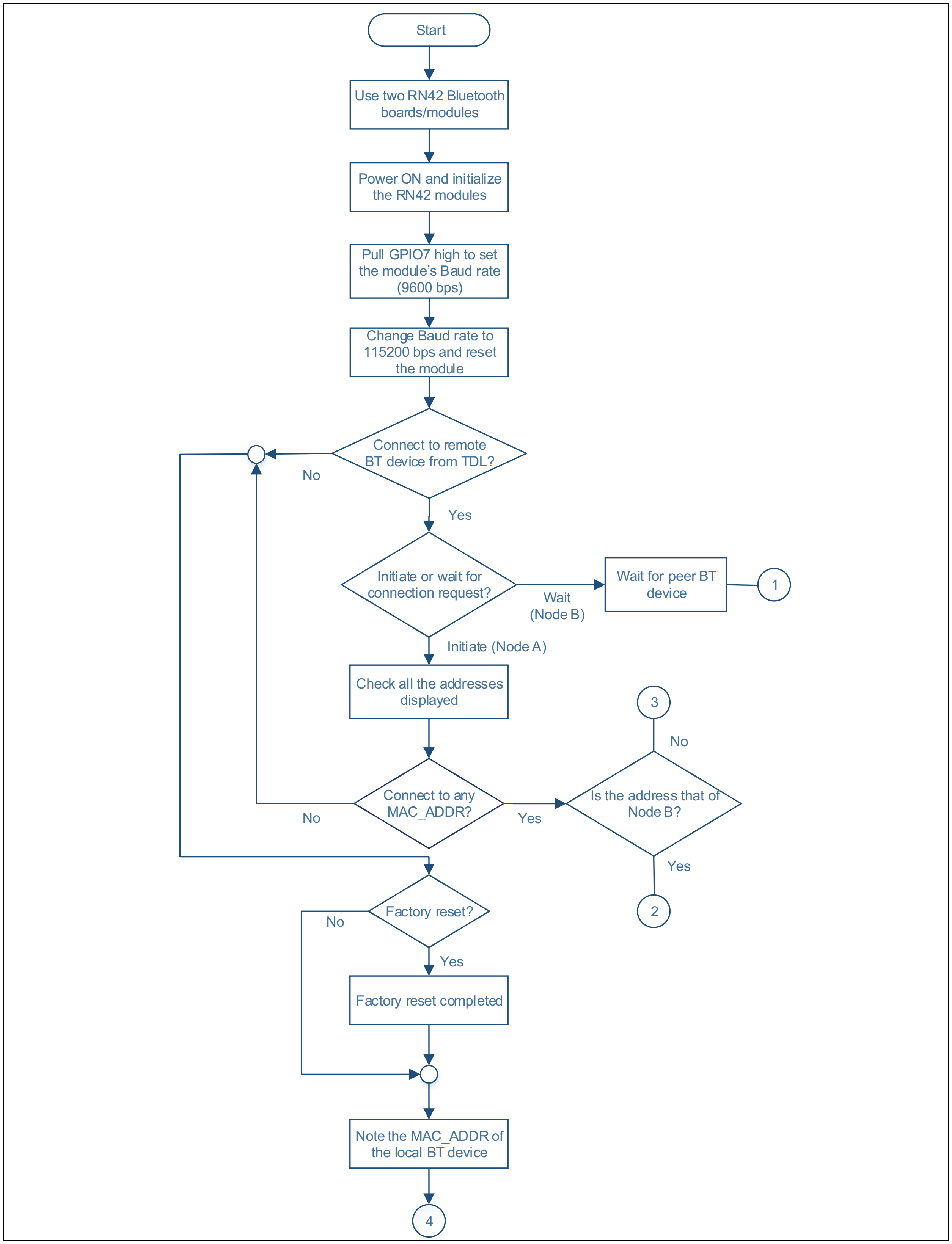
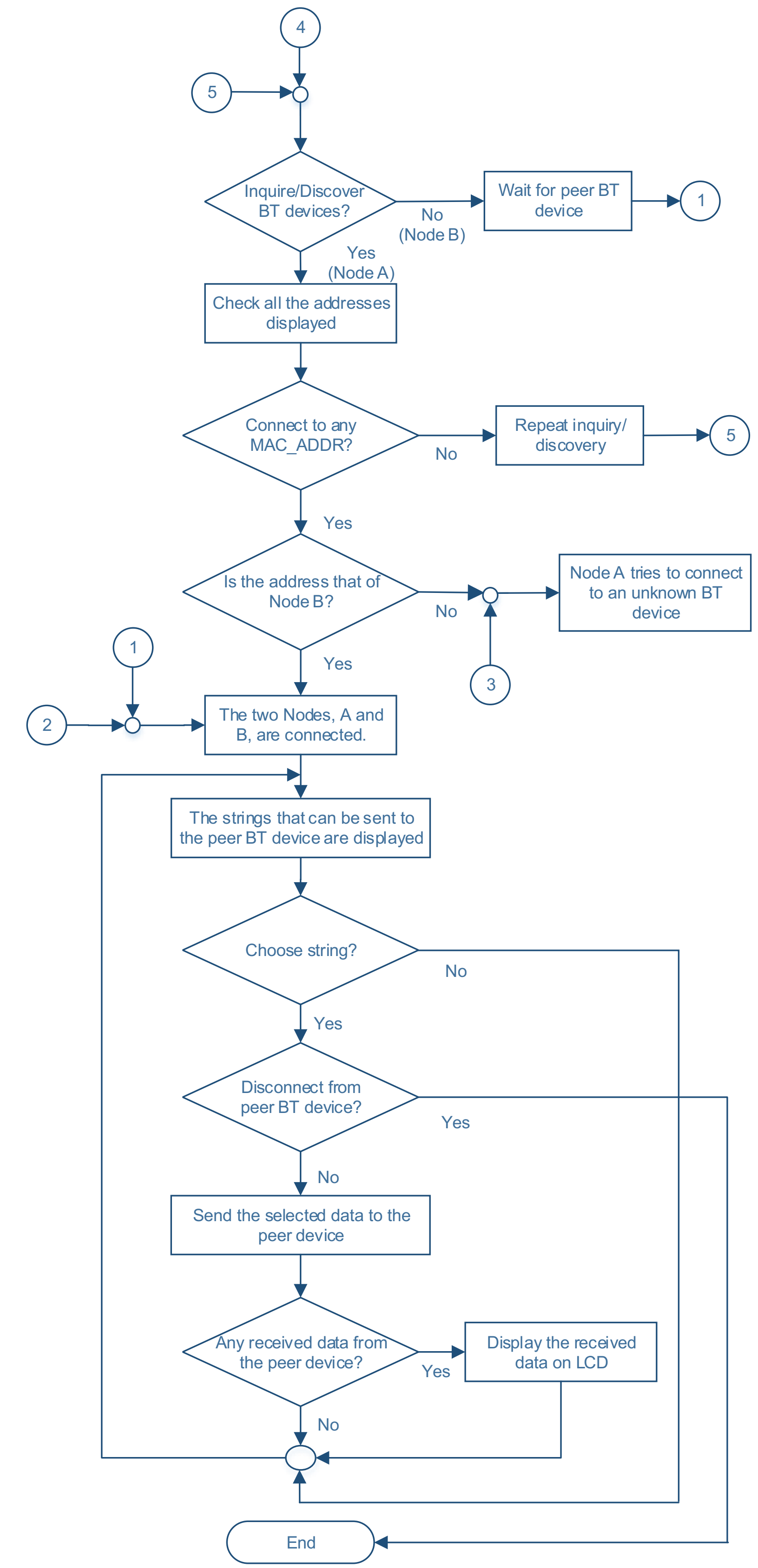


FIGURE 9: APPLICATION DEMO FLOW CHART (CONTINUED)

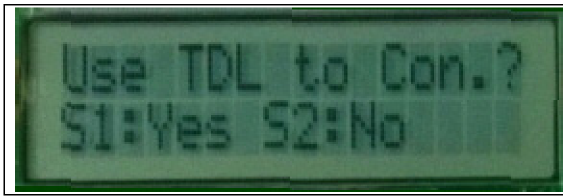


Configuring the Two Wireless Nodes

To configure the nodes, follow these steps:

1. The user can connect to any of the last Paired/Connected devices. Refer to Figure 10. If connection is through the previous connection details, then configure the modules such that one of the nodes (for example, Node A) initiates the connection and the other node (for example, Node B) waits for Node A to initiate the connection by sending the request to join. The LED (green) is still toggling, which indicates the unconnected state of the nodes. Refer to Figure 10. For a new connection, see Step 7.
2. At Node A and Node B, press switch S1 to connect to the previous node.

FIGURE 10: CONNECTING TO LAST PAIRED/CONNECTED DEVICES



3. The user can now press switch S1 to initiate connection from Node A. At Node B, also press switch S1 to wait for peer connection request. Refer to Figure 11.

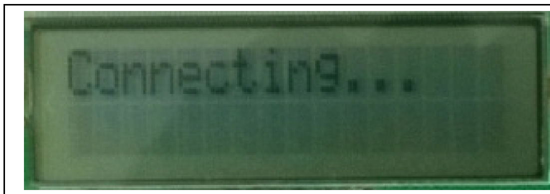
FIGURE 11: INITIATING OR WAITING FOR A CONNECTION REQUEST

4. Node A can select one of the MAC_ADDR (MAC addresses) from the Trusted Device List (TDL) to enable sending a connection request to Node B. Refer to Figure 12.
5. Press switch S1 at Node A to connect to any of the previously connected peer device (in this case, Node B)

FIGURE 12: SELECTING ADDRESSES FROM TRUSTED DEVICE LIST (TDL)

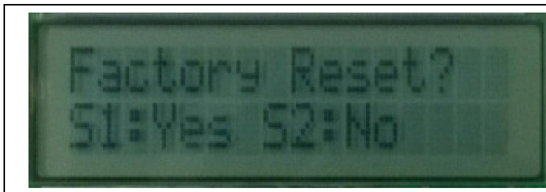
6. An attempt to connect to the selected MAC_ADDR is done (that is, Node A connecting to Node B). Refer to Figure 13. Upon successful connection, the toggling LED (green) becomes stable, see Step 1 of Post Connection. If there are no devices selected in the previous step, then Node A *cannot* connect to any of the devices from TDL.

FIGURE 13: CONNECTING TO THE SELECTED DEVICE/NODE



7. The user can choose whether or *not* a factory reset must be done. Refer to Figure 14.

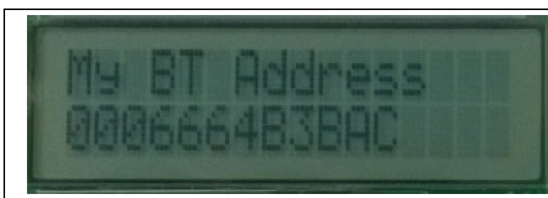
FIGURE 14: FACTORY RESET



Note:	Avoid performing a factory reset before testing the code if the module contains important settings. Upon factory reset, all the factory defaults are restored and any previous settings are erased.

8. The module displays its MAC_ADDR and the user must make a note of the address for further selection and configuration. Refer to Figure 15.

FIGURE 15: DEVICE MAC_ADDRESS DISPLAY

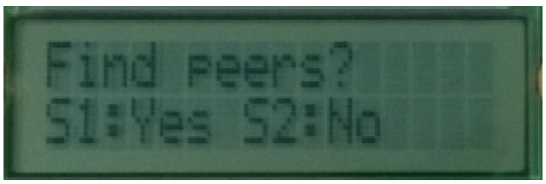


Connecting Two Wireless Nodes

To connect the nodes, follow these steps:

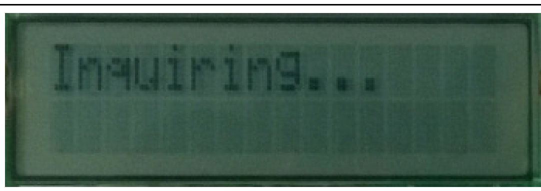
1. At Node A, press switch S1 to find the peer nodes in the network. At Node B, press switch S2 to wait for the request from the connection initiating node/device (Node A).
2. Node A can start the Inquiry/Discovery process to find the Bluetooth devices available, as shown in Figure 16 and Figure 17, to enable sending a connection request to the selected device (Node B). Node B continuously waits for the incoming connection request by *not* carrying out the Inquiry process. Refer to Figure 18.

FIGURE 16: INQUIRY/DISCOVERY PROCEDURE



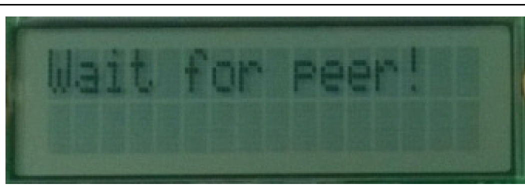
3. Node A now discovers the peer devices and displays the Inquiring process.

FIGURE 17: INQUIRY/DISCOVERY PROCEDURE IN PROGRESS



4. Node B waiting for the connection request from the peer device (Node A). Refer to Figure 18.

FIGURE 18: WAITING FOR A CONNECTION REQUEST

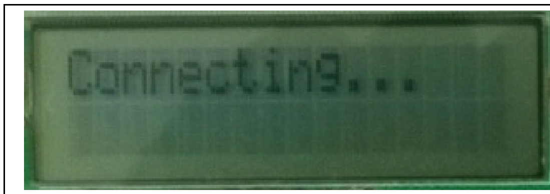


5. At Node A, a list of devices found during the Inquiry/Discovery process is displayed and the user can select one of the MAC_ADDR from the list. Refer to Figure 19.

FIGURE 19: INQUIRY/DISCOVERY SCAN RESULT

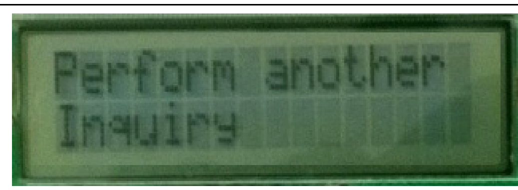
6. After Inquiry/Discovery process, Node B address has to be found from the address list by pressing switch S2. Once the Node B address is found, select the device by pressing switch S1.
7. Node A attempts to connect to the selected device (Node B) as shown in Figure 20, if the device Node B MAC adress is found in the list. Otherwise, a new inquiry scan is performed.

FIGURE 20: CONNECTING TO THE SELECTED DEVICE/NODE



8. If Inquiry process fails to find the expected device, perform inquiry once again at Node A by pressing switch S2 to find peers again (see Step 1) as shown in Figure 21.

FIGURE 21: DISPLAY PROMPTING USER TO PERFORM A NEW INQUIRY/DISCOVERY



Post Connection

When the nodes established a connection, follow these steps:

1. After the two nodes (A and B) established a connection, the toggling green LED becomes stable or solid. Refer to Figure 22. After connection, one of the four data strings, “Message 1”, “Message 2”, “Message 3” or “Message 4” can be selected at the Sending node to be sent to the Receiving node. Refer to Figure 23.
2. At Node A, press switch S1 to send the Message 1. Press switch S2 to send another message to Node B and vice versa.

FIGURE 22: TWO WIRELESS NODES IN CONNECTED STATE

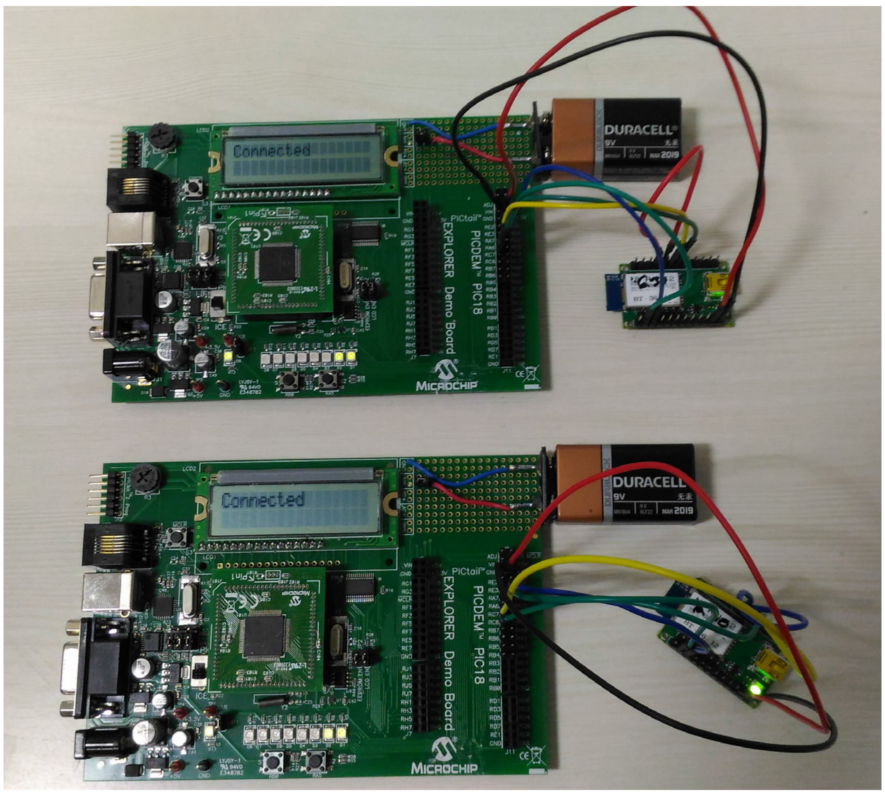
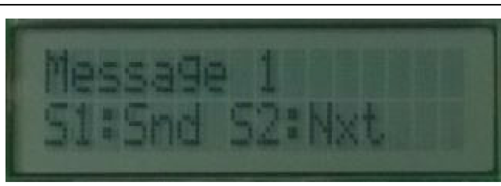
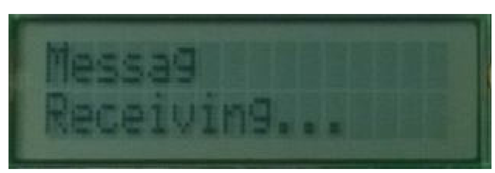


FIGURE 23: DATA STRING SELECTION AT DATA SENDING NODE



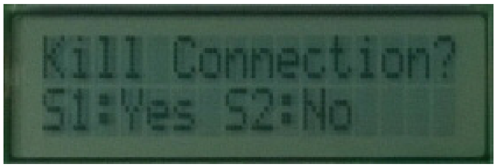
3. The data received at a particular node (Node A/ Node B) displays and can be verified. Refer to Figure 24.

FIGURE 24: DATA STRING GETTING RECEIVED AT THE RECEIVING NODE



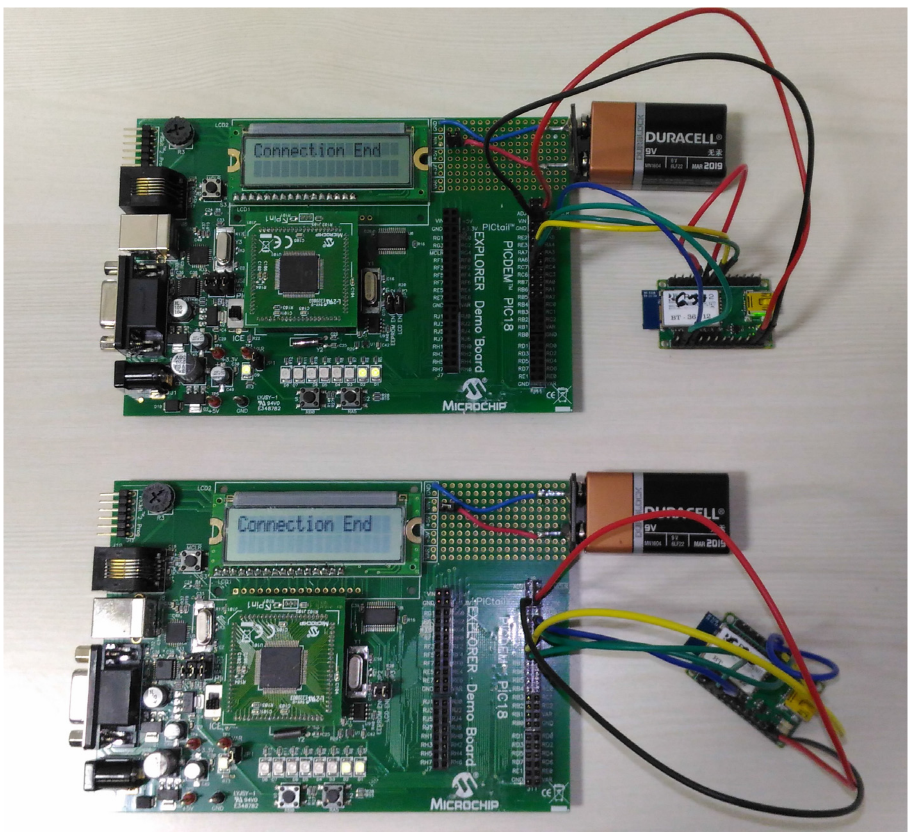
4. The user can choose to end the connection by selecting the options from any of the two nodes as shown in Figure 25.
5. Selecting the Kill Connection option disconnects the two connected nodes. Press S1 to disconnect the devices at Node A or Node B.

FIGURE 25: DISCONNECTING THE TWO CONNECTED WIRELESS NODES



6. The nodes get into the disconnected state as shown in Figure 26.

FIGURE 26: WIRELESS NODES IN DISCONNECTED STATE



After disconnection, the user can restart the Configuration/Connection process for running the demo application again by enabling the Reset () function in `AfterConnect.c` and `Checkresponse.c` file in the project.

CONCLUSION

This application note is designed to enable Microchip Bluetooth customers to acquire basic understanding of interfacing Microchip RN41/42 Bluetooth Classic module with a PIC18 series of microcontrollers (8-bit MCU platform). Thus, the application interface provides a framework to support custom-based applications. This application note also provides sample codes for enabling the RN41/42 modules to act as an Inquiry/ Discovery node and a Connecting node using a 8-bit PIC microcontroller using the ASCII commands supported over UART. The interface and code examples can be further used as a start-up code for any of the user applications or projects using basic Bluetooth SPP profile. However, the application demo and the code is *not* intended to address all of the scenarios and decisions required in creating a wireless solution.

REFERENCES

This section lists the Microchip Technology Inc. documents and other resources that are referenced in this application note.

- *"RN42/RN42N Class 2 Bluetooth Module with EDR Support Data Sheet"* (DS50002328A)
- *"RN41/RN42 Bluetooth Data Module Command Reference User's Guide"*
- *"RN41/42 Evaluation Kit User's Guide"* (DS50002325A)
- *"PICDEM™ PIC18 Explorer Demonstration Board User's Guide"* (DS50001721B)
- *"PIC18F87J11 Family Data Sheet"* (DS39778E)
- *"MPLAB® ICD 3 In-Circuit Debugger User's Guide for MPLAB X IDE"* (DS50002081B)
- *Bluetooth Core Specification 4.1* Adopted Documents: www.bluetooth.org/en-us/specification/adopted-specifications
- *Bluetooth 4.1 GATT Definitions Browser*: https://developer.bluetooth.org/gatt/Pages/Definition-Browser.aspx

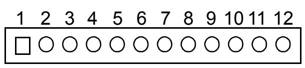
Note:	The referenced documents are identified with a "DS" number. The numbering convention for the DS number is "DSXXXXXXXXA", where "XXXXXXXX" is the document number and "A" is the revision level of the document. Visit the Microchip website to get the latest documentation available.

APPENDIX A: RN-42-EK BOARD SIGNAL HEADERS

Table 2 shows the RN-42-EK board signal headers and the pin descriptions, while Table 3 describes the status LED modes.

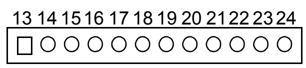
TABLE 2: RN-42-EK BOARD SIGNAL HEADERS

GPIO Signal Header 1 (J1)



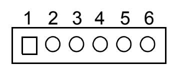
Pin	Description
1	GPIO6
2	GPIO7
3	RESET_N
4	No Connect
5	No Connect
6	No Connect
7	No Connect
8	Sensor Input 1 (AIO1)
9	Shutdown (SHDN)
10	No Connect
11	3.3V
12	GND

GPIO Signal Header 2 (J2)



Pin	Description
13	GPIO4
14	GPIO3
15	GPIO2
16	CTS
17	RTS
18	TXD
19	RXD
20	GPIO11
21	GPIO10
22	GPIO9
23	GPIO8
24	Sensor Input 0 (AIO0)

Programming Header (J3)



Pin	Description
1	SPI_MISO
2	SPI_MOSI
3	SPI_SCK
4	SPI_SS
5	3.3V
6	GND

Configuration Switches

Switch	Related GPIO	Description
1	GPIO4	Restore Factory Defaults
2	GPIO3	Automatic Discovery
3	GPIO6	Automatic Master
4	GPIO7	Default Baud Rate

TABLE 3: STATUS LED

Mode	Green LED (S)	Red LED (A)
Fast blink, 10 times per second	Command mode	—
Blinks twice per second	Boot up, remotely configurable	—
Blinks once per second	Discoverable/idle	Data over the UART
Solid on	Connected	—

APPENDIX B: CONFIGURING THE RN42 MODULES USING ASCII COMMANDS

The terminal emulator programs (for example, TeraTerm for Windows® OS and CoolTerm for MAC OS®) can be used to control and monitor the RN42 module. This is helpful only if you want to work with the PICtail boards without the interface with MCU.

Table 4 and Table 5 list the sequence of commands which must be used to setup a simple Bluetooth connection between two RN42 Bluetooth modules.

TABLE 4: SEQUENCE OF COMMANDS USED FOR THE WIRELESS NODE TO WAIT FOR THE CONNECTION

User ASCII Commands	Expected ASCII Responses
`$$$`	`CMD\r\n`
`+\r\n`	`ECHO ON\r\n`
`SM,0\r\n`	`AOK\r\n`
`SA,0\r\n or SA,4\r\n`	`AOK\r\n`
`R,1\r\n`	`Reboot!\r\n`

TABLE 5: SEQUENCE OF COMMANDS USED FOR THE WIRELESS NODE TO INITIATE THE CONNECTION

User ASCII Commands	Expected ASCII Responses
`$$$`	`CMD\r\n`
`+\r\n`	`ECHO ON\r\n`
`SM,0\r\n`	`AOK\r\n`
`SA,0\r\n or SA,4\r\n`	`AOK\r\n`
`R,1\r\n`	`Reboot!\r\n`
`$$$`	`CMD\r\n`
`+\r\n`	`ECHO ON\r\n`
`I,10\r\n`	`Inquiry,T=10,COD=0\r\nFound <x>\r\n<List of MAC_Addresses>\r\n`
`C,<MAC_Address>\r\n`	`Connected\r\n`
`K,\r\n`	`Disconnected\r\n`

> **Note:** When the modules are connected, the data bytes that are sent from one BT module are received by the other BT module, and vice versa. Later, if required, the modules can be disconnected.

APPENDIX C: SOURCE CODE

All softwares covered in this application note are available as a single WinZip archive file. This archive file can be downloaded from the Microchip website at: www.microchip.com

C.1 Source Code File List

Table 6 provides the list of files that are used as part of the Application Demo.

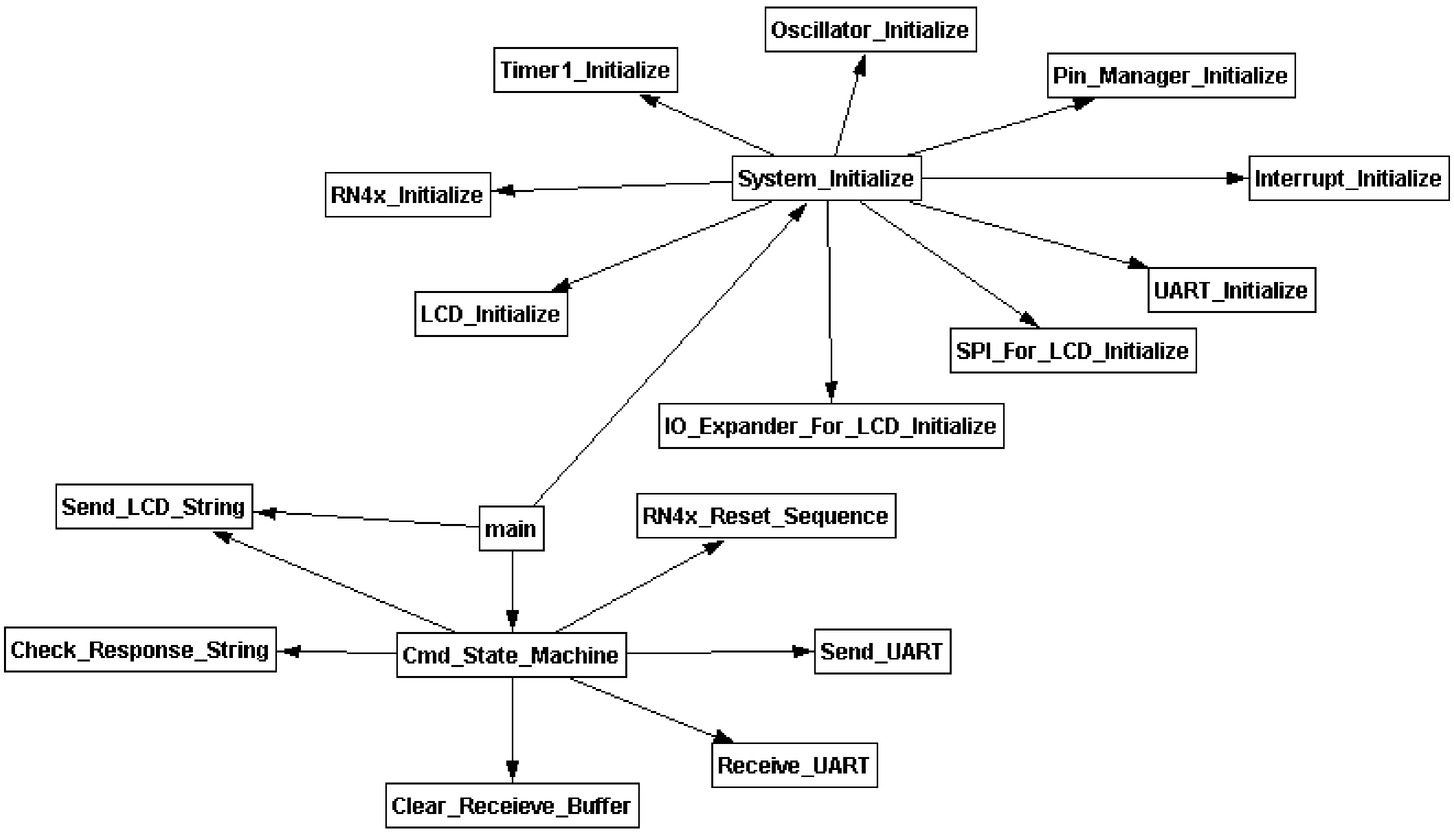
TABLE 6: SOURCE CODE FILE LIST

File name	File type	Description
Main	`.c` and `.h`	Handles the state machine which is used to demonstrate the Demo Application
Init	`.c` and `.h`	Handles the initialization of the PIC18F87J11 and its various peripherals
UART	`.c` and `.h`	Handles the UART peripheral of PIC18F87J11
CheckResponse	`.c` and `.h`	Checks the response strings obtained from the RN42 Bluetooth module
AfterConnect	`.c` and `.h`	Handles the transfer and display of data strings which are exchanged between the Bluetooth modules
LCD	`.c` and `.h`	LCD interface
Globals	`.c` and `.h`	Contains global variables used in the code

C.2 Call Graph

Figure 27 shows the source code call graph.

FIGURE 27: SOURCE CODE 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.

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 ═

Trademarks

ISBN: 978-1-5224-0774-4

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: 905-695-1980
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 - Chongqing
Tel: 86-23-8980-9588
Fax: 86-23-8980-9500

China - Dongguan
Tel: 86-769-8702-9880

China - Guangzhou
Tel: 86-20-8755-8029

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

ASIA/PACIFIC

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

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

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: 886-7-213-7828

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 - Karlsruhe
Tel: 49-721-625370

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

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

06/23/16