INTRODUCTION

Nowadays, most of the embedded applications require real-time communications to support various applications and user environment. Bluetooth® has emerged as the standards of choice for connecting local embedded applications within operable range, a requirement in line with most of the Internet of Things (IoT) technology.

Bluetooth Classic (BTC) technology is originally designed for continuous data and voice streaming applications, and has successfully eliminated wires in many consumers, industrial, and medical applications. Classic Bluetooth technology continues to provide a robust wireless connection and can stream data between devices, ranging from infotainment in cars to industrial controllers and medical sensors.

Bluetooth Low Energy (BLE) technology is introduced through Bluetooth version 4.0 specification 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 consumption, has unique features, and also supports new services/profiles. Coin cell battery-operated sensors and actuators in medical, consumer, and fitness applications can now smoothly connect to BLE technology enabled smartphones, tablets or gateways. BLE is ideal for applications requiring periodic transfer of small amounts of data.

Bluetooth Classic and BLE 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 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 RN4677 module and the PIC18 (8-bit) microcontroller over the UART using the ASCII commands. It essentially supports BTC, BLE, and Dual mode configurations.

This application note also showcases switching between BTC and BLE modes and vice versa to realize a Gateway using RN4677 Dual Mode module. The RN4677 communicates both in SPP and Transparent UART modes to send messages via the Gateway node.

CLASSIC BLUETOOTH COMMUNICATION

Bluetooth is a wireless technology standard for exchanging data over short distances from fixed and mobile devices which are part of the Personal Area Networks (PANs). Classic Bluetooth is characterized to provide easy, temporary connectivity to smartphones and tablets, and is supported by Android™ and iOS® applications. It provides a convenient cable replacement option for applications, such as audio and data streaming between devices. Bluetooth initially supported 1 Mbps data transfer rate (Bluetooth 1.2) that has increased to 3 Mbps with the Enhanced Data Rate version (Bluetooth 2.1 + EDR), and further advanced to a high-speed version (Bluetooth 3.0 + HS) to support large file transfers.

Bluetooth Classic uses short-wavelength UHF radio waves that are part of the globally unlicensed Industrial, Scientific and Medical (ISM) 2.4 GHz frequency band. Bluetooth uses frequency-hopping spread spectrum. Classic Bluetooth divides transmitted data into packets, and transmits each packet in one of the 79 designated channels. Bluetooth operates at frequencies between 2400 MHz and 2483.5 MHz that 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. It usually performs 1600 hops per second, with Adaptive Frequency Hopping (AFH) enabled. The maximum transmit power within the band is limited to 10 mW as per ISM standards.
BLUETOOTH SMART COMMUNICATION

Bluetooth Low Energy is intended for energy-constrained applications such as sensors or disposable devices, and for low-duty cycle devices which support low-data throughput which can operate for a longer duration compared to other protocols from a coin cell battery. Key benefits of implementing the technology are inexpensive silicon, much less MCU processing requirements, and reduced memory. These are suitable for applications related to the temperature, proximity, alerts, fitness, and sports which represents a connectivity bubble belonging to the Body Area Network (BAN).

BLE operates in the spectrum band of 2.400 GHz to 2.4835 GHz, same as Classic Bluetooth technology, but uses a different set of channels. BLE operates in 40 channels, each of 2 MHz wide. Within a channel, data is transmitted using Gaussian Frequency Shift Keying (GFSK) modulation technique, which is similar to Classic Bluetooth's FSK modulation. The maximum over-the-air bit rate is 1 Mbps, and the maximum transmit power is 10 mW.

For additional information related to Bluetooth and its specifications, refer to the “Bluetooth Core Specification V4.0” from the following web site: http://www.bluetooth.org

SIMILARITIES IN BLUETOOTH TECHNOLOGIES AND ARCHITECTURE

Bluetooth Classic and BLE both operate in the 2.4 GHz ISM band and have similar Radio Frequency (RF) output power; however, because a BLE device is in Sleep mode most of the time and wakes up only during data transfer, hence this reduces the power consumption as the number of connection times are only a few milliseconds. BLE connections lose potential power savings as the utilization approaches continuous transmission.

Originally, GFSK modulation is the only modulation scheme available. After the introduction of Bluetooth 2.0+EDR, π/4-DQPSK (Differential Quadrature Phase Shift Keying) and 8DPSK modulation schemes 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. The term Enhanced Data Rate (EDR) is used to describe π/4-DQPSK and 8DPSK schemes, each supporting 2 and 3 Mbps, respectively.

Many features of Classic Bluetooth technology are inherited in Bluetooth Low Energy technology, including the Adaptive Frequency Hopping (AFH) and also part of the logical link control and adaptation protocol (L2CAP) interface. Bluetooth Low Energy technology also implements the same link security with simple pairing modes, secure authentication, and encryption. These inheritances make BLE devices easy to setup, robust, and reliable in rough and varying environments.

Bluetooth protocol supports Master-Slave network architecture. One master can communicate with 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.

DIFFERENCES IN BLUETOOTH TECHNOLOGIES AND PROFILE SUPPORT

The behavior of a Bluetooth connection, whether Classic or Low Energy, is determined by the Bluetooth profiles or services that the device/module supports. Devices can exchange data only if they both share a common Bluetooth profile/service implemented in it. However, there are differences in the available profiles in Classic Bluetooth technology compared to the services in Bluetooth Low Energy technology.

The Classic Bluetooth profiles include Headset (HSP), Hands Free (HFP), Object Exchange (OBEX), Audio Distribution (A2DP), Video Distribution (VDP), and File Transfer (FTP). Many other profiles are not offered for Bluetooth Low Energy due to the differences in the connection models. Bluetooth Low Energy also supports a lot of services through profiles. The BLE profiles are based on the Generic Attribute Profile (GATT), a general specification for sending and receiving short pieces of data known as attributes over a low-energy link. Typical profiles include Health Care, Sports and Fitness, Proximity, Alert and Battery, supporting related services.

Differences are also seen in the serial port emulation. For example, Classic Bluetooth supports the Serial Port Profile (SPP) for emulation of serial data connections. BLE technology, on the other hand, provides no such support in the standard Specification v4.0 although many suppliers provide different services to emulate serial connections. Microchip provides a good level support on (Serial/UART) profiles like Microchip Low-energy Data Profile (MLDP v1 and v2) with RN4020 BLE modules, and services such as Transparent UART with RN4677 Dual Mode module.

In Bluetooth Low Energy specification, the master can be connected to several slave devices. The number of slaves can be very large depending on the
implementation and available memory. In case of BLE, the peripheral (slave) devices can advertise via advertisement packets to the central (master) devices that are scanning for peripherals. The advertisement packets are editable to contain custom information.

**Note:** 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.

### SERIAL PORT PROFILE (SPP)

The SPP defines the specific protocol format and procedures for devices using Bluetooth for RS232 serial cable emulation. SPP is one of the most frequently used Bluetooth profiles to replace RS232 cables as it enables sending bursts of data between two devices. There are no fixed Master/Slave roles in this profile. The Radio Frequency Communication (RFCOMM), transport layer of Bluetooth, is used to transport the user data, modem control signals, and configuration commands.

For the execution of the SPP profile, use of security 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 this profile, therefore support for this is optional.

### TRANSPARENT UART

In addition to SPP for Bluetooth Classic connectivity, the RN4677 introduces a private GATT service for serial data transfer between two BLE devices. This BLE data streaming service provided in the RN4677 is named Transparent UART.

### SINGLE MODE AND DUAL MODE BT DEVICE TOPOLOGIES

Bluetooth Smart Ready indicates a Dual mode device compatible with both Classic and Low Energy peripherals. Bluetooth Smart indicates a Low Energy only device which requires either a Smart Ready or another Smart device in order to function.

The two Bluetooth technologies are fundamentally different, giving two implementation options:

- **Single mode devices** – These devices are stand-alone Bluetooth Low Energy devices or Bluetooth Classic devices are usually optimized for small battery-operated devices with low cost and low-power consumption in focus. A typical Single mode device is a heart rate sensor.

- **Dual mode devices** – These devices (also known as Smart Ready devices) include both Bluetooth Low Energy technology and Classic Bluetooth technology. Dual mode devices rarely gain in power saving as there is a need to support both technology implementations. Typical Dual mode devices are mobile phone, PC or an Embedded Gateway.

### MICROCHIP RN4677 DUAL MODE BLUETOOTH MODULE

The Microchip RN4677 is a fully certified Bluetooth version 4.0 (BR/EDR/LE) Dual Mode module, which enables the designers to easily add Classic Bluetooth and Bluetooth Smart capability to their products. Delivering local connectivity for the IoT, the RN4677 bridges the product to smartphones and tablets to ensure convenient data transfer, control, and access to cloud applications. This Bluetooth SIG certified module provides a complete wireless solution with Bluetooth stack on board, integrated antenna, and worldwide radio certifications in a compact surface mount package, 22 x 12 x 2.4 mm.

Figure 1 illustrates the RN4677 module mounted on the RN4677 PICtail™/PICtail Plus Daughter board. It supports Generic Access Profile (GAP), Service Discovery Protocol (SDP), SPP, and GATT profiles. Data is transferred over the Bluetooth link via SPP for Bluetooth Classic and Transparent UART for Low Energy, making it easy to integrate with any processor or microcontroller with a UART interface. Configuration is easily made through ASCII commands via UART.

**FIGURE 1:** RN4677 PICtail™/PICtail Plus Daughter Board

---

© 2016 Microchip Technology Inc.
A Microcontroller Unit (MCU) or host processor sends commands to configure module features, read status, and to manage Bluetooth data connections. The UART TX and RX lines are required to communicate with the module and transfer data through the Bluetooth SPP/Transparent UART connection. Connecting the hardware flow control lines, CTS and RTS, is highly recommended for applications that transmit a continuous stream of data.

**Note:** The RN4677 Bluetooth module can be configured over the Bluetooth link or through the module’s UART using a simple ASCII command language by entering Command mode. Set commands configure the module and Get commands echo the configuration. For details on ASCII commands, refer to the “RN4677 Bluetooth® 4.0 Dual Mode Module User’s Guide” (DS50002377).

**RN4677 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 via a Gateway node, data in the form of strings/characters are transferred between these nodes, showcasing the SPP profile in BTC mode and Transparent UART service in BLE mode, both emulating the serial RS232 type of connection.

This application note provides the users with the following functionalities:

- Framework for any user application platform using RN4677 Dual Mode Bluetooth module and PIC18F series of microcontrollers
- Specific interface between the RN4677 Bluetooth module and the PIC18F87J11 microcontroller
- Reference source code to manage BTC, BLE, and Dual mode connections of RN4677 module through PIC microcontroller
- Technique to enable switching between BTC and BLE modes and vice versa to implement a Gateway using RN4677 Dual Mode module
- Demonstration of the Classic Bluetooth technology Serial Port Profile (SPP) and the Bluetooth Low Energy Private Service called Transparent UART for emulation of serial data connections

The hardware interface of the RN4677 module with any of the PIC microcontrollers can be called a wireless node. In this demo, the interface between the RN4677 PICtail/PICtail Plus Daughter board and the PIC18 Explorer Development board is considered as a BT wireless node. Refer to Figure 2. The demo uses commands issued by the PIC18F87J11 microcontroller to configure and setup the BT wireless nodes.

There are three components/node types in this demo application:

1. Node A - BTC/BLE device
2. Node C - BTC/BLE device
3. Node B - Gateway (Dual mode) device

One of the nodes, say Node A, has to be configured to be either BTC or BLE node and for either Data Sending or Receiving node. The other node, say Node B, is configured as a Gateway node. Node C follows the complementary configuration of Node A (BLE or BTC).

User input or data transmission are enabled through one of the switches on the PIC18 Explorer Development board of Node A and is sent to Node C via the Gateway (Node B) and vice versa. Messages and status information is periodically displayed on the LCD display of the PIC18 Explorer Development board (on all of the nodes). After successfully establishing a Bluetooth connection between the nodes, data strings are transferred between these nodes showcasing the SPP in BTC mode, and showcasing the Transparent UART service in BLE mode.

**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:

- Three Microchip RN4677 PICtail/PICtail Plus Daughter boards
- Three PIC18 Explorer Development boards with PIC18F87J11 PIMs mounted
- Any of the following Microchip development tools for programming and debugging purposes: MPLAB® REAL ICE™ In-Circuit Emulator, MPLAB ICD 3 or PICkit™ 3
- Three power supplies: 9V/0.75A

**Note:** The application note demo code uses RN4677 module mounted on RN4677 PICtail/PICtail Plus Daughter boards.
SOFTWARE/UTILITY REQUIREMENTS

This demo application intends to showcase gateway communication between two Classic or Low Energy Bluetooth wireless nodes using the Dual mode RN4677 modules. 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 web site at www.microchip.com. The code is compiled using the Microchip XC8 compiler v1.34 and MPLAB X IDE v3.05.

Note: The RN4677 modules must have firmware version 1.00 and above for the demo code to work. To know the details of the firmware version, refer to the product page from the Microchip website.

Demo source code is available for download from the Documentation and Software link section of the RN4677 product page at www.microchip.com/RN4677.

FIGURE 2: APPLICATION DIAGRAM OF COMMUNICATION VIA GATEWAY USING BT NODES

<table>
<thead>
<tr>
<th>BTC Node (BLE Node)</th>
<th>Gateway Node</th>
<th>BLE Node (BTC Node)</th>
</tr>
</thead>
<tbody>
<tr>
<td>PIC® 8-bit MCU</td>
<td>PIC 8-bit MCU</td>
<td>PIC 8-bit MCU</td>
</tr>
<tr>
<td>RN4677 Module</td>
<td>RN4677 Module</td>
<td>RN4677 Module</td>
</tr>
<tr>
<td>Node A</td>
<td>Node B</td>
<td>Node C</td>
</tr>
</tbody>
</table>

Use the precompiled BTC_BLE_TL Node.X.production.hex file of the demo or compile the BTC_BLE_TL Node project code if required. Ensure the compilation is successful. For additional information on the source code, related files with description, and call graph, refer to Appendix A: "Source Code". From MPLAB X, user can generate call graphs related to specific functions of the demo code.
HARDWARE DEMO SETUP

This RN4677 based communication requires three wireless nodes. The demo setup consists of three PIC18 Explorer Development boards interfaced with three identical RN4677 PICtail/PICtail Plus Daughter boards as shown in Figure 2. Thus, the three identical RN4677 Dual Mode module-based wireless nodes are used for this application demonstration. For more information on the RN4677 module, refer to the “RN4677 PICtail™/PICtail Plus Board User’s Guide” (DS50002388).

PIC18 Explorer Development Board and RN4677 Module Connections

The RN4677 PICtail/PICtail Plus Daughter board is inserted into the connector socket of the PIC18 Explorer Development board. This connection supplies 3.3V power, 2-wire or 4-wire UART, wake-up, SW_BTN, and Reset functions to the RN4677 module from the microcontroller. Table 1 shows the connection details of the RN4677 PICtail/PICtail Plus Daughter board to the PIC18 Explorer Development board.

Figure 3 illustrates pin to pin connections used in the application demo code between the PIC18F87J11 MCU PIM residing on the PIC18 Explorer Development board and the RN4677 Bluetooth Dual Mode module mounted on the RN4677 PICtail/PICtail Plus Daughter board.

### TABLE 1: CONNECTION DETAILS BETWEEN THE RN4677 PICTAIL™/PICTAIL PLUS DAUGHTER BOARD AND THE PIC18 EXPLORER DEVELOPMENT BOARD

<table>
<thead>
<tr>
<th>RN4677 PICtail™/PICtail Plus Daughter Board</th>
<th>PIC18 Explorer Development Board</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Signal Header (J1)</strong></td>
<td><strong>Pin No.</strong></td>
</tr>
<tr>
<td>BT_UART_TXD</td>
<td>9</td>
</tr>
<tr>
<td>BT_UART_RXD</td>
<td>11</td>
</tr>
<tr>
<td>BT_WAKE_UP</td>
<td>19</td>
</tr>
<tr>
<td>BT_UART_CTS</td>
<td>25</td>
</tr>
<tr>
<td>BT_UART_RTS</td>
<td>6</td>
</tr>
<tr>
<td>BT_RST_N</td>
<td>18</td>
</tr>
<tr>
<td>SW_BTN</td>
<td>1</td>
</tr>
<tr>
<td>EXT_3V3/VDD</td>
<td>26</td>
</tr>
<tr>
<td>GND</td>
<td>28</td>
</tr>
</tbody>
</table>

**Note:** VAR/VDD (J9) and GND (J9) indicate the only connections from J9 connector. All the other connections are from J3 connector.

### FIGURE 3: PIC18F87J11 TO RN4677 BLUETOOTH INTERFACE DIAGRAM
Hardware Modifications Required in RN4677 PICtail/PICtail Plus Daughter Board

When using the RN4677 PICtail Daughter board with the PIC18 Explorer Development board, the user must be careful not to insert the PICtail directly into the J3/J9 connector.

Prior use, ensure to perform the following steps on the RN4677 PICtail/PICtail Plus Daughter board:

1. Remove P3_2 (Pin 8), P3_3 (Pin 10), and P3_4 (Pin 12) pins from the RN4677 PICtail/PICtail Plus Daughter board as these pins interfere with the functionality of the microcontroller pins.

2. The BT_RST_N pin should be separately connected to the RE0/AD8/PMRD(3)/P2D pin on the PIC18 Explorer Development board.

Figure 4 illustrates the RN4677 board connected to the PIC18 Explorer Development board.

**FIGURE 4: RN4677 BOARD CONNECTED TO THE PIC18 EXPLORER DEVELOPMENT BOARD**
GETTING STARTED

Setting Up the Bluetooth Nodes

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

1. Insert the RN4677 PICtail/PICtail Plus Daughter board into the connector socket of the PIC18 Explorer Development board.
2. Make note of the switch S4 and set the following jumper positions in the PIC18 Explorer Development board before running the demo code:
   - JP3 to enable LCD
   - JP1 to enable LEDs
   - J13 to ensure that the communication is routed through the RS-232 socket
   - J4 to ensure that the main PIC is programmed
   - Switch S4 to enable Processor In Module (PIM) (pointing towards MPLAB REAL ICE In-Circuit Emulator when ON)

   Figure 5 shows the position of jumpers and switches on the PIC18 Explorer board.

3. Use switches (S1 and S2) and LCD (LCD1) on the PIC18 Explorer Development board 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 5.
6. Open the downloaded application demo source code BTC_BLE_TL Node.x and compile in Release mode if required. Alternatively, use the pre-compiled BTC_BLE_TL Node.hex file available in the downloaded folder.
7. The generated or pre-compiled BTC_BLE_TL Node.hex file can be programmed into the three wireless nodes, A, B, and C, using any of the Microchip programmers supporting the PIC18F87J11.
8. The boards are ready to run the demo. If needed, the user must perform 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 5.
2: For the application demo, a 9VMN1604 (9V-6LF22, alkaline manganese dioxide) battery is used.

For additional information on programming and 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 5: SWITCH AND JUMPER POSITIONS ON THE PIC18 EXPLORER BOARD
Application Block Diagram and Flow Chart

Figure 6 shows the Block Diagram of the Application Demo and the components of each BT node.

**FIGURE 6: APPLICATION BLOCK DIAGRAM WITH INPUT-OUTPUT INTERFACES**
Running the Demo Application

Running the RN4677 demo application involves the following steps:

1. Configure the three RN4677 wireless nodes as follows:
   - One of the wireless nodes (for example, Node B) acts as a Gateway node in Dual mode.
   - From the two remaining wireless nodes (for example, Node A and Node C), configure Node A to be either BTC/BLE node for Sending/Receiving data and then configure Node C in complementary mode to Node A as shown in Table 2.

2. Establish connection between Node A and Node B (Gateway node).

3. Send and receive the data strings between nodes A and C using Node B (via Gateway node). The Gateway node receives data from Node A and then sends to Node C using the appropriate BTC or BLE protocol format.

4. Kill connections and restart the demo.

The LCDs 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.

TABLE 2: CONFIGURATION OPTIONS FOR NODES A AND C

<table>
<thead>
<tr>
<th>Options</th>
<th>Node A</th>
<th>Node C</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>BTC-SendTime</td>
<td>BLE-Receiving</td>
</tr>
<tr>
<td>2</td>
<td>BTC-Receiving</td>
<td>BLE-Sending</td>
</tr>
<tr>
<td>3</td>
<td>BLE-Sending</td>
<td>BTC-Receiving</td>
</tr>
<tr>
<td>4</td>
<td>BLE-Receiving</td>
<td>BTC-Sending</td>
</tr>
</tbody>
</table>

Note: When the user configures Node A to be a BTC/BLE Sending Device and Node C as a BTC/BLE Receiving Device, Node A inquires for devices nearby and initiates connection while Node B, which is a Gateway Node, acknowledges the request and then connects to Node A for the data transfer.
Figure 7 through Figure 9 illustrate the complete cycle of the application demo.

FIGURE 7: APPLICATION DEMO FLOW CHART
FIGURE 8: APPLICATION DEMO FLOW CHART (CONTINUED)

3
Inquire Gateway/Translation node?
Yes
Check all the addresses displayed
No
Is any of the addresses a Gateway node?
Yes
Gateway node and the Sending node are connected

5
Inquire Gateway/Translation node?
Yes
Check all the addresses displayed
No
Is any of the addresses a Gateway node?
Yes
FIGURE 9: APPLICATION DEMO FLOW CHART (CONTINUED)

1. Make note of the local BT address
   - Factory Reset?
     - Yes: Factory Reset complete
     - No: Wait for connection request from the Sending node
       - Connection successful?
         - No: Disconnected from Sending node?
           - Yes: Initiate a connection request to the Receiving node
           - No: Wait for connection request from the Sending node
         - Yes: Receiving data from the Sending BT node
           - Disconnected from Sending node?
             - Yes: Initiate a connection request to the Receiving node
             - No: Connection successful?
               - No: Disconnect from Sending node?
                 - Yes: End
                 - No: Send all the data received from the Sending node to the Receiving node and then disconnect
               - Yes: Send all the data received from the Sending node to the Receiving node and then disconnect

2. Waiting for connection request from Gateway node
   - Connection successful?
     - No: Display all the data received from Gateway/Translation node
     - Yes: End

4. Display all the data received from Gateway/Translation node
   - Disconnect from Gateway node?
     - Yes: End
     - No: Display all the data received from Gateway/Translation node
Configuring the Three Wireless Nodes

After programming the boards and connecting the PIC18 Explorer Development boards (BT nodes) to the power supply, switch on the supply for all nodes. The LCD on the boards displays the message as shown in Figure 10.

To configure the nodes, follow these steps:

1. Configure one of the three wireless nodes as a Gateway node. Press S1 or S2 to configure. Once a node is configured as a Gateway node, see Step 1 of Configuring and Connecting with Gateway Node.

FIGURE 10: CONFIGURING WIRELESS NODE AS GATEWAY NODE

2. Configure the two remaining nodes as Bluetooth Classic (BTC) node and Bluetooth Low Energy (BLE) node. Refer to Figure 11. If a node is configured as a BTC node, see Step 1 of Configuring, Connecting, and Sending Data with BTC Node. Otherwise, if a node is configured as a BLE node, see Step 1 of Configuring and Connecting with BLE Node.

FIGURE 11: CONFIGURING WIRELESS NODE AS BTC OR BLE NODE
Configuring, Connecting, and Sending Data with BTC Node

1. At this stage, the user can decide if a factory reset is required. Refer to Figure 12.

   **Note:** Only a few settings such as the mode, authentication method, and extended status string are considered in the demo code. For other settings affecting the code execution, a factory reset is recommended.

   **FIGURE 12:** CHOOSING FACTORY RESET

2. Node A (BTC node) displays its MAC_ADDR (MAC address) and the user must make a note of the address to configure connections after the Inquiry process. Refer to Figure 13.

   **FIGURE 13:** BTC NODE DISPLAYING ITS OWN MAC ADDRESS
3. The user can decide whether to send data from the node or to receive data at the node. Refer to Figure 14. For reception of data, see Step 1 of Data Receiving with BLE Node.

**FIGURE 14: CONFIGURING WIRELESS NODE AS DATA SENDING OR DATA RECEIVING NODE**

4. Start inquiry for BLE node if BTC node is selected as the Data Sending (Inquiry) node and the BLE node as the Data Receiving (Joining/Connecting) node. Refer to Figure 15.

**FIGURE 15: INQUIRY PROCEDURE FOR BLE NODES**
5. Select one BLE device from the list displayed and the data is sent to this node. Refer to Figure 16. If there are no devices selected, the user is prompted to carry out the Inquiry process again. Refer to Figure 17.

FIGURE 16: INQUIRY SCAN RESULTS

FIGURE 17: REPEAT INQUIRY PROCEDURE
6. Start an inquiry for the Gateway node which is a Dual Mode node (Smart Ready). Refer to Figure 18.

FIGURE 18: INQUIRY PROCEDURE FOR GATEWAY NODES

7. Select a Gateway node to enable data transfer between BTC and BLE nodes. Refer to Figure 19. If there are no devices selected, the user is prompted to carry out the Inquiry process again. Refer to Figure 20.

FIGURE 19: INQUIRY SCAN RESULTS
8. An attempt to connect to the selected Gateway node is done. Refer to Figure 21. If the connection is successful, see Step 1 of Connection between Data Sending Node and Gateway Node.
Configuring and Connecting with BLE Node

1. The user can decide if a factory reset is required. Refer to Figure 22.

Note: Only a few settings such as the mode, authentication method, and extended status string are considered in the demo code. For other settings affecting the code execution, a factory reset is recommended.

FIGURE 22: CHOOSING FACTORY RESET

2. Node C (BLE node) displays its MAC_ADDR (MAC address) and the user must make a note of the address to configure connections after the Inquiry process. Refer to Figure 23.

FIGURE 23: BLE NODE DISPLAYING ITS OWN MAC ADDRESS
3. The user can decide whether to send data from
the node or to receive data at the node. Refer to
Figure 24. For data reception, see Step 1 of
Data Receiving with BLE Node.

FIGURE 24: CONFIGURING WIRELESS NODE AS DATA SENDING OR DATA RECEIVING NODE

4. Start inquiry for BTC node if BLE node is
selected as the Data Sending node and the BTC
node as the Data Receiving node. Refer to
Figure 25.

FIGURE 25: INQUIRY PROCEDURE FOR BTC NODES
5. Select one BTC device from the list displayed and the data is sent to this node. Refer to Figure 26. If there are no devices selected, the user is prompted to carry out the Inquiry process again. Refer to Figure 27.

FIGURE 26: INQUIRY SCAN RESULTS

FIGURE 27: REPEAT INQUIRY PROCEDURE
6. Start an inquiry for the Gateway node which is a Dual Mode node (Smart Ready). Refer to Figure 28.

FIGURE 28: INQUIRY PROCEDURE FOR GATEWAY NODES

7. Select a Gateway node to enable data transfer between BTC and BLE nodes. Refer to Figure 29. If there are no devices selected, the user is prompted to carry out the Inquiry process again. Refer to Figure 30.

FIGURE 29: INQUIRY SCAN RESULTS
8. An attempt to connect to the selected Gateway node is done. Refer to Figure 31. If the connection is successful, see Step 1 of Connection between Data Sending Node and Gateway Node.
Configuring and Connecting with Gateway Node

1. Node B (Gateway node) displays its MAC_ADDR (MAC address) and the user must make a note of the address. Refer to Figure 32.

FIGURE 32: GATEWAY NODE DISPLAYING ITS OWN MAC ADDRESS

2. The user can decide if a factory reset is required. Refer to Figure 33.

Note: Only a few settings such as the mode, authentication method, and extended status string are considered in the demo code. For other settings affecting the code execution, a factory reset is recommended.

FIGURE 33: CHOOSING FACTORY RESET
3. The Gateway node enters the Wait state where it waits for a connection request from the BTC/BLE Data Sending node. Refer to Figure 34.

**FIGURE 34:** GATEWAY NODE WAITING FOR CONNECTION REQUEST FROM SENDING NODE

**Connection between Data Sending Node and Gateway Node**

1. Once the Gateway node and the Data Sending node established a connection, one of the four data strings “Message 1”, “Message 2”, “Message 3”, “Message 4”, or the combination of these strings can be selected at the Data Sending node to be sent to the Gateway node. Refer to Figure 35.

**FIGURE 35:** SENDING NODE AND GATEWAY NODE IN CONNECTED STATE
2. After the data is sent, the user can end the connection from the Data Sending node. Refer to Figure 36.

FIGURE 36: DISCONNECTING SENDING NODE AND GATEWAY NODE

3. When the user opted to end the connection, the Gateway node, which stored all the data received from the Data Sending node, attempts to connect to the Data Receiving node. Refer to Figure 37.

FIGURE 37: GATEWAY NODE CONNECTING TO RECEIVING NODE
Connection between Gateway Node and Data Receiving Node

1. The Gateway node and the Data Receiving node established a connection and the Gateway node sends the data received from the Data Sending node to the Data Receiving node. Refer to Figure 38.

FIGURE 38: GATEWAY NODE SENDING DATA TO THE RECEIVING NODE

2. Once the data transfer is completed, the Gateway node disconnects from the Data Receiving node. Refer to Figure 39 and Figure 40.

FIGURE 39: GATEWAY NODE DISCONNECTING FROM RECEIVING NODE
Data Receiving with BLE Node

1. The Data Receiving node continuously waits for a connection request from the Gateway node. Refer to Figure 41.
2. If the connection request is received, see Step 1 of Connection between Gateway Node and Data Receiving Node.

Note: 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

Considering both Bluetooth technologies, it is important to understand that adding of Bluetooth communication to any application is simple. For some applications, Classic is the best choice while for others Low Energy is the best choice. In other words, both Bluetooth Low Energy technology and Classic Bluetooth technology are irreplaceable in their own application space. BLE technology is quite different from Classic Bluetooth technology; hence the user must carefully consider which feature best fits the application needs. With the introduction of BT technology newer capabilities, there has been a great impact on the interest in developers and the market regarding its capabilities and possible applications.

This application note is designed to enable Microchip Bluetooth customers to acquire basic understanding of interfacing Microchip RN4677 Dual Mode module with a PIC18 series of microcontrollers (8-bit MCU platform) and the requirements and configuration methods to communicate from the BTC node to a BLE node via a communication Gateway using RN4677 Dual Mode module. The Gateway functions help in understanding the configuration changes required for protocol transition between the BTC and BLE modes when used in specific application. It also showcases the BT application developers, on how to implement the Low Data Rate Streaming with Serial Port Profile and the Transparent UART Service using the Microchip RN4677 Dual Mode module and the 8-bit PIC® Microcontroller interface.

This application note also provides sample source code related to PIC18 MCU for enabling the RN4677 Dual Mode module to function as a BTC device (Inquiry node), a BLE device (Connecting node), and as a Gateway node by using its Dual Mode feature. The interface and code examples can be further used as a framework for any of the user applications or projects using SPP profile and Transparent UART service.

REFERENCES

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

Microchip Technology Inc. Resources:
- “RN4677 Bluetooth® Dual Mode Module Data Sheet” (DS50002370A)
- “RN4677 Bluetooth® 4.0 Dual Mode Module User’s Guide” (DS50002377A)
- “RN4677 PICtail™/PICtail Plus Board User’s Guide” (DS50002388A)
- “PICDEM™ PIC18 Explorer Demonstration Board User’s Guide” (DS00051721B)
- “PIC18F87J11 Family Data Sheet” (DS39778E)
- “MPLAB® ICD 3 In-Circuit Debugger User’s Guide for MPLAB X IDE” (DS50002081B)

Specification References:
- 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: SOURCE CODE

Software License Agreement

The software supplied herewith by Microchip Technology Incorporated (the “Company”) is intended and supplied to you, the Company’s customer, for use solely and exclusively with products manufactured by the Company.

The software is owned by the Company and/or its supplier, and is protected under applicable copyright laws. All rights are reserved. Any use in violation of the foregoing restrictions may subject the user to criminal sanctions under applicable laws, as well as to civil liability for the breach of the terms and conditions of this license.

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

All 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

A.1 Source Code File List

Table 3 provides the list of files that are used as part of the Application Demo.

<table>
<thead>
<tr>
<th>File name</th>
<th>File type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Main</td>
<td>.c and .h</td>
<td>Initializes the state machine which is used to demonstrate the Demo Application</td>
</tr>
<tr>
<td>Init</td>
<td>.c and .h</td>
<td>Handles the initialization of the PIC18F87J11 and its various peripherals</td>
</tr>
<tr>
<td>StateMachines</td>
<td>.c and .h</td>
<td>Handles the state machines of the three wireless Bluetooth nodes</td>
</tr>
<tr>
<td>UART</td>
<td>.c and .h</td>
<td>Handles the UART peripheral of the PIC18F87J11</td>
</tr>
<tr>
<td>CheckResponse</td>
<td>.c and .h</td>
<td>Checks the response strings obtained from the RN4677 Bluetooth module</td>
</tr>
<tr>
<td>AfterConnect</td>
<td>.c and .h</td>
<td>Handles the transfer and display of data strings exchanged between the Bluetooth modules</td>
</tr>
<tr>
<td>LCD</td>
<td>.c and .h</td>
<td>LCD interface</td>
</tr>
<tr>
<td>Globals</td>
<td>.c and .h</td>
<td>Contains few global variables used in the code</td>
</tr>
</tbody>
</table>
A.2  Source Code Call Graph

Figure 42 shows the source code call graph.

FIGURE 42: SOURCE CODE CALL GRAPH
APPENDIX B: CONFIGURING THE RN4677 MODULES USING ASCII COMMANDS

Table 4 through Table 6 list the sequence of commands which needs to be used to setup a Bluetooth Gateway communication between BTC and BLE devices, the BTC device being a Data Sending node and the BLE device being a Data Receiving node.

**TABLE 4: BTC NODE**

<table>
<thead>
<tr>
<th>User ASCII Commands</th>
<th>Expected ASCII Responses</th>
</tr>
</thead>
<tbody>
<tr>
<td>$$$</td>
<td>CMD&gt;</td>
</tr>
<tr>
<td>+\r\n</td>
<td>ECHO ON\r\nCMD&gt;</td>
</tr>
<tr>
<td>SG,2\r\n</td>
<td>AOK\r\nCMD&gt;</td>
</tr>
<tr>
<td>SA,2\r\n</td>
<td>AOK\r\nCMD&gt;</td>
</tr>
<tr>
<td>SM,0\r\n</td>
<td>AOK\r\nCMD&gt;</td>
</tr>
<tr>
<td>R,1\r\n</td>
<td>Rebooting\r\n%REBOOT%</td>
</tr>
<tr>
<td>I\r\n</td>
<td>Inquiry, T=8, COD=0\r\n\n</td>
</tr>
<tr>
<td>C,&lt;MAC_Address&gt;\r\n</td>
<td>Trying\r\n%NEW_PAIRING%\r\n\nCONNECT,&lt;MAC_ADDR&gt;,0%/r/n</td>
</tr>
<tr>
<td>K,1\r\n</td>
<td>Disconnected\r\n</td>
</tr>
</tbody>
</table>

**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.

**TABLE 5: GATEWAY NODE (DUAL MODE BT)**

<table>
<thead>
<tr>
<th>User ASCII Commands</th>
<th>Expected ASCII Responses</th>
</tr>
</thead>
<tbody>
<tr>
<td>$$$</td>
<td>CMD&gt;</td>
</tr>
<tr>
<td>+\r\n</td>
<td>ECHO ON\r\nCMD&gt;</td>
</tr>
<tr>
<td>SA,2\r\n</td>
<td>AOK\r\nCMD&gt;</td>
</tr>
<tr>
<td>R,1\r\n</td>
<td>Rebooting\r\n%REBOOT%</td>
</tr>
<tr>
<td>IL\r\n</td>
<td>BLE Scanning\r\n\nList of MAC addresses&gt;\r\nBLE Scan Done\r\nCMD&gt;</td>
</tr>
<tr>
<td>C,0,&lt;MAC_Address&gt;\r\n</td>
<td>%NEW_PAIRING%\r\n\nCONNECT,0,&lt;MAC_ADDR&gt;,2%/r/n\nSESSION_OPEN%</td>
</tr>
<tr>
<td>K,1\r\n</td>
<td>Disconnected\r\n</td>
</tr>
</tbody>
</table>

**TABLE 6: BLE NODE**

<table>
<thead>
<tr>
<th>User ASCII Commands</th>
<th>Expected ASCII Responses</th>
</tr>
</thead>
<tbody>
<tr>
<td>$$$</td>
<td>CMD&gt;</td>
</tr>
<tr>
<td>+\r\n</td>
<td>ECHO ON\r\nCMD&gt;</td>
</tr>
<tr>
<td>SA,2\r\n</td>
<td>AOK\r\nCMD&gt;</td>
</tr>
<tr>
<td>R,1\r\n</td>
<td>Rebooting\r\n%REBOOT%</td>
</tr>
<tr>
<td>K,1\r\n</td>
<td>Disconnected\r\n</td>
</tr>
</tbody>
</table>
Note the following details of the code protection feature on Microchip devices:

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

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

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

Trademarks

The Microchip name and logo, the Microchip logo, AnyRate, dsPIC, FlashFlex, flexPWR, Heldo, JukeBlox, KeeLoq, KeeLoq logo, Kleer, LANCheck, LINK MD, MediaLB, MOST, MOST logo, MPLAB, OptoLyzer, PIC, PICSTART, PIC32 logo, RightTouch, SpyNIC, SST, SST Logo, SuperFlash and Uni/O are registered trademarks of Microchip Technology Incorporated in the U.S.A. and other countries.

ClockWorks, The Embedded Control Solutions Company, ETHERSYNCH, Hyper Speed Control, HyperLight Load, IntelliMOS, mTouch, Precision Edge, and QUIET-WIRE are registered trademarks of Microchip Technology Incorporated in the U.S.A.


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

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

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

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

© 2016, Microchip Technology Incorporated, Printed in the U.S.A., All Rights Reserved.

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: 61-2-9888-6733
Fax: 61-2-9838-6755

China - Beijing
Tel: 86-10-8569-7000
Fax: 86-10-8528-2104

China - Chongqing
Tel: 86-23-8960-9588
Fax: 86-23-8960-9500

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

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