



Bluetooth SPP-only Demonstration for the PIC32MX270F256D Plug-in Module (PIM) Demonstration ReadMe

1.1 DESCRIPTION

Note: This ReadMe provides detailed demonstration information. For additional information, please consult the companion document, “*PIC32 Bluetooth Audio Development Kit Reference Guide*” (DS70005140). This reference guide is available for download from www.microchip.com.

The Bluetooth SPP-only demonstration runs on the PIC32MX270F256D PIM (P/N: MA320013) mounted on the 100-pin socket of the PIC32 Bluetooth Audio Development Kit (P/N: DV320032) as a sink. The source could be an Android or Windows phone and a Bluetooth-enabled PC. A serial port terminal emulator is needed on source devices to run this demonstration (see [Section 1.4 “Demonstration Setup”](#) for details).

During the demonstration, commands can be sent from or to the source or sink. Any command sent from the source, loops back and is displayed on the terminal emulator of the source. Any button (SW1-SW5) pressed on the board in the PIC32 Bluetooth Audio Development Kit appears on the terminal emulator of the source. Table 4 provides details of the available commands for the demonstration. More commands can be added and processed using the `btapp_sppcmd.c` file. Commands received from the source are saved in `sppRxBuffer.cmdBuffer[][]` with the exception of when the source is a PC. For such instances, commands are saved in `cmdFromPC.cmdBuffer[]`. See [Section 1.5 “Running the Bluetooth SPP-only Demonstration”](#) for command and command processing information.

The PIM is based upon the PIC32MX270F256D, which is a 44-pin low-end device, with 256 KB Flash and 64 KB SRAM and an operating speed of 50 MHz. The device can also operate at a lower frequency of 40 MHz with a higher temperature limit. For information on this device, refer to the “*PIC32MX1XX/2XX Data Sheet*” (DS60001168), which is available from www.microchip.com.

- Note 1:** The PIC32 Bluetooth Audio Development Kit (DV320032) is sold separately. Contact your local Microchip Sales Office or visit www.microchip.com.
- 2:** The display of the PIC32 Bluetooth Audio Development Kit cannot be connected to the PIC32MX270F256D PIM.

Bluetooth SPP-only PIC32MX270F256D PIM Demonstration

1.2 REQUIREMENTS

1.2.1 Hardware and Software Resources

The following Microchip hardware and software are required to run this demonstration:

- PIC32MX270F256D Plug-in Module (PIM) - P/N: [MA320013](#)
- PIC32 Bluetooth Audio Development Kit - P/N: [DV320032](#)
- In-circuit Debugger (one of the following):
 - PICKit™ 3 In-circuit Debugger - P/N: [PG164130](#)
 - MPLAB® REAL ICE™ Probe Kit - P/N: [DV244005](#)
- MPLAB X IDE v2.10 or later (available from www.microchip.com/mplabx)
- MPLAB XC32 C/C++ Compiler for PIC32 MCUs (available from www.microchip.com/xc32)

In addition, the following hardware is required:

- 9V DC Power Supply
- Smartphone (Android or Windows)
- Bluetooth-enabled PC (see **Note**)

Note: A Bluetooth-enabled PC is required only when using the PC as the source. For programming the device, a PC is required. Alternatively, if the PC is not Bluetooth-enabled, a Bluetooth-enabled USB dongle can be purchased and connected to the USB port of the PC.

TABLE 1-1: RESOURCE REQUIREMENTS

Resource	Requirement
Program Memory	94 Kbytes
Data Memory	3.5 Kbytes
MIPS	8 estimated

1.2.2 Demonstration Code

The latest Bluetooth SPP-only demonstration, as well as application source code, can be obtained from Microchip free-of-charge by visiting www.microchip.com/pic32btsuites. Please refer to “PIC32 Bluetooth Audio Development Kit Reference Guide” (DS70005140) for additional information to set up and run the demonstration code on the development board. This document is available for download from www.microchip.com.

- Note 1:** The Bluetooth Stack for Serial Port Profile (SPP) is available free-of-charge in binary form only.
- 2:** An audio demonstration is planned for a future release of the demonstration.

The configuration file, `user_config.h`, contains some of the programmable parameters such as the name of the device (`_BT_AUDIO_CONNECT_NAME`, `_BT_AUDIO_DISPLAY_NAME`), etc.

1.3 BASIC FUNCTIONALITY

Bluetooth data support only Bluetooth Stack with SPP profile. In this demonstration, the basic Bluetooth audio profiles and decoders have been removed, (i.e., A2DP, AVRCP, SBC and/or AAC). This demonstration provides basic data transport of non-audio full-duplex data transfers over the Bluetooth link. This demonstration does not provide any USB audio support. The demonstration allows a user to perform terminal emulation and echo characters from an Android/Windows smartphone or PC over a Bluetooth connection and then back to the PC or smartphone emulation application menu.

D5-D9 led (LED1 - LED5) from Development Board (DV320032) are connected to I²C bus of the PIM MA320013 through MCP23008 expander. The address of the expander can be configured to 0x22 or 0x20 through the jumper J1.

TABLE 1-2: SWITCHES AND LEDS

Component	Label	Bluetooth Mode
Switch	SW1	Button 1.
	SW2	Button 2.
	SW3	Button 3.
	SW4	Button 4.
	SW5	Button 5.
LED	D5	Indicates the device is powered on and code is running.
	D6	Indicates the device is paired and connected with the master device.
	D7	This LED can be controlled through user commands from the master device.
	D8	This LED is on when a receive (RX) is in progress.
	D9	This LED is on when a transmit (TX) is in progress.

1.3.1 Bluetooth Module

The PIC32 Bluetooth Audio Development Kit provides hardware support for the BlueCore® CSR8811™ and the RDA Microelectronics RDA5876 through compile-time switches.

1.3.1.1 CSR8811

The CSR8811 is a single-chip radio and baseband IC for Bluetooth 2.4 GHz systems including Enhanced Data Rate (EDR) to 3 Mbps and Bluetooth low energy. The CSR8811 supports Bluetooth Class 1 transmission, and supports multiple device connection. The PIC32 Bluetooth Audio Development Kit uses a module based on the CSR8811 radio in its default configuration (see **Note**). Software projects using the default board configuration should select the CSR8811 configuration in MPLAB X IDE.

Note: The CSR8811 daughter board is included in the PIC32 Bluetooth Audio Development Kit.

1.3.1.2 RDA5876

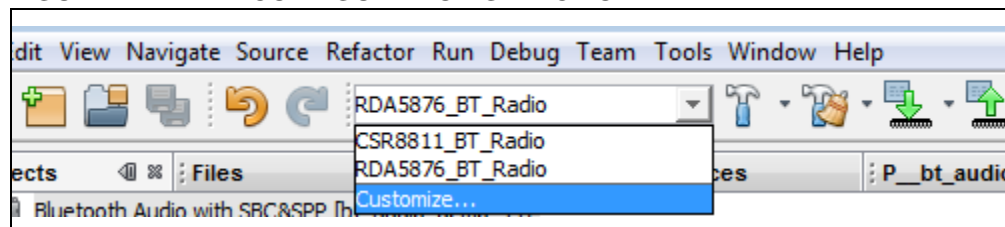
The RDA5876, integrates Bluetooth and a FM radio tuner into one device and is optimized for mobile applications. Bluetooth and FM can work simultaneously and independently, with low-power consumption levels targeted to battery powered devices. For the highest integration level, the required board space has been minimized and customer cost has been reduced. The RDA5876 meets Class 2 and Class 3 transmitting power requirements. The RDA5876 radio solution is a low-cost alternative for single point Bluetooth audio and data applications.

Note: To request an RDA5876 daughter board, please contact your local Microchip sales office.

1.3.1.3 SELECTING THE CONFIGURATION

A compile-time switch in the application code provides software support for either the CSR8811 or the RDA5876. This can be done by selecting the desired configuration during compilation, as shown in the following figure.

FIGURE 1-1: CONFIGURATION SELECTION



1.3.2 Connections

By default, if the Bluetooth device is connected and paired but the user walks out of Bluetooth range, the system will continuously search for the last Bluetooth connection source (see [Section 1.3.6 “Connection Retry Time Limit”](#)). Then, when and if, the Bluetooth-enabled device comes back into range, it will automatically reconnect. If the smartphone or Bluetooth-enabled device, while still in range, disables its Bluetooth connectivity, the Bluetooth Audio Development Board Bluetooth software will not attempt to automatically reconnect with the device.

1.3.3 Bluetooth Device IDs

The Bluetooth software remembers and stores in Flash memory the last 10 unique Bluetooth device IDs to which it successfully paired to facilitate faster automatic reconnection when there is no currently active Bluetooth connection. If desired, the maximum number of devices to remember and store can be configured using `BTMGR_MAX_LINK_KEYS` in `user_config.h`. If Bluetooth is turned OFF on a user smartphone that is currently connected and re-enabled later, it will automatically reconnect if in range or when it comes back into range. In addition, when the Bluetooth Audio Development Board is powered on, the Bluetooth software will automatically pair and connect to the last Bluetooth-enabled device, assuming it is still active; otherwise, it will search for the next most recently connected device in the list and repeat.

1.3.4 Bluetooth Pair/Unpair

If the user presses and holds SW1, which forces Bluetooth to unpair, the user must manually force their smartphone to “forget” the Bluetooth demonstration name of the development kit to enable their smartphone to rediscover and subsequently re-pair with the development kit. If the user selects SW2 (Bluetooth disconnect), the user does not need to force their smartphone to forget the demonstration name of the development kit and can reconnect at will.

- Note:** This demonstration does not support data transfer from an Apple iPhone application. To do so requires the following:
- The iAP software layer addition for the Bluetooth Stack
 - Special Apple controlled hardware. Apple controlled hardware is only available through the Apple MFi program. Please contact applesupport@microchip.com for more information.
 - At the time of publication, only one active Bluetooth connection at a time is supported; however, multi-connect features are being examined.

1.3.5 Bluetooth Device Address

By default, when the development kit is powered on for the first time, it generates a one-time random unique Bluetooth Device Address for any given development kit hardware. Optionally, at design time, the user can specify a Bluetooth Device Address in the application code of the development kit.

The device address is a six byte hexadecimal value. The macro, `BT_DEVICE_ID`, defines the first four bytes of the hexadecimal value and `BT_DEVICE_ID_2LSB` defines the last two bytes of the hexadecimal value. The last two bytes of the hexadecimal value can be randomized by enabling `BT_DEVICE_ID_2LSB_RANDOMIZE`. These macros are defined in `HardwareProfile_PIC32_Bluetooth_Audio_Development_Board.h`.

Setting a specific hard-coded device address is not recommended during the design and development state, as Bluetooth connection problems may be experienced if another development board with the same Bluetooth Device Address is within range.

1.3.6 Connection Retry Time Limit

Starting with demonstration v2.0, a new feature enables a connection retry time limit. The limit will define a set period (in approximate seconds) that the unit will continue to retry to connect to the Bluetooth device from which it has lost a connection. After this period, the device will discontinue trying to automatically connect. However, the device can still manually establish a previously paired connection, or form a new pair as previously stated. The feature can be enabled in the `user_config.h` file.

1.4 DEMONSTRATION SETUP

This demonstration can be run on either an Android or Windows smartphone, and a Bluetooth-enabled PC. To run this demonstration a Serial Port Profile (SPP) emulator is required on the source (master device). The following sections provide information on obtaining an SPP emulator.

1.4.1 Android Phone

The free SPP emulator application for Android smart phones, **Bluetooth spp tools pro**, is available for download from the Google Play store by visiting:

https://play.google.com/store/apps/details?id=mobi.dzs.android.BLE_SPP_PRO&hl=en

1.4.2 Windows Phone

The free Windows smartphone application, **BT Terminal**, is available for download by visiting:

<http://www.windowsphone.com/en-us/store/app/bt-terminal/09d679af-bdd8-40b2-b54e-56d68aeb03e0>

1.4.3 PC

A terminal emulator is needed to run this demonstration on the Bluetooth-enabled PC. A free terminal emulator, **PuTTY**, is available for download by visiting:

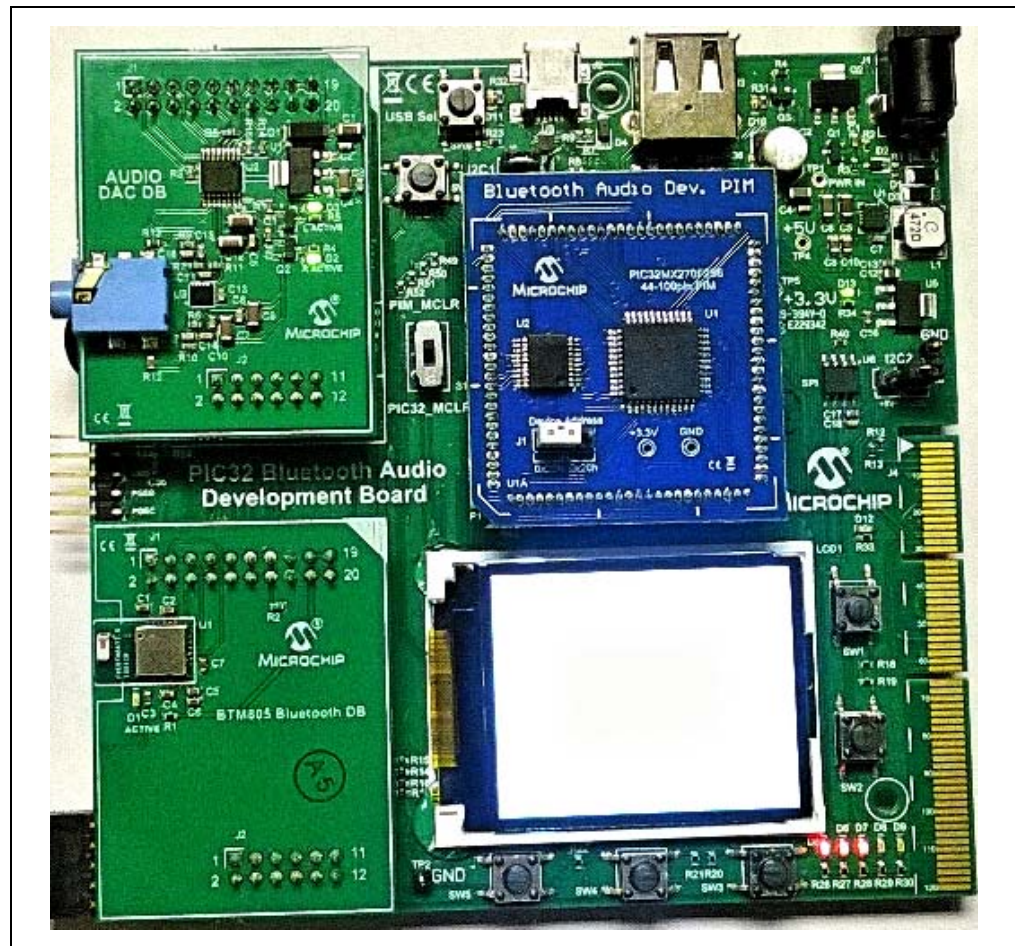
<http://www.chiark.greenend.org.uk/~sgtatham/putty/download.html>

1.5 RUNNING THE BLUETOOTH SPP-ONLY DEMONSTRATION

Note: Please refer to “PIC32 Bluetooth Audio Development Kit Reference Guide” (DS70005140) before attempting to run this demonstration.

Once the board is set up correctly, you are ready to run the Bluetooth SPP-only demonstration on the PIC32MX270F256D PIM. Be sure to mount the PIM on the development board, as shown in figure, and set switch S1 to the PIM_MCLR position (away from the display).

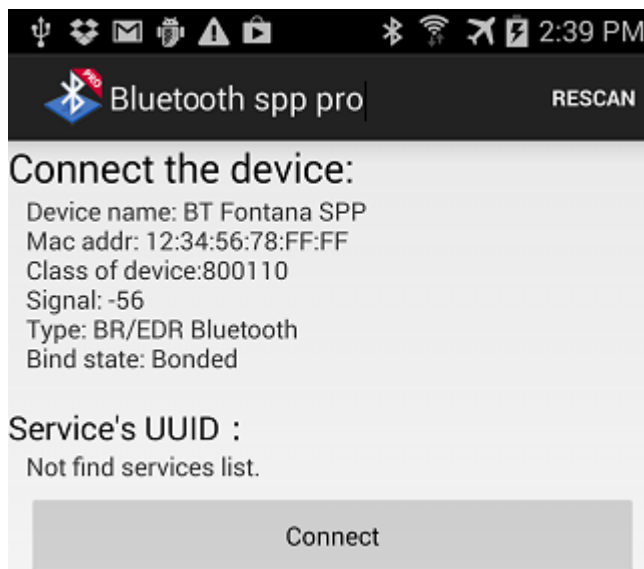
FIGURE 1-2: BOARD SETUP



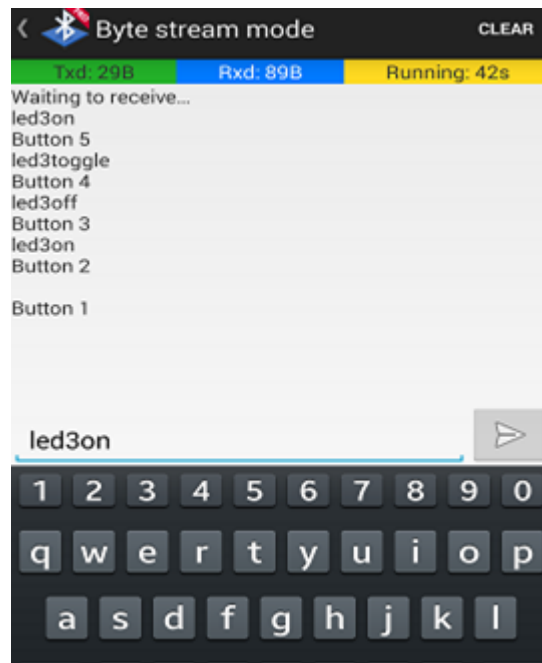
Bluetooth SPP-only PIC32MX270F256D PIM Demonstration

1.5.1 Running the Demonstration on an Android Smartphone

1. Program the PIM device with the hex file BT_SPPOnly_MA320013Pim.hex.
2. Enable Bluetooth on the smartphone and pair with BT MA320013 SPP.
3. Open the Android terminal emulator application, Bluetooth spp pro.



4. Connect the smartphone with the development kit to the device named BT MA320013 SPP.
5. If prompted by your Bluetooth device for a PIN, enter 0000. The device should connect and pair.
6. After pairing, the smartphone must be connected to the development kit. Please note that some terminal emulator applications do this automatically upon pairing. Once the smartphone is connected, LED D5 and LED D6 should illuminate.
7. Enter characters in the smartphone applications under "CMD Line Mode", "Keyboard Mode", or "Byte Stream Mode". The characters will be echoed back by the PIC32 device and displayed on the smartphone menu. When SW1 through SW5 are pressed on the development kit a message is sent to the smartphone. For example, "Button 1" will appear on the smartphone when SW1 is pressed. The same applies for SW2 through SW5. Pressing SW2 echoes "Button 2" and so on.



8. Send a command from terminal emulator application on the smartphone. The same command is echoed back to the terminal emulator application. [Table 1-3](#) lists the available commands.

TABLE 1-3: COMMANDS

Command	Destination	Action
led3on	Sink	Turn on LED D7
led3off	Sink	Turn off LED D7
led3toggle	Sink	Toggle LED D7
SW1	Source	Button 1
SW2	Source	Button 2
SW3	Source	Button 3
SW4	Source	Button 4
SW5	Source	Button 5

9. The command received from the master device is saved in `sppRxBuffer` (`btapp_sppcmd.c`). The buffer size and the number of commands can be changed by modifying the `CMDLENGTH` and `NUMOFCMD` macros in `btapp_sppcmd.h`.

1.5.2 Running the Demonstration on a Windows Smartphone

Note: In this demonstration, a Lumia Windows smartphone was used.

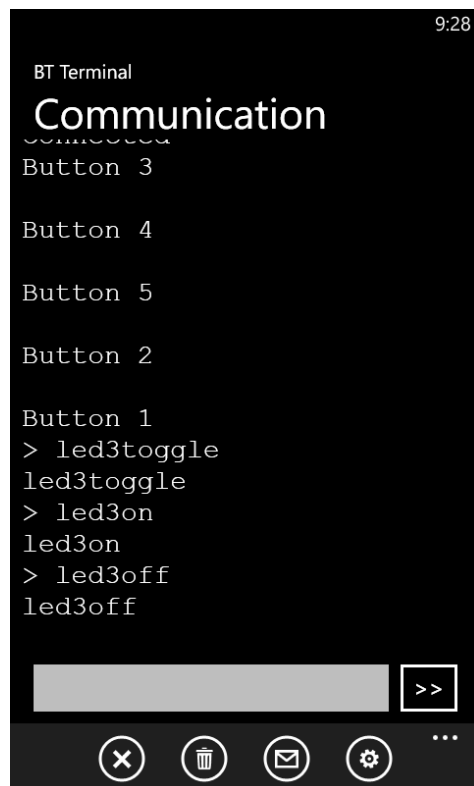
1. Program the PIM device with the hex file `BT_SPPOnly_MA320013Pim.hex`.
2. Enable Bluetooth on the smartphone and pair with BT MA320013 SPP.
3. Open the Windows terminal emulator, BT Terminal.



4. Connect the smartphone with the development kit to the device named BT MA320013 SPP.
5. If prompted by your Bluetooth device for a PIN, enter 0000. The device should connect and pair.

Bluetooth SPP-only PIC32MX270F256D PIM Demonstration

6. After pairing, the smartphone must be connected to the development kit. Please note that some terminal emulator applications do this automatically upon pairing. Once the smartphone is connected, LED D5 and LED D6 illuminate.
7. Enter characters in the smartphone applications. The characters will be echoed back by the PIC32 device and displayed on the smartphone menu. When SW1 through SW5 are pressed on the development kit a message is sent to the smartphone. For example, "Button 1" will appear on the smartphone when SW1 is pressed. The same applies for SW2 through SW5. Pressing SW2 echoes "Button 2" and so on.

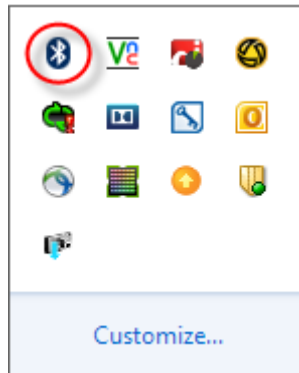


8. Send a command from the BT terminal application. The same command is echoed back to the BT terminal. [Table 1-3](#) shows the commands.

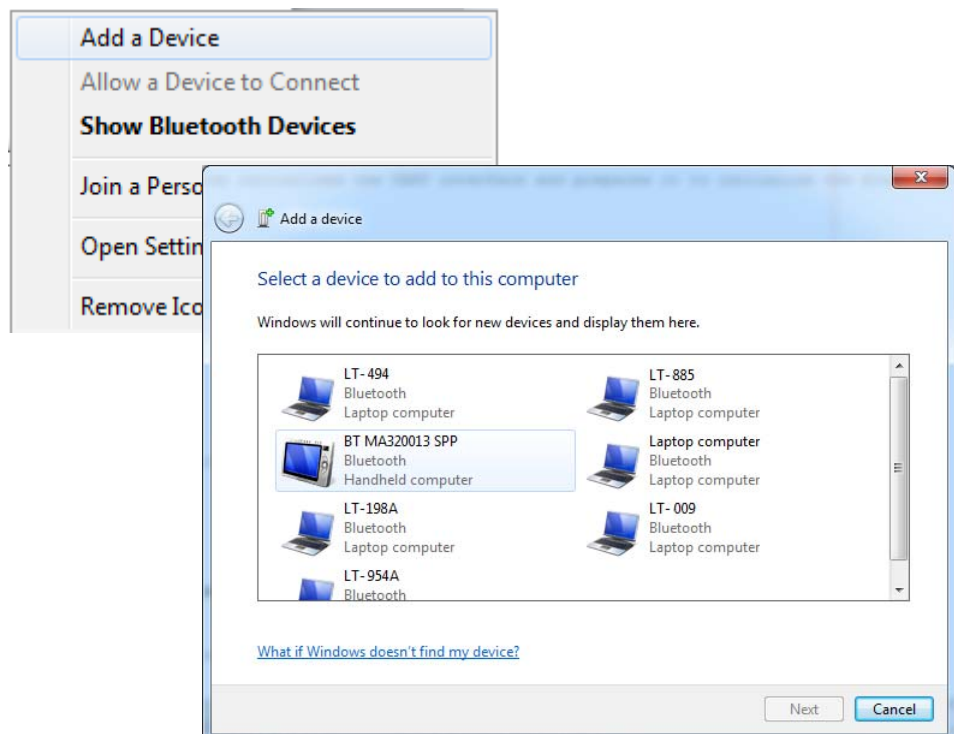
Note: The Lumia Windows 8 smartphone sends '`\r\n`' as a terminating character for a carriage return. Therefore, care must be taken to remove it and insert '`\0`' for the command processor to work.

1.5.3 Using a Bluetooth-enabled PC With the Demonstration

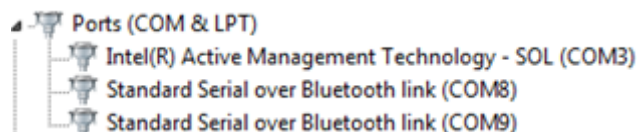
1. Program the PIM device with the hex file BT_SPPOnly_MA320013Pim.hex.
2. If not visible in your system tray, open the list of hidden icons and click the Bluetooth icon, as shown in the following figure.



3. Click **Add a Device**, and from the list of discovered devices, select BT MA320013 SPP, as shown in the following figure and then click **Next**.

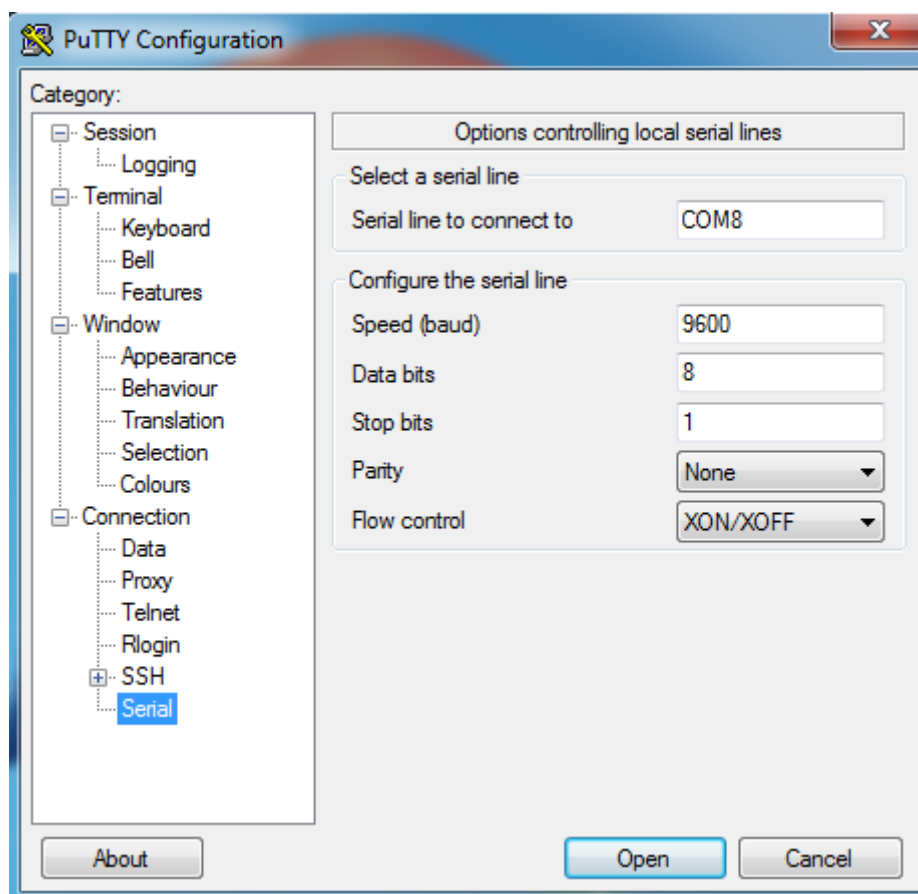


4. A configuring Bluetooth popup window appears and creates two serial ports.
5. Open the Device Manager to determine the ports that were created (COM8 and COM9), as shown in the following figure.

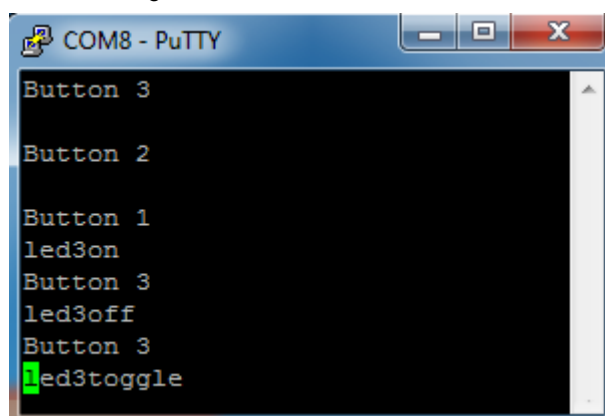


Bluetooth SPP-only PIC32MX270F256D PIM Demonstration

6. Open the PuTTY or any terminal emulator and configure it as shown in the following figure. The Baud rate can be either 9600 or 115200.



7. Enter characters in the terminal emulator application. The characters will be echoed back by the PIC32 device and displayed in the terminal emulator window. When SW1 through SW5 are pressed on the development kit a message is sent to the terminal emulator. For example, "Button 1" will appear when SW1 is pressed. The same applies for SW2 through SW5. Pressing SW2 echoes "Button 2" and so on.



8. Send a command from the terminal emulator. The same command is echoed back to the terminal emulator window. The available commands are listed in [Table 1-3](#).

Note: The PC sends one character at a time followed by the terminating character '\r' until the Enter key is pressed (i.e., carriage return, 0xd). These characters are stored in a buffer until '\r' is detected, which makes one command ('\r' is replaced by '\0' in the buffer for processing this command). This command is compared with known commands and if they match, the appropriate action is taken.

1.6 KNOWN ISSUES

The following issue is known to exist in this demonstration:

After programming the PIC32MX270F256D PIM with the RD BT module, the board will not be discoverable until its power has been turned OFF and then ON.

1.7 REVISION HISTORY

Version 1.0.0

This is the initial version of the demonstration.

Bluetooth SPP-only PIC32MX270F256D PIM Demonstration

NOTES: