Bluetooth Data Demonstration 4 with SPP 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.

Bluetooth data support only. Bluetooth Stack with SPP profile.

> **Note:** This demonstration *does not* support data transfer from an Apple iPhone application.

Refer to Table 3-1 in **Chapter 3. "Interoperability Testing Results"** of the *"PIC32 Bluetooth Audio Development Kit Reference Guide"* (DS70005140) for the list of tested Bluetooth-enabled devices.

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 smartphone or PC over a Bluetooth connection to the graphic display of the development board, and then back to the PC or smartphone emulation application menu.

1.2 BASIC FUNCTIONALITY

1.2.1 Bluetooth Module

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

1.2.1.1 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.

1.2.1.2 RDA5876

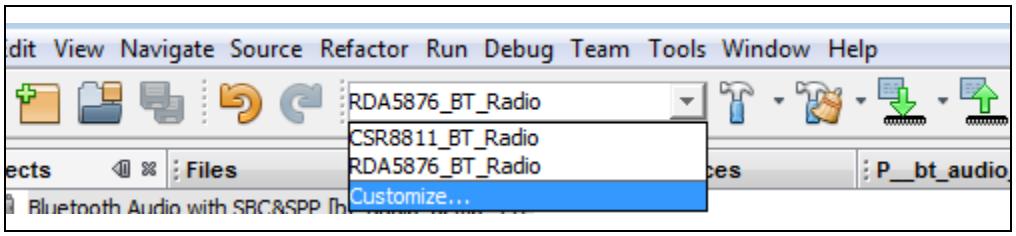
The RDA5876, integrates Bluetooth and a FM radio tuner into one device and is optimized for mobile applications. Bluetooth and FM can work simultaneously and independently, with low-power consumption levels targeted to battery powered devices. For the highest integration level, the required board space 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.

1.2.1.3 SELECTING THE CONFIGURATION

A compile-time switch in the application code provides software support for the 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



1.2.2 Connections

If the Bluetooth device is connected and paired but the user walks out of Bluetooth range, by default, the system will continuously search for the last Bluetooth connection source (see **Section 1.2.6 "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 it's Bluetooth connectivity, the Bluetooth Audio Development Board Bluetooth software will not attempt to automatically reconnect with the device.

1.2.3 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 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.

1.2.4 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.

1.2.5 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 4 bytes of the hexadecimal value and `BT_DEVICE_ID_2LSB` defines the last 2 bytes of the hexadecimal value. The last two bytes of the device address 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.

1.2.6 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.

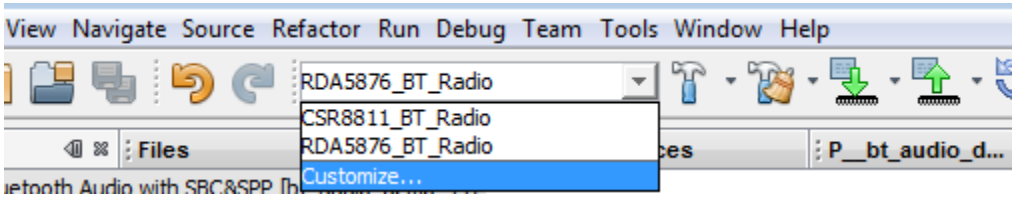
1.3 BLUETOOTH DATA SMARTPHONE DEMONSTRATION SETUP

Download a terminal emulator echo application. An Android application can be obtained by visiting:

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

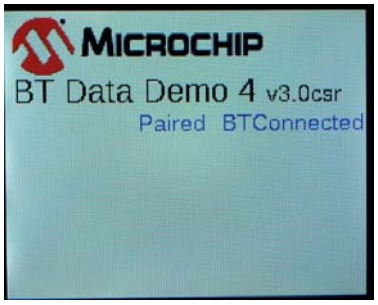
1.4 RUNNING BLUETOOTH DATA SMARTPHONE DEMONSTRATION 4

1. Program the device with the hex file, `BT Data Demo 4.hex` for the CSR8811_BT_RADIO or RDA5876_BT_RADIO Bluetooth controller. This can be done by selecting the hardware support from the drop down box during compilation, as shown in the following figure.



2. Enable Bluetooth on the smartphone.
3. Open the Android terminal emulator application.
4. Connect and pair the smartphone with the Bluetooth Audio Development Board to the device named **BT Data Demo 4**.
5. If prompted by your Bluetooth device for a PIN, enter `0000`. The device should connect and pair, and indicate as such on the display, as shown in the following figure.

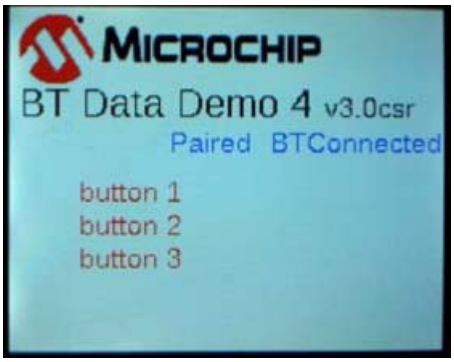
6. After pairing, the smartphone must be connected to the development kit. Please note that some terminal emulator applications do this automatically upon pairing. Once the smartphone is connected, the display updates from “Not Connected” to “BT Connected” (see the following figure), and LED D5 illuminates.



7. Enter characters in the smartphone applications under “CMD Line Mode”, “Keyboard Mode”, or “Byte Stream Mode”. The characters will be displayed on the LCD of the development board, and are then 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, “Button 1” will appear on the hand set when SW1 is pressed. The same applies for when “SWx” (x = 1-5) is pressed, the corresponding “Button x” (x = 1-5) will appear on the smartphone (see the following figure).

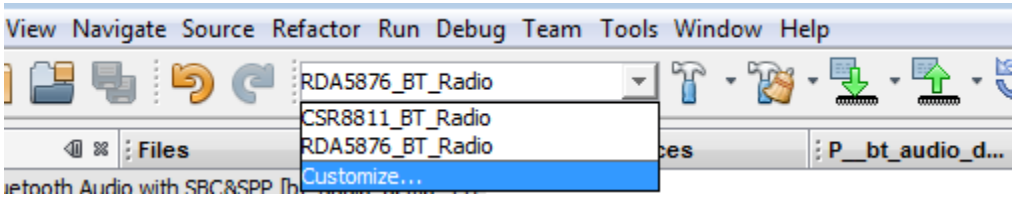
In Keyboard Mode, phases can be saved to buttons (see the previous figure). Once the button is pressed or a string is entered, it will appear on the display of the development kit, as shown in the following figure.



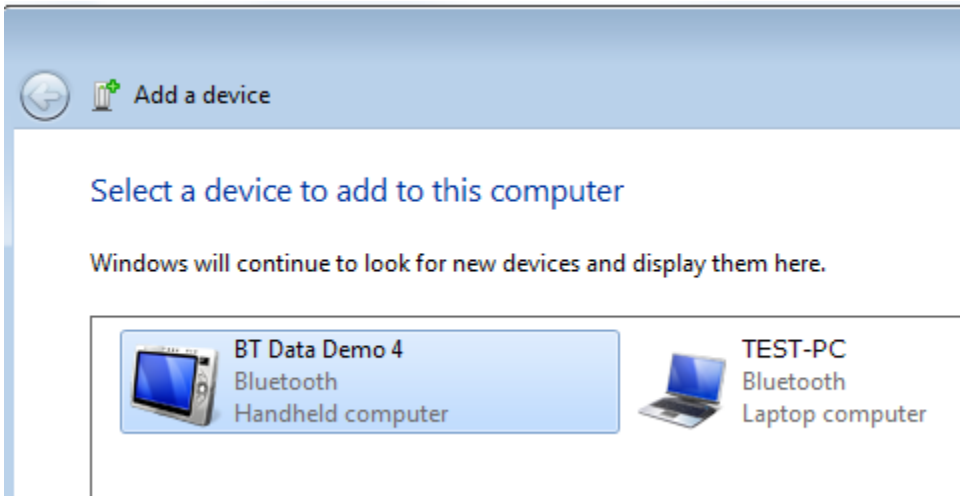
The received text will “round robin” on the bottom three lines. In this example, the next string sent takes the place of Button 1, which is moved down one line until it repeats and starts from the top again.

1.5 BLUETOOTH DATA PC DEMONSTRATION SETUP

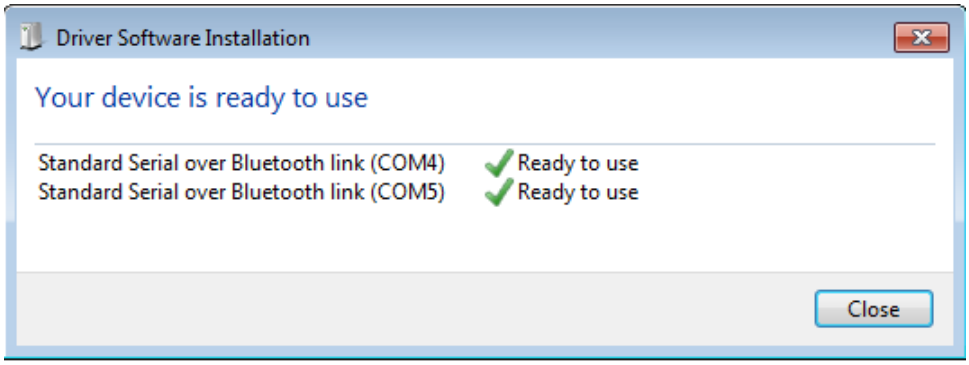
1. Program the device with the hex file, BT Data Demo 4.hex for the CSR8811_BT_RADIO or RDA5876_BT_RADIO Bluetooth controller. This can be done by selecting the hardware support from the drop down box during compilation, as shown in the following figure.



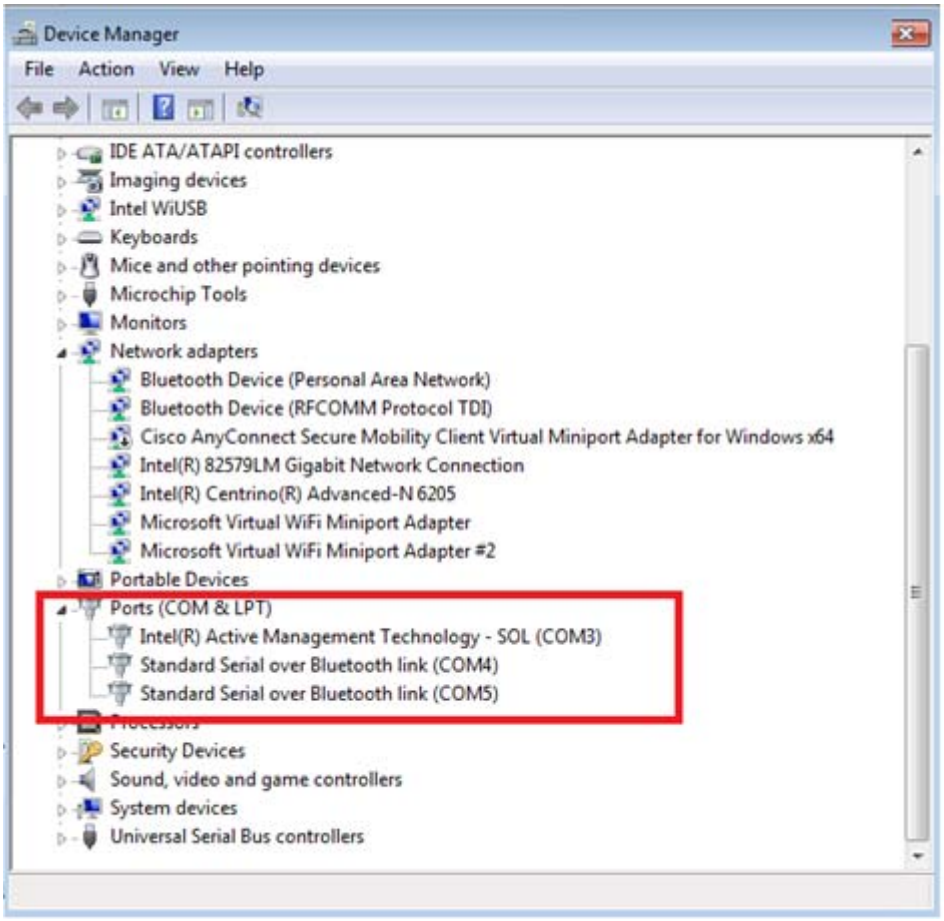
2. On the PC, select *Start > Control Panel > Hardware and Sounds >Add a device*. A dialog appears with a list of available bluetooth devices. From the list, select **BT Data Demo 4**.



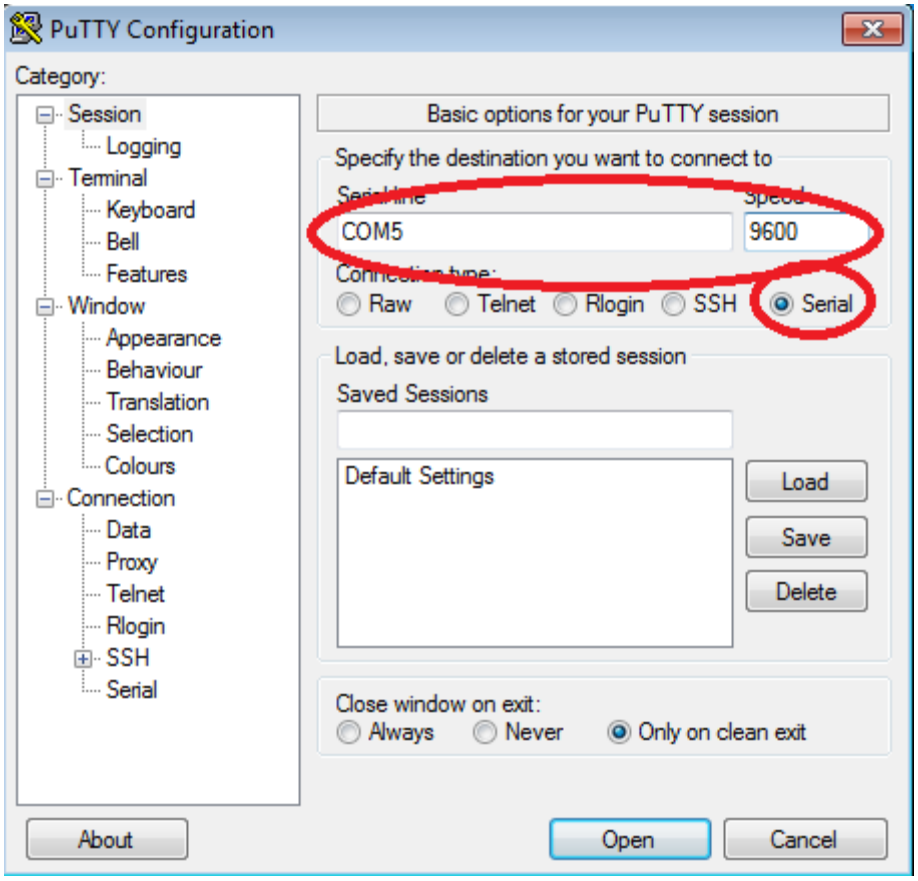
3. Once BT Data Demo 4 is selected, the PC will install the drivers and will allocate COM ports. When driver installations are complete, a pop-up message appears indicating the device is ready for use. The allocated COM ports vary, depending on the PC.



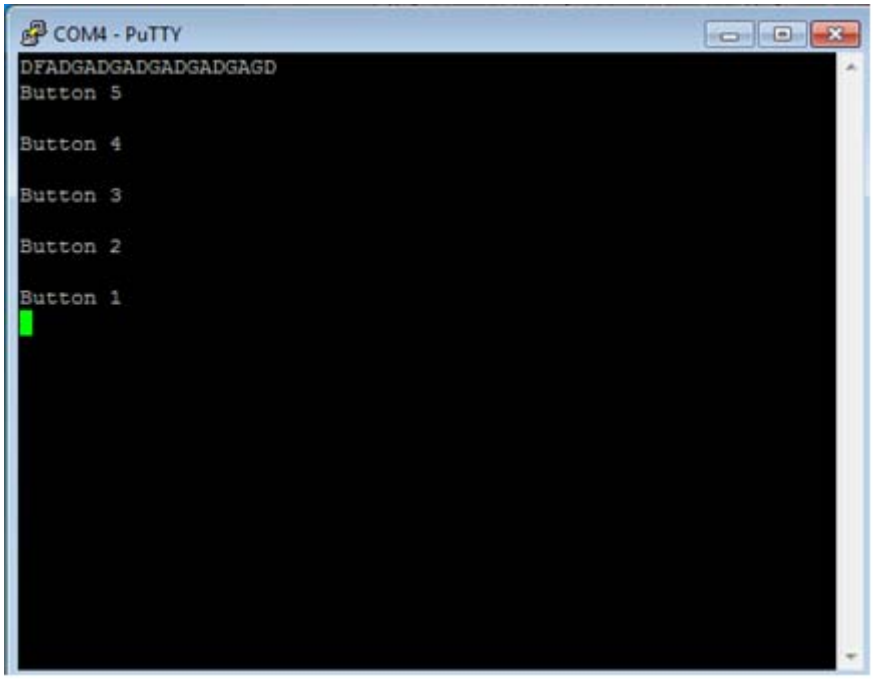
4. The COM ports used can also be identified from the Device Manager.



5. Open a terminal emulator. For this example, PuTTY was used. This PC application can be obtained by visiting: http://www.chiark.greenend.org.uk/~sgtatham/putty/download.html.



6. Once the terminal connection is established, a new terminal window opens and the transmitted and received data will be displayed, as shown in the following figure.



7. The transmitted data is displayed, as shown in the following figure.

TABLE 1-1: BLUETOOTH SPP DEMONSTRATION 4 CONTROLS

Component	Label	Bluetooth Mode
Switch	SW1	Button 1
	SW2	Button 2
	SW3	Button 3
	SW4	Button 4
	SW5	Button 5
LED	D5	ON (Bluetooth Connected)
	D6	ON
	D7	N/A
	D8	Message Send/Receive
	D9	Message Send/Receive

1.6 BLUETOOTH DEMONSTRATION 4 SWITCH DESCRIPTIONS

1.6.1 Buttons

When SW1 through SW5 are pressed, a message is sent to the smartphone. The string “Button 1” will appear on the handset when SW1 is pressed. The same applies when SW1 through SW5 are pressed, the corresponding Button *x*, where *x* = 1 through 5, respectively, will appear on the smartphone or PC terminal.

1.6.2 LEDs

When D5 illuminates, this indicates a Bluetooth device is connected and paired. If D5 is off, no device is connected to the development kit. When D8 and D9 are illuminated, this indicates a SPP message was sent or received.

1.7 REVISION HISTORY

Version1.0.1

This revision includes the following updates:

- Updates to RTOS files
- XC32 v1.3 complier compatibility improvements

Version 1.0.2

This revision includes the following updates:

- Missing library components were added to enable compilation

Version 2.0

This revision includes the following updates:

- RTOS was removed
- The Graphics Driver was updated to support future displays
- A top-level user configuration header file (`user_config.h`), which includes global constraints was added

Version 3.0

This revision includes the following updates:

- RDA Bluetooth module support was added
- A global definition for Bluetooth Device ID value (CSR only) was added
- The common display design defined by `display_config.h` was added
- The application architecture was updated to enable compatibility with other demonstrations

Version 3.0.1

This revision includes the following updates:

- The assert diagnostics were revised, which are now user configurable (default is OFF)
- Stack usage diagnostics were added (default is OFF)
- Options in the `user_config.h` file were revised
- The Bluetooth SPP Library (1.3.1 - 1.5.1) was revised