



PIC32 Bluetooth® Starter Kit Demonstration ReadMe

1.1 DESCRIPTION

Note: This ReadMe provides detailed demonstration information. For additional information, please consult the companion document, “PIC32 Bluetooth® Starter Kit User’s Guide” (DS70005190). This reference guide is available for download from www.microchip.com.

This demonstration performs full-duplex data transfer over the Bluetooth link.

The demonstration allows the user to send and receive SPP data with an Android OS-based smartphone using Microchip's PIC32 Bluetooth Starter Kit application.

The PIC32 Bluetooth Starter Kit Android Application allows users to send and receive SPP data with an Android OS-based smartphone to Microchip's PIC32 Bluetooth Starter Kit. The application file, BTSK_Android_App.apk, is contained in the PIC32 Bluetooth Starter Kit Quick Start.ZIP archive file. This archive file is available for download from: www.microchip.com/pic32btsk.

For further development of the PIC32 or Android applications, the SDK is located in the <install-dir>/apps/bluetooth/data_temp_sens_rgb folder within your installation of MPLAB® Harmony, which includes embedded source, Android source, and the Bluetooth binary library. MPLAB Harmony is available for download from: www.microchip.com/harmony.

Note: This demonstration does not support data transfer from an Apple® iPhone application. Using the demonstration with a non-Android OS is possible; however, it is not supported by the APK. See [Section 1.5 “Additional Bluetooth Resources”](#) for alternative OS support.

1.2 BASIC FUNCTIONALITY

1.2.1 Bluetooth Module

The PIC32 Bluetooth Starter Kit provides hardware support for the BlueCore® CSR8811™.

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 Starter 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 Flairmicro BTM805 module using the CSR8811 device is integrated in the PIC32 Bluetooth Starter Kit.

1.2.2 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 1.2.5 "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.

1.2.3 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.

Note: Currently, the demonstration does not have support to SPI Flash memory due to limitations in MPLAB Harmony, and therefore, the pairing information will not be stored or recovered on a power or hardware reset.
--

1.2.4 Bluetooth Device Address

When the starter kit is powered on, it generates a random unique Bluetooth Device Address for any given starter 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 device address can be randomized by enabling `BT_DEVICE_ID_2LSB_RANDOMIZE`. These macros are defined in `btconfig.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.

1.2.5 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.

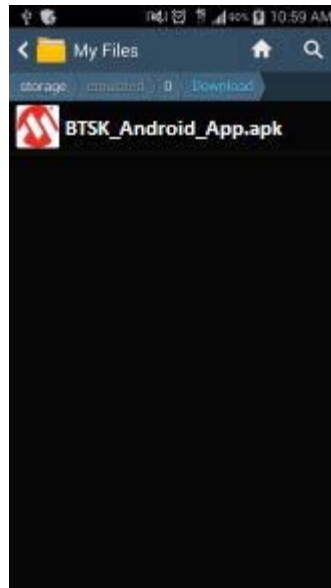
1.3 DEMONSTRATION SETUP

This demonstration allows the SPP data transfer/receive of temperature sensor and RGB data. Before running the demonstration, it is necessary to install the PIC32 Bluetooth Starter Kit Android application to your Android v4.0 or later smartphone.

If an Android device is not available, refer to [Section 1.5 “Additional Bluetooth Resources”](#) to run the demonstration from a PC.

1.3.1 Installing the PIC32 Bluetooth Starter Kit Android Application

1. Install the Android application, `BTSK_Android_App.apk`, to your Android 4.0 or later smartphone.
2. Connect the Android device to a computer using a mini-B USB connector.
3. It is suggested to copy the Android application into the Download folder of the Android device.
4. On the Android device, select *My Files > All Files > Download > BTSK_Android_App.apk*.



5. After selecting the .apk file, the warning message, "blocking installation", will appear.

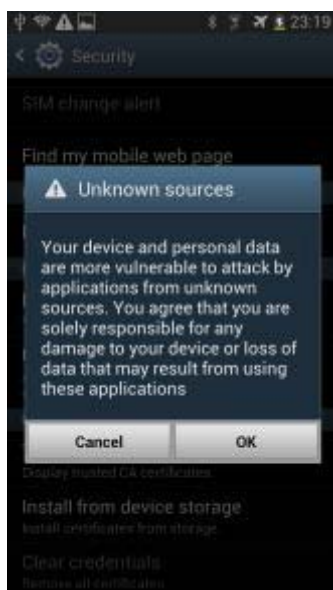


PIC32 Bluetooth Starter Kit SPP Demonstration

6. Select **Settings** and the security window will appear.
7. Choose the option to install applications from unknown sources.

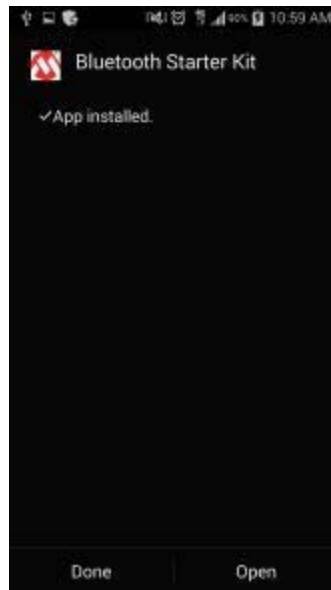


8. Once the option is selected a warning will appear, as shown in the following figure.



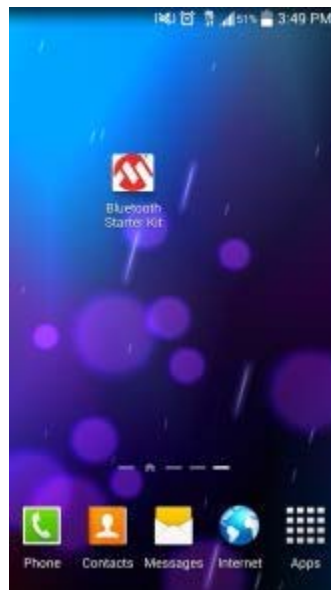
9. After selecting **OK** a window will appear requesting confirmation for installing the application.

10. Once the installation is complete, select **Open** to run the application, as shown in the following figure.



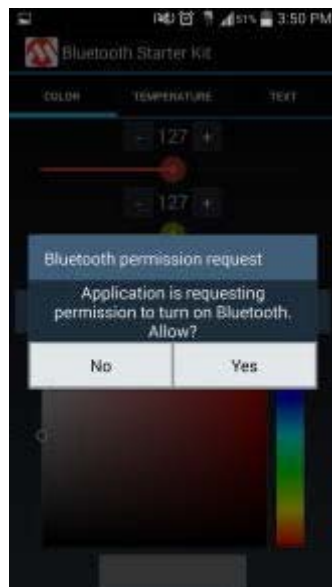
1.4 RUNNING THE PIC32 BLUETOOTH STARTER KIT DEMONSTRATION

1. Select the Bluetooth Starter Kit icon in the application window.



PIC32 Bluetooth Starter Kit SPP Demonstration

2. If prompted, turn on Bluetooth by selecting **Yes**.



3. There are three methods for performing the next steps depending on the phone and Android version you are using. **[A]** Pressing the Microchip logo and the words "Bluetooth Starter Kit", **[B]** pressing the on-screen Menu button [B] (if supported), or **[C]** Pressing the Menu button (hardware). In general, if you have a hardware button you will not have an on-screen button and vice versa. After opening the menu, select "Connect a device - Insecure".



Method [A]

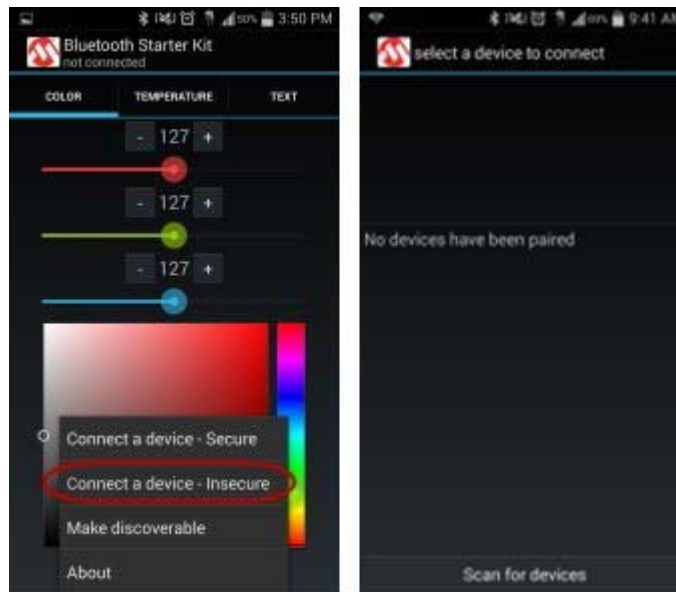


Method [B]



Method [C]

4. Select the option "Connect a device - Insecure" and a window will appear showing a list of paired devices and an option to scan for devices, as shown in the following figure.

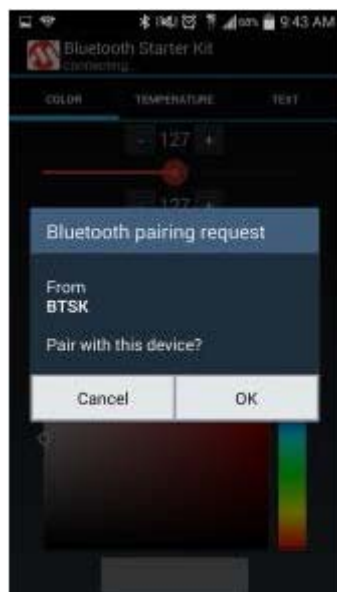


5. A list of paired devices will appear. If this is the first time connecting with the PIC32 Bluetooth Starter Kit, click "Scan for Devices". It should be noted that sometimes the name is not resolved and will appear as "null". After some time the name will change from "null" to "BTSK". The MAC Address will be fixed for the first eight digits and the last four will vary "12:34:45:78: XX: XX" when the device is programmed with code.

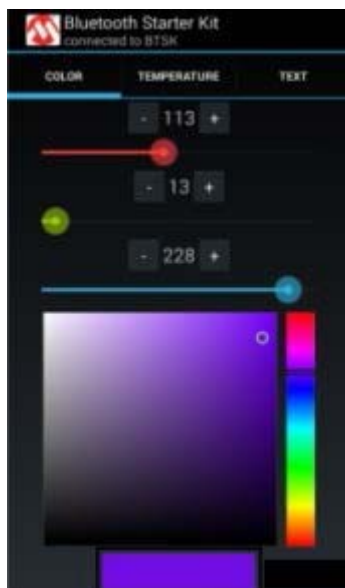


PIC32 Bluetooth Starter Kit SPP Demonstration

- When selected, the Android application will pair and connect with the PIC32 Bluetooth Starter Kit. Accept any pair requests, as follows.



- Status indicators will confirm that the connection was successful, as shown in the following figure.

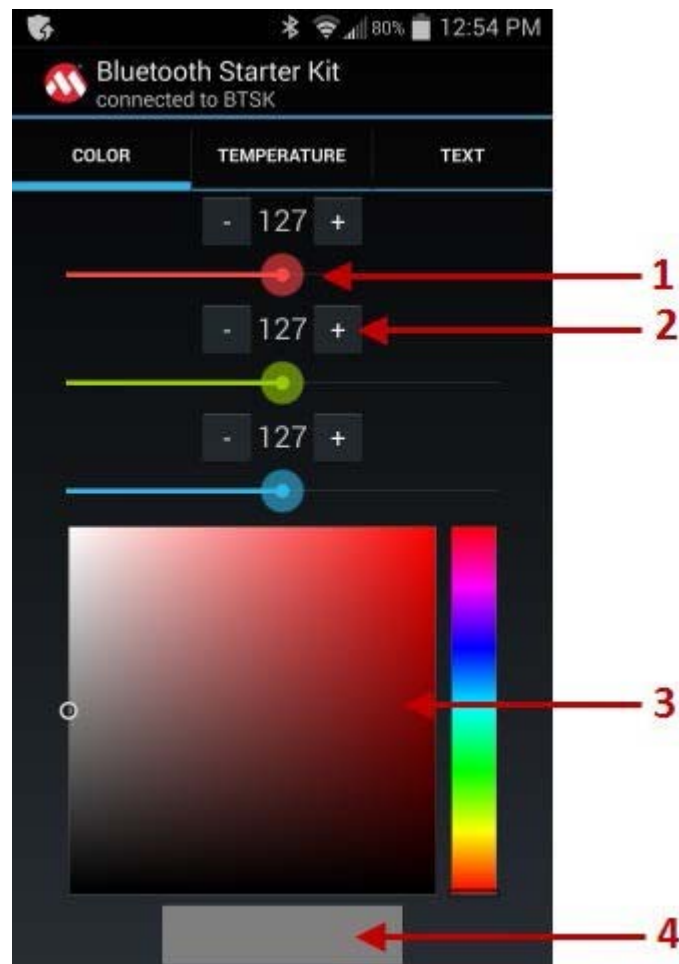


Note: If the application was unable to connect, verify that the PIC32 Bluetooth Starter Kit is powered on, within range, and is not connected to another Bluetooth device.

1.4.1 LED Color Control

The application consists of three color sliders for Red (R), Green (G), and Blue (B), respectively. The color of the LED is programmed as the slider **(1)** is modified. The color of the LED can also be modified using the increment button or the decrement button **(2)**. Similarly, the color can be modified by selecting the color based on hue and saturation on the color palette **(3)**. The slider and the increment/decrement buttons send commands to the PIC32 Bluetooth Starter Kit via SPP to program the LED color. The colors of R, G, and B can vary from 0 to 254, respectively. The color bar **(4)** indicates the resulting modified color of the RGB combination. The LED changes in real-time to approximate the color in the color bar. The LED color will be of an uncalibrated nature of the integrated three color diodes.

This is a demonstration of full duplex data transmission from and to the Android device to the development board. Each time the color is modified, a command is sent in a string format via SPP to the development board. The command sent can be viewed in the "Text" menu.



1.4.2 Temperature Sensor

Select the Temperature menu on the Android application and the window will appear as shown in the next figure. Fahrenheit temperature readings are shown in a graph, which updates automatically as new readings are received. The graph is zoomable and scrollable in the X direction. The last received reading is also displayed as large text.

The PIC32 Bluetooth Starter Kit has a temperature sensor and once Start is selected, the temperature will be updated for every second by default in a periodic mode. The update rate can be modified from 250 milliseconds to 8 seconds by using the increment or the decrement button **(1)**. The time duration can also be modified by the slider provided in the application **(2)**. Once the time duration is set, select "Set Timer" to initiate periodic update for the modified duration. The temperature update can be halted by selecting Stop. Start/Stop and Set Timer send/receive commands to/from the PIC32 Bluetooth Starter Kit through SPP full-duplex transmission. The commands sent and received can be viewed in the Text menu.



1.4.3 Text Control

1.4.3.1 LED CONTROL

The LED color can be modified by sending command via Terminal emulator of the app. Select the "Text" submenu to view the terminal window of the application. To modify the LED color the following command is sent in the format, 255, 03, R, G, B, where:

- 255 is the command to modify the LED color.
- 03 determines the number of Bytes to be sent, currently it is 3 (R, G, B)
- R denotes the value for Red color from 0-254
- G denotes the value for Green color from 0-254
- B denotes the value for Blue from 0-254

For example, 255, 03, 127, 60, 128 (R = 127, G = 60, B = 128)

The commands 'r' or 'g' or 'b' set the LED to 50% of Red, Green, or Blue, respectively. Similarly, the commands 'R' or 'G' or 'B' set the LED to 100% of Red, Green, or Blue, respectively. Refer to [Table 1-1](#) for details.

1.4.3.2 TEMPERATURE CONTROL

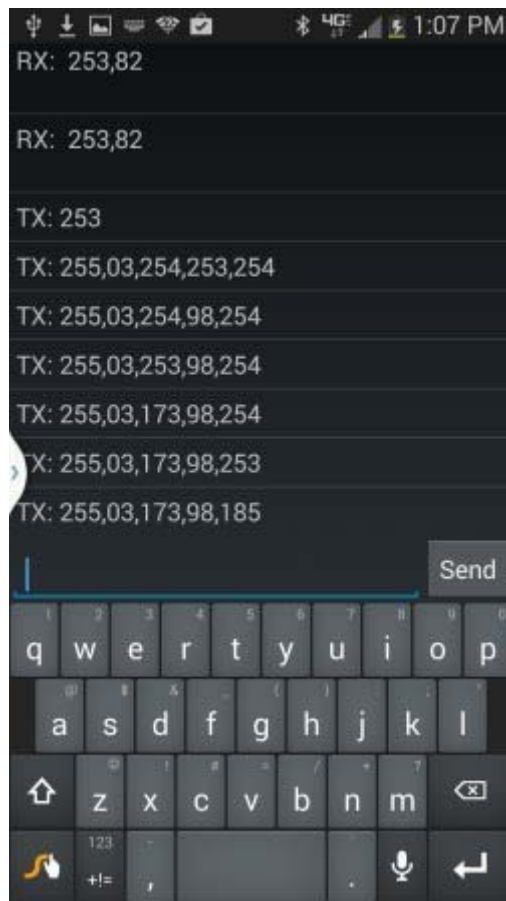
The command 254 is sent to the PIC32 Bluetooth Starter Kit via SPP data transmission. This command transmits the current temperature to the Android device terminal emulator once on request. Similarly, command 253 is sent to the PIC32 Bluetooth Starter Kit to receive the current temperature on a periodic time period for every one second by default. The rate of the update can be modified by the following command 252, with the time in milliseconds.

252 - Command the update timer rate change. The periodic temperature update can be stopped by sending command 253. Refer to [Table 1-1](#) for details.

TABLE 1-1: TEXT COMMANDS

Feature	Command	TX Format	RX Format	Description	Example
LED	255	255,03,R,G,B	N/A	The led color is modified depending on the value of R, G and B varying from 0-254 respectively.	255,03,127,127,127
LED	R	R	N/A	Programs LED for 100% of Red	R
LED	G	G	N/A	Programs LED for 100% of Green	G
LED	B	B	N/A	Programs LED for 100% of Blue	B
LED	r	r	N/A	Programs LED for 50% of Red	r
LED	g	g	N/A	Programs LED for 50% of Green	g
LED	b	b	N/A	Programs LED for 50% of Blue	b
Temperature	254	254	Temperature in Fahrenheit	On transmitting 254 command, the PIC32 Bluetooth Starter Kit sends the current temperature once per request	254
Temperature	253	253	253, temperature in Fahrenheit	On transmitting 253, the PIC32 Bluetooth Starter Kit updates the temperature for every one second	TX: 253 RX:253,80
Temperature	252	252, rate in milliseconds	253, temperature	Programs LED for 100% of Green	TX: 253,399 RX: 253,80
Temperature	253	253	N/A	The periodic display can be halted by resending 253	N/A

PIC32 Bluetooth Starter Kit SPP Demonstration



1.5 ADDITIONAL BLUETOOTH RESOURCES

1.5.1 Other Android Applications

The Bluetooth data smartphone demonstration can also be executed using the Bluetooth SPP-pro application. This app can be downloaded by visiting:

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

Once installed, the commands can be sent and received in CMD line mode/Keyboard mode of the terminal emulator of the application. The commands will be sent and received in the format shown in the following table.

1.5.2 Windows Handset Applications

For a Windows mobile handset, the commands can be sent and received via data terminal. The application can be downloaded by visiting:

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

Once installed, the commands can be sent and received from the terminal emulator of the application. The commands can be sent and received in the format shown in the following table.

1.5.3 Bluetooth Data Windows PC Demonstration Setup

1. On the Windows PC, select *Start > Control Panel > Hardware and Sounds > Add a device*. A dialog appears with a list of available Bluetooth devices.
2. From the list, select **BTSK**.
3. Open a terminal emulator. For this example PuTTY was used. This Windows application can be obtained by visiting:
<http://www.chiark.greenend.org.uk/~sgtatham/putty/download.html>
4. A connection is required to be established between PuTTY and the PIC32 Bluetooth Starter Kit development board. This can be done by selecting the "Serial" Connection type and the COM port assigned to the board in the PuTTY configuration window. The assigned COM ports can be identified from *Device Manager > Ports*.
5. Once connection is established, the following commands can be sent.

Command	Description
R	Programs LED for 100% of Red
G	Programs LED for 100% of Green
B	Programs LED for 100% of Blue
r	Programs LED for 50% of Red
g	Programs LED for 50% of Green
b	Programs LED for 50% of Blue

1.6 KNOWN ISSUES

The following issues are known to exist in this demonstration:

- This demonstration application will not work with Apple iOS devices
- Occasionally, the demonstration experiences connectivity problems. This issue is scheduled to be corrected in a future release.
- This demonstration application does not use a MPLAB Harmony NVM driver for storing the randomly generated last word of Bluetooth device address into NVM. The system will not store the Bluetooth link-key, and therefore, must be paired again upon each power-on.

1.7 REVISION HISTORY

Version 1.0.0

This is the initial version of the demonstration.