Quick Start Guide

WICED Smart

WICED Smart™ Development System

WICED-Smart-QSG202-R

5300 California Avenue • Irvine, CA 92617 • Phone: 949-926-5000 • Fax: 949-926-5203

July 29, 2014

Revision History

Revision	*Date*	*Change Description*
WICED-Smart-QSG202-R	07/29/14	**Updated:** • The document structure by converting to sections. • “WICED Smart Development System Overview” on page 8. • “Setting Up the WICED Smart SDK” on page 10. • “Using the WICED Smart IDE” on page 13. • Appendix B: “IDE Hints & Tips,” on page 25. • Appendix D: “Recovering a Corrupt WICED Smart Tag,” on page 27. **Added:** • “Development Process” on page 9. **Removed:** • Appendix C: “Using WICED Smart SDK With Eclipse on Linux.”
WICED-Smart-QSG201-R	05/26/14	**Updated:** • Figure 2: “Reset Button,” on page 14 • Figure 9: “Application Button,” on page 21 • Appendix E, “Recovering a Corrupt WICED Smart Tag”, on page 26 • Figure 11: “Boot ROM Button Location–SW2, 3, and 4 Positions,” on page 26
WICED-Smart-QSG200-R	04/02/14	Initial release for WICED Smart SDK 2.0

Broadcom Corporation
5300 California Avenue
Irvine, CA 92617

Table of Contents

List of Figures

List of Tables

About This Document

Purpose and Scope

This document provides detailed instructions to set up the Broadcom® Wireless Internet Connectivity for Embedded Devices (WICED; pronounced "wick-ed") Smart development system.

The WICED™ Smart development system supports application development using a WICED Smart Tag. The development system is compatible with the Windows, OS X, and Linux operating systems. This document describes the software components included in the WICED Smart development system and provides instructions for compiling WICED Smart sample applications using the WICED Smart Integrated Development Environment (IDE).

The instructions in this document must be completed before the WICED Smart SDK and WICED Smart Tag can be used.

Note: This document applies to **WICED Smart SDK 2.1.0.**

Audience

This document is for software developers who are using the WICED Smart Development System to create applications for Broadcom Bluetooth Smart devices.

Acronyms and Abbreviations

In most cases, acronyms and abbreviations are defined on first use. For a comprehensive list of acronyms and other terms used in Broadcom documents, go to: http://www.broadcom.com/press/glossary.php.

Document Conventions

The following conventions may be used in this document:

Convention	*Description*
Bold	Buttons, tabs, lists and other GUI items: click **Next**, select the **Startup** tab.
`Monospace`	`Command lines and application outputs:` `hello_sensor-BCM920737TAG_Q32 download`
`< >`	`Placeholders for` *`required`* `elements: <WICED-Smart-SDK>`
`''`	Application Names, Configuration Parameters: 'heart_rate_monitor'

References

The references in this section may be used with this document.

Note: Broadcom provides customer access to technical documentation and software through its Broadcom Support Community website (community.broadcom.com). Additional restricted material may be provided through the Customer Support Portal (CSP) and Downloads (support.broadcom.com).

For Broadcom documents, replace the "xx" in the document number with the largest number available to ensure you have the most current version of this document.

	Document (or Item) Name	Number	Source
[1]	*WICED™ Smart Tag User Guide*	920737TAG03-HWUM1xx-R	WICED Smart SDK
[2]	*Smart Designer: Using the WICED SMART IDE to Develop Custom Applications*	WICED-SWUM10x-R	WICED Smart SDK
[3]	*BCM20736 Data Sheet*	20736-DS100-R	Broadcom CSP
[4]	*BCM20737 Data Sheet*	20737-DS100-R	Broadcom CSP

Technical Support

Broadcom provides customer access to a wide range of information, including technical documentation, schematic diagrams, product bill of materials, PCB layout information, and software updates through its customer support portal. For a CSP account, contact your Broadcom Sales or Engineering support representative.

General WICED™ support is available to registered users via the Broadcom Support Community website:

community.broadcom.com

Section 1: WICED Smart Development System Overview

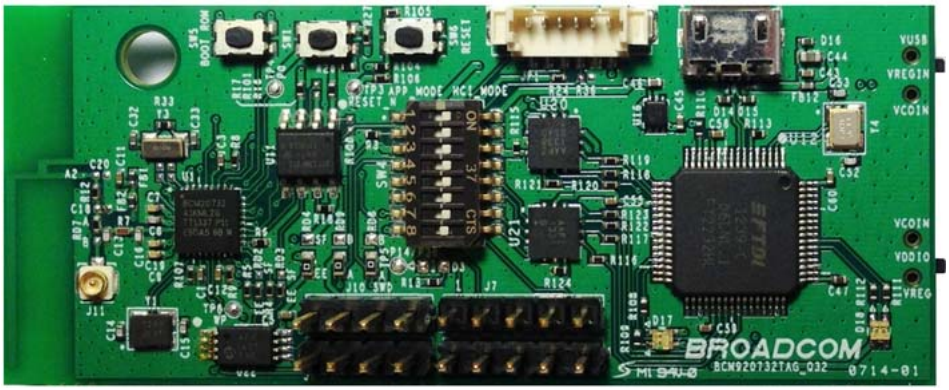
The WICED Smart development system is composed of a WICED Smart tag, the WICED Smart Software Development Kit (SDK), and the WICED Smart Integrated Development Environment (IDE).

WICED Smart Tag

The Broadcom WICED Smart tag (Figure 1) incorporates a Broadcom BCM20737 Bluetooth Smart chip and additional circuitry to enable programming, debugging, and evaluation of Bluetooth Smart applications. The BCM20737TAG supports a superset of BCM20736 and BCM20737 features.

The BCM20737TAG, included in the BCM92073X_LE_KIT, can be used for feature evaluation, debugging, and developing Bluetooth Smart applications for either the BCM20736 or the BCM20737. For final production, the desired feature must be mapped to the chip that it is available on. Refer to the specific chip's data sheet or the product brief for that purpose.

Figure 1: WICED Smart Tag



Software Development Kit

The WICED Smart SDK includes:

- Bluetooth Smart software stack including GAP, ATT, GATT, and SMP profiles
- Generic profile-level API
- Drivers to access onboard peripherals including UART, SPI, I^2C, ADC, PWM, Keyscan, etc.
- Reference applications for the devices with profiles defined by the Bluetooth SIG
- WICED Smart API documentation and related documents
- Utilities to support development in Windows, OS X, and Linux environments

Directory Structure

Table 1 is an overview of the top-level directory of the WICED Smart SDK.

Table 1: Overview of the WICED Smart SDK Top-Level Directory

WICED-Smart SDK Directories	*Directory Contents*
Apps	ROM & RAM-based example applications
Doc	API, reference documentation, and schematics
Drivers	Windows USB-serial drivers for the WICED Smart tag
include	WICED API function prototypes, constants, and defaults
Platforms	Configuration files and information for supported hardware platforms
Tools	Tool chain including compiler, download tool, and other utilities/scripts
Wiced-Smart	WICED Smart core Bluetooth components

Hardware and Software Requirements

- The WICED Smart SDK runs on 32- and 64-bit versions of Microsoft Windows, OS X, and Linux.
- The SDK is distributed as:
  - A standalone 7zip file suitable for all operating systems.
  - A bundle with the WICED Integrated Development Environment as an executable installer for Windows and the Mac operating systems.
- The development computer requires a single USB port to connect to the WICED Smart tag.

Note: The 7zip extraction utility is available from 7-zip.org. 7zip is needed to extract the standalone WICED Smart SDK 7zip archive. The standard Windows zip file extraction utility may silently corrupt the SDK archive during the extraction process. Do *NOT* use it.

Development Process

To prepare and run an application, perform the following high-level steps:

1. Download and install the WICED Smart SDK and IDE “Install the WICED Smart SDK and IDE” on page 10.
2. Connect the WICED Smart tag (see “Connect the WICED Smart Tag” on page 11.)
3. Create and load an application (see “Build and Load a Sample Application” on page 13 and Reference [2] on page 7).
4. Run the application (see “Run an Application” on page 17).

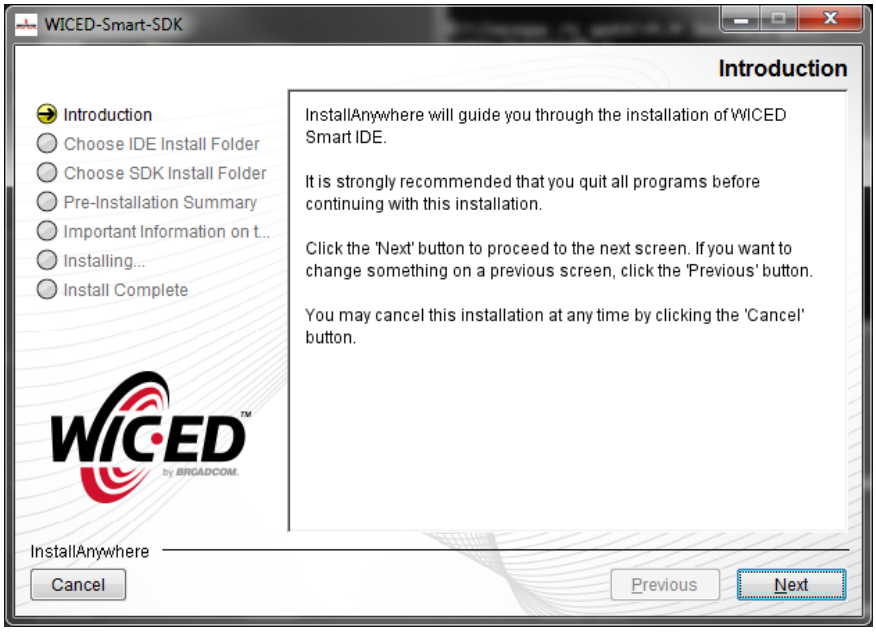
Section 2: Setting Up the WICED Smart SDK

Download the WICED Smart SDK from the Broadcom Support Community website (community.broadcom.com) or from the Broadcom Customer Support Portal (support.broadcom.com).

Windows Instructions

Install the WICED Smart SDK and IDE

The WICED Smart SDK is provided as a self-installing executable file. Double-click the **Wiced-Smart-SDK-2.x.x-IDE-Installer.exe** file to begin the installation. A setup window similar to the screenshot below is displayed.



After being presented with the above screen:

1. Click **Next** to continue with the installation.
2. In the Choose IDE Install Folder window, choose the IDE installation folder and click **Next**.
3. In the Choose SDK Install Folder window, choose the SDK installation folder and click **Next**.
4. In the Pre-Installation Summary window, click **Next** to install using the current IDE and SDK folder selections. (Click **Previous** once or twice to modify the SDK and/or IDE installation folders, respectively.)

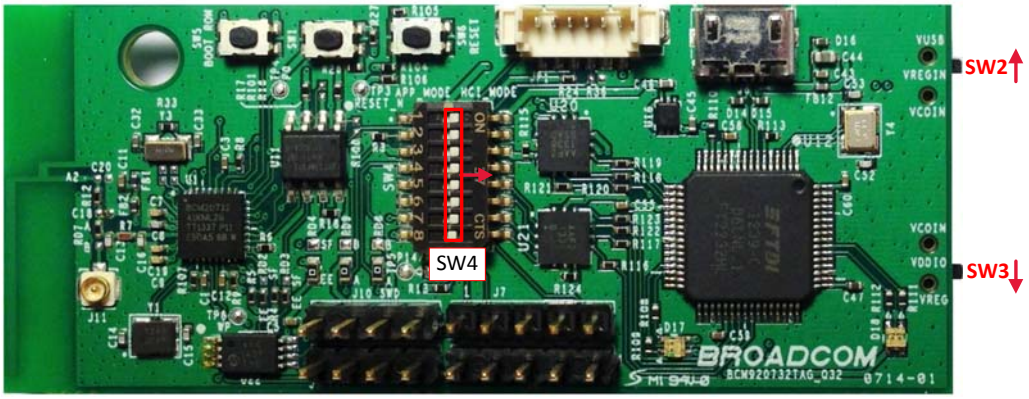
After installation has completed, start the WICED Smart IDE by using the WICED Smart IDE desktop icon.

Connect the WICED Smart Tag

The USB interface connects the WICED Smart tag to the PC. The USB interface provides a serial-UART interface for programming and debugging, and can also provide +5V power.

To enable the USB interface of the tag, verify the following switch settings before proceeding:

1. Each of the eight DIP switches on the **SW4** mini-switch are set to ON.
2. **SW2** should be in the VUSB position (up) and **SW3** should be in the VREG position (down).



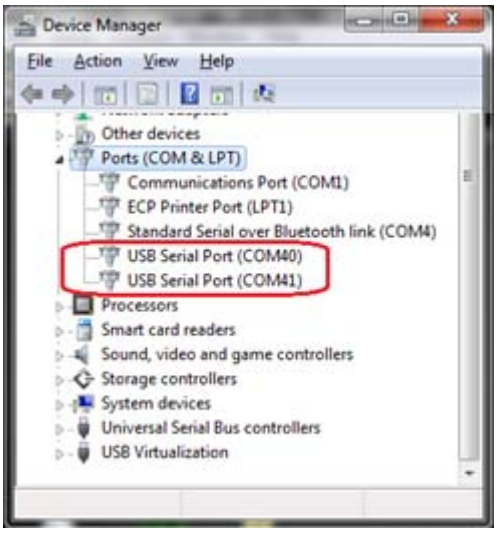
Connect the WICED Smart tag to the development PC with a USB cable; the USB UART driver will load automatically.

Verify Driver Installation

To verify that driver installation is complete:

1. Open the **Device Manager** (right-click **My Computer** and select **Properties**).
2. In the **System** Properties window, select **Hardware, Device Manager**. The two WICED Smart tag USB serial ports are listed under **<PC-NAME>\Ports (COM & LPT)** as shown in the screenshot below.

Note: In the screenshot, the Device Manager identifies the WICED Smart tag USB serial COM ports as COM40 and COM41. The assigned port numbers vary among systems. The first port (HCI UART) is used to download the application to the WICED Smart tag; the second can be used to talk to the chip over the peripheral UART.



Note: If an error occurs during driver installation, download new drivers from Windows Update. Verify you have an Internet connection, disconnect from and then reconnect to the WICED Smart tag, and wait for the drivers to automatically install. If the WICED Smart tag does not appear in the Device Manager after reinstalling drivers, then check all switch settings (see "Connect the WICED Smart Tag") on the tag and/or replace the USB cable.

Section 3: Using the WICED Smart IDE

This section describes how to:

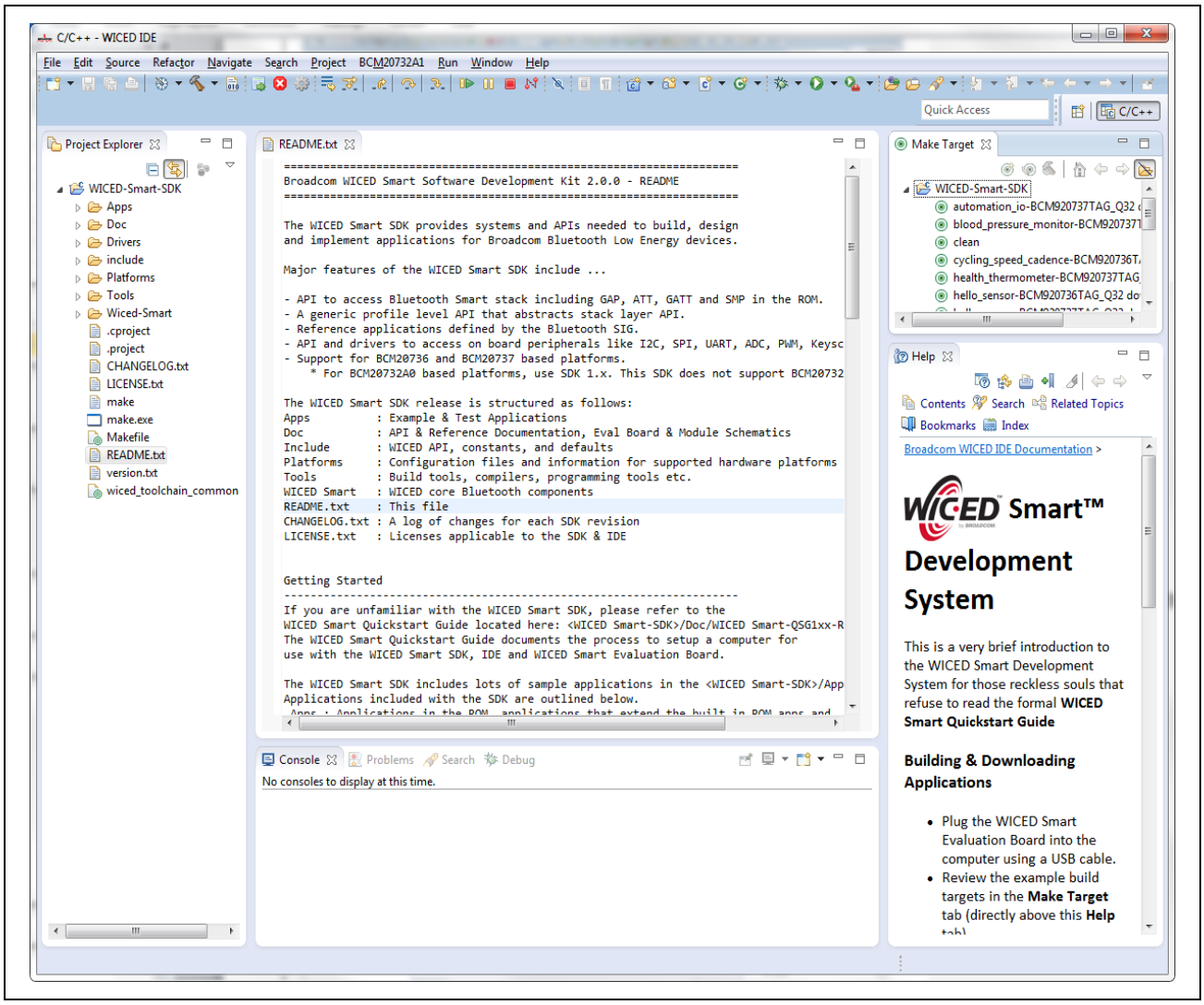
- Use the WICED Smart IDE to create application build targets for the WICED Smart tag.
- Download applications to the tag.
- Select the power supply for the tag.
- Verify that the application running on the tag is working correctly using an iOS or Android mobile device with support for Bluetooth Low Energy.

If a Bluetooth Low Energy capable iOS or Android mobile device is not available, "Testing a WICED Smart Tag with Windows 8" on page 22 describes an alternate method for testing applications running on the WICED Smart tag. You can also use a Windows 7 PC equipped with a Broadcom chip or with a Broadcom chip-based Bluetooth 4.0 dongle.

Build and Load a Sample Application

Start the WICED Smart IDE by selecting **START > All Programs > Broadcom > WICED Smart IDE**. The WICED Smart IDE looks similar to the screenshot shown in Figure 2 on page 14.

Figure 2: WICED Smart IDE



The **Help** pane in the lower-right corner of the IDE (see Figure 2) describes how to build and download the sample applications shown in the **Make Target** pane, which is located above the Help pane. The Make Target pane contains multiple build targets that are preconfigured for various sample applications that run on the BCM20737TAG_Q32 WICED Smart tag platform.

The example below creates a new target for the Heart Rate Monitor application. The target will then be used to build and download the application to the WICED Smart tag.

In the IDE Make Target pane:

1. Right-click the **hello_sensor-BCM920737TAG_Q32 download** target and select **Copy**.
2. Right-click the target again and select **Paste**: a dialog box is displayed.
3. In the dialog box, in the **Target Name** field, replace **Copy of hello_sensor** with **heart_rate_monitor** and click **OK.**

Double-click the newly created **heart_rate_monitor-BCM920737TAG_Q32 download** target to build the heart rate monitor application: the IDE console pane (bottom center of the IDE window) displays the build progress. The build target is appended with the *download* option, which tells the tool chain to download the firmware when the build is complete.

Note: If the *download* option is not automatically appended, users must type it in.

The build output looks similar to the following:

```
20:58:12 **** Build of configuration Release for project WICED-Smart-SDK ****
"C:\\Users\\Victor\\Documents\\WICED\\WICED-Smart-SDK-2.1.0\\WICED-Smart-SDK\\make.exe"
heart_rate_monitor-BCM920736TAG_Q32 download
Compiling spar_setup.c
Compiling heart_rate_monitor.c
Compiling sparinit.c
Compiling lib_installer.c
Linking target ELF
OK, made elf.
..\..\Tools\ARM_GNU\bin\Win32\arm-none-eabi-objdump: section '.data' mentioned in a -j option, but not
found in any input file
Call to heart_rate_monitor_spar_crt_setup @ 002053f9
OK, made C:/Users/Victor/Documents/WICED/WICED-Smart-SDK-2.1.0/WICED-Smart-SDK/Wiced-Smart/spar/../../
build/heart_rate_monitor-BCM920736TAG_Q32-rom-ram-Wiced-release/A_20736A1-heart_rate_monitor-rom-ram-
spar.cgs. MD5 sum is:
7bc441df83a594091fc8fe53aa37382a *../../build/heart_rate_monitor-BCM920736TAG_Q32-rom-ram-Wiced-release/
A_20736A1-heart_rate_monitor-rom-ram-spar.cgs

--------------------------------------------------------------------------------
Patches start at                  0x00204568 (RAM address)
Patches end at                    0x00205280 (RAM address)
Application starts at             0x00204F9C (RAM address)
Application ends at               0x002053F8 (RAM address)

Patch size (including reused RAM)       3352 bytes
Patch size                              2612 bytes
Application size                        1116 bytes
                                       ------
Total RAM footprint                     3728 bytes (3.6kiB)
--------------------------------------------------------------------------------
Converting CGS to HEX...
Conversion complete

Creating OTA images...
Conversion complete
OTA image footprint in NV is 4973 bytes

Detecting device...
Device found

Downloading application...
Download complete

Move DIP switch 2 of SW4 to off position and push Reset button to start application

20:58:24 Build Finished (took 12s.702ms)
```

During firmware download, a red LED labeled 'D2' on the WICED Smart tag flashes to indicate that a firmware download is in progress.

If the WICED Smart tag is not recognized or detected by the programming tools, the following message may appear:

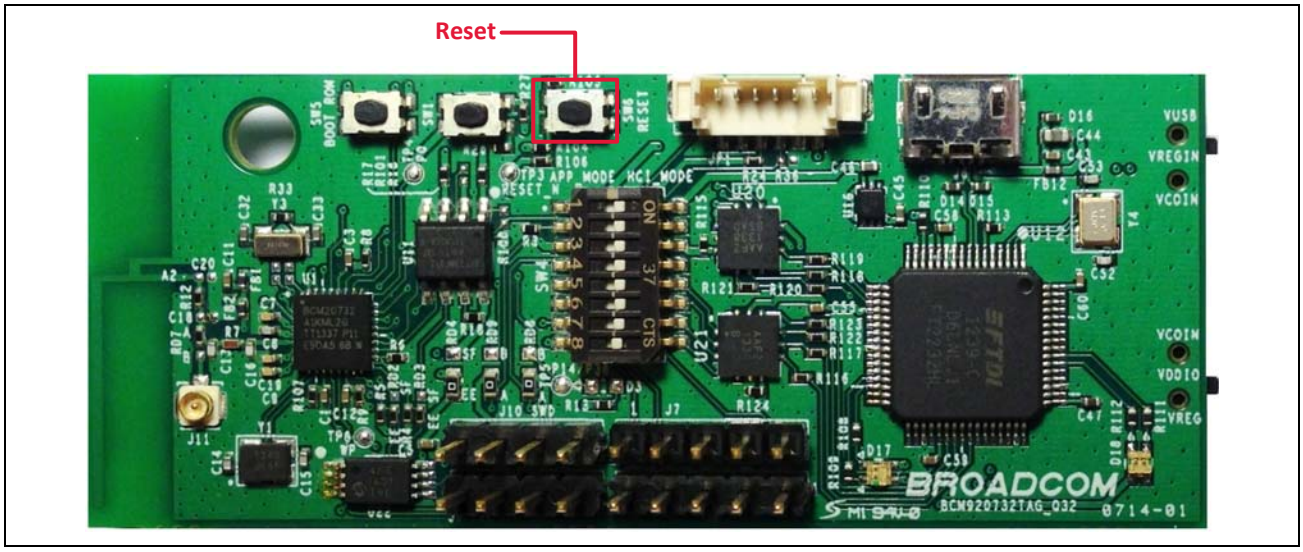
```
Detecting device...
+-----------------------------------------------------------------------------------------+
| No BCM20736 detected                                                                    |
| 1. Verify the BCM20736 tag is connected _AND_ powered                                   |
| 2. Verify SW2 and SW3 switches are towards Vusb and Vreg respectively and all SW4       |
|     switches are set to ON                                                              |
| 3. Press the reset button on the tag and retry                                          |
|                                                                                         |
| If this problem persists, the EEPROM on the tag may be corrupted                        |
| Please see Appendix D in the Quick Start Guide for recovery instructions                |
+-----------------------------------------------------------------------------------------+

Download failed. This version of the SDK only supports download to BCM20736A1 and BCM20737A1 devices
```

If this message appears, do the following steps:

1. Verify that the WICED Smart tag is plugged into the development PC.
2. Verify that the USB-UART driver is correctly installed and all switches (**SW2**, **SW3**, and **SW4**) are correctly set as described in “Connect the WICED Smart Tag” on page 11.
3. Press the WICED Smart tag **Reset** button (see Figure 3) to prepare the tag for firmware download on the UART interface.
4. If a console window or terminal emulation program was used to monitor traces, stop the traces or close the terminal emulation program. To view traces, see “Viewing Traces from the WICED Smart Tag” on page 19.
5. Double-click the build target to rebuild and download the application to the tag.

Figure 3: Reset Button



Note: If the problem persists after following the instructions above, the EEPROM on the WICED Smart tag may be corrupted. See the instructions in "Recovering a Corrupt WICED Smart Tag" on page 27 to erase the EEPROM on the tag.

Run an Application

This section assumes that "Build and Load a Sample Application" on page 13 has been successfully completed and that the heart rate monitor application has successfully been downloaded to the WICED Smart tag.

Before the application can be tested, the power supply for the tag must be selected.

USB Powered Application

During programming, the tag is powered by the USB interface. After programming is complete, start the application by moving DIP switch 2 of **SW4** to the OFF position and then pressing the **Reset** button. Further attempts to download an application will fail because the UART port is no longer configured for downloads.

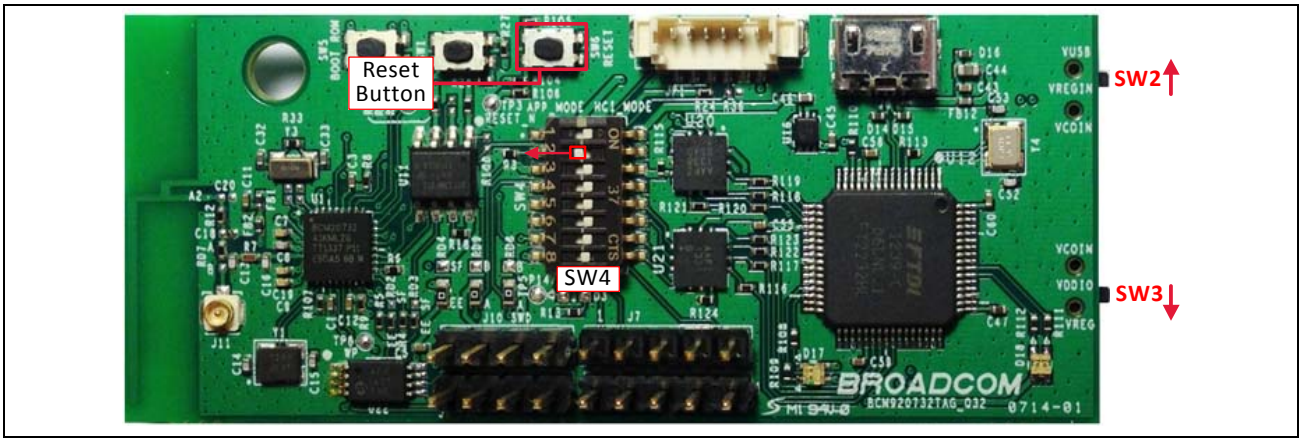
To enable downloads again, move DIP switch 2 of **SW4** to the ON position and press the **Reset** button.

Coin Cell Powered Application

To run the application from a CR2032 coin cell:

1. Disconnect the tag from the PC USB interface.
2. Set the position of the switches on the right side of the board (**SW2** and **SW3**) to the VCOIN positions, as shown in Figure 4. The switches should now be set to the opposite positions of those used during programming.
3. Install a CR2032 coin cell in the battery holder on the bottom of the tag. Ensure that the battery polarity is correct.
4. Move DIP switch 2 of **SW4** to the OFF position (left) and press the **Reset** button.

Figure 4: Coin Cell Power Switch Positions

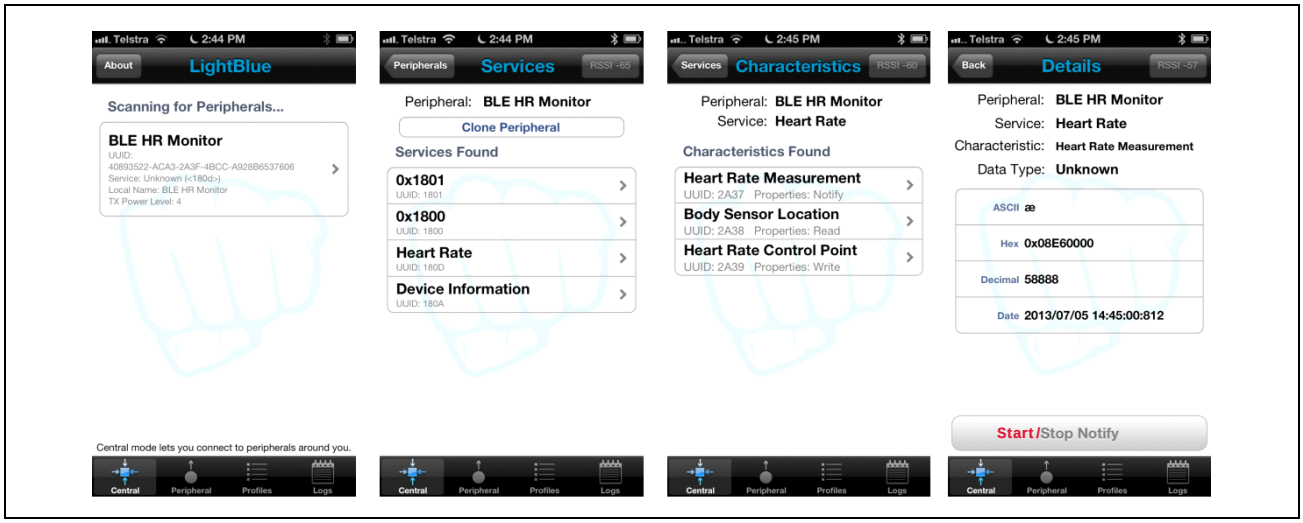


Application Test Example

A number of BLE applications are available for phones and tablets. In this example, the LightBlue iOS application is used on an Apple iPhone 5.

Download the LightBlue App (from the Apple App Store) to the iOS device and start the application. Ensure that **Central** is selected on the bottom taskbar. If the heart rate monitor app is working correctly, the LightBlue App shows **BLE HR Monitor** in a box, as shown in the upper-portion of the left-most LightBlue window in the screenshot in Figure 5. The heart rate monitor application is configured to regularly send dummy heart rate measurements. To see measurements update in real time: tap **BLE HR Monitor > Heart Rate > Heart Rate Measurement** and then tap **Start Notify** at the bottom of the **Details** screen.

Figure 5: LightBlue iOS Application Showing the WICED Smart Tag Heart Rate Monitor Application



Note: A description of the heart rate monitor application is provided in the header comments of the heart_rate_monitor.c source file located in the WICED Smart SDK at:
`<WICED-Smart-SDK>/Apps/heart_rate_monitor/heart_rate_monitor.c`

Viewing Traces from the WICED Smart Tag

Traces from the application can be viewed by configuring the WICED Smart tag to route traces to the HCI UART, and then to the COM port of the platform.

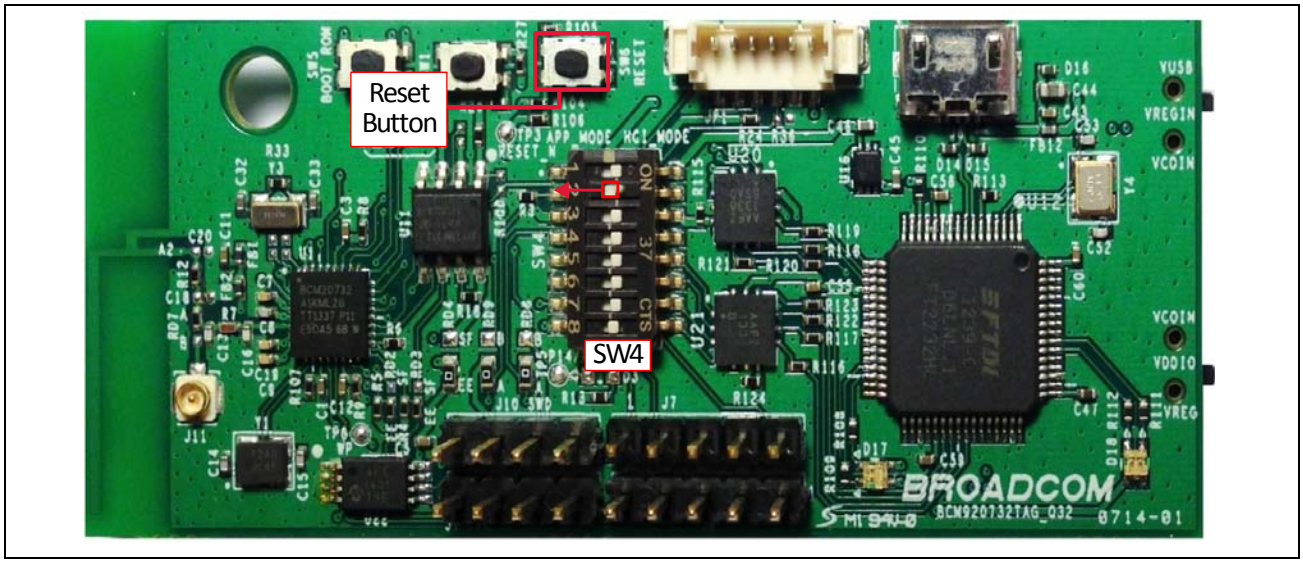
To view the traces:

1. In WICED Smart IDE, in the **Project Explorer pane** (left side of the IDE, see the screenshot on page 13), in the **WICED-Smart folder**, locate and open the `WICED-Smart-SDK\Wiced-Smart\spar\common\sparinit.c` **file**.
2. Make sure that the application_setup function includes the following line:
   `BLE_APP_ENABLE_TRACING_ON_HCI_UART();`
3. Compile and download the application to the WICED Smart tag as described in "Build and Load a Sample Application" on page 13.

Note: To successfully download the application to the WICED Smart tag, DIP switches 1 through 8 on the mini-switch must be set to the ON position.

4. Wait for the *Build Finished* message to display in the console window.
5. Locate the mini-switch (**SW4**) on the WICED Smart tag, and set DIP switch 2 to the OFF position. The final switch settings of mini-switch **SW4** are: 1 = ON, 2 = OFF, and 3 through 8 = ON.

Figure 6: Configure UART for Tracing



6. Press the **Reset** button on the WICED Smart tag (Figure 6).
7. In the WICED Smart IDE menu select **Trace > Start Debug Traces**.

Note: The content of the ble_trace statements should appear in the console window of the WICED Smart IDE.

8. To restart the application and view the startup traces of your application, press the **Reset** button on the WICED Smart tag (Figure 6).

Instead of using the console window of the IDE, a terminal emulation program, such as Tera Term or Hyperterm, can be used. Configure the terminal emulation program to use the COM port used by the WICED Smart tag (see "Verify Driver Installation" on page 12 for instructions on identifying the COM port to use).

Note: Verify that the terminal emulation program serial port settings are configured as:

1. Baud Rate: **115200**
2. Parity: **None**
3. Stop bits: **1**
4. Flow Control: **None**.

In addition, the terminal should be set to **VT100 terminal** and the **New Line > Receive** parameter should be set to **Auto**.

To exit Debug mode and download a new version of the application to the WICED Smart tag:

1. In the WICED Smart IDE, select **Trace > Stop Debug Traces** (or exit the terminal emulation program used to monitor debug output traces).
2. Make sure all DIP switches of mini-switch **SW4** are set to the ON position.
3. Press the **Reset** button on the WICED Smart tag (see Figure 6 on page 19).

Note: Downloading a new application to the WICED Smart tag should work exactly as described in "Build and Load a Sample Application" on page 13 and "Run an Application" on page 17.

What Now

Now that you have a basic understanding of how to compile and download a WICED Smart application, we recommend building and running the example applications provided in the WICED Smart SDK applications directory.

Some applications configure or extend the applications in the WICED Smart device ROM. Others provide examples that run primarily from RAM, and make use of additional functions located in the ROM. The header of the main source file of every application provides additional information on the features demonstrated by the application and the usage model.

For developers working with Windows® operating systems, “Testing a WICED Smart Tag with Windows 8” on page 22 provides instructions that describe how to use a computer running Windows 8 with the Hello Sensor example application included with the WICED Smart SDK.

“IDE Hints & Tips” on page 25 contains hints and tips about navigating the WICED Smart code base.

For help getting started with creating your own application, refer to *Smart Designer: Using the WICED SMART IDE to Develop Custom Applications* (see Reference [2] on page 7) and watch *WICED SMART SDK2.0: Smart Designer Overview*, which is available at community.broadcom.com/community/wiced-smart.

We hope you enjoy using the WICED Smart Development System!

-- *The WICED Development Team*

Appendix A: Testing a WICED Smart Tag with Windows 8

This appendix describes how to use applications running on a Windows 8 PC to interact with a WICED Smart tag. The PC used for testing may be the same PC used to develop WICED Smart applications, or a different PC can be used.

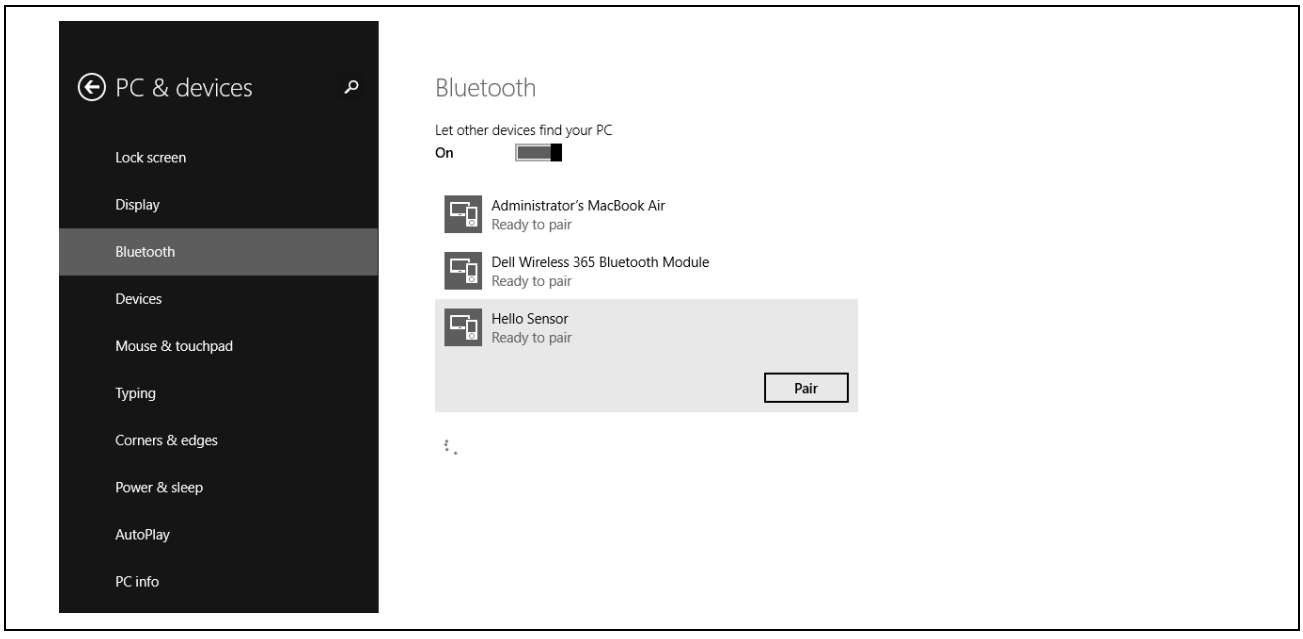
The Hello Sensor Tag Application

The following example describes how to use the WICED Smart SDK Hello Sensor (hello_sensor) sample tag application. The steps outlined in the text apply to Windows 8.1 and may vary slightly compared to Windows 8.

To connect the WICED Smart tag with the Windows PC.

1. Using the WICED Smart SDK:
   a. Build and download the hello_sensor application (see "Build and Load a Sample Application" on page 13).
   b. Run the hello_sensor application (see "Run an Application" on page 17).
2. On the Windows PC open the **Settings** charm (move the mouse to the lower- or upper-right corner of the screen, then up or down, and click **Settings**).
3. Click **Change PC Settings** and select the **Bluetooth** tab (see Figure 7).
4. Click **Add a device** and wait while the PC searches for devices in range.
5. Select the **Hello Sensor** device, click **Pair**, and wait for the device connection to complete. The tag emits an audible beep when connection is complete.

Figure 7: Connecting the WICED Smart Tag to a PC Running Windows 8



The Hello Client PC Application

The Hello Client (hello_client) PC application is provided with the SDK to complement the Hello Sensor (hello_sensor) tag application. It is located here:
`<WICED-Smart-SDK>\Wiced-Smart\Apps\hello_sensor\peerapps\Windows\HelloClient\Release`

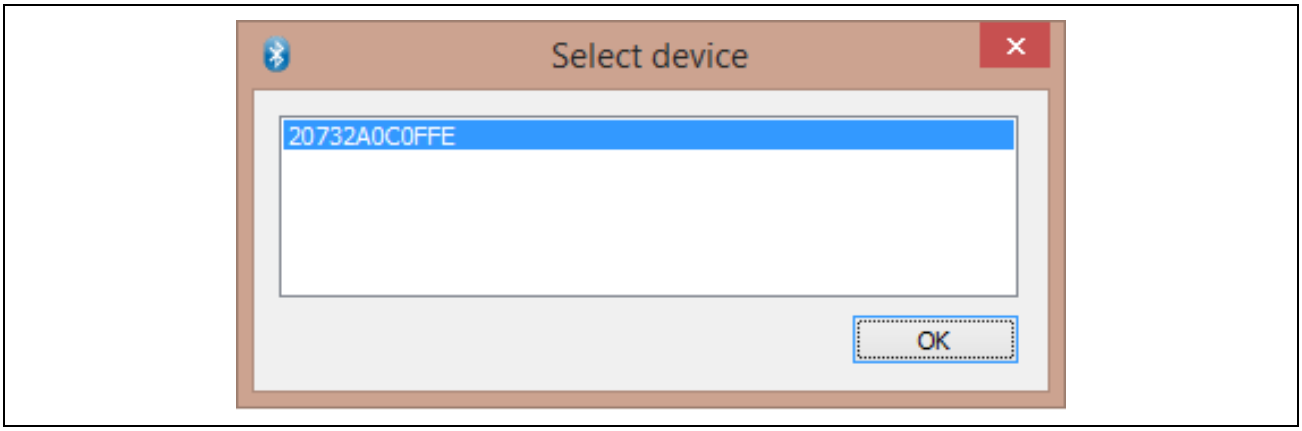
The application is provided as full source code, along with an executable binary that runs on Windows 32-bit and 64-bit machines.

Run HelloClient.exe. A HelloClient **Select device** window similar to that shown in Figure 8 is displayed. The window shows a list of Bluetooth device addresses for Hello Sensors paired to the PC. The Bluetooth device address programmed in the tag is located in:
`<WICED-Smart-SDK>\Platforms\BCM20737TAG_Q32\20737_EEPROM.btp.`

To change the Bluetooth device address, open the file and modify the **DLConfigBD_ADDRBase** variable.

Select the correct device (if it is not already selected) and click OK to initiate a connection to the tag. The connection process may take 5–10 seconds. The tag will emit an audible beep when the connection is complete.

Figure 8: HelloClient Select Device Window



The Hello Sensor application provides paired devices the following information:

- A Hello Service (a proprietary service) with two proprietary characteristics:
  - The value of the Hello Input read-only characteristic may be retrieved using either of the following methods:
    - Manually by using a mouse to click the **Read** button on the HelloClient PC application.
    - Automatically by pressing the **Application** button (see Figure 10) on the WICED Smart tag (the **Allow Notifications** drop-down must be selected to allow automatic notifications).
  - The Hello Configuration read-write characteristic is used to configure how many times a LED on the tag blinks when the **Application** button (see Figure 10) is pressed.
- A Device Information Service that provides information including:
  - Manufacturer Name
  - Model Number
  - System ID
- A Battery Service that provides a battery-level indication

Testing the Hello Application

To test the application with a WICED Smart tag, follow the instructions provided for each of the Hello Service characteristics (see Figure 9, below).

- Hello Input Characteristic
  a. Select **Allow Notifications** in the combo box.
  b. Push the middle (**Application**) button on the WICED Smart tag (see Figure 10). The *Hello X* message is displayed in the Value field. Each time the button is pressed, the message number increments.
- Hello Configuration Characteristic
  a. Change the value for the Hello Configuration to 5 and then click **Write**. The tag emits an audible beep and the LED flashes five times to acknowledge the write.
  b. Push the **Application** button (see Figure 10) on the WICED Smart tag. The LED on the tag blinks five times.

Figure 9: HelloClient Application

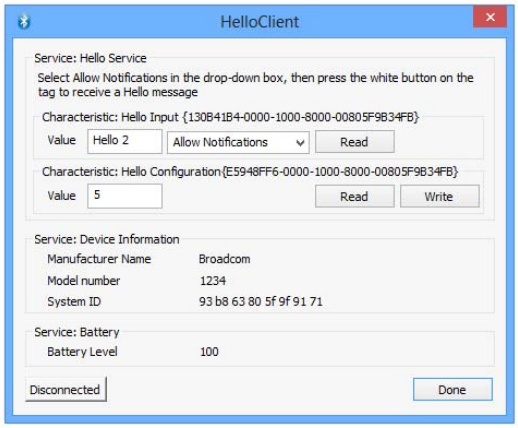
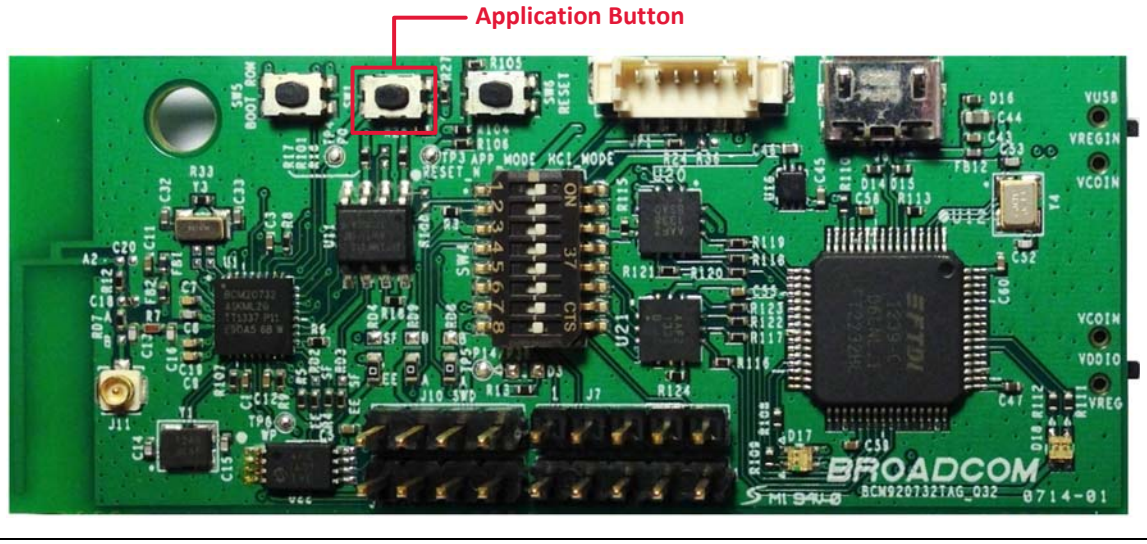


Figure 10: Application Button



Appendix B: IDE Hints & Tips

Hints

1. The **Help** tab (and any other tab) may be click-dragged to any window pane to customize the IDE layout.
2. To revert to the C/C++ perspective (rather than the Debug perspective for example), click the C/C++ icon in the top-right corner of the window.

Shortcuts

A useful cheat-sheet outlining shortcuts for the WICED Smart IDE (Eclipse) is included in the <WICED-Smart-IDE>/Readme directory or online here:

http://www.cheat-sheets.org/saved-copy/eclipseCDT8.0-cheatsheet.pdf

Particularly useful keystrokes are listed below:

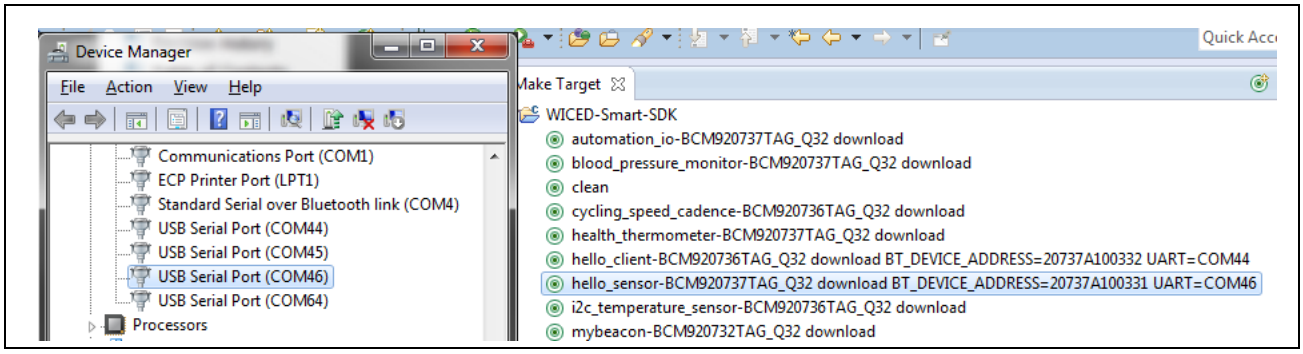
- General search: to search the WICED-Smart-SDK tree for a variable:
  a. **Click** the root WICED-Smart-SDK folder in the Project Explorer pane.
  b. Press **CTRL-H** (on Windows).
  c. In the File Search tab, enter the variable name (regular expressions work too).
  d. Click **Search.**
- Search for a *C* source element (variable, function, enum, etc.).
  a. Open a *C* source file, for example: `<WICED-Smart-SDK/Apps/hello_sensor/hello_sensor.c`.
  b. Press **CTRL-SHIFT-T**
  c. Start typing an element, for example, *bleprofile_Generate.*
  d. Suggestions appear in the popup window.
- Press **ALT-Left** (arrow) and **ALT-Right** (arrow) to navigate between open files.

Appendix C: Multiple WICED Smart Tags

Multiple tags can be programmed from a single computer to run the same or different applications. To use the feature, edit the make target for the required application to add the `UART=COMx` and `BT_DEVICE_ADDRESS=<address>` parameters.

Figure 11 shows two WICED Smart tags connected to a PC and appropriate targets to build and download the hello client and hello sensor applications.

Figure 11: Configuration for Two WICED Smart Tags

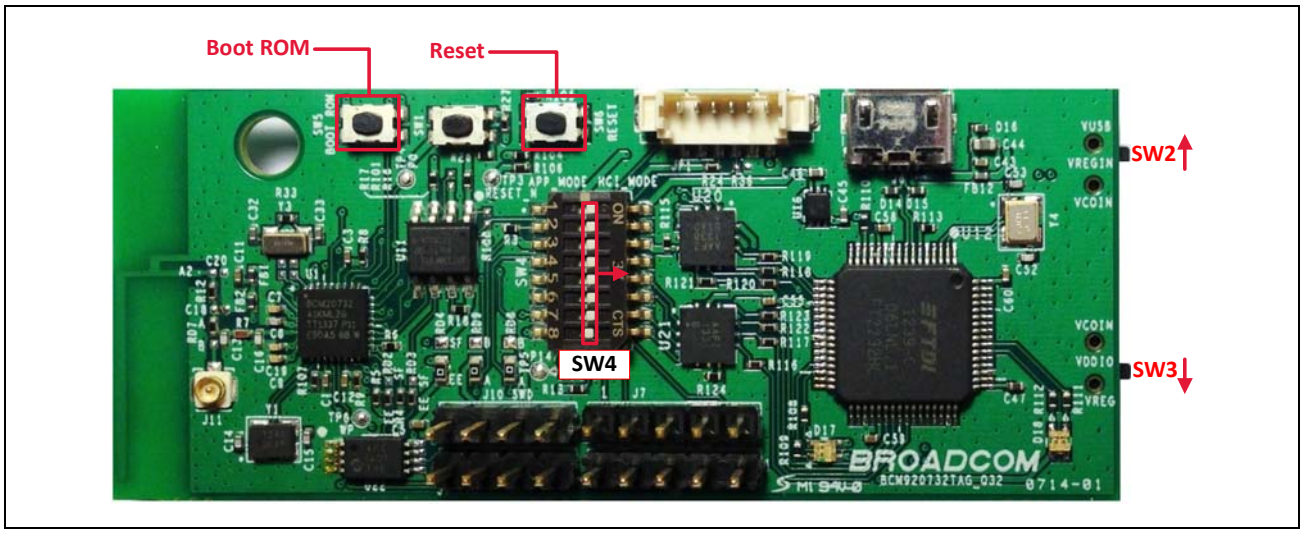


Appendix D: Recovering a Corrupt WICED Smart Tag

The following steps describe how to recover a WICED Smart tag if the EEPROM is corrupted.

1. Identify the USB-serial COM port assigned to the WICED Smart tag. Refer to the instructions provided in "Verify Driver Installation" on page 12, if required.
2. Make sure that the position of all DIP switches on mini-switch **SW4** are set to ON (right). **SW2** should be in the **VUSB** position (up) and **SW3** should be in the **VREG** position (down) (see Figure 12).
3. Press and hold the **Boot ROM** button. The **Boot ROM** button is outlined by a red rectangle in Figure 12.
4. Continuing to hold the **Boot ROM** button, push and release the **Reset** button.
5. Release the **Boot ROM** button.

Figure 12: Boot ROM Button Location—SW2, SW3, and SW4 Positions



6. Create the following make target (where COM*x* is the UART COM port assigned to the tag, replace *x* with the number assigned to the tag).

   ```
   proximity-BCM920737TAG_Q32 recover UART=COMx
   ```

   See "Build and Load a Sample Application" on page 13 for instructions, if required. The build target includes the *recover* option to indicate the recovery procedure should be performed after the build completes.
7. Double-click the newly created build target to start the recovery procedure. The IDE console window displays the build progress. At the end of the build, the following messages are displayed:

   ```
   Converting CGS to HEX...
   Conversion complete

   Recovering platform...
   Recovery complete
   ```


WICED
TM
by BROADCOM