WICED™ Development System

Revision History

Revision	Date	Change Description
WICED-QSG202-R	April 19, 2013	Update for WICED-SDK-2.3.0
WICED-QSG201-R	December 14, 2012	Update for WICED-SDK-2.2.0
WICED-QSG200-R	November 22, 2012	Released
WICED-QSG200-D1	September 26, 2012	Updates for WICED-SDK-2.0
WICED-QSG102-D1	December 23, 2011	Include notes for integrated WICED™ IDE
WICED-QSG101-R	November 9, 2011	Release with instructions for SDK Installer
WICED-QSG101-D4	November 2, 2011	Draft 4 Release for WICED-SDK-1.1
WICED-QSG100-R	June 2, 2011	Initial Release

Table of Contents

1 About this Document

1.1 Purpose and Scope

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

The WICED development system supports application development using a WICED evaluation board. The development system is compatible with Windows®, Mac OS X and Linux. This document describes the software components included in the WICED development system and provides instructions for compiling WICED sample applications in Windows® and OS X using the WICED integrated development environment (IDE).

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

> **Note:** This document applies to **WICED SDK 2.3.0**.

1.2 Audience

This document is for software developers who are using the WICED Development System to create applications for secure embedded wireless networked devices.

1.3 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

1.4 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: `snip.ping-BCM943362WCD4 download run`
< >	Placeholders for *required* elements: <WICED-SDK>
' '	Application Names, Configuration Parameters: 'YOUR_AP_SSID'

1.5 References

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

Note : Broadcom provides customer access to technical documentation and software through the WICED website (http://go.broadcom.com/wiced). Additional restricted material may be provided through the Customer Support Portal (CSP) and Downloads.

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™ Evaluation Board User Guide	WICED-EUM2xx-R	WICED website

1.6 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 on the Broadcom forum at the URL shown below. After signing up to the forum, please apply to be a member of the WICED users group and identify yourself to the forum moderator. Access to the WICED forum is restricted to bona-fide WICED customers only.

http://forum.broadcom.com/forum.php

2 WICED Development System

The WICED development system is composed of a WICED evaluation board, the WICED software development kit (SDK) and the WICED Integrated Development Environment (IDE).

2.1 Evaluation Board

The Broadcom® WICED evaluation board (Figure 1) incorporates a WICED module and circuitry to enable the programming and debugging of applications on the module.
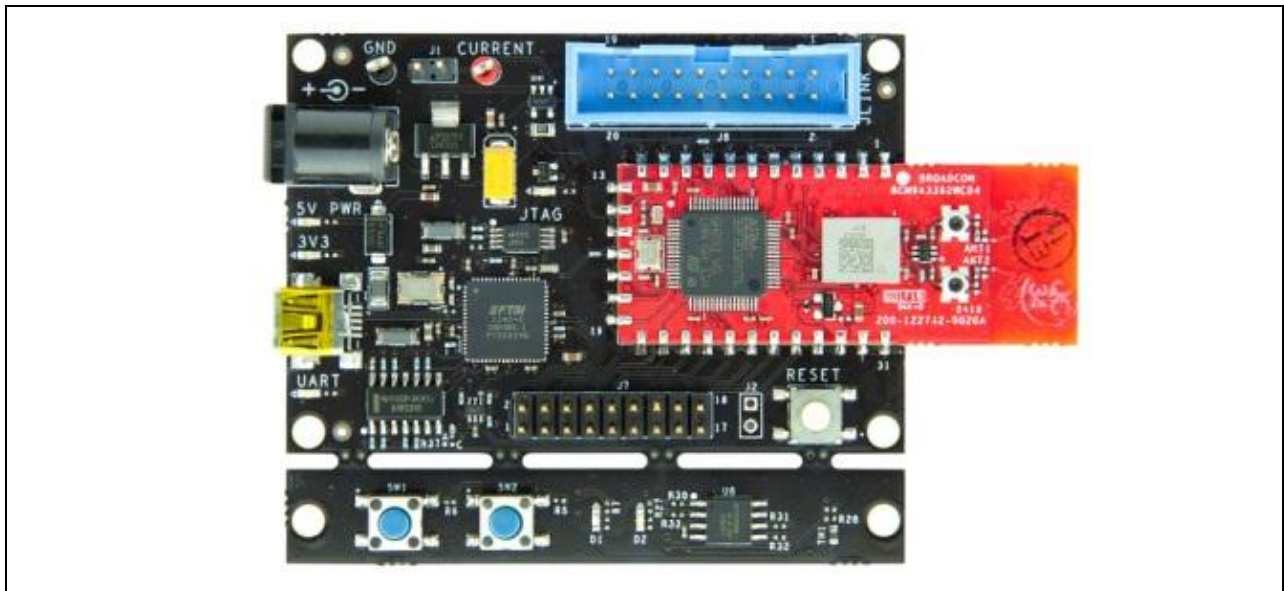

Figure 1: WICED Evaluation Board with BCM943362WCD4 WICED Module

2.2 Software Development Kit

The WICED SDK includes:

- WICED software development tools:
  - Utilities and OS drivers to support development in the Windows® environment
  - WICED software stack, development tools and demonstration applications
  - ThreadX and FreeRTOS Real-Time Operating Systems (RTOS), and NetX/NetXDuo IPv4/IPv6 and LwIP IPv4 TCP/IP network stack implementations.
  - Embedded security libraries including TLS & HTTPS
  - A WICED Wi-Fi driver and API
  - WICED Application Framework (WAF)
  - Manufacturing test and Iperf applications to enable system performance testing
- WICED API Documentation, (this) Quickstart Guide and related documents

2.3 Directory Structure

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

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

WICED-SDK Directories	Directory Contents
Apps	Demo applications and snippets, test utilities, and WAF components
Doc	API Documentation, Reference Documentation, Schematics
Drivers	Windows® USB Drivers for the WICED evaluation board
include	WICED API function prototypes, constants and defaults
Include/platforms	Platform description and I/O definitions
Library	Daemons, servers, protocols and peripheral libraries
Resources	Resources used by the WICED webserver incl. HTML, images, styles, etc.
Tools	Toolchain including compiler, debugger, and other utilities/scripts
Wiced	Wiced core components: RTOS, TCP stack, Security & Platform definitions
Wiced/WWD	The WICED Wi-Fi Driver

2.3.1 Hardware and Software Requirements

- The WICED SDK runs on 32- and 64-bit versions of Microsoft Windows® XP and Win7, Mac OS X 10.5 or later, and 32/64-bit Linux.
- The SDK is distributed as a standalone 7zip zipfile suitable for all operating systems, and bundled together with the WICED Integrated Development Environment as an executable installer. The installer is only provided for Windows® and OS X operating systems.
- The development computer requires a single USB port to connect to the WICED evaluation board.
- A terminal emulation application such as PuTTY (Windows®) or CoolTerm (OS X).

Note : The 7zip extraction utility is available from 7-zip.org. 7zip is needed if you plan to extract the standalone WICED SDK 7zip archive. The standard Windows® zipfile extraction utility may silently corrupt the SDK archive during the extraction process, do **NOT** use it!

3   Setting Up the WICED SDK

The WICED SDK is available for download from the Broadcom® WICED website or from the Broadcom® Customer Support Portal.

To use the SDK, the following steps must be performed:

1. Download and install the WICED SDK and IDE
2. Choose and compile an application
3. Load the application into the WICED module on the WICED evaluation board
4. Run the application
5. Debug the application

This section provides instructions for Step 1 for Windows® and Mac OS X. Steps 2-5 are discussed in Section 4.

> **Note** : DO **NOT** PLUG THE WICED EVALUATION BOARD INTO THE DEVELOPMENT COMPUTER PRIOR TO INSTALLING THE WICED DEVELOPMENT SYSTEM. DOING SO MAY CAUSE THE INCORRECT DRIVER TO LOAD!

3.1   Instructions for Windows® XP / Windows® 7

3.1.1   Install the WICED SDK and IDE

The WICED SDK is provided as a self-installing executable file. Double-click the **Wiced-SDK-2.3.x-IDE-Installer.exe** file to begin the installation. A Setup window similar to the following appears.
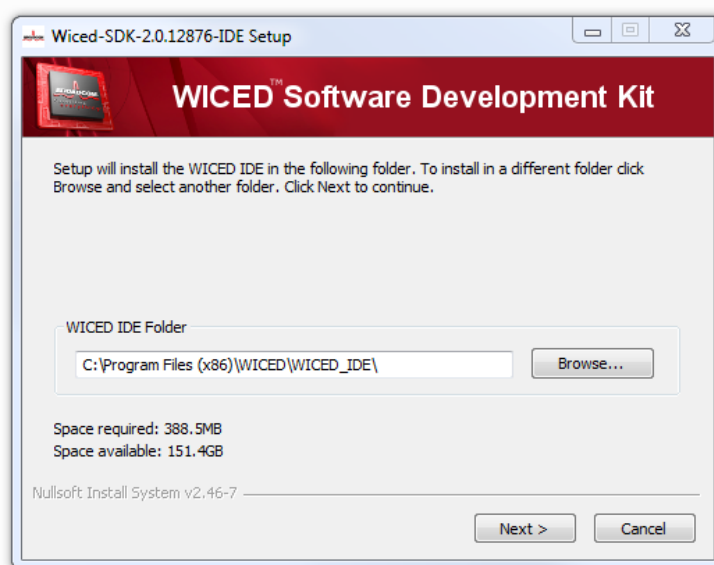
Choose the installation folder for the WICED IDE and click **Next**, then choose the installation workspace folder for the WICED SDK and click **Install**. Once the installation completes, click **Finish** to immediately start the WICED IDE, or deselect the **Start WICED IDE now** checkbox and then click **Finish** to exit.

3.1.2 Connect the WICED Evaluation Board

The USB interface connects the WICED evaluation board to the PC. The USB interface provides +5V power as well as individual programming/debug and UART interfaces to the microcontroller on the WICED module. A separate +5V power supply is NOT needed.

The WICED evaluation board has two logical USB devices: a USB-JTAG device and a USB-UART device. A driver for the WICED evaluation board was automatically installed during the SDK installation process.

Plug the WICED evaluation board into the development PC with a USB cable, the driver automatically loads.

3.1.3 Verifying Driver Installation

To verify that driver installation has completed successfully, do the following:

1. Open the **Device Manager** (right-click **My Computer**, select **Properties**).

2. In the **System** Properties window, select **Hardware**, **Device Manager**.

   a. The **WICED USB Serial Port** is listed under **MY-LAPTOP\Ports (COM & LPT)** as shown in the adjacent screen capture (shaded yellow).

   b. The **WICED USB JTAG Port** is listed under **MY-LAPTOP\WICED USB JTAG Devices** as shown in the adjacent screen capture (shaded blue).

In the screen capture, the **Device Manager** window identifies the WICED USB Serial COM port as **COM21**. The assigned port number varies between systems.

Troubleshooting

If an error occurred during the automatic driver installation process, the driver may be manually installed from the <WICED-SDK>\Drivers directory.
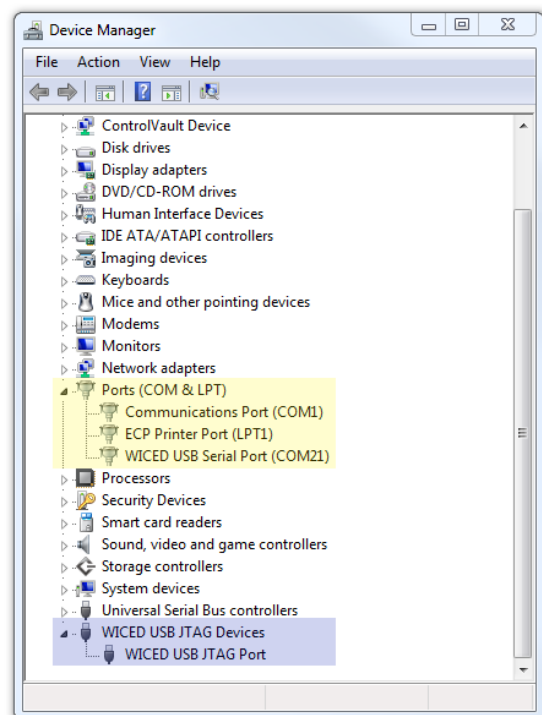
If the WICED evaluation board still does not appear in the **Device Manager**, verify the 3V3 LED is turned ON and/or replace the USB cable.

3.2 Instructions for Mac OS X

3.2.1 Install the WICED SDK and IDE

The WICED SDK is provided as an OS X package installer. Double-click the **Wiced-SDK-2.3.x.pkg** file to begin the installation. A Setup window similar to the following appears.

Click **Continue** once to read the welcome note, then click **Continue** again and select installation permissions. After installation permissions are selected, click **Continue** and then click **Install**. Enter your account password if asked, and follow the prompts to complete the installation.

3.2.2 Connect the WICED Evaluation Board

The USB interface connects the WICED evaluation board to the Mac. The USB interface provides +5V power as well as individual programming/debug and UART interfaces to the microcontroller on the WICED module. A separate +5V power supply is NOT needed.

The WICED evaluation board has two logical USB devices: a USB-JTAG device and a USB-UART device. A driver for the WICED evaluation board is included with the OS X operating system.

Plug the WICED evaluation board into the Mac with a USB cable, the driver automatically loads. Verify the WICED evaluation board 3V3 LED is ON indicating the board is powered. If the LED is not on, replace the USB cable and try again.

4   Using the WICED IDE

4.1 Build and Download a Sample Application

On a Windows® PC, start the WICED IDE by selecting **START** > **All Programs** > **Broadcom** > **WICED IDE**. On a Mac running OS X, use the Finder application to locate the **Applications** > **WICED** directory, then double click the **WICED IDE** shortcut.

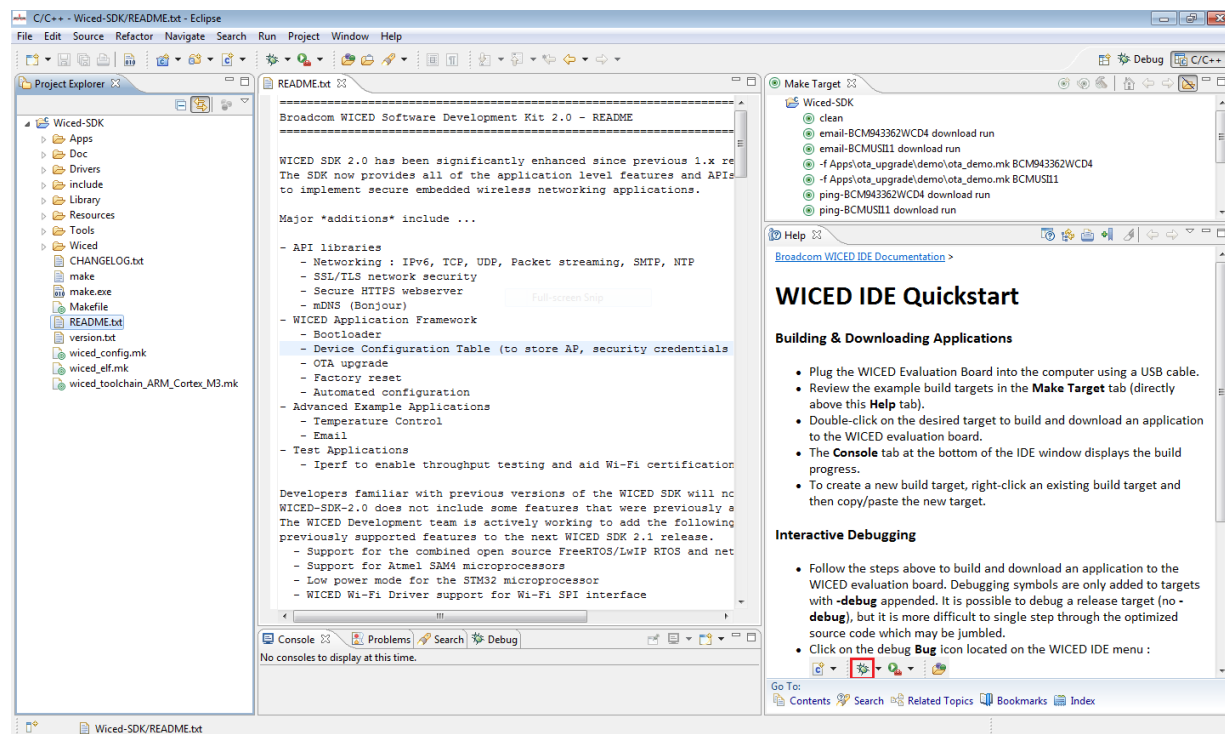After startup, the WICED IDE looks similar to the screen capture shown in Figure 2.

Figure 2. The WICED Integrated Development Environment

The **Help** tab to the right of the WICED IDE window describes how to build and download the sample applications shown in the **Make Target** tab (located above the **Help** tab). Multiple build targets are pre-configured for a number of sample applications and at least two WICED hardware platforms including 'BCM943362WCD4' and 'BCM9WCDUSI09'.

For this example, we will build and download the 'ping' application for the 'BCM943362WCD4' WICED module on the WICED Evaluation Board. If the module on your evaluation board is different, follow the instructions to modify or copy one of the 'ping' build targets to match your hardware platform.

Double-click the **snip.ping-BCM943362WCD4** target to build the ping application; the IDE console window displays the build progress. The build target is appended with the 'download' and **'**run' options, these options tell the toolchain to download the firmware and run the application after the build completes.

The build output looks similar to the following:

```
**** Build of configuration Default for project Wiced-SDK ****
"C:\My Documents\Wiced-SDK\make.exe" snip.ping-BCM943362WCD4 download run
Applying changes made to: Wiced/Wiced.mk
*****************************************************************************
The target Wi-Fi MAC address is defined in <Wiced-SDK>/generated_mac_address.txt
Ensure each target device has a unique address!
*****************************************************************************
Building Bootloader
Building App
Making .gdbinit
Making DCT image
Compiling App_Ping
Compiling Platform_BCM943362WCD4
Compiling ThreadX
Compiling NetX_Duo
Compiling Wiced
Compiling Lib_http_server
Compiling Lib_Gedday
Compiling STM32F2xx
Compiling WWD_ThreadX_Interface
Compiling Wiced_ThreadX_Interface
Compiling WWD_NetX_Duo_Interface
Compiling Wiced_NetX_Interface
Compiling Supplicant_uSSL
Compiling Lib_dns_redirect_daemon
Compiling Lib_dns
Compiling WWD_for_SDIO_ThreadX
Compiling Supplicant_wps_brcm
Compiling SPI_Flash_Library_BCM943362WCD4
Compiling Lib_dhcp_server
Compiling Wiced_Wifi_image
Making snip_ping-BCM943362WCD4.elf
Making snip_ping-BCM943362WCD4.bin
Build complete

snip_ping-BCM943362WCD4
--------------------------------|---------|---------|
                                |         |  Static |
            Module              |  Flash  |   RAM   |
--------------------------------+---------+---------|
App                             |    7778 |    1012 |
Bootloader                      |     133 |       0 |
dhcp_server                     |    1111 |       0 |
dns                             |    3028 |      44 |
Gedday                          |    4942 |     721 |
http_server                     |    1175 |       0 |
libc                            |   50186 |    2632 |
Networking                      |    3719 |   29320 |
NetX                            |   52818 |     644 |
NetX-Duo - Interfaces & Stacks  |       0 |    5896 |
Platform                        |     786 |       0 |
RAM Initialisation              |    2464 |       0 |
Startup Stack & Link Script fill|     475 |     844 |
STM32 library                   |    9736 |    2602 |
Supplicant - SSL/TLS            |   59894 |     424 |
ThreadX                         |   11692 |    1604 |
Wi-Fi Firmware                  |  189776 |       0 |
Wiced                           |    2161 |     750 |
WWD                             |   13346 |    1035 |
--------------------------------+---------+---------|
TOTAL (bytes)                   |  415872 |   47532 |
--------------------------------|---------|---------|
Downloading Bootloader ...
Download complete
Downloading Application ...
Download complete
Downloading DCT ...
Download complete
Resetting target
Target running
```

During firmware download, a blue LED labeled 'JTAG' on the WICED evaluation board illuminates to indicate that a JTAG firmware download is in progress.

If the WICED evaluation board is not recognized by the programming tools, it may be necessary to disconnect and then reconnect the board to the computer before trying again. The following message indicates there was an error with the download process:

```
"**** OpenOCD failed - ensure you have installed the driver from the drivers
directory, and that the debugger is not running **** In Linux this may be due to
USB access permissions. In a virtual machine it may be due to USB passthrough
settings ****"
Resetting target
make: *** [run] Error 1
```

4.2 Run an Application

This section assumes you have successfully completed Section 4.1, and the ping application is running on the WICED module.

Several methods are available to verify the ping application is working. After power-on-reset, the ping application prints status messages to the USB UART of the WICED evaluation board.

To verify printing, follow the instructions in Appendix A to setup a terminal application such as PuTTY (Windows®) or CoolTerm (OS X). Start the terminal application and connect to the WICED evaluation board then press the reset button on the board. The following text appears:

```
Platform BCM943362WCD4 initialised
Started ThreadX v5.5
Initialising NetX v5.6
Creating Packet pools
Starting Wiced v2.3.0
Joining : YOUR_AP_SSID
Successfully joined : YOUR_AP_SSID
Obtaining IPv4 address via DHCP
Setting IPv6 link-local address
IPv4 network ready IP: 192.168.1.100
Pinging 192.168.1.1 every 1000ms with a 900ms timeout.
Ping Reply : 3 ms
Ping reply : 1 ms
...
```

> **Note** : A description of the ping application is provided in the header comments of the ping.c source file located in the WICED SDK at <WICED-SDK>/Apps/snip/ping/ping.c.

4.3 Debug an Application

The WICED development system supports single-step thread-aware interactive application debugging with the WICED IDE. Brief instructions to start debugging an application are provided in the WICED IDE **Help** tab. The following example demonstrates how to start debugging the 'ping' application.

1. Setup a breakpoint:

    a. Using the **Project Explorer** tab on the left of the IDE, navigate to the 'WICED-SDK/Apps/snip/ping' directory. Double-click on 'ping.c' to open the source file in the WICED IDE editor window.

    b. Scroll to the 'application_start' function and locate the call to 'wiced_init' (at approximately line 180).

    c. Double-click in the column to the left of the 'wiced_init' function to place a break point at this function (alternately right-click in the column and select 'Toggle Breakpoint'). A green dot appears in the column.

2. Setup the 'ping' debug build target: In the WICED IDE **Make Target** tab, right click on the 'ping-BCM943362WCD4 download run' build target and click **Edit** (alternately, copy-paste and then edit a new build target). Change the build target to 'ping-BCM943362WCD4-debug download'.

3. Build and download the debug firmware image: Double-click the newly created 'ping' debug target, the debug firmware image builds and downloads to the 'BCM943362WCD4' module on the WICED evaluation board.

4. Launch the debugger: Click the green bug icon on the WICED IDE toolbar (or press F11 on Windows® systems). The debugger starts. If the WICED IDE displays a **Confirm Perspective Switch** dialogue box, click **Yes** to show the debug view.

    To confirm the debugger is running, verify the blue JTAG LED on the WICED evaluation board is flashing, and that the WICED IDE looks similar to the following screen capture.
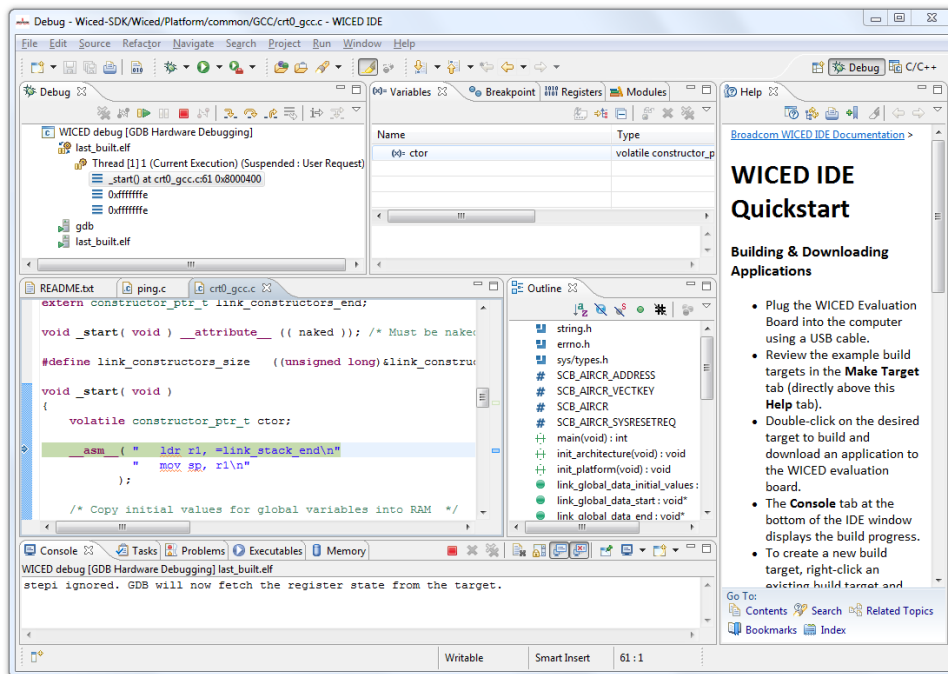
5. Run to a breakpoint: To run the application to the 'wiced_init' breakpoint configured in Step 1, click the yellow/green pause/play button in the **Debug** tab. The **Debug** tab control buttons are highlighted in yellow in the following screen capture.
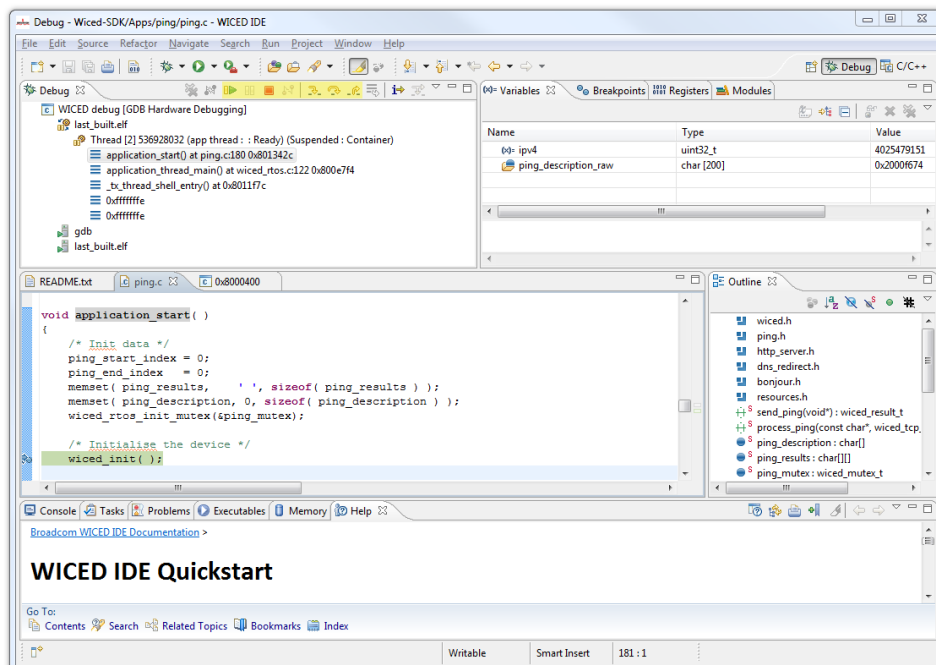
6. View a breakpoint: When the debugger halts at a breakpoint, the WICED IDE does not automatically switch to the current thread as the context for the debugger. It is necessary to manually check the current task in the running thread is selected before analyzing debug information.

To find the breakpoint, click on the 'application_start()' function under Thread[2] in the **Debug** tab. The 'ping.c' source file opens and the wiced_init function is highlighted in green to show where the program halted.

7. Step program execution: Step Into, Step Over and Step Return options are available in the WICED IDE **Run** menu. Alternately, Step shortcut icons are provided in the **Debug** tab, and on Windows® Systems, by pressing **F5**, **F6** and **F7**, respectively.

8. Stop debugging: To stop debugging, click the square red stop button in the **Debug** tab.

Note. If the debugger fails to launch, it may be necessary to terminate an existing debug process. On Windows®, press Ctrl-Alt-Delete to open the Windows® **Task Manager**, then select the **Processes** tab. Search for, and terminate, all 'arm-none-eabi-gdb' processes.

4.4 What now?

Now you have a basic understanding of how to compile, download and debug a WICED application, we recommend that you try to build and run each of the snippet and demo applications provided in the WICED SDK Applications directory.

Each snippet application provides a relatively simple example demonstrating how to use a particular API or feature of the WICED SDK. The more sophisticated demo applications combine multiple APIs and features to demonstrate advanced functionality.

To help you use the WICED IDE, Appendix B contains some quick hints and tips that will save you time navigating around the WICED code base. Read these now, then move onto trying WICED applications.

We hope you enjoy using the WICED Development System!

   -- The WICED Development Team

APPENDIX A – Configuring a Terminal Application

The following instructions describe how to obtain and install a serial terminal application for use on computers running a Windows® or OS X operations system. Broadcom recommends using PuTTY for Windows® systems and CoolTerm for OS X systems, however other equivalent applications may work equally well.

Ensure the WICED Development System is already installed on the computer, then plug the WICED evaluation board into the computer using a USB cable before continuing.

Set Up PuTTY for Windows®

PuTTY is available as a free download from http://putty.org. Download and install PuTTY.

The following procedure describes how to establish a UART serial interface between PuTTY and the WICED evaluation board.

1. Start the PuTTY application. The PuTTY Configuration window opens. Set the configuration options as follows:
   - Category: **Serial**
   - Serial line to connect to: *type in the COM port that was assigned after the USB and serial port drivers were installed. Refer to Step 2(a) in Section 3.1.3.*
   - Speed (baud): **115200**
   - Data bits: **8**
   - Stop bits: **1**
   - Parity: **None**
   - Flow control : **None**
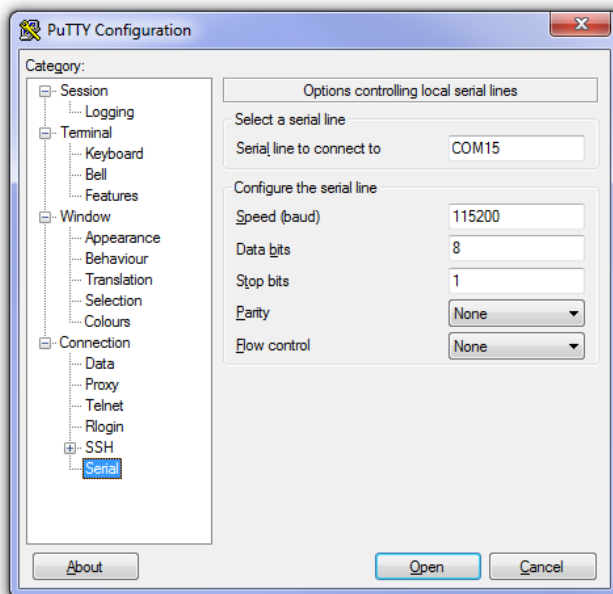
2. In the Category pane, select **Session**.
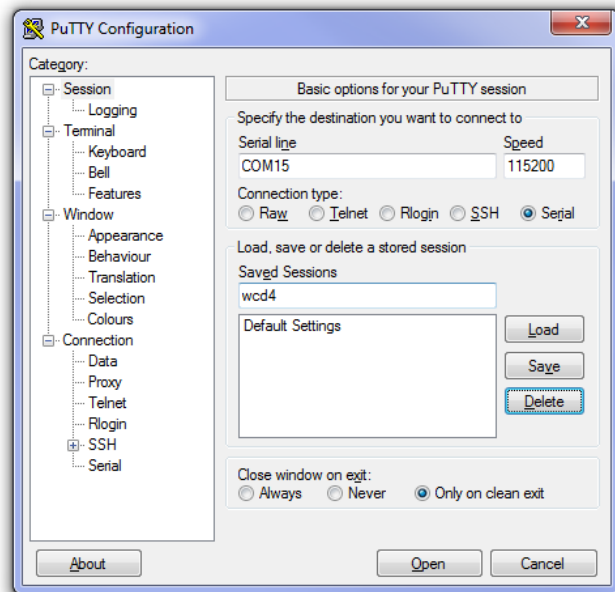
3. Under the **Connection type**, select the **Serial** option, then click **Open**.

4. A blank terminal window opens with the selected COM port specified in the window title. If the specified COM port is incorrect or unavailable, PuTTY displays an error message as shown in the following screen capture. If this happens, verify the correct COM port has been selected and try again.
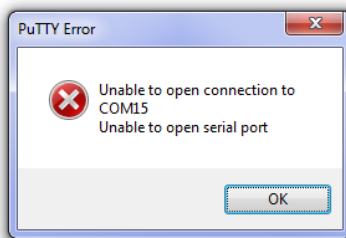
5. Assuming the ping application is running on the WICED evaluation board, press the reset button on the board to view application prints during the boot and run process.
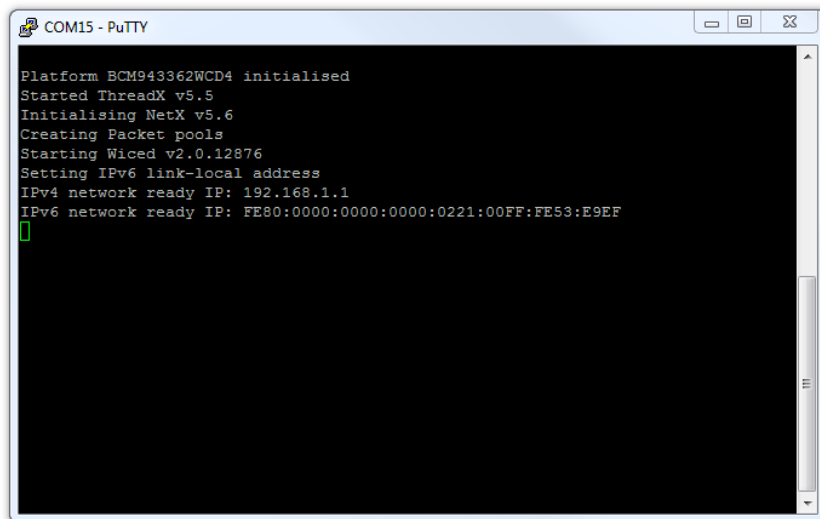

Set Up CoolTerm for OS X

CoolTerm is available as a free download from http://freeware.the-meiers.org/CoolTermMac.zip. Download and install CoolTerm.

The following procedure describes how to establish a UART serial interface between CoolTerm and the WICED evaluation board.

1. Start the CoolTerm application and click the **Options** menu icon. The CoolTerm Configuration window opens. Set the Serial Port configuration options as follows:

   - Port: **usbserial-XXXXXXXX**
   - Baudrate: **115200**
   - Data bits: **8**
   - Parity: **none**
   - Stop bits: **1**
   - Flow control : Deselect all options
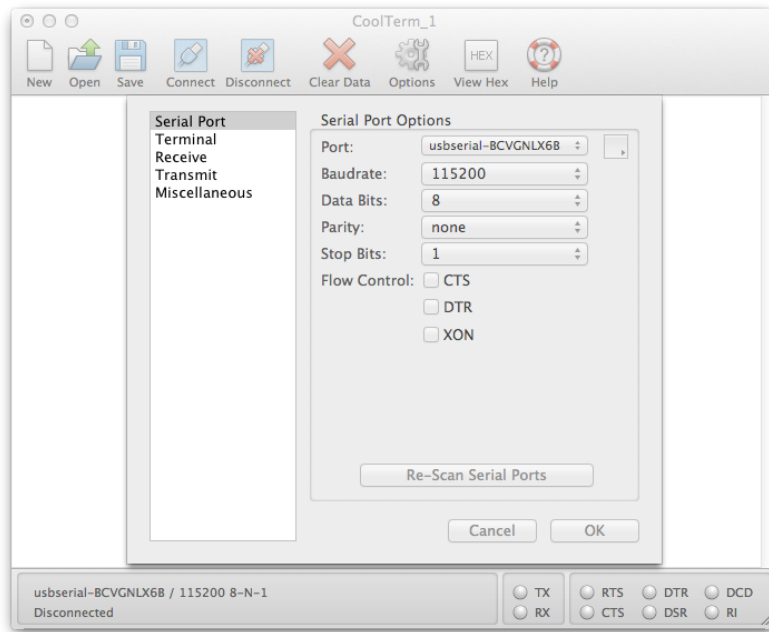
2. Click **OK**.

3. Click the **Connect** menu icon. The CoolTerm application connects to the WICED evaluation board.

4. Assuming the ping application is running on the WICED evaluation board, press the reset button on the board to view application prints during the boot and run process.
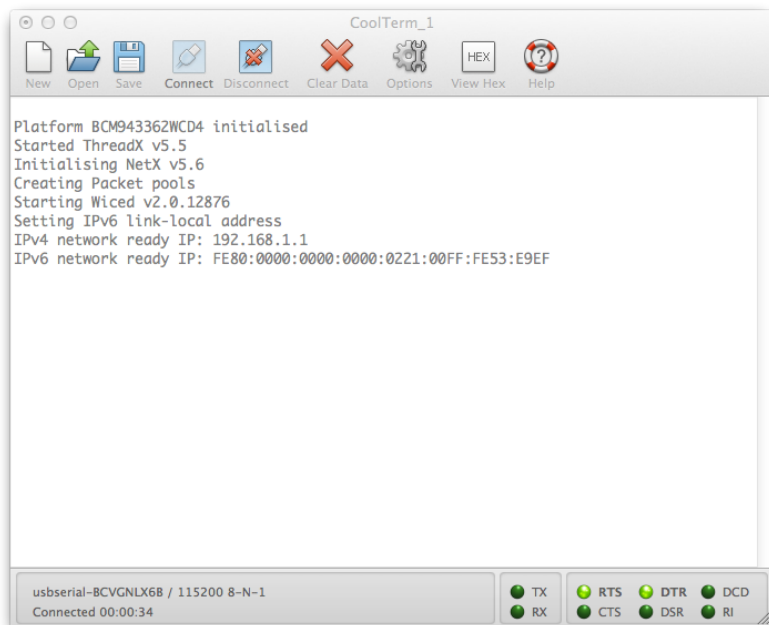
APPENDIX B – WICED IDE Hints & Tips

Hints

1. The **Help** tab (and any other tab) may be click-dragged to any window pane if desired to customize the IDE layout.

2. To revert to the C/C++ perspective (rather than the Debug perspective), click the C/C++ icon in the top-right corner of the window.

Shortcuts

A useful cheat-sheet outlining short cuts for the WICED IDE (Eclipse) is included in the <WICED-IDE>/Readme directory or on 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-SDK tree for 'my_variable':
    - **Left-click** the root WICED-SDK folder in the Project Explorer pane
    - Press **CTRL-H** (on Windows®)
    - Enter the text 'my_variable' into the File Search tab (regular expressions work too!)
    - Click **Search**

- Search for a 'c' source element (variable, function, enum, etc)
    - Open a 'c' source file eg. <WICED-SDK>/Apps/snip/ping/ping.c
    - Press **CTRL-SHIFT-T**
    - Start typing an element eg. 'wiced_time_'
    - Suggestions appear in the popup window

- Navigation / Edit History
    - To return to the previous file, press **CTRL-Left** (arrow)
    - To return to the previous file, press **CTRL-Right** (arrow)

Appendix C – Importing the WICED SDK into Eclipse

This advanced HOW-TO is **ONLY** for customers that need to import the WICED SDK into an existing version of Eclipse. Instructions in this guide apply to the Juno release of Eclipse CDT. The guide is tailored to a Windows® installation, but may be used as a reference for other operating systems.

Broadcom provides a WICED IDE installer that automatically performs all of the following steps and configures the WICED SDK development environment for you. Before continuing, please strongly consider using the WICED IDE installer instead!

> **Note**: Broadcom does not provide direct customer support for Eclipse setup. Usage and setup instructions are provided as a helpful guide.

The following instructions assume a Broadcom WICED BCM943362WCD4 evaluation board is plugged into your computer, and that you have installed drivers for the board. Drivers are provided in the WICED-SDK/Drivers directory.

1 Importing the WICED SDK into Eclipse

Eclipse[1] CDT (C/C++ Development Tooling) provides a fully functional C/C++ Integrated Development Environment based on the Eclipse platform. Complete details are available at http://eclipse.org/cdt/

> **Note**: Eclipse is a self-contained package based on Java®. It does not use the standard Windows installation procedure; instead, it runs directly from the directory into which it has been extracted. If your development PC does not already have an installation of Java, go to http://java.com to obtain and install Java before installing Eclipse.

1.1 Setup Instructions to Build and Download Applications

This subsection contains the procedure to set up Eclipse to build and download an application using the WICED development system target : `snip.scan-BCM943362WCD4-debug`

1.1.1 Preparation

1. Eclipse uses a workspace to manage projects: create a workspace directory and copy the extracted WICED development system directory to it.
2. Download and install the **Juno** version of Eclipse CDT from
   http://eclipse.org/cdt/downloads.php
3. Launch Eclipse.

---

[1] Go to www.eclipse.org for more information on the Eclipse IDE.

1.1.2 Build

1. In the Eclipse Workspace Launcher window, enter the **workspace directory**, click **OK**, and wait for the main Eclipse program window to open. If a splash screen tab is displayed, close it by clicking the **X** in the upper-left corner.

2. From the **File** menu, select **New Makefile Project with Existing Code**.

3. In the **Import Existing Code** window, click **Browse** and navigate to the WICED-SDK directory.

4. Click **Finish;** the WICED-SDK appears in the Eclipse **Project Explorer** pane on the LEFT of the screen.

5. In the Eclipse program window, in the RIGHT pane, at the top next to the **Outline** tab, select the **Make Target** tab (green bull's eye icon)

6. Right-click on the WICED-SDK root directory name, and then, from the shortcut menu, select **New**.

7. In the **Create Make Target | Target Name** field, type:

```
snip.scan-BCM943362WCD4-debug download run
```

8. Click **OK** to dismiss the window.

9. In the **Make Target** tab :

    a. Left-click the Open Folder icon

    b. Left-click the arrow next to WICED-SDK to view the target that was just created

10. Double-click the newly created scan make target. The scan application compiles and downloads to the WICED evaluation board. Progress is shown in the Console tab, in the lower window pane.

The screen capture in Figured C1 shows an example of the Eclipse program window with the Apps/snip/scan.c source together with the results of a JTAG download session in the console window.
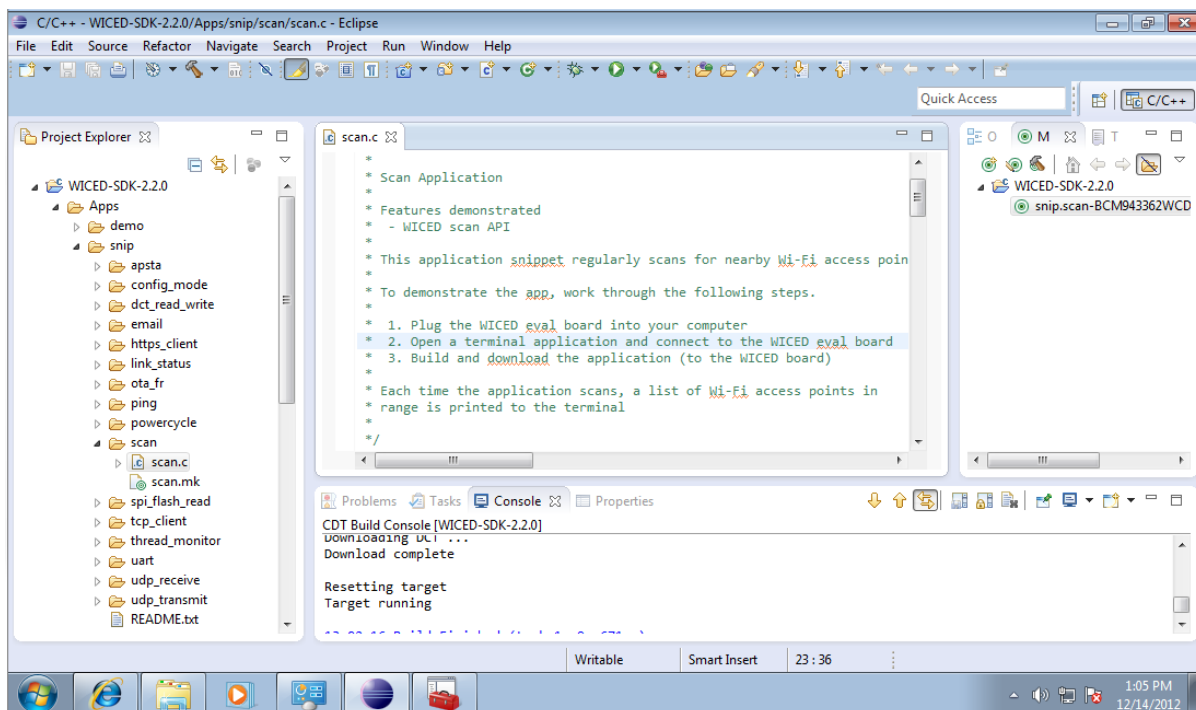

Figure C1: Eclipse – Apps/snip/scan.c source plus the results of a JTAG download

1.2 Setup Instructions for Application Debugging

To enable hardware debugging, it is necessary to install an additional Eclipse CDT component. The following instructions detail the necessary steps:

1. From the **Help** menu, select **Install New Software**, and then click the **Available Software Sites** link.

2. Click **Add**:

    a. In the **Add Site** window, in the **Name** field, type **JunoCDT**

    b. In the **Location** field, type http://download.eclipse.org/tools/cdt/releases/juno and then click **OK**.

    c. In the **Preferences** window, from the **Available Software Sites** list, locate the **JunoCDT** site, verify that it is selected, and then click **OK**.

3. In the **Work with** list:

    a. Select **JunoCDT**, wait for the CDT features to update, and then click the arrow to **expand** the **CDT Optional Features** list.

    b. Select the **C/C++ GDB Hardware Debugging** option and then click **Next** twice

    c. Accept the license agreement (click **Accept**) and then click **Finish** to install the hardware debugger. Eclipse will restart when installation is complete.

4. After Eclipse restarts, in the **Project Explorer** pane (left side of the Eclipse window), click to select the WICED-SDK-2.3.0 directory structure.

5. From the **Run** menu select **Debug Configurations**.

    a. In the Debug Configurations window, double-click **GDB Hardware Debugging** to automatically create a submenu item named **Wiced-SDK-2.3.0 Default**.

    b. In the **C/C++ Application** field, type : `build/eclipse_debug/last_built.elf`

    c. Select the **Disable auto build** radio button

6. On the Debugger tab:

    a. Use the Browse button next to the **GDB Command** field to locate the gdb application `<WICED-SDK-2.3.0>\Tools\ARM_GNU\bin\Win32\arm-none-eabi-gdb.exe` to the GDB Command text

    b. From the **JTAG Device** list, select **Generic TCP/IP**

    c. In the **Hostname or IP address field**, type **localhost**

    d. In the **Port Number** field, type **3333**

    e. Select **Force thread list update on suspend**

7. On the **Startup** tab:

    a. In the **Initialization Commands** panel

        i. clear the **Reset and Delay (seconds)** option

        ii. add the following text in the text pane
             `add-symbol-file build/eclipse_debug/last_bootloader.elf 0x8000000`

    b. In the **Load Image and Symbols** panel, clear the **Load image** option

    c. In the **Run Commands** pane, type : `stepi`

8. On the **Common** tab, select the **Debug** option, click **Apply**, and the click **Close**.

9. In the Eclipse main window, open the list attached to the green bug icon
    a. Click **Wiced-SDK-2-2.0 Default**
    b. In the **Confirm Perspetive Switch** window, click **Yes**.

To verify that the debugger is running correctly, confirm that the following line of source code is highlighted green:

```
__asm__( "   ldr r1, =link_stack_end\n"
```

The blue JTAG LED on the WICED evaluation board flashes each time the debugger communicates with the WICED module (for example, when single-stepping through code).

With the debugger running, the Eclipse window looks similar to Figure C2.
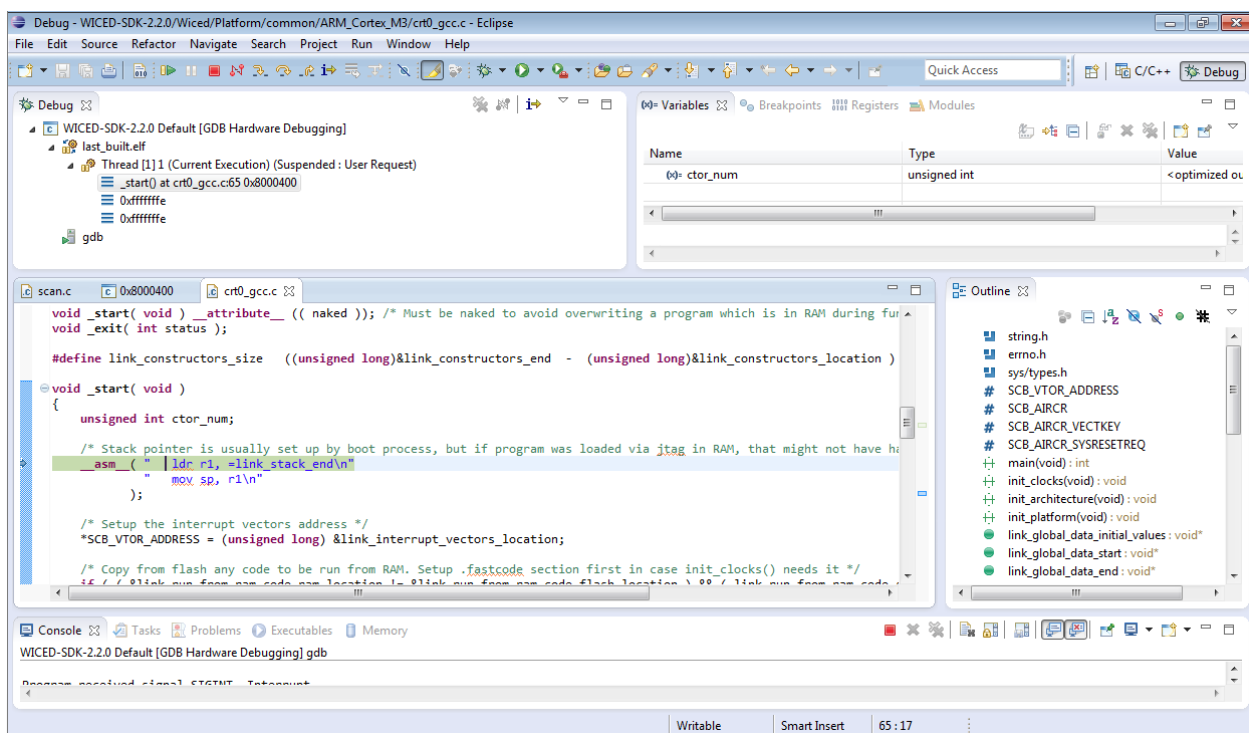
Figure C2: Eclipse – Debugging the Apps/snip/scan application

Connecting
everything®

BROADCOM CORPORATION

5300 California Avenue
Irvine, California, 92677
© 2013 by BROADCOM CORPORATION. All rights reserved.

Phone : +1-949-926-5000
Fax: +1-949-926-5203
E-mail: info@broadcom.com
Web: www.broadcom.com

WICED-QSG202-R   April 19, 2013