ANCYWICED003

WICED™ Studio 4
WICED Quick Start Guide For ZB CYW20729

Associated Part Family: CYW207x9

This document describes how to use WICED Studio to develop ZigBee applications.

Contents

1 Introduction

This document provides detailed instructions to set up the Cypress® Wireless Internet Connectivity for Embedded Devices (WICED; pronounced "wick-ed") Studio Development System for ZigBee.

WICED™ Studio 4 supports application development using a WICED development board (CYW9207x9WCDEVAL). The development system is compatible with the Windows, OS X, and Linux operating systems. This document describes the software components included in the WICED Studio Development System and provides instructions for compiling WICED sample applications using the WICED Studio Integrated Development Environment (IDE).

The instructions in this document must be completed before the WICED development board can be used.

Note: This document applies to **WICED Studio 4 and WICED ZigBee 20729 modules**

2 IoT Resources and Technical Support

Cypress provides a wealth of data at http://www.cypress.com/internet-things-iot to help you to select the right IoT device for your design, and quickly and effectively integrate the device into your design. Cypress provides customer access to a wide range of information, including technical documentation, schematic diagrams, product bill of materials, PCB layout information, and software updates. Customers can acquire technical documentation and software from the Cypress Support Community website (http://community.cypress.com/).

3 WICED Studio Development System Overview

WICED Studio for ZigBee comprises a development board, a Software Development Kit (SDK), and the Eclipse Integrated Development Environment (IDE).

3.1 WICED Development Board

The Cypress WICED development board (CYW9207x9WCDEVAL) incorporates a Cypress CYW20729 and additional circuitry to enable application programming, debugging, and evaluation.

The CYW9207x9WCDEVAL board can be used for feature evaluation, debugging, and developing ZigBee applications for designs based on the CYW20729.

Figure 1 shows the front of the CYW9207x9WCDEVAL board.

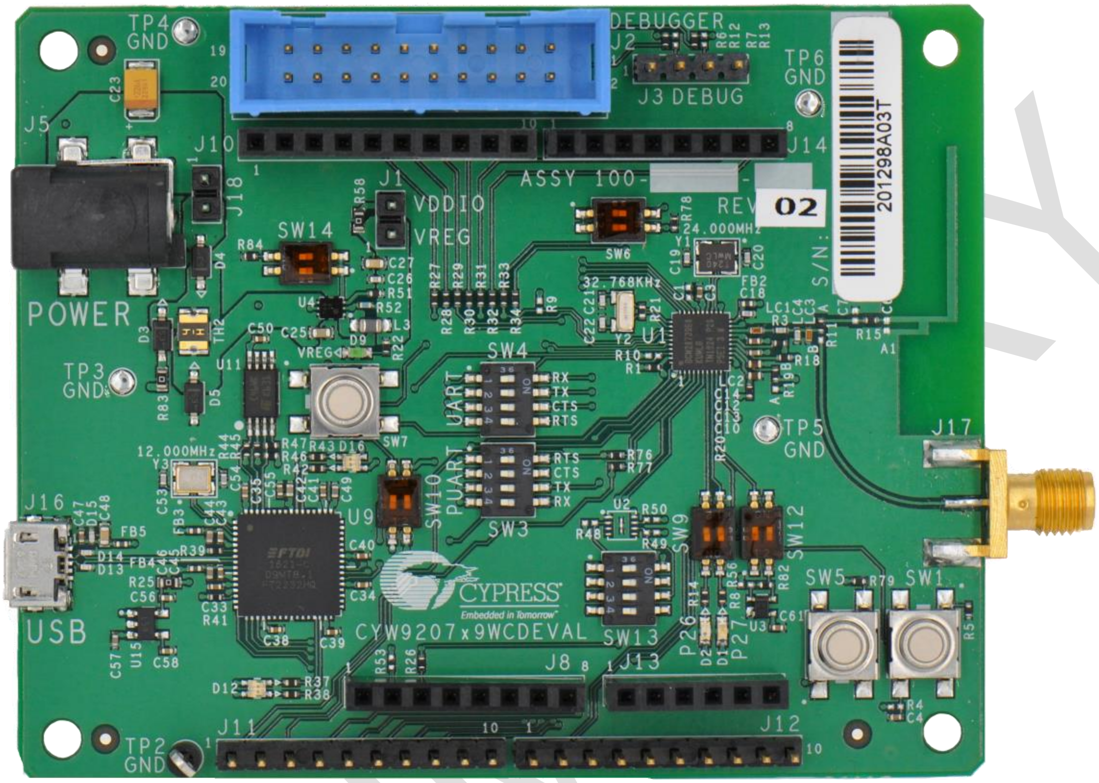


Figure 1: CYW9207x9WCDEVAL WICED Development Board

3.2 Software Development Kit

WICED Studio includes:

- The ZigBee software stack, including ZHA (ZigBee Home Automation)
- A generic profile-level API
- Drivers to access onboard peripherals including UART, SPI, I^{2}C, ADC, PWM, etc.
- Reference applications for the devices with clusters defined by the ZigBee Alliance
- WICED Studio API documentation
- Utilities to support development in Windows, OS X, and Linux environments

3.3 Directory Structure

WICED Studio may support multiple different types of WICED modules depending on installed components. Table 1 is an overview of the directory structure of WICED Studio for the CYW20729 ZigBee device.

Table 1: WICED Studio for ZigBee Directory Structure

WICED Studio Directory	***Directory Contents***
apps	Sample applications
build	Output files of built applications

WICED Studio Directory	***Directory Contents***
doc	API, reference documentation, and schematics
Drivers	USB drivers for the development board
include	WICED API function prototypes and definitions
platforms	Configuration files and information for supported hardware platforms
tools	Common utilities used by the IDE build processes
WICED	WICED core components
wiced_tools	Tool chain including compiler, download tool, and other utilities and scripts

3.4 Hardware and Software Requirement

- WICED Studio runs on 32- and 64-bit versions of Microsoft Windows, OS X, and Linux
- WICED Studio is distributed as a bundle with the Eclipse IDE as executable installers for the Windows, OS X, and Linux operating systems
- The development computer requires a single USB port to connect to a WICED development board

3.5 Development Process

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

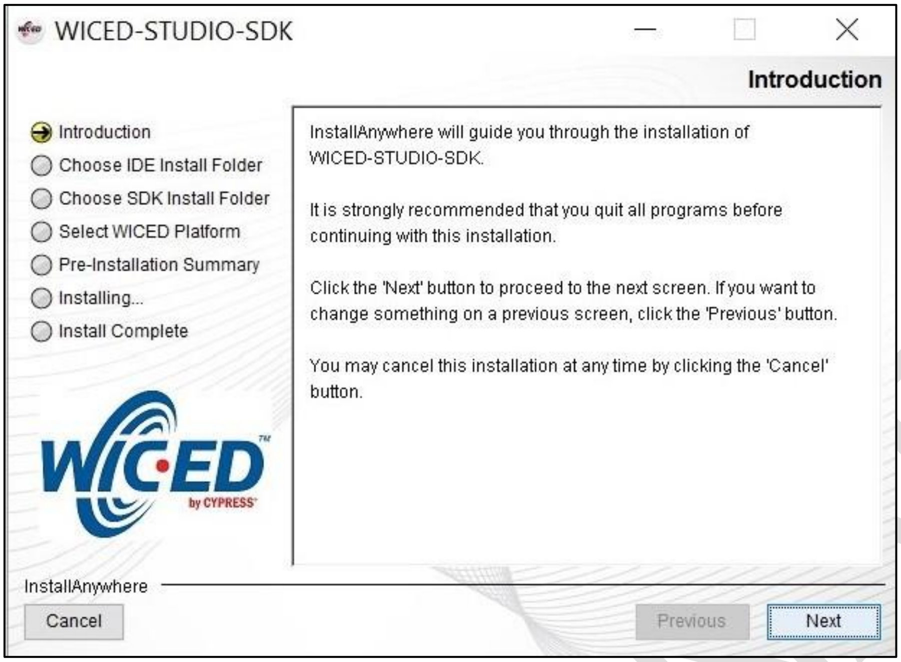
1. Download and install WICED Studio 4 (see “Install WICED Studio”)
2. Connect the WICED Development board (see “Connect the WICED Development Board”)
3. Create and load an application (see “Build and Load a Sample Application”)

4 Setting up WICED Studio

Download WICED Studio 4 from the Cypress WICED Products website or Cypress Customer Support Portal.

4.1 Install WICED Studio

The WICED Studio distribution is provided as a self-installing executable files. Double-click the **WICED-Studio-4.0.x.x-IDE-Installer.exe** file to begin the installation. A setup window similar to the screenshot below is displayed.



WICED-STUDIO-SDK

Introduction

InstallAnywhere will guide you through the installation of WICED-STUDIO-SDK.

It is strongly recommended that you quit all programs before continuing with this installation.

Click the 'Next' button to proceed to the next screen. If you want to change something on a previous screen, click the 'Previous' button.

You may cancel this installation at any time by clicking the 'Cancel' button.

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 Select WICED Platform window, select “20729-B0 ZigBee device” to use and click **Next.**
5. In the Pre-Installation Summary window, click **Next** to install using the current selections. (Click **Previous** one or more times to modify the selections.)

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

4.2 Connect the WICED Development Board

Figure 2 shows the CYW9207x9WCDEVAL WICED development board. The figure shows call-outs to the ports, switches, and switch positions relevant to this document.

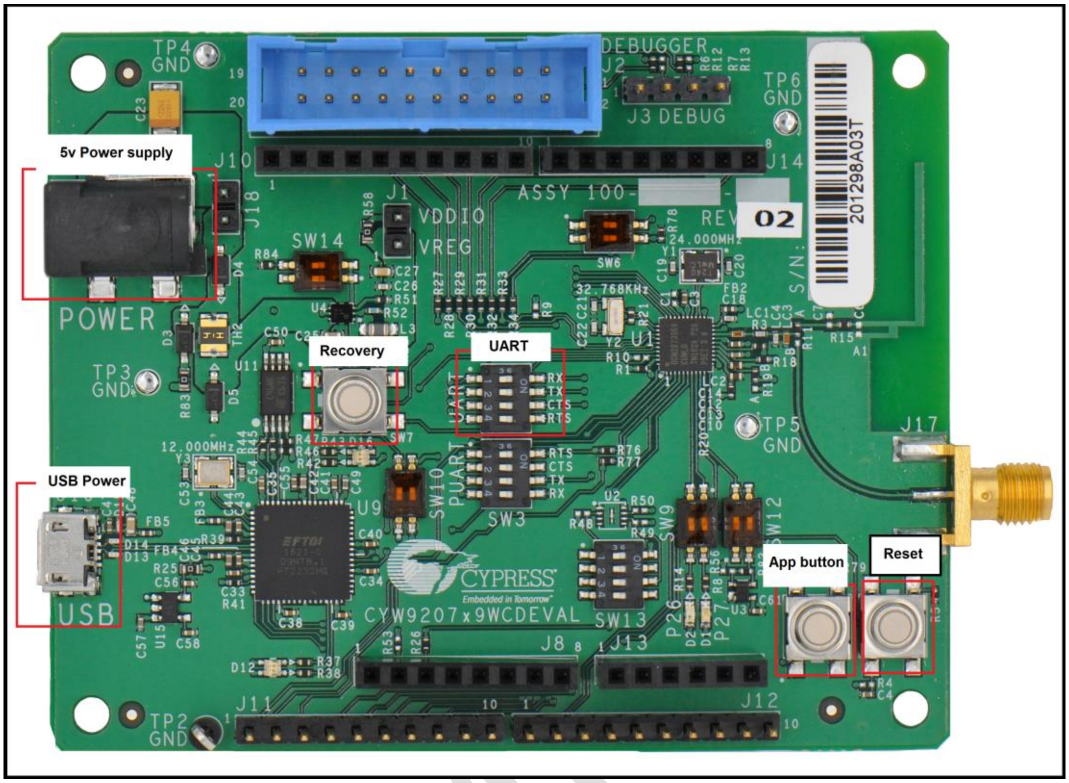


Figure 2: CYW9207x9WCDEVAL WICED Development Board Settings

The Micro-USB connector (J16) supports UART connections and provides +5V power to the board. The barrel connector J5 can also be used to power the board when UART connectivity to the PC is not required.

Perform the following steps before verifying driver installation:

1. Set **SW2** to position 3.
2. Set the PIN 2 of the **SW4** mini-switch to **ON** so that the UART logs can be output to PC via USB.
3. Set the PIN 2 of the **SW6** min-switch to **ON** so that the button (SW5) can be used as a light switch in the later sample.
4. Connect J16 of the WICED development board to the development PC with a USB cable. The USB UART driver will load automatically.

The LEDs called out in Figure 3 serve the following purposes:

- D9 (green) indicates that 3.3V power is on.
- D12 (red, blinking) indicates that HCI UART is in use, for example, the firmware is being downloaded to the development bard.
- D16 (green, blinking) indicates that PUART is in use, for example, the logs are being sent to PC via USB.
- D1 (blue) and D2 (red) are for application to use.

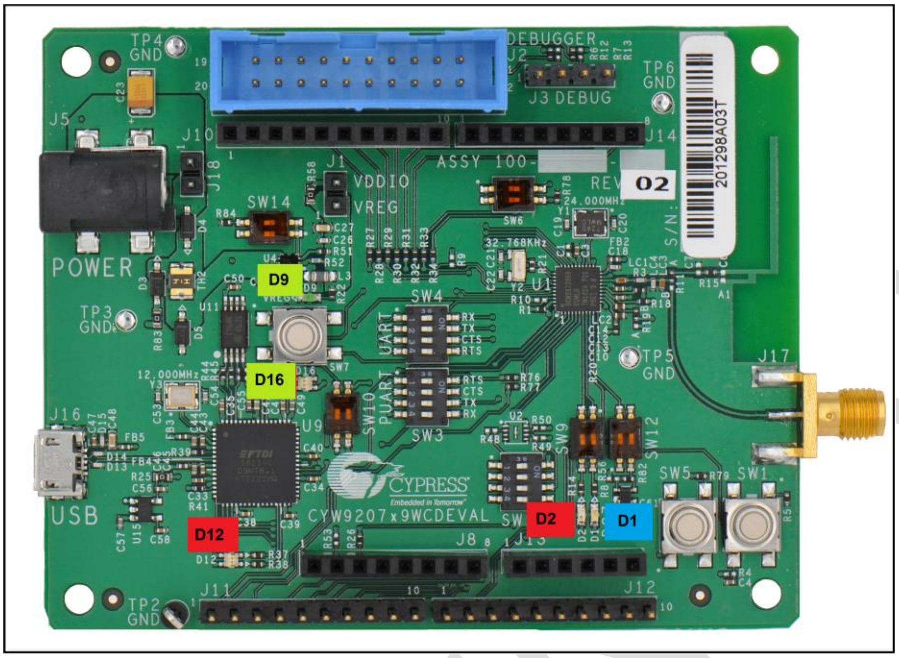
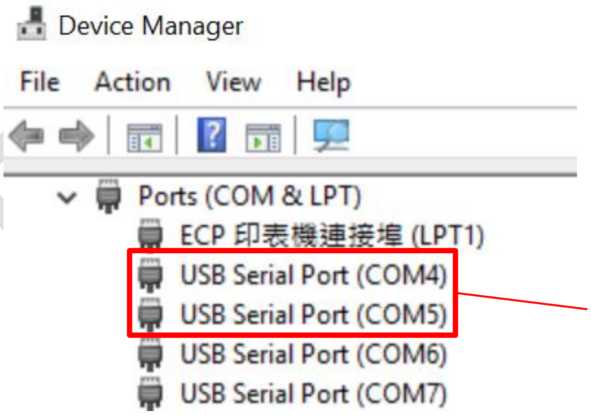


Figure 3: LEDs on a WICED Development Board

4.3 Verify Driver Installation

To verify that driver installation is complete:

1. On a Windows system, open the **Device Manager** (right-click **My Computer**, select **Properties**, and then select **Device Manager**).
2. In the **Device Manager** window, verify that two new USB serial COM ports are listed under **Ports (COM & LPT)**. **Note:** In the below screenshot, the Device Manager identifies the new (WICED development board) USB serial COM ports as COM4 and COM5. Assigned port numbers vary among systems.



Ports (COM & LPT)
ECP 印表機連接埠 (LPT1)
USB Serial Port (COM4)
USB Serial Port (COM5)
USB Serial Port (COM6)
USB Serial Port (COM7)

New serial COM ports for the WICED device. (COM4 and COM5 are only examples.)

Note: If an error occurs during driver installation, download new drivers from Windows Update. Verify you have an Internet connection, disconnect and then reconnect the board and wait for the drivers to automatically install. If the new WICED development board serial COM ports do not appear in the Device Manager after reinstalling drivers via Windows Update, then the drivers can be manually installed from the Drivers folder of the SDK installation. If the error persists, then check all switch settings (see "Connect the WICED Development Board") on the board and/or replace the USB cable.

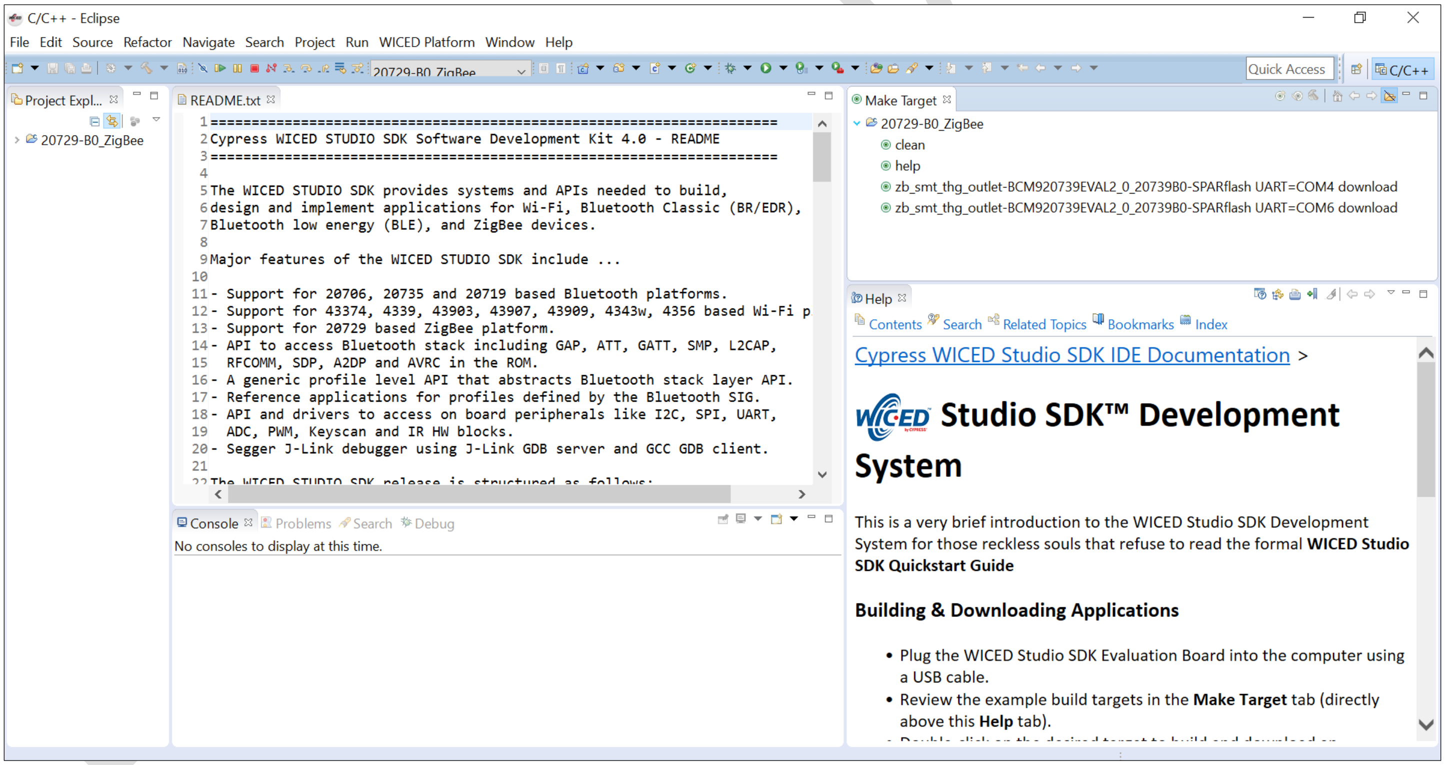
5 Using the WICED Studio IDE

This section describes how to:

- Use the WICED Studio IDE to create application build targets for the WICED development board.
- Download applications to the board.
- Verify that the application running on the board is working correctly using a Samsung SmartThings Hub.

5.1 Build and Load a Sample Application

Start the IDE by selecting **START > All Programs > Cypress > WICED-Studio** or double-click the **WICED-Studio** icon on desktop. The WICED Studio IDE looks similar to the screenshot shown in Figure 4.



```
====================================================================
Cypress WICED STUDIO SDK Software Development Kit 4.0 - README
====================================================================

The WICED STUDIO SDK provides systems and APIs needed to build,
design and implement applications for Wi-Fi, Bluetooth Classic (BR/EDR),
Bluetooth low energy (BLE), and ZigBee devices.

Major features of the WICED STUDIO SDK include ...

- Support for 20706, 20735 and 20719 based Bluetooth platforms.
- Support for 43374, 4339, 43903, 43907, 43909, 4343w, 4356 based Wi-Fi p
- Support for 20729 based ZigBee platform.
- API to access Bluetooth stack including GAP, ATT, GATT, SMP, L2CAP,
  RFCOMM, SDP, A2DP and AVRC in the ROM.
- A generic profile level API that abstracts Bluetooth stack layer API.
- Reference applications for profiles defined by the Bluetooth SIG.
- API and drivers to access on board peripherals like I2C, SPI, UART,
  ADC, PWM, Keyscan and IR HW blocks.
- Segger J-Link debugger using J-Link GDB server and GCC GDB client.

The WICED STUDIO SDK release is structured as follows:
```

No consoles to display at this time.

20729-B0_ZigBee
- clean
- help
- zb_smt_thg_outlet-BCM920739EVAL2_0_20739B0-SPARflash UART=COM4 download
- zb_smt_thg_outlet-BCM920739EVAL2_0_20739B0-SPARflash UART=COM6 download

Cypress WICED Studio SDK IDE Documentation >

WICED Studio SDK™ Development System

This is a very brief introduction to the WICED Studio SDK Development System for those reckless souls that refuse to read the formal **WICED Studio SDK Quickstart Guide**

Building & Downloading Applications

- Plug the WICED Studio SDK Evaluation Board into the computer using a USB cable.
- Review the example build targets in the **Make Target** tab (directly above this **Help** tab).

Figure 4: WICED Studio IDE

The **Help** pane in the lower-right corner of the IDE (see Figure 4) 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 a build targets that is preconfigured for a sample application that run on the CYW9207x9WCDEVAL development boards.

The example below shows how to build and run the sample application, which can then connect the WICED development board to a Samsung SmartThings Hub. The SmartThings application should be installed on a phone before doing the following example.

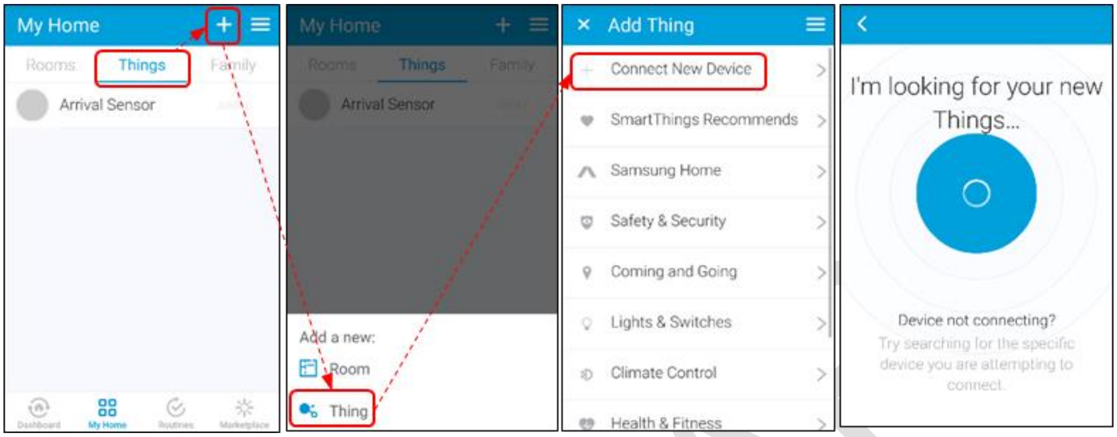
1. Make sure the WICED development board is connected to the PC and located close to the SmartThings Hub.
2. Check the UART port after WICED development board is connected to the PC (see "Verify Driver Installation")
3. Right-Click the target **zb_smt_thg_outlet-BCM920739WCDEVAL-SPARflash UART=COM4 download** in the **Make Target** pane and select **Edit** to change the UART port number.
4. Double-click the modified **zb_smt_thg_outlet-BCM920739WCDEVAL-SPARflash UART=COMxx download** target start the build. The IDE console pane (bottom center of the IDE window) will display the build and download progress.

   The build output looks similar to the following:

```
09:09:53 **** Build of configuration Default for project WICED ****
make zb_smt_thg_outlet-BCM920739WCDEVAL-SPARflash download
Compiling spar_setup.c
Compiling app_init.c
Compiling zb_ao_smt_thg_outlet.c
.
.
.
Generating lib_installer.c
Linking target ELF
OK, made elf.
   text    data     bss     dec     hex         filename
381012     9964    3148  394124   6038c        ../../build/zb_smt_thg_outlet-
BCM920739WCDEVAL-rom-flash-Wiced-release/A_20739B0-zb_smt_thg_outlet-rom-flash-spar.elf
Call to zb_smt_thg_outlet_spar_crt_setup @ 00551001
OK, made C:/BTInfo/WICED/WICED_repo4/Wiced-SDK/20729-
B0_ZigBee/WICED/wpan/../../build/zb_smt_thg_outlet-BCM920739WCDEVAL-rom-flash-Wiced-
release/A_20739B0-zb_smt_thg_outlet-rom-flash-spar.cgs. MD5 sum is:
75d325bb63a99f20f45f809453b3b9fa *../../build/zb_smt_thg_outlet-CM920739WCDEVAL-rom-
flash-Wiced-release/A_20739B0-zb_smt_thg_outlet-rom-flash-spar.cgs


Converting CGS to HEX...
Conversion complete

Creating OTA images...
Conversion complete
OTA image footprint in NV is 21935 bytes

Downloading application...
Download complete

Application running

09:11:40 Build Finished (took 1m:46s.379ms)
```

5. Because the sample target includes the ***download*** option, the tool will download the firmware to development board automatically when the build is complete.

6. After finishing the downloading, start the SmartThings App on your phone and add a new device in the tab “Things” as following.

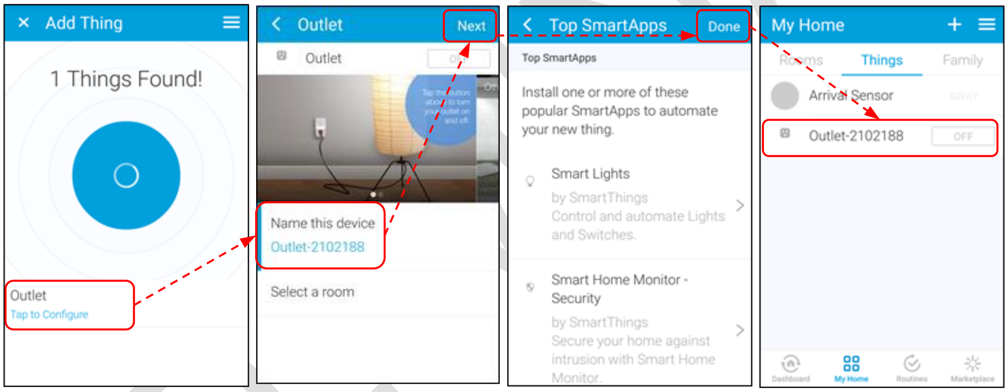


I'm looking for your new Things...

Device not connecting?

Try searching for the specific device you are attempting to connect.

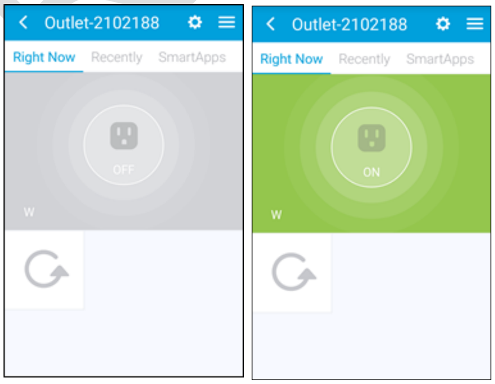
7. When the SmartThings starts to search a new device, press the reset button (SW1 in Figure 2) on the development board. Then LED D1 on development board starts blinking (see Figure 3), which means the development board is now trying to connect a ZigBee network. After a while, the LED D1 stops blinking and SmartThings app shows that it detects a new device as following.



Install one or more of these popular SmartApps to automate your new thing.

8. Click **Tap to Configure** for the detected device. You can also re-name the device. After Click **Next** and **Done**, the development board is successfully connected to the SmartThings Hub and is added into the list, which can be controlled by SmartThings app.

9. Now, you can use SmartThings App to turn the LED D2 (see Figure 3) ON or OFF. While the SmartThings App shows the device is ON, the LED D2 on the development board is ON as well. You can also press the button on development board (SW5 in Figure 2) to toggle the state.



5.2 Viewing Traces

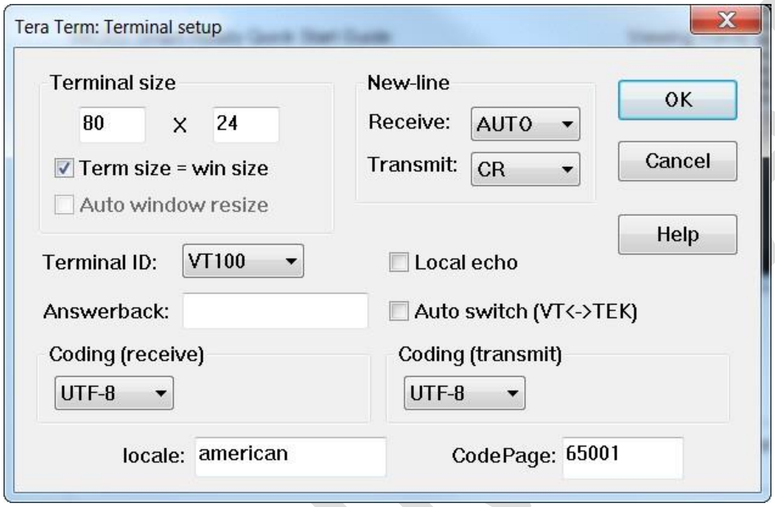
To view the traces:

1. Compile and download an application to the WICED development board as described in “Build and Load a Sample Application”.
2. Wait for the *Build Finished* message to display in the console window.
3. Use a terminal emulation program (see “Viewing Traces Using a Terminal Emulation Program” below).

5.2.1 Viewing Traces Using a Terminal Emulation Program

To view traces with a terminal emulation program (such as Tera Term):

1. Start the terminal emulation program.
2. Set the **Terminal ID** to **VT100** and New-Line **Receive** to **AUTO**.



Tera Term: Terminal setup

Terminal size: 80 X 24

☑ Term size = win size

☐ Auto window resize

New-line

Receive: AUTO

Transmit: CR

Terminal ID: VT100

Answerback:

☐ Local echo

☐ Auto switch (VT<->TEK)

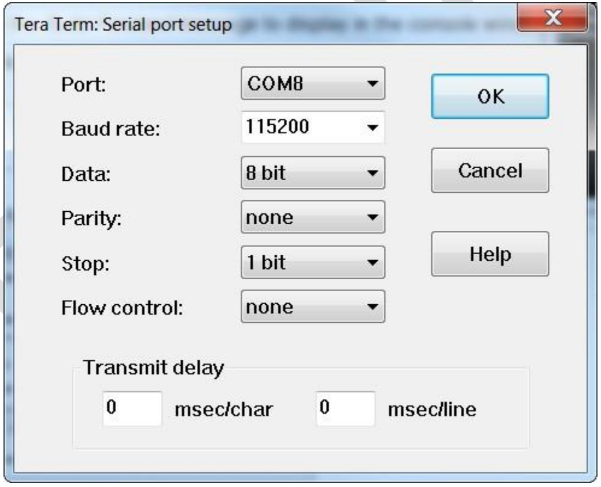
Coding (receive): UTF-8

Coding (transmit): UTF-8

locale: american

CodePage: 65001

3. In the terminal emulator, initiate a connection with the following serial port settings:



Tera Term: Serial port setup

Port: COM8

Baud rate: 115200

Data: 8 bit

Parity: none

Stop: 1 bit

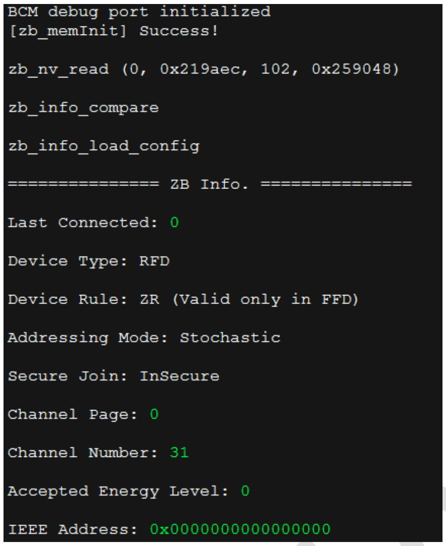
Flow control: none

Transmit delay

0 msec/char 0 msec/line

4. Press the **Reset** button (see Figure 2) on the WICED development board to view the application start-up messages.

Figure 5 shows the application start-up messages for a sample application.



```
BCM debug port initialized
[zb_memInit] Success!

zb_nv_read (0, 0x219aec, 102, 0x259048)

zb_info_compare

zb_info_load_config

=============== ZB Info. ===============

Last Connected: 0

Device Type: RFD

Device Rule: ZR (Valid only in FFD)

Addressing Mode: Stochastic

Secure Join: InSecure

Channel Page: 0

Channel Number: 31

Accepted Energy Level: 0

IEEE Address: 0x0000000000000000
```

Figure 5: Example Application Start-Up Messages

5.3 What’s NEXT?

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

The header of the main source file of every application provides additional information on the features demonstrated by the application and the usage model.

“IDE Hints & Tips” contains hints and tips about navigating the WICED Studio code base.

We hope you enjoy using the WICED Studio Development System!

-- The WICED Development Team

Appendix A: IDE Hints & Tips

A.1 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.

A.2 Shortcuts

A useful cheat-sheet outlining shortcuts for the WICED Studio IDE (Eclipse) is included online at:

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

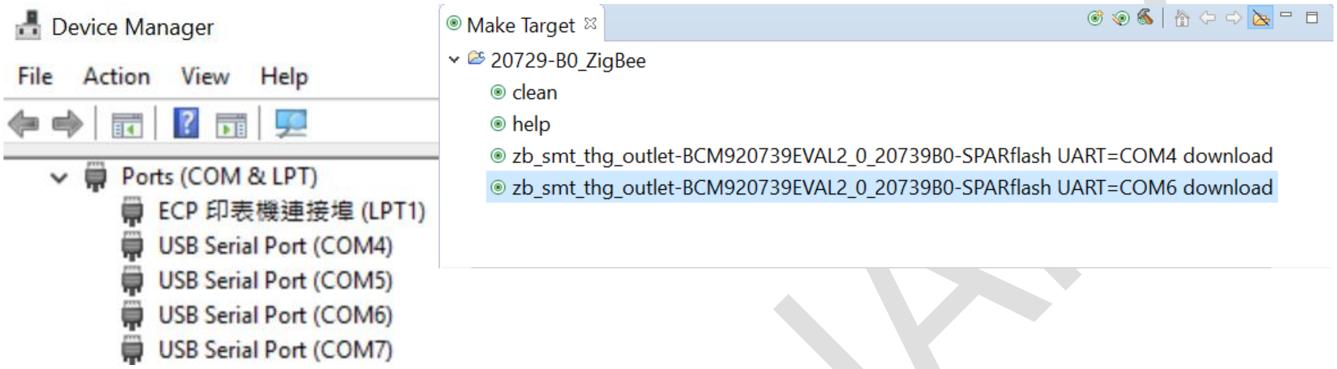
Particularly useful keystrokes are listed below:

- General search: to search the WICED-Studio tree for a variable:
  1. **Click** the root WICED-Studio folder in the Project Explorer pane.
  2. Press **CTRL-H** (on Windows).
  3. In the File Search tab, enter the variable name (regular expressions work too).
  4. Click **Search.**
- Search for a *C* source element (variable, function, enum, etc.).
  1. Open a *C* source file, for example: `<20729-B0_ZigBee/apps/zb_smt_thg_outlet/zb_ao_smt_thg_outlet.c.`
  2. Press **CTRL-SHIFT-T**
  3. Start typing an element, for example, *ZB_AO_SMT_THG_t*
  4. Suggestions appear in the pop-up window.
- Press **ALT-Left** (arrow) and **ALT-Right** (arrow) to navigate between open files.

Appendix B: Multiple WICED Development Boards

Multiple boards 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` parameters.

Figure 6 shows two WICED development boards connected to a PC and appropriate targets to build and download the hello sensor application.



Ports (COM & LPT)
ECP 印表機連接埠 (LPT1)
USB Serial Port (COM4)
USB Serial Port (COM5)
USB Serial Port (COM6)
USB Serial Port (COM7)

20729-B0_ZigBee
clean
help
zb_smt_thg_outlet-BCM920739EVAL2_0_20739B0-SPARflash UART=COM4 download
zb_smt_thg_outlet-BCM920739EVAL2_0_20739B0-SPARflash UART=COM6 download

Figure 6: Configuration for Two WICED Development Boards

Appendix C: Connecting to Linux Platforms

An additional step is required when connecting a WICED board to a computer running Linux. On common Linux distributions, the serial UART ports (usually `/dev/ttySx or /dev/ttyUSBx devices`) belong to the root user and to the dialout group. Standard users are not allowed to access these devices.

An easy way to allow the current user access to Linux's serial ports is by adding the user to the dialout group. This can be done using the following command:

```
$sudo usermod -a -G dialout $USER
```

Note: For this command to take effect, the user must log out and then log back in.

Appendix D: Recovering a Corrupted Board

If you need to recover the CYW9207x9WCDEVAL board from a bricked state, then refer to Figure 3 and follow these steps:

1. Verify the steps sections 4.2 and 4.3.
2. Press and hold **SW7** the recover button.
3. Press and hold **SW1** the reset button.
4. Release **SW1**, wait for one second and then release **SW7**.
5. Continue the download process as in section 5.1.

Document History

Document Title: ANCYWICED003 – WICED Quick Start Guide for ZB CYW20729

Document Number: 001-CYWICED003

Revision	Submission Date	Description of Change
*B	11/15/16	Update to latest CYW9207x9WCDEVAL board
*A	10/19/16	Update to WCDEVAL board information, add Appendix D
**	09/07/16	Initial release - preliminary

References

[1] 20729 Data Sheet

[2] CYW9207x9WCDEVAL Hardware User Manual (ANCYWICED007)

Worldwide Sales and Design Support

Cypress maintains a worldwide network of offices, solution centers, manufacturer’s representatives, and distributors. To find the office closest to you, visit us at Cypress Locations.

Cypress Products

ARM® Cortex® Microcontrollers	cypress.com/arm
Automotive	cypress.com/automotive
Clocks & Buffers	cypress.com/clocks
Interface	cypress.com/interface
Internet of Things	cypress.com/iot
Lighting & Power Control	cypress.com/powerpsoc
Memory	cypress.com/memory
PSoC	cypress.com/psoc
Touch Sensing	cypress.com/touch
USB Controllers	cypress.com/usb
Wireless/RF	cypress.com/wireless

PSoC® Solutions

PSoC 1 | PSoC 3 | PSoC 4 | PSoC 5LP

Cypress Developer Community

Forums | Projects | Videos | Blogs | Training | Components

WICED IoT

Uniting CDC and WICED Solutions

Technical Support

cypress.com/support

PSoC is a registered trademark and WICED and PSoC Creator are trademarks of Cypress Semiconductor Corporation. All other trademarks or registered trademarks referenced herein are the property of their respective owners.

Cypress Semiconductor
198 Champion Court
San Jose, CA 95134-1709

Phone	: 408-943-2600
Fax	: 408-943-4730
Website	: www.cypress.com

TO THE EXTENT PERMITTED BY APPLICABLE LAW, CYPRESS MAKES NO WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, WITH REGARD TO THIS DOCUMENT OR ANY SOFTWARE OR ACCOMPANYING HARDWARE, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. To the extent permitted by applicable law, Cypress reserves the right to make changes to this document without further notice. Cypress does not assume any liability arising out of the application or use of any product or circuit described in this document. Any information provided in this document, including any sample design information or programming code, is provided only for reference purposes. It is the responsibility of the user of this document to properly design, program, and test the functionality and safety of any application made of this information and any resulting product. Cypress products are not designed, intended, or authorized for use as critical components in systems designed or intended for the operation of weapons, weapons systems, nuclear installations, life-support devices or systems, other medical devices or systems (including resuscitation equipment and surgical implants), pollution control or hazardous substances management, or other uses where the failure of the device or system could cause personal injury, death, or property damage (“Unintended Uses”). A critical component is any component of a device or system whose failure to perform can be reasonably expected to cause the failure of the device or system, or to affect its safety or effectiveness. Cypress is not liable, in whole or in part, and you shall and hereby do release Cypress from any claim, damage, or other liability arising from or related to all Unintended Uses of Cypress products. You shall indemnify and hold Cypress harmless from and against all claims, costs, damages, and other liabilities, including claims for personal injury or death, arising from or related to any Unintended Uses of Cypress products.

Cypress, the Cypress logo, Spansion, the Spansion logo, and combinations thereof, WICED, PSoC, CapSense, EZ-USB, F-RAM, and Traveo are trademarks or registered trademarks of Cypress in the United States and other countries. For a more complete list of Cypress trademarks, visit cypress.com. Other names and brands may be claimed as property of their respective owners.