CYPRESS

Embedded in Tomorrow™

ANCYWICED002

WICED™ Studio 4
WICED Quick Start Guide For BT CYW20735

Associated Part Family: CYW20735B0

This document describes how to use WICED Studio to develop Bluetooth applications for CYW20735 devices.

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 Bluetooth Basic Rate (BR) and Low Energy (LE) device based on CYW20735.

WICED™ Studio 4 supports application development using a WICED development board (CYW20735WCDEVAL). 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 Bluetooth 20735 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 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 (CYW20735WCDEVAL) incorporates a Cypress CYW20735 and additional circuitry to enable application programming, debugging, and evaluation.

The CYW20735WCDEVAL board can be used for feature evaluation, debugging, and developing BR/LE applications for designs based on the CYW20735.

Figure 1 shows the front of the CYW20735WCDEVAL board.

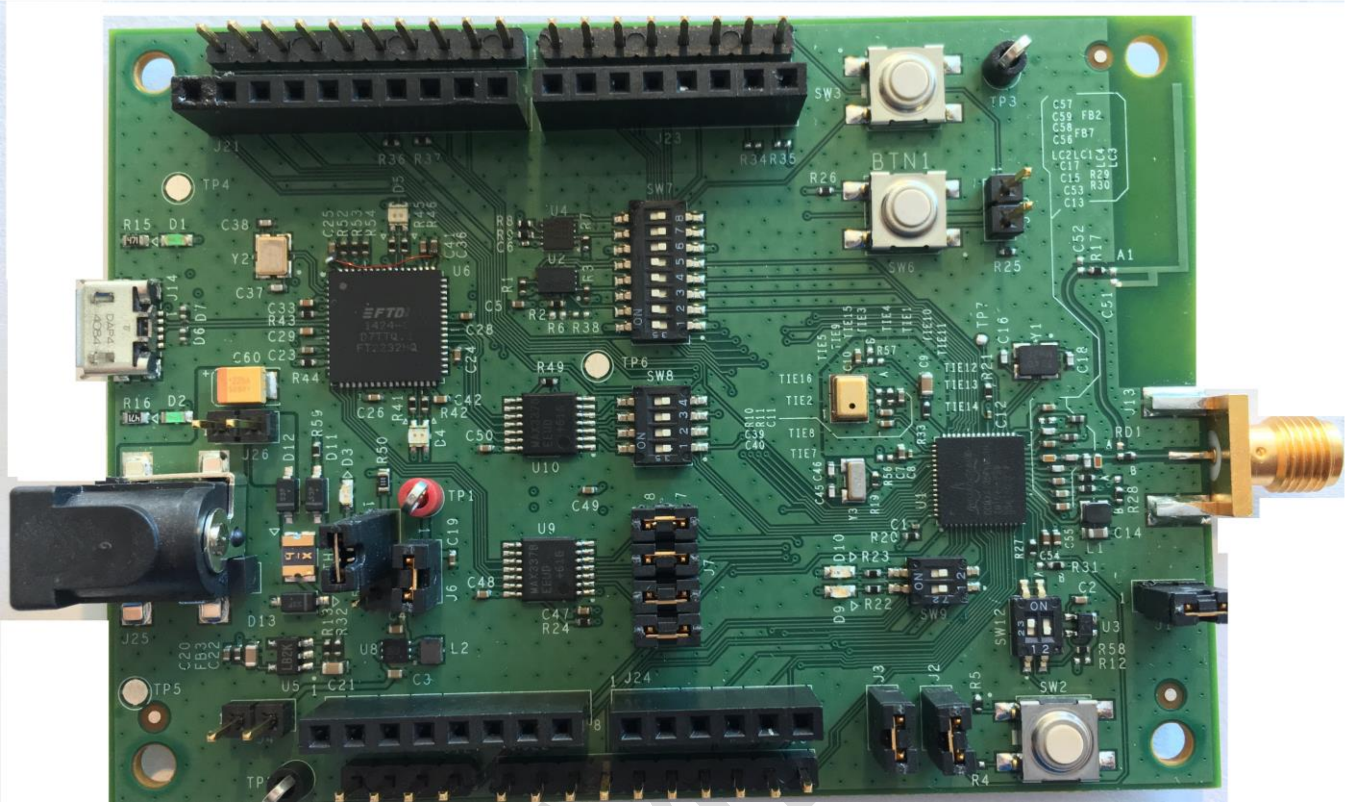


Figure 1: CYW20735WCDEVAL WICED Development Board

3.2 Software Development Kit

The WICED Studio SDK includes:

- The Bluetooth Basic Rate and Low Energy software stack, including a number of sample applications that demonstrate the use of API.
- A generic profile-level API called the WICED BT API.
- Drivers to access on-chip peripherals including UART, SPI, I^{2}C, ADC, PWM, etc.
- WICED BT API documentation
- Utilities to support development, testing and mass production on 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 CYW20735 Bluetooth and Low Energy device.

Table 1: WICED Studio Directory Structure

WICED Studio Directory	*Directory Contents*
Doc	API, reference documentation, and schematics
Drivers	USB drivers for the development board

wiced_tools	Tools including download tool, and other utilities and scripts
43xxx_Wi-Fi\tools\ARM_GNU	Toolchain including compiler, linker, libraries, and headers
20735-B0_Bluetooth\apps	Sample applications
20735-B0_Bluetooth\build	Output files of built applications
20735-B0_Bluetooth\include	WICED API function prototypes and definitions
20735-B0_Bluetooth\libraries	Sources for various WICED interface libraries
20735-B0_Bluetooth\platforms	Configuration files and information for supported hardware platforms
20735-B0_Bluetooth\tools	Common utilities used by the IDE build processes
20735-B0_Bluetooth\WICED	WICED core components

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:

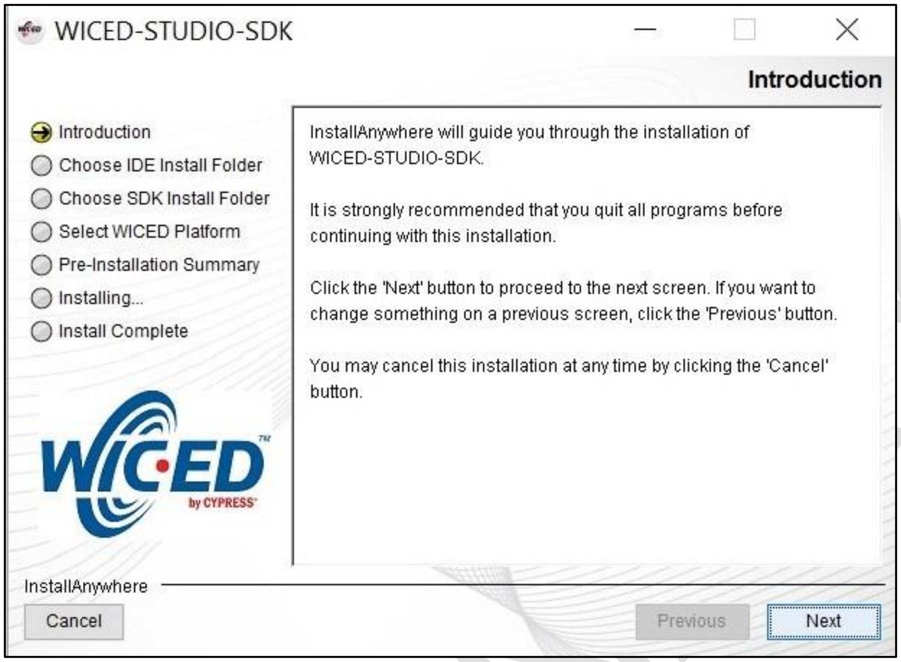
1. Download and install WICED Studio 4 (see "Install WICED Studio")
2. Connect the WICED Development board (see "Connect the WICED Development Board")
3. Create and load an application (see "Build and Load a Sample Application")

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.

Figure 2: WICED Studio Installation Screen

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 “20735-B0 Bluetooth 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 3shows the CYW20735WCDEVAL WICED development board. The figure shows call-outs to the ports, switches, and switch positions relevant to this document.

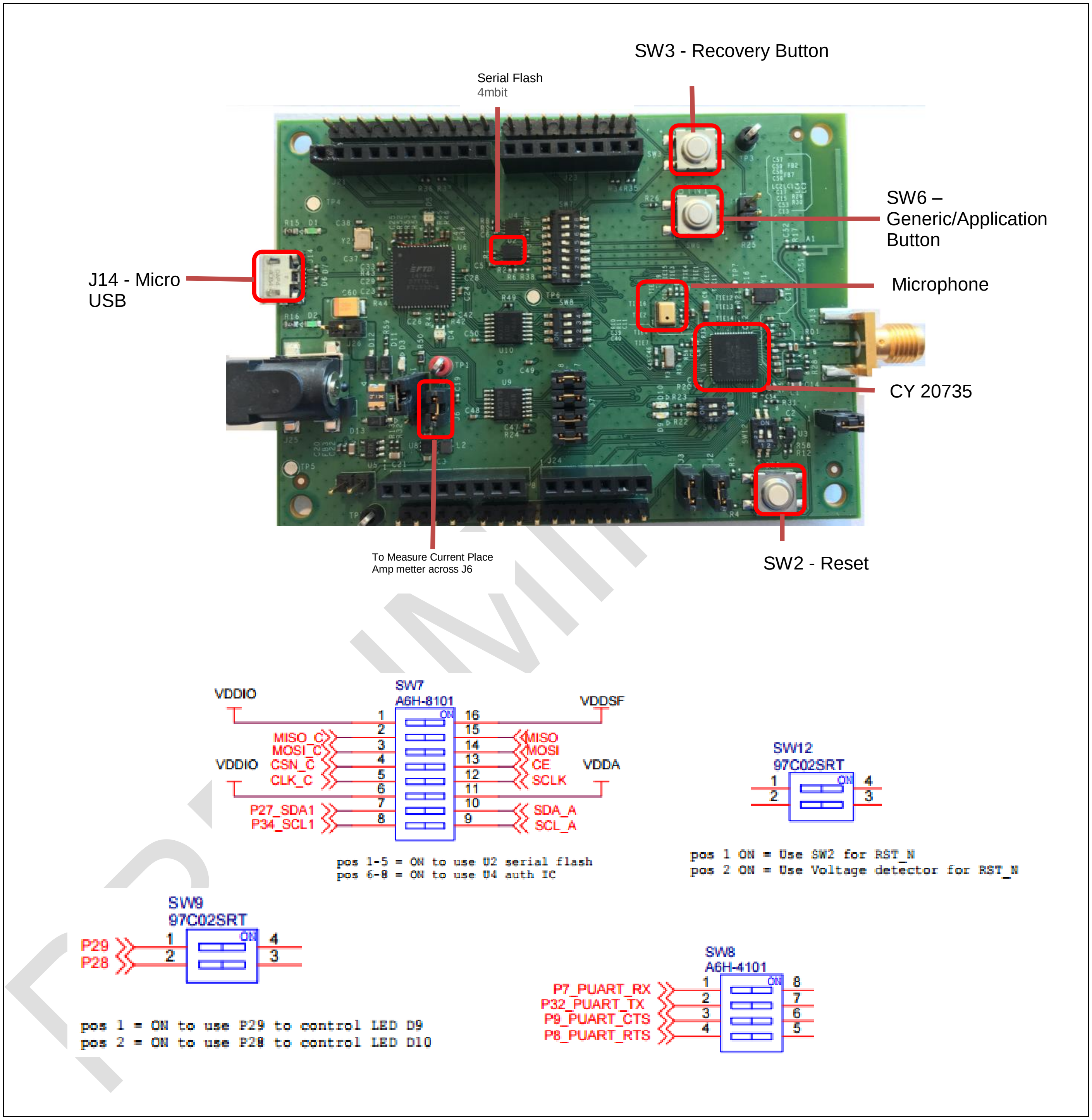

SW3 - Recovery Button
Serial Flash
4mbit
SW6 – Generic/Application Button
J14 - Micro USB
Microphone
CY 20735
To Measure Current Place Amp metter across J6
SW2 - Reset
SW7
A6H-8101
VDDIO
VDDSF
MISO_C
MOSI_C
CSN_C
CLK_C
P27_SDA1
P34_SCL1
MISO
MOSI
CE
SCLK
SDA_A
SCL_A
VDDA
pos 1-5 = ON to use U2 serial flash
pos 6-8 = ON to use U4 auth IC
SW12
97C02SRT
pos 1 ON = Use SW2 for RST_N
pos 2 ON = Use Voltage detector for RST_N
SW9
97C02SRT
P29
P28
pos 1 = ON to use P29 to control LED D9
pos 2 = ON to use P28 to control LED D10
SW8
A6H-4101
P7_PUART_RX
P32_PUART_TX
P9_PUART_CTS
P8_PUART_RTS

Figure 3: CYW20735WCDEVAL WICED Development Board Settings

The Micro-USB connector (J16) supports UART connections and provides +5V power to the board if SW2 is in position 3.

Perform the following steps before verifying driver installation:

1. Set the PIN 1-6 of the **SW7** mini-switch to **ON** so in order to enable Serial flash
2. Set the PIN 1-4 of the **SW8** min-switch to **ON** if you are directing your UART logs to PUART
3. Connect J14 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 4 serve the following purposes:

D1: Green, Indicating that Power switched on.

D2: Green, Indicating that USB cable is plugged in.

D3: Green, Indicating that 3P3V is on

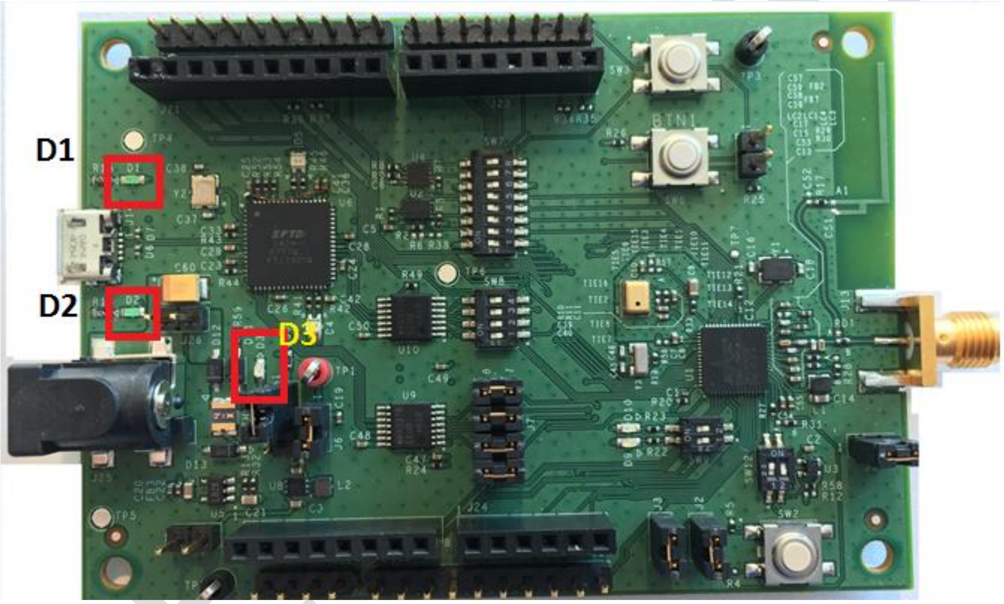
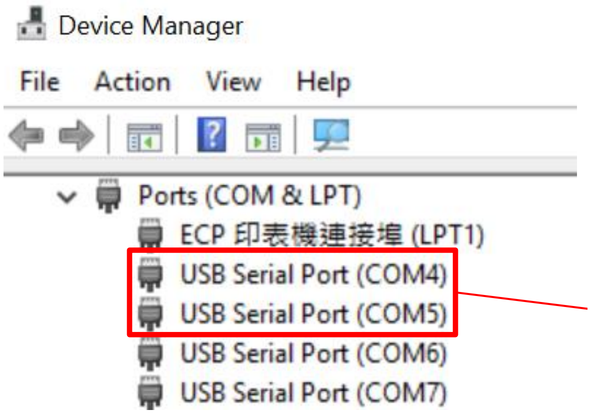


Figure 4: LED Indicators

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.)

Figure 5: Device Manager COM Ports

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 WICED Studio 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.

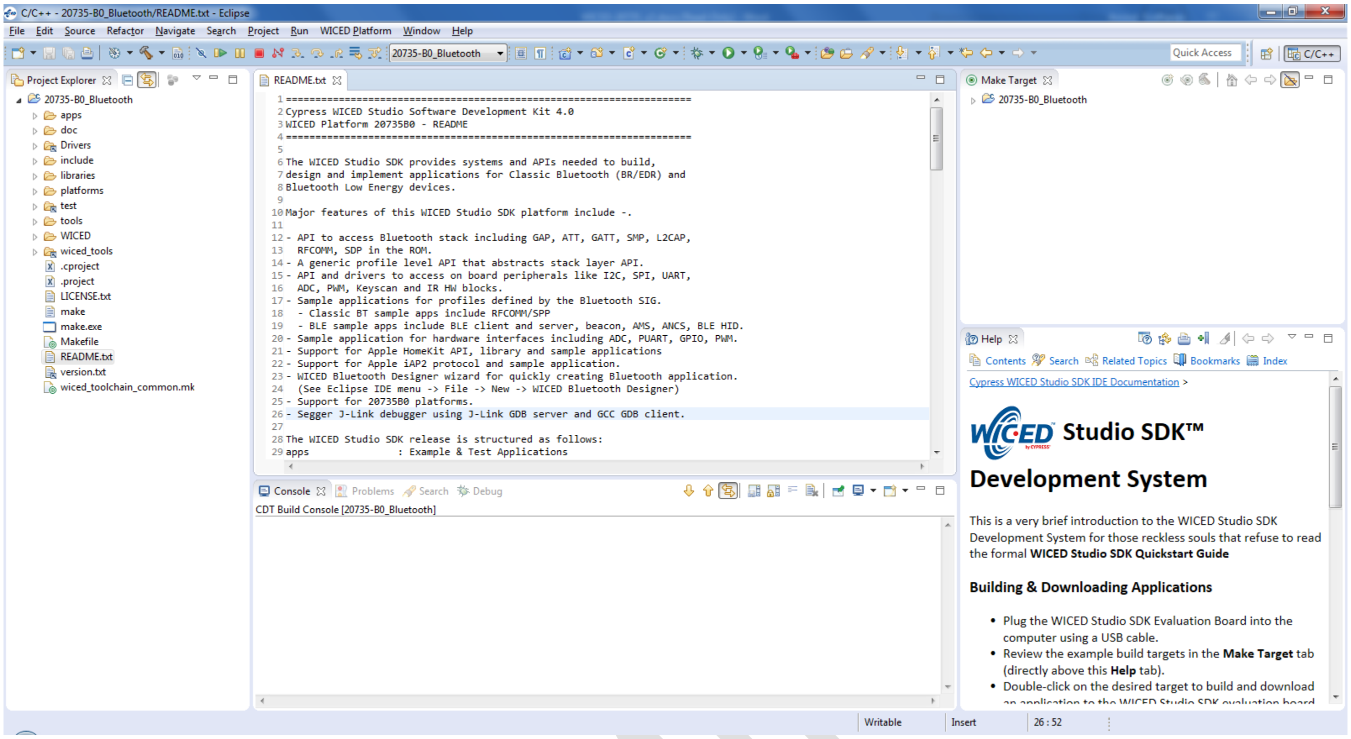
5 Using the WICED Studio IDE

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 Windows 8.x PC with Bluetooth capability.

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 6.



```
==========================================================================
Cypress WICED Studio Software Development Kit 4.0
WICED Platform 20735B0 - README
==========================================================================

The WICED Studio SDK provides systems and APIs needed to build,
design and implement applications for Classic Bluetooth (BR/EDR) and
Bluetooth Low Energy devices.

Major features of this WICED Studio SDK platform include -.

- API to access Bluetooth stack including GAP, ATT, GATT, SMP, L2CAP,
  RFCOMM, SDP in the ROM.
- A generic profile level API that abstracts stack layer API.
- API and drivers to access on board peripherals like I2C, SPI, UART,
  ADC, PWM, Keyscan and IR HW blocks.
- Sample applications for profiles defined by the Bluetooth SIG.
  - Classic BT sample apps include RFCOMM/SPP
  - BLE sample apps include BLE client and server, beacon, AMS, ANCS, BLE HID.
- Sample application for hardware interfaces including ADC, PUART, GPIO, PWM.
- Support for Apple HomeKit API, library and sample applications
- Support for Apple iAP2 protocol and sample application.
- WICED Bluetooth Designer wizard for quickly creating Bluetooth application.
  (See Eclipse IDE menu -> File -> New -> WICED Bluetooth Designer)
- Support for 20735B0 platforms.
- Segger J-Link debugger using J-Link GDB server and GCC GDB client.

The WICED Studio SDK release is structured as follows:
apps                : Example & Test Applications
```

CDT Build Console [20735-B0_Bluetooth]

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).
- Double-click on the desired target to build and download an application to the WICED Studio SDK evaluation board

Figure 6: WICED Studio IDE

The **Help** pane in the lower-right corner of the IDE (see Figure 6) 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 CYW20735WCDEVAL development boards.

The example below shows how to build and run the sample application, which can then connect the WICED development board to a Windows 8 host PC. A sample host application is provided as part of WICED Studio under peerapps in the application folder.

1. Make sure the WICED development board is connected to the PC.
2. Check the UART port after WICED development board is connected to the PC (see “Verify Driver Installation”)
3. Right-Click the target **hello_sensor-BCM920735EVAL_Q40 download** in the **Make Target** pane and select **Edit** to change it and add **UART=COMx** where COMx is the port number in section 4.3 above.
4. Double-click the modified **hello_sensor-BCM920735EVAL_Q40 UART=COMx 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:

```
22:24:45 **** Build of configuration Release for project 20735-B0_Bluetooth ****

"C:\\Users\\arvinds\\Documents\\WICED\\WICED-STUDIO-SDK-4.0\\20735-
B0_Bluetooth\\make.exe" hello_sensor-BCM920735EVAL_Q40 download
Compiling spar_setup.c
Compiling hello_sensor.c
Compiling wiced_bt_cfg.c
Compiling lib_installer.c
Linking target ELF
OK, made elf.
```

```
..\..\..\43xxx_Wi-Fi\tools\ARM_GNU\bin\Win32\arm-none-eabi-objdump: section '.aon'
mentioned in a -j option, but not found in any input file
Call to hello_sensor_spar_crt_setup @ 0021844d
OK, made C:/Users/arvinds/Documents/WICED/WICED-STUDIO-SDK-4.0/20735-
B0_Bluetooth/WICED/wpan/../../build/hello_sensor-BCM920735EVAL_Q40-rom-ram-Wiced-
release/A_20735B0-hello_sensor-rom-ram-spar.cgs. MD5 sum is:
806c5323f8b67625baf701f62d71e7c9 *../../build/hello_sensor-BCM920735EVAL_Q40-rom-ram-
Wiced-release/A_20735B0-hello_sensor-rom-ram-spar.cgs

--------------------------------------------------------------------------------
Patch code starts at                  0x00270400 (RAM address)
Patch code ends at                    0x00271C2C (RAM address)
Patch RW/ZI size                      2140 bytes
Application starts at                 0x002173FC (RAM address)
Application ends at                   0x00218449 (RAM address)

Patch code size                             6188 bytes
Application RAM footprint                   4173 bytes
                                          ------
Total RAM footprint                         6313 bytes (6.2kiB)
--------------------------------------------------------------------------------


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)
```

NOTE: The warning ‘section '.aon' mentioned in a -j option, but not found in any input file’ above is not critical, this is only an indication that the application did not use any data in retention RAM.

Note: if the download fails, see Appendix D for instructions on how to recover a possibly corrupted serial flash.

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, on the Windows 8.1 PC, open the **Settings** charm (move the mouse to the lower or the upper right corner of the screen, then up or down, and click **Settings.**
7. Click **Change PC Settings** and select **Bluetooth** tab (see Figure 7).
8. Click **Add a device** and wait while the PC searches for devices in range.
9. Select **Hello Sensor** device, click **Pair**, and wait for the device connection to complete.

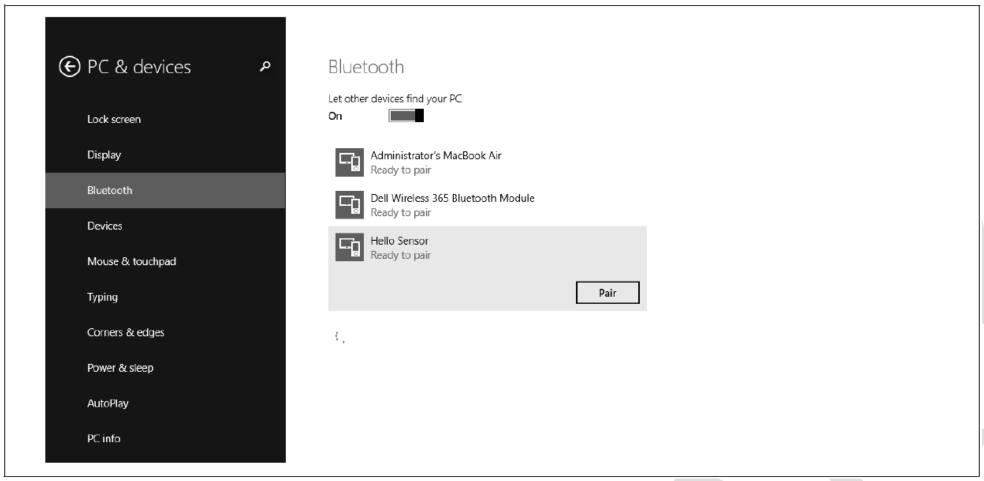


Figure 7: Windows Bluetooth Pairing Screen

5.2 The Hello Client PC application

The Hello Client (hello_client) PC application is provided with WICED Studio to complement the Hello Sensor (hello_sensor) tag application. It is located here:

<WICED-Studio>\20735-B0_Bluetooth\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-Studio>\20735-B0_Bluetooth\platforms\BCM920735EVAL_Q40\20735_OCF.btp

To change the Bluetooth device address, open the file and modify the **DLConfigBD_ADDRBase** variable (or you may set **BT_DEVICE_ADDRESS=xxxxxxxxxxxx** where xxxxxxxxxxxx are the 12 hex nibbles of the device address).

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.

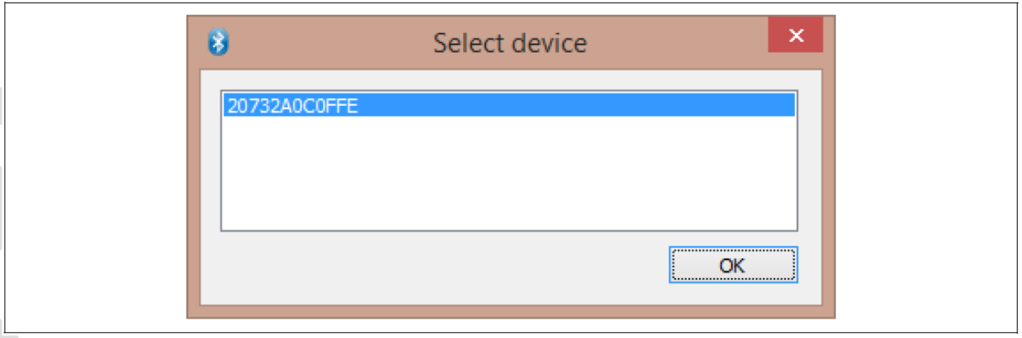


Figure 8: Hello Client Select Device Window

The Hello Sensor application provides paired devices the following information:

- A Hello Service (a proprietary service) with two proprietary characteristics:

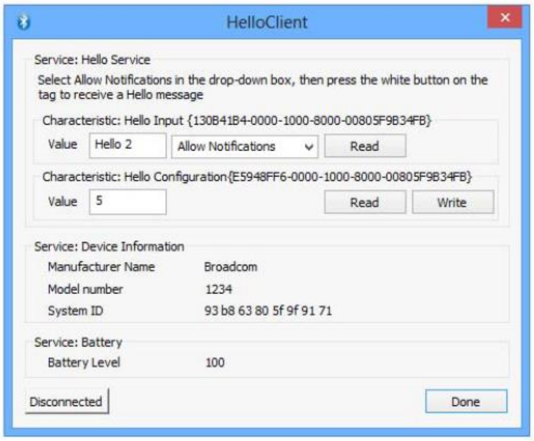
- The value of the Hello Input read-only characteristic may be retrieved using either of the following methods:
  - Manually by using a mouse to click the Read button on the HelloClient PC application.
  - Automatically by pressing the **Application** button **SW6** (see Figure 3) on the WICED development board (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 (see Figure 9) when the **Application** button 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.

5.3 Testing the Hello Sensor application

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

- Hello Input Characteristic
  - Select **Allow Notifications** in the combo box
  - Push the **Application** button **SW6** on the WICED board (see Figure 3). The Hello X message is displayed in the Value field. Each time the button is pressed, the message number increments.
- Hello Configuration Characteristic
  - Change the value for the Hello Configuration to 5 and then click Write.
  - Push the **Application** button **SW6** on the WICED board (see Figure 3). The LED on the tag blinks five times.



HelloClient

Service: Hello Service

Select Allow Notifications in the drop-down box, then press the white button on the tag to receive a Hello message

Characteristic: Hello Input {130B41B4-0000-1000-8000-00805F9B34FB}

Value: Hello 2 | Allow Notifications

Characteristic: Hello Configuration{E5948FF6-0000-1000-8000-00805F9B34FB}

Value: 5

Service: Device Information

Manufacturer Name	Broadcom
Model number	1234
System ID	93 b8 63 80 5f 9f 91 71

Service: Battery

Battery Level	100

Figure 9: Hello Client Services Window

5.4 Viewing Traces from the WICED Development Board

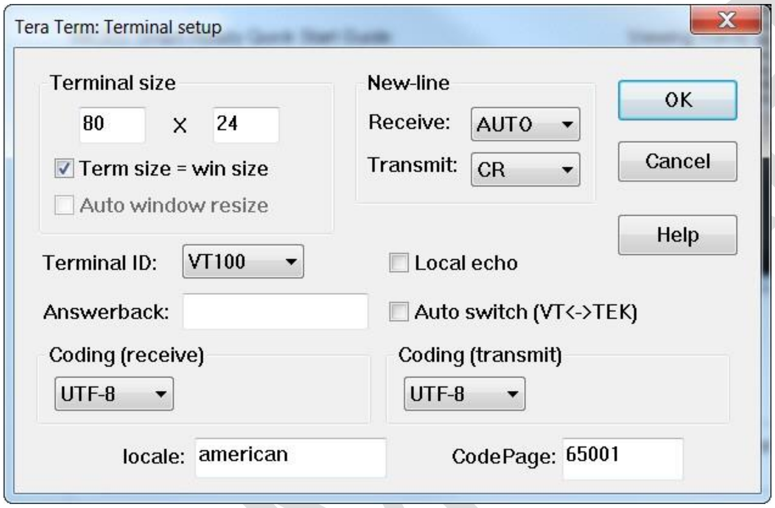
To view the traces:

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).

5.4.1 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

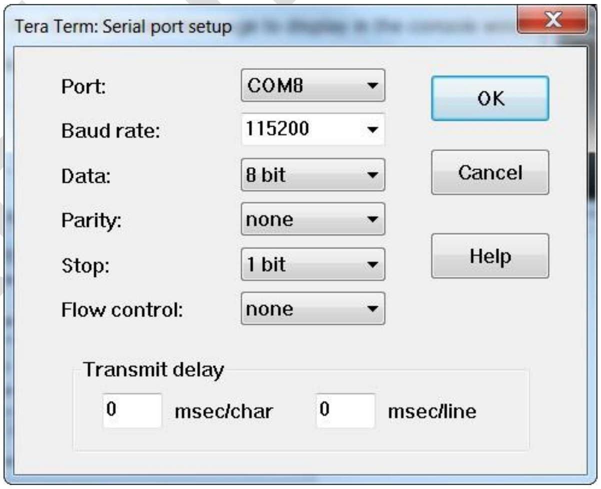
Coding (transmit): UTF-8

locale: american

CodePage: 65001

Figure 10: Tera Term Settings

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

Figure 11: Serial Port Settings

4. Press the **Reset** button (see Figure 3) on the WICED development board to view the application start-up messages.

5.5 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 WICED Studio 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

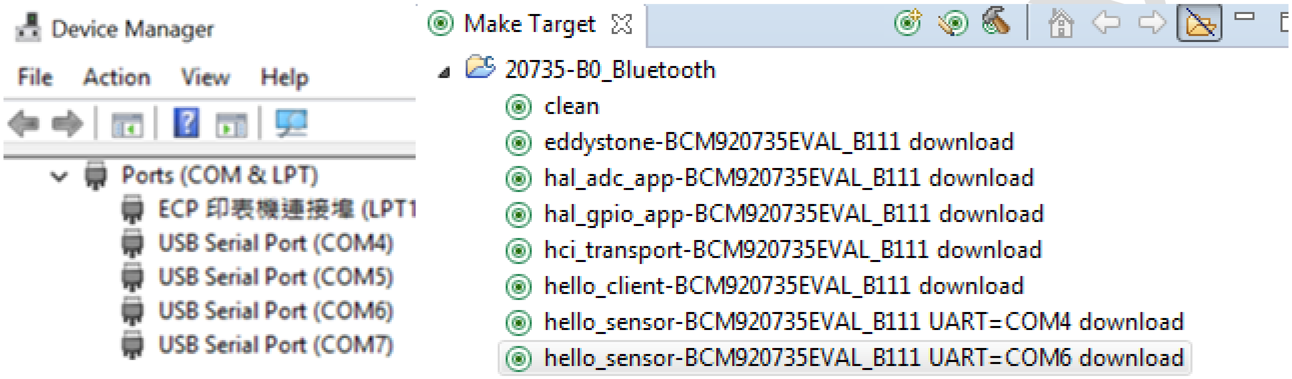
Particularly useful keystrokes are listed below:

- 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: `<20735-B0_Bluetooth/apps/hello_sensor/hello_sensor.c.`
  2. Press **CTRL-SHIFT-T**
  3. Start typing an element, for example, BTM_BLE_ADVERT_
  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 12 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)

20735-B0_Bluetooth
- clean
- eddystone-BCM920735EVAL_B111 download
- hal_adc_app-BCM920735EVAL_B111 download
- hal_gpio_app-BCM920735EVAL_B111 download
- hci_transport-BCM920735EVAL_B111 download
- hello_client-BCM920735EVAL_B111 download
- hello_sensor-BCM920735EVAL_B111 UART=COM4 download
- hello_sensor-BCM920735EVAL_B111 UART=COM6 download

Figure 12: Configuration for Two WICED Development Boards

Appendix C: Connecting a WICED Development Board 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 Corrupt WICED Development Board

The following steps describe how to recover a WICED development board if the serial flash has been corrupted.

Note: See Figure 3 for port and switch references associated with the recovery steps provided in this section.

First, verify the following:

1. Verify that J14 of the WICED development board is connected to the development PC.
2. Verify that the USB-UART driver is correctly installed (see Verify Driver Installation).
3. Verify that the SWx mini-switches are in their default positions

After those verifications, use the following method to recover a corrupt serial flash:

1. Press and hold the Recovery button (SW3).
2. Press the Reset button (SW2).
3. Release SW3.

Document History

Document Title: ANCYWICED002 – WICED Quick Start Guide for BT CYW20735

Document Number: 001-CYWICED002

Revision	ECN	Submission Date	Description of Change
*A		10/19/16	Minor corrections
**		10/12/16	Initial release

References

[1] CYW20735 Data Sheet

[2] CYW920735WCDEVAL Hardware User Manual

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.