OTP Programming and NVRAM Development in SDIO Mode

Associated Part Family: CYW43438/4343S/4343W

This application note describes the method for creating and programming an nvram.txt file. This file is used to test a new board design, optimize NVRAM values, and program the one-time programmable (OTP) nonvolatile memory in the CYW43438/4343S/4343W devices.

Contents

1 About This Document

1.1 Purpose and Audience

This document is intended for design and applications engineers. It contains information on:

- NVRAM content development and OTP programming flow
- SDIO Windows® XP driver installation
- Customizing the nvram.txt file
- OTP programming procedure

1.2 Cypress part numbering scheme

Cypress is converting the acquired IoT part numbers from Broadcom to the Cypress part numbering scheme. Due to this conversion, there is no change in form, fit, or function as a result of offering the device with Cypress part number marking. The table provides Cypress ordering part number that matches an existing IoT part number.

Table 1. Mapping Table for Part Number between Broadcom and Cypress

Broadcom Part Number	Cypress Part Number
BCM43438	CYW43438
BCM4343S	CYW4343S
BCM4343W	CYW4343W

1.3 Before You Begin

It is recommended that the users of this application note request the following items from Cypress:

- A CYW43438/4343S/4343W board reference design package that contains:
  - The reference board schematic, bill of materials, and layout. Be sure to specify either the WLBGA or WLCSP package.
  - An nvram.txt template file for the reference board.
- A Windows XP or Linux® device driver for the relevant SDIO device
- Cypress transmit signal strength indicator (TSSI) calibration tools

2 IoT Resources

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 Introduction

The Cypress CYW43438/4343S/4343W are single-chip IEEE802.11 b/g/n + BT/FM devices intended for embedded applications. For the WLAN section of the device, a one-time programmable (OTP) nonvolatile memory is available for storing board-specific information such as product ID, manufacturer ID, MAC address, and more. Excluding header information, up to 306 bytes of OTP memory is available for WLAN information on the CYW43438/4343S/4343W. This application note addresses only SDIO applications.

The OTP memory content, together with an editable NVRAM file (referred to throughout this document as the nvram.txt file), combines to create a complete card information structure (CIS) that the device driver uses to initialize and configure the CYW43438/4343S/4343W.

4 OTP Programming Considerations

For designs where the host and device are permanently connected together, which is typically done with a hardwired SDIO interface, programming the OTP memory in production is optional. It is equally acceptable to store all NVRAM parameters in host firmware and keep the OTP blank in production. For devices that may be installed on different hosts, the OTP can be programmed to protect the unique MAC address and to prevent end-users from altering power control parameters (such as maximum output power and other power amplifier parameters).

For host platforms running the Linux® or Windows® XP operating system, it is not necessary to program the OTP memory during board bring-up and hardware tuning. Instead, store all required board variables in the nvram.txt file. Although OTP programming is not required for devices used on these host operating systems, nvram.txt file development is still required.

The initial state of all OTP bits in an unprogrammed device is 0. Individual bits can be set to 1, but once set, they can never be reset back to 0. The entire OTP array can be programmed in a single-write cycle using wl commands provided with the SDIO driver. Alternatively, multiple-write cycles can be used to selectively program specific fields, but only the bits that are still in the 0 state can be set to the 1 state during each programming cycle.

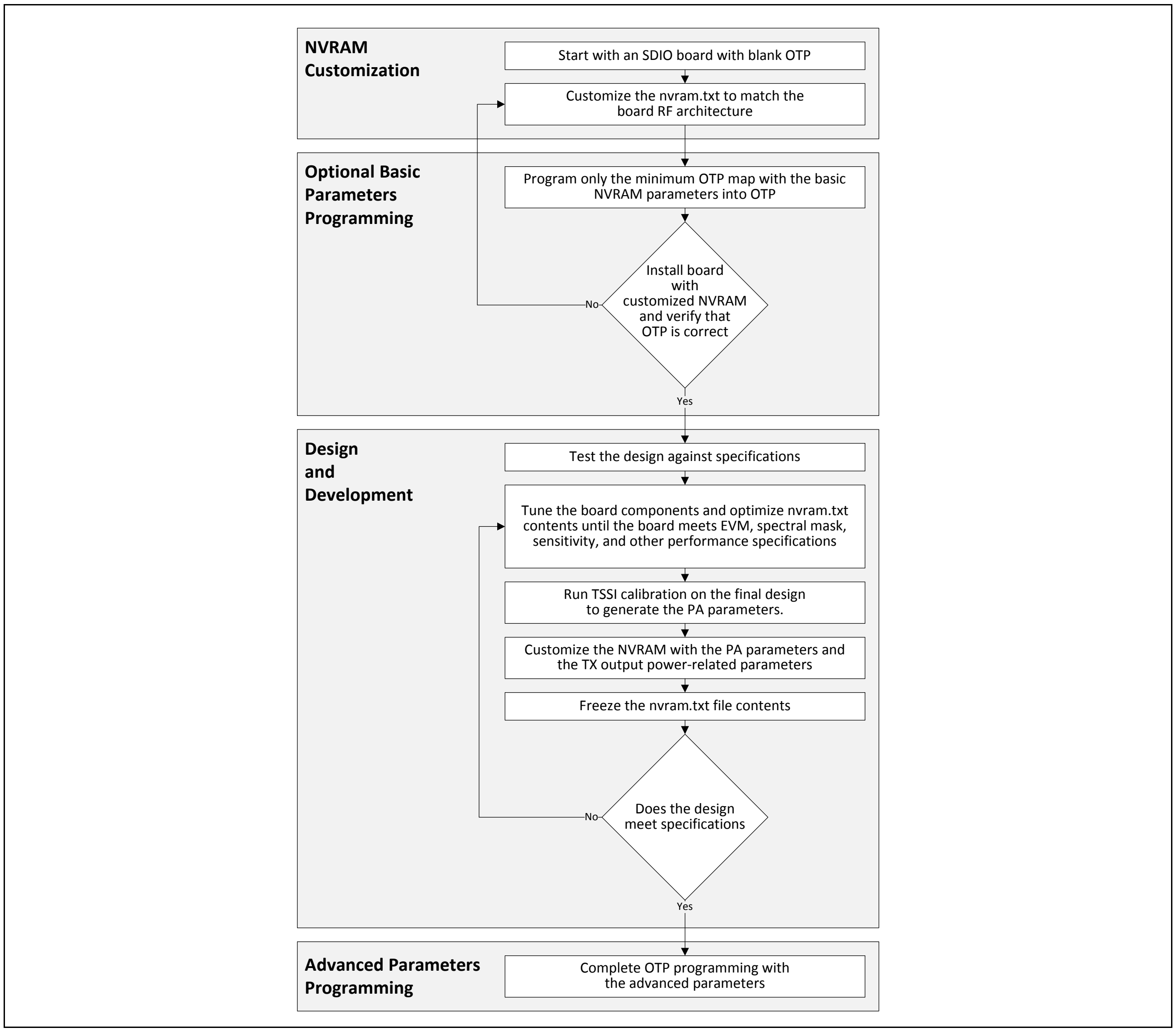
Because the OTP programming process is irreversible, Cypress recommends that board designers finalize all parameters before programming the OTP memory. Boards and modules should be tested using only the editable nvram.txt file. The nvram.txt parameters are loaded by the driver into on-chip RAM, allowing the chip to be tested even if the OTP memory has not yet been programmed. This method lets board designers tune RF components and alter critical parameters while testing boards using different versions of the nvram.txt file. As an option, a few basic parameters (such as the board type and MAC address) can be programmed into the OTP prior to board testing during development. If a parameter is present in both the on-chip OTP and the nvram.txt file, the value from the OTP overrides the value from the nvram.txt file; the WLAN driver ignores the corresponding value in the nvram.txt file.

Note: Due to the irreversible OTP programming process, board development should be done on boards with blank OTP memory using the parameters in the editable nvram.txt file. Do not program the OTP memory until the contents of the nvram.txt file have been verified and frozen

5 NVRAM Content Development and OTP Programming Flow

Figure 1 shows the nvram.txt file content development and the OTP programming flow. Parameters in the nvram.txt can be divided into two groups: basic parameters and advanced parameters. Pertinent OTP programming details for each phase can be found in OTP Programming Procedure on page 7.

Figure 1. NVRAM Development and OTP Programming Flow


NVRAM Customization
Start with an SDIO board with blank OTP
Customize the nvram.txt to match the board RF architecture
Optional Basic Parameters Programming
Program only the minimum OTP map with the basic NVRAM parameters into OTP
Install board with customized NVRAM and verify that OTP is correct
No
Yes
Design and Development
Test the design against specifications
Tune the board components and optimize nvram.txt contents until the board meets EVM, spectral mask, sensitivity, and other performance specifications
Run TSSI calibration on the final design to generate the PA parameters.
Customize the NVRAM with the PA parameters and the TX output power-related parameters
Freeze the nvram.txt file contents
Does the design meet specifications
No
Yes
Advanced Parameters Programming
Complete OTP programming with the advanced parameters

Note: The OTP programming flow shown in Figure 1 is used only during the development stages of the project on small quantities of boards or modules. Once this process is complete and a “golden” nvram.txt file or OTP file is established, the development phase can be bypassed, and the programming can be done in high volume for mass production, following the correct manufacturing procedure defined by each manufacturer.

6 Customizing the nvram.txt File

This section describes customizing, editing, and finalizing the nvram.txt file for OTP programming.

6.1 Using the nvram.txt File Template

For each Cypress reference board design an nvram.txt file is provided, which is exactly matched to that specific-board design. Typically, the file is named after the board it supports (for example, bcm943430wlselgs.txt). It may be provided with the reference board design package or with the driver release. The latest version of the file can be obtained by submitting a request on the Cypress CSP. Use this file as a sample or template to begin the customization to match your own board design.

Note: When a change is made in the nvram.txt file, the file must be saved, and the wireless device driver must be disabled and reenabled in the Windows Device Manager for the change to take effect. Save the file as “nvram.txt” and place it in the C:\Windows\system32\drivers\ directory. Delete or replace any previous file with the same name in this directory.

A sample nvram.txt file, with parameters that are common to Cypress CYW43438/4343S/4343W SDIO reference design boards, is shown in Table 2 on page 5. No specific order is required for the parameters in the nvram.txt file.

Parameters listed in Table 2 are design variables which must be reviewed prior to starting board or module testing. Specifically, the boardflags, the swctrlmap variables, and the number of antennas must be customized to match the board RF architecture. During the board development phase, start with the default power amplifier (PA) parameters provided in the nvram.txt template. The PA parameters are eventually optimized using Cypress's transmit signal strength indicator (TSSI) calibration tools.

Note: The parameters in Table 2 typically require tuning to each specific-board or module design. This is not an exhaustive list. Additional parameters may be added by Cypress at any time to control RF performance-related attributes of the driver. Always check with Cypress for the latest version of the nvram.txt file for the reference design before starting customization for your board design.

Table 2. NVRAM Parameters Requiring Customizing for Each Board Design

NVRAM Parameter	Example Data	Description
manfid	0x2d0	Defines the board manufacturer using standard PCI/CIS ID codes.The default is Cypress. No 4343 code uses this variable but it is available so that code in the future can key off functionality. The combination of the manfid and prodid should uniquely identify the design.
prodid	0x0726	Uniquely identifies the board to the driver software. If needed, the driver software can use this to enable board-specific tuning or features.
vendid	0x14e4	Vendor ID–identifies the IC vendor: always 0x14e4, which is the Cypress PCI vendor ID.
devid	0x43e2	Device ID–identifies the device. Does not necessarily correspond to the chip number.
boardtype	0x0726	Board type: a critical parameter that should be copied form a similar Cypress reference design.
boardrev	0x1101	Board revision tracked by the Cypress internal test tool (optional). **Example:** ■ 0x1101 converts to P101 ■ 0x1208 converts to P208
boardnum	22	Uniquely identifies the board instance.
boardflags	0x00404201	Board configuration flags that define power topology, external components (ePA, eLNA), etc.
macaddr	00:90:4c:c5:12:38	Sets the device MAC address.
sromrev	11	Revision number of this file.
boardflags	0x00404201	Sets flags for board configuration. For example, LSB 1 specifies that the board implements Bluetooth coexistance.
xtalfreq	37400	Onboard XTAL or oscillator frequency, in kHz.
nocrc	1	On SDIO firmware images, the image has a 32-bit CRC appended to it, and when it starts up after download it does an internal CRC check and fails if the CRC fails. nocrc=1 causes the image to skip this check.
ag0	0x82	An 8-bit number that describes antenna gain in quarter dB increments. Used in conjunction with the regulatory tables to set power appropriately in locations with radiated limits. ■ Bit [5:0] are integer dB units ■ Bit 7 is half dB units ■ Bit 6 is quarter dB units. **Example:** 0x82 = 2.5 dB (2 + 2 × 0.25)
aa2g	1	Number of antennas available for the 2.4 GHz, in bit-mapped binary format: 1 = 01b for one antenna
ccode	All	Country code.
extpagain2g	0	Selects internal PA operation for 2.4 GHz.
pa2ga0	–168, 7161, –820	Parameter values are obtained from TSSI calibrations.

Table 2. NVRAM Parameters Requiring Customizing for Each Board Design (Cont.)

NVRAM Parameter	Example Data	Description
AvVmid_c0=0x0	0xc8	RSSI midpoint select (values can be adjusted, depends on TSSI curve plot.
cckpwroffset0	5	CCK power offset adjustment
maxp2ga0	74	Maximum output power for the 2.4 GHz band in hexadecimal or decimal format. Units of 0.25 dB. This applies to all complementary code keying (CCK) rates as measured at antenna port. The nominal target power in dBm for CCK packets is (0.25 × maxp2ga0 in decimal) – 1.5 dB. In the example shown for 74, the maximum output power is 0.25 * 74 = 18.5 dBm and the nominal power is 18.5 – 1.5 = 16 dBm.
txpwrbckoff	6	The amount of TX power back-off used to set the target power from the minimum regulatory and board limits.
cckbw202gpo	0	CCK power reduction offsets for 20 MHz rates (11, 5.5, 2, and 1 Mbps).
legofdmbw202gpo	0x66666666	Per-rate TX power back-off for legacy-OFDM rates.
mcsbw202gpo	0x88888888	Per-rate TX power back-off for MCS rates
ofdmdigfilttype	18	OFDM digital filter type.
ofdmdigfilttypebe	18	OFDM digital filter type.
il0macaddr[a]	00:90:4c:c5:12:38	The il0 MAC address format enables the firmware to use a default (dummy) MAC address if the OTP is blank (that is, if no valid MAC address has previously been programmed into the OTP).
wl0id	0x431b	Default ID used by mfg driver.
muxenab[b]	0x10	Specifies how the UART/JTAG/GPIO are configured: ■ 0x1 is configured for UART enable ■ 0x1 is configured for GPIOs ■ 0x8f is configured for JTAG ■ 0x10 is configured for HW OOB interrupt/signal (WL_host_wake).
sd_gpout	0	OOB parameter to select GPIO0 for WLAN host wake[b].
sd_gpval	1	OOB parameter to select GPIO0 as output[b].
sd_oobonly	1	OOB parameter to select OOB interrupt[b].
cldo_pwm	0x4	CLDO PWM voltage settings - 0x4 - 1.1 volt.
btc_params 8	45000	BT COEX deferral limit setting.
btc_params 10	20000	BT COEX deferral limit setting.

a.When devices with blank OTP are used, the firmware may fail to load unless a MAC address is provided. Cypress recommends using an il0macaddr value during board development, which provides the driver with a dummy MAC address. With il0macaddr in nvram.txt, the driver loads even when the OTP is blank. As an alternative a unique MAC address can be programmed into OTP using the basic parameters OTP map shown in Figure 2 on page 8. In production, each board or module should have a valid and unique MAC address programmed into its OTP. The il0macaddr is ignored by the driver when a MAC address is programmed in the OTP.

b.Set this value if software OOB is used.

6.2 Editing the nvram.txt File

The nvram.txt file content should be edited in a properly formatted text editor, such as Notepad++ or WordPad++, so that the original format of the file is preserved. Using a non-formatted text editor (such as Notepad) may corrupt the format of the NVRAM map, thus causing the driver to fail to correctly read the nvram.txt file.

6.3 Finalizing the nvram.txt File

After the final PA parameters for the design have been generated, edit the nvram.txt file to update the PA parameters derived from using the TSSI tool, and then adjust the Tx output power-related parameters in the nvram.txt file. Run output power tests (using the updated nvram.txt file) to verify that these parameters are providing the correct output power. Verify that the RF performance (such as EVM, spectral mask, and rxper) meets design specifications.

Cypress recommends running a regulatory prescan to verify that the required output power can be delivered without violating the band-edge limits. If the band-edge limits cannot be met, it may be necessary to reduce the output power at the band-edge channels.

After all prototype tests have passed and all nvram.txt file parameters have been optimized and frozen, users can select the needed parameters to program the OTP for production.

The CYW43438/4343S/4343W have 306 bytes of space in the OTP memory available for user data. Given the limited space in the OTP, it is impossible to program the entire nvram.txt file to the OTP. The programmer must be very careful to select only the necessary parameters that go into the OTP. Parameters that typically go into the OTP are those that are unique to the board (such as MAC address) and those that are required to satisfy local regulatory requirements, which are usually output power-related parameters (such as maximum output power, power offset per-rate, PA parameters, country code, etc.).

7 OTP Programming Procedure

Prior to OTP programming, an OTP binary map file must be prepared and edited with correct values. An OTP binary map completely defines the parameters that have to be programmed to the OTP memory. The SDIO OTP data format is based on the CIS as defined by the PCMCIA/SD Card Association. The CIS data contains the hardware header followed by one or more data blocks, where each data block (or tuple) contains the type, length, and value of the tuple. Refer to Appendix A.1: "CIS Map," on page 12 for details.

At the start of the OTP map, a string called the SDIO hardware header must be present preceding any NVRAM variables. When a driver detects content in the OTP, the SDIO hardware header is required to boot up the CYW43438/4343S/4343W devices via the SDIO interface. Therefore, the SDIO hardware header is the minimum set of parameters when programming an OTP. The hardware header is shown in Figure 2 on page 8. Any other parameters needed to be programmed to the OTP are appended after the SDIO hardware header (refer to Creating and Editing the OTP Binary Map on page 8).

7.1 Programming Basic Parameters into OTP

Parameters in the nvram.txt file that are to be programmed to the OTP must follow the SDIO hardware header in the OTP binary map. Each parameter requires a CIS tuple in the CIS structure. Most parameters in the nvram.txt file have a unique identifier called the CIS tuple tag. The driver recognizes and parses each CIS tuple by its tag number. For a list of the CIS tuples and their tag numbers, see Appendix A.1: "CIS Map," on page 12.

Table 3 lists the common basic nvram.txt file parameters with their tag numbers and the byte size they occupy in the OTP memory space. Basic parameters are typically values fixed to a specific device or board and tend to retain their values across the life of the device/board. For this reason, it is generally acceptable to program these basic parameters to the OTP early in the development, before the design is frozen.

Table 3. Basic NVRAM Parameters CIS Tuple Tags

NVRAM Parameter	CIS Tuple Tag	Length of Value (in Bytes)
sromrev	0x00	1
boardrev	0x02	2
boardtype	0x1b	2
macaddr	0x19	6
ccode[a]	0x0a	2

a.The value for `ccode` in the nvram.txt file is in ASCII format. It must be converted to hexadecimal format before entering it into the OTP map (for example, "US" = "0x55 0x53").

In the OTP binary map, each tuple is formed by the four fragments described in Table 4.

Table 4. CIS Tuple Format

Fragment	Description
80	This number indicates the beginning of a new tuple. 0x80 is specific to Cypress tuple subtags.

Table 4. CIS Tuple Format

Fragment	Description
Length	The length defines the total size (in bytes) of the tag plus the value of the tuple that occupies the OTP memory space.
Tag	The tag identifies a parameter in the nvram.txt file. A tag usually takes one byte in memory.
Value	The value of the parameter is in little-endian format (that is, the first byte is the least-significant byte).

For example, a tuple that looks like the following is defined by the fragments listed in Table 5 on page 8:

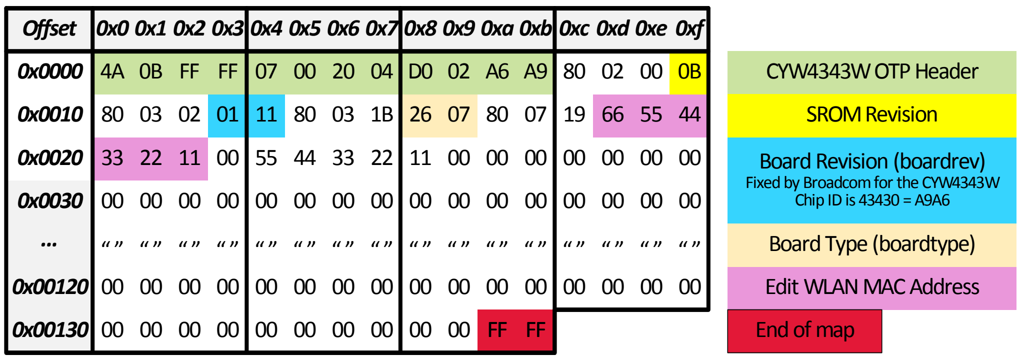
```
8    0    0    4    0
0    3    2    0    0
```

Table 5. An Example of Tuple Definition

Fragment	Description
80	Beginning of a new tuple.
03	The tag (1 byte) and the value (2 bytes) will occupy 3 bytes total in the OTP memory.
02	Tag of 0x02 is the identifier for boardrev in the nvram.txt file.
00 40	The value of boardrev in reverse binary byte, or 0x4000.

Figure 2 shows an example of the OTP binary map for the CYW43438/4343S/4343W devices that contains some of the nvram.txt file parameters listed in Table 3 on page 7.

Figure 2. Example OTP Binary Map Containing Basic Parameters



Offset	0x0	0x1	0x2	0x3	0x4	0x5	0x6	0x7	0x8	0x9	0xa	0xb	0xc	0xd	0xe	0xf
0x0000	4A	0B	FF	FF	07	00	20	04	D0	02	A6	A9	80	02	00	0B
0x0010	80	03	02	01	11	80	03	1B	26	07	80	07	19	66	55	44
0x0020	33	22	11	00	55	44	33	22	11	00	00	00	00	00	00	00
0x0030	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
...	“ ”	“ ”	“ ”	“ ”	“ ”	“ ”	“ ”	“ ”	“ ”	“ ”	“ ”	“ ”	“ ”	“ ”	“ ”	“ ”
0x00120	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
0x00130	00	00	00	00	00	00	00	00	00	00	FF	FF				

CYW4343W OTP Header
SROM Revision
Board Revision (boardrev)
Fixed by Broadcom for the CYW4343W Chip ID is 43430 = A9A6
Board Type (boardtype)
Edit WLAN MAC Address
End of map

In this example, the values for each parameter are as follows:

- sromrev = 0x0B
- boardrev = 0x1101
- boardtype = 0x0726
- macaddr = 66:55:44:33:22:11

Note:

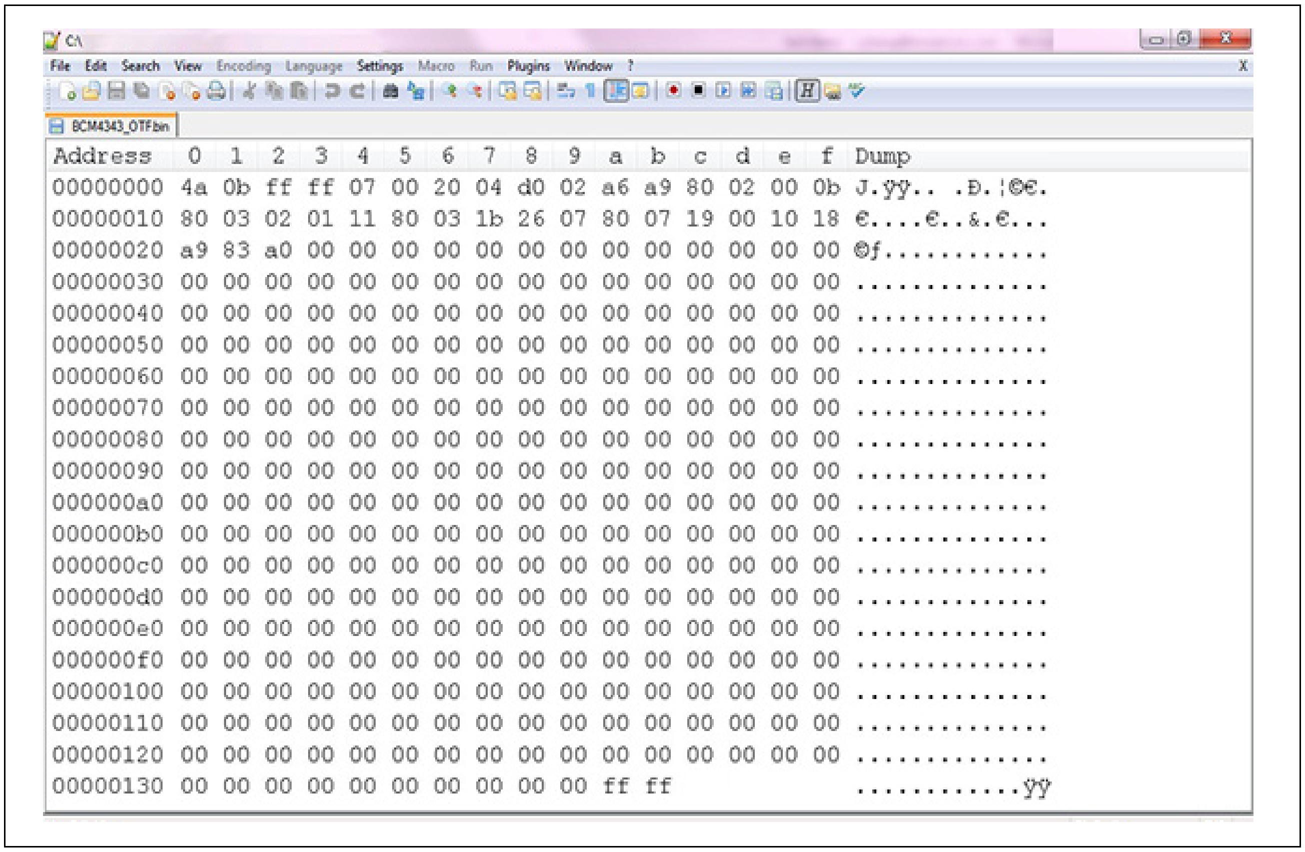
- CIS tuples do not have to be in a particular order because each tuple begins with a unique identifier.
- OTP bytes can be written to only once, so only blank or zero-programmed bytes can be programmed on subsequent write cycles.

7.2 Creating and Editing the OTP Binary Map

Use a hexadecimal text editor to create and edit an OTP binary map. A hexadecimal text editor preserves formatting of the nvram.txt file. Do not use Notepad as it modifies formatting and corrupts the nvram.txt file. Writing to the

OTP requires a ".bin" file that fits within the OTP size. For the CYW43438/4343S/4343W devices, the OTP maximum size limit is 316 bytes. Figure 3 shows the Hex OTP map template.

Figure 3. Hexadecimal OTP Map Template


C:\
File Edit Search View Encoding Language Settings Macro Run Plugins Window ?
BCM4343_OTP.bin

```
Address   0  1  2  3  4  5  6  7  8  9  a  b  c  d  e  f  Dump
00000000 4a 0b ff ff 07 00 20 04 d0 02 a6 a9 80 02 00 0b J.ÿÿ.. .Ð.¦©€.
00000010 80 03 02 01 11 80 03 1b 26 07 80 07 19 00 10 18 €....€..&.€...
00000020 a9 83 a0 00 00 00 00 00 00 00 00 00 00 00 00 00 ©ƒ............
00000030 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ..............
00000040 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ..............
00000050 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ..............
00000060 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ..............
00000070 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ..............
00000080 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ..............
00000090 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ..............
000000a0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ..............
000000b0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ..............
000000c0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ..............
000000d0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ..............
000000e0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ..............
000000f0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ..............
00000100 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ..............
00000110 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ..............
00000120 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ..............
00000130 00 00 00 00 00 00 00 00 00 00 ff ff             ............ÿÿ
```

1. Add or edit each byte in the map to fill in the SDIO hardware header and the CIS tuple according to the OTP binary map instructions described earlier in this section. The map shown in Figure 3 has been edited to match the CYW43438/4343S/4343W OTP binary map example in Figure 2 on page 8.
2. When editing is complete, save the file and manually change the .txt file extension to .bin. The file name must have .bin extension so that it can be programmed to the OTP. Store this .bin file in the working directory that contains the wl.exe file.

For example purposes, this file is referred to as 4334_OTP.bin in the following instructions.

7.3 Programming Procedure Using wl Commands

To program the OTP binary map to the CYW43438/4343S/4343W devices:

1. Load the driver (refer to Appendix A.2: "Driver Installation and Setup," on page 16) to the CYW43438/4343S/4343W devices with the customized nvram.txt file. Verify the driver installation was successful by giving a few wl commands (for example, wl ver).
2. Type the following wl command to program the 4334_OTP.bin to the OTP:

```
> wl ciswrite 4334_OTP.bin
```

3. In the Windows Device Manager, disable the wireless device and then enable it.
4. Confirm OTP is programmed successfully by running the command `> wl cisdump`.

The following MAC address is used as an example: a0:83:a9:18:10:00. The output should match exactly the OTP binary map created in Figure 3 on page 9.

Example: wl cisdump output

```
[root@orbiter-dut 4343]# wl up
[root@orbiter-dut 4343]# wl ver
```

```
7.10 RC226.46
wl0: Apr  6 2014 14:49:23 version 7.10.226.7 (r468234 WLTEST) FWID 01-a011548d
[root@orbiter-dut 4343]# wl cisdump
Source: 2 (Internal OTP)
Maximum length: 316 bytes
Byte 0:    0x4a 0x0b 0xff 0xff 0x07 0x00 0x20 0x04
Byte 8:    0xd0 0x02 0xa6 0xa9 0x80 0x02 0x00 0x0b
Byte 16:   0x80 0x03 0x02 0x01 0x11 0x80 0x03 0x1b
Byte 24:   0x26 0x07 0x80 0x07 0x19 0x00 0x10 0x18
Byte 32:   0xa9 0x83 0xa0 0x00 0x00 0x00 0x00 0x00
Byte 40:   0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00
Byte 48:   0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00
Byte 56:   0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00
Byte 64:   0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00
Byte 72:   0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00
Byte 80:   0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00
Byte 88:   0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00
Byte 96:   0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00
Byte 104:  0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00
Byte 112:  0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00
```

Byte	*120:*	0x00	0x00	0x00	0x00	0x00	0x00	0x00	0x00
Byte	*128:*	0x00	0x00	0x00	0x00	0x00	0x00	0x00	0x00
Byte	*136:*	0x00	0x00	0x00	0x00	0x00	0x00	0x00	0x00
Byte	*144:*	0x00	0x00	0x00	0x00	0x00	0x00	0x00	0x00
Byte	*152:*	0x00	0x00	0x00	0x00	0x00	0x00	0x00	0x00
Byte	*160:*	0x00	0x00	0x00	0x00	0x00	0x00	0x00	0x00
Byte	*168:*	0x00	0x00	0x00	0x00	0x00	0x00	0x00	0x00
Byte	*176:*	0x00	0x00	0x00	0x00	0x00	0x00	0x00	0x00
Byte	*184:*	0x00	0x00	0x00	0x00	0x00	0x00	0x00	0x00
Byte	192:	0x00	0x00	0x00	0x00	0x00	0x00	0x00	0x00
Byte	200:	0x00	0x00	0x00	0x00	0x00	0x00	0x00	0x00
Byte	208:	0x00	0x00	0x00	0x00	0x00	0x00	0x00	0x00
Byte	216:	0x00	0x00	0x00	0x00	0x00	0x00	0x00	0x00
Byte	224:	0x00	0x00	0x00	0x00	0x00	0x00	0x00	0x00
Byte	232:	0x00	0x00	0x00	0x00	0x00	0x00	0x00	0x00
Byte	240:	0x00	0x00	0x00	0x00	0x00	0x00	0x00	0x00
Byte	248:	0x00	0x00	0x00	0x00	0x00	0x00	0x00	0x00
Byte	256:	0x00	0x00	0x00	0x00	0x00	0x00	0x00	0x00

```
B    2          0          0          0          0          0          0          0          0
y    6          x          x          x          x          x          x          x          x
t    4          0          0          0          0          0          0          0          0
e    :          0          0          0          0          0          0          0          0

B    2          0          0          0          0          0          0          0          0
y    7          x          x          x          x          x          x          x          x
t    2          0          0          0          0          0          0          0          0
e    :          0          0          0          0          0          0          0          0

B    2          0          0          0          0          0          0          0          0
y    8          x          x          x          x          x          x          x          x
t    0          0          0          0          0          0          0          0          0
e    :          0          0          0          0          0          0          0          0

B    2          0          0          0          0          0          0          0          0
y    8          x          x          x          x          x          x          x          x
t    8          0          0          0          0          0          0          0          0
e    :          0          0          0          0          0          0          0          0

B    2          0          0          0          0          0          0          0          0
y    9          x          x          x          x          x          x          x          x
t    6          0          0          0          0          0          0          0          0
e    :          0          0          0          0          0          0          0          0

B    3          0          0          0          0          0          0          0          0
y    0          x          x          x          x          x          x          x          x
t    4          0          0          0          0          0          0          0          0
e    :          0          0          0          0          0          0          0          0

B    3          0          0          0          0
y    1          x          x          x          x
t    2          0          0          f          f
e    :          0          0          f          f
        [root@orbiter-dut 4343]#
```

If the cisdump is verified to match the OTP binary map, the OTP programming is complete. Once programmed, additional blank spaces (00) in the OTP can still be written by filling in those corresponding blank spaces in the OTP binary map. There is no restriction on how many times a device can be programmed, provided that each programming is writing to only blank, unwritten spaces. Follow the same procedure to program the additional blank spaces.

A. Appendix

A.1 CIS Map

Table 6 and Table 7 list the CIS map (standard tuple tags and Cypress subtags) for SDIO devices.

Table 6. Standard Tuple Tags

Name	Tag	Length	Format	Variables	Description
CISTPL_VERS_1	0x15			manf	CIS version, manufacturer, device, and version strings.
				productname	
CISTPL_MANFID	0x20	4		manfid	Manufacturer and device ID.
				prodid	
CISTPL_FUNCID	0x21				Function identification
CISTPL_FUNCE	0x22				Function extensions
CISTPL_FUNCE	0x22	8			Subtype = FUNCE_mac(0x4), value: 6 bytes MAC address.
CISTPL_CFTABLE	0x1b	2		regwindowsz	Configuration table entry
CISTPL_FID_SDIO	0x0c				Extensions defined by SDIO specification

Table 6. Standard Tuple Tags

Name	Tag	Length	Format	Variables	Description
CISTPL_BRCM_HNBU	0x80				Cypress specific tuple subtag identifier
CISTPL_END	0xff				End of the CIS tuple chain

Table 7. Cypress Tuple Subtags

Name	Tag	Length	Format	Variables	Description
HNBU_SROMREV	0x00	1		sromrev	SROM revision
HNBU_CHIPID	0x01	4/6/8/10		vendid	Vendor and device ID
				devid	
				chiprev	
				subvendid	
				subdevid	
				boardtype	
HNBU_BOARDREV	0x02	1/2		boardrev	Board revision
HNBU_PAPARMS	0x03	2/8/9		pa0b0	PA parameters: 8 (sromrev = 1) or 9 (sromrev > 1) bytes
				pa0b1	
				pa0b2	
				pa0itssit	
				pa0maxpwr	
				opo	
HNBU_AA	0x06	1/2		aa2g	Antennas available
				aa5g	
HNBU_AG	0x07	1/2/3/4		ag0	Antenna gain
				ag1	
				ag2	
				ag3	
HNBU_BOARDFLAGS				boardflags	Board flags depend on the front-end module used. Please confirm this value with Cypress.
HNBU_CCODE	0x0a	3		ccode	Country code (2 bytes ASCII + 1 byte CCTL) The CCTL means indoor/outdoor, but it is never used.
				cctl	
HNBU_CCKPO	0x0b	2		cckpo	CCK power offsets
HNBU_OFDMPO	0x0c	4		ofdmpo	11g OFDM power offsets

Table 7. Cypress Tuple Subtags (Cont.)

Name	Tag	Length	Format	Variables	Description
HNBU_PAPARMS5G	0x0e	22		pa1b0	5G PA parameters for low/mid/high band.
				pa1b1	
				pa1b2	
				pa1lob0	
				pa1lob1	
				pa1lob2	
				pa1hib0	
				pa1hib1	
				pa1hib2	
				pa1itssit	
				pa1maxpwr	
				pa1lomaxpwr	
				pa1himaxpwr	
HNBU_ANT5G	0x0f	2		aa5g	5G antennas available/gain
				ag1	
HNBU_XTALFREQ	0x13	4		xtalfreq	Crystal frequency in kilohertz.
HNBU_TRI2G	0x14	1		triso2g	2G TR isolation
HNBU_TRI5G	0x15	3		triso5gl	5G TR isolation
				triso5g	
				triso5gh	
HNBU_RXPO2G	0x16	1		rxpo2g	2G RX power offset
HNBU_RXPO5G	0x17	1		rxpo5g	5G RX power offset
HNBU_BOARDNUM	0x18	2		boardnum	Board serial number, independent of MAC address.
HNBU_MACADDR	0x19	6		macaddr	MAC address override for the standard CIS LAN_NID.
HNBU_BOARDTYPE	0x1b	2		boardtype	Board type
HNBU_FEM	0x23	2/4		antswctl2g(15-11)	Front-end module variables.
				triso2g(10-8)	
				pdetrange2g(7-3)	
				extpagain2g(2-1)	
				tssipos2g(0)	
				antswctl5g(15-11)	
				triso5g(10-8)	
				pdetrange5g(7-3)	
				extpagain5g(2-1)	
				tssipos5g(0)	
HNBU_PO_CCKOFDM	0x28	6/18		cck2gpo	Power offset for 2G in CCK and OFDM.
				ofdm2gpo	
				ofdm5gpo	
				ofdm5glpo	
				ofdm5ghpo	

Table 7. Cypress Tuple Subtags (Cont.)

Name	Tag	Length	Format	Variables	Description
HNBU_OFDMPO5G	0x37	12		ofdm5gpo	Power offset for 5G in OFDM.
				ofdm5glpo	
				ofdm5ghpo	
HNBU_PO_MCS2G	0x29	16		mcs2gpo0	Power offset for 2G in modulation coding scheme (MCS) rate.
				mcs2gpo1	
				mcs2gpo2	
				mcs2gpo3	
				mcs2gpo4	
				mcs2gpo6	
				mcs2gpo7	
HNBU_PO_MCS5GM	0x2a	16		mcs5gpo0	Power offset for 5G mid-band in MCS rate.
				mcs5gpo1	
				mcs5gpo2	
				mcs5gpo3	
				mcs5gpo4	
				mcs5gpo6	
				mcs5gpo7	
HNBU_PO_MCS5GLH	0x2b	32		mcs5glpo0	Power offset for 5G low/high band in MCS rate.
				mcs5glpo1	
				mcs5glpo2	
				mcs5glpo3	
				mcs5glpo4	
				mcs5glpo6	
				mcs5glpo7	
				mcs5ghpo0	
				mcs5ghpo1	
				mcs5ghpo2	
				mcs5ghpo3	
				mcs5ghpo4	
				mcs5ghpo6	
				mcs5ghpo7	
HNBU_PO_40M	0x2e	2		bw40po	2g: bits 0–3
					5g: bits 4–7
					5gl: bits 8–11
					5gh: bits 12–15
HNBU_PO_40MDUP	0x2f	2		bwduppo	2g: bits 0–3
					5g: bits 4–7
					5gl: bits 8–11
					5gh: bits 12–15
HNBU_CCKFILTTYPE	0x36	1		cckdigfilttype	CCK digital filter selection option.

A.2 Driver Installation and Setup

This appendix provides steps:

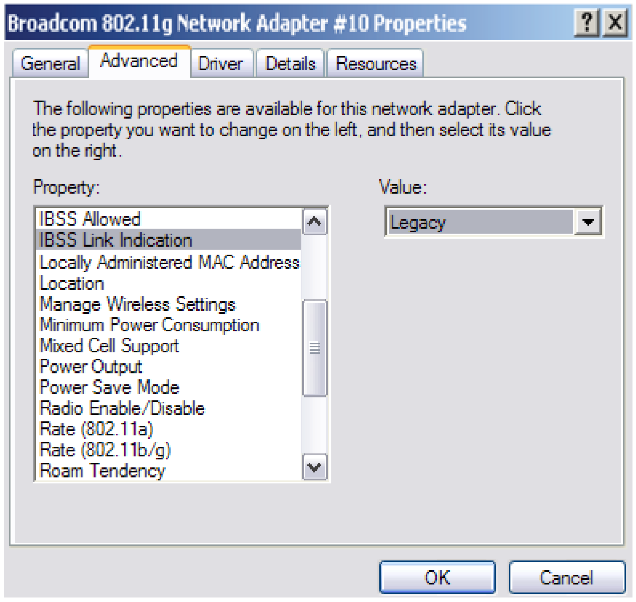
- To install a Windows XP driver for an SDIO device.
- To set the static IP address of the Cypress WLAN adapter.
- To install the Cypress WLAN tools for testing.

A.2.1 Installing an SDIO Driver for Windows XP

Note: In development environments where previous drivers have been installed, it may be necessary to uninstall a previously installed driver before proceeding with driver installation. If so, refer to Appendix A.3: "Driver Removal," on page 17.

To install an SDIO device driver:

1. Rename the NVRAM file (named after the board it supports) to "nvram.txt" and copy it to the C:\Windows\system32\drivers\ directory.
2. Turn off the power to the Windows XP-based PC test system.
3. Install the Cypress adapter in the PC.
4. Power-on the PC and allow Windows XP to start.
5. In Windows Control Panel, double-click **Administrative Tools**.
6. Within the **Administrative Tools**, double-click **Computer Management, and then click Device Manager.**
7. In the right pane of **Computer Management**, right-click **Network adapters**, and then click **Scan for hardware changes**.
8. Follow the Windows on-screen instructions to install the SDIO device driver.
9. Double-click on the newly installed network adapter to view the adapter properties.
10. On the **Advanced** tab of the **Network Adapter Properties**, set the **IBSS Link Indication** property value to **Legacy**, scroll down and set the **IBSS 54g™ Mode** property value to **54g-Auto**, and then click **OK**.



The following properties are available for this network adapter. Click the property you want to change on the left, and then select its value on the right.

Property:
IBSS Allowed
IBSS Link Indication
Locally Administered MAC Address
Location
Manage Wireless Settings
Minimum Power Consumption
Mixed Cell Support
Power Output
Power Save Mode
Radio Enable/Disable
Rate (802.11a)
Rate (802.11b/g)
Roam Tendency

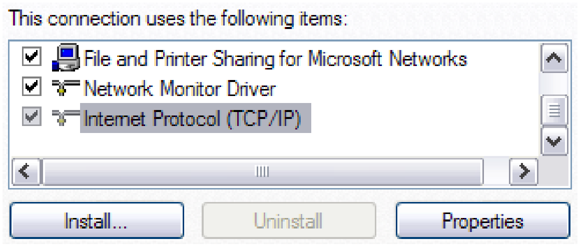
Value: Legacy

A.2.2 Setting a Static IP Address for the WLAN Adapter

To set the static IP address of the Cypress IEEE 802.11g SDIO WLAN adapter:

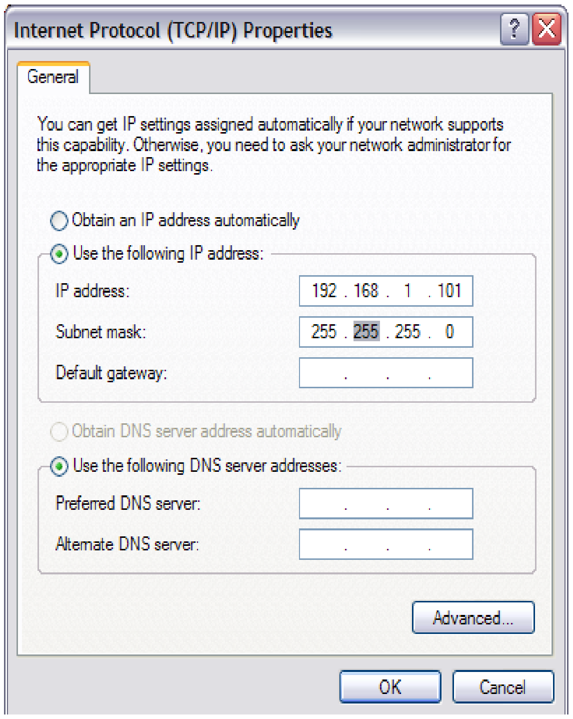
1. In Windows Control Panel, double-click **Network Connections**.
2. Right-click **Wireless Network Connection, and then** select **Properties**.

3. In **Wireless Network Connection Properties**, click **Internet Protocol (TCI/IP),** and then click the **Properties** button.


This connection uses the following items:

- [✓] File and Printer Sharing for Microsoft Networks
- [✓] Network Monitor Driver
- [✓] Internet Protocol (TCP/IP)

4. Select the **Use the following IP address** option.
5. Set the IP address to **192.168.1.101** and the Subnet mask to **255.255.255.0**, then click **OK**.


Internet Protocol (TCP/IP) Properties

General

You can get IP settings assigned automatically if your network supports this capability. Otherwise, you need to ask your network administrator for the appropriate IP settings.

( ) Obtain an IP address automatically
[✓ Use the following IP address:]

IP address:	192 . 168 . 1 . 101
Subnet mask:	255 . 255 . 255 . 0
Default gateway:	. . .

( ) Obtain DNS server address automatically
[✓ Use the following DNS server addresses:]

Preferred DNS server:	. . .
Alternate DNS server:	. . .

A.2.3 Installing the WLAN Test Tools

To install the WLAN test tools for enabling the driver-test commands, follow these instructions:

1. Copy the wl.exe file to the C:\Windows\system32\ directory.
2. Copy the brcm_wlu.dll file to the C:\Windows\system32\ directory.

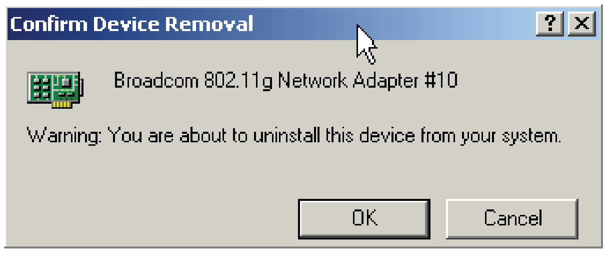
A.3 Driver Removal

A.3.1 Removing a Driver

In development environments where previous drivers have been installed, it may be necessary to remove a previously installed driver before proceeding with the new driver installation. Follow these instructions to remove a driver:

1. In Windows Control Panel, double-click **Administrative Tools**.
2. Within the **Administrative Tools**, double-click **Computer Management, and then click Device Manager.**
3. In the right pane of **Computer Management** under **Network adapters**, right-click **Broadcom 802.11g Network Adapter #10**, and then click **Uninstall**.

4. Click **OK** to **Confirm Device Removal**.



Broadcom 802.11g Network Adapter #10

Warning: You are about to uninstall this device from your system.

5. Delete the following files (refer to Using the nvram.txt File Template on page 4):

Filename	Location
bcmsddhd.sys	C:\Windows\system32\drivers\
nvram.txt	C:\Windows\system32\drivers\
oem#.inf	C:\Windows\inf\
oem#.pnf	C:\Windows\inf\
wl.exe	C:\Windows\system32\
brcm_wlu.dll	C:\Windows\system32\

The following files are typical SDIO driver files released with the Windows XP driver or design package:

nvram.txt
bcmsddhd.inf
bcmsddhd.sys
brcm_wlu.dll
wl.exe

8 Deleting the oem#.inf and oem#.pnf Files

In some cases, multiple instances of Cypress network adapters might be installed. To locate the correct oem#.inf and oem#.pnf files for deletion, follow these steps:

1. Start the Windows **Search option**.
2. In the **Search Companion** pane of **Search Results**, type **C:\Windows\inf\oem*.inf.** In the **All or part of the file name** box, type **Broadcom Corporation.** In the **A word or phrase in the file** box, select **Local Hard Drives (C:)** from the **Look in** list, and then click **Search**.
3. If more than one INF file appears in the **Search Results**, open each in a text editor.

Delete the Broadcom oem#.inf file and the associated Broadcom oem#.pnf file that looks similar to the oem#.inf file shown here.

```
;; bcmsddhd.inf
;;
;; Copyright 1998-2005, Broadcom Corporation.
;; All Rights Reserved.
;;
;; This is UNPUBLISHED PROPRIETARY SOURCE CODE of Broadcom Corporation;
;; the contents of this file may not be disclosed to third parties, copied or
;; duplicated in any form, in whole or in part, without the prior written
;; permission of Broadcom Corporation.
;;

[version]
    Signature   = "$Windows NT$"        ; Combined Win9x/Win2k inf
    Class=Net
    ClassGUID   = {4d36e972-e325-11ce-bfc1-08002be10318}
    Provider    = %V_BCM%
    Compatible  = 1
DriverVer=06/10/2009, 4.218.84.1
    CatalogFile=BCM43XX.CAT
    CatalogFile.NTamd64=BCM43XX64.CAT

[Manufacturer]
    %V_BCM% = BROADCOM, NTamd64
```

Document History Page

Document Title: AN214807 - OTP Programming and NVRAM Development in SDIO Mode Document Number: 002-14807				
Rev.	**ECN No.**	**Orig. of Change**	**Submission Date**	**Description of Change**
**	–	–	06/10/2014	4343X-AN100-R Initial release
*A	–	–	09/29/2014	4343X-AN101-R Updated: Figure 2: “Example OTP Binary Map Containing Basic Parameters,” on page 8. Creating and Editing the OTP Binary Map on page 8. Figure 3: “Hexadecimal OTP Map Template,” on page 9. Programming Procedure Using wl Commands on page 9
*B	5445052	UTSV	09/30/2016	Updated in Cypress template Added Cypress part numbering scheme
*C	5803968	AESATP12	07/07/2017	Updated logo and copyright.

Worldwide Sales and Design Support

Worldwide Sales and Design Support

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

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
Memory	cypress.com/memory
Microcontrollers	cypress.com/mcu
PSoC	cypress.com/psoc
Power Management ICs	cypress.com/pmic
Touch Sensing	cypress.com/touch
USB Controllers	cypress.com/usb
Wireless Connectivity	cypress.com/wireless

PSoC® Solutions

PSoC 1 | PSoC 3 | PSoC 4 | PSoC 5LP | PSoC 6

Cypress Developer Community

Forums | WICED IOT Forums | Projects | Video | Blogs | Training | Components

Technical Support

cypress.com/support

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