AN214808

OTP Programming and NVRAM Development in SDIO Mode - CYW4354/ CYW4356/BCM4358

Associated Part Family: CYW4354/CYW4356/BCM4358

This application note describes the method for creating an nvram.txt file, which is then used to test a new board design, optimize NVRAM values, and program the one-time programmable (OTP) nonvolatile memory in a device that uses a WLAN SDIO host interface.

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 memory programming flow
- SDIO driver installation
- Customizing the nvram.txt file
- OTP memory programming procedure

1.2 Before You Begin

It is recommended that the users of this application note request the following items from Cypress's Customer Support Portal (CSP):

- A CYW4354, CYW4356, or BCM4358 SDIO board reference design package that contains:
  - The reference board schematic, bill of materials, and layout.
  - An nvram.txt template file for the reference board.
  - Make sure GPIO_4 is pulled lowso that OTP memory can be programmed. Customers should work with the Cypress HW AE team to configure this pin correctly.
- A device driver for the relevant SDIO device
- Cypress transmit signal strength indicator (TSSI) calibration tools

1.3 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
BCM20734	CYW4354
BCM4356	CYW4356

1.4 Acronyms and Abbreviations

In most cases, acronyms and abbreviations are defined on first use. For a more complete list of acronyms and other terms used in Cypress documents, go to: http://www.cypress.com/glossary.

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 CYW4354, CYW4356, and BCM4358 are single-chip IEEE 802.11ac 2×2 MIMO WLAN + BT4.1/FM RX devices for embedded applications. One-time programmable (OTP) nonvolatile memory is included in the WLAN section of the devices for storing board-specific information such as product ID, manufacturer ID, and MAC address. Excluding the internal header information, up to 484 bytes of user-accessible OTP memory is available on the CYW4354, CYW4356, and BCM4358 for WLAN information. Although the WLAN section provides the option of using an SDIO or PCIe host interface, this application note addresses only SDIO applications. See *OTP Programming and NVRAM Development in PCIe Mode* (Cypress document number 4356-AN10x-R) for CYW4356 PCIe-based applications.

The OTP memory content, along with an editable NVRAM file (nvram.txt file), provides all the configuration information used by the WLAN device driver to initialize and configure the CYW4354, CYW4356, and BCM4358.

4 OTP Memory Programming Considerations

In embedded designs, where the host and device are permanently connected, programming the OTP memory is optional. All NVRAM parameters can be stored in host nonvolatile memory rather than OTP memory. Optionally, some items, such as power control related variables may be programmed into OTP memory to make them permanent.

For non-embedded devices that may be installed on different hosts, the OTP memory can be programmed to protect the unique MAC address and prevent end-users from altering the power-control parameters such as maximum output power.

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

Because the OTP memory programming process is irreversible, Cypress recommends that board designers finalize all NVRAM parameters before programming any of them into the OTP memory. Boards and modules should be tested using only the editable nvram.txt file.

The parameters stored in the nvram.txt file are loaded into on-chip RAM by the driver, allowing the chip to be tested even if the OTP memory has not been programmed. This method lets board designers tune the RF components and alter critical parameters using different versions of the nvram.txt file while testing boards. Optionally, a few basic parameters, such as the board type and MAC address, can be programmed into the OTP memory prior to board testing during development.

Note: If a parameter is present in both the on-chip OTP memory and the nvram.txt file, the value in the OTP memory takes priority over the value in the nvram.txt file.

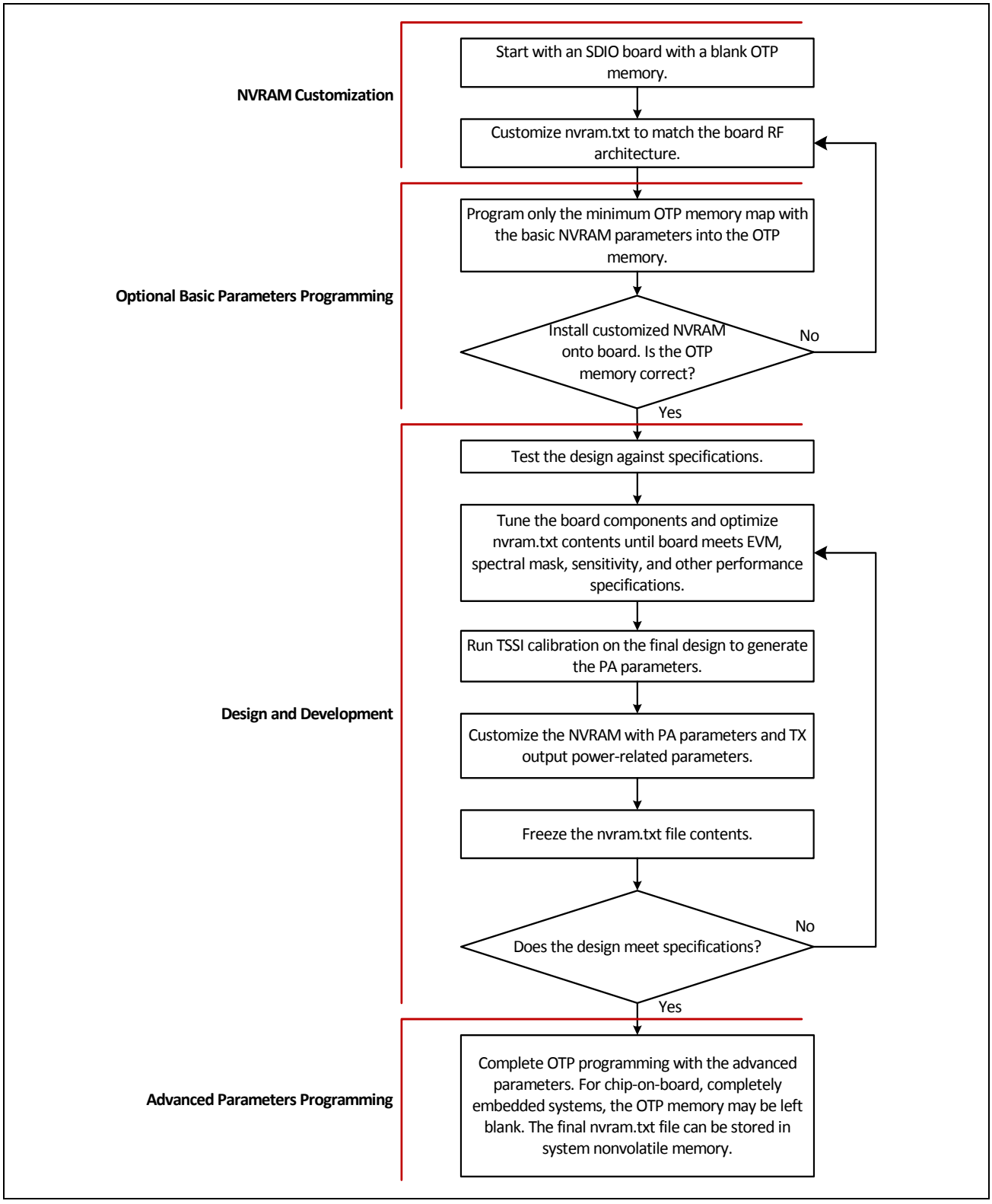
Caution! The OTP memory programming process is irreversible. Cypress strongly recommends conducting development on boards using the parameters provided in the editable nvram.txt file. Do not program the OTP memory until the contents of the nvram.txt file have been verified and the file has been finalized for production use.

5 NVRAM Content Development and OTP Memory Programming Flow

Figure 1 on page 4 shows the nvram.txt file content development and the OTP memory programming flow. Parameters in the nvram.txt file can be divided into basic and advanced categories.

Note: The NVRAM development and OTP memory programming flow shown in Figure 1 should be conducted on small quantities of boards/modules during the product development stage. Once this process is complete and the production version of the nvram.txt file and OTP memory file are approved for production use, programming can begin for high volume mass production as defined by each manufacturer.

Figure 1. NVRAM Development and OTP Memory Programming Flow



6 Customizing the nvram.txt File

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

6.1 Using the nvram.txt File Template

For each Cypress reference board design, Cypress provides an nvram.txt file for the specific board design. Typically, the file is named in accordance with the board it supports (for example, bcm94356wlsagbl.txt).

The nvram.txt file might be included with the reference board design package or the driver release. The latest version of the file can be downloaded from the Cypress developer community.

Note: After editing an nvram.txt file, save the file, then disable and reenable the wireless device driver using the Windows Device Manager for the change to take effect. Save the file as nvram.txt and store it in the C:\Windows\system32\drivers\ directory. Delete or overwrite any previous version of the file located in this directory.

Table 2 and Table 3 on page 6 provide a list of parameters in a typical nvram.txt file that are common to Cypress dual-band IEEE 802.11ac 2×2 MIMO SDIO reference design boards.

Parameters in the nvram.txt file do no need to be entered in any specific order.

Note: The parameters listed in Table 2 are used and specified by Cypress and should only be changed by Cypress. It is important that a customer's design is reviewed by Cypress early in the development process. Some of the parameters in Table 2 may need to be changed by Cypress in order to accommodate differences in the RF front end between a customer design and the Cypress reference design from which it was derived.

Table 2. Cypress-Specific NVRAM Parameters

NVRAM Parameter	Example Data	Description
sromrev	11	SROM revision for IEEE 802.11ac chips.
boardtype	0x0703	This is a critical parameter that should be copied from a similar Cypress reference board design.
tssipos2g	1	This represents if TSSI has positive slope for 2.4 GHz. Set this to 1.
tssipos5g	1	This represents if TSSI has positive slope for 5 GHz. Set this to 1.
rxchain	3	This specifies the number of RX paths (bit mask). Set this to 3.
txchain	3	This specifies the number of TX paths (bit mask). Set this to 3.
antswitch	0	This enables switch-based diversity. • 0: disable • 1: enable
paprdis	0	This parameter is for Cypress internal use only. **Note:** Do not modify.
pdgain5g pdgain2g	4 4	5 GHz and 2.4 GHz power detector parameter used by the driver to program the TSSI loopback path. **Note:** Do not modify.
vendid	0x14E4	Vendor ID
devid	0x43EC	Chip ID, CYW4356
manfid	0x2D0	Manufacturer ID
nocrc	1	Check for CRC errors when loading FW.
boardflags boardflags2 boardflags3	0x12401001 0x00802000 0x4800018A	Board configuration flag that defines the power topology, external components (iPA, eLNA), etc.
tworangetssi2g tworangetssi5g	0 0	2.4 GHz and 5 GHz TSSI dual power range flag, which iPA chips support.
femctrl	10	Defines front-end RF switch or front-end module (FEM) control logic for both bands.
xtalfreq	37400	Describes the reference oscillator frequency in kHz. 37400 stands for 37.4 MHz.
extpagain2g	2	Support 2.4 GHz external PA. Use 2 for iPA boards, and use 1 for ePA boards.

Table 2. Cypress-Specific NVRAM Parameters (Cont.)

NVRAM Parameter	Example Data	Description
extpagain5g	2	Supports 5 GHz external PA. Use 2 for iPA boards, and use 1 for ePA boards
aa2g, aa5g	3	Number of antennas available for the 2.4 GHz and 5 GHz bands, respectively, in bit-mapped binary format: • 1 = 01b for one antenna • 3 = 11b for two antennas (applies to CYW4356)
subband5gver	0x4	Defines 5 GHz subband allocation
tempthresh	120	This parameter is for Cypress internal use only.
tempoffset	255	This parameter is for Cypress internal use only.
rawtempsense	0x1FF	This parameter is for Cypress internal use only. **Note:** Do not modify.
phycal_tempdelta	25	This parameter is for Cypress internal use only.
temps_period	15	This parameter is for Cypress internal use only.
temps_hysteresis	15	This parameter is for Cypress internal use only.
AvVmid_c0, AvVmid_c1	2,140, 2,145, 2,145, 2,145, 2,145	Cypress internal use only. **Note:** Do not modify.
swctrlmap_2g, swctrlmap_5g	0x02020202, 0x05050404, 0x04040000, 0x000000, 0x047	Describes how to control the external 2.4 GHz and 5 GHz FEM or TR-SW (TR switch). **Note:** Do not modify.

The design variables listed in Table 3 must be reviewed prior to beginning board or module testing. During the development phase, start with the default power amplifier (PA) parameters contained in the provided nvram.txt file. The PA parameters are eventually optimized using Cypress TSSI calibration tools.

Note: The parameters in Table 3 typically require tuning for each specific board or module design. This is not an exhaustive list. Additional parameters may be added by Cypress at any time to control the RF performance-related attributes of the driver. Always check with Cypress for the latest reference design nvram.txt file before starting any board customization efforts.

Note: To avoid unexpected operating results, contact a technical support representative before attempting to add NVRAM parameters.

Table 3. NVRAM Parameters Requiring Customization

NVRAM Parameter	Example Data	Description
boardrev	0x1102	Board revision used by the WLAN driver. Examples: 0x1102 converts to P102 0x1210 converts to P210
ccode	0	Country code for regulatory. Specifies which regulatory tables are to be loaded. **Note:** Together, the `ccode` and `regrev` parameters set the power and other limitations necessary to meet the country-specific regulatory requirements.
regrev	0	The regulatory revision code for regulatory use, and which regulatory tables are to be loaded. **Note:** Together, the `ccode` and `regrev` parameters set the power and other limitations necessary to meet the country-specific regulatory requirements.
rxgains2gtrelnabypa0 rxgains2gtrelnabypa1	1	This variable defines the isolation that a 2.4 GHz eLNA provides when put in bypass mode. 'a0' and 'a1' apply for Core 0 and Core 1, respectively.

Table 3. NVRAM Parameters Requiring Customization (Cont.)

NVRAM Parameter	Example Data	Description
rxgains5gtrelnabypa0 rxgains5gtrelnabypa1	1	This variable defines the isolation that the 5 GHz eLNA provides when put in bypass mode. 'a0' and 'a1' apply for Core 0 and Core 1, respectively. Apply to low subband.
rxgains5gmtrelnabypa0 rxgains5gmtrelnabypa1	1	This variable defines the isolation that the 5 GHz eLNA provides when put in bypass mode. 'a0' and 'a1' apply for Core 0 and Core 1, respectively. Apply to mid subband.
rxgains5ghtrelnabypa0 rxgains5ghtrelnabypa1	1	This variable defines the isolation that the 5 GHz eLNA provides when put in bypass mode. 'a0' and 'a1' apply for Core 0 and Core 1, respectively. Apply to high/X1 subband.
rxgains2gelnagaina0 rxgains2gelnagaina1	3	This variable defines the 2.4 GHz eLNA gain in dB. 'a0' and 'a1' apply to Core 0 and Core 1, respectively.
rxgains2gtrisoa0 rxgains2gtrisoa1	6	This variable defines the 2.4 GHz isolation that the TR switch provides when in "T" mode. 'a0' and 'a1' apply to Core 0 and Core 1, respectively.
rxgains5gelnagaina0 rxgains5gelnagaina1	3	This variable defines the 5 GHz eLNA gain in dB. 'a0' and 'a1' apply to Core 0 and Core 1, respectively. Applies to low subband.
rxgains5gtrisoa0 rxgains5gtrisoa1	6	This variable defines the 5 GHz isolation that the TR switch provides when in "T" mode. 'a0' and 'a1' apply to Core 0 and Core 1, respectively. Applies to low subband.
rxgains5gmelnagaina0 rxgains5gmelnagaina1	3	This variable defines the 5 GHz eLNA gain in dB. 'a0' and 'a1' apply to Core 0 and Core 1, respectively. Applies to mid subband.
rxgains5gmtrisoa0 rxgains5gmtrisoa1	6	This variable defines the 5 GHz isolation that the TR switch provides when in "T" mode. 'a0' and 'a1' apply to Core 0 and Core 1, respectively. Applies to mid subband.
rxgains5ghelnagaina0 rxgains5ghelnagaina1	3	This variable defines the 5 GHz eLNA gain in dB. 'a0' and 'a1' apply to Core 0 and Core 1, respectively. Applies to high/X1 subband.
rxgains5ghtrisoa0 rxgains5ghtrisoa1	6	This variable defines the 5 GHz isolation that the TR switch provides when in "T" mode. 'a0' and 'a1' apply for Core 0 and Core 1, respectively. Applies to high/X1 subband.
agbg0 aga0 agbg1 aga1	0X7F	Antenna gain (in dBi) defined by converting hexadecimal to 8-bit binary: (agba0: 2.4 GHz antenna gain, aga0: 5 GHz antenna gain) • Lower 0–5 bits = signed 2's complement in units of dB. • Higher 6–7 bits = unsigned number in units of quarter dB. Suffices '0' and '1' apply for Core 0 and Core 1, respectively. Examples: 0x82 = 2.5 dB (2 + 2 × 0.25) 0X7F = –0.75 dB (–1 + 1 × 0.25)
pa2ga0 pa2ga1 pa2gccka0 pa2gccka1	–148, 5828, –679	PA parameters for the 2.4 GHz band based on TSSI calibration. pa2ga0/a1 – OFDM (Orthogonal Frequency Division Multiplexing) pa2gccka0/a1 – CCK. 'a0' and 'a1' apply for Core 0 and Core 1, respectively.
pa5ga0 pa5ga1	83, 6045, –553, 57, 5940, –566, 12, 5919, –605, –17, 5899, –640	PA parameters for the 5 GHz band based on TSSI calibration (Low/Mid/High/X1 subband frequency range). Channel range: • Low 5180 to 5240: 36–48 • Mid 5260 to 5320: 52–64 • High 5500 to 5700: 100–140 • X1 5745 to 5825: 149–165 (pa5ga0/a1). 'a0' and 'a1' apply for Core 0 and Core 1, respectively.

Table 3. NVRAM Parameters Requiring Customization (Cont.)

NVRAM Parameter	Example Data	Description
pdoffset40ma0 pdoffset40ma1	0x0000	5 GHz, 40 MHz BW power detector (PD) offset (1/4 dB steps) in 2's complement format, 4 bits for each subband. 'a0' and 'a1' apply for Core 0 and Core 1, respectively.
pdoffset80ma0 pdoffset80ma1	0x0000	5 GHz, 80 MHz BW PD offset (1/4 dB steps) in 2's complement format, 4 bits for each subband. 'a0' and 'a1' apply for Core 0 and Core 1, respectively.
maxp2ga0 maxp2ga1	0x46	Maximum output power for the 2.4 GHz band in hexadecimal format. Units of 0.25 dB. This applies to all complementary code keying (CCK) rates as measured at the antenna port. The nominal target power in dBm for CCK packets is: (0.25 × maxp2ga0 in decimal) – 1.5 dB. The value can be entered in either hexadecimal or decimal formats. For the 0x46 example, the maximum output power is: (16 x 4 + 6)/4 = 17.5 dBm, and the nominal power is: 17.5 – 1.5 = 16.0 dBm. 'a0' and 'a1' apply for Core 0 and Core 1, respectively.
cckbw202gpo	0x0000	CCK power offsets for 20 MHz rates (11, 5.5, 2, and 1 Mbps).
cckbw20ul2gpo	0x0000	CCK power offsets for 20 U/L rates (11, 5.5, 2, and 1 Mbps).
dot11agofdmhrbw202gpo	0x6666	OFDM power offset. Specified in half-dBm units: 54, 48, 36, and 24 Mbps.
ofdmlrbw202gpo	0x0033	OFDM 2.4 GHz power offset. Specified in half-dBm units: • MCS1 and MCS2: 11n and 11ac 40 MHz BW • MCS1 and MCS2: 11n and 11ac 20 MHz BW • 12 and 18 Mbps: 11g • 6 and 9 Mbps: 11g
mcsbw202gpo	0xAA886664	11n/ac MCS0/1/2, 3-7, C8, C9 2.4 MHz power offset. Specified in half dBm units – C9/C8/M7/M6/M5/M4/M3/M0-2. (If separate control of MCS1 and MCS2 is required, then use `ofdmlrbw202gpo`).
maxp5ga0 maxp5ga1	0x4A, 0x4A, 0x4A, 0x4A	Maximum output power for the 5 GHz band in hexadecimal format. Units of 0.25 dB. This applies to all legacy OFDM rates as measured at the antenna port. The nominal target power in dBm is (0.25 × `maxp5ga0` in decimal) – 1.5 dB. The value can be entered in either hexadecimal or decimal format. 'a0' and 'a1' apply for Core 0 and Core 1, respectively.
mcslr5glpo	0x0000	5 GHz band low subband 12/18 & M1/M2: • (0) 20 MHz • (1) 40 MHz • (2) 80 MHz • (3) 160 MHz
mcsbw205glpo	0xAA886662	5 GHz low band 11n/ac MCS0/ 1/2, 3-7, C8, C9 power offset for 20 MHz BW – C9/C8/M7/M6/M5/M4/M3/M0-2.
mcsbw405glpo	0xAA886664	5 GHz low band 11n/ac MCS0/ 1/2, 3-7, C8, C9 power offset for 40 MHz BW – C9/C8/M7/M6/M5/M4/M3/M0-2.
mcsbw805glpo	0xAA886664	5 GHz low band 11n/ac MCS0/ 1/2, 3-7, C8, C9 power offset for 80 MHz BW – C9/C8/M7/M6/M5/M4/M3/M0-2.
mcslr5gmpo	0x0000	5 GHz band mid subband 11ag/11n/11ac QPSK power offset with respect to BPSK - mcs 1/2 with respect to mcs 0/ 1/2 and 12/18 Mbps with respect to 6/9 Mbps. LSB to MSB nibble: • (0) 20 MHz • (1) 40 MHz • (2) 80 MHz • (3) 160 MHz
mcsbw205gmpo	0xAA886664	5 GHz mid band 11n/ac MCS0/ 1/2, 3-7, C8, C9 power offset for 20 MHz – C9/C8/M7/M6/M5/M4/M3/M0-2.
mcsbw405gmpo	0xAA886664	5 GHz mid band 11n/ac MCS0/ 1/2, 3-7, C8,C9 power offset for 40 MHz – C9/C8/M7/M6/M5/M4/M3/M0-2.
mcsbw805gmpo	0xAA886664	5 GHz mid band 11n/ac MCS0/ 1/2, 3-7, C8, C9 power offset for 80 MHz – C9/C8/M7/M6/M5/M4/M3/M0-2.

Table 3. NVRAM Parameters Requiring Customization (Cont.)

NVRAM Parameter	Example Data	Description
mcslr5ghpo	0x0000	5 GHz band high/X1 subband 11ag/11n/11ac QPSK power offset with respect to BPSK - mcs 1/2 with respect to mcs 0/1/2 and 12/18 Mbps with respect to 6/9 Mbps. LSB to MSB nibble: • (0) 20 MHz • (1) 40 MHz • (2) 80 MHz • (3) 160 MHz
mcsbw205ghpo	0xAA886664	5 GHz high/X1 band 11n/ac MCS0/1/2,3-7,C8,C9 power offset for 20 MHz – C9/C8/M7/M6/M5/M4/M3/M0-2
mcsbw405ghpo	0xAA886664	5 GHz high/X1 band 11n/ac MCS0/1/2,3-7,C8,C9 power offset for 40 MHz – C9/C8/M7/M6/M5/M4/M3/M0-2
mcsbw805ghpo	0xAA886664	5 GHz high/X1 band 11n/ac MCS0/1/2,3-7,C8,C9 power offset for 80 MHz – C9/C8/M7/M6/M5/M4/M3/M0-2
sb20in40hrrpo	0	20in40 OFDM signed power offsets with respect to 20in20 for 64 QAM and above. LSB nibble to MSB nibble: • (0) 2.4 GHz band • (1) 5 GHz low subband • (2) 5 GHz mid subband • (3) 5 GHz high subband
sb20in80and160hr5glpo	0	5 GHz low subband 20in80, 20in160 OFDM signed power offsets for 64 QAM and above. LSB nibble to MSB nibble: • (0) 20in80 with respect to 20in20 • (1) 20in160 with respect to 20in20 • (2) 20in80 - 20LL/UU with respect to 20LU/UL • (3) 20in160 - 20LLL/UUU with respect to other 20in160 subbands
sb40and80hr5glpo	0	5 GHz low subband 40in80, 40in160 OFDM signed power offsets for 64 QAM and above. LSB nibble to MSB nibble: • (0) 40in80 with respect to 40in40 • (1) 40in160 with respect to 40in40 • (2) 80in160 with respect to 80in80 • (3) 40in160 -40LL/UU with respect to 40LU/UL
sb20in80and160hr5gmpo	0	5 GHz mid subband 20in80, 20in160 OFDM signed power offsets for 64 QAM and above. LSB nibble to MSB nibble: • (0) 20in80 with respect to 20in20 • (1) 20in160 with respect to 20in20 • (2) 20in80 - 20LL/UU with respect to 20LU/UL • (3) 20in160 - 20LLL/UUU with respect to other 20in160 subbands
sb40and80hr5gmpo	0	5 GHz mid subband 40in80, 40in160 OFDM signed power offsets for 64 QAM and above. LSB nibble to MSB nibble: • (0) 40in80 with respect to 40in40 • (1) 40in160 with respect to 40in40 • (2) 80in160 with respect to 80in80 • (3) 40in160 -40LL/UU with respect to 40LU/UL
sb20in80and160hr5ghpo	0	5 GHz high/X1 subband 20in80, 20in160 OFDM signed power offsets for 64 QAM and above. LSB nibble to MSB nibble: • (0) 20in80 with respect to 20in20 • (1) 20in160 with respect to 20in20 • (2) 20in80 - 20LL/UU with respect to 20LU/UL • (3) 20in160 - 20LLL/UUU with respect to other 20in160 subbands
sb40and80hr5ghpo	0	5 GHz high/X1 subband 40in80, 40in160 OFDM signed power offsets for 64 QAM and above. LSB nibble to MSB nibble: • (0) 40in80 with respect to 40in40 • (1) 40in160 with respect to 40in40 • (2) 80in160 with respect to 80in80 • (3) 40in160 -40LL/UU with respect to 40LU/UL

Table 3. NVRAM Parameters Requiring Customization (Cont.)

NVRAM Parameter	Example Data	Description
sb20in40lrpo	0	20in40 OFDM signed power offsets with respect to 20in20 for 16 QAM and below. LSB nibble to MSB nibble: • (0) 2.4 GHz band • (1) 5 GHz low subband • (2) 5 GHz mid subband • (3) 5 GHz high/X1 subband
sb20in80and160lr5glpo	0	5 GHz low subband 20in80, 20in160 OFDM signed power offsets for 16 QAM and below. LSB nibble to MSB nibble: • (0) 20in80 with respect to 20in20 • (1) 20in160 with respect to 20in20 • (2) 20in80 - 20LL/UU with respect to 20LU/UL • (3) 20in160 - 20LLL/UUU with respect to other 20in160 subbands
sb40and80lr5glpo	0	5 GHz low subband 40in80, 40in160 OFDM signed power offsets for 16 QAM and below. LSB nibble to MSB nibble: • (0) 40in80 with respect to 40in40 • (1) 40in160 with respect to 40in40 • (2) 80in160 with respect to 80in80 • (3) 40in160 -40LL/UU with respect to 40LU/UL
sb20in80and160lr5gmpo	0	5 GHz mid subband 20in80, 20in160 OFDM signed power offsets for 16 QAM and below. LSB nibble to MSB nibble: • (0) 20in80 with respect to 20in20 • (1) 20in160 with respect to 20in20 • (2) 20in80 - 20LL/UU with respect to 20LU/UL • (3) 20in160 - 20LLL/UUU with respect to other 20in160 subbands
sb40and80lr5gmpo	0	5 GHz mid subband 40in80, 40in160 OFDM signed power offsets for 16 QAM and below. LSB nibble to MSB nibble: • (0) 40in80 with respect to 40in40 • (1) 40in160 with respect to 40in40 • (2) 80in160 with respect to 80in80 • (3) 40in160 -40LL/UU with respect to 40LU/UL
sb20in80and160lr5ghpo	0	5 GHz high/X1 subband 20in80, 20in160 OFDM signed power offsets for 16 QAM and below. LSB nibble to MSB nibble: • (0) 20in80 with respect to 20in20 • (1) 20in160 with respect to 20in20 • (2) 20in80 - 20LL/UU with respect to 20LU/UL • (3) 20in160 - 20LLL/UUU with respect to other 20in160 subbands
sb40and80lr5ghpo	0	5 GHz high/X1 subband 40in80, 40in160 OFDM signed power offsets for 16 QAM and below. LSB nibble to MSB nibble: • (0) 40in80 with respect to 40in40 • (1) 40in160 with respect to 40in40 • (2) 80in160 with respect to 80in80 • (3) 40in160 - 40LL/UU with respect to 40LU/UL
dot11agduphrpo	0	11a/g duplicate mode signed power offsets for 64 QAM. Common power offset for Dup40, Dup40in80, and Dup40in160 with respect to 40in40 11n/11ac, Quad80 and Quad80in160 with respect to 11ac 80in80, Oct160 with respect to 11ac 160in160. LSB to MSB nibble: • (0) 2.4 GHz band • (1) 5 GHz low subband • (2) 5 GHz mid subband • (3) 5 GHz high/X1 subband
dot11agduplrpo	0	Bits 11a/g duplicate mode signed power offsets for 16 QAM and below. Common power offset for Dup40, Dup40in80, and Dup40in160 with respect to 40in40 11n/11ac, Quad80 and Quad80in160 with respect to 11ac 80in80, Oct160 with respect to 11ac 160in160. LSB to MSB nibble: • (0) 2.4 GHz band • (1) 5 GHz low subband • (2) 5 GHz mid subband • (3) 5 GHz high/X1 subband
mux_enab	0x11	Specifies GPIO pin for OOB interrupts.
btc_mode	1	Specifies BT-COEX mode.

Table 3. NVRAM Parameters Requiring Customization (Cont.)

NVRAM Parameter	Example Data	Description
ltecxmux	0x534201	Specifies LTE coexistence settings.
cckdigfilttype	2	Specifies filter type for IEEE 802.11b mode.
ofdmfilttype	1	Specifies filter type for OFDM.
rssicorrnorm_c0 rssicorrnorm_c1	–2, 0	Adds specified offset in the raw 2.4 GHz RSSI value. The final RSSI value will be raw RSSI plus this offset. First value is for 20 MHz channels and 2nd for 40 MHz channels (20M / 40M). **Note:** Applies to Core 0 and Core 1, respectively.
rssicorrnorm5g_c0 rssicorrnorm5g_c1	2, 3, 0, 1, 1, –1, –2, 0, –2, –2, 0, –2	Adds specified offset in the raw 5 GHz RSSI value. The final RSSI value will be raw RSSI plus this offset. The first three values are for the first subband of 5 GHz, each corresponding to 20, 40, and 80 MHz. The next three are for the 2nd subband and so on. In the example, the first value is for subband 1/20 MHz and the last value is subband 4/80 MHz. (Low 20 / Low 40 / Low 80 / Mid 20 / Mid 40 / Mid 80 / High 20 / High 40 / High 80 / X1 20 / X1 40 / X1 80). 'c0' and 'c1' apply for Core 0 and Core 1, respectively.
powoffs2gtna0 powoffs2gtna1	–3, –2, 0, 0, 0, 0, 0, 0, 0, 0, 0, –5, –5	Specifies power offset per channel in 2.4 GHz (channel 1 to 13).
powoffs5g20mtna0 powoffs5g20mtna1	–2, –2, –2, –2, –3, –3	Specifies power offset for band-edge channels, 5 GHz, 20 MHz BW [36, 64, 100, 140, 149, 165].
powoffs5g40mtna0 powoffs5g40mtna1	–2, –2, –3, –3	Specifies power offset for band-edge channels, 5 GHz, 40 MHz BW [38, 62, 102, 151].
powoffs5g80mtna0 powoffs5g80mtna1	–2, –2, –3, –3	Specifies power offset for band-edge channels, 5 GHz, 80 MHz BW [42, 58, 106, 155].
tssifloor5g	220, 213, 218, 228	Enables meeting input-overload specifications of external 5 GHz PAs.
tssifloor2g	245	Enables meeting input-overload specifications of external 2.4 GHz PAs.
txidxcap2g	15	Enables meeting input-overload specifications of external 2.4 GHz PAs.
txidxcap5g	0	Enables meeting input-overload specifications of external 5 GHz PAs.

6.2 Editing the nvram.txt File

The nvram.txt file should be edited using a properly formatted text editor such as Notepad++ or WordPad++ to preserve the original format of the file. Using a non-formatted text editor such as Notepad could corrupt the format of the NVRAM map, causing the driver to incorrectly read the nvram.txt file.

6.3 Finalizing the nvram.txt File

After the final PA parameters have been generated, edit the nvram.txt file to update the PA parameters derived using the Cypress TSSI tool, and then adjust the TX output power-related parameters in the file. Using the updated nvram.txt file, run output power tests to verify that the parameters are providing the correct output power. Also, verify that RF performance (EVM, spectral mask, and rxper) meets the 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 finalized, the needed parameters can be selected and the OTP memory can be programmed for production.

Note: The CYW4354, CYW4356, and BCM4358 each has 484 bytes of OTP memory available for user data. Given the limited space in the OTP memory, it is impossible to program the entire nvram.txt file to the OTP memory. The programmer must be very careful to select only the necessary parameters that go into the OTP memory.

Parameters that typically go into the OTP memory 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, and country code. Alternately, with many embedded systems, all of the various NVRAM variables are stored in the system's nonvolatile memory as opposed to OTP memory.

7 Programming OTP Memory

Prior to programming the OTP memory, an OTP binary map file must be prepared and populated with the correct values. The OTP binary map completely defines the parameters that have to be programmed into the OTP memory. The SDIO OTP memory 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 (tuples), where each tuple contains the type, length, and the value of the tuple.

The SDIO hardware header string must be present at the beginning of the OTP binary map and must precede all NVRAM variables. When a driver detects content in the OTP memory, the SDIO hardware header is required to boot up the device via the SDIO interface. Therefore, the SDIO hardware header must include the minimum set of parameters necessary to program the OTP memory.

For an example of the SDIO hardware header, see the first 11 bytes of Figure 2 on page 14. All other parameters to be programmed into the OTP memory are added after the SDIO hardware header (see Creating and Editing the OTP Memory Binary Map on page 15).

When an OTP binary map contains only the SDIO hardware header, the binary map is called a minimum OTP binary map.

7.1 Programming the Basic Parameters into OTP Memory

Parameters in the nvram.txt file that are to be programmed into the OTP memory must be entered in the OTP binary map after the SDIO hardware header. A CIS tuple is required for each parameter 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.

Table 4 lists the basic NVRAM parameters, the associated tag number, and the number of bytes each parameter occupies in the OTP memory. Basic parameters typically have fixed values specific to a particular device or board. The value of these parameters is often retained throughout the life of the device/board. For this reason, it is generally acceptable to program these basic parameters into the OTP memory early in the development, before the design is finalized.

Table 4. Basic NVRAM Parameters and 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
Customer specific information	0x81	4

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 binary map (for example, “US” = “0x55 0x53”).

In the OTP binary map, each tuple is formed by the four fragments described in Table 5.

Table 5. CIS Tuple Format

Fragment	Description
80	Indicates the beginning of a new tuple. 0x80 is specific to Cypress tuple subtags.
Length	Defines the total size (in bytes) of the tag plus the value of the tuple that occupies the OTP memory space.
Tag	Identifies a parameter in the nvram.txt file. A tag usually takes one byte in memory.
Value	Specifies the value of the parameter in little-endian format (first byte is the least-significant byte (LSB)).

For example, the following tuple is defined by the fragments that follow:

80	03	02	00	11

- 80 Beginning of a new tuple.
- 03 The tag (1 byte) and the value (2 bytes) occupy 3 bytes (total) in the OTP memory.
- 02 Tag of 0x02 is the identifier for boardrev in the nvram.txt file.
- 00 11The value of boardrev in reverse hexadecimal bytes or 0x1100.

Figure 2 on page 14 provides an example OTP binary map for a device that contains some of the nvram.txt file parameters listed in Table 4 on page 12.

Note:

- CIS tuples do not have to be listed in any particular order because each tuple begins with a unique identifier.
- OTP memory bytes can be written to only once. Only blank and zero-programmed bytes can be programmed during subsequent write cycles.

Figure 2. SDIO OTP Memory Map

Offset	0x0	0x1	0x2	0x3	0x4	0x5	0x6	0x7	0x8	0x9	0xA	0xB	0xC	0xD	0xE	0xF
00000000	4B	00	FF	FF	00	00	20	04	D0	02	54	43	80	02	00	0B
00000010	80	03	02	02	11	80	03	1B	DC	06	80	07	19	66	55	44
00000020	33	22	11	00	00	00	00	00	00	00	00	00	00	00	00	00
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	00	00	00	00	00	00
00000140	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
00000150	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
00000160	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
00000170	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
00000180	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
00000190	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
000001A0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
000001B0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
000001C0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
000001D0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
000001E0	00	00	00	FF												

SDIO HW Header

OTP End

boardrev=0x1102

boardtype=0x06dc

macaddr=66:55:44:33:22:11

sromrev=11

Note:

1. This example is for a CYW4354. Change the red byte in the SDIO HW header to **56** or **58**, respectively, for the CYW4356 and BCM4358.
2. The maximum storage in bits = 4896 – 1024 = 3872 (or 484 bytes).

7.2 Creating and Editing the OTP Memory Binary Map

Use a hexadecimal text editor to create and edit an OTP binary map. A hexadecimal text editor preserves the formatting of the nvram.txt file. Writing to the OTP memory requires a bin file that fits in the OTP memory space. For the CYW4354, CYW4356, and BCM4358, the maximum size of the OTP memory is 484 bytes.

Note: Do not use Notepad to edit the nvram.txt file. Edit the nvram.txt file using a properly formatted text editor such as Notepad++ or WordPad++ to preserve the original format of the file. Using a non-formatted text editor such as Notepad could corrupt the format of the NVRAM map, causing the driver to incorrectly read the nvram.txt file.

1. Add or edit each byte in the OTP binary map to populate the CIS tuples, as described in the OTP binary map instructions provided in Programming the Basic Parameters into OTP Memory on page 12.

Note: The OTP binary map file (see Figure 3 on page 16) has been edited to match the example OTP binary map described in Figure 2 on page 14.

2. Save the OTP binary map as a binary image file (.bin extension) to the directory containing the wl.exe file.

Note: The file name must be saved with a .bin file extension so that the data it contains can be programmed into the OTP memory. For example purposes, this file is referred to later in this document as the OTP.bin file.

Figure 3 shows the hexadecimal OTP binary map template.

Figure 3. Hexadecimal OTP Memory Binary Map Template

Offset	0x0	0x1	0x2	0x3	0x4	0x5	0x6	0x7	0x8	0x9	0xA	0xB	0xC	0xD	0xE	0xF
00000000	4B	00	FF	FF	00	00	20	04	D0	02	**54**	43	80	02	00	0B
00000010	80	03	02	02	11	80	03	1B	DC	06	80	07	19	66	55	44
00000020	33	22	11	00	00	00	00	00	00	00	00	00	00	00	00	00
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	00	00	00	00	00	00
00000140	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
00000150	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
00000160	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
00000170	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
00000180	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
00000190	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
000001A0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
000001B0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
000001C0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
000001D0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
000001E0	00	00	00	FF												

Note:

1. This example is for a CYW4354. Change the red byte in the SDIO HW header to **56** or **58**, respectively, for the CYW4356 and BCM4358.

8 Programming OTP Memory on an Android or x86 Platform

This section outlines the procedure to program OTP memory on an Android or x86 platform.

Required Hardware:

- 1× CYW4354, CYW4356, or BCM4358 – this is the DUT.
- 1× Android or x86 platform to which the DUT is connected.

Required Software:

- **Cypress SDIO MFG driver package**: contains driver files for the CYW4354, CYW4356, and BCM4358. This driver package is typically provided by Cypress.
- **OTP.bin file**: contains the information that is going to be programmed into device OTP memory. The OTP.bin file is typically provided by Cypress.

OTP Memory Programming Steps:

1. For platforms running the Android OS, load the DHD driver with MFG firmware:

```
adb shell
ifconfig wlan0 down
echo "/system/vendor/firmware/fw_bcmdhd_mfg.bin" > /sys/module/bcmdhd/parameters/
firmware_path
ifconfig wlan0 up
```

Note: After the driver successfully loads, the OTP memory can be programmed.

Note: Follow the normal procedures for installing the WLAN driver on x86 platforms running Linux or Windows. Information on the normal procedures are outside the scope of this document.

2. Run the following command to check the contents of the OTP memory:

```
> wl cisdump
```

3. Verify that the OTP memory has never been programmed, by verifying that all bytes are 0x00:

```
Source: 2 (Internal OTP)
Maximum length: 484 bytes
Byte   0: 0x00 0x00 0x00 0x00 0x00 0x00
0x00 0x00
Byte   8: 0x00 0x00 0x00 0x00 0x00 0x00
0x00 0x00
Byte  16: 0x00 0x00 0x00 0x00 0x00 0x00
0x00 0x00
Byte  24: 0x00 0x00 0x00 0x00 0x00 0x00
0x00 0x00
Byte  32: 0x00 0x00 0x00 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
Byte 264: 0x00 0x00 0x00 0x00 0x00 0x00
0x00 0x00
Byte 272: 0x00 0x00 0x00 0x00 0x00 0x00
0x00 0x00
Byte 280: 0x00 0x00 0x00 0x00 0x00 0x00
0x00 0x00
Byte 288: 0x00 0x00 0x00 0x00 0x00 0x00
0x00 0x00
Byte 296: 0x00 0x00 0x00 0x00 0x00 0x00
0x00 0x00
Byte 304: 0x00 0x00 0x00 0x00 0x00 0x00
0x00 0x00
Byte 312: 0x00 0x00 0x00 0x00 0x00 0x00
0x00 0x00
Byte 320: 0x00 0x00 0x00 0x00 0x00 0x00
0x00 0x00
Byte 328: 0x00 0x00 0x00 0x00 0x00 0x00
0x00 0x00
Byte 336: 0x00 0x00 0x00 0x00 0x00 0x00
0x00 0x00
Byte 344: 0x00 0x00 0x00 0x00 0x00 0x00
0x00 0x00
Byte 352: 0x00 0x00 0x00 0x00 0x00 0x00
0x00 0x00
Byte 360: 0x00 0x00 0x00 0x00 0x00 0x00
0x00 0x00
Byte 368: 0x00 0x00 0x00 0x00 0x00 0x00
0x00 0x00
Byte 376: 0x00 0x00 0x00 0x00 0x00 0x00
0x00 0x00
Byte 384: 0x00 0x00 0x00 0x00 0x00 0x00
0x00 0x00
Byte 392: 0x00 0x00 0x00 0x00 0x00 0x00
0x00 0x00
Byte 400: 0x00 0x00 0x00 0x00 0x00 0x00
0x00 0x00
Byte 408: 0x00 0x00 0x00 0x00 0x00 0x00
0x00 0x00
Byte 416: 0x00 0x00 0x00 0x00 0x00 0x00
0x00 0x00
Byte 424: 0x00 0x00 0x00 0x00 0x00 0x00
0x00 0x00
Byte 432: 0x00 0x00 0x00 0x00 0x00 0x00
0x00 0x00
Byte 440: 0x00 0x00 0x00 0x00 0x00 0x00
0x00 0x00
Byte 448: 0x00 0x00 0x00 0x00 0x00 0x00
0x00 0x00
Byte 456: 0x00 0x00 0x00 0x00 0x00 0x00
0x00 0x00
Byte 464: 0x00 0x00 0x00 0x00 0x00 0x00
0x00 0x00
Byte 472: 0x00 0x00 0x00 0x00 0x00 0x00
0x00 0x00
Byte 480: 0x00 0x00 0x00 0x00
```

4. In the directory where the OTP.bin file is stored, run the following command to program the OTP memory:

   ```
   > wl ciswrite OTP.bin
   ```

5. Run the following command to check the contents of the OTP memory:

   ```
   > wl cisdump
   ```

6. Verify that the OTP memory was successfully programmed.

If programming is successful, then the OTP memory contents will look similar to what is shown below for a CYW4354. All programmed bytes are shown in red text. All unprogrammed bytes are shown in black text.

```
Source: 2 (Internal OTP)
Maximum length: 484 bytes
Byte   0: 0x4b 0x00 0xff 0xff 0x00 0x00
0x20 0x04
Byte   8: 0xd0 0x02 0x54 0x43 0x80 0x02
0x00 0x0b
Byte  16: 0x80 0x03 0x02 0x02 0x11 0x80
0x03 0x1b
Byte  24: 0xdc 0x06 0x80 0x07 0x19 0x66
0x55 0x44
Byte  32: 0x33 0x22 0x11 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
Byte 264: 0x00 0x00 0x00 0x00 0x00 0x00
0x00 0x00
Byte 272: 0x00 0x00 0x00 0x00 0x00 0x00
0x00 0x00
Byte 280: 0x00 0x00 0x00 0x00 0x00 0x00
0x00 0x00
Byte 288: 0x00 0x00 0x00 0x00 0x00 0x00
0x00 0x00
Byte 296: 0x00 0x00 0x00 0x00 0x00 0x00
0x00 0x00
Byte 304: 0x00 0x00 0x00 0x00 0x00 0x00
0x00 0x00
Byte 312: 0x00 0x00 0x00 0x00 0x00 0x00
0x00 0x00
Byte 320: 0x00 0x00 0x00 0x00 0x00 0x00
0x00 0x00
Byte 328: 0x00 0x00 0x00 0x00 0x00 0x00
0x00 0x00
Byte 336: 0x00 0x00 0x00 0x00 0x00 0x00
0x00 0x00
Byte 344: 0x00 0x00 0x00 0x00 0x00 0x00
0x00 0x00
Byte 352: 0x00 0x00 0x00 0x00 0x00 0x00
0x00 0x00
Byte 360: 0x00 0x00 0x00 0x00 0x00 0x00
0x00 0x00
Byte 368: 0x00 0x00 0x00 0x00 0x00 0x00
0x00 0x00
Byte 376: 0x00 0x00 0x00 0x00 0x00 0x00
0x00 0x00
Byte 384: 0x00 0x00 0x00 0x00 0x00 0x00
0x00 0x00
Byte 392: 0x00 0x00 0x00 0x00 0x00 0x00
0x00 0x00
Byte 400: 0x00 0x00 0x00 0x00 0x00 0x00
0x00 0x00
Byte 408: 0x00 0x00 0x00 0x00 0x00 0x00
0x00 0x00
Byte 416: 0x00 0x00 0x00 0x00 0x00 0x00
0x00 0x00
Byte 424: 0x00 0x00 0x00 0x00 0x00 0x00
0x00 0x00
Byte 432: 0x00 0x00 0x00 0x00 0x00 0x00
0x00 0x00
Byte 440: 0x00 0x00 0x00 0x00 0x00 0x00
```

```
0x00 0x00                                          Byte 464: 0x00 0x00 0x00 0x00 0x00 0x00
Byte 448: 0x00 0x00 0x00 0x00 0x00 0x00            0x00 0x00
0x00 0x00                                          Byte 472: 0x00 0x00 0x00 0x00 0x00 0x00
Byte 456: 0x00 0x00 0x00 0x00 0x00 0x00            0x00 0x00
0x00 0x00                                          Byte 480: 0x00 0x00 0x00 0xff
```

If the CIS dump matches your OTP.bin file, then OTP memory programming is successful, and the OTP.bin is correctly programmed.

9 References

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

Document (or Item) Name	Number	Source
[1] *OTP Programming and NVRAM Development in PCIe Mode,* Application Note	4356-AN10x-R	Cypress Developer Community

Document History Page

Document Title: AN214808 - OTP Programming and NVRAM Development in SDIO Mode - CYW4354/CYW4356/BCM4358 Document Number: 002-14808				
Rev.	**ECN No.**	**Orig. of Change**	**Submission Date**	**Description of Change**
**	–	–	08/19/2014	4354_4356_4358-AN100-R Initial release
*A	5459962	UTSV	11/14/2016	Updated in Cypress template Added Cypress part numbering scheme

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
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 | WICED IoT Forum | Projects | Video | Blogs | Training | Components

Technical Support

cypress.com/support

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

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