AN225021

CYW43012 OTP Programming and NVRAM Development

Associated Part Family: CYW43012

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 CYW43012 devices.

Contents

1 Introduction

The Cypress CYW43012 is a 28-nm, ultra-low power single-chip device that supports single-stream, dual-band IEEE 802.11n-compliant Wi-Fi MAC/baseband/radio and Bluetooth 5.0. When used with IEEE 802.11ac access points, the CYW43012 provides superior performance in terms of throughput and power consumption than other 802.11n products by leveraging 802.11ac-friendly™ features.

A one-time programmable (OTP) nonvolatile memory is available for storing board-specific information such as product ID, manufacturer ID, MAC address, and more. The size of HW+SW usable region in OTP is 216 Bytes is available on the CYW43012.

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 CYW43012.

1.1 Purpose and Audience

This document is intended for design and application engineers. It contains information on the following:

- NVRAM content development and OTP programming flow
- Customizing the *nvram.txt* file
- OTP programming procedure

1.2 Before You Begin

Cypress recommends that you request the following items from cypress.com/support:

- A CYW43012 board reference design package that contains the following:
  - The reference board schematic, bill of materials, and layout.
  - The *nvram.txt* template file for the reference board.
- The correct WICED-SDK package.

See IoT Resources for details on accessing the Cypress' SFDC. If necessary, contact your Sales or Engineering support representative.

1.3 Acronyms and Abbreviations

In most cases, acronyms and abbreviations are defined upon 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-iott 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 (https://community.cypress.com/)

3 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 Linux or Windows, 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, the *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.

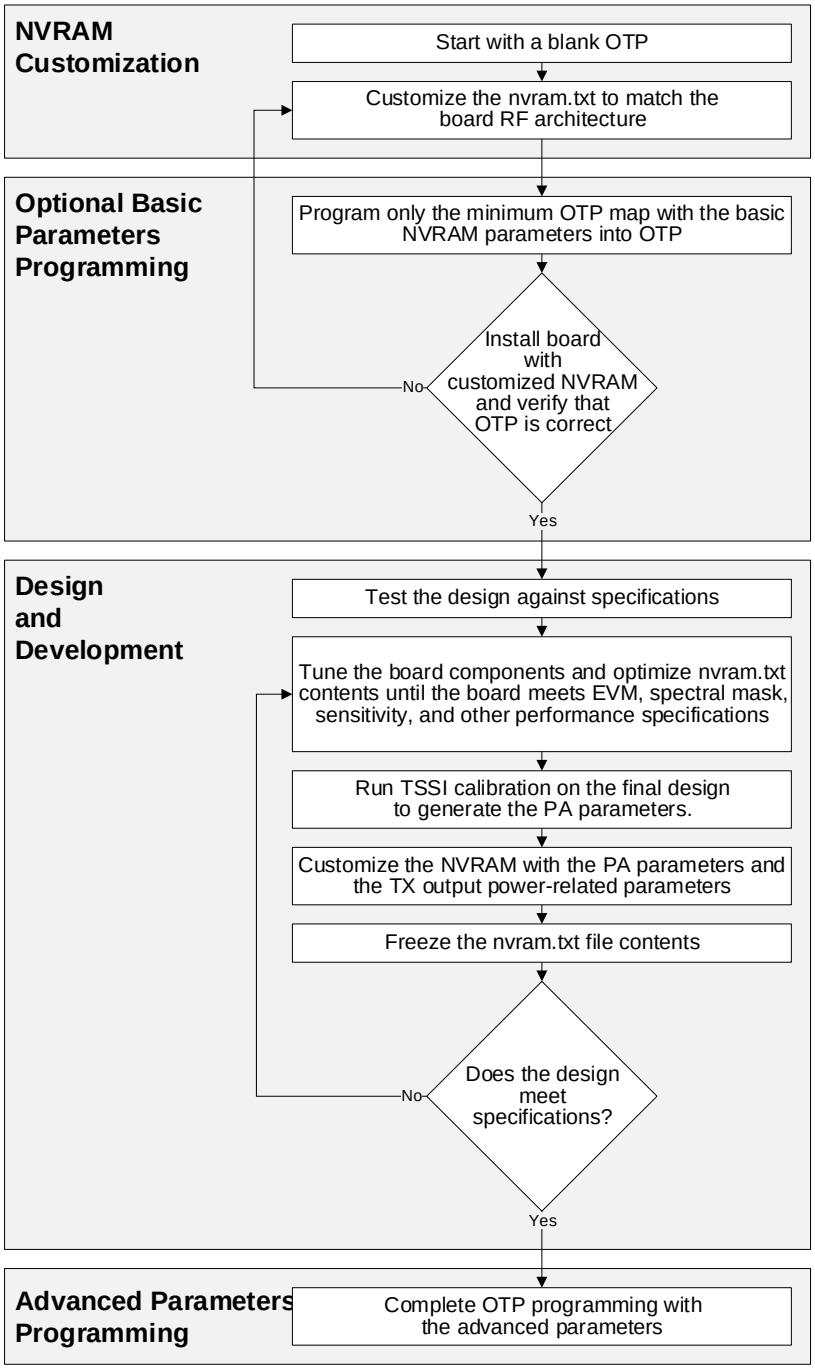
Because the OTP programming process is irreversible, Cypress recommends that board designers finalize all parameters before programming the OTP memory. Boards and modules should be tested using only the editable *nvram.txt* file. The *nvram.txt* file parameters are loaded by the driver into on-chip RAM, 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 before 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.

CAUTION: 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.

4 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* file 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.

Figure 1. NVRAM Development and OTP Programming Flow



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.

5 Customizing the nvram.txt File

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

5.1 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, *cyw943012fcref.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 with Cypress support. Use this file as a sample or template to begin the customization to match your own board design.

A sample *nvram.txt* file, with parameters that are common to Cypress CYW43012 reference design boards, is shown in Table 1. No specific order is required for the parameters in the *nvram.txt* file.

Parameters listed in Table 1 are design variables that must be reviewed before starting board or module testing. Specifically, boardflags, 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. PA parameters are eventually optimized using Cypress' transmit signal strength indicator (TSSI) calibration tools.

Note: The parameters in Table 1 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 1. NVRAM Parameters That Require Customizing for Each Board Design

NVRAM Parameter	Example Data	Description	Comments
sromrev	0x0b	SROM revision for IEEE 802.11ac chips.	Do not modify
vendid	0x14e4	Vendor ID – identifies the IC vendor: always 0x14e4, which is the Cypress PCI vendor ID	Do not modify
devid	0x43d0	Device ID – identifies the device. Does not necessarily correspond to the chip number.	Do not modify
macaddr	00:90:4c:2b:3${maclo12}	Sets the device MAC address.	
nocrc	1	Check for CRC errors when loading firmware.	Do not modify
boardtype	0x084d	This is a critical parameter that should be copied from a similar Cypress reference design.	Do not modify
boardrev	0x1101	Board revision tracked by the internal test tool (optional). 0x1101 converts to P101 0x1208 converts to P208	
xtalfreq	37400	Onboard XTAL or oscillator frequency, in kHz.	Do not modify
boardflags	0xa00	Board configuration flags	Do not modify
boardflags2	0x40002000		
rxgains2gelnagaina0	0	This variable defines the 2.4-GHz eLNA gain in dB.	
rxgains2gtrisoa0	0	This variable defines the 2.4-GHz isolation that the TR switch provides when in 'T' mode.	
rxgains2gtrelnabypa0	0	This variable defines the isolation that a 2.4-GHz eLNA provides when put in bypass mode.	
rxgains5gelnagaina0	0	This variable defines the 5-GHz eLNA gain in dB.	

NVRAM Parameter	Example Data	Description	Comments
rxgains5gtrisoa0	0	This variable defines the 5 GHz isolation that the TR switch provides when in 'T' mode. [dB]	This needs to be measured and entered, and is required for reporting accurate RSSI.
pdgain5g	0	5-GHz and 2.4-GHz power detector parameter respectively, used by the driver to program the TSSI loop back path.	Do not modify
pdgain2g	0		
rxchain	1	This specifies the number of RX paths (bit mask).	Do not modify
txchain	1	This specifies the number of TX paths (bit mask).	
aa2g	1	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	
aa5g	1		
swdiv_en	1	For antenna diversity, to enable the SW diversity feature.	
swdiv_gpio	0	This variable is a bit offset for diversity-related PHY register setting if there is a need to trigger a different antenna.	
swdiv_swctrl_en	2	This variable is a diversity mode selection entry.	
swdiv_swctrl_ant0	0	These are entries used as shared memory setting for ucode activity. This device uses mode 1 only.	
swdiv_swctrl_ant1	1		
swdiv_antmap2g_main	1	Select antenna.	
swdiv_antmap5g_main	1	Select antenna.	
tssipos5g	0	Indicates whether TSSI has a positive slope for 5 GHz. Set this to '1'.	
tssipos2g	1	Indicates whether TSSI has a positive slope for 2.4 GHz. Set this to '1'.	
femctrl	0	Defines front-end RF switch or front-end module (FEM) control logic for both bands.	Do not modify
pa2ga0	-179, 5709, -665	PA parameters for the 2.4-GHz band based on TSSI calibration. pa2ga0 - OFDM, for high-range power	
pa5ga0	45, 6701, -710, 55, 6738, -697, 28, 6666, -694, -130, 5655, -758	PA parameters for the 5-GHz band based on TSSI calibration. (Low/Mid/High/X1 subband frequency range, for 20, 40 BWs) pa5ga0 - OFDM, For high-range power 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	
pdoffset40ma0	0x0		

NVRAM Parameter	Example Data	Description	Comments
pdoffset80ma0	0x0	5 GHz and 20-MHz bandwidths respectively, power detector (PD) offset (1/4 dB steps) in 2's complement format, 4 bits for each subband (Format: X1/High/Mid/Low subband frequency range).	
extpagain5g	2	Supports 5-GHz external PA. Use two for iPA boards, and use one for ePA boards.	
extpagain2g	2	Supports 2.4-GHz external PA. Use two for iPA boards, and use one for ePA boards.	
maxp2ga0	0x4a	Maximum output power for the 2.4-GHz band in hexadecimal format in 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 0x4a example, the maximum output power is: 0.25 x 74 = 18.5 dBm, and the nominal power is: 18.5–1.5 = 17 dBm.	
maxp5ga0	0x46, 0x46, 0x42, 0x42	Maximum output power for the 5-GHz band in hexadecimal format in 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; in Low, Mid, High, X1 subbands sequence. For the 0x46 example, the maximum output power is: 0.25 x 70 = 17.5 dBm, and the nominal power is: 17.5 – 1.5 = 16 dBm.	
mcsbw202gpo	0xb8544433	11n/ac MCS0/1/2, 3-7, C8, C9 2.4-MHz power offset for 20-MHz bandwidth. 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).	
mcsbw205glpo	0xa8754100	5-GHz low band 11n/ac MCS0/ 1/2, 3-7, C8, C9 power offset for 20 MHz BW. Specified in half dBm units – C9/C8/M7/M6/M5/M4/M3/M0-2. For the 0xa8754100 example, power backoff for rate C9 is 10 (in decimal)/2 = 5 dB. Thus, with nominal power of 16 dBm, the power backoff (in dB) and the target power (in dBm) respectively for the 0xa8754100 example are: C9 : 5 and 11 C8 : 4 and 12 M7 : 3.5 and 12.5 M6 : 2.5 and 13.5 M5 : 2 and 14 M4 : 0.5 and 15.5 M3 : 0 and 16 M0-2 : 0 and 16	
mcsbw205gmpo	0xb9865200	5-GHz mid band 11n/ac MCS0/ 1/2, 3-7, C8, C9 power offset for 20 MHz BW. Specified in half dBm units – C9/C8/M7/M6/M5/M4/M3/M0-2.	
mcsbw205ghpo	0xba865200	5-GHz high/X1 band 11n/ac MCS0/ 1/2, 3-7, C8, C9 power offset for 20 MHz BW. Specified in half dBm units – C9/C8/M7/M6/M5/M4/M3/M0-2.	

NVRAM Parameter	Example Data	Description	Comments
dot11agofdmhrbw202gpo	0x4443	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	
swctrlmap_2g	0x00002111, 0x00002212, 0x00002212, 0x000000, 0x0ff	Describes how to control the external 2.4-GHz and 5-GHz FEM or TRSW (TR switch).	Do not modify
swctrlmap_5g	0x00002414, 0x00002818, 0x00002818, 0x000000, 0x0ff		
swctrlmapext_5g	0x00000000, 0x00000000, 0x00000000, 0x000000, 0x000		
swctrlmapext_2g	0x00000000, 0x00000000, 0x00000000, 0x000000, 0x000		

5.2 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.

5.3 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 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 provide 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, you can select the needed parameters to program the OTP for production.

CYW43012 chips have up to 356 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. You 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, and so on).

6 Preparing for Programming

6.1 OTP

Before OTP programming, an OTP binary map file must be prepared and edited with the correct values. 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 CIS Map for details.

At the start of the OTP map, a string called the SDIO hardware header must be present before any NVRAM variables. When a driver detects the content in the OTP, the SDIO hardware header is required to boot up CYW43012 devices via the SDIO interface. Therefore, the SDIO hardware header is the minimum set of parameters when programming an OTP. Any other parameters needed to be programmed to the OTP are appended after the SDIO hardware header (see Creating and Editing the OTP Binary Map).

6.1.1 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 CYW43012 devices, the user partition of the OTP has a maximum size of 356 bytes. Figure 2 shows an example of this partition for CYW43012.

Figure 2. Basic CIS Map for CYW43012

43012 WLAN OTP Map

Offset	0x0	0x1	0x2	0x3	0x4	0x5	0x6	0x7	0x8	0x9	0xa	0xb	0xc	0xd	0xe	0xf
00000000	5b	00	ff	ff	C7	00	20	04	d0	02	**04**	a8	80	02	00	0b
00000010	80	03	02	01	13	80	03	1b	43	08	80	07	19	00	90	4c
00000020	2a	99	88	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														

Hardware Header
sromrev
macaddress
OTP end

43012 BT OTP Map

Offset	0x0	0x1	0x2	0x3	0x4	0x5	0x6	0x7	0x8	0x9	0xa	0xb	0xc	0xd	0xe	0xf
00000170									50	4F	10	06	FF	EE	DD	CC
00000180	BB	AA	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	00	00	00	00	00	00	00	00	00	00	00	00	00
000001f0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
00000200	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
00000210	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
00000220	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
00000230	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
00000240	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
00000250	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
00000260	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
00000270	00	00	00	00	00	00	00	00								

BT OTP start

Offset = 3008 bits

BT OTP Header

BD ADDR

BT OTP end

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 2 has been edited to match the CYW43012 OTP binary map example.

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 remote WL utility, which is discussed in the next section.

For example purposes, this file is referred to as *cyw43012cis_map_V2.bin* in the following instructions.

7 OTP Programming Procedure

Do the following to program the above OTP binary map and some other important parameters to the CYW43012 devices:

1. Program the OTP binary map:

   ```
   wl ciswrite cyw43012cis_map_V2.bin
   ```

   Note: The **cis_map.bin* binary file is different for each device.

2. Confirm that the OTP binary map is programmed successfully by running the following command:

   ```
   > wl cisdump
   ```

3. Optionally, use the following command to output OTP contents to the console:

   ```
   > wl otpdump
   ```

   The `wl otpdump` command outputs the entire set of OTP contents with absolute addresses. The user partition is a subset that spans address spaces from 0x40 to 0x01A3, both inclusive.

4. Power cycle the CYW43012 WICED board.

Appendix A. CIS Map

Table 2 and Table 3 list the CIS map (standard tuple tags and Cypress subtags) for SDIO devices.

Table 2. 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 the SDIO specification.
CISTPL_BRCM_HNBU	0x80				Cypress-specific tuple subtag identifier.
CISTPL_END	0xff				End of the CIS tuple chain.

Table 3. 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.

Name	Tag	Length	Format	Variables	Description
				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. 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.
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.

Name	Tag	Length	Format	Variables	Description
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	
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	

Name	Tag	Length	Format	Variables	Description
				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.

Document History

Document Title: AN225021 – CYW43012 OTP Programming and NVRAM Development

Document Number: 002-25021

Revision	ECN	Orig. of Change	Submission Date	Description of Change
**	6332215	SWPA	11/20/2018	New Application Note

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.

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 MCU

Cypress Developer Community

Community | Projects | Videos | Blogs | Training | Components

Technical Support

cypress.com/support

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