AN225666

OTP Memory Programming and NVRAM Development - CYW8X359/CYW8X342

Author: Chuck Wang

Associated Part Family: CYW8X359/CYW8X342

AN225666 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 the CYW8X359/CYW8X342 device using the PCIe or SDIO host interface for WLAN.

Contents

1 Introduction

The Cypress CYW8X359/CYW8X342 is a single-chip IEEE 802.11 ac 2x2 MIMO / RSDB WLAN + BT 4.2 device for embedded applications. One-time programmable (OTP) nonvolatile memory is included in the WLAN section of the device to store board-specific information such as PCIe header, product ID, manufacturer ID, and MAC address. Excluding the internal header information, up to 1150 bytes of user accessible OTP memory is available on CYW8X359/CYW8X342 for WLAN information. The application note provides OTP programming information for both PCIe and SDIO host interfaces.

The OTP memory content, along with an editable NVRAM file (*nvram.txt* file), provides all configuration information used by the WLAN device driver to initialize and configure CYW8X359/CYW8X342.

1.1 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/).

2 OTP Memory Programming Considerations

In embedded designs where the host and device are permanently connected, which is typically done using a hardwired PCIe or SDIO interface. The only entry which is mandatory to be programmed into OTP memory is the PCIe or SDIO header. This is because there are certain PCIe or SDIO function settings (such as L1 sub-state for low power) which are read before the firmware and NVRAM are downloaded. To properly set these settings, the PCIe or SDIO header must be programmed into their OTP memory.

Other than the PCIe or SDIO header, all other NVRAM parameters can be stored in the host's nonvolatile memory rather than in OTP memory. 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 bits in an unprogrammed device is 0. Individual bits can be set to 1, but once set, the bits can never be reset to 0. The entire OTP array can be programmed in a single-write cycle using the `wl` commands provided with the PCIe or 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.

The OTP programming process is irreversible, so it is recommended that you finalize all NVRAM parameters before programming any of the parameter into the OTP memory. Test the boards and modules using only the editable *nvram.txt* file.

The driver loads the parameters stored in the *nvram.txt* file onto on-chip RAM, allowing the chip to be tested even if the OTP memory has only been programmed with the PCIe or SDIO header. This method allows you to 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.

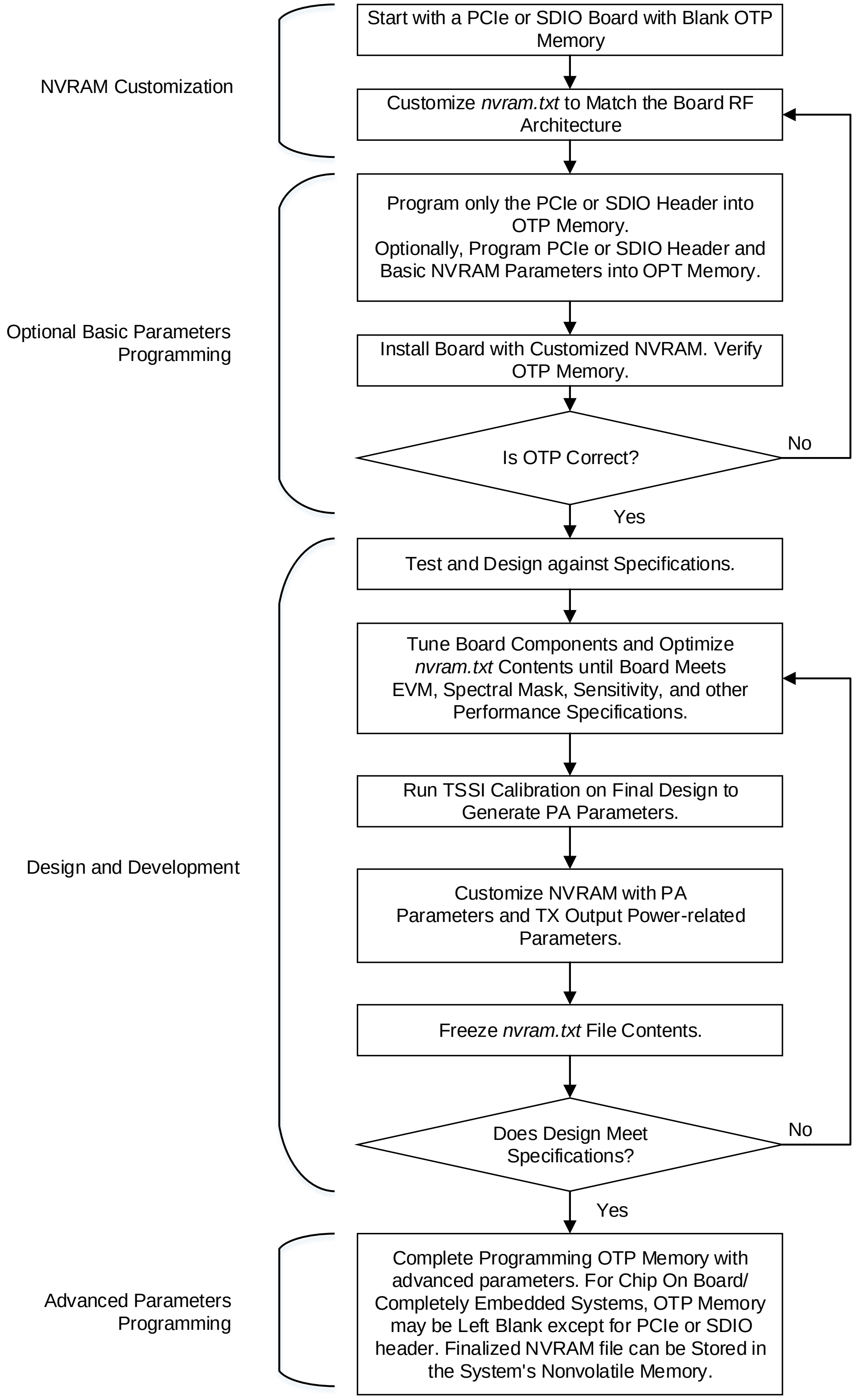
Note: The programming process of an OTP memory is irreversible. Cypress strongly recommends conducting development on boards using the parameters provided in the editable *nvram.txt* file. Do not program the OTP memory until the contents of the *nvram.txt* file have been verified and the file has been finalized for production use. The one exception to this is the PCIe or SDIO header, which must be programmed into OTP memory for full PCIe or SDIO functionality.

3 NVRAM Content Development and Memory Programming Flow

Figure 1 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: Conduct the NVRAM development and OTP programming flow shown in Figure 1 on fewer 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 is approved for production use, programming can begin for high volume mass production as defined by each manufacturer.

Figure 1. NVRAM Development and Programming Flow of OTP Memory


NVRAM Customization
Start with a PCIe or SDIO Board with Blank OTP Memory
Customize *nvram.txt* to Match the Board RF Architecture
Optional Basic Parameters Programming
Program only the PCIe or SDIO Header into OTP Memory.
Optionally, Program PCIe or SDIO Header and Basic NVRAM Parameters into OPT Memory.
Install Board with Customized NVRAM. Verify OTP Memory.
Is OTP Correct?
No
Yes
Design and Development
Test and Design against Specifications.
Tune Board Components and Optimize *nvram.txt* Contents until Board Meets EVM, Spectral Mask, Sensitivity, and other Performance Specifications.
Run TSSI Calibration on Final Design to Generate PA Parameters.
Customize NVRAM with PA Parameters and TX Output Power-related Parameters.
Freeze *nvram.txt* File Contents.
Does Design Meet Specifications?
No
Yes
Advanced Parameters Programming
Complete Programming OTP Memory with advanced parameters. For Chip On Board/ Completely Embedded Systems, OTP Memory may be Left Blank except for PCIe or SDIO header. Finalized NVRAM file can be Stored in the System's Nonvolatile Memory.

4 Customizing nvram.txt File

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

4.1 Using nvram.txt File Template

For each 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, cyw8x359wliparef.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.

Table 1 provides a list of parameters in a typical *nvram.txt* file that are common to dual-band 802.11ac 2x2 MIMO PCIe or 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 1 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 1 may need to be changed by Cypress to accommodate differences in the RF front end between the customer's design and the Cypress reference design from which it was derived.

Table 1. Cypress-specific NVRAM Parameters

NVRAM Parameter	Example Data	Description
sromrev	11	SROM revision for 802.11ac chips
boardtype	0x0776	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. For CYW8X359/CYW8X342, set the value to 1.
tssipos5g	1	This represents if TSSI has positive slope for 5 GHz. For CYW8X359/CYW8X342, set the value to 1.
rxchain	3	This specifies the number of rx paths (bit mask). For CYW8X359/CYW8X342, set the value to 3.
txchain	3	This specifies the number of tx paths (bit mask). For CYW8X359/CYW8X342, set the value to 3.
antswitch	0	This enables switch-based diversity: ▪ 0: disable ▪ 1: enable
venid	0x14e4	Vendor ID
devid	0x43ef	Chip ID, CYW8X359/CYW8X342
manfid	0x2d0	Manufacturer ID
nocrc	1	Check for CRC errors when loading firmware
boardflags boardflags2 boardflags3	0x12401001 0x00802000 0x4800018a	Board configuration flag that defines the power topology, external components (iPA, eLNA), and so on
tworangetssi2g tworangetssi5g	0 0	2.4 GHz and 5 GHz TSSI dual power range flag, which iPA chips support
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.
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 CYW8X359/CYW8X342)
subband5gver	0x4	Defines 5 GHz sub-band allocation

NVRAM Parameter	Example Data	Description
tempthresh	255	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	15	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	This parameter is for Cypress internal use only. **Note**: Do not modify.
swctrlmap_2g, swctrlmap_5g, swctrlmapext_2g/5g	0x02020202, 0x05050404, 0x04040000, 0x000000	Describes how to control the external 2.4 GHz and 5 GHz FEM (front-end module) or TR-SW.

The design variables listed in Table 2 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 transmit signal strength indicator (TSSI) calibration tools.

The parameters in Table 2 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 version of the nvram.txt file for the reference design before starting for any board customization efforts.

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

Table 2. 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 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.
rxgains2gtrelnabypa0, rxgains2gtrelnabypa1	1	This variable defines the isolation that 5 GHz eLNA provides when put in bypass mode. ‘a0’ and ‘a1’ apply for Core 0 and Core 1, respectively, and to low sub-band.
rxgains5gmtrelnabypa0, rxgains5gmtrelnabypa1	1	This variable defines the isolation that 5 GHz eLNA provides when put in bypass mode. ‘a0’ and ‘a1’ apply for Core 0 and Core 1, respectively, and to mid sub-band.
rxgains5ghtrelnabypa0, rxgains5ghtrelnabypa1	1	This variable defines the isolation that 5 GHz eLNA provides when put in bypass mode. ‘a0’ and ‘a1’ apply for Core 0 and Core 1, respectively, and to high/X1 sub-band.
rxgains2gelnagaina0, rxgains2gelnagaina1	1	This variable defines the 2.4 GHz eLNA gain in dB. ‘a0’ and ‘a1’ apply to Core 0 and Core 1, respectively.

NVRAM Parameter	Example Data	Description
rxgains2gtrisoa0, rxgains2gtrisoa1	7	This variable defines the 2.4 GHz isolation that 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 sub-band.
rxgains5gtrisoa0, rxgains5gtrisoa1	6	This variable defines the 5 GHz isolation that TR switch provides when in "T" mode. ‘a0’ and ‘a1’ apply to Core 0 and Core 1, respectively. Applies to low sub-band
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 sub-band.
rxgains5gmtrisoa0, rxgains5gmtrisoa1	6	This variable defines the 5 GHz isolation that TR switch provides when in "T" mode. ‘a0’ and ‘a1’ apply to Core 0 and Core 1, respectively. Applies to mid sub-band.
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 sub-band.
rxgains5ghtrisoa0, rxgains5ghtrisoa1	6	This variable defines the 5 GHz isolation that TR switch provides when in "T" mode. ‘a0’ and ‘a1’ apply for Core 0 and Core 1, respectively. Applies to high/X1 sub-band.
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 2s 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 / 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). Sub-band 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.

NVRAM Parameter	Example Data	Description
pa5gbw4080a0	-152, 8169, -994, -150, 8190, -999, -138, 8514, -1034, -130, 8806, -1058	Core 0 PA parameters for 5G (Low / Mid / High / X1). (subband5gver=4), for 40 MHz/80 MHz BW Sub-band 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.
pa5gbw4080a1	-169, 7695, -945, -156, 8053, -980, -160, 8075, -984, -158, 8219, -997	Core 1 PA parameters for 5G (Low / Mid / High / X1). (subband5gver=4), for 40 MHz/80 MHz BW Sub-band frequency range. Channel Range: ▪ Low 5180 to 5240 36-48 ▪ Mid 5260 to 5320 52-64 ▪ High 5500 to 5700 100-1-40 X1 5745 to 5825 149-165.
pdoffset40ma0, pdoffset40ma1	0x0000	5 GHz, 40 MHz BW PD offset (1/4 dB steps) in 2's complement format 4 bits for each sub-band. '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 sub-band. '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 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. In the example shown for 0 x 46, 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, 1 Mbps)
cckbw20ul2gpo	0x0000	CCK power offsets for 20 U/L rates (11, 5.5, 2, 1 Mbps)
cckpwroffset0	0x4	Core 0 2g CCK PD offset (1/4 dB steps) in 2's complement format - For example, if 1dB reduction is required then the value is 0x4, but if 1dB higher offset is required then it is 0xc.
cckpwroffset1	0x4	Core 1 2g CCK PD offset (1/4 dB steps) in 2's complement format – For example, if 1dB reduction is required then the value is 0x4, but if 1dB higher offset is required then it is 0xc.
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 orthogonal frequency division multiplexing (OFDM) rates as measured at 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.

NVRAM Parameter	Example Data	Description
mcslr5glpo	0x0000	5 GHz band low sub-band 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 sub-band 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.
mcslr5ghpo	0x0000	5 GHz band high/X1 sub-band 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
sb20in40hrpo	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 sub-band ▪ (2) 5 GHz mid sub-band ▪ (3) 5 GHz high/X1 sub-band
sb20in80and160hr5glpo	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 sub-band ▪ (2) 5 GHz mid sub-band ▪ (3) 5 GHz high/X1 sub-band
sb20in80and160hr5glpo	0	5 GHz low sub-band 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 sub-bands

NVRAM Parameter	Example Data	Description
sb40and80hr5glpo	0	5 GHz low sub-band 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 sub-band 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 sub-bands
sb40and80hr5gmpo	0	5 GHz mid sub-band 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 sub-band 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 sub-bands
sb40and80hr5ghpo	0	5 GHz high/X1 sub-band 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
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 sub-band ▪ (2) 5 GHz mid sub-band ▪ (3) 5 GHz high/X1 sub-band
sb20in80and160lr5glpo	0	5 GHz low sub-band 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 sub-bands
sb40and80lr5glpo	0	5 GHz mid sub-band 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 sub-bands
sb40and80hr5gmpo	0	5 GHz mid sub-band 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

NVRAM Parameter	Example Data	Description
sb20in80and160hr5ghpo	0	5 GHz high/X1 sub-band 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 sub-bands
sb40and80hr5ghpo	0	5 GHz high/X1 sub-band 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
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 sub-band ▪ (2) 5 GHz mid sub-band ▪ (3) 5 GHz high/X1 sub-band
sb20in80and160lr5glpo	0	5 GHz low sub-band 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 sub-bands
sb40and80lr5glpo	0	5 GHz low sub-band 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 sub-band 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 sub-bands
sb40and80lr5gmpo	0	5 GHz mid sub-band 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 sub-band 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 sub-bands
sb40and80lr5ghpo	0	5 GHz high/X1 sub-band 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

NVRAM Parameter	Example Data	Description
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 sub-band ▪ (2) 5 GHz mid sub-band ▪ (3) 5 GHz high/X1 sub-band
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 sub-band ▪ (2) 5 GHz mid sub-band ▪ (3) 5 GHz high/X1 sub-band
mux_enab	0x11	Specifies GPIO pin for OOB interrupts.
btc_mode	1	Specifies BT-COEX mode. Needed only for sLNA configuration.
ltecxmux	0x534201	Specifies LTE Coex settings.
cckdigfilttype	4	Specifies filter type for 11b mode.
fdss_level_2g	4, 4	Specifies FDSS setting for 2 GHz, shaping OFDM spectrum appropriately.
fdss_level_5g	4, 4	Specifies FDSS setting for 5 GHz, shaping OFDM spectrum appropriately.
fdss_interp_en=1	1	Enables interpolator in FDSS mode for avoiding EVM degradation.
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].

4.2 Editing 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.

4.3 Finalizing 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 PER) meets design specifications.

Cypress recommends running a regulatory pre-scan 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 programmed for production.

Note: The CYW8X359/CYW8X342 has 1150 bytes of space in the 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. Make sure that you 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, various NVRAM variables are stored in the system's nonvolatile memory as opposed to OTP memory.

5 Programming OTP Memory

One item that is required in the OTP memory is the PCIe or SDIO header. When using the PCIe or SDIO interface with the CYW8X359/CYW8X342, there are certain PCIe or SDIO function settings (such as L1 sub-state for low power) which are read before the firmware and NVRAM are downloaded. To properly set these settings, the PCIe or SDIO header must be programmed into their OTP (one-time programmable, nonvolatile memory).

Note that the PCIe or SDIO header should be created as a collaboration between Cypress and the customer. A majority of the PCIe or SDIO header fields are either generic (and do not need to be changed) or are Cypress-specific. There are a few fields that are customer-specific. Coordinate with the Cypress Hardware Applications team supporting the design to confirm the appropriate PCIe or SDIO header. Note that the PCIe or SDIO header is a set block of data with a predetermined order. It does not use tuples.

5.1 Programming 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 PCIe or SDIO 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.

Note: The PCIe or SDIO header does not use tuples, but is a set block of data with a specific ordering.

Table 3 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 3. Basic NVRAM Parameters and CIS Tuple Tags

NVRAM Parameter	CIS Tuple Tag	Length of Value (in Bytes)
sromrev	0x00	1
boardrev	0x02	2
broadtype	0x1b	2
macaddr	0x19	6
ccode[1]	0x0a	2
pa2ccka0	0x86	6
pa5gbw4080a0, pa5gbw4080a1	0x89	48
subband5gver	0x8A	2
pa2ccka1	0xA1	6
subband5gver, maxp2ga0, pa2ga0, maxp5ga0, pa5ga0	0x59	38
maxp2ga1, pa2ga1, maxp5ga1, pa5ga1	0x5A	36

[1] 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 4.

Table 4. 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 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 broadrev in the *nvram.txt* file.
- 00 11 – The value of boardrev in reverse hexadecimal byte or 0x1100.

Table 5 and Table 6 provide an example OTP binary map for a CYW8X359/CYW8X342 that contains the PCIe or SDIO header and some of the *nvram.txt* file parameters listed in Table 3.

Note: CIS tuples do not have to be listed in any order because each tuple begins with a unique identifier.

Note: OTP bytes can be written to only once. Only blank and zero-programmed bytes can be programmed during subsequent write cycles.

Note: The PCIe or SDIO header is a set block of data with a predetermined order. In PCIe or SDIO header order, do not use tuples. The tuples must be programmed into OTP memory for all PCIe or SDIO functions (such as L1SS) to operate properly.

Table 5. CYW8X359/CYW8X342 OTP Map for PCIe (Required in OTP)

Offset	0x0	0x1	0x2	0x3	0x4	0x5	0x6	0x7	0x8	0x9	0xa	0xb	0xc	0xd	0xe	0xf
00000000	0f	38	00	38 (1)	51	7 (2)	e4	14 (3)	1c	02	7e	1b	00	8a	00	00
00000010	00	00	00	00	00	00	00	00	54	00	3c	21	64	21	03	32
00000020	5f	18	05	96	28	9f	b6	79	80	80	03	0c	00	40	40	32
00000030	00	5f	f4	75	90	80	00	ee	00	84	08	F0	0b	00 (4)	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	ef	43 (5)	00	80	02	00	00	00	f5	3f	00	18	00	00	00	00
00000070	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
00000080	80	02	00	0b	80	03	02	01	11	80	03	1b	51	07	80	07
00000090	19	66	55	44	33	22	11	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

PCIe Header
(Required in OTP)

Other NVRAM Variables
(Optional in OTP)

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

boardtype = 0x074c

boardrev = 0x1101

sromrev = 11

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	00	00	00	00	00	00	00	00
00000280	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
00000290	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
000002a0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
000002b0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
000002c0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
000002d0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
000002e0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
000002f0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
00000300	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
00000310	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
00000320	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
00000330	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
00000340	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
00000350	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
00000360	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
00000370	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
00000380	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
00000390	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
000003a0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
000003b0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
000003c0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
000003d0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00

000003e0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
000003f0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
00000400	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
00000410	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
00000420	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
00000430	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
00000440	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
00000450	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
00000460	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
00000470	00	00	00	00	00	00	00	00	00	00	00	00	FF	FF		

OTP End

CYW8X359 OTP Header

00	38 (1)	XTAL frequency. 0x3800 for 37.4 MHz
51	7(2)	PCI Subsystem ID, vendor specific. Use 0x0000 if unknown
e4	14 (3)	PCI Subsystem vendor ID. 0x14e4 is Cypress' vendor ID
0b	00 (4)	Time for power on. Use 0x000b for 50 μS for default, unless specified
ef	43 (5)	Device ID. 0x43ef is device ID for CYW8X359/CYW8X342

Max WLAN SW/HW Region size = 9200 bits = 1150 bytes

Table 6. CYW8X359 OTP Map for SDIO

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	59	43	80	07	19	66
00000010	55	44	33	22	11	80	03	02	01	11	80	02	00	0B	80	03
00000020	1B	51	07	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	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	00	00	00	00	00	00	00	00
00000280	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
00000290	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
000002a0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
000002b0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
000002c0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
000002d0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
000002e0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
000002f0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
00000300	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
00000310	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
00000320	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
00000330	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
00000340	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
00000350	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
00000360	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
00000370	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
00000380	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
00000390	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
000003a0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
000003b0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
000003c0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
000003d0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00

000003e0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
000003f0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
00000400	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
00000410	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
00000420	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
00000430	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
00000440	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
00000450	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
00000460	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
00000470	00	00	00	00	00	00	00	00	00	00	00	00	FF	FF		

🟩	**SDIO HW Header**
🟧	**macaddr=66:55:44:33:22:11**
🟦	**sromrev=11**
🟪	**boardrev=0x1101**
🟩	**boardtype=0x074c**
🟨	**OTP end**

Max WLAN SW/HW Region size = 9200 bits = 1150 bytes

.

5.2 Creating and Editing 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. Writing to the OTP memory requires a bin file that fits in the OTP memory space.

For the CYW8X359/CYW8X342, the maximum size of the OTP memory is 1150 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 PCIe hardware header and the CIS tuple, as described in the OTP binary map instructions provided in Programming Basic Parameters into OTP Memory.

Note: The OTP binary map file (see Table 7 and Table 8) has been edited to match the example CYW8X359/CYW8X342 OTP binary map described in Table 5 and Table 6.

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 save with a *.bin* file extension so that the data it contains can be programmed into the OTP memory. In this application note, this file is referred *8X359_OTP.bin.*

Table 7 and Table 8 show the hexadecimal OTP binary map template for the CYW8X359/CYW8X342 PCIe revision and SDIO revision, respectively.

Table 7. CYW8X359/CYW8X342 PCIe Hexadecimal OTP Binary Map Template

Offset	0x0	0x1	0x2	0x3	0x4	0x5	0x6	0x7	0x8	0x9	0xa	0xb	0xc	0xd	0xe	0xf
00000000	0f	38	00	38	51	7	e4	14	1c	02	7e	1b	00	8a	00	00
00000010	00	00	00	00	00	00	00	00	54	00	3c	21	64	21	03	32
00000020	5f	18	05	96	28	9f	b6	79	80	80	03	0c	00	40	40	32
00000030	00	5f	f4	75	90	80	00	ee	00	84	08	F0	0b	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	ef	43	00	80	02	00	00	00	f5	3f	00	18	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	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	00	00	00	00	00	00	00	00
00000280	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
00000290	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
000002a0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
000002b0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
000002c0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
000002d0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
000002e0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
000002f0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
00000300	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
00000310	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
00000320	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
00000330	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
00000340	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
00000350	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
00000360	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
00000370	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
00000380	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
00000390	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
000003a0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
000003b0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
000003c0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
000003d0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00

000003e0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
000003f0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
00000400	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
00000410	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
00000420	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
00000430	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
00000440	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
00000450	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
00000460	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
00000470	00	00	00	00	00	00	00	00	00	00	00	00	FF	FF		

Table 8. CYW8X359 SDIO Hexadecimal OTP Binary Map Template

Offset	0x0	0x1	0x2	0x3	0x4	0x5	0x6	0x7	0x8	0x9	0xa	0xb	0xc	0xd	0xe	0xf
00000000	4b	0	ff	ff	0	0	20	4	d0	2	59	43	80	7	19	66
00000010	55	44	33	22	11	80	3	2	1	11	80	2	0	0B	80	3
00000020	1B	51	7	0	0	0	0	0	0	0	0	0	0	0	0	0
00000030	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
00000040	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
00000050	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
00000060	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
00000070	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
00000080	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
00000090	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
000000a0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
000000b0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
000000c0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
000000d0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
000000e0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
000000f0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
00000100	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
00000110	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
00000120	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

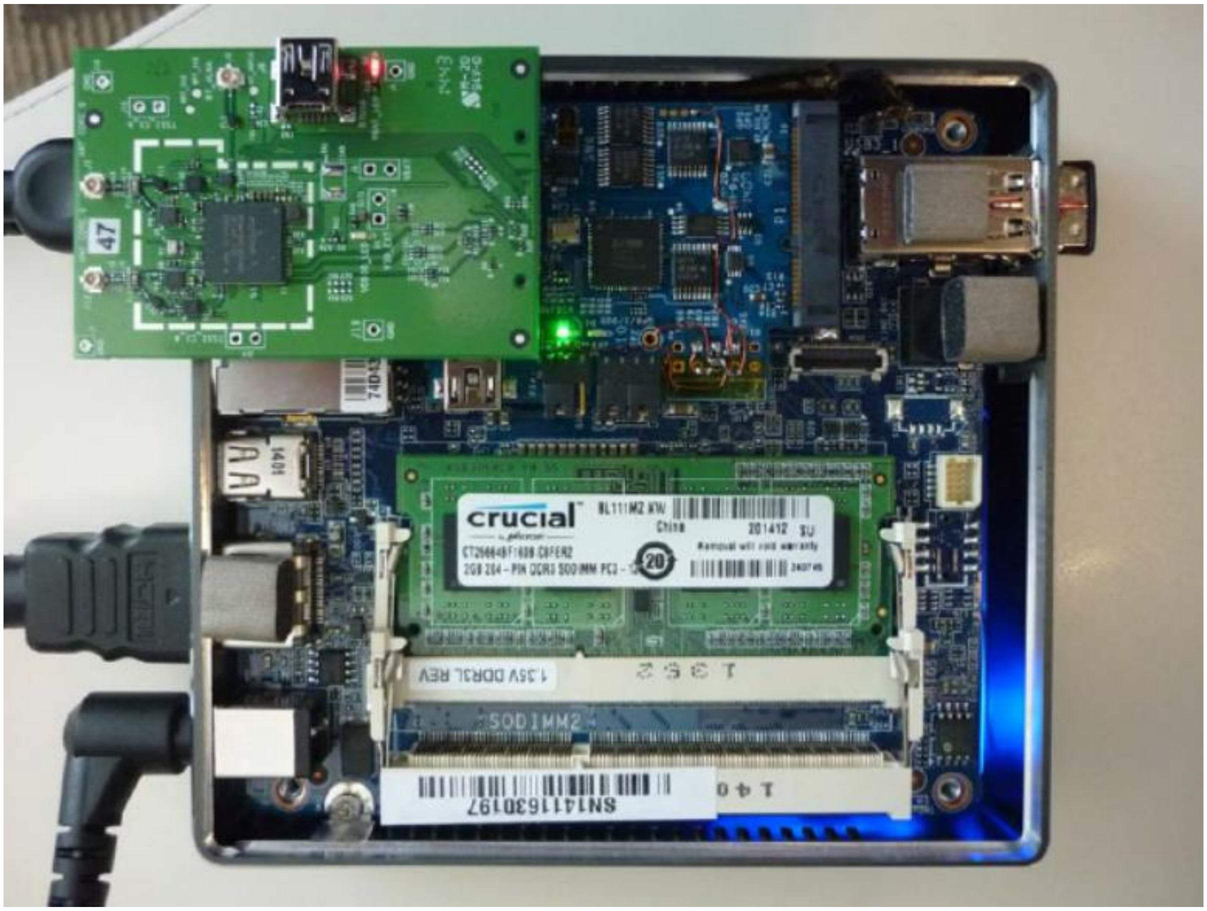
00000130	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
00000140	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
00000150	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
00000160	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
00000170	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
00000180	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
00000190	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
000001a0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
000001b0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
000001c0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
000001d0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
000001e0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
000001f0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
00000200	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
00000210	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
00000220	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
00000230	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
00000240	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
00000250	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
00000260	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
00000270	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
00000280	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
00000290	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
000002a0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
000002b0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
000002c0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
000002d0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
000002e0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
000002f0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
00000300	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
00000310	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
00000320	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

00000330	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
00000340	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
00000350	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
00000360	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
00000370	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
00000380	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
00000390	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
000003a0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
000003b0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
000003c0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
000003d0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
000003e0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
000003f0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
00000400	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
00000410	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
00000420	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
00000430	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
00000440	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
00000450	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
00000460	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
00000470	0	0	0	0	0	0	0	0	0	0	0	0	ff	ff		

6 Programming CYW8X359/CYW8X342 OTP Memory Using Gigabyte Brix

This section outlines the procedure to program the PCIe header to the OTP of a CYW8X359/CYW8X342 device using a Gigabyte Brix Linux box

Figure 2. Brix System Example



The required hardware includes:

- 1x CYW8X359 PCIe board with 60-pin Samtec connector – this is the “DUT”
- 1x Brix box with Cypress image that has FC19 Kernel installed in USB memory
- 1x USB mouse
- 1x USB keyboard
- 1x external monitor with HDMI or DVI connection
- 1x HDMI-to-HDMI cable or HDMI-to-DVI cable
- 1x Ethernet cable
- 1x CYW9MC2EMB60AD interposer card (inserted into the MC slot on Brix)

The required software includes:

- Cypress PCIe MFG driver package containing driver files for CYW8x359 in Linux FC19 (3.11.1) 64-bit platform (typically provided by Cypress).
- *OTP.bin* file containing the CYW8X359 PCIe or SDIO header information. Follow the procedure in 6.1 Programming OTP Memory to program OTP memory using the *OTP_bin* file.

6.1 Programming OTP Memory

Use MFG firmware and follow these steps to program the OTP memory:

1. While powered OFF, connect the Brix to Ethernet, USB mouse and keyboard, and monitor (a HDMI-to-DVI cable is required to connect to a DVI monitor).
2. Connect DUT to the 60-pin connector located in the Brix.
3. Plug in the power to the Brix and the Brix system should be turned ON automatically. On the monitor, you should see the screen booting up to Linux FC19.
4. At prompt, log in as "root". When logged in, type the following to go to the Linux GUI:

```
> startx
```

5. Go to Activities > Terminal to open a command prompt. At the terminal:

   a. Type `> ifconfig -a`.

   b. Copy the mac address for eth0 (for example, 74:d4:35:47:84:d9).

   c. Open another terminal using a text editor of your choice.

   If using vi, type:

```
> vi /etc/sysconfig/network-scripts/ifcfg-eth0p
```

   In this file, modify the MAC address to match it with the copied MAC address (for example, HWADDR=74:d4:35:47:84:d9). Then, save the file.

6. On the terminal, type `> reboot` to reboot the Brix. The Ethernet should work after reboot, and the Brix should be able to connect to network.
7. Once in Linux, copy the CYW8X359 driver files and the *OTP.bin* file to a desired directory.

Note: Check that the results returned by `lspci` includes the slot number of the DUT (03:00.0). The command can be used to check the revision ID after programming and a power cycle.

8. Go to the directory where you copied the CYW8X359 driver files. Issue the driver load command as you would normally do on a Linux system, or:

```
> insmod dhd.ko firmware_path=rtecdc.bin nvram_path=nvram.txt clm_path=4359b1.clm_blob
> ./wl rsdb_mode 0
> ifconfig eth1 192.168.1.101 up
> ./wl ver
```

Note: If driver loads successfully, the command `wl ver` will return the WL version and the driver version.

9. Once the driver is loaded successfully, you are ready to program OTP.

   a. Run the following command to check the CIS dump in the OTP:

```
> wl cisdump
```

b. If your CYW8X359 device has never been programmed with the PCIe or SDIO header in the OTP, check if the cisdump is similar to the following:

```
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
```

c. If you can confirm that CYW8X359 device has never been programmed, then your device has blank CIS and is ready to be programmed. Go to the directory where you copied the *OTP.bin* file.

   For PCIe, run the following command:

   ```
   >wl ciswrite -p OTP.bin
   ```

   For SDIO, run the following command:

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

d. After programming is completed, confirm the OTP by dumping CIS again:

   ```
   > wl cisdump
   ```

   If programming is successful, you should see the dump that looks similar to the following:

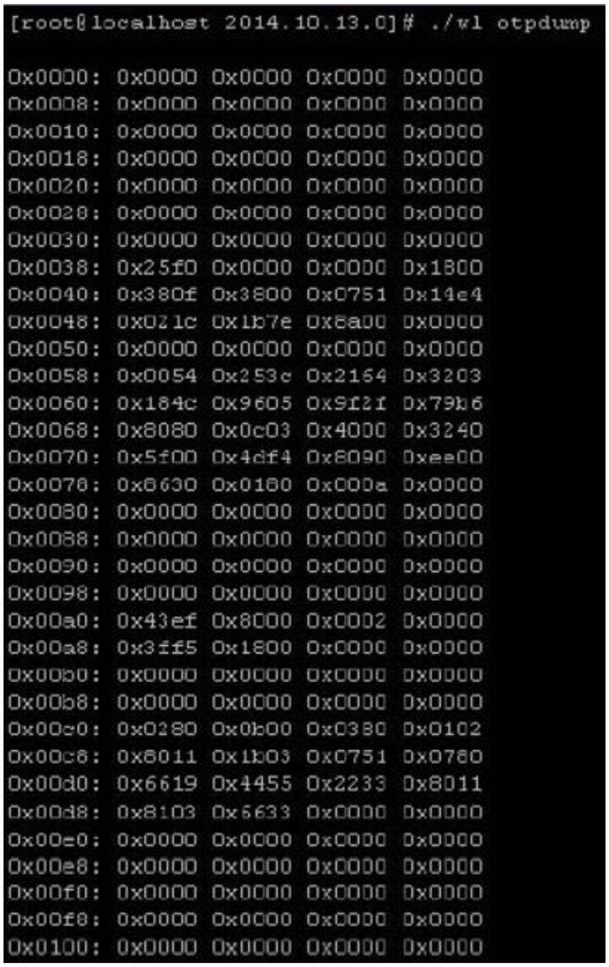
Note: Depending on the contents of your *.bin* file, the CIS dump might vary.

```
Source: 2 (Internal OTP)
Maximum length: 484 bytes
Byte 0: 0x0f 0x38 0x00 0x38 0x37 0x07 0xe4 0x14
Byte 8: 0x1c 0x02 0x7e 0x1b 0x00 0x0a 0x00 0x00
Byte 16: 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00
Byte 24: 0xd4 0x00 0x3c 0x25 0x64 0x21 0x03 0x32
Byte 32: 0x5f 0x3e 0x05 0x96 0x2f 0x9f 0xb6 0x79
Byte 40: 0x80 0x80 0x03 0x0c 0x00 0x40 0x40 0x32
Byte 48: 0x00 0x5f 0xf4 0x4d 0x90 0x80 0x00 0xee
Byte 56: 0x30 0x86 0x80 0x01 0x2b 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 0x88 0x0a 0x03
Byte 88: 0x60 0x01 0x00 0x00 0x00 0x00 0x00 0x00
Byte 96: 0xec 0x43 0x00 0x80 0x02 0x00 0x00 0x00
Byte 104: 0xf5 0x3f 0x00 0x18 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 0xff
```

If the CIS dump matches your *OTP.bin* file, the OTP programming is successful, and the PCIe or SDIO header is correctly programmed to your CYW8X359/CYW8X342 device.

Figure 3 and Figure 4 show the OTP dump after a power cycle which includes revision information. This screen appears when the OTP programming is successful.

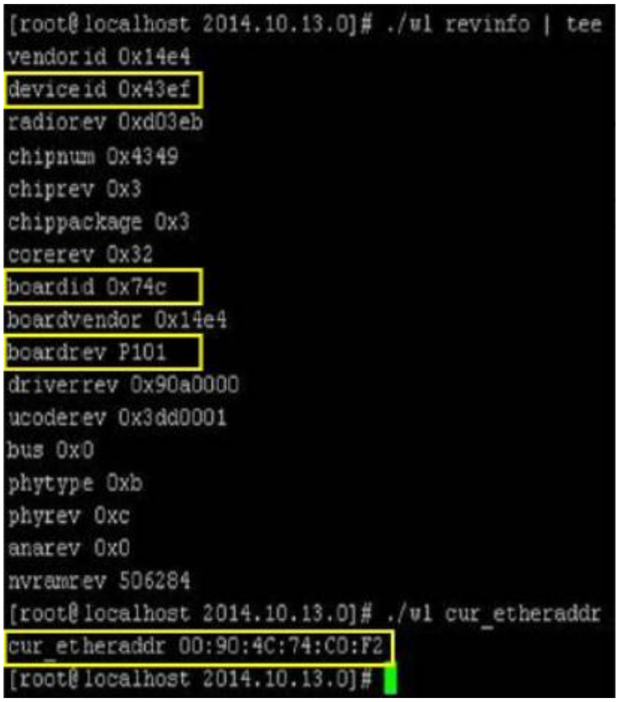
Figure 3. OTP Dump after Power Cycle



```
[root@localhost 2014.10.13.0]# ./wl otpdump

0x0000: 0x0000 0x0000 0x0000 0x0000
0x0008: 0x0000 0x0000 0x0000 0x0000
0x0010: 0x0000 0x0000 0x0000 0x0000
0x0018: 0x0000 0x0000 0x0000 0x0000
0x0020: 0x0000 0x0000 0x0000 0x0000
0x0028: 0x0000 0x0000 0x0000 0x0000
0x0030: 0x0000 0x0000 0x0000 0x0000
0x0038: 0x25f0 0x0000 0x0000 0x1800
0x0040: 0x380f 0x3800 0x0751 0x14e4
0x0048: 0x021c 0x1b7e 0x8a00 0x0000
0x0050: 0x0000 0x0000 0x0000 0x0000
0x0058: 0x0054 0x253c 0x2164 0x3203
0x0060: 0x184c 0x9605 0x9f2f 0x79b6
0x0068: 0x8080 0x0c03 0x4000 0x3240
0x0070: 0x5f00 0x4df4 0x8090 0xee00
0x0078: 0x8630 0x0180 0x000a 0x0000
0x0080: 0x0000 0x0000 0x0000 0x0000
0x0088: 0x0000 0x0000 0x0000 0x0000
0x0090: 0x0000 0x0000 0x0000 0x0000
0x0098: 0x0000 0x0000 0x0000 0x0000
0x00a0: 0x43ef 0x8000 0x0002 0x0000
0x00a8: 0x3ff5 0x1800 0x0000 0x0000
0x00b0: 0x0000 0x0000 0x0000 0x0000
0x00b8: 0x0000 0x0000 0x0000 0x0000
0x00c0: 0x0280 0x0b00 0x0380 0x0102
0x00c8: 0x8011 0x1b03 0x0751 0x0780
0x00d0: 0x6619 0x4455 0x2233 0x8011
0x00d8: 0x8103 0x6633 0x0000 0x0000
0x00e0: 0x0000 0x0000 0x0000 0x0000
0x00e8: 0x0000 0x0000 0x0000 0x0000
0x00f0: 0x0000 0x0000 0x0000 0x0000
0x00f8: 0x0000 0x0000 0x0000 0x0000
0x0100: 0x0000 0x0000 0x0000 0x0000
```

Figure 4. Revision Information and MAC Address



```
[root@localhost 2014.10.13.0]# ./wl revinfo | tee
vendorid 0x14e4
deviceid 0x43ef
radiorev 0xd03eb
chipnum 0x4349
chiprev 0x3
chippackage 0x3
corerev 0x32
boardid 0x74c
boardvendor 0x14e4
boardrev P101
driverrev 0x90a0000
ucoderev 0x3dd0001
bus 0x0
phytype 0xb
phyrev 0xc
anarev 0x0
nvramrev 506284
[root@localhost 2014.10.13.0]# ./wl cur_etheraddr
cur_etheraddr 00:90:4C:74:C0:F2
[root@localhost 2014.10.13.0]#
```

Note: Make sure that you remove the device from the PCIe or SDIO slot before power cycling.

7 Programming CYW8X359/CYW8X342 OTP BD Address

Command:

```
#wl.exe otpraw <bitoffset> <length> <value>
```

Bit offset:

"9744" in case of CYW8X359

Length:

"80"

Value:

50 // Signature. OTP bit offset 9744 is the start of the BT OTP signature.

4F // Signature. Only 2 bytes of signature are for this chip.

10 // Header. Use fixed value of 0x10.

06 // Size of OTP after this byte itself. If only need to program the BD ADDR, use the size value of 0x06.

Ff // BDADDR, 6 bytes; Assuming BDADDR Aa Bb Cc Dd Ee Ff

Ee // BDADDR, 6 bytes

Dd // BDADDR, 6 bytes

Cc // BDADDR, 6 bytes

Bb // BDADDR, 6 bytes

Aa // BDADDR, 6 bytes

9744	9752	9760	9768	9776	9784	9792	9800	9808	9816
50	4F	10	6	Ff	Ee	Dd	Cc	Bb	Aa

Command example:

```
#./wl otpraw 9744 80 0xAaBbCcDdEeFf06104F50
```

Document History

Document Title: AN225666 - OTP Memory Programming and NVRAM Development - CYW8X359/CYW8X342

Document Number: 002-25666

Revision	ECN	Orig. of Change	Submission Date	Description of Change
**	6396956	PCWA	12/14/2018	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

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

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

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