OTP memory programming and NVRAM development - CYW8X373

About this document

Scope and purpose

This application note describes the method for creating an *nvram.txt* file, which is then used to test a new board design, optimize NVRAM values, and program the one-time programmable (OTP) nonvolatile memory in the CYW8X373 device using the PCIe or SDIO host interface for WLAN.

Intended audience

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

- PCIe/SDIO HW header programming
- NVRAM content development and programming basic parameters into OTP memory
- WLAN MAC address programming
- BD_ADDR programming

Associated part family

CYW8X373

Table of contents

About this document **1**
Table of contents **2**
1 Introduction **3**
1.1 IoT resources 3
2 OTP memory programming considerations **4**
3 NVRAM content development and memory programming flow **5**
4 Customizing *nvram.txt* file **6**
4.1 Using *nvram.txt* file template 6
4.2 Editing *nvram.txt* file 18
4.3 Finalizing *nvram.txt* file 18
5 Programming OTP memory **19**
5.1 Programming basic parameters into OTP memory 19
5.2 Creating and editing OTP binary map 25
6 Programming CYW8X373 OTP memory using iMAX6sx **28**
6.1 Programming OTP memory 29
7 Programming CYW8X373 OTP BD address **33**
Revision history **35**
Disclaimer **36**

1 Introduction

CYW8X373 is a single-chip IEEE 802.11a/b/g/n/ac + Bluetooth® 5.0 device for embedded and IoT applications. OTP nonvolatile memory is included in the WLAN section of the device to store board-specific information such as the PCIe header, product ID, manufacturer ID, and MAC address. Excluding the internal header information, up to 768 bytes of user accessible OTP memory is available on CYW8X373 for WLAN information. The application note provides OTP programming information for both PCIe and SDIO host interfaces.

The OTP memory content and an editable NVRAM file (*nvram.txt* file), provides all configuration information used by the WLAN device driver to initialize and configure CYW8X373.

1.1 IoT resources

Infineon provides a wealth of data in Embedded security for the Internet of Things (IoT) to help select the right IoT device for the design and quickly and effectively integrate the device into the design. Infineon 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 Infineon community.

2 OTP memory programming considerations

In embedded designs, the host and device are permanently connected, which is typically done using a hardwired PCIe or SDIO interface. The only entry that is mandatory to be programmed into OTP memory is the PCIe or SDIO header. This is because certain PCIe or SDIO function settings (such as the L1 sub-state for low-power) 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 them 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 an 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 testing the board 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. Infineon 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 (see Table 3) and advanced (see Table 2) 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.

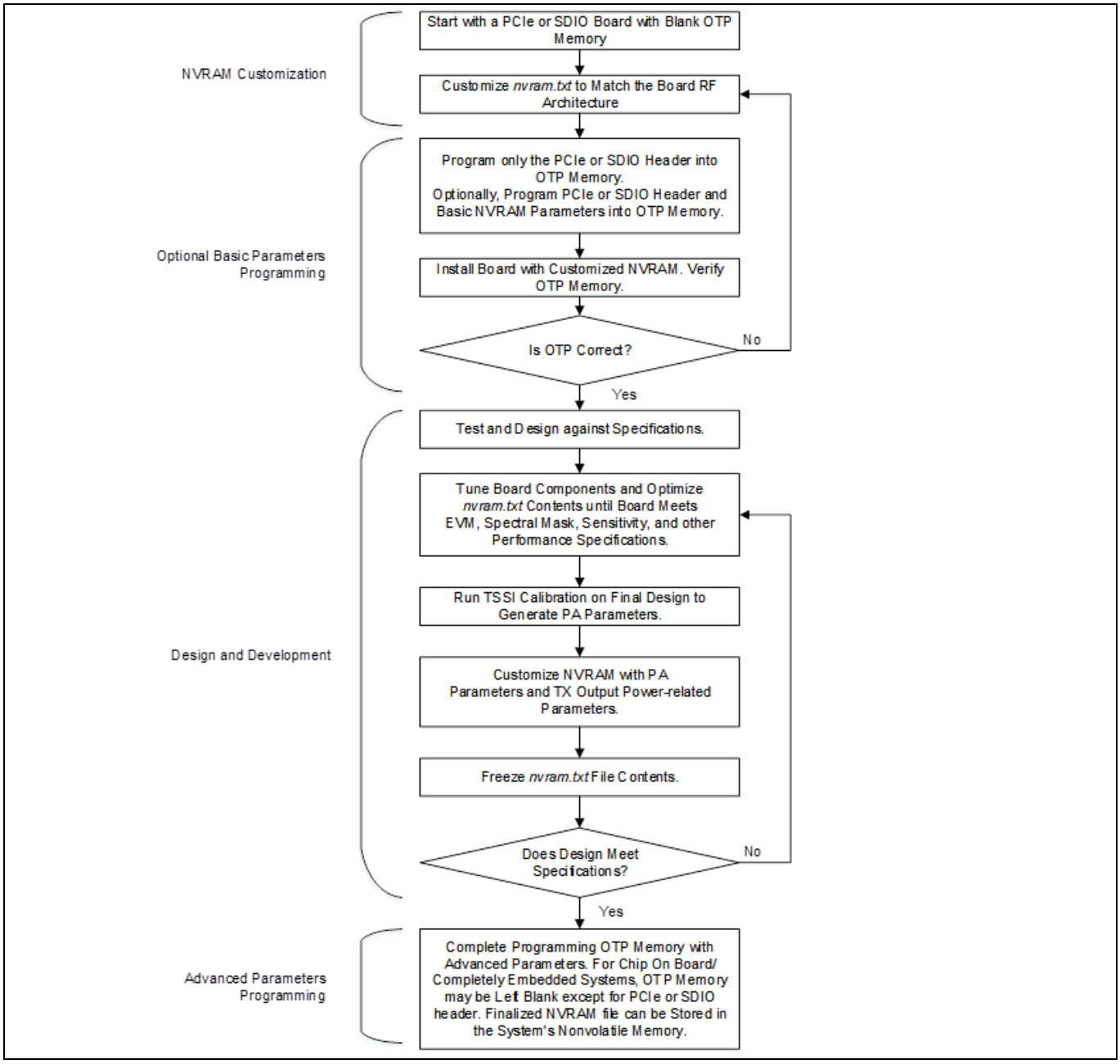


Figure 1 NVRAM development and programming flow of OTP memory

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, Infineon provides an *nvram.txt* file for the specific board design. Typically, the file is named in accordance with the board it supports (for example, *cyw989373wlpsd.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 Infineon community.

Table 1 provides a list of parameters in a typical *nvram.txt* file that are common to dual-band 802.11ac PCIe or SDIO reference design boards.

Parameters in the *nvram.txt* file do not need to be entered in any specific order.

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

Table 1 Infineon-specific NVRAM parameters

NVRAM parameter	Example data	Description
sromrev	11	SROM revision for 802.11ac chips
boardtype	0x84a	This is a critical parameter that should be copied from a similar Infineon reference board design.
tssipos2g	1	This represents if TSSI has positive slope for 2.4 GHz. For CYW8X373, set the value to 1.
tssipos5g	1	This represents if TSSI has positive slope for 5 GHz. For CYW8X373, set the value to 1.
rxchain	1	This specifies the number of rx paths (bit mask). For CYW8X373, set the value to 1.
txchain	1	This specifies the number of tx paths (bit mask). For CYW8X373, set the value to 1.
venid	0x14e4	PCI Vendor ID
devid	0x4418	Chip ID, CYW8X373
manfid	0x2d0	Manufacturer ID
nocrc	1	Check for CRC errors when loading firmware
boardflags boardflags2 boardflags3	0x00000001 0x00000000 0x48202100	Board configuration flag that defines the power topology, external components (iPA/iLNA or ePA/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 0 for ePA boards.

Customizing nvram.txt file

NVRAM parameter	Example data	Description
extpagain5g	2	Supports 5 GHz external PA. Use 2 for iPA boards, and use 0 for ePA boards.
aa2g, aa5g	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
subband5gver	0x4	Defines 5 GHz sub-band allocation
tempthresh	105	This parameter is for Infineon internal use only *Note: Do not modify.*
temps_txduty_lowlimit	0	This parameter is for Infineon internal use only *Note: Do not modify.*
phycal_tempdelta	15	This parameter is for Infineon internal use only *Note: Do not modify.*
temps_period	1	This parameter is for Infineon internal use only *Note: Do not modify.*
temps_hysteresis	20	This parameter is for Infineon internal use only *Note: Do not modify.*
AvVmid_c0	1, 115, 1, 120, 1, 120, 1, 120, 1, 120	This parameter is for Infineon internal use only. *Note: Do not modify.*
swctrlmap_2g, swctrlmap_5g, swctrlmapext_2g/5g	0x00000010, 0x00000050, 0x00000000, 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 before 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 Infineon's transmit signal strength indicator (TSSI) calibration tools.

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

Customizing nvram.txt file

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	0x1107	Board revision used by the WLAN driver. Examples: 0x1107 converts to P107 0x1203 converts to P203
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	1	This variable indicates whether an external LNA bypass is used instead of a TR switch when transmitting. Set to 1 if using an external LNA bypass (for 2.4 GHz).
rxgains5gtrelnabypa0	1	This variable indicates whether an external LNA bypass is used instead of a TR switch when transmitting. Set to 1 if using an external LNA bypass (for 5 GHz). Applies to the low subband.
rxgains5gmtrelnabypa0	1	This variable indicates whether an external LNA bypass is used instead of a TR switch when transmitting. Set to 1 if using an external LNA bypass (for 5 GHz). Applies to the mid subband.
rxgains5ghtrelnabypa0	1	This variable indicates whether an external LNA bypass is used instead of a TR switch when transmitting. Set to 1 if using an external LNA bypass (for 5 GHz). Applies to the high/X1 subband.
rxgains2gelnagaina0	3	This variable defines the 2.4 GHz eLNA gain. Gain (dB) = 2 × rxgains2gelnagaina0 + 6. For rxgains2gelnagaina0 = 3, the gain is 12 dB.
rxgains2gtrisoa0	6	This variable defines the 2.4 GHz isolation provided by the TR switch when transmitting.

Customizing nvram.txt file

NVRAM parameter	Example data	Description
		Isolation (dB) = 2 × rxgains2gtrisoa0 + 8. For rxgains2gtrisoa0 = 6, the isolation is 20 dB.
rxgains5gelnagaina0	3	This variable defines the 5 GHz, low-subband, eLNA gain. Gain (dB) = 2 × rxgains5gelnagaina0 + 6. For rxgains5gelnagaina0 = 3, the gain is 12 dB
rxgains5gtrisoa0	6	This variable defines the 5 GHz, low-subband isolation provided by the TR switch when transmitting. Isolation (dB) = 2 × rxgains5gtrisoa0 + 8. For rxgains5gtrisoa0 = 6, the isolation is 20 dB.
rxgains5gmelnagaina0	3	This variable defines the 5 GHz, mid-subband, eLNA gain. Gain (dB) = 2 × rxgains5gelnagaina0 + 6. For rxgains5gelnagaina0 = 3, the gain is 12 dB
rxgains5gmtrisoa0	6	This variable defines the 5 GHz, mid-subband isolation provided by the TR switch when transmitting. Isolation (dB) = 2 × rxgains5gtrisoa0 + 8. For rxgains5gtrisoa0 = 6, the isolation is 20 dB.
rxgains5ghelnagaina0	3	This variable defines the 5 GHz, high/X1-subband, eLNA gain. Gain (dB) = 2 × rxgains5gelnagaina0 + 6. For rxgains5gelnagaina0 = 3, the gain is 12 dB
rxgains5ghtrisoa0	6	This variable defines the 5 GHz, high/X1-subband isolation provided by the TR switch when transmitting. Isolation (dB) = 2 × rxgains5gtrisoa0 + 8. For rxgains5gtrisoa0 = 6, the isolation is 20 dB.
agbg0, aga0	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. Examples: 0x82 = 2.5 dB (2 + 2 × 0.25) 0x7f = –0.75 dB (–1 + 1 × 0.25)
pa2ga0	-148, 5828, -679	PA parameters for the 2.4 GHz band based on TSSI calibration. pa2ga0 – OFDM
pa5ga0	83, 6045, -553, 57, 5940, -566, 12, 5919, -605, -17,	PA parameters for the 5 GHz band based on TSSI calibration (Low / Mid / High / X1). Sub-band frequency range.

Customizing nvram.txt file

NVRAM parameter	Example data	Description
	5899, -640	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)
pdoffset40ma0	0x0000	5 GHz, 40 MHz BW power detect (PD) offset (1/4 dB steps) in 2s complement format. 4 bits for each sub-band. The most significant nibble is the X1-subband offset.
pdoffset80ma0	0x0000	5 GHz, 80 MHz BW PD offset (in 1/4 dB steps) in 2s complement format 4 bits for each sub-band. The most significant nibble is the X1-subband offset.
maxp2ga0	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.
cckbw202gpo	0x0000	CCK unsigned power offsets (in 1/2 dB steps) for the 20 MHz rates (11, 5.5, 2, 1 Mbps). The most significant nibble is the 11 Mbps offset.
cckbw20ul2gpo	0x0000	CCK unsigned power offsets in 1/2 dB steps for 20 U/L rates (11, 5.5, 2, 1 Mbps). The most significant nibble is the 11 Mbps offset
pdoffsetcckma0	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.
dot11agofdmhrbw202gpo	0x6666	OFDM unsigned power offsets in 1/2 dB steps for 54, 48, 36, and 24 Mbps. The most significant nibble is the 54 Mbps rate offset
ofdmlrbw202gpo	0x0033	OFDM 2.4 GHz, unsigned power offsets in 1/2 dB steps: MCS1 and MCS2: 11n and 11ac 40 MHz BW (most significant nibble) 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 unsigned power offsets in 1/2 dB steps – C9/C8/M7/M6/M5/M4/M3/M0-2.

Customizing nvram.txt file

NVRAM parameter	Example data	Description
		(If separate control of MCS1 and MCS2 is required, then use ofdmlrbw202gpo).
maxp5ga0	0x4A, (low) 0x4A, (mid) 0x4A, (high) 0x4A (X1)	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.
mcslr5glpo	0x0000	5 GHz band low sub-band 12/18 Mbps and MCS1/2, unsigned power offsets in 1/2 dB steps: (0) 20 MHz (least significant nibble) (1) 40 MHz (2) 80 MHz (3) 160 MHz
mcsbw205glpo	0xAA886662	5 GHz, low-subband, 11n/ac, 20 MHz, unsigned power offsets in 1/2 dB steps. The most significant nibble is the power offset for MCS9 and the least significant nibble is for MCS0–2. C9/C8/M7/M6/M5/M4/M3/M0-2
mcsbw405glpo	0xAA886664	5 GHz, low-subband, 11n/ac, 40 MHz, unsigned power offsets in 1/2 dB steps. The most significant nibble is the power offset for MCS9 and the least significant nibble is for MCS0–2. C9/C8/M7/M6/M5/M4/M3/M0-2
mcsbw805glpo	0xAA886664	5 GHz, low-subband, 11n/ac, 80 MHz, unsigned power offsets in 1/2 dB steps. The most significant nibble is the power offset for MCS9 and the least significant nibble is for MCS0–2. C9/C8/M7/M6/M5/M4/M3/M0-2
mcslr5gmpo	0x0000	5 GHz, mid-subband, 11ag/11n/11ac, QPSK, unsigned power offsets in 1/2 dB steps with respect to BPSK: MCS1/2 with respect to MCS0/1/2, and 12/18 Mbps with respect to 6/9 Mbps. (0) 20 MHz (least significant nibble) (1) 40 MHz (2) 80 MHz (3) 160 MHz
mcsbw205gmpo	0xAA886664	5 GHz, mid-subband, 11n/ac, 20 MHz, unsigned power offsets in 1/2 dB steps. The most significant nibble is the power offset for MCS9 and the least significant nibble is for MCS0–2. C9/C8/M7/M6/M5/M4/M3/M0-2

Customizing nvram.txt file

NVRAM parameter	Example data	Description
mcsbw405gmpo	0xAA886664	5 GHz, mid-subband, 11n/ac, 40 MHz, unsigned power offsets in 1/2 dB steps. The most significant nibble is the power offset for MCS9 and the least significant nibble is for MCS0–2. C9/C8/M7/M6/M5/M4/M3/M0-2
mcsbw805gmpo	0xAA886664	5 GHz, mid-subband, 11n/ac, 80 MHz, unsigned power offsets in 1/2 dB steps. The most significant nibble is the power offset for MCS9 and the least significant nibble is for MCS0–2. C9/C8/M7/M6/M5/M4/M3/M0-2
mcslr5ghpo	0x0000	5 GHz, high/X1 band 11ag/11n/11ac, QPSK, unsigned power offsets in 1/2 dB steps with respect to BPSK: MCS1/2 with respect to MCS0/1/2, and 12/18 Mbps with respect to 6/9 Mbps. (0) 20 MHz (least significant nibble) (1) 40 MHz (2) 80 MHz (3) 160 MHz
mcsbw205ghpo	0xAA886664	5 GHz, high/X1 subband, 11n/ac, 20 MHz, unsigned power offsets in 1/2 dB steps. The most significant nibble is the power offset for MCS9 and the least significant nibble is for MCS0–2. – C9/C8/M7/M6/M5/M4/M3/M0-2
mcsbw405ghpo	0xAA886664	5 GHz, high/X1 subband, 11n/ac, 40 MHz, unsigned power offsets in 1/2 dB steps. The most significant nibble is the power offset for MCS9 and the least significant nibble is for MCS0–2. – C9/C8/M7/M6/M5/M4/M3/M0-2
mcsbw805ghpo	0xAA886664	5 GHz, high/X1 subband, 11n/ac, 80 MHz, unsigned power offsets in 1/2 dB steps. The most significant nibble is the power offset for MCS9 and the least significant nibble is for MCS0–2. – C9/C8/M7/M6/M5/M4/M3/M0-2
sb20in40hrpo	0	20in40 OFDM signed power offsets (in 1/2 dB steps) 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 (in 1/2 dB steps) with respect to 20in20 for 64 QAM and above. LSB nibble to MSB nibble: (0) 2.4 GHz band

Customizing nvram.txt file

NVRAM parameter	Example data	Description
		(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 (in 1/2 dB steps) 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
sb40and80hr5glpo	0	5 GHz low sub-band 40in80, 40in160 OFDM signed power offsets (in 1/2 dB steps) 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 (in 1/2 dB steps) 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 (in 1/2 dB steps) 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 (in 1/2 dB steps) 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 (in 1/2 dB steps) for 64 QAM and above. LSB nibble to MSB nibble:

Customizing nvram.txt file

NVRAM parameter	Example data	Description
		(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 (in 1/2 dB steps) 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 (in 1/2 dB steps) 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 (in 1/2 dB steps) 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 (in 1/2 dB steps) 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 (in 1/2 dB steps) 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

Customizing nvram.txt file

NVRAM parameter	Example data	Description
sb40and80hr5ghpo	0	5 GHz high/X1 sub-band 40in80, 40in160 OFDM signed power offsets (in 1/2 dB steps) 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 (in 1/2 dB steps) 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 (in 1/2 dB steps) 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 (in 1/2 dB steps) 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 (in 1/2 dB steps) 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 (in 1/2 dB steps) 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

Customizing nvram.txt file

NVRAM parameter	Example data	Description
sb20in80and160lr5ghpo	0	5 GHz high/X1 sub-band 20in80, 20in160 OFDM signed power offsets (in 1/2 dB steps) 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 (in 1/2 dB steps) for 16 QAM and below. LSB nibble to MSB nibble: (0) 40in80 with respect to 40in40 (1) 40in160 with respect to 40in40 (2) 80in160 with respect to 80in80 (3) 40in160 - 40LL/UU with respect to 40LU/UL
dot11agduphrpo	0	11a/g duplicate mode signed power offsets (in 1/2 dB steps) 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 (in 1/2 dB steps) 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 out-of-band (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.
rssi_delta_2g_c0	-1, -1, -1, -1	Array of measured delta from expected power (in 1 dB step) during cal for 2G (recommended channel 6); first two for 20 MHz, and next two for 40 MHz. There are two for gi 1 and 4 in each bandwidth mode. LSB to MSB nibble:

Customizing nvram.txt file

NVRAM parameter	Example data	Description
		(0) 40MHz BW, -70dBm ref signal power (1) 40MHz BW, -25dBm ref signal power (2) 20MHz BW, -70dBm ref signal power (3) 20MHz BW, -25dBm ref signal power
rssi_delta_5gl_c0	-2, -2, -4, -4, -3, -3	Array of measured delta from expected power (in 1 dB step) during cal for 5G (lower sub-band, recommended channel 5180 MHz); first two for 20MHz, next two for 40 MHz, and last two for 80 MHz. There are two for gi 1 and 4 in each bandwidth mode. LSB to MSB nibble: (0) 80MHz BW, -70dBm ref signal power (1) 80MHz BW, -25dBm ref signal power (2) 40MHz BW, -70dBm ref signal power (3) 40MHz BW, -25dBm ref signal power (4) 20MHz BW, -70dBm ref signal power (5) 20MHz BW, -25dBm ref signal power
rssi_delta_5gml_c0	-2, -2, -4, -4, -4, -4	Array of measured delta from expected power (in 1 dB step) during cal for 5G (mid-low sub-band, recommended channel 5500 MHz); first two for 20 MHz, next two for 40 MHz, and last two for 80 MHz. There are two for gi 1 and 4 in each bandwidth mode. LSB to MSB nibble: (0) 80MHz BW, -70dBm ref signal power (1) 80MHz BW, -25dBm ref signal power (2) 40MHz BW, -70dBm ref signal power (3) 40MHz BW, -25dBm ref signal power (4) 20MHz BW, -70dBm ref signal power (5) 20MHz BW, -25dBm ref signal power
rssi_delta_5gmu_c0	-2, -2, -4, -4, -2, -2	Array of measured delta from expected power (in 1 dB step) during cal for 5G (mid-upper sub-band, recommended channel 5640 MHz); first two for 20 MHz, next two for 40 MHz, and last two for 80 MHz. There are two for gi 1 and 4 in each bandwidth mode. LSB to MSB nibble: (0) 80MHz BW, -70dBm ref signal power (1) 80MHz BW, -25dBm ref signal power (2) 40MHz BW, -70dBm ref signal power (3) 40MHz BW, -25dBm ref signal power (4) 20MHz BW, -70dBm ref signal power (5) 20MHz BW, -25dBm ref signal power
rssi_delta_5gh_c0	-2, -2, -3, -3, -2, -2	Array of measured delta from expected power (in 1 dB step) during cal for 5G (upper sub-band, recommended channel 5795 MHz); first two for 20 MHz, next two for 40 MHz, and last two for 80 MHz. There are two for gi 1 and 4 in each bandwidth mode. LSB to MSB nibble: (0) 80MHz BW, -70dBm ref signal power

Customizing nvram.txt file

NVRAM parameter	Example data	Description
		(1) 80MHz BW, -25dBm ref signal power (2) 40MHz BW, -70dBm ref signal power (3) 40MHz BW, -25dBm ref signal power (4) 20MHz BW, -70dBm ref signal power (5) 20MHz BW, -25dBm ref signal power
powoffs2gtna0	-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).

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 Infineon 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.

Infineon 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 CYW8X373 has 368 bytes of space in the OTP memory available for user data and this is for Wifi only (CIS dump). Total OTP contents are 768 bytes (OTP dump). 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 CYW8X373, there are certain PCIe or SDIO function settings (such as the L1 sub-state for low-power) 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 Infineon and the customer. A majority of the PCIe or SDIO header fields are either generic (and do not need to be changed) or Infineon-specific. There are a few fields that are customer-specific. Coordinate with the Infineon 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
subband5gver	0x8A	2
subband5gver, maxp2ga0, pa2ga0, maxp5ga0, pa5ga0	0x59	38

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

[1] The value for ccode in the *nvram.txt* file is in ASCII format. This value must be converted to hexadecimal format before entering it into the OTP binary map (for example, “US” = “0x55 0x53”).

Programming OTP memory

Table 4 CIS tuple format

Fragment	Description
80	Indicates the beginning of a new tuple. 0x80 is specific to Infineon 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).

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

80 03 02 07 11

- 80 – Beginning of a new tuple.
- 03 – The tag (1 byte) and the value (2 bytes) occupy 3 bytes (total) in the OTP memory.
- 02 – Tag of 0x02 is the identifier for boardrev in the nvram.txt file.
- 07 11 – The value of boardrev in reverse hexadecimal byte or 0x1107.

Table 5 and Table 6 and provide an example OTP binary map for a CYW8X373 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 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 CYW8x373 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	2b	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	18	44 (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	07	11	80	03	1b	4a	08	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 (6)

PCIe Header (Required in OTP)

Other NVRAM Variables (Optional in OTP)

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

boardtype = 0x084a

boardrev = 0x1107

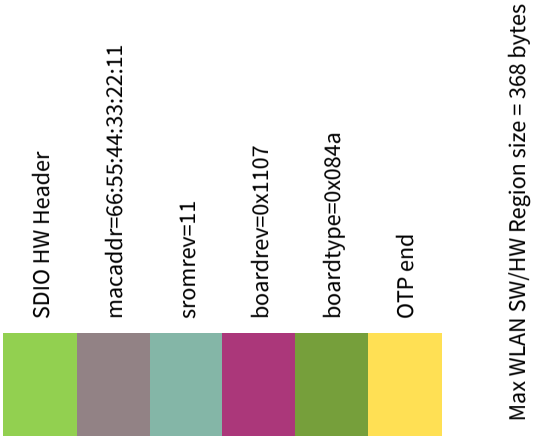
sromrev = 11

CYW8X373 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 Infineon' vendor ID
2b	00 (4)	Time for power on. Use 0x002b for 50 µS for default, unless specified
18	44 (5)	Device ID. 0x4418 is device ID for CYW8X373
00	00 (6)	0x0000 for the end of OTP CIS map
Max WLAN SW/HW Region size = 368 bytes		

Table 6 CYW8X373 OTP Map for SDIO

Offset	0x0	0x1	0x2	0x3	0x4	0x5	0x6	0x7	0x8	0x9	0xa	0xb	0xc	0xd	0xe	0xf
00000000	5b	00	ff	ff	00	00	20	04	d0	02	73	43	80	07	19	66
00000010	55	44	33	22	11	80	03	02	07	11	80	02	00	0b	80	03
00000020	1b	4a	08	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

Programming OTP memory



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 the formatting of the *nvram.txt* file. Writing to the OTP memory requires a bin file that fits in the OTP memory space.

For the CYW8X373, the maximum size of the OTP memory is 368 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 CYW8X373 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* file.

Note: *Save the file 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 as 8X373_OTP.bin.*

Table 7 and Table 8 show the hexadecimal OTP binary map template for the CYW8X373 PCIe revision and SDIO revision, respectively.

Table 7 CYW8X373 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	07	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	2b	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	18	44	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	07	11	80	03	1b	4a	08	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

Table 8 CYW8X373 SDIO hexadecimal OTP binary map template

Offset	0x0	0x1	0x2	0x3	0x4	0x5	0x6	0x7	0x8	0x9	0xa	0xb	0xc	0xd	0xe	0xf
00000000	5b	00	ff	ff	00	00	20	04	d0	02	73	43	80	07	19	66
00000010	55	44	33	22	11	80	03	02	07	11	80	02	00	0b	80	03
00000020	1b	4a	08	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

6 Programming CYW8X373 OTP memory using iMAX6sx

This section outlines the procedure to program the PCIe header to the OTP of a CYW8X373 device using an iMAX6sx FMAC system PC.

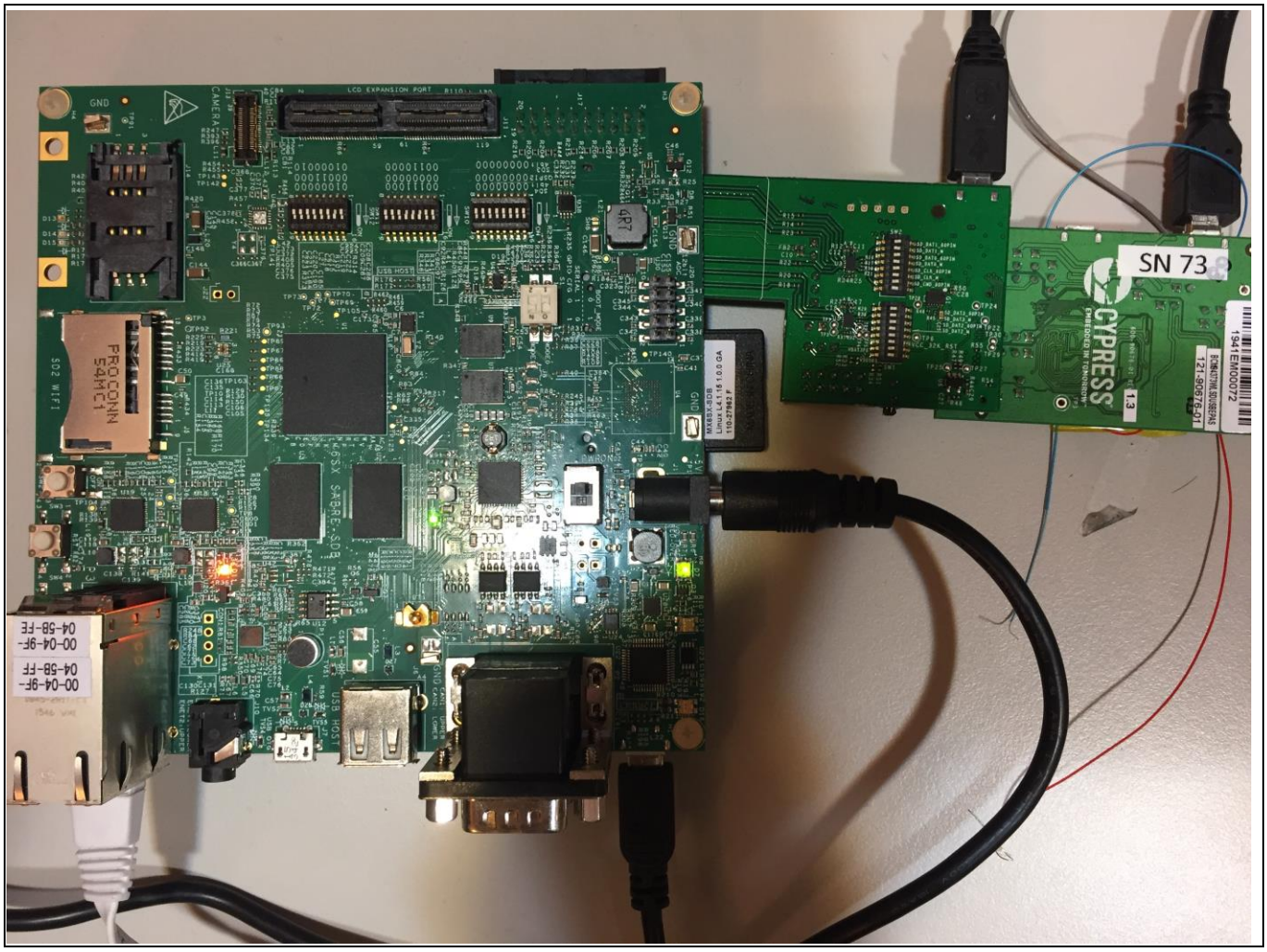


Figure 2 iMAX6sx FMAC system example

The required hardware includes:

- 1x CYW8X373 SDIO board – this is the “DUT”
- 1x iMAX6sx system with Infineon image that has FMAC Kernel installed (4.14.0 or later) in SD card
- 1x Ethernet cable
- 1x CYW9SDIOAD_1 interposer card (inserted into the SD3 slot on iMAX6sx)
- The required software includes:
- Infineon PCIe MFG driver package containing driver files for CYW8x373 in FMAC (4.14.0) platform (typically provided by Infineon).
- *OTP.bin* file containing the CYW8X373 PCIe or SDIO header information. Follow the procedure in 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 iMAX6sx to Ethernet.
2. Connect DUT to the 60-pin connector located in the iMAX6sx.
3. Plug in the power to the iMAX6sx and the iMAX6sx system will be turned ON automatically.
4. At the prompt, with a specific COM port for iMAX6sx, log in as "root".
5. Copy the CYW8X373 driver files and the *OTP.bin* file to a desired directory.
6. Go to the directory where you copied the CYW8X373 driver files. Issue the driver load command as you would normally do on a FMAC system, or:

```
> insmod compat.ko
> insmod mmc_core.ko
> modprobe sdhci-pci
> modprobe rfkill
> insmod cfg80211.ko
> insmod brcmutil.ko
> insmod brcmfmac.ko debug=0x100004
> ifconfig wlan0 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.*

7. 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 CYW8X373 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: 368 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
```

c. If you can confirm that CYW8X373 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
   ```

d. For SDIO, run the following command:

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

e. 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 (for PCIE OTP):

Programming CYW8X373 OTP memory using iMAX6sx

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

```
Source: 2 (Internal OTP)
Maximum length: 368 bytes
Byte   0: 0x0f 0x38 0x00 0x38 0x51 0x07 0xe4 0x14
Byte   8: 0x1c 0x02 0x7e 0x1b 0x00 0x8a 0x00 0x00
Byte  16: 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00
Byte  24: 0x54 0x00 0x3c 0x21 0x64 0x21 0x03 0x32
Byte  32: 0x5f 0x18 0x05 0x96 0x28 0x9f 0xb6 0x79
Byte  40: 0x80 0x80 0x03 0x0c 0x00 0x40 0x40 0x32
Byte  48: 0x00 0x5f 0xf4 0x75 0x90 0x80 0x00 0xee
Byte  56: 0x00 0x84 0x08 0xf0 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 0x00 0x00 0x00
Byte  88: 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00
Byte  96: 0x18 0x44 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: 0x80 0x02 0x00 0x0b 0x80 0x03 0x02 0x07
Byte 136: 0x11 0x80 0x03 0x1b 0x4a 0x08 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
```

Programming CYW8X373 OTP memory using iMAX6sx

```
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
```

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 CYW8X373 device.

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

7 Programming CYW8X373 OTP BD address

Command:

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

Bit offset:

The offset of BD address will change when the OTP patch changes. it is recommended that you follow these steps to calculate the bit offset dynamically:

1. Check the value at 16-bits offset 0x0228. If the value is 0x4f50 (Signature), the OTP contents are valid.
2. Check the value at 8-bits offset 0x022a. If the value is 0xab, it indicates OTP patch.
3. Check the value at 8-bits offset 0x022b. The value indicates the length of OTP patch.
4. Bit offset of OTP BD address = 0x022b + length of OTP patch + 0x01.

For example:

The value at 16-bits offset 0x0228 is 0x4f50.

The value at 8-bits offset 0x022a is 0xab.

The value at 8-bits offset 0x022b is 0x4e.

The bit offset of OTP BD address = 0x022b + 0x4e + 0x01 = 0x027a.

Length:

“64”

Value:

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

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

ff // BDADDR, 6 bytes; Assuming BD address is 0xaabbccddeeff, and bit offset is 0x027a (= 5072 bits).

ee // BDADDR, 6 bytes.

dd // BDADDR, 6 bytes.

cc // BDADDR, 6 bytes.

bb // BDADDR, 6 bytes.

aa // BDADDR, 6 bytes.

Bit offset:	0x027a	0x027b	0x027c	0x027d	0x027e	0x027f	0x0280	0x0281
Values:	10	06	ff	ee	dd	cc	bb	aa

Programming CYW8X373 OTP BD address

Command example:

```
#./wl otpraw 5072 64 0xaabbccddeeff0610
```

Revision history

Document version	Date of release	Description of changes
**	2020-01-16	New Application Note.
*A	2020-03-20	Content updates
*B	2020-06-22	Corrected PCIe Tpoweron default value to 0x2b for 50uS. Modified SDIO header 0x0 to '5b' from '4b' in the hex value corresponds to - device in SDIO 3.0 mode with Max F2 block size as 256.
*C	2023-05-08	Add 0xFFFF to the end of OTP CIS map. Migrated to IFX template.
*D	2024-01-04	Deleted 0xFFFF at the end of OTP CIS map for the latest firmware.

Trademarks
All referenced product or service names and trademarks are the property of their respective owners.

Edition 2024-01-04
Published by

Infineon Technologies AG
81726 Munich, Germany

© 2024 Infineon Technologies AG. All Rights Reserved.

Do you have a question about this document?
Email: erratum@infineon.com

Document reference
002-29472 Rev. *D

Important notice
The information contained in this application note is given as a hint for the implementation of the product only and shall in no event be regarded as a description or warranty of a certain functionality, condition or quality of the product. Before implementation of the product, the recipient of this application note must verify any function and other technical information given herein in the real application. Infineon Technologies hereby disclaims any and all warranties and liabilities of any kind (including without limitation warranties of non-infringement of intellectual property rights of any third party) with respect to any and all information given in this application note.

The data contained in this document is exclusively intended for technically trained staff. It is the responsibility of customer's technical departments to evaluate the suitability of the product for the intended application and the completeness of the product information given in this document with respect to such application.

Warnings
Due to technical requirements products may contain dangerous substances. For information on the types in question please contact your nearest Infineon Technologies office.

Except as otherwise explicitly approved by Infineon Technologies in a written document signed by authorized representatives of Infineon Technologies, Infineon Technologies' products may not be used in any applications where a failure of the product or any consequences of the use thereof can reasonably be expected to result in personal injury.