AR92xx Family EEPROM Device Configuration Guide

March 2010

ATHEROS®
COMMUNICATIONS

Subject to change without notice.

Notice

The information in this document has been carefully reviewed and is believed to be accurate. Nonetheless, this document is subject to change without notice, and Atheros Communications, Inc. (Atheros) assumes no responsibility for any inaccuracies that may be contained in this document, and makes no commitment to update or to keep current the contained information, or to notify a person or organization of any updates. Atheros reserves the right to make changes, at any time, in order to improve reliability, function or design and to attempt to supply the best product possible. Atheros does not represent that products described herein are free from patent infringement or from any other third party right.

No part of this document may be reproduced, adapted or transmitted in any form or by any means, electronic or mechanical, for any purpose, except as expressly set forth in a written agreement signed by Atheros. Atheros or its affiliates may have patents or pending patent applications, trademarks, copyrights, maskwork rights or other intellectual property rights that apply to the ideas, material and information expressed herein. No license to such rights is provided except as expressly set forth in a written agreement signed by Atheros.

ATHEROS MAKES NO WARRANTIES OF ANY KIND WITH REGARD TO THE CONTENT OF THIS DOCUMENT. IN NO EVENT SHALL ATHEROS BE LIABLE FOR DIRECT, INDIRECT, SPECIAL, INCIDENTAL SPECULATORY OR CONSEQUENTIAL DAMAGES ARISING FROM THE USE OR INABILITY TO USE THIS PRODUCT OR DOCUMENTATION, EVEN IF ADVISED OF THE POSSIBLITY OF SUCH DAMAGES. IN PARTICULAR, ATHEROS SHALL NOT HAVE LIABILITY FOR ANY HARDWARE, SOFTWARE, OR DATA TRANSMITTED OR OTHERWISE USED WITH THE PRODUCT, INCLUDING THE COSTS OF REPAIRING, REPLACING, INTEGRATING, INSTALLING OR RECOVERING SUCH HARDWARE, SOFTWARE OR DATA. ATHEROS SPECIFICALLY DISCLAIMS THE IMPLIED WARRANTIES OF MERCHANTIBILITY AND FITNESS FOR A PARTICULAR PURPOSE AS THEY MIGHT OTHERWISE APPLY TO THIS DOCUMENT AND TO THE IDEAS, MATERIAL AND INFORMATION EXPRESSED HEREIN.

Document Number: 984-00019-017 MKG-0375 Rev. 4

Revision History

Revision	Description of Changes
March 2010	Updated the data structure for Table 2-6; format the descriptor in Appendix A
July 2009	Added the USB Information in EEPROM Appendix
April 2009	Updated EEPROM book: ■ Changes to Table 1-5, Switch Table Operation ■ Changes to Table 2-5, Additional PCI Express Configuration Register Information
December 2007	Updated EEPROM contents for version 14.8 and 14.9. Update to explicitly list support for chipsets AR913x, AR916x, AR922x, and AR928x.
October 2007	Updated EEPROM contents for version 14.7
June 2007	■ Byte reversed all EEPROM tables to show Little Endian format with LSB shown on the right ■ Added additional description of the PCI/PCI Express configuration register initialization ■ Added EEPROM location offset to some of the byte based tables for additional clarity
January 2007	Updated to minor revision 14.3 ■ Added new parameters: deviceType, bswAtten, bswMargin, opFlags bits HT40 5 GHz and HT40 2 GHz. ■ Revamped figure with new parameters as well as 16-bit offset EEPROM location information for Base and Modal structures. ■ Added application details for switch table and the parameters it uses.
April 2006	AR92xx initial release for EEPROM revision 14.0

Contents

List of Tables

Preface

This document provides information on board calibration and board variation values. Storing board data and board calibration information of the Atheros AR5008, AR913x, AR916x, AR922x, and AR928x devices is required for these solutions. This data is typically stored in an EEPROM or target Flash memory system. While not always existing in a physical EEPROM, this collection of data is often referred to throughout this document as EEPROM configuration data. Starting from the AR5008, EEPROM contents are based on the same parameters used in previous Atheros wireless solutions, but the parameters have been restructured to work with these new hardware features.

NOTE: This document uses the term AR92xx to refer to the AR5008, AR913x, AR916x, AR922x, and AR928x family of products.

About this Document

This document consists of the following chapters and appendix:

Chapter 1 **EEPROM Device Configuration**—Describes the contents stored on the EEPROM.

Chapter 2 **Board Data**—Describes the EEPROM board data.

Appendix A **EEPROM Board Data Structure File**—Describes the EEPROM board data contents structure file.

Appendix B **AR7010 USB Information in the EEPROM**—Describes the the USB-related contents in AR7010 platform EEPROM.

Audience

This document is intended for Atheros customers involved with the definition, design, and implementation of modules deploying the Atheros AR92xx chip sets.

Additional Resources

Atheros Reference Design hardware, software, and documentation contain proprietary information of Atheros Communications, Inc., and are provided under a license agreement containing restrictions on use and disclosure, and are also protected by copyright law. Reverse engineering of this hardware, software, or documentation is prohibited.

This guide assumes that the reader has studied and is familiar with the *AR5008 Sample Manufacturing Test Flow* and the *AR5008 ART Reference Guide.*

1

EEPROM Device Configuration

This chapter describes the details of the device configuration information stored on the AR92xx solution. The target drivers use configuration information to ensure optimum and regulatory certified performance of the wireless network interface.

NOTE: This document uses the term AR92xx to refer to the AR5008, AR913x, AR916x, AR922x, and AR928x family of products.

The target driver loads three types of information from the board data information: solution-specific parameters to make the device function correctly for all external board components and regulatory requirements, individual card calibration data to account for part variance and achieve matching system results across a solution, and AR92xx-specific values that identify the version of the board.

NOTE: The AR92xx board data layout design contains similar parameters to previous Atheros chip sets, but the data organization has been streamlined to give maximum flexibility for varying design solutions.

The data layout has also been reformatted to become a programming data structure presented in Appendix A, so data can be instantly mapped by a compiler making reads and writes of the data efficiently.

The current minimum space required for an AR92xx design is 4 KB, whether residing on an EEPROM or inside local flash. To learn more about how this calibration information is obtained and programmed for each unit, see the *AR5008 Sample Manufacturing Test Flow.*

NOTE: Though complexity has increased with up to three radios working in concert, Atheros has tried to maintain the known working techniques used by previous designs by extending them to cover three radios working as a single WLAN device.

Determining Concepts

This section discusses concepts used to determine what information is stored in board data and how to use that information. These concepts include:

- "Piecewise Linear Abstraction"
- "Frequency Piers"
- "Power Detector Calibration"
- "Power Detector Calibration Frequencies"
- "Target Power"
- "Target Power Frequencies"
- "CTL Indexes"
- "Conformance Testing Limits"
- "Country or Domain Code"
- "Support of Multiple Regulatory Domains"
- "Operating Power Algorithm"
- "Switch Table Operation"

This section presents the techniques Atheros uses to calibrate the analog properties of each individual card. In most cases these techniques take a large amount of measured data across frequency and power levels and presents the data representation in the EEPROM's confined space.

Piecewise Linear Abstraction

Piecewise linear abstraction (PLA) technique captures general dependence accurately if it is sampled at appropriate turning points (TPs) and linearly interpolated between the TPs. Figure 1-1 demonstrates how the PLA scheme maintains general dependence accuracy if appropriate TPs are selected. This example shows the broader 5 GHz spectrum with various turning points.

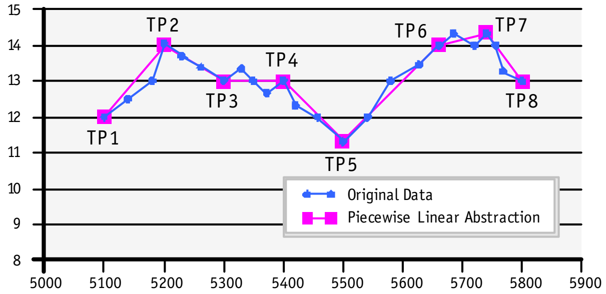


Figure 1-1. **PLA Scheme Applied to a General Dependence**

NOTE: Because these dependences arise from statistical variations of parameters and their interplay, the nature of dependences can vary from card to card. Therefore TPs may be at different locations for each card and fixing the locations of sampling points will, in general, not preserve the data with high accuracy for all cards.

A high degree of accuracy for each card can be preserved if the locations of the TPs are also stored with the sampled values for a sufficient number of TPs. This theme is central in the approach adopted by Atheros to store any NIC-specific or subsystem-specific calibration information on the EEPROM. This enables card manufacturers to deliver the highest level of performance accuracy tailored for each individual card.

Frequency Piers

The PLA concept applies to any kind of dependence. Figure 1-2 demonstrates how the PLA scheme can extend to a set of curves to accurately reproduce an original dataset by sampling a few TPs if the sampling points are chosen well.

When the PLA scheme is applied to the dataset obtained by measuring the output power over a range of frequencies, the TPs for this family of curves are referred to as the frequency piers.

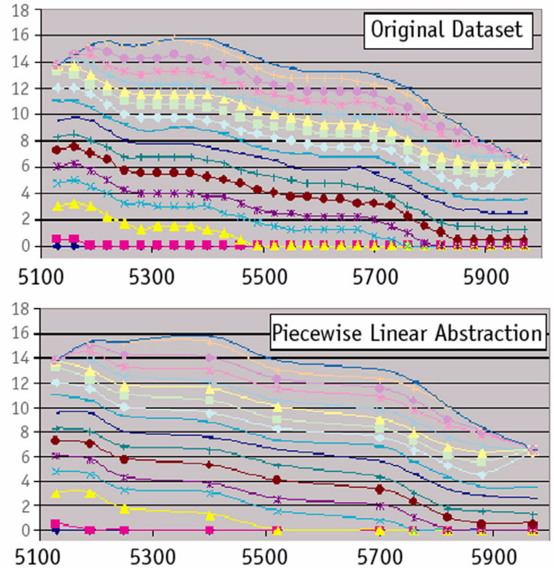


Figure 1-2. **Abstraction of a Set of Curves Through PLA Scheme**

For AR92xx, the output power level is controlled by internal static gain tables. The power level is corrected in a feedback loop through an external power detector. The power detector's response voltage correlates to a calibrated dBm value. The nature of RF circuits is sensitive to the impedance match between various stages. The response of external components, such as the power amplifier (PA), the power detector, and passive elements, typically depends on the frequency. Thus the power detector's voltage over the entire range of channels must be conveyed accurately to the driver for each unit.

Though the manufacturing calibration can measure power detector voltages across channels, correlate these values for similarity then design a list of the best TP to capture the data, for calibration expediency this is performed once per solution and then a fixed list of frequency piers applies to all boards using that solution. This per solution calibration is called FORCE_PIERS.

In the FORCE_PIERS mode, up to eight frequency piers for 5 GHz and up to four for 2 GHz are determined from the pilot runs. The list of piers is specified as FORCE_PIERS_LIST in the **calSetup_XXXX.txt** files. Power measurements are performed only for these channels and stored on the EEPROM as the calibration data. Because measurements are not performed at all frequencies from 4.9 GHz to 5.85 GHz in steps of 10 MHz, the FORCE_PIERS mode runs faster. This speed enhancement comes at the cost of accuracy, therefore, a thorough evaluation should be made before deciding on the piers to use.

Power Detector Calibration

Each wireless device is required to comply with local emission regulations and 802.11 spectral limitations. Due to the sensitive nature of RF circuit design, for each manufactured device to provide an optimal level of throughput performance and output power, it is essential to calibrate each device and store the raw performance capability information in the EEPROM.

For transmit power control at any channel, the AR92xx must be programmed with a 128-entry power detector analog to digital convertor (PDADC) table indexed in 0.5-dB steps. Figure 1-3 shows an example of PDADC curves across frequency in 5 GHz.

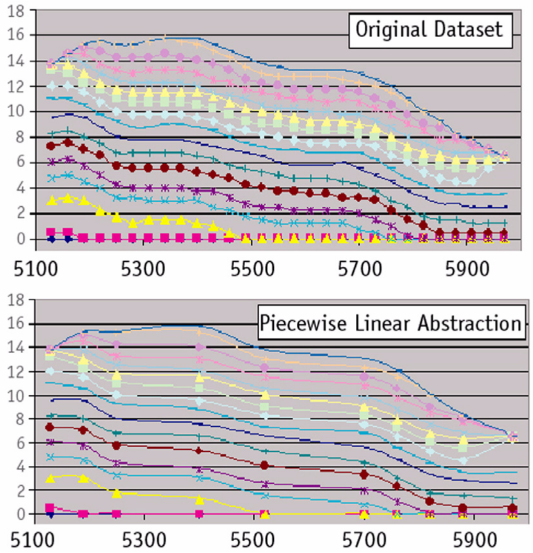


Figure 1-3. **Abstraction of the PDADC**

The PDADC table essentially conveys the calibration information on the power detector feedback voltage as measured by the built-in ADC. The ADC can be used with any combination of up to four gain values (1/2x, 1x, 2x, 4x) to cover a wide dynamic range. PDADC values for all pd_gain values used are spliced at appropriate transition levels to create the 128-entry PDADC table. PDADC values are stored for every 0.5-dB step in output power.

At the transition levels, some amount of overlap is maintained for pd_gains both above and below. Transition levels and overlap is programmed into the chip by the driver with the configuration file.

NOTE: Although the PDADC table supports up to four gain sets, only two gain sets are required to be calibrated and stored in the AR92xx board data, which should provide 25 dB dynamic range in power if the right gain sets are selected.

A snapshot of the raw power capability of the wireless device over the entire frequency range is stored in the board data. For all intermediate channels, the 128-entry PDADC table is reconstructed by the driver from this snapshot at eight frequency piers through interpolation using the PLA scheme.

At each frequency pier, an idealized power versus PDADC dependence for AR92xx-based designs is shown in Figure 1-4. A piecewise linear approximation of the dependence is stored in the board data. To accurately cover the entire range of output power, calibration information for up to two pd_gain values is stored in the board data. Typically pd_gains of 1x and 4x are sufficient to accurately cover the entire range with sufficient overlap.

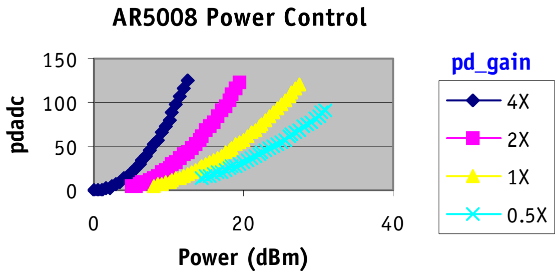


Figure 1-4. **Typical PDADC Dependence on Output Power for a Given Channel**

This section summarizes the format for data stored at a frequency pier. A piecewise linear approximation of PDADC versus output power dependence is stored between appropriate transition levels for all pd_gains. The smallest pd_gain is used for the highest power levels. Five intercepts are stored for all pd_gains for greater accuracy across the operating power region.

For example, AR92xx-based designs may use only two pd_gains: 1x and 4x. pd_gain = 1x is used for 11 dBm and above power levels and five intercepts are stored for this pd_gain. pd_gain = 4x is used for power levels smaller than 11 dBm and also using five intercepts. An overlap of 6 dB is used for both pd_gains around the transition level of 11 dBm.

The AR92xx design can have up to three radios, or chains. Each radio that transmits must have its own calibrated power detector. Thus the AR92xx design provides calibration space for three PDADC tables across frequency. The radio specific arrays are laid out corresponding to physical radio attachment. A design using transmit on radios 0 and 2 would have PDADC information stored in PDADC chain array element 0 and 2. The XPD gain is a 4-bit mask that determines which pd_gains are used as well as how many are present in the PDADC values shown in Table 1-1.

Table 1-1. **PDADC Values**

Value	Description
pwr_PdGain_0	An array of five 8-bit power values describing the power level in 0.5 dB steps achieved for the given Vpd array element.
Vpd_PdGain_0	PDADC value corresponding to the pwr_PdGain_0 array element for the lower pd_gain. Vpd_PdGain values are stored in a 8-bit unsigned integer format and represent the output of the power detector ADC. PDADC values for intermediate power levels are linearly interpolated from these sampling points.
pwr_PdGain_1	The same value type as the pwr_PdGain_0 but for the second pd_gain.
Vpd_PdGain_1	The same value type as the Vpd_PdGain_0 but for the second pd_gain.
pwr_PdGain_2	The same value type as the pwr_PdGain_0 but for the third pd_gain.
Vpd_PdGain_2	The same value type as the Vpd_PdGain_0 but for the third pd_gain.
pwr_PdGain_3	The same value type as the pwr_PdGain_0 but for the fourth pd_gain.
Vpd_PdGain_3	The same value type as the Vpd_PdGain_0 but for the fourth pd_gain.

Power Detector Calibration Frequencies

A block of 12 bytes stores pier locations for up to eight 5 GHz and four 2 GHz frequency piers expressed in 8-bit frequency representation. See Table 1-2.

Table 1-2. **Power Detector Calibration Frequencies**

Formula	Description
fbin = (freq – 4800) / 5 if 4800 ≤ freq < 6080	Relates the frequency in the 5 GHz range (freq) to the 8-bit value stored in the board data (fbin)
fbin = (freq – 2300) for 2300 < freq ≤ 2555	Relates the frequency in 2 GHz range (freq) to the 8-bit value stored on the EEPROM (fbin)

A value of 0xFF for fbin indicates an unused pier.

Target Power

The maximum power that satisfies all IEEE specification requirements (e.g., spectral mask) and performance criteria (that is, < 10% packet error rate (PER)) is determined for a particular subsystem design through a pilot run over a statistical ensemble of NICs. This power is referred to as target power.

This measurement is not performed individually for each card, and the target power does not take into account the regulatory domain's limited power. Target power is an indication of raw capability of a particular card type, regardless of the regulatory domain where the cards are used. Target powers can also be solution specific as required by a vendor to match previous product power level constraints.

Generally, a unique target power exists for each rate. However, for all Atheros reference designs, the rates of 6–24 Mbps have the same target power (spectral mask limited) and the rates 1, 2, 5.5, 11, 36, 48, and 54 Mbps have their own target power (PER-limited) over the entire frequency range.

The AR92xx designs support target powers for the new 802.11n MCS rates. Target powers can be individually specified for MCS 0 to MCS 7. The hardware maps these values one to one onto MCS 8 to MCS 15 (MCS 8 to MCS 15 have two in band phase different data streams compared to MCS 0 to MCS 7). Two sets of MCS rates (HT20 and HT40) are stored and retrieved across frequencies.

The 802.11n draft document also contains a convention of using a control and an extension channel to allow HT40 transmissions while monitoring and using control traffic to manage two legacy channels. When running in dynamic HT40 mode, calibration target powers are computed separately for the frequencies of the control channel, the extension channel, and the HT40 channel that operates on top of the two legacy channels. No extension channel specific to EEPROM target powers as the extension channel uses the normal target power information for its operating channel.

Target Power Frequencies

The target power for all rates or rate groups has a dependence on the frequency. The board data has provisions for vendors to specify a number of TPs per target power rate group of this dependence under the PLA scheme. These TPs, used for conveying the target power information, are referred to as target power frequencies and are determined by the vendor after analyzing data gathered during the pilot run and conveyed to the calibration routine by the file **calTargetPower.txt**. Refer to the *AR5008 Sample Manufacturing Test Flow* for more information.

CTL Indexes

In a regulatory domain, the frequency bands open for public infrastructure are typically interspersed with the restricted bands for military or government use. The extreme operating channels in the open frequency bands are referred to as band edges. For example, the band edge channel centers in a UNII band of 5.15 GHz–5.35 GHz are 5.18 GHz and 5.32 GHz.

Special consideration is needed to determine the transmit power at the band edges to ensure compliance with the regulations in the adjacent restricted frequency band. Channels that fall within the open band (between the band edges) do not require this special consideration.

Because OFDM, CCK, HT20 and HT40 modes have different power signatures, they require different power levels. The 802.11b band edges provide CCK band edge information. Given 802.11g operation requires both CCK and OFDM, the 802.11g band edges specify the OFDM operation limits and an 802.11g CTL implies a search through the CTL indexes for a matching domain 802.11b CTL to get the CCK band edge limitations. AR92xx designs require specifying a number of regulators as well as a number of modes, thus 24 indexes are provided in the EEPROM storage.

The world mode operation initializes without knowing the current country location (until an 802.11d beacon is heard). Thus, world mode must initialize its power levels with the lowest power across all CTL indexes that applies to the channel being used.

Conformance Testing Limits

Significant similarities exist in the boundaries of the open bands in several regulatory domains, because of how the frequency bands have been opened for allocation in the 5 GHz and 2 GHz range worldwide. Therefore several regulatory domains exist with identical sets of band edges. Conformance testing limits (CTLs) leverage this overlap, delivering a simplified mechanism supporting several regulatory domains in manufacturing. It is essential to convey band edge maximum power information to the driver using the EEPROM for all regulatory domains where the NIC is targeted. This data is subsystem design-specific and gathered during the pilot run. If a regulatory domain-based approach is used to store this information on the EEPROM, considerable redundancy exists for domains with overlapping band edges.

AR92xx designs contain multiple radios of which all of or one of the radios can be set to transmit at one time. Given the power signature changes based on the number of radios transmitting, the CTL power space is triplicated and can then be indexed based on the number of radios transmitting. Index 0 of this chain space array is used when one radio is transmitting. Index 1 applies to two radios, and Index 2 applies to three radios.

To alleviate this redundancy and maximize the number of regulatory domains that can be supported by a NIC, a CTL is defined to be a unique set of band edges and adjacent restricted band regulations.

For example:

- If RD1 and RD2 have a permitted band from 5180–5240 MHz and the same set of restrictions for frequencies below 5180 and frequencies higher than 5240, then RD1 and RD2 can belong to the same CTL.
- If RD3 also has a permitted band from 5180–5240 MHz, but tolerates higher power for frequencies below 5180 (i.e., 2 dB higher tolerance at 5160), then RD3 can not belong to the same CTL as RD1 and RD2.
- If RD4 does not permit any transmission in 5180–5240 MHz, but has a permitted band from 5400–5520 MHz, then RD4 can belong to the same CTL as RD1 and RD2 because the two bands do not overlap. The CTL would now contain band edges 5180, 5240, 5400, and 5520. This is possible because the software contains a list of legal channels in each regulatory domain, so for RD4, it will not even look at 5180 and 5240.

Regulatory domains of the band edges that appear in a CTL belong to that CTL. The CTL may contain additional band edge pairs, thus providing data for one CTL enables support for all regulatory domains belonging to that CTL. Up to 24 CTLs are supported in the EEPROM layout.

EEPROM CTLs also contain continuous application flags in the CTLs: one for each CTL frequency. In these 1-bit flags, 0 indicates that the CTL frequency is a band edge and 1 indicates that the CTL acts as a band edge and a continuing limit for frequencies greater than this band edge until the next band edge in the list. When a CTL flag is set, it applies to all frequencies greater than or equal to the CTLs frequency up to but not including the frequency listed in the following CTL. In some cases, the regulatory stipulations imposed outside of this band may restrict power output at not only the band edges, but also for some channels within the band. It then becomes necessary to specify limits on these in-band channels as well, for that CTL. The non-edge flags are introduced to handle such cases. The design to use these flags is:

- All frequencies specified in a CTL must be arranged in ascending order.
- An in-band frequency marks the beginning of the channel range to apply the corresponding CTL limit to. This range goes up to and includes all following channels. It is permitted to specify only in-band frequency CTLs or even a single in-band CTL to cover an entire regulatory band.

These examples demonstrate how in-band frequency can be used.

Example 1

A band exists from 5400–5600 MHz. Out of band regulations require the power be limited to 12 dBm at the band edges, 13 dBm for 5420–5520, and 12.5 dBm for 5540–5580 MHz. Table 1-3 on page 1-9 demonstrates how to convey this information using the non-edge flags.

Table 1-3. **Non-Edge Flag Usage in CTLs (Example 1)**

	Band Edge	In-band Freq	In-band Freq	Band Edge	Next Band
CTL Freq	5400	5420	5540	5600	...
CTL Limit	12	13	12.5	12	...
Non-edge Flag	0	1	1	0	...

Example 2

A band exists from 5400–5500 MHz, out of band regulations require the power be limited to 13 dBm at starting band edge (5400), 15 dBm for 5420–5480 MHz, and 12 dBm at the ending band edge (5500). Table 1-4 demonstrates how to convey this information using the non-edge flags.

Table 1-4. **Non-Edge Flag Usage in CTLs (Example 2)**

	Previous Band	Band Edge	In-band Freq	Band Edge	Next Band
CTL Freq	...	5400	5420	5500	...
CTL Limit	...	13	15	12	...
Non-edge Flag	...	0	1	0	...

Country or Domain Code

A unique 14-bit code identifies the intended country or domain, of operation/ sale. The target driver uses this code with the Country Code Selector (CCS) and worldwide roaming (WWR) flags to determine the current operating region and overlay the appropriate regulatory domain requirements on top of the target power and the band edge maximum power data. See the support bulletin *Worldwide Roaming Design Specification* for details on how this information is used.

Support of Multiple Regulatory Domains

The following information is coded in the driver to allow support of multiple regulatory domains:

- A mapping of each country code to a regulatory domain
- An association of all regulatory domains to the appropriate CTLs
- All allowed channels and the maximum legal power limits in all regulatory domains

It is important to program a comprehensive set of CTLs in the EEPROM at manufacturing calibration. Thus supporting new frequency allocations in various countries (domains), or changes in regulations in existing regulatory domains, becomes possible through a software release of the NDIS driver or AP software update with the NICs already deployed in the field.

Operating Power Algorithm

The target driver uses information stored in the board data to determine the maximum transmit power for a given channel using the algorithm:

1. Read the country code from EEPROM.
2. Obtain a list of permitted channels for this country from the driver's regulatory domain table. If the current channel does not appear in the list of permitted channels, no transmission is initiated at this channel.
3. Reconstruct the calibration table for the current channel from the calibration data sampled at the frequency piers stored in the EEPROM interpolating as appropriate under the PLA scheme. Program the calibration table into MAC/baseband processor chip.
4. Obtain the target power for each rate at the current channel from the data stored in board data. Target powers are set on a per-chain basis.
5. The driver determines the CTL for this country code and retrieves data for this CTL from board data. If the current channel is determined to be a band edge in this CTL, obtain band edge maximum power at this channel based on the number of radios transmitting. CTL values are measured and set on a per-chain basis.
6. The driver determines the current channel local regulatory power limit as well as any user configured or outside power limits. Software derived regulatory maximum and power limit values are often set in total power and are decreased when applying to multiple chains operating at once. Often –3 dB for two radios and –4.5 dB for three radios though this delta can bet set by the EEPROM.
7. Compute the minimum of the target power, band edge max power, and local regulatory power limit at the current channel values for each rate and program the max power for all supported rates into the MAC/ baseband processor chip.

Switch Table Operation

This section describes the switch table used by the baseband to control external and radio control signals as well as AGC receive attenuation depending on the state the device has entered (see Table 1-5).

The switching table is broken into two tables for AR92xx. The first provides chain specific AGC control to attenuate large input signals. The second allows control of external signals for changing between transmit, idle, receive and blueTooth coexistence, but is not chain specific. Both tables drive external pins and the analog internal LNA enable. The tables can be programmed in whatever manner the solution requires.

Table 1-5 shows valid states for the device that select an output set. Receive attenuation steps 4, 5, and 6 are only used by the chain-specific switch table.

Table 1-5. **Switch Table Operation**

Setting	Bits	State	Definition
EEPROM word for the chain-specific table is defined as:	1:0	table_idle	Idle state. table_idle [1:0]
	3:2	table_transmit	Transmit state. table_transmit [3:2]
	5:4	table_receive	Receive state. table_receive [5:4]
	7:6	table_atten1	Receive with first attenuation addition. table_atten1 [7:6]
	9:8	table_atten1&2	Receive with both attenuation additions. table_atten1&2 [9:8]
	11:10	table_blueTooth	BlueTooth coexistence state table_blueTooth [11:10]
	This parameter word is repeated once for each chain		
The EEPROM word for the common table is defined as:	3:0	table_com_idle	table_com_idle [3:0] {sw_com[3, 2, 1, 0]} when idle
	7:4	table_com_transmit	table_com_transmit [7:4] {sw_com[3, 2, 1, 0]} when tx ant
	11:8	table_com_receive	table_com_receive [11:8] {sw_com[3, 2, 1, 0]} when rx ant
	15:12	table_com_blueTooth	table_com_blueTooth [15:12] {sw_com[3, 2, 1, 0]} when BlueTooth

These switch common lines are connected to any external components that need to be flipped between state transitions: LNAs, PAs, and Tx/Rx switches. As there are both 5 GHz and 2 GHz copies of both of these tables, lines may also be used for 2 GHz to 5 GHz switching on dual-band products. The table should be defined based on the board layout needs of polarity and external component switching.

The large signal receive has two attenuation stages that require two additional sets of parameters. Both the atten 1 and atten 2 stage require setting EEPROM parameters to specify when the large signal attenuation should be activated, which acts as a hysteresis, and how much attenuation is gained by the attenuation stage (see Table 1-6).

Table 1-6. **Attenuation Steps**

Step	Definition
Atten1	First attenuation step. The DB attenuation AGC can expect when using atten 1 must be written to the BswAtten parameter. The margin/hysteresis for AGC to use for this stage must be written to BswMargin.
Atten2	Second attenuation step. The DB attenuation AGC can expect when using atten 2 must be written to the txrxatten field. The margin/hysteresis for AGC to use for this stage must be written to the rxTxmargin field.

Open-Loop Power Control

To accommodate the data gathered during calibration on a board using open-loop power control, a union has been added to the struct AR5416_EEPROM (see "EEPROM Board Data Structure File" on page A-1). This union also accommodates the data for boards using closed-loop power control. To remain backwards compatible, the footprint in the EEPROM remains the same for either power control scheme.

2

Board Data

This chapter describes the details of the board data configuration for AR92xx solutions.

NOTE: This document uses the term AR92xx to refer to the AR5008, AR913x, AR916x, AR922x, and AR928x family of products.

EEPROM version 14.8 (supported by 0.5 ART versions) supports the AR5008, AR913x and AR916x chipsets. EEPROM version 14.9 (supported by 0.6 ART versions) supports AR922x and AR928x.

Board Data Locations Description

The first 512 bytes of the EEPROM space hold different information depending on whether the device has a hardware EEPROM connected or maps the EEPROM information to a flash device. Beyond the EEPROM initialization space there are three types of information stored on the board data: generic, solution-specific, and device-specific. Groups of various board data locations are described and appropriately categorized in this section. The corresponding category of each parameter displayed in Figure 2-1 is color-coded to the type of information contained in the parameter.

NOTE: Board data locations are presented with byte- and EEPROM-location ranges. Although EEPROMs may be written in 16-bit chunks, the data should be stored and retrieved in the same manner so the EEPROM calibration structure can be block-mapped over the entire data contents.

- Board Layout
  The board layout displays the data in 16-bit offset groups for the base and modal parameters (so the EEPROM location can be identified) and then in 32-bit word chunks for the calibration tables.
- 16-Bit Offset Locations
  The 16-bit offset locations are the ones used by ART to selectively change some parameters (such as MAC address or regulatory, which will require a flipping of XORed checksum bits to match the bit changes to the parameters).

EEPROM Initialization Information

The initial 512 bytes of EEPROM space (either starting at location 0 of a hardware EEPROM or relative location 0 of a flash sector allocated for EEPROM storage) contain non WLAN calibration initialization for the board. Table 2-1 shows the values contained if a physical EEPROM is present (such as on Mini PCI, CardBus, and PCI Express devices).

Table 2-1. **EEPROM Values**

Value	Definition
Magic Half Word	A 16-bit magic half word, 0xA55A, that indicates that the EEPROM has been programmed at 16-bit offset 0.
EEPROM Read/ Write Mask	A 16-bit EEPROM read/write mask that can protect the EEPROM contents at 16-bit offset 1.
Location Link	A 16-bit location link to the beginning of board setup tuples at 16-bit offset 2. This value is always 0x0003 to point to 16-bit offset 3.
Offset 3	The 16-bit offset 3, begins groups of 48-bit board setup tuples containing a 16-bit register location followed by 32 bits of register data (least significant 2 bytes first). The first address of 0xFFFF ends the auto initialization.

Upon cold reset, hardware parses through these board setup tuples and programs hardware registers to initialize register space including: PCI configuration space setup, PCI Express configuration setup, LED initialization, and possibly sleep or timer registers that require setup before the device is probed by the attached bus. The 16-bit offset specified in the triple tuple should contain the full register offset. Table 2-2 specifies the offsets that should be used to access PCI or PCI Express configuration registers.

Table 2-2. **PCI Configuration Address Mapping**

Register Group	Offset
PCI Configuration	0x6000
PCI Express Configuration	0x5000

If a physical EEPROM is not present (such as for an integrated access point), the device responds to bus probing with default hardware deviceID and subvendorDeviceID information. In this case, the initial 512 byte area can store out-of-band information such as LAN MAC addresses, subVendorDeviceIDs, or software-required version/revision information.

Board Data Generic Information

Generic information, such as length, version, and checksum constitutes the bulk of this section. Information stored in this section is generic for each design and does not need to be measured for each design. The checksum is calculated once all other board data is known and starts at the beginning of this section (it does not cover the first 512 bytes of space).

The beginning of the board data contains a simple XOR checksum that covers the EEPROM starting at offset 0x100 over the number of locations specified by the length field. It checksums across this range and produces a value of 0xFFFF or the board data is not loaded and device initialization fails.

The version field follows the checksum and allows software to identify the board data as AR92xx-specific (major revision set to 0xE). The minor version number provides an enhancement-oriented nature to the board data so new fields can be added or field meanings enhanced by newer software. Some minor version introduce incompatible changes with previous revisions and in these cases the minor revision may also prevent EEPROM attach.

Board Data Solution-Specific Information

This information is typically obtained through a pilot run on a statistical ensemble of devices of this solution type. All manufactured devices of this design are expected to result in an optimum level of performance upon use of these settings by the software. These settings are not individually measured or calibrated for each device.

In this revision of the board data layout, space has been allocated to support operation in 802.11a, 802.11b, 802.11g, and 802.11n (5 GHz and 2 GHz) modes. Care has been taken to allocate sufficient space for all calibration parameters based on Atheros' history with a wide variety of customer solutions.

These values are conveyed by the Atheros partners to the manufacturing test flow by the **calSetup.txt** file to be stored onto the EEPROM. (Refer to the *AR5008 Sample Manufacturing Test Flow* document for more information.)

Board Data Device-Specific Information

The device-specific information accounts for all variances across boards due to component differences to produce solutions that maintain similar performance. The principle device-specific information is the "calibration" curves across frequency that describe power levels seen at the power detector. These levels take into account the gain variance across all the gain stages and provide reference level calibration of the transmit power engine to reach exacting power levels.

NOTE: Although most of the specific parameters in the board data are fixed across a solution, the board data device-specific information accounts for a small part of the board data. The entire board is calibrated and the entire EEPROM chunk written once.

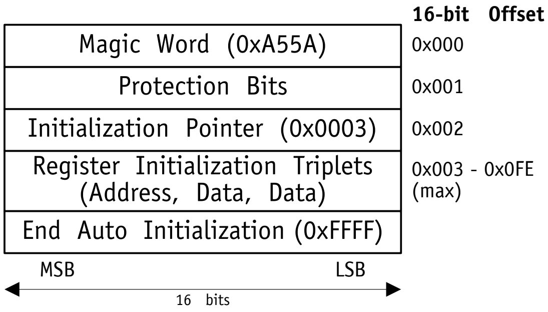
Board Data Layout

Table 2-3 shows the board data categories.

Table 2-3. **Board Data Categories**

Byte Offset	Bytes	EEPROM Locations	Category	Figure	Table
0x000–0x1FF	512	0x000–0x0FF	EEPROM Initialization	Figure 2-1	—
0x200–0x23F	64	0x100–0x11F	Base EEPROM Header	Figure 2-2	Table 2-6
0x240–0x27F	64	0x120–0x13F	Customer Data	—	Table 2-7
0x280–0x2E7	104	0x140–0x0x173	Modal EEPROM Header 5 GHz	Figure 2-3	Table 2-8
0x2E8–0x34F	104	0x174–0x1A7	Modal EEPROM Header 2 GHz	Figure 2-3	Table 2-8
0x350–0x35B	12	0x1A8–0x1AD	Power Calibration Channels	Figure 2-4	See "Power Detector Calibration" on page 1-4
0x35C–0x71B	960	0x1AE–0x38D	Power Calibration Data 5 GHz	Figure 2-5	
0x71C–0x8FB	480	0x38E–0x47D	Power Calibration Data 2 GHz	Figure 2-5	
0x8FC–0x923	40	0x47E–0x491	Target Powers 802.11a OFDM	Figure 2-6	See "Target Power" on page 1-6
0x924–0x96B	72	0x492–0x4B5	Target Powers 5 GHz 802.11n HT20 OFDM	Figure 2-7	
0x96C–0x9B3	72	0x4B6–0x4D9	Target Powers 5 GHz 802.11n HT40 OFDM	Figure 2-8	
0x9B4–0x9C2	15	0x4DA–0x4E1	Target Powers 802.11b/g CCK	Figure 2-9	
0x9C3–0x9D6	20	0x4E1–0x4EB	Target Powers 802.11g OFDM	Figure 2-10	
0x9D7–0x9FA	36	0x4EB–0x4FD	Target Powers 2 GHz 802.11n HT20 OFDM	Figure 2-11	
0x9FB–0xA1E	36	0x4FD–0x50F	Target Powers 2 GHz 802.11n HT40 OFDM	Figure 2-12	
0xA1F–0xA36	24	0x50F–0x51B	CTL Indexes	Figure 2-13	Table 2-9
0xA37–0xEB6	1152	0x51B–0x75B	CTL Data	Figure 2-14	Table 2-9
0xEB7	1	0x75B	1-Byte Pad to End on Word Boundary	—	—

Figure 2-1 shows the EEPROM initialization board data details.



	16-bit Offset
Magic Word (0xA55A)	0x000
Protection Bits	0x001
Initialization Pointer (0x0003)	0x002
Register Initialization Triplets (Address, Data, Data)	0x003 - 0x0FE (max)
End Auto Initialization (0xFFFF)	

Figure 2-1. **EEPROM Initialization**

Table 2-4 lists the location in EEPROM of the initialization triplets of the common PCI and PCI Express configuration registers that typically get initialized. Note that both PCI and PCI Express configuration registers are written to each card type regardless of whether the card is PCI or PCI Express since the hardware will ignore the appropriate registers.

Table 2-4. **Common Register Initialization Triplets**

EEPROM Offset	Description	
0x0003	Contains 0x6000	The address for PCI vendor and DeviceID register
0x0004	PCI VendorID	Fixed at 0x168C for Atheros cards
0x0005	PCI DeviceID	Fixed at 0x0023 for AR5008 PCI cards
0x0006	Contains 0x6008	Address for the PCI class code register
0x0007	Contains 0x0001	Revision ID and the LSB of PCI class code
0x0008	Contains 0x0280	2 MSB of PCI class code
0x0009	Contains 0x602C	Address for the PCI subsystem/subvendor register
0x000A	PCI SubvendorID	Configurable through calsetup.txt
0x000B	PCI SubsystemID	Varies by card type, set in .eep file
0x000C	Contains 0x5000	Address for PCI Express vendor and DeviceID register
0x000D	PCI Express VendorID	Fixed at 0x168C for Atheros cards
0x000E	PCI Express DeviceID	Fixed at 0x0024 for AR5008 PCI Express cards
0x000F	Contains 0x5008	The address for the PCI Express class code register
0x0010	Contains 0x0001	The revision ID and the LSB of PCI Express class code
0x0011	Contains 0x0280	2 MSB of PCI class code
0x0012	Contains 0x502C	Address for the PCI Express subsystem/subvendor register
0x0013	PCI Express SubvendorID	Configurable through calsetup.txt
0x0014	PCI Express SubsystemID	Varies by card type, set in .eep file
0x0015 – 0xFE	Contains vary depending on supported card features, initialization section is ended with 0xFFFF. See Table 2-6 for further details on contents	

Beyond the regular PCI/PCI Express registers listed in Table 2-4, the PCI Express has additional configuration registers that will be programmed into the EEPROM depending on which flags are set in the eep files. Table 2-5 describes these registers in the order in which they will appear in the EEPROM, however, absolute address values have not all been provided because many of the sections are optional. The address will vary depending on which flags have been enabled in the eep files.

Table 2-5. **Additional PCI Express Configuration Register Information**

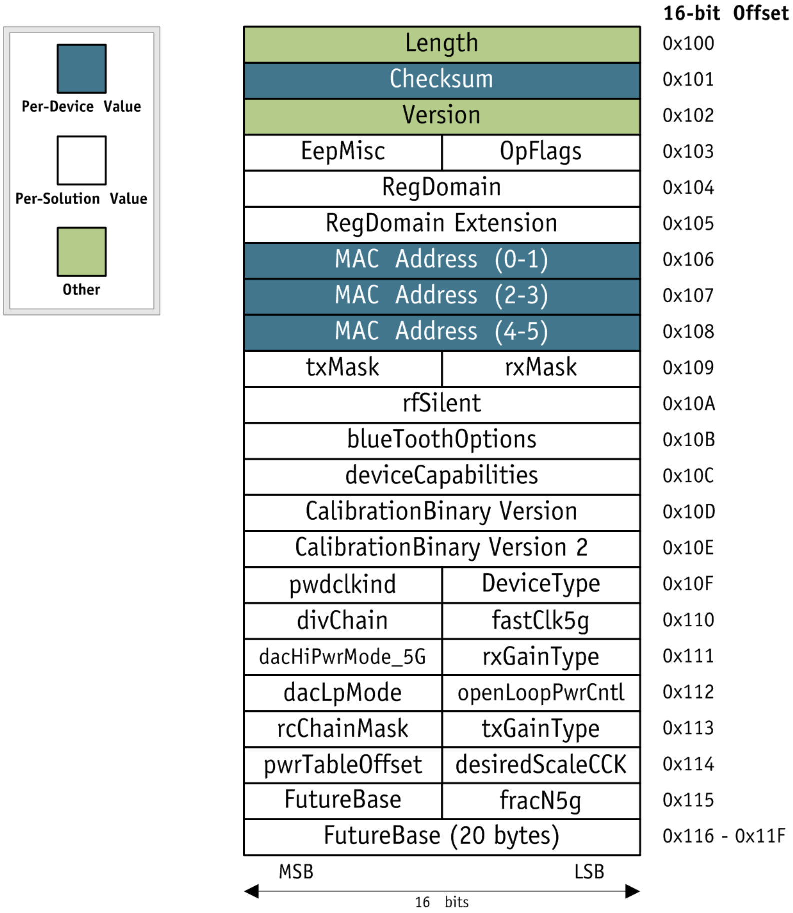
Set Location[1]	Value Written	eep File Flag	Description
0	0x5064	ASPM_LATENCY = 1 (Note: this section is not present if ASPM_LATENCY = 0)	Configure ASPM latency
1	0x0CC0		
2	0x0504		
0	0x570C	LOs_BYPASS = 0	For AR5418/AR9280/AR9281/ AR9283
1	0x3F01		
2	0x0F00		
0	0x506C	ASPM_SUPPORT = 0	ASPM support level is L0 (chip default is L0s/L1)
1	0x3011		
2	0x0003		
0	0x506C	ASPM_SUPPORT = 1	ASPM support level is L0s (chip default is L0s/L1)
1	0x3411		
2	0x0003		
0	0x506C	ASPM_SUPPORT = 2	ASPM support level is L1 (chip default is L0s/L1)
1	0x3811		
2	0x0003		
0	0x506C	ASPM_SUPPORT = 3	ASPM support level is L0s/L1 (chip default is L0s/L1)
1	0x3C11		
2	0x0003		
0	0x4084	RF_SILENT = 1 (Note: this section is not present if RF_SILENT = 0)	Set the GPIOs polarity
1	GPIO Polarity (bit 1)		
2	0x0000		
3	0x405C		Configure the GPIO being used for RF silent mode. GPIO information is provided in **atheros-eep.txt** location 0xF
4	GPIO SeleCtion-SpeCifiC		
5	0x0000		
6	0x404C		
7	0x0000		
8	0x0000		
9	0x4054		
10	0x8000		
11	0x0000		
0	0x4004	Regardless of EEP settings	Needed by some systems
1	0x073B		
2	0x0040		
0	0x4074	Regardless of EEP settings	PCI SIG test
1	0x0003		
2	0x0000		
0	0x4000	Regardless of EEP settings	PM capabilities register address
1	Default Read From Chip	Regardless of EEP settings	Lower 16 bits of the PM capabilities register as read from the chip
2	0xFFC2	ENABLE_WAKE_ON_WLAN = 1	supports all D0->D3 modes
2	0x01C2	ENABLE_WAKE_ON_WLAN = 0	Disable PME
0	0x4014	ENABLE_WAKE_ON_WLAN = 1	Let PM capabilities write take effect
1	0x0400		
2	0x3A00		
0	0x6034	Regardless of EEP Setting	Set capabilities pointer
1	0x0044		
2	0x0000		

[1] Absolute address values are no longer provided since these sections are optional.

For designs that support USB interfaces using the AR7010 platform, the EEPROM needs to contain the USB-related information for the enumeration of the USB device. This information needs to include the device's description data to the host device. After the information listed in Table 2-6, the AR7010 USB information starts at byte 256 of the EEPROM and contains 256 bytes in the EEPROM Initialization area.

The USB information defines the USB device description and string descriptors permitted for customization. It contains the USB VendID and ProductID, manufacturer name, product name and serial number. The locations and values in USB information in the EEPROM are listed in Appendix B.

Figure 2-2 shows the base EEPROM header board data details.



MSB	LSB	16-bit Offset
Length		0x100
Checksum		0x101
Version		0x102
EepMisc	OpFlags	0x103
RegDomain		0x104
RegDomain Extension		0x105
MAC Address (0-1)		0x106
MAC Address (2-3)		0x107
MAC Address (4-5)		0x108
txMask	rxMask	0x109
rfSilent		0x10A
blueToothOptions		0x10B
deviceCapabilities		0x10C
CalibrationBinary Version		0x10D
CalibrationBinary Version 2		0x10E
pwdclkind	DeviceType	0x10F
divChain	fastClk5g	0x110
dacHiPwrMode_5G	rxGainType	0x111
dacLpMode	openLoopPwrCntl	0x112
rcChainMask	txGainType	0x113
pwrTableOffset	desiredScaleCCK	0x114
FutureBase	fracN5g	0x115
FutureBase (20 bytes)		0x116 - 0x11F

Figure 2-2. **Base EEPROM Header**

Table 2-6 describes the fields shown in Figure 2-2.

Table 2-6. **Base EEPROM Header Parameter Descriptions**

Register	Bytes	Description	
Length	2	A length field describing the total bytes of the EEPROM structure.	
Checksum	2	A 16-bit value that is X-OR'ed with the rest of the EEPROM calibration structure starting from the Length field to the end of the structure as specified by the Length field. The checksum should cause the resulting 16-bit X-OR to result in 0xFFFF. If the checksum fails, the software should not load the device.	
Version	2	Allows the software to decipher the EEPROM contents. Any time the EEPROM layout is changed, the major and minor version combination should be used to convey that information to the driver. The 4 MSBs are the major version and the 12 LSBs are the minor version number.	
OpFlags	1	OpFlags describe the modal operational configuration of the board.	
		Bit [0]	Set if 5 GHz operation allowed
		Bit [1]	Set if 2 GHz operation allowed
		Bit [2]	Set if 5 GHz HT40 operation should be disabled
		Bit [3]	Set if 2 GHz HT40 operation should be disabled
		Bit [4]	Set if 5 GHz HT20 (all 11n rates) operation should be disabled
		Bit [5]	Set if 2 GHz HT20 (all 11n rates) operation should be disabled
EepMisc	1	A collection of EEPROM miscellaneous flags	
		Bit [0]	Only bit 0 is defined as Big Endian. This bit should be written as 1 when the structure is interpreted in big Endian byte ordering. This bit must be reviewed before any larger than byte parameters can be interpreted.
		Bit [1]	Enabled is adapter supports WOW
		Bit [7:2]	Reserved. Should be set to 0.
regDomain	2	A 14-bit code that identifies the currently selected country or domain of operation. The NIC driver or the AP software makes use of this value to determine the channels available for operation and the operating power of those channels available for operation and the operating power at those channels for all data rates (see also the support bulletin *Setup for Country or Regulatory Domain*) Most significant bit 15 is the country selector bit and identifies the type 14-bit code. Bit [14] is the world-wide roaming enable bit and signifies the card should perform 802.11d regulatory operations.	

Table 2-6. **Base EEPROM Header Parameter Descriptions (continued)**

Register	Bytes	Description	
RegDomain Extension	2	A 16-bit field partially reserved for future use. The remaining reserved bits are envisioned for use by upcoming regulatory setups, or for boards that change regulatory during their operating lifetime.	
		Bit [0]	en_fcc_mid Setting enables operation in FCC band from 5.47–5.7 GHz
		Bit [1]	en_jap_mid Setting enables operation in Japan band from 5.47–5.7 GHz
		Bit [2]	en_fcc_dfs_ht40 Setting enables operation in FCC band for HT40 support in DFS channels
		Bit [3]	en_jap_ht40 Setting enables operation in Japan band for HT40 support in 2 and 5 GHz
		Bit [4]	en_jap_dfs_ht40 Setting enables operation in Japan band for HT40 support in DFS channels
		Bit [15:5]	Reserved
MAC Address	6	The device's unique Ethernet MAC address	
rxMask	1	Provides a bit mask detailing which radios of this device are setup to receive	
		Bit [0]	Setting bit [0] corresponds to physical radio layout 0 being enabled for transmit. For all current AR92xx designs bit [0] must be set.
		Bit [2:1]	Bits [1] and [2] of the mask respectively correspond to radio layout 1 and radio layout 2
txMask	1	The Tx mask provides a bit mask detailing which radios are set up for transmit.	
		Bit [0]	Setting bit [0] corresponds to physical radio layout 0 being enabled for transmit. For all current AR92xx designs bit [0] must be set.
		Bits [2:1]	Bits [1] and [2] of the mask respectively correspond to radio layout 1 and radio layout 2
rfSilent	2	This bit is only used by a hardware switch. It is ignorned by the hardware if set to 0.	
		Bit [0]	If set to 1, a pull up resistor must be placed on the rfSilent GPIO, providing a hardware interface to an external on/off switch that allows manual termination of any RF activity.
		Bit [1]	Designates the polarity of the rfSilent GPIO switch (0 for rfSilent on low, 1 for rfSilent on high).
		Bits [4:2]	Allow the EEPROM to select which GPIO acts as the rfSilent GPIO controller
blueToothOptions	2	This field is currently reserved but will be used to set any initial states required by the WLAN's Bluetooth coexistence register set.	
deviceCapabilities	2	A reserved register for changing the device's capabilities. Should be written as 0.	
CalibrationBinary Version	4	An Atheros used location that stores the binary version and build number of the application used to calibrate this card for tracking purposes.	

Table 2-6. **Base EEPROM Header Parameter Descriptions (continued)**

Register	Bytes	Description	
DeviceType	1	Provided for definition of device type/form factor. Currently, neither hardware nor software utilize these bits. It is recommended that these bits be set for the appropriate implementation as future versions of hardware and/or software may contain device dependent options. Device Type Definition (recommended):	
		001	CardBus
		010	PCI
		011	Mini PCI
		100	Access Point
		101	PCIe_mini
		110	PCIe_express
		111	PCIe_desktop
		110–111	Reserved
pwdclkind	1	One bit, bit [0]. Value for field "an_top2_pwdclkind" under certain designs.	
fastClk5g	1	One bit, bit [0]. When set, PLL fast clock operation is enabled at 5GHz.	
divChain	1	8 bit value. Indicates which chain is used for diversity (0,1,2, etc) (MB93 specific).	
rxGainType	1	8 bit value. Indicates Rx gain table support.	
		000	23dB backoff
		001	13dB backoff
		010	Original
		011-111	Reserved
dacHiPwrMode_5G	1	One bit, bit [0]. When set, it indicates TB352 5G operation.	
openLoopPwrCntl	1	One bit, bit [0]. When set, it enables operation of open loop power control scheme.	
dacLpMode	1	One bit, bit [0]. Value for field " an_top1_daclpmode" under certain designs.	
txGainType	1	One bit, bit [0]. When set, it indicates high power tx gain table support.	
rcChainMask	1	One bit, bit [0]. When set, it indicates that the card is an HB93 1x2.	
desiredScaleCCK	1	Value for field "bb_desired_scale_cck" under certain designs.	
pwrTableOffset	1	8 bit value; offset in dB to add to the beginning of the pdadc table during calibration.	
fracN5g	1	8 bit value. When set, fracN synth mode applies to all 5G channels.	
futureBase	21	Expansion room for EEPROM base parameters. Should be written as 0.	

Table 2-7 describes the customer data field.

Table 2-7. **Customer Data Parameter Description**

Register	Bytes	Description
Customer Data	64	These 64 bytes are initially written by the manufacturing flow to contain an Atheros label, but the contents are intended for customers to write their own label or serial number for board tracking purposes.

Figure 2-3 shows the modal EEPROM header board data details.

2 GHz	16-bit Offset		5 GHz
0x174	Antenna Control Chain_0		0x140
0x175	Antenna Control 2 Chain_0		0x141
0x176	Antenna Control Chain_1		0x142
0x177	Antenna Control 2 Chain_1		0x143
0x178	Antenna Control Chain_2		0x144
0x179	Antenna Control 2 Chain_2		0x145
0x17A	Antenna Control Common		0x146
0x17B	Antenna Control Common 2		0x147
0x17C	AntGain chain_1	AntGain chain_0	0x148
0x17D	SwSettle	AntGain chain_2	0x149
0x17E	TxRxAtten chain_1	TxRxAtten chain_0	0x14A
0x17F	RxTxMagin chain_0	TxRxAtten chain_2	0x14B
0x180	RxTxMargin chain_2	RxTxMargin chain_1	0x14C
0x181	pgaDesired	adcDesired	0x14D
0x182	xlnaGain chain_1	xlnaGain chain_0	0x14E
0x183	TxEndToXpaOff	xlnaGain chain_2	0x14F
0x184	TxFrameToXpaOn	TxEndToRxOn	0x150
0x185	nfThresh chain_0	Thresh62	0x151
0x186	nfThresh chain_2	nfThresh chain_1	0x152
0x187	xpd	xpdGain	0x153
0x188	iqCalI chain_1	iqCalI chain_0	0x154
0x189	iqCalQ chain_0	iqCalI chain_2	0x155
0x18A	iqCalQ chain_2	iqCalQ chain_1	0x156
0x18B	outputBias	pdGainOverlap	0x157
0x18C	xpaBiasLevel	driverBias	0x158
0x18D	Three-chain Power Decrease	Two-chain Power Decrease	0x159
0x18E	TxFrameToPaOn	TxFrameToDataStart	0x15A
0x18F	bsw_atten chain_0	HT40PowerInc ForPdadc	0x15B
0x190	bsw_atten chain_2	bsw_atten chain_1	0x15C
0x191	bsw_margin chain_1	bsw_margin chain_0	0x15D
0x192	SwSettleHt40	bsw_margin chain_2	0x15E
0x193	xatten2Db chain_1	xatten2Db chain_0	0x15F
0x194	xatten2margin ch_0	xatten2Db chain_2	0x160
0x195	xatten2margin ch_2	xatten2margin ch_1	0x161
0x196	driverBias ch1	outputBias ch1	0x162
0x197	miscBits	lnaControl	0x163
0x198	bias_lvl1	bias_lvl_freq_range1	0x164
0x199	bias_lvl2	bias_lvl_freq_range2	0x165
0x19A	bias_lvl3	bias_lvl_freq_range3	0x166
0x19B - 0x19D	futureModal (6 bytes)		0x167 - 0x169
0x19E - 0x1A7	Spur Channel Info (x5)		0x16A - 0x173
	MSB	LSB	
	16 bits		

Figure 2-3. **Modal EEPROM Header 5 GHz/2 GHz**

Table 2-8 describes the fields shown in Figure 2-3.

Table 2-8. **Modal EEPROM Header Parameter Descriptions**

Register	Bytes	Description
Antenna Control Chain	12	Antenna control settings for operation on Antenna radio chain 0, 1, and 2. This parameter, similar to all per-chain parameters, is laid out so that index 0, 1, and 2 correspond to physical chain 0, 1, and 2, respectively. Each word is comprised of a repeated 2-bit setting controlling the output pins (SW_X1 and SW_X0) and specified for six possible transmission/reception states. Starting at the LSB of the register, these states include: idle, transmit, receive, receive with the first attenuation addition, receive with the first and second attenuation addition, and blueTooth coexistence state.
Antenna Common Control	4	Antenna Control Settings Common across chains. A 4-bit setting controlling the output of pins (SW_COM0, SW_COM1, SW_COM2, and SW_COM3) is specified for four possible transmission/reception states. Starting at the LSB of the register, these states include: idle, transmit, receive, and blueTooth coexistence state.
AntGain	3	Antenna gain. This antenna gain is added to the calibrated power by the driver to compute the final output power with respect to regulatory domain. An 8-bit signed quantity in 0.5 dB steps (e.g., +12 Antenna_Gain = +6 dB). This parameter is per-chain.
SwSettle	1	Switch settling time. Tx/Rx switch settling time can be set according to the settling time of the external switch. This switch settling time should be the largest of: the external Tx/Rx switch; or the largest settling time of any switch that is controlled by the switch table's attenuation 1 or attenuation 2 settings. The equation to calculate switch settling time register is: (switch settling time register) = (switching settling time / 25 ns) + 19
TxRxAtten (xatten1_hyst_ margin)	3	Specifies the difference in attenuation (in dB) achieved by switching to attenuation receive stage 2. This should be the attenuation for maximum input reduction provided by the internal or external device that is controlled by the second receive stage in the switch table. If the switch table does not enable any attenuation for stage 2, this value should be 0. Most AR92xx reference devices disable the internal LNA in their switch table for attenuation step 2 and hence the value in the EEPROM matches the LNA attenuation when disabled. RxTxMargin controls when this stage of attenuation is enabled. The name of the parameter has origins to the switch table on earlier devices switching the transmit/receive switch into transmit mode for further receive attenuation. This parameter is per-chain.
TxRxMargin (xatten2_hyst_ margin)	3	Margin (in dB) that controls when the final stage of attenuation (stage r1x12, or r2x12, in antenna control switch table) is enabled. A higher value for RxTxMargin means the final attenuation stage enables in at a lower input signal level to attenuate the received signal. The amount of attenuation is specified by the TxRxAtten parameter. This parameter is per-chain.
adcDesired	1	ADC desired size. Desired amplitude of signal to be presented to the analog to digital converter (ADC). This signed 8-bit value is used by the automatic gain control stage to output the appropriate signal size to make the best use of ADC range. The value specified is in 0.5 dB steps (e.g, –32 ADC_Desired_Size_11a = –16 dBm).

Table 2-8. **Modal EEPROM Header Parameter Descriptions (continued)**

<table>
<tr><th>Register</th><th>Bytes</th><th colspan="2">Description</th></tr>
<tr><td>pgaDesired</td><td>1</td><td colspan="2">PGA desired size. Desired amplitude of the output of the programmable analog gain stage presented as input to the baseband gain stage.
This signed 8-bit value is used to ensure optimal input signal for the external PA. The value specified is in 0.5 dB steps. For example, a typical value of –72 corresponds to a –36 dBm signal level.</td></tr>
<tr><td>xlnaGain</td><td>3</td><td colspan="2">XLNA Gain. Total gain provided by the LNA present on the target board.
This value is for consumption by the NDIS driver or AP software and is not programmed into any device register for operation. This parameter is per-chain.</td></tr>
<tr><td>TxEndToXpaOff</td><td>1</td><td colspan="2">Specifies the time difference from when the baseband is finished sending a frame to when the external PA switch is deactivated.
This parameter can be adjusted based on the ramp-down time of the external PA. For example, if the external PA ramp-down time is very fast, then it would be desirable to delay deactivating the external PA to ensure that the end of the frame being sent is not prematurely truncated.</td></tr>
<tr><td>TxEndToRxOn</td><td>1</td><td colspan="2">Specifies the time difference from when the baseband is finished sending a frame to when the external low noise amplifier (LNA) switch is activated.
This parameter can be adjusted based on the ramp-up time of the external LNA. For example, if the external LNA ramp-up time is slow, then it would be desirable to turn on the external LNA sooner so that the beginning of the receive frame is not missed.</td></tr>
<tr><td>TxFrameToXpaOn</td><td>1</td><td colspan="2">Specifies the time difference in 100ns increments from when the medium access control (MAC) sends the frame to when the external power amplifier switch is activated.
This parameter can be adjusted based on the ramp-up time of the external PA. For example, if the external PA ramp-up time is very fast, then it would be desirable to activate the external PA sooner so that the beginning of the frame being sent is not prematurely truncated.</td></tr>
<tr><td>Thresh62</td><td>1</td><td colspan="2">Adjusts clear channel assessment (CCA) sensitivity to meet the IEEE 802.11 specification. Section 17.3.10.5 of the IEEE 802.11 specification specifies CCA sensitivity as “A start of a valid OFDM transmission at receive level equal or greater than minimum 6 Mbps sensitivity (–82 dBm) shall cause CCA to indicate Busy with probability > 90% within 4 µs. If the preamble portion of a frame was missed, the receiver shall hold the carrier sense (CS) signal Busy for any signal 20 dB above minimum 6 Mbps sensitivity (–62 dBm)”. A lower threshold can be chosen for better performance in the presence of collisions by changing the setting in this register.</td></tr>
<tr><td>nfThresh</td><td>3</td><td colspan="2">Noise floor threshold is a signed 8-bit value in 1 dB steps (that is, a typical value of –85 Noise_Floor_Thresh = –85 dB). The noise floor threshold is written to registers and monitored in software to prevent transmission if the noise floor calibration detects a constant carrier above this threshold. This parameter is per-chain.</td></tr>
<tr><td rowspan="4">xpdGain</td><td rowspan="4">1</td><td colspan="2">The xpdGain controls the gain for the external power detector output. This field is used as an “pd_gain_mask” for up to two pd_gains for which the cal data is stored in the EEPROM. The LSB indicates whether cal data is stored for the lowest pd_gain, and MSB indicates whether was used for the highest pd_gain, pd_gains are in the order (LSB to MSB) 1/2, 1, 2 and 4</td></tr>
<tr><td>xpd_gain = 1001b</td><td>Indicates cal data for pdgain 1/2 and 4 is stored in EEPROM</td></tr>
<tr><td>xpd_gain = 1010b</td><td>Indicates cal data for pd_gain 1 and 4 is stored in EEPROM</td></tr>
<tr><td>xpd_gain = 0010b</td><td>Indicates cal data for only pd_gain of 1 is stored in EEPROM</td></tr>
</table>

Table 2-8. **Modal EEPROM Header Parameter Descriptions (continued)**

Register	Bytes	Description
xpd	1	The XPD selects between the internal or external detector for power control. XPD = 1 selects the external detector and XPD = 0 selects the internal.
iqCalI	3	I coefficient obtained from the iq_cal to correct for the iq_mismatch in the receive path. This coefficient is used to correct for the iq_mismatch and improve receive sensitivity. This parameter is per-chain.
iqCalQ	3	Q coefficient obtained from the iq_cal to correct for the iq_mismatch in the receive path. This coefficient is used to correct for the iq_mismatch and improve receive sensitivity. This parameter is per-chain.
pdGainOverlap	1	This permits a per-solution basis to decide on the overlap between each adjacent set of bias curves for the power detector.
outputBias	1	Used to set the bias current for the output stage of the internal PA. For the AR92xx this represents the output bias for chain 0.
driverBias	1	Used to set the bias current for the driver stage of the internal PA. For the AR92xx this represents the driver Bias for chain 0.
xpaBiasLevel	1	Permits a per-solution basis to change the bias level of the external PA. New for AR916x: if this field contains the value 0xff, then frequency range based xpabias levels are being used, as specified later in the header parameters.
Two-chain power decrease	1	For differing antenna setups, this unsigned value changes the amount subtracted from a total output power selection to determine the per-chain output power level when operating with two chains. Used by user power control selection whether affecting all packets or per descriptor power control.
Three-chain power decrease	1	For differing antenna setups, this unsigned value changes the amount subtracted from a total output power selection to determine the per-chain output power level when operating with three chains. Used by user power control selection whether affecting all packets or per descriptor power control.
TxFrameToDataStart	1	Tx frame to data start. Time in 100ns increments between the tx frame and the data start. The value must be ≥16 to function correctly. This timing parameter works along with txFrameToPaOn and txFrameToXpaOn to control the sequencing and power initialization when transmitting a frame.
TxFrameToPaOn	1	Time in 100-ns increments between the Tx frame and the internal PA enable. This timing parameter works along with txFrameToDataStart and txFrameToXpaOn to control the sequencing and power initialization when transmitting a frame.
HT40PowerInc ForPdadc	1	HT40 power increase for PDADC. The value in half dB that the HT40 target power must be increased to achieve the dBm desired target power while working with the same PDADC detector curve used by =<20 MHz width transmissions. The value should be measured and correlated per board design.
bsw_atten (Atten 1)	3	Specifies the difference in attenuation (in dB) achieved by switching to attenuation receive stage 1. This should be the attenuation for maximum input reduction provided by the internal or external device that is controlled by the first receive stage in the chain specific switch table. If the chain specific switch table does not enable any attenuation for stage 1, this value should be 0. Most AR92xx reference devices do not use this attenuation stage and hence these values are 0. BswMargin controls when this stage of attenuation is enabled. The name of the parameter has origins to a secondary switch (B-Switch) that was flipped for further max input attenuation. This parameter is per-chain.
bsw_margin (Margin 1)	3	Margin (in dB) that controls when the first stage of attenuation (stage r1x1, or r2x1, in antenna control switch table) is enabled. A higher value for BswMargin means the first attenuation stage enables in at a lower input signal level to attenuate the received signal. The amount of attenuation is specified by the BswAtten parameter. This parameter is per-chain.

Table 2-8. **Modal EEPROM Header Parameter Descriptions (continued)**

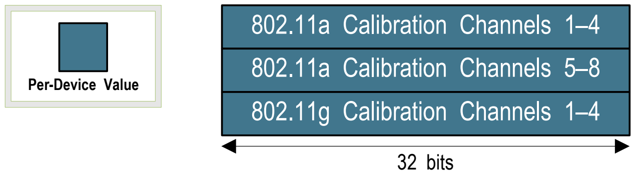
Register	Bytes	Description		
SwSettleHt40	1	This switch settling time HT40 parameter is used when operating in HT40 mode. Unlike most EEPROM parameters, the timing for switch settling in HT40 mode requires a larger constant pad. (switch settling time HT40 register) = (switching settling time / 25 ns) + 19		
xatten2Db[] [1]	3	Used for the AR92xx and represents the expected attenuation of second external switch.		
xatten2margin[] [1]	3	Used for the AR92xx and represents the margin (in dB) above the minimum on chip gain (0 dB) to switch open.		
outputBias 1 [1]	1	Used for the AR92xx this represents the output bias (bias current for the output stage of the internal PA) for chain 1.		
driverBias 1 [1]	1	Used for the AR92xx this represents the driver Bias (bias current for the driver stage of the internal PA) for chain 1.		
lnaCntl [1]	1	Bit fields for controlling the external LNA:		
		Bit [0]	Set this bit if the xLNA should be controlled with voltage mode bias, otherwise current mode is selected.	
		Bits [2:1]	If bit [0] is set to 0 for current bias, these bits control the bias current:	
			0x0	= 5 mA
			0x1	= 10 mA
			0x2	= 15 mA
			0x3	= 20 mA
		Bit [3]	If bit [0] is set to 1 for voltage bias, this bit controls the voltage bias mode:	
			0	Pull-down
			1	Pull-up
		Bit [4]	Set if GPIO9 is connect to the FEM and needs to be controlled for model switching	
		Bit [5]	local bias bit[0]: if set to “1”, the resistor rather than current source is used to set xosc bias	
		Bit [6]	bit[0]: “1” forces the xpa on, “0” forces the xpa off	
		Bit [7]	bit[0]: specific to MB93, “1”: uses ANT1 for ch0, “0”: uses ANT0 for ch0	
miscBits	1	Bits[1:0]	Tx cck scaling before DAC	
			0x0	: 100%
			0x1	: 50%
			0x2	: 25%
		Bits[4:2]	Tx clipping/scaling	
xpaBiasFreqRange[]	2 (less 2 bits)	Apply xpaBiasFreqVal for this frequency and above, up to the next xpaBiasFreqRange. A value of 0 or a freqRange less than the previous freqRange will be ignored. Frequency values are encoded as follows:		
		2 GHz	(Freq–2300)	
		5 GHz	(Freq–4800)/5	

Table 2-8. **Modal EEPROM Header Parameter Descriptions (continued)**

Register	Bytes	Description
xpaBiasFreqVal[] [2]	(2 bits)	2 bit xpaBiasLevel value that should be applied for the frequency range specified in xpaBiasFreqRange
futureModal	6	Stores noise floor cal results as follows: ch0,flo; ch0, fmi; ch0, fhi ch1,flo; ch1,fmi, ch1, fhi

[1] These fields are used only by the AR92xx and are present in EEPROM version 14.9 and above
[2] Currently this option is not needed for AR92xx and is not available in version 14.9 EEPROM

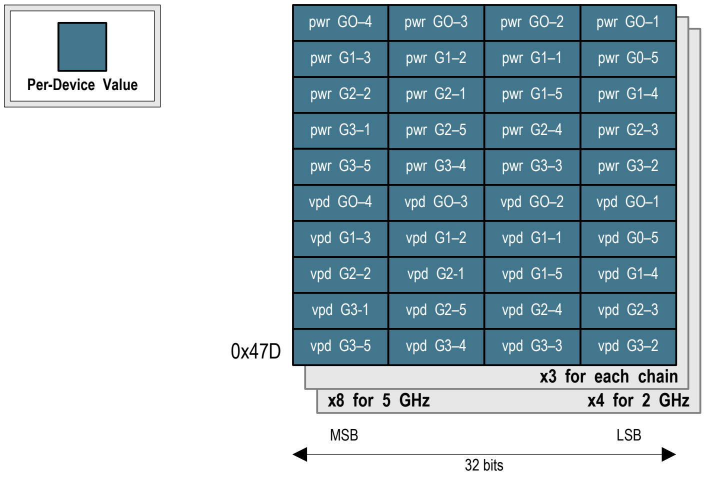
Figure 2-4 shows the power calibration channels board data details (at EEPROM locations: 0x1A8–0x1AD).



802.11a Calibration Channels 1–4
802.11a Calibration Channels 5–8
802.11g Calibration Channels 1–4

Figure 2-4. **Power Calibration Channels**

Figure 2-5 shows the power calibration data board data details (at EEPROM locations: 5 GHz = 0x1AE–0x38D and 2 GHz = 0x38E–0x47D).



pwr GO–4	pwr GO–3	pwr GO–2	pwr GO–1
pwr G1–3	pwr G1–2	pwr G1–1	pwr G0–5
pwr G2–2	pwr G2–1	pwr G1–5	pwr G1–4
pwr G3–1	pwr G2–5	pwr G2–4	pwr G2–3
pwr G3–5	pwr G3–4	pwr G3–3	pwr G3–2
vpd GO–4	vpd GO–3	vpd GO–2	vpd GO–1
vpd G1–3	vpd G1–2	vpd G1–1	vpd G0–5
vpd G2–2	vpd G2-1	vpd G1–5	vpd G1–4
vpd G3-1	vpd G2–5	vpd G2–4	vpd G2–3
vpd G3–5	vpd G3–4	vpd G3–3	vpd G3–2

Figure 2-5. **Power Calibration Data 5 GHz/2 GHz**

Figure 2-6 shows the target powers 802.11a OFDM board data details (at EEPROM locations: 0x47E–0x491).

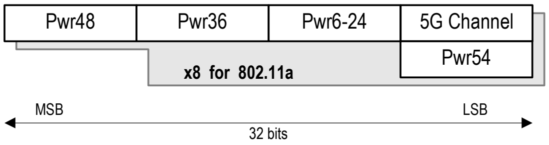


Figure 2-6. **Target Powers 802.11a OFDM**

Figure 2-7 details the target powers 5 GHz 802.11n HT20 OFDM board data (at EEPROM location: 0x492–0x4B5).

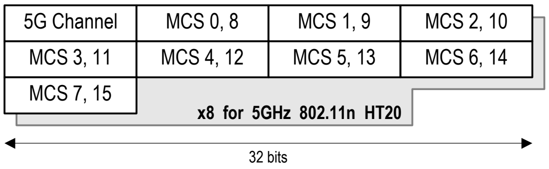


Figure 2-7. **Target Powers 5 GHz 802.11n HT20 OFDM**

Figure 2-8 details the target powers 5 GHz 802.11n HT40 OFDM board data (at EEPROM location: 0x4B6–0x4D9).

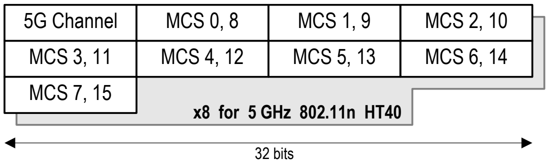


Figure 2-8. **Target Powers 5 GHz 802.11n HT40 OFDM**

Figure 2-9 details the target powers 802.11b/g CCK board data (at EEPROM location: 0x4DA–0x4E1).

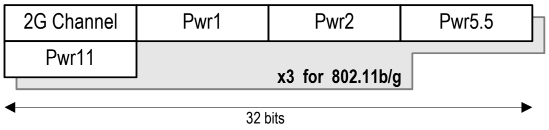


Figure 2-9. **Target Powers 802.11b/g CCK**

Figure 2-10 details the target powers 802.11g OFDM board data (at EEPROM location: 0x4E1–0x4EB).

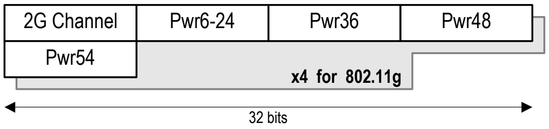
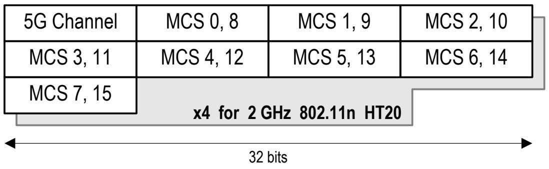


Figure 2-10. **Target Powers 802.11g OFDM**

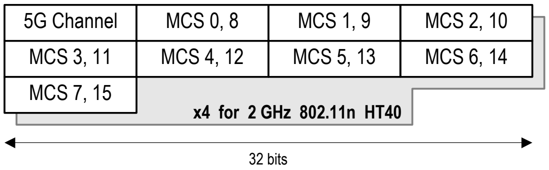
Figure 2-11 details the target powers2 GHz 802.11n HT20 OFDM board data (at EEPROM location: 0x4EB–0x4FD).



5G Channel	MCS 0, 8	MCS 1, 9	MCS 2, 10
MCS 3, 11	MCS 4, 12	MCS 5, 13	MCS 6, 14
MCS 7, 15			

Figure 2-11. **Target Powers 2 GHz 802.11n HT20 OFDM**

Figure 2-12 details the target powers 2 GHz 802.11n HT40 OFDM board data (at EEPROM location: 0x4FD–0x50F).



5G Channel	MCS 0, 8	MCS 1, 9	MCS 2, 10
MCS 3, 11	MCS 4, 12	MCS 5, 13	MCS 6, 14
MCS 7, 15			

Figure 2-12. **Target Powers 2 GHz 802.11n HT40 OFDM**

Figure 2-13 details the CTL indexes board data (at EEPROM location: 0x50F–0x51B).

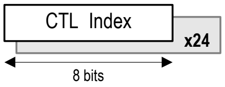
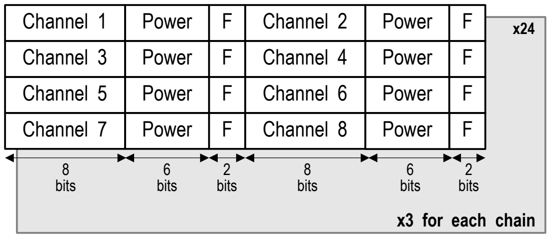


Figure 2-13. **CTL Indexes**

Figure 2-14 details the CTL data board data (at EEPROM location: 0x51B–0x75B).



Channel 1	Power	F	Channel 2	Power	F
Channel 3	Power	F	Channel 4	Power	F
Channel 5	Power	F	Channel 6	Power	F
Channel 7	Power	F	Channel 8	Power	F
8 bits	6 bits	2 bits	8 bits	6 bits	2 bits

Figure 2-14. **CTL Data**

Table 2-9 describes the CTL fields.

Table 2-9. **Board Data Parameter Descriptions**

Register	Bytes	Description	
CTL 1 … CTL 24	8	CTLs Hex codes (for that CTLs that this card is calibrated for) The CTL that the driver determines the current country code index belongs to is matched against these hex codes to figure out the location of correct data to retrieve from the EEPROM. The lower four bits (bits [3:0]) of the hex code identify which operating mode (802.11a/802.11b/802.11g) the CTL pertains to:	
		0	802.11a mode
		1	802.11b mode (CCK 11g)
		2	802.11g mode
		5	802.11n 2 GHz HT20
		6	802.11n 5 GHz HT20
		7	802.11n 2 GHz HT40
		8	802.11n 5 GHz HT40
channel 1...8	1	Band edge channel, encoded using the formula from Table 1-2 on page 1-6	
power	6 bits	Band edge power specified in half dB	
F	2 bits	Band edge flag to specify edge (1) or inband channel (0)	

A

EEPROM Board Data Structure File

This appendix shows the board data structure file.

Format Description

```
/*
 * Copyright (c) 2002-2009 Atheros Communications, Inc.
 * All rights reserved.
 *
 * AR92xx Board Data Structure
 * This structure should be precompiler packed
 */

#define AR5416_EEP_VER                 0xE
#define AR5416_EEP_START_LOC           256
#define AR5416_NUM_5G_CAL_PIERS          8
#define AR5416_NUM_2G_CAL_PIERS         4
#define AR5416_NUM_5G_20_TARGET_POWERS   8
#define AR5416_NUM_5G_40_TARGET_POWERS   8
#define AR5416_NUM_2G_CCK_TARGET_POWERS  3
#define AR5416_NUM_2G_20_TARGET_POWERS   4
#define AR5416_NUM_2G_40_TARGET_POWERS   4
#define AR5416_NUM_CTLS                 24
#define AR5416_NUM_BAND_EDGES            8
#define AR5416_NUM_PD_GAINS              4
#define AR5416_PD_GAIN_ICEPTS            5
#define AR5416_EEPROM_MODAL_SPURS        5
#define AR5416_MAX_CHAINS                3
```

```
typedef struct BaseEepHeader {
    A_UINT16  length;
    A_UINT16  checksum;
    A_UINT16  version;
    EEP_FLAGS opCapFlags;
    A_UINT16  regDmn[2];
    A_UINT8   macAddr[6];
    A_UINT8   rxMask;
    A_UINT8   txMask;
    A_UINT16  rfSilent;
    A_UINT16  blueToothOptions;
    A_UINT16  deviceCap;
    A_UINT32  binBuildNumber;
    A_UINT8   deviceType;
    A_UINT8   pwdclkind;
    A_UINT8   fastClk5g;
    A_UINT8   divChain;
    A_UINT8   rxGainType;
    A_UINT8   dacHiPwrMode_5G;
    A_UINT8   openLoopPwrCntl;
    A_UINT8   dacLpMode;
    A_UINT8   txGainType;
    A_UINT8   rcChainMask;
    A_UINT8   desiredScaleCCK;
    A_UINT8   pwrTableOffset;
    A_UINT8   fragN5G;
    A_UINT8   futureBase[21];
} __ATTRIB_PACK BASE_EEP_HEADER; // 64 B

typedef struct spurChanStruct {
    A_UINT16   spurChan;
    A_UINT8    spurRangeLow;
    A_UINT8    spurRangeHigh;
} SPUR_CHAN;

typedef struct calDataPerFreqOpLoop {
    A_UINT8 pwrPdg[2][5]; /* power measurement                  */
    A_UINT8 vpdPdg[2][5]; /* pdadc voltage at power measurement */
    A_UINT8 pcdac[2][5];  /* pcdac used for power measurement   */
    A_UINT8 empty[2][5];  /* future use */
} CAL_DATA_PER_FREQ_OP_LOOP;

typedef struct calDataPerFreq {
    A_UINT8    pwrPdg[AR5416_NUM_PD_GAINS][AR5416_PD_GAIN_ICEPTS];
    A_UINT8    vpdPdg[AR5416_NUM_PD_GAINS][AR5416_PD_GAIN_ICEPTS];
} CAL_DATA_PER_FREQ;

typedef union calDataPerFreq_u {
    struct calDataPerFreqOpLoop calDataOpen;
    struct calDataPerFreq       calDataClose;
} CAL_DATA_PER_FREQ_U;

typedef struct CalTargetPowerLegacy {
    A_UINT8    bChannel;
    A_UINT8    tPow2x[4];
} CAL_TARGET_POWER_LEG;
```

```
                  typedef struct CalTargetPowerHt {
                      A_UINT8    bChannel;
                      A_UINT8    tPow2x[8];
                  } CAL_TARGET_POWER_HT;

                  #ifdef BIG_ENDIAN
                  typedef struct CalCtlEdges {
                      A_UINT8    bChannel;
                      A_UINT8    flag   :2,
                                 tPower :6;
                  } CAL_CTL_EDGES;
                  #else
                  typedef struct CalCtlEdges {
                      A_UINT8    bChannel;
                      A_UINT8    tPower :6;
                                 flag   :2,
                  } CAL_CTL_EDGES;
                  #endif

                  typedef struct CalCtlData {
                      CAL_CTL_EDGES ctlEdges[AR5416_MAX_CHAINS][AR5416_NUM_BAND_EDGES];
                  } CAL_CTL_DATA;


typedef struct ar5416Eeprom {
        BASE_EEP_HEADER      baseEepHeader;
        A_UINT8              custData[64];
        MODAL_EEP_HEADER     modalHeader[2];
        A_UINT8              calFreqPier5G[AR5416_NUM_5G_CAL_PIERS];
        A_UINT8              calFreqPier2G[AR5416_NUM_2G_CAL_PIERS];
        CAL_DATA_PER_FREQ_U  calPierData5G[AR5416_MAX_CHAINS][AR5416_NUM_5G_CAL_PIERS];
        CAL_DATA_PER_FREQ_U  calPierData2G[AR5416_MAX_CHAINS][AR5416_NUM_2G_CAL_PIERS];
        CAL_TARGET_POWER_LEG calTargetPower5G[AR5416_NUM_5G_20_TARGET_POWERS];
        CAL_TARGET_POWER_HT  calTargetPower5GHT20[AR5416_NUM_5G_20_TARGET_POWERS];
        CAL_TARGET_POWER_HT  calTargetPower5GHT40[AR5416_NUM_5G_40_TARGET_POWERS];
        CAL_TARGET_POWER_LEG calTargetPowerCck[AR5416_NUM_2G_CCK_TARGET_POWERS];
        CAL_TARGET_POWER_LEG calTargetPower2G[AR5416_NUM_2G_20_TARGET_POWERS];
        CAL_TARGET_POWER_HT  calTargetPower2GHT20[AR5416_NUM_2G_20_TARGET_POWERS];
        CAL_TARGET_POWER_HT  calTargetPower2GHT40[AR5416_NUM_2G_40_TARGET_POWERS];
        A_UINT8              ctlIndex[AR5416_NUM_CTLS];
        CAL_CTL_DATA         ctlData[AR5416_NUM_CTLS];
        A_UINT8              padding;
} AR5416_EEPROM;
```

```
typedef struct ModalEepHeader {
        A_UINT32  antCtrlChain[AR5416_MAX_CHAINS];  // 12
        A_UINT32  antCtrlCommon;                    // 4
        A_INT8    antennaGainCh[AR5416_MAX_CHAINS]; // 3
        A_UINT8   switchSettling;                   // 1
        A_UINT8   txRxAttenCh[AR5416_MAX_CHAINS];   // 3    //xatten1_hyst_margin for
                                                            AR928x (0x9848/0xa848 13:7)
        A_UINT8   rxTxMarginCh[AR5416_MAX_CHAINS];  // 3    //xatten2_hyst_margin for
                                                            AR928x (0x9848/0xa848 20:14)
        A_INT8    adcDesiredSize;                   // 1
        A_INT8    pgaDesiredSize;                   // 1
        A_UINT8   xlnaGainCh[AR5416_MAX_CHAINS];    // 3
        A_UINT8   txEndToXpaOff;                    // 1
        A_UINT8   txEndToRxOn;                      // 1
        A_UINT8   txFrameToXpaOn;                   // 1
        A_UINT8   thresh62;                         // 1
        A_INT8    noiseFloorThreshCh                // 3
                  [AR5416_MAX_CHAINS];
        A_UINT8   xpdGain;                          // 1
        A_UINT8   xpd;                              // 1
        A_INT8    iqCalICh[AR5416_MAX_CHAINS];      // 1
        A_INT8    iqCalQCh[AR5416_MAX_CHAINS];      // 1
        A_UINT8   pdGainOverlap;                    // 1
        A_UINT8   ob;                               // 1    //for AR928x this is chain0
        A_UINT8   db;                               // 1    //for AR928x this is chain0
        A_UINT8   xpaBiasLvl;                       // 1
        A_UINT8   pwrDecreaseFor2Chain;             // 1
        A_UINT8   pwrDecreaseFor3Chain;             // 1
                                                    ->48 B
        A_UINT8   txFrameToDataStart;               // 1
        A_UINT8   txFrameToPaOn;                    // 1
        A_UINT8   ht40PowerIncForPdadc;             // 1
        A_UINT8   bswAtten[AR5416_MAX_CHAINS];      // 3    //xatten1_db for AR928x
                                                            (0xa20c/b20c 5:0)
        A_UINT8   bswMargin[AR5416_MAX_CHAINS];     // 3    //xatten1_margin for AR928x
                                                            (0xa20c/b20c 16:12)
        A_UINT8   swSettleHt40;                     // 1
        A_UINT8   xatten2Db[AR5416_MAX_CHAINS];     // 3    //new for AR928x (0xa20c/
                                                            b20c 11:6)
        A_UINT8   xatten2Margin[AR5416_MAX_CHAINS]; // 3    //new for AR928x (0xa20c/
                                                            b20c 21:17)
        A_UINT8   ob_ch1;                           // 1    //ob and db become chain
                                                            specific in AR928x
        A_UINT8   db_ch1;                           // 1    //AR928x is 2-chain, only
                                                            adding 1 extra chain now
        A_UINT8   lna_cntl;                         // 8    // bit0: xlnabufmode
                                                            // bit1, bit2: xlnaisel
                                                            // bit3: xlnabufin
                                                            // bit4: femBandSelectUsed
                                                            // bit5: localbias
                                                            // bit6: force_xpaon
                                                            // bit7: useAnt1
        A_UINT8   miscBits;                         // 5    // bit0, bit1:
                                                            bb_tx_dac_scale_cck
                                                            // bit2, bit3, bit4:
                                                            bb_tx_clip
        A_UINT16  xpaBiasLvlFreq[3];                        // modal external PA bias
        A_UINT8   futureModal[6];                           // noise floor cal results
        SPUR_CHAN spurChans                         // 20 B
                  [AR5416_EEPROM_MODAL_SPURS];
} __ATTRIB_PACK MODAL_EEP_HEADER;                    // == 100 B
```

B

AR7010 USB Information in the EEPROM

This appendix shows the USB information in the EEPROM of the AR7010 platform. Table B-1 lists the USB-related data stored on the EEPROM. The AR7010 USB information consists of 256 bytes, starting at byte 256 (i.e. 0x80 base address in the half-word address mode) of a hardware EEPROM.

Table B-1 lists the locations and values of USB information located in the EEPROM for the enumeration of a USB device, which needs to provide the device description data to the host device. The USB device description and string descriptor permitted for customization are also listed in below table. The USB VendorID and ProductID are placed in the device descriptor. The string descriptors are used to display the company name, product name and serial number. The string should be in the UTF-16 unicode format.

To customize the USB Information, the Atheros ART tool provides the user interface to modify the description strings of the VID/PID, manufacturer, product name and serial number.

Table B-1. **USB Information in the AR7010 EEPROM**

EEPROM Offset	Description	
0x0080	Contains 0x4154	EEPROM Valid ID, "ATHR" in ASCII characters, used to identify whether the EEPROM contains the correct USB information for emulation.
0x0081	Contains 0x4852	
0x0082	Descriptor Type and Size	Contains 0x0112. 0x01 for the device descriptor type and 0x12 for the size of this descriptor (*hLength*)
0x0083	USB Version	Fixed at 0x0200 for the USB specification version (*bcdUSB*)
0x0084	Device class and subclass	Contains 0xFFFF, device class 0xFF (*bDeviceClass*)and device subclass 0xFF (*bDeviceSubClass*)
0x0085	EP0 Max Packet Size	Contains 0x40FF, maximum packet size of endpoint zero is 0x40 (*bMaxPacketSize0*) and device protocol is 0xFF (*bDeviceProtocol*)

Table B-1. **USB Information in the AR7010 EEPROM**

EEPROM Offset	Description	
0x0086	USB VendorID	Primary user-configured Vendor ID (*idVendor*)
0x0087	USB ProductID	Primary user-configured Vendor ID (*idProduct*)
0x0088	Device Release Number	Device Release Version. Internal use only (*bcdDevice*)
0x0089	Product and Manufacturer Index on Descriptor	Defaulted as 0x02010. 0x20 (*iProduct*) is used as the index of the string descriptor describing the product and 0x10 (*iManufacturer*) is used as the index of the string descriptor describing the manufacturer
0x008A	Serial Number Index on Descriptor	Fixed on 0x0130, the configuration number is 0x01(*bNumConfiguration*) and the index of the string descriptor describing the serial number is 0x30 (*iSerialNumber*)
0x008B	Reserved	Default as 0xFFFF
0x008C	Atheros Device ID	Contains 0x002A, for internal use only
0x008D	Atheros Specific ID	Contains 0x168C, for internal use only
0x008E	Atheros Sub System ID	Fixed on 0xB097, or 0xB093, for internal use only
0x008F	Atheros Specific ID	Contains 0x168C, for internal use only
0x0090 - 0x0091	Reserved	Default as 0xFFFF
0x0092	Language Descriptor Header	The High Byte is the string descriptor type, fixed as 0x03. The Low Byte is the size of the entire descriptor in bytes (the size of the descriptor header plus the size of the descriptor contents)
0x0093	Language Value	Contains the value for the USB string descriptor 0x00 for language. Default as 0x0409.
0x0094 - 0x0097	Reserved	Default as 0xFFFF
0x0098	Manufacturer Descriptor header	The High Byte is the string descriptor type, fixed as 0x03. The Low Byte is the size of the entire descriptor, in bytes (the size of the descriptor header plus the size of the descriptor contents).
0x0099 - 0x00A3	Manufacturer Name	User-specific string containing the Manufacturer Name stored in UTF-16 Unicode format. The maximum string length is eleven characters.
0x00A4	Product Descriptor Header	The High Byte is the string descriptor type, fixed as 0x03. The Low Byte is the size of the entire descriptor, in bytes (the size of the descriptor header plus the size of the descriptor contents).
0x00A5 - 0x00B3	Product Name	User-specific Product Name stored in UTF-16 Unicode format. The maximum string length is fifteen characters.
0x00B4	Serial Number Descriptor Header	The High Byte is the string descriptor type, fixed as 0x03. The Low Byte is the size of the descriptor in bytes (the size of the descriptor header plus the size of the descriptor contents).
0x00B5 - 0x00BB	Serial Number	User-specific Serial Number in UTF-16 Unicode format. The maximum string length is seven characters.
0x00BC - 0x00FD	Reserved	Reserved, fixed at 0xFFFF
0x00FE - 0x00FF	Reserved	Internal use only. Must not be changed.

Index

Atheros Communications, Incorporated

5480 Great America Parkway
Santa Clara, CA 95054
t: 408/773-5200
f: 408/773-9940
www.atheros.com