AR5005 Sample Manufacturing Test Flow

PRELIMINARY

Revision January 2005

ATHEROS®
COMMUNICATIONS

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: 984-00013-012

Document Conventions

Text Conventions

bold	Bold type within paragraph text indicates commands, file names, directory names, paths, output, or returned values. Example: The DK_Client package will not function unless you use the **wdreg_install** batchfile.
italic	Within commands, italics indicate a variable that the user must specify. Example: **mem_alloc** *size_in_bytes* Titles of manuals or other published documents are also set in italics.
`Courier`	The Courier font indicates output or display. Example: `Error:Unable to allocate memory for transfer!`
[ ]	Within commands, items enclosed in square brackets are optional parameters or values that the user can choose to specify or omit.
{ }	Within commands, items enclosed in braces are options from which the user must choose.
|	Within commands, the vertical bar separates options.
…	An ellipsis indicates a repetition of the preceding parameter.
>	The right angle bracket separates successive menu selections. Example: Start > Programs > DK > wdreg_install.

Notices

NOTE: This message denotes neutral or positive information that calls out important points to the text. A note provides information that may apply only in special cases.

Revision History

Revision	Description of Changes
October 2004	Initial AR5513 release. Updated Chapter 3, Setup Test Equipment, for dual chain requirements.
January 2005	Updated c source files listings for AR2414/AR2413 and AR5413/AR5414 chipsets

Contents

List of Figures

List of Tables

Preface

This document is intended to provide information about manufacturing test flow for station modules implementing the Atheros chip sets. This document describes using the Atheros Radio Test (ART) for a module manufacturing test flow.

NOTE: All information related to EEPROM for Reference Designs is based on the AR521x, AR241x and AR541x chips and, unless otherwise specified, applies to the Flash memory in AR5001 AP Access Point Reference Designs based on the AR5311, and the AR5002AP AP Reference Designs based on the AR5312 and AR2312 chip sets.

About this Document

The document consists of the following chapters and appendixes:

Chapter 1 — **Overview**—Description of Atheros provided manufacturing test library and system hardware and software requirements.

Chapter 2 — **Sample Manufacturing Test Flow**—Manufacturing test procedure descriptions and examples.

Chapter 3 — **Set Up Test Equipment**—Test equipment and software setup used for calibration of the Atheros Wireless chip sets in a CardBus, Mini PCI or PCI design.

Chapter 4 — **Manufacturing Test Program Source Code**—Description of the manufacturing test program source code.

Appendix A — **Sample File calsetup.txt**—IEEE 802.11a-related regulatory and standards measurement matrix.

Appendix B — **Sample File calTargetPower_ar5001a_cb.txt**—A sample of the file **calTargetPower_ar5005a_cb.txt**.

Appendix C — **Sample .eep File ar5005x_cb.eep**—A sample of the file **ar5005x_cb.eep**.

Audience

This document is intended for those involved with the definition, design and implementation of manufacturing test flows for modules deploying the Atheros AR5001 and AR5002 chip sets. Included is a sample Manufacturing Test Flow as well as a recommended calibration procedure. Additional regulatory and standards test requirements are provided in an appendix.

NOTE: Atheros will provide worldwide information pertaining to regulatory, standards and manufacturing test issues as that information becomes available. However, the responsibility for regulatory and standards compliance for products deploying Atheros chip sets remains exclusively the responsibility of the Atheros customers developing those products.

Additional Resources

This guide assumes that the reader has studied and is familiar with the following:

- *AR5005 EEPROM Device Configuration Guide*
- *AR5004 Atheros Radio Test Reference Guide*
- *AR5001/AR5002 Manufacturing Library Reference*

The following resources should be referenced regarding topics that are not addressed in this document:

- *AR2111 Radio-on-a-Chip for 2.4 GHz Wireless LANs data sheet*
- *AR2112 Multi-Mode, Radio-on-a-Chip for IEEE 802.11b/g Wireless LANs data sheet*
- *AR5111 Radio-on-a-Chip for 5 GHz Wireless LANs data sheet*
- *AR5112 Dual-Band, Multi-Mode, Radio-on-a-Chip for IEEE 802.11a/b/g Wireless LANs data sheet*
- *AR5211 Multiprotocol MAC/Baseband Processor for 5 GHz and 2.4 GHz Wireless LANs data sheet*
- *AR5212 Multiprotocol MAC/Baseband Processor for 5 GHz and 2.4 GHz Wireless LANs data sheet*
- *AR5213 Multiprotocol MAC/Baseband Processor for 5 GHz and 2.4 GHz Wireless LANs data sheet*
- *AR5311 MAC/Baseband Processor for 5 GHz Wireless LANs data sheet*
- *AR2312 Wireless System-on-a-Chip (WiSoC) for 2.4 GHz and 5 GHz Wireless LANs data sheet*
- *AR5312 Wireless System-on-a-Chip (WiSoC) for 2.4 GHz and 5 GHz Wireless LANs data sheet*

1
Overview

The Atheros Radio Test (ART) is a unified tool enabling Atheros partners to conveniently perform common radio evaluation and manufacturing tests. This tool performs various transmission, receive, and link tests, and allows card calibration and testing during the manufacturing flow.

NOTE: All information related to EEPROM for Reference Designs is based on the AR521x, AR241x and AR541x chips and, unless otherwise specified, applies to the Flash memory in AR5001 AP Access Point Reference Designs based on the AR5311, and the AR5002AP AP Reference Designs based on the AR5312 and AR2312 chip sets.

This document describes a sample manufacturing test flow for station (STA) and access point (AP) implementations using the Atheros AR5001, AR5002, and AR5005 chip sets. Atheros partners should develop a complete manufacturing test suite to meet their manufacturing test requirements.

MFG Test System Requirements

This section describes the equipment required in sample manufacturing test flow to capitalize on the C-based manufacturing tests and instrument control library provided with the release of ART.

- Power Meter: Agilent E4416A, HP 436A, HP 4531 or Agilent E9327A Peak and Average Power Sensor
- Attenuator: HP 8496H 110dB, HP 8495H 70dB, HP 8494H 11dB attenuators with a switch driver or HP 11713A Attenuator/Switch Driver
- Spectrum Analyzer: Agilent 4404B, Agilent 4405B or HP 8595E
- National Instruments GPIB-ENET/100
- Appropriate cables for instrument attachment
- 2 Laptop PCs with Windows 2000 OS
- A “golden unit” (GU)
- A device under test (DUT)
- ART for Windows 2000
- Atheros provided instrument library

Manufacturing Test System

Figure 1-1 provides a diagrammatic view of the manufacturing test system that the sample manufacturing test flow uses.

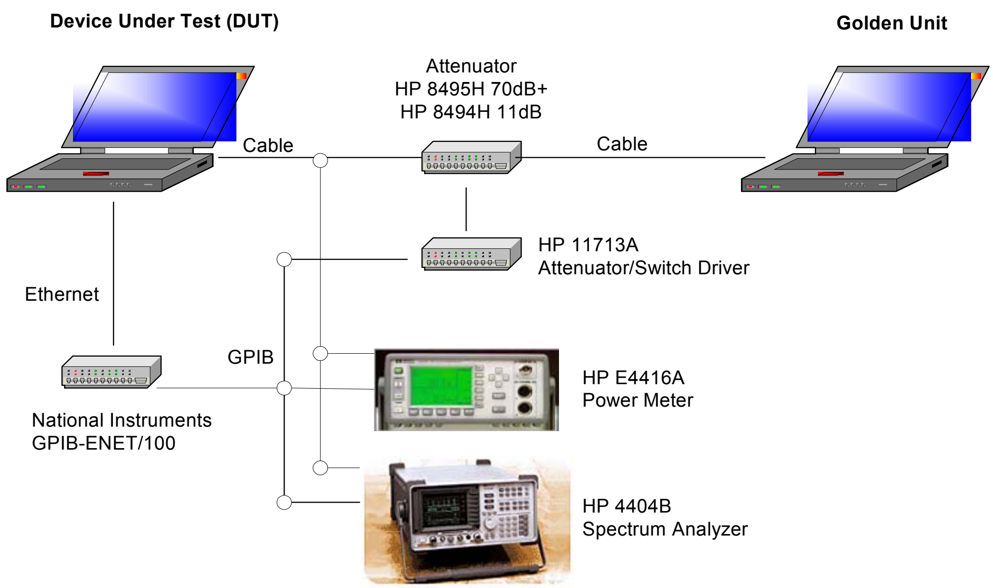

Figure 1-1. **System Requirements for Sample Manufacturing Test Flow**

Atheros customers who use different instruments in their manufacturing test suite must modify the instrument control subroutines in the C library and verify instrument readings before modifying the manufacturing test program.

Users must also gather some calibrated data from a GU and cable/splitter loss that used in the manufacturing test software before starting the manufacturing test flow. Therefore, a GU should be fully calibrated having known output power, receive sensitivity, and reference oscillator, with cable and splitter loss specifications obtained. The initial software setup uses these known data in the manufacturing test program through the file **calsetup.txt** (see Appendix A). With these known values, users can calculate the settings for the attenuator to be used for the packet error rate (PER) and receive sensitivity tests in the manufacturing test program.

2

Sample Manufacturing Test Flow

Due to the sensitive nature of RF circuit design, it is strongly recommended that each station (STA) or access point (AP) in the manufacturing process be calibrated to guarantee the performance of the product designed with the Atheros chip sets. The device under test (DUT) and golden unit (GU) must also be isolated in a shielded casing when running the manufacturing test.

The frequency response flatness of the matching balun, external PA, filters, and other analog circuits may cause transmit power variations on different operating channels. The process variation at the foundry manufacturing of the chipset may also cause slight output power variation even with the same internal register settings.

Therefore, the first step in this sample manufacturing test flow is to set up instruments used in calibrating and testing the DUT for performance and regulatory compliance.

Table 2-1 summarizes the sample manufacturing test procedures in recommended execution order.

Table 2-1. **MFG Test Flow**

Procedure	Description
Program EEPROM	Programs EEPROM with common data, such as PCI configuration data, CIS tuple, and common registers setting, but without calibration data. Most of this data is read from the file **atheros-eep.txt**. For details, see "Program EEPROM with Common Data" on page 2-5.
Temperature Margin Test	Tests the margin to support the target powers for each mode across a temperature range of 0–85 °C.
IQ_Cal	A calibration step that determines the mismatch between the I and Q receive paths. When this mismatch is corrected for, it results in an improved sensitivity for higher rates.

Table 2-1. **MFG Test Flow (continued)**

Procedure	Description
Output Power Calibration	Finds the actual power reading from Power Meter with various PCDAC register settings. Measurements are made over the entire frequency range, and the 10 frequency piers are automatically computed. Snapshot of the measured data at these frequency piers is then programmed as the calibration data into EEPROM. (Refer to the document *AR5005 EEPROM Device Configuration Guide* for details.) If all three modes of operation: 802.11a, 802.11b, and 802.11g, are supported by a card. A unique set of measurements is required for each mode.
Target Power Control Test	After calibration, measures power at the test frequencies of various rates and measures the output power to ensure that the cards can accurately control the output power at the desired target power levels. If a card supports all three modes of operation (802.11a, 802.11b, and 802.11g), a unique set of this test is performed for each mode. This test can be performed at any of the test channels listed in the file **calsetup.txt** (see "Rules for Test Channel Matrices Setup" on page 3-12).
Spectral Mask Test	Obtains the Spectrum Analyzer reading to ensure that the spectral density of the transmitted signal falls within spectral mask requirements (IEEE 802.11a section 17.3.9.2). If a card supports all three modes of operation (802.11a, 802.11b, and 802.11g), a unique set of this test is performed for each mode. This test can be performed at any of the test channels listed in the file **calsetup.txt** (see "Rules for Test Channel Matrices Setup" on page 3-12).
Occupied Bandwidth (OBW)[1] Test	Checks the Occupied Bandwidth reading from Spectrum Analyzer to make sure the TELEC-required OBW is in compliance. Valid only for 802.11a mode. This test can be performed at any of the test frequencies listed in the file **calsetup.txt** (see "Rules for Test Channel Matrices Setup" on page 3-12).
Packet Error Rate (PER) and Local clock accuracy (PPM-Pulse Per Million)[2] Test	Transmits packets from the DUT to the GU and read back the PER test result from the GU. In this test, the GU also reports the clock difference between itself and the DUT in ppm to check the local oscillator accuracy of DUT. This test can be performed at any of the test frequencies listed in the file **calsetup.txt** (see "Rules for Test Channel Matrices Setup" on page 3-12). If a card supports all three modes of operation (802.11a, 802.11b, and 802.11g), a unique set of this test is performed for each mode.

Table 2-1. **MFG Test Flow (continued)**

Procedure	Description
Receive Sensitivity Test	Transmits packets from the GU to the DUT to check sensitivity of the DUT receive path (see IEEE 802.11a section 17.3.10.1). This test can be performed at any of the test frequencies listed in the file **calsetup.txt** (see "Rules for Test Channel Matrices Setup" on page 3-12). If a card supports all three modes of operation (802.11a, 802.11b, and 802.11g), a unique set of this test is performed for each mode.
Data Integrity Test	Transmits 2400 byte packets of various data patterns and verify the data at the receiver. This test checks the on-chip memory to store the packet data. This test does not need to be repeated for all modes for the chip uses the same on-chip buffer to store the frame data in all modes. Test is performed at 5260 MHz if enabled for 802.11a mode, and at 2462 MHz if enabled for 802.11g mode.
Throughput test	Transmits packets from the DUT to the GU to measure the unicast transmit throughput. Throughput measured is closer to the UDP throughput and does not represent the TCP throughput. This test can be performed at any of the test frequencies listed in the file **calsetup.txt** (see "Rules for Test Channel Matrices Setup" on page 3-12). A threshold can be specified in the file **calsetup.txt** for a pass/fail criterion. If a card supports all three modes of operation (802.11a, 802.11b, and 802.11g), a unique set of this test is performed for each mode.
Write MAC ID	Assigns the DUT a unique WLAN MAC ID and programs it into the EEPROM after the DUT passes all tests listed in this table. Even if one or more of the above tests fail, this test prompts the operator to indicate whether they wish to proceed with assigning a MAC ID. This provision is made to accommodate marginal failures. If the DUT is an AP, an Ethernet MAC ID is also written for each Ethernet port present (specified under NUM_ETHERNET_PORTS in the file **calsetup.txt**).

1 Occupied Bandwidth measurements are mandated by Japanese Regulatory requirements.

2 This test procedure requires the GU to have a calibrated reference crystal as described in the *AR5002*, *AR5004*, and *AR5005 STA Reference Guide*s.

Manufacturing Test Flow

Figure 2-1 depicts the manufacturing test program flow for the DUT and GU.

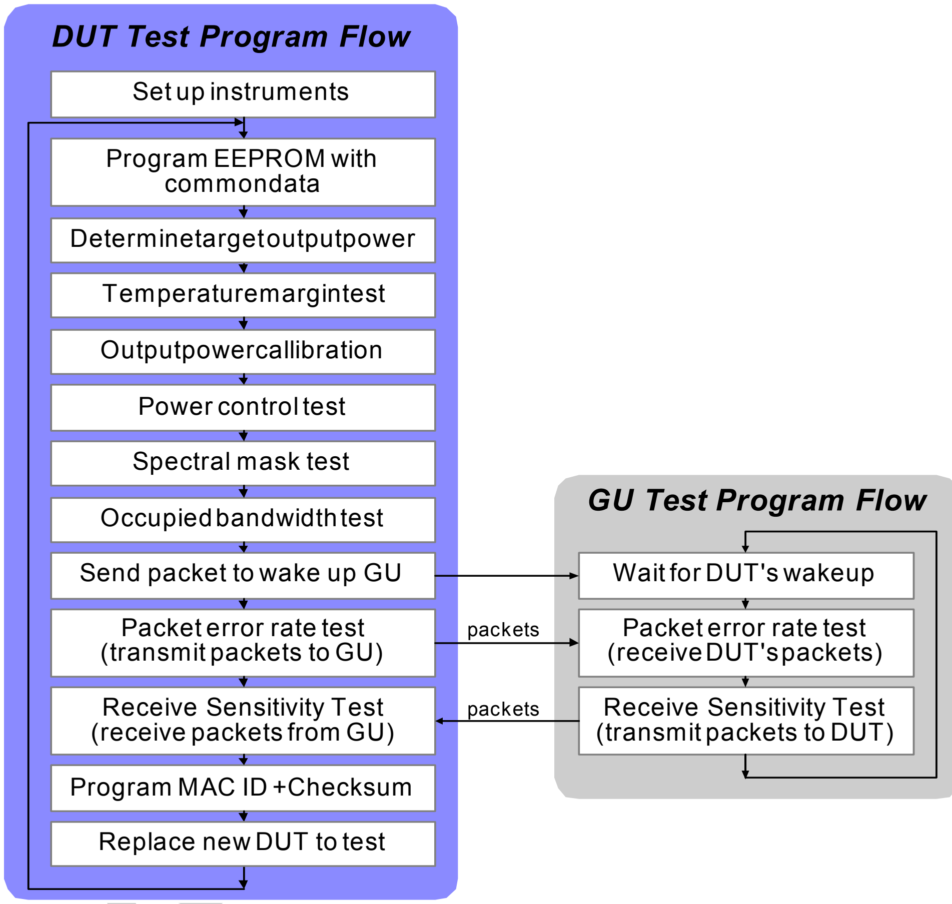

Figure 2-1. **Block Diagram of MFG Test Flow**

This test sequence is performed for each of the three modes of operation: 802.11a, 802.11b, and 802.11g. The only exception is the occupied bandwidth test, which is performed only for the 802.11a mode. The manufacturing flow is modular and provides an independent flag to control each test for each mode.

Set Up Instruments

During the manufacturing test, the DUT test program controls all instruments through the Ethernet/GPIB interface. Customers must modify the instrument library to use instruments it does not support. The current release supports:

Spectrum Analyzer:	HP E4404B, HP 8595E
Power Meter:	HP 436A, HP E4416A, HP 4531
Attenuator Control Box:	HP 11713A

Program EEPROM with Common Data

Along with PCI configuration and CIS tuples data, EEPROM defines many HW-dependent registers in hexadecimal locations 0xBF–0xFF. Users must determine optimized values for these registers during development and program the values into EEPROM before calibration and manufacturing tests.

Determine Target Output Power[1]

Limitations exist in deciding target output power for each board design using the Atheros chip sets. Per requirements, Spectral Mask, Occupied Bandwidth requirements, and production yield are key in deciding maximum output power for calibration. Target output level must be decided during development and pilot runs for a design to achieve best performance, while having reasonable production yield on the manufacturing line.

Appendix B outlines a sample target output power. After the STA adapter designs pilot run in the manufacturing test calibration program, modify the sample appropriately. A card passes manufacturing tests if it meets the specified target output power and passes all specification and performance tests, such as Spectral Mask, OBW, PER, and Receive Sensitivity.

The target power must be specified for each data rate. In several reference designs with the AR5001 and AR5002 chip sets, for data rates of 6–24 Mbps, the target power is spectral mask limited. For data rates over 24 Mbps, the target power is PER limited, because the 64 QAM and 16 QAM modulations used for the higher data rates require a higher signal-to-noise ratio (SNR) to successfully reconstruct the received signal than do the BPSK and QPSK modulations used for lower data rates. Thus on the transmitter side, the transmit power for 36–54 Mbps must be reduced to maintain linearity of the transmitted signal. The PER determines the power level for these data rates.

IQ_Cal

The IQ_cal step calibrates the mismatch between I and Q receive paths. It sets up the DUT for receive, and transmits a predetermined number of frames from the GU. The chip is set in a special mode during receive, and estimates of energy in I and Q channels are recorded. At the end of receive, these estimates are used to compute the mismatch in the I and Q receive paths.

The mismatch is programmed into appropriate registers in the chip, and its appropriate correction is applied by the demodulation circuitry before deciphering symbols from the constellation space, thus preventing an erroneous detection and helping to improve the receive sensitivity, especially for the higher data rate modulations.

1. The determination of this target power has to be done once for each new design before production.

Manufacturing Test Sequence

Temperature Margin Test

The temperature margin test checks that the test channel transmit output power meets target levels over a temperature range of 0–85 °C. Note that the RF front-end circuits must be carefully matched to minimize gain loss so they do not compromise the usable adjustable range of the transmit output power.

It is extremely important for the optimization accuracy to provide a measured operating temperature of the chip case in the calibration environment while it transmits at a 6 Mbps target power level, as set in the CASE_TEMPERATURE parameter in **calsetup.txt**. The gainF_upper and gainF_lower limits are based on CASE_TEMPERATURE, and the temperature during the test is considered CASE_TEMPERATURE, so this parameter must be accurately conveyed.

The chip is operated in open-loop power control mode and the output power is recorded for two values of gainI: gainF_lower and gainF_upper (pwr_lo and pwr_hi, respectively). For this test to pass at a given channel, the pwr_hi must be greater than the 6 Mbps target power, and pwr_lo must be less than the 54 Mbps target power.

Output Power Calibration

This section describes output power calibration performances.

eep_map = 0

For 802.11a mode calibration, measurements are made in steps of 30 MHz from 5.150 – 5.850 GHz. At each frequency, it measures output power is for several PCDAC values from 1 to 63. From this raw dataset, the calibration routine computes the ten most important frequency piers. This calibration:

- Computes the values pcdacMin and pcdacMax at each frequency pier
- Interpolates output power at pcdacMax and pcdacMin intercepts for each frequency pier (0%, 10%, 20%, 30%, 40%, 50%, 60%, 70%, 80%, 90%, and 100%) then programs this data into EEPROM as calibration data
- For 802.11b and 802.11g: employs a similar scheme to store the calibration data, except that 802.11 b uses fixed pier locations of 2.412 GHz, 2.447 GHz, and 2.484 GHz, and 802.11g uses 2.312 GHz, 2.412 GHz, 2.484 GHz

eep_map = 1

AR5112 uses an advanced power control scheme to provide accurate power control down to the lowest attainable power levels. To cover the maximum range of output power, power versus PCDAC data is recorded for up to two values of xpd_gain settings. The xpd_gain values calibration is performed for are specified in the rf_pdgain_lo and rf_pdgain_hi settings in the .eep files. If the two values are the same, calibration performs for only one xpd_gain. This calibration is faster, but can compromise the power control range at low power levels. For eep_map = 1, specify the desired forced_piers_list. Measurements described here are performed at each frequency specified in the list, and stored in an appropriate format in the EEPROM.

For this calibration:

- For lower xpd_gain setting: obtains the highest PCDAC (pcd_max) for which internal Tx gain is not saturated
- pcd_1 = 1 or 25
- Measures the highest linear power (Pwr3)
- Measures the saturated power (Pwr4) at pcd_4
- Measures power at 70% intercept (Pwr2) of pcd_1 and pcd_3
- For higher xpd_gain setting: measures power at PCDACs of 20, 35, and 63
- Linearly interpolates power between these levels.

eep_map = 2

All fifth-generation and beyond Atheros chipsets use an average power-based power control mechanism. Fundamentally, instead of dynamically controlling the peak power at the beginning of a packet during short preamble, the state machine now measures the average power over the previous packet and adjusts the controls deterministically to output correct power for the next packet. Thus, PCDAC is replaced by a power detector analog-to-digital converter (PDADC). To cover the maximum range of output power, power versus PDADC data is recorded for up to two values of pd_gain settings. The performed pd_gain values calibration are currently fixed to 1 (3 in the ART software) to adequately cover the power control range.

For eep_map = 2, it is important to specify the desired forced_piers_list in the **calstup.txt** file. Measurements described here are performed at each frequency specified in the list, and stored in an appropriate format in the EEPROM. In this calibration:

- For higher pd_gain setting: reads back PDADCs for power levels of 0, 3, 6, and 9 dBm
- For lower pd_gain setting: reads back PDADCs for power levels of 7, 11, 15, 19, and 23 dBm (approximately)
- Linearly interpolates PDADC-vs-power between these levels

Power Control Test

After calibration, this test verifies whether the DUT can control the output at desired target power levels. Typically, the DUT controls power within ±0.5 dB in the upper 8 to 10 dBm from Psat, and within ±1 dB for the next 6 to 8 dBm. It is possible to specify upper and lower tolerances for the power control test in the file **calsetup.txt**. The same tolerances are used for tests in all three modes of operation.

Spectral Mask and Occupied Bandwidth Test

The manufacturing test program then executes the spectral mask test and the occupied bandwidth (OBW) test. These tests are performed to guarantee that the target output power set in the calibration process will not violate the regulatory requirements. These tests are performed at the test frequencies marked for the mask test in the file **calsetup.txt**. This test is performed only for the 802.11a mode of operation.

After completing DUT power calibration, the DUT sends a sync to the GU over the Ethernet and uses the ACK for the sync confirmation to proceed to the PER/PPM and Receive Sensitivity tests. The test program running on the GU depends on receiving the sync over the Ethernet from the DUT to synchronize the timing of the PER/PPM and Receive Sensitivity tests.

Packet Error Rate Test

The packet error rate (PER) test checks the transmit path and transmitted signal quality of the DUT to guarantee that the calibrated power level for the DUT causes no signal saturation on the transmit path. In the PER test, the DUT is the transmitter and GU is the receiver. The DUT transmits 100 packets with a packet size of 1000 bytes to the GU. The DUT sends the same packet at four different data rates (6 Mbps, 36 Mbps, 48 Mbps, and 54 Mbps) on the test frequencies marked for the PER test in the file **calsetup.txt**. After receiving these packets, the GU returns the receive status, including PER, RSSI, and the clock difference between the DUT and GU in ppm.

If the flag TEST_TURBO_MODE is set in the file **calSetup.txt**, the same test is repeated in Atheros Turbo Mode™. This test provides independent control flags to turn on the test for all three modes of operation. The turbo mode test is supported only for 5 GHz.

Receive Sensitivity Test

The Receive Sensitivity test checks the receiving path of the DUT to guarantee that the receiving sensitivity of the DUT meets IEEE 802.11a specifications. In the receive sensitivity test, the DUT is the receiver and GU is the transmitter. The GU transmits 100 packets with packet size of 1000 bytes to the DUT. The GU transmits the same packet with two different data rates (48 Mbps and 54 Mbps) on the test frequencies marked for PER test in the file **calsetup.txt**. After receiving the packets from the GU, the DUT checks the PER to determine whether this DUT has passed the test. Usually, the receiving sensitivity for data rates of 36 Mbps and lower have better margins, so this test only selects 48 Mbps and 54 Mbps to shorten testing time. The flag LO_RATE_SEN also allows optional test sensitivity at 6 Mbps.

For sensitivity testing at 48 and 54 Mbps, adjust the attenuator to present a signal level equal to the target sensitivity specified in the SEN_TGT column of the test channel matrix.

If the flag TEST_TURBO_MODE is set in the file **calSetup.txt**, the same test is repeated in Atheros Turbo Mode.

If the flag TEST_HALF_RATE_MODE is set in the file **calSetup.txt**, the same test is repeated in the half rate mode, where the transmission occupies only 10 MHz bandwidth rather than the normal 20 MHz. This mode is introduced to support the lower Japan band (4.9–5.15 GHz), but the test can be performed on any 802.11a channel.

If the flag LOW_RATE_SEN is checked for a frequency in the test channel matrix, additional packets at 6 Mbps are also transmitted by the GU and their receive statistics are reported by the DUT. Attenuation is appropriately adjust for the low rate sensitivity test using the LO_RATE_SEN_TGT value from the test channel matrix as the target 6 Mbps sensitivity level at that channel.

This test provides independent control flags to turn on the test for all three modes of operation. The turbo mode test is supported only for 5 GHz.

After the Receive Sensitivity test, the program running on the GU loops back to wait for the next wakeup call from the DUT.

Throughput Test

The throughput test checks the DUT transmit path to guarantee no delays in delivering the packet over the wireless interface. The PER test may not reveal any deficiency, yet the throughput could be less than desired. Such situations arise when a contention is detected in getting on the wireless medium. Thus when the transmitter can get on the air it successfully transmits a packet, but if it takes too long to get on the air it brings the throughput down. This test is designed to catch this type of throughput.

In this test, the DUT is the transmitter and GU is the receiver. The DUT transmits 500 packets at 54 Mbps (100 packets at 11 Mbps for 11b) with packet size of 1500 bytes to the GU. This test is performed for modes enabled in the TEST_THROUGHPUT flag in the file **calsetup.txt**. The DUT transmits the packets on the test frequencies marked for throughput testing in the test channel matrix in the file **calsetup.txt**. After completing unicast transmission of these packets, the DUT computes the throughput and compares it against the PASS_THRESHOLD specified for the throughput test for that channel in **calsetup.txt** to determine whether this DUT has passed the test.

Data Integrity Test

In the data integrity test checks the on-chip buffer that stores the data for the transmit packet to guarantee that no stuck at faults in that buffer. This test transmits two jumbo frames of size 2400 bytes with various data patterns (all zeroes, all ones, walking zeroes, walking ones, 0xAAAA, 0x5555, etc.) and verifies the packet data at the receiver. The DUT is the transmitter and GU is the receiver for this test.

This test could be enabled for either the 802.11a or 802.11g mode, but can test only one mode.

Program MAC ID

If the DUT passes all of these tests, the manufacturing test program writes a legitimate MAC ID to the DUT EEPROM. A file is used to track the MAC ID assigned to a DUT once it passes the MFG tests. This file stores the Starting ID and the Ending ID for the block of MAC IDs assigned to the production line, as well as the last MAC ID used. The manufacturing test program increments the last MAC ID used automatically once that ID is assigned to a DUT.

If some tests fail, however, the MAC ID is not automatically programmed, but the operator is prompted whether or not to program a MAC ID.

3
Set Up Test Equipment

This chapter describes the test equipment and software setup used for calibration of the Atheros chip sets in a CardBus, Mini PCI, PCI, or access point (AP) design. This documentation assumes that the test equipment used is listed in the Atheros-recommended list. If the test equipment used differs from the suggested list, software modifications to the Atheros Radio Test (ART) may be required.

Hardware Test Equipment Setup

Connect all test equipment used as shown in Figure 1-1, then set up according to the appropriate procedure.

GPIB-ENET/100 from National Instruments

To set up GPIB-ENET/100:

1. Install the National Instruments software.
2. Configure the PC's Ethernet controller with the same IP address subnet of the GPIB-ENET/100 (that is, 192.168.1.1).
3. Configure GPIB-ENET/100 from Start > Programs > NI-488.2M for Windows NT > GPIB-ENET100 Utilities > Device Configuration.
   a. A dialog box similar to Figure 3-1 appears.

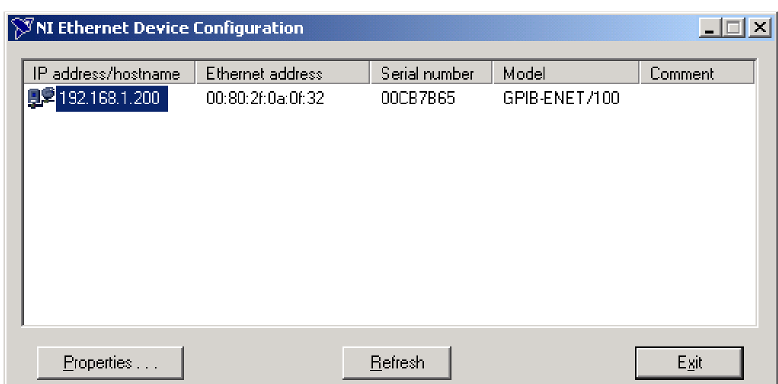

Figure 3-1. **GPIB Device Configuration Dialog**

b. Select the Property box and assign the IP address (that is, 192.168.1.200) and subnet mask for the GPIB-ENET/100.

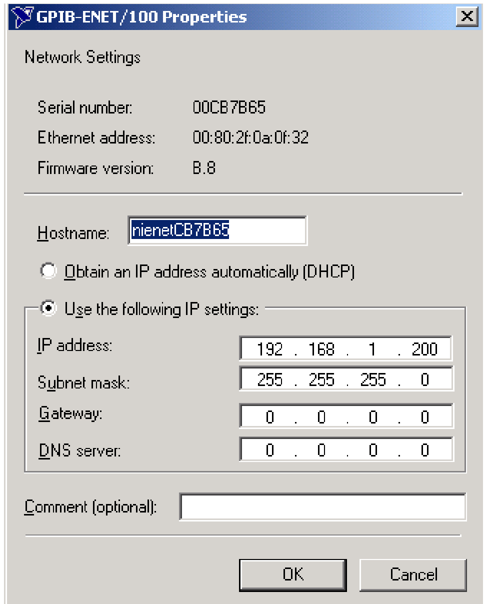

Figure 3-2. **GPIB Device Configuration Properties Dialog**

4. Open a DOS window and Ping 192.168.1.200 to make sure the PC can communicate with the GPIB-ENET/100. If it sees no responses, make sure it is using a crossover Ethernet cable if connecting the PC straight to the GPIB-ENET/100, or a straight Ethernet cable for a hub configuration.

5. Configure the GPIB-ENET/100 interface to ART software:

   a. In the Control Panel, select the GPIB icon.

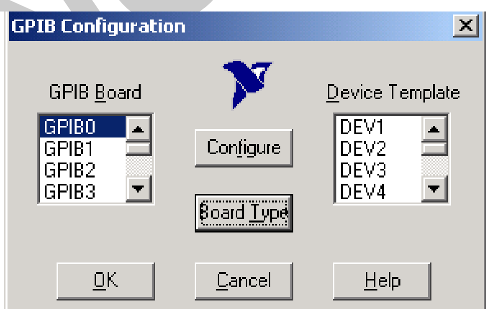

Figure 3-3. **GPIB Software Interface Configuration**

   b. Select Board Type as shown in Figure 3-4 and click OK.

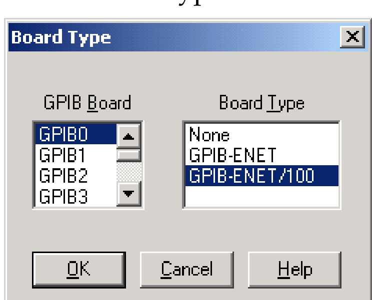

Figure 3-4. **GPIB Board Type Configuration**

c. Select Configure and enter the same IP address that was assigned to the GPIB-ENET/100 and click OK.

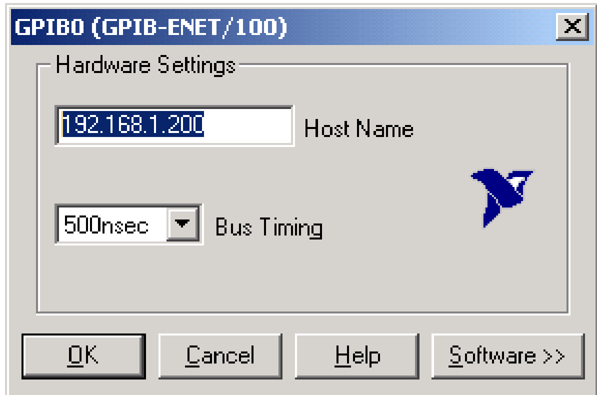

Figure 3-5. **GPIB IP Address Configuration**

Power Meter E4416A from Agilent Technologies

To set up the power meter E4416A:

1. Make sure Self Calibration and Zeroing is complete.
2. Connect Senor to the Power Ref connector and press the Zero/Cal button (first make sure that Power Ref is in the Off state).
3. Assign a GPIB address for the power meter by choosing System > Remote Interface > Configure Interface > GPIB from the menu. Enter **13** for the GPIB address.

Spectrum Analyzer E4404B from Agilent Technologies

Assign the GPIB address for spectrum analyzer by choosing System > Remote Port from the menu. Enter **18** for the GPIB address.

Attenuator Switch Driver 11713A from Agilent Technologies

To set up switch driver 11713A:

1. Assign the GPIB address for the attenuator switch by adjusting the dip switches behind the instrument. Enter **6** for the GPIB address.
2. Connect the 1 dB increment Attenuator 8494H to the X Atten connector behind the instrument.
3. Connect the 10 dB increment Attenuator 8496H to the Y Atten connector behind the instrument.

NOTE: Ensure that the GPIB address assigned to the test equipment matches the GPIB address in the ART software (in the file **calsetup.txt**).

Software Setup

This section describes software setup for the ART.

ART Setup

Follow the instructions in the *Atheros Radio Test Reference Guide* to install ART on the device under test (DUT) and golden unit (GU) machines, then ensure that the DUT_CARD_TYPE variable in the file **artsetup.txt** refers to the correct card type. This is important because all of the **calsetup_<cardtype>.txt** files for all cards exist in the same directory, and the DUT_CARD_TYPE variable is the only way to indicate the correct calsetup file to the GU.

To launch ART on the GU machine:

1. Enter the command **art** in the appropriate directory.
2. From the main menu in ART, select **m** for manufacturing.
3. On the calibration menu, select **g** for golden unit (GU).
4. Next, on the DUT machine:
   a. Type **art** or **art \card=<cardtype>** in the appropriate run directory if calibrating a client.
   b. Type **art \remote=xyz.abc.pqr.lmn** in the appropriate directory if calibrating an AP, where xyz.abc.pqr.lmn is the IP address of the AP.
5. From the main menu in ART, select **m** for manufacturing.
6. Select **d** in the subsequent calibration menu on the DUT machine.

Configuration Files

Table 3-1 summarizes the four files that must be configured correctly for the manufacturing flow to work smoothly. These files are located in the ART installation directory.

Table 3-1. **Configuration Files**

File	Description
atheros-eep.txt	Contains the EEPROM generic data programmed in the beginning of the calibration sequence, before the power measurements are made. Contents of this file and how they are used are described in detail in the *AR5005 EEPROM Device Configuration Guide*.
macID.txt	Contains the start and end values of the WLAN MAC address and the Ethernet MAC address used in the current production line setup. This file also holds the current programmed MAC addresses and increments the address when an address has been used. The Ethernet MAC address is programmed only for the AP calibration. The parameter NUM_ETHERNET_PORTS in the file **calsetup_<cardtype>.txt** specifies the number of Ethernet MAC addresses to assign to each unit.
calsetup.txt	Conveys the calibration setup information to the ART. This file contains information about instrumentation, including GPIB addresses, attenuation factors, GU calibration values, benchID, flags controlling which tests to run, and test channel matrices for modes 802.11a, 802.11b, and 802.11g, describing which tests to run at which channels in various modes.

Table 3-1. **Configuration Files (continued)**

ar500*x*_*??*.eep	Contains the subsystem design-specific hardware settings programmed in EEPROM locations 0xBF-0x150. This file also contains the name of the appropriate target power file to use. For proper synchronization of test frequencies in the PER and receive sensitivity tests, use the same target power file for both the DUT and GU by specifying the subsystem ID of the DUT in the file **artsetup.txt** with the parameter DUT_CARD_SSID. Settings in the config_section of these files are also used by ART for general operation besides manufacturing calibration. Refer to the *Atheros Radio Test Reference Guide* for details.
calTargetPower_ar500*x*_*??*.txt	Conveys the target power and CTL information to the manufacturing test routine programmed into the EEPROM (see the *AR5005 EEPROM Device Configuration Guide*.) The name of this file can be changed. The name of the appropriate file to use for this purpose should be mentioned in **ar500*x*_*??*.eep** for the parameter TARGET_POWER_FILENAME.
artsetup.txt	Refer to the *Atheros Radio Test Reference Guide*.

Explanation of calSetup.txt Entries

Table 3-2 and Table 3-3 explain the entries made in the file **calsetup.txt**.

Table 3-2. **Test Station Setup Parameters in calsetup.txt**

Test Setup Parameter	Description
CASE_TEMPERATURE	Temperature of the DUT during the calibration procedure. This temperature is required for the transmit power optimization procedure.
GOLDEN_IP_ADDR	IP address of the machine running ART for the GU with Ethernet communication between ART sessions for the DUT and GU to synchronize actions. This entry speeds up the calibration process. (Note: if the GU is an AP with ART running remotely on the DUT machine, this address should be specified as ".")
GOLDEN_PPM	The GU used should be tested for PPM offset from the desired center frequency at the channels used for FCC and MKK, and an average should be taken. Measure this from a spectrum analyzer. It is possible to get a negative PPM.
GOLDEN_Tx_POWER	Power output in dBm desired from the GU for the 802.11a sensitivity test.
11b_GOLDEN_Tx_POWER	Power output in dBm desired from the GU for the 802.11b sensitivity test.
11g_GOLDEN_Tx_POWER	Power output in dBm desired from the GU for the 802.11g sensitivity test.
MAX_POWER_CAP	During power measurements for 802.11a calibration, do not exceed this limit on output power. Provided as a safety mechanism.

Table 3-2. **Test Station Setup Parameters in calsetup.txt (continued)**

11b_MAX_POWER_CAP	During power measurements for 802.11b calibration, do not exceed this limit on output power. Provided as a safety mechanism.
11g_MAX_POWER_CAP	During power measurements for 802.11g calibration, do not exceed this limit on output power. Provided as a safety mechanism.
TARGET_POWER_TOLERANCE_UPPER	Upper tolerance in dB for target power control test for all modes 802.11a/802.11b/802.11g.
TARGET_POWER_TOLERANCE_LOWER	Lower tolerance in dB for target power control test for all modes 802.11a/802.11b/802.11g.
PER_PASS_LIMIT	Pass criterion for PER tests for all modes. Default 90%.
SEN_PASS_LIMIT	Pass criterion for Rx_SEN tests for all modes. Default 90%.
PPM_MAX_LIMIT	Upper Pass criterion for PPM readback in Rx_SEN test. Default 9.
PPM_MIN_LIMIT	Lower Pass criterion for PPM readback in Rx_SEN test. Default –9.
NUM_MASK_FAIL_POINTS	Pass criterion for spectral mask test.
CAL_FIXED_GAIN	Fixed gain step to use for 802.11a calibration measurements. If specified in the file **calsetup.txt,** this parameter suppresses the automatic determination of the optimum fixed gain used during cal. It represents the dynamic optimization ladder step number. Default value is 6.
11b_CAL_FIXED_GAIN	Fixed gain step to use for 802.11b calibration measurements. If specified in the file **calsetup.txt,** this parameter suppresses the automatic determination of the optimum fixed gain used during cal. It represents the dynamic optimization ladder step number. Default value is 6.
11g_CAL_FIXED_GAIN	Fixed gain step to use for 802.11g calibration measurements. If specified in the file **calsetup.txt,** this parameter suppresses the automatic determination of the optimum fixed gain used during cal. It represents the dynamic optimization ladder step number. Default value is 6.
ATTEN_DUT_PM	The attenuation at 5 GHz between the DUT and the power meter, including all cables and splitters. Make sure the splitter has a termination if it has an open end.
ATTEN_DUT_SA	The attenuation at 5 GHz between the DUT and the spectrum analyzer, including all cables and splitters. Make sure the splitter has a termination if there is an open end.
ATTEN_FIXED_DUT_GOLDEN	The attenuation between the GU and the DUT at 5 GHz, including all cables, splitters, connectors, and fixed attenuators (if any), but not including the variable attenuators.

Table 3-2. **Test Station Setup Parameters in calsetup.txt (continued)**

11b_ATTEN_DUT_PM	The attenuation at 2.4 GHz between the DUT and the power meter, including all cables and splitters. Make sure the splitter has a termination if it has an open end.
11b_ATTEN_DUT_SA	The attenuation at 2.4 GHz between the DUT and the spectrum analyzer, including all cables and the splitter. Make sure the splitter has termination if it has an open end.
11b_ATTEN_FIXED_DUT_GOLDEN	The attenuation between the GU and DUT at 2.4 GHz, including all cables, splitters, connectors, and fixed attenuators (if any), but not including variable attenuators.
PM_MODEL	The power meter model to use the correct GPIB commands. Supported models are: ■ HP 436A ■ Agilent E4416A ■ 4531
PM_GPIB_ADDR	GPIB address of the power meter.
SA_GPIB_ADDR	GPIB address of the spectrum analyzer.
ATT_GPIB_ADDR	GPIB address of the attenuator controller.

Table 3-3 summarizes the configuration settings in **calsetup.txt** that control the test flow.

Table 3-3. **Test Flow Control Settings in calsetup.txt**

Setting	Description
TEST_TEMP_MARGIN	Indicates whether to test for the margin needed to control target powers in 802.11a mode over a temperature range of 0–85 °C. The channels to perform the test on are determined by the frequencies marked for TEMP_MARGIN_TEST in the 802.11a test channel matrix in **calsetup.txt**.
CAL_POWER	Indicates whether to collect raw power calibration data. ART uses A_MODE, B_MODE, and G_MODE settings in the respective .eep file to figure out which modes it needs to collect data for.
CAL_PHASE	Indicates whether to collect calibration data for phase delta between two chains (valid for AR5513 only). ART uses A_MODE and G_MODE settings in the respective .eep file to figure out which modes it needs to collect data for. Predetermined channels are used: A_MODE: 5000, 5200, 5400, 5600, 5800 B_MODE: 2412, 2472
DO_IQ_CAL	Performs IQ mismatch calibration and stores the coefficients in the EEPROM. IQ_cal is performed for modes supporting OFDM modulation (802.11a and 802.11g).
TEST_TARGET_POWER	Indicates whether the target power control test should be performed for 802.11a mode. The channels and rates to perform the test on are determined by the frequencies marked for TGT_PWR_TEST in the 802.11a test channel matrix in **calsetup.txt**. The 4-bit mask determines target powers to test rates for.

Table 3-3. **Test Flow Control Settings in calsetup.txt (continued)**

Setting	Description
11b_TEST_TARGET_POWER	Indicates whether the target power control test should be performed for 802.11b mode. The channels and rates to perform the test on are determined by the frequencies marked for TGT_PWR_TEST in the 802.11b test channel matrix in **calsetup.txt**. The 4-bit mask determines target powers to test rates for.
11g_TEST_TARGET_POWER	Indicates whether the target power control test should be performed for 802.11g mode. The channels and rates to perform the test on are determined by the frequencies marked for TGT_PWR_TEST in the 802.11g test channel matrix in **calsetup.txt**. The 4-bit mask determines target powers to test rates for.
TEST_SPEC_MASK	Indicates whether the spectral mask test should be performed for 802.11a mode. The channels and rates to perform the test on are determined by the frequencies marked for MASK_TEST in the 802.11a test channel matrix in **calsetup.txt**.
11b_TEST_SPEC_MASK	Indicates whether the spectral mask test should be performed for 802.11b mode. The channels and rates to perform the test on are determined by the frequencies marked for MASK_TEST in the 802.11b test channel matrix in **calsetup.txt**.
11g_TEST_SPEC_MASK	Indicating whether the spectral mask test should be performed for 802.11g mode. The channels to perform the test on are determined by the frequencies marked for MASK_TEST in the 802.11g test channel matrix in **calsetup.txt**.
TEST_OBW_MASK	Indicates whether the OBW test should be performed for 802.11a mode. The channels to perform the test on are determined by the frequencies marked for OBW_TEST in the 802.11a test channel matrix in **calsetup.txt**.
TEST_TxPER	Indicates whether the transmit packet error rate test should be performed for 802.11a mode. The channels to perform the test on are determined by the frequencies marked for PER_TEST in the 802.11a test channel matrix in **calsetup.txt**.
11b_TEST_TxPER	Indicates whether the transmit packet error rate test should be performed for 802.11b mode. The channels to perform the test on are determined by the frequencies marked for PER_TEST in the 802.11b test channel matrix in **calsetup.txt**.
11g_TEST_TxPER	Indicates whether the transmit packet error rate test should be performed for 802.11g mode. The channels to perform the test on are determined by the frequencies marked for PER_TEST in the 802.11g test channel matrix in **calsetup.txt**.
TEST_RxSEN	Indicates whether the receive sensitivity test should be performed for 802.11a mode. The channels to perform the test on are determined by the frequencies marked for PER_TEST in the 802.11a test channel matrix in **calsetup.txt**.

Table 3-3. **Test Flow Control Settings in calsetup.txt (continued)**

Setting	Description
11b_TEST_RxSEN	Indicates whether the receive sensitivity test should be performed for 802.11b mode. The channels to perform the test on are determined by the frequencies marked for PER_TEST in the 802.11b test channel matrix in **calsetup.txt**.
11g_TEST_RxSEN	Indicates whether the receive sensitivity test should be performed for 802.11g mode. The channels to perform the test on are determined by the frequencies marked for PER_TEST in the 802.11g test channel matrix in **calsetup.txt**.
TEST_TURBO_MODE	Indicates whether the PER and receive sensitivity tests should be performed for 802.11a TURBO mode. The channels to perform the test on are determined by the frequencies marked for TURBO in the 802.11a test channel matrix in **calsetup.txt**.
TEST_HALF_RATE_MODE	Indicates whether the PER and receive sensitivity tests should be performed for 802.11a HALF_RATE mode (10 MHz bandwidth). The channels to perform the test on are determined by the frequencies marked for HALF_RATE in the 802.11a test channel matrix in **calsetup.txt**.
TEST_THROUGHPUT	Indicates whether the 54 Mbps throughput test should be performed for 802.11a mode. The channels to perform the test on are determined by the frequencies marked for throughput test in the 802.11a test channel matrix in **calsetup.txt**.
11g_TEST_THROUGHPUT	Indicates whether the 54 Mbps throughput test should be performed for 802.11g mode. The channels to perform the test on are determined by the frequencies marked for throughput test in the 802.11g test channel matrix in **calsetup.txt**.
11b_TEST_THROUGHPUT	Indicates whether the 11 Mbps (short preamble) throughput test should be performed for 802.11b mode. The channels to perform the test on are determined by the frequencies marked for throughput test in the 802.11b test channel matrix in **calsetup.txt**.
TEST_DATA_INTEGRITY	Indicates whether the data integrity test should be performed for 802.11a mode. The test is performed at 5260 MHz channel if enabled.
11g_TEST_DATA_INTEGRITY	Indicates whether the data integrity test should be performed for 802.11g mode. The test is performed at 2462 MHz channel if enabled.
TEST_BOTH_CHAINS	Indicates whether to repeat all the tests listed above for the 2nd chain as well (valid for AR5513 only).
SUB_VENDOR_ID	The hex ID to store in EEPROM location 0x08 as the subsystem vendor ID.
COUNTRY_OR_DOMAIN_FLAG	Indicates whether the code stored is a country code or a regulatory domain code. ■ 0: Regulatory domain code ■ 1: Country code
WORLD_WIDE_ROAMING_FLAG	Indicates a single SKU covering worldwide roaming. (Refer to the Support Bulletin "*Worldwide Roaming Design Specification*" for more information.)

Table 3-3. **Test Flow Control Settings in calsetup.txt (continued)**

Setting	Description
COUNTRY_OR_DOMAIN_CODE	A 12-bit country or regulatory domain code which the card operates in. This code can be changed to any supported country/domain code, and the calibration remains useful as long as the appropriate CTLs are programmed in the EEPROM. (See the Support Bulletin "*Set up for Country or Regulatory Domain*" for details on how station (STA) and AP software use this field.)
READ_FROM_FILE	Indicates whether to bypass power measurements and read the raw data for 802.11a mode from a file.
RAW_DATA_FILENAME	Name of the file to read the raw data for 802.11a mode from, if READ_FROM_FILE is set to 1.
11b_READ_FROM_FILE	Indicates whether to bypass power measurements and read the raw data for 802.11b mode from a file.
11b_RAW_DATA_FILENAME	Name of the file to read the raw data for 802.11b mode from, if 11b_READ_FROM_FILE is set to 1.
11g_READ_FROM_FILE	Indicates whether to bypass power measurements and read the raw data for 802.11g mode from a file.
11g_RAW_DATA_FILENAME	Name of the file to read the raw data for 802.11g mode from, if 11g_READ_FROM_FILE is set to 1.
FORCE_PIERS	Indicates whether to make measurements over the AUTO_PIERS range (default: 4900 – 5850 MHz in steps of 70 MHz) and automatically compute the pier locations for each card, or to force the frequency piers to locations specified in the FORCE_PIERS_LIST and perform power measurements only at these frequencies: ■ 0: make measurements over the AUTO_PIERS range ■ 1: make measurements only at the frequencies in FORCE_PIERS_LIST Use the full measurement option for most accurate results for each card. If all cards in the pilot run are nearly identical in output power measurements over the entire frequency range, this option can be used to save time at the expense of accuracy. Results of power measurements are stored in **cal_AR5211_power.log** after each calibration run. FORCE_PIERS must be set to 1 for eep_map = 1.
FORCE_PIERS_LIST	List of channels used as frequency piers. Although the list allows a maximum of 10 piers, fewer can be specified to save time.
FORCE_PIERS_11b	Indicates whether to make 802.11b cal measurements over the built-in 3 channels (2412, 2447, and 2484 MHz), or only at the frequencies specified in FORCE_PIERS_LIST_11b: ■ 0: make measurements over 3 built-in channels ■ 1: make measurements only at the frequencies in FORCE_PIERS_LIST_11b Use the full measurement option for the most accurate results for each card. If the 802.11b power measurements over the desired frequency range are flat within 0.5 dB in the pilot run, 1 or 2 channels in the FORCE_PIERS_LIST_11b can be used to save time at the expense of accuracy.

Table 3-3. **Test Flow Control Settings in calsetup.txt (continued)**

Setting	Description
FORCE_PIERS_LIST_11b	List of channels used to make 802.11b cal measurements. Specifying more than two channels defeats the purpose because default uses three. To keep things backwards compatible, the cal data at the 802.11b forced piers maps to the built-in three channels then stored in the EEPROM. So EEPROM always has data for 2412, 2447, and 2484 MHz.
FORCE_PIERS_11g	Indicates whether to make 802.11g cal measurements over the built-in three channels (2312, 2412, and 2484 MHz), or only at the frequencies specified in FORCE_PIERS_LIST_11g: ■ 0: make measurements over three built-in channels ■ 1: make measurements only at the frequencies in FORCE_PIERS_LIST_11g The full measurement option is recommended for most accurate results for each card. If the 802.11g power measurements over the desired frequency range are flat within 0.5 dB in the pilot run, 1 or 2 channels in the FORCE_PIERS_LIST_11g can be used to save time at the expense of accuracy.
FORCE_PIERS_LIST_11g	List of channels at which to make 802.11g cal measurements. Specifying more than two channels defeats the purpose because the default uses three channels. To keep things backwards compatible, the cal data at the 802.11b forced piers is mapped to the built-in three channels then stored in the EEPROM. So EEPROM always had data for 2312, 2412, and 2484 MHz.
USE_11g_CAL_FOR_11b	If the power measurements for 802.11b mode are empirically within 0.5 dB of the 802.11g measurements in the pilot run, it may be possible to use the 802.11g cal data for 802.11b mode. This must be empirically verified on each board-type in the pilot run.
AUTO_PIERS_START	The start channel for measurements for automatic computation of pier locations if FORCE_PIERS is 0. Default value is 4900 MHz. Feature not supported using the AR5002 chip set.
AUTO_PIERS_STOP	The stop channel for measurements for automatic computation of pier locations if FORCE_PIERS is 0. Default value is 5850 MHz. Feature not supported using the AR5002 chip set.
AUTO_PIERS_STEP	The channel step-size for measurements for automatic computation of pier locations if FORCE_PIERS is 0. Default value is 70 MHz. Recommended to keep between 10 and 100 MHz. Feature not supported using the AR5002 chip set.
MACID_FILENAME	Name of the file to be used for computing the MAC ID.
CUSTOMER_DEBUG	If set to 1, generates special debug files used by Atheros technical support. The recommended setting is 0.
SHOW_TIMING_REPORT	Prints a summary of time taken by various tests.

Rules for Test Channel Matrices Setup

A test channel matrix is entered in the file **calsetup.txt** for each mode (802.11a/802.11b/802.11g). The matrix allows the customer more control over which tests to run at which channels. These matrices can reduce the test time considerably.

The rules for setting up a test channel matrix are:

- The tags #BEGIN_<11a>_TEST_CHANNEL_MATRIX and #END_<11a>_TEST_CHANNEL_MATRIX are required to mark the beginning and end of the test channel matrix definition. Substitute <11a> with the appropriate mode.
- Blank lines and lines beginning with a "#" are ignored, and therefore can be used for comments.
- Up to 32 test channels can be specified for each mode.
- Flags listed in Table 3-4 are required in the order in which they are listed. They can be space or tab delimited. A comment at the end of line with a "#" is permitted, but not preferable to comment on a separate line.
- Columns for OBW_TEST and HALF_RATE should not be specified for 802.11b and 802.11g modes
- Column for TURBO should not be specified for 802.11b mode.

Table 3-4. **Test Channel Matrix Flags**

Test Flag	Description
PER_TEST	Test PER and sensitivity at this channel
SEN_TGT	Target sensitivity for 54 Mbps for the channel in dBm
LO_RATE_SEN	Test 6 Mbps receive SEN
LO_RATE_SEN_TGT	Target sensitivity for 6 Mbps for the channel in dBm
TURBO_PER_SEN	Test PER and sensitivity in TURBO mode at this channel. This column should not exist in the test channel matrices for 802.11b mode.
HALF_RATE	Test PER and sensitivity in 10 MHz mode at this channel. This mode supports the lower Japanese band at 4.9–5.15 GHz, but can be used at any 802.11a channel. Available only in 802.11a mode. This column should not exist in the test channel matrices for 802.11b and 802.11g modes.
MASK_TEST	Test spectral mask at this channel
OBW_TEST	Test OBW at this channel. Available only in 802.11a mode. This column should not exist in the test channel matrices for 802.11b and 802.11g modes.
TGT_PWR_TEST	Perform target power tests at this channel. The rates at which target power can be tested are: 6, 36, 48, and 54 Mbps. For example: ■ 1111: test target power at all 4 rates ■ 1011: test target power at 6, 48, and 54 Mbps ■ 1110: test target power at 6, 36, and 48 Mbps
TEMP_MARGIN_TEST	Test for temperature margin at this channel
TEST_T/P	Test 54 Mbps (or 11 Mbps in 802.11b mode) throughput at this channel
PASS_THRESHOLD	Throughput test pass criterion (in Mbps) at this channel

Explanation of .eep File Entries

The .eep files contain cal and config sections. ART also uses the config section for operations other than manufacturing calibration, and contains settings critical to proper card operation, such as antenna control, xpd_gain, and amplifier stages biases. See the *Atheros Radio Test Reference Guide* for details.

Table 3-5 explains .eep entries. with the entries' column-to-mode mapping:

- Column 1: 802.11a mode
- Column 2: 802.11a Turbo mode (Atheros proprietary)
- Column 3: 802.11b mode
- Column 4: 802.11g mode
- Column 5: 802.11g Turbo mode (Atheros proprietary)

Column 2 values should be identical to those in column 1. Entries in this column are for use by ART only, and are not stored on the EEPROM. For details on all fields, see the *AR5005 EEPROM Device Configuration Guide*.

Table 3-5. **Config Section Entries in the .eep File**

Test Setup Parameter	Description
bb_switch_table_t1	Stored as Antenna_CTL_11a/b/g_1 on the EEPROM
bb_switch_table_r1	Stored as Antenna_CTL_11a/b/g_2 on the EEPROM
bb_switch_table_r1x1	Stored as Antenna_CTL_11a/b/g_3 on the EEPROM
bb_switch_table_r1x2	Stored as Antenna_CTL_11a/b/g_4 on the EEPROM
bb_switch_table_r1x12	Stored as Antenna_CTL_11a/b/g_5 on the EEPROM
bb_switch_table_t2	Stored as Antenna_CTL_11a/b/g_6 on the EEPROM
bb_switch_table_r2	Stored as Antenna_CTL_11a/b/g_7 on the EEPROM
bb_switch_table_r2x1	Stored as Antenna_CTL_11a/b/g_8 on the EEPROM
bb_switch_table_r2x2	Stored as Antenna_CTL_11a/b/g_9 on the EEPROM
bb_switch_table_r2x12	Stored as Antenna_CTL_11a/b/g_10 on the EEPROM
bb_switch_settling	Stored as Switch_Settling_Time_11a/b/g on the EEPROM
bb_txrxatten	Stored as TxRxatten_11a/b/g on the EEPROM
bb_pga_desired_size	Stored as PGA_Desired_Size_11a/b/g on the EEPROM
bb_adc_desired_size	Stored as ADC_Desired_Size_11a/b/g on the EEPROM
rf_ob	Stored as OB_11a/b/g on the EEPROM
rf_db	Stored as DB_11a/b/g on the EEPROM
rf_b_ob	Stored as b_OB_11b and b_OB_11g on the EEPROM
rf_b_db	Stored as b_DB_11b and b_DB_11g on the EEPROM
rf_xpd	Stored as XPD_11a/b/g on the EEPROM
rf_xpd_gain	Stored as XPD_Gain_11a/b/g on the EEPROM
rf_pdgain_lo, rf_pdgain_hi	XPD_Gain_11a/b/g on the EEPROM in AR5112/AR2112 pdgain_lo). For AR5112/AR2112, two xpd_gain values can be stored on the analog chip as pdgain_hi and pdgain_lo. PCDAC selection MSB toggles on a per-packet basis. Calibration performs for pdgain_lo and pdgain_hi. Specify pdgain_lo = 0 and pdgain_hi = 3 to cover extended power control range. The same value can be specified for a quicker calibration at the expense of power control range.
bb_thresh62	Stored as Thresg62_11a/b/g on the EEPROM
bb_tx_end_to_xpab_off	Stored as Tx_end_to_xpa_off_11a/b/g on the EEPROM
bb_tx_end_to_xpaa_off	Make identical to bb_tx_end_to_xpab_off
bb_tx_frame_to_xpab_on	Stored as Tx_frame_to_xpa_on_11a/b/g on the EEPROM
bb_tx_frame_to_xpaa_on	Make identical to bb_tx_frame_to_xpab_on
bb_tx_end_to_xlna_on	Stored as Tx_end_to_xlna_on_11a/b/g on the EEPROM

Table 3-5. **Config Section Entries in the .eep File (continued)**

chn1_bb_switch_table_t1	Stored as Antenna_CTL_11a/b/g_1 for chain 1
chn1_bb_switch_table_r1	Stored as Antenna_CTL_11a/b/g_2 for chain 1
chn1_bb_switch_table_r1x1	Stored as Antenna_CTL_11a/b/g_3 for chain 1
chn1_bb_switch_table_r1x2	Stored as Antenna_CTL_11a/b/g_4 for chain 1
chn1_bb_switch_table_r1x12	Stored as Antenna_CTL_11a/b/g_5 for chain 1
chn1_bb_switch_table_t2	Stored as Antenna_CTL_11a/b/g_6 for chain 1
chn1_bb_switch_table_r2	Stored as Antenna_CTL_11a/b/g_7 for chain 1
chn1_bb_switch_table_r2x1	Stored as Antenna_CTL_11a/b/g_8 for chain 1
chn1_bb_switch_table_r2x2	Stored as Antenna_CTL_11a/b/g_9 for chain 1
chn1_bb_switch_table_r2x12	Stored as Antenna_CTL_11a/b/g_10 for chain 1

Table 3-6 details the cal section of the .eep files entries. ART uses these settings in the manufacturing calibration flow without impacting general operation.

Table 3-6. **Cal Section Entries in the .eep File**

Test Setup Parameter	Description
SUBSYSTEM_ID	Subsystem ID to be stored in the EEPROM.
EEPROM_MAP_TYPE	Flag indicating the format type of the calibration data stored on the EEPROM. Due to an advanced power control scheme employed for AR5112, the EEPROM format needs to be slightly different: EEPROM_MAP_TYPE = 0 for AR5111 designs EEPROM_MAP_TYPE = 1 for AR5112 designs
TARGET_POWER_FILENAME	Name of the file used to read the target power and test group information. (Refer to Appendix A.)
TURBO_DISABLE	Refer to register TD in the *AR5005 EEPROM Device Configuration Guide.*
RF_SILENT	Refer to register RFK in the *AR5005 EEPROM Device Configuration Guide.*
DEVICE_TYPE	Refer to register DeviceType in the *AR5005 EEPROM Device Configuration Guide.*
TURBO_MAXPOWER_5G	Refer to register 5G_Turbo_2WmaxPower in the *AR5005 EEPROM Device Configuration Guide.*
TURBO_MAXPOWER_2p5G	Refer to register 2p5G_Turbo_2WmaxPower in the *AR5005 EEPROM Device Configuration Guide.*
A_MODE	Refer to register Amode in the *AR5005 EEPROM Device Configuration Guide.*
B_MODE	Refer to register Bmode in the *AR5005 EEPROM Device Configuration Guide.*
G_MODE	Refer to register Gmode in the *AR5005 EEPROM Device Configuration Guide.*
ANTENNA_GAIN_5G	Refer to register Antenna_Gain_5G in the *AR5005 EEPROM Device Configuration Guide.*
ANTENNA_GAIN_2p5G	Refer to register Antenna_Gain_2.5G in the *AR5005 EEPROM Device Configuration Guide.*
XLNA_GAIN	Refer to register XLNA_Gain_11a in the *AR5005 EEPROM Device Configuration Guide.*
NOISE_FLOOR_THRESHOLD	Refer to register Noise_Floor_Thresh_11a in the *AR5005 EEPROM Device Configuration Guide.*
11b_XLNA_GAIN	Refer to register Locations 0xF2-0xFB in the *AR5005 EEPROM Device Configuration Guide.*
11b_NOISE_FLOOR_THRESHOLD	Refer to register Locations 0xF2-0xFB in the *AR5005 EEPROM Device Configuration Guide.*

Table 3-6. **Cal Section Entries in the .eep File**

11g_XLNA_GAIN	Refer to register Locations 0x10D-0x117 in the *AR5005 EEPROM Device Configuration Guide.*
11g_NOISE_FLOOR_THRESHOLD	Refer to register Locations 0x10D-0x117 in the *AR5005 EEPROM Device Configuration Guide.*
11a_FALSE_DETECT_BACKOFF	Refer to register False_Detect_Backoff_11a in the *AR5005 EEPROM Device Configuration Guide.*
11b_FALSE_DETECT_BACKOFF	Refer to register Locations 0xF2-0xFB in the *AR5005 EEPROM Device Configuration Guide.*
11g_FALSE_DETECT_BACKOFF	Refer to register Locations 0x10D-0x117 in the *AR5005 EEPROM Device Configuration Guide.*
CCK_OFDM_DELTA	Refer to register CCK_OFDM_Pwr_Delta in the *AR5005 EEPROM Device Configuration Guide.*
ENABLE_32KHZ	Flag indicating whether a 32 kHz crystal is present on the NIC. This slower clock reduces the power consumption when the device is in the sleep mode.
NUM_ETHERNET_PORTS	Required for an AP or a router reference design. Indicates the number of Ethernet ports present and enabled on the design. This number of unique Ethernet MAC IDs is assigned after calibration.
START_ETHERNET_PORT	0-based index of the first enabled Ethernet port
MODE_MASK_FOR_RADIO_0	Mask for modes supported on radio 0 interface. lsb = 1 indicates support for 802.11b/802.11g, msb = 1 indicates support for 802.11a.
MODE_MASK_FOR_RADIO_1	Mask of modes supported on radio 1 interface. lsb = 1 indicates support for 802.11b/802.11g, msb = 1 indicates support for 802.11a.
CH14_FILTER_CCK_DELTA	Difference in power between ch13 and ch14 for CCK rates for the same pcdac. Specified in dB with a 0.1 dB resolution. Typical value: 1.5 dB.
ENABLE_WAKE_ON_WLAN	If set to 1, indicates the client supports the wake-on-WLAN capability. If set to 0 or not specified, it indicates the client does not support the wake-on-WLAN capability. (Location 0x0C in the EEPROM is set to 0xF9C2 if enabled, otherwise set to 0x01C2.)
Tx_CHAIN_MASK	Mask indicating which chains populate for Tx (only valid for AR5513). Calibration and Tx-related tests perform for appropriate chains based on this mask. 0x1-->chain0 0x2-->chain1
Rx_CHAIN_MASK	Mask indicating which chains are populated for Rx (only valid for AR5513) Rx-related tests perform for appropriate chains based on this mask. 0x1-->chain0 0x2-->chain1
MBAR_MASK	Mask indicating which MBARs to hide upon PCI configuration (only valid for AR5513). 0x0 --> Hide none 0x1 --> Hide MBAR1 0x2 --> Hide MBAR2 0x3 --> Hide MBAR1 and MBAR2
PCI_CONFIG_BASE_OFFSET	Base offset used for PCI configuration information
USE_EEPROM	Flag indicating whether the current design uses an EEPROM or FLASH: 0 --> FLASH 1 --> EEPROM

Support for Dual Chain Calibration

Because the AR5513 supports dual chain operation, if Tx_CHAIN_MASK = 0x3 in the .eep file, then ART performs power calibration for the chains and the phase calibration. For FLASH-based AR5513 designs, calibration information is stored in the last sector of the FLASH. For EEPROM-based designs, this information starts at the first address. Calibration information for the first chain is stored at offsets 0x00–0x3FF in the last sector, and the second chain information at offsets 0x400–0x7FF. If Tx_CHAIN_MASK = 1 or 2, this indicates only one Tx chain stuffed and the phase calibration is skipped. The calibration information is stored at offset 0x00–0x3FF in this case.

The presence of two chains does not require a substantially different calibration setup. It is sufficient to combine the output at default antennas from the two chains using a splitter/combiner before hooking up to the golden unit and other instruments. It is strongly recommended to ensure that the path loss from the combiner to the individual chains be identical.

Explanation of Various Modes in AR5513

Table 3-7 shows the combinations of interfaces supported by the AR5513-based chipset in various modes. Mode selection can be made by tying the Modesel[2:0] pins to GND or VCC on the board as appropriate. On AV10 the reference design, these signals are brought out to the miniPCI connector (pins 103, 105 and 107) for added convenience. This enables the user to modify the adapter board in which AV10 is plugged in for calibration or bring up tuning and bring up all AV10s in a certain mode without having to modify all AV10s to switch the mode, which is handy if the boards are eventually deployed in a mode other than the PCI + CPU mode used for calibration.

Table 3-7. **AR5513 Supported Interfaces**

Modesel [2:0]	Mode	Interfaces
0	Local Bus	MPEG + Local Bus
1	ERMA	
2	PCI + CPU	PCI
3	Cardbus + CPU	PCI
4	PCI only (default)	PCI
5	Cardbus only	PCI
6	GPIO only	GPIOs
7	Reserved	

PCI Configuration Information Base Offset

AR5513 serves up the PCI configuration information from a properly programmed FLASH or EEPROM. In PCI only and Cardbus only modes, the PCI configuration information is read from byte offset 0x00. In PCI + CPU and Cardbus + CPU modes where, the PCI configuration information is read from byte offset 0x800. Users must put appropriate PCI configuration information in this space for desired behavior in end usage and during calibration.

PCI MBAR Masking Capability

AR5513 presents three MBARs to the host during configuration. Listed below are the resource requirements for various MBARs:

MBAR	Size	Target
0	128K	AR5513 Register read/write
1	64M	SDRAM
2	4M	FLASH

On some notebooks, it was found that the cardbus bridge fails to allocate the necessary resources either due to lack of physical resources or the software configuration. In such scenarios, it is possible to program a mask to hide MBAR1 and MBAR2 by programming appropriate bits in the FLASH or EEPROM at byte offsets 0x1E and 0x1F within the PCI configuration section (remember to use appropriate base offset of 0x00 or 0x800 based on the Modesel[2:0]. see PCI Configuration Information Base Offset):

MBAR To Mask	Masking Byte Offset	Masking Bit
MBAR1	PCI config base offset + 0x1E	7
MBAR2	PCI config base offset + 0x1F	1

PCI MBAR Masking Update Capability

At the end of calibration, ART programs the MBAR_MASK as read from the eep file into the PCI configuration information at the PCI_CONFIG_BASE_OFFSET (read from the eep file).

Additionally, there is a facility provided in the ART eeprom menu to either update just the MBAR mask, or program the PCI configuration data from atheros-eep.txt file with the MBAR_MASK read from the appropriate eep file.

FLASH Sector Usage

ART for AR5513 supports 2 MB (32 sector) and 4 MB (64 sector) flash sizes. It autodetects the size of the flash. Here is the layout of sector usage for a flash with N sectors:

- Sectors 0...3: Bootloader
- Sectors 4...(N-4): File system
- Sector N-3: VxWorks Bootline Sector
- Sector N-2: Board Data
- Sector N-1: Calibration Data

Recommended Manufacturing Flow

Use these steps to calibrate using the ART over PCI with as wide a variety of laptops as possible:

1. Prepare the golden FLASH parts. AR5513VA has a paired set of AV10s (Tx and Rx). In typical builds, half the units are Tx, and half Rx. These FLASH parts have the desired final software configuration and software image (e.g, MPEG direction, operating mode, default SSID, etc.).
2. Make sure there is appropriate PCI configuration information exists at the appropriate offset in sector 0, embedded in the bootloader. Atheros released bootloaders contain a valid PCI configuration at offset 0x800 in sector 0. The MBAR_MASK is set to 0x3 in this so as to enable the widest selection of laptops to be able to allocate resources for AV10 in PCI + CPU mode.
3. Burn the FLASH parts from the golden FLASH before mounting on the board.
4. Configure the adapter in which AV10 is plugged-in on the calibration station to put the AR5513 in PCI + CPU mode (ModeSel = 2) by tying the ModeSel[2:0] coming out on the miniPCI connector pins of AV10 to GND or VCC appropriately. It is best to modify the adapter rather than each AV10 individually.
5. Launch ART with appropriate SSID as specified in artsetup.txt and proceed to calibrate.
6. After the calibration, ART will update sector N-1 with calibration data, sector N-2 with MAC addresses for local bus, MPEG and PCI interfaces. It also updates the PCI configuration data in sector 0 with appropriate SSID and MBAR_MASK.

Explanation of Target Power File Entries

This section lists the data setup rules from **calTargetPower_<cardtype>.txt**.

Rules to Set Up Target Power

The rules to set up target power are:

- Lines beginning with "#" are treated as comments and are ignored.
- Start this section with the line "#BEGIN_11a_TARGET_POWER_TABLE".
- Specify up to a maximum of eight test frequencies.
- Specified test frequencies do not need to cover the entire range of 5180 to 5850 MHz, they can provide data for a smaller range. For all channels outside of the test frequency range, target power is assumed 0 dB.
- Specify the spectral mask or PER limited target power for various rates.
- End this section with "#END_11a_TARGET_POWER_TABLE" on a line.
- Repeat these steps for 802.11b target power. Replace "11a" with "11b" in the beginning and end labels. Specify exactly two test frequencies.
- Repeat these steps for 802.11g target powers. Replace "11a" with "11g" in the beginning and end labels. Specify up to three test frequencies.

Rules to Set Up Conformance Testing Limits

The rules to set up conformance testing limits (CTLs) are:

- Lines beginning with "#" are treated as comments and are ignored.
- Start this section with "#BEGIN_TEST_GROUPS" on a line.
- Information for each CTL is specified in three lines:
  - Line 1: the CTL code in hex. The lower three bits of this hex code indicate which mode this CTL pertains to. Subsequent entries on this line are the channels in MHz.
  - Line 2: the max output power permitted on the respective channel. These values must be obtained in the pilot run.
  - Line 3: the in-band flags for various channels. Flag = 0 indicates a band edge and flag=1 indicates an in-band channel. For details on using in-band flags, refer to the *AR5005 EEPROM Device Configuration Guide*.
- Up to eight band edges can be specified for each CTL.
- If no power limitation is desired at a band edge, specify 31.5. Other criteria, such as packet error rate (PER), spectral mask, or regulatory domain constraints limit the operating power at that channel in this case.
- Up to 32 CTLs can be defined.

Executing a Calibration Run

This section details how to execute a successful calibration and manufacturing test run using ART. It is assumed that:

- The desired instruments have been hooked up as specified.
- ART has been installed.
- The files **calsetup.txt**, **calTargetPower_<cardtype>.txt**, **artsetup.txt**, **macID.txt**, and **atheros-eep.txt** have been set up appropriately. Key items to check when setting up a new board or switching between boards are:
  - Specify the correct DUT_CARD_SSID in **artsetup.txt**. If starting with blank cards, set the BLANK_EEP_ID appropriately.
  - Ensure that an appropriate .eep file for the DUT_CARD_SSID has been set up and an entry made in the CFG_TABLE.
  - Ensure that only desired tests are turned on in **calsetup.txt**.

NOTE: For 802.11a-only DUTs, make sure the 802.11b and 802.11g mode tests are turned off.

To start the manufacturing test on the GU side first:

1. Launch a cmd window on the GU.
2. Change to the ART directory.
3. Type **art**.
4. Press the **m** key for manufacturing test.
5. Press the **g** key for GU.

To start the manufacturing test on the DUT side:

1. Launch a cmd window on the DUT machine.
2. Change to the ART directory.
3. Type **art**, **art \card=<cardtype>**, or **art \remote=<IPaddress>**, as appropriate.
4. Press **g** to enable logging, if so desired.
5. Press **m** for manufacturing test.
6. Press **d** for DUT, and follow the on-screen instructions.

4

Manufacturing Test Program Source Code

The Atheros Radio Test (ART) provides features to perform radio testing as well as a sample manufacturing test flow. ART calls the manufacturing library. This chapter describes the this C-based program's architecture.

ART Source Files

Figure 4-1 shows the directory tree for the ART source code.

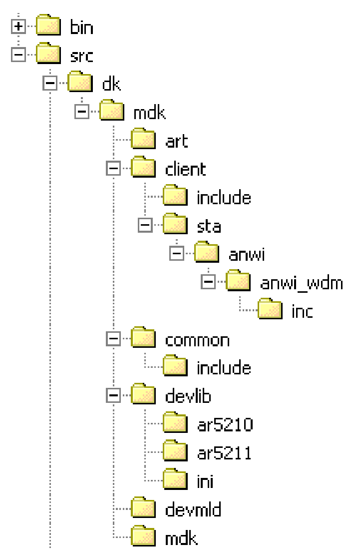

Figure 4-1. **ART Source Directory Tree**

The folder **art\src\dk\mdk\art** contains the Visual C++ project workspace file **art_rel.dsw**. Double click this file to open the workspace (see Figure 4-2).

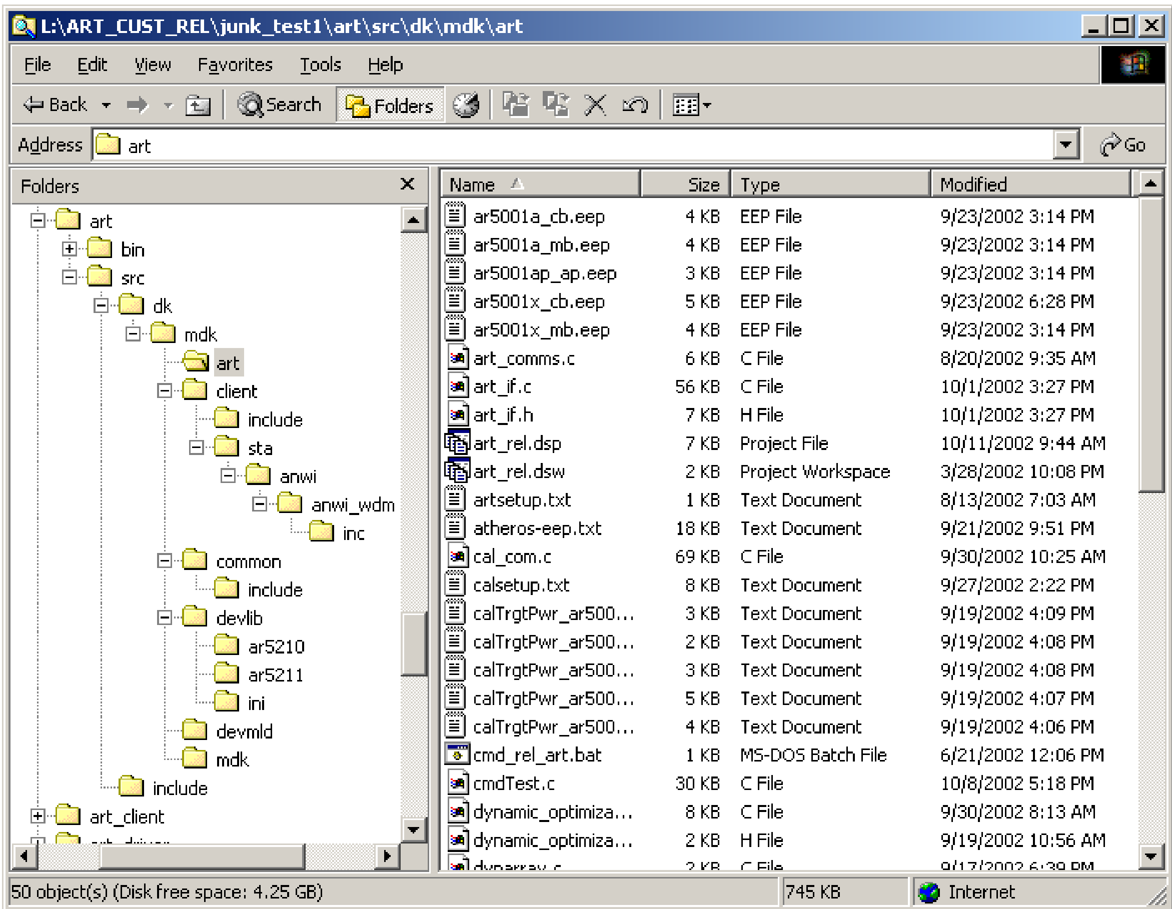

Figure 4-2. **ART Directory of Files Showing Workspace**

Table 4-1 summarizes the source files (under File View, after opening art_rel files and opening Source Files) used to build **art.exe**.

Table 4-1. **art.exe Source Files**

File	Contents
anwi_hw.c	The functions used to access the hardware. Makes calls to the ANWI device driver
MLIBif.c	Wrapper functions for MLIB
test.c	The main entry of the source program of **art.exe**. This file also contains the subroutine for other test functions that are not included in the sample manufacturing test program, such as continuous transmit, link test, and EEPROM content check.
maui_cal.c	All of the subroutines for the sample manufacturing test program (eep_map = 0 format)
cal_com.c	Additional subroutines use for sample manufacturing
cal_gen3.c	Additional subroutines for eep_map = 1 format calibration
art_if.c	Invokes the manufacturing library functions. Either calls the local functions or sends command over a socket interface to a remote AP.
eeprom.c	Functions for formatted display of the EEPROM contents
osWrap_win.c	Functions for handling socket communication to a remote AP
parse.c	Functions to parse the **ar5001*.eep** files
dynArray.c	Generic function for creation and managing a dynamic array
dynamic_optimizations.c	Functions to perform the dynamic optimization, when enabled, for continuous Tx, link test and throughput tests
art_comms.c	Functions for handling ethernet communications used by the manufacturing code
nrutil.c	Reserved for future use
mathRoutines.c	Reserved for future use
rssi_power.c	Reserved for future use
cal_gen5.c	Functions specific to calibration of generation 5 products (eep_map = 2 format)
cmdTest.c	Functions to perform the command line tests available in ART
parseLoadEar.c	Functions to parse .ear files and write EAR information to EEPROM
hw_routines.c	Additional hardware access routines to access the adapter
dk_mem.c	Functions for memory block management

The manufacturing library is located under devlib files\Source Files. Table 4-2 summarizes the source files to build the manufacturing library file **devlib.dll**.

Table 4-2. **devlib.dll Source Files**

File	Contents
athreg.c	Code to parse and add fields from an external Atheros register file
mAlloc.c	Memory allocation and de-allocation functions
mCal.c	Library calls used during adapter calibration
mConfig.c	Configuration and setup functions for devlib
mCont.c	Continuous transmit functions (txContBegin and txContEnd)
mData.c	Data frame transmit functions (txDataSetup, txDataBegin, rxDataSetup, rxDataBegin, txrxDataBegin, txGetStats, rxGetStats, and rxGetData)
mDevtb_rel.c	The register and function table mapping for each of the devices supported by devlib
mEeprom.c	The functions to read and apply a version 3 EEPROM calibration
mInst.c	All the function calls for instruments control
artEar.c	Functions to read EAR from EEPROM and apply register updates
art_ani.c	Functions for performing noise immunity
rate_constants.c	Rate code management functions
stats_routines.c	Hardware statistics gathering functions
usb_pm.c	USB Power Meter functions

The devlib also includes four subdirectories. The AR5210, AR5211, AR5212 and AR2413 directories contain the device specific functions for the generations of products supported by the devlib. Each directory holds equivalent name C-source files as described in Table 4-3.

Table 4-3. **Device Specific Files**

File	Contents
mCfg21x.c	Device specific configuration functions
mData21x.c	Device specific data setup for transmit and receive functionality
mEEP21x.c	Device specific EEPROM load functionality
mAni21x.c	Automatic noise immunity functionality introduced for the AR5212-based chipset.

NOTE: Not all hardware functionality changes with each new generation of chipset. In this case, newer generation chipsets will use the hardware functions from previous generation chipsets.

In addition to the device specific directories, devlib also contains a .ini directory. This directory contains the register configuration tables compiled into ART and all the register field names and values they initialize with on a resetDevice. Each adapter type supported by devlib has a separate .ini file. Table 4-4 lists the adapters supported by these .ini files.

Table 4-4. **INI Files Supported by ART**

ini File	Adapter Type
dk_crete_fez.ini	Adapter containing AR5210 and AR5110 devices.
dk_boss_0012.ini/mod	Adapters containing AR5001 family devices
dk_boss_0013.ini/mod	Adapters containing AR5001+ family devices
dk_0014.ini/mod and dk_0016.ini/mod	Adapters containing AR5002 family devices
dk_0017.ini/mod	Adapters containing AR5004 family devices
dk_0018.ini/mod	Adapters containing AR2413/AR2414 chipsets
dk_0019.ini/mod	Adapters containing AR5413/AR5414 chipsets
dk_00b0.ini/mod	Adapters containing AR5523-based chipsets (USB)

Building ART

To build art.exe and the manufacturing library:

1. Click on "Build" under the VC++ development platform.
2. Click "Set Active Configuration" to select the target build file **art_rel – Win32 customerRelease** or **art_rel - Win32 Debug**, if a debug version is required.
3. Click "Build" or "Rebuild All" to recompile the source. This builds both **art.exe** and the manufacturing library.

The locations where the files will be located after the build if making a release build (customer release) are:

- **art.exe**:
  src\dk\mdk\art\customerRel\art.exe
- **devlib.dll:**
  src\dk\mdk\devlib\customerRel\devlib.dll

or, if a debug version is built, then:

- **art.exe:**
  src\dk\mdk\art\debug\art.exe
- **devlib.dll:**
  src\dk\mdk\devlib\debug\devlib.dll

4. Copy the files **art.exe** and **devlib.dll** to the DUT and the GU test PC platform.

5. Run the **art.exe** program to start the manufacturing test.

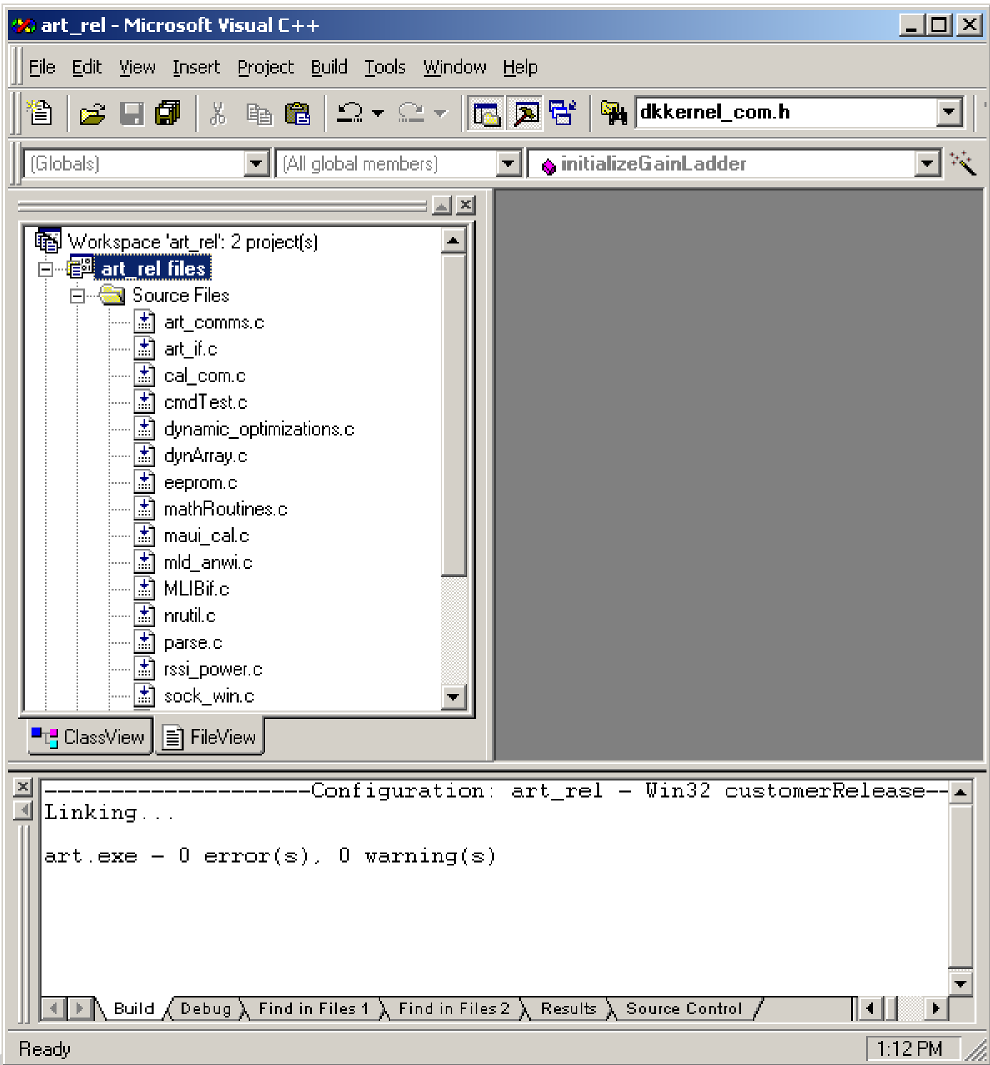

Figure 4-3. **ART Workspace Opened in Visual Studio**

The file **maui_cal.c** contains several key routines. This section describes the functional routines in the program.

CalibrationMenu()

D: dutBegin()/*DUT Test Program*/

- Calibration Setup for instruments used
- dutCalibration()
  - ResetDevice
  - Write_eeprom_Common_Data
  - Setup_raw_datasets
  - Measure_all_channels
  - Make_cal_dataset_from_raw_dataset
  - Program_eeprom

- dutTest()
  - if(CalSetup.testSpecMask)
    dutTestSpecMask()
  - if(CalSetup.testTargetPower)
    dutTestTargetPower(MODE_11a)
  - if(CalSetup.testOBW)
    dutTestOBW()
  - dutSendWakeupCall()
  - if(CalSetup.testTxPER)
    dutTestTxPER()
  - if(CalSetup.testRxSEN)
    dutTestRxSEN()
  - dutTest_2p4(MODE_11g)
  - dutTest_2p4(MODE_11b)
  - if(Test Pass)
    ProgramMACID()
  - dutTest_2p4(mode)
  - if(CalSetup.testSpecMask_2p4[mode])
    dutTestSpecMask_2p4(mode)
  - if(CalSetup.testTargetPower_2p4[mode])
    dutTestTargetPower(mode)
  - dutSendWakeupCall_2p4(mode)
  - if(CalSetup.testTxPER_2p4[mode])
    dutTestTxPER_2p4(mode)
  - if(CalSetup.testRxSEN_2p4[mode])
    dutTestRxSEN_2p4(mode)
- G: goldenTest() /*Golden Unit Test Program*/
  - goldenWait4WakeupCall()
  - goldenTestTxPER
  - goldenTestRxSEN
  - goldenTest_2p4(MODE_11g)
  - goldenTest_2p4(MODE_11b)
- G: goldenTest_2p4(mode) /*GU Test Program for 11b or 11g*/
  - goldenWait4WakeupCall_2p4(mode)
  - goldenTestTxPER_2p4(mode)
  - goldenTestRxSEN_2p4(mode)

A

Sample File calsetup.txt

This appendix contains a sample of the **calsetup.txt** file.

```
# calsetup file updated for eeprom version 3.3 ART Version 2.4
# General Info

COUNTRY_OR_DOMAIN_FLAG   = 0;       # 1 indicates the code is a country code.
                                    # 0=>regulatory domain code
WORLD_WIDE_ROAMING_FLAG  = 0;       # 1 indicates a single SKU covering
                                    # worldwide roaming
COUNTRY_OR_DOMAIN_CODE   = 0x000;   # 12-bit country or domain code for intended
                                    # region-of-use (hex)
SUB_VENDOR_ID            = 0x168C;  # SubVendor ID in hex
MACID_FILENAME           = L:\ART\macid.txt;  # central macID file
CUSTOMER_DEBUG           = 0;                 # not to be turned off in normal operation
SHOW_TIMING_REPORT       = 0;                 # show time taken by various tests


# Instrumentation Info

USE_INSTRUMENTS          = 1;       # Select whether Instruments are Connected
PM_MODEL                 = 2;       # 1 => HP 436A, 2 => Agilent E4416A,
                                    # 3 => 4531
PM_GPIB_ADDR             = 14;      # Power meter GPIB Address
SA_GPIB_ADDR             = 18;      # Spectrum analyzer GPIB Address
ATT_GPIB_ADDR            = 4;       # Attenuator GPIB Address
GOLDEN_IP_ADDR           = 10.10.12.223;      # IP addr of the golden machine
GOLDEN_IP_ADDR           = 10.10.12.103;      # IP addr of the golden machine
#GOLDEN_IP_ADDR          = .;                 # Use . if both DUT and GU are controlled
                                                by the same machine
```

```
# Calibration Setup Info

CASE_TEMPERATURE            = 45;     # Case temperature of the AR5111 on the DUT
                                      # during calibration (int)
GOLDEN_PPM                  = 0;      # Measured PPM of golden device (int)
ATTEN_DUT_SA                = 14.5;   # Attenuation at 5.5 GHz between DUT and
                                      # spectrum analyzer (double)
ATTEN_FIXED_DUT_GOLDEN      = 35.5;   # Fixed attenuation (35.5 for ap30 GU) at 5.5 GHz
                                       between DUT # and GOLDEN (double)
ATTEN_DUT_PM                = 12.8;   # Attenuation at 5.5 GHz between DUT and
                                      # power meter (double)
11b_ATTEN_DUT_SA            = 12.5;   # Attenuation at 2.5 GHz between DUT and
                                      # spectrum analyzer (double)
11b_ATTEN_FIXED_DUT_GOLDEN  = 32.4;   # Fixed attenuation (32.4 for ap30 GU) at 2.5 GHz
                                        between DUT # and GOLDEN (double)
11b_ATTEN_DUT_PM            = 11.5;   # Attenuation at 2.5 GHz between DUT and
                                      # power meter (double)

# Test Margins
NUM_MASK_FAIL_POINTS        = 0;      # Manufacturing margin for spectral mask test
PER_PASS_LIMIT              = 90;     # Minimum number of good packets in Tx_PER tests
                                        to pass

SEN_PASS_LIMIT              = 90;     # Minimum number of good packets in Rx_SEN tests
                                        to pass

PPM_MAX_LIMIT               = 15;     # Maximum permitted PPM, after correcting for
                                        GOLDEN_PPM

PPM_MIN_LIMIT               = -15;    # Minimum permitted PPM, after correcting for
                                        GOLDEN_PPM

TARGET_POWER_TOLERANCE_UPP= 1;        # Tolerance in dB to allow for target power
                                      # control (double)
TARGET_POWER_TOLERANCE_LOWER= 1.5;    # Tolerance in dB to allow for target power
                                      # control (double)

MAX_RETEST_NUM              = 3;      # Maximum number of times a test is run. Set
                                        value >= 1

# Calibration flags
CAL_POWER                   = 1;      # Calibrate power levels for 11a/b/g for
                                      # modes supported
CAL_PHASE                   = 1;      # Calibrate phase delta for falcon dual chains
USE_11g_CAL_FOR_11b         = 0;      # Use 11g calibration data for 11b calibration
REPROGRAM_TARGET_POWER      = 0;      # Re-program target power. No power
                                      # measurements done
DO_IQ_CAL                   = 1;      # Perform ig_cal mismatch calibration

# Use the following FIXED_GAIN setting to disable auto FIXED_GAIN algorithm for 11a and
11g mode
# You can force FIXED_GAIN to the value between 0 (lowest) and 8 (highest)
# If you comment out the following two lines, ART program will find the best FG setting
automatically
# FIXED_GAIN setting for 11b is fixed at 4 (auto or manual)

CAL_FIXED_GAIN              = 6;      # Fixed gain to use for 11a cal measurements
11g_CAL_FIXED_GAIN          = 6;      # Fixed gain to use for 11g cal measurements
11b_CAL_FIXED_GAIN          = 6;      # Fixed gain to use for 11b cal measurements
```

```
########################################
###  calsetup specific to 11a mode  ###
########################################

GOLDEN_Tx_POWER            = 10;     # Desired output power from the GU for 11a RxSEN
                                       test (double)
MAX_POWER_CAP              = 20;     # Cap power measurements during calibration
FORCE_PIERS                = 1;      # calibrate power only at channels in the
                                       FORCE_PIERS_LIST
FORCE_PIERS_LIST           = 5170, 5220, 5280, 5320, 5400, 5500, 5650, 5725, 5825; # a
                                      maximum of up to 10 piers.
READ_FROM_FILE             = 0;      # skip 11a cal. read data from a file instead.
RAW_DATA_FILENAME          = cal_AR5211_power.log;     # read 11a data from this file

# AUTO_PIERS:     When FORCE_PIERS is off, set the following three parameters
#                  to allow ART to automatically find the turning points over the
#                  desired range.

#AUTO_PIER_START            = 5150;   # (default 4900 if not specified)
#AUTO_PIER_STOP             = 5500;   # (default 5850 if not specified)
#AUTO_PIER_STEP             = 30;     # (default 70 if not specified) keep between 10-
                                       100 MHz

TEST_BOTH_CHAINS             = 1;      # Repeat tests for both chains for Falcon

# 11a mode test flags
TEST_TEMP_MARGIN            = 1;      # Test for target power control across 0-85 C at
                                       11a test frequencies.
TEST_OBW_MASK                = 1;      # Test occupied bandwidth mask at 11a test
                                       frequencies
TEST_SPEC_MASK               = 1;      # Test spectral mask at 11a test frequencies
TEST_TxPER                   = 1;      # Test Transmit Packet Error Rate at 11a test
                                       frequencies
TEST_RxSEN                   = 1;      # Test Receive Sensitivity at 11a test
                                        frequencies
TEST_TURBO_MODE              = 1;      # Test PER and RxSEN in the turbo mode at 11a
                                       test frequencies
TEST_HALF_RATE_MODE          = 0;      # Test PER and RxSEN in the half rate mode at
                                       11a test frequencies
TEST_TARGET_POWER            = 1;      # Test Target Power at 11a test frequencies

# 11a, 11b and 11g Test Channel Matrices:
#        Rules:
#               1. Up to 32 test frequencies for each mode.
#               2. If flags are turned on above to perform the TEMP_MARGIN_TEST,
                  spectral mask, TxPER, RxSEN and target power tests for 11a/b/g modes,
                  they will be performed at the following test frequencies based upon
                  the values in the PER_TEST, MASK_TEST, TGT_PWR_TEST and
                  TEMP_MARGIN_TEST columns.
#               3. OBW_TEST can be specified ONLY for 11a mode.
#
```

```
#BEGIN_11a_TEST_CHANNEL_MATRIX
#test_frequencies  PER/   SEN    LO_RATE  LO_RATE   TURBO   HALF_   MASK   OBW    TGT_   TEMP_
#                  SEN    TGT    SEN      SEN_TGT   PER/    RATE    TEST   TEST   PWR    MARGIN
#                  TEST                             SEN     PER/                  TEST   TEST
#                                                           SEN
UNII-1&2 (5.15 - 5.35 GHz) test channels
5180               1      -68    0        -88       0       0       1      0      1111   1
5200               1      -68    0        -88       0       0       1      0      1111   0
5220               1      -68    0        -88       0       0       1      0      1111   0
5240               1      -68    0        -88       0       0       1      0      1111   0
5260               1      -68    0        -88       0       0       0      0      1111   1
5280               1      -68    0        -88       0       0       1      0      1111   0
5300               1      -68    0        -88       0       0       1      0      1111   0
5320               1      -68    0        -88       0       0       1      0      1111   1

# FCC turbo test channels
5210               0      -68    0        -88       1       0       0      0      0000   0
5250               0      -68    0        -88       1       0       0      0      0000   0
5290               0      -68    0        -88       1       0       0      0      0000   0

# Dynamic turbo test channels
5200               0      -68    0        -88       1       0       0      0      0000   0
5240               0      -68    0        -88       1       0       0      0      0000   0
5280               0      -68    0        -88       1       0       0      0      0000   0
5765               0      -68    0        -88       1       0       0      0      0000   0
5805               0      -68    0        -88       1       0       0      0      0000   0
# UNII-3 & ISM band test channels
5745               1      -68    0        -88       0       0       0      0      1111   1
5765               1      -68    0        -88       0       0       0      0      1111   0
5785               0      -68    0        -88       1       0       0      0      0000   0
5805               1      -68    0        -88       0       0       0      0      1111   1
5825               1      -68    0        -88       0       0       0      0      1111   1

# ETSI (5.47-5.7 GHz) test channels
5500               1      -68    0        -88       0       0       0      0      1111   0
5520               1      -68    0        -88       0       0       0      0      1111   0
```

```
#test_frequencies PER/  SEN  LO_RATE LO_RATE TURBO HALF_ MASK OBW  TGT_ TEMP_
#                 SEN   TGT  SEN     SEN_TGT PER/  RATE  TEST TEST PWR  MARGIN
#                 TEST                       SEN   PER/            TEST TEST
#                                                  SEN

UNII-1&2 (5.15 - 5.35 GHz) test channels

5540              1     -68  0       -88     0     0     0    0    1111 0

5560              1     -68  0       -88     0     0     0    0    1111 0

5580              1     -68  0       -88     0     0     0    0    1111 0

5600              1     -68  0       -88     0     0     0    0    1111 0

5620              1     -68  0       -88     0     0     0    0    1111 0

5640              1     -68  0       -88     0     0     0    0    1111 0

5660              1     -68  0       -88     0     0     0    0    1111 0

5680              1     -68  0       -88     0     0     0    0    1111 0

5700              1     -68  0       -88     0     0     0    0    1111 0


# MKK (5.15-5.25 GHz) test channels

5170              0     -68  0       -88     0     0     0    1    1111 0

5190              0     -68  0       -88     0     0     0    1    1111 0

5210              0     -68  0       -88     0     0     0    1    1111 0

5230              0     -68  0       -88     0     0     0    1    1111 0


# MKK (4.9-5.1 GHz) test channels

4900              0     -68  0       -88     0     0     0    1    1111 0

4920              0     -68  0       -88     0     0     0    1    1111 0

4940              0     -68  0       -88     0     0     0    1    1111 0

4960              0     -68  0       -88     0     0     0    1    1111 0

4980              0     -68  0       -88     0     0     0    1    1111 0

5040              0     -68  0       -88     0     0     0    1    1111 0

5060              0     -68  0       -88     0     0     0    1    1111 0

5080              0     -68  0       -88     0     0     0    1    1111 0


#END_11a_TEST_CHANNEL_MATRIX
```

```
########################################
###  calsetup specific to 11b mode  ###
########################################
11b_GOLDEN_Tx_POWER           = 18;     # Desired output power from the GU for 11b RxSEN
                                         test (double)
11b_MAX_POWER_CAP              = 20;     # Cap power measurements during calibration
FORCE_PIERS_11b                = 1;      # calibrate power only at channels in the
                                         FORCE_PIERS_LIST_11b
FORCE_PIERS_LIST_11b           = 2412, 2472, 2484;   # a maximum of up to 3 piers.
11b_READ_FROM_FILE             = 0;      # skip 11b cal. read data from a file instead.
11b_RAW_DATA_FILENAME          = cal_AR5211_power_11b.log;    # read 11b data from this
                                                               file

# 11b mode test flags
11b_TEST_TEMP_MARGIN          = 1;      # Test for target power control across 0-85 C at
                                         11b test frequencies.
11b_TEST_SPEC_MASK             = 1;      # Test spectral mask at 11b test frequencies
11b_TEST_TxPER                 = 1;      # Test Transmit Packet Error Rate at 11b test
                                         frequencies
11b_TEST_RxSEN                 = 1;      # Test Receive Sensitivity at 11b test
                                         frequencies
11b_TEST_TARGET_POWER          = 1;      # Test Target Power at 11b test frequencies

#BEGIN_11b_TEST_CHANNEL_MATRIX

 #test_frequencies  PER/   SEN    LO_RATE  LO_RATE  MASK   TGT_   TEMP_
 #                  SEN    TGT    SEN      SEN_TGT  TEST   PWR    MARGIN
 #                  TEST                                   TEST   TEST
 #

 2412                 1     -90     1       -95       1    1111    1

 2417                 1     -90     1       -95       0    1111    0

 2422                 1     -90     1       -95       0    1111    0

 2427                 1     -90     1       -95       0    1111    0

 2432                 1     -90     1       -95       0    1111    0

 2437                 1     -90     1       -95       1    1111    0

 2442                 0     -90     1       -95       1    0000    0

 2447                 1     -90     1       -95       0    1111    0

 2452                 1     -90     1       -95       0    1111    0

 2457                 1     -90     1       -95       0    1111    0

 2462                 1     -90     1       -95       1    1111    0

 2467                 1     -90     1       -95       0    1111    0

 2472                 1     -90     1       -95       0    1111    0

 2484                 1     -90     1       -95       0    1111    1


#END_11b_TEST_CHANNEL_MATRIX
```

```
#######################################
###  calsetup specific to 11g mode  ###
#######################################
11g_GOLDEN_Tx_POWER           = 10;     # Desired output power from the GU for 11g RxSEN
                                         test (double)
11g_MAX_POWER_CAP             = 20;     # Cap power measurements during calibration
FORCE_PIERS_11g               = 1;      # calibrate power only at channels in the
                                         FORCE_PIERS_LIST
FORCE_PIERS_LIST_11g          = 2412, 2472, 2484;   # a maximum of up to 3 piers.
11g_READ_FROM_FILE            = 0;      # skip 11g cal. read data from a file instead.
11g_RAW_DATA_FILENAME         = cal_AR5211_power_11g.log;    # read 11g data from this
                                                              file.

# 11g mode test flags
11g_TEST_TEMP_MARGIN          = 1;      # Test for target power control across 0-85C at
                                         11g test frequencies.
11g_TEST_SPEC_MASK            = 1;      # Test spectral mask at 11g test frequencies.
11g_TEST_TxPER                = 1;      # Test Transmit Packet Error Rate at 11g test
                                         frequencies.
11g_TEST_RxSEN                = 1;      # Test Receive Sensitivity at 11g test
                                         frequencies.
11g_TEST_TARGET_POWER         = 1;      # Test Target Power at 11g test frequencies.
11g_TEST_TURBO_MODE           = 1;      # Test PER and RxSEN in the turbo mode at 11g
                                         test frequencies

#BEGIN_11g_TEST_CHANNEL_MATRIX
 #test_frequencies  PER/  SEN   LO_RATE LO_RATE  TURBO    MASK  TGT_  TEMP_
 #                  SEN   TGT   SEN     SEN_TGT  PER/SEN  TEST  PWR   MARGIN
 #                  TEST                                        TEST  TEST
 #

 2412                 1    -72     1       -90       0       1    1111    1

 2417                 1    -72     1       -90       0       0    1111    0

 2422                 1    -72     1       -90       0       0    1111    0

 2427                 1    -72     1       -90       0       0    1111    0

 2432                 1    -72     1       -90       0       0    1111    0

 2437                 1    -72     1       -90       1       1    1111    0

 2442                 0    -72     1       -90       0       0    1111    0

 2447                 1    -72     1       -90       0       0    1111    0

 2452                 1    -72     1       -90       0       0    1111    0

 2457                 1    -72     1       -90       0       0    1111    0

 2462                 1    -72     1       -90       0       1    1111    0

 2467                 1    -72     1       -90       0       0    1111    0

 2472                 1    -72     1       -90       0       0    1111    0

 2484                 1    -72     1       -90       0       1    1111    1


#END_11g_TEST_CHANNEL_MATRIX
```

B

Sample File calTargetPower_ar5001a_cb.txt

This appendix provides a sample of the **calTargetPower_ar5001a_cb.txt** file.

```
# target power file for ar5001a_cb card
# 11a Target Power table:
#   Rules:
#      1. up to a maximum of 8 test frequencies
#      2. test frequencies DO NOT need to cover the entire range of
#         5180-5850. It is allowed to provide data for a smaller
#         range. for all channels outside of test frequencies range,
#         target power will be assumed 0dB.
#      3. specify mask/PER limited target power for various rates
#
#BEGIN_11a_TARGET_POWER_TABLE

 #test_frequencies   6-24_target   36_target    48_target    54_target
        5170             18            17           16           14
        5240             18            17           16           14
        5320             18            17           16           14
        5500             18            16           15           12
        5700             18            16           15           12
        5745             18            16           13           10
        5850             18            16           13           10
```

```
#END_11a_TARGET_POWER_TABLE
# Test Groups:
#   Rules:
#      1. Specify up to 8 band edges for each test group.
#      2. If no backoff desired at a band edge, give a large number
#          (e.g, 30) so that the target power becomes the driver
#          determined limit.
#


#BEGIN_TEST_GROUPS
# Test Group 1: US and CANADA (FCC) (base mode 802.11a)
#BEGIN_11a_TARGET_POWER_TABLE

 # test_group_code   BE1   BE2   BE3   BE4   BE5   BE6   BE7   BE8
        0x10         5180  5200  5320  5745  5765  5785  5805  5825
                     14    14    14.5  16.5  16.5  16.5  16.5  16.5  # Band Edge Max Power
                     0     0     0     0     0     0     0     0     # in-band flag


# Test Group 2: US and CANADA (FCC) (turbo mode 802.11a)

 # test_group_code   BE1   BE2   BE3   BE4   BE5
        0x13         5210  5250  5290  5760  5800
                     14.5  14.5  15    15.5  15.5  # Band Edge Max Power
                     0     0     0     0     0     # in-band flag


# Test Group 3: JAPAN (MKK)
 # test_group_code   BE1   BE2   BE3   BE4   BE5   BE6   BE7   BE8
        0x40         5170  5230
                     30    30    # Band Edge Max Power
                     0     0     # in-band flag


# Test Group 4: EUROPE (ETSI)
 # test_group_code   BE1   BE2   BE3   BE4   BE5   BE6   BE7   BE8
 0x30                5180  5240  5260  5320  5500  5700  5725  5825
                     30    30    30    30    30    30    30    30    # Band Edge Max Power
                     0     0     0     0     0     0     0     0     # in-band flag


#END_TEST_GROUPS
```

C

Sample .eep File ar5005x_cb.eep

This appendix contains a sample of the **ar5005x_cb.eep** file.

```
#EEPROM file for AR5513 AV10

@cal_section_begin                        # begin @cal section

TARGET_POWER_FILENAME      =  calTrgtPwr_ar5005x_av10.txt; # target power file for
calibration
SUBSYSTEM_ID               =  0x2056;   # Subsystem ID in hex
EEPROM_MAP_TYPE            =  1;        # Flag indicating eeprom layout type
TURBO_DISABLE              =  0;        # Prevents software from using TURBO modes
RF_SILENT                  =  0;        # Cards enabled with RFSilent can be easily silenced
DEVICE_TYPE                =  1;        # 1=>Cardbus, 2=>PCI, 3=>miniPCI, 4=>AP
TURBO_MAXPOWER_5G          =  16.0;     # Recommended max power in turbo mode to consume
                                      less than 2W.
TURBO_MAXPOWER_2p5G        =  16.0;     # Recommended max power in turbo mode at 2.5 GHz
                                      to consume less than 2W.
A_MODE                     =  1;        # Whether the adapter supports 802.11a functionality
B_MODE                     =  1;        # Whether the adapter supports 802.11b functionality
G_MODE                     =  1;        # Whether the adapter supports 802.11g functionality
ANTENNA_GAIN_5G            =  2;        # Antenna gain at 5GHz. 8-bit signed val in 0.5dB
                                      steps.
ANTENNA_GAIN_2p5G          =  2;        # Antenna gain at 2.5GHz. 8-bit signed val in 0.5dB
                                      steps.
XLNA_GAIN                  =  13;       # xLNA gain in dB (per AS 6/14/02)
NOISE_FLOOR_THRESHOLD      = -54;       # noise floor threshold value
11b_XLNA_GAIN              =  13;       # xLNA gain in dB
11b_NOISE_FLOOR_THRESHOLD  = -1;        # noise floor threshold value
11g_XLNA_GAIN              =  13;       # xLNA gain in dB
11g_NOISE_FLOOR_THRESHOLD  = -1;        # noise floor threshold value
11a_FALSE_DETECT_BACKOFF   =  0;        # in dB. only affects channels w/ clock harmonic
                                      overlap.
11b_FALSE_DETECT_BACKOFF   =  0;        # in dB. only affects channels w/ clock harmonic
                                      overlap.
11g_FALSE_DETECT_BACKOFF   =  0;        # in dB. only affects channels w/ clock harmonic
                                      overlap.
11g_TURBO_DISABLE          =  0;        # Set to 1 to disable TURBO mode in 11g
CCK_OFDM_DELTA             =  1;        # in dB with 0.1dB resolution. In 11g, delta in
                                      output power for 1mbps and 6mbps
                                     # for any given pcdac.
```

```
CH14_FILTER_CCK_DELTA      =  1.5;      # in dB with 0.1dB resolution. In 11g & 11b, delta in
                                      output power for ch14 and ch1 - ch13
# for any given pcdac. This delta arises due to
                                      special filter requirement for ch14 (2484).
ENABLE_32KHZ               =  0;        # Flag indicating the presence of the 32kHz sleep
                                      crystal

Tx_CHAIN_MASK              =  3;        # mask indicating which chains are populated for Tx
                                      (0x1-->chain0, 0x2-->chain1)
Rx_CHAIN_MASK              =  3;        # mask indicating which chains are populated for Rx
                                      (0x1-->chain0, 0x2-->chain1)
MBAR_MASK                  =  0;        # mask to hide mbars : 0-->none, 1-->hide mbar1,
                                        # 2-->hide mbar2, 3-->hide mbar1 and mbar2
PCI_CONFIG_BASE_OFFSET     =  0x800     # base offset for pci configuration

USE_EEPROM                 =  0;        # whether design has a flash (0) or eeprom (1)

@cal_section_end                        # end @cal section

@config_section_begin                   # begin @config section

#Antenna Switch Table
#6 bit (msb:lsb) value are mapped to [atten5, atten2, antD, antC, antB, antA]
#----------------------------------------------------------------------------------
@MODE: MODE_SPECIFIC             11a     11a_turbo      11b        11g    11g_turbo
#----------------------------------------------------------------------------------
bb_switch_table_r1x12           0x00        0x00        0x01       0x01     0x01 #(AntCtl 5)
bb_switch_table_r1x2            0x00        0x00        0x01       0x01     0x01 #(AntCtl 4)
bb_switch_table_r1x1            0x00        0x00        0x01       0x01     0x01 #(AntCtl 3)
bb_switch_table_r1              0x05        0x05        0x09       0x09     0x09 #(AntCtl 2)
bb_switch_table_t1              0x02        0x02        0x02       0x02     0x02 #(AntCtl 1)

bb_switch_table_r2x12           0x00        0x00        0x02       0x02     0x02 #(AntCtl 10)
bb_switch_table_r2x2            0x00        0x00        0x02       0x02     0x02 #(AntCtl 9)
bb_switch_table_r2x1            0x00        0x00        0x02       0x02     0x02 #(AntCtl 8)
bb_switch_table_r2              0x06        0x06        0x0A       0x0A     0x0A #(AntCtl 7)
bb_switch_table_t2              0x01        0x01        0x01       0x01     0x01 #(AntCtl 6)

chn1_bb_switch_table_r1x12      0x00     0x00     0x01     0x01     0x01 #(AntCtl 5)
chn1_bb_switch_table_r1x2       0x00     0x00     0x01     0x01     0x01 #(AntCtl 4)
chn1_bb_switch_table_r1x1       0x00     0x00     0x01     0x01     0x01 #(AntCtl 3)
chn1_bb_switch_table_r1         0x05     0x05     0x09     0x09     0x09 #(AntCtl 2)
chn1_bb_switch_table_t1         0x02     0x02     0x02     0x02     0x02 #(AntCtl 1)
chn1_bb_switch_table_r2x12      0x00     0x00     0x02     0x02     0x02 #(AntCtl 10)
chn1_bb_switch_table_r2x2       0x00     0x00     0x02     0x02     0x02 #(AntCtl 9)
chn1_bb_switch_table_r2x1       0x00     0x00     0x02     0x02     0x02 #(AntCtl 8)
chn1_bb_switch_table_r2         0x06     0x06     0x0A     0x0A     0x0A #(AntCtl 7)
chn1_bb_switch_table_t2         0x01     0x01     0x01     0x01     0x01 #(AntCtl 6)

bb_rxtx_margin_2ghz                0           0          23         23         23
bb_switch_settling              0x2d        0x5a        0x23       0x31       0x31
bb_txrxatten                      15          15          38         38         38
bb_pga_desired_size              -80         -80         -80        -80        -80
bb_adc_desired_size              -32         -32         -38        -32        -32
rf_ob                              2           2           2          2          2
rf_db                              2           2           2          2          2
rf_b_ob                            2           2           2          2          2
rf_b_db                            2           2           2          2          2
rf_xpdsel                          1           1           1          1          1
rf_pdgain_lo                     0x0         0x0         0x0        0x0       0x0
#rf_pdgain_hi                    0x0         0x0         0x0        0x0       0x0
rf_pdgain_hi                     0x3         0x3         0x3        0x3       0x3
bb_thresh62                       15          15          28         28         28
bb_tx_end_to_xpab_off              0           0           0          0          0
bb_tx_end_to_xpaa_off              0           0           0          0          0
bb_tx_frame_to_xpab_on          0x0e         0xe         0x7        0xe        0xe
bb_tx_frame_to_xpaa_on          0x0e         0xe         0x7        0xe        0xe
bb_tx_end_to_xlna_on               2           2           2          2          2

rf_gain_I                       0x0a        0x0a        0x0a       0x0a       0x0a

@config_section_end                 # end @config section
```

Atheros Communications Incorporated
529 Almanor Avenue
Sunnyvale, CA 94086
t: 408/773-5200
f: 408/773-9940
www.atheros.com