AR4100(P)

Manufacturing and Regulatory Test Solutions

80-Y2220-1 Rev. A

January 30, 2013

Qualcomm Atheros, Inc.
1700 Technology Drive
San Jose, CA 95110
U.S.A.

Revision history

Revision	Date	Description
A	January 2013	Initial release

Contents

Figures

1 Overview

This document summarizes the set of tools available that automate manufacturing test and configuration for Internet of Everything (IoE) products based on the AR4100(P) release 2.1. These tools include RF calibration, verification, the regulatory testing, and FCC certification.

NOTE: This document covers both the AR4100 and the AR4100P products. The devices are functionally equivalent in many ways; where the information is common, the devices are called out as AR4100(P) to cover both products. Otherwise, the product is specifically called out as AR4100 or AR4100P to refer to a particular product.

This document describes:

- Manufacturing test setup, including:
  - MAC address programming
    This tool is run from the application driver through a WMI interface to facilitate the MAC programming process
  - Board data merge tool
    This tool enables customers to use their customized board data file to replace the board data in the AR4100(P) firmware flash binary
- Required hardware/software tools
- Tx calibration and verification procedures required for the AR4100(P) to use the LitePoint IQ2010 multi communication tester

2 Manufacturing Test Setup

2.1 Hardware

The manufacturing test setup requires this hardware (see Figure 2-1):

- Device under test (DUT)
  Based on AR4100(P), a microcontroller (MCU), and Flash
  - SPI bus between the MCU (host) and the AR4100(P)
  - SPI bus between the AR4100(P) and the Flash
- Host PC based on Windows 7
  - Connected to the DUT via UART (RS232)-based communications
  - Power-up DUT
- LitePoint IQ2010 (LP) / power meter(Agilent E4405B Power Meter)
  - RF cable between the LP/power meter Wi-Fi interface connector and DUT/AR4100(P)
  - LP-to-host PC connection via USB2.0 or Ethernet, dependent on the LP model version

NOTE: RF cable loss should be measured in different frequencies depending on calibration/verification test frequencies. These losses should be documented for configuring the test procedures.

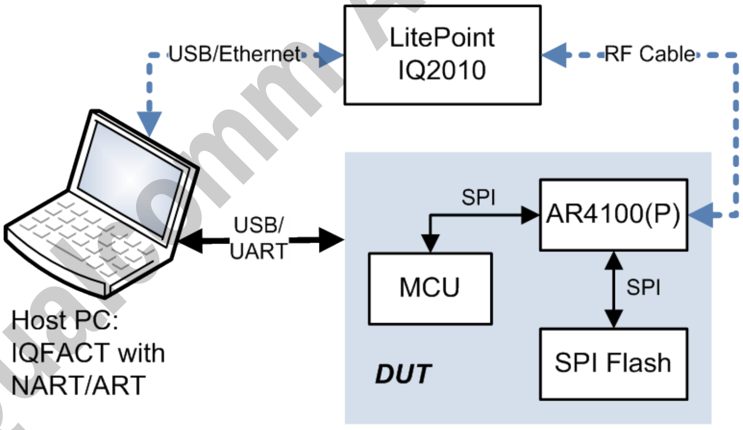


Figure 2-1 Manufacturing Test Setup

2.2 Software

2.2.1 DUT

The DUT block consists of the software modules:

- **AR4100(P)**
  Firmware image burned onto the SPI flash (**flashotp.bin,raw_flashimage.bin**)
  - **flashotp.bin**
    This firmware binary suite combines several firmware binaries. It selects firmware to load based on the application programmed on the MCU and executes from SPI flash memory.
  - **raw_flashimage.bin**
    The AR4100(P) raw flash image of the firmware can be used with a flash programmer instead of using the standard flash programming tools available for the AR4100(P). It selects firmware to load based on the application programmed on the MCU and executes from SPI flash memory.
- **MCU**
  Programmed with flash agent/UTF agent/throughput demo based on the test requirement

2.2.2 Host PC

The host PC connected to the DUT must have this software installed:

- MCU-specific flash tool (Energy Aware Commander for EFM SP137, Code Warrier 10.2 for K60)
- Necessary driver for respective platforms (SP137, K60)
  See the *Internet of Everything Wireless System Release (I-WSR) 2.1 Release Notes* for more information.
- Software release binary package
- ART / NART package distributed by Qualcomm Atheros
- IQFACT software package customized for the AR4100(P) with the LitePoint driver installed

NOTE: ART/NART/IQFACT Software packages will be available from http://support.atheros.com.

2.2.3 UTF agent and UTF firmware binary

The UTF agent is a driver software running on the MCU that enables the ART PC tool running on the host PC to talk to the UTF firmware binary running on the AR4100(P). The UTF binary is a test binary used by the Qualcomm Atheros test tool for RF Tx/Rx calibration and verification.

The UTF agent mediates the communication between Host PC ART/NART tool and the UTF firmware.

3 Manufacturing Flow Process

This section discusses the manufacturing flow for two variants of the AR4100(P) Wi-Fi chip, where the AR4100P includes premium features that the AR4100 does not have. The AR4100P is preconfigured with NVRAM configuration and one-time programmable (OTP) memory contents. The AR4100 requires additional procedures before manufacturing test.

Figure 3-1 shows the manufacturing test flow for the AR4100 and AR4100P. The flow uses the setup described in section 2.

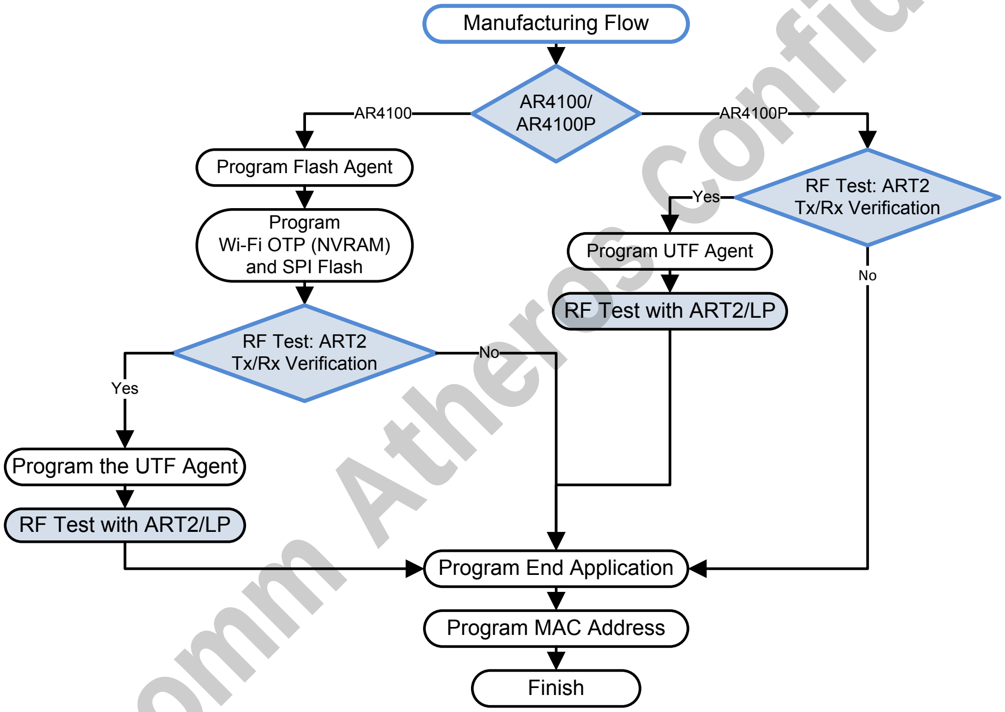


Figure 3-1 Manufacturing Flow

NOTE: It is strongly recommended that for volume manufacturing, SPI flash devices are pre-programmed with the raw SPI flash image before mounting to minimize the flow time.

Manufacturing flow RF tests can be run from the ART command line tool or using the IQFact Litepoint software solution. Customers wishing to validate Tx power using power meters other than LitePoint can connect the power meter and use the ART command line tool to run Tx test cases and measure Tx power. For more extensive testing, customers can use the setup shown in Figure 3-1 and run IQFact software, which in-turn uses NART to run multiple RF test suites.

For more information on the process, Tx calibration, and handling the RF Test with ART see:

- Tx/Rx Verification using ART/NART
- Manufacturing With the SP137 Platform

3.1 Programming flash agent

Programming the flash agent on MCU will allow it to communicate to the host PC for NVRAM and OTP Programming on the AR4100. The process and the software tools for upgrading the firmware of AR4100(P) are documented in the *Internet of Things Wireless System Release (I-WSR) 2.1 Release Notes*.

3.2 OTP and NVRAM configuration

The *Programming the SP137 OTP/Flash Application Note* details the host PC tool (**host.exe**) and its options used for programming the OTP and NVRAM configuration.

- **Host.exe** takes the firmware COM port, firmware path, and MAC address as command line options. This tool looks for specified path for **flashotp.bin** firmware, and the MAC address format uses 12 hex digits.
- The flash OTP programming tool combines the MAC address and the firmware image and sends the resulting image over the UART to the host MCU, which sends it over the SPI bus to the target.
- The firmware image is executed and proceeds to write flash and OTP.
- Firmware messages back to the host MCU upon completion, which relays the message to the PC application.

NOTE: The MAC address can be programmed as mentioned in MAC Address Programming with a WMI interface command. Customers programming the MAC in this stage can skip the MAC Address Programming section.

3.3 Programming the UTF agent on the MCU

The *Internet of Things Wireless System Release (I-WSR) 2.1 Release Notes* discuss downloading binaries to the MCU. Please refer to the respective MCU variant on how to flash binaries to the MCU. The UTF agent is flashed to MCU in the same way as any other demo application.

Note that **flashotp.bin** in the SPI flash and the UTF Agent in the MCU is a required combination to start any RF test with the ART PC tool.

4 Tx/Rx Verification using ART/NART

The Qualcomm Atheros ART radio test is used for radio evaluation and manufacturing tests. It performs various transmissions, receives, and link tests, and calibrates during manufacturing flow. NART is an evaluation and manufacturing test tool operating with LitePoint and Agilent testers in production. For NART, the server listens at a configured TCP port and allows communications from test application software (such as IQFACT and Agilent), as well as enabling the GUI.

NOTE: The AR4100(P) does not have an EEPROM device; instead it uses OTP memory to store chip-specific configuration information, most notably power calibration data. Given the small OTP size (512 B), all non-chip-specific but design-specific settings are stored in a host file (the board data file). In this document, this board data file and OTP are referred to as EEPROM.

NOTE: Running ART also creates the board data file. Please see flashotp.bin Golden Board Data for information on replacing AR4100(P) firmware board data with withis the customer board data.

4.1 Configuration Files

The software kit includes a host PC ART tool for Windows XP and Windows 7 that provides an ART and NART configuration file and contains default configuration parameters for the AR4100(P) SIP. These external environment configuration files are updated as required for calibration and verification:

- **artsetup.txt**
- **calsetup.txt**
- **remoteIP.ini**
- **ar*.eep**

Artsetup.txt must check for these values before processing:

Value	Setting		
EEPROM_LOAD_OVERRIDE	Set to 1 to override loading EEPROM calibration info		
EEPROM_PRESENT	0	Flag indicating the EEPROM on the board	
EEPROM_FILE	**./SP137-010-EEPROM.bin**		
OTP_MANUFCAL_OVER_EEPROM	1	Enabled	
OTP_ATECAL_OVER_EEPROM	0	Disabled	
OTP_ATETHERM_OVER_EEPROM	0	Disabled	
EEP_FILE_DIR	**..\..\config**		
DUT_CARD_SSID	0x6309		
DEVDRV_INTERFACE	3	1	SDIO
		2	ETH
		3	UART (recommended setting)
ENABLE_LABEL_SCHEME	0	Enable the Qualcomm Atheros-specific label scheme	
CFG_TABLE	0x6309	**Ar6003nx_9.eep** (SP137)	

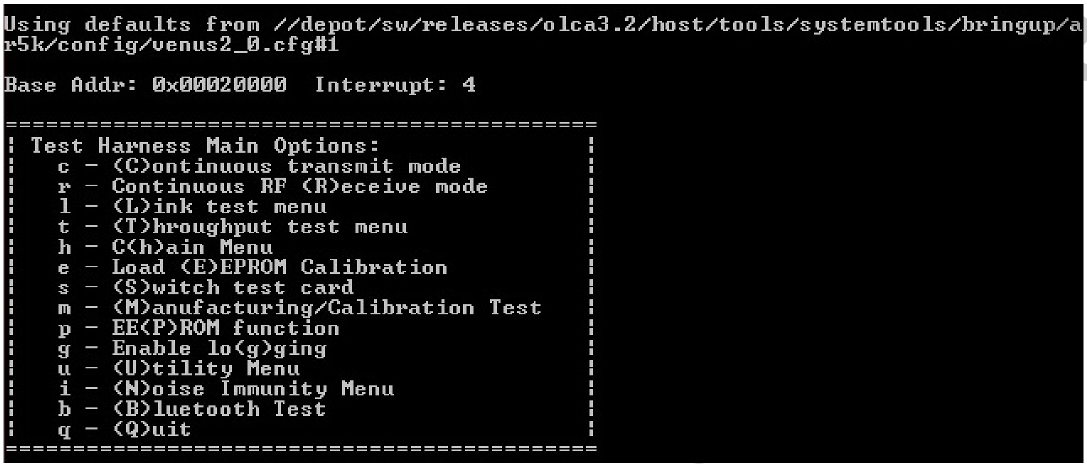
RemoteIP.ini must be updated with the correct UART settings:

- UART_PORT = COMx
- UART_SETTING = 115200, n, 8, 1

On startup, ART reads the subsystem ID from the EEPROM then looks for a match to this ID in the CFG_TABLE entries in the file **artsetup.txt**. If a match is found, the EEPROM parses the corresponding .eep file and applies it to the adapter. If it reads a subsystem ID from the EEPROM, but finds no match in the CFG_TABLE entries, ART exits with an error.

4.2 ART commands

Running the executable **art.exe** opens a menu of options to run a test.

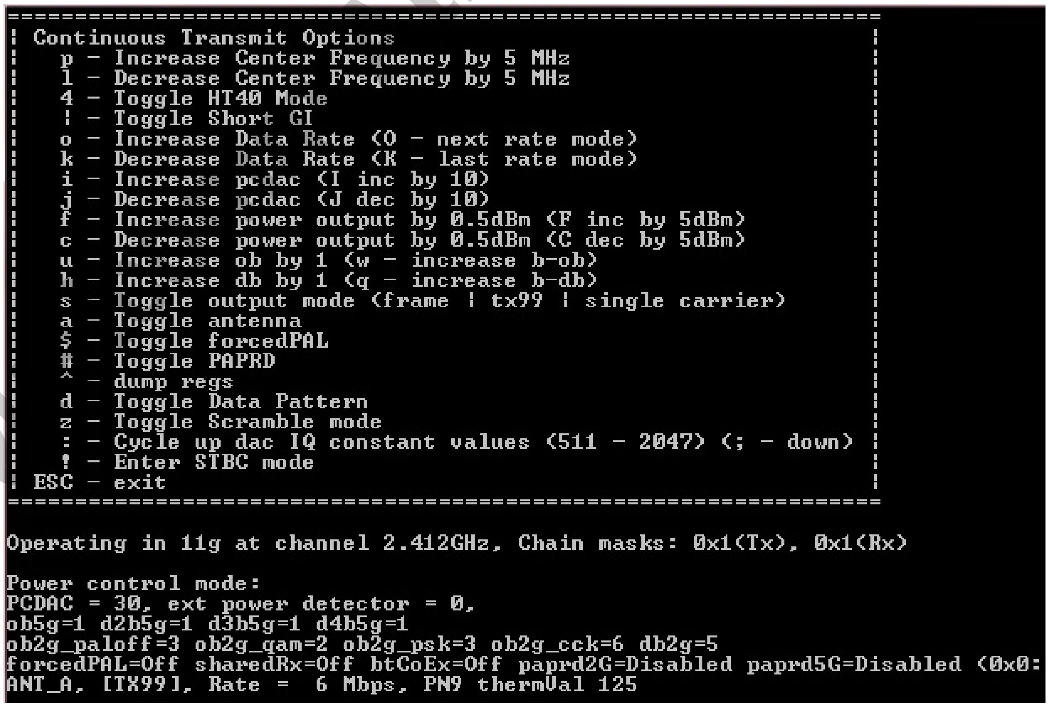


```
Using defaults from //depot/sw/releases/olca3.2/host/tools/systemtools/bringup/a
r5k/config/venus2_0.cfg#1

Base Addr: 0x00020000  Interrupt: 4

============================================
| Test Harness Main Options:               |
|   c - (C)ontinuous transmit mode         |
|   r - Continuous RF (R)eceive mode       |
|   l - (L)ink test menu                   |
|   t - (T)hroughput test menu             |
|   h - C(h)ain Menu                       |
|   e - Load (E)EPROM Calibration          |
|   s - (S)witch test card                 |
|   m - (M)anufacturing/Calibration Test   |
|   p - EE(P)ROM function                  |
|   g - Enable lo(g)ging                   |
|   u - (U)tility Menu                     |
|   i - (N)oise Immunity Menu              |
|   b - (B)luetooth Test                   |
|   q - (Q)uit                             |
============================================
```

Figure 4-1 ART Menu on the Command Line

4.2.1 Continuous transmit mode (c)

Continuous transmit (Tx) mode allows dynamic change of the channel frequency, data rate, and a number of radio characteristics. They also allow switching between single-carrier mode and 99 percent, 100 percent, or dynamic duty cycle data mode.



```
================================================================
| Continuous Transmit Options                                  |
|   p - Increase Center Frequency by 5 MHz                     |
|   l - Decrease Center Frequency by 5 MHz                     |
|   4 - Toggle HT40 Mode                                       |
|   | - Toggle Short GI                                        |
|   o - Increase Data Rate (O - next rate mode)                |
|   k - Decrease Data Rate (K - last rate mode)                |
|   i - Increase pcdac (I inc by 10)                           |
|   j - Decrease pcdac (J dec by 10)                           |
|   f - Increase power output by 0.5dBm (F inc by 5dBm)        |
|   c - Decrease power output by 0.5dBm (C dec by 5dBm)        |
|   u - Increase ob by 1 (w - increase b-ob)                   |
|   h - Increase db by 1 (q - increase b-db)                   |
|   s - Toggle output mode (frame | tx99 | single carrier)     |
|   a - Toggle antenna                                         |
|   $ - Toggle forcedPAL                                       |
|   # - Toggle PAPRD                                           |
|   ^ - dump regs                                              |
|   d - Toggle Data Pattern                                    |
|   z - Toggle Scramble mode                                   |
|   : - Cycle up dac IQ constant values (511 - 2047) (; - down) |
|   ! - Enter STBC mode                                        |
| ESC - exit                                                   |
================================================================

Operating in 11g at channel 2.412GHz, Chain masks: 0x1(Tx), 0x1(Rx)

Power control mode:
PCDAC = 30, ext power detector = 0,
ob5g=1 d2b5g=1 d3b5g=1 d4b5g=1
ob2g_paloff=3 ob2g_qam=2 ob2g_psk=3 ob2g_cck=6 db2g=5
forcedPAL=Off sharedRx=Off btCoEx=Off paprd2G=Disabled paprd5G=Disabled (0x0:
ANT_A, [TX99], Rate =  6 Mbps, PN9 thermVal 125
```

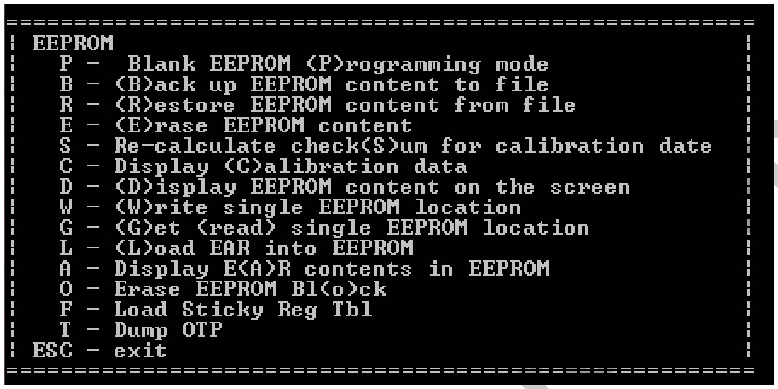
Figure 4-2 Continuous Tx Options with Tx Started on Default Channel and Rate

4.2.2 Power control

To obtain accurate power control within 0.5 dBm, ensure that the EEPROM calibration information has been loaded from the main menu, then use the **(c)** and (**f**) options from this menu. Once selected, the power value shown by the software is the measurable power output by the reference design. The AR4100(P) only supports open loop power control. The desired gain is calculated from the target power and the gain correction value, which is obtained during calibration.

4.2.3 EEPROM function (p)

This option can back up to file, read from a file, erase, or display EEPROM contents as either a byte stream or formatted information. It can also load an EAR file into EERPROM, or read and display an existing EAR file.

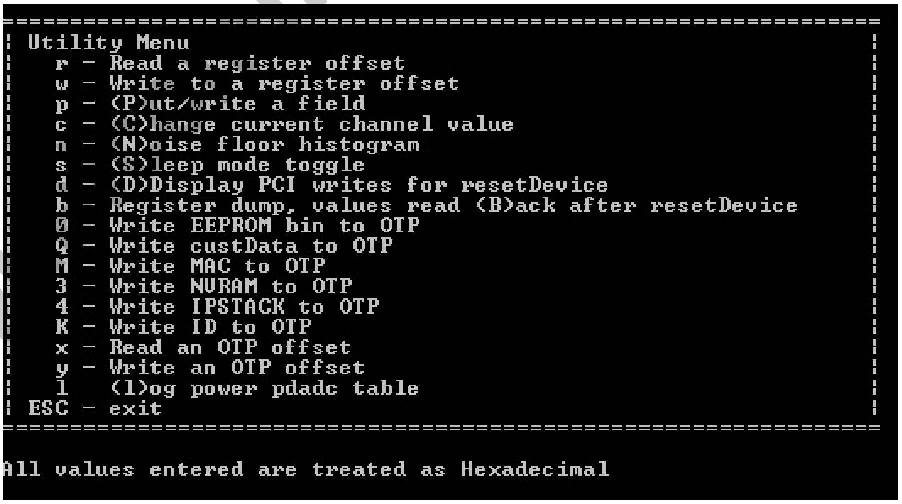


```
=======================================================
¦ EEPROM                                              ¦
¦   P -  Blank EEPROM <P>rogramming mode              ¦
¦   B - <B>ack up EEPROM content to file              ¦
¦   R - <R>estore EEPROM content from file            ¦
¦   E - <E>rase EEPROM content                        ¦
¦   S - Re-calculate check<S>um for calibration date  ¦
¦   C - Display <C>alibration data                    ¦
¦   D - <D>isplay EEPROM content on the screen        ¦
¦   W - <W>rite single EEPROM location                ¦
¦   G - <G>et (read) single EEPROM location           ¦
¦   L - <L>oad EAR into EEPROM                        ¦
¦   A - Display E<A>R contents in EEPROM              ¦
¦   O - Erase EEPROM Bl<o>ck                          ¦
¦   F - Load Sticky Reg Tbl                           ¦
¦   T - Dump OTP                                      ¦
¦ ESC - exit                                          ¦
=======================================================
```

Figure 4-3 EEPROM Function Menu

4.2.4 Utility Menu (u)

The utility menu provides some miscellaneous utilities, including the ability to read and write single registers, to write register fields, to tune to a new channel, to display a noise-floor histogram of channel noise, and to put the adapter into sleep mode.



```
=================================================================
¦ Utility Menu                                                  ¦
¦   r - Read a register offset                                  ¦
¦   w - Write to a register offset                              ¦
¦   p - <P>ut/write a field                                     ¦
¦   c - <C>hange current channel value                          ¦
¦   n - <N>oise floor histogram                                 ¦
¦   s - <S>leep mode toggle                                     ¦
¦   d - <D>Display PCI writes for resetDevice                   ¦
¦   b - Register dump, values read <B>ack after resetDevice     ¦
¦   0 - Write EEPROM bin to OTP                                 ¦
¦   Q - Write custData to OTP                                   ¦
¦   M - Write MAC to OTP                                        ¦
¦   3 - Write NVRAM to OTP                                      ¦
¦   4 - Write IPSTACK to OTP                                    ¦
¦   K - Write ID to OTP                                         ¦
¦   x - Read an OTP offset                                      ¦
¦   y - Write an OTP offset                                     ¦
¦   l   <l>og power pdadc table                                 ¦
¦ ESC - exit                                                    ¦
=================================================================

All values entered are treated as Hexadecimal
```

Figure 4-4 Utility Menu

5 Manufacturing With the SP137 Platform

The SP137 platform is organized with the MCU connected to the AR4100(P) and SPI flash as shown in Figure 2-1. The DUT is ready to flash the UTF agent and start running RF test cases.

The **flashotp.bin** firmware consists of:

- Firmware for the WLAN driver
- Firmware for the WLAN test driver

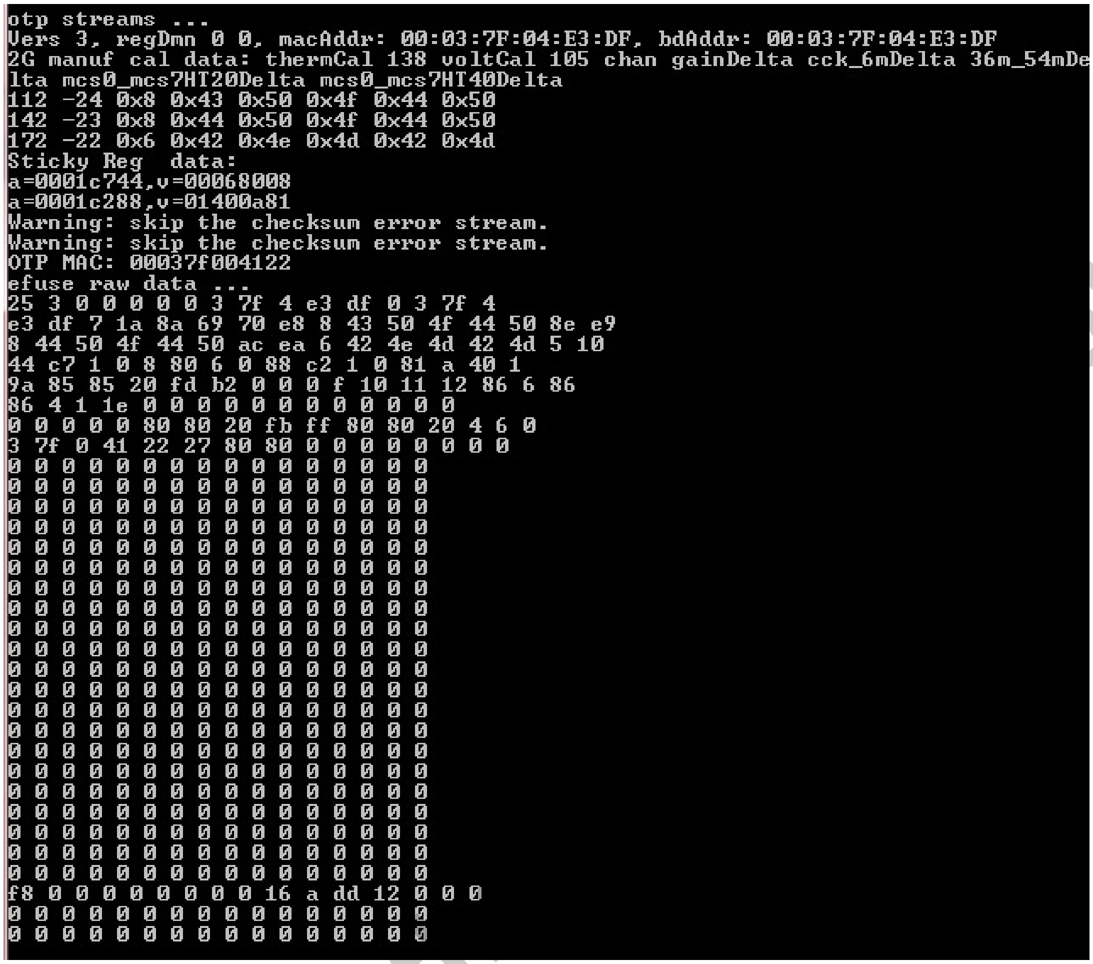
The UTF agent programming setup is similar to programming any binary image to the EFM32. The UTF agent can be programmed from the energyAware commander utility included in the Energy Micro® Simplicity Studio Program. Enter the energyAware commander and connect to the SP137. Another method of programming the internal flash of the EFM microprocessor is through UART (RS232) using the internal default bootloader in the microprocessor. See the *SP137 Development Platform Getting Started Guide* for more information.

Based on the MCU driver, the respective firmware will be loaded in AR4100(P) during bringup. On successful programming, the SP137 is ready with the test firmware binary to interface with ART tool. The UTF Agent bridges the command and response interface between the host PC verification tool (ART) and the test firmware binary UTF.

5.1 Running ART with the SP137

Start the ART command line tool to initialize the DUT and display menu for RF validation. Figure 5-1 shows the output of the command:
C:\Art>**art \remote=sdio \id=6309**

Entering this command initializes test firmware and displays the menu to start the RF test, which can be minimal or extensive based on customer requirements. See Tx/Rx Verification using ART/NART. Customers can select test cases from menu and follow the instructions displayed. Figure 4-2 depicts the continuous Tx test ART command line options. It also has a sub menu with which the output mode, data rate, frequency, and data pattern can be adjusted.



```
otp streams ...
Vers 3, regDmn 0 0, macAddr: 00:03:7F:04:E3:DF, bdAddr: 00:03:7F:04:E3:DF
2G manuf cal data: thermCal 138 voltCal 105 chan gainDelta cck_6mDelta 36m_54mDe
lta mcs0_mcs7HT20Delta mcs0_mcs7HT40Delta
112 -24 0x8 0x43 0x50 0x4f 0x44 0x50
142 -23 0x8 0x44 0x50 0x4f 0x44 0x50
172 -22 0x6 0x42 0x4e 0x4d 0x42 0x4d
Sticky Reg  data:
a=0001c744,v=00068008
a=0001c288,v=01400a81
Warning: skip the checksum error stream.
Warning: skip the checksum error stream.
OTP MAC: 00037f004122
efuse raw data ...
25 3 0 0 0 0 0 3 7f 4 e3 df 0 3 7f 4
e3 df 7 1a 8a 69 70 e8 8 43 50 4f 44 50 8e e9
8 44 50 4f 44 50 ac ea 6 42 4e 4d 42 4d 5 10
44 c7 1 0 8 80 6 0 88 c2 1 0 81 a 40 1
9a 85 85 20 fd b2 0 0 0 f 10 11 12 86 6 86
86 4 1 1e 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 80 80 20 fb ff 80 80 20 4 6 0
3 7f 0 41 22 27 80 80 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
f8 0 0 0 0 0 0 0 0 16 a dd 12 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
```

Figure 5-1 OTP Contents of the AR4100(P)

Display the OTP contents by selecting the options **(p)** and **(T)** from the main menu displayed in Figure 4-1; it will show something similar to Figure 5-1.

5.2 Running IQFact on LitePoint with NART

Qualcomm Atheros provides the ART command line RF test solution for customers. For extensive RF tests, Qualcomm Atheros also supports the LitePoint Multicom Tester to run Tx calibration and verification for Wi-Fi devices with NART. The automated IQFact application is customized for the AR4100(P) solution and enables customers to extensively test the AR4100(P).

The IQFact test requires a similar UTF agent in the MCU as ART; NART is predominately used by applications that control Tx parameters using network sockets, such as the IQFact application, where the RF test commands are communicated to NART from IQFact application. Test results are given back through the IQ2010 and host PC communication mechanism. After completing the tests, results are displayed on the application. The test run is fully automated. This section discusses this process using the IQ2010X and shows a sample of a running Tx verification with Litepoint.

5.2.1 LitePoint IQ2010X Multicom Tester

Figure 5-2 IQ2010X

The LitePoint IQ201X connectivity test system is the first product to specifically address the testing needs of multi-radio products and devices (multicom), which include some combination of multiple wireless functionalities. The IQ201X test system provides a variety of RF ports allowing flexible connection to accommodate a number of device configurations.

LitePoint IQFact$^+$ is software running on Host PC provides easy ways to build and execute test flows dynamically. It also allows changing the test parameters such as test frequencies, pass/fail limits.

The IQWizard application helps to create a Wi-Fi test flow and run a test. More details on using IQFact+ and its software application can be found in the document ***LitePoint IQFact$^+$ User Guide***, which is shipped with LitePoint or downloadable from http://www2.LitePoint.com.

NOTE: The LitePoint IQFact$^+$ application for the AR4100(P) is delivered as separate package for manufacturing. The LitePoint IQFact$^+$ is customized for the AR4100(P)to comply with the NART test application.

5.2.2 Tx verification with LitePoint

This template is used by the LitePoint to run RF test suites. IQWizard is the IQ2010X tool that helps organize the test cases.

Table 5-1 shows the sample template for RF tests.

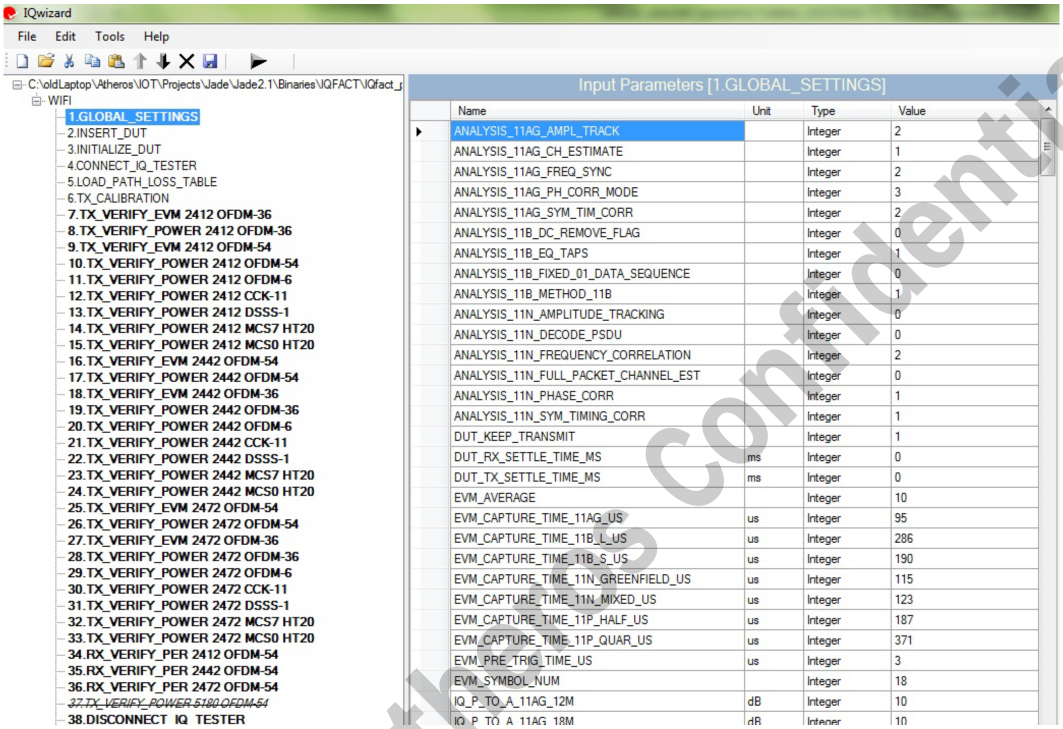
Table 5-1 Major Test Flow Steps the IQWizard Uses in any Wi-Fi Test

Step	IQFACT+ Functions	Comments
1	<Initialize DUT>	Calling this function initializes the DUT
2	<CONNECT IQ_TESTER>	Calling this function establishes a connection to the test system; the test PC connected to IQ2010X equipment through USB connectors establishes an application level connection
3	<LOAD_PATH_LOSS_TABLE>	Path loss is determined based on the frequency set for the test item
4	<TX_VERIFY_EVM>	Allows users to capture the EVM of the captured signal
5	<REMOVE_DUT>	Prepares the test system for removing the DUT
6	<DISCONNECT_IQ_TESTER>	Disconnects the test system from the software

LitePoint uses the Qualcomm Atheros NART solution for all Tx/Rx calibration and verification. Before starting with Litepoint, make sure that the NART/IQFACT package is installed in the host PC and ensure that LitePoint is connected to the host PC. See Figure 2-1.

5.3 IQWizard

Start the IQWizard from the package. Load the IQWizard with the test suite provided by Qualcomm Atheros or create a customized set of the test suites for the DUT. Figure 5-3 shows the IQWizard with the default test suite loaded and ready for testing. Customers can select the required modulation and rates to run Tx verification to reduce test time, or customize the test suite to align with requirements by selecting test cases to run or skip from the list in the left panel.



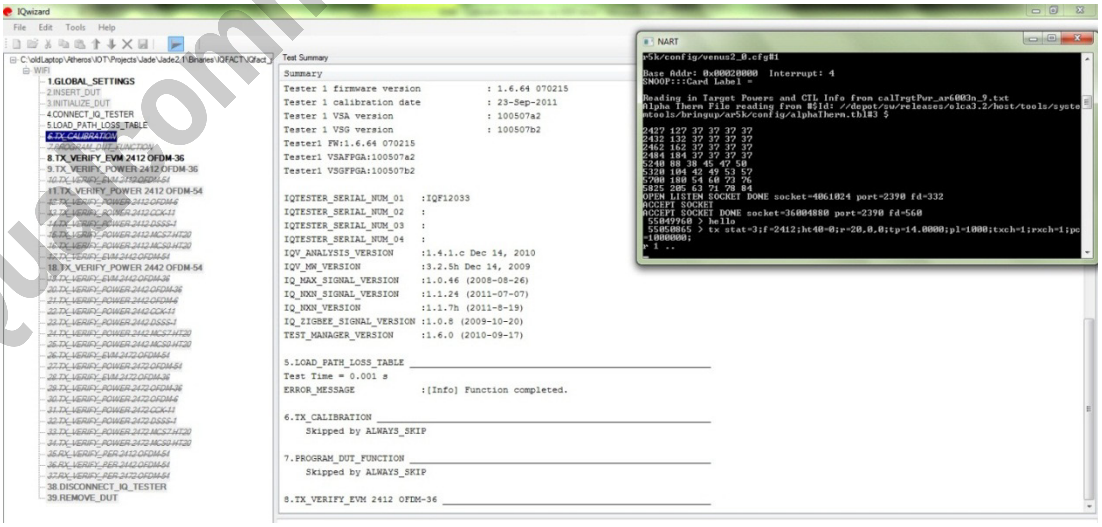
C:\oldLaptop\Atheros\IOT\Projects\Jade\Jade2.1\Binaries\IQFACT\IQfact_
- WIFI
  - 1.GLOBAL_SETTINGS
  - 2.INSERT_DUT
  - 3.INITIALIZE_DUT
  - 4.CONNECT_IQ_TESTER
  - 5.LOAD_PATH_LOSS_TABLE
  - 6.TX_CALIBRATION
  - 7.TX_VERIFY_EVM 2412 OFDM-36
  - 8.TX_VERIFY_POWER 2412 OFDM-36
  - 9.TX_VERIFY_EVM 2412 OFDM-54
  - 10.TX_VERIFY_POWER 2412 OFDM-54
  - 11.TX_VERIFY_POWER 2412 OFDM-6
  - 12.TX_VERIFY_POWER 2412 CCK-11
  - 13.TX_VERIFY_POWER 2412 DSSS-1
  - 14.TX_VERIFY_POWER 2412 MCS7 HT20
  - 15.TX_VERIFY_POWER 2412 MCS0 HT20
  - 16.TX_VERIFY_EVM 2442 OFDM-54
  - 17.TX_VERIFY_POWER 2442 OFDM-54
  - 18.TX_VERIFY_EVM 2442 OFDM-36
  - 19.TX_VERIFY_POWER 2442 OFDM-36
  - 20.TX_VERIFY_POWER 2442 OFDM-6
  - 21.TX_VERIFY_POWER 2442 CCK-11
  - 22.TX_VERIFY_POWER 2442 DSSS-1
  - 23.TX_VERIFY_POWER 2442 MCS7 HT20
  - 24.TX_VERIFY_POWER 2442 MCS0 HT20
  - 25.TX_VERIFY_EVM 2472 OFDM-54
  - 26.TX_VERIFY_POWER 2472 OFDM-54
  - 27.TX_VERIFY_EVM 2472 OFDM-36
  - 28.TX_VERIFY_POWER 2472 OFDM-36
  - 29.TX_VERIFY_POWER 2472 OFDM-6
  - 30.TX_VERIFY_POWER 2472 CCK-11
  - 31.TX_VERIFY_POWER 2472 DSSS-1
  - 32.TX_VERIFY_POWER 2472 MCS7 HT20
  - 33.TX_VERIFY_POWER 2472 MCS0 HT20
  - 34.RX_VERIFY_PER 2412 OFDM-54
  - 35.RX_VERIFY_PER 2442 OFDM-54
  - 36.RX_VERIFY_PER 2472 OFDM-54
  - ~~37.TX_VERIFY_POWER 5180 OFDM-54~~
  - 38.DISCONNECT_IQ_TESTER

Input Parameters [1.GLOBAL_SETTINGS]

Name	Unit	Type	Value
ANALYSIS_11AG_AMPL_TRACK		Integer	2
ANALYSIS_11AG_CH_ESTIMATE		Integer	1
ANALYSIS_11AG_FREQ_SYNC		Integer	2
ANALYSIS_11AG_PH_CORR_MODE		Integer	3
ANALYSIS_11AG_SYM_TIM_CORR		Integer	2
ANALYSIS_11B_DC_REMOVE_FLAG		Integer	0
ANALYSIS_11B_EQ_TAPS		Integer	1
ANALYSIS_11B_FIXED_01_DATA_SEQUENCE		Integer	0
ANALYSIS_11B_METHOD_11B		Integer	1
ANALYSIS_11N_AMPLITUDE_TRACKING		Integer	0
ANALYSIS_11N_DECODE_PSDU		Integer	0
ANALYSIS_11N_FREQUENCY_CORRELATION		Integer	2
ANALYSIS_11N_FULL_PACKET_CHANNEL_EST		Integer	0
ANALYSIS_11N_PHASE_CORR		Integer	1
ANALYSIS_11N_SYM_TIMING_CORR		Integer	1
DUT_KEEP_TRANSMIT		Integer	1
DUT_RX_SETTLE_TIME_MS	ms	Integer	0
DUT_TX_SETTLE_TIME_MS	ms	Integer	0
EVM_AVERAGE		Integer	10
EVM_CAPTURE_TIME_11AG_US	us	Integer	95
EVM_CAPTURE_TIME_11B_L_US	us	Integer	286
EVM_CAPTURE_TIME_11B_S_US	us	Integer	190
EVM_CAPTURE_TIME_11N_GREENFIELD_US	us	Integer	115
EVM_CAPTURE_TIME_11N_MIXED_US	us	Integer	123
EVM_CAPTURE_TIME_11P_HALF_US	us	Integer	187
EVM_CAPTURE_TIME_11P_QUAR_US	us	Integer	371
EVM_PRE_TRIG_TIME_US	us	Integer	3
EVM_SYMBOL_NUM		Integer	18
IQ_P_TO_A_11AG_12M	dB	Integer	10
IQ_P_TO_A_11AG_18M	dB	Integer	10

Figure 5-3 IQWizard Loaded with the Qualcomm Atheros Test Suite

Figure 5-4 displays the start of AR4100(P) verification test for selected (2412 and 2442) and the selected modulation (OFDM at 36 and 54 M rates).



C:\oldLaptop\Atheros\IOT\Projects\Jade\Jade2.1\Binaries\IQFACT\IQfact_
- WIFI
  - 1.GLOBAL_SETTINGS
  - 2.INSERT_DUT
  - 3.INITIALIZE_DUT
  - 4.CONNECT_IQ_TESTER
  - 5.LOAD_PATH_LOSS_TABLE
  - ~~6.TX_CALIBRATION~~
  - ~~7.PROGRAM_DUT_FUNCTION~~
  - 8.TX_VERIFY_EVM 2412 OFDM-36
  - 9.TX_VERIFY_POWER 2412 OFDM-36
  - ~~10.TX_VERIFY_EVM 2412 OFDM-54~~
  - 11.TX_VERIFY_POWER 2412 OFDM-54
  - ~~12.TX_VERIFY_POWER 2412 OFDM-6~~
  - ~~13.TX_VERIFY_POWER 2412 CCK-11~~
  - ~~14.TX_VERIFY_POWER 2412 DSSS-1~~
  - ~~15.TX_VERIFY_POWER 2412 MCS7 HT20~~
  - ~~16.TX_VERIFY_POWER 2412 MCS0 HT20~~
  - ~~17.TX_VERIFY_EVM 2442 OFDM-54~~
  - 18.TX_VERIFY_POWER 2442 OFDM-54
  - ~~19.TX_VERIFY_EVM 2442 OFDM-36~~
  - ~~20.TX_VERIFY_POWER 2442 OFDM-36~~
  - ~~21.TX_VERIFY_POWER 2442 OFDM-6~~
  - ~~22.TX_VERIFY_POWER 2442 CCK-11~~
  - ~~23.TX_VERIFY_POWER 2442 DSSS-1~~
  - ~~24.TX_VERIFY_POWER 2442 MCS7 HT20~~
  - ~~25.TX_VERIFY_POWER 2442 MCS0 HT20~~
  - ~~26.TX_VERIFY_EVM 2472 OFDM-54~~
  - ~~27.TX_VERIFY_POWER 2472 OFDM-54~~
  - ~~28.TX_VERIFY_EVM 2472 OFDM-36~~
  - ~~29.TX_VERIFY_POWER 2472 OFDM-36~~
  - ~~30.TX_VERIFY_POWER 2472 OFDM-6~~
  - ~~31.TX_VERIFY_POWER 2472 CCK-11~~
  - ~~32.TX_VERIFY_POWER 2472 DSSS-1~~
  - ~~33.TX_VERIFY_POWER 2472 MCS7 HT20~~
  - ~~34.TX_VERIFY_POWER 2472 MCS0 HT20~~
  - ~~35.RX_VERIFY_PER 2412 OFDM-54~~
  - ~~36.RX_VERIFY_PER 2442 OFDM-54~~
  - ~~37.RX_VERIFY_PER 2472 OFDM-54~~
  - 38.DISCONNECT_IQ_TESTER
  - 39.REMOVE_DUT

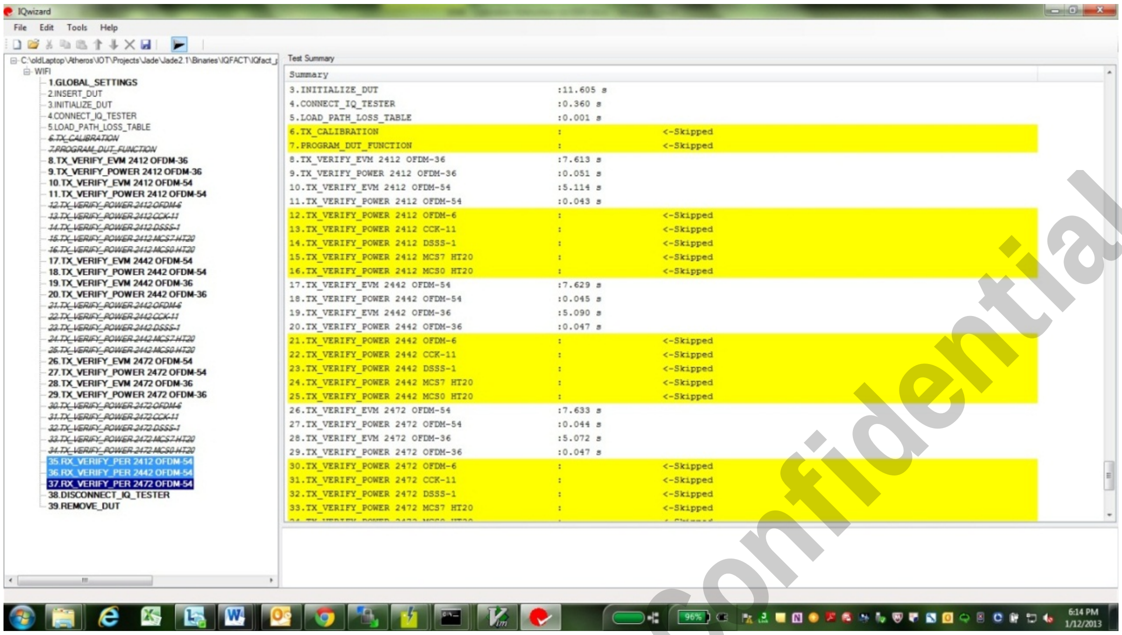
Test Summary

```
Summary
Tester 1 firmware version          : 1.6.64 070215
Tester 1 calibration date          : 23-Sep-2011
Tester 1 VSA version               : 100507a2
Tester 1 VSG version               : 100507b2
Tester1 FW:1.6.64 070215
Tester1 VSAFPGA:100507a2
Tester1 VSGFPGA:100507b2

IQTESTER_SERIAL_NUM_01   :IQF12033
IQTESTER_SERIAL_NUM_02   :
IQTESTER_SERIAL_NUM_03   :
IQTESTER_SERIAL_NUM_04   :
IQV_ANALYSIS_VERSION     :1.4.1.c Dec 14, 2010
IQV_MW_VERSION           :3.2.5h Dec 14, 2009
IQ_MAX_SIGNAL_VERSION    :1.0.46 (2008-08-26)
IQ_NXN_SIGNAL_VERSION    :1.1.24 (2011-07-07)
IQ_NXN_VERSION           :1.1.7h (2011-8-19)
IQ_ZIGBEE_SIGNAL_VERSION :1.0.8 (2009-10-20)
TEST_MANAGER_VERSION     :1.6.0 (2010-09-17)

5.LOAD_PATH_LOSS_TABLE ________________________________________________
Test Time = 0.001 s
ERROR_MESSAGE            :[Info] Function completed.

6.TX_CALIBRATION ______________________________________________________
    Skipped by ALWAYS_SKIP

7.PROGRAM_DUT_FUNCTION ________________________________________________
    Skipped by ALWAYS_SKIP

8.TX_VERIFY_EVM 2412 OFDM-36 __________________________________________
```

```
r5k/config/venus2_0.cfg#1

Base Addr: 0x00020000  Interrupt: 4
SNOOP:::Card Label =

Reading in Target Powers and CTL Info from calTrgtPwr_ar6003n_9.txt
Alpha Therm File reading from #$Id: //depot/sw/releases/olca3.2/host/tools/syste
mtools/bringup/ar5k/config/alphaTherm.tbl#3 $

2427 127 37 37 37 37
2432 132 37 37 37 37
2462 162 37 37 37 37
2484 184 37 37 37 37
5240 88 38 45 47 50
5320 104 42 49 53 57
5700 180 54 60 73 76
5825 205 63 71 78 84
OPEN LISTEN SOCKET DONE socket=4061024 port=2390 fd=332
ACCEPT SOCKET
ACCEPT SOCKET DONE socket=36004880 port=2390 fd=560
 55049960 > hello
 55050865 > tx stat=3;f=2412;ht40=0;r=20,0,0;tp=14.0000;pl=1000;txch=1;rxch=1;pc
=1000000;
r 1 ..
```

Figure 5-4 IQWizard Running NART for the AR4100(P)

Figure 5-5 shows the test conducted on Tx and Rx verification for the OFDM module on various channels at different rates. The right panel displays test results and the time taken to run the tests.



C:\oldLaptop\Atheros\IOT\Projects\Jade\Jade2.1\Binaries\IQFACT\IQfact_
WIFI
1.GLOBAL_SETTINGS
2.INSERT_DUT
3.INITIALIZE_DUT
4.CONNECT_IQ_TESTER
5.LOAD_PATH_LOSS_TABLE
~~6.TX_CALIBRATION~~
~~7.PROGRAM_DUT_FUNCTION~~
8.TX_VERIFY_EVM 2412 OFDM-36
9.TX_VERIFY_POWER 2412 OFDM-36
10.TX_VERIFY_EVM 2412 OFDM-54
11.TX_VERIFY_POWER 2412 OFDM-54
~~12.TX_VERIFY_POWER 2412 OFDM-6~~
~~13.TX_VERIFY_POWER 2412 CCK-11~~
~~14.TX_VERIFY_POWER 2412 DSSS-1~~
~~15.TX_VERIFY_POWER 2412 MCS7 HT20~~
~~16.TX_VERIFY_POWER 2412 MCS0 HT20~~
17.TX_VERIFY_EVM 2442 OFDM-54
18.TX_VERIFY_POWER 2442 OFDM-54
19.TX_VERIFY_EVM 2442 OFDM-36
20.TX_VERIFY_POWER 2442 OFDM-36
~~21.TX_VERIFY_POWER 2442 OFDM-6~~
~~22.TX_VERIFY_POWER 2442 CCK-11~~
~~23.TX_VERIFY_POWER 2442 DSSS-1~~
~~24.TX_VERIFY_POWER 2442 MCS7 HT20~~
~~25.TX_VERIFY_POWER 2442 MCS0 HT20~~
26.TX_VERIFY_EVM 2472 OFDM-54
27.TX_VERIFY_POWER 2472 OFDM-54
28.TX_VERIFY_EVM 2472 OFDM-36
29.TX_VERIFY_POWER 2472 OFDM-36
~~30.TX_VERIFY_POWER 2472 OFDM-6~~
~~31.TX_VERIFY_POWER 2472 CCK-11~~
~~32.TX_VERIFY_POWER 2472 DSSS-1~~
~~33.TX_VERIFY_POWER 2472 MCS7 HT20~~
~~34.TX_VERIFY_POWER 2472 MCS0 HT20~~
35.RX_VERIFY_PER 2412 OFDM-54
36.RX_VERIFY_PER 2442 OFDM-54
37.RX_VERIFY_PER 2472 OFDM-54
38.DISCONNECT_IQ_TESTER
39.REMOVE_DUT

```
Summary
3.INITIALIZE_DUT                               :11.605 s
4.CONNECT_IQ_TESTER                            :0.360 s
5.LOAD_PATH_LOSS_TABLE                         :0.001 s
6.TX_CALIBRATION                               :                  <-Skipped
7.PROGRAM_DUT_FUNCTION                         :                  <-Skipped
8.TX_VERIFY_EVM 2412 OFDM-36                   :7.613 s
9.TX_VERIFY_POWER 2412 OFDM-36                 :0.051 s
10.TX_VERIFY_EVM 2412 OFDM-54                  :5.114 s
11.TX_VERIFY_POWER 2412 OFDM-54                :0.043 s
12.TX_VERIFY_POWER 2412 OFDM-6                 :                  <-Skipped
13.TX_VERIFY_POWER 2412 CCK-11                 :                  <-Skipped
14.TX_VERIFY_POWER 2412 DSSS-1                 :                  <-Skipped
15.TX_VERIFY_POWER 2412 MCS7 HT20              :                  <-Skipped
16.TX_VERIFY_POWER 2412 MCS0 HT20              :                  <-Skipped
17.TX_VERIFY_EVM 2442 OFDM-54                  :7.629 s
18.TX_VERIFY_POWER 2442 OFDM-54                :0.045 s
19.TX_VERIFY_EVM 2442 OFDM-36                  :5.090 s
20.TX_VERIFY_POWER 2442 OFDM-36                :0.047 s
21.TX_VERIFY_POWER 2442 OFDM-6                 :                  <-Skipped
22.TX_VERIFY_POWER 2442 CCK-11                 :                  <-Skipped
23.TX_VERIFY_POWER 2442 DSSS-1                 :                  <-Skipped
24.TX_VERIFY_POWER 2442 MCS7 HT20              :                  <-Skipped
25.TX_VERIFY_POWER 2442 MCS0 HT20              :                  <-Skipped
26.TX_VERIFY_EVM 2472 OFDM-54                  :7.633 s
27.TX_VERIFY_POWER 2472 OFDM-54                :0.044 s
28.TX_VERIFY_EVM 2472 OFDM-36                  :5.072 s
29.TX_VERIFY_POWER 2472 OFDM-36                :0.047 s
30.TX_VERIFY_POWER 2472 OFDM-6                 :                  <-Skipped
31.TX_VERIFY_POWER 2472 CCK-11                 :                  <-Skipped
32.TX_VERIFY_POWER 2472 DSSS-1                 :                  <-Skipped
33.TX_VERIFY_POWER 2472 MCS7 HT20              :                  <-Skipped
```

Figure 5-5 IQWizard with Additional Test Cases and Results

At the end of the verification, IQFact displays the results in the right panel with FAIL or PASS displayed for each test case. The failure test cases are marked in red for identification. After the pass results, user can remotely disconnect the DUT and connect the next DUT for validation.

5.4 Programming the end application (throughput demo)

At the end of the RF test, the AR4100(P) is programmed with an end application for functionality test cases. Qualcomm Atheros provides the default application driver called **throughput_demo**; this end application driver user can connect to an access point and run ping tests as required. See the *Internet of Things Wireless System Release (I-WSR) 2.1 Release Notes* for more information on the flash application driver. The common driver has one of key feature for MAC address programming using the command line feature.

6 MAC Address Programming

MAC address programming allows a host MCU to program a new MAC address into OTP memory; once in OTP, it is used on subsequent system bring-ups as the device MAC address. This feature was created so customers could program a permanent MAC address into the device. Because each device should have a unique MAC address, this operation is critical to efficient customer production processes.

The MAC address programming feature is included in the host MCU throughput demo driver; the MAC address can be programmed using WMI interface in the host driver, so that users can simply program the MAC using the **wmiconfig** command:
wmiconfig --macstore 001122334455

Upon operation completion, the throughput demo responds with one of these printed statements:

Statement	Description
Invalid MAC address length	The MAC address does not contain 12 characters
Invalid character	The MAC address contains characters other than 0123456789abcdef
MAC PROGRAM ERROR: Unknown driver error	The driver has failed the IOCTL for some unknown reason; this condition can occur if the driver does not have the necessary code compiled
MAC PROGRAM ERROR: Driver was unable to complete the request	The driver could not send the command to the AR4100(P); this condition can occur when the AR4100(P) firmware does not have the necessary MAC OTP programming image
MAC PROGRAM ERROR: Firmware failed to program the MAC address	The firmware contains the proper image but still failed to program the MAC address as requested; this condition can occur when the requested MAC address equals the current MAC address
MAC PROGRAM ERROR: Device unknown failure	The firmware contains the proper image but failed in the process of programming the MAC address for some unexpected reason
MAC PROGRAM SUCCESS	The firmware has successfully programmed the new MAC address into OTP

NOTE: Because OTP memory is a relatively small resource and each byte can only be written one time, if this memory becomes filled no further writes to OTP can be made.

NOTE: This mechanism is meant when only the MAC must be programmed and no per-board MAC or OTP tweaking is required.

7 flashotp.bin Golden Board Data

This host application allows customers to update a fine-tuned board by replacing board data with the default (golden) board data file firmware binary supplied included in the AR4100(P) software package **flashotp.bin** file.

This tool allows customers to update their RF parameters. See the *Replacing AR4100(P) Firmware Board Data with Customer Board Data Application Note* for more information.

8 FCC Certification Test Suites

FCC certification validates these DUT wireless capabilities:

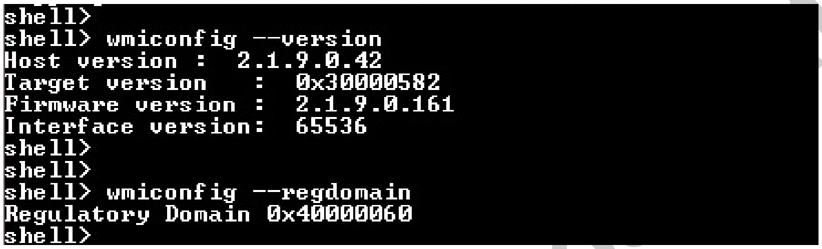
- Facility to change the power levels.
- Facility to change the data rates
- Facility to select modes (802.11b, 802.11g, 802.11n, HT20, HT40, etc.)
- Facility to set the radio to Rx-only mode.

These facilities are provided in the ART command line tool; using the command line option **(c)** for continuous Tx mode enables all of these options except for Rx mode. To select Rx-only mode, choose option **(r)**. See section 4.2.

9 Regulatory Test

Regulatory testing involves verifying Tx power for the regulatory domain set in the WLAN driver. The regulatory test ensures that the DUT follows the correct regulatory Tx power setting and uses the setup shown in Figure 2-1. The DUT is loaded in world wide mode by default; after flashing the end application as described in section 5.4, the DUT is ready to verify regulatory.

Load the DUT with the default driver and type the wmi command to find the current regulatory test:
wmiconfig –regdomain



```
shell>
shell> wmiconfig --version
Host version :  2.1.9.0.42
Target version    :  0x30000582
Firmware version :  2.1.9.0.161
Interface version:  65536
shell>
shell>
shell> wmiconfig --regdomain
Regulatory Domain 0x40000060
shell>
```

Figure 9-1 Regulatory Command

Figure 9-1 displays the default regulatory (World Mode) immediately after loading the driver in the DUT. See the *Regulatory Configuration of the AR4100 Application Note* for supported regulatory modes and their usage.

To calculate the regulatory power levels, software provides a special **wmi** command to transmit raw packets at required modulation and rate (see the *AR4100 Throughput Tool User's Guide* for raw mode command line options). From the regulatory setup, start the VSA and then run the RAW mode command to transmit raw packets (this example transmits raw packets in channel 5 of the 6-Mbps data rate and a size of 1400 bytes for 14 transmits):
wmiconfig – raw 4 14 1400 5 2

Measure the Tx power on the respective channel in the world wide regulatory chart. Output power in world wide mode will be in channel 5 with a maximum data rate of 14 dBm.

9.1 Special binary on regulatory

Qualcomm Atheros provides the **flashotp.bin** for other regulatory settings. Similar tests can be conducted for special binary (FCC3_FCCA).

The command **wmiconfig –regdomain** gives the regulatory domain set in the firmware. For example, the FCC regdomain is 0x3A (in hex). Users can verify their regulatory using this **wmiconfig** command.

See the *Regulatory Configuration of the AR4100 Application Note.*

9.2 UDP Tx option

The DUT can be connected to an AP in parallel with Litepoint. Using the throughput demo, customers can connect it to an AP with required regulatory setting then do a UDP data transfer using LitePoint to measure Tx power on that particular regulatory domain. Refer to the command **benchtx** and its usage in the *AR4100 Throughput Tool User's Guide* for information on transmitting UDP Packets.