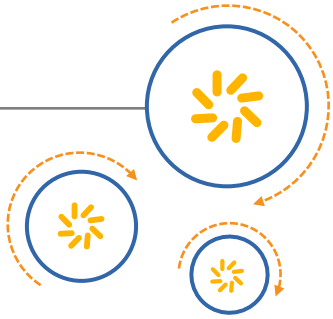
Qualcomm Atheros, Inc.



QCA9563.ILQ.1.1 CS

Release Notes

80-Y9137-2 Rev. A

January 9, 2015

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

Revision history

Revision	Date	Description
A	January 2015	Initial release

Contents

1 Introduction 5

1.1 Purpose 5
1.2 Identification 5
1.3 Related documentation 5

2 Features 6

3 Supported Hardware 8

3.1 AP152: QCA9563 reference design 8
3.2 AP152 configuration 8
3.3 CUS223: QCA9880, 5 GHz 802.11ac-based reference design 8

4 Performance Results and Known Issues 9

4.1 AP152-H100 (11ng 3x3) + CUS223 (11ac 3x3) CST test coverage 9
4.2 Build 9
4.3 Performance Results 10
4.3.1 KPI 10
4.3.2 RVR 11
4.4 Errata 12

5 Download and Build Instructions 13

5.1 Set up the build environment 14
5.2 Download the software 15
5.3 Build the software 15
5.4 Flashing instructions 15
5.4.1 Setup the flashing environment 15
5.4.2 Execute flashing commands for NOR: 8M platforms 16
5.4.3 Execute flashing commands for NOR: 16M platforms 16
A.1 Overview 17
A.2 Supported functionalities 17
A.3 New test_flow operation 17
A.3.1 MAC addresses 17
A.3.2 Initialization scripts 18
A.3.3 Shift operation 18
A.4 Known limitations 18
A.5 Commands to run NART on AP152 with QSDK 18

Figures

Figure 4-1 WDS RvR – 11ACVHT80 - Channel 149 (5 GHz) 11
Figure 4-2 WDS RvR – 11NGHT40 - Channel 6 (2 GHz) 11

Tables

Table 2-1 Features unit-tested for QCA9563 2.4 GHz radios 6
Table 2-2 Features unit-tested for QCA9880 radio in 5 GHz 7
Table 3-1 AP152: QCA9563-based reference design 8
Table 3-2 Basic AP1 configuration details 8
Table 3-3 CUS223: QCA9880/9890/9882/9892, 5 GHz 802.11ac-based reference design 8
Table 4-1 AP152-H100 (11ng 3x3) + CUS223 (11ac 3x3) CST test coverage 9
Table 4-2 KPI Table 10
Table 5-1 Packages used in one or more profiles 13

1 Introduction

1.1 Purpose

This document provides the details related to the QCA9563.ILQ.1.1 CS release.

Despite being downloaded from the Qualcomm® ChipCode site, the Qualcomm Atheros Support site, or embedded on Equipment received from Qualcomm Atheros, Inc. ("QCA") the Qualcomm Atheros QCA9563.ILQ.1.1 CS software release ("SW Package") shall be considered Deliverables and is subject to the terms and conditions of the Qualcomm Atheros, Inc. Limited Use Agreement ("Agreement"). The applicable Use Period, as that term is defined in the Agreement, for the SW Package starts on the Effective Date of your Agreement or the date you received the SW Package, whichever is later, and expires on January 8, 2016 (unless a different Use Period for the SW Package is specified in the Agreement, in which case the Use Period in the Agreement shall prevail). By receiving and/or using the SW Package, you acknowledge and agree that your use of the SW Package is subject to the terms and conditions of the signed Agreement. If you do not agree to the terms of the Agreement, have not signed such Agreement, or have not received the written approval from QCA set forth below, you shall immediately delete the SW Package from all storage media and destroy any and all copies made.

If you wish to utilize the SW Package commercially, you must obtain written approval from QCA, which may be obtained by contacting QCA by email at QCA.Contracts@qualcomm.com or via facsimile to (408) 516-5825, that the SW Package is approved for use under the terms of an applicable definitive agreement in effect between you and QCA or you must enter into a new definitive license agreement relating to the SW Package.

1.2 Identification

This release is:	**QCA9563.ILQ.1.1 CS**
The Linux Foundation hosted open source label that corresponds to this release (caf_TAG) is:	**caf_AU_LINUX_QSDK_RELEASE_BANANA_SW_TARGET_ALL.2.7.035.xml**

1.3 Related documentation

In addition to this release notes, refer to these documents for more information on using this release.

Document Number	Title
80-Y7572-1	AP152-030 Hardware Reference Guide
80-Y1926-2	CUS223-523 Hardware Reference Guide
80-Y7207-1	AP 10.2 Programmer's Guide – Wireless LAN
80-Y0861-1	ART2 Reference Design for AR93xx/94xx/95xx & QCA98xx Reference Guide
80-Y0852-1	QCA98xx ART2 Non-Volatile Memory Structure Reference Guide
80-Y7186-3	Switch Software Development Kit Diagnostic Shell User Guide

2 Features

This release is made up of the following components supporting the Qualcomm XSPAN™ QCA9563 SoC with MIPS 74Kc Processor and Qualcomm VIVE™ QCA9880:

- The QSDK package that provides basic open source Linux support, cross-compilers, boot loaders, file system images, and script files.
- Qualcomm Atheros Wi-Fi software driver; see Table 2-1 and Table 2-2.
- QCA9880 10.2.3 firmware source code (only for specific customers).
- WAPI source package for software requiring a third party license (only for specific customers).
- WLAN driver is Wi-Fi 802.11ac certification ready.

This release is tested with these platforms going through CST:

- AP152 (QCA9563 3x3) + CUS223 (QCA9880 3x3)

Table 2-1 Features unit-tested for QCA9563 2.4 GHz radios

Category	Description
Basic HT (802.11n) support	▪ 2.4 GHz ▪ AP mode (beaconing, association, disassociate, deauth) ▪ AP legacy power save ▪ 20/40 MHz channel width, 3 spatial streams, MCS 0–23 (basic and supported rate set) ▪ A-MPDU ▪ A-MSDU RX ▪ Block ACK mechanism ▪ SGI ▪ Tx/Rx STBC, LDPC
Encryption/ Decryption	▪ WEP ▪ TKIP ▪ AES
Security modes	▪ Open ▪ Shared key, WPA/WPA2, WPA-PSK/WPA2-PSK ▪ Extended security (EAP methods)
Modes of operation	▪ AP mode ▪ Extender AP mode (3-address) ▪ WDS (Root AP and Repeater AP) (4-address) ▪ Repeater independent mode ▪ Enhanced independent repeater mode ▪ QWRAP mode (Supported with low Uplink throughput due to software decryption)
Retail features	▪ mBSSID (16 VAPs) ▪ Aggregation limits, WMM, WDS, repeater, Rx defrag, ACS, multicast enhancements, WPS/WPS2.0, extender AP, ANI, DCS, ACS, QWRAP
Miscellaneous features	▪ ANI ▪ RIFS
CoreBSP	▪ QCA956x SoC BSP patches for 3.3.8 Linux kernel ▪ SoC GMAC driver with Ethernet switch driver ▪ SPI controller driver for NOR flash
Tools	▪ Basic CFG commands, pktlog, basic athstats, regdump

Table 2-2 Features unit-tested for QCA9880 radio in 5 GHz

Category	Description
Basic 802.11a support	▪ 5 GHz ▪ AP mode (beaconing, association, disassociate, deauth) ▪ AP legacy power save ▪ STA modes including scanning ▪ OFDM rates (6−54 Mbps) (basic and supported rate set) ▪ RTS/CTS ▪ DFS
Basic HT (802.11n) support	▪ 5 GHz ▪ AP mode (beaconing, association, disassociate, deauth) ▪ AP legacy power save ▪ STA modes including scanning ▪ 20/40 MHz channel width ▪ 3 spatial streams ▪ MCS 0-23 in 11n mode ▪ A-MPDU, A-MSDU RX ▪ Block ACK mechanism ▪ SGI, TX/RX STBC, LDPC, DFS
Basic VHT (802.11ac) support	▪ 5 GHz ▪ AP mode (beaconing, association, disassociate, deauth) ▪ AP legacy power save ▪ STA modes including scanning ▪ 20/40/80 MHz channel width ▪ MCS 0−9 (basic and supported rate set) ▪ 3 spatial streams ▪ VHT A-MPDU delimiter for Rx/Tx for single MPDU ▪ Rx/Tx A-MPDU, A-MSDU RX ▪ CCA on secondary ▪ Short guard interval ▪ LDPC, Tx/Rx STBC ▪ CTS with bandwidth signaling in response to RTS with bandwidth signaling (dynamic bandwidth switching) ▪ UAPSD, DFS
Encryption/Decryption	WEP, TKIP, AES
Security modes	▪ Open ▪ Shared Key, WPA/WPA2, WPA-PSK/WPA2-PSK ▪ Extended security (EAP methods) ▪ WAPI (AP only)
Retail feature set	▪ MBSSID ▪ Aggregation limits ▪ WMM ▪ WDS ▪ Repeater ▪ Rx defrag ▪ ACS, DCS ▪ Multicast enhancements ▪ WPS/WPS2.0 ▪ 128 clients
Basic interop	Interop with various 802.11ac APs put in STA mode and various 802.11n STA cards available in the market today
Tools	Basic CFG commands, PKTLOG, Basic athstats, Regdump, Firmware Debug Log

3 Supported Hardware

3.1 AP152: QCA9563 reference design

Table 3-1 AP152: QCA9563-based reference design

Hardware board	Supported	Comments
AP152-030	Yes	–

3.2 AP152 configuration

Table 3-2 Basic AP1 configuration details

CPU	775 MHz
DDR	650/400 MHz
AHB	258 MHz
DDR	16-bit DDR2/DDR1
WLAN	QCA9563 3x3 in 2.4 GHz
Switch	SGMII, QCA8337N
Boot	Boot from NOR flash
USB	2xUSB 2.0 in host mode
PCIE	PCIE 1.1

3.3 CUS223: QCA9880, 5 GHz 802.11ac-based reference design

Table 3-3 CUS223: QCA9880/9890/9882/9892, 5 GHz 802.11ac-based reference design

Hardware board	Supported	Comments
CUS223-523	Yes	3x3 configuration, high power

4 Performance Results and Known Issues

4.1 AP152-H100 (11ng 3x3) + CUS223 (11ac 3x3) CST test coverage

Table 4-1 AP152-H100 (11ng 3x3) + CUS223 (11ac 3x3) CST test coverage

Name	Coverage (%)	(#)	Fail (%)	(#)	Pass (%)	(#)	Block (%)	(#)	P1	P2	P3
(F) Performance	100%	60	0%	0	100%	60	0%	0	0	0	0
(F) Cert	100%	69	0%	0	100%	69	0%	0	0	0	0
(F) Func	100%	166	0%	0	100%	166	0%	0	0	0	1
(F) IOT	100%	32	0%	0	100%	32	0%	0	0	0	0
QCA9563.ILQ.1.1_CUS223-3x3-QT-SP_Overall	100%	327	0%	0	100%	327	0%	0	0	0	0

4.2 Build

META **NHSS.ILQ.2.7-00029-P-1**

HW **AP152**-H100 (11ng 3x3) (ART2 4.9.849) + **CUS223**- (11ac 3x3) (ART2 4.9.849)

4.3 Performance Results

4.3.1 KPI

NOTE: All throughput peak performance values in Mbps.

Setup Details:

Build Used	**NHSS.ILQ.2.7-00035-P-1**
Root AP HW	AP152 (755H100D1917) (3x3) + CUS223 (N10J3CX9T) (3X3) (ART4.9.849)
Client AP HW	AP152 (7555H100D1941) (3X3) + CUS223 (N10J3D0XY) (3X3) (ART4.9.849)
Infra Client 2 GHz	XB112-045-D5410
Infra Client 5 GHz	Mac Book Pro
Test Setup	Cabled
Tools used	IxChariot
Script	High-Throughput.scr, UDP_Throughput.scr
Channel	2G-6, 5G-149

Table 4-2 KPI Table

Stream	TCP_DownLink		TCP_UpLink		TCP_Bidi		UDP_DownLink		UDP_UpLink		UDP_Bidi	
Modes	Peak throughput	CPU utilization	Peak throughput	CPU utilization	Peak throughput	CPU utilization	Peak throughput	CPU utilization	Peak throughput	CPU utilization	Peak throughput	CPU utilization
WDS-11ACHT20	215	28.00%	214	27.00%	213	26.00%	229	24.00%	226	26.00%	228	28.00%
WDS-11NAHT40PLUS	346	42.00%	347	45.00%	354	46.00%	370	52.00%	371	48.00%	370	49.00%
WDS-11ACVHT80	894	89.00%	891	97.00%	890	85.00%	898	88.00%	891	87.00%	970	86.00%
WDS-11ACHT40	440	48.00%	440	50.00%	446	47.00%	471	53.00%	472	49.00%	474	51.00%
Infra-11NGHT40PLUS	276	72.00%	284	66.00%	301	70.00%	271	74.00%	277	69.00%	273	69.00%
Infra-11NAHT40PLUS	328	37.00%	313	39.00%	330	36.00%	343	39.00%	337	33.00%	339	32.00%
DBDC-11NGHT40PLUS,11ACVHT80	810	100.00%	789	100.00%	840	100.00%	840	94.00%	811	98.00%	885	99.00%
WDS-11NGHT40PLUS	276	69.00%	277	68.00%	302	70.00%	315	85.00%	310	64.00%	312	68.00%
Infra-11ACVHT80	880	88.00%	850	98.00%	871	99.00%	920	93.00%	903	98.00%	970	97.00%

4.3.2 RVR

Setup Details:

Build Used	**NHSS.ILQ.2.7-00035-P-1** (unified NART – 4.9.849)
HW	AP152 (755H100D1917)(3X3)+CUS223(N10J3CX9T)(3X3)

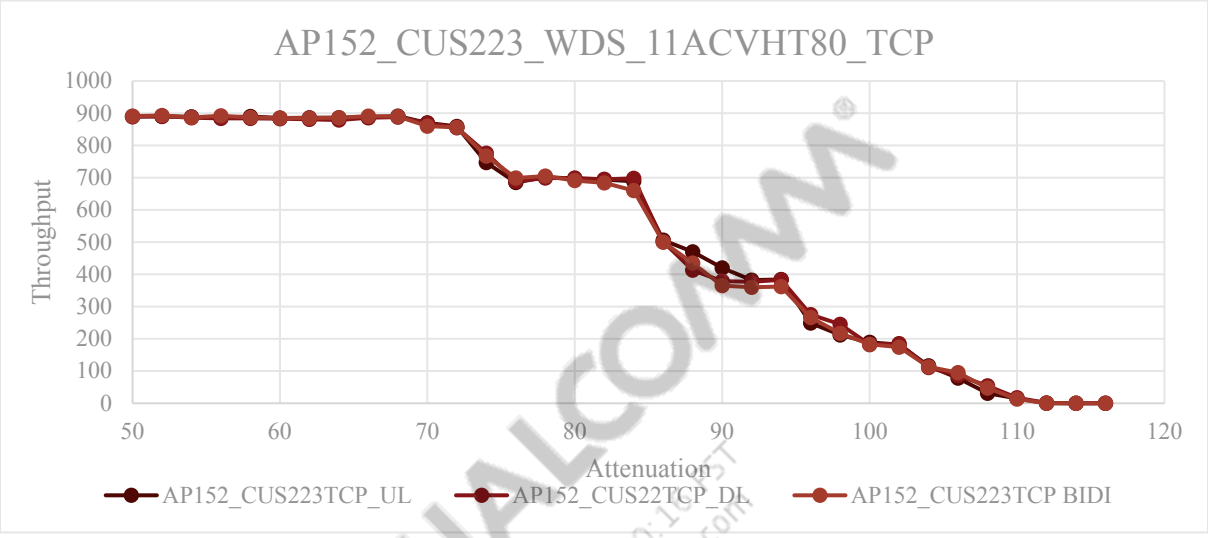


Figure 4-1 WDS RvR – 11ACVHT80 - Channel 149 (5 GHz)

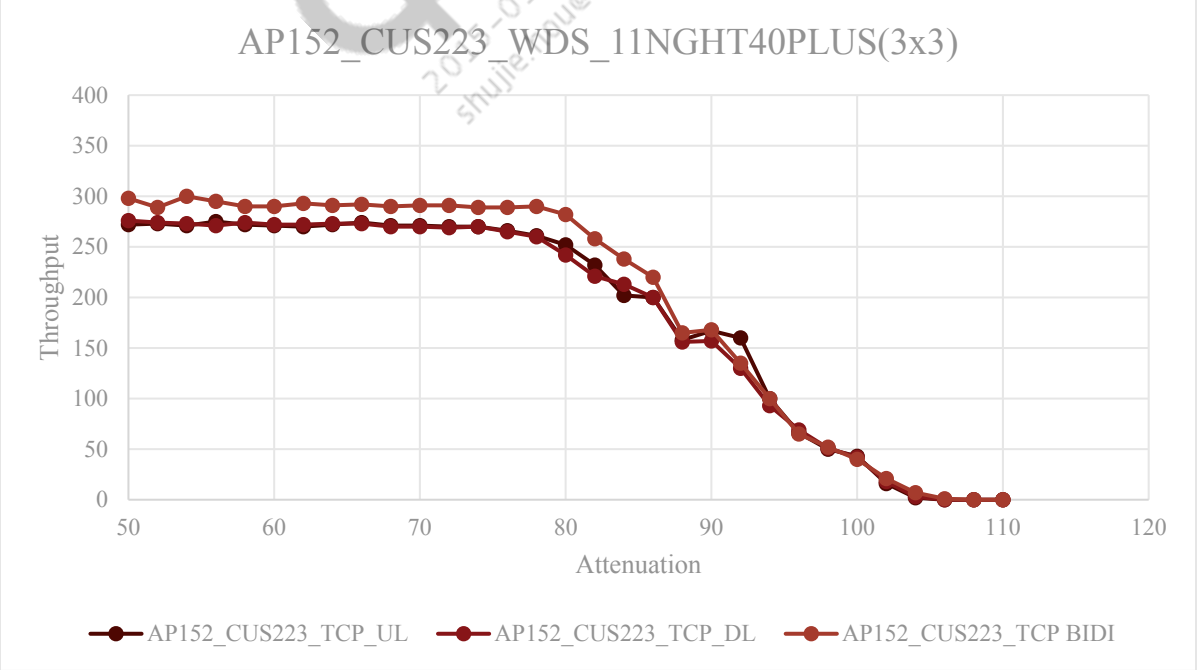


Figure 4-2 WDS RvR – 11NGHT40 - Channel 6 (2 GHz)

4.4 Errata

	CR title
1.	Ping resuming after few tens of seconds when restart command is given through the uci command. Ping is likely to take 30 seconds to resume when restart is given through the uci command. It is recommended to use reload instead of restart to solve this issue. A network restart tears down the entire network stack, whereas reload just reloads the new configuration.

5 Download and Build Instructions

This release consists of the packages listed in Table 5-1.

Table 5-1 Packages used in one or more profiles

LinuxART2CS10.2v4.9.363.tar.bz2
bridge-utils-1.5.tar.gz
busybox-1.19.4.tar.bz2
dnsmasq-2.62.tar.gz
dosfstools-3.0.12.tar.gz
dropbear-2011.54.tar.bz2
e2fsprogs-1.42.4.tar.gz
ethtool-3.4.1.tar.xz
fast-classifier-gc8d70b1.tar.gz
hotplug2-201.tar.gz
iozone3_420.tar
iperf-2.0.5.tar.gz
iputils-s20101006.tar.bz2
libdaemon-0.14.tar.gz
json-c-0.9.tar.gz
libnfnetlink-1.0.0.tar.bz2
readline-5.2.tar.gz
libubox-2013-07-04-11e8afea0f7eb34f8c23a8e589ee659c46f3f8aa.tar.gz
lua-5.1.4.tar.gz
mcproxy-1.1.0.y.tar.bz2
miniupnpd-1.8.20130426.tar.gz
monit-5.4.tar.gz
netifd-2013-05-13-bc4a4bb127622c76085ecec7fd20448aad7bafaf.tar.gz
ntfs-3g_ntfsprogs-2011.4.12.tgz
qca-acfg-10.2.3.31.tar.bz2
qca-hostap-10.2.3.31.tar.bz2
qca-spectral-10.2.3.31.tar.bz2
qca-ssdk-ge424ca4.tar.gz
qca-ssdk-shell-gb6e94dd.tar.gz
qca-wapid-10.2.3.31.tar.bz2
qca-wifi-10.2.3.31.tar.bz2
qca-wrapd-10.2.3.31.tar.bz2

radvd-1.9.1.tar.gz
rp-pppoe-3.10.tar.gz
shortcut-fe-gc8d70b1.tar.gz
sigma-dut-2013-04-13-8ca210b26d3db58adf842a31171613c544d4b84e.tar.gz
sysstat-10.1.7.tar.bz2
tftp-hpa-0.48.tar.gz
u-boot-2012.04.01.tar.bz2
ubus-2013-01-13-bf566871bd6a633e4504c60c6fc55b2a97305a50.tar.gz
uci-2013-01-04.1.tar.gz
uClibc++-0.2.4.tar.bz2
uhttpd-2012-10-30-e57bf6d8bfa465a50eea2c30269acdfe751a46fd.tar.gz
util-linux-2.21.2.tar.xz
wide-dhcpv6-20080615.tar.gz
wireless_tools.29.tar.gz
xtables-addons-1.42.tar.xz
zlib-1.2.7.tar.bz2

5.1 Set up the build environment

This framework has been developed using Ubuntu (from version 10.04 to version 13.04), and Debian. However, the QSDK framework regenerates the critical tools required to compile the firmware, at build-time. In that sense, the framework is independent from the host environment. Though the framework is developed using the above distributions, it is expected to work fine on other distributions such as RedHat, Mint, or Fedora.

The command for Debian/Ubuntu is (needs to be customized for other distributions):

```
$ sudo apt-get install gcc g++ binutils patch bzip2 flex make gettext \
  pkg-config unzip zlib1g-dev libc6-dev subversion libncurses5-dev gawk \
  sharutils curl libxml-parser-perl ocaml-nox
```

As the framework automatically downloads the required open source components, make sure that an internet connection is active on the build host while creating the build.

Install repo and git

Repo is used to download and update packages from multiple repositories by reading an XML manifest; the **repo** script must be available in the path. Newer versions of Ubuntu create the folder **~/bin**; add that folder to the PATH variable and place repo there; some older versions do not. Qualcomm Atheros recommends manually creating the folder **~/bin**, adding it to PATH, and placing repo there.

Qualcomm ChipCode requires git version 1.8.1.2 or higher configured for openssl. Ubuntu 12.04 installs git 1.7.8 configured for gnuTLS. The recommended solution is to download the latest git version from github.com, configure, compile, and install it manually.

5.2 Download the software

Generate the QSDK framework by re-assembling the code from ChipCode and The Linux Foundation. The example given in this section assumes that these packages are copied and downloaded:

Copy this package from ChipCode and download to the top-level directory	Download from ChipCode at
qsdk-qca-wifi-2.7.035.tar.bz2	apss_proc\out\proprietary\Wifi
qsdk-qca-wlan-2.7.035.tar.bz2	apss_proc\out\proprietary\Wifi
qsdk-qca-shortcut-fe-2.7.035.tar.bz2	apss_proc\out\proprietary\Shortcut-fe

```
$ repo init -u git://codeaurora.org/quic/qsdk/releases/manifest/qstak -b
   release -m caf_AU_LINUX_QSDK_RELEASE_BANANA_SW_TARGET_ALL.2.7.035.xml
$ repo sync
$ mkdir -p qsdk/dl
$ tar xjvf qsdk-qca-wifi-2.7.035.tar.bz2 -C qsdk
$ tar xjvf qsdk-qca-wlan-2.7.035.tar.bz2 -C qsdk
$ tar xjvf qsdk-qca-shortcut-fe-2.7.035.tar.bz2 -C qsdk
```

5.3 Build the software

1. Install the different feeds in the build framework.
   ```
   $ cd qsdk
   $ make package/symlinks
   ```
2. Copy the base configuration to use for the build. Two profiles are available: premium and wireless. In this command, replace <profile> with the profile to generate.
   ```
   $ cp qca/configs/qca955x.ln/ar71xx_<profile>.config .config
   ```
3. Regenerate a complete configuration file and start the build:
   ```
   $ make defconfig
   $ make V=s
   ```

These instructions download the packages required for the corresponding profile and create the image. Once the build is complete, the corresponding firmware should be available in the qsdk/bin/ar71xx directory (where <board> is the board supported by the generated profile):

- kernel: openwrt-ar71xx-generic-<board>-kernel.bin
- squashfs: openwrt-ar71xx-generic-<board>-rootfs-squashfs.bin

5.4 Flashing instructions

5.4.1 Setup the flashing environment

1. As a preliminary step, ensure that the board console port is connected to the PC using the following RS232 parameters:
   - 115200 bps
   - 8N1
2. Ensure that the PC is connected to the board using one of the Ethernet ports. The PC should have a TFTP server launched and listening on the interface to which the board is connected. At this stage, power up the board, and after a few seconds, press any key during the countdown.

5.4.2 Execute flashing commands for NOR: 8M platforms

All flashing commands start with the following U-Boot configuration. The board IP address as well as the TFTP server IP address must reflect the current network topology. To ensure this, execute the following commands from U-Boot:

```
setenv bc <BOARD_NAME>
setenv ipaddr <YOUR_IP@>
setenv serverip <YOUR_TFTP_SERVER_IP@>
setenv bootcmd 'bootm 0x9f680000'
tftp 0x80060000 openwrt-ar71xx-${bc}-qca-legacy-uboot.bin && erase
0x9f000000 +0x30000 && cp.b $fileaddr 0x9f000000 $filesize
setenv lok 'tftp 0x80060000 openwrt-ar71xx-generic-${bc}-kernel.bin &&
erase 0x9f680000 +${filesize} && cp.b $fileaddr 0x9f680000 0x160000'
setenv lof 'tftp 0x80060000 openwrt-ar71xx-generic-${bc}-rootfs-
squashfs.bin && erase 0x9f050000 +${filesize} && cp.b $fileaddr 0x9f050000
$filesize'
setenv lqsdk 'run lof && run lok'
saveenv
run lqsdk
```

5.4.3 Execute flashing commands for NOR: 16M platforms

All flashing commands start with the following U-Boot configuration. The board IP address as well as the TFTP server IP address must reflect the current network topology. To ensure this, execute the following commands from U-Boot:

```
setenv bc <BOARD_NAME>
setenv ipaddr <YOUR_IP@>
setenv serverip <YOUR_TFTP_SERVER_IP@>
setenv bootcmd 'bootm 0x9fe80000'
tftp 0x80060000 openwrt-ar71xx-${bc}-qca-legacy-uboot.bin && erase
0x9f000000 +0x30000 && cp.b $fileaddr 0x9f000000 $filesize
setenv lok 'tftp 0x80060000 openwrt-ar71xx-generic-${bc}-kernel.bin &&
erase 0x9fe80000 +${filesize} && cp.b $fileaddr 0x9fe80000 0x160000'
setenv lof 'tftp 0x80060000 openwrt-ar71xx-generic-${bc}-rootfs-
squashfs.bin && erase 0x9f050000 +${filesize} && cp.b $fileaddr 0x9f050000
$filesize'
setenv lqsdk 'run lof && run lok'
saveenv
run lqsdk
```

NOTE:

1. The BOARD_NAME for AP152 is either AP152-8M or AP152-16M.
2. Reset the device after flashing the U-Boot binary.

A ART2

A.1 Overview

This section describes the Qualcomm Atheros ART2 v4.9.849 package to support the chips and boards identified within this release note.

A.2 Supported functionalities

The following functionalities are supported in this build:

- Tx power calibration of a DBDC configuration (QCA9880, QCA9563)
- Saving calibration data to flash
- Saving calibration data of QCA9880 device to EEPROM
- DBDC operation system
- Users can vary the interframe spacing to effectively vary the transmitted signal duty cycle
- Power accuracy measurement using LitePoint
- Frequency accuracy measurement using LitePoint
- Spectral mask testing using LitePoint
- EVM measurement using LitePoint
- Rx sensitivity using LitePoint
- Calibration structure updates can be made via a **set** command with the **commit** command
- A GUI controlled test flow is included
- ART Windows driver to support QCA9880

A.3 New test_flow operation

The structure and operation of the `test_flow` script has changed. This section describes the changes.

A.3.1 MAC addresses

The MAC addresses are no longer calculated from the Qualcomm Atheros label. The MAC addresses are now generated from a text file that acts as a "cartridge" representing a block of MAC addresses. This file is **C:\ART2\station_files\macid.txt**, and it contains a starting address, the current address, and the final or limit address. The cart command `macid` reads the current MAC address from this file, and checks that it is not greater than the limit. If the address is less than the limit, cart increments the current address.

A.3.2 Initialization scripts

Most of the functionality that was earlier in the start.art script is now in a set of scripts in C:\ART2\station_files. These scripts are as follows:

Script	Description
conn_load.art	Contains connect and load commands for the board being processed.
directories.art	Contains the paths to the directories used for the report and log files.
equipment.art	Contains the commands describing the calibration equipment for this STA.
pathloss.art	Contains the pathloss data for this STA.
ref_devices.art	Describes the number of radios on each board being processed, as well as the internal numbers for each radio.

A.3.3 Shift operation

The `test_flow` script now runs in a loop that iterates over boards as well as radios per board. A 'shift' consists of a number of boards to be processed. The `test_flow` script should be left running in cart for the entire 'shift'. For each board, the script prompts the user for the serial number of the board, runs the `conn_load` script, and runs calibration and verification for each radio on the board. At the end of the 'shift', the user can exit and the summary of pass/fail for the shift is displayed.

For details on the new operation, see the *ART2 Reference Design for AR93xx/94xx/95xx & QCA98xx Reference Guide* (80-Y0861-1).

A.4 Known limitations

This build has the following limitations:

- Windows device drivers must be installed via the Windows device manager.
- Users can commit calibration data only a small number of times. In case of OTP storage, there are 1024 bytes of OTP available, and the typical CAL data takes up around 200 to 300 bytes.
- Composite EVM measurement is supported only for 802.11ac rates.

A.5 Commands to run NART on AP152 with QSDK

- After QSDK boots up, enter the following commands on the DUT console:

```
cd /usr/sbin
/etc/init.d/art start
```

- There are no changes to the Windows side commands.