



Application Note  
**WICED™ APOLLO**

---

## WICED™ APOLLO Wireless Audio



## Revision History

<i>Revision</i>	<i>Date</i>	<i>Change Description</i>
WICED-APOLLO-1.0	December 18, 2015	Initial revision

Broadcom Corporation  
5300 California Avenue  
Irvine, CA 92617

© 2015 by Broadcom Corporation  
All rights reserved  
Printed in the U.S.A.

## Table of Contents

1	About this Document.....	4
1.1	Purpose and Scope.....	4
1.2	Audience.....	4
2	Apollo Wireless Audio Distribution System.....	4
2.1	Audio Speaker Configurations.....	5
2.2	Stereo/2.1.....	5
2.3	5.1/7.1 Surround Sound.....	5
2.4	Equipment Needed .....	6
2.5	Building the Firmware.....	6
2.6	Configuration.....	7
2.7	Available Commands.....	8
2.8	Extra Debugging .....	11
3	iOS Apollo Audio Configuration Application.....	12
3.1	Setup .....	12
3.2	System configuration via the iOS Apollo Audio application .....	13
	.....	14
4	Roadmap.....	16

# 1 About this Document

## 1.1 Purpose and Scope

This document provides instructions to use the WICED Apollo wireless audio distribution system.



**Note:** This document applies to **WICED SDK 3.5.2** or higher.

## 1.2 Audience

This document is for software developers who are using the WICED Development System to create Wireless audio distribution systems using BCM4390x product family (e.g. BCM943909WCD1\_3, BCM943907WAE\_1, etc).

# 2 Apollo Wireless Audio Distribution System

Apollo is the software stack for implementing a whole-home multichannel Wireless Audio distribution system. Central to the design of Apollo is the concept of a sender (transmitter) or "source" device and one or more receiver or "sink devices". The system allows a source device or "source" to transmit music via Wi-Fi (over an 802.11 network) to one or more receivers (Wireless speakers) allowing for scalable speaker configurations from 2 speakers (stereo) up to 5.1 (6 speakers) and later 22.2 (UHD) surround sound. Core features in the Apollo wireless audio streaming system:

- 802.1AS Precision Time Protocol and Grand Master Clock (GMC) supporting 802.11v
- Reliable Multicast Streaming protocol for one or more speaker devices
- Audio Packet Loss Concealment
- End to end delay from 10ms to 50ms

The following configurations below are supported:

Input Source	Source Device	Sink Device(s)	Configuration	Availability
File (WAV)	Gigabyte BRIX + BCM4356	2 x BCM943907WAE_1	Stereo	Contact sales
File (WAV)	Gigabyte BRIX + BCM4356	6 x BCM943907WAE_1	5.1 Surround	Contact sales
Bluetooth A2DP	BCM943907WAE_1	BCM943907WAE_1	Stereo	WICED 3.5.2
Bluetooth A2DP	BCM943907WAE_1	2 x BCM943907WAE_1	Multi-zone (Stereo)	WICED 3.5.2
Analog Stereo	BCM943907WAE_1/BCM943909WCD1_3	BCM943907WAE_1	Stereo	WICED 3.5.2
Analog Stereo	BCM943907WAE_1/BCM943909WCD1_3	2 x BCM943907WAE_1	Multi-zone (Stereo)	WICED 3.5.2
SPDIF (Optical)	BCM943907WAE_2	6 x BCM943907WAE_1	5.1 Surround	Q2 2016

2.1 Audio Speaker Configurations

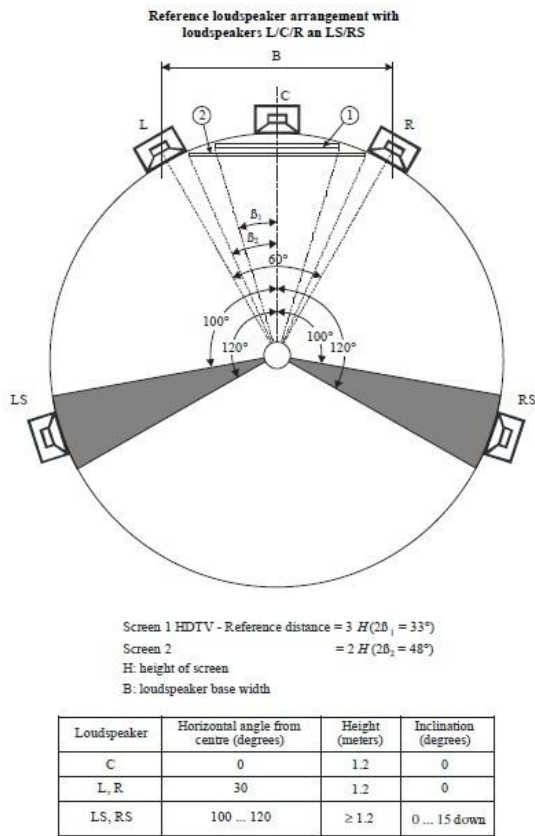
Apollo also scales and addresses the growth needs of the user as more speakers are added to the system - Stereo (2), 2.1 (3) (L/R/Bass), 6 (5.1), and larger systems may be built. Apollo supports up to 16 different speaker/channel configurations all of which may be adapted to the user’s entertainment requirements.

2.2 Stereo/2.1

The most common use case which Apollo addresses are two independent speakers (stereo) where (rather than having one receiver driving 2 speakers) each receiver renders one of the speaker pairs - a Left (L) or Right (R) channel independently. This system may be further enhanced to support a Center (C) configuration.

2.3 5.1/7.1 Surround Sound

The true benefit of Apollo comes from support for more than 2 speakers; the most relevant use case is "Surround sound" or a 5.1 system and larger. The 5.1 channel sound system has been specified in Recommendation ITU-R BS.775. The system is widely used as a part of digital broadcasting services. It enhances the directional stability of the frontal sound image and the sensation of spatial reality (ambience). The reference loudspeaker arrangement is shown below in which each loudspeaker is set at the same height as a listener’s ears.



## 2.4 Equipment Needed

- 2 Broadcom WICED platforms of the above configurations (minimum for multi-zone Stereo)
- iPhone or Android phone supporting Bluetooth
- 2 Speakers (Amplified) or Headphones supporting standard 3.5mm Audio plug
- Power connectors
- WICED Debug board

## 2.5 Building the Firmware

Follow the steps below to build the firmware image for the platforms.

### 2.5.1 Build Prerequisites

Install Apollo RMC/AVB WLAN firmware (for all 4390x platforms)

```
cd WICED-SDK-xxx/WICED-SDK/resources/firmware/43909
cp 43909B0-apollo.bin 43909B0.bin
```

Edit WICED-SDK/apps/demo/apollo/wifi\_config\_dct.h to specify SSID and channel; the defaults below are recommended for first time use:

```
/* This is the soft AP available for normal operation */
#define SOFT_AP_SSID "apollo"
#define SOFT_AP_CHANNEL 132
/* This is the default AP the device will connect to (as a client) */
#define CLIENT_AP_SSID "apollo"
#define CLIENT_AP_BSS_TYPE WICED_BSS_TYPE_ADHOC
#define CLIENT_AP_SECURITY WICED_SECURITY_OPEN
#define CLIENT_AP_CHANNEL 132
#define CLIENT_AP_BAND WICED_802_11_BAND_5GHZ
```

Note: supported 5Ghz DFS channels are: 132, 116, 108

### 2.5.2 Platform Build strings

In the WICED IDE, create the following two build targets. Build and install on two systems (not necessarily the same HW)

Wireless Audio Edition 1 (WAE\_1)

```
demo.apollo-BCM943907WAE_1.B0 download run
```

Wireless Connectivity Device 1 (WCD1\_3)

```
demo.apollo-BCM943909WCD1_3.B0 download run
```

NOTE: WCD1\_3 platforms use AKM audio output by default. You can change this with the "config ad\_tx" option (to use for example the Wolfson codec)

## 2.6 Configuration

Start by designating one system as the source device and the other as the sink device. Connect the speakers and power. Now (with the USB serial port attached), issue the below commands on each device:

### 2.6.1 Multi-Zone - Bluetooth Re-broadcaster (> 2 devices)

Source configuration

```
> config apollo_role source
> config src_t bt
> config network_name <name>
> config network_channel <channel>
> config mac <mac>
> config bt_name <name>
> config pll 1
> config save
```

Stereo Sink Configuration(s)

```
> config apollo_role sink
> config network_name <name>
> config network_channel <channel>
> config speaker_channel FL FR
> config mac <mac>
> config clock 1
> config pll 1
> config save
```

NOTE: name, channel must be the same on source/sink

NOTE: the mac address should be unique on each board

Finally reboot each board after “config save”. Use your iPhone/Android phone and discover the device under your Bluetooth settings menu. Connect and play music. You will hear the music on both the source and sink devices.

### 2.6.2 Multi-Zone - Analog Re-broadcaster

Source configuration

```
> config apollo_role source
> config src_t capture
> config network_name <name>
> config network_channel <channel>
> config mac <mac>
> config save
```

Stereo Sink Configuration

```
> config apollo_role sink
```

```
> config network_name <name>
> config network_channel <channel>
> config speaker_channel FL FR
> config mac <mac>
> config clock 1
> config pll 1
> config save
```

Connect the audio input to the source device. As before, reboot the system and start playback.

### 2.6.3 Multi-Speaker

To support multiple speakers, each sink device is designated a channel which it would play. For a stereo system you would need 1 set of speakers on the source (playing stereo (FL/FR)) and a speaker attached to 2 more sink devices (one playing left – FL, the other playing right – FR).

Sink Configuration

```
> config speaker_channel FL FR
> config save
```

### 2.6.4 5.1 Surround sound

With BRIX platforms, the following channel mappings must be established on each sink device.

```
> config speaker_channel FL
> config save
```

```
> config speaker_channel FR
> config save
```

```
> config speaker_channel FC
> config save
```

```
> config speaker_channel LFE
> config save
```

```
> config speaker_channel BL
> config save
```

```
> config speaker_channel BR
> config save
```

## 2.7 Available Commands

Use the “config help” command for details on console commands.

```
> config help
```



## WICED™ APOLLO

---

Config commands:

```
config                                : output current config

config <?|help>                       : show this list

config auto_start <0|off|1|on>        : 0 = auto start off, 1 = auto start
on                                     :

    auto <0|off|1|on>

config buffering_ms <xxx>             : xxx = milliseconds

    buff_ms <xxx>                     : (range:0 <= xx <= 1000)

config clock <0|disable|1|enable>     : 0 = disable AS clock, 1 = enable

config pll <0|disable|1|enable>       : 0 = disable audio PLL tuning, 1 =
enable

config pll_tuning <0|disable|1|enable>

config mac_addr <xx:xx:xx:xx:xx:xx>   : xx:xx:xx:xx:xx:xx = new MAC address

config mac <xx:xx:xx:xx:xx:xx>       : Shortcut:

config mac <xx>                       : enter 1 octet to change last
octet

config network_channel <xxx>          : xxx = channel

    net_chan <xxx>                   : (1-11,36,40,44,48,52,56,60,64,
                                     : 100,104,108,112,116,120,124,128,
                                     : 132,136,140,149,153,157,161,165)

config network_name <ssid_name>       : name of AP (max 32 characters)

    net_name <ssid_name>

    ssid <ssid_name>

config network_passphrase <pass>      : passkey/password (max 64
characters)

    net_pass <pass>

    pass <pass>

config network_security <type>        : security type is one of:

    net_sec <type>                   :
open,none,ibss,wep,wepshared,wpaatkip,wpaaes,
```

## WICED™ APOLLO

---

```

:
wpa2tkip,wpa2aes,wpa2mix,wpsopen,wpsaes

    config speaker_name <name>                : speaker name (max 36 characters)

        spkr_name <name>

    config speaker_channel <ch> [ch]... : channel mix - all will be OR'ed
together

        spkr_chan    <ch>    [ch]...          :
FL,FR,FC,LFE1,BL,BR,FLC,FRC,BC,LFE2,

:

SIL,SIR,TPFL,TPFR,TPFC,TPC,TPBL,TPBR,

:

        TPSIL,TPSIR,TPBC,BTFC,BTFL,BTFR

    config threshold_ms <xx>                : xx = milliseconds

        thresh_ms <xx>                    :      (range:0 <= xx <= 1000)

    config volume <xx>                      : xx = volume level

        vol <xx>                          :      (range:0 <= xx <= 100)

    config payload_size <size_in_bytes> : from 64 to 1432 bytes

    config fec_order    <string>            : "pre" = before audio, "post" =
after audio

    config fec_length   <packet_count>     : from 0 to 16

    config fec          <packet_count>

    config source_type  <string>            : "bt" = BT A2DP audio, "capture" =
using local ADC

    config src_t        <string>

    config audio_device_rx <device_X>      : enter X as in WICED_AUDIO_X, X
starts at 0 for all WICED platforms

    config ad_rx        <device_X>

    config audio_device_tx <device_X>      : enter X as in WICED_AUDIO_X, X
starts at 0 for all WICED platforms

    config ad_tx        <device_X>

    config apollo_role <source | sink>     : Configure as a source or a sink

    config role <source | sink>

    config clientaddr <IP address>        : Client IP address for sender to use
```

```
config addr <IP address>

config rtp_port <port number>          : RTP port number

config port <port number>

config log_level <level>               : Set the default application logging
level

config log <level>

config is_configured <0|no|1|yes>      : Set to 0 to force BT GATT
configuration

config is_conf <0|no|1|yes>

config bt_name <name>                  : Set the bluetooth device name

config bt_dev <name>

config bt_mac <xx:xx:xx:xx:xx:xx>      : xx:xx:xx:xx:xx:xx = new Bluetooth
MAC address

config bt_mac <xx>                     :   enter 1 octet to change last
octet

config bt_class <xx:xx:xx>             : xx:xx:xx = new Bluetooth device
class

config save                             : save data to flash NOTE: Changes
not

                                     :   automatically saved to flash!

>
```

## 2.8 Extra Debugging

Use the log command to get additional information on the source or sink. To make the setting persistent across reboot, use the “config loglevel” command. Valid values are 0 (no messages) to 10 (full debug messages). NOTE: when increasing the log-level for debug, audio artifacts may be heard due to too many UART interrupts during playback. This will be addressed in future releases.

```
> log 10
Setting new log level to 10 (0 - off, 9 - max debug)
>
```

### Sink Configuration

```
> config log_level 3
> config save
```

## 3 iOS Apollo Audio Configuration Application

The iOS Apollo Configuration App demonstrates how to configure Wi-Fi and speaker settings of WICED Audio based speakers running the Apollo Wireless Audio stack vthrough Bluetooth Low Energy (BLE) on an iOS device.

### 3.1 Setup

- BCM43907WAE\_1 or BCM43907WAE\_2 WICED Audio boards. Up to 6 can be configured at a time. Follow the build instructions above.
- Install WICED-SDK-3.5.2 or later on your computer. NOTE: The Apollo iOS application is distributed as a patch available upon request from Broadcom.
- Apple Mac
  - Clone and build the "Apollo Audio Config" source in Xcode from the SDK release 3.5.2 or later plus patch located in apps\demo\apollo\peerapps
- iPhone or iPad
  - Connect the iPad/iPhone to the Mac PC and install the iOS application
- WiFi Access point with internet connection.

Once the system starts, the application will come up as a remote speaker in its default configuration and you should see console output below

```
Config Info: * = dirty
```

```
Apollo app DCT:
```

```
is configured: yes
```

```
apollo role: sink
```

```
speaker name: Apollo
```

```
channel: (0x00000003) FL FR
```

```
buffering: 50 ms
```

```
threshold: 40 ms
```

```
auto_start: on
```

```
clock enable: enable

PLL tuning enable: enable

volume: 50

source type: bt

payload size: 1432
```

## 3.2 System configuration via the iOS Apollo application

- Step 1: Press the "BACK" button on the BCM43907WAE\_1 or BCM43907WAE\_2 WICED Audio boards. The Apollo audio application starts up, resets the device configuration and then launches a BLE GATT server advertising the Apollo service. You should see the console output below:

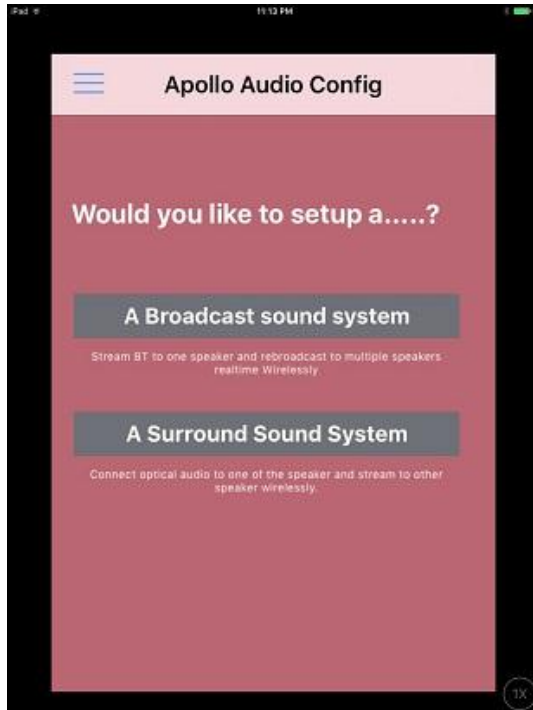
```
0000 00:00:02.217 *** Waiting for GATT configuration... ***

wiced_bt_ble_set_advertisement_data 0
```

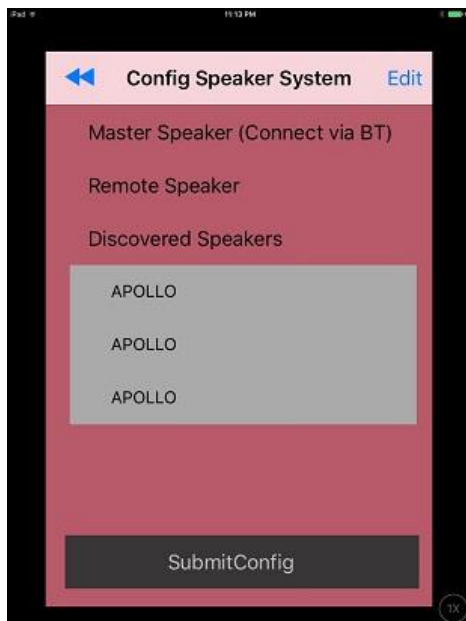
- Step 2: On the iPhone or iPad, turn BT ON. Open the installed Apollo application whose icon appears as below:



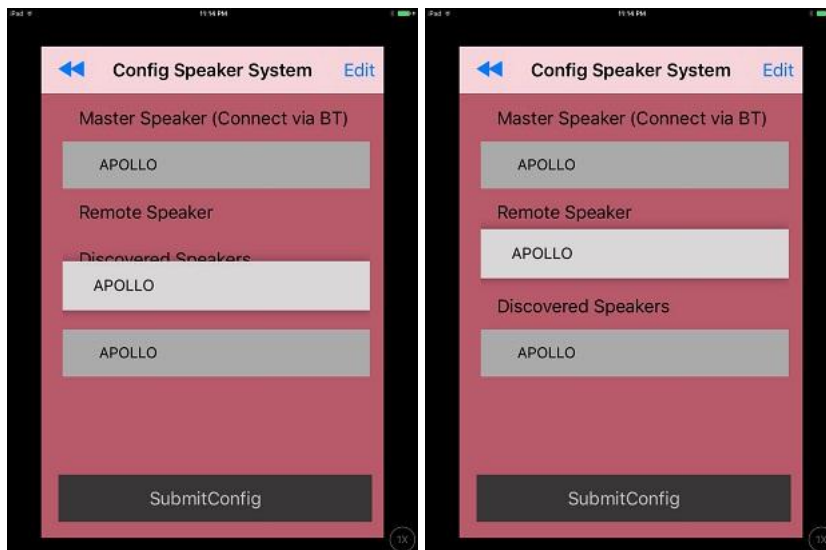
- Step 3: The application's initial home screen will come up. Note: If the screen does not come up there are no Speaker devices present running in GATT configuration mode. Separately, if no devices are found, confirm that BT is enabled on your iPhone/iPad.



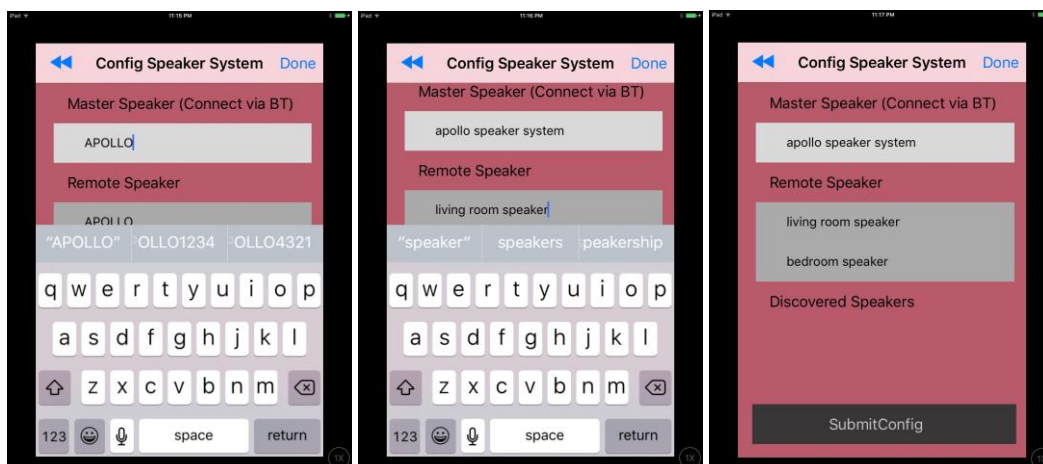
- Step 4: The iOS app only configures the Broadcast sound system in the first release. Press the button titled "Configure A Broadcast sound system". This brings up the screen as below. Note: Pressing "Configure A Surround Sound System" will not do anything.



- Step 5: In the above screen, the app lists all the discovered WICED devices which are running Apollo firmware in configuration mode. If you do not see a device or only one device being listed in the "Discovered Speakers" Section please go back to Step 1 to insure all devices are in the configuration state.
- Step 6: With at least two devices listed under the Discovered Speakers section, drag and drop each of these devices to the master and remote speaker section respectively. Tap and hold on the device in the Discovered speakers area and then start dragging to a destination source device. Make sure the Master Speaker section and the Remote speaker section have at least one device. The following screens below show the drag and drop association in action:



- Step 7: Tap on the Edit button in the top right corner, this brings up the soft keyboard to enter the Network Name in the Master Speaker Section. Once entered tap the enter key in the soft keyboard of iPad to close the keyboard. Now similarly enter speaker names for the speakers in Remote Section by tapping on each of them (and which pops up the soft keyboard). The following screens shows all of this in action:



- Step 8: Once done tap the "Done" button in the top right corner to signal the application that you are Done with the Editing of the devices.

- ```

writing HANDLE_ARMOLO_CONFIG_SERVICE_CHAS_MODE_VAL
spollo_gett_dct_mode = 0

writing HANDLE_ARMOLO_CONFIG_SERVICE_CHAS_ID_VAL
spollo_gett_dct_id_chas_mode = ARMOLO

writing HANDLE_ARMOLO_CONFIG_SERVICE_CHAS_PASSHARSHAL_VAL
spollo_gett_dct_service_chas_passhars = 12345678

writing HANDLE_ARMOLO_CONFIG_SERVICE_SERVICE_CHAS_ID_VAL
spollo_gett_dct_service_type = 1

writing HANDLE_ARMOLO_CONFIG_SERVICE_SERVICE_CHAS_ID_VAL
spollo_gett_dct_service_val = 80

load_by_file_get_advertisement_data 0

Starting NICE v1.0 DEVELOPMENT
Extract NICE327FWAL 1.01 installed
Started Thread v1.0
Initializing NICE_Drv v1.0
Creating Packet Pool
NICE ARMOLO interface installed
NICE MAC Address is 10:10:10:10:10:10
NICE IP Address is 192.168.0.1
NICE IP Netmask is 255.255.255.0
NICE IP Gateway is 192.168.0.1
NICE IP DNS is 192.168.0.1
NICE IP DNS2 is 192.168.0.1
NICE IP DNS3 is 192.168.0.1
NICE IP DNS4 is 192.168.0.1
NICE IP DNS5 is 192.168.0.1
NICE IP DNS6 is 192.168.0.1
NICE IP DNS7 is 192.168.0.1
NICE IP DNS8 is 192.168.0.1
NICE IP DNS9 is 192.168.0.1
NICE IP DNS10 is 192.168.0.1
NICE IP DNS11 is 192.168.0.1
NICE IP DNS12 is 192.168.0.1
NICE IP DNS13 is 192.168.0.1
NICE IP DNS14 is 192.168.0.1
NICE IP DNS15 is 192.168.0.1
NICE IP DNS16 is 192.168.0.1
NICE IP DNS17 is 192.168.0.1
NICE IP DNS18 is 192.168.0.1
NICE IP DNS19 is 192.168.0.1
NICE IP DNS20 is 192.168.0.1
NICE IP DNS21 is 192.168.0.1
NICE IP DNS22 is 192.168.0.1
NICE IP DNS23 is 192.168.0.1
NICE IP DNS24 is 192.168.0.1
NICE IP DNS25 is 192.168.0.1
NICE IP DNS26 is 192.168.0.1
NICE IP DNS27 is 192.168.0.1
NICE IP DNS28 is 192.168.0.1
NICE IP DNS29 is 192.168.0.1
NICE IP DNS30 is 192.168.0.1
NICE IP DNS31 is 192.168.0.1
NICE IP DNS32 is 192.168.0.1
NICE IP DNS33 is 192.168.0.1
NICE IP DNS34 is 192.168.0.1
NICE IP DNS35 is 192.168.0.1
NICE IP DNS36 is 192.168.0.1
NICE IP DNS37 is 192.168.0.1
NICE IP DNS38 is 192.168.0.1
NICE IP DNS39 is 192.168.0.1
NICE IP DNS40 is 192.168.0.1
NICE IP DNS41 is 192.168.0.1
NICE IP DNS42 is 192.168.0.1
NICE IP DNS43 is 192.168.0.1
NICE IP DNS44 is 192.168.0.1
NICE IP DNS45 is 192.168.0.1
NICE IP DNS46 is 192.168.0.1
NICE IP DNS47 is 192.168.0.1
NICE IP DNS48 is 192.168.0.1
NICE IP DNS49 is 192.168.0.1
NICE IP DNS50 is 192.168.0.1
NICE IP DNS51 is 192.168.0.1
NICE IP DNS52 is 192.168.0.1
NICE IP DNS53 is 192.168.0.1
NICE IP DNS54 is 192.168.0.1
NICE IP DNS55 is 192.168.0.1
NICE IP DNS56 is 192.168.0.1
NICE IP DNS57 is 192.168.0.1
NICE IP DNS58 is 192.168.0.1
NICE IP DNS59 is 192.168.0.1
NICE IP DNS60 is 192.168.0.1
NICE IP DNS61 is 192.168.0.1
NICE IP DNS62 is 192.168.0.1
NICE IP DNS63 is 192.168.0.1
NICE IP DNS64 is 192.168.0.1
NICE IP DNS65 is 192.168.0.1
NICE IP DNS66 is 192.168.0.1
NICE IP DNS67 is 192.168.0.1
NICE IP DNS68 is 192.168.0.1
NICE IP DNS69 is 192.168.0.1
NICE IP DNS70 is 192.168.0.1
NICE IP DNS71 is 192.168.0.1
NICE IP DNS72 is 192.168.0.1
NICE IP DNS73 is 192.168.0.1
NICE IP DNS74 is 192.168.0.1
NICE IP DNS75 is 192.168.0.1
NICE IP DNS76 is 192.168.0.1
NICE IP DNS77 is 192.168.0.1
NICE IP DNS78 is 192.168.0.1
NICE IP DNS79 is 192.168.0.1
NICE IP DNS80 is 192.168.0.1
NICE IP DNS81 is 192.168.0.1
NICE IP DNS82 is 192.168.0.1
NICE IP DNS83 is 192.168.0.1
NICE IP DNS84 is 192.168.0.1
NICE IP DNS85 is 192.168.0.1
NICE IP DNS86 is 192.168.0.1
NICE IP DNS87 is 192.168.0.1
NICE IP DNS88 is 192.168.0.1
NICE IP DNS89 is 192.168.0.1
NICE IP DNS90 is 192.168.0.1
NICE IP DNS91 is 192.168.0.1
NICE IP DNS92 is 192.168.0.1
NICE IP DNS93 is 192.168.0.1
NICE IP DNS94 is 192.168.0.1
NICE IP DNS95 is 192.168.0.1
NICE IP DNS96 is 192.168.0.1
NICE IP DNS97 is 192.168.0.1
NICE IP DNS98 is 192.168.0.1
NICE IP DNS99 is 192.168.0.1
NICE IP DNS100 is 192.168.0.1
NICE IP DNS101 is 192.168.0.1
NICE IP DNS102 is 192.168.0.1
NICE IP DNS103 is 192.168.0.1
NICE IP DNS104 is 192.168.0.1
NICE IP DNS105 is 192.168.0.1
NICE IP DNS106 is 192.168.0.1
NICE IP DNS107 is 192.168.0.1
NICE IP DNS108 is 192.168.0.1
NICE IP DNS109 is 192.168.0.1
NICE IP DNS110 is 192.168.0.1
NICE IP DNS111 is 192.168.0.1
NICE IP DNS112 is 192.168.0.1
NICE IP DNS113 is 192.168.0.1
NICE IP DNS114 is 192.168.0.1
NICE IP DNS115 is 192.168.0.1
NICE IP DNS116 is 192.168.0.1
NICE IP DNS117 is 192.168.0.1
NICE IP DNS118 is 192.168.0.1
NICE IP DNS119 is 192.168.0.1
NICE IP DNS120 is 192.168.0.1
NICE IP DNS121 is 192.168.0.1
NICE IP DNS122 is 192.168.0.1
NICE IP DNS123 is 192.168.0.1
NICE IP DNS124 is 192.168.0.1
NICE IP DNS125 is 192.168.0.1
NICE IP DNS126 is 192.168.0.1
NICE IP DNS127 is 192.168.0.1
NICE IP DNS128 is 192.168.0.1
NICE IP DNS129 is 192.168.0.1
NICE IP DNS130 is 192.168.0.1
NICE IP DNS131 is 192.168.0.1
NICE IP DNS132 is 192.168.0.1
NICE IP DNS133 is 192.168.0.1
NICE IP DNS134 is 192.168.0.1
NICE IP DNS135 is 192.168.0.1
NICE IP DNS136 is 192.168.0.1
NICE IP DNS137 is 192.168.0.1
NICE IP DNS138 is 192.168.0.1
NICE IP DNS139 is 192.168.0.1
NICE IP DNS140 is 192.168.0.1
NICE IP DNS141 is 192.168.0.1
NICE IP DNS142 is 192.168.0.1
NICE IP DNS143 is 192.168.0.1
NICE IP DNS144 is 192.168.0.1
NICE IP DNS145 is 192.168.0.1
NICE IP DNS146 is 192.168.0.1
NICE IP DNS147 is 192.168.0.1
NICE IP DNS148 is 192.168.0.1
NICE IP DNS149 is 192.168.0.1
NICE IP DNS150 is 192.168.0.1
NICE IP DNS151 is 192.168.0.1
NICE IP DNS152 is 192.168.0.1
NICE IP DNS153 is 192.168.0.1
NICE IP DNS154 is 192.168.0.1
NICE IP DNS155 is 192.168.0.1
NICE IP DNS156 is 192.168.0.1
NICE IP DNS157 is 192.168.0.1
NICE IP DNS158 is 192.168.0.1
NICE IP DNS159 is 192.168.0.1
NICE IP DNS160 is 192.168.0.1
NICE IP DNS161 is 192.168.0.1
NICE IP DNS162 is 192.168.0.1
NICE IP DNS163 is 192.168.0.1
NICE IP DNS164 is 192.168.0.1
NICE IP DNS165 is 192.168.0.1
NICE IP DNS166 is 192.168.0.1
NICE IP DNS167 is 192.168.0.1
NICE IP DNS168 is 192.168.0.1
NICE IP DNS169 is 192.168.0.1
NICE IP DNS170 is 192.168.0.1
NICE IP DNS171 is 192.168.0.1
NICE IP DNS172 is 192.168.0.1
NICE IP DNS173 is 192.168.0.1
NICE IP DNS174 is 192.168.0.1
NICE IP DNS175 is 192.168.0.1
NICE IP DNS176 is 192.168.0.1
NICE IP DNS177 is 192.168.0.1
NICE IP DNS178 is 192.168.0.1
NICE IP DNS179 is 192.168.0.1
NICE IP DNS180 is 192.168.0.1
NICE IP DNS181 is 192.168.0.1
NICE IP DNS182 is 192.168.0.1
NICE IP DNS183 is 192.168.0.1
NICE IP DNS184 is 192.168.0.1
NICE IP DNS185 is 192.168.0.1
NICE IP DNS186 is 192.168.0.1
NICE IP DNS187 is 19
```

[illegible]

```

nls000_get_dev_mode = 1
writing SAMPLE_AGGLO_CFGFWS_MW_SERVICE_CHAN_SSID_VAL
agglo_get_dev_name_ssid_name = AGGLO
writing SAMPLE_AGGLO_CFGFWS_MW_SERVICE_CHAN_PASSPHRASE_VAL
agglo_get_dev_name_pass_phrase = 1234567
writing SAMPLE_AGGLO_CFGFWS_SPEAKER_SERVICE_CHAN_NAME_VAL
agglo_get_dev_spk_name = AGGLO
writing SAMPLE_AGGLO_CFGFWS_SPEAKER_SERVICE_CHAN_NAME_VAL
agglo_get_dev_spk_channel_name = 3
writing SAMPLE_AGGLO_CFGFWS_SPEAKER_SERVICE_CHAN_OUTPUT_VOL_VAL
agglo_get_dev_spk_advertisement_data = 50
agglo_get_dev_spk_advertisement_data = 50
writing MICKED v3.0.30/ELMENT
Platform ARMv7/Rev_1.3.0 initialized
Scatted Thread# 0-6
Creating packets MCK_DEV_0-5/7-9
Creating packets PCKS
MW SOC43909 interface initialized
MW SOC Address : 8018123456789
Mick Firmware : v101 Dec 4 2013 14:11:17 version 7.13.148.3 (T2MD 01-76d916
)
Config Info : "d12ty
Agilo app MCK
Is configured: yes
agilo peer MCK
speaker name: Agglo
channel: (0000000000) PL FR
buffering: 50 ms
threshold: 40 ms
auto_start: on
clock enable: enable
Pis tuning enable: enable
Volume: 50
source input 1
payload size: 1432
FEC code type:
FEC length: 9
Audio Input Device(s):
Pckl 44100 Hz 4 conductor 3.5mm 0 23
Audio Output Device(s):
Pckl 44100 Hz 4 conductor 3.5mm 0 23
client IP address: 224.0.0.5
RTT peer: 5004
log_level: 1 (currently 1)
Network DCT:
Interface: STA
IP addr: 0.0.0.0
RSSI: 0 dbm
WiFi DCT:
WIFI MAC (STA): 8018123456789abc123456
WIFI MAC (P2P): f1123456789abc123456
SSID: AGGLO
Security: open
Passphrase: none
Channel: 149
Radio Mode
BT DCT:
BT Device name: Agglo BT_Support
BT MAC: f1123456789abc123456
BT Device class: 2010610
Current:
Channel 1
Mode 2.4024
Pairing : AGGLOspk_speaker_name[2345678]
Successfully joined : AGGLOspk_speaker_name[2345678]
Receiving IP4 address via DHCP
MCK CLIENT hostname Agglo
IP4v6 on checks If active via DHCP, Receiving AUTO_IP
IP4v6 AUTO IP network ready IP: 169.254.76.184
Setting IP4v6 link-local address
IP4v6 network ready IP: FE80::0000:0000:0000:0000::FE80::FE80::FE80::FE80
SOC 8018123456789 *****Chip version is 81

```

## Future support for SPDIF and surround sound with BCM943907\_WAE1

Connecting  
everything®





**BROADCOM CORPORATION**

5300 California Avenue

Irvine, California, 92677

© 2015 by BROADCOM CORPORATION. All rights reserved.

Phone : +1-949-926-5000

Fax: +1-949-926-5203

E-mail: [info@broadcom.com](mailto:info@broadcom.com)

Web: [www.broadcom.com](http://www.broadcom.com)

WICED-APOLLO

December 18, 2015