GB9619 User Guide for Android

Revision History

Date	Version	Description	Author
2010/07/09	0.1	Initial revision of WLAN function 1. Driver version: 4.218.195.0	Terence Hsieh
2010/07/25	0.2	1. Refine the user guide 2. Add WLAN power saving mode	Fred Chen
2010/08/18	0.3	Add SoftAP guide	Terence Hsieh

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

INTRODUCTION

This user guide is intended to give GB9619 users a general guide of how to enable the WLAN function and we'll talk about the power saving mode as well.

WLAN SOFTWARE ARCHITECTURE OVERVIEW

GB9619 WLAN DONGLE BASIC CONCEPT

The GB9619 WLAN software package contains the dongle host driver for the host, a downloadable binary image for the GB9619, and management utilities.

The wireless driver runs on the GB9619 dongle. The SDIO host controller passes IEEE 802.3 packets, and the necessary control packets, back and forth over the SDIO bus. A special Broadcom Device Class protocol is used to encapsulate control packets on a separate logical control channel and to add packet information to the data channel.

The advantage of using the dongle concept is that the wireless driver is executed externally from a host device, which means the host device does not have to use CPU or memory resources in order to execute the wireless driver's functionality. The use of the dongle provides the following benefits to the host:

- Power savings
- A reduction in driver size and complexity
- Processor offloading for activities such as checksum calculation and Address Resolution Protocol (ARP) execution

WLAN DONGLE OVERVIEW

The Dongle Host Driver (DHD) is the executable module that provides encapsulated communication between the host device and the GB9619 module over the SDIO bus.

The dongle software architecture is based on two major components:

- Dongle Host Driver: A host-based driver used to provide a communication channel with the dongle device firmware.
- User-space configuration utilities, WL and DHD: These executable binaries are called "wl" and "dhd" respectively.

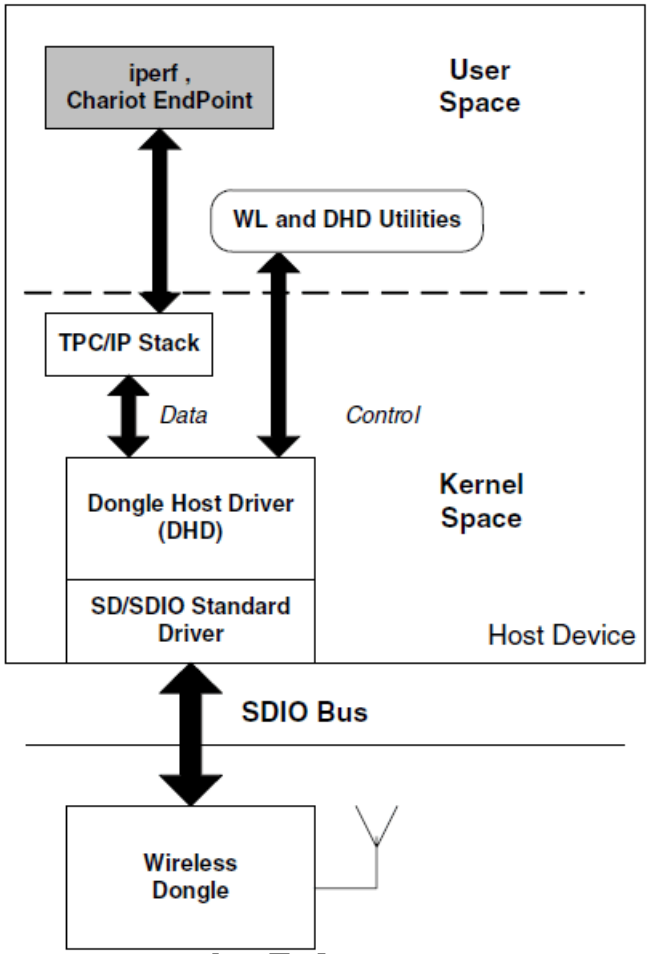

Figure 1: GB9619 SDIO WLAN Dongle Concept

WLAN SOFTWARE PACKAGE

The provided WLAN software package contains following files:

- Dongle host driver (dhd.ko)
- Dongle device firmware (sdio-g-cdc-full11n-reclaim-roml-wme.bin)
- User space configuration utility (dhd and wl)
- nvram.txt

WLAN DRIVER INSTALLATION

ENABLE WIRELESS EXTENSION OF LINUX KERNEL

Please add following items into your kernel configuration:
CONFIG_NET=y
CONFIG_WIRELESS=y
CONFIG_CFG80211=y
CONFIG_NL80211=y
CONFIG_WIRELESS_EXT=y
CONFIG_WIRELESS_EXT_SYSFS=y

INSTALL DONGLE HOST DRIVER

insmod dhd.ko
./dhd -i [Interface Name] download sdio-g-cdc-full11n-reclaim-roml-wme.bin nvram.txt
ifconfig [Interface Name] up
./wl up

```
#
# cd /data/tmp
# pwd
/data/tmp
# ls -l
-rwxrwxrwx root     root       239104 2010-06-18 06:01 sdio-g-cdc-fullll1n-reclaim-roml-wme.bin
-rwxrwxrwx root     root       236356 2010-04-27 19:23 wl
-rwxrwxrwx root     root         1405 2010-06-28 18:34 nvram.txt
-rwxrwxrwx root     root      1712426 2010-07-05 07:59 dhd.ko
-rwxrwxrwx root     root       520724 2010-04-27 19:23 dhd
-rwxrwxrwx root     root        35592 2010-04-27 19:23 iwlist
-rwxrwxrwx root     root        31156 2010-04-27 19:23 iwconfig
#
# insmod dhd.ko
wlan0 (): not using net_device_ops yet
wlan0: Broadcom Dongle Host Driver

Dongle Host Driver, version 4.218.195.0
#
# ./dhd -i wlan0 download sdio-g-cdc-fullll1n-reclaim-roml-wme.bin nvram.txt
#
# ifconfig wlan0 up
# ./wl up
#
```

Figure 2: Install dongle host driver

WLAN OPERATION

SCAN NETWORK

#./iwlist [Interface Name] scan

```
# ./iwlist wlan0 scan
wlan0     Scan completed :
          Cell 01 - Address: 00:0A:79:BF:EE:D0
                    ESSID:"tttb"
                    Mode:Managed
                    Frequency:2.422 GHz (Channel 3)
                    Quality:5/5  Signal level:-57 dBm  Noise level:-92 dBm
                    IE: Unknown: DD830050F204104A0001101044000102103B0001031047
00102880288028801880A880000A79BFEED01021000B436F72656761204B2E4B2E1023000B43472
D574C424152474E5310240008574C424152474E5310420008313233343536373810540008000600
50F2040001101000B43472D574C424152474E5310080002008A103C000101
                    Encryption key:off
                    Bit Rates:1 Mb/s; 2 Mb/s; 5.5 Mb/s; 11 Mb/s; 9 Mb/s
                              18 Mb/s; 36 Mb/s; 54 Mb/s; 6 Mb/s; 12 Mb/s
                              24 Mb/s; 48 Mb/s
          Cell 02 - Address: 00:0A:79:BF:EE:D1
                    ESSID:"CG-Guest"
                    Mode:Managed
                    Frequency:2.422 GHz (Channel 3)
                    Quality:5/5  Signal level:-57 dBm  Noise level:-92 dBm
                    Encryption key:off
                    Bit Rates:1 Mb/s; 2 Mb/s; 5.5 Mb/s; 11 Mb/s; 9 Mb/s
                              18 Mb/s; 36 Mb/s; 54 Mb/s; 6 Mb/s; 12 Mb/s
                              24 Mb/s; 48 Mb/s
          Cell 03 - Address: 00:26:87:03:F9:05
                    ESSID:"00268703F905_2nd"
                    Mode:Managed
                    Frequency:2.457 GHz (Channel 10)
                    Quality:2/5  Signal level:-74 dBm  Noise level:-92 dBm
                    Encryption key:on
                    Bit Rates:1 Mb/s; 2 Mb/s; 5.5 Mb/s; 11 Mb/s; 9 Mb/s
                              18 Mb/s; 36 Mb/s; 54 Mb/s; 6 Mb/s; 12 Mb/s
                              24 Mb/s; 48 Mb/s
```

Figure 3: Scan WLAN network

CONNECT TO AP

./iwconfig [Interface Name] essid off // reset essid
./iwconfig [Interface Name] mode managed // set to infrastructure mode
./iwconfig [Interface Name] essid tttb // connect to tttb

```
#
# ./iwconfig wlan0 essid off
# ./iwconfig wlan0 mode managed
# ./iwconfig wlan0 essid tttb
# ./iwconfig wlan0
wlan0     IEEE 802.11-DS  ESSID:"tttb"  Nickname:""
          Mode:Managed  Frequency:2.422 GHz  Access Point: 00:0A:79:BF:EE:D0
          Bit Rate=72 Mb/s   Tx-Power:32 dBm
          Retry min limit:7   RTS thr:off   Fragment thr:off
          Encryption key:off
          Power Management:off
          Link Quality=4/5  Signal level=-58 dBm  Noise level=-57 dBm
          Rx invalid nwid:0  Rx invalid crypt:0  Rx invalid frag:0
          Tx excessive retries:45  Invalid misc:0   Missed beacon:0

# ifconfig wlan0 192.168.1.199 netmask 255.255.255.0
# netcfg
lo       UP    127.0.0.1       255.0.0.0       0x00000049
wlan0    UP    192.168.1.199   255.255.255.0   0x00001043
#
# ping -c 3 192.168.1.1
PING 192.168.1.1 (192.168.1.1) 56(84) bytes of data.
64 bytes from 192.168.1.1: icmp_seq=2 ttl=64 time=1762 ms

--- 192.168.1.1 ping statistics ---
3 packets transmitted, 1 received, 66% packet loss, time 2014ms
rtt min/avg/max/mdev = 1762.383/1762.383/1762.383/0.000 ms, pipe 2
# ping -c 3 192.168.1.1
PING 192.168.1.1 (192.168.1.1) 56(84) bytes of data.
64 bytes from 192.168.1.1: icmp_seq=1 ttl=64 time=271 ms
64 bytes from 192.168.1.1: icmp_seq=3 ttl=64 time=242 ms

--- 192.168.1.1 ping statistics ---
3 packets transmitted, 2 received, 33% packet loss, time 2009ms
rtt min/avg/max/mdev = 242.163/256.588/271.014/14.434 ms
#
```

Figure 4: Connect to WLAN AP

WLAN POWER SAVING MODE

There're three different power saving mode settings as follows:

- PM_OFF
  - The driver is not in Power Saving mode
  - Can be activated by following command
    ./wl PM 0
- PM_MAX
  - The driver is in Maximum Power Saving mode. The driver always goes into Sleep mode and uses the PS_POLL mechanism to retrieve packets from the AP. Performance is sacrificed for maximum power savings.
  - Can be activated by following command
    ./wl PM 1
- PM_FAST
  - Fast Power Saving mode. As long as there're active data transfers, the driver does not go into Sleep mode. After data traffic stops, driver goes into Sleep mode. This allows for power savings in IDLE times, but provides full performance when needed.
  - Can be activated by following command
    ./wl PM 2

SOFTAP FUNCTIONALITY OVERVIEW

Many smart phones have high-bandwidth cellular data connections using technologies such as HSPA, UMTS, EVDO, and so on. In addition to provide access to network-based services from the phone itself, many carriers want to enable their customers to use the connection for other WLAN-enabled devices, such as computers and handheld video games.

For other devices to connect to the cellular phone, the WLAN chipset must be configured for SoftAP operation. Multiple WLAN-enabled devices can then connect to the cellular phone and share the connection. This capability is known as tethering.

Several other protocols are required, however, before the WLAN-enabled devices can successfully share a connection with the cellular phone. These protocols include DHCP (for giving associated devices their own IP address) and Network Address Translation (NAT). There protocols are layer-3 protocols that exist entirely above Ampak GB9619 SoftAP implementation.

Following table introduce the SoftAP features:

Feature	Description
Stations supported	8
Station power save support	IEEE and WMM-PS
Security	Open, WEP, WPA-PSK(TKIP), and WPA2-PSK(TKIP+AES)
WEP keys supported	4
SSID broadcast disable	Yes
Allow/deny list	Yes, through MAC address filtering
Association station list	Yes
Limit station associations	Yes, the maximum = 8

Table 1: SoftAP Features

The vast majority of the functionality required to implement a SoftAP solution is implemented in the firmware that is executed by the on-chip processor. Consequently, the software and CPU load on the host is relatively small.

SOFTAP FIRMWARE

If the target device requires SoftAP support, a firmware file with SoftAP support must be downloaded to the chipset. The firmware binary file must have an ap in the file name, such as
sdio-g-cdc-full11n-reclaim-roml-apsta-idsup-idauth.bin.

SOFTAP OPERATION

The following command sequence can be used as a template for bringing up a SoftAP on the primary WLAN interface.

```
/* Enable SoftAP mode */
./wl mpc 0
./wl down
./wl ap 1
/* Set the operating channel */
./wl channel 11
./wl up

/* For open authentication, no security */
./wl wsec 0
./wl wpa_auth 0

/* OR for WEP security */
./wl wsec x
./wl addwep 0 xxxxxxxxxx
./wl wpa_auth x

/* OR for WPA-Personal security */
./wl wsec x                  /* 2 for TKIP, 4 for AES, 6 for both */
./wl addwep 0 xxxxxxxxxx  /* raw 64-byte HEX PMK */
./wl wpa_auth x              /* 4 for WPA-PSK, 128 for WPA2-PSK, 132 for both
*/

/* Set maximum allowed connections */
```

```
./wl maxassoc 8
/* Set allow/deny MAC address list */
./wl macmode 1              /* 0 = disable, 1 = allow, 2 = deny */
./wl mac xx:xx:xx:xx:xx:xx [xx:xx:xx:xx:xx:xx …]

/* if desired, disable SSID broadcast */
./wl closed 0               /* 0 = open, 1 = hidden */
./wl ssid xxxxx             /* set SSID, enable BSS */
```