Application Note

User Guide for ATWILC3000 on Linux system

AN-XXXX

Prerequisites

- **Hardware Prerequisites**
  - Linux host target platform
  - Atmel WILC3000 evaluation board
  - Micro USB Cable (TypeA / MicroB)
  - USB to Serial Adaptor (for DBGU port)

- **Build Prerequisites**
  - Linux Host PC
  - Linux Software Package

Introduction

The ATWILC3000 is a single chip IEEE 802.11 b/g/n RF, baseband, MAC, Bluetooth 4.0 and FM receiver optimized for low-power mobile applications. The WILC3000 utilizes highly optimized 802.11 – Bluetooth coexistence protocols. It provides multiple peripheral interfaces including UART, SPI, I2C and SDIO.

This application note describes how to integrate the Atmel WILC3000 evaluation board via SDIO for Wi-Fi and UART for Bluetooth. In addition, this application note introduces the wireless tools for example, **iw** command and **WPA supplicant**. The following links also are available to get more information on Atmel wireless drivers, Linux kernel and prebuilt images.

- Atmel Linux For SAM Site: http://www.at91.com/linux4sam
- GitHub Linux For SAM Site: http://github.com/linux4sam
- GitHub Linux For ATWILC3000 Site: https://github.com/atwilc3000

Table of Contents

1. ATWILC3000 Linux Software Package

The latest ATWILC3000 device driver is available on the WILC3000 GitHub. Download the WILC3000 driver.

```
git clone https://github.com/atwilc3000/driver.git
```

For more information, visit the GitHub wiki pages: https://github.com/atwilc3000/driver/wiki

2. Build, install and load the driver

The wilc3000 linux driver is tested on linux kernel 3.10, NL80211 and WPA supplicant for the Wi-Fi. The Bluetooth is also verified on Bluez 4.98 or higher. To build in-tree module, see the following instructions:

- Download the latest driver and firmware.

  Refer to https://github.com/atwilc3000/driver/releases

  Refer to https://github.com/atwilc3000/firmware/releases

- Create the **atmel** directory to the target kernel tree (/drivers/net/wireless/) if not exist.

  $ cd /drivers/net/wireless

  $ make atmel

- Copy the ATWILC3000 driver downloaded in the atmel directory.

  $cp –r driver/ drivers/net/wireless/atmel/

- Add the following in the **Kconfig** located in the **/drivers/net/wireless/Kconfig .**

  source "drivers/net/wireless/atmel/Kconfig"

- Add the following in the **Makefile** located in the **/drivers/net/wireless/Makefile .**

  Obj-$(CONFIG_ATMEL_SMARTCONNECT) += atmel/

- Configure your kernel for instance with issuing the following command.

  $ make menuconfig

- Then, build your kernel.

This should produce the wilc3000.ko. Copy the wilc3000.ko to the target platform and load the wilc3000 driver by issuing *insmod wilc3000.ko* on the target platform. The wilc3000 driver will output network interface **wlan0** which should be listed by issuing *ifconfig wlan0*. The last step is to bring up the interface newly created by *ifconfig wlan0 up*. The Bluetooth firmware as well as Wi-Fi is loaded and started by issuing *ifconfig wlan0 up*. So, it should be first issued before attaching the Bluetooth device in your target platform.

For practical information, refer to the following porting guides in the release package.

- **WILC3000-SAMA5D3-Xplained-Quick-Start-Guide.pdf**
- **WILC3000-SAMA5D4EK-Quick-Start-Guide.pdf**
- **WILC3000-SAM9N12EK-Quick-Start-Guide.pdf**

3. Wireless tools

This chapter describes how to use Wi-Fi tools to bring up the Atmel network drivers on the Linux platform. Find more information on the Wi-Fi tools in the GitHub wiki page.

3.1 Enable the network interface

This will enable the wilc3000 device and WiFi/BT firmware will be downloaded during bring-up. Make sure the wilc3000 module already is loaded in the kernel by issuing *insmod wilc3000.ko*.

```
$ ifconfig wlan0 up
```

Use *ifconfig* to see the interface information.

```
$ ifconfig wlan0

wlan0     Link encap:Ethernet  HWaddr 00:80:C2:5E:A2:FF
          inet addr:192.168.0.125  Bcast:192.168.0.255  Mask:255.255.255.0
          inet6 addr: fe80::280:c2ff:fe5e:a2ff/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:1734 errors:0 dropped:0 overruns:0 frame:0
          TX packets:21 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:164773 (160.9 KiB)  TX bytes:2742 (2.6 KiB)
```

3.2 Scan for networks

Scan for neighboring networks. A list of networks with information on e.g. ssid, WPS and signal levels will be shown when the scan is completed. The wireless tool, **iw** can be used to scan networks in neighbor. The wireless tool, **iw** can scan networks, list wireless interfaces, show connection status, connect the network and also dump the statistic information. However, it is not available to connect the WPA/WPA2 secured networks. The WPA supplicant will be replaced instead of **iw** command later in this chapter to show simple demo for the WPA/WPA2 networks. The following shows how to scan networks with **iw** command.

```
$ iw wlan0 scan
```

3.3 Connect to an unprotected network

Connect to an open network that was found during the scan. Assume that the SSID of open AP is DEMO_AP..

```
$  iw wlan0 connect DEMO_AP
```

We can check the link status by invoking the following command.

```
$  iw wlan0 link

Connected to 00:26:66:23:05:a4 (on wlan0)
          SSID: DEMO_AP
          freq: 2452
          signal: -61 dBm
          tx bitrate: 36.0 MBit/s
```

3.4 Assign an IP address

If a static IP address is used, set it using *ifconfig*

```
$ ifconfig wlan0 192.168.0.127
```

If DHCP is used, invoke the DHCP client available on the platform:

```
$ udhcpc -i wlan0

udhcpc (v1.21.1) started
Sending discover...
Sending select for 192.168.0.125...
Lease of 192.168.0.125 obtained, lease time 864000
deleting routers
route: SIOCDELRT: No such process
adding dns 203.248.252.2
adding dns 164.124.101.2
```

Check the interface by issuing *ifconfig wlan0* to make sure the correct IP was allocated.

```
$ ifconfig wlan0

wlan0     Link encap:Ethernet  HWaddr 00:80:C2:5E:A2:FF
          inet addr:192.168.0.125  Bcast:192.168.0.255  Mask:255.255.255.0
          inet6 addr: fe80::280:c2ff:fe5e:a2ff/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:1734 errors:0 dropped:0 overruns:0 frame:0
          TX packets:21 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:164773 (160.9 KiB)  TX bytes:2742 (2.6 KiB)
```

3.5 Verify the connection

Ping the access point to verify the connection by issuing *ping* command.

```
$ ping atmel.com
PING atmel.com (192.199.1.11): 56 data bytes
64 bytes from 192.199.1.11: seq=0 ttl=51 time=268.997 ms
64 bytes from 192.199.1.11: seq=1 ttl=51 time=260.619 ms
64 bytes from 192.199.1.11: seq=2 ttl=51 time=210.690 ms

--- atmel.com ping statistics ---
3 packets transmitted, 3 packets received, 0% packet loss
round-trip min/avg/max = 210.690/246.768/268.997 ms
```

3.6 Disconnect from the network

```
$  iw wlan0 disconnect
```

Check the link status with the following command.

```
$  iw wlan0 link

Not connected.
```

3.7 Connect to an WEP encrypted network

This chapter shows how to connect to an access point encrypted with WEP 64-bit. Assume that key index 1 set to *1122334455*. Key index 1 should be the default transmit key and no other keys should be configured on the access point.

```
$ iw wlan0 connect DEMO_AP key 0:1122334455

wlan0 (phy #0): connected to 00:26:66:23:05:a4
```

Check the link status with the following command.

```
$ iw wlan0 link

Connected to 00:26:66:23:05:a4 (on wlan0)
        SSID: DEMO_AP
        freq: 2452
        signal: -58 dBm
        tx bitrate: 12.0 MBit/s
```

3.8 Connect to an WEP encrypted network with WPA supplicant

For more information on the WPA supplicant, refer to here which explains more details for WPA supplicant with the CLI (Command Line Interface) client, wpa_cli. Make sure the WPA supplicant daemon runs in background before using wpa_supplicant.

```
$ wpa_supplicant –b –Dnl80211 –iwlan0 –c /etc/wpa_supplicant.conf
```

The configuration file is usually located in the **/etc/wpa_supplicant.conf**. The following is simple WPA supplicant configuration.

```
ctrl_interface=/var/run/wpa_supplicant
update_config=1
```

The following shows how to connect the access point secured with open WEP 64 bit security. Assume the SSID is con_system with WEP key, 1122334455.

```
$ wpa_cli –iwlan0 add_network
$ wpa_cli -iwlan0 set_network 0 mode 0
$ wpa_cli -iwlan0 set_network 0 auth_alg OPEN
$ wpa_cli -iwlan0 set_network 0 key_mgmt NONE
$ wpa_cli -iwlan0 set_network 0 wep_key0 1122334455
$ wpa_cli -iwlan0 set_network 0 ssid '"con_system"'
$ wpa_cli -iwlan0 enable_network 0
```

Check the link status by issuing the following command.

```
$ wpa_cli 0iwlan0 status
```

3.9 Connect to WPA network

Issue the following commands to connect secured network with WPA. Assume that the SSID is con_system and pre-shared key is 12345678.

```
$ wpa_cli -iwlan0 add_network
$ wpa_cli -iwlan0 set_network 0 mode 0
$ wpa_cli -iwlan0 set_network 0 auth_alg OPEN
$ wpa_cli -iwlan0 set_network 0 key_mgmt WPA-PSK
$ wpa_cli -iwlan0 set_network 0 psk '"12345678"'
$ wpa_cli -iwlan0 set_network 0 ssid '"con_system"'
$ wpa_cli -iwlan0 enable_network 0
```

3.10 Connect to WPA2 network

The following is sample commands to connect to WPA2 network. Assume that the SSID is con_system and the key is 1345678.

```
$ wpa_cli -iwlan0 add_network
$ wpa_cli -iwlan0 set_network 0 mode 0
$ wpa_cli -iwlan0 set_network 0 auth_alg OPEN
$ wpa_cli -iwlan0 set_network 0 key_mgmt WPA-PSK
$ wpa_cli -iwlan0 set_network 0 proto RSN
$ wpa_cli -iwlan0 set_network 0 psk '"12345678"'
$ wpa_cli -iwlan0 set_network 0 ssid '"con_system"'
$ wpa_cli -iwlan0 enable_network 0
```

3.11 AP mode

The access point can be supported with the hostapd which is a user space daemon. The hostapd configuration file, for example, **/etc/hostapd.conf**, has various configurable options to define the soft AP mode operation. This document describes only how to configure the AP mode with WPA/WPA2 encryption. For more information, visit the GitHub wiki page.

Write the **hostapd.conf** like the following.

```
interface=wlan0
driver=nl80211
ctrl_interface=/var/run/hostapd
ssid=SoftAP
dtim_period=2
beacon_int=100
channel=6
auth_algs=1
hw_mode=g
max_num_sta=8
ap_max_inactivity=300
wpa=3
wpa_passphrase=12341234
wpa_key_mgmt=WPA-PSK
wpa_pairwise=CCMP TKIP
```

Configure the DHCP server by writing **/etc/dhcod.conf**. Refer to the following sample file.

```
ddns-update-style none;
default-lease-time 600;
max-lease-time 7200;

option subnet-mask 255.255.255.0;
option domain-name-servers 168.126.63.1, 164.124.101.2; # DNS Server IP
option domain-name "sample_domain";                    # domain name

subnet 192.168.0.0 netmask 255.255.255.0 {
  range 192.168.0.100 192.168.0.110;                  # range ip
  option broadcast-address 192.168.0.255;
  option routers 192.168.0.1;                         # gateway ip
}
Log-facility local7;
```

Then, update the /etc/init.d/S80dhcp-server** like the following.

```
INTERFACES=”wlan0”
. . .
test -f /etc/dhcpd.conf || exit 0
```

Run the hostpad and DHCP.

```
$ hostapd /etc/hostapd.conf -B
$ ifconfig wlan0 192.168.0.1
$ /etc/init.d/S80dhcp-server start
```

3.12 P2P mode

WPA supplicant also supports the WiFi direct mode and also configuration file is needed as **/etc/p2p_supplicant.conf**. The following is simple configuration for the simple demonstration.

```
ctrl_interface=/var/run/wpa_supplicant
device_name=wilc3000
device_type=1-0050F204-1
update_config=1
config_methods=virtual_push_button physical_display keypad
```

Run the WPA supplicant in background.

```
$ wpa_supplicant -Dnl80211 -ip2p0 -c/etc/p2p_supplicant.conf -N -Dnl80211 \
-c/etc/wpa_supplicant.conf -i wlan0 &
```

Configure the IP for p2p0 interface.

```
$ ifconfig p2p0 192.168.0.1 netmask 255.255.255.0
```

The p2p commands are founded in wpa_cli interactive mode by issuing the following.

```
$ wpa_cli –ip2p0
```

Then, the following commands are used to run P2P mode.

```
p2p_find
p2p_stop_find
p2p_peers
p2p_connect peers_MAC_address pbc go_intent=15
```

4. BlueZ tools

Make sure the Bluetooth firmware should be downloaded before attaching the Bluetooth device in your target platform. You can start the BT firmware by the following procedures.

- *insmod wilc3000.ko* on the target platform
- Check the interface of wlan0 by issuing *ifconfig –a*
- Bring up the wlan0 by *ifconfig wlan0 up*

The firmware should be located in **/lib/firmware** on your target platform. Then, it's possible to attach the wilc3000 device to the target platform.

4.1 Attach serial device via UART HCI to BlueZ stack

The wilc3000 Bluetooth driver provides the UART interface right now so the following command should be issued to attach the device. Make sure the **/dev/ttyUSBx** exists on the target platform. The BT firmware has the baud rate, 921600 and no flow control.

```
# hciattach ttyUSB0 any 921600 noflow
Bluetooth: HCI UART driver ver 2.2
Bluetooth: HCI H4 protocol initialized
Bluetooth: HCI BCSP protocol initialized
Bluetooth: HCILL protocol initialized
Bluetooth: HCIATH3K protocol initialized
Bluetooth: HCI Three-wire UART (H5) protocol initialized
Device setup complete
```

Make sure the HCI interface is created.

```
# hciconfig -a
hci0:   Type: BR/EDR  Bus: UART
        BD Address: AB:89:67:45:23:01  ACL MTU: 1021:9  SCO MTU: 255:4
        DOWN
        RX bytes:574 acl:0 sco:0 events:27 errors:0
        TX bytes:411 acl:0 sco:0 commands:27 errors:0
        Features: 0xff 0xff 0xcd 0xfe 0xdb 0xff 0x7b 0x87
        Packet type: DM1 DM3 DM5 DH1 DH3 DH5 HV1 HV2 HV3
        Link policy: RSWITCH HOLD SNIFF PARK
        Link mode: SLAVE ACCEPT
```

4.2 Enable the network interface

Bring up the wilc3000 Bluetooth HCI interface.

```
# hciconfig hci0 up
# hciconfig -a
hci0:   Type: BR/EDR  Bus: UART
        BD Address: AB:89:67:45:23:01  ACL MTU: 1021:9  SCO MTU: 255:4
        UP RUNNING
        RX bytes:1141 acl:0 sco:0 events:53 errors:0
        TX bytes:818 acl:0 sco:0 commands:53 errors:0
        Features: 0xff 0xff 0xcd 0xfe 0xdb 0xff 0x7b 0x87
        Packet type: DM1 DM3 DM5 DH1 DH3 DH5 HV1 HV2 HV3
        Link policy: RSWITCH HOLD SNIFF PARK
        Link mode: SLAVE ACCEPT
        Name: 'NMI NMC3000'
        Class: 0x000000
        Service Classes: Unspecified
        Device Class: Miscellaneous,
        HCI Version: 4.0 (0x6)  Revision: 0x709
        LMP Version: 4.0 (0x6)  Subversion: 0x709
        Manufacturer: RivieraWaves S.A.S (96)
```

4.3 Scan for networks

Scan for neighboring networks. A list of networks with information on BD_ADDR and name will be shown when the scan is completed.

```
# hcitool scan
Scanning ...
        60:6C:66:A4:29:63   D247-PC
        60:03:08:89:93:E7   damiank-mbp1
        94:63:D1:06:52:B5   Jude_android
        48:D2:24:63:5B:C3   n/a.
        E0:06:E6:BE:A8:FA   APDN194
        D8:57:EF:C7:20:4D   SHV-E210K
        78:DD:08:B2:91:C9   ALEX-PC
```

4.4 Connect to network

You can use DBUS interface so that you connect to a network that was found during the scan. In this case, you have to choose the MAC address. Make sure Bluetooth daemon should first run.

```
# bluetoothd &

# export BT_ADAPTER=`dbus-send --system --dest=org.bluez --print-reply /
org.bluez.Manager.DefaultAdapter | tail -n 1 | sed 's/^.*"\(.*\)".*$/\1/'`

# dbus-send --system --print-reply --dest=org.bluez $BT_ADAPTER
org.bluez.Adapter.CreatePairedDevice string: D8:57:EF:C7:20:4D
objpath:/org/bluez/agent string:NoInputNoOutput
```

You can list up the paired device.

```
# dbus-send --system --print-reply --dest=org.bluez $BT_ADAPTER
org.bluez.Adapter.ListDevices
method return sender=:1.2 -> dest=:1.11 reply_serial=2
  array [
    object path "/org/bluez/1348/hci0/dev_D8_57_EF_C7_20_4D"
  ]
```

You can remove the paired device by issuing the following command:

```
# dbus-send --system --print-reply --dest=org.bluez  $BT_ADAPTER
org.bluez.Adapter.RemoveDevice objpath:$BT_ADAPTER/dev_D8_57_EF_C7_20_4D
```

5. Conclusion

This application note described how to integrate the Atmel WILC3000 Combo driver in the Linux kernel in general and also showed how to use wireless and Bluetooth tools for the combo WILC3000 Soc.

6. Revision history

Doc. Rev.	Date	Comments
XXXXXA	11/2014	Initial document release
	11/2014	Add wilc3000 bluetooth
	2/2015	Update the contents

Atmel | Enabling Unlimited Possibilities®

Atmel Corporation
1600 Technology Drive
San Jose, CA 95110
USA
Tel:(+1)(408) 441-0311
Fax: (+1)(408) 487-2600
www.atmel.com

Atmel Asia Limited
Unit 01-5 & 16, 19F
BEA Tower, Millennium City 5
418 Kwun Tong Road
Kwun Tong, Kowloon
HONG KONG
Tel:(+852) 2245-6100
Fax: (+852) 2722-1369

Atmel Munich GmbH
Business Campus
Parkring 4
D-85748 Garching b. Munich
GERMANY
Tel:(+49) 89-31970-0
Fax: (+49) 89-3194621

Atmel Japan G.K.
16F Shin-Osaki Kangyo Bldg.
1-6-4 Osaki, Shinagawa-ku
Tokyo 141-0032
JAPAN
Tel: (+81)(3) 6417-0300
Fax: (+81)(3) 6417-0370