StarFive
赛昉科技

JH7110 WiFi and Bluetooth Developing and Porting Guide

VisionFive 2

Version: 1.0

Data: 2023/07/27

Doc ID: JH7110-DGEN-018

Legal Statements

Important legal notice before reading this documentation.

PROPRIETARY NOTICE

Information in this document is provided "as is," with all faults. Contents may be periodically updated or revised due to product development. Shanghai StarFive Technology Co., Ltd. (hereinafter "StarFive") reserves the right to make changes without further notice to any products herein.

StarFive expressly disclaims all warranties, representations, and conditions of any kind, whether express or implied, including, but not limited to, the implied warranties or conditions of merchantability, fitness for a particular purpose, and non-infringement.

StarFive does not assume any liability arising out of the application or use of any product or circuit, and specifically disclaims any and all liability, including without limitation indirect, incidental, special, exemplary, or consequential damages.

Contact Us

Address: Room 502, Building 2, No. 61 Shengxia Rd., China (Shanghai) Pilot Free Trade Zone, Shanghai, 201203, China

Website: http://www.starfivetech.com

Email:

- Sales: sales@starfivetech.com
- Support: support@starfivetech.com

Contents

List of Tables

List of Figures

Preface

About this guide and technical support information.

About this document

This document mainly provides the SDK developers with the programing basics and debugging know-how for the WiFi and Bluetooth of the StarFive next generation SoC platform - JH7110.

Audience

This document mainly serves the WiFi and Bluetooth relevant driver developers. If you are developing other modules, place a request to your sales or support consultant for our complete documentation set on JH7110.

Revision History

Table 0-1 Revision History

Version	Released	Revision
1.0	2023/07/27	The First Official Release.

Notes and notices

The following notes and notices might appear in this guide:

- **Tip:**
  Suggests how to apply the information in a topic or step.
- **Note:**
  Explains a special case or expands on an important point.
- **Important:**
  Points out critical information concerning a topic or step.
- **CAUTION:**
  Indicates that an action or step can cause loss of data, security problems, or performance issues.
- **Warning:**
  Indicates that an action or step can result in physical harm or cause damage to hardware.

1. WiFi Development Instructions

1.1. Introduction

WiFi is a wireless local area network technology. The working principle of WiFi technology is data transmission through wireless signals. Devices are connected to wireless networks through WiFi chips, and then connected to the internet through routers. When the device needs to transmit data, it will send the data to the router, which then sends the data to the target device to achieve data transmission.

This chapter displays the following two parts:

1.1.1. WiFi Work Mode

On the JH7110 DevKits, the WiFi module generally has the following two work modes: Station mode and AP mode.

- Station mode: The endpoint connected to the wireless network. This is the most common work mode, most network adapters work in this mode.
- AP mode: The wireless access point, also called hotspot. In this mode, your device works as a wireless router.

1.1.2. WiFi Software Architecture

1.1.2.1. WiFi Dongle Basic Concept

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

The wireless driver runs on the WLAN 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:

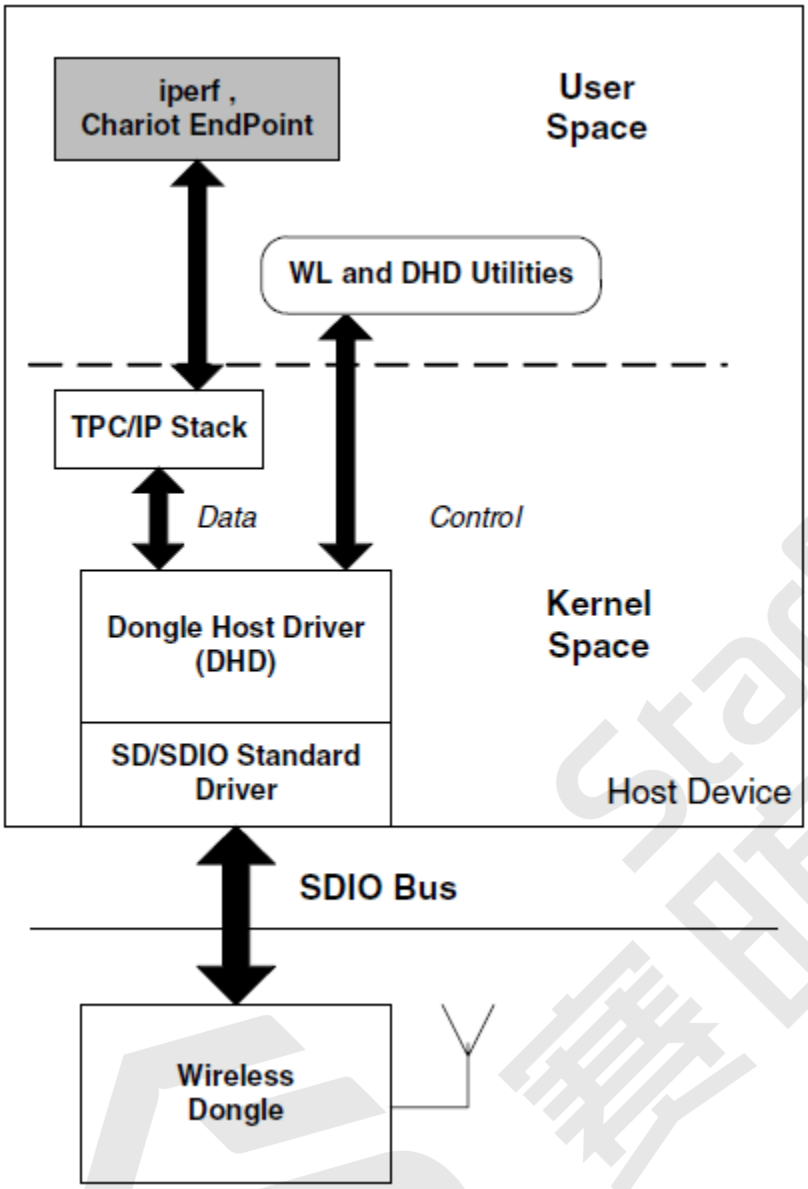
- Power savings
- A reduction in driver size and complexity
- Processor offloading for activities such as checksum calculation and Address Resolution Protocol (ARP) execution

1.1.2.2. WiFi Dongle Overview

The Dongle Host Driver (DHD) is the executable module that provides encapsulated communication between the host device and the Ampak module over the SDIO bus. The dongle software architecture is based on DHD, which is a host-based driver used to provide a communication channel with the dongle device firmware.

The following is the digram of the Broadcom SDIO WLAN Dongle concept.

Figure 1-1 Broadcom SDIO WLAN Dongle Concept



Tip:

Chaiot EndPoint is one of the top network business and performance testing software, capable of simulating numerous commercial applications for testing.

1.1.2.3. WiFi Software Package

The WiFi software package of JH7110 contains following files:

- Dongle host driver: WLAN adapter host driver.
- Dongle device firmware: WLAN adapter device firmware.

1.2. Porting Instruction

This chapter describes the porting instruction in the following parts:

1.2.1. WiFi Driver Construction

Follow the steps below to port the WiFi driver:

1. Place the `aic` driver package under `/linux/driver/net/wireless` directory.
2. Add the following content to `/linux/driver/net/wireless/Makefile`:

```
obj-$(CONFIG_AIC_WLAN_SUPPORT) += aic8800/
```

3. Add the following content to `/linux/driver/net/wireless/Kconfig`:

```
source "drivers/net/wireless/aic8800/Kconfig"
```

4. Place the WiFi firmware and configuration file to the specified directory: Place the WiFi firmware package (`aic8800`) to `/lib/firmware` path.

1.2.2. Kernel Menu Configuration

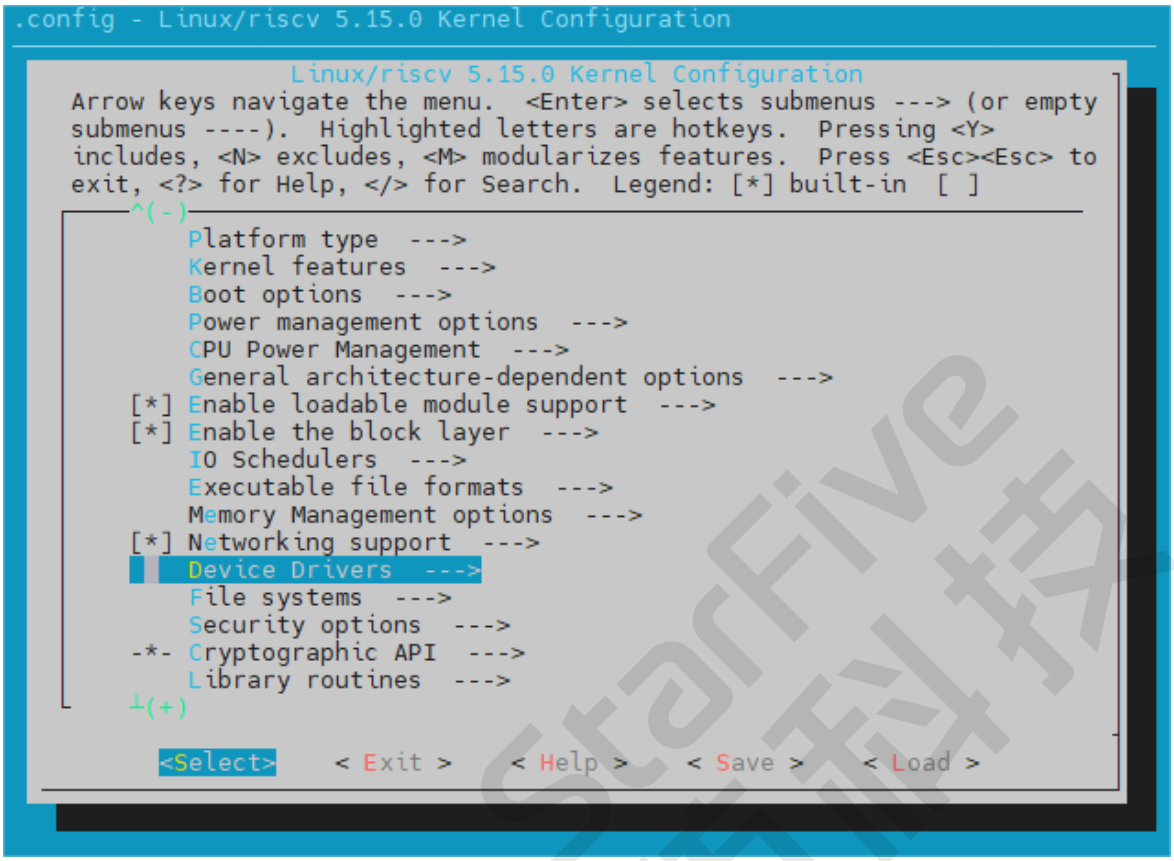
Follow the steps below to enter the kernel menu to enable the kernel configuration for WiFi.

1. Under the root directory of VisionFive 2, type the following command to enter the kernel menu configuration GUI.

```
make linux-menuconfig
```

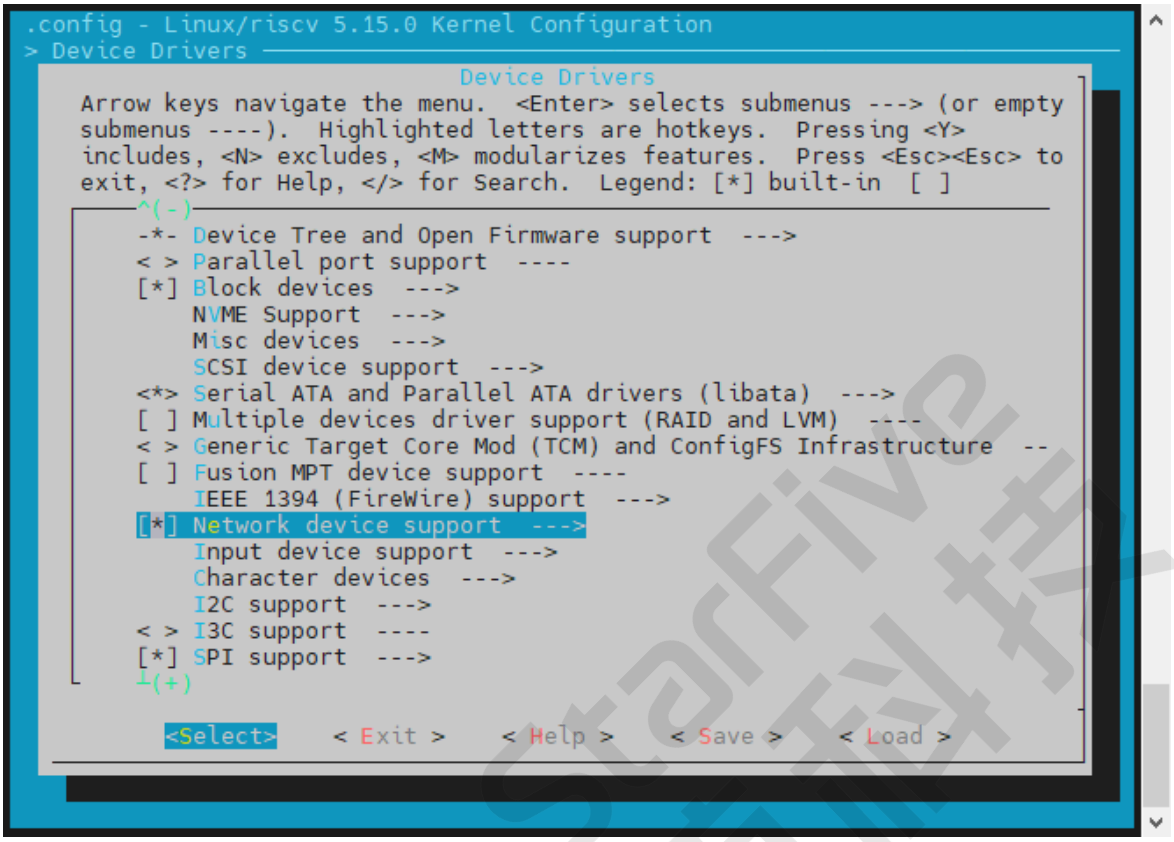
2. Enter the **Device Drivers** menu option.

Figure 1-2 Device Drivers



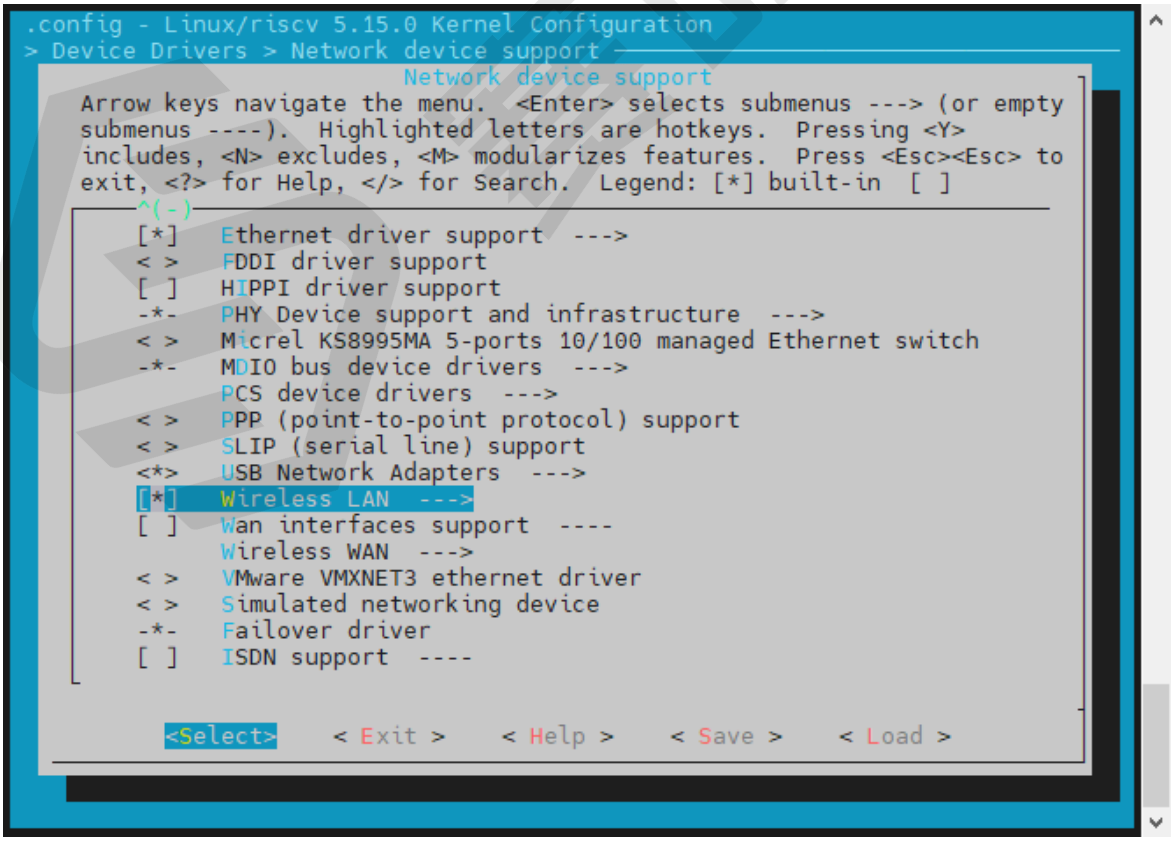
3. Enter the **Network device support** menu option.

Figure 1-3 Network device support



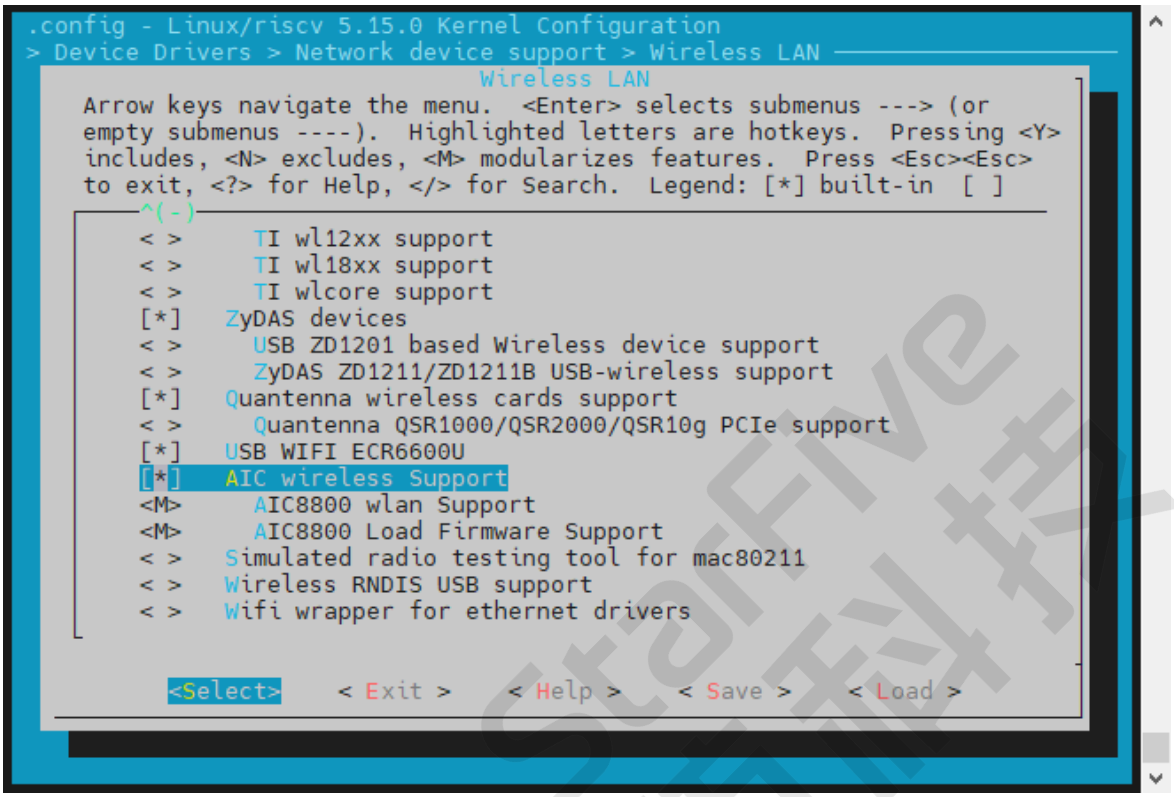
4. Enter **Wireless LAN** menu option.

Figure 1-4 Wireless LAN



5. Choose **AIC wireless Support** option.

Figure 1-5 AIC wireless Support



6. Compile **AIC8800 wlan Support** and **AIC8800 Load Firmware Support** into a module.

Figure 1-6 AIC8800 Load Firmware Support

7. Save your change before you exit the kernel configuration dialog.

1.2.3. Install Dongle Host Driver

The following is the method to install WiFi Dongle host driver:

- To compile the Dongle host driver into kernel, execute the following command:

```
ifconfig wlan0 up
```

1.3. WiFi Operating Instructions

This chapter describes the WiFi operation instruction in the following two parts:

- Station Mode Operation *(on page 14)*
- SoftAP Mode Operation *(on page 18)*

1.3.1. Station Mode Operation

1.3.1.1. Add WPA Supplicant Configuration File

Follow the steps below to create the `wpa_supplicant.conf` file:

1. Manually creating the `wpa_supplicant.conf` configuration file or use `/etc/wpa_supplicant.conf`.
2. Add the following content to the configuration file:

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

Note:
This allows `wpa_supplicant` to overwrite configuration files after modifying the configuration. For example, add new network statement blocks through `wpa_cli` tool, write configuration to `wpa_gui`, change password, etc.

Network Settings

There are 3 examples to add network for `wpa_supplicant.conf` file:

- Open system without encryption:

```
network={
ssid="tttb"
key_mgmt=NONE
}
```

- `Open/Shared` authentication with WEP encryption:

```
network={
ssid="tttb"
key_mgmt=NONE
auth_alg=OPEN SHARED
wep_key0=1234567890
}
```

- WPA/WPA2-PSK authentication with TKIP/AES encryption:

```
network={
ssid="tttb"
psk="12345678"
}
```

You can also use `wpa_passphrase` tool to add encrypted network automatically for your `wpa_supplicant.conf` file:

```
wpa_passphrase ssid >> wpa_supplicant.conf
password
```

1.3.1.2. Install WPA Supplicant

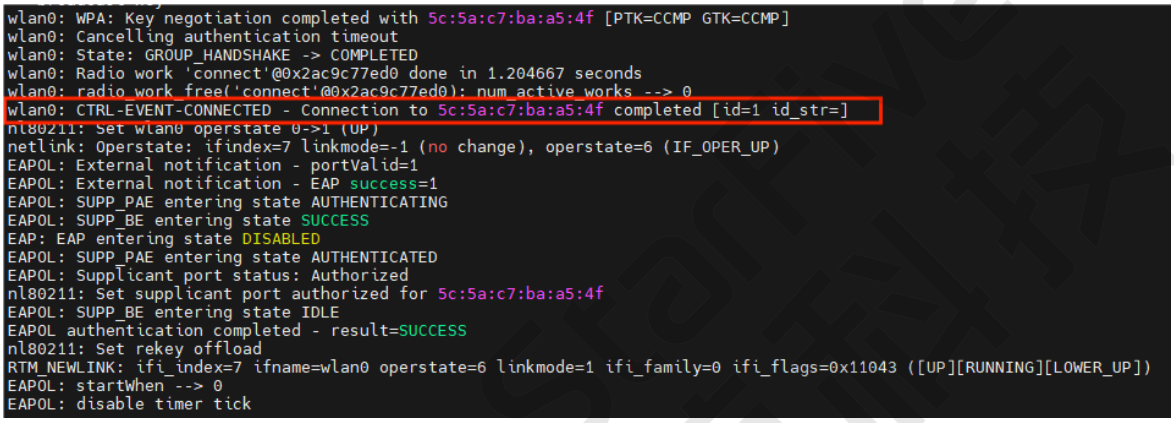
Follow the steps below to install WPA supplicant.

1. Execute the following command to connect to the network configured in `wpa_supplicant.conf`:

```
wpa_supplicant -Dnl80211 -i wlan0 -c wpa_supplicant.conf -d&
```

Example Output

Figure 1-7 Example Output



2. After connect successfully, execute the following command to obtain the IP of **wlan0** by using the `udhcpc` tool:

```
udhcpc -i wlan0
```

Example Output

Figure 1-8 Wlan0 IP

```
# udhcpc -i wlan0
udhcpc: started, v1.34.1
udhcpc: broadcasting discover
udhcpc: broadcasting select for 192.168.125.101, server 192.168.110.101
udhcpc: lease of 192.168.125.101 obtained from 192.168.110.101, lease time 34200
deleting routers
adding dns 192.168.110.101
adding dns 202.207.240.225
adding dns 8.8.8.8
# ifconfig
eth0      Link encap:Ethernet  HWaddr 66:34:B0:6C:08:AD
          inet6 addr: fe80::6434:b0ff:fe6c:8ad/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:126 errors:0 dropped:6 overruns:0 frame:0
          TX packets:18 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:9758 (9.5 KiB)  TX bytes:2172 (2.1 KiB)
          Interrupt:38

eth1      Link encap:Ethernet  HWaddr 66:34:B0:7C:08:5D
          UP BROADCAST MULTICAST  MTU:1500  Metric:1
          RX packets:0 errors:0 dropped:0 overruns:0 frame:0
          TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:0 (0.0 B)  TX bytes:0 (0.0 B)
          Interrupt:41

lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          inet6 addr: ::1/128 Scope:Host
          UP LOOPBACK RUNNING  MTU:65536  Metric:1
          RX packets:2 errors:0 dropped:0 overruns:0 frame:0
          TX packets:2 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:140 (140.0 B)  TX bytes:140 (140.0 B)

wlan0     Link encap:Ethernet  HWaddr B8:13:32:98:27:F4
          inet addr:192.168.125.101  Bcast:192.168.125.255  Mask:255.255.255.0
          inet6 addr: fe80::ba13:32ff:fe98:27f4/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:644 errors:0 dropped:80 overruns:0 frame:0
          TX packets:34 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:69644 (68.0 KiB)  TX bytes:3720 (3.6 KiB)
```

1.3.1.3. `wpa_cli` Tool

`wpa_cli` is a text based front-end program used to communicate with `wpa_supplicant`. It is used to check the current state, change configuration, trigger events, and request interactive user input.

In addition, it can be used to configure variables such as EAPOL state machine parameters and trigger events such as re-association and IEEE 802.1X logout/login.

`wpa_cli` has many functions, which can be seen through execute `wpa_cli -h` command. The following list are some example commands:

- Check the WLAN status:

```
wpa_cli -p/var/run/wpa_supplicant -iwlan0 status
```

- Scan the nearby WLAN:

```
wpa_cli -i wlan0 scan
```

- Scan the nearby WLAN and list the devices:

```
wpa_cli -i wlan0 scan_result
```

- View all accessible networks and the currently connected network:

```
wpa_cli -i wlan0 list_networks
```

- Obtain an ID that stores the WLAN structure, assuming 1:

```
wpa_cli -i wlan0 add_network
```

- Set the hotspot SSID with ID 1:

```
wpa_cli -i wlan0 set_network 1 ssid '"HO4428"'
```

- Set the password for the hotspot with ID 1:

```
wpa_cli -i wlan0 set_network 1 psk '"442701102"'
```

1.3.2. SoftAP Mode Operation

1.3.2.1. Add Hostapd Configuration File

Follow the steps to create the `hotsapd.conf` file:

1. Create the `hostapd.conf` configuration file manually or use `/etc/hostapd.conf`.
2. Add the following content to the configuration file.

AP Hotpot Settings

The following are 2 examples to add AP hotpot for your `hostapd.conf` file:

- Open system without encryption:

```
interface=wlan0
driver=nl80211
ctrl_interface=/var/run/hostapd
ssid=AndroidAP
channel=6
```

```
ieee80211n=1
hw_mode=g
ignore_broadcast_ssid=0
```

- WPA2-PSK authentication with AES encryption:

```
interface=wlan0
driver=nl80211
ctrl_interface=/var/run/hostapd
ssid=AndroidAP
channel=6
ieee80211n=1
hw_mode=g
ignore_broadcast_ssid=0
wpa=2
wpa_key_mgmt=WPA-PSK
wpa_passphrase=12345678
```

1.3.2.2. Install Hostapd

Follow the steps below to install hostapd.

1. Execute the following command to create AP hotpot configured in `hostapd.conf`:

```
hostapd hostapd.conf -B
```

Example Output

Figure 1-9 Example Output

```
# hostapd hostapd.conf -B
Configuration file: hostapd.conf
HT (IEEE 802.11n) with WPA/WPA2 requires CCMP/GCMP to be enabled, disabling HT capabilities
rfkill: Cannot open RFKILL control device
[  185.743102] [dhd] [wlan0] wl_cfg80211_del_station : Disconnect STA : ff:ff:ff:ff:ff:ff scb_val.val 3
Using interface wlan0 with hwaddr b8:13:32:98:27:f4 and ssid "AndroidAP"
[  185.812673] [dhd] [wlan0] wl_cfg80211_set_channel : netdev_ifidx(6) chan_type(1) target channel(2g-5 20MHz)
[  185.828841] [dhd] [wlan0] wl_cfg80211_bcn_bringup_ap : Creating AP with sec=wpa2/psk/mfpn/tkip
[  185.931729] [dhd] [wlan0] wl_ext_iapsta_link : [A] Link up w/o creating? (etype=16)
[  185.939469] [dhd] [wlan0] wl_ext_iapsta_link : [A] Link up w/o creating? (etype=16)
[  185.949070] [dhd] [wlan0] wl_notify_connect_status_ap : AP/GO Link up (2g-5 20MHz)
wlan0: interface state UNINITIALIZED->ENABLED
wlan0: AP-ENABLED
```

You can use your phone or computer to locate the hotspot, but you cannot connect to it. To solve this issue, you need to assign an IP address to this hotspot.

2. Add subnet to the main configuration file `/etc/dhcp/dhcpconf of dhcp`, for example:

```
subnet 192.168.0.0 netmask 255.255.255.0 {
range 192.168.0.10 192.168.0.20;
option routers 192.168.0.1;
}
```

Then, execute the following command to configure the IP address for the WiFi interface:

```
ifconfig wlan0 192.168.0.1
```

3. Execute the following command to troubleshoot the DHCP server:

```
dhcpd
```

   Now, you can connect to the hotspot through devices such as mobile phone or PC and ping it for test.

4. (Optional) If you want to access the internet through this hotspot, you need to set up IP forwarding. IP forwarding means that forwards the interface connecting the network cable on SBC to the wlan0 interface. The following is an example:

```
echo 1 > /proc/sys/net/ipv4/ip_forward
iptables -F
iptables -P FORWARD ACCEPT
iptables -t nat -A POSTROUTING -o ethX -j MASQUERADE
```

2. Bluetooth Development Instructions

2.1. Introduction

Bluetooth is a wireless technology standard whose purpose is to connect gadgets without a cable. The Bluetooth module is a tiny part of the chip in a device, which lets it wirelessly communicate with a Bluetooth module on any other devices. Generally speaking, Bluetooth is used to transfer small amounts of data while being efficient with battery usage. Among the various wireless standards (like WiFi), Bluetooth is known for maintaining a stable connection in short distances, and transferring small amounts of data without taking too much power.

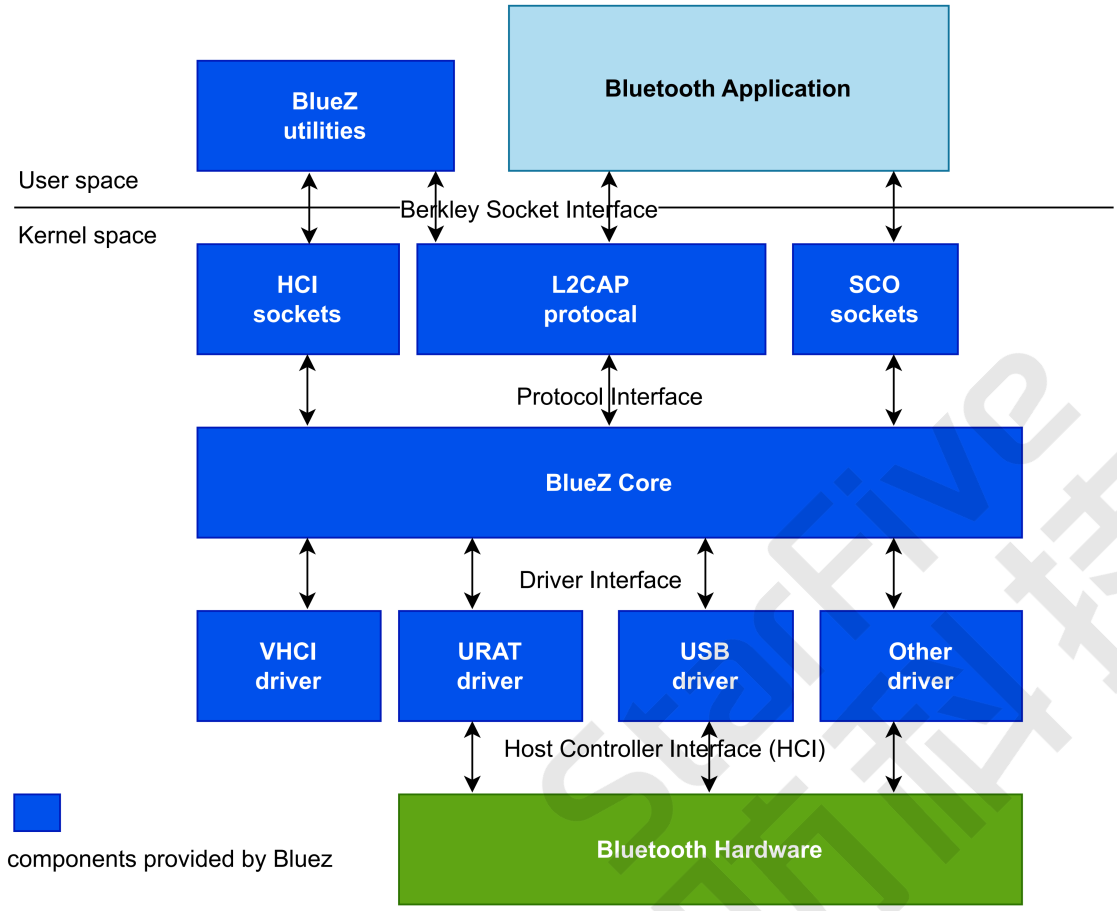
This chapter is intended to give Ampak Bluetooth module users a general guide of how to bring up the Bluetooth hci interface and customer can use bluez stack to enable full bluetooth function.

2.1.1. Bluetooth Software Architecture Overview

BlueZ is the official Linux Bluetooth stack as well as Android. It provides support for core Bluetooth layers and protocols. We use it to provide Bluetooth profiles on GB86XX and it consists of following components:

- HCI Core
- HCI UART, USB and Virtual HCI device drivers
- L2CAP protocol module
- Configuration and testing utilities

Figure 2-1 BlueZ Overview Diagram



In our case, we use UART as the **Host Controller Interface** (HCI) and AIC8800 is the **Bluetooth Hardware** in figure above.

2.2. Porting Instruction

2.2.1. Bluetooth Driver Construction

Follow the steps below to port the Bluetooth driver:

1. Place the `aic8800` bluetooth driver package under `/linux/driver/bluetooth` directory.
2. Add the following code under `linux/drivers/bluetooth/Kconfig`:

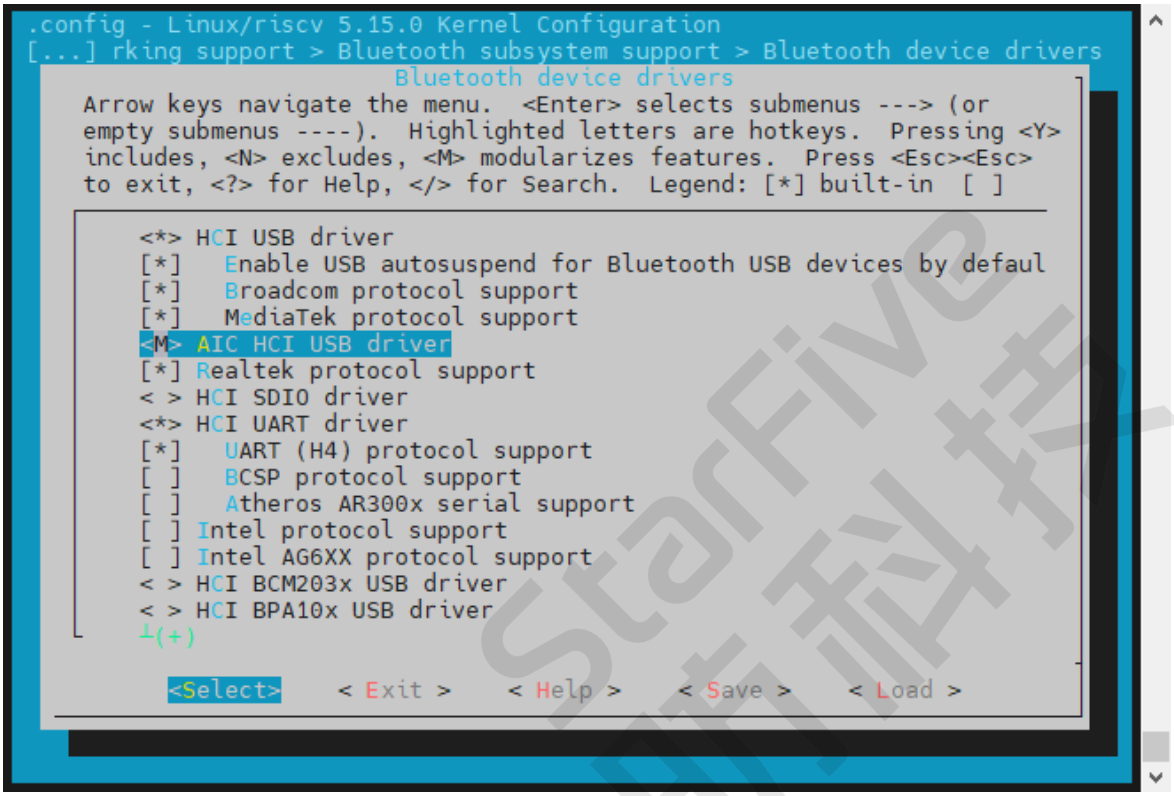
```
config BT_AICBTUSB
tristate "AIC HCI USB driver"
depends on USB
help
AIC Bluetooth HCI USB driver
```

3. Add the following code to `linux/driver/Bluetooth/Makefile`:

```
obj-$(CONFIG_BT_AICBTUSB) += aic_btusb.o
```

4. Compile **AIC HCI USB driver** as a module.

Figure 2-2 AIC HCI USB driver



2.2.2. Enable Bluetooth

Following the steps below to enable the Bluetooth:

1. Perform the steps bellow to initialize Bluetooth:

   a. Perform the following command to bring up **hci** interface.

   ```
   #hciconfig hci0 up
   ```

2. Perform the following command to check Bluetooth device status:

   ```
   # hciconfig
   hci0: Type: BR/EDR Bus: UART
   BD Address: 43:30:B1:00:00:00 ACL MTU: 1021:8 SCO MTU: 64:1
   UP RUNNING
   RX bytes:1011 acl:0 sco:0 events:39 errors:0
   TX bytes:208 acl:0 sco:0 commands:39 errors:0
   ```

3. Perform the following to scan Bluetooth devices:

```
# hcitool scan
Scanning ...
00:22:43:A0:A7:0A n/a
00:10:60:56:56:7B hhhh
00:1A:6B:85:F3:67 n/a
00:22:43:A0:A7:48 AmUrO
00:1F:E1:E1:A1:8F GEMTEK-8AE51F68
```

4. You can also enter Bluetooth interactive interface by using **Bluetoothctl** tool. **Bluetoothctl** has many functions, which can be seen through execute `help` command. The following list are some example commands:

    - Enter the tool:

      ```
      bluetoothctl
      ```

        - Check:

          ```
          default-agent
          ```

        - Register agent:

          ```
          agent on
          ```

        - Scan:

          ```
          scan on
          ```

        - Stop scan:

          ```
          scan off
          ```

        - View the matching devices:

          ```
          devices
          ```

        - Find the pair devices:

          ```
          pair xx:xx:xx:xx:xx:xx
          ```

        - Add trust devices:

          ```
          trust xx:xx:xx:xx:xx:xx
          ```

        - Connect devices:

          ```
          connect xx:xx:xx:xx:xx:xx
          ```