



Quick Guide for Wake on WLAN

Date: 2016/10/25

Version: 0.6

Quick Start Guide for Wake on Lan

(1) Support list:

- .) USB interface: 8188EU, 8188CU, 8192DU, 8192EU, 8723BU, 8812AU, 88x2BU.
- .) SDIO interface: 8189ES, 8188FS, 8723BS, 8703C, 8192ES

(2) Requirements of wakeup via in-band and out-band methods:

.) In-band requirements:

- SDIO Interface:
 - ✓ SDIO host **MUST** support remote wakeup feature.
 - ✓ SDIO data1 **MUST** be wakeup source in the host platform.
 - ✓ The platform **MUST** keep power to WiFi chip in suspend state.
 - ✓ The platform **MUST** work fine between suspend and resume.
- USB Interface:
 - ✓ USB host **MUST** support remote wakeup feature.
 - ✓ The platform **MUST** keep power to WiFi chip in suspend state.
 - ✓ The platform **MUST** work fine between suspend and resume.

.) Out-band requirements:

- ✓ The GPIO of the **PLATFORM** **MUST** be wakeup source.
- ✓ The platform **MUST** keep power to WiFi chip in suspend state.
- ✓ The platform **MUST** work fine between suspend and resume.
- ✓ The WIFI module **MUST** have the GPIO wakeup pin.

(3) Driver Configuration for Wake on Lan:

.) In-band configuration:

If using **SDIO DATA1 pin** or **USB protocol D+/D- toggle** in-band method to wakeup the host, driver need to do is only switch **CONFIG_WOWLAN** from “n” to “y” in Makefile as Figure 1.

```
CONFIG_EXT_CLK = n
CONFIG_WOWLAN = y
CONFIG_GPIO_WAKEUP = n
```

(Figure 1)

.) Out-band configuration:

If using out-band method, driver need to do is modify Makefile and config GPIO. The detail is as following:

- Makefile Configuration:

Switch **CONFIG_WOWLAN** and **CONFIG_GPIO_WAKEUP** from “n” to “y” as Figure 2.

```
CONFIG_EXT_CLK = n
CONFIG_WOWLAN = y
CONFIG_GPIO_WAKEUP = y
```

(Figure 2)

- GPIO Configuration:

- If use the module package, please use the driver default value. The default value depends on HDK document.
- If there is any customized requirement about modify WIFI GPIO number, please modify the value of CONFIG_WAKEUP_GPIO_IDX in Makefile and **please contact with RTK technical support team first**.
- User could use “proc” subsystem to modify the behavior of WIFI GPIO when receive wakeup up packet in non-suspend state.
 - wowlan_gpio_info to show WIFI wakeup host GPIO number and high_active value:
cat /proc/net/rtlxxxx/wlanX/wowlan_gpio_info
 - modify high_active form 0 to 1 in wowlan_gpio_info:
echo 1 > /proc/net/rtlxxxx/wlanX/wowlan_gpio_info
high_active = 0 means pull low wake. (default)
high_active = 1 means pull high wake.

```
isaac@isaac-B33E:~$ cat /proc/net/rtl8723bu/wlan50/wowlan_gpio_info
wakeup_gpio_idx: 14
high_active: 0
isaac@isaac-B33E:~$ echo 1 > /proc/net/rtl8723bu/wlan50/wowlan_gpio_info
isaac@isaac-B33E:~$ cat /proc/net/rtl8723bu/wlan50/wowlan_gpio_info
wakeup_gpio_idx: 14
high_active: 1
```

(Figure 3)

.) CONFIG_DEFAULT_PATTERNS_EN:

If **CONFIG_DEFAULT_PATTERNS_EN** = y, the WIFI driver always provide the following conditons to wake up the host:

1. The TCP unicast packet which MAC address and IP address is matched.
2. The ICMP unicast packet which MAC address and IP address is matched.
3. If you want to add more patterns, please refer “Setup the wake up pattern” session. The new pattern will not overwrie the above default patterns.

If **CONFIG_DEFAULT_PATTERNS_EN** = n, the host will be awake when the MAC address and the IP adrress of destination are matched. User could add other wake up condition. For example the specific multicast type packet. The default value of **CONFIG_DEFAULT_PATTERNS_EN** is disable.

For 88x2B, **must** set **CONFIG_DEFAULT_PATTERNS_EN** = y.

.) Setup the wake up pattern, **ONLY** support on v5.1.0 or later:

- **iwpriv:**

`iwpriv wlanX wow_set_pattern pattern=[pattern]`

Examples:

wake up on any packets sent to MAC 00:E0:4C:01:F0:EE

`iwpriv wlanX wow_set_pattern pattern=00:E0:4C:01:F0:EE`

- **echo pattern into wow_pattern_info:**

`echo [pattern] > /proc/net/rtl8xxx/wlanx/wow_pattern_info`

Examples:

wake up on any packets sent to MAC 00:E0:4C:01:F0:EE

`$ echo 00:E0:4C:01:F0:EE > /proc/net/rtl8xxx/wlanx/wow_pattern_info`

.) clean wake up patterns, **ONLY** support on v5.1.0 or later:

- **iwpriv:**

`iwpriv wlanX wow_set_pattern clean`

Examples:

wake up on any packet sent to MAC 00:E0:4C:01:F0:EE

- **echo clean:**

`$ echo clean > /proc/net/rtl8xxx/wlanx/wow_pattern_info`

- **Patter Format:**

The pattern begins with an 802.3 (Ethernet) header with the correct src/dest MACs base on IPv4. All of the following parameters are need to use **HEX format**. The more information is as following:

AA:AA:AA:AA:AA:AA:BB:BB:BB:BB:BB:BB:CC:CC:DD:--:--:--:--:EE:--:

FF:FF:FF:FF:GG:GG:GG:GG:HH:HH:II:II

A: Ethernet destination address

B: Ethernet source address

C: Ethernet protocol type

D: IP header VER + Hlen, use: 0x45 (4-is for ver. 4, 5 is for len. 20)

E: IP protocol

F: IP source address (192.168.0.1 → C0:A8:00:01)

G: IP destination address (192.168.0.4 → C0:A8:00:04)

H: Source port (1024: 04:00)

I: Destination port (1024: 04:00)

(4) The wake up reason table:

The DUT could be waked up by the WIFI chip with the following reasons:

| Reason Value | Description | Note |
|--------------|---|--|
| 0x01 | Receive pairwise key change packet. | |
| 0x02 | Receive group key change packet. | GTK offload support list: 8723BS/BU, 8192ES/EU, 8812AU |
| 0x04 | Receive disassociate packet. | |
| 0x08 | Receive de-auth. Packet. | |
| 0x10 | AP power off, or could not receive AP's beacon in a period time | |
| 0x21 | Receive magic packet. | |
| 0x22 | Receive unicast packet. | The unicast packet included IP level. |
| 0x23 | Pattern Match | The device could be waked up by specific pattern. |