



---

## **Quick Guide for Wake on WLAN**

Date: 2019/11/12

Version: 0.9

# Quick Start Guide for Wake on Lan

## (1) Support list:

- .) USB interface: 8188EU, 8188CU, 8192DU, 8192EU, 8723BU, 8812AU, 8821AU, 88x2BU, 8188FU, 8723DU, 8814AU, 8821CU, 8192FU, 88x2CU, 8725AU, 8814BU.
- .) SDIO interface: 8189ES, 8189FS, 8723BS, 8703C, 8192ES, 88x2BS, 8821AS, 8703BS, 8723DS, 8723CS, 8821CS, 8192FS, 88x2CS, 8725AS..)
- PCI-E interface: 8812AE, 8821AE, 88x2BE, 8821CE, 8723BE, 8192EE, 8723DE, 8814AE, 8192FE, 88x2CE, 8814BE.

## (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 MSUT 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 MSUT work fine between suspend and resume.
- PCI Interface:
  - ✓ PCI host MUST support remote wakeup feature.
  - ✓ The platform MUST keep power to WiFi chip in suspend state.
  - ✓ The platform MSUT 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 MSUT work fine between suspend and resume.
- ✓ The WIFI module MUST have the GPIO wakeup pin.

## (3) Driver Configuration for Wake on WLAN:

.) 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)

Realtek

.) 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 from 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\_WAKEUP\_TYPE:

If the setting of Makefile is CONFIG\_WAKEUP\_TYPE = 0x7, it means that WOWLAN supports “deauth wake up”, “unicast wake up” and “magic packet wake up”. The detail description is bit0: magic pkt, bit1: unicast and bit2: deauth.

.) Setup the wake up pattern, **ONLY** support on driver version 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 driver version 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.                                |                                                   |
| 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. |

#### (5) wpa\_supplicant Configuration for Wake on WLAN:

The configuration file of wpa\_supplicant should add "wowlan\_triggers=any" when the driver adopt CFG80211 interface in the linux kernel. If there is no "wowlan\_triggers=any", the CFG80211 module will send disconnect command to wifi driver and the wake on WLAN function will fail.

Ex:

```
ctrl_interface=/var/run/wpa_supplicant
#update_config=1
wowlan_triggers=any
#connect to open network
network={
    ssid="SSID"
    psk="12345678"
}
```

The wpa\_supplicant need to restart again after the wifi device/driver remove and insert.

The below command can check the wowlan status of CFG80211:

```
//get the phy number mapping to wlan interface
#iw dev
phy#135
    Interface wlan1
        ifindex 138
        type managed
phy#134
    Interface wlan20
```

ifindex 137  
type managed

//get the wowlan status, below is correct

#iw phy134 wowlan show

WoWLAN is enabled:

\* wake up on special any trigger

//below wowlan status is wrong

iw phy134 wowlan show

WoWLAN is disabled.