Quick Guide for Wake on WLAN

Date: 2017/04/14
Version: 0.6

Realtek

Quick Start Guide for Wake on Lan

(1) Support list:

.) USB interface: 8188EU, 8188CU, 8192DU, 8192EU, 8723BU, 8812AU, .

.) 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 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.

.) Out-band requirements:

- ✓ The GPIOof 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 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 modiy 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 address 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.

Realtek

.) 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)

Realtek

.) **CONFIG_WAKEUP_BY_DATA_FRAME_EN**:

If **CONFIG_WAKEUP_BY_DATA_FRAME_EN** = y, the WIFI driver could wake up the host when it receives any kind of 802.11 data frame; It is highly recommended that **CONFIG_DEFAULT_PATTERNS_EN** is disabled (**CONFIG_DEFAULT_PATTERNS_EN**=n) as **CONFIG_WAKEUP_BY_DATA_FRAME_EN = y.**

.) **CONFIG_UNWAKE_PATTERN_EN**:

If **CONFIG_UNWAKE_PATTERN_EN** = y, the WIFI driver would check the packet triggers the waking event. If it matches any of unwake pattern, the WIFI driver does not wake up the host. Setup and clean the unwake up pattern, **ONLY** support on v5.1.0 or later:

- **echo pattern into wow_unwake_pattern_info:**

  echo [pattern] >

  /proc/net/rtl8xxx/wlanx/wow_unwake_pattern_info

  Examples:

  do not 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 .

- **echo clean:**

  $echo clean >
  /proc/net/rtl8xxx/wlanx/wow_unwake_pattern_info

- **Patter Format:**

  The pattern begins with 802.11 a1/a2/a3 MACs and follows with data payload from the protocol type field within 802.2 (LLC) headers. 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:CC:CC:XC:CC:DD:DD:EE:-:-:-:-:-:-:-:-:-:FF:-:-:GG:GG:GG:GG:HH:HH:HH:HH:II:II:JJ:JJ

  A: 802.11 address 1

  B: 802.11 address 2

  C: 802.11 address 3

  X: wildcard

  D: Ethernet protocol type

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

  F: IP protocol

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

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

I: Source port (1024: 04:00)

J: 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.
0x25	Data frame	The device could be waked up by 802.11 data frame
0x26	Receive matched SSDP packet.	
0x27	Receive matched WSD packet.	
0x28	Receive matched SLP packet.	
0x29	Receive matched LLTD packet.	

0x2A	Receive matched MDNS packet.	
0x50	Receive mismatched SNMP packet.	
0x51	Receive designated mac packet.	

Realtek