AN14465

Wake on Wi-Fi

Rev. 1.0 — 27 May 2025 Application note

Document information

Information	Content
Keywords	Wi-Fi, wake-up, configuration, host, sleep mode, in-band, out-of-band, wake-up reason, event
Abstract	Describes the wake on Wi-Fi feature.

1 About this document

Wake on Wi-Fi is a power saving method, where the host is put in sleep while the Wi-Fi subsystem remains active. When the host is in sleep, Wi-Fi data traffic and functionality is not affected. The Wi-Fi subsystem is configured to wake up the host under certain wake-up conditions.

1.1 Supported products

The following products support the wake on Wi-Fi feature:

- 88W8987 ref.[1]
- 88W8997 ref.[2]
- 88W9098 ref.[3]
- IW416 ref.[4]
- IW610 ref.[5]
- AW611 ref.[6]
- IW611 ref.[7]
- IW612 ref.[8]

Refer to the release notes in the software release of the supported products for more information.

2 Wake-up conditions

The Wi-Fi device[1] is connected to the host system in STA or uAP mode. The host driver configures the wake-up conditions in the Wi-Fi device before the host goes to sleep. When receiving packets and if any wake-up condition is met, the Wi-Fi device triggers an interrupt to wake up the host. The interrupt cancels the host sleep mode.

Multiple conditions can be configured to wake up the host. For example:

- The Wi-Fi device receives broadcast data from the external AP.
- The Wi-Fi device receives unicast data from the external AP.
- MAC event on the Wi-Fi device:
  - ADDBA (Add Block ACK) request
  - DELBA (Delete Block ACK)
  - Group Rekeying
  - Disconnect
- The Wi-Fi device receives multicast data from the external AP.
- The Wi-Fi device receives a management frame (for example association, authentication, or beacon) from the external AP.

More complex conditions can also be configured. See Section 3.2.

1 The Wi-Fi device is one of the supported products listed in Section 1.1.

2.1 Wake-up methods

Two methods are possible for the Wi-Fi radio to wake up the host CPU.

- **In-band** – wakes up the host via the Wi-Fi host interface.
- **Out-of-band (OOB)** – wakes up the host through a wake-up signal muxed on a GPIO pin of the Wi-Fi device. Table 1 lists the GPIO pin number for the supported products. The wake-up signal is an active LOW signal. If the wake-up conditions are met, the wake-up signal is asserted for a configurable time duration (GAP time). When the host receives the signal, the host exits Sleep mode.

Table 1. Out-of-band GPIO pin numbers of the supported Wi-Fi devices

Supported product	Out-of-band GPIO
88W8987	GPIO[1]
88W8997	GPIO[14]
88W9098	GPIO[15]
AW611	GPIO[17]
IW416	GPIO[1]
IW610	GPIO[4]
IW611	GPIO[17]
IW612	GPIO[17]

Figure 1 illustrates a wake-up signal (active low).

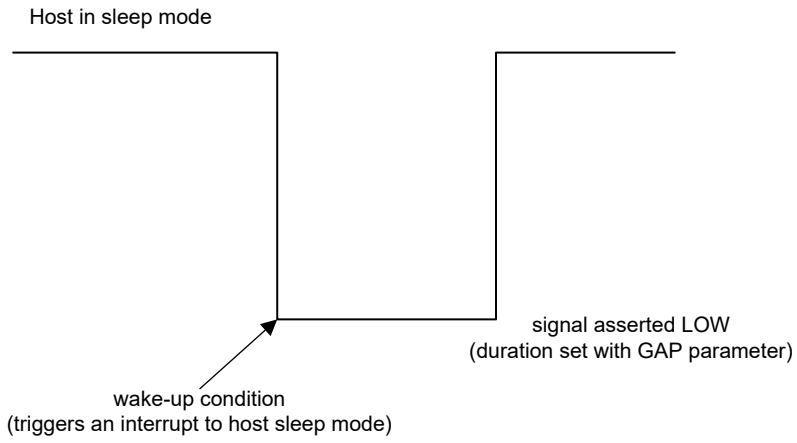

Figure 1. Illustration of a wake-up signal (active low)

3 Configuration

The wake-up conditions are configured via `hssetpara` command or using the MEF configuration file. The `hssetpara` command is used to set the wake-up method and the individual wake-up conditions. The MEF configuration file is used to set more complex or multiple wake-up conditions.

3.1 hssetpara command

The mlanutl / proc command `hssetpara`, is used to set the host sleep parameters, the wake-up condition, and the wake-up method. Find more information in the README_MLAN included in the software release package of the supported product (Section 1.1.

Command syntax for `mlanutl`:

```
./mlanutl mlan0 hssetpara <condition> <gpio> <gap> [extended configuration: <type>
 <parameter> ..]
```

Command syntax for `proc`:

```
echo "hssetpara=<condition> <gpio> <gap> [extended configuration: <type>
 <parameters> ..]" > /proc/mwlan/adapter0/config
```

Table 2. `hssetpara` command parameters

Parameter	Description
`condition`	Set to a bit to enable the wake-up condition -1 = cancel wake on Wi-Fi Bit 0 = 1 (receive Broadcast data) Bit 1 = 1 (receive Unicast data) Bit 2 = 1 (MAC event) Bit 3 = 1 (receive multicast data) Bit[4:5]: reserved Bit 6 = 1 (receive management frame) Bit 31 = 1 (Do not wake up when an IPv6 packet is received) Default is 0x7
`gpio`	Pin number of the GPIO used to wake up the host with the out-of-band method. 0x01 to 0x07 = GPIO pin number. Refer to Table 1 0xff = The in-band method is used to wake up the host (default).
`gap`	Time duration between the wake-up signal assertion and the wake-up event in ms (Figure 1). 0xff = default 0x01 to 0xFE = 1 ms to 254 ms Set to the desired GAP time.
type 1: `ind_gpio_level`	`ind_gpio`: GPIO number used to indicate the wake-up source. `level` = 0 – normal wake-up source (default) `level` = 1 – abnormal wake-up source

Table 2. `hssetpara` command parameters*...continued*

Parameter	Description
Group of parameters (optional) type 2: `event_force_ignore`	`event_force_ignore` bitmap, where each bit represents one wake-up reason event. The firmware ignores the wake-up reasons set in the bitmap. Wake-up reason event definition of each bit: Bit 0 = 1 – Disconnection from the AP Bit 1 = 1 – GTK/iGTK rekey failure (type of frame protection) Bit 2 = 1 – Extensible authentication protocol over LAN (Eapol) other bits – Reserved
Group of parameters (optional) type 3: `hs_wakeup_interval`	`hs_wakeup_interval`: time duration of the host sleep in ms.
Group of parameters (optional) type 4: `min_wake_holdoff`	`min_wake_holdoff`: minimum time duration of the wake holdoff in ms.

If this command is executed with no parameters, a get action is performed.

3.2 MEF configuration file

For more complex wake-up conditions, a memory efficient filtering (MEF) configuration file (*mef.conf*) is used. The location of the *mef.conf* file in the software release is:

mapp/mlanconfig/config/mef.conf.

The *mef.conf* file includes the data structure *mefcfg*.

Example of *mefcfg*:

```
mefcfg={
      Criteria=2   # Unicast frames are received during host sleep mode
      NumEntries=1 # Number of activated MEF entries
      mef_entry_0={
         mode= 1    # HostSleep mode
         action=1   # Discard Packet and Wake host
         filter_num=1     # Number of filters
         RPN=Filter_0     # only one filter is used
         Filter_0={
             type=0x41  # Byte comparison filter
             repeat=16  # Num of times to repeat Byte pattern
             byte=00:50:43:00:01:02  # Byte Pattern which is DUT MAC address
             offset=14  # offset in bytes into the received packet
             }
        }
}
```

In the example above, the host wakes up upon receiving a Magic Packet comprised of 16 repetitions of the DUT MAC address.

For a more detailed description of the parameters, refer to the README_MLAN file available in the software release of the supported product (Section 1.1).

Table 3 describes the parameters in `mefcfg` structure.

Table 3. mefcfg parameters

Parameter	Description
`criteria`	1 = Broadcast 2 = Unicast 8 = Multicast
`mode`	1 = Host Sleep 2 = Not in Host Sleep
`action`	0 = Discard packet, do not wake-up the host 1 = Discard packet, wake-up the host 3 = Allow packet, wake-up the host
`type`	0x41 = byte comparison 0x42 = decimal comparison 0x43 = bit comparison
`RPN`	Allows filters to be combined using logical AND (&&) and logical OR (||)
`filter_num`	Number of filters
`Filter_n`	Definition of each filter 0, 1, 2
`repeat`	Number of times the pattern is repeated
`byte`	Decimal value, hex value, or string of hexadecimal values separated by “:” as selected by Type

Table 3. mefcfg parameters*...continued*

Parameter	Description
`offset`	Number of bytes into the packet to start the comparison

Command to set *mef.conf*:

./`mlanutl mlan0 mefcfg <mef.conf>`

Note: *`hssetpara` must be used to configure the wake-up method.*

`wakeupreason` command

The mlanutl command `wakeupreason` is used to check the reason for the host wake-up.

Command syntax:

./`mlanutl <interface> wakeupreason`

Table 4. Command parameters

Parameter	Description
`interface`	Interface of the Wi-Fi device `mlan0` = interface for STA mode `uap0` = interface for AP mode

Table 5. `wakeupreason` command return parameters

Parameter	Description
`Reason`	0 = unknown 1 = Broadcast data matched 2 = Multicast data matched 3 = Unicast data matched 4 = Maskable event matched 5 = Non-maskable event matched 6 = Non-maskable condition matched (EAPOL rekey) 7 = Magic pattern matched 8 = Control frame matched 9 = Management frame matched

3.3 `wakeupreason` command

The mlanutl command `wakeupreason` is used to check the reason to interrupt the host sleep mode.

Command syntax:

./`mlanutl <interface> wakeupreason`

Table 6. Command parameters

Parameter	Description
`interface`	Interface of the Wi-Fi device `mlan0` = interface for STA mode `uap0` = interface for AP mode

Table 7 describes the command return parameters.

Table 7. `wakeupreson` command return parameters

Parameter	Description
`reason`	0 = unknown 1 = Broadcast data matched 2 = Multicast data matched 3 = Unicast data matched 4 = Maskable event matched 5 = Non-maskable event matched 6 = Non-maskable condition matched (EAPOL rekey) 7 = Magic pattern matched 8 = Control frame matched 9 = Management frame matched

4 Examples

This section contains examples of host wake-up configured by `hssetpara` command and MEF configuration.

4.1 `hssetpara` command examples

Command to receive multicast data using in-band method:

```
./mlanutl mlan0 hssetpara 8
```

Where:

- `8` is the value of `condition` parameter to receive multicast data.

Command to set a MAC event using out-of-band method and the default GAP time:

```
./mlanutl mlan0 hssetpara 4 16
```

Where:

- `4` is the value of `condition` parameter for MAC event.
- `16` is the value of `gpio` parameter for out-of-band.

Command to set a MAC event using out-of-band method and 160 ms GAP time:

```
./mlanutl mlan0 hssetpara 4 16 0xa0
```

Where:

- `4` is the value of `condition` parameter for MAC event.
- `16` is the value of `gpio` parameter for out-of-band.
- `0xa0` is the value of `gap` parameter for 160 ms.

Command to cancel the host wake-up condition:

```
./mlanutl mlan0 hssetpara -1
```

Where:

- `-1` is the value of `condition` parameter to cancel the host wake-up.

4.2 MEF configuration examples

This section shows examples of host wake-up for single and multiple conditions using MEF configuration.

4.2.1 Single condition example

In this example, the wake-up condition is the reception of a Magic Packet that contains 16 repetitions of the DUT MAC address.

Step 1 – Set in-band wake-up method using `hssetpara`. The condition will be defined later in *mef.conf*:

```
./mlanutl mlan0 hssetpara 0 0xff
```

Step 2 – Edit *mef.conf* with parameters. One condition is defined. [comment: not clear – are we doing something or is the condition already set? The example below is the same as in section MEF configuration file]

```
mefcfg={
   Criteria=2   # Unicast frames are received during host sleep mode
   NumEntries=1 # Number of activated MEF entries
   mef_entry_0={
      mode= 1    # HostSleep mode
      action=1   # Discard Packet and Wake host
      filter_num=1     # Number of filters
      RPN=Filter_0     # only one filter is used
      Filter_0={
         type=0x41  # Byte comparison filter
         repeat=16  # Num of times to repeat Byte pattern
         byte=00:50:43:00:01:02  # Byte Pattern which is DUT MAC address
         offset=14  # offset in bytes into the received packet
         }
      }
}
```

Step 3 – Set the host sleep configuration defined in *mef.conf*.

```
./mlanutl mlan0 mefcfg config/mef.conf
```

4.2.2 Example with multiple conditions

In this example, all three conditions listed below must be met for the host to wake up:

1. The Wi-Fi device received an ARP packet.
2. The source IP address is 192.168.0.104.
3. Destination Broadcast MAC address

Step 1 – Set out-of-band wake-up with 100 ms GAP time.

```
./mlanutl mlan0 hssetpara 0 16 0x64
```

Where:

- `0` is the value of `condition` parameter.
- `16` is the value of `gpio` parameter for out-of-band.
- `0x64` is the value of `gap` parameter for 100 ms.

Step 2 – Edit *mef.conf* to set the parameters.

```
mefcfg={
        mef_entry_0={
        mode=1   # HostSleep mode
        action=3   # Allow packet and Wake host
        filter_num=3   # Number of filters
        RPN=Filter_0 && Filter_1 && Filter_2 # Filters 0, 1 and 2 are required
        #Filter_0 looking for rx pkt with broadcast as the destination address
        Filter_0={
            type=0x41 # byte comparison
            repeat=6
            byte=ff # 6 x ff
            offset=0 # start of the packet
            }
        #Filter_1 looking for rx pkt with EtherType is 0x0806 (ARP)
        Filter_1={
            type=0x41 # byte comparison
            repeat=1
            byte=08:06 # ARP packet
            offset=20
            }
        #Filter_2 looking for rx pkt with ARP protocol IP address 192.168.0.104
        Filter_2={
            type=0x41 # byte comparison
            repeat=1
            byte=c0:a8:00:68   # 192.168.0.104 in hexadecimal
            offset=46
            }
        }
        }
```

Step 3 – Set the host sleep configuration defined in *mef.conf*.

```
./mlanutl mlan0 mefcfg config/mef.conf
```

Step 4 – Get the wake-up reason once the Wi-Fi device wakes up the host.

Example of command output for `reason = 7` (magic pattern matched):

```
Get wakeup reason response:  7
```

4.3 Host wake-up with unicast data

This example details how IW416 wakes up the host with unicast data.

Step 1 – Load the Wi-Fi driver in the kernel.

```
modprobe moal mod_para=nxp/wifi_mod_para.conf
```

Example of command output:

```
mlan: loading out-of-tree module taints kernel.
wlan: Loading MWLAN driver
wlan: Register to Bus Driver...
vendor=0x02DF device=0x9159 class=0 function=1
Attach moal handle ops, card interface type: 0x108
rps set to 0 from module param
SDIW416: init module param from usr cfg
card_type: SDIW416, config block: 0
cfg80211_wext=0xf
max_vir_bss=1
cal_data_cfg=none
ps_mode = 1
auto_ds = 1
host_mlme=enable
fw_name=nxp/sdiouartiw416_combo_v0.bin
SDIO: max_segs=128 max_seg_size=65535
rx_work=1 cpu_num=4
Enable moal_recv_amsdu_packet
Attach mlan adapter operations.card_type is 0x108.
wlan: Enable TX SG mode
wlan: Enable RX SG mode
Request firmware: nxp/sdiouartiw416_combo_v0.bin
 Wlan: FW download over, firmwarelen=591920 downloaded 583236
WLAN FW is active
on_time is 61875645706
VDLL image: len=8684
FW country code WW does not match with US
fw_cap_info=0x187ccf03, dev_cap_mask=0xffffffff
max_p2p_conn = 8, max_sta_conn = 8
Register NXP 802.11 Adapter mlan0
Register NXP 802.11 Adapter uap0
Register NXP 802.11 Adapter wfd0
wlan: version = SDIW416---16.92.21.p137.4-MM6X16437.p31-(FP92)
wlan: Register to Bus Driver Done
wlan: Driver loaded successfully
```

Step 2 – Use wpa_supplicant to connect the Wi-Fi device in STA mode (client) to the external access point.

```
wpa_supplicant -i mlan0 -Dnl80211 -c /etc/wpa_supplicant.conf
```

Example of command output:

```
Successfully initialized wpa_supplicant
rfkill: Cannot open RFKILL control device
wlan: mlan0 START SCAN
wlan: SCAN COMPLETED: scanned AP count=38
wlan: HostMlme mlan0 send auth to bssid bc:XX:XX:XX:b3:4d
mlan0:
wlan: HostMlme Auth received from bc:XX:XX:XX:b3:4d
wlan: HostMlme mlan0 Connected to bssid bc:XX:XX:XX:b3:4d successfully
mlan0:
wlan: Send EAPOL pkt to bc:XX:XX:XX:b3:4d
mlan0:
wlan: Send EAPOL pkt to bc:XX:XX:XX:b3:4d
woal_cfg80211_set_rekey_data return: gtk_rekey_offload is DISABLE
```

Step 3 – Verify the connectivity by running the ping command from Wi-Fi device in STA mode (client) to the external AP.

Step 4 – Disable auto address resolution protocol (ARP) for the host sleep mode.

```
./mlanutl mlan0 auto_arp 0
```

Step 5 – Verify that auto ARP is disabled.

```
./mlanutl mlan0 auto_arp
```

Example of command output:

```
Auto ARP is Disabled
```

Step 6 – Get the PHY number for mlan0 interface.

```
iw dev
```

Example of command output:

```
phy#0
        Interface wfd0
                ifindex 5
                wdev 0x3
                addr 9e:50:d1:45:37:09
                type managed
                txpower 24.00 dBm
        Interface uap0
                ifindex 4
                wdev 0x2
                addr 9e:50:d1:45:38:09
                type AP
                txpower 24.00 dBm
        Interface mlan0
                ifindex 3
                wdev 0x1
                addr 9c:50:d1:45:37:09
                ssid Netgear_2G
                type managed
                channel 6 (2437 MHz), width: 20 MHz, center1: 2437 MHz
                txpower 24.00 dBm
```

Step 7 – Enable wake on Wi-Fi on the PHY number. The PHY number (`phy#`) is taken from **Step 4**.

Command syntax:

```
iw phy#0 wowlan enable any
```

Step 8 – Verify that wake on Wi-Fi is enabled.

```
iw phy#0 wowlan show
```

Example of command output:

```
WoWLAN is enabled:
 * wake up on special any trigger
```

Step 9 – Enable SDIO wake-up. This step is not needed for PCIe interface.

```
echo enabled > /sys/bus/platform/devices/30b50000.mmc/power/wakeup
```

Step 10 – Set the host sleep parameters. Refer to Section 3.1.

```
echo "hssetpara=2 0xff" > /proc/mwlan/adapter0/config
```

OR

```
./mlanutl mlan0 hssetpara 2 0xff
```

Step 11 – Set the host in sleep mode.

```
echo mem >> /sys/power/state
```

Example of command output:

```
PM: suspend entry (deep)
Filesystems sync: 0.002 seconds
Freezing user space processes
Freezing user space processes completed (elapsed 0.001 seconds)
OOM killer disabled.
Freezing remaining freezable tasks
Freezing remaining freezable tasks completed (elapsed 0.001 seconds)
printk: Suspending console(s) (use no_console_suspend to debug)
PM: suspend devices took 0.236 seconds
Disabling non-boot CPUs ...
psci: CPU1 killed (polled 4 ms)
psci: CPU2 killed (polled 0 ms)
psci: CPU3 killed (polled 0 ms)
```

Step 12 – From the external AP, send unicast ping data to the wireless SoC. This step will trigger the host to wake up.

```
ping <DUT IP> -c 1
```

Expected output:

```
Enabling non-boot CPUs ...
Detected VIPT I-cache on CPU1
GICv3: CPU1: found redistributor 1 region 0:0x00000000388a0000
CPU1: Booted secondary processor 0x0000000001 [0x410fd034]
CPU1 is up
Detected VIPT I-cache on CPU2
GICv3: CPU2: found redistributor 2 region 0:0x00000000388c0000
CPU2: Booted secondary processor 0x0000000002 [0x410fd034]
CPU2 is up
Detected VIPT I-cache on CPU3
GICv3: CPU3: found redistributor 3 region 0:0x00000000388e0000
CPU3: Booted secondary processor 0x0000000003 [0x410fd034]
CPU3 is up
[drm] Pixel clock: 0 KHz, character clock: 0, bpc is 0-bit, fmt 0
[drm] Pixel clk (0 KHz) not supported, color depth (0-bit)
[drm:cdns_hdmi_phy_set_imx8mq] *ERROR* failed to set phy pclock
caam 30900000.crypto: registering rng-caam
PM: resume devices took 0.028 seconds
OOM killer enabled.
Restarting tasks ... done.
random: crng reseeded on system resumption
Hot alarm is canceled. GPU3D clock will return to 64/64
PM: suspend exit
```

Step 13 – Check the host wake-up reason.

```
./mlanutl mlan0 wakeupreason
```

Command output

```
Get wakeup reason response:  3
```

5 Note about the source code in the document

The example code shown in this document has the following copyright and BSD-3-Clause license:

Copyright 2025 NXP Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials must be provided with the distribution.
3. Neither the name of the copyright holder nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT HOLDER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

6 References

[1] Webpage – 88W8987: 2.4/5 GHz Dual-Band 1x1 Wi-Fi® 5 (802.11ac) + Bluetooth® Solution (link)
[2] Webpage – 88W8997: 2.4/5 GHz Dual-Band 2x2 Wi-Fi® 5 (802.11ac) + Bluetooth® Solution (link)
[3] Webpage – 88W9098: 2.4/5 GHz Dual-Band 2x2 Wi-Fi® 6 (802.11ax) + Bluetooth® (link)
[4] Webpage – IW416: 2.4/5 GHz Dual-Band 1x1 Wi-Fi® 4 (802.11n) + Bluetooth® Solution (link)
[5] Webpage – IW610: 2.4/5 GHz Dual-Band 1x1 Wi-Fi® 6 + Bluetooth Low Energy + 802.15.4 Tri-Radio Solution (link)
[6] Webpage – AW611: 2.4/5 GHz Dual-Band 1x1 Wi-Fi® 6 (802.11ax) + Bluetooth® Automotive Solution (link)
[7] Webpage – IW611: 2.4/5 GHz Dual-band 1x1 Wi-Fi® 6 (802.11ax) + Bluetooth® Solution (link)
[8] Webpage – IW612: 2.4/5 GHz Dual-Band 1x1 Wi-Fi® 6 (802.11ax) + Bluetooth® + 802.15.4 Tri-Radio Solution (link)

7 Revision history

Table 8. Revision history

Document ID	Release date	Description
AN14465 v.1.0	27 May 2025	• Initial version

Legal information

Definitions

Draft — A draft status on a document indicates that the content is still under internal review and subject to formal approval, which may result in modifications or additions. NXP Semiconductors does not give any representations or warranties as to the accuracy or completeness of information included in a draft version of a document and shall have no liability for the consequences of use of such information.

Disclaimers

Limited warranty and liability — Information in this document is believed to be accurate and reliable. However, NXP Semiconductors does not give any representations or warranties, expressed or implied, as to the accuracy or completeness of such information and shall have no liability for the consequences of use of such information. NXP Semiconductors takes no responsibility for the content in this document if provided by an information source outside of NXP Semiconductors.

In no event shall NXP Semiconductors be liable for any indirect, incidental, punitive, special or consequential damages (including - without limitation - lost profits, lost savings, business interruption, costs related to the removal or replacement of any products or rework charges) whether or not such damages are based on tort (including negligence), warranty, breach of contract or any other legal theory.

Notwithstanding any damages that customer might incur for any reason whatsoever, NXP Semiconductors' aggregate and cumulative liability towards customer for the products described herein shall be limited in accordance with the Terms and conditions of commercial sale of NXP Semiconductors.

Right to make changes — NXP Semiconductors reserves the right to make changes to information published in this document, including without limitation specifications and product descriptions, at any time and without notice. This document supersedes and replaces all information supplied prior to the publication hereof.

Suitability for use — NXP Semiconductors products are not designed, authorized or warranted to be suitable for use in life support, life-critical or safety-critical systems or equipment, nor in applications where failure or malfunction of an NXP Semiconductors product can reasonably be expected to result in personal injury, death or severe property or environmental damage. NXP Semiconductors and its suppliers accept no liability for inclusion and/or use of NXP Semiconductors products in such equipment or applications and therefore such inclusion and/or use is at the customer's own risk.

Applications — Applications that are described herein for any of these products are for illustrative purposes only. NXP Semiconductors makes no representation or warranty that such applications will be suitable for the specified use without further testing or modification.

Customers are responsible for the design and operation of their applications and products using NXP Semiconductors products, and NXP Semiconductors accepts no liability for any assistance with applications or customer product design. It is customer's sole responsibility to determine whether the NXP Semiconductors product is suitable and fit for the customer's applications and products planned, as well as for the planned application and use of customer's third party customer(s). Customers should provide appropriate design and operating safeguards to minimize the risks associated with their applications and products.

NXP Semiconductors does not accept any liability related to any default, damage, costs or problem which is based on any weakness or default in the customer's applications or products, or the application or use by customer's third party customer(s). Customer is responsible for doing all necessary testing for the customer's applications and products using NXP Semiconductors products in order to avoid a default of the applications and the products or of the application or use by customer's third party customer(s). NXP does not accept any liability in this respect.

Terms and conditions of commercial sale — NXP Semiconductors products are sold subject to the general terms and conditions of commercial sale, as published at https://www.nxp.com/profile/terms, unless otherwise agreed in a valid written individual agreement. In case an individual agreement is concluded only the terms and conditions of the respective agreement shall apply. NXP Semiconductors hereby expressly objects to applying the customer's general terms and conditions with regard to the purchase of NXP Semiconductors products by customer.

Export control — This document as well as the item(s) described herein may be subject to export control regulations. Export might require a prior authorization from competent authorities.

Suitability for use in non-automotive qualified products — Unless this document expressly states that this specific NXP Semiconductors product is automotive qualified, the product is not suitable for automotive use. It is neither qualified nor tested in accordance with automotive testing or application requirements. NXP Semiconductors accepts no liability for inclusion and/or use of non-automotive qualified products in automotive equipment or applications.

In the event that customer uses the product for design-in and use in automotive applications to automotive specifications and standards, customer (a) shall use the product without NXP Semiconductors' warranty of the product for such automotive applications, use and specifications, and (b) whenever customer uses the product for automotive applications beyond NXP Semiconductors' specifications such use shall be solely at customer's own risk, and (c) customer fully indemnifies NXP Semiconductors for any liability, damages or failed product claims resulting from customer design and use of the product for automotive applications beyond NXP Semiconductors' standard warranty and NXP Semiconductors' product specifications.

HTML publications — An HTML version, if available, of this document is provided as a courtesy. Definitive information is contained in the applicable document in PDF format. If there is a discrepancy between the HTML document and the PDF document, the PDF document has priority.

Translations — A non-English (translated) version of a document, including the legal information in that document, is for reference only. The English version shall prevail in case of any discrepancy between the translated and English versions.

Security — Customer understands that all NXP products may be subject to unidentified vulnerabilities or may support established security standards or specifications with known limitations. Customer is responsible for the design and operation of its applications and products throughout their lifecycles to reduce the effect of these vulnerabilities on customer's applications and products. Customer's responsibility also extends to other open and/or proprietary technologies supported by NXP products for use in customer's applications. NXP accepts no liability for any vulnerability. Customer should regularly check security updates from NXP and follow up appropriately.

Customer shall select products with security features that best meet rules, regulations, and standards of the intended application and make the ultimate design decisions regarding its products and is solely responsible for compliance with all legal, regulatory, and security related requirements concerning its products, regardless of any information or support that may be provided by NXP.

NXP has a Product Security Incident Response Team (PSIRT) (reachable at PSIRT@nxp.com) that manages the investigation, reporting, and solution release to security vulnerabilities of NXP products.

NXP B.V. — NXP B.V. is not an operating company and it does not distribute or sell products.

Trademarks

Notice: All referenced brands, product names, service names, and trademarks are the property of their respective owners.

NXP — wordmark and logo are trademarks of NXP B.V.

Tables

Figures

Contents