AN13958

Host Wake-up over Bluetooth or Bluetooth Low Energy (LE)

Rev. 1 — 14 August 2023 Application note

Document information

Information	Content
Keywords	Host wake-up over Bluetooth/Low Energy (LE) connectivity, GPIO interrupt by firmware
Abstract	Describes the procedure for host wake-up using Bluetooth or Bluetooth Low Energy (LE) on NXP wireless solutions (wireless SoCs).

1 Revision history

Revision history

Rev	Date	Description
v.1	14 August 2023	Initial version

2 Overview

This document describes the steps for host wake-up using Bluetooth/Bluetooth Low Energy (LE) on NXP Wi-Fi and Bluetooth combo solutions (wireless SoCs).

The implementation described in this document assumes the following:

- The host CPU is powered down, and the host stack is not running.
- The Wi-Fi/Bluetooth module is powered on, and the firmware is running.
- The host has a general-purpose input/output (GPIO) handler, which monitors the GPIO interrupt by firmware.
- Upon a GPIO interrupt, the host is brought to a running state and the host stack is initialized.

Note: *The implementation of the GPIO handler is not covered in this application note.*

For example, in a typical TV-remote control pair setup, both the TV and the remote control have Bluetooth integrated. If the TV is in Standby mode, and the power button on the remote control is pressed:

- The remote control initiates a Bluetooth connection.
- Upon receiving connection request from the remote control, the firmware generates the GPIO interrupt to wake up the host.

The general idea is to configure the firmware so it generates a GPIO interrupt when certain criterion is met, that is: when receiving the basic rate (BR)/enhanced data rate (EDR), and/or the Bluetooth LE connection request, and/or the advertising packet.

The following sections describe the different triggering points for the GPIO interrupt by firmware.

2.1 GPIO interrupt

This section describes the different trigger points for the GPIO interrupt by firmware.

Host wake-up over Bluetooth connectivity

Bluetooth asynchronous connectionless link (ACL) connection wake-up

The host wake-up is triggered when the ACL connection request from the remote device is received.

Host wake-up with Bluetooth LE connection

Upon receiving the Bluetooth LE connection request from the peer devices (included in the allowlist), the firmware generates the interrupt on the configured GPIO pin to wake up the host.

Host wake-up with Bluetooth LE scanning

Upon receiving the advertising packet from the peer devices (included in the allowlist) or through the defined scan filters, the firmware generates the interrupt on the configured GPIO pin to wake up the host.

Host wake-up with RX data

The Bluetooth LE connection is active with the peer device and the host is in Sleep mode. After receiving the data packet from the peer devices, the firmware generates the interrupt on the configured GPIO pin to wake up the host.

GPIO pin polarity

The GPIO pin remains active high in Standby mode. When the firmware receives a trigger packet, the GPIO pin is pulled active low for the GPIO_GAP (~250us) time duration. See Figure 1.

Note: *The GPIO_GAP time duration cannot be changed when using UART interface for Bluetooth.*

In Figure 1, GPIO is the GPIO pin number used to wake up the host. It can be any valid GPIO pin number. Section 5 for the GPIO pin number of the supported products.

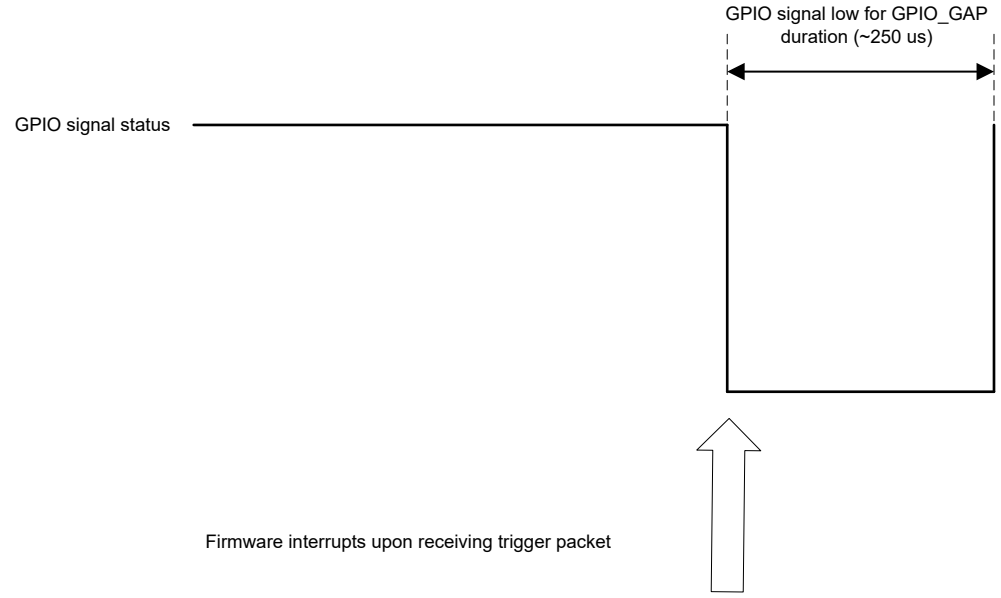

Figure 1. GPIO signal status

2.2 GPIO pin configuration

The GPIO pin configuration is specific to the wireless SoC and host interface. The GPIO pin can also be changed based on availability.

Table 1. GPIO pin configurations

Wireless SoC	Controller to host GPIO configuration pin
88W8987	GPIO[1] to GPIO[20]
IW416	GPIO[1] to GPIO[15]
88W9098	GPIO[1] to GPIO[24]
IW612	GPIO[19]

3 Host wake-up over Bluetooth

3.1 Host wake-up via Bluetooth ACL connection request

In this implementation, the host wake-up is triggered when there is an ACL connection request from any Bluetooth remote device.

Step 1 - Enable page scan.

Enable page scan on the Device Under Test (DUT) to accept the incoming ACL connection request.

```
hciconfig hci0 pscan      // Set the device in connectable mode
```

Step 2 - Configure the GPIO pin for the interrupt.

Refer to Section 5 for the commands to configure the GPIO pin of the supported products. The firmware will generate an interrupt on the configured GPIO pin.

Step 3 - Enable host sleep.

Refer to Section 6 for the commands to enable host sleep for the supported products.

Step 4 - Initiate the ACL connection from the remote device to the DUT.

```
hcitool -i hci0 cmd cc <BD_Address_DUT>
```

The host monitors the configured GPIO pin for interrupts. After receiving the ACL connection request, the firmware generates the interrupt on the configured GPIO pin. Once the interrupt is detected, the host mode changes to active.

4 Host wake-up over Bluetooth LE

4.1 Host wake-up over Bluetooth LE connection

In this case, the Bluetooth controller is advertising, and the host is asleep.

Step 1 - Add the peer device to the DUT allowlist:

```
hcitool -i hci0 cmd 0x08 0x0011 <Address_Type> <BD_Address>
```

Table 2. Command parameters

Parameter	Description
`Address_Type`	The device address type 0x00 = public device address 0x01 = random device address
`BD_Address`	The public or random Bluetooth device address to be added to the allowlist

Step 2 - Configure the GPIO pin.

Refer to Section 5 for the commands to configure the GPIO pin of the supported products. The firmware will generate an interrupt on the configured GPIO pin.

Step 3 - Start advertising on the DUT.

- Set the advertising parameter

```
hcitool -i hci0 cmd 08 06 00 02 00 02 00 00 00 00 00 00 00 00 00 07 03
```

Where `00` is the advertising type set to connectable and scannable undirected advertisement, `07` is the advertising policy set to scan, and `03` is for connection requests allowed only from devices in the allowlist.

- Set the advertising data

```
hcitool -i hci0 cmd 08 08 1F 00 99 88 77 66 55 44 33 22 11 00 99 88 77 66 55 44 33 22 11
 00 99 88 77 66 55 44 33 22 11 00
```

- Enable advertising

```
hcitool -i hci0 cmd 08 0A 01
```

Step 4 - Enable host sleep

Refer to Section 6 for the commands to enable host sleep for the supported products.

Step 5 - Enable Bluetooth LE connection from the remote device with *Initiator_Filter_Policy* set to the allowlist.

Expected results

After receiving the Bluetooth LE connection request from the peer device (included in the allowlist), the firmware generates the interrupt on the configured GPIO pin.

4.2 Host wake-up over Bluetooth LE scanning

In this case, the Bluetooth controller is scanning, and the host is asleep.

Step 1 - Add the peer device to the DUT allowlist

```
hcitool -i hci0 cmd 0x08 0x0011 <Address_Type> <Address>
```

Table 3. Command parameters

Parameter	Description
Address_Type	The device address type 0x00 = public device address 0x01 = random device address
Address	The public or random device address to be added to the allowlist

Step 2 - Set the scan parameters on the DUT

```
hcitool -i hci0 cmd 0x08 0x000B <Parameters>
```

Table 4. Command parameters

Parameter	Description
Parameters	Use one of the following values: LE_Scan_Type LE_Scan_Interval LE_Scan_Window Own_AddresS_Type Scanning_Filter_Policy (0x01 = set to allowlist) Other parameters can be referred through Bluetooth specification 4.1.

Step 3 - Configure the GPIO pin.

Refer to Section 5 for the commands to configure the GPIO pin of the supported products. The firmware will generate an interrupt on the configured GPIO pin.

Step 4 - Enable Bluetooth LE scan on the DUT.

```
hcitool -i hci0 cmd 0x08 0x000C 0x01 0x01
```

Step 5 - Enable host sleep.

Refer to Section 6 for the commands to enable host sleep for the supported products.

Step 6 - Start advertising on the peer device.

- Set the advertising parameter.

```
hcitool -i hci0 cmd 08 06 00 02 00 02 00 00 00 00 00 00 00 00 00 07 03
```

Where **00** is the advertising type set to connectable and scannable undirected advertisement, **07** is the advertising policy set to scan, and **03** is for connection requests allowed only from devices in the allowlist.

- Set the advertising data.

```
hcitool -i hci0 cmd 08 08 1F 00 99 88 77 66 55 44 33 22 11 00 99 88 77 66 55 44 33 22 11
 00 99 88 77 66 55 44 33 22 11 00
```

- Enable advertising.

```
hcitool -i hci0 cmd 08 0A 01
```

Expected results

After receiving the Bluetooth LE advertising packet from the peer device (from the allowlist, scan filter), the firmware generates an interrupt on the configured GPIO pin.

4.3 Host wake-up over Rx data packet and active Bluetooth LE connection

In this case, the Bluetooth LE connection is active, and the host is asleep.

Step 1 - Add the peer device to the DUT allowlist.

```
hcitool -i hci0 cmd 0x08 0x0011 <Address_Type> <BD_Address>
```

Table 5. Command parameters

Parameter	Description
`Address_Type`	The device address type 0x00 = public device address 0x01 = random device address
`BD_Address`	The public or random Bluetooth device address to be added to the allowlist

Step 2 - Configure the GPIO pin.

Refer to Section 5 for the commands to configure the GPIO pin of the supported products. The firmware will generate an interrupt on the configured GPIO pin.

Step 3 - Start advertising on the DUT.

- Set the advertising parameter.

```
hcitool -i hci0 cmd 08 06 00 02 00 02 00 00 00 00 00 00 00 00 00 07 03
```

Where `00` is the advertising type set to connectable and scannable undirected advertisement, `07` is the advertising policy set to scan, and `03` is for connection requests allowed only from devices in the allowlist.

- Set the advertising data.

```
hcitool -i hci0 cmd 08 08 1F 00 99 88 77 66 55 44 33 22 11 00 99 88 77 66 55 44 33 22 11
 00 99 88 77 66 55 44 33 22 11 00
```

- Enable advertising.

```
hcitool -i hci0 cmd 08 0A 01
```

Step 4 - Establish the Bluetooth LE connection with the remote device.

Step 5 - Send Bluetooth LE data packets from the remote device.

When the connection is established and host sleep is enabled, the firmware sends an interrupt on the configured GPIO pin.

Example of command on Bluez stack to send a single data packet:

```
hcitool -i hci1 acldat -p dcd -d 0x5454 -c 1 -s 10 -P i <BD_Addr_Dut> -H 128
```

Expected results

After receiving any data packet from the remote device, the firmware generates the interrupt on the configured GPIO pin.

5 Commands to configure GPIO pins

The configuration command is specific to the wireless SoC and host interface. The first interface is for Wi-Fi, and the second is for Bluetooth. For example:

- PCIe-UART: the Wi-Fi interface uses PCIe and the Bluetooth interface uses UART.
- SD-UART: the Wi-Fi interface uses SDIO and the Bluetooth interface uses UART.

Table 6. Commands to configure GPIO pin using UART interface

SoC	Interface	Command	GPIO
88W8987	SD-UART	`hcitool -i hci0 cmd 0x3F 0x53 0x03 0x04 0x01 0xFF`	GPIO[4]
IW416	SD-UART	`hcitool -i hci0 cmd 0x3F 0x53 0x03 0x04 0x01 0xFF`	GPIO[4]
88W9098	SD-UART/ PCIe-UART (88W9098)	`hcitool -i hci0 cmd 0x3F 0x53 0x03 0x10 0x01 0xFF`	GPIO[16]
IW612	SD-UART (IW612)	`hcitool -i hci0 cmd 0x3F 0x53 0x03 0x13 0x01 0xFF`	GPIO[19]

6 Command to configure host sleep

The configuration command is specific to the wireless SoC and host interface. The first interface is for Wi-Fi, and the second is for Bluetooth.

For example:

- PCIe-UART: the Wi-Fi interface uses PCIe, and Bluetooth interface uses UART.
- SD-UART: the Wi-Fi interface uses SDIO, and Bluetooth interface uses UART.

Table 7. Commands to configure host sleep using UART interface

SoC	Interface	Command
88W8987	SD-UART	`echo "psmode=1" > /proc/mbt_uart/hci0/config`
IW416	SD-UART	
88W9098	SD–UART/ PCIe-UART (88W9098)	
IW612	SD-UART (IW612)	

7 Examples

7.1 Host wake-up over 88W8987 SD-UART with Rx data packet and active Bluetooth LE connection

In this example, the Bluetooth LE connection is active and the host is asleep.

Step 1 - Add the peer device to the DUT allowlist.

```
hcitool –i hci0 cmd 0x08 0x0011 00 F7 EE 6B 83 15 00
```

Where `00` is the address type of the public device, and `F7 EE 6B 83 15 00` is the Bluetooth device address

Step 2 - Configure the GPIO pin

```
hcitool -i hci0 cmd 0x3F 0x53 0x03 0x04 0x01 0xFF  // Configure GPIO[4] for interrupt
```

The firmware will generate the interrupt on the configured GPIO[4] pin.

Step 3 - Start advertising on the DUT.

- Set the advertising parameter

```
hcitool -i hci0 cmd 08 06 00 02 00 02 00 00 00 00 00 00 00 00 00 07 03
```

- Set the advertising data

```
hcitool -i hci0 cmd 08 08 1F 00 99 88 77 66 55 44 33 22 11 00 99 88 77 66 55 44 33 22 11
 00 99 88 77 66 55 44 33 22 11 00
```

- Enable advertising

```
hcitool -i hci0 cmd 08 0A 01
```

Step 4 - Set the Bluetooth LE connection from the remote device

```
hcitool –i hci1 lecc 00:50:43:21:30:CF
```

Where `00:50:43:21:30:CF` is the DUT address

Step 5 - Enable Host sleep with keeping LE connection active.

```
echo "psmode=1" >/proc/mbt_uart/hci0/config
```

Step 6 - Send Bluetooth LE data packet from the remote device to the DUT

```
hcitool –i hci1 acldat –p dcd –d 0x5454 –c 1 –s 10 –P i 00:50:43:21:30:CF -H 128
```

Where `00:50:43:21:30:CF` is the DUT address

Expected results

Upon receiving a data packet from the remote device, the firmware generates the interrupt on the configured GPIO[4] pin.

7.2 Host wake-up over Bluetooth LE connection on 88W9098 SD-UART

In this example, the Bluetooth controller is advertising and the host is asleep.

Step 1 - Add the peer device to the DUT allowlist.

```
hcitool -i hci0 cmd 0x08 0x0011 <Address_Type> <BD_Address>
```

Table 8. Command parameters

Parameter	Description
`Address_Type`	The device address type 0x00 = public device address 0x01 = random device address
`BD_Address`	The public or random Bluetooth device address to be added to the allowlist

Step 2 - Configure the GPIO pin.

```
hcitool -i hci0 cmd 0x3F 0x53 0x03 0x10 0x01 0xFF  // Configure GPIO[16] for interrupt
```

Step 3 - Start advertising on the DUT

- Set the advertising parameter

```
hcitool -i hci0 cmd 08 06 00 02 00 02 00 00 00 00 00 00 00 00 00 07 03
```

Where `00` s the advertising type set to connectable and scannable undirected advertisement , `07` is the advertising policy set to scan, and `03` is for connection requests allowed only from devices in the allowlist.

- Set the advertising data

```
hcitool -i hci0 cmd 08 08 1F 00 99 88 77 66 55 44 33 22 11 00 99 88 77 66 55 44 33 22 11
 00 99 88 77 66 55 44 33 22 11 00
```

- Enable advertising

```
hcitool -i hci0 cmd 08 0A 01
```

Step 4 - Enable host sleep

```
echo "psmode=1" > /proc/mbt_uart/hci0/config
```

Step 5 - Establish the Bluetooth LE connection from the remote device

```
hcitool -i hci1 lecc 00:50:43:21:30:CF
```

Where `00:50:43:21:30:CF` is the DUT address

Expected result

After receiving the Bluetooth LE connection request from the remote device, the controller accepts the Bluetooth LE connection request. the firmware generates an interrupt on the configured GPIO[16] pin.

8 References

Table 9. References

Reference type	Description
Data sheet	88W8987 - Wi-Fi 5 and Bluetooth 5 Single-chip SoC
Data sheet	IW416 - Wi-Fi 4 and Bluetooth 5.2 Combo SoC
Data sheet	88W9098 - Wi-Fi 6 Concurrent Dual Wi-Fi (CDW) and Bluetooth 5.3 Combo SoC
Data sheet	IW612 - Wi-Fi 6 Concurrent Dual Wi-Fi (CDW), Bluetooth 5.2 and 802.15.4 Tri-Radio SoC

9 Acronyms and abbreviations

Table 10. Acronyms and abbreviations

Acronym	Definition
ACK	Acknowledgment
ACL	Asynchronous connectionless link
BR	Basic rate
CPU	Central processor unit
DUT	Device under test
EDR	Enhanced data rate
GPIO	General Purpose Input/Output
HCI	Host controller interface
HID	Human interface device
L2CAP	Logical link control and adaptation protocol
LE	Low energy
LMP	Link manager protocol
PSM	Protocol service multiplexer
SD	Secure digital
SDIO	Secure digital input/output
SoC	System-on-Chip
PCIe	Peripheral component interconnect express
UART	Universal asynchronous receiver/transmitter
USB	Universal serial bus

10 Legal information

10.1 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.

10.2 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 http://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.

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.

10.3 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.

Bluetooth — the Bluetooth wordmark and logos are registered trademarks owned by Bluetooth SIG, Inc. and any use of such marks by NXP Semiconductors is under license.

Tables

Figures

Contents

For more information, please visit: http://www.nxp.com

Date of release: 14 August 2023
Document identifier: AN13958