GigaDevice Semiconductor Inc.

Process description about USB suspend and wake up

Application Note
AN077

Table of Contents

List of Figures

List of Tables

1. Introduction

This application note mainly introduces the suspend and wake-up process of USB devices, which are divided into two types of peripherals: USBD and USBFS. Since the related process of USBHS and USBFS are not obviously different, they will not be introduced separately. The product series which is applicable to this article are shown in ***Table 1-1. Applicable products***.

Table 1-1. Applicable products

Type	peripheral	selection
MCU	USBD	GD32F103xx
		GD32F150xx
		GD32F303xx
		GD32E503xx
		GD32EPRTxx
		GD32L233xx
	USBFS	GD32F105xx/GD32F107xx
		GD32F205xx/GD32F207xx
		GD32F305xx/GD32F307xx/GD32F403xx
		GD32F350xx
		GD32F4xx
		GD32E103xx/GD32E113xx
		GD32C103xx/GD32C113xx
		GD32VF103xx
		GD32W515xx
	USBHS	GD32F4xx
		GD32E505xx/GD32E507xx/GD32E508xx

In particular, waking up is relative to suspend. While the device is in suspend state, any activity on the bus (not idle signal) can wake up the device and exit from the low-power mode. This function is called as "normal wake-up".

Similarly, the device can also wake up the host, for example, USB keyboard wake up the computer when the computer is in standby mode. This function is called as "remote wake up".

2. USBD suspend and wake up

2.1. USBD suspend condition

Normally, the USB host will send a SOF packet at intervals (the interval is 1 millisecond for full speed device and 125 microsecond for high speed device). According to the USB protocol, if the USB bus is in the idle state for more than 3ms, the device will take it as a suspend signal and then require the device to enter the suspended state within 10ms.

2.2. USBD suspend process

When the USB host enters the sleep state, it will no longer issue SOF. And the ESOFIF bit will be set, triggering the device ESOF interrupt:

- When the ESOFIF flag is firstly set: ISR software clears this flag.
- When the ESOFIF flag is secondly set: ISR software clears this flag.
- When the ESOFIF flag is thirdly set: SPSIF is set at the same time, the device enters the suspend interrupt service function, and the MCU enters in a low power state. The ESOFIF flag is still in setting.

2.3. USBD normal wake up

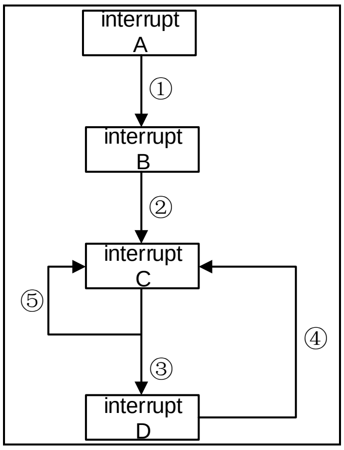
Taking the USB HID Keyboard device as an example, after the user adjusts the USB host to sleep mode, the USB bus is in the Idle state. After a period of time, the GD32-EVAL device enters the suspend state and the MCU enters a low-power mode. When the user operates other USB devices which are connected to the USB host, causing the host to wake up. A non-Idle signal will appear on the USB bus to wake up the GD32-EVAL device.

A series of interrupts will be triggered in the normal wake-up process of USBD. There are four main interrupts in USBD devices as below:

A. USBD_WKUP_IRQ
B. USBD_LP_IRQ.Suspend
C. USBD_LP_IRQ.ESOF
D. USBD_LP_IRQ.Wakeup

As shown in the figure below, after the host sends the resume signal, device triggers interrupt A to wake up the MCU firstly. Then, continue to execute from the position where the MCU halted before (interrupt B). Because the ESOFIF flag is also set when it is suspended, interrupt C is followed to be executed. Next, enter the D interrupt, and restore the MCU clock in this interrupt, continue to enter the interrupt C until restore normal communication.

Figure 2-1. USBD device normal wake-up interrupt process



Note: When configuring the USBD_LP_IRQ and USBD_WKUP_IRQ priorities, the latter must have a higher preemption priority than the former.

2.4. USBD remote wake up

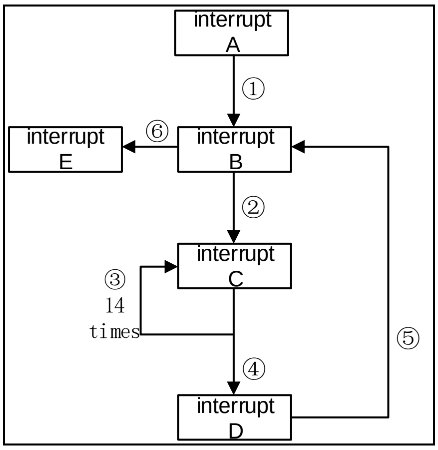
Taking the USB HID Keyboard device as an example, after the user adjusts the USB host to sleep mode, the USB bus is in the Idle state. After a period of time, the GD32 device enters the suspend state and the MCU enters a low-power mode. When the user presses the specific Key on the EVAL board, the EXTI interrupt will be triggered and the MCU will be activated. Then device send a resume signal to the USB bus in the EXTI ISR to wake up the Host. After the resume signal has been sent, the device will receive the SOF packet and other information again.

A series of interrupts will be triggered in the remote wake-up process of USBD. There are five main interrupts in USBD devices as below:

A. KEY_EXTI_IRQ
B. USBD_LP_IRQ.Suspend
C. USBD_LP_IRQ.ESOF
D. USBD_WKUP_IRQ
E. USBD_LP_IRQ.Wakeup

When the specific KEY is pressed, the program enters the A interrupt handler. The MCU clock is reconfigured and device starts the resume signal. Then, the program continues to execute from the position where the MCU stopped before (interrupt B). Because the ESOFIF flag is also set when it is suspended, interrupt C is followed to be executed. Then program will continue to execute C (a total of 14+1 times). After the Resume signal has been sent, and the Resume signal is turned off in the C interrupt. Based on test, there will be two rounds of suspend and wake-up interrupts D→B→E, with C interrupts triggered in the course.

Figure 2-2. USBD device remote wake-up interrupt process



Note: When user configure the interrupt priority of USBD_LP_IRQ, USBD_WKUP_IRQ, and KEY_EXTI_IRQ, the preemption priority of the last two interrupts must be higher than the first one, and comparison between the preemption priority of the latter two interrupts is not certain.

3. USBFS suspend and wake up

3.1. USBFS suspend condition

Refer to Section 2.1 for details.

3.2. USBFS suspend Process

When the USB host enters the sleep state, the device detects that the USB bus is idle for 3ms, and will generate an early suspend interrupt (ESP), and generate a suspend interrupt (SP) after 3ms. Device enter the low-power mode in the SP ISR.

3.3. USBFS normal wake up

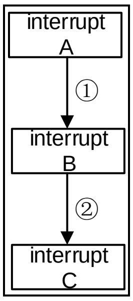
Taking the USB HID Keyboard device as an example, after the user adjusts the USB host to sleep mode, the USB bus is in the Idle state. After a period of time, the GD32-EVAL Device enters the Suspend state. And the MCU enters a low-power mode. When the user operates other USB devices which is connected to the USB host, causing the host to wake up. A non-Idle signal will appear on the USB bus to wake up the GD32-EVAL device.

A series of interrupts will be triggered during the normal wake-up process of USBFS. There are three main interrupts as below:

A. USBFS_WKUP_IRQ
B. USBFS_IRQ.Suspend
C. USBFS_IRQ.Wakeup

As shown in the figure below, after the host sends the resume signal, device triggers interrupt A to wake up the MCU firstly, restore the MCU clock and USB clock. Then, continue to execute from the position where the MCU stopped (interrupt B). Subsequently, enter interrupt C to restore the USB transfer state.

Figure 3-1. USBFS devices normal wake-up interrupt process



Note: When configuring the USBFS_IRQ and USBFS_WKUP_IRQ priorities, the latter must have a higher preemption priority than the former.

3.4. USBFS remote wake up

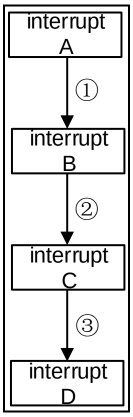
Taking the USB HID Keyboard device as an example, after the user adjusts the USB host to sleep mode, the USB bus is in the Idle state. After a period of time, the GD32 Device enters the suspend state and the MCU enters a low-power mode. When the user presses the specific Key on the EVAL board, the EXTI interrupt will be triggered and the MCU will be activated. Then device send a resume signal to the USB bus in the EXTI ISR to wake up the host.

A series of interrupts will be triggered during the remote wake-up process of USBFS. There are four main interrupts in USBFS devices as below:

A. KEY_EXTI_IRQ
B. USBFS_WKUP_IRQ
C. USBFS_IRQ.Suspend
D. USBFS_IRQ.Wakeup

When the specific KEY is pressed, the program enters the A interrupt handler, the MCU clock is reconfigured and device starts the resume signal. After 5ms, device turn off the resume signal, and the USB host has been woken up at this time. Then, the host will restore the USB signal, and then trigger interrupt B (because the interrupt priority of B is higher than the interrupt priority of C, B should be executed first). Next, continue to execute from the position where the MCU stopped running before the suspension (at interrupt C). Then, execute the D interrupt.

Figure 3-2. USBFS device remote wake-up interrupt process



Note:

1. When user configure the interrupt priority of USBFS_LP_IRQ, USBFS_WKUP_IRQ, and KEY_EXTI_IRQ, the preemption priority of the last two interrupts must be higher than the first one, and comparison between the preemption priority of the latter two interrupts is not certain.

2. When the preemption priority of USBFS_WKUP_IRQ is higher than that of

KEY_EXTI_IRQ, interrupt B preempts interrupt A in the sequence A→B→A→C→D.

4. Revision history

Table 4-1. Revision history

Revision No.	Description	Date
1.0	Initial Release	Sep.15 2022

Important Notice