

**GigaDevice Semiconductor Inc.**

**GD32F5HCP-START**

**Arm<sup>®</sup> Cortex<sup>®</sup>-M33 32-bit MCU**

## **User Guide**

Revision 1.0

(Apr. 2026)

# Table of Contents

<b>Table of Contents .....</b>	<b>1</b>
<b>List of Figures .....</b>	<b>3</b>
<b>List of Tables .....</b>	<b>4</b>
<b>1. Summary .....</b>	<b>5</b>
<b>2. Function Pin Assign .....</b>	<b>6</b>
<b>3. Getting started .....</b>	<b>7</b>
<b>4. Hardware layout overview.....</b>	<b>8</b>
4.1. Power supply .....	8
4.2. Boot option .....	8
4.3. LED .....	8
4.4. KEY .....	9
4.5. USART .....	9
4.6. Extension .....	10
4.7. GD-Link .....	10
4.8. USB.....	10
4.9. MCU .....	11
<b>5. Routine use guide.....</b>	<b>12</b>
<b>5.1. GPIO_Runing_Led .....</b>	<b>12</b>
5.1.1. DEMO Purpose.....	12
5.1.2. DEMO Running Result .....	12
<b>5.2. GPIO_Key_Polling_mode.....</b>	<b>12</b>
5.2.1. DEMO Purpose.....	12
5.2.2. DEMO Running Result .....	12
<b>5.3. EXTI_Key_Interrupt_mode.....</b>	<b>13</b>
5.3.1. DEMO Purpose.....	13
5.3.2. DEMO Running Result .....	13
<b>5.4. USART_Echo_Interrupt_mode.....</b>	<b>13</b>
5.4.1. DEMO Purpose.....	13
5.4.2. DEMO Running Result .....	13
<b>5.5. TIMER_Breath_LED .....</b>	<b>14</b>
5.5.1. DEMO Purpose.....	14
5.5.2. DEMO Running Result .....	14
<b>5.6. USB_Device .....</b>	<b>14</b>
5.6.1. HID_Keyboard .....	14
5.6.2. CDC_ACM .....	15
<b>5.7. USB_Host.....</b>	<b>16</b>
5.7.1. HID_Host .....	16

---

5.7.2.	MSC_Host.....	16
<b>5.8.</b>	<b>Trustzone .....</b>	<b>17</b>
5.8.1.	DEMO Purpose.....	17
5.8.2.	DEMO Running Result .....	17
<b>6.</b>	<b>Revision history .....</b>	<b>18</b>

## List of Figures

Figure 4-1. Schematic diagram of power supply.....	8
Figure 4-2. Schematic diagram of boot option .....	8
Figure 4-3. Schematic diagram of LED function .....	8
Figure 4-4. Schematic diagram of Key function .....	9
Figure 4-5. Schematic diagram of USART .....	9
Figure 4-6. Schematic diagram of Extension.....	10
Figure 4-7. Schematic diagram of GD-Link.....	10
Figure 4-8. Schematic diagram of USB .....	10
Figure 4-9. Schematic diagram of MCU.....	11

## List of Tables

Table 2-1. Function pin assignment.....	6
Table 6-1. Revision history .....	18

## 1. Summary

GD32F5HCP-START uses GD32F5HCPIQ6 as the main controller. It uses GD-Link Type-C interface to supply 5V power. SWD, Reset, Boot, User button key, LED, USB, GD-Link and Extension Pins are also included. For more details, please refer to GD32F5HCP-START-V1.0 schematic.

## 2. Function Pin Assign

**Table 2-1. Function pin assignment**

Function	Pin	Description
LED	PA6	LED1
	PA7	LED2
	PC4	LED3
	PB0	LED4
RESET	NRST	K1-Reset
KEY	PA2	K2-Wakeup & Tamper
	PA4	K3- User key2
	PA5	K4-User key1
USART2	PB10	USART2_TX
	PB11	USART2_RX
USB	PB13	USBFS_DM
	PB12	USBFS_DP

### 3. Getting started

The START board uses GD-Link Type-C interface to get power DC +5V, which is the hardware system normal work voltage. A J-Link tool or GD-Link on board is necessary in order to download and debug programs. Select the correct boot mode and then power on, the LEDPWR will turn on, which indicates that the power supply is OK.

There are Keil version, IAR version and GD32EBuilder version of all projects. Keil version of the projects are created based on Keil MDK-ARM 5.29 uVision5. IAR version of the projects are created based on IAR Embedded Workbench for ARM 8.32.1 and GD32EBuilder version of the projects are created based on GD32EmbeddedBuilder\_v1.5.5\_Rel. During use, the following points should be noted:

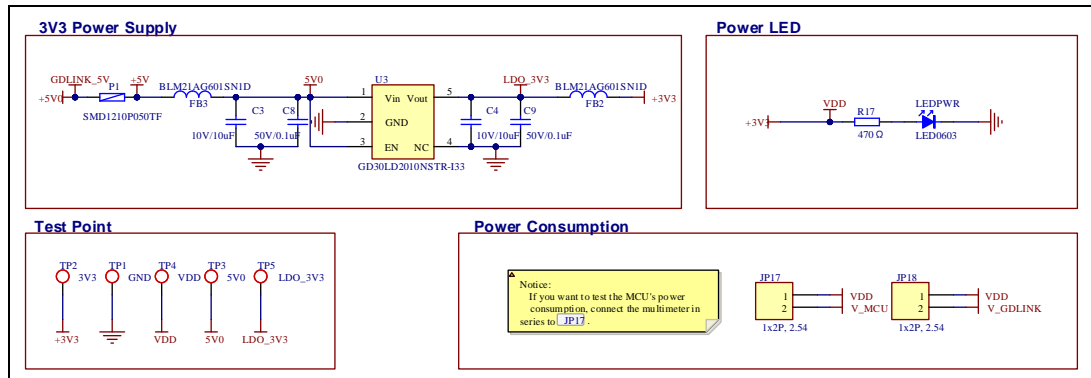
1. If you use Keil uVision5 to open the project. In order to solve the "Device Missing (s)" problem, the latest version of GigaDevice.GD32W51x\_F5HC\_DFP (URL: <https://www.gd32mcu.com>) should be installed to load related files.
2. If you use IAR to open the project, the latest version of IAR\_GD32W51x\_F5HC\_ADDON (URL: <https://www.gd32mcu.com>) should be installed to load related files.



## 4. Hardware layout overview

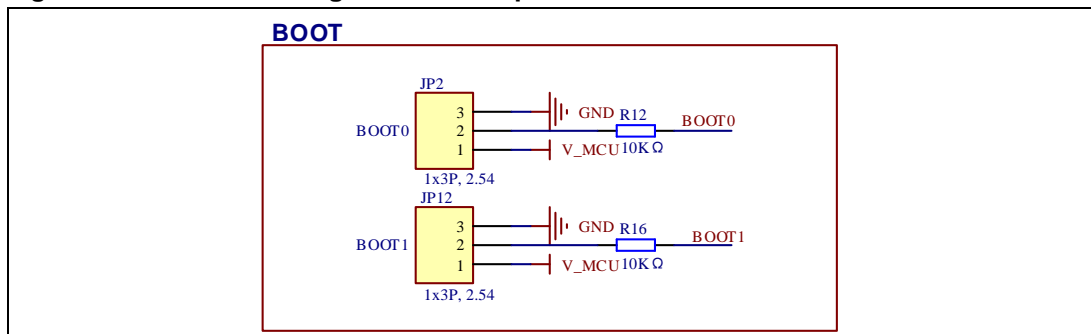
### 4.1. Power supply

Figure 4-1. Schematic diagram of power supply



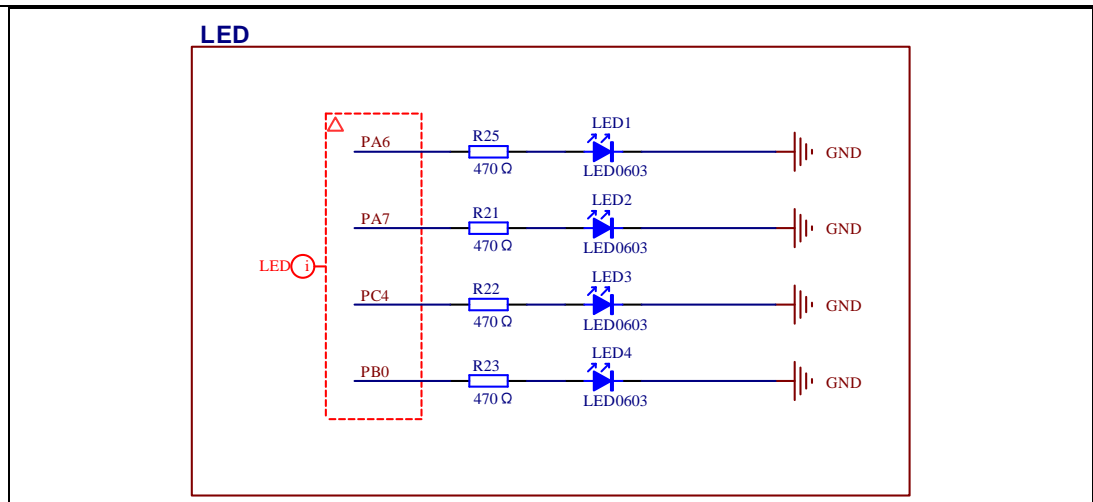
### 4.2. Boot option

Figure 4-2. Schematic diagram of boot option



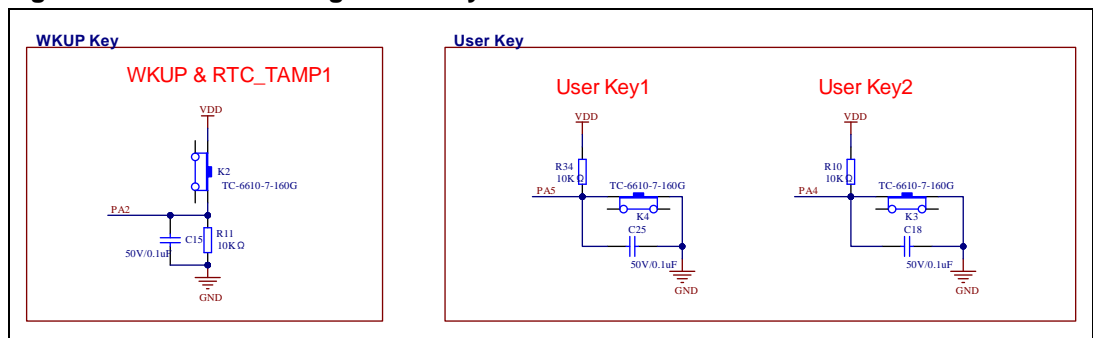
### 4.3. LED

Figure 4-3. Schematic diagram of LED function



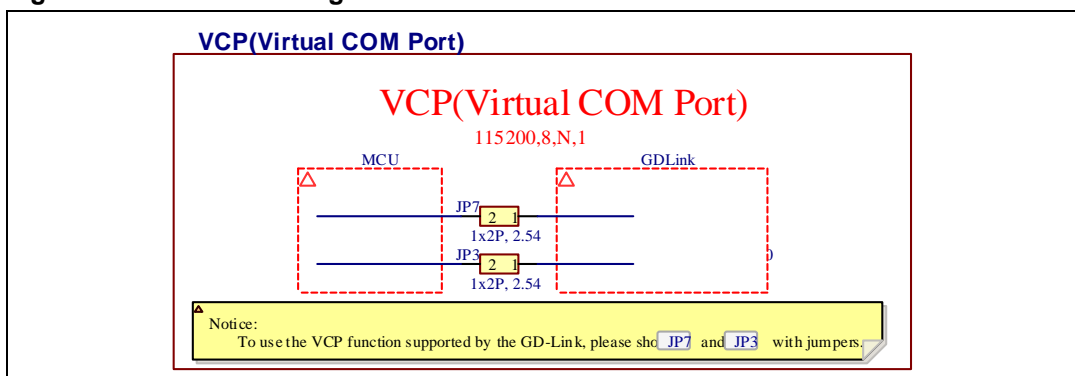
#### 4.4. KEY

Figure 4-4. Schematic diagram of Key function



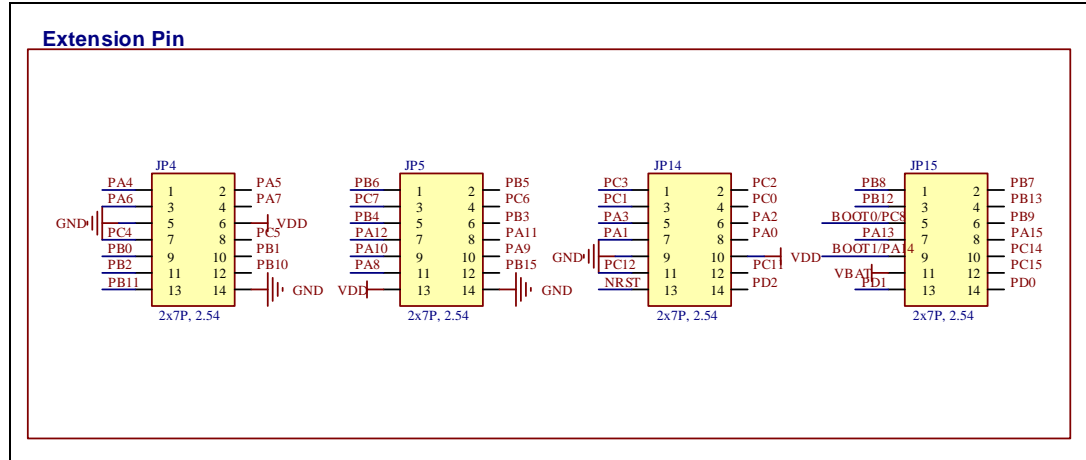
#### 4.5. USART

Figure 4-5. Schematic diagram of USART



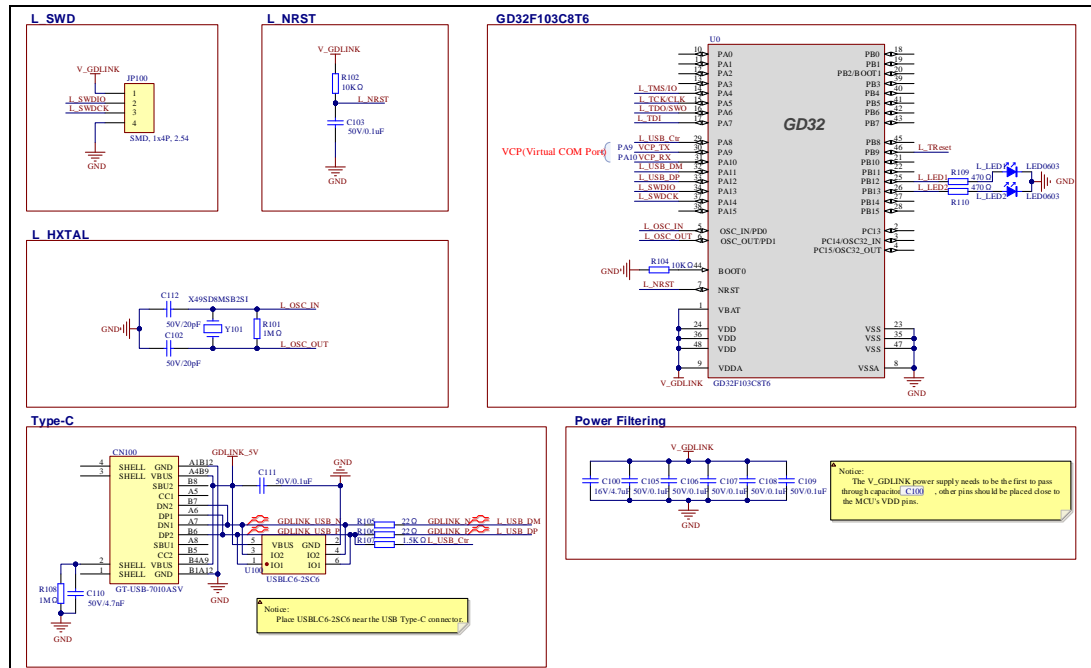
## 4.6. Extension

Figure 4-6. Schematic diagram of Extension



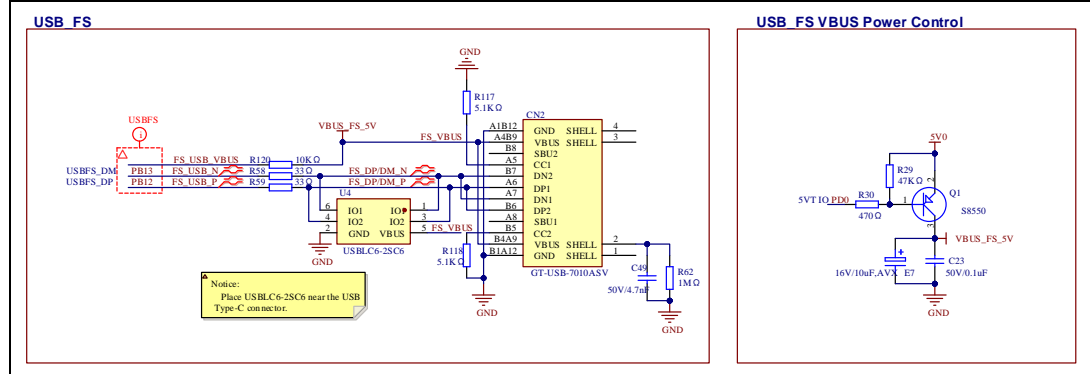
## 4.7. GD-Link

Figure 4-7. Schematic diagram of GD-Link



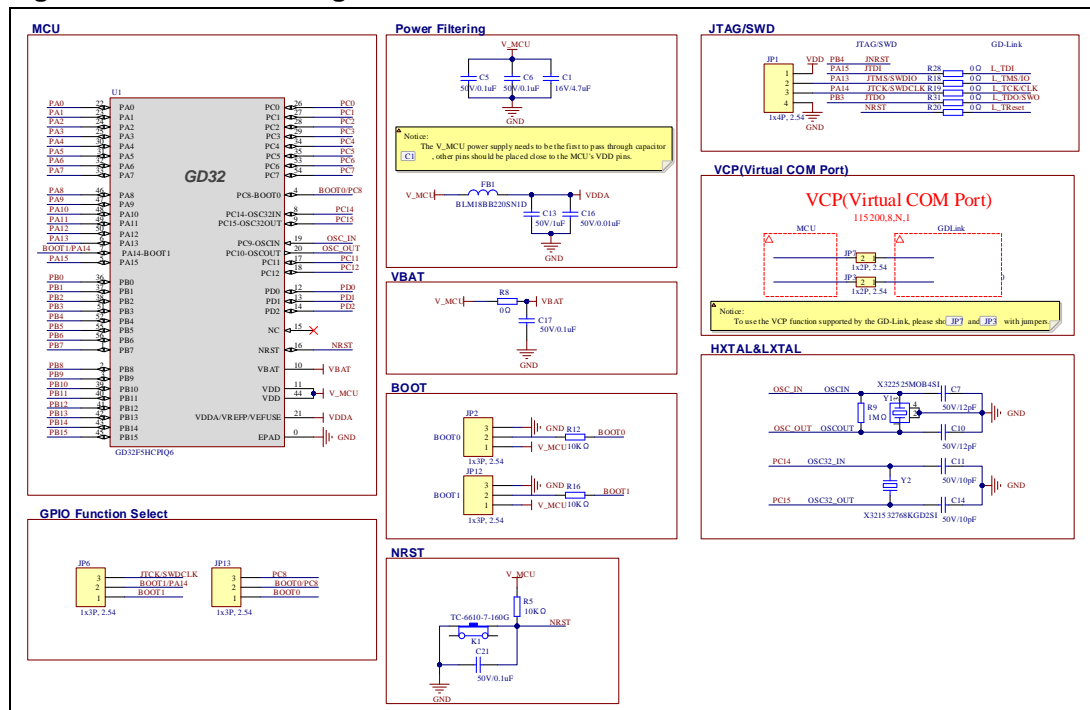
## 4.8. USB

Figure 4-8. Schematic diagram of USB



## 4.9. MCU

**Figure 4-9. Schematic diagram of MCU**



## 5. Routine use guide

### 5.1. GPIO\_Runing\_Led

#### 5.1.1. DEMO Purpose

This demo includes the following functions of GD32 MCU:

- Learn to use GPIO control the LED
- Learn to use SysTick to generate 1ms delay

GD32F5HCP-START-V1.0 board has four LEDs. The LED1, LED2, LED3 and LED4 are controlled by GPIO. This demo will show how to light the LEDs.

#### 5.1.2. DEMO Running Result

Download the program <01\_GPIO\_Running\_Led> to the START board, LED1, LED2, LED3 and LED4 turn on in sequence for 500ms and then turn off, after which the sequence repeats.

### 5.2. GPIO\_Key\_Polling\_mode

#### 5.2.1. DEMO Purpose

This demo includes the following functions of GD32 MCU:

- Learn to use GPIO control the LED and the KEY
- Learn to use SysTick to generate 1ms delay

GD32F5HCP-START-V1.0 board has four keys and four LEDs. The four keys are Reset key, Wakeup key, UserKey1 key and UserKey2 key. The LED1, LED2, LED3, LED4 are controlled by GPIO.

This demo will show how to use the UserKey2 key to control the LED2. When press down the UserKey2 key, it will check the input value of the IO port. If the value is 0 and will wait for 100ms. Check the input value of the IO port again. If the value still is 0, it indicates that the button is pressed successfully and toggle LED2.

#### 5.2.2. DEMO Running Result

Download the program <02\_GPIO\_Key\_Polling\_mode> to the EVAL board, press down the UserKey2 Key, LED2 will be turned on. Press down the Tamper Key again, LED2 will be turned off.

## 5.3. EXTI\_Key\_Interrupt\_mode

### 5.3.1. DEMO Purpose

This demo includes the following functions of GD32 MCU:

- Learn to use GPIO control the LED and the KEY
- Learn to use EXTI to generate external interrupt

GD32F5HCP-START-V1.0 board has four keys and four LEDs. The four keys are Reset key, Wakeup key, UserKey1 key and UserKey2 key. The LED1, LED2, LED3, LED4 are controlled by GPIO.

This demo will show how to use the EXTI interrupt line to control the LED2. When press down the UserKey2 Key, it will produce an interrupt. In the interrupt service function, the demo will toggle LED2.

### 5.3.2. DEMO Running Result

Download the program <03\_EXTI\_Key\_Interrupt\_mode> to the EVAL board, LED2 is turned on and off for test. When press down the UserKey2 Key, LED2 will be turned on. Press down the UserKey2 Key again, LED2 will be turned off.

## 5.4. USART\_Echo\_Interrupt\_mode

### 5.4.1. DEMO Purpose

This demo includes the following functions of GD32 MCU:

- Learn to use the USART transmit and receive interrupts to communicate with the serial terminal tool

### 5.4.2. DEMO Running Result

Download the program < 04\_USART\_Echo\_Interrupt\_mode > to the EVAL board, connect serial cable to USART. Firstly, all the LEDs are turned on and off for test. Then, the USART sends the tx\_buffer array (from 0x00 to 0xFF) to the serial terminal tool supporting hex format communication and waits for receiving data of BUFFER\_SIZE bytes from the serial terminal. The data MCU has received is stored in the rx\_buffer array. After that, compare tx\_buffer with rx\_buffer. If tx\_buffer is same with rx\_buffer, LED1, LED2, LED3, LED4 turns on. Otherwise, LED1, LED2, LED3, LED4 toggle together.

The output information via the serial port is as following.

```

00 01 02 03 04 05 06 07 08 09 0A 0B 0C 0D 0E 0F 10 11 12 13 14 15 16 17 18 19 1A 1B
1C 1D 1E 1F 20 21 22 23 24 25 26 27 28 29 2A 2B 2C 2D 2E 2F 30 31 32 33 34 35 36 37
38 39 3A 3B 3C 3D 3E 3F 40 41 42 43 44 45 46 47 48 49 4A 4B 4C 4D 4E 4F 50 51 52 53
54 55 56 57 58 59 5A 5B 5C 5D 5E 5F 60 61 62 63 64 65 66 67 68 69 6A 6B 6C 6D 6E 6F
70 71 72 73 74 75 76 77 78 79 7A 7B 7C 7D 7E 7F 80 81 82 83 84 85 86 87 88 89 8A 8B
8C 8D 8E 8F 90 91 92 93 94 95 96 97 98 99 9A 9B 9C 9D 9E 9F A0 A1 A2 A3 A4 A5 A6 A7
A8 A9 AA AB AC AD AE AF B0 B1 B2 B3 B4 B5 B6 B7 B8 B9 BA BB BC BD BE BF C0 C1 C2 C3
C4 C5 C6 C7 C8 C9 CA CB CC CD CE CF D0 D1 D2 D3 D4 D5 D6 D7 D8 D9 DA DB DC DD DE DF
E0 E1 E2 E3 E4 E5 E6 E7 E8 E9 EA EB EC ED EE EF F0 F1 F2 F3 F4 F5 F6 F7 F8 F9 FA FB
FC FD FE FF

```

## 5.5. TIMER\_Breath\_LED

### 5.5.1. DEMO Purpose

This demo includes the following functions of GD32 MCU:

- Learn to use Timer output PWM wave
- Learn to update channel value

### 5.5.2. DEMO Running Result

Use the DuPont line to connect the TIMERO\_CH0 (PA8) and LED (PB6). Then download the program <05\_TIMER\_Breath\_LED> to the EVAL board and run. PA8 should not be reused by other peripherals.

When the program is running, you can see LED lighting from dark to bright gradually and then gradually darken, ad infinitum, just like breathing as rhythm.

## 5.6. USB\_Device

### 5.6.1. HID\_Keyboard

#### DEMO purpose

This demo includes the following functions of GD32 MCU:

- Learn how to use the USBFS peripheral mode
- Learn how to implement USB HID(human interface device)

GD32F5HCP-START-V1.0 board has four keys and one USB\_FS interface. The four keys are Reset key, Wakeup key, Tamper key and User key. In this demo, the GD32F5HCP-START-V1.0 board is enumerated as a USB Keyboard, which uses the native PC Host HID driver, as shown below. The USB Keyboard uses three keys (wakeup key, tamper key and user key) to output three characters ('b', 'a' and 'c'). In addition, the demo also supports remote wakeup which is the ability of a USB device to bring a suspended bus back to the active condition, and the wakeup key is used as the remote wakeup source.



## DEMO running result

Download the program < 06\_USB\_FS\USB\_Device\_HID\_Keyboard > to the EVAL board and run. If you press the Wakeup key, will output 'b'. If you press the User key, will output 'c'. If you press the Tamper key, will output 'a'.

If you want to test USB remote wakeup function, you can do as follows:

- Manually switch PC to standby mode
- Wait for PC to fully enter the standby mode
- Push the Wakeup key
- If PC is ON, remote wakeup is OK, else failed.

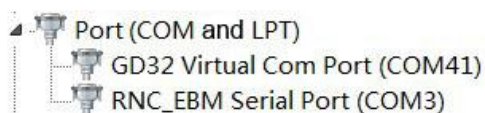
## 5.6.2. CDC\_ACM

### DEMO purpose

This demo includes the following functions of GD32 MCU:

- Learn how to use the USBFS peripheral
- Learn how to implement USBFS CDC device


GD32F5HCP-START-V1.0 board has one USBFS interface. In this demo, the GD32F5HCP-START-V1.0 board is enumerated as a USB virtual COM port, which was shown in device manager of PC as below. This demo makes the USB device look like a serial port, and loops back the contents of a text file over USB port. To run the demo, input a message using the PC's keyboard. Any data that shows in HyperTerminal is received from the device.



### DEMO running result

Download the program < 06\_USB\_FS\USB\_Device\_CDC\_ACM > to the EVAL board and run. When you input message through computer keyboard, the HyperTerminal will receive and shown the message. For example, when input "GigaDevice MCU", the HyperTerminal will get and show it as below.





GigaDevice MCU

## 5.7. USB\_Host

### 5.7.1. HID\_Host

#### DEMO purpose

This demo includes the following functions of GD32 MCU:

- Learn to use the USBFS as a HID host
- Learn the operation between the HID host and the mouse device
- Learn the operation between the HID host and the keyboard device

GD32F5HCP-START-V1.0 board integrates the USBFS module, and the module can be used as a USB device or a USB host. This demo mainly shows how to use the USBFS as a USB HID host to communicate with external USB HID device.

#### DEMO running result

Download the program <06\_USB\_FS\USB\_Host\_HID> to the EVAL board and run.

If a mouse has been attached, the user will see the information of mouse enumeration. First pressing the user key will see the inserted device is mouse, and then moving the mouse will show the position of mouse and the state of button in the screen.

If a keyboard has been attached, the user will see the information of keyboard enumeration. First pressing the user key will see the inserted device is keyboard, and then pressing the keyboard will show the state of the button in the screen.

### 5.7.2. MSC\_Host

#### DEMO purpose

This demo includes the following functions of GD32 MCU:

- Learn to use the USBFS as a MSC host
- Learn the operation between the MSC host and the Udisk

GD32F5HCP-START-V1.0 board integrates the USBFS module, and the module can be used as a USB device or a USB host. This demo mainly shows how to use the USBFS

as a USB MSC host to communicate with external Udisk.

### **DEMO running result**

Download the program <06\_USB\_FS\USB\_Host\_MSC > to the EVAL board and run.

If an Udisk has been attached, the user will see the information of Udisk enumeration. First pressing the user key will see the Udisk information, next pressing the tamper key will see the root content of the Udisk, then press the wakeup key will write file to the Udisk, finally the user will see information that the msc host demo is end.

## **5.8. Trustzone**

### **5.8.1. DEMO Purpose**

This demo includes the following functions of GD32 MCU:

- Learn to use MCU when TZEN = 1
- Learn to use SAU/IDAU to configure NSC and NS address area
- Learn to use option bytes to configure secure mark pages
- Learn to use code to enable Trustzone
- Learn to use TZPCU to configure non-secure SRAM area
- Learn to configure GPIO to non-secure
- Learn to use TZPCU to configure USART to secure
- Learn how secure code jump to non-secure code
- Learn how secure code call non-secure code function
- Learn how non-secure code call secure code function by non-secure callable function

### **5.8.2. DEMO Running Result**

Download the program < 07\_Trustzone> to the START board and run. LED1 and LED2 can light cycles. And HyperTerminal will print “secure code print: secure code toggle LED1.” and “non-secure code print: non-secure code toggle LED2.” cycles.

## 6. Revision history

Table 6-1. Revision history

Revision No.	Description	Date
1.0	Initial Release	Apr.15, 2026

## Important Notice

This document is the property of GigaDevice Semiconductor Inc. and its subsidiaries (the "Company"). This document, including any product of the Company described in this document (the "Product"), is owned by the Company according to the laws of the People's Republic of China and other applicable laws. The Company reserves all rights under such laws and no Intellectual Property Rights are transferred (either wholly or partially) or licensed by the Company (either expressly or impliedly) herein. The names and brands of third party referred thereto (if any) are the property of their respective owner and referred to for identification purposes only.

To the maximum extent permitted by applicable law, the Company makes no representations or warranties of any kind, express or implied, with regard to the merchantability and the fitness for a particular purpose of the Product, nor does the Company assume any liability arising out of the application or use of any Product. Any information provided in this document is provided only for reference purposes. It is the sole responsibility of the user of this document to determine whether the Product is suitable and fit for its applications and products planned, and properly design, program, and test the functionality and safety of its applications and products planned using the Product. The Product is designed, developed, and/or manufactured for ordinary business, industrial, personal, and/or household applications only, and the Product is not designed or intended for use in (i) safety critical applications such as weapons systems, nuclear facilities, atomic energy controller, combustion controller, aeronautic or aerospace applications, traffic signal instruments, pollution control or hazardous substance management; (ii) life-support systems, other medical equipment or systems (including life support equipment and surgical implants); (iii) automotive applications or environments, including but not limited to applications for active and passive safety of automobiles (regardless of front market or aftermarket), for example, EPS, braking, ADAS (camera/fusion), EMS, TCU, BMS, BSG, TPMS, Airbag, Suspension, DMS, ICMS, Domain, ESC, DCDC, e-clutch, advanced-lighting, etc.. Automobile herein means a vehicle propelled by a self-contained motor, engine or the like, such as, without limitation, cars, trucks, motorcycles, electric cars, and other transportation devices; and/or (iv) other uses where the failure of the device or the Product can reasonably be expected to result in personal injury, death, or severe property or environmental damage (collectively "Unintended Uses"). Customers shall take any and all actions to ensure the Product meets the applicable laws and regulations. The Company is not liable for, in whole or in part, and customers shall hereby release the Company as well as its suppliers and/or distributors from, any claim, damage, or other liability arising from or related to all Unintended Uses of the Product. Customers shall indemnify and hold the Company, and its officers, employees, subsidiaries, affiliates as well as its suppliers and/or distributors harmless from and against all claims, costs, damages, and other liabilities, including claims for personal injury or death, arising from or related to any Unintended Uses of the Product.

Information in this document is provided solely in connection with the Product. The Company reserves the right to make changes, corrections, modifications or improvements to this document and the Product described herein at any time without notice. The Company shall have no responsibility whatsoever for conflicts or incompatibilities arising from future changes to them. Information in this document supersedes and replaces information previously supplied in any prior versions of this document.