

**GigaDevice Semiconductor Inc.**

**GD32A490Z-START**

**Arm<sup>®</sup> Cortex<sup>®</sup>-M4 32-bit MCU**

## **User Guide**

Revision 1.1

(May. 2025)

# 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. GD-Link .....	10
4.7. MCU .....	11
4.8. Arduino.....	12
<b>5. ROUTINE USE GUIDE .....</b>	<b>13</b>
5.1. GPIO_Running_LED .....	13
5.1.1. DEMO purpose .....	13
5.1.2. DEMO running result .....	13
5.2. GPIO_Key_Polling_mode.....	13
5.2.1. DEMO purpose .....	13
5.2.2. DEMO running result .....	13
5.3. EXTI_Key_Interrupt_mode .....	14
5.3.1. DEMO purpose .....	14
5.3.2. DEMO running result .....	14
5.4. USART_HyperTerminal_Interrupt.....	14
5.4.1. DEMO purpose .....	14
5.4.2. DEMO running result .....	14
5.5. TIMER_Key_EXTI .....	15
5.5.1. DEMO purpose .....	15
5.5.2. DEMO running result .....	15
5.6. TRIGSEL_TIMER_Trigger_EXTI .....	15
5.6.1. DEMO purpose .....	15
5.6.2. DEMO running result .....	16
<b>6. 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 4-7. Schematic diagram of GD-Link.....	10
Figure 4-8. Schematic diagram of MCU .....	11
Figure 4-9. Schematic diagram of Arduino .....	12

# List of Tables

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

## 1. Summary

GD32A490Z-START uses GD32A490ZKT7 as the main controller. It uses GD-Link Mini USB interface to supply 5V power. Reset, Boot, Key, LED, USART to USB, Arduino are also included. For more details, please refer to GD32A490Z-START-V1.0 schematic.

## 2. Function Pin Assign

Table 2-1. Function pin assignment

Function	Pin	Description
LED	PE4	LED1
	PE5	LED2
	PE6	LED3
	PC13	LED4
RESET		K1-Reset
KEY	PA0	K2-USER
USART	PD8	RS232_TX
	PD9	RS232_RX

### 3. Getting started

The EVAL board uses GD-Link Mini USB connector to get power DC +5V, which is the hardware system normal work voltage. A 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 the power supply is OK.

There are Keil version and IAR version of all projects. Keil version of the projects are created based on Keil MDK-ARM 4.74 uVision4. IAR version of the projects are created based on IAR Embedded Workbench for ARM 7.40.2. During use, the following points should be noted:

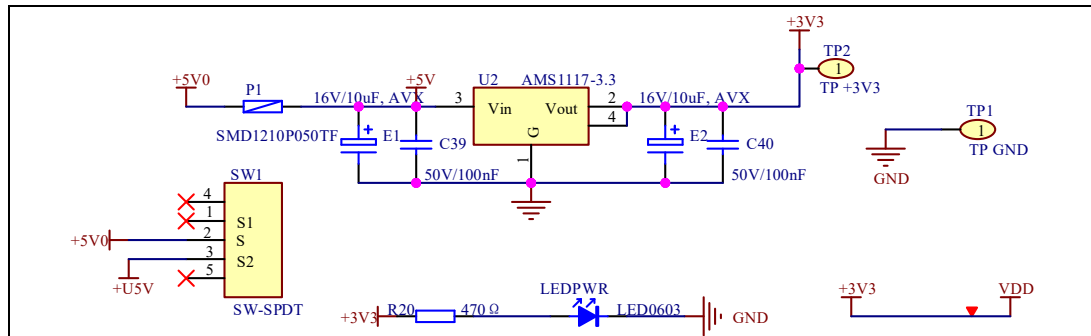
1. If you use Keil uVision4 to open the project. In order to solve the "Device Missing (s)" problem, you can install GigaDevice.GD32A490\_DFP.1.0.0.pack.
2. If you use IAR to open the project, install IAR\_GD32A490\_ADDON\_1.0.0.exe to load the associated files.



## 4. Hardware layout overview

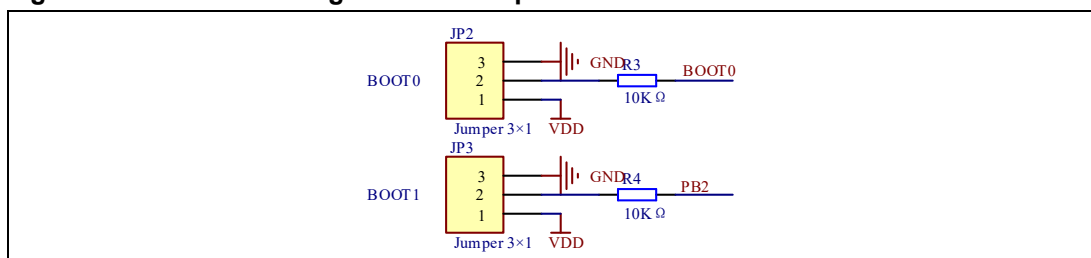
### 4.1. Power supply

Figure 4-1. Schematic diagram of power supply



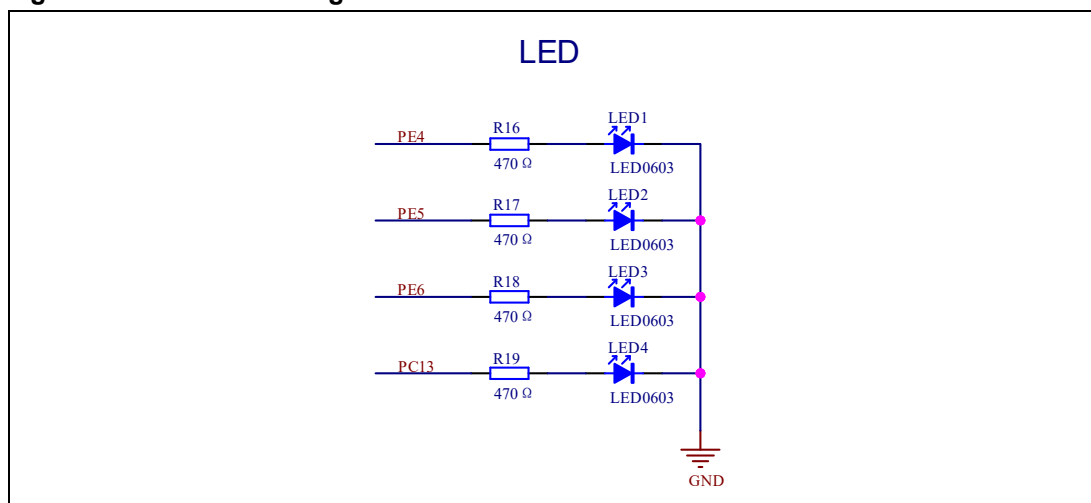
### 4.2. Boot option

Figure 4-2. Schematic diagram of boot option



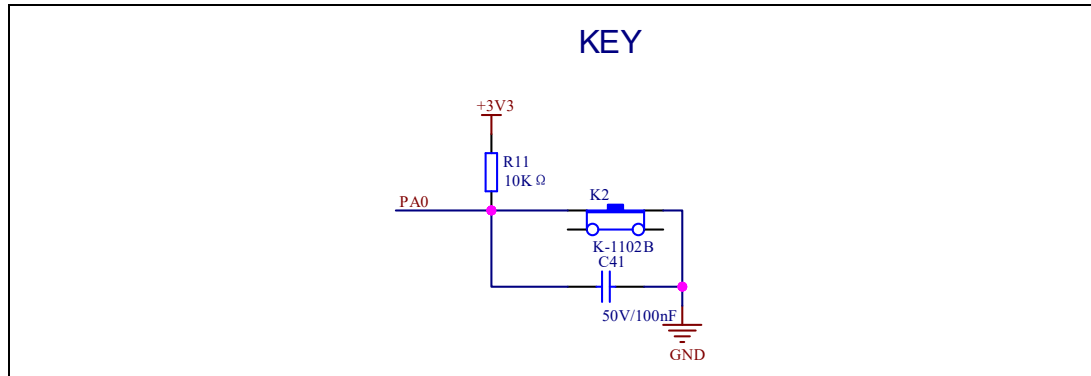
### 4.3. LED

Figure 4-3. Schematic diagram of LED function



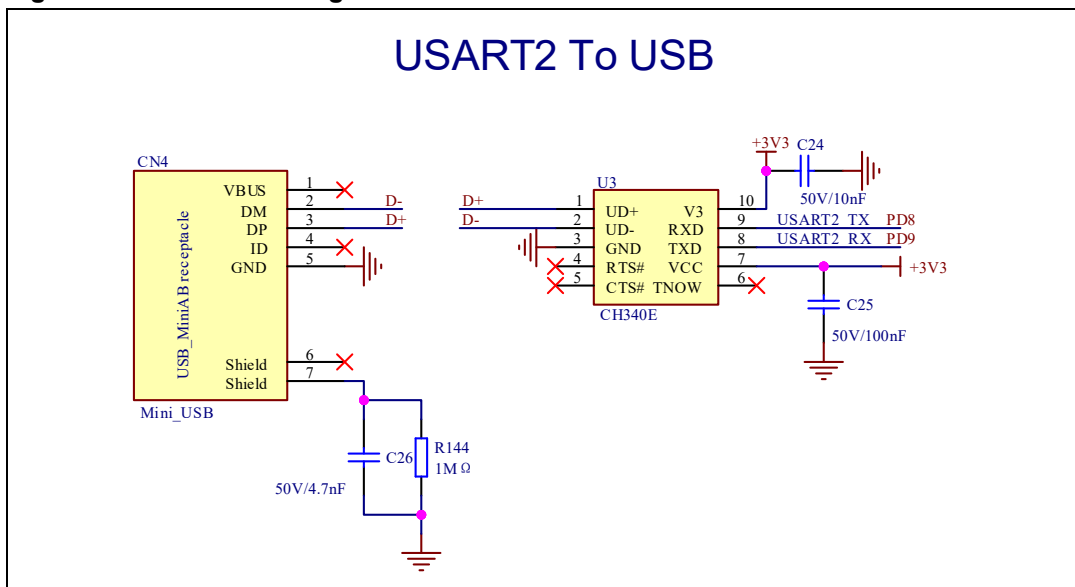
## 4.4. KEY

Figure 4-4. Schematic diagram of Key function



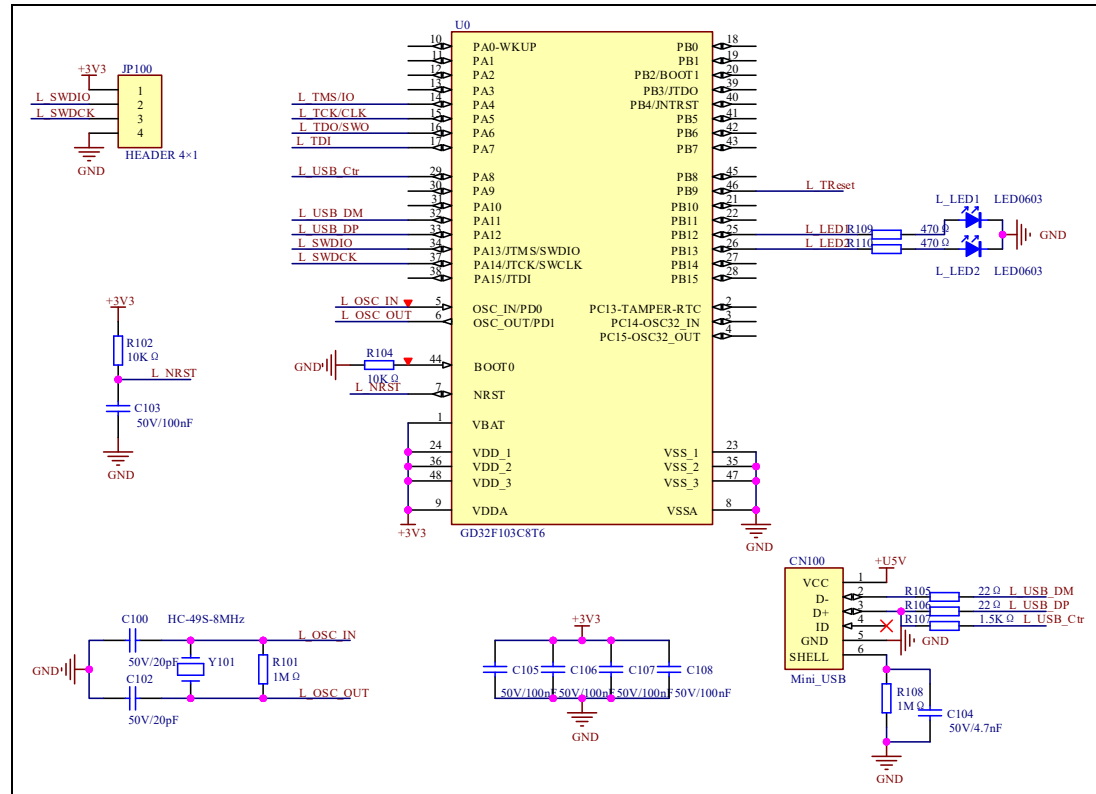
## 4.5. USART

Figure 4-5. Schematic diagram of USART



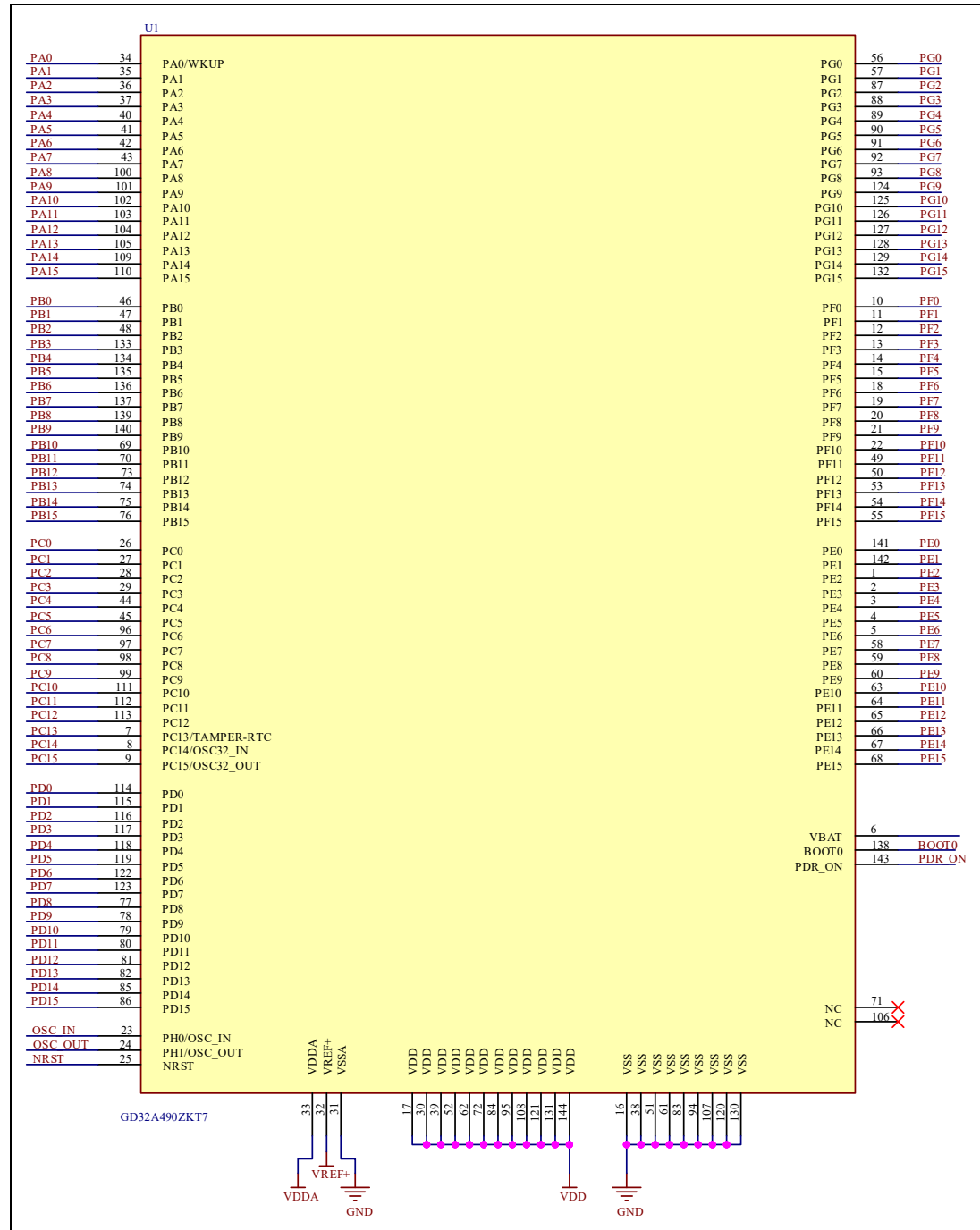
## 4.6. GD-Link

Figure 4-6. Schematic diagram of GD-Link



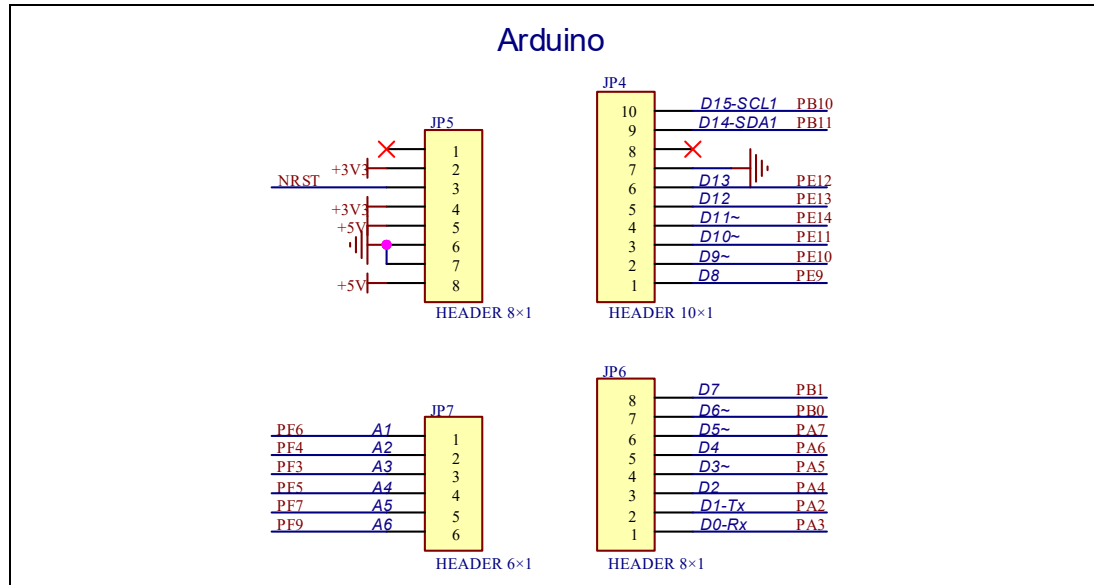
## 4.7. MCU

Figure 4-7. Schematic diagram of MCU



## 4.8. Arduino

Figure 4-8. Schematic diagram of Arduino



## 5. Routine use guide

### 5.1. GPIO\_Running\_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

GD32A490Z-START board has 2 keys and 4 LEDs. The 2 keys are Reset key and User key. The LEDs 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 EVAL board, LED1, LED2, LED3 and LED4 will turn on in sequence with interval of 1s, and repeat the process.

### 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

GD32A490Z-START board has 2 keys and 4 LEDs. The 2 keys are Reset key and User key. The LEDs are controlled by GPIO.

This demo will show how to use the User key to control the LED2. When press down the User key, it will check the input value of the IO port. If the value is 1 and will wait for 100ms. Check the input value of the IO port again. If the value still is 1, 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 START board, press down the User key, LED2 will be turned on. Press down the User 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.

GD32A490Z-START board has 2 keys and 4 LEDs. The 2 keys are Reset key and Wakeup key. The LEDs are controlled by GPIO.

This demo will show how to use the EXTI interrupt line to control the LED2. When press down the Wakeup 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 Wakeup key, LED2 will be turned on. Press down the Wakeup key again, LED2 will be turned off.

## 5.4. USART\_HyperTerminal\_Interrupt

### 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\_HyperTerminal\_Interrupt > 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 is ON. Otherwise, LED1 toggles.

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\_Key\_EXTI

### 5.5.1. 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
- Learn to use TIMER to generate PWM

The board has 2 keys and 4 LEDs. The 2 keys are Reset key and User key. The LEDs are controlled by GPIO.

This demo will show how to use the TIMER PWM to trigger EXTI interrupt to toggle the state of LED1 and EXTI interrupt line to control the LED1. When press down the User Key, it will produce an interrupt. In the interrupt service function, the demo will toggle LED1.

### 5.5.2. DEMO running result

Download the program <05\_TIMER\_Key\_EXTI> to the START board, the LED1 is flashed once for test, press down the Wakeup Key, LED1 will be turned on. Press down the Wakeup Key again, LED1 will be turned off. Connect PA7 (TIMER1\_CH1) and PA4 with DuPont line. The LED1 will be toggled every 500ms.

## 5.6. TRIGSEL\_TIMER\_Trigger\_EXTI

### 5.6.1. DEMO purpose

This demo includes the following functions of GD32 MCU:

- Learn to use TIMER output PWM wave
- Learn to use EXTI generate interrupt
- Learn to use TRIGSEL select trigger source and trigger peripheral



### 5.6.2. DEMO running result

Download the program < 06\_TRIGSEL\_TIMER\_Trigger\_EXTI > to the EVAL board, LED1 is turned on and off for test.

When the program is running, the EXTI10 interrupt is triggered by TRIGSEL output every 200ms, and LED1 state is toggled by the interrupt.

## 5.7. USB\_MSC\_Device

### 5.7.1. DEMO purpose

This demo includes the following functions of GD32 MCU:

- Learn how to use the USBFS
- Learn how to implement USB MSC (mass storage) device

This demo mainly implements a U disk. U disk is currently very widely used removable MSC devices. MSC, the Mass Storage Device Class, is a transport protocol between a computer and mobile devices, which allow a universal serial bus (USB) equipment to access a host computing device, file transfer between them, mainly including mobile hard disk, mobile U disk drive, etc. The MSC device must have a storage medium, and this demo uses the MCU's internal SRAM as the storage medium. For more details of the MSC protocol please refer to the MSC protocol standard.

MSC device will use a variety of transport protocols and command formats for communication, so it need to choose the appropriate protocol and command format in the realization of the application. This demo selects the BOT (bulk only transport) protocol and the required SCSI (small computer interface) command, and is compatible with a wide variety of Window operating systems. Specific BOT protocol and SCSI command specification please refer to the standard of their agreement.

### 5.7.2. DEMO running result

Download the program <05\_USB\_MSC\_Device> to the START board and run. When the start board is connected to the PC, you will find a USB large capacity storage device is in the universal serial bus controller, and there is 1 more disk drives in the equipment manager of PC.

Then, after opening the resource manager, you will see more of the 1 disk, as shown in the following diagram:



At this point, the write/read/formatting operation can be performed as the other mobile devices.

## 5.8. USB\_MSC\_Host

### 5.8.1. 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 U-disk

GD32A490Z-START-V1.0 board integrates USBFS module, and the USBFS module can be used as USB device, USB host or OTG device. This demo mainly shows how to use the USBFS as a USB MSC host to communicate with external U-disk.

### 5.8.2. DEMO running result

Insert the OTG cable to USB port. Then, download the program <06\_USB\_MSC\_Host> to the START board and run.

If a U-disk has been attached, the user will see the information of U-disk enumeration on the serial Assistant. Pressing the User key the user will see the root content of the U-disk, then the MSC Host write file to the U-disk, and the user will see information that the MSC host demo is end.

```
++++USB host library started++++
> Reset the USB device.
> Full speed device detected.
> Device Attached.
VID: 14CDh
PID: 2536h
> Mass storage device connected.
Manufacturer: Generic USB Device
Product: Mass Storage Device
Serial Number: 2536201609180000
> Enumeration completed.
>To see the disk information:
>Press User Key...
> File System initialized.
> Disk capacity: 3909090816d Bytes.
> Exploring disk flash ...
|_ _RECYCLER
|_ _GD32.TXT
|_ _System Volume Information
|_ _RecognizeUSBFlugs
|_ _format_err.wav
|_ _GD32F4xx_AddOn
|   |_ _IAR_GD32F4xx_ADDON.3.0.1.exe
|   |_ _IAR_GD32F4xx_ADDON.3.0.2.exe
> Writing File to disk flash ...
> GD32.TXT created in the disk.
> The MSC host demo is end.
```

## 6. Revision history

Table 6-1. Revision history

Revision No.	Description	Date
1.0	Initial Release	Jul.31, 2023
1.1	Update the Important Notice.	May.13, 2025

## 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, aeronautic or aerospace applications, pollution control or hazardous substance management; (ii) life-support systems, other medical equipment or systems (including life support equipment and surgical implants); and/or (iii) 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.

While the Company has implemented advanced security features, the Product may be subject to unidentified vulnerabilities. Customers are responsible for the design and operation of their applications and products to reduce the effect of these vulnerabilities on Customer's applications and products, and to the maximum extent permitted by applicable law, the Company accepts no liability for any vulnerability that is discovered. Customers should implement appropriate design and operating safeguards to minimize the risks associated with their applications and products.

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.