

**GigaDevice Semiconductor Inc.**

**GD32C231C-EVAL**

**Arm<sup>®</sup> Cortex<sup>®</sup>-M23 32-bit MCU**

## **User Guide**

Revision 1.0

(Jun. 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 .....	9
4.4. KEY .....	9
4.5. ADC .....	9
4.6. CMP .....	10
4.7. USART .....	10
4.8. I2C.....	10
4.9. I2S.....	11
4.10. SPI .....	11
4.11. SLCD .....	11
4.12. Type-C.....	12
4.13. Extension.....	12
4.14. GD-Link.....	13
4.15. MCU.....	13
<b>5. Routine use guide.....</b>	<b>15</b>
<b>5.1. GPIO_Running_LED .....</b>	<b>15</b>
5.1.1. DEMO purpose .....	15
5.1.2. DEMO running result .....	15
<b>5.2. GPIO_Key_Polling_mode.....</b>	<b>15</b>
5.2.1. DEMO purpose .....	15
5.2.2. DEMO running result .....	15
<b>5.3. EXTI_Key_Interrupt_mode .....</b>	<b>16</b>
5.3.1. DEMO purpose .....	16
5.3.2. DEMO running result .....	16
<b>5.4. USART_Printf.....</b>	<b>16</b>
5.4.1. DEMO purpose .....	16
5.4.2. DEMO running result .....	16
<b>5.5. USART_HyperTerminal_Interrupt.....</b>	<b>17</b>
5.5.1. DEMO purpose .....	17
5.5.2. DEMO running result .....	17

<b>5.6. USART_DMA</b>	<b>17</b>
5.6.1. DEMO purpose	17
5.6.2. DEMO running result	17
<b>5.7. ADC_Temperature_Vrefint</b>	<b>18</b>
5.7.1. DEMO purpose	18
5.7.2. DEMO running result	18
<b>5.8. Comparator_Obtain_Brightness</b>	<b>19</b>
5.8.1. DEMO purpose	19
5.8.2. DEMO running result	19
<b>5.9. I2C_EEPROM</b>	<b>19</b>
5.9.1. DEMO purpose	19
5.9.2. DEMO running result	19
<b>5.10. SPI_SPI_FLASH</b>	<b>20</b>
5.10.1. DEMO purpose	20
5.10.2. DEMO running result	20
<b>5.11. I2S_Audio_Player</b>	<b>22</b>
5.11.1. DEMO purpose	22
5.11.2. DEMO running result	22
<b>5.12. RCU_Clock_Out</b>	<b>22</b>
5.12.1. DEMO purpose	22
5.12.2. DEMO running result	22
<b>5.13. PMU_Sleep_Wakeup</b>	<b>23</b>
5.13.1. DEMO purpose	23
5.13.2. DEMO running result	23
<b>5.14. RTC_Calendar</b>	<b>23</b>
5.14.1. DEMO purpose	23
5.14.2. DEMO running result	23
<b>5.15. TIMER_Breath_LED</b>	<b>24</b>
5.15.1. DEMO purpose	24
5.15.2. DEMO running result	24
<b>5.16. SPI_LCD</b>	<b>24</b>
5.16.1. DEMO purpose	24
5.16.2. DEMO running result	24
<b>6. Revision history</b>	<b>26</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 .....	9
Figure 4-4. Schematic diagram of Key function .....	9
Figure 4-5. Schematic diagram of ADC .....	9
Figure 4-7. Schematic diagram of CMP .....	10
Figure 4-9. Schematic diagram of USART .....	10
Figure 4-10. Schematic diagram of I2C .....	10
Figure 4-11. Schematic diagram of I2S.....	11
Figure 4-12. Schematic diagram of SPI .....	11
Figure 4-13. Schematic diagram of SLCD .....	11
Figure 4-14. Schematic diagram of Type-C.....	12
Figure 4-15. Schematic diagram of Extension.....	12
Figure 4-16. Schematic diagram of GD-Link.....	13
Figure 4-17. Schematic diagram of MCU.....	13

# List of Tables

Table 2-1. Function pin assignment.....	6
Table 6-1. Revision history .....	26

## 1. Summary

GD32C231C-EVAL uses GD32C231C8T6 as the main controller. It uses GD-Link Type-C interface to supply 5V power. Reset, Boot, Button key, LED, I2S, I2C-EEPROM, SLCD, QSPI-Flash, Type-C interface are also included. For more details, please refer to GD32C231C-EVAL\_Rev1.0 schematic.

## 2. Function Pin Assign

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

Function	Pin	Description
LED	PA15	LED1
	PD1	LED2
	PD3	LED3
	PB4	LED4
RESET	NRST	Reset
KEY	PA0	K2(Wakeup)
	PA4	K3(User)
ADC	PA2	ADC_IN2
USART	PB7	USART0_TX
	PB6	USART0_RX
I2C	PA11	I2C1_SCL
	PA12	I2C1_SDA
I2S	PA7	I2S1_SD
	PA5	I2S1_CK
	PB0	I2S1_WS
	PA6	I2S1_MCK
SPI Flash	PB11	SPI1_FLASH_CS
	PB13	SPI1_SCK
	PB14	SPI1_MISO
	PB15	SPI1_MOSI
SLCD	PA8	SPILCD_CS
	PC6	SPILCD_RESET
	PC7	SPILCD_D/C
	PB13	SPI1_SCK
	PB14	SPI1_MISO
	PB15	SPI1_MOSI
CMP	PA1	CMP0_IP

### 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 5.35 uVision5. IAR version of the projects are created based on IAR Embedded Workbench for ARM 8.32.1. During use, the following points should be noted:

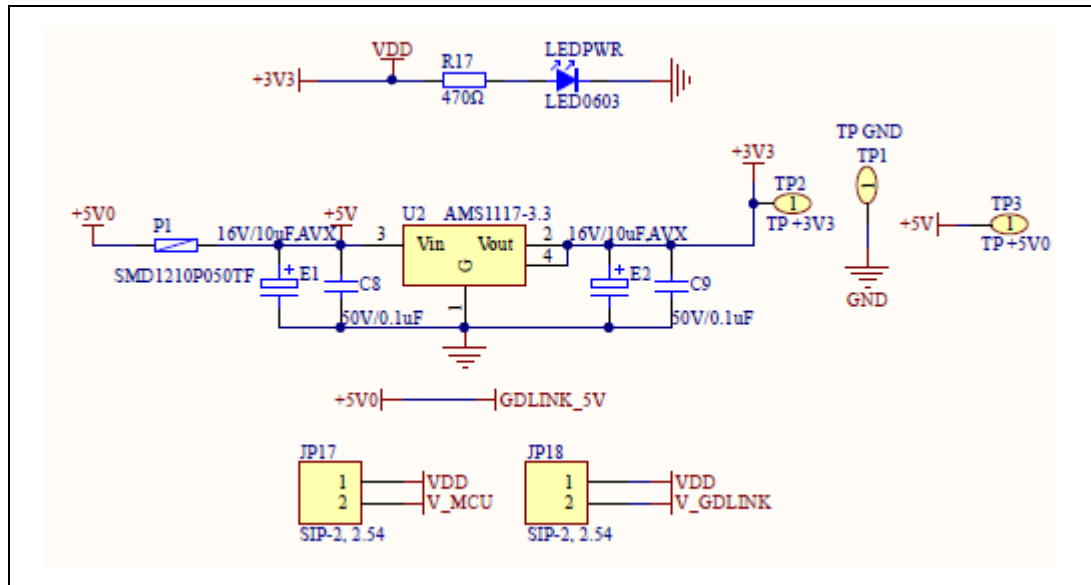
1. If you use Keil uVision5 to open the project. In order to solve the "Device Missing (s)" problem, you can install GigaDevice.GD32C2x1\_DFP.1.0.0.
2. If you use IAR to open the project, install IAR\_GD32L23x\_ADDON.1.0.0.exe to load the associated files.



## 4. Hardware layout overview

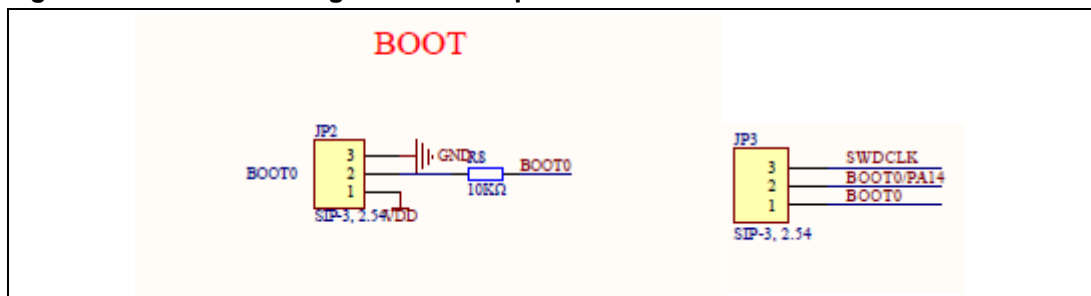
#### 4.1. Power supply

**Figure 4-1. Schematic diagram of power supply**



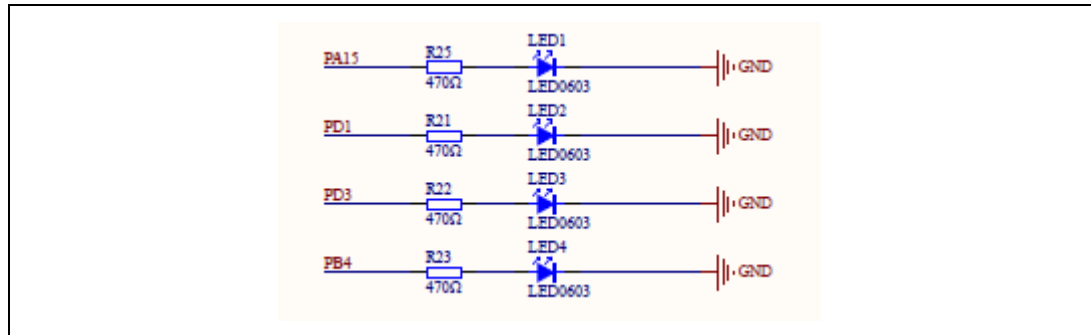
## 4.2. Boot option

**Figure 4-2. Schematic diagram of boot option**



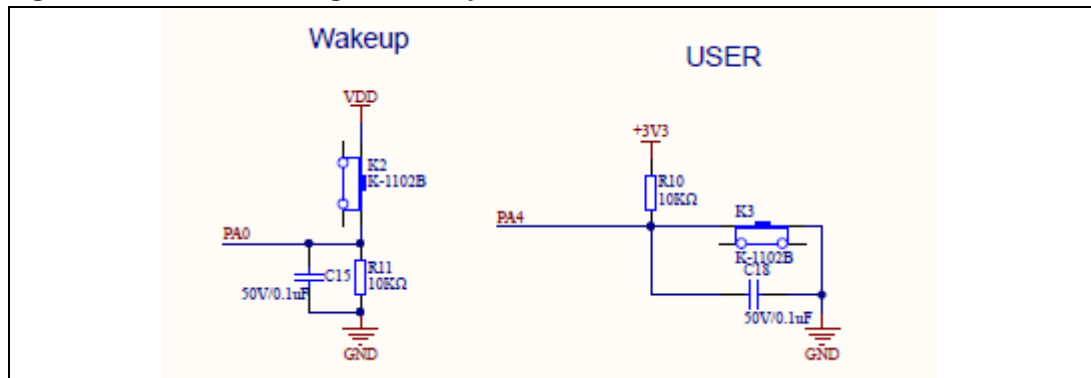
### 4.3. LED

Figure 4-3. Schematic diagram of LED function



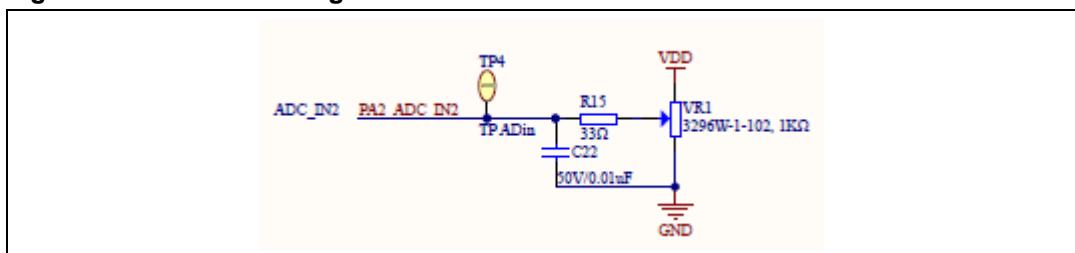
### 4.4. KEY

Figure 4-4. Schematic diagram of Key function



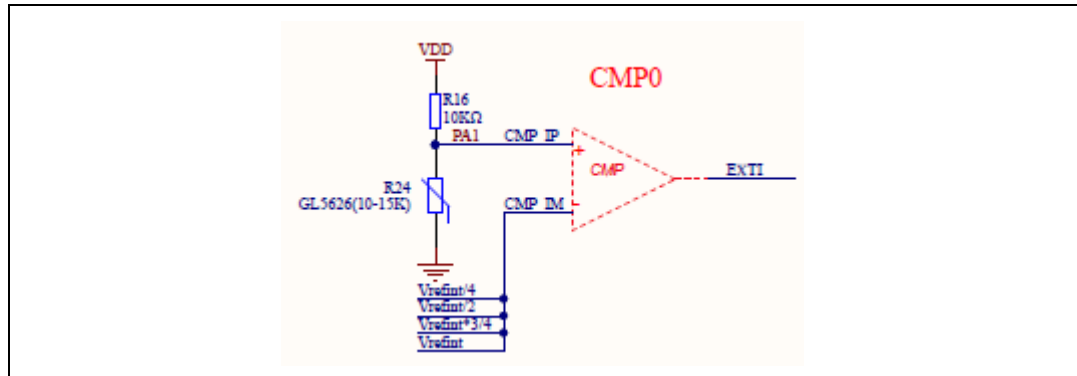
### 4.5. ADC

Figure 4-5. Schematic diagram of ADC



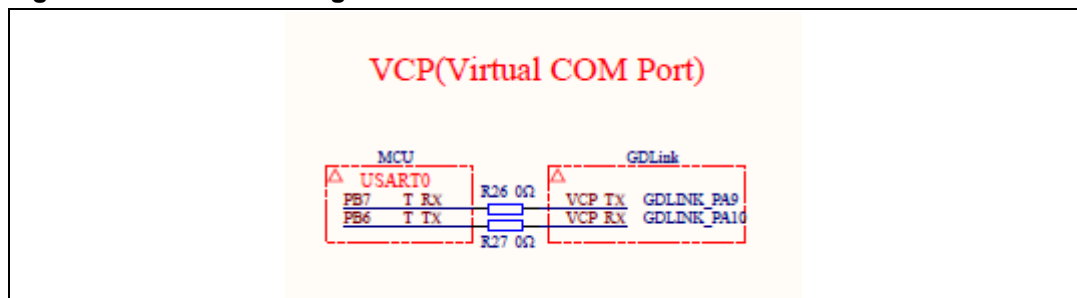
## 4.6. CMP

Figure 4-6. Schematic diagram of CMP



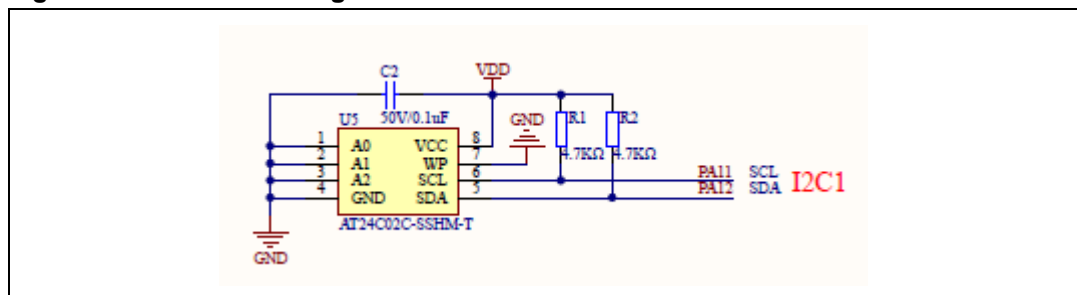
## 4.7. USART

Figure 4-7. Schematic diagram of USART



## 4.8. I2C

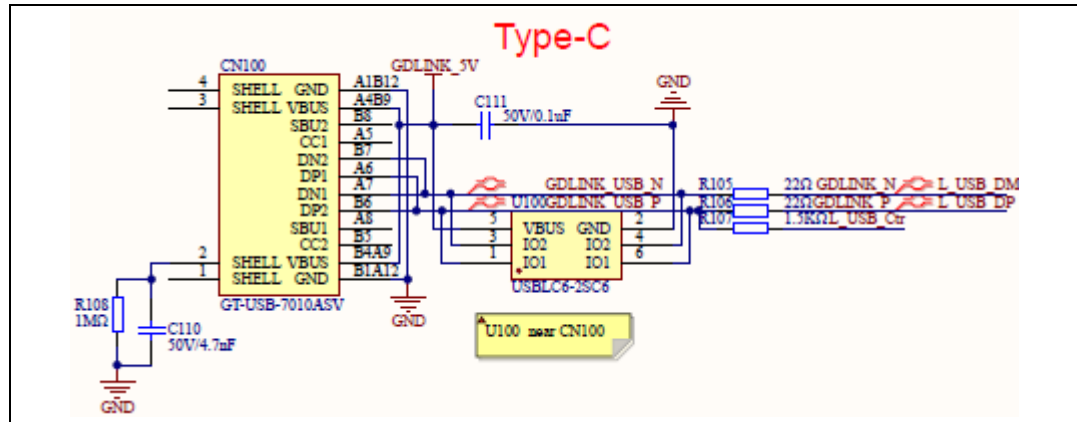
Figure 4-8. Schematic diagram of I2C





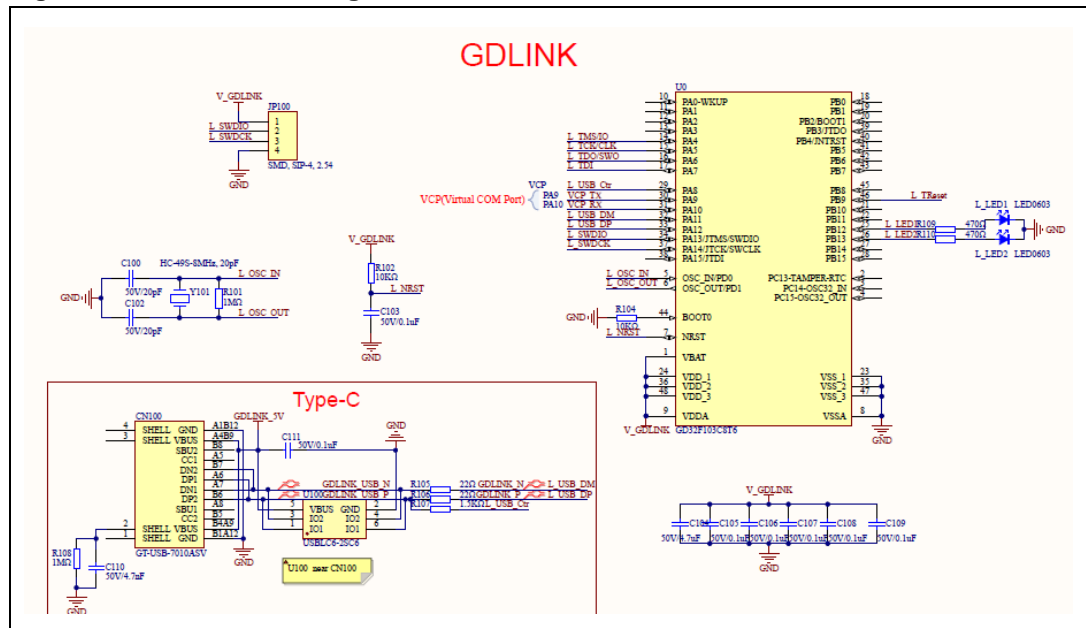
## 4.12. Type-C

Figure 4-12. Schematic diagram of Type-C



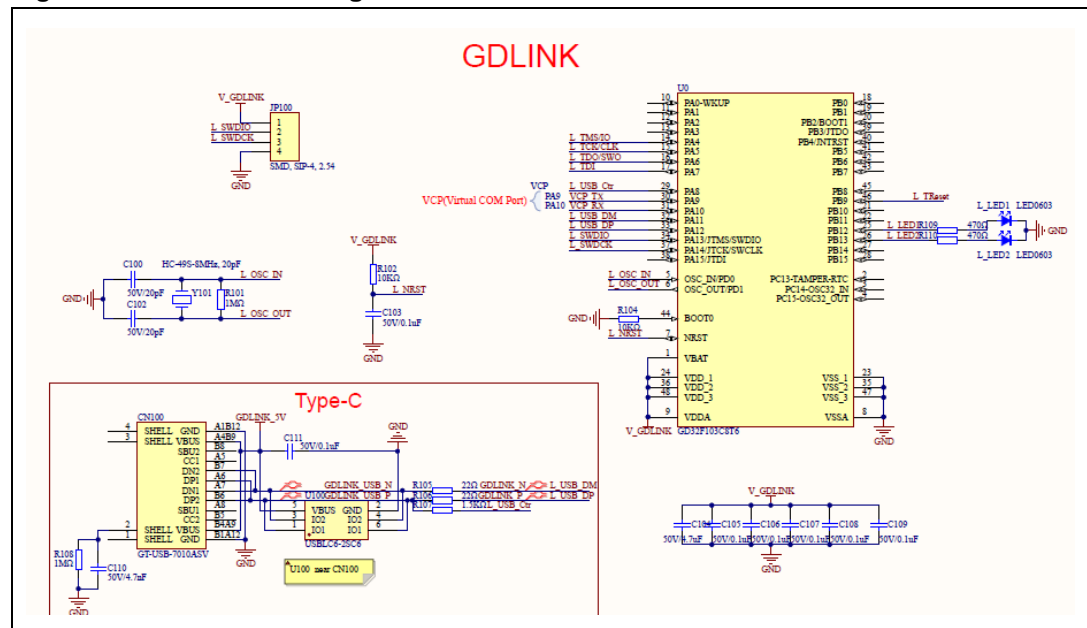
## 4.13. Extension

Figure 4-13. Schematic diagram of Extension



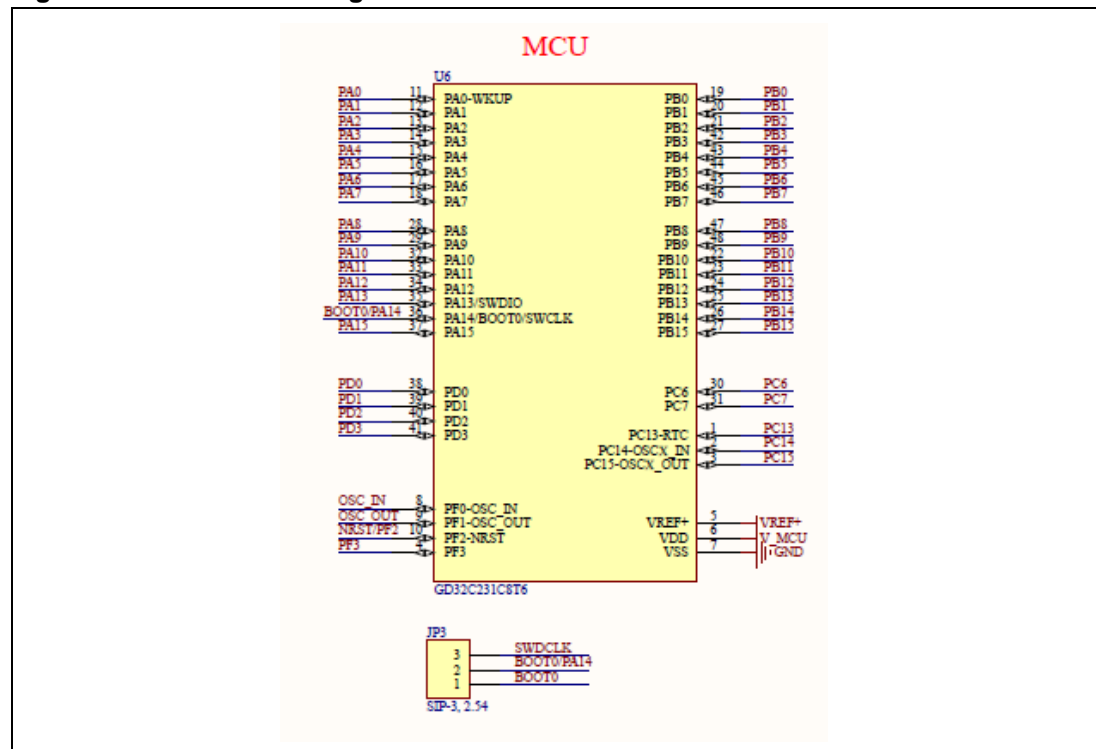
#### 4.14. GD-Link

**Figure 4-14. Schematic diagram of GD-Link**

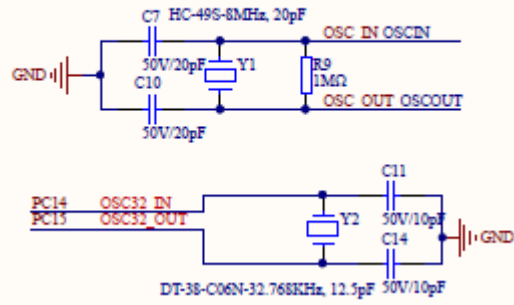


#### 4.15. MCU

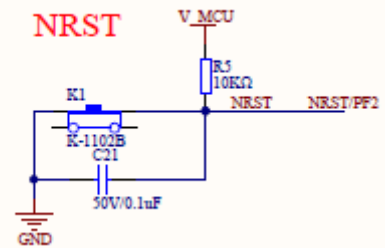
**Figure 4-15. Schematic diagram of MCU**



## HXTAL&LXTAL



## NRST



## 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

GD32C231C-EVAL board has three user keys and four LEDs. The keys are User key, Wakeup key and Reset 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, four LEDs can light cycles.

### 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

GD32C231C-EVAL board has three user keys and four LEDs. The keys are User key, Wakeup key and Reset key. The LEDs are controlled by GPIO.

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

GD32C231C-EVAL board has two user keys and four LEDs. The keys are User key and Wakeup key. The LED1, LED2, LED3 and LED4 are controlled by GPIO.

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

## 5.4. USART\_Printf

### 5.4.1. DEMO purpose

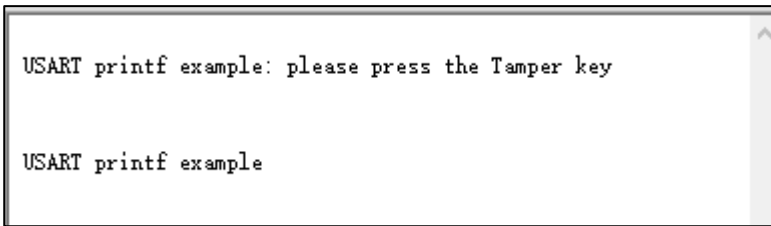
This demo includes the following functions of GD32 MCU:

- Learn to use GPIO control the LED
- Learn to retarget the C library printf function to the USART

### 5.4.2. DEMO running result

Download the program < 04\_USART\_Printf > to the EVAL board, connect serial cable to USART. Firstly, all the LEDs flash 2 times for test. Then, this implementation outputs "USART printf example: please press the Tamper key" on the HyperTerminal using USART. Press the Tamper key, the serial port will output "USART printf example".

The output information via the HyperTerminal is as following:



```
USART printf example: please press the Tamper key

USART printf example
```

## 5.5. USART\_HyperTerminal\_Interrupt

### 5.5.1. DEMO purpose

This demo includes the following functions of GD32 MCU:

- Learn to use the USART transmit and receive interrupts to communicate with the HyperTerminal.

### 5.5.2. DEMO running result

Download the program <05\_USART\_HyperTerminal\_Interrupt> to the EVAL board and 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 hyperterminal and waits for receiving data from the hyperterminal that you must send. The string that you have sent is stored in the rx\_buffer array. The receive buffer have a BUFFER\_SIZE bytes as maximum. After that, compare tx\_buffer with rx\_buffer. If tx\_buffer is same with rx\_buffer, LED1, LED2 flash by turns. Otherwise, LED1, LED2 toggle together.

The output information via the HyperTerminal 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
```

## 5.6. USART\_DMA

### 5.6.1. DEMO purpose

This demo includes the following functions of GD32 MCU:

- Learn to use the USART transmit and receive data using DMA.

### 5.6.2. DEMO running result

Download the program <06\_USART\_DMA> to the EVAL board and connect serial cable to USART. Firstly, the USART sends "USART DMA interrupt receive and transmit example, please input 10 bytes:" to hyperterminal and waits for receiving 10 bytes data from the hyperterminal that you must send. After MCU receives the data, the USART will continue to

output the received data to the hyper terminal.

The output information via the HyperTerminal is as following:

```
USART DMA interrupt receive and transmit example, please
input 10 bytes:

abcdefghijkl
```

## 5.7. ADC\_Temperature\_Vrefint

### 5.7.1. DEMO purpose

This demo includes the following functions of GD32 MCU:

- Learn to use the ADC to convert analog signal to digital data
- Learn to get the value of inner channel 13 (temperature sensor channel) and channel 14 (Vrefint channel)

### 5.7.2. DEMO running result

Download the program <07\_ADC\_Temperature\_Vrefint> to the GD32C231C-EVAL board. Connect serial cable to USART, open the HyperTerminal.

When the program is running, HyperTerminal display the value of temperature and internal voltage reference.

```
the Temperature data is 31 degrees Celsius
the Reference voltage data is 1.171V

the Temperature data is 31 degrees Celsius
the Reference voltage data is 1.170V

the Temperature data is 31 degrees Celsius
the Reference voltage data is 1.170V

the Temperature data is 31 degrees Celsius
the Reference voltage data is 1.171V

the Temperature data is 31 degrees Celsius
the Reference voltage data is 1.171V

the Temperature data is 31 degrees Celsius
the Reference voltage data is 1.171V
```

## 5.8. Comparator\_Obtain\_Brightness

### 5.8.1. DEMO purpose

This Demo includes the following functions of GD32 MCU:

- Learn to use comparator output compare result

The comparator has two inputs, in this demo, one input is PA1, and the other one is the reference voltage. Compare the two input voltages, the output is a high or low level, and the LED1 will performs the corresponding action.

### 5.8.2. DEMO running result

Download the program <08\_Comparator\_Obtain\_Brightness> to the EVAL board, comparing two input voltage, if output level is high, LED1 is on, otherwise LED1 is off.

## 5.9. I2C\_EEPROM

### 5.9.1. DEMO purpose

This demo includes the following functions of GD32 MCU:

- Learn to use the master transmitting mode of I2C module
- Learn to use the master receiving mode of I2C module
- Learn to read and write the EEPROM with I2C interface

### 5.9.2. DEMO running result

Download the program <09\_I2C\_EEPROM> to the EVAL board and run. Connect serial cable to VCP, open the HyperTerminal to show the print message.

Firstly, the data of 256 bytes will be written to the EEPROM from the address 0x00 and printed by the serial port. Then, reading the EEPROM from address 0x00 for 256 bytes and the result will be printed. Finally, compare the data that were written to the EEPROM and the data that were read from the EEPROM. If they are the same, the serial port will output "I2C-AT24C02 test passed!" and the two LEDs lights flashing, otherwise the serial port will output "Err:data read and write aren't matching." and all the two LEDs light.

The output information via the serial port is as following.

```
I2C-24C02 configured...

The I2C is hardware interface
The speed is 400K
AT24C02 writing...
0x00 0x01 0x02 0x03 0x04 0x05 0x06 0x07 0x08 0x09 0x0A 0x0B 0x0C 0x0D 0x0E 0x0F
0x10 0x11 0x12 0x13 0x14 0x15 0x16 0x17 0x18 0x19 0x1A 0x1B 0x1C 0x1D 0x1E 0x1F
0x20 0x21 0x22 0x23 0x24 0x25 0x26 0x27 0x28 0x29 0x2A 0x2B 0x2C 0x2D 0x2E 0x2F
0x30 0x31 0x32 0x33 0x34 0x35 0x36 0x37 0x38 0x39 0x3A 0x3B 0x3C 0x3D 0x3E 0x3F
0x40 0x41 0x42 0x43 0x44 0x45 0x46 0x47 0x48 0x49 0x4A 0x4B 0x4C 0x4D 0x4E 0x4F
0x50 0x51 0x52 0x53 0x54 0x55 0x56 0x57 0x58 0x59 0x5A 0x5B 0x5C 0x5D 0x5E 0x5F
0x60 0x61 0x62 0x63 0x64 0x65 0x66 0x67 0x68 0x69 0x6A 0x6B 0x6C 0x6D 0x6E 0x6F
0x70 0x71 0x72 0x73 0x74 0x75 0x76 0x77 0x78 0x79 0x7A 0x7B 0x7C 0x7D 0x7E 0x7F
0x80 0x81 0x82 0x83 0x84 0x85 0x86 0x87 0x88 0x89 0x8A 0x8B 0x8C 0x8D 0x8E 0x8F
0x90 0x91 0x92 0x93 0x94 0x95 0x96 0x97 0x98 0x99 0x9A 0x9B 0x9C 0x9D 0x9E 0x9F
0xA0 0xA1 0xA2 0xA3 0xA4 0xA5 0xA6 0xA7 0xA8 0xA9 0xAA 0xAB 0xAC 0xAD 0xAE 0xAF
0xB0 0xB1 0xB2 0xB3 0xB4 0xB5 0xB6 0xB7 0xB8 0xB9 0xBA 0xBB 0xBC 0xBD 0xBE 0xBF
0xC0 0xC1 0xC2 0xC3 0xC4 0xC5 0xC6 0xC7 0xC8 0xC9 0xCA 0xCB 0xCC 0xCD 0xCE 0xCF
0xD0 0xD1 0xD2 0xD3 0xD4 0xD5 0xD6 0xD7 0xD8 0xD9 0xDA 0xDB 0xDC 0xDD 0xDE 0xDF
0xE0 0xE1 0xE2 0xE3 0xE4 0xE5 0xE6 0xE7 0xE8 0xE9 0xEA 0xEB 0xEC 0xED 0xEE 0xEF
0xF0 0xF1 0xF2 0xF3 0xF4 0xF5 0xF6 0xF7 0xF8 0xF9 0xFA 0xFB 0xFC 0xFD 0xFE 0xFF
AT24C02 reading...
0x00 0x01 0x02 0x03 0x04 0x05 0x06 0x07 0x08 0x09 0x0A 0x0B 0x0C 0x0D 0x0E 0x0F
0x10 0x11 0x12 0x13 0x14 0x15 0x16 0x17 0x18 0x19 0x1A 0x1B 0x1C 0x1D 0x1E 0x1F
0x20 0x21 0x22 0x23 0x24 0x25 0x26 0x27 0x28 0x29 0x2A 0x2B 0x2C 0x2D 0x2E 0x2F
0x30 0x31 0x32 0x33 0x34 0x35 0x36 0x37 0x38 0x39 0x3A 0x3B 0x3C 0x3D 0x3E 0x3F
0x40 0x41 0x42 0x43 0x44 0x45 0x46 0x47 0x48 0x49 0x4A 0x4B 0x4C 0x4D 0x4E 0x4F
0x50 0x51 0x52 0x53 0x54 0x55 0x56 0x57 0x58 0x59 0x5A 0x5B 0x5C 0x5D 0x5E 0x5F
0x60 0x61 0x62 0x63 0x64 0x65 0x66 0x67 0x68 0x69 0x6A 0x6B 0x6C 0x6D 0x6E 0x6F
0x70 0x71 0x72 0x73 0x74 0x75 0x76 0x77 0x78 0x79 0x7A 0x7B 0x7C 0x7D 0x7E 0x7F
0x80 0x81 0x82 0x83 0x84 0x85 0x86 0x87 0x88 0x89 0x8A 0x8B 0x8C 0x8D 0x8E 0x8F
0x90 0x91 0x92 0x93 0x94 0x95 0x96 0x97 0x98 0x99 0x9A 0x9B 0x9C 0x9D 0x9E 0x9F
0xA0 0xA1 0xA2 0xA3 0xA4 0xA5 0xA6 0xA7 0xA8 0xA9 0xAA 0xAB 0xAC 0xAD 0xAE 0xAF
0xB0 0xB1 0xB2 0xB3 0xB4 0xB5 0xB6 0xB7 0xB8 0xB9 0xBA 0xBB 0xBC 0xBD 0xBE 0xBF
0xC0 0xC1 0xC2 0xC3 0xC4 0xC5 0xC6 0xC7 0xC8 0xC9 0xCA 0xCB 0xCC 0xCD 0xCE 0xCF
0xD0 0xD1 0xD2 0xD3 0xD4 0xD5 0xD6 0xD7 0xD8 0xD9 0xDA 0xDB 0xDC 0xDD 0xDE 0xDF
0xE0 0xE1 0xE2 0xE3 0xE4 0xE5 0xE6 0xE7 0xE8 0xE9 0xEA 0xEB 0xEC 0xED 0xEE 0xEF
0xF0 0xF1 0xF2 0xF3 0xF4 0xF5 0xF6 0xF7 0xF8 0xF9 0xFA 0xFB 0xFC 0xFD 0xFE 0xFF
I2C-AT24C02 test passed!
```

## 5.10. SPI\_SPI\_FLASH

### 5.10.1. DEMO purpose

This demo includes the following functions of GD32 MCU:

- Learn to use the master mode of SPI unit to read and write NOR Flash with the SPI interface

### 5.10.2. DEMO running result

The computer serial port line connected to the GD-Link port of development board, set the baud rate of HyperTerminal software to 115200, 8 bits data bit, 1 bit stop bit.

Download the program <10\_SPI\_SPI\_Flash> to the EVAL board, the HyperTerminal software can observe the operation condition and will display the ID of the flash, 256 bytes data which

are written to and read from flash. Compare the data that were written to the flash and the data that were read from the flash. If they are the same, the serial port will output “SPI-GD25Q16 Test Passed!”, otherwise, the serial port will output “Err: Data Read and Write aren’t Matching.”. At last, turn on and off the LEDs one by one. The following is the experimental results.

```
#####
#####

GD32C231C_EVAL_1.0 System is Starting up...

GD32C231C_EVAL_1.0 Flash:65535K

GD32C231C_EVAL_1.0 The CPU Unique Device ID:[FFFFFFFF-FFFFFFFF-FFFFFFFF]

GD32C231C_EVAL_1.0 SPI Flash:GD25Q16 configured...

The Flash_ID:0xC84015

Write to tx_buffer:

0x00 0x01 0x02 0x03 0x04 0x05 0x06 0x07 0x08 0x09 0x0A 0x0B 0x0C 0x0D
0x0E 0x0F

0x10 0x11 0x12 0x13 0x14 0x15 0x16 0x17 0x18 0x19 0x1A 0x1B 0x1C 0x1D
0x1E 0x1F

0x20 0x21 0x22 0x23 0x24 0x25 0x26 0x27 0x28 0x29 0x2A 0x2B 0x2C 0x2D
0x2E 0x2F

0x30 0x31 0x32 0x33 0x34 0x35 0x36 0x37 0x38 0x39 0x3A 0x3B 0x3C 0x3D
0x3E 0x3F

0x40 0x41 0x42 0x43 0x44 0x45 0x46 0x47 0x48 0x49 0x4A 0x4B 0x4C 0x4D
0x4E 0x4F

0x50 0x51 0x52 0x53 0x54 0x55 0x56 0x57 0x58 0x59 0x5A 0x5B 0x5C 0x5D
0x5E 0x5F

0x60 0x61 0x62 0x63 0x64 0x65 0x66 0x67 0x68 0x69 0x6A 0x6B 0x6C 0x6D
0x6E 0x6F
```

## 5.11. I2S\_Audio\_Player

### 5.11.1. DEMO purpose

This Demo includes the following functions of GD32 MCU :

- Learn to use I2S module to output audio file
- Parsing audio files of wav format

GD32C231C-EVAL board integrates the I2S (Inter-IC Sound) module, and the module can communicate with external devices using the I2S audio protocol. This Demo mainly shows how to use the I2S interface of the board for audio output.

### 5.11.2. DEMO running result

Download the program<11\_I2S\_Audio\_Player>to the EVAL board, insert the headphone into the audio port, and then listen to the audio file.

## 5.12. RCU\_Clock\_Out

### 5.12.1. DEMO purpose

This demo includes the following functions of GD32 MCU:

- Learn to use GPIO control the LED
- Learn to use the clock output function of RCU
- Learn to communicate with PC by USART

### 5.12.2. DEMO running result

Download the program <12\_RCU\_Clock\_Out> to the EVAL board and run. Connect serial cable to USART, open the HyperTerminal. When the program is running, HyperTerminal will display the initial information. Then user can choose the type of the output clock by pressing the User key. After pressing, the corresponding LED will be turned on and HyperTerminal will display which mode be selected. The frequency of the output clock can be observed through the oscilloscope by PA8 pin.

Information via a serial port output as following:

```

/===== Gigadevice Clock Output Demo =====/
press user key to select clock output source
CK_OUT: system clock, DIV: 4
CK_OUT: IRC48M, DIV: 8
CK_OUT: IRC32K, DIV: 1
CK_OUT: LXTAL, DIV: 1
CK_OUT: HXTAL, DIV: 1

```

## 5.13. PMU\_Sleep\_Wakeup

### 5.13.1. DEMO purpose

This Demo includes the following functions of GD32 MCU:

- Learn to use the USART receive interrupt to wake up the PMU from sleep mode

### 5.13.2. DEMO running result

Download the program < 13\_PMU\_sleep\_wakeup > to the EVAL board, connect serial cable to USART. After power-on, LED1 and LED2 flash once. The MCU will enter sleep mode and the software stop running. When the USART receives a byte of data from the HyperTerminal, the MCU will wake up from a receive interrupt. And LED1 and LED2 will flash together.

## 5.14. RTC\_Calendar

### 5.14.1. DEMO purpose

This demo includes the following functions of GD32 MCU:

- Learn to use RTC module to implement calendar function
- Learn to use LCD module to display the time of calendar

### 5.14.2. DEMO running result

Download the program <14\_RTC\_Calendar> to the EVAL board and run. When the program is running, the four LEDs, LED1 to LED4 turn on, then turn off. And then the LCD prints out the information of the board, and the calendar. When you press the Wakeup key, the time will be configured to 2024-09-13, 12:00:00



## 5.15. TIMER\_Breath\_LED

### 5.15.1. DEMO purpose

This Demo includes the following functions of GD32 MCU:

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

### 5.15.2. DEMO running result

Download the program <21\_TIMER\_Breath\_LED> to the GD32C231C-EVAL board and run. When the program is running, you can see LED1 lighting from dark to bright gradually and then gradually darken, ad infinitum, just like breathing as rhythm.

## 5.16. SPI\_LCD

### 5.16.1. DEMO purpose

This demo includes the following functions of GD32 MCU:

- Learn how to use SPI to drive TFT LCD screen and display

GD32C231C-EVAL board has a TFT LCD screen which supports SPI interface. In this demo, tests of font, number, draw and color are displayed on the LCD screen respectively

### 5.16.2. DEMO running result

LCD is controlled by SPI module on GD32C231C-EVAL board. Download the program <16\_SPI\_LCD > to the EVAL board. All the LEDs are turned on and then turned off for test. After that, the LCD screen on the board will display the GUI tests in infinite loop.



## 6. Revision history

Table 6-1. Revision history

Revision No.	Description	Date
1.0	Initial Release	Jun.3, 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, 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.