GigaDevice Semiconductor Inc.

GD32190R-EVAL

User Guide
V3.1

(Dec.31. 2021)

Table of Contents

List of Figures

List of Tables

1. Summary

GD32190R-EVAL-V1.2 evaluation board uses GD32F190RBT6 as the main controller. As a complete development platform of GD32F1x0 powered by ARM® Cortex™-M3 core, the board supports full range of peripherals. It uses mini-USB interface or AC/DC adapter to supply 3.3V power. SWD, Reset, Boot, User button key, LED, I2C, I2S, USART, RTC, SLCD, HDMI-CEC, TSI, IFRP LED, IR Receiver, RTC, SPI, ADC, DAC and Extension Pin are also included. This document details its hardware schematic and the relevant applications.

2. Function Pin Assign

Table 2-1 Pin assignment

Function	Pin	Description
LED	PA11	LED1
	PA12	LED2
	PB6	LED3
	PB7	LED4
RESET	-	K1-Reset
KEY	PC13	KEY2
	PA0	KEY3
IFRR	PB9	IR_OUT
	PA7	IR_IN
CMP	PA1	CMP0
HDMI_CEC	PB8	CEC
CAN	PA5	CAN0_H
	PA6	CAN0_L
	PB12	CAN1_RX
	PB13	CAN1_TX
I2C	PB10	I2C0_SCL
	PB11	I2C0_SDA
I2S	PC10	I2S2_CK
	PC11	I2S2_MCK
	PC12	I2S2_SD
	PA15	I2S2_WS
OPA	PC1	OPA2_VINP
	PC2	OPA2_VINM
	PC3	OPA2_VOUT
USART0	PA9	USART0_TX
	PA10	USART0_RX
USART1	PA2	USART1_TX
	PA3	USART1_RX
SLCD	PA8	SLCD_COM0

Function	Pin	Description
	PA9	SLCD_COM1
	PA10	SLCD_COM2
	PB9	SLCD_COM3
	PB14	SLCD_SEG14
	PB15	SLCD_SEG15
	PC4	SLCD_SEG22
	PC5	SLCD_SEG23
	PC6	SLCD_SEG24
	PC7	SLCD_SEG25
	PC8	SLCD_SEG26
	PC9	SLCD_SEG27
	PF4	SLCD_SEG28
	PF5	SLCD_SEG29
	PF6	SLCD_SEG30
	PF7	SLCD_SEG31
SPI	PB3	SPI0_SCK
	PB4	SPI0_MISO
	PB5	SPI0_MOSI
	PD2	SPI_CS
ADC	PC0	ADC_IN10
DAC	PA4	DAC_OUT0
	PA5	DAC_OUT1
TSI	PB0	TSI_G2_IO1
	PB1	TSI_G2_IO2
	PB2	TSI_G2_IO3

3. Getting started

The EVAL Board uses mini-USB connecter or AC/DC adapter to get power, the hardware system power is +3.3V. A mini-USB cable and a J-Link tool are necessary to down programs. Select the correct boot mode and then power on, the LED5 will turn on, which indicates the power supply is ready.

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, install the GD32F1x0_AddOn.3.2.0.exe to load the associated files.

2. If you use Keil uVision5 to open the project, there are two ways to solve the "Device Missing (s)" problem. One is to install GigaDevice.GD32F1x0_DFP.3.2.0.pack. In Project

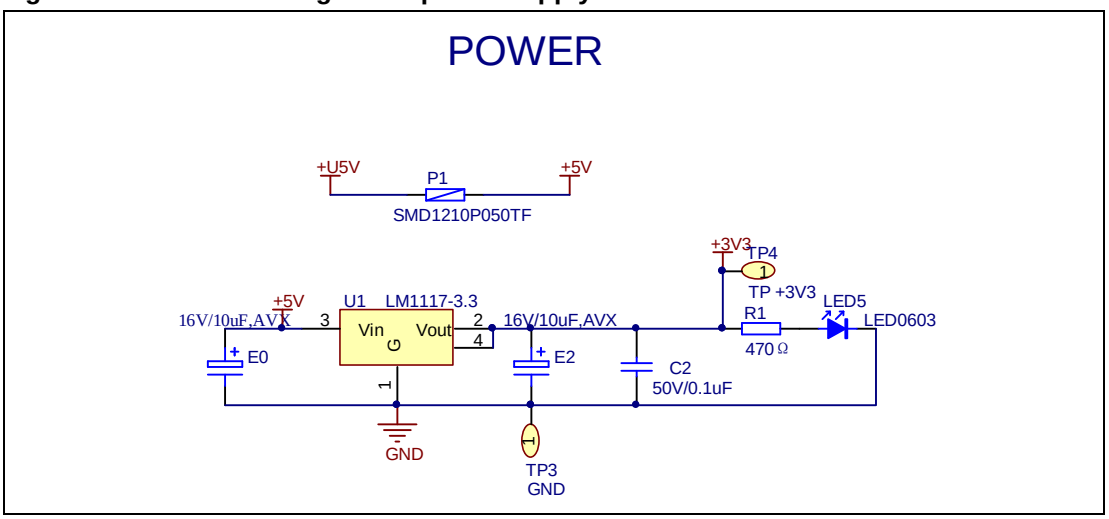
menu, select the Manage sub menu, click on the “Version Migrate 5 Format...” menu, the Keil uVision4 project will be converted to Keil uVision5 project. Then add “C:\Keil_v5\ARM\Pack\ARM\CMSIS\4.2.0\CMSIS\Include” to C/C++ in Option for Target. The other is to install Addon directly. Select the installation directory of Keil uVision5 software, such as C:\Keil_v5, in Destination Folder of Folder Selection. Select the corresponding device in Device of Option for Target and add “C:\Keil_v5\ARM\Pack\ARM\CMSIS\4.2.0\CMSIS\Include” to C/C++ in Option for Target.

3. If you use IAR to open the project, install IAR_GD32F1x0_ADDON.3.2..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

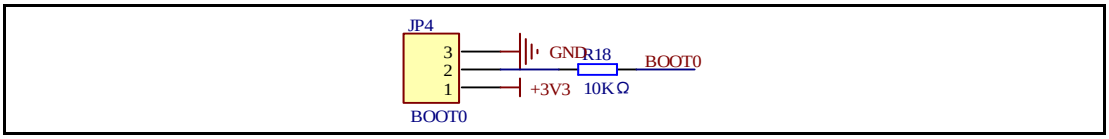
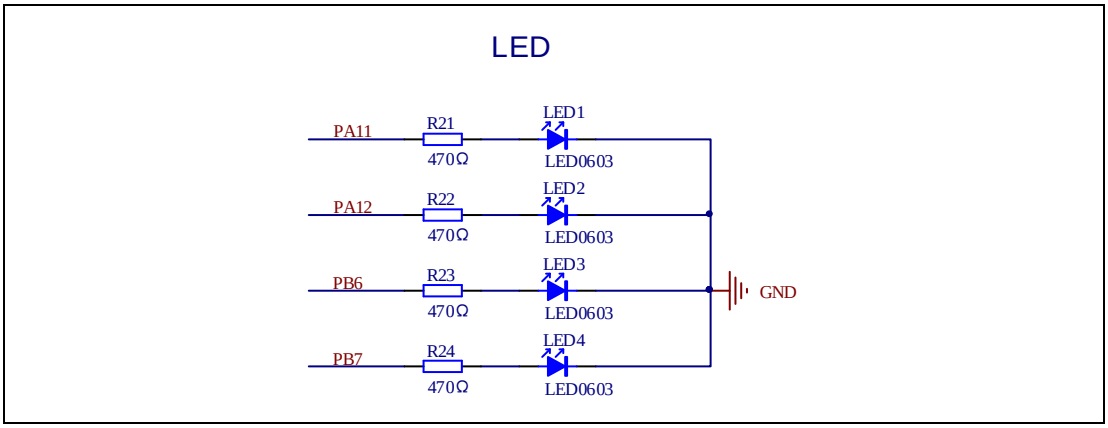


Table 4-1 Boot configuration

BOOT1	BOOT0	Boot Mode
Default	2-3	User memory
	1-2	System memory
Changed by ISP	1-2	SRAM memory

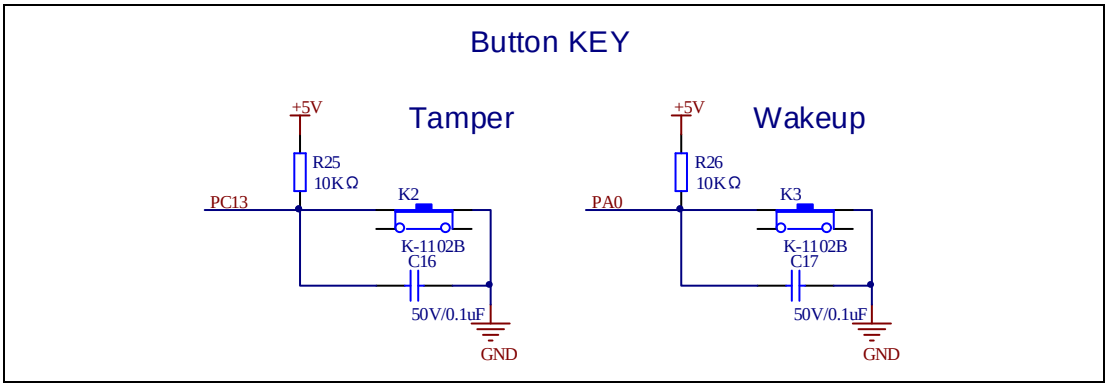
4.3. LED

Figure 4-3 Schematic diagram of LED function



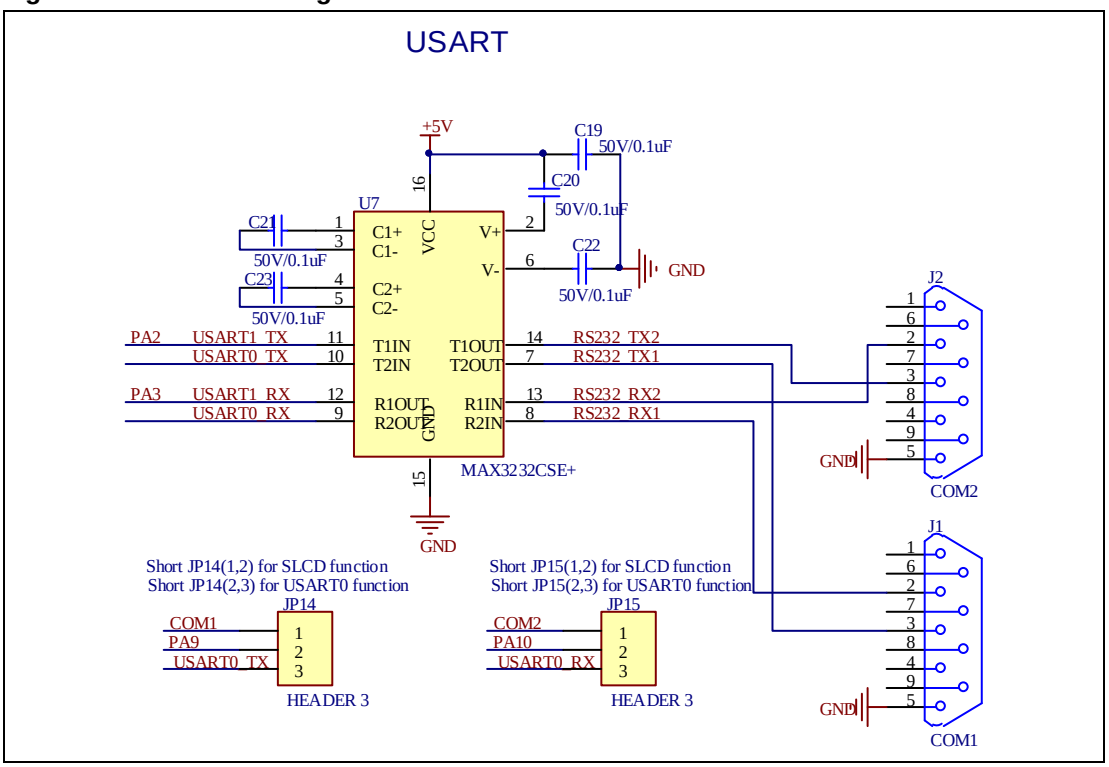
4.4. KEY

Figure 4-4 Schematic diagram of Key function



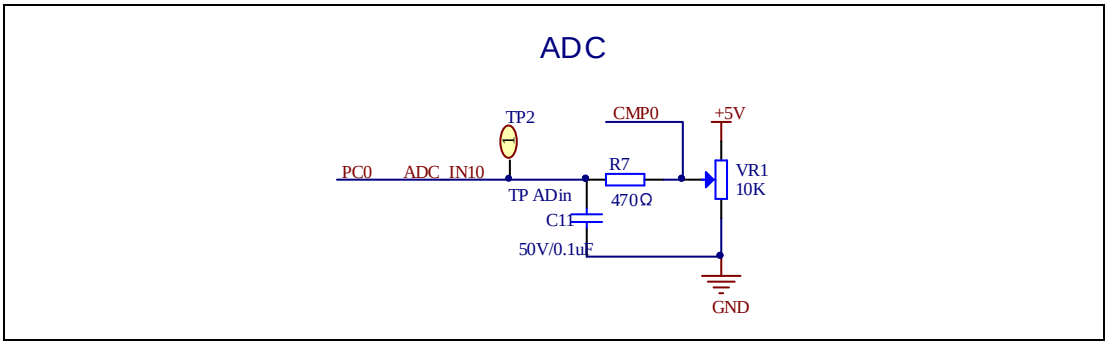
4.5. USART

Figure 4-5 Schematic diagram of USART function



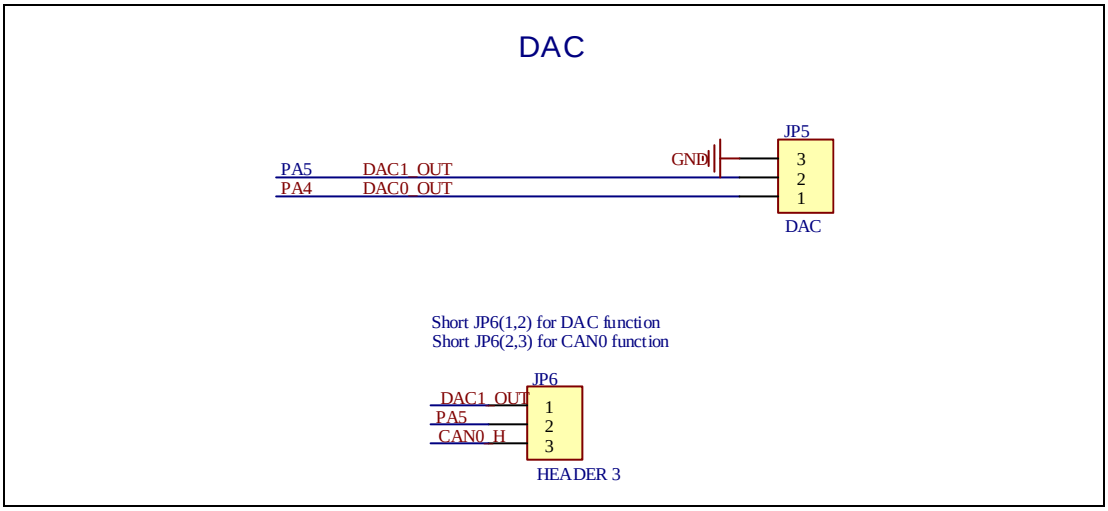
4.6. ADC

Figure 4-6 Schematic diagram of ADC function



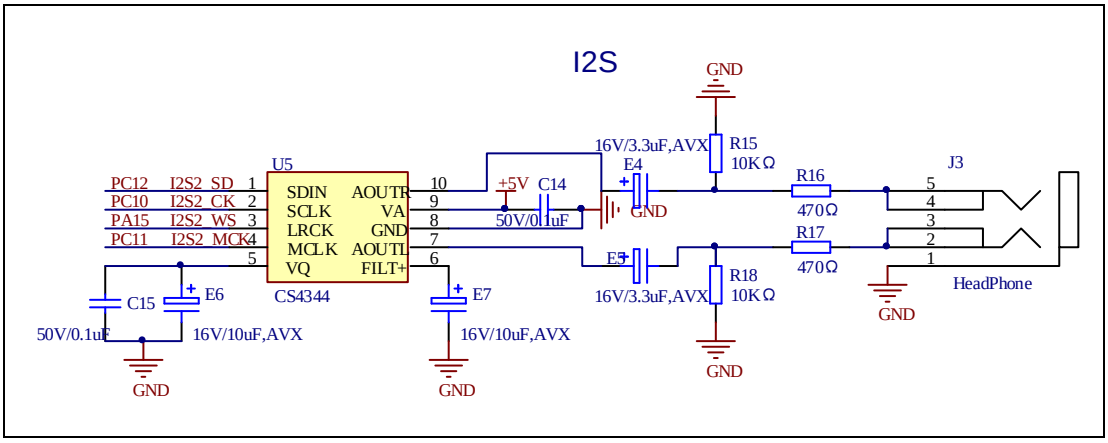
4.7. DAC

Figure 4-7 Schematic diagram of DAC function



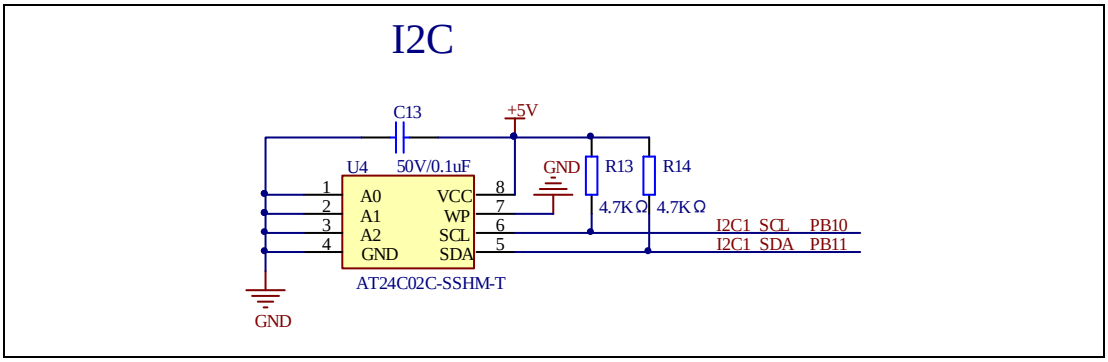
4.8. I2S

Figure 4-8 Schematic diagram of I2S function



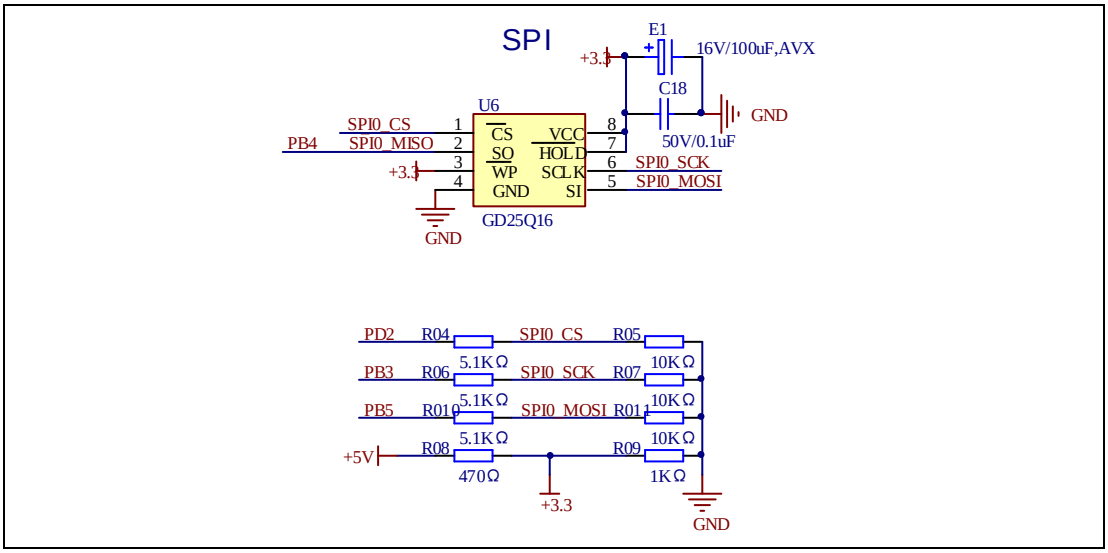
4.9. I2C

Figure 4-9 Schematic diagram of I2C function



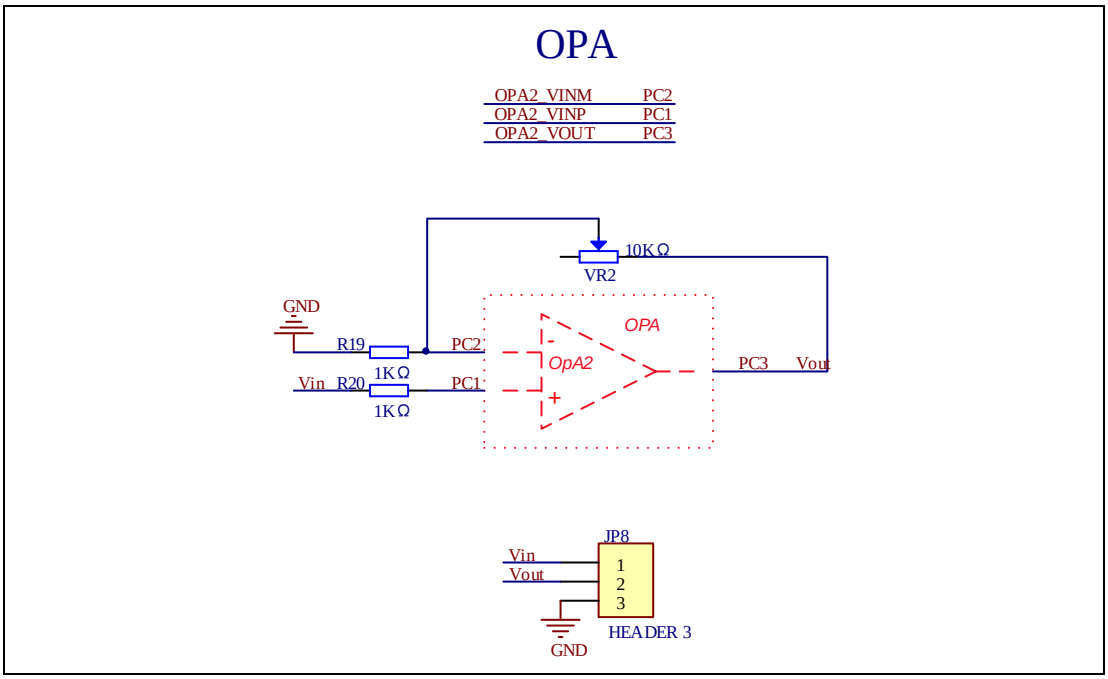
4.10. SPI

Figure 4-10 Schematic diagram of SPI function



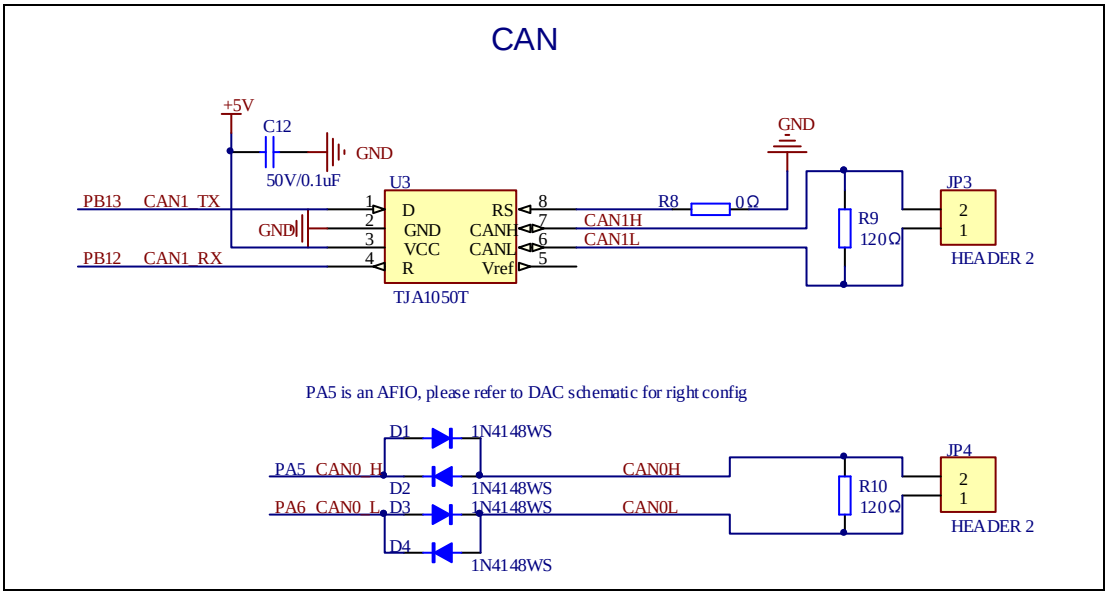
4.11. OPA

Figure 4-11 Schematic diagram of OPA function



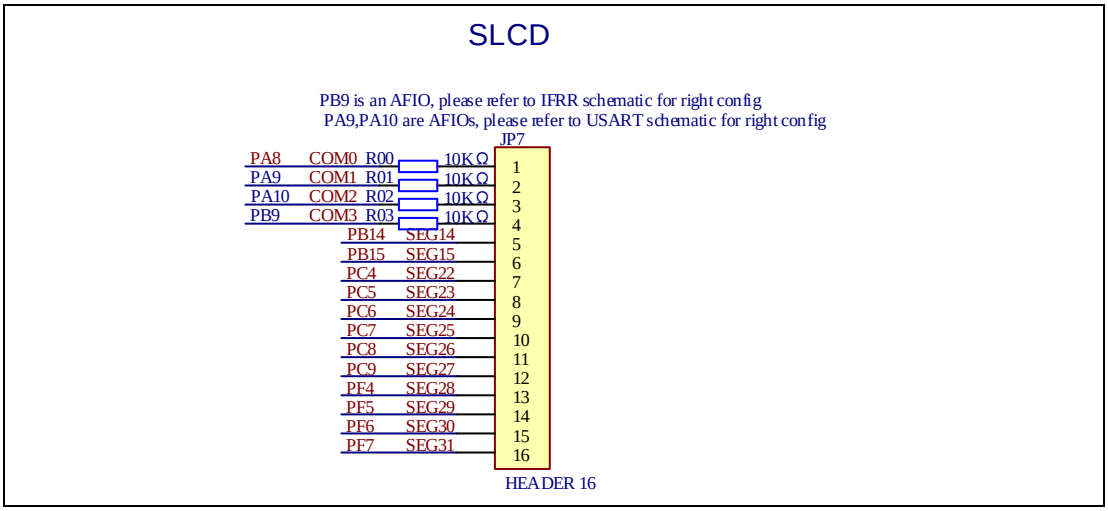
4.12. CAN

Figure 4-12 Schematic diagram of CAN function



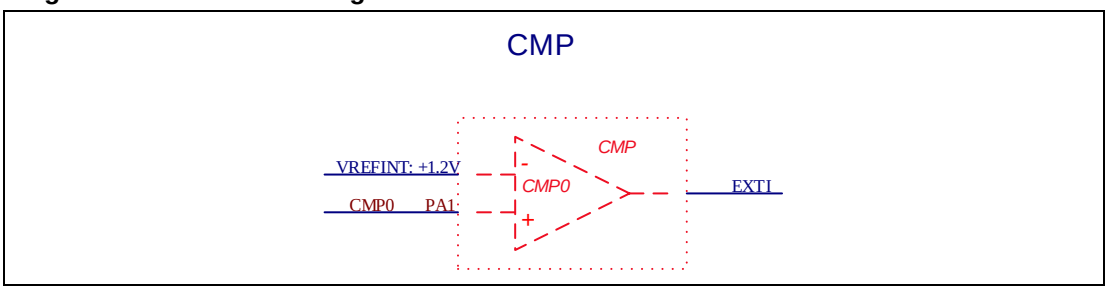
4.13. SLCD

Figure 4-13 Schematic diagram of SLCD function



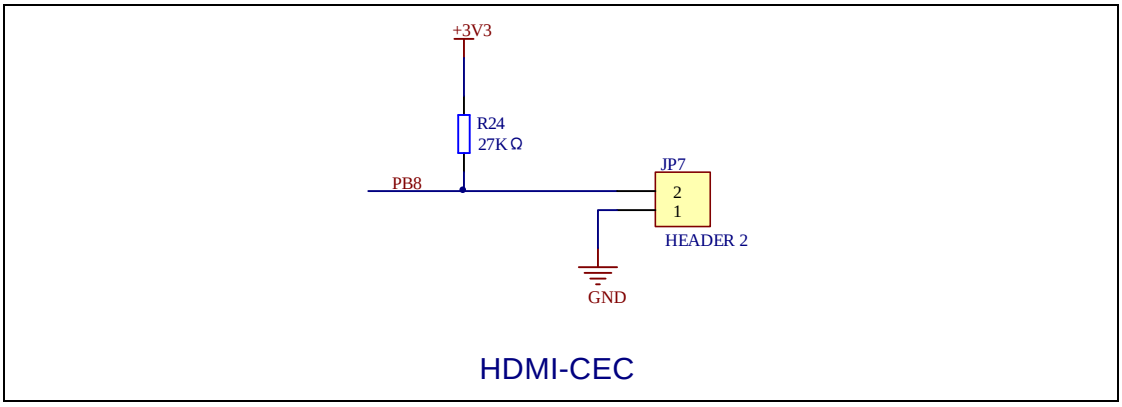
4.14. CMP

Figure 4-14 Schematic diagram of CMP function



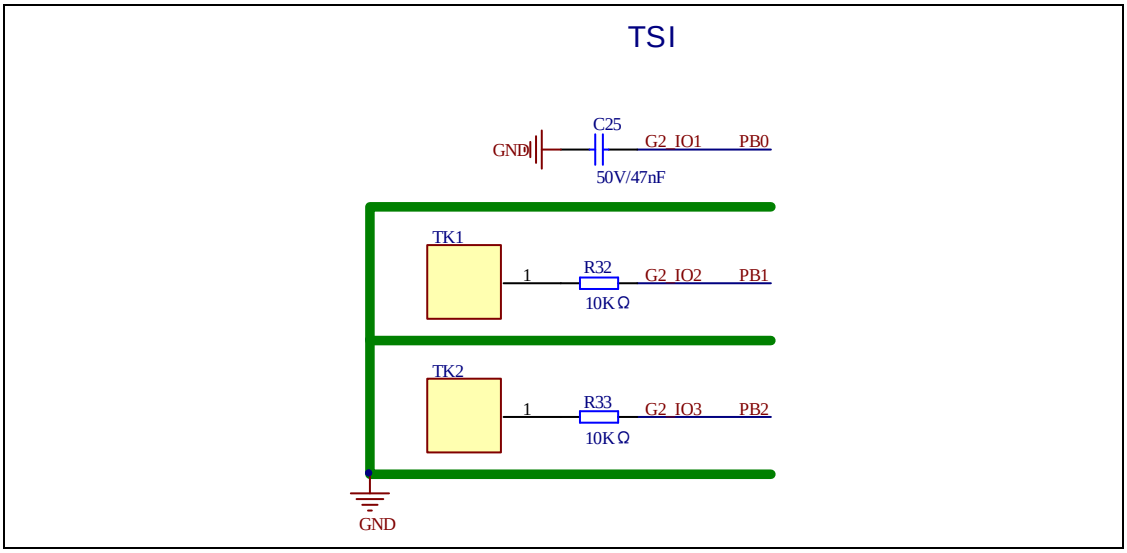
4.15. HDMI-CEC

Figure 4-15 Schematic diagram of HDMI-CEC function



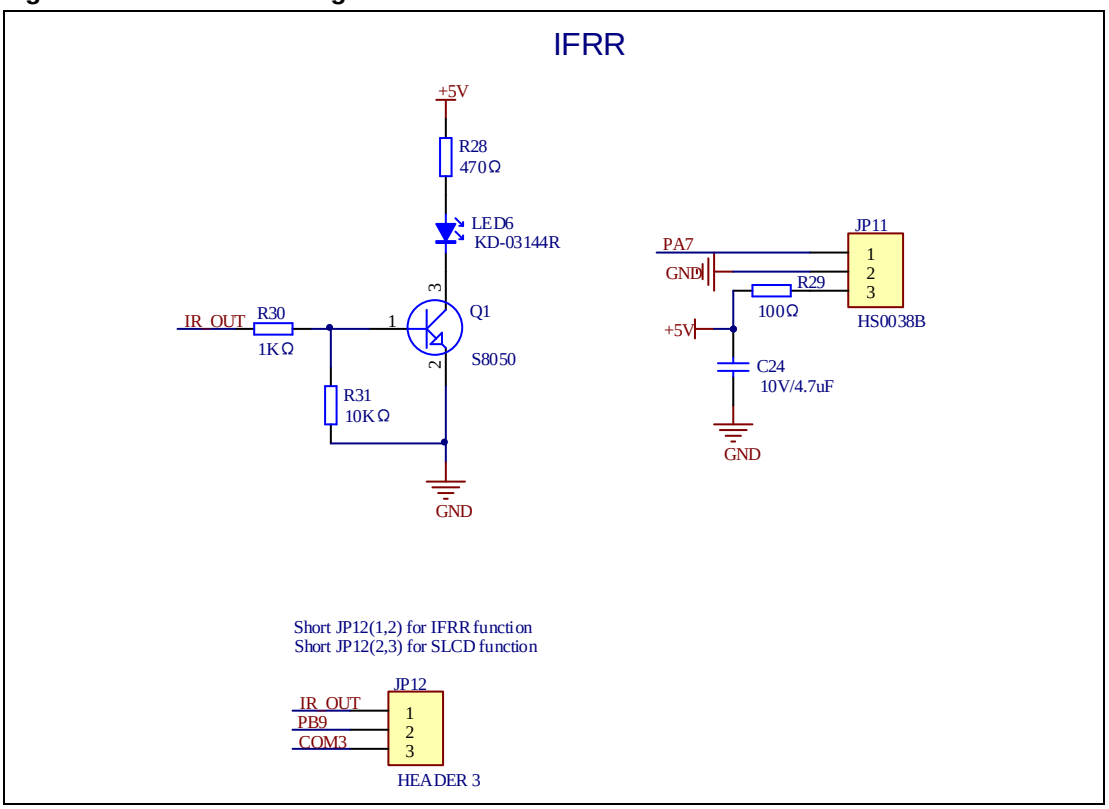
4.16. TSI

Figure 4-16 Schematic diagram of TSI function



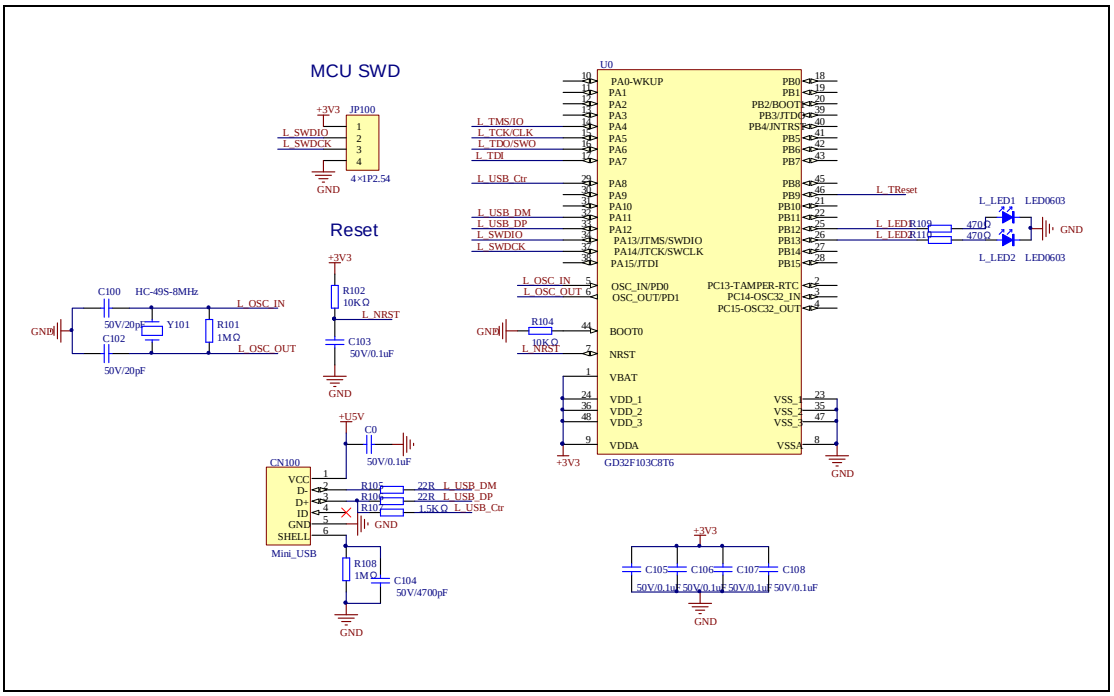
4.17. IFRP

Figure 4-17 Schematic diagram of IFRP function



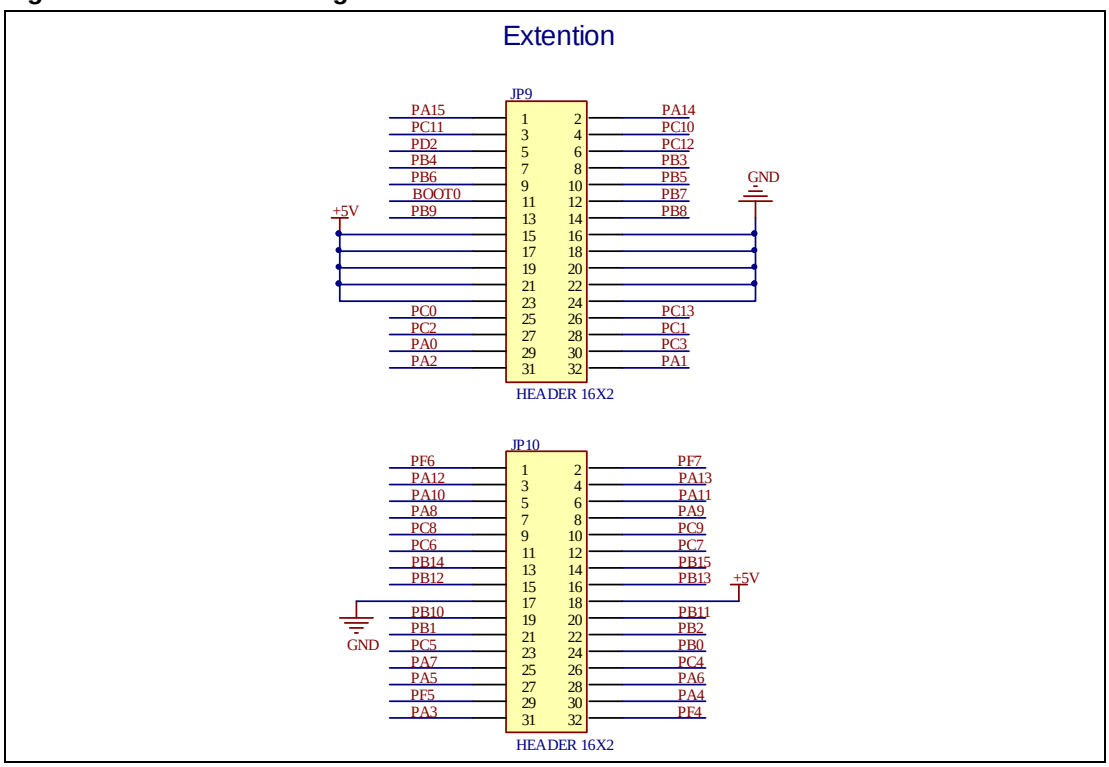
4.18. GD-Link

Figure 4-18 Schematic diagram of GD-Link function



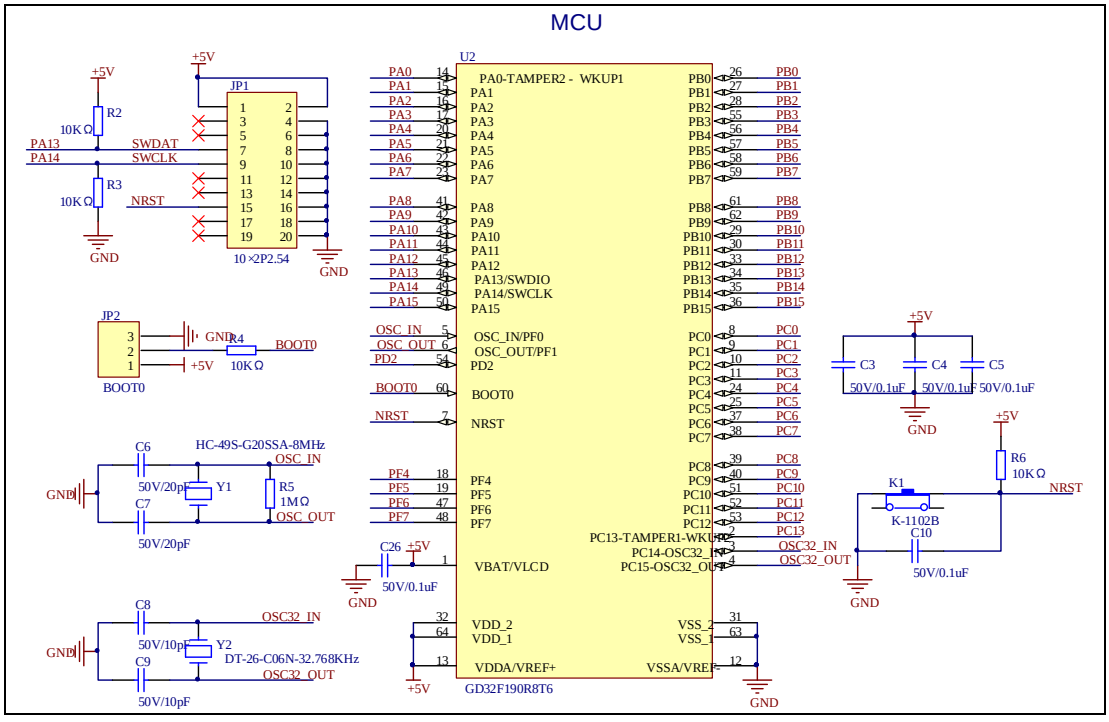
4.19. Extension

Figure 4-19 Schematic diagram of Extension Pin



4.20. MCU

Figure 4-20 Schematic diagram of MCU Pin



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 for controlling the LED
- Learn to use SysTick to generate 1ms delay

GD32190R-EVAL-V1.2 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 EVAL board, four LEDs will turn on one by one from LED1 to LED4 every 200ms, and then turn off together. 200ms later, the four LEDs work like previous again.

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

GD32190R-EVAL-V1.2 board has three keys and four LEDs. The four keys are Reset key, Tamper key and Wakeup key. The LED1, LED2, LED3 and LED4 are controlled by GPIO.

This demo will show how to use the Tamper 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 0, wait for 50ms. Then check the input value of the IO port again. If the value is still 0, indicates that the button is pressed down successfully, and light the LED2.

5.2.2. DEMO running result

Download the program <02_GPIO_Key_Polling_mode> to the EVAL board, first of all, all the LEDs will be flashed once for test. Then press down the Tamper 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 to control the LED and the KEY
- Learn to use EXTI to generate external interrupt

GD32190R-EVAL-V1.2 board has three keys and four LEDs. The four keys are Reset key, Wakeup key and Tamper key. The LED1, LED2, LED3 and LED4 are controlled by GPIO.

This demo will show how to use EXTI interrupt line to control the LED2. When press down the Tamper 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, first of all, all the LEDs will be flashed once for test. Then press down the Tamper Key, LED2 will be turned on. Press down the Tamper 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: the Tamper key control the LED

5.4.2. DEMO running result

Download the program <04_USART_Printf> to the EVAL board and connect serial cable to COM2. This implementation outputs “please press the Tamper Key” on the hyperterminal. Press the Tamper key, serial port will output “USART Printf Example”. The information via a serial port output as following.

```
please press the Wakeup 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 EVAL_COM 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 run. Firstly, all the LEDs are turned on and off for test. Then, the COM 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 and LED2 are turned on, LED3 and LED4 are turned off. Otherwise, LED1 and LED2 are turned off, LED3 and LED4 are turned on.

The information via a serial port output 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 COM transmit and receive using DMA

5.6.2. DEMO running result

Download the program <06_USART_DMA> to the EVAL board and run. Firstly, all the LEDs are turned on and off for test. Then, the COM sends the tx_buffer array 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, LED3 and LED4 are turned on. Otherwise, LED1, LED2, LED3 and LED4 are turned off, LED3 and LED4 are turned on.

The information via a serial port output 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
```

5.7. ADC_Conversion_Triggered_By_Timer

5.7.1. DEMO purpose

This Demo includes the following functions of GD32 MCU:

- Learn to use ADC to convert analog to digital
- Learn to use TIMER to generate a CH event
- Learn to use LCD to show the ADC converted result

TIMER1 CH1 event triggers ADC conversion, the value displayed on the LCD corresponds to the ADC analog input, and changes with it. The converted data are moved to SRAM through DMA continuously.

5.7.2. DEMO running result

Download the program <07_ADC_Conversion_Triggered_By_Timer> to the GD32190R-EVAL-V1.2 board, adjust the adjustable potentiometer knob to change the analog input. The ADC, which is triggered by TIMER1 CH1 event, will convert the analog input and the converted data coinciding with the input, moved from ADC RDATA register to SRAM(through DMA), is sended to the Hyperterminal window.

5.8. DAC_Output_Voltage_Value

5.8.1. DEMO purpose

This Demo includes the following functions of GD32 MCU:

- Learn to use DAC channel to generate different voltages on DAC output

5.8.2. DEMO running result

Download the program <08_DAC_Output_Voltage_Value> to the EVAL board, the digital value is 0x7ff0, its converted analog voltage should be VREF/2, using the voltmeter to measure PA4, its value is 2.548V.

5.9. Comparator_Obtain_Brightness

5.9.1. DEMO purpose

This Demo includes the following functions of GD32 MCU:

- Learn to use comparator output compare result

There are two comparators on GD32190R-EVAL-V1.2 board and each comparator has two inputs. In this demo, one input is slide rheostat output voltage, and the other one is the reference voltage. Compare the two input voltages, the output is a high or low level, and the LEDs will performs the corresponding action.

5.9.2. DEMO running result

Download the program <09_Comparator_obtain_brightness> to the GD32190R-EVAL-V1.2 board, change the slide rheostat output voltage, comparing it with reference voltage, if it is larger than reference voltage LED1 is on, LED2 is off, otherwise LED1 is off, LED2 is on.

5.10. I2C_EEPROM

5.10.1. DEMO purpose

This Demo includes the following functions of GD32 MCU:

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

5.10.2. DEMO running result

Download the program <10_I2C_EEPROM> to the EVAL board and run. Connect serial cable to COM2, and 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 four LEDs lights flashing, otherwise the serial port will output "Err: data read and write aren't matching." and all the four LEDs light.

The output information via the serial port is as following.

```
I2C-24C02 configured....
The I2C0 is hardware interface
The speed is 400000
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!
```

5.11. SPI FLASH

5.11.1. DEMO purpose

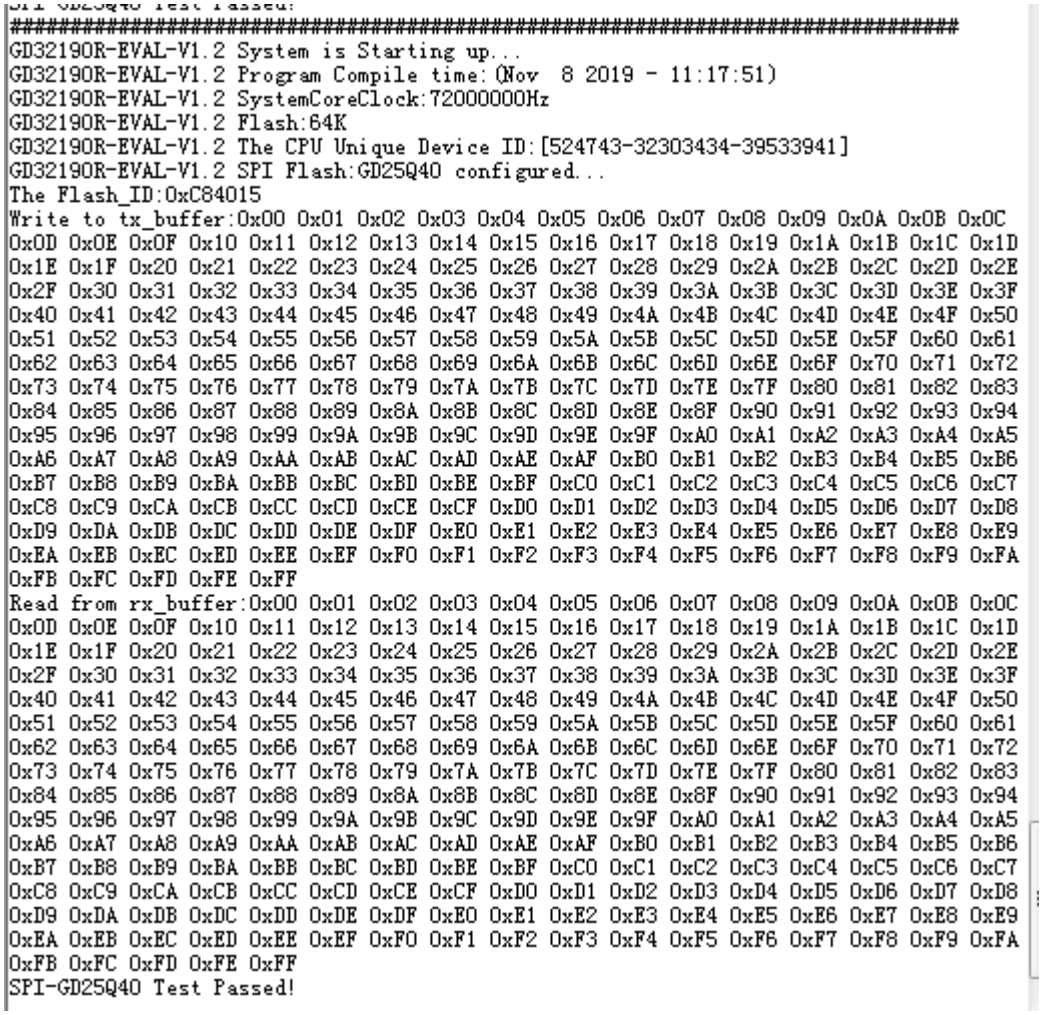
This Demo includes the following function of GD32 MCU:

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

5.11.2. DEMO running result

The computer serial port line connected to the COM1 port of development board, set the baud rate of serial assistant software to 115200, 8 bits data bit, 1 bit stop bit. Download the program <11_SPI_FLASH> to the EVAL board, through the serial assistant software can observe the operation condition and will display the ID of the flash, 256 bytes data which

write 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-GD25Q40 Test Passed!”, otherwise the serial port will output “Err: Data Read and Write aren't Matching.”. At last, turn on and off the led one by one. When use GD32190R-EVAL-V1.2 board, you should jump the JP14 and JP15 to COM1. The following is the experimental results.



5.12. HDMI-CEC_HostSlaveCommunication

5.12.1. DEMO purpose

This Demo includes the following functions of GD32 MCU:

- Learn the communication function of HDMI-CEC

In the process of communication, the sender sends data to receiver through the key interrupt, the receiver for receiving data in the CEC interrupt. The entire communication process does not make the error processing.

5.12.2. DEMO running result

This routines need to prepare two EVAL board, one board as a sender, the other as a receiver. First use the DuPont to connect CEC bus (PB8) and ground wire (GND) pins in the two board, and then download the program <12_HDMI_CEC_HostSlaveCommunication> to the board for running. When the program runs, the first development board of the SLCD display is data 0, press one of the development board TAMPER key, the other piece of the development board SLCD number will increase, which shows the end of a data transmission. Each it increases to 9, it will clear to 0 to re-increase; press the WAKEUP key, the number will decline, which also shows the end of a data transmission. Every time it decreases to 0, it will return to the number 9 to re-decrease.

5.13. I2S_Audio_Player

5.13.1. DEMO purpose

This Demo includes the following functions of GD32 MCU:

- Learn to use I2S module to output audio file

GD32190R-EVAL-V1.2 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.

5.13.2. DEMO running result

Download the program <13_I2S_Audio_Player> to the EVAL board. After downloading the program, insert the earphone into the audio port J3, then listen to the audio file.

5.14. RCU_Clock_Out

5.14.1. 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

5.14.2. DEMO running result

Download the program <14_RCU_Clock_Out> to the EVAL board and run. Connect serial cable to EVAL_COM1, 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 button. 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: IRC28M, DIV:1
CK_OUT: IRC40K, DIV:1
CK_OUT: LXTAL, DIV:1
CK_OUT: CKSYS, DIV:4
CK_OUT: IRC8M, DIV:1
```

5.15. PMU_sleep_wakeup

5.15.1. 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

5.15.2. DEMO running result

Download the program <15_PMU_sleep_wakeup> to the EVAL board, connect serial cable to EVAL_COM. After power-on, all the LEDs are off. The MCU will enter sleep mode and the software stop running. When the USART0 receives a byte of data from the HyperTerminal, the MCU will wake up from a receive interrupt. And all the LEDs will flash together.

5.16. RTC_Calendar

5.16.1. DEMO purpose

This Demo includes the following functions of GD32 MCU:

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

5.16.2. DEMO running result

Jump the JP12, JP14 and JP15 to SLCD with the jumper cap, and download the program<16_RTC_Calendar> to the EVAL board and run. When the program is running, SLCD displays the time of calendar, when you press the Tamper Key, the day time, the year, the month and the date successively display on the SLCD.

5.17. IRInfrared_Transceiver

5.17.1. DEMO purpose

This Demo includes the following functions of GD32 MCU:

- Learn to use general timer output PWM wave
- Learn to use general timer generated update interrupt
- Learn to use general timer capture interrupt
- Learn to use general timer TIMER15 and TIMER16 implement Infrared function

5.17.2. DEMO running result

Download the program <17_IRInfrared_Transceiver> to the EVAL board and run. When the program is running, if the infrared receiver received data is correct, LED1, LED2, LED3, LED4 light in turn, otherwise LED1, LED2, LED3, LED4 toggle together.

5.18. TIMER_Breath_LED

5.18.1. DEMO purpose

This Demo includes the following functions of GD32 MCU:

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

5.18.2. DEMO running result

Use the DuPont line to connect the TIMER0_CH0 (PA8) and LED1 (PA11), and then download the program <18_TIMER_Breath_LED> to the GD32190R-EVAL-V1.2 board and run. PA8 should not be reused by other peripherals.

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.

5.19. TSI_TouchKey_leds

5.19.1. DEMO purpose

This Demo includes the following functions of GD32 MCU:

- Learn to use TSI module implement Touch Key function

5.19.2. DEMO running result

Download the program <19_TSI_TouchKey_Leds> to the EVAL board and run. When the program is running, you can use a finger touch the TSI Sensor on the EVAL board, and then the associated LED1 and LED2 will light.

5.20. OPA_Amplify

5.20.1. DEMO purpose

This Demo includes the following functions of GD32 MCU:

- Learn to use OPA module amplify the input voltage
- Learn to use ADC to convert analog to digital

5.20.2. DEMO running result

Download the program <20_OPA_Amplify> to the EVAL board and run. When the program is running, connect the input voltage to the pin1 pin of JP8, then adjust the slide rheostat VR2 to change the amplification factor of the input voltage, and the input voltage is amplified by OPA2.Finally, the result of the output voltage value is printed on the Hyperterminal window through ADC1 analog-to-digital conversion.

5.21. SLCD_Glass

5.21.1. DEMO purpose

This Demo includes the following functions of GD32 MCU:

- Learn to use SLCD module to display number

5.21.2. DEMO running result

Jump the JP12 to SLCD with the jumper cap, and download the program<21_SLCD_Glass> to the EVAL board and run. When the program is running, you can see the SLCD displaying the number which adds 1 per second.

5.22. CAN_DualCAN

5.22.1. DEMO purpose

This Demo includes the following functions of GD32 MCU:

- Learn to use the CAN0 and CAN1 communication on one board
- Learn to communicate with PC by USART

5.22.2. DEMO running result

Download the program <22_CAN_DualCAN> to the EVAL board and run. When the program is running, the CAN0 and CAN1 peripherals will auto send and receive CAN frames by 1000ms in normal mode. When the frames are received, the receive data will be printed via COM2. And JP6 should connect to CAN1, JP4 CAN_L should connect to JP3 CAN_L, JP4 CAN_H should connect to JP3 CAN_H.

The output information via the serial port is as following.

```
GD32F1x0 Dual CAN
BAUDRATE = 1MBps   CAN1 receive data: 0
CAN0 receive data: 1000

CAN1 receive data: 1
CAN0 receive data: 999

CAN1 receive data: 2
CAN0 receive data: 998

CAN1 receive data: 3
CAN0 receive data: 997

CAN1 receive data: 4
CAN0 receive data: 996
```

6. Revision history

Table 6-1 Revision history

Revision No.	Description	Date
1.0	Initial Release	Jun.28, 2017
2.0	Updated format across the whole document	Jun.1, 2019
3.0	Updated format across the whole document	Nov.20, 2019
3.1	Updated format across the whole document	Dec.31, 2021

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 under the intellectual property laws and treaties of the People’s Republic of China and other jurisdictions worldwide. The Company reserves all rights under such laws and treaties and does not grant any license under its patents, copyrights, trademarks, or other intellectual property rights. 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.

The Company makes no warranty of any kind, express or implied, with regard to this document or any Product, including, but not limited to, the implied warranties of merchantability and fitness for a particular purpose. The Company does not assume any liability arising out of the application or use of any Product described in this document. Any information provided in this document is provided only for reference purposes. It is the responsibility of the user of this document to properly design, program, and test the functionality and safety of any application made of this information and any resulting product. Except for customized products which has been expressly identified in the applicable agreement, the Products are designed, developed, and/or manufactured for ordinary business, industrial, personal, and/or household applications only. The Products are not designed, intended, or authorized for use as components in systems designed or intended for the operation of weapons, weapons systems, nuclear installations, atomic energy control instruments, combustion control instruments, airplane or spaceship instruments, transportation instruments, traffic signal instruments, life-support devices or systems, other medical devices or systems (including resuscitation equipment and surgical implants), pollution control or hazardous substances management, or other uses where the failure of the device or Product could cause personal injury, death, property or environmental damage ("Unintended Uses"). Customers shall take any and all actions to ensure using and selling the Products in accordance with the applicable laws and regulations. The Company is not liable, in whole or in part, and customers shall and hereby do release the Company as well as it’s suppliers and/or distributors from any claim, damage, or other liability arising from or related to all Unintended Uses of the Products. Customers shall indemnify and hold the Company as well as it’s 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 Products.

Information in this document is provided solely in connection with the Products. The Company reserves the right to make changes, corrections, modifications or improvements to this document and Products and services described herein at any time, without notice.