GigaDevice Semiconductor Inc.

GD32L233R-EVAL
Arm® Cortex®-M23 32-bit MCU

用户指南

1.4 版本

(2026 年 8 月)

目录

目录 1

图 5

表 6

1. 简介 7

2. 功能引脚分配 8

3. 入门指南 10

4. 硬件设计概述 11

4.1. 供电电源 11

4.2. 启动方式选择 11

4.3. LED 指示灯 11

4.4. 按键 12

4.5. ADC 12

4.6. DAC 12

4.7. CMP 13

4.8. USART 13

4.9. I2C 13

4.10. I2S 14

4.11. SPI 14

4.12. SLCD 14

4.13. USB 15

4.14. Extension 15

4.15. GD-Link 16
4.16. MCU 17
5. 例程使用指南 18
5.1. GPIO 流水灯 18
5.1.1. DEMO 目的 18
5.1.2. DEMO 执行结果 18
5.2. GPIO 按键轮询模式 18
5.2.1. DEMO 目的 18
5.2.2. DEMO 执行结果 18
5.3. EXTI 按键中断模式 19
5.3.1. DEMO 目的 19
5.3.2. DEMO 执行结果 19
5.4. 串口打印 19
5.4.1. DEMO 目的 19
5.4.2. DEMO 执行结果 19
5.5. 串口中断收发 20
5.5.1. DEMO 目的 20
5.5.2. DEMO 执行结果 20
5.6. 串口 DMA 收发 20
5.6.1. DEMO 目的 20
5.6.2. DEMO 执行结果 20
5.7. LPUART Deepsleep 唤醒 21
5.7.1. DEMO 目的 21
5.7.2. DEMO 执行结果 21
5.8. ADC 温度传感器_内部参考电压 21
5.8.1. DEMO 目的 21
5.8.2. DEMO 执行结果 21
5.9. DAC 输出电压值 22
5.9.1. DEMO 目的 22

5.9.2. DEMO 执行结果 22
5.10. 比较器输出获取指示灯 22
5.10.1. DEMO 目的 22
5.10.2. DEMO 执行结果 23
5.11. I2C 访问 EEPROM 23
5.11.1. DEMO 目的 23
5.11.2. DEMO 执行结果 23
5.12. QSPI FLASH 24
5.12.1. DEMO 目的 24
5.12.2. DEMO 执行结果 24
5.13. I2S 音频播放 25
5.13.1. DEMO 目的 25
5.13.2. DEMO 执行结果 26
5.14. TRNG 随机数 26
5.14.1. DEMO 目的 26
5.14.2. DEMO 执行结果 26
5.15. CAU 加密处理器 26
5.15.1. DEMO 目的 26
5.15.2. DEMO 执行结果 27
5.16. RCU 时钟输出 28
5.16.1. DEMO 目的 28
5.16.2. DEMO 执行结果 28
5.17. CTC 校准 28
5.17.1. DEMO 目的 28
5.17.2. DEMO 执行结果 29
5.18. PMU 睡眠模式唤醒 29
5.18.1. DEMO 目的 29
5.18.2. DEMO 执行结果 29
5.19. RTC 日历 29
5.19.1. DEMO 目的 29

5.19.2. DEMO 执行结果 29

5.20. TIMER 呼吸灯 30

5.20.1. DEMO 目的 30
5.20.2. DEMO 执行结果 31

5.21. LPTIMER 睡眠模式的 PWM 输出 31

5.21.1. DEMO 目的 31
5.21.2. DEMO 执行结果 31

5.22. SLCD 液晶显示 31

5.22.1. DEMO 目的 31
5.22.2. DEMO 执行结果 31

5.23. USBD 键盘 32

5.23.1. DEMO 目的 32
5.23.2. DEMO 执行结果 32

5.24. USBD 虚拟串口 32

5.24.1. DEMO 目的 32
5.24.2. DEMO 执行结果 33

6. 版本历史 34

图

图 4-1. 供电电源原理图 11
图 4-2. 启动方式选择原理图 11
图 4-3. LED 功能原理图 11
图 4-4. 按键功能原理图 12
图 4-5. ADC 原理图 12
图 4-6. DAC 原理图 12
图 4-7. CMP 原理图 13
图 4-8. USART 原理图 13
图 4-9. I2C 原理图 13
图 4-10. I2S 原理图 14
图 4-11. SPI 原理图 14
图 4-12. SLCD 原理图 14
图 4-13. USB 原理图 15
图 4-14. Extension 原理图 15
图 4-15. GD-Link 原理图 16
图 4-16. MCU 原理图 17

表

表 2-1. 引脚分配 8
表 6-1. 版本历史 34

1. 简介

GD32L233R-EVAL 评估板使用 GD32L233RCT6 作为主控制器。评估板使用 GD-Link Mini USB 接口提供 5V 电源。提供包括扩展引脚在内的及 Reset，Boot，Button key，LED，I2S，I2C-EEPROM，SLCD，QSPI-Flash，USB，USART 转 USB 接口等外设资源。更多关于开发板的资料可以查看 GD32L233R-EVAL_Rev1.1 原理图。

2. 功能引脚分配

表 2-1. 引脚分配

功能	引脚	描述
LED	PC7	LED1
	PC8	LED2
	PC9	LED3
	PC11	LED4
RESET		Reset
KEY	PA0	K3(Wakeup)
	PC13	K2(Tamper)
ADC	PA1	ADC_IN1
DAC	PA4	DAC_OUT
USART	PA2	USART1_TX
	PA3	USART1_RX
I2C	PB10	I2C1_SCL
	PB11	I2C1_SDA
I2S	PC12	I2S1_SD
	PC10	I2S1_CK
	PA15	I2S1_WS
	PC6	I2S1_MCK
SPI	PD2	SPI0_CS
	PB3	SPI0_SCK
	PB4	SPI0_MISO
	PB5	SPI0_MOSI
	PB6	SPI0_IO2
	PB7	SPI0_IO3
SLCD	PA8	COM0
	PA9	COM1
	PA10	COM2
	PB9	COM3
	PB1	SEG6
	PB8	SEG16
	PB12	SEG12
	PB13	SEG13
	PB14	SEG14
	PB15	SEG15
	PC0	SEG18
	PC1	SEG19
	PC2	SEG20
	PC3	SEG21

功能	引脚	描述
	PC4	SEG22
	PC5	SEG23
CMP	PA1	CMP0_IP
USB	PA11	USB_DM
	PA12	USB_DP

3. 入门指南

评估板使用 GD-Link Mini USB 提供 5V 电源。下载程序到评估板需要使用 GD-Link 工具，在选择了正确的启动方式并且上电后，LED5 将被点亮，表明评估板供电正常。

所有例程提供了 Keil 和 IAR 两个版本,其中 Keil 版的工程是基于 Keil MDK-ARM 5.26 uVision5 创建的，IAR 版的工程是基于 IAR Embedded Workbench for ARM 8.32.1 创建的。在使用过程中有如下几点需要注意：

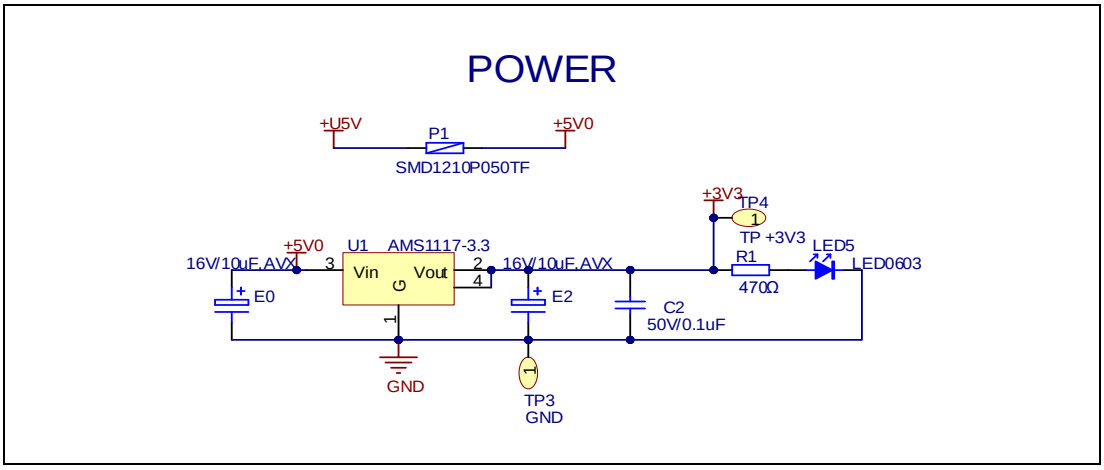
1、如果使用 Keil uVision5 打开工程，安装 GigaDevice.GD32L23x_DFP_1.0.0，以加载相关文件。

2、如果使用 IAR 打开工程，安装 IAR_GD32L23x_ADDON_1.0.0.exe，以加载相关文件。

4. 硬件设计概述

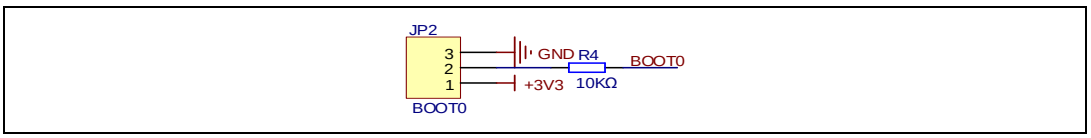
4.1. 供电电源

图4-1. 供电电源原理图



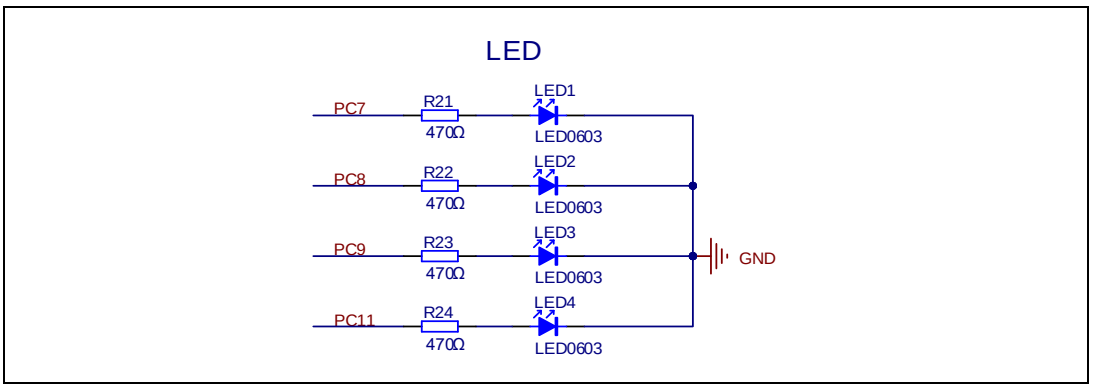
4.2. 启动方式选择

图4-2. 启动方式选择原理图



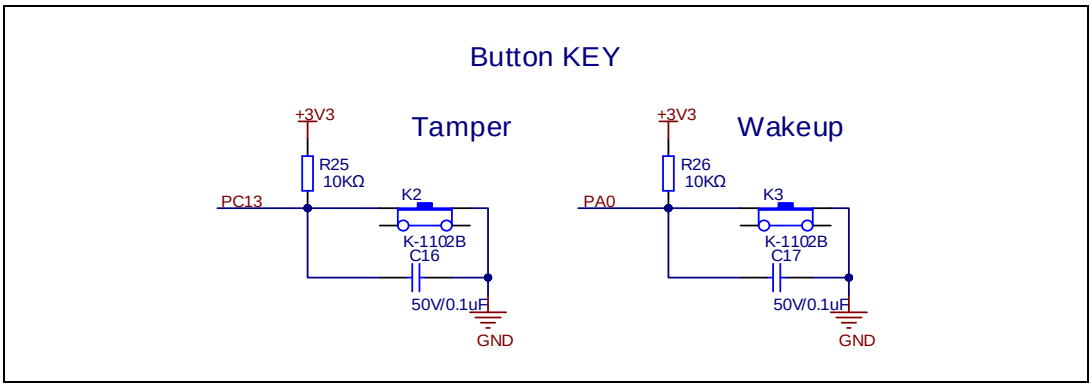
4.3. LED 指示灯

图4-3. LED功能原理图



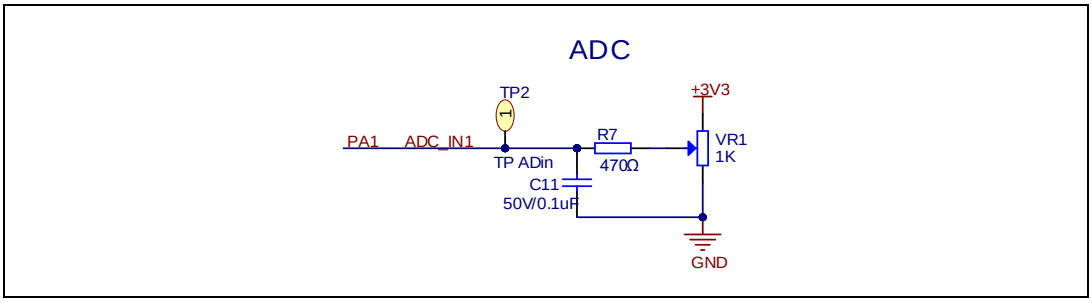
4.4. 按键

图4-4. 按键功能原理图



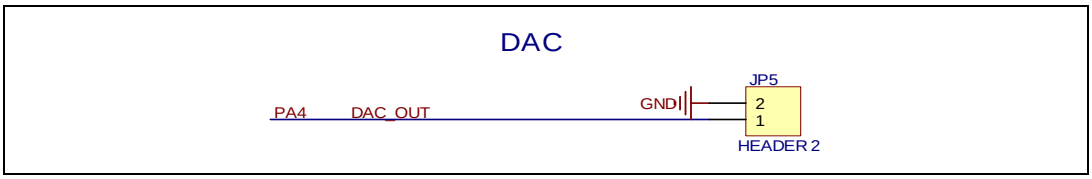
4.5. ADC

图4-5. ADC原理图



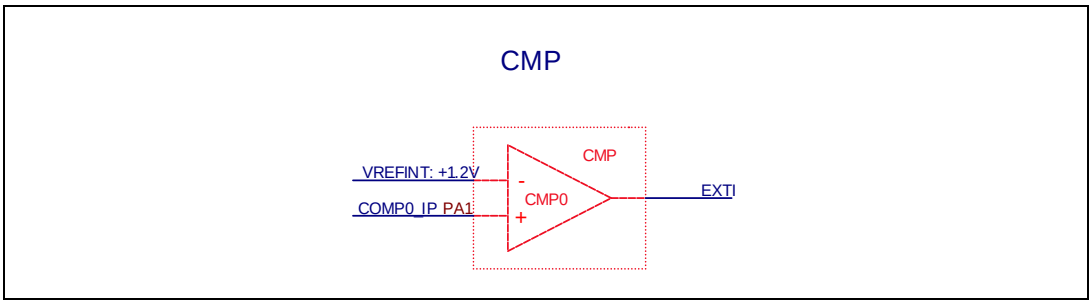
4.6. DAC

图4-6. DAC原理图



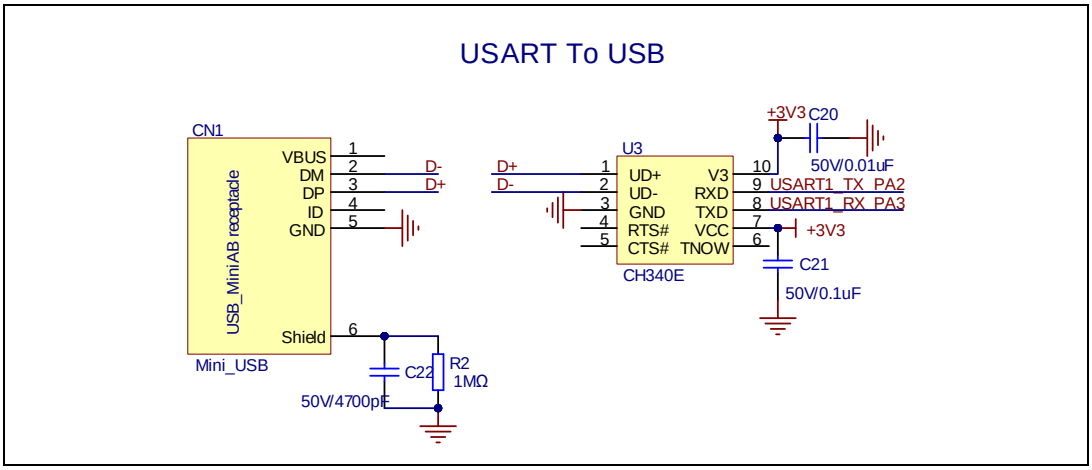
4.7. CMP

图4-7. CMP原理图



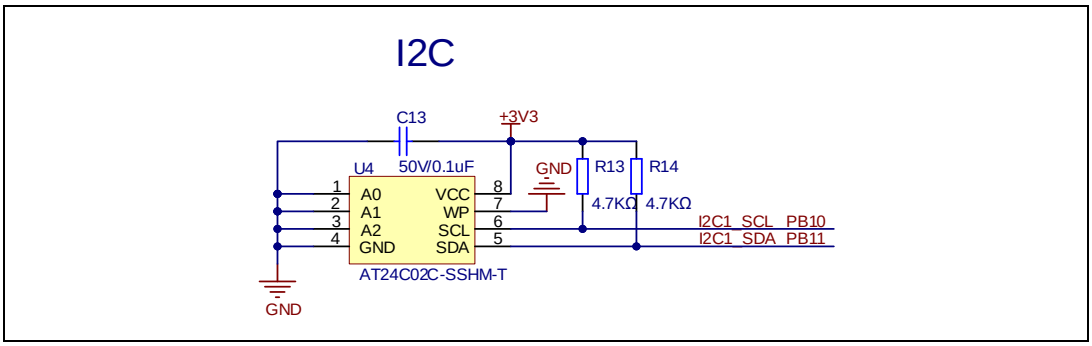
4.8. USART

图4-8. USART原理图



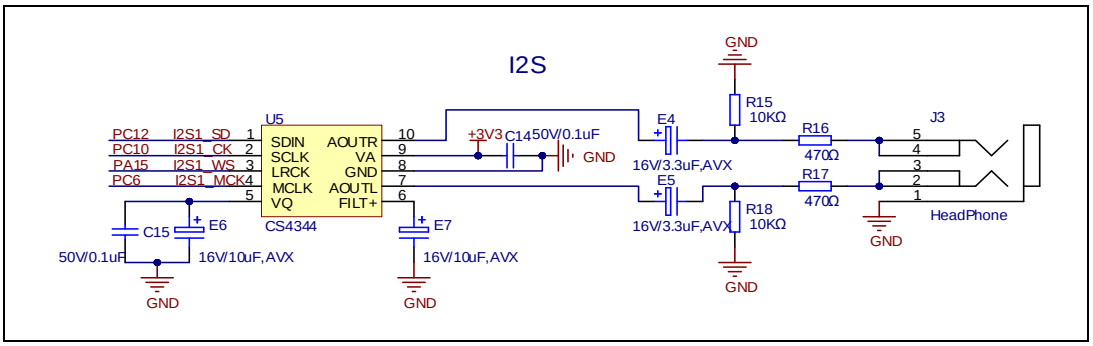
4.9. I2C

图4-9. I2C原理图



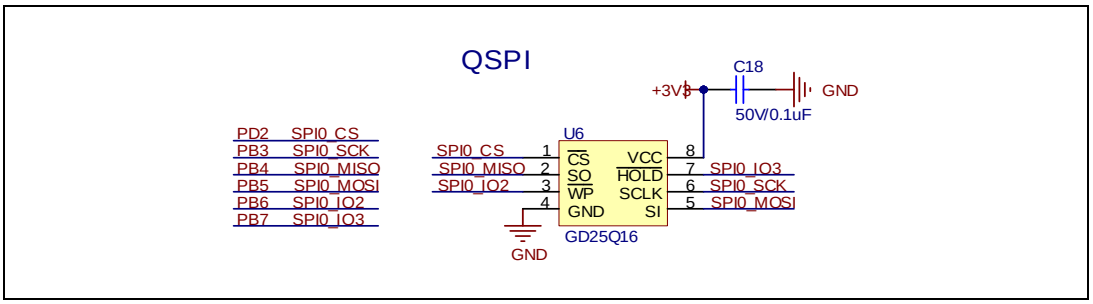
4.10. I2S

图4-10. I2S原理图



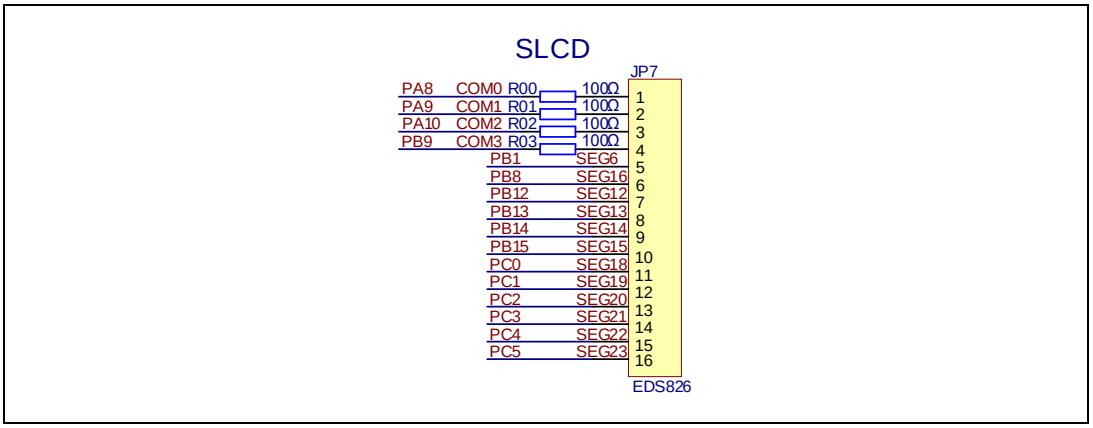
4.11. SPI

图4-11. SPI原理图



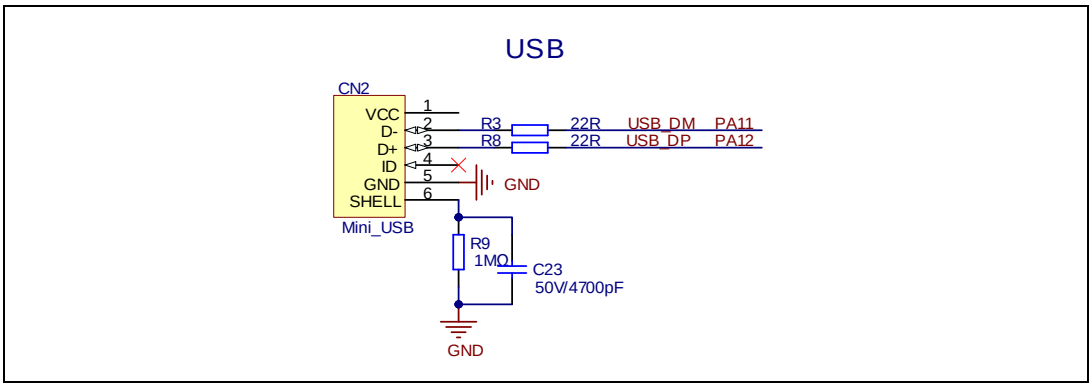
4.12. SLCD

图4-12. SLCD原理图



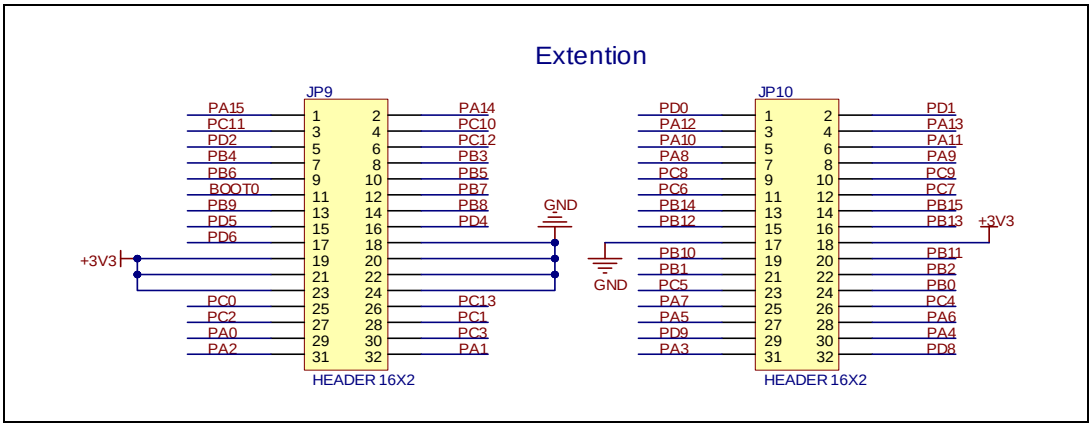
4.13. USB

图4-13. USB原理图



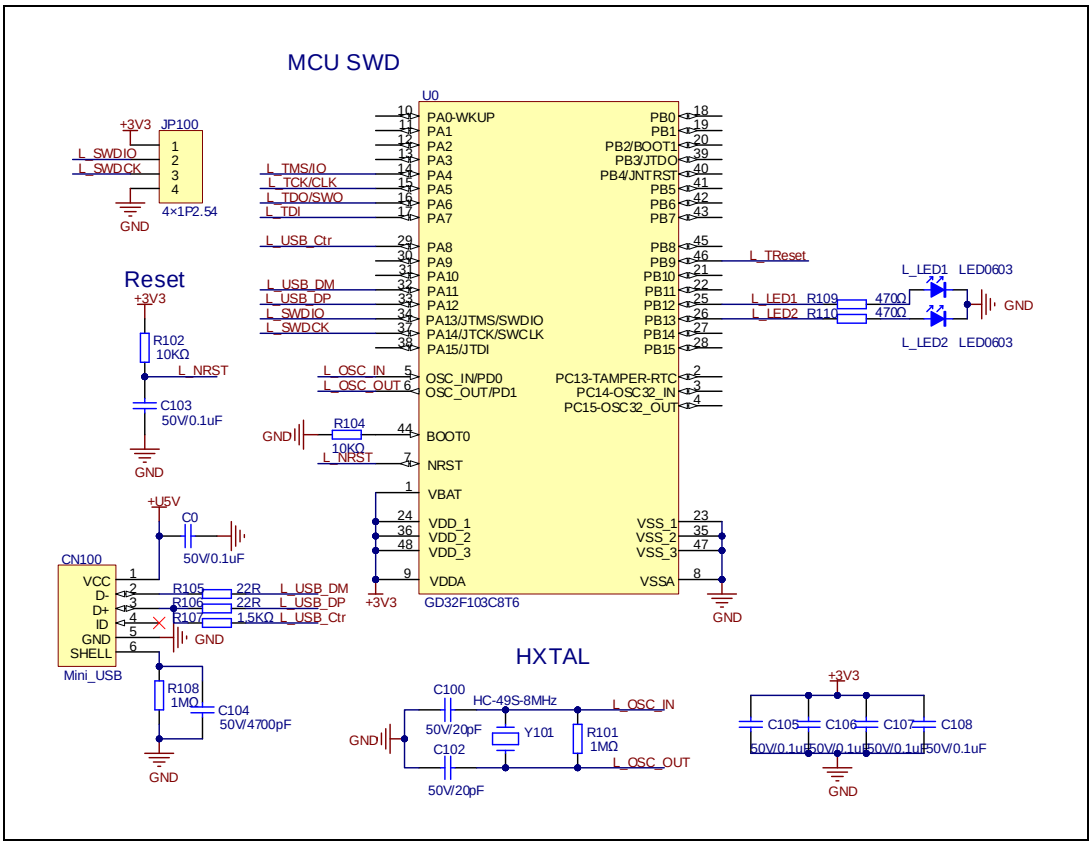
4.14. Extension

图4-14. Extension原理图



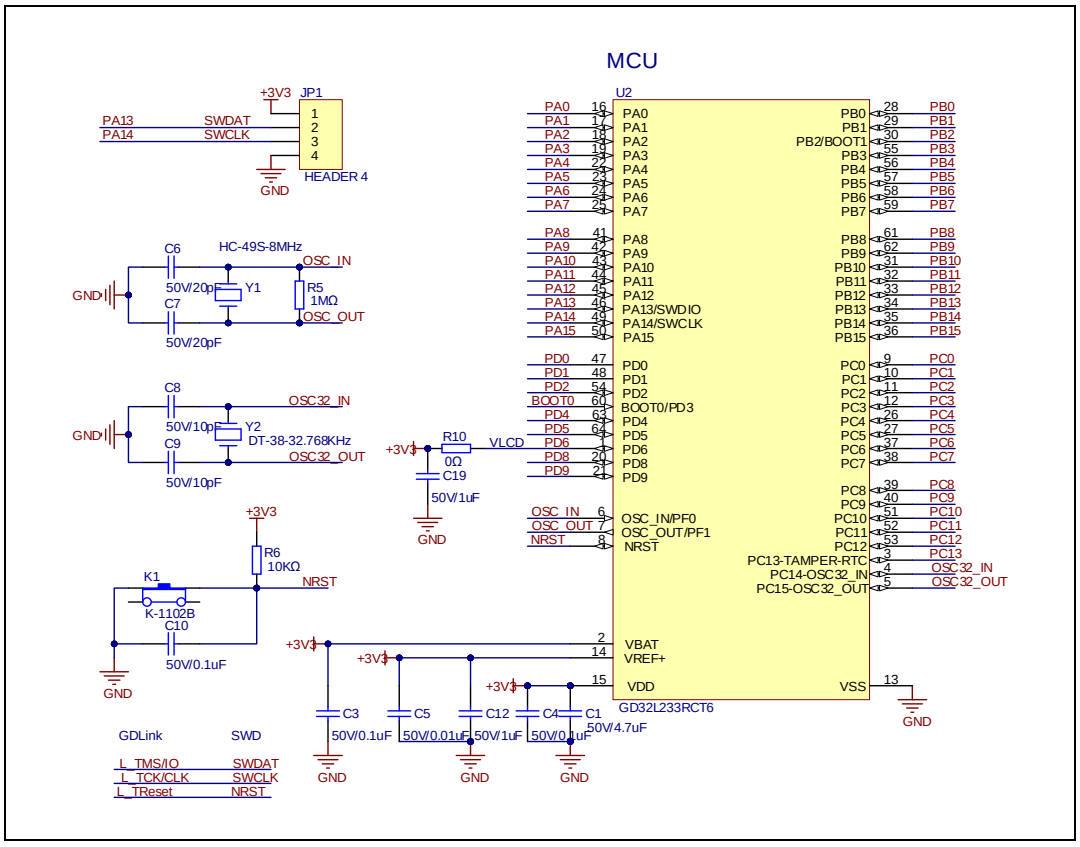
4.15. GD-Link

图4-15. GD-Link原理图



4.16. MCU

图4-16. MCU原理图



5. 例程使用指南

5.1. GPIO 流水灯

5.1.1. DEMO 目的

这个例程包括了 GD32 MCU 的以下功能：

- 学习使用 GPIO 控制 LED
- 学习使用 SysTick 产生 1ms 的延时

GD32L233R-EVAL 开发板上有 2 个用户按键和 4 个 LED。这些按键是 Tamper key 和 Wakeup key，所有 LED 通过 GPIO 控制。

这个例程将讲述怎么点亮这些 LED。

5.1.2. DEMO 执行结果

下载程序< 01_GPIO_Running_LED >到开发板上，LED 将被循环点亮。

5.2. GPIO 按键轮询模式

5.2.1. DEMO 目的

这个例程包括了 GD32 MCU 的以下功能：

- 学习使用 GPIO 控制 LED 和按键
- 学习使用 SysTick 产生 1ms 的延时

GD32L233R-EVAL 开发板上有 2 个用户按键和 4 个 LED。这些按键是 Tamper key 和 Wakeup key，所有 LED 通过 GPIO 控制。

这个例程讲述如何使用按键 Tamper key 控制 LED2。当按下 Tamper key，将检测 IO 端口的输入值，如果输入为低电平，将等待延时 100ms。之后，再次检测 IO 端口的输入状态。如果输入仍然为低电平，表明按键成功按下，翻转 LED2 的输出状态。

5.2.2. DEMO 执行结果

下载程序< 02_GPIO_Key_Polling_mode >到开发板上，按下 Tamper key，LED2 将会点亮，再次按下用 Tamper key，LED2 将会熄灭。

5.3. EXTI 按键中断模式

5.3.1. DEMO 目的

这个例程包括了 GD32 MCU 的以下功能：

- 学习使用 GPIO 控制 LED 和按键；
- 学习使用 EXTI 产生外部中断；

GD32L233R-EVAL 开发板有 2 个用户按键和 4 个 LED。这些按键是 Tamper 按键和 Wakeup 按键。LED1，LED2，LED3 和 LED4 可通过 GPIO 控制。

这个例程讲述如何使用 EXTI 外部中断线控制 LED2。当按下 Tamper 按键，将产生一个外部中断。在中断服务函数中，应用程序翻转 LED2 的输出状态。

5.3.2. DEMO 执行结果

下载程序< 03_EXTI_Key_Interrupt_mode >到开发板，LED2 亮灭一次用于测试。按下 Tamper 按键，LED2 将会点亮，再次按下 Tamper 按键，LED2 将会熄灭。

5.4. 串口打印

5.4.1. DEMO 目的

这个例程包括了 GD32 MCU 的以下功能：

- 学习使用 GPIO 控制 LED
- 学习将 C 库函数 Printf 重定向到 USART

5.4.2. DEMO 执行结果

下载程序< 04_USART_Printf >到开发板，将串口线连到开发板的 USART 上。首先，所有灯亮灭 2 次用于测试。然后 USART 将输出"USART printf example: please press the Tamper key"到超级终端。按下按键 Tamper 键，串口继续输出"USART printf example"。

超级终端输出的信息如下图所示：

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

USART printf example
```

5.5. 串口中断收发

5.5.1. DEMO 目的

这个例程包括了 GD32 MCU 的以下功能：

■ 学习使用串口发送和接收中断与超级终端之间的通信

5.5.2. DEMO 执行结果

下载程序< 05_USART_HyperTerminal_Interrupt >到开发板，将串口线连到开发板的 USART 上。首先，所有灯亮灭一次用于测试。然后 USART 将输出数组 tx_buffer 的内容(从 0x00 到 0xFF)到支持 hex 格式的超级终端并等待接收由超级终端发送的 BUFFER_SIZE 个字节的数据。MCU 将接收到的超级终端发来的数据存放在数组 rx_buffer 中。在发送和接收完成后，将比较 tx_buffer 和 rx_buffer 的值，如果结果相同，LED1，LED2，LED3，LED4 轮流闪烁；如果结果不相同，LED1，LED2，LED3，LED4 一起闪烁。

超级终端输出的信息如下图所示：

```
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. 串口 DMA 收发

5.6.1. DEMO 目的

这个例程包括了 GD32 MCU 的以下功能：

■ 学习使用串口 DMA 功能发送和接收

5.6.2. DEMO 执行结果

下载程序< 06_USART_DMA >到开发板，将串口线连到开发板的 USART 上。首先，USART 将输出"USART DMA interrupt receive and transmit example, please input 10 bytes:"并等待接收由超级终端发送 10 个字节的数据。MCU 接收到数据后，串口将接收到的数据继续输出到超级终端。

超级终端输出的信息如下图所示：

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



abcdefghij
```

5.7. LPUART Deepsleep 唤醒

5.7.1. DEMO 目的

这个例程包括了 GD32 MCU 的以下功能：

- 学习使用 LPUART 在 DEEPSLEEP 模式下唤醒 MCU

5.7.2. DEMO 执行结果

下载程序< 07_LPUART_Deepsleep_Wakeup >到开发板，将串口线连到开发板的 LPUART(USART)上。首先，LED1 闪烁两次后 MCU 进入 DEEPSLEEP 模式，LED1 停止闪烁。通过超级终端发送任意字符，MCU 被唤醒，LED1 保持闪烁。

5.8. ADC 温度传感器_内部参考电压

5.8.1. DEMO 目的

这个例程包括了 GD32 MCU 的以下功能：

- 学习使用 ADC 将模拟量转换成数字量
- 学习如何获取 ADC 内部通道 16(温度传感器通道)、内部通道 17(内部参考电压通道)

5.8.2. DEMO 执行结果

下载<08_ADC_Temperature_Vrefint>至 GD32L233R-EVAL 开发板并运行。将开发板的 USART1 口连接到电脑，打开电脑串口软件。

当程序运行时，串口软件会显示温度和内部参考电压值。

```
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
```

5.9. DAC 输出电压值

5.9.1. DEMO 目的

这个例程包括了 GD32 MCU 的以下功能:

■ 学习使用 DAC 在 DAC0_OUT0 输出端生成电压

5.9.2. DEMO 执行结果

下载程序<09_DAC_Output_Voltage_Value>至评估板并运行。

所有的 LED 灯先亮灭一次用于测试。数字量 0x7FF0，在 3.3V 的参考电压下，它的转换值应该为 1.65V(VREF/2)，将会在 PA4 引脚输出。
PA4 输出的电压可以通过示波器观测。

5.10. 比较器输出获取指示灯

5.10.1. DEMO 目的

该例程包含 GD32 MCU 以下功能:

■ 学会使用比较器输出比较结果

比较器有两个输入端，其中一个输入设置为 PA1，另一个是参考电压，比较这两个输入电压，输出高电平或低电平，然后 LED2 灯就会执行相应动作。

5.10.2. DEMO 执行结果

下载程序<10_Comparator_Obtain_Brightness>到开发板中，比较两个输入电压大小，如果输出电平为高，LED2 亮，否则，LED2 灭。

5.11. I2C 访问 EEPROM

5.11.1. DEMO 目的

这个例程包括了 GD32 MCU 的以下功能：

- 学习使用 I2C 模块的主机发送模式
- 学习使用 I2C 模块的主机接收模式
- 学习读写带有 I2C 接口的 EEPROM

5.11.2. DEMO 执行结果

下载程序<11_I2C_EEPROM >到开发板上。将开发板的 USART 口连接到电脑，通过超级终端显示打印信息。

程序首先从 0x00 地址顺序写入 256 字节的数据到 EEPROM 中，并打印写入的数据，然后程序又从 0x00 地址处顺序读出 256 字节的数据，最后比较写入的数据和读出的数据是否一致，如果一致，串口打印出"I2C-AT24C02 test passed!"，同时开发板上的四个 LED 灯开始顺序闪烁，否则串口打印出"Err:data read and write aren't matching."，同时四个 LED 全亮。

通过串口输出的信息如下图所示。

```
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!
```

5.12. QSPI FLASH

5.12.1. DEMO 目的

这个例程包括了 GD32 MCU 的以下功能:

■ 学习使用 SPI 模块的 SPI 四线模式读写带有 SPI 接口的 NOR Flash。

5.12.2. DEMO 执行结果

把电脑串口线连接到开发板的 COM 口,设置超级终端(HyperTerminal)软件波特率为 115200，数据位 8 位，停止位 1 位。

下载程序 <12_QSPI_FLASH> 到开发板上，通过超级终端可观察运行状况，会显示 FLASH

的 ID 号，写入和读出 FLASH 的 256 字节数据。然后比较写入的数据和读出的数据是否一致，如果一致，串口打印出"SPI-GD25Q16 Test Passed!"，否则，串口打印出"Err: Data Read and Write aren't Matching."。最后，四个 LED 灯依次循环点亮。

下图是实验结果图。

```
################################################################################
GD32L233R-EVAL System is Starting up...
GD32L233R-EVAL Flash:256K
GD32L233R-EVAL The CPU Unique Device ID:[FFFFFFFF-FFFFFFFF-FFFFFFFF]
GD32L233R-EVAL 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
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

Read from rx_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
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

SPI-GD25Q16 Test Passed!
```

5.13. I2S 音频播放

5.13.1. DEMO 目的

这个例程包括了 GD32 MCU 的以下功能：

- 学习使用 I2S 接口输出音频文件
- 解析 wav 音频文件的格式

GD32L233R-EVAL 开发板集成了 I2S 模块，该模块可以和外部设备通过音频协议通信。这个例程演示了如何通过开发板的 I2S 接口播放音频文件。

5.13.2. DEMO 执行结果

下载程序<13_I2S_Audio_Player>到开发板并运行，插上耳机可听到播放的音频文件声音。

5.14. TRNG 随机数

5.14.1. DEMO 目的

这个例程包括了 GD32 MCU 的以下功能：

- 学习使用 TRNG 模块生成随机数
- 学习使用 USART 模块与电脑进行通讯

5.14.2. DEMO 执行结果

下载程序<14_TRNG_Get_Random>到开发板上并运行。将开发板的 USART 连接到电脑，打开支持 hex 格式的串口助手。当程序运行时，串口助手将显示初始信息。通过串口助手输入期望的最小值与最大值(如最小值为 0x011，最大值为 0x33)，之后会自动生成输入范围内的随机数并通过串口助手显示。

串口输出如下图所示：

```
/==========Gigadevice TRNG test==========/
TRNG init ok
Please input min num (hex format, the range is 0~0xFF):
The input min num is 0x11
Please input max num hex format, the range is 0~0xFF):
The input max num is 0x33
Generate random num1 is 0x15
Generate random num2 is 0x27
Please input min num (hex format, the range is 0~0xFF):
```

5.15. CAU 加密处理器

5.15.1. DEMO 目的

这个例程包括了 GD32 MCU 的以下功能：

- 学习 DES，TDES，AES 算法；
- 学习电子密码本(ECB)，密码块链接(CBC)，计数器(CTR)模式，伽罗瓦/计数器(GCM)模式，复合密码机(CCM)模式，密码反馈(CFB)模式，和输出反馈(OFB)模式；
- 学习使用 CAU 模块进行加密和解密；
- 学习使用 USART 模块与电脑进行通讯。

5.15.2. DEMO 执行结果

下载程序<15_CAU>到开发板上并运行。当程序运行时，串口助手将显示如下图所示信息。分别是用于测试的明文数据值，可以选择的加密算法，以及算法模式。用户按照串口输出信息指示进行算法设置后，串口会打印出所选择的算法和模式，如下图所示。

```
Plain data :
0x6B 0xC1 0xBE 0xE2 0x2E 0x40 0x9F 0x96 0xE9 0x3D 0x7E 0x11 0x73 0x93 0x17 0x2A   [Block 0]
0xAE 0x2D 0x8A 0x57 0x1E 0x03 0xAC 0x9C 0x9E 0xB7 0x6F 0xAC 0x45 0xAF 0x8E 0x51   [Block 1]
0x30 0xC8 0x1C 0x46 0xA3 0x5C 0xE4 0x11 0xE5 0xFB 0xC1 0x19 0x1A 0x0A 0x52 0xEF   [Block 2]
0xF6 0x9F 0x24 0x45 0xDF 0x4F 0x9B 0x17 0xAD 0x2B 0x41 0x7B 0xE6 0x6C 0x37 0x10   [Block 3]
========Choose CAU algorithm=======
1: DES algorithm
2: TDES algorithm
3: AES algorithm


You choose to use DES algorithm
========Choose CAU mode========
1: ECB mode
2: CBC mode
3: CTR mode only when choose AES algorithm
4: GCM mode only when choose AES algorithm
5: CCM mode only when choose AES algorithm
6: CFB mode only when choose AES algorithm
7: OFB mode only when choose AES algorithm


You choose to use ECB mode
```

选择完成后，程序开始进行加解密操作，将结果通过串口打印。

```
 Encrypted data with DES  Mode  ECB :

0x6E 0xDF 0xD1 0xB7 0xA0 0x01 0xCD 0x17 0xCD 0xC5 0x7F 0xF7 0x9C 0xF8 0x72 0xD0   [Block 0]
0x11 0x97 0xA6 0xD2 0x13 0x59 0x4F 0x7A 0x3D 0x7C 0x7C 0xEC 0xBC 0xDD 0xD2 0x20   [Block 1]
0x3A 0x75 0x8B 0x06 0x75 0x2E 0x18 0x0D 0x55 0x0F 0xDD 0x57 0x5A 0xF1 0x3B 0x94   [Block 2]
0x18 0x3D 0x4D 0xA1 0x1E 0x14 0x75 0x6B 0x0F 0xD9 0xD9 0x64 0x16 0xA0 0x60 0x14   [Block 3]

 Decrypted data with DES  Mode  ECB :

0x6B 0xC1 0xBE 0xE2 0x2E 0x40 0x9F 0x96 0xE9 0x3D 0x7E 0x11 0x73 0x93 0x17 0x2A   [Block 0]
0xAE 0x2D 0x8A 0x57 0x1E 0x03 0xAC 0x9C 0x9E 0xB7 0x6F 0xAC 0x45 0xAF 0x8E 0x51   [Block 1]
0x30 0xC8 0x1C 0x46 0xA3 0x5C 0xE4 0x11 0xE5 0xFB 0xC1 0x19 0x1A 0x0A 0x52 0xEF   [Block 2]
0xF6 0x9F 0x24 0x45 0xDF 0x4F 0x9B 0x17 0xAD 0x2B 0x41 0x7B 0xE6 0x6C 0x37 0x10   [Block 3]

 Example restarted...
```

之后重新回到开始界面供用户选择其他算法及模式观察 Demo 结果。如下图所示。

```
Plain data :
0x6B 0xC1 0xBE 0xE2 0x2E 0x40 0x9F 0x96 0xE9 0x3D 0x7E 0x11 0x73 0x93 0x17 0x2A   [Block 0]
0xAE 0x2D 0x8A 0x57 0x1E 0x03 0xAC 0x9C 0x9E 0xB7 0x6F 0xAC 0x45 0xAF 0x8E 0x51   [Block 1]
0x30 0xC8 0x1C 0x46 0xA3 0x5C 0xE4 0x11 0xE5 0xFB 0xC1 0x19 0x1A 0x0A 0x52 0xEF   [Block 2]
0xF6 0x9F 0x24 0x45 0xDF 0x4F 0x9B 0x17 0xAD 0x2B 0x41 0x7B 0xE6 0x6C 0x37 0x10   [Block 3]
========Choose CAU algorithm=======
1: DES algorithm
2: TDES algorithm
3: AES algorithm
```

5.16. RCU 时钟输出

5.16.1. DEMO 目的

这个例程包括了 GD32 MCU 的以下功能:

- 学习使用 GPIO 控制 LED
- 学习使用 RCU 模块的时钟输出功能
- 学习使用 USART 模块与电脑进行通讯

5.16.2. DEMO 执行结果

下载程序<16_RCU_Clock_Out>到开发板上并运行。将开发板的 USART 口连接到电脑，打开超级终端。当程序运行时，超级终端将显示初始信息。之后通过按下 TAMPER 按键可以选择输出时钟的类型，对应的 LED 灯会被点亮，并在超级终端显示选择的模式类型。测量 PA8 引脚，可以通过示波器观测输出时钟的频率。

串口输出如下图所示:

```
/========== Gigadevice Clock Output Demo ==========/
press tamper key to select clock output source
CK_OUT: system clock, DIV: 4
CK_OUT: IRC16M, DIV: 1
CK_OUT: IRC48M, DIV: 1
CK_OUT: IRC32K, DIV: 1
CK_OUT: LXTAL, DIV: 1
CK_OUT: HXTAL, DIV: 1
CK_OUT: PLL/2, DIV: 1
```

5.17. CTC 校准

5.17.1. DEMO 目的

这个例程包括了 GD32 MCU 的以下功能:

- 学习使用外部晶振 LXTAL 来实现 CTC 校准功能
- 学习使用 CTC 校准控制器校准内部 48MHz RC 振荡器时

CTC 单元基于外部精确的参考信号源来校准内部 48MHz RC 振荡器。它可以自动调整校准值，以提供精确的 IRC48M 时钟。

5.17.2. DEMO 执行结果

下载程序<17_CTC_Calibration>到开发板上，运行程序。如果内部 48MHz RC 校准成功，LED2 将会点亮。否则，LED2 灯熄灭。

5.18. PMU 睡眠模式唤醒

5.18.1. DEMO 目的

这个例程包括了 GD32 MCU 的以下功能：

- 学习使用串口接收中断唤醒 PMU 睡眠模式

5.18.2. DEMO 执行结果

下载程序<18_PMU_Sleep_Wakeup>到开发板上，并将串口线连到开发板的 USART 上。板子上电后，所有 LED 都熄灭。MCU 将进入睡眠模式同时软件停止运行。当从超级终端接收到一个字节数据时，MCU 将被 USART 接收中断唤醒。所有的 LED 灯同时闪烁。

5.19. RTC 日历

5.19.1. DEMO 目的

这个例程包括了 GD32 MCU 的以下功能：

- 学习使用 RTC 模块实现日历功能
- 学习使用 USART 模块实现时间显示

5.19.2. DEMO 执行结果

下载程序<19_RTC_Calendar>到开发板上，使用串口线连接电脑到开发板 USART 接口，打开串口助手软件。在开发板上电后，程序需要请求通过串口助手设置时间。日历会显示在串口助手上。

```
****************** RTC calendar demo ******************
=======Configure RTC Time========
 please set the last two digits of current year:
 2021
 please input month:

 08
 please input day:

 12
 please input hour:

 12
 please input minute:

 12
 please input second:

 12

** RTC time configuration success! **
Current time: 2021-08-12 : 12:12:12
Current time: 2021-08-12 : 12:12:12
```

5.20. TIMER 呼吸灯

5.20.1. DEMO 目的

这个例程包括了 GD32 MCU 的以下功能:

- 学习使用 TIMER 输出 PWM 波
- 学习更新 TIMER 通道寄存器的值

5.20.2. DEMO 执行结果

下载程序<20_TIMER_Breath_LED>到 GD32L233R-EVAL 开发板，并运行程序。

当程序运行时，可以看到 LED1 由暗变亮，由亮变暗，往复循环，就像人的呼吸一样有节奏。

5.21. LPTIMER 睡眠模式的 PWM 输出

5.21.1. DEMO 目的

这个例程包括了 GD32 MCU 的以下功能：

- 学习使用 LPTIMER 输出 PWM 波
- 学习使用 LPTIMER 中断唤醒 PMU 睡眠模式

5.21.2. DEMO 执行结果

下载程序<21_LPTIMER_Deepsleep_Pwmout>到 GD32L233R-EVAL 开发板上，并运行程序。当程序运行时，LED1 闪烁，LPTIMER_O(PA4)输出 PWM 波形。

按下 Wakeup(K3)，MCU 进入睡眠模式，LED1 停在一个固定状态(点亮或熄灭)。当 LPTIMER 的计数器值匹配比较寄存器值或自动重载寄存器值时，MCU 将被 LPTIMER 中断唤醒，LED1 恢复闪烁。在此期间，LPTIMER_O(PA4)一直输出 PWM 波形。

5.22. SLCD 液晶显示

5.22.1. DEMO 目的

这个例程包括了 GD32 MCU 的以下功能：

- 学习使用 SLCD 模块实现显示数字的功能。

5.22.2. DEMO 执行结果

下载<22_SLCD_Glass>程序到开发板上并运行。当程序运行时，段码 LCD 会显示按秒累加的数字。

5.23. USBD 键盘

5.23.1. DEMO 目的

这个例程包括了 GD32 MCU 的以下功能：

- 学习如何使用 USBD 的设备模式
- 学习如何实现 USB HID(人机接口)设备

在本例程中，GD32L233R-EVAL 开发板被 USB 主机利用内部 HID 驱动枚举为一个 USB 键盘，如下图所示,按下 Tamper 键和 Wakeup 键可依次输出两个字符('a'，'b')。另外，本例程支持 USB 键盘远程唤醒主机，其中字符'b'按键被作为唤醒源。



5.23.2. DEMO 执行结果

将<23_USBD_Keyboard>例程下载到开发板中，并运行。按下 Tamper 键，输出'a'；按下 Wakeup 键，输出'b'。

可利用以下步骤所说明的方法验证 USB 远程唤醒的功能：

- 手动将 PC 机切换到睡眠模式；
- 等待主机完全进入睡眠模式；
- 按下 Wakeup 按键；
- 如果 PC 被唤醒，表明 USB 远程唤醒功能正常，否则失败。

5.24. USBD 虚拟串口

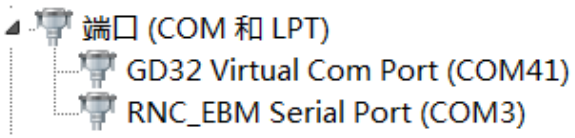
5.24.1. DEMO 目的

这个例程包括了 GD32 MCU 的以下功能：

- 学习如何使用 USBD 设备

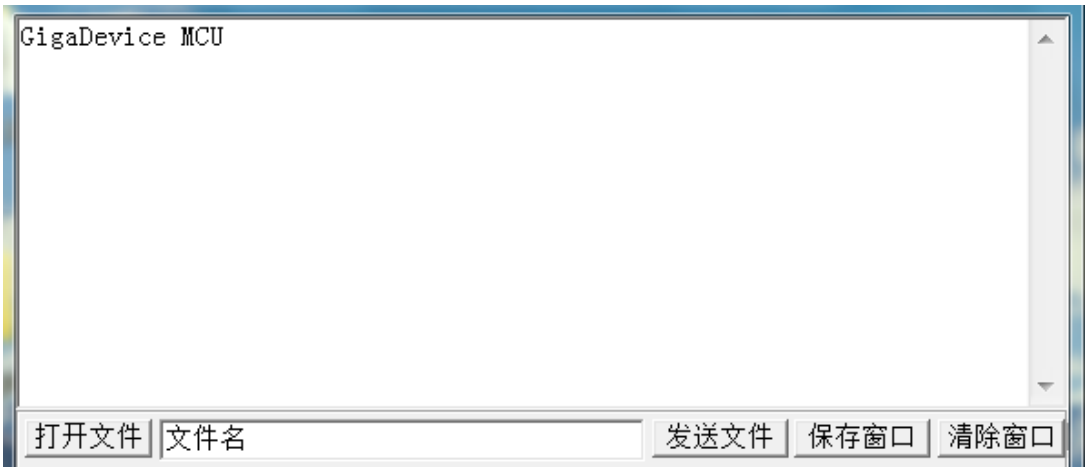
■ 学习如何实现 USB CDC 设备

GD32L233R-EVAL 开发板具有一个 USBD 接口。在本例程中，GD32L233R-EVAL 开发板被 USB 主机枚举为一个 USB 虚拟串口,如下图所示,可在 PC 端设备管理器中看到该虚拟串口。该例程使得 USB 设备看起来像是个串口，也可以通过 USB 口回传数据。通过键盘输入某些信息，虚拟串口可以接收并显示这些信息。



5.24.2. DEMO 执行结果

将<24_USBD_CDC_ACM>例程下载到开发板中，并运行。通过键盘输入某些数据，虚拟串口可以接收并显示这些数据。比如通过虚拟串口的输入框输入"GigaDevice MCU"，PC 回传这些信息给虚拟串口，并得以显示。



6. 版本历史

表 6-1. 版本历史

版本号	说明	日期
1.0	初稿发布	2021 年 08 月 15 日
1.1	更新格式	2021 年 11 月 15 日
1.2	更新部分图格式	2023 年 07 月 18 日
1.3	1. 修改 ***5.24.1*** 章节中内容，将描述"USB 键盘看起来像是个串口"改为"USB 设备看起来像是个串口"。	2023 年 11 月 16 日
1.4	1. 更新全文显示字体	2026 年 8 月 21

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.

© 2026 GigaDevice Semiconductor Inc. - All rights reserved