GigaDevice Semiconductor Inc.

GD32403Z-EVAL 评估板

用户指南

V2.0

目录

目录 1

图 5

表 6

1. 简介 7

2. 功能引脚分配 7

3. 入门指南 9

4. 硬件设计概述 10

4.1. 供电电源 10

4.2. 启动方式选择 10

4.3. LED 指示灯 11

4.4. 按键 11

4.5. 串口 12

4.6. 模数转换器 12

4.7. 数模转换器 12

4.8. I2S 13

4.9. I2C 13

4.10. SPI 14

4.11. CAN 14

4.12. SDIO 15

4.13. NAND 15

4.14. LCD 16
4.15. USBFS 17
4.16. 扩展电路 17
4.17. GD-Link 18
5. 例程使用指南 19
5.1. GPIO 流水灯 19
5.1.1. DEMO 目的 19
5.1.2. DEMO 执行结果 19
5.2. GPIO 按键轮询模式 19
5.2.1. DEMO 目的 19
5.2.2. DEMO 执行结果 20
5.3. EXTI 按键中断模式 20
5.3.1. DEMO 目的 20
5.3.2. DEMO 执行结果 20
5.4. 串口打印 21
5.4.1. DEMO 目的 21
5.4.2. DEMO 执行结果 21
5.5. 串口中断收发 21
5.5.1. DEMO 目的 21
5.5.2. DEMO 执行结果 21
5.6. 串口 DMA 收发 22
5.6.1. DEMO 目的 22
5.6.2. DEMO 执行结果 22
5.7. ADC 温度传感器_Vrefint 23
5.7.1. DEMO 目的 23
5.7.2. DEMO 执行结果 23
5.8. ADC0 和 ADC1 跟随模式 24
5.8.1. DEMO 目的 24
5.8.2. DEMO 执行结果 24

5.9. ADC0 和 ADC1 常规并行模式 25
5.9.1. DEMO 目的 25
5.9.2. DEMO 执行结果 25
5.10. DAC 输出电压值 26
5.10.1. DEMO 目的 26
5.10.2. DEMO 执行结果 26
5.11. I2C 访问 EEPROM 27
5.11.1. DEMO 目的 27
5.11.2. DEMO 执行结果 27
5.12. SPI FLASH 29
5.12.1. DEMO 目的 29
5.12.2. DEMO 执行结果 29
5.13. I2S 音频播放 30
5.13.1. DEMO 目的 30
5.13.2. DEMO 执行结果 31
5.14. NAND 存储器 31
5.14.1. DEMO 目的 31
5.14.2. DEMO 执行结果 31
5.15. LCD 触摸屏 32
5.15.1. DEMO 目的 32
5.15.2. DEMO 执行结果 32
5.16. SD 卡测试 33
5.16.1. DEMO 目的 33
5.16.2. DEMO 执行结果 33
5.17. CAN 网络通信 34
5.17.1. DEMO 目的 34
5.17.2. DEMO 执行结果 34
5.18. RCU 时钟输出 35
5.18.1. DEMO 目的 35
5.18.2. DEMO 执行结果 35

5.19. CTC 校准 36
5.19.1. DEMO 目的 36
5.19.2. DEMO 执行结果 36
5.20. PMU 睡眠模式唤醒 36
5.20.1. DEMO 目的 36
5.20.2. DEMO 执行结果 36
5.21. RTC 日历 37
5.21.1. DEMO 目的 37
5.21.2. DEMO 执行结果 37
5.22. 呼吸灯 38
5.22.1. DEMO 目的 38
5.22.2. DEMO 执行结果 38
5.23. USB 设备 38
5.23.1. HID_键盘 38
5.23.2. MSC_U 盘 39
5.24. USB 主机 41
5.24.1. HID_Host(HID 主机) 41
5.24.2. MSC_Host(MSC 主机) 43
6. 版本历史 43

图

图 4-1 供电电源原理图 10
图 4-2 启动方式选择原理图 10
图 4-3 LED 功能原理图 11
图 4-4 按键功能原理图 11
图 4-5 串口 0 功能原理图 12
图 4-6 模数转换器功能原理图 12
图 4-7 数模转换器功能原理图 12
图 4-8 I2S 功能原理图 13
图 4-9 I2C 功能原理图 13
图 4-10 SPI 功能原理图 14
图 4-11 CAN 功能原理图 14
图 4-12 SDIO 功能原理图 15
图 4-13 NAND 存储器功能原理图 15
图 4-14 LCD 功能原理图 16
图 4-15 USBFS 功能原理图 17
图 4-16 扩展电路原理图 17
图 4-17 GD-Link 原理图 18

表

表 2-1 引脚分配 7
表 6-1 版本历史 43

1. 简介

GD32403Z-EVAL 评估板使用 GD32F403ZET6 作为主控制器。评估板使用 Mini USB 接口或者 DC-005 连接器提供 5V 电源。提供包括扩展引脚在内的及 SWD, Reset, Boot, User button key, LED, CAN, I2C, I2S, USART, RTC, LCD, SPI, ADC, DAC, EXMC, CTC, SDIO, USBFS, GD-Link 等外设资源。更多关于开发板的资料可以查看 GD32403Z-EVAL-V1.1 原理图。

2. 功能引脚分配

表 2-1 引脚分配

功能	引脚	描述
LED	PF0	LED2
	PF1	LED3
	PF2	LED4
	PF3	LED5
RESET		K1-Reset
KEY	PA0	K2-Wakeup
	PC13	K3-Tamper
	PF5	K4-User key1
	PF4	K5-User key2
USART0	PA9	USART0_TX
	PA10	USART0_RX
USART1	PA2	USART1_TX
	PA3	USART1_RX
ADC	PC3	ADC012_IN13
DAC	PA4	DAC_OUT0
	PA5	DAC_OUT1
I2C	PB6	I2C0_SCL
	PB7	I2C0_SDA
SPI	PA5	SPI0_SCK
	PA6	SPI0_MISO
	PA7	SPI0_MOSI
	PE3	SPI0_CS
I2S	PA4	MSEL
	PA5	MCLK
	PA7	MDIN

	PB12	I2S_WS
	PB13	I2S_CK
	PB15	I2S_DIN
	PC6	I2S_MCK
CAN	PD0	CAN0_RX
	PD1	CAN0_TX
SDIO	PD2	SDIO_CMD
	PC12	SDIO_CLK
	PC8	SDIO_DAT0
	PC9	SDIO_DAT1
	PC10	SDIO_DAT2
	PC11	SDIO_DAT3
NAND Flash	PD14	EXMC_D0
	PD15	EXMC_D1
	PD0	EXMC_D2
	PD1	EXMC_D3
	PE7	EXMC_D4
	PE8	EXMC_D5
	PE9	EXMC_D6
	PE10	EXMC_D7
	PD11	EXMC_A16
	PD12	EXMC_A17
	PD4	EXMC_NOE
	PD5	EXMC_NWE
	PD6	EXMC_NWAIT
	PG6	EXMC_INT
	PD7	EXMC_NCE1
LCD	PD14	EXMC_D0
	PD15	EXMC_D1
	PD0	EXMC_D2
	PD1	EXMC_D3
	PE7	EXMC_D4
	PE8	EXMC_D5
	PE9	EXMC_D6
	PE10	EXMC_D7
	PE11	EXMC_D8
	PE12	EXMC_D9
	PE13	EXMC_D10
	PE14	EXMC_D11
	PE15	EXMC_D12
	PD8	EXMC_D13
	PD9	EXMC_D14

	PD10	EXMC_D15
	PE2	EXMC_A23
	PD4	EXMC_NOE
	PD5	EXMC_NWE
	PG9	EXMC_NE1
USBFS	PA9	USB_VBUS
	PA11	USB_DM
	PA12	USB_DP
	PA10	USB_ID

3. 入门指南

评估板使用 Mini USB 或者 DC-005 连接器提供 5V 电源。下载程序到评估板需要一套 J-Link 或者使用 GD-Link 工具，在选择了正确的启动方式并且上电后，LED1 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 4.74 uVision4 创建的，IAR 版的工程是基于 IAR Embedded Workbench for ARM 7.40.2 创建的。在使用过程中有如下几点需要注意：

1、如果使用 Keil uVision4 打开工程，安装 GD32F4xx_Addon.2.0.0.exe，以加载相关文件；

2、如果使用 Keil uVision5 打开工程，有两种方法解决"Missing Device(s)"问题。第一种是方法先安装 GigaDevice.GD32F4xx_DFP.2.0.0.pack，在 Project 菜单中选择 Manage 子菜单，点击 Migrate to Version 5 Format...菜单，将 Keil uVision4 工程转为 Keil uVision5 工程，同时在 Option for Target 的 C/C++ 中添加路径 C:\Keil_v5\ARM\Pack\ARM\CMSIS\4.2.0\CMSIS\Include；第二种方法是直接安装 Addon，在 Folder Selection 中的 Destination Folder 那一栏选择 Keil uVision5 软件的安装目录，如 C:\Keil_v5，然后在 Option for Target 的 Device 选择对应的器件，同时在 Option for

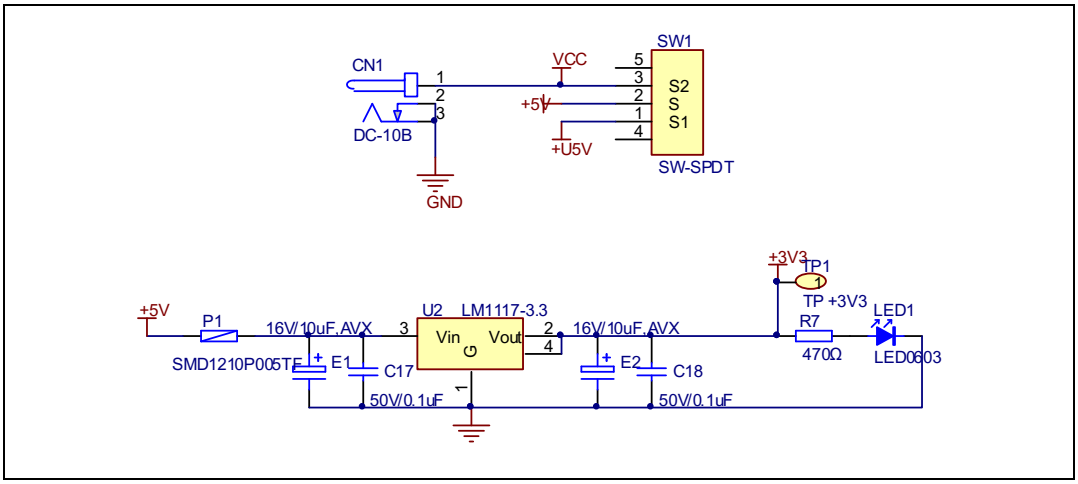
Target 的 C/C++中添加路径 C:\Keil_v5\ARM\Pack\ARM\CMSIS\4.2.0\CMSIS\Include。

3、如果使用 IAR 打开工程，安装 IAR_GD32F4xx_ADDON.2.0.0.exe，以加载相关文件。

4. 硬件设计概述

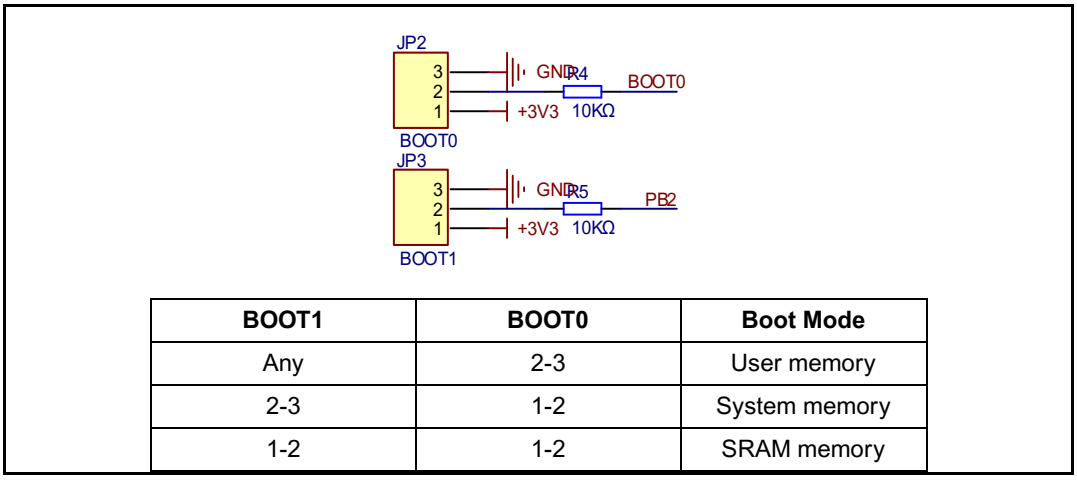
4.1. 供电电源

图 4-1 供电电源原理图



4.2. 启动方式选择

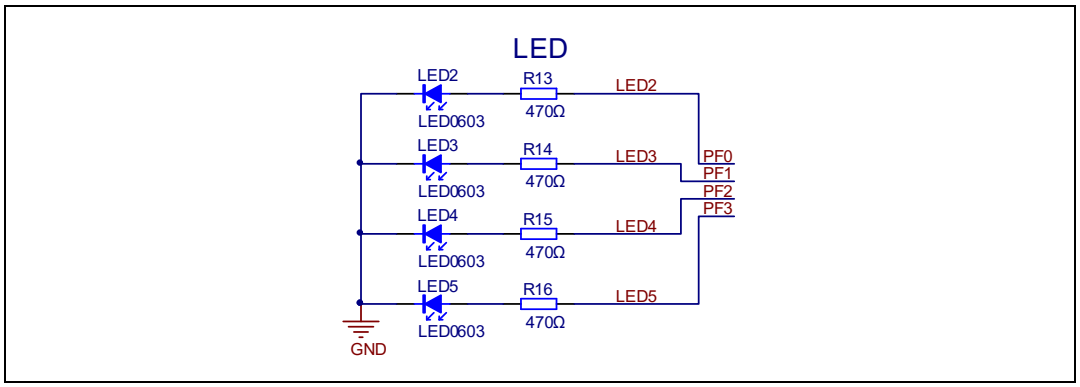
图 4-2 启动方式选择原理图



BOOT1	BOOT0	Boot Mode
Any	2-3	User memory
2-3	1-2	System memory
1-2	1-2	SRAM memory

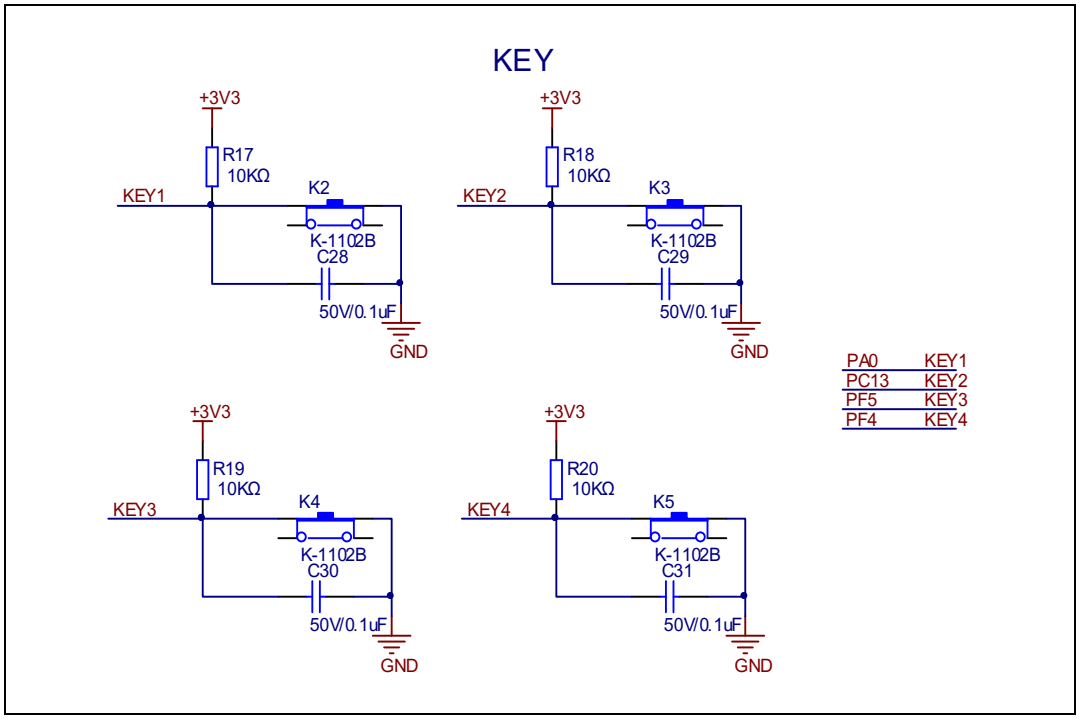
4.3. LED 指示灯

图 4-3 LED功能原理图



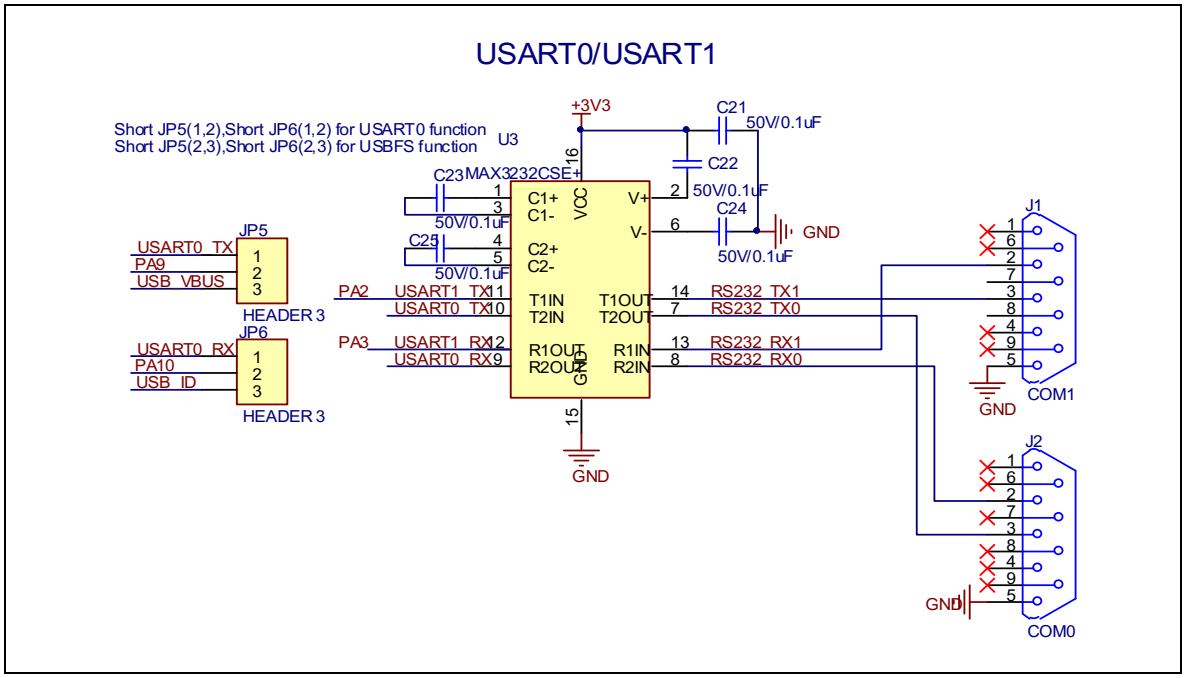
4.4. 按键

图 4-4 按键功能原理图



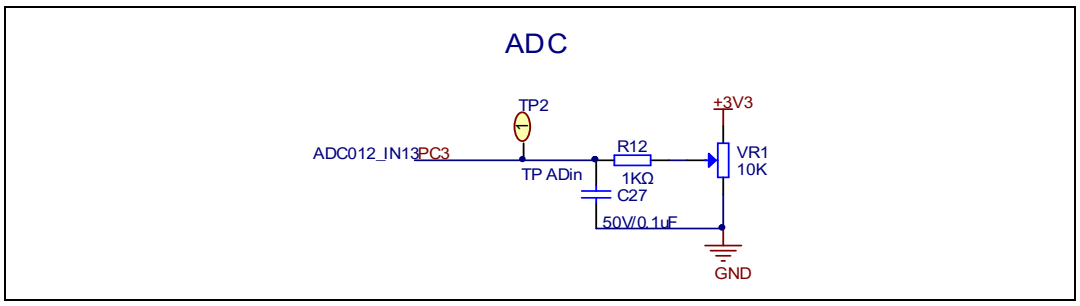
4.5. 串口

图 4-5 串口功能原理图



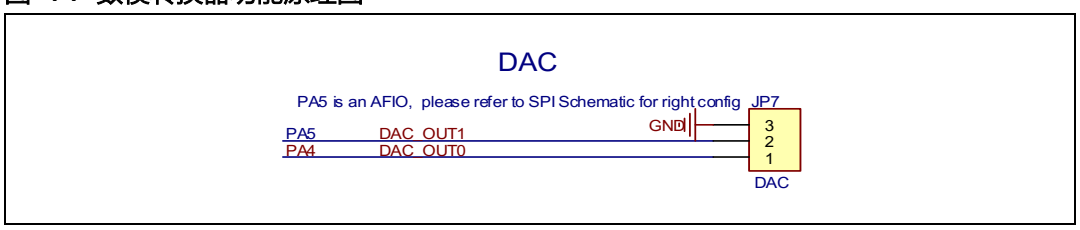
4.6. 模数转换器

图 4-6 模数转换器功能原理图



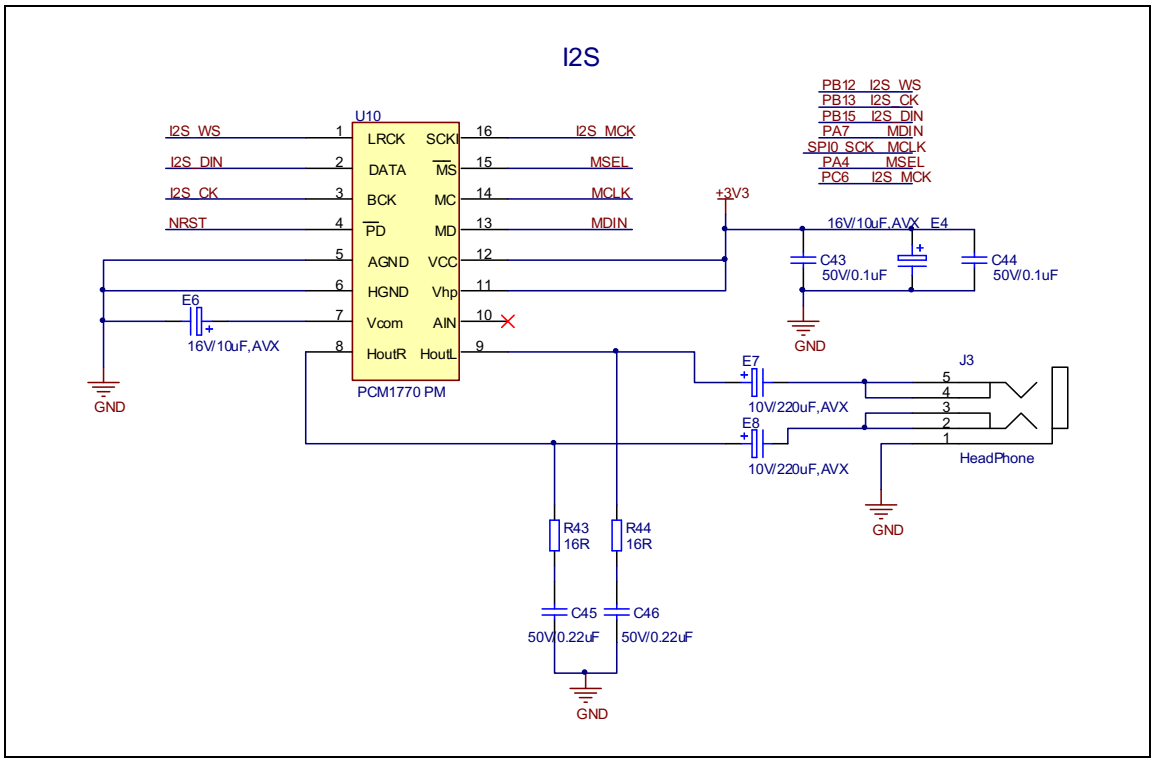
4.7. 数模转换器

图 4-7 数模转换器功能原理图



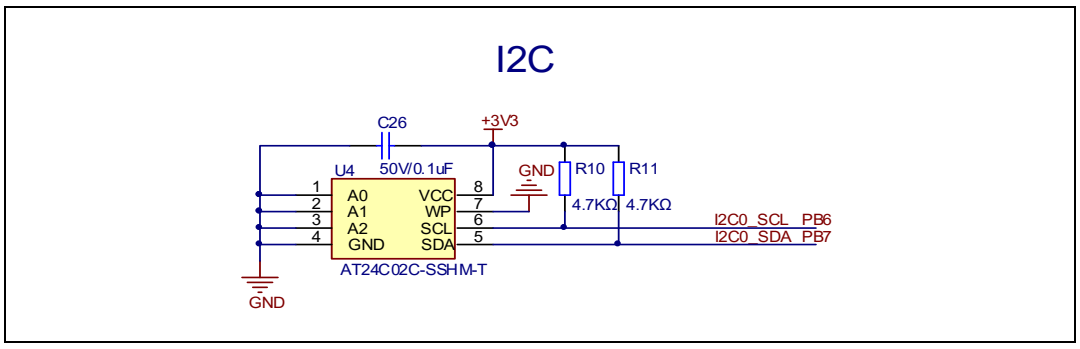
4.8. I2S

图 4-8 I2S功能原理图



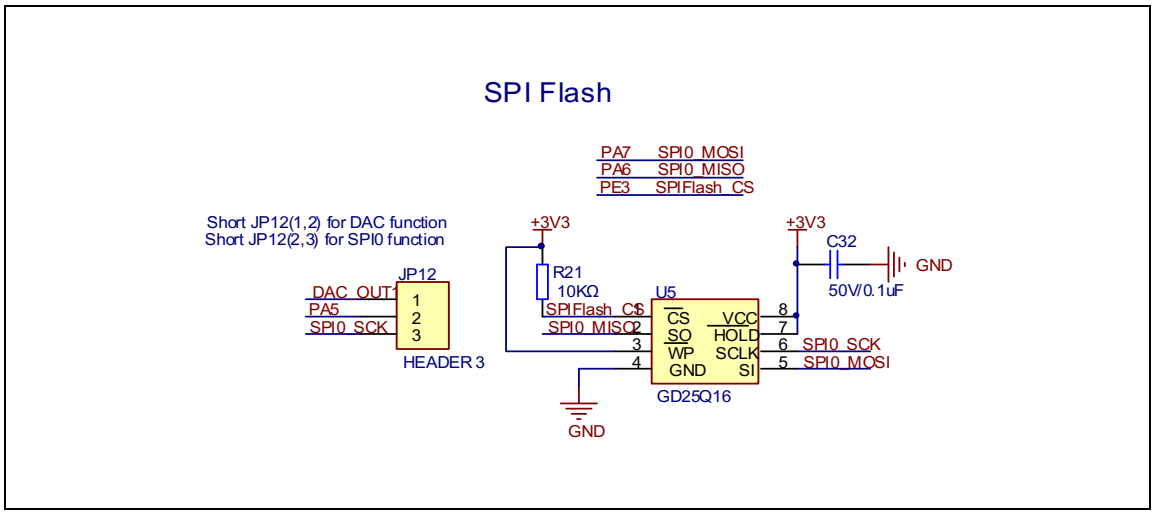
4.9. I2C

图 4-9 I2C功能原理图



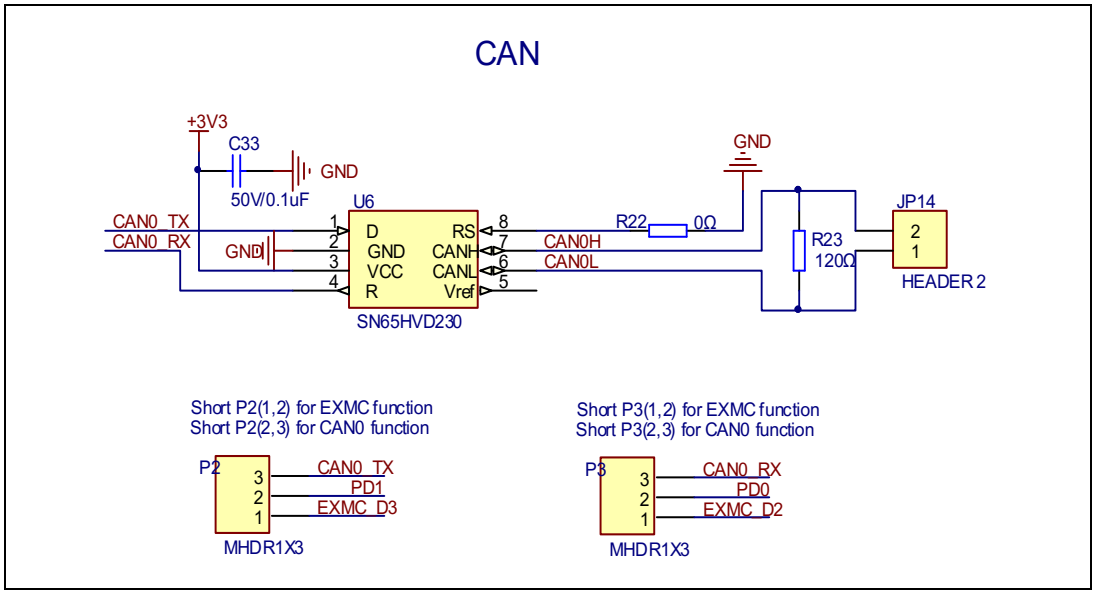
4.10. SPI

图 4-10 SPI功能原理图



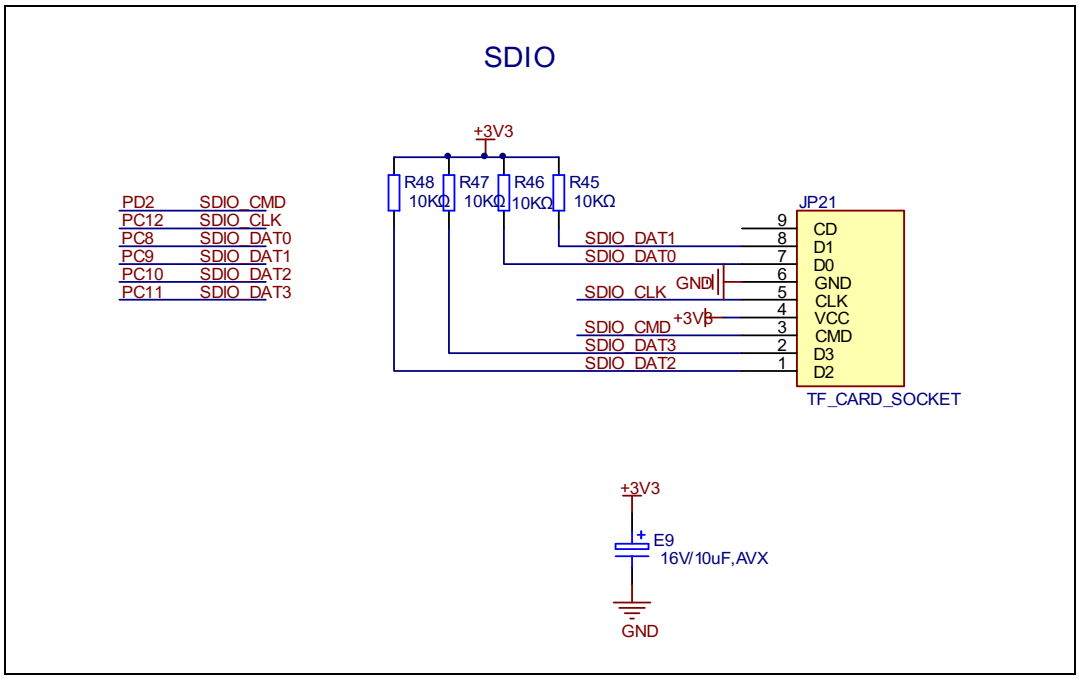
4.11. CAN

图 4-11 CAN功能原理图



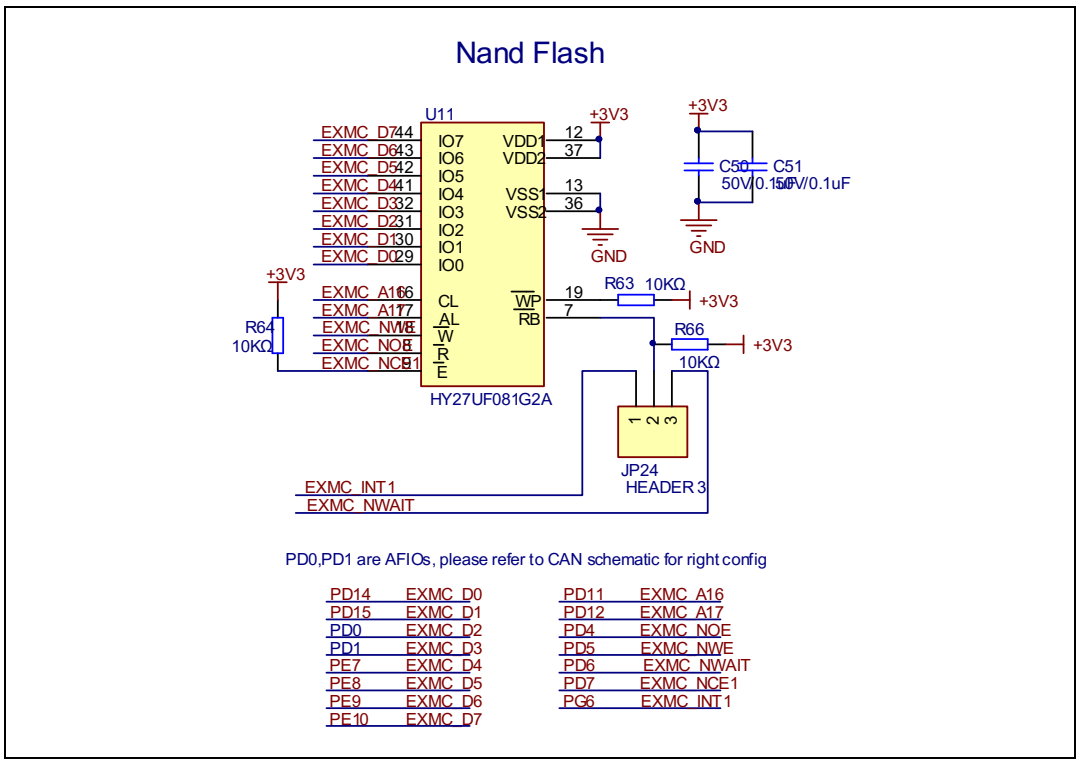
4.12. SDIO

图 4-12 SDIO功能原理图



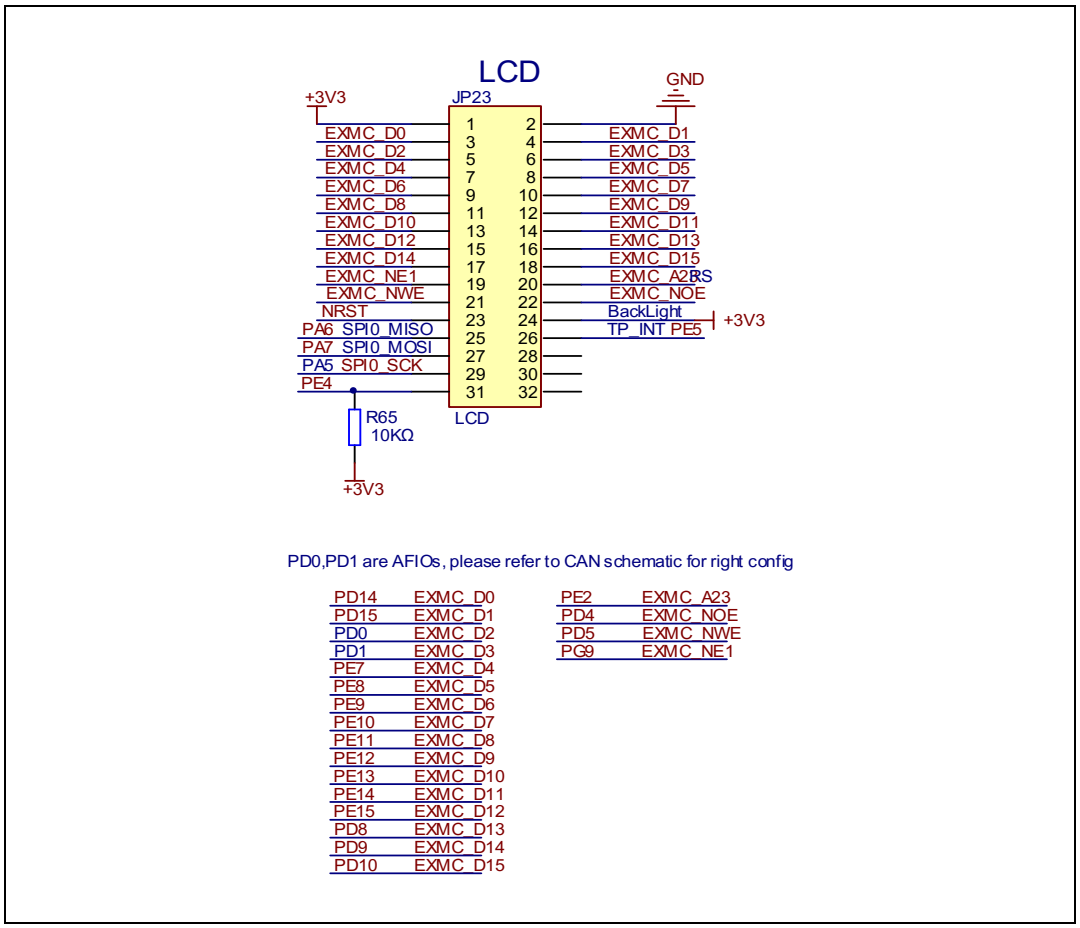
4.13. NAND

图 4-13 NAND存储器功能原理图



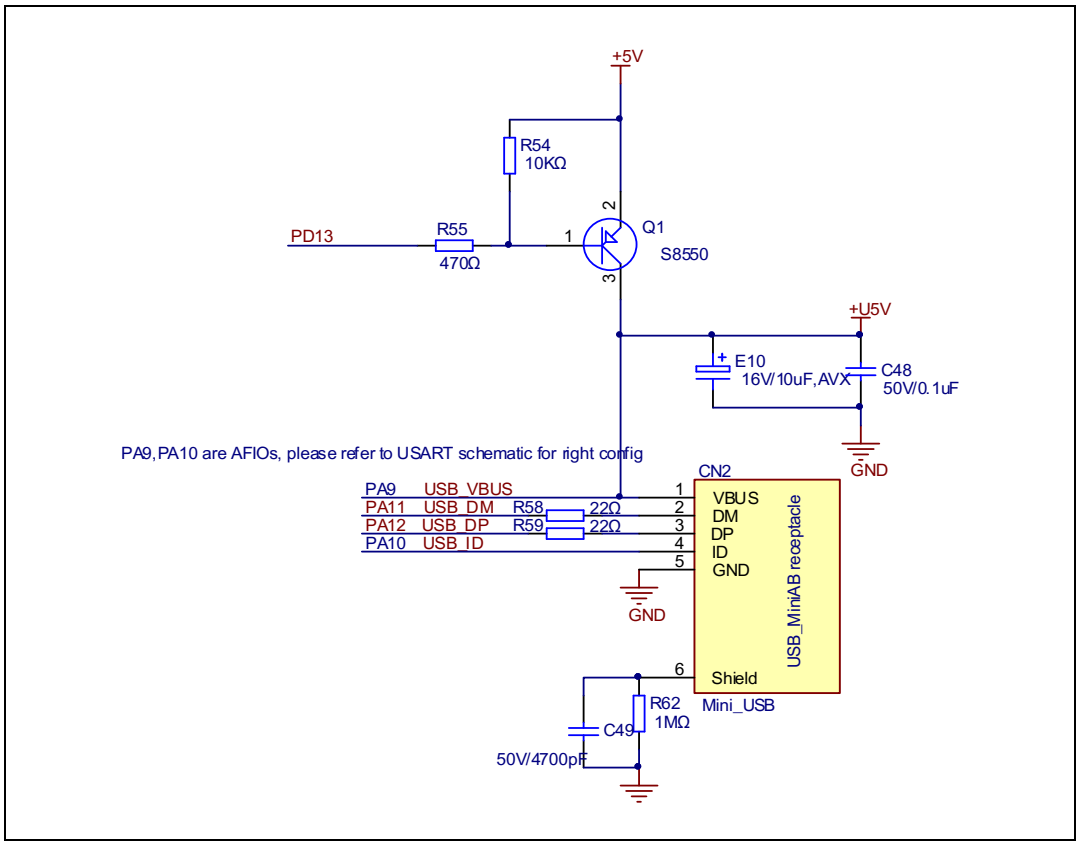
4.14. LCD

图 4-14 LCD功能原理图



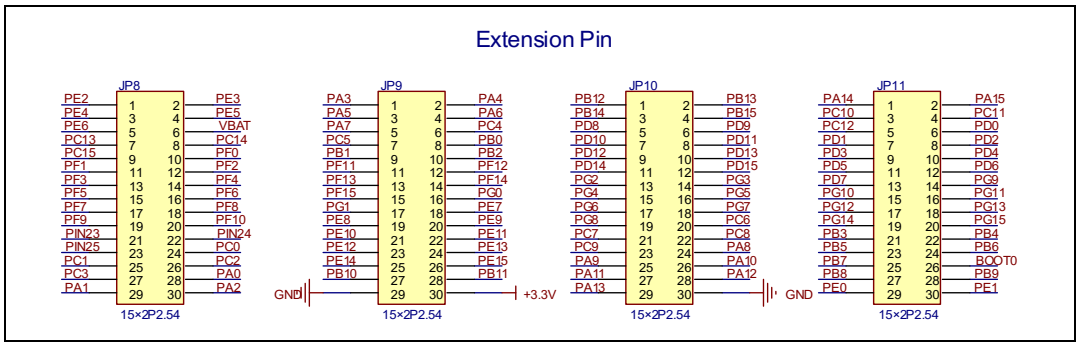
4.15. USBFS

图 4-15 USBFS功能原理图



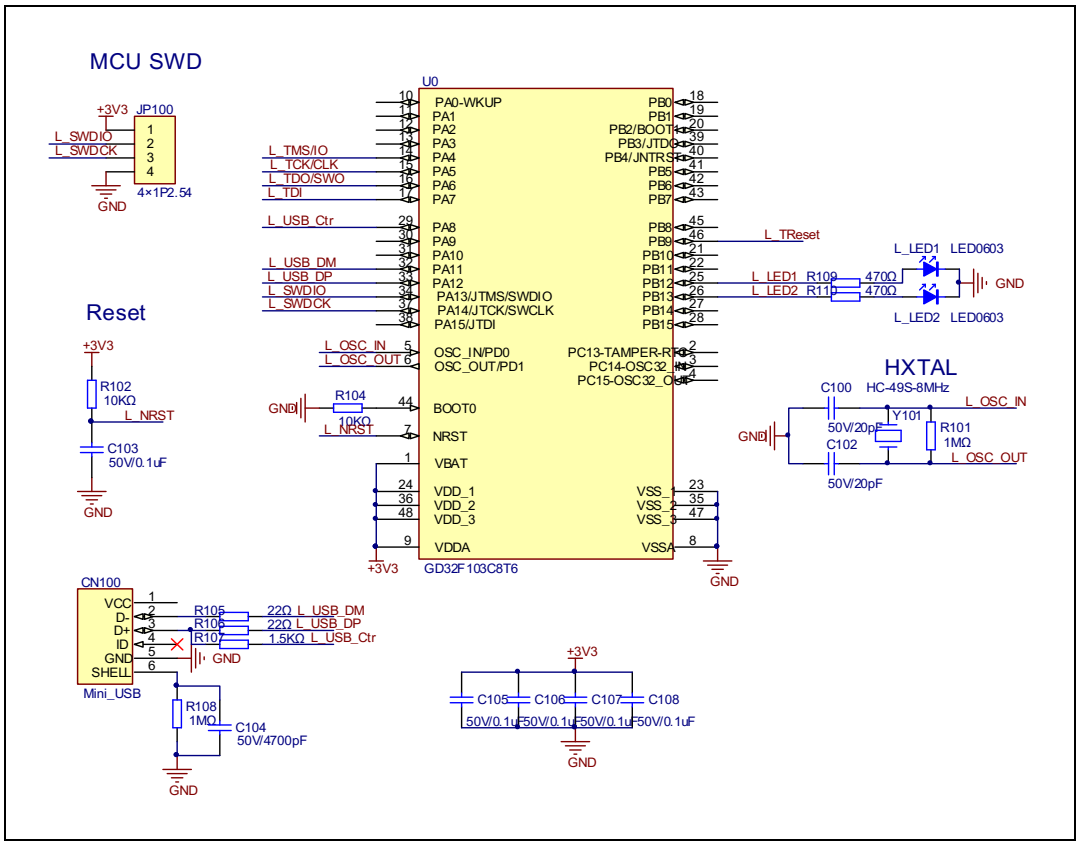
4.16. 扩展电路

图 4-16 扩展电路原理图



4.17. GD-Link

图 4-17 GD-Link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

GD32403Z-EVAL-V1.1 开发板上有 4 个 LED。LED2，LED3，LED4 和 LED5 通过 GPIO 控制着。这个例程将讲述怎么点亮 LED。

5.1.2. DEMO 执行结果

下载程序<01_GPIO_Running_LED>到开发板上，LED2，LED3，LED4 和 LED5 将顺序每间隔 1000 毫秒点亮，然后重复前面的过程。

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

GD32403Z-EVAL-V1.1 开发板有五个按键和四个 LED。其中，五个按键是 Reset 按键，Tamper 按键，Wakeup 按键，User1 按键，User2 按键；LED2，LED3，LED4 和 LED5 可通过 GPIO 控制。

这个例程讲述如何使用 Tamper 按键控制 LED2。当按下 Tamper 按键，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 按键，LED2 将会点亮，再次按下 Tamper 按键，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

GD32403Z-EVAL-V1.1 开发板有五个按键和四个 LED。其中，五个按键是 Reset 按键，Tamper 按键，Wakeup 按键，User1 按键，User2 按键；LED2，LED3 和 LED4，LED5 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，用跳线帽将 JP5 和 JP6 跳到 USART 上，并将串口线连到开发板的 COM0 上。例程首先将输出"USART printf example: please press the Tamper key"到超级终端。按下 Tamper 键，点亮 LED3 同时串口继续输出"USART printf example"。

通过串口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串口助手之间的通信

5.5.2. DEMO 执行结果

下载程序< 05_USART_Echo_Interrupt_mode >到开发板，用跳线帽将 JP5 和 JP6 跳到

USART 上，并将串口线连到开发板的 COM0 上。首先，所有灯亮灭一次用于测试。然后 COM0 将首先输出数组 tx_buffer 的内容(从 0x00 到 0xFF)到支持 hex 格式的串口助手并等待接收由串口助手发送的 BUFFER_SIZE 个字节的数据。MCU 将接收到的串口助手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，LED2，LED3，LED4，LED5 轮流闪烁；如果结果不相同，LED2，LED3，LED4，LED5 一起闪烁。

通过串口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，用跳线帽将 JP5 和 JP6 跳到 USART 上，并将串口线连到开发板的 COM0 上。首先，所有灯亮灭一次用于测试。然后 COM0 将首先输出数组 tx_buffer 的内容(从 0x00 到 0xFF)到支持 hex 格式的串口助手并等待接收由串口助手发送的与 tx_buffer 字节数相同的数据。MCU 将接收到的串口助手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。LED2，LED3，LED4，LED5 轮流闪烁；如果结果不相同，LED2，LED3，LED4，LED5 一起闪烁。

通过串口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
```

5.7. ADC 温度传感器_Vrefint

5.7.1. 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 Vrefint 通道)

5.7.2. DEMO 执行结果

将 JP5 和 JP6 跳到 USART 用于通过超级终端显示打印信息。下载<07_ADC_Temperature_Vrefint>至开发板并运行。将开发板的 COM0 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、内部参考电压和电池电压的值。

注意：由于温度传感器存在偏差，如果需要测量精确的温度，应该使用一个外置的温度传感器来校准这个偏移错误。

the temperature data is 31 degrees Celsius
the reference voltage data is 1.205V

the temperature data is 31 degrees Celsius
the reference voltage data is 1.206V

the temperature data is 31 degrees Celsius
the reference voltage data is 1.206V

the temperature data is 31 degrees Celsius
the reference voltage data is 1.206V

the temperature data is 31 degrees Celsius
the reference voltage data is 1.205V

the temperature data is 31 degrees Celsius
the reference voltage data is 1.205V

5.8. ADC0 和 ADC1 跟随模式

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 ADC0 和 ADC1 工作在跟随模式

5.8.2. DEMO 执行结果

将 JP5 和 JP6 跳到 USART 用于通过超级终端显示打印信息。下载<08_ADC0_ADC1_Follow_up_mode>至开发板并运行。将开发板的 COM0 口连接到电脑，打开电脑串口软件。

TIMER0_CH0 作为 ADC0 和 ADC1 的触发源。当 TIMER0_CH0 的上升沿到来，ADC0 立即启动，经过几个 ADC 时钟周期后，ADC1 启动。ADC0 和 ADC1 的值通过 DMA 传送给 adc_value[0]和 adc_value[1]。

当 TIMER0_CH0 的第一个上升沿到来，ADC0 转换的 PC3 引脚的电压值存储到 adc_value[0]的低半字，经过几个 ADC 时钟周期后，ADC1 转换的 PC5 引脚的电压值存储到 adc_value[0]的高半字。当 TIMER0_CH0 的第二个上升沿到来，ADC0 转换的 PC5

引脚的电压值存储到 adc_value[1]的低半字，经过几个 ADC 时钟周期后，ADC1 转换的 PC3 引脚的电压值存储到 adc_value[1]的高半字。

当程序运行时，当程序运行时，串口软件会显示 adc_value[0] 和 adc_value[1]的值。

```
the data adc_value[0] is 000009B2
the data adc_value[1] is 09C00000

the data adc_value[0] is 000009B8
the data adc_value[1] is 09BD0000

the data adc_value[0] is 000009B8
the data adc_value[1] is 09BD0000

the data adc_value[0] is 000009B3
the data adc_value[1] is 09BF0000

the data adc_value[0] is 000009B7
the data adc_value[1] is 09BF0000

the data adc_value[0] is 000009B5
the data adc_value[1] is 09BE0000
```

5.9. ADC0 和 ADC1 常规并行模式

5.9.1. 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 ADC0 和 ADC1 工作在常规并行模式

5.9.2. DEMO 执行结果

将 JP5 和 JP6 跳到 USART 用于通过超级终端显示打印信息。下载<09_ADC0_ADC1_Regular_Parallel_mode>至开发板并运行。将开发板的 COM 口连接到电脑，打开电脑串口软件。

TIMER0_CH0 作为 ADC0 和 ADC1 的触发源。当 TIMER0_CH0 的上升沿到来，ADC0 和 ADC1 会立即启动，并行转换常规序列通道。ADC0 和 ADC1 的值通过 DMA 传送给 adc_value[0]和 adc_value[1]。

当 TIMER0_CH0 的第一个上升沿到来，ADC0 转换的 PC3 引脚的电压值存储到 adc_value[0]的低半字，并且 ADC1 转换的 PC5 引脚的电压值存储到 adc_value[0]的高半字。当 TIMER0_CH0 的第二个上升沿到来，ADC0 转换的 PC5 引脚的电压值存储到 adc_value[1]的低半字，并且 ADC1 转换的 PC3 引脚的电压值存储到 adc_value[1]的高半字。

当程序运行时，当程序运行时，串口软件会显示 adc_value[0]和 adc_value[1]的值。

```
the data adc_value[0] is 000009BB
the data adc_value[1] is 09BB0000

the data adc_value[0] is 000009BB
the data adc_value[1] is 09BB0000

the data adc_value[0] is 000009B9
the data adc_value[1] is 09BB0000

the data adc_value[0] is 000009BB
the data adc_value[1] is 09BA0000

the data adc_value[0] is 000009BA
the data adc_value[1] is 09BA0000
```

5.10. DAC 输出电压值

5.10.1. 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

5.10.2. DEMO 执行结果

下载程序<10_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。

5.11. 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，使用跳线帽 JP5、JP6 跳线到 USART。将开发板的 COM0 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 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!
```

```
GD32403z-EVAL 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!
```

5.12. SPI 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 主模式读写带有 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 COM0 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。同时，将 JP5 和 IP6 跳线到 USART，将 JP12 跳线到 SPI。

下载程序 <12_SPI_SPI_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。

```
################################################################################
GD32403Z-EVAL-V1.1 System is Starting up...
GD32403Z-EVAL-V1.1 Flash:512K
GD32403Z-EVAL-V1.1 The CPU Unique Device ID:[37373431-7363533-600200]
GD32403Z-EVAL-V1.1 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

GD32403Z-EVAL-V1.1 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. NAND 存储器

5.14.1. DEMO 目的

这个例程包括了 GD32 MCU 的以下功能：

■ 学习使用 EXMC 控制 NAND Flash

5.14.2. DEMO 执行结果

GD32403Z -EVAL-V1.1 开发板使用 EXMC 模块来控制 NAND Flash。在运行例程之前，JP5 连接到 USART，P2 和 P3 连接到 EXMC，JP24 连接到 Nwait。下载程序<14_EXMC_NandFlash>到开发板。这个例程演示 EXMC 对 NAND 的读写操作，最后会把读写的操作进行比较，如果数据一致，点亮 LED2，否则点亮 LED4。超级终端输出信息如下：

```
NAND flash initialized!
Read NAND ID!
Nand flash ID:0xC8 0xF1 0x80 0x1D

Write data successfully!
Read data successfully!
Check the data!
Access NAND flash successfully!
The data to be read:
    0     1     2     3     4     5     6     7     8     9     a     b     c     d     e     f
   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
```

5.15. LCD 触摸屏

5.15.1. DEMO 目的

这个例程包括了 GD32 MCU 的以下功能：

■ 学习使用 EXMC 控制 LCD

5.15.2. DEMO 执行结果

GD32403Z-EVAL-V1.1 开发板使用 EXMC 模块来控制 LCD。在运行例程之前，JP12 连接到 SPI，P2 和 P3 连接到 EXMC。下载程序<15_EXMC_TouchScreen>到开发板。这个例程将通过 EXMC 模块在 LCD 屏上显示 GigaDevice 的 logo 和 4 个绿色按钮。用户可以通过触摸屏上的按钮来点亮开发板中对应的 LED，同时屏上触摸过的按钮颜色将变成红色。



5.16. SD 卡测试

5.16.1. DEMO 目的

这个例程包括了 GD32 MCU 的以下功能：

- 学习使用 SDIO 单个数据块或多个数据块读写操作
- 学习使用 SDIO 对 SD 卡进行擦除、上锁和解锁操作

GD32403Z-EVAL-V1.1 开发板有一个 SDIO 接口，它定义了 SD/SD I/O /MMC CE-ATA 卡主机接口。这个例程讲述了如何使用 SDIO 接口来操作 SD 卡。

5.16.2. DEMO 执行结果

将 JP5 跳到 USART 用于通过超级终端显示打印的信息。下载<16_SDIO_SDCardTest>至评估板并运行。将开发板的 COM0 口连接到电脑，打开超级终端。所有的 LED 灯先亮灭一次用于测试目的。然后初始化卡并打印卡的相关信息。接着再测试单块操作、上锁/解锁卡操作、擦除操作和多块操作。如果发生错误，打印错误信息并点亮 LED2 和 LED4，熄灭 LED3 和 LED5。否则，点亮所有 LED。

取消宏 DATA_PRINT 的注释，可以打印数据信息。通过对相关语句取消或加上注释，可以设置不同的总线模式(1-bit 或 4-bit)和数据传输模式(轮询模式或 DMA 模式)。

串口输出如下图所示：

```
Card init success!

Card information:
## Card version 3.0x ##
## SDHC card ##
## Device size is 7782400KB ##
## Block size is 512B ##
## Block count is 15564800 ##
## CardCommandClasses is: 5b5 ##
## Block operation supported ##
## Erase supported ##
## Lock unlock supported ##
## Application specific supported ##
## Switch function supported ##

Card test:
Block write success!
Block read success!
The card is locked!
Erase failed!
The card is unlocked!
Erase success!
Block read success!
Multiple block write success!
Multiple block read success!
```

5.17. CAN 网络通信

5.17.1. DEMO 目的

这个例程包括了 GD32 MCU 的以下功能：

■ 学习使用 CAN0 实现两个板子之间的通信

GD32403Z-EVAL-V1.1 开发板集成了 CAN(控制器局域网络)总线控制器，他是一种常用的工业控制总线。CAN 总线控制器遵循 2.0A 和 2.0B 总线协议。该例程演示了在两个板子之间通过 CAN0 进行通信。

5.17.2. DEMO 执行结果

该例程的测试需要两个 GD32F403Z-EVAL 开发板。用跳线帽将 JP5 和 JP6 跳到 USART 上，P2，P3 跳到 CAN 上。将两个板子的 JP14 的 L 引脚和 H 引脚分别相连，用于发送或者接收数据帧。下载程序<17_CAN_Network>到两个开发板中，并将串口线连到开发板的 COM0 上。例程首先将输出"please press the Tamper key to transmit data!"到超级终

端。按下 Tamper 键，数据帧通过 CAN0 发送出去同时通过串口打印出来。当接收到数据帧时，接收到的数据通过串口打印，同时 LED2 状态翻转一次。通过串口输出的信息如下图所示。

```
please press the Tamper key to transmit data!
CAN0 transmit data: ab,cd
CAN0 recive data: ab,cd
```

5.18. RCU 时钟输出

5.18.1. 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

5.18.2. DEMO 执行结果

使用跳线帽 JP5 和 JP6 跳线到 USART，下载程序<18_RCU_Clock_Out>到开发板上并运行。将开发板的 COM0 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_OUT0: system clock
CK_OUT0: IRC8M
CK_OUT0: HXTAL
CK_OUT0: system clock
```

5.19. CTC 校准

5.19.1. 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 振荡器时钟

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。

5.19.2. DEMO 执行结果

下载程序<19_CTC_Calibration>到开发板上，运行程序。首先，所有的灯依次闪烁用于测试目的，如果内部 48MHz RC 校准成功，LED2 将会点亮。否则，所有 LED 灯均熄灭。

5.20. PMU 睡眠模式唤醒

5.20.1. 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 MCU 睡眠模式

5.20.2. DEMO 执行结果

下载程序<20_PMU_sleep_wakeup>到开发板上，用跳线帽将 JP5，JP6 跳到 USART 上，并将串口线连到开发板的 COM0 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。

5.21. RTC 日历

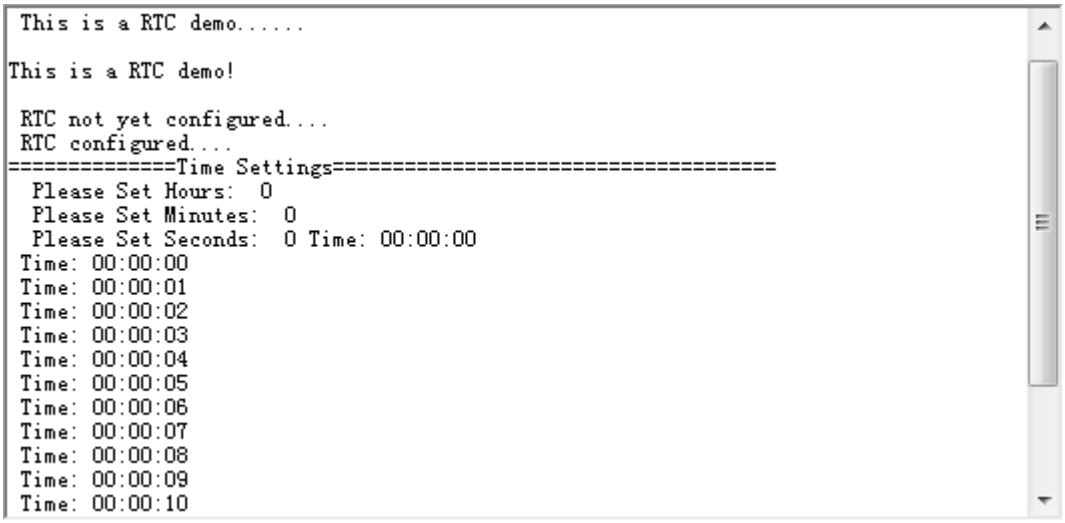
5.21.1. DEMO 目的

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

5.21.2. DEMO 执行结果

将 JP5 和 JP6 跳到 USART，下载程序<21_RTC_Calendar>到开发板上，使用串口线连接电脑到开发板 COM0 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。



5.22. 呼吸灯

5.22.1. DEMO 目的

这个例程包括了 GD32 MCU 的以下功能：

- 学习使用定时器输出 PWM 波
- 学习更新定时器通道寄存器的值

5.22.2. DEMO 执行结果

使用杜邦线连接 TIMER0_CH0(PA8)和 LED2(PF0)，然后下载程序<22_TIMER_Breath_LED>到开发板，并运行程序。

PA8 不要用于其他外设。

可以看到 LED2 由暗变亮，由亮变暗，往复循环，就像人的呼吸一样有节奏。

5.23. USB 设备

5.23.1. HID_键盘

DEMO 目的

这个例程包括了 GD32 MCU 的以下功能：

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

GD32403Z 开发板具有五个按键和一个 USBFS 接口，这五个按键分别是 Reset 按键、Wakeup 按键、Tamper 按键、User 按键 1 和 User 按键 2。在本例程中，GD32403Z 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，USB 键盘利用 Wakeup 键、Tamper 键和 User 键输出三个字符('b'，'a'和'c')。另外，本例程支持 USB

键盘远程唤醒主机，其中 Wakeup 按键被作为唤醒源。



DEMO 执行结果

在运行程序之前，确保将 JP5 与 JP6 跳到 OTG，然后将<23_USBFS\USB_Device\HID_Keyboard >例程下载到开发板中，并运行。按下 Wakeup 键，输出'b'；按下 User 键，输出'c'；按下 Tamper 键，输出'a'。

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。

5.23.2. MSC_U 盘

DEMO 目的

这个例程包括了 GD32 MCU 的以下功能：

- 学习如何使用 USBFS 的设备模式
- 学习如何实现 USB MSC(大容量存储)设备

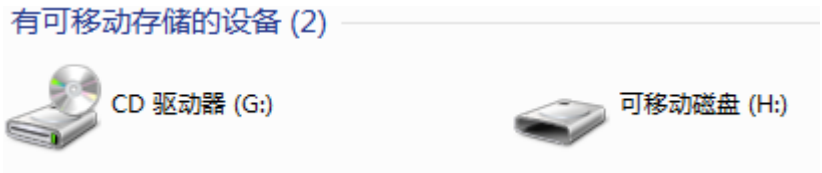
本 DEMO 主要实现了一个 U 盘。U 盘是现今非常普遍的可移动 MSC 类设备。MSC，即 Mass Storage device Class(大容量存储设备类)，是一种计算机和移动设备之间的传输协议，它允许一个通用串行总线(USB)设备来访问主机的计算设备，使两者之间进行文件传输，主要包括移动硬盘、移动光驱和 U 盘等。MSC 类设备必须有存储介质，DEMO 中使用了 MCU 的内部 flash 作为存储介质。具体的 MSC 类协议内容请自行查阅与参考其协议标准。

MSC 类设备会使用多种传输协议与命令格式进行通信，所以在实现时需要自行选择合适的协议与命令格式。本 DEMO 中选择 BOT(仅批量传输)协议和所需的 SCSI(小型计算机接口)命令，并和多种 Window 操作系统兼容。具体的 BOT 协议内容与 SCSI 命令规格请自行查阅与参考其协议标准。

DEMO 执行结果

在运行程序之前，确保将 JP5 与 JP6 跳到 OTG，然后下载<23_USBFS\USB_Device\MSC(Internal_sram)>到开发板中并运行。当开发板连到 PC 后，可以在计算机的设备管理器中看到通用串行总线控制器里面多出了一个 USB 大容量存储设备，同时看到磁盘驱动器里面多了 1 个磁盘驱动器。

此时，打开资源管理器后会看到里面多了 1 个磁盘，如下图所示：



此时，写/读/格式化操作可以像其他移动设备一样进行。

5.24. USB 主机

5.24.1. HID 主机

DEMO 目的

这个例程包括了 GD32 MCU 的以下功能：

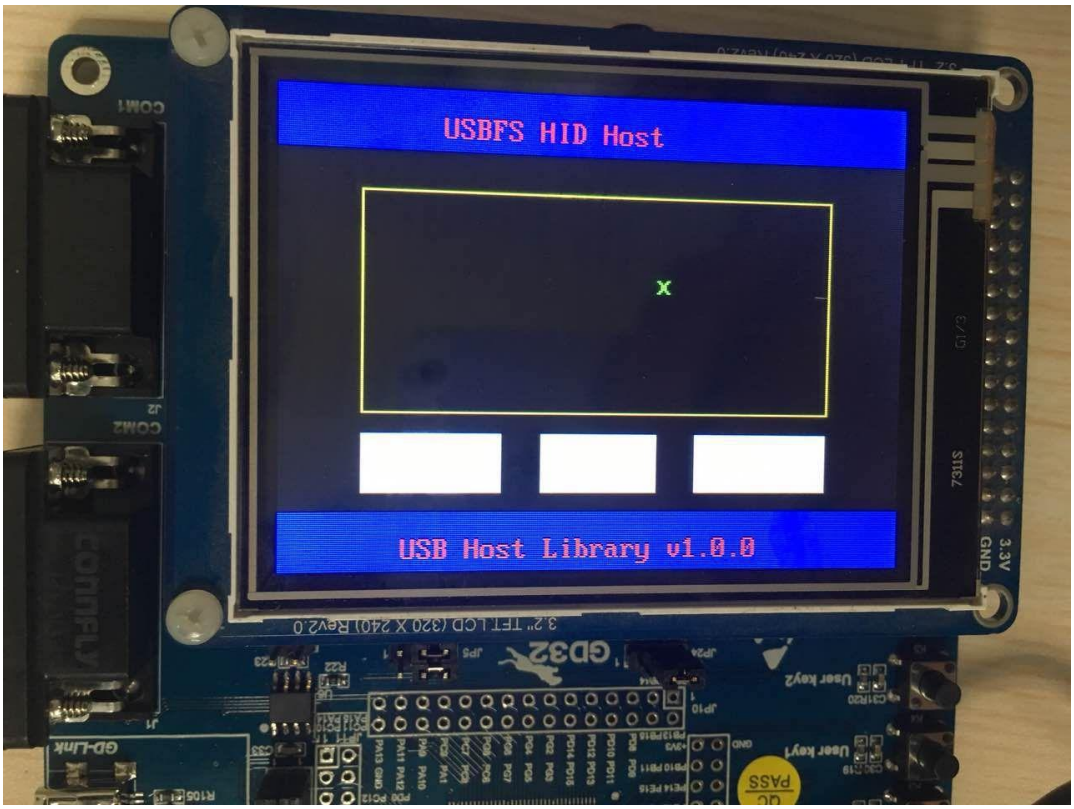
- 学习使用 USBFS 模块作为 HID 主机
- 学习 HID 主机和鼠标设备之间的操作
- 学习 HID 主机和键盘设备之间的操作

GD32403Z-EVAL-V1.1 开发板内部包含 USBFS 模块，该模块可以被使用作为一个 USB 设备、一个 USB 主机或者一个 OTG 设备。该示例主要展示了如何使用 USBFS 作为一个 USB HID 主机和外部 USB HID 设备进行通信。

DEMO 执行结果

将 JP5 和 JP6 引脚跳到 OTG，将<23_USBFS\USB_Host\HID_Host>代码下载到开发板并运行。

如果一个鼠标被连入，用户将会看到鼠标枚举的信息。首先按下 User 按键 1，将会看到插入的设备是鼠标；然后移动鼠标，将会在液晶上看到鼠标的位置和按键的状态。



如果一个键盘被连入，用户将会看到键盘枚举的信息。首先按下 User 按键 1 将会看到插入的设备是键盘，然后按下键盘按键，将会通过液晶显示按键状态。



5.24.2. MSC 主机

DEMO 目的

这个例程包括了 GD32 MCU 的以下功能：

■ 学习使用 USBFS 作为 MSC 主机

■ 学习 MSC 主机和 U 盘之间的操作

GD32403Z-EVAL-V1.1 开发板包含 USBFS 模块，并且该模块可以被用于作为一个 USB 设备、一个 USB 主机或一个 OTG 设备。本示例主要显示如何使用 USBFS 作为一个 USB MSC 主机来与外部 U 盘进行通信。

DEMO 执行结果

将 JP5 和 JP6 引脚跳到 OTG。然后将 OTG 电缆线插入到 USB 接口，将 <23_USBFS\USB_Host\MSC_Host >工程下载到开发板中并运行。

如果一个 U 盘被连入，用户将会看到 U 盘枚举信息。首先按下 User 按键将会看到 U 盘信息；之后按下 Tamper 按键将会看到 U 盘根目录内容；然后按下 Wakeup 按键将会向 U 盘写入文件；最后用户将会看到 MSC 主机示例结束的信息。

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	2017 年 5 月 19 日
2.0	更新开发板版本	2019 年 01 月 07 日

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.

© 2019 GigaDevice - All rights reserved