

GigaDevice Semiconductor Inc.

GD32E230C-EVAL 评估板
用户指南

目录

目录.....	1
图.....	4
表.....	5
1. 简介.....	6
2. 功能引脚分配.....	6
3. 入门指南.....	7
4. 硬件设计概述.....	7
4.1. 供电电源.....	7
4.2. 启动方式选择.....	7
4.3. LED 指示灯.....	8
4.4. 按键.....	8
4.5. 串口.....	8
4.6. 模数转换器.....	9
4.7. I2S.....	9
4.8. I2C.....	9
4.9. QSPI-FLASH.....	9
4.10. SPI-TFT LCD.....	10
4.11. 比较器.....	10
4.12. 红外线接口.....	11
4.13. GD-Link.....	11
4.14. 扩展电路.....	12
4.15. MCU.....	12
5. 例程使用指南.....	13
5.1. GPIO 流水灯.....	13
5.1.1. DEMO 目的.....	13
5.1.2. DEMO 执行结果.....	13
5.2. GPIO 按键轮询模式.....	13
5.2.1. DEMO 目的.....	13
5.2.2. DEMO 执行结果.....	13

5.3. GPIO 按键中断模式	13
5.3.1. DEMO 目的	13
5.3.2. DEMO 执行结果	14
5.4. 串口打印	14
5.4.1. DEMO 目的	14
5.4.2. DEMO 执行结果	14
5.5. 串口中断收发	14
5.5.1. DEMO 目的	14
5.5.2. DEMO 执行结果	14
5.6. 串口 DMA 收发	15
5.6.1. DEMO 目的	15
5.6.2. DEMO 执行结果	15
5.7. 定时器触发模数转换	15
5.7.1. DEMO 目的	15
5.7.2. DEMO 执行结果	16
5.8. 比较器输出获取指示灯	16
5.8.1. DEMO 目的	16
5.8.2. DEMO 执行结果	16
5.9. I2C 访问 EEPROM	16
5.9.1. DEMO 目的	16
5.9.2. DEMO 执行结果	16
5.10. QSPI FLASH	17
5.10.1. DEMO 目的	17
5.10.2. DEMO 执行结果	17
5.11. SPI 驱动 LCD 液晶屏	18
5.11.1. DEMO 目的	18
5.11.2. DEMO 执行结果	18
5.12. 音频播放器	19
5.12.1. DEMO 目的	19
5.12.2. DEMO 执行结果	19
5.13. RCU 时钟输出	19
5.13.1. DEMO 目的	19
5.13.2. DEMO 执行结果	20
5.14. PMU 睡眠模式唤醒	20
5.14.1. DEMO 目的	20
5.14.2. DEMO 执行结果	20
5.15. RTC 实时时钟	20
5.15.1. DEMO 目的	20
5.15.2. DEMO 执行结果	20

5.16.	红外收发器	21
5.16.1.	DEMO 目的	21
5.16.2.	DEMO 执行结果	21
5.17.	TIMER 呼吸灯	21
5.17.1.	DEMO 目的	21
5.17.2.	DEMO 执行结果	21
6.	版本更新历史.....	22

图

图 4-1 供电电源原理图.....	7
图 4-2 启动方式选择原理图.....	7
图 4-3 LED 功能原理图.....	8
图 4-4 按键功能原理图.....	8
图 4-5 串口 0 功能原理图.....	8
图 4-6 模数转换器功能原理图.....	9
图 4-7 I2S 功能原理图.....	9
图 4-8 I2C 功能原理图.....	9
图 4-9 QSPI-FLASH 功能原理图.....	9
图 4-10 SPI-TFT LCD 功能原理图.....	10
图 4-11 比较器功能原理图.....	10
图 4-12 红外线接口功能原理图.....	11
图 4-13 GD-Link 功能原理图.....	11
图 4-14 扩展电路功能原理图.....	12
图 4-15 MCU 功能原理图.....	12

表

表 2-1 引脚分配.....	6
表 4-1 启动方式配置.....	7
表 6-1 版本更新历史.....	22

1. 简介

GD32E230C-EVAL 评估板使用 GD32E230C8T6 作为主控制器。该评估板为采用 Cortex[™]-M23 内核的 GD32E230 芯片提供了一个完整的开发平台，支持全方位的外围设备。评估板使用 MiniUSB 接口作为供电电源。提供包括扩展引脚在内的以及 GD-Link、Reset、Boot、User button key、LED、I2C、I2S、USART、TFT-LCD、IFRP LED、IFRP Transceiver、RTC、SPI、ADC 等外设资源。本文档提供详细的硬件原理图和相关应用程序。

2. 功能引脚分配

表 2-1 引脚分配

Function	Pin	Description
LED	PA8	LED1
	PA11	LED2
	PA12	LED3
	PA15	LED4
RESET		K1-Reset
KEY	PA0	K2-Wakeup
	PC13	K3-Tamper
IR	PB4	IR_RX
	PB9	IR_TX
I2C	PB6	I2C0_SCL
	PB7	I2C0_SDA
I2S	PA4	I2S0_WS
	PA5	I2S0_CK
	PA7	I2S0_SD
	PA6	I2S0_MCK
USART0	PA9	USART0_TX
	PA10	USART0_RX
SPI	PB13	SPI1_SCK
	PB14	SPI1_MISO
	PB15	SPI1_MOSI
	PB10	SPI1_IO2
	PB11	SPI1_IO3
	PB12	SPIFlash_CS
	PB0	TFT_CS
	PB1	TFT_RESET
ADC	PA2	ADC_IN2
COMPARATOR	PA1	COMP0_INP

3. 入门指南

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

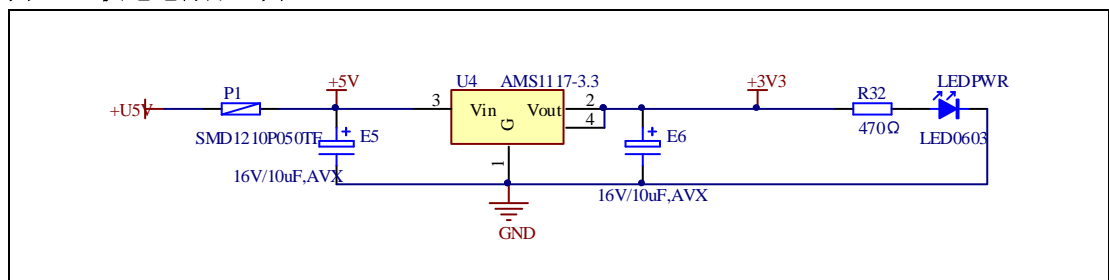
所有例程提供了 Keil 和 IAR 两个版本，其中 Keil 版的工程是基于 Keil MDK-ARM 5.25 uVision5 创建的，IAR 版的工程是基于 IAR Embedded Workbench for ARM 8.32.1 创建的。在使用过程中有如下几点需要注意：

- 1、如果使用 Keil uVision5 打开工程，安装 GigaDevice.GD32E230_DFP.1.0.0.pack，以加载相关文件。
- 2、如果使用 IAR 打开工程，安装 IAR_GD32E230_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

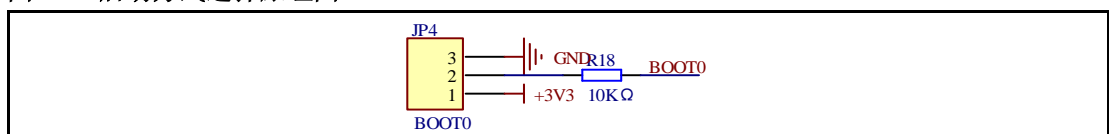
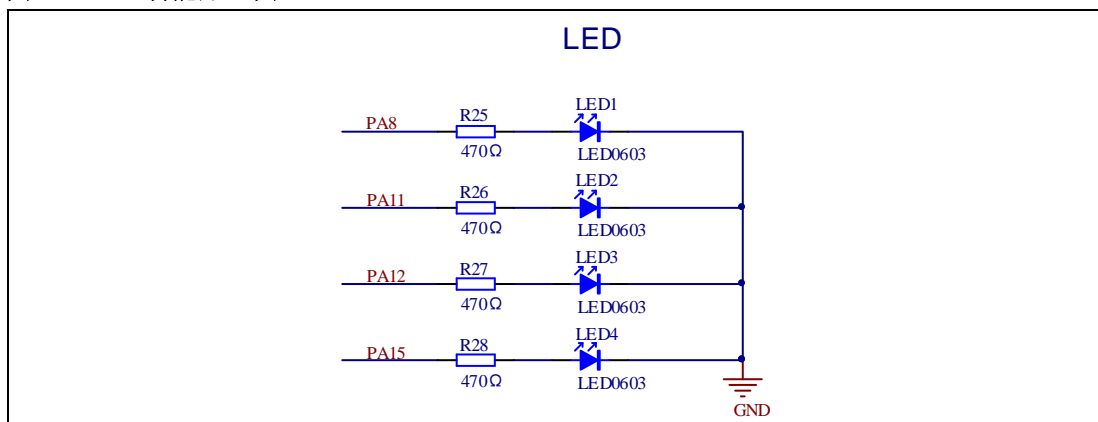


表 4-1 启动方式配置

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

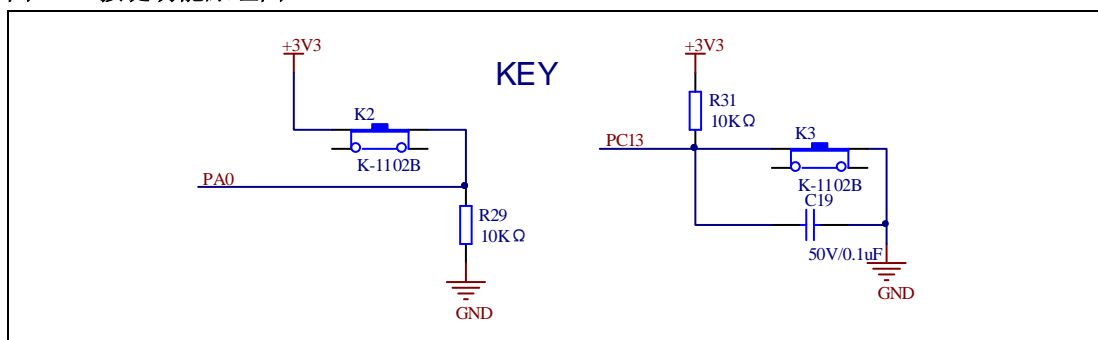
4.3. LED 指示灯

图 4-3 LED功能原理图



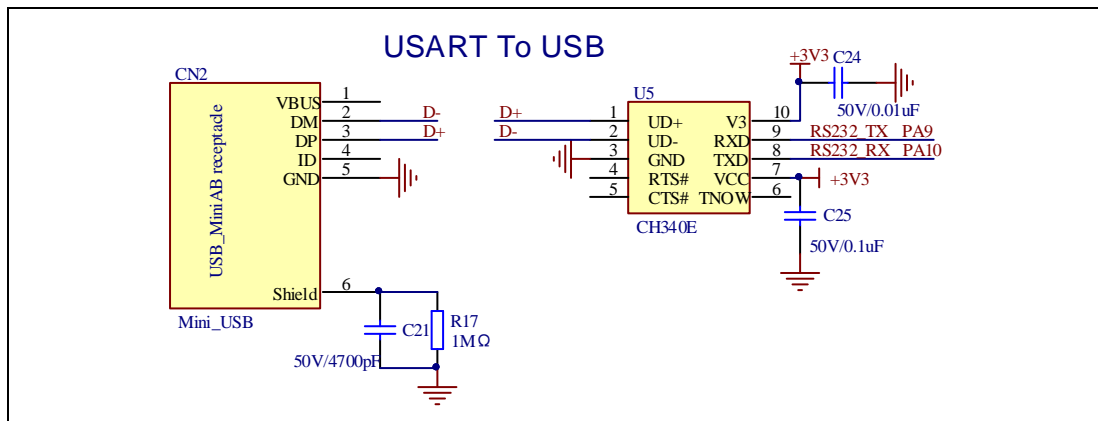
4.4. 按键

图 4-4 按键功能原理图



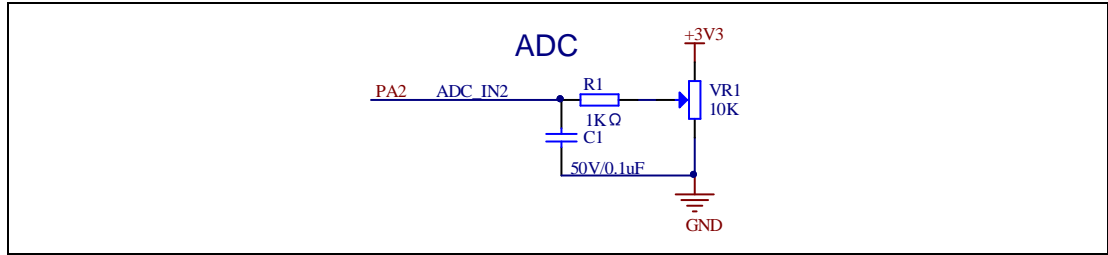
4.5. 串口

图 4-5 串口0功能原理图



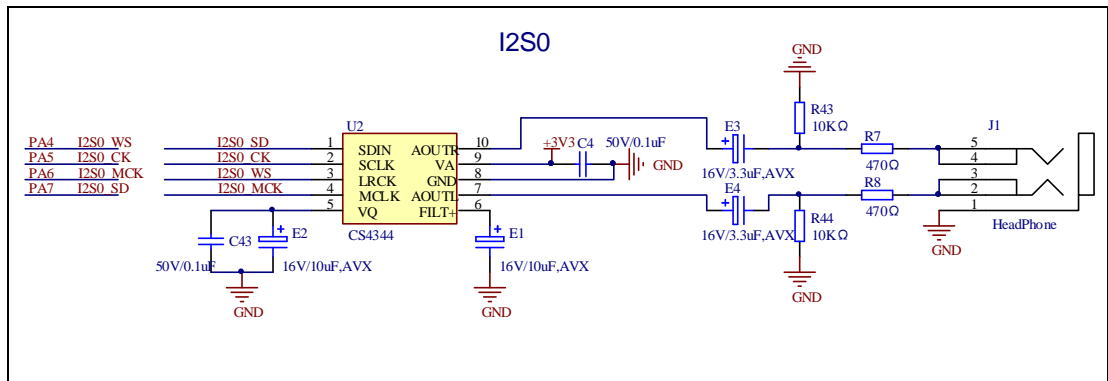
4.6. 模数转换器

图 4-6 模数转换器功能原理图



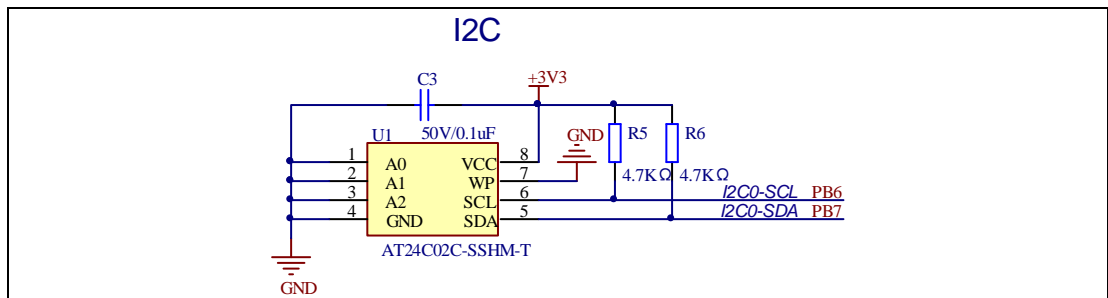
4.7. I2S

图 4-7 I2S功能原理图



4.8. I2C

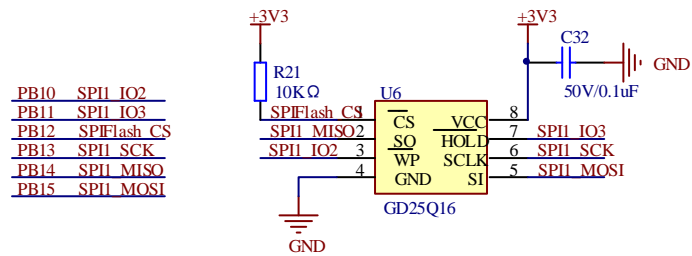
图 4-8 I2C功能原理图



4.9. QSPI-FLASH

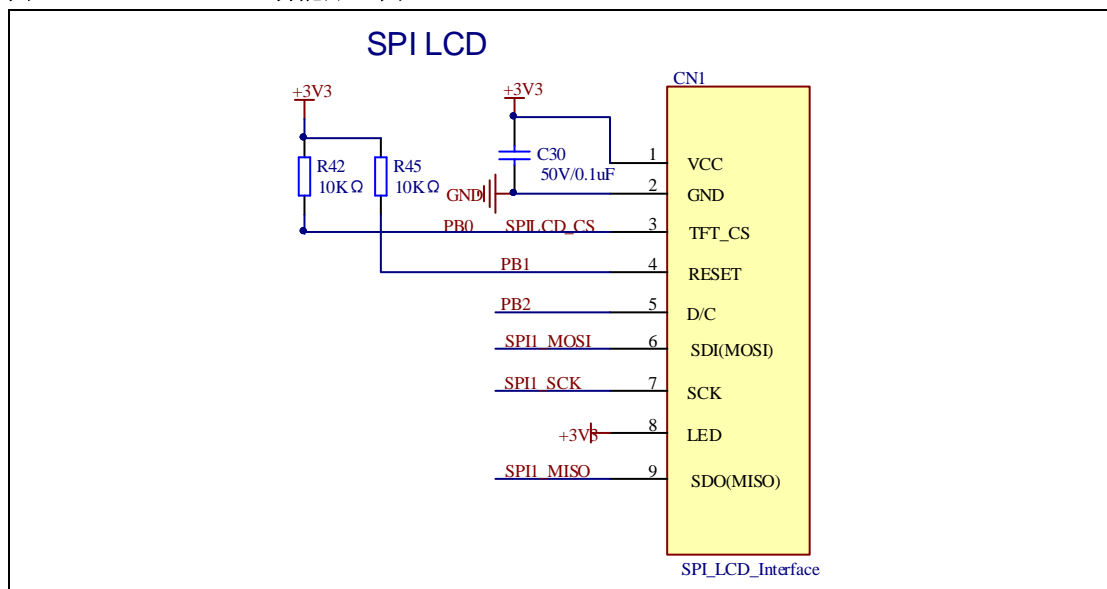
图 4-9 QSPI-FLASH功能原理图

SPI Flash



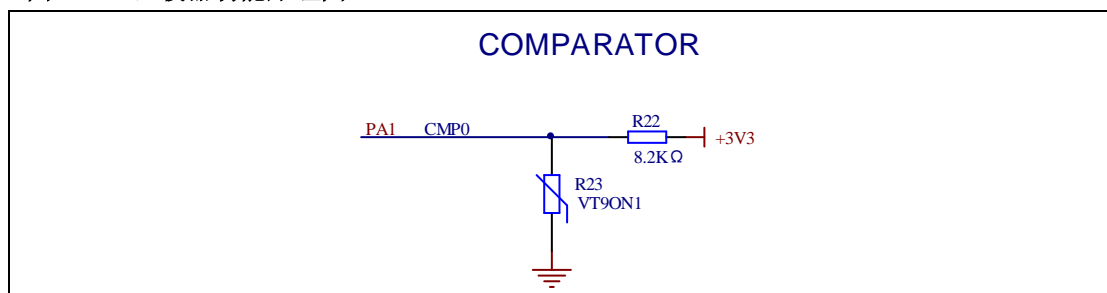
4.10. SPI-TFT LCD

图 4-10 SPI-TFT LCD功能原理图



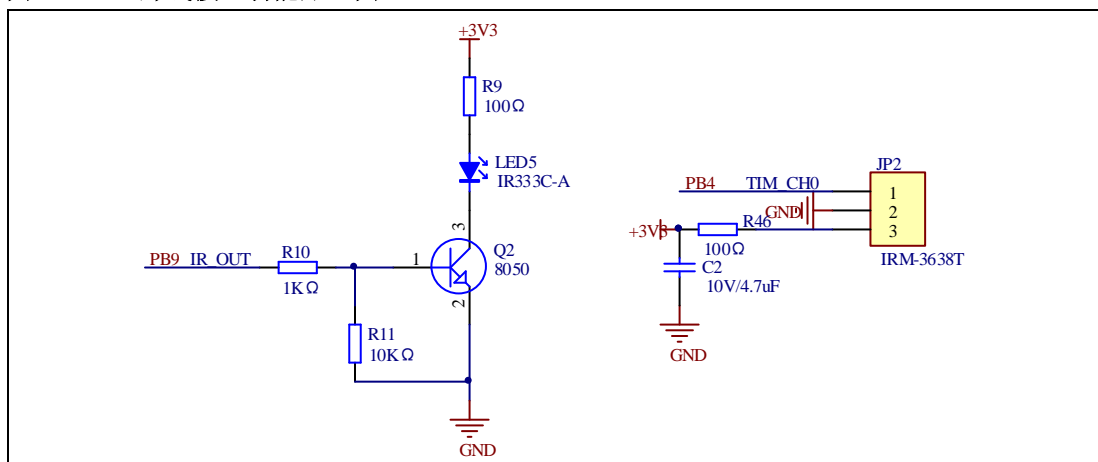
4.11. 比较器

图 4-11 比较器功能原理图



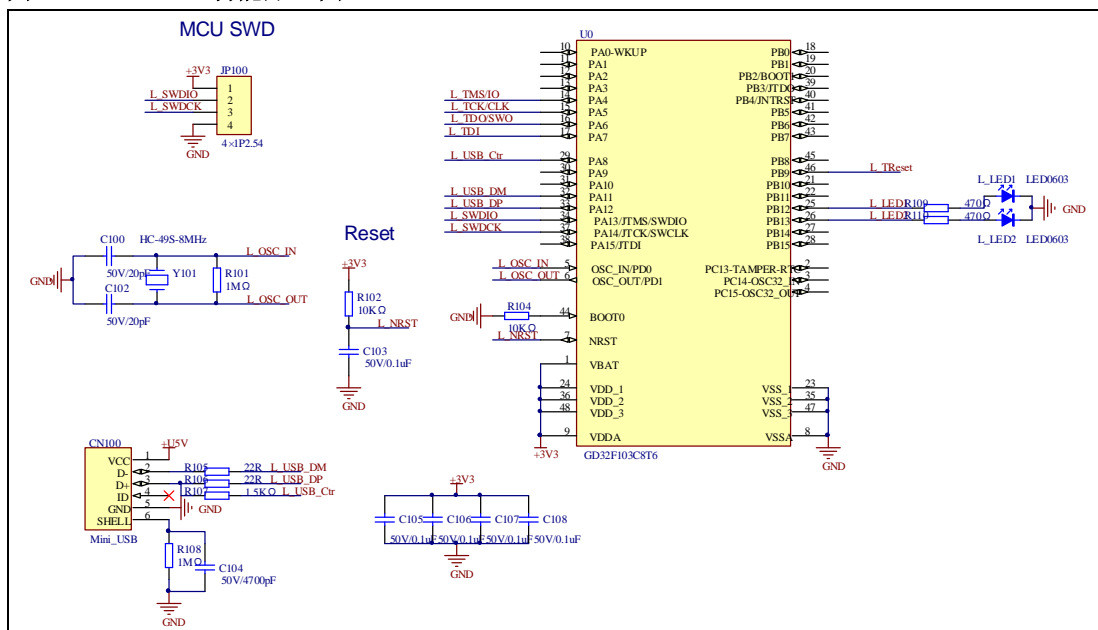
4.12. 红外线接口

图 4-12 红外线接口功能原理图



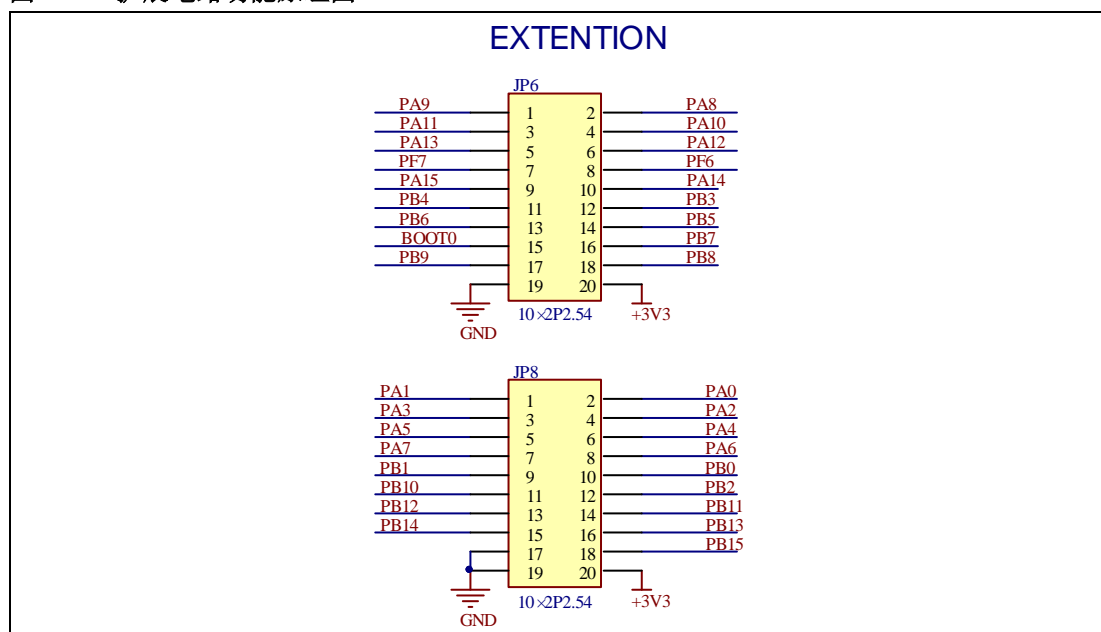
4.13. GD-Link

图 4-13 GD-Link功能原理图



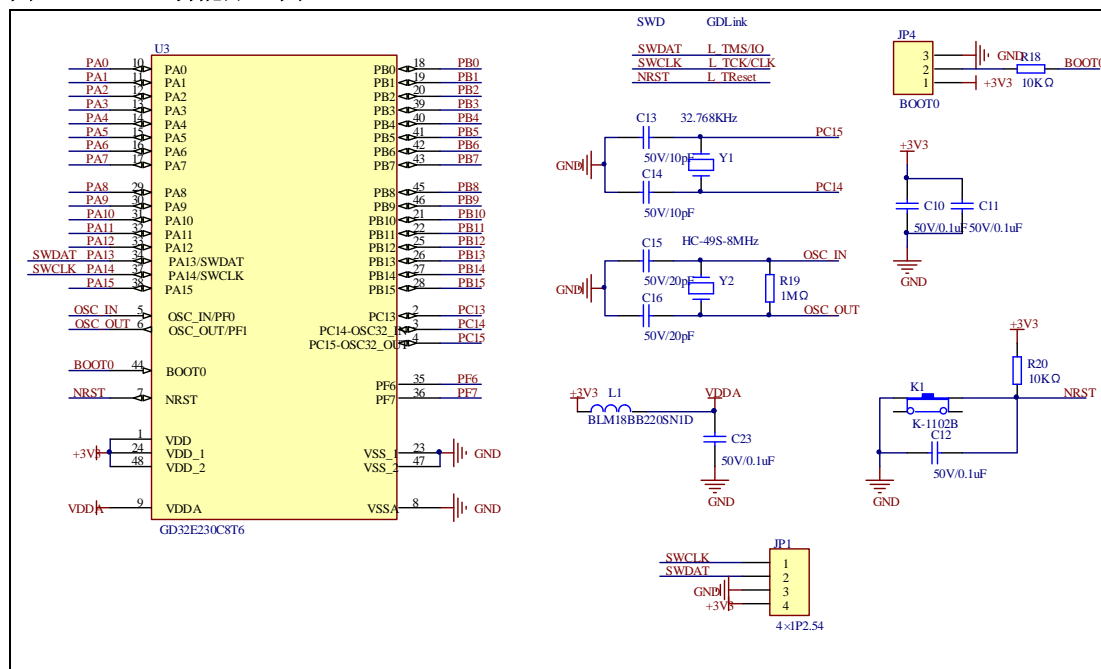
4.14. 扩展电路

图 4-14 扩展电路功能原理图



4.15. MCU

图 4-15 MCU功能原理图



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

GD32E230C-EVAL 开发板上有 4 个 LED。LED1, LED2, LED3 和 LED4 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_Runing_Led>到开发板上，LED1 到 LED4 每隔 200ms 依次点亮，然后全部熄灭，200ms 后，又重复这个过程。

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

GD32E230C-EVAL 开发板有三个按键和四个 LED。这三个按键是 Reset 按键，Tamper 按键，Wakeup 按键，LED1, LED2, LED3 和 LED4 通过 GPIO 控制。

这个例程将讲述怎么使用 Tamper 按键控制 LED2。当按下 Tamper 按键，将检测 IO 端口的输入值，如果输入值为 0，将等待 50ms。再次检测 IO 端口的输入值。如果数值仍然为 0 表示按键成功按下，并点亮 LED2。

5.2.2. DEMO 执行结果

下载程序<02_GPIO_KeyBoard_Polling_mode>到开发板上，按下 Tamper 按键，LED2 将被点亮。再按下 Tamper 按键，LED2 将会熄灭。

5.3. GPIO 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 实现控制 LED 和按键
- 学习使用 EXTI 产生外部中断

GD32E230C-EVAL 有三个按键和四个 LED 灯。这三个按键分别为 Reset 按键、Wakeup 按键和 Tamper 按键。LED1, LED2, LED3 和 LED4 通过 GPIO 控制。

这个例程实现怎样使用 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_GPIO_KeyBoard_Interrupt_mode>到开发板, 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, 并将 USB 线连到开发板的 USART 上。例程将输出“usart printf test example!”到超级终端。

通过串口输出的信息如下图所示。

```
usart printf test example!
```

5.5. 串口中断收发

5.5.1. DEMO 目的

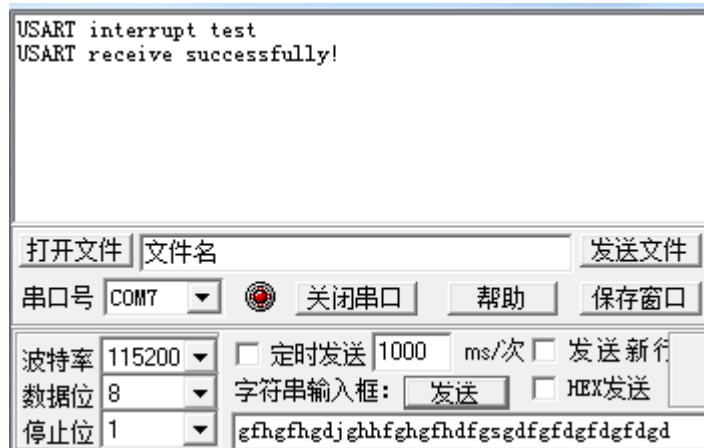
这个例程包括了 GD32 MCU 的以下功能:

- 学习串口的发送和接收中断的使用

5.5.2. DEMO 执行结果

下载程序 < 05_USART_HyperTerminal_Interrupt > 到开发板并运行。首先, COM 将首先输出“USART interrupt test”到超级终端并等待接收数据。接收到的数据将存在数组 receiver_buffer, 接收缓冲区的最大字节数为 receivesize=32。在发送和接收完成后, 将

比较 rxcount 和 receivesize 的值，如果结果相同，COM 将输出“USART receive successfully!”。程序输出如下：



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：

- 学习串口 DMA 发送和接收功能的使用

5.6.2. DEMO 执行结果

下载程序 < 06_USART_DMA > 到开发板并运行。首先，COM 将首先输出“a usart dma function test example!”

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

”到超级终端并循环等待接收数据。然后，每次输入的数据大于等于 10 个字节，USART 将向超级终端发送 10 个字节。程序输出如下：

```
a usart dma function test example!  
USART DMA receive and transmit example, please input 10 bytes:  
gfhgfhgdgf
```

5.7. 定时器触发模数转换

5.7.1. DEMO 目的

该例程包含 GD32 MCU 以下功能：

- 学会使用 ADC 转换模拟量为数字量
- 学会使用定时器生成比较事件

■ 学会使用 LCD 液晶屏来显示 ADC 转换结果

定时器 0 的比较事件 0 触发 ADC 转换, ADC 转换的结果将随着模拟值输入的改变而改变。转换结果由 DMA 搬运到 SRAM 中, 最后在 LCD 上画出相应曲线。

5.7.2. DEMO 执行结果

将<07_ADC_conversion_triggered_by_timer>程序下载至评估板, 定时器 0 的比较捕获事件 0 触发 ADC 转换, 调节电位器改变输入, ADC 的转换结果将会改变, 同时可以在 LCD 液晶屏上看到转换的电压值曲线随着电位器调节而变化。

5.8. 比较器输出获取指示灯

5.8.1. 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

在评估板上有两个比较器, 每个比较器有两个输入端。本例程中使用比较器 0, 其中一个输入设置为 3.3V 电压, 另一个是 1/4 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。

5.8.2. DEMO 执行结果

下载程序<08_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。

5.9. I2C 访问 EEPROM

5.9.1. 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

5.9.2. DEMO 执行结果

下载程序<09_I2C_EEPROM >到开发板上。将开发板的 COM 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!
```

5.10. QSPI FLASH

5.10.1. 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。

5.10.2. 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 <10_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。

```
#####
GD32E230C_EVAL_1.0 System is Starting up...
GD32E230C_EVAL_1.0 Flash:64K
GD32E230C_EVAL_1.0 The CPU Unique Device ID:[FFFFFFFF-FFFFFFFF-FFFFFFFF]
GD32E230C_EVAL_1.0 SPI Flash:GD25Q16 configured...
The Flash_ID:0xC84015

Write to tx_buffer:
0x00 0x01 0x02 0x03 0x04 0x05 0x06 0x07 0x08 0x09 0x0A 0x0B 0x0C 0x0D 0x0E 0x0F 0x10
0x11 0x12 0x13 0x14 0x15 0x16 0x17 0x18 0x19 0x1A 0x1B 0x1C 0x1D 0x1E 0x1F 0x20 0x21
0x22 0x23 0x24 0x25 0x26 0x27 0x28 0x29 0x2A 0x2B 0x2C 0x2D 0x2E 0x2F 0x30 0x31 0x32
0x33 0x34 0x35 0x36 0x37 0x38 0x39 0x3A 0x3B 0x3C 0x3D 0x3E 0x3F 0x40 0x41 0x42 0x43
0x44 0x45 0x46 0x47 0x48 0x49 0x4A 0x4B 0x4C 0x4D 0x4E 0x4F 0x50 0x51 0x52 0x53 0x54
0x55 0x56 0x57 0x58 0x59 0x5A 0x5B 0x5C 0x5D 0x5E 0x5F 0x60 0x61 0x62 0x63 0x64 0x65
0x66 0x67 0x68 0x69 0x6A 0x6B 0x6C 0x6D 0x6E 0x6F 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!
```

5.11. SPI 驱动 LCD 液晶屏

5.11.1. DEMO 目的

这个例程包括了 GD32 MCU 的以下功能：

- 学习使用如何利用 SPI 驱动 TFT LCD 屏并显示

GD32E230C-EVAL 开发板上有一个 TFT LCD 显示屏，它支持 SPI 接口。在这个 Demo 中，分别进行了文字测试、数字测试、画图测试和颜色测试，最终在 LCD 屏上显示。

5.11.2. DEMO 执行结果

下载程序<11_SPI_TFT_LCD_Driver>到开发板并运行。所有的 LED 先被打开然后关闭，接着 LCD 屏循环显示 GUI 测试项目。



5.12. 音频播放器

5.12.1. 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

GD32E230C-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。

5.12.2. DEMO 执行结果

下载程序 <12_I2S_Audio_Player> 到开发板并运行，将耳机插到 J1 端口，即可听到播放的音频文件声音。

5.13. RCU 时钟输出

5.13.1. 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 EXTI 产生外部中断
- 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

5.13.2. DEMO 执行结果

下载程序<13_RCU_Clock_Out>到开发板上并运行。将开发板的 COM 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，LED 灯会被依次循环点亮，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  
CK_OUT: IRC8M  
CK_OUT: IRC28M  
CK_OUT: IRC40K  
CK_OUT: LXTAL  
CK_OUT: HXTAL  
CK_OUT: PLL/2
```

5.14. PMU 睡眠模式唤醒

5.14.1. 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

5.14.2. DEMO 执行结果

下载程序<14_PMU_sleep_wakeup>到开发板上，并将串口线连到开发板的 COM 上。板上电后，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。

5.15. RTC 实时时钟

5.15.1. DEMO 目的

这个例程包括了 GD32 MCU 的以下功能：

- 学习使用 RTC 模块实现实时时钟的功能
- 学习使用 LCD 模块实现显示时间的功能

5.15.2. DEMO 执行结果

下载程序到<15_RTC_Calendar>开发板上并运行。当程序运行时，首先 LED1 到 LED4 这 4 个 LEDs 会点亮再熄灭，然后 LCD 液晶屏会显示开发板的信息，以及实时时间。每次按下 Wakeup 按键，会校时到 2018-05-13, 12: 00: 00。

5.16. 红外收发器

5.16.1. DEMO 目的

这个例程包括了 GD32 MCU 的以下功能：

- 学习使用通用定时器输出 PWM 的方法
- 学习使用通用定时器更新中断的方法
- 学习使用通用定时器捕获中断功能
- 学习使用定时器 TIMER15 和 TIMER16 实现红外功能

5.16.2. DEMO 执行结果

下载程序<16_IRInfrared_Transceiver>到开发板上并运行。当程序运行时，如果红外接收器接收到正确信号，可以看到 LED1~LED4 依次点亮，否则，可以看到 LED1~LED4 同时翻转，即同时点亮和熄灭。

5.17. TIMER 呼吸灯

5.17.1. 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

5.17.2. DEMO 执行结果

下载程序<17_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 LED1 由暗变亮，由亮变暗，往复循环，就像人的呼吸一样有节奏。

6. 版本更新历史

表 6-1 版本更新历史

版本号	描述	日期
1.0	初始发布版本	2018 年 11 月 1 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 its 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 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 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.