

GigaDevice Semiconductor Inc.

GD32190R-EVAL 评估板

用户指南

V3.1

(2021 年 12 月 31 日)

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.....	8
4.1. 供电电源.....	8
4.2. 启动方式选择	8
4.3. LED 指示灯	9
4.4. 按键	9
4.5. 串口	10
4.6. 模数转换器.....	10
4.7. 数模转换器.....	11
4.8. I2S.....	11
4.9. I2C.....	11
4.10. SPI.....	12
4.11. OPA	12
4.12. CAN	13
4.13. SLCD.....	13
4.14. 比较器.....	14
4.15. HDMI-CEC	14
4.16. TSI	14
4.17. 红外线接口	15
4.18. GD-Link	15
4.19. 扩展电路.....	16
4.20. MCU	16
5. 例程使用指南.....	17
5.1. GPIO 流水灯	17
5.1.1. DEMO 目的	17

5.1.2. DEMO 执行结果.....	17
5.2. GPIO 按键轮询模式	17
5.2.1. DEMO 目的	17
5.2.2. DEMO 执行结果.....	17
5.3. EXTI 按键中断模式	18
5.3.1. DEMO 目的	18
5.3.2. DEMO 执行结果.....	18
5.4. 串口打印.....	18
5.4.1. DEMO 目的	18
5.4.2. DEMO 执行结果.....	18
5.5. 串口中断收发	19
5.5.1. DEMO 目的	19
5.5.2. DEMO 执行结果.....	19
5.6. 串口 DMA 收发	19
5.6.1. DEMO 目的	19
5.6.2. DEMO 执行结果.....	19
5.7. 定时器触发模数转换	20
5.7.1. DEMO 目的	20
5.7.2. DEMO 执行结果.....	20
5.8. 数模转换.....	20
5.8.1. DEMO 目的	20
5.8.2. DEMO 执行结果.....	20
5.9. 比较器输出获取指示灯	21
5.9.1. DEMO 目的	21
5.9.2. DEMO 执行结果.....	21
5.10. I2C 访问 EEPROM	21
5.10.1. DEMO 目的	21
5.10.2. DEMO 执行结果.....	21
5.11. SPI 驱动 FLASH	22
5.11.1. DEMO 目的	22
5.11.2. DEMO 执行结果.....	22
5.12. HDMI_CEC 通信	23
5.12.1. DEMO 目的	23
5.12.2. DEMO 执行结果.....	23
5.13. 音频播放器	24
5.13.1. DEMO 目的	24
5.13.2. DEMO 执行结果.....	24
5.14. RCU 时钟输出	24
5.14.1. DEMO 目的	24

5.14.2.	DEMO 执行结果	24
5.15.	PMU 睡眠模式唤醒.....	25
5.15.1.	DEMO 目的	25
5.15.2.	DEMO 执行结果	25
5.16.	RTC 实时时钟.....	25
5.16.1.	DEMO 目的	25
5.16.2.	DEMO 执行结果	25
5.17.	红外收发器	25
5.17.1.	DEMO 目的	25
5.17.2.	DEMO 执行结果	25
5.18.	TIMER 呼吸灯	26
5.18.1.	DEMO 目的	26
5.18.2.	DEMO 执行结果	26
5.19.	TSI 触摸按键	26
5.19.1.	DEMO 目的	26
5.19.2.	DEMO 执行结果	26
5.20.	OPA 运放.....	26
5.20.1.	DEMO 目的	26
5.20.2.	DEMO 执行结果	27
5.21.	SLCD 液晶显示	27
5.21.1.	DEMO 目的	27
5.21.2.	DEMO 执行结果	27
5.22.	CAN 通信	27
5.22.1.	DEMO 目的	27
5.22.2.	DEMO 执行结果	27
6.	版本更新历史.....	29

图

图 4-1. 供电电源原理图	8
图 4-2. 启动方式选择原理图	8
图 4-3. LED 功能原理图	9
图 4-4. 按键功能原理图	9
图 4-5. 串口功能原理图	10
图 4-6. 模数转换器功能原理图	10
图 4-7. 数模转换器功能原理图	11
图 4-8. I2S 功能原理图	11
图 4-9. I2C 功能原理图	11
图 4-10. SPI 功能原理图	12
图 4-11. OPA 功能原理图	12
图 4-12. CAN 功能原理图	13
图 4-13. SLCD 功能原理图	13
图 4-14. 比较器功能原理图	14
图 4-15. HDMI-CEC 功能原理图	14
图 4-16. TSI 功能原理图	14
图 4-17. 红外线接口功能原理图	15
图 4-18. GD-Link 功能原理图	15
图 4-19. 扩展电路功能原理图	16
图 4-20. MCU 功能原理图	16

表

表 2-1. 引脚分配	6
表 4-1. 启动方式配置	8
表 6-1. 版本更新历史	29

1. 简介

GD32190R-EVAL-V1.2 评估板使用 GD32F190R8T6 作为主控制器。该评估板为采用 Cortex™-M3 内核的 GD32F1x0 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 mini-USB 接口提供 5V 电源。提供包括扩展引脚在内的以及 SWD、Reset、Boot、User button key、LED、I2C、I2S、USART、RTC、SLCD、HDMI-CEC、TSI、IFRP LED、IFRP Receiver、RTC、SPI、ADC、DAC 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

功能	引脚	描述
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

功能	引脚	描述
	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. 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，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 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 GD32F1x0_AddOn.3.2.0.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\Library\Firmware\GigaDevice.GD32F1x0_DFP.3.2.0.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

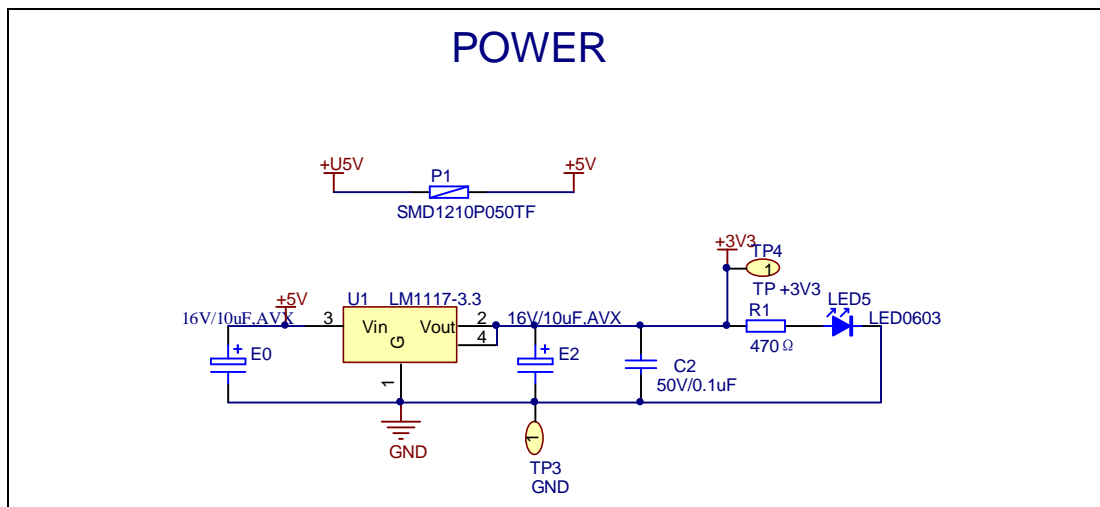
中的 Destination Folder 那一栏选择 Keil uVision5 软件的安装目录，如 C:\Keil_v5，然后在 Option for Target 的 Device 选择对应的器件，同时在 Option for Target 的 C/C++ 中添加路径 C:\Keil_v5\ARM\Pack\ARM\CMSIS\4.2.0\CMSIS\Include。

3、如果使用 IAR 打开工程，安装 IAR_GD32F1x0_ADDON.3.2.0.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

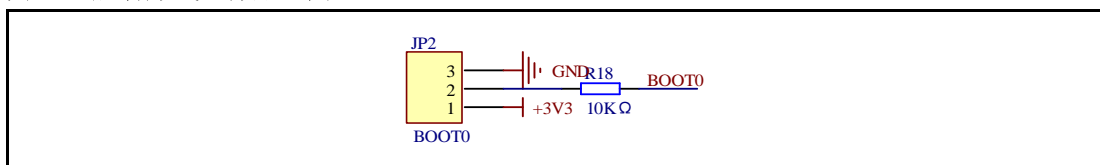
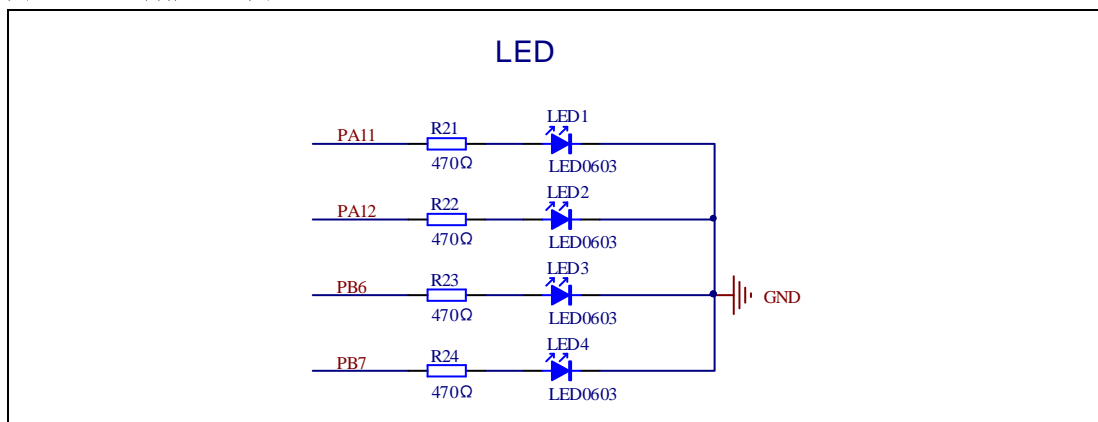


表4-1. 启动方式配置

BOOT1	BOOT0	启动模式
默认值	2-3	主 FLASH 存储器
	1-2	系统存储器
经过 ISP 修改	1-2	片上 SRAM

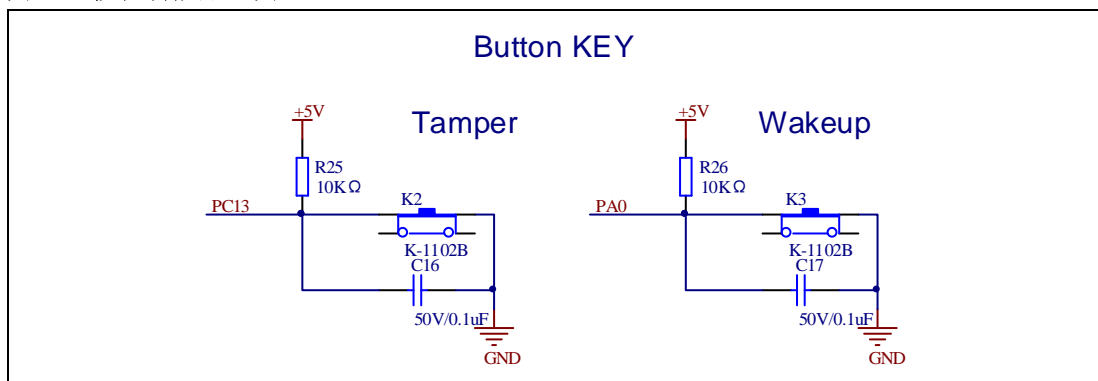
4.3. LED 指示灯

图4-3. LED功能原理图



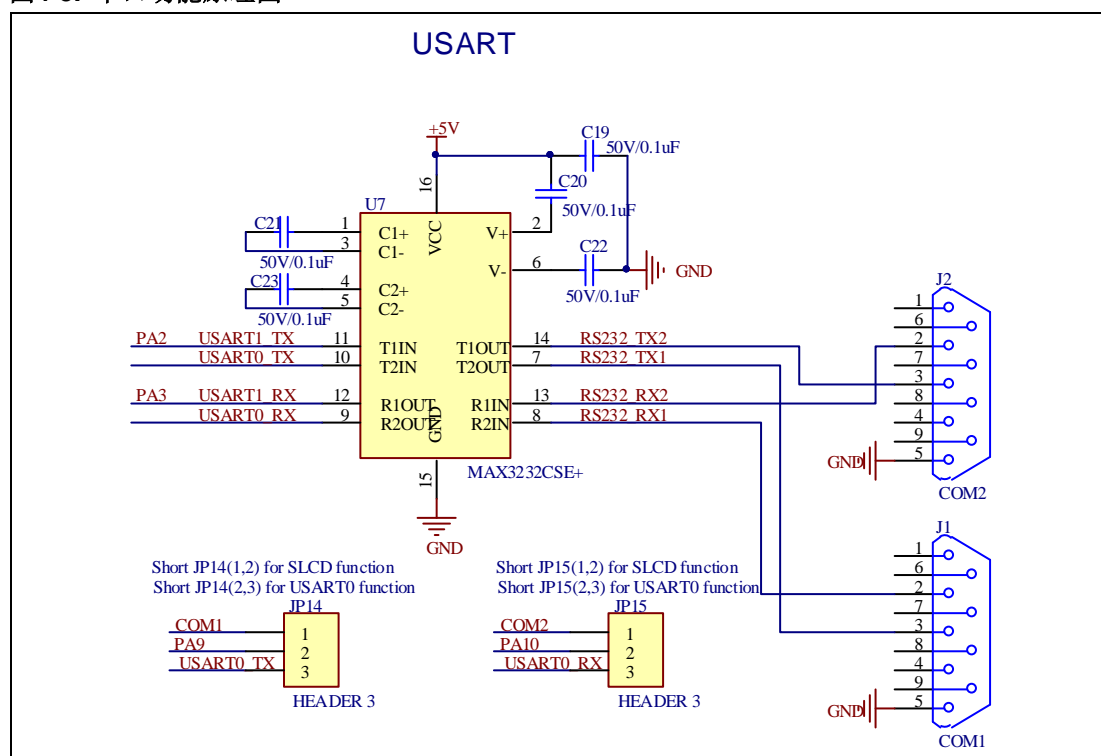
4.4. 按键

图4-4. 按键功能原理图



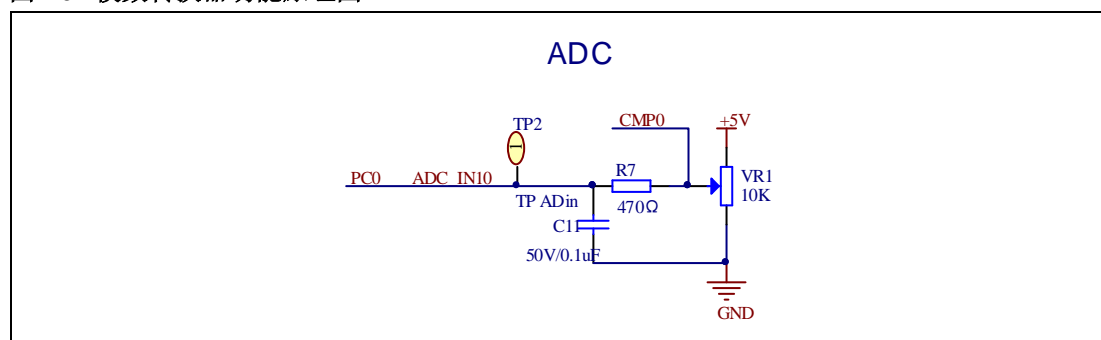
4.5. 串口

图4-5. 串口功能原理图



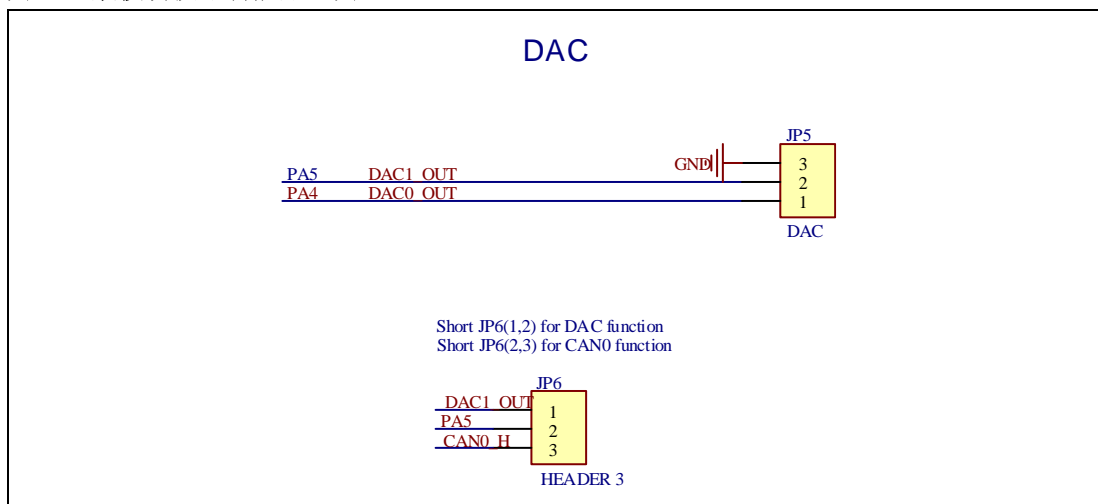
4.6. 模数转换器

图4-6. 模数转换器功能原理图



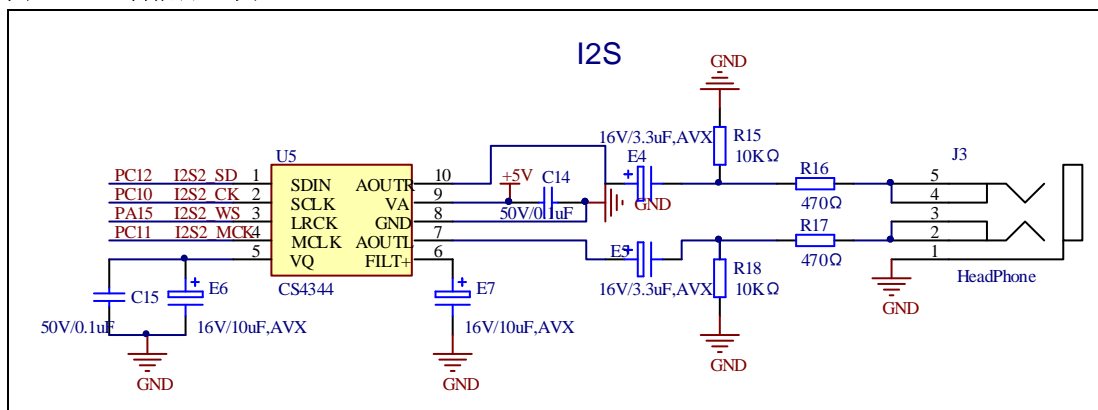
4.7. 数模转换器

图4-7. 数模转换器功能原理图



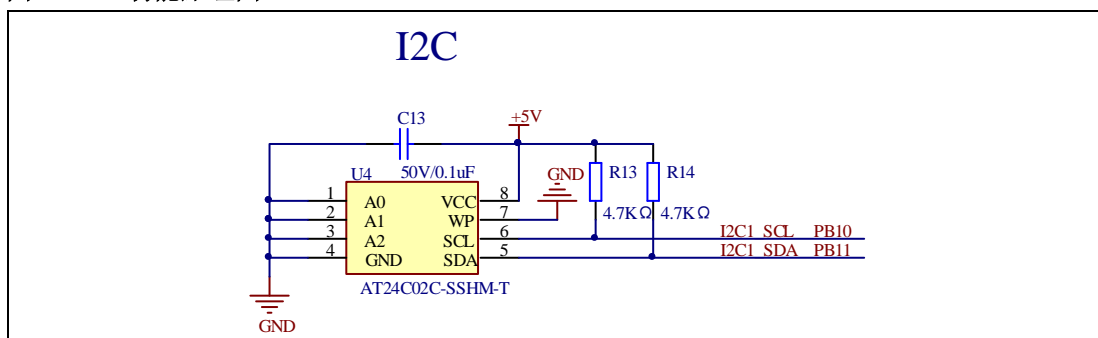
4.8. I2S

图4-8. I2S功能原理图



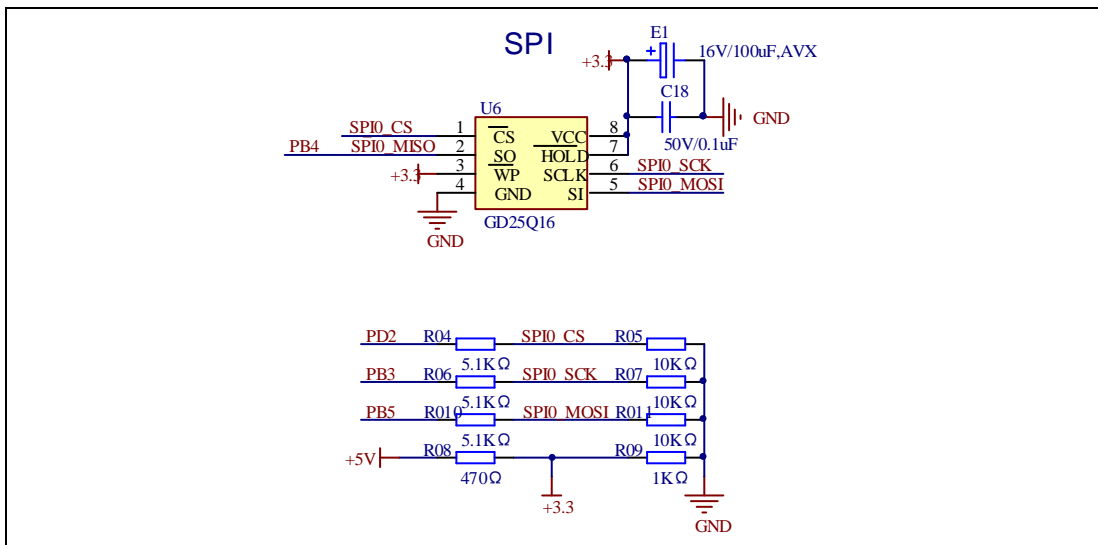
4.9. I2C

图4-9. I2C功能原理图



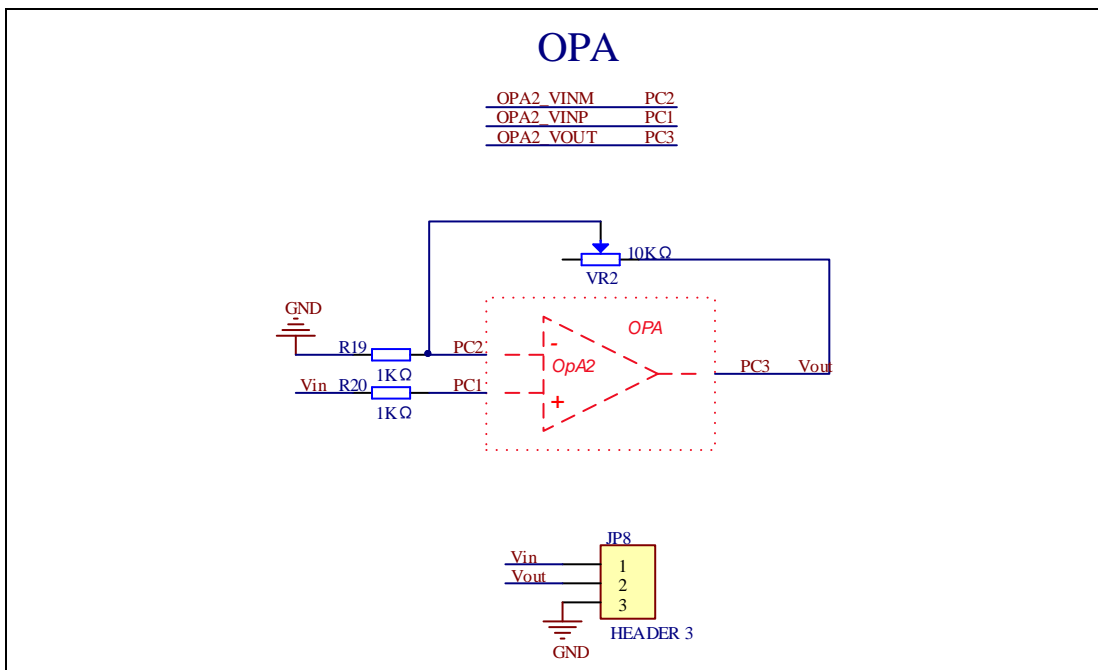
SPI

图4-10. SPI功能原理图



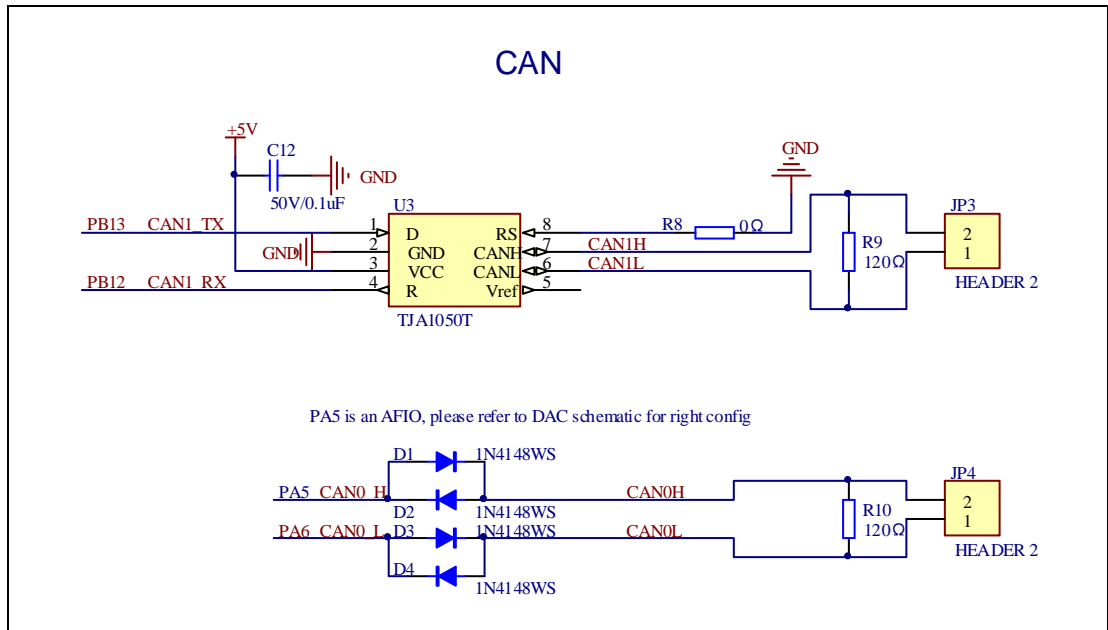
OPA

图4-11. OPA功能原理图



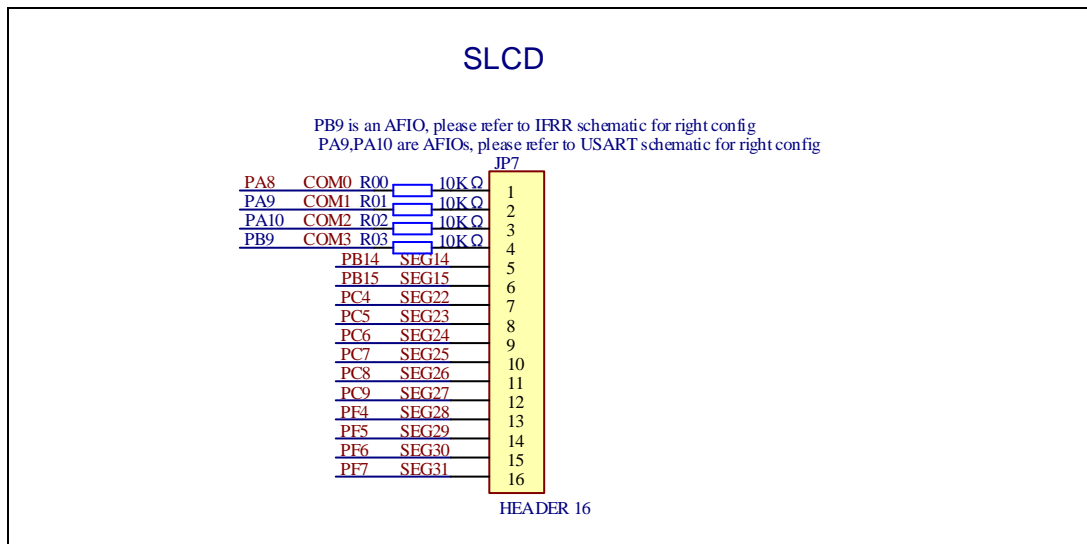
4.12. CAN

图4-12. CAN功能原理图



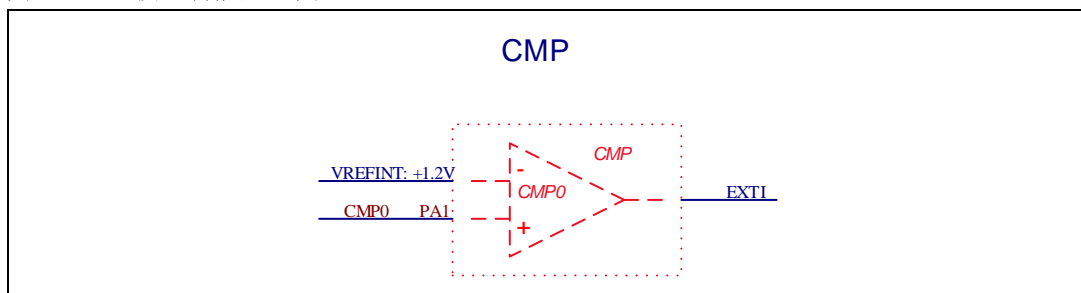
4.13. SLCD

图4-13. SLCD功能原理图



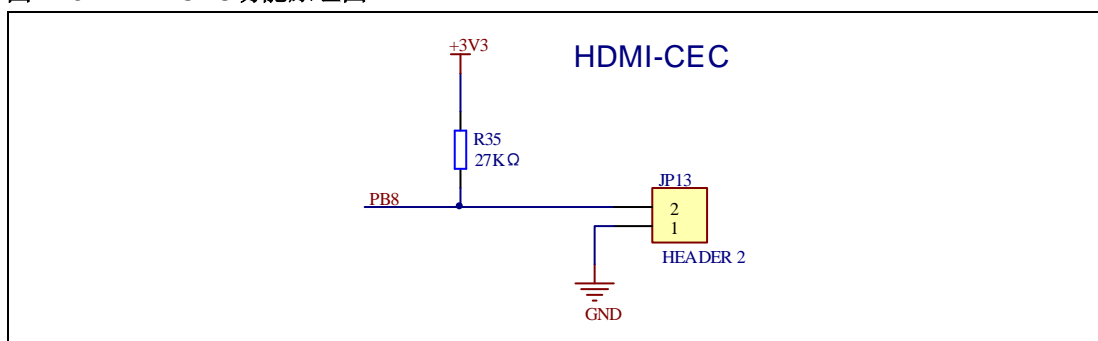
4.14. 比较器

图4-14. 比较器功能原理图



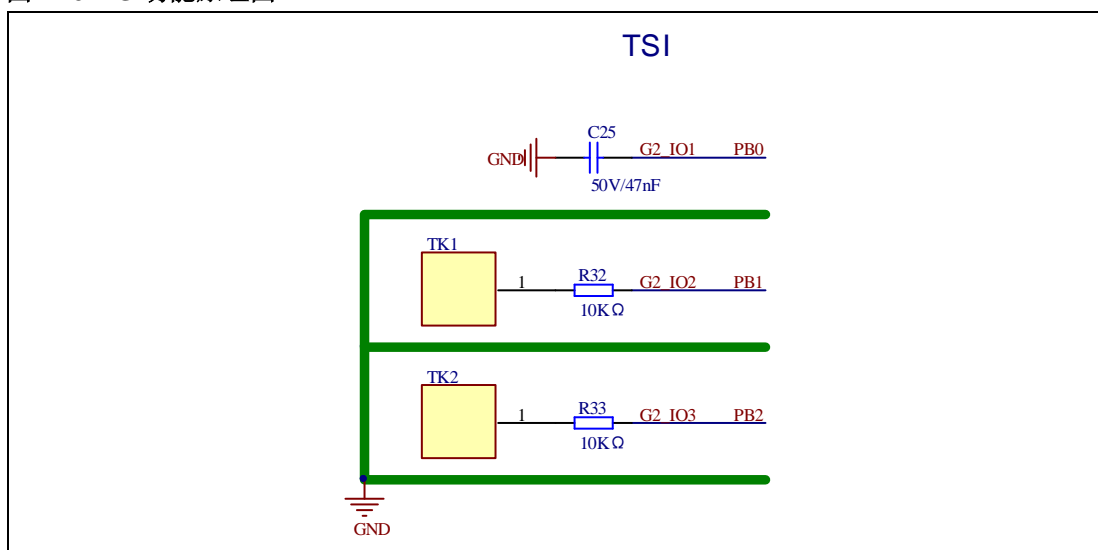
4.15. HDMI-CEC

图4-15. HDMI-CEC功能原理图



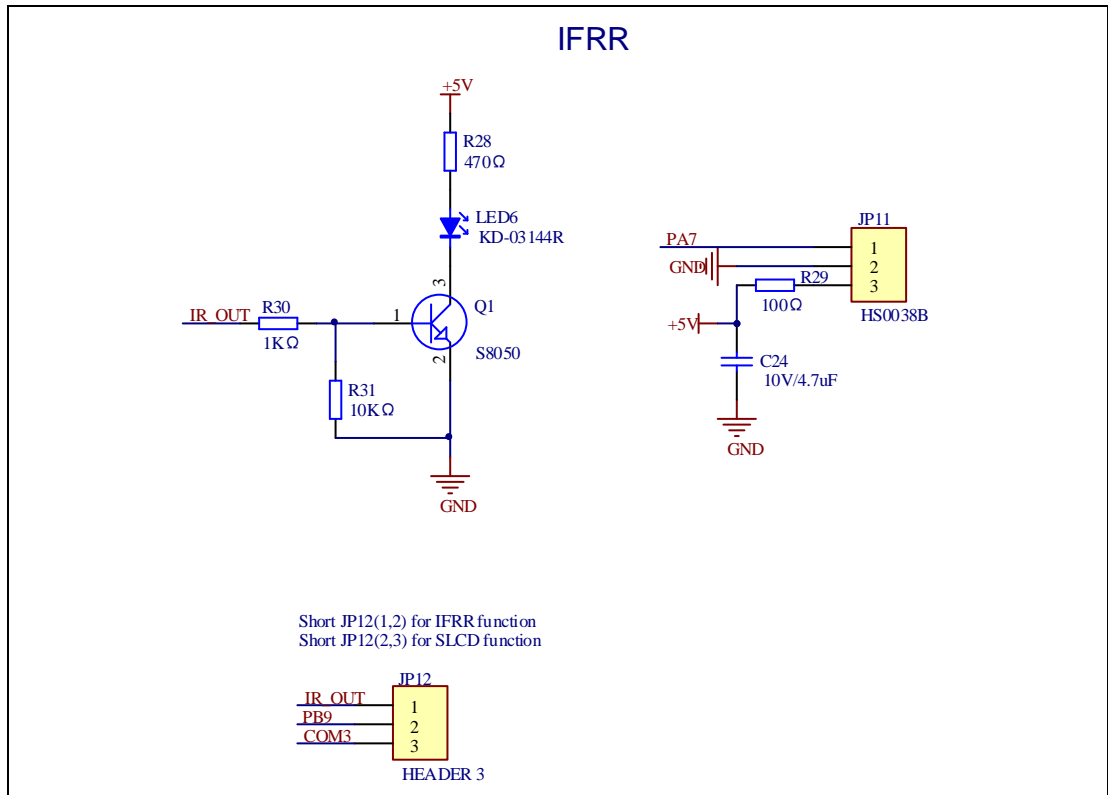
4.16. TSI

图4-16. TSI功能原理图



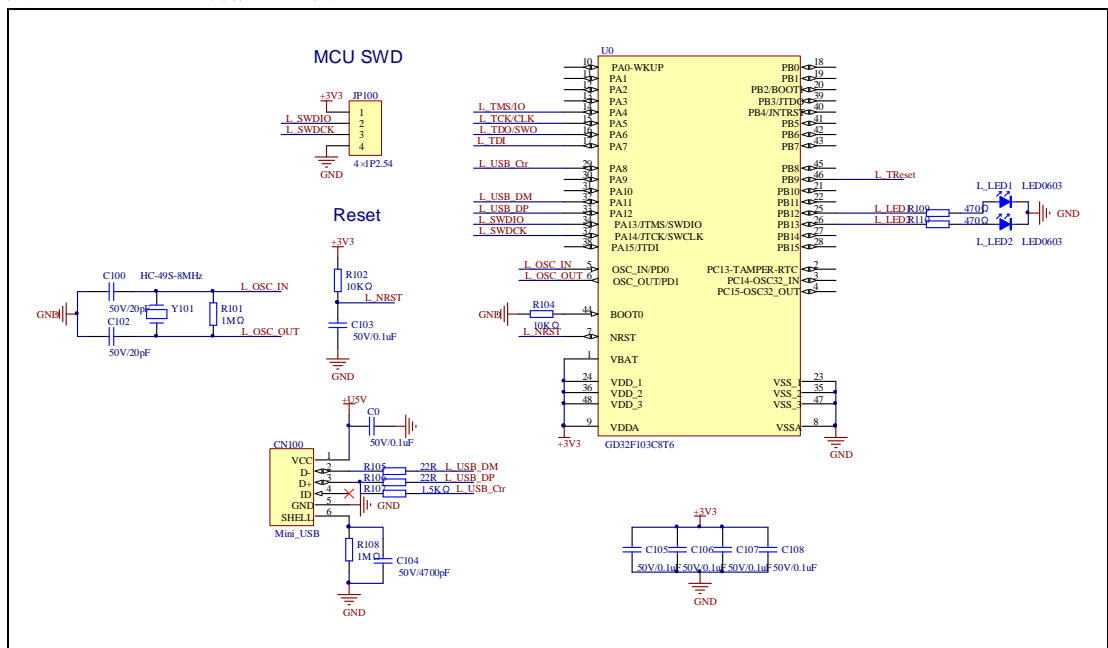
4.17. 红外线接口

图4-17. 红外线接口功能原理图



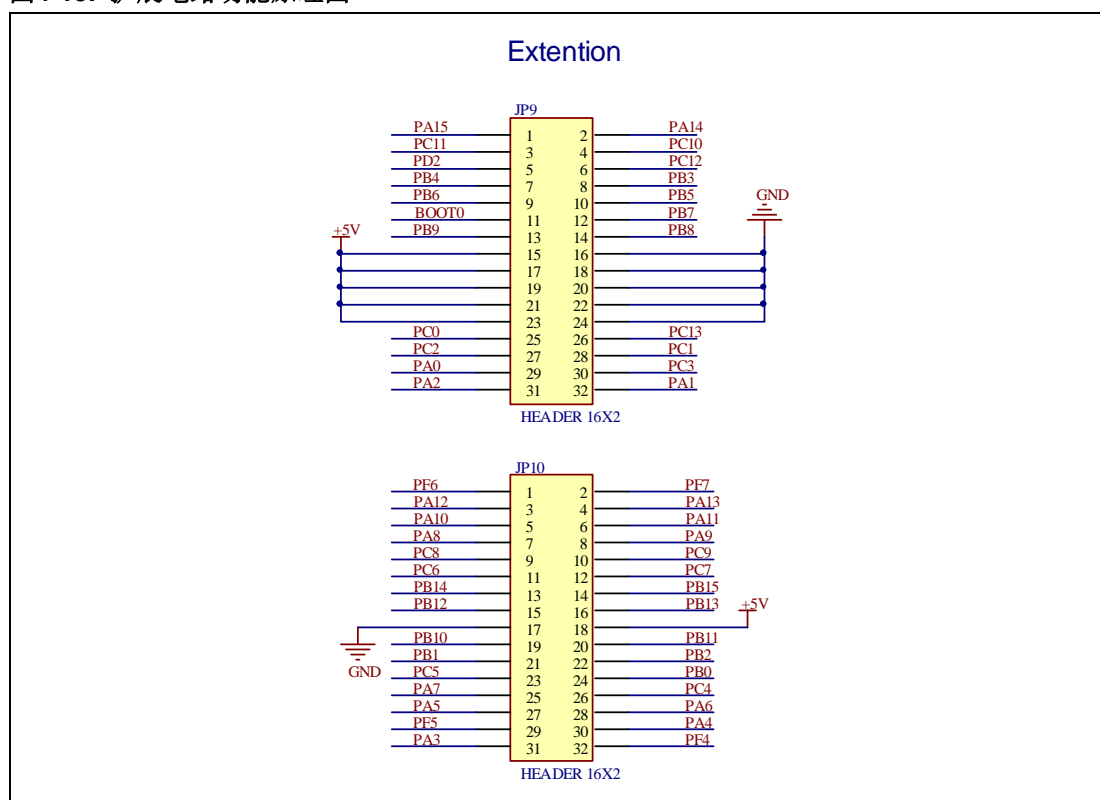
4.18. GD-Link

图4-18. GD-Link功能原理图



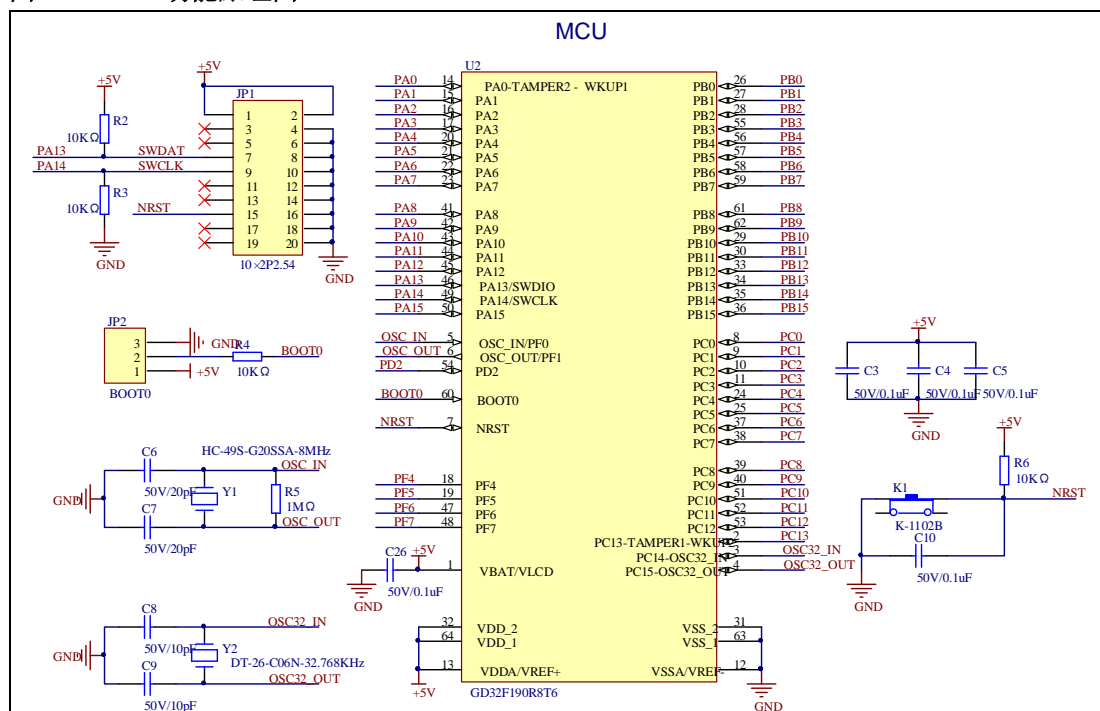
4.19. 扩展电路

图4-19. 扩展电路功能原理图



4.20. MCU

图4-20. 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

GD32190R-EVAL-V1.2 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_Running_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

GD32190R-EVAL-V1.2 开发板有三个按键和四个 LED。这四个按键是 Reset 按键，Tamper 按键，Wakeup 按键，LED1，LED2，LED3 和 LED4 通过 GPIO 控制着。

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_Key_Polling_mode>到开发板上，首先，所有的 LED 亮灭一次用于测试。然后，按下 Tamper 按键，LED2 将被点亮。再按下 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 实现控制 LED 和按键
- 学习使用 EXTI 产生外部中断

GD32190R-EVAL-V1.2 有三个按键和四个 LED 灯。这四个按键分别为 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_EXTI_Key_Interrupt_mode>到开发板，首先，所有的 LED 亮灭一次用于测试。然后，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

5.4.2. DEMO 执行结果

下载程序<04_USART_Printf>到开发板，并将串口线连到开发板的 COM 上。例程首先将输出“please press the Tamper key”到超级终端。按下 Tamper 键，串口继续输出“USART printf example”。

通过串口输出的信息如下图所示。

```
please press the Wakeup 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，所有的 LED 灯亮灭一次用于测试。然后，COM 将首先输出数组 tx_buffer 的内容（从 0x00 到 0xFF）到超级终端并等待接收数据。接收到的数据将存在数组 rx_buffer，接收缓冲区的最大字节数为 BUFFER_SIZE。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 和 LED2 灯亮，LED3 和 LED4 灯灭，否则，LED3 和 LED4 灯亮，LED1 和 LED2 灯灭。

程序输出如下：

```
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，所有的 LED 灯亮灭一次用于测试。然后，COM 将首先输出数组 tx_buffer 的内容（从 0x00 到 0xFF）到超级终端并等待接收数据。接收到的数据将存在数组 rx_buffer，接收缓冲区的最大字节数为 BUFFER_SIZE。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 和 LED4 灯亮，否则 LED1、LED2、LED3 和 LED4 灯灭。程序输出如下：

```

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. 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 1 的比较事件 1 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 1 的比较捕获事件 1 触发 ADC 转换，调节电位器改变输入，ADC 的转换结果将会改变，并通过 DMA 从 ADC 的 RDATA 寄存器搬运到 SRAM 中，然后在上位机端口打印。

5.8. 数模转换

5.8.1. DEMO 目的

该例程包含 GD32 MCU 以下功能：

- 学会使用 DAC 在 DAC 输出端生成不同的电压

5.8.2. DEMO 执行结果

将<08_DAC_Output_Voltage_Value>程序下载至评估板，将数字输出值设置为 0x7ff0，它的转换电压应该是 $V_{REF}/2$ ，使用电压表测量 PA4 引脚，得知其值为 2.548V。

5.9. 比较器输出获取指示灯

5.9.1. 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 GD32190R-EVAL-V1.2 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，其中一个输入是滑动变阻器输出电压，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 LED 灯就会执行相应动作。

5.9.2. DEMO 执行结果

下载程序<09_Comparator_Obtain_Brightness>到 GD32190R-EVAL-V1.2 开发板中，改变滑动变阻器输出电压，比较其与参考电压大小，如果它大于参考电压，LED1 亮，LED2 灭，否则，LED1 灭，LED2 亮。

5.10. I2C 访问 EEPROM

5.10.1. 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

5.10.2. DEMO 执行结果

下载程序<10_I2C_EEPROM>到开发板上。将开发板的 COM2 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!

```

5.11. SPI 驱动 FLASH

5.11.1. 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。

5.11.2. DEMO 执行结果

把电脑串口线连接到开发板的 COM1 口，设置串口助手软件波特率为 115200，数据位 8 位，停止位 1 位。将程序<11_SPI_FLASH>烧录到开发板，通过串口调试助手软件可观察运行状况，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 1 打印出“SPI-GD25Q40 Test Passed!”，否则，串口打印出“Err: Data Read and Write aren't Matching.”。最后，LED 等依次循环点亮。使用 COM1 口时，需要将 JP14 和 JP15 跳线到 USART0。下图是实验结果图。


```

#####
GD32190R-EVAL-V1.2 System is Starting up...
GD32190R-EVAL-V1.2 Program Compile time: (Nov  8 2019 - 11:17:51)
GD32190R-EVAL-V1.2 SystemCoreClock: 72000000Hz
GD32190R-EVAL-V1.2 Flash: 64K
GD32190R-EVAL-V1.2 The CPU Unique Device ID: [524743-32303434-39533941]
GD32190R-EVAL-V1.2 SPI Flash: GD25Q40 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-GD25Q40 Test Passed!

```

5.12. HDMI_CEC 通信

5.12.1. DEMO 目的

这个例程包括了 GD32 MCU 的以下功能：

- 学习 HDMI-CEC 的通信功能

通信过程中通过按键中断发送方将数据发给接收方，接收方在 CEC 中断中进行数据接收。整个通信过程中不作错误处理。

5.12.2. DEMO 执行结果

此例程需要准备两个开发板，一个作为发送方，另一个作为接收方。先使用杜邦线将两块开发板上的 HDMI-CEC 的 CEC 总线（PB8）和地线（GND）两个引脚连接起来，然后下载程序 <12_HDMI_CEC_HostSlaveCommunication> 到开发板上运行。当程序运行时，首先开发板的 SLCD 显示的都是数据 0，按下其中一块开发板的 TAMPER 键，另一块开发板上 SLCD 的数

字会递增，这说明一次数据传输结束。每次递增到 9 后会清 0 重新递增；按下 WAKEUP 键后，数字会递减，这同样说明一次数据传输结束。每次递减到 0 后会回到数字 9 重新递减。

5.13. 音频播放器

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

GD32190R-EVAL-V1.2 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，将耳机插到 J3 端口，即可听到播放的音频文件声音。

5.14. RCU 时钟输出

5.14.1. 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

5.14.2. DEMO 执行结果

下载程序<14_RCU_Clock_Out>到开发板上并运行。将开发板的 COM1 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 USER 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 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 睡眠模式唤醒

5.15.1. 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

5.15.2. DEMO 执行结果

下载程序<15_PMU_sleep_wakeup>到开发板上，并将串口线连到开发板的 COM 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。

5.16. RTC 实时时钟

5.16.1. 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 SLCD 模块实现显示时间的功能

5.16.2. DEMO 执行结果

使用跳线帽 JP12, JP14, JP15 跳线到 SLCD，然后下载程序到<16_RTC_Calendar>开发板上并运行。当程序运行时，段码 SLCD 会显示实时时间，每次按下 Tamper 按键，时分秒，年份，月份日期会依次显示出来。

5.17. 红外收发器

5.17.1. 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

5.17.2. DEMO 执行结果

下载程序<17_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。

5.18. TIMER 呼吸灯

5.18.1. 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

5.18.2. DEMO 执行结果

使用杜邦线连接 TIMER0_CH0（PA8）和 LED1（PA11），然后下载程序 <18_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。

5.19. TSI 触摸按键

5.19.1. DEMO 目的

这个例程包括了 GD32 MCU 的以下功能：

- 学习使用 TSI 模块实现触摸按键的功能

5.19.2. DEMO 执行结果

下载程序 <19_TSI_TouchKey_Leds> 到开发板上并运行。当程序运行时，用手指触摸 TSI 的两个按键，然后相应的 LED1 和 LED2 灯将被点亮。

5.20. OPA 运放

5.20.1. DEMO 目的

这个例程包括了 GD32 MCU 的以下功能：

- 学习使用 OPA 模块实现输入电压的放大功能
- 学会使用 ADC 转换模拟量为数字量

5.20.2. DEMO 执行结果

下载程序<20_OPA_Amplify>到开发板上并运行。当程序运行时，将输入电压接入 JP8 的 pin1 引脚，然后调节滑动变阻器 VR2 来改变输入电压的放大倍数，并由 OPA2 对输入电压进行放大，最后通过 ADC1 模数转换将输出电压值的结果在上位机端口打印。

5.21. SLCD 液晶显示

5.21.1. DEMO 目的

这个例程包括了 GD32 MCU 的以下功能：

- 学习使用 LCD 模块实现显示数字的功能。

5.21.2. DEMO 执行结果

使用跳线帽 JP12 跳线到 LCD，然后下载<21_LCD_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。

5.22. CAN 通信

5.22.1. DEMO 目的

这个例程包括了 GD32 MCU 的以下功能：

- 学习使用在一块板子上进行 CAN0 和 CAN1
- 学习使用 USART 和 PC 通讯

5.22.2. DEMO 执行结果

下载程序<22_CAN_DualCAN>到开发板上并运行。当程序运行时，CAN0 和 CAN1 外设工作在正常模式并且能每隔 1000ms 进行 CAN 帧的自动接收和发送。当 CAN 帧被接收将通过 COM2 打印。开发板上的 JP6 应当与 CAN1 相连，JP4 的 CAN_L 应当与 JP3 的 CAN_L 相连，JP4 的 CAN_H 应当与 JP3 的 CAN_H 相连。

经由串口输出的信息如下。

```
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. 版本更新历史

表6-1. 版本更新历史

版本号	说明	日期
1.0	初稿发布	2017 年 7 月 28 日
2.0	更新文档格式	2019 年 7 月 1 日
3.0	更新升级	2019 年 11 月 20 日
3.1	更新升级	2021 年 12 月 31 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.