

GigaDevice Semiconductor Inc.

GD32M531R-EVAL

Arm[®] Cortex[®]-M33 32-bit MCU

用户指南

1.0 版本

(2026 年 2 月)

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.....	8
4.3. LED 指示灯.....	8
4.4. 按键	8
4.5. UART	9
4.6. ADC	9
4.7. CMP	9
4.8. DAC	10
4.9. I2C.....	10
4.10. SPI.....	10
4.11. CAN	11
4.12. Extension.....	11
4.13. GD-Link.....	12
4.14. MCU.....	13
5. 例程使用指南	14
5.1. GPIO 流水灯	14
5.1.1. DEMO 目的	14
5.1.2. DEMO 执行结果	14
5.2. GPIO 按键轮询模式	14
5.2.1. DEMO 目的	14
5.2.2. DEMO 执行结果	14
5.3. EXTI 按键中断模式	15
5.3.1. DEMO 目的	15
5.3.2. DEMO 执行结果	15

5.4. 串口打印	15
5.4.1. DEMO 目的	15
5.4.2. DEMO 执行结果	15
5.5. 串口中断收发	16
5.5.1. DEMO 目的	16
5.5.2. DEMO 执行结果	16
5.6. 串口 DMA 收发	16
5.6.1. DEMO 目的	16
5.6.2. DEMO 执行结果	16
5.7. ADC 温度传感器_Vrefint	17
5.7.1. DEMO 目的	17
5.7.2. DEMO 执行结果	17
5.8. 定时器触发 ADC 转换	17
5.8.1. DEMO 目的	17
5.8.2. DEMO 执行结果	18
5.9. DAC 输出电压值	18
5.9.1. DEMO 目的	18
5.9.2. DEMO 执行结果	18
5.10. 比较器输出读取指示灯	19
5.10.1. DEMO 目的	19
5.10.2. DEMO 执行结果	19
5.11. I2C 访问 EEPROM	19
5.11.1. DEMO 目的	19
5.11.2. DEMO 执行结果	19
5.12. SPI 四线发 FLASH	20
5.12.1. DEMO 目的	20
5.12.2. DEMO 执行结果	20
5.13. SPI TFT 显示	22
5.13.1. DEMO 目的	22
5.13.2. DEMO 执行结果	22
5.14. RCU 时钟输出	22
5.14.1. DEMO 目的	22
5.14.2. DEMO 执行结果	23
5.15. PMU 睡眠模式唤醒	23
5.15.1. DEMO 目的	23
5.15.2. DEMO 执行结果	23
5.16. TIMER 呼吸灯	23
5.16.1. DEMO 目的	23
5.16.2. DEMO 执行结果	23

5.17.	CAN 网络通信	24
5.17.1.	DEMO 目的	24
5.17.2.	DEMO 执行结果	24
6.	版本历史	25

图

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

表

表 2-1. 引脚分配.....	6
表 6-1. 版本历史.....	25

1. 简介

GD32M531R-EVAL 评估板使用 GD32M531RCT7 作为主控制器。评估板使用 GD-Link Type-C 接口提供 5V 电源。提供包括扩展引脚在内的及 Reset, Boot, Wakeup 按键, USER 按键, LED, ADC, CAN, CMP, DAC, I2C_EEPROM, SPI_LCD, QSPI_FLASH, USART 转 USB 接口等外设资源。更多关于开发板的资料可以查看 GD32M531R-EVAL-V1.2 原理图。

2. 功能引脚分配

表 2-1. 引脚分配

功能	引脚	描述
LED	PD9	LED1
	PD10	LED2
	PD11	LED3
	PD12	LED4
RESET	PN5	K1-Reset
KEY	PA0	K2-Wakeup
	PB0	K3-User
ADC	PC0	ADC0_IN0
CMP	PC1	CMP_IP
	PC11	CMP_IM
	PC12	
	PD4	
	PD5	
DAC	PD4	DAC0_OUT
I2C	PF9	I2C_SCL
	PF10	I2C_SDA
SPI Flash	PD8	SPIFlash_CS
	PF11	SPI_SCK
	PF8	SPI_MOSI
	PF12	SPI_MISO
	PF9	SPI_IO2
	PF10	SPI_IO3
SPI_LCD	PC7	SPILCD_D/C
	PC6	SPILCD_RESET
	PG15	SPILCD_CS
	PF11	SPI_SCK
	PF8	SPI_MOSI
	PF12	SPI_MISO
UART	UART3_TX	PF13
	UART3_RX	PF14

功能	引脚	描述
CAN	CAN_TX	PF13
	CAN_RX	PF14

3. 入门指南

评估板使用 GD-Link Type-C 接口提供 5V 电源。下载程序到评估板需要使用 J-Link 或 GD-Link 工具，在选择了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。

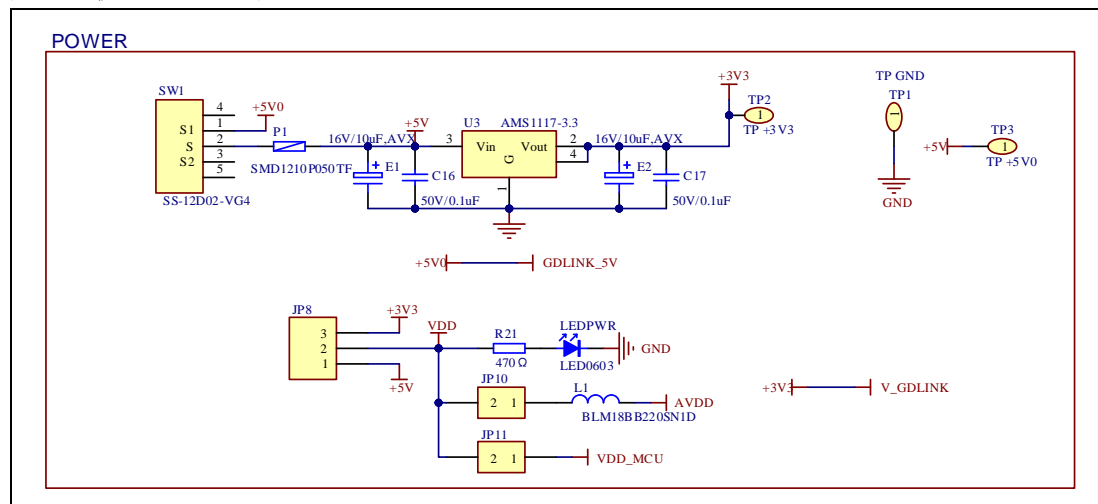
所有例程提供了 Keil、IAR 和 GD32EBuilder 三个版本，Keil 版的工程是基于 Keil MDK-ARM 5.29 uVision5 创建的，IAR 版的工程是基于 IAR Embedded Workbench for ARM 8.32.1 创建的，GD32EBuilder 版是基于 GD32 Embedded Builder_v1.5.5_Rel 创建的。在使用过程中有如下几点需要注意：

- 1、如果使用 Keil uVision5 打开工程，安装（网址：<https://www.gd32mcu.com>）最新版本 GigaDevice.GD32M53x_DFP，以加载相关文件。
- 2、如果使用 IAR 打开工程，安装（网址：<https://www.gd32mcu.com>）最新版本 IAR_GD32M53x_ADDON，以加载相关文件。

4. 硬件设计概述

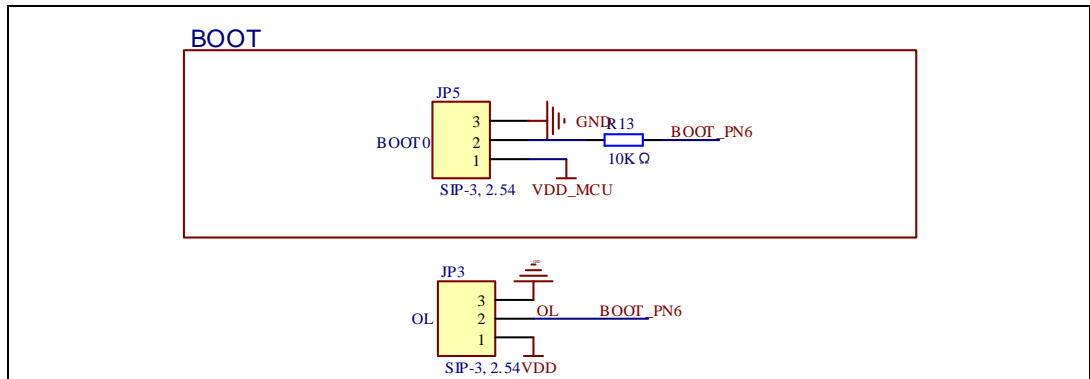
4.1. 供电电源

图4-1. 供电电源原理图



4.2. 启动方式选择

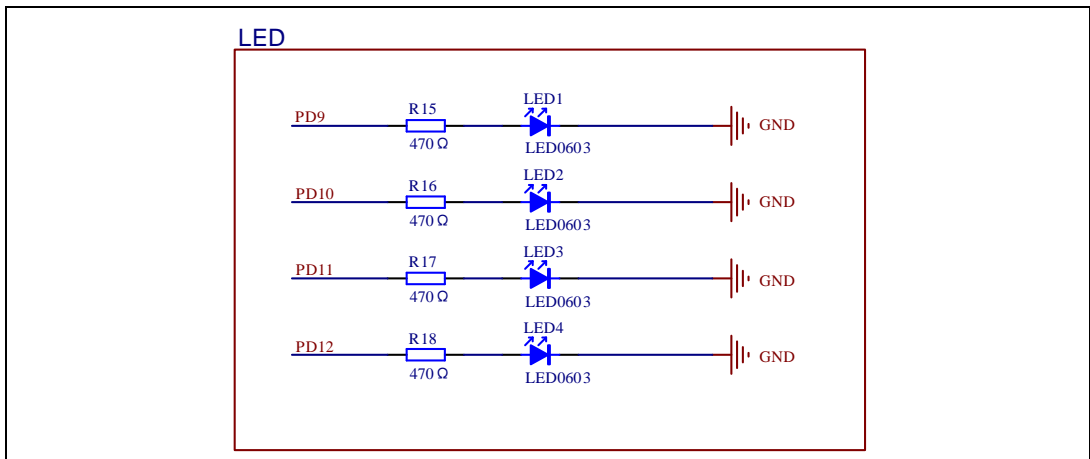
图4-2. 启动方式选择原理图



注意： 由于板载调试器 OL 脚默认为内部上拉，若从系统存储器启动，需要配置 JP3 到 GND；若从主 Flash 启动，需要配置 JP5 到 VDD_MCU 且不要配置 JP3 到 GND。

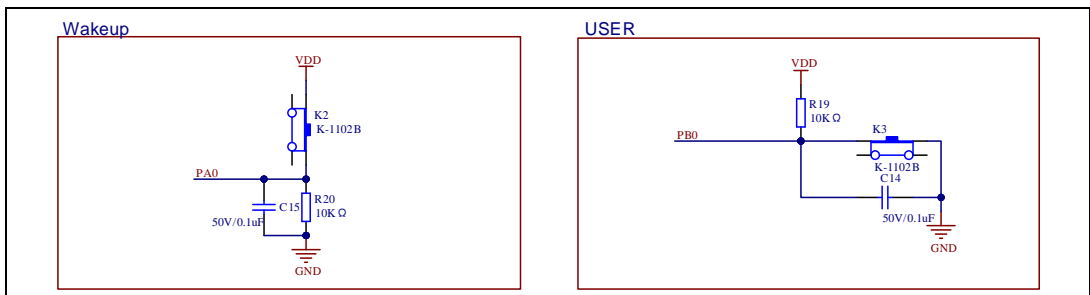
4.3. LED 指示灯

图4-3. LED功能原理图



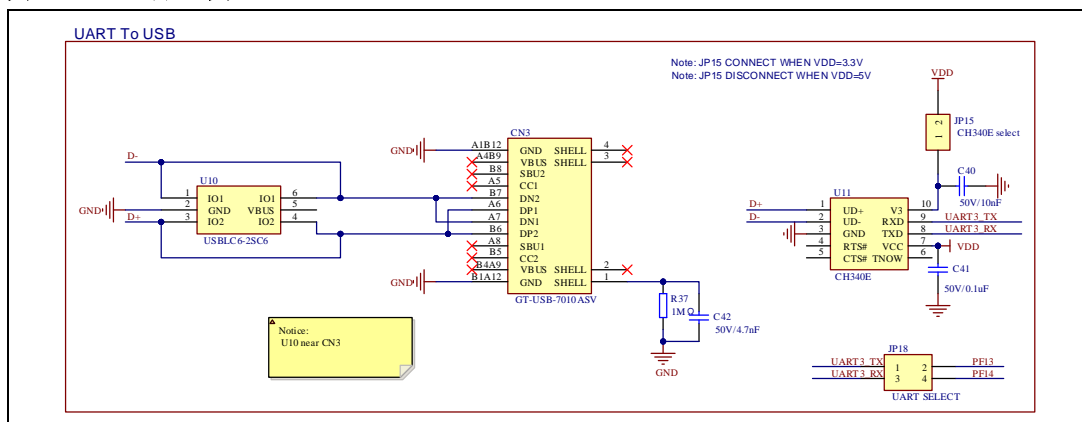
4.4. 按键

图4-4. 按键功能原理图



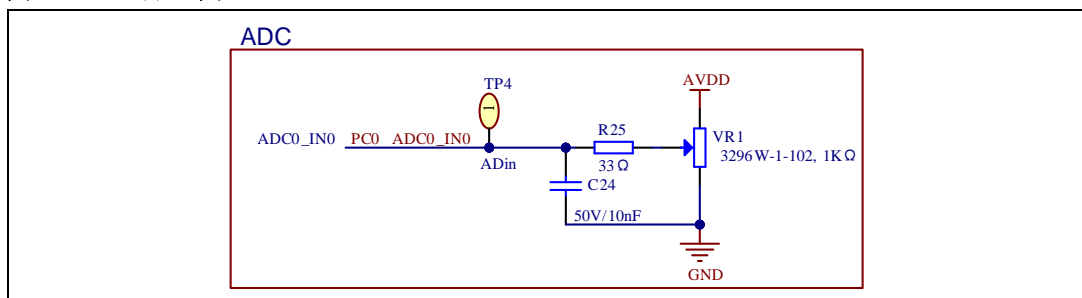
4.5. UART

图4-5.UART原理图



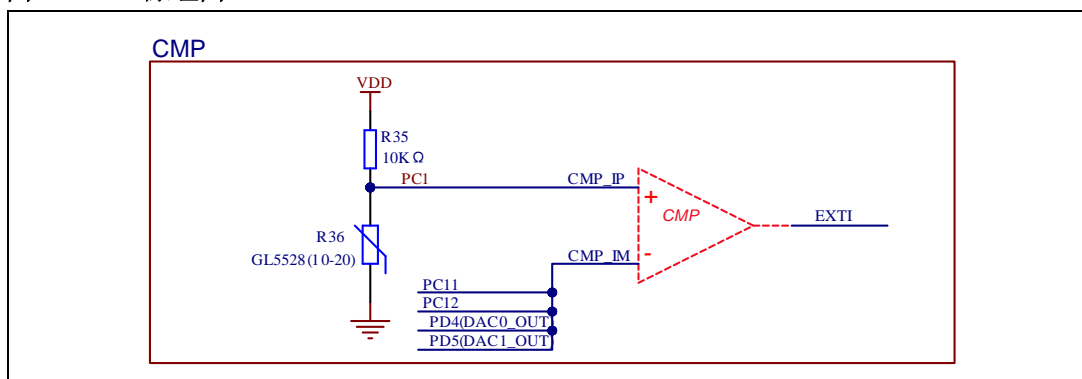
4.6. ADC

图4-6. ADC原理图



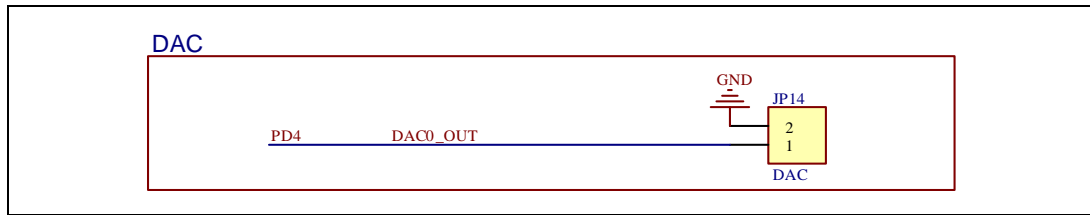
4.7. CMP

图4-7. CMP原理图



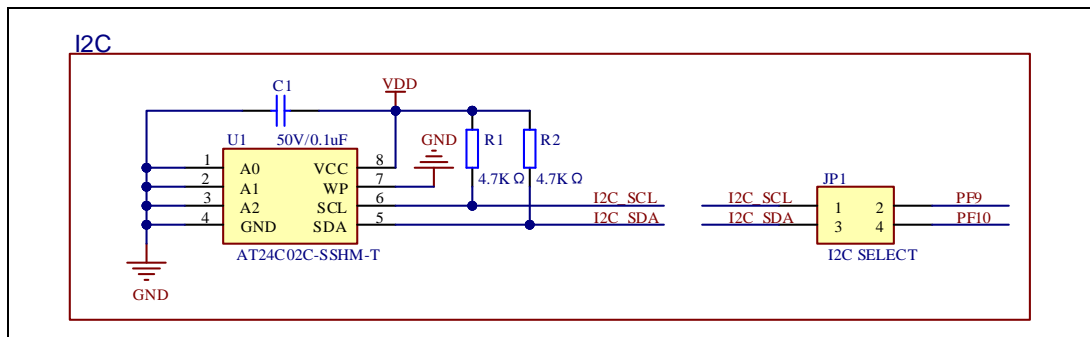
4.8. DAC

图4-8. DAC原理图



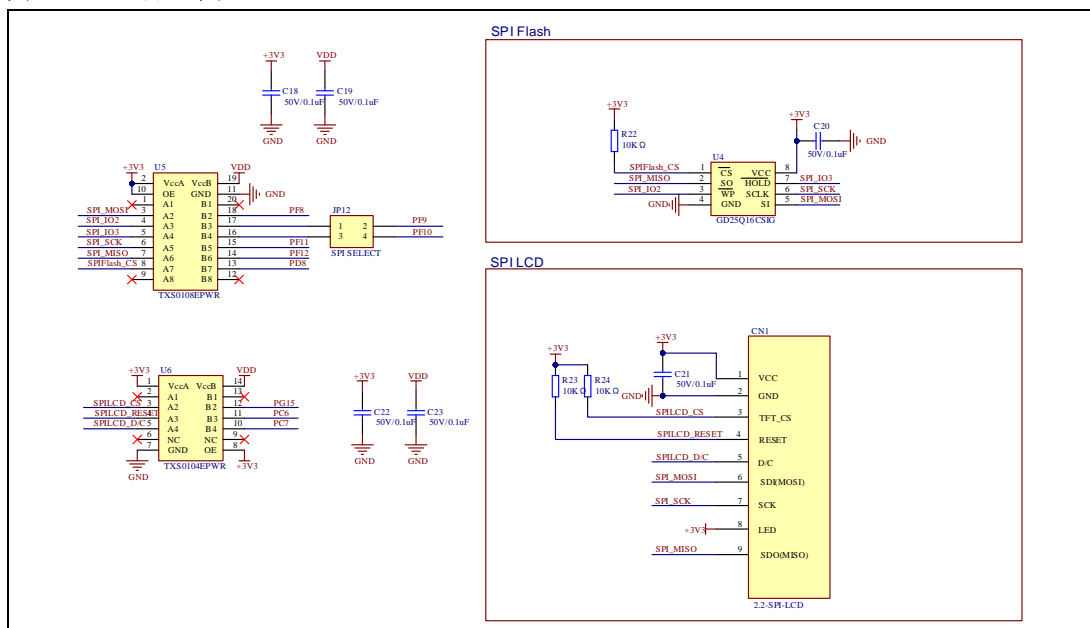
4.9. I2C

图4-9. I2C原理图



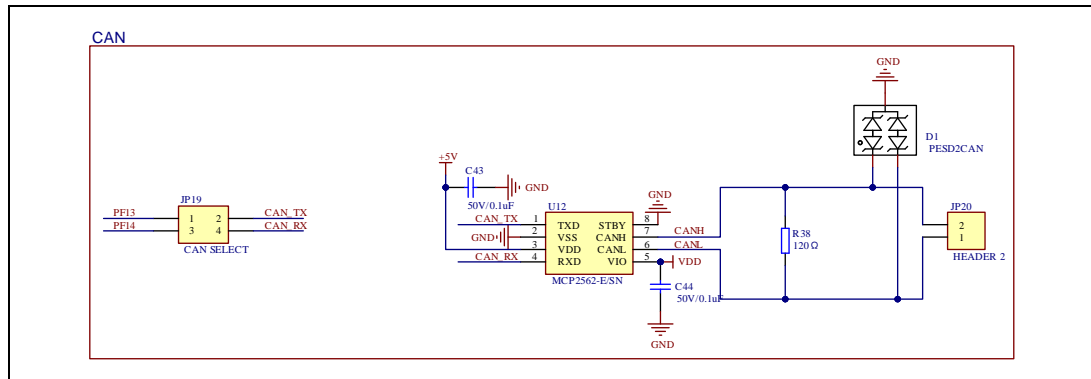
4.10. SPI

图4-10. SPI原理图



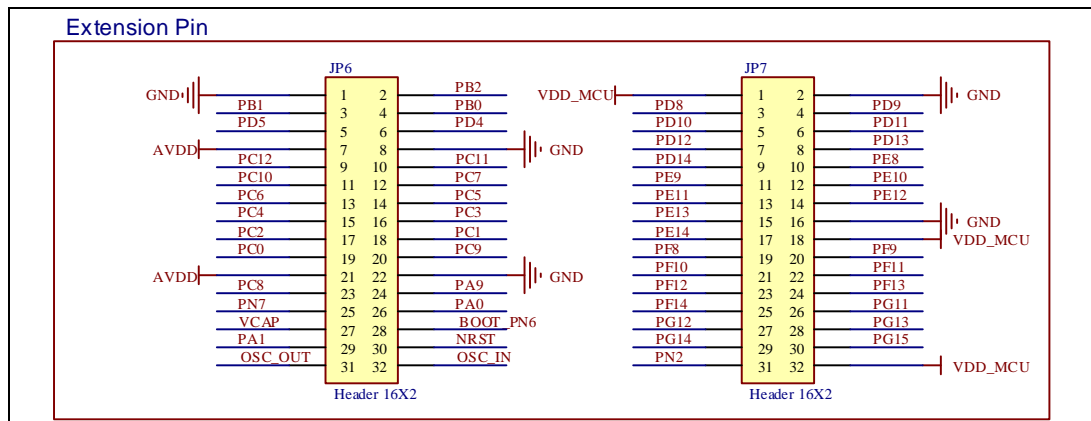
4.11. CAN

图4-11. CAN原理图



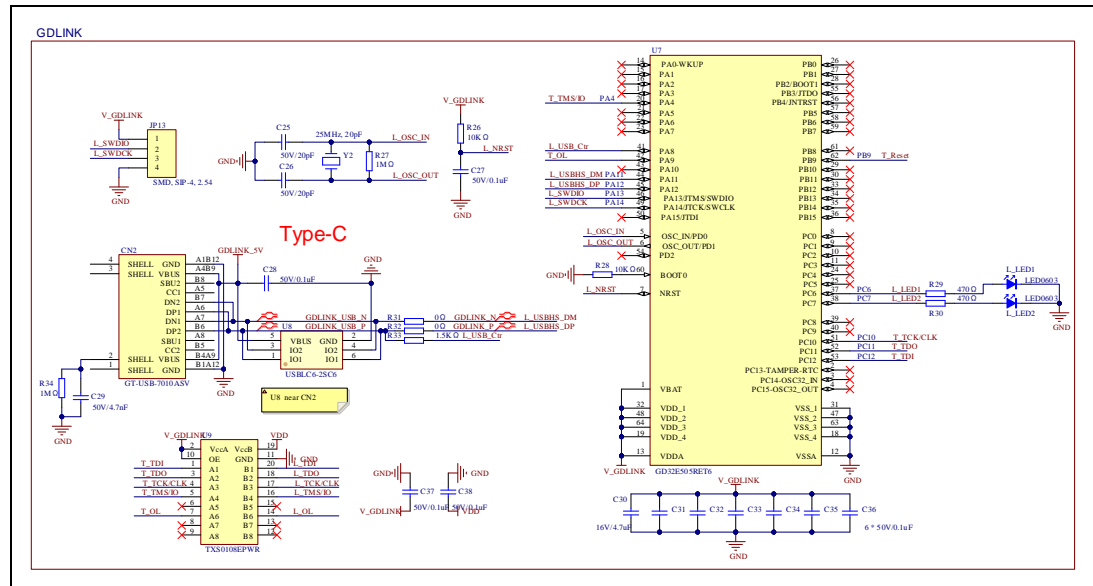
4.12. Extension

图4-12. Extension原理图



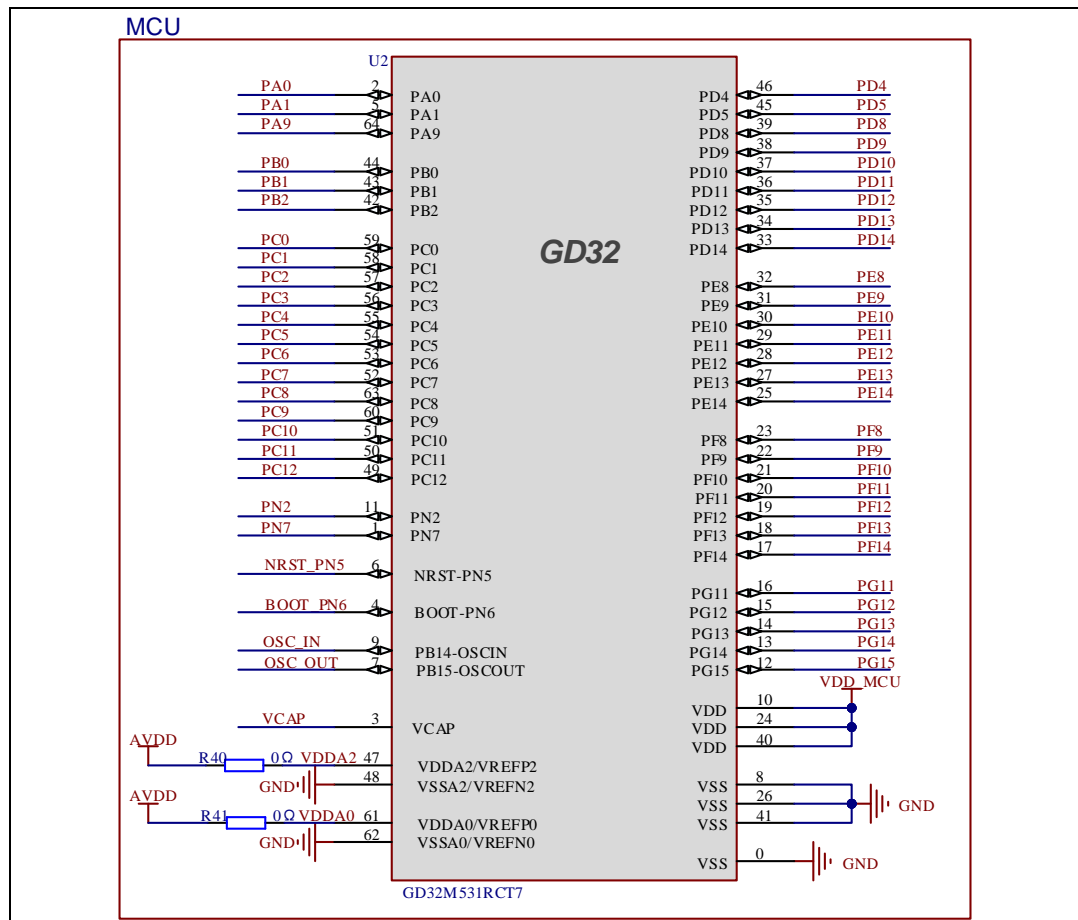
4.13. GD-Link

图4-13. GD-Link原理图



4.14. MCU

图4-14. 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。

GD32M531R-EVAL 开发板上有 4 个 LED。LED1，LED2，LED3，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，LED2，LED3，LED4 将以流水的状态改变，然后循环重复整个过程。

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。

GD32M531R-EVAL 开发板有三个按键和四个 LED。其中，三个按键是 Reset 按键，Wakeup 按键，User 按键；LED1，LED2，LED3 和 LED4 可通过 GPIO 控制。

这个例程讲述如何使用 User 按键控制 LED1。当按下 User 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 LED1 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 User 按键，LED1 将会点亮，再次按下 User 按键，LED1 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

GD32M531R-EVAL 开发板有三个按键和四个 LED。其中，三个按键是 Reset 按键，Wakeup 按键，User 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 LED1。当按下 User 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 LED1 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。启动后，LED1 闪烁一次，按下 User 按键，LED1 将会点亮，再次按下 User 按键，LED1 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 UART

5.4.2. DEMO 执行结果

下载程序<04_UART_Printf>到开发板，确保 JP18 已连接，然后将串口线连接到 UART 接口。首先，所有灯亮灭 2 次用于测试。然后 UART 将输出“UART printf example: please press the User key”到超级终端。按下按键 User 键，串口继续输出“UART printf example”。

超级终端输出的信息如下图所示：

```
UART printf example: please press the User key

UART 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_UART_HyperTerminal_Interrupt>到开发板，确保 JP18 已连接，然后将串口线连接到 UART 接口。首先，所有灯亮灭一次用于测试。然后 UART 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 和 LED4 轮流闪烁；如果结果不相同，LED1，LED2，LED3 和 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_UART_DMA>到开发板，确保 JP18 已连接，然后将串口线连接到 UART 接口。首先，所有灯亮灭一次用于测试。然后 UART 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 和 LED4 轮流闪烁；如果结果不相同，LED1，LED2，LED3 和 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
```

5.7. 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
- 学习转换温度传感器通道和内部参考电压通道

5.7.2. DEMO 执行结果

下载<07_ADC_Temperature_Vrefint>至开发板并运行，确保 JP18 已连接。

将开发板的 COM 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（V_{REFINT}）的值。

当程序运行时，LED 会闪烁。

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 27 degrees Celsius
the reference voltage data is 1.188V

the temperature data is 29 degrees Celsius
the reference voltage data is 1.193V
```

5.8. 定时器触发 ADC 转换

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 转换模拟量为数字量

- 学习使用定时器生成比较事件

5.8.2. DEMO 执行结果

下载< 08_ADC_Conversion_Triggered_By_Timer >至开发板并运行，确保 JP18 已连接。

定时器 0 的 TIMERO0_TRGOF 事件触发 ADC 转换，ADC 转换的结果将随着模拟值输入的改变而改变。通过串口可以观察转换结果。

当程序运行时，LED 会闪烁。

```
the result of ADC0_IN0 is 1608
the result of ADC0_IN0 is 1611
the result of ADC0_IN0 is 1613
the result of ADC0_IN0 is 1613
the result of ADC0_IN0 is 1612
the result of ADC0_IN0 is 1610
the result of ADC0_IN0 is 1611
the result of ADC0_IN0 is 1602
the result of ADC0_IN0 is 1566
the result of ADC0_IN0 is 1544
the result of ADC0_IN0 is 1533
the result of ADC0_IN0 is 1533
the result of ADC0_IN0 is 1535
the result of ADC0_IN0 is 1531
the result of ADC0_IN0 is 1529
the result of ADC0_IN0 is 1534
the result of ADC0_IN0 is 1535
the result of ADC0_IN0 is 1534
the result of ADC0_IN0 is 1535
the result of ADC0_IN0 is 1530
the result of ADC0_IN0 is 1534
```

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（VREF/2）的参考电压下，它的转换值应该为 1.65V，将会在 PD4 引脚输出。

PD4 输出的电压可以通过示波器观测。

5.10. 比较器输出读取指示灯

5.10.1. DEMO 目的

这个例程包括了 GD32 MCU 的以下功能：

- 学习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 PC1，另一个是 DAC 输出电压，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。JP1 和 JP18 需要连接上，且将开发板的 UART 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 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 四线发 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，确保 JP12、JP18 已连接，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。然后比较写入的和读出是否一致，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。

```
#####
GD32M531R-EVAL System is Starting up...
GD32M531R-EVAL Flash:256K
GD32M531R-EVAL The CPU Unique Device ID:[FFFF0048-70FF17-32450CF0]
GD32M531R-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. SPI TFT 显示

5.13.1. 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 屏显示。

GD32M531R-EVAL-V1.2 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。

5.13.2. DEMO 执行结果

下载程序<13_SPI_TFT_LCD_Driver>到开发板上。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。



5.14. 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 UART 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，确保 JP18 已连接。将开发板的 UART 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 按钮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 PF12 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_OUT0: system clock divided by 16  
CK_OUT0: PLL clock / 8 divided by 4  
CK_OUT0: HXTAL clock  
CK_OUT0: IRC32M clock divided by 2  
CK_OUT0: IRC32K clock  
CK_OUT0: system clock divided by 16
```

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，确保 JP18 已连接，并将串口线连到开发板的 UART 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 UART 接收中断唤醒。所有的 LED 灯同时闪烁。

5.16. TIMER 呼吸灯

5.16.1. 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

5.16.2. DEMO 执行结果

下载程序<16_TIMER_Breath_LED>到开发板，并运行程序。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。

5.17. 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

GD32M531R-EVAL 开发板集成了 CAN（控制器局域网）总线控制器，它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 GD32M531R-EVAL 开发板。JP19 必须连接。将两个板子的 JP20 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。

如果需要打印功能，请确保 JP1 和 JP12 上没有连接跳线帽。然后使用杜邦线将 PF10 连接到 JP18 的引脚 1，PF9 连接到 JP18 的引脚 3。同时，将 gd32m53x_eval.h 文件中 EVAL_COM 宏的定义修改为 EVAL_COM1。连接 CN3 至 PC。

例程首先将输出“please press the Tamper key to transmit data!”到超级终端。按下 User 键，数据帧通过 CAN 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 User key to transmit data!

can transmit data: a0 a1 a2 a3 a4 a5 a6 a7
can receive data: a0 a1 a2 a3 a4 a5 a6 a7
```

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	2026 年 2 月 28 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 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.