GigaDevice Semiconductor Inc.

GD32350R-EVAL
User Manual

Table of Contents

目录

Table of Contents *1*

List of Figures *4*

List of Tables *5*

1. Summary *6*

2. Function Pin Assign *6*

3. Getting started *7*

4. Hardware layout overview *7*

4.1. Power supply 7

4.2. Boot option 8

4.3. LED 8

4.4. KEY 8

4.5. USART0 9

4.6. RS485 9

4.7. ADC/DAC 10

4.8. I2S 10

4.9. I2C 11

4.10. SPI-TF CARD 11

4.11. SPI-TFT LCD 12

4.12. USBFS 12

4.13. CMP 13

4.14. HDMI-CEC 13

4.15. TSI 13

4.16. IFRP 14

4.17. RTC 14

4.18. GD-Link 15

4.19. Extension 16

4.20. MCU 17

5. Routine use guide 17

5.1. GPIO_Runing_Led 17
5.1.1. DEMO Purpose 17
5.1.2. DEMO Running Result 17

5.2. GPIO_KeyBoard_Polling_mode 18
5.2.1. DEMO Purpose 18
5.2.2. DEMO Running Result 18

5.3. GPIO_KeyBoard_Interrupt_mode 18
5.3.1. DEMO Purpose 18
5.3.2. DEMO Running Result 18

5.4. USART_Printf 19
5.4.1. DEMO Purpose 19
5.4.2. DEMO Running Result 19

5.5. USART_HyperTerminal_Interrupt 19
5.5.1. DEMO Purpose 19
5.5.2. DEMO Running Result 19

5.6. USART_DMA 20
5.6.1. DEMO Purpose 20
5.6.2. DEMO Running Result 20

5.7. RS485_Test 20
5.7.1. DEMO Purpose 20
5.7.2. DEMO Running Result 21

5.8. ADC_Conversion_Triggered_By_Timer 21
5.8.1. DEMO Purpose 21
5.8.2. DEMO Running Result 22

5.9. DAC_Output_Voltage_Value 22
5.9.1. DEMO Purpose 22
5.9.2. DEMO Running Result 22

5.10. Comparator_Obtain_Brightness 22
5.10.1. DEMO Purpose 22
5.10.2. DEMO Running Result 22

5.11. I2C_EEPROM 23
5.11.1. DEMO Purpose 23
5.11.2. DEMO Running Result 23

5.12. SPI_TF_Card_Block_Operation 24
5.12.1. DEMO Purpose 24
5.12.2. DEMO Running Result 24

5.13. SPI_TF_Card_FATFS_Operation 25
5.13.1. DEMO Purpose 25

5.13.2. DEMO Running Result 25

5.14. SPI_TFT_LCD_Driver 26

5.14.1. DEMO Purpose 26

5.14.2. DEMO Running Result 26

5.15. HDMI-CEC_HostSlaveCommunication 27

5.15.1. DEMO Purpose 27

5.15.2. DEMO Running Result 27

5.16. I2S_Audio_Player 28

5.16.1. DEMO Purpose 28

5.16.2. DEMO Running Result 28

5.17. RCU_Clock_Out 28

5.17.1. DEMO Purpose 28

5.17.2. DEMO Running Result 28

5.18. PMU_sleep_wakeup 29

5.18.1. DEMO Purpose 29

5.18.2. DEMO Running Result 29

5.19. RTC_Calendar 29

5.19.1. DEMO Purpose 29

5.19.2. DEMO Running Result 29

5.20. IRInfrared_Transceiver 29

5.20.1. DEMO Purpose 29

5.20.2. DEMO Running Result 30

5.21. TIMER_Breath_LED 30

5.21.1. DEMO Purpose 30

5.21.2. DEMO Running Result 30

5.22. TSI_TouchKey_leds 30

5.22.1. DEMO Purpose 30

5.22.2. DEMO Running Result 30

5.23. USB_Device 30

5.23.1. USBD_CDC_ACM 30

5.23.2. USBD_Keyboard 31

5.24. USB_Host 32

5.24.1. USBH_ HID_Host 32

5.24.2. USBH_MSC_Host 33

6. Revision history 33

List of Figures

Figure 4-1 Schematic diagram of power supply 7
Figure 4-2 Schematic diagram of boot option 8
Figure 4-3 Schematic diagram of LED function 8
Figure 4-4 Schematic diagram of Key function 8
Figure 4-5 Schematic diagram of USART0 function 9
Figure 4-6 Schematic diagram of RS485 function 9
Figure 4-7 Schematic diagram of ADC/DAC function 10
Figure 4-8 Schematic diagram of I2S function 10
Figure 4-9 Schematic diagram of I2C function 11
Figure 4-10 Schematic diagram of SPI-TF CARD function 11
Figure 4-11 Schematic diagram of SPI-TFT LCD function 12
Figure 4-12 Schematic diagram of USBFS function 12
Figure 4-13 Schematic diagram of CMP function 13
Figure 4-14 Schematic diagram of HDMI-CEC function 13
Figure 4-15 Schematic diagram of TSI function 13
Figure 4-16 Schematic diagram of IFRP function 14
Figure 4-17 Schematic diagram of RTC function 14
Figure 4-18 Schematic diagram of GD-Link function 15
Figure 4-19 Schematic diagram of Extension Pin 16
Figure 4-20 Schematic diagram of MCU Pin 17

List of Tables

Table 2-1 Pin assignment *6*
Table 4-1 Boot configuration *8*
Table 6-1 Revision history *33*

1. Summary

GD32350R-EVAL evaluation board uses GD32F350RBT6 as the main controller. As a complete development platform of GD32F3x0 powered by ARM® Cortex™-M4 core, the board supports full range of peripherals. It uses mini-USB interface or AC/DC adapter to supply 5V power. SWD, Reset, Boot, User button key, LED, I2C, I2S, USART, RS485, TFT-LCD, HDMI-CEC, LDR, TSI, IFRP LED, IR Receiver, RTC, SPI, USB, ADC, DAC and Extension Pin are also included. This document details its hardware schematic and the relevant applications.

2. Function Pin Assign

Table 2-1 Pin assignment

Function	Pin	Description
LED	PC10	LED1
	PC11	LED2
	PC12	LED3
	PD2	LED4
RESET		K1-Reset
KEY	PA0	K2-Wakeup
	PC13	K4-Temper
	PF7	K3-User Key
USB	PA11	USBDM
	PA12	USBDP
	PA9	USB VBUS
IR	PC6	IR_RX
	PB9	IR_TX
I2C	PB6	I2C0_SCL
	PB7	I2C0_SDA
I2S	PA4	I2S_WS
	PA5	I2S_CK
	PA7	I2S_DIN
	PA15	MSEL
	PB3	MCLK
	PB5	MDIN
	PA6	I2S_MCK
USART0	PA9	USART0_TX
	PA10	USART0_RX
RS485	PA2	RS485_TX
	PA3	RS485_RX
	PA1	RS485_DIR
TSI	PB11	TSI_G5_IO0

	PB14	TSI_G5_IO3
	PB13	TSI_G2_IO2
	PB12	TSI_G5_IO1
	PC5	TSI_G2_IO0
	PB0	TSI_G2_IO1
SPI	PB3	SPI0_SCK
	PB4	SPI0_MISO
	PB5	SPI0_MOSI
	PF5	TFT_CS
	PF4	TF_CARD_CS
	PC4	TFT_RESET
ADC	PC1	ADC_IN11
HDMI-CEC	PB8	CEC
COMPARATOR	PA1	COMP0_INP
DAC	PA4	DAC_OUT

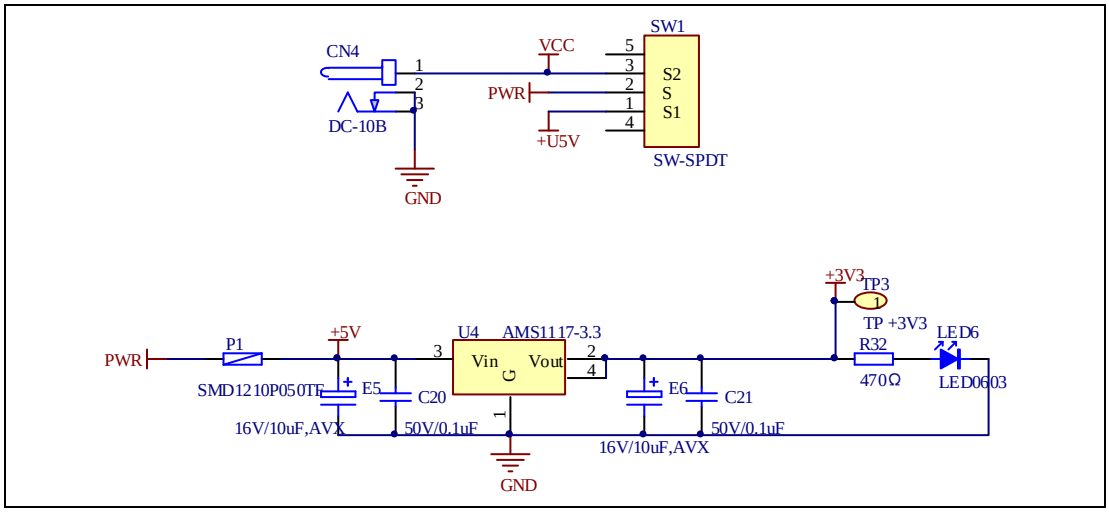
3. Getting started

The EVAL Board uses mini-USB connecter or AC/DC adapter to get power, the hardware system power is +3.3V. A mini-USB cable and a J-Link tool are necessary to down programs. Select the correct boot mode and then power on, the LED6 will turn on, which indicates the power supply is ready.

4. Hardware layout overview

4.1. Power supply

Figure 4-1 Schematic diagram of power supply



4.2. Boot option

Figure 4-2 Schematic diagram of boot option

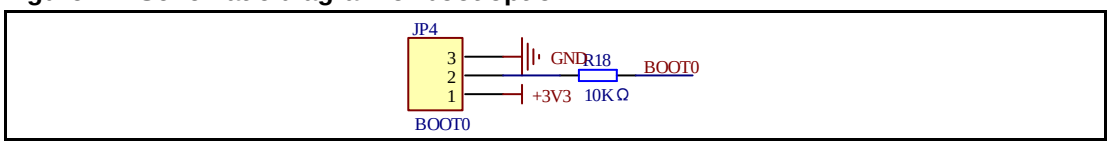
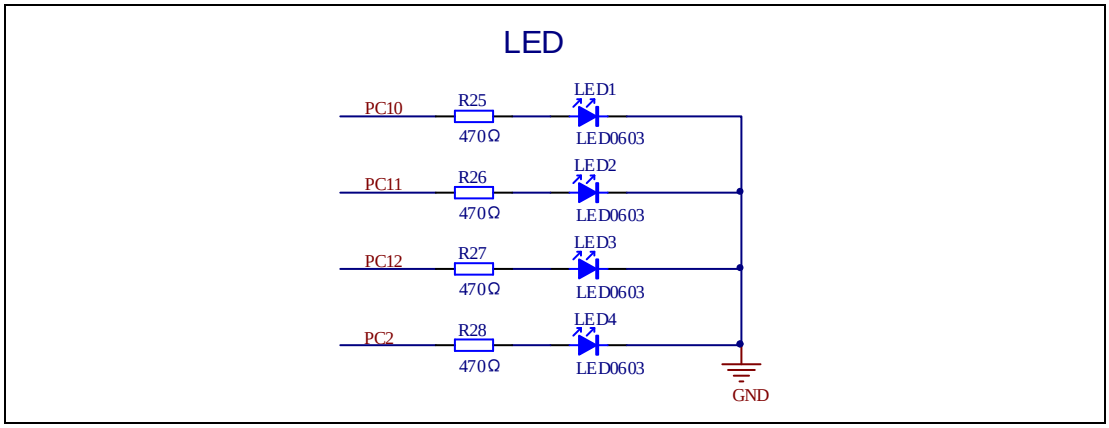


Table 4-1 Boot configuration

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

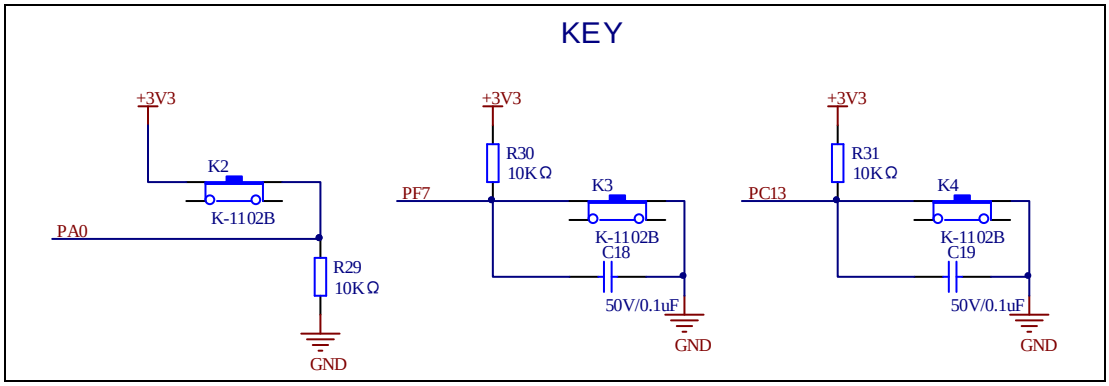
4.3. LED

Figure 4-3 Schematic diagram of LED function



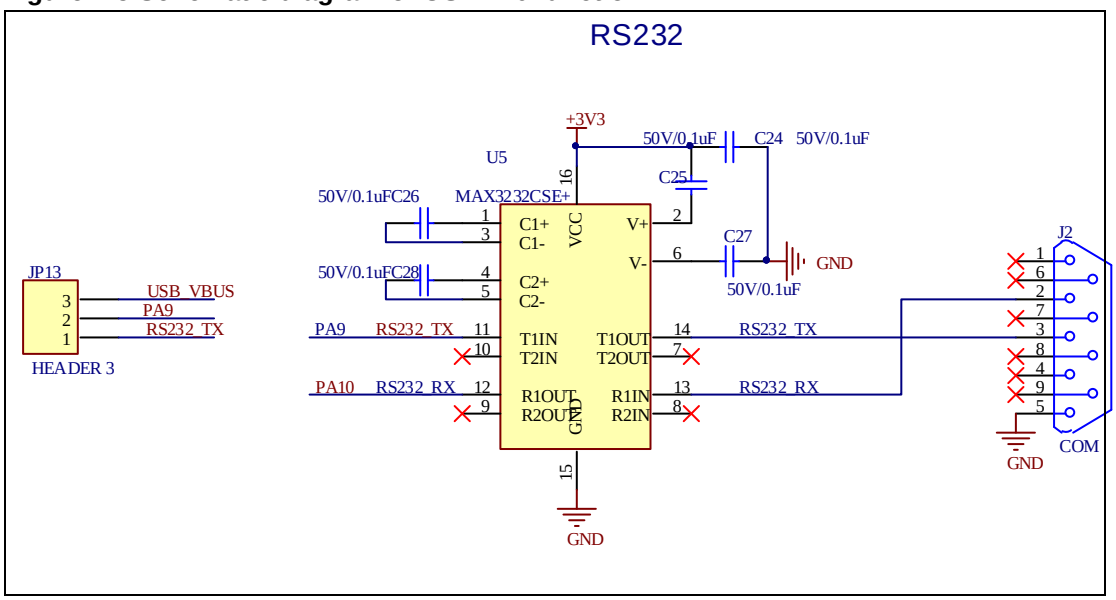
4.4. KEY

Figure 4-4 Schematic diagram of Key function



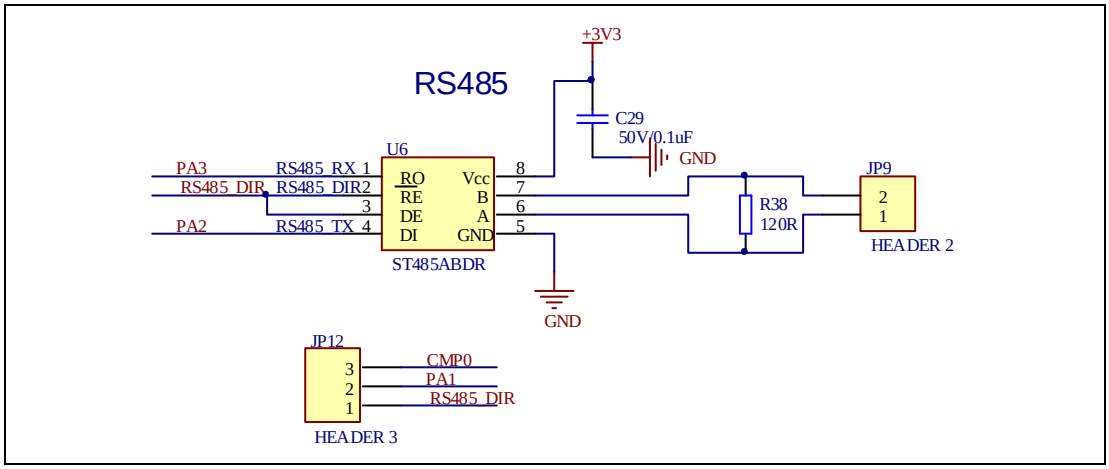
4.5. USART0

Figure 4-5 Schematic diagram of USART0 function



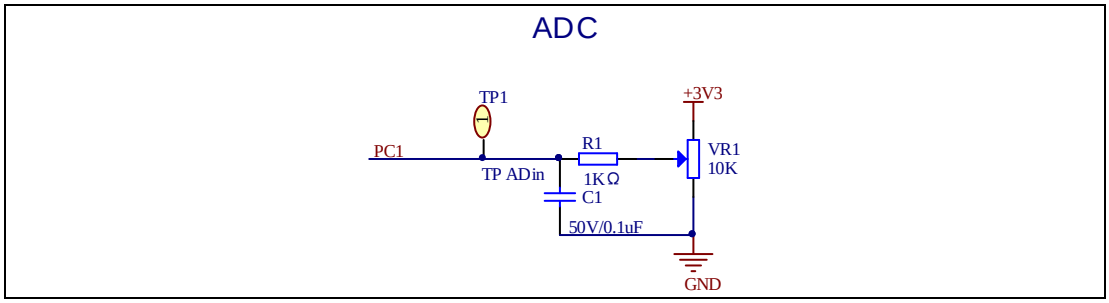
4.6. RS485

Figure 4-6 Schematic diagram of RS485 function



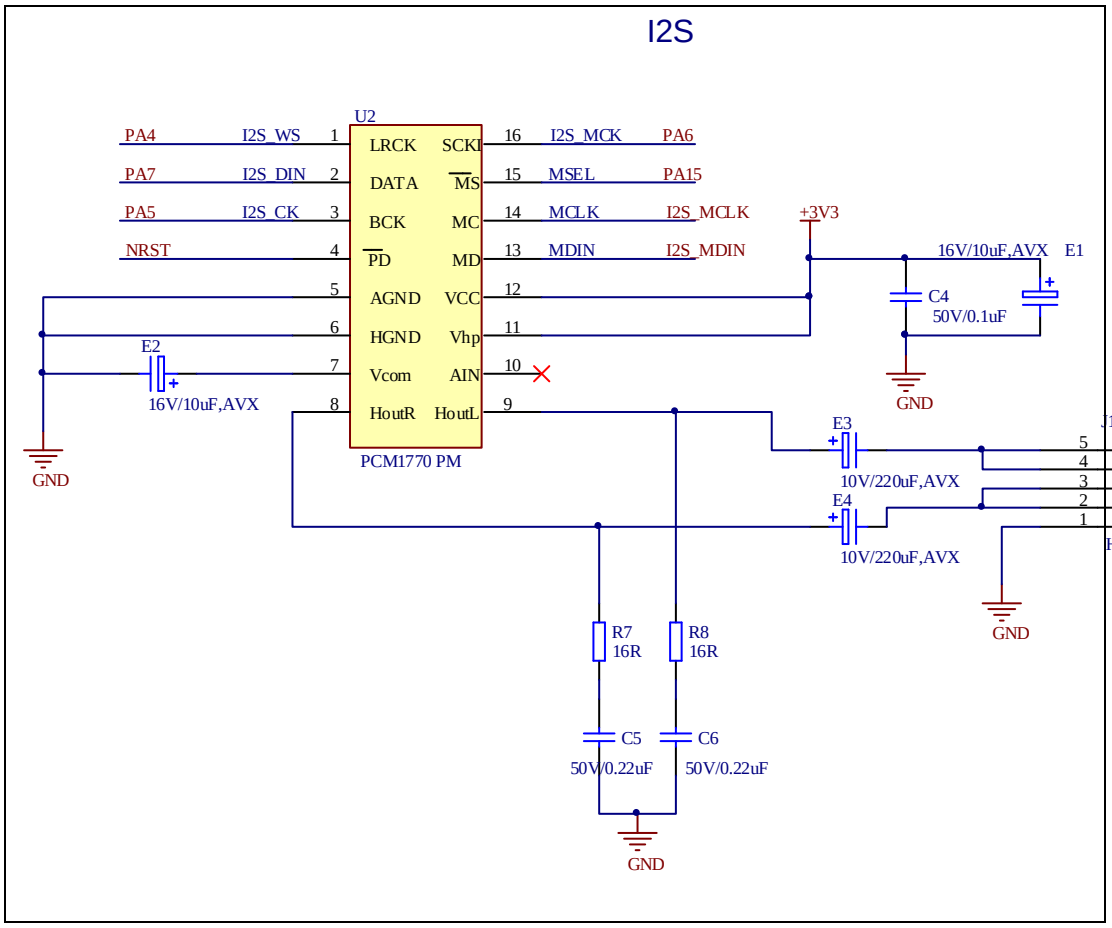
4.7. ADC/DAC

Figure 4-7 Schematic diagram of ADC/DAC function



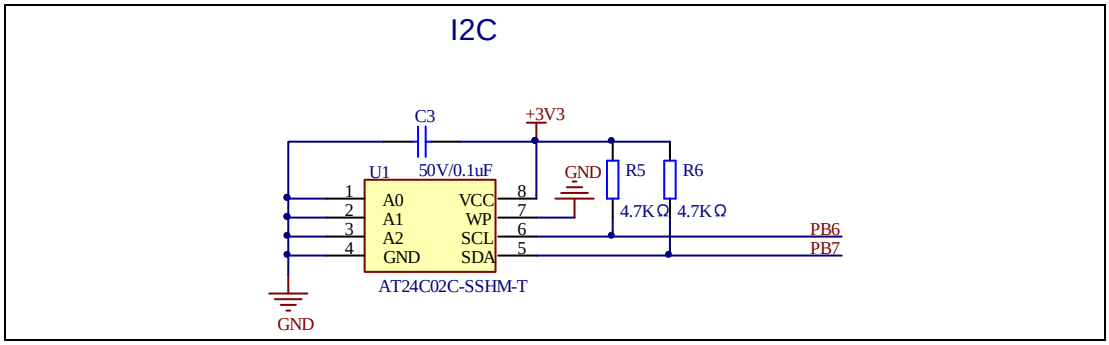
4.8. I2S

Figure 4-8 Schematic diagram of I2S function



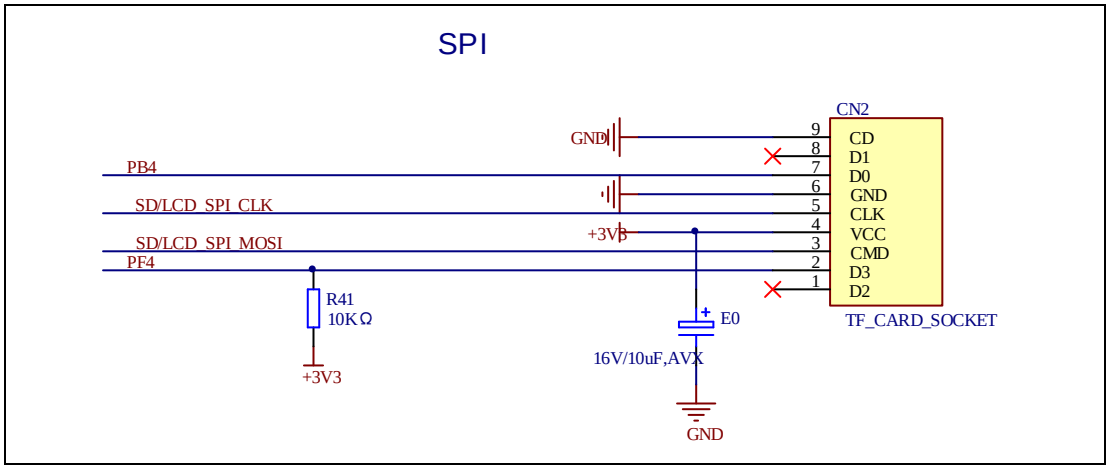
4.9. I2C

Figure 4-9 Schematic diagram of I2C function



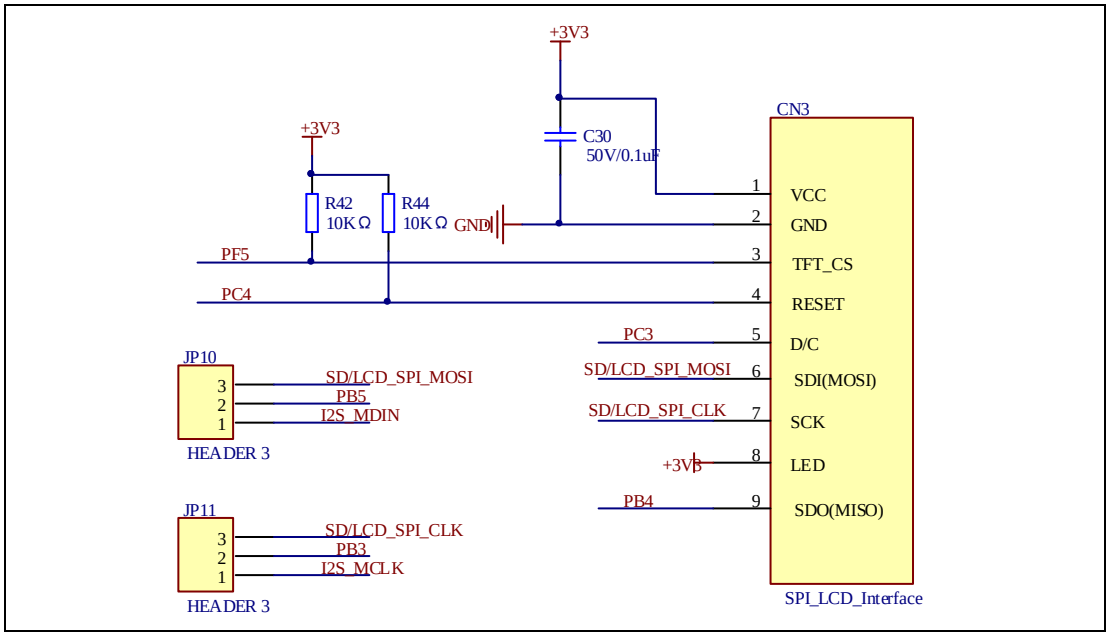
4.10. SPI-TF CARD

Figure 4-10 Schematic diagram of SPI-TF CARD function



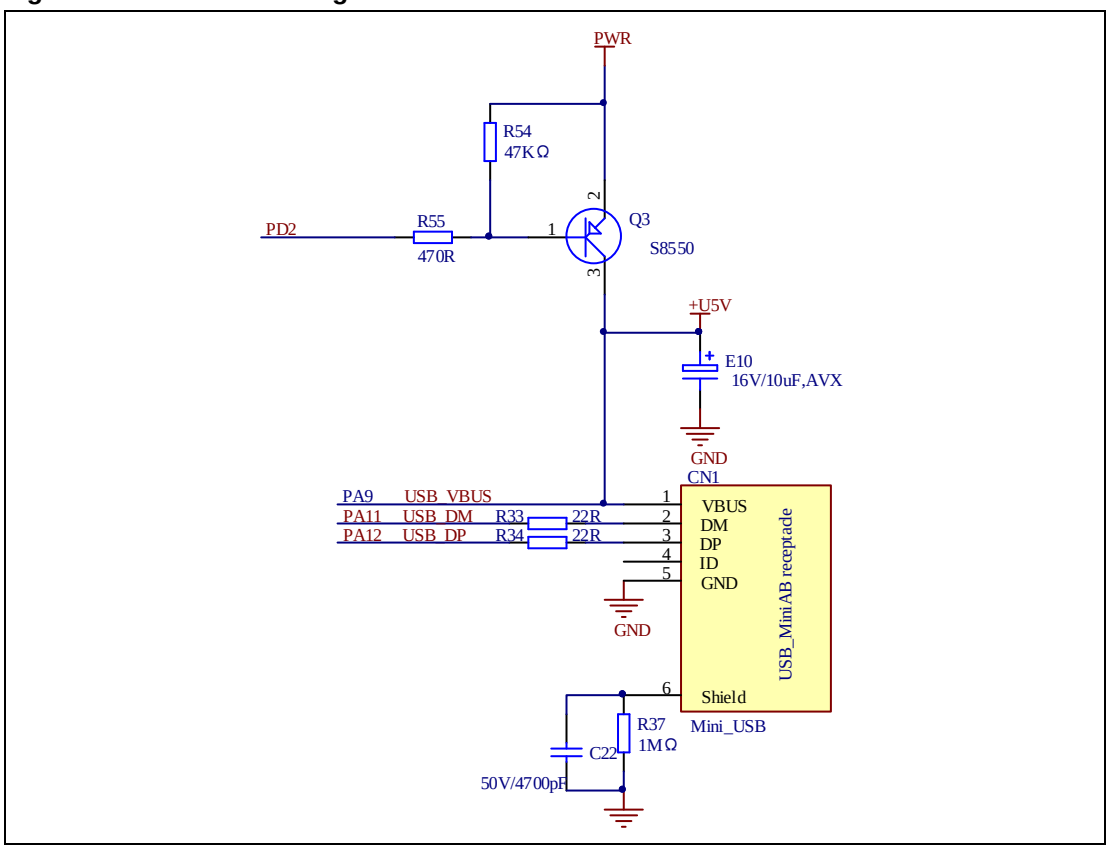
4.11. SPI-TFT LCD

Figure 4-11 Schematic diagram of SPI-TFT LCD function



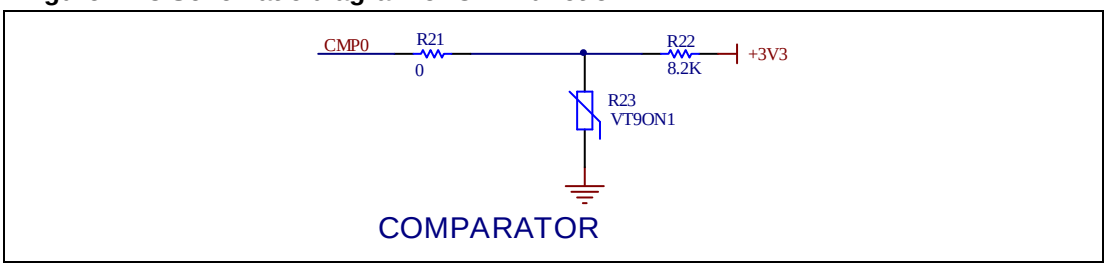
4.12. USBFS

Figure 4-12 Schematic diagram of USBFS function



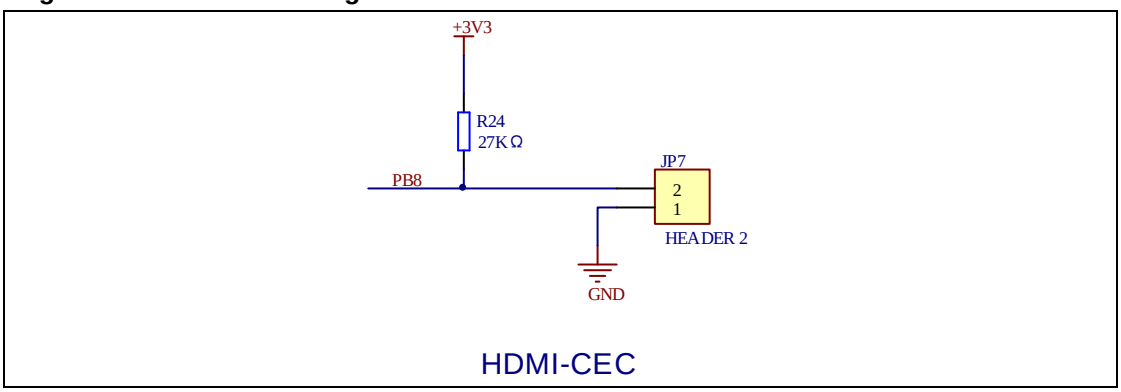
4.13. CMP

Figure 4-13 Schematic diagram of CMP function



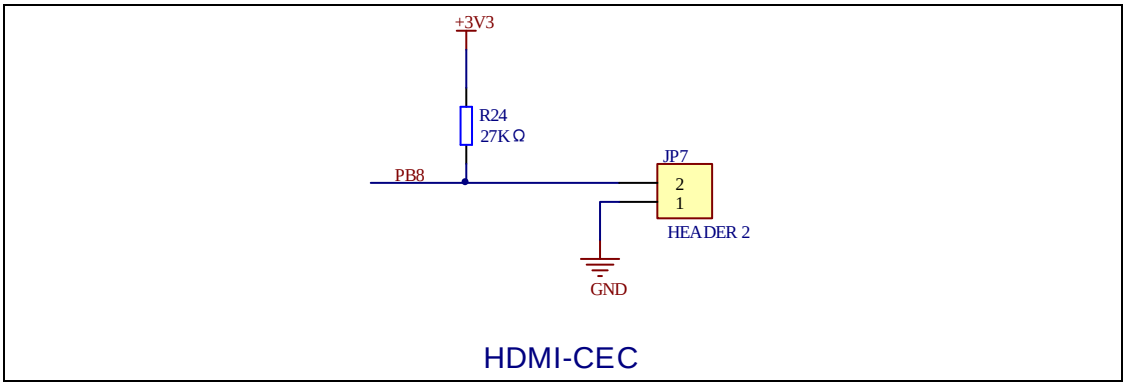
4.14. HDMI-CEC

Figure 4-14 Schematic diagram of HDMI-CEC function



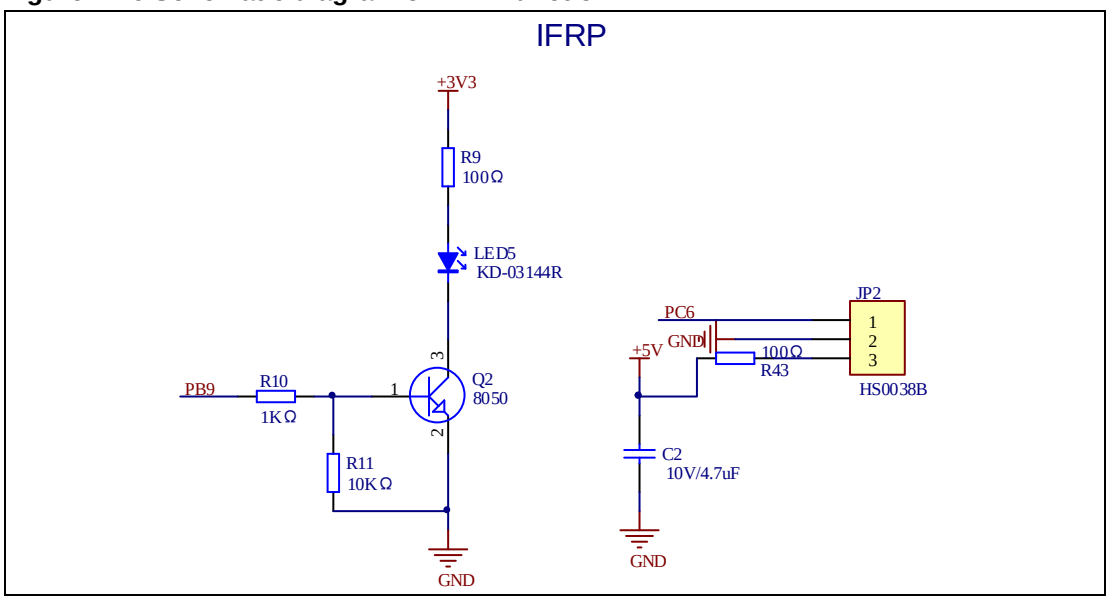
4.15. TSI

Figure 4-15 Schematic diagram of TSI function



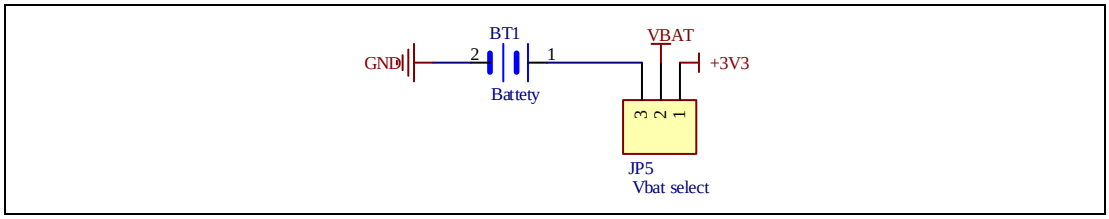
4.16. IFRP

Figure 4-16 Schematic diagram of IFRP function



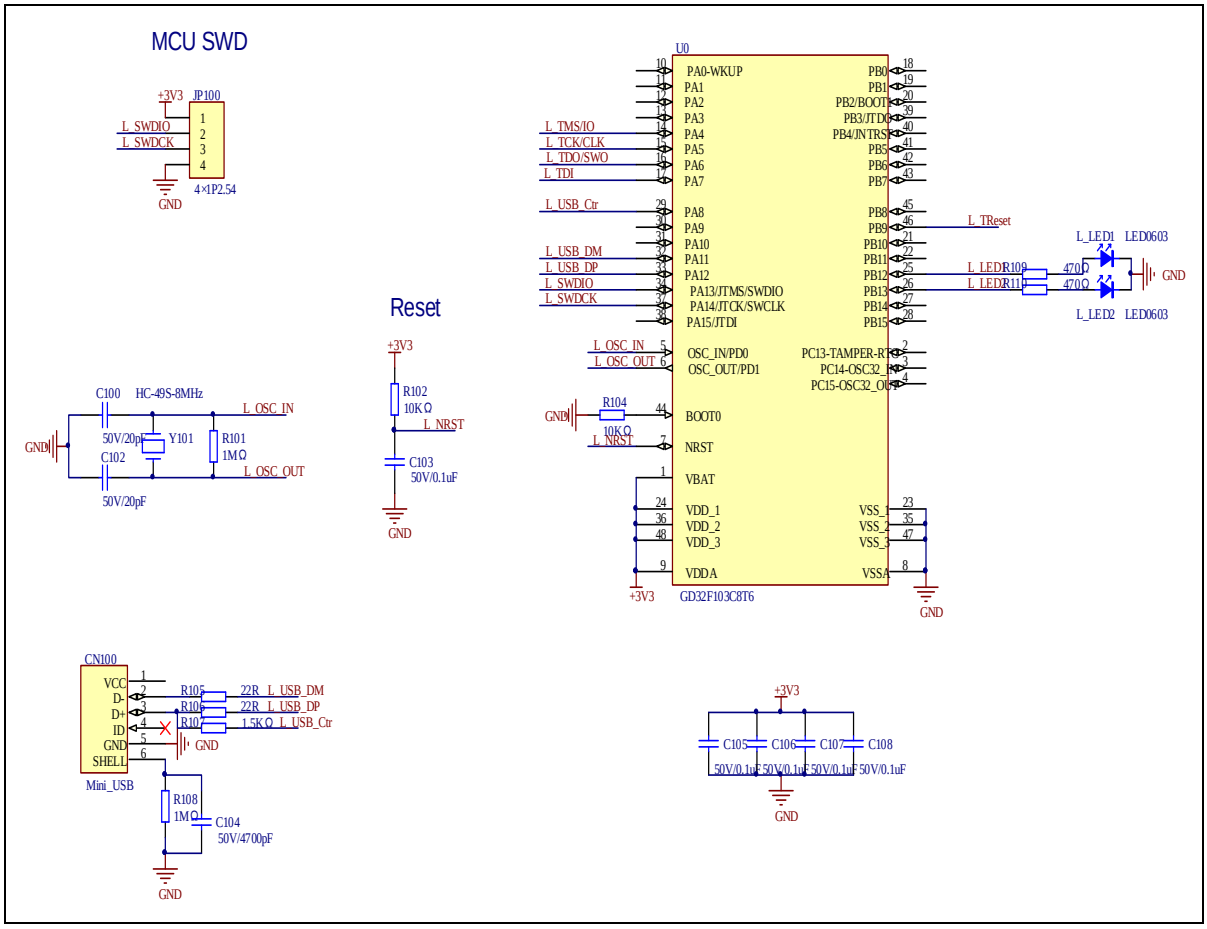
4.17. RTC

Figure 4-17 Schematic diagram of RTC function



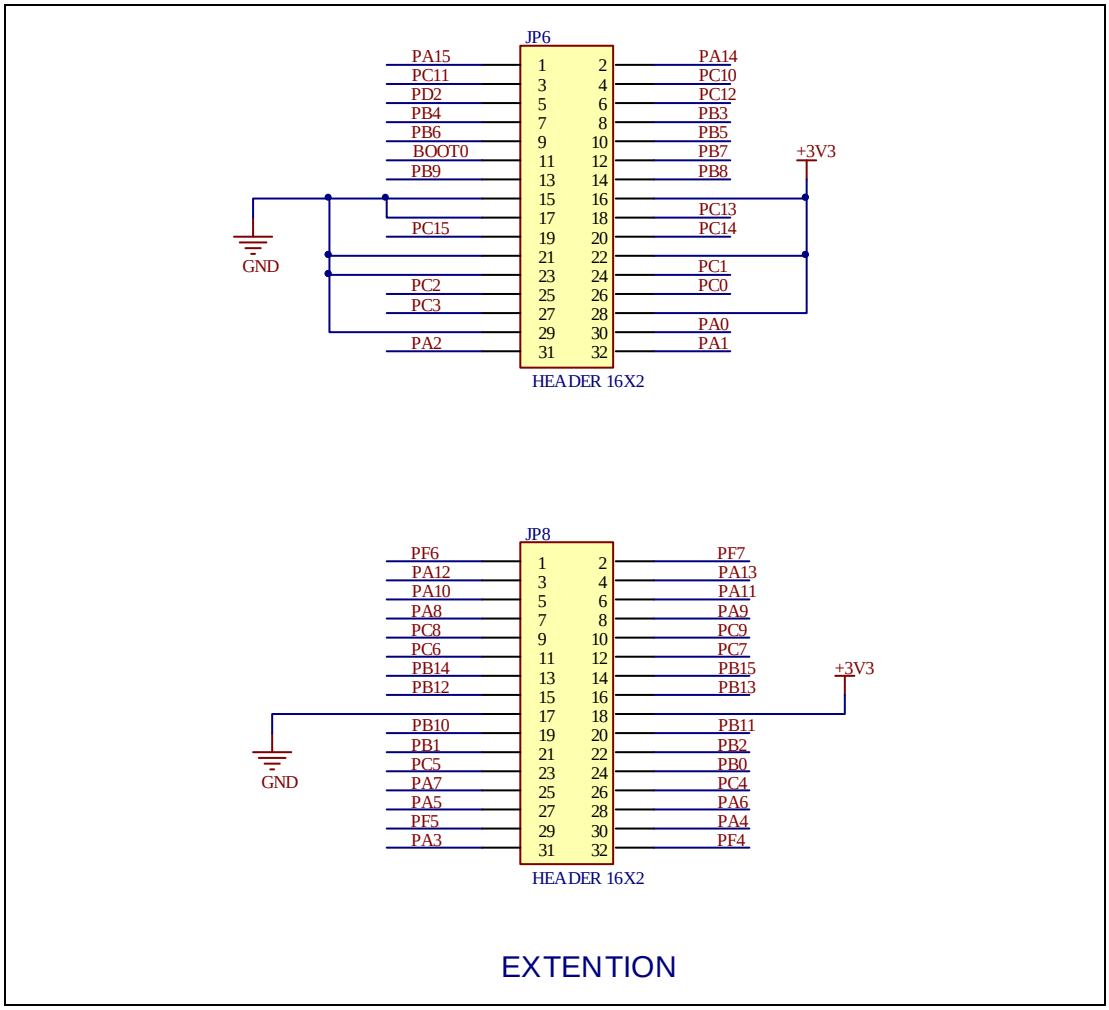
4.18. GD-Link

Figure 4-18 Schematic diagram of GD-Link function



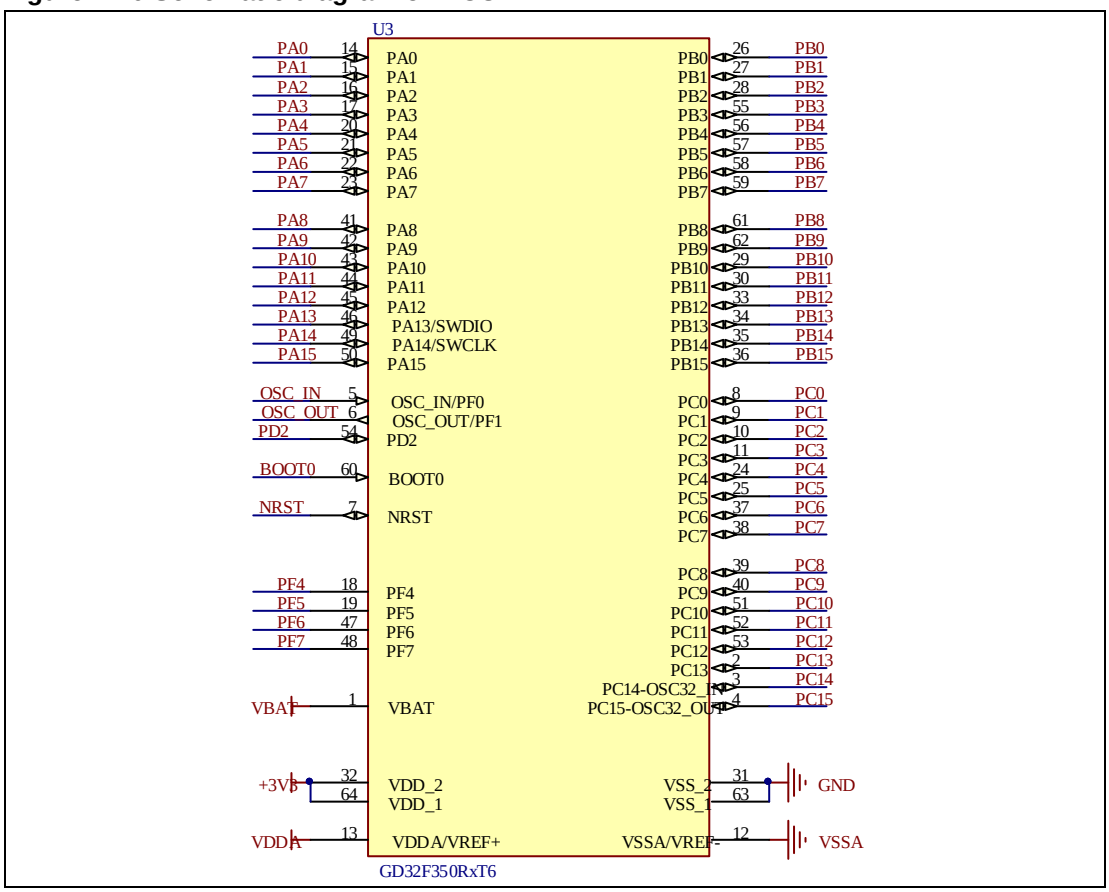
4.19. Extension

Figure 4-19 Schematic diagram of Extension Pin



4.20. MCU

Figure 4-20 Schematic diagram of MCU Pin



5. Routine use guide

5.1. GPIO_Runing_Led

5.1.1. DEMO Purpose

This Demo includes the following functions of GD32 MCU:

- Learn to use GPIO for controlling the LED
- Learn to use SysTick to generate 1ms delay

GD32350R-EVAL board has four LEDs. The LED1, LED2, LED3 and LED4 are controlled by GPIO. This demo will show how to light the LEDs.

5.1.2. DEMO Running Result

Download the program <01_GPIO_Runing_Led> to the EVAL board, four LEDs will turn on one by one from LED1 to LED4 every 200ms, and then turn off together. 200ms later,

the four LEDs work like previous again.

5.2. GPIO_KeyBoard_Polling_mode

5.2.1. DEMO Purpose

This Demo includes the following functions of GD32 MCU:

- Learn to use GPIO control the LED and the KEY
- Learn to use SysTick to generate 1ms delay

GD32350R-EVAL board has four keys and four LEDs. The four keys are Reset key, Tamper key, User key and Wakeup key. The LED1, LED2, LED3 and LED4 are controlled by GPIO.

This demo will show how to use the Tamper key to control the LED2. When press down the User Key, it will check the input value of the IO port. If the value is 0, wait for 50ms. Then check the input value of the IO port again. If the value is still 0, indicates that the button is pressed down successfully, and light the four LED2.

5.2.2. DEMO Running Result

Download the program <02_GPIO_KeyBoard_Polling_mode> to the EVAL board, first of all, all the LEDs will be flashed once for test. Then press down the Tamper Key, LED2 will be turned on. Press down the Tamper Key again, LED2 will be turned off.

5.3. GPIO_KeyBoard_Interrupt_mode

5.3.1. DEMO Purpose

This Demo includes the following functions of GD32 MCU:

- Learn to use GPIO to control the LED and the KEY
- Learn to use EXTI to generate external interrupt

GD32350R-EVAL board has four keys and four LEDs. The four keys are Reset key, Wakeup key, User key and Tamper key. The LED1, LED2, LED3 and LED4 are controlled by GPIO.

This demo will show how to use EXTI interrupt line to control the LED2. When press down the Tamper Key, it will produce an interrupt. In the interrupt service function, the demo will toggle LED2.

5.3.2. DEMO Running Result

Download the program <03_GPIO_KeyBoard_Interrupt_mode> to the EVAL board, first of all, all the LEDs will be flashed once for test. Then press down the Tamper Key, LED2

will be turned on. Press down the Tamper Key again, LED2 will be turned off.

5.4. USART_Printf

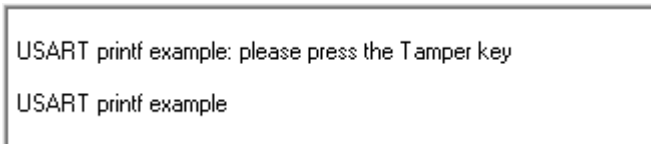
5.4.1. DEMO Purpose

This Demo includes the following functions of GD32 MCU:

- Learn to use GPIO: the Tamper key control the LED
- Learn to retarget the C library printf function to the USART

5.4.2. DEMO Running Result

Jump JP13 to USART, then download the program <04_USART_Printf> to the EVAL board and run. This implementation outputs “USART printf example: please press the Tamper Key” on the hyperterminal using COM. Press the Tamper key, serial port will output “USART Printf Example”. The information via a serial port output as following.



5.5. USART_HyperTerminal_Interrupt

5.5.1. DEMO Purpose

This Demo includes the following functions of GD32 MCU:

- Learn to use the EVAL_COM transmit and receive interrupts to communicate with the hyperterminal

5.5.2. DEMO Running Result

Jump JP13 to USART, then download the program <05_USART_HyperTerminal_ Interrupt> to the EVAL board and run. Firstly, all the LEDs are turned on and off for test. Then, the COM sends the tx_buffer array (from 0x00 to 0xFF) to the hyperterminal and waits for receiving data from the hyperterminal that you must send. The string that you have sent is stored in the rx_buffer array. The receive buffer have a BUFFER_SIZE bytes as maximum. After that, compare tx_buffer with rx_buffer. If tx_buffer is same with rx_buffer, LED1 and LED2 are turned on, LED3 and LED4 are turned off. Otherwise, LED1 and LED2 are turned off, LED3 and LED4 are turned on.
The information via a serial port output as following:

```
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. USART_DMA

5.6.1. DEMO Purpose

This Demo includes the following functions of GD32 MCU:

- Learn to use the COM transmit and receive using DMA

5.6.2. DEMO Running Result

Jump JP13 to USART, then download the program < 06_USART_DMA > to the EVAL board and run. Firstly, all the LEDs are turned on and off for test. Then, the COM sends the tx_buffer array to the hyperterminal and waits for receiving data from the hyperterminal that you must send. The string that you have sent is stored in the rx_buffer array. The receive buffer have a BUFFER_SIZE bytes as maximum. After that, compare tx_buffer with rx_buffer. If tx_buffer is same with rx_buffer, LED1 and LED2 are turned on, LED3 and LED4 are turned off. Otherwise, LED1 and LED2 are turned off, LED3 and LED4 are turned on.

The information via a serial port output as following:

```
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. RS485_Test

5.7.1. DEMO Purpose

This Demo includes the following functions of GD32 MCU:

- Learn to use the USART RS485

5.7.2. DEMO Running Result

Jump the JP12 to RS485 and JP13 to USART with the jumper cap. This routines need to prepare two GD32F350R_EVAL boards, one board as a sender, the other as a receiver. First connect two GD32350R-EVAL boards through RS485 line A and B, and then download the program < 07_RS485_Test > to the board for running. When press the Wakeup key on one board, the board is set as RS485 transmitter and LED2 is on and when press the Tamper key on one board, the board is set as RS485 receiver and LED3 is on.

Download the program < 07_RS485_Test > to the EVAL board and run, the information via a serial port output as following:

```
------------------------------------------------------------------

  GD32F350R_EVAL RS485_Test

--> Press down KEY_WAKEUP to set GD32F350R_EVAL as RS485 transmitter

--> Press down KEY_TAMPER to set GD32F350R_EVAL as RS485 receiver

------------------------------------------------------------------
```

According to the tips, press down Wakeup key to set one board as a transmitter and press down Tamper key to the other as a receiver. The transmitter output as following.

```
RS485 transmitter is enabled

Data is being transmitted: GD32MCU

Data is being transmitted: GD32MCU

Data is being transmitted: GD32MCU

Data is being transmitted: GD32MCU
```

The receiver output as following.

```
RS485 receiver is enabled

Waiting for received data

The received data: [GD32MCU]

The received data: [GD32MCU]

The received data: [GD32MCU]

The received data: [GD32MCU]
```

5.8. ADC_Conversion_Triggered_By_Timer

5.8.1. DEMO Purpose

This Demo includes the following functions of GD32 MCU:

- Learn to use ADC to convert analog to digital
- Learn to use TIMER to generate a CC event

- Learn to use LCD to show the ADC converted result

TIMER1 CH1 event triggers ADC conversion, the value displayed on the LCD corresponds to the ADC analog input, and changes with it. The converted data are moved to SRAM through DMA continuously.

5.8.2. DEMO Running Result

Download the program <08_ADC_conversion_triggered_by_timer> to the GD32350R-EVAL board, adjust the adjustable potentiometer knob to change the analog input. The ADC, which is triggered by TIMER1 CH1 event, will convert the analog input, and you will see the result, a voltage curve, on the LCD. The curve adjusts with the analog input.

5.9. DAC_Output_Voltage_Value

5.9.1. DEMO Purpose

This Demo includes the following functions of GD32 MCU:

- Learn to use DAC channel to generate different voltages on DAC output

5.9.2. DEMO Running Result

Download the program <09_DAC_Output_Voltage_Value> to the EVAL board, the digital value is 0x7ff0, its converted analog voltage should be VREF/2, using the voltmeter to measure PA4, its value is 1.648V.

5.10. Comparator_Obtain_Brightness

5.10.1. DEMO Purpose

This Demo includes the following functions of GD32 MCU:

- Learn to use comparator output compare result

There are two comparators on EVAL board and each comparator has two inputs. In this demo, one input is 3.3V, and the other one is the 1/4 reference voltage. Compare the two input voltages, the output is a high or low level, and the LED2 will performs the corresponding action.

5.10.2. DEMO Running Result

Download the program <10_Comparator_obtain_brightness> to the EVAL board, comparing two input voltage, if output level is high, LED2 is off, otherwise LED2 is off.

5.11. I2C_EEPROM

5.11.1. DEMO Purpose

This Demo includes the following functions of GD32 MCU:

- Learn how to use the master transmitting mode of the I2C module
- Learn how to use the master receiving mode of the I2C module
- Learn to read and write the EEPROM with the I2C interface

5.11.2. DEMO Running Result

Jump JP13 to USART, then download the program <11_I2C_EEPROM> to the EVAL board and run. Connect serial cable to COM, and open the HyperTerminal to show the print message.

Firstly, the data of 256 bytes will be written to the EEPROM from the address 0x00 and printed by the serial port. Then, reading the EEPROM from address 0x00 for 256 bytes and the result will be printed. Finally, compare the data that were written to the EEPROM and the data that were read from the EEPROM. If they are the same, the serial port will output "I2C-AT24C02 test passed!" and the four LEDs lights flashing, otherwise the serial port will output "Err: data read and write aren't matching." and all the four LEDs light.

The output information via the serial port is as following.

```
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_TF_Card_Block_Operation

5.12.1. DEMO Purpose

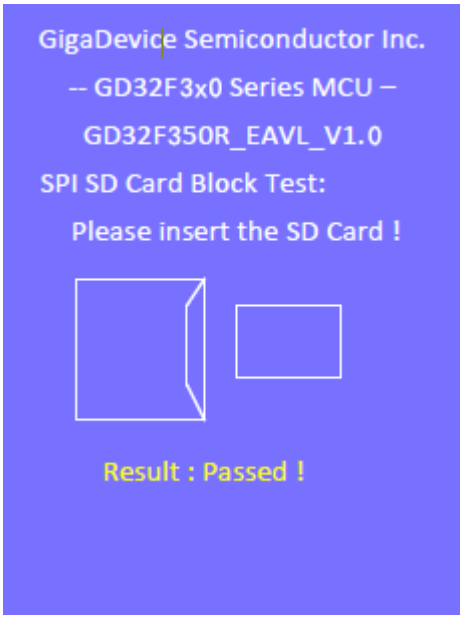
This Demo includes the following function of GD32 MCU:

- Learn how to use SPI to block read and write from TF card

In this demo, the SPI interface is used for reading from and writing to TF card. Write to TF card with 0 to 255, a total of 2048 bytes directly. Then read data from the original address, check the correctness of the written and read data. Note that, the FATFS is not used in this demo and in the absence of FATFS, read and write to the TF card will destroy the file system, please make a backup before testing.

5.12.2. DEMO Running Result

Firstly, JP10 and JP11 must be fitted to SPI and then download the program <12_SPI_TF_Card_Block_Operation> to the EVAL board. All the LEDs are turned on and then turned off for test. Insert the TF card to the board, the test results will be displayed on the LCD screen.



5.13. SPI_TF_Card_FATFS_Operation

5.13.1. DEMO Purpose

This Demo includes the following function of GD32 MCU:

- Learn how to use SPI to read and write from TF card with FATFS

In this demo, the SPI interface is used for reading from and writing to TF card. Make sure the TF card has FAT file system. If not, some information will be displayed on the LCD screen.

5.13.2. DEMO Running Result

Firstly, JP10 and JP11 must be fitted to SPI and then download the program <13_SPI_TF_Card_FATFS_Operation> to the EVAL board. All the LEDs are turned on and then turned off for test. Insert a formatted TF card, a text file will be created in this TF card. When the screen appears "FATFS FILE Create Success", you can remove the TF card and check whether the file is successfully created in the TF card with a card reader. When the exception occurs, operate follow the screen prompts.

5.14. SPI_TFT_LCD_Driver

5.14.1. DEMO Purpose

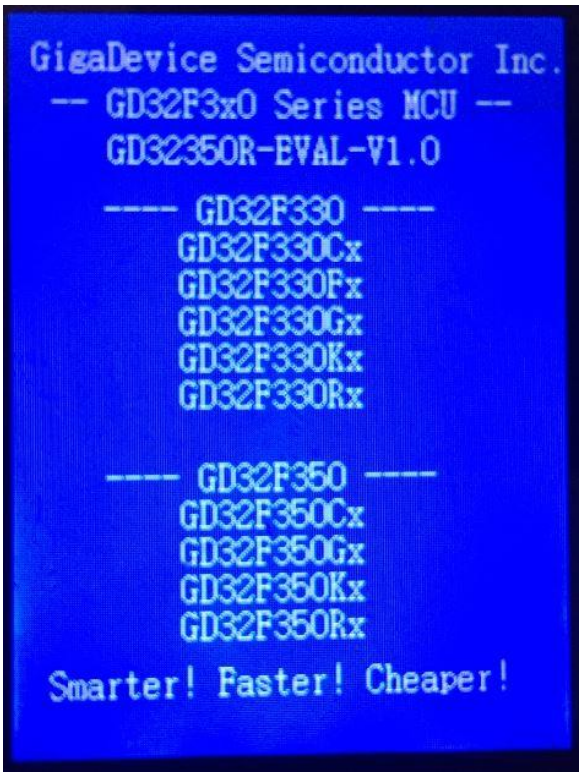
This Demo includes the following function of GD32 MCU:

- Learn how to use SPI to drive TFT LCD screen and display

GD32350R-EVAL board has a TFT LCD screen which supports SPI interface. In this demo, tests of font, number, draw and color are displayed on the LCD screen respectively.

5.14.2. DEMO Running Result

Firstly, JP10 and JP11 must be fitted to SPI port and then download the program <14_SPI_TFT_LCD_Driver> to the EVAL board. All the LEDs are turned on and then turned off for test. After that, the LCD screen on the board will display the GUI tests in infinite loop.



5.15. HDMI-CEC_HostSlaveCommunication

5.15.1. DEMO Purpose

This Demo includes the following functions of GD32 MCU:

- Learn the communication function of HDMI-CEC

In the process of communication, the sender sends data to receiver through the key interrupt, the receiver for receiving data in the CEC interrupt. The entire communication process does not make the error processing.

5.15.2. DEMO Running Result

This routines need to prepare two GD32F350_EVAL board, one board as a sender, the other as a receiver. First use the DuPont to connect CEC bus (PB8) and ground wire (GND) pins in the two board, and then download the program <15_HDMI_CEC_HostSlaveCommunication> to the board for running. When the program runs, the first development board of the LCD display is data 0, press one of the development board TAMPER key, the other piece of the development board LCD number will increase, which shows the end of a data transmission. Each it increases to 9, it will clear to 0 to re-increase; press the WAKEUP key, the number will decline, which also shows the end of a data transmission. Every time it decreases to 0, it will return to the number 9 to re-decrease.

5.16. I2S_Audio_Player

5.16.1. DEMO Purpose

This Demo includes the following functions of GD32 MCU:

- Learn to use I2S module to output audio file

GD32350R-EVAL board integrates the I2S (Inter-IC Sound) module, and the module can communicate with external devices using the I2S audio protocol. This Demo mainly shows how to use the I2S interface of the board for audio output.

5.16.2. DEMO Running Result

Download the program <16_I2S_Audio_Player>. After downloading the program, insert the earphone into the audio port J1, then listen to the audio file.

5.17. RCU_Clock_Out

5.17.1. DEMO Purpose

This Demo includes the following functions of GD32 MCU:

- Learn to use GPIO control the LED
- Learn to use the clock output function of RCU
- Learn to communicate with PC by USART

5.17.2. DEMO Running Result

Download the program <17_RCU_Clock_Out> to the EVAL board and run. Connect serial cable to EVAL_COM, open the HyperTerminal. When the program is running, HyperTerminal will display the initial information. Then user can choose the type of the output clock by pressing the USER button. After pressing, the corresponding LED will be turned on and HyperTerminal will display which mode be selected. The frequency of the output clock can be observed through the oscilloscope by PA8 pin.
Information via a serial port output as following:

```
 /=========== 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
```

5.18. PMU_sleep_wakeup

5.18.1. DEMO Purpose

This Demo includes the following functions of GD32 MCU:

- Learn to use the USART receive interrupt to wake up the PMU from sleep mode

5.18.2. DEMO Running Result

Download the program < 18_PMU_sleep_wakeup > to the EVAL board, connect serial cable to EVAL_COM. After power-on, all the LEDs are off. The MCU will enter sleep mode and the software stop running. When the USART0 receives a byte of data from the HyperTerminal, the MCU will wake up from a receive interrupt. And all the LEDs will flash together.

5.19. RTC_Calendar

5.19.1. DEMO Purpose

This Demo includes the following functions of GD32 MCU:

- Learn to use RTC module to implement calendar function
- Learn to use LCD module to display the time of calendar

5.19.2. DEMO Running Result

Download the program <19_RTC_Calendar> to the EVAL board and run. When the program is running, the four LEDs, LED1 to LED4 turn on, then turn off. And then the LCD prints out the information of the board, and the calendar. When you press the Wakeup key, the time will be configured to 2016-05-13, 12:00:00.

If place a battery on GD32F350R_EVAL board, it can realize no losing of time when power down. In this case, notice that JP5 jump to the VBAT.

5.20. IRInfrared_Transceiver

5.20.1. DEMO Purpose

This Demo includes the following functions of GD32 MCU:

- Learn to use general timer output PWM wave
- Learn to use general timer generated update interrupt
- Learn to use general timer capture interrupt
- Learn to use general timer TIMER15 and TIMER16 implement Infrared function

5.20.2. DEMO Running Result

Download the program <20_IRInfrared_Transceiver> to the EVAL board and run. When the program is running, if the infrared receiver received data is correct, LED1, LED2, LED3, LED4 light in turn, otherwise LED1, LED2, LED3, LED4 toggle together.

5.21. TIMER_Breath_LED

5.21.1. DEMO Purpose

This Demo includes the following functions of GD32 MCU:

- Learn to use Timer output PWM wave
- Learn to update channel value

5.21.2. DEMO Running Result

Use the DuPont line to connect the TIMER0_CH0 (PA8) and LED1 (PC10), and then download the program <21_TIMER_Breath_LED> to the GD32350R-EVAL board and run. PA8 should not be reused by other peripherals.
When the program is running, you can see LED1 lighting from dark to bright gradually and then gradually darken, ad infinitum, just like breathing as rhythm.

5.22. TSI_TouchKey_leds

5.22.1. DEMO Purpose

This Demo includes the following functions of GD32 MCU:

- Learn to use TSI module implement Touch Key function

5.22.2. DEMO Running Result

Download the program <22_TSI_TouchKey_leds> to the EVAL board and run. When the program is running, you can use a finger slide the Touch Sensor (A-1, B, C or A-2) on the EVAL board, and then the associated LED is light.

5.23. USB_Device

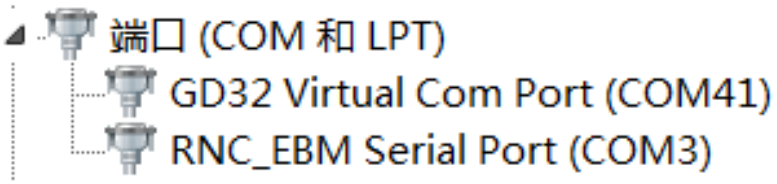
5.23.1. USBD_CDC_ACM

DEMO Purpose

This demo includes the following functions of GD32 MCU:

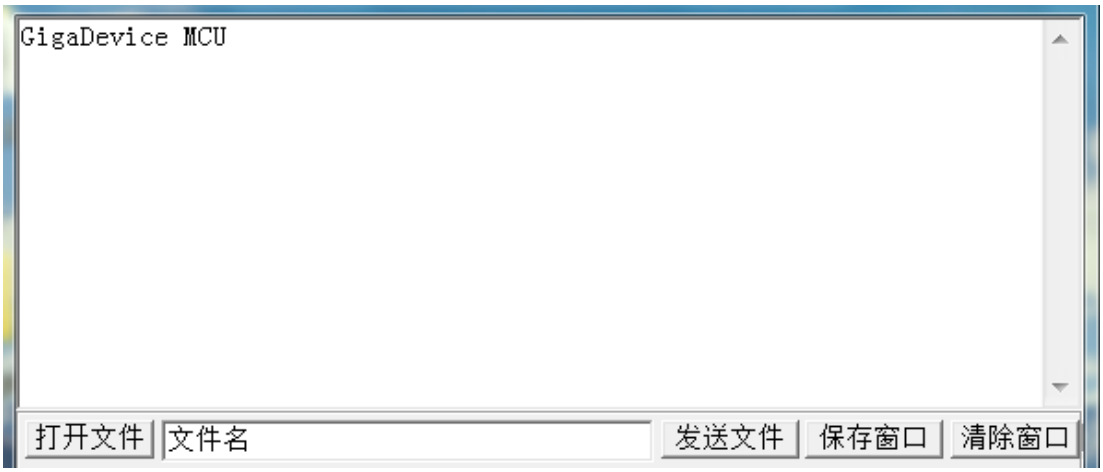
- Learn how to use the USBFS peripheral
- Learn how to implement USB CDC device

EVAL board has one USBFS interface. In this demo, the EVAL board is enumerated as an USB virtual COM port, which was shown in device manager of PC as below. This demo makes the USB device look like a serial port, and loops back the contents of a text file over USB port. To run the demo, input a message using the PC's keyboard. Any data that shows in HyperTerminal is received from the device.



DEMO Running Result

Download the program <23_USB_Device\CDC_ACM> to the EVAL board and run. When you input message through computer keyboard, the HyperTerminal will receive and shown the message. For example, when you input “GigaDevice MCU”, the HyperTerminal will get and show it as below.



5.23.2. USBD_Keyboard

DEMO Purpose

This demo includes the following functions of GD32 MCU:

- Learn how to use the USBFS peripheral mode
- Learn how to implement USB HID(human interface) device

EVAL board has four keys and one USB_FS interface. The five keys are Reset key, Wakeup key, Tamper key and User key. In this demo, the EVAL board is enumerated as an USB Keyboard, which uses the native PC Host HID driver, as shown below. The USB Keyboard uses three keys(wakeup key, tamper key and user key) to output three

characters (‘b’, ‘a’ and ‘c’). In addition, the demo also supports remote wakeup which is the ability of a USB device to bring a suspended bus back to the active condition, and the wakeup key is used as the remote wakeup source.

DEMO Running Result

Before running the demo, please ensure that jumper JP13 jump to USB. After doing this, download the program <23_USB_Device\HID_Keyboard)> to the EVAL board and run. If you press the Wakeup key, will output ‘b’. If you press the User key, will output ‘c’. If you press the Tamper key, will output ‘a’.
If you want to test USB remote wakeup function, you can do as follows:
- Manually switch PC to standby mode
- Wait for PC to fully enter the standby mode
- Push the Wakeup key
- If PC is ON, remote wakeup is OK, else failed.

5.24. USB_Host

5.24.1. USBH_HID_Host

DEMO Purpose

This demo includes the following functions of GD32 MCU:

- Learn to use the USBFS as a HID host
- Learn the operation between the HID host and the mouse device
- Learn the operation between the HID host and the keyboard device

EVAL evaluation board integrates the USBFS module, and the module can be used as a USBFS device, a USBFS host or OTG device. This demo mainly shows how to use the USBFS as a USB HID host to communicate with external USB HID device.

DEMO Running Result

Jump the JP13 to USB. Then download the program <24_USB_Host\USBH_HID> to the EVAL board and run.

If a mouse has been attached, the user will see the information of mouse enumeration. First pressing the user key will see the inserted device is mouse, and then moving the mouse will show the position of mouse and the state of button in the screen.

If a keyboard has been attached, the user will see the information of keyboard enumeration. First pressing the user key will see the inserted device is keyboard, and then pressing the keyboard will show the data which keyboard inserted in the screen.

5.24.2. USBH_MSC_Host

DEMO Purpose

This demo includes the following functions of GD32 MCU:

- Learn to use the USBFS as a MSC host
- Learn the operation between the MSC host and the Udisk

EVAL evaluation board integrates the USBFS module, and the module can be used as USBFS device, a USBFS host or OTG device. This demo mainly shows how to use the USBFS as a USB MSC host to communicate with external Udisk.

DEMO Running Result

Jump the JP13 to USB. Then insert the OTG cable to the USB port, download the program <24_USB_Host\USBH_MSC > to the EVAL board and run.

If an Udisk has been attached, the user will see the information of Udisk enumeration. First pressing the user key will see the Udisk information, next pressing the tamper key will see the root content of the Udisk, then press the wakeup key will write file to the Udisk, finally the user will see information that the MSC host demo is end.

6. Revision history

Table 6-1 Revision history

Revision No.	Description	Date
1.0	Initial Release	Jun.28, 2017

X-ON Electronics

Largest Supplier of Electrical and Electronic Components

Click to view similar products for Development Boards & Kits - Other Processors ***category:***

Click to view products by Gigadevice ***manufacturer:***

Other Similar products are found below :

KIT_AURIX_TC233LP_TRB EVB-MEC1418MECC SPC56XVTOP-M ADZS-BF506F-EZLITE ADZS-SADA2-BRD 20-101-1252 T1023RDB-PC 20-101-1267 T1042D4RDB-PA ML610Q174 REFERENCE BOARD MPC574XG-MB BSC9132QDS C29XPCIE-RDB KIT_TC1793_SK CC-ACC-18M433 P1010RDB-PB P1020RDB-PD P2020COME-DS-PB STM8S/32-D/RAIS T4240RDB-PB TRK-USB-MPC5604B TWR-56F8200 CY3674 SPC58XXADPT176S MAX1464EVKIT TRK-MPC5606B RTE510Y470TGB00000R STM8128-MCKIT MAXQ622-KIT# YRPBRL78G11 SPC58EEMU QB-R5F10JGC-TB YQB-R5F11BLE-TB SPC564A70AVB176 RTE5117GC0TGB00000R QB-R5F100LE-TB YR0K50571MS000BE YQB-R5F1057A-TB QB-R5F104PJ-TB CC-ACC-ETHMX LFM34INTPQA SPC563M64A176S Y-BLDC-SK-RL78F14 P1021RDB-PC SPC58XCADPT176S RTE510MPG0TGB00000R YRPBRX71M LFMAJ04PLT KITAURIXTC234LPSTRBTOBO1 OV-7604-C7-EVALUATION-BOARD