GigaDevice Semiconductor Inc.

GD32207C-EVAL
Arm® Cortex®-M3 32-bit MCU

User Guide

Revision 2.2

(Oct. 2021)

Table of Contents

List of Figures

List of Tables

1. Summary

GD32207C-EVAL evaluation board uses GD32F207VCT6 as the main controller. As a complete development platform of GD32F207xx connectivity line powered by ARM® Cortex®-M3 core, the board supports full range of peripherals. It uses Mini USB interface or AC/DC adapter as 5V power supply. JTAG, Reset, Boot, User button key, LED, CAN, I2C, USART, RTC, EXMC, SPI, USBFS, ADC, DAC, GD-Link、LCD、SDIO, ENET and Extension Pin are also included. This document details its hardware schematic and the relevant applications.

2. Function pin assignment

Table 2-1. Function pin assignment

Function	Pin	Description
LED	PC0	LED2
	PC2	LED3
	PE0	LED4
	PE1	LED5
RESET		K1-Reset
KEY	PA0	KEY1
	PC13	KEY2
	PB14	KEY3
USBFS	PA9	USB_VBUS
	PA11	USB_DM
	PA12	USB_DP
	PD13	VBUS control pin
CAN	PD0	CAN0_RX
	PD1	CAN0_TX
	PB5	CAN1_RX
	PB6	CAN1_TX
I2C	PB6	I2C0_SCL
	PB7	I2C0_SDA
USART0	PA9	USART0_TX
	PA10	USART0_RX
EXMC	PD14	EXMC_D0
	PD15	EXMC_D1
	PD0	EXMC_D2
	PD1	EXMC_D3
	PE7	EXMC_D4
	PE8	EXMC_D5
	PE9	EXMC_D6
	PE10	EXMC_D7
	PD11	EXMC_A16
	PD12	EXMC_A17
	PD4	EXMC_NOE
	PD5	EXMC_NWE
	PD6	EXMC_NWAIT
	PD7	EXMC_NCE1
SPI	PA2	SPI0_IO3
	PA3	SPI0_IO4
	PA5	SPI0_SCK

Function	Pin	Description
	PA6	SPI0_MISO
	PA7	SPI0_MOSI
	PE3	SPIFlash_CS
ADC	PC3	ADC012_IN13
DAC	PA4	DAC_OUT0
	PA5	DAC_OUT1
SDIO	PC12	SDIO_CLK
	PD2	SDIO_CMD
	PC8	SDIO_DAT0
	PC9	SDIO_DAT1
	PC10	SDIO_DAT2
	PC11	SDIO_DAT3
LCD	PC6	HSYNC
	PA4	VSYNC
	PE14	PCLK
	PE13	DE
	PD7	LCD_CS
	PE2	LCD_RS
	PA5	LCD_SCK
	PA7	LCD_MOSI
	PE4	T_CS
	PE6	T_SCK
	PD8	T_MOSI
	PD9	T_MISO
	PE5	T_IQR
	PE15	D15
	PB1	D14
	PA12	D13
	PA11	D12
	PB0	D11
	PD3	D10
	PC7	D09
	PB11	D08
	PB10	D07
	PE11	D06
	PA6	D05
	PB9	D04
	PB8	D03
	PA3	D02
	PE12	D01
	PD10	D00

Function	Pin	Description
	RESET	K1-Reset
ENET	PB11	RMII_TX_EN
	PB12	RMII_TXD0
	PB13	RMII_TXD1
	PC4	RMII_RXD0
	PC5	RMII_RXD1
	PA7	RMII_CRS_DV
	PC1	RMII_MDC
	PA2	RMII_MDIO
	PB0	RMII_INT
	PA1	RMII_REF_CLK

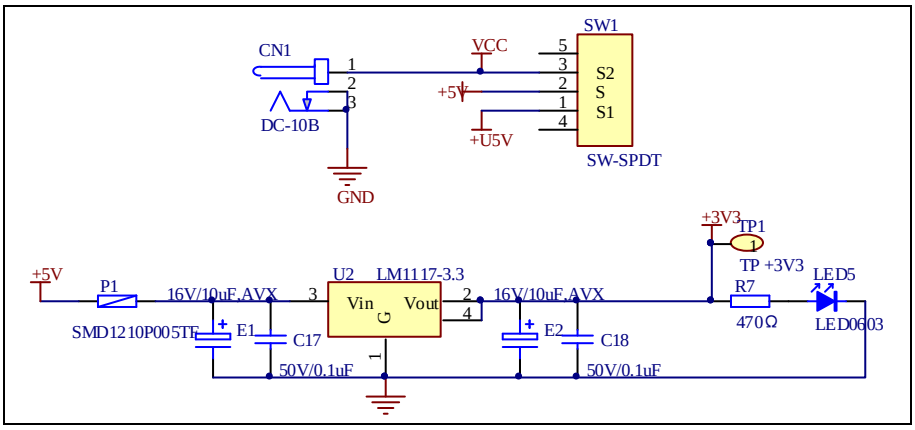
3. Getting started

The EVAL Board uses Mini USB connecter 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 LED5 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

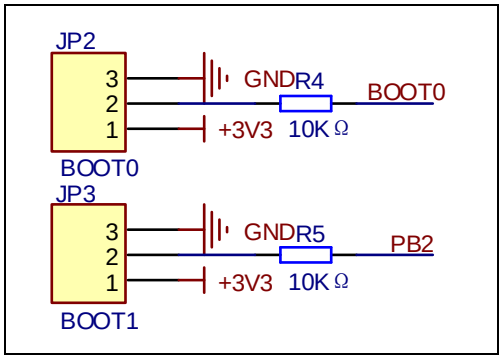
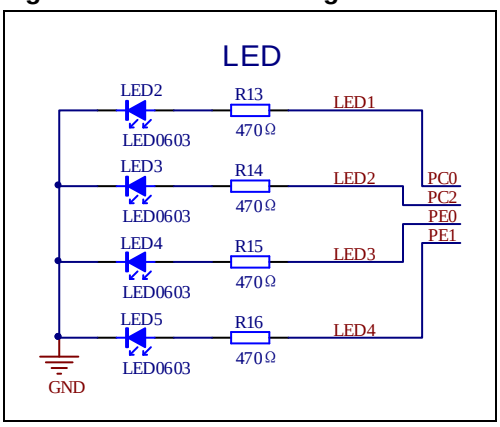


Table 4-1. Boot configuration

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

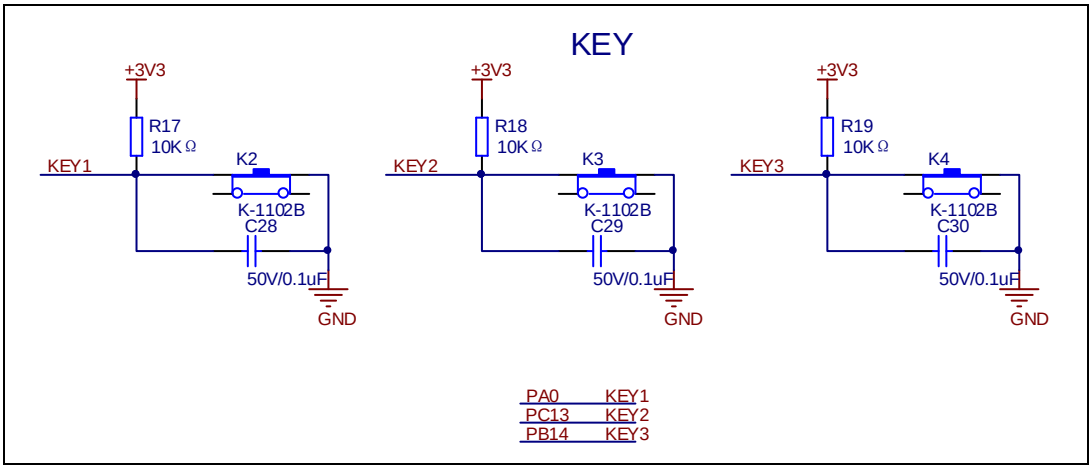
4.3. LED

Figure 4-3. Schematic diagram of LED function



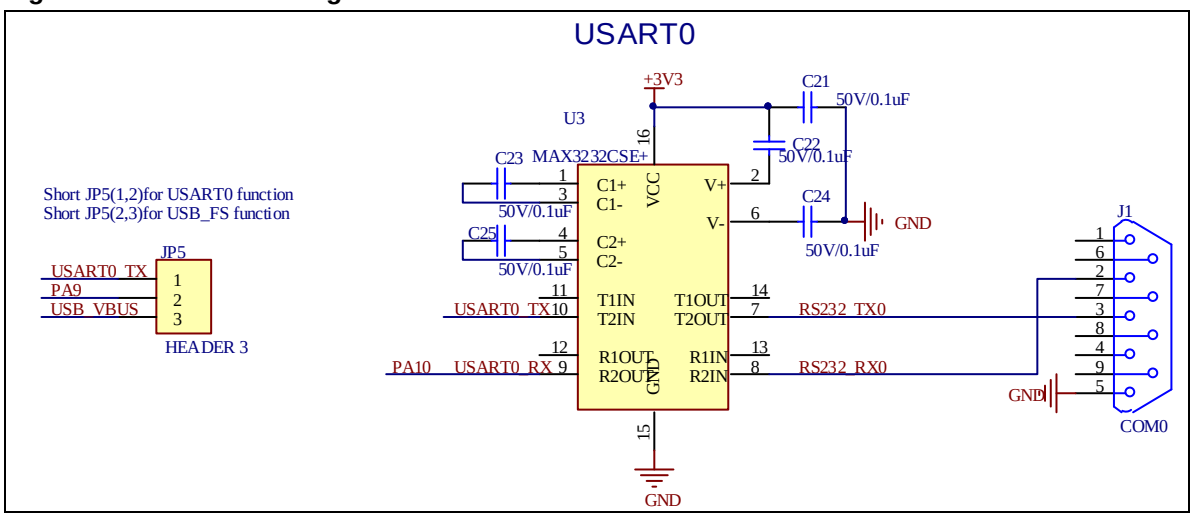
4.4. KEY

Figure 4-4. Schematic diagram of Key function



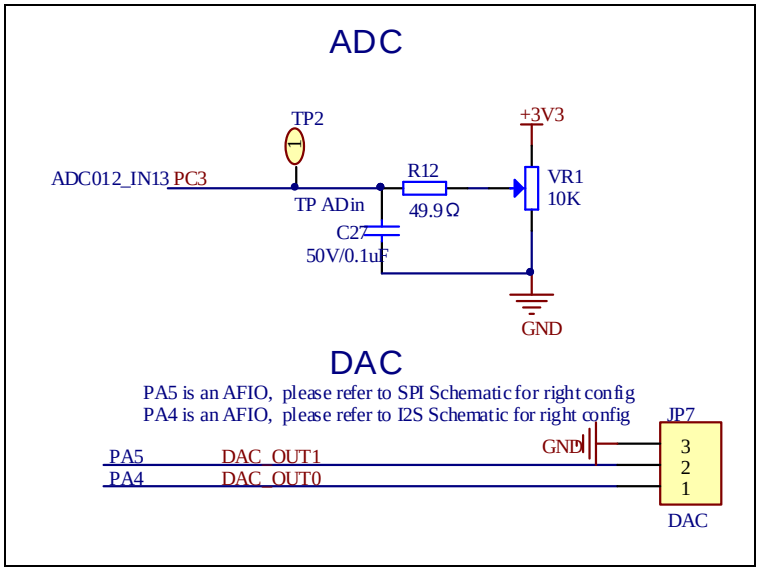
4.5. USART

Figure 4-5. Schematic diagram of USART0



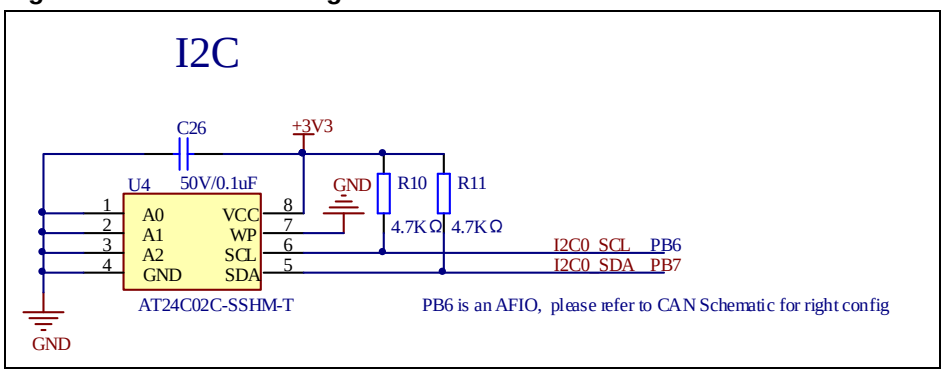
4.6. ADC/DAC

Figure 4-6. Schematic diagram of ADC/DAC



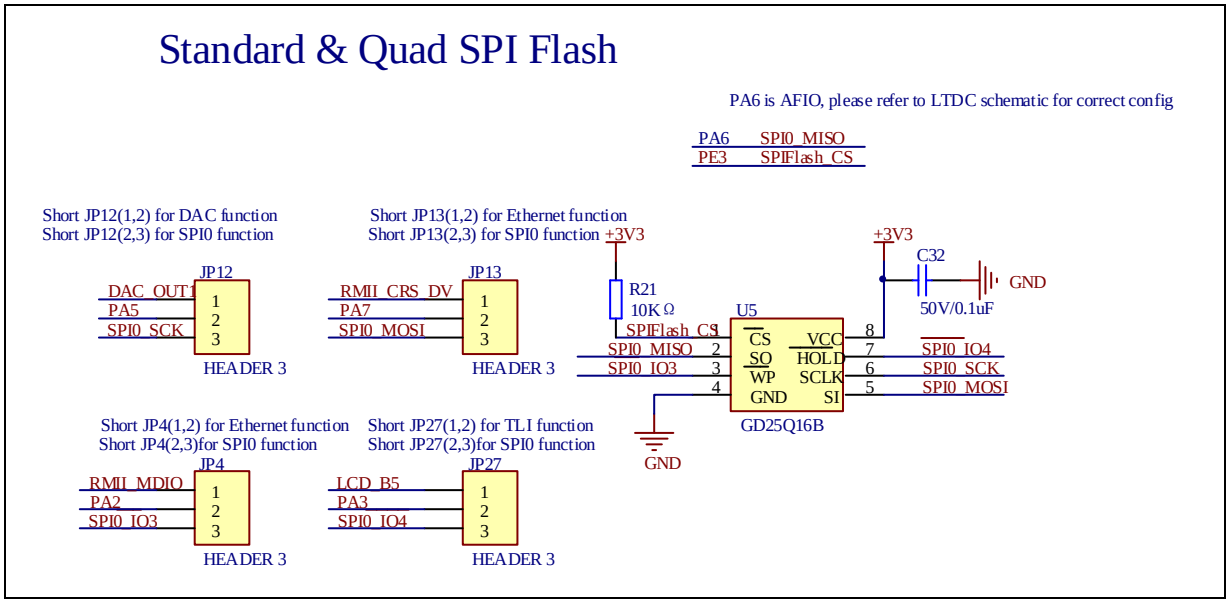
4.7. I2C

Figure 4-7. Schematic diagram of I2C



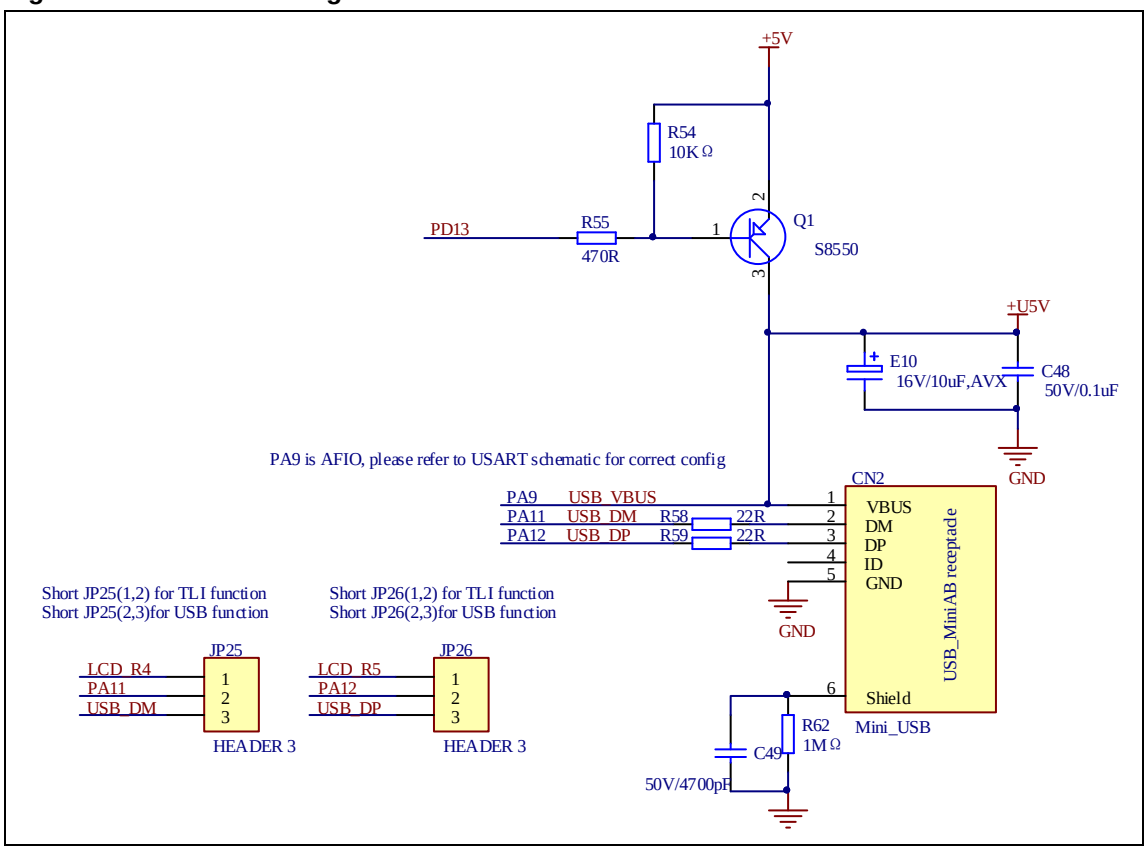
4.8. SPI-Serial Flash

Figure 4-8. Schematic diagram of SPI-Serial Flash



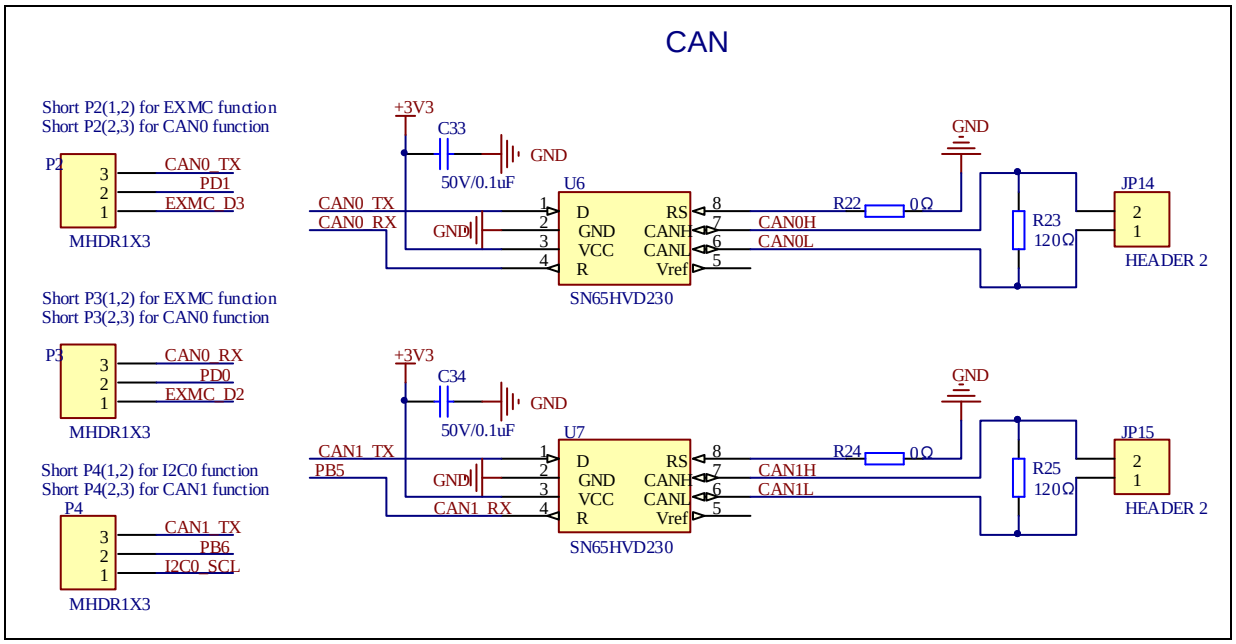
4.9. USB

Figure 4-9. Schematic diagram of USB



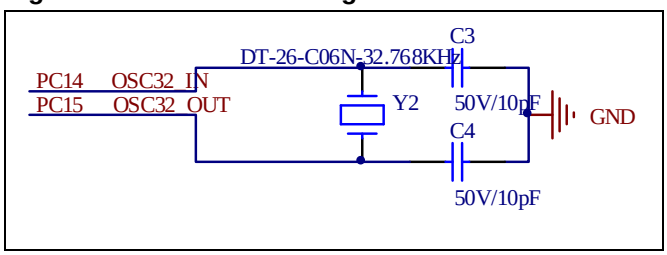
4.10. CAN

Figure 4-10. Schematic diagram of CAN



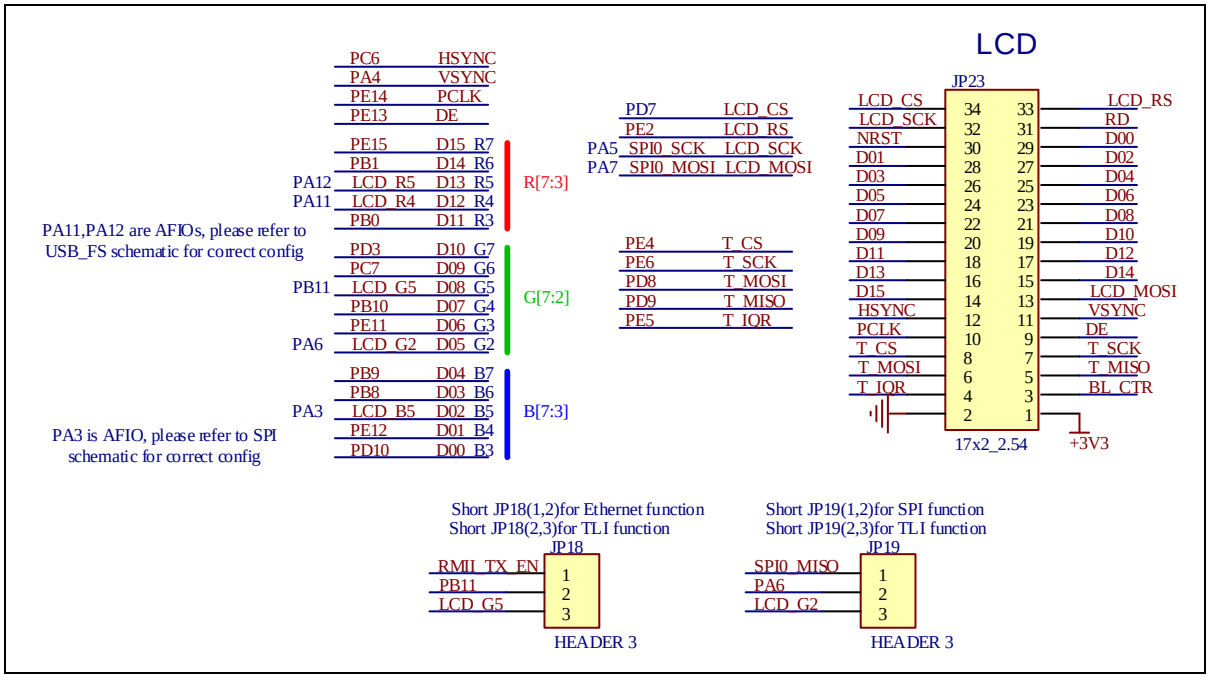
4.11. RTC

Figure 4-11. Schematic diagram of RTC



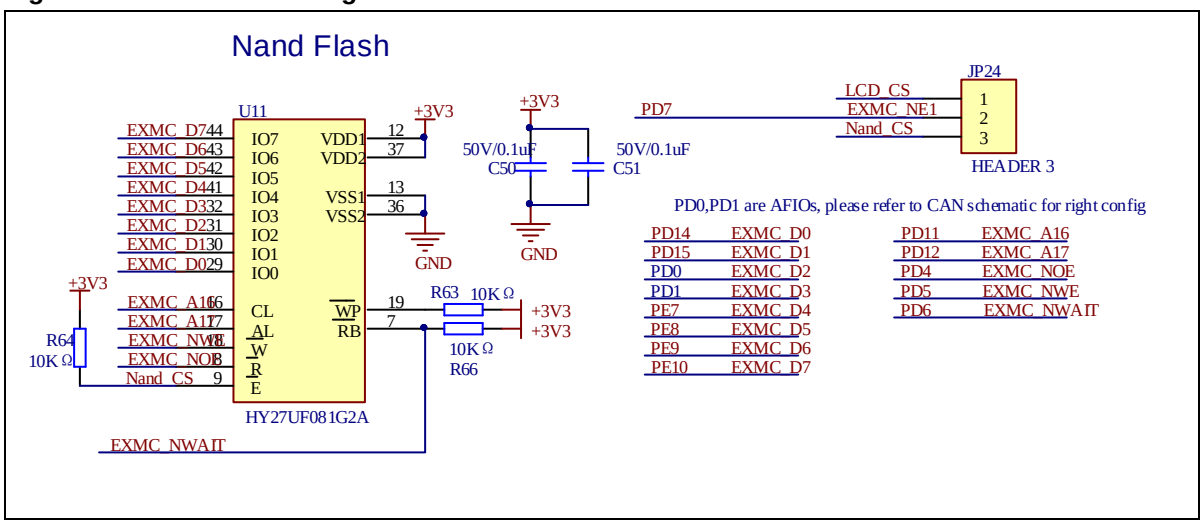
4.12. LCD

Figure 4-12. Schematic diagram of LCD



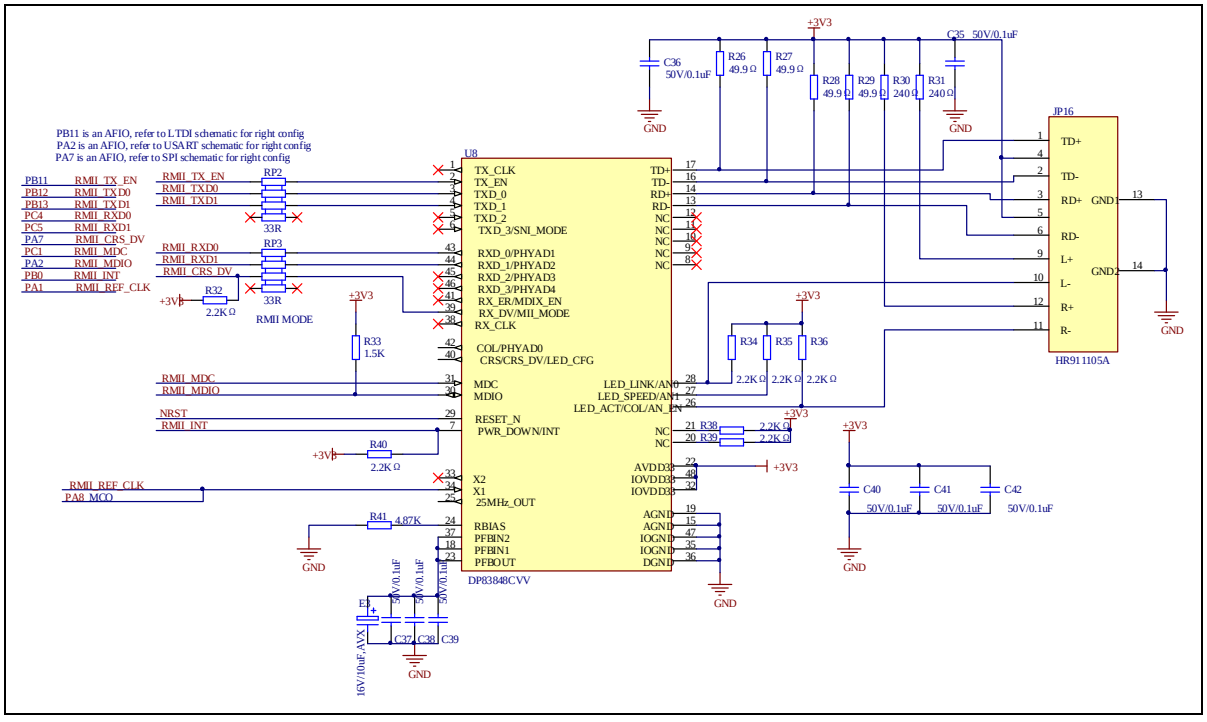
4.13. EXMC-NAND Flash

Figure 4-13. Schematic diagram of EXMC-NAND Flash



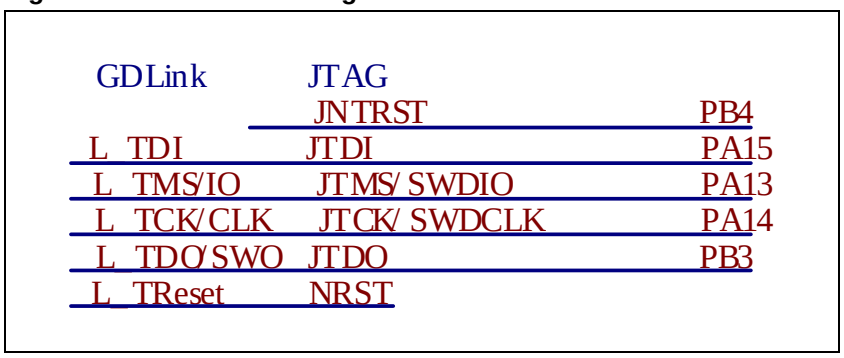
4.14. ENET

Figure 4-14. Schematic diagram of ENET



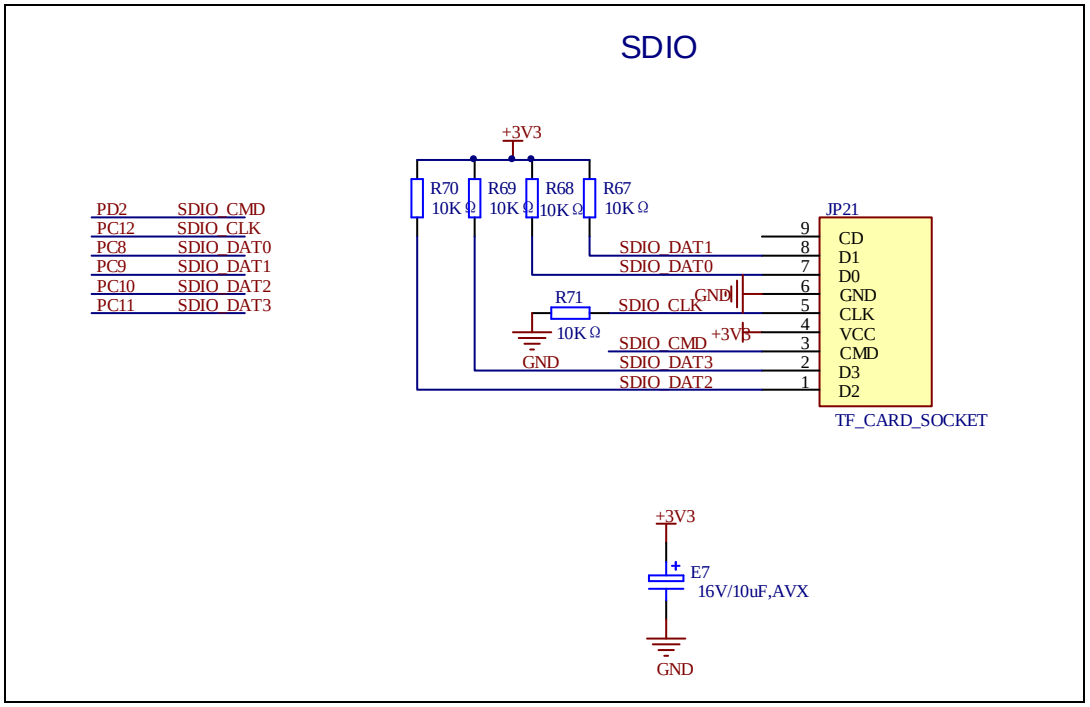
4.15. GD-Link

Figure 4-15. Schematic diagram of GD-Link



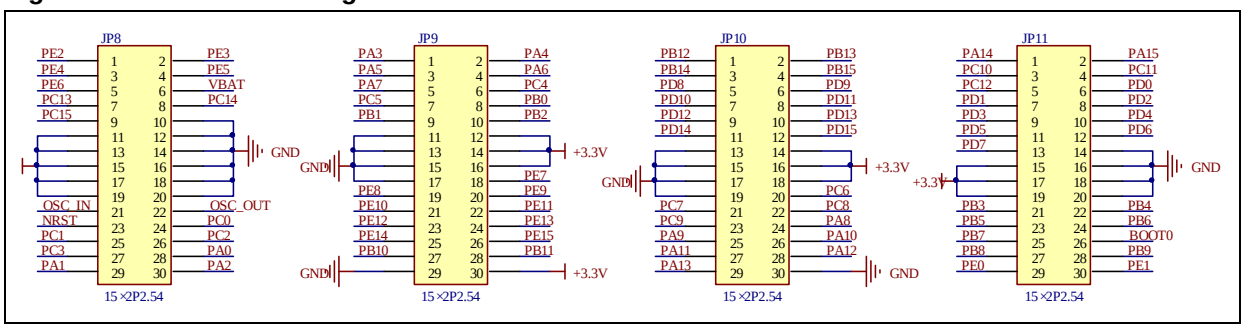
4.16. SDIO

Figure 4-16. Schematic diagram of SDIO



4.17. Extension

Figure 4-17. Schematic diagram of Extension Pin



4.18. Pin jumper comparison table

The GD32207C-EVAL evaluation boards involved in this paper all mean to the GD32207C-EVAL-V1.2 evaluation board, the pin jumpers involved in this paper are based on the GD32207C-EVAL-V1.2 evaluation board hardware schematic, the following table is a pin jumper comparison table.

Table 4-2. Pin jumper comparison table

Hardware Schematic	User Manual	PCB Screen Printing
JP5(1,2)for USART0	JP5(1,2)for USART0	JP5(1,2)for USART1
JP4(1,2) for Ethernet	JP4(1,2) for Ethernet	JP4(1,2) for Eth
JP4(2,3)for SPI0	JP4(2,3)for SPI0	JP4(2,3)for SPI
JP7 pin1 for DAC0	JP7 pin1 for DAC0	JP7 pin1 for DAC1
JP7 pin2 for DAC1	JP7 pin2 for DAC1	JP7 pin2 for DAC2
JP12(1,2) for DAC	JP12(1,2) for DAC	JP12(1,2) for DAC2
JP12(2,3) for SPI0	JP12(2,3) for SPI0	JP12(2,3) for SPI/Lcd
JP13(1,2) for Ethernet	JP13(1,2) for Ethernet	JP13(1,2) for Eth
JP13(2,3) for SPI0	JP13(2,3) for SPI0	JP13(2,3) for SPI/Lcd
JP14 for CAN0	JP14 for CAN0	JP14 for CAN1
JP15 for CAN1	JP15 for CAN1	JP15 for CAN2
JP18(2,3)for TLI	JP18(2,3)for TLI	JP18(2,3)for Lcd
JP18(1,2)for Ethernet	JP18(1,2)for Ethernet	JP18(1,2)for Eth
JP19(2,3)for TLI	JP19(2,3)for TLI	JP19(2,3)for Lcd
LED1	LED1	LED2
LED2	LED2	LED3
LED3	LED3	LED4
LED4	LED4	LED5
LED5	LED5	LED1
P2(2,3) for CAN0	P2(2,3) for CAN0	P2(2,3) for CAN1
P3(2,3) for CAN0	P3(2,3) for CAN0	P3(2,3) for CAN1

Hardware Schematic	User Manual	PCB Screen Printing
P4(2,3) for CAN1	P4(2,3) for CAN1	P4(2,3) for CAN2
JP25(1,2) for TLI	JP25(1,2) for TLI	JP25(1,2) for Lcd
JP26(1,2) for TLI	JP26(1,2) for TLI	JP26(1,2) for Lcd

5. Routine use guide

5.1. GPIO_Running_LED

5.1.1. DEMO purpose

This demo includes the following functions of GD32 MCU:

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

GD32207C-EVAL evaluation board has four LEDs: LED2~LED5. The LEDs 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, LED2, LED3, LED4, LED5 will turn on in sequence with interval of 1000ms, firstly, LED2 on, then, LED3 on, four LEDs can light periodically.

5.2. GPIO_Key_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

GD32207C-EVAL evaluation board has three keys and four LEDs. The three keys are Tamper key, Wakeup key and User key. LED2~LED5 are controlled by GPIO.

This demo will show how to use the Tamper key to control the LED2. When press down the Tamper Key, it will check the input value of the IO port. If the value is 0 and will wait for 100ms. Check the input value of the IO port again. If the value still is 0, it indicates that the button is pressed successfully and toggle LED2.

5.2.2. DEMO running result

Download the program <02_GPIO_Key_Polling_mode> to the EVAL board, press down the Tamper key, LED2 will be turned on. Press down the Tamper key again, LED2 will be turned off.

5.3. EXTI_Key_Interrupt_mode

5.3.1. DEMO Purpose

This demo includes the following functions of GD32 MCU:

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

GD32207C-EVAL evaluation board has three keys and four LEDs. The three keys are Tamper key, Wakeup key and User key. LED2~LED5 are controlled by GPIO.

This demo will show how to use the 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_EXTI_Key_Interrupt_mode> to the EVAL board, 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 control the LED
- Learn to retarget the C library printf function to the USART

5.4.2. DEMO Running Result

Download the program <04_USART_Printf> to the EVAL board, fit the JP5 to USART and connect serial cable to USART. Firstly, all the LEDs flash 2 times for test. Then, this implementation outputs “USART printf example: please press the Tamper key” on the HyperTerminal using USART. Press the Tamper key, the serial port will output “USART printf example”.

The output information via the serial port is as following.

```
USART printf example: please press the Tamper key
USART printf example
```

5.5. USART_HyperTerminal_Interrupt

5.5.1. DEMO Purpose

This demo includes the following functions of GD32 MCU:

- Learn to use the USART transmit and receive interrupts to communicate with the serial terminal tool

5.5.2. DEMO Running Result

Download the program <05_USART_HyperTerminal_Interrupt> to the EVAL board, fit the JP5 to USART and connect serial cable to USART0. Firstly, all the LEDs are turned on and off for test. Then, the USART 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, LED2, LED3, LED4, LED5 flash by turns. Otherwise, LED2, LED3, LED4, LED5 toggle together.

The output information via the HyperTerminal is 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 USART transmit and receive data using DMA

5.6.2. DEMO Running Result

Download the program <06_USART_DMA> to the EVAL board, fit the JP5 to USART and connect serial cable to USART0. Firstly, the USART sends “USART DMA interrupt receive and transmit example, please input 10 bytes:” to hyperterminal and waits for receiving 10 bytes data from the hyperterminal that you must send. After MCU receives the data, the

USART will continue to output the received data to the hyper terminal.

The output information via the HyperTerminal is as following:

5.7. ADC_Temperature_Vrefint

5.7.1. DEMO Purpose

This demo includes the following functions of GD32 MCU:

- Learn to use the ADC to convert analog signal to digital data
- Learn to get the value of inner channel 16 (temperature sensor channel) and channel 17 (V_{REFINT} channel)

5.7.2. DEMO Running Result

The computer serial port line connected to the COM0 port of development board, set the baud rate of HyperTerminal software to 115200, 8 bits data bit, 1 bit stop bit. At the same time you should jump the JP5 to USART1.

Download the program <07_ADC_Temperature_Vrefint> to the EVAL board, the HyperTerminal software can observe the operation condition. When the program is running, HyperTerminal will display the value of temperature and internal voltage reference (VREFINT).

Notice: because there is an offset, when inner temperature sensor is used to detect accurate temperature, an external temperature sensor part should be used to calibrate the offset error.

The following is the experimental results.

```
the temperature data is 48 degrees Celsius
the reference voltage data is 1.192V

the temperature data is 48 degrees Celsius
the reference voltage data is 1.192V

the temperature data is 48 degrees Celsius
the reference voltage data is 1.193V

the temperature data is 49 degrees Celsius
the reference voltage data is 1.187V

the temperature data is 48 degrees Celsius
the reference voltage data is 1.188V

the temperature data is 49 degrees Celsius
the reference voltage data is 1.189V

the temperature data is 48 degrees Celsius
the reference voltage data is 1.190V

the temperature data is 49 degrees Celsius
the reference voltage data is 1.189V

the temperature data is 49 degrees Celsius
the reference voltage data is 1.190V
```

5.8. ADC0_ADC1_Follow_up_mode

5.8.1. DEMO Purpose

This demo includes the following functions of GD32 MCU:

- Learn to use the ADC to convert analog signal to digital data
- Learn to use ADC0 and ADC1 follow-up mode

5.8.2. DEMO Running Result

The computer serial port line connected to the COM0 port of development board, set the baud rate of HyperTerminal software to 115200, 8 bits data bit, 1 bit stop bit. At the same time you should jump the JP5 to USART1.

Download the program <08_ADC0_ADC1_Follow_up_mode> to the EVAL board, the HyperTerminal software can observe the operation condition. When the program is running, HyperTerminal display the regular value of ADC0 and ADC1 by adc_value.

TIMER0_CH0 is the trigger source of ADC0 and ADC1. When the rising edge of

TIMER0_CH0 coming, ADC0 starts immediately and ADC1 starts after a delay of several ADC clock cycles. The values of ADC0 and ADC1 are transmitted to array adc_value by DMA.

When the rising edge of TIMER0_CH0 coming, the value of the ADC0 conversion of PC3 pin is stored into the low half word of adc_value, and after a delay of several ADC clock cycles the value of the ADC1 conversion of PC3 pin is stored into the high half word of adc_value.

The following is the experimental results.

```
the data adc_value is 03A203A9
the data adc_value is 03A203A9
the data adc_value is 03A103A7
the data adc_value is 03A103A9
the data adc_value is 03A103A8
the data adc_value is 03A103A6
the data adc_value is 03A303A7
the data adc_value is 03A303A9
the data adc_value is 03A203A8
the data adc_value is 03A003A8
the data adc_value is 03A103A8
```

5.9. ADC0_ADC1_Regular_Parallel_mode

5.9.1. DEMO Purpose

This demo includes the following functions of GD32 MCU:

- Learn to use the ADC to convert analog signal to digital data
- Learn to use ADC0 and ADC1 regular parallel mode

5.9.2. DEMO Running Result

The computer serial port line connected to the COM0 port of development board, set the baud rate of HyperTerminal software to 115200, 8 bits data bit, 1 bit stop bit. At the same time you should jump the JP5 to USART1.

Download the program <09_ADC0_ADC1_Regular_Parallel_mode> to the EVAL board, the HyperTerminal software can observe the operation condition. When the program is running, HyperTerminal display the regular value of ADC0 and ADC1 by adc_value[0] and adc_value[1].

TIMER0_CH0 is the trigger source of ADC0 and ADC1. When the rising edge of TIMER0_CH0 coming, ADC0 and ADC1 starts immediately. The values of ADC0 and ADC1 are transmitted to array adc_value[] by DMA.

When the first rising edge of TIMER0_CH0 coming, the value of the ADC0 conversion of PC3 pin is stored into the low half word of adc_value[0], the value of the ADC1 conversion of PC5 pin is stored into the high half word of adc_value[0]. When the second rising edge of TIMER0_CH0 coming, the value of the ADC0 conversion of PC5 pin is stored into the low half word of adc_value[1], the value of the ADC1 conversion of PC3 pin is stored into the high half word of adc_value[1].

The following is the experimental results.

```
the data adc_value[0] is 000003AD
the data adc_value[1] is 03AA0003

the data adc_value[0] is 000103AC
the data adc_value[1] is 03AB0001

the data adc_value[0] is 000003AC
the data adc_value[1] is 03AA0008

the data adc_value[0] is 000103AC
the data adc_value[1] is 03AB0000

the data adc_value[0] is 000203AC
the data adc_value[1] is 03A90000

the data adc_value[0] is 000103AD
the data adc_value[1] is 03A80000

the data adc_value[0] is 000103AB
the data adc_value[1] is 03AA0000

the data adc_value[0] is 000003AC
the data adc_value[1] is 03A90000
```

5.10. DAC_Output_Voltage_Value

5.10.1. DEMO Purpose

This demo includes the following functions of GD32 MCU:

- Learn to use DAC to output voltage on DAC0 output

5.10.2. DEMO Running Result

Download the program <10_DAC_Output_Voltage_Value> to the EVAL board and run, all the LEDs will turn on and turn off for test. The digital value is 0x7FF0, its converted analog voltage should be 1.65V (VREF/2), using the voltmeter to measure PA4 or DA0 on JP7, its value is 1.65V.

5.11. I2C_EEPROM

5.11.1. Demo Purpose

This demo includes the following functions of GD32 MCU:

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

5.11.2. DEMO Running Result

Jump the JP5 to USART1 with the jumper cap, and download the program <11_I2C_EEPROM> to the EVAL board and run. Connect serial cable to COM0, 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_QSPI_Flash

5.12.1. DEMO Purpose

This demo use SPI0 interface of GD32207C-EVAL evaluation board to read and write SPI NOR FLASH at quad SPI mode. The SPI NOR FLASH is a serial FLASH memory chip GD25Q16B which size is 16Mbit. The chip supports standard SPI and quad SPI operation instructions.

5.12.2. DEMO Running Result

Ensure GD32207C-EVAL evaluation board JP4/JP12/JP13/JP19/JP27 jumper cap jump to SPI, computer serial port line connected to the COM0 port of development board, set the baud rate of HyperTerminal software to 115200, 8 bits data bit, 1 bit stop bit. Download the program <12_SPI_QSPI_Flash> to the EVAL board, then the ID of the flash and 256 bytes data which write to and read from flash will be displayed on the HyperTerminal. The following is the experimental a part of results. If the data written to and read from the flash is the same, then we can see “ SPI-GD25Q16 Test Passed! ”.

5.13. EXMC_NandFlash

5.13.1. DEMO Purpose

This demo includes the following functions of GD32 MCU:

- Learn to use EXMC control the NAND flash

5.13.2. DEMO Running Result

GD32207C-EVAL evaluation board has EXMC module to control NAND flash. Before running the demo, JP24 must be fitted to Nand, P2 and P3 must be fitted to EXMC, JP5 must be fitted to USART0. Download the program <13_EXMC_NandFlash> to the EVAL board. This demo shows the write and read operation process of NAND flash memory by EXMC module. If the test pass, LED2 will be turned on. Otherwise, turn on the LED4. Information via a HyperTerminal output as following:

```
read NAND ID
Nand flash ID:0xAD 0xF1 0x80 0x1D

write data successfully!
read data successfully!
the result to access the nand flash:
access NAND flash successfully!
printf data to be read:
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
```

5.14. TRNG_Get_Random

5.14.1. DEMO Purpose

This demo includes the following functions of GD32 MCU:

- Learn to use TRNG to generate the random number
- Learn to communicate with PC by USART

5.14.2. DEMO Running Result

Jump the JP5 to USART1 with the jumper cap, and download the program <14_TRNG_Get_Random> to the EVAL board and run. Connect serial cable to EVAL_COM0, open the serial terminal tool supporting hex format communication. When the program is running, the serial terminal tool will display the initial information. User can use the serial terminal tool to input the minimum and maximum values (for example, the minimum value is 0x00, the maximum value is 0xDD), then application will generate random number in the input range and display it by the serial terminal tool.

Information via a serial port output as following:

```
/==============Gigadevice TRNG test==============/
TRNG init ok
Please input min num (hex format):
Please input max num (hex format):
Input min num is 0
Input max num is 221
Generate random num1 is 112
Generate random num2 is 26
```

5.15. CAU

5.15.1. DEMO Purpose

This demo includes the following functions of GD32 MCU:

- Learn DES, Triple-DES and AES algorithm
- Learn Electronic codebook (ECB) mode, Cipher block chaining (CBC) mode and Counter mode (CTR) mode
- Learn to use CAU to encrypt and decrypt
- Learn to communicate with PC by USART

5.15.2. DEMO Running Result

Jump the JP5 to USART with the jumper cap, and download the program <15_CAU> to the EVAL board and run. Connect serial cable to COM0. When the program is running, the serial terminal tool will display the information, as shown in the following figure. Plaintext data value, the encryption algorithm, and the mode can be selected are shown. After the user setting the algorithm and mode according to the serial output information indicating, serial port will print out selected algorithm and mode, as shown below.

```
Plain data :
0x6B 0xC1 0xBE 0xE2 0x2E 0x40 0x9F 0x96 0xE9 0x3D 0x7E 0x11 0x73 0x93 0x17 0x2A  [Block 0]
0xAE 0x2D 0x8A 0x57 0x1E 0x03 0xAC 0x9C 0x9E 0xB7 0x6F 0xAC 0x45 0xAF 0x8E 0x51  [Block 1]
0x30 0xC8 0x1C 0x46 0xA3 0x5C 0xE4 0x11 0xE5 0xFB 0xC1 0x19 0x1A 0x0A 0x52 0xEF  [Block 2]
0xF6 0x9F 0x24 0x45 0xDF 0x4F 0x9B 0x17 0xAD 0x2B 0x41 0x7B 0xE6 0x6C 0x37 0x10  [Block 3]
========Choose CAU algorithm=======
1: DES algorithm
2: TDES algorithm
3: AES algorithm

You choose to use AES algorithm
========Choose CAU mode=======
1: ECB mode
2: CBC mode
3: CTR mode only when choose AES algorithm

You choose to use CTR mode
```

After selection, the program starts encryption and decryption operations, the results are printed through the serial port.

```
Encrypted Data with AES 128  Mode  CTR :

0x3B 0x3F 0xD9 0x2E 0xB7 0x2D 0xAD 0x20 0x33 0x34 0x49 0xF8 0xE8 0x3C 0xFB 0x4A   [Block 0]
0x01 0x0C 0x04 0x19 0x99 0xE0 0x3F 0x36 0x44 0x86 0x24 0x48 0x3E 0x58 0x2D 0x0E   [Block 1]
0xA6 0x22 0x93 0xCF 0xA6 0xDF 0x74 0x53 0x5C 0x35 0x41 0x81 0x16 0x87 0x74 0xDF   [Block 2]
0x2D 0x55 0xA5 0x47 0x06 0x27 0x3C 0x50 0xD7 0xB4 0xF8 0xA8 0xCD 0xDC 0x6E 0xD7   [Block 3]


Decrypted Data with AES 128  Mode  CTR :

0x6B 0xC1 0xBE 0xE2 0x2E 0x40 0x9F 0x96 0xE9 0x3D 0x7E 0x11 0x73 0x93 0x17 0x2A   [Block 0]
0xAE 0x2D 0x8A 0x57 0x1E 0x03 0xAC 0x9C 0x9E 0xB7 0x6F 0xAC 0x45 0xAF 0x8E 0x51   [Block 1]
0x30 0xC8 0x1C 0x46 0xA3 0x5C 0xE4 0x11 0xE5 0xFB 0xC1 0x19 0x1A 0x0A 0x52 0xEF   [Block 2]
0xF6 0x9F 0x24 0x45 0xDF 0x4F 0x9B 0x17 0xAD 0x2B 0x41 0x7B 0xE6 0x6C 0x37 0x10   [Block 3]
```

```
Encrypted Data with AES 192  Mode  CTR :

0xCD 0xC8 0x0D 0x6F 0xDD 0xF1 0x8C 0xAB 0x34 0xC2 0x59 0x09 0xC9 0x9A 0x41 0x74   [Block 0]
0x37 0xD8 0xA6 0x39 0x17 0x1F 0xDC 0xCA 0x63 0xEB 0xD1 0x7C 0xE2 0xD7 0x32 0x1A   [Block 1]
0x79 0xA0 0xC9 0x6B 0x53 0xC7 0xEE 0xEC 0xD9 0xED 0x71 0x57 0xC4 0x44 0xFC 0x7A   [Block 2]
0x84 0x5C 0x37 0xB2 0xF5 0x11 0x69 0x7B 0x0E 0x89 0xD5 0xED 0x60 0xC4 0xD4 0x9E   [Block 3]


Decrypted Data with AES 192  Mode  CTR :

0x6B 0xC1 0xBE 0xE2 0x2E 0x40 0x9F 0x96 0xE9 0x3D 0x7E 0x11 0x73 0x93 0x17 0x2A   [Block 0]
0xAE 0x2D 0x8A 0x57 0x1E 0x03 0xAC 0x9C 0x9E 0xB7 0x6F 0xAC 0x45 0xAF 0x8E 0x51   [Block 1]
0x30 0xC8 0x1C 0x46 0xA3 0x5C 0xE4 0x11 0xE5 0xFB 0xC1 0x19 0x1A 0x0A 0x52 0xEF   [Block 2]
0xF6 0x9F 0x24 0x45 0xDF 0x4F 0x9B 0x17 0xAD 0x2B 0x41 0x7B 0xE6 0x6C 0x37 0x10   [Block 3]
```

```
Encrypted Data with AES 256  Mode  CTR :

0xDC 0x7E 0x84 0xBF 0xDA 0x79 0x16 0x4B 0x7E 0xCD 0x84 0x86 0x98 0x5D 0x38 0x60   [Block 0]
0xD5 0x77 0x78 0x8B 0x8D 0x8A 0x85 0x74 0x55 0x13 0xA5 0xD5 0x0F 0x82 0x1F 0x30   [Block 1]
0xFF 0xE9 0x6D 0x5C 0xF5 0x4B 0x23 0x8D 0xCC 0x8D 0x67 0x83 0xA8 0x7F 0x3B 0xEA   [Block 2]
0xE9 0xAF 0x54 0x63 0x44 0xCB 0x9C 0xA4 0xD1 0xE5 0x53 0xFF 0xC0 0x6B 0xC7 0x3E   [Block 3]


Decrypted Data with AES 256  Mode  CTR :

0x6B 0xC1 0xBE 0xE2 0x2E 0x40 0x9F 0x96 0xE9 0x3D 0x7E 0x11 0x73 0x93 0x17 0x2A   [Block 0]
0xAE 0x2D 0x8A 0x57 0x1E 0x03 0xAC 0x9C 0x9E 0xB7 0x6F 0xAC 0x45 0xAF 0x8E 0x51   [Block 1]
0x30 0xC8 0x1C 0x46 0xA3 0x5C 0xE4 0x11 0xE5 0xFB 0xC1 0x19 0x1A 0x0A 0x52 0xEF   [Block 2]
0xF6 0x9F 0x24 0x45 0xDF 0x4F 0x9B 0x17 0xAD 0x2B 0x41 0x7B 0xE6 0x6C 0x37 0x10   [Block 3]

 Example restarted...
```

And then restart for users to select a different algorithm and mode to repeat demo, as shown below.

```
Plain data :
0x6B 0xC1 0xBE 0xE2 0x2E 0x40 0x9F 0x96 0xE9 0x3D 0x7E 0x11 0x73 0x93 0x17 0x2A   [Block 0]
0xAE 0x2D 0x8A 0x57 0x1E 0x03 0xAC 0x9C 0x9E 0xB7 0x6F 0xAC 0x45 0xAF 0x8E 0x51   [Block 1]
0x30 0xC8 0x1C 0x46 0xA3 0x5C 0xE4 0x11 0xE5 0xFB 0xC1 0x19 0x1A 0x0A 0x52 0xEF   [Block 2]
0xF6 0x9F 0x24 0x45 0xDF 0x4F 0x9B 0x17 0xAD 0x2B 0x41 0x7B 0xE6 0x6C 0x37 0x10   [Block 3]
========Choose CAU algorithm=======
1: DES algorithm
2: TDES algorithm
3: AES algorithm
```

5.16. HAU

5.16.1. DEMO Purpose

This demo includes the following functions of GD32 MCU:

- Learn SHA-1, SHA-224, SHA-256 and MD5 algorithm
- Learn HASH mode and HMAC (keyed-hash message authentication code) mode
- Learn to use HAU to calculate digest for the input message
- Learn to communicate with PC by USART

5.16.2. DEMO Running Result

Jump the JP5 to USART with the jumper cap, and download the program <16_HAU> to the EVAL board and run. Connect serial cable to COM0. When the program is running, the serial terminal tool will display the information, as shown in the following figure. After the user setting the algorithm and mode according to the serial output information indicating, serial port will print out selected algorithm and mode, as shown below.

```
message to be hashed:

The GD32 F2 series is the result of a perfect symbiosis of the real-time control
capabilities of an MCU and the signal processing performance of a DSP, and thus
complements the GD32 portfolio with a new class of devices, digital signal
controllers (DSC).

========Choose HAU algorithm=======
1: SHA1 algorithm
2: SHA224 algorithm
3: SHA256 algorithm
4: MD5 algorithm

You choose to use SHA1 algorithm


========Choose HAU mode=======
1: HASH mode
2: HMAC mode

Choose error: please choose again!
You choose to use HASH mode


message digest with SHA-1  Mode  HASH (160 bits):
```

After selection, the program starts digest calculation, the results are printed through the serial port. And then restart for users to select a different algorithm and mode to repeat demo, as shown below.

```
message digest with SHA-1  Mode  HASH (160 bits):

0x74 0x91 0x90 0xEA
0xEC 0x35 0x11 0xF6
0x04 0xA2 0xDC 0x76
0x58 0x13 0x2A 0x09
0x8A 0x87 0x70 0xCC

 Example restarted...
message to be hashed:

The GD32 F2 series is the result of a perfect symbiosis of the real-time control
capabilities of an MCU and the signal processing performance of a DSP, and thus
complements the GD32 portfolio with a new class of devices, digital signal
controllers (DSC).

=======Choose HAU algorithm=======
1: SHA1 algorithm
2: SHA224 algorithm
3: SHA256 algorithm
4: MD5 algorithm

Choose error: please choose again!
```

5.17. Tamper_Detection

5.17.1. DEMO Purpose

This demo includes the following functions of GD32 MCU:

- Learn BKP tamper function

5.17.2. DEMO Running Result

Download the program <17_Tamper_Detection> to the EVAL board and run. It writes the data to all backup data registers, then check whether the data were correctly written. If data written correctly, LED2 is on, otherwise LED3 is on. When the Tamper key (TAMPER0 pin) is pressed, the backup data registers are reset and the tamper0 interrupt is generated. In the corresponding ISR, it checks whether the backup data registers are cleared or not. If backup data registers are cleared, LED4 is on, otherwise LED5 is on.

5.18. SDIO_SD CardTest

5.18.1. DEMO Purpose

This demo includes the following functions of GD32 MCU:

- Learn to use SDIO to single block or multiple block write and read
- Learn to use SDIO to erase, lock and unlock a SD card

EVAL board has a secure digital input/output interface (SDIO) which defines the SD/SD I/O /MMC CE-ATA card host interface. This demo will show how to use SDIO to operate on SD

card.

5.18.2. DEMO Running Result

Jump the JP5 to USART1 to show the print message through HyperTerminal, and download the program <18_SDIO_SDCardTest> to the EVAL board and run. Connect serial cable to EVAL_COM0, and open the HyperTerminal. Firstly, all the LEDs flash once for test. Then initialize the card and print out the information of the card. After that, test the function of single block operation, lock and unlock operation, erase operation and multiple blocks operation. If any error occurs, print the error message and turn on LED2, LED4 and turn off LED3 and LED5. Otherwise, turn on all the LEDs.

Uncomment the macro DATA_PRINT to print out the data and display them through HyperTerminal. Set bus mode (1-bit or 4-bit) and data transfer mode (polling mode or DMA mode) by comment and uncomment the related statements.

Information via a serial port output as following.

```
Card init success!

Card information:
## Card version 3.0x ##
## SDHC card ##
## Device size is 15558144KB ##
## Block size is 512B ##
## Block count is 31116288 ##
## CardCommandClasses is: 5b5 ##
## Block operation supported ##
## Erase supported ##
## Lock unlock supported ##
## Application specific supported ##
## Switch function supported ##


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

5.19. CAN_Network

5.19.1. DEMO Purpose

This demo includes the following functions of GD32 MCU:

- Learn to use the CAN0 communication between two boards

GD32207C-EVAL development board integrates the CAN(Controller Area Network) bus controller, which is a common industrial control bus. CAN bus controller follows the CAN bus protocol of 2.0 A and 2.0 B. This demo mainly shows how to communicate two EVAL boards through CAN0.

5.19.2. DEMO Running Result

This example is tested with two GD32F207C-EVAL boards. Jump the JP5 to USART and P2, P3 to CAN with the jumper cap. Connect L pin to L pin and H pin to H pin of JP14 on the boards for sending and receiving frames. Download the program <19_CAN_Network> to the two EVAL boards, and connect serial cable to COM0. Firstly, the COM0 sends “please press the Tamper key to transmit data!” to the HyperTerminal. The frames are sent and the transmit data are printed by pressing Tamper Key push button. When the frames are received, the receive data will be printed and the LED2 will toggle one time.

The output information via the serial port is as following.

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

5.20. RCU_Clock_Out

5.20.1. 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

5.20.2. DEMO Running Result

Jump the JP5 to USART1 with the jumper cap, and download the program <20_RCU_Clock_Out> to the EVAL board and run. Connect serial cable to EVAL_COM0, 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 TAMPER 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 tamper key to select clock output source
CK_OUT0: system clock
CK_OUT0: IRC8M
CK_OUT0: HXTAL
CK_OUT0: system clock
```

5.21. PMU_sleep_wakeup

5.21.1. DEMO Purpose

This demo includes the following functions of GD32 MCU:

- Learn to use PMU deepsleep function
- Learn to use the EXTI interrupt to wakeup the MCU in the deepsleep mode

5.21.2. DEMO Running Result

Download the program <21_PMU_sleep_wakeup > to the EVAL board and run. It shows PMU how to enter deepsleep mode and wakeup it. Press Wakeup key to enter deepsleep mode, led stop flashing. When you press Tamper key to generate an exti interrupt, MCU will be wakeuped from deepsleep mode, led sparks again. But the led sparks slower, because at this time IRC8M is the system clock.

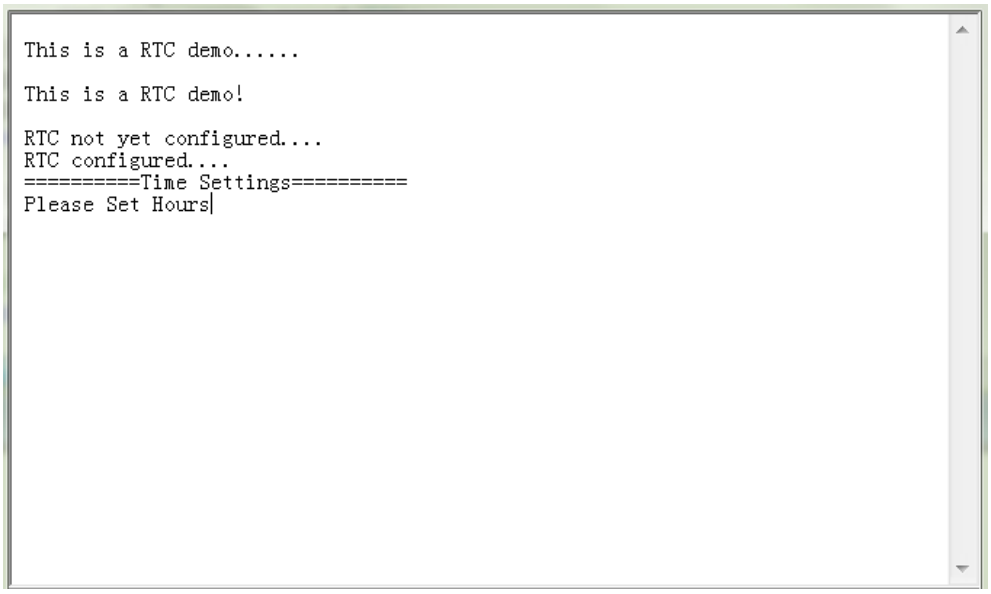
5.22. RTC_Calendar

5.22.1. DEMO Purpose

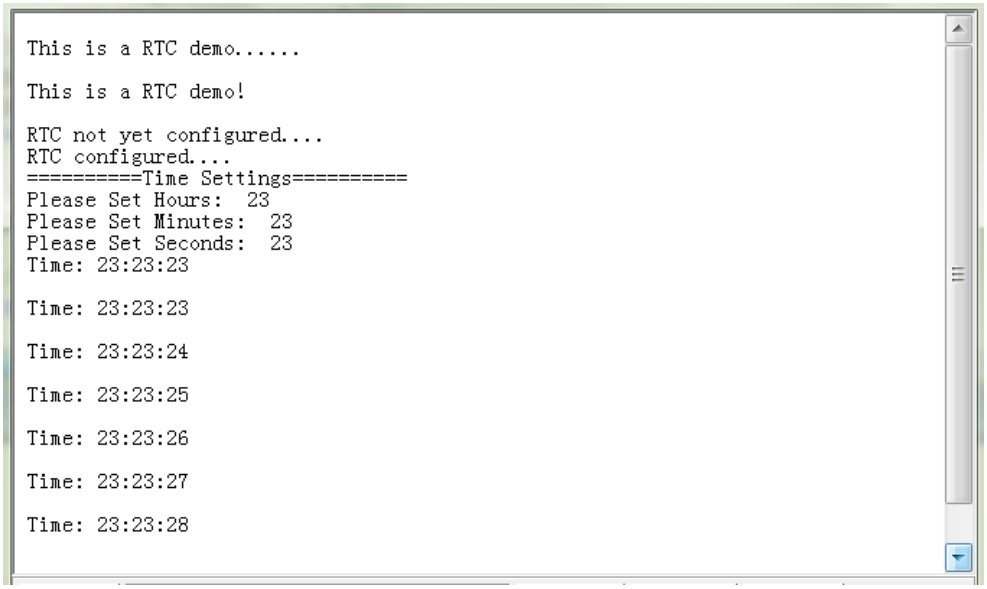
GD32207C-EVAL evaluation board integrated RTC (clock Real-time) real-time clock. If the battery has been installed, the accuracy of the current date and time can be guaranteed when the system is reset or power down RTC is essentially an independent timer, usually used for calendar clocks. This Demo is used to demonstrate the function and usage of the RTC module in the GD32207C-EVAL evaluation board.

5.22.2. DEMO Running Result

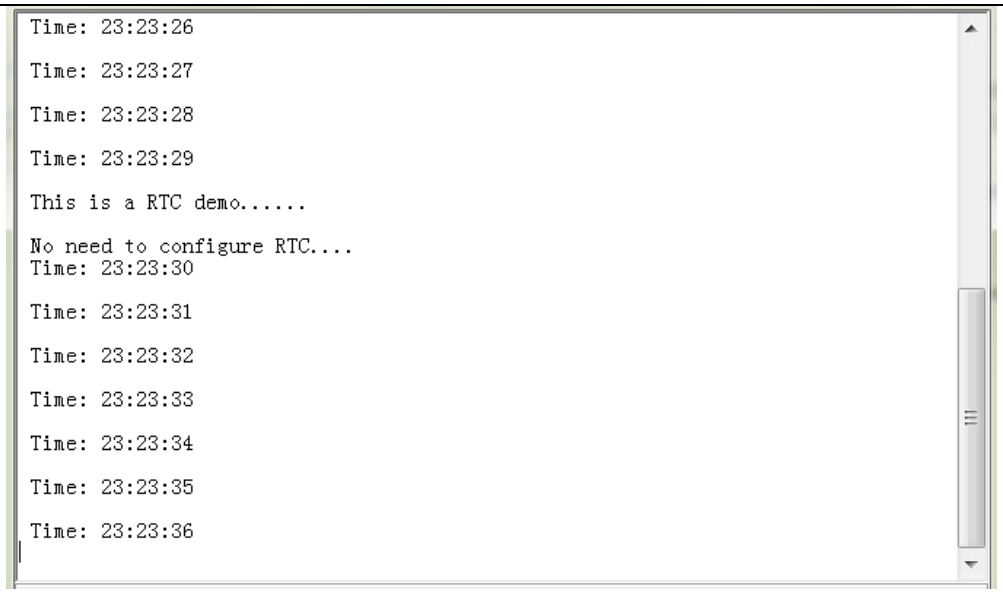
Download the program to the development board, serial port output information, as shown in the following figure. If the development board run the program for the first time, serial port output following information "RTC not yet configured…." It requires the user to set up hours、minutes and seconds.



According to the serial port output information prompt, setting time, serial port will print out the current time every second, as shown below.



If the development board is not the first run of the program, time has been set up in the last run, after the system reset or battery power restart, as shown below, serial port output following information " No need to configured RTC....", serial port continue printing time information.



5.23. TIMER_Breath_LED

5.23.1. DEMO Purpose

This demo includes the following functions of GD32 MCU:

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

5.23.2. DEMO Running Result

Use the DuPont line to connect the TIMER1 CH2 (PA2) and LED2 (PC0), and then download the program <23_TIMER_Breath_LED> to the board and run.

When the program is running, you can see LED2 lighting from dark to bright gradually and then gradually darken, ad infinitum, just like breathing as rhythm.

5.24. TLI_without_GUI

5.24.1. DEMO Purpose

This demo includes the following functions of GD32 MCU:

- Learn to use TLI to control LCD for displaying different images

5.24.2. DEMO Running Result

Jump the JP12, JP13, JP18, JP19, JP24, JP25, JP26, JP27 to LCD. Select the corresponding

macro (USE_LCD_VERSION_x_y) according to the specific version number of LCD. Download the program <24_TLI_without_GUI> to the EVAL board and run. After downloading program to board, a running cheetah on the background of GD logo is appeared on the LCD, which outputs as following.

5.25. ENET

5.25.1. FreeRTOS_tcpudp

DEMO Purpose

This demo includes the following functions of GD32 MCU:

- Learn to use Lwip stack
- Learn to use FreeRTOS operation system
- Learn to use netconn and socket API to handle with a task
- Learn how to realize a tcp server
- Learn how to realize a tcp client
- Learn how to realize a udp server/client
- Learn how to use DHCP to allocate ip address automatically

This demo is based on the GD32207C-EVAL evaluation board, it shows how to configure the enet peripherals to send and receive frames in normal mode and use lwip tcp/ip stack to realize ping, telnet and server/client functions.

JP4, JP13, JP18 must be fitted.

It is configured in RMII mode, and 25MHz oscillator is used, the system clock is configured to 120MHz.

This demo realizes three applications:

1) Telnet application, the eval board acts as tcp server. Users can link the client with the eval board server, using 8000 port. Users can see the reply from the server, and can send the name(should input enter key) to server.

2) tcp client application, the eval board acts as tcp client. Users can link the eval board client with the server, using 10260 port. Users can send information from server to client, then the client will send back the information.

3) udp application. Users can link the eval board with other station, using 1025 port. Users can send information to board from station, then the board will send back the information.

If users need dhcp function, it can be configured from the private defines in main.h. This function is closed by default.

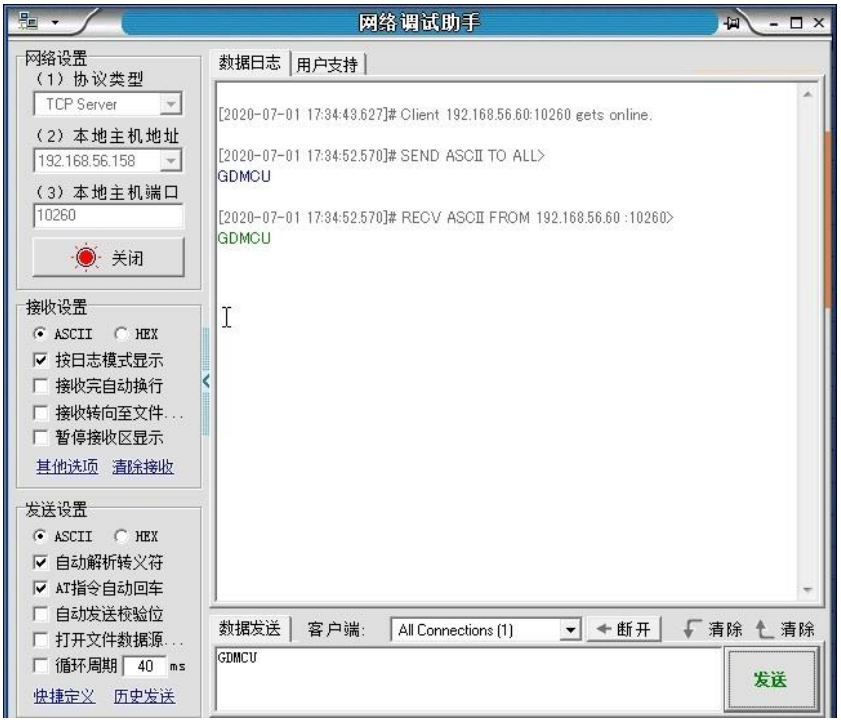
Note: Users should configure ip address, mask and gw of GD32207C-EVAL evaluation board or served according to the actual net situation from the private defines in main.h.

DEMO Running Result

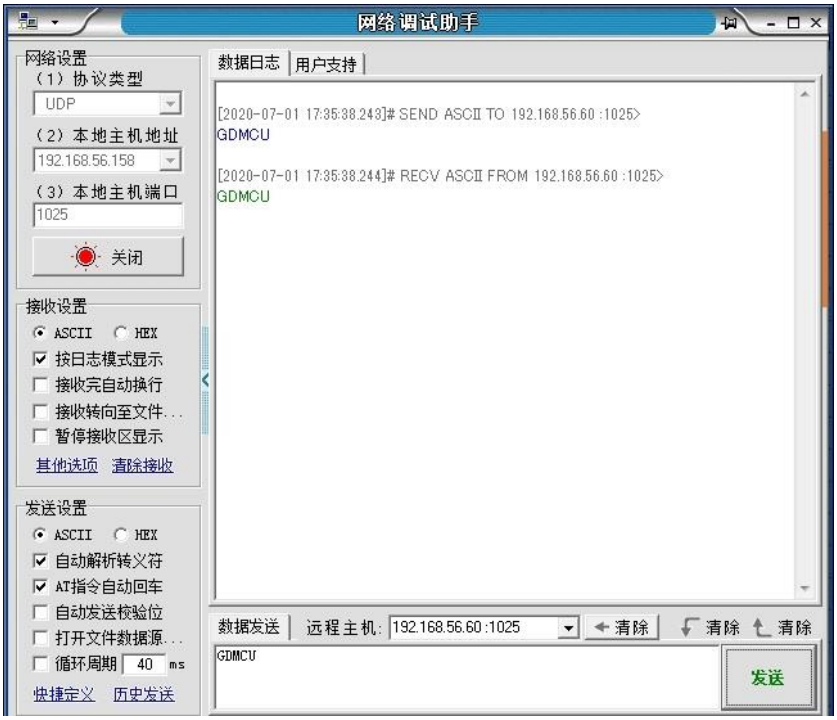
Download the program <FreeRTOS_tcpudp> to the EVAL board, LED3 will light every 500ms.

Using Network assistant software, configure the pc side to tcp client, using 8000 port, and when send something through the assistant, users can see the reply from the server:

Using Network assistant software, configure the pc side to tcp server, using 10260 port, and when send something through the assistant, users can see the echo reply from the client:



Using Network assistant software, configure to use udp protocol, using 1025 port, and when send something through the assistant, users can see the echo reply from the board:



Open the DHCP function in main.h, using a router to connect the board with the pc, users can see the automatic allocated ip address of the board from the HyperTerminal.

5.25.2. Raw_tcpudp

DEMO Purpose

This demo includes the following functions of GD32 MCU:

- Learn to use Lwip stack
- Learn to use raw API to handle with a task
- Learn how to realize a tcp server
- Learn how to realize a tcp client
- Learn how to realize a udp server/client
- Learn how to use DHCP to allocate ip address automatically
- Learn to handle with received packet in polling mode and in interrupt mode

This demo is based on the GD32207C-EVAL evaluation board, it shows how to configure the enet peripherals to send and receive frames in normal mode and use lwip tcp/ip stack to realize ping, telnet and server/client functions.

JP4, JP13, JP18 must be fitted. JP5 jump to USART0

It is configured in RMII mode, and 25MHz oscillator is used, the system clock is configured to 120MHz.

This demo realizes three applications:

1) Telnet application, the eval board acts as tcp server. Users can link the client with the eval board server, using 8000 port. Users can see the reply from the server, and can send the name(should input enter key) to server.

2) tcp client application, the eval board acts as tcp client. Users can link the eval board client with the server, using 10260 port. Users can send information to client from server, then the client will send back the information. If the server is not online at first, or is break during process, when the server is ready again, users can press tamper key to reconnect with server, and communicate.

3) udp application, Users can link the eval board with other station, using 1025 port. Users can send information from station to board, then the board will send back the information.

By default, the packet reception is polled in while(1). If users want to receive packet in interrupt service, uncomment the macro defined USE_ENET_INTERRUPT in main.h.

If users need dhcp function, it can be configured from the private defines in main.h. This function is closed in default.

Note: Users should configure ip address, mask and gw of GD32207C-EVAL evaluation board, or server according to the actual net situation from the private defines in main.h.

DEMO Running Result

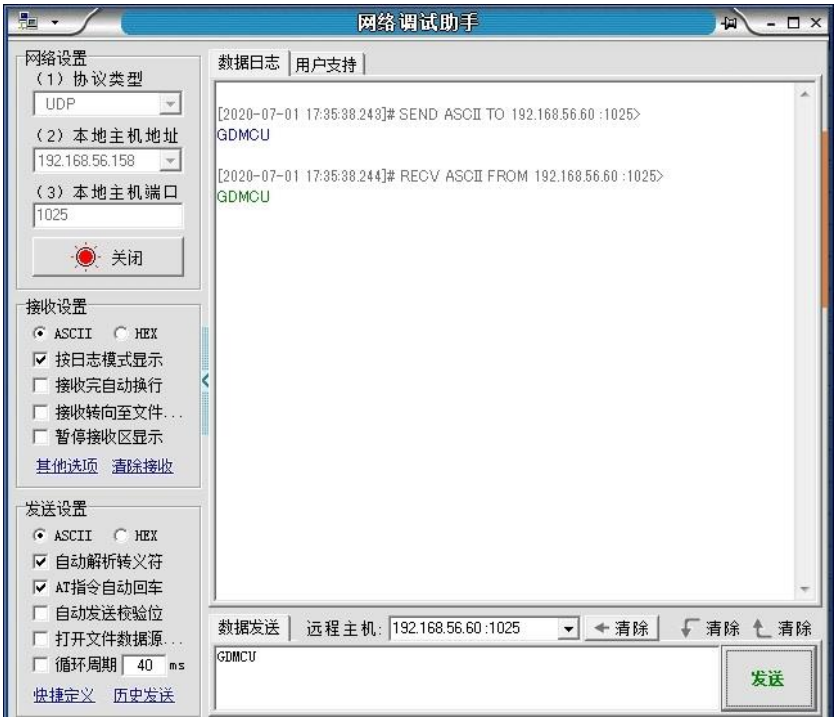
Download the program <Raw_tcpudp> to the EVAL board.

Using Network assistant software, configure the pc side to tcp client, using 8000 port, and when send something through the assistant, users can see the reply from the server:

Using Network assistant software, configure the pc side to tcp server, using 10260 port, press

the Tamper key, and when send something through the assistant, users can see the echo reply from the client:

Using Network assistant software, configure to use udp protocol, using 1025 port, and when send something through the assistant, users can see the echo reply from the board:



Open the DHCP function in main.h, using a router to connect the board with the pc, users can see the automatic allocated ip address of the board from the HyperTerminal.

5.25.3. Raw_webserver

DEMO Purpose

This demo includes the following functions of GD32 MCU:

- Learn to use Lwip stack
- Learn to use raw API to handle with a task
- Learn how to realize a web server
- Learn how to use a web server to control LEDs
- Learn how to use a web server to monitor the board V_{REFINT} voltage
- Learn how to use DHCP to allocate ip address automatically
- Learn to handle with received packet in polling mode and in interrupt mode

This demo is based on the GD32207C-EVAL evaluation board, it shows how to configure the enet peripherals to send and receive frames in normal mode and use lwip tcp/ip stack to realize webserver application.

JP4, JP13, JP18 must be fitted. JP5 jump to USART0.

It is configured in RMII mode, and 25MHz oscillator is used, the system clock is configured to 120MHz.

This demo realizes webserver application:

Users can visit the eval board through Internet Explorer, the eval board acts as a webserver, and the url is the local ip address of the eval board. There are two experiments realized, one is the LEDs control, the other one is the ADC monitoring VREFINT voltage in real-time.

If users need dhcp function, it can be configured from the private defines in main.h. This function is closed by default. Users can use a router to connect the eval board, and use the COM port to print the automatic allocated ip address, then connect your mobile phone to the wifi which the router send. Users can visit the eval board and control it on your mobile phone.

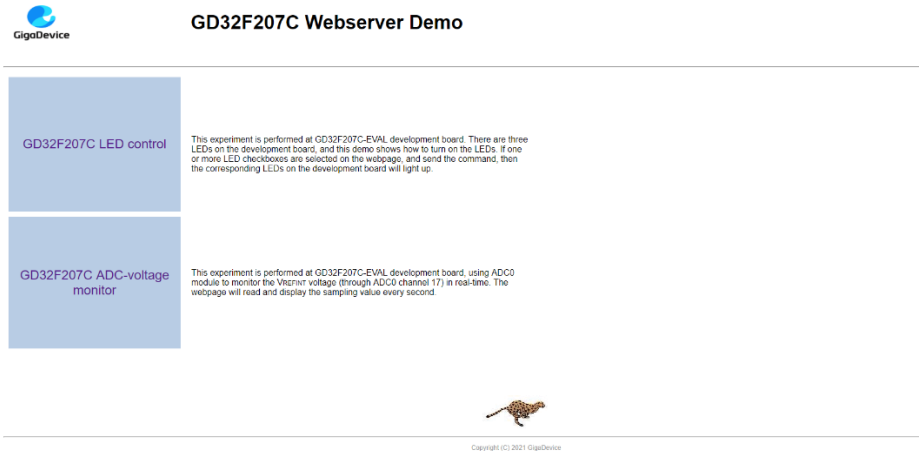
By default, the packet reception is polled in while(1). If users want to receive packet in interrupt service, uncomment the macro define USE_ENET_INTERRUPT in main.h.

Note: Users should configure ip address, mask and gw of GD32207C-EVAL evaluation board according to the actual net situation from the private defines in main.h.

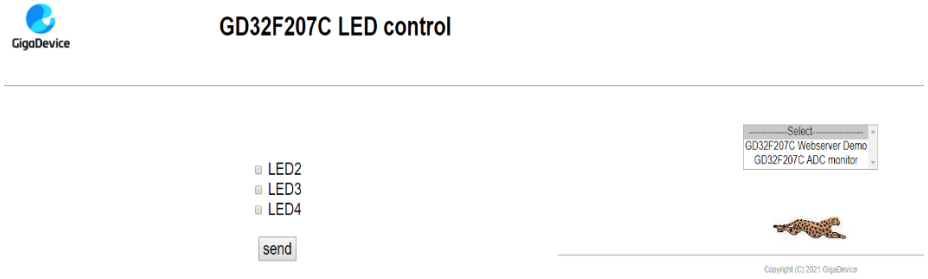
DEMO Running Result

Download the program <Raw_webserver> to the EVAL board, using Internet Explorer software, enter in the ip address of the board, click on the LED control linker, choose the LED checkboxes users want to light, and “send”, the corresponding LEDs will light. Click on the ADC monitor linker, the real-time VREFINT voltage is showed on the webpage, and the data refreshes every second automatically.

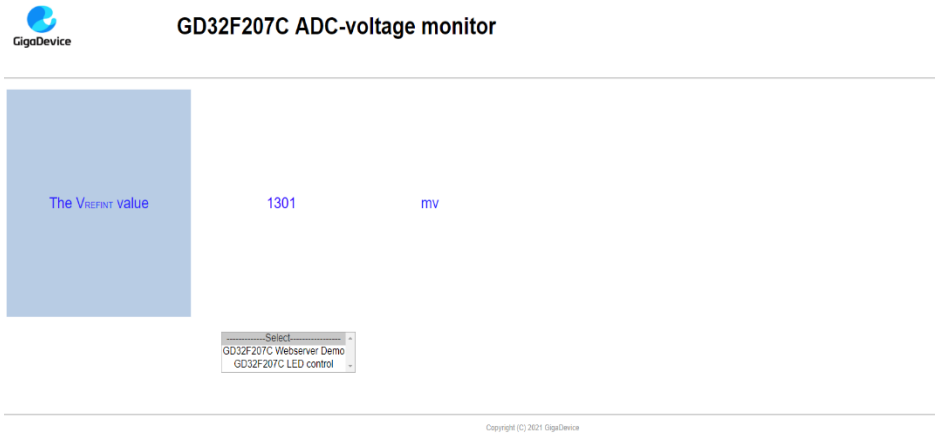
The web home page shows as below:



The LED control page shows as below:



The ADC monitor page shows as below:



Open the DHCP function in main.h, using a router to connect the board, and use the HyperTerminal to print the automatic allocated ip address, then connect your mobile phone to the wifi which the router send. Users can visit the eval board and control it on your mobile phone.

5.26. USB_Device

5.26.1. HID_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

GD32207C-EVAL board has four keys and one USB_FS interface. The four keys are Reset key, Wakeup key, Tamper key, User key. In this demo, the GD32207C-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 JP25, JP26 jump to USB. After doing this, download the program <26_USBFS\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.

5.26.2. MSC_Udisk

DEMO Purpose

This demo includes the following functions of GD32 MCU:

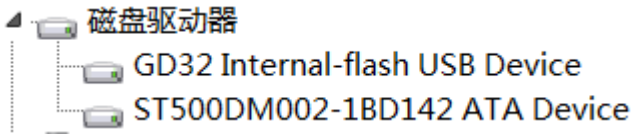
- Learn how to use the USB_FS peripheral mode
- Learn how to implement USB MSC(mass storage) device

This demo mainly implements a U disk. U disk is currently very widely used removable MSC devices. MSC, the Mass Storage device Class, is a transport protocol between a computer and mobile devices, which allow a universal serial bus (USB) equipment to access a host computing device, file transfer between them, mainly including mobile hard disk, mobile U disk drive, etc... The MSC device must have a storage medium, and this Demo uses the MCU's internal SRAM as the storage medium. For more details of the MSC protocol please refer to the MSC protocol standard.

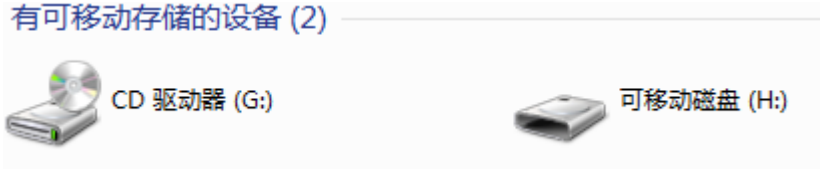
MSC device will use a variety of transport protocols and command formats for communication, so it need to choose the appropriate protocol and command format in the realization of the application. This Demo selects the BOT (bulk only transport) protocol and the required SCSI (small computer interface) command, and is compatible with a wide variety of Window operating systems. Specific BOT protocol and SCSI command specification please refer to the standard of their agreement.

DEMO Running Result

Before running the demo, please ensure that jumper JP25, JP26 jump to USB. After doing this, download the program <26_USBFS\USB_Device\MSC_Udisk> to the EVAL board and run. When the EV-board connect to the PC, you will find a USB large capacity storage device is in the universal serial bus controller, and there is new one disk drive in the equipment manager of PC, as shown below:



Then, after opening the resource manager, you will see more of the 1 disk, as shown in the following diagram:



At this point, the write/read/formatting operation can be performed as the other mobile devices.

5.27. USB_Host

5.27.1. 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

GD32207C-EVAL evaluation board integrates the USBFS module, and the module can be used as a USB device, a USB host or an 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 JP25 and the JP26 to USB. Then download the program <26_USBFS\USB_Host\Host_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 movement of mouse in the hyperterminal.

```
##### USB Host library started #####
> Device Attached.
> Reset the USB device.
> Low speed device detected.
> VID: 046Dh
> PID: C077h
> HID device connected.
> Manufacture string is : Logitech
> Product string is : USB Optical Mouse
> Serial Number string is : N/A
> Enumeration completed.
> To start the HID class operations:
> Press User Key...
> Wait for user input!
> User has input!
> HID Demo Device : Mouse.
MoveRight 32 units---*---MoveUp 7f units---*---No button is pressed.
MoveRight 0 units---*---MoveUp 14 units---*---No button is pressed.
MoveRight 1 units---*---MoveUp 1 units---*---No button is pressed.
MoveRight 3 units---*---MoveDown 0 units---*---No button is pressed.
MoveRight 1 units---*---MoveDown 0 units---*---No button is pressed.
MoveRight 2 units---*---MoveDown 0 units---*---No button is pressed.
MoveRight 5 units---*---MoveDown 0 units---*---No button is pressed.
MoveRight 6 units---*---MoveDown 1 units---*---No button is pressed.
MoveRight 9 units---*---MoveDown 2 units---*---No button is pressed.
MoveRight 9 units---*---MoveDown 1 units---*---No button is pressed.
MoveRight 8 units---*---MoveDown 1 units---*---No button is pressed.
MoveRight 6 units---*---MoveDown 1 units---*---No button is pressed.
```

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, the input will be printed in the hyperterminal.

5.27.2. 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

GD32207C-EVAL evaluation board integrates the USBFS module, and the module can be used as a USB device, a USB host or an 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 JP25 and the JP26 to USB. Then insert the OTG cable to the USB port, download the program <26_USBFS\USB_Host\Host_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	Initialize version	Jul. 15th, 2015
2.0	Update version	Jun. 5th, 2017
2.1	Update EVAL board	Oct. 31st, 2018
2.2	Rebase version	Oct. 31st, 2021

Important Notice

This document is the property of GigaDevice Semiconductor Inc. and its subsidiaries (the "Company"). This document, including any product of the Company described in this document (the “Product”), is owned by the Company under the intellectual property laws and treaties of the People’s Republic of China and other jurisdictions worldwide. The Company reserves all rights under such laws and treaties and does not grant any license under its patents, copyrights, trademarks, or other intellectual property rights. The names and brands of third party referred thereto (if any) are the property of their respective owner and referred to for identification purposes only.

The Company makes no warranty of any kind, express or implied, with regard to this document or any Product, including, but not limited to, the implied warranties of merchantability and fitness for a particular purpose. The Company does not assume any liability arising out of the application or use of any Product described in this document. Any information provided in this document is provided only for reference purposes. It is the responsibility of the user of this document to properly design, program, and test the functionality and safety of any application made of this information and any resulting product. Except for customized products which has been expressly identified in the applicable agreement, the Products are designed, developed, and/or manufactured for ordinary business, industrial, personal, and/or household applications only. The Products are not designed, intended, or authorized for use as components in systems designed or intended for the operation of weapons, weapons systems, nuclear installations, atomic energy control instruments, combustion control instruments, airplane or spaceship instruments, transportation instruments, traffic signal instruments, life-support devices or systems, other medical devices or systems (including resuscitation equipment and surgical implants), pollution control or hazardous substances management, or other uses where the failure of the device or Product could cause personal injury, death, property or environmental damage ("Unintended Uses"). Customers shall take any and all actions to ensure using and selling the Products in accordance with the applicable laws and regulations. The Company is not liable, in whole or in part, and customers shall and hereby do release the Company as well as it’s suppliers and/or distributors from any claim, damage, or other liability arising from or related to all Unintended Uses of the Products. Customers shall indemnify and hold the Company as well as it’s suppliers and/or distributors harmless from and against all claims, costs, damages, and other liabilities, including claims for personal injury or death, arising from or related to any Unintended Uses of the Products.

Information in this document is provided solely in connection with the Products. The Company reserves the right to make changes, corrections, modifications or improvements to this document and Products and services described herein at any time, without notice.