GigaDevice Semiconductor Inc.

GD32450Z-EVAL

User Guide
V2.0

Table of Contents

List of Figures

List of Tables

1. Summary

GD32450Z-EVAL uses GD32F450ZKT6 as the main controller. It uses Mini USB interface or DC-005 connector to supply 5V power. SWD, Reset, Boot, User button key, LED, CAN, I2C, I2S, USART, RTC, LCD, SPI, ADC, DAC, EXMC, CTC, SDIO, ENET, USBFS, USBHS, GD-Link and Extension Pins are also included. For more details please refer to GD32450Z-EVAL-V1.1 schematic.

2. Function pin assignment

Table 2-1 Function pin assignment

Function	Pin	Description
LED	PD4	LED1
	PD5	LED2
	PG3	LED3
RESET		K1-Reset
KEY	PA0	K2-Wakeup
	PC13	K3-Tamper
	PB14	K4-User key
USART0	PA9	USART0_TX
	PA10	USART0_RX
ADC	PF6	ADC012_IN4
DAC	PA4	DAC_OUT0
I2C	PB6	I2C0_SCL
	PB7	I2C0_SDA
SPI	PG10	SPI5_IO2
	PG11	SPI5_IO3
	PG13	SPI5_SCK
	PG14	SPI5_MOSI
	PG12	SPI5_MISO
	PG9	SPI5_CS
I2S	PC6	I2S1_MCK
	PC7	I2S1_CK
	PB9	I2S1_WS
	PC1	I2S1_SD
CAN	PD0	CAN0_RX
	PD1	CAN0_TX
SDRAM	PC0	EXMC_SDNWE

	PC2	EXMC_SDNE0
	PC3	EXMC_SDCKE0
	PD0	EXMC_D2
	PD1	EXMC_D3
	PD8	EXMC_D13
	PD9	EXMC_D14
	PD10	EXMC_D15
	PD14	EXMC_D0
	PD15	EXMC_D1
	PE0	EXMC_NBL0
	PE1	EXMC_NBL1
	PE7	EXMC_D4
	PE8	EXMC_D5
	PE9	EXMC_D6
	PE10	EXMC_D7
	PE11	EXMC_D8
	PE12	EXMC_D9
	PE13	EXMC_D10
	PE14	EXMC_D11
	PE15	EXMC_D12
	PF0	EXMC_A0
	PF1	EXMC_A1
	PF2	EXMC_A2
	PF3	EXMC_A3
	PF4	EXMC_A4
	PF5	EXMC_A5
	PF11	EXMC_NRAS
	PF12	EXMC_A6
	PF13	EXMC_A7
	PF14	EXMC_A8
	PF15	EXMC_A9
	PG0	EXMC_A10
	PG1	EXMC_A11
	PG2	EXMC_A12
	PG4	EXMC_A14
	PG5	EXMC_A15
	PG8	EXMC_SDCLK
	PG15	EXMC_NCAS
SDIO	PD2	SDIO_CMD
	PC12	SDIO_CK
	PC8	SDIO_D0

	PC9	SDIO_D1
	PC10	SDIO_D2
	PC11	SDIO_D3
LCD	PC6	TLI_HSYNC
	PA4	TLI_VSYNC
	PG7	TLI_PIXCLK
	PF10	TLI_DE
	PG6	TLI_R7
	PA8	TLI_R6
	PA12	TLI_R5
	PA11	TLI_R4
	PB0	TLI_R3
	PD3	TLI_G7
	PC7	TLI_G6
	PB11	TLI_G5
	PB10	TLI_G4
	PG10	TLI_G3
	PA6	TLI_G2
	PB9	TLI_B7
	PB8	TLI_B6
	PA3	TLI_B5
	PG12	TLI_B4
	PG11	TLI_B3
Ethernet	PA1	ETH_RMII_REF_CLK
	PA2	ETH_MDIO
	PA7	ETH_RMII_CRS_DV
	PB11	ETH_RMII_TX_EN
	PB12	ETH_RMII_TXD0
	PB13	ETH_RMII_TXD1
	PC1	ETH_MDC
	PC4	ETH_RMII_RXD0
	PC5	ETH_RMII_RXD1
USB_FS	PA9	USB_VBUS
	PA11	USB_DM
	PA12	USB_DP
	PD13	USB_VBUS_CTRL
USB_HS	PC3	USB_HS_ULPI_NXT
	PC2	USB_HS_ULPI_DIR
	PC0	USB_HS_ULPI_STP
	PA5	USB_HS_ULPI_CK
	PB5	USB_HS_ULPI_D7

	PB13	USB_HS_ULPI_D6
	PB12	USB_HS_ULPI_D5
	PB11	USB_HS_ULPI_D4
	PB10	USB_HS_ULPI_D3
	PB1	USB_HS_ULPI_D2
	PB0	USB_HS_ULPI_D1
	PA3	USB_HS_ULPI_D0

3. Getting started

The EVAL board uses Mini USB connecter or DC-005 connector to get power DC +5V, which is the hardware system normal work voltage. Three kinds of different USB power supply which are USB_FS, USB_HS_ULPI, GD-Link can be chosen through JP4. A J-Link tool or GD-Link on board is necessary in order to download and debug programs. Select the correct boot mode and then power on, the LED5 will turn on, which indicates that the power supply is OK.

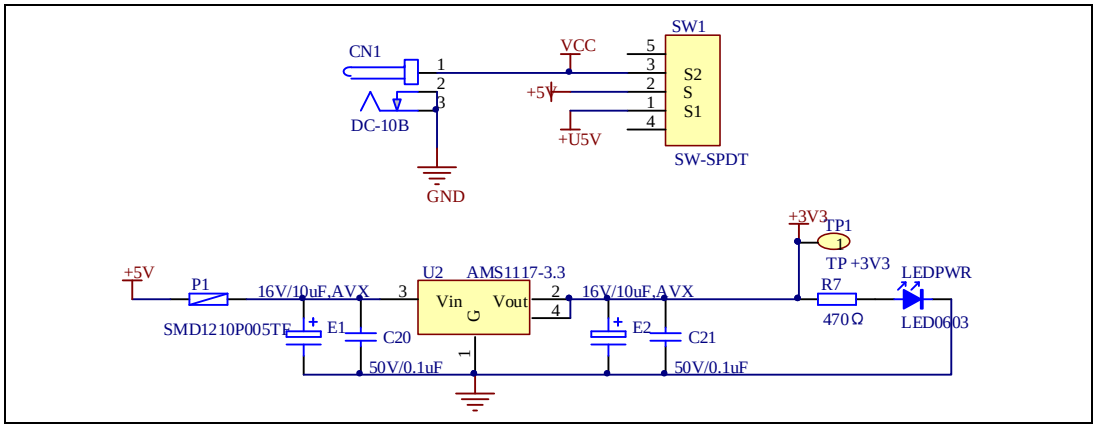
There are Keil version and IAR version of all projects. Keil version of the projects are created based on Keil MDK-ARM 4.74 uVision4. IAR version of the projects are created based on IAR Embedded Workbench for ARM 7.40.2. In Firmware folder, Addon and Software Pack are used to add the devices, peripherals and others to IDE. During use, the following points should be noted:

1. If you use Keil uVision4 to open the project, install the GD32F4xx_Addon.1.0.2.exe which is in \Library\Firmware to load the associated files.
2. If you use Keil uVision5 to open the project, there are two ways to solve the "Device Missing (s)" problem. One is to install GigaDevice.GD32F4xx_DFP.1.0.4.pack which is in \Library\Firmware. In Project menu, select the Manage sub menu, click on the “Version Migrate 5 Format...” menu, the Keil uVision4 project will be converted to Keil uVision5 project. Then add “C:\Keil_v5\ARM\Pack\ARM\CMSIS\4.2.0\CMSIS\Include” to C/C++ in Option for Target. The other is to install Addon directly. Select the installation directory of Keil uVision5 software, such as C:\Keil_v5, in Destination Folder of Folder Selection. Select the corresponding device in Device of Option for Target and add “C:\Keil_v5\ARM\Pack\ARM\CMSIS\4.2.0\CMSIS\Include” to C/C++ in Option for Target.
3. If you use IAR to open the project, install IAR_GD32F4xx_ADDON.1.0.1.exe which is in \Library\Firmware to load the associated files.

4. Hardware layout overview

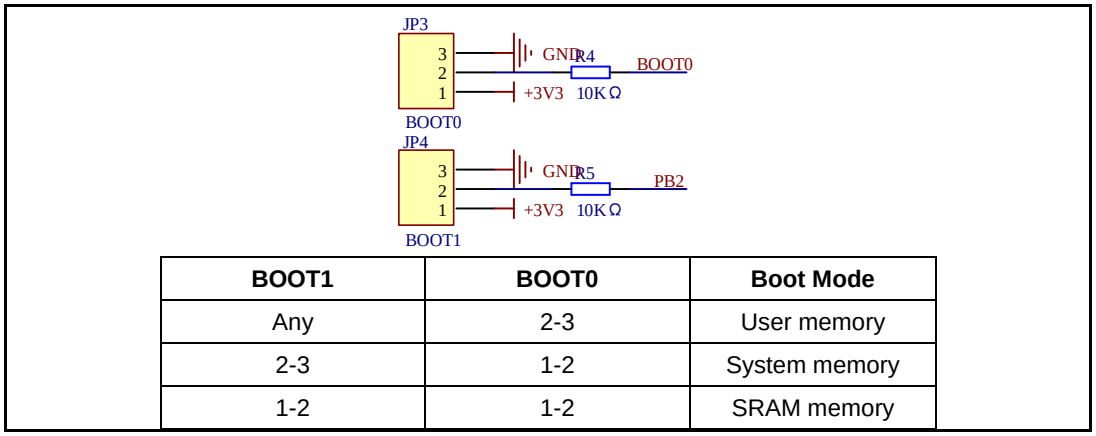
4.1. Power supply

Figure 4-1 Schematic diagram of power supply



4.2. Boot option

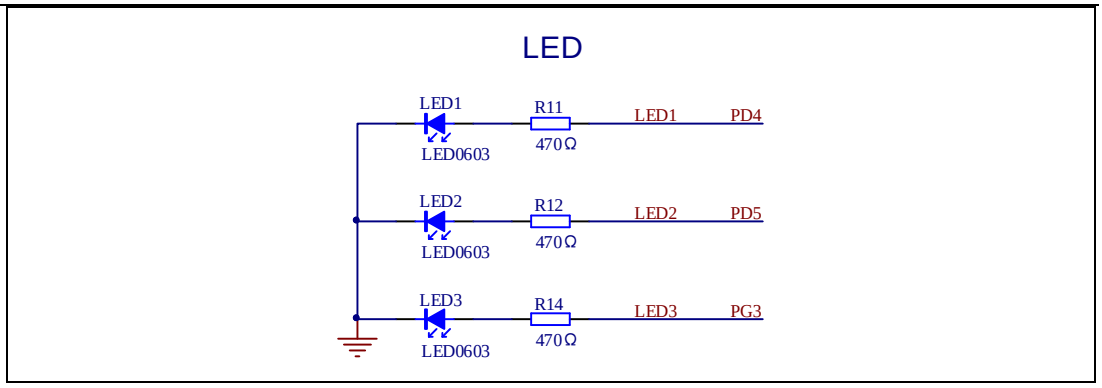
Figure 4-2 Schematic diagram of boot option



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

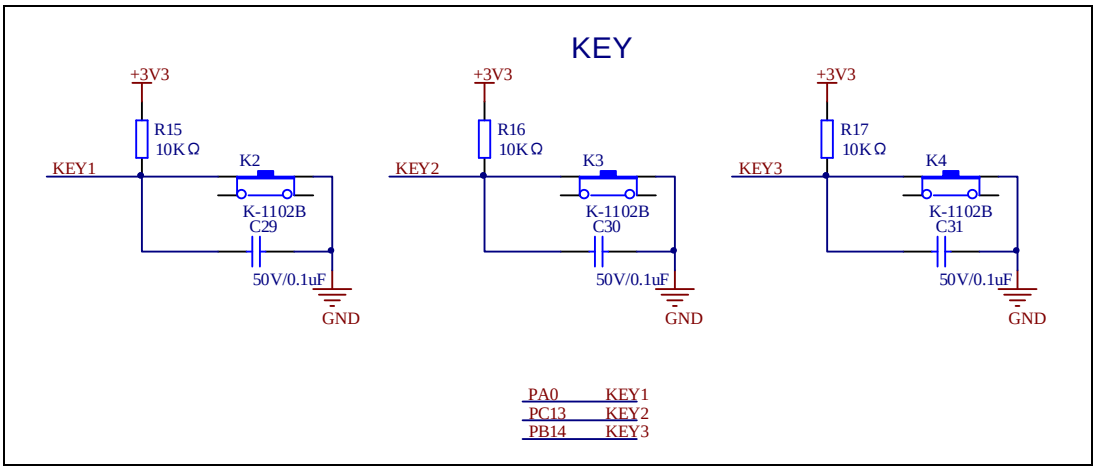
4.3. LED

Figure 4-3 Schematic diagram of LED function



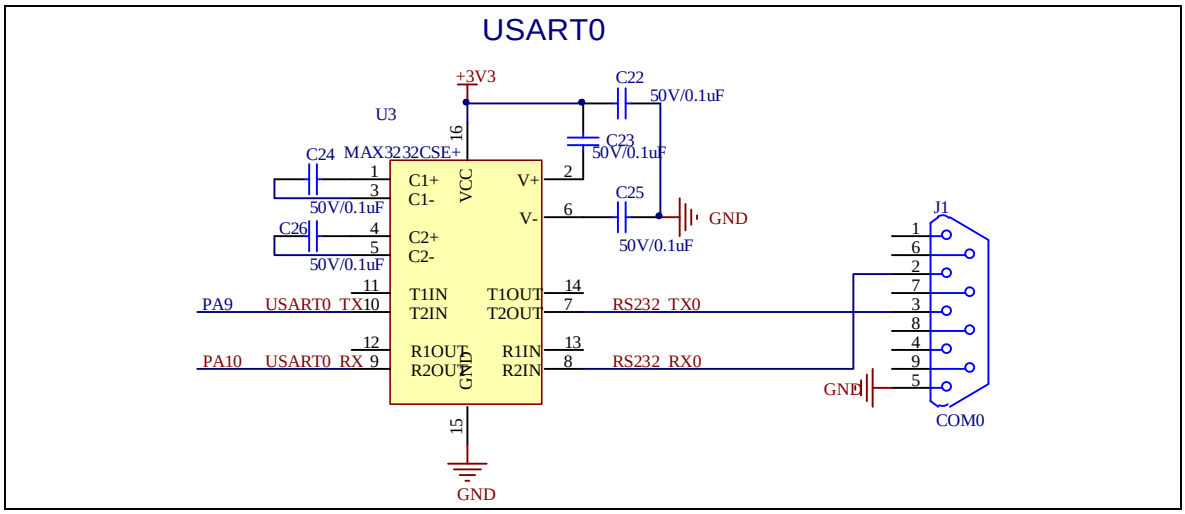
4.4. KEY

Figure 4-4 Schematic diagram of Key function



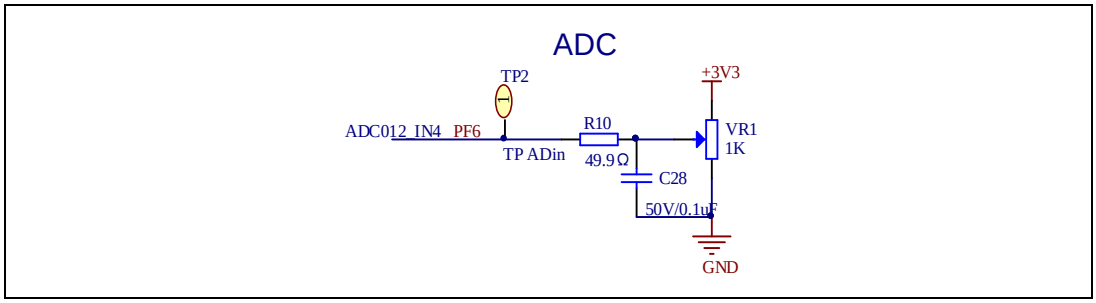
4.5. USART

Figure 4-5 Schematic diagram of USART0 function



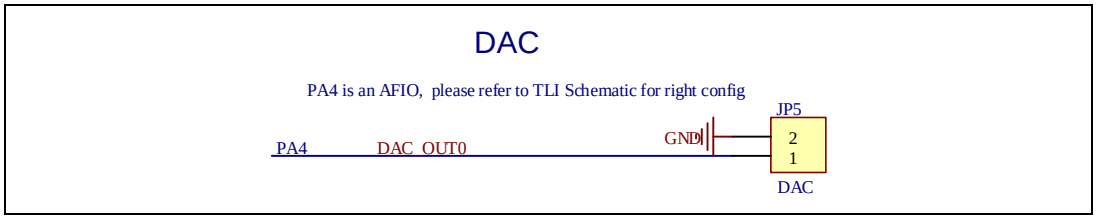
4.6. ADC

Figure 4-6 Schematic diagram of ADC function



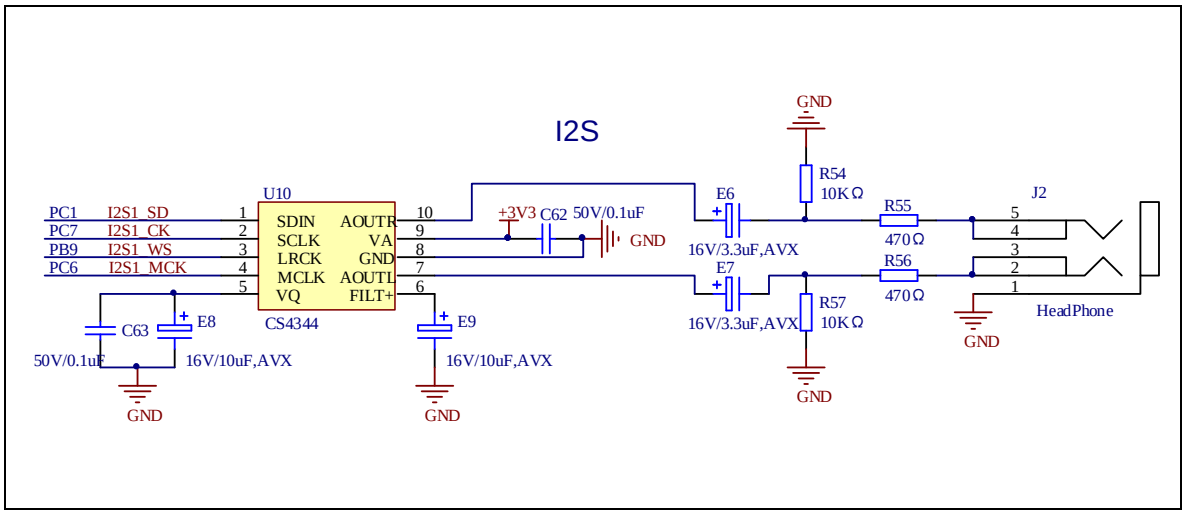
4.7. DAC

Figure 4-7 Schematic diagram of DAC function



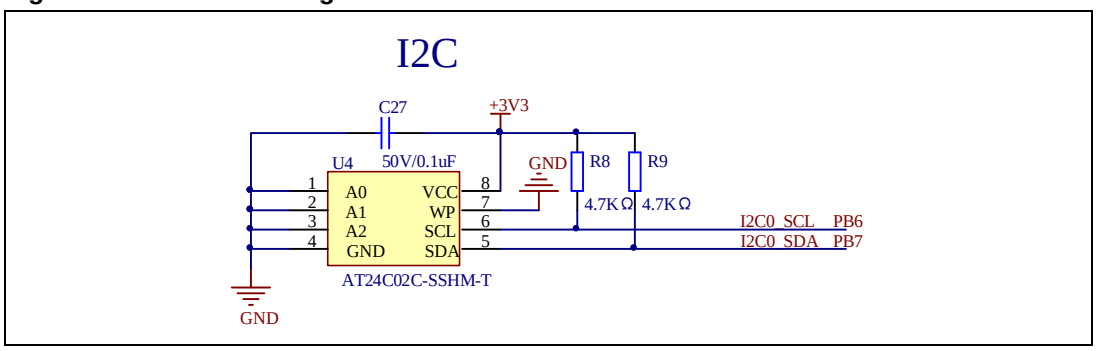
4.8. I2S

Figure 4-8 Schematic diagram of I2S function



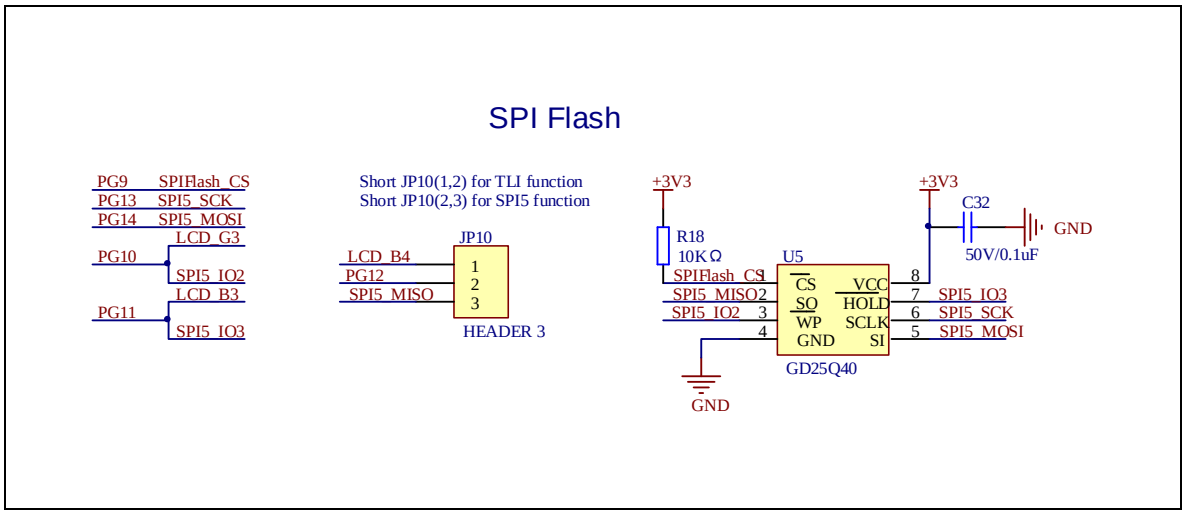
4.9. I2C

Figure 4-9 Schematic diagram of I2C function



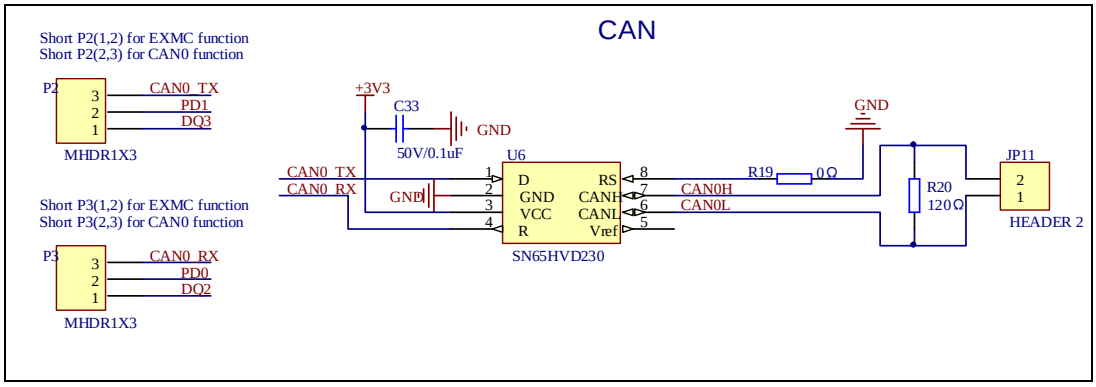
4.10. SPI

Figure 4-10 Schematic diagram of SPI function



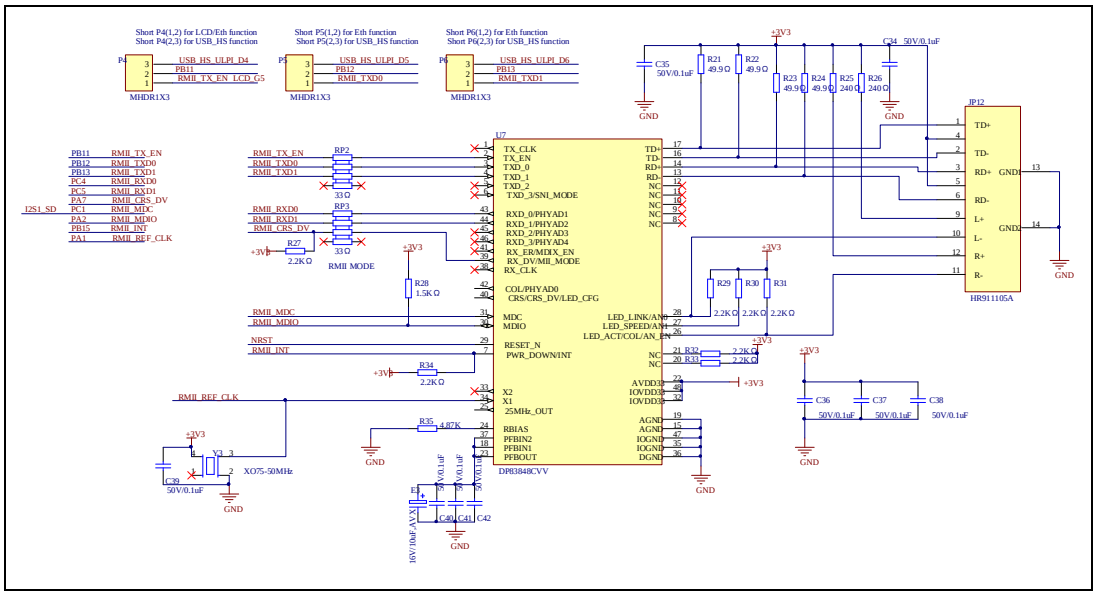
4.11. CAN

Figure 4-11 Schematic diagram of CAN function



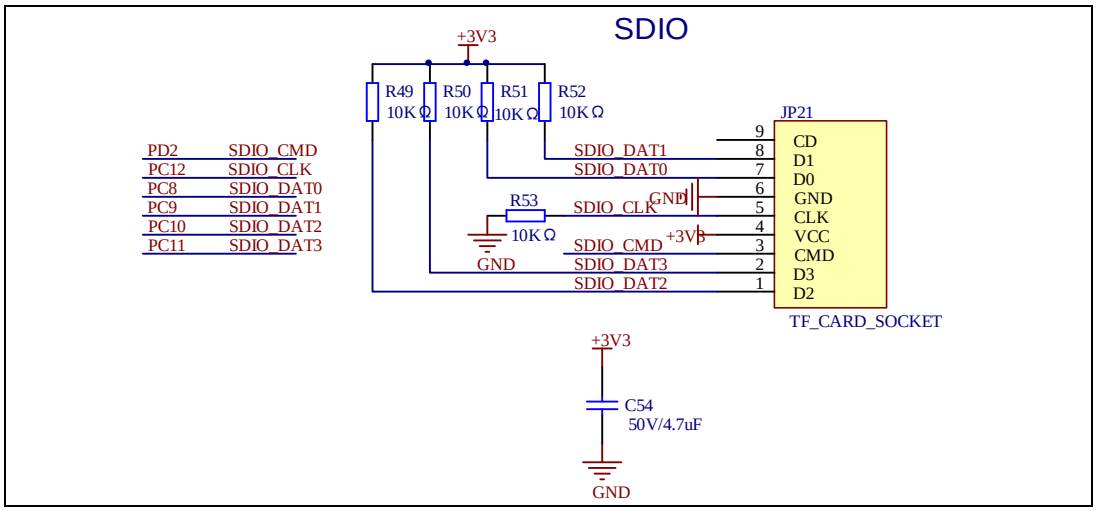
4.12. ENET

Figure 4-12 Schematic diagram of Ethernet function



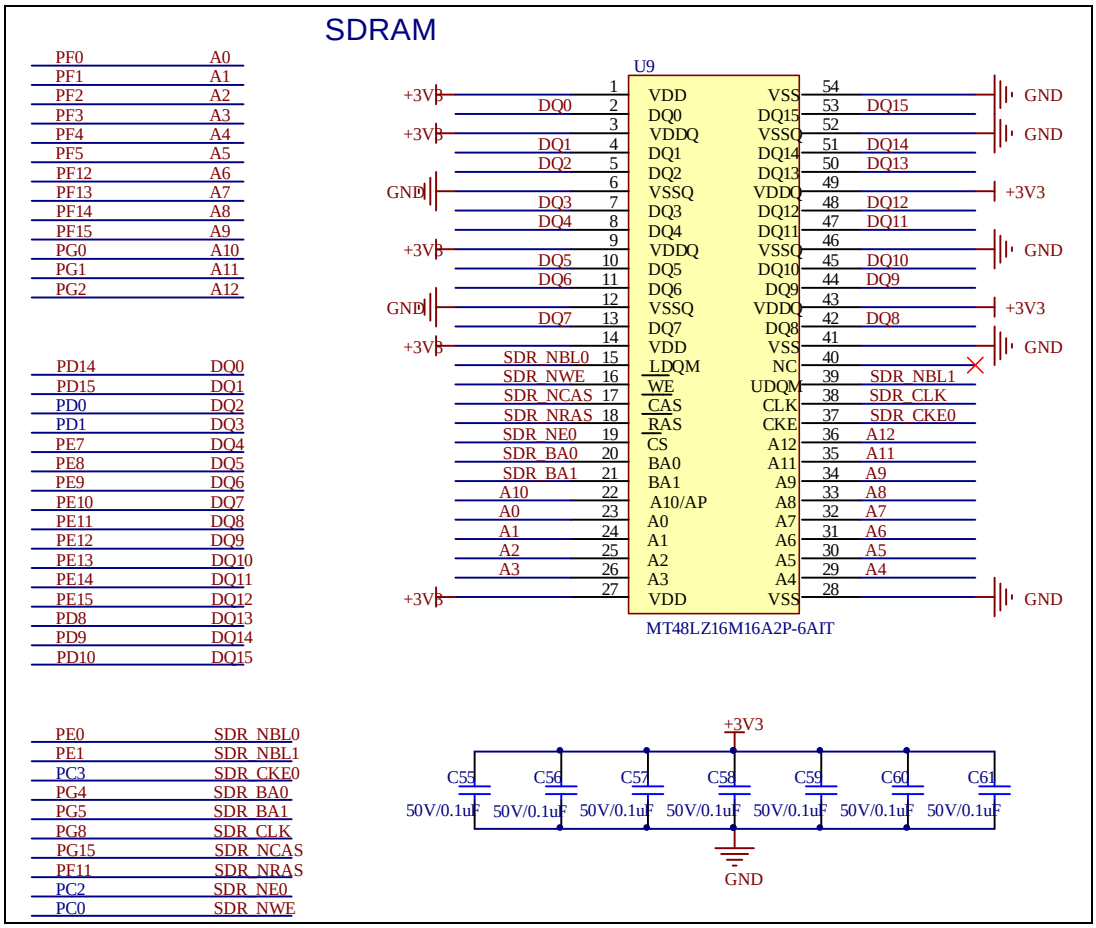
4.13. SDIO

Figure 4-13 Schematic diagram of SDIO function



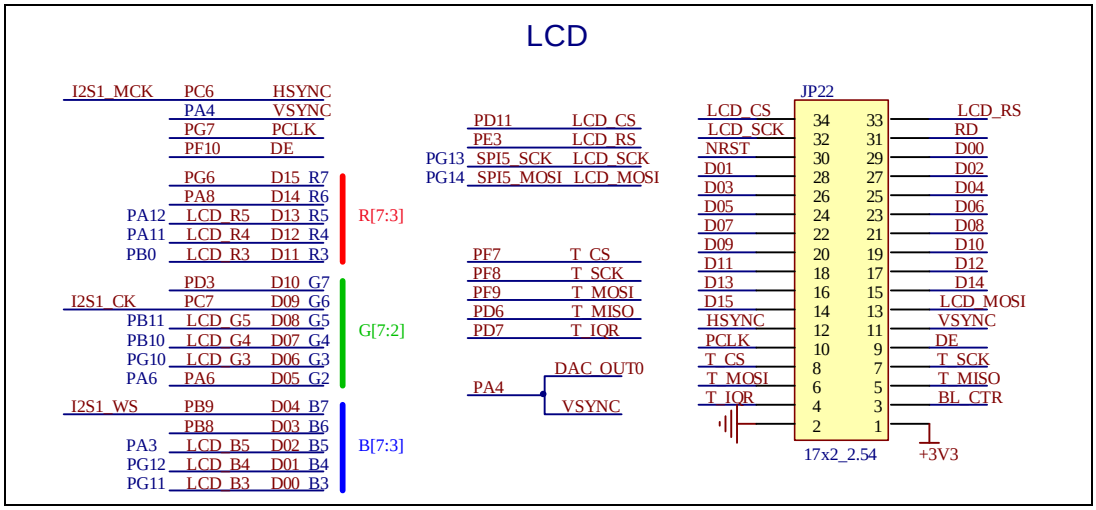
4.14. SDRAM

Figure 4-14 Schematic diagram of SDRAM function



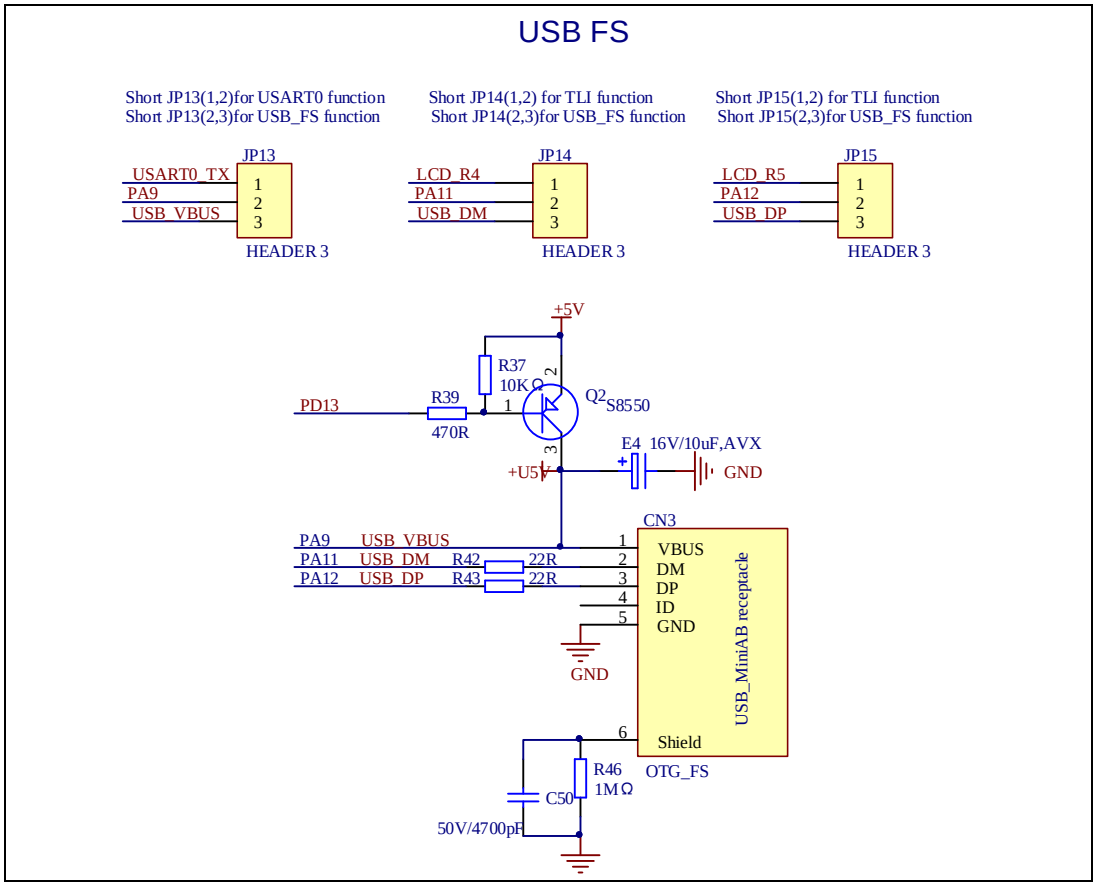
4.15. LCD

Figure 4-15 Schematic diagram of LCD function



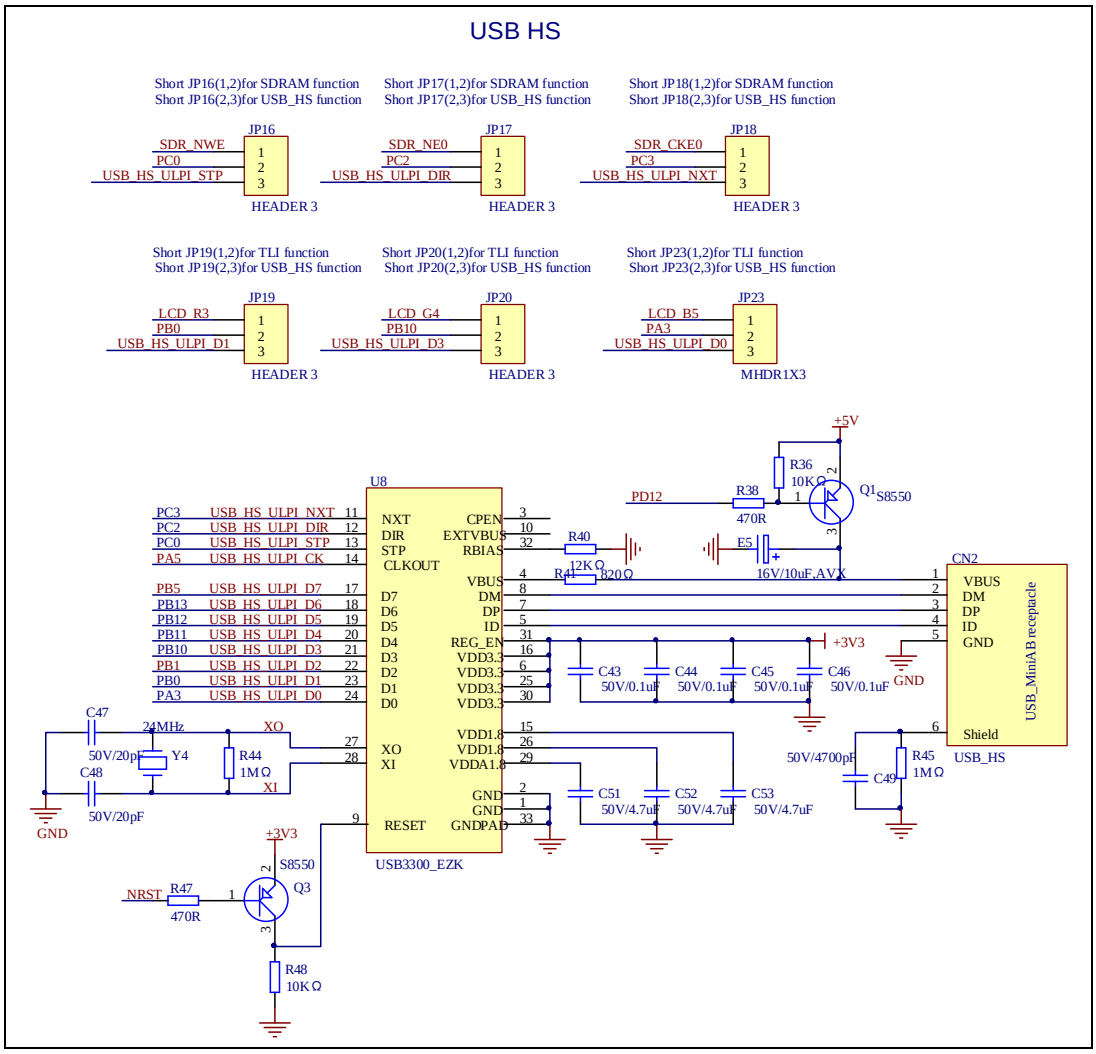
4.16. USBFS

Figure 4-16 Schematic diagram of USBFS function



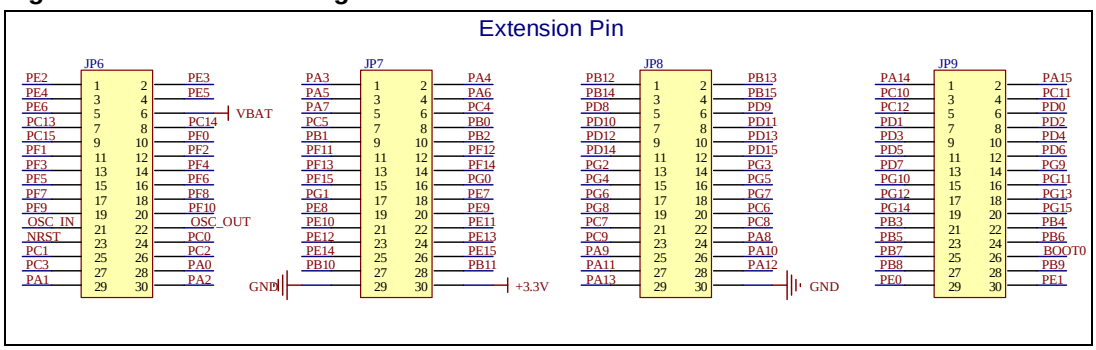
4.17. USBHS

Figure 4-17 Schematic diagram of USBHS function



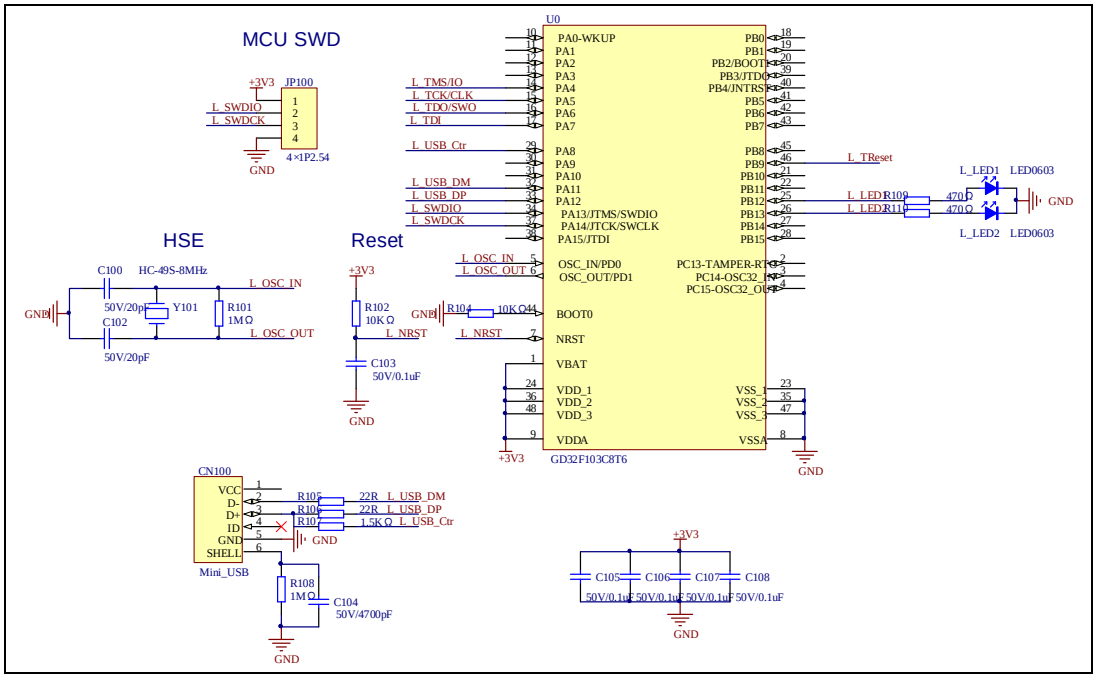
4.18. Extension

Figure 4-18 Schematic diagram of Extension Pin



4.19. GD-Link

Figure 4-19 Schematic diagram of GD-Link



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

GD32450Z-EVAL-V1.1 board has three LEDs. The LED1, LED2 and LED3 are controlled by GPIO. This demo will show how to light the LEDs.

5.1.2. DEMO running result

Download the program <01_GPIO_Running_LED> to the EVAL board, LED1, LED2 and LED3 will turn on in sequence with interval of 400ms, and repeat the process.

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

GD32450Z-EVAL-V1.1 board has four keys and three LEDs. The four keys are Reset key, Tamper key, Wakeup key and User key. The LED1, LED2 and LED3 are controlled by GPIO.

This demo will show how to use the Tamper key to control the LED1. 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 LED1.

5.2.2. DEMO running result

Download the program <02_GPIO_Key_Polling_mode> to the EVAL board, Press down the Tamper Key, LED1 will be turned on. Press down the Tamper Key again, LED1 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

GD32450Z-EVAL-V1.1 board has four keys and three LEDs. The four keys are Reset key, Tamper key, Wakeup key and User key. The LED1, LED2 and LED3 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 LED1.

5.3.2. DEMO running result

Download the program <03_EXTI_Key_Interrupt_mode> to the EVAL board, press down the Tamper Key, LED1 will be turned on. Press down the Tamper Key again, LED1 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, connect serial cable to COM0 and jump JP13 to USART. This implementation outputs “USART printf example: please press the Tamper key” on the HyperTerminal using COM0. Press the Tamper key, the LED1 will be turned on and 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_Echo_Interrupt_mode

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_Echo_Interrupt_mode > to the EVAL board, connect serial cable to COM0 and jump JP13 to USART. Firstly, all the LEDs are turned on and off for test. Then, the COM0 sends the tx_buffer array (from 0x00 to 0xFF) to the serial terminal tool supporting hex format communication and waits for receiving data of same bytes from the serial terminal. The data MCU has received is stored in the rx_buffer array. After that, compare tx_buffer with rx_buffer. If tx_buffer is same with rx_buffer, LED1, LED2, LED3 flash by turns. Otherwise, LED1, LED2, LED3 toggle together.
The output information via the serial port 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, connect serial cable to COM0 and jump JP13 to USART. Firstly, all the LEDs are turned on and off for test. Then, the COM0 sends the tx_buffer array (from 0x00 to 0xFF) to the serial terminal tool supporting hex format communication and waits for receiving data of same bytes as tx_buffer from the serial terminal. The data MCU have received is stored in the rx_buffer array. After that, compare tx_buffer with rx_buffer. If tx_buffer is same with rx_buffer, LED1, LED2, LED3 flash by turns. Otherwise, LED1, LED2, LED3 toggle together.

```
00 01 02 03 04 05 06 07 08 09 0A 0B 0C 0D 0E 0F 10 11 12 13 14 15 16 17 18 19 1A 1B
1C 1D 1E 1F 20 21 22 23 24 25 26 27 28 29 2A 2B 2C 2D 2E 2F 30 31 32 33 34 35 36 37
38 39 3A 3B 3C 3D 3E 3F 40 41 42 43 44 45 46 47 48 49 4A 4B 4C 4D 4E 4F 50 51 52 53
54 55 56 57 58 59 5A 5B 5C 5D 5E 5F 60 61 62 63 64 65 66 67 68 69 6A 6B 6C 6D 6E 6F
70 71 72 73 74 75 76 77 78 79 7A 7B 7C 7D 7E 7F 80 81 82 83 84 85 86 87 88 89 8A 8B
8C 8D 8E 8F 90 91 92 93 94 95 96 97 98 99 9A 9B 9C 9D 9E 9F A0 A1 A2 A3 A4 A5 A6 A7
A8 A9 AA AB AC AD AE AF B0 B1 B2 B3 B4 B5 B6 B7 B8 B9 BA BB BC BD BE BF C0 C1 C2 C3
C4 C5 C6 C7 C8 C9 CA CB CC CD CE CF D0 D1 D2 D3 D4 D5 D6 D7 D8 D9 DA DB DC DD DE DF
E0 E1 E2 E3 E4 E5 E6 E7 E8 E9 EA EB EC ED EE EF F0 F1 F2 F3 F4 F5 F6 F7 F8 F9 FA FB
FC FD FE FF
```

5.7. ADC_Temperature_Vrefint

5.7.1. DEMO purpose

This demo includes the following functions of GD32 MCU:

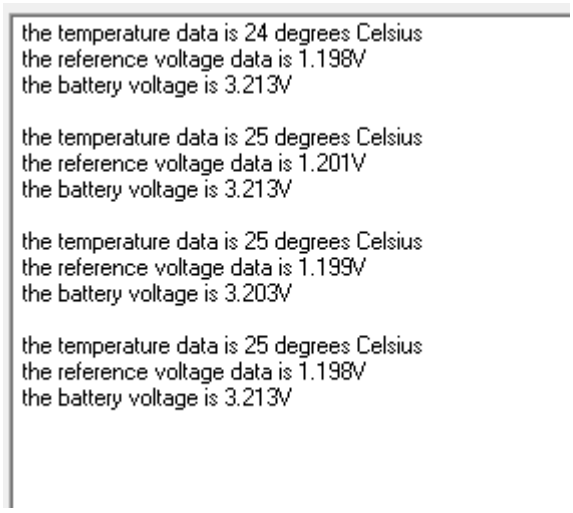
- Learn to use the ADC to convert analog signal to digital data
- Learn to get the value of inner channel 16(temperature sensor channel), channel 17 (VREFINT channel) and channel 18(VBAT/4 channel)

5.7.2. DEMO running result

Jump the JP13 to USART with the jumper cap, and then download the program <07_ADC_Temperature_Vrefint_Vbat> to the board. Connect serial cable to COM0, open the HyperTerminal.

When the program is running, HyperTerminal display the value of temperature, internal voltage reference (VREFINT) and external battery voltage VBAT.

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.



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

Jump the JP13 to USART with the jumper cap, and then download the program <08_ADC0_ADC1_Follow_up_mode> to the board. Connect serial cable to COM0, open the HyperTerminal. PC5 pin connect to the external voltage input. PC3 is the output voltage of the slide rheostat VR1 on board. Keep PC5 pin should not be reused by other peripherals. JP17 should not be connected.

TIMER1_CH1 is the trigger source of ADC0 and ADC1. When the rising edge of TIMER1_CH1 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[0] and adc_value [1] by DMA.

When the first rising edge of TIMER1_CH1 coming, the value of the ADC0 conversion of PC3 pin is stored into the low half word of adc_value [0], and after a delay of several ADC clock cycles 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 TIMER1_CH1 coming, the value of the ADC0 conversion of PC5 pin is stored into the low half word of adc_value [1], 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 [1].

When the program is running, HyperTerminal display the regular value of ADC0 and ADC1 by adc_value [0] and adc_value [1].

```
the data adc_value[0] is 00DE0FF3
the data adc_value[1] is 0FFF00A3

the data adc_value[0] is 00E30FFE
the data adc_value[1] is 0FFF00A4

the data adc_value[0] is 00EA0FF9
the data adc_value[1] is 0FF400B2

the data adc_value[0] is 00DE0FFF
the data adc_value[1] is 0FFE00A9

the data adc_value[0] is 00E00FF1
the data adc_value[1] is 0FF500A6
```

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

Jump the JP13 to USART with the jumper cap, and then download the program <09_ADC0_ADC1_Regular_Parallel_mode> to the board. Connect serial cable to COM0, open the HyperTerminal. PC5 pin connect to the external voltage input. PC3 is the output voltage of the slide rheostat VR1 on board. Keep PC5 pin should not be reused by other peripherals. JP17 should not be connected.

TIMER1_CH1 is the trigger source of ADC0 and ADC1. When the rising edge of TIMER1_CH1 coming, ADC0 and ADC1 convert the regular channel group parallelly. The values of ADC0 and ADC1 are transmitted to array adc_value[0] and adc_value [1] by DMA.

When the first rising edge of TIMER1_CH1 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 TIMER1_CH1 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].

When the program is running, HyperTerminal displays the regular value of ADC0 and ADC1 stored in adc_value [0] and adc_value [1].

```
the data adc_value[0] is 06210000
the data adc_value[1] is 00000627

the data adc_value[0] is 06290B29
the data adc_value[1] is 0B40061F

the data adc_value[0] is 06250B49
the data adc_value[1] is 0B590629

the data adc_value[0] is 06280B3F
the data adc_value[1] is 0B320628

the data adc_value[0] is 06230B30
the data adc_value[1] is 0B430622
```

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 DAC_OUT0 on JP5, its value is 1.65V. And the signal can be observed through the oscilloscope.

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

Download the program <11_I2C_EEPROM> to the EVAL board and run. Connect serial cable to COM0, jump JP13 to USART, then 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 LEDs lights flashing, otherwise the serial port will output "Err: data read and write aren't matching." and all the LEDs light.

The output information via the serial port is as following.

```
GD32450Z-EVAL 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 includes the following functions of GD32 MCU:

- Learn to use the Quad-SPI mode of SPI unit to read and write NOR Flash with the SPI interface

GD32450Z-EVAL-V1.1 board integrates SPI5 module with Quad-SPI mode and the mode can communicate with external NOR Flash devices. The SPI NOR FLASH is a serial FLASH memory chip GD25Q40 which size is 40Mbit, the chip supports standard SPI and quad SPI operation instructions.

5.12.2. 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 JP13 to USART, and jump the JP10 to SPI.

Download the program <12_SPI_QSPI_Flash> to the EVAL board, the HyperTerminal

software can observe the operation condition and will display the ID of the flash, 256 bytes data which are written to and read from flash. Compare the data that were written to the flash and the data that were read from the flash. If they are the same, the serial port will output “SPI-GD25Q40 Test Passed!”, otherwise, the serial port will output “Err: Data Read and Write aren't Matching.”. At last, turn on and off the leds one by one. The following is the experimental results.

```
################################################################################

GD32450Z-EVAL System is Starting up...
GD32450Z-EVAL SystemCoreClock:200000000Hz
GD32450Z-EVAL Flash:65535K
GD32450Z-EVAL The CPU Unique Device ID:[514B3738-C363931-34383B36]
GD32450Z-EVAL SPI Flash:GD25Q40 configured...
The Flash_ID:0xC84015

Write to tx_buffer:
0x00 0x01 0x02 0x03 0x04 0x05 0x06 0x07 0x08 0x09 0x0A 0x0B 0x0C 0x0D 0x0E 0x0F 0x10
0x11 0x12 0x13 0x14 0x15 0x16 0x17 0x18 0x19 0x1A 0x1B 0x1C 0x1D 0x1E 0x1F 0x20 0x21
0x22 0x23 0x24 0x25 0x26 0x27 0x28 0x29 0x2A 0x2B 0x2C 0x2D 0x2E 0x2F 0x30 0x31 0x32
0x33 0x34 0x35 0x36 0x37 0x38 0x39 0x3A 0x3B 0x3C 0x3D 0x3E 0x3F 0x40 0x41 0x42 0x43
0x44 0x45 0x46 0x47 0x48 0x49 0x4A 0x4B 0x4C 0x4D 0x4E 0x4F 0x50 0x51 0x52 0x53 0x54
0x55 0x56 0x57 0x58 0x59 0x5A 0x5B 0x5C 0x5D 0x5E 0x5F 0x60 0x61 0x62 0x63 0x64 0x65
0x66 0x67 0x68 0x69 0x6A 0x6B 0x6C 0x6D 0x6E 0x6F 0x70 0x71 0x72 0x73 0x74 0x75 0x76
0x77 0x78 0x79 0x7A 0x7B 0x7C 0x7D 0x7E 0x7F 0x80 0x81 0x82 0x83 0x84 0x85 0x86 0x87
0x88 0x89 0x8A 0x8B 0x8C 0x8D 0x8E 0x8F 0x90 0x91 0x92 0x93 0x94 0x95 0x96 0x97 0x98
0x99 0x9A 0x9B 0x9C 0x9D 0x9E 0x9F 0xA0 0xA1 0xA2 0xA3 0xA4 0xA5 0xA6 0xA7 0xA8 0xA9
0xAA 0xAB 0xAC 0xAD 0xAE 0xAF 0xB0 0xB1 0xB2 0xB3 0xB4 0xB5 0xB6 0xB7 0xB8 0xB9 0xBA
0xBB 0xBC 0xBD 0xBE 0xBF 0xC0 0xC1 0xC2 0xC3 0xC4 0xC5 0xC6 0xC7 0xC8 0xC9 0xCA 0xCB
0xCC 0xCD 0xCE 0xCF 0xD0 0xD1 0xD2 0xD3 0xD4 0xD5 0xD6 0xD7 0xD8 0xD9 0xDA 0xDB 0xDC
0xDD 0xDE 0xDF 0xE0 0xE1 0xE2 0xE3 0xE4 0xE5 0xE6 0xE7 0xE8 0xE9 0xEA 0xEB 0xEC 0xED
0xEE 0xEF 0xF0 0xF1 0xF2 0xF3 0xF4 0xF5 0xF6 0xF7 0xF8 0xF9 0xFA 0xFB 0xFC 0xFD 0xFE
0xFF

Read from rx_buffer:
0x00 0x01 0x02 0x03 0x04 0x05 0x06 0x07 0x08 0x09 0x0A 0x0B 0x0C 0x0D 0x0E 0x0F 0x10
0x11 0x12 0x13 0x14 0x15 0x16 0x17 0x18 0x19 0x1A 0x1B 0x1C 0x1D 0x1E 0x1F 0x20 0x21
0x22 0x23 0x24 0x25 0x26 0x27 0x28 0x29 0x2A 0x2B 0x2C 0x2D 0x2E 0x2F 0x30 0x31 0x32
0x33 0x34 0x35 0x36 0x37 0x38 0x39 0x3A 0x3B 0x3C 0x3D 0x3E 0x3F 0x40 0x41 0x42 0x43
0x44 0x45 0x46 0x47 0x48 0x49 0x4A 0x4B 0x4C 0x4D 0x4E 0x4F 0x50 0x51 0x52 0x53 0x54
0x55 0x56 0x57 0x58 0x59 0x5A 0x5B 0x5C 0x5D 0x5E 0x5F 0x60 0x61 0x62 0x63 0x64 0x65
0x66 0x67 0x68 0x69 0x6A 0x6B 0x6C 0x6D 0x6E 0x6F 0x70 0x71 0x72 0x73 0x74 0x75 0x76
0x77 0x78 0x79 0x7A 0x7B 0x7C 0x7D 0x7E 0x7F 0x80 0x81 0x82 0x83 0x84 0x85 0x86 0x87
0x88 0x89 0x8A 0x8B 0x8C 0x8D 0x8E 0x8F 0x90 0x91 0x92 0x93 0x94 0x95 0x96 0x97 0x98
0x99 0x9A 0x9B 0x9C 0x9D 0x9E 0x9F 0xA0 0xA1 0xA2 0xA3 0xA4 0xA5 0xA6 0xA7 0xA8 0xA9
0xAA 0xAB 0xAC 0xAD 0xAE 0xAF 0xB0 0xB1 0xB2 0xB3 0xB4 0xB5 0xB6 0xB7 0xB8 0xB9 0xBA
0xBB 0xBC 0xBD 0xBE 0xBF 0xC0 0xC1 0xC2 0xC3 0xC4 0xC5 0xC6 0xC7 0xC8 0xC9 0xCA 0xCB
0xCC 0xCD 0xCE 0xCF 0xD0 0xD1 0xD2 0xD3 0xD4 0xD5 0xD6 0xD7 0xD8 0xD9 0xDA 0xDB 0xDC
0xDD 0xDE 0xDF 0xE0 0xE1 0xE2 0xE3 0xE4 0xE5 0xE6 0xE7 0xE8 0xE9 0xEA 0xEB 0xEC 0xED
0xEE 0xEF 0xF0 0xF1 0xF2 0xF3 0xF4 0xF5 0xF6 0xF7 0xF8 0xF9 0xFA 0xFB 0xFC 0xFD 0xFE
0xFF
SPI-GD25Q40 Test Passed!
```

5.13. I2S_Audio_Player

5.13.1. DEMO purpose

This Demo includes the following functions of GD32 MCU:

- Learn to use I2S module to output audio file
- Parsing audio files of wav format

GD32450Z-EVAL-V1.1 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.

5.13.2. DEMO running result

Download the program<13_I2S_Audio_Player>to the EVAL board, insert the headphone into the audio port, and then listen to the audio file.

5.14. EXMC_SDRAM

5.14.1. DEMO purpose

This demo includes the following functionns of GD32 MCU:

- Learn to use EXMC control the SDRAM

5.14.2. DEMO running result

GD32450Z-EVAL-V1.1 board has EXMC module to control SDRAM. Before running the demo, P2,P3,JP16,JP17,JP18 must be fitted to SDRAM, JP13 must be fitted to USART. Download the program <14_EXMC_SDRAM> to the EVAL board. This demo shows the write and read operation process of SDRAM memory by EXMC module. If the test succeed, LED1 will be turned on. Otherwise, turn on the LED3. Information via a HyperTerminal output as following:

```
SDRAM initialized!
SDRAM write data completed!
SDRAM read data completed!
Check the data!
SDRAM test successed!
The data is:
    0    1    2    3    4    5    6    7    8    9    a    b    c    d    e    f
   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
    0    1    2    3    4    5    6    7    8    9    a    b    c    d    e    f
   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
    0    1    2    3    4    5    6    7    8    9    a    b    c    d    e    f
   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
    0    1    2    3    4    5    6    7    8    9    a    b    c    d    e    f
   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
```

5.15. EXMC_SDRAM_DeepSleep

5.15.1. DEMO purpose

This demo includes the following functions of GD32 MCU:

- Learn to use EXMC control the SDRAM
- Learn to use deepsleep mode

5.15.2. DEMO running result

GD32450Z-EVAL-V1.1 board has EXMC module to control SDRAM. Before running the demo, P2,P3,JP16,JP17,JP18 must be fitted to SDRAM, JP13 must be fitted to USART. Download the program <15_EXMC_SDRAM_DeepSleep> to the EVAL board. This demo shows how to use SDRAM in the deepsleep mode. Firstly, MCU works in the normal mode, SDRAM auto-refresh cycles are performed by MCU, we write the specified data to the SDRAM. Secondly, we make the MCU to deepsleep mode, at the time, SDRAM auto-refresh cycles are performed by itself and LED2 will light on. Thirdly, press the user key to wake up MCU, compare the data which read from SDRAM with the write data, if the test pass, LED1 will be turned on. Otherwise, turn on the LED3. Information via a HyperTerminal output as following:

```
SDRAM initialized!
SDRAM write data completed!
Enter deepsleep mode!
Press the user key to wakeup the MCU!

User key has been pressed!
SDRAM read data completed!
Check the data!
SDRAM test successed!
The data is:
    0   1   2   3   4   5   6   7   8   9   a   b   c   d   e   f
   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
    0   1   2   3   4   5   6   7   8   9   a   b   c   d   e   f
   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
    0   1   2   3   4   5   6   7   8   9   a   b   c   d   e   f
   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
    0   1   2   3   4   5   6   7   8   9   a   b   c   d   e   f
   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
```

5.16. SDIO_SDCardTest

5.16.1. 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

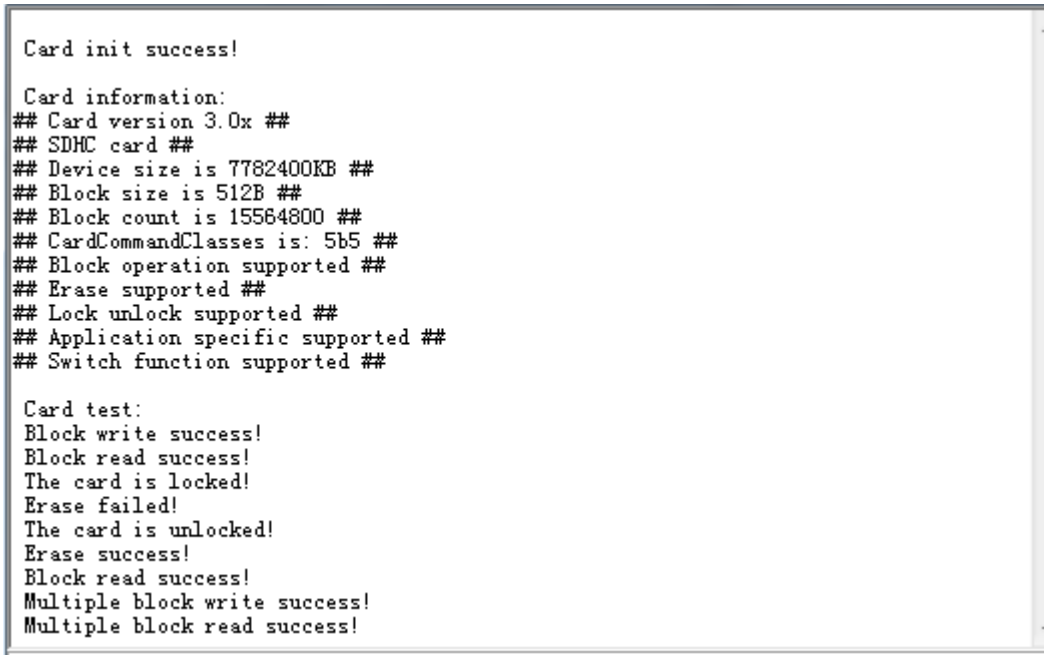
GD32450Z-EVAL-V1.1 board has a secure digital input/output interface (SDIO) which defines the SD/SD I/O /MMC CE-ATA card host interface. This demo will show how to use SDIO to operate on SD card.

5.16.2. DEMO running result

Jump the JP5 to USART to show the print message through HyperTerminal, and download the program <16_SDIO_SDCardTest> to the EVAL board and run. Connect serial cable to COM0, open the HyperTerminal. Firstly, all the LEDs are turned on and off 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 LED1, LED3 and turn off LED2. 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.



5.17. CAN_Network

5.17.1. DEMO purpose

This demo includes the following functions of GD32 MCU:

- Learn to use the CAN0 communication between two boards
- Learn to communicate with PC by USART

5.17.2. DEMO running result

This example is tested with two GD32F450Z-EVAL boards. Jump the JP13 to USART and P2, P3 to CAN0 with the jumper cap. Connect L pin to L pin and H pin to H pin of JP11 on the boards for sending and receiving frames. Download the program <17_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 recive data: ab,cd
```

5.18. RCU_Clock_Out

5.18.1. 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

5.18.2. DEMO running result

Jump the JP5 to USART with the jumper cap, and download the program <18_RCU_Clock_Out> to the EVAL board and run. Connect serial cable to 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 and PC9 pin.

Information via a serial port output as following:

```
 /============ Gigadevice Clock output Demo ============/
press tamper key to select clock output source
CK_OUT0: IRC16M, CK_OUT1: system clock/5
CK_OUT0: LXTAL, CK_OUT1: PLLI2SR/5
```

5.19. CTC_Calibration

5.19.1. DEMO purpose

This demo includes the following functions of GD32 MCU:

- Learn to use external low speed crystal oscillator (LXTAL) to implement the CTC calibration function
- Learn to use clock trim controller (CTC) to trim internal 48MHz RC oscillator (IRC48M) clock

The CTC unit trim the frequency of the IRC48M based on an external accurate reference signal source. It can automatically adjust the trim value to provide a precise IRC48M clock.

5.19.2. DEMO running result

Download the program <19_CTC_Calibration > to the GD32450Z-EVAL-V1.1 board and run. The LED1 will turn on if the internal 48MHz RC oscillator (IRC48M) clock trim is OK.

5.20. PMU_sleep_wakeup

5.20.1. 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

5.20.2. DEMO running result

Download the program < 20_PMU_Sleep_Wakeup > to the EVAL board, jump the JP13 to USART with the jumper cap and connect serial cable to COM0. 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.

5.21. FMC DualBoot

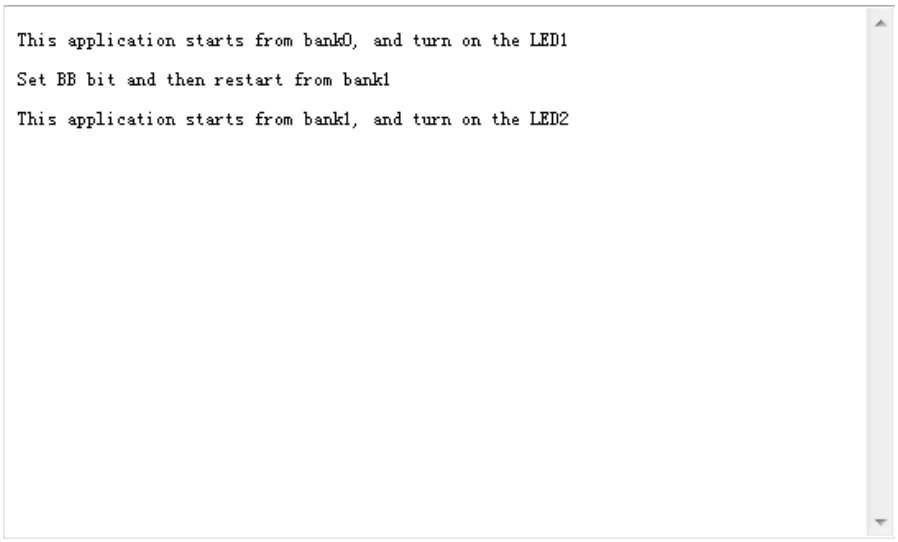
5.21.1. DEMO purpose

This demo includes the following functions of GD32 MCU:

- Learn to use FMC module to implement boot from bank0 and bank1 function
- Learn to set vector table offset

5.21.2. DEMO running result

Jumper JP13 to usart0, open program <21_FMC_DualBoot>, and then select “START_FROM_BANK0” or “START_FROM_BANK1” in the target selection box, compile them and download them to the EVAL board and run. Connect the COM0 port of the EVAL board to the computer and open the HyperTerminal. When the program is running, the serial port software displays the bank area where the currently running program is located. Each time the tamper key is pressed, the BB bit in the FMC option byte will be flipped and a system reset will be generated. When the program boot from bank0, LED1 will be lighten and the serial port will output relevant information. When the program boot from bank1, LED2 will be lighten and the serial port will output relevant information.



5.22. RTC_Calendar

5.22.1. DEMO purpose

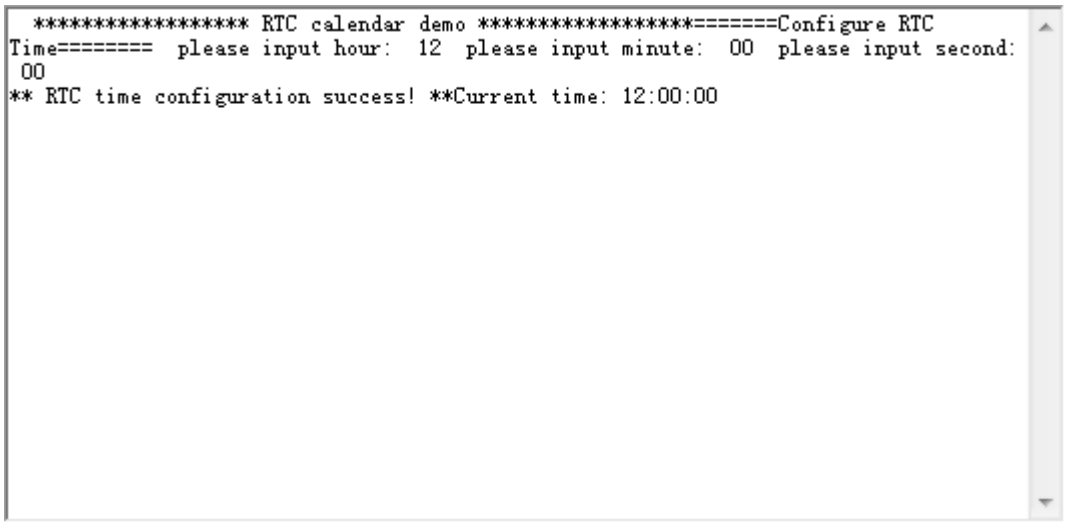
This demo includes the following functions of GD32 MCU:

- Learn to use RTC module to implement calendar function

- Learn to use USART module to implement time display

5.22.2. DEMO running result

Jump the JP5 to USART with the jumper cap, and download the program <22_RTC_Calendar> to the EVAL board and run. Connect serial cable to COM0, open the HyperTerminal. After start-up, the program will ask to set the time on the HyperTerminal. The calendar will be displayed on the HyperTerminal.



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 (PB10) and LED1 (PD4), and then download the program <23_TIMER_Breath_LED> to the board and run.When the program is running, you can see LED1 lighting from dark to bright gradually and then gradually darken, just like breathing as rhythm.

5.24. TLI_IPA

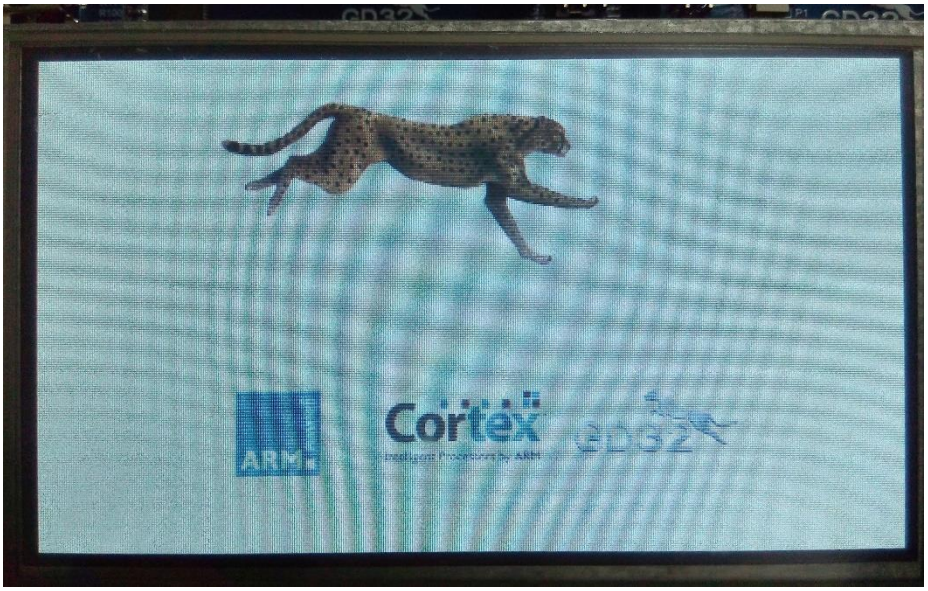
5.24.1. DEMO purpose

This demo includes the following functions of GD32 MCU:

- Learn to use TLI to control LCD for displaying different images
- Learn to use IPA to process image data

5.24.2. DEMO running result

Jump the JP12 to LCD, and download the program <24_TLI_IPA> 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. DC-5V power supply is recommended due to the large current consumption caused by LCD screen.



5.25. TRNG_Get_Random

5.25.1. DEMO purpose

This demo includes the following functions of GD32 MCU:

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

5.25.2. DEMO running result

Jump the JP13 to USART with the jumper cap, and download the program <25_TRNG_Get_Random> to the EVAL board and run. Connect serial cable to 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 0x03, the maximum value is 0x0F), 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 51
Input max num is 54
Generate random num1 is 53
Generate random num2 is 53
Please input min num (hex format):
Please input max num (hex format):
Input min num is 50
Input max num is 57
Generate random num1 is 53
Generate random num2 is 57
Please input min num (hex format):
```

5.26. ENET

5.26.1. 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 GD32450Z-EVAL-V1.1 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, JP5, JP6 must be fitted. JP13 jump to Usart.
It is configured in RMII mode, and 25MHz oscillator is used, the system clock is configured to 200MHz.
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 1026 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 another station, using 1025 port. Users can send information from station to board, then the board will send back the information.

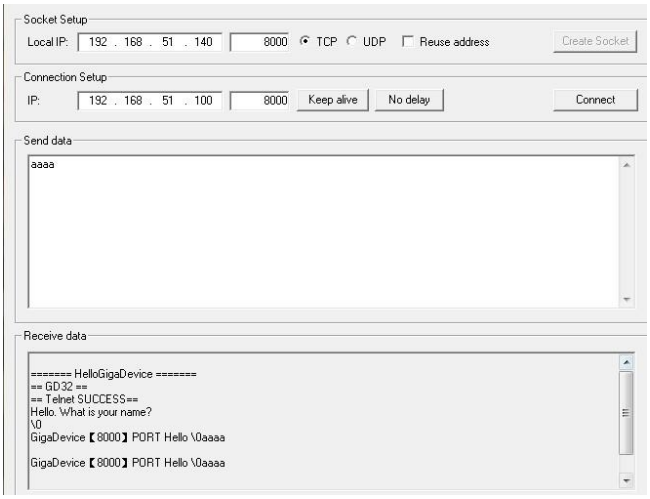
If users need dhcp function, it can be configured from the private defines in main.h. This function is closed by default.

Note: Users should configure ip address, mask and gw of GD32450Z-EVAL-V1.1 board or served according to the actual net situation from the private defines in main.h.

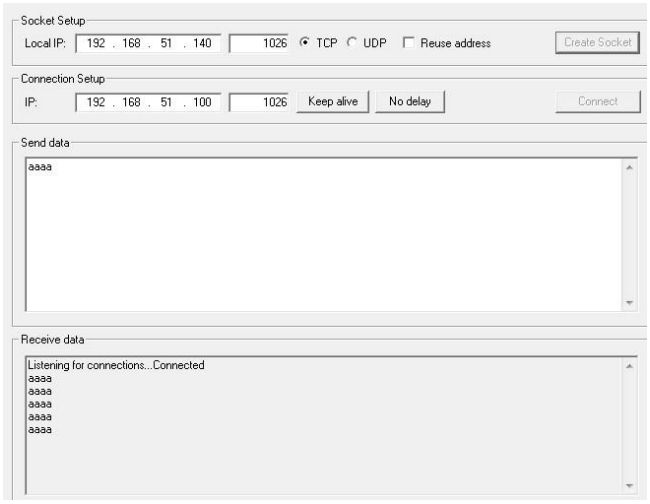
DEMO running result

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

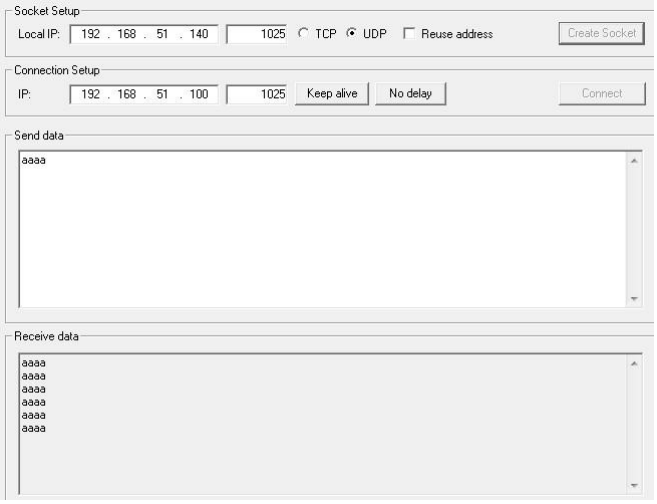
Using Network assistant software, configure the pc side to tcp client, using 8000 port, and when send something through the assistant, users can see the reply from the server:



Using Network assistant software, configure the pc side to tcp server, using 1026 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.

5.26.2. 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 GD32450Z-EVAL-V1.1 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, JP5, JP6 must be fitted. JP13 jump to Usart.
It is configured in RMII mode, and 25MHz oscillator is used, the system clock is configured to 200MHz.
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 1026 port. Users can send information from server to client, 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 another 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.

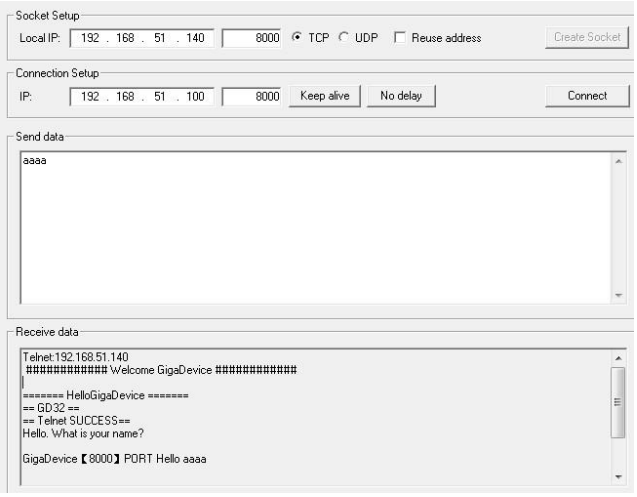
If users need dhcp function, it can be configured from the private defines in main.h. This function is closed in default.

Note: Users should configure ip address, mask and gw of GD32450Z-EVAL-V1.1 board, or server according to the actual net situation from the private defines in main.h.

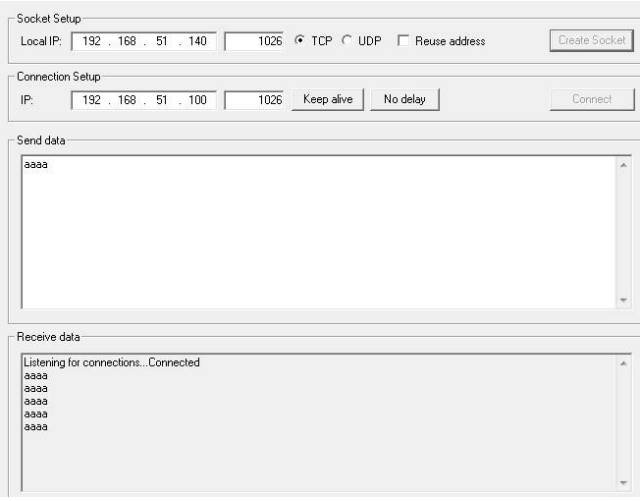
DEMO running result

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:

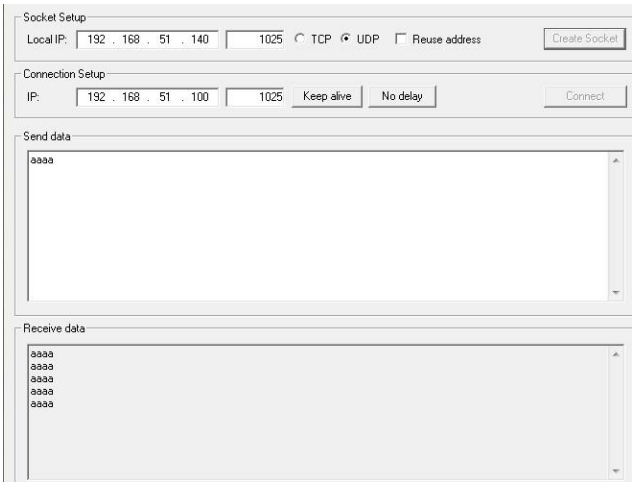


Using Network assistant software, configure the pc side to tcp server, using 1026 port, press the Tamper key, and when send something through the assistant, users can see the echo reply from the client:



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.

5.26.3. 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 GD32450Z-EVAL-V1.1 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, JP5, JP6 must be fitted. JP13 jump to Usart.

It is configured in RMII mode, and 25MHz oscillator is used, the system clock is configured to 200MHz.

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 V_{REFINT} 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.

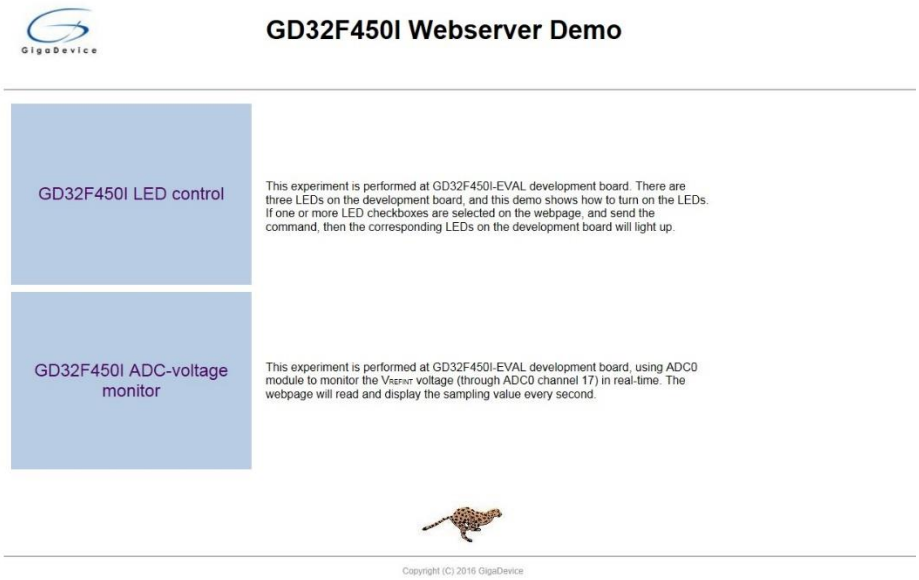
By default, the packet reception is polled in while(1). If users want to receive packet in interrupt service, uncomment the macro define USE_ENET_INTERRUPT in main.h.

Note: Users should configure ip address, mask and gw of GD32450Z-EVAL-V1.1 board according to the actual net situation from the private defines in main.h.

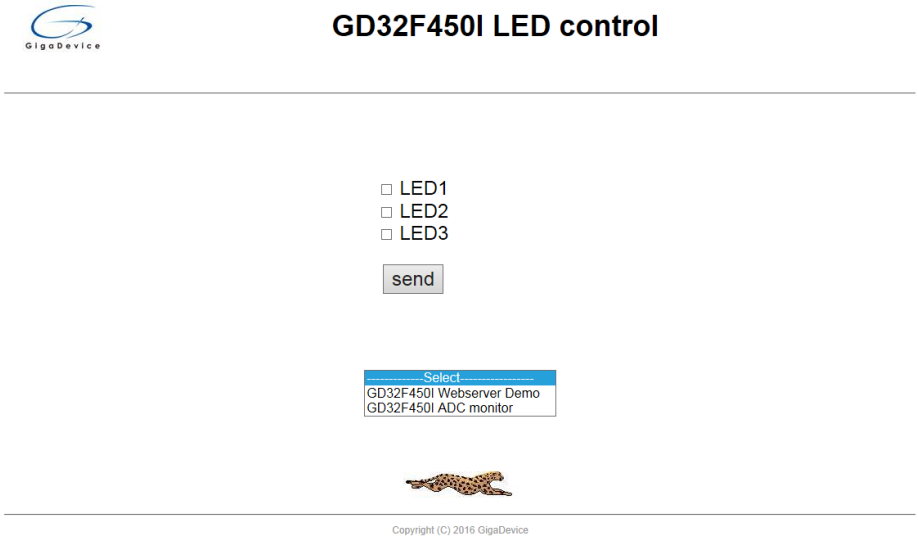
DEMO running result

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 V_{REFINT} voltage is showed on the webpage, and the data refreshes every second automatically.

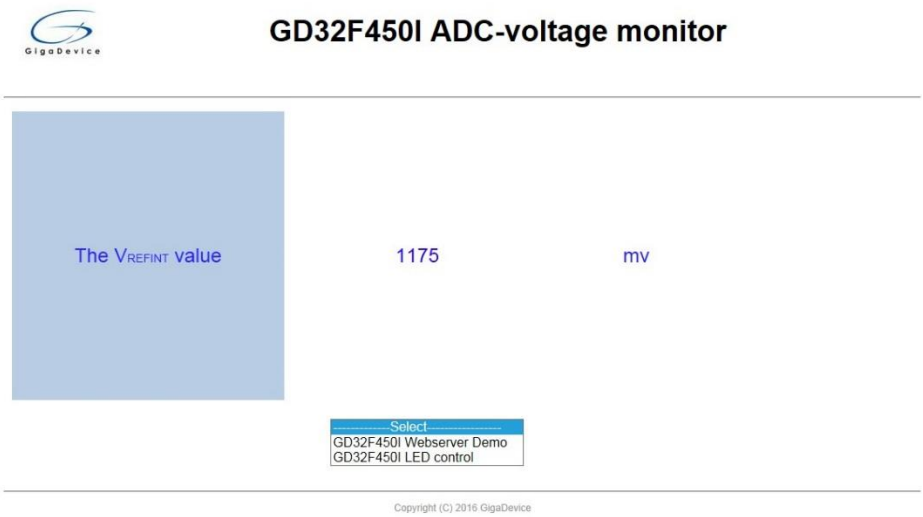
The web home page shows as below:



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.

5.27. USB_Device

5.27.1. 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

GD32450Z-EVAL-V1.1 board has four keys and one USB_FS interface. The four keys are Reset key, Wakeup key, User key and Tamper key. In this demo, the GD32450Z-EVAL-V1.1 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

According to the VBUSIG bit in USBFS_GCCFG register, user can decide whether or not to jump JP13 to USB_FS. Then connect the EVAL board to the PC through USB cable to the USB_FS connector. After doing this, download the program <27_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.

5.27.2. MSC_Udisk

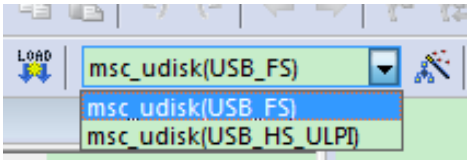
DEMO Purpose

This demo includes the following functions of GD32 MCU:

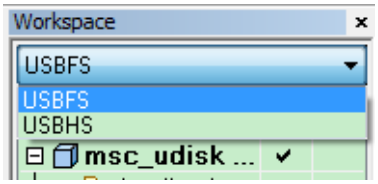
- Learn how to use the USBFS/USBHS peripheral mode
- Learn how to implement USB MSC(mass storage) device

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.

In the workspace toolbar, user should select the project configuration,
In the keil4:
- GD32450Z_EVAL_USBFS: to configure the project for GD32F450Z devices using USBFS peripheral
- GD32450Z_EVAL_USBHS (ULPI): to configure the project for GD32F450Z devices using USBHS peripheral

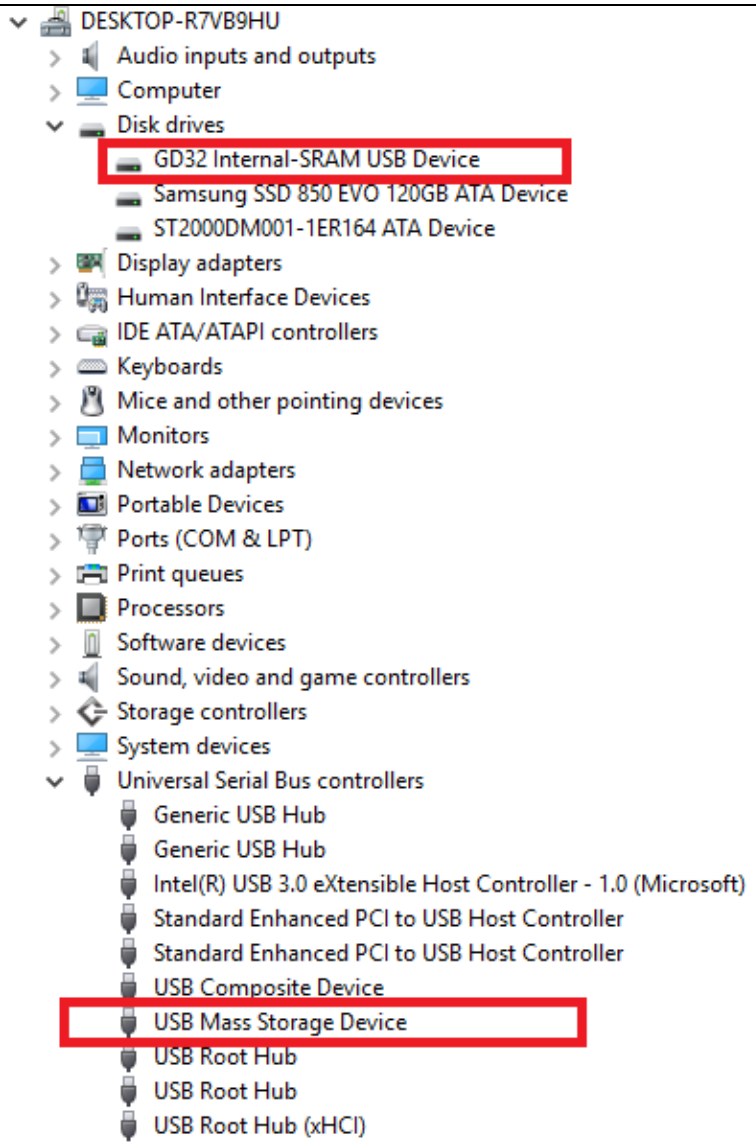


In the IAR:
- USBFS: to configure the project for GD32F450Z devices using USBFS peripheral
- USBHS: to configure the project for GD32F450Z devices using USBHS peripheral

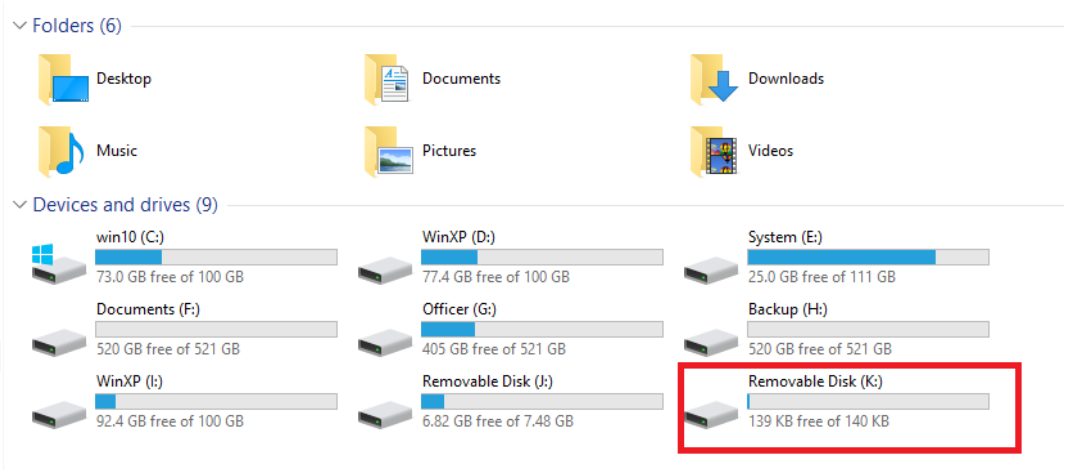


DEMO Running Result

According to the VBUSIG bit in USBFS_GCCFG register, user can decide whether or not to jump JP13 to USB_FS if user use USBFS core. Then connect the EVAL board to the PC through USB cable to the USB_FS connector if user use USBFS core or the USB_HS_ULPI connector if use USBHS core. After doing this, download the program <27_USB_Device\MSC_Udisk> to the EVAL board and run. When the EVAL board connect to the PC, you will find a USB large capacity storage device is in the universal serial bus controller, and there is 1 more disk drives 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.

5.28. USB_Host

5.28.1. 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

GD32450Z-EVAL-V1.1 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.

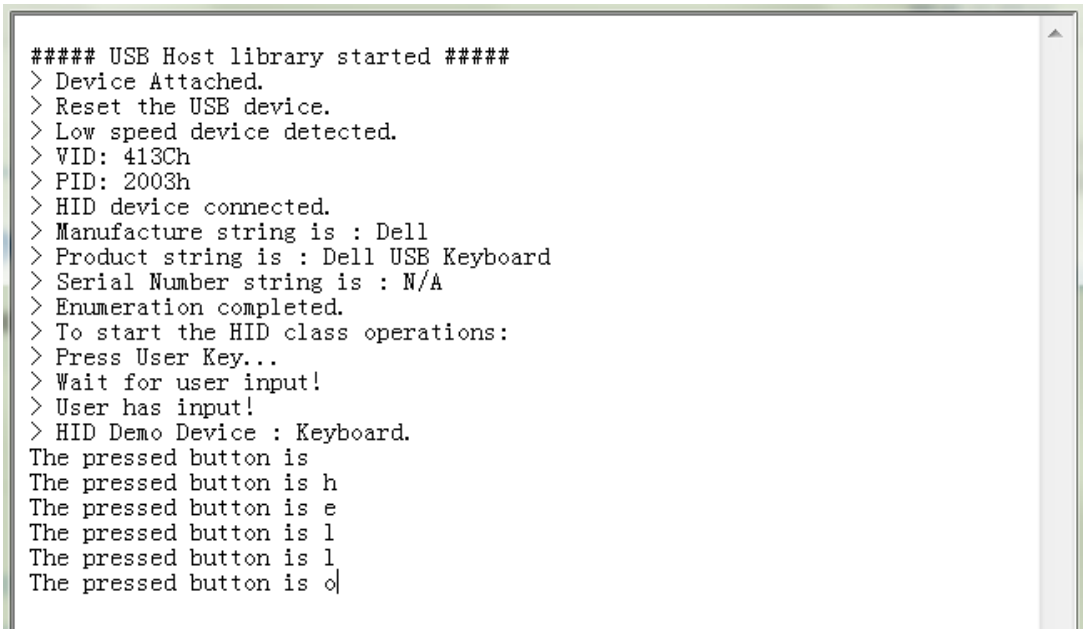
DEMO Running Result

Jump the JP13, JP14 and the JP15 to USB_FS. Then download the program <28_USB_Host\HID_Host> 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.



5.28.2. 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

GD32450Z-EVAL-V1.1 evaluation board integrates the USBFS module and the USBHS 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 and USBHS as a USB MSC host to communicate with external Udisk.

DEMO Running Result

Jump the JP13, JP14 and the JP15 to USB_FS or jump the JP16-20, JP23, P5 and P6 to USB_HS, then insert the OTG cable to the USB port, download the program <28_USB_Host\MSC_Host > 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	Oct.16, 2016
2.0	Update EVAL board	Jan.06,2019

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.