GigaDevice Semiconductor Inc.

GD32F310C-EVAL
Arm® Cortex®-M4 32-bit MCU

User Guide

Revision 1.0

(Mar. 2022)

Table of Contents

List of Figures

List of Tables

1. Summary

GD32F310C-EVAL evaluation board uses GD32F310C8T6 as the main controller. As a complete development platform of GD32F310 powered by Arm® Cortex®-M4 core, the board supports full range of peripherals. It uses mini-USB interface to supply 5V power. GD-Link, Reset, Boot, User button key, LED, I2C, USART, TFT-LCD, IFRP LED、IFRP Transceiver, RTC, SPI, ADC and Extension Pin are also included. This document details its hardware schematic and the relevant applications.

2. Function Pin Assign

Table 2-1. Pin assignment

Function	Pin	Description
LED	PA8	LED1
	PA11	LED2
	PA12	LED3
	PA15	LED4
RESET	-	K1-Reset
KEY	PA0	K2-Wakeup
	PC13	K3-Tamper
IR	PB4	IR_RX
	PB9	IR_TX
I2C	PB6	I2C0_SCL
	PB7	I2C0_SDA
USART0	PA9	RS232_TX
	PA10	RS232_RX
SPI	PB13	SPI1_SCK
	PB14	SPI1_MISO
	PB15	SPI1_MOSI
	PB10	SPI1_IO2
	PB11	SPI1_IO3
	PB12	SPIFlash_CS
	PB0	TFT_CS
	PB1	TFT_RESET
ADC	PA2	ADC_IN2

3. Getting started

The EVAL Board uses mini-USB connecter to get power, the hardware system power is +5.0V. A GD-Link on board is necessary in order to download and debug programs. Select the correct boot mode and then power on, the LEDPWR 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. During use, the following points should be noted:

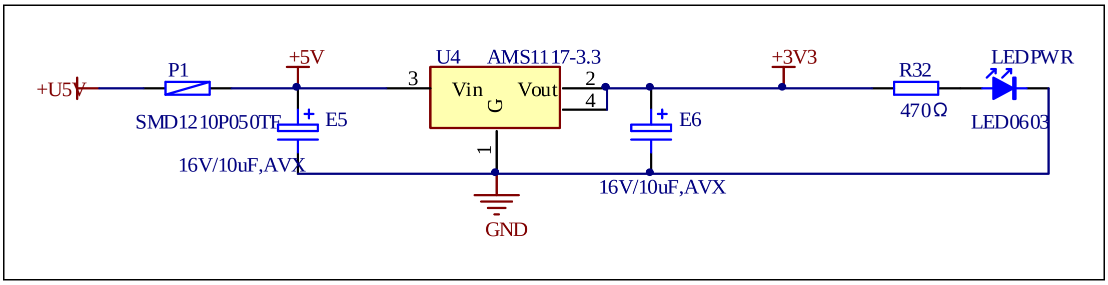
1. If you use Keil uVision4 to open the project. In order to solve the "Device Missing (s)" problem, you can install GigaDevice.GD32F3x0_DFP.3.0.0.pack.

2. If you use IAR to open the project, install IAR_GD32F3x0_ADDON_3.0.0.exe 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

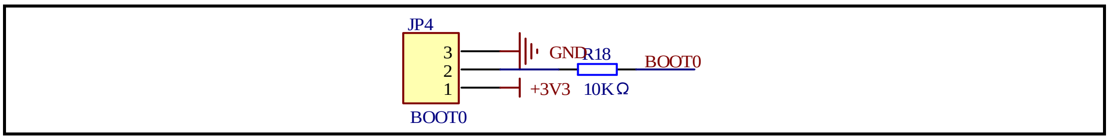
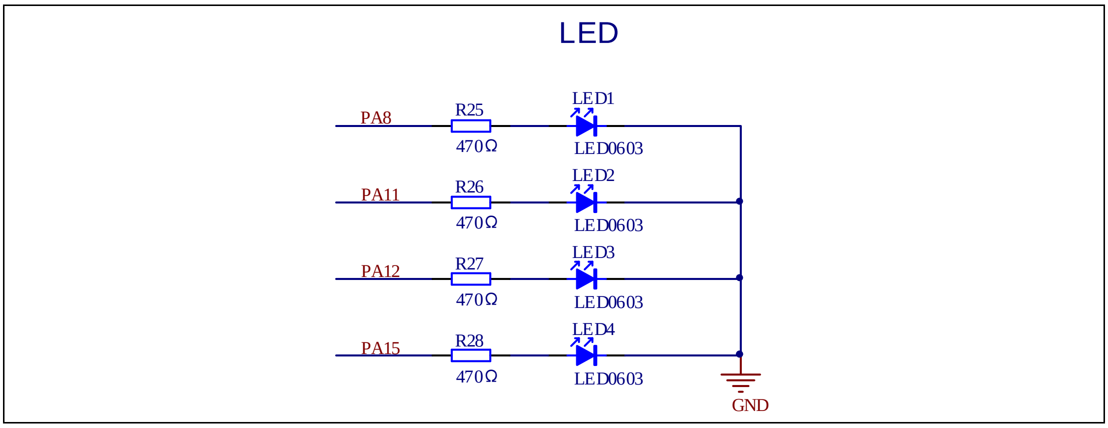


Table 4-1. Boot configuration

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

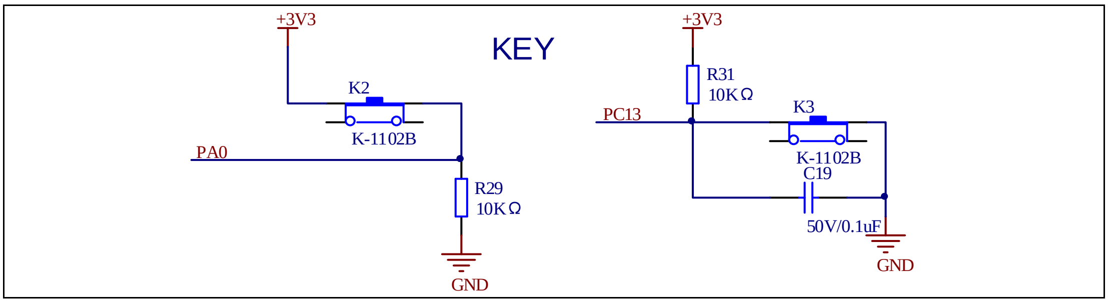
4.3. LED

Figure 4-3. Schematic diagram of LED function



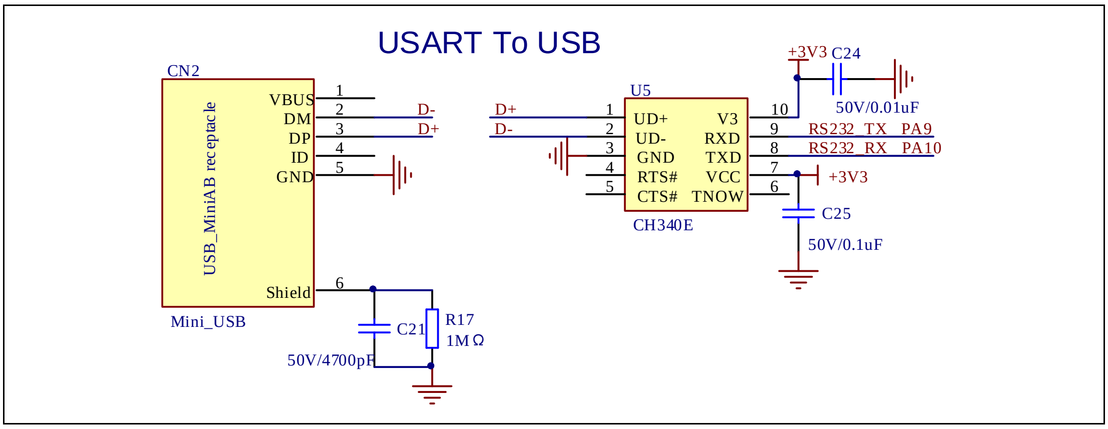
4.4. KEY

Figure 4-4. Schematic diagram of Key function



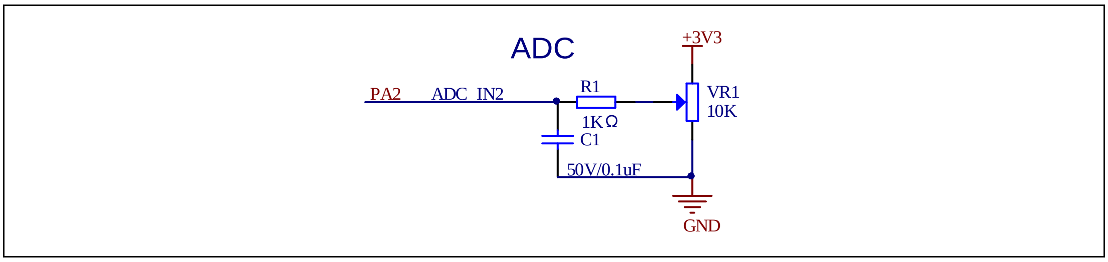
4.5. USART0

Figure 4-5. Schematic diagram of USART0 function



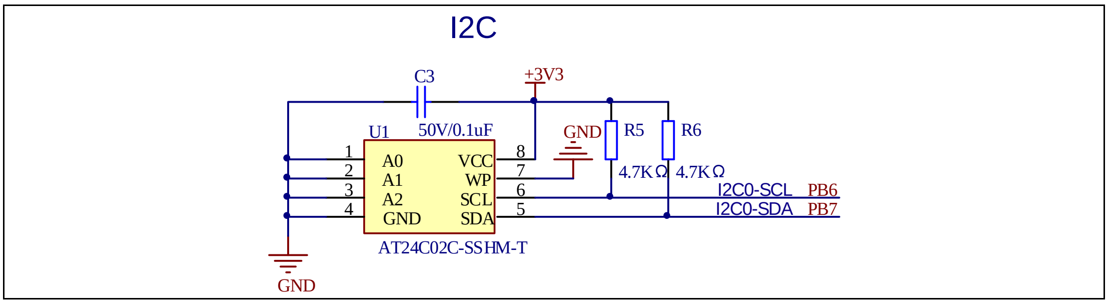
4.6. ADC

Figure 4-6. Schematic diagram of ADC function



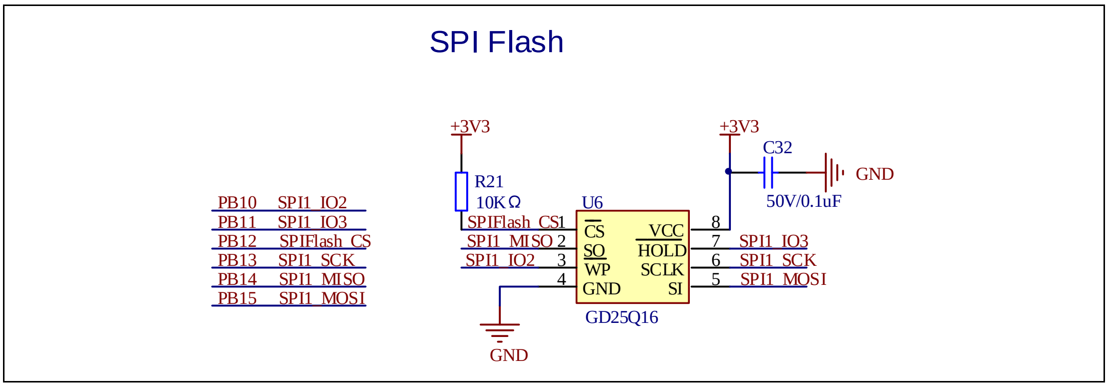
4.7. I2C

Figure 4-7 Schematic diagram of I2C function



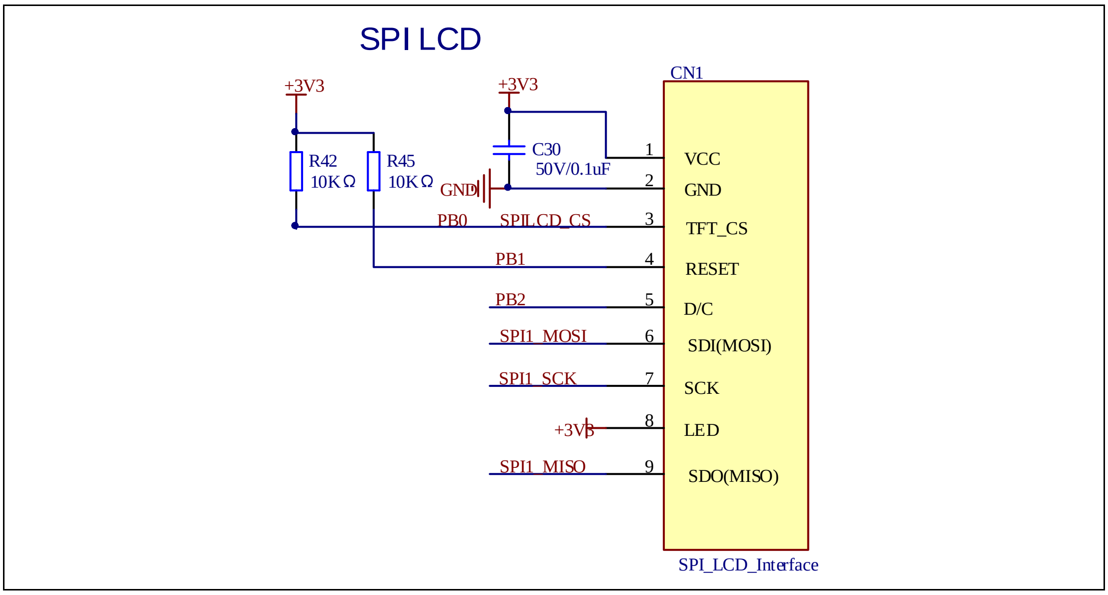
4.8. QSPI-FLASH

Figure 4-8. Schematic diagram of QSPI-FLASH function



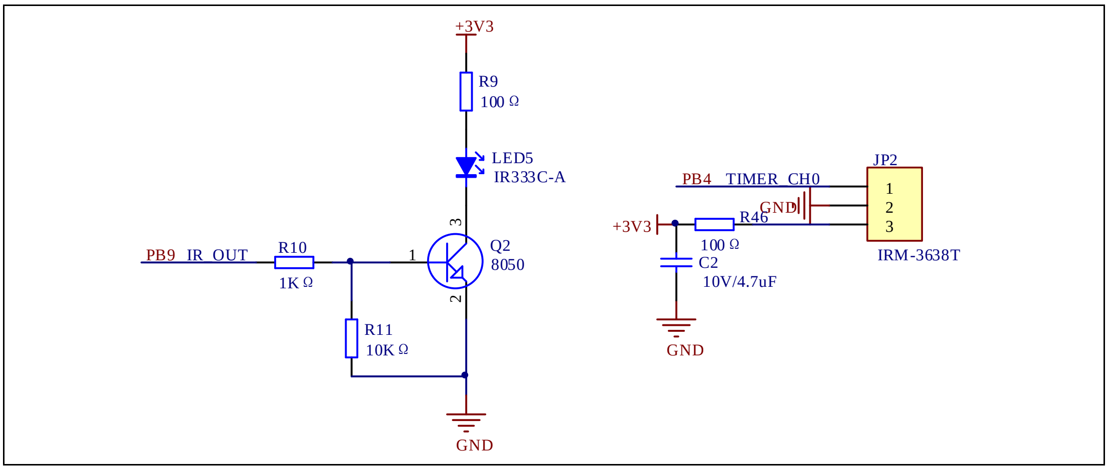
4.9. SPI-TFT LCD

Figure 4-9. Schematic diagram of SPI-TFT LCD function



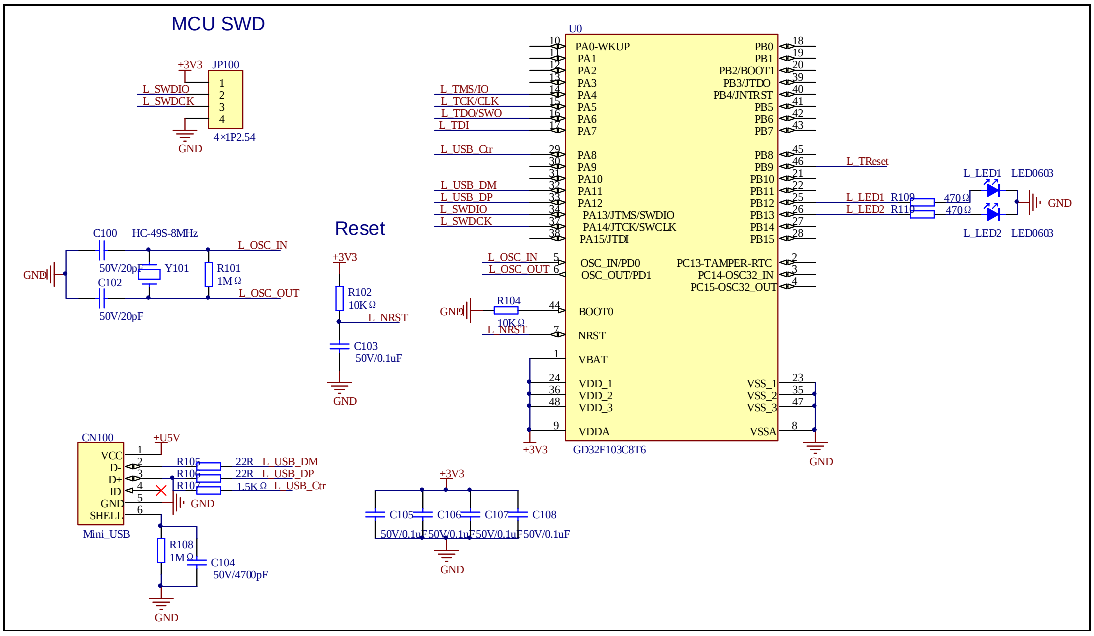
4.10. IFRP

Figure 4-10. Schematic diagram of IFRP function



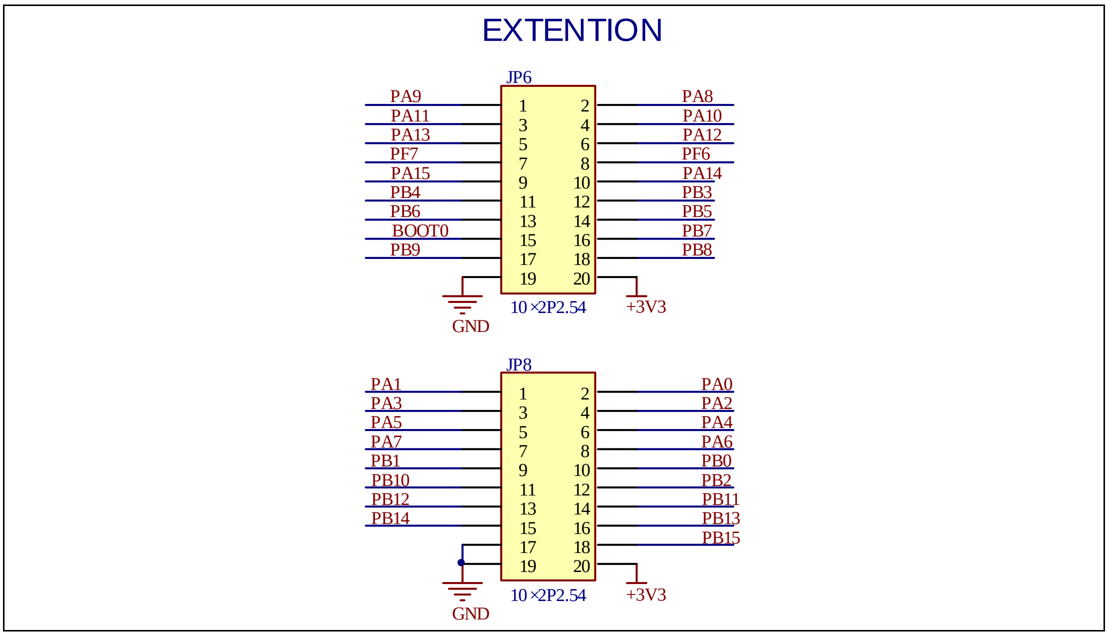
4.11. GD-Link

Figure 4-11. Schematic diagram of GD-Link function



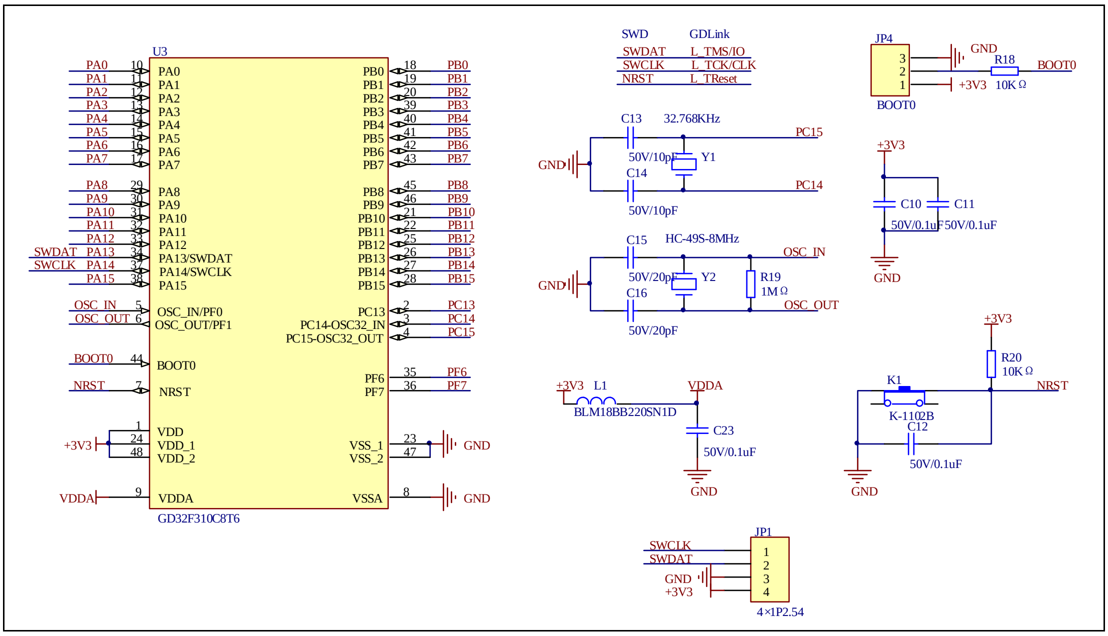
4.12. Extension

Figure 4-12. Schematic diagram of Extension Pin



4.13. MCU

Figure 4-13. Schematic diagram of MCU Pin



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

GD32F310C-EVAL board has four LEDs. The LED1, LED2, LED3 and LED4 are controlled by GPIO. This demo will show how to light the LEDs.

5.1.2. DEMO running result

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

5.2. GPIO_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.

GD32F310C-EVAL board has three keys and four LEDs. The three keys are Reset key, Tamper key and Wakeup key. The LED1, LED2, LED3 and LED4 are controlled by GPIO.

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

5.2.2. DEMO running result

Download the program <02_GPIO_Key_Polling_mode> to the EVAL board, When 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 to control the LED and the KEY
- Learn to use EXTI to generate external interrupt

GD32F310C-EVAL board has three keys and four LEDs. The three keys are Reset key, Wakeup key and Tamper key. The LED1, LED2, LED3 and LED4 are controlled by GPIO.

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

5.3.2. DEMO running result

Download the program <03_EXTI_Key_Interrupt_mode> to the EVAL board, when 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

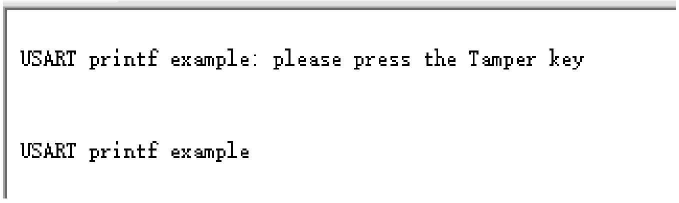
5.4.1. DEMO purpose

This Demo includes the following functions of GD32 MCU:

- Learn to 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 and run. And connect the serial line to COM of EVAL board. This implementation outputs “USART printf example: please press the Tamper Key” on the hyperterminal using COM. Press the Tamper key, serial port will output “USART Printf Example”. The information via a serial port output as following.



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

USART printf example
```

5.5. USART_HyperTerminal_Interrupt

5.5.1. DEMO purpose

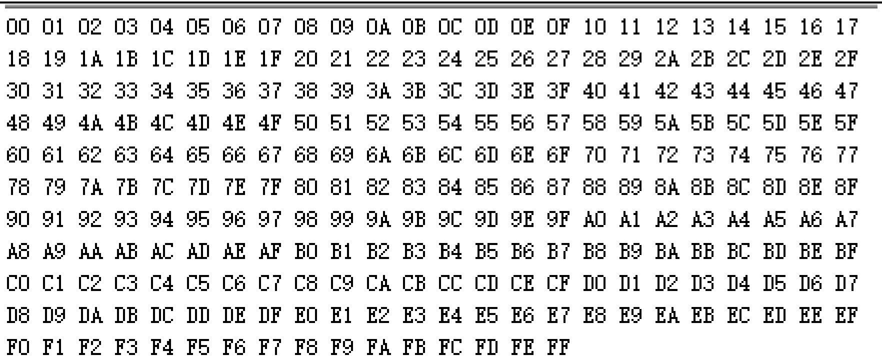
This Demo includes the following functions of GD32 MCU:

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

5.5.2. DEMO running result

Download the program <05_USART_HyperTerminal_Interrupt> to the EVAL board and run. And connect the serial line to COM of EVAL board. Firstly, all the LEDs are turned on and off for test. Then, the COM sends the tx_buffer array (from 0x00 to 0xFF) to the hyperterminal and waits for receiving data from the hyperterminal that you must send. The string that you have sent is stored in the rx_buffer array. The receive buffer have a BUFFER_SIZE bytes as maximum. After that, compare tx_buffer with rx_buffer. If tx_buffer is same with rx_buffer, LED1 and LED2 are turned on, LED3 and LED4 are turned off. Otherwise, LED1 and LED2 are turned off, LED3 and LED4 are turned on.

The information via a serial port output as following:



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

5.6. USART_DMA

5.6.1. DEMO purpose

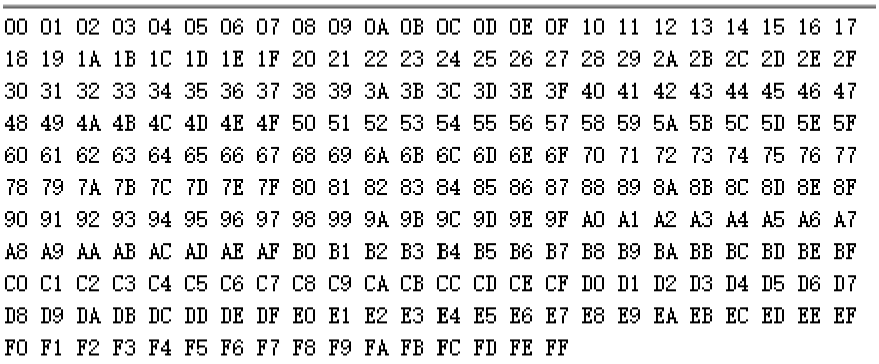
This Demo includes the following functions of GD32 MCU:

- Learn to use the COM transmit and receive using DMA

5.6.2. DEMO running result

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

The information via a serial port output as following:



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

5.7. ADC_conversion_triggered_by_timer

5.7.1. DEMO purpose

This Demo includes the following functions of GD32 MCU:

- Learn to use ADC to convert analog to digital
- Learn to use TIMER to generate a channel compare event
- Learn to use LCD to show the ADC converted result

TIMER0 CH0 event triggers ADC conversion, the value displayed on the LCD corresponds to the ADC analog input, and changes with it. The converted data are moved to SRAM through DMA continuously.

5.7.2. DEMO running result

Download the program <07_ADC_conversion_triggered_by_timer> to the GD32F310C-EVAL board, adjust the adjustable potentiometer knob to change the analog input. The ADC, which is triggered by TIMER0 CH0 event, will convert the analog input, and you will see the result, a voltage curve, on the LCD. The curve adjusts with the analog input.

5.8. I2C_EEPROM

5.8.1. DEMO purpose

This Demo includes the following functions of GD32 MCU:

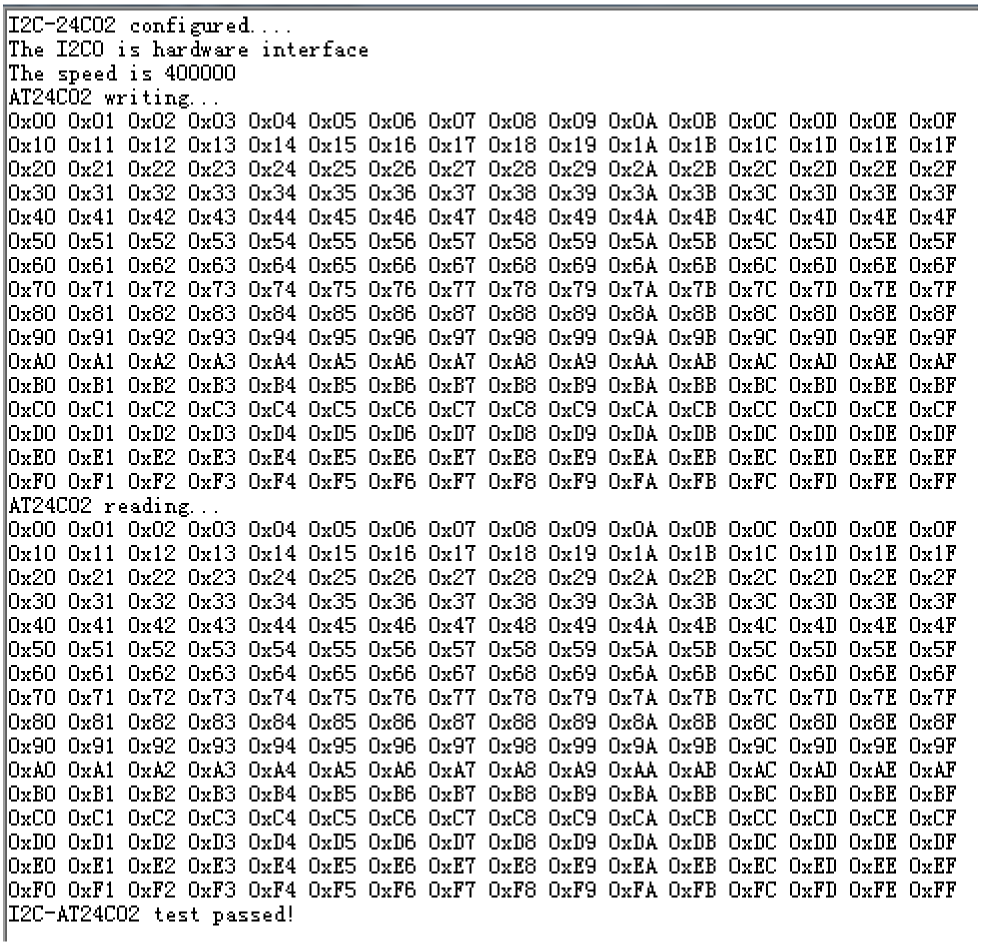
- Learn how to use the master transmitting mode of the I2C module
- Learn how to use the master receiving mode of the I2C module
- Learn to read and write the EEPROM with the I2C interface

5.8.2. DEMO running result

Download the program <08_I2C_EEPROM> to the EVAL board and run. Connect serial cable to COM, and open the HyperTerminal to show the print message.

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

The output information via the serial port is as following.



```
I2C-24C02 configured....
The I2C0 is hardware interface
The speed is 400000
AT24C02 writing...
0x00 0x01 0x02 0x03 0x04 0x05 0x06 0x07 0x08 0x09 0x0A 0x0B 0x0C 0x0D 0x0E 0x0F
0x10 0x11 0x12 0x13 0x14 0x15 0x16 0x17 0x18 0x19 0x1A 0x1B 0x1C 0x1D 0x1E 0x1F
0x20 0x21 0x22 0x23 0x24 0x25 0x26 0x27 0x28 0x29 0x2A 0x2B 0x2C 0x2D 0x2E 0x2F
0x30 0x31 0x32 0x33 0x34 0x35 0x36 0x37 0x38 0x39 0x3A 0x3B 0x3C 0x3D 0x3E 0x3F
0x40 0x41 0x42 0x43 0x44 0x45 0x46 0x47 0x48 0x49 0x4A 0x4B 0x4C 0x4D 0x4E 0x4F
0x50 0x51 0x52 0x53 0x54 0x55 0x56 0x57 0x58 0x59 0x5A 0x5B 0x5C 0x5D 0x5E 0x5F
0x60 0x61 0x62 0x63 0x64 0x65 0x66 0x67 0x68 0x69 0x6A 0x6B 0x6C 0x6D 0x6E 0x6F
0x70 0x71 0x72 0x73 0x74 0x75 0x76 0x77 0x78 0x79 0x7A 0x7B 0x7C 0x7D 0x7E 0x7F
0x80 0x81 0x82 0x83 0x84 0x85 0x86 0x87 0x88 0x89 0x8A 0x8B 0x8C 0x8D 0x8E 0x8F
0x90 0x91 0x92 0x93 0x94 0x95 0x96 0x97 0x98 0x99 0x9A 0x9B 0x9C 0x9D 0x9E 0x9F
0xA0 0xA1 0xA2 0xA3 0xA4 0xA5 0xA6 0xA7 0xA8 0xA9 0xAA 0xAB 0xAC 0xAD 0xAE 0xAF
0xB0 0xB1 0xB2 0xB3 0xB4 0xB5 0xB6 0xB7 0xB8 0xB9 0xBA 0xBB 0xBC 0xBD 0xBE 0xBF
0xC0 0xC1 0xC2 0xC3 0xC4 0xC5 0xC6 0xC7 0xC8 0xC9 0xCA 0xCB 0xCC 0xCD 0xCE 0xCF
0xD0 0xD1 0xD2 0xD3 0xD4 0xD5 0xD6 0xD7 0xD8 0xD9 0xDA 0xDB 0xDC 0xDD 0xDE 0xDF
0xE0 0xE1 0xE2 0xE3 0xE4 0xE5 0xE6 0xE7 0xE8 0xE9 0xEA 0xEB 0xEC 0xED 0xEE 0xEF
0xF0 0xF1 0xF2 0xF3 0xF4 0xF5 0xF6 0xF7 0xF8 0xF9 0xFA 0xFB 0xFC 0xFD 0xFE 0xFF
AT24C02 reading...
0x00 0x01 0x02 0x03 0x04 0x05 0x06 0x07 0x08 0x09 0x0A 0x0B 0x0C 0x0D 0x0E 0x0F
0x10 0x11 0x12 0x13 0x14 0x15 0x16 0x17 0x18 0x19 0x1A 0x1B 0x1C 0x1D 0x1E 0x1F
0x20 0x21 0x22 0x23 0x24 0x25 0x26 0x27 0x28 0x29 0x2A 0x2B 0x2C 0x2D 0x2E 0x2F
0x30 0x31 0x32 0x33 0x34 0x35 0x36 0x37 0x38 0x39 0x3A 0x3B 0x3C 0x3D 0x3E 0x3F
0x40 0x41 0x42 0x43 0x44 0x45 0x46 0x47 0x48 0x49 0x4A 0x4B 0x4C 0x4D 0x4E 0x4F
0x50 0x51 0x52 0x53 0x54 0x55 0x56 0x57 0x58 0x59 0x5A 0x5B 0x5C 0x5D 0x5E 0x5F
0x60 0x61 0x62 0x63 0x64 0x65 0x66 0x67 0x68 0x69 0x6A 0x6B 0x6C 0x6D 0x6E 0x6F
0x70 0x71 0x72 0x73 0x74 0x75 0x76 0x77 0x78 0x79 0x7A 0x7B 0x7C 0x7D 0x7E 0x7F
0x80 0x81 0x82 0x83 0x84 0x85 0x86 0x87 0x88 0x89 0x8A 0x8B 0x8C 0x8D 0x8E 0x8F
0x90 0x91 0x92 0x93 0x94 0x95 0x96 0x97 0x98 0x99 0x9A 0x9B 0x9C 0x9D 0x9E 0x9F
0xA0 0xA1 0xA2 0xA3 0xA4 0xA5 0xA6 0xA7 0xA8 0xA9 0xAA 0xAB 0xAC 0xAD 0xAE 0xAF
0xB0 0xB1 0xB2 0xB3 0xB4 0xB5 0xB6 0xB7 0xB8 0xB9 0xBA 0xBB 0xBC 0xBD 0xBE 0xBF
0xC0 0xC1 0xC2 0xC3 0xC4 0xC5 0xC6 0xC7 0xC8 0xC9 0xCA 0xCB 0xCC 0xCD 0xCE 0xCF
0xD0 0xD1 0xD2 0xD3 0xD4 0xD5 0xD6 0xD7 0xD8 0xD9 0xDA 0xDB 0xDC 0xDD 0xDE 0xDF
0xE0 0xE1 0xE2 0xE3 0xE4 0xE5 0xE6 0xE7 0xE8 0xE9 0xEA 0xEB 0xEC 0xED 0xEE 0xEF
0xF0 0xF1 0xF2 0xF3 0xF4 0xF5 0xF6 0xF7 0xF8 0xF9 0xFA 0xFB 0xFC 0xFD 0xFE 0xFF
I2C-AT24C02 test passed!
```

5.9. QSPI_FLASH

5.9.1. DEMO purpose

This demo includes the following functions of GD32 MCU:

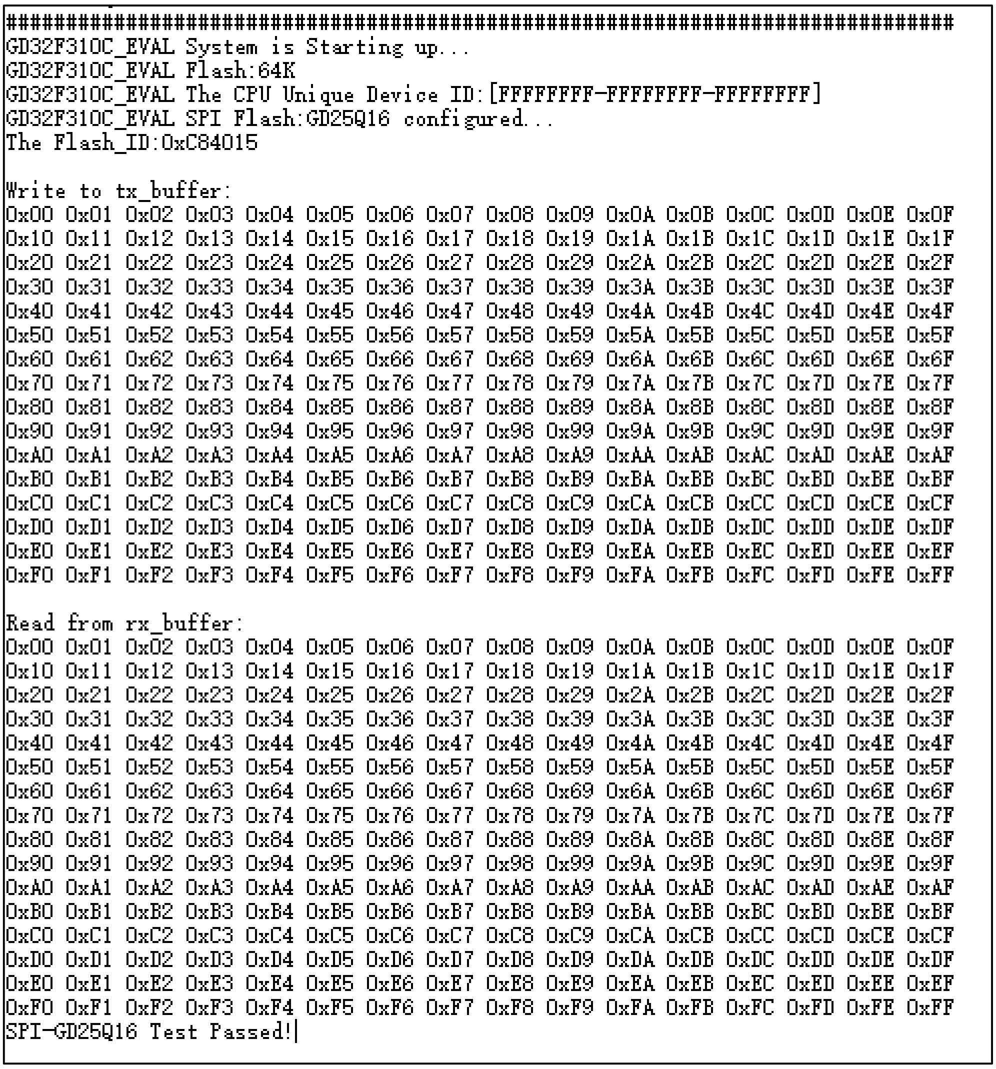
- Learn to use the Quad-SPI mode of SPI unit to read and write NOR Flash with the SPI interface

5.9.2. DEMO running result

The computer serial port line connected to the COM port of development board, set the baud rate of HyperTerminal software to 115200, 8 bits data bit, 1 bit stop bit.

Download the program <09_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-GD25Q16 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.



```
################################################################################
GD32F310C_EVAL System is Starting up...
GD32F310C_EVAL Flash:64K
GD32F310C_EVAL The CPU Unique Device ID:[FFFFFFFF-FFFFFFFF-FFFFFFFF]
GD32F310C_EVAL SPI Flash:GD25Q16 configured...
The Flash_ID:0xC84015

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

Read from rx_buffer:
0x00 0x01 0x02 0x03 0x04 0x05 0x06 0x07 0x08 0x09 0x0A 0x0B 0x0C 0x0D 0x0E 0x0F
0x10 0x11 0x12 0x13 0x14 0x15 0x16 0x17 0x18 0x19 0x1A 0x1B 0x1C 0x1D 0x1E 0x1F
0x20 0x21 0x22 0x23 0x24 0x25 0x26 0x27 0x28 0x29 0x2A 0x2B 0x2C 0x2D 0x2E 0x2F
0x30 0x31 0x32 0x33 0x34 0x35 0x36 0x37 0x38 0x39 0x3A 0x3B 0x3C 0x3D 0x3E 0x3F
0x40 0x41 0x42 0x43 0x44 0x45 0x46 0x47 0x48 0x49 0x4A 0x4B 0x4C 0x4D 0x4E 0x4F
0x50 0x51 0x52 0x53 0x54 0x55 0x56 0x57 0x58 0x59 0x5A 0x5B 0x5C 0x5D 0x5E 0x5F
0x60 0x61 0x62 0x63 0x64 0x65 0x66 0x67 0x68 0x69 0x6A 0x6B 0x6C 0x6D 0x6E 0x6F
0x70 0x71 0x72 0x73 0x74 0x75 0x76 0x77 0x78 0x79 0x7A 0x7B 0x7C 0x7D 0x7E 0x7F
0x80 0x81 0x82 0x83 0x84 0x85 0x86 0x87 0x88 0x89 0x8A 0x8B 0x8C 0x8D 0x8E 0x8F
0x90 0x91 0x92 0x93 0x94 0x95 0x96 0x97 0x98 0x99 0x9A 0x9B 0x9C 0x9D 0x9E 0x9F
0xA0 0xA1 0xA2 0xA3 0xA4 0xA5 0xA6 0xA7 0xA8 0xA9 0xAA 0xAB 0xAC 0xAD 0xAE 0xAF
0xB0 0xB1 0xB2 0xB3 0xB4 0xB5 0xB6 0xB7 0xB8 0xB9 0xBA 0xBB 0xBC 0xBD 0xBE 0xBF
0xC0 0xC1 0xC2 0xC3 0xC4 0xC5 0xC6 0xC7 0xC8 0xC9 0xCA 0xCB 0xCC 0xCD 0xCE 0xCF
0xD0 0xD1 0xD2 0xD3 0xD4 0xD5 0xD6 0xD7 0xD8 0xD9 0xDA 0xDB 0xDC 0xDD 0xDE 0xDF
0xE0 0xE1 0xE2 0xE3 0xE4 0xE5 0xE6 0xE7 0xE8 0xE9 0xEA 0xEB 0xEC 0xED 0xEE 0xEF
0xF0 0xF1 0xF2 0xF3 0xF4 0xF5 0xF6 0xF7 0xF8 0xF9 0xFA 0xFB 0xFC 0xFD 0xFE 0xFF
SPI-GD25Q16 Test Passed!|
```

5.10. SPI_TFT_LCD_Driver

5.10.1. DEMO purpose

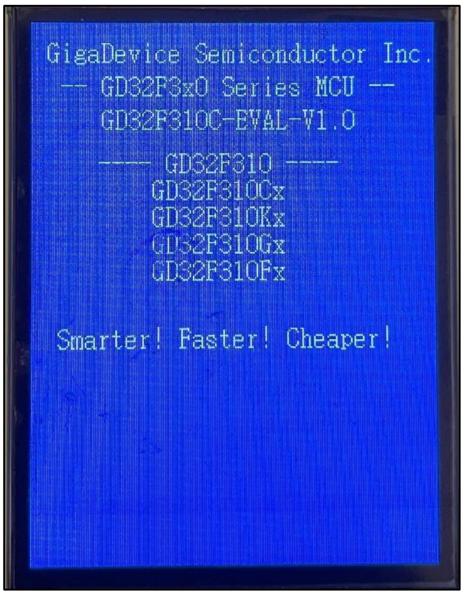
This Demo includes the following function of GD32 MCU:

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

GD32F310C-EVAL board has a TFT LCD screen which supports SPI interface. In this demo, tests of font, number, draw and color are displayed on the LCD screen respectively.

5.10.2. DEMO running result

Download the program <10_SPI_TFT_LCD_Driver> to the EVAL board. All the LEDs are turned on and then turned off for test. After that, the LCD screen on the board will display the GUI tests in infinite loop.



GigaDevice Semiconductor Inc.
-- GD32F3x0 Series MCU --
GD32F310C-EVAL-V1.0
----- GD32F310 -----
GD32F310Cx
GD32F310Kx
GD32F310Gx
GD32F310Fx
Smarter! Faster! Cheaper!

5.11. RCU_Clock_Out

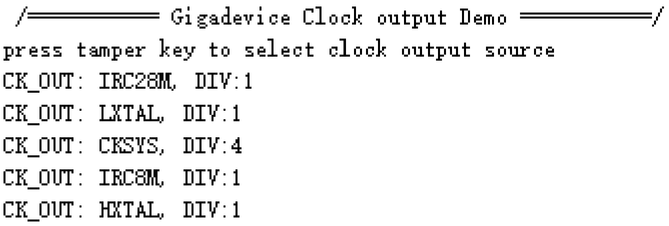
5.11.1. DEMO purpose

This Demo includes the following functions of GD32 MCU:

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

5.11.2. DEMO running result

Download the program <11_RCU_Clock_Out> to the EVAL board and run. Connect serial cable to EVAL_COM, open the HyperTerminal. When the program is running, HyperTerminal will display the initial information. Then user can choose the type of the output clock by pressing the Tamper button. After pressing, the LED will be lit in turn 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_OUT: IRC28M, DIV:1
CK_OUT: LXTAL, DIV:1
CK_OUT: CKSYS, DIV:4
CK_OUT: IRC8M, DIV:1
CK_OUT: HXTAL, DIV:1
```

5.12. CTC_Calibration

5.12.1. 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 automaticly adjust the trim value to provide a precise IRC48M clock.

5.12.2. DEMO running result

Download the program <12_CTC_Calibration> to the EVAL board and run. The LED1 will turn on if the internal 48MHz RC oscillator (IRC48M) clock trim is OK.

5.13. PMU_sleep_wakeup

5.13.1. 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

5.13.2. DEMO running result

Download the program <13_PMU_sleep_wakeup> to the EVAL board, connect serial cable to EVAL_COM. After power-on, all the LEDs are off. The MCU will enter sleep mode and the software stop running. When the USART0 receives a byte of data from the HyperTerminal, the MCU will wake up from a receive interrupt. And all the LEDs will flash together.

5.14. RTC_Calendar

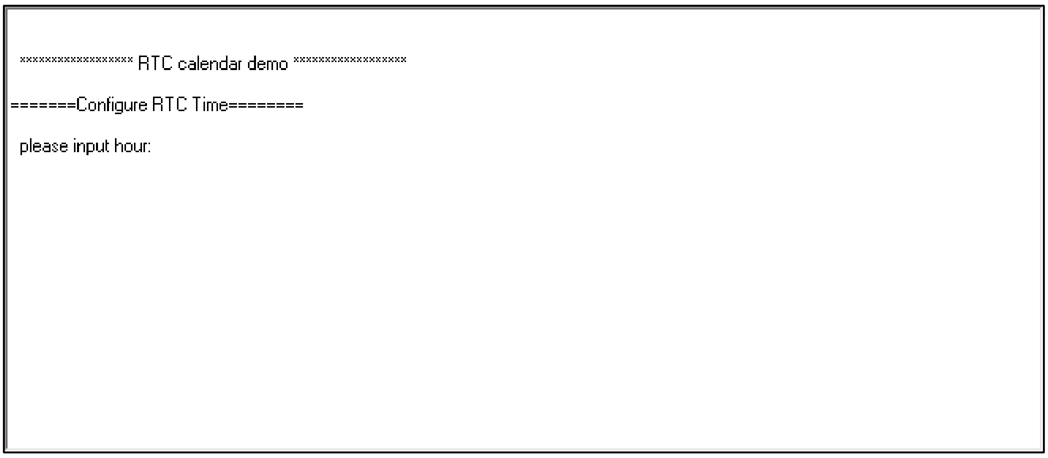
5.14.1. DEMO purpose

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

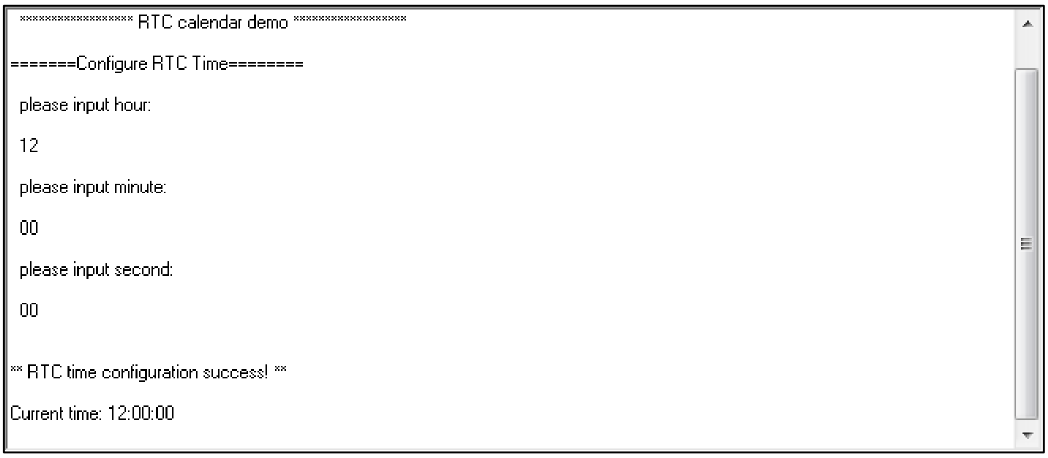
5.14.2. DEMO running result

Download the program <14_RTC_Calendar> to the EVAL board and run. If the development board run the program for the first time, serial port output following information "Configure RTC time" It requires the user to set up hours、minutes and seconds.



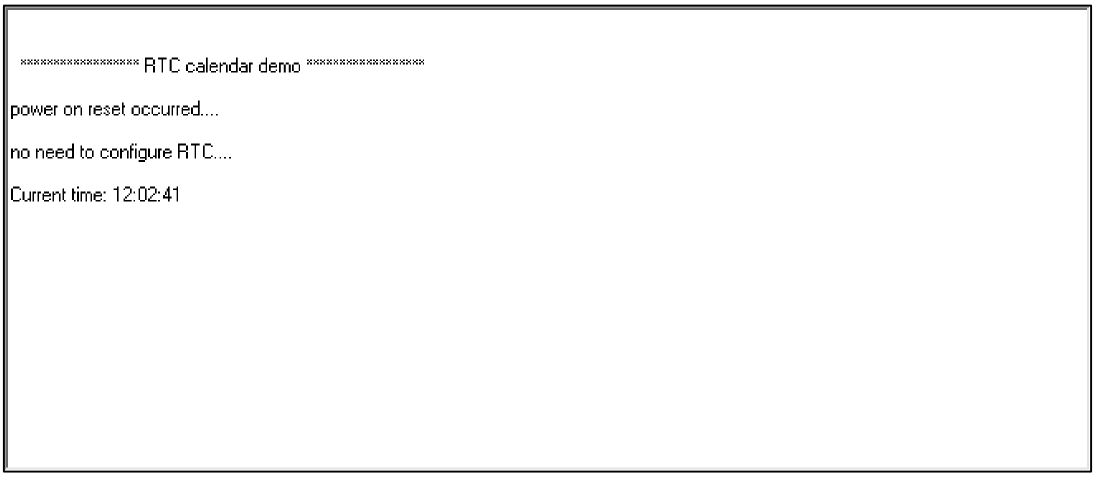
```
  ****************** RTC calendar demo ******************

=======Configure RTC Time========

  please input hour:
```

According to the serial port output information prompt, setting time, as shown below, serial port output following information.



```
  ****************** RTC calendar demo ******************

=======Configure RTC Time========

  please input hour:

  12

  please input minute:

  00

  please input second:

  00


** RTC time configuration success! **

Current time: 12:00:00
```

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, as shown below, serial port output following information " No need to configured RTC....", serial port continue printing time information.



```
 ****************** RTC calendar demo ******************
power on reset occurred....
no need to configure RTC....
Current time: 12:02:41
```

5.15. IRInfrared_Transceiver

5.15.1. DEMO purpose

This Demo includes the following functions of GD32 MCU:

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

5.15.2. DEMO running result

Download the program <15_IRInfrared_Transceiver> to the EVAL board and run. When the program is running, if the infrared receiver received data is correct, LED1, LED2, LED3, LED4 light in turn, otherwise LED1, LED2, LED3, LED4 toggle together.

5.16. TIMER_Breath_LED

5.16.1. DEMO purpose

This Demo includes the following functions of GD32 MCU:

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

5.16.2. DEMO running result

Download the program <16_TIMER_Breath_LED> to the GD32F310C-EVAL board and run. PA8 should not be reused by other peripherals.

When the program is running, you can see LED1 lighting from dark to bright gradually and then gradually darken, ad infinitum, just like breathing as rhythm.

6. Revision history

Table 6-1 Revision history

Revision No.	Description	Date
1.0	Initial Release	Mar.06, 2022

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.