

GigaDevice Semiconductor Inc.

GD32G553R-EVAL

Arm[®] Cortex[®]-M33 32-bit MCU

User Guide

Revision 1.0

(Nov. 2024)

Table of Contents

Table of Contents	1
List of Figures	4
List of Tables	5
1. Summary	6
2. Function Pin Assign	7
3. Getting started	9
4. Hardware layout overview	10
4.1. Power supply	10
4.2. Boot option	10
4.3. LED	10
4.4. KEY	11
4.5. ADC	11
4.6. CMP	11
4.7. CAN	12
4.8. USART	12
4.9. I2C	12
4.10. QSPI	13
4.11. SPI LCD	13
4.12. HPDF	13
4.13. IFRP	14
4.14. Extension	14
4.15. GD-Link	15
4.16. MCU	15
5. Routine use guide	16
5.1. GPIO_Running_LED	16
5.1.1. DEMO purpose	16
5.1.2. DEMO running result	16
5.2. GPIO_Key_Polling_mode	16
5.2.1. DEMO purpose	16
5.2.2. DEMO running result	16
5.3. EXTI_Key_Interrupt_mode	17
5.3.1. DEMO purpose	17
5.3.2. DEMO running result	17
5.4. USART_Printf	17
5.4.1. DEMO purpose	17
5.4.2. DEMO running result	17
5.5. USART_HyperTerminal_Interrupt	18
5.5.1. DEMO purpose	18

5.5.2.	DEMO running result	18
5.6.	USART_DMA	18
5.6.1.	DEMO purpose	18
5.6.2.	DEMO running result	18
5.7.	ADC_Temperature_Vrefint	19
5.7.1.	DEMO purpose	19
5.7.2.	DEMO running result	19
5.8.	Comparator_Obtain_Brightness	20
5.8.1.	DEMO purpose	20
5.8.2.	DEMO running result	20
5.9.	I2C_EEPROM	20
5.9.1.	DEMO purpose	20
5.9.2.	DEMO running result	20
5.10.	QSPI_Flash	21
5.10.1.	DEMO purpose	21
5.10.2.	DEMO running result	21
5.11.	SPI_TFT_LCD_Driver	22
5.11.1.	DEMO purpose	22
5.11.2.	DEMO running result	22
5.12.	TRNG_Get_Random	23
5.12.1.	DEMO purpose	23
5.12.2.	DEMO running result	23
5.13.	CAU	23
5.13.1.	DEMO purpose	23
5.13.2.	DEMO running result	23
5.14.	CAN_Network	25
5.14.1.	DEMO purpose	25
5.14.2.	DEMO running result	26
5.15.	RCU_Clock_Out	26
5.15.1.	DEMO purpose	26
5.15.2.	DEMO running result	26
5.16.	PMU_Sleep_Wakeup	27
5.16.1.	DEMO purpose	27
5.16.2.	DEMO running result	27
5.17.	RTC_Calendar	27
5.17.1.	DEMO purpose	27
5.17.2.	DEMO running result	27
5.18.	TIMER_Breath_LED	28
5.18.1.	DEMO purpose	28
5.18.2.	DEMO running result	28
5.19.	LPTIMER_Deepsleep_Pwmout	29
5.19.1.	DEMO purpose	29
5.19.2.	DEMO running result	29
5.20.	SHRTIMER_TIMER_Breath_LED	29

5.20.1.	DEMO purpose	29
5.20.2.	DEMO running result	29
5.21.	CLA_logic_AND_function_interrupt	30
5.21.1.	DEMO purpose	30
5.21.2.	DEMO running result	30
5.22.	TMU_Calculation	30
5.22.1.	DEMO purpose	30
5.22.2.	DEMO running result	30
5.23.	FAC_Calculation	31
5.23.1.	DEMO purpose	31
5.23.2.	DEMO running result	31
5.24.	FFT_Calculation	31
5.24.1.	DEMO purpose	31
5.24.2.	DEMO running result	31
5.25.	IRInfrared_Transceiver	32
5.25.1.	DEMO_purpose	32
5.25.2.	DEMO running result	32
6.	Revision history	33

List of Figures

Figure 4-1. Schematic diagram of power supply.....	10
Figure 4-2. Schematic diagram of boot option	10
Figure 4-3. Schematic diagram of LED function	10
Figure 4-4. Schematic diagram of Key function	11
Figure 4-5. Schematic diagram of ADC	11
Figure 4-6. Schematic diagram of CMP	11
Figure 4-7. Schematic diagram of CAN	12
Figure 4-8. Schematic diagram of USART	12
Figure 4-9. Schematic diagram of I2C	12
Figure 4-10. Schematic diagram of QSPI	13
Figure 4-11. Schematic diagram of SPI LCD	13
Figure 4-12. Schematic diagram of HPDF	13
Figure 4-13. Schematic diagram of IFRP.....	14
Figure 4-14. Schematic diagram of Extension.....	14
Figure 4-15. Schematic diagram of GD-Link.....	15
Figure 4-16. Schematic diagram of MCU.....	15

List of Tables

Table 2-1. Function pin assignment.....	7
Table 6-1. Revision history	33

1. Summary

GD32G553R-EVAL uses GD32G553RET6 as the main controller. It uses GD-Link Mini USB interface to supply 5V power. Reset, Boot, Button key, LED, CAN, I2C-EEPROM, SPI LCD, SPI Flash, HPDF, IFRP and USART to USB interface are also included. For more details, please refer to GD32G553R-EVAL schematic.

2. Function Pin Assign

Table 2-1. Function pin assignment

Function	Pin	Description
LED	PC9	LED1
	PA8	LED2
	PC12	LED3
	PC10	LED4
RESET		Reset
KEY	PA0	K2-Wakeup
	PC13	K3-Tamper
ADC	PA2	ADC01_IN1
CAN	PA11	CAN0_RX
	PA12	CAN0_TX
USART	PA9	USART0_TX
	PA10	USART0_RX
I2C	PB6	I2C0_SCL
	PB7	I2C0_SDA
QSPI	PA7	QSPI_IO2
	PA6	QSPI_IO3
	PB11	QSPI_CS
	PB10	QSPI_SCK
	PB2	QSPI_MISO
	PB1	QSPI_MOSI
SPI LCD	PB12	SPI1_CS
	PB13	SPI1_SCK
	PB14	SPI1_MISO
	PB15	SPI1_MOSI
	PA5	TFT_RESET
HPDF	PC1	HPDF_DTIN0
	PC3	HPDF_DTIN1
	PC5	HPDF_DTIN2
	PC7	HPDF_DTIN3
	PC0	HPDF_CKIN0
	PC2	HPDF_CKIN1
	PC4	HPDF_CKIN2
	PC6	HPDF_CKIN3
	PC11	HPDF_EXTI11
	PB0	HPDF_CKOUT
CMP	PA3	CMP_IP
IFRP	PB9	IFRP_OUT

3. Getting started

The EVAL board uses GD-Link Mini USB connector to get power DC +5V, which is the hardware system normal work voltage. A GD-Link on board is necessary in order to download and debug programs. Select the correct boot mode and then power on, the LED3 will turn on, which indicates the power supply is OK.

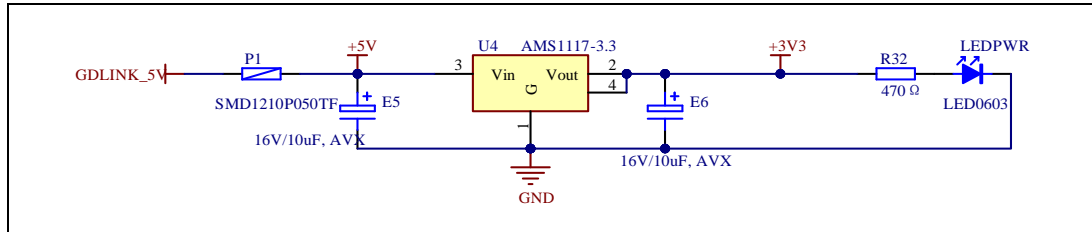
There are Keil version and IAR version of all projects. Keil version of the projects are created based on Keil MDK-ARM 5.29 uVision5. IAR version of the projects are created based on IAR Embedded Workbench for ARM 8.32.1. During use, the following points should be noted:

1. If you use Keil uVision5 to open the project. In order to solve the "Device Missing (s)" problem, the latest version of GigaDevice.GD32G5x3_DFP (URL: <https://www.gd32mcu.com>) should be installed to load related files.
2. If you use IAR to open the project, the latest version of IAR_GD32G5x3_ADDON (URL: <https://www.gd32mcu.com>) should be installed to load related files.

4. Hardware layout overview

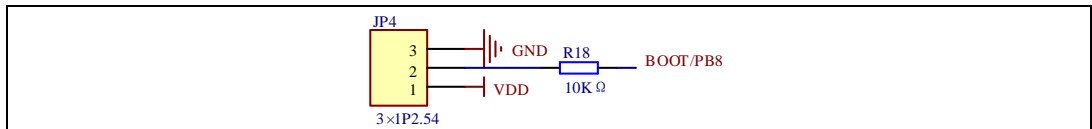
4.1. Power supply

Figure 4-1. Schematic diagram of power supply



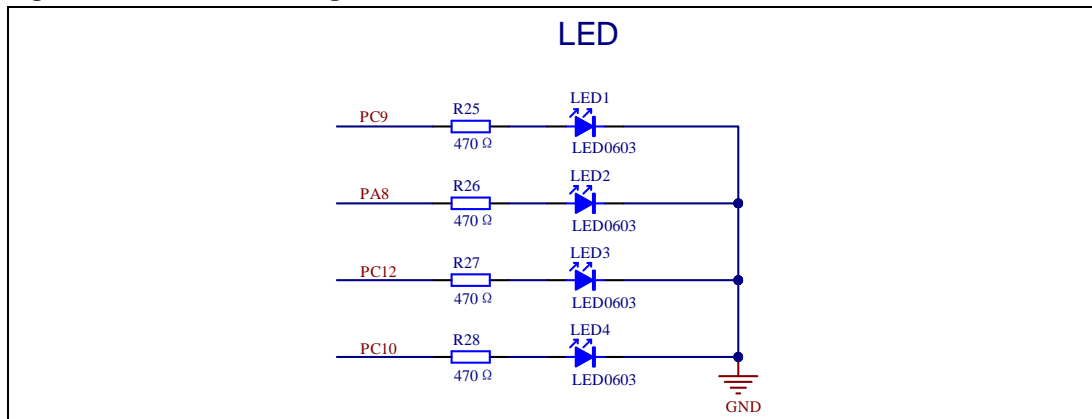
4.2. Boot option

Figure 4-2. Schematic diagram of boot option



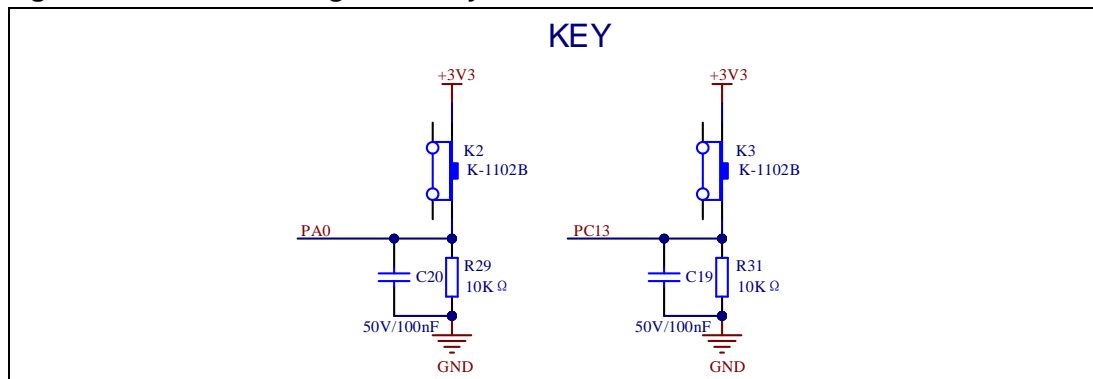
4.3. LED

Figure 4-3. Schematic diagram of LED function



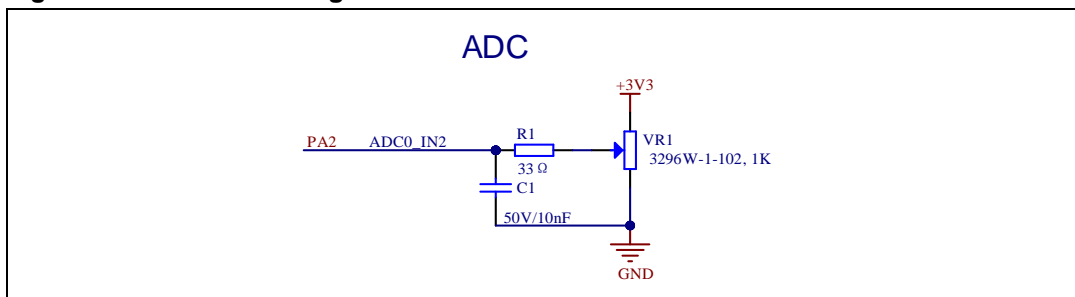
4.4. KEY

Figure 4-4. Schematic diagram of Key function



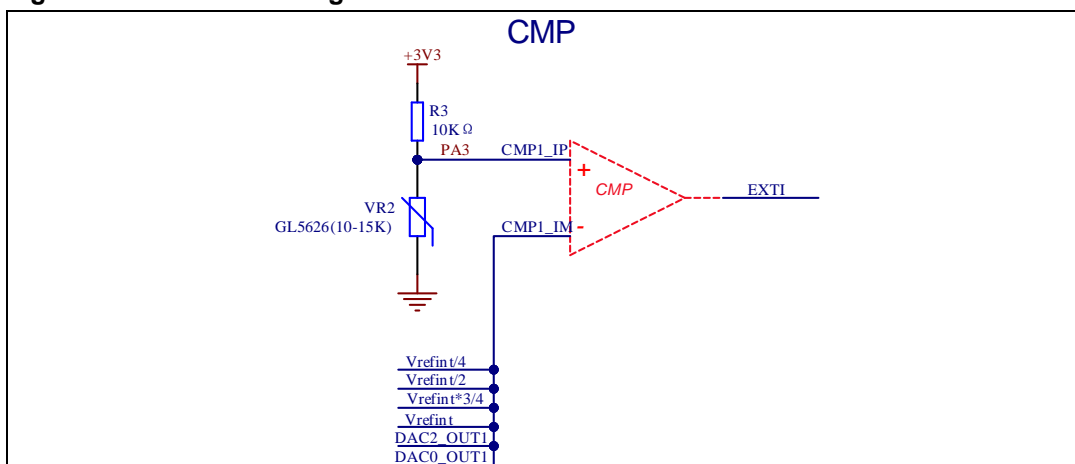
4.5. ADC

Figure 4-5. Schematic diagram of ADC



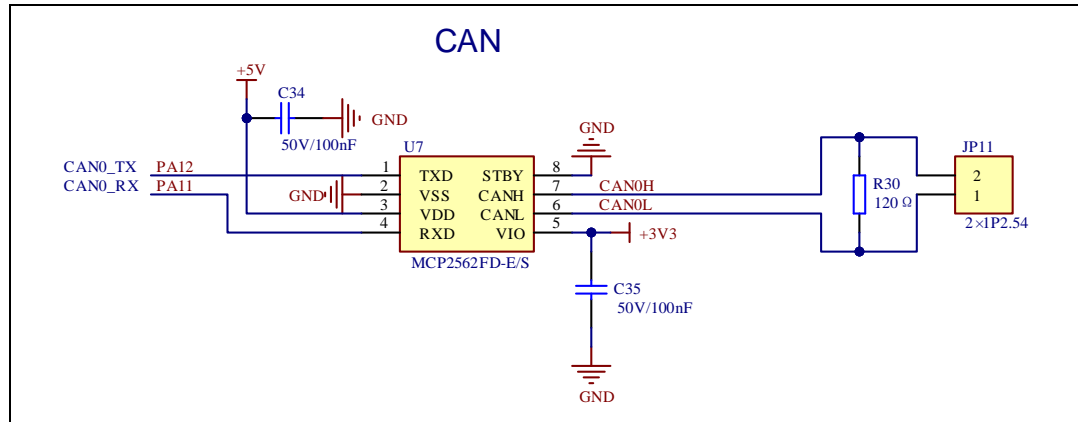
4.6. CMP

Figure 4-6. Schematic diagram of CMP



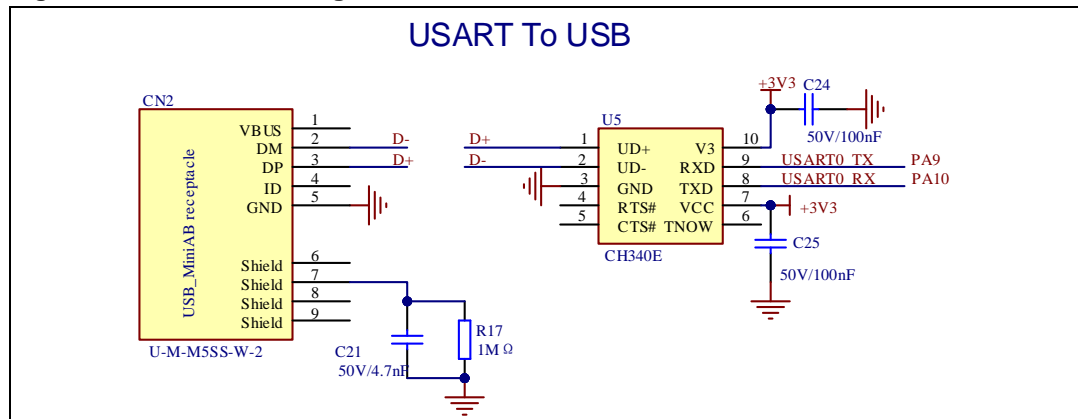
4.7. CAN

Figure 4-7. Schematic diagram of CAN



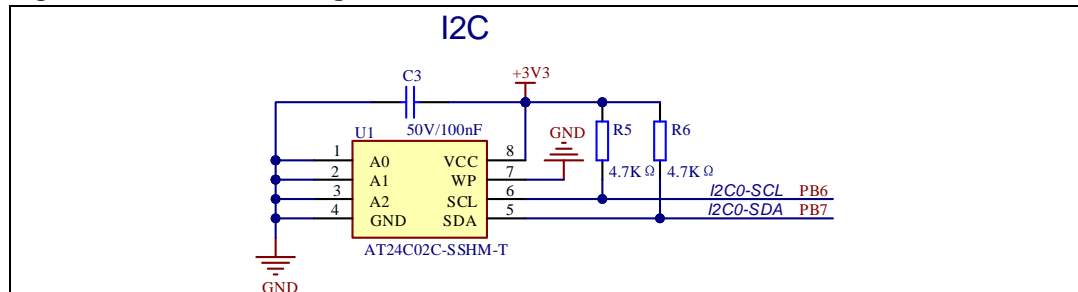
4.8. USART

Figure 4-8. Schematic diagram of USART



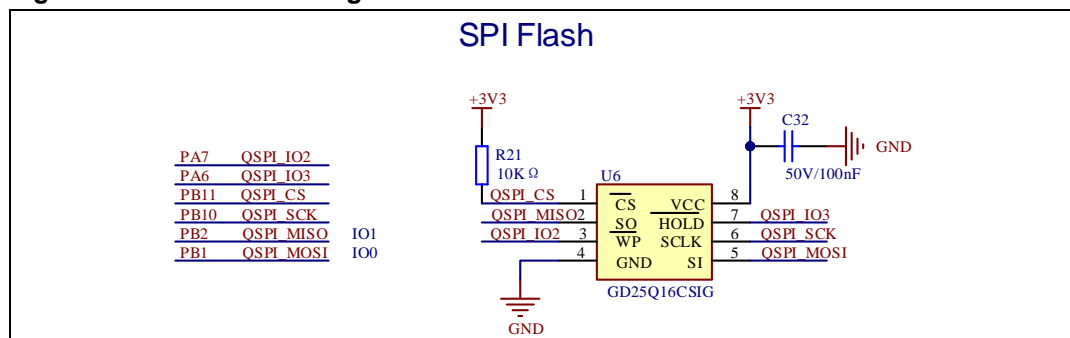
4.9. I2C

Figure 4-9. Schematic diagram of I2C



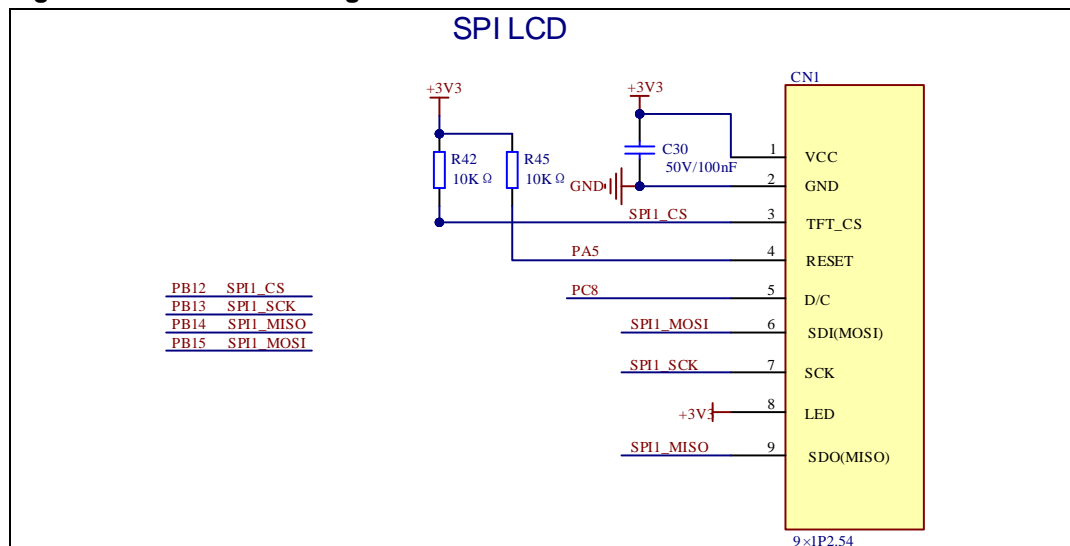
4.10. QSPI

Figure 4-10. Schematic diagram of QSPI



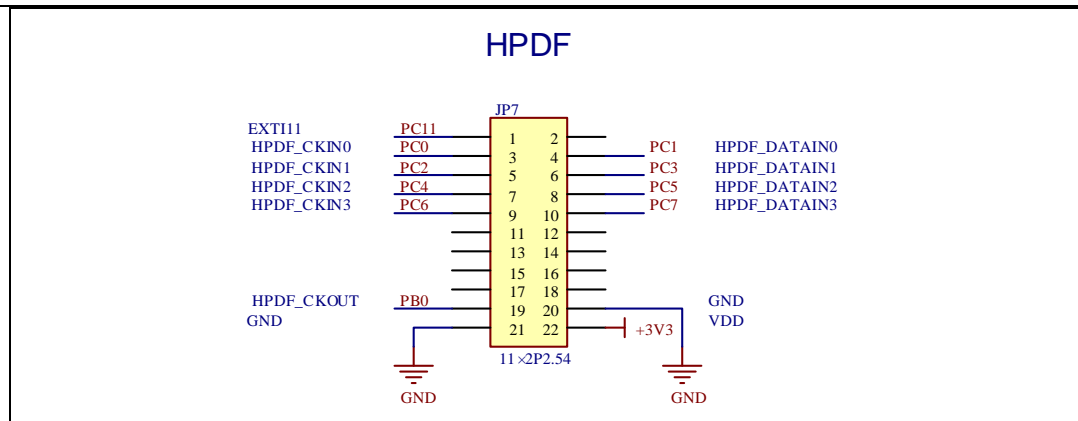
4.11. SPI LCD

Figure 4-11. Schematic diagram of SPI LCD



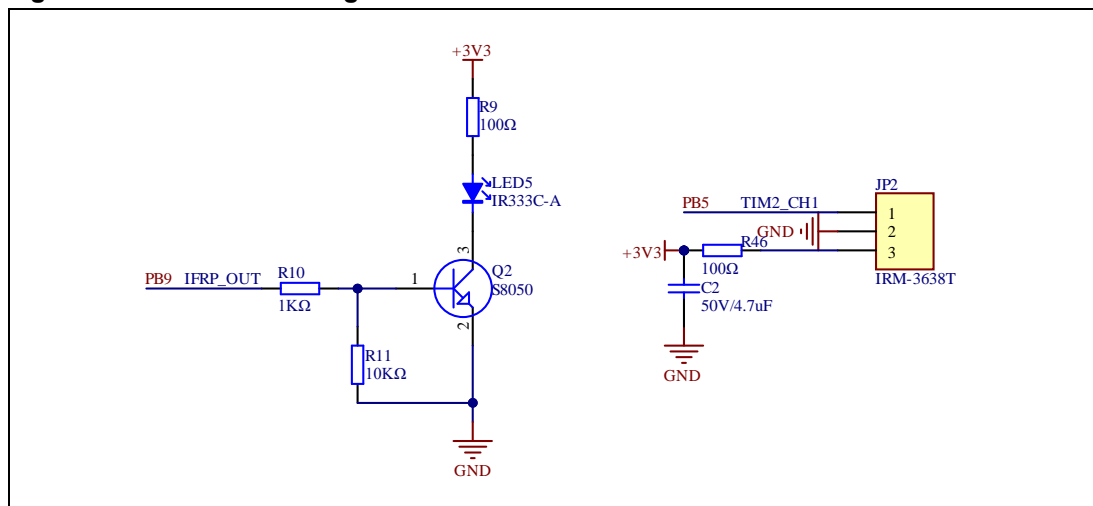
4.12. HPDF

Figure 4-12. Schematic diagram of HPDF



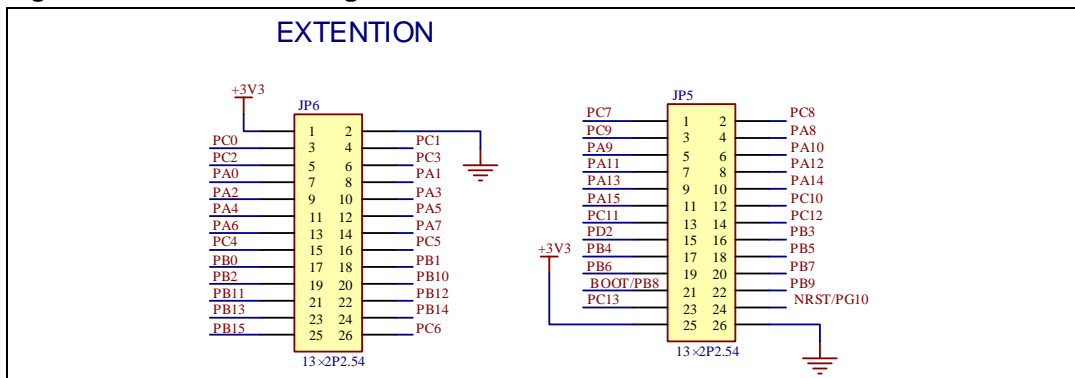
4.13. IFRP

Figure 4-13. Schematic diagram of IFRP



4.14. Extension

Figure 4-14. Schematic diagram of Extension



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

GD32G553R-EVAL board has two user keys and four LEDs. The keys are Tamper key, and Wakeup key. The LEDs are controlled by GPIO.

This demo will show how to light the LEDs.

5.1.2. DEMO running result

Download the program < 01_GPIO_Running_LED > to the EVAL board, two LEDs can light cycles.

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

GD32G553R-EVAL board has two user keys and four LEDs. The keys are Tamper key, and Wakeup key. The LEDs 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 1 and will wait for 100ms. Check the input value of the IO port again. If the value still is 1, it indicates that the button is pressed successfully and toggle LED2.

5.2.2. DEMO running result

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

5.3. EXTI_Key_Interrupt_mode

5.3.1. DEMO purpose

This demo includes the following functions of GD32 MCU:

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

GD32G553R-EVAL board has two user keys and four LEDs. The keys are Tamper key, and Wakeup key. The LEDs are controlled by GPIO.

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

5.3.2. DEMO running result

Download the program < 03_EXTI_Key_Interrupt_mode > to the EVAL board, LED2 is turned on and off for test. 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 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 USART. Firstly, all the LEDs flash 2 times for test. Then, this implementation outputs "USART printf example: please press the Tamper key" on the HyperTerminal using USART. Press the Tamper key, the serial port will output "USART printf example".

The output information via the HyperTerminal is as following:

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

USART printf example
```

5.5. USART_HyperTerminal_Interrupt

5.5.1. DEMO purpose

This demo includes the following functions of GD32 MCU:

- Learn to use the USART transmit and receive interrupts to communicate with the HyperTerminal.

5.5.2. DEMO running result

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

The output information via the HyperTerminal is as following:

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

5.6. USART_DMA

5.6.1. DEMO purpose

This demo includes the following functions of GD32 MCU:

- Learn to use the USART transmit and receive data using DMA.

5.6.2. DEMO running result

Download the program <06_USART_DMA> to the EVAL board and connect serial cable to USART. Firstly, the USART sends "USART DMA interrupt receive and transmit example, please input 32 bytes:" to hyperterminal and waits for receiving 32 bytes data from the hyperterminal that you must send. After MCU receives the data, the USART will continue to

output the received data to the hyper terminal.

The output information via the HyperTerminal is as following:

```
USART DMA interrupt receive and transmit example, please input 32 bytes:  
  
adadsddafafagagafgfgafgafgagadad
```

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 ADC0 inner channel 14 (temperature sensor channel) and ADC0 channel 18 (Vrefint channel)

5.7.2. DEMO running result

Download the program <07_ADC_Temperature_Vrefint> to the board. Connect serial cable to USART, open the HyperTerminal.

When the program is running, HyperTerminal display the value of temperature and internal voltage reference.

```
the temperature data is 35 degrees Celsius  
the reference voltage data is 1.201V  
  
the temperature data is 35 degrees Celsius  
the reference voltage data is 1.201V  
  
the temperature data is 35 degrees Celsius  
the reference voltage data is 1.201V  
  
the temperature data is 35 degrees Celsius  
the reference voltage data is 1.201V  
  
the temperature data is 35 degrees Celsius  
the reference voltage data is 1.201V  
  
the temperature data is 35 degrees Celsius  
the reference voltage data is 1.201V  
  
the temperature data is 35 degrees Celsius  
the reference voltage data is 1.202V
```

5.8. Comparator_Obtain_Brightness

5.8.1. DEMO purpose

This Demo includes the following functions of GD32 MCU:

- Learn to use comparator output compare result

The comparator has two inputs, in this demo, one input is PA3, and the other one is the reference voltage. Compare the two input voltages, the output is a high or low level, and the LED2 will performs the corresponding action.

5.8.2. DEMO running result

Download the program <08_Comparator_Obtain_Brightness> to the EVAL board, comparing two input voltage, if output level is high, LED2 is on, otherwise LED2 is off.

5.9. I2C_EEPROM

5.9.1. 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

5.9.2. DEMO running result

Download the program <09_I2C_EEPROM> to the EVAL board and run. Connect serial cable to USART, 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 two LEDs lights flashing, otherwise the serial port will output "Err:data read and write aren't matching." and all the two LEDs light.

The output information via the serial port is as following.

```

I2C-24C02 configured...

The I2C is hardware interface
The speed is 400K
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!

```

5.10. QSPI_Flash

5.10.1. DEMO purpose

This demo includes the following functions of GD32 MCU:

- Learn to use QSPI to write and read spi flash data

5.10.2. DEMO running result

Download the program <10_QSPI_Flash> to the EVAL board and run. Connect serial cable to USART, open the HyperTerminal to show the print message. If the data read from the flash is the same with the data write to the flash, the USART will print "SPI FLASH WRITE AND READ TEST SUCCESS!". Otherwise, the USART will print "SPI FLASH WRITE AND READ TEST ERROR!".

The output information via the serial port is as following.

```
SSCOM 3.3
QSPI flash writing...
QSPI flash reading...
QSPI FLASH WRITE AND READ TEST SUCCESS!
```

5.11. SPI_TFT_LCD_Driver

5.11.1. DEMO purpose

This Demo includes the following function of GD32 MCU:

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

GD32G553R-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.

5.11.2. DEMO running result

LCD is controlled by SPI module on GD32G553R-EVAL board. Download the program <11_SPI_TFT_LCD_Driver> to the EVAL board. All the LEDs are turned on and then turned off for test. After that, the LCD screen on the board will display the GUI tests in infinite loop.



5.12. TRNG_Get_Random

5.12.1. 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

5.12.2. DEMO running result

Download the program <12_TRNG_Get_Random> to the EVAL board and run. Connect serial cable to USART, 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 0x011, the maximum value is 0x33), 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 successful
Please input min num (hex format):
Please input max num (hex format):
Input min num is 17
Input max num is 51
Generate random num1 is 42
Generate random num2 is 35
Please input min num (hex format):

```

5.13. CAU

5.13.1. DEMO purpose

This demo includes the following functions of GD32 MCU:

- Learn DES, Triple-DES and AES algorithm.
- Learn Electronic codebook (ECB) mode, Cipher block chaining (CBC) mode, Counter (CTR) mode, Galois/counter (GCM) mode, combined cipher machine (CCM) mode, Cipher Feedback (CFB) mode, and Output Feedback (OFB) mode.
- Learn to use CAU to encrypt and decrypt.
- Learn to communicate with PC by USART.

5.13.2. DEMO running result

Download the program <13_CAU> to the EVAL board and run. When the program is running,

the serial terminal tool will display the information, as shown in the following figure. Plaintext data value, the encryption algorithm, and the mode can be selected are shown. After the user setting the algorithm and mode according to the serial output information indicating, serial port will print out selected algorithm and mode, as shown below.

```

Plain data :
0x30 0x31 0x32 0x33 0x34 0x35 0x36 0x37
0x38 0x39 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 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 0x7A 0x7A
=====Choose CAU algorithm=====
1: DES algorithm
2: TDES algorithm
3: AES algorithm

You choose to use DES algorithm
=====Choose CAU mode=====
1: ECB mode
2: CBC mode
3: CTR mode only when choose AES algorithm
4: GCM mode only when choose AES algorithm
5: CCM mode only when choose AES algorithm
6: CFB mode only when choose AES algorithm
7: OFB mode only when choose AES algorithm

You choose to use ECB mode

```

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

```
Encrypted data with DES Mode ECB :

0xB3 0x9F 0xBD 0x94 0xC4 0xE7 0xC2 0xAA
0x2F 0x5E 0xDE 0x61 0x21 0x36 0x36 0x62
0x61 0x84 0xF8 0xCA 0x4D 0x4E 0x55 0x14
0x93 0x08 0xFC 0xE4 0x82 0x65 0x48 0x8F
0xC6 0x02 0x1C 0xAD 0xF9 0xA0 0xEB 0x51
0x3C 0x29 0xEF 0x55 0xDB 0x15 0x15 0x8F
0x6E 0x5E 0x78 0xAA 0x61 0xDD 0xEB 0xA6
0x2A 0xDA 0xBA 0x87 0x6C 0xD3 0xB1 0x23

Decrypted data with DES Mode ECB :

0x30 0x31 0x32 0x33 0x34 0x35 0x36 0x37
0x38 0x39 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 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 0x7A 0x7A
```

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

```
Example restarted...

Plain data :
0x30 0x31 0x32 0x33 0x34 0x35 0x36 0x37
0x38 0x39 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 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 0x7A 0x7A
=====Choose CAU algorithm=====
1: DES algorithm
2: TDES algorithm
3: AES algorithm
```

5.14. CAN_Network

5.14.1. DEMO purpose

This demo includes the following functions of GD32 MCU:

- Learn to use the CAN communication between two boards

The EVAL development board integrates the CAN (Controller Area Network) bus controller,

which is a common industrial control bus. CAN bus controller supports the CAN 2.0A/B protocol, ISO 11898-1:2015 and BOSCH CAN-FD specification. This demo mainly shows how to communicate two EVAL boards through CAN.

5.14.2. DEMO running result

This example is tested with two EVAL boards. Connect L pin to L pin and H pin to H pin of JP11. Download the program <14_CAN_Network> to the two EVAL boards, and connect serial cable to EVAL_COM. Firstly, the EVAL_COM sends “please press the WAKEUP key to transmit data!” to the HyperTerminal. The frames are sent and the transmit data are printed by pressing WAKEUP Key push button. When the frames are received, the receive data will be printed and the LED1 will toggle one time. The output information via the serial port is as following.

```
communication test CAN0, please press WAKEUP key to start!

CAN0 transmit data:
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

CAN0 receive data:
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
```

5.15. RCU_Clock_Out

5.15.1. 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

5.15.2. DEMO running result

Download the program <15_RCU_Clock_Out> to the EVAL board and run. Connect serial cable to USART, 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 key. After pressing, the corresponding LED will be turned on and HyperTerminal will display which mode be selected. The frequency of the output clock can be observed through the oscilloscope by PA8 pin.

Information via a serial port output as following:

```
/===== Gigadevice clock output demo =====/  
press key tamper to select clock output source  
CK_OUT: system clock, DIV: 4  
CK_OUT: IRC8M, DIV: 1  
CK_OUT: HXTAL, DIV: 1  
CK_OUT: LXTAL, DIV: 1  
CK_OUT: PLLP, DIV: 4  
CK_OUT: IRC32K, DIV: 1
```

5.16. PMU_Sleep_Wakeup

5.16.1. 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

5.16.2. DEMO running result

Download the program < 16_PMU_sleep_wakeup > to the EVAL board, connect serial cable to USART. After power-on, all the LEDs are off. The MCU will enter sleep mode and the software stop running. When the USART receives a byte of data from the HyperTerminal, the MCU will wake up from a receive interrupt. And the LED1 and LED2 will flash together.

5.17. RTC_Calendar

5.17.1. 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

5.17.2. DEMO running result

Download the program <17_RTC_Calendar> to the EVAL board and run. Connect serial cable to USART, 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.

```
***** RTC calendar demo *****  
  
=====Configure RTC Time=====  
  
please set the last two digits of current year:  
  
2021  
  
please input month:  
  
08  
  
please input day:  
  
12  
  
please input hour:  
  
12  
  
please input minute:  
  
12  
  
please input second:  
  
12  
  
** RTC time configuration success! **  
  
Current time: 2021-08-12 : 12:12:12  
Current time: 2021-08-12 : 12:12:12
```

5.18. TIMER_Breath_LED

5.18.1. DEMO purpose

This Demo includes the following functions of GD32 MCU:

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

5.18.2. DEMO running result

Download the program <18_TIMER_Breath_LED> to the GD32G553R-EVAL board and run.

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

5.19. LPTIMER_Deepsleep_Pwmout

5.19.1. DEMO purpose

This Demo includes the following functions of GD32 MCU:

- Learn to use LPTIMER output PWM wave
- Learn to use the LPTIMER interrupts to wake up the PMU from sleep mode

5.19.2. DEMO running result

Download the program <19_LPTIMER_Deepsleep_Pwmout> to the GD32G553R-EVAL board and run. When the program is running, you can see LED1 sparks and the LPTIMER0_O (PA4) outputs the PWM signal.

Press KEY_A to enter deepsleep mode, LED1 stops in a certain status (on or off). When the LPTIMER0 count value matches the value of compare register or auto reload register, MCU will be wakeup from deepsleep mode by LPTIMER0, the transfer goes on normally and LED1 sparks again. During this period, LPTIMER0_O (PC1) always outputs the PWM signal.

5.20. SHRTIMER_TIMER_Breath_LED

5.20.1. DEMO purpose

This demo includes the following functions of GD32 MCU:

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

5.20.2. DEMO running result

Use the DuPont line to connect the TIMER0_CH0 (PA8) and LED2 (PA8). Use the DuPont line to connect the SHRTIMER_ST0CH1 (PA9) and LED1 (PC9). Then download the program <20_SHRTIMER_TIMER_Breath_LED> to the EVAL board and run. PA8/PA9 should not be reused by other peripherals.

When the program is running, you can see LED1 and LED2 lighting from dark to bright gradually and then gradually darken, ad infinitum, just like breathing as rhythm.

5.21. CLA_logic_AND_function_interrupt

5.21.1. DEMO purpose

This Demo includes the following functions of GD32 MCU:

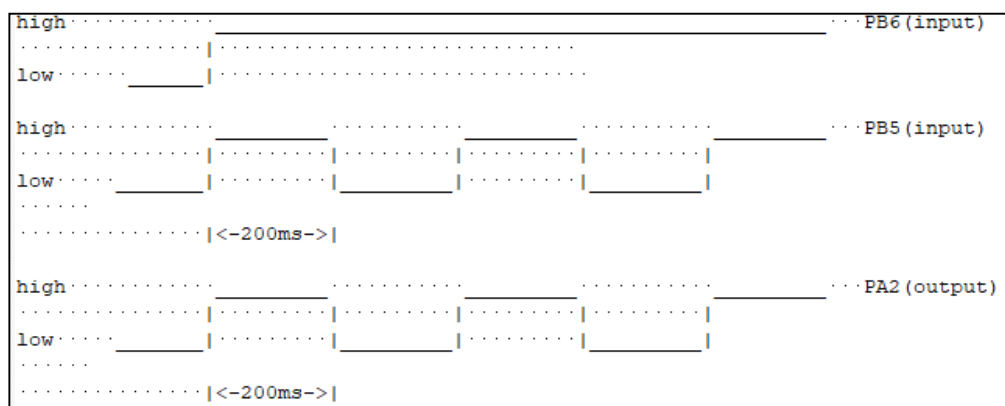
- Learn to use CLA logic AND function

In this demo, PB6 and PB5 is selected as input of SIGS0 and SIGS1 respectively. PA2 is used as CLA0OUT. PB6 outputs HIGH while PB5 toggled every 200ms.

5.21.2. DEMO running result

Download the program < 21_CLA_logic_AND_function_interrupt > to the START board, LED1 will be toggled every 200ms.

Connect the PB5, PB6 and PA2 to an oscilloscope to monitor waveform. The information via a serial port output like this:



5.22. TMU_Calculation

5.22.1. DEMO purpose

This demo includes the following functions of GD32 MCU:

- Learn to use TMU operation mode for calculation.
- Learn to use USART module to communicate with the HyperTerminal.

5.22.2. DEMO running result

Download the program <22_TMU_calculation> to the EVAL board. In this demo, the operation mode of the TMU is configured as mode 0. The input value and output value are in IEEE 32-bit single precision floating-point format. If no overflow error occurs, read the output data and open LED3 and LED4, otherwise open LED1 and LED2.

Use the HyperTerminal to enter the input value. The first input data is θ/π , and the range is $(-2^{24}, 2^{24})$. The second input data is m . If no overflow error occurs, the result of calculation will be printed by `usart0`. For example, if the first input data is 0.25 and the second input data is still 0.25, the result is 0.177.

```
TMU Caculation Test: m*cos(theta)
```

```
Please input the first value: theta/pi (-2^24, 2^24)
```

```
Please input the second value: m
```

```
The TMU calculation result is:      0.177
```

5.23. FAC_Calculation

5.23.1. DEMO purpose

This demo includes the following functions of GD32 MCU:

- Learn to configure FAC module.
- Learn to use this module to filter input datas.

5.23.2. DEMO running result

Download the program <23_FAC_Calculation> to the EVAL board and run. When the FAC calculation is over, the FAC filter data can be seen from corresponding array.

5.24. FFT_Calculation

5.24.1. DEMO purpose

This demo includes the following functions of GD32 MCU:

- Learn to configure FFT module.
- Learn to use this module to calculate FFT transformation for input data.

5.24.2. DEMO running result

Download the program <24_FFT_Calculation> to the EVAL board and run. When the program is running, the FFT transformation result of the input data will be printed out through the serial port, as shown below:

```
-----
FFT calculation completion!
-----
```

```
FFT calculation result:
```

```
5.474439+(0.000000i)
5.584536+(-0.281510i)
5.948057+(-0.598801i)
6.695495+(-1.009116i)
8.216001+(-1.648507i)
12.045313+(-3.022722i)
35.120644+(-10.640221i)
-18.522923+(6.675328i)
-4.914906+(2.140441i)
-1.545752+(0.927651i)
0.243159+(0.215375i)
1.665232+(-0.404974i)
-----
```

5.25. IRInfrared_Transceiver

5.25.1. 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

5.25.2. DEMO running result

Download the program < 25_IRInfrared_Transceiver > to the EVAL board and run. When the program is running, if the infrared receiver received data is correct, LED1, LED2, LED3 light in turn, otherwise LED1, LED2, LED3 toggle together.

6. Revision history

Table 6-1. Revision history

Revision No.	Description	Date
1.0	Initial Release	Nov.8, 2024

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 its 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 its suppliers and/or distributors harmless from and against all claims, costs, damages, and other liabilities, including claims for personal injury or death, arising from or related to any Unintended Uses of the 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.