GigaDevice Semiconductor Inc.

GD32190R-EVAL
User Manual

Table of Contents

目录

List of Tables

1 Summary

GD32190R-EVAL uses GD32F190R8T6 as the main controller. It uses Mini USB interface to supply 5V power. SWD, Reset, Boot, User button key, LED, CAN, I2C, I2S, USART, RTC, LCD, SPI, CEC, ADC, DAC, TSI, OPAMP and Extension Pins are also included. For more details please refer to GD32190R-EVAL-V1.1 schematic.

2 Function Pin Assign

Table 1 Function pin assign

Function	Pin	Description
LED	PA11	LED1
	PA12	LED2
	PB6	LED3
	PB7	LED4
RESET		K1-Reset
KEY	PC13	KEY2
	PA0	KEY3
IFRR	PB9	IR_OUT
	PA7	IR_IN
CMP	PA1	CMP1
HDMI_CEC	PB8	CEC
CAN	PA5	CAN1_H
	PA6	CAN1_L
	PB12	CAN2_RX
	PB13	CAN2_TX
I2C	PB10	I2C2_SCL
	PB11	I2C2_SDA
I2S	PC10	I2S3_CK
	PC11	I2S3_MCK
	PC12	I2S3_SD
	PA15	I2S3_WS
OPAMP	PC1	OPAMP3_VINP
	PC2	OPAMP3_VINM
	PC3	OPAMP3_VOUT
USART2	PA2	USART2_TX
	PA3	USART2_RX
LCD	PA8	LCD_COM0
	PA9	LCD_COM1

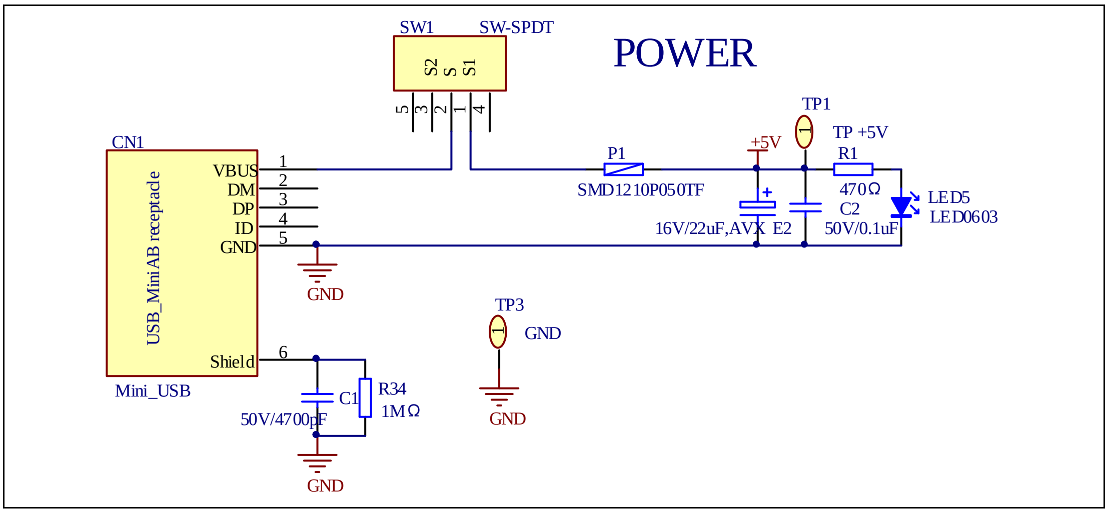
	PA10	LCD_COM2
	PB9	LCD_COM3
	PB14	LCD_SEG14
	PB15	LCD_SEG15
	PC4	LCD_SEG22
	PC5	LCD_SEG23
	PC6	LCD_SEG24
	PC7	LCD_SEG25
	PC8	LCD_SEG26
	PC9	LCD_SEG27
	PF4	LCD_SEG28
	PF5	LCD_SEG29
	PF6	LCD_SEG30
	PF7	LCD_SEG31
SPI	PB3	SPI1_SCK
	PB4	SPI1_MISO
	PB5	SPI1_MOSI
	PD2	SPI_CS
ADC	PC0	ADC_IN10
DAC	PA4	DAC_OUT1
	PA5	DAC_OUT2
TSI	PB0	TSI_G3_IO2
	PB1	TSI_G3_IO3
	PB2	TSI_G3_IO4

3 Getting started

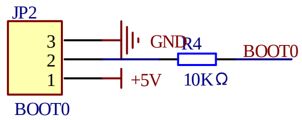
The EVAL board uses Mini USB connecter to get power DC +5V, which is the hardware system normal work voltage. A J-Link tool are necessary in order to download programs. Select the correct boot mode and then power on, the LED5 will turn on, which indicates that the power supply is OK.

4 Hardware layout overview

4.1 Power

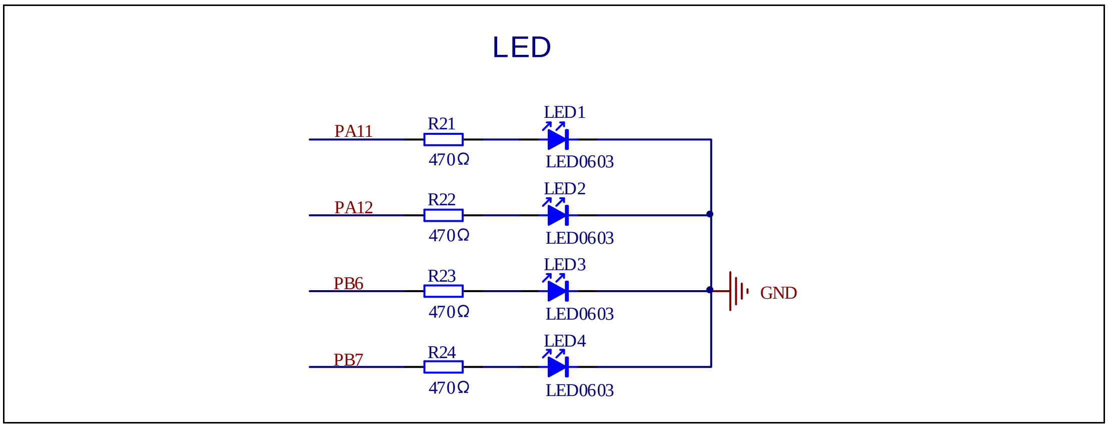


4.2 Boot

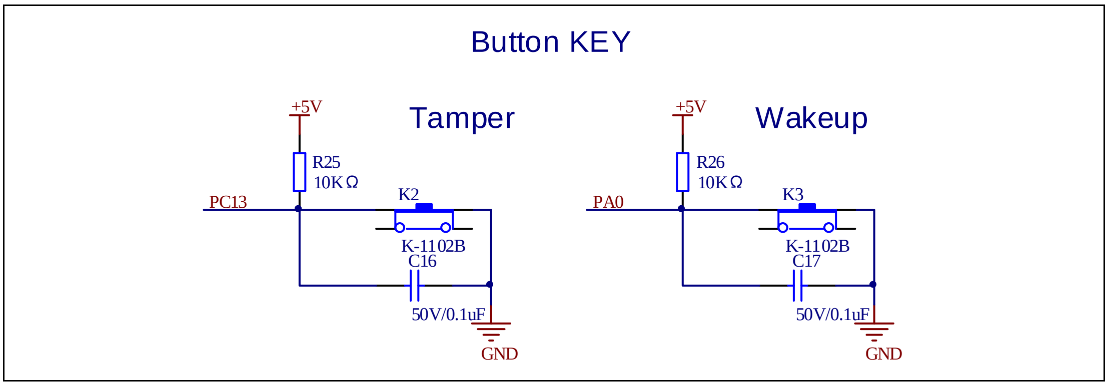


BOOT1(decide by Option byte)	BOOT0	Boot Mode
Any	2-3	User memory
2-3	1-2	System memory
1-2	1-2	SRAM memory

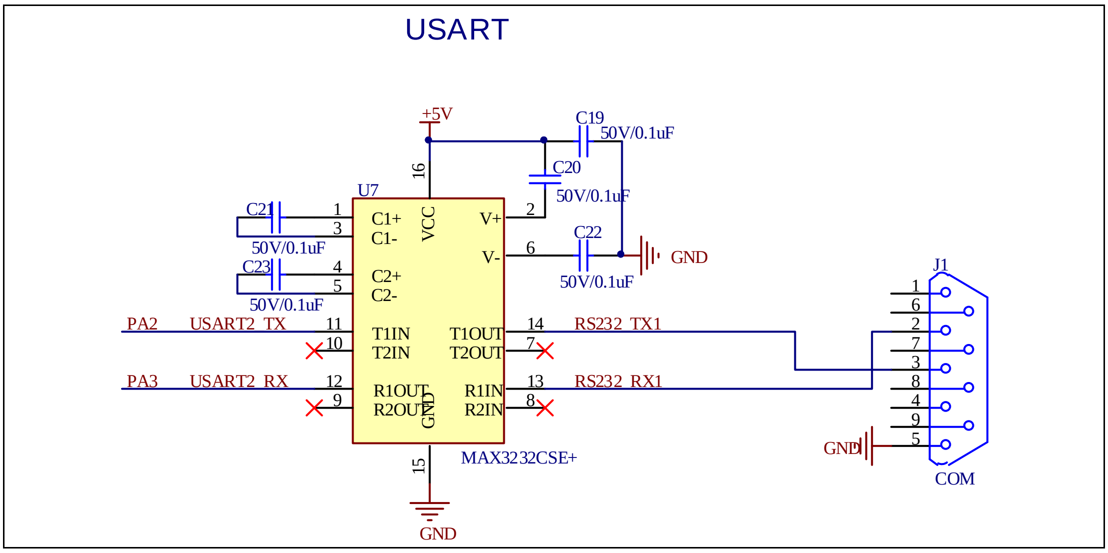
4.3 LED


LED
PA11
PA12
PB6
PB7
R21
R22
R23
R24
470Ω
470Ω
470Ω
470Ω
LED1
LED2
LED3
LED4
LED0603
LED0603
LED0603
LED0603
GND

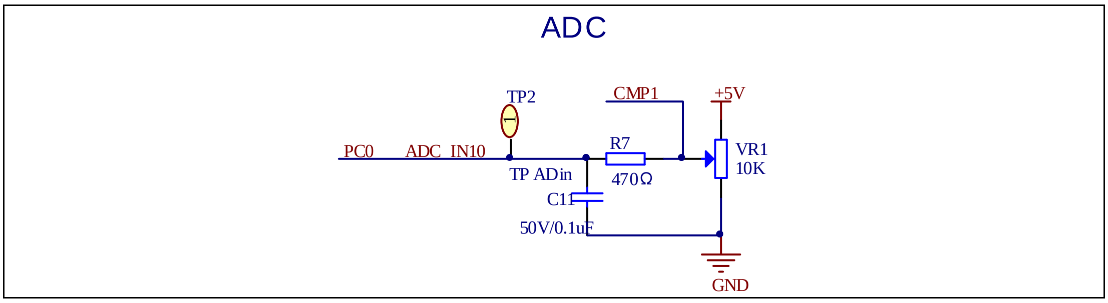
4.4 KEY


Button KEY
Tamper
Wakeup
+5V
+5V
R25
10KΩ
R26
10KΩ
PC13
PA0
K2
K3
K-1102B
K-1102B
C16
C17
50V/0.1uF
50V/0.1uF
GND
GND

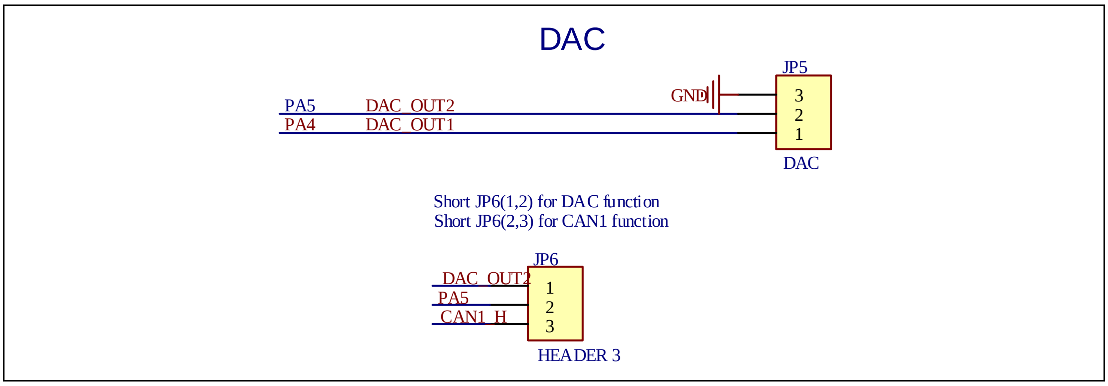
4.5 USART2



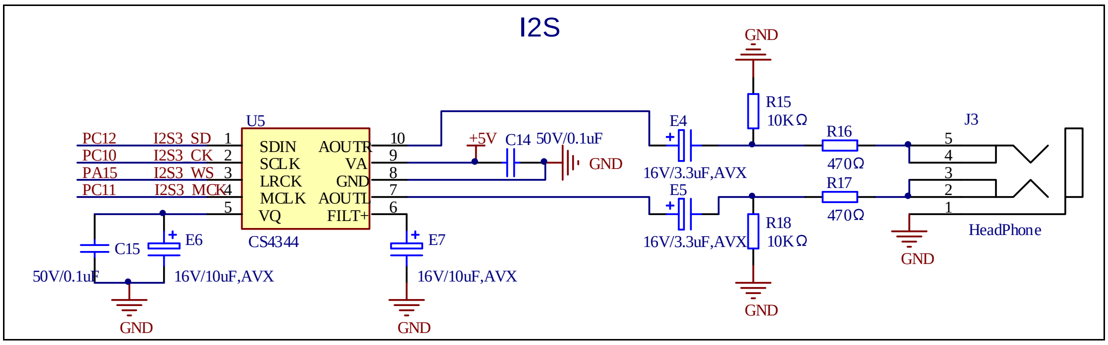
4.6 ADC



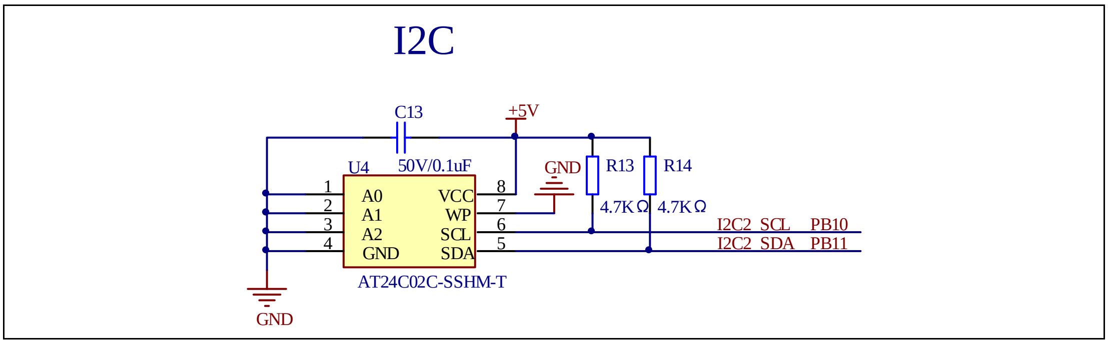
4.7 DAC



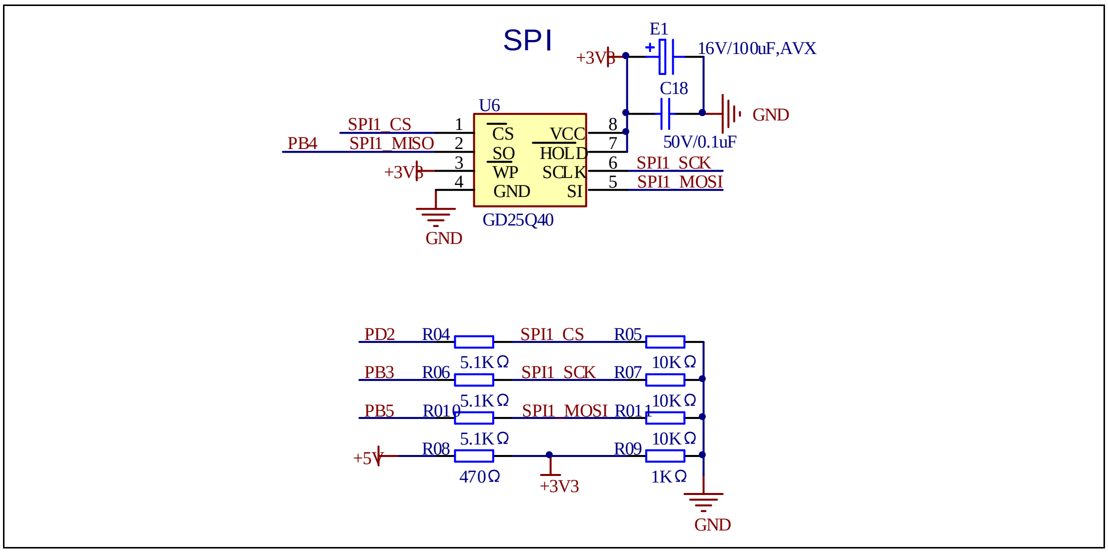
4.8 I2S



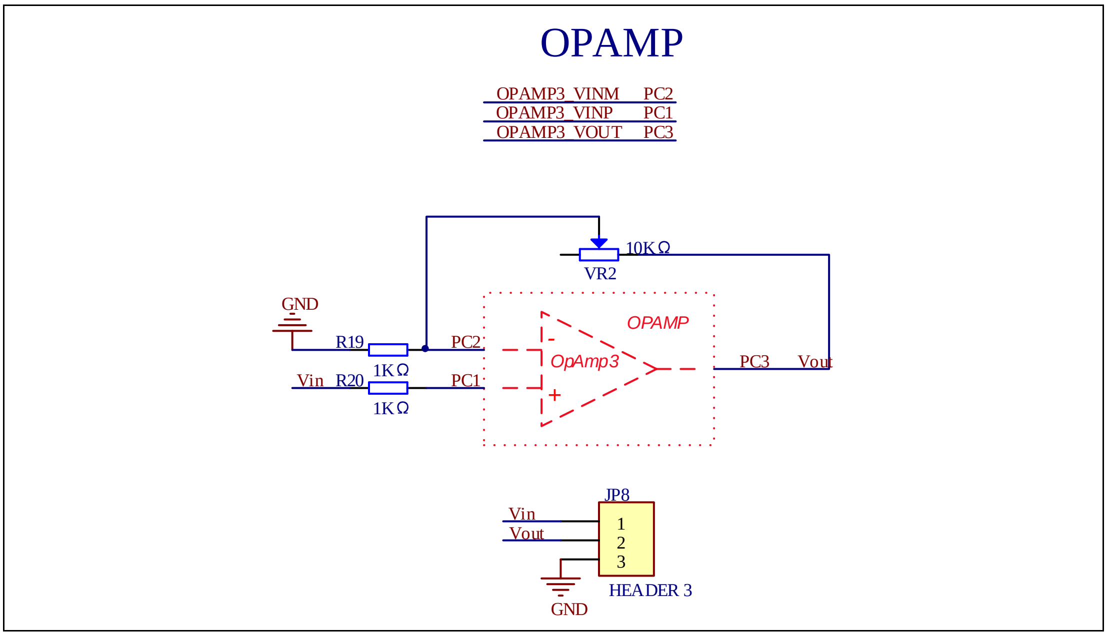
4.9 I2C



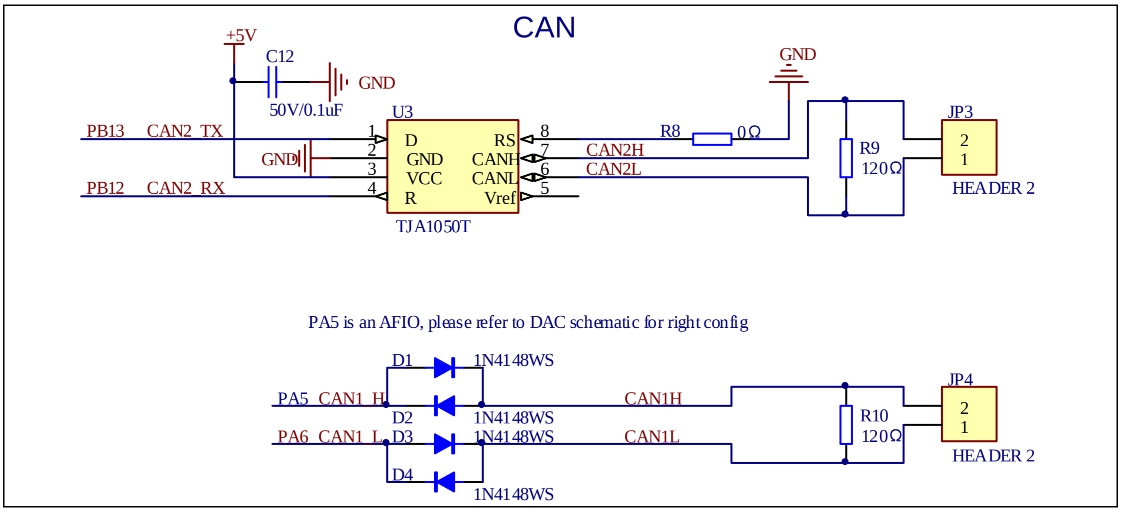
4.10 SPI



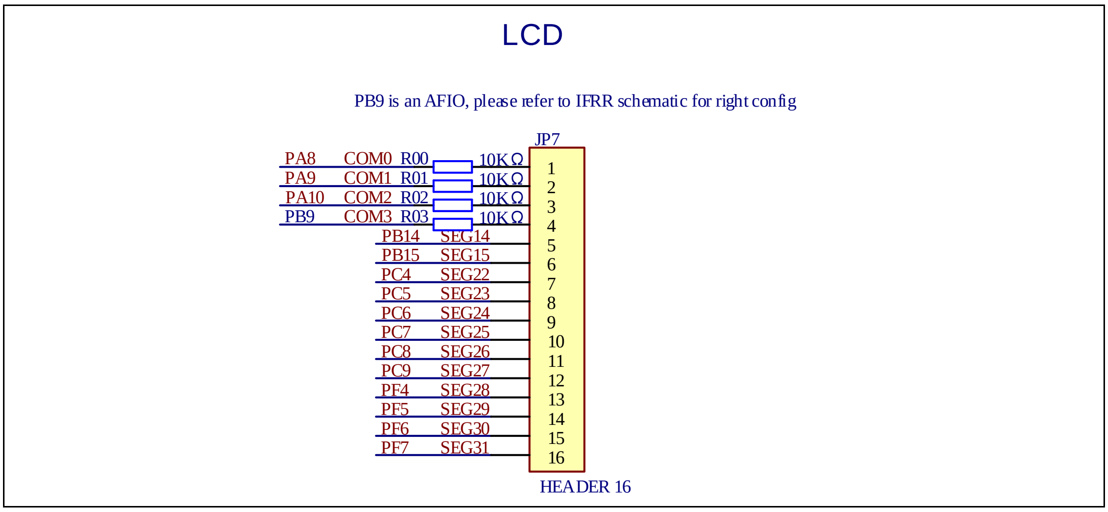
4.11 OPAMP



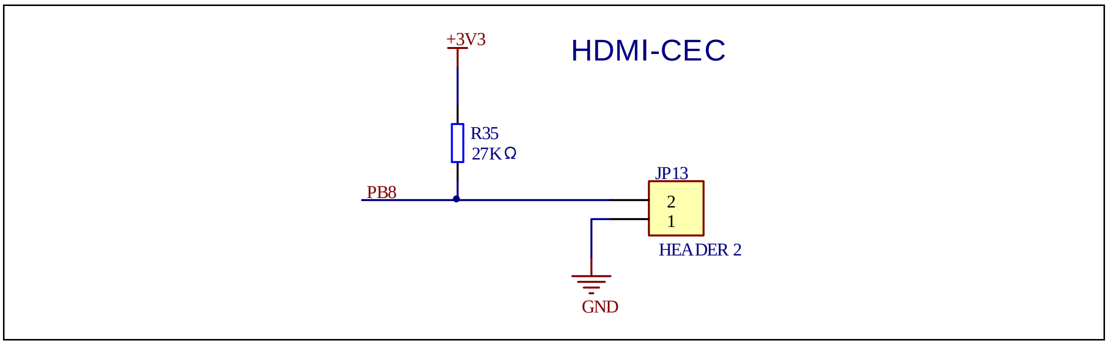
4.12 CAN



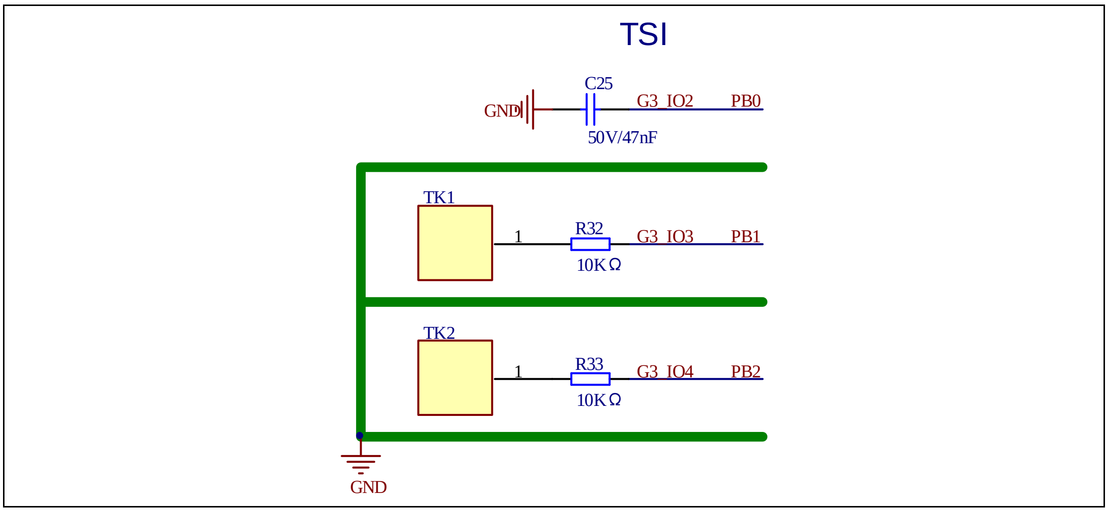
4.13 LCD



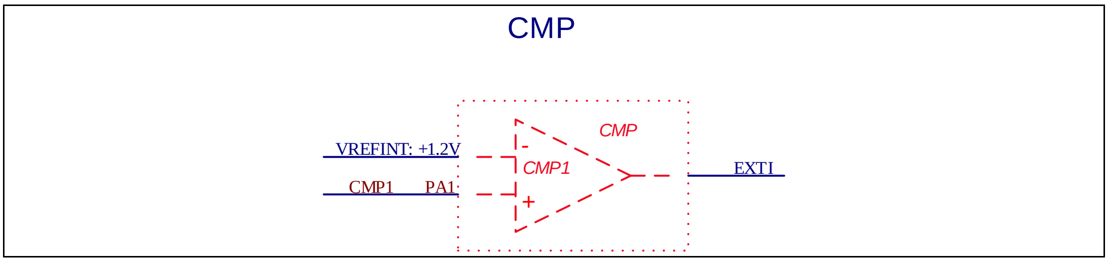
4.14 HDMI-CEC



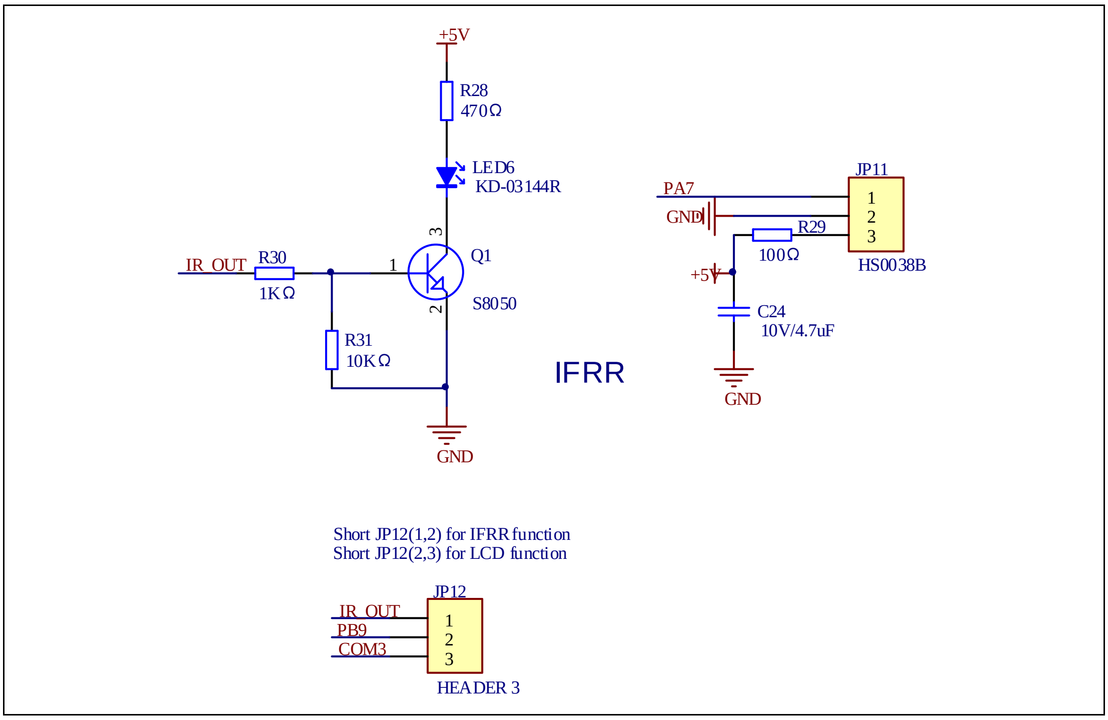
4.15 TSI



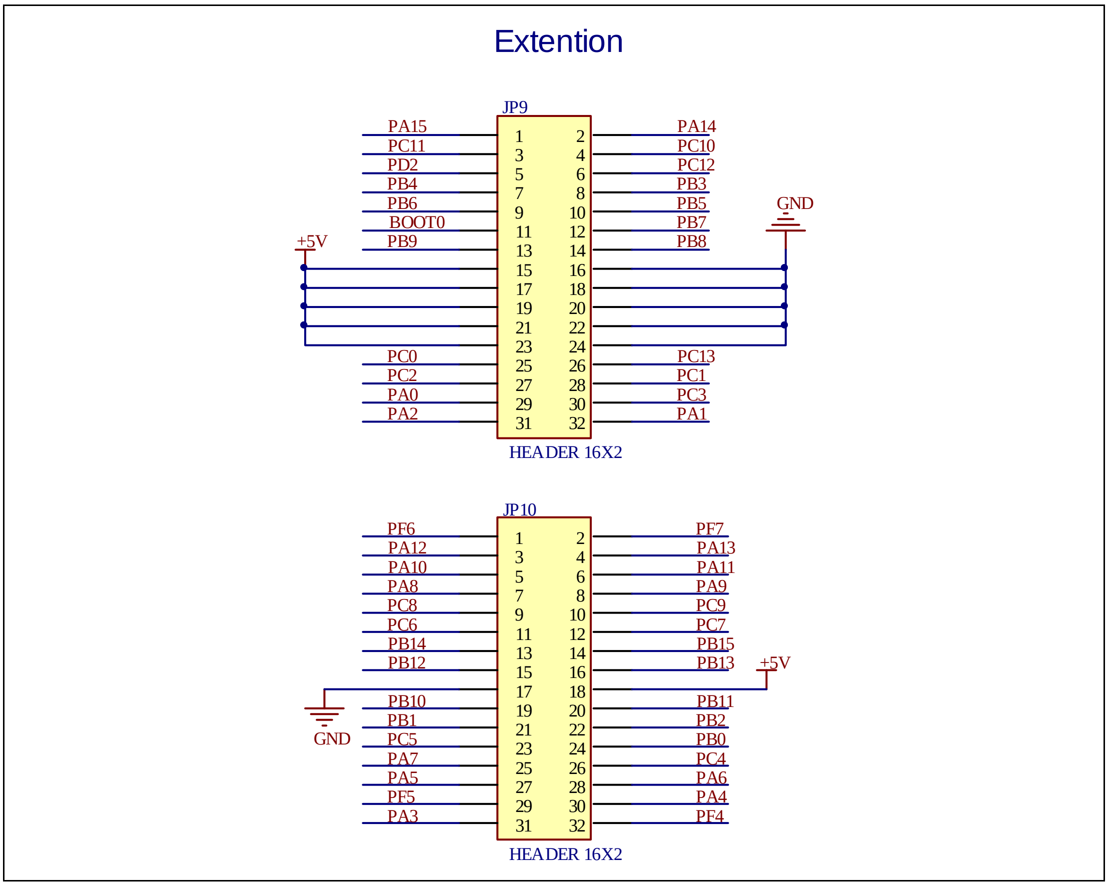
4.16 CMP



4.17 IFRR


+5V
R28
470Ω
LED6
KD-03144R
IR_OUT
R30
1KΩ
Q1
S8050
R31
10KΩ
GND
IFRR
PA7
GND
R29
100Ω
JP11
HS0038B
+5V
C24
10V/4.7uF
GND
Short JP12(1,2) for IFRR function
Short JP12(2,3) for LCD function
JP12
IR_OUT
PB9
COM3
HEADER 3

4.18 Extension



5 Routine use guide

5.1 GPIO_Runing_Led

5.1.1 DEMO Purpose

This Demo includes the following functions of GD32 MCU：

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

GD32190R-EVAL-V1.1 board has five LEDs. The LED1, LED2, LED3 and LED4 are controlled by GPIO, and if the LED5 lights up, it indicates the system is power on. This Demo will show how to light the LEDs.

5.1.2 DEMO Running Result

Download the program<01_GPIO_Runing_Led> to the EVAL board, four LEDs on and then LED1, LED2, LED3, LED4 will turn on and turn off every 500ms.

5.2 GPIO_KeyBoard_Polling_mode

5.2.1 DEMO Purpose

This Demo includes the following functions of GD32 MCU：

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

GD32190R-EVAL-V1.1 board has three keys and five LEDs. The three keys are Reset key, Tamper key and Wakeup key. The LED1, LED2, LED3 and LED4 are controlled by GPIO, and if the LED5 lights up, it indicates the system is power on. 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, Indicates that the button is pressed success and toggle LED1.

5.2.2 DEMO Running Result

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

5.3 GPIO_KeyBoard_Interrupt_mode

5.3.1 DEMO Purpose

This Demo includes the following functions of GD32 MCU：

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

GD32190R-EVAL-V1.1 board has three keys and five LEDs. The three keys are Reset key, Tamper key and Wakeup key. The LED1, LED2, LED3 and LED4 are controlled by GPIO, and if the LED5 lights up, it indicates the system is power on. This Demo will show how to use the EXTI interrupt line to control the LED1. When press down the Tamper Key, it will produce an interrupt. In the interrupt service function toggle LED1.

5.3.2 DEMO Running Result

Download the program<03_GPIO_KeyBoard_Interrupt_mode> to the EVAL board, Press down the Tamper Key, LED1 will be turn on. Press down the Tamper Key again, LED1 will be turn off.

5.4 USART_Printf

5.4.1 DEMO Purpose

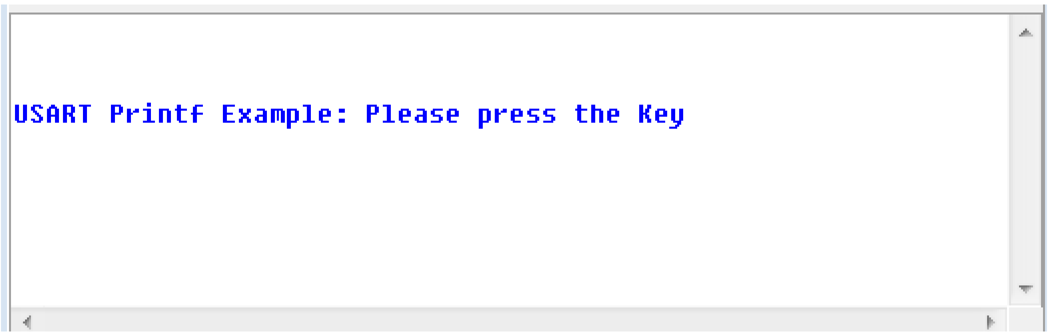
This Demo includes the following functions of GD32 MCU：

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

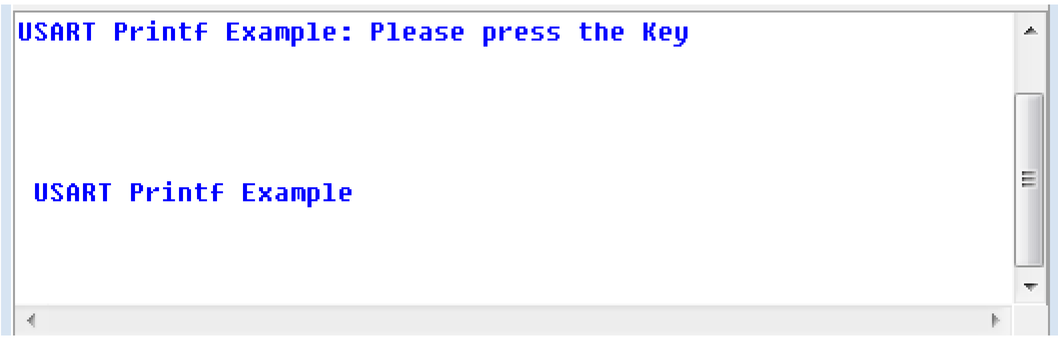
5.4.2 DEMO Running Result

Download the program <04_USART_Printf> to the EVAL board and run. This implementation output “USART Printf Example: Please press the Key” on the hyperterminal using EVAL_COM1. Press K3 key, serial port will output “USART Printf Example”.

Information via a serial port output as following.



Press K3 key, serial port output as following.



5.5 USART_HyperTerminal_Interrupt

5.5.1 DEMO Purpose

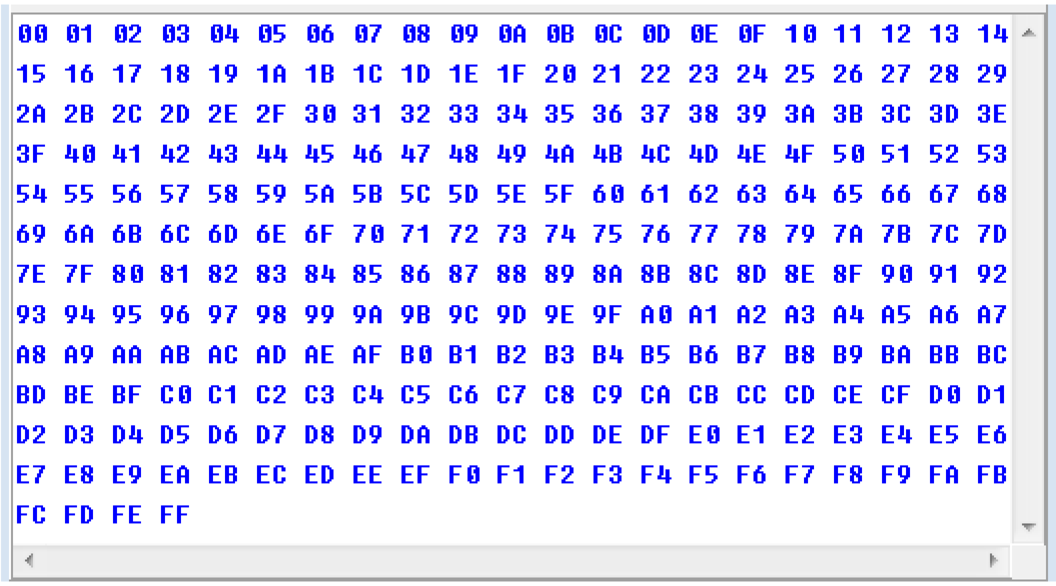
This Demo includes the following functions of GD32 MCU：

- Learn to use the EVAL_COM1 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. When the program is running, the EVAL_COM1 sends the array TxBuffer to the hyperterminal and still waits for data from the hyperterminal that you must enter. The data that you have entered is stored in the RxBuffer array. The receive buffer have a NbrOfDataToRead bytes as maximum. Then,compare TxBuffer with RxBuffer and if TxBuffer is same with RxBuffer,LED1,LED2,LED3,LED4 light in turn, otherwise LED1,LED2,LED3,LED4 toggle together .

Information via a serial port output as following.



5.6 USART_DMA

5.6.1 DEMO Purpose

This Demo includes the following functions of GD32 MCU：

- Learn to to use the EVAL_COM1 Transmit and Receive using DMA

5.6.2 DEMO Running Result

Download the program <06_USART_DMA > to the EVAL board and run. When the program is running, the EVAL_COM1 sends the array TxBuffer to the hyperterminal and waits for data from the hyperterminal that you must enter. The data that you have entered is stored in the RxBuffer array. The receive buffer have a NbrOfDataToRead bytes as maximum. Then compare TxBuffer with RxBuffer and if TxBuffer is same with RxBuffer,LED1,LED2,LED3,LED4 light in turn,otherwise LED1,LED2,LED3,LED4 toggle together .

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 I2C_EEPROM

5.7.1 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

5.7.2 DEMO Running Result

Download the program<07_I2C_EEPROM> to the EVAL board, and under normal circumstances, serial print out the following information:

```
###############################################################################
GD32190R-EVAL-V1.1 System is Starting up...
GD32190R-EVAL-V1.1 Program Version number:GD1.0.0
GD32190R-EVAL-V1.1 Program Compile time:(Jan 29 2016 - 09:02:21)
GD32190R-EVAL-V1.1 GD32F1x0_StdPeriph_Version:2.0.0
GD32190R-EVAL-V1.1 SystemCoreClock:72000000Hz
GD32190R-EVAL-V1.1 Flash:128K Bytes
GD32190R-EVAL-V1.1 The CPU Unique Device ID:[7C000000-32303635-37320836]
GD32190R-EVAL-V1.1 I2C-24C02 configured....
---------> The I2C2 is Hardware interface
---------> The Speed is 400000AT24C02 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!
```

Firstly, the serial port will output some information related to the EVAL board. Then the data of 256 bytes will be written to the EEPROM from the address 0x00 and printed. Finally, reading the EEPROM from address 0x00 for 256 bytes and the result will be printed. 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 "Test Passed I2C-AT24C02!" and the four LEDs lights flashing, otherwise the serial port will outpu t"Read and Write are't Matching Err: Data" and all the four LEDs light.

5.8 SPI_Flash

5.8.1 DEMO Purpose

This Demo includes the following functions of GD32 MCU：

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

5.8.2 DEMO Running Result

The computer serial port line connected to the COM1 port of development board, set the baud rate of serial assistant software to 115200, 8 bits data bit, 1 bit stop bit. Download the program <08_SPI_FLASH> to the EVAL board, through the serial assistant software can observe the operation condition and will display the ID of the flash, 256 bytes data which write to and read from flash. The following is the experimental results.

```
################################################################################
GD32190R-EVAL-V1.1 System is Starting up...
GD32190R-EVAL-V1.1 Program Version number:GD1.0.0
GD32190R-EVAL-V1.1 Program Compile time:(Jan 29 2016 - 11:34:14)
GD32190R-EVAL-V1.1 SystemCoreClock:72000000Hz
GD32190R-EVAL-V1.1 Flash:64K
GD32190R-EVAL-V1.1 The CPU Unique Device ID:[524743-32303434-39530B3C]
GD32190R-EVAL-V1.1 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!
```

5.9 I2S_Audio_Player

5.9.1 DEMO Purpose

This Demo includes the following functions of GD32 MCU：

- Learn to use I2S module to output audio file

GD32190R-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.

5.9.2 DEMO Running Result

Download the program <09_I2S_Audio_Player> to the EVAL board, insert the headphone into the audio port, and then listen to the audio file.

5.10 IRInfrared_Transceiver

5.10.1 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 TIMER16 and TIMER17 implement Infrared function

5.10.2 DEMO Running Result

Jump the JP12 to IFRR with the jumper cap, and then download the program <10_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.

5.11 TIMER_Breath_LED

5.11.1 DEMO Purpose

This Demo includes the following functions of GD32 MCU:

- Learn to use advanced timer output PWM wave
- Learn to use general timer generated update interrupt

5.11.2 DEMO Running Result

Use the DuPont line to connect the TIMER1 CH1 (PA8) and LED1 (PA11), and then download the program <11_TIMER_Breath_LED> to the EVAL board and run.
When the program is running, you can see LED1 from dark to bright gradually and by light gradually darken, ad infinitum, just like breathing as rhythm.

5.12 HDMI_CEC_HostSlaveCommunication

5.12.1 DEMO Purpose

This Demo includes the following functions of GD32 MCU：

- Learn the communication function of HDMI-CEC

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

5.12.2 DEMO Running Result

This routines need to prepare two GD32F1x0_EVAL board, one board as a sender, the other as a receiver. First use the DuPont to connect CEC bus (PB8) and ground wire (GND) pins in the two board, and then download the program <12_HDMI_CEC_HostSlaveCommunication> to the board for running. When the program runs, the first development board of the LCD display is data 0, press one of the development board TAMPER key, the other piece of the development board LCD number will increase, which shows the end of a data transmission. Each it increases to 9, it will clear to 0 to re-increase; press the WAKEUP key, the number will decline, which also shows the end of a data transmission. Every time it decreases to 0, it will return to the number 9 to re-decrease.

5.13 Comparator_Obtain_Brightness

5.13.1 DEMO Purpose

This Demo includes the following functions of GD32 MCU：

- Learn to use DAC as an input of comparator
- Learn to use comparator to output compare result

Every GD32190R-EVAL board comparator has two input, one is DAC input, and the other one is slide rheostat output voltage. Compare two input voltages, output a high or low level, then corresponding LED is on.

5.13.2 DEMO Running Result

Download the program <13_Comparator_Obtain_Brightness> to the EVAL board, change the slide rheostat output voltage, comparing it with DAC output voltage, if it is larger than DAC output voltage, LED1 is on, and if it is less than DAC output voltage, LED4 is on

5.14 ADC_Conversion_Triggered_By_Timer

5.14.1 DEMO Purpose

This Demo includes the following functions of GD32 MCU：

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

TIMER2 CC2 event triggers an ADC conversion, the ADC output digital value corresponds to its analog input, and it will change with its analog input.

5.14.2 DEMO Running Result

Download the program <14_ADC_Conversion_Triggered_By_Timer > to the GD32190R-EVAL board, change the analog input, the ADC, which is triggered by TIMER2 CC2 event, will output corresponding data. Measuring TP2 on board, and check if its value matches the converted data through the hyperterminal window.

Information via a serial port output as following.

```
ADC_Conversion_Triggered_By_Timer demo:
ADC value is a8c
ADC value is a87
ADC value is a8e
ADC value is a8c
```

5.15 DAC_Digital_To_Analog_Conversion

5.15.1 DEMO Purpose

This Demo includes the following functions of GD32 MCU：

- Learn to use DAC channel to generate different voltages on DAC1 output

5.15.2 DEMO Running Result

Download the program <15_DAC_Digital_To_Analog_Conversion> to the EVAL board, the digital value is 0x7ff0, its converted analog voltage should be 2.5V, using the voltmeter to measure PA4, its value is 2.5V.

5.16 RTC_Calendar

5.16.1 DEMO Purpose

This Demo includes the following functions of GD32 MCU：

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

5.16.2 DEMO Running Result

Jump the JP12 to LCD with the jumper cap, and download the program <16_RTC_Calendar> to the EVAL board and run. When the program is running, LCD displays the time of calendar, when you press the Tamper Key, the day time, the year, the month and the date successively display on the LCD.

5.17 TSI_TouchKey_leds

5.17.1 DEMO Purpose

This Demo includes the following functions of GD32 MCU：

- Learn to use TSI module implement Touch Key function

5.17.2 DEMO Running Result

Download the program <17_TSI_TouchKey_leds> to the EVAL board and run. When the program is running, you can use a finger touch the TouchKey (TK1 or TK2) on the EVAL board, and then the associated LED is light.

5.18 OPAMP_Amplify

5.18.1 DEMO Purpose

This Demo includes the following functions of GD32 MCU：

- Learn to use SysTick to generate 1ms delay
- Learn to use OPAMP to amplify the input voltage
- Learn to use ADC acquisition voltage
- Learn to retarget the C library printf function to the USART

GD32190R-EVAL-V1.1 board has OPAMP3.This Demo will show how to use OPAMP3 to amplify the input voltage. The input voltage is connected to the JP8 Pin 1. The amplifier output pin can be configured to the ADC sample channel 13. We can adjust the VR2 to change the magnification. The amplified voltage can be measured on the JP8 pin 2 with a universal meter or oscilloscope. Amplified voltage can also be sampled by ADC1 and printed out through the serial port.

5.18.2 DEMO Running Result

Download the program<18_OPAMP_Amplify > to the EVAL board. The amplified voltage will be sampled by ADC1 and printed out through the serial port. The running results are as follows.

```
ADC_Value 4.309

ADC_Value 4.311

ADC_Value 4.311

ADC_Value 4.136

ADC_Value 3.156

ADC_Value 2.692

ADC_Value 2.694
```

5.19 LCD_Glass

5.19.1 DEMO Purpose

This Demo includes the following functions of GD32 MCU：

- Learn to use LCD module to display number

5.19.2 DEMO Running Result

Jump the JP12 to LCD with the jumper cap, and download the program <19_LCD_Glass> to the EVAL board and run. When the program is running, you can see the LCD displaying the number which adds 1 per second.

5.20 CAN_DualCAN

5.20.1 DEMO Purpose

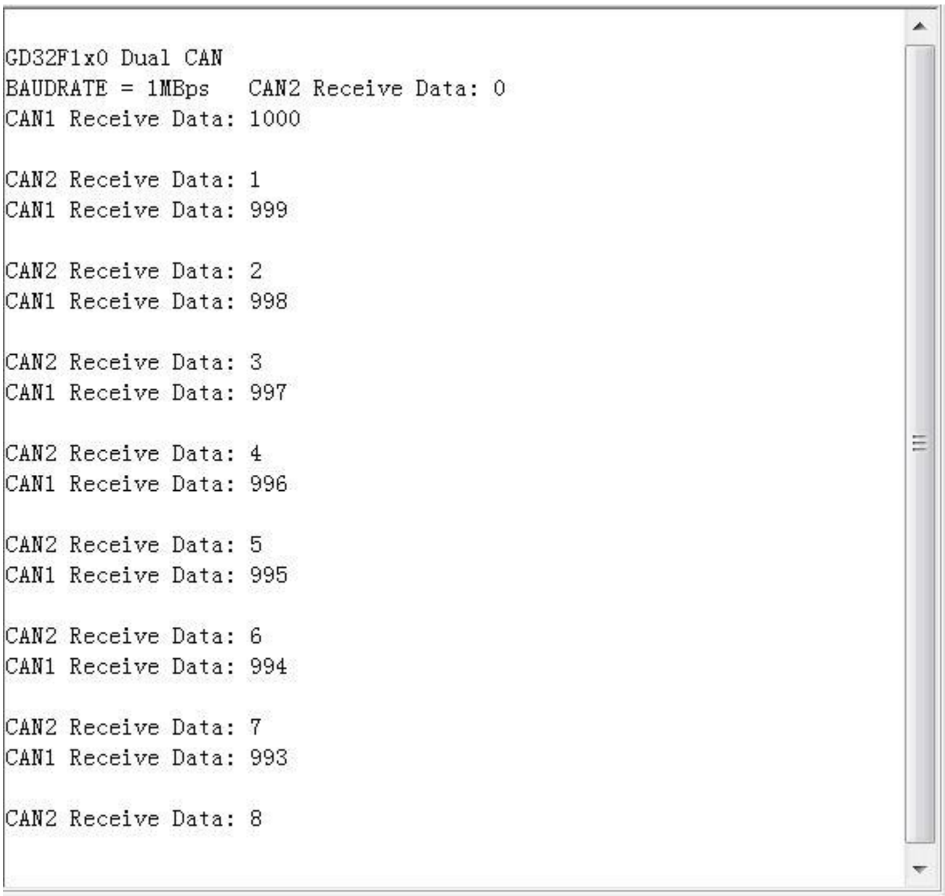
This Demo includes the following functions of GD32 MCU：

- Learn the communication between CAN1 and CAN2

GD32190R-EVAL-V1.1 development board integrates the CAN (Controller Area Network) bus controller, which is a common industrial control bus. CAN bus controller follows the CAN bus protocol of 2.0 A and 2.0 B. This Demo mainly shows how to communicate between CAN1 and CAN2.

5.20.2 DEMO Running Result

Jump the JP6 to CAN1 with the jumper cap, connect the signal pins of CAN1 and that of CAN2 together, CANH connects to CANH, and CANL connects to CANL. Download the program <20_CAN_DualCAN> to the EVAL board, the following information will be showed through the serial output.
Information via a serial port output as following



```
GD32F1x0 Dual CAN
BAUDRATE = 1MBps   CAN2 Receive Data: 0
CAN1 Receive Data: 1000

CAN2 Receive Data: 1
CAN1 Receive Data: 999

CAN2 Receive Data: 2
CAN1 Receive Data: 998

CAN2 Receive Data: 3
CAN1 Receive Data: 997

CAN2 Receive Data: 4
CAN1 Receive Data: 996

CAN2 Receive Data: 5
CAN1 Receive Data: 995

CAN2 Receive Data: 6
CAN1 Receive Data: 994

CAN2 Receive Data: 7
CAN1 Receive Data: 993

CAN2 Receive Data: 8
```

CAN1 sends data 0 and do self-increment for the next transmission, if CAN2 receives the data successfully, CAN2 will send data 1000 and do self-decrement for the next transmission. Each time the data is successfully received, it will be printed through the serial output.

6 Revision history

Table 2 Revision history

Revision No.	Description	Date
1.0	Initial Release	Jan.19, 2016