# Table of Contents

<table>
<thead>
<tr>
<th>Section</th>
<th>Title</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Summary</td>
<td>4</td>
</tr>
<tr>
<td>2</td>
<td>Function Pin Assign</td>
<td>4</td>
</tr>
<tr>
<td>3</td>
<td>Getting started</td>
<td>7</td>
</tr>
<tr>
<td>4</td>
<td>Hardware layout overview</td>
<td>8</td>
</tr>
<tr>
<td>4.1</td>
<td>Power</td>
<td>8</td>
</tr>
<tr>
<td>4.2</td>
<td>Boot</td>
<td>8</td>
</tr>
<tr>
<td>4.3</td>
<td>LED</td>
<td>9</td>
</tr>
<tr>
<td>4.4</td>
<td>KEY</td>
<td>9</td>
</tr>
<tr>
<td>4.5</td>
<td>USART</td>
<td>10</td>
</tr>
<tr>
<td>4.6</td>
<td>ADC</td>
<td>10</td>
</tr>
<tr>
<td>4.7</td>
<td>DAC</td>
<td>10</td>
</tr>
<tr>
<td>4.8</td>
<td>I2S</td>
<td>11</td>
</tr>
<tr>
<td>4.9</td>
<td>I2C</td>
<td>11</td>
</tr>
<tr>
<td>4.10</td>
<td>SPI</td>
<td>11</td>
</tr>
<tr>
<td>4.11</td>
<td>CAN</td>
<td>12</td>
</tr>
<tr>
<td>4.12</td>
<td>Ethernet</td>
<td>12</td>
</tr>
<tr>
<td>4.13</td>
<td>SDIO</td>
<td>13</td>
</tr>
<tr>
<td>4.14</td>
<td>SDRAM</td>
<td>13</td>
</tr>
<tr>
<td>4.15</td>
<td>LCD</td>
<td>14</td>
</tr>
<tr>
<td>4.16</td>
<td>USBFS</td>
<td>14</td>
</tr>
<tr>
<td>4.17</td>
<td>USBHS</td>
<td>15</td>
</tr>
<tr>
<td>4.18</td>
<td>Extension</td>
<td>15</td>
</tr>
<tr>
<td>4.19</td>
<td>GD-Link</td>
<td>16</td>
</tr>
<tr>
<td>5</td>
<td>Routine use guide</td>
<td>16</td>
</tr>
<tr>
<td>5.1</td>
<td>GPIO_Runing_Led</td>
<td>16</td>
</tr>
<tr>
<td>5.2</td>
<td>GPIO_Keyboard_Polling_mode</td>
<td>17</td>
</tr>
<tr>
<td>5.3</td>
<td>GPIO_Keyboard_Interrupt_mode</td>
<td>17</td>
</tr>
<tr>
<td>5.4</td>
<td>USART_Printf</td>
<td>18</td>
</tr>
<tr>
<td>5.5</td>
<td>USART_Echo_Interrupt_mode</td>
<td>18</td>
</tr>
<tr>
<td>5.6</td>
<td>USART_DMA</td>
<td>19</td>
</tr>
<tr>
<td>5.7</td>
<td>ADC_Temperature_Vrefint_Vbat</td>
<td>20</td>
</tr>
<tr>
<td>5.8</td>
<td>ADC0_ADC1_Follow_up_mode</td>
<td>20</td>
</tr>
<tr>
<td>5.9</td>
<td>ADC0_ADC1-Regular_Parallel_mode</td>
<td>21</td>
</tr>
</tbody>
</table>
5.10 DAC_Output_Voltage_Value .................................................................22
5.11 I2C_EEPROM .................................................................23
5.12 SPI_QSPI_Flash .................................................................24
5.13 I2S_Audio_Player .................................................................25
5.14 EXMC_SDRAM .................................................................26
5.15 EXMC_SDRAM_DeepSleep .........................................................26
5.16 SDIO_SDCardTest .................................................................27
5.17 CAN_Network .................................................................28
5.18 RCU_Clock_Output .................................................................29
5.19 CTC_Calibration .................................................................30
5.20 PMU_sleep_wakeup .................................................................30
5.21 FMC_Dualboot .................................................................31
5.22 RTC_Calendar .................................................................32
5.23 TIMER_Breath_LED .................................................................33
5.24 TLI_IPA .................................................................33
5.25 TRNG_Get_Random .................................................................34
5.26 ENET .................................................................35
5.27 USB_Device .................................................................41
5.28 USB_Host .................................................................45

6 Revision history ........................................................................47
List of Tables

Table 1 Function pin assign .......................................................... 4
Table 2 Revision history .............................................................. 47
1 Summary

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

2 Function Pin Assign

Table 1 Function pin assign

<table>
<thead>
<tr>
<th>Function</th>
<th>Pin</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>LED</td>
<td>PD4</td>
<td>LED1</td>
</tr>
<tr>
<td></td>
<td>PD5</td>
<td>LED2</td>
</tr>
<tr>
<td></td>
<td>PG3</td>
<td>LED3</td>
</tr>
<tr>
<td>RESET</td>
<td>PA0</td>
<td>K1-Reset</td>
</tr>
<tr>
<td>KEY</td>
<td>PC13</td>
<td>K3-Tamper</td>
</tr>
<tr>
<td></td>
<td>PB14</td>
<td>K4-User key</td>
</tr>
<tr>
<td>USART0</td>
<td>PA9</td>
<td>USART0_TX</td>
</tr>
<tr>
<td></td>
<td>PA10</td>
<td>USART0_RX</td>
</tr>
<tr>
<td>ADC</td>
<td>PF6</td>
<td>ADC012_IN4</td>
</tr>
<tr>
<td>DAC</td>
<td>PA4</td>
<td>DAC_OUT0</td>
</tr>
<tr>
<td>I2C</td>
<td>PB6</td>
<td>I2C0_SCL</td>
</tr>
<tr>
<td></td>
<td>PB7</td>
<td>I2C0_SDA</td>
</tr>
<tr>
<td>SPI</td>
<td>PG10</td>
<td>SPI5_IO2</td>
</tr>
<tr>
<td></td>
<td>PG11</td>
<td>SPI5_IO3</td>
</tr>
<tr>
<td></td>
<td>PG13</td>
<td>SPI5_SCK</td>
</tr>
<tr>
<td></td>
<td>PG14</td>
<td>SPI5_MOSI</td>
</tr>
<tr>
<td></td>
<td>PG12</td>
<td>SPI5_MISO</td>
</tr>
<tr>
<td></td>
<td>PG9</td>
<td>SPI5_CS</td>
</tr>
<tr>
<td>I2S</td>
<td>PC6</td>
<td>I2S1_MCK</td>
</tr>
<tr>
<td></td>
<td>PC7</td>
<td>I2S1_CK</td>
</tr>
<tr>
<td></td>
<td>PB9</td>
<td>I2S1_WS</td>
</tr>
<tr>
<td></td>
<td>PC1</td>
<td>I2S1_SD</td>
</tr>
<tr>
<td>CAN</td>
<td>PD0</td>
<td>CAN0_RX</td>
</tr>
<tr>
<td></td>
<td>PD1</td>
<td>CAN0_TX</td>
</tr>
<tr>
<td>SDRAM</td>
<td>PC0</td>
<td>EXMC_SDNWE</td>
</tr>
<tr>
<td></td>
<td>PC2</td>
<td>EXMC_SDNE0</td>
</tr>
<tr>
<td></td>
<td>PC3</td>
<td>EXMC_SDCKE0</td>
</tr>
<tr>
<td></td>
<td>PD0</td>
<td>EXMC_D2</td>
</tr>
<tr>
<td>Pin</td>
<td>Function</td>
<td></td>
</tr>
<tr>
<td>-------</td>
<td>----------------</td>
<td></td>
</tr>
<tr>
<td>PD1</td>
<td>EXMC_D3</td>
<td></td>
</tr>
<tr>
<td>PD8</td>
<td>EXMC_D13</td>
<td></td>
</tr>
<tr>
<td>PD9</td>
<td>EXMC_D14</td>
<td></td>
</tr>
<tr>
<td>PD10</td>
<td>EXMC_D15</td>
<td></td>
</tr>
<tr>
<td>PD14</td>
<td>EXMC_D0</td>
<td></td>
</tr>
<tr>
<td>PD15</td>
<td>EXMC_D1</td>
<td></td>
</tr>
<tr>
<td>PE0</td>
<td>EXMC_NBL0</td>
<td></td>
</tr>
<tr>
<td>PE1</td>
<td>EXMC_NBL1</td>
<td></td>
</tr>
<tr>
<td>PE7</td>
<td>EXMC_D4</td>
<td></td>
</tr>
<tr>
<td>PE8</td>
<td>EXMC_D5</td>
<td></td>
</tr>
<tr>
<td>PE9</td>
<td>EXMC_D6</td>
<td></td>
</tr>
<tr>
<td>PE10</td>
<td>EXMC_D7</td>
<td></td>
</tr>
<tr>
<td>PE11</td>
<td>EXMC_D8</td>
<td></td>
</tr>
<tr>
<td>PE12</td>
<td>EXMC_D9</td>
<td></td>
</tr>
<tr>
<td>PE13</td>
<td>EXMC_D10</td>
<td></td>
</tr>
<tr>
<td>PE14</td>
<td>EXMC_D11</td>
<td></td>
</tr>
<tr>
<td>PE15</td>
<td>EXMC_D12</td>
<td></td>
</tr>
<tr>
<td>PF0</td>
<td>EXMC_A0</td>
<td></td>
</tr>
<tr>
<td>PF1</td>
<td>EXMC_A1</td>
<td></td>
</tr>
<tr>
<td>PF2</td>
<td>EXMC_A2</td>
<td></td>
</tr>
<tr>
<td>PF3</td>
<td>EXMC_A3</td>
<td></td>
</tr>
<tr>
<td>PF4</td>
<td>EXMC_A4</td>
<td></td>
</tr>
<tr>
<td>PF5</td>
<td>EXMC_A5</td>
<td></td>
</tr>
<tr>
<td>PF11</td>
<td>EXMC_NRAS</td>
<td></td>
</tr>
<tr>
<td>PF12</td>
<td>EXMC_A6</td>
<td></td>
</tr>
<tr>
<td>PF13</td>
<td>EXMC_A7</td>
<td></td>
</tr>
<tr>
<td>PF14</td>
<td>EXMC_A8</td>
<td></td>
</tr>
<tr>
<td>PF15</td>
<td>EXMC_A9</td>
<td></td>
</tr>
<tr>
<td>PG0</td>
<td>EXMC_A10</td>
<td></td>
</tr>
<tr>
<td>PG1</td>
<td>EXMC_A11</td>
<td></td>
</tr>
<tr>
<td>PG2</td>
<td>EXMC_A12</td>
<td></td>
</tr>
<tr>
<td>PG4</td>
<td>EXMC_A14</td>
<td></td>
</tr>
<tr>
<td>PG5</td>
<td>EXMC_A15</td>
<td></td>
</tr>
<tr>
<td>PG8</td>
<td>EXMC_SDCLK</td>
<td></td>
</tr>
<tr>
<td>PG15</td>
<td>EXMC_NCAS</td>
<td></td>
</tr>
<tr>
<td>PD2</td>
<td>SDIO_CMD</td>
<td></td>
</tr>
<tr>
<td>PC12</td>
<td>SDIO_CK</td>
<td></td>
</tr>
<tr>
<td>PC8</td>
<td>SDIO_D0</td>
<td></td>
</tr>
<tr>
<td>PC9</td>
<td>SDIO_D1</td>
<td></td>
</tr>
<tr>
<td>PC10</td>
<td>SDIO_D2</td>
<td></td>
</tr>
<tr>
<td>PC11</td>
<td>SDIO_D3</td>
<td></td>
</tr>
<tr>
<td>PC6</td>
<td>TLI_HSYNC</td>
<td></td>
</tr>
<tr>
<td>LCD</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Pin</td>
<td>Function</td>
<td></td>
</tr>
<tr>
<td>-------</td>
<td>-----------------------</td>
<td></td>
</tr>
<tr>
<td>PA4</td>
<td>TLI_VSYNC</td>
<td></td>
</tr>
<tr>
<td>PG7</td>
<td>TLI_PIXCLK</td>
<td></td>
</tr>
<tr>
<td>PF10</td>
<td>TLI_DE</td>
<td></td>
</tr>
<tr>
<td>PG6</td>
<td>TLI_R7</td>
<td></td>
</tr>
<tr>
<td>PA8</td>
<td>TLI_R6</td>
<td></td>
</tr>
<tr>
<td>PA12</td>
<td>TLI_R5</td>
<td></td>
</tr>
<tr>
<td>PA11</td>
<td>TLI_R4</td>
<td></td>
</tr>
<tr>
<td>PB0</td>
<td>TLI_R3</td>
<td></td>
</tr>
<tr>
<td>PD3</td>
<td>TLI_G7</td>
<td></td>
</tr>
<tr>
<td>PC7</td>
<td>TLI_G6</td>
<td></td>
</tr>
<tr>
<td>PB11</td>
<td>TLI_G5</td>
<td></td>
</tr>
<tr>
<td>PB10</td>
<td>TLI_G4</td>
<td></td>
</tr>
<tr>
<td>PA1</td>
<td>ETH_RMII_REF_CLK</td>
<td></td>
</tr>
<tr>
<td>PA2</td>
<td>ETH_MDIO</td>
<td></td>
</tr>
<tr>
<td>PA7</td>
<td>ETH_RMII_CRS_DV</td>
<td></td>
</tr>
<tr>
<td>PB11</td>
<td>ETH_RMII_TX_EN</td>
<td></td>
</tr>
<tr>
<td>PB12</td>
<td>ETH_RMII_TXD0</td>
<td></td>
</tr>
<tr>
<td>PB13</td>
<td>ETH_RMII_TXD1</td>
<td></td>
</tr>
<tr>
<td>PC1</td>
<td>ETH_MDC</td>
<td></td>
</tr>
<tr>
<td>PC4</td>
<td>ETH_RMII_RXD0</td>
<td></td>
</tr>
<tr>
<td>PC5</td>
<td>ETH_RMII_RXD1</td>
<td></td>
</tr>
<tr>
<td>PA9</td>
<td>USB_VBUS</td>
<td></td>
</tr>
<tr>
<td>PA11</td>
<td>USB_DM</td>
<td></td>
</tr>
<tr>
<td>PA12</td>
<td>USB_DP</td>
<td></td>
</tr>
<tr>
<td>PD13</td>
<td>USB_VBUS_CTRL</td>
<td></td>
</tr>
<tr>
<td>PC3</td>
<td>USB_HS_ULPI_NXT</td>
<td></td>
</tr>
<tr>
<td>PC2</td>
<td>USB_HS_ULPI_DIR</td>
<td></td>
</tr>
<tr>
<td>PC0</td>
<td>USB_HS_ULPI_STP</td>
<td></td>
</tr>
<tr>
<td>PA5</td>
<td>USB_HS_ULPI_CK</td>
<td></td>
</tr>
<tr>
<td>PB5</td>
<td>USB_HS_ULPI_D7</td>
<td></td>
</tr>
<tr>
<td>PB13</td>
<td>USB_HS_ULPI_D6</td>
<td></td>
</tr>
<tr>
<td>PB12</td>
<td>USB_HS_ULPI_D5</td>
<td></td>
</tr>
<tr>
<td>PB11</td>
<td>USB_HS_ULPI_D4</td>
<td></td>
</tr>
<tr>
<td>PB10</td>
<td>USB_HS_ULPI_D3</td>
<td></td>
</tr>
<tr>
<td>PB1</td>
<td>USB_HS_ULPI_D2</td>
<td></td>
</tr>
</tbody>
</table>

---

**Ethernet**

**USB_FS**

**USB_HS**
3 Getting started

The EVAL board uses Mini USB connector or DC-005 connector to get power DC +5V, which is the hardware system normal work voltage. A J-Link tool or GD-Link on board is necessary in order to download and debug programs. Select the correct boot mode and then power on, the 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. In Firmware folder, Addon and Software Pack are used to add the devices, peripherals and others to IDE. During use, the following points should be noted:

1. If you use Keil uVision4 to open the project, install the GD32F4xx_Addon.1.0.0.exe which is in \Library\Firmware to load the associated files.
2. If you use Keil uVision5 to open the project, there are two ways to solve the "Device Missing (s)" problem. One is to install GigaDevice.GD32F4xx_DFP.1.0.0.pack which is in \Library\Firmware. In Project menu, select the Manage sub menu, click on the "Version Migrate 5 Format..." menu, the Keil uVision4 project will be converted to Keil uVision5 project. Then add "C:\Keil_v5\ARM\Pack\ARM\CMSIS\4.2.0\CMSIS\Include" to C/C++ in Option for Target. The other is to install Addon directly. Select the installation directory of Keil uVision5 software, such as C:\Keil_v5, in Destination Folder of Folder Selection. Select the corresponding device in Device of Option for Target and add "C:\Keil_v5\ARM\Pack\ARM\CMSIS\4.2.0\CMSIS\Include" to C/C++ in Option for Target.
3. If you use IAR to open the project, install IAR_GD32F4xx_Addon.exe which is in \Library\Firmware to load the associated files.
4 Hardware layout overview

4.1 Power

[Diagram of power supply circuit]

4.2 Boot

[Diagram of boot mode selection]

<table>
<thead>
<tr>
<th>BOOT1</th>
<th>BOOT0</th>
<th>Boot Mode</th>
</tr>
</thead>
<tbody>
<tr>
<td>Any</td>
<td>2-3</td>
<td>User memory</td>
</tr>
<tr>
<td>2-3</td>
<td>1-2</td>
<td>System memory</td>
</tr>
<tr>
<td>1-2</td>
<td>1-2</td>
<td>SRAM memory</td>
</tr>
</tbody>
</table>
4.3 LED

![LED Diagram]

4.4 KEY

![KEY Diagram]
4.5 USART

![USART Diagram]

4.6 ADC

![ADC Diagram]

4.7 DAC

![DAC Diagram]
4.8 I2S

![I2S Diagram]

4.9 I2C

![I2C Diagram]

4.10 SPI

![SPI Flash Diagram]
4.11 CAN

Short P1(1,2) for EXMC function
Short P2(2,3) for CAN0 function

4.12 Ethernet

Short P1(1,2) for USB function
Short P2(2,3) for USB_HS function
4.13 SDIO

4.14 SDRAM
4.15 LCD

LCD

4.16 USBFS
4.17 USBHS

USB HS
Short JP16(1,2) for SDRAM function
Short JP17(1,2) for SDRAM function
Short JP16(2,3) for USB_HS function
Short JP17(2,3) for USB_HS function
Short JP16(1,2) for USB_HS function
Short JP17(2,3) for USB_HS function

4.18 Extension

Extension Pin
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

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

5.1.2  DEMO Running Result

Download the program <01_GPIO_Runing_Led> to the EVAL board, LED1, LED2, LED3 will turn on in sequence with interval of 200ms, and turn off together, 200ms later, repeat the process.
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

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

This demo will show how to use the Tamper key to control the LED1. When press down the Tamper Key, it will check the input value of the IO port. If the value is 0 and will wait for 100ms. Check the input value of the IO port again. If the value still is 0, it indicates that the button is pressed successfully and toggle LED1.

5.2.2 DEMO Running Result

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

5.3 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

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

This demo will show how to use the EXTI interrupt line to control the LED1. When press down the Tamper Key, it will produce an interrupt. In the interrupt service function, the demo will toggle LED1.
5.3.2 DEMO Running Result

Download the program `<03_GPIO_Keyboard Interrupt_mode>` to the EVAL board, Press down the Tamper Key, LED1 will be turned on. Press down the Tamper Key again, LED1 will be turned off.

5.4 USART_Printf

5.4.1 DEMO Purpose

This demo includes the following functions of GD32 MCU:
- Learn to use GPIO control the LED
- Learn to retarget the C library printf function to the USART

5.4.2 DEMO Running Result

Download the program `<04_USART_Printf>` to the EVAL board, jump the JP13 to USART with the jumper cap and connect serial cable to EVAL_COM1. This implementation outputs "USART printf example: please press the Tamper key" on the HyperTerminal using EVAL_COM1. Press the Tamper key, serial port will output "USART printf example". The output information via the serial port is as following.

<table>
<thead>
<tr>
<th>USART printf example: please press the Tamper key</th>
</tr>
</thead>
<tbody>
<tr>
<td>USART printf example</td>
</tr>
</tbody>
</table>

5.5 USART_Echo_Interrupt_mode

5.5.1 DEMO Purpose

This demo includes the following functions of GD32 MCU:
- Learn to use the USART transmit and receive interrupts to communicate with the serial terminal tool

5.5.2 DEMO Running Result

Download the program `<05_USART_Echo_Interrupt_mode>` to the EVAL board, jump
the JP13 to USART with the jumper cap and connect serial cable to EVAL_COM1. Firstly, all the LEDs are turned on and off for test. Then, the EVAL_COM1 sends the tx_buffer array (from 0x00 to 0xFF) to the serial terminal tool supporting hex format communication and waits for receiving data of BUFFER_SIZE bytes from the serial terminal. The data MCU have received is stored in the rx_buffer array. After that, compare tx_buffer with rx_buffer. If tx_buffer is same with rx_buffer, LED1, LED2, LED3 flash by turns. Otherwise, LED1, LED2, LED3 toggle together.

The output information via the serial port is as following.

| 00 01 02 03 04 05 06 07 08 09 0A 0B 0C 0D 0E 0F |
|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|
| 10 11 12 13 14 15 16 17 18 19 1A 1B 1C 1D 1E 1F |
| 20 21 22 23 24 25 26 27 28 29 2A 2B 2C 2D 2E 2F |
| 30 31 32 33 34 35 36 37 38 39 3A 3B 3C 3D 3E 3F |
| 40 41 42 43 44 45 46 47 48 49 4A 4B 4C 4D 4E 4F |
| 50 51 52 53 54 55 56 57 58 59 5A 5B 5C 5D 5E 5F |

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, jump the JP13 to USART with the jumper cap and connect serial cable to EVAL_COM1. Firstly, all the LEDs are turned on and off for test. Then, the EVAL_COM1 sends the tx_buffer array (from 0x00 to 0xFF) to the serial terminal tool supporting hex format communication and waits for receiving data of the same bytes as tx_buffer from the serial terminal. The data MCU have received is stored in the rx_buffer array. After that, compare tx_buffer with rx_buffer. If tx_buffer is same with rx_buffer, LED1, LED2, LED3 flash by turns. Otherwise, LED1, LED2, LED3 toggle together.

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

5.7.1 DEMO Purpose

This demo includes the following functions of GD32 MCU:

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

5.7.2 DEMO Running Result

Jump the JP13 to USART with the jumper cap, and then download the program <07_ADC_Temperature_Vrefint_Vbat> to the GD32450Z-EVAL board. Connect serial cable to EVAL_COM1, open the HyperTerminal.

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

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

---

<table>
<thead>
<tr>
<th>Temperature Data</th>
<th>Reference Voltage Data</th>
<th>Battery Voltage</th>
</tr>
</thead>
<tbody>
<tr>
<td>24 degrees Celsius</td>
<td>1.198V</td>
<td>3.213V</td>
</tr>
<tr>
<td>25 degrees Celsius</td>
<td>1.201V</td>
<td>3.213V</td>
</tr>
<tr>
<td>25 degrees Celsius</td>
<td>1.193V</td>
<td>3.203V</td>
</tr>
<tr>
<td>25 degrees Celsius</td>
<td>1.198V</td>
<td>3.213V</td>
</tr>
</tbody>
</table>

5.8 ADC0_ADC1_Follow_up_mode

5.8.1 DEMO Purpose

This demo includes the following functions of GD32 MCU:

- Learn to use the ADC to convert analog signal to digital data
5.8.2 DEMO Running Result

Jump the JP13 to USART with the jumper cap, and then download the program <08_ADC0_ADC1_Follow_up_mode> to the GD32450Z-EVAL board. Connect serial cable to EVAL_COM1, open the HyperTerminal. PA3 and PB0 pin voltage access by external voltage, JP23 and JP19 no connect.

TIMER1_CH1 is the trigger source of ADC0 and ADC1. When the rising edge of TIMER1_CH1 coming, ADC0 starts immediately and ADC1 starts after a delay of several ADC clock cycles. The values of ADC0 and ADC1 are transmitted to array adc_value[0] and adc_value [1] by DMA.

When the first rising edge of TIMER1_CH1 coming, the value of the ADC0 conversion of PA3 pin is stored into the low half word of adc_value [0], and after a delay of several ADC clock cycles the value of the ADC1 conversion of PB0 pin is stored into the high half word of adc_value [0]. When the second rising edge of TIMER1_CH1 coming, the value of the ADC0 conversion of PB0 pin is stored into the low half word of adc_value [1], and after a delay of several ADC clock cycles the value of the ADC1 conversion of PA3 pin is stored into the high half word of adc_value [1].

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

```
the data adc_value[0] is 00DE0FF3
the data adc_value[1] is 0FFF00A3
the data adc_value[0] is 00E30FFE
the data adc_value[1] is 0FFF00A4
the data adc_value[0] is 00EA0FF9
the data adc_value[1] is 0FF400B2
the data adc_value[0] is 00DE0FFF
the data adc_value[1] is 0FF500A9
the data adc_value[0] is 00E00FF1
the data adc_value[1] is 0FF500A6
```

5.9 ADC0_ADC1.Regular_Parallel_mode

5.9.1 DEMO Purpose

This demo includes the following functions of GD32 MCU:
5.9.2 DEMO Running Result

Jump the JP13 to USART with the jumper cap, and then download the program <09_ADC0_ADC1-Regular_Parallel_mode> to the GD32450Z-EVAL board. Connect serial cable to EVAL_COM1, open the HyperTerminal. PA3 and PB0 pin connect to external voltage input, JP23 and JP19 no connect.

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

When the first rising edge of TIMER1_CH1 coming, the value of the ADC0 conversion of PA3 pin is stored into the low half word of adc_value [0], the value of the ADC1 conversion of PB0 pin is stored into the high half word of adc_value [0]. When the second rising edge of TIMER1_CH1 coming, the value of the ADC0 conversion of PB0 pin is stored into the low half word of adc_value [1], the value of the ADC1 conversion of PA3 pin is stored into the high half word of adc_value [1].

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

```
the data adc_value[0] is 06210000
the data adc_value[1] is 00000627
the data adc_value[0] is 06290829
the data adc_value[1] is 08400617
the data adc_value[0] is 06250849
the data adc_value[1] is 08500629
the data adc_value[0] is 0628083F
the data adc_value[1] is 08320628
the data adc_value[0] is 06230830
the data adc_value[1] is 08430622
```

5.10 DAC_Output_Voltage_Value

5.10.1 DEMO Purpose

This demo includes the following functions of GD32 MCU:
5.10.2 DEMO Running Result

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

5.11 I2C_EEPROM

5.11.1 DEMO Purpose

This demo includes the following functions of GD32 MCU:

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

5.11.2 DEMO Running Result

Jump the JP13 to USART with the jumper cap, and download the program <11_I2C_EEPROM> to the EVAL board and run. Connect serial cable to COM1, 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 three LEDs lights flashing, otherwise the serial port will output "Err: data read and write aren't matching." and all the three LEDs light.

The output information via the serial port is as following.
5.12 SPI_QSPI_Flash

5.12.1 DEMO Purpose

This demo includes the following functions of GD32 MCU:

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

GD32450Z-EVAL board integrates SPI5 module with Quad-SPI mode and the mode can communicate with external NOR Flash devices. The SPI NOR FLASH is a serial FLASH memory chip GD25Q16B which size is 16Mbit, the chip supports standard SPI and quad SPI operation instructions.

5.12.2 DEMO Running Result

The computer serial port line connected to the COM1 port of development board, set the baud rate of HyperTerminal software to 115200, 8 bits data bit, 1 bit stop bit. At the same time you should jump the JP13 to USART, and jump JP10 to SPI5.

Download the program `<12_SPI_QSPI_Flash>` to the EVAL board, the HyperTerminal software can observe the operation condition and will display the ID of the flash, 256 bytes data which are written to and read from flash. Compare the data that were written to the flash and the data that were read from the flash. If they are the same, the serial port will output "SPI-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.
5.13 I2S_Audio_Player

5.13.1 DEMO Purpose

This Demo includes the following functions of GD32 MCU:

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

GD32450Z-EVAL board integrates the I2S(Inter-IC Sound) module, and the module can communicate with external devices using the I2S audio protocol. This Demo mainly shows how to use the I2S interface of the board for audio output.

5.13.2 DEMO Running Result

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

5.14 EXMC_SDRAM

5.14.1 DEMO Purpose

This demo includes the following functions of GD32 MCU:
- Learn to use EXMC control the SDRAM

5.14.2 DEMO Running Result

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

```plaintext
SDRAM initialized!
SDRAM write data completed!
SDRAM read data completed!
Check the data!
SDRAM test successed!
the data is:
0 1 2 3 4 5 6 7 8 9 a b c d e f
10 11 12 13 14 15 16 17 18 19 1a 1b 1c 1d 1e 1f
20 21 22 23 24 25 26 27 28 29 2a 2b 2c 2d 2e 2f
30 31 32 33 34 35 36 37 38 39 3a 3b 3c 3d 3e 3f
40 41 42 43 44 45 46 47 48 49 4a 4b 4c 4d 4e 4f
50 51 52 53 54 55 56 57 58 59 5a 5b 5c 5d 5e 5f
60 61 62 63 64 65 66 67 68 69 6a 6b 6c 6d 6e 6f
70 71 72 73 74 75 76 77 78 79 7a 7b 7c 7d 7e 7f
```

5.15 EXMC_SDRAM_DeepSleep

5.15.1 DEMO Purpose

This demo includes the following functions of GD32 MCU:
- Learn to use EXMC control the SDRAM
- Learn to use deepsleep mode

5.15.2 DEMO Running Result

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

SDRAM initialized!
SDRAM write data completed!
Enter deepsleep model!
Press the user key to wake up the MCU!

User key has been pressed!
SDRAM read data completed!
Check the data!
SDRAM test succeeded!
The data is:

<table>
<thead>
<tr>
<th>0</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>a</th>
<th>b</th>
<th>c</th>
<th>d</th>
<th>e</th>
<th>f</th>
</tr>
</thead>
<tbody>
<tr>
<td>10</td>
<td>11</td>
<td>12</td>
<td>13</td>
<td>14</td>
<td>15</td>
<td>16</td>
<td>17</td>
<td>18</td>
<td>19</td>
<td>1a</td>
<td>1b</td>
<td>1c</td>
<td>1d</td>
<td>1e</td>
<td>1f</td>
</tr>
<tr>
<td>20</td>
<td>21</td>
<td>22</td>
<td>23</td>
<td>24</td>
<td>25</td>
<td>26</td>
<td>27</td>
<td>28</td>
<td>29</td>
<td>2a</td>
<td>2b</td>
<td>2c</td>
<td>2d</td>
<td>2e</td>
<td>2f</td>
</tr>
<tr>
<td>30</td>
<td>31</td>
<td>32</td>
<td>33</td>
<td>34</td>
<td>35</td>
<td>36</td>
<td>37</td>
<td>38</td>
<td>39</td>
<td>3a</td>
<td>3b</td>
<td>3c</td>
<td>3d</td>
<td>3e</td>
<td>3f</td>
</tr>
<tr>
<td>40</td>
<td>41</td>
<td>42</td>
<td>43</td>
<td>44</td>
<td>45</td>
<td>46</td>
<td>47</td>
<td>48</td>
<td>49</td>
<td>4a</td>
<td>4b</td>
<td>4c</td>
<td>4d</td>
<td>4e</td>
<td>4f</td>
</tr>
<tr>
<td>50</td>
<td>51</td>
<td>52</td>
<td>53</td>
<td>54</td>
<td>55</td>
<td>56</td>
<td>57</td>
<td>58</td>
<td>59</td>
<td>5a</td>
<td>5b</td>
<td>5c</td>
<td>5d</td>
<td>5e</td>
<td>5f</td>
</tr>
<tr>
<td>60</td>
<td>61</td>
<td>62</td>
<td>63</td>
<td>64</td>
<td>65</td>
<td>66</td>
<td>67</td>
<td>68</td>
<td>69</td>
<td>6a</td>
<td>6b</td>
<td>6c</td>
<td>6d</td>
<td>6e</td>
<td>6f</td>
</tr>
<tr>
<td>70</td>
<td>71</td>
<td>72</td>
<td>73</td>
<td>74</td>
<td>75</td>
<td>76</td>
<td>77</td>
<td>78</td>
<td>79</td>
<td>7a</td>
<td>7b</td>
<td>7c</td>
<td>7d</td>
<td>7e</td>
<td>7f</td>
</tr>
</tbody>
</table>

5.16 SDIO_SDCardTest

5.16.1 DEMO Purpose

This demo includes the following functions of GD32 MCU:
- Learn to use SDIO to single block or multiple block write and read
- Learn to use SDIO to erase, lock and unlock a SD card

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

5.16.2 DEMO Running Result

Jump the JP13 to USART to show the print message through HyperTerminal, and download the program <16_SDIO_SDCardTest> to the EVAL board and run. Connect serial cable to EVAL_COM1, open the HyperTerminal. Firstly, all the LEDs are turned on and off for test. Then initialize the card and print out the information of the card. After that, test the function of single block operation, lock and unlock operation, erase operation and multiple blocks operation. If any error occurs, print the error message and turn on
5.17 CAN_Network

5.17.1 DEMO Purpose

This demo includes the following functions of GD32 MCU:
- Learn to use the CAN0 communication between two boards

GD32450Z-EVAL development board integrates the CAN(Controller Area Network) bus controller, which is a common industrial control bus. CAN bus controller follows the CAN bus protocol of 2.0 A and 2.0 B. This demo mainly shows how to communicate between two EVAL boards through CAN0.

5.17.2 DEMO Running Result

This example is tested with two GD32F450-EVAL boards. Jump the JP13 to USART and P2, P3 to CAN with the jumper cap. Connect L pin to L pin and H pin to H pin of JP11 on the boards for sending and receiving frames. Download the program <17_CAN_Network> to the two EVAL boards, and connect serial cable to EVAL_COM1. Firstly, the EVAL_COM1 sends “please press the Tamper key to transmit data!” to the
HyperTerminal. The frames are sent and the transmit data are printed by pressing Tamper Key push button. When the frames are received, the receive data will be printed and the LED2 will toggle a time.

The output information via the serial port is as following.

please press the Tamper key to transmit data!

CAN0 transmit data: ab,cd
CAN0 receive data: ab,cd

5.18 RCU_Clock_Output

5.18.1 DEMO Purpose

This demo includes the following functions of GD32 MCU:
- Learn to use GPIO control the LED
- Learn to use the clock output function of RCU
- Learn to communicate with PC by USART

5.18.2 DEMO Running Result

Jump the JP13 to USART with the jumper cap, and download the program <18_RCU_Clock_Out> to the EVAL board and run. Connect serial cable to EVAL_COM1, open the HyperTerminal. When the program is running, HyperTerminal will display the initial information. Then user can choose the type of the output clock by pressing the TAMPER button. After pressing, the corresponding LED will be turned on and HyperTerminal will display which mode be selected. The frequency of the output clock can be observed through the oscilloscope by PA8 and PC9 pin. Information via a serial port output as following:

/---------- Gigadevice Clock output Demo ----------/
pres tamper key to select clock output source
CK_OUT0: IRC16M, CK_OUT1: system clock/5
CK_OUT0: LXTAL, CK_OUT1: PLLI2SR/5
5.19 CTC_Calibration

5.19.1 DEMO Purpose

This demo includes the following functions of GD32 MCU:
- Learn to use external low speed crystal oscillator (LXTAL) to implement the CTC calibration function
- Learn to use clock trim controller (CTC) to trim internal 48MHz RC oscillator (IRC48M) clock

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

5.19.2 DEMO Running Result

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

5.20 PMU_sleep_wakeup

5.20.1 DEMO Purpose

This demo includes the following functions of GD32 MCU:
- Learn to use the USART receive interrupt to wake up the PMU from sleep mode

5.20.2 DEMO Running Result

Download the program < 20_PMU_sleep_wakeup > to the EVAL board, jump the JP13 to USART with the jumper cap and connect serial cable to EVAL_COM1. After power-on, all the LEDs are off. The mcu will enter sleep mode and the software stop running. When the USART0 receives a byte of data from the HyperTerminal, the mcu will wake up from a receive interrupt. And all the LEDs will flash together.
5.21 FMC_Dualboot

5.21.1 DEMO Purpose

This demo includes the following functions of GD32 MCU:
- Learn to use FMC dual boot mode
- Learn to use FMC option bytes

5.21.2 DEMO Running Result

Jump the JP13 to USART to show the print message through HyperTerminal, and download the program <21_FMC_DualBoot> to the EVAL board and run. Connect serial cable to EVAL_COM1, open the HyperTerminal.
If you want to change the boot mode, press the user key.
If the dual bank mode is disabled, a message will be printed by COM1, information via a serial port output is as following. At the same time the LED1 will flash once.

```
press user key button :
swap BB :
SYSCFG : 0x00000000
OB : 0x5510aaef
ADD1 : 0x200000478
ADD2 : 0x080001bd
```

If the dual bank mode is enabled, a message will be printed by COM1, information via a serial port output is as following. At the same time the LED2 will flash once.

```
press user key button :
swap BB :
SYSCFG : 0x00000000
OB : 0x5510aaef
ADD1 : 0x200000460
ADD2 : 0x1fff019d
```

Warning: When the demo has been tested, you must ensure boot from bank0.
- Information via a serial port output as following indicates boot from bank0:
  press user key button :
  swap BB :
  SYSCFG : 0x00000000
  OB : 0x5510aaef
  ADD1 : 0x20000478
  ADD2 : 0x080001bd
5.22 RTC_Calendar

5.22.1 DEMO Purpose

This demo includes the following functions of GD32 MCU:
- Learn to use RTC module to implement calendar function
- Learn to use USART module to implement time display

5.22.2 DEMO Running Result

Jump the JP13 to USART with the jumper cap, and download the program `<22_RTC_Calendar>` to the EVAL board and run. Connect serial cable to EVAL_COM1, open the HyperTerminal. When the program is running, HyperTerminal display the default time. When you press wakeup key, time will be reconfigured and display.

```
Today is : 2000-01-01
Now Time is : PM 00:00:21
Today is : 2000-01-01
Now Time is : PM 00:00:22
Today is : 2000-01-01
Now Time is : PM 00:00:23
Today is : 2000-01-01
Now Time is : PM 00:00:24
Today is : 2000-01-01
Now Time is : PM 00:00:24
Today is : 2015-08-15
Now Time is : PM 12:00:00
Today is : 2015-08-15
Now Time is : PM 12:00:01
Today is : 2015-08-15
Now Time is : PM 12:00:02
Today is : 2015-08-15
Now Time is : PM 12:00:03
Today is : 2015-08-15
Now Time is : PM 12:00:04
Today is : 2015-08-15
Now Time is : PM 12:00:05
```
5.23 TIMER_Breath_LED

5.23.1 DEMO Purpose

This demo includes the following functions of GD32 MCU:
- Learn to use Timer output PWM wave
- Learn to update channel value

5.23.2 DEMO Running Result

Use the DuPont line to connect the TIMER1_CH2 (PB10) and LED2 (PD4), and then download the program <23_TIMER_Breath_LED> to the GD32450Z-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.

5.24 TLI_IPA

5.24.1 DEMO Purpose

This demo includes the following functions of GD32 MCU:
- Learn to use TLI to control LCD for displaying different images
- Learn to use IPA to process image data

5.24.2 DEMO Running Result

Jump the JP4, JP10, JP14, JP15, JP19, JP20, JP23 to LCD, and download the program <24_TLI_IPA> to the EVAL board and run. After downloading program to board, a running cheetah on the background of GD logo is appeared on the LCD, which outputs as following.
5.25 TRNG_Get_Random

5.25.1 DEMO Purpose

This demo includes the following functions of GD32 MCU:
- Learn to use TRNG generate the random number
- Learn to communicate with PC by USART

5.25.2 DEMO Running Result

Jump the JP13 to USART with the jumper cap, and download the program <25_TRNG_Get_Random> to the EVAL board and run. Connect serial cable to EVAL_COM1, open the serial terminal tool supporting hex format communication. When the program is running, the serial terminal tool will display the initial information. User can use the serial terminal tool to input the minimum and maximum values (for example, the minimum value is 0x03, the maximum value is 0x0F), then application will generate random number in the input range and display it by the serial terminal tool.

Information via a serial port output as following:
5.26 ENET

5.26.1 FreeRTOS_tcpudp

DEMO Purpose

This demo includes the following functions of GD32 MCU:

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

This demo is based on the GD32450Z-EVAL board, it shows how to configure the enet peripherals to send and receive frames in normal mode and use lwip tcp/ip stack to realize ping, telnet and server/client functions.
P4, P5, P6 must be fitted. JP13 jump to Usart.
It is configured in RMII mode, and 25MHz oscillator is used, the system clock is configured to 200MHz.
This demo realizes three applications:

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

2) tcp client application, the eval board acts as tcp client. Users can link the eval board client with the server, using 1026 port. Users can send information from server to client,
then the client will send back the information.

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

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

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

**DEMO Running Result**

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

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

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

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

5.26.2 Raw_tcpudp

DEMO Purpose

This demo includes the following functions of GD32 MCU:
- Learn to use Lwip stack
- Learn to use raw API to handle with a task
- Learn how to realize a tcp server
- Learn how to realize a tcp client
- Learn how to realize a udp server/client
- Learn how to use DHCP to allocate ip address automatically
- Learn to handle with received packet in polling mode and in interrupt mode

This demo is based on the GD32450Z-EVAL board, it shows how to configure the enet peripherals to send and receive frames in normal mode and use lwip tcp/ip stack to realize ping, telnet and server/client functions.
P4, P5, P6 must be fitted. JP13 jump to Usart.
It is configured in RMII mode, and 25MHz oscillator is used, the system clock is configured to 200MHz.
This demo realizes three applications:

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

2) tcp client application, the eval board acts as tcp client. Users can link the eval board client with the server, using 1026 port. Users can send information from server to client, then the client will send back the information. If the server is not online at first, or is break...
during process, when the server is ready again, users can press tamper key to reconnect with server, and communicate.

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

By default, the packet reception is polled in while(1). If users want to receive packet in interrupt service, uncomment the macro defined USE_ENET_INTERRUPT in main.h. If users need dhcp function, it can be configured from the private defines in main.h. This function is closed in default.

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

**DEMO Running Result**

Download the program `<Raw_tcpudp>` to the EVAL board. Using Network assistant software, configure the pc side to tcp client, using 8000 port, and when send something through the assistant, users can see the reply from the server:

Using Network assistant software, configure the pc side to tcp server, using 1026 port, press the Tamper key, and when send something through the assistant, users can see the echo reply from the client:
Using Network assistant software, configure to use udp protocol, using 1025 port, and when send something through the assistant, users can see the echo reply from the board:

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

### 5.26.3 Raw_webserver

#### DEMO Purpose

This demo includes the following functions of GD32 MCU:

- Learn to use Lwip stack
- Learn to use raw API to handle with a task
- Learn how to realize a web server
- Learn how to use a web server to control LEDs
- Learn how to use a web server to monitor the board $V_{REFINT}$ voltage
- Learn how to use DHCP to allocate ip address automatically
- Learn to handle with received packet in polling mode and in interrupt mode
This demo is based on the GD32450Z-EVAL board, it shows how to configure the enet peripherals to send and receive frames in normal mode and use lwip tcp/ip stack to realize webserver application.

P4, P5, P6 must be fitted. JP13 jump to Usart.

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

This demo realizes webserver application:

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

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

By default, the packet reception is polled in while(1). If users want to receive packet in interrupt service, uncomment the macro define USE_ENET_INTERRUPT in main.h. Note: Users should configure ip address, mask and gw of GD32450Z-EVAL board according to the actual net situation from the private defines in main.h.

**DEMO Running Result**

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

The web home page shows as below:

The LED control page shows as below:
The ADC monitor page shows as below:

The value is 1206 mV.

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

5.27 USB_Device

5.27.1 HID_Keyboard

DEMO Purpose

This demo includes the following functions of GD32 MCU:

- Learn how to use the USBFS peripheral mode
Learn how to implement USB HID (human interface) device

GD32450Z-EVAL board has four keys and one USB_FS interface. The four keys are Reset key, Wakeup key, User key and Tamper key. In this demo, the GD32450Z-EVAL board is enumerated as an USB Keyboard, which uses the native PC Host HID driver, as shown below. The USB Keyboard uses three keys (wakeup key, tamper key and user key) to output three characters (‘b’, ‘a’ and ‘c’). In addition, the demo also supports remote wakeup which is the ability of a USB device to bring a suspended bus back to the active condition, and the wakeup key is used as the remote wakeup source.

DEMO Running Result

Before running the demo, please ensure that jumper JP14, JP15 jump to USB_FS. According to the VBUSIG bit in USBFS_GCCFG register, user can decide whether or not to jump JP13 to USB_FS. After doing this, download the program <27_USB_Device\HID_Keyboard> to the EVAL board and run. If you press the Wakeup key, will output ‘b’. If you press the User key, will output ‘c’. If you press the Tamper key, will output ‘a’. 

If you want to test USB remote wakeup function, you can do as follows:
- Manually switch PC to standby mode
- Wait for PC to fully enter the standby mode
- Push the Wakeup key
- If PC is ON, remote wakeup is OK, else failed.

5.27.2 MSC_Udisk

DEMO Purpose

This demo includes the following functions of GD32 MCU:
- Learn how to use the USB_FS peripheral mode
- Learn how to implement USB MSC (mass storage) device

This demo mainly implements a U disk. U disk is currently very widely used removable MSC devices. MSC, the Mass Storage device Class, is a transport protocol between a
computer and mobile devices, which allow a universal serial bus (USB) equipment to access a host computing device, file transfer between them, mainly including mobile hard disk, mobile U disk drive, etc... The MSC device must have a storage medium, and this Demo uses the MCU's internal SRAM as the storage medium. For more details of the MSC protocol please refer to the MSC protocol standard. MSC device will use a variety of transport protocols and command formats for communication, so it need to choose the appropriate protocol and command format in the realization of the application. This Demo selects the BOT (bulk only transport) protocol and the required SCSI (small computer interface) command, and is compatible with a wide variety of Window operating systems. Specific BOT protocol and SCSI command specification please refer to the standard of their agreement.

**DEMO Running Result**

Before running the demo, please ensure that jumper JP14, JP15 jump to USB_FS. According to the VBUSIG bit in USBFS_GCCFG register, user can decide whether or not to jump JP13 to USB_FS. After doing this, download the program `<27_USB_Device\MSC_Udisk>` to the EVAL board and run. When the EV-board connect to the PC, you will find a USB large capacity storage device in the universal serial bus controller, and there is 1 more disk drives in the equipment manager of PC, as shown below:
Then, after opening the resource manager, you will see more of the 1 disk, as shown in the following diagram:

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

5.28.1 HID_Host

**DEMO Purpose**

This demo includes the following functions of GD32 MCU:

- Learn to use the USBFS as a HID host
- Learn the operation between the HID host and the mouse device
- Learn the operation between the HID host and the keyboard device

GD32450Z-EVAL board integrates the USBFS module, and the module can be used as a USB device, a USB host or an OTG device. This demo mainly shows how to use the USBFS as a USB HID host to communicate with external USB HID device.

**DEMO Running Result**

Jump the JP13 to USART to show the print message through HyperTerminal, jump the JP14 and the JP15 to USB_FS. Then open the HyperTerminal and connect the COM1 to the PC through the serial port line. Finally, download the program `<28_USB_Host\HID_Host>` to the EVAL board and run.

If a mouse has been attached, the user will see the information of mouse enumeration. First pressing the user key will see the inserted device is mouse, and then moving the mouse will print the position of the mouse and the state of button in the screen through the serial port.

```
#### USB Host library started ####
> Device Attached
> Reset the USB device.
> Low speed device detected.
> VID: 0450h
> PID: C073h
> HID device connected.
> Manufacturer string is : Logitech
> Product string is : USB Optical Mouse
> Serial Number string is : N/A
> Enumeration completed.
> To start the HID class operations:
> Press User Key...
> Wait for user input!
> User has input!
> HID Demo Device : Mouse.
MoveRight 0 units-------MoveDown 0 units-------Left button is pressed.
MoveRight 0 units-------MoveDown 0 units-------The button is released.
MoveLeft 1 units-------MoveUp 2 units-------No button is pressed.
MoveRight 0 units-------MoveUp 2 units-------No button is pressed.
```

If a keyboard has been attached, the user will see the information of keyboard enumeration. First pressing the user key will see the inserted device is keyboard, and then pressing the keyboard will print the state of the button in the screen through the serial port.
5.28.2 MSC_Host

DEMO Purpose

This demo includes the following functions of GD32 MCU:

- Learn to use the USBS as a MSC host
- Learn the operation between the MSC host and the Udisk

GD32450Z-EVAL board integrates the USBFS module, and the module can be used as a USB device, a USB host or an OTG device. This demo mainly shows how to use the USBFS as a USB MSC host to communicate with external Udisk.

DEMO Running Result

Jump the JP13 to USART to show the print message through HyperTerminal, jump the JP14 to USB_FS and jump the JP15 to USB_FS. Jump the JP13 to USART to show the print message through HyperTerminal, jump the JP14/JP15 to USB_FS, and jump the JP16/JP17/JP18/JP20/P4/P5/P6 to USBHS. Then open the HyperTerminal and connect the COM1 to the PC through the serial port line. Then open the HyperTerminal and connect the COM1 to the PC through the serial port line.

Insert the OTG cable to the USB_FS port and choose the 'GD32450Z_EVAL_Fullspeed' project in the program <28_USB_HostMSC_Host>, or insert the OTG cable to the USB_HS port and choose the 'GD32450Z_EVAL_Highspeed' project in the program <28_USB_HostMSC_Host>. Then download it to the EVAL board and run.

If an Udisk has been attached, the user will see the information of Udisk enumeration. First pressing the user key will see the Udisk information, next pressing the tamper key will see the root content of the Udisk, then press the wakeup key will write file to the Udisk, finally the user will see information that the msc host demo is end.
### USB Host library started ###
> Device Attached
> Reset the USB device
> Full speed device detected
> VID: 0701h
> PID: 5501h
> Mass storage device connected
> manufacture string is: SanDisk
> product string is: SanDisk Ultra
> Serial Number string is: A20042C2D5035555
> Enumeration completed
> To see the disk information:
> Press User Key...
> File System initialized
> Disk capacity: 28995657204 bytes.
> Exploring disk flash ...
> To see the root content of the disk
> Press Tamper key...

```text
G032.TXT
```

SANDES™
> Press Wakeup Key to write file
> Writing File to disk flash ...
> G032.TXT created in the disk.
> The MSC host demo is end.

---

## 6 Revision history

### Table 2 Revision history

<table>
<thead>
<tr>
<th>Revision No.</th>
<th>Description</th>
<th>Date</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.0</td>
<td>Initial Release</td>
<td>Oct. 19, 2016</td>
</tr>
</tbody>
</table>

---