PY32F403 series
32-bit ARM® Cortex®-M4 microcontroller

HAL Library Sample Manual

1 ADC

1.1 ADC_AnalogWatchdog

此样例演示了 ADC 的模拟看门狗功能，当开启看门狗的通道的电压值不在设定的上下限中，会进入看门狗中断。

This example demonstrates the analog watchdog function of ADC. When the voltage value of the channel that opens the watchdog is not within the set upper or lower limits,Will enter watchdog interrupt.

1.2 ADC_DualModeRegsimult

此样例演示了 ADC1 和 ADC2 的同步规则功能。

This example demonstrates the synchronization rule function of ADC1 and ADC2.

1.3 ADC_MultiChannelsSingleConversion_TriggerSW_DMA

此样例演示了 ADC 的多通道 DMA 传输的功能。

This example demonstrates the functionality of multi-channel DMA transmission in ADC.

1.4 ADC_SingleConversion_TriggerTimer_IT

此样例演示了 ADC 通过 TIM 触发转换的功能，每隔 1s，TIM 触发 ADC 转换一次，并通过串口打印出来。

This example demonstrates the function of ADC triggering conversion through TIM. Every 1 second, TIM triggers ADC conversion and prints it out through the serial port.

1.5 ADC_TempSensor

此样例演示了 ADC 模块的 Tempsensor 功能，并通过串口打印出温度值。

This example demonstrates the Tempsensor function of the ADC module and prints the temperature value through the serial port.

1.6 ADC_Vrefint

此样例演示了 ADC 的 Vrefint 功能，通过 Vrefint 的值，可以反推出 MCU 的供电电压值。

This example demonstrates the Vrefint function of ADC. By using the value of Vrefint, the power supply voltage value of MCU can be inferred

2 CANFD

2.1 CANFD_BaseID_polling

此样例演示了采用 CANFD 协议标准帧轮询方式与 PCAN-View 的通信功能，MCU 首先自动向 PCAN-View 发送 64byte 数据 0x0~0x3F，PCAN-View 接收到数据后，然后手动通过 PCAN-View 向 MCU 发送 ID 为 0x12F 的 64byte 数据，MCU 会自动将接收到数据通过串口打出。

This example demonstrates the communication function of using the CANFD protocol standard frame polling method with PCAN-View. The MCU first automatically sends 64byte data 0x0~0x3F to PCAN-View. After PCAN-View receives the data, it manually sends 64byte data with ID 0x12F to the MCU through PCAN-View. The MCU will automatically print the received data through the serial port.

2.2 CANFD_Classic_BaseID_polling

此样例演示了采用 CAN2.0 协议标准帧轮询方式与 PCAN-View 的通信功能，MCU 首先自动向 PCAN-View 发送 8byte 数据 0x1~0x8，PCAN-View 接收到数据后，然后手动通过 PCAN-View 向 MCU 发送 ID 为 0x12F 的 8byte 数据，MCU 会自动将接收到数据通过串口打出。

This example demonstrates the communication function with PCAN View using the CAN2.0 protocol standard frame polling method. The MCU first automatically sends 8-byte data 0x1~0x8 to PCAN View. After PCAN View receives the data, it manually sends 8-byte data with ID 0x12F to the MCU through PCAN View. The MCU will automatically print the received data through the serial port.

2.3 CANFD_ExtendedID_IT

此样例演示了采用 CANFD 协议扩展帧中断方式与 PCAN-View 的通信功能，MCU 首先自动向 PCAN-View 发送 64byte 数据 0x0~0x3F，PCAN-View 接收到数据后，然后手动通过 PCAN-View 向 MCU 发送 ID 为 0x1234567F 的 64byte 数据，MCU 会自动将接收到数据通过串口打出。

This example demonstrates the communication function between the CANFD protocol extension frame interrupt method and the PCAN View. The MCU first automatically sends 64byte data 0x0~0x3F to the PCAN View. After the PCAN View receives the data, it manually sends 64byte data with ID 0x1234567F to the MCU through the PCAN View. The MCU will automatically print the received data through the serial port.

2.4 CANFD_ExtendedID_LBME_polling

此样例演示了采用 CANFD 协议、扩展帧、外部回环的轮询方式与 PCAN-View 的通信功能，MCU 首先自动向发送 64byte 数据 0x0~0x3f，MCU 接收到数据后，自动将接收到数据通过串口打出。

This example demonstrates the communication function with PCAN-View using the CANFD protocol, extended frames, and external loopback polling method. The MCU first automatically sends 64byte data 0x0~0x3f to the. After receiving the data, the MCU automatically prints the received data through the serial port.

2.5 CANFD_ExtendedID_LBMI_polling

此样例演示了采用 CANFD 协议、扩展帧、内部回环的轮询方式与 PCAN-View 的通信功能，MCU 首先自动向发送 64byte 数据 0x0~0x3f，MCU 接收到数据后，自动将接收到数据通过串口打出。

This example demonstrates the communication function with PCAN-View using the CANFD protocol, extended frames, and internal loopback polling method. The MCU first automatically sends 64byte data 0x0~0x3f to the. After receiving the data, the MCU automatically prints the received data through the serial port.

3 CRC

3.1 CRC_CalculateCheckValue

此样例演示了 CRC 校验功能，通过对一个数组里的数据进行校验，得到的校验值与理论校验值进行比较，相等则 LED 灯亮，否则 LED 灯熄灭。

This example demonstrates the CRC verification function. By verifying the data in an array, the obtained verification value is compared with the theoretical verification value. If it is equal, the LED light will be on, otherwise the LED light will be off.

4 CTC

4.1 CTC_Autotrim

此样例演示了 CTC 使用 LSE 做参考时钟自动校准 HSI48M 时钟的功能。

This example demonstrates the function of CTC using LSE as a reference clock to automatically calibrate the HSI48M clock.

5 DAC

5.1 DAC_SingleGeneration

此样例演示了 DAC 的软件触发功能,通道 PA4 能够输出 1/2 的供电电压值。

This example demonstrates the software triggering function of DAC,channel PA4 can output 1/2 of the power supply voltage value.

5.2 DAC_TIMTrigger_DMA

此样例演示了 DAC 的 TIM 触发功能，每隔 1s，DAC 的通道 1 输出数组中的数据。

This example demonstrates the TIM triggering function of DAC, channel 1 of DAC outputs data from the array every 1 second.

6 DMA

6.1 DMA_SramToSram

此样例演示了 DMA 从 SRAM 到 SRAM 传输数据的功能（SRAM 和外设之间传输的样例请参考相关外设样例工程）。

This example demonstrates the function of DMA transferring data from SRAM to SRAM (please refer to the relevant peripheral sample project for the example of transfer between SRAM and peripherals).

7 ESMC

7.1 ESMC_MemoryMapping

此样例演示了 ESMC 的 memory mapping 功能，把预先编译好的 bin 文件，下载到 P25Q64 芯片中，然后把 P25Q64 地址映射到 0x00000000，主程序跳转到 0x00000000 地址开始执行 bin 文件中的程序，样例中 bin 程序执行的任务是闪烁 LED（PA01）灯。

This example demonstrates the memory mapping function of ESMC. The pre compiled bin file is downloaded to the P25Q64 chip, and then the P25Q64 address is mapped to 0x000000. The main program jumps to the 0x000000 address to start executing the program in the bin file. In the example, the bin program executes the task of flashing the LED (PA01) light.

7.2 ESMC_ReadWriteQuad_IT

此样例演示了 ESMC 在间接模式下的中断数据传输功能，对 P25Q64 的芯片进行擦除，写入数据，读取数据，然后把读取的数据和写入的数据进行对比，数据正确则 LED 灯闪烁，否则 LED 灯不闪烁。

This example demonstrates the interrupt data transmission function of ESMC in indirect mode, erasing, writing data, reading data on the P25Q64 chip, and then comparing the read data with the written data. If the data is correct, the LED light will flash, otherwise the LED light will not flash.

7.3 ESMC_ReadWrite_DMA

此样例演示了 ESMC 在间接模式下的 DMA 数据传输功能，对 P25Q64 的芯片进行擦除，写入数据，读取数据，然后把读取的数据和写入的数据进行对比，数据正确则 LED 灯闪烁，否则 LED 灯不闪烁。

This example demonstrates the DMA data transmission function of ESMC in indirect mode, erasing, writing data, reading data on the P25Q64 chip, and then comparing the read data with the written data. If the data is correct, the LED light will flash, otherwise the LED light will not flash.

7.4 ESMC_ReadWrite_DTR_Polling

此样例演示了 ESMC 在间接模式下的 DTR 的 polling 传输功能，对 P25Q64SH 的芯片进行擦除，写入数据，DTR 读取数据，然后把读取的数据和写入的数据进行对比，数据正确则 LED 灯闪烁，否则 LED 灯不闪烁。

This example demonstrates the polling transmission function of ESMC's DTR in indirect mode, erasing the chip of P25Q64SH, writing data, DTR reading data, and then comparing the read data with the written data. If the data is correct, the LED light will flash, otherwise the LED light will not flash.

7.5 ESMC_ReadWrite_Polling

此样例演示了 ESMC 在间接模式下的 polling 传输功能，对 P25Q64 的芯片进行擦除，写入数据，读取数据，然后把读取的数据和写入的数据进行对比，数据正确则 LED 灯闪烁，否则 LED 灯不闪烁。

This example demonstrates the polling transmission function of ESMC in indirect mode, erasing, writing data, reading data on the P25Q64 chip, and then comparing the read data with the written data. If the data is correct, the LED light will flash, otherwise the LED light will not flash.

7.6 ESMC_ReadWrite_TwoFlash_Polling

此样例演示了 ESMC 在间接模式下的双 Flash polling 传输功能，对两个 P25Q64 的芯片进行擦除，写入数据，读取数据，然后把读取的数据和写入的数据进行对比，数据正确则 LED 灯闪烁，否则 LED 灯不闪烁。

This example demonstrates the dual Flash polling transmission function of ESMC in indirect mode, erasing, writing data, reading data on two P25Q64 chips, and then comparing the read data with the written data. If the data is correct, the LED light will flash, otherwise the LED light will not flash.

8 EXTI

8.1 EXTI_ToggleLed_IT

此样例演示了 GPIO 外部中断功能，PA0 引脚上的每一个下降沿都会产生中断，中断函数中 LED 灯会翻转一次。

This example demonstrates the GPIO external interrupt function, where each falling edge on the key (PA0) pin generates an interrupt, and the LED light in the interrupt function toggle once.

8.2 EXTI_WakeUp_Event

此样例演示了通过 PA6 引脚唤醒 MCU 的功能。下载程序并运行后，LED 灯处于常亮状态；按下用户按键后，LED 灯处于常暗状态，且 MCU 进入 STOP 模式；拉低 PA6 引脚后，MCU 唤醒，LED 灯处于闪烁状态。

This example demonstrates the function of waking up an MCU through the PA6 pin. After downloading the program and running it, the LED light is constantly on; After pressing the user button, the LED light is in a constant dark state and the MCU enters STOP mode; After pulling down the PA6 pin, the MCU wakes up and the LED light is in a flashing state.

9 FLASH

9.1 Flash_BlockEraseAndWrite

此样例演示了 flash block 擦除和 page 写功能。

This example demonstrates the flash block erase and page write functions.

9.2 Flash_OptionByteWrite_RST_STOP

此样例演示了通过软件方式将进入 STOP 模式改为复位芯片。

This example demonstrates how to change the mode of entering STOP to resetting the chip through software

9.3 Flash_PageEraseAndWrite

此样例演示了 flash page 擦除和 page 写功能。

This example demonstrates the flash page erase and page write functions.

9.4 Flash_SectorEraseAndWrite

此样例演示了 flash sector 擦除和 page 写功能。

This example demonstrates the flash sector erase and page write functions.

10 GPIO

10.1 GPIO_Toggle

此样例演示了 GPIO 输出模式，配置 LED 引脚为数字输出模式，并且每隔 250ms 翻转一次 LED 引脚电平。运行程序后，可以看到 LED 灯以 2Hz 的频率闪烁。

This sample demonstrates the GPIO output mode. It configures the LED pin as a digital output mode and toggles the LED pin level every 250ms. After running the program, you can observe the LED blinking at a frequency of 2Hz.

11 I2C

11.1 I2C_TwoBoard_CommunicationMaster_10BitAddr_IT

此样例演示了 I2C 通过中断方式进行通讯，主机先向从机发送 15byte 数据，然后再接收从机发送的 15byte 数据,主机、从机接收数据成功后，主机和从机板上的小灯处于“常亮”状态。

This sample demonstrates communication between I2C devices using interrupts. The master device sends 15 bytes of data to the slave device and then receives 15 bytes of data from the slave. After successful data transmission and reception between the master and slave, the LEDs on both boards remain constantly lit.

11.2 I2C_TwoBoard_CommunicationMaster_DMA

此样例演示了 I2C 通过 DMA 方式进行通讯，主机先向从机发送 15byte 数据，然后再接收从机发送的 15byte 数据,主机、从机接收数据成功后，主机和从机板上的小灯处于“常亮”状态。

This sample demonstrates communication between I2C devices using DMA. The master device sends 15 bytes of data to the slave device and then receives 15 bytes of data from the slave. After successful data transmission and reception between the master and slave, the LEDs on both boards remain constantly lit.

11.3 I2C_TwoBoard_CommunicationMaster_DMA_MEM

此样例演示了主机 I2C 通过 DMA 方式进行通讯，从机使用 EEPROM 外设芯片 P24C32，按下 user 按键，主机先向从机写 15bytes 数据为 0x1~0xf，然后再从 EEPROM 中将写入的数据读出，读取成功后，主机板上的小灯处于“常亮”状态。

This sample demonstrates communication between the master device using I2C and the slave device using the EEPROM peripheral chip P24C32. When the user button on the master device is pressed, the master device first writes 15 bytes of data to the slave device, ranging from 0x1 to 0xF. Then it reads the written data from the EEPROM. Once the data is successfully read, the LED on the master board will remain constantly lit.

11.4 I2C_TwoBoard_CommunicationMaster_DualAddr_IT

此样例演示了 I2C 通过中断方式进行通讯，主机先向从机发送 15byte 数据，然后再接收从机发送的 15byte 数据,主机、从机接收数据成功后，主机和从机板上的小灯处于“常亮”状态。

This sample demonstrates communication between I2C devices using interrupts. The master device sends 15 bytes of data to the slave device and then receives 15 bytes of data from the slave. After successful data transmission and reception between the master and slave, the LEDs on both boards remain constantly lit.

11.5 I2C_TwoBoard_CommunicationMaster_IT

此样例演示了 I2C 通过中断方式进行通讯，主机先向从机发送 15byte 数据，然后再接收从机发送的 15byte 数据,主机、从机接收数据成功后，主机和从机板上的小灯处于“常亮”状态。

This sample demonstrates communication between I2C devices using interrupts. The master device sends 15 bytes of data to the slave device and then receives 15 bytes of data from the slave. After successful data transmission and reception between the master and slave, the LEDs on both boards remain constantly lit.

11.6 I2C_TwoBoard_CommunicationMaster_Polling

此样例演示了 I2C 通过轮询方式进行通讯，主机先向从机发送 15byte 数据，然后再接收从机发送的 15byte 数据,主机、从机接收数据成功后，主机和从机板上的小灯处于“常亮”状态。

This sample demonstrates communication between I2C devices using polling. The master device sends 15 bytes of data to the slave device and then receives 15 bytes of data from the slave. After successful data transmission and reception between the master and slave, the LEDs on both boards remain constantly lit.

11.7 I2C_TwoBoard_CommunicationSlave_10BitAddr_IT

此样例演示了 I2C 通过中断方式进行通讯，主机先向从机发送 15byte 数据，然后再接收从机发送的 15byte 数据,主机、从机接收数据成功后，主机和从机板上的小灯处于“常亮”状态。

This sample demonstrates communication between I2C devices using interrupts. The master device sends 15 bytes of data to the slave device and then receives 15 bytes of data from the slave. After successful data transmission and reception between the master and slave, the LEDs on both boards remain constantly lit.

11.8 I2C_TwoBoard_CommunicationSlave_DMA

此样例演示了 I2C 通过 DMA 方式进行通讯，主机先向从机发送 15byte 数据，然后再接收从机发送的 15byte 数据,主机、从机接收数据成功后，主机和从机板上的小灯处于“常亮”状态。

This sample demonstrates communication between I2C devices using DMA. The master device sends 15 bytes of data to the slave device and then receives 15 bytes of data from the slave. After successful data transmission and reception between the master and slave, the LEDs on both boards remain constantly lit.

11.9 I2C_TwoBoard_CommunicationSlave_DualAddr_IT

此样例演示了 I2C 通过中断方式进行通讯，主机先向从机发送 15byte 数据，然后再接收从机发送的 15byte 数据,主机、从机接收数据成功后，主机和从机板上的小灯处于“常亮”状态。

This sample demonstrates communication between I2C devices using interrupts. The master device sends 15 bytes of data to the slave device and then receives 15 bytes of data from the slave. After successful data transmission and reception between the master and slave, the LEDs on both boards remain constantly lit.

11.10 I2C_TwoBoard_CommunicationSlave_IT

此样例演示了 I2C 通过中断方式进行通讯，主机先向从机发送 15byte 数据，然后再接收从机发送的 15byte 数据,主机、从机接收数据成功后，主机和从机板上的小灯处于“常亮”状态。

This sample demonstrates communication between I2C devices using interrupts. The master device sends 15 bytes of data to the slave device and then receives 15 bytes of data from the slave. After successful data transmission and reception between the master and slave, the LEDs on both boards remain constantly lit.

12 I2S

12.1 I2S_TwoBoard_CommunicationMaster_DMA

此样例是对I2S主机与I2S从机以DMA方式进行通信的演示，I2S主机先向I2S从机发送数据0x1~0x10，I2S 从机接收到数据后，再向 I2S 主机回发数据 0x0x1~0x10，当 I2S 主机、I2S 从机成功接收数据时，小灯处于常亮状态，否则小灯处于闪烁状态。

This sample demonstrates communication between the I2S master and I2S slave using DMA. The I2S master sends data 0x1 to 0x10 to the I2S slave. The I2S slave receives the data and sends back data 0x1 to 0x10 to the I2S master. When both the I2S master and I2S slave successfully receive the data, the LED will be constantly on. Otherwise, the LED will be blinking.

12.2 I2S_TwoBoard_CommunicationMaster_IT

此样例是对I2S主机与I2S从机以中断方式进行通信的演示，I2S主机先向I2S从机发送数据0x1~0x10，I2S 从机接收到数据后，再向 I2S 主机回发数据 0x0x1~0x10，当 I2S 主机、I2S 从机成功接收数据时，小灯处于常亮状态，否则小灯处于闪烁状态。

This sample demonstrates communication between the I2S master and I2S slave using interrupts. The I2S master sends data 0x1 to 0x10 to the I2S slave. The I2S slave receives the data and sends back data 0x1 to 0x10 to the I2S master. When both the I2S master and I2S slave successfully receive the data, the LED will be constantly on. Otherwise, the LED will be blinking.

12.3 I2S_TwoBoard_CommunicationMaster_Polling

此样例是对 I2S 主机与 I2S 从机以 polling 方式进行通信的演示，I2S 主机先向 I2S 从机发送数据 0x1~0x10，I2S 从机接收到数据后，再向 I2S 主机回发数据 0x0x1~0x10，当 I2S 主机、I2S 从机成功接收数据时，小灯处于常亮状态，否则小灯处于闪烁状态。

This sample demonstrates communication between the I2S master and I2S slave using polling. The I2S master sends data 0x1 to 0x10 to the I2S slave. The I2S slave receives the data and sends back data 0x1 to 0x10 to the I2S master. When both the I2S master and I2S slave successfully receive the data, the LED will be constantly on. Otherwise, the LED will be blinking.

12.4 I2S_TwoBoard_CommunicationSlave_DMA

此样例是对I2S主机与I2S从机以DMA方式进行通信的演示，I2S主机先向I2S从机发送数据0x1~0x10，I2S 从机接收到数据后，再向 I2S 主机回发数据 0x0x1~0x10，当 I2S 主机、I2S 从机成功接收数据时，小灯处于常亮状态，否则小灯处于闪烁状态。

This sample demonstrates communication between the I2S master and I2S slave using DMA. The I2S master sends data 0x1 to 0x10 to the I2S slave. The I2S slave receives the data and sends back data 0x1 to 0x10 to the I2S master. When both the I2S master and I2S slave successfully receive the data, the LED will be constantly on. Otherwise, the LED will be blinking.

12.5 I2S_TwoBoard_CommunicationSlave_IT

此样例是对I2S主机与I2S从机以中断方式进行通信的演示，I2S主机先向I2S从机发送数据0x1~0x10，I2S 从机接收到数据后，再向 I2S 主机回发数据 0x0x1~0x10，当 I2S 主机、I2S 从机成功接收数据时，小灯处于常亮状态，否则小灯处于闪烁状态。

This sample demonstrates communication between the I2S master and I2S slave using interrupts. The I2S master sends data 0x1 to 0x10 to the I2S slave. The I2S slave receives the data and sends back data 0x1 to 0x10 to the I2S master. When both the I2S master and I2S slave successfully receive the data, the LED will be constantly on. Otherwise, the LED will be blinking.

12.6 I2S_TwoBoard_CommunicationSlave_Polling

此样例是对 I2S 主机与 I2S 从机以 polling 方式进行通信的演示，I2S 主机先向 I2S 从机发送数据 0x1~0x10，I2S 从机接收到数据后，再向 I2S 主机回发数据 0x0x1~0x10，当 I2S 主机、I2S 从机成功接收数据时，小灯处于常亮状态，否则小灯处于闪烁状态。

This sample demonstrates communication between the I2S master and I2S slave using polling. The I2S master sends data 0x1 to 0x10 to the I2S slave. The I2S slave receives the data and sends back data 0x1 to 0x10 to the I2S master. When both the I2S master and I2S slave successfully receive the data, the LED will be constantly on. Otherwise, the LED will be blinking.

13 IWDG

13.1 IWDG_Reset

此样例演示了 IWDG 看门狗功能，配置看门狗重载计数值，计数 800ms 后复位，然后通过调整每次喂狗的时间（main 函数 while 循环中代码），可以观察到，如果每次喂狗时间 750ms，程序能一直正常运行（LED 灯闪烁），如果喂狗时间 850ms，程序会一直复位（LED 灯熄灭）。

This example demonstrates the function of IWDG (Independent Watchdog).Set IWDG to count 800ms and then reset.By adjusting the time of refresh the dog each time (code in the main function while loop), it can be observed that if the time is 750ms, the program can always run normally (LED blink), if the time is 850ms, the program will always reset (LED off).

14 PWR

14.1 PWR_PVD

此样例演示了 PVD 电压检测功能。当供电电压低于 3.0V 时，LED 会点亮，高于 3.0V 时，LED 灯会熄灭。

This sample demonstrates the PVD (Programmable Voltage Detector) voltage detection functionality. When the supply voltage is lower than 3.0V, the LED will light up. When the supply voltage is higher than 3.0V, the LED will turn off.

14.2 PWR_SLEEP_WFE

此样例演示了 sleep 模式下，通过 GPIO 事件唤醒功能。

This sample demonstrates the GPIO wake-up feature in sleep mode. When the program starts running, the LED will be on. Press the button, and the LED will turn off, entering sleep mode. When a falling edge is detected on PA06, the program will exit sleep mode. The LED will toggle at intervals of 500ms.

14.3 PWR_SLEEP_WFI

此样例演示了 sleep 模式下，GPIO 外部中断唤醒功能。

This sample demonstrates the GPIO external interrupt wake-up feature in sleep mode. When the program starts running, the LED will be on. Press the button, and the LED will turn off, entering sleep mode. When a falling edge is detected on PA06, the program will exit sleep mode. The LED will toggle at intervals of 500ms.

14.4 PWR_STANDBY

此样例演示了 standby 模式下，通过 wakeuppin 唤醒功能。

This sample demonstrates the wake-up feature using the wakeup pin in standby mode. When the program starts running, the LED will be on. Press the button, and the LED will turn off, entering standby mode. When a rising edge is detected on PA02, the program will exit standby mode. The LED will blink for 5 seconds and then be turned on again.

14.5 PWR_STOP_WFE

此样例演示了 stop 模式下，通过 GPIO 事件唤醒功能。

This sample demonstrates the GPIO event wake-up feature in stop mode.

14.6 PWR_STOP_WFI

此样例演示了 stop 模式下，GPIO 外部中断唤醒功能。

This sample demonstrates the GPIO external interrupt wake-up feature in stop mode.

15 RCC

15.1 RCC_HSE_Div

此样例配置系统时钟为 HSE，并通过 MCO（PA08）引脚输出。

This sample configures the system clock to use the HSE (High-Speed External) clock source and outputs it through the MCO (PA08) pin.

15.2 RCC_HSI_Output

此样例配置系统时钟为 HSI，并通过 MCO（PA08）引脚输出。

This sample configures the system clock to use the HSI (High-Speed Internal) clock source and outputs it through the MCO (PA08) pin.

15.3 RCC_PLLOutput

此样例配置系统时钟为 PLL，并通过 MCO（PA08）引脚输出，PLL 的输入时钟源选择 HSE。

This sample configures the system clock to use the PLL (Phase-Locked Loop) clock source with HSE (High-Speed External) as the input clock source and outputs it through the MCO (PA08) pin. The PLL frequency is configured to be 144MHz.

16 RTC

16.1 RTC_AlarmSecond_IT

此样例演示 RTC 的秒中断和闹钟中断功能，每次秒中断，在中断函数中会打印字符“RTC_IT_SEC”，并且输出实时时间。

This sample demonstrates the second interrupt and alarm interrupt functionality of the RTC. Each time the second interrupt occurs, the interrupt function will print the string "RTC_IT_SEC" and output the current RTC count time.

16.2 RTC_WakeUpAlarm

此样例演示通过 RTC 闹钟中断每隔 1S 将 MCU 从 STOP 模式下唤醒，每次唤醒会翻转 LED，LED 翻转间隔为 1s。

This example demonstrates waking the MCU from STOP mode every 1 second using an RTC alarm clock interrupt. Each wake-up will flip the LED, with an LED flip interval of 1 second.

16.3 RTC_WakeUpSecond

此样例演示了通过 RTC 的秒中断唤醒 MCU 的功能。下载程序并运行后，LED 灯处于常亮状态；按下用户按键后，LED 灯处于常暗状态，且 MCU 进入 STOP 模式；RTC 秒中断唤醒 MCU 后，LED 灯处于闪烁状态。

This sample demonstrates the functionality of waking up the MCU using the RTC second interrupt. After downloading and running the program, the LED will be constantly on. Pressing the user button will turn off the LED and put the MCU into the STOP mode. When the RTC second interrupt wakes up the MCU, the LED will blink.

17 SPI

17.1 SPI_TwoBoards_FullDuplexMaster_DMA

此样例是对串口外设接口（SPI）与外部设备以全双工串行方式进行通信的演示,此接口设置为主模式，为外部从设备提供通信时钟 SCK。主机通过 MOSI 引脚发送数据,从 MISO 引脚接收从机的数据，数据以主机提供的 SCK 沿同步被移位，完成全双工通信。

This example demonstrates that SPI communicates with external devices in full duplex serial mode.This interface is set in master mode to provide communication clock SCK for external and slave devices.The host sends data through the MOSI pin, receives data from the MISO pin, and the data is shifted synchronously along the SCK provided by the host, completing the full-duplex communication.

17.2 SPI_TwoBoards_FullDuplexMaster_IT

此样例是对串口外设接口（SPI）与外部设备以全双工串行方式进行通信的演示,此接口设置为主模式，为外部从设备提供通信时钟 SCK。主机通过 MOSI 引脚发送数据,从 MISO 引脚接收从机的数据，数据以主机提供的 SCK 沿同步被移位，完成全双工通信。

This example demonstrates that SPI communicates with external devices in full duplex serial mode.This interface is set in master mode to provide communication clock SCK for external and slave devices.The host sends data through the MOSI pin, receives data from the MISO pin, and the data is shifted synchronously along the SCK provided by the host, completing the full-duplex communication.

17.3 SPI_TwoBoards_FullDuplexMaster_Polling

此样例是对串口外设接口（SPI）与外部设备以全双工串行方式进行通信的演示,此接口设置为主模式，为外部从设备提供通信时钟 SCK。主机通过 MOSI 引脚发送数据,从 MISO 引脚接收从机的数据，数据以主机提供的 SCK 沿同步被移位，完成全双工通信。

This example demonstrates that SPI communicates with external devices in full duplex serial mode.This interface is set in master mode to provide communication clock SCK for external and slave devices.The host sends data through the MOSI pin, receives data from the MISO pin, and the data is shifted synchronously along the SCK provided by the host, completing the full-duplex communication.

17.4 SPI_TwoBoards_FullDuplexSlave_DMA

此样例是对串口外设接口（SPI）与外部设备以全双工串行方式进行通信的演示,此接口设置为主模式，为外部从设备提供通信时钟 SCK。主机通过 MOSI 引脚发送数据,从 MISO 引脚接收从机的数据，数据以主机提供的 SCK 沿同步被移位，完成全双工通信。

This example demonstrates that SPI communicates with external devices in full duplex serial mode.This interface is set in master mode to provide communication clock SCK for external and slave devices.The host sends data through the MOSI pin, receives data from the MISO pin, and the data is shifted synchronously along the SCK provided by the host, completing the full-duplex communication.

17.5 SPI_TwoBoards_FullDuplexSlave_IT

此样例是对串口外设接口（SPI）与外部设备以全双工串行方式进行通信的演示,此接口设置为主模式，为外部从设备提供通信时钟 SCK。主机通过 MOSI 引脚发送数据,从 MISO 引脚接收从机的数据，数据以主机提供的 SCK 沿同步被移位，完成全双工通信。

This example demonstrates that SPI communicates with external devices in full duplex serial mode.This interface is set in master mode to provide communication clock SCK for external and slave devices.The host sends data through the MOSI pin, receives data from the MISO pin, and the data is shifted synchronously along the SCK provided by the host, completing the full-duplex communication.

17.6 SPI_TwoBoards_FullDuplexSlave_Polling

此样例是对串口外设接口（SPI）与外部设备以全双工串行方式进行通信的演示,此接口设置为主模式，为外部从设备提供通信时钟 SCK。主机通过 MOSI 引脚发送数据,从 MISO 引脚接收从机的数据，数据以主机提供的 SCK 沿同步被移位，完成全双工通信。

This example demonstrates that SPI communicates with external devices in full duplex serial mode.This interface is set in master mode to provide communication clock SCK for external and slave devices.The host sends data through the MOSI pin, receives data from the MISO pin, and the data is shifted synchronously along the SCK provided by the host, completing the full-duplex communication.

18 TIM

18.1 TIM1_6Step

此样例是对高级定时器功能"六步 PWM 的产生"的演示，通过 systick 中断作为 COM commutation 事件的触发源，实现（无刷电机的）换向,下表是换向步骤，比如第一步中的 CH1 和 CH3N 为 1，即设置打开这两个通道的 PWM 输出。

This sample demonstrates advanced timer function 'six-step PWM generation', systick interrupt as COM commutation event trigger source to achieve commutation (brushless motor). The following table shows the commutating steps. For example, CH1 and CH3N in the first step are set to 1, that mean the PWM output of these two channels is set to start

18.2 TIM1_AutoReloadPreload

此样例实现了定时器的基本计数功能，以及演示了 ARR 自动重载功能，样例在定时器重载中断中翻转 LED 灯。

This sample demonstrates base count function of the timer,and show ARR register autoreload function.Example toggle LED in timer update interrupt.

18.3 TIM1_ComplementarySignals_DeadTime

此样例实现了定时器的刹车功能，CH1 和 CH1N 互补 pwm 输出，接收到外部 IO 口的刹车信号（低电平）后，PWM 信号关闭，由于 BDTR.AOE 置位，所以刹车信号取消（高电平）后，继续 pwm 输出，此样例实现了死区功能。

This example realizes the braking function of the timer, CH1 and CH1N complement pwm output, after receiving the brake signal (low level) of the external IO port the PWM signal is turned off.Because the BDTR.AOE is set, so after the brake signal is cancelled (high level), and the pwm output continues. The dead time is inserted in the complementary output of CH1 and CH1N

18.4 TIM1_DmaBurst_twice

此样例演示了在 TIM1 中使用 DMA 连续两次 burst 传输数据的功能，burst 每传输一次更新三个寄存器，PSC,ARR,RCR，在更新事件中断中，LED 会闪烁，通过逻辑分析仪监测，可看到 PA1 的翻转间隔会从第一次的 2s，第二次 2s，第三次 1s,第四次及后续变为 0.5s，此时两次 burst 传输完成，并且 PCS,ARR,RCR 均更新完毕。

This sample demonstrates the function to transfer data in TIM1 using DMA in two consecutive bursts.burst updates three registers(PSC,ARR,RCR) per transfer.In the interruption of update event, LED will be blinked. Through the monitoring of logic analyzer, it can be seen that the flipping interval of PA1 will change from 2s for the first time, 2s for the second time, 1s for the third time, and 0.5s for the fourth and subsequent times. At this time, the two burst transmission is completed, and PCS、ARR and RCR are all updated.

18.5 TIM1_EncoderTI2AndTI1

此样例实现了 TIM1 中的编码器计数功能，TI1(PA8)和 TI2(PA9)作为编码器输入引脚，通过 CNT 寄存器可观察到计数器变化，通过 uwDirection 变量可观察到计数器的计数方向，通过打印数据也可观察计数方向和 CNT 寄存器计数值，打印数据 Direction = 0 为向上计数，Direction = 1 为向下计数。

This sample demonstrates encoder count function of the TIM1,TI1(PA8) and TI2(PA9) configured as encoder input pins.The change of the counter can be observed through the CNT register, and the counting direction of the counter can be observed through the uwDirection variable.The counting Direction and CNT register can also be observed by printing data. The printed data Direction = 0 indicates CounterMode:Up, and direction = 1 indicates CounterMode:down.

18.6 TIM1_ExternalClockMode1

此样例演示了 TIM1 的外部时钟模式 1 功能，选择 ETR(PA12)引脚作为外部时钟输入源，并使能更新中断，在中断中翻转 LED 灯

This sample demonstrates external clock mode 1 function of the TIM1.Select the ETR(PA12) pin as the external clock input source and enable the update interrupt to flip the LED light in the interrupt.

18.7 TIM1_ExternalClockMode1_TI1F

此样例演示了 TIM1 的外部时钟模式 1 功能，选择 TI1FD(PA8)引脚作为外部时钟输入源，并使能更新中断，在中断中翻转 LED 灯。

This sample demonstrates the external clock mode 1 function of TIM1, selects the TI1FD(PA8) pin as the external clock input source, and enables the update interrupt and toggle the LED light in the interrupt

18.8 TIM1_ExternalClockMode2

此样例演示了 TIM1 的外部时钟模式 2 功能，选择 ETR(PA12)引脚作为外部时钟输入源，并使能更新中断，在中断中翻转 LED 灯。

This sample demonstrates the external clock mode 2 function of TIM1, selects the ETR(PA12) pin as the external clock input source, and enables the update interrupt and toggle the LED light in the interrupt

18.9 TIM1_InputCapture_TI1FP1

此样例演示了在 TIM1(PA8)输入捕获功能，PA8 输入时钟信号，TIM1 捕获成功后，会进入捕获中断，每进一次中断，翻转一次 LED。

This sample demonstrates the input capture function of TIM1(PA8), PA8 input clock signal, when TIM1 capture success, will enter the capture interrupt,and toggle the LED in the interrupt

18.10 TIM1_InputCapture_XORCh1Ch2Ch3

此样例演示了在 TIM1 输入捕获功能，PA8 或 PA9 或 PA10 输入时钟信号，TIM1 捕获成功后，会进入捕获中断，每进一次中断，翻转一次 LED。

This sample demonstrates the input capture function of TIM1, input clock signal into PA8 or PA9 or PA10 will generate the capture interrupt after TIM1 capture successfully.Toggle the LED once per interruption

18.11 TIM1_OCToggle

此样例演示了 TIM1 比较模式下的 OC 翻转输出功能，使能 CH1(PA08),CH2(PA09),CH3(PA10),CH4(PA11)四个通道的输出功能，并且当计数器 TIMx_CNT 与 TIMx_CCRx 匹配时输出信号翻转，频率为 100KHz。

This sample demonstrates the OC toggle output function in TIM1 comparison mode, enabling CH1(PA08),CH2(PA09),CH3(PA10),CH4(PA11) four channel output function, then the output signal toggle when the counter TIMx_CNT matches TIMx_CCRx. The frequency is 100KHz

18.12 TIM1_OCToggle_IT

此样例演示了 TIM1 比较模式下的中断功能，在中断中翻转 GPIO。

This sample demonstrates the interrupt function in TIM1 comparison mode, toggle the GPIO in an interrupt.

18.13 TIM1_OnePulseOutput

此样例演示了 TIM1 的单脉冲模式，CH2(PA09)引脚上的上升沿，触发计数器开始计数，当计数值与 CCR1 匹配时，CH1(PA08)输出高电平，直到计数器溢出，CH1 再次输出低电平，计数器溢出后，定时器停止工作。

This sample demonstrates the one pulse mode of TIM1.The rising edge on the CH2(PA09) pin triggers the counter to start counting. when the count value matches CCR1,CH1(PA08) outputs a high level. When the counter overflows ,CH1 outputs the low level again. After the counter overflows, the timer stops working.

18.14 TIM1_PWM

本例程输出 4 路 PWM，通道 1 的占空比为 20%，通道 2 为 40%，通道 3 为 60%，通道 4 为 80%。

This sample outputs 4 channels PWM, the duty cycle of channel 1 is 20%,channel 2 is 40%, channel 3 is 60%,channel 4 is 80%

18.15 TIM1_SynchronizationEnable

定时器 1 的使能由定时器 3 控制，当定时器 3 计数时，LED 会常亮，当定时器 3 发生更新事件时，更新事件会触发定时器 1，定时器 1 开始计数后，LED 会以 5Hz 的频率进行翻转。

The enable of TIM1 is controlled by TIM3. When TIM3 counts, the LED will be steady on. The update event generated by TIM3 will triggers TIM1, and when TIM1 starts counting, the LED is toggled at a frequency of 5Hz

18.16 TIM1_TIM3_Cascade

此样例实现了 TIM1 和 TIM3 级联成 32 位计数器，TIM3 做主机，TIM3 的计数溢出信号作为 TIM1 的输入时钟，通过配置 TIM1 和 TIM3 的重载寄存器值，(在 TIM1 中断回调函数中）实现 LED 灯以 0.5Hz 频率闪烁。

This example realizes the cascade of TIM1 and TIM3 into a 32-bit counter, with TIM3 as the host.The count overflow signal of TIM3 acts as the input clock of TIM1.By configuring the reloaded register values of TIM1 and TIM3, the LED is toggled at 0.5Hz (in the TIM1 interrupt callback function).

19 USART

19.1 USART_HyperTerminal_AutoBaund_IT

此样例演示了 USART 的自动波特率检测功能。调试助手发送一个字符 0x7F，MCU 反馈字符串：Auto BaudRate Test。

This sample demonstrates the automatic baud rate detection feature of USART. When the debugging assistant sends a character 0x7F, the MCU will respond with the string: "Auto BaudRate Test".

19.2 USART_HyperTerminal_DMA

此样例演示了 USART 的 DMA 方式发送和接收数据，USART 配置为 115200，数据位 8，停止位 1，校验位 None，下载并运行程序后，通过上位机下发 12 个数据，例如 0x1~0xC，则 MCU 会把接收到的数据再次发送。

This sample demonstrates the USART data transmission and reception using the DMA method. The USART is configured with a baud rate of 115200, 8 data bits, 1 stop bit, and no parity. After compiling and downloading the program to the MCU, when the host computer sends 12 data bytes (0x1 to 0xC), the MCU will send the received data back to the host computer.

19.3 USART_HyperTerminal_IT

此样例演示了 USART 的中断方式发送和接收数据，USART 配置为 115200，数据位 8，停止位 1，校验位 None，下载并运行程序后，上位机通过 USART 会接收到 0x1-0xC，然后通过上位机下发 12 个数据，例如 0x1~0xC，则 MCU 会把接收到的数据再次发送到上位机。

This sample demonstrates the USART data transmission and reception using the interrupt method. The USART is configured with a baud rate of 115200, 8 data bits, 1 stop bit, and no parity. After compiling and downloading the program to the MCU, the host computer will receive data from 0x1 to 0xC through USART. Then, when the host computer sends 12 data bytes (0x1 to 0xC), the MCU will send back the received data to the host computer.

19.4 USART_HyperTerminal_Polling

此样例演示了 USART 的 POLLING 方式发送和接收数据，USART 配置为 115200，数据位 8，停止位 1，校验位 None，下载并运行程序后，通过 USART 会接收到 0x1-0xC，然后通过上位机下发 12 个数据，例如 0x1~0xC，MCU 会把接收到的数据再次发送。

This sample demonstrates the USART data transmission and reception using the POLLING method. The USART is configured with a baud rate of 115200, 8 data bits, 1 stop bit, and no parity. After compiling and downloading the program to the MCU, the host computer will receive data from 0x1 to 0xC through USART. Then, when the host computer sends 12 data bytes (0x1 to 0xC), the MCU will send back the received data to the host computer.

20 WWDG

20.1 WWDG_IT

此样例演示了 WWDG 的提前唤醒中断功能，看门狗计数器向下计数到 0x40 时产生中断，中断中喂狗，可以确保看门狗不会复位。

This example demonstrates early wake up interrupt function of the WWDG.When the watchdog counter counts down to 0x40 will generates an interrupt.Refresh the WWDG in interrupt to ensure that the WWDG does not reset.

20.2 WWDG_Window

此样例演示了 WWDG 的 窗口看门狗功能，配置 WWDG 的窗口上限（下限固定是 0x3F），程序中通过 delay 延时函数，确保程序是在 WWDG 计数窗口内进行喂狗动作，通过 LED 灯闪烁，可以判断窗口内喂狗并未产生复位。

This example demonstrates the window watchdog function of WWDG. Set the upper limit of the window of WWDG (the lower limit is fixed at 0x3F).The program ensures that the WWDG is refreshed in the WWDG counting window through the delay function, and can judge that the WWDG is refreshed in the window without resetting through the LED light blinking.