PY32F071C series
32-bit ARM® Cortex®-M0+ microcontroller

LL Library Sample Manual

1 ADC

1.1 ADC_AnalogWatchdog

此样例演示了 ADC 的模拟看门狗功能，当开启模拟看门狗通道的电压值超过上下限时，会进入看门狗中断。

This example demonstrates the analog watchdog function of ADC. When the voltage value of the analog watchdog channel exceeds the upper and lower limits, it will enter the watchdog interrupt.

1.2 ADC_MultiChannelSingleConversion_TriggerSW_DMA

此样例演示了 ADC 的 DMA 多通道传输功能，在 DMA 完成中断中打印多通道的电压值。

This example demonstrates the DMA multi-channel transmission function of ADC, which prints the voltage values of multiple channels during the DMA completion interrupt.

1.3 ADC_SingleConversion_TriggerSW_IT

此样例演示了 ADC 的中断功能，在 ADC 的中断中打印当前的电压值。

This example demonstrates the interrupt function of ADC, which prints the current voltage value in the interrupt of ADC.

1.4 ADC_SingleConversion_TriggerTimer_Polling

此样例演示了 ADC 的 TIM 触发功能,TIM 每隔 1s 触发 ADC 进行采样,并通过串口打印出来。

This example demonstrates the TIM triggering function of ADC. TIM triggers ADC for sampling every 1 second 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_Vrefbuf

此样例演示了 ADC 模块的 Vrefbuf 功能，利用 vrefbuf 作为基准去采样通道的值，并转换成电压通过串口打印出来。

This example demonstrates the Vrefbuf function of the ADC module, which uses Vrefbuf as the reference to sample channel values and convert them into voltage Print it out through the serial port.

1.7 ADC_Vrefint

此样例演示了 ADC 模块的 Vrefint 采样功能，通过采样 Vrefint 的值，计算得出 VCC 的值，并通过串口打印出来。

This example demonstrates the Vrefint sampling function of the ADC module. By sampling the value of Vrefint, the VCC value is calculated and printed through the serial port.

2 COMP

2.1 COMP_CompareGpioVsVrefint_IT

此样例演示了比较器的中断功能，在中断中翻转 LED。

This example demonstrates the interrupt function of the comparator, flipping the LED during the interrupt.

2.2 COMP_CompareGpioVsVrefint_Polling

此样例演示了比较器的轮询功能，当比较器的正端电压大于 Vrefint 时，LED 灯亮，小于 Vrefint 电压时，LED 灯灭。

This example demonstrates the polling function of the comparator. When the positive terminal voltage of the comparator is greater than Vrefit, the LED light will turn on, and when the voltage is less than Vrefit, the LED light will turn off.

2.3 COMP_CompareGpioVsVrefint_WakeUpFromStop

此样例演示了 COMP 比较器唤醒功能，PA0 作为比较器正端输入，VREFINT 作为比较器负端输入，上完电 LED 灯会常亮，用户点击按键，LED 灯灭，进入 stop 模式，通过调整 PA0 上的输入电压，产生中断唤醒 stop 模式。

This example demonstrates the COMP comparator wake-up function, with PA0 as the positive input and VREFINT as the negative input. After power on, the LED light will remain on. When the user clicks the button, the LED light will go out and enter stop mode. By adjusting the input voltage on PA0, an interrupt wake-up stop mode is generated.

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 sample demonstrates the CRC function, which performs a CRC calculation on the data in an array and compares the result with the theoretical value; if equal,the LED is on, otherwise the LED is off.

4 CTC

4.1 CTC_Autotrim_Init

此样例演示了 CTC 使用 LSE 做参考时钟自动校准 PLL48M 时钟的功能。

This example demonstrates the function of CTC using LSE as a reference clock to automatically calibrate the PLL48M clock.

5 DAC

5.1 DAC_GenerateConstantSignal_TriggerSW

此样例演示了 DAC 的输出功能，通过 PA4 输出 1/2 供电电压的值。

This example demonstrates the output function of DAC, which outputs a value of 1/2 supply voltage through PA4.

6 DIV

6.1 DIV_Signed

此样例演示了硬件除法器计算有符号除法。

This example demonstrates how a hardware divider calculates signed division.

6.2 DIV_Unsigned

此样例演示了硬件除法器计算无符号除法。

This example demonstrates how a hardware divider calculates unsigned division.

7 DMA

7.1 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).

8 EXTI

8.1 EXTI_Toggled_IT_Init

此样例演示了 GPIO 外部中断功能，PB0 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 PB0 pin generates an interrupt, and the LED light in the interrupt function flips 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 sample demonstrates the function to wake up the MCU via the PA6 pin. After downloading the program and running, the LED remains on; After pressing the user button, the LED remains off, and the MCU enters the STOP mode; After pulling down the PA6 pin, the MCU wakes up and the LED light is toggling.

9 Flash

9.1 FLASH_PageEraseAndWrite

此样例演示了 flash page 擦除和 page 写功能。

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

9.2 FLASH_SectorEraseAndWrite

此样例演示了 flash sector 擦除和 page 写功能。

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

10 GPIO

10.1 GPIO_FastIO

本样例主要展示 GPIO 的 FAST IO 输出功能，FAST IO 速度可以达到单周期翻转速度。

This sample demonstrates the FAST IO output function of GPIO, and the FAST IO speed can reach the single cycle toggled speed.

10.2 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 100ms 翻转一次 LED 引脚电平，运行程序，可以看到 LED 灯闪烁。

This sample demonstrates the GPIO output mode, configure the LED pin as digital output mode and toggle the LED pin level every 100ms, run the program,you can see the LED toggle.

10.3 GPIO_Toggle_Init

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 100ms 翻转一次 LED 引脚电平，运行程序，可以看到 LED 灯闪烁。

This sample demonstrates the GPIO output mode, configure the LED pin as digital output mode and toggle the LED pin level every 100ms, run the program,you can see the LED toggle.

11 I2C

11.1 I2C_TwoBoard_CommunicationMaster_10BitAddr_IT_Init

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，主机和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 I2C communication using interrupts. The master device first sends 15 bytes of data to the slave device, and then receives 15 bytes of data from the slave. When both the master and slave successfully receive the data, the LEDs on the master and slave boards are continuously on.

11.2 I2C_TwoBoard_CommunicationMaster_DMA_Init

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，主机和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 I2C communication using DMA. The master device first sends 15 bytes of data to the slave device, and then receives 15 bytes of data from the slave. When both the master and slave successfully receive the data, the LEDs on the master and slave boards are continuously on.

11.3 I2C_TwoBoard_CommunicationMaster_DMA_MEM_Init

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。按下用户按键后，主机先向从机写入 15 字节的数据 (0x1-0xf)，然后从 EEPROM 中读取写入的数据。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 and the slave device using I2C with DMA. The slave device uses the EEPROM peripheral chip P24C32. When the user button is pressed, the master device first writes 15 bytes of data (0x1-0xf) to the slave device, and then reads the written data from the EEPROM. Upon successful read, the LED on the master board remains continuously on.

11.4 I2C_TwoBoard_CommunicationMaster_DualAddr_IT_Init

此样例演示了通过 I2C 中断方式进行通讯。主机先向从机发送 15 字节的数据，然后再接收从机发送的 15 字节的数据。当主机和从机成功接收数据后，主机和从机板上的小灯将保持“常亮”状态。

This sample demonstrates communication between the master device and the slave device using I2C with interrupt. The master device first sends 15 bytes of data to the slave device and then receives 15 bytes of data from the slave device. Upon successful data transmission and reception, the LEDs on both the master and slave boards will remain continuously on.

11.5 I2C_TwoBoard_CommunicationMaster_IT_Init

此样例演示了通过 I2C 中断方式进行通讯。主机先向从机发送 15 字节的数据，然后再接收从机发送的 15 字节的数据。当主机和从机成功接收数据后，主机和从机板上的小灯将保持“常亮”状态。

This sample demonstrates communication between the master device and the slave device using I2C with interrupt. The master device first sends 15 bytes of data to the slave device and then receives 15 bytes of data from the slave device. Upon successful data transmission and reception, the LEDs on both the master and slave boards will remain continuously on.

11.6 I2C_TwoBoard_CommunicationMaster_Polling_Init

此样例演示了通过 I2C 轮询方式进行通讯。主机先向从机发送 15 字节的数据，然后再接收从机发送的 15 字节的数据。当主机和从机成功接收数据后，主机和从机板上的小灯将保持“常亮”状态。

This sample demonstrates communication between the master device and the slave device using I2C with polling. The master device first sends 15 bytes of data to the slave device and then receives 15 bytes of data from the slave device. Upon successful data transmission and reception, the LEDs on both the master and slave boards will remain continuously on.

11.7 I2C_TwoBoard_CommunicationSlave_10BitAddr_IT_Init

此样例演示了通过中断方式进行 I2C 通讯。主机先向从机发送 15 字节的数据，然后再接收从机发送的 15 字节的数据。当主机和从机成功接收数据后，主机和从机板上的小灯将保持“常亮”状态。

This sample demonstrates communication between the master device and the slave device using I2C with interrupt. The master device first sends 15 bytes of data to the slave device and then receives 15 bytes of data from the slave device. Upon successful data transmission and reception, the LEDs on both the master and slave boards will remain continuously on.

11.8 I2C_TwoBoard_CommunicationSlave_DMA_Init

此样例演示了通过 DMA 方式进行 I2C 通讯。主机先向从机发送 15 字节的数据，然后再接收从机发送的 15 字节的数据。当主机和从机成功接收数据后，主机和从机板上的小灯将保持“常亮”状态。

This sample demonstrates communication between the master device and the slave device using I2C with DMA. The master device first sends 15 bytes of data to the slave device and then receives 15 bytes of data from the slave device. Upon successful data transmission and reception, the LEDs on both the master and slave boards will remain continuously on.

11.9 I2C_TwoBoard_CommunicationSlave_DualAddr_IT_Init

此样例演示了通过中断方式进行 I2C 通讯。主机先向从机发送 15 字节的数据，然后再接收从机发送的 15 字节的数据。当主机和从机成功接收数据后，主机和从机板上的小灯将保持“常亮”状态。

This sample demonstrates communication between the master device and the slave device using I2C with interrupts. The master device first sends 15 bytes of data to the slave device and then receives 15 bytes of data from the slave device. Upon successful data transmission and reception, the LEDs on both the master and slave boards will remain continuously on.

11.10 I2C_TwoBoard_CommunicationSlave_IT_Init

此样例演示了通过中断方式进行 I2C 通讯。主机先向从机发送 15 字节的数据，然后再接收从机发送的 15 字节的数据。当主机和从机成功接收数据后，主机和从机板上的小灯将保持“常亮”状态。

This sample demonstrates communication between the master device and the slave device using I2C with interrupts. The master device first sends 15 bytes of data to the slave device and then receives 15 bytes of data from the slave device. Upon successful data transmission and reception, the LEDs on both the master and slave boards will remain continuously on.

11.11 I2C_TwoBoard_MasterTxIndefiniteLengthData_IT_Init

此样例演示了通过中断方式，主机发送不定长数据，从机接收不定长数据。主机向从机发送 10 字节的数据（0~9），然后从机接收数据（0~9）并通过串口打印；主机向从机发送 100 字节数据（1~100），然后从机接收数据（1~100）并通过串口打印；主机向从机发送 10 字节的数据（0~9），然后从机接收数据（0~9）并通过串口打印。

This example demonstrates how the host sends variable length data and the slave receives variable length data through interrupt mode. The host sends 10 bytes of data (0-9) to the slave, and then the slave receives the data (0-9) and prints it through the serial port; The host sends 100 bytes of data (1-100) to the slave, and then the slave receives the data (1-100) and prints it through the serial port; The host sends 10 bytes of data (0-9) to the slave, and then the slave receives the data (0-9) and prints it through the serial port.

11.12 I2C_TwoBoard_SlaveRxIndefiniteLengthData_IT_Init

此样例演示了通过中断方式，主机发送不定长数据，从机接收不定长数据。主机向从机发送 10 字节的数据（0~9），然后从机接收数据（0~9）并通过串口打印；主机向从机发送 100 字节数据（1~100），然后从机接收数据（1~100）并通过串口打印；主机向从机发送 10 字节的数据（0~9），然后从机接收数据（0~9）并通过串口打印。

This example demonstrates how the host sends variable length data and the slave receives variable length data through interrupt mode. The host sends 10 bytes of data (0-9) to the slave, and then the slave receives the data (0-9) and prints it through the serial port; The host sends 100 bytes of data (1-100) to the slave, and then the slave receives the data (1-100) and prints it through the serial port; The host sends 10 bytes of data (0-9) to the slave, and then the slave receives the data (0-9) and prints it through the serial port.

12 I2S

12.1 I2S_TwoBoard_CommunicationMaster_DMA

此样例演示了通过 DMA 方式进行 I2S 主机与 I2S 从机的通信。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 0x1~0x10。当 I2S 主机和 I2S 从机成功接收数据时，LED 灯将保持常亮状态；否则 LED 灯将处于闪烁状态。

This sample demonstrates communication between an I2S master and an I2S slave using DMA. The I2S master device first sends data from 0x1 to 0x10 to the I2S slave device. The I2S slave device receives the data and then sends data from 0x1 to 0x10 back to the I2S master device. When both the I2S master and I2S slave successfully receive the data, the LED lights remain continuously on. Otherwise, the LED lights will blink.

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 an I2S master and an I2S slave using interrupt mode. The I2S master sends data 0x1 to 0x10 to the I2S slave. After receiving the data, the I2S slave sends data 0x1 to 0x10 back to the I2S master. When the I2S master and slave successfully receive the data, the LED remains lit. Otherwise, the LED blinks.

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 an I2S master and an I2S slave using polling mode. The I2S master sends data 0x1 to 0x10 to the I2S slave. After receiving the data, the I2S slave sends data 0x1 to 0x10 back to the I2S master. When the I2S master and slave successfully receive the data, the LED remains lit. Otherwise, the LED blinks.

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 an I2S master and an I2S slave using DMA. The I2S master sends data 0x1 to 0x10 to the I2S slave. After receiving the data, the I2S slave sends data 0x1 to 0x10 back to the I2S master. When the I2S master and slave successfully receive the data, the LED remains lit. Otherwise, the LED blinks.

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 an I2S master and an I2S slave using interrupt. The I2S master sends data 0x1 to 0x10 to the I2S slave. After receiving the data, the I2S slave sends data 0x1 to 0x10 back to the I2S master. When the I2S master and slave successfully receive the data, the LED remains lit. Otherwise, the LED blinks.

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 an I2S master and an I2S slave using polling. The I2S master sends data 0x1 to 0x10 to the I2S slave. After receiving the data, the I2S slave sends data 0x1 to 0x10 back to the I2S master. When the I2S master and slave successfully receive the data, the LED remains lit. Otherwise, the LED blinks.

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 1s 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，如果每次喂狗时间小于 1s，程序能一直正常运行（LED 灯闪烁），如果喂狗时间超过 1s，程序会一直复位（LED 灯不亮）。

This sample demonstrates the IWDG (Independent Watchdog) functionality. It configures the watchdog reload value to count for 1 second before resetting. By adjusting the time interval for feeding the watchdog (code in the main function's while loop), you can observe the behavior: if the feeding time is less than 1 second, the program runs normally (LED blinks), but if the feeding time exceeds 1 second, the program keeps resetting (LED remains off).

14 LCD

14.1 LCD_Display_Init

此样例是对单色无源液晶显示器(LCD)的演示，将偏置产生电路配置为内部电阻分压，使 LCD 全显，显示“88:88”字样。

This sample demonstrates the usage of a monochrome passive LCD (Liquid Crystal Display). The bias generation circuit is configured with internal resistor voltage division to achieve a fully displayed LCD showing the "88:88" characters.

15 LPTIM

15.1 LPTIM_ContinuousMode_WakeUp_WFE

此样例演示了 LPTIM 连续模式事件唤醒 STOP 模式。

This example demonstrates the LPTIM continuous mode event wake-up STOP mode.

15.2 LPTIM_ContinuousMode_WakeUp_WFI

此样例演示了 LPTIM 连续模式中断唤醒 STOP 模式。

This sample demonstrates waking up from stop mode by LPTIM(contiunus mode) interrupt request.

15.3 LPTIM_OnceMode_WakeUp_WFE

此样例演示了 LPTIM 单次模式事件唤醒 STOP 模式。

This example demonstrates the LPTIM once mode event wake-up STOP mode.

15.4 LPTIM_OnceMode_WakeUp_WFI

此样例演示了 LPTIM 单次模式中断唤醒 STOP 模式。

This sample demonstrates waking up from stop mode by LPTIM(once mode) interrupt request.

16 OPA

16.1 OPA_VoltageFollow

此样例演示了 OPA 的电压跟随功能。PA7 为正端输入，PA5 为负端输入，PA6 为输出，PA6 会输出和 PA7 相同的电压值。

This sample demonstrates the voltage follower functionality of the OPA (Operational Amplifier). PA7 is the positive input, PA5 is the negative input, and PA6 is the output. PA6 will output the same voltage as PA7.

17 PWR

17.1 PWR_PVD

此样例演示了 PVD (电压检测功能)。样例中配置 PB07 引脚的电压与 VREF (1.2V) 进行比较。当 PB07 引脚的电压高于 VREF 时，LED 灯灭；当低于 VREF 时，LED 灯亮。

This sample demonstrates the PVD (Voltage Detection) function. The voltage at PB07 pin is compared with VREF (1.2V). When the voltage at PB07 is higher than VREF, the LED light is off. When the voltage is lower than VREF, the LED light is on.

17.2 PWR_SLEEP_WFE

此样例演示了在 sleep 模式下，使用 GPIO 事件唤醒。

This sample demonstrates the usage of GPIO event to wake up the MCU from sleep mode.

17.3 PWR_SLEEP_WFI

此样例演示了在 sleep 模式下，使用 GPIO 中断唤醒。

This sample demonstrates the usage of GPIO interrupt to wake up the MCU from sleep mode.

17.4 PWR_STOP_WFE

此样例演示了在 stop 模式下，使用 GPIO 事件唤醒。

This sample demonstrates the usage of GPIO event to wake up the MCU from stop mode.

17.5 PWR_STOP_WFI

此样例演示了在 stop 模式下，使用 GPIO 中断唤醒。

This sample demonstrates the usage of GPIO interrupt to wake up the MCU from stop mode.

18 RCC

18.1 RCC_HSE_OUTPUT

此样例演示了时钟输出功能，可输出 HSE 波形。

This sample demonstrates the clock output feature that can output the HSE waveform.

18.2 RCC_HSI_OUTPUT

此样例演示了时钟输出功能，可输出 HSI 波形。

This sample demonstrates the clock output feature that can output the HSI waveform.

18.3 RCC_LSE_OUTPUT

此样例演示了将系统时钟设置为 LSE，并通过 MCO 引脚输出系统时钟。

This example demonstrates setting the system clock to LSE and outputting the system clock through the MCO pin.

18.4 RCC_LSI_OUTPUT

此样例演示了将系统时钟设置为 LSI，并通过 MCO 引脚输出系统时钟。

This example demonstrates setting the system clock to LSI and outputting the system clock through the MCO pin.

18.5 RCC_PLL_OUTPUT

此样例演示了时钟输出功能，可输出 PLL 波形（32MHz）。

This sample demonstrates the clock output function, which can output the PLL waveform (32MHz).

18.6 RCC_Sysclock_Switch

此样例演示了时钟切换，由 LSI（32.768KHz）切换至 HSE（24MHz）。

This sample demonstrates clock switching from LSI (32.768KHz) to HSE (24MHz).

19 RTC

19.1 RTC_Alarm_Init

此样例演示 RTC 的闹钟中断功能，在数组 aShowTime 中显示当前时间，在数组 aShowDate 中显示当前日期，当达到闹钟值时，LED 灯会亮起。

This sample demonstrates the alarm interrupt function of the RTC. It displays the current time in the aShowTime array and the current date in the aShowDate array. When the alarm value is reached, the LED will light up.

19.2 RTC_WakeUpAlarm_Init

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

This sample demonstrates waking up the MCU from STOP mode approximately every 1 second using RTC alarm interrupt. Each time the MCU wakes up, the LED will toggle, with an interval of approximately 1 second between each toggle.

19.3 RTC_WakeUpSecond_Init

此样例演示通过 RTC 秒中断从 STOP 模式下唤醒，唤醒后，小灯处于闪烁状态；否则处于熄灭状态。

This sample demonstrates waking up the MCU from STOP mode using RTC second interrupt. After waking up, the LED will either blink or remain off.

20 SPI

20.1 SPI_TwoBoards_FullDuplexMaster_DMA_Init

此样例是利用 DMA 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 uses DMA for full-duplex communication with external devices.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.

20.2 SPI_TwoBoards_FullDuplexMaster_IT_Init

此样例是利用中断对串口外设接口 (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 uses interrupt for full-duplex communication with external devices.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.

20.3 SPI_TwoBoards_FullDuplexMaster_Polling_Init

此样例是通过轮询方式对串口外设接口（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 uses polling mode for full-duplex communication with external devices.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.

20.4 SPI_TwoBoards_FullDuplexSlave_DMA_Init

此样例是利用 DMA 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 uses DMA for full-duplex communication with external devices.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.

20.5 SPI_TwoBoards_FullDuplexSlave_IT_Init

此样例是利用中断对串口外设接口 (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 uses interrupt for full-duplex communication with external devices.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.

20.6 SPI_TwoBoards_FullDuplexSlave_Polling_Init

此样例是通过轮询方式对串口外设接口（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 uses polling mode for full-duplex communication with external devices.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.

21 TIM

21.1 TIM1_6Step_Init

此样例演示了使用 TIM1 产生“六步 PWM 信号”，每间隔 1ms 在 SysTick 中断中触发换向，实现无刷电机的换向。

This sample demonstrates how TIM1 can be used to generate a "six-step PWM signal." The commutation is triggered in the SysTick interrupt every 1ms to realize the commutation of the brushless motor.

21.2 TIM1_ComplementarySignals_Init

此样例演示了使用 TIM1 输出三路频率为 10Hz 占空比分别为 25%、50%、75%的 PWM 波形以及他们的互补信号，并实现了 250ns 的死区和刹车功能。

This sample demonstrates the generation of three PWM waveforms with frequencies of 10Hz and duty cycles of 25%, 50%, and 75% using TIM1. It also generates their complementary signals. It also generates their complementary signals and implements a 250ns dead-time and brake function.

21.3 TIM1_DmaBurst_Init

此样例演示了 TIM1 的 DMA Burst 传输，配置 TIM1 为 PWM 模式，更新中断触发 DMA 传输请求。每次产生更新中断时将 TIM1DataBuff[]中的值按顺序写入 RCR 和 CCR1 寄存器，改变 PWM 脉冲的占空比和该占空比的脉冲数量。

This sample demonstrates the DMA Burst transfer of TIM1. It configures TIM1 in PWM mode and triggers DMA transfer requests on update interrupt. Each time an update interrupt occurs, the values in TIM1DataBuff[] are sequentially written to RCR and CCR1 registers, changing the duty cycle and the number of pulses for the PWM waveform.

21.4 TIM1_EncoderTI2AndTI1_Init

此样例演示了 TIM1 的编码器接口模式。TIM1 配置为编码器接口模式 3，PA8 和 PA9 配置为通道 1 和通道 2,当 PA8 输入信号的上升沿在前，PA9 输入信号上升沿在后时 TIM1 向上计数，反之向下计数。开启通道 1 和通道 2 的捕获中断，在中断中打印当前 CNT 值。

This sample demonstrates the encoder interface mode of TIM1. TIM1 is configured in encoder interface mode 3, with PA8 and PA9 configured as channel 1 and channel 2, respectively. When the rising edge of the input signal on PA8 occurs before the rising edge of the input signal on PA9, TIM1 counts up; otherwise, it counts down. The capture interrupts for channel 1 and channel 2 are enabled, and the current CNT value is printed in the interrupt.

21.5 TIM1_InputCapture

此样例演示了 TIM1 的输入捕获功能。配置 PA8 为通道 1 的输入引脚,每当引脚电平出现上升沿时会触发捕获中断,并在中断处理中翻转 LED。

This sample demonstrates the input capture functionality of TIM1 . Configure PA8 as the input capture pin. Whenever an rising edge is detected on PA8, it triggers the capture interrupt and toggles the LED in the interrupt callback function.

21.6 TIM1_InputCapture_XORCh1Ch2Ch3

此样例演示了 TIM1 的三通道异或输入捕获功能。配置 PA8、PA9、PA10 为通道 1、通道 2、通道 3 的输入引脚。每当有一个引脚电平变化时会触发捕获中断，并在中断处理中翻转 LED。

This sample demonstrates the XOR input capture functionality of TIM1 using three channels: PA8, PA9, and PA10 as the input pins for channel 1, channel 2, and channel 3, respectively. Whenever there is a change in the level of any of the input pins, it triggers the capture interrupt and toggles the LED in the interrupt handler.

21.7 TIM1_OC_Toggle

此样例演示了 TIM1 的输出比较模式。将捕获/比较通道 1(CH1)的输出映射到 PA8，开启捕获/比较通道

1(CH1)并设置为比较输出翻转模式

This sample demonstrates the output compare mode of TIM1. The output of capture/compare channel 1 (CH1) is mapped to pin PA8. Capture/compare channel 1 (CH1) is enabled and set to compare output toggle mode.

21.8 TIM1_OnePulseOutput

此样例演示了 TIM1 的单脉冲模式。配置 TIM1 为从模式触发模式，触发源为 TI2FP2，通道 1 为 PWM2 模式，映射到 PA8，通道 2 为输入模式，映射到 PA9。当 PA9 上检测到一个上升沿时，PA8 延迟 20ms 后产生一个宽度为 80ms 的脉冲。

This sample demonstrates the single pulse mode of TIM1. TIM1 is configured in slave mode trigger mode with TI2FP2 as the trigger source. Channel 1 is configured as PWM mode 2 and mapped to pin PA8, while channel 2 is configured as input mode and mapped to pin PA9. When an rising edge is detected on PA9, a 20ms delay is applied, and then PA8 will output a pulse with a width of 80ms.

21.9 TIM1_PWM_Init

此样例演示了使用 TIM1 PWM2 模式输出三路频率为 10Hz 占空比分别为 25%、50%、75%的 PWM 波形。

This example demonstrates the use of TIM1 PWM2 mode to output three PWM waves with a frequency of 10Hz and a duty cycle of 25%, 50% and 75%, respectively.

21.10 TIM1_TIM3_Cascade

此样例演示了 TIM1 和 TIM3 级联成 32 位计数器，TIM3 做主机，TIM3 的溢出信号作为 TIM1 的输入时钟。TIM3 每 1ms 计数一次，计数 1000 次后产生溢出，TIM1 计数一次。

This sample demonstrates the cascading of TIM1 and TIM3 as a 32-bit counter, with TIM3 as the master and the overflow signal of TIM3 as the input clock of TIM1. TIM3 counts every 1ms, and after counting 1000 times, it overflows and TIM1 counts once.

21.11 TIM1_TimeBase_Init

此样例演示了 TIM1 的更新中断功能，在更新中断中翻转 LED。

This sample demonstrates the update interrupt function of TIM1, and toggle the LED in update interrupt

21.12 TIM1_Update_DMA_Init

此样例演示了在 TIM1 中使用 DMA 传输数据的功能,通过 DMA 从 SRAM 中搬运数据到 ARR 寄存器实现 TIM1 更新周期变化,TIM1 第一次溢出后 LED 会翻转,此次翻转时间间隔为 1000ms,DMA 将数据搬运到 TIM1_ARR,第二次 LED 翻转间隔为 900ms,以此类推,最后 LED 翻转间隔为 100msDMA 搬运结束,LED 保持 100ms 的翻转间隔闪烁。

This sample demonstrates the use of DMA to transfer data in TIM1, copying data from SRAM to the ARR register to achieve varying update periods for TIM1. After the first overflow of TIM1, the LED will toggle, with a time interval of 1000ms. After the data is transferred to TIM1_ARR using DMA, the LED toggling interval gradually decreases: 900ms, 800ms, 700ms, 600ms, 500ms, 400ms, 300ms, 200ms, 100ms. Finally, the LED will blink with a constant toggling interval of 100ms.

22 USART

22.1 USART_HyperTerminal_AutoBaund_IT_Init

此样例演示了 USART 的自动波特率检测功能,上位机发送 1 字节的波特率检测字符 0x55，如果 MCU 检测成功，则返回字符：Auto BaudRate Test。

This example demonstrates the automatic baud rate detection function of USART. If the MCU detects successfully after the upper computer sends 1 byte baud rate detection character 0x55, it will returns the string: Auto BaudRate Test.

22.2 USART_HyperTerminal_DMA_Init

此样例演示了 USART 的 DMA 方式发送和接收数据，USART 配置为 9600，数据位 8，停止位 1，校验位 None,下载并运行程序后，打印提示信息，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，然后打印结束信息。

This example demonstrates how to use USART to send an amount of data in DMA mode. USART configuration is 9600 baud rate, data bit 8, stop bit 1, check bit None. After download and run the program,Print the prompt message, and then send 12 data through the upper computer, such as 0x1~0xC, the MCU will send the received data to the upper computer again, Then print the end message.

22.3 USART_HyperTerminal_IndefiniteLengthData_IT

此样例演示了 USART 的中断方式发送和接收不定长数据，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，然后通过上位机下发任意长度个数据（不超过 128bytes），例如 0x1~0xC,则 MCU 会把接收到的数据再次发送到上位机。

This example demonstrates the interrupt method of USART to send and receive variable length data. USART is configured as 115200, with data bit 8, stop bit 1, and check bit None. After downloading and running the program, the MCU will send any length of data (not exceeding 128bytes) through the upper computer, such as 0x1~0xC. The MCU will send the received data to the upper computer again.

22.4 USART_HyperTerminal_IT_Init

此样例演示了 USART 的中断方式发送和接收数据，USART 配置为 9600，数据位 8，停止位 1，校验位 None,下载并运行程序后，打印提示信息，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，然后打印结束信息。

This example demonstrates how to use USART to send an amount of data in interrupt mode. USART configuration is 9600 baud rate, data bit 8, stop bit 1, check bit None. After download and run the program,Print the prompt message, and then send 12 data through the upper computer, such as 0x1~0xC, the MCU will send the received data to the upper computer again, Then print the end message.

22.5 USART_HyperTerminal_Polling_Init

此样例演示了 USART 的轮询方式发送和接收数据，USART 配置为 9600，数据位 8，停止位 1，校验位 None,下载并运行程序后，打印提示信息，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，然后打印结束信息。

This example demonstrates how to use USART to send an amount of data in polling mode. USART configuration is 9600 baud rate, data bit 8, stop bit 1, check bit None. After download and run the program,Print the prompt message, and then send 12 data through the upper computer, such as 0x1~0xC, the MCU will send the received data to the upper computer again, Then print the end message.

23 UTILS

23.1 UTILS_ConfigureSystemClock

本样例主要演示如何配置 SYSCLK(系统时钟), HCLK(AHB 时钟), PCLK(APB 时钟)。通过 MCO 输出系统时钟的 8 分频 9MHz。

This example shows how to configure SYSCLK(system clock), HCLK(AHB clock), and PCLK(APB clock).Output the 8-frequency division(9MHz) of the system clock via MCO pin

24 WWDG

24.1 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 sample demonstrates the early wake-up interrupt function of the WWDG. When the watchdog counter counts down to 0x40, an interrupt is generated. In the interrupt handler, refresh WWDG counter to ensure that the watchdog does not reset the system.

24.2 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.