PY32F003 series
32-bit ARM® Cortex®-M0+ 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，当开启看门狗通道的电压值不在设定的上下限中，会进入看门狗中断。

This sample demonstrates the analog watchdog function of the ADC. When the voltage value of the channel with the watchdog on is not in the set upper and lower limits, the will enter the watchdog interrupt.

1.2 ADC_ContinuousConversion_DMA

此样例演示了通过 ADC 触发 DMA 搬运 ADC_DR 中的数据，配置 CH0(PA00)为 ADC 的模拟输入通道，ADC 触发 DMA 搬运采样数据，DMA 搬运完成，在循环中打印搬运的数据。

This sample demonstrates the transfer of ADC data from ADC_DR using DMA triggered by ADC. It configures CH0(PA00) as the analog input channel for ADC. The ADC triggers DMA to transfer the sampled data. After DMA transfer completion, the transferred data is printed in a loop.

1.3 ADC_MultiChannelSingleConversion_TrigggerSW

此样例演示了 ADC 单次轮询(Polling)的方式采样 AN0,AN1,AN4 通道,并通过串口打印出来。

This sample demonstrates the ADC single-channel polling mode to sample channels AN0, AN1, and AN4. The sampled values are then printed via the serial port.

1.4 ADC_MultiChannelSwitch

此样例演示了 ADC 的多通道切换。

This sample demonstrates the multichannel switching of ADC.

1.5 ADC_SingleConversion_TriggerTimer_DMA

此样例演示了通过 TIM1 触发 ADC,再通过 ADC 触发 DMA 搬运 ADC_DR 中的数据，配置 CH0(PA00)为 ADC 的模拟输入通道，TIM1 配置为主模式，TIM1 每产生一次更新事件，触发一次 ADC 采样，然后 ADC 触发 DMA 搬运采样数据，采样数据在中断中打印。

This sample demonstrates the use of TIM1 to trigger ADC conversion, followed by DMA transfer of data from ADC_DR. CH0(PA00) is configured as the analog input channel for ADC. TIM1 is set to the master mode, triggering ADC conversion on each update event. The sampled data is then transferred by DMA, and the data is printed in an interrupt handler.

1.6 ADC_SingleConversion_TriggerTimer_IT

此样例演示了通过 TIM1 触发 ADC 模块的通道采样功能，配置 CH0(PA00)为 ADC 的模拟输入通道，TIM1 配置为主模式，TIM1 每产生一次更新事件，触发一次 ADC 采样，采样数据在中断中打印。

This sample demonstrates the use of TIM1 to trigger channel sampling in the ADC module. CH0(PA00) is configured as the analog input channel for ADC. TIM1 is set to the master mode, triggering ADC sampling on each update event. The sampled data is printed in an interrupt handler.

1.7 ADC_TempSensor

此样例演示了 ADC 模块的温度传感器功能，串口每隔 500ms 打印当前的温度。

This sample demonstrates the temperature sensor functionality of the ADC module. The current temperature is printed via the serial port every 500 milliseconds.

1.8 ADC_Vrefint

此样例演示了 ADC 模块的 VCC 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 sample demonstrates the VCC sampling functionality of the ADC module. By sampling the value of VREFINT and performing calculations, the VCC voltage value is obtained and printed via the serial port.

2 COMP

2.1 COMP_CompareGpioVsVrefint_HYST

此样例演示了 COMP 比较器迟滞功能，PA03 作为比较器正端输入，VREFINT 作为比较器负端输入，PA07 作为比较器的输出端口，通过调整 PA03 上的输入电压，观测 PA07 引脚上的电平变化。

This sample demonstrates the hysteresis function of the COMP comparator. PA03 serves as the positive input of the comparator, VREFINT serves as the negative input, and PA07 is the output port of the comparator. By adjusting the input voltage on PA03, the level change on pin PA07 can be observed.

2.2 COMP_CompareGpioVsVrefint_IT

此样例演示了 COMP 比较器功能，PA01 作为比较器正端输入，VREFINT 作为比较器负端输入，PA06 作为比较器的输出端口，通过调整 PA01 上的输入电压，观测 PA06 引脚上的电平变化和 LED 翻转。

This sample demonstrates the interrupt function of the comparator. PA01 serves as the positive input of the comparator, VREFINT serves as the negative input, and PA06 is the output port of the comparator. By adjusting the input voltage on PA01, the level change on pin PA06 and the voltage toggle on LED pin can be observed.

2.3 COMP_CompareGpioVsVrefint_Polling

此样例演示了 COMP 比较器功能，PA01 作为比较器正端输入，VREFINT 作为比较器负端输入，PA06 作为比较器的输出端口，通过调整 PA01 上的输入电压，观测 PA06 引脚上的电平变化。

This sample demonstrates the polling function of the COMP comparator. PA01 serves as the positive input of the comparator, VREFINT serves as the negative input, and PA06 is the output port of the comparator. By adjusting the input voltage on PA01, the level change on pin PA06 can be observed.

2.4 COMP_CompareGpioVsVrefint_WakeUpFromStop

此样例演示了 COMP 比较器唤醒功能，PA01 作为比较器正端输入，VREFINT 作为比较器负端输入，PA06 作为比较器的输出端口，进入 stop 模式后，通过调整 PA01 上的输入电压，产生中断唤醒 stop 模式。

This sample demonstrates the wake-up function of the COMP comparator. PA01 serves as the positive input of the comparator, VREFINT serves as the negative input, and PA06 is the output port of the comparator. After entering the stop mode, by adjusting the input voltage on PA01, an interrupt is generated to wake up the stop mode.

2.5 COMP_CompareGpioVsVrefint_Window

此样例演示了比较器窗口功能，PA01 作为比较器正端输入，内部通过 COMP1 和 COMP2 的正极相连，PA1 输入大于 VREF 电压，LED 以 200ms 的间隔进行翻转；PA1 输入小于 1/4VREF 的电压，LED 熄灭；PA1 输入的电压小于 VREF 大于 1/4VREF，LED 常亮。

This sample demonstrates the window function of the comparator. PA01 serves as the positive input of the comparator, and it is internally connected to the positive terminals of COMP1 and COMP2. When PA1 inputs a voltage greater than VREF, the LED toggles every 200ms. When PA1 inputs a voltage less than 1/4VREF, the LED turns off. When PA1 inputs a voltage less than VREF but greater than 1/4VREF, the LED remains on.

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 verification function. It performs a CRC check on an array of data and compares the obtained checksum with the expected checksum. If they are equal, the LED lights up; otherwise, the LED turns off.

4 DMA

4.1 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 sample demonstrates the functionality of transferring data from SRAM to SRAM using DMA (for examples of data transfer between SRAM and peripherals, please refer to the relevant peripheral sample projects).

5 EXTI

5.1 EXTI_ToggleLed_IT

此样例演示了 GPIO 外部中断功能，PA12 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 sample demonstrates the GPIO external interrupt functionality. Every falling edge on pin PA12 generates an interrupt, and the LED toggles once in the interrupt function.

5.2 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 functionality of waking up the MCU using pin PA6. After downloading and running the program, the LED remains constantly lit. Pressing the user button puts the LED in a constant off state, and the MCU enters STOP mode. Pulling the PA6 pin low wakes up the MCU, and the LED blinks.

6 FLASH

6.1 FLASH_OptionByteWrite_RST

此样例演示了通过软件方式将 RESET 引脚改为普通 GPIO。

This sample demonstrates how to change the RESET pin to a regular GPIO using software.

6.2 FLASH_PageEraseAndWrite

此样例演示了 flash page 擦除和 page 写功能。

This sample demonstrates flash page erasure and page writing functionality.

6.3 FLASH_SectorEraseAndWrite

此样例演示了 flash sector 擦除和 page 写功能。

This sample demonstrates flash sector erasure and page writing functionality.

7 GPIO

7.1 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 functionality of GPIO. FAST IO speed can achieve single-cycle toggling speed.

7.2 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 250ms 翻转一次 LED 引脚电平，运行程序，可以看到 LED 灯以 2Hz 的频率闪烁。

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

8 I2C

8.1 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.

8.2 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.

8.3 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.

8.4 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.

8.5 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.

8.6 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.

9 IWDG

9.1 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 灯熄灭）。

This sample demonstrates the use of the IWDG (Independent Watchdog) functionality. The watchdog is configured with a reload value, and the system will be reset when the watchdog counter reaches 1 second. By adjusting the time interval between each watchdog feed (located in the main function's while loop), the following observations can be made: if the watchdog is fed within an interval shorter than 1 second, the program will continue running normally (LED blinking); if the feeding interval exceeds 1 second, the program will be reset by the watchdog timer (LED turned off).

10 LPTIM

10.1 LPTIM_Wakeup

此样例演示了 LPTIM 中断唤醒 stop 模式，每次唤醒后再次进入 stop 模式，每 200ms 唤醒一次。

This sample demonstrates the LPTIM interrupt wake-up in stop mode. After each wake-up, it re-enters stop mode and wakes up again every 200ms.

11 PWR

11.1 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 feature. In the sample, the voltage on pin PB07 is compared with VREF (1.2V). When the voltage on PB07 is higher than VREF, the LED will be turned off. When the voltage is lower than VREF, the LED will be turned on.

11.2 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 event wake-up feature in sleep mode.

11.3 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.

11.4 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.

11.5 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.

12 RCC

12.1 RCC_HSEOutput

此样例配置系统时钟为 HSE，并通过 MCO（PA01）引脚输出

This sample configures the system clock as HSE and outputs it through the MCO (PA01) pin.

12.2 RCC_HSICalibration

此样例使用外接 1KHz 高精度的时钟来校准内部 HSI。

This sample uses an external 1KHz high-precision clock to calibrate the internal HSI.

12.3 RCC_HSIOutput

此样例配置系统时钟为 HSI，并通过 MCO（PA01）引脚输出

This sample configures the system clock as HSI and outputs it through the MCO (PA01) pin.

12.4 RCC_LSIOutput

此样例使能 LSI，并通过 MCO（PA01）引脚输出。

This sample enables the LSI and is output via the MCO (PA08) pin.

12.5 RCC_SysclockSwitch

此样例演示系统时钟切换功能，样例中配置系统时钟从 HSI 切换到 HSE，并通过 MCO（PA01）引脚输出系统时钟。

This sample demonstrates the system clock switching functionality. The sample configures the system clock to switch from HSI to HSE and outputs the system clock through the MCO (PA01) pin.

13 RTC

13.1 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；当到达闹钟时间后会打印“RTC_IT_ALARM”。

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;When the alarm time is reached, print "RTC-IT-ALARM".

13.2 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 waking up the MCU from the STOP mode every 1 second using the RTC alarm interrupt. Each time the MCU wakes up, the LED will toggle, with a toggle interval of 1 second.

14 SPI

14.1 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 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.

14.2 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 sample demonstrates communication with an external device using the Serial Peripheral Interface (SPI) in full-duplex serial mode. The SPI interface is configured as the slave, receiving the communication clock signal (SCK) from the external master device. The slave receives data from the master via the MOSI pin and sends data to the master via the MISO pin. The data is shifted synchronously with the clock signal provided by the master, enabling full-duplex communication.

15 TIMER1

15.1 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 灯修改配置 TimHandle.Init.AutoReloadPreload = TIM_AUTORELOAD_PRELOAD_ENABLE;使能自动重载功能，新的 ARR 值在第四次进中断时生效，配置 TimHandle.Init.AutoReloadPreload = TIM_AUTORELOAD_PRELOAD_DISABLE;禁止自动重载功能，新的 ARR 值在第三次进中断时生效，生效后，LED 灯每隔 400ms 翻转一次

This sample implements the basic counting function of the timer, as well as demonstrating the ARR auto-reload function, the sample interrupts the timer reload in the The sample flips the LED in the timer reload interrupt. Modify the configuration TimHandle.Init.AutoReloadPreload = TIM_AUTORELOAD_PRELOAD_ENABLE; to enable the auto reload function, the new ARR value will be set on the fourth interrupt entry. Reload function, the new ARR value takes effect on the fourth incoming interrupt, configure TimHandle.Init.AutoReloadPreload = TIM_AUTORELOAD_PRELOAD_DISABLE; disable the auto-reeload function, the new ARR value will take effect at the third interrupt entry. After the new ARR value takes effect at the third interrupt entry, the LEDs will flip every 400ms.

15.2 TIM1_ComplementarySignals

此样例实现了定时器的互补输出功能，一组互补共两路 pwm 输出，此样例没有实现死区功能。

This sample implements the complementary output function of the timer, a set of complementary total two pwm outputs, this sample does not implement the deadband function.

15.3 TIM1_ComplementarySignals_break

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。通过调整 OCxE,CCxP,OISx,CCxNE,CCxNP,OISxN 的配置，可实现刹车功能的各种应用。

This sample demonstrates the brake function of the timer. It generates complementary PWM outputs using CH1 and CH1N. When receiving the brake signal (low level) from the external IO port (PA6), the PWM signal is turned off. Due to BDTR.AOE being set, when the brake signal is canceled (high level), PWM output resumes. This sample implements the dead time feature. By adjusting the configuration of OCxE, CCxP, OISx, CCxNE, CCxNP, OISxN, various brake function applications can be achieved.

15.4 TIM1_ComplementarySignals_break_it

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。本样例开启了刹车中断，并在刹车中断里翻转 LED 灯。通过调整 OCxE,CCxP,OISx,CCxNE,CCxNP,OISxN 的配置，可实现刹车功能的各种应用

This sample demonstrates the brake function of the timer. It generates complementary PWM outputs using CH1 and CH1N. When receiving the brake signal (low level) from the external IO port, the PWM signal is turned off. Due to BDTR.AOE being set, when the brake signal is canceled (high level), PWM output resumes. This sample implements the dead time feature. The sample also enables the brake interrupt and toggles an LED in the brake interrupt handler. By adjusting the configuration of OCxE, CCxP, OISx, CCxNE, CCxNP, OISxN, various brake function applications can be achieved.

15.5 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，PA0 会进行翻转，通过逻辑分析仪监测，可看到 PA0 的翻转间隔会从第一次的 400ms，第二次 400ms，第三次 10ms,第四次及后续变为 100us，此时两次 burst 传输完成，并且 PCS,ARR,RCR 均更新完毕。

This sample demonstrates the functionality of using DMA to perform consecutive burst data transfers in TIM1. Each burst transfer updates three registers: PSC, ARR, and RCR. In the update event interrupt, PA0 will toggle. By using a logic analyzer, you can observe that the interval between PA0 toggles will be 400ms for the first and second time, 20ms for the third time, and 200us for the fourth time onwards. This indicates that the two burst transfers are completed, and PSC, ARR, RCR are all updated.

15.6 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 the External Clock Mode 1 functionality of TIM1. It selects the ETR (PA12) pin as the external clock input source and enables the update interrupt. In the interrupt, the LED light will toggle.

15.7 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 functionality of TIM1. It selects the ETR (PA12) pin as the external clock input source and enables the update interrupt. In the interrupt, the LED light will toggle.

15.8 TIM1_InputCapture_TI1FP1

此样例演示了在 TIM1(PA3)输入捕获功能，PA3 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 (PA3). When a clock signal is input to PA3, TIM1 captures it successfully and enters the capture interrupt. With each interrupt, the LED will toggle.

15.9 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(PA3),CH2(PA13),CH3(PA0),CH4(PA1)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，周期为 200KHz

This sample demonstrates the output compare flip-flop functionality in TIM1 compare mode. It enables the output function for CH1 (PA03), CH2 (PA13), CH3 (PA0), and CH4 (PA1) channels. When the counter TIMx_CNT matches TIMx_CCRx, the output signal will toggle at a frequency of 200KHz.

15.10 TIM1_OnePulseOutput

此样例演示了 TIM1 的单脉冲模式，CH2(PA13)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(PA3)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。本例程脉冲宽度计算 (TIM1_ARR-TI1_CCR1)/CLK= (65535-16383) /16000000=3.072ms

This sample demonstrates the single pulse mode of TIM1. When the rising edge is detected on CH2 (PA13) pin, the counter starts counting. When the counter value matches CCR1, CH1 (PA03) outputs a high level until the counter overflows. After the counter overflows, CH1 outputs a low level and the timer stops working. The pulse width in this example is calculated using the formula (TIM1_ARR-TIM1_CCR1)/CLK = (65535-16383)/16000000 = 3.072ms

15.11 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

This sample demonstrates the functionality where Timer 1 is enabled by Timer 3. When Timer 3 starts counting, the LED remains constantly on. When Timer 3 triggers an update event, it enables Timer 1 to start counting. Once Timer 1 starts counting, the LED toggles at a frequency of 5Hz.

15.12 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

率闪烁此例程计算方式为 TIM3_ARR*TIM3_PSC*TIM1_ARR*TIM1_PSC/时钟=800*100*100*1/8000000=1Hz

This sample demonstrates the functionality of cascading Timer 1 (TIM1) and Timer 3 (TIM3) to form a 32-bit counter. Timer 3 acts as the master counter, and its overflow signal serves as the input clock for Timer 1. By configuring the reload register values of Timer 1 and Timer 3, the LED blinks at a frequency of 0.5Hz in the Timer 1 interrupt callback function.

15.13 TIM1_Update_DMA

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 周期变化，在 TIM1 第一次溢出后，PA0 会翻转，此时翻转间隔为 400ms，DMA 开始搬运数据到 TIM1_ARR，第二次 PA0 翻转间隔为 400ms，第三次翻转间隔为 100ms，第四次翻转间隔为 200ms，第四次翻转间隔为 300ms，此时 DMA 搬运结束，后续翻转间隔均为 300ms

This sample demonstrates the functionality of using DMA to transfer data in Timer 1 (TIM1). DMA is used to move data from SRAM to the ARR register of TIM1, resulting in a changing period of TIM1. After the first overflow of TIM1, pin PA0 will toggle with an interval of 400ms. Once the DMA starts transferring data to TIM1_ARR, the second toggle interval of PA0 will be 400ms, the third will be 100ms, the fourth will be 200ms, and the fifth will be 300ms. After the DMA transfer is completed, the subsequent toggle intervals will remain at 300ms.

15.14 TIM1_Update_IT

此样例演示了在 TIM1 中基本计数功能，并使能了更新中断，每次重装 ARR 值时会产生一次更新中断，并在中断中翻转 LED 灯，LED 灯会以 5Hz 的频率进行翻转。

This sample demonstrates the basic counting functionality in Timer 1 (TIM1) and enables the update interrupt. Every time the ARR value is reloaded, an update interrupt is triggered, and the LED light will toggle in the interrupt. The LED light will toggle at a frequency of 5Hz.

16 TIMER16

16.1 TIM16_Counter

此样例演示了在 TIM16 中基本计数功能，并使能了更新中断，每次重装 ARR 值时会产生一次更新中断，并在中断中翻转 LED 灯，LED 灯会以 5Hz 的频率进行翻转。

This sample demonstrates the basic counting functionality in TIM16 and enables the update interrupt. Every time the ARR value is reloaded, an update interrupt is triggered, and the LED light is toggled. The LED light will toggle at a frequency of 5Hz.

17 TIMER17

17.1 TIM17_Counter

此样例演示了在 TIM17 中基本计数功能，并使能了更新中断，每次重装 ARR 值时会产生一次更新中断，并在中断中翻转 LED 灯，LED 灯会以 5Hz 的频率进行翻转。

This sample demonstrates the basic counting functionality in TIM17 and enables the update interrupt. Every time the ARR value is reloaded, an update interrupt is triggered, and the LED light is toggled. The LED light will toggle at a frequency of 5Hz.

18 TIMER3

18.1 TIM3_GatedMode

此样例演示了 TIM3 从模式下的门控触发功能，配置 CH1(PA06)作为门控触发输入信号，并使能触发中断，每次进触发中断，翻转一次 LED 灯，在 IDE 仿真界面里，可以看到，当 PA06 输入低电平时，定时器 CNT 寄存器停止计数，当 PA06 输入高电平时，定时器 CNT 寄存器持续在计数。并且 PA06 引脚上的每产生一次触发中断，LED 都会翻转一次

This sample demonstrates the gated trigger functionality in Timer 3 (TIM3). It configures CH1 (PA06) as the gated trigger input signal and enables the trigger interrupt. Every time the trigger interrupt is triggered, the LED light will toggle. In the IDE simulation interface, you can observe that when the input level on PA06 is low, the timer's CNT register stops counting, and when the input level on PA06 is high, the timer's CNT register continues counting. Every time the trigger interrupt occurs, the LED light will toggle.

18.2 TIM3_InputCapture_DMA

此样例演示了在 TIM3 输入捕获功能，捕获数据通过 DMA 传输到变量 CC1_Capture 中，并通过串口打印输出。

This sample demonstrates the input capture feature in TIM3. The captured data is transferred to the variable CC1_Capture using DMA and printed through the UART.

18.3 TIM3_SynchronizationGated

此样例演示了 TIM1 和 TIM3 同步触发（门控触发模式）功能，TIM1 作为主机，配置为比较输出功能，OC1 输出频率 1kHz，占空比 50%（即 1ms 高电平 1ms 低电平循环），TIM3 作为从机，配置为外部门控触发模式，TIM1 的 OC1 信号连接到 TIM3 作为 TIM3 的门控触发输入，配置 TIM3 的计数溢出周期为 125us，并允许溢出中断，且在溢出中断中对 LED 灯翻转。通过运行程序，通过逻辑分析仪可以看到，当 TIM1_CH1(PA3)输出高电平时，LED 电平翻转；当 TIM1_CH1(PA3)输出低电平时，LED 电平不翻转

This sample demonstrates the synchronous trigger feature (gate trigger mode) of TIM1 and TIM3. TIM1 is configured as the master and set as a compare output with a frequency of 1kHz and a duty cycle of 50% (1ms high level, 1ms low level cycle). TIM3 is configured as the slave and set as an external gate trigger mode with TIM1's OC1 signal connected as TIM3's gate trigger input. TIM3's counter overflow period is set to 125us, and overflow interrupt is enabled. In the overflow interrupt, the LED toggles.By running the program and using a logic analyzer, it can be observed that when TIM1_CH1 (PA3) outputs a high level, the LED toggles; when TIM1_CH1 (PA3) outputs a low level, the LED remains unchanged.

18.4 TIM3_SynchronizationReset

此样例演示了 TIM1 和 TIM3 同步触发（复位触发模式）功能，TIM1 作为主机，配置复位 TRGO 输出(TIM1_CR2.MMS=000)，TIM3 作为从机并且使能事件更新中断，配置收到主机的 TRGO 信号后复位(TIM3_SMCR.SMS=100)。循环打印 TIM3 计数值。

This sample demonstrates the synchronous trigger feature (reset trigger mode) of TIM1 and TIM3. TIM1 is configured as the master and set as a reset trigger output (TIM1_CR2.MMS=000). TIM3 is configured as the slave and enables the update event interrupt. It is set to reset when receiving the reset trigger signal from the master (TIM3_SMCR.SMS=100). The program continuously prints the value of TIM3's counter.

18.5 TIM3_Update

此样例演示了在 TIM3 中基本计数功能，并使能了更新中断，每次重装 ARR 值时会产生一次更新中断，并在中断中翻转 LED 灯，LED 灯会以 5Hz 的频率进行翻转。

This sample demonstrates the basic counting functionality in TIM3 and enables the update interrupt. Every time the ARR value is reloaded, an update interrupt is triggered, and the LED light is toggled. The LED light will toggle at a frequency of 5Hz.

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 the USART. When a character 0x7F is sent from the debug assistant, 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，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 115200 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

19.3 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.

19.4 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，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 115200 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

19.5 USART_HyperTerminal_Polling

此样例演示了 USART 的轮询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，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 115200 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

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 sample demonstrates the early wakeup interrupt feature of the WWDG (Window Watchdog). When the watchdog counter counts down to 0x40, an interrupt is generated. In the interrupt handler, the watchdog is refreshed to ensure that the watchdog does not cause a system 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 sample demonstrates the window function of the WWDG (Window Watchdog). The upper limit of the watchdog window is configured (the lower limit is fixed at 0x3F). The program uses a delay function (delay) to ensure that the watchdog is refreshed within the watchdog counting window. The LED blinking indicates whether the watchdog is fed within the window and no reset occurs.