



# **PY32F420 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\_TempSensor\_TriggerTimer\_IT

此样例演示了 ADC 模块的 Tempsensor 功能和外部触发功能，并通过串口打印出温度值。

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

### 1.5 ADC\_TriggerEPWM1\_SOCA

此样例演示了 EPWM1\_SOCA 外部触发源触发 ADC1 对参考电压进行采样的功能，每次触发、采样和转换完成后，串口每秒打印 4 次 VCC 电压值。

This example demonstrates the function of the EPWM1\_SOCA external trigger source triggering ADC1 to sample the reference voltage. After each trigger, sampling and conversion are completed, the serial port prints the VCC voltage value 4 times per second.

### 1.6 ADC\_Vrebuf

此样例演示了 ADC 的 Vrebuf 功能。

This example demonstrates the Vrebuf function of ADC.

## 1.7 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 COMP

### 2.1 COMP\_CompareGpioVsGpio\_IT

此样例演示了 COMP 比较器中断功能,, PC2 作为比较器正端输入, PC3 作为比较器的负端输入, 当 PC2 的电压大于 PC3 的电压时, LED 灯亮, 小于 PC3 的电压时,LED 灯灭。

This example demonstrates the interrupt function of the COMP comparator,where PC2 serves as the positive input of the comparator and PC3 as the negative input, When the voltage of PC2 is higher than that of PC3 the LED light turns on; when it is lower, the LED light turns off.

### 2.2 COMP\_CompareGpioVsGpio\_Polling

此样例演示了 COMP 比较器轮询功能, PC2 作为比较器正端输入, PC3 作为比较器的负端输入, 当 PC2 的电压大于 PC3 的电压时, LED 灯亮, 小于 PC3 的电压时, LED 灯灭。

This example demonstrates the polling function of the COMP comparator, where PC2 serves as the positive input of the comparator and PC3 as the negative input, When the voltage of PC2 is higher than that of PC3, the LED light turns on; when it is lower, the LED light turns off.

### 2.3 COMP\_CompareGpioVsGpio\_WakeUpFromStop

此样例演示了 COMP 比较器唤醒功能, PC2 作为比较器正端输入, PC3 作为比较器的负端输入, 进入 stop 模式后, 通过调整 PC2 上的输入电压, 产生中断唤醒 stop 模式。

This example demonstrates the COMP comparator wake-up function, where PC2 serves as the positive input of the comparator and PC3 as the negative input, After, entering stop mode, the interrupt wake-up stop mode is generated by adjusting the input voltage on PC2.

## 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 DAC

### 4.1 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, where channel PA4 can output 1/2 of the supply voltage value.

## 5 DMA

### 5.1 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).

## 6 EPWM

### 6.1 EPWM1\_Chopper

此样例演示了 EPWM 的斩波功能，斩波器子模块使用高频载波信号对 AQ 生成的 PWM 波形进行调制。

This sample demonstrates the chopping function of EPWM. The chopper submodule modulates the PWM waveform generated by AQ using a high-frequency carrier signal.

### 6.2 EPWM1\_ComplementarySignals\_Break

此样例演示 EPWM 互补输出刹车功能，跳闸区子模块根据输入信号电平对 AQ 生成的 PWM 波形电平进行强制设置。

This sample demonstrates the EPWM complementary output braking function. The trip zone submodule forcibly sets the PWM waveform level generated by AQ based on the input signal level.

### 6.3 EPWM1\_ComplementarySignals\_DeadTime

此样例演示了 EPWM 互补输出死区功能，死区发生器子模块给 AQ 生成的 PWM 波形添加死区。

This sample demonstrates the dead-time function of EPWM complementary output. The dead-time generator submodule adds dead-time to the PWM waveforms generated by AQ.

### 6.4 EPWM1\_InputFreqDivision

此样例演示了 EPWM 输入信号分频功能，数字比较(DC)子模块对输入信号进行分频处理后供给 AQ 模块生成波形使用。

This sample demonstrates the input signal frequency division function of EPWM. The Digital Comparator (DC) submodule performs frequency division on the input signal, and the processed signal is then supplied to the AQ module for waveform generation.

### 6.5 EPWM1\_PWM

此样例演示了 EPWM 输出基本 PWM 信号功能，动作限定(AQ)子模块根据设置好的周期、比较器的值生成不同占空比的 PWM 信号。

This sample demonstrates the basic PWM signal output function of EPWM. The Action Qualifier (AQ) submodule generates PWM signals with different duty cycles based on the configured period and comparator values.



## 6.6 EPWM1\_Sync\_GPIO

此样例演示 EPWM 外部信号触发不同 EPWM 同步以及移相功能。

This sample demonstrates the external signal-triggered synchronization and phase-shifting functions of different EPWMs.

## 6.7 EPWM1\_Sync\_HRPWM1

此样例演示某一 EPWM 触发其他 EPWM 同步以及移相功能。

This sample demonstrates the synchronization and phase-shifting functionality where one EPWM triggers other EPWMs.

## 6.8 EPWM1\_Update\_IT

此样例演示了在 EPWM1 中基本计数功能，并使能了事件触发(ET)子模块的中断，事件在 TBCTR=TBPRD 时生成，同时开启事件分频，每两次 TBCTR=TBPRD 时会产生一次中断，并在中断中翻转 LED 灯。

This sample demonstrates the basic counting function in EPWM1, with the Event Trigger (ET) submodule's interrupt enabled. The event is generated when TBCTR equals TBPRD. Additionally, event prescaling is activated, producing an interrupt every two occurrences of TBCTR equaling TBPRD, and the LED is toggled within the interrupt service routine.

## 7 EXTI

### 7.1 EXTI\_ToggleLed\_IT

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 PB0 generates an interrupt, and the LED light in the interrupt function toggle once.

## 8 FDCAN

### 8.1 FDCAN\_Classic\_StandardID\_IT

此样例演示了采用 CAN2.0 协议标准帧中断方式与 PCAN-View 的通信功能，MCU 首先自动向 PCAN-View 发送 ID 为 0x111 的 2 帧数据，PCAN-View 接收到数据后，然后手动通过 PCAN-View 向 MCU 发送 ID 为 0x111 的 2 帧数据。

This sample demonstrates the communication function between CAN2.0 protocol standard frame interrupt mode and PCAN-View. MCU automatically sends 2 frames of data with ID 0x111 to PCAN-View at first. After PCAN-View receives the data, MCU manually sends 2 frames of data with ID 0x111 to MCU through PCAN-View.

### 8.2 FDCAN\_Classic\_StandardID\_Loopback\_polling

此样例演示了采用 CAN2.0 协议、标准帧、轮询方式的内部回环通信。

This sample demonstrates internal loopback communication using CAN2.0 protocol, standard frames, polling.

### 8.3 FDCAN\_ExtendID\_polling

此样例演示了采用 CANFD 协议扩展帧轮询方式与 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 0x12345678 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 polling 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 0x12345678 to the MCU through the PCAN View. The MCU will automatically print the received data through the serial port.

## 9 FLASH

### 9.1 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.

### 9.2 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\_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.

### 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 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.

## 11 I2C

### 11.1 I2C\_TwoBoards\_Com\_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 I2C communication using DMA. The master device 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 transmit and receive data, the LEDs on both boards will be constantly lit.

### 11.2 I2C\_TwoBoards\_Com\_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.

### 11.3 I2C\_TwoBoards\_Com\_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 I2C communication using interrupt. The master device 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 transmit and receive data, the LEDs on both boards will be constantly lit.

### 11.4 I2C\_TwoBoards\_Com\_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 I2C communication using polling. The master device 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 transmit and receive data, the LEDs on both boards will be constantly lit.

## 12 IWDG

### 12.1 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 1000ms 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 900ms，程序能一直正常运行（LED 灯闪烁），如果喂狗时间 1100ms，程序会一直复位（LED 灯熄灭）。

This example demonstrates the function of IWDG (Independent Watchdog). Set IWDG to count 1000ms 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 900ms, the program can always run normally (LED blink), if the time is 1000ms, the program will always reset (LED off).

## 13 LPTIM

### 13.1 LPTIM\_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.

### 13.2 LPTIM\_Wakeup\_WFI

此样例演示了 LPTIM 连续模式中断唤醒 STOP 模式。

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



## 14 LPUART

### 14.1 LPUART\_HyperTerminal\_DMA

此样例演示了 LPUART 的 DMA 方式发送和接收数据，LPUART 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 LPUART to send an amount of data in DMA mode. LPUART 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

### 14.2 LPUART\_HyperTerminal\_IT

此样例演示了 LPUART 的中断方式发送和接收数据，LPUART 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 LPUART to send an amount of data in interrupt mode. LPUART 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

### 14.3 LPUART\_HyperTerminal\_Polling

此样例演示了 LPUART 的轮询方式发送和接收数据，LPUART 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 LPUART to send an amount of data in polling mode. LPUART 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

## 15 OPA

### 15.1 OPA\_FOLLOWER

此样例演示了 OPA 的压随模式功能，输出端会输出和正端一样的电压值。

This example demonstrates the follower mode function of OPA, the output terminal outputs the same electrical output as the positive terminal value.

### 15.2 OPA\_PGA

此样例演示了 OPA 的可编程增益放大器功能,OPA 的输出端会输出正端 2 倍的电压值。

This example demonstrates the programmable gain amplifier function of OPA, where the output terminal outputs a voltage value twice of the positive terminal.

### 15.3 OPA\_STANDALONE

此样例演示了 OPA 的独立模式功能,OPA 的负端和输出端短接，输出端会输出和正端一样的电压值。

This example demonstrates the independent mode function of OPA, where the negative and output terminals of OPA are short circuited, and the output terminal outputs the same electrical output as the positive terminal value.

## 16 PWR

### 16.1 PWR\_LPRUN

此样例演示了进入和退出 lprun 模式。

This example demonstrates entering and exiting the lprun mode.

### 16.2 PWR\_LPSLEEP\_WFI

此样例演示了 lpsleep 模式下，GPIO 外部中断唤醒功能。

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

### 16.3 PWR\_PVD

此样例演示了 PVD 电压检测功能。当供电电压低于 3.1V 时，LED 会点亮，高于 3.1V 时，LED 灯会熄灭。

This sample demonstrates the PVD (Programmable Voltage Detector) voltage detection functionality. When the supply voltage is lower than 3.1V, the LED will light up. When the supply voltage is higher than 3.1V, the LED will turn off.

### 16.4 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.

### 16.5 PWR\_STANDBY

此样例演示了 standby 模式下，通过 wakeuppín 唤醒功能。

This sample demonstrates the wake-up feature using the wakeup pin in standby mode.

### 16.6 PWR\_STOP0\_WFI

此样例演示了在 stop0 模式下，使用 GPIO 中断唤醒。

This example demonstrates using GPIO interrupt wake-up in stop0 mode.

## 16.7 PWR\_STOP1\_WFE

此样例演示了 stop1 模式下, GPIO 外部事件唤醒功能。

This sample demonstrates the GPIO external event wake-up feature in stop1 mode.

## 17 RCC

### 17.1 RCC\_HSE\_Output

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.

### 17.2 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.

### 17.3 RCC\_LSE\_Output

此样例使能 LSE，并通过 MCO (PA08) 引脚输出。

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

### 17.4 RCC\_LSI\_Output

此样例使能 LSI，并通过 MCO (PA08) 引脚输出。

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

### 17.5 RCC\_PLL\_Output

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 HSI。

This sample configures the system clock to use the PLL (Phase-Locked Loop) clock source with HSI as the input clock source and outputs it through the MCO (PA08) pin.

## 18 RTC

### 18.1 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；当到达闹钟时间时，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".

### 18.2 RTC\_Tamper\_IT

此样例演示 TAMPER 中断功能，TAMPER 中断事件把数据备份寄存器的内容清除。

This example demonstrates the TAMPER interrupt function, where the TAMPER interrupt event clears the contents of the data backup register.

### 18.3 RTC\_WakeUpAlarm

此样例演示通过 RTC 闹钟中断每隔 1s 将 MCU 从 STOP1 模式下唤醒，每次唤醒会翻转 LED，LED 翻转间隔为 1s。

This sample demonstrates waking up the MCU from the STOP1 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.

## 19 SPI

### 19.1 SPI\_FullDuplex\_ExternalFLASH

此样例演示了主机 SPI 通过轮询方式进行通讯，从机使用 FLASH 外设芯片 P25Q64，按下 user 按键，主机先向从机写 15bytes 数据为 0x1~0xf，然后再从 FLASH 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 the host SPI communication through polling, the slave uses FLASH peripheral chip P25Q64, press the user button, the host first to the slave to write 15bytes of data for the 0x1 ~ 0xf, and then from the FLASH will be written to read out the data, read the success of the host board, the small light is in the “always on” state! After successful reading, the small light on the host board is in the “always on” state.

### 19.2 SPI\_TwoBoards\_FullDuplex\_DMA

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，主设备提供通信时钟 SCK，通过 MOSI/MISO 引脚发送/接收数据。从设备通过 MOSI/MISO 引脚接收/发送数据。数据以主机提供的 SCK 沿同步被移位，完成全双工通信。

This sample is a demonstration of using DMA to communicate with a serial peripheral interface (SPI) and an external device in full-duplex serial mode. The master device provides the communication clock SCK and sends/receives data through the MOSI/MISO pin. The slave device receives/transmits data through the MOSI/MISO pins. The data is shifted synchronously along the SCK provided by the master to complete full-duplex communication.

### 19.3 SPI\_TwoBoards\_FullDuplex\_IT

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，主设备提供通信时钟 SCK，通过 MOSI/MISO 引脚发送/接收数据。从设备通过 MOSI/MISO 引脚接收/发送数据。数据以主机提供的 SCK 沿同步被移位，完成全双工通信。

This sample is a demonstration of using interrupt to communicate with a serial peripheral interface (SPI) and an external device in full-duplex serial mode. The master device provides the communication clock SCK and sends/receives data through the MOSI/MISO pin. The slave device receives/transmits data through the MOSI/MISO pins. The data is shifted synchronously along the SCK provided by the master to complete full-duplex communication.

### 19.4 SPI\_TwoBoards\_FullDuplex\_Polling

此样例是利用轮询对串口外设接口 (SPI) 与外部设备以全双工串行方式进行通信的演示，主设备提供通信时钟 SCK，通过 MOSI/MISO 引脚发送/接收数据。从设备通过 MOSI/MISO 引脚接收/发送数据。数据以主机提供的 SCK 沿同步被移位，完成全双工通信。

This sample is a demonstration of using polling to communicate with a serial peripheral interface (SPI) and an external device in full-duplex serial mode. The master device provides the communication clock SCK and sends/receives data through the MOSI/MISO pin. The slave device receives/transmits data through the MOSI/MISO pins. The data is shifted synchronously along the SCK provided by the master to complete full-duplex communication.



## 20 TIM

### 20.1 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

### 20.2 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（高电平）后，PWM 信号关闭，由于 BDTR.AOE 置位，所以刹车信号取消（低电平）后，继续 pwm 输出，此样例实现了死区功能。CH1 -> PA8 CH1N -> PB13 刹车输入 -> PA6 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 brake function of the timer, the CH1 and CH1N complementary pwm outputs. After receiving the brake signal (high level) from the external IO port, the PWM signal is turned off. Because BDTR.AOE is set, the pwm output continues after the brake signal is cancelled (low level). This example realizes the dead zone function CH1 -> PA8 CH1N -> PB13 Brake input -> PA6 By adjusting the OCxE, CCxP, OISx, CCxNE, CCxNP, OISxN configuration, which can realize the brake function of a variety of applications

### 20.3 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 20ms，第四次及后续变为 200us，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, PA0 will be flipped. Through the monitoring of logic analyzer, it can be seen that the flipping interval of PA0 will change from 400ms for the first time, 400ms for the second time, 20ms for the third time, and 200us for the fourth and subsequent times. At this time, the two burst transmission is completed, and PCS, ARR and RCR are all updated.

### 20.4 TIM1\_ExternalClockMode1

此样例演示了 TIM1 的外部时钟模式 1 功能，选择 ETR(PA3)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(PA3) pin as the external clock input source and enable the update interrupt to flip the LED light in the interrupt.

## 20.5 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

## 20.6 TIM1\_OnePulseOutput

此样例演示了 TIM1 的单脉冲模式，CH2(PA9)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(PA8)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  $(TIM1\_ARR - TIM1\_CCR1) / CLK = (65535 - 16383) / 8000000 = 6.144ms$

This sample demonstrates the one pulse mode of TIM1. The rising edge on the CH2(PA9) pin triggers the counter to start counting. When the count value matches CCR1, CH1(PA8) outputs a high level. When the counter overflows, CH1 outputs the low level again. After the counter overflows, the timer stops working. This example pulse width calculation  $(TIM1\_ARR - TIM1\_CCR1) / CLK = (65,535 - 16383) / 8,000,000 = 6.144ms$

## 20.7 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%，本例程周期为  $8000000 / 50 / 800 = 200Hz$

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%. The period is  $8000000 / 50 / 800 = 200Hz$

## 20.8 TIM1\_TIM2\_Cascade

此样例实现了 TIM1 和 TIM2 级联成 48 位计数器，TIM2 做主机，TIM2 的计数溢出信号作为 TIM1 的输入时钟，通过配置 TIM1 和 TIM2 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 TIM2 into a 48-bit counter, with TIM2 as the host. The count overflow signal of TIM2 acts as the input clock of TIM1. By configuring the reloaded register values of TIM1 and TIM2, the LED is toggled at 0.5Hz (in the TIM1 interrupt callback function).

## 20.9 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，第一次 PA0 翻转间隔为 400ms，第二次翻转间隔为 100ms，第三次翻转间隔为 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 function of using DMA to transfer data in TIM1, carrying data from SRAM to ARR register by DMA to achieve TIM1 cycle change. After the first overflow of TIM1, PA0 will toggle, at this time the toggle interval is 400ms. DMA starts to carry data to TIM1\_ARR, the first PA0 toggle interval is 400ms, the second toggle interval is 100ms, the third toggle interval is 200ms, the fourth toggle interval is 300ms, at this time the DMA carrying ends, the subsequent toggle interval are 300ms

## 20.10 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 basic count function of the TIM1 and enable update interrupt. Each time an update interrupt is generated, the ARR value is reloaded and the LED light is toggled in the interrupt. The LED light is toggled at a frequency of 5Hz.

## 21 UART

### 21.1 UART\_HyperTerminal\_DMA

此样例演示了 UART 的 DMA 方式发送和接收数据，UART 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 UART to send an amount of data in DMA mode. UART 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

### 21.2 UART\_HyperTerminal\_IT

此样例演示了 UART 的中断方式发送和接收数据，UART 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 UART to send an amount of data in interrupt mode. UART 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

### 21.3 UART\_HyperTerminal\_Polling

此样例演示了 UART 的轮询方式发送和接收数据，UART 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 UART to send an amount of data in polling mode. UART 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

## 22 USART

### 22.1 SCI\_HyperTerminal\_AutoBaud\_IT

此样例演示了 SCI 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 SCI. When the debugging assistant sends a character 0x7F, the MCU will respond with the string: "Auto BaudRate Test".

### 22.2 SCI\_HyperTerminal\_DMA

此样例演示了 SCI 的 DMA 方式发送和接收数据，SCI 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 SCI to send an amount of data in DMA mode. SCI 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.

### 22.3 SCI\_HyperTerminal\_IT

此样例演示了 SCI 的中断方式发送和接收数据，SCI 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 SCI to send an amount of data in interrupt mode. SCI 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.

### 22.4 SCI\_HyperTerminal\_Polling

此样例演示了 SCI 的轮询方式发送和接收数据，SCI 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 SCI to send an amount of data in polling mode. SCI 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.



## 23 WWDG

### 23.1 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.

### 23.2 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.