



PY32F005 series

32-bit ARM® Cortex®-M0+ microcontroller

HAL Library Sample Manual

---

**PY32F005 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 sample demonstrates the analog watchdog function of the ADC, which enters the watchdog interrupt when the voltage value of the channel that opens the analog watchdog exceeds the upper and lower limit.

### 1.2 ADC\_MulChannels

此样例演示了 ADC 的多通道转换功能。

This example demonstrates the multi-channel conversion function of ADC.

### 1.3 ADC\_SingleConversion\_TriggerTimer\_IT

此样例演示了 ADC 的 TIM 触发和中断的功能。

This sample demonstrates the TIM trigger function and IT function of the ADC.

### 1.4 ADC\_Temperature

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.

### 1.5 ADC\_Vrefbuf

此样例演示了 ADC 的通道 1 采样功能和 VREFBUF 的功能，通过 VREFBUF 推算出通道 1 的输入电压。

This sample demonstrates the channel 1 sampling function of the ADC and the function of VREFBUF, from which the input voltage of channel 1 is deduced.

### 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 COMP

### 2.1 COMP\_CompareGpioVs1\_2VCC\_Polling

此样例演示了 COMP 比较器轮询功能，PA02 作为比较器正端输入，1/2VCCA 作为负端输入，通过调整 PA02 上的输入电压，当检测到比较器输出状态为高时，LED 灯亮，比较器输出状态为低时，LED 灯灭。

This example demonstrates the COMP comparator polling function, with PA02 as the negative input of the comparator and 1/2VCCA as the positive input. By adjusting the input voltage on PA02, the LED lights up when the comparator output state is detected to be high, and turns off when the comparator output state is low.

## 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 EXTI

### 4.1 EXTI\_ToggleLed\_IT

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

This example demonstrates the GPIO external interrupt function, where each rising edge on the PA0 pin generates an interrupt, and the LED light in the interrupt function flips once.

### 4.2 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.

## 5 FLASH

### 5.1 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.

### 5.2 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.

## 6 GPIO

### 6.1 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 example mainly demonstrates the FAST IO output function of GPIO, which can achieve a single cycle flip speed.

### 6.2 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 example demonstrates the GPIO output mode, configuring the LED pin to be in digital output mode, and flipping the LED pin level every 250ms. Running the program, you can see that the LED light flashes at a frequency of 2Hz.



## 7 I2C

### 7.1 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 example demonstrates I2C communication through interrupt mode. The master first sends 15byte data to the slave, and then receives the 15byte data sent by the slave. After the master and slave successfully receive the data, the small lights on the master and slave boards are in a "constant on" state.

### 7.2 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 example demonstrates I2C communication through polling. The host first sends 15byte data to the slave, and then receives the 15byte data sent by the slave. After the host and slave receive the data successfully, the small lights on the host and slave boards are in a "constantly on" state.

### 7.3 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 example demonstrates I2C communication through interrupt mode. The master first sends 15byte data to the slave, and then receives the 15byte data sent by the slave. After the master and slave successfully receive the data, the small lights on the master and slave boards are in a "constant on" state.

## 8 IWDG

### 8.1 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 钟，程序能一直正常运行（LED 灯闪烁），如果喂狗时间超过 1s 钟，程序会一直复位（LED 灯熄灭）。

This example demonstrates the IWDG watchdog function, configuring the watchdog overload count value, resetting after counting for 1 second, and then adjusting each time The feeding time of the dog (code in the main function while loop) can be observed that if the feeding time is less than 1 second each time, the program Can continue to operate normally (LED flashing), if the dog feeding time exceeds 1 second, the program will continue to reset (LED light off).

## **9 LPTIM**

### **9.1 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.

### **9.2 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.

## 10 OPA

### 10.1 OPA\_Follower

此样例演示了 OPA 的电压跟随功能，输出端会输出和同相端一样的电压

This example demonstrates the voltage follow function of OPA. the output terminal will output the same voltage as the non-inverting terminal.

### 10.2 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.

### 10.3 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.

## 11 PWM

### 11.1 PWM\_PWM

本例程输出 1 路 PWM，通道 1 的占空比为 20%，本例程周期为  $8000000/50/200=800\text{Hz}$

This sample outputs 1 channel PWM, the duty cycle of channel 1 is 20%.The period is  $8000000/50/200=800\text{Hz}$

## 12 PWR

### 12.1 PWR\_DEEPSTOP\_WFE

此样例演示了在 deep stop 模式下，使用 GPIO 事件唤醒。

This example demonstrates using GPIO event wake-up in deep stop mode.

### 12.2 PWR\_PVD

此样例演示了 PVD 电压检测功能。当供电电压低于 2.8V 时，LED 会点亮，高于 2.8V 时，LED 灯会熄灭。

This sample demonstrates the PVD (Programmable Voltage Detector) voltage detection functionality. When the supply voltage is lower than 2.8V, the LED will light up. When the supply voltage is higher than 2.8V, the LED will turn off.

### 12.3 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.

### 12.4 PWR\_STOP\_WFE

此样例演示了在 stop 模式下，使用 GPIO 事件唤醒。

This example demonstrates using GPIO event wake-up in stop mode.

### 12.5 PWR\_STOP\_WFI

此样例演示了在 stop 模式下，使用 GPIO 外部中断唤醒。

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

## 13 RCC

### 13.1 RCC\_HSEBypassOutput

此样例演示了时钟输出功能，可输出 HSE 波形。

This example demonstrates the clock output function, which can output HSE waveforms.

### 13.2 RCC\_HSIOutput

此样例配置系统时钟为 HSI，并通过 MCO (PB1) 引脚输出。

This example configures the system clock as HSI and outputs it through the MCO (PB1) pin.

### 13.3 RCC\_LSEOutput

此样例使能 LSE，并通过 MCO (PB1) 引脚输出。

This sample enables LSE and outputs it on the MCO (PB1) pin.

### 13.4 RCC\_LSIOutput

此样例使能 LSI，并通过 MCO (PB1) 引脚输出。

This sample enables the LSI and is output via the MCO (PB1) pin.

## 14 SPI

### 14.1 SPI\_TwoBoards\_FullDuplexMaster\_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 interrupts 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.

### 14.2 SPI\_TwoBoards\_FullDuplexMaster\_Polling

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。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 the Serial Peripheral Interface (SPI) communicating with an external device in full-duplex serial mode by polling. The master device provides the communication clock SCK and sends/receives data via 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.

### 14.3 SPI\_TwoBoards\_FullDuplexSlave\_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 interrupts 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.

### 14.4 SPI\_TwoBoards\_FullDuplexSlave\_Polling

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。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 the Serial Peripheral Interface (SPI) communicating with an external device in full-duplex serial mode by polling. The master device provides the communication clock SCK and sends/receives data via 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.

## 15 TIM

### 15.1 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

### 15.2 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。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 (low 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 (high level). This example realizes the dead zone function. By adjusting the OCxE, CCxP, OISx, CCxNE, CCxNP, OISxN configuration, which can realize the brake function of a variety of applications

### 15.3 TIM1\_ExternalClockMode1

此样例演示了 TIM1 的外部时钟模式 1 功能，选择 ETR(PA2)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(PA2) pin as the external clock input source and enable the update interrupt to flip the LED light in the interrupt.

### 15.4 TIM1\_InputCapture\_TI1FP1

此样例演示了在 TIM1(PB2)输入捕获功能，PB2 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(PB2), PB2 input clock signal, when TIM1 capture success, will enter the capture interrupt,and toggle the LED in the interrupt

## 15.5 TIM1\_OnePulseOutput

此样例演示了 TIM1 的单脉冲模式, CH2(PB0)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(PB2)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  $(TIM1\_ARR-TI1\_CCR1)/CLK = (65535-16383)/8000000=6.144ms$

This sample demonstrates the one pulse mode of TIM1. The rising edge on the CH2(PB0) pin triggers the counter to start counting. when the count value matches CCR1, CH1(PB2) 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  $(TIM1\_ARR-TI1\_CCR1)/CLK = (65,535-16383)/8,000,000 = 6.144ms$

## 15.6 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  $8000000/50/800=200Hz$

## 15.7 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.

## 16 UART

### 16.1 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

### 16.2 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

## 17 USART

### 17.1 SCI\_HyperTerminal\_AutoBaud\_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 example demonstrates USART's automatic Baud detection function. The debugging assistant sends a character 0x7F and MCU feedback string: Auto BaudRate Test.

### 17.2 SCI\_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 128byte），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.

### 17.3 SCI\_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

### 17.4 SCI\_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

