



# **PY32F005 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 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\_Init

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\_Init

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\_Init

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\_2Vrefbuf\_Polling\_Init

此样例演示了 COMP 比较器轮询功能, PA2 作为比较器正端输入,  $1/2V_{REFBUF}$  作为比较器负端输入, 当 PA2 的电压大于  $1/2V_{REFBUF}$  电压时, LED 灯亮, 小于  $1/2V_{REFBUF}$  电压时, LED 灯灭。

This example demonstrates the polling function of the COMP comparator, with PA2 as the positive input and  $1/2V_{REFBUF}$  as the negative input. When the voltage of PA2 is greater than  $1/2V_{REFBUF}$  voltage, the LED lights up, and when it is less than  $1/2V_{REFBUF}$  voltage, the LED lights up.

## 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 EXTI

### 4.1 EXTI\_ToggleLed\_IT\_Init

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

This example demonstrates the GPIO external interrupt function, each falling edge on the PA0 pin will generate an interrupt, and the LED will toggle once in the interrupt handle function.

### 4.2 EXTI\_WakeUp\_Event

此样例演示了 stop 模式下，GPIO 外部事件唤醒功能。

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

## **5 FLASH**

### **5.1 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.

### **5.2 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.

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

### 6.3 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.



## 7 I2C

### 7.1 I2C\_TwoBoard\_Communication\_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、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.

### 7.2 I2C\_TwoBoard\_Communication\_Polling\_Init

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.

### 7.3 I2C\_TwoBoard\_Com\_MEM\_Init

此样例演示了主机 I2C 通过中断方式进行通讯，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 interrupt. 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.

### 7.4 I2C\_TwoBoard\_IndefiniteLengthData\_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.

## 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，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 watchdog function. Configure the watchdog to count for 1s and then reset. By adjusting the time of each feed dog (code in the while loop of the main function), it can be observed following situation: if each dog feeding time is less than 1s, the program can always run normally (LED toggle). if the dog feeding time is more than 1s, the program will always reset (LED off)

## 9 LPTIM

### 9.1 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.

### 9.2 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.

## 10 OPA

### 10.1 OPA\_Follower

此样例演示了 OPA 的电压跟随功能，输出端会输出和正端一样的电压

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\_Init

此样例演示了使用 PWM PWM2 模式输出一路频率为 40Hz 占空比分别为 60%的 PWM 波形。

This example demonstrates the use of PWM PWM2 mode to output PWM wave with a frequency of 40Hz and a duty cycle of 60%, respectively.

## 12 PWR

### 12.1 PWR\_DEEPSTOP\_WFI

此样例演示了在 deepstop 模式下，使用 GPIO 中断唤醒。

This sample demonstrates waking up from deepstop mode using GPIO interrupt.

### 12.2 PWR\_PVD

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

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

### 12.3 PWR\_SLEEP\_WFE

此样例演示了在 sleep 模式下，使用 GPIO 事件唤醒。

This sample demonstrates waking up in sleep mode using GPIO events.

### 12.4 PWR\_SLEEP\_WFI

此样例演示了在 sleep 模式下，使用 GPIO 中断唤醒。

This sample demonstrates waking up in sleep mode using GPIO interrupt.

### 12.5 PWR\_STOP\_WFE

此样例演示了在 stop 模式下，使用 GPIO 事件唤醒。

This sample demonstrates waking up in stop mode using GPIO event.

### 12.6 PWR\_STOP\_WFI

此样例演示了在 stop 模式下，使用 GPIO 中断唤醒。

This sample demonstrates waking up from stop mode using GPIO interrupt.

## 13 RCC

### 13.1 RCC\_HSE\_Bypass\_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 function, which can output HSE waveforms.

### 13.2 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 function, which can output the HSI waveform.

### 13.3 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.

### 13.4 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.



## 14 SPI

### 14.1 SPI\_TwoBoards\_FullDuplex\_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，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\_FullDuplex\_Polling\_Init

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.

## 15 TIM

### 15.1 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.

### 15.2 TIM1\_InputCapture\_Init

此样例演示了 TIM1 的输入捕获功能，配置 PB5 作为输入捕获引脚，PB5 每检测到一个下降沿触发捕获中断在捕获中断回调函数中翻转 LED 灯。

This sample demonstrates the input capture function of TIM1. Configure PB5 as input capture pin. Whenever PB5 detects a falling edge it triggers a capture interrupt and toggle the LED in the capture interrupt callback function.

### 15.3 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 PB5，通道 2 为输入模式，映射到 PB4。当 PB4 上检测到一个上升沿时，PB5 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 PB5, while channel 2 is configured as input mode and mapped to pin PB4. When an rising edge is detected on PB4, a 20ms delay is applied, and then PB5 will output a pulse with a width of 80ms.

### 15.4 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.

### 15.5 TIM1\_TIM13\_Cascade

此样例演示了 TIM1 和 TIM13 级联成 32 位计数器，TIM13 做主机，TIM3 的 CH1 信号作为 TIM1 的输

入时钟。TIM13 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 TIM13 as a 32-bit counter, with TIM3 as the master and the overflow signal of TIM13 as the input clock of TIM1. TIM3 counts every 1ms, and after counting 1000 times, it overflows and TIM1 counts once.

## 15.6 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 , LED toggled when the update interrupt is generated.

## 16 UART

### 16.1 UART\_HyperTerminal\_IndefiniteLengthData\_IT

此样例演示了 UART 的中断方式发送和接收不定长数据，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，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 UART to send and receive variable length data. UART 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.

### 16.2 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

### 16.3 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

## 17 USART

### 17.1 SCI\_HyperTerminal\_AutoBaud\_IT\_Init

此样例演示了 SCI 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 SCI. 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.

### 17.2 SCI\_HyperTerminal\_IndefiniteLengthData\_IT

此样例演示了 SCI 的中断方式发送和接收不定长数据, 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, 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 SCI to send and receive variable length data. SCI 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\_Init

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.

### 17.4 SCI\_HyperTerminal\_Polling\_Init

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 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.