PY32F002A series
32-bit ARM® Cortex®-M0+ microcontroller

HAL Library Sample Manual

1 ADC

1.1 ADC_MultiChannelSwitch

此样例演示了 ADC 的多通道切换。

This sample demonstrates the multichannel switching of ADC.

1.2 ADC_SingleConversion_AWD

此样例演示了 ADC 的模拟看门狗功能。

This sample demonstrates the analogue watchdog function of the ADC.

1.3 ADC_SingleConversion_TriggerSW_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 function of the ADC module, which samples the value of VREFINT, calculates the value of VCC, and prints it out via the serial port to print it out.

1.4 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.

1.5 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.

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 比较器中断功能，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 比较器轮询功能，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 引脚上的电平变化情况。

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 的电压，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 的电压，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_Computing_Results

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 EXTI

4.1 EXTI_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.

4.2 EXTI_IT

此样例演示了 GPIO 外部中断功能，PB2 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 PB2 generates an interrupt, and the LED toggles once in the interrupt function.

5 FLASH

5.1 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.

5.2 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.

5.3 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.

6 GPIO

6.1 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.

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 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.

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 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.

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 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 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 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).

9 LPTIM

9.1 LPTIM_WakeUp

此样例演示了 LPTIM 中断唤醒 stop 模式，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.

10 PWR

10.1 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.

10.2 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.

11 RCC

11.1 RCC_HSE_OUTPUT

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.

11.2 RCC_HSI_OUTPUT

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.

11.3 RCC_LSI_OUTPUT

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 (PA01) pin.

12 SPI

12.1 SPI_TwoBoards_FullDuplexMaster_IT

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 master, providing the communication clock signal (SCK). The master sends data via the MOSI pin, and receives data from the slave via the MISO pin. The data is shifted synchronously with the clock signal provided by the master, enabling full-duplex communication.

12.2 SPI_TwoBoards_FullDuplexSlave_IT

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,此接口设置为从模式，通信时钟 SCK 由外部主设备提供。从机通过 MOSI 引脚接收主机数据,从 MISO 引脚发送数据，数据以主机提供的 SCK 沿同步被移位，完成全双工通信。

This sample 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.

13 TIM1

13.1 TIM1_6StepPWM

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

This sample demonstrates the "Six-step PWM Generation" feature of the advanced timer. It uses the systick interrupt as the trigger source for the COM commutation event to achieve commutation of a brushless motor.

13.2 TIM1_ARR

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 main.c 中 的 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 灯以 1.875HZ 的频率翻转

This sample implements the basic counting function of the timer, as well as demonstrates the ARR AutoReload function, the sample toggles the LED in the timer reload interrupt. Enable the auto-reload function by configuring TimHandle.Init.AutoReloadPreload = TIM_AUTORELOAD_PRELOAD_ENABLE in main.c. The new ARR value takes effect the fourth time the interrupt is entered; configure TimHandle.Init. AutoReloadPreload=TIM_AUTORELOAD_PRELOAD_DISABLE to disable the AutoReload function, the new ARR value will take effect in the third time into the interrupt, and after that, the LED will toggle at a frequency of 1.875HZ.

13.3 TIM1_InputCapture

此样例演示了 TIM1 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 PA8, TIM1 captures it successfully and enters the capture interrupt. With each interrupt, the LED will toggle.

13.4 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+1)/800=196Hz

This sample demonstrates the generation of 4 PWM signals. Channel 1 has a duty cycle of 20%, channel 2 has a duty cycle of 40%, channel 3 has a duty cycle of 60%, and channel 4 has a duty cycle of 80%. The period of this example is calculated as 8000000/(50+1)/800=196Hz.

14 TIM16

14.1 TIM16_ARR

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.

15 USART

15.1 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".

15.2 USART_HyperTerminal_IndefiniteLengththData_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.

15.3 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

15.4 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