PY32F002B series
32-bit ARM® Cortex®-M0+ microcontroller
LL 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_TriggerTimer_AWD

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.

1.3 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 triggr function and IT function of the ADC.

1.4 ADC_SingleConversion_TriggerTimer_Polling

此样例演示了 ADC 的 TIM 触发和轮询的功能。

This sample demonstrates the TIM triggr function and polling function of the ADC.

1.5 ADC_Temperature_Init

此样例演示了 ADC 的 Tempsensor 功能。

This sample demonstrates the Tempsensor function of the ADC.

1.6 ADC_VrefbufManualOffsetCalibration

此样例演示了 ADC 在 Vrefbuf1.5V 的情况下手动校准 Offset 的功能，通过设置 PA3 为推挽输出模式，手动修改 Offset 值，直到采样 PA3 的 DR 值在 0-1 范围内。

This example demonstrates the function of manually calibrating the Offset of ADC in the case of Vrefluf1.5V, by setting PA3 as a push-pull output Mode, manually modify the Offset value until the DR

value of sampled PA3 is within the range of 0-1.

1.7 ADC_VrefintAndVrefbuf2P5_Init

此样例演示了 ADC 的通道 4 采样功能和 VREFBUF 的功能，通过 VREFBUF 推算出通道 4 的输入电压。

This sample demonstrates the channel 4 sampling function of the ADC and the function of VREFBUF, from which the input voltage of channel 4 is deduced.

1.8 ADC_VrefintAndVrefbuf_Init

此样例演示了 ADC 的 VREFINT 采样功能和 VREFBUF 的功能，通过 VREFINT 推算出 VREFBUF 的电压。

This sample demonstrates the ADC's VREFINT sampling function and the VREFBUF function,which calculates the voltage of VREFBUF from VREFINT.

2 COMP

2.1 COMP_CompareGpioVs1_2VCC_Polling_Init

此样例演示了 COMP 比较器轮询功能，PA04 作为比较器负端输入，1/2VCCA 作为正端输入，通过调整 PA04 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 sample demonstrates the COMP polling function, with PA04 as the negative comparator input and 1/2VCCA as the positive input. Adjust the input voltage on PA04, the LED will be on when the comparator output state is detected as high and be off when the comparator output state is low.

2.2 COMP_CompareGpioVs1_2VCC_WakeupFromSleep

此样例演示了 COMP 比较器唤醒功能，PA04 作为比较器负端输入，1/2VCC 作为比较器正端输入，上完电 LED 灯会常亮，用户点击按钮，LED 灯灭，进入 sleep 模式，通过调整 PA04 上的输入电压，产生中断唤醒 sleep 模式。

This example demonstrates the wake-up function of the COMP comparator, with PA04 as the negative input and 1/2VCC as the positive input. After power on, the LED light will remain on. When the user clicks the button, the LED light will go out and enter sleep mode. By adjusting the input voltage on PA04, an interrupt wake-up sleep mode is generated.

2.3 COMP_CompareGpioVs1_2VCC_Window

此样例演示了 COMP 比较器的 window 功能，比较器 1 正端用比较器 2 的正端(VREFCMP)作为输入，PB0 作为比较器负端输入，当 PB0 的电压值大于 1.65V 时,LED 灯灭，小于 1.65V 时,LED 灯亮。

This example demonstrates the window function of the COMP. The positive terminal of the COMP1 is connected the positive terminal of the COMP2(VREFCMP).PB0 used as the negative terminal of the COMP1.When the voltage value of PB0 is greater than 1.65V, the LED is off, and when it is less than 1.65V, the LED is 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 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 function to wake up the MCU via the PA6 pin. After downloading the program and running, the LED remains on; After pressing the user button, the LED remains off, and the MCU enters the STOP mode; After pulling down the PA6 pin, the MCU wakes up and the LED light is toggling.

5 FLASH

5.1 FLASH_OptionByteWrite_Boot_LoadFlash

此样例演示了修改启动模式从 LoadFlash 启动，并设置 LoadFlash 的大小为 3k。

This sample demonstrates modifying the boot mode to boot from LoadFlash and setting the size of LoadFlash to 3k.

5.2 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 the change of the RESET pin to a normal GPIO by software.

5.3 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.

5.4 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_TwoBoards_MasterTx_SlaveRx_Polling

此样例演示了主机 I2C、从机 I2C 通过轮询方式进行通讯，当按下从机单板的用户按键，再按下主机单板的用户按键后，主机 I2C 向从机 I2C 发送"LED ON"数据。当主机 I2C 成功发送数据，从机 I2C 成功接收数据时，主机单板和从机单板 LED 灯分别亮起。

This sample demonstrates that I2C(as master and as slave) communicates with polling mode.Press the user key of the slave board first and then press the user key of the host board,the master I2C will send "LED ON" data to the slave I2C. When the master I2C successfully sends data and the slave I2C successfully receives data, the LED lights on the host board and slave board respectively.

7.2 I2C_TwoBoard_CommunicationMaster_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 that I2C communicates with interrupt mode, the host first sends 15byte data to the slave, and then receives 15byte data from the slave.After the host and slave successfully receive data, the LEDs on the host and slave board are in the state of "steady on".

7.3 I2C_TwoBoard_CommunicationMaster_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 that I2C communicates with polling mode, the host first sends 15byte data to the slave, and then receives 15byte data from the slave. After the host and slave successfully receive data, the LEDs on the host and slave board are in the state of "steady on".

7.4 I2C_TwoBoard_CommunicationSlave_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 that I2C communicates with interrupt mode, the host first sends 15byte data to the slave, and then receives 15byte data from the slave.After the host and slave successfully receive data, the LEDs on the host and slave board are in the state of "steady on".

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.

9.3 LPTIM_OnceMode_WakeUp_WFE

此样例演示了 LPTIM 单次模式事件唤醒 STOP 模式。

This sample demonstrates waking up from stop mode by LPTIM(single mode) event request.

9.4 LPTIM_OnceMode_WakeUp_WFI

此样例演示了 LPTIM 单次模式中断唤醒 STOP 模式。

This sample demonstrates waking up from stop mode by LPTIM(single mode) interrupt request.

10 PWR

10.1 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.

10.2 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.

10.3 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.

10.4 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.

11 RCC

11.1 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.

11.2 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 HSi waveforms.

11.3 RCC_LSE_Output

此样例演示了时钟输出功能，可输出 LSE 波形。

This sample demonstrates the clock output function, which can output LSE waveforms.

11.4 RCC_LSI_Output

此样例演示了时钟输出功能，可输出 LSI 波形。

This sample demonstrates the clock output function, which can output LSI waveforms.

12 SPI

12.1 SPI_TwoBoards_FullDuplexMaster_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.

12.2 SPI_TwoBoards_FullDuplexMaster_Polling_Init

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.

12.3 SPI_TwoBoards_FullDuplexSlave_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.

12.4 SPI_TwoBoards_FullDuplexSlave_Polling_Init

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.

13 TIM

13.1 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.

13.2 TIM1_InputCapture_Init

此样例演示了 TIM1 的输入捕获功能，配置 PA0 作为输入捕获引脚，PA0 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 PA0 as input capture pin,Whenever PA0 detects a falling edge it triggers a capture interrupt and toggle the LED in the capture interrupt callback function.

13.3 TIM1_InputCapture_XORCh1Ch2Ch3

此样例演示了 TIM1 的三通道异或输入捕获功能。配置 PA0、PA3、PA4 为通道 1、通道 2、通道 3 的输入引脚。每当有一个引脚电平变化时会触发捕获中断，并在中断处理中翻转 LED。

This sample demonstrates the 3 channels XOR input capture function of TIM1. Configure PA0 、PA3 、PA4 as CH1 、CH2 、CH3 input pin.Whenever pin of any of the three pin (PA0\PA3\PA4) detects a polarity change it triggers a capture interrupt and toggle the LED in the capture interrupt callback function.

13.4 TIM1_OC_Toggle

此样例演示了 TIM1 的输出比较模式。将捕获/比较通道 1（CH1）的输出映射到 PA5，开启捕获/比较通道 1（CH1）并设置为比较输出翻转模式

This sample demonstrates the output compare function of TIM1.CH1 map to PA5,and set CH1 as output compare channel and in toggle mode

13.5 TIM1_OC_Toggle_IT

此样例演示了 TIM1 的输出比较模式。将捕获/比较通道 1（CH1）的输出映射到 PA5，开启捕获/比较通道 1（CH1）并设置为比较输出翻转模式，并使能比较中断，在中断中翻转 LED。

This sample demonstrates the output compare function of TIM1.CH1 map to PA5,and set CH1 as output compare channel and in toggle mode.Enable compare interrupt and toggle LED in interrupt callback.

13.6 TIM1_PWM3CH_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 sample demonstrates how to use TIM1 PWM2 mode to output three 10Hz frequency PWM waveform with duty cycles of 25%, 50% and 75% separately

13.7 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.

14 USART

14.1 USART_HyperTerminal_AutoBaund_IT_Init

此样例演示了 USART 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 USART. 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.

14.2 USART_HyperTerminal_IndefiniteLengthData_IT_Init

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 9600，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200byte),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 9600, 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 200bytes) through the upper computer, such as 0x1~0xC. The MCU will send the received data to the upper computer again.

14.3 USART_HyperTerminal_IT_Init

此样例演示了 USART 的中断方式发送和接收数据，USART 配置为 9600，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 9600 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.

14.4 USART_HyperTerminal_Polling_Init

此样例演示了 USART 的轮询方式发送和接收数据，USART 配置为 9600，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 9600 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.