PY32F072 系列
32 位 ARM® Cortex®-M0+ 微控制器

LL 库样例手册

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。

1.2 ADC_MultiChannelSingleConversion_TriggerSW_DMA

此样例演示了 ADC 的 DMA 多通道传输功能，在 DMA 完成中断中打印多通道的电压值。

1.3 ADC_SingleConversion_TriggerSW_IT

此样例演示了 ADC 的中断功能，在 ADC 的中断中打印当前的电压值。

1.4 ADC_SingleConversion_TriggerTimer_Polling

此样例演示了 ADC 的 TIM 触发功能,TIM 每隔 1s 触发 ADC 进行采样,并通过串口打印出来。

1.5 ADC_Vrefint

此样例演示了 ADC 模块的 Vrefint 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。

2 COMP

2.1 COMP_CompareGpioVsVrefint_IT

此样例演示了比较器的中断功能，在中断中翻转 LED。

2.2 COMP_CompareGpioVsVrefint_Polling

此样例演示了比较器的轮询功能，当比较器的正端电压大于 Vrefint 时，LED 灯亮，小于 Vrefint 电压时，LED 灯灭。

2.3 COMP_CompareGpioVsVrefint_WakeUpFromStop

此样例演示了 COMP 比较器唤醒功能，PA0 作为比较器正端输入，VREFINT 作为比较器负端输入，上完电 LED 灯会常亮，用户点击按键，LED 灯灭，进入 stop 模式，通过调整 PA0 上的输入电压，产生中断唤醒 stop 模式。

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。

4 CTC

4.1 CTC_Autotrim_Init

此样例演示了 CTC 使用 LSE 做参考时钟自动校准 PLL48M 时钟的功能。

5 DAC

5.1 DAC_GenerateConstantSignal_TriggerSW

此样例演示了 DAC 的输出功能，通过 PA4 输出 1/2 供电电压的值。

6 DIV

6.1 DIV_Signed

此样例演示了硬件除法器计算有符号除法。

6.2 DIV_Unsigned

此样例演示了硬件除法器计算无符号除法。

7 DMA

7.1 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)。

8 EXTI

8.1 EXTI_Toggled_IT_Init

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。

8.2 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。

9 Flash

9.1 FLASH_OptionByteWrite_RST

此样例演示了通过软件方式将 RESET 引脚改为普通 GPIO。

9.2 FLASH_PageEraseAndWrite

此样例演示了 flash page 擦除和 page 写功能

9.3 FLASH_SectorEraseAndWrite

此样例演示了 flash sector 擦除和 page 写功能。

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。

10.2 GPIO_Toggle

此样例演示了 GPIO 输出模式，配置 LED 引脚(PB2)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。

10.3 GPIO_Toggle_Init

此样例演示了 GPIO 输出模式，配置 LED 引脚(PB2)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。

11 I2C

11.1 I2C_TwoBoard_CommunicationMaster_10BitAddr_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。

11.2 I2C_TwoBoard_CommunicationMaster_DMA_Init

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。

11.3 I2C_TwoBoard_CommunicationMaster_DMA_MEM_Init

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。

11.4 I2C_TwoBoard_CommunicationMaster_DualAddr_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。

11.5 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。

11.6 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。

11.7 I2C_TwoBoard_CommunicationSlave_10BitAddr_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。

11.8 I2C_TwoBoard_CommunicationSlave_DMA_Init

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。

11.9 I2C_TwoBoard_CommunicationSlave_DualAddr_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。

11.10 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。

12 I2S

12.1 I2S_TwoBoard_CommunicationMaster_DMA

此样例是对I2S主机与I2S从机以DMA方式进行通信的演示，I2S主机先向I2S从机发送数据0x1~0x10，I2S 从机接收到数据后，再向 I2S 主机回发数据 0x0x1~0x10，当 I2S 主机、I2S 从机成功接收数据时，小灯处于常亮状态，否则小灯处于闪烁状态。

12.2 I2S_TwoBoard_CommunicationMaster_IT

此样例是对I2S主机与I2S从机以中断方式进行通信的演示，I2S主机先向I2S从机发送数据0x1~0x10，I2S 从机接收到数据后，再向 I2S 主机回发数据 0x0x1~0x10，当 I2S 主机、I2S 从机成功接收数据时，小灯处于常亮状态，否则小灯处于闪烁状态。

12.3 I2S_TwoBoard_CommunicationMaster_Polling

此样例是对 I2S 主机与 I2S 从机以 polling 方式进行通信的演示，I2S 主机先向 I2S 从机发送数据 0x1~0x10，I2S 从机接收到数据后，再向 I2S 主机回发数据 0x0x1~0x10，当 I2S 主机、I2S 从机成功接收数据时，小灯处于常亮状态，否则小灯处于闪烁状态。

12.4 I2S_TwoBoard_CommunicationSlave_DMA

此样例是对I2S主机与I2S从机以DMA方式进行通信的演示，I2S主机先向I2S从机发送数据0x1~0x10，I2S 从机接收到数据后，再向 I2S 主机回发数据 0x0x1~0x10，当 I2S 主机、I2S 从机成功接收数据时，小灯处于常亮状态，否则小灯处于闪烁状态。

12.5 I2S_TwoBoard_CommunicationSlave_IT

此样例是对I2S主机与I2S从机以中断方式进行通信的演示，I2S主机先向I2S从机发送数据0x1~0x10，I2S 从机接收到数据后，再向 I2S 主机回发数据 0x0x1~0x10，当 I2S 主机、I2S 从机成功接收数据时，小灯处于常亮状态，否则小灯处于闪烁状态。

12.6 I2S_TwoBoard_CommunicationSlave_Polling

此样例是对 I2S 主机与 I2S 从机以 polling 方式进行通信的演示，I2S 主机先向 I2S 从机发送数据 0x1~0x10，I2S 从机接收到数据后，再向 I2S 主机回发数据 0x0x1~0x10，当 I2S 主机、I2S 从机成功接收数据时，小灯处于常亮状态，否则小灯处于闪烁状态。

13 IWDG

13.1 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）。

14 LCD

14.1 LCD_Display_Init

此样例是对单色无源液晶显示器(LCD)的演示，将偏置产生电路配置为内部电阻分压，使 LCD 全显,显示“88:88”字样。

15 LPTIM

15.1 LPTIM_ContinuousMode_WakeUp

此样例演示了 LPTIM 连续模式事件唤醒 STOP 模式。

16 OPA

16.1 OPA_VoltageFollow

此样例演示了 OPA 的电压跟随功能,PA7 为正端输入,PA5 为负端输入,PA6 为输出,PA6 会输出和 PA7 相同的电压值。

17 PWR

17.1 PWR_PVD

此样例演示了 PVD 电压检测功能,样例中配置 PB07 引脚的电压与 VREF(1.2v)进行比较,当 PB07 引脚的电压高于 VREF 时,LED 灯灭,当低于 VREF 时,LED 灯亮。

17.2 PWR_SLEEP_WFE

此样例演示了在 sleep 模式下，使用 GPIO 事件唤醒。

17.3 PWR_SLEEP_WFI

此样例演示了在 sleep 模式下，使用 GPIO 中断唤醒。

17.4 PWR_STOP_WFE

此样例演示了在 stop 模式下，使用 GPIO 事件唤醒。

17.5 PWR_STOP_WFI

此样例演示了在 stop 模式下，使用 GPIO 中断唤醒。

18 RCC

18.1 RCC_HSE_OUTPUT

此样例演示了时钟输出功能，可输出 HSE 波形。

18.2 RCC_HSI_OUTPUT

此样例演示了时钟输出功能，可输出 HSI 波形。

18.3 RCC_LSE_OUTPUT

此样例演示了时钟输出功能，可输出 LSE 波形。

18.4 RCC_LSI_OUTPUT

此样例演示了时钟输出功能，可输出 LSI 波形。

18.5 RCC_PLL_OUTPUT

此样例演示了时钟输出功能，可输出 PLL 波形（32MHz）。

18.6 RCC_Sysclock_Switch

此样例演示了时钟切换，由 LSI（32.768KHz）切换至 HSE（24MHz）。

19 RTC

19.1 RTC_Alarm_Init

此样例演示 RTC 的闹钟中断功能，在数组 aShowTime 中显示当前时间，在数组 aShowDate 中显示当前日期，当达到闹钟值时，LED 灯会亮起。

19.2 RTC_WakeUpAlarm_Init

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

19.3 RTC_WakeUpSecond_Init

此样例演示通过 RTC 秒中断从 STOP 模式下唤醒，唤醒后，小灯处于闪烁状态；否则处于熄灭状态。

20 SPI

20.1 SPI_TwoBoards_FullDuplexMaster_DMA_Init

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,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。

20.2 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,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。

20.3 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,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。

20.4 SPI_TwoBoards_FullDuplexSlave_DMA_Init

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,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。

20.5 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,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。

20.6 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,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。

21 TIM

21.1 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。

21.2 TIM1_ComplementarySignals_Init

此样例演示了使用 TIM1 输出三路频率为 10Hz 占空比分别为 25%、50%、75%的 PWM 波形以及他们的互补信号。

21.3 TIM1_DmaBurst_Init

此样例演示了 TIM1 的 DMA Burst 传输，配置 TIM1 为 PWM 模式，更新中断触发 DMA 传输请求。每次产生更新中断时将 TIM1DataBuff[]中的值按顺序写入 RCR 和 CCR1 寄存器，改变 PWM 脉冲的占空比和该占空比的脉冲数量。

21.4 TIM1_EncoderTI2AndTI1_Init

此样例演示了 TIM1 的编码器接口模式。TIM1 配置为编码器接口模式 3，PA8 和 PA9 配置为通道 1 和通道 2,当 PA8 输入信号的上升沿在前，PA9 输入信号上升沿在后时 TIM1 向上计数，反之向下计数。开启通道 1 和通道 2 的捕获中断，在中断中打印当前 CNT 值。

21.5 TIM1_InputCapture

此样例演示了 TIM1 的输入捕获功能。配置 PA8 为通道 1 的输入引脚,每当引脚电平出现上升沿时会触发捕获中断,并在中断处理中翻转 LED。

21.6 TIM1_InputCapture_XORCh1Ch2Ch3

此样例演示了 TIM1 的三通道异或输入捕获功能。配置 PA8、PA9、PA10 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。

21.7 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 PA8，开启捕获/比较通道

1(CH1)并设置为比较输出翻转模式

21.8 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 PA8，通道 2 为输入模式，映射到 PA9。当 PA9 上检测到一个上升沿时，PA8 延迟 20ms 后产生一个宽度为 80ms 的脉冲。

21.9 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。

21.10 TIM1_TIM3_Cascade

此样例演示了 TIM1 和 TIM3 级联成 32 位计数器，TIM3 做主机，TIM3 的溢出信号作为 TIM1 的输入时钟。TIM3 每 1ms 计数一次，计数 1000 次后产生溢出，TIM1 计数一次。

21.11 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。

21.12 TIM1_Update_DMA_Init

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 更新周期变化,TIM1 第一次溢出后 LED 会翻转,此次翻转时间间隔为 1000ms,DMA 将数据搬运到 TIM1_ARR,第二次 LED 翻转间隔为 900ms,以此类推,最后 LED 翻转间隔为 100msDMA 搬运结束,LED 保持 100ms 的翻转间隔闪烁。

22 USART

22.1 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。

22.2 USART_HyperTerminal_DMA_Init

此样例演示了通过 DMA 收发数据的功能，复位 MCU 并重新运行，PC 端收到字符串“UART Test”；PC 端发送 12 个字符，MCU 会反馈同样的 12 个字符给 PC 端。

22.3 USART_HyperTerminal_IT_Init

此样例演示了通过 USART 中断收发数据的功能，复位 MCU 并重新运行，PC 端收到字符串：UART Test；PC 端发送 12 个字符，MCU 会反馈同样的 12 个字符给 PC 端。

22.4 USART_HyperTerminal_Polling_Init

此样例演示了通过 USART 轮询收发数据的功能，MCU 复位后会向 PC 端发送"UART Test",PC 端发送 12 个字符，MCU 会反馈同样的 12 个字符给 PC 端。

23 UTILS

23.1 UTILS_ConfiggureSystemClock

本样例主要演示如何配置 SYSCLK(系统时钟), HCLK(AHB 时钟), PCLK(APB 时钟)。通过 MCO 输出系统时钟的 8 分频 9MHz。

24 WWDG

24.1 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。

24.2 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。