



PY32M010 系列

32 位 ARM® Cortex®-M0+ 微控制器

LL 库样例手册

PY32M010 系列

32 位 ARM® Cortex®-M0+ 微控制器

LL 库样例手册

1 ADC

1.1 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。

1.2 ADC_SingleConversion_TriggerTimer_IT

此样例演示了 ADC 的 TIM 触发和中断的功能。

1.3 ADC_SingleConversion_TriggerTimer_Polling

此样例演示了 ADC 的 TIM 触发和轮询的功能。

1.4 ADC_Temperature_Init

此样例演示了 ADC 的 Tempsensor 功能。

1.5 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。

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。

2.2 COMP_CompareGpioVs1_2VCC_WakeupFromStop

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 stop 模式，通过调整 PA04 上的输入电压，产生中断唤醒 stop 模式。

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。

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 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。

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。

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。

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。

5.3 FLASH_PageEraseAndWrite

此样例演示了 flash page 擦除和 page 写功能。

5.4 FLASH_SectorEraseAndWrite

此样例演示了 flash sector 擦除和 page 写功能。

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。

6.2 GPIO_Toggle

此样例演示了 GPIO 输出模式，配置 LED 引脚(PA1)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。

6.3 GPIO_Toggle_Init

此样例演示了 GPIO 输出模式，配置 LED 引脚(PA1)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。

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。

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。

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。

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。

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）。

9 LPTIM

9.1 LPTIM_ContinuousMode_WakeUp

此样例演示了 LPTIM 连续模式事件唤醒 STOP 模式。

9.2 LPTIM_OnceMode_WakeUp

此样例演示了 LPTIM 单次模式事件唤醒 STOP 模式。

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。

10.2 PWR_SLEEP_WFI

此样例演示了在 sleep 模式下，使用 GPIO 中断唤醒。

10.3 PWR_STOP_WFE

此样例演示了在 stop 模式下，使用 GPIO 事件唤醒。

10.4 PWR_STOP_WFI

此样例演示了在 stop 模式下，使用 GPIO 中断唤醒。

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。

11.2 RCC_HSI_Output

此样例演示了时钟输出功能，可输出 HSI 波形。

11.3 RCC_LSE_Output

此样例演示了时钟输出功能，可输出 LSE 波形。

11.4 RCC_LSI_Output

此样例演示了时钟输出功能，可输出 LSI 波形。

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,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。

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,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。

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,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。

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,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。

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。

13.2 TIM1_InputCapture_Init

此样例演示了 TIM1 的输入捕获功能，配置 PA1 作为输入捕获引脚，PA1 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。

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。

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

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。

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。

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。

14 USART

14.1 USART_HyperTerminal_AutoBaud_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。

14.2 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。

14.3 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。