PY32T020 series

32-bit ARM® Cortex®-M0+ microcontroller

LL Library Sample Manual

1 ADC

1.1 ADC_MultichannelSwitch

此样例演示了 ADC 的多通道切换转换。

This example demonstrates the channel switching for an 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_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，PA2 作为比较器负端输入，1/2VCCA 作为正端输入，通过调整 PA2 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 PA2 as the negative comparator input and 1/2VCCA as the positive input. Adjust the input voltage on PA2, 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_Window

此样例演示了 COMP 比较器的 window 功能，比较器 1 的 Plus 端用比较器 2 的 IO4(1/2VCCA)作为输入，PB1 作为比较器 1 负端输入，当 PB1 的电压值大于 1.65V 时,LED 灯灭，小于 1.65V 时,LED 灯亮。PA2 作为比较器 2 负端输入，当 PA2 的电压值大于 1.65V 时,PA4 拉低，小于 1.65V 时,PA4 拉高

This example demonstrates the window function of the COMP comparator. The Plus end of comparator 1 uses the IO4 (1/2VCCA) of comparator 2 as the input, and PB1 as the negative end input. When the voltage value of PB1 is greater than 1.65V, the LED light turns off, and when it is less than 1.65V, the LED light turns on. PA2 is input as the negative end of comparator 2,when the voltage value of PA2 is greater than 1.65V,the PA4 pull down, and when it is less than 1.65V, the PA4 pull up

2.3 COMP_CompareGpioVs32_64VCC_WakeupFromStop

此样例演示了 COMP 比较器唤醒功能，PA0 作为比较器负端输入，32/64VCC 作为比较器正端输入，进入 stop 模式后，通过调整 PA0 上的输入电压，产生中断唤醒 stop 模式。

This example demonstrates the COMP comparator wake-up function, with PA0 as the negative input and 32/64VCC as the positive input of the comparator. After entering stop mode, the interrupt wake-up stop mode is generated by adjusting the input voltage on PA0.

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 checksum function. It performs a checksum on the data in an array and compares the calculated checksum with the expected checksum. If they are equal, the LED turns on; otherwise, the LED turns off.

4 EXTI

4.1 EXTI_ToggleLed_IT_Init

此样例演示了 GPIO 外部中断功能，PA15 引脚上的每一个上升沿都会产生中断，中断函数中 LED 灯会翻转一次。

This example demonstrates the GPIO external interrupt function, each rising edge on the PA15 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 PA5 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 PA5 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 PA5 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 PA5 pin, the MCU wakes up and the LED light is toggling.

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 example demonstrates changing the RESET pin to regular GPIO through software.

5.2 FLASH_PageEraseAndWrite

此样例演示了 flash page 擦除和 page 写功能。

This example demonstrates the flash page erase and page write functions.

5.3 FLASH_SectorEraseAndWrite

此样例演示了 flash sector 擦除和 page 写功能。

This example 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_CommunicationMaster_IT_Init

==
样 例 使 用 说 明   Sample
Description===
============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".

7.2 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".

7.3 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".

7.4 I2C_TwoBoard_MasterTxIndefiniteLengthData_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.

7.5 I2C_TwoBoard_SlaveRxIndefiniteLengthData_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 PWR

9.1 PWR_DEEPSTOP_WFE

此样例演示了在 deepstop 模式下，使用 GPIO 事件唤醒。

This sample demonstrates using GPIO event to wake up the MCU from deepstop mode.

9.2 PWR_DEEPSTOP_WFI

此样例演示了在 deepstop 模式下，使用 GPIO 中断唤醒。

This sample demonstrates using GPIO interrupt to wake up the MCU from deepstop mode.

9.3 PWR_HIBERNATE_WFE

此样例演示了在 hibernate 模式下，使用 GPIO 事件唤醒。

This sample demonstrates using GPIO event to wake up the MCU from hibernate mode.

9.4 PWR_HIBERNATE_WFI

此样例演示了在 hibernate 模式下，使用 GPIO 中断唤醒。

This sample demonstrates using GPIO interrupt to wake up the MCU from hibernate mode.

9.5 PWR_SLEEP_WFE

此样例演示了在 sleep 模式下，使用 GPIO 事件唤醒。

This sample demonstrates using GPIO event to wake up the MCU from sleep mode.

9.6 PWR_SLEEP_WFI

此样例演示了在 sleep 模式下，使用 GPIO 中断唤醒。

This sample demonstrates using GPIO interrupt to wake up the MCU from sleep mode.

9.7 PWR_STOP_WFE

此样例演示了在 stop 模式下，使用 GPIO 事件唤醒。

This sample demonstrates using GPIO event to wake up the MCU from stop mode.

9.8 PWR_STOP_WFI

此样例演示了在 stop 模式下，使用 GPIO 中断唤醒。

This sample demonstrates using GPIO interrupt to wake up the MCU from stop mode.

10 RCC

10.1 RCC_HSE_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 the HSE waveform.

10.2 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.

10.3 RCC_LSE_Output

此样例配置系统时钟为 LSE，并通过 MCO（PA08）引脚输出。

This sample configures the system clock to LSE and outputs it through the MCO (PA08) pin.

10.4 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.

10.5 RCC_Sysclock_Switch

此样例演示了时钟切换，由 LSI 切换至 HSE。

This sample demonstrates clock switching from LSI to HSE.

11 RTC

11.1 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。

This sample demonstrates the alarm interrupt function of the RTC. It displays　the current time in the array aShowTime and the current date in the array　aShowDate. When the alarm value is reached, the LED will light up.

11.2 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。

This sample demonstrates waking up the MCU from STOP mode approximately　every 1 second using RTC alarm interrupt. Each time the MCU wakes up,　the LED will toggle. The interval between LED toggling is also approximately　1 second.

11.3 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。

This sample demonstrates waking up the MCU from STOP mode using RTC second　interrupt. After waking up, the LED will be in a blinking state. Otherwise,　it will be turned off.

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

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 example demonstrates the use of TIM1 to generate a "six step PWM signal", which triggers commutation in the SysTick interrupt every 1ms to achieve commutation of a brushless motor.

13.2 TIM1_ComplementarySignals

此样例实现了定时器的互补输出功能，三组互补共六路 pwm 输出。

This sample demonstrates complementary output function of the timer,Three sets of complementary outputs total six pwm outputs.

13.3 TIM1_ExternalClockMode1_Init

此样例演示了 TIM1 的外部时钟模式 1 的功能，选择 ETR(PA12)引脚作为外部时钟输入源，并使能更新中断，在中断中翻转 LED 灯。

This sample demonstrates the functionality of TIM1 in external clock mode 1. It selects ETR (PA12) pin as the external clock input source and enables the update interrupt. In the interrupt, the LED light is toggled.

13.4 TIM1_ExternalClockMode1_TI1F_Init

此样例演示了 TIM1 的外部时钟模式 1 功能，选择 TI1FP(PA3)引脚作为外部时钟输入源，并使能更新中断，在中断中翻转 LED 灯

This sample demonstrates the external clock mode 1 function of TIM1, selects the TI1FP(PA3) pin as the external clock input source, and enables the update interrupt and toggle the LED light in the interrupt

13.5 TIM1_ExternalClockMode2_Init

此样例演示了 TIM1 的外部时钟模式 2 功能，选择 ETR(PA12)引脚作为外部时钟输入源，并使能更新中断，在中断中翻转 LED 灯。

This sample demonstrates the external clock mode 2 function of TIM1, selects the ETR(PA12) pin as the external clock input source, and enables the update interrupt and toggle the LED light in the interrupt

13.6 TIM1_InputCapture_Init

此样例演示了 TIM1 的输入捕获功能，配置 PA3 作为输入捕获引脚，PA3 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 example demonstrates the input capture function of TIM1, where PA3 is configured as the input capture pin. Every time PA3 detects a falling edge, it triggers a capture interrupt and flips the LED light in the capture interrupt callback function.

13.7 TIM1_InputCapture_XORCh1Ch2Ch3

此样例演示了 TIM1 的三通道异或输入捕获功能。配置 PA3、PA5、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 example demonstrates the three channel XOR input capture function of TIM1. Configure PA3, PA5, and PA4 as input pins for channels 1, 2, and 3. Whenever a pin level changes, a capture interrupt is triggered and the LED is flipped during interrupt processing.

13.8 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 PA3，开启捕获/比较通道 1（CH1）并设置为比较输出翻转模式。

This example demonstrates the output comparison mode of TIM1. Map the output of capture/compare channel 1 (CH1) to PA3, turn on capture/compare channel 1 (CH1), and set it to compare output flipping mode.

13.9 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 PA3，开启捕获/比较通道 1（CH1）并设置为比较输出翻转模式，同时开启捕获/比较中断，每次计数值与比较值匹配时翻转输出电平（PA4），在捕获/比较中断处理中翻转 LED 灯。

This sample demonstrates the output compare mode of TIM1. The output of capture/compare channel 1 (CH1) is mapped to pin PA3. Capture/compare channel 1 (CH1) is enabled and set to compare output toggle mode. Capture/compare interrupt is also enabled. Whenever the counter value matches the compare value, the output (PA4) level will toggle. In the interrupt handler of capture/compare, the LED will also toggle.

13.10 TIM1_OnePulseOutput

此样例演示了 TIM1 的单脉冲模式，CH2(PA05)引脚上的上升沿，触发计数器开始计数，当计数值与

CCR1 匹配时，CH1(PA03)输出高电平，直到计数器溢出，CH1 再次输出低电平，计数器溢出后，定时器停止工作，本例程脉冲宽度计算 (TIM1_ARR-TI1_CCR1)/CLK= (65535-16383) /24000000=2.048ms

This sample demonstrates the one pulse mode of TIM1.The rising edge on the CH2(PA05) pin triggers the counter to start counting. when the count value matches CCR1,CH1(PA03) outputs a high level. When the counter overflows ,CH1 outputs the low level again. After the counter overflows, the timer stops working. This example pulse width calculation (TIM1_ARR-TI1_CCR1)/CLK= (65,535-16383)/ 24,000,000 =2.048ms

13.11 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

13.12 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 UART

14.1 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

14.2 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