



DPM32M08 series

32-bit ARM® Cortex®-M0

Development board example V0.0.3



目录

32-bit ARM® Cortex®-M0	1
1. ACMP	5
1.1. ACMP_BEMFR	5
1.2. ACMP_Interrupt	5
1.3. ACMP_Compare	5
2. ADC	6
2.1. ADC_DMASingleContinue	6
2.2. ADC_InjectSample	6
2.3. ADC_OverSampleInterrupt	6
2.4. ADC_ScanSample	7
2.5. ADC_SoftTriggerSingle	7
2.6. ADC_Temperature	7
2.7. ADC_TemperatureNTC	8
2.8. ADC_TimerTrigger	8
2.9. ADC_WatchdogSample	8
3. CCT	9
3.1. CCT_DMA	9
3.2. CCT_InputCapture	9
3.3. CCT_OutputCompare	9
4. CRC	10
4.1. CRC_CalculateModbus	10
5. DAC	10
5.1. DAC_Output	10
6. DBG	10
6.1. DBG_DebugEPWM	10
7. DES	11
7.1. DES_GetInfo	11
8. DMA	11
8.1. DMA_Software	11
9. DSP	11
9.1. DPS_WDGReset	11
9.2. DPS_WriteReadCalculate	12
9.3. DSP_Debug	12
9.4. DSP_DMA	12
9.5. DSP_Interrupt	13
10. EPWM	13
10.1. EPWM_DeadZoneInsertion	13
10.2. EPWM_DMA	13
10.3. EPWM_EmergencyStop	14
10.4. EPWM_ForceOutput	14
10.5. EPWM_OutputInterrupt	14



11. FLASH	15
11.1. Flash_EraseWriteRead	15
11.2. Flash_ReadWaitAndPrefetch	15
11.3. Flash_UserOption	15
12. GPIO	16
12.1. GPIO_KeyIrqMode	16
12.2. GPIO_KeyPollingMode	16
12.3. GPIO_ToggleLed	16
13. I2C	17
13.1. I2C_EEPROM	17
13.2. I2C_MasterBlock	17
13.3. I2C_MasterDMA	17
13.4. I2C_MasterInterrupt	17
13.5. I2C_SlaveBlock	17
13.6. I2C_SlaveDMA	18
13.7. I2C_SlaveInterrupt	18
14. LPTIM	18
14.1. LPTIM_CounterInterrupt	18
15. PGA	18
15.1. PGA_DifferentialInput	18
16. PVD	19
16.1. PVD_Interrupt	19
16.2. PVD_Reset	19
17. RCC	19
17.1. RCC_MCOoutput	19
17.2. RCC_SysclkConfig	20
18. LowPower	20
18.1. RCC_LowPowerDeepSleep_UART	20
18.2. RCC_LowPowerStop_LPTIM	20
18.3. RCC_LowPowerStop_GPIO	21
19. SPI	21
19.1. SPI_MasterBlock	21
19.2. SPI_MasterDMA	21
19.3. SPI_MasterInterrupt	22
19.4. SPI_SlaveBlock	22
19.5. SPI_SlaveDMA	22
19.6. SPI_SlaveInterrupt	23
19.7. SPI_w25qxx	23
20. TIM	23
20.1. TIM_CuontInterrupt	23
20.2. TIM_DMA	24
21. UART	24
21.1. UART_232_Interrup	24
21.2. UART_485_MasterInterrupt	24



21.3. UART_Block.....	25
21.4. UART_DMA.....	25
21.5. UART_HalfDuplex.....	25
21.6. UART_Interrupt.....	26
21.7. UART_Printf.....	26
22. WDG.....	26
22.1. WDG_InterruptFeedDog.....	26



1. ACMP

1.1. ACMP_BEMFR

此样例演示了 ACMP 的反电动势检测功能，初始化 ACMP0、1、2 三个通道，P 端接引脚，N 端接内部电阻，使能反电动势功能后，可测量出通道之间的阻值，LED1 每隔 1s 翻转一次。

This example demonstrates the back electromotive force detection function of ACMP, initializing ACMP channels 0, 1, and 2, P-terminal pin, N-terminal internal electrical connection. After enabling the anti electromotive force function, the resistance value between channels can be measured, LED1 flips every 1 second.

1.2. ACMP_Interrupt

该样例演示了 ACMP 中断功能。P 端：PD12，N 端：DAC 1.5V，旋转开发板上的滑动电阻器 R11，当比较结果产生上升沿或下降沿中断时，LED2 翻转一次。LED1 每个 1s 翻转一次。

This example demonstrates the ACMP interrupt function. P-end: PD12, N-end: DAC 1.5V, sliding resistor R11 on the rotating development board, when compared to When there is an interruption in the rising or falling edge of the result, LED2 state flip. Flip LED1 once every 1 second.

1.3. ACMP_Compare

该样例演示了 ACMP 比较功能。P 端：PD12，N 端：PD13，旋转开发板上的滑动电阻器 R11 和 R24，当比较结果为 1 时，LED2 亮，结果为 0 时，LED2 灭，LED1 每隔 1s 翻转一次。

This example demonstrates the ACMP comparison function. P-end: PD12, N-end: PD13, sliding resistors R11 and R24 on the rotating development board, when compared. When the result is 1, When LED2 is on and the result is 0, LED2 goes out, LED1 flips every 1

second.

2. ADC

2.1. ADC_DMA SingleContinue

该样例演示了单通道连续采样，DMA 自动传输数据。旋转开发板上的滑动电阻器 R11 调节端口输入电压，串口打印采样数据。程序正常运行中，LED9 白灯闪烁。

This example demonstrates single channel continuous sampling and DMA automatic data transmission. Rotate the sliding resistor R11 on the development board to adjust the input voltage of the port, and print the sampling data through the serial port. During normal program operation, LED9 white light flashes.

2.2. ADC_InjectSample

该样例演示了 ADC 通道插队采样功能，通道 1-13 序列采样，通道 0 插队采样，触发插队后，从插队结果寄存器读取采样值，旋转开发板上的滑动电阻器 R11 调节输入电压，串口实时打印采样数据，LED1 每隔 1s 翻转一次。

This example demonstrates the ADC channel queueing sampling function, with sequence sampling from channels 1-13 and queueing sampling from channel 0. After triggering queueing, the queueing process starts from The result register reads the sampling value, rotates the sliding resistor R11 on the development board to adjust the input voltage, and the serial port prints the sampling data in real time, LED1 flips every 1 second.

2.3. ADC_OverSampleInterrupt

该样例演示了 ADC 过采样功能，通道 0 配置 4 倍过采样，中断获取采样数据，通过串口将实时电压值打印出来，旋转开发板上的滑动电阻器 R11 调节电压，LED1 每隔 1s 翻转一次。



This example demonstrates the ADC oversampling function, where channel 0 is configured with 4x oversampling, interrupt acquisition of sampling data, and real-time voltage values are transmitted through a serial port. Print it out and rotate the sliding resistor R11 on the development board to adjust the voltage, LED1 flips every 1 second.

2.4. ADC_ScanSample

该样例演示了序列扫描采样，扫描全部通道，旋转开发板上的滑动电阻器 R11 调节 Chane10 输入电压，旋转开发板上的滑动电阻器 R24 调节 Chane11 输入电压，串口实时打印数据，LED1 每隔 1s 翻转一次。

This example demonstrates sequence scanning sampling, scanning all channels, rotating the sliding resistor R11 on the development board to adjust the input voltage of Chane10, Rotate the sliding resistor R24 on the development board to adjust the input voltage of Chane11, and print data in real-time through the serial port,, LED1 flips every 1 second.

2.5. ADC_SoftTriggerSingle

该样例演示了 ADC 软件触发单通道单次采样，每隔 1s 触发一次采样，采样结果打印到串口，旋转开发板上的滑动电阻器 R11 调节端口输入电压,串口打印采样数据，LED1 每隔 1s 翻转一次。

This example demonstrates how ADC software triggers a single channel single sampling, which occurs every 1 second. The sampling results are printed to the serial port and rotated for development. The sliding resistor R11 on the board adjusts the input voltage of the port, and the serial port prints sampling data, LED1 flips every 1 second.

2.6. ADC_Temperature

该样例演示了 ADC 采样芯片内部温度传感器的值，通过串口将实时电压值打印出来，LED1 每隔 1s 翻转一次。

This example demonstrates the value of the temperature sensor inside the ADC sampling chip, which prints out the real-time voltage value through a serial port, LED1 every 1 second flip once.

2.7. ADC_TemperatureNTC

该样例演示了 ADC 采样外部 NTC 温度传感器，通过串口将实时温度值打印出来，LED1 每隔 1s 翻转一次。

This example demonstrates the ADC sampling of an external NTC temperature sensor, which prints real-time temperature values through a serial port, LED1 flips every 1 second.

2.8. ADC_TimerTrigger

该样例演示了定时器触发 ADC 采样，Time0 定时 1ms，ADC0-Channel0 采样一次。旋转开发板上的滑动电阻器 R11 调节输入电压，串口实时打印数据。程序正常运行中，LED9 白灯闪烁。

This example demonstrates how a timer triggers ADC sampling, Time0 timed for 1 second, Sample once for ADC0 Chanle0. Sliding on the rotating development board Resistor R11 adjusts the input voltage of the port, and the serial port prints sampling data, LED1 flips every 1 second.

2.9. ADC_WatchdogSample

该样例演示了该样例演示了 ADC 的模拟看门狗功能，当开启看门狗的通道的电压值不在设定的上下限之间，会进入 ADC 看门狗中断，LED2 亮。旋转开发板上的滑动电阻器 R11 调节输入电压，串口实时打印数据，LED1 每隔 1s 翻转一次。



This example demonstrates the analog watchdog function of ADC, where the voltage value of the channel that opens the watchdog is not within the set upper and lower limits. Between them, the ADC watchdog interrupt will enter, LED2 is on. Rotate the sliding resistor R11 on the development board to adjust the input voltage and activate the serial port in real-time. Printing data, LED1 flips every 1 second.

3. CCT

3.1. CCT_DMA

该样例演示了 CCT 计数器溢出触发 DMA 传输功能，复位串口打印传输前的数据，等待 DMA 触发后，打印传输后的数据，LED1 每隔 1s 翻转一次。

This example demonstrates the CCT counter overflow triggering DMA transmission function, resetting the serial port to print data before transmission, and waiting for DMA to trigger, Print the transmitted data, LED1 flips every 1 second.

3.2. CCT_InputCapture

该样例演示了 CCT 的输入捕获功能，PA3 为输入捕获引脚，捕获上升沿，计算两个上升沿之间的宽度并打印到串口。

This example demonstrates the input capture function of CCT, PA3 is the input capture pin, capturing the rising and falling edges, and calculating the distance between two rising edges Width and print to serial port.

3.3. CCT_OutputCompare

该样例演示了 CCT 的输出比较和中断功能，PA2,PA3 输出比较波形，通道 0 产生比较中断翻转 LED2，通道 2 产生比较中断翻转 LED3，LED1 每隔 1s 翻转一次。



This example demonstrates the output comparison and interrupt function of CCT, PA2 and PA3 output comparison waveforms, channel 0 generates comparison interrupt flip LED2, Channel 2 generates a comparison interrupt flip LED3, LED1 flips every 1 second.

4. CRC

4.1. CRC_CalculateModbus

该样例演示了硬件 CRC 功能，CRC 配置为 ModBus 模式，计算 0x1234 的 CRC 值，与实际值相比较，如果计算出错则 LED2 亮，LED1 每隔 1s 翻转一次。

This example demonstrates the hardware CRC function, CRC configured in ModBus mode, calculate the CRC value of 0x1234 and compare it with the actual value. If there is a calculation error, LED2 will light up, LED1 flips every 1 second.

5. DAC

5.1. DAC_Output

该样例演示了 DAC 的输出功能，每隔 1ms DAC 输出 Code 值加一，可在 PD11 引脚读出电压变化，LED1 每隔 1s 翻转一次。

This example demonstrates the output function of DAC, where the DAC output Code value is increased by one every 1ms, and the voltage change can be read at the PD11 pin, LED1 Flip every 1 second.

6. DBG

6.1. DBG_DebugEPWM

该样例演示了 EPWM 在调试模式下正常输出功能，进入调试模式，配置 EPWM 在调试时正常输出，将程序暂停运行，可观察到 EPWM 正常输出，配置 PWM 在调试时不输出，将程序暂停运行，可观察到 EPWM 不输出，LED1 每隔 1s 翻转一次。

This example demonstrates the normal output function of EPWM in debugging mode, entering debugging mode, configuring EPWM to output normally during debugging, and completing the process. Pause the program and observe the normal output of EPWM. Configure PWM to not output during debugging and pause the program to observe EPWM No output, LED1 flips every 1 second.

7. DES

7.1. DES_GetInfo

该样例演示了 DES 的读取芯片基本信息功能，复位打印芯片进本信息，LED1 每隔 1s 翻转一次。

This example demonstrates the basic chip information reading function of DES, resetting and printing chip input information, LED1 flips every 1 second.

8. DMA

8.1. DMA_Software

该样例演示了 DMA 软件触发传输和中断功能，复位后串口打印传输前的数据，等待传输完成后，打印传输后的数据，LED1 每隔 1s 翻转一次。

This example demonstrates the DMA software triggering the transmission function. After resetting, the serial port prints the data before transmission, and after waiting for the transmission to be completed, prints the transmission Subsequent data, LED1 flips every 1 second.

9. DSP

9.1. DPS_WDGReset

该样例演示了 DSP 的看门狗触发复位功能，当 DPS 计算时间超出配置的复位时间时，触发复位，串口打印复位日志，LED1 每隔 1s 翻转一次。

This example demonstrates the watchdog trigger reset function of DSP. When the DPS calculation time exceeds the configured reset time, a reset is triggered, and the serial port Print reset logs, LED1 flips every 1 second.

9.2. DPS_WriteReadCalculate

该样例演示了 DSP 的计算功能，通过上位机生成 DSP 计算代码，软件写入 DSP 对应寄存器，软件触发 DSP 计算，等待计算完成，打印计算结果，LED1 每隔 1s 翻转一次。

This example demonstrates the calculation function of DSP, which generates DSP calculation code through the upper computer, writes the software into the corresponding register of DSP, and triggers the software DSP calculation, wait for the calculation to be completed, print the calculation result, LED1 flips every 1 second.

9.3. DSP_Debug

该样例演示了 DSP 的断点调试功能，配置 DSP 断点后，DSP 计算到对应位置会停止，此时可读取对应寄存器的值进行调试，清除断点后，需软件再次触发运行，串口输出日志，LED1 每隔 1s 翻转一次。

This example demonstrates the breakpoint debugging function of DSP. After configuring DSP breakpoints, The DSP will stop when it reaches the corresponding position, and at this time, the corresponding data can be read Debugging the value of the register, clearing the breakpoint, the software needs to trigger the run again, and the serial port outputs a log, LED1 flips every 1 second.

9.4. DSP_DMA

该样例演示了 DSP 的 DMA 功能，DMA 传输完成触发 DSP 计算，DSP 计算完成触发 DMA 传



输，串口打印最终计算结果，LED1 每隔 1s 翻转一次。

This example demonstrates the DMA function of DSP, DMA transmission completion triggers DSP calculation, DSP calculation completion triggers DMA transmission, serial port printing final Calculation results, LED1 flips every 1 second.

9.5. DSP_Interrupt

该样例演示了 DSP 的中断功能，开启 DSP 计算完成中断，软件触发计算，等待计算完成中断，串口打印计算结果，LED1 每隔 1s 翻转一次。

This example demonstrates the interrupt function of DSP, enabling DSP calculation completion interrupt, software triggering calculation, waiting for calculation completion interrupt, serial port printing Calculation results, LED1 flips every 1 second.

10. EPWM

10.1. EPWM_DeadZoneInsertion

该样例演示了 EPWM 的死区插入功能，PA7,PA8 输出带死区 EPWM 波形，LED1 每隔 1s 翻转一次。

This example demonstrates the dead zone insertion function of EPWM, PA7 and PA8 output EPWM waveforms with dead zones, LED1 flips every 1 second.

10.2. EPWM_DMA

该样例演示了 EPWM 溢出触发 DMA 传输功能，复位串口输出 DMA 传输前的数据，等待 EPWM 触发 DMA 传输后，打印传输后的数据，LED1 每隔 1s 翻转一次。

This example demonstrates the DMA transmission function triggered by EPWM overflow, resetting the serial port to output data before DMA transmission, and waiting

for EPWM to trigger DMA transmission Afterwards, print the transmitted data, LED1 flips every 1 second.

10.3. EPWM_EmergencyStop

该样例演示了 EPWM 的软件触发紧急停止功能，复位后 EPWM 正常输出波形 1s，软件触发急停 1s，最后恢复正常输出，LED1 每隔 1s 翻转一次。

This example demonstrates the software triggered emergency stop function of EPWM. After resetting, EPWM outputs a normal waveform for 1 second, the software triggered emergency stop for 1 second, and finally Restore normal output, LED1 flips every 1 second.

10.4. EPWM_ForceOutput

该样例演示了 EPWM 的强制输出功能，复位 EPWM 正常输出波形 1s，然后配置强制输出电平 1s，最后恢复正常输出，LED1 每隔 1s 翻转一次。

This example demonstrates the forced output function of EPWM, resetting the normal output waveform of EPWM for 1 second, then configuring the forced output level for 1 second, and finally Restore normal output, LED1 flips every 1 second.

10.5. EPWM_OutputInterrupt

该样例演示了 EPWM 的正常输出和中断功能，PA7,PA8 正常输出 EPWM 波形，EPWM 溢出中断翻转 LED2，LED1 每隔 1s 翻转一次。

This example demonstrates the normal output and interrupt functions of EPWM, PA7 and PA8 output EPWM waveforms normally, EPWM overflow interrupt flips LED2, LED1 flips every 1 second.

11. FLASH

11.1. Flash_EraseWriteRead

该样例演示了 Flash 擦除、读、写功能。串口打印数据，若出现错误，LED8 蓝灯闪烁。该样例演示了 FLASH 的擦除、写入、读出功能，先擦除 flash 最后一页，读取数据判断是否擦除成功，然后写入数据，读取数据判断是否写入成功，串口输出日志，如果有错则快闪 LED2，LED1 每隔 1s 翻转一次。

This example demonstrates the erase, write, and read functions of FLASH. First, erase the last page of the flash and read the data to determine whether the erase is successful. Then write the data, read the data to determine if the writing is successful, output a log through the serial port, and if there is an error, flash LED2, LED1 every 1 second Flip once.

11.2. Flash_ReadWaitAndPrefetch

该样例演示了 FLASH 修改读 flash 等待时间和开启 flash 预取指令功能，开启后可提高程序执行效率。LED1 每隔 1s 翻转一次。

This example demonstrates how FLASH can modify the read flash wait time and enable the flash prefetch instruction function, which can improve program execution efficiency when enabled. LED1 flips every 1 second.

11.3. Flash_UserOption

该样例演示了 FLASH 操作用户选项字节区的功能，复位后配置对应用户选项字节区功能，如果出错，LED2 快闪，LED1 每隔 1s 翻转一次。

This example demonstrates the function of operating the user option byte area with FLASH. After resetting, configure the corresponding user option byte area function. If an

error occurs, LED2 flash, LED1 flips every 1 second.

12. GPIO

12.1. GPIO_KeyIrqMode

此样例演示了 GPIO 中断功能，配置 4 个按键引脚为输入中断模式，KEY_UP 按键配置上升沿中断，其余配置下降沿，当按下按键后，翻转对应的 LED。

This example demonstrates the GPIO interrupt function, with 4 key pins configured as input interrupt mode, and the KEY_UP key configured with rising edge interrupt, Other configurations have a falling edge. When the button is pressed, flip the corresponding LED.

12.2. GPIO_KeyPollingMode

此样例演示了 GPIO 输入模式，配置 4 个按键引脚为数字输入模式，KEY_UP 按键配置下拉输入，其余配置上拉输入，当按下按键后，翻转对应的 LED。

This example demonstrates the GPIO input mode, where four key pins are configured as digital input mode, and the KEY_UP button is configured as a pull-down input, For other configurations, pull up the input and flip the corresponding LED when the button is pressed..

12.3. GPIO_ToggleLed

此样例演示了 GPIO 输出模式，配置 4 个 LED 引脚为数字输出模式，并且每隔 1000ms 依次点亮 4 个 LED 灯。运行程序后，可以看到 4 个 LED 依次点亮。

This example demonstrates GPIO output mode, with 4 LED pins configured for digital output mode and lit up sequentially every 1000ms 4 LED lights. After running the program,

you can see that four LEDs light up in sequence.

13. I2C

13.1. I2C_EEPROM

此样例演示了 I2C 操作 eeprom 芯片 AT24C02, I2C 写入数据后读出校验, 成功 LED1 慢闪, 失败 LED1 快闪, 串口输出日志。

This example demonstrates the I2C operation of the EEPROM chip AT24C02, I2C reads out verification after writing data, successful LED1 slow flashing, failed LED1 Flash, serial port output logs.

13.2. I2C_MasterBlock

此样例演示了 I2C 主机以阻塞模式读写从机, 串口输出日志, LED1 每隔 1s 翻转一次。

This example demonstrates that the I2C host reads and writes to the slave in blocking mode, and the serial port outputs logs, LED1 flips every 1 second.

13.3. I2C_MasterDMA

此样例演示了 I2C 主机以 DMA 方式模式读写从机, 串口输出日志, LED1 每隔 1s 翻转一次。

This example demonstrates that the I2C host reads and writes to the slave in DMA mode, and the serial port outputs logs, LED1 flips every 1 second.

13.4. I2C_MasterInterrupt

此样例演示了 I2C 主机以中断模式读写从机, 串口输出日志, LED1 每隔 1s 翻转一次。

This example demonstrates that the I2C host reads and writes to the slave in interrupt mode, and the serial port outputs logs, LED1 flips every 1 second.

13.5. I2C_SlaveBlock

此样例演示了 I2C 从机以阻塞方式模式读写数据，串口输出日志，LED1 每隔 1s 翻转一次。

This example demonstrates that the I2C slave reads and writes to the slave in blocking mode, and the serial port outputs logs, LED1 flips every 1 second.

13.6. I2C_SlaveDMA

此样例演示了 I2C 从机以 DMA 方式模式读写数据，串口输出日志，LED1 每隔 1s 翻转一次。

This example demonstrates that the I2C slave reads and writes to the slave in DMA mode, and the serial port outputs logs, LED1 flips every 1 second.

13.7. I2C_SlaveInterrupt

此样例演示了 I2C 从机以中断方式模式读写数据，串口输出日志，LED1 每隔 1s 翻转一次。

This example demonstrates that the I2C slave reads and writes to the slave in interrupt mode, and the serial port outputs logs, LED1 flips every 1 second.

14. LPTIM

14.1. LPTIM_CounterInterrupt

此样例演示了 LPTIM 的计数和中断功能，在 LPTIM 计数器溢出中断翻转 LED2，串口输出日志，LED1 每隔 1s 翻转一次。

This example demonstrates the counting and interrupt functions of LPTIM. When the LPTIM counter overflows, the interrupt flips LED2 and the serial port outputs a log, LED1 Flip every 1 second.

15. PGA

15.1. PGA_DifferentialInput

此样例演示了 PGA 的差分输入 ADC 采样功能，调节两个滑动变阻器，改变 PGA 两端输入电压，



串口实时打印 PGA 放大后的电压值，LED1 每隔 1s 翻转一次。

This example demonstrates the differential input ADC sampling function of PGA, adjusting two sliding resistors, changing the input voltage at both ends of PGA, and the serial port Real time printing of the amplified voltage value of PGA, LED1 flips every 1 second.

16. PVD

16.1. PVD_Interrupt

此样例演示了 PVD 触发复位功能，当芯片电源电压小于 2.6V 时，触发 PVD 中断，翻转 LED2，串口打印日志，LED1 每隔 1s 翻转一次。

This example demonstrates the PVD trigger reset function. When the chip power supply voltage is less than 2.6V, the PVD interrupt is triggered, the LED2 is flipped, and the serial port is printed Logs, LED1 flips every 1 second.

16.2. PVD_Reset

此样例演示了 PVD 触发复位功能，当芯片电源电压小于 2.6V 时，系统复位，串口打印日志，LED1 每隔 1s 翻转一次。

This example demonstrates the PVD trigger reset function. When the chip power supply voltage is less than 2.6V, the system resets and the serial port prints logs, LED1 Flip every 1 second.

17. RCC

17.1. RCC_MCOoutput

此样例演示了 MCO 输出时钟功能，PA8 输出 HSI 的 8 分频，LED1 每隔 1s 翻转一次。

This example demonstrates the MCO output clock function, PA8 outputs HSI's 8-division frequency, LED1 flips every 1 second.

17.2. RCC_SysclkConfig

此样例演示了切换系统时钟功能。先初始化系统时钟为 PLL, HCLK 分频为系统时钟一分频, PCLK 分频为 HCLK 一分频。再切换系统时钟为 HSI, LSCLK 时钟源为 LSI。再切换系统时钟为 PLL, LSCLK 时钟源为 LSI。

This example demonstrates the function of switching the system clock. First, initialize the system clock to PLL, HCLK division to system clock one division, and PCLK division to HCLK one division. Switch the system clock to HSI again, The LSCLK clock source is LSI. Switch the system clock to PLL and the LSCLK clock source to LSI.

18. LowPower

18.1. RCC_LowPowerDeepSleep_UART

此样例演示了 MCU 进入 DeepSleep 模式和串口接收中断唤醒功能, 串口接收中断唤醒后, LED1 闪烁一次, 再次进入 DeepSleep 模式, 一直循环。

This example demonstrates the MCU entering DeepSleep mode and the serial port receiving interrupt wake-up function. After the serial port receives interrupt wake-up, LED1 flashes once, then again Enter DeepSleep mode and keep looping.

18.2. RCC_LowPowerStop_LPTIM

此样例演示了 MCU 进入 Stop 模式和 LPTIM 中断唤醒功能, LPTIM 中断唤醒后, LED1 闪烁一次, 再次进入 Stop 模式, 一直循环。

This example demonstrates the MCU entering Stop mode and LPTIM interrupt

wake-up function, After LPTIM interrupt wakes up, LED1 flashes once, then again Enter Stop mode and keep looping.

18.3. RCC_LowPowerStop_GPIO

此样例演示了 MCU 进入 Stop 模式和 GPIO 中断唤醒功能，GPIO 中断唤醒后，LED1 闪烁一次后进入 Stop 模式。给 PA8 输入上升沿信号，唤醒，同时 LED2 状态翻转。

This example demonstrates the MCU entering Stop mode and GPIO interrupt wake-up function. After GPIO interrupt wake-up, LED1 flashes once before entering Stop mode. Input a rising edge signal to PA8 wake up, and simultaneously flip the LED2 state.

19. SPI

19.1. SPI_MasterBlock

此样例演示了 SPI 主机阻塞收发数据功能，SPI 主机发送 5 个数据，并从从机读取 5 个数据，串口打印日志，LED1 每隔 1s 翻转一次。

This example demonstrates the SPI host block sending and receiving data function, SPI host sends 5 pieces of data and reads 5 pieces of data from slave, printing through serial port Logs, LED1 flips every 1 second.

19.2. SPI_MasterDMA

此样例演示了 SPI 主机 DMA 收发数据功能，SPI 主机发送 5 个数据，并从从机读取 5 个数据，串口打印日志，LED1 每隔 1s 翻转一次。

This example demonstrates the SPI host DMA sending and receiving data function, SPI host sends 5 pieces of data and reads 5 pieces of data from slave, printing through serial port Logs, LED1 flips every 1 second.

19.3. SPI_MasterInterrupt

串口打印发送与接收的数据。此样例演示了 SPI 主机中断收发数据功能, SPI 主机发送 5 个数据, 并从从机读取 5 个数据, 串口打印日志, LED1 每隔 1s 翻转一次。

This example demonstrates the SPI host interrupt sending and receiving data function, SPI host sends 5 pieces of data and reads 5 pieces of data from slave, printing through serial port Logs, LED1 flips every 1 second.

19.4. SPI_SlaveBlock

该样例演示了 SPI 作为从机查询方式发送与接收功能。串口打印发送与接收的数据。此样例演示了 SPI 从机阻塞收发数据功能, 主机发起读数据流程, 从机发送 spi_tx_buf 里面的数据, 主机发起写数据流程, 从机接收数据, LED1 每隔 1s 翻转一次。

This example demonstrates the SPI slave block data sending and receiving function, where the host initiates the data reading process and the slave sends the spi_tx_buf file Data, the host initiates the data writing process, the slave receiving data, and LED1 flips every 1 second.

19.5. SPI_SlaveDMA

该样例演示了 SPI 作为从机 DMA 自动传输方式发送与接收功能。串口打印发送与接收的数据。此样例演示了 SPI 从机 DMA 收发数据功能, 主机发起读数据流程, 从机发送 spi_tx_buf 里面的数据, 主机发起写数据流程, 从机 DMA 接收数据, LED1 每隔 1s 翻转一次。

This example demonstrates the SPI slave DMA data sending and receiving function, where the host initiates the data reading process and the slave sends the spi_tx_buf file Data, the host initiates the data writing process, the slave DMA receiving data, and LED1

flips every 1 second.

19.6. SPI_SlaveInterrupt

此样例演示了 SPI 从机中断收发数据功能，主机发起读数据流程，从机发送 spi_tx_buf 里面的数据，主机发起写数据流程，从机中断接收数据，LED1 每隔 1s 翻转一次。此样例演示了 SPI 主机操作 W25Q80 外部 flash 功能，SPI 读取外部 flash 信息，并进行擦除，写入，读取等操作，串口打印日志，LED1 每隔 1s 翻转一次。

This example demonstrates the SPI slave interrupt data sending and receiving function, where the host initiates the data reading process and the slave sends the spi_tx_buf file Data, the host initiates the data writing process, the slave interrupts receiving data, and LED1 flips every 1 second.

19.7. SPI_w25qxx

此样例演示了 SPI 主机操作 W25Q80 外部 flash 功能，SPI 读取外部 flash 信息，并进行擦除，写入，读取等操作，串口打印日志，LED1 每隔 1s 翻转一次。

This example demonstrates the SPI host operation W25Q80 external flash function, SPI reads external flash information, erases and writes it, Read and other operations, print logs through serial port, LED1 flips every 1 second.

20. TIM

20.1. TIM_CuontInterrupt

此样例演示了 TIM0 定时和中断功能，TIM0 定时 100ms 产生一次中断，在中断里面翻转 LED2，LED1 每隔 1s 翻转一次。

This example demonstrates the TIM0 timing and interrupt functions. TIM0 generates

an interrupt every 1 second and flips LED2 within the interrupt. LED1 flips every 1 second Rotate once.

20.2. TIM_DMA

此样例演示了 TIM0 触发 DMA 功能，系统复位 UART0 打印要搬运的数据，等待 DMA 搬运完成，打印搬运后的数据，LED1 每隔 1s 翻转一次。

This example demonstrates the DMA function triggered by TIM0, where the system resets UART0 to print the data to be moved, waits for DMA transfer to complete, and then prints After transportation, LED1 flips the data every 1 second.

21. UART

21.1. UART_232_Interrup

此样例演示了 UART2 的 232 中断收发数据功能，系统复位后，UART2 通过 232 输出字符串，输出完成

后等待接收 32 字节数据完成，接收完成串口打印日志，LED1 每个 1s 翻转一次。

This example demonstrates the 232 interrupt data sending and receiving function of UART2. After the system is reset, UART2 outputs a string through 232. After the output is completed, it waits for the reception of 32 bytes of data to be completed. After the reception is completed, the serial port prints the log, and LED1 flips every 1 second.

21.2. UART_485_MasterInterrupt

此样例演示了 UART1 通过 485 中断发送接收功能，按下 KYE_UP 按键，485 发送一个字符串并接收一个字符串，LED1 每隔 1s 翻转一次。

This example demonstrates UART1's function of sending and receiving interrupts

through 485. By pressing the KYE.UP button, 485 sends a string and receives it String, LED1 flips every 1 second.

21.3. UART_Block

此样例演示了 UART0 发送接收功能，系统复位 UART0 发送 tx_buf 的数据，发送完成后开始接收数据，接收满 255 字节或者收到字符 '#' 后，串口打印收到的字符串，LED1 每隔 1s 翻转一次。

This example demonstrates the UART0 sending and receiving function. The system resets the data sent by UART0 to tx_buf, and starts receiving the data after it is sent, After receiving 255 bytes or the character '#', the serial port prints the received string, and LED1 flips it every 1 second.

21.4. UART_DMA

该样例演示了串口 DMA 自动传输模式发送与接收数据，串口打印数据。此样例演示了 UART0 DMA 发送接收功能，复位 UART0 DMA 发送字符串 "UART DMA receive and transmit example, please input 10 bytes:\r\n"，发送完成后，等待 DMA 接收 10 字节完成，接收完成将收到的数据打印出来，LED1 每隔 1s 翻转一次。

This example demonstrates the UART0 DMA sending and receiving function, resetting the UART0 DMA sending string "UART DMA receive and transmit example, please input 10 bytes: \r\n". After sending, wait for the DMA to receive 10 bytes, After receiving, print out the received data and flip LED1 every 1 second.

21.5. UART_HalfDuplex

此样例演示了 UART1 半双工模式，配置 PD0 为开漏上拉输出、复用串口模式，需接外部上拉，串口 1 接收一个字节后立马发送出来，并从串口 0 打印接收的数据，LED1 每隔 1s 翻转一次。

This example demonstrates UART1 half duplex mode, with PD0 configured as open drain pull-up output and multiplexed serial port mode, requiring external pull-up connection, Serial port 1 receives a byte and immediately sends it out and print the received data from serial port 0, while LED1 flips every 1 second.

21.6. UART_Interrupt

此样例演示了 UART0 中断发送接收功能，系统复位 UART0 中断发送 tx_buf 的数据，发送完成后开始接收数据，接收满 32 字节数据后，串口打印 "UART receive successfully!"，LED1 每隔 1s 翻转一次。

This example demonstrates the UART0 interrupt sending and receiving function. The system resets the data sent by the UART0 interrupt tx_buf, and starts after the transmission is completed Receive data, after receiving a full 32 bytes of data, print "UART receive successfully!" on the serial port, and LED1 flips every 1 second Rotate once.

21.7. UART_Printf

此样例演示了 UART0 打印功能，复位打印 "Uart printf example"，LED1 每隔 1s 翻转一次。

This example demonstrates the UART0 printing function, with resetting to print "Uart printf example" and LED1 flipping every 1 second.

22. WDG

22.1. WDG_InterruptFeedDog

程序正常运行中，LED9 白灯闪烁。此样例演示了 WDG 的中断和喂狗功能，WDG 配置 WDG_MODE_14BIT 模式，每隔 500ms 产生一次中断，并在中断里面喂狗和翻转 LED2，LED1 作为运行指示灯，每隔 1000ms 翻转一次。

This example demonstrates the interrupt and dog feeding functions of WDG. WDG is configured with WDG-MODE-14BIT mode, which generates every 500ms Interrupt and feed the dog and flip LED2 within the interrupt. LED1 serves as the running indicator light and flips once every 1000ms.