栏目	描述
软件 IDE	AndeSight RSD V5.3.2
调试器	ADP-AICE-MINI+

SPINTROL

目录

版本历史 5

2 MCU Power 6

2.1 MCU 电源过压/欠压触发中断 6

2.2 MCU 电源过压/欠压触发复位 6

3 Clock 7

3.1 时钟错误触发中断 7

3.2 时钟错误触发复位 7

4 GPIO 7

4.1 GPIO 电平翻转检测 7

4.2 GPIO 电平高低检测 8

6 CRC 8

6.1 CRC 正常工作检测 8

7 Timer 9

7.1 Timer 触发中断 9

8 WDT 9

8.1 WDT 喂狗 9

8.2 WDT 触发中断 9

9 PWM 10

9.1 PWM 输出互补波形 10

9.2 非级联同步方式设置 PWM 相位差-GPIO 触发 10

9.3 PWM 信号封锁-GPIO 触发 11

9.4 PWM 触发 ADC 采集 12

9.5 PWM 输出单通道波形 12

9.6 非级联同步方式设置 PWM 相位差-软件触发 12

9.7 级联同步方式设置 PWM 相位差-首节点软件触发 13

9.8 级联同步方式设置 PWM 相位差-首节点时基触发 14

9.9 非级联同步方式设置 PWM 相位差-Timer 触发 14

9.10 PWM 输出三相波形 15

10 ECAP 16

10.1 ECAP APWM 模式 16

10.2 ECAP 单通道持续捕获-Absolute 模式 16

SPINTROL

©2026 旋智电子科技（上海）有限公司

10.3 ECAP 单通道持续捕获-Delta 模式 16
10.4 ECAP 双通道持续捕获-Delta 模式 17
10.5 ECAP 单通道一次性捕获- Delta 模式 17
10.6 ECAP 单通道一次性捕获- Absolute 模式 18

11 ADC 18
11.1 ADC 开路检测 18
11.2 ADC 短路检测 19
11.3 ADC 双端采样-软件触发 20
11.4 ADC 单端采样 -定时器触发 20
11.5 ADC 单端采样-软件触发 20
11.6 ADC 增益及偏移校准 21

12 T-Sensor 21
12.1 T-sensor 21

13 UART 22
13.1 UART 自动波特率 22
13.2 UART 接收和发送 22
13.3 UART 接收不定长数据 22
13.4 UART 发送和接收 23

14 I2C 23
14.1 I2C 主机模式-阻塞方式 23
14.2 I2C 从机模式-阻塞方式 23

15 SPI 24
15.1 SPI 主机模式-阻塞方式 24
15.2 SPI 从机模式-阻塞方式 24
15.3 SPI 主机模式-非阻塞中断方式 25
15.4 SPI 从机模式-非阻塞中断方式 25

16 FLASH 26
16.1 FLASH 基础操作 26
16.2 在 RAM 中运行代码及数据 26

17 DMA 27
17.1 ADC 搬移到 Memory 27
17.2 ADC 搬移到 SPI 27
17.3 Memory 搬移到 Memory 27
17.4 Memory 搬移到 SPI 28
17.5 Memory 搬移到 UART 28

SPINTROL

17.6 SPI 搬移到 Memory 29
17.7 SPI 搬移到 UART 29
17.8 UART 搬移到 Memory 29
17.9 UART 搬移到 SPI 30

18 CAN 30

18.1 CAN 接收标准帧-阻塞方式 30
18.2 CAN 发送标准帧-阻塞方式 31
18.3 CANFD 接收标准帧-阻塞方式 31
18.4 CANFD 发送标准帧-阻塞方式 31
18.5 CAN 接收标准帧-非阻塞中断方式 32
18.6 CAN 发送标准帧-非阻塞中断方式 32

19 Multi-core 33

19.1 CPU0 启动 CPU1、CPU2、CPU3 33
19.2 CPU1 触发 WDT 中断 33
19.3 CPU2 触发 WDT 中断 34
19.4 CPU3 触发 WDT 中断 34

SPINTROL

©2026 旋智电子科技（上海）有限公司

版本历史

版本	日期	作者	状态	变更
C/0	2025-06-19	Xqhe	Released	1. 首次发布

SPINTROL

2 MCU Power

2.1 MCU 电源过压/欠压触发中断

栏目	描述
工程名	BOD_UVOV_Int
功能	MCU 电源过压/欠压触发中断
硬件环境	1. 需可调电源供电。
步骤	1. 设置 VDD3.3V 电源域的欠压检测，设置为 3.02V 时，欠压事件触发；3.11V 时，欠压事件消失； 设置 VDD3.3V 电源域的过压检测，设置为 3.78V 时，过压事件触发；3.65V 时，过压事件消失； 2. 使能 VDD3.3V 电源域触发的中断为不可屏蔽中断，填写中断服务函数，在函数中查询过压/欠压事件发生标志，并清除中断。
预期结果	1. 微调可调电源，当低于 3.02V 时，打印 VDD33 触发欠压中断，直至调整电源高于 3.11V 时，打印消失； 2. 微调可调电源，当高于 3.78V 时，打印 VDD33 触发过压中断，直至调整电源低于 3.65V 时，打印消失。
注意点	

2.2 MCU 电源过压/欠压触发复位

栏目	描述
工程名	BOD_UVOV_Reset
功能	MCU 电源过压/欠压触发复位
硬件环境	1. 需可调电源供电。
步骤	1. 查询过压/欠压复位事件发生标志； 2. 设置 VDD3.3V 电源域的欠压检测，设置为 3.02V 时，欠压事件触发； 设置 VDD3.3V 电源域的过压检测，设置为 3.78V 时，过压事件触发； 3. 使能 VDD3.3V 电源域的过压/欠压事件触发复位。
预期结果	1. 微调可调电源，当低于 3.02V 时，打印 VDD33 触发欠压复位； 2. 微调可调电源，当高于 3.78V 时，打印 VDD33 触发过压复位。
注意点	1. 撤销过压/欠压复位事件的电压值必须与触发过压/欠压复位事件的电压值一致。

©2026 旋智电子科技（上海）有限公司

3 Clock

3.1 时钟错误触发中断

栏目	描述
工程名	Clock_Detect_Error_Int
功能	时钟错误触发中断
硬件环境	
步骤	1. 初始化时钟探测，配置探测的 PLL 时钟为 300MHz，误差为 1%； 2. 使能 PLL 失锁产生不可屏蔽中断，并填写中断服务函数，在函数中查询 PLL 失锁发生标志，并清除中断； 3. 使能时钟检测。
预期结果	1. 程序运行后，触发时钟探测错误中断，打印中断次数持续增加。
注意点	

3.2 时钟错误触发复位

栏目	描述
工程名	Clock_Detect_Error_Reset
功能	时钟错误触发复位
硬件环境	
步骤	1. 查询 PLL 失锁发生标志； 2. 初始化时钟探测，配置探测的 PLL 时钟为 300MHz，误差为 1%； 3. 使能 PLL 失锁产生复位； 4. 使能时钟检测。
预期结果	1. 程序运行后，先打印没有发生时钟错误事件； 2. 程序继续运行，产生失锁事件后，芯片复位； 3. 芯片复位后，打印发生时钟错误事件。
注意点	

4 GPIO

4.1 GPIO 电平翻转检测

栏目	描述
工程名	GPIO_Detect_Edge
功能	采用边沿检测方式，判断检测 GPIO 是否发生电平反转。
硬件环境	1. 将 DEMO_GPIO_INPUT 脚与 DEMO_GPIO_OUTPUT 脚连接。

©2026 旋智电子科技（上海）有限公司

栏目	描述
步骤	1. 配置 DEMO_GPIO_OUTPUT 脚为输出； 2. 配置 DEMO_GPIO_INPUT 脚为输入，并配置抗尖峰脉冲滤波； 3. 配置 DEMO_GPIO_INPUT 脚为上升沿检测，使能边沿中断，填写 GPIO 边沿中断服务函数，在函数中查询 DEMO_GPIO_INPUT 脚是否存在上升沿标志； 4. 通过 DEMO_GPIO_OUTPUT 脚翻转输出，在 DEMO_GPIO_INPUT 脚产生上升沿。
预期结果	1. 程序运行后，触发检测上升沿中断，打印中断次数持续增加。
注意点	

4.2 GPIO 电平高低检测

栏目	描述
工程名	GPIO_Detect_Level
功能	检测 GPIO 电平为高还是低
硬件环境	1. 将 DEMO_GPIO_INPUT 脚与 DEMO_GPIO_OUTPUT 脚连接。
步骤	1. 配置 DEMO_GPIO_OUTPUT 脚为输出； 2. 配置 DEMO_GPIO_INPUT 脚为输入； 3. 每次 DEMO_GPIO_OUTPUT 脚输出翻转后，使用 DEMO_GPIO_OUTPUT 脚进行回采，并使用 DEMO_GPIO_INPUT 脚进行输入电平采集。
预期结果	1. 程序运行后， DEMO_GPIO_OUTPUT 脚的回采电平与 DEMO_GPIO_INPUT 脚的电平一致。
注意点	

6 CRC

6.1 CRC 正常工作检测

栏目	描述
工程名	CRC
功能	检测 CRC 是否工作正常
硬件环境	
步骤	1. 配置CRC并传入一段字符串，将计算结果与网上计算过的结果对比，以检查 CRC 是否工作正常。
预期结果	1. 程序运行后，CRC 计算结果与网上计算过的结果一致。
注意点	

©2026 旋智电子科技（上海）有限公司

7 Timer

7.1 Timer 触发中断

栏目	描述
工程名	TIMER_Int
功能	设置 Timer 开始计时，结束后进入中断
硬件环境	
步骤	1. 初始化 Timer，配置为每 1000ms 触发 Timer 中断； 2. 使能 Timer 中断，填写 Timer 中断服务函数，在函数中记录中断触发次数，并清除中断； 3. 使能 Timer。
预期结果	1. 每 1000ms Timer 中断触发次数加一。
注意点	

8 WDT

8.1 WDT 喂狗

栏目	描述
工程名	WDT_Feed_Dog
功能	WTD 计时一定时间，但在结束之前重新喂狗，使其不进入超时中断
硬件环境	
步骤	1. 检查复位事件是否被置位； 2. 初始化 WDT，配置为每 50ms 触发一次中断； 3. 使能 WDT 超时产生复位事件； 4. 使能 WDT 中断，填写中断服务函数； 5. 使能 WDT； 6. 每 30ms 进行一次喂狗，喂 100 次； 7. 不再喂狗。
预期结果	1. 停止喂狗后，触发复位。
注意点	

8.2 WDT 触发中断

栏目	描述
工程名	WDT_Int
功能	WTD 计时一定时间，结束后进入中断。

©2026 旋智电子科技（上海）有限公司

栏目	描述
硬件环境	
步骤	1. 初始化 WDT， 配置为每 1000ms 触发一次中断； 2. 使能 WDT 中断，填写中断服务函数，在函数中查询 WDT 超时发生标志。如标志有效，则记录中断触发次数，并清除中断； 3. 使能 WDT。
预期结果	1. 每 1000ms WDT 中断触发次数加一。
注意点	

9 PWM

9.1 PWM 输出互补波形

栏目	描述
工程名	PWM_Complementary_Pair_Channel
功能	设置某路 PWM，并使其两个通道（A/B）产生一对互补的波形
硬件环境	1. 芯片 DEMO_PWMA_PIN、DEMO_PWMB_PIN 引脚可外接
步骤	1. 配置 DEMO_PWMA_PIN、DEMO_PWMB_PIN 为 PWM 的输出引脚； 2. 配置 PWM 为双通道，频率为 20Khz，互补波形死区为 1us； 3. 配置 PWM 占空比； 4. 使能 PWM。
预期结果	1. 通过示波器等检测工具测量 DEMO_PWMA_PIN、DEMO_PWMB_PIN，可获得频率为 20Khz、死区为 1us 的互补波形。
注意点	

9.2 非级联同步方式设置 PWM 相位差-GPIO 触发

栏目	描述
工程名	PWM_GPIO_SYNCI
功能	设置 PWM 输出电机三相波形，使用非级联同步方式，通过 GPIO 高电平触发，产生 120° 相位差
硬件环境	1. 芯片 DEMO_PWMA_U_PIN、DEMO_PWMB_U_PIN、DEMO_PWMA_V_PIN、DEMO_PWMB_V_PIN、DEMO_PWMA_W_PIN、DEMO_PWMB_W_PIN 引脚可外接
步骤	1. 配置 DEMO_GPIO_INPUT 为输入模式并使能下拉； 2. 配置 PWM_U、PWM_V、PWM_W 输出通道引脚； 3. 配置 PWM_U、PWM_V、PWM_W，频率为 20Khz，互补波形死区为 1000ns； 4. 配置 PWM_U、PWM_V、PWM_W 占空比为 33.33%；

©2026 旋智电子科技（上海）有限公司

栏目	描述
	5. 配置非级联同步，由 DEMO_GPIO_INPUT 高电平触发，同步后三相相位差 120°； 6. 使能 PWM_U、PWM_V、PWM_W；
预期结果	1. 通过示波器等检测工具测量 PWM_U 引脚，可获得频率为 20Khz、占空比为 33.33%、死区为 1000ns 的互补波形； 2. 通过示波器等检测工具测量 PWM_V 引脚，可获得频率为 20Khz、占空比为 33.33%、死区为 1000ns 的互补波形； 3. 通过示波器等检测工具测量 PWM_W 引脚，可获得频率为 20Khz、占空比为 33.33%、死区为 1000ns 的互补波形； 4. 将 DEMO_GPIO_INPUT 拉高； 5. 将 DEMO_GPIO_INPUT 拉低； 6. 通过示波器等测量工具测量 PWM_U、PWM_V、PWM_W 的各自 A/B 通道的相位差为 120°。
注意点	同步信号需要为脉冲波形，否则 DEMO_GPIO_INPUT 上的高电平后会一直触发 PWM 同步

9.3 PWM 信号封锁-GPIO 触发

栏目	描述
工程名	PWM_GPIO_Trigger_TZ
功能	1. 设置 GPIO 为 TZ0 的触发源，并设置为单次封锁模式，当电平满足触发要求时，触发 TZ 事件，关闭 PWM 输出，直到清除 TZ 标志位。 2. 另一组 GPIO 分别作为 TZ1/2 触发源，并设置为周期封锁模式，当电平满足触发要求时，触发 TZ 事件，关闭当前周期 PWM 输出。
硬件环境	芯片 DEMO_PWMB_PIN 引脚可外接
步骤	1. 配置 DEMO_TZ0_PIN、DEMO_TZ1_PIN、DEMO_TZ2_PIN 为 GPIO 引脚，并使能滤波； 2. 配置 DEMO_PWMB_PIN 为 PWM 引脚； 3. 设置 PWM 单通道，频率为 20Khz ，占空比为 50%； 4. 配置 DEMO_TZx_PIN 极性； 5. 配置 TZ 动作时，PWM 输出为高阻态； 6. 使能 TZ 中断； 7. 使能 PWM。
预期结果	1. DEMO_TZ0_PIN 为低、DEMO_TZ1_PIN 为低、DEMO_TZ2_PIN 为高时，通过示波器等检测工具测量 DEMO_PWMB_PIN，可获得频率为 20Khz、占空比为 25%的波形； 2. 当 DEMO_TZ0_PIN 为高，或 DEMO_TZ1_PIN 为高，或 DEMO_TZ2_PIN 为低时，触发 PWM 封锁。
注意点	

©2026 旋智电子科技（上海）有限公司

9.4 PWM 触发 ADC 采集

栏目	描述
工程名	PWM_Request_ADC_and_Observation
功能	设置 PWM 时基为 0 时，触发 ADC 采集。
硬件环境	1. 芯片 DEMO_PWMA_PIN、DEMO_PWMB_PIN 引脚可外接。
步骤	1. 初始化 ADC，并配置 ADC 平均值次数、采集通道、采集引脚并使能 ADC。 2. 配置 PWM 通道的 GPIO 引脚。 3. 配置 PWM 为双通道，频率为 20Khz，互补波形死区为 1000ns。 4. 配置 PWM 占空比为 25%。 5. 配置每次 PWM 时基为 0 时，触发 ADC 采集，将触发信号输出到 GPIO。 6. 填写 ADC 的 IRQ 中断服务函数，使能 ADC 中断，使能中断。 7. 使能 ADC，并运行 PWM。
预期结果	1. 通过示波器等检测工具测量 DEMO_PWMA_PIN、DEMO_PWMB_PIN，可获得频率为 20Khz、占空比为 25%、死区为 1000ns 的互补波形。 2. 打印 ADC 采集的电压。 3. 通过示波器等观测工具测量 PIN_GPIO3，可观测到 ADC 采集触发信号。
注意点	

9.5 PWM 输出单通道波形

栏目	描述
工程名	PWM_Single_Channel
功能	PWM 输出单通道波形
硬件环境	芯片 DEMO_PWMA_PIN 引脚可外接。
步骤	1. 配置 PWM 通道引脚； 2. 配置 PWM 为单通道，频率为 20Khz； 3. 配置 PWM 占空比为 50%； 4. 使能 PWM 中断； 5. 使能 PWM。
预期结果	通过示波器等检测工具测量 DEMO_PWMA_PIN，可获得频率为 20Khz、占空比为 50%的 PWM 波形。
注意点	

9.6 非级联同步方式设置 PWM 相位差-软件触发

栏目	描述
工程名	PWM_Software_Global_SYNCI
功能	设置 PWM 输出电机三相波形，使用非级联同步方式，通过软件信号，产生 120° 相位差。

©2026 旋智电子科技（上海）有限公司

栏目	描述
硬件环境	1. 芯片 DEMO_PWMA_U_PIN、DEMO_PWMB_U_PIN、DEMO_PWMA_V_PIN、DEMO_PWMB_V_PIN、DEMO_PWMA_W_PIN、DEMO_PWMB_W_PIN 引脚可外接
步骤	1. 配置 PWM_U、PWM_V、PWM_W 输出通道引脚； 2. 配置 PWM_U、PWM_V、PWM_W； 3. 配置 PWM_U、PWM_V、PWM_W 占空比为 33.33%； 4. 配置非级联同步，由软件触发，同步后三相相位差 120°； 5. 使能 PWM_U、PWM_V、PWM_W； 6. 强制 PWM_U、PWM_V、PWM_W 软件同步。
预期结果	1. 通过示波器等检测工具测量 PWM_U 引脚，可获得频率为 20Khz、占空比为 33.33%、死区为 1000ns 的互补波形； 2. 通过示波器等检测工具测量 PWM_V 引脚，可获得频率为 20Khz、占空比为 33.33%、死区为 1000ns 的互补波形； 3. 通过示波器等检测工具测量 PWM_W 引脚，可获得频率为 20Khz、占空比为 33.33%、死区为 1000ns 的互补波形； 4. 通过示波器等测量工具测量 PWM_U、PWM_V、PWM_W 的各自 A/B 通道的相位差为 120°。
注意点	

9.7 级联同步方式设置 PWM 相位差-首节点软件触发

栏目	描述
工程名	PWM_Software_Group_SYNCI
功能	设置 PWM 输出电机三相波形，使用级联同步方式，通过软件信号施加首节点，产生 120° 相位差。
硬件环境	1. 芯片 DEMO_PWMA_U_PIN、DEMO_PWMB_U_PIN、DEMO_PWMA_V_PIN、DEMO_PWMB_V_PIN、DEMO_PWMA_W_PIN、DEMO_PWMB_W_PIN 引脚可外接
步骤	1. 配置 PWM_U、PWM_V、PWM_W 输出通道引脚； 2. 配置 PWM_U、PWM_V、PWM_W； 3. 配置 PWM_U、PWM_V、PWM_W 占空比为 33.33%； 4. 配置级联同步，由软件触发，同步后三相相位差 120°； 5. 使能 PWM_U、PWM_V、PWM_W； 6. 强制 PWM_U 软件同步。
预期结果	1. 通过示波器等检测工具测量 PWM_U 引脚，可获得频率为 20Khz、占空比为 33.33%、死区为 1000ns 的互补波形； 2. 通过示波器等检测工具测量 PWM_V 引脚，可获得频率为 20Khz、占空比为 33.33%、死区为 1000ns 的互补波形； 3. 通过示波器等检测工具测量 PWM_W 引脚，可获得频率为 20Khz、占空比为 33.33%、死区为 1000ns 的互补波形；

SPINTROL

栏目	描述
	4. 通过示波器等测量工具测量 PWM_U、PWM_V、PWM_W 的各自 A/B 通道的相位差为 120°。
注意点	

9.8 级联同步方式设置 PWM 相位差-首节点时基触发

栏目	描述
工程名	PWM_TBCNT_0_SYNCO
功能	设置 PWM 输出电机三相波形，使用级联同步方式，通过首节点时基为 0 时触发，产生 120°相位差。
硬件环境	芯片 DEMO_PWMA_U_PIN、DEMO_PWMB_U_PIN、DEMO_PWMA_V_PIN、DEMO_PWMB_V_PIN、DEMO_PWMA_W_PIN、DEMO_PWMB_W_PIN 引脚可外接
步骤	1. 配置 PWM_U、PWM_V、PWM_W 输出通道引脚； 2. 配置 PWM_U、PWM_V、PWM_W； 3. 配置 PWM_U、PWM_V、PWM_W 占空比为 33.33%； 4. 配置级联同步，由时基零触发，同步后三相相位差 120°； 5. 使能 PWM_U、PWM_V、PWM_W。
预期结果	1. 通过示波器等检测工具测量 PWM_U 引脚，可获得频率为 20Khz、占空比为 33.33%、死区为 1000ns 的互补波形； 2. 通过示波器等检测工具测量 PWM_V 引脚，可获得频率为 20Khz、占空比为 33.33%、死区为 1000ns 的互补波形； 3. 通过示波器等检测工具测量 PWM_W 引脚，可获得频率为 20Khz、占空比为 33.33%、死区为 1000ns 的互补波形； 4. 通过示波器等测量工具测量 PWM_U、PWM_V、PWM_W 的各自 A/B 通道的相位差为 120°。
注意点	

9.9 非级联同步方式设置 PWM 相位差-Timer 触发

栏目	描述
工程名	PWM_Timer_SYNCI
功能	设置 PWM 输出电机三相波形，使用非级联同步方式，通过 Timer 计时触发，产生 120°相位差。
硬件环境	芯片 DEMO_PWMA_U_PIN、DEMO_PWMB_U_PIN、DEMO_PWMA_V_PIN、DEMO_PWMB_V_PIN、DEMO_PWMA_W_PIN、DEMO_PWMB_W_PIN 引脚可外接
步骤	1. 配置 TIMER 定时器 1000ms 触发 PWM 同步事件。

©2026 旋智电子科技（上海）有限公司

栏目	描述
	2. 配置 PWM_U、PWM_V、PWM_W 输出通道引脚，并初始化 PWM_U、PWM_V、PWM_W，频率为 20Khz，互补波形死区为 1000ns。 3. 配置 PWM_U、PWM_V、PWM_W 占空比为 33.33%。 4. 配置非级联同步，由软件触发，同步后三相相位差 120° 5. 使能 PWM_U、PWM_V、PWM_W，和 TIMER 定时器。
预期结果	1. 通过示波器等检测工具测量 PWM_U 引脚，可获得频率为 20Khz、占空比为 33.33%、死区为 1000ns 的互补波形。 2. 通过示波器等检测工具测量 PWM_V 引脚，可获得频率为 20Khz、占空比为 33.33%、死区为 1000ns 的互补波形。 3. 通过示波器等检测工具测量 PWM_W 引脚，可获得频率为 20Khz、占空比为 33.33%、死区为 1000ns 的互补波形。 4. 通过示波器等测量工具测量 PWM_U、PWM_V、PWM_W 的各自 A/B 通道的相位差为 120° 。
注意点	

9.10 PWM 输出三相波形

栏目	描述
工程名	PWM_TripPhase_Waveform
功能	设置 PWM 输出电机三相波形，三相波形无相位差
硬件环境	1. 芯片 DEMO_PWMA_U_PIN、DEMO_PWMB_U_PIN、DEMO_PWMA_V_PIN、DEMO_PWMB_V_PIN、DEMO_PWMA_W_PIN、DEMO_PWMB_W_PIN 引脚可外接
步骤	1. 配置 PWM 引脚； 2. 配置 PWM_U、PWM_V、PWM_W，频率为 20Khz，互补波形死区为 1000ns； 3. 使能 PWM_U、PWM_V、PWM_W。
预期结果	1. 通过示波器等检测工具测量 LO_U、HO_U，可获得频率为 20Khz、占空比为 75%、死区为 1000ns 的互补波形。 2. 通过示波器等检测工具测量 LO_V、HO_V，可获得频率为 20Khz、占空比为 75%、死区为 1000ns 的互补波形。 3. 通过示波器等检测工具测量 LO_W、HO_W，可获得频率为 20Khz、占空比为 75%、死区为 1000ns 的互补波形。
注意点	

©2026 旋智电子科技（上海）有限公司

10 ECAP

10.1 ECAP APWM 模式

栏目	描述
工程名	ECAP_APWM_Function
功能	设置 ECAP 工作在 APWM 模式，并输出 PWM 波形
硬件环境	芯片 DEMO_APWM_PIN 引脚可外接。
步骤	1. 初始化 ECAP 对应 APWM 引脚； 1. 配置 PWM 频率（10kHz）、PWM 占空比（50%）； 2. 运行 ECAP
预期结果	2. 通过示波器等检测工具测量 DEMO_APWM_PIN 输出配置的 PWM 脉冲。
注意点	

10.2 ECAP 单通道持续捕获-Absolute 模式

栏目	描述
工程名	ECAP_Continue_Absolute
功能	设置 ECAP 工作在连续采样的 Absolute 模式，并连续对 PWM 波形进行多次事件采样，以此连续计算 PWM 的频率
硬件环境	
步骤	1. 初始化 PWM 为频率（10khz）、占空比（50%）的单通道输出； 2. 配置 ECAP； 3. 配置输入滤波； 4. 配置 ECAP_CEVT3 中断； 5. 使能 ECAP。
预期结果	1. 持续打印 PWM 的频率
注意点	1. ECAP_CaptureModeInit()函数中默认配置为，CAP0 捕获上升沿、CAP1 捕获下降沿、CPA2 捕获上升沿、CAP3 捕获下降沿（详见代码 NOTE-1）

10.3 ECAP 单通道持续捕获-Delta 模式

栏目	描述
工程名	ECAP_Countinue_Delta
功能	1. 设置 ECAP 工作在连续采样的 Delta 模式，并连续对 PWM 波形进行多次事件采样，以此连续计算 PWM 的频率
硬件环境	
步骤	1. 初始化 PWM 为频率（10khz）、占空比（50%）的单通道输出； 2. 配置 ECAP； 3. 配置输入滤波；

©2026 旋智电子科技（上海）有限公司

栏目	描述
	4. 配置 ECAP_CEVT3 中断； 5. 使能 ECAP。
预期结果	1. 持续打印 PWM 的频率
注意点	1. ECAP_CaptureModeInit()函数中默认配置为，CAP0 捕获上升沿、CAP1 捕获下降沿、CPA2 捕获上升沿、CAP3 捕获下降沿（详见代码 NOTE-1）

10.4 ECAP 双通道持续捕获-Delta 模式

栏目	描述
工程名	ECAP_Countinue_Absolute_Dual_Pin
功能	使用连续捕获模式，以绝对时间戳计数方式，对 PWM 双通道波形进行频率计算。
硬件环境	
步骤	1. 初始化 PWM 为频率（10khz）、占空比（50%）的单通道输出； 2. 配置 ECAP； 3. 配置 ECAP_CEVT3 中断； 4. 使能 ECAP。
预期结果	1. 持续打印 PWM 的频率。
注意点	

10.5 ECAP 单通道一次性捕获- Delta 模式

栏目	描述
工程名	ECAP_Oneshot_Delta
功能	设置 ECAP 工作在 oneshot 的 Absolute 模式，并对 PWM 波形进行多次事件采样，以此计算 PWM 的频率
硬件环境	
步骤	1. 初始化 PWM 为频率（10khz）、占空比（50%）的单通道输出； 2. 配置 ECAP； 3. 配置输入滤波； 4. 配置 ECAP_CEVT3 中断； 5. 使能 ECAP。
预期结果	1. 只发生一次 ECAP 中断，PWM 波形与设定一致。
注意点	1. ECAP_CaptureModeInit()函数中默认配置为，CAP0 捕获上升沿、CAP1 捕获下降沿、CPA2 捕获上升沿、CAP3 捕获下降沿（详见代码 NOTE-1）

©2026 旋智电子科技（上海）有限公司

10.6 ECAP 单通道一次性捕获- Absolute 模式

栏目	描述
工程名	ECAP_Oneshot_Absolute
功能	设置 ECAP 工作在 oneshot 的 Absolute 模式，并对 PWM 波形进行多次事件采样，以此计算 PWM 的频率
硬件环境	
步骤	6. 初始化 PWM 为频率（10khz）、占空比（50%）的单通道输出； 7. 配置 ECAP； 8. 配置输入滤波； 9. 配置 ECAP_CEVT3 中断； 10. 使能 ECAP。
预期结果	2. 只发生一次 ECAP 中断，PWM 波形与设定一致。
注意点	2. ECAP_CaptureModeInit()函数中默认配置为，CAP0 捕获上升沿、CAP1 捕获下降沿、CPA2 捕获上升沿、CAP3 捕获下降沿（详见代码 NOTE-1）

11 ADC

11.1 ADC 开路检测

栏目	描述
工程名	ADC_SHA_Open_Circuit
功能	检测采样器通路上被采样的 ADC GPIO 是否存在虚焊等导致开路的情况
硬件环境	
步骤	1. 将 PIN 脚切成 GPIO 功能，设置为输入，并使能 GPIO 下拉，释放 IO 上残存的电量；将 ADC 采样保持的正端强制接地，释放通路上的残存电量； 2. 将 ADC 采样保持的正端恢复断开状态，将 PIN 脚切成 ADC 功能，设置较长采样时间，以期获得更为准确的测试数据，随后开始测量，获得电压数值 Result1； 3. 将 PIN 脚切成 GPIO 功能，设置为输入，并使能 GPIO 上拉，给 IO 的通路进行预充电；将 ADC 采样保持的正端强制拉高，对采样保持器通路进行预充电； 4. 将 ADC 采样保持的正端恢复断开状态，将 PIN 脚切成 ADC 功能，设置较长采样时间，以期获得更为准确的测试数据，随后开始测量，获得电压数值 Result2；
预期结果	1. 如果 Result1 小于 200mV 且 Result2 大于 3200mV，则打印 ADC 引脚断开。 2. 否则，打印 ADC 引脚连接。
注意点	

©2026 旋智电子科技（上海）有限公司

11.2 ADC 短路检测

栏目	描述
工程名	ADC_SHA_Short_Circuit
功能	检测采样器通路上被采样的 ADC GPIO 是否存在短路情况
硬件环境	
步骤	1. ADC 测量输入 PIN 脚电压，设为结果 DATA1。 2. 将 PIN 脚切成 GPIO 功能，设置为输入，给输入 PIN 接弱下拉 GND，ADC 测量输入 PIN 脚电压，设为结果 DATA2。 3. 将 PIN 脚切成 GPIO 功能，设置为输入，给输入 PIN 接弱上拉 VDD，ADC 测量输入 PIN 脚电压，设为结果 DATA3。
预期结果	如果如果 DATA1 与 DATA2 与 DATA3 的差值在 200mV 内，那么输入引脚短路。
注意点	

SPINTROL

11.3 ADC 双端采样-软件触发

栏目	描述
工程名	ADC_Differential_Result
功能	设置 ADC 工作在双端采样。
硬件环境	芯片 GPIOA0、GPIOA1 引脚可外接。
步骤	1. 初始化 ADC 通道，P 端配置为 GPIOA0，N 端配置 GPIOA1； 2. 配置采样时间、转换时间，设置为软件触发； 3. 进行 ADC 采集； 4. 获取采样结果。
预期结果	1. 循环打印向 GPIOA0 施加电压与 GPIOA1 施加电压的压差值。
注意点	1. 向 GPIOA0、GPIOA1 施加的电压值控制在 0~3.3V 范围内。

11.4 ADC 单端采样 -定时器触发

栏目	描述
工程名	ADC_Triggered_by_Timer
功能	通过定时器触发 ADC 采样
硬件环境	芯片 GPIOA0 引脚可外接
步骤	1. 初始化 Timer，设置为 Timer 触发 ADC； 2. 初始化 ADC 通道，P 端配置为 GPIOA0，N 端配置为 GND； 3. 使能 ADC 中断； 4. 使能 Timer 和 ADC。
预期结果	1. 每秒打印向 GPIOA0 施加的电压。
注意点	1. 向 GPIOA0 施加的电压值控制在 0~3.3V 范围内。

11.5 ADC 单端采样-软件触发

栏目	描述
工程名	ADC_Single_End_Result
功能	设置 ADC 工作在单端采样（有一端被接到 GND）。
硬件环境	芯片 GPIOA0 引脚可外接。
步骤	1. 初始化 ADC 通道，P 端配置为 GPIOA0，N 端配置为 GND; 2. 使能 ADC; 3. 软件触发 ADC 采样； 4. 获取 ADC 结果。
预期结果	1. 循环打印向 GPIOA0 施加的电压值。
注意点	1. 向 GPIOA0 施加的电压值控制在 0~3.3V 范围内。

©2026 旋智电子科技（上海）有限公司

11.6 ADC 增益及偏移校准

栏目	描述
工程名	ADC_Trim
功能	校准 ADC 的增益及偏移数值
硬件环境	芯片 GPIOA0、GPIOA1 引脚可外接。
步骤	1. 初始化 ADC 的输入通道均为 GND； 2. 获得 Offset； 3. 初始化 ADC 的输入通道，采集 3.3V； 4. 获取 3.3V 对应 Code； 5. 计算 Gain。
预期结果	1. ADCOFFSET 以及 ADCGAIN 相对于原值的偏差在可接受范围内。
注意点	

12 T-Sensor

12.1 T-sensor

栏目	描述
工程名	Tsensor
功能	设置 MCU 内部温度传感器，检测芯片温度
硬件环境	
步骤	1. 采集并打印 MCU 温度。
预期结果	2. 与环境温度基本一致。
注意点	

SPINTROL

13 UART

13.1 UART 自动波特率

栏目	描述
工程名	UART_Auto_Baudrate
功能	此示例为 UART 自动设置波特率
硬件环境	
步骤	1. 配置 UART2 引脚； 2. 开始自动波特率检测； 3. 等待自动波特率标志位起来； 4. 检测接受到的数据是否符合预期。
预期结果	1. 上位机向 UART2 发送完 0x7F 后，UART0 打印”Auto-baud success!”。
注意点	

13.2 UART 接收和发送

栏目	描述
工程名	UART_RX_and_SentBack
功能	此示例为 UART 的数据接收端代码，采用 polling 方式接收发送端的数据，并将接收的数据发送回发送端
硬件环境	
步骤	1. 配置 UART2 引脚； 2. 等待接收数据； 3. 将接收到的数据发送出来。
预期结果	1. 见”UART_Tx_and_Check_Rx”。
注意点	

13.3 UART 接收不定长数据

栏目	描述
工程名	UART_RX_Indefinite_Length
功能	此示例为 UART 接收不定长度的数据
硬件环境	
步骤	1. 配置 UART2 引脚； 2. 配置 RX 阈值以及超时时间； 3. 配置 RX 阈值中断以及超时中断；
预期结果	上位机向 UART2 发送完小于 UART RX FIFO 深度的数据后，UART0 将其全部打印出来。

©2026 旋智电子科技（上海）有限公司

栏目	描述
注意点	

13.4 UART 发送和接收

栏目	描述
工程名	UART_TX_and_CheckRX
功能	此示例为 UART 的数据发送端代码，采用 polling 方式发送数据给接收端，并将接收接收端返回的数据进行数据对比
硬件环境	
步骤	1. 配置 UART2 引脚； 2. 发送数据； 3. 接收数据； 4. 校验数据。
预期结果	1. 校验通过，打印” Test pass”。
注意点	

14 I2C

14.1 I2C 主机模式-阻塞方式

栏目	描述
工程名	I2C_Master_Polling_TxRx
功能	I2C 作为 Master，采用阻塞方式，周期性向 Slave 发送和接收数据。
硬件环境	
步骤	1. 配置 PIN_MUX 用于 I2C 的 SDA 和 SCK。 2. 配置 I2C 为主机模式，并设置通信速率。 3. 使能 I2C。 4. 阻塞方式向从机设备发送数据。 5. 阻塞方式接收从机设备的数据。 6. 检测发送和接收的数据是否一致。
预期结果	1. 串口打印发送和接收的数据一致。
注意点	

14.2 I2C 从机模式-阻塞方式

栏目	描述
工程名	I2C_Slave_Polling_TxRx

©2026 旋智电子科技（上海）有限公司

栏目	描述
功能	与“I2C_Master_Polling_TxRx”配套，作为 Slave，接收 Master 的数据，并将其发送回 Master。
硬件环境	
步骤	1. 配置 PIN_MUX 用于 I2C 的 SDA 和 SCK。 2. 配置 I2C 为从机模式，并设置从机地址和通信速率。 3. 使能 I2C。 4. 阻塞方式接收主机设备数据。 5. 阻塞方式将接收到的数据，发送回主机设备。
预期结果	1. 串口打印接收到主机设备的字节数，并发送回主机设备。
注意点	

15 SPI

15.1 SPI 主机模式-阻塞方式

栏目	描述
工程名	SPI_Master_Polling_TxRx
功能	I2C 作为 Master，采用阻塞方式，向 Slave 发送和接收数据。
硬件环境	
步骤	1. 配置 PIN_MUX 用于 CS、CLK、MISO 和 MSIO。 2. 配置 SPI 为主机模式，并设置数据宽度和通信速率。 3. 使能 SPI。 4. 阻塞全双工方式向从机设备发送数据，并检测接收的数据与发送的数据是否一致。
预期结果	1. 串口打印操作成功。
注意点	

15.2 SPI 从机模式-阻塞方式

栏目	描述
工程名	SPI_Slave_Polling_TxRx
功能	与“SPI_Master_Polling_TxRx”配套，向 Master 发送和接收数据。
硬件环境	
步骤	1. 配置 PIN_MUX 用于 CS、CLK、MISO 和 MSIO。 2. 配置 PIN 为从机模式，并设置数据宽度和通信速率。 3. 使能 SPI。 4. 阻塞全双工方式向主机设备发送数据，并检测接收的数据与发送的数据是否一致。

©2026 旋智电子科技（上海）有限公司

栏目	描述
预期结果	1. 串口打印操作成功。
注意点	

15.3 SPI 主机模式-非阻塞中断方式

栏目	描述
工程名	SPI_Master_Int_TxRx
功能	设置 SPI 作为 Master，采用中断方式向 Slave 发送数据，并采用中断方式接收 Slave 返回的数据，对比发送和接收的数据是否一致。
硬件环境	
步骤	1. 配置 PIN_MUX 用于 CS、CLK、MISO 和 MSIO。 2. 配置 PIN 为主机模式，并设置数据宽度和通信速率。 3. 使能 SPI NVIC 中断 4. 使能 SPI。 5. 中断方式向从机设备发送数据。 6. 比较发送的数据与接收的数据是否一致。
预期结果	1. 串口打印发送与接收数据一致。
注意点	

15.4 SPI 从机模式-非阻塞中断方式

栏目	描述
工程名	SPI _Slave_TxRx_INT
功能	与“SSP_Master_TxRx_INT”配套，作为 Slave，接收 Master 的数据，并将其发送回 Master。
硬件环境	
步骤	1. 配置 PIN_MUX 用于 CS、CLK、MISO 和 MSIO。 2. 配置 PIN 为主机模式，并设置数据宽度和通信速率。 3. 使能 SPI NVIC 中断 4. 使能 SPI。 5. 向主机发送数据。
预期结果	1. 串口打印发送与接收数据一致。
注意点	

©2026 旋智电子科技（上海）有限公司

16 FLASH

16.1 FLASH 基础操作

栏目	描述
工程名	Flash_Operation
功能	演示 flash 擦，写，读
硬件环境	
步骤	1. 对测试扇区进行擦，写，读测试。
预期结果	1. 打印擦，写，读成功。
注意点	

16.2 在 RAM 中运行代码及数据

栏目	描述
工程名	Flash_With_ISR_In_RAM
功能	此示例代码演示，如何新建一个工程，将所有的代码及数据全部放在 RAM 中运行（包括中断向量表）
硬件环境	
步骤	1. 将中断向量表搬移到“u32aRAMVector”数组； 1. 设置中断向量表地址为“u32aRAMVector”数组的首地址； 2. 初始化 Timer； 3. 使能 Timer 中断，创建 Timer 中断服务函数并用“_RAM_FUNC_”前缀修饰； 4. 使能 Timer； 5. 在 while 循环中擦除 flash 扇区，并打印 Timer 中断服务函数处理结果。
预期结果	1. 循环打印 Timer 中断服务函数处理结果。
注意点	“u32aRAMVector”数组需设置为 0x400 对齐（详见代码 NOTE-1） 中断服务函数需通过“_RAM_FUNC_”前缀修饰（详见代码 NOTE-2）

©2026 旋智电子科技（上海）有限公司

17 DMA

17.1 ADC 搬移到 Memory

栏目	描述
工程名	DMA_ADC_to_Memory
功能	演示 ADC 搬移到 Memory
硬件环境	
步骤	1. ADC 初始化； 2. DMA 初始化； 3. 使能中断； 4. 使能 DMA 通道以及触发 ADC 采样； 5. 重设目的地址，开始下一次传输。
预期结果	1. 打印的电压与外接电压一致。
注意点	

17.2 ADC 搬移到 SPI

栏目	描述
工程名	DMA_ADC_to_SPI
功能	演示 ADC 搬移到 SPI
硬件环境	
步骤	1. ADC 初始化； 2. SPI 初始化； 3. DMA 初始化； 4. 使能中断； 5. 使能 DMA 通道以及触发 ADC 采样。
预期结果	1. 打印的电压与外接电压一致。
注意点	

17.3 Memory 搬移到 Memory

栏目	描述
工程名	DMA_Memory_to_Memory
功能	演示 Memory 搬移到 Memory
硬件环境	
步骤	1. DMA 初始化； 2. 使能中断； 3. 使能 DMA 通道；

©2026 旋智电子科技（上海）有限公司

栏目	描述
	4. 重设源地址和目的地址，开始下一次传输。
预期结果	1. 打印” 0x22334411”, “0x55334411”, “0x88334411”。
注意点	

17.4 Memory 搬移到 SPI

栏目	描述
工程名	DMA_Memory_to_SPI
功能	演示 Memory 搬移到 SPI
硬件环境	
步骤	1. SPI 初始化； 2. DMA 初始化； 3. 使能中断； 4. 使能 DMA 通道； 5. 重设源地址，开始下一次传输。
预期结果	1. 打印” 0x82131415”, “0x86171819”, “0x80212223” , “0x84252627”。
注意点	

17.5 Memory 搬移到 UART

栏目	描述
工程名	DMA_Memory_to_UART
功能	演示 Memory 搬移到 UART
硬件环境	
步骤	1. UART 初始化； 2. DMA 初始化； 3. 使能中断； 4. 使能 DMA 通道； 5. 重设源地址，开始下一次传输。
预期结果	1. 打印” 0x82131415”。
注意点	

©2026 旋智电子科技（上海）有限公司

17.6 SPI 搬移到 Memory

栏目	描述
工程名	DMA_SPI_to_Memory
功能	演示 SPI 搬移到 Memory
硬件环境	
步骤	1. SPI 初始化； 2. DMA 初始化； 3. 使能中断； 4. 使能 DMA 通道以及触发 SPI； 5. 重设目的地址，开始下一次传输。
预期结果	1. 打印” 0x82131415”, “0x86171819”, “0x80212223” , “0x84252627”。
注意点	

17.7 SPI 搬移到 UART

栏目	描述
工程名	DMA_SPI_to_UART
功能	演示 SPI 搬移到 UART
硬件环境	
步骤	1. SPI 初始化； 2. UART 初始化； 3. DMA 初始化； 4. 使能中断； 5. 使能 DMA 通道以及触发 SPI。
预期结果	1. 打印” 0x82131415”。
注意点	

17.8 UART 搬移到 Memory

栏目	描述
工程名	DMA_UART_to_Memory
功能	演示 UART 搬移到 Memory
硬件环境	
步骤	1. SPI 初始化； 2. DMA 初始化； 3. 使能中断；

©2026 旋智电子科技（上海）有限公司

栏目	描述
	4. 使能 DMA 通道以及触发 SPI； 5. 重设目的地址，开始下一次传输。
预期结果	1. 打印” 0x82131415”。
注意点	

17.9 UART 搬移到 SPI

栏目	描述
工程名	DMA_UART_to_SPI
功能	演示 UART 搬移到 SPI
硬件环境	
步骤	1. UART 初始化； 2. SPI 初始化； 3. DMA 初始化； 4. 使能中断； 5. 使能 DMA 通道以及触发 UART。
预期结果	1. 打印” 0x82131415”。
注意点	

18 CAN

18.1 CAN 接收标准帧-阻塞方式

栏目	描述
工程名	CAN_Standard_Frame_RX
功能	演示 CAN 使用阻塞方式接收标准的远程帧和数据帧
硬件环境	
步骤	1. 配置 CAN GPIO 引脚 2. 使能 CAN 协议，设置 CAN 的采样点为 80%、速率为 1M 3. 等待接收 CAN 协议的标准格式的远程帧 4. 等待接收 CAN 协议的标准格式的数据帧 5. 校验接收数据帧的数据
预期结果	发送的数据和接收的数据一致
注意点	

©2026 旋智电子科技（上海）有限公司

18.2 CAN 发送标准帧-阻塞方式

栏目	描述
工程名	CAN_Standard_Frame_TX
功能	演示 CAN 使用阻塞方式发送标准的远程帧和数据帧
硬件环境	
步骤	1. 配置 CAN GPIO 引脚 2. 使能 CAN 协议，设置 CAN 的采样点为 80%、速率为 1M 3. 配置 CAN 协议的标准远程帧 4. 等待 CAN 协议的标准格式的远程帧发送完 5. 配置 CAN 协议的标准数据帧 6. 等待 CAN 协议的标准格式的数据帧发送完
预期结果	发送的数据和接收的数据一致
注意点	

18.3 CANFD 接收标准帧-阻塞方式

栏目	描述
工程名	CAN_Standard_Frame_RX
功能	演示 CAN 使用阻塞方式接收标准的数据帧
硬件环境	
步骤	1. 配置 CAN GPIO 引脚 2. 使能 CANFD 协议，设置 CAN 的仲裁域采样点为 80%、速率为 1M，数据域采样点为 90%、速率为 5M 3. 等待接收 CANFD 协议的标准格式的数据帧 4. 校验接收数据帧的数据
预期结果	发送的数据和接收的数据一致
注意点	

18.4 CANFD 发送标准帧-阻塞方式

栏目	描述
工程名	CAN_Standard_Frame_TX
功能	演示 CAN 使用阻塞方式发送标准的数据帧
硬件环境	

©2026 旋智电子科技（上海）有限公司

栏目	描述
步骤	1. 配置 CAN GPIO 引脚 2. 使能 CANFD 协议，设置 CAN 的仲裁域采样点为 80%、速率为 1M，数据域采样点为 90%、速率为 5M 3. 配置 CAN 协议的标准数据帧 4. 等待 CAN 协议的标准格式的数据帧发送完
预期结果	发送的数据和接收的数据一致
注意点	

18.5 CAN 接收标准帧-非阻塞中断方式

栏目	描述
工程名	CAN_Int_RX
功能	演示 CAN 使用中断方式接收标准的数据帧
硬件环境	
步骤	1. 配置 CAN GPIO 引脚 2. 使能 CAN 协议，设置 CAN 的采样点为 80%、速率为 1M 3. 配置接收 CAN 协议的标准格式的数据帧 4. 使能邮箱中断 5. 等待数据帧接收完成，校验数据
预期结果	发送的数据和接收的数据一致
注意点	

18.6 CAN 发送标准帧-非阻塞中断方式

栏目	描述
工程名	CAN_Int_TX
功能	演示 CAN 使用中断方式发送标准的数据帧
硬件环境	
步骤	1. 配置 CAN GPIO 引脚 2. 使能 CAN 协议，设置 CAN 的采样点为 80%、速率为 1M 3. 配置发送 CAN 协议的标准格式的数据帧 4. 使能邮箱中断 5. 等待数据帧发送完成，再填充下一帧数据
预期结果	发送的数据和接收的数据一致

©2026 旋智电子科技（上海）有限公司

栏目	描述
注意点	

19 Multi-core

19.1 CPU0 启动 CPU1、CPU2、CPU3

栏目	描述
工程名	CPU0_Start_OtherCPU
功能	演示 CPU0 启动 CPU1、CPU2、CPU3
硬件环境	
步骤	1. 先将 CPU1、CPU2、CPU3 的程序下载到对应的 Flash 2. 使能 CPU1、CPU2、CPU3 3. 获取编号 0 的互斥锁 4. 访问 0x31000000 地址资源 5. 释放编号 0 的互斥锁
预期结果	1. CPU1、CPU2、CPU3 开始运行程序 2. 访问 0x31000000 地址资源互斥
注意点	

19.2 CPU1 触发 WDT 中断

栏目	描述
工程名	CPU1_WTD_Int
功能	演示 CPU1 触发 WDT 中断
硬件环境	
步骤	1. 更新时钟信息，确保 CPU1 时钟相关的变量正常 2. 初始化 WDT 3. 使能 WDT 中断 4. 获取编号 0 的互斥锁 5. 访问 0x31000000 地址资源 6. 释放编号 0 的互斥锁
预期结果	1. CPU1 触发 WDT 中断 2. 访问 0x31000000 地址资源互斥

©2026 旋智电子科技（上海）有限公司

栏目	描述
注意点	

19.3 CPU2 触发 WDT 中断

栏目	描述
工程名	CPU2_WTD_Int
功能	演示 CPU2 触发 WDT 中断
硬件环境	
步骤	1. 更新时钟信息，确保 CPU2 时钟相关的变量正常 2. 初始化 WDT 3. 使能 WDT 中断 4. 获取编号 0 的互斥锁 5. 访问 0x31000000 地址资源 6. 释放编号 0 的互斥锁
预期结果	1. CPU2 触发 WDT 中断 2. 访问 0x31000000 地址资源互斥
注意点	

19.4 CPU3 触发 WDT 中断

栏目	描述
工程名	CPU3_WTD_Int
功能	演示 CPU3 触发 WDT 中断
硬件环境	
步骤	1. 更新时钟信息，确保 CPU3 时钟相关的变量正常 2. 初始化 WDT 3. 使能 WDT 中断 4. 获取编号 0 的互斥锁 5. 访问 0x31000000 地址资源 6. 释放编号 0 的互斥锁
预期结果	1. CPU3 发 WDT 中断 2. 访问 0x31000000 地址资源互斥
注意点	

©2026 旋智电子科技（上海）有限公司

20 SDFM

20.1 SDFM 并行采样 ADC 数据

栏目	描述
工程名	SDFM_Parallel_Filter_Int
功能	演示 SDFM 并行采样 ADC 数据
硬件环境	
步骤	1. 配置 ADC 进行采样 GPIO0 输入的正玄波 2. 配置 DMA 将 ADC 采样数据搬运到 SDFM->SDFMDIN 对应的寄存器 3. 初始化 SDFM 为并行模式（滤波类型为 SINC2，过采样率为 16） 4. 使能 SDFM 通道滤波器 5. 使能 DAM 传输完成中断和 SDFM 数据 Ready 中断 6. 使能 SDFM 通道 7. 软件触发 ADC 采样 8. DMA 传输完成后重新触发 ADC 采样并进行 DMA 传输
预期结果	SDFM 输出过滤后的电压也是正玄波
注意点	

20.2 SDFM 串行采样数据

栏目	描述
工程名	SDFM_Serial_Filter_Int
功能	演示 SDFM 串行采样 GPIO1 输出的正玄波数据
硬件环境	
步骤	1. 配置 SDFM 为串行模式（上升沿采样数据，SCK 时钟和数据码率相等） 2. 配置 SDFM 通道 0 使用通道 0 的时钟 3. 使能 SDFM 中断 4. 使能 SDFM 通道 5. 获取 SDFM 采样的数据
预期结果	SDFM 输出过滤后的数据也是正玄波
注意点	1. SDFM 输入的数据码率和 SCK 时钟频率需要相当 2. SDFM 上升沿进行采样数据

©2026 旋智电子科技（上海）有限公司