栏目	描述
软件 IDE	1. KEIL MDK523 2. IAR EWARM-CD-8324-20889
调试器	JLink_Windows_V614b

SPINTROL

目录

版本历史 5

2 FLASH 6

2.1 FLASH 基础操作 6

2.2 扇区写/擦保护 6

2.3 在 RAM 中运行代码及数据 6

3 MCU Power 8

3.1 MCU 电源过压/欠压触发中断 8

3.2 MCU 电源过压/欠压触发复位 8

4 Clock 9

4.1 时钟错误触发中断 9

4.2 时钟错误触发复位 9

4.3 PLL 时钟源选择 XO 或者 RCO 9

5 DMA 10

5.1 Memory 搬移到 Memory 10

5.2 Memory 搬移到 Memory（使用连续模式进行多块传输） 10

5.3 Memory 搬移到 Memory（使用链表模式进行多块传输） 11

5.4 Memory 搬移到 Memory（使用自动加载模式进行多块传输） 11

5.5 Memory 搬移到 SPI 12

5.6 Memory 搬移到 UART 12

5.7 SPI 搬移到 Memory 13

5.8 UART 搬移到 Memory 13

6 ADC 14

6.1 ADC 开路检测 14

6.2 ADC 短路检测 14

6.3 ADC 双端采样-软件触发 15

6.4 ADC 单端采样 -定时器触发 15

6.5 ADC 同时采样 15

6.6 ADC 单端采样-软件触发 16

6.7 ADC 增益及偏移校准 16

7 TSENSER 16

7.1 测量芯片内部温度 16

8 PGA 17

SPINTROL

©2026 旋智电子科技（上海）有限公司

8.1 PGA 差分放大 17
8.2 PGA 单端放大 17
9 PWM 18
9.1 PWM 输出互补波形 18
9.2 非级联同步方式设置 PWM 相位差-GPIO 触发 18
9.3 PWM 触发 ADC 采集 19
9.4 PWM 输出单通道波形 19
9.5 非级联同步方式设置 PWM 相位差-软件触发 20
9.6 非级联同步方式设置 PWM 相位差-Timer 触发 20
9.7 PWM 信号封锁-GPIO 触发 21
9.8 PWM 输出三相波形 22
10 ECAP 22
10.1 ECAP APWM 模式 22
10.2 ECAP 双通道持续捕获 23
10.3 ECAP 四通道持续捕获 23
10.4 ECAP 单通道一次性捕获 23
11 EQEP 24
11.1 M 法测速 24
11.2 T 法测速 24
12 CAU 25
12.1 CAU 计算 25
13 Timer 25
13.1 Timer 事件计数器 25
13.2 Timer 门控计数器 26
13.3 Timer 触发中断 26
14 WDT 27
14.1 WDT 喂狗 27
14.2 WDT 触发中断 27
15 CRC 27
15.1 CRC 正常工作检测 27
16 GPIO 28
16.1 GPIO 电平翻转检测 28
16.2 GPIO 电平高低检测 28
16.3 GPIO 触发不可屏蔽中断 29

SPINTROL

17 COMP 29
17.1 UART 自动波特率 29
18 UART 29
18.1 UART 自动波特率 29
18.2 UART 中断接收和发送 30
18.3 UART 轮询接收和发送 30
19 SPI 31
19.1 SPI 主机模式-阻塞方式 31
19.2 SPI 从机模式-阻塞方式 31
19.3 SPI 主机模式-非阻塞中断方式 31
19.4 SPI 从机模式-非阻塞中断方式 32
20 I2C 32
20.1 I2C 主机模式-阻塞方式 32
20.2 I2C 从机模式-阻塞方式 33
20.3 I2C 主机模式-非阻塞中断方式 33
20.4 I2C 从机模式-非阻塞中断方式 34
21 CAN 34
21.1 CAN 接收-中断方式 34
21.2 CAN 发送-中断方式 34
21.3 CAN 接收-阻塞方式 35
21.4 CAN 发送-阻塞方式 35
21.5 CANFD 接收-中断方式 36
21.6 CANFD 发送-中断方式 36
21.7 CANFD 接收-阻塞方式 36
21.8 CANFD 发送-阻塞方式 37

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	2025-03-11	HangSu	Released	1. 首次发布

SPINTROL

2 FLASH

2.1 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	

2.2 扇区写/擦保护

栏目	描述
工程名	Flash_Sector_Protect
功能	设置 flash 的某一扇区（sector）为保护模式，对此扇区的写入及擦除操作将失败
硬件环境	
步骤	1. 对测试扇区进行擦，写，读测试； 1. 对测试扇区开写保护； 2. 对测试扇区进行擦，写，读测试。
预期结果	1. 设置写保护之前，打印 Flash 操作成功； 2. 设置写保护之后，打印 Flash 操作失败。
注意点	

2.3 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“gu32aRAMVector”数组； 2. 设置中断向量表地址为“gu32aRAMVector”数组的首地址； 3. 初始化 Timer； 4. 使能 Timer 中断，创建 Timer 中断服务函数并用“_RAM_FUNC_”前缀修饰； 5. 使能 Timer；

©2026 旋智电子科技（上海）有限公司

栏目	描述
	6. 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	“gu32aRAMVector”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）

SPINTROL

3 MCU Power

3.1 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 2.79V 时，欠压事件触发；3.03V 时，欠压事件消失； 设置 VDD3.3V 电源域的过压检测，设置为 3.66V 时，过压事件触发；3.53V 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 2.79V 时，打印 VDD33 触发欠压中断，直至调整电源高于 3.03V 时，打印消失； 2. 微调可调电源，当高于 3.66V 时，打印 VDD33 触发过压中断，直至调整电源低于 3.53V 时，打印消失。
注意点	3. 想通过可调电源触发过压中断时，务必进行微调，高出 3.66V 即可（不可高出 4.24V），防止烧毁芯片。

3.2 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 2.79V 时，欠压事件触发；3.03V 时，欠压事件消失。 设置 VDD3.3V 电源域的过压检测，设置为 3.66V 时，过压事件触发；3.53V 时，过压事件消失； 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 2.79V 时，打印 VDD33 触发欠压复位；直至调整电源高于 3.03V 时，复位消失。 2. 微调可调电源，当高于 3.66V 时，打印 VDD33 触发过压复位，直至调整电源低于 3.53V 时，复位消失。
注意点	3. 想通过可调电源触发过压复位时，务必进行微调，高出 3.66V 即可（不可高出 4.24V），防止烧毁芯片。

©2026 旋智电子科技（上海）有限公司

4 Clock

4.1 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 80MHz，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	

4.2 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 80MHz，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	

4.3 PLL 时钟源选择 XO 或者 RCO

栏目	描述
工程名	Clock_Use_XO_Or_RCO
功能	PLL 时钟源选择 XO 或者 RCO
硬件环境	
步骤	1. 根据用户配置进行选用 XO 或者 RCO 作为 PLL 时钟源进行初始化时钟。 2. 打印时钟配置状态 3. 将时钟源的时钟信号输出到引脚

©2026 旋智电子科技（上海）有限公司

栏目	描述
预期结果	1. 运行程序后打印正常。
注意点	

5 DMA

5.1 Memory 搬移到 Memory

栏目	描述
工程名	DMA_MemtoMem
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； 4. 随机设置源端数据； 5. 使能 DMA 开始传输； 6. 传输完成后，进行比对目的端的数据和源端数据是否一致； 7. 重新设置源端和目的端的地址，开始下一次传输。
预期结果	1. DMA 传输完成，传输数据比对一致
注意点	

5.2 Memory 搬移到 Memory（使用连续模式进行多块传输）

栏目	描述
工程名	DMA_MemToMemWithContiguous
功能	演示使用连续模式进行多块传输
硬件环境	
步骤	1. 设置 block0/block1/block2 源端为连续模式和目的端为链表模式； 2. 设置 DMA 传输的通道为通道 7； 3. 设置 block0 的通道配置； 4. 设置 link 指针的地址为 block0 的目的地址； 5. 使能 DMA 传输完成中断； 6. 使能 DMA 模块； 7. 随机设置源端数据； 8. 传输完成后，进行比对目的端的数据和源端数据是否一致； 9. 重新设置源端和目的端的地址，开始下一次传输。
预期结果	1. DMA 传输完成，传输数据比对一致

©2026 旋智电子科技（上海）有限公司

栏目	描述
注意点	

5.3 Memory 搬移到 Memory（使用链表模式进行多块传输）

栏目	描述
工程名	DMA_MemToMemWithLinkedList
功能	演示使用链表模式进行多块传输
硬件环境	
步骤	1. 设置 block0/block1/block2 源端为链表模式和目的端为链表模式； 2. 设置 DMA 传输的通道为通道 7； 3. 设置 block0 的通道配置； 4. 设置 link 指针的地址为 block0 的目的地址； 5. 使能 DMA 传输完成中断； 6. 使能 DMA 模块； 7. 随机设置源端数据； 8. 传输完成后，进行比对目的端的数据和源端数据是否一致； 9. 重新设置源端和目的端的地址，开始下一次传输。
预期结果	1. DMA 传输完成，传输数据比对一致
注意点	

5.4 Memory 搬移到 Memory（使用自动加载模式进行多块传输）

栏目	描述
工程名	DMA_UartToMemWithAutoReload
功能	演示使用自动加载模式进行多块传输
硬件环境	
步骤	1. 设置 block0/block1/block2 源端为自动加载模式和目的端为链表模式； 2. 设置 DMA 传输的通道为通道 7； 3. 设置 block0 的通道配置； 4. 设置 link 指针的地址为 block0 的目的地址； 5. 使能 DMA 传输完成中断； 6. 使能 DMA 模块； 7. 随机设置源端数据； 8. 传输完成后，进行比对目的端的数据和源端数据是否一致； 9. 重新设置源端和目的端的地址，开始下一次传输。

©2026 旋智电子科技（上海）有限公司

栏目	描述
预期结果	1. DMA 传输完成，传输数据比对一致
注意点	

5.5 Memory 搬移到 SPI

栏目	描述
工程名	DMA_MemToSpi
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. 随机设置源端数据； 6. 使能 DMA 开始传输； 7. 传输完成后，进行比对目的端的数据和源端数据是否一致； 8. 重新设置源端和目的端的地址，开始下一次传输。
预期结果	1. DMA 传输完成，传输数据比对一致
注意点	

5.6 Memory 搬移到 UART

栏目	描述
工程名	DMA_MemToUart
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. 随机设置源端数据； 6. 使能 DMA 开始传输； 7. 传输完成后，进行比对目的端的数据和源端数据是否一致； 8. 重新设置源端和目的端的地址，开始下一次传输。
预期结果	1. DMA 传输完成，传输数据比对一致
注意点	

©2026 旋智电子科技（上海）有限公司

5.7 SPI 搬移到 Memory

栏目	描述
工程名	DMA_SpiToMem
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； 5. 随机设置源端数据； 6. 使能 DMA 开始传输； 7. 传输完成后，进行比对目的端的数据和源端数据是否一致； 8. 重新设置源端和目的端的地址，开始下一次传输。
预期结果	1. DMA 传输完成，传输数据比对一致
注意点	

5.8 UART 搬移到 Memory

栏目	描述
工程名	DMA_UartToMem
功能	演示 UART 搬移到 Memory
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. 随机设置源端数据； 6. 使能 DMA 开始传输； 7. 传输完成后，进行比对目的端的数据和源端数据是否一致； 8. 重新设置源端和目的端的地址，开始下一次传输。
预期结果	1. DMA 传输完成，传输数据比对一致
注意点	

SPINTROL

6 ADC

6.1 ADC 开路检测

栏目	描述
工程名	ADC_DetectOpenCircuit
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	

6.2 ADC 短路检测

栏目	描述
工程名	ADC_DetectShortCircuit
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 200 内，那么输入引脚短路。
注意点	

©2026 旋智电子科技（上海）有限公司

6.3 ADC 双端采样-软件触发

栏目	描述
工程名	ADC_DifferentialResult
功能	设置 ADC 工作在双端采样。
硬件环境	芯片 ANA2、ANA3 引脚可外接。
步骤	1. 初始化 ADC 通道，P 端配置为 ANA2，N 端配置 ANA3； 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 ANA2、ANA3 压差值。
注意点	1. 向 ANA2、ANA3 施加的电压值控制在 0~3.3V 范围内。

6.4 ADC 单端采样 -定时器触发

栏目	描述
工程名	ADC_RequestByTIMER
功能	通过定时器触发 ADC 采样
硬件环境	芯片 ANA2 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 ANA2，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 ANA2 施加的电压。
注意点	1. 向 ANA2 施加的电压值控制在 0~3.3V 范围内。

6.5 ADC 同时采样

栏目	描述
工程名	ADC_Simultaneous
功能	设置 ADC SH0 与 SH1 同时采样，并获取结果。
硬件环境	芯片 ANA2、ANA4 引脚可外接
步骤	1. 配置 SH0 与 SH1 同时采样； 2. 使能 ADC 中断； 3. 使能 ADC； 4. 软件触发 ADC 采样
预期结果	1. 循环打印向 ANA2、ANA4 施加的电压值。
注意点	1. 向 ANA2、ANA4 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

6.6 ADC 单端采样-软件触发

栏目	描述
工程名	ADC_SingleEndedResult
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 ANA2 引脚可外接。
步骤	1. 初始化 ADC 通道，P 端配置为 ANA2，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 ANA2 施加的电压值。
注意点	1. 向 ANA2 施加的电压值控制在 0~3.3V 范围内。

6.7 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 GPIO0、GPIO1 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	

7 TSENSER

7.1 测量芯片内部温度

栏目	描述
工程名	Tsensor
功能	测量芯片内部温度
硬件环境	
步骤	1. 获取芯片内部温度；
预期结果	1. 与环境温度基本一致。

©2026 旋智电子科技（上海）有限公司

栏目	描述
注意点	

8 PGA

8.1 PGA 差分放大

栏目	描述
工程名	Differential_PGA_To_ADC
功能	ADC 采样 PAG 差分放大数据
硬件环境	
步骤	1. 初始化 PGA（采样差分电机模式，增益为 2）； 2. 使能 PGA 模块； 3. 初始化 ADC 进行采样 PGA 两端电压； 4. 使能 ADC 模块； 5. 获取 ADC 采样结果。
预期结果	1. ADC 采样的电压为 PGA 输入的 2 倍。
注意点	

8.2 PGA 单端放大

栏目	描述
工程名	Single_Ended_PGA_To_ADC
功能	ADC 采样 PAG 单端放大数据
硬件环境	
步骤	1. 初始化 PGA（采样单端模式，增益为 2）； 2. 使能 PGA 模块； 3. 初始化 ADC 进行采样 PGA 单端电压； 4. 使能 ADC 模块； 5. 获取 ADC 采样结果。
预期结果	6. ADC 采样的电压为 PGA 输入的 2 倍。
注意点	

SPINTROL

9 PWM

9.1 PWM 输出互补波形

栏目	描述
工程名	PWM_ComplementaryPairChannel
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_GPIOSYNCI
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%； 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；

©2026 旋智电子科技（上海）有限公司

栏目	描述
	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 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_GPIO8，可观测到 ADC 采集触发信号。
注意点	

9.4 PWM 输出单通道波形

栏目	描述
工程名	PWM_SingleChannel
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。

©2026 旋智电子科技（上海）有限公司

栏目	描述
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	

9.5 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。
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	

9.6 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

©2026 旋智电子科技（上海）有限公司

栏目	描述
步骤	1. 配置 TIMER 定时器 1000ms 触发 PWM 同步事件。 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	

9.7 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。

©2026 旋智电子科技（上海）有限公司

栏目	描述
注意点	

9.8 PWM 输出三相波形

栏目	描述
工程名	PWM_TripPhaseWaveform
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	

10 ECAP

10.1 ECAP APWM 模式

栏目	描述
工程名	ECAP_APWMFunction
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； 2. 配置 PWM 频率（10kHz）、PWM 占空比（50%）； 3. 运行 ECAP
预期结果	1. 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	

©2026 旋智电子科技（上海）有限公司

10.2 ECAP 双通道持续捕获

栏目	描述
工程名	ECAP_ContinueDule
功能	设置 ECAP 工作在连续采样的 Continue 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. 配置为，CAP0 捕获上升沿、CAP1 捕获上升沿沿、CPA2 捕获下降沿、CAP3 捕获下降沿

10.3 ECAP 四通道持续捕获

栏目	描述
工程名	ECAP_ContinueQuad
功能	设置 ECAP 工作在连续采样的 Continue 模式，并连续对 GPIO 输入引脚进行多次事件采样，以此连续计算输入信号之间的相位差
硬件环境	
步骤	1. 配置 ECAP； 2. 配置输入滤波； 3. 配置 ECAP_CEVT3 中断； 4. 使能 ECAP。 5. 依次 ECAP 捕获的引脚输入低电平和高电平
预期结果	1. 持续捕获的相位差
注意点	1. 配置为，CAP0 捕获上升沿、CAP1 捕获上升沿、CPA2 捕获上升沿、CAP3 捕获上升沿

10.4 ECAP 单通道一次性捕获

栏目	描述
工程名	ECAP_OneshotSingle
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；

©2026 旋智电子科技（上海）有限公司

栏目	描述
	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. 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

11 EQEP

11.1 M 法测速

栏目	描述
工程名	EQEP_Quadrature_Counting_Mode_M
功能	使用 M 法测量编码器转速
硬件环境	
步骤	1. 初始化 EQEP（采用正交编码模式）； 2. 配置定时器的加载值； 3. 使能定时器超时中断； 4. 使能定时器单元； 5. 使能 EQEP；
预期结果	1. 打印输出 EQEP 测量的转速。
注意点	

11.2 T 法测速

栏目	描述
工程名	EQEP_Quadrature_Counting_Mode_T
功能	使用 T 法测量编码器转速
硬件环境	
步骤	1. 初始化 EQEP（采用正交编码模式）； 2. 配置事件分频比； 3. 使能捕获事件中断； 4. 使能捕获单元； 5. 使能 EQEP；
预期结果	1. 打印输出 EQEP 测量的转速。

©2026 旋智电子科技（上海）有限公司

栏目	描述
注意点	

12 CAU

12.1 CAU 计算

栏目	描述
工程名	CAU
功能	CAU 读写访问/加法/减法/乘法计算
硬件环境	
步骤	1. 初始化 CAU； 2. 配置 CAU 任务的入口地址； 3. 使能 CAU 中断； 4. 使能 CAU； 5. 设置 CAU 任务的各参数； 6. 获取任务计算结果。
预期结果	1. CAU 读写访问正常，计算正常。
注意点	

13 Timer

13.1 Timer 事件计数器

栏目	描述
工程名	TIMER_EventCounter
功能	外部输入引脚当作计数事件，触发 TIMER 超时
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. 初始化 Timer，配置 DEMO_GPIO_INPUT 脚为外部输入引脚，计数模式为事件计数器并重新配置事件加载值； 4. 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； 5. 使能 Timer； 6. 周期性翻转 DEMO_GPIO_OUTPUT 脚 100 次触发 Timer 超时中断。

©2026 旋智电子科技（上海）有限公司

栏目	描述
预期结果	1. DEMO_GPIO_OUTPUT 脚翻转 100 次触发一次 Timer 超时中断。
注意点	

13.2 Timer 门控计数器

栏目	描述
工程名	TIMER_GatedTIMER
功能	外部输入引脚当作 Timer 使能信号
硬件环境	
步骤	1. 配置 DEMO_GPIO_INPUT 脚为输入； 2. 初始化 Timer，配置 DEMO_GPIO_INPUT 脚为外部输入引脚，高电平有效，并配置 Timer 模式为门控定时器； 3. 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； 4. 使能 Timer。
预期结果	1. 程序运行后，当 DEMO_GPIO_INPUT 脚输入高电平时，使能 Timer 计数，达到触发时间后触发超时中断； 2. 当 DEMO_GPIO_INPUT 脚输入低电平时，不使能 Timer 计数，不触发超时中断。
注意点	

13.3 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	

©2026 旋智电子科技（上海）有限公司

14 WDT

14.1 WDT 喂狗

栏目	描述
工程名	WDT_FeedDog
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	

14.2 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。
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	

15 CRC

15.1 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

©2026 旋智电子科技（上海）有限公司

栏目	描述
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	

16 GPIO

16.1 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。
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	

16.2 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	

©2026 旋智电子科技（上海）有限公司

16.3 GPIO 触发不可屏蔽中断

栏目	描述
工程名	GPIO_TriggerNMI
功能	触发不可屏蔽中断
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. 设置 DEMO_GPIO_INPUT 低电平触发不可屏蔽中断
预期结果	1. 程序运行后，每隔 500ms 触发不可屏蔽中断
注意点	

17 COMP

17.1 UART 自动波特率

栏目	描述
工程名	Comparator
功能	此示例为 COMP 进行电压比较
硬件环境	
步骤	1. 配置 DAC0 输出 2000mv； 2. 使能 DAC0； 3. 初始化 COMP； 4. 使能 COMP 电压过压和过低中断。 5. 使能 COMP。
预期结果	1. COMP 正端高于 2000mv 触发电压过高中断，低于 2000mv 触发电压过低中断。
注意点	

18 UART

18.1 UART 自动波特率

栏目	描述
工程名	UART_AutoBaudrate
功能	此示例为 UART 自动设置波特率

©2026 旋智电子科技（上海）有限公司

栏目	描述
硬件环境	
步骤	2. 配置 UART1 引脚； 3. 开始自动波特率检测； 4. 等待自动波特率标志位起来； 5. 检测接受到的数据是否符合预期。
预期结果	1. 上位机向 UART1 发送完 0x55 后，UART0 打印”Auto-baud success!”。
注意点	

18.2 UART 中断接收和发送

栏目	描述
工程名	UART_IntRxTx
功能	此示例为 UART 进行中断接收和发送数据
硬件环境	
步骤	1. 配置 UART1 引脚； 2. 设置接收 FIFO 阈值； 3. 使能接收请求中断和接收超时中断； 4. 使能 UART1。 5. 中断进行接收数据，并将接收的数据进行发送。
预期结果	1. 发送的数据和接收数据一致。
注意点	

18.3 UART 轮询接收和发送

栏目	描述
工程名	UART_PollRxTx
功能	此示例为 UART 进行轮询接收和发送数据
硬件环境	
步骤	1. 配置 UART1 引脚； 2. 使能 UART1。 3. 轮询进行接收数据，并将接收的数据进行发送。
预期结果	1. 发送的数据和接收数据一致。
注意点	

©2026 旋智电子科技（上海）有限公司

19 SPI

19.1 SPI 主机模式-阻塞方式

栏目	描述
工程名	SPI_MasterPollingTxRx
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	

19.2 SPI 从机模式-阻塞方式

栏目	描述
工程名	SPI_SlavePollingTxRx
功能	与“SPI_MasterPollingTxRx”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。
预期结果	1. 串口打印操作成功。
注意点	

19.3 SPI 主机模式-非阻塞中断方式

栏目	描述
工程名	SPI_MasterIntTxRx
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

©2026 旋智电子科技（上海）有限公司

栏目	描述
	4. 使能 SPI。 5. 通过阻塞方式发送，“数据发送”命令。 6. 中断方式向从机设备发送数据。 7. 通过阻塞方式发送，“数据接收”命令 8. 中断方式接收从机设备的数据。 9. 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	

19.4 SPI 从机模式-非阻塞中断方式

栏目	描述
工程名	SPI _SlaveTxRxINT
功能	与“SPI_MasterTxRxINT”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. 接收到主机发送命令后，通过中断方式进行数据接收。 6. 校验接收到数据 CRC 是否正确。 7. 接收到主机接收命令后，通过中断方式将接收到数据发回主机。
预期结果	1. 串口打印接收到主机设备数据，CRC 校验码正确。
注意点	

20 I2C

20.1 I2C 主机模式-阻塞方式

栏目	描述
工程名	I2C_MasterPollingTxRx
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。

©2026 旋智电子科技（上海）有限公司

栏目	描述
	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	

20.2 I2C 从机模式-阻塞方式

栏目	描述
工程名	I2C_SlavePollingTxRx
功能	与“I2C_MasterPollingTxRx”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	

20.3 I2C 主机模式-非阻塞中断方式

栏目	描述
工程名	I2C_MasterIntTxRx
功能	设置 I2C 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 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 NVIC 中断。 4. 使能 I2C。 5. 中断方式向从机设备发送数据。 6. 中断方式接收从机设备数据。 7. 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	

SPINTROL

20.4 I2C 从机模式-非阻塞中断方式

栏目	描述
工程名	I2C_SlaveIntTxRx
功能	与“I2C_MasterINTTxRx”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 I2C0 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 NVIC 中断。 4. 使能 I2C。 5. 中断方式接收主机设备数据。 6. 检查接收到的主机设备数据，CRC 校验码是否正确。 7. 如 CRC 校验码正确，则将接收到的数据，通过中断方式发送回主机设备。
预期结果	1. 串口打印接收到主机设备数据，CRC 校验码正确。
注意点	

21 CAN

21.1 CAN 接收-中断方式

栏目	描述
工程名	CAN_IntRx
功能	CAN 中断接收数据。
硬件环境	
步骤	1. 初始化 CAN（波特率 1M，采样点 75%）。 2. 使能 CAN。 3. 配置邮箱。 4. 使能邮箱。 5. 接收数据完成后，再重新接收数据。
预期结果	1. 接收的数据和发送的数据一致。
注意点	

21.2 CAN 发送-中断方式

栏目	描述
工程名	CAN_IntTx
功能	CAN 中断发送数据。

©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 CAN（波特率 1M，采样点 75%）。 2. 使能 CAN。 3. 配置邮箱。 4. 使能邮箱。 5. 发送数据完成后，再重新发送数据。
预期结果	1. 接收的数据和发送的数据一致。
注意点	

21.3 CAN 接收-阻塞方式

栏目	描述
工程名	CAN_PollingRx
功能	CAN 阻塞接收数据。
硬件环境	
步骤	1. 初始化 CAN（波特率 1M，采样点 75%）。 2. 使能 CAN。 3. 配置邮箱。 4. 使能邮箱。 5. 接收数据完成后，再重新接收数据。
预期结果	1. 接收的数据和发送的数据一致。
注意点	

21.4 CAN 发送-阻塞方式

栏目	描述
工程名	CAN_PollingTx
功能	CAN 阻塞发送数据。
硬件环境	
步骤	1. 初始化 CAN（波特率 1M，采样点 75%）。 2. 使能 CAN。 3. 配置邮箱。 4. 使能邮箱。 5. 发送数据完成后，再重新发送数据。
预期结果	1. 接收的数据和发送的数据一致。
注意点	

©2026 旋智电子科技（上海）有限公司

21.5 CANFD 接收-中断方式

栏目	描述
工程名	CANFD_IntRx
功能	CANFD 中断接收数据。
硬件环境	
步骤	1. 初始化 CANFD（仲裁阶段，波特率 1M，采样点 75%；数据阶段，波特率 2M，采样点 80%）。 2. 使能 CANFD。 3. 配置邮箱。 4. 使能邮箱。 5. 接收数据完成后，再重新接收数据。
预期结果	1. 接收的数据和发送的数据一致。
注意点	

21.6 CANFD 发送-中断方式

栏目	描述
工程名	CANFD_IntTx
功能	CANFD 中断发送数据。
硬件环境	
步骤	1. 初始化 CANFD（仲裁阶段，波特率 1M，采样点 75%；数据阶段，波特率 2M，采样点 80%）。 2. 使能 CANFD。 3. 配置邮箱。 4. 使能邮箱。 5. 发送数据完成后，再重新发送数据。
预期结果	1. 接收的数据和发送的数据一致。
注意点	

21.7 CANFD 接收-阻塞方式

栏目	描述
工程名	CANFD_PollingRx
功能	CANFD 阻塞接收数据。
硬件环境	

©2026 旋智电子科技（上海）有限公司

栏目	描述
步骤	1. 初始化 CANFD（仲裁阶段，波特率 1M，采样点 75%；数据阶段，波特率 2M，采样点 80%）。 2. 使能 CANFD。 3. 配置邮箱。 4. 使能邮箱。 5. 接收数据完成后，再重新接收数据。
预期结果	1. 接收的数据和发送的数据一致。
注意点	

21.8 CANFD 发送-阻塞方式

栏目	描述
工程名	CANFD_PollingTx
功能	CANFD 阻塞发送数据。
硬件环境	
步骤	1. 初始化 CANFD（仲裁阶段，波特率 1M，采样点 75%；数据阶段，波特率 2M，采样点 80%）。 2. 使能 CANFD。 3. 配置邮箱。 4. 使能邮箱。 5. 发送数据完成后，再重新发送数据。
预期结果	1. 接收的数据和发送的数据一致。
注意点	

SPINTROL