



# 32 位 ARM® Cortex®-M0+ 微控制器 JS32L011 系列

注意事项

V1.0

珠海巨晟科技股份有限公司

地 址: 广东省珠海市高新区金唐路 1 号港湾 1 号湾 8 栋 4 楼

电 话: 0756-3335384      传 真: 0756-3335384

网 站: [www.honor-ic.com](http://www.honor-ic.com)      邮 编: 519080



## 版本历史

变更类型：A - 增加 M - 修订 D - 删除

变更版本号	日期	变更类型	修改人	审核	摘要
V1.0					正式版本

## 版权声明

1、本资料是为了让用户根据自身需求选择合适的产品而提供的参考资料，相关的知识产权属于珠海巨晟科技股份有限公司或来自第三方的合法授权；提供上述资料不构成对相关知识产权的许可或转让，未经珠海巨晟科技股份有限公司的许可，任何人不得翻印或者复制本资料的全部或部分内容。

2、在使用本资料所记载的信息并对有关产品是否适用做出最终判断前，请您务必将所有信息作为一个整体来评价。对于本资料所记载的信息使用不当而引起的任何损失，珠海巨晟科技股份有限公司概不负责。

3、本资料所记载的产品会持续更新迭代并发布，在购买本资料所记载的产品时，请预先向珠海巨晟科技股份有限公司确认最新信息，并请您通过公司网站、微信公众号等各种方式关注珠海巨晟科技股份有限公司公布的信息，相关更新恕不另行通知。

4、如果您需要进一步了解有关本资料所记载的信息或产品的详情，请与珠海巨晟科技股份有限公司的技术服务部门联系，我们会为您提供全方位的技术支持。

## 目录

<b>1. 适用性</b> .....	<b>1</b>
<b>2. 注意事项摘要</b> .....	<b>1</b>
<b>3. 注意事项描述</b> .....	<b>1</b>
3.1. 内核.....	1
3.2. Flash.....	1
3.3. GPIO.....	2
3.4. I2C.....	2
3.5. USART & LPUART.....	3
3.6. TIM1.....	3
3.7. LPTIM.....	4
3.8. ADC.....	4

## 1. 适用性

本文档适用于表：适用型号中描述的所有芯片型号。

表 1-1 适用型号

芯片系列	芯片型号
JS32L011 系列	JS32L011L7L6、JS32L011P7L6、JS32L011T7L6

## 2. 注意事项摘要

表 2-1 注意事项摘要

功能	章节	限制
Flash	3.2.1	CRKEY、OPTKEY、PDKEY 解锁过程中，在写入密钥 KEY1、KEY2 之间，插入对当前 LOCK 位置 1 的操作，写密钥 KEY2 时会进入 HardFault
	3.2.2	经过上电复位或退出 Standby 复位后，对其他选项字节寄存器进行编程，Option bytes 区中写保护区 B 结束地址域 (FLASH_WRP1BE) 会清零
GPIO	3.3.1	PB13 在 tamp 外部检测第二路使能时数字输出禁止
I2C	3.4.1	从模式 SBC 置 1，RELOAD 为 0，发送完字节计数器 NBYTES[7:0] 的字节数后，此时主机若继续读数据，会触发数据发送
	3.4.2	I2C1 在系统时钟大于 16MHz 进入 stop 模式后，I2C1 的地址匹配不能唤醒，唤醒功能失效
USART、LPUART	3.5.1	在使能地址匹配时，MSB 位为 “1” 的数据自动识别为地址，如果同时使能奇偶校验，校验位自动计算并填充 MSB 位，会导致数据被误识别为地址
	3.5.2	LPUART 中，低功耗模式唤醒标志 LPWK 选择触发条件为 “RXNE 置 1 ” 时，在 LPWK 置 1 后，需先清零 RXNE 标志，才可以使用 LPWKCF 位将 LPWK 标志清零
TIM1	3.6.1	软件方式同时触发断路和断路 2 时，通道输出未按照高优先级的方式输出
	3.6.2	配置为单脉冲模式，且从模式选择为组合复位+触发模式，当计数到达 ARR，触发信号有效时，TIM 无单脉冲信号输出
LPTIM	3.7.1	正交和非正交模式下，递减计数时，ARR 和 CMP 寄存器预装载功能出错
	3.7.2	异步时钟预装载模式下，更新 ARR 或 CMP 时，存在更新不成功的情况
ADC	3.8.1	在单次扫描和循环扫描转换模式，外部硬件触发转换+WAIT+AUTOFF 配置时，转换通道序列中的每个通道转换都需要等待外部硬件触发

## 3. 注意事项描述

### 3.1. 内核

Arm® Cortex®-M0+ 内核版本的用户手册可从 <http://infocenter.arm.com> 获取。

### 3.2. Flash

#### 3.2.1. CRKEY、OPTKEY、PDKEY 解锁过程中，在写入密钥 KEY1、KEY2 之间，插入对当前 LOCK 位置 1 的操作，写密钥 KEY2 时会进入 HardFault

**描述：**

示例，执行以下操作步骤：

- 1) 向 FLASH\_CRKEY 寄存器写入密钥 1；
- 2) 对 LOCK 位置 1，FLASH->CR = 0x80000000；
- 3) 向 FLASH\_CRKEY 寄存器写入密钥 2。

执行步骤 3 后，会进入 Hardfault。

**解决方法：**

避免在解锁过程中插入对 LOCK 位置 1 的操作。

### 3.2.2. 经过上电复位或退出 Standby 复位后, 对其他选项字节寄存器进行编程, Option bytes 区中写保护区 B 结束地址域 (FLASH\_WRP1BE) 会清零

#### 描述:

执行以下操作步骤:

- 1) 更改选项字节寄存器 FLASH\_WRP1BE, 执行选项字节编程;
- 2) 进行上电复位或退出 Standby 复位;
- 3) 对其他选项字节寄存器进行编程, 并执行加载, 此时 FLASH\_WRP1BE 寄存器会清零。

#### 解决方法:

经过上电复位或退出 Standby 复位后, 再次对其他选项字节进行编程时, 需同时更新选项字节 FLASH\_WRP1BE。

## 3.3. GPIO

### 3.3.1. PB13 在 TAMP 外部入侵检测第二路使能时输出被禁止

#### 描述

应用 TAMP 外部入侵检测功能, 配置检测引脚为 TAMP\_IN2 条件下, PB13 引脚的输出功能被禁止。

- 1) GPIO 功能: 输入无影响, 无法用作输出;
- 2) 复用功能:  
LCD\_SEG13、USART3\_CTS、LPUART\_CTS 不受影响;  
SPI2\_SCK、TIM1\_CH1N 不能作为输出;
- 3) 附加功能: WKUP3、ADC\_IN16, 不受影响。

#### 解决方法

在应用 TAMP 外部入侵检测 TAMP\_IN2 时, 避免使用 PB13 引脚的输出功能, 软件配置 PB13 引脚功能时, 请参考上述描述。

## 3.4. I2C

### 3.4.1. 从模式 SBC 置 1, RELOAD 为 0, 发送完字节计数器 NBYTES[7:0]的字节数后, 此时主机若继续读数据, 会触发数据发送

#### 描述

当作为从设备发送数据时, 使能从模式字节计数控制 I2Cx\_CR1 的 SBC 置 1, RELOAD 为 0, 当 I2Cx\_CR2 的 NBYTES[7:0]待传输字节数传输完成时, 此时主机继续读数据, 会将 I2Cx\_TDR 寄存器 TXDATA[7:0]中的数据发送到总线上。

#### 解决方法

在最后一字节数据写入 I2Cx\_TDR 后数据已发送完成, 此时将 SBC 位清 0, 禁止从模式字节计数功能, 此时主机继续读数据会根据 I2Cx\_CR2 的 NOSTRETCH 的值拉低 SCL 等待数据写入 I2Cx\_TDR 或出现溢出错误发送 0xFF。在下次地址匹配 I2Cx\_ISR 的 ADDR 置 1 时, 将待发送数据的字节数写入 NBYTES[7:0]后, 再将 SBC 置 1。

### 3.4.2. I2C1 在系统时钟大于 16MHz 时, 进入 stop 模式后, I2C1 的地址匹配不能唤醒

#### 描述

当作为从设备并使能从 Stop 模式唤醒功能，若进入 Stop 模式前系统时钟大于 16MHz，进入 Stop 模式后，I2C1 的地址匹配不能从 Stop 模式唤醒。

#### 解决方法

在进入 Stop 模式前将系统时钟频率配置为不大于 16MHz 的时钟，可正常通过 I2C1 的地址匹配唤醒。

### 3.5. USART & LPUART

#### 3.5.1. 在使能地址匹配时，MSB 位为“1”的数据自动识别为地址，如果同时使能奇偶校验，校验位自动计算并填充 MSB 位，会导致数据被误识别为地址

##### 描述

在使用地址匹配功能时，MSB 为 1 的字符识别为“地址”，否则识别为“数据”。如果同时使能奇偶校验功能（CR1 寄存器中的 PEN = 1），校验位会自动填充数据的 MSB 位，校验位为“1”的数据会被误识别为地址。

USARTx 与 LPUART 均存在被误识别为地址的问题。

##### 解决方法

在使用地址匹配功能时，避免同时使能奇偶校验功能。

#### 3.5.2. LPUART 中，低功耗模式唤醒标志 LPWK 选择触发条件为“RXNE 置 1”时，在 LPWK 置 1 后，需先清零 RXNE 标志，才可以使用 LPWKCF 位将 LPWK 标志清零

##### 描述

LPWKS[1:0]位域选择为“11”：RXNE 置 1 时，LPWK 位置 1，则在 RXNE 置 1 期间 LPWK 会自动置 1，向 LPWKCF 位写“1”不能有效清零 LPWK 位

由于此种情况下 LPWK 标志会被 RXNE=1 的状态自动置 1，所以在中断使能(LPWKIE=1)时，如果仅向 LPWKCF 位写“1”来清零 LPWK 位，不清零 RXNE 标志，会导致反复进出中断，无法正常退出中断服务程序。

##### 解决方法

先将 RXNE 位清零再向 LPWKCF 位写“1”清零 LPWK 位。

将低功耗使能位 UEWK 位清零时，LPWK 同步清零。

如果应用场景中需要在退出 LPWK 中断服务程序同时保留用于唤醒的接收数据，所有接收数据在后续流程中统一接收，即退出 LPWK 中断服务程序的同时保留 RXNE 标志置 1，可以在以下方案中任选其一：

- 1) LPWK 标志触发源选择为“检测到起始位”；
- 2) LPWK 标志触发源仍为“RXNE 置 1”，在 LPWK 标志触发的中断服务程序中关闭中断使能，即将 LPWKIE 位清零；
- 3) LPWK 标志触发源仍为“RXNE 置 1”，在 LPWK 标志触发的中断服务程序中将低功耗唤醒使能 UEWK 位清零。

### 3.6. TIM1

#### 3.6.1. 软件方式同时触发断路和断路 2 时，通道输出未按照高优先级的方式输出

##### 描述

配置 TIM1\_EVTG 寄存器中的 BG 位和 B2G 位同时置 1，即同时触发断路 1 和断路 2 时，OCx/OCxN 的输出按照断路 2 方式输出，即发生短路后，直接输出无效电平。

##### 解决方法

软件方式触发断路时，应尽量避免软件配置断路 1 和断路 2 同时有效。

### 3.6.2. 配置为单脉冲模式，且从模式选择为组合复位+触发模式，当计数到达 ARR，触发信号有效时，TIM 无单脉冲信号输出

#### 描述

配置 TIM 为从模式，且从模式选择为组合复位+触发模式，此时当计数器计到 ARR，触发信号有效时，TIM 不产生单脉冲信号。

同样，当配置 TIMx 级联，且 MS\_MOD=1 时，当主 TIM 计数器计到 ARR，触发信号有效时，主从 TIM 均不会产生单脉冲信号。但当配置 MS\_MOD=0 时，主 TIM 不会产生单脉冲信号，而从 TIM 可正确输出单脉冲信号。

#### 解决方法

单脉冲模式下，应尽量避免计数器计到 ARR 与触发信号在同一时刻有效。

## 3.7. LPTIM

### 3.7.1. 正交和非正交模式下，递减计数时，ARR 和 CMP 寄存器预装载功能出错

#### 描述

递减计数模式下，当预载的 ARR 值小于当前 ARR 值时，计数功能正确，但预装载功能异常，此时 ARRM 标志无法置起。

#### 解决方法

在正交和非交模式下，应尽量避免使用 ARRM 标志作为判断条件。

### 3.7.2. 异步时钟预装载模式下，更新 ARR 或 CMP 时，存在更新不成功的情况

#### 描述

异步时钟模式下，使用 LPTIM\_IN1 作为 LPTIM 的工作时钟时，当 LPTIM\_KCLK 的频率小于 LPTIM\_IN1，并开启预装载功能（配置 LPTIM\_CFG 寄存器中的 PRELOAD 位为 1）时，动态修改 LPTIM\_ARR 或 LPTIM\_CMP，存在更新不成功的情况。

#### 解决方法

异步时钟模式下，LPTIM\_IN1 的频率应大于 LPTIM\_KCLK 的 2 倍。

## 3.8. ADC

### 3.8.1. 在单次扫描和循环扫描转换模式，外部硬件触发转换+WAIT+AUTOFF 配置时，转换通道序列中的每个通道转换都需要等待外部硬件触发

#### 描述：

在单次扫描和循环扫描转换模式，外部硬件触发转换+WAIT+AUTOFF 配置时，转换序列中的每个通道转换都需要等待外部硬件触发。预期是只在转换通道序列开始时需要外部触发，而转换序列中的各个通道转换只需要清除 EOC 标志位或读取 ADC\_DR 寄存器即可继续下一通道转换。

#### 解决方法：

方法 1: ADC 触发转换改为软件触发方式，即软件触发转换+WAIT+AUTOFF 配置。外部触发产生后，在其中断处理中，通过软件置位 START 启动转换。

方法 2: ADC 初始配置不变，仍为外部硬件触发转换+WAIT+AUTOFF 配置，当外部触发启动第一次转换时，在 EOC 中断处理中修改配置为软件触发方式+WAIT+AUTOFF，之后软件置位 START 启动后续

转换。