



巨晟科技
Jusheng Technology

32 位基于 ARM Cortex M0 微控制器

JS32L010 系列

软件设计注意事项

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

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

版权声明

本资料是为了让用户根据用途选择合适的产品而提供的参考资料，不转让属于珠海巨晟科技股份有限公司或者第三方所有的知识产权以及其他权利的许可。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系统来评价。对于本资料所记载的信息使用不当而引起的损害、责任问题或者其他损失，珠海巨晟科技股份有限公司将不承担责任。未经珠海巨晟科技股份有限公司的许可，不得翻印或者复制全部或部分本资料的内容。

今后日常产品的更新会在适当的时候发布，恕不另行通知。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，并请您通过各种方式关注珠海巨晟科技股份有限公司公布的信息。

如果您需要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

1	注意事项	1
1.1	程序设计注意事项	1
1.2	UART 功能.....	3
1.3	复位功能(MCLR)关闭	4
1.4	系统寄存器和 LVD 寄存器配置.....	4
1.5	SWD 使用注意事项.....	4
1.6	LCD 使用注意事项.....	4
1.7	RTC 使用注意事项	4
1.8	芯片连接 J-LINK 注意事项.....	4

1 注意事项

1.1 程序设计注意事项

1.1.1 程序文件说明

Boot.hex: boot 文件, 单线升级用到, 与 makecode.ini 中 BOOTLOADER_DIS 标志配合使用。

Project.hex: 用户程序编译后生成的应用程序文件。

usr_cfg.bin: 调用 makecode.exe 后生成的用户配置程序, 需要用烧录器或者其它烧写工具写入芯片。

app_project.bin: 调用 makecode.exe 后生成的用户应用程序, 需要用烧录器或者其它烧写工具写入芯片。

makecode.exe: 用户应用程序和用户配置程序生成工具, 用户程序编译工程配置了每次编译后会自动调用此程序。

1.1.2 用户应用程序使用

- 1) 若芯片不使用在 makecode.ini 中指定的功能, 或者只用芯片默认功能, 那么用户编译出的程序, 可以直接烧录芯片且在芯片不保护而且不校验 main 区的情况下, 可以直接用仿真工具下载并 debug 程序。
- 2) 若芯片使用 makecode.ini 中指定的功能, 就需要配置 makecode.ini 中的内容并调用 makecode.exe (实现用户应用程序和配置程序的二次生成), 会把 Boot.hex 和 Project.hex 组合生成 app_project.bin/app_project.hex (用户应用程序) 和 usr_cfg.bin/usr_cfg.hex (用户配置程序), 并将两个程序分别用烧录器或其它烧写工具烧录。
- 3) **发布的 SDK 中, 用户程序编译工程已经配置了每次编译后自动调用 makecode.exe 文件, 从而实现更新 usr_cfg.bin/usr_cfg.hex 和 app_project.bin/app_project.bin 文件。**

1.1.3 用户配置程序使用

1) 用户配置程序生成

- ☆ 芯片可通过 makecode.ini 配置【用户配置程序说明】中详细列出的功能。
- ☆ 用户应用程序需要和使用 makecode.ini 生成的用户配置程序配合使用才能实现在【用户配置程序说明】中列出的功能。
- ☆ 用户编译出应用代码后需要调用我司开发的 makecode.exe 执行文件实现用户程序和用户配置程序的组合(app_project.bin/app_project.hex 和 usr_cfg.bin/usr_cfg.hex)。**用户程序编译工程已经配置了每次编译后自动调用 makecode.exe 文件, 用户可通过烧录工具烧录用户程序和用户配置程序。**

2) 用户配置程序说明

用户配置程序的修改可通过用户配置文件 makecode.ini 实现, **所有配置为 16 进制, 前面不加 0x, 配置内容中 '=' 前后不要加空格等额外符号。**

makecode.ini 中有如下功能可设置:

- ☆ CODE_PROTECT_DIS 代码不保护
0: 代码保护
1: 代码不保护 (默认)
- ☆ BOOTLOADER_DIS bootloader 不使能
0: bootloader 使能, 用于单 pin 升级
1: bootloader 不使能 (默认)
- ☆ BOOTLOADER_BACKUP_DIS bootloader 备份不使能, 非必要情况不建议使用此备份功能, 因为
0: bootloader 备份使能, 备份在用户程序 main 区最后 1.5k 空间
1: bootloader 备份不使能 (默认)
- ☆ NVR0_1_WRITE_EN Nvr0~1 使能擦写标识
0: Nvr0~1 禁止擦写
1: Nvr0 使能擦写, Nvr1 禁止擦写

- 2: Nvr0 禁止擦写, Nvr1 使能擦写
- 3: Nvr0 使能擦写, Nvr1 使能擦写 (默认)
- ☆ MAIN_WRITE_EN 用户程序 main 区使能擦写标识。
- 配置的内容是 32bits 数据, 每一个 bit 控制 1k (2 个 sector) 内容, 32 个 bit 共控制 32k 内容, bit0~bit31 分别控制 sector0~sector31, 默认全部 sector 使能擦写功能。
- 例如: MAIN_WRITE_EN=FFFFFFE 表示 sector0 和 sector1 不能擦写, 其它 sector 可以擦写。
- ☆ MAIN_CODE_CHECK_EN 用户 main 程序区代码校验使能标识,
- 0: 代码校验不使能 (默认)
- 1: 代码校验使能。校验不通过, 用户程序不能启动
- 若用户使用了 makecode.ini 生成的 bin/hex 文件, 而且用户程序是配置有校验功能, 那么用户使用 eflash 保存参数到用户程序校验区内, 若重新开机时, 芯片检测到用户程序校验不正确则不能开机。故在保存参数时, 需把保存参数的 eflash 地址挪到不统计 crc 校验的位置。**
- ☆ MAIN_CODE_LENGTH 用户程序 main 区代码大小, 16 进制, 代码大小的值为一个 sector(512byte) 的倍数。
- 用户 main 区程序的 4 字节校验放在最大代码长度处。main 区域除去该程序区域后, 剩下的为 data 区域。
- ☆ SWD_EN SWD 使能
- 0: SWD 失能, 需要用户程序调用
- SYSCTRL_REG_OPT(SYSCTRL->SYS_CON1 &= (~LL_SYSCTRL_CON1_SWD_EN));同时作用才能关闭 SWD 接口, 关闭 SWD 前需要延时 150ms 以上, 为了能方便使用 swd 接口烧录/升级程序。
- 1: SWD 使能
- ☆ MCLR_EN MCLR(Reset pin PE2)使能
- 0: 普通 IO
- 1: MCLR pin (默认复位引脚使能)
- ☆ SWD_CLK_PULLUP SWD_CLK 引脚上下拉配置, 芯片默认上拉 (PC9), **切记, 只能配置上拉或下拉。**
- 使用我司开发的配置工具配置如下:
- 0: 下拉
- 1: 上拉 (默认)
- ☆ SWD_MAP SWD mapping PA3,PA2 **这组 swd 非必要情况不建议使用, 切换 SWD 不方便程序代码编写和烧录处理**
- 0: PA3 作为 SWD_CLK, PA2 作为 SWD_DATA
- 1: PC9 作为 SWD_CLK, PC8 作为 SWD_DATA (默认)
- ☆ UART_SINGLE_PIN_UPDATE_MAP UART 单 pin 升级端口映射, 单 pin 升级配置只能选择一个 IO 端口,如下所写 IO map 在用户手册中有描述。
- 00~01: 分别对应 PA0~PA1
- 04~0F: 分别对应 PA4~PA15
- 10~1F: 分别对应 PB0~PB15
- 20~27: 分别对应 PC0~PC7
- 2A: 对应 PE1
- 2B: 对应 PE0
- 2C: 对应 PE2
- 2D~32: 分别对应 PC10~PC15

33~38: 分别对应 PD0~PD5

FF: 无效, 没有单 pin 升级映射端口 (默认)。

其它: 禁止

例如: UART_SINGLE_PIN_UPDATE_MAP =06 表示使用 PA6 作为单 pin 升级端口。

1.1.4 用户 main 区域 FLASH 操作注意事项

1) 用户 main 区编译空间配置

若使用用户 main 区代码校验功能, 就要同时配置 MAIN_CODE_LENGTH 的大小, 使编译出的代码大小要小于等于配置文件中配置的 MAIN_CODE_LENGTH 的配置值减去 4 字节 (此 4 字节放校验值)。

举例如下:

打开 makecode.ini 修改为 MAIN_CODE_LENGTH=7C00。

打开 project.sct 修改为 ER_FLASH 0x00000000 0x00007BFC

其中 0x7BFC~0x7BFF 就是保存代码 crc 的。

内容显示如下:

```
LR_FLASH 0x00000000 0x00008000 {      ; 第一个加载域, 名字是 LR_FLASH, 起始地址 0x00000000
    大小 0x00008000
ER_FLASH 0x00000000 0x00007BFC {    ; 第一个运行时域, 名字是 ER_IROM1 起始地址 0x00000000
    大小 0x00008000
*.o (RESET, +First)      ; IAP 第一阶段在 FLASH 中运行
*(InRoot$$Sections)     ; All library sections that must be in a root region
.ANY (+RO)              ; .ANY 与*功能相似, 用.ANY 可以把已经被指定的具有 RW,ZI 属性的数据排除
}
```

2) 芯片 sector 擦除

对同一个地址重复写数据需要执行擦除操作然后再写数据。请不要频繁保存参数, 可以把保存参数的空间分成若干段, 每一段都保存完成, 然后需要保存就再次执行擦除操作, 继续保存, 避免频繁擦除芯片的某个 sector。

1.1.5 用户 NVR 区 FLASH 操作注意事项

1) NVR0~1, NVR2 前面部分区域

NVR0~1 以及 NVR2 的前面部分区域用于单 pin 升级时需要放置我司提供的 bootloader 程序。如果不用单 pin 升级功能就可以不使用 boot, 用户可以在 NVR0~1 保存数据信息, **但是 NVR2 不允许保存数据信息。**

2) NVR2 后面部分区域

此区域作为用户特殊功能配置信息保存区, 配置 makecode.ini 文件内容, 调用 makecode.exe 生成 usr_cfg.bin/hex。

1.2 UART 功能

1) UART 单线升级功能开启

芯片多个 pin 脚支持单线升级程序, 若是需要用到单 pin 升级功能, 需要写入我司开发的 boot 程序和用户配置程序, 在用户配置文件 makecode.ini 中 UART_UPDATE_MAP 可以配置支持单线升级的 IO, 配置为 UART_UPDATE_MAP=FF 表示不支持单线升级。芯片上电, 通过单线升级协议, 可实现对芯片的程序升级。用单线升级功能时, 应用程序启动后延时 100ms 以上, uart 可关闭单线功能。使端口用作其它功能。例如: UART_UPDATE_MAP=06 表示 PA6, 配置的值是 16 进制表示。

2) UART 单线升级功能关闭

用户程序可以把 uart 单线升级功能取消, 然后用作其它功能。为了不影响单线升级功能(**开机后芯片会在这小段时间内侦测升级序列**), 用户程序需要在开机后延时 100ms 以上再关闭单线升级功能。

用户程序通过如下代码实现关闭单线升级功能:

```
SYSCTRL_REG_OPT(SYSCTRL->SYS_CON1 &= ~(LL_SYSCTRL_CON1_UART0_UD_ENABLE));
```

1.3 复位功能(MCLR)关闭

芯片 PE2 引脚默认复位功能(MCLR)，低电平有效，在设计电路时此引脚不要接能影响芯片开机的外设。复位功能(MCLR)可以通过用户配置程序关闭，然后把配置信息写入芯片，这样芯片就不会有外部管脚复位功能。详见：【用户配置程序使用】中的说明。

1.4 系统寄存器和 LVD 寄存器配置

配置系统寄存器和 LVDCON 寄存器，若已经是锁定状态，就需要先调用解锁命令后再配置寄存器，若需要再次锁定可调用锁定命令。

例如：

```
#define SYSCTRL_REG_OPT(expression) {SYSCTRL->SYS_KEY = 0x3fac87e4; expression; SYSCTRL->SYS_KEY = 0;}  
  
SYSCTRL_REG_OPT(SYSCTRL->SYS_CON1 &= ~(LL_SYSCTRL_CON1_UART0_UD_ENABLE));
```

1.5 SWD 使用注意事项

1.5.1 SWD_CLK,SWD_DATA 作为通用 IO 的配置

- 1) 在用户配置文件 makecode.ini 中 SWD_EN 配置为 SWD_EN=0,
- 2) 在用户程序中使用如下代码失能 SWD。

```
SYSCTRL_REG_OPT(SYSCTRL->SYS_CON1 &= (~LL_SYSCTRL_CON1_SWD_EN));
```

这样程序就能在运行此代码后把 SWD 关闭，SWD_CLK 和 SWD_DATA 可作为通用 IO 使用。

在 SWD 失能前，**需要延时 150ms 以上**，为了可以在 SWD 失能前烧录设备重新控制芯片用 SWD 烧录程序（否者需要进入超级模式通过 SWD 接口和 PB4 控制芯片进行烧录）。

1.5.2 SWD IO MAP

SWD_CLK(PC9)，SWD_DATA(PC8)可切换到 SWD_CLK(PA3)，SWD_DATA(PA2)，需要配置 make_code.ini 文件中的 SWD_MAP=0 实现。SWD_CLK(PA3)/SWD_DATA(PA2)这组 swd 非必要情况不建议使用，切换 SWD 不方便程序代码编写和烧录处理。

1.5.3 SWD_CLK 引脚电平对正常开机的影响

用户配置区需要配置 SWD_CLK 引脚开机后是否有上下拉功能（用户配置区全空或校验失败会默认上拉），SWD_CLK, SWD_DAT, PB4 三个引脚需要在芯片开机时满足特殊电平才能保证芯片正常开机。为了方便用户设计硬件电路，若 SWD_CLK 配置是上拉/下拉功能，开机时 SWD_CLK 电平要保证一直是高/低电平（SWD_CLK 配置上拉，开机时此引脚电平要为高；SWD_CLK 配置下拉，开机时此引脚电平要为低），这样，芯片便会正常开机。次引脚不能接开机时影响电平的外设，例如：ADC，LED 等。

1.6 LCD 使用注意事项

当 LCD 模块把 SWD 口 IO 作为 LCD 功能，SWD 功能自动关闭，最好不用 SWD 口 IO 做 LCD 功能，方便用户使用 SWD 功能。

1.7 RTC 使用注意事项

要读 RTC 模块工作中计数值，把 RTCC_CR 寄存器第 4 位 REGSEL 置 1。

要读 RTC 模块初始设置值，把 RTCC_CR 寄存器第 4 位 REGSEL 置 0。

1.8 芯片连接 J-Link 注意事项

当 J-Link 连接芯片寻找 SWD 时，需要做到同电源同地，即芯片的电源要和 J-Link 的参考电源一致且要同地。