

# 应用笔记

## Application Note

**AN1121**

**G32A14xx 系列应用注意事项**

**版本: V1.1**

# 1 引言

本应用笔记提供在 G32A14xx 系列应用时需要注意的事项。

## 目录

|          |                                         |           |
|----------|-----------------------------------------|-----------|
| <b>1</b> | <b>引言.....</b>                          | <b>1</b>  |
| <b>2</b> | <b>硬件应用设计注意.....</b>                    | <b>3</b>  |
| 2.1      | MCU 滤波电容设计.....                         | 3         |
| 2.2      | MCU 供电注意事项.....                         | 3         |
| 2.3      | 外设模块使用 VDD、VSS.....                     | 4         |
| 2.4      | 外部及高频信号.....                            | 5         |
| 2.5      | Input/Output 信号端口串接限流电阻、 并接电容到 VSS..... | 5         |
| 2.6      | IO 电压不能高于 VDD 电压.....                   | 6         |
| 2.7      | IO 灌电流注意事项 .....                        | 6         |
| 2.8      | 暴露在外的端口注意防护.....                        | 7         |
| <b>3</b> | <b>软件应用注意事项.....</b>                    | <b>8</b>  |
| 3.1      | GPIO 操作注意事项.....                        | 8         |
| 3.2      | 使用 LPO 做时钟源注意事项（WDT/EWDT/RTC...） .....  | 8         |
| 3.3      | PLL 的时钟注意事项.....                        | 8         |
| 3.4      | LSI 时钟注意事项 .....                        | 8         |
| 3.5      | I2C 注意事项 .....                          | 8         |
| 3.6      | 擦写 FLASH 注意事项.....                      | 9         |
| 3.7      | 安全模式使用注意事项 .....                        | 9         |
| <b>4</b> | <b>仿真器/烧录工具使用注意事项 .....</b>             | <b>12</b> |
| 4.1      | 开发环境支持 IAR、KEIL、Eclipse .....           | 12        |
| 4.2      | 其它开发环境注意事项 .....                        | 12        |
| <b>5</b> | <b>版本历史.....</b>                        | <b>14</b> |

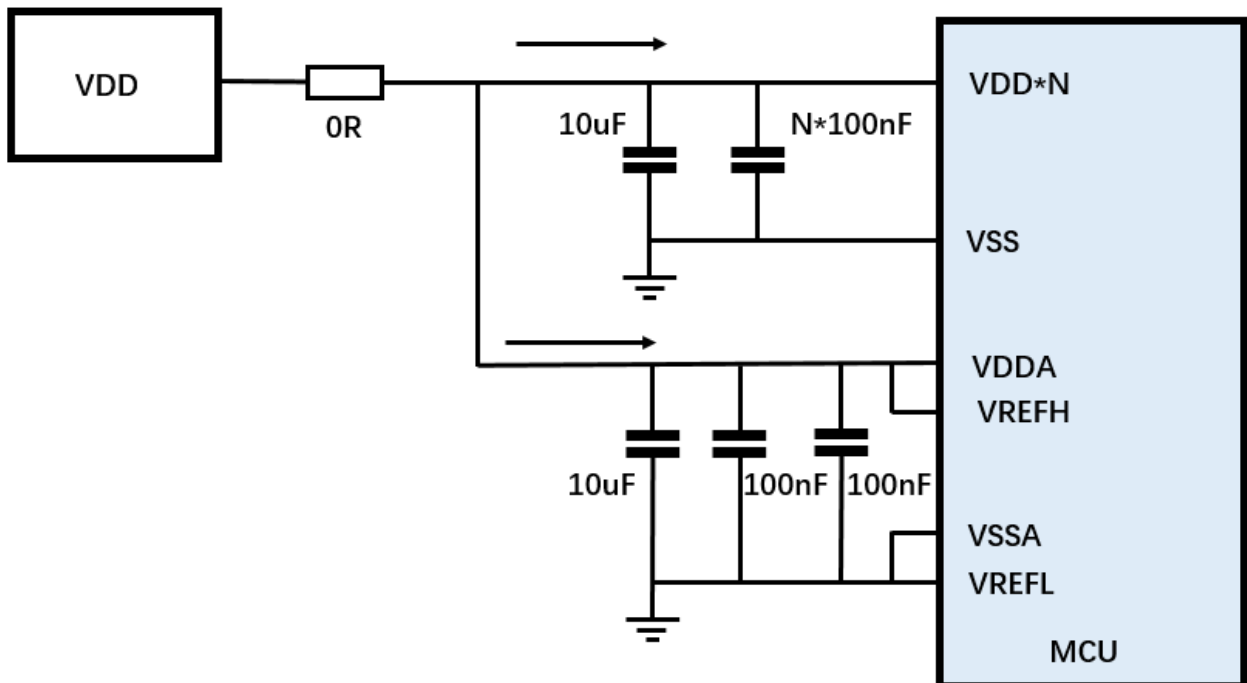
## 2 硬件应用设计注意

### 2.1 MCU 滤波电容设计

芯片电源端口 (VDD、VSS) 需要并联大小滤波电容, 参考如下图 1, N 为各封装的 VDD PIN 数。电容的有效性取决于最佳放置和连接类型, PCB 布局需星型走线注意外部电源先经过大小电容再接入芯片, 大小滤波电容尽量靠近芯片 4mm 以内放置。

图 1 中的电阻 0R 可替换为磁珠, 在 EMI 测试项 (RE) 可增加测试裕量, 参考参数 600R@100MHz。

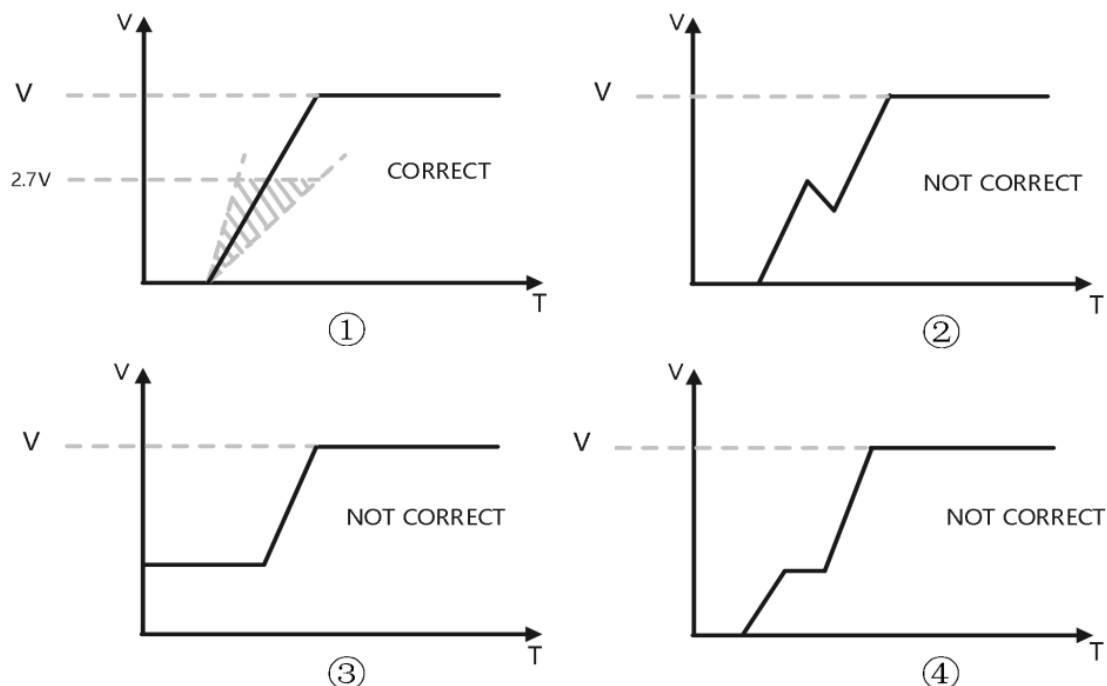
图 1 滤波电容设计



### 2.2 MCU 供电注意事项

MCU 供电斜坡率需满足数据手册要求的最大值和最小值限值, 即大于 0.5V/min 和小于 100V/ms。从 0V 到 5.5V 的时间需要至少需 55us, 从 0V 到 2.7V 的时间至少需要 27us, 上电速率过快或过慢可能导致 MCU 无法正常工作; MCU 掉电要求掉电至 300mV 以下再重新上电, 以上时间需在全温范围下满足要求。上电启动波形需满足图 2 中①波形, 其余三种波形均不正确。

图 2 上电波形



### 2.3 外设模块使用 VDD、VSS

在使用 MCU 控制外部模块的供电或上电过程中, 应当避免过快地上电速率, 以防止因电流突变而导致的 VDD 电压被拉低的问题。为确保系统稳定运行并保护电路免受潜在损害, 建议采用适当的延时措施来平稳地进行供电转换。可以通过增加适当的缓冲元件, 例如电容或电阻, 来调整上电的速度, 不至于 VDD 因为瞬时的大电流需求导致电压骤降, 如因此导致 VDD 骤降, 下降速度也必须符合 ECU 供电斜坡率要求。(如图 3 和图 4 使用示例)

图 3 IO 控制外设模块

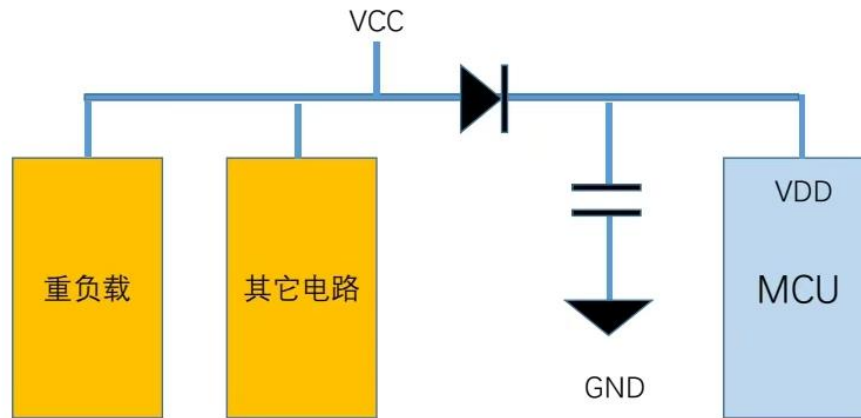


为了规避电源波动, 在 MCU 供电 3.6V 以上的应用中, 在电路上串一个低压降二极管或串一个 10R 电阻和并一个 10uF 以上电容, 组成抗快速掉电电路。如果电容足够大, 掉电将很缓, 且降电幅度可以很小。

注: 如锗二极管导通时会有 0.2V 压降, 这会让 MCU 的供电比原来降低 0.2V, 如使用

ADC/DAC 可能精度受影响。

图 4 外设共用电源



## 2.4 外部及高频信号

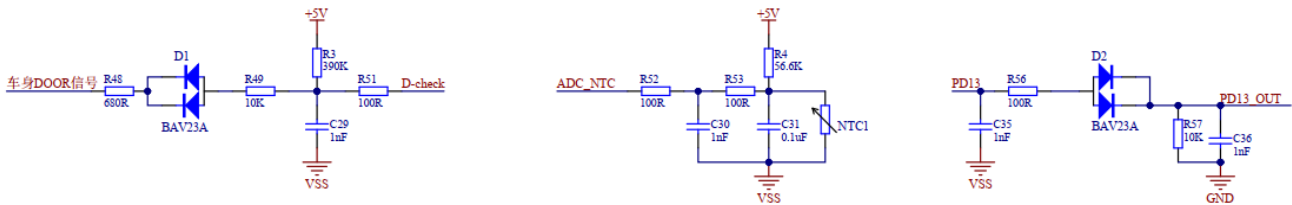
- 如过零信号、继电器信号、AC 负载等，需注意是否会产生瞬间加大负载的情况，可通过旁路电容方式减缓负载冲击。
- 如高频 IIC 时钟线、高频 SPI 时钟线、CAN 高频信号线（CANH、CANL 线）等，需注意是否会产生瞬间尖峰电流。

## 2.5 Input/Output 信号端口串接限流电阻、 并接电容到 VSS

Input/Output 端口（比如：ADC，外部中断，通信接口等），建议电路设计时串接保护电阻，阻值需考虑并计算理论最大电流不能超出芯片端口承受的极限参数；串接的电阻和并接电容均要尽可能地靠近芯片引脚放置，且须先串接电阻，再并接电容。外设模块的滤波电容的地一定要接(VDD、VSS)经过大小滤波电容后的地。如图 5，车身 DOOR 信号、ADC 检测、IO\_OUT 的电阻电容连接示例，IO 端口串接 100R 电阻，并接 1nF 电容。

注意：PCB 相互连接的设计中（类似灯具和控制模块分开、传感器和控制模块分开），这种串接电阻，并接电容一定要放置在芯片的 PCB 板上，合理的参数及放置位置能有效防止 ESD/EOS 对芯片端口的损坏。

图 5 车身 DOOR 信号/ADC 检测/IO\_OUT 的电路示例



如果是按键电路，推荐如下：

图 6 按键电路

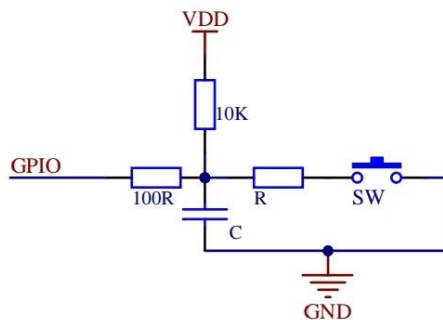


图 6 的 C 推荐用 220nF 电容，在按键旁串一个小电阻 R，如 1k 或 100R，同时在画板时尽量将电容 C 靠近按键 SW 放置。规避按键产生的冲击负压。

## 2.6 IO 电压不能高于 VDD 电压

当 IC 未有 VDD 供电，而 IO 口却存在电压时，则该电压会通过上拉保护二极管向 IC 进行供电。或是当 IC 有 VDD 供电，但 IO 口却存在比 VDD 更高电压时，该电压与 VDD 之间的压差使得上拉保护二极管导通，使得电流流入 VDD。正常情况下 IO 不能高于超过 VDD 0.3V。

此现象易引发以下危害：

- (1) 电流太大会将使 IO 口上的钳位二极管迅速过载并使其损坏。
- (2) 会使单片机复位不成功。
- (3) 会使芯片程序紊乱。
- (4) 会出现门锁效应。

## 2.7 IO 灌电流注意事项

在使用 IO 驱动外部电路时须注意规格书中的电流极限参数，包括但不限于：

- (1) IOL，IO 灌电流，IO 口能够输出的典型灌电流。

(2) IOH, IO 源电流, IO 口能够输出的典型源电流。

(3) IOLS, IO 强灌电流, IO 口能够输出的典型强灌电流。

(4) IOHS, IO 强源电流, IO 口能够输出的典型强源电流。

超过极限参数所规定的范围将对芯片造成损害, 无法预期芯片在上述标示范围外的工作状态, 而且若长期在标示范围外的条件下工作, 可能影响芯片的可靠性。

## 2.8 暴露在外的端口注意防护

- 如通信口, 需要加 ESD 防护措施。可在信号线上串联一个小电阻 100R 限制 ESD 电流的幅度; 可使用瞬态电压抑制器二极管(TVS)并联在信号线上, 靠近接口位置。
- 烧录口存在热拔插风险, 需注意防护, 连接时先接触 GND 再接触 IO。

## 3 软件应用注意事项

### 3.1 GPIO 操作注意事项

1. 未使用到的 GPIO 端口，需将其设置为输出低电平或外部接 100Ω 电阻下拉。
2. GPIO 端口的复用功能，需避免不同 GPIO 引脚配置相同的复用功能！
3. 芯片 VDD 上电从 0V-2.7V 区间，GPIO 端口处于不确定态，需注意 GPIO 不定态对后级驱动电路稳定性的影响；VDD 上电高于 2.7V，芯片完成上电复位后，GPIO 端口状态按照程序的设置执行。

### 3.2 使用 LPO 做时钟源注意事项（WDT/EWDT/RTC...）

LPO 时钟精度较差，建议尽量不选用 LPO 作为时钟源。如果只能选 LPO 作为时钟源，建议考虑极端环境下可能的精度偏差情况。如 WDT 使用 LPO 时钟源时，建议喂狗频率比设置复位时间快一倍以上，即假设看门狗设置 2s 复位，建议每 1s 内喂一次狗。

### 3.3 PLL 的时钟注意事项

G32A14xx 的 PLL 时钟源只能是外部时钟 OSC。

PLL 不能低于 23MHZ。

### 3.4 LSI 时钟注意事项

LSI 为内部 8Mhz 时钟，误差较大（具体查看相关数据手册描述），极端环境下会有 ±10% 的频率误差，建议避免在有精度要求的场合使用，通信外设不使用此时钟源，建议用 OSC 或 PLL 时钟源。

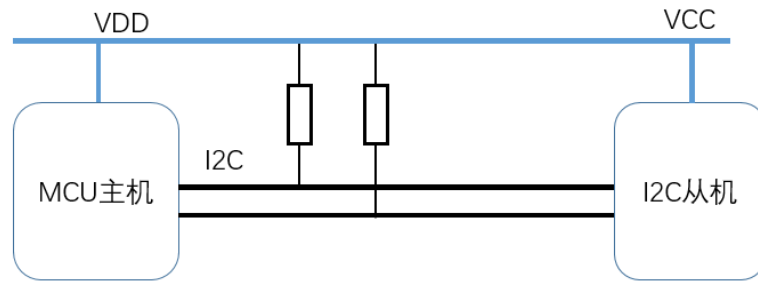
### 3.5 I2C 注意事项

SCL/SDA 在上拉电源突然掉电再回升的过程中，SCL/SDA 电压和逻辑时序的组合会存在触发 START 而无法触发 STOP 的可能，导致总线一直处于 BUSY 状态。建议规避方案：

1. I2C 的上拉电源须保证稳定，预防使用过程中电源不稳掉落过低的情况。常见 I2C 主从通信的场景，建议硬件如下设计：

- a. 主机和从机共用电源，如图 7 电路的，VDD 和 VCC 为共用电源；

图 7 I2C 通信电路



b. 主机和从机不能共用电源时, 建议上拉电阻电源连主机 MCU 电源, 如图 7, 上拉电阻连 VDD。

2. 通过复位从机也可实现释放总线。如果是 EEPROM 作为从机时, 无法使用软件复位从机的方法, 需要在 I2C 主机在建立新的通信时, 增加总线释放功能。由于总线锁死是概率性的, 可以增加总线 BUSY 状态超时功能。二者结合可以提高系统的鲁棒性。

3. 总线上的 I2C 设备, 如果能识别断电, 在断电前把 I2C 模块关闭。

### 3.6 擦写 FLASH 注意事项

1. 如果需要进行连续多页擦除操作, 请确保擦除期间正确清除看门狗计数, 防止擦除时间过长触发看门狗复位。

2. FLASH 擦除写期间请确保供电稳定, 如果供电不稳 (例如出现掉电) 会导致 FLASH 模块工作异常进而出现错误, 以至于下次上电在访问该 FLASH 地址时会触发 ECC 错误。(可以在程序中增加电压监测功能, 如果供电电压过低, 则停止对 FLASH 的操作)

3. 上电延时 100ms 再操作 FLASH 和 EEPROM。

4. 开启 ECC 错误中断, 在中断里将访问错误的 FLASH 地址进行擦除操作, 即可恢复该 FLASH 地址的访问。使能 FLASH 的 ECC 中断将 FLASH\_ECFG.DBFIEN 置 1, 定义中断入口函数 void FLASH\_Fault\_IRQHandler()。

5. 产生 ECC 错误后, 需要将数据擦除。为了数据不丢失, 可用第二个 FLASH 扇区做数据备份, 当出现 ECC 错误后, 从备份区域读取。即擦写完一个扇区后, 再擦写第二个 FLASH 扇区。第二个扇区只有在上电时和出现 ECC 错误才会有读取的情况, 需注意区分是读取哪个 flash 扇区出现 ECC 错误, 擦除对应 FLASH 扇区即可。区分方法可以在读取该地址前做一个信号标记。

### 3.7 安全模式使用注意事项

芯片的安全模式通过配置 0x400-0x40F 地址区域设置, 它是一个专门的 16 字节配置区域用来存储安全、保护及 MCU 启动的一些配置信息。芯片配置处于安全模式, 大规模擦除也禁止, 这时候不能用 JLINK 等烧录工具连接芯片, 也就不能在 J-LINK commander 工具中通过输入命令 unlock Kinetis 执行 mass erase 指令来解密 MCU, 只能执行 FLASH 指令, 用芯片内部指令

**Erase All Blocks** 指令 **0x44** 擦除程序。这个指令只能在程序里执行，也只能通过这样的方式恢复芯片再次可烧录程序。

可以在代码上增加一个程序功能，通过程序里隐藏的串口指令或其它通信接口指令执行 **Erase All Blocks command** 指令 **0x44** 擦除程序，这样既可以保护芯片不被恶意读取程序，也可以在需要程序更新的情况下恢复芯片，解除安全状态。下面是关键代码：（SDK 里有相关代码）

//关键函数

```
STATUS_T FLASH_EraseAllBlock(const FLASH_SSD_CONFIG_T * ssdCfg)
{
    STATUS_T returnValue;

    /* Check OCIFLG to verify the previous command is completed */
    if(0U != (FLASH->STS.reg & 0x80U)>>7U)
    {
        /* Clear COLEFLG & ACCEFLG & PROVFLG flag in flash status register. Write 1 to clear */
        FLASH->STS.reg = (uint8_t)0x70U;

        FLASH->CCMDDATA0.reg = FTFx_ERASE_ALL_BLOCK;//擦除指令

        /* Execute the command */
        returnValue = FLASH_CommandSequence(ssdCfg);
    }
    else {
        returnValue = STATUS_BUSY;
    }
    return returnValue;
}
```

//相关函数

```
FUNCTION_DEFINITION_AT_RAMSECTION_START
static STATUS_T FLASH_CommandSequence(const FLASH_SSD_CONFIG_T * ssdCfg)
{
    STATUS_T returnValue = STATUS_SUCCESS;
```

```
/* Clear OCIFLG to launch command */

FLASH->STS.reg = (uint8_t)(0x80U);

while (((FLASH->STS.reg >> 7U) & 0x01U) == 0)
{
    /* Wait till OCIFLG bit is set

    * Serve callback function as often as possible
    */

    if(ssdCfg->callBack != CALLBACK_IS_NULL)
    {
        /* Temporarily disable compiler's check for ROM access call from within a ram
function.

        * The use of a function pointer type makes this check irrelevant.

        * Nevertheless, it is imperative that the user-provided callback be defined in
RAMSECTION */

        FUNCTION_CALL_AT_RAMSECTION_CHECK_DISABLE
(ssdCfg->callBack)();

        FUNCTION_CALL_AT_RAMSECTION_CHECK_ENABLE

    }
}

/* Check for protection violation error */

if ((FLASH->STS.reg & (0x01U | 0x10U | 0x20U | 0x40U)) != 0U)
{
    returnValue = STATUS_ERROR;
}

return returnValue;
}

FUNCTION_DEFINITION_AT_RAMSECTION_END
```

## 4 仿真器/烧录工具使用注意事项

### 4.1 开发环境支持 IAR、KEIL、Eclipse

1. IAR 建议使用 V8.50.5 版本及以上。

2. KEIL 建议使用 MDK-ARM V5.36 版本。

注意：使用低版本的 MDK 编译不通过的工程，更新到高版本后，原报错的工程不能用，需重新解压 SDK 程序使用。

3. J-Link 软件建议使用 V7.96o 版本及以上。

4. Eclipse 建议使用 V2022-06(4.24.0)版本及以上，可到 Eclipse 下载安装包。

用 Eclipse 编译时，如果提示如下：

Cannot run program "make": Launching failed

Error: Program "make" not found in PATH

解决方法：在工程上右击选择 properties--->C/C++Build--->Tool Chain Editor--->Current builder:

将原来的 Gnu Make Builder 改成 CDT Internal Builder, 点击确定即可。

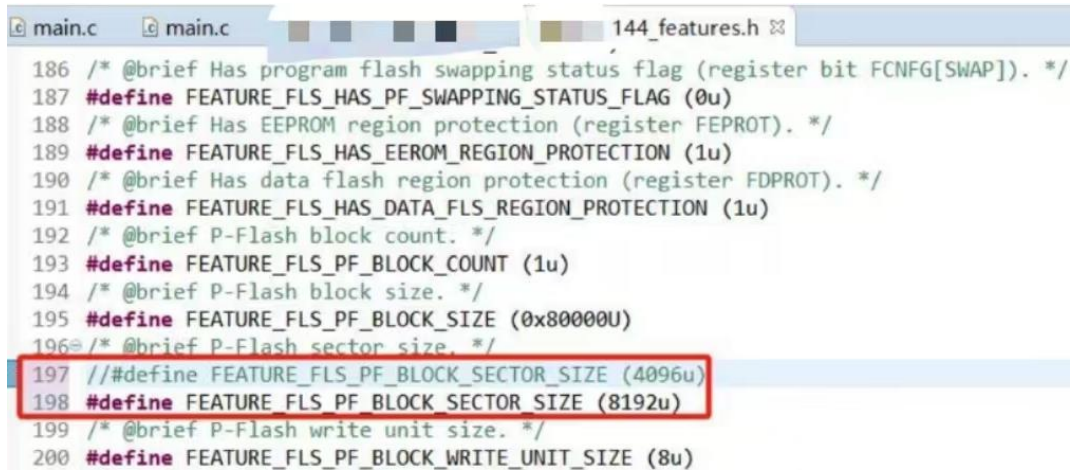
### 4.2 其它开发环境注意事项

如使用第三方软件开发（SxxDS）时注意：G32A14xx 的一个扇区是 8K，而第三方软件可能是擦写时操作的是 4K 或其它。

1. 仿真会有异常或无法仿真，如有问题可以通过如下方式恢复：先用 J-Link/Jflash 烧录程序，再使用第三方软件中的 PE/J-Link 进行仿真调试。

2. G32A14xx P-flash 扇区大小为 8K，执行 boot loader 升级 APP 段程序时，保证擦写原存储区域正常实现，需要修改 P-flash 扇区大小的定义。修改 xx144\_features.h 文档如下图 8 红框定义位置，将 4096u 改为 8192u，修改后要先点击保存再编译，否则编译时无包含最新更改。

图 8 features 文件修改位置



```
186 /* @brief Has program flash swapping status flag (register bit FCNFG[SWAP]). */
187 #define FEATURE_FLS_HAS_PF_SWAPPING_STATUS_FLAG (0u)
188 /* @brief Has EEPROM region protection (register FEPROT). */
189 #define FEATURE_FLS_HAS_EEPROM_REGION_PROTECTION (1u)
190 /* @brief Has data flash region protection (register FDPROT). */
191 #define FEATURE_FLS_HAS_DATA_FLS_REGION_PROTECTION (1u)
192 /* @brief P-Flash block count. */
193 #define FEATURE_FLS_PF_BLOCK_COUNT (1u)
194 /* @brief P-Flash block size. */
195 #define FEATURE_FLS_PF_BLOCK_SIZE (0x80000U)
196 /* @brief P-Flash sector size. */
197 //#define FEATURE_FLS_PF_BLOCK_SECTOR_SIZE (4096u)
198 #define FEATURE_FLS_PF_BLOCK_SECTOR_SIZE (8192u)
199 /* @brief P-Flash write unit size. */
200 #define FEATURE_FLS_PF_BLOCK_WRITE_UNIT_SIZE (8u)
```

3.如为 BootA+BootB+APP 的代码升级模式时，BootB 和 APP 程序都需要保证起始地址定义规则符合 8192（8K）的倍数。APP 的存放起始地址必须是 8192（8K）的倍数，其它信息存放起始地址也必须是 8192（8K）的倍数。

## 5 版本历史

表格 1 文件版本历史

| 日期      | 版本  | 变更历史     |
|---------|-----|----------|
| 2024.12 | 1.0 | 新建       |
| 2025.04 | 1.1 | 增加软件注意事项 |

# 声明

本手册由珠海极海半导体有限公司（以下简称“极海”）制订并发布，所列内容均受商标、著作权、软件著作权相关法律法规保护，极海保留随时更正、修改本手册的权利。使用极海产品前请仔细阅读本手册，一旦使用产品则表明您（以下称“用户”）已知悉并接受本手册的所有内容。用户必须按照相关法律法规和本手册的要求使用极海产品。

## 1、权利所有

本手册仅应当被用于与极海所提供的对应型号的芯片产品、软件产品搭配使用，未经极海许可，任何单位或个人均不得以任何理由或方式对本手册的全部或部分内容进行复制、抄录、修改、编辑或传播。

本手册中所列带有“®”或“™”的“极海”或“Geehy”字样或图形均为极海的商标，其他在极海产品上显示的产品或服务名称均为其各自所有者的财产。

## 2、无知识产权许可

极海拥有本手册所涉及的全部权利、所有权及知识产权。

极海不应因销售、分发极海产品及本手册而被视为将任何知识产权的许可或权利明示或默示地授予用户。

如果本手册中涉及任何第三方的产品、服务或知识产权，不应被视为极海授权用户使用前述第三方产品、服务或知识产权，也不应被视为极海对第三方产品、服务或知识产权提供任何形式的保证，包括但不限于任何第三方知识产权的非侵权保证，除非极海在销售订单或销售合同中另有约定。

## 3、版本更新

用户在下单购买极海产品时可获取相应产品的最新版的手册。

如果本手册中所述的内容与极海产品不一致的，应以极海销售订单或销售合同中的约定为准。

#### 4、信息可靠性

本手册相关数据经极海实验室或合作的第三方测试机构批量测试获得，但本手册相关数据难免会出现校正笔误或因测试环境差异所导致的误差，因此用户应当理解，极海对本手册中可能出现的该等错误无需承担任何责任。本手册相关数据仅用于指导用户作为性能参数参照，不构成极海对任何产品性能方面的保证。

用户应根据自身需求选择合适的极海产品，并对极海产品的应用适用性进行有效验证和测试，以确认极海产品满足用户自身的需求、相应标准、安全或其它可靠性要求；若因用户未充分对极海产品进行有效验证和测试而致使用户损失的，极海不承担任何责任。

#### 5、合规要求

用户在使用本手册及所搭配的极海产品时，应遵守当地所适用的所有法律法规。用户应了解产品可能受到产品供应商、极海、极海经销商及用户所在地等各国有关出口、再出口或其它法律的限制，用户（代表其本身、子公司及关联企业）应同意并保证遵守所有关于取得极海产品及/或技术与直接产品的出口和再出口适用法律与法规。

#### 6、免责声明

本手册由极海“按原样”（as is）提供，在适用法律所允许的范围内，极海不提供任何形式的明示或暗示担保，包括但不限于对产品适销性和特定用途适用性的担保。

极海产品并非设计、授权或担保适合用于军事、生命保障系统、污染控制或有害物质管理系统中的关键部件，亦非设计、授权或担保适合用于在产品失效或故障时可导致人员受伤、死亡、财产或环境损害的应用。

如果产品未标明“汽车级”，则表示不适用于汽车应用。如果用户对产品的应用超出极海提供的规格、应用领域、规范，极海不承担任何责任。

用户应该确保对产品的应用符合相应标准以及功能安全、信息安全、环境标准等要求。用户对极海产品的选择和使用负全部的责任。对于用户后续在针对极海产品进行设计、使用的过程中所引起的任何纠纷，极海概不承担责任。

#### 7、责任限制

在任何情况下，除非适用法律要求或书面同意，否则极海和/或以“按原样”形式提供本手册及产品的任何第三方均不承担损害赔偿责任，包括任何一般、特殊因使用或无法使用本手册及产品而产生的直接、间接或附带损害（包括但不限于数据丢失或数据不准确，或用户或第三方遭受的损失），这涵盖了可能导致的人身安全、财产或环境损害等情况，对于这些损害极海概不承担责任。

## 8、适用范围

本手册的信息用以取代本手册所有早期版本所提供的信息。

©2025 珠海极海半导体有限公司 – 保留所有权利