

# 应用笔记

## Application Note

文档编号: **AN1131**

**G32R501 Tool** 用户手册

版本: **V1.0**

# 1 引言

由于 G32R5xx 系列 MCU 的芯片特性, 该芯片在特定例程前需要对一些生成文件进行数据流设置。例如, flashapi\_ex2\_uart\_kernel 的数据流。“geehy\_tool”将帮助您生成所需数据流格式, 确保程序正常运行。

在正式使用“geehy\_tool”工具前请您先阅读本说明文档。工具运行环境:

- Windows 10/11 系统

## 目录

<b>1</b>	<b>引言 .....</b>	<b>1</b>
<b>2</b>	<b>Geehy_Bin 使用说明 .....</b>	<b>3</b>
2.1	Geehy_Bin.exe 支持的命令 .....	3
2.2	使用示例 .....	4
<b>3</b>	<b>Geehy_Serial_flash_Programmer.....</b>	<b>6</b>
3.1	Geehy_Serial_flash_Programmer.exe 支持的命令.....	6
3.2	使用示例 .....	6
<b>4</b>	<b>版本历史.....</b>	<b>7</b>

## 2 Geehy\_Bin 使用说明

### 2.1 Geehy\_Bin.exe 支持的命令

-h: 显示帮助信息。

-v: 显示详细的日志信息。

-boot: 将所有段转换为可引导形式。

-uart8: 指定 UART 格式, 8 位模式。

-spi8: 指定 SPI 格式, 8 位模式。

-lospcp=value: 指定 lospcp 寄存器的初始值。

-spibrr=value: 指定 spibrr 寄存器的初始值。

-i2c8: 指定 I2C 格式, 8 位模式。

-i2cpsc=value: 指定 i2cpsc 寄存器的初始值。

-i2cclkh=value: 指定 i2cclkh 寄存器的初始值。

-i2cckl=value: 指定 i2cckl 寄存器的初始值。

-pc=value: 指定 PC 的值。

-a: 选择 ASCII 十六进制。

-o: 指定输出文件。

-m: 选择主要的安全引导模式。

-e: 选择扩展的安全引导模式。

-me: 选择主要安全引导和扩展安全引导模式。

-mesta=: 如果选择了主要和扩展引导模式, 则选择扩展引导的起始地址。

-melen=: 如果选择了主要和扩展安全引导模式, 则扩展安全引导的闪存长度。

--cmac=file: 指定存储 CMAC 密钥的 txt 文件。

xxx.cmd: 指定存储 CMAC 计算起始地址、检查的闪存长度和填充值的 CMD 文件。

-tag=value: 指定放置标签值的起始地址。

-load=value: 指定文件下载的起始地址。

--load\_image: 选择生成 BIN 文件。

--image: 选择生成 TXT 文件。

## 2.2 使用示例

本节将介绍如何使用 `Geehy_Bin.exe` 工具执行一些常见的命令操作。这些示例将帮助理解如何将二进制文件转换为不同格式，配置启动参数，以及实现安全引导模式等功能。通过下面的例子，可以了解工具的基本用法，并根据需求进行修改和应用。

### 2.2.1 UART\_Example

在本部分，将介绍如何使用 `Geehy_Bin.exe` 将文件转换为 UART 8 位模式。每个示例展示了如何设置不同的 PC 值，并将结果输出为 ASCII 的十六进制格式。

- 将 `flashapi_ex2_uart_kernel_CPU0_RAM.bin` 文件转换为可引导的 UART 8 位模式格式，初始 PC 值设为 `0x00000000`，输出为 ASCII 的十六进制格式，并将结果保存到 `flashapi_ex2_uart_kernel_CPU0_RAM.txt` 中。

```
Geehy_Bin.exe -boot -uart8 -pc=0x00000000 -a -o flashapi_ex2_uart_kernel_CPU0_RAM.txt  
flashapi_ex2_uart_kernel_CPU0_RAM.bin
```

- 与前一个指令类似，但将 PC 的初始值设为 `0x08000000`，输出文件为 `flashapi_ex2_uart_kernel_BANK0_LDFU.txt`。

```
Geehy_Bin.exe -boot -uart8 -pc=0x08000000 -a -o flashapi_ex2_uart_kernel_BANK0_LDFU.txt  
flashapi_ex2_uart_kernel
```

- 执行与前两条类似的操作，只是将 PC 的初始值设为 `0x08020000`，输出文件为 `flashapi_ex2_uart_kernel_BANK1_LDFU.txt`

```
Geehy_Bin.exe -boot -uart8 -pc=0x08020000 -a -o flashapi_ex2_uart_kernel_BANK1_LDFU.txt  
flashapi_ex2_uart_kernel_BANK1_LDFU.bin
```

### 2.2.2 SPI\_Example

将 `project.bin` 文件转换为可引导的 SPI 8 位模式格式，`lospcp` 为 2，`spibr` 为 127，初始 PC 值设为 `0x08000000`，输出为 ASCII 的十六进制格式，并输出到 `project.txt` 中。

```
Geehy_Bin.exe -boot -spi8 -lospcp=0x02 -spibr=0x7F -pc=0x08000000 -a project.bin -o  
project.txt
```

### 2.2.3 I2C\_Example

将 `project.bin` 文件转换为可引导的 I2C 8 位模式格式，`i2cpsc` 为 4，`i2cclk` 为 45，初始 PC 值设为 `0x08000000`，输出为 ASCII 的十六进制格式，并输出到 `project.txt` 中。

```
Geehy_Bin.exe -boot -i2c8 -i2cpsc=0x04 -i2cclk=0x2d -i2cclk=0x2d -pc=0x08000000 -a  
project.bin -o project.txt
```

## 2.2.4 Secure\_Example

下面介绍一些安全引导模式的示例。通过命令，可以使用 **CMAC** 密钥对文件进行加密处理，并指定文件生成的格式（如 **txt** 或 **bin**）。以下示例展示了主要和扩展安全引导模式的不同应用场景，以及如何配置标签值和下载起始地址。

- 使用 **TEST.cmd** 文件来指定 **CMAC** 计算的起始地址和其他相关参数。它选择主要的安全引导模式，生成 **txt** 文件格式，使用 **key.txt** 中的 **CMAC** 密钥对 **user.bin** 进行处理，并在地址 **0x08000008** 处放置标签值，最终输出到 **test\_major\_cmac.txt** 中。

```
Geehy_Bin.exe test_major.cmd -m --image --cmac=key.txt user.bin -tag=0x08000008 -o test_major_cmac.txt
```

- 与前一个指令类似，但生成 **bin** 文件格式，结果保存到 **CMAC.bin** 中。

```
Geehy_Bin.exe test_major.cmd -m --load_image --cmac=key.txt user.bin -tag=0x08000008 -o test_major_cmac.bin
```

- 使用扩展的安全引导模式，生成 **txt** 文件格式，使用 **key.txt** 中的 **CMAC** 密钥。它对 **user.bin** 进行处理，在地址 **0x08003000** 放置标签值，从地址 **0x08000000** 开始下载文件，最终输出到 **test\_extend\_cmac.txt** 中。

```
Geehy_Bin.exe test_extend.cmd -e --image --cmac=key.txt user.bin -tag=0x08003000 -load=0x08000000 -o test_extend_cmac.txt
```

- 与前一个指令类似，但生成的是 **bin** 文件格式，结果保存到 **test\_extend\_cmac.bin** 中。

```
Geehy_Bin.exe test_extend.cmd -e --load_image --cmac=key.txt user.bin -tag=0x08003000 -load=0x08000000 -o test_extend_cmac.bin
```

## 3 Geehy\_Serial\_flash\_Programmer

在本章中, 将介绍如何使用 `Geehy_Serial_flash_Programmer.exe` 及其相关命令。该工具主要用于通过串行端口对设备进行编程。

### 3.1 Geehy\_Serial\_flash\_Programmer.exe 支持的命令

- d <device>: 加载的设备名称。例如: r501。
- k <file>: 闪存内核的文件名。该文件必须为 UART 引导格式。
- a <file>: 下载使用的文件名。该文件必须为 UART 引导格式。
- p COM<num>: 设置用于通信的 COM 端口。
- b <num>: 设置 COM 端口的波特率。
- ? 或 -h: 显示帮助信息。
- q: 安静模式。禁用输出到 `stdout`。
- w: 在退出前等待按键。
- v: 启用详细输出。

注意: -d、-f 和 -p 是必需参数。如果省略波特率, 通信将以 9600 波特进行。

### 3.2 使用示例

在 `G32R5xx_SDK\utilities\geehey_tool` 路径下包含两个不同的上位机版本, 其中 `Geehy_Serial_flash_Programmer.exe` 与 MCU BOOTROM 配合使用, 并不能进行程序的烧录。而 `Geehy_Serial_flash_Programmer_appln.exe` 与 `uart_flash_kernel` 程序配合使用。下面提供了一些使用命令行工具的示例。这些示例展示了如何设置设备、指定闪存内核文件、下载文件以及配置通信端口和波特率。

- 如何使用 `Geehy_Serial_flash_Programmer.exe` 为设备 r501 进行编程。我们设定了闪存内核文件和下载文件名, 并指定使用 COM11 端口进行通信, 波特率设为 38400。

```
Geehy_Serial_flash_Programmer.exe -d r501 -k flashapi_ex2_uart_kernel_CPU0_RAM.txt -a LED1_0x0800C000.txt -p COM11 -b 38400
```

```
Geehy_Serial_flash_Programmer_appln.exe -d r501 -k flashapi_ex2_uart_kernel_CPU0_RAM.txt -a LED2_0x0800C000.txt -p COM11 -b 38400
```

## 4 版本历史

表格 1 文件版本历史

日期	版本	变更历史
2025.01	1.0	新建

# 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