

KF32 系列

编程小工具使用说明 V1.0.1

2023 年 12 月

1. 概述

“ProgrammerAssistant.exe”为基于 windows 32 平台设计控制台程序，使用 KF32 编程器实现灵活的编程需求。支持 windows xp、windows 7、windows 8、windows 10 的系统平台。

Windows 系统需要安装编程器的驱动程序或使用系统支持的兼容驱动

【win10】，该驱动随 ChipON KF32 IDE 或 PRO 软件发布，即安装软件后选项安装，或主动软件安装目录下的 KFLink 文件夹内维护，该目录下同时存在对应发布的编程器固件程序。

“ProgrammerAssistant”为基于 linux X64 平台设计控制台程序，使用 KF32 编程器及 libusb 驱动模式实现灵活的编程需求。支持 centOS、Ubuntu 等系统平台。

Linux 系统应配置程序的 root 权限使能 libusb 驱动工作。

应使用配套的编程器固件从而确保编程适应性的最佳匹配。

2. 功能介绍

使用命令行或批处理文件调用程序并传递参数完成编程相关的下载或读取。也可以通过其他上位机程序调用完成编程的相关操作。

运行命令行格式如：ProgrammerAssistant -port COM3 -mcu KF32F350MQV -file "d:\a.hex" hex 其中程序为相对当前路径，若该程序与工作目标不一致时，可以使用完整路径输入可执行程序，若路径存在空格，需要使用双引号格式输入。

示例测试用例(Windows 平台)：创建文件文件 test.txt 并输入如下格式内容并调整匹配参数

```
ProgrammerAssistant -port COM3 -mcu KF32F350MQV -file "d:\a.hex" hex
Pause
```

保存并关闭文件，重命名文件为 test.bat，双击启动运行，并控制台 cmd 程序下显示程序的运行过程。其他 pasue 仅为了结束时暂停查看控制台的过程信息。

支持指定地址段编程，最小划分单元为 1K，即 1024 字节。

2.1 命令行参数

需要在指定文件相关命令之前执行芯片型号的选择选项，即-mcu 选项。当未指定 block 或 addr 选项时，默认为整个程序空间的编程。支持多组的多个文件和程序空间参数输入。参数通过空格间隔，区分大小写。地址与大小的数值字符串支持 10 进制与 16 进制格式，如 1234, 0x1234。

2.1.1 -mcu chipname

该命令指定当前编程的目标型号芯片，如-mcu KF32F350MQV，芯片型号名须要按大写字符传递。若缺失该选项或错误，给予错误指示信息：

Error->this chip name [XXXXX] is not support now!

2.1.2 -port COMx

KungFu32 的编程器工作在虚拟串口模式，通过该参数选择工作编程器，windows 平台有效端口号为 1-255。具体以设备管理器中识别为准。若未传递该参数，或串口号错误，或者编程器状态异常下，给予错误指示信息：

Error->KF_DEVICE_NOT_FOUND。参数的 COM 必须大写。

仅 Windows 平台有效，Linux 平台自动识别编程器并使用。

2.1.3 -file filename fileformat [addr]

指定下载加载的文件与文件格式，**仅格式为 bin 时，存在指定入口地址参数。**另支持 hex 格式和 s19 格式文件的输入，若文件名中存在空格等特殊符号时，传递参数时必须使用双引号，参数通过空格间隔。

参数示例如 -file ". /abc.hex" hex

支持相对路径与绝对路径。支持组合程序的多个命令行参数输入，若文件包含相同的地址段内容，以最终的为准，即加载替换。

2.1.4 -block startid stopid

基于 Flash 的容量的直观参数传递，参数应大于 0，并小于该芯片的容量空间，如 64K 容量型号的最小起始为 1，最大终止为 64，示例：-block 2 3 对应编程芯片的第 2K 空间，第 3K 空间。

支持多个参数输入，即支持不连续空间配置。

2.1.5 -addr startaddr size

相对 block 的基于绝对地址的选择编程空间。起始地址应该基于整 1K 空间对齐，大小应该整 1K 的倍数，如编程第 2K 空间的示例：-addr 0x400 0x400，数据地址与大小参数支持 10 进制表达，如-addr 1024 1024

支持多个参数输入，即支持不连续空间配置。

2.1.6 -copy filename fileformat

非编程功能文件格式转换，即-file 指定相关文件，并该功能实现文件另存

功能，格式支持 hex bin s19。当文件已经存在时，可以优先添加-autocreat 选项实现默认覆盖，否则提示文件存在错误。该选项支持多个存在，即同时输出 hex bin s19。

该模式下仍有效或部分有效的参数：

- mcu (必须)
- file (必须)
- pid (源文件微调)
- replace (源文件微调)
- block(唯一存在仅输出 bin 有效) 或 -addr(唯一存在仅输出 bin 有效)
- flashdata 选择保存 hex s19 时输出数据区
- writeconfig 选择保存 hex s19 时输出模式与加密选项

另外，保存 bin 时根据 addr 范围可以选择保存 flashdata 或 config，默认保存 flash，未通过 block 或 addr 修饰时的另存 bin 为自动范围，即基于 32 字节对齐的从行非空输出，并输出到尾行不空，支持 flash 程序段不连续。

2.1.7 -flashdata

KungFu32 型号提供 1K 的可选数据空间，默认情况不编写，该选项启动该 1K 空间的编写。

2.1.8 -writeconfig

KungFu32 型号具有可选择的功能配置选项和加密等级选项，默认情况下不编写的更改，该选项启动基于文件或默认空值启动该写入。

2.1.9 -nocapability

执行编程时，考虑板级容性差异的握手时刻不同，默认使用编程固件的握手尝试并自动选择参数，若裸片或容性无差异影响下，可通过该参数关闭该选项。默认握手会额外占用 200-800ms 左右不等的时长。

2.1.10 -pid "formatString"

格式化输入命令格式为：

-pid "area,offsetaddr,Department,Materiel,
Channel,year,mouth,day,StartNumber,StopNumber"

该 pid 意味着选择 4 字节对齐的 12 字节的 Flash 空间或 FlashData 空间存储产品信息，其中 Department 占用 1 个字节 Materiel 占用 2 个字节 Channel 占用 1 个字节 year 占用 2 个字节，mouth 占用 1 个字节 day 占用一个字节。

StartNumber 占用 4 个字节 StopNumber 不占用字节的描述终止系列号。本设计为尾 4 字节对应自动+1 的唯一序号值。当 StartNumber== StopNumber==0 时作为无自增需求表达。该程序为单次执行，故自增实现无体现，由上层应用进行提供新的起始数据，该工具程序执行对应格式的 12 字节内容的替换后执行编程。故该实现也可由 replace 选项功能替代。

支持多组输入。

如：-pid "flash,0,4,256,0x45,2020,12,23,12345,0xFFFFFFFF"

-pid "data,0x300,4,256,0x45,2020,12,23,12345,0xFFFFFFFF"

area 对应字符串 flash 或 data，其他为数值，可以为 10 进制也可以为 16 进制。

2.1.11 -replace "formatString"

格式化输入命令格式为：

-replace "area, offsetaddr, size, date1, date2, ... daten"

同 pid 意图一致，在原有程序内容的基础上主动执行指定区域的内容的替换，可以是产品信息。

数据以字节为基本单元，单次最多 1024 个字节的替换。支持多组输入。

area 对应字符串 flash 或 data，其他为数值，可以为 10 进制也可以为 16 进制。

如：-replace "flash,0,4,0x00,0x00,0x01,0x10"

area 对应字符串 flash 或 data，其他为数值，可以为 10 进制也可以为 16 进制。

2.1.12 -read filename

该选项明确操作类型为读取芯片，仅支持保存为 hex 格式的文件。若文件名中存在空格，传递参数时应使用双引号。如-read ". /123.hex"，支持相对路径与绝对路径。

使用该选项时选项-block -addr 不生效，不需要指定源码文件-file。不需要配置是否编写数据空间-flashdata 和配置空间-writeconfig。

当文件名中存在空格等特殊符号时应使用双引号输入，否则可以直接传递。

2.1.13 -noskipzerowrite -noskipzerocheck

使用-addr，-block 或默认完整编程下，编程自动跳过以 1K 为单位的空代码区域，当需要进行空片段的擦除并编写时可以使用 -noskipzerowrite 选项，同样的默认的空代码片段不给予读验证编程正确性，作为非代码区域跳过，但使用 -noskipzerocheck 选项明确指定需要进行空内容的校验。

2.1.14 listPort

唯一的独立命令行选项，罗列识别的串口设备。

使用该选项时选项-block -addr 不生效，不需要指定源码文件-file。不需要配置是否编写数据空间-flashdata 和配置空间-writeconfig。

仅 Windows 平台有效。

2.1.15 -?

查看命令行选项信息。

2.1.16 -v

查看版本信息

2.1.17 -which

用于返回芯片唯一 id 值，即识别芯片功能，打印显示格式：

Note->UniqueSerialNumber [XXXXXXXX-XXXXXXXX-XXXXXXXX-XXXXXXXX]

该信息对应 128bit 整型数组为 0, 1, 2, 3.

2.1.18 -erase

编程功能自带擦除编程与校验。该擦除方法旨在将芯片擦除为出厂的原始一定空状态。该方法将对 flash 部分的擦除是否为空做快速检测，即特征值方法，异常后推送信息。

注 1：当 addr 或 block 参数将认为权限下的指定范围的擦除。

注 2：应考虑配置-writeconfig 和 -flashdata，扩展擦除范围。

2.1.19 -empty

独立功能方法，用于验证芯片程序是否为空，该方法验证是否为空采用基于编程最小单元的特征值校验方法，若识别首个非空，将非空的信息给予显示。

注 1：当 addr 或 block 参数将认为权限下的指定范围的擦除。

注 2：应考虑配置-writeconfig 和 -flashdata，扩展擦除范围。

2.1.20 -check

该模式用于检测芯片内容是否与加载的程序内容是否一致，其中选项 数据空间和配置模式选项参与检测范围的控制，同时基于 block 或 addr 的范围同时参与 Flash 选择的控制。

2.1.21 -fastcheck

针对查空、擦除、校验以及编程进程的结果验证机制选项，多数芯片默认编程采用读取并比较的模式，或单个对齐内存单元如 1K 的特征值验证。该配置参数联合相同属性内存对齐块 nK 的特征码验证，适合进程包括：擦除和查空进程的芯片程序区是否为空确认，校验和编程进程的程序与芯片结果特征值是否一致。

注 1：使用该校验码验证机制将一定程度上提升操作速度。

注 2：针对 USB 总线空闲时，单次数据交换耗时可忽略，此时部分进程提升不大，当 USB 总线忙碌时，该机制较少通讯交换此时有助改善速度。

注 3：当如查空进程下芯片首个对齐内存单元非空时，该参数将消耗更多的时间。即 1K 特征值与 nK 特征值的芯片层计算耗时差异。

2.1.22 -boot_secure "Start , Length , Mode , Swap"

仅支持同 A158 特性系列处理器。

该选项用于指定命令行编程时，预设程序的安全启动信息，即起始地址、长度、模式、交互配置。需严格按照该格式的输入，逗号用于参数间隔。

Start, Length 支持 10 进制、16 进制模式的值输入。

Mode 指定配置是否严格安全启动(异常时不执行 APP 程序)或宽松安全启动(APP 程序识别结果)。严格：0xA55A。宽松：0x0000。支持 32bit 的输入，其高 16bit 可任意指定，该值将作为 APP 阶段匹配值的修改安全启动的关键字验证，默认值 0x0000 (APP 设定参数模式配置，高位不匹配的条件否决)。

Swap 指定是否前 1M 空间和后 1M 空间交互。软件限定值小于 5bits。其中仅 bit0 位有效位，0 默认不切换，1 切换。

建议：安全启动应根据程序的空间使用情况，不推荐配置长度大于 256K。模式用于配置是否严格安全启动(异常时不执行 APP 程序)或宽松安全启动(APP 程序识别结果)。

示例参数：-boot_secure "0, 64, 0xA55A, 0x30"

2.1.23 -secret_boot_key keyvalue

仅支持同 A158 特性系列处理器。

该选项指定用于安全启动的密钥指定，可以联合 -boot_secure 选项编程时默认启动安全启动的配置(更新芯片中该安全启动的密钥值)。此选项等效于实施安全启动密钥场景下更新密钥，无该配置时，安全启动密钥可来源于项目源码文件编译的输出(项目代码配置默认下载初始化)，否则无此选项时 boot_secure 使用芯片中存在(出厂默认空值或历史配置的密钥)。

示例参数：-secret_boot_key 2b7e151628aed2a6abf7158809cf4f3c

注：该密钥必须采用 16 进制的 128bits 完整输入。其为大端输入模式，即 0x2b 为大端模式下的高位值，存储在小端处理器的最高字节值 (u_char arr[16] ;arr[15]=0x27)。

2.2 过程信息与错误说明

基本信息通过前缀表面类型，分 4 个类型，即信息 Note、错误 Error、耗时与过程 Time 以及其他。

2.2.1 耗时与过程

“Time:” 开头的信息指示工作过程，格式：Time: HH:MM:SS percent: X %
过程指示如下：

Time: 00:00:02 percent: 17 % // 即 2 秒时进度完成约 17%。

2.2.2 错误

“Error->” 开头的信息对应过程的错误产生。

可能错误但非全部参考如下，具体以软件为准：

```
Error-> MFC initialization failed
Error-> Parameter Fail, [%s]
Error->start Thread Fail, Fail code:0xFFFF
Error->this chip name [%s] is not support now!
Error->this block start [%s] is invalid!
Error->this block stop [%s] is invalid!
Error->this address size [%s] is invalid!
Error->replace area is not support or input error [%s]!
Error->replace format error [%s], separator need by', '!
Error->pid area is not support or input error [%s]!
Error->pid format error [%s], separator need by', '!
Error->KF_PROGRAM_PidFail, no PID Increase
Resource: [%d] at [0x%X] Resource [0x%X-0x%X], now [0x%X]!!!
Error->KF_DEVICE_NOT_FOUND
Error->KF_PROGRAM_ErrChip, Chip no match , read [%X], but should be [%X]!!!
Error->KF_PROGRAM_PowerONFaile
Error->KF_PROGRAM_PowerOffFaile
Error->KF_PROGRAM_TransmitFail
Error->KF_PROGRAM_ProFail, [%d] at section [%X]
Error->KF_PROGRAM_EarseFaile, [%d] at section [%X]
Error->KF_PROGRAM_CheckFail, [%d] at address [%X], (0x%X, 0x%X)
Error->KF_PROGRAM_CheckFail, [%d] with Value [%X]
Error->KF_PROGRAM_CheckFail
```


2.2.3 信息

“Note->”开头的信息对应过程的状态指示，如操作启动、操作完成。

主要信息如下：

Note->begin Write .

Note->Write finish!

Note->begin Read .

Note->Read finish!

Note->begin .

Note->finish!

增加芯片信息的显示：

Note->UniqueSerialNumber [XXXXXXXX-XXXXXXXX-XXXXXXXX-XXXXXXXX]

该信息对应 128bit 整型数组为 0, 1, 2, 3.

2.2.4 其他

Linux 平台使用 libusb 驱动操作，当发生错误时，其输出格式以 libusb 起始，如：

```
ibusb: error [submit_bulk_transfer] submiturb failed error -1 errno=16
```

3. 正常过程图示

3.1 正常编程下载过程示例

典型 512K 容量下编程时间约 22 秒, 同容量可通过 -block 选项过滤空区域不参与编程。

```
Note->Welcome to Use ChipON KungFu32 Programmer Assistant!  
Note->begin Write .  
Time: 00:00:01    percent: 4      %  
Time: 00:00:02    percent: 8      %  
Time: 00:00:03    percent: 12     %  
Time: 00:00:04    percent: 16     %  
Time: 00:00:05    percent: 20     %  
Time: 00:00:06    percent: 24     %  
Time: 00:00:07    percent: 28     %  
Time: 00:00:08    percent: 32     %  
Time: 00:00:09    percent: 36     %  
Time: 00:00:10    percent: 40     %  
Time: 00:00:11    percent: 44     %  
Time: 00:00:12    percent: 49     %  
Time: 00:00:13    percent: 54     %  
Time: 00:00:14    percent: 59     %  
Time: 00:00:15    percent: 64     %  
Time: 00:00:16    percent: 69     %  
Time: 00:00:17    percent: 74     %  
Time: 00:00:18    percent: 79     %  
Time: 00:00:19    percent: 84     %  
Time: 00:00:20    percent: 89     %  
Time: 00:00:21    percent: 94     %  
Time: 00:00:22    percent: 100    %  
Note->Write finish!
```

3.2 正常编程读取过程示例

```
Note->Welcome to Use ChipON KungFu32 Programmer Assistant!  
Note->begin Read .  
Time: 00:00:01    percent: 7      %  
Time: 00:00:02    percent: 17     %  
Time: 00:00:03    percent: 27     %  
Time: 00:00:04    percent: 37     %  
Time: 00:00:05    percent: 47     %  
Time: 00:00:06    percent: 57     %  
Time: 00:00:07    percent: 67     %  
Time: 00:00:08    percent: 78     %  
Time: 00:00:09    percent: 88     %  
Time: 00:00:10    percent: 100    %  
Note->Read finish!
```

4. 维护记录

文档版本	日期	变更	备注
1.0.1	2021-01-04	首版本	
1.0.2	2021-03-29	读校验失败的地址与内容的显示匹配调整	
1.0.3	2021-4-11	更新-pid -replace 的参数细节描述 软件更新匹配芯片的最佳编程模式	
1.0.4	2021-5-14	修正加载 s19 文件编程，添加-copy 选项 实现文件格式转换	
1.0.5	2022-5-18	编程模式、读取模式时打印芯片唯一 id 值 增加独立读取 uid 进程功能 增加整体擦空进程功能(附加特征值查空) 增加查空进程功能 增加校验进程功能	更新时软件版本 1.0.6
1.0.6	2023-1-9	增加快速检测参数-fastcheck 适配擦除、查空、校验、编程校验的模式。 部分描述添加 linux 支持特性。	更新时软件版本 1.0.9
1.0.7	2023-12-2	增加 A158 系列编程时配置默认安全启动行为的功能参数选项	更新时软件版本 1.0.9.4