

## TG71XX\_Programmer 命令行使用说明

在命令行模式下，可以通过后接指定的参数来运行本程序以进行烧入指定的 hex（或 hexf）文件，或者将几个指定的文件合并成 hexf 文件（参照 2.6 节相关内容）。另外也可以选择既执行合并又执行烧写（实际执行顺序为先合并后烧写）。如果在命令行下不加参数打开本程序则会直接进入 GUI 模式。

本程序所支持的命令行参数如下（注意：参数除了 -a 要跟在 -r 后面外（具体参见 -a 参数的说明），其他参数理论上没有顺序先后的要求，另外每一个参数和它的缩写形式完全等价，可任意进行相互替换）：

参数	缩写	值	说明
--combine	-c	无	用来指示程序进行 hex 文件合并生成 hexf 文件，该参数不需要跟随一个值。 <b>生成的 hexf 文件默认根据 app 的文件名来确定</b> ，即如果 app 文件为 app.hex，则合并后的 hexf 文件和 app 在同一个目录并且文件名为 app.hexf 该选项和 -w 选项必须至少存在一个（可以同时存在）
--boot	-b	OTA Bootloader 文件 hex 格式	只能包含一个 OTA Bootloader 程序，如果为 No OTA 模式则该参数不是必要参数，其他情况下合并生成 hexf 文件时该参数为必要参数 参数为字符串，可以包含路径，路径有空格需要把整个字符串通过双引号包起来，和选项之间通过空格符号分开 例如：--boot ota.hex 以及 -b "c:\data app\ota.hex"
--app	-p	App 文件 hex 格式	只能包含一个 APP 固件程序，该参数是必要参数 参数为字符串，可以包含路径，路径有空格需要把整个字符串通过双引号包起来，和选项之间通过空格符号分开 例如：-p app.hex
--res	-r	Resource 文件 bin 或 hex 格式	可以包含不止一个 resource 文件（但是至多 3 个），该参数不是必要参数 参数为字符串，可以包含路径，路径有空格需要把整个字符串通过双引号包起来，和选项之间通过空格符号分开 如果为 bin 格式文件则需要在其后用 -a 选项指定它的起始地址（16 进制），如果为 hex 格式文件则不需要（如果 bin 文件后没有接起始地址则会报错并退出） 例如：-r res.bin -a 70000（即从 0x70000 地址开始写入）或者 -r res.hex

--addr	-a	bin 格式的 Resource 文件的 写入起始地址	起始地址为 <b>16 进制格式</b> ，数值为 Flash 从 0 开始的偏移地址，数据大小为文件大小，本程序会对数据长度以及地址的合法性做检查，若为非法地址或者出现地址重叠则会提示错误， 例子见上 (hex 文件自带地址信息所以不需要该选项来对其起始地址进行设定)
--mode	-m	合并模式	合并模式需要为以下 5 个中的一个：SH (Single Has FCT), SN (Single No FCT), DH (Dual Has FCT), DN (Dual No FCT), NO (No OTA), 若为其他值则报错，该参数是必要参数 例：-m DH
无	-e	数据区保留模式	该参数不是必要参数， <b>若无该参数则默认为 Flash 全擦除</b> 如果有该参数，则表示 Erase 除该区域之外的 Flash 区域。参数为 Offset，起始地址默认为 0x11000000 例：-e 4000-4fff 表示为保留 0x11004000 ~ 0x11004fff 数据区域
--write	-w	写入的文件 hex 或 hexf 格式	该选项和 -c 选项必须 <b>至少存在一个</b> （可以同时存在），若同时存在则先执行 -c 选项进行合并再执行该选项进行写入，即可以同时使用这两个选项来进行合并后写入 参数为字符串，可以包含路径，路径有空格需要把整个字符串通过双引号包起来，和选项之间通过空格符号分开 例如：-w target.hexf <b>注意：在写入开始前程序默认会进行一次擦除操作！</b>
--uart	-u	更新波特率	该参数不是必要参数，若无该参数，默认波特率为：115200；波特率值设置有：1000000、500000、250000、115200、76800、38400、9600 等，根据 uart 配置可进行修改 例：-u 500000
--Run	-R	Base run address (1FFF4000-- PHY6202, 1FFF4800-- PHY6212)	该参数是必要参数，需要修改该值对不同的芯片类型的固件进行烧写；通过该命令进行烧写工具中 configuration—base run address 中的修改，对 hex 文件中的 bin 文件进行烧写的起始地址 例：-R 1FFF4000（对应 6202 的芯片），-R 1FFF4800（对应 6212 的芯片）

--Port	-P	获取的 Port 的 name	多个 uart 与主机相连，获取多个 uart 的 COM 口的名称，对指定的 COM 口的开发板进行固件烧写 例如：-P COM3
--config	-f	配置文件.csv	该参数不是必要参数，主要是用来烧写 MAC 地址，在 csv 文件中设置多行 12 位 mac 地址，通过 4000 和 4004 地址进行 mac 地址值的烧写（备注：有其他配置需求，也可自行添加，获取对应的写入地址 address 和 value） 例如：-f *.csv
--line	-l	配置文件的 line 值设定	该参数不是必要参数，配置文件中可输入多行信息，每次烧写，可以通过改变 line 的值来改变 mac 地址或者其他参数设置 例如：-l 3（这里是英文字符 l（小写 L），不是数字 1）
--help	-h 或 -?	无	显示帮助

注意：

- 1.烧写的 TG71XX\_Programmer.exe 工具、烧写固件、配置文件.csv 等需要在同一目录下
- 2.log 文件的保存时以追加的形式写入的。多次烧写流程会保存在一个 log 文件中，默认产生的 log 文件名为 zUpgrade.log

示例 1（全擦模式固件烧录）：

指令：**TG71XX\_Programmer.exe -P COM3 -u 1500000 -R 1FFF4800 -w**

**total\_image.hexf**

描述：指定 Uart 口为 COM5, 设置波特率为 1500000, 设置 run address 为 1FFF4800, total\_image.hexf (和主程序在同一目录) 写入到芯片中 (芯片需要通过串口和电脑连接), 写入前会自动执行擦除。(注意：TM 拉高, P24 和 P25 拉低)

效果截图

```
1 Current port: COM5
2 Current baudrate: 115200
3 Current stopBits: 1
4 Current parity: No
5 Serial opened!!
6 *****
7 Send erase successfully!
8 Receive #OK!
9 Erase successfully!
10 Receive #OK!
11 Send update command successfully!
12 Baud rate modified successfully!
13 Current baudRate: 1500000
14 Send cpnum successfully!
15 Receive #OK!
16 Receive >>: successful!
17
18 =====Write hexf File [01/10]=====
19 Send cpbin successfully!
20 UART RX ASCII:by hex mode:
21 Receive image request!
22 Send image successfully! Waiting to receive checksum...
23 Send checksum successfully!
24 UART RX ASCII:checksum is: 0x000001
25 #OK>>:
26 Receive #OK!
27 Receive >>: successful!
28
29 =====Write hexf File [02/10]=====
30 Send cpbin successfully!
31 UART RX ASCII:by hex mode:
32 Receive image request!
33 Send image successfully! Waiting to receive checksum...
34 Send checksum successfully!
35 UART RX ASCII:checksum is: 0x001ce548
36 #OK>>:
37 Receive #OK!
38 Receive >>: successful!
```

## 示例 2（保留数据区模式固件烧录）：

指令：**TG71XX\_Programmer.exe -P COM5 -u 1500000 -R 1FFF4800 -e 4000-4fff -w total\_image.hexf**

描述：指定Uart口为COM5, 设置波特率为1500000, 设置run address为1FFF4800, 保留数据区 0x11004000 ~ 0x11004fff, 将 total\_image.hexf（和主程序在同一目录）写入到芯片中（芯片需要通过串口和电脑连接），写入前会自动执行除保留区外的 Flash 擦除。

效果截图

```
70 Send erase successfully!
71 Receive #OK!
72 Erase successfully!
73 Preserve erase successfully!
74 Receive #OK!
75 Send update command successfully!
76 Baud rate modified successfully!
77 Current baudrate: 1500000
78 Send cpnum successfully!
79 Receive #OK!
80 Receive >>: successful!
81
82 =====Write hexf File [01/10]=====
83 Send cpbin successfully!
84 UART RX ASCII:by hex mode:
85 Receive image request!
86 Send image successfully! Waiting to receive checksum...
87 Send checksum successfully!
88 UART RX ASCII:checksum is: 0x000001be
89 #OK>>:
90 Receive #OK!
91 Receive >>: successful!
92
93 =====Write hexf File [02/10]=====
94 Send cpbin successfully!
95 UART RX ASCII:by hex mode:
96 Receive image request!
97 Send image successfully! Waiting to receive checksum...
98 Send checksum successfully!
99 UART RX ASCII:checksum is: 0x001ce548
100 #OK>>:
101 Receive #OK!
102 Receive >>: successful!
103
104 =====Write hexf File [03/10]=====
105 Send cpbin successfully!
106 UART RX ASCII:by hex mode:
107 Receive image request!
```

注：

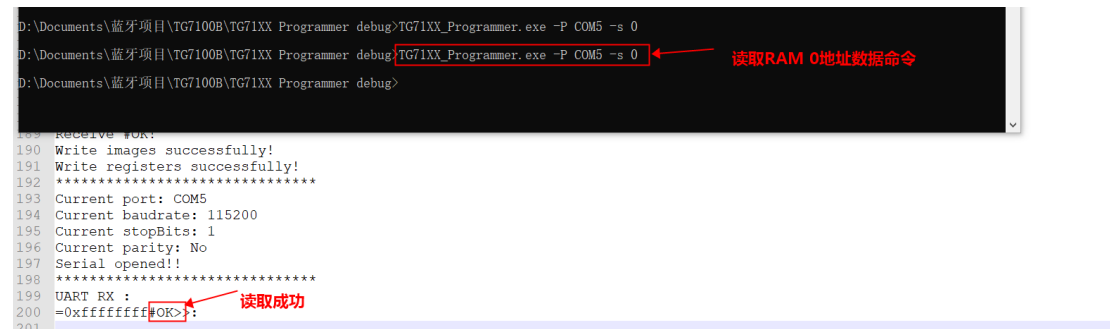
由于PC机串口驱动程序限制，存在不支持波特率为1500000的问题，此时可以降低波特率来适配，例如设置波特率为115200进行烧录。

### 示例 3（读取 RAM 0 地址的数据）：

指令：**TG71XX\_Programmer.exe -P COM5 -s 0**

描述：指定 Uart 口为 COM5，默认波特率为 115200， 读取 RAM 0 地址的值。通常用于判断芯片是否已上电并正常连接。

效果截图



```
D:\Documents\蓝牙项目\TG7100B\TG71XX Programmer debug>TG71XX_Programmer.exe -P COM5 -s 0
D:\Documents\蓝牙项目\TG7100B\TG71XX Programmer debug>TG71XX_Programmer.exe -P COM5 -s 0
D:\Documents\蓝牙项目\TG7100B\TG71XX Programmer debug>

189 RECEIVE OK!
190 Write images successfully!
191 Write registers successfully!
192 *****
193 Current port: COM5
194 Current baudrate: 115200
195 Current stopBits: 1
196 Current parity: No
197 Serial opened!!
198 *****
199 UART RX :
200 =0xffffffff#OK>:
```

读取RAM 0地址数据命令

读取成功