



WizPro 系列烧写器控制开发说明

RV5

1 烧写器监控命令说明

1.1 通讯方式:

- 烧写器采用 USB 转 UART 串口;
- UART 参数设定: 1M 波特率, 8-Bit Data, 无校验, 2-Bit Stop;

1.2 数据包的结构:

- 一个完整数据包为: 0xC0 + 数据 + 0xC0;
- 若数据是 0xC0, 则转换成: 0xDB+ 0xDC;
- 若数据是 0xDB, 则转换成: 0xDB+ 0xDD;

1.3 数据的组成:

1.3.1 数据分类:

- 一类是控制烧写器得数据, 有应用程序发给烧写器;
- 一类是返回数据, 是烧写器对命令的响应或者自动向外输出;

1.3.2 控制数据:

- 复位编程器: 0xC0+0x8A+'REST'+0xA8+0x02+0x00+0x00+0x00+0xCD+0x02+0xC0; 烧写器收到该命令后, 系统复位, 进入上电类似模式, 烧写器首先进行数据的自检, 以判断下载数据是否正常, 之后输出相关的信息和数据包, LED 指示相关状态; 。
- 启动烧写命令: 0xC0+0xF9+"APRG"+0x9F+0x00+0x00+0xE3+0x4D+0xC0;
- 获取烧写通道和预置产品名称信息: C0 F3 0C 52 44 4D 43 00 00 9E 61 C0;

1.3.3 烧写器返回数据: 详见表 2.

- 烧写器过程中状态数据: 固定长度为 9 个字节:

0xFD + 状态码 1 (1B) +状态码 2 (1B) +数据(4B) + Check Sum (1B);



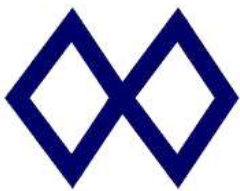
- 其中 CheckSum 为前面 8 个字节的累加和的补码，也就是该 9 个字节累加和为 0；

符表：返回数据说明：该数据是自动输出，用户 APP 只要去监控该输出即可，不同产品输出会有差异，用户可以根据实际需要来解析处理即可，不关心信息输出可以忽略。每次启动烧写时，0x10,0xF0/F1/F2 状态一定时有输出的，0x10 表示建立芯片通讯，F0/F1/F2 则是烧写结果输出；

状态码 1	状态码 2	数据	说明
0x0A	0x00/01/0x03	NoData	Power On Self Cehck
0x10	0x00	Current Page + Total page	进入烧写模式，当前芯片 Page 数据
	0x01	CRC Sum(2B) + ACC Sum(2B)	进入烧写模式，已下载数据的校验和
	0x02	ACC Sum (32-Bit)	进入烧写模式，已下载数据的校验和
0x1E	00	SN Hexdecimal Data	返回当前烧写的 SN 码数据 4 字节
0x1F	5 th ASCII	SN ASCII 数字字符数据（5 个字	如：“1F3534333231”即是字符“51234
0x18	0x00	设置允许烧写的最大数量	
0x1A	0x00	当前的已烧写计数值	
0x20	0x00	Current Page + Total page	执行芯片查空
0x23	0x00	Current Page + Total page	执行芯片数据检查
0x25	0x00	Chip ACC32 Sum	回传读取的芯片数据 ACC32 Sum
0x2F	Numb	Current Page + Total page	执行芯片 Block # 查空
0x30	0x00	Current Page + Total page	执行芯片擦除
0x3F	Numb	Current Page + Total page	执行芯片 Block # 擦除
0x40	0x00	Current Page + Total page	写芯片主 Flash 数据区
0x49	0x00	4Byte Data	返回原始数据计算的 ACC/CRC
0x4A	0x00	Current Page + Total page	写芯片 Customer/User 数据区
0x4E	0x00	Don't Care	恢复芯片校准数据
0x50	0x00	Current Page + Total page	写芯片 EEP 或用户数据区
0x56	0x00	Current Page + Total page	写芯片 BOOT 数据区
0x70	0x00	Reserved Data 4 Bytes	校验主数据区数据
0x73	0x00	读取的芯片 ACC16/CRC16 Sum	返回主数据区数据的校验和
0x75	0x00	Reserved Data 4 Bytes	Verify Ext. Flash memory



0x79	0x00	读取的芯片 ACC/CRC Sum	返回主数据区数据 ACC/CRC Sum
0x7F	Numb		执行芯片 Block # 校验
0x80	0x00	Current Page + Total page	校验 EEP 或用户数据区数据
0x83	0x00	EEP 数据的 ACC16/CRC16 Sum	返回 EEP/用户数据区数据校验和
0x85	0x00	EEP 数据的 ACC32 Sum	返回 EEP/用户数据区数据校验和
0x8F	0x00	CRC16/ACC16	输出当前数据计算校验和（原始数据）
0x8F	0x01	CRC16/ACC16	输出芯片回读数据计算校验和
0x90	0x00	Current Page + Total page	开始写 HSE 到 Chip Flash
0x95	0x00	Reserved Data 4 Bytes	开始安装 HSE 固件
0x9A	0x00	Reserved Data 4 Bytes	开始 HSE 状态检测
0x9D	0x00	Reserved Data 4 Bytes	指示 HSE 已安装好
0x9F	0x00	Reserved Data 4 Bytes	指示 HSE 安装完成
0xA0	0x00	Current Page + Total page	写芯片配置数据
	0x0F	烧写的序列号数据	写芯片配置数据
0xB5	0x00	Program NVM	开始写 NVM 数据
0xB5	0x03	Total Programmed NVM Bytes	写了 NVM 数据总字节数
0xB7	0x00	Verify NVM	开始校验 NVM 数据
0xB7	0x03	Total Verified NVM Bytes	校验了 NVM 数据的总字节数
0xE0	0x00	Current SN	返回芯片校验和数据
0xEA	Numb		Start Block # Programming
0xF0/0xF1	0x06	芯片校验和数据（CRC + ACC）	烧写完成，成功
0xF0/0xF1	0xFF	芯片校验和数据（CRC + ACC）	烧写完成，失败，状态 2 为错误代码
0xF2	0x06/0xFF	芯片 ACC32/CRC32 数据(APP 选择)	烧写完成，06:OK/Else:Error
0xFA	0x00	0x50,0x43,0x53,0x52（PCSR）	芯片已拿开
0xFA	0x00	0x50,0x43,0x4E,0x49（PCNI）	座子里有芯片
0xFA	0x00	0x4C,0x4F,0x43,0x4B（LOCK）	座子不可用（Locked）
0xFA	0x01	0x57,0x50,0x4B,0x4F（WPKO）	编程器上电正常，可以烧写



0xFA	0x01	0x57,0x50,0x4E,0x47（WPGN）	编程器上电错误，数据异常，不能烧写
0xFA	0x03	0x43,0x45,0x52,0x52（CERR）	Command Error。
0xFA	0xA0	Data don't care(set 0x00000000)	Start to Key Unlock SDAP.
0xFA	0xA3	Data don't care(set 0x00000000)	Start to AES key Unlock.
0xFB	0x00	Chip 32-bits CheckSum Return	Passed from APP.
0xFB	0x03	File 32-bits CheckSum Return	Passed from APP
0xFA	0x02	Byte1: Has Chip(0x0:No, 0xF:Yes) Byte2: Prg OK(0x06)/Prg NG(0x6F) Byte3:SocketLock(7F)/No Lock(06)	Check All Status Return
0xFE	0x00	ARM Chip Error Data（4B）	Arm 芯片错误代码数据输出
0xFE	0x01	Read ARM Chip Data:4Bytes	ARM 芯片读数据输出

注：“0xFA+数据”为检查烧写器状态命令返回数据包；可以不用解析；

MD5 校验码输出（有些特殊的版本和型号才支持）：

FB F0 10 00 MD5 Data(Total 16 个字节)，例如：MD5=EA4ED1F73D865CA3E9721AA4F0D1CBB8，输出如下：
FB F0 10 00 EA 4E D1 F7 3D 86 5C A3 E9 72 1A A4 F0 D1 C B B8 (16 进制数据)。

示例说明：

- 复位编程器： 0xC0+0x8A+’REST’+0xA8+0x02+0x00+0x00+0x00+0xCD+0x02+0xC0;烧录器将复位自校验，并返回结果：
 - 编程器上电正常，可以烧写；
 - 编程器上电错误，数据异常，不能烧写；
- 启动烧写： 0xC0+0xF9+’APRG’+0x9F+0x00+0x00+0xE3+0x4D+0xC0; 烧录器将启动一轮烧写，并返回状态信息，见表 2，其中状态码 0x10 数据包可用于检查判断启动命令是否发送成功：
 - 进入烧写模式，初始化接口并执行一系列烧写操作；
 - 烧写 OK，返回成功信息；
 - 烧写失败，返回失败信息和错误代码；

```
> 11:02:51 0xFD 0xF0 0x06 0xEA 0xA4 0x1A 0x07 0x5E
> 11:02:51 Progrmming Completed : Succesed !Chip ACC16=0x071A,CRC16=0xA4EA.
```

图：烧录成功

```
> 14:18:42 0xFD 0xF0 0xE0 0x00 0x00 0x00 0x00 0x33
> 14:18:42 Progrmming Completed : Failed : Error=0xE0!
```



图：烧录失败

- Check 烧写器（有些型号可能没支持）：0xC0+0xF6+'CKPG'+0x6F+0x00+0x00+0x7A+0x3A+0xC0;
将返回烧写器状态返
 - 编程器上电正常，可以烧写；
 - 编程器上电错误，数据异常，不能烧写；

附表 1：一些特殊信息输出（如 NXP HSE 固件烧写）

状态码 1	状态码 2	数据	说明
0x90	0x00	4 字节数据，忽略	开始烧写 HSE 数据到芯片
0x95	0x00	4 字节数据，忽略	开始启动 HSE 数据的安装
0x9A	0x00	4 字节数据，忽略	开始检查 HSE 状态
0x9D	0x00	4 字节数据，忽略	HSE 已经安装并激活了
0x9F	0x00	4 字节数据，忽略	安装 HSE 固件成功
0xFE	0x00/01	4 字节数据	输出测试数据

注：关于数据包的转换示例：某些情况下,返回的数据包里可能存在 0xC0 和 0xDB 的情况,例如：
返回数据包为： C0 FD F0 06 DB DC 3E 05 20 EA C0，此时解析时需要将 DB DC 转换成实际数据 0xC0.

- 通道和预置产品名称命令返回信息：若返回数据长度大于 80 则解析方法 1，否则解析方法 2.
 - 解析方法 1：
 - ◆ 偏移量 8 是通道编号值，0-9；
 - ◆ 偏移量 26 开始为产品型号和版本信息，FF 结束；
 - 解析方法 2：
 - ◆ 偏移量 9 是通道编号值，0-9；
 - ◆ 偏移量 10-12 是名称信息，3 个字符；如：MWP 对应为 P(10th)-W(11th)-M(12th)
 - ◆ 偏移量 28 开始为产品型号和版本信息，FF 结束
 - 注：产品型号和版本信息也可以从第 13 个偏移量开始扫描，扫描到“Wiz”字符及后面就是。



2. 工程文件的下载： 工程文件的下载可在用户的 APP 中通过调用如下的函数方式来：（以 ST 芯片为例）

theApp := '[PATH]\WizProGenSTX.exe' // MaxWiz 应用程，不同厂家选择不同 APP，也可通过 INI 文件来指定

thePar := '工程文件名称（可带路径） OFF（此 Option 用于自动关闭 APP，若需要手工关闭 APP，则不用此参数）'

ShellExecute(theHandle, 'Open', theApp, thePar, NIL, SW_HIDE); 调用此函数即可启动 APP 自动下载工程文件，下载的通道如果在配置文件中指定，则按指定的通道下载，否则自动搜索可用的设备。下载的结果在可通过检测 ResDW.Txt 文件来判断；

命令行夏的输入如下：

```
Microsoft Windows [版本 10.0.26100.3194]
(c) Microsoft Corporation。保留所有权利。

D:\WizProGenX\PcApp_GenXSTx>D:\WizProGenX\PcApp_GenXSTx\WizProGenSTX.exe D:\Test\STM32F103X.PMW OFF
```

3. 用 MaxWiz 通用控制软件来实现对烧写器的控制；在批量烧写时，客户若不想自己去开发通讯来控制烧写器，可采用我司的命令行调用 APP（GenXxCmdLinM.exe）来实现烧写，并通过查询 APP 生成的 TXT 文件来获取完整的信息。具体参见“GenX_CmdLin 命令行参数使用说明”资料的说明；

4. 命令行参数加载数据和设置：有些 APP 支持，雨鞋还没支持，具体请联系我司相关人员确定。具体参数如下：

- -CP=xxxx: 设置芯片型号，例如-CP=FM33FH025， 设置芯片为： FM33FH025.
- -LF=xxxxx: 加载数据文件，带路劲，例如：“-LF=D:\DEMOHEX\FM33FH0xx-codeflash+optbytes.hex”
- -CPES: 设置为整片擦除模式，缺省是这个模式；
- -BKES: 设置成按实际数据文件大小来擦除，例如文件有效数据是 5K，则只擦除 5K 的内容.
- -3V: 设置接口电压输出为 3.3V， 缺省是 3.3V 输出；
- -5V: 设置接口电压输出为 5.0V；

以上控制和逻辑为我司通用操作方式，和具体芯片以及产品型号无关，用户只需开发一次，就可支持我司所有相关的产品。

开发中如有任何疑问或需求，可随时联系如下邮箱：alan@maxwiz.com.cn. 或电话： 13602669824，方先生，谢谢！