

# ATGM 模块 基于主机的在线

# 升 级 协 议

V2.4

2015 年 02 月 02 日

杭州中科微电子有限公司  
杭州市滨江区江南大道 3850 号创新大厦 10 楼  
传真：0571-28918101  
网站：<http://www.icofchina.com>

## 版本更新历史

| 版本  | 日期         | 更新内容                   |
|-----|------------|------------------------|
| 1.0 | 2013/05/08 | 初稿                     |
| 2.0 | 2013/05/15 | 修改固件升级命令，增加代码或数据烧写地址参数 |
| 2.1 | 2014/01/13 | 增加参数命令的解析              |
| 2.2 | 2015/01/07 | 修改数据包长度字段描述            |
| 2.3 | 2015/01/30 | 增加模块上电时间参数             |
| 2.4 | 2015/02/02 | 修改接收完的超时时间             |

技术支持：

0571-28918103

0571-28918126

【适用性】

本协议仅适用于 ATGM 定位模块的在线升级以及上电升级，以完成模块代码的更新。

【升级流程】

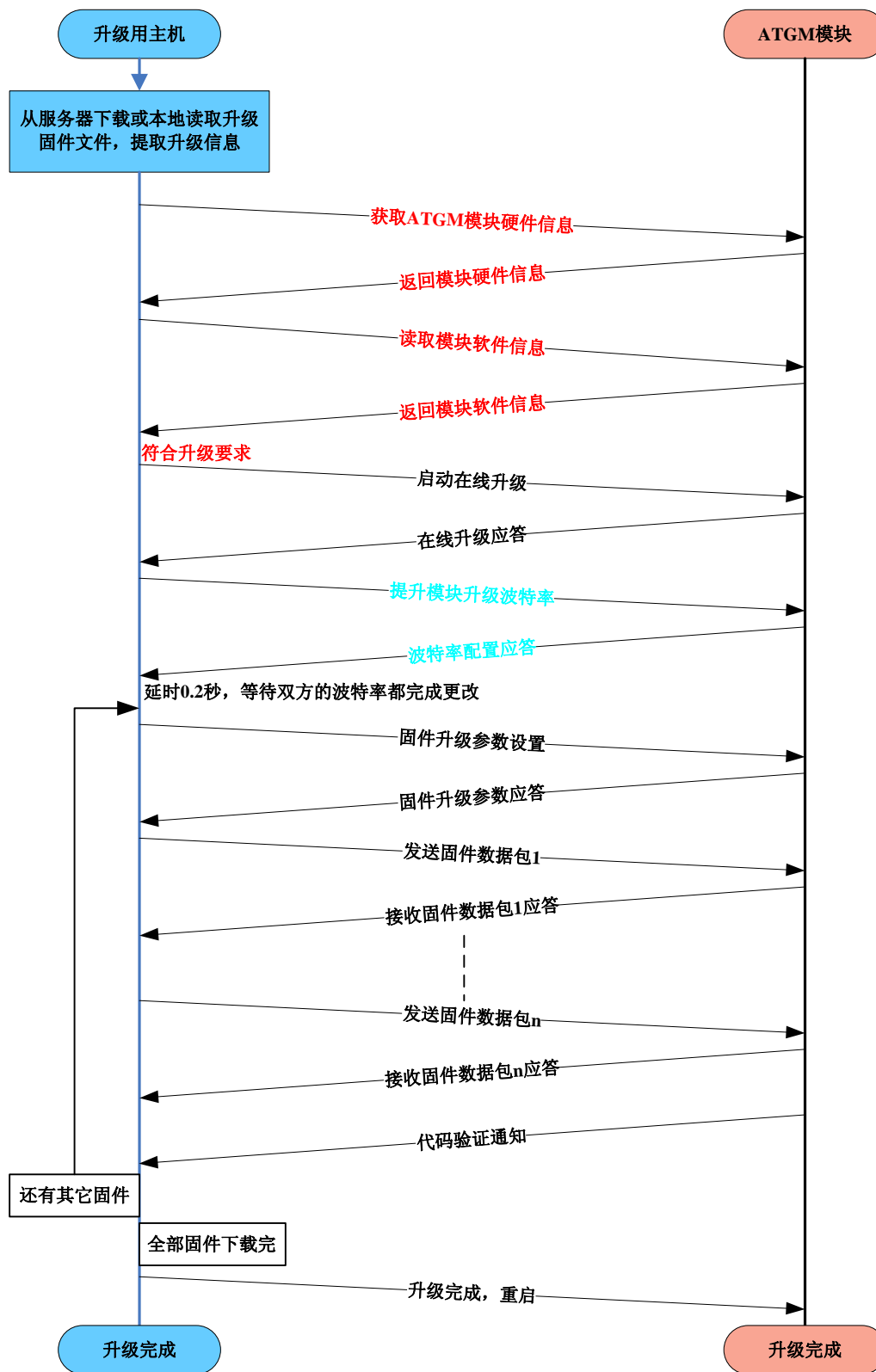


图 1 固件升级过程



固件升级过程中模块信息的获取(图中红色部分)不是必须的，一般情况下可以跳过。如果不考虑升级效率，提升波特率(图中青色部分)流程也可以跳过。当主机需要升级固件时，可以从服务器下载固件升级用文件，或者从本地文件存储介质中读取固件升级文件。附录 1 为固件升级文件的具体格式。

主机升级时可以采用如下的升级流程。

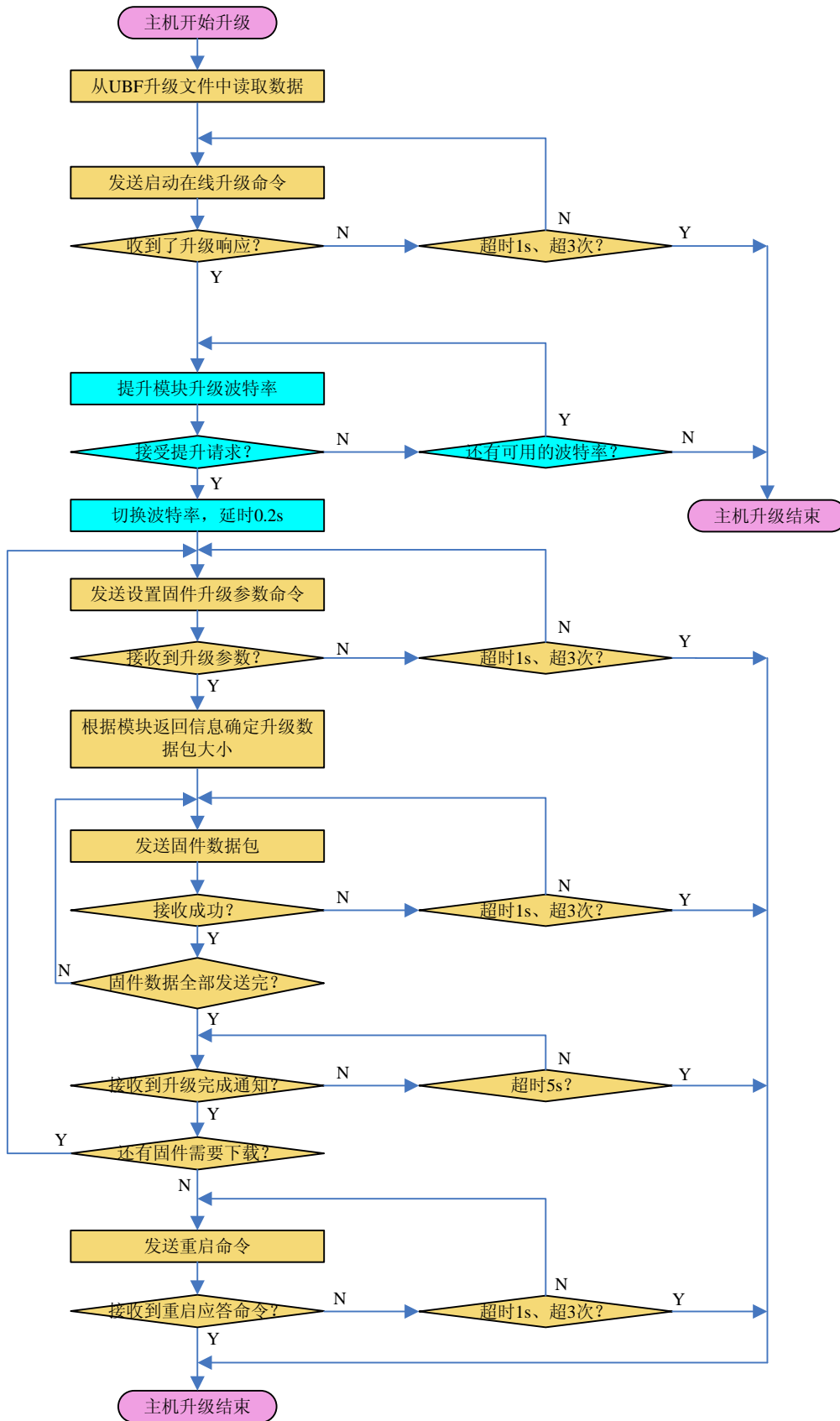


图 2 基于主机的固件在线升级参考流程图

其中青色部分要特别注意，确保通信的波特率始终保持一致。

**【升级命令】**

升级开始到提升模块波特率之间的命令采用本公司自定义的 NMEA 扩展命令。

**1、模块硬件信息查询命令，可选项**

主机发送：

\$PCAS06,1\*1A<CR><LF> // <CR><LF>分别表示回车符(0x0D)和换行符(0x0A)

模块返回：

\$GPTXT,01,01,02,HW=ATGM330B,0032011101304\*18<CR><LF>

其中的“ATGM330B”为模块型号，“0032011101304”为本模块编号。

**2、模块软件信息查询命令**

主机发送：

\$PCAS06,0\*1B<CR><LF>，可选项

模块返回：

\$GPTXT,01,01,02,SW=URANUS2,V2.0.5.0\*1B<CR><LF>

其中的“URANUS2”为软件代号，“V2.0.5.0”为软件版本号。

**3、在线升级启动命令，必选项**

主机发送：

\$PCAS20\*03<CR><LF>

模块返回：

\$PCAS30,3\*1D<CR><LF>

主机接收到上述升级响应命令后，开始使用下面定义的 AT-BUP 命令同模块进行后续的升级通信。



\*\*\*\*\*

### 1、提升模块升级波特率 —— 0x01

\*\*\*\*\*

为了提高模块升级时数据的下载速度，可以根据设备的实际能力，通过协商提高通信波特率。本协议规定可以使用的波特率为 9600、19200、38400、57600、115200。协商过程可以从高速到低速，逐个进行查询。本命令由主机发起。

主机：

|       |    |        |      |      |          |     |    |
|-------|----|--------|------|------|----------|-----|----|
| 字节数   | 1B | 2B     | 1B   | 1B   | 1B       | 1B  | 1B |
| 提升波特率 | 包头 | 0x0004 | 0x01 | 0x01 | Baudrate | 校验和 | 包尾 |

| Baudrate | 波特率(bps) | 说明 |
|----------|----------|----|
| 0x01     | 9600     |    |
| 0x02     | 19200    |    |
| 0x03     | 38400    |    |
| 0x04     | 57600    |    |
| 0x05     | 115200   |    |

如：DB 04 00 01 01 05 01 DE --- 提升波特率到 115200

模块应答：

|       |    |        |      |      |          |     |     |    |
|-------|----|--------|------|------|----------|-----|-----|----|
| 字节数   | 1B | 2B     | 1B   | 1B   | 1B       | 1B  | 1B  | 1B |
| 提升波特率 | 包头 | 0x0005 | 0x01 | 0x01 | Baudrate | ACK | 校验和 | 包尾 |

其中的 Baudate 同前，ACK 解释如下

| ACK  | 说明                                       |
|------|--|
| 0x00 | 配置成功                                     |
| 0x01 | 不支持，配置失败，下调波特率，最低到 9600。否则，退出升级，发送重启模块命令 |
| 0x10 | 命令错误，需要重发。否则，退出升级，发送重启模块命令               |

如：DB 05 00 01 01 05 00 00 DE --- 允许按 115200 进行升级

模块的应答命令使用主机发出请求时的波特率进行回应，一旦应答配置成功，即发送完本应答命令，模块就自行切换到主机通知的波特率，进入后续的升级流程。所以，主机在收到配置成功的应答命令后，也要同步切换到新的波特率进行通信，否则，无法完成后续的所有操作。

本命令不是必须的，如果模块的导航命令已经以 115200bps 输出，可以直接跳过进入下面的固件升级参数设置阶段。



\*\*\*\*\*  
**2、设置固件升级参数 —— 0x02**  
 \*\*\*\*\*

主机通知模块当前待升级固件的长度，模块返回可以支持的最大数据包长度。

主机：

|        |    |        |      |      |          |        |       |     |    |
|--------|----|--------|------|------|----------|--------|-------|-----|----|
| 字节数    | 1B | 2B     | 1B   | 1B   | 2B       | 4B     | 4B    | 1B  | 1B |
| 设置升级参数 | 包头 | 0x000d | 0x01 | 0x02 | CodeType | Length | StAdd | 校验和 | 包尾 |

CodeType 为代码类型，见下表

| CodeType | 说明     |
|----------|--------|
| 0x0001   | 导航代码   |
| 0x0002   | 升级代码自身 |
| 0x0003   | 工作配置参数 |
| 其它       | 无效     |

Length 为固件长度，单位为字节数

StAdd 为固件在 Flash 中的起始地址。当 CodeType=3 时, StAdd=0x3e000; 其它值时, StAdd=0

如：DB 0D 00 01 02 01 00 E0 F7 01 00 00 00 00 00 19 DE --- 导航代码长度为 0x0001f7e0

模块应答：

|      |    |        |      |      |       |     |     |    |
|------|----|--------|------|------|-------|-----|-----|----|
| 字节数  | 1B | 2B     | 1B   | 1B   | 2B    | 1B  | 1B  | 1B |
| 升级参数 | 包头 | 0x0006 | 0x01 | 0x02 | MaxPk | ACK | 校验和 | 包尾 |

MaxPk 为模块支持的最大数据包长度 (MaxLength = MaxPk + 9)，即包头后长度参数的极限值。表示主机发送的数据包长度不能超过 MaxPk，但也不要太小，否则会影响到下载效率。

ACK 为命令响应状态

| ACK  | 说明                         |
|------|----------------------------|
| 0x00 | 支持当前固件的升级                  |
| 0x01 | 代码类型错误                     |
| 0x02 | 代码长度错误，代码长度必须小于 256K       |
| 0x10 | 命令错误，需要重发。否则，退出升级，发送重启模块命令 |

如：DB 06 00 01 02 00 20 00 25 DE --- 模块支持的最大数据包为 0x2000 字节

每个用户的 MCU 配置不同，能够支持的数据包长度也不一样，这就需要一个协商的过程。本命令就是模块通知主机，模块本身的数据包接收发送能力。

\*\*\*\*\*

### 3、发送固件数据包 —— 0x05

\*\*\*\*\*

主机根据模块的 MaxPk 和自身能力，将固件文件分成 N 个数据包进行传输。

主机：

| 字节数   | 1B | 2B     | 1B   | 1B   | 2B      | 2B   | 2B     | nB     | 1B  | 1B |
|-------|----|--------|------|------|---------|------|--------|--------|-----|----|
| 传输数据包 | 包头 | Length | 0x01 | 0x05 | TotalPk | PkNo | PkSize | PkData | 校验和 | 包尾 |

说明：Length = n + 9，以小端格式表示，以下 2 字节整数的表示方法相同。

TotalPk 为总的数据包数量

PkNo 为当前数据包序号，从 1 开始

PkSize 为当前数据包(PkData)的实际大小

PkData 为 n 字节的数据

如：DB 09 08 01 05 3F 00 03 00 00 08 0B 48..... A9 53 DE --- 共发送 0x003f 个数据包，现在是在第 0x0003 个，实际包大小为 0x0800，0B 48..... A9 为数据

模块应答：

| 字节数     | 1B | 2B     | 1B   | 1B   | 2B   | 1B  | 1B  | 1B |
|---------|----|--------|------|------|------|-----|-----|----|
| 传输数据包应答 | 包头 | 0x0006 | 0x01 | 0x05 | PkNo | ACK | 校验和 | 包尾 |

说明：PkNo 为当前数据包序号，错误时为 0

ACK 为命令响应状态

| ACK  | 说明                         |
|------|----------------------------|
| 0x00 | 接收到正确的数据包                  |
| 0x01 | 程序代号、版本号等参数错误              |
| 0x02 | 版本号没有变化。                   |
| 0x10 | 命令错误，需要重发。否则，退出升级，发送重启模块命令 |

如：DB 06 00 01 05 03 00 00 01 DE --- 模块接收到第 0x0003 个数据包，数据完整。

注意，除了最后一个剩余数据包的大小不同外，其余数据包的大小必须相同。

当返回参数为 0x02 时，表明固件版本号没有变化。本返回参数是在模块接收到 8KB 数据以后给出的，仅出现 1 次，用户可以自行决定是否继续进行升级。如果版本相同不需要升级可以直接发送重启模块命令，终止整个升级流程。如果版本相同需要继续升级，可以发送后续数据包即可完成强制升级。



\*\*\*\*\*

#### 4、升级完成，重启模块 —— 0x06

\*\*\*\*\*

当主机将全部代码或数据都下载到模块后，完成全部升级工作，重启模块。

主机：

|      |    |        |      |      |     |    |
|------|----|--------|------|------|-----|----|
| 字节数  | 1B | 2B     | 1B   | 1B   | 1B  | 1B |
| 重启模块 | 包头 | 0x0003 | 0x01 | 0x06 | 校验和 | 包尾 |

如：DB 03 00 01 06 04 DE

模块应答：

|        |    |        |      |      |     |     |    |
|--------|----|--------|------|------|-----|-----|----|
| 字节数    | 1B | 2B     | 1B   | 1B   | 1B  | 1B  | 1B |
| 重启模块应答 | 包头 | 0x0004 | 0x01 | 0x06 | ACK | 校验和 | 包尾 |

说明：

ACK 为命令响应状态

|      |                   |
|------|-------------------|
| ACK  | 说明                |
| 0x00 | 接收到正确的重启命令        |
| 0x10 | 命令错误，需要重发。否则，退出升级 |

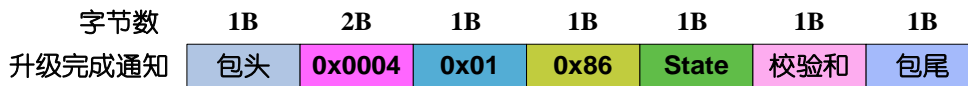
如：DB 04 00 01 06 00 03 DE --- 接收到正确的重启命令

本命令可以在不断电的情况下重新启动模块，运行升级后的代码。

\*\*\*\*\*  
**5、固件升级完成通知 —— 0x86**  
 \*\*\*\*\*

模块完成代码烧写后向主机发送固件升级完成通知。如果升级失败，主机也必须发送重启模块命令，延时 1s，然后再重新启动升级过程。

模块：



说明：State 为升级完成情况，见下表

| State | 说明         |
|-------|------------|
| 0x00  | 升级成功       |
| 0x01  | 接收到的代码数据错误 |
| 0x02  | 代码烧写错误     |
| 0x03  | 代码验证错误     |

如：DB 04 00 01 86 00 83 DE

主机应答：

收到本通知后，如果没有后续的代码或数据需要下载，主机可以向模块发送重启命令。如果主机还有另外的代码需要下载，可以从【设置固件升级参数】开始重复之后的固件更新步骤。

如果升级失败，主机可以在模块重启后再次启动升级流程。

**后面的命令仅适用于上电升级模式。**



\*\*\*\*\*

### 6、通知模块的基带 ID —— 0x81

\*\*\*\*\*

该命令在上电升级时使用，在线升级时本命令无效。

模块：

| 字节数    | 1B | 2B     | 1B   | 1B   | 4B | 1B  | 1B |
|--------|----|--------|------|------|----|-----|----|
| 通知基带ID | 包头 | 0x0007 | 0x01 | 0x81 | ID | 校验和 | 包尾 |

主机应答：

无应答命令。



\*\*\*\*\*

7. 通知主机，模块已经准备好上电升级 —— 0x82

\*\*\*\*\*

该命令在上电升级时使用，在线升级时本命令无效。

模块：

|               |    |        |      |      |      |     |    |
|---------------|----|--------|------|------|------|-----|----|
| 字节数           | 1B | 2B     | 1B   | 1B   | 1B   | 1B  | 1B |
| 通知主机已经准备好上电升级 | 包头 | 0x0004 | 0x01 | 0x82 | RomU | 校验和 | 包尾 |

说明：RomU 为上电升级类型，见下表

|      |             |
|------|-------------|
| RomU | 说明          |
| 0x00 | Flash 型上电升级 |
| 0x01 | Rom 型上电升级   |
| 其它   | 未定义         |

主机应答：

无应答命令。



\*\*\*\*\*

### 8. 获取模块名称和编号 — 0x10

\*\*\*\*\*

该命令在上电升级时使用，在线升级时本命令无效。

主机：

|           |    |        |      |      |     |    |
|-----------|----|--------|------|------|-----|----|
| 字节数       | 1B | 2B     | 1B   | 1B   | 1B  | 1B |
| 获取模块名称及编号 | 包头 | 0x0003 | 0x01 | 0x10 | 校验和 | 包尾 |

模块应答：

|           |    |        |      |      |     |     |      |     |    |
|-----------|----|--------|------|------|-----|-----|------|-----|----|
| 字节数       | 1B | 2B     | 1B   | 1B   | 4B  | 4B  | 16B  | 1B  | 1B |
| 返回模块名称及编号 | 包头 | 0x001b | 0x01 | 0x10 | Sn1 | Sn2 | Name | 校验和 | 包尾 |



\*\*\*\*\*

### 9、烧写模块名称和编号 —— 0x11

\*\*\*\*\*

该命令在上电升级时使用，在线升级时本命令无效。

主机：

| 字节数       | 1B | 2B     | 1B   | 1B   | 4B  | 4B  | 16B  | 1B  | 1B |
|-----------|----|--------|------|------|-----|-----|------|-----|----|
| 烧写模块名称及编号 | 包头 | 0x001b | 0x01 | 0x11 | Sn1 | Sn2 | Name | 校验和 | 包尾 |

说明：

Sn1 为生产加工识别号

Sn2 为模块序号，0~99999

Name 为模块名称，占用 16 字节，不足部分添 0

模块应答：

| 字节数       | 1B | 2B     | 1B   | 1B   | 1B    | 1B  | 1B |
|-----------|----|--------|------|------|-------|-----|----|
| 烧写模块名称及编号 | 包头 | 0x0004 | 0x01 | 0x11 | State | 校验和 | 包尾 |

说明：

| State | 说明                |
|-------|-------------------|
| 0x00  | 模块名称和编号烧写成功       |
| 0x01  | 模块名称和编号烧写错误       |
| 0x10  | 命令错误，需要重发。否则，退出升级 |



**【附录 1】固件升级二进制文件格式**

固件升级二进制文件用于 GNSS 模块的在线升级，后缀名为“ubf”，格式说明如下：

1. +0x00  
文件头，占 2 个字节，固定为 0x41,0x54(字符“AT”的 ASC 码)
2. +0x02  
固件升级用二进制代码长度 firmLength，不包含后面的 4 字节固件校验和，占 4 个字节，以小端 int 形式保存
3. +0x06  
固件烧写到 Flash 中的起始地址，4 个字节，以小端 int 形式保存
4. +0x0A  
升级用二进制代码在本文件中的起始地址 CS，占 4 个字节，以小端 int 形式保存
5. +0x0E  
代码类型，占 2 个字节，以小端 short 形式保存。1—导航代码；2—升级代码
6. +0x10  
模块型号，占 16 个字节，以字符串形式保存，不足部分填 0。如：“ATGM330B”
7. +0x20  
代码版本，占 16 个字节，以字符串形式保存，不足部分填 0。如：“V2.0.5.0”
8. +0x30  
原始代码文件名称，占 128 个字节，以字符串形式保存，不足部分填 0。如：  
“Uranus2\_CM3\_ATGM330B....bin”
9. +0xb0  
原始代码生成日期及时间，占 32 个字节，以字符串形式保存，不足部分填 0。如：  
“2013-05-09 12:00:00”
10. +0xd0  
留空，代码日期后与代码文件前的空白。
11. +CS  
升级用二进制代码
12. +CS + firmLength  
升级用二进制代码的 32 位 xor 校验和(xor4\_checksum)，占 4 个字节，以小端 int 形式保存。  
目的是验证固件数据的有效性。

如果有多个模块需要升级，可以在后面重复 1~12 项参数。目前有 3 类固件数据：

- 1、导航代码，完成定位功能的主代码
- 2、自举代码，完成导航代码加载、固件升级的代码
- 3、工作参数，用户定制的一些特有参数，如导航命令格式、NMEA 的输出波特率等。

| 起始地址   | 字段名称      | 字节  | 数据类型    | 说明   |
|--|-----------|-----|---------|--|
| 0x00   | 文件头       | 2   | 字符      | 固定为 0x41,0x54(“AT”的ASC码)   |
| 0x02   | 固件长度      | 4   | Int 型   | 升级用固件长度(firmLength = N, 不包含后面的 4 字节固件校验和)                                  |
| 0x06   | 固件烧写起始地址  | 4   | Int 型   | 固件在 Flash 中的起始地址<br>导航代码 --- 0x8000<br>升级代码 --- 0x0000<br>工作参数 --- 0x3e000 |
| 0x0a   | 固件起始地址 CS | 4   | Int 型   | 升级用二进制固件在*.ubf 文件当前块中的起始地址   |
| 0x0e   | 固件类型      | 2   | Short 型 | 1—导航代码, 2—升级代码, 3—工作参数   |
| 0x10   | 模块型号      | 16  | 字符串     | 模块型号的字符串,如:ATGM330B  |
| 0x20   | 代码版本      | 16  | 字符串     | 代码版本的字符串,如:V2.0.5.0  |
| 0x30   | 代码名称      | 128 | 字符串     | 原始代码的文件名   |
| 0xb0   | 代码日期      | 32  | 字符串     | 原始代码的生成日期、时间   |
| 0xd0   | 留空        | 不定  | N/A     | 代码日期后与代码文件前的空白,填 0   |
| CS   | 固件        | N   | 二进制     | 升级用二进制固件   |
| CS+N   | 固件校验和     | 4   | Int 型   | 升级用二进制固件的 4 字节异或和(xor4_checksum), 用以判断本固件文件是否合法有效。                         |
| 如果有多个固件(代码或数据), 可以重复前面从文件头到固件校验和的所有字段, 即相当于将几个 ubf 文件简单合并起来使用。 <b>注意, 下一个固件紧跟在前一个之后。</b> |           |     |         |  |
| CS+N +4  | 文件头       | 2   | 字符      | 固定为 0x41,0x54(“AT”的ASC码)   |
| CS+N +6  | 文件长度      | 4   | Int 型   | 升级用二进制代码长度(codeLength)   |
| ....   |           |     |         |  |

代码 4 字节异或和 xor4\_checksum 计算方法.

二进制代码数组 char \*buffer

二进制代码长度 firmLength

```
int xor4_checksum = 0;
```

```
int iTmp;
```

```
for(int i=0; i<firmLength/4; i++)
```

```
{
```

```
    memcpy(&iTmp, buffer+i*4, 4);
```

```
    xor4_checksum ^= iTmp;
```

```
}
```

## 固件升级代码实例分析

参考 UBF 文件：V2420.ubf

固件文件头数据结构：

|          | 0  | 1  | 2  | 3  | 4  | 5  | 6  | 7  | 8  | 9  | A  | B  | C  | D  | E  | F  | 0123456789ABCDEF |       |
|----------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|------------------|-------|
| 00000000 | 41 | 54 | E0 | F7 | 01 | 00 | 00 | 80 | 00 | 00 | 00 | 01 | 00 | 00 | 01 | 00 | AT.....          |       |
| 00000010 | 41 | 54 | 47 | 4D | 33 | 33 | 31 | 43 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | ATGM331C.....    |       |
| 00000020 | 56 | 32 | 2E | 34 | 2E | 32 | 2E | 30 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | V2.4.2.0.....    |       |
| 00000030 | 55 | 72 | 61 | 6E | 75 | 73 | 32 | 5F | 43 | 4D | 33 | 5F | 41 | 54 | 47 | 4D | Uranus2_CM3_ATGM |       |
| 00000040 | 33 | 33 | 31 | 43 | 5F | 56 | 32 | 34 | 32 | 30 | 5F | 47 | 50 | 53 | 42 | 44 | 331C_V2420_GPSBD |       |
| 00000050 | 53 | 5F | 43 | 54 |    |    |    |    |    |    |    |    | 00 | 00 | 00 | 00 | S_CTTIC.bin..... |       |
| 00000060 | 00 | 00 | 00 | 00 |    |    |    |    |    |    |    |    | 00 | 00 | 00 | 00 | .....            |       |
| 00000070 | 00 | 00 | 00 | 00 |    |    |    |    |    |    |    |    | 00 | 00 | 00 | 00 | .....            |       |
| 00000080 | 00 | 00 | 00 | 00 |    |    |    |    |    |    |    |    | 00 | 00 | 00 | 00 | .....            |       |
| 00000090 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | .....            |       |
| 000000A0 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | .....            |       |
| 000000B0 | 32 | 30 | 31 | 33 | 2D | 31 | 32 | 2D | 31 | 31 | 20 | 30 | 39 | 3A | 34 | 34 | 2013-12-11 09:44 |       |
| 000000C0 | 3A | 35 | 32 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | :52.....         |       |
| 000000D0 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | .....            |       |
| 000000E0 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | .....            |       |
| 000000F0 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | .....            |       |
| 00000100 | 48 | AD | 04 | 10 | 99 | 01 | 00 | 10 | B5 | 09 | 00 | 10 | BB | 09 | 00 | 10 | H.....           |       |
| 00000110 | C1 | 09 | 00 | 10 | C7 | 09 | 00 | 10 | CD | 09 | 00 | 10 | 00 | 00 | 00 | 00 | .....            |       |
| 00000120 | 00 | 00 | 00 | 00 | 00 |    |    |    |    |    |    |    | 00 | D9 | 09 | 00 | 10               | ..... |
| 00000130 | D3 | 09 | 00 | 10 | 00 |    |    |    |    |    |    |    | 10 | E5 | 09 | 00 | 10               | ..... |
| 00000140 | 09 | 0A | 00 | 10 | E7 |    |    |    |    |    |    |    | 10 | F7 | 09 | 00 | 10               | ..... |
| 00000150 | F9 | 09 | 00 | 10 | FB |    |    |    |    |    |    |    | 10 | FF | 09 | 00 | 10               | ..... |
| 00000160 | 01 | 0A | 00 | 10 | 03 |    |    |    |    |    |    |    | 10 | 07 | 0A | 00 | 10               | ..... |
| 00000170 | 0B | 0A | 00 | 10 | 77 | 0A | 00 | 10 | 79 | 0A | 00 | 10 | 7B | 0A | 00 | 10 | ...w...y...{...  |       |

导航代码文件名

固件数据

固件文件尾数据

|          | 0  | 1  | 2  | 3  | 4  | 5  | 6  | 7  | 8  | 9  | A  | B  | C  | D  | E  | F  | 0123456789ABCDEF |        |
|----------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|------------------|--------|
| 0001F870 | 44 | 00 | 03 | 10 | 04 | AD | 01 | 00 | 10 | 01 | 00 | 10 | 1C | 00 | 00 | 00 | D.....           |        |
| 0001F880 | 43 | 00 | 00 | 00 | F8 |    |    |    |    |    |    |    | 00 | 0E | 00 | 00 | 00               | C..... |
| 0001F890 | 0F | 00 | 00 | 00 | 2E |    |    |    |    |    |    |    | 03 | 03 | 00 | 00 | 00               | .....  |
| 0001F8A0 | 00 | 00 | 00 | 00 | 00 |    |    |    |    |    |    |    | 00 | 00 | 00 | 00 | 40               | .....@ |
| 0001F8B0 | FF | 00 | 00 | 00 | 09 |    |    |    |    |    |    |    | 0E | 06 | 01 | 08 | 07               | .....  |
| 0001F8C0 | 06 | 01 | 08 | 07 | 0C |    |    |    |    |    |    |    | 0A | FF | FF | 00 | 00               | .....  |
| 0001F8D0 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | F7 | EE | BD | 58 | .....X           |        |
| 0001F8E0 | 5F | DD | 6F | 10 |    |    |    |    |    |    |    |    |    |    |    |    |                  | ...o.  |

固件数据

固件长度：0x0001F7E0

固件在 Flash 中的烧写起始地址：0x00008000

固件在本文件中的起始地址：0x00000100

固件类型：0x0001 --- 导航代码

模块型号：“ATGM331C”

代码版本：“V2.4.2.0”

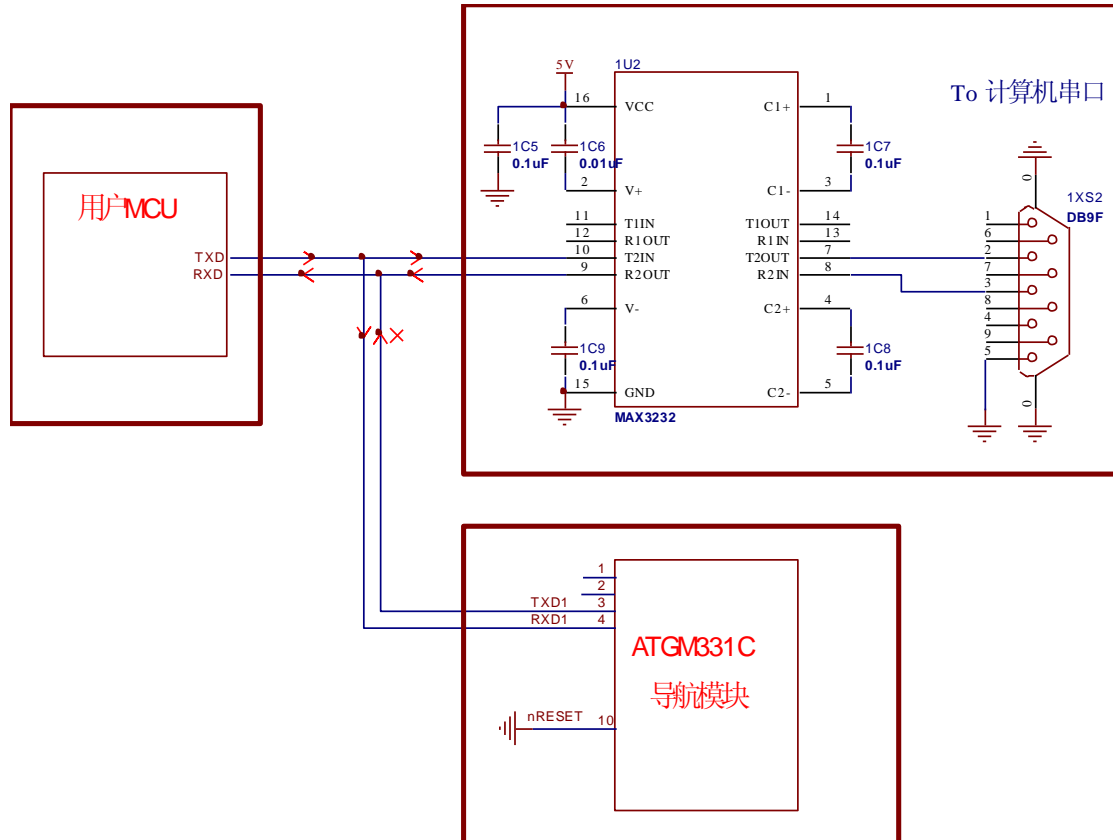
代码名称：“Uranus2\_CM3\_ATGM331C\_V2420\_GPSBDS\_CTTIC.bin”

代码生成日期：“2013-12-11 09:44:52”

固件数据：从 0x00000100 开始存放，共 0x0001F7E0 字节

固件校验和：0x106FDD5F

为了便于客户完成在线升级程序的设计调试，我们建议利用没有安装导航模块的平台或者直接将模块的 nRESET 引脚接地(目的是将模块的 TXD1 输出屏蔽掉),把用户的 MCU 串口通过 RS232 电平转换芯片连接到计算机,利用我公司提供的导航模块模拟软件完成升级的所有流程。我们提供的在线升级模拟软件可以实时显示升级过程,能极大地提高调试速度。



杭州中科微电子有限公司还提供模拟主机的基于 PC 机在线升级软件,用户可以通过虚拟串口工具(如 VSPD XP 5)将 2 个模拟软件互连,实现计算机模拟整个在线升级过程,让客户对升级流程有个比较全面的了解。

**导航模块模拟器v1.0 - 杭州中科微电子 2013-04-07**

**参数设置**

串口号: COM6  
 正常工作波特率: 9600  
 升级模式波特率: 115200  
 数据包最大长度: 8KB  
 当前版本号: 0.0.0.0

NMEA格式版本查询消息: (填入\$符和\*符之间的部分)  
 NMEA格式版本查询应答: (填入\$符和\*符之间的部分)

代码文件: F:/Casic/GPS/CASIC 基带芯片/ATGB03/CodeOnlineUpdate/test.bin 选择

显示数据传输消息  禁止数据包应答  保存文件 停止模拟

**串口数据**

```

24 50 59 41 58 4F 4E 30 30 2A 31
31 0D 0A AA 75 01 81 00 85 00 00
00 01 05 F2 AA 75 01 81 00 91 00
00 00 04 00 01 F7 DC 14 AA 75 01
81 00 93 00 00 00 01 01 E8 AA 75
01 81 00 94 00 00 00 01 01 E7
          
```

**升级状态**

```

数据应答: 当前包号: 54.
检测到数据包. 信息长度: 2264, 总包数: 58, 当前包号: 55, 长度: 2252字节.
数据包应答. 当前包号: 55.
检测到数据包. 信息长度: 2264, 总包数: 58, 当前包号: 56, 长度: 2252字节.
数据包应答. 当前包号: 56.
检测到数据包. 信息长度: 640, 总包数: 58, 当前包号: 57, 长度: 628字节.
数据包应答. 当前包号: 57.
发送固件检查结果: 检验成功.
检测到固件检查结果应答.
发送固件升级结果: 升级成功.
检测到固件升级应答.
固件升级完成, 本次耗时: 2秒. 返回正常状态.
          
```

升级计数: 3次      计时: 66秒

**车载主机模拟器v1.1 - 杭州中科微电子系统部 内部版本 2013-06-13**

**参数设置**

串口号: COM5  
 正常模式波特率: 9600  
 升级模式波特率: 115200  
 当前数据包长度: 2.20KB  
 最大重发次数: 3

代码文件: vdeOnlineUpdate/20131211发布版本/Uranus2\_CM3\_ATGM331C\_V2420\_GPSBDS\_CTTIC.bin 选择

强制升级  原始文件模式 生成代码

重发数据包  自动测试 开始升级

**串口数据**

```

30 35 AA 75 01 81 00 12 00 00 00 05 01 00 00
00 31 34 AA 75 01 81 00 12 00 00 00 05 01 00
00 00 32 33 AA 75 01 81 00 12 00 00 00 05 01
00 00 00 33 32 AA 75 01 81 00 12 00 00 00 05
01 00 00 00 34 31 AA 75 01 81 00 12 00 00 00
05 01 00 00 00 35 30 AA 75 01 81 00 12 00 00
00 05 01 00 00 00 36 2F AA 75 01 81 00 12 00
00 00 05 01 00 00 00 37 2E AA 75 01 81 00 12
00 00 00 05 01 00 00 00 38 2D AA 75 01 81 00
12 00 00 00 05 01 00 00 00 39 2C AA 75 01 81
00 13 00 00 00 01 01 68 AA 75 01 81 00 14 00
00 00 01 01 67
          
```

**升级状态**

```

-----
Sending code file...
Progress: 100.00%
Send size: 128992bytes.
Waiting for data check:
Data checked!
Send data check result ack.
Waiting for update result:
Update succeeded!
Send update result ack.
Cost: 2secs
Update finished!
          
```

**【使用注意事项】**

- 1、模块上电后，0.5 秒左右就可以输出模块的基本信息。如果用户设备 MCU 的启动时间比较长，往往会错误这些信息，当确实需要这些信息时，可以在设备 MCU 完成所有初始化后，根据需要向模块发送冷启动或者是热启动命令，模块会重启。

```
$GPTXT,01,01,02,MA=CASIC*27
```

```
$GPTXT,01,01,02,HW=ATGM331C,0012345612345*18
```

```
$GPTXT,01,01,02,IC=ATGB03+ATGR201*71
```

```
$GPTXT,01,01,02,SW=URANUS2,V2.3.2.0*1F
```

```
$GPTXT,01,01,02,TB=2013-09-16,09:08:34*42
```

```
$GPTXT,01,01,02,MO=GB*77
```

```
$GPTXT,01,01,02,CI=00000001*7B
```

```
$GPTXT,01,01,02,BS=SPI_BootLoader,V5.0.3.3*22
```

具体含义可以参见模块的详细使用说明。

- 2、一旦启动模块升级流程，各个命令的发送间隔时间必须小于 7 秒，否则，模块会认为是超时而自动退出升级模式，重启后进入导航模式，定时输出 NMEA 导航电文。
- 3、建议采用数据重传机制，可以极大改善升级的可靠性。如果 1 秒(数据发送完为 5 秒，因为烧写代码需要 3、4 秒)之内模块没有返回正确的数据包，重新发送原来的数据，重复 3 次还是无果才退出升级流程。