※ 原文刊登在《QEX》杂志 2012 年第 11/12 月刊

APRS 揭秘

作者 / 鲍伯 · 西蒙斯（Bob Simmons，WB6EYV）
编译 / 翁凯（BA5AG）

编者按：APRS 是自动位置报告系统的英文名 Automatic Position Reporting System 的缩写，是一项新兴的业余无线电活动内容，近年来在国内逐渐流行。本文就为您讲述关于 APRS 消息的比特级别的秘密……

任何想要“从头”开始制作一个 APRS 设备的人，都会立刻面临一个问题：找不到完整的、详细的关于所有的 APRS 的资料。要找到这些详细资料需要超乎寻常的努力，而不仅仅是有一点点“运气”就可以；除此之外还需要一份术语表，显然一个 APRS“初学者”是不会有这样的术语表的。缺乏一个简单、精确而且完整的资料，可能已经令很多试图创造 APRS 新技术的人受到了挫折。

本文提出了这个问题（希望如此），逐个字节地给出了一个典型的 APRS 消息字符串的详细描述，并提供了完整的例子。基本上，这就是足够的“消息协议”信息，能用于创造新的电路，可以插入一个不支持 APRS 的电台的话筒插座，然后产生能成功地在一个 APRS 网络中传播的 APRS 包。

这里描述的内容已经经过了测试和验证。作者用它们开发了一个 2m APRS 信标发射机，已经用了两年了，还被其他人使用过了。关于这个信标本身，你可以在作者的网站：http://www.silcom.com/~pelican2/PicoDopp/XDOPP.htm#MBCN 找到更多的信息。

Bell 202调制方式

一般的 APRS 数据都是以 1200 波特率在 144.390MHz 的（美国）全国 APRS 频道上发射的。数据是以调频方式调制在射频载波上的，调制成两个音频音调（1200/2200Hz），符合修改版的 Bell 202 调制解调器标准。在发送单个数据比特的时候，一个逻辑“零”比特是在两个音调间“翻转”（比如 1200 变成 2200 或相反）来表示的，而一个逻辑“一”比特是以不“翻转”（稳定的频率，要么 1200Hz 要么 2200Hz）来表示的。

Bell 202 标准要求音调之间的“翻转”必须是“相位连续”的，大概就是说在不同的音调之间的发送必须要尽可能地平滑。音频波形的相位角（在音频切换的瞬间的幅度）必须保持，并用作下一个音频波形的“起点”。这样就减弱了开关瞬态效应，降低了信号所需的带宽，从而提高了信噪比（SNR）。图 1 是一个调制波形的例子。

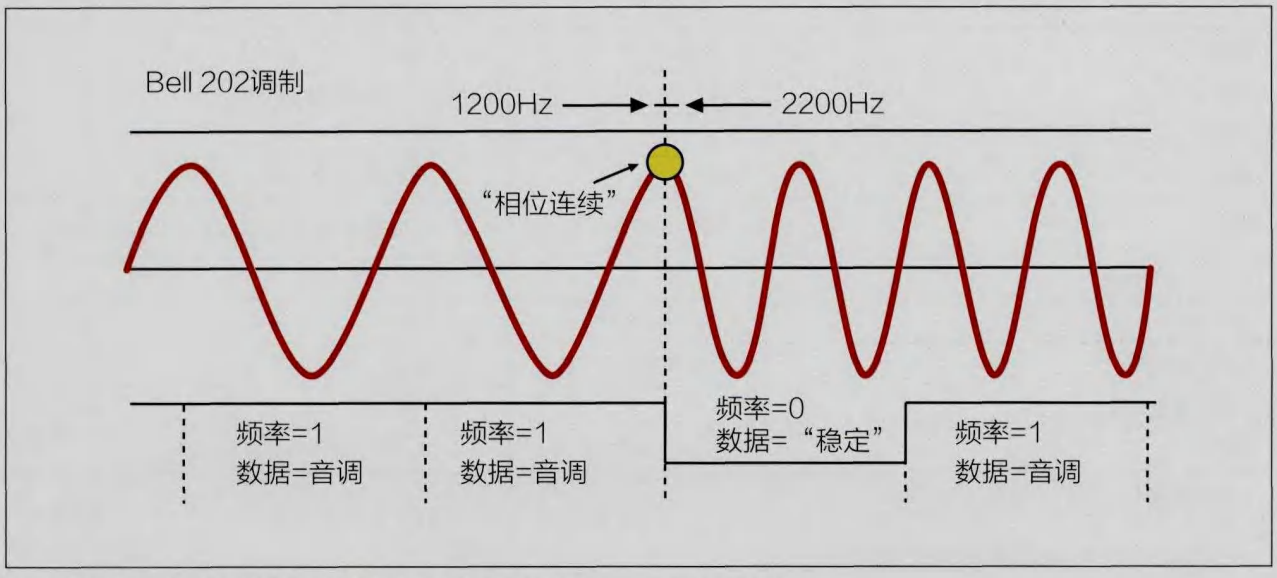

图 1 此图描述了 Bell 202 调制解调器标准对表达一个 0 比特时在两个音调之间翻转的调制要求

表1 例子APRS消息

名称	值	发送出去的十六进制数据
(开始发送)		
空	(5个<null>)	0x00 0x00 0x00 0x00 0x00
标志	(5个<波浪号>)	0x7e 0x7e 0x7e 0x7e 0x7e
(CRC校验从这里开始计算)		
(字节填充从这里开始)		
(注意：下面的字节做了左移一位以保证第零位=扩展位)		
目的地址	APDF00	0x82 0xa0 0x88 0x8c 0x60 0x60
目的SSID	<SSID=0>	0x60
源地址	W6XYZ<空格>	0xae 0x6c 0xb0 0xb2 0xb4 0x40
源SSID	<SSID=15>	0x7e
转发1	WIDE1<空格>	0xae 0x92 0x88 0x8a 0x62 0x40
转发1SSID	<SSID=1>	0x62
转发2	WIDE2<空格>	0xae 0x92 0x88 0x8a 0x64 0x40
转发2SSID	<SSID=2>	0x65
(左移结束)		
控制字符	<控制>	0x03
协议字符	<协议>	0xf0
消息类型	<消息类型>	0x21
纬度	3426.22N	0x33 0x34 0x32 0x36 0x2e 0x32 0x32 0x4e
符号表	<主表>	0x2f
经度	11943.57W	0x31 0x31 0x39 0x34 0x33 0x2e 0x35 0x37 0x57
符号码	<小汽车>	0x3e
方向	264	0x32 0x36 0x34
分隔符	/	0x2f
速度	000	0x30 0x30 0x30
注释	COMMENT	0x43 0x4f 0x4d 0x4d 0x35 0x4e 0x54
(CRC计算结束)		
CRC 低字节	<CRC低字节>	0xf9
CRC 高字节	<CRC高字节>	0x3c
(字节填充结束)		
标志	(5个<波浪号>)	0x7e 0x7e 0x7e 0x7e 0x7e
(发送结束)		
(总计发送81字节，以1200波特率要540ms)		

八比特还是字节?

APRS 消息字节发送的时候没有 START（开始）或 STOP（结束）比特（每个字节叫做一个八比特 octet），这与传统的 RS232 字节不同。大多数的消息数据是用普通的 ASCII 字符编码的，不过也有些例外的情况，后面再说。在任何情况下，一个字节（八比特）的数据的各个比特是从最小比特（LSB）开始的（比特的顺序是从 B0 到 B7）。

APRS位置报告：消息格式

本文给出的 APRS 例子消息符合 APRS 标准文档（版本 1.0.1，日期 2009 年 8 月 29 日）的第 33 页所标记的格式。在那份文档中（第 33 页），这个消息格式标记为：“经纬度位置报告格式，有数据扩展（无时间戳）”。每个字节的定义见表 2。

逐个字节的，发送出去的消息有如本页底部的示例。

要记得“在另一端”的接收机可能是一个普通的语音电台，有普通的静噪电路，会需要 50 到 100 毫秒（或更多）的时间来检测到频道上的信号，然后才会输出音频信号。在 1200 波特率上，100 毫秒的时间等于发送 15 个字节的数据……所以遵循这个典型的“友好”实践经验，也就是多发送一些起始标志字节，可能是比遵守 APRS 标准的定义更为重要。

■ 标志字节和比特填充

APRS 消息里没有起始和结束位，意味着必须有一些另外的方式来使得 APRS 解码器能与收到的消息中的比特流实现“同步”。

相邻的两个比特之间的边界可以通过观察 1200/2200Hz 音频来辨认，但是相邻的两个消息之间的边界（一个消息结束而另一个消息开始）就必须通过一些手段来得到。这是由一个特殊的叫做“标志”字节的东西实现的，它具有“01111110”这样的比特模式（十六进制的 0x7e=ASCII 码的“波浪号”字符：~）。

在 APRS 标准中，这个比特模式（6 个连续的“1”比特）是保留专属这个标志字节的。因此，任何“碰巧”出现的相同的字节（在发送的数据中），必须被识别出来并避免发送，但是数据本身（无论如何）又必须能在接收端被保护和识别出来。为了达到这个目的，就用了一种叫做“比特填充”的技术。

在“比特填充”中，每个要发出去的消息在发送之前都要被检查（一个比特一个比特的），以发现任何（不慎）出现的五个连续的“1”比特。如果这样的事情发生了，第六个比特（可能是 0 也可能是 1）就被推迟发送，先插入发送一个“0”比特（“填充”进数据流）然后再跟上原本（被推迟了）的第六个比特。

在接收消息的那端，发现五个连续的“1”比特就会提醒软件，接下去的（第六个）比特会决定这个数据是标志字节还是正常消息中的一个字节。如果第六个比特是“1”，那么这个字节就被判定为一个标志字节……否则的话，这第六个“0”会被忽略，从比特流里丢弃掉。

■ 前导：概述

消息的前导部分包括源地址、目的地址和转发地址字节，以及这些呼号关联的 SSID 字节。根据 APRS 标准，转发台站的数量（由用户指定）可以是零到八个，本文提供的例子里用了两个。

源地址实际上就是发射机操作员的 FCC 分配的呼号（6 个字符，总是以大写字母拼写），加上跟在后面的一个 SSID 字节（这样总共 7 个字节，SSID 的进一步资料在后面）。如果呼号少于 6 个字符，就左对齐，在 SSID 字节后面跟上几个 ASCII“空白”字符（0x20）。

目的地址并没有真正在 APRS 中使用……这是包封通信的过时传统，APRS 是包封的特殊子集，用不到这个数据字段。APRS 在这个字段填入特殊的字符串来指出用来产生这个 APRS 消息的软件。本文的例子中所给出的字符串是分配给一位软件作者鲍伯 · 布鲁尼加（Bob Bruninga）（APRS 的开发者）的，它的内容是“APDF00”，SSID 字符是 0。这个“分配”是一种社团礼仪，这样如果产生的 APRS 消息有任何的问题，就可以追踪到这个软件的作者，然后能得以纠正。任何想要创建自己软件的人都可以联系鲍伯 · 布鲁尼加来得到一个类似的分配。

转发呼号（以及它们的 SSID 字符）是可选的（本文的例子用了两个）。这些是补充性的名字，提供关于这个消息最好以怎样的路径或方向转发，以及消息最好能被发给谁。

常见的两个转发路径是“WIDE1-1”和“WIDE2-2”。

FDDDDDDdSSSSSSsVVVVVVvVVVVVVvCPsLLLLLLLHsLLLLLLLLHsCCCsSSSCccFF

表2

（开始发送）	（注意：发送一开始，音调调制就立刻开始了，没有“载波静默”的时间）	
起始标志字节	（最少一个字节，通常是几个相同的字节）	
F	标志字节，总是=0x7E	
前导字节		
DDDDDDd	目的地址	（7个字节）（本例为“APDF00”+SSID=“0”）
SSSSSSs	源地址	（7个字节=6个字节呼号+1个字节SSID）
VVVVVVv	转发地址1	（7个字节=6个字节呼号+1个字节SSID）
VVVVVVv	转发地址2	（7个字节=6个字节呼号+1个字节SSID）
可以有0到8个转发地址，本例中用了两个		
（前导字节结束）		
控制与协议字节		
C	控制字段	（1个字节，总是=0x03）
P	协议字段	（1个字节，总是=0xF0）
（控制与协议字节结束）		
信息字段	（APRS位置报告，没有时间戳，没有消息）	
s	符号	（1个字节，APRS消息类型标识符）（!=感叹号=没有APRS消息，没有时间戳）
LLLLLLLH	纬度	（8个字节，XXxx.xxH）（XX=纬度的度，00到89）（xx.xx=纬度的分，十进制表示）（H=半球，N或S）
s	符号	（1个字节，选择使用主或副APRS符号表）（这个符号决定了在APRS地图上显示的图标类型）
LLLLLLLLH	经度	（9个字节，XXXxx.xxH）（XXX=经度的度，000到179）（xx.xx=经度的分，十进制表示）（H=半球，E或W）
s	符号	（1个字节)（=APRS屏幕上显示的地图图标）
（扩展数据从这里开始：方向和速度）		
CCC	方向	（3个字节，xxx=001-359，旋转角度，000表示静止）
s	符号	（1个字节)（分隔符=“/”）
SSS	速度	（3个字节，xxx=000-999节）
（扩展数据数据）		
帧校验字节		
cc	FCS字段	（2个字节，CRC校验和，发送为低字节/高字节）
结束标志字节		
FF	标志	至少两个字节
（消息结束，发送结束）		

注意：为了对接收解码器友好，常常会在发送第一个标志字节之前，先发送几个字节的0x00。若干个连续的”0”比特会使得Bell 202音调持续地翻转，这样就使得在接收解码器里识别连续的数据比特之间的界限变得容易。为了类似的理由，也常常会在开始（以及结束）的时候发送几个标志字节，尽管APRS标准只要求一个标志字节。

ASCII “3”字符 = 0x33 = 00110011 旋转的字符 = 0x66 = 01100110（如果扩展位为 0）= 0x67 = 01100111（如果扩展位为 1）

这两个特殊的转发地址实际上是要求听到这个消息的任何数字中继自动“消息转发”这个消息。

■ 前导：SSID 字节

SSID 的意思是“台站第二识别码”（台站的第二个身份）。SSID 以单个字节编码，可以被解释为一个 0 到 15 的数字。在 APRS 标准里有几个不同的（有点复杂）“规则”说明如何选择 SSID 数字，不过实际的 SSID 值看起来消息在 APRS 网络的识别和传播中并不重要。本文的例子里，目标台站（= APDF00）的 SSID 是 0。对于 WIDE1 转发路径，SSID 应该是 1；而 WIDE2 转发路径，SSID 应该是 2。

有必要指出的是，在普通的计算机显示上（以及在印刷的文章中），SSID 值总是带着一个横线字符，所以“W6XYZ-0”表示 SSID 为 0 的 W6XYZ 台站，不过在实际发送的消息中，并不会发送这个横线字符。

进一步得指出，这个 SSID 字符并不是一个 ASCII 字符，ASCII 数字是一个编码在一个 8 位的字节中的 4 位二进制数字。剩下的比特是用作其他用途的，这些比特的描述如下：

- SSID 位 0：扩展位（= 1 表示最后一个前导字段，= 0 表示后面还有内容）
- SSID 位 1 到 4：台站第二识别码（0 到 15，=“SSID”数字）
- SSID 位 5 和 6：保留，始终为 1
- SSID 位 7：控制信息，C 比特，始终为 0

■ 前导：扩展位和字节旋转

APRS 标准允许一条消息中列零到八个转发台站，因此必须有办法来指出某条消息实际放了多少个转发台站（同时也标志了数据的前导块的结束）。这是由每个前导字节的最低位（LSB）来实现的。如果在一个前导字节的 LSB（= 第 0 位）为 0，那么消息中还有更多的前导字节（这个字节不是最后一个）；如果这一位为 1，那么就没有更多的前导字节了。这一位就是“扩展位”。

这个位在普通的 ASCII 编码中是有用的，因此并不能简单地用于这个目的。为了解决这个矛盾，在前导中只能使用 7 比特的 ASCII 码（最高位必须永远是 0）。这包括了所有“可打印”的 ASCII 字符，是整个 ASCII 集合的一个子集。

进一步的，每个 ASCII 字节（只有在前导中的）需要被“旋转左移”一个比特，也就等于把每个字符的二进制值乘以了二。这样做并不会丧失信息，因为所有 7 比特的 ASCII 字符的最高位永远等于 0。这样“旋转“以后的结果，每个前导字节的 LSB 就被“解放”出来，可以用做 APRS 的“扩展位”了。

比如，ASCII 字符“3”一般是被解析为一个十六进制数字 0x33，但是在一个 APRS 前导中，（由于做了字节旋转）发送出去的就是十六进制的 0x66（如果扩展位是 0）或十六进制的 0x67（如果扩展位为 1）。APRS 标准的第二部分的附录 3 里有一份 ASCII 字符的正常版和“旋转”版的列表。

这个“字节旋转”仅仅只用于前导字节——不是用在 APRS 发送的全部内容中的。第一个“旋转”的字节是目的地址的第一个字节，最后一个“旋转”的字节是最后一个转发地址的最后一个字节（最后一个转发地址的 SSID 字节）。如果没有用到任何转发地址，那么最后一个“旋转”的字节就是源地址的 SSID 字节。

总结一下，所有的前导字符的扩展位必须为 0，除了前导中最后的那个字节，那个字节的扩展位必须为 1。

■ 控制和协议字符

控制和协议字符包括紧接在前导之后发送的两个字节。首先发送的是控制字节，这个控制字节永远是 0x03。然后发送协议字节，也永远是 0xf0（这里就不解释为什么是这两个字节了）。

■ 消息体（信息字段）

消息体（在 APRS 协议里叫做“信息字段”）根据发送的 APRS 消息类型的不同而有不同的格式。第一个字符（“符号”）标识了这个字段里的数据的格式，不同的 APRS 消息在这个位置上有不同的字符（参见 APRS 标准）。

■ APRS 地图符号

APRS 地图符号是用消息体中的两个 ASCII 字节标识的。一个就紧跟在纬度字段后面，另一个紧跟在经度字段后面（尽管在汉语中我们习惯说经纬度，因为经是第一声而纬是第三声，但是在英语中他们习惯说纬度经度，因为纬度 latitude 在字典里排在经度 longitude 前面）。这两个字符代表的意义定义在 APRS 标准的附录 2 中。

第一个字符用以标识附录中的两张“符号表”中的一张

（主表或副表），每张表里面有 93 个“符号”，当消息被收到时，会显示在地图上。第二个字符就用来标识所对应的表中的 93 个符号中的哪一个。

■ 数据扩展

数据扩展是一个可选的 7 个字节的字段，如果用上的话，可以说明额外的信息，如 APRS 标准第七章所述。在这里的例子中，用了一个数据扩展字段来说明报告数据的台站的方向和速度。

■ 注释字段

注释字段是可选的。不同类型的 APRS 消息的注释字段的最大长度是不同的（见 APRS 标准中每个消息类型的细节描述）。

■ 帧校验和

帧校验和是用 CRC 计算方法计算的。CRC 的意思是“循环冗余校验”，计算的结果是两个特殊的“校验和”字节，能在消息被收到后，用于检验消息数据是否完整。CRC 校验和（ = 帧校验和）是在每个消息发送的时候产生的，然后在消息被收到后在接收方做校验。

CRC 校验和的产生是用一个特殊的“公式”对发送的消息中的每个字节做运算得到的。这个特殊的“公式”计算的结果是两个字节的数字，代表的是 CRC（帧）校验和。

作为“比特填充”加到比特流里去的比特是不包含在 CRC 校验和的计算里的。两个字节的校验和本身也是排出在计算之外的。

CRC 校验和计算是从前导块的第一个字节开始的（紧跟在最后一个起始标志之后），到注释块的最后一个字节为止。两个 CRC 字节以 LSB/MSB（先是低字节，再是高字节）的顺序发送。

我不想在这里再用作者自己的原话来重复解释 CRC 校验和计算，向大家介绍我自己学习这个计算的地方——非常感谢斯高特 · 米勒（Scott Miller，N1VG），他在他自己的网站贴出来了下文的 CRC 校验和计算方法的简明扼要的解释。

■ 帧校验序列

AX.25 格式有一个值得注意的细节是帧校验序列（FCS）校验和。这是加在每个帧后面的两个字节的校验和。它是用 CRC-CCITT 多项式产生的，发送的时候先发送低字节。

CRC-CCITT 算法已经有很多公开发表的例子代码了，不过我需要而且已经查过错的，是能一位一位地计算 FCS 而不是一次一个字节的算法。这个算法如下所述：

首先将一个 16 位的 FCS 设为 0xffff。对于发出去的每个数据位，右移这个 FCS 值一位。如果移出去的那一位（之前的第一位）不等于要发出去的那一位，则将这个 FCS 值与 0x8408 做异或。到了最后一个数据位，就将这个 FCS 值取补（反转）然后发送低字节。

以上这段文字（以及更多有用的信息）可以在斯高特 · 米勒的网站找到：http://n1vg.net/packet/index.php。

要是有谁仔细核对 1998 年 7 月版的 AX.25.2.2 的 AX.25 协议标准，就会发现在 3.8 节中，FCS 数据的位的发送顺序与数据包中其他数据的顺序是不同的。在标准中，FCS 应该要首先发送最高位（两个 FCS 字节以 B15 到 B0 的顺序发送），而数据包中所有其他字节是最低位先发送（位顺序 =B0 到 B7）。

但是这与作者的经验相违背，实际在空中成功的信标发射机的测试发出的（以及 iGate 所发出的）APRS 包中的 FCS 都是先发送最低位。在斯高特 · 米勒的相关说明里也没有提到这个“反转”的位顺序，因此 AX.25 标准在这点上是存疑的。

一个消息例子

表 1 中所给出的消息例子是一个完整的 APRS 消息，是以与本文所描述的指引一致的方式所产生的。为清楚起见，作为“比特填充”而加入的比特没有出现在数据中。由于有些字节包含不可打印的 ASCII 字符，因此数据以十六进制的形式标注。PE

作者简介：

鲍伯 · 西蒙斯（Bob Simmons，WB6EYV），1964年13岁时首次获得初学者执照，自此始终保持执照注册。他在1967年还获得了FCC的商业执照。他在美国海军预备役服役，是雷达技术员（ETR2），共有6个月的出海时间。他在洛杉矶港的民用航海电子行业工作了几年，又在航空界做了5年商业喷气机的工作。他于1992年移居加州的圣巴巴拉，在一家做真空蒸镀的企业做了5年，其他时间则从事各种奇奇怪怪的工程工作。

目前，鲍伯自己开业，开了一个网站，制作和销售无线电测向设备和模块，他的主要的“新”工作是创作嵌入式软件/硬件和研究能进行互联网连接的测向台站。他的主要兴趣是针对老问题研究和应用新技术，以及推进测向这门“艺术”。