

基于 CP16 的业余卫星通信系统

——电子系统设计结题报告

无 42 林子恒 2014011054

无 42 刘若洋 2014011051

无 42 王靖宇 2014011053

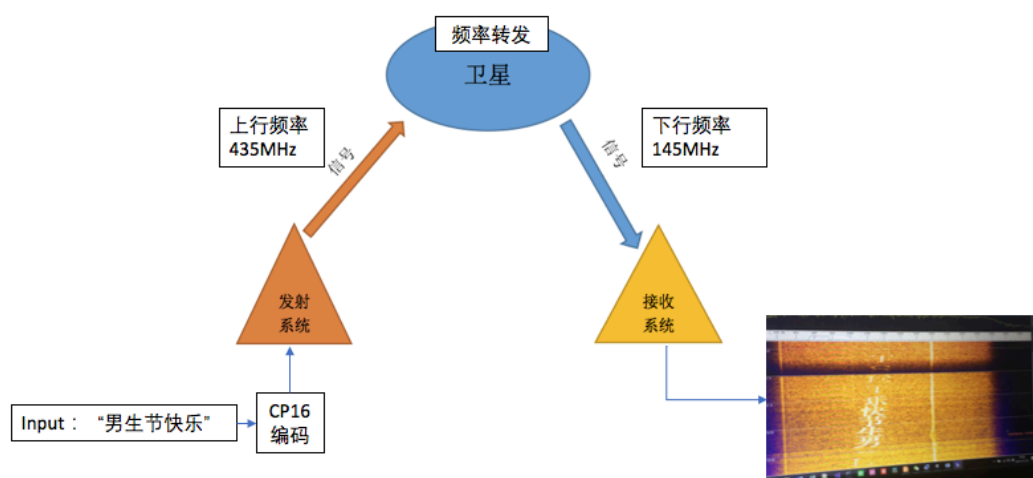
无 48 黄志超 2013012062

无 48 王 禹 2014011241

Table of Contents

一、	课题介绍	3
二、	系统设计	4
1	系统总体框架设计	4
2	各部分具体设计	5
2.1	CP16	5
2.2	UI	8
2.3	使用 GPREDICT 控制收发信机	10
2.4	自制模拟卫星	11
三、	系统结果展示	12
四、	总结与体会	12
五、	附录	13
1	设置收发信机频率使用说明	13
2	自制模拟卫星技术手册、使用说明	13

一、课题介绍



业余卫星通讯，简称 OSCAR (Orbiting Satellite Carrying Amateur Radio)，是把绕行于地球轨道上的人造卫星，当作无线电通讯的中继器，以达到远程通讯的目的。在本项目中，我们将使用希望二号系列卫星，传输一段经由 CP16 编码过后的文字，经过卫星中继重新接收后，恢复出所发射的文字。

二、系统设计

1 系统总体框架设计



本系统由以下几个部分组成，用户首先通过我们开发的 GUI 输入一段想要传输的文字，然后经过 CP16 编码模块后产生一段信号，这段信号再通过发射机调制到 435MHz 左右的频段，并发射向希望二号卫星。卫星接收到这段信号后，会将信号的频率转发至 145MHz 左右的频段，我们再通过接收机接受到这个信号，解调并绘制出瀑布图，即可恢复出用户传输的信息。

我们这个项目是基于二字班的学长学姐尹希玲、陈璋美、余金城、胡亚威、李昭奇在两年前所做的工作，他们在两年前的电子系统设计课程上，已经初步实现使用 Matlab 进行 CP16 编码，以及使用收发信机与卫星通信的功能，但他们仍存在许多问题，下面一一列出并给出我们的解决方案：

1. Matlab 来进行 CP16 编码操作复杂，需要每次执行完程序后保存下这段音频，再点击播放传入发射机中，需要一个用户友好的 GUI 来简化此流程
2. Matlab 来进行 CP16 编码性能不佳，现存也没有开源的高效实现，急需一份高效稳定的 C++ 实现来造福社会
3. 希望二号每天只有一个时段会经过北京的上空，需要一个自制模拟卫星来模拟其功能便与调试
4. 与卫星通信存在多普勒效应，若是手动调节收发频率太不科学，需要研究如何用软件自动根据多普勒频率控制收发频率。

2 各部分具体设计

2.1 CP16

2.1.1 CP16 编码介绍：

CP16 是一种新开发的无线电通信模式，是英文“ChinesePattern16”的缩写。工作方式采用由 16×16 点阵汉字信息直接对 16 个载频进行幅度调制（16ASK）后生成的基带信号，经由基带信号载入电台发送载有 16×16 点阵汉字字形信息的无线电信号；接收方将接收机收到解调后的基带信号载入电脑，经软件在显示器上的时间-频率瀑布流上依据信号有无显示出汉字字形从而达到汉字报文信息传递

CP16 的收发系统框图如下所示：

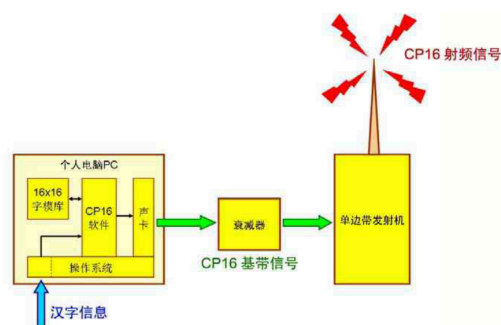


图1 试验系统配置（发射端）

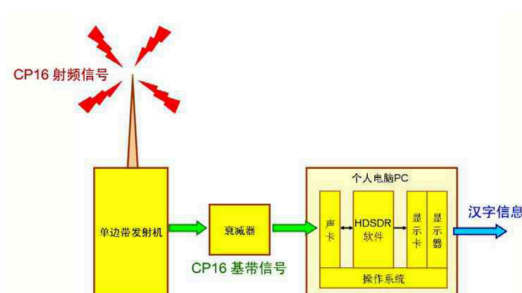


图2 试验系统配置（接收端）

CP16 具有明显优点有：

2.1.1.1 高抗噪能力

与任何其他依靠数字编码和严格的解码来传送文字的通信方法或者话音通信相比，CP16 应该具有显著的抗干扰优势。这是因为：

1. 由于发送端的汉字没有经过编码，接收端不需要进行解码，所以传送系统没有正确还原初始信息的误码率门槛；
2. 汉字信息的还原过程充分发挥了人类大脑的智慧。当接收端得到的 CP16 信号的信噪比较高时，应该可以从瀑布图得很清晰的汉字图形。当接收端信噪比降低时，得到带有噪点背景的汉字图形。当信噪比进一步恶化时，只要还能勉强从噪点背景中看出亮点，所传送的汉字还是可以有相当的可读性。即使字形中的某些像素缺损，但使用者还是有可能根据经验准确还原出原来的信息，这种智能判断能力是机器难以达到的；
3. 一般人的视觉对瀑布图响应比听觉语音响应灵敏，因此在同样微弱的信号条件下，CP16 的信息更容易被捕捉到；
4. 瀑布图的信息比话音暂留更长的时间，允许操作者有更多的时间进行信息匹配和检查，与话音通信相比，提高了信息传输的可靠度。

2.1.1.2 信号占用带宽仅 400Hz

CP16 的信号带宽设计为窄于 400Hz, 相当于 CW 人工莫尔斯电报的带宽。

2.1.1.3 允许普通接收机同时读取多路信号的汉字

由于 CP16 信号带宽比较窄, 带宽为 2.7kHz 的单边带话音信道可以同时容纳 6 路 CP16 信号, 而且汉字的显示不需要特别的解调。任何普通单边带接收机都可以把相邻的 6 路 CP16 射频信号搬移为音频范围的基带信号, 送入使用普通声卡将音频信号转换为频率-时间瀑布图显示的电脑, 就可以同时显示出 6 路汉字报文信息, 而不需要变换接收机的调谐。

这一功能对于应急通信具有积极的意义, 不但简化了操作, 而且可以使操作者同时监视多路通信, 了解全局。即使由于某种原因相邻信道的频带发生重叠干扰, 接收者也还是有可能较好地地区分出各自的报文。这些特点是其他汉字传输方式难以实现的。

2.1.2 核心编码程序的编写

2.1.2.1 语言和框架的选取：

在语言与框架的选取过程中, 我们选择了使用 C++ 进行核心编码程序的编写, 理由有二：一是因为 C++ 属于较为底层的编程语言, 在运行效率、速度以及封装完整性上比起 Python 和 Matlab 等脚本语言有着较大的优势；二是因为我们的程序需要进行 GUI 封装, 在进行 GUI 封装时, 为了配合 C# 的数据结构与格式, 使用 C++ 编程可以较大的减少工作量

2.1.2.2 hzk16 介绍

HZK16 字库是符合 GB2312 标准的 16×16 点阵字库, HZK16 的 GB2312-80 支持的汉字有 6763 个, 符号 682 个。其中一级汉字有 3755 个, 按声序排列, 二级汉字有 3008 个, 按偏旁部首排列。

一个 GB2312 汉字是由两个字节编码的, 范围为 A1A1~FEFE。A1-A9 为符号区, B0 到 F7 为汉字区。每一个区有 94 个字符（注意：这只是编码的许可范围, 不一定都有字型对应, 比如符号区就有很多编码空白区域）

在 hzk16 中, 汉字字符的位置应当为：区码=汉字的第一个字节-0xa0(160), 位码=汉字的第二个字节-0xa0(160)。因此汉字的 $offset=(94*(区码-1)+(位码-1))*32=(94*(第一字节-161)+(第二字节-161))*32$

对于半角的英文、数字字符, 其半角的位置, 与全角编码差距如下陈述：对于第一个字节, 所有半角字符转为的全角字符, 第一字节均为 0xa3。而第二字节的编码则是和半角编码相差 128。按照此规则, 则可以将半角的字符转化为全角字符。

而对于空格，其半角编码较为特殊为 0x20，可以直接将其映射为 0xa1, 0xa1，即全角的空格

2.1.2.3 编码核心部分（文本-波形转换库）

编码核心部分按照如下的示例图流程进行：

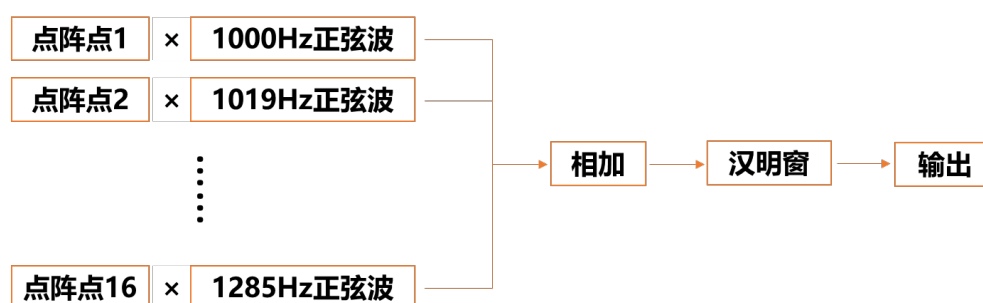


具体的代码编写方法，可以详见随附的代码。由上述汉字库的说明可以实现取址与汉字字形的编码，声音编码的具体实现，可以见下述相应板块。

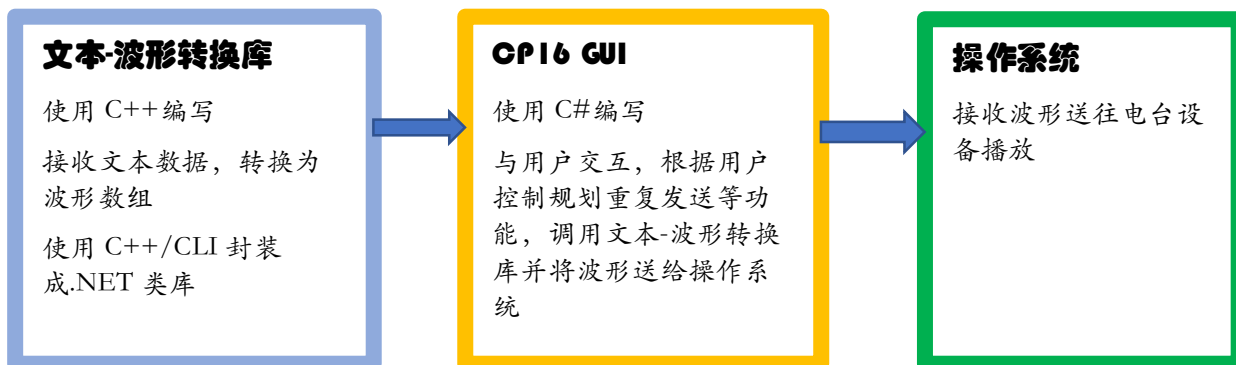
2.1.2.4 声音生成

汉字点阵为 16*16，在 CP16 协议中，每行的 16 个点对应 16 个频点，16 行代表 16 个帧。根据针对汉字点阵同一行的每个点，设置一个频点，最低频率 1000Hz，最高频率为 1285Hz，间隔 19Hz。如果一个点阵点为 1，生成一段对应频率的正弦波，长度为对应的帧长，即 125ms。将这些不同频率的正弦波进行叠加就是需要输出的信号。进行短时傅里叶变换后就可以看到文字信息。

由于发送的帧长不是很长，矩形窗会存在较严重频谱泄漏现象。所以我们通过添加汉明窗进行平滑，减少频谱泄露。一个帧需要 125ms，16 帧需要 2s，这是发送一个汉字所需要的时间。



2.2 UI



虽然我们已经有了根据待发送字符串生成波形数值的 C++ 代码，为了使用户能够方便的利用 CP16 编码进行通讯，还必须设计一个简洁易懂的 UI 界面，方便用户与软硬件进行交互。

2.2.1 语言和框架选取

在 GUI 编程语言和框架方面，我们选择了使用 C# 语言和 WPF 用户界面框架。选择 C# 的原因是作为一种拥有自动垃圾回收机制和大量内置库函数（例如线程操作相关函数）的语言，C# 使用起来非常方便，同时作为 .NET 语言家族的一员，C# 可以很简单的调用其他 .NET 语言生成的类库；我们只需要将使用 C++ 编写的文本-波形转换库包装为同为 .NET 语言的 C++/CLI 类，即可生成便于调用的 dll 库。选择 WPF 框架则是因为使用该框架制作 GUI 程序非常方便，同时生成的程序不会像传统的 MFC 或 WinForm 架构下生成的程序那样，在高分辨率屏幕下显示模糊。

2.2.2 程序结构设计

程序主要分为两个线程：一个是程序主线程，用于控制 UI 元素的显示；一个是后台工作线程，用于从 UI 获取用户输入的文本，使用文本-波形转换库转换为波形数据，然后使用系统音频播放 API 进行播放，同时通知主线程更新 UI 的相关元素。

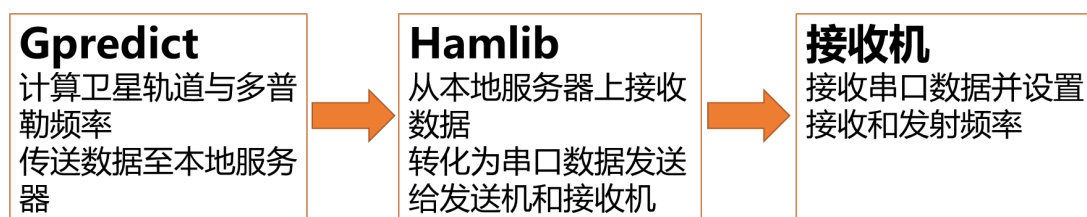
在进行音频播放相关代码编写时我们遇到了一个问题，C# 内置库函数中，从内存数据流播放音频的函数只能播放 wav 文件格式的数据，而不能播放无格式的 PCM 数据。因此我们必须现在内存中手动构建 wav 文件头，并将浮点 PCM 数据转换为可用的 16 位整型数据。

2.2.3 UI 界面设计和效果

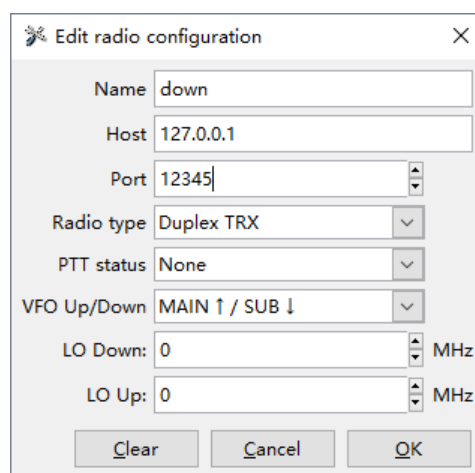


如图所示，最终实现的 UI 界面类似于 QQ 等即时聊天工具，允许用户输入任意待发送文本并选择单次发送或重复发送功能。另外由于 CP16 文本发送较慢，还添加了使用不同颜色的提示文字来提示用户文本是否已经发送完毕。同时在老师的建议下，还添加了无线电呼号的设置，能够自动在每条发送的文本前添加相应的呼号文本。界面右侧的空白部分本来是预留给 UI 界面功能扩展（例如增加发送内容瀑布图显示等），不过由于时间原因最终没有用到。

2.3 使用 gpredict 控制收发信机



gpredict 是免费的能够进行卫星追踪和轨道预测的软件。它能够根据卫星轨道和接受位置进行多普勒频率的计算, 并进行数据的传送。并频率传送至对应服务器的端口。如下图所示, 在 Edit-Preference-Interface 中选择 Add New 并新建一个传送通道, 配置参数为接听的发送的端口号为 12345, 控制选项为 Duplex TRX (同时接收发送)。之后 gpredict 就会向对应的端口发送发射接收器消除多普勒的频率。



Hamlib 是一个可以对发射接收器进行控制的软件库, 除了可以手动输入命令, 还可以进行自动控制。通过在本地建立服务器并监听对应端口, 它可以将 gpredict 的频率转换为串口数据发送给发射接收器控制频率。设置串口号为 COM4、波特率为 19200, 设置监听的服务器端口号为 12345、模型号为 368 (对应接收机型号 IC-9100)。启动软件命令如下:

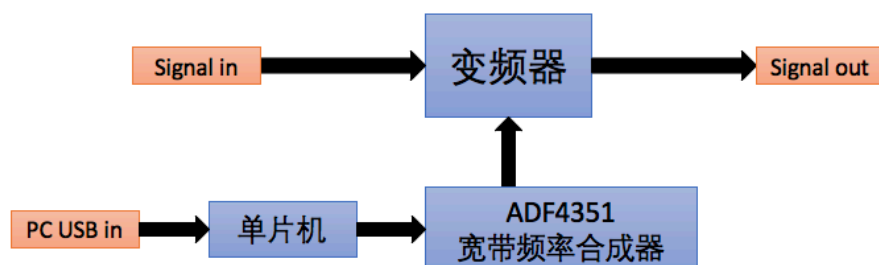
```
C:\Program Files (x86)\hamlib-w32-3.1\bin>rigctld.exe -r COM4 -s 19200 -t 12345 -m 368
```

之后通过 gpredict 的频率控制窗口就能进行频率的自动控制。遗憾的是由于 IC-9100 进行控制时只能控制接收频率而无法改变发射频率, 我们没有办法单纯通过软件完全消除多普勒, 最终接受仍是利用手动控制进行的。

详细的设置教程请参考附录。

2.4 自制模拟卫星

通过这个系统，我们意图模拟一个卫星对信号的操作，有两个输入端口：signal in 和 USB in，有一个输出端口 Signal out。



其中 signal in 是输入的频宽 20kHz，中心频率约为 435MHz 的信号（模拟卫星上行），pc usb in 是通过 usb 端口输入的单片机控制信号。这个控制信号控制单片机生成信号接入 ADF4351 宽带频率合成器，生成点频 535MHz 的信号，再通过变频器，将输入的中心频率为 435MHz 的信号镜像搬移到 100MHz 附近，这样系统整体就模拟完成了希望卫星系统的功能。

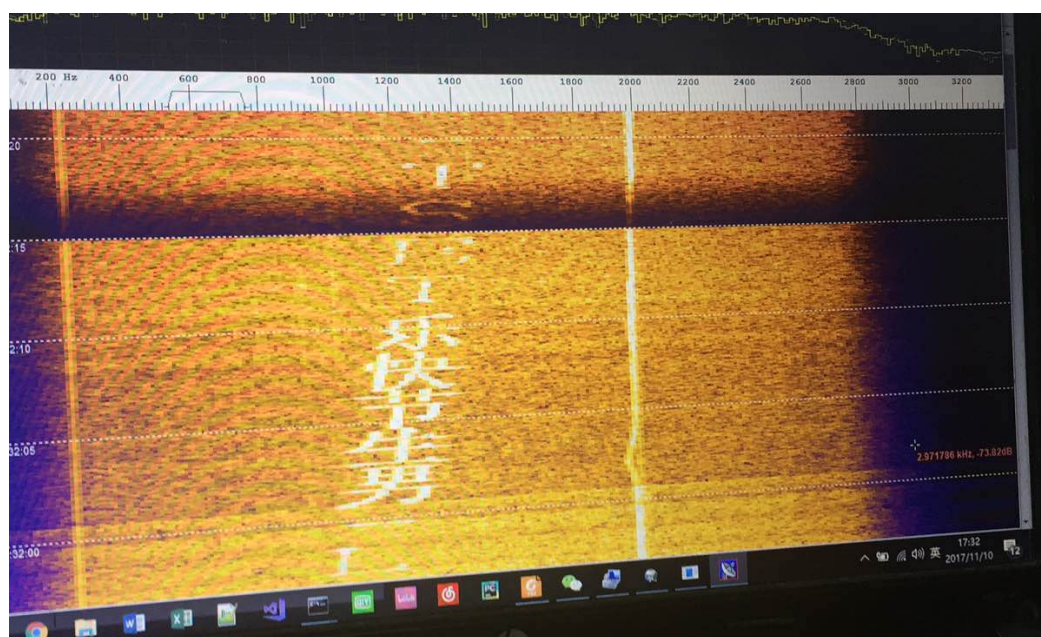
其中我们通过 usb 串口控制单片机，这个单片机比较简单直接通过 controller 自动控制八个管脚的输出，从而向变频器输出正确的点频。我们尝试使用通用单片机配合 Arduino 实现相同的功能，但是没有成功。



三、系统结果展示

如下图所示，我们成功发射，并通过卫星转发，接收了“男生节快乐”等文字。可以在以下链接观看我们成功发射的视频：

http://v.youku.com/v_show/id_XMzI4Mjc3OTQwOA==.html



四、总结与体会

这次试验中，我们从零开始完整地做了一个颇具规模的项目，在做实验的过程中，我们不仅新学习到了很多东西，同时也深刻的体会到了什么叫做学以致用感觉，如果没有之前那些通信课程的积累，我们是不可能在这这么短的时间内就完成这样的工作的。

从 CP16 的 C++ 代码实现、到 GUI 的编写、到自动实现控制收发频率、再到各种追卫星的调试，整个实验做下来，我们一开始觉得很简单很直观的事情其实充满了各种阻碍，比如编码时全角半角的问题、比如卫星每天只经过北京上空一小会、比如做模拟卫星时分析不出哪里有 bug 甚至还用上了逻辑分析仪、比如用软件控制收发频率时文档不明确痛苦…这些困难都是我们一开始没有想象会遇到的，在——解决他们的过程当中，真的是很好的锻炼了我们的能力。

作为大学四年的最后一门课，可以说是感慨万千，这门课里我们做的项目不仅仅只是一个项目，更像是对我们大学四年所学的一个总结。

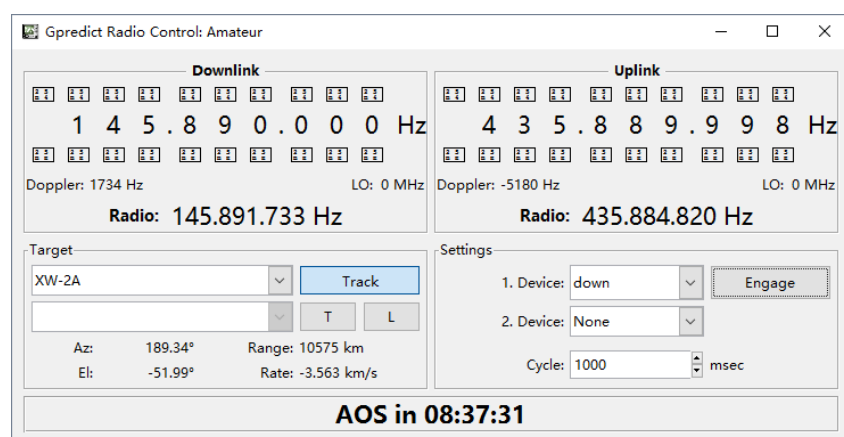
五、附录

1 设置收发信机频率使用说明

首先打开命令行以管理员模式启动，进行 Hamlib/bin 所在目录，输入命令：

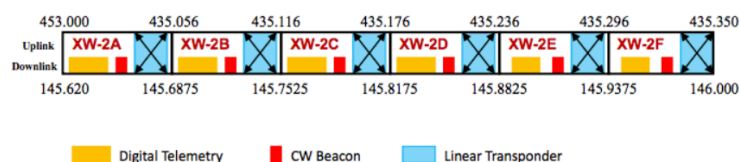
```
rigctld.exe -r COMx -s 19200 -t 12345 -m 368
```

其中 COMx 代表对应发射接收机的串口号。然后在 gpredict 中的 Radio Control 打开频率控制窗口，并输入对应的中心频率选择需要追踪的卫星，选择 Track 并 Engage。



2 自制模拟卫星技术手册、使用说明

2.1.1 输入信号：上行频率约为 435MHz，频宽 20kHz。



3. XW-2 series of satellites:

Uplink:			
Satellite	Frequency (MHz)	Bandwidth (kHz)	Application
XW-2A (CAS-3A)	435.030 - 435.050	20	Linear Transponder (Spectrum Inverting)
XW-2B (CAS-3B)	435.090 - 435.110		
XW-2C (CAS-3C)	435.150 - 435.170		
XW-2D (CAS-3D)	435.210 - 435.230		
XW-2E (CAS-3E)	435.270 - 435.290		
XW-2F (CAS-3F)	435.330 - 435.350		

2.1.2 ADF 宽带频率合成器与：生成频率 35MHz 至 4400MHz 的信号。

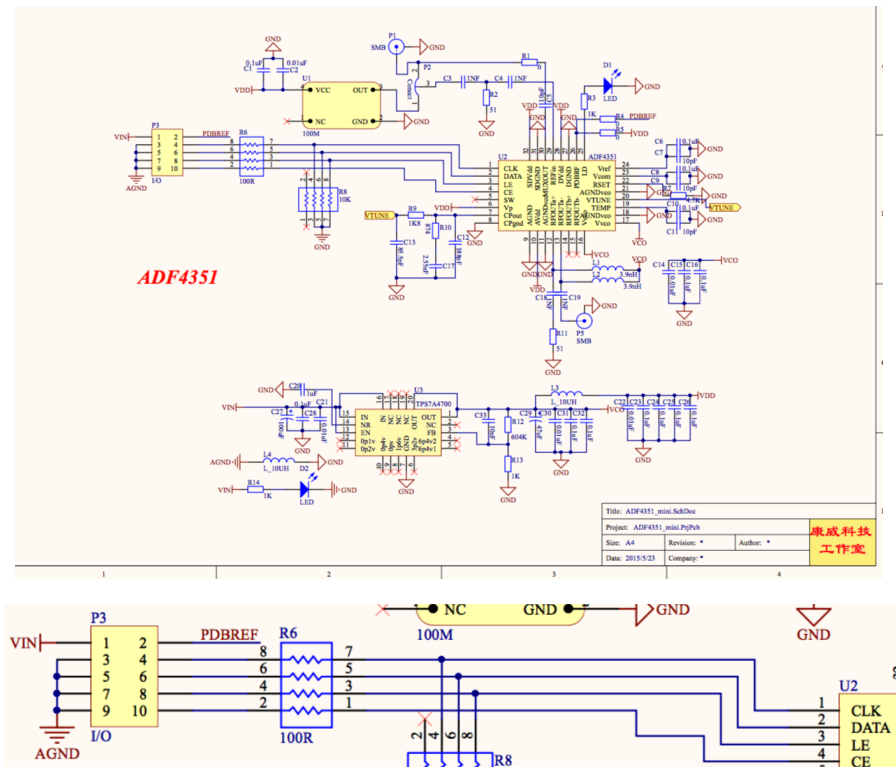


表5. 引脚功能描述		
引脚编号	引脚名称	描述
1	CLK	串行时钟输入。数据在CLK上升沿时逐个输入32位移位寄存器。此输入为高阻抗CMOS输入。
2	DATA	串行数据输入。串行数据以MSB优先方式加载，三个LSB用作控制位。此输入为高阻抗CMOS输入。
3	LE	加载使能。当LE变为高电平时，存储在32位移位寄存器中的数据载入三个控制位所选择的寄存器。此输入为高阻抗CMOS输入。
4	CE	芯片使能。此引脚的逻辑低电平将关断器件，并使电荷泵进入三态模式。根据关断位的状态不同，此引脚的逻辑高电平将使器件上电。
26	PDB _{RF}	RF关断。此引脚为逻辑低电平时，RF输出静音。此功能也是软件可编程的。

2.1.3 pc usb in: 控制单片机产生八个管脚的输入信号

