1、ADC的主要特性：采样率和动态范围。

采样率是ADC测量模拟信号的速度，USRP上有4个采样率为64MHZ的ADC。动态范围是ADC区别最低信号值和最大信号值的精度，决定于ADC的位数，如8位的ADC最多能代表256个信号梯度。ADC的物理特性（主要从转换原理区分）和价格决定了这两项指标。

2、我们为什么需要一个RF前端？

根据奈奎斯特采样定律，如果想要无混叠的将信号从采样数据中恢复出来，那么采样率至少应是信号最大频率的2倍（低通采样定理）。如果ADC的采样频率是20MHz，则正常情况下无法接收90M—100M的FM广播。但是通过RF前端，我们可以实现这个功能。RF前端的功能就是降频，将高频降到中频。我们可以简单的把RF前端想像成一个黑盒，它的输入是高频信号，而输出则是以中频（IF）为中心频率的低频信号。

3、GNU radio的两个开发工具：C++和Python。其中复杂的信号处理模块block是由C++编写的，而Python将各个模块粘接起来，形成流图flow graph。dial\_tone.py是GNU radio中的Hello World，在没有USRP硬件设备的情况下，只通过声卡就可以发出拔号音。代码是通过产生两个频率分别为350Hz和440Hz的正弦信号，叠加到audio.sink上进行播放。audio.sink是一个接收器，它将接收到的数据输入到声卡中进行播放。

4、URSP：GNU radio最好的伙伴

虽然gnu radio支持很多的硬件设备来进行开发，但无疑usrp系统是它最好的伙伴。从USRP的启动过程中，可以看出它所包含的各部分：USB控制器（8051固件）、FPGA的配置（bitsteam,通过verilog编写）。硬件层次的开发处于开发的最高层，高于c++、Python层次代码的编写。

5、FPGA的作用就是做上下变频，在数字中频和基带信号之间进行转换。在接收模式下，标准的FPGA配置可以让我们选择所感兴趣的频段，并完成基带化和抽取、滤波等功能。和RF功能类似，只是这是在数字域的采样变换。这里涉及数字下变频转换器（Digital Down Converter,DDC）的原理。

5.python对波形文件做方差

Import wave

利用wave.open()函数打开波形函数，可参考网址http://sebug.net/paper/books/scipydoc/wave\_pyaudio.html

f1=wave.open(r'/home/yun/gqrx-20150320-085746.wav','rb')

#read the wave's format infomation,and return a tuple

params = f1.getparams()

#get the info

nchannels, sampwidth, framerate, nframes = params[:4]

#Reads and returns nframes of audio, as a string of bytes.

expectedfile = f1.readframes(nframes)

#close the stream

expected\_data = np.fromstring(expectedfile, dtype = np.short)

f1.close()

下面是方差函数

import numpy

def variance(nlist)

narray=numpy.array(nlist)

sum1=narray.sum()

narray2=narray\*narray

sum2=narray2.sum()

mean=sum1/N

var=sum2/N-mean\*\*2

return var

6.将线程设置为后台进程，设置setDaemom为true。

def my\_generator():

###程序

t=threading.Thread(target=my\_generator)

t.setDaemon(True)

t.start()

7.结束进程

①os.\_exit(0)，整个程序退出，不建议，要等子进程结束以后再退出thread.join()

②sys.exit(0),主要用于子进程的退出

③os.kill(os.getpid(),signal.SIGKILL),杀死进程

④用于multiprocess ,from multiprocess import Process

p = Process(target=f, args=('P',))  
         p.start()  
         p.terminate(),用来结束进程

⑤设置标志

Thread.start()

Thread.running=false

继承threading.Thread

粗鲁的将一个线程关闭是不可取的，因为该线程可能正在访问一些需要主动关闭的资源或者该线程还产生了另外一些需要关闭的线程。因此Python threading并没有提供线程关闭的API。

8.python文件和路径操作

Path = /home/yun/gqrx.wav

os.path.dirname(path)=/home/yun

**os**.path.basename(path)=gqrx.wav

os.path.exists(path)路径是否存在

具体操作可参考：http://www.cnblogs.com/bluescorpio/archive/2009/11/21/1607446.html

9.在一段时间后停止程序

<http://zhidao.baidu.com/link?url=xujO-wwH74wfnCcslxX4UqnTCtvOUfQtTOkuW78CvvYlcoHhgWb4m8RZbp1iZ4IKBJsOhzLYfStQCrgBQ6Ap9a>

10 python的测试框架pytest

<http://my.oschina.net/lionets/blog/269892>

<http://pytest.org/latest/example/index.html>