

RSA306B 使用手册

RSA306B Handbook

目录

1. 设备简介.....	2
2. 设备的基本操作.....	2
2.1 与电脑进行连接	2
2.2 SignalVu-PC 的操作.....	2
3. API 接口.....	3
3.1 介绍	3
3.2 RSA_API 安装方法.....	4
4. RSA306B 应用实例：通过频谱图识别射频信号种类.....	4
4.1 修改实例代码，使其自动获取 DPXogram	4
4.2 安装 TensorFlow 频谱图识别所需环境	5
4.3 执行训练步骤.....	5
4.4 测试结果	6
4.5 改进	6
4.6 重新训练	7

1. 设备简介

RSA306B 配合泰克 SignalVu-PC™ RF 信号分析软件，为 9 kHz ~6.2 GHz 信号提供实时频谱分析、流式捕获和深入信号分析功能。主要性能指标包括：

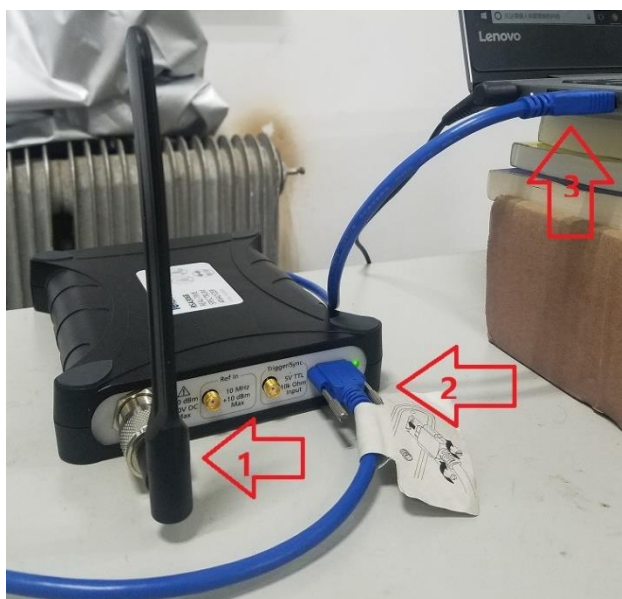
- 9 kHz ~ 6.2 GHz 频率范围，满足各种分析需求
- +20 dBm ~ -160 dBm 测量范围
- 40 MHz 采集带宽可以对现代标准执行宽带矢量分析
- 以 100% 侦听概率捕获持续时间最短 100 μs 的信号

主要特点包括：

- 使用标配泰克 SignalVu-PC™ 软件，获得全功能频谱分析功能
- 地图绘制、调制分析、WLAN、LTE 和蓝牙标准选项支持
- 实时频谱/三维频谱图显示，使查找瞬态信号和干扰的时间达到最小
- 标配应用编程接口 (API)，适用于 Microsoft Windows 环境

2. 设备的基本操作

2.1 与电脑进行连接



- 1、旋紧天线
- 2、数据线一端连接至设备
- 3、另一端连接电脑 USB3.0 接口
- 4、绿灯为连接正确；红灯为连接不正确；闪烁为正在工作

2.2 SignalVu-PC 的操作

导航栏 Setup-Displays 可选择要观测的频谱图类型，因为我们使用的是免费版，所以可观测的类型有限，简单介绍一个常用的类型：

- 1、Spectrum：最基本的频谱图，横坐标为频率，纵坐标为信号的幅度。
- 2、DPX Spectrum：Tek 公司的专利技术，可以看见几乎无法通过传统频谱分析仪看见的频谱特征。
- 3、Spectrogram：瀑布图，横坐标为频率，纵坐标为时间，坐标系中的颜色代表

信号的强度。

导航栏 View 可以勾选常用的工具栏：

- 1、Chan Nav Toolbar：频道导航工具栏，帮助你快速选择特性信号的频率，例如 Wi-Fi、蓝牙和 4G 等。
- 2、Basic Toolbar：可以调整频谱的频率、参考电平、跨度和分辨率带宽。
- 3、Status Bar：显示工作状态。
- 4、Markers Toolbar：调整标记，方便选取信号。

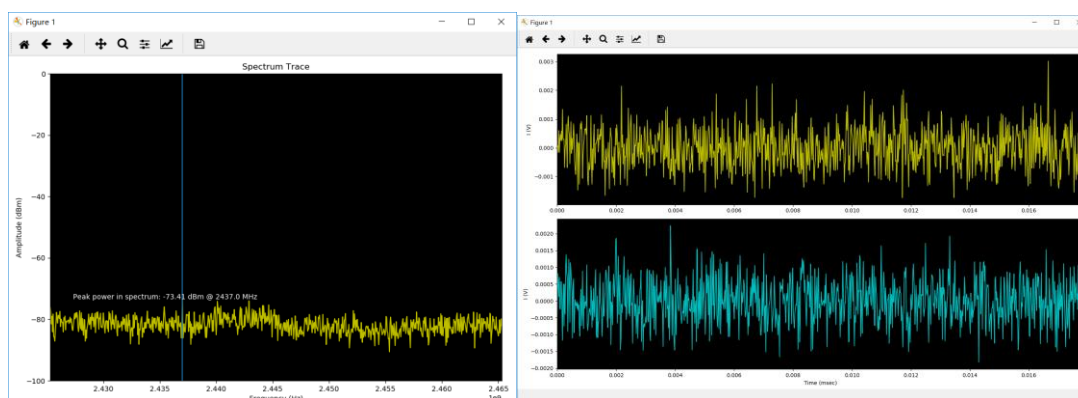
一个简单的实例：收听 FM 广播（[参考视频](#)）

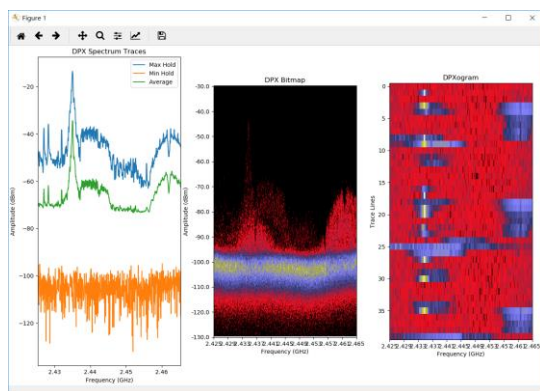
- 1、导航栏 Presets-Main。
- 2、频率调整至 100 MHz，参考电平调整至 -30 dBm，使信号清晰可见。
- 3、频谱图内右键选择 Marker to peak，然后选择 Markers Toolbar 中的 To Center。
- 4、导航栏 Setup-Audio，点击 Run 即可收听，Stop 即停止。
- 5、换台利用 Markers Toolbar 的左右按钮，然后再次 To Center 即可。

3. API 接口

3.1 介绍

Tek 公司为 RSA306B 提供了丰富的 API 接口，使得我们可以通过编程轻松获取实时数据，例如 Audio、DPX、Power、Spectrum 等（详见 RSA-API 编程参考手册）。官方提供的 API 参考实例（rsa_api_full_example.py）中包括 Spectrum、Block IQ 和 DPX 三个例子：





在后续的分析中我们要用到的是 DPX 中的 DPXogram（第三个图中的第三个），因为它的特征明显，并且不同信号之间差别很大，易于区分。

3.2 RSA_API 安装方法（以 Python 为例，API 还支持其他编程语言例如 C#、C++、Matlab、VB.Net，详见[代码实例](#)）

- 1、[下载](#)并安装 Anaconda，Anaconda 是一个开源的 Python 发行版本，其包含了 conda、Python 等 180 多个科学包及其依赖项。
- 2、[下载](#)并安装最新版本的 RSA_API。
- 3、使用 API 需要有一些 Python 的编程基础，例如知道如何用命令行来执行 Python 程序。
- 4、安装成功后，[下载](#)上述参考实例程序，需要用到的是 RSA_API.py 和 rsa_api_full_example.py。**注意！在执行程序前需要正确连接设备(绿灯亮)，且不要同时打开 SignalVu，因为这样会导致冲突。**打开命令行，将命令行路径选择到.py 文件所在位置（两个.py 文件需放在一起），执行“python rsa_api_full_example.py”，此时能分别显示上述三个图片即表示安装成功。

4. RSA306B 应用实例：通过频谱图识别射频信号种类(Wi-Fi、FM、手机信号)

4.1 修改实例代码，使其自动获取 DPXogram([修改后的代码](#))

为什么要使用 DPXogram？通过 SignalVu，我们可以看到非常清晰的 Spectrogram，运用它来分析是最好的。但是 API 参考手册中并未发现 Spectrogram 的调用方法，所以才用实例中的 DPXogram 来代替 Spectrogram，二者只是表现形式上存在区别，内容是一样的，最终也能实现分辨。

- 1、将无用的程序块删除，只保留 config_DPX、acquire_dpx_frame、extract_dpxogram、dpx_example。
- 2、去除保存图片中的空白（[可参考](#)）。
- 3、加入循环，自动获取。

- 4、main 函数中只保留 dpx_example()。
- 5、程序中可以修改的地方：dpx_example 中的 cf/reflevel/span/rbw; for 循环的范围；fig.savefig 中保存图片的路径。
- 6、自动获取时，建议每类频谱图至少 500 张，以保证准确性。

4.2 安装 TensorFlow 频谱图识别所需环境

什么是 TensorFlow? TensorFlow 是谷歌研发并开源的第二代人工智能学习系统, TensorFlow 是将复杂的数据结构传输至人工智能神经网络中进行分析 and 处理过程的系统。TensorFlow 可被用于语音识别或图像识别等多项机器学习和深度学习领域。

- 1、[安装](#)只支持 CPU 的 TensorFlow([以 Windows 为例](#)), 这里不涉及大的计算量, 所以用只支持 CPU 版本的足够。这里有两种方法: "native" pip 和 Anaconda。这里只介绍 "native" pip 方法, 其他方法及支持 GPU 的版本请[参考](#)。由于我们之前安装过 Anaconda, 已经包含了 Python, 所以我们直接打开命令行, 并输入:

```
C:\> pip3 install --upgrade tensorflow
```

- 2、验证是否安装成功。命令行中输入:

```
$ python

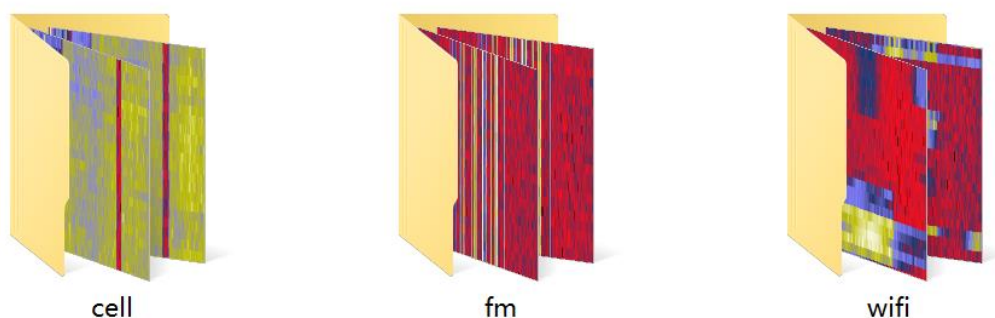
>>> import tensorflow as tf
>>> hello = tf.constant('Hello, TensorFlow!')
>>> sess = tf.Session()
>>> print(sess.run(hello))
```

如果系统输出: "Hello, TensorFlow!", 则代表安装成功, 若出现错误请参考[常见安装问题](#)。

- 3、安装训练频谱图时所需的 TensorFlow Hub, 安装时需要 TensorFlow 版本 $\geq 1.7.0$:
\$ pip install "tensorflow>=1.7.0"
\$ pip install tensorflow-hub

4.3 执行训练步骤 ([官网英文教程](#))

- 1、将自动获取的 DPXogram 放入以信号类型命名的文件夹内, 如图。



2、[下载](#)训练的代码

```
mkdir ~/example_code
cd ~/example_code
curl -LO https://github.com/tensorflow/hub/raw/r0.1/examples/
image_retraining/retrain.py
```

3、执行训练的代码（推荐使用绝对路径：E:/photos）

```
python retrain.py --image_dir ~/flower_photos
```

大概耗时半小时（取决于 CPU 的计算速度）。

4.4 测试结果

1、[下载](#)测试代码

```
curl -LO
https://github.com/tensorflow/tensorflow/raw/master/tensor
flow/examples/label_image/label_image.py
python label_image.py
```

2、执行测试代码

```
python label_image.py --graph=C:/tmp/output_graph.pb
--labels=C:/tmp/output_labels.txt
--input_layer=Placeholder
--output_layer=final_result --image="测试图片的路径"
```

4.5 改进

由于之前获取 DPXogram 时，只选择了特定频率的 Wi-Fi 和手机信号每样 500 张。现在我们可以选用不同频率的信号，例如 Wi-Fi 选取 802.11g-信道 1、802.11g-信道 6、802.11g-信道 11 和 802.11n-信道 44 的信号（可以选取设备周围强度大的信号，易于检测），还有三家运营商 2G、3G、4G 的不同信号。

4.6 重新训练

- 1、 将 C 盘内的 tmp 缓存文件以及 C:\Users\“用户名”\AppData\Local\Temp\tfhub_modules 内的缓存文件删除，以便重新训练时生成新的文件。
- 2、 将命令行路径选择到 .py 文件所在位置，执行“python rsa_api_full_example.py”。
- 3、 再次测试结果。