



*EagleSDR*

鹰击长空 软件无限

*HackRF One*

软件无线电平台

用户使用手册

品质 服务 信誉 专业

1 HackRF One 入门.....	3
1.1 简介.....	3
1.2 硬件分析.....	4
2 Windows 环境下 HackRF 开发指南.....	6
2.1 安装 VS2012 Redistributable.....	6
2.2 安装 USB 驱动.....	6
2.3 安装 SDRSharp.....	7
2.4 使用 VS2010 编译 libhackrf.....	9
2.4.1 编译 hackrflib.....	9
2.4.2 编译 HackRFLibTest.....	13
3 Linux 环境下 HackRF 开发指南.....	17
3.1 安装 Linux 操作系统.....	17
3.2 搭建开发环境.....	17
3.2.1 安装 GNURadio 和 gqrx.....	18
3.2.2 使用现成虚拟机文件.....	18
3.2.3 验证开发工具安装是否成功.....	18
4 GNURadio 使用指南.....	19
4.1 GNURadio 基本操作.....	19
4.2 常用功能模块介绍.....	21
4.3 GNURadio 工程示例.....	28
5 HackRF One 开发实例.....	29
5.1 HackRF One 发射 FM 音频.....	29
5.2 HackRF One 接收 FM 广播.....	33

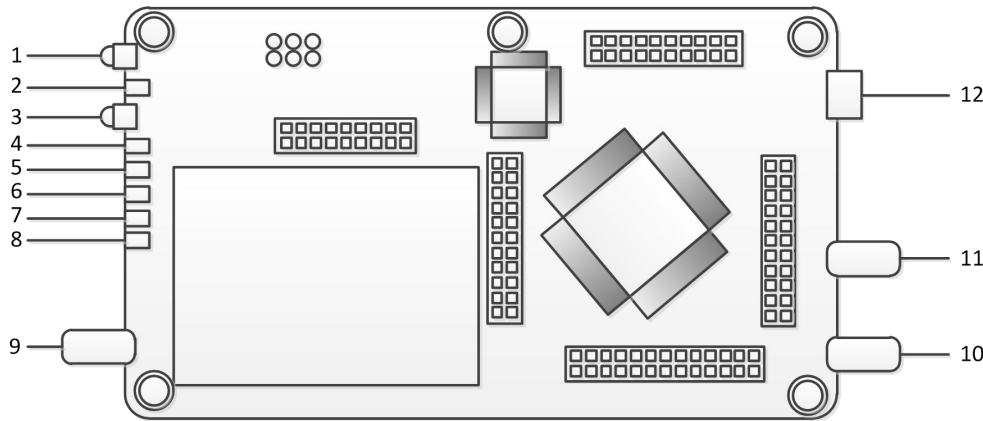
# 1 HackRF One 入门

## 1.1 简介

在开始使用 HackRF One 软件无线电平台之前，请先检查其各项配件是否齐全，配件列表如下所示：

部件名称	数量	备注
HackRf One 主板	1 块	
亚克力外壳	1 套	
USB 线	1 根	
天线	1 根	GSM 频段
螺丝	1 套	
用户使用手册	1 份	
资料光盘	1 张	

HackRF One 软件无线电平台包括了大量的数据接口和状态指示灯，其接口说明如下图所示：



- 1 复位按键----用于硬件复位
- 2 电源指示----3.3V 供电正常
- 3 固件更新按键----用于固件更新
- 4 电源指示----1.8V 供电正常
- 5 电源指示----RF 模块供电正常

6 USB 状态指示----USB 连接正常

7 RF 接收状态指示----表示 HackRF One 处于接收状态

8 RF 发送状态指示----表示 HackRF One 处于发送状态

9 射频信号输入、输出接口----连接天线

10 时钟输入接口----用于多个 HackRF One 同步或级联

11 时钟输出接口----用于多个 HackRF One 同步或级联

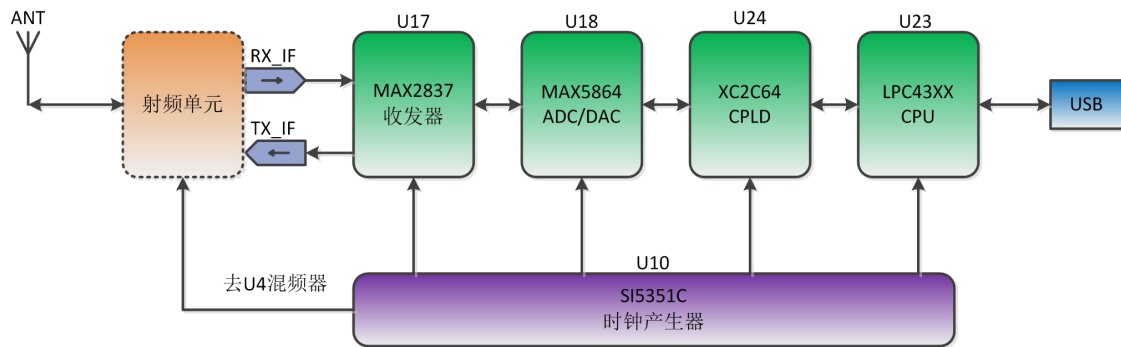
12 USB 接口----用于连接电脑

HackRF One 软件无线电平台的技术参数如下表所示：

技术参数	参数值
RF 范围	10MHz-6GHz
信号带宽	20MHz
采样精度（ADC/DAC）	8bit
采样速率（ADC/DAC）	20Msps
RF 发射功率	14dBm
带内平坦度	0.75dB/20MHz@3.005HGz-3.025GHz
动态范围内增益偏差	2dB
USB 接口	USB 2.0

## 1.2 硬件分析

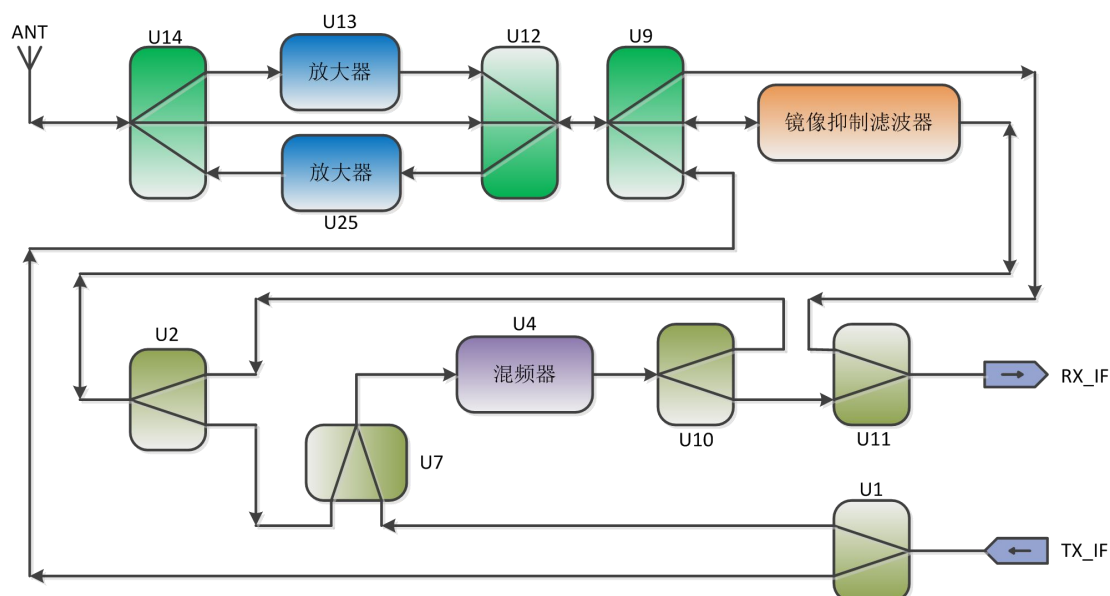
HackRF One 在单板上集成了射频链路所需的所有关键模块，包括 MCU、CPLD、ADC/DAC、射频收发器、混频器、时钟产生器和射频单元，完成射频信号到中频的频谱搬移、模数转换、调制解调和 USB 数据通信。其系统框图如下所示：



PC 机通过 USB 接口与 NXP LP43xx 微控制器通信，微控制器通过并行总线及 GPIO 模拟的 JTAG 接口与 CPLD 连接，MAX5864 对发送的数据进行数模转换，对接收到的数据进行模数转换，MAX2837 射频收发器实现上/下变频，并通过宽带混频器 RFFC5072 进行频率变换，射频单元采用一级 LNA/PA 进行放大，并最终通过射频开关与天线连接。

- ◆ LPC4320: NXP ARM Cortex M4 处理器，主频 204MHz
- ◆ XC2C64A: Xilinx, CoolRunner-II 系列 CPLD, 1500 门
- ◆ MAX2837: 2.3GHz to 2.7GHz 无线宽带射频收发器
- ◆ RFFC5072: 混频器，提供 80MHz 到 4200MHz 的本振
- ◆ MAX5864: ADC/DAC, 8-bit, 22MHz 采样率
- ◆ Si5351C: I<sup>2</sup>C 可编程任意 CMOS 时钟生成器
- ◆ MGA81563: 0.1 - 6GHz, 14 dBm 放大器
- ◆ SKY13317: 20 MHz-6.0 GHz 射频单刀三掷(SP3T)开关
- ◆ SKY13350: 0.01-6.0 GHz 射频单刀双掷(SPDT)开关

射频单元详细框图如下图所示，其中 U14、U12、U9 为射频单刀三掷(SP3T)开关 SKY13317，U13 和 U25 为射频放大器 MGA81563，U2、U7、U10、U11 和 U1 为射频单刀双掷(SPDT)开关 SKY13350，U4 为混频器 RFFC5072。镜像混频器由 U5 和 U6 两个射频单刀双掷(SPDT)开关 SKY13350、U3 低通滤波器和 U8 高通滤波器组成。



接收状态时，射频信号从天线 ANT 输入，经 U14 射频单刀双掷(SPDT)开关 SKY13350，根据是否旁路射频放大，射频信号经 U13 放大或直通到 U12。U12 输入到 U9，射频输入信号分为两路，一路直接到 U11，输出到 RX\_IF；另一路经镜像抑制滤波器到 U2，经 U7 到 U4 混频器，混频后输出到 U10，从 U11 输出到 RX\_IF。

发射状态时，中频信号进入 U1 分为两路，一路直接送到 U9；另一路经 U7 进入混频器 U4，混频后经 U10 至 U2，并送至镜像抑制滤波器，滤波后到 U9。射频发射信号经 U9 和 U12 输入到射频放大器 U25，放大后的射频信号送至 U14，最终通过天线发射出去。

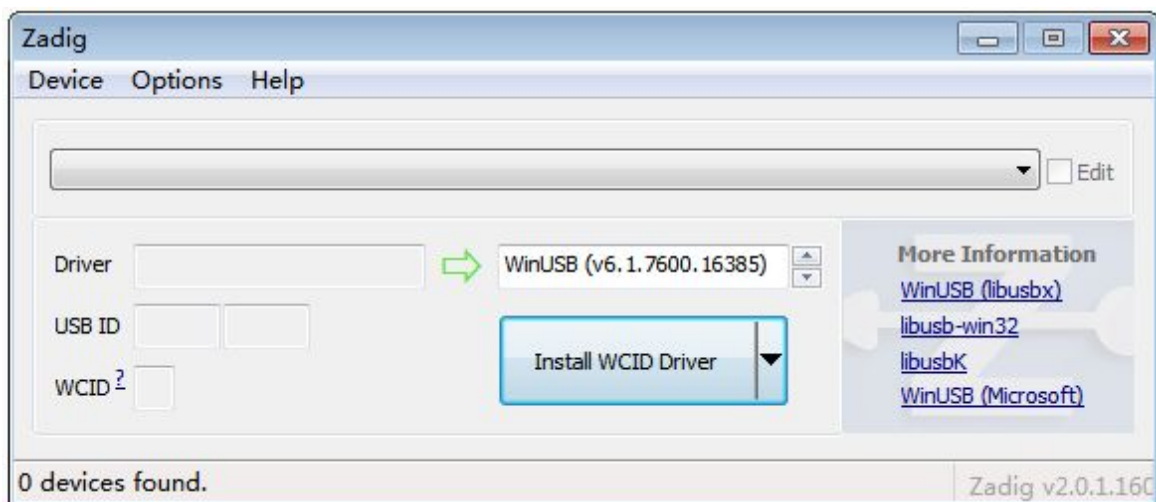
## 2 Windows 环境下 HackRF 开发指南

### 2.1 安装 VS2012 Redistributable

双击安装 VS2012\_Redistributable 目录下的 vcredist\_x86.exe。

### 2.2 安装 USB 驱动

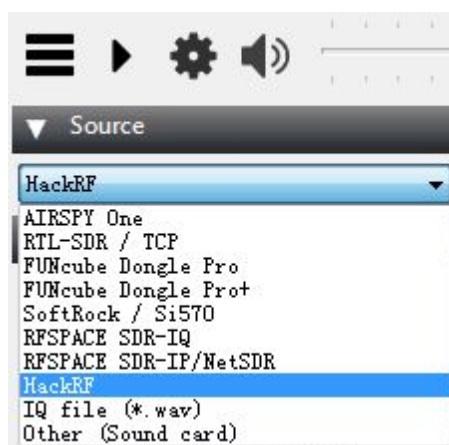
将 HackRF One 连接电脑，然后运行“资料光盘\Windows 软件\Zadig\zadig.exe”，如下图所示，在下拉菜单里选择“HackRF One”设备，点击“Install WCID Driver”即可。



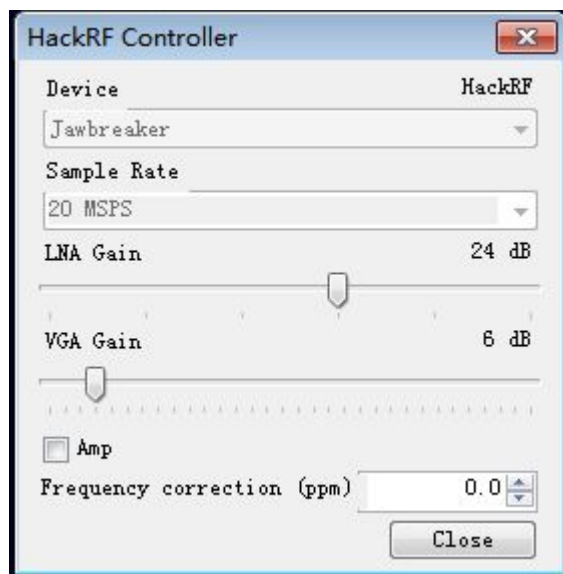
## 2.3 安装 SDRSharp

SDRSharp 为绿色软件，无需安装，解压后直接运行即可。该软件在“资料光盘\Windows 软件\sdrsharp”目录下，将其解压完成后，即可用 SDRSharp 收听 FM 广播。

- 1) 在 Source 栏中选择 HackRF 设备，如下图所示：

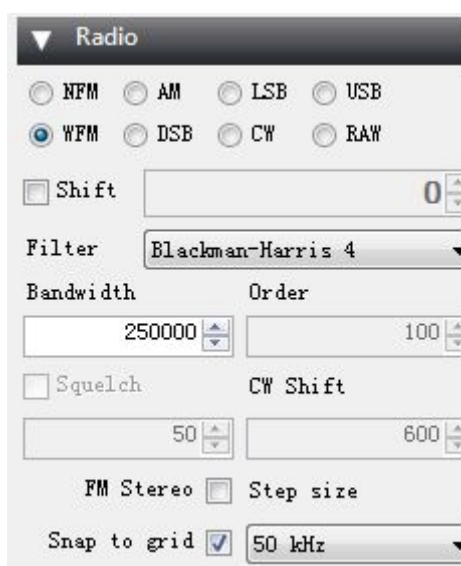


- 2) 点击“Configure”进行参数配置，拖动滑块设置相应的 LNA Gain（低噪放大器增益）和 VGA Gain（可变增益放大器增益），如下图所示：



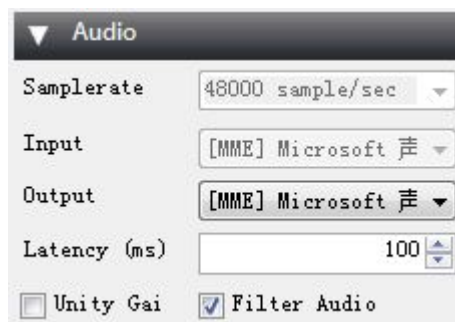
注意，要先点击“Start”按钮，否则，配置窗口的滑块为灰色不可拖动状态。

3) 展开“Radio”选项卡，选中“WFM”，表示解调类型为宽带调频，滤波器类型设置为“Blackman-Harris 4”，滤波器带宽设置为“250000”，步进频率设置为“50KHz”，并勾选 Snap to grid。详细配置如下：

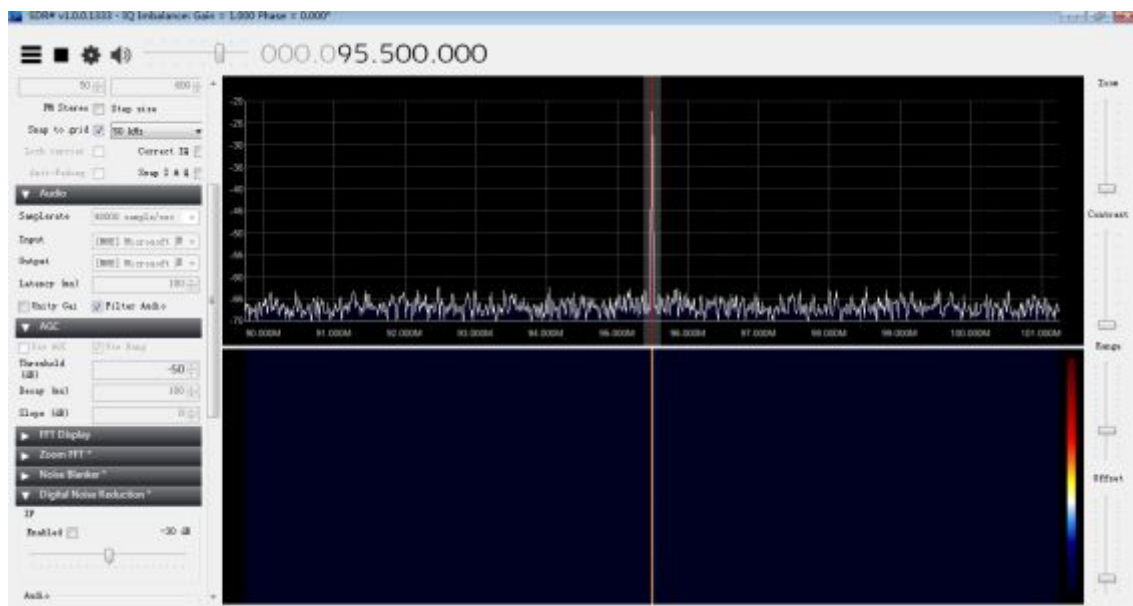


4) 展开“Audio”选项卡，对音频参数进行设置。Output 为声音输出设备，一般选择电脑的声卡，详细配置如下：





5) 参数设置完毕后，点击“Run”即可通过电脑的扬声器收听到调频广播，如下所示：



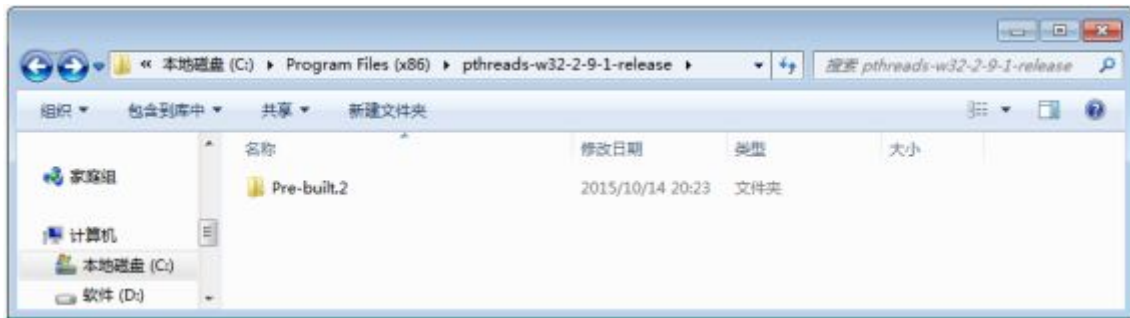
SDRSharp 不仅可以收听调频广播，通过设置相应的解调类型和参数，还可以接收 AM 广播、航空波段广播、LSB、USB 等。

## 2.4 使用 VS2010 编译 libhackrf

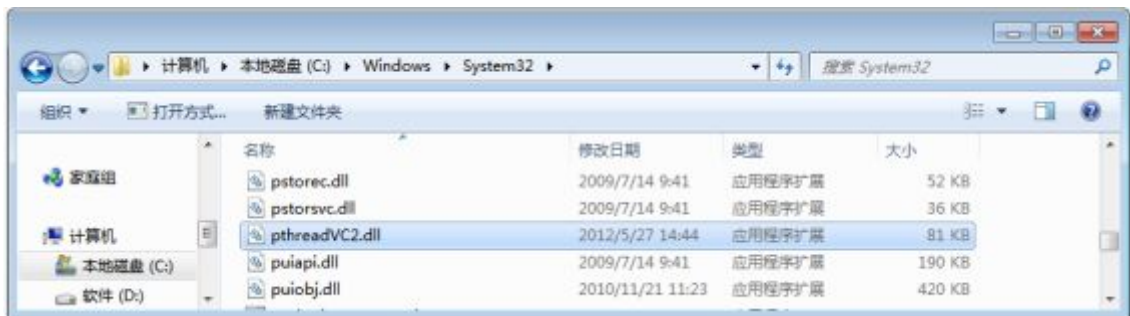
### 2.4.1 编译 hackrflib

1) 安装 pthread-win32。

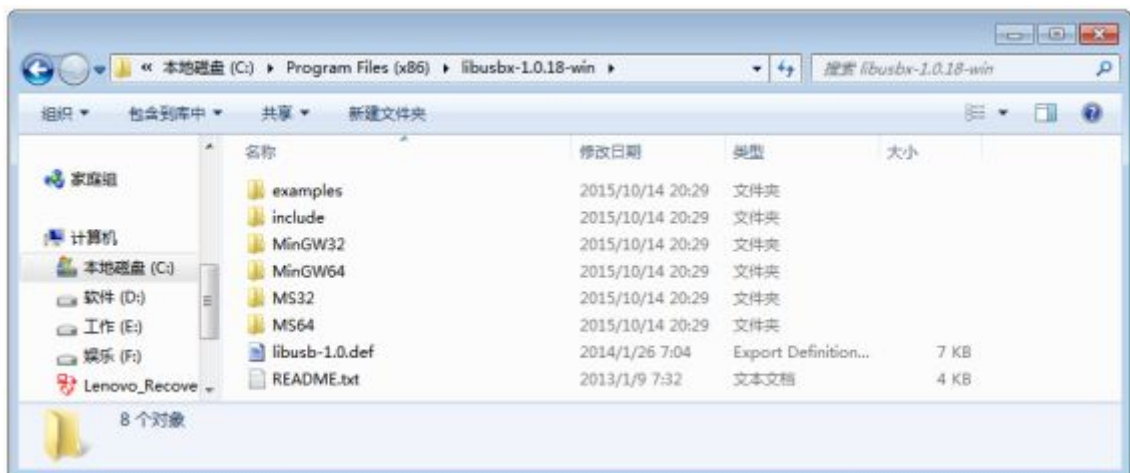
(1) 打开“资料光盘/Windows 软件/pthreads-w32-2-9-1-release.zip”压缩包，里面有三个文件夹，将 Pre-built.2 解压到 C:\Program Files (x86)\pthreads-w32-2-9-1-release (路径可以自己定义，这里只是建议路径)，如下图所示：



(2) 将 C:\Program Files (x86)\pthread-w32-2-9-1-release\Pre-built.2 \dll\x64\pthreadVC2.dll 拷贝到 C:\Windows\System32，如下图所示：



2) 将资料光盘/Windows 软件/libusb-1.0.18-win。解压到如下位置：



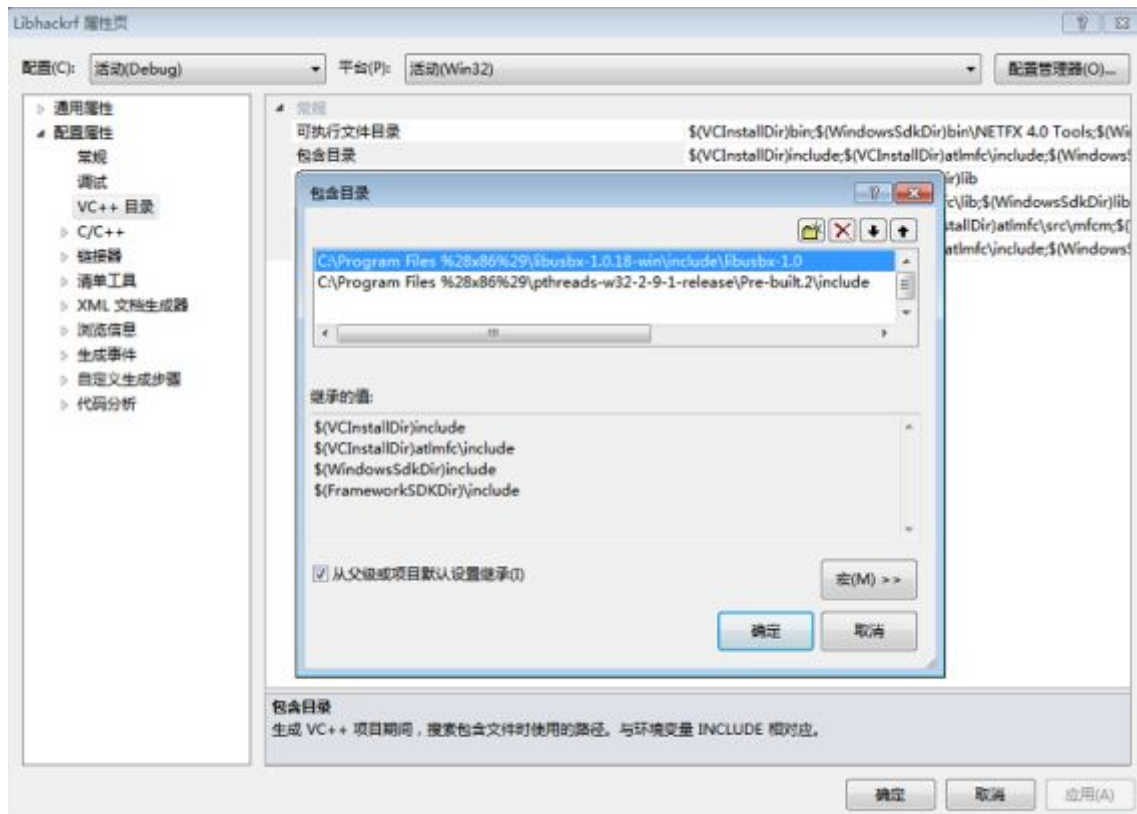
3) 编译 libhackrf

(1) 打开 VS2010 创建 Libhackrf 工程 (Win32 项目)，配置如下

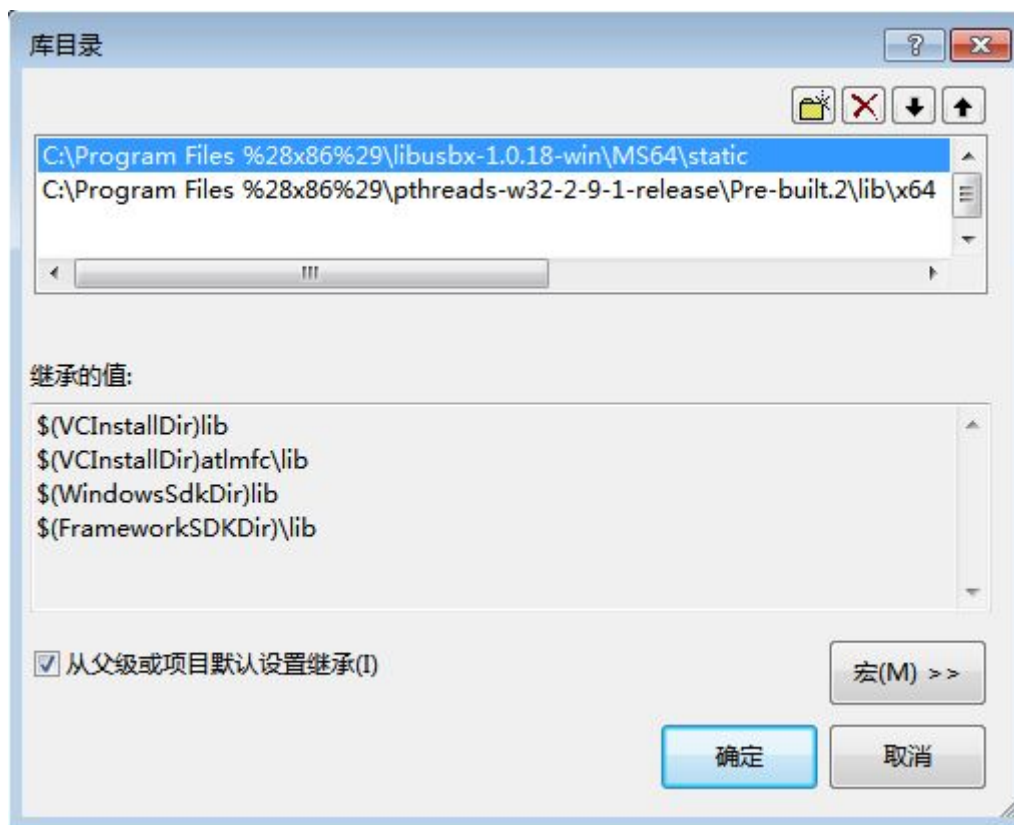


(2) 添加 `hackrf.c` 和 `hackrf.h`，这两个文件在“资料光盘/Windows 软件”目录下

(3) 配置头文件和依赖库的路径以及输入输出，选中项目点击右键->属性，弹出属性窗口，展开“配置属性”，选中“VC++目录”，点击右侧的“包含目录”的下拉箭头，点击“编辑”，弹出“包含目录”子窗口，增加两个目录，设置如下所示：

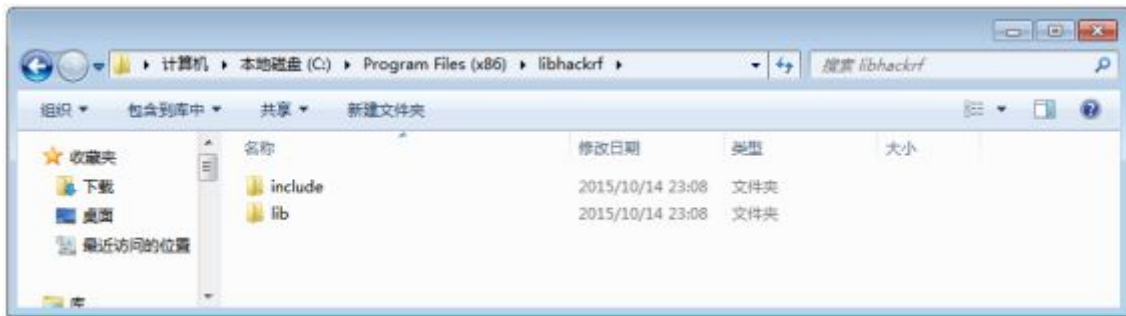


(4) 选中“VC++目录”，点击右侧的“库目录”的下拉箭头，点击“编辑”，弹出“包含目录”子窗口，增加两个目录，设置如下所示：



(5) 选中“常规”，点击右侧的“配置类型”设置成“静态库(.lib)”。

(4) 编译生成 Libhackrf.lib。为了方便调用，我将 hackrf.h 和 libhackrf.lib 拷贝到如下地址，hackrf.h 放在 include 文件夹，libhackrf.lib 放在 lib 文件夹。



## 2.4.2 编译 HackRFLibTest

(1) 创建 win32 console 工程，并输入如下代码：

```
#include "hackrf.h"
#include <stdio.h>
#include <stdlib.h>

int _tmain(int argc, _TCHAR* argv[])
{
    hackrf_device* device = NULL;
    int result = HACKRF_SUCCESS;
    uint8_t board_id = BOARD_ID_INVALID;
    char version[255 + 1];
    read_partid_serialno_t read_partid_serialno;

    result = hackrf_init();
    if (result != HACKRF_SUCCESS) {
        fprintf(stderr, "hackrf_init() failed: %s (%d)\n",
            hackrf_error_name((hackrf_error)result), result);
        getchar();
        return EXIT_FAILURE;
    }

    result = hackrf_open(&device);
    if (result != HACKRF_SUCCESS) {
        fprintf(stderr, "hackrf_open() failed: %s (%d)\n",
            hackrf_error_name((hackrf_error)result), result);
        getchar();
        return EXIT_FAILURE;
    }
}
```

```
}

printf("Found HackRF board.\n");

result = hackrf_board_id_read(device, &board_id);
if (result != HACKRF_SUCCESS) {
    fprintf(stderr, "hackrf_board_id_read() failed: %s (%d)\n",
            hackrf_error_name((hackrf_error)result), result);
    getchar();
    return EXIT_FAILURE;
}
printf("Board ID Number: %d (%s)\n", board_id,
       hackrf_board_id_name((hackrf_board_id)board_id));

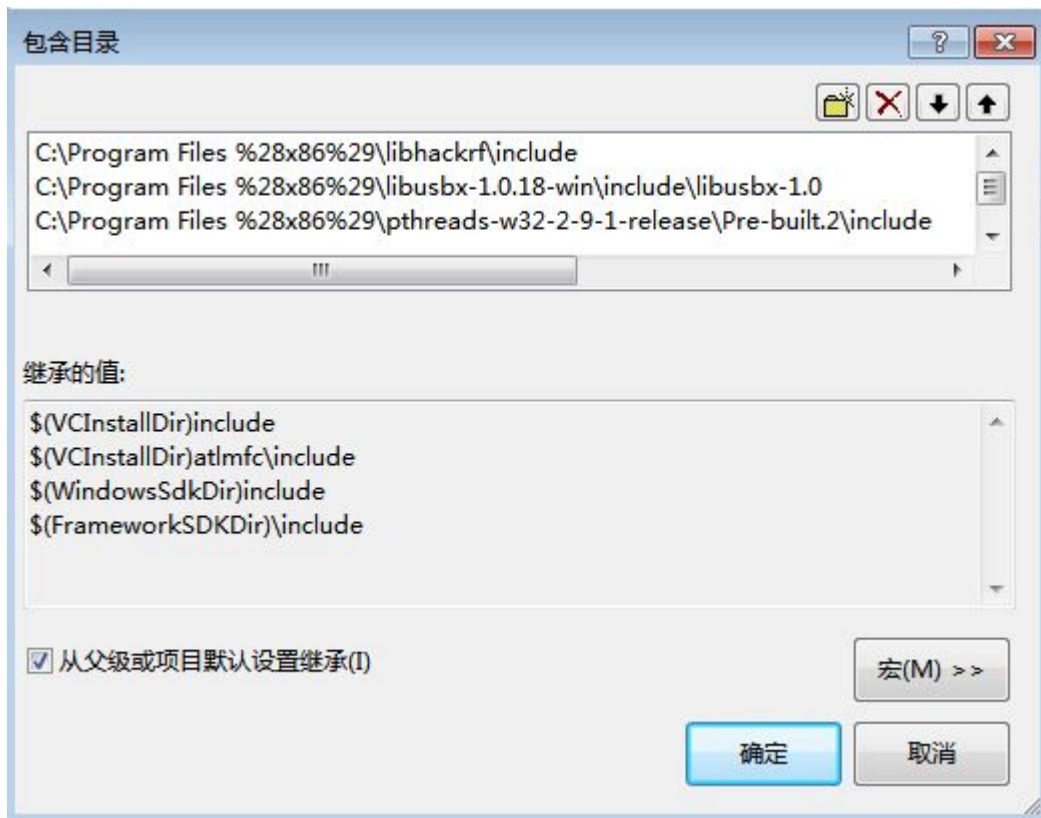
result = hackrf_version_string_read(device, &version[0], 255);
if (result != HACKRF_SUCCESS) {
    fprintf(stderr, "hackrf_version_string_read() failed: %s (%d)\n",
            hackrf_error_name((hackrf_error)result), result);
    getchar();
    return EXIT_FAILURE;
}
printf("Firmware Version: %s\n", version);

result = hackrf_board_partid_serialno_read(device, &read_partid_serialno);
if (result != HACKRF_SUCCESS) {
    fprintf(stderr, "hackrf_board_partid_serialno_read() failed: %s (%d)\n",
            hackrf_error_name((hackrf_error)result), result);
    getchar();
    return EXIT_FAILURE;
}
printf("Part ID Number: 0x%08x 0x%08x\n",
       read_partid_serialno.part_id[0],
       read_partid_serialno.part_id[1]);
printf("Serial Number: 0x%08x 0x%08x 0x%08x 0x%08x\n",
       read_partid_serialno.serial_no[0],
       read_partid_serialno.serial_no[1],
       read_partid_serialno.serial_no[2],
       read_partid_serialno.serial_no[3]);

result = hackrf_close(device);
if (result != HACKRF_SUCCESS) {
    fprintf(stderr, "hackrf_close() failed: %s (%d)\n",
            hackrf_error_name((hackrf_error)result), result);
```

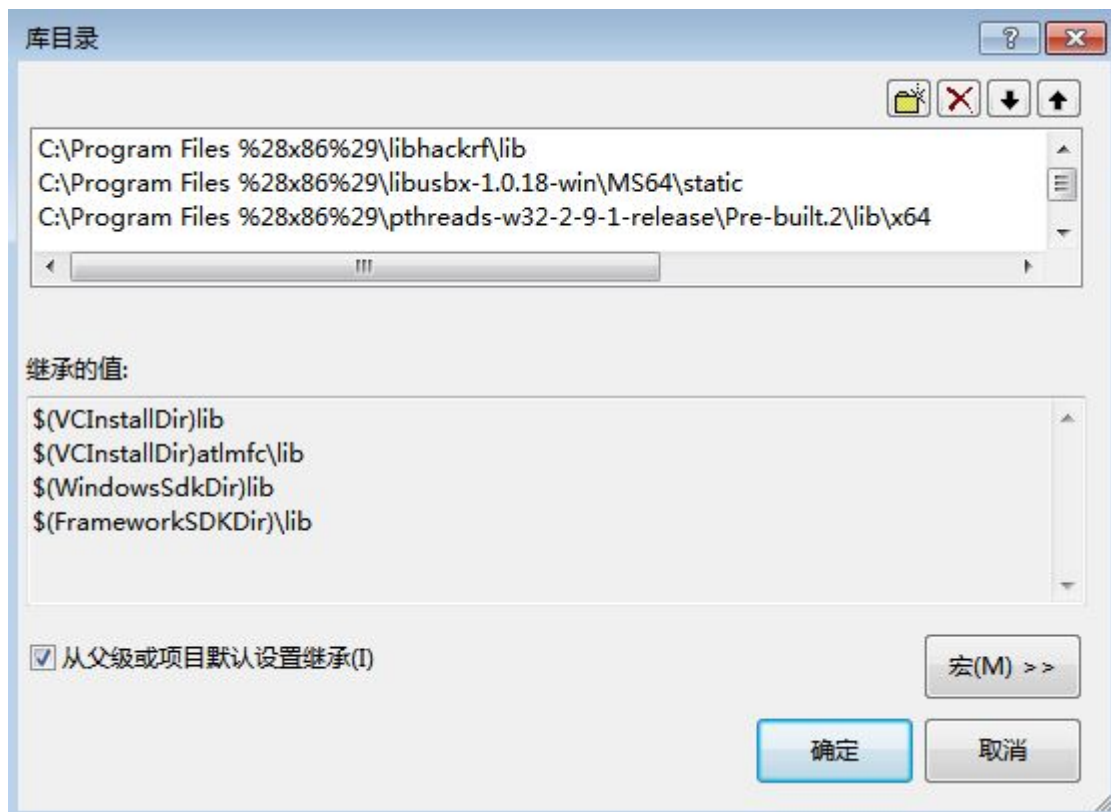
```
    getchar();  
    return EXIT_FAILURE;  
}  
  
hackrf_exit();  
printf("input ENTER to quite.\n");  
getchar();  
return 0;  
}
```

(2) 配置工程属性。包含目录设置与上面类似，配置如下图所示：

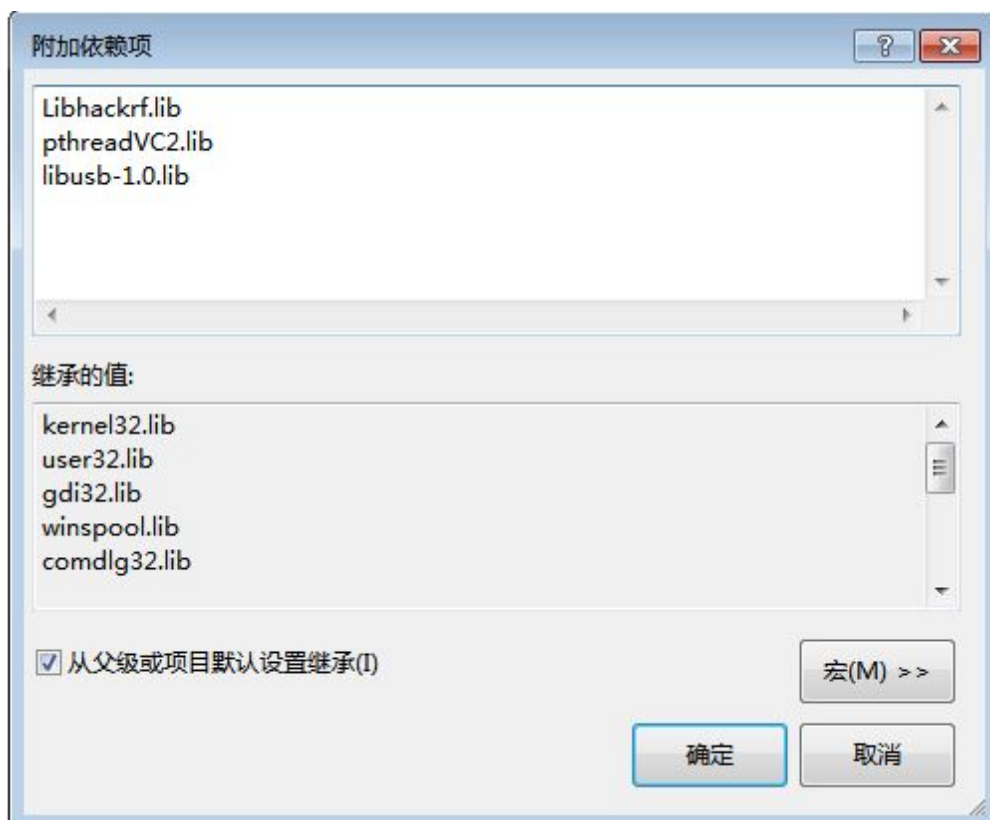


库目录设置稍有不同，还要加入 Libhackrf，配置如下图所示：





展开“链接器”中的“输入”选项，点击右侧的“附加依赖项”，编辑，添加如下内容：

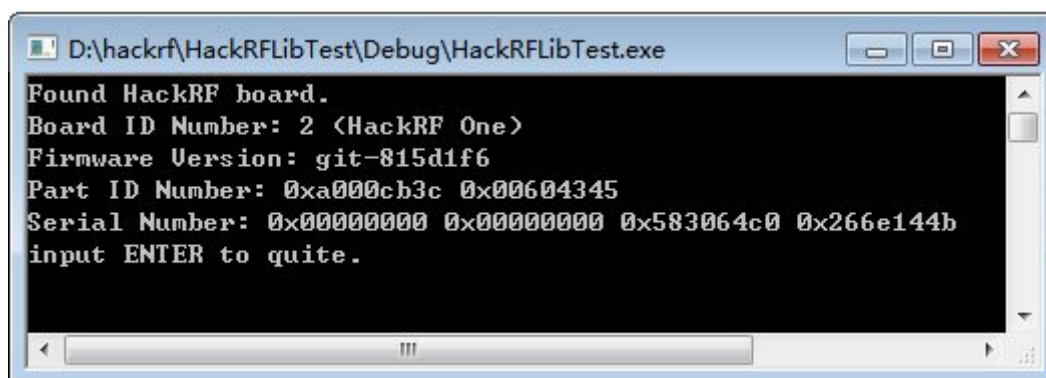




注意，如果编译报错提示找不到库文件，可直接将 C:\Program Files (x86)\libusb-1.0.18-win\MS32\static 里面的 libusb-1.0.lib 和 C:\Program Files (x86)\pthread-w32-2-9-1-release\Pre-built.2\lib\x86 里面的 pthreadVC2.lib 两个库文件直接添加到工程。

(3) 编译运行即可得到如下所示的演示效果。（注意要用 zadig\_2.1.0 安装好 hackrf 驱动）

若运行报错提示找不到 pthreadVC2.dll，则可将该文件放置在工程目录的 Debug 目录下即可。



## 3 Linux 环境下 HackRF 开发指南

### 3.1 安装 Linux 操作系统

GNURadio 和 gqrx 等开发工具均是基于 Linux 系统，因此，要使用这些开发工具，必须首先安装 Linux。Linux 有多个发行版本，推荐使用 Ubuntu 14.04 LTS。Ubuntu 是一个启动速度超快、界面友好、安全性好的开源操作系统，它由全球顶尖开源软件专家开发，适用于桌面电脑、笔记本电脑、服务器以及上网本等，并且它可以永久免费使用。

如果不想安装双系统，那么可以先装一个虚拟机，比如 VirtualBox 或 VMware 等，然后在虚拟机上安装 Linux。

### 3.2 搭建开发环境

HackRF One 在 Linux 下的开发工具包括 GNURadio 和 gqrx，可以自己安装，也可以使用我们提供的现成虚拟机文件。

### 3.2.1 安装 GNURadio 和 gqrx

Ubuntu 14.04 安装完成后，按“Ctrl+Alt+T”打开终端，输入以下命令即可：

```
sudo add-apt-repository ppa:gqrx/releases  
sudo apt-get update  
sudo apt-get upgrade  
sudo apt-get install gqrx gnuradio gr-osmosdr hackrf
```

如果安装时出现‘Held packages’类型的报错，那么应该这是由于之前已经进行过安装，这时，输入以下命令即可：

```
sudo apt-get dist-upgrade
```

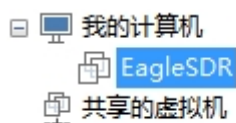
### 3.2.2 使用现成虚拟机文件

由于服务器不稳定等问题，自己动手安装 GNURadio 和 gqrx 等开发工具，经常会遇到各种各样的问题，而且就算一切顺利，也要耗时几个到十几个小时，可谓耗时耗力。为了方便用户，解除安装软件之苦，我们在光盘里提供了包括搭建好的完整开发工具的虚拟机文件，只需导入虚拟机即可使用，其步骤如下：

(1) 将光盘中“VM.rar”文件解压到硬盘；

(2) 打开虚拟机 VMware 12.0，点击“文件”—>“打开”，在弹出的对话框中选择“EagleSDR.vmx”，并点击打开；

(3) 在 VMware 左侧窗口将出现名为“EagleSDR”的虚拟机，点击“开启此虚拟机”；

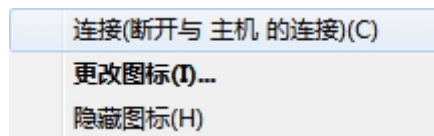


(4) 在后续出现的提示框中，均按默认选项设置，直至出现登录界面，输入密码“123456”；

(5) 虚拟机设置完成。

### 3.2.3 验证开发工具安装是否成功

将 HackRF One 连接电脑 USB 接口，并点击右侧下方的 USB 图标，并点击“连接（断开与主机的连接）”，如下所示：



### 1) 验证 USB 驱动

按“Ctrl+Alt+T”打开终端，输入

```
sudo hackrf_info
```

如果显示如下界面，则表示 USB 驱动安装成功

```
eaglesdr@ubuntu: ~  
eaglesdr@ubuntu:~$ hackrf_info  
Found HackRF board 0:  
Board ID Number: 2 (HackRF One)  
Firmware Version: git-815d1f6  
Part ID Number: 0xa000cb3c 0x00604345  
Serial Number: 0x00000000 0x00000000 0x583064c0 0x266e144b  
eaglesdr@ubuntu:~$
```

### 2) 验证 GNURadio

输入

```
gnuradio-companion
```

如果弹出 GNURadio 界面，则表示 GNURadio 安装成功。

### 3) 验证 gqrx

输入

```
gqrx
```

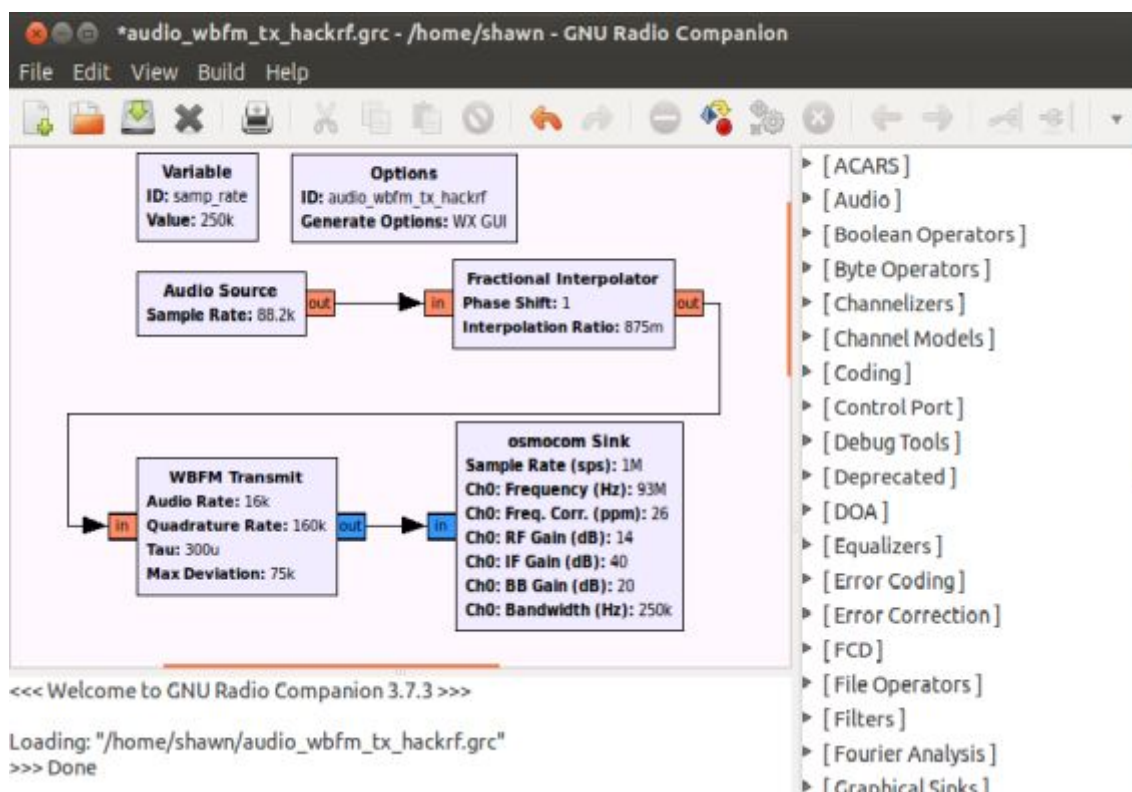
如果弹出 gqrx 界面，则表示 gqrx 安装成功。

## 4 GNURadio 使用指南

### 4.1 GNURadio 基本操作

#### 1) GNURadio 界面

GNURadio 界面与常用软件类似，顶部为菜单和快捷菜单，正中为画布，底部为日志窗口，显示程序运行时的输出信息，右侧为模块列表，用于选择添加功能模块。GNURadio 界面如下所示：



## 2) 添加功能模块

点击右侧的模块列表，可向画布中添加相应的功能模块，由于 GNURadio 中包含的功能模块各类较多，可用“Ctrl+F”打开搜索窗口，搜索相应的功能模块。找到所需的功能模块，双击即可完成添加。添加过程如下所示：



模块添加完成后，需要按相应的功能进行连接，连接模块时，分别点击需要连接的两个模块，即可自动完成两个模块之间的连线。若要断开连接，单击选中连线，按“Del”删除即可。

## 3) 设置模块属性

完成模块添加和连接之后，还需对模块进行设置。双击模块弹出属性窗口，可对模块的各项参数进行设置，如下所示为“Options”模块的属性窗口，设置整个工程的属性，如下所示：

**Parameters:**

ID	audio_wbfmtx_hackrf
Title	
Author	
Description	
Window Size	1280, 1024
Generate Options	WX GUI
Run	Autostart
Max Number of Output	0
Realtime Scheduling	Off

**Documentation:**

The options block sets special parameters for the flow graph. Only one option block is allowed per flow graph.

Title, author, and description parameters are for identification purposes.

Cancel OK

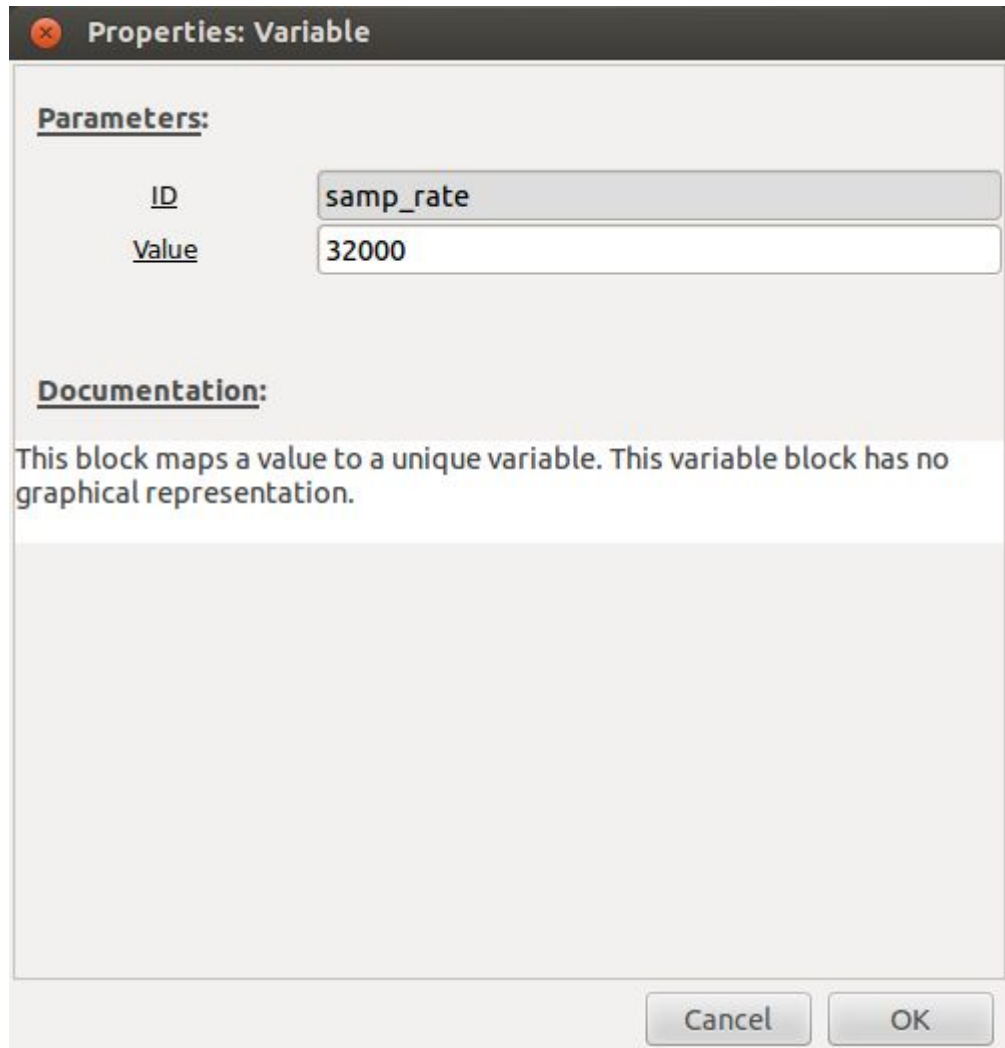
- ◆ ID: 工程 ID
- ◆ Title: 工程标题
- ◆ Author: 作者
- ◆ Description: 工程描述
- ◆ Window Size: 设置窗口大小
- ◆ Generate Options: 设置生成代码的选项，可选择
  - “WX GUI”：使用 WX 工具包
  - “QT GUI”：使用 QT 工具包
  - “NO GUI”：无 GUI，命令提示符模式
  - “Hier Block”：层次化模块
- ◆ Run: 设置运行选项，可选择“Autostart”或“off”
- ◆ Realtime Scheduling: 设置是否打开实时调度，可选择“on”或“off”

## 4.2 常用功能模块介绍

除了刚才已经介绍过的“Options”模块，常用功能模块还有“Variable”、“Signal Source”、“Add”、“WX GUI Slider”、“WX GUI Scope Sink”等。

### 1) Variable 模块

Variable 模块用于定义全局变量，并设置相应的变量值，可以在整个工程的别的模块中直接调用，如下所示：



上图定义了一个“sample\_rate”变量，并设置其值为 32000，则在整个工程的其它模块中均可使用该变量，如果要改变采样率，则只需改变“Variable”模块中的 sample\_rate 的值即可，而不必每个地方都进行更改，既提高的工作效率，又减小了错误设置的机率。

### 2) Signal Source 模块

Signal Source 模块用于产生常用的信号，包括直流信号、正弦信号、余弦信号、方波信号、三角波信号和锯齿波信号，其属性窗口如下所示：





- ◆ ID: 模块 ID
- ◆ Output Type: 输出类型，可选择
  - “Complex”：复数类型
  - “Float”：浮点类型
  - “Int”：整型
  - “Short”：短整型
- ◆ Sample Rate: 设置采样率，可以设置具体数值，也可以输入变量名
- ◆ Waveform: 选择输出波形类型
  - “Constant”：常数
  - “Cosine”：余弦波
  - “Sine”：正弦波
  - “Square”：方波
  - “Triangle”：三角波

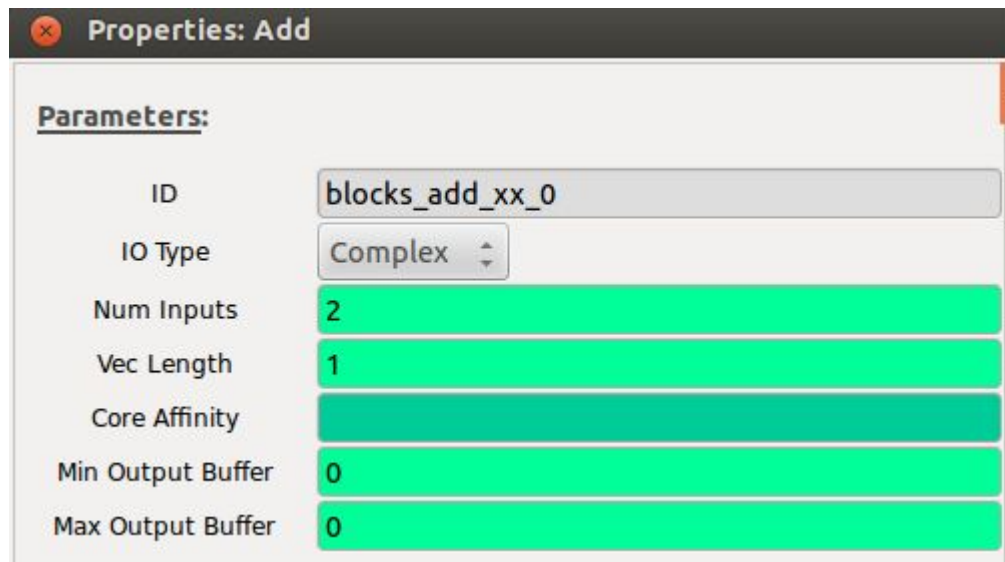
“Saw Tooth”：锯齿波

- ◆ Frequency: 设置波形频率
- ◆ Amplitude: 设置波形幅度

对于“Core Affinity”、“Min Out Buffer”和“MaxOut Buffer”一般不用设置，使用默认值即可。

### 3) Add 模块

Add 模块用于信号之间叠加，其属性窗口如下：



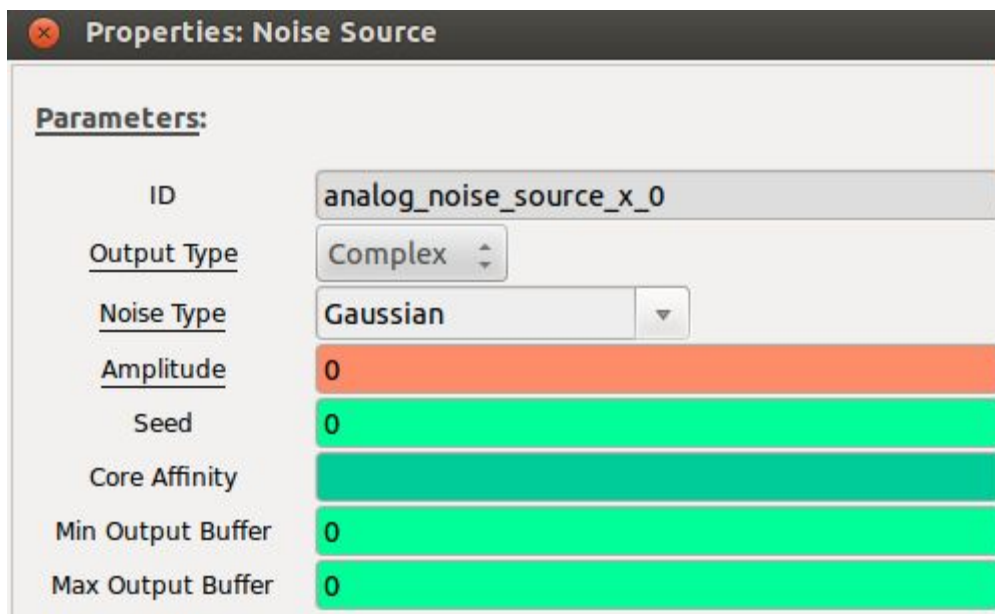
- ◆ ID: 模块 ID
- ◆ IO Type: IO 类型，可选择
  - “Complex”：复数类型
  - “Float”：浮点类型
  - “Int”：整型
  - “Short”：短整型
- ◆ Num Inputs: 设置输入端口数量
- ◆ Vec Length: 设置矢量长度

对于“Core Affinity”、“Min Out Buffer”和“MaxOut Buffer”一般不用设置，使用默认值即可。

### 4) Noise Source 模块

Noise Source 模块用于产生噪声信号，其属性窗口如下所示：





- ◆ ID: 模块 ID
  - ◆ Output Type: IO 类型，可选择
    - “Complex”：复数类型
    - “Float”：浮点类型
    - “Int”：整型
    - “Short”：短整型
  - ◆ Noise Type: 设置噪声类型，可选择
    - “Gaussian”：高斯噪声
    - “Uniform”：均匀噪声
    - “Laplacian”：拉普拉斯噪声
    - “Impulse”：脉冲噪声
  - ◆ Amplitude: 设置噪声幅度
  - ◆ Seed: 设置种子
- 对于“Core Affinity”、“Min Out Buffer”和“MaxOut Buffer”一般不用设置，使用默认值即可。

#### 5) WX GUI Slider 模块

WX GUI Slider 模块用于定义一个变量，并可以在程序运行过程中通过滑块改变变量值，其属性窗口如下所示：



- ◆ ID: 变量 ID
- ◆ Label: 变量标签, 如果本项为空, 则使用 ID 名作为变量名
- ◆ Default Value: 变量默认值
- ◆ Minimum: 变量最小值
- ◆ Maximum: 变量最大值
- ◆ Num Steps: 变量步进数
- ◆ Style: 滑块样式, 可选择:

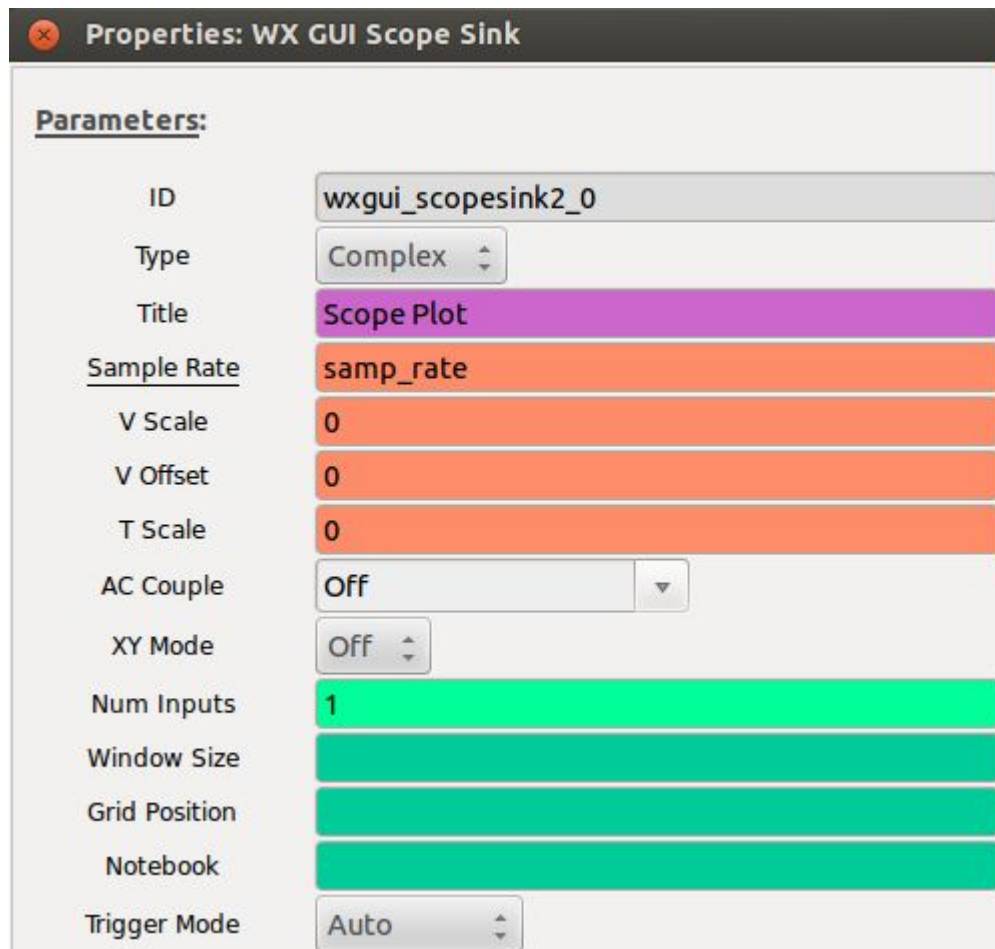
Horizontal: 水平滑块

Vertical: 垂直滑块

如上图所示, 定义了一个变量 `freq`, 默认值为 50, 最小值和最大值分别为 0 和 100, 步进数为 10, 亦即步进为  $100/10=10$ , 滑块样式为水平滑块。

#### 6) WX GUI Scope Sink 模块

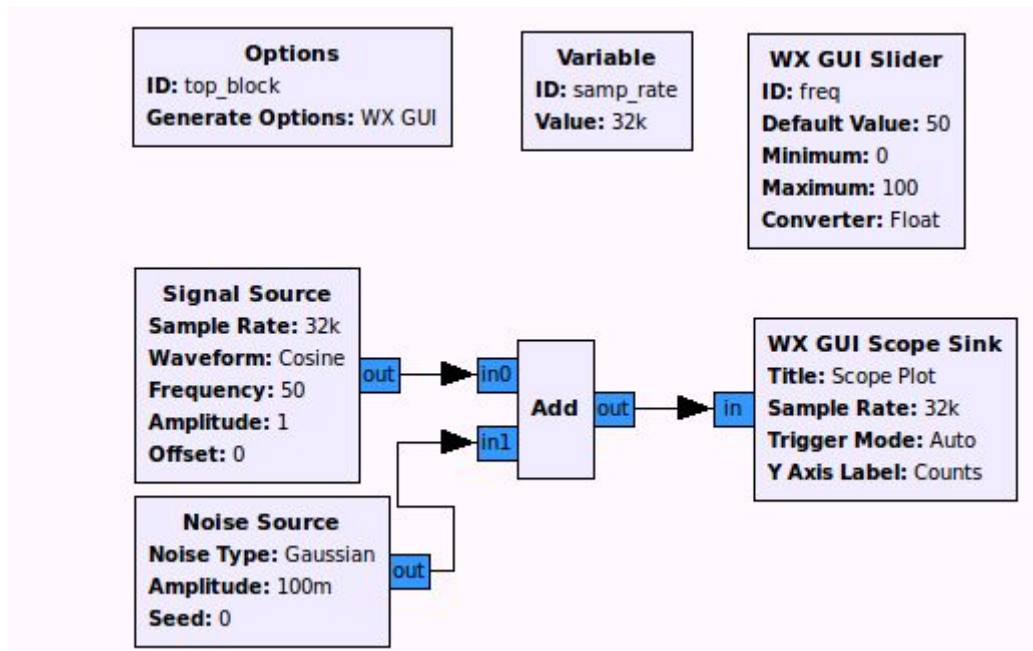
WX GUI Scope Sink 模块, 类似于一个数字示波器, 用于观察信号波形, 其属性窗口如下:



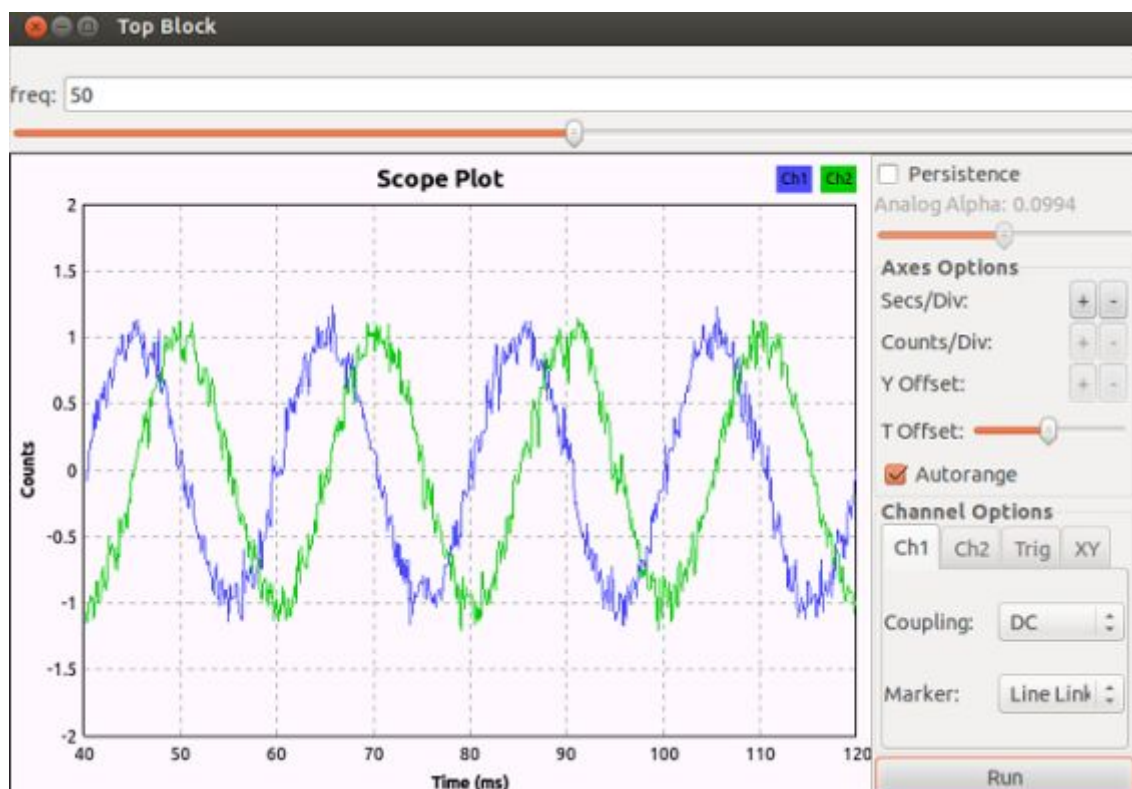
- ◆ ID: 模块 ID
- ◆ Type: 信号类型，可选择
  - “Complex”：复数类型
  - “Float”：浮点类型
- ◆ Title: 模块标题
- ◆ Sample Rate: 设置采样率
- ◆ V Scale: 设置 V 轴范围
- ◆ V Offset: 设置 V 轴偏移
- ◆ T Scale: 设置 T 轴范围
- ◆ AC Couple: 是否打开 AC 耦合
  - “on”
  - “off”
- ◆ XY Mode: 是否打开 XY 模式
- ◆ Num Inputs: 输出信号通道数目
- ◆ Trigger Mode: 选择触发模式

## 4.3 GNURadio 工程示例

根据上述对 GNURadio 常用模块和基本操作的介绍,我们创建一个简单的工程示例,其框图如下所示:



照上图添加功能模块,并进行连接,设置各相关参数后,点击“Generate the flow graph”生成流图对应的 python 代码,然后点击“Execute the flow graph”运行流图,出现如下的波形图,可以看出在余弦波上叠加了高斯噪声,通过拖动滑块,可以改变信号的频率。



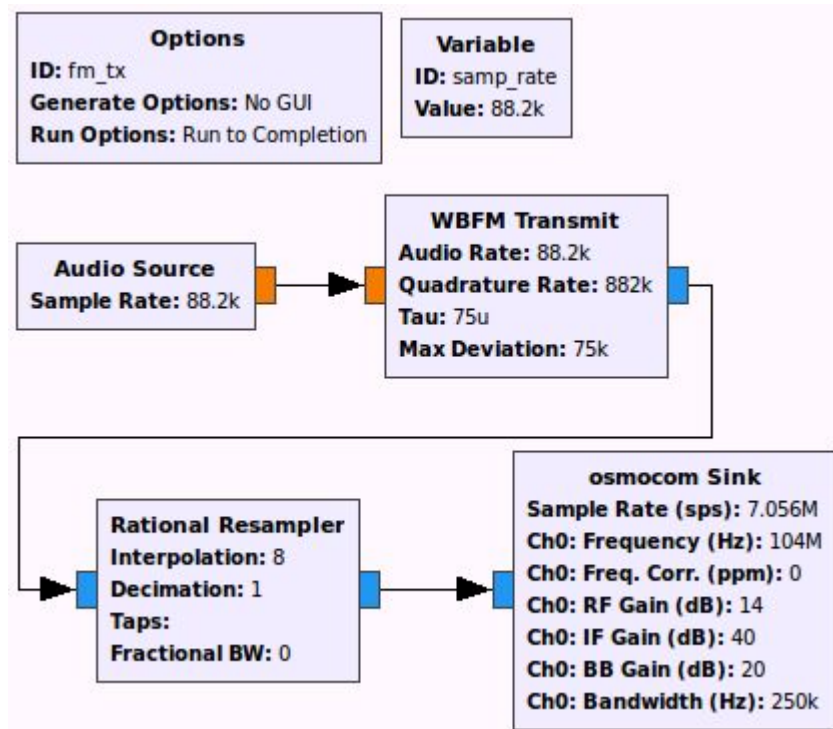
本章仅对 GNURadio 中的基本模块进行了简要介绍，GNURadio 包括了几百种模块，不可能一一尽述，但每个模块的“Documentation”选项页均有模块功能的详细说明，读者可以自行参照学习。

## 5 HackRF One 开发实例

本章通过两个简单的例子来说明 HackRF One 与 GNURadio 联合开发软件无线电应用的方法，开发环境为 Ubuntu 12.04.3 LTS，GNURadio v3.7.8.1。在“资料光盘\Linux 软件\示例工程”目录下提供了两个实例的工程文件。

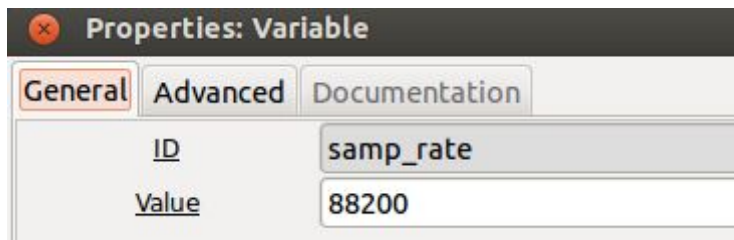
### 5.1 HackRF One 发射 FM 音频

本节介绍 HackRF One 将 PC 麦克风音频通过 FM 方式调制发射的过程。打开 GNURadio，按下图所示，搭建 Flow Graph，由图可知，本例中使用到的模块包括“Variable”、“Audio Source”、“WBFM Transmit”、“Rational Resampler”和“osmocom Sink”。



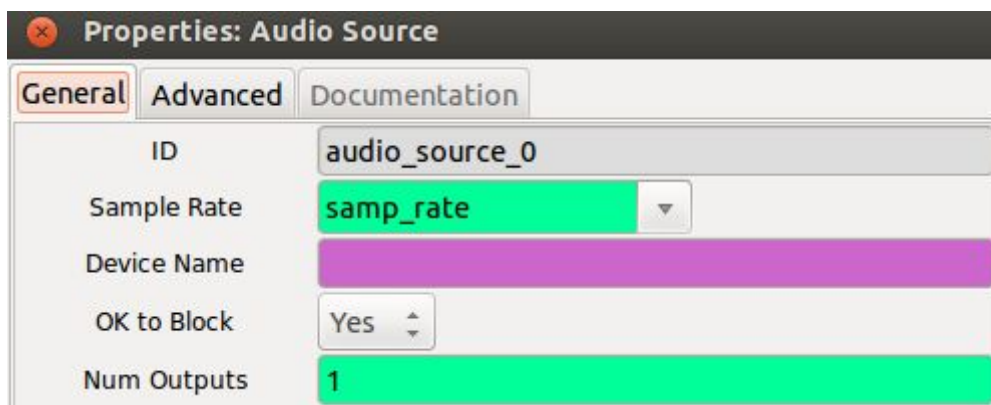
各功能模块参数如下：

#### 1) Variable 模块



- ◆ ID: samp\_rate
- ◆ Value: 88200

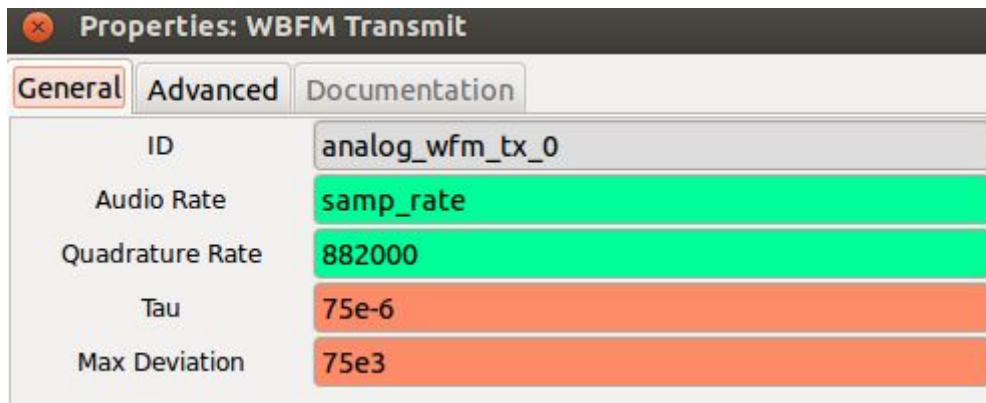
#### 2) Audio Source 模块



- ◆ ID: audio\_souvce\_0
- ◆ Sample Rate: samp\_rate

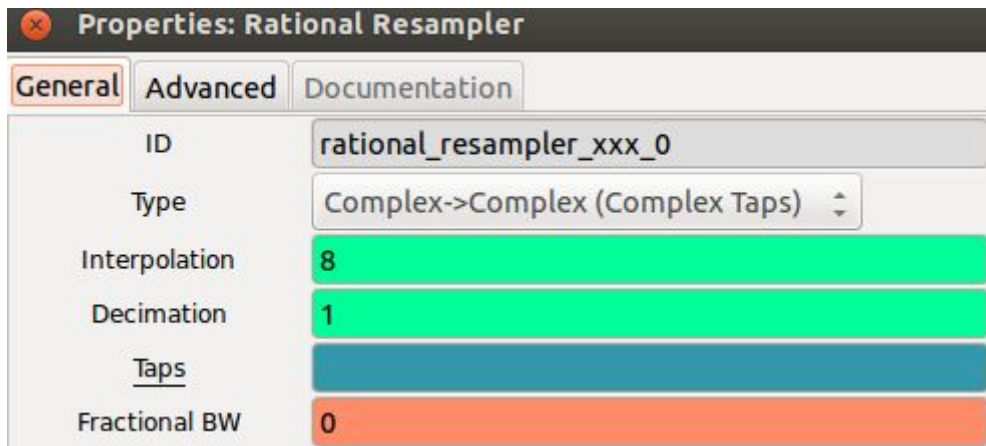


## 3) WBFM Transmit 模块



- ◆ ID: analog\_wfm\_tx\_0
- ◆ Audio Rate: samp\_rate
- ◆ Quadrature Rate: 882000
- ◆ Tau: 75e-6
- ◆ Max Deviation: 75e3

## 4) Rational Resampler 模块



- ◆ ID: rational\_resampler\_xxx\_0
- ◆ Type: Complex->Complex(Complex Taps)
- ◆ Interpolation: 8
- ◆ Decimation: 1
- ◆ Fractional BW: 0

## 5) osmocom Sink 模块

Properties: osmocom Sink	
General   Advanced   Documentation	
ID	osmosdr_sink_0
Input Type	Complex float32
Device Arguments	
Sync	don't sync
Num Mboards	1
Mb0: Clock Source	Default
Mb0: Time Source	Default
Num Channels	1
Sample Rate (sps)	7056e3
Ch0: Frequency (Hz)	104e6
Ch0: Freq. Corr. (ppm)	0
Ch0: RF Gain (dB)	14
Ch0: IF Gain (dB)	40
Ch0: BB Gain (dB)	20
Ch0: Antenna	
Ch0: Bandwidth (Hz)	250000

- ◆ ID: osmosdr\_sink\_0
- ◆ Input Type: Complex float32
- ◆ Sample Rate: 7056e3
- ◆ Ch0:Frequency(Hz): FM 发射频率, 88M-108M
- ◆ Ch0:RF Gain(dB): 射频增益, 对于 HackRF One, 其射频增益只有两个档位, 打开射频放大器时为 14dB, 关闭射频变压器时为 0dB
- ◆ Ch0:IF Gain(dB): 中频增益, 可调
- ◆ Ch0:BB Gain(dB): 基带增益, 可调
- ◆ Ch0:Bandwidth(Hz): 信号带宽

参数设置完毕, 将 HackRF One 连接电脑, 点击 “Generate the flow graph” 生成流程图对应的 python 代码, 然后点击 “Execute the flow graph” 运行流程图, 打开 FM 收音机, 并将频率设置为 104M, 收音机中应能收到电脑麦克风的语音。

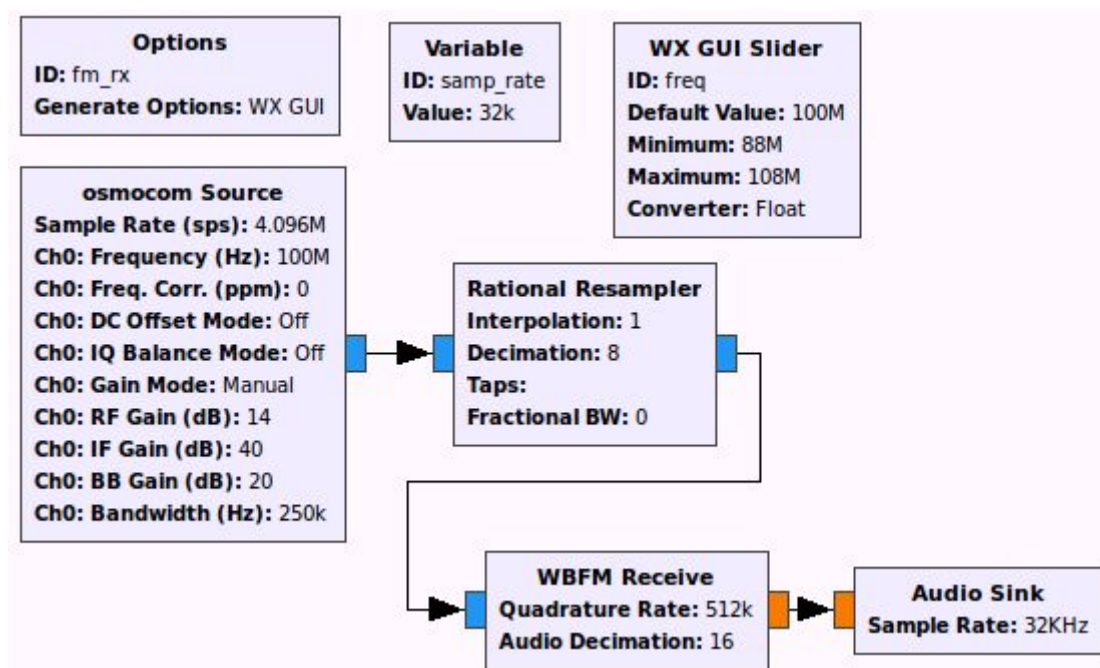
说明: 上述设置和采样率、积分率、抽取、插值参数需满足如下公式:



由于 HackRF One 板上的 ADC/DAC 速率最大为 20Msps，因此，上述公司中的采样速率不能超过 20Msps。

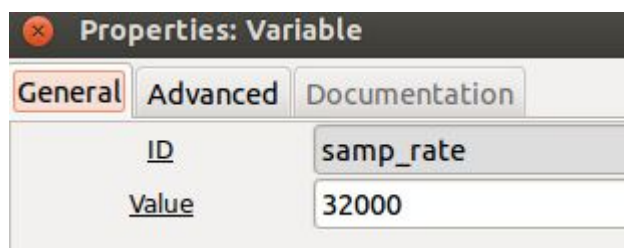
## 5.2 HackRF One 接收 FM 广播

本节介绍 HackRF One 接收 FM 广播的过程。打开 GNURadio，按下图所示，搭建 Flow Graph，由图可知，本例中使用到的模块包括“Variable”、“WX GUI Slider”、“osmocom Source”、“Rational Resampler”、“WBFM Receive”和“Audio Sink”。



各功能模块参数如下：

### 1) Variable 模块



### 2) WX GUI Slider 模块

WX GUI Slider 模块用于在程序运行时，调整收信频率，其设置如下：

**Properties: WX GUI Slider**

General Advanced Documentation

ID	freq
Label	
Default Value	100e6
Minimum	88e6
Maximum	108e6
Num Steps	1000
Style	Horizontal
Converter	Float
Grid Position	
Notebook	

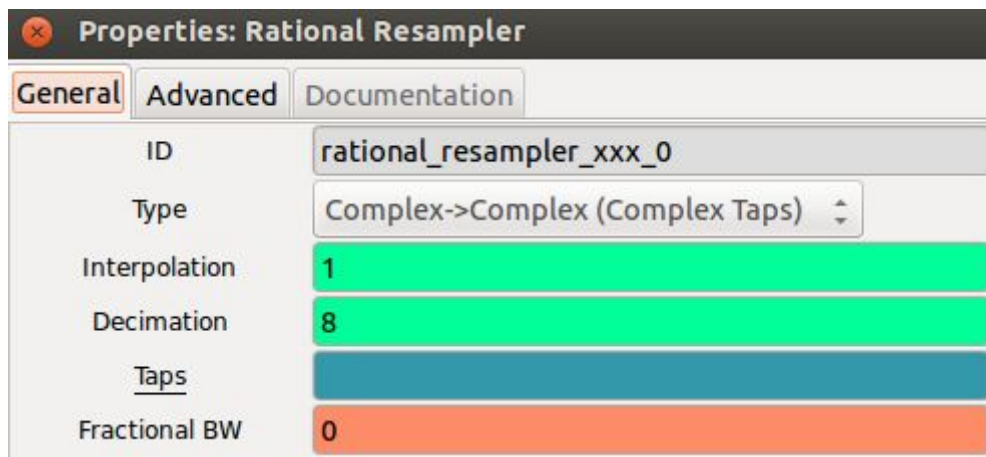
3) osmocom Source 模块

**Properties: osmocom Source**

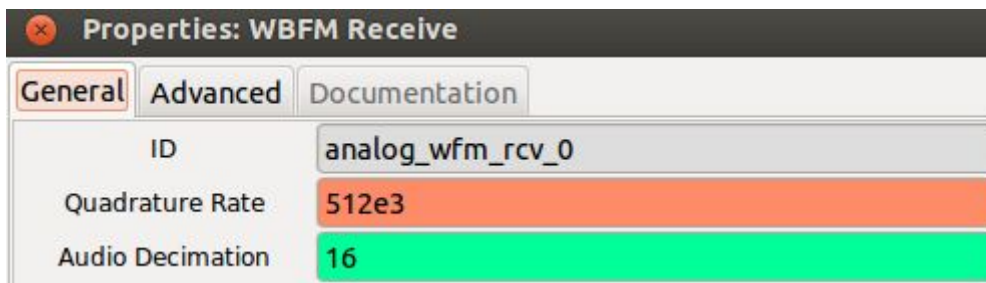
General Advanced Documentation

Num Mboards	1
Mb0: Clock Source	Default
Mb0: Time Source	Default
Num Channels	1
Sample Rate (sps)	4096000
Ch0: Frequency (Hz)	freq
Ch0: Freq. Corr. (ppm)	0
Ch0: DC Offset Mode	Off
Ch0: IQ Balance Mode	Off
Ch0: Gain Mode	Manual
Ch0: RF Gain (dB)	14
Ch0: IF Gain (dB)	40
Ch0: BB Gain (dB)	20
Ch0: Antenna	
Ch0: Bandwidth (Hz)	250e3

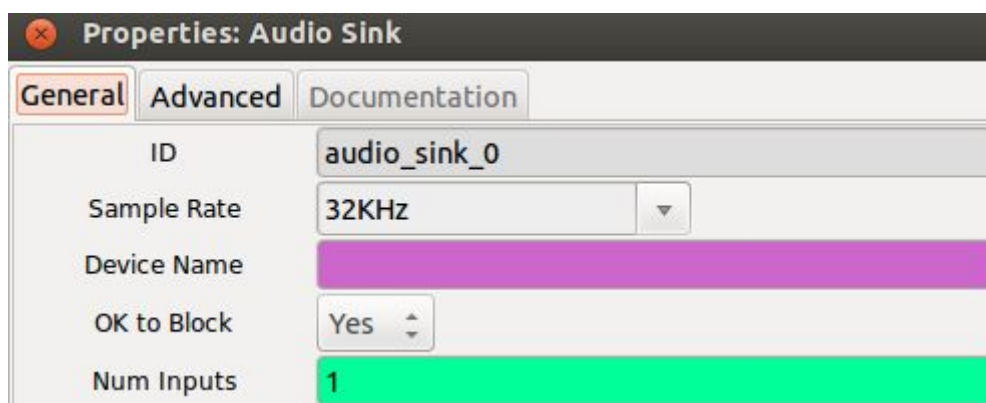
## 4) Rational Resampler 模块



## 5) WBFM Receive 模块



## 6) Audio Sink 模块



参数设置完毕，将 HackRF One 连接电脑，点击“Generate the flow graph”生成流图对应的 python 代码，然后点击“Execute the flow graph”运行流图，将显示如下界面



拖动滑块，将接收频率设置到相应的电台频率，电脑扬声器应能收到 FM 电台

的声音。