

jamesshao8



10

23

80

主题

帖子

积分

注册会员



积分

80

发消息



10

23

80

主题

帖子

积分

注册会员



积分

80

发消息

360

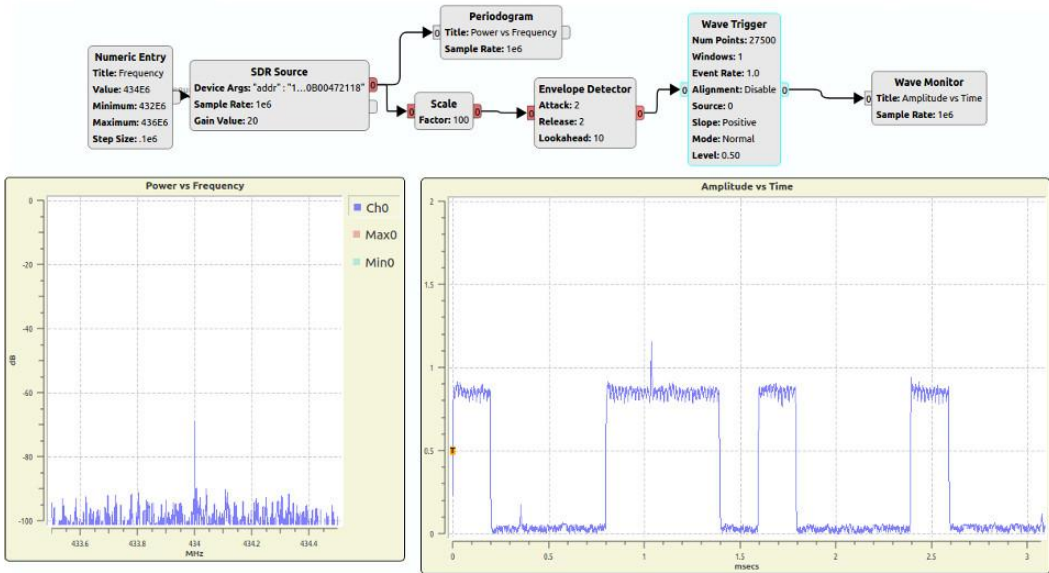
无线电安全研究院

件无线电 [LimeSDR] Made Simple 6 其它环境（上）

[LimeSDR] Made Simple 6 其它环境（上） [复制链接]

发表于 2018-12-22 11:31:28 | 只看该作者

楼主 电梯直达



Power vs Frequency

Amplitude vs Time

在Pothos和GNU Radio中接收ASK/OOK信号

这是第六篇LimeSDR教程。上一篇文章我们在Octave中发射和接收ASK信号。我们没怎么详细讲调制技术。

在接下来的文章里，我们会研究单级ASK的发射和接收，这种ASK也被叫做OOK开关键控。这个调制方式很好理解，经常用在门铃和车库开门钥匙里。

如果你要做这个实验，你需要额外的硬件，比如433MHz（UHF频段）的ASK/OOK设备。如果你有一个遥控钥匙，那么就够了。

我们还是要警告你，把天线从不用的TX口处取走，这样就不会不小心发射出信号了。要小心一些，不要去干扰邻居的门铃。

深入学习ASK

接下来我们会讲到一些通信理论。前面说过，ASK是最简单的调制方式。ASK的实现方式里最简单的一种是用单音来表示1，然后没有单音表示0。这也是OOK，2个等级：开和关。这个单音一般来说就是载波频率，在我们这里就是433.9MHz（我们待会儿会讲如何得到这个频率）。

我们说过这个协议很简单，所以我们可以用一个足够好的示波器观察，并直接用眼睛看出数据。

和很多简单的概念一样，我们还需要了解一些其它知识。在实际的无线电接收中，我们的电路需要时间去调整和锁定频率和增益。对于SDR和传统电路都一样。比如遥控钥匙：它们的发射频率经常会少许变化，中心频率会偏移。

频率偏移

在模拟电路里，我们使用受控环路锁定载波，并调整接收电路达到匹配。PLL或者一个简单的共振电路就可以完成，原理都比较类似。在SDR里，我们也能用同样的方法来解决，或者直接忽略掉这个问题，而是用后期的数字滤波来解决。这需要假设我们有足够的带宽来捕捉信号。正确的ASK接收机对于频率偏移和相位变化能过很好的免疫，所以我们一般不需要担心这些。

增益

信号振幅对于传统电路和SDR都是要注意的事项。最终都需要用某种增益控制来解决。比如AGC（自动增益控制）是最好的，但是手动调整也许。

在硬件中，我们可以调节LNA或者在软件里乘以一个信号解决，各有优点。

前导码

要解决锁定和增益问题需要时间，我们不能在信号一发出就立即完成。增益控制和频率控制都需要一定时间实现。这时可以引入前导码来解决。简单来说，这是一段信号，在这段时间内接收机会去完成锁定，锁定后才接收真正的数据。这个技术不但在射频通信里用到。你现在使用的电脑

jamesshao8




10  
主题

23  
帖子

80  
积分

注册会员



积分80

发消息

社区

找到频率

我们从eBay买了一个无线钥匙，用于ASK发射机，我们会解码它发出的信号。在欧洲，这类钥匙的频率一般都是433MHz的频段。但是有效解码，我们需要找到准确的频率值。

我们使用LimeSuiteGUI来做这个工作，非常简单，你可以仿照第二篇文章，用FFT viewer来观察发射出的频率。把SXR频率改为433MHz，我们就能看到FFT图了，然后按下钥匙上的按钮，会出现一个新的尖峰。然后重新调整中心频率，直到尖峰在中间，这时我们就找到了钥匙的频率。我们的钥匙频率是433.9MHz。

本主题由 mobier 于 2018-12-25 13:52 设置高亮

 收藏

回复

举报

 楼主

发表于 2018-12-22 11:32:10 | 只看该作者

沙发

本帖最后由 jamesshao8 于 2018-12-22 11:34 编辑

设计一个SDR接收机

迄今为止，我们用的都是现成的设计，比如Josh的FM接收机。接下来我们要自己设计了。

观察一个一般化的接收链路，我们需要一些增益，可以在接收硬件中，也可以在数字部分（或者两者都有），最终我们需要检波，把它转为数据。

我们之前用过Pothos，所以这次还是用它。加入一个SDR Source（包含频率控制）和一个Wave Monitor，我们就有了最简单的ASK接收机了。把调谐频率设置为433.9MHz，然后把钥匙拿到接收天线旁边，按下钥匙按钮，我们就能看到微弱的波形了。这是最简单的接收机，效果不是太好，需要改进。

我们需要加入一些增益，并加入一个Periodogram。这个模块的作用就跟一个频谱分析仪一样。我们需要观察频谱图，这样我们就知道达到什么样的增益时会导致失真。理想情况下，我们需要得到在发生失真前的最大增益（失真的时候具体的样子可能是出现了超过1个尖峰）。在我们这里20dB是可以的，再上去信号就失真了。

使用Periodogram，我们可以看到中心频率上有一个很大的尖峰。这是RF信号的直流分量。我们可以用DC removal模块来去掉它。

到现在为止，信号还是优点小，我们可以：  
在数字部分增加信号的增益  
把图像上的增益开大点


用一个放大镜看信号也可以看着大一点，但是没必要这样。wave monitor有一个坐标轴的控制功能，可以实现这个要求，虽然这对于解调波形没有用，因为图形界面本身会消耗性能，而且即使是反应最快的人在解调信号时还是会跟不上。

我们把前面说的几点实现一下。为了实现增益，我们乘以一个数字，这样可以把信号放大20倍（很简单）。然后用一个简单的解调算法，叫做Envelope Detector。这样就能检测载波的包络线了（对应于值为1的时候），然后把检测出的结果转变为数字输出。

www.radiohack.net/forum.php?mod=viewthread&tid=27&extra=page%3D2

2/4

jamesshao8



10

23


80

主题

帖子

积分

注册会员




积分

80

发消息

jamesshao8



10

23

80

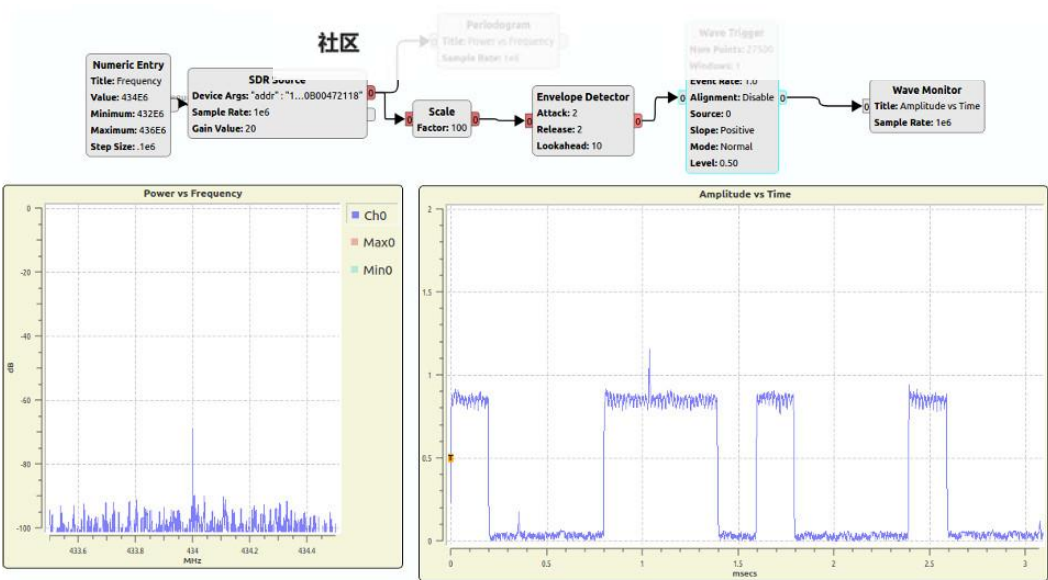
主题

帖子

积分

注册会员





我们可以加入一个Wave Trigger，并设置为normal模式，这样就能捕捉信号了（类似于一个示波器上的功能）。点的数量可以在Wave Trigger和Wave Monitor里设置，这样你就能观察到整个过程。如果这个数据是静态的，那可以手动解调，并保存为文件。这样我们就搞定了：一个简单但是有效的（尽管不是自动的）ASK接收机。

重量级程序：GNU Radio

迄今为止，我们都没说GNU Radio。Pothos和GNU Radio很接近，而且可以支持与GNU Radio的协同使用。你用过Pothos GUI后GNU Radio companion应该也会很熟悉了。如果你没安装，可以现在装一下

和Pothos一样，我们的流程图也类似：无线电接收->增益->解调->分析。要在GNU Radio里使用LimeSDR，我们需要有osmocore Source，并且在device arguments里填入driver=lime,soapy=0。

这里我们要讲一些与Pothos的区别。打开一个新文件后，我们会自动生成两个模块。一个是Variable，这里有sample\_rate这个变量。这个模块的功能顾名思义，存储这我们要使用的采样率。另一个模块叫作Options，里面是一些设置选项。我们需要把采样率设置为1e6（1Mps/s），并且把Generate options设置为WX，我们可以双击这些模块来更改这些设置。QT/WX代表不同的图形库，我们待会儿想用WX，所以要这么改。

我们可以在GNU Radio用各种方式实现前面在Pothos里做的工作。我们使用了下列模块（但是这不是唯一的方法，你也可以试试其它方法）：


- DC blocking模块
- RMS模块
- Thresholds模块
- Freq Xlating FIR Filter
- AM解调模块
- Clock Recovery MM

我们的目标是制作一个ASK解码设备，我们现在使用最合适的方式实现。对我们来说Clock Recovery模块可能是最合适的，因为这个模块可以把信号恢复成二进制数据，并且保存下来。我们发现这个模块设计的时候比较复杂，下面的设置肯定不是最佳的设计，要做到最优化很难。我们调整的参数是omega值。我们试了几次，就设置为18。我们的采样率是1Mps/s。这个设置是一个猜测的值，因为我们不知道符号率到底是多少。如果我们这时候接上一个wave monitor，这里的采样率等于采样率/符号率。

Clock Recovery模块在消除载波后表现得更好，因为载波的存在会影响很多事情。我们使用AM模块来实现。另一种方式是使用RMS模块，实现一个包络线检波器。要把它与Clock Recovery模块结合使用，波形需要加上一个负常数，然后移到0点附近。最后加上一个binary slicer和file sink就完成了这个ASK解析设备。

回复

举报

 楼主 | 发表于 2018-12-22 11:32:51 | 只看该作者

板凳

等一下，这个设备还没那么好，因为输出的格式还需要翻译。改变数据格式是另一个话题了，现在有一些程序可以自动完成这个任务，比如grc\_bit\_converter。我们把前面生成的数据经过这个程序转换后输出如下图。我们可以清晰地看到前导码和后续的二进制数据。

从这个数据可以发现，我们每个符号对应的采样太多，因此说明我们的omega值是错的。我们把omega值增加到25，这样每个符号对应的二进制位输出就少了一些，但是还是有问题。

我们可以进一步地精调，或者想办法计算出正确的omega值。然而，如果只是为了解码数据流，现在这样就够了。

根据现在的数据，我估计现在相当于用了1~1.5个字节来表示原来的一个比特，这显然太多了，但是我们使用现在的输出可以方便的用程序来转换出原来的数据。比如假设输出的数据低于A0就表示0，高于A0就表示1。比如，输出的1个字节对应于原来的1个比特，而3个字节对应于原来的2个比特。我们现在只是大概的估计，要做得更好也可以，但是没必要。

积分80

发消息

jamesshao8



10

23

80

主题

帖子

积分

注册会员



积分80

发消息

社区

我们可以把之前做的假设与包络线检波器做对比，你可以发现它们对应得很好，这样我们就可以认为差不多可以了。我们下次再回来继续优化，你也可以自己做一些优化。

最后

现在我们的结果已经比较好了，这篇文章暂时就讲到这。还有许多可以做的优化，比如AGC。还有许多其它的实现方案。写程序就这样，要实现一个目标会有多种方法，有些方法比另一些好。如果我们只是要看波形，Pothos和Wave Trigger模块就够了。如果我们要嗅探所有ASK数据，并且要实时解码，那么Clock Recovery模块就会更好。还有一点：记录ASK数据可能很有趣，但是请勿用于不良用途。

如果你还没尽兴，我们下次还会深入讲ASK。

回复

举报

返回列表