Hands-on 3

Transmissão e recepção da modulação AM utilizando o GNURadio (loopback)

**Introdução Teórica**

Modular significa preparar o sinal a ser transmitido de forma que ele se propague pelo meio de transmissão. Geralmente, a mensagem é impressa sobre uma **portadora**, a qual, no caso de modulações de onda contínua, é um sinal senoidal da seguinte forma:



sendo

* *A*c(*t*) = Amplitude
* *ω*c(*t*) = 2*πf*c(*t*) = frequência
* *φ*(*t*) = Fase

O processo de modulação por portadora senoidal explora os três parâmetros acima, gerando três tipos de modulação:

**Modulação em Amplitude (AM)**

* + *Ac*(*t*) ~*k*a*m*(*t*) - carrega a informação (varia linearmente de acordo com a mensagem)
  + *ω*c(*t*) e *φ*(*t*) são constantes

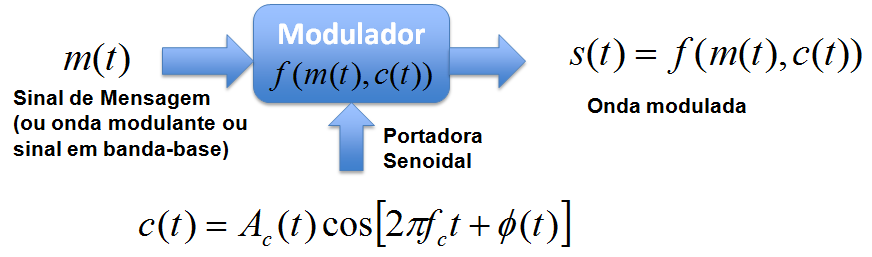
**Modulação em Frequência (FM)**

* + *ω*c(*t*) ~*k*f*m*(*t*) - carrega a informação (varia linearmente de acordo com a mensagem)
  + *Ac*(*t*) e *φ*(*t*) são constantes

**Modulação em Fase (PM)**

* + *φ*(*t*)~*k*p*m*(*t*) - carrega a informação (varia linearmente de acordo com a mensagem)
  + *Ac*(*t*) e *ω*c(*t*) são constantes

A figura a seguir ilustra de forma genérica o que deve ser entendido de um modulador de onda contínua. Considerando que *m*(*t*) é o sinal a ser transmitido (e.g. um sinal de voz) e *c*(*t*) é a onda portadora, a modulação é uma função de *m*(*t*) e *c*(*t*), a qual gera um sinal *s*(*t*) apropriado a transmissão sem fio.

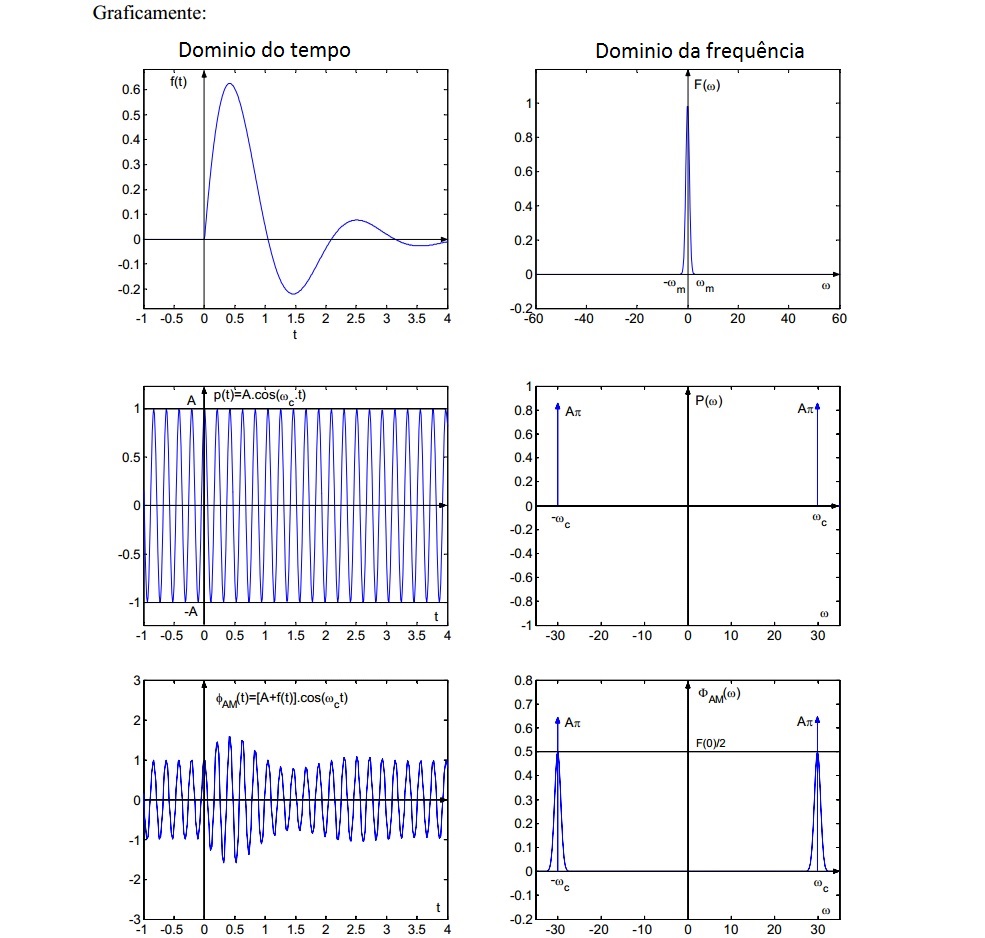
****

Na modulação AM-DSB (AM comercial),o sinal modulado tem a seguinte forma.



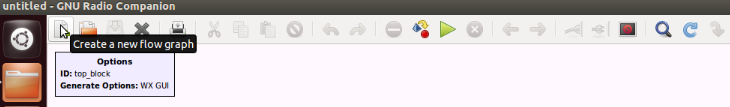
Note que a amplitude de *s*(*t*) varia de acordo com o sinal *m*(*t*). Se considerarmos uma entrada *m*(*t*) senoidal (i.e. um tom), o processo de modulação pode ser sumarizado na figura a seguir. Note que *s*(*t*) tem o formatos temporal e espectral bem peculiares.

Para um sinal de entrada qualquer, podemos ilustrar o AM-DSB como similar a figura a seguir[[1]](#footnote-1).

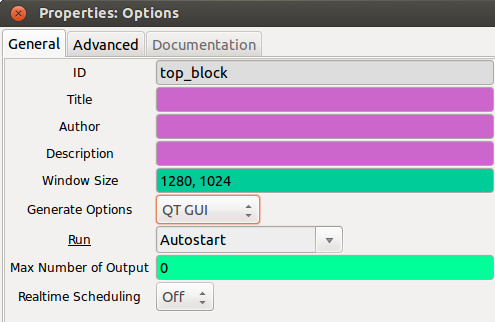
****

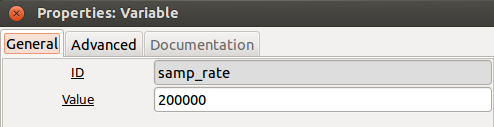
**Exercício  
  
OBJETIVO:** Usar conceitos básicos de manipulação de sinais e algumas dicas aprendidas em exercícios passados para, com o uso do conhecimento teórico, possamos construir um “loop-back” da transmissão e recepção de sinais AM-DSB (AM comercial).

1. Com o GNU RADIO COMPANION aberto, crie um novo projeto

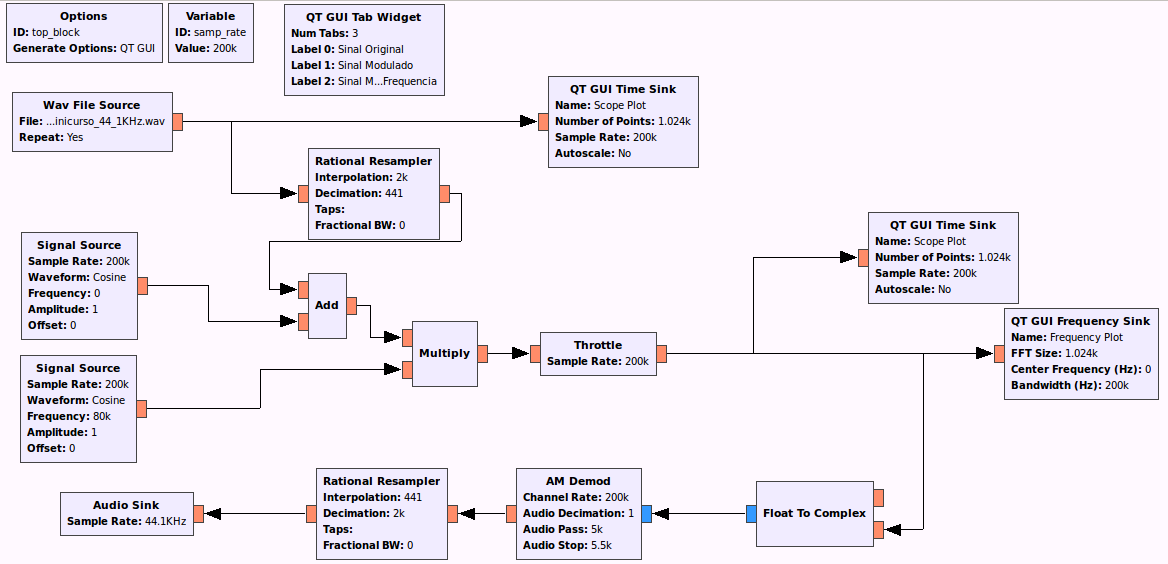


1. Modifique o campo *Generate Options* do bloco ***Options*** para a opção QT GUI.Abra o bloco ***Variable*** e modifique o *Sample Rate para 200000*.





1. Construa um projeto utilizando os blocos ***Wav File Source***, dois blocos ***Rational Resampler***,dois blocos ***Signal Source*,** um ***Add***, um ***Multiply***, um ***Throttle***, um ***FloattoComplex*,** um ***QT GUI Frequency Sink***, dois ***QT GUI Time Sink***, um ***QT GUI Tab Widget***,um ***AM Demod*** e um ***AudioSink***. Como de costume altere o campo *Type*de todos os blocos para *Float*, ao não ser os conectados a saída do bloco ***Float to Complex*** e a entrada do bloco ***AM Demod***. Conecte os elementos de forma que sua área de trabalho fique similar à figura a seguir.

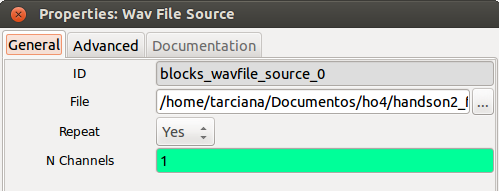


1. Como geradores de sinais, existem dois blocos ***Signal Source*** e um bloco ***Wav File Source***. Lembre que a forma da onda AM-DSB é: *.*

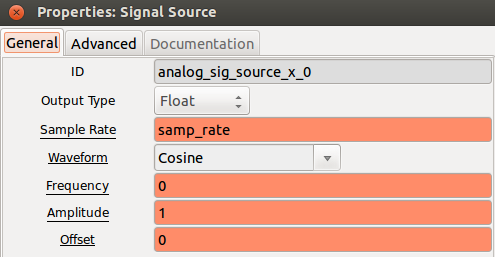


O bloco ***Wav File Source*** modela o sinal *m*(*t*). Quanto aos dois blocos ***Signal Source***, um modela a portadora **cos(2*πf*c*t*)** e o outro uma constante DC, representando *A*c. O restante dos blocos servem para demodular o sinal e tocar a saída na placa de som do computador. Os próximos passos orientam a configuração de todos os blocos.

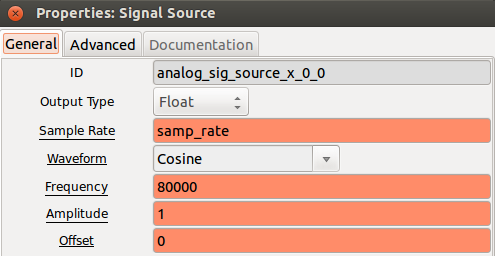
1. Clique duas vezes no bloco ***Wav File Source***. Clique nos “três pontinhos”. Localize a pasta onde existe algum arquivo **wav**. O caminho para o arquivo será mostrado no campo *File*. Modifique a opção *Repeat* para *Yes*. Isso fará com que o sinal do arquivo seja tocando continuamente. A sua configuração deve ficar similar à da figura a seguir. Escolha um arquivo com taxa de amostragem de 44,1 kHz.



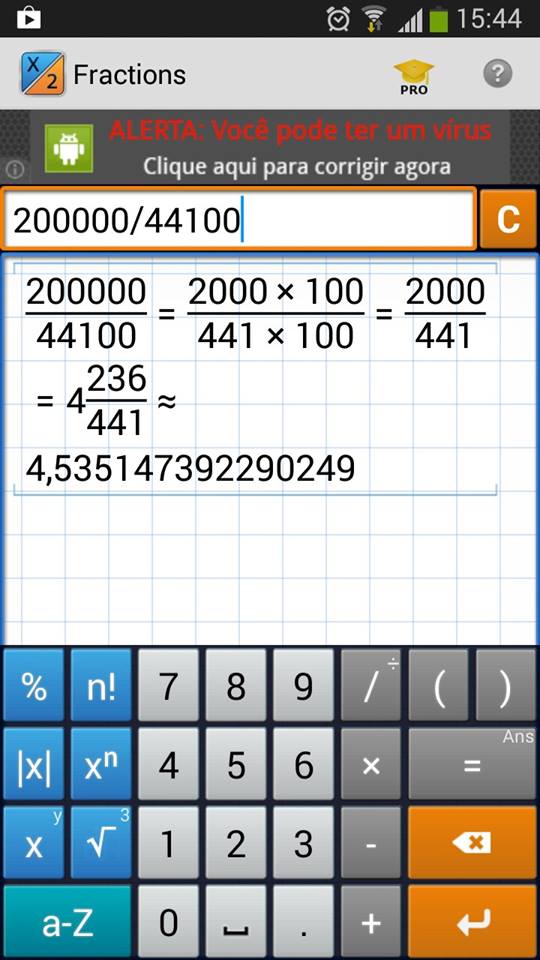
1. Clique duas vezes no *bloco* ***Signal* Source** que entra no bloco ***Add***. Ele será configurado com a constante *A*c. O bloco deve ser configurado com amplitude 1 e freqüência 0. A configuração deve ficar similar ao mostrado na figura a seguir.



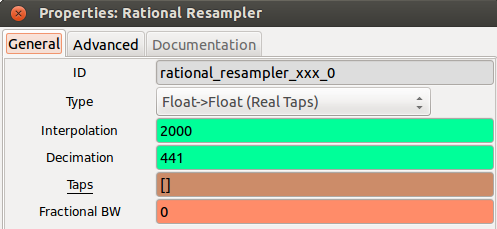
1. Configure agora o segundo bloco ***Signal Source***, o qual terá o papel da portadora. Como a portadora precisa ter uma frequência relativamente mais alta, você irá precisar de uma taxa de amostragem maior para formar o sinal, lembrando da regra da amostragem de Nyquist. Por isso escolhemos 200 kHz como taxa de amostragem de todo o experimento (configurado no Bloco ***Variable***, campo *Sample Rate*), pois iremos configurar a portadora para 80 kHz. Assim, configure o bloco com frequência de 80 kHz e taxa de amostragem igual a *sample\_rate***.** A configuração deve ficar similar ao mostrado na figura a seguir.



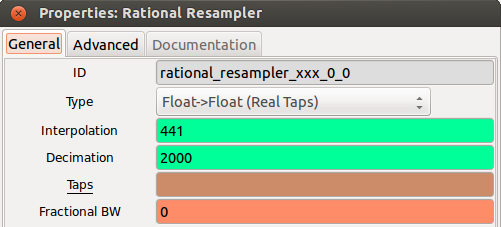
1. Não se esqueça de configurar o campo *Sample Rate* do bloco ***Audio Sink*** para 44100. Como usamos uma taxa de amostragem de 200 kHz no nosso projeto para sintetizar a portadora e vários outros blocos, precisamos adequar a taxa de amostragem dos blocos ***Wav File Source*** e do ***Audio Sink****, que operam a 44,1 kHz,* com a do resto do projeto. Para isso que serve o bloco ***Rational Resampler***. Operá-lo é simples. Tudo o que precisamos é a taxa que queremos que o sinal esteja e sua taxa atual. Colocamos em uma divisão, na qual a taxa que queremos é o numerador e a taxa atual é o divisor. Simplificamos essa operação a ponto de se ter os menores valores inteiros possíveis dessa divisão. Uma boa dica para quem usa *smartphones* com OS Android (ou IOS) é fazer o download gratuito da calculadora “*Fraction Calculator*”. Com os valores obtidos, o numerador passa a ser o valor do *Interpolation* e o divisor passa a ser o valor do *Decimation.* A figura a seguir mostra a saída do “*Fraction Calulator*” para o que precisamos.



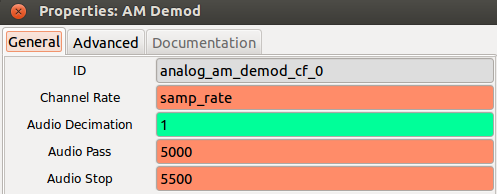
1. De posse desses números, configure o ***Rational Resampler*** conectado ao bloco **Wav File Source** como na figura a seguir.



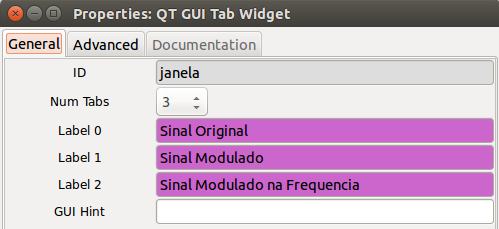
1. Agora podemos entender com certa facilidade que o bloco ***Rational Resampler*** ligado ao bloco ***Audio Sink*** será igual ao bloco que acabamos de configurar, mas com uma inversão de valores, já que queremos sair de 200 kHz para 44,1 kHz (exatamente o inverso). Assim, só precisamos trocar os valores de *interpolation* e *decimation*, como mostra a figura a seguir.



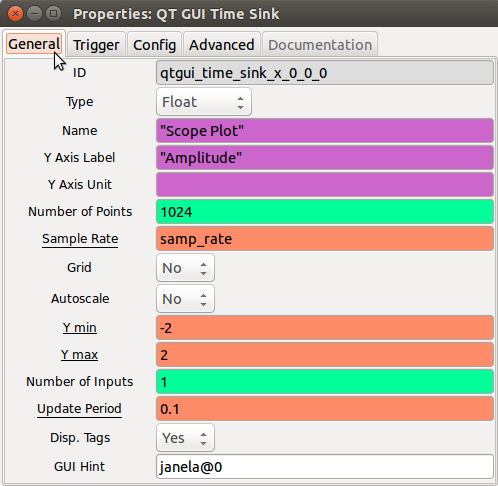
1. Os blocos ***Add****,* ***Multiply****,* ***Throttle****,* ***Audio Sink*** *e* ***Float To Complex*** não necessitam de configurações extras. Basta certificar-se que eles estão com o tipo correto.
2. Iremos agora configurar o bloco ***AM Demod***. Este bloco tem a função de demodular o signal AM gerado pelo projeto. Ele faz operações de modo extrair o sinal em banda base *m*(*t*) do sinal em banda passante *s*(*t*). Em seguida, já com o sinal em banda base, ele usa um filtro passa-baixa para demodular por completo o sinal. Por causa desse filtro passa-baixa, iremos alterar uma configuração no bloco ***AM Demod***. Como nas rádios comerciais, nosso projeto irá apenas demodular até 5 kHz do sinal. Por isso precisamos configurar a variável *Audio Pass* como 5 kHz, para deixar o receptor similar ao rádios AM comerciais. E o *Audio Stop* será exatamente o final do corte do filtro passa-baixa. Nessa variável, iremos deixar em 5,5 kHz. O bloco deve ficar similar a figura a seguir.



1. Vamos agora configurar os blocos da família ***QT GUI****.* Primeiramente, abra as propriedades do bloco ***QT GUI Tab Widget****,* que é responsável por organizarem janelas e abas as saídas dos blocos ***QT*** e preencha os campos como na figura a seguir.



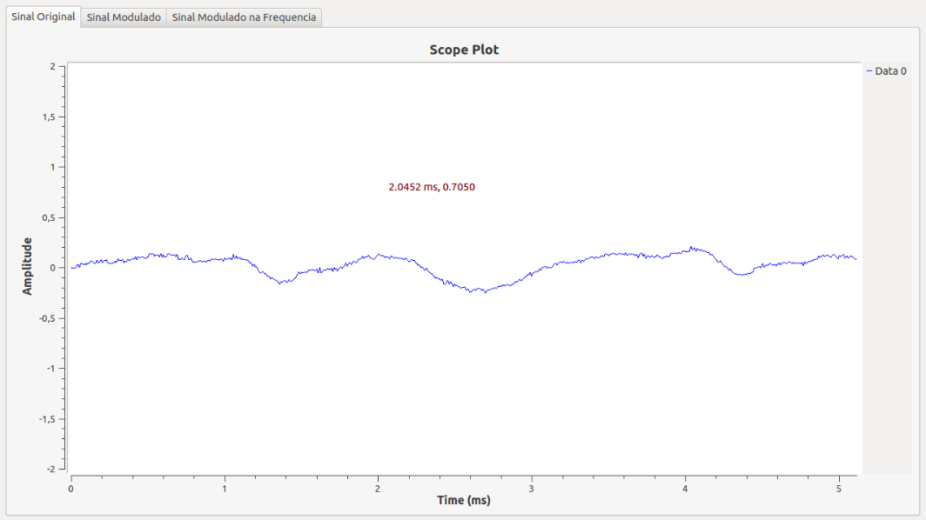
1. Depois, abra o ***QT GUI Time Sink*** conectado ao bloco ***WAV File Source***. Certifique-se de que seus parâmetros ficarão como na figura a seguir. Observe que no campo *GUI Hint* é necessário colocar o ID da janela, previamente definido, e o index da aba, no formato: id\_janela@index\_aba. Como só teremos um gráfico por aba, não será preciso especificar a posição de cada gráfico. Os demais parâmetros foram definidos apenas para facilitar a visualização.



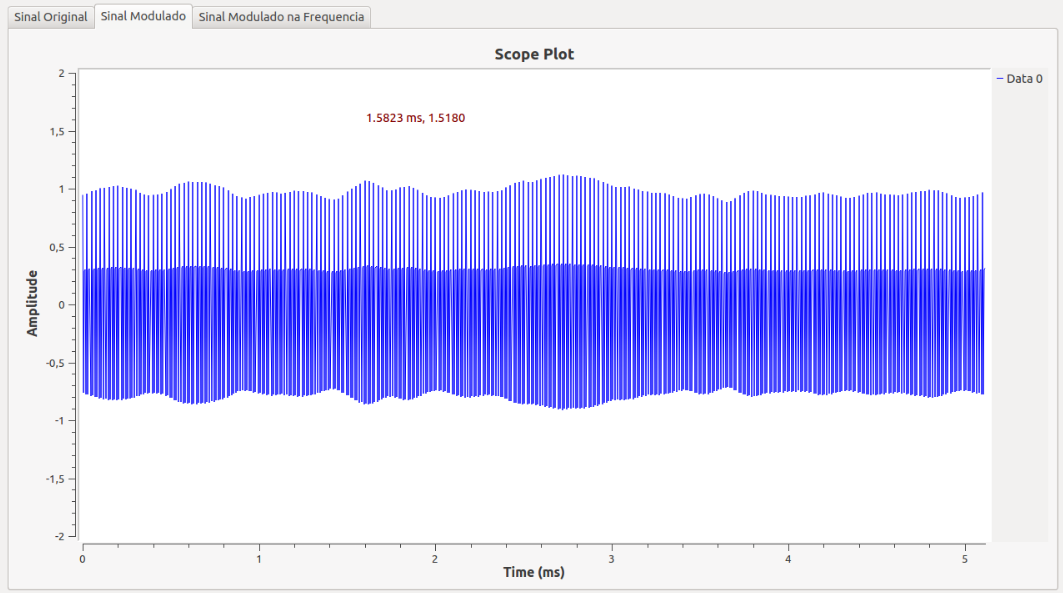
1. Da mesma maneira, preencha as demais janelas QT GUI como será indicado abaixo.

|  |  |
| --- | --- |
|  |  |

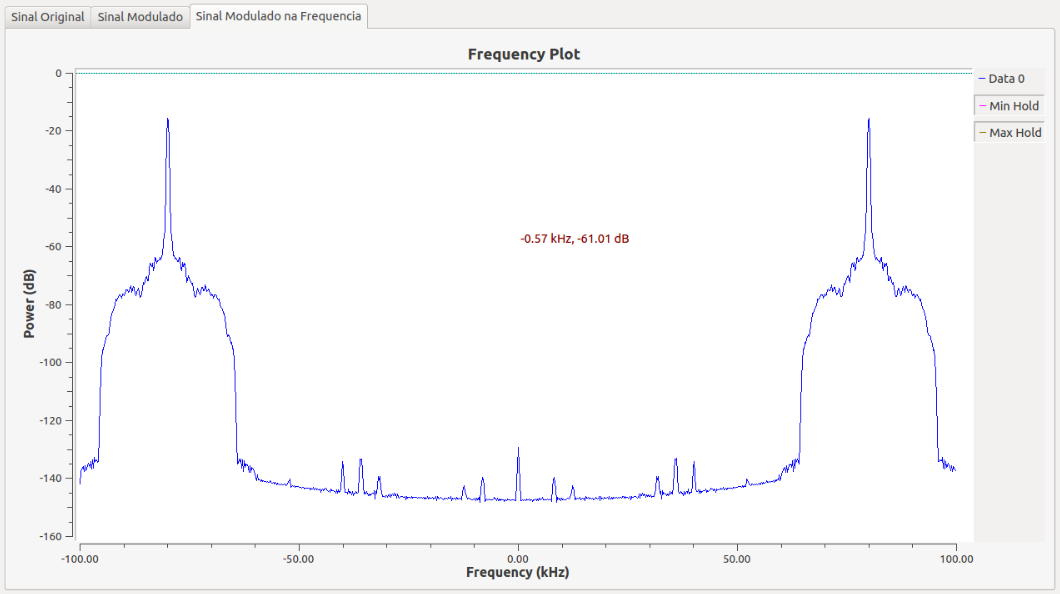
1. Agora com todos os blocos ligados e configurados, certifique-se que os blocos ***QT GUI Frequency Sink*** e o ***QT GUI Time Sink*** estejam operando no modo “Float”. Depois de feito isso, podemos executar o projeto. As saídas do projeto serão similares as da figuras a seguir.



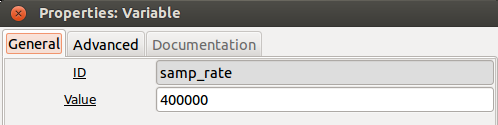
1. **Sinal Original no Tempo**



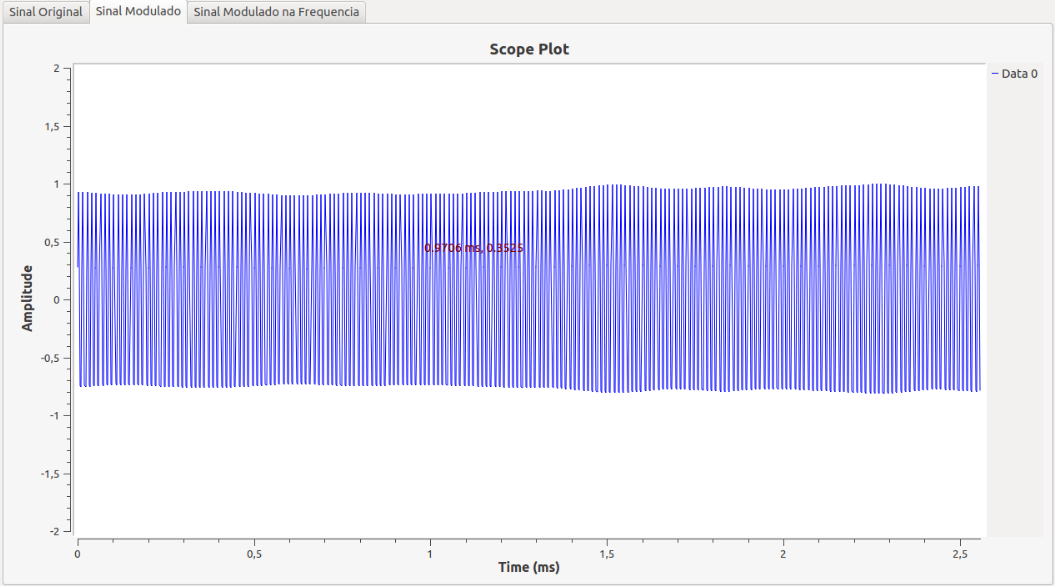
1. **Sinal Modulado no Tempo**



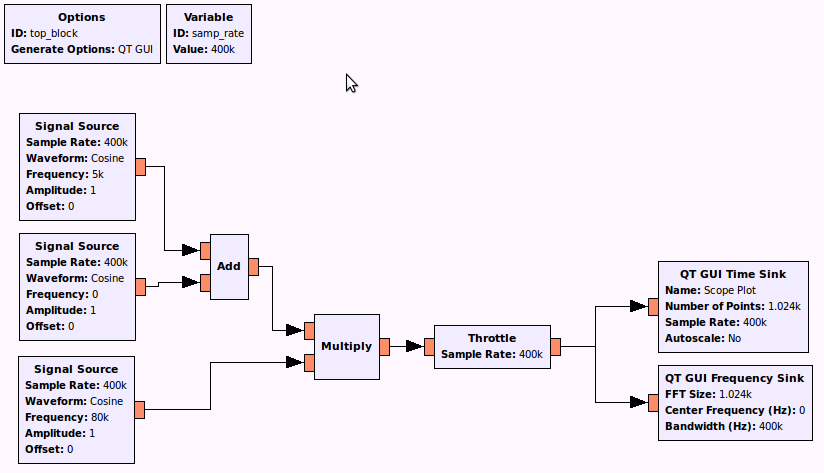
1. **Sinal Modulado na Frequência**
2. Observe que o sinal modulado no tempo, apresenta as mesmas variações de amplitude que o sinal original, como é característica das modulações AM. Podemos perceber também que o sinal modulado no tempo está apresentando falhas em sua apresentação, apesar de a taxa de amostragem ser superior a de Nyquist por uma boa margem. Isso se dá provavelmente devido a um método bastante fraco de recuperação do sinal. Aumente o sample\_rate para 400 KHz, 5 vezes a frequência da portadora) e veja como a exibição do sinal modulado melhora consideravelmente. **OBS:** Não se esqueça de alterar o *interpolation* do primeiro ***Rational Resampler*** e o *decimation* do segundo para 4000, pelas razões já expostas acima.



1. Feito isso, o resultado do seu gráfico deve ficar semelhante à figura abaixo.



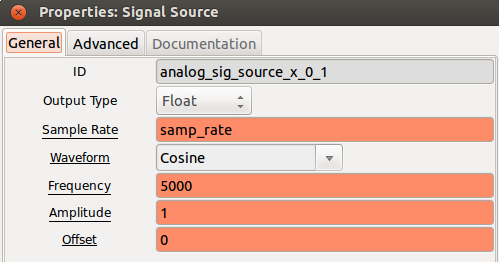
1. Tendo feito todos os passos o programa deve funcionar e deve-se ouvir o áudio demodulado, mas com uma qualidade inferior, já que ele é demodulado limitado a 5KHz do áudio original. Se tiver curiosidade altere o campo *Audio Stop* do ***AM Demod*** e veja como isso altera a qualidade do áudio. Quanto maior a diferença entre o *Audio Pass* e o *Audio Stop,* menor será a ordem do filtro aplicado, exigindo, portanto, menos processamento do computador.
2. Para melhor visualizarmos o sinal modulado, vamos trocar o bloco ***Wav File Source*** por um novo bloco ***Signal Source.***  Assim poderemos ver as duas bandas laterais perfeitamente, pois estaremos transmitindo um tom em vez de um sinal de áudio complexo (música do arquivo .wav).
3. Salve seu projeto, usando o recurso “*save as*”, com o nome ***am\_loopback\_vicente\_step\_2.grc.*** Adicione mais um bloco ***Signal Source*** e o substitua no lugar dos blocos **Wav File Source** e ***Rational Resampler***. Delete os blocos responsáveis pela demodulação, o primeiro ***QT GUI Time Sink***, e o ***QT GUI Tab Widget***. O projeto ficará como na figura a seguir.



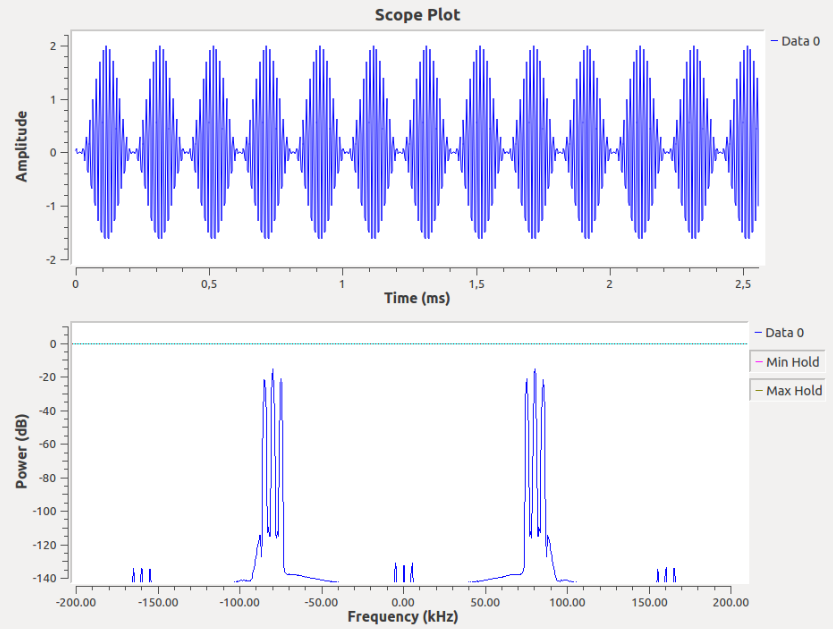
1. Configure o ***QT GUI Time Sink*** e o ***QT GUI Frequency Sink*** como indicado nas figuras abaixo para facilitar a visualização e remover as configurações de janela.

|  |  |
| --- | --- |
|  |  |

1. Configure o novo ***Signal Source*** para gerar um sinal de 5 kHz. O bloco deve ficar com a configuração similar ao da figura a seguir.



1. Agora compile e execute o projeto. Dado que a entrada do sistema é apenas um sinal senoidal, podemos ver claramente o sinal modulado, com sua portadora e bandas laterais.



25. Agora mude a amplitude do sinal para avaliar os casos de sobremodulação e a supermodulação AM. Faça um relatório incluindo como você gerou cada caso. Mostre e disserte sobre os gráficos de amplitude no tempo e potência na frequência.

1. **Fonte**: http://www.cic.unb.br/~lamar/te060/Apostila/Capitulo2.pdf [↑](#footnote-ref-1)