Hands-on 8

Transmissor/Receptor WBFM em 2,4GHz (GRC)

**Introdução Teórica**

A teoria sobre FM para este exercício pode ser encontrada no hands-on 6.

**Amostragem de sinais em banda-base**

Sinais (em banda base ou banda passante) são amostrados usando a taxa de amostragem determinada pela teoria de Nyquist. Em 1928, ele provou que

*“Seja um sinal s*(*t*) *limitado em frequência, tal que S*(*f*) = 0 *para w* >*wm, então s*(*t*) *é unicamente determinado por suas amostras s*[*NT*]*, com n*=0,±1,±2,...,*se:*

com

O parâmetro *T*s é conhecimento como tempo de amostragem (seu inverso com taxa de amostragem), i.e., o tempo no qual se deve colher amostras igualmente espaçadas do sinal.

Assim, de acordo com o Teorema de Nyquist, a quantidade de amostras por unidade de tempo de um sinal, chamada taxa ou frequência de amostragem, deve ser maior que o dobro da maior frequência contida no sinal a ser amostrado, para que o mesmo possa ser reproduzido integralmente sem erro.

O erro decorrente de uma amostragem com taxa menor que a de Nyquist é chamado de aliasing (mascaramento). As figuras abaixo ilustram esse efeito.

Suponha um sinal *s*(*t*) limitado em frequência como ilustrado na figura a seguir.

|  |  |
| --- | --- |
| **Sinal s(t) no domínio do tempo** | **Sinal s(t) no domínio da frequência** |

Ao amostrar o sinal *s*(t), seu espectro é replicado como ilustrado na figura a seguir.

|  |  |
| --- | --- |
| **Sinal s(t) amostrado no domínio do tempo** | **Sinal s(t) amostrado no domínio da frequência** |

Assim, o aliasing acorre devido a sobreposição espectral das réplicas do sinal amostrado, caso a frequência de amostragem não for escolhida apropriadamente, como ilustrado na figura a seguir.



Contudo, isso é perfeito para sinais em banda-base que tem espectro que se estende a algumas centenas que kHz. Considerando o processamento em SDR e de sinais em banda-passante com frequência de algumas centenas de MHz, o processamento do sinal amostrado utilizando a taxa de Nyquist (duas vezes a maior frequência do sinal) pode causar problemas de processamento e/ou gasto excessivo de bateria de terminais portáteis. Na verdade, amostrar sinais de 2,4 GHz utilizando a taxa de Nyquist já inviabiliza experimentos usando a USRP, até mesmo devido a limitações do próprio hardware. Dessa forma, lançamos mão da teoria de amostragem de sinais em banda-passante para evitar a aliasing, mesmo amostrando sinais de alta frequência a uma taxa de amostragem menor que a de Nyquist.

**Amostragem de Sinais em Banda Passante**

Uma Alternativa para realizar a amostragem com taxa menor que a de Nyquist é incluir múltiplos estágios de translação de frequência, levando a um gasto indesejado com hardware entre a antena e o ADC. Outra alternativa é a utilização de amostragem em banda-passante (*bandpass sampling*). Amostragem em banda-passante é um a forma especial de sub-amostragem que translada um sinal em banda-passante de alta frequência (ou um sinal com múltiplas bandas) em um sinal em banda-base. A taxa de amostragem necessária passa a depender mais da banda do sinal do que da sua frequência máxima.

Da teoria da amostragem de sinais em banda-passante, um sinal em alta frequência e limitado em banda pode ser exatamente reconstruído se a taxa de amostragem satisfaz as seguintes condições:

1. O limite inferior de frequência (*f*L) do sinal em banda-passante é menor que um múltiplo da metade do valor da taxa de amostragem, i.e.,
2. Limite superior de frequência (*f*H) do sinal em banda-passante é menor que um múltiplo da metade do valor da taxa de amostragem, i.e.,

Rearranjando as duas condições acima, temos:

sendo:

* *f*Lo limite inferior de frequência do sinal em banda-passante;
* *f*Ho limite superior de frequência do sinal em banda-passante;
* *n* um inteiro entre 1 e .
* *BW* é a banda do sinal em banda-passante (*f*H-*f*L)

Tais parâmetros são ilustrados na figura a seguir.



Assim, o maior valor para *n* é:

As duas expressões podem ser expressas graficamente, normalizando a frequência de amostragem e a maior componente espectral do sinal em questão, gerando o gráfico a seguir. Ele mostra que dependo do valor de *n* (menor a taxa de amostragem), menor é a região propícia a evitar o aliasing. Isso é facilmente observado pela expressão de *f*s. Quanto maior o valor de *n*, menor é a diferença entre os limites inferiores e superiores de *f*s.



Alguns autores propuseram algoritmos para calcular os valores ótimos de frequência de amostragem de sinais em banda-passante. Os algoritmos consistem basicamente na resolução das equações que regem a teoria da amostragem banda-passante, mas a degradação da SNR é um fator importante que merece ser levado em conta.Uma solução é estabelecer um nível de degradação, e a partir daí calcular a frequência de amostragem que satisfaça as condições.

**Referências**

[1] http://www.snowymtn.ca/GNURadio/GNURAdioDoc-7.pdf - acesso em:04/12/2012

[2]A. Latiri. *Architecture et conception de récepteur reconﬁgurable à échantillonnage RF pour les applications multistandard*. PhD thesis, Telecom ParisTech, July 2008.

[3] R.G. Vaughan, N.L. Scott, and D.R. White.*The theory of bandpass sampling*.SignalProcessing, IEEE Transactions39(9):1973–1984, Sept. 1991.

[4] Angrisani, L. D’Arco, M. Schiano Lo Moriello, R. & Vadursi, M. (2004). *Optimal sampling strategies for bandpass measurement signals*, Proc. of the IMEKO TC-4 Interational Symposium on Measurements for Research and Industry Applications*.* pp. 343-348, September 2004.

[5] J. Bae, J. Park. *An efficient algorithm for bandpass sampling of multiple RF signals.* SignalProcessing, IEEE Letters13(4):193- 196, April 2006.

**Exercício**

**OBJETIVO:** Através do uso de conceitos sobre modulação em frequência (FM) e da teoria de modulação em banda-passante, iremos criar um transmissor FM capaz de transmitir em 2,4 GHz. Para tal precisaremos usar blocos de sub- e sobre-amostragem.

1. Caso ainda não esteja aberto, inicialize o GNU Radio Companion.
   1. Abra um terminal digitando CRTL+ALT+t
   2. Digite: gnuradio-companion e pressione ENTER

|  |
| --- |
|  |

Alternativa:

1. Clique em Dash Home

|  |
| --- |
| fig_02-1.png |

1. Digite gnuradio e clique no ícone correspondente ao GRC

|  |
| --- |
|  |

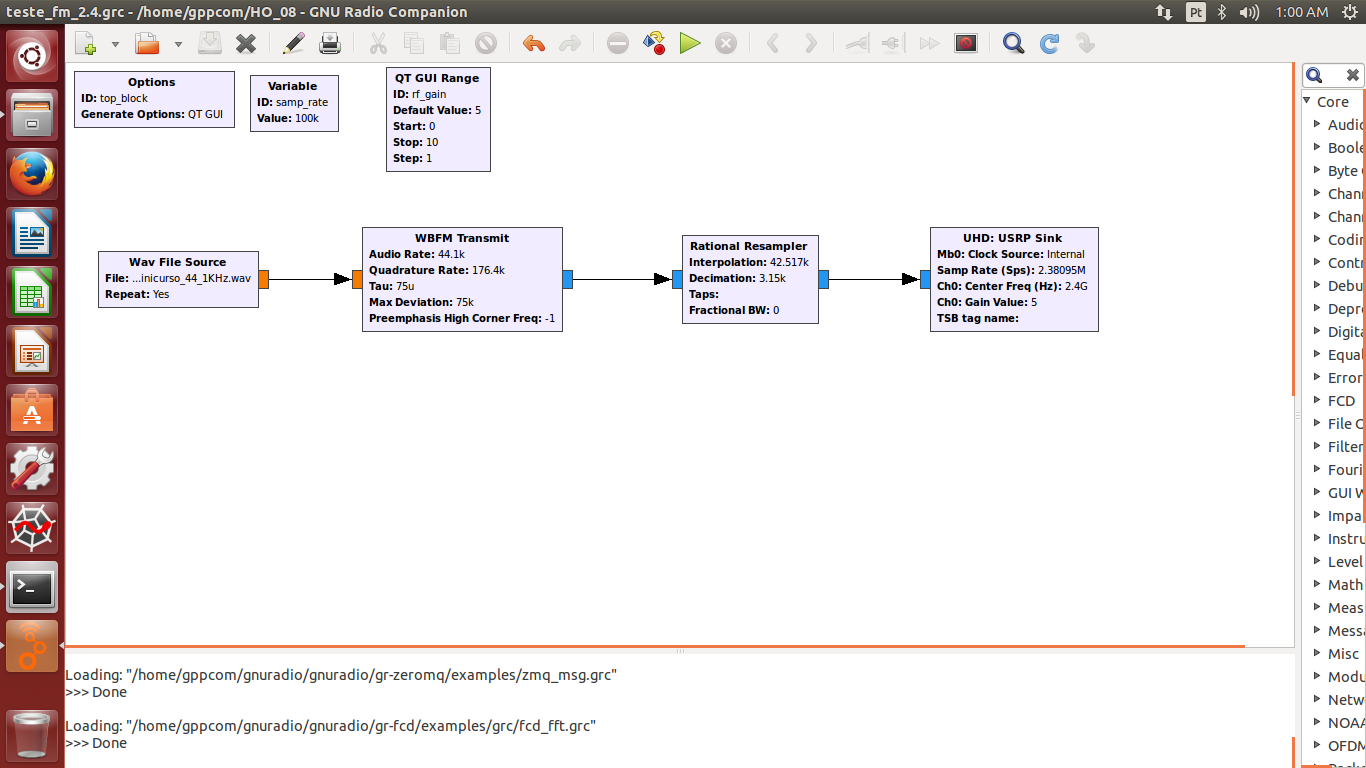
1. Com o GNU RADIO COMPANION aberto, crie um novo projeto

|  |
| --- |
| fig04.png |

1. Clique duas vezes no Bloco **Options**. Esse bloco configura alguns parâmetros gerais de flowgraph. Mantenha o ID como top\_block. Digite um título para o projeto e um autor. Selecione *GenerateOptions* com QT GUI, *Run* para Autostart e *Realtime Scheduling* para Off. Então, feche a janela de propriedades.

|  |
| --- |
|  |

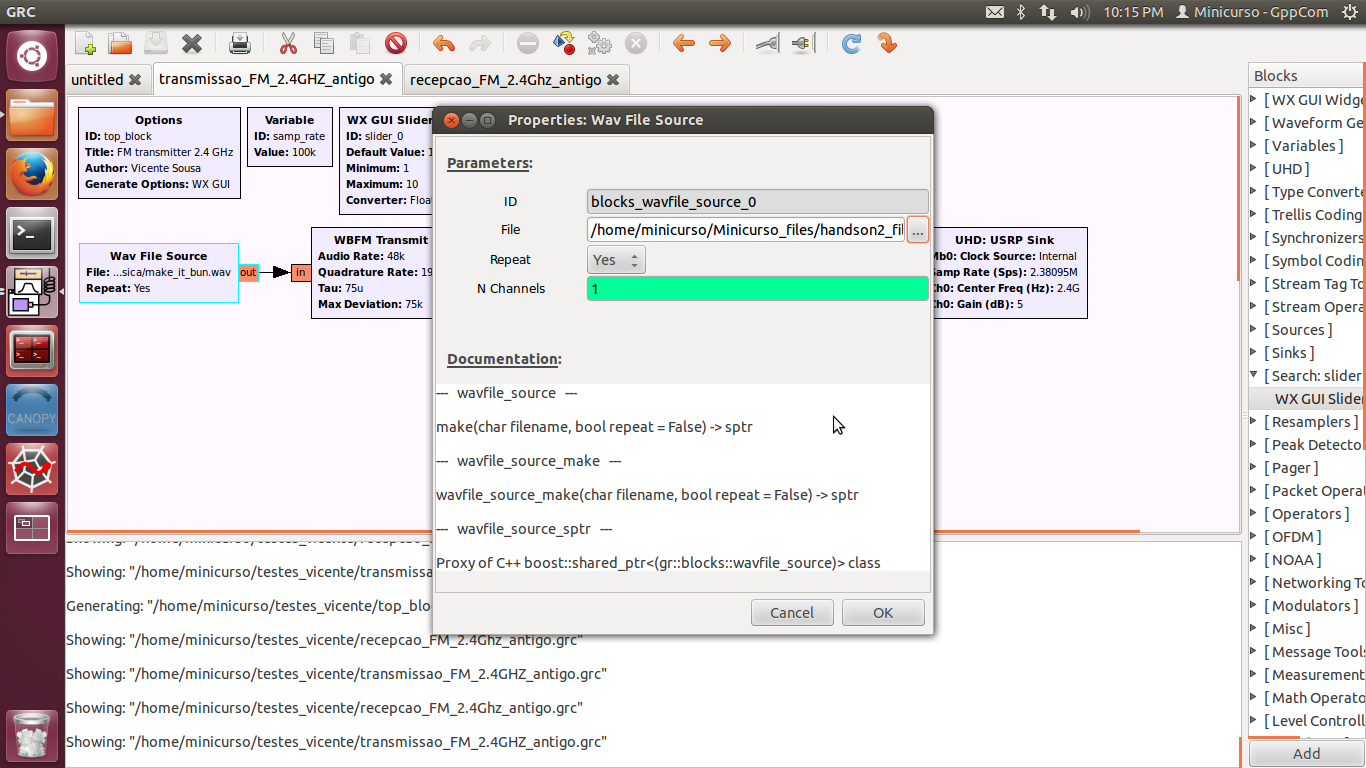
1. Primeiro iremos construir o transmissor, para só depois construirmos o receptor. Construa um projeto utilizando os blocos ***Wav File Source****,* ***WBFM Transmit, Rational Resampler, QT GUI RANGE*** e ***UHD: USRP Sink***. Conecte os elementos de forma que sua área de trabalho fique igual à figura a seguir. Altere o campo *Type* para *Complex* apenas do bloco ***Rational Resampler*** que estará ligado à saída do bloco ***WBFM Transmit***.



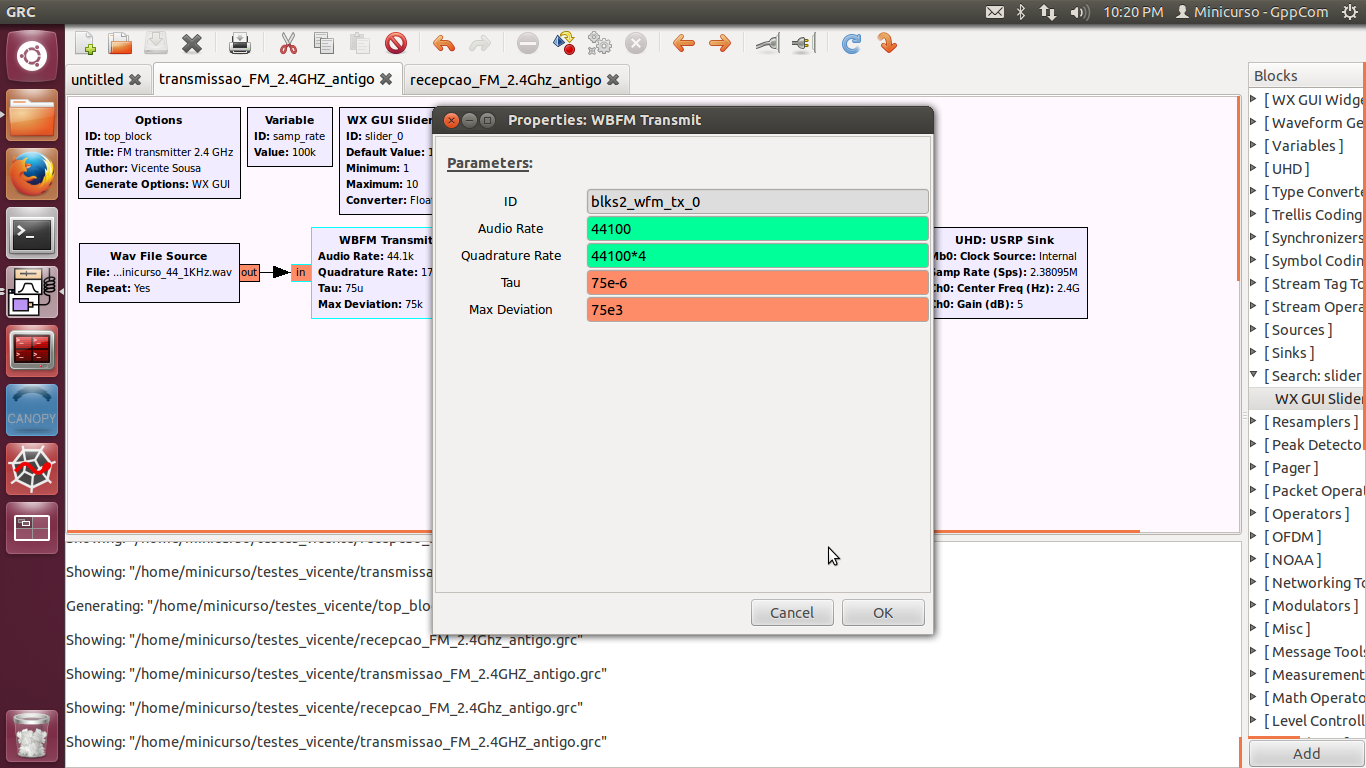
1. Configure o bloco ***Variable*** com o valor de *samp\_rate* para 100000 amostras. Esse valor foi escolhido como razoável para evitar grande complexidade e permitir que o experimento funcione na maioria dos computadores. Também é um valor bom o bastante para que todos os filtros possam ser realizáveis. O bloco deve ficar como na figura a seguir.

|  |
| --- |
|  |

1. Configure o bloco ***Wav File Source*** com um arquivo **wav** para transmitir, procure um arquivo com taxa de 44,1 kHz (padrão na maioria das placas de som). O bloco deve ficar como na figura a seguir.



1. Agora vamos configurar o bloco ***WBFM Transmit***. Altere o campo *Audio Rate* para taxa de amostragem do arquivo de som, que por padrão deve ser 44100. O campo *Quadrature Rate* está diretamente ligado ao campo *Audio Rate, no qual* só é permitido colocar valores múltiplos do valor que está no campo *Audio Rate*. No nosso caso colocaremos 176400 que é um valor 4 vezes maior que 44100. Se por acaso você colocar um valor muito grande na variável *Quadrature Rate* poderá dar problemas, pois esse bloco usa filtros de *Chebyshov* e usa o algoritmo de Remez para concebê-los. Assim, se o valor for muito grande, a aproximação de Remez gerará um filtro com muitos estágios e de muita complexidade para o computador. O campo *Tau* e *Max Deviation* deixaremos o valor padrão. O *Tau* remete ao valor da constante de tempo de um circuito RC, enquanto o *Max Deviation* é a frequência máxima de desvio determinada pela Agência reguladora (ANATEL) que é de75 kHz. O bloco deve ficar como na figura a seguir.



1. Dê dois cliques no bloco ***UHD: USRP Sink*** e altere o campo *Ch0:Center Freq* para a frequência que deseja alocar a transmissão. No nosso caso, optaremos pela frequência de 2,4GHz. Agora altere o campo *Ch0:Gain(dB)* que é o valor de ganho da antena para *rf\_gain*. Finalmente, alteraremos o campo *Samp Rate (Sps)* para o valor da taxa de amostragem que encontraremos seguindo o teorema da amostragem em banda-passante. Os próximos passos mostram como fazer esse cálculo.
   1. O sinal em banda-passante, com sua maior frequência sendo e sua menor frequência , pode ser completamente reconstruído (sem aliasing) se a taxa de amostragem satisfizer as seguintes condições:

e

com *n* inteiro entre 1 e . Isolando , temos:

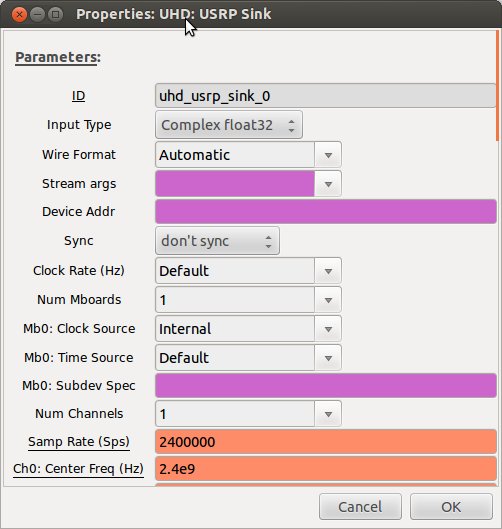
* 1. Então, o maior valor para *n* é
  2. Para o nosso caso, temos que a frequência da portadora é 2,4GHz, o valor de *β* (índice de modulação) é 5 e o sinal de entrada será limitado em 15 kHz (*W*m). Portanto, o desvio máximo de frequência é de *Δ*f = 75 KHz (similar ao estabelecido pela ANATEL para rádio FM comercial).
  3. E sabemos, pela regra de Carson, que a banda do sinal FM é BW = 2(*W*m+*Δ*f) = 2 (75+15) = 180 kHz.
  4. Assim, temos que *H*= 2400000000 + 90000 Hz. Da mesma forma que *L* = 2400000000-90000 Hz. Assim temos:

*n*max

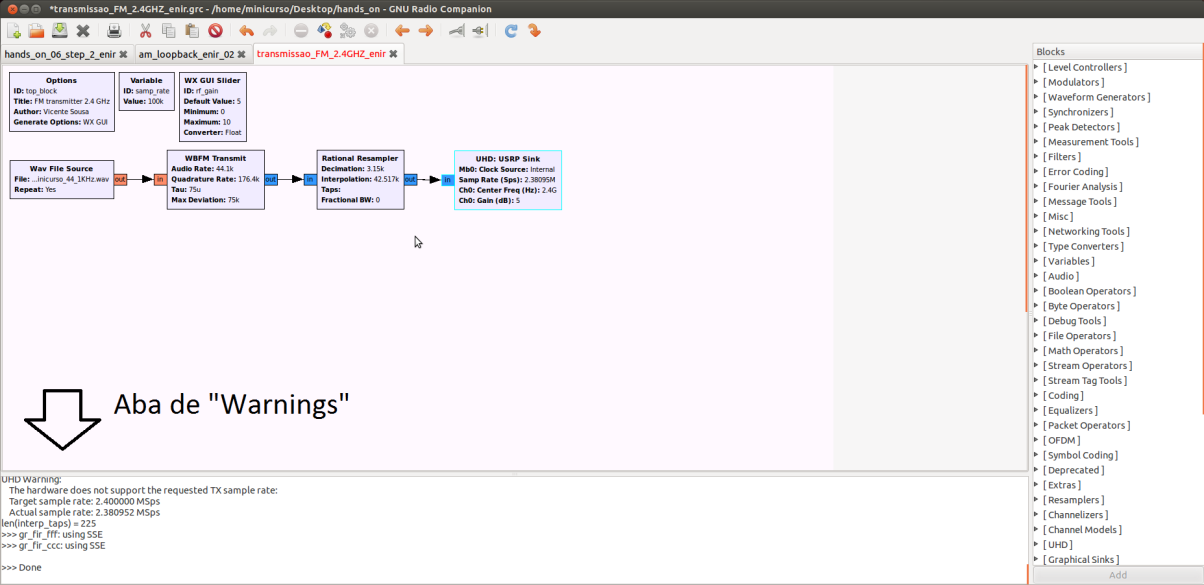
* 1. Assim, podemos verificar que é possível amostrar um sinal de 2,4GHz com a taxa de amostragem de 360 kHz. O que é incrível, dado as dimensões!!! Contudo, o cálculo acima mostra que a taxa de amostragem só pode variar de a Hz, o que nos dá uma variação muito pequena de 0,0017 Hz!!! Muito fácil de ficar fora da margem de precisão de um equipamento SDR.
  2. Então, em vez de usarmos o *n* igual a *n*max (i.e. 13333), iremos usar 2000 como valor para *n* (escolhido arbitrariamente). Assim teremos uma taxa de amostragem mais alta, porém, teremos uma taxa que admite maior variação.
  3. Refazendo as contas para *n* igual a 2000, temos que:

a taxa de amostragem pode variar de a . Dando assim uma margem de variação da taxa de amostragem de 1020 Hz. Então, usaremos a taxa de amostragem de 2,4 Msps, que é muito próximo à frequência mínima de amostragem. Como veremos mais a frente, fatalmente teremos que mudar esse valor para um mais próximo que a USRP aceita.

* 1. A configuração final do bloco ficará como mostrado na figura a seguir.



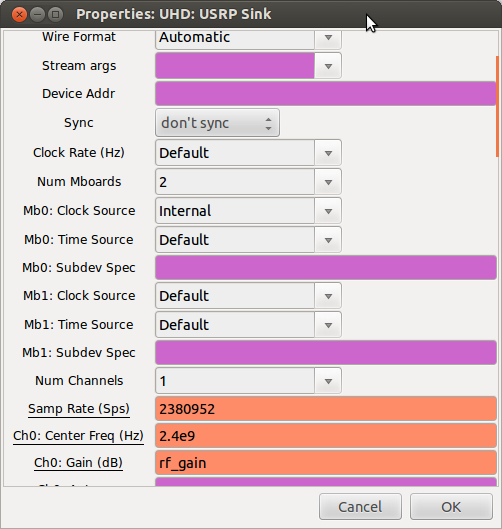
1. Contudo, isso não é tudo. Ao executar o projeto, receberemos a seguinte mensagem de warning.



1. Isso significa que, mesmo que configuremos a taxa de amostragem de 2,4 Msps, o hardware USRP não consegue fornecer exatamente essa taxa. Isso depende de fatores construtivos de cada SDR, e no caso da nossa USRP N210, a taxa disponível mais próxima de 2,4 Msps é 2,380952 Msps. Verifique que em “*Target sample rate*” estará o valor que escolhemos como *samp\_rate* no bloco ***UHD USRP SINK***. O “*Actual sample rate*” é o valor mais próximo que o USRP chega do valor configurado no parâmetro *samp rate*. Assim, esse número é importante para configurar tanto o bloco ***UHD: USRP Sink*** quanto o bloco ***Rational Resampler***. O bloco *UHD: USRP Sink* deve ficar como na figura a seguir.
2. Contudo esse valor novo da taxa de amostragem pode ser testado para se concluir se pertence a alguma faixa de valores aceitáveis de *f*s, isto é, o valor de *n* não ultrapasse *n*max. Assim, temos:

APROXIMADAMENTE:

Assim podemos concluir que esse valor da taxa de amostragem está contido em uma faixa de valores possíveis. Que nesse caso é para um *n* = 2017.

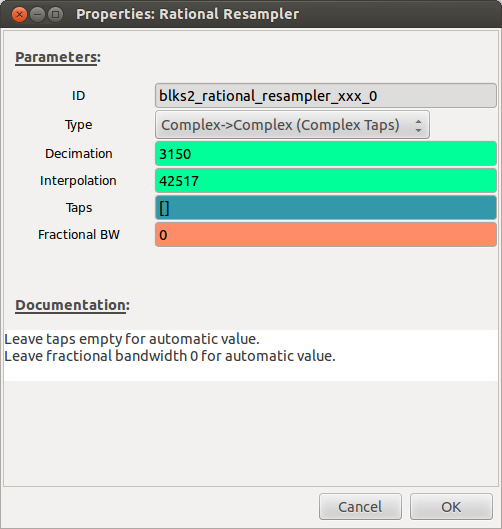


1. Agora chegou a hora de configurarmos o bloco ***Rational Resampler****.* Como já dito em hands-on anteriores é bem simples o cálculo. Na verdade, é uma pequena divisão entre a taxa de amostragem que queremos (2.380952 GHz) e a que temos (176.4 kHz). Senão quiser calcular pode inserir os valores já calculados abaixo. Veja na figura abaixo com o cálculo foi realizado usando o programa *Fractional Calulator* do Android.

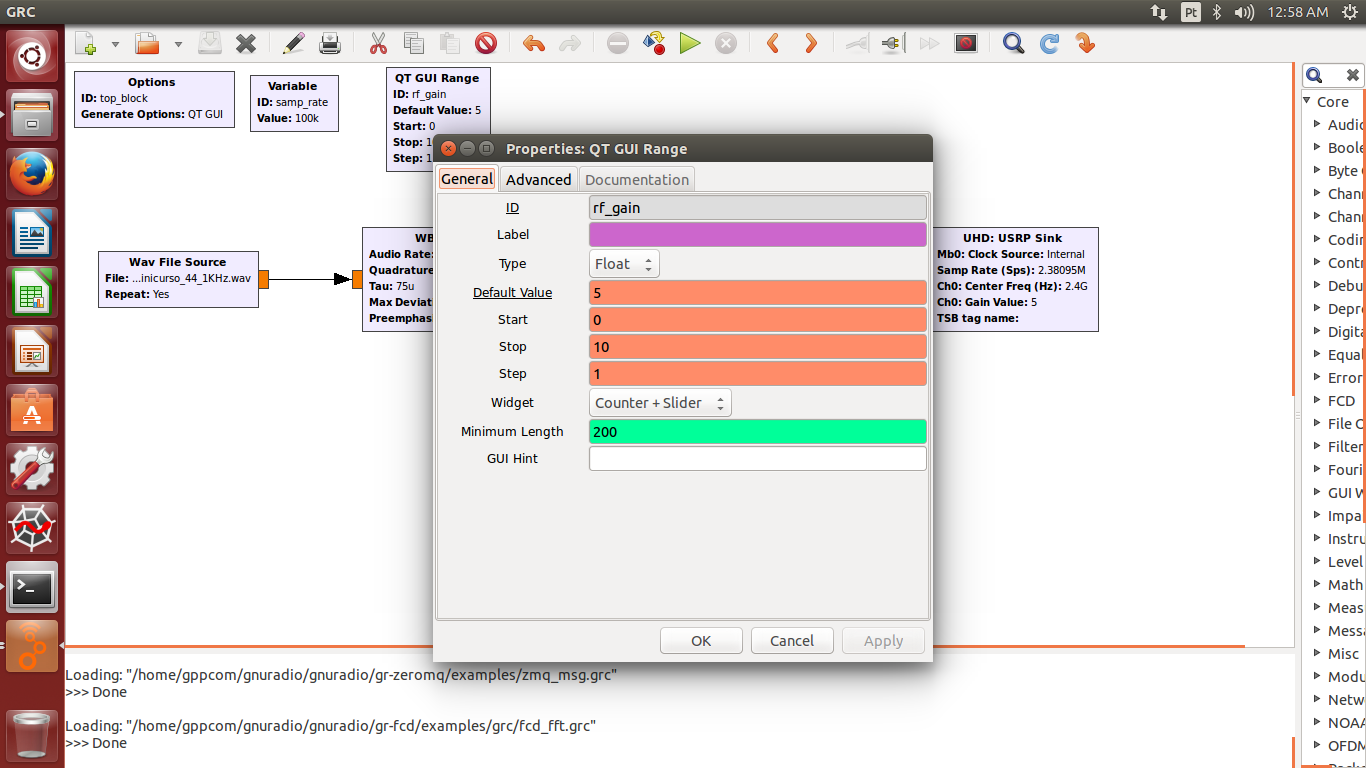


**Obs.:** todo o projeto foi baseado em um arquivo .wav com taxa de amostragem de 44,1KHz. Se o arquivo estiver com amostragem diferente, você pode usar um bloco ***Rational Resampler***entre o bloco ***Wav File Source*** e o bloco ***WBFM Transmit****.*

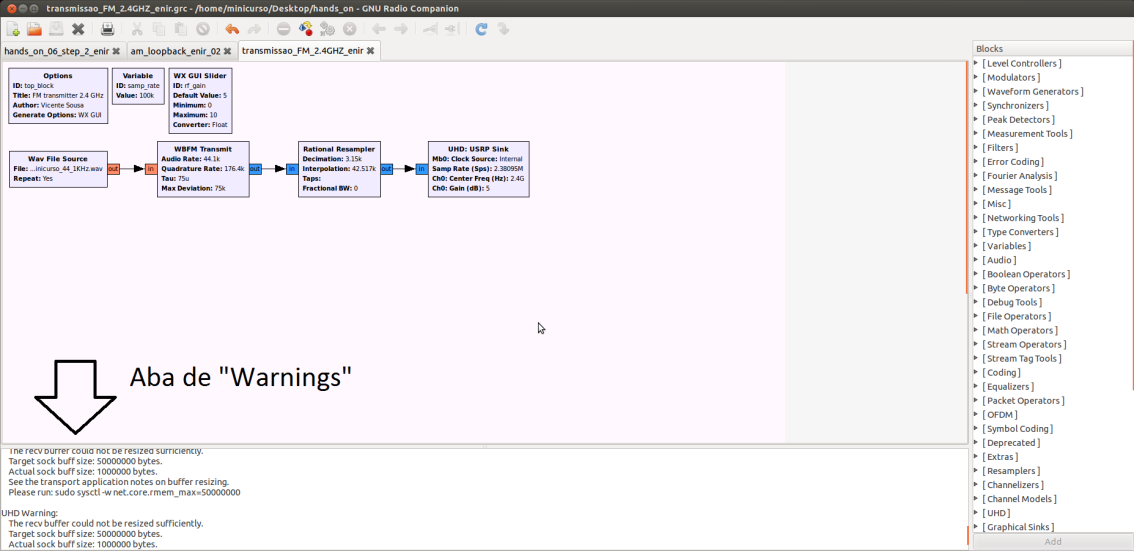
1. O bloco ***Rational Resampler***deve ficar configurado como na figura a seguir.



1. Configure o bloco ***QT GUI Range*** para controlar o valor do ganho de RF no bloco ***UHD: USRP Sink*** (já configurado). Altere o *ID* do bloco ***QT GUI Range*** para *rf\_gain*. Altere o campo *Default Value* para 5, mude o valor dos campos *Start*, *Stop* e *step* para 0, 10 e 1, respectivamente. O bloco deve ficar configurado como na figura a seguir.



1. Quando a execução do projeto for feita, você pode verificar alguns problemas devido a parâmetros que o GNU Radio não tem permissão para mudar sozinho. Para isso iremos mudar os valores manualmente. Quando você executar o projeto aparecerá na aba de “*warnings*” uma mensagem que diz que o computador não conseguiu mudar os valores de alguns buffers. Mas felizmente, o computador já fornece o comando pronto para ser executado. A figura a seguir mostra um exemplo desse tipo de warning.

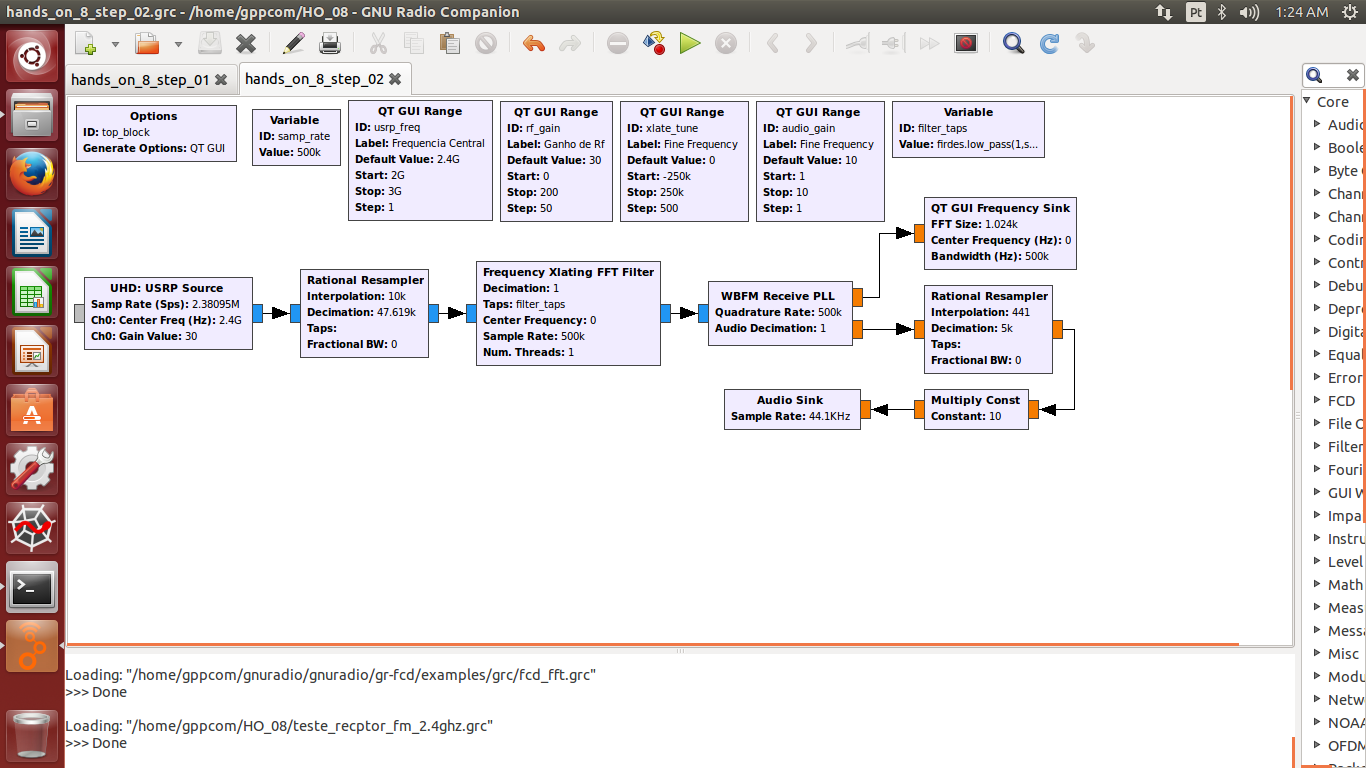


A mensagem é repetida abaixo caso a figura não seja legível.

UHD Warning:

The recv buffer could not be resized sufficiently.  
 Target sock buff size: 50000000 bytes.  
 Actual sock buff size: 1000000 bytes.  
 See the transport application notes on buffer resizing.  
***Please run: sudosysctl -w net.core.rmem\_max=5000000***

1. Agora abra um Terminal, copie e cole exatamente o comando que foi indicado no warning. Ele pedirá a senha do PC (no caso da nossa máquina virtual, é “gppcom”. Execute todos os comandos que foram indicados na aba warning. E ao termino o projeto estará pronto para ser executado sem nenhum problema.
2. Agora podemos executar o projeto, e o mesmo deverá funcionar perfeitamente. Uma rádio FM está transmitindo na frequência de 2,4 MHz.
3. Para testar e ouvir o que o transmissor está irradiando, iniciaremos agora o projeto do receptor. Configure um novo *flowgraph* com os seguintes blocos: 3 blocos ***QT GUI Range,*** um ***UHD USRP Source,*** dois ***Rational Resampler,*** um ***Frequency Xlating FIR Filter,*** um ***WBFM Receiver,*** um ***Variable,*** um ***QT GUI Frequency Sink,*** um ***Multiply Const,*** um ***Audio Sink****.* Disponha os blocos de forma similar a figura a seguir.



1. Em ordem vamos configurar os blocos slides do projeto. São 3 blocos responsáveis por parâmetros ajustáveis do receptor. Configure os blocos como nas figuras a seguir.

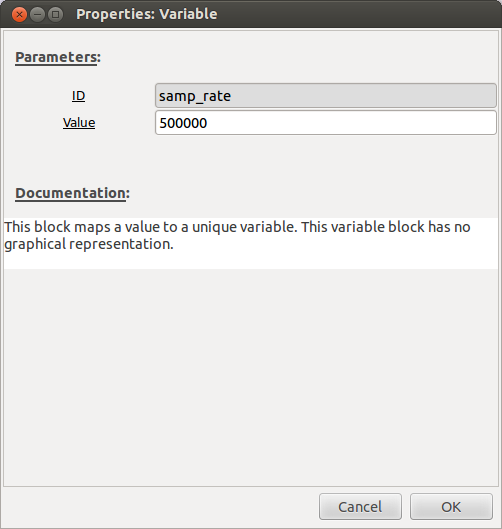
|  |  |
| --- | --- |
|  |  |

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

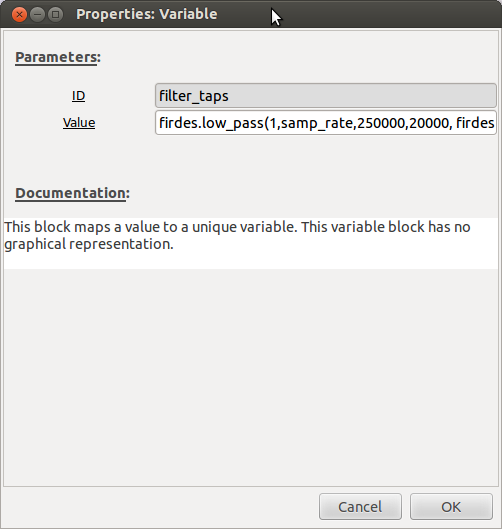
1. Configurados, agora eles podem substituir valores em outros blocos. Configuraremos agora o bloco ***UHD USRP Source***, o qual é responsável pela entrada de sinal no receptor. Ele deverá ter a mesma taxa de amostragem (*Samp Rate*) que a do transmissor. Logo sua configuração é semelhante a do transmissor. Mas agora em vez de usar valores exatos, usaremos os blocos slides para podermos variar esses valores. Configure o bloco como na figura a seguir. Pode-se perceber que nos blocos slides estão os valores do experimento do transmissor.

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

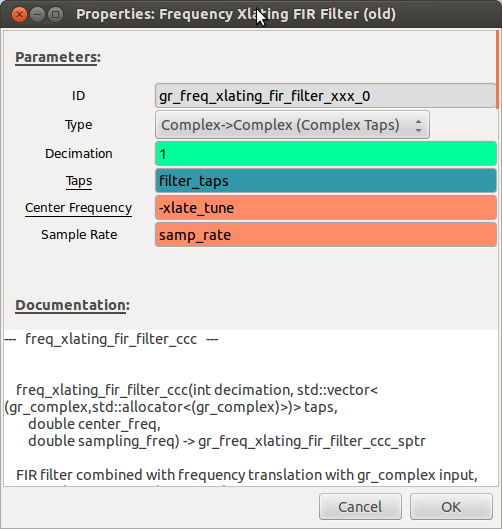
1. Agora iremos configurar o bloco ***Variable*** que contém a variável *samp\_rate*. Vamos usar um valor de 500000 amostras por segundo. Valor escolhido com base nos filtros que serão configurados mais a frente. Um grande exemplo disso é o bloco ***Frequency Xlating FIR Filter*** que necessita de filtros para que possa transladar o sinal de banda-passante para banda-base. Configure o bloco ***Variable*** como na figura a seguir.



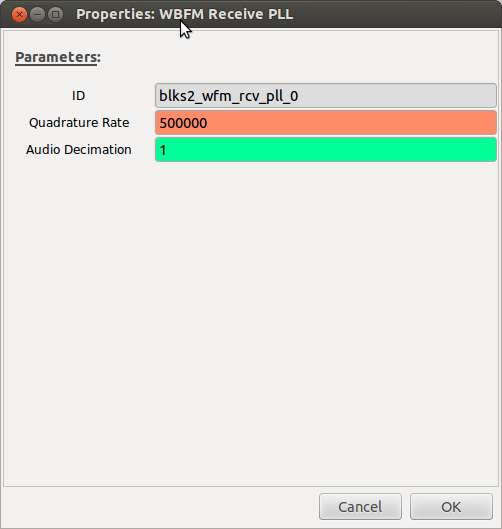
1. Iremos agora configurar o outro bloco ***Variable*** que adicionamos ao projeto. Esse bloco guardará o valor da variável *taps* que iremos usar no bloco ***Frequency Xlating FIR Filter****.* Essa variável é responsável por todos os parâmetros importantes do bloco, tais como: largura de banda, frequência de corte, numero de “taps” e algumas outras variáveis. Configure o campo *value* do bloco com   
   firdes.low\_pass(1,samp\_rate,250000,20000, firdes.WIN\_HAMMING, 6.76). O bloco ficará como na figura a seguir.



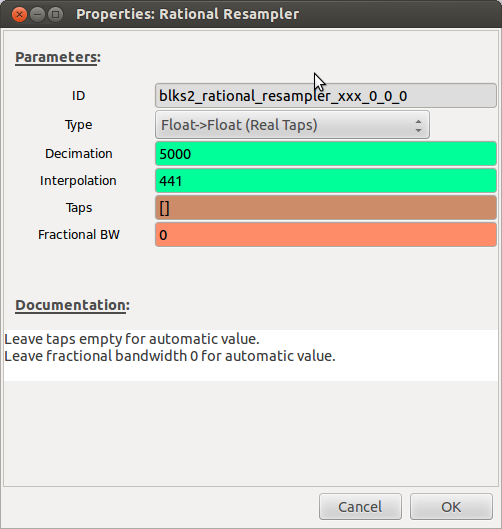
1. Agora podemos configurar o bloco ***Frequency Xlating FIR Filter****.* Configure o bloco como na figura a seguir. Configurar o campo *Taps* com *filter\_taps*, o campo *Center Frequency* para *–xlate\_tune* e o campo *Sample Rate* para *samp\_rate*.



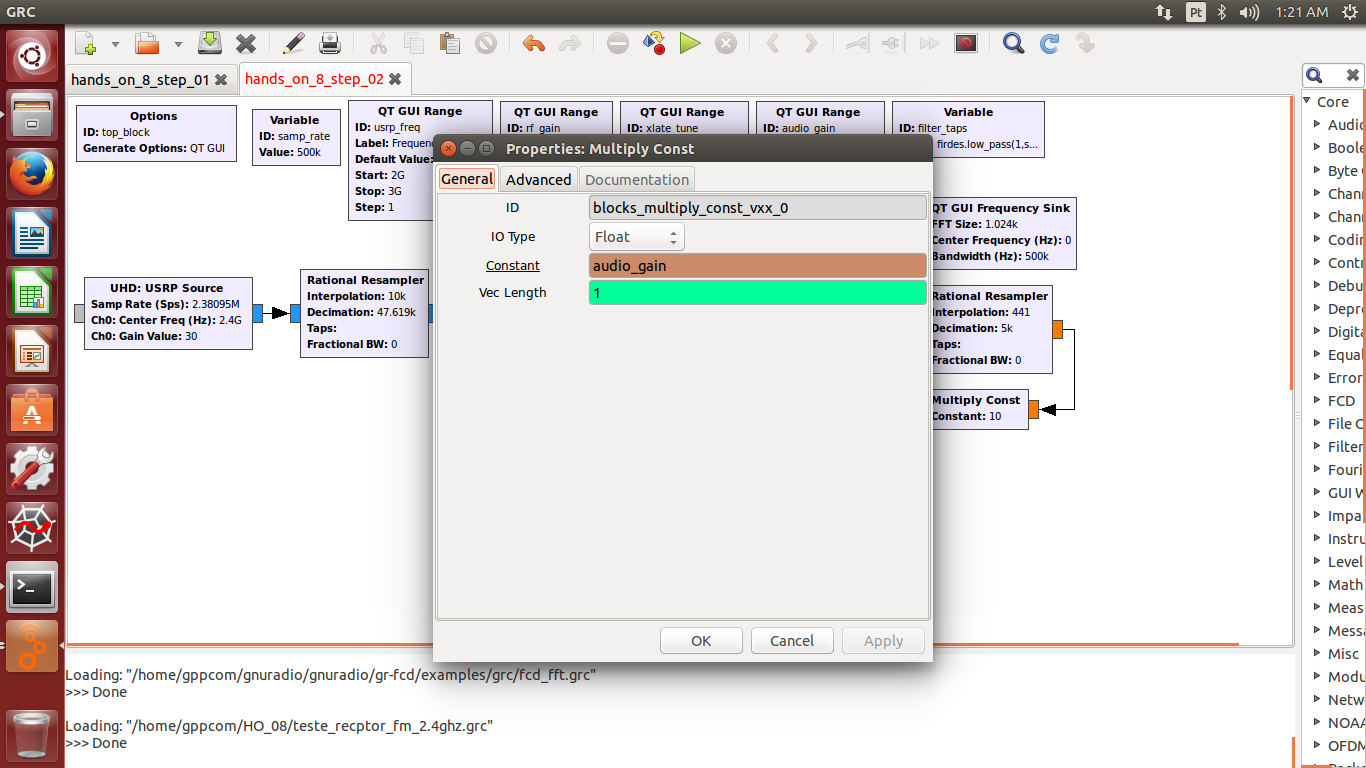
1. Agora vamos configurar o bloco ***WBFM Receiver.*** A variável *Quadrature rate* diz respeito a taxa de amostragem do sinal de entrada. *Audio Decimation* diz respeito a decimação que o sinal de saída do bloco sofrerá, podendo ser 1 quando não se deseja decimação alguma. No nosso caso, deixaremos a decimação em 1, pois usaremos um bloco ***Rational Resampler***para mudar a taxa de amostragem de 500 kHz para 44,1 kHz (taxa de amostragem da placa de som do computador). O bloco deve ficar como na figura a seguir.



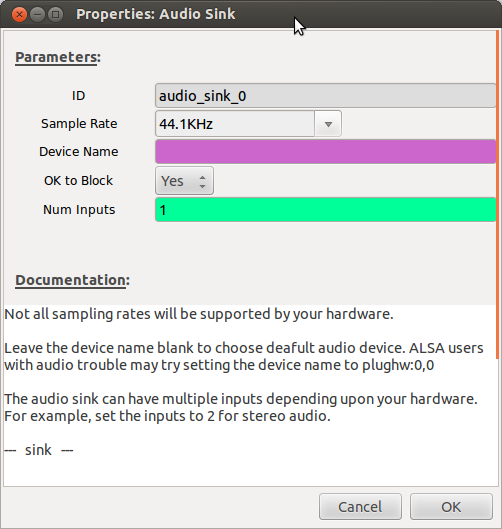
1. Agora vamos configurar o bloco ***Rational Resampler*** que está ligado entre os blocos ***WBFM Receiver*** e o bloco ***Audio Sink****.* Como explicado anteriormente, é muito simples usar este bloco. É apenas uma divisão entre a taxa de amostragem de entrada e a taxa desejada na saída. No nosso caso, estamos em 500 kHz e queremos 44.1 kHz. O bloco ***Rational Resampler***não aceita valores decimais. Logo vemos que a melhor razão para estes valores é de 5000/441. Como mostra o bloco a seguir.



1. Configure o bloco ***Multiply const*** como a seguir.



1. Finalmente, o bloco ***Audio Sink*** *ligado* ao bloco ***WBFM Receiver*** não precisa de configuração. Somente cheque se ele está configurado com 44,1 kHz de *Sample Rate*. Ele deve ficar como na figura a seguir.



1. Como o transmissor ligado, execute seu projeto!!!