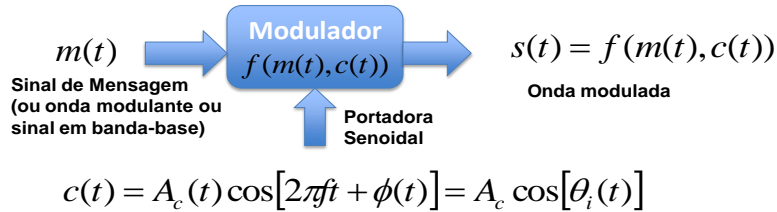


# Hands-on 6

## Parte 01: Receptor WBFM (GRC)

## Introdução Teórica

Todo processo de modulação analógica envolve uma operação (função) entre a onda modulante  $m(t)$  e a onda portadora  $c(t)$ .



**Figura 1:** processo de modulação analógica.

O modulador FM produz uma onda no qual a frequência instantânea varia linearmente com o sinal mensagem,

$$f_i(t) = f_c + k_f m(t) \quad (1)$$

sendo  $k_f$  um parâmetro de projeto chamado **sensibilidade de frequência do modulador (Hz/volt)**. Sabemos que a relação ângulo-frequência é

$$f_i(t) = \frac{1}{2\pi} \frac{d\theta(t)}{dt} \quad (2)$$

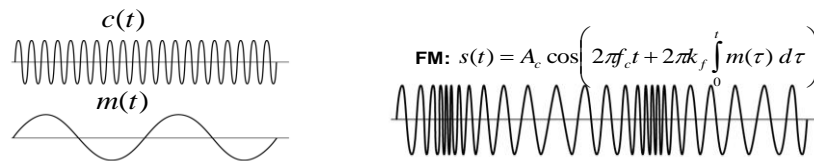
$$\theta_i(t) = 2\pi \int_0^t f_i(\tau) d\tau \quad (3)$$

Dessa forma, desconsiderando o desvio de fase da onda portadora ( $\phi = 0$ ), o modulador FM produz a onda  $s(t)$  a seguir (Eq. (3)).

$$s(t) = A_c \cos \left( 2\pi f_c t + 2\pi k_f \int_0^t m(\tau) d\tau \right) \quad (4)$$

Então, a onda FM  $s(t)$  é uma função não linear da onda moduladora  $m(t)$ . Portanto, a modulação em frequência é um processo não linear de modulação. Sua caracterização temporal pode ser visualizada na figura 2.

**Caracterização temporal:** modulação FM



**Figura 2:** caracterização temporal da modulação FM.

Por simplicidade, consideremos  $m(t)$  um tom, como abaixo.

$$m(t) = A_m \cos(2\pi f_m t) \quad (5)$$

A frequência instantânea da onda FM pode ser reescrita como abaixo.

$$f_i(t) = f_c + k_f m(t) = f_c + k_f A_m \cos(2\pi f_m t) = f_c + \Delta f \cos(2\pi f_m t) \quad (6)$$

Da mesma forma, o ângulo de  $s(t)$  pode ser reescrito como abaixo.

$$\theta_i(t) = 2\pi \int_0^t f_c + \Delta f \cos(2\pi f_m \tau) d\tau = 2\pi f_c t + \frac{2\pi \Delta f}{2\pi f_m} \sin(2\pi f_m t) = 2\pi f_c t + \frac{\Delta f}{f_m} \sin(2\pi f_m t) \quad (7.1)$$

$$\theta_i(t) = 2\pi f_c t + \beta \sin(2\pi f_m t) \quad (7.2)$$

$$\beta = \frac{\Delta f}{f_m} \quad (7.3)$$

Assim,  $s(t)$  assume a forma a seguir.

$$s(t) = A_c \cos(2\pi f_c t + \beta \sin(2\pi f_m t))$$

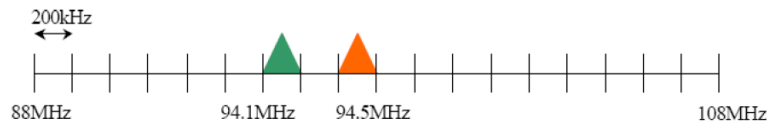
O parâmetro  $\beta$  é chamado **de índice de modulação** e  $\Delta f$  de **desvio de frequência**. Enquanto  $k_f$  serve para controlar a variação de frequência da onda FM, o  $\beta$  determina a variação máxima de ângulo.

Transmissores FM comerciais (FCC, ANATEL) usam um desvio de frequência máximo de 75kHz e uma frequência máxima do sinal mensagem de 15 kHz, resultando em um índice de modulação máximo de 5.

Dependendo do índice de modulação são definidos dois tipos de modulação FM: (i) modulação FM banda estreita (NBFM); e (ii) modulação FM banda larga (WBFM).

- **Caso 1:** FM de banda estreita (narrowband FM):  $\beta$  é pequeno em comparação com 1 radiano ( $< 0,3$  radiano). Comportamento similar a modulação AM. **Algumas aplicações dessa modulação são:** *magnetic tape storage* (porções de luminância do sinal de vídeo em videocassetes); serviços de voz (quando a fidelidade de áudio não é importante, e.g. radio amador); e GSM (GMSK).
- **Caso 2:** FM de banda larga (wideband FM):  $\beta$  é grande em comparação com 1 radiano. Esse é o caso do FM comercial e outras aplicações tais como áudio da TV e sistemas AMPS ( $f_m = 3$  kHz;  $\Delta f = 12$  kHz).

No Brasil, a faixa de FM comercial permitida é de 88 a 108 MHz. Essa faixa é dividida em porções de 200kHz, o que daria um total de 100 estações de rádio. Na prática, são utilizadas no máximo 50 estações de rádio numa mesma região, pois não é permitido que duas emissoras ocupem faixas vizinhas. Isso é uma maneira de prevenir interferências de uma estação na outra e para permitir transmissão de sinais de áudio estéreo (veja figura abaixo).



**Figura 2:** FM no Brasil.

Até 1961, toda a transmissão FM era monofônica, a partir dessa época passaram a ser autorizadas as transmissões FM comerciais em dois canais. O principal problema em introduzir transmissões estereofônicas era a compatibilidade com os receptores monofônicos.

Assim, devia ser criada uma estratégia que permitisse que os sinais  $l(t)$  (left = esquerdo) e  $r(t)$  (right = direito), que são as informações de estéreo, fossem codificados de tal forma que os receptores estéreos pudessem decodificá-las e os receptores monofônicos também.

De maneira simples, no transmissor FM, o canal esquerdo  $l(t)$  e direito  $r(t)$  são misturados, gerando os sinais de soma  $l(t) + r(t)$  e diferença  $l(t) - r(t)$ . Assim, receptores monofônicos poderiam trabalhar com o sinal de soma  $l(t) + r(t)$ , enquanto que receptores estéreos recuperariam os dois canais como abaixo:

- Canal  $l(t)$ : soma do sinal soma e do sinal diferença =  $l(t) + r(t) + l(t) - r(t) \sim l(t)$
- Canal  $r(t)$ : diferença do sinal soma e do sinal diferença =  $l(t) + r(t) - l(t) + r(t) \sim r(t)$

## Exercício

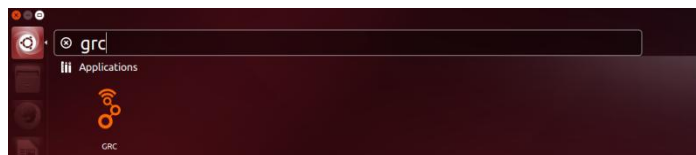
**OBJETIVO:** construir um receptor FM usando o USRP N210 e o GNU Radio Companion. GNU Radio Companion (GRC) é uma interface gráfica do usuário (GUI) que permite construir projetos com o GNU Radio (GNU Radio flowgraphs). É uma maneira excelente de aprender o básico em GNU Radio.

1. Caso ainda não esteja aberto, inicialize o GNU Radio Companion.
  - a. Abra um terminal digitando CTRL+ALT+t e digite: `gnuradio-companion` e pressione ENTER

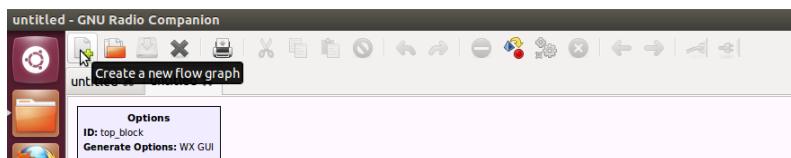
```
juliana@juliana-VPCEG13EB: ~  
juliana@juliana-VPCEG13EB:~$ gnuradio-companion
```

Alternativa:

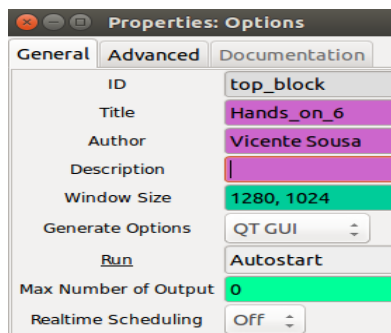
- a. Clique em Dash Home e digite `grc` e clique no ícone correspondente ao GRC



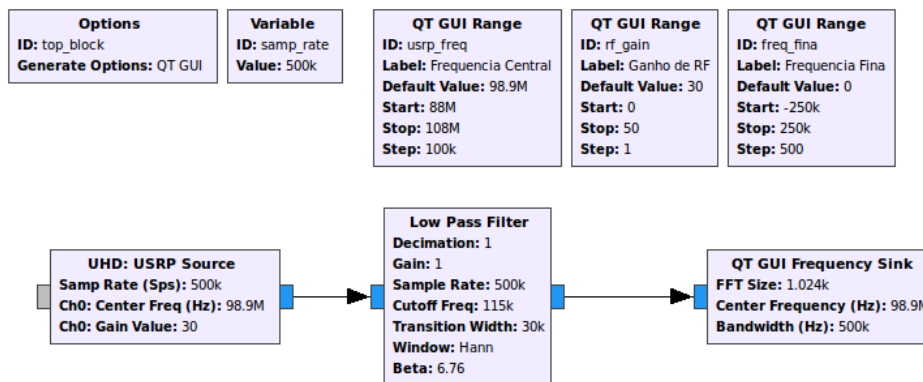
2. Com o GNU RADIO COMPANION aberto, crie um novo projeto.



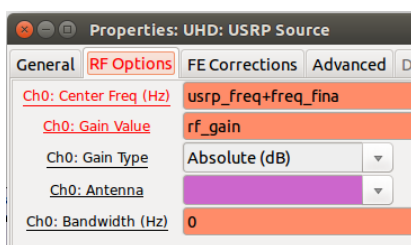
3. Clique duas vezes no Bloco Options. Esse bloco configura alguns parâmetros gerais de flowgraph. Mantenha o *ID* como *top\_block*. Digite um título para o projeto e um autor. Selecione *Generate Options* com *QT GUI*, *Run* para *Autostart* e *Realtime Scheduling* para *Off*. Então, feche a janela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



4. Monte um flowgraph como os seguintes blocos: **UHD: USRP Source**, 3 blocos **QT GUI Range**, um **Low Pass Filter**, e um **QT GUI Frequency Sink**. Os blocos devem ser conectados como na figura a seguir.

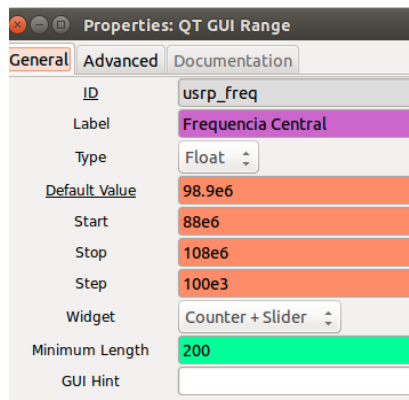


5. Clique duas vezes no bloco **UHD: USRP Source** e configure-o como abaixo, defina o *Ch0: Center Freq (Hz)* como *usrp\_freq* e o *Ch0: Gain (dB)* como *rf\_gain*.



Obs.: mesmo depois de todas essas configurações o bloco ainda continuará vermelho, principalmente porque as variáveis ainda não foram definidas.

6. Clique duas vezes no primeiro bloco **QT GUI Range** e configure-o como na figura a seguir. Defina o *ID* como *usrp\_freq*, *Label* como *Frequencia Central*, *Default Value* como *98.9e6*, *Start* como *88e6*, *Stop* como *108e6*, *Step* como *100e3* e selecione *Widget* para Counter + Slider.

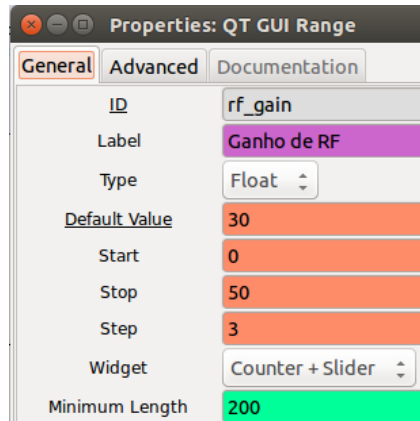


Obs.: Não use acentos para os *Labels*. Isso pode gerar problemas de salvamento no arquivo.

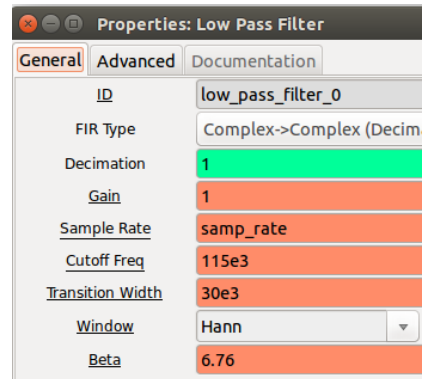
Obs.: note que depois de configurar esse bloco, só existe uma variável em vermelho no bloco UHD: USRP Source.

7. Clique duas vezes no segundo bloco **QT GUI Range** e configure-o como na figura a seguir. Defina o *ID* como *rf\_gain*, *Label* como *Ganho de RF*, *Default Value* como *30*, *Start* como *0*, *Stop* como *50*,

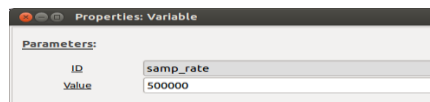
Step como 3 e *Widget* como Counter + Slider.



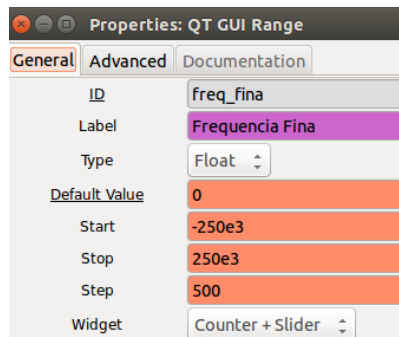
8. Clique duas vezes no bloco **Low Pass Filter** e configure-o como na figura a seguir. Defina *Decimation* como 1, *Gain* como 1, *Cutoff Freq* como 115e3, *Transition Width* como 30e3, *Window* Hann e *Beta* 6.76.



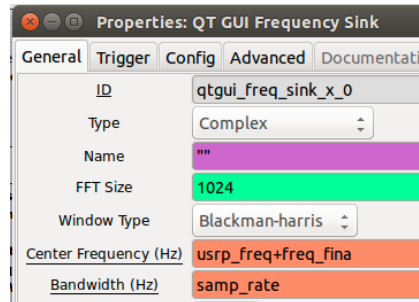
9. Agora precisamos aumentar a taxa de amostragem. Clique duas vezes no bloco *Variable* que tem o ID *samp\_rate*. Mude o valor para 500000 (500 kHz). O bloco ficará similar a figura a seguir.



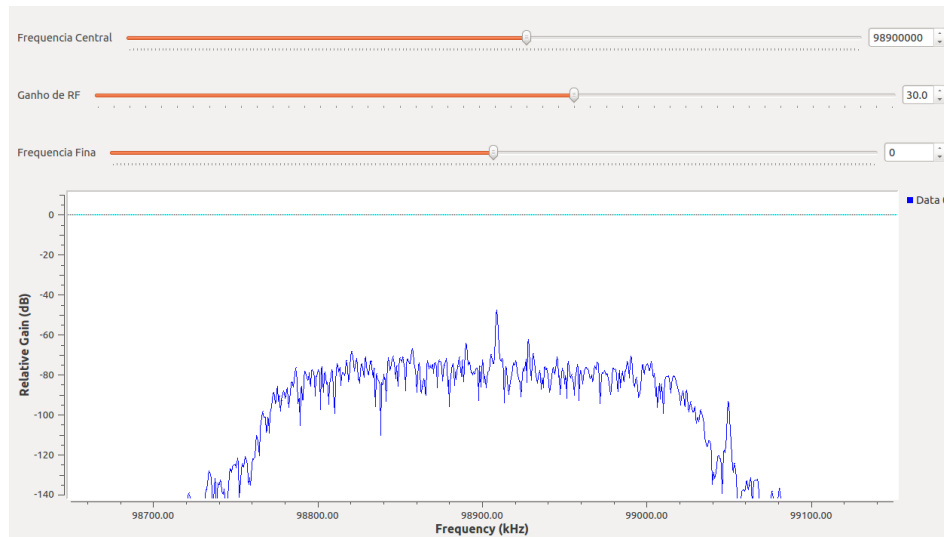
10. Clique duas vezes no terceiro bloco **QT GUI Range** e configure-o como na figura a seguir. Defina o *ID* como *freq\_fina*, *Label* como *Frequencia Fina*, *Default Value* como 0, *Start* como -250e3, *Stop* como 250e3, *Step* como 500 e *Widget* como Counter + Slider.



11. Configure o bloco **QT GUI Frequency Sink** como a seguir:

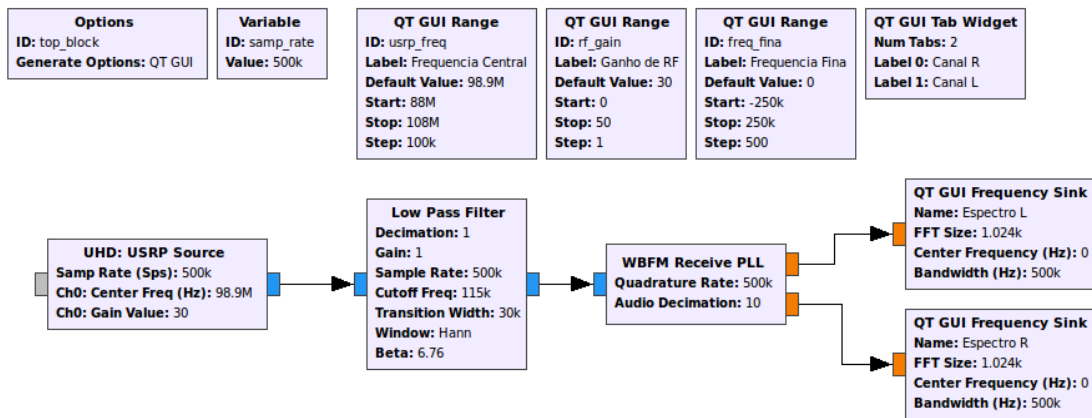


12. Salve o projeto com o nome **hands\_on\_05\_step\_01.grc**, gere o *flowgraph* e execute-o. Agora você será capaz de visualizar o espectro do sinal de RF na frequência selecionada no *slider*. A saída deve ser similar ao mostrado na figura a seguir.

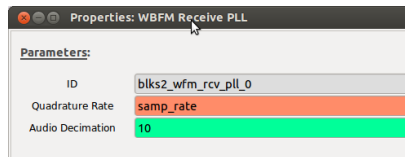


13. O próximo passo é inserirmos o receptor FM e visualizarmos sua saída. Procure e Arraste um bloco **WBFM Receive PLL** para a área de trabalho, conecte-o com a saída do **Low Pass Filter**. Insira dois blocos **QT GUI Frequency SINK** e um bloco **QT GUI Tab Widget**. Conecte-os como na figura a seguir. Lembre-se de ajustar as variáveis de modo que elas fiquem com tipos compatíveis.

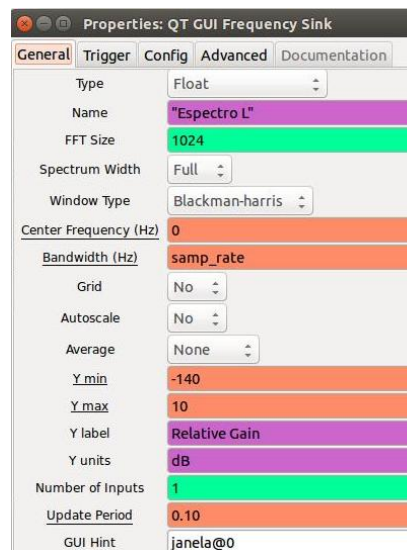




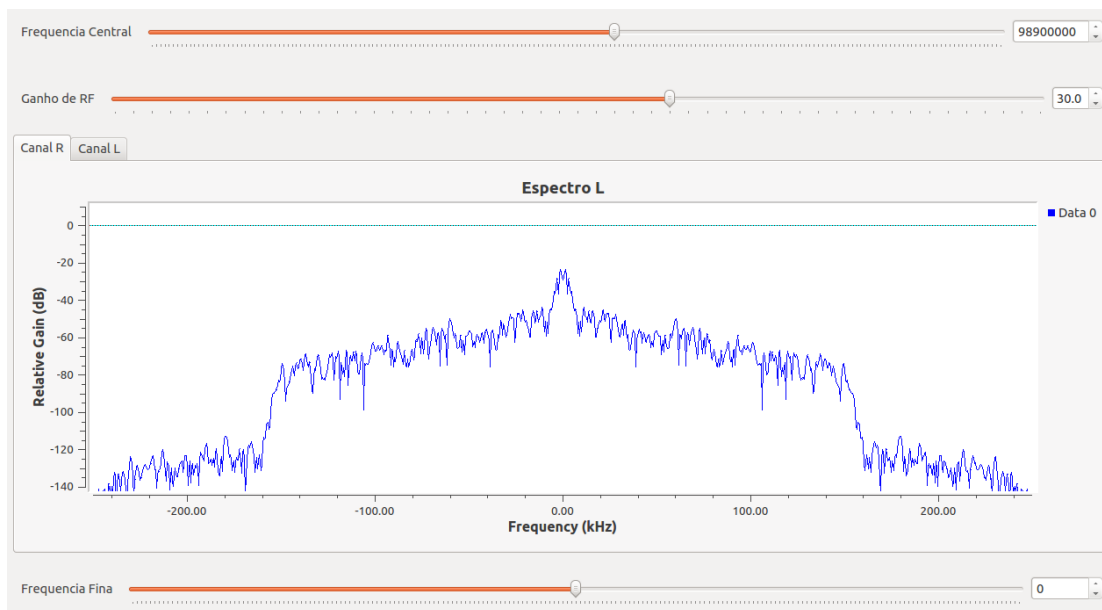
14. Dê dois cliques no bloco **WBFM Receive PLL** e configure-o como na figura a seguir. Defina o *Quadrature Rate* como *samp\_rate* e *Audio Decimation* como *10*. Assim, o sinal na saída terá a taxa de amostragem 10x menor (50KHz).



15. Agora precisamos configurar os blocos **QT GUI Frequency Sink**. Configure o campo *Name* do **QT GUI Frequency** ligado à saída **Lout** do receptor FM para "Espectro L". Configure o *GUI Hint* para *janela@0*. Configure o campo *Name* do **QT GUI Frequency** ligado à saída **Rout** do receptor FM para "Espectro R". Configure o *GUI Hint* para *janela@1*. Configure o *Center Frequency* de ambos os blocos para *0*.



16. Salve o projeto com o nome **hands\_on\_06\_step\_02.grc**, gere o *flowgraph* e execute-o. Como a frequência central está configurada para 98.9 MHz (e você está em Natal, RN) aparecerá um sinal demodulado similar ao da figura a seguir.



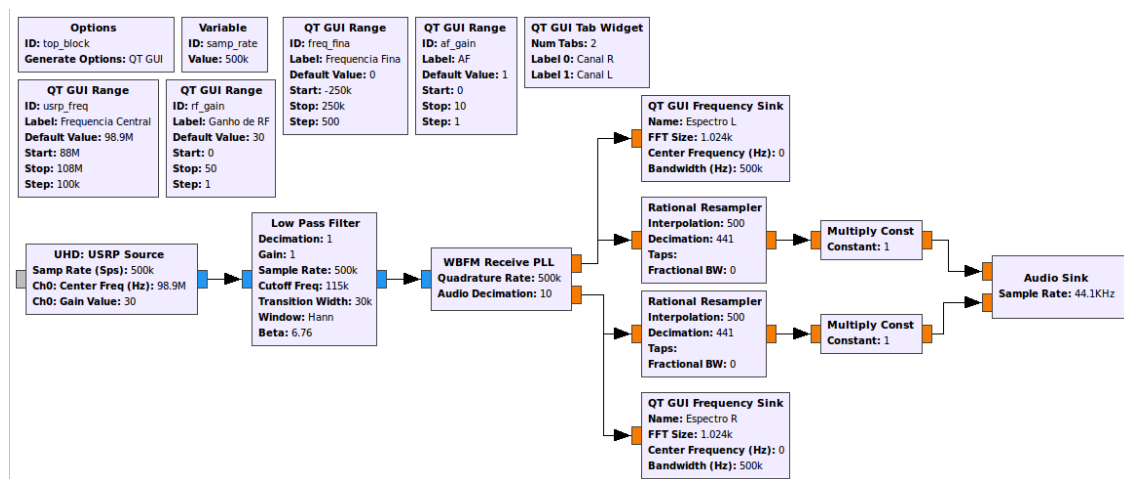
17. Agora vamos escutar a recepção através da placa de som do computador. Para isso, usaremos o auxílio de quatro blocos adicionais, descritos abaixo:

**RationalResampler:** para adequar a taxa de amostragem do sinal recebido para a taxa da placa de áudio.

**Multiply Constant e Slider:** para permitirmos o ajuste via **slider** da amplitude do sinal (controle de volume) a ser enviado à placa de som.

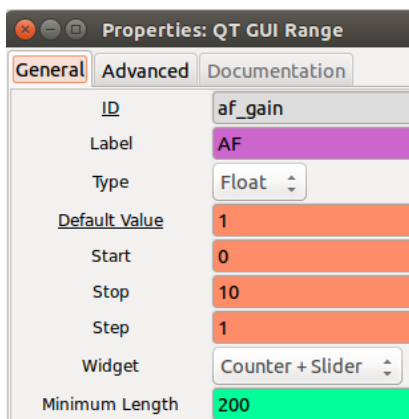
**Audiosink:** para que o sinal demodulado seja reproduzida pela placa de som.

18. Salve o flowgraph atual como **hands\_on\_06\_step\_03.grc**. Procure e arraste para sua área de trabalho os seguintes blocos: outro **QT GUI Range**, dois blocos **Rational Resampler**, dois blocos **Multiply Const**, um bloco **AudioSink**. Conecte-os como na figura a seguir. Lembre-se de ajustar as variáveis de modo que elas fiquem com tipos compatíveis.

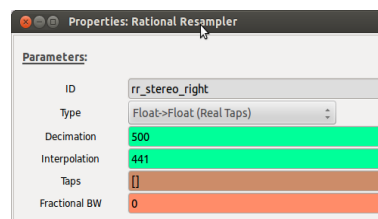
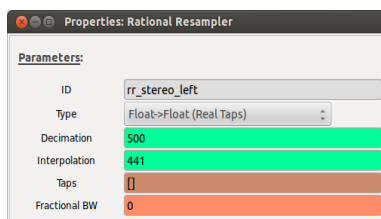


19. Configure o novo **QT GUI Range** como na figura a seguir. Defina **ID** como **af\_gain**, **Label** como **AF**,

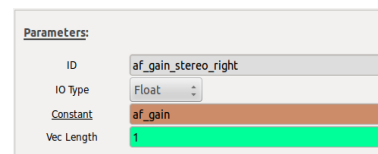
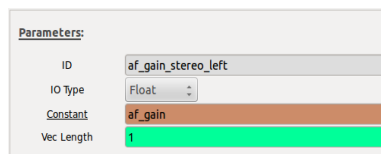
Default Value como 1, Start como 0, Stop como 10, Step como 1.



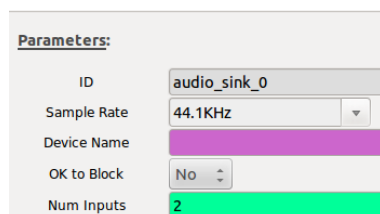
20. Configure os dois blocos **Rational Resampler** como na figura a seguir. Defina o *Type* como *Float->Float (Real Taps)*, *Decimation* como 500, *Interpolation* como 441. Os blocos receberão o áudio em uma frequência de amostragem de 50 kHz e queremos reproduzir ele em 44.1KHz. Faremos os blocos com esses valores, já que eles são exatamente a divisão entre 50 e 44.1. Defina um como o ID *rr\_stereo\_left* e o outro com *rr\_stereo\_right*.



21. Configure os dois blocos **MultiplyConst** como na figura a seguir. Defina *IO Type* como *Float* e o campo *Constant* com o *af\_gain*. Defina um como o ID *af\_gain\_stereo\_left* e o outro com *af\_gain\_stereo\_right*.

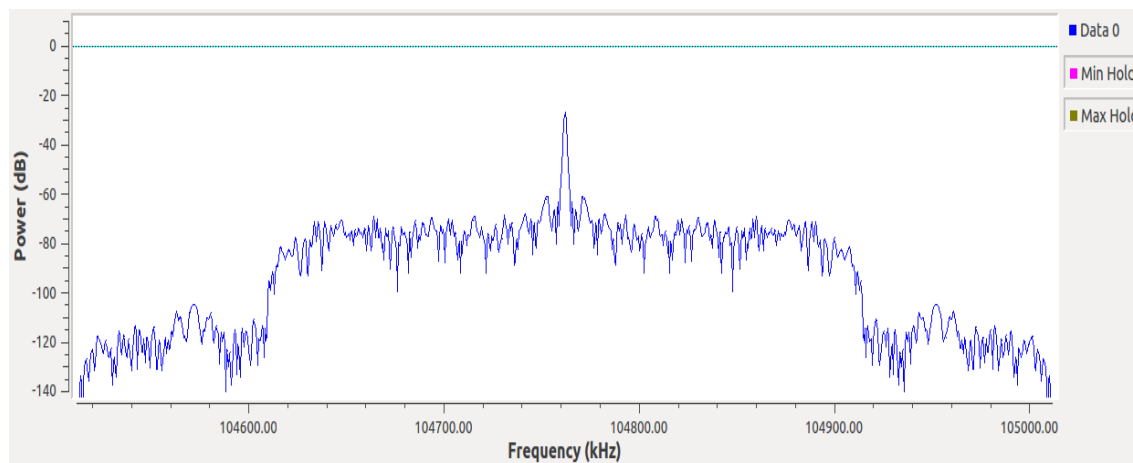


22. Configure o bloco **AudioSink** como na figura a seguir. Defina *Sample Rate* como 44,1kHz e *Num Inputs* como 2.

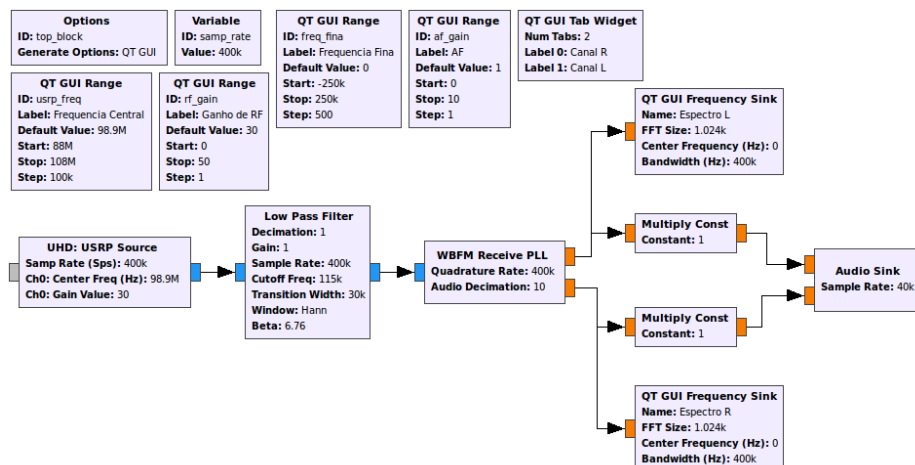


23. Caso não queira mais ver o espectro do sinal de áudio, os dois **QT GUI FrequencySink** podem ser apagados ou desativados.

24. Gere e execute o *flowgraph* novamente. Com a frequência central ajustada para 104.7 MHz aparecerá um sinal demodulado e o som é reproduzido na caixa de som (claro, se você estiver em Natal/RN).



25. Caso seu computador não tenha processamento suficiente para rodar o experimento, gere um novo *flowgraph* mudando a taxa de amostragem para 400kHz (variável *samp\_rate*), o *Sample Rate* do bloco **Audio Sink** para 40kHz e retirando os blocos **Rational Resamplers**, como na figura a seguir:



26. Esse experimento mais simples rodou até em máquina virtual.