

Hands-on 7

Receptor WBFM usando o Dongle

Introdução Teórica

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

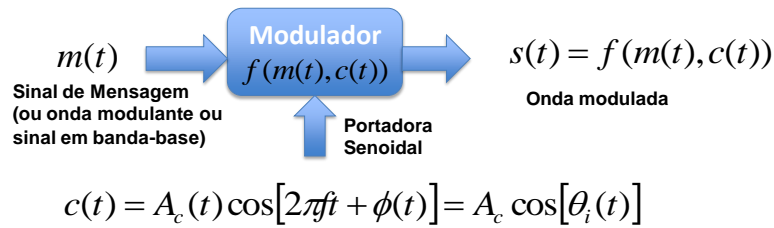


Figura 1: processo de modulação analógica.

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

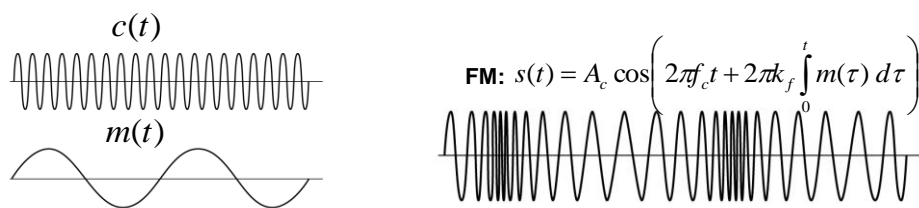


Figura 2: caracterização temporal da modulação FM.

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 β é chamado de **índice de modulação** e Δf de **desvio de frequência**. Enquanto k_f serve para controlar a variação de frequência da onda FM, o β 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): β 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): β 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).

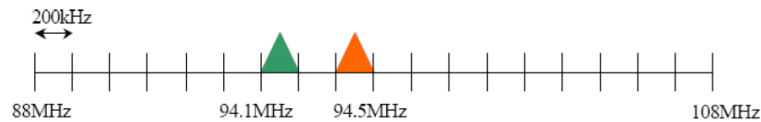


Figura 2: FM no Brasil.

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 = $l(t) + r(t) - l(t) + r(t) \sim r(t)$

Dongle

Dongle RTL SDR é um dispositivo SDR barato que possibilita taxa de amostragem de até 2.56MS/s sem perdas e usado para DAB/DAB+/ Demodulação FM, a faixa de frequência varia de 52-2200 Mhz.

O Dongle E4000 pode ser comprado no Amazon por apenas \$20, por isso se torna uma boa escolha para uso de fins acadêmicos, pode utiliza-lo no GNURadio e em diversos outros programas.



Exercício

OBJETIVO: Demodulação FM usando o software GNURadio GRC e o dispositivo Dongle SDR Tv Digital.

1. Plug o dongle e abra um terminal para testá-lo. Digite: rtl_test

```
gppcom@ubuntu:~$ rtl_test
Found 1 device(s):
 0: Generic, RTL2832U, SN: 77771111153705700

Using device 0: Generic RTL2832U
Found Rafael Micro R820T tuner
Supported gain values (29): 0.0 0.9 1.4 2.7 3.7 7.7 8.7 12.5 14.4 15.7 16.6 19.7
20.7 22.9 25.4 28.0 29.7 32.8 33.8 36.4 37.2 38.6 40.2 42.1 43.4 43.9 44.5 48.0
49.6
[R82XX] PLL not locked!
Sampling at 2048000 S/s.

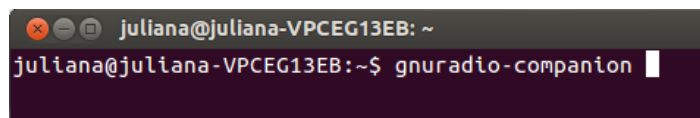
Info: This tool will continuously read from the device, and report if
samples get lost. If you observe no further output, everything is fine.

Reading samples in async mode...
```

Para qualquer problema, consulte a apresentação sobre instalação do dongle, mas uma solução rápida para o erro “Kernel driver is active” é digitar:

```
sudo rmmod rtl2832_sdr dvb_usb_rtl28xxu
```

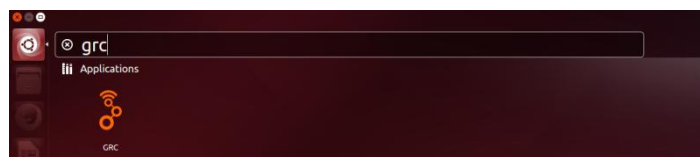
2. Caso ainda não esteja aberto, inicialize o GNU Radio Companion.
 - a. Abra um terminal digitando CTRL+ALT+t e digite: gnuradio-companion e pressione ENTER



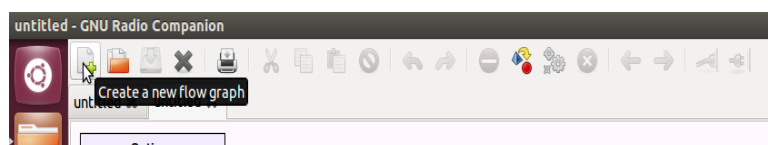
```
juliana@juliana-VPCEG13EB: ~
juliana@juliana-VPCEG13EB:~$ gnuradio-companion
```

Alternativa:

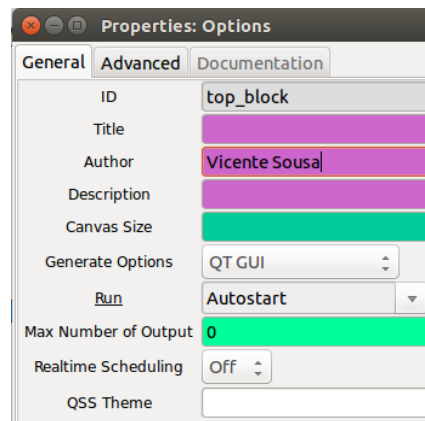
- a. Clique em Dash Home e digite gnuradio e clique no ícone correspondente ao GRC



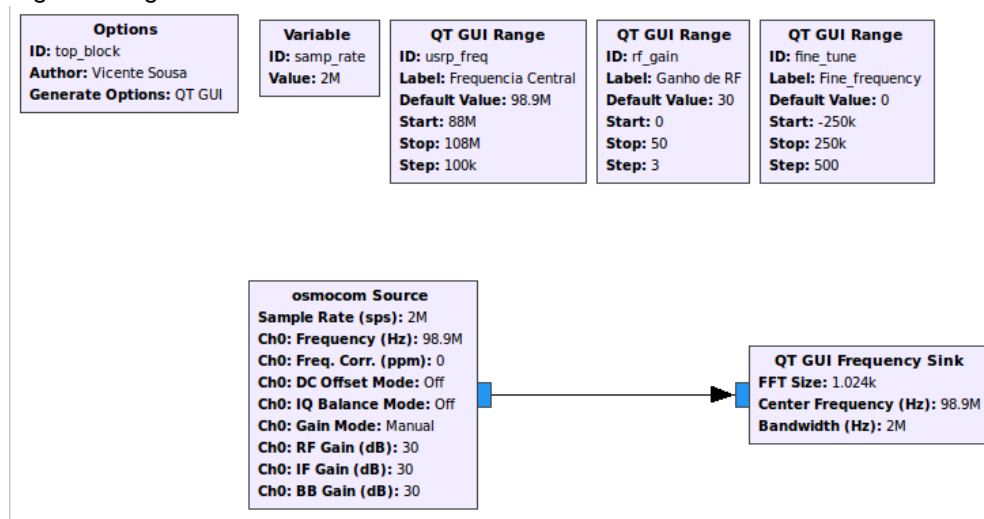
3. Com o GNU RADIO COMPANION aberto, crie um novo projeto.



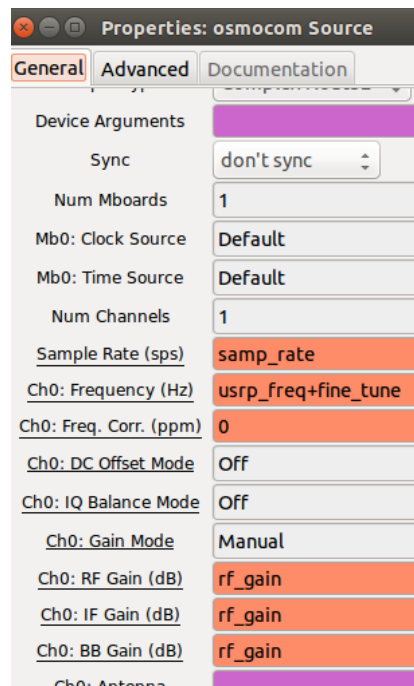
4. Clique duas vezes no Bloco **Options**. Esse bloco configura alguns parâmetros gerais de flowgraph. Mantenha o ID como top_block. Digite um título para o projeto e um autor. Selecione *GenerateOptions* com QT GUI, *Run* para Autostart e *Realtime Scheduling* para Off. Então, feche a janela de propriedades.



5. Monte um flowgraph como os seguintes blocos: **OsmoSDR Source**, 3 blocos **QT GUI Range**, um **Variable**, e um **QT GUI Frequency Sink**. Os blocos devem ser conectados como na figura a seguir.

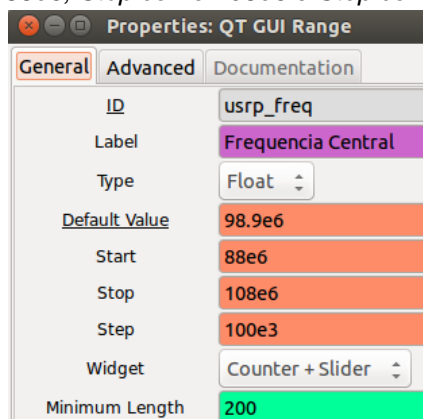


6. Clique duas vezes no bloco **OsmoSDR Source** e configure-o como abaixo, defina o *Ch0: Center Freq (Hz)* como *usrp_freq+fine_tune* e o *Ch0: Gain (dB, Ch0: IF Gain (dB)* e *Ch0: BB Gain (dB))* como *rf_gain*.



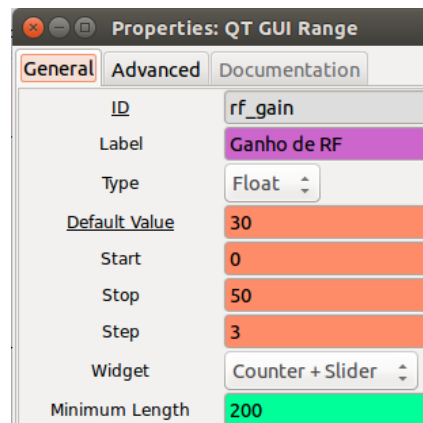
Obs.: mesmo depois de todas essas configurações o bloco ainda pode continuar vermelho, principalmente porque as variáveis ainda não foram definidas.

7. 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* e *Step* como *100e3*.

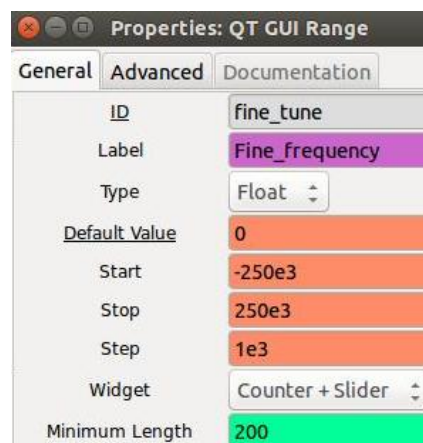


Obs.: Não use acentos para os *Labels*. Isso pode gerar problemas de salvamento no arquivo.

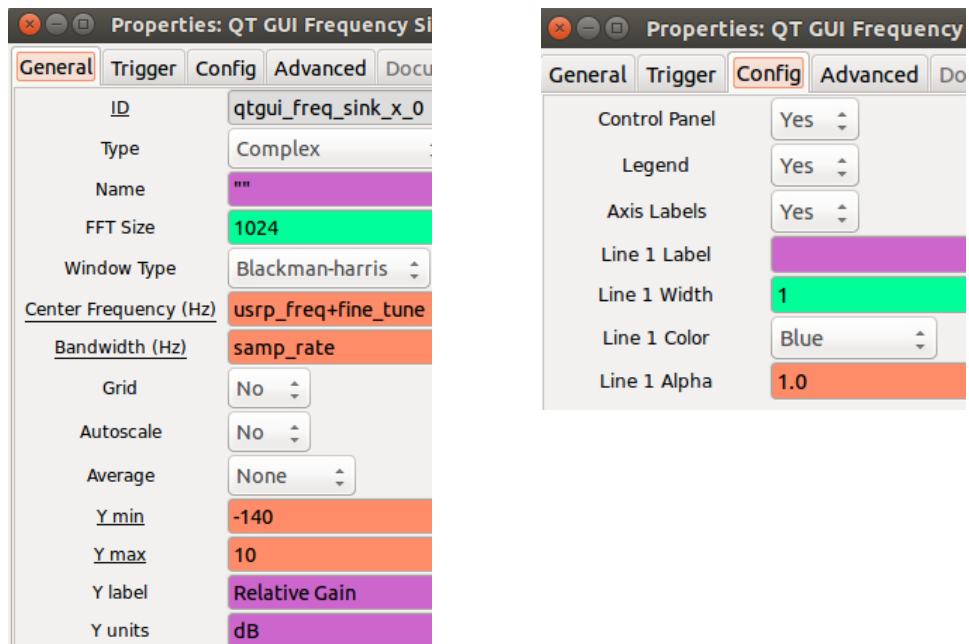
8. Clique duas vezes no segundo bloco **QT GUI Range** e configure-o como na figura a seguir. Defina o *ID* como *rf_gain*, *Label* como *Ganho de RF*, *Default Value* como *30*, *Start* como *0*, *Stop* como *50* e *Step* como *3*.



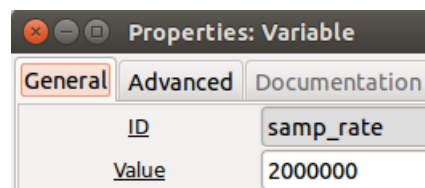
9. Clique duas vezes no terceiro bloco **QT GUI Range** e configure-o como na figura a seguir. Defina o *ID* como *fine_tune*, *Label* como *Fine Frequency*, *Default Value* como 0, *Start* como -250e3, *Stop* como 250e3 e *Step* 1e3.



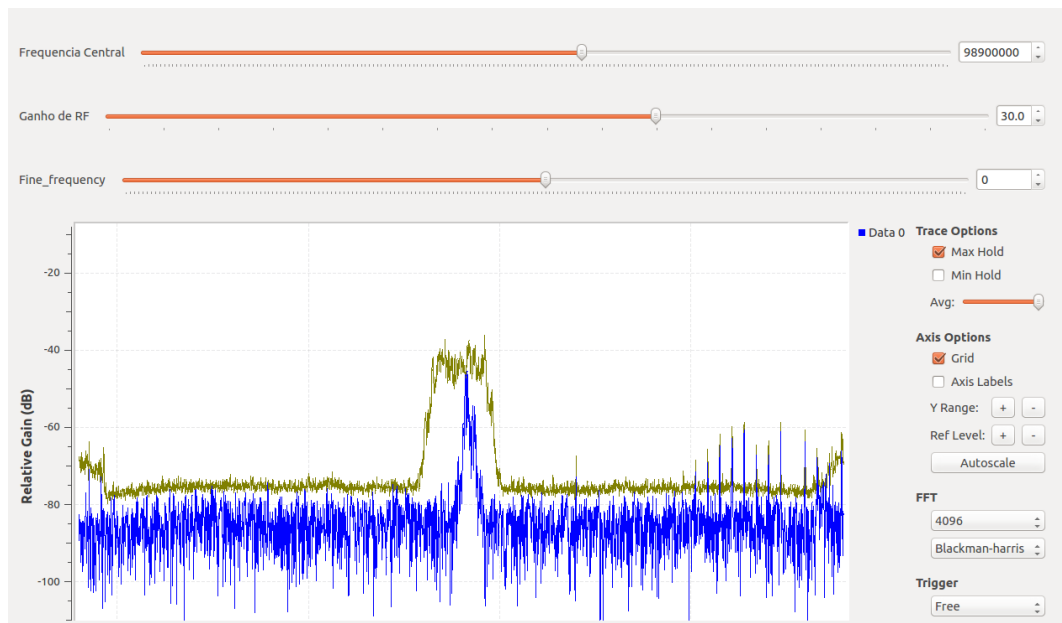
10. Clique duas vezes no bloco **QT GUI Frequency Sink** e configure-o como na figura a seguir. Defina *Center Frequency (Hz)* como *usrp_freq+fine_tune*, *Bandwidth (Hz)* como *samp_rate*, *Y Label* como *Relative Gain*, *Y Units* como *dB*. Na aba *Config*, configure *Control Panel* para *Yes*.



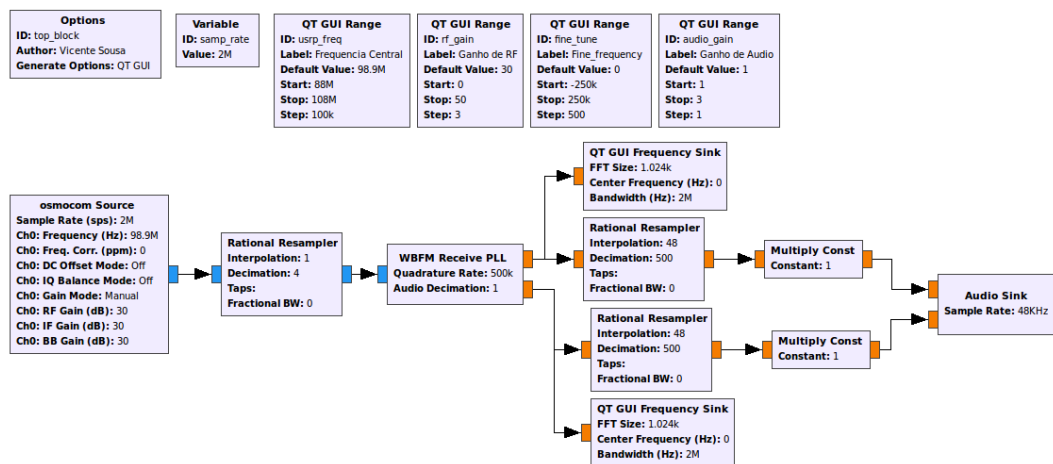
11. Agora precisamos aumentar a taxa de amostragem. Clique duas vezes no bloco **Variable** que tem o ID *samp_rate*. Mude o valor para 2000000 (2 MHz). O bloco ficará similar a figura a seguir.



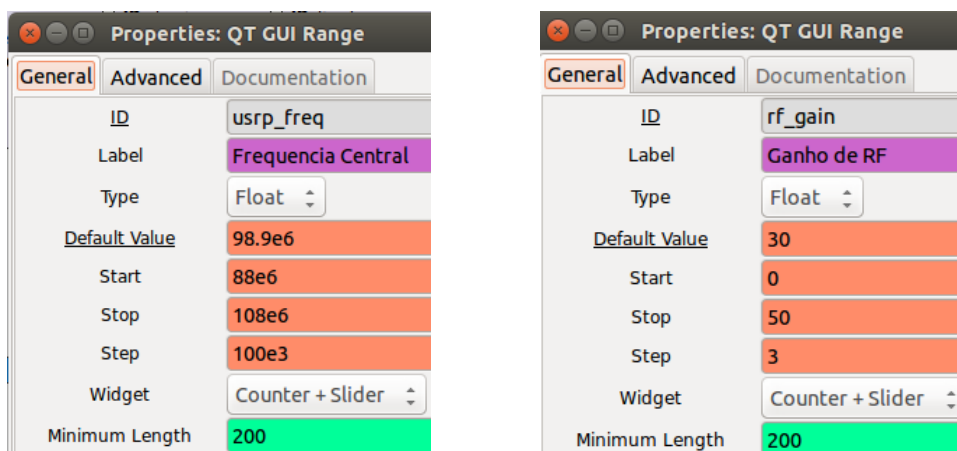
12. Salve o projeto com o nome ***hands_on_07_step_01.grc***, gere o *flowgraph* e execute-o. Agora você será capaz de visualizar o espectro do sinal de RF na frequência selecionada no *QT Range*. Manipule o painel de controle ao lado e você pode obter a saída similar a mostrado na figura a seguir.

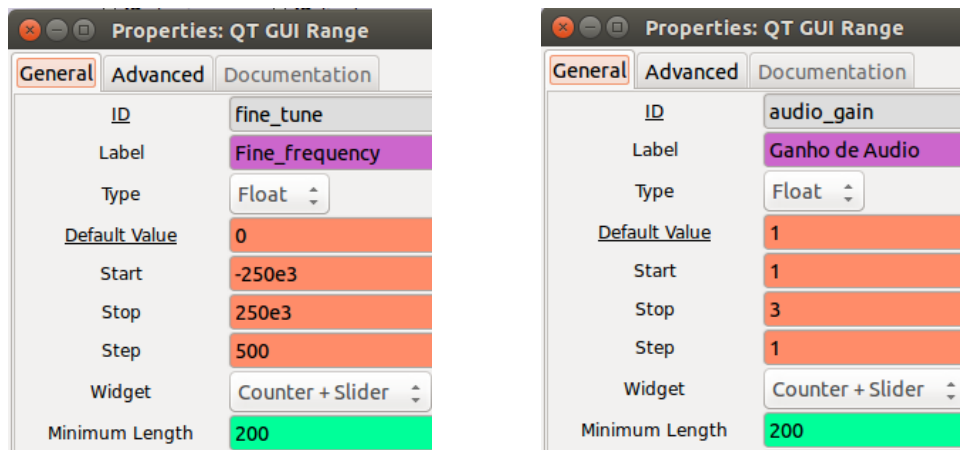


13. Agora vamos criar um receptor WBFM. Crie um novo projeto, deixando-o como na figura abaixo:

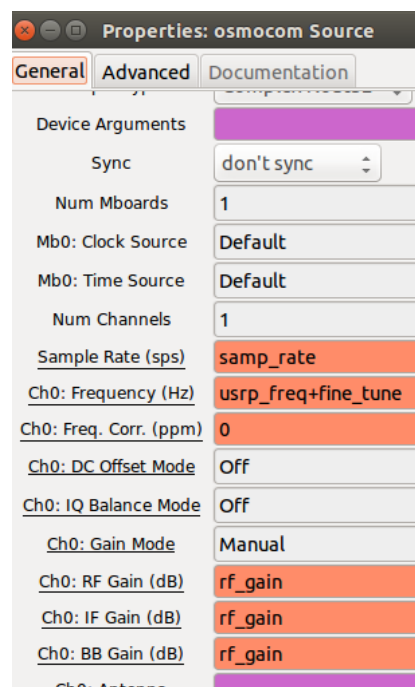


14. Os blocos **QT Range** devem ser configurados como a seguir.

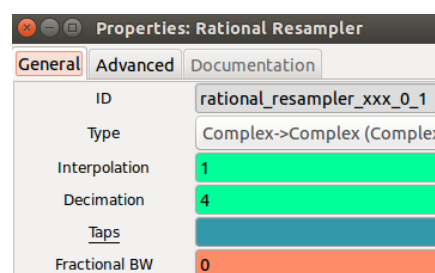




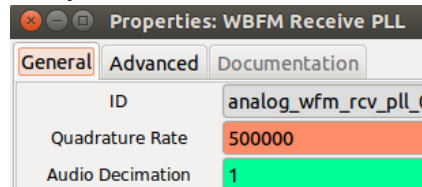
15. O bloco **OsmoSDR Source** deve ser configurado como antes. Defina o *Ch0: Center Freq (Hz)* como *usrp_freq+fine_tune* e o *Ch0: Gain (dB, Ch0: IF Gain (dB) e Ch0: BB Gain (dB))* como *rf_gain*.



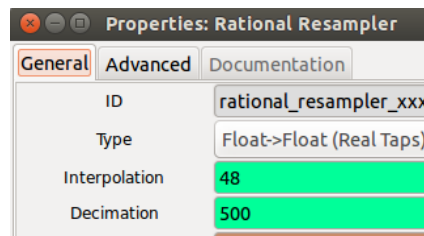
16. O próximo bloco a ser configurado é o **Rational Resampler** depois do bloco **OsmoSDR Source**, que tem a função de converter a taxa de amostragem. Ele usa a seguinte fórmula para calcular a nova taxa de amostragem: $tx_{saida} = tx_{entrada} \left(\frac{\text{interpolação}}{\text{decimação}} \right)$. No nosso caso, a taxa desejada é de 500kHz e a de entrada é de 2MHz, por isso configuramos o **decimation** 1 e o **interpolation** é 4.



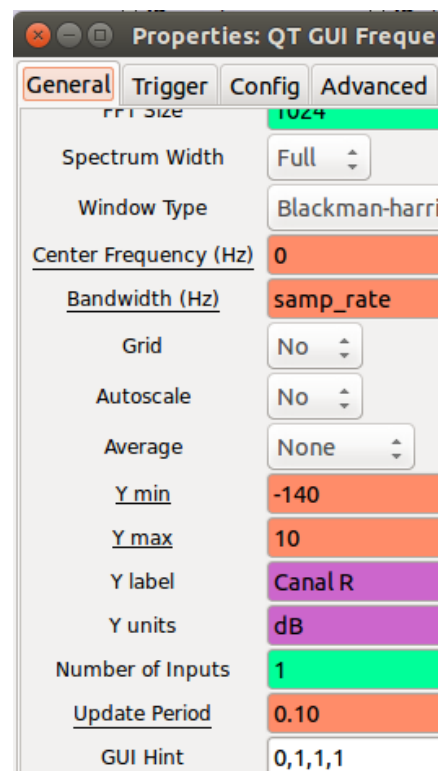
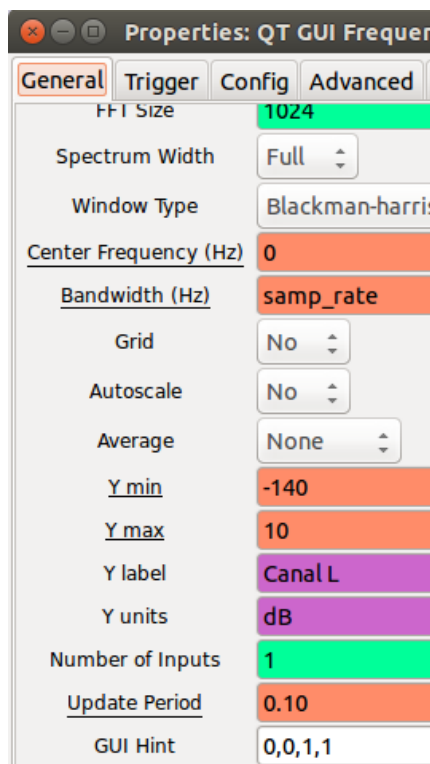
17. O terceiro bloco a ser configurado é o bloco de Demodulação FM Estéreo, o **WBFM Receive PLL**. Este bloco possui duas saídas na Demodulação, a saída esquerda **Lout** e a direita **Rout**. Ele também pode realizar a compactação no tempo do áudio recebido, o parâmetro **Audio Decimation** controla o quanto o áudio será compactado. No nosso caso, o **Quadrature Rate** não sofrerá redução.



18. Os próximos blocos a serem configurados são os **Rational Resamplers** na saída do bloco **WBFM Receive PLL**, que tem a função de converter a taxa de amostragem adequada para a placa de som do computador. No nosso caso, a taxa desejada é de 48kHz e a de entrada é de 500kHz, por isso configuramos o **decimation** 500 e o **interpolation** é 48.



19. Configure os blocos **QT GUI Frequency Sink** como a seguir. Em um bloco configure **Y Label** para **Canal L** e **GUI Hint** para **0,0,1,1**. Em um bloco configure **Y Label** para **Canal R** e **GUI Hint** para **0,1,1,1**.



20. Vamos agora inserir um amplificador de sinal nas duas saídas dos blocos **Rational Resampler**, ou seja, dois blocos **Multiply Const** com parâmetro *Const* igual a *audio_gain* e conecta-los na entrada do bloco **Audio Sink**, este será responsável pela saída do áudio demodulado. Feito isso, gere o Flowgraph e execute-o. Deverá aparecer o espectro em banda-base de cada canal demodulado pelo Dongle.

