



## Lucrare de laborator 3, 4 – Adalm-Pluto și GNU Radio

Ș.I. dr. ing. Zsuzsanna Șuta

### GNU Radio

GNU Radio este un software gratuit și open-source care oferă blocuri de procesare ale semnalelor pentru SDR publicat în 2001. Se poate folosi atât cu dispozitive hardware SDR cât și ca un mediu de simulare. Este folosit în cercetare, industrie, mediul academic pentru studiul unor sisteme radio.

GNU Radio efectuează toate operațiile de procesare de semnal. Se poate folosi pentru a crea aplicații pentru recepția și/sau transmiterea unor semnale digitale, care ulterior vor fi transmise folosind hardware-ul conectat. GNU Radio are implementate filtre, coduri de canal, elemente de sincronizare, egalizatoare, demodulatoare, decodare și multe alte elemente, numite blocuri, care se găsesc în sisteme radio. Include de asemenea și un mod de interconectare și comunicare între aceste blocuri. Este posibilă și extinderea GNU Radio prin definirea unor blocuri proprii.

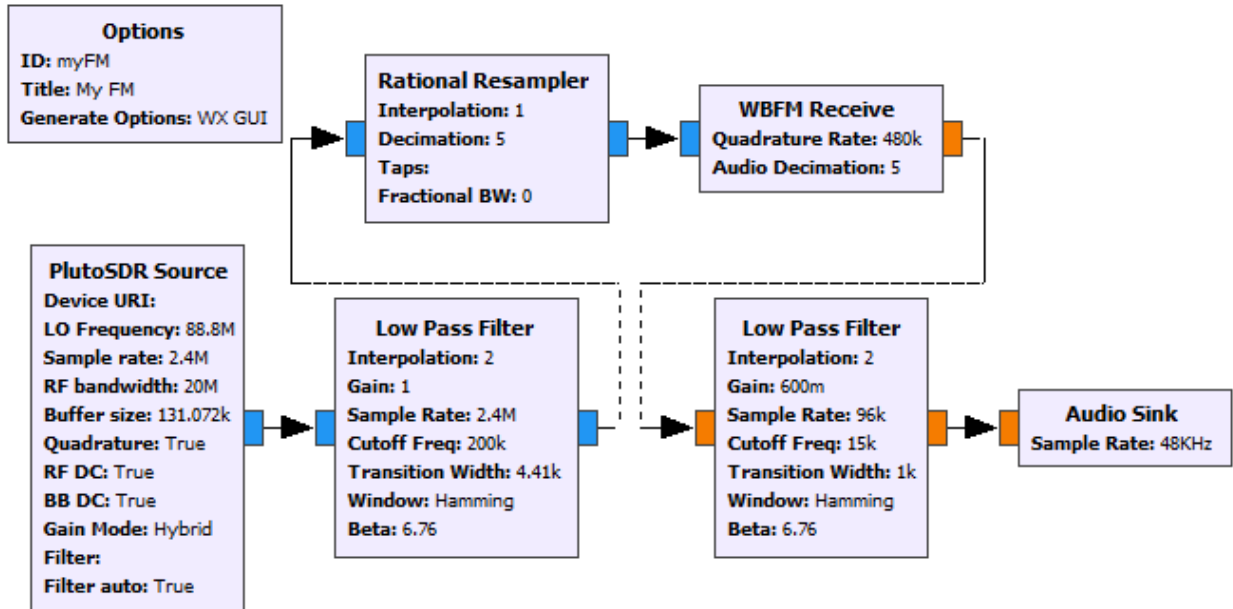
GNU Radio lucrează doar cu date digitale. De obicei tipul de date folosit de receptoare și transmițătoare este de fapt reprezentat de eșantioanele bandă bază complexe. Hardware-ul este folosit pentru translația pe frecvența purtătoare.

Aplicațiile GNU Radio sunt scrise în Python, dar blocurile critice din punctul de vedere al performanței pentru procesare semnalelor sunt scrise în C++.

GNU Radio Companion este o interfață grafică similară cu Simulink care permite realizarea unor diagrame GNU Radio prin drag & drop. GNU Radio are unele blocuri de operații de bază gata implementate, cum ar fi osciloscopul.

### Receptor radio FM

Folosind GNU Radio Companion realizați modelul de mai jos pentru un receptor radio FM.



Blocurile necesare sunt:

- Options – setează parametrii modelului, cum ar fi titlul, autorul, descriere, etc.
- PlutoSDR Source – este receptorul PlutoSDR, se comportă ca sursă pentru blocurile de procesare următoare.
- Low Pass Filter – este filtru trece jos, se poate configura frecvența de eșantionare, frecvența de tăiere și fereastra de tranziție în Hz.
- Rational Resampler – este un filtru FIR polifazic. Intrarea are frecvența  $f_s$ , iar ieșirea  $\text{interpolation} \cdot f_s / \text{decimation}$ . Parametrii interpolation și decimation trebuie păstrați cât mai mici posibili.
- WBFM Receive – este blocul de demodulare al semnalului FM. Intrarea este semnalul BB complex, iar ieșirea este semnalul audio demodulat.
- Audio Sink – reprezintă dispozitivul de redare al sunetului. Numele se lasă gol, astfel alege dispozitivul implicit. Se alege rata de eșantionare.

Aceste blocuri se vor configura după cum urmează.

### Options



Property	Value
ID	myFM
Title	My FM
Author	
Description	
Canvas Size	
Generate Options	WX GUI
Run	Autostart
Max Number of Output	0
Realtime Scheduling	Off

Se specifică opțiunile modelului: ID, titlu, autor, descriere, dimensiunea paginii, opțiunile de generare a simulării (QT GUI, **WX GUI**, No GUI, etc.). Opțiunile Run și Realtime scheduling se lasă pe valorile implicite.

### **PlutoSDR Source**

Property	Value
ID	pluto_source_0
Device URI	
LO Frequency	88800000
Sample rate	2400000
RF bandwidth	20000000
Buffer size	0x20000
Quadrature	True
RF DC	True
BB DC	True
Gain Mode	Hybrid
Filter	
Filter auto	True

Este receptorul PlutoSDR. Se specifică următorii parametri:

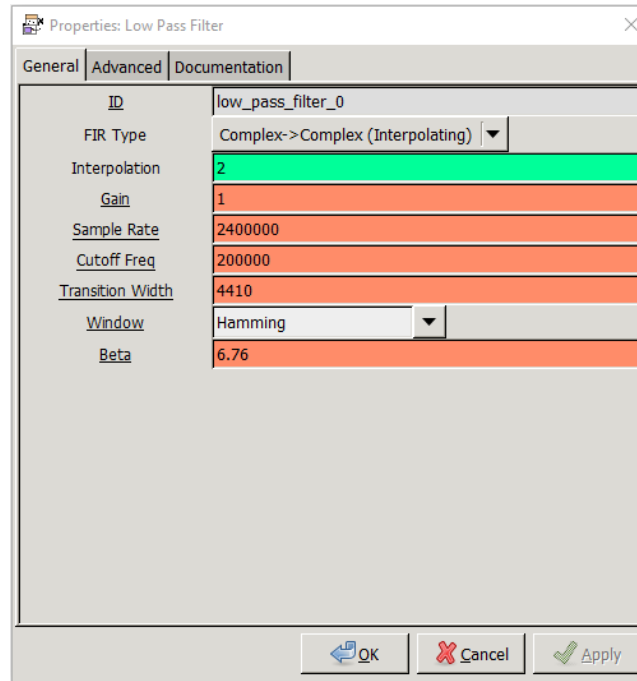
- ID – identificatorul blocului
- LO Frequency – frecvența semnalului recepționat (a postului radio în acest caz) în Hz



- Sample rate – frecvența de eșantionare în Hz
- RF bandwidth – lărgime benzii RF în Hz
- Buffer size – lungime bufferului folosit de PlutoSDR
- Opțiunile RF DC, BB DC, Quadrature și Filter auto se vor seta pe True.
- Gain Mode – câștigul receptorului se setează pe Hybrid.

### Low Pass Filter 1

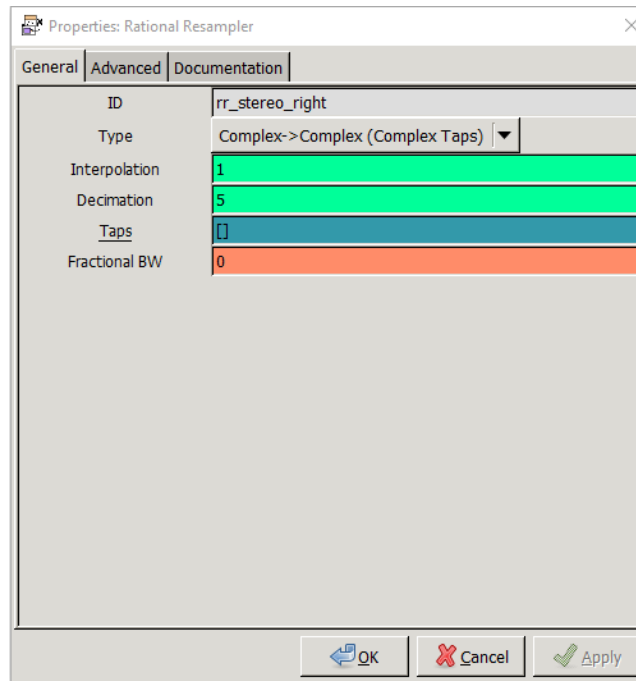
Este un filtru trece jos FIR, care va filtra semnalul recepționat de PlutoSDR.



Se va configura identificatorul filtrului, tipul FIR pentru specificarea tipurilor de date de intrare și de ieșire (se alege Complex -> Complex (Interpolating)), interpolarea filtrului, câștigul filtrului, frecvența de eșantionare în Hz, frecvența de tăiere în Hz, lărgimea de tranziție între banda de oprire și banda de trecere în Hz, tipul ferestrei și valoarea lui Beta (aplicabil doar pentru ferestre de tip Kaiser).

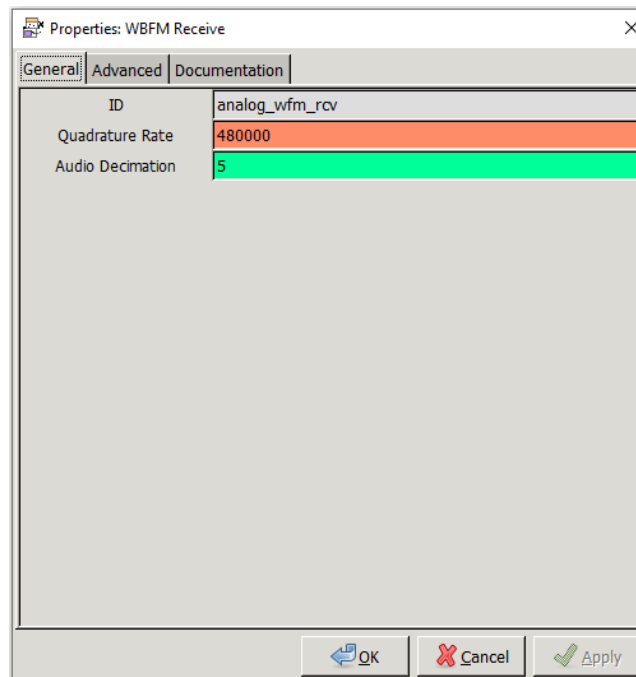
### Rational Resampler

Blocul realizează re-eșantionarea semnalului filtrat. Se alege tipul Complex -> Complex, interpolarea și factorul de decimare. Ceilalți parametri se lasă neschimbați.



### **WBFM Receive**

Este blocul de demodulare al semnalului FM. Se configurează frecvența de cuadratură în Hz și factorul de decimare.



### **Low Pass Filter 2**

Se configurează similar cu primul FTJ, însă cu alți parametrii.



Properties: Low Pass Filter

General | Advanced | Documentation

ID	low_pass_filter_0_0
FIR Type	Float->Float (Interpolating)
Interpolation	2
Gain	0.6
Sample Rate	96000
Cutoff Freq	15000
Transition Width	1000
Window	Hamming
Beta	6.76

OK Cancel Apply

## Audio Sink

Properties: Audio Sink

General | Advanced | Documentation

ID	audio_sink_0
Sample Rate	48KHz
Device Name	
OK to Block	Yes
Num Inputs	1

OK Cancel Apply

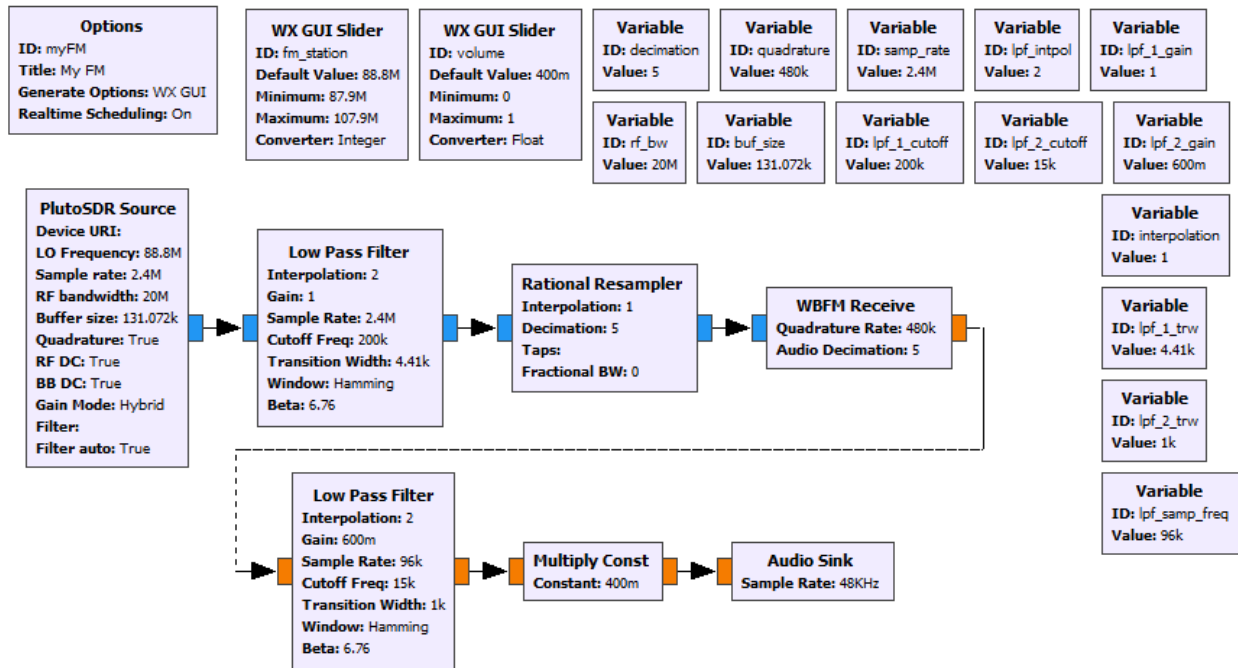
Se configurează frecvența de eșantionare. Nu toate plăcile audio suportă toate frecvențele de eșantionare. Numele dispozitivului se lasă gol, astfel se va alege dispozitivul de redare al sunteului implicit.

După conectarea blocurilor se va porni simularea prin apăsarea butonului Execute the flow graph.

## Paramaterizarea modelului

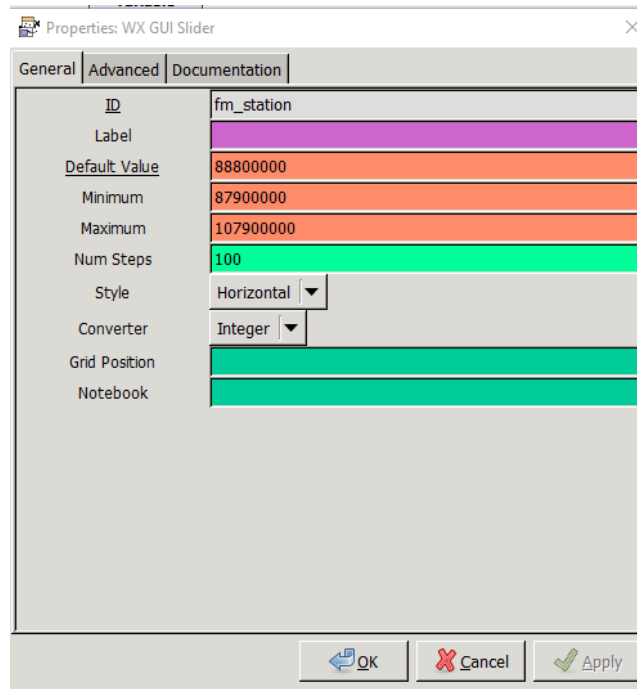


În locul valorilor hard-codate în pașii anteriori se vor folosi variabile. Se vor defini toate variabilele necesare. Pentru frecvența posturilor radio și pentru reglarea volumului se va folosi WX GUI Slider.



Suplimentar, între al doile FTJ și Audio Sink se introduce un bloc de **Multiply Const** care va permite reglajul efectiv al volumului între valorile minime și maxime specificate de WX GUI Slider.

Frecvențele posibile ale sistemului se configurează astfel:





WX GUI Slider pentru volum se configurează conform figurii următoare:

Property	Value
ID	volume
Label	
Default Value	0.4
Minimum	0
Maximum	1
Num Steps	100
Style	Horizontal
Converter	Float
Grid Position	
Notebook	

## Extinderea modelului cu blocuri de analiză a semnalelor

GNU Radio dispune de mai multe blocuri care fac posibilă analiza semnalelor recepționate și/sau procesate. Se vor folosi blocurile WX GUI Histo Sink, WX GUI FFT Sink și WX GUI Scope Sink.

### WX GUI Scope Sink

Este osciloscopul clasic, afișează variațiile semnalului în timp.

Property	Value
ID	wxgui_scopesink2_1
Type	Complex
Title	Scope Plot
Sample Rate	samp_rate
V Scale	0.025
V Offset	0
T Scale	0.001
AC Couple	Off
XY Mode	Off
Num Inputs	1
Window Size	
Grid Position	
Notebook	
Trigger Mode	Auto
Y Axis Label	Amplitude





Parametrii blocului sunt:

- Title – se configurează titlul diagramei
- Sample Rate – frecvența de eșantionare a semnalului afișat
- V Scale – se pune pe 0 ca să se ajusteze automat axa Y. Pentru receptorul FM se poate configura cu valoarea 0.025.
- V Offset – este offsetul axei Y, se lasă configurat cu 0.
- T Scale – se configurează cu 0 pentru ajustarea automată a axei X. Pentru receptorul FM se poate configura cu 0.001.
- AC Couple – este cuplajul AC, se poate configura pe Off sau On.
- XY Mode – permite modul de afișare X-Y. Pentru vizualizarea semnalului în timp trebuie să fie Off.
- Window Size – dacă nu se specifică alte valori, va configura automat dimensiunea ferestrei. Se poate seta folosind o pereche de lățime/înălțime în pixeli.
- Grid Position – se folosește pentru poziționarea graficului în fereastra definită.
- Notebook Param – se folosește pentru integrarea graficului într-o pagină.
- Trigger Mode – se configurează pentru pornirea osciloscopului. În mod Auto va porni la executarea diagramei.
- Y Axis Label – se poate configura numele axei Y.

După pornirea simulării este de asemenea posibilă configurarea osciloscopului din interfața grafică.

### **WX GUI FFT Sink**

Este analizorul spectral integrat în GNU Radio, permite vizualizarea spectrului unui semnal.





Parametrii blocului sunt:

- Title – se configurează titlul diagramei
- Type – se configurează tipul semnalului Complex sau Float
- Sample Rate – frecvența de eșantionare a semnalului afișat
- Baseband Freq – este frecvența centrală a semnalului (frecvența postului radio în cazul receptorului FM)
- Y per Div – pasul axei Y dat în dB
- Y Divs – numărul intervalelor de lungime Y per Divs pe axa Y
- Ref Level (dB) – nivelul de referință al semnalului
- Ref Scale (p2p) – se folosește pentru scalarea semnalului afișat
- FFT Size – numărul de puncte FFT
- Refresh Rate – rata de actualizare a diagramei
- Peak Hold – suprapune valorile maxime cu spectrul curent
- Average – suprapune valorile medii cu spectrul curent
- Window Size, Grid Position, Notebook Param – identic cu parametrii osciloscopului

După pornirea simulării este de asemenea posibilă configurarea unor parametri ai analizorului spectral din interfața grafică.

### ***WX GUI Histo Sink***

Se folosește pentru afișarea histogramei eșantioanelor. Permite configurarea numărului de puncte de afișat și numărul de bin-uri. Axele X și Y se scalează automat.

### **Experimente**

1. Creați modelul pentru receptorul radio FM folosind GNU Radio Companion și PlutoSDR. Setati frecvența postului radio pe postul vostru favorit.
2. Extindeți implementarea folosind variabile, astfel încât nici un bloc să nu conțină parametrii configurați static. Adăugați slider pentru controlul frecvenței postului radio și al volumului.
3. Extindeți implementarea cu osciloscop și vizualizați semnalul recepționat procesat. Interpretați rezultatul.
4. Înlocuiți osciloscopul cu analizor spectral, interpretați rezultatul.
5. Înlocuiți analizorul spectral cu histogramă, interpretați rezultatul.
6. Adăugați osciloscop la ieșirea receptorului PlutoSDR. Interpretați rezultatul.
7. Înlocuiți osciloscopul cu analizor spectral, interpretați rezultatul.
8. Înlocuiți analizorul spectral cu histogramă, interpretați rezultatul.
9. Folosind blocurile de mai sus comparați semnalele de la ieșirea PlutoSDR și de la intrarea Audio Sink. Interpretați diferențele.
10. Schimbați tipul diagramei la QT GUI și folosiți blocurile QT GUI Constellation Sink, QT GUI Waterfall Sink, QT GUI Frequency Sink, QT GUI Histogram Sink și QT GUI Time Sink pentru analiza semnalelor.