T126

Rádio Definido por Software

Aula #01
Introdução

O que é Rádio Definido por Software?

Rádio Definido por Software (RDS) é um sistema de radiocomunicação onde componentes tipicamente implementados em hardware (mixers, filtros, amplificadores, moduladores/demoduladores, detectores, etc) são implementados em software, em um PC ou sistema embarcado.

Fonte: Wikipedia - Software-Defined Radio

Vantagens

- Habilidade de receber e transmitir sinais com várias modulações usando um mesmo conjunto de hardware;
- Habilidade de alterar a funcionalidade do rádio fazendo o download e execução de um novo software;
- Possibilidade de escolher adaptativamente uma frequência de operação ou modo de operação de acordo com as condições do sistema;
- Oportunidade de reconhecer e evitar interferências de outros canais de comunicação;
- Eliminação de hardware analógico, reduzindo custos e simplificando a arquitetura do sistema;
- Facilidade e flexibilidade de prototipação e realização de experimentos.

Desvantagens

- Dificuldade na escrita de software para várias arquiteturas e plataformas;
- Necessidade de interfaces para sinais digitais e algoritmos;
- Faixa dinâmica reduzida para a grande gama de frequências de interesse em alguns projetos de dispositivos para RDS;
- Alta complexidade dos projetos de software e hardware é o preço pago pela grande flexibilidade no uso do RDS.

Histórico

1991 - Joseph Mitola III

Primeiro uso do termo Rádio Definido por Software.

Fonte: Software Radios: Survey, Critical Evaluation and Future Directions

Histórico

1990 até 1995
Projeto do rádio SpeakEasy II
(US Air Force Research Laboratory)

Primeiro projeto a usar um dispositivo FPGA para processamento digital de dados de rádio.

Emulava mais de 15 rádios militares existentes na época.

Fonte: SpeakEasy: The Military Software Radio

O uso de RDS

- Sistemas de radiocomunicação
- Sistemas de celular 2G, 3G, 4G e 5G
- Sistemas de TV Digital (ISDB-T): Link
- Sistemas de satélite: Link
- Sistemas de rádio amador: Link
- Sistemas de rádio digital (DAB): Link
- Receptor de GPS: Link
- Repositório de módulos não oficiais do GNU Radio: Link
- Sistemas de monitoramento aéreo (ADS-B): Link 01, Link 02
- Wi-Fi: Link
- Rádio AM/FM: Link
- LoRa: Link
- Emulação de instrumentos de medida
- Radar doppler: Link

Modelos de RDS

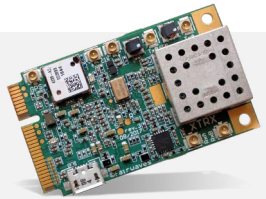
Analog Devices

ADALM-PLUTO SDR

325 MHz até 3,8 GHz

20 MHz BW (USB 2.0)

US$ 149



Fairwaves

XTRX Pro

30 MHz até 3,7 GHz

120 MHz BW (mPCIe)

US$ 599

Myriad RF

LimeSDR Pcie

100 kHz até 3,8 GHz

61,44 MHz BW (PCIe)

US$ 799

Ettus Research

USRP B210

70 MHz até 6 GHz

56 MHz BW (USB 3.0)

US$ 1.259

NI (Ettus)

USRP 2954 (X310)

10 MHz até 6 GHz

160 MHz BW (PCIe)

US$ 5.612

NI (Ettus)

USRP 2974 (USRP 2974)

10 MHz até 6 GHz

160 MHz BW (integrado)

US$ 13.455

Ferramentas e linguagens

VHDL

NATIONAL INSTRUMENTS LabVIEW

C/C++

MATLAB SIMULINK

python™

GNURadio
THE FREE & OPEN SOFTWARE RADIO ECOSYSTEM

Ferramentas e linguagens

Analog Devices

ADALM-PLUTO SDR

325 MHz até 3,8 GHz

20 MHz BW (USB 2.0)

US$ 149

+

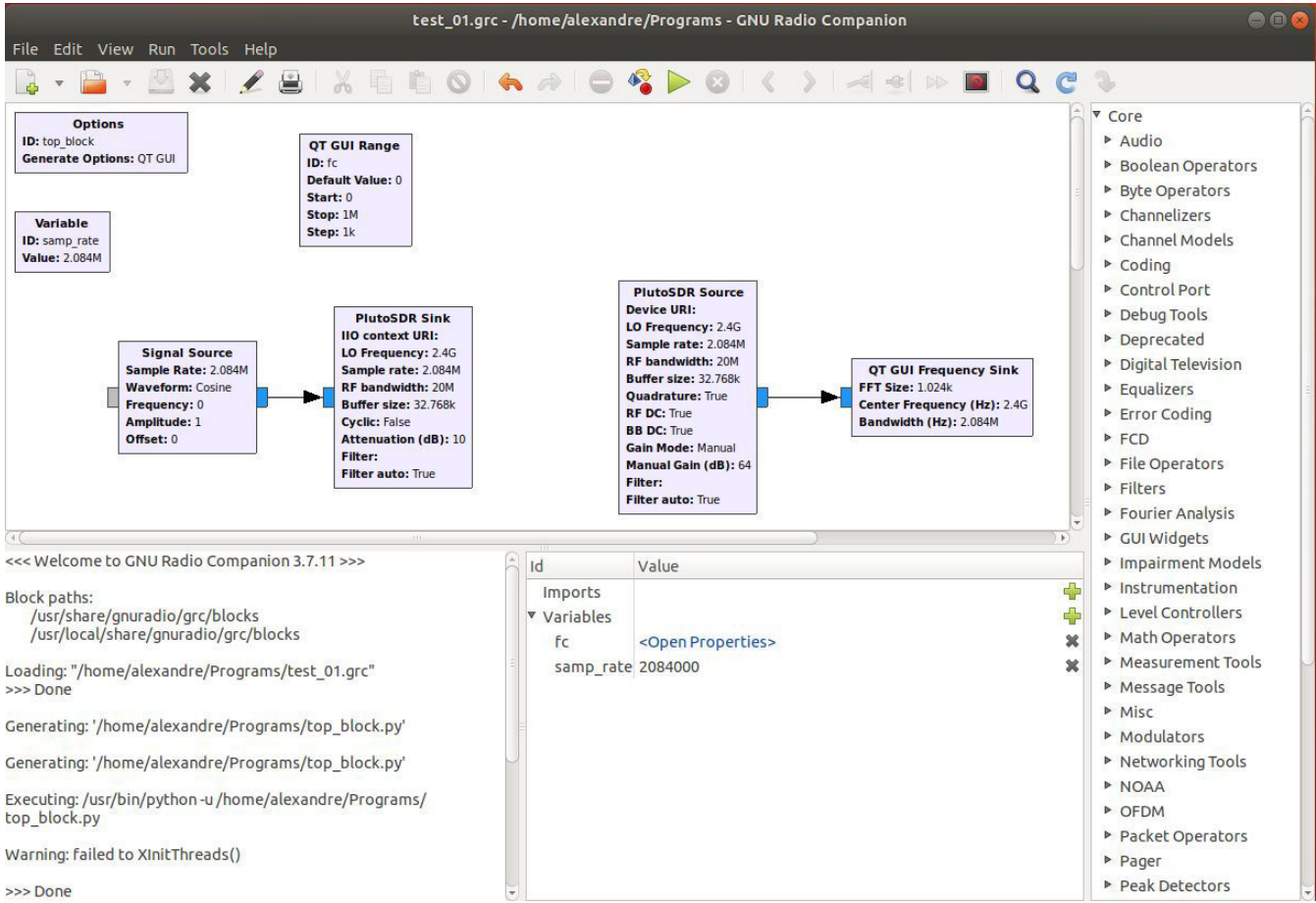
GNURadio

THE FREE & OPEN SOFTWARE RADIO ECOSYSTEM

python™

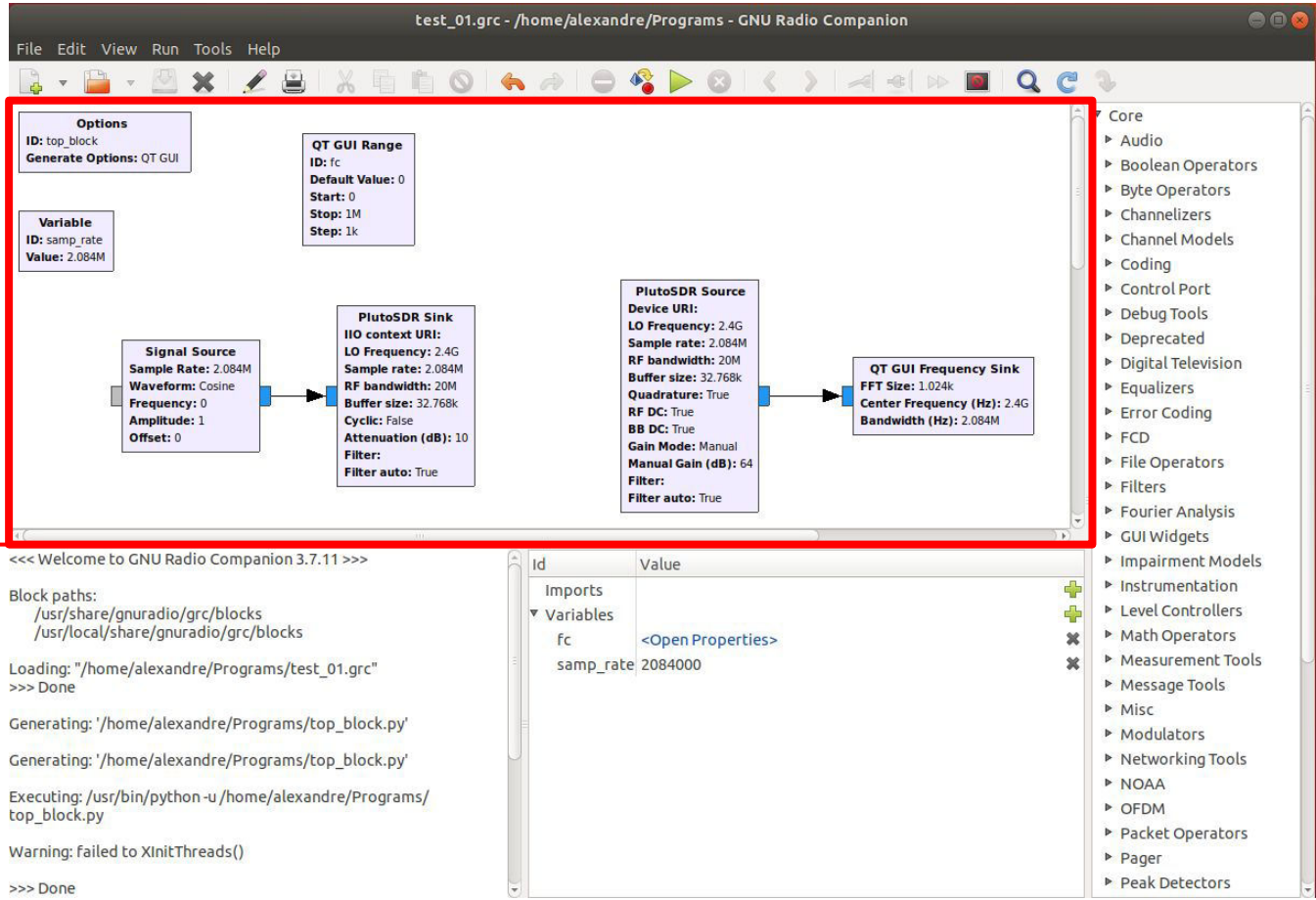
C/C++

GNU Radio e sua interface



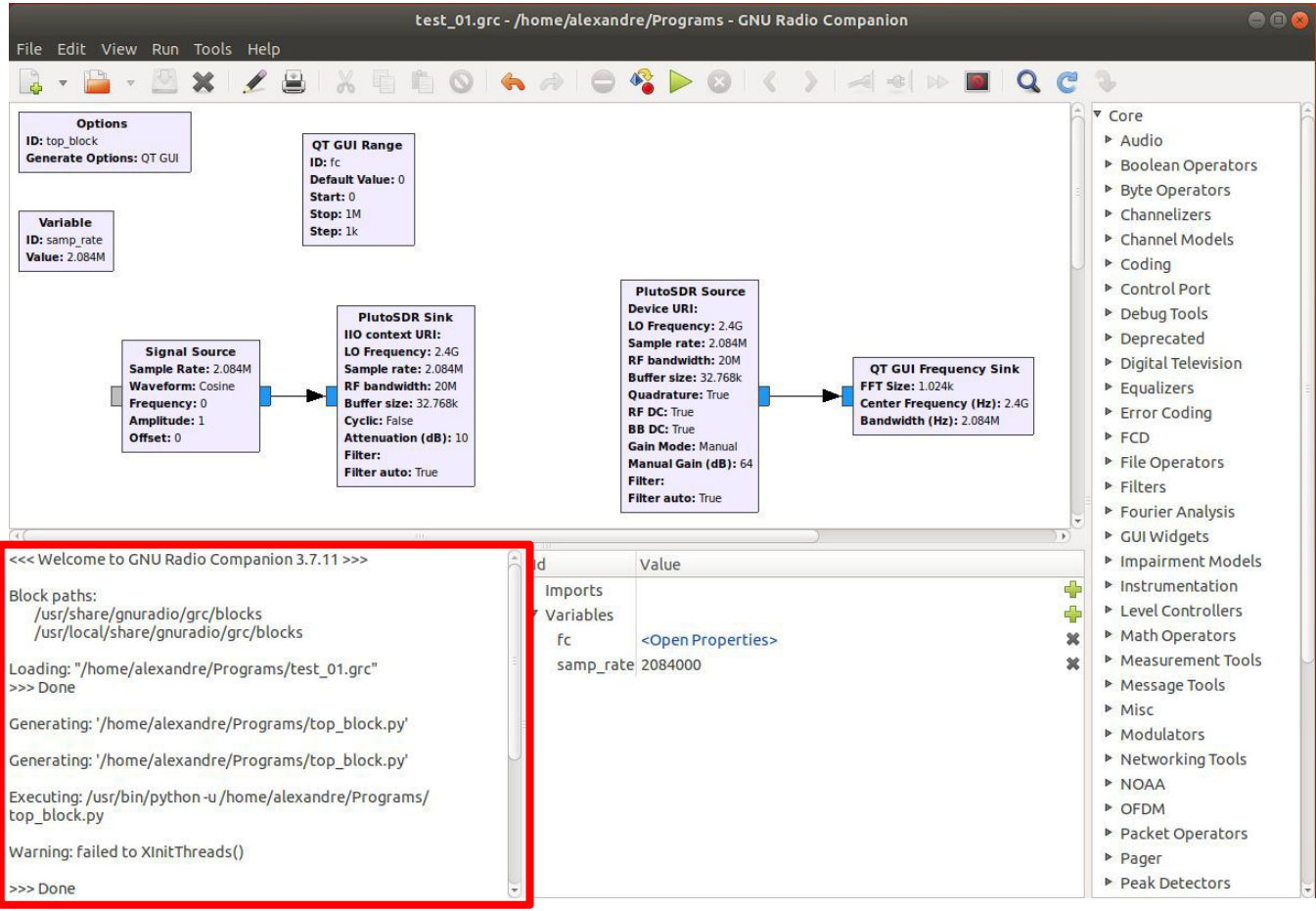
GNU Radio e sua interface

Flow graph



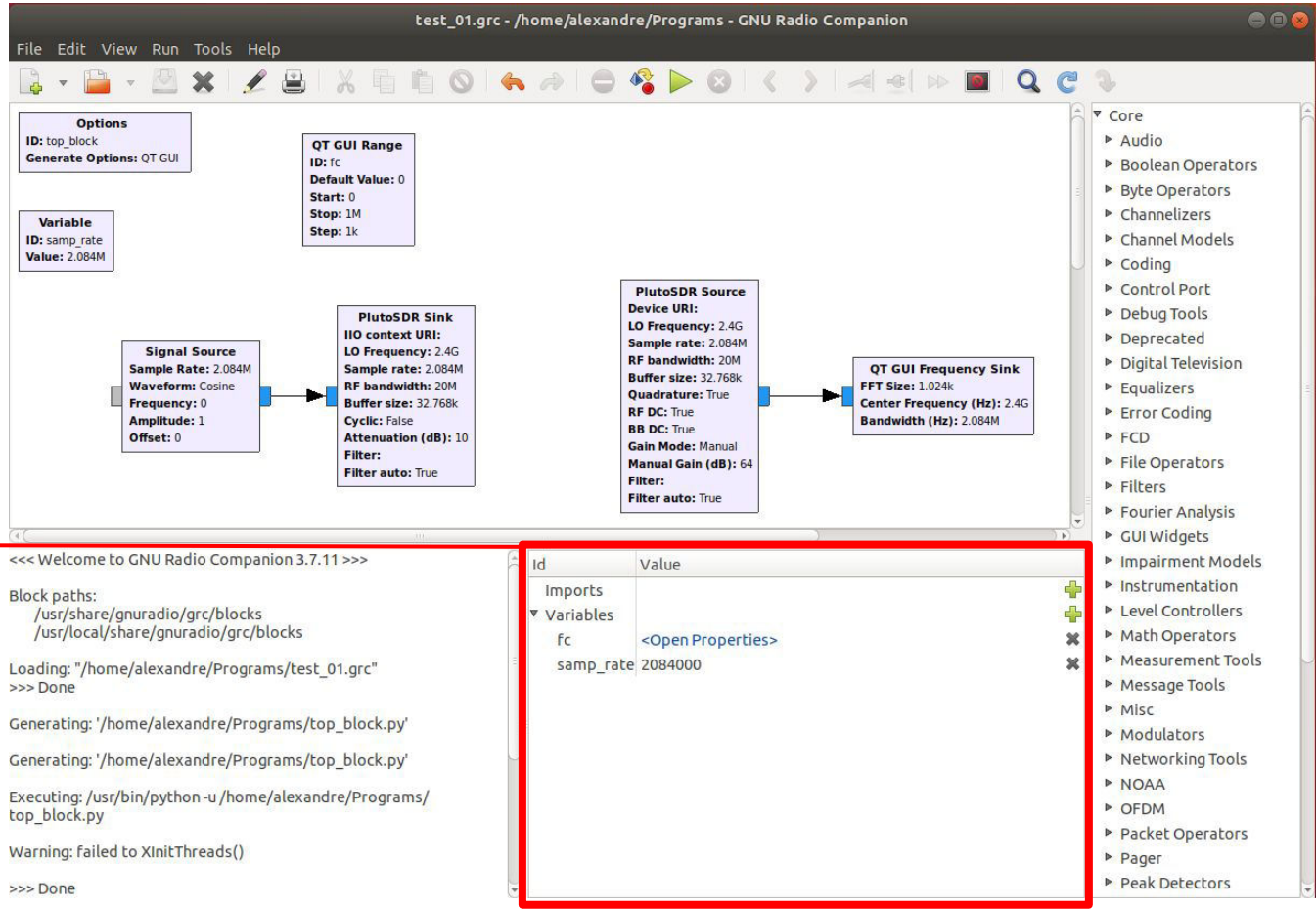
GNU Radio e sua interface

Console Panel



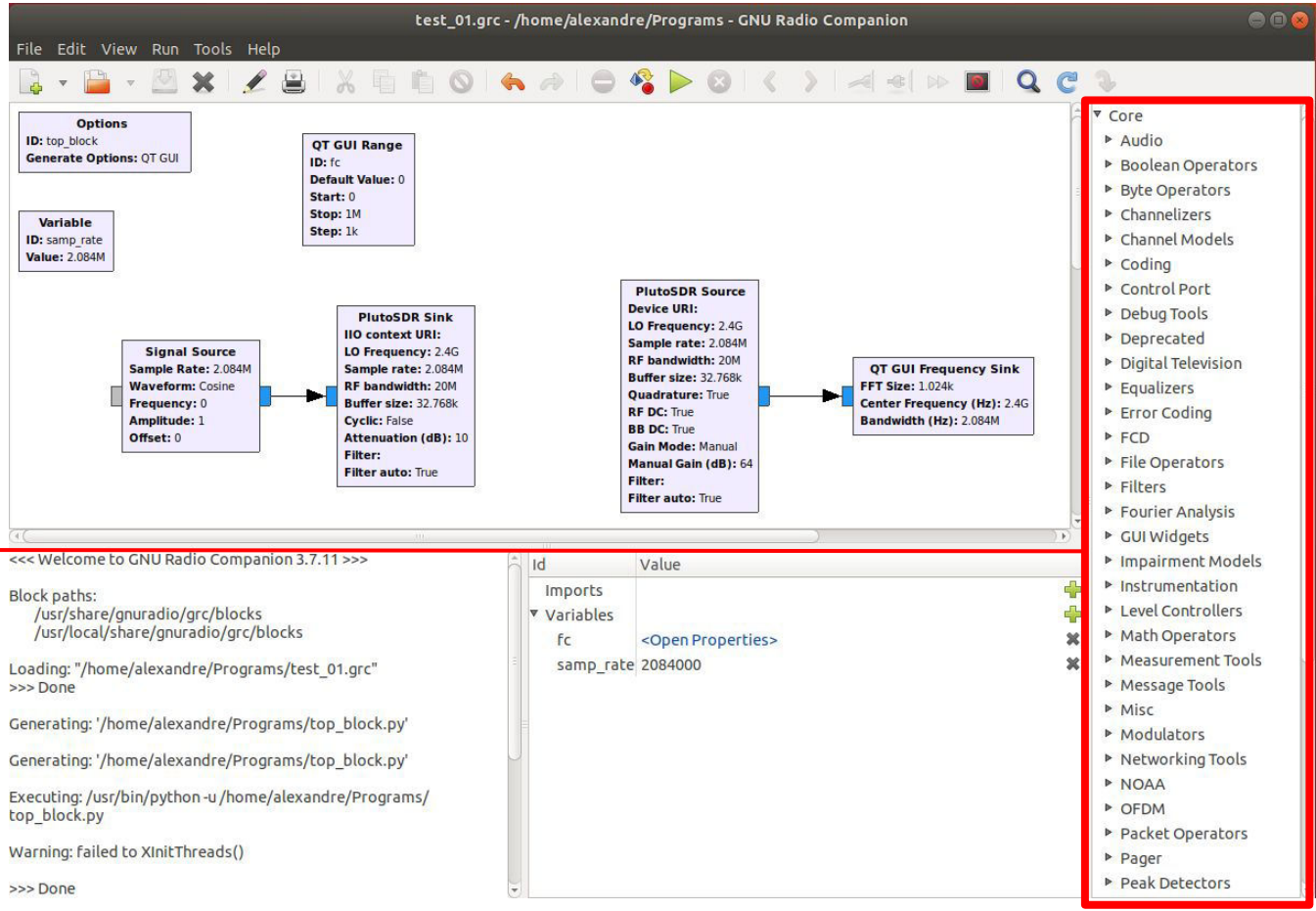
GNU Radio e sua interface

Variable Editor

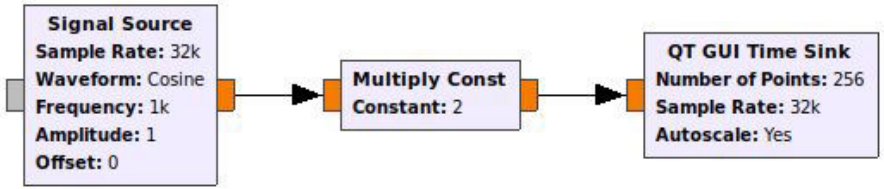


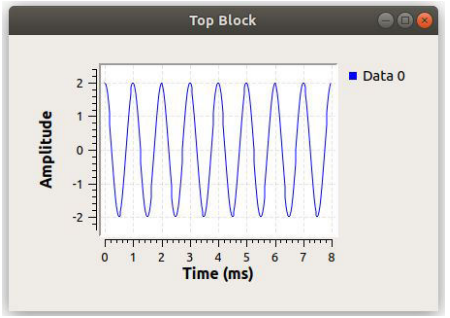
GNU Radio e sua interface

Block Tree Panel



Fluxo de dados entre blocos



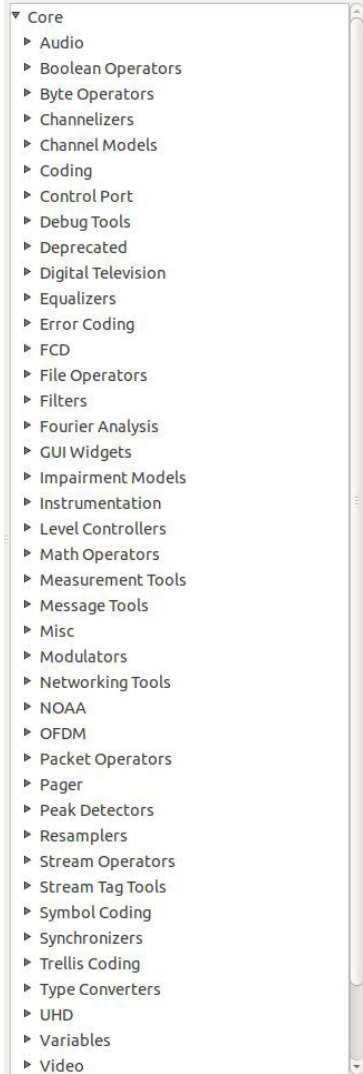


- Cada bloco no GNU Radio tem sua própria *thread* no sistema operacional.
  Nesse caso, os blocos se comportam como pequenos programas sendo executados de forma independente no sistema operacional;
- Cada bloco no GNU Radio possui um *buffer* circular de saída onde são armazenados os dados gerados ou processados pelo bloco.
  No caso do projeto acima, os blocos Signal Source e Multiply Const possuem esse *buffer* de saída;
- Ao executar um *flow graph*, o *scheduler* do GNU Radio informa continuamente a cada uma das *threads* quanto há de espaço nos *buffers* de saída;
- Enquanto houver espaço nos *buffers* de saída, os blocos produzem ou processam dados para completar o *buffer*.
  O bloco subsequente na cadeia é notificado quando há dados a serem consumidos no *buffer* de saída do bloco anterior.

Fonte: Behind the Veil: A Peek at GNU Radio's Buffer Architecture

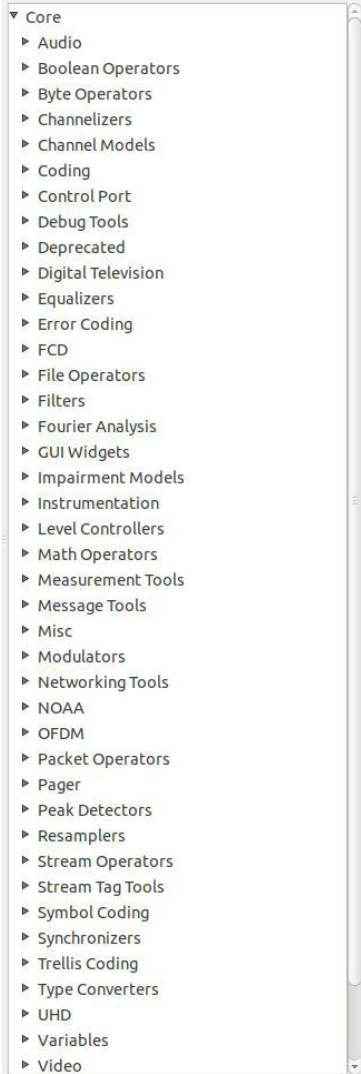
Biblioteca de blocos do GNU Radio

- A instalação padrão do GNU Radio já vem com uma série de blocos pré instalados e oficialmente suportados. Estes blocos podem ser encontrados no *Block Tree Panel*, conforme a figura ao lado;
- A documentação para estes blocos pode ser encontrada no site do GNU Radio;
- A lista de blocos pré instalados varia de acordo com a versão do GNU Radio instalada.



Adição de novos blocos à biblioteca

- O GNU Radio permite a adição de novos blocos ou módulos de blocos à sua biblioteca. Eles são chamados de módulos *Out-Of-Tree* (OOT);
- É possível importar um módulo OOT já pronto, não oficialmente suportado pelo GNU Radio, obtido na internet;
- Existe também a possibilidade de criação do seu próprio módulo customizado que poderá ser incorporado à biblioteca do GNU Radio e distribuído para outros usuários;
- Os blocos para o módulo OOT customizado podem ser construídos usando duas linguagens de programação: Python e C/C++. As vantagens e desvantagens no uso destas linguagens serão discutidas mais à frente na disciplina.



Analog Devices Pluto SDR

Specifications	Typical
Power	
DC Input (USB)	4.5 V to 5.5 V
Conversion Performance and Clocks	
ADC and DAC Sample Rate	65.2 kSPS to 61.44 MSPS
ADC and DAC Resolution	12 bits
Frequency Accuracy	±25 ppm
RF Performance	
Tuning Range	325 MHz to 3800 MHz
Tx Power Output	7 dBm
Rx Noise Figure	<3.5 dB
Rx and Tx Modulation Accuracy (EVM)	–34 dB (2%)
RF Shielding	None
Digital	
USB	2.0 On-the-Go
Core	Single ARM Cortex®-A9 @ 667 MHz
FPGA Logic Cells	28k
DSP Slices	80
DDR3L	4 Gb (512 MB)
QSPI Flash	256 Mb (32 MB)
Physical	
Dimensions	117 mm × 79 mm × 24 mm 4.62" × 3.11" × 0.95"
Weight	114 g
Temperature	10°C to 40°C

Fonte: ADALM-PLUTO SDR - Product Highlight

- Qual o porquê do nome Pluto? É uma referência ao corpo celestial Plutão, um planeta anão, que não possui certos critérios técnicos para ser considerado um planeta. Da mesma maneira, o PLUTO é um dispositivo destinado ao aprendizado, não possuindo alguns critérios técnicos e de desempenho para ser equiparado a um RDS comercial;

Fonte: Analog Devices - Wiki

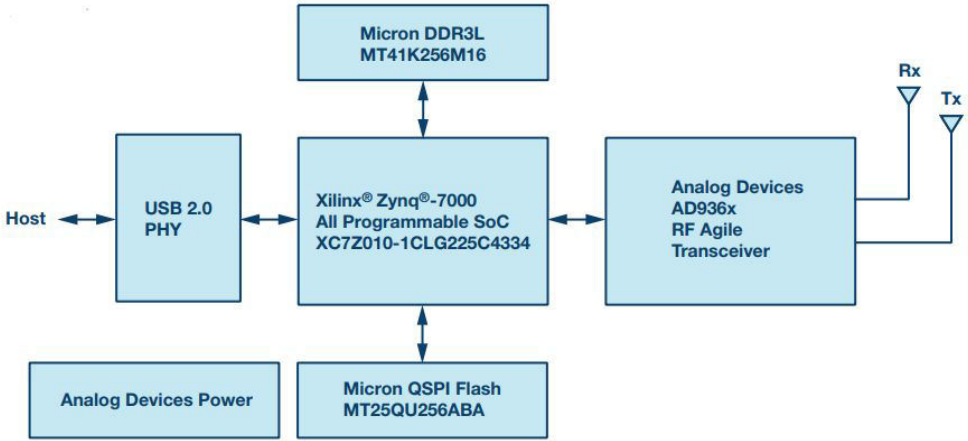
AD9363

- Transceptor 2x2
- 12-bit DACs and ADCs
- Banda: até 20 MHz
- Frequência: 325 MHz até 3,8 GHz

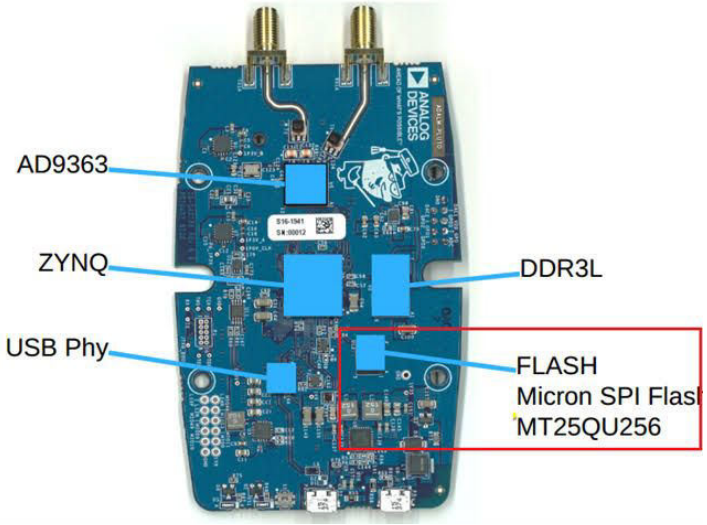
AD9361

- Transceptor 2x2
- 12-bit DACs and ADCs
- Banda: até 56 MHz
- Frequência: 70 MHz até 6 GHz

Analog Devices Pluto SDR

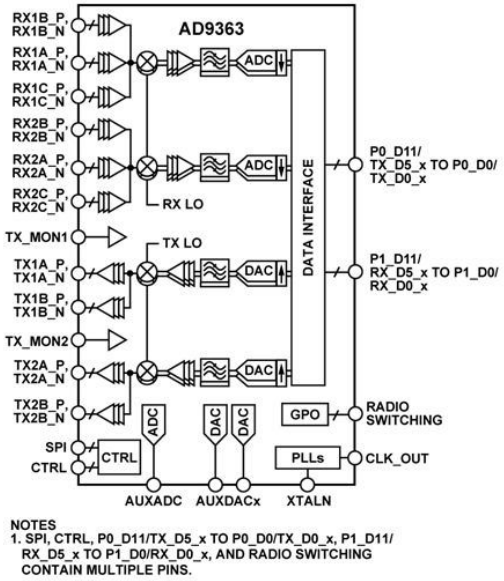


Fonte: ADALM-PLUTO SDR - Product Highlight

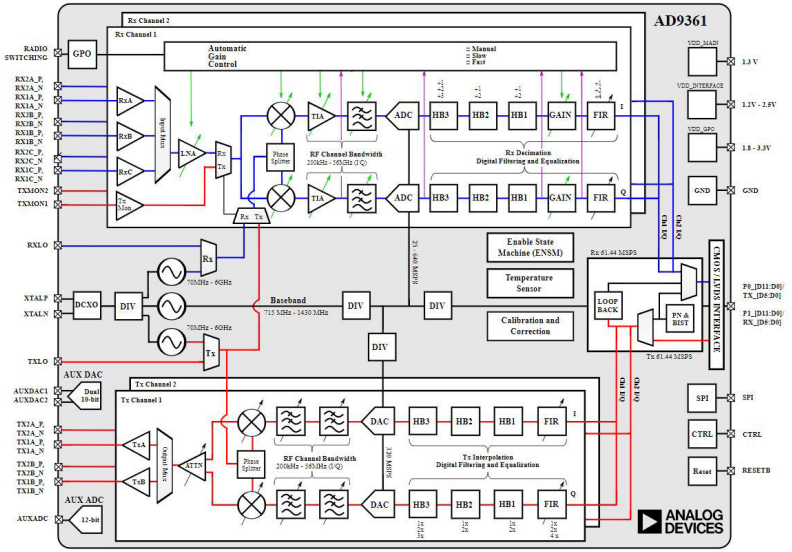


Fonte: Analog Devices - Engineer Zone

Analog Devices Pluto SDR



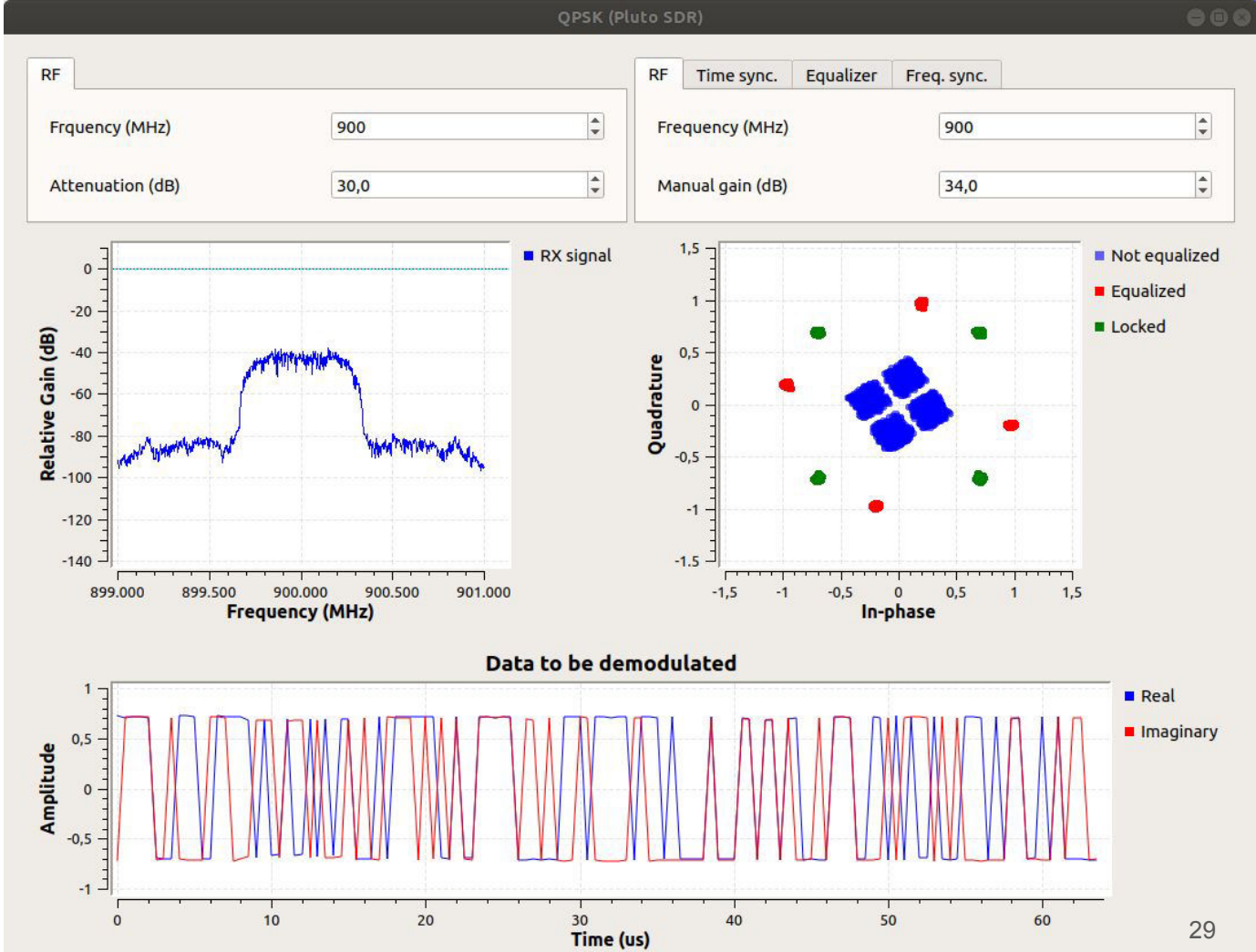
Fonte: [Analog Devices - AD9363 Data Sheet]



Fonte: [Analog Devices - Wiki AD9361, AD9364 and AD9363]

Transceptor QPSK

- Transmissor e receptor em portadora única com modulação QPSK;
- Projeto a ser desenvolvido no curso.



Transmissor de TV Digital ISDB-Tb

- Link do projeto: Link
- Instalação:

```
git clone https://github.com/git-artes/gr-isdbt.git
cd gr-isdbt
mkdir build
cd build
cmake ../
make && sudo make install
sudo ldconfig
```

- Para usar o VOLK, executar:

```
volk_profile
```

- Download de arquivos de vídeo (.ts): Link
- Lista de canais de TV Digital no Brasil: Link

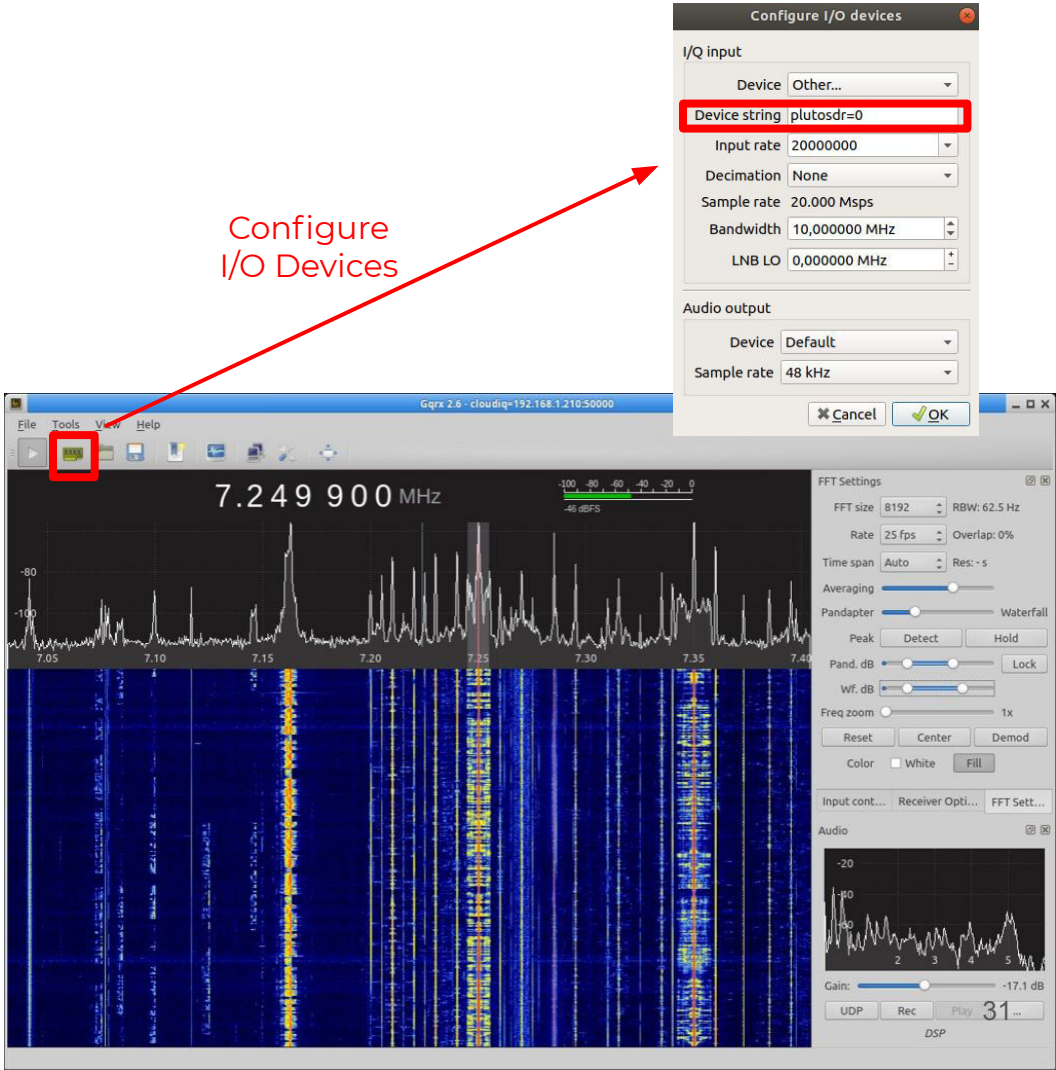
Instrumento de medida: receptor Gqrx

- Link do projeto: Gqrx
- Instalação:

```
git clone https://github.com/csete/gr-osmosdr-gqrx
cd gr-osmosdr-gqrx/
git checkout plutosdr
mkdir build
cd build/
cmake ../
make
sudo make install
sudo ldconfig

git clone https://github.com/csete/gqrx.git gqrx.git
cd gqrx.git
mkdir build
cd build
cmake ..
make
sudo make install
sudo ldconfig
```

Configure
I/O Devices

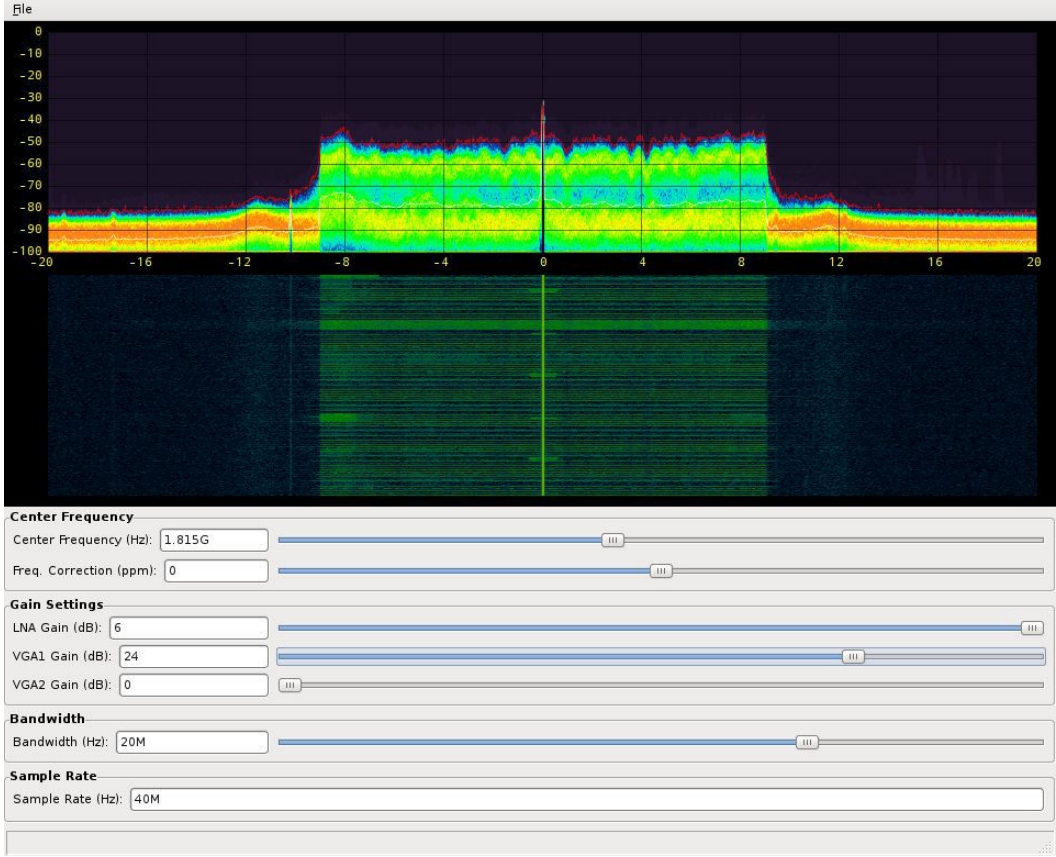


Instrumento de medida: receptor Fosphor

- Link do projeto: Fosphor
- Características
  - Alto desempenho (tempo real);
  - Utiliza a GPU do PC;
  - Baseado em OpenCL e OpenGL;
- Verificar compatibilidade do PC com OpenCL:

```
lspci | grep VGA
```

- De acordo com a Wikipedia, processadores Haswell (HSW-ULT) podem utilizar OpenCL versão 1.2.



Instrumento de medida: receptor Fosphor

- Instalação de dependências:

```
sudo apt install ocl-icd-* opencl-headers
cmake xorg-dev libglu1-mesa-dev
```

- Procurar por dispositivo OpenCL no PC:

```
sudo apt install clinfo
clinfo
```

- Instalar pacotes para o OpenCL:

```
sudo apt install beignet
```

- Instalação e solução para problema de compartilhamento OpenCL e OpenGL (link):

```
sudo apt install qt4-default
sudo apt install libglfw3-dev

git clone git://git.osmocom.org/gr-fosphor
cd gr-fosphor
git checkout 7b6b9961bc2d9b84daeb42a5c8f8aeba293d207c
```

Procurar pelo arquivo `lib/fosphor/cl.c` e no início da função `cl_do_init` comentar a linha `self->flags |= FLG_FOSPHOR_USE_CLGL_SHARING ;`

```
mkdir build
cd build
cmake -DCMAKE_INSTALL_PREFIX=/usr ..
```

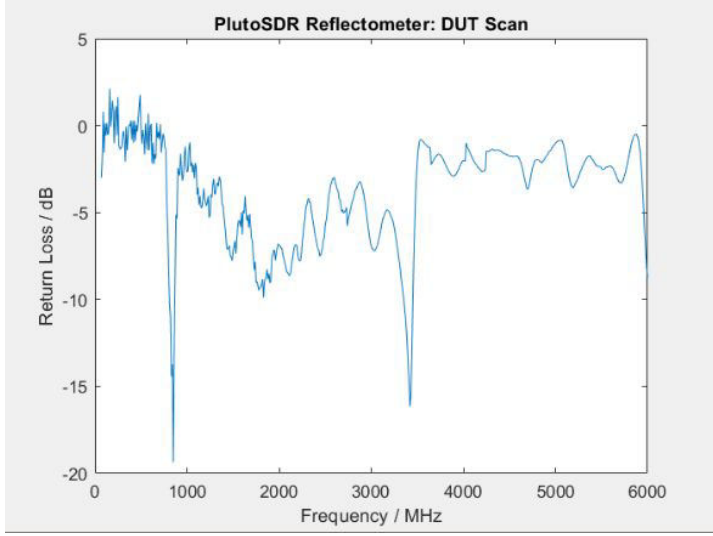
Confirmar que qt e glfw estejam adicionados na compilação.

```
make
sudo make install
sudo ldconfig
```

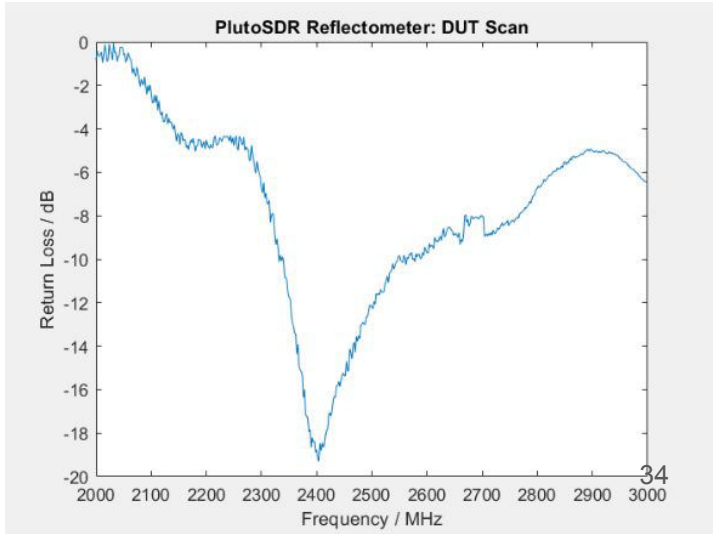
Instrumento de medida: analisador de sinais

- Link do projeto: Link
- Medida de Return Loss vs Frequency

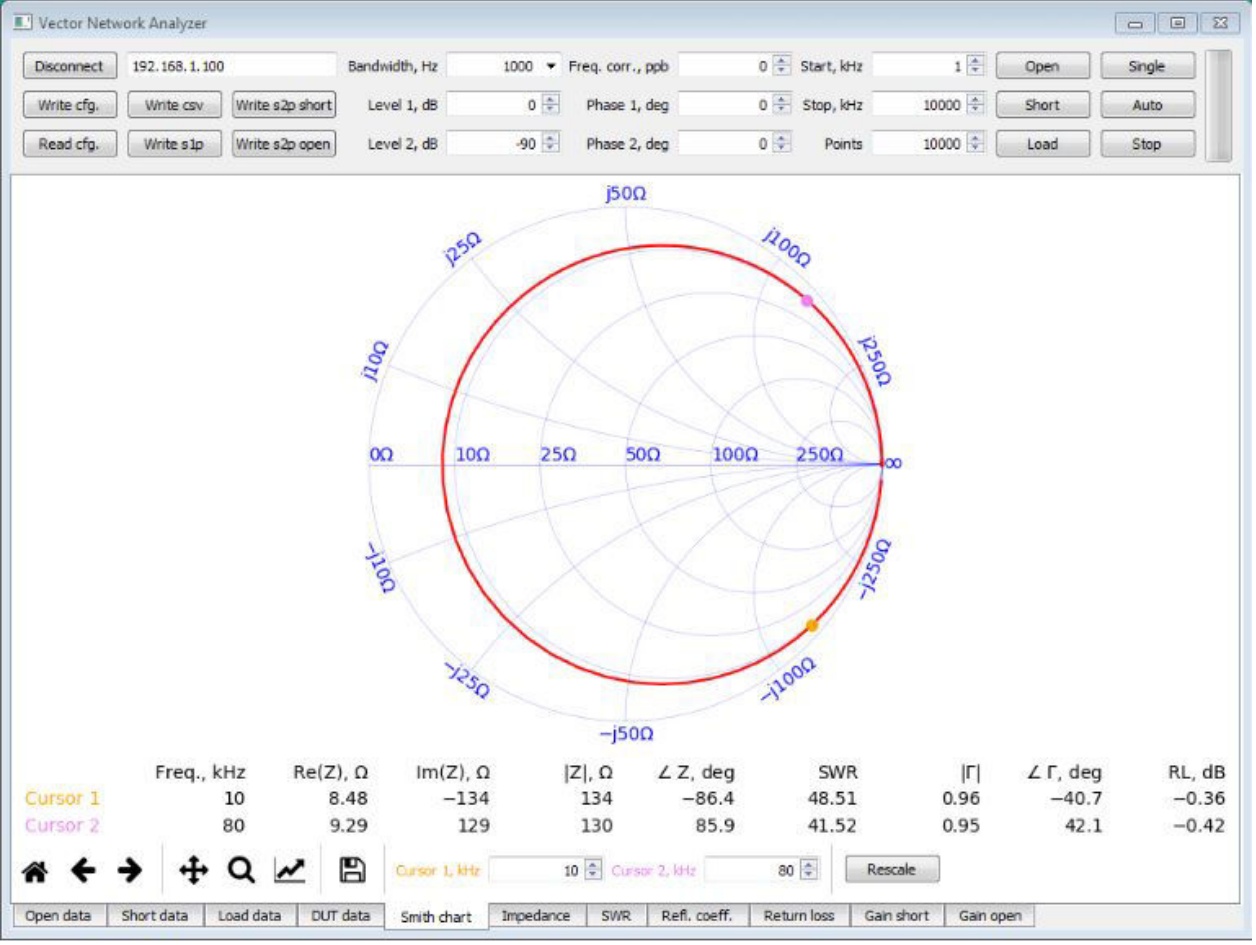
Antena do kit da Pluto SDR



Antena de Wi-Fi



Instrumento de medida: analisador de sinais



- Link do projeto: [Link]
- Vector Network Analyzer

Atividade extraclasse

- Familiarização com a linguagem Python e o pacote NumPy;
- Tutorial de Python: Link;
- Tutoriais de NumPy: Link 1, link 2, link 3.