T126

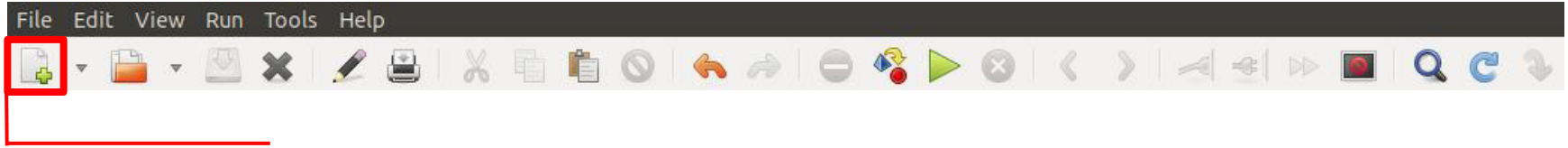
Rádio Definido por Software

Aula #03

Implementação de um novo bloco no GNU Radio

Utilizando blocos da árvore

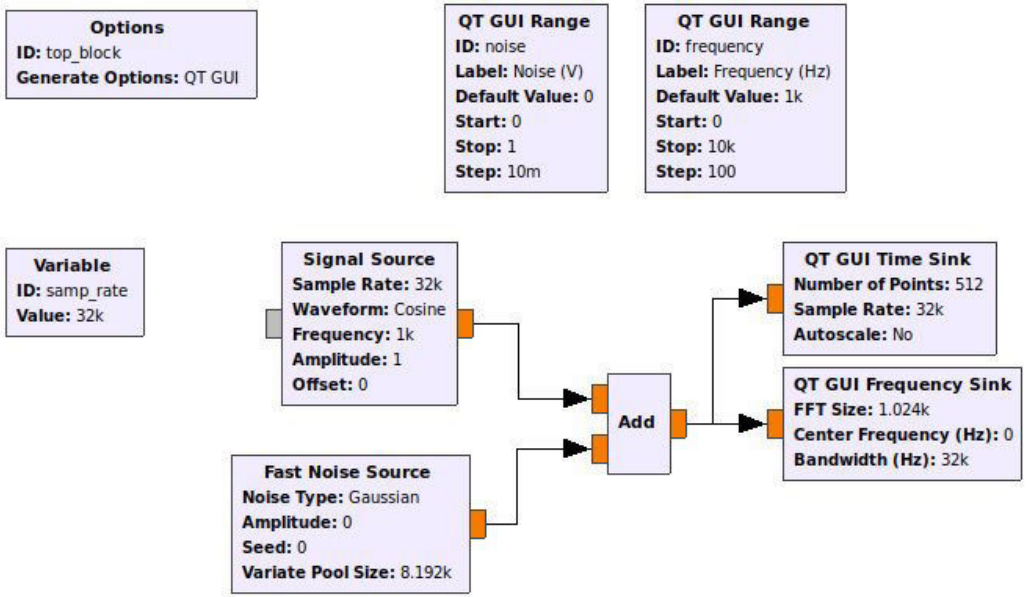
- Para criar um novo *flow graph*, pressione o botão *Create a new flow graph* na barra de ícones:



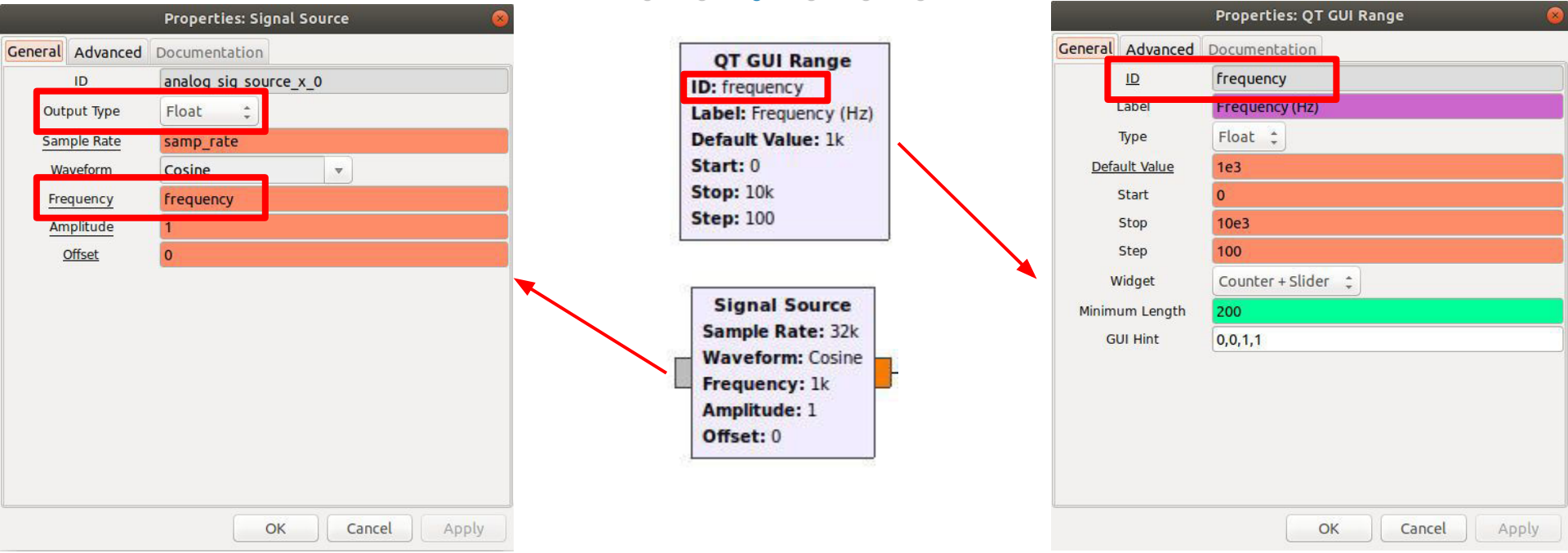
- Salvar o *flow graph* como *first_contact_with_gnuradio* em:

```
/Home/Programs
```

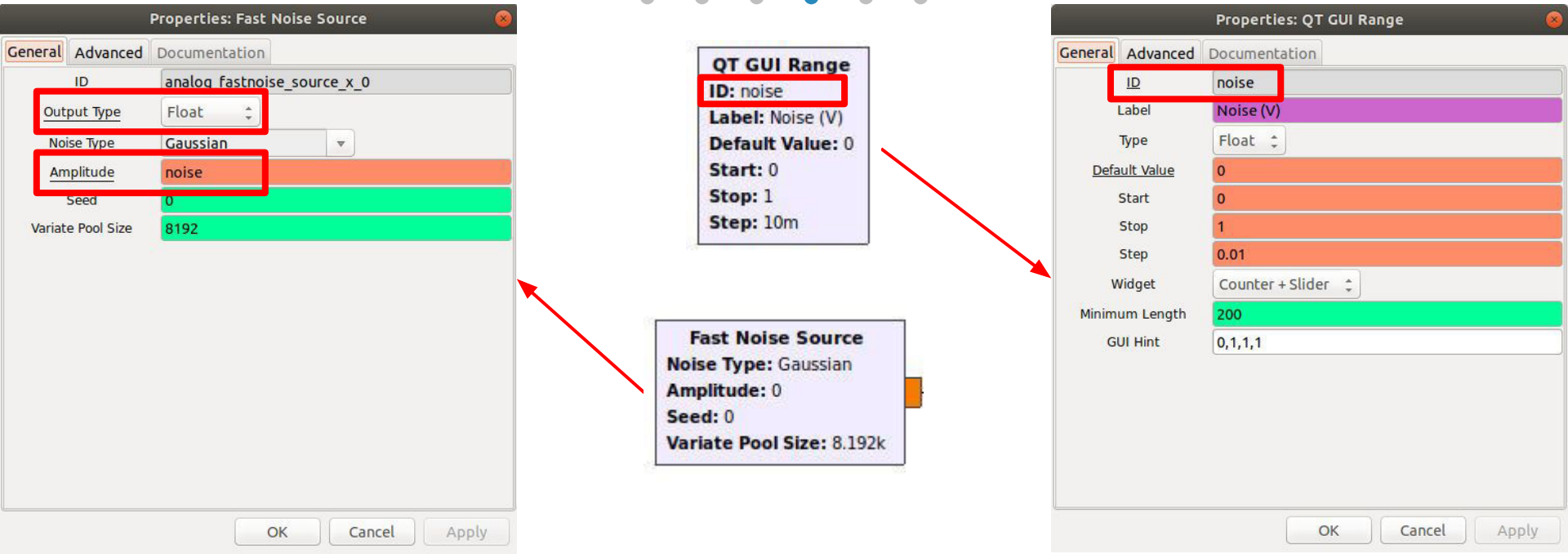
Utilizando blocos da árvore



Utilizando blocos da árvore

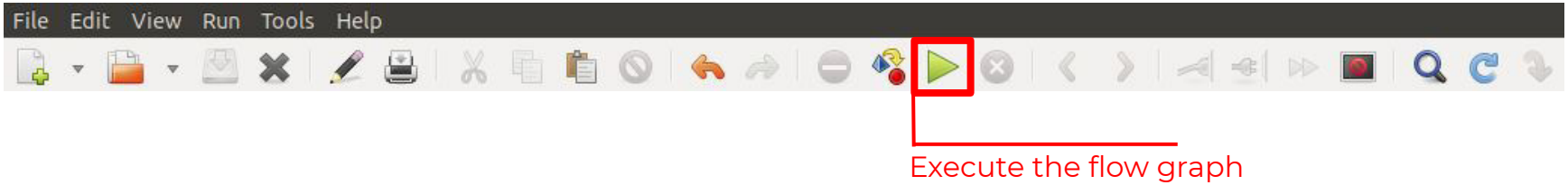


Utilizando blocos da árvore

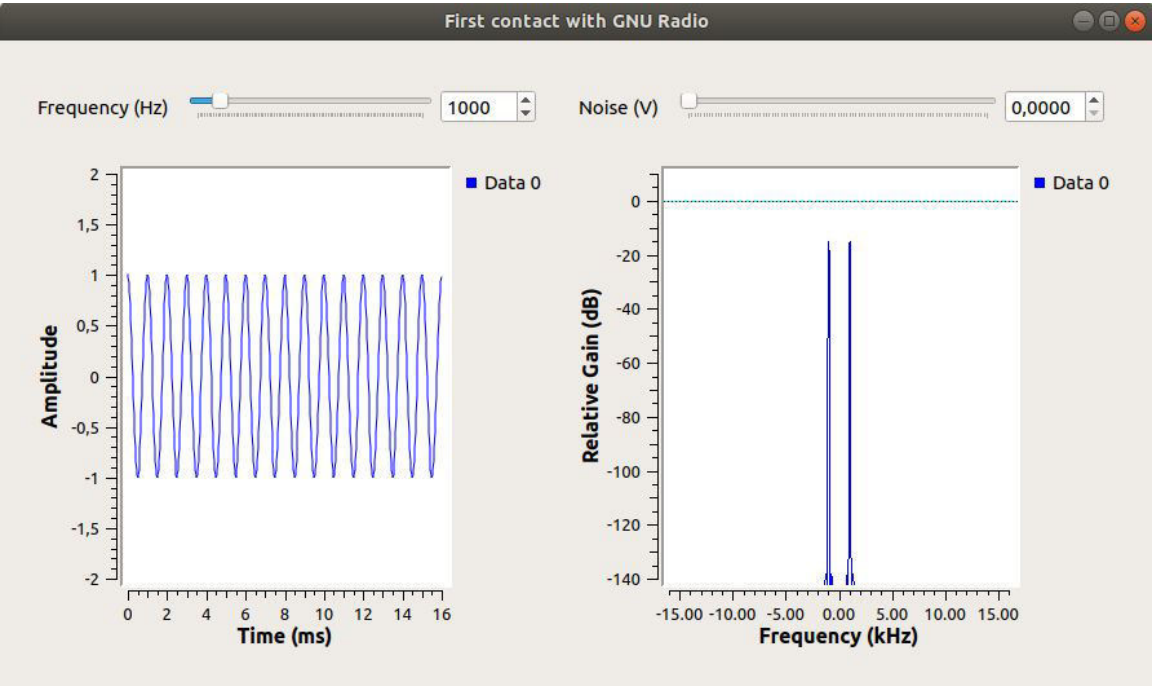


Utilizando blocos da árvore

- Para executar o *flow graph*, pressione *Execute the flow graph* na barra de ícones:

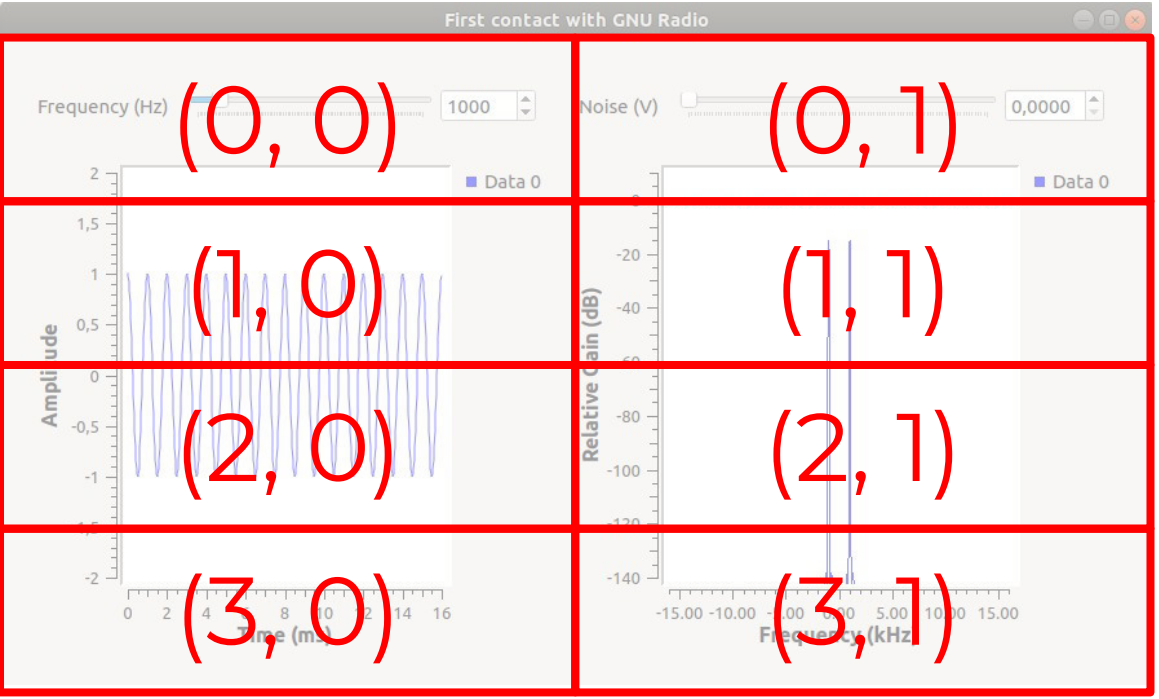


Utilizando blocos da árvore



- O projeto em execução deve ser parecido com o da figura ao lado.

Utilizando o campo *GUI Hint* para layout



- Organização do layout dos componentes na GUI (*Graphical User Interface*);
- (*row, column, row span, column span*);
- QT GUI Range (freq.): (0,0,1,1)
- QT GUI Range (noise): (0,1,1,1)
- QT GUI Time Sink: (1,0,3,1)
- QT GUI Frequency Sink (1,1,3,1)

Como criar um bloco *Out-of-Tree* (OOT)?

Existe um *script* do GNU Radio que auxilia a criação de um novo módulo ou bloco:

`gr_modtool`

Principais comandos:

- `gr_modtool newmod name_of_module` (cria um novo módulo com o nome `gr-name_of_module`);
- `gr_modtool add -t block_type -l language` (adiciona um novo bloco ao módulo)
  - `block_type: sink, source, sync, decimation, interpolation, general, tagged_stream, hier, noblock`
  - `language: cpp, python`

Como criar um bloco *Out-of-Tree* (OOT)?

Help do comando `gr_modtool`

Digitar no console: `gr_modtool help`

```
alexandre@note-linux:~/Programs/gr-t126$ gr_modtool help
Usage:
gr_modtool <command> [options] -- Run <command> with the given options.
gr_modtool help -- Show a list of commands.
gr_modtool help <command> -- Shows the help for a given command.

List of possible commands:

Name      Aliases          Description
==========================================================================
disable   dis              Disable block (comments out CMake entries for files)
info      getinfo,inf      Return information about a given module
remove    rm,del           Remove block (delete files and remove Makefile entries)
makexml   mx               Make XML file for GRC block bindings
add       insert           Add block to the out-of-tree module.
newmod    nm,create        Create a new out-of-tree module
rename    mv               Rename a block in the out-of-tree module.
```

Criação de um novo módulo

Vamos criar um módulo chamado `t126` para ser usado durante a disciplina:

1) Abrir o terminal no Ubuntu: Ctrl + Alt + t;
2) Ir até a pasta onde se deseja criar o módulo;
   ```
   cd ~/Programs
   ```
3) Criar o módulo `t126` utilizando a ferramenta `gr_modtool`;
   ```
   gr_modtool newmod t126
   ```
4) Verificar se a pasta do módulo (`gr-t126`) foi mesmo criada;
   ```
   ls
   ```
5) Verificar o conteúdo da pasta `gr-t126` (a pasta deve conter os arquivos do módulo criado)
   ```
   cd gr-t126
   ls
   ```

```
alexandre@note-linux:~/Programs/gr-t126$ ls
apps  cmake  CMakeLists.txt  docs  examples  grc  include  lib  MANIFEST.md  python  swig
```

Criação de um novo bloco

Vamos criar um bloco do tipo `sync` utilizando a linguagem Python:

1) Entrar na pasta do módulo criado anteriormente (caso já não esteja nela);

```
cd gr-t126
```

2) Adicionar o novo bloco;

```
gr_modtool add -t sync -l python
```

3) Entrar com o nome do bloco;

```
mapper
```

Criação de um novo bloco

4) Entrar com a lista de variáveis de entrada do bloco;

   Sem variáveis para este bloco. Apenas pressione a tecla `Enter`.

5) Adicionar QA code em Python;

   `Y`

6) Os arquivos do bloco são criados como na figura ao lado.

```
alexandre@note-linux:~/Programs/gr-t126$ gr_modtool add -t sync -l python
GNU Radio module name identified: t126
Language: Python
Enter name of block/code (without module name prefix): mapper
Block/code identifier: mapper
Enter valid argument list, including default arguments:
Add Python QA code? [Y/n] Y
Adding file 'python/mapper.py'...
Adding file 'python/qa_mapper.py'...
Editing python/CMakeLists.txt...
Adding file 'grc/t126_mapper.xml'...
Editing grc/CMakeLists.txt...
alexandre@note-linux:~/Programs/gr-t126$
```

Remoção de um bloco do módulo

1) Entrar na pasta do módulo criado (caso já não esteja nela);

```
cd gr-t126
```

2) Executar o comando de remoção;

```
gr_modtool remove
```

3) Digitar o nome do bloco a ser removido;

```
mapper
```

4) Os arquivos criados junto com o bloco são apagados.

Importante: antes de remover um bloco do módulo, certificar-se de que o módulo foi desinstalado do GNU Radio anteriormente.

```
alexandre@note-linux:~/Programs/gr-t126$ gr_modtool rm
GNU Radio module name identified: t126
Which blocks do you want to delete? (Regex): mapper
Searching for matching files in lib/:
None found.
Searching for matching files in include/t126/:
None found.
Searching for matching files in swig/:
None found.
Searching for matching files in python/:
Really delete python/qa_mapper.py? [Y/n/a/q]: Y
Deleting python/qa_mapper.py.
Deleting occurrences of qa_mapper.py from python/CMakeLists.txt...
Really delete python/mapper.py? [Y/n/a/q]: Y
Deleting python/mapper.py.
Deleting occurrences of mapper.py from python/CMakeLists.txt...
Searching for matching files in grc/:
Really delete grc/t126_mapper.xml? [Y/n/a/q]: Y
Deleting grc/t126_mapper.xml.
Deleting occurrences of t126_mapper.xml from grc/CMakeLists.txt...
```

Edição de arquivos do bloco

Vamos instalar um editor de texto (sugestão: Microsoft Visual Studio Code)

1) Instalar dependências;

```
sudo snap install --classic code
```

ou

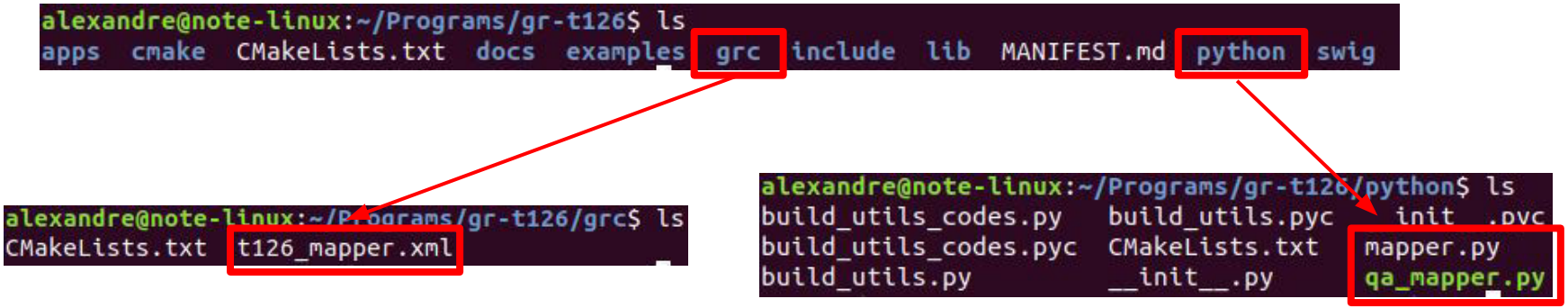
1) Fazer o download do arquivo de instalação .deb em https://code.visualstudio.com/Download;
2) Ir até a pasta do download e instalar o programa;

```
cd Downloads
sudo apt install ./<file_name>.deb
```

Edição de arquivos do bloco

Os arquivos que devem ser editados para o projeto do bloco são:

```
/gr-t126/python/mapper.py
/gr-t126/python/qa_mapper.py
/gr-t126/grc/t126_mapper.xml
```



Edição do arquivo `/gr-t126/python/mapper.py`

```
22  import numpy
23  from gnuradio import gr
24
25  class mapper(gr.sync_block):
26      """
27      docstring for blocl mapper
28      """
29      def __init__(self):
30          gr.sync_block.__init__(self,
31              name="mapper",
32              in_sig=[<+numpy.float32+>],
33              out_sig=[<+numpy.float32+>])
34
35
36      def work(self, input_items, output_items):
37          in0 = input_items[0]
38          out = output_items[0]
39          # <+signal processing here+>
40          out[:] = in0
41          return len(output_items[0])
42
43
```

```
22  import numpy as np
23  from gnuradio import gr
24
25  class mapper(gr.sync_block):
26      """
27      QPSK mapper.
28      """
29      def __init__(self):
30          gr.sync_block.__init__(self,
31              name="mapper",
32              in_sig=[np.uint32],
33              out_sig=[np.complex64])
34          self.constellation = {'qpsk' : np.array([(+1+1j), (+1-1j),
35                                                   (-1-1j), (-1+1j)])}
36
37
38      def work(self, input_items, output_items):
39          in0 = input_items[0]
40          out = output_items[0]
41          out[:] = self.constellation['qpsk'].take(in0)
42          return len(output_items[0])
43
```

Edição do arquivo `/gr-t126/grc/t126_mapper.xml`

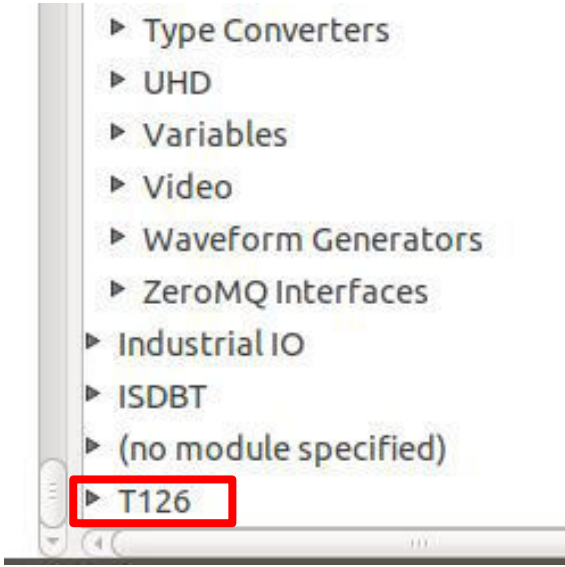
```xml
<?xml version="1.0"?>
<block>
  <name>mapper</name>
  <key>t126_mapper</key>
  <category>[t126]</category>
  <import>import t126</import>
  <make>t126.mapper()</make>
  <!-- Make one 'param' node for every Parameter you want settable from the GUI.
       Sub-nodes:
       * name
       * key (makes the value accessible as $keyname, e.g. in the make node)
       * type -->
  <param>
    <name>...</name>
    <key>...</key>
    <type>...</type>
  </param>

  <!-- Make one 'sink' node per input. Sub-nodes:
       * name (an identifier for the GUI)
       * type
       * vlen
       * optional (set to 1 for optional inputs) -->
  <sink>
    <name>in</name>
    <type><!-- e.g. int, float, complex, byte, short, xxx_vector, ...--></type>
  </sink>

  <!-- Make one 'source' node per output. Sub-nodes:
       * name (an identifier for the GUI)
       * type
       * vlen
       * optional (set to 1 for optional inputs) -->
  <source>
    <name>out</name>
    <type><!-- e.g. int, float, complex, byte, short, xxx_vector, ...--></type>
  </source>
</block>
```

```xml
<?xml version="1.0"?>
<block>
  <name>Mapper</name>
  <key>t126_mapper</key>
  <category>[T126]</category>
  <import>import t126</import>
  <make>t126.mapper()</make>
  <!-- Make one 'param' node for every Parameter you want settable from the GUI.
       Sub-nodes:
       * name
       * key (makes the value accessible as $keyname, e.g. in the make node)
       * type -->

  <!-- Make one 'sink' node per input. Sub-nodes:
       * name (an identifier for the GUI)
       * type
       * vlen
       * optional (set to 1 for optional inputs) -->
  <sink>
    <name>in</name>
    <type>int</type>
  </sink>

  <!-- Make one 'source' node per output. Sub-nodes:
       * name (an identifier for the GUI)
       * type
       * vlen
       * optional (set to 1 for optional inputs) -->
  <source>
    <name>out</name>
    <type>complex</type>
  </source>
</block>
```

Instalação do módulo `gr-t126`

Após abrir o terminal (Ctrl+Alt+t), digitar a sequência:

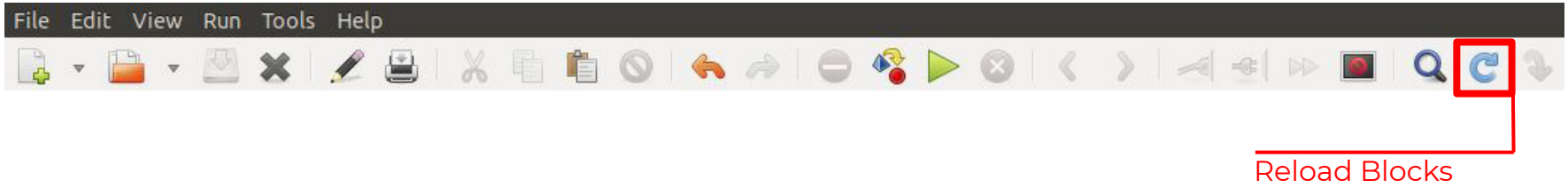
```
cd ~/Programs/gr-t126/
mkdir build
cd build
cmake ../
make
sudo make install
sudo ldconfig
```



O módulo aparecerá no *Block Tree Panel* após o GNU Radio for aberto.

Instalação do módulo `gr-t126`

Caso o GNU Radio já esteja aberto, será necessário pressionar o botão *Reload Blocks* na barra de ícones;



A cada vez que os arquivos dos blocos forem alterados, o módulo deverá ser instalado novamente para que as modificações tenham efeito.

Desinstalação do módulo `gr-t126`

Após abrir o terminal (Ctrl+Alt+t), digitar a sequência:

```
cd ~/Programs/gr-t126/build
sudo make uninstall
sudo ldconfig
```

Observação: caso os arquivos do módulo já tenham sido apagados (pasta `gr-t126`), há a opção de apagar manualmente o módulo (retirá-lo do *Block Panel Tree*).
Na pasta `/usr/local/share/gnuradio/grc/blocks` apagar o arquivo .xml correspondente ao bloco que se deseja retirar do *Block Panel Tree*.