Hints and Tips for GNU Radio: Out-of-Tree Module

Following hints and help was prepared based on Live DVD from GNU Radio website (Ubuntu ver. 14.04.1)

Prepared by Maciej Zawodniok

Block is added but does not execute

- **Errors** shown might look like this:
  - AttributeError: 'module' object has no attribute '**your_block_name**'
- **Reason**: XML file in "grc" is incorrectly generated (**grc/BLOCK_NAME.xml**)
- **Correction**: after "gr_modtool makexml" manually edit the XML file:
  - 1) **Replace**:
    - `<import>import MODULE_NAME</import>`
    - with:
    - `<import>from MODULE_NAME import MODULE_NAME_swig</import>`
  - 2) **Replace**:
    - `<make>MODULE_NAME.BLOCK_NAME()</make>`
    - With:
    - `<make>MODULE_NAME_swig.BLOCK_NAME()</make>`
  - 3) **Re-make the module** (i.e. "make; sudo make install; sudo ldconfig")
  - Note: Add "_swig" to module name since it is a C++/swig type module
  - Note: BLOCK_NAME() may have parameters (seen in GRC block properties) - keep them in the new <make> entry! (e.g. "BLOCK_NAME($param1)")
  - Note: It is also needed to execute "sudo ldconfig" after installation.
    - This way the "BLOCK_NAME_swig.so" file is visible to the GRC

Adding Parameters to Block

- These are the parameters in block properties window (“right-click”→Properties) withing the companion GUI
- Easiest way is to:
  - Start with new block creation and list ALL parameters during the block setup (see “gr_modtool” commands in intro)
  - If you have old code – copy the content of the two main classes:
    - Forecast
    - General Work
  - Make sure to define input/output types and regenerate XML config file

OR….

Adding Block Parameters: (1) with gr_modtool

- Block parameters (block properties in GRC) can be added:
  - During the block creation with “gr_modtool” command – add parameters for constructor
    - List all intended parameters in C/C++ syntax (i.e. “TYPE PARAM_NAME, TYPE2 PARAM_NAME2, ...”)
    - For example: “`int my_int_param, char param2`”
- Additionally, you need to store them inside the block class:
  - Create specific variables inside the class (members)
  - In constructor, copy the constructor parameters into the appropriate member variables
- Also, remember to remake the XML file in grc **and** fix the <import> and <make> lines before rebuilding and installing the module
- Now, you can use the parameters in GNU Radio block

Adding Block Parameters: (2) manual method – part 1/2

- Three files have to be modified (for block named **BLOCK_NAME**)
  - File “include/MODULE_NAME/BLOCK_NAME.h”
    - Add declaration of “make” function member inside the main class “BLOCK_NAME”:
      - “`static sptr make (int my_int_param, char param2)`”
    - List all intended parameters in C/C++ syntax (i.e. “TYPE PARAM_NAME, TYPE2 PARAM_NAME2, ...”)
      - For example: “`int my_int_param, char param2`”
  - File “lib/BLOCK_NAME_impl.h”
    - Add the intended parameters (see above) to the constructor declaration
      - “`BLOCK_NAME_impl(int my_int_param, char param2);`”
    - Add appropriate variable class members to store the variables
      - “`int my_int_param_in_class;        char param2_in_class;`”

Adding Block Parameters: (2) manual method – part 2/2

-
  - File “lib/BLOCK_NAME_impl.cc”
    - Define implementation of the “make” member
      ```
      – BLOCK_NAME::sptr BLOCKNAME::make(int my_int_param, char param2)
        {
           return gnuradio::get_initial_sptr(new BLOCK_NAME_impl(int
        my_int_param, char param2) );
        }
      ```
    - Initialize the variables in the constructor
      ```
      – BLOCK_NAME_impl::BLOCK_NAME_impl()
        … // the rest of the default constructor definitions
        {
           my_int_param_in_class = my_in_param;
           param2_in_class = param2;
        }
      ```
- Also, remember to remake the XML file in grc **and** fix the <import> and <make> lines before rebuilding and installing the module
- Now, you can use the parameters in GNU Radio block

(Last update: 10/12/2021)