

Developer documentation for gr-cogra

The GNU Radio CoGRA (Cognitive GNU Radio Applications or Cognitive Radio) package is a project to extend the GNU Radio Companion with out of tree Signal Blocks. It does this by using the standard GNU Radio way, which is by using CMake. In order to compile the code a complete build of all GNU Radio modules is necessary. This has to include the GNU Radio Companion, the GNU Radio Python libraries like the Modulation Toolbox from gr-digital and the C++ libraries. Additionally the Ettus UHD device drivers might be necessary in order to utilize a USRP.

The implemented Signal Blocks can be useful for IEEE 802.15.4 and IEEE 802.15.1 standard implementations. This project is focused on providing many ports for a Cognitive Engine. Parameters are exported as variables, and can be used for adaptive changes. The GRC XMLRPC server automatically generates callback functions for variables in a Flow-Graph so that on-the-fly changes of variables by a Cognitive Engine result into immediate effects.

Most of the code is re-written from the reference implementations of gr-bluetooth (Michael Ossman, Dominic Spill) and gr-ieee802_15_4 (Thomas Schmid, UCLA). Other used resources include contributions from the GNU Radio mailing list.

Eclipse CDT and CMake for multi-language project management

In the following there are suggestions on how to utilize Eclipse in order to extend the CogRa framework. CogRa uses Python, XML and C++ in order to extend GNU Radio in various ways.



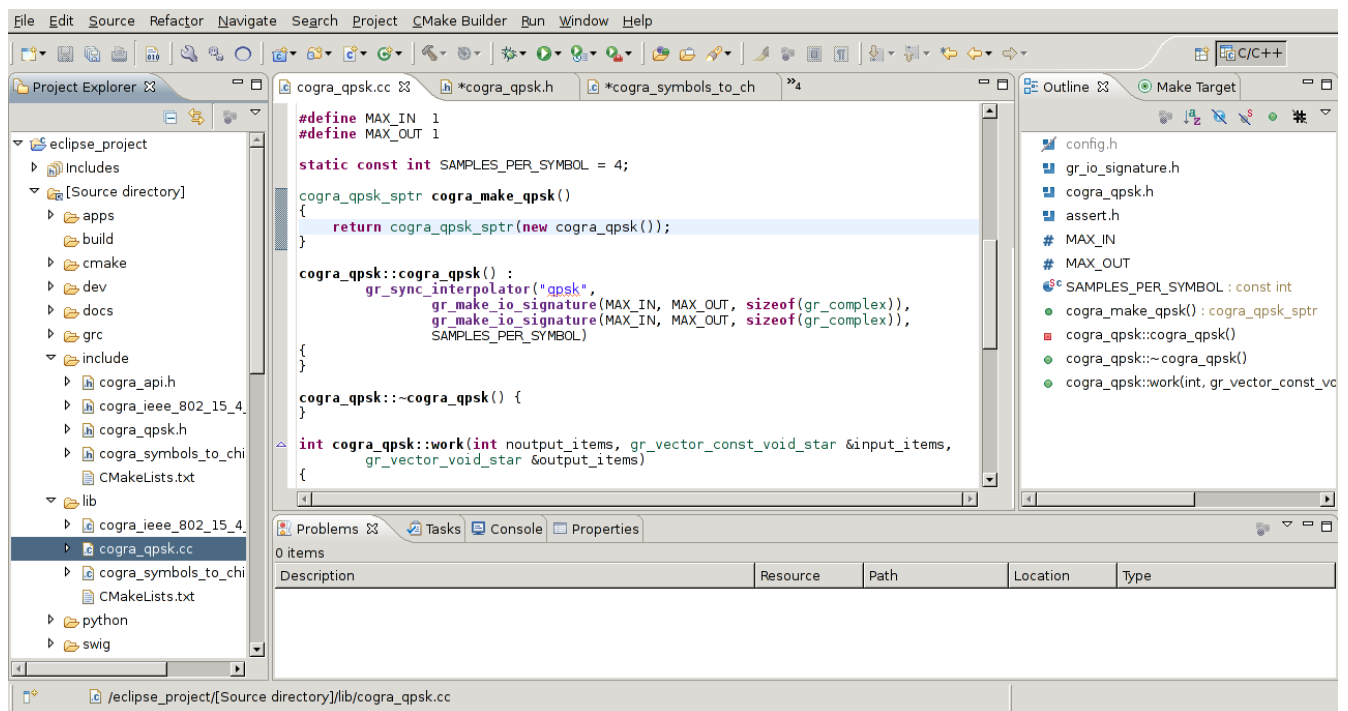
CMake-GUI generates an Eclipse project from the CogRa source

CMake (since 2.6) is able to generate an Eclipse CDT export. The Eclipse Makefiles the CMake-Gui generates shouldn't be located in a sub-folder of gr-cogra due to Eclipse CDT's indexing habits. It is best to create the Eclipse project in a separate location. It's furthermore recommended to use PyDev¹

¹ <http://pydev.sourceforge.net/> - PyDev is an extension for Eclipse for Python developers

and Eclipse CDT 4². CmakeBuilder³ can be very useful to locate project resources. These are freely available tools to extend Eclipse's C++, Python and XML editing features. The red warning can be ignored, since Eclipse uses its own indexer, that will have the paths for Boost, if it's installed in a standard path.

1. Use CMake-GUI as it is shown and select the creation of “Makefiles for Eclipse CDT 4”.
2. In Eclipse, generate a “New project from existing Makefiles”.
3. If the Eclipse CDT indexer shows that classes like `gr_basic_block()` cannot be located, it's not configured. You can edit the include-path right at the project settings. This will have the effect that Eclipse's code-checks can include constructors and GNU Radio library functions.
4. Now you have successful project setup. Changes will be directly applied to the original source folder - directly from Eclipse. Here Eclipse is just a frontend for C++, Python and XML. It's advised not to use the project build and install features. Latter may require root permissions in any case, and Eclipse is not a deployment tool.
5. Leave all integration of new C++ classes to `gr_modtool`⁴. That goes for deletion, too. That way you don't need to work with the SWIG and CMake configuration directly. Regenerate the Eclipse project each time. Then refresh (right-click on the project). You can leave Eclipse open.



Eclipse CDT 4 with the CogRa project

This is a way to use Eclipse CDT4 for GNU Radio development. A project started with the `gr_modtool`'s parser and code-generation should only be extended that way. Otherwise it's likely to fail.

For more information please refer to `./gnuradio/students/marius/thesis/Cognitive\ Radio\ Framework\ für\ GNU\ Radio\ Companion_Rev.pdf` – section 6 (German).

- 2 <http://www.eclipse.org/cdt/> - Eclipse CDT 4 is a version of Eclipse for C and C++ developers with many extensions
- 3 <http://www.cmakebuilder.com/> - CMakeBuilder extends Eclipse to allow editing of CMake based projects
- 4 <https://www.cgran.org/wiki/devtools> – `gr_modtool` is a useful tool to work with GNU Radio extensions

Installation of GNU Radio on a Ubuntu 12.04

You'll need a reasonably fast computer, Ubuntu 12.04 and administrative privileges. GNU Radio performs sophisticated computations under time-critical circumstances.

1. Download the GNU Radio build script from:
<http://gnuradio.org/redmine/projects/gnuradio/wiki/InstallingGR>
2. Execute: `./build-gnuradio prereqs` to get all the dependencies
3. If you need a USRP device, refer to the Ettus website. There is no special configuration necessary. <http://ettus-apps.sourcerepo.com/redmine/ettus/projects/uhd/wiki>
4. Download GNU Radio 3.6: <http://gnuradio.org/redmine/news/13>
5. Untar the file, create a build folder and run `cmake`. Ensure that CMake was able to track the dependencies. You will need the GNU Radio Companion, Python libraries for WX, Python libraries for GNU Radio and optionally `gr-uhd`.
6. Compile GNU Radio.
7. Navigate into `gr-cogra`, create a build folder and run `cmake`.
8. Compile and install.
9. Run `gnuradio-companion` and open a GRC Flow-Graph.