

GNURADIO RUNNING ON EMBEDDED BOARDS: PORTING TO BUILDROOT

G. Goavec-Merou

FEMTO-ST Time & Frequency, Besançon, France

GNURadio is a set of libraries implementing data acquisition and processing functions. GNURadio-Companion is a graphical user interface generating a Python script defining how processing blocks are connected with each other. Once the Python script is generated, no graphical user interface is needed any more to run the processing script, making GNURadio well suited to embedded environments such as Armadeus Systems APF6 boards. However, GNURadio is a complex software framework requiring multiple dependencies: we have ported GNURadio and associated libraries to Buildroot to make it compatible with most embedded platforms. We demonstrate simple flowchart execution on ARM-based embedded boards.

Keywords: embedded, buildroot, ARM

1 Introduction

GNURadio is a digital signal processing framework with multiple dependencies considering the wide range of applications, including prototyping on synthetic data, running on multiple platforms from x86-based personal computers to embedded boards and networked prototyping platforms (e.g. Cortex Lab ¹).

The ability of GNURadio-Companion, a graphical user interface defining the data flow between processing blocks, to generate Python code running on deeply embedded boards without graphical user interface increases the flexibility and the range of applications of the GNURadio framework. However, GNURadio requires a wide range of dependencies even when running without graphical user interface.

Historically, the developers of the core GNURadio framework have been focusing on integrating GNURadio in the OpenEmbedded framework, a powerful cross-compiling environment providing such functionalities as inheritance of platform functionalities. Despite this flexibility provided by such a functionality in which platforms are assembled from System on Chips (SoC) themselves assembled from individual components, OpenEmbedded is a complex framework requiring multiple tens of GB to store all the files needed to create a distribution, from which a user picks the packages needed in her own application.

On the opposite, Buildroot is a light cross-compilation framework providing a consistent set of bootloader, cross-compilation toolchain, Linux kernel, libraries and selected applications. Our use of buildroot on a wide range of platforms, including but not limited to Xilinx based Zynq platforms and

Armadeus Systems iMX based platforms, hinted at porting GNURadio to this framework.

2 Experimental setup

Buildroot supports a wide range of platforms. As an example of the space needed to integrate GNURadio in an embedded project generated using such a framework, running GNURadio including the graphical user interface support requires a total space of 11 GB for the Armadeux iMX6 based platform.

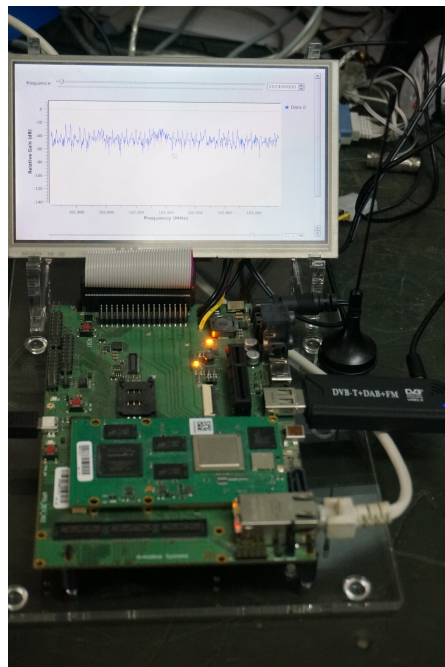


Figure 1: ARM based embedded board running GNURadio compiled using buildroot

¹<http://www.cortexlab.fr/>

3 State of GNURadio in buildroot

Currently buildroot provides packages (rules to compile applications or libraries) for `librtlsdr` (RTL2832U based DVB-T receiver used as SDR source), `osmosdr` and `gnuradio`. For the specific case of GNURadio, options available are: `gr-audio`, `gr-blocks`, `gr-ctrlport`, `gr-fec`, `gr-python`, `gr-utils`, `gr-analog`, `gr-channels`, `gr-digital`, `gr-fft`, `gr-filter`, `gr-pager`, `gr-qtgui`, `gr-trellis` [1].

```
[ ] empty
[*] gnuradio
[*] gr-audio
-*- blocks support
[*] ctrlport support
[*] gr-fec support
[*] python support
[*] gr-utils support
-*- gr-analog support
[*] gr-channels support
-*- gr-digital support
-*- gr-fft support
-*- gr-filter support
[*] gr-flex support
[*] gr-qtgui
[*] gr-trellis support
[ ] Google font directory
*** gqrx needs qt5, gnuradio, fftw
[*] gr-osmosdr
[*] python support
[*] IQ File Source support
[*] Osmocom RTLSDR support
[*] RTLSDR TCP Client support
[*] RFSPACE Receivers support
[ ] gsettings-desktop-schemas
```

Figure 2: curses interface of GNURadio configuration with full options active.

Firmware configuration is done through a curses based series of menus (Fig. 2). When a package is selected, mandatory libraries are automatically selected (like `numpy` for `gr-python`). For some others, as indicated by the user through the curses menu, the GNURadio option is hidden due to the need for the user to select manually missing dependencies: this is the case for `gr-fft` which needs `fftw` with single precision configuration, or as shown in Fig. 2 the case of `gqrx` whose mandatory dependencies are indicated.

The ability to finely enable or disable each `gr-xx` package allows to finely tailor the buildroot image to suit the user’s needs and adapt the storage footprint accordingly, depending on the level of

generality targeted (general purpose development board or finely taylorred dedicated board).

4 Results

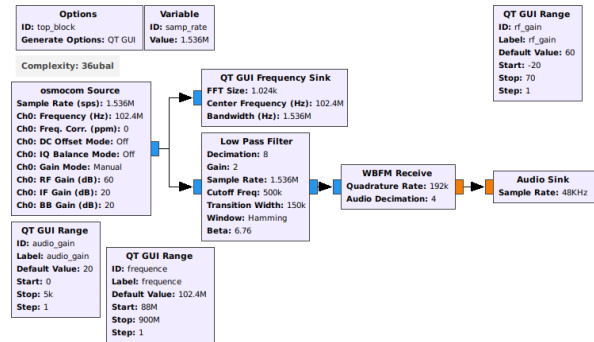


Figure 3: GNURadio Companion flow graph for acquiring and processing data: here the basic commercial broadcast FM demodulation scheme run on the APF6 board.

We demonstrate real time commercial broadcast FM demodulation by running the basic introductory flowchart shown in Fig. 3 on Armadeus Systems iMX6 board. In this case, while GNURadio does not need a graphical user interface to run, the spectrum display capability through the Qt interface is demonstrated as well, although we emphasize that this graphical framework is only displayed to show off the display capability and is not mandatory (No GUI option of GNURadio Companion).

5 Conclusion

GNURadio packages have been ported to Buildroot for easy accessibility on embedded platforms, including but not restricted to ARM based boards. While our current favored hardware interface is the RTL2832U based DVB-T receiver used as general purpose SDR receiver, many other interfaces would deserved being ported to the same framework. Extending the set of packages supported by Buildroot is hence still under progress, with additional contributions from the community sought to support this work.

References

[1] GNURadio package log: git.busybox.net/buildroot/log/package/gnuradio