

Tutorial 2: Before Diving into GNU Radio, You Should ...

Dawei Shen*

May 21, 2005

Abstract

GNU Radio requires both strong computer skills and extensive knowledge on communications and digital signal processing. This article lists some useful resources including textbooks, web links, on-line tutorials. The purpose of this sheet is to help the GNU Radio fans to get prepared for this exciting tool.

I definitely believe you have found GNU Radio interesting and are eager to play around with it. Unfortunately, it contains more challenge than fun. You need extensive knowledge on a wide range of areas, including communication (wireless) systems, digital signal processing, basic hardware and circuit design, OOP programming, etc. However, your interest and passion could make them much easier. In this page, I list some useful articles and resources, which might be important to you before you dive into the GNU Radio.

1 Having a clearer picture of GNU Radio ...

If you haven't, please read this on-line article first. *Eric Blossom, "Exploring GNU Radio"*. A very brief and excellent introduction to software radio. Eric is the founder of the whole GNU Radio project. Make sure you understand how ADC works and why we need an RF frontend. Recall the sampling theory learnt from your Signal and Systems class. Then take a close look at the two parts "**The Universal Software Radio Peripheral**" and "**What Goes in the FPGA**". This article also provides two examples, a simple dial tone output and an FM receiver. At least you should understand the first one. Can't understand the FM receiver? Never mind, read the second article: *Eric Blossom, "Listen to FM Radio in Software, Step by Step"*. You don't need to understand the code line by line, but you should know how the signal flows from the air to the soundcard. Then it's better for you to know more about what USRP does, the two pages "*USRP wiki*" and "*USRP User's Guide*" would be very valuable. Assuming you have gone through the articles above, you may go to the "*GNU Radio wiki*" pages to look for more information.

2 Programming on GNU Radio ...

To really "play" with GNU Radio, you should be able to write your own code. From the article "Exploring GNU Radio", you should have known that the software structure of GNU Radio contains two levels. All the signal processing blocks are written in C++ and Python is used to create a network or graph and glue these blocks together. So in this particular scenario, Python is a higher level language. Many useful and frequently used blocks have been provided by the GNU Radio project, so in many cases you don't need to touch C++, just using Python to finish your task. However, to do more sophisticated work, you have to use C++ to create your own block. In such cases, an on-line

*The author is affiliated with Networking Communications and Information Processing (NCIP) Laboratory, Department of Electrical Engineering, University of Notre Dame. Welcome to send me your comments or inquiries. Email: dshen@nd.edu

article *Eric Blossom*, "how to write a block" is what you need. You may want to know, what blocks have been provided to us? Unfortunately, unlike some other developing tools, such as TinyOS, GNU Radio is badly documented at this point. But you still have two very useful documents generated using Doxygen. After installing the "gnuradio-core" and "usrp" modules, you can find two html packages located at

- /usr/local/share/doc/gnuradio-core-x.xcvs/html/index.html
- /usr/local/share/doc/usrp-x.xcvs/html/index.html

I bookmark them in my browser. Although they are not clear enough, they can tell you quite much information. The first one is also available on-line here.

If you haven't had a chance to use Python, please go through the Python on-line tutorials. The most important sections are:

- Section 2: Using the Python Interpreter
- Section 3 An Informal Introduction to Python
- Section 6: Modules
- Section 7: Input and Output
- Section 9: Classes

They will be frequently used in GNU Radio programming. If Object-Oriented Programming (OOP) sounds unfamiliar to you, you should read section 9 more carefully. The following links also may help you to grasp the essence of OOP.

- Lesson: Object-Oriented Programming Concepts
- Introduction to Object-Oriented Programming Using C++
- The Object Oriented Programming Web

Anyway, Python seems to be crucial at this stage, so make sure you master it well.

3 Digital Signal Processing (DSP)

Most of us have taken the "Signals and Systems" class, I guess. What we learnt in this course is extremely important here. However, that's not enough. Make sure you won't get lost if I change the signal to analog or to digital world; to the time domain or to the frequency domain. The bottom line is, you need to know what is sampling theorem, what is z-transform, how to get the spectrum of a signal, and the concepts of FIR and IIR filters. I recommend several books here, which are famous and classical.

- "Signals and Systems (2nd edition)" - Alan V. Oppenheim, Alan S. Willsky
- "Discrete-Time Singal Processing (2nd edition)" - Alan V. Oppenheim, Ronald W. Schaffer, John R. Buck
- "Digital Signal Processing: Principles, Algorithms and Applications (3rd edition)" - John G. Proakis, Dimitris Manolakis

Read the chapters about discrete-time Fourier tranform and FIR, IIR filters. I know the books are expensive and tedious, there are some other useful on-line resources:

- Digital Signal Processing Tutorial
- The Scientist and Engineer's Guide to Digital Signal Processing

4 Communications

We know the real signals we send and receive can't be in base band. They need to be modulated and demodulated. I believe you have studied the concepts of AM and FM radios in some courses. Both of them belong to the analog world. To develop more useful and interesting schemes, we need digital communications. Of particular importance and interest at this point, is digital modulation and demodulation, synchronization. The course in your senior year, "Communication Systems" may be of interest to you. Further more, I recommend you read:

- Chapters 4 and 5, "**Digital Communications (4th edition)**" - John G. Proakis

The knowledge introduced in these two chapters is exactly what we are going to try out recently.

5 Ready to start?

I list four topics above. It doesn't mean you should complete them one by one before you can do something with GNU Radio. You certainly can learn them through this study process. But at least you should read the articles I mentioned above in section 1 and section 2. Then you can try this "homework":

In the module "*gnuradio-examples*", you can find many sample codes in the folder *gnuradio-example/python/usrp/*. Can you read the following two programs and understand the code line by line?

- *gnuradio-examples/python/usrp/am_rcv.py*
- *gnuradio-examples/python/usrp/wfm_rcv_gui.py*

If so, I would say you have moved a huge step.

References

All the book chapters, on-line resources mentioned in this article.