Technology
Arts Sciences
TH Köln

Faculty 07
Digital Communications Lab

SMP: Signal Processing with Micro Controller and Python

Lab 1: Introduction to GNU Radio

Version 1.0

October 18, 2023

1 Introduction

1.1 General rules

Questions must be answered in written form at home in advance.

Measurements will be worked out and documented during the lab.

Extra measurements are not mandatory but can be elaborated to gain a deeper understanding.

Please note:

- Figures are to be labeled before printing. They should have a title and marked axes.
- Play around and modify the schematics to try out your own ideas.

1.2 Bibliography and links

- GNU Radio Tutorials https://wiki.gnuradio.org/index.php/Tutorials, access October 10, 2018
- Ohm, Jens, Lüke, Hans Dieter, 'Signalübertragung', Springer, 2014
- Elders-Boll, 'Vorlesungsskript ASS/DSS TH Köln', 2018
- Schlichthärle, Dietrich, 'Digital Filters', Springer, 2011

2 Overview

In this semester we use GNU Radio Companion to generate and test signal processing algorithms in the audio range in real time.

GNU Radio is a open-source software development toolkit that provides signal processing blocks to implement software radios. It can be used either stand alone for simulation purposes using, e.g., internal or external audio sources connected to the computer or it can be

combined with external RF hardware to create software-defined radios. It is widely used in research, industry, academia, government, and hobbyist environments.

2.1 Learning goals

- using a free open source software for real time signal processing
- understand some simple programs for audio processing
- learn how to simulate communication and signalprocessing algorithms

3 Block diagram of a signal processing system

Figure 1 shows a basic DSP System suitable for processing audio frequency signals consisting of an A/D converter, a digital signal processor, and D/A converter to provide the processed input signal.

Figure 1: block diagramm of a DSP system

With GNU radio such systems can be implemented either on a standard computer, a RaspberryPI or other computing hardware either standalone or in conjunction with external RF hardware components that include the ADC and DAC conversion functions shown in figure 1.

To get familiar with the principal usage of GNU Radio in this lab you will conduct some beginners tutorials as provided on the GNU radio wiki page. Additionally, you will design two simple simulation setups. In the following lab you then will use GNU radio to design and test FIR filters .

4 Tutorials

4.1 Your first flowgraph

This tutorial explains a basic flowgraph design.

Lab Exercise: Your First Flowgraph

1. Go to the web page https://wiki.gnuradio.org/index.php?title=Your_First_Flowgraph and follow the instructions.
2. Use the mouse wheel to open the control panel and to control the properties of the figures.
3. Change the sampling rate from 32k to 8k and see what happens
4. Now, what happens if you set the signal frequency to 2 kHz and afterwards to 5 kHz? Can you explain this behaviour?

Your Solution:

When changing the sampling rate to the given frequencies, there is no change in the time-domain, nor frequency-domain plot.
Since the Niquist-Frequency (2kHz) is fullfilled on all given rates for our signal-frequency of 1kHz.

If the sampling rate is changed below the Nuquist-Frequency (e.g. 500Hz), the sinusoid is recreated wrongly from the discrete sampling points, because of insufficient sample-data. Aliasing occurs.

4.2 Python variables in GRC

Lab Exercise: Python variables in GRC

1. Go to the web page https://wiki.gnuradio.org/index.php?title=Python_Variables_in_GRC and read through the text.

4.3 Variables in flowgraphs

Lab Exercise: Variables in flowgraphs

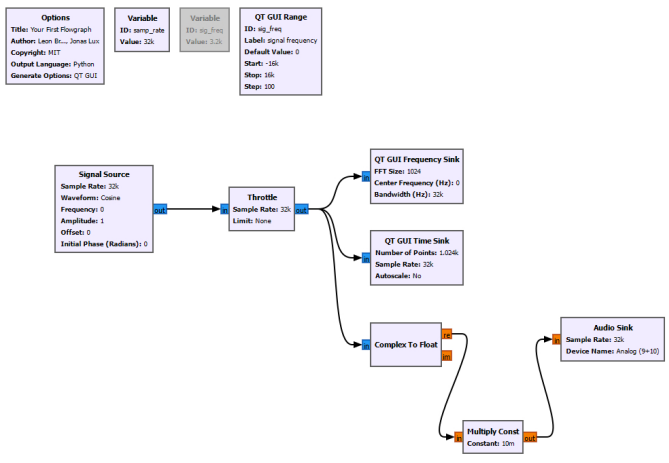
1. Go to the web page https://wiki.gnuradio.org/index.php?title=Variables_in_Flowgraphs and read through the text
2. Modify your "first flowgraph" according to the manual.
3. Play along with the control panel parameters to optimize the presentation of the figures.

4.4 Runtime Updating Variables

Lab Exercise: Runtime Updating Variables

1. Go to the web page https://wiki.gnuradio.org/index.php?title=Runtime_Updating_Variables and follow the instructions
2. Modify your the last setup according to the manual.
3. Add an audio sink to listen to the sine tones. For this you need the blocks "Complex To FLoat" and "Audio Sink". Be careful with the volume!

Your Solution:



4.5 Further basic tutorials

The tutorials "Signal Data Types", "Converting Data Types", "Packing Bits", "Streams and Vectors" and "Hierarchical Blocks and Parameters" provide further information to get a general overview. It is recommended that you conduct these tutorials offline at home.

5 Creating and Modifying Python Blocks

This section is of special importance since in your project you can use Python blocks to solve many different specific problems by Python numpy programming.

5.1 Creating Your First Block

Lab Exercise: Creating Your First Block

1. Go to the web page https://wiki.gnuradio.org/index.php?title=Creating_Your_First_Block and follow the instructions
2. Which signals are generated by the signal source blocks if the complex number type is chosen? Can you explain the shape of the two output signals, if the inputs are added or multiplied, respectively?
3. Build up the schematic for real valued input and output signals and compare the results with the complex valued schematic.
4. Write down the appropriate addition theorem to describe the multiplication of two cosine functions of different frequency.

Your Solution:

Generated signals by signal blocks:
Signal1: 1kHz cosine, amplitude: 1, complex
Signal2: 3kHz cosine, amplitude: 1, complex

Addition:
Signals get added in the time-domain, whereas they stay seperated in the frequency domain (spikes at 1kHz, 3kHz))

Multiplication:
When multiplied, the signals form 1 sinusoid in the time-domain.
In the frequency domain they get summed to one spike at 4kHz (frequencies added together).

6 Get creative

At the end of this lab you may build up a simple pass through and echo schematic and listen to the effects on the microphone signal and an audio file.

Lab Exercise: Creating two simple schemes

1. create a schematic that takes the audio input from a microphone or a wave file, respectively, and passes it to a loudspeaker.
   For this you may use the blocks “Audio Source”, “Wave File Source” “Selector”, “Throttle”, “Audio sink”, and “QT GUI Chooser”.
2. Now generate a simple echo signal by using additionally the “Delay” block.

Your Solution:

7 Conclusions

At the end of this lab you should have become familiar with the the basic use of GNU Radio blocks and schematics. A focus should be on the Python block which can be used to solve many diffent problems, e.g., as for your final project.

7.1 Further basic tutorials

You should work through the tutorials “Python Block with Vectors”, “Python Block Message Passing”, and “Python Block Tags” at home to complete your knowledge on how to program your own Python block before the final projects will start in December.