Digital TV Techniques 2012

GNU Radio Exercise

By Stefan Grönroos (stgronro@abo.fi)

In this exercise you will play around a bit with GNU Radio, and the graphical tool GNU Radio Companion. GNU Radio is an open source framework for building software defined radio (SDR) systems, i.e. systems where almost all the signal processing is performed in software instead of hardware.

For GNU Radio there are for example implementations of GSM base stations (OpenBTS), digital TV receivers, weather satellite image receivers, etc. All of these can be run on ordinary PCs.

Today you will look at an FM radio receiver example for GNU Radio and build a simple FM transmitter using ready-made signal processing blocks.

Note: This exercise was written in a bit of a haste, so hopefully I've not forgotten some crucial bit of info here. In that case, just use your heads, and/or google it! This is the best way to learn after all.

There won't be any supervision, and you may all work as a team (there's only one set of equipment after all!).

Equipment

In the lab, there are two laptops set up. These laptops are connected to two Ettus Research USRP2 devices, which are used to capture and transmit signals on the air. The signals are transferred as complex (IQ) samples across a gigabit ethernet connection to/from the laptops.

The laptops are set up with an Ubuntu linux 11.10 system, with a user account as follows:

username: sdr

password: sdr

By running the *gnuradio-companion* program from a terminal or *Applications->Programming->GRC* you will access the graphical GRC interface for designing software radios. Under the hood, GNU Radio signal processing blocks are specified using a combination of python for high level stuff, and C++ for the performance critical parts. Actually, GRC will generate python code when you run the models you build there.

FM receiver (use the laptop labeled “receiver” for this part)

If you open /home/sdr/gnuradio_exercise/uhd_wbfm_receive.grc within GRC, you will see an example of an FM radio receiver implementation.

The "UHD: USRP Source" block is used to retrieve samples received on the USRP device connected to the computer. Note that the IP address of the USRP is 192.168.10.2 by default. The USRP2 samples internally at 100 Msps (million/Mega samples per second). In the USRP block you can specify a lower rate, however, and the USRP2 will try to match that by downconversion done inside the USRP. In this case the rate is set to 480 ksps. The USRP can not exactly deliver 480 ksps, since it's not evenly divisible by 100 Msps, but will match quite closely.

I chose the 480 ksps sampling rate since it's nicely divisible with the supported 48000 Hz sampling rate of the sound card in the computer, and is enough to reproduce the full ~100 kHz wide FM radio station. The center frequency is the frequency that the USRP should tune into. So with a 480 ksps sample rate, we will be able receive a 480 kHz band around the center frequency.

The low pass filter following the USRP block simply filters out some of the higher frequencies in the 480 kHz band, since we only need about 100 kHz to reproduce an FM station. The WBFM block does the actual heavy work of decoding the FM signal and converting it into an audio signal that can be processed by the sound card, represented by the Audio Sink block.

The WBFM block is actually comprised of several simpler signal processing blocks, and is quite well described here: http://radioware.nd.edu/documentation/basic-gnuradio/exploring-the-fm-receiver (http://bit.ly/HQs1yM)

Turn on the USRP connected to the laptop. If it is not done automatically, select the connection with usrp in its name in the network settings on the taskbar of Ubuntu Linux. You should now be able to **ping 192.168.10.2** successfully.

Run the FM receiver by pressing the execute button in GRC. You can now try to tune into a normal FM radio station by using the frequency slider in the GUI that pops up. You also see a frequency graph (FFT) of the tuned frequency range.

You will probably not get excellent audio quality from any station. This might be due to the suboptimal antenna, or some bad sampling rate decisions made by me :) Try to move the antenna around a bit, and perhaps raise the gain slider to around 10 dB if needed.

FM radio transmitter

Now, on the laptop labeled transmitter, you should try implementing an FM radio transmitter. Basically this should consist of the receiver, but backwards. You probably don't need the low pass filter though.

In the end of the chain you will have a USRP Sink block. Found under the category "UHD" in GRC. You get to edit the properties of the placed block by double clicking it. You at least need to specify the "Device address" to be "addr=192.168.10.2", and the sample rate. For sample rate, you can use the same as the receiver, for convenience. As input to the system, you can for example use the “Signal Source” block (found under “Sources”). The WBFM transmitter (under “Modulators”) block expects float values as input, so you can choose that as the output type of the signal source block. The sample rate is the same as for example the audio sampling rate of the system (48k), and frequency is the frequency of the tone generated by the block. You can also use the Audio Source to get input from the microphone.

Specify a center frequency of, say, 830 MHz for your FM radio transmission. A frequency this high is needed, as the USRP2 connected to the “transmitter” laptop can only handle frequencies around 800-1000 Mhz due to the FLEX 900 daughterboard inside it. The USRP2 connected to the “receiver” laptop is equipped with a WBX daughterboard, which can handle anything in the interval 50 MHz to 2200 MHz.

Modify the receiver accordingly, to use the new higher center frequency, and try to listen for your transmission!

Note that we don't use an antenna on the “transmitter” in order to not disturb other radio traffic (such as cell phones) too much with our experiments. **Here also, you might need to move the receiver antenna a bit, adjust the gain of the receiver to 10 or so (better not touch the transmitter gain too much), and raise the volume slider (and speaker volume) quite a bit.**

More info about many things GNU Radio can be found at:
http://gnuradio.org
...and by Googling for stuff.

Report!

Write a short report outlining your findings and results, and submit one report per person (it's allowed to collaborate on the report though!). If you can submit the report to the usual system for exercises, do that. Otherwise, contact me (stefan.gronroos@abo.fi).

Good luck!