Russell Silva
AMRUPT 2018

Weekly Report – 5/8/2018

**Goals:** Retrieving real-time angle of arrivals with the RTL-SDR coherent receiver.

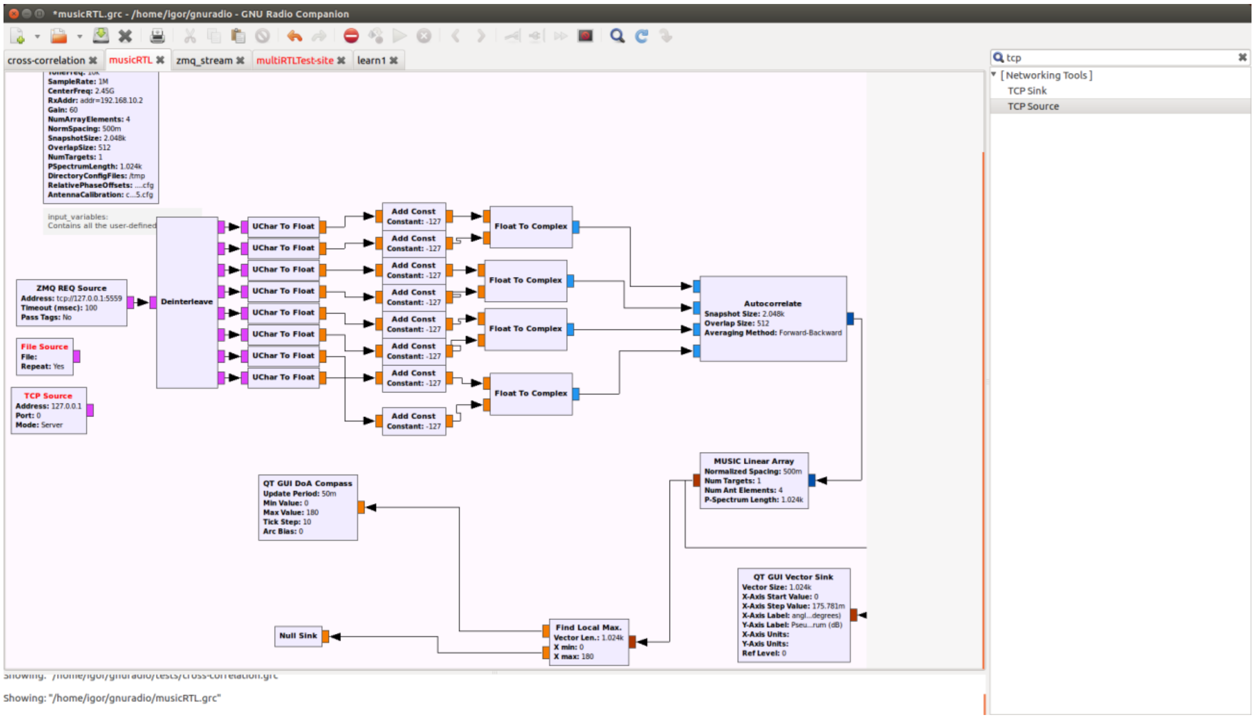
**Problems:** Even though the OutOfTreeModules were working on my laptop's Ubuntu subsystem, this subsystem could not register RTL SDR devices or any USB device. Therefore, I proceeded to install Ubuntu on a Raspberry Pi 3 with Ubuntu Mate installed. This did not work as this version experienced configuration issues. Peidong and I then proceeded to installing Ubuntu Core on a Raspberry Pi 3 server, which had issues with displaying GUI applications (e.g. GNU Radio). Finally, Justin and I installed Ubuntu 16 on Virtual Box, which successfully ran Ettus GRC simulations. Even though we finally got the software working without runtime errors, this took up a huge amount of time this past weekend and Monday, limiting our work with GNU Radio debugging and I2C protocols.

**Solutions:** I have been working with GNU Radio Companion the last couple of hours to try to get some angle of arrival measurements without RF switching. The expected result of this execution would be an angle of arrival that is inaccurate, but precise - the angle of arrival does not change if the transmitter is stagnant. The reasoning for this is that noise switching would correct an unchanging offset between the clock inputs which only affects the start time of sample collections at each RTL SDR.

In GNU Radio Companion, I designed a rudimentary flowchart based on Figure 1. I have been working with two rtl-sdr receivers at once instead of four. Therefore, I have adjusted the autocorrelation overlap size to 1024k instead of 512k (2048k total vector size/2 rtl-sdr receivers = 1024k overlap size), the MUSIC Linear Array's Num_Ant_Elements variable to 2 instead of 4, and other parts of the flowchart accordingly. The other difference between the current rudimentary flowchart and the one constructed in Figure 1, is that RTL-SDR sources transmit data in real-time to other blocks as shown in Figure 2, instead of file transfer from a remote receiver to computer via TCP/ZMQ.

The latter difference is likely the source of the current issue: an angle of arrival appears for a very short time, and then disappears to 0/-1. The GUI compass updates around every 50 milliseconds, therefore a sampling offset could result in autocorrelation malfunctions after a first iteration of sampling collection. That is, the autocorrelation would receive samples numbered at perhaps [0 – 2000], and then [2000 – 4000] which would be inconsistent with the 2048k snapshot size of the autocorrelation block. I have looked into a "head" block which takes finite samples. This only works for 1 iteration, and I am currently looking into timing/iteration blocks that could allow for continuous angle of arrival collection.

## Code/Debugging:



SampleRate: 1M
CenterFreq: 2.45G
RxAddr: addr=192.168.10.2
Gain: 60
NumArrayElements: 4
NormSpacing: 500m
SnapshotSize: 2.048k
OverlapSize: 512
NumTargets: 1
PSpectrumLength: 1.024k
DirectoryConfigFiles: /tmp
RelativePhaseOffsets: ....cfg
AntennaCalibration: c...5.cfg

ZMQ REQ Source
Address: tcp://127.0.0.1:5559
Timeout (msec): 100
Pass Tags: No

File Source
File:
Repeat: Yes

TCP Source
Address: 127.0.0.1
Port: 0
Mode: Server

Add Const
Constant: -127

Autocorrelate
Snapshot Size: 2.048k
Overlap Size: 512
Averaging Method: Forward-Backward

MUSIC Linear Array
Normalized Spacing: 500m
Num Targets: 1
Num Ant Elements: 4
P-Spectrum Length: 1.024k

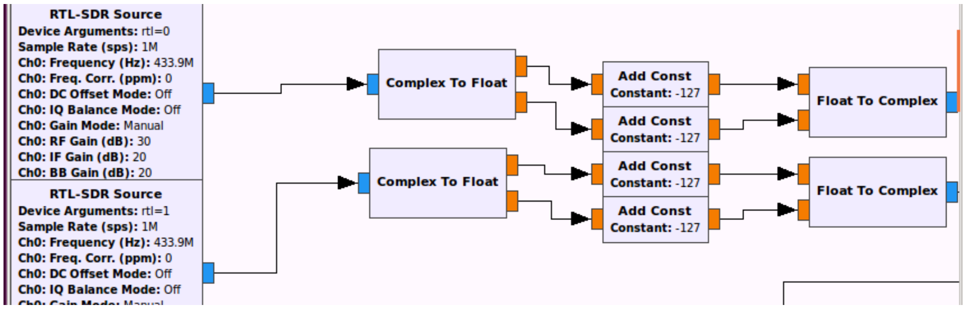
QT GUI DoA Compass
Update Period: 50m
Min Value: 0
Max Value: 180
Tick Step: 10
Arc Bias: 0

QT GUI Vector Sink
Vector Size: 1.024k
X-Axis Start Value: 0
X-Axis Step Value: 175.781m
X-Axis Label: angl...degrees)
Y-Axis Label: Pseu...rum (dB)
X-Axis Units:
Y-Axis Units:
Ref Level: 0

Find Local Max.
Vector Len.: 1.024k
X min: 0
X max: 180

```
Showing: "/home/igor/gnuradio/tests/cross-correlation.grc"
Showing: "/home/igor/gnuradio/musicRTL.grc"
```

Figure 1: Source - https://coherent-receiver.com/getting-started



RTL-SDR Source
Device Arguments: rtl=0
Sample Rate (sps): 1M
Ch0: Frequency (Hz): 433.9M
Ch0: Freq. Corr. (ppm): 0
Ch0: DC Offset Mode: Off
Ch0: IQ Balance Mode: Off
Ch0: Gain Mode: Manual
Ch0: RF Gain (dB): 30
Ch0: IF Gain (dB): 20
Ch0: BB Gain (dB): 20

RTL-SDR Source
Device Arguments: rtl=1
Sample Rate (sps): 1M
Ch0: Frequency (Hz): 433.9M
Ch0: Freq. Corr. (ppm): 0
Ch0: DC Offset Mode: Off
Ch0: IQ Balance Mode: Off
Ch0: Gain Mode: Manual

Add Const
Constant: -127

Figure 2:  Real-time data collection instead of File Transfer.

**Note:** I have fallen behind on my other classes because of time spent on this project in the last week. I have several finals and projects that have not received attention. I will try my best to accommodate a few additional hours of work this week.