

In the lab we created an FM radio using GNU Radio and the HackRF. For the homework assignment you will also create an **FM Radio** but with some minor tweaks.

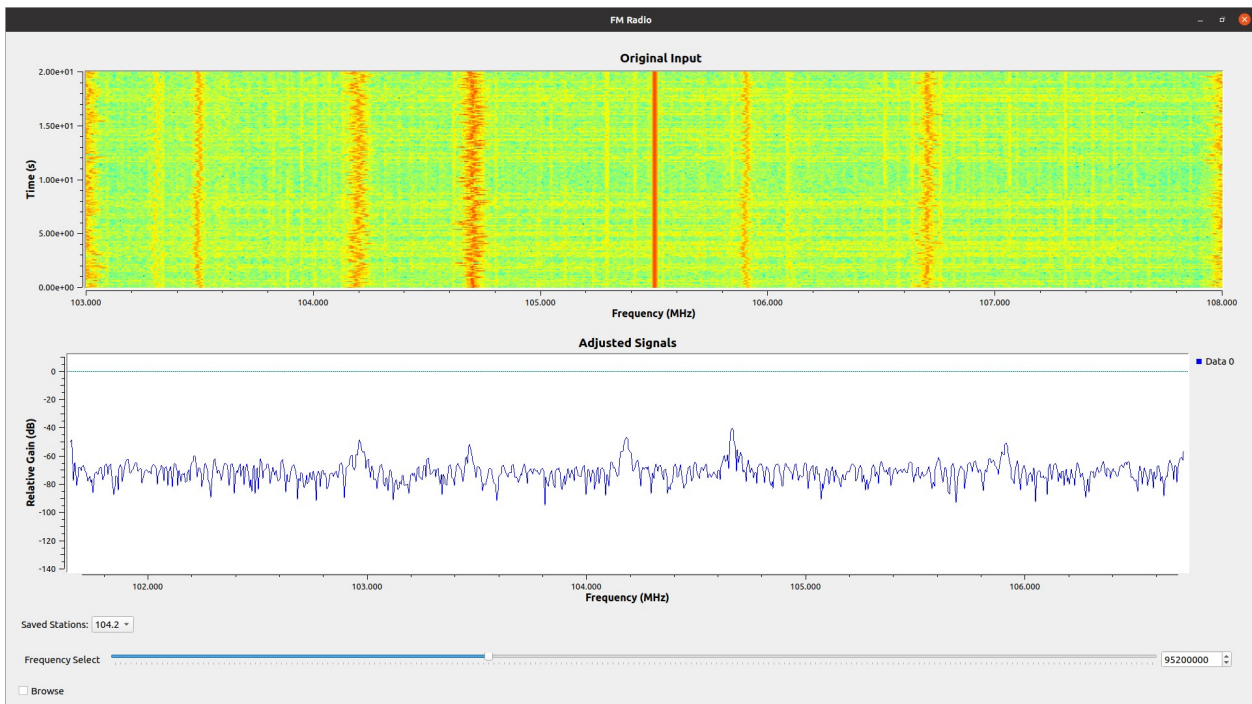
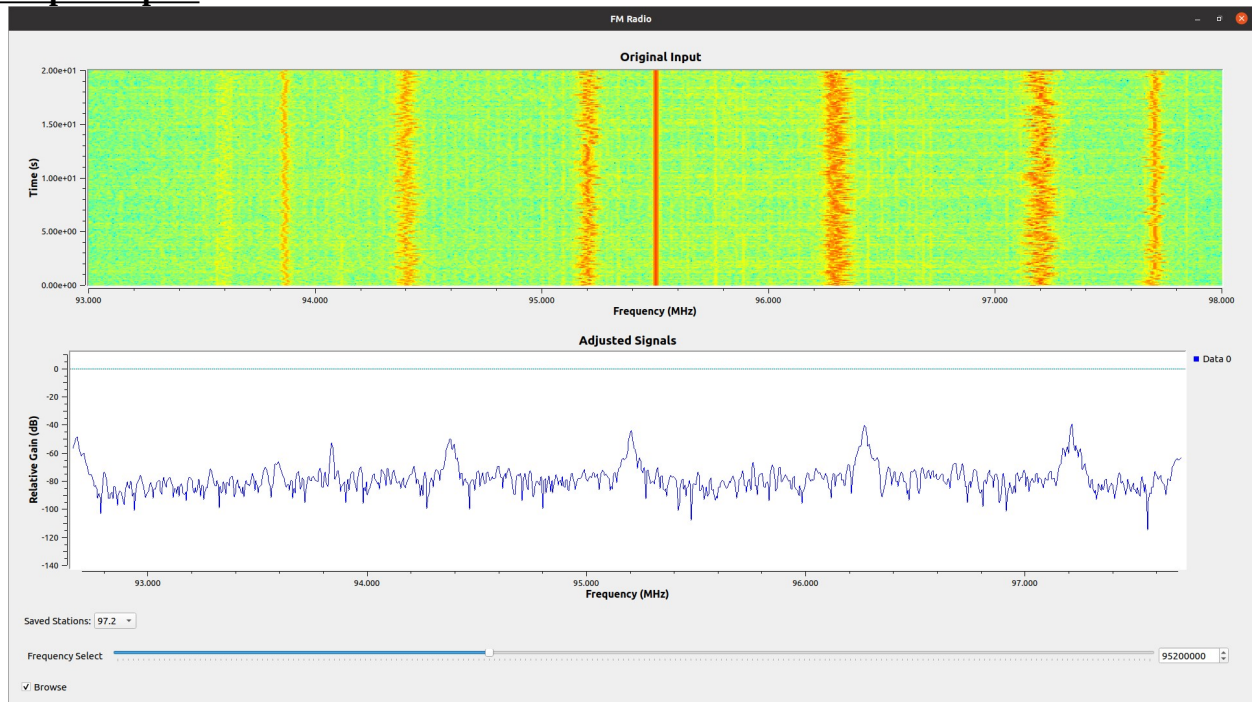
### **FM Radio construction:**

- The graph should have a **GUI Range block** which allows a user to scroll the entire FM radio spectrum (88 to 108 MHz). You can choose your own label for this block and any default frequency value in the range 88 to 108 MHz. Any Widget type can be used. **[0.25 pts]**
- A **GUI Waterfall sink block** should be attached to the output of the HackRF. The current frequency being received should always be displayed on the graph's x-axis i.e. as the HackRF's center frequency changes the center frequency of the GUI waterfall should change as well. Label this graph "Original Input". **[0.25 pts]**
- There should also be 5 predefined radio stations (in any order). Use the **GUI Chooser block** to set these stations. Only the station call number should be seen on the executed program i.e. 97.2 instead of 97.2e6 or 97200000. You can choose your own label for this block and any Widget type, however, there should be at least one station in each of the following frequency ranges (*read further for why*): **[0.25 pts]**
  - **88 MHz to 92.9 MHz**, some stations in this range (*in Tartu*): 89, 89.7, 90.3, 92.1 (MHz)
  - **93 MHz to 97.9 MHz**, some stations in this range: 94.4, 95.2, 97.2, 97.7 (MHz)
  - **98 MHz to 102.9 MHz**, some stations in this range: 98.6, 99.4, 100.2, 101.2 (MHz)
  - **103 MHz to 108 MHz**, some stations in this range: 103, 104.2, 104.7, 106.7 (MHz)
- To allow the user to switch between the browsing option and predefined stations, use a **GUI Check Box**. **[0.5 pts]**
  - If the check box is selected (there is a tick) then the user should only be able to use the scroll bar to browse the spectrum. The GUI Chooser should not affect the incoming signal.
  - If the check box is not selected (there is no tick) then the user should only be able to set the radio station using the GUI Chooser options. The GUI Range should not affect the incoming signals.
- The sample rate (and hence bandwidth) of the HackRF should be a constant 5MHz. Additionally a user should be able to browse/scroll to any radio stations in that bandwidth/range without the HackRF changing its center frequency. This means that to view the entire radio spectrum:
  - The **HackRF block** would need to have 4 different center frequencies. These are: **[0.25 pts]**
    - if the selected frequency (predefined or browse) is in the range 88 MHz to 92.9 MHz the center frequency should be **90.5MHz**
    - else if the selected frequency is in the range 93 MHz to 97.9 MHz the center frequency should be **95.5 MHz**
    - else if the selected frequency is in the range 98 MHz to 103 MHz, the center frequency should be **100.5 MHz**
    - finally if the selected frequency falls between 103 MHz and 108 MHz, the center frequency should be **105.5 MHz**
  - Use the **Python module block** to add the above conditions to your graph **[1 pt]**
- Use a **Signal Source block** to multiply the input of the HackRF with a frequency that would allow the user to move to any station in the 5 MHz range (*see lab manual*). **[0.25 pts]**
- Add a **GUI Frequency Sink**, labelled "Adjusted Signals", to the output of the Multiply block to show the current frequency (and possibly FM station) the user is tuned into. That is as the user moves the scroll bar or the radio buttons the center frequency of the frequency sink should change as well. **[0.25 pts]**

## Useful Resources:

- <https://greatscottgadgets.com/sdr/1/>
- HackRF One Tutorial\_F2017 – Report.docx.pdf. Located on Moodle

## Sample output:



## Note

- As was done in Homework 1, the HackRF will only receive a frequency value from either the GUI Range or the GUI Chooser.
- Depending on the chosen frequency, the HackRF will only select one of 4 values: 90.5 MHz, 95.5 MHz, 100.5 MHz or 105.5 MHz. For example, if the value provided by the GUI

Chooser or GUI Range is 95.2 MHz then the HackRF frequency should be 95.5 MHz. This gives us a view of the spectrum from 93 MHz to 97.9 MHz.

- To hear the audio produced on 95.2 MHz, the output from the HackRF (current center frequency 95.5 MHz) should be multiplied with the output from the Signal Source block. As was shown in the lab manual, multiplying these 2 signals can shift the center frequency that is sent to the next blocks.
- The main challenge is determining what frequency value the Signal Source should have to change 95.5 MHz to 95.2 MHz.

### **Bonus**

If the homework is too easy and you want extra points or more of a challenge (only attempt if the above works well):

- Implement volume control for the radio **[0.5 pts]**
- Implement stereo audio (i.e. Left and Right) for the radio with volume control for each channel (right ear and left ear should be able to have different volume levels) **[1.5 pts]**

### **Hints**

- GUI Hint can be used, if desired, to rearrange objects in the output. However, please remember that if one object is configured you will need to configure all or the arrangement will be disordered (in some cases it is impossible to interact with the GUI objects when the script runs).
- The assignment requires an FM radio to be built, please make a radio. Don't simply follow the construction list above.
  - The lab manual explains how to build a working radio
  - The lab video shows how to create a working radio with the RTL-SDR
    - Please remember that the HackRF block (osmocom Source) needs the DC Blocker, the RTL-SDR does not
- As the bandwidth of the radio is 5 MHz (instead of 2 MHz), the Low Pass Filter's and/or the Rational Resampler's values need to be adjusted
- Radio stations can only appear every 0.1 MHz e.g. 92.3 MHz, 92.4 MHz, 92.5 MHz etc... the scroll bar should be set with this in mind
- If the python module block gives an error even if it is added correctly, please check that your code has no syntax errors
  - Please remember to upload this file as well in your submission
- Also, please remember to configure the options block with your name.

**Due Date:** Check Moodle for date

**Late Submission Deadline:** Check Moodle for date (usually 1 week after due date)

**Submission Files:** .grc file and python file from the Python module block (and execution .py file if modified)

**Location:** Moodle Homework 2 link