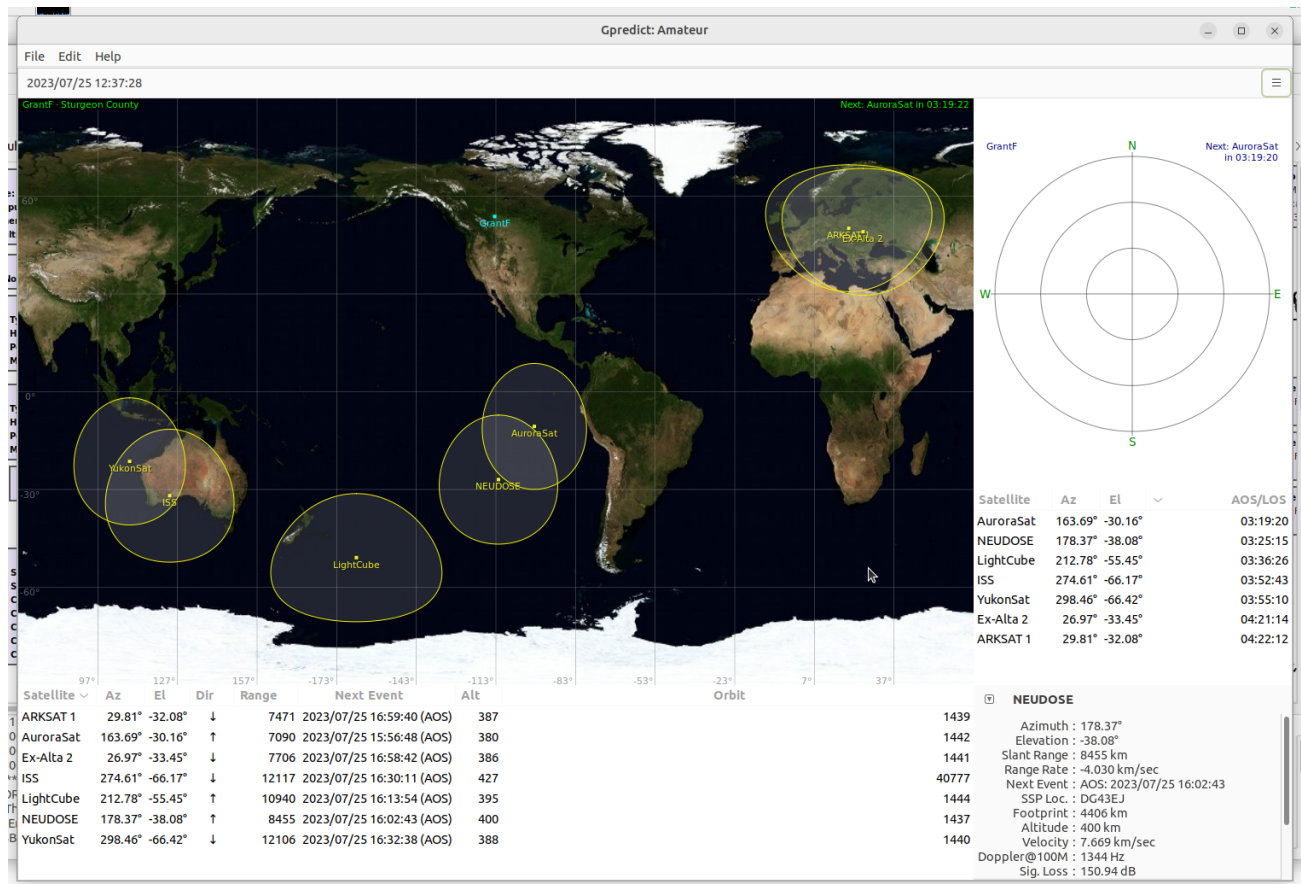


Run GPredict to track satellites and provide Doppler correction

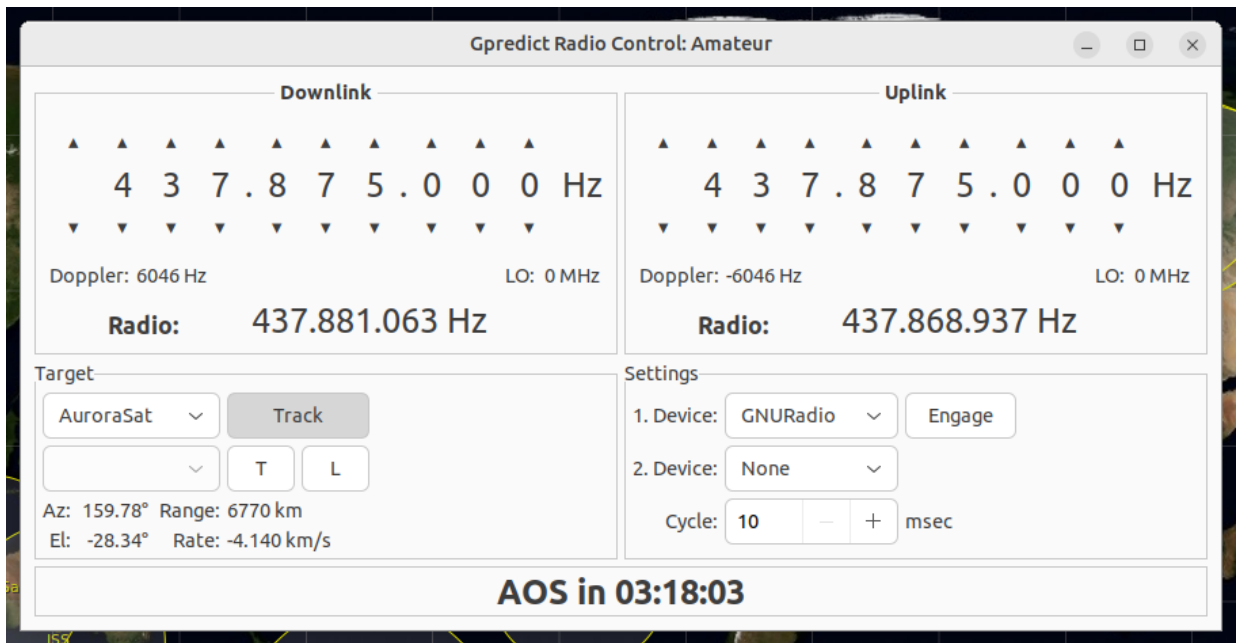
1. Open a terminal (shortcut is ctrl-alt-t)
2. Enter the command

```
gpredict
```

3. GPredict will start and all the satellites of interest should be displayed.



1. Under the "hamburger" menu at the top right, select "Radio Control". The GPredict Radio Control window will open. Select the satellite you want to track and set the up/down frequencies to match.
2. Click the "Track" button, but not "Engage", yet.

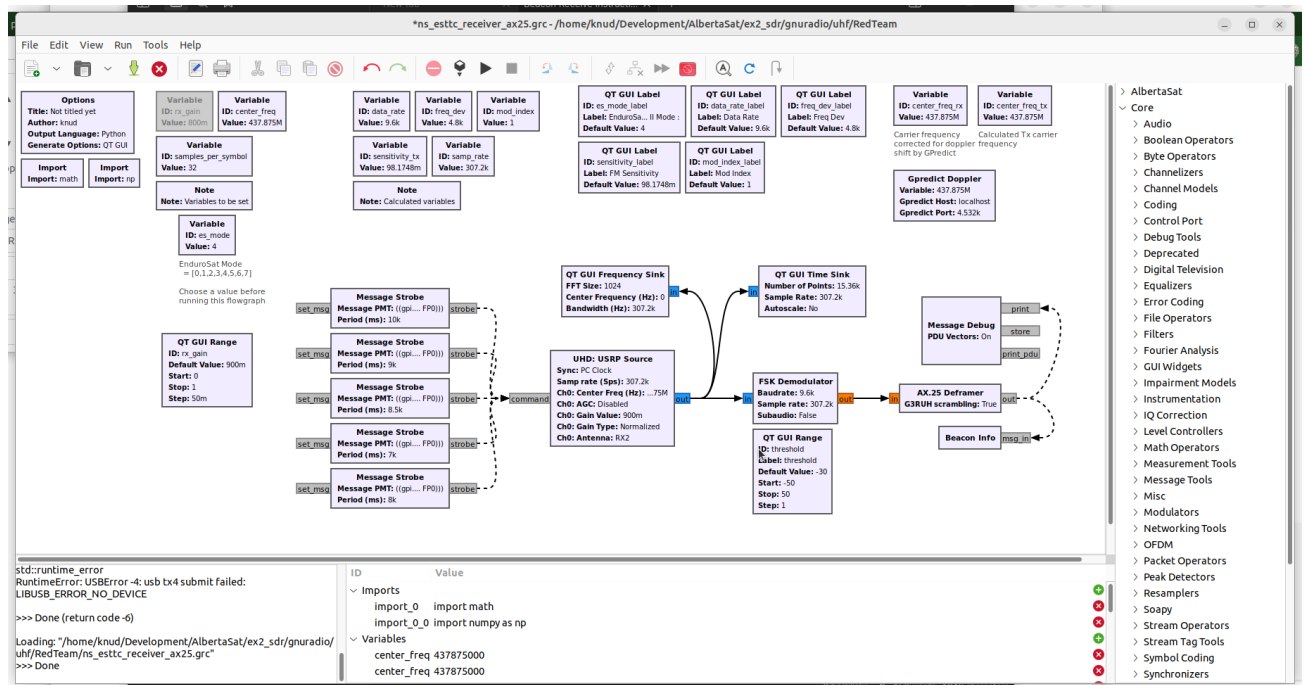


The GNU Radio flowgraph will connect to this instance of GPredict to get real-time updates for the corrected receive frequency. The GNU Radio flowgraph uses that value directly for direct receive down-conversion to baseband, and calculates the transmit frequency for transmit up-conversion from baseband.

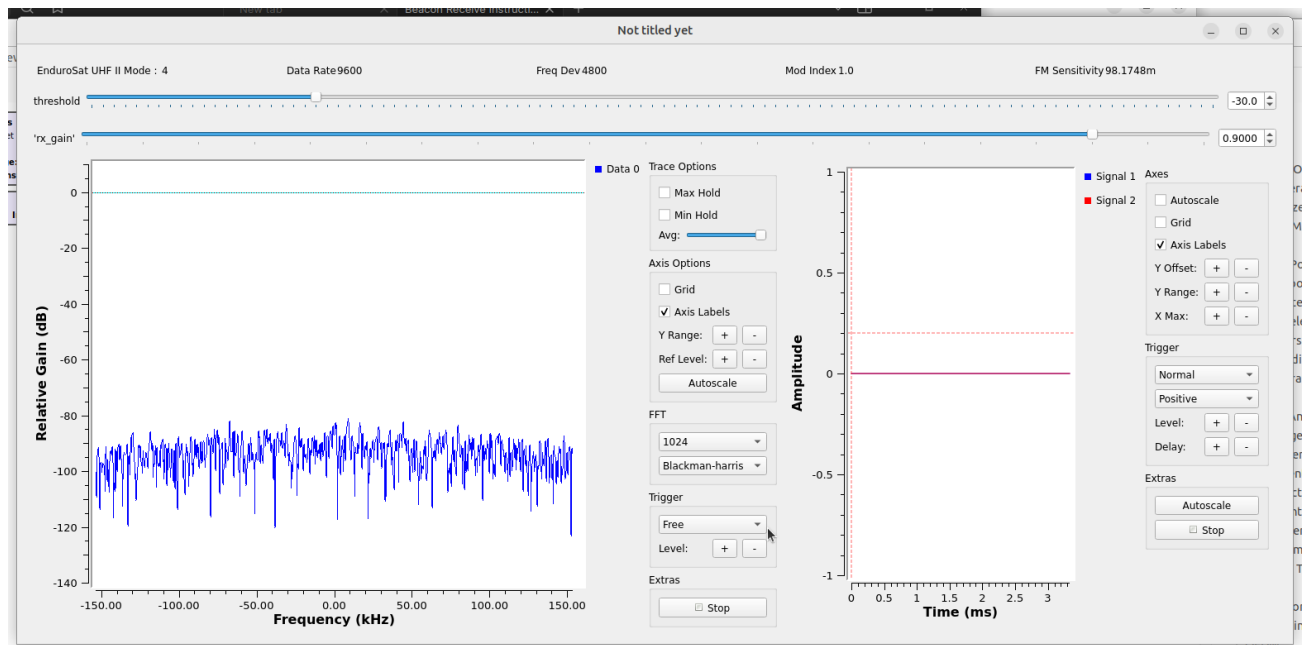
4. Open a second terminal (shortcut ctrl-alt-t)
5. Enter the commands

```
conda activate gnuradio
gnuradio-companion
```

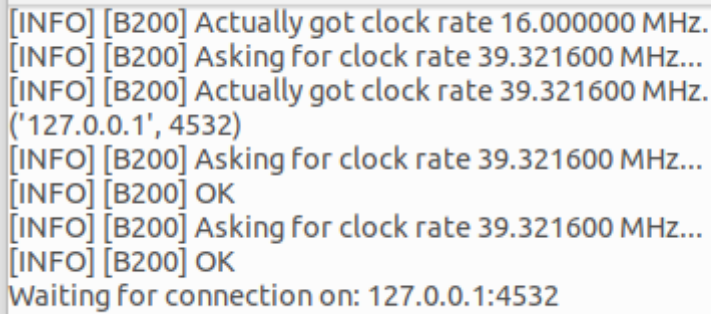
6. This will start the GNU Radio graphical application.
7. Make sure the "GPredict Doppler" block is enabled; it is near the top right in the image below. To enable it, right-click on it and select "enable"



- Run the flowgraph, but clicking the triangle start button near the top middle. The "frequency and time domain" display window shown below will appear.

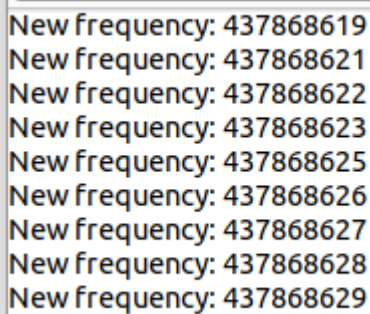


- The console at the bottom left of the main GNU Radio window should show the B200mini being loaded with firmware and at the end show the application waiting to connect to GPredict as shown in the image below



```
[INFO] [B200] Actually got clock rate 16.000000 MHz.  
[INFO] [B200] Asking for clock rate 39.321600 MHz...  
[INFO] [B200] Actually got clock rate 39.321600 MHz.  
( '127.0.0.1', 4532)  
[INFO] [B200] Asking for clock rate 39.321600 MHz...  
[INFO] [B200] OK  
[INFO] [B200] Asking for clock rate 39.321600 MHz...  
[INFO] [B200] OK  
Waiting for connection on: 127.0.0.1:4532
```

9. To connect, return to the GPredict program and in the control window click "Engage"
10. On the GNU Radio window console, you should see the frequency being updated



```
New frequency: 437868619  
New frequency: 437868621  
New frequency: 437868622  
New frequency: 437868623  
New frequency: 437868625  
New frequency: 437868626  
New frequency: 437868627  
New frequency: 437868628  
New frequency: 437868629
```

11. Now wait for beacons to be received. Any that are will be decoded and appear in the console. It's got a limited field of view, but you will be able to scroll back.
12. To halt GNU Radio, just dismiss the frequency and time domain display window.
 1. Unfortunately, GNURadio locks onto the GPredict socket, so you will have to kill GNURadio to start again.
 2. Do that by dismissing the main window and waiting for Linux to ask you to kill the program.
13. If that doesn't work, then open a new terminal and enter the command

```
ps ax
```

14. The processes running will be shown as depicted below. In this example, process number 15641 corresponds to the GNU Radio application. You can tell because the process name includes "gnuradio"

```
knud@G62-6QD:~  
File Edit View Search Terminal Help  
14949 ? Sl 2:09 /snap/obsidian/12/obsidian --type=renderer --enable-crash-reporter=1b47  
15044 ? Sl 0:18 /snap/bitwarden/90/bitwarden --no-sandbox  
15111 ? S 0:00 /snap/bitwarden/90/bitwarden --type=zygote --no-zygote-sandbox --no-san  
15112 ? S 0:00 /snap/bitwarden/90/bitwarden --type=zygote --no-sandbox  
15133 ? Sl 0:04 /snap/bitwarden/90/bitwarden --type=gpu-process --no-sandbox --enable-c  
15142 ? Sl 0:00 /snap/bitwarden/90/bitwarden --type=utility --utility-sub-type=network.  
15161 ? Sl 0:22 /snap/bitwarden/90/bitwarden --type=renderer --enable-crash-reporter=7c  
15216 ? Sl 0:10 /opt/google/chrome/chrome --type=renderer --crashpad-handler-pid=10609  
15304 ? Sl 0:03 /opt/google/chrome/chrome --type=renderer --crashpad-handler-pid=10609  
15317 ? Sl 0:00 /opt/google/chrome/chrome --type=renderer --crashpad-handler-pid=10609  
15361 ? I 0:00 [kworker/u16:3-events_unbound]  
15388 ? I 0:00 [kworker/7:0-events]  
15557 ? I 0:00 [kworker/3:2-rcu_gp]  
15615 ? I< 0:00 [kworker/u17:0-rb_allocator]  
15641 pts/1 Sl+ 4:25 /home/knud/anaconda3/envs/gnuradio/bin/python -u /home/knud/Development  
15687 ? I 0:00 [kworker/3:1-events]  
15798 ? I 0:00 [kworker/4:1-events]  
15838 ? I 0:00 [kworker/2:3]  
15858 ? I 0:00 [kworker/5:2-mm_percpu_wq]  
15859 ? I 0:00 [kworker/6:1-events]  
15995 ? I 0:00 [kworker/1:1-events]  
15996 ? I 0:00 [kworker/u16:1-events_power_efficient]  
16080 ? I 0:00 [kworker/0:1-events]  
16119 ? I< 0:00 [kworker/u17:1-rb_allocator]  
16128 ? I 0:00 [kworker/u16:0-events_power_efficient]  
16144 ? I 0:00 [kworker/7:1-events]  
16154 pts/4 Ss 0:00 bash  
16167 ? I 0:00 [kworker/1:2]  
16168 pts/4 R+ 0:00 ps ax  
(base) knud@G62-6QD:~$
```

15. To end the process, enter the `kill -9` command with the matching process number

```
kill -9 15641
```

16. If you can't find the process number easily, try the following command in the terminal

```
ps ax | grep gnuradio
```

It will filter only those processes that have something to do with gnuradio, and it's safe to kill all of them (though there should be only one.)