

Cybercore Summer Camp

NOAA Satellite Receiver

Project Overview

NOAA Satellite Receiver

Project Level

Advanced

Estimated Time:

What you'll need for this project

RTL-SDR (software defined radio)	Advanced antenna	RG 58, 174, or 316 (50 ohm coax) 10ft with sma connector
Power supply for pi when outside	3/8" acr copper tubing 25ft	Soldering iron
	3/8" acr or 1/4" plumbing elbow 8q	Some way to stand antenna up, like a tripod
	1 1/4" pvc pipe 4ft	

Additional Resources

RTL-SDR

https://www.amazon.com/gp/product/B011HUVEME/ref=as_li_tl?ie=UTF8&camp=1789&creative=9325&creativeASIN=B011HUVEME&linkCode=as2&tag=rsv0f-20&linkId=b3bd3c48a6a7e921144609cb59359f0e

Best website for code

<https://fuzzthepiguy.tech/noaa-receiver/>

<https://www.youtube.com/watch?v=RfkPUVIGBW0>

Learning about NOAA Satellites

https://www.nasa.gov/pdf/111742main_noaa_n_booklet.pdf

<https://www.youtube.com/watch?v=-cY5gXXqatY>

https://www.youtube.com/watch?v=EEQnAfo_XIA

Advanced antenna (optional)

https://www.matec-conferences.org/articles/mateconf/pdf/2018/77/mateconf_iciee2018_02002.pdf

NOAA Satellite Receiver

Step 1:

In terminal run

```
sudo apt update  
sudo apt install libusb-1.0 cmake sox at predict
```

Install RTL Software:

```
git clone https://github.com/keenerd/rtl-sdr.git  
cd rtl-sdr/  
mkdir build  
cd build
```

```
cmake ../-DINSTALL_UDEV_RULES=ON
```

```
make  
sudo make install  
sudo ldconfig  
cd ~  
sudo cp ./rtl-sdr/rtl-sdr.rules /etc/udev/rules.d/
```

Step 2:

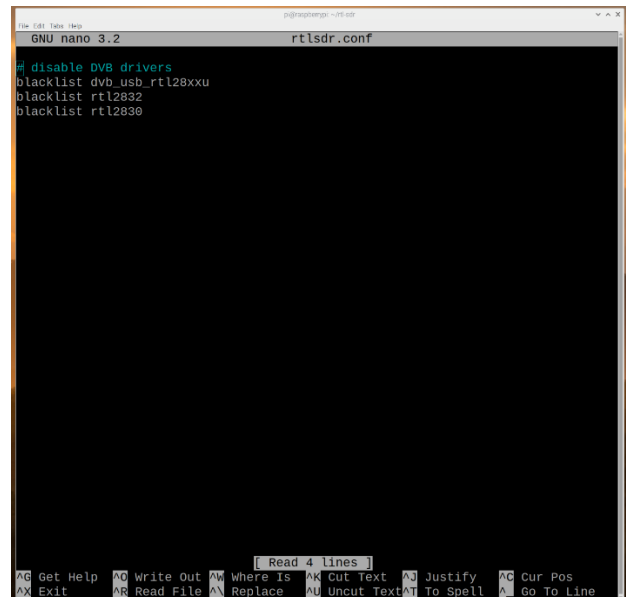
Blacklist RTL:

```
sudo nano /etc/modprobe.d/no-rtl.conf
```

```
blacklist dvb_usb_rtl28xxu  
blacklist rtl2832  
blacklist rtl2830
```

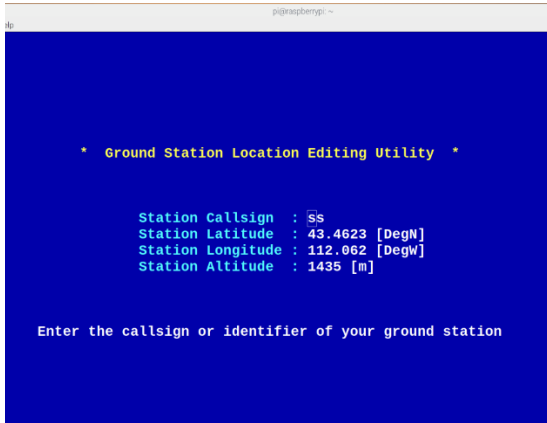
```
sudo reboot
```

(in nano use ctrl x, then y, then enter to save and exit)



```
rtlsdr@rtlsdr: ~$ nano rtl-sdr.conf  
# disable DVB drivers  
blacklist dvb_usb_rtl28xxu  
blacklist rtl2832  
blacklist rtl2830
```

NOAA Satellite Receiver

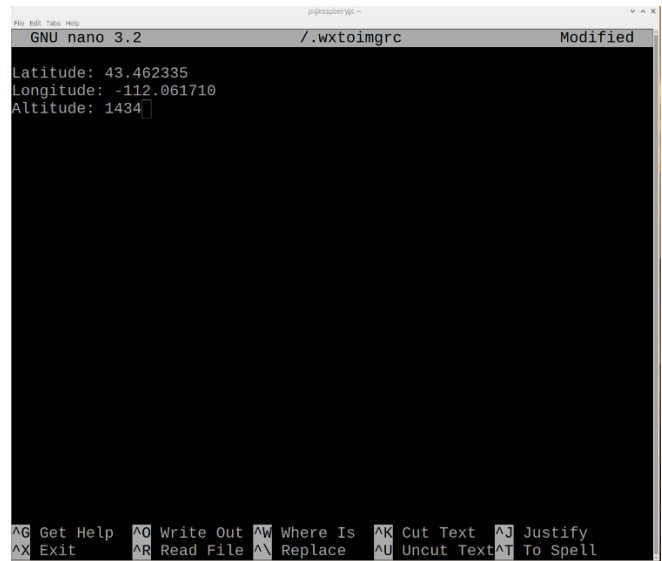
Step 3:	
Wxtoimg Install: <pre>wget https://wxtoimgrestored.xyz/beta/wxtoimg-armhf-2.11.2-beta.deb</pre> <pre>sudo dpkg -i wxtoimg-armhf-2.11.2-beta.deb</pre>	
Step 4:	
Now Run The Following Command And Select The Letter G: Now Fill In Your Info Predict Run The Following Command: And Accept The Terms Wxtoimg Open The Following File: <pre>sudo nano /.wxtoimgrc</pre>	
Step 5:	

NOAA Satellite Receiver

And Paste The Following And Update Info And Save:
Altitude Is In Meters And Not Feet:

Change to your coordinates, must match predict

Latitude: 39.0000
Longitude: -71.0000
Altitude: 75



```
GNU nano 3.2 /wxtoimgrc Modified
Latitude: 43.462335
Longitude: -112.061710
Altitude: 1434
^G Get Help  ^O Write Out ^W Where Is  ^K Cut Text  ^J Justify
^X Exit      ^R Read File ^_ Replace   ^U Uncut Text ^T To Spell
```

Step 6:

Create The Following Dir:

```
cd ~
mkdir weather
cd weather
mkdir predict
cd predict
```

Step 7:

NOAA Satellite Receiver

Make The Following File: (ctrl x, y, enter to save)

```
sudo nano schedule_all.sh
```

```
#!/bin/bash
```

```
# Update Satellite Information
```

```
wget -qr
```

```
https://www.cesstrak.com/NORAD/elements/weath  
er.txt -O /home/pi/weather/predict/weather.txt
```

```
grep "NOAA 15"
```

```
/home/pi/weather/predict/weather.txt -A 2 >
```

```
/home/pi/weather/predict/weather.tle
```

```
grep "NOAA 18"
```

```
/home/pi/weather/predict/weather.txt -A 2 >>
```

```
/home/pi/weather/predict/weather.tle
```

```
grep "NOAA 19"
```

```
/home/pi/weather/predict/weather.txt -A 2 >>
```

```
/home/pi/weather/predict/weather.tle
```

```
grep "METEOR-M 2"
```

```
/home/pi/weather/predict/weather.txt -A 2 >>
```

```
/home/pi/weather/predict/weather.tle
```

```
#Remove all AT jobs
```

```
for i in `atq | awk '{print $1}';do atrm $i;done
```

```
#Schedule Satellite Passes:
```

```
/home/pi/weather/predict/schedule_satellite.sh
```

```
"NOAA 19" 137.1000
```

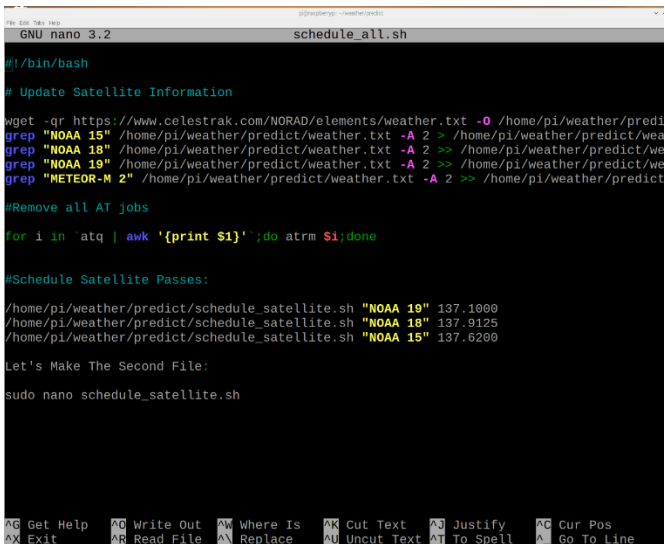
```
/home/pi/weather/predict/schedule_satellite.sh
```

```
"NOAA 18" 137.9125
```

```
/home/pi/weather/predict/schedule_satellite.sh
```

```
"NOAA 15" 137.6200
```

Inse



```
GNU nano 3.2 schedule_all.sh

#!/bin/bash

# Update Satellite Information

wget -qr https://www.cesstrak.com/NORAD/elements/weath
er.txt -O /home/pi/weather/predict/weather.txt
grep "NOAA 15"
/home/pi/weather/predict/weather.txt -A 2 >
/home/pi/weather/predict/weather.tle
grep "NOAA 18"
/home/pi/weather/predict/weather.txt -A 2 >>
/home/pi/weather/predict/weather.tle
grep "NOAA 19"
/home/pi/weather/predict/weather.txt -A 2 >>
/home/pi/weather/predict/weather.tle
grep "METEOR-M 2"
/home/pi/weather/predict/weather.txt -A 2 >>
/home/pi/weather/predict/weather.tle

#Remove all AT jobs

for i in `atq | awk '{print $1}';do atrm $i;done

#Schedule Satellite Passes:

/home/pi/weather/predict/schedule_satellite.sh "NOAA 19" 137.1000
/home/pi/weather/predict/schedule_satellite.sh "NOAA 18" 137.9125
/home/pi/weather/predict/schedule_satellite.sh "NOAA 15" 137.6200

Let's Make The Second File:

sudo nano schedule_satellite.sh
```

rt Photo

Step 8:

NOAA Satellite Receiver

Let's Make The Second File: (ctrl x, y, enter)

```
sudo nano schedule_satellite.sh

#!/bin/bash
PREDICTION_START=/usr/bin/predict -t
/home/pi/weather/predict/weather.tle -p "${1}" | head -1`
PREDICTION_END=/usr/bin/predict -t /home/pi/weather/predict/weather.tle
-p "${1}" | tail -1`

var2=`echo $PREDICTION_END | cut -d " " -f 1`

MAXELEV=/usr/bin/predict -t /home/pi/weather/predict/weather.tle -p "${1}"
| awk -v max=0 '{if($5>max){max=$5}}END{print max}`

while [ `date --date="TZ=\"UTC\" @${var2}" +%D` == `date +%D` ]; do

START_TIME=`echo $PREDICTION_START | cut -d " " -f 3-4`
var1=`echo $PREDICTION_START | cut -d " " -f 1`

var3=`echo $START_TIME | cut -d " " -f 2 | cut -d ":" -f 3`

TIMER=`expr $var2 - $var1 + $var3`

OUTDATE=`date --date="TZ=\"UTC\" $START_TIME" +%Y%m%d-%H%M%S`

if [ $MAXELEV -gt 19 ]
then
echo ${1/" "} ${OUTDATE} $MAXELEV

echo "/home/pi/weather/predict/receive_and_process_satellite.sh \"${1}\""
$2 /home/pi/weather/${1/" "} ${OUTDATE}
/home/pi/weather/predict/weather.tle $var1 $TIMER" | at `date
--date="TZ=\"UTC\" $START_TIME" +%H:%M %D`"

fi

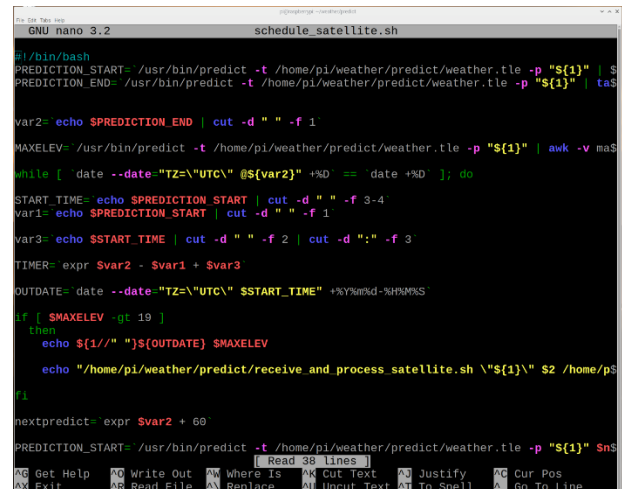
nextpredict=`expr $var2 + 60`

PREDICTION_START=/usr/bin/predict -t
/home/pi/weather/predict/weather.tle -p "${1}" $nextpredict | head -1`
PREDICTION_END=/usr/bin/predict -t /home/pi/weather/predict/weather.tle
-p "${1}" $nextpredict | tail -1`

MAXELEV=/usr/bin/predict -t /home/pi/weather/predict/weather.tle -p "${1}"
$nextpredict | awk -v max=0 '{if($5>max){max=$5}}END{print max}`

var2=`echo $PREDICTION_END | cut -d " " -f 1`

done
```



```
GNU nano 3.2 schedule_satellite.sh

#!/bin/bash
PREDICTION_START=/usr/bin/predict -t /home/pi/weather/predict/weather.tle -p "${1}" | S
PREDICTION_END=/usr/bin/predict -t /home/pi/weather/predict/weather.tle -p "${1}" | ta

var2=`echo $PREDICTION_END | cut -d " " -f 1`

MAXELEV=/usr/bin/predict -t /home/pi/weather/predict/weather.tle -p "${1}" | awk -v ma$
while [ `date --date="TZ=\"UTC\" @${var2}" +%D` == `date +%D` ]; do

START_TIME=`echo $PREDICTION_START | cut -d " " -f 3-4`
var1=`echo $PREDICTION_START | cut -d " " -f 1`

var3=`echo $START_TIME | cut -d " " -f 2 | cut -d ":" -f 3`

TIMER=`expr $var2 - $var1 + $var3`

OUTDATE=`date --date="TZ=\"UTC\" $START_TIME" +%Y%m%d-%H%M%S`

if [ $MAXELEV -gt 19 ]
then
echo ${1/" "} ${OUTDATE} $MAXELEV
echo "/home/pi/weather/predict/receive_and_process_satellite.sh \"${1}\"" $2 /home/p$
fi

nextpredict=`expr $var2 + 60`

PREDICTION_START=/usr/bin/predict -t /home/pi/weather/predict/weather.tle -p "${1}" $n$
[ Read 38 lines ]
AG Get Help  AG Write Out  AW where Is  AK Cut Text  AJ Justify  AC Cur Pos
AX Exit      AR Read File  AN Replace  AU Uncut Text  AT To Spell  AL Go To Line
```

Step 9:

NOAA Satellite Receiver

Let's Make The Last One:

```
sudo nano receive_and_process_satellite.sh
```

```
#!/bin/bash
```

```
# $1 = Satellite Name
```

```
# $2 = Frequency
```

```
# $3 = FileName base
```

```
# $4 = TLE File
```

```
# $5 = EPOC start time
```

```
# $6 = Time to capture
```

```
sudo timeout $6 rtl_fm -f ${2}M -s 60k -g 45 -p 55 -E  
wav -E deemp -F 9 - | sox -t wav - $3.wav rate 11025
```

```
PassStart=`expr $5 + 90`
```

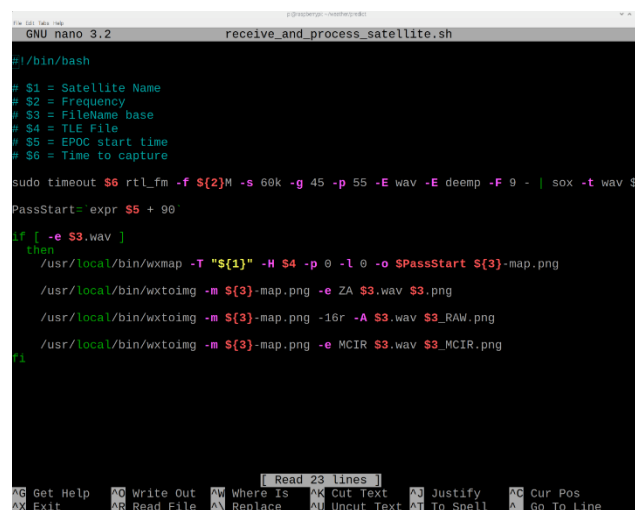
```
if [ -e $3.wav ]
```

```
then
```

```
    /usr/local/bin/wxmap -T "${1}" -H $4 -p 0 -l 0 -o  
    $PassStart ${3}-map.png
```

```
    /usr/local/bin/wxtoimg -m ${3}-map.png -e ZA  
    $3.wav $3.png
```

```
Fi
```

A screenshot of a terminal window with the title 'receive_and_process_satellite.sh'. The terminal shows the GNU nano 3.2 editor with the script content. The script includes variable definitions for satellite name, frequency, filename base, TLE file, EPOC start time, and time to capture. It uses 'sudo timeout' to run 'rtl_fm' and 'sox' to process the audio. It then checks if a file exists and uses 'wxmap' and 'wxtoimg' to generate map and image files. The terminal window has a standard Linux desktop background and a taskbar at the bottom.

Step 10:

Run Each Line One At A Time:

```
sudo chmod +x schedule_all.sh
```

```
sudo chmod +x schedule_satellite.sh
```

```
sudo chmod +x receive_and_process_satellite.sh
```

Step 11:

NOAA Satellite Receiver

Run The Following Command:

To Make A Cron Job:

```
crontab -e
```

Add The Following And Save:

```
1 0 * * * /home/pi/weather/predict/schedule_all.sh
```

Run The Following Command:

```
/home/pi/weather/predict/schedule_all.sh
```

Check To See If ATQ Is Working:

```
Atq
```

Finished

Step 12:

Extra add on edit script to add 2 more lines (optional)

https://www.youtube.com/watch?v=1xA_VUADKZ0

```
#!/bin/bash
```

```
# $1 = Satellite Name
```

```
# $2 = Frequency
```

```
# $3 = FileName base
```

```
# $4 = TLE File
```

```
# $5 = EPOC start time
```

```
# $6 = Time to capture
```

NOAA Satellite Receiver

```
sudo timeout $6 rtl_fm -f ${2}M -s 60k -g 45 -p 0 -E wav -E deemp -F 9 - | sox -t wav - $3.wav rate 11025
```

```
PassStart=`expr $5 + 90`
```

```
if [ -e $3.wav ]
```

```
then
```

```
/usr/local/bin/wxmap -T "${1}" -H $4 -p 0 -l 0 -o $PassStart ${3}-map.png
```

```
/usr/local/bin/wxtoimg -m ${3}-map.png -e ZA $3.wav $3.png
```

```
/usr/local/bin/wxtoimg -m ${3}-map.png -16r -A $3.wav $3_RAW.png
```

```
/usr/local/bin/wxtoimg -m ${3}-map.png -e MCIR $3.wav $3_MCIR.png
```

```
Fi
```



NOAA Satellite Receiver

