

Getting Started with Data Reporting Using A Personal Space Weather Station

Ver. 7.2

W. Engelke, AB4EJ

Nov. 2023

Table of Contents

Introduction.....	2
Step 1. Creating an Account on the Central Control System.....	2
Step 2. Creating a Station.....	5
Step 3. Creating an Instrument.....	7
Using the Grape1-DRF Prebuilt Image.....	11
Introduction.....	11
About the System and Its UI.....	11
Principles of Operation.....	12
Some key points about how the system works.....	13
Using the Grape1 DRF System.....	14
Errors During System Restart:	15
Uploading.....	16
How the gnuradio scheduler works.....	19
Configuration file (uploader.config).....	20
Here is a configuration file defining the settings:	21
Example working configuration file:	22
Hardware Architecture.....	23
How Gnuradio Is used	24
An Introduction to Digital RF.....	27

We are grateful to the United States National Science Foundation (NSF) Division of Atmospheric and Geospace Sciences for support of this work through grants AGS-1932972, AGS-1932997, AGS-2002278, AGS-2230345, and AGS-2230346.

Introduction

If you have built a Personal Space Weather Station (including a Grape or TAPR magnetometer), your next step is to set up your system to upload your collected data to the PSWS Central Control System, where it can be included with other collected data for scientific analysis. Follow the instructions below and you'll be uploading data in no time.

Step 1. Creating an Account on the Central Control System

First, you need to create an account on the Central Control System (there is no charge). Using your browser, connect to: <https://pswsnetwork.caps.ua.edu> .¹



Your first step is to create a User-ID. Click the **Register your station** button; you will see the Sign Up Form:

¹ This will eventually change to psws.hamsci.org once we get the DNS issues figured out.

Personal Space Weather Station

Central Control System

Home	Stations	Observations	Users	Log In	Register your station	About
------	----------	--------------	-------	--------	-----------------------	-------

Sign up Form

IMPORTANT: This is a public project funded by the National Science Foundation. By signing up on this website as a Personal Space Weather Network user, you attest: You are at least 18 years of age. You give consent for the project to use any data you enter or upload in any way. You recognise that you have no expectation of privacy and should not enter anything you deem personal or private. (Amateur radio operators should enter the same information as what they have on file with the FCC.) The project will make a reasonable effort to maintain the integrity of the web site, but this cannot be guaranteed. If the web site or database is breached, no assurances are given that you will be notified.

Username:

Email:

Password:

Your password can't be too similar to your other personal information,Your password must contain at least 8 characters,
Your password can't be a commonly used password,your password can't be entirely numeric.

Password confirmation:

Sign up

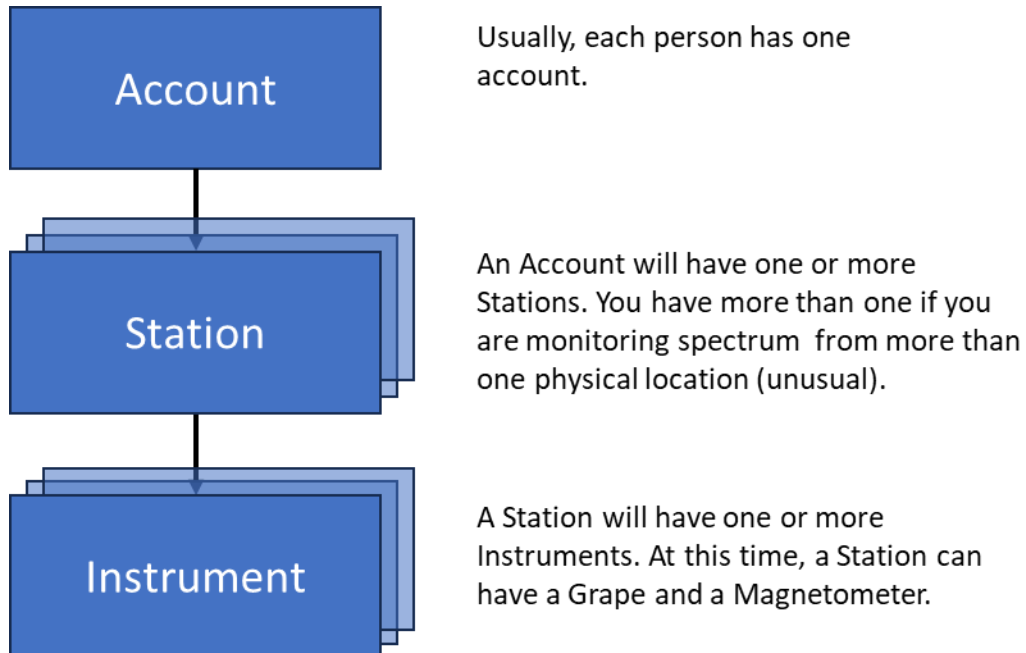
Invent a Username that you can remember (amateur radio operators commonly use their callsign here). Enter that with your email, and a complex password (entered twice), and click the Sign up button.

In a few minutes, you should receive an email at that email address asking you to verify your email address; click the link. You should now be able to log in with that User-ID and password. You can also see your User-ID and email address on the Users page.

Contact the webmaster here if you run into problems trying
to create your Account:
bengelke@cs.ua.edu.

Logical Layout of Account/Station/Instrument

Here is the relationship between Account, Station(s) and Instrument(s):



Step 2. Creating a Station

Now that you have an account, it is time to create a Station.

From the Home Page, click the Stations button. You will see a screen like the following, which is a list of existing Stations.

Personal Space Weather Station
Central Control System
Stations

Home **Stations** Observations Users Log In Register your station About

[View your stations](#)
[Add new station](#)

ID	User	Nickname	Grid	Elevation	Station status
N000003	ab4ej-3	N000003_A	EM63fj	112.0	Online
N000004	ab4ej-3	Grape N000019	EM63fj	100.0	Possiblyonline
N000015	w2naf	W2NAF	FN21ei	507.0	Offline
N000016	ab4ej-3	N000016	EM63fj	110.0	Online

[View your stations](#)
[Add new station](#)

To add a station, click the [Add new station](#) link. Now you will see a form you can fill out to create a station (next page).

If you have privacy concerns, you may omit any information you seem sensitive; however, we do have to have a Maidenhead grid square² of at least 4 characters (6 is better).

² Maidenhead grid square is a 4- or 6-character location indicator that maps to latitude/longitude. You can find out information online on how to determine your grid square from your latitude and longitude.

Central Control System

Add Station Form

Home	Stations	Observations	Users	Log In	Register your station	About
----------------------	-----------------	------------------------------	-----------------------	------------------------	---------------------------------------	-----------------------

IMPORTANT: All information on this page except for station nickname and Maidenhead grid square is optional. For grid square, we need a minimum of a 4-character grid (e.g., EM64). If you wish to omit entry of data in any field, leave it blank. Anything else you enter becomes public information. (Amateur radio operators: we suggest you use what is on file with the FCC, as it is already in the public record). We collect this information so that we can give you credit if your data is used in publications such as peer-reviewed journals. Mandatory fields are marked with an asterisk (*).

Nickname: *

Maidenhead Grid Square: *

Elevation:

 meters above sea level

Antenna 1:

Antenna 2:

Street address:

City:

State:

Postal code:

Phone number:

All fields on the Add Station form are optional except for Nickname and Maidenhead grid square. Fill out the form and click the [Add Station](#) button. You should now see your newly added station at the bottom of the Stations list.

Step 3. Creating an Instrument

Any device that collects data for the PSWS network is generally referred to as an “Instrument.” This includes the Grape and (other supported SDRs like Tangerine and/or Clementine and/or rx888), and the TAPR magnetometer, plus new future instruments in concept development such as an Extremely Low Frequency (ELF) receiver, a Jovian radio receiver, and other geomagnetic or geological data collectors such as seismometers, lightning and meteor detectors, etc. Once you have created a station, you can add instruments to it.

To do this, click the [View your stations](#) link. This will show you the details of your station, with a list of instruments at the bottom. Here is an example:

Personal Space Weather Station Central Control System ab4ej-3's Stations

Home	Stations	Observations	Users	Log In	Register your station	About
------	----------	--------------	-------	--------	-----------------------	-------

[View all registered stations](#)

[Add new station](#)

To add a new instrument, click on the station's ID.

To look at the details of a station (or edit it), click on its link in the ID column.

ID	Nickname	Grid	Elevation	Antenna1	Antenna2	Station status
N000003	N000003_A	EM63fj	112.0	—	—	Online
N000004	Grape N000019	EM63fj	100.0	—	—	Possiblyonline
N000016	N000016	EM63fj	110.0	—	—	Online
N000017	test_sta	EM66	—	—	—	Offline

[View all registered stations](#)

[Add new station](#)

This user (above) has 4 stations. You create multiple stations if you will be collecting data from multiple locations (might be done by school or club); location is saved independently for each station. To add a new instrument, determine which Station this instrument is to be a part of and click the ID link (of that station) on the left (e.g., [N000003](#)).

Now you will see the details about this station. You can edit the station details by clicking the [Edit Station Details \(this station\)](#) link, or, in this case you can add an instrument by clicking the [Add New Instrument](#) link.

Personal Space Weather Station

Central Control System

test_sta

Home	Stations	Observations	Users	Log In	Register your station	About
------	----------	--------------	-------	--------	-----------------------	-------

[Back to your stations](#)

[Edit Station Details \(this station\)](#)

[Add New Instrument to this station](#)

Details of this station:

- ID: N000017
- Token: 8888970c6da3ab9a37e9
- User: ab4ej-3
- Latitude: 36.500
- Longitude: -87.000
- Grid: EM66
- Elevation: None
- Antenna 1:
- Antenna 2:
- Street Address:
- City:
- State:
- Postal Code:
- Date Created: July 15, 2022, 5:19 p.m.

Below is a list of instruments of this station. To edit details of an instrument, click on its ID.

ID	Instrument	Nickname	SerialNo	InstrumentType
----	------------	----------	----------	----------------

[Edit Station Details \(this station\)](#)

[Add New Instrument](#)

Next, you will see the Add New Instrument screen (below). Here, there are just two mandatory fields: the Instrument (its name), and the InstrumentType (this drop down lets you pick from a list of the supported instruments). Optionally, you may log the date you added this instrument, its nickname and serial number (if applicable), and its status (e.g., in service, out of service, etc.). Click the [Add Instrument](#) button to add the new instrument. Once you have added your new instrument, the system will take you back to the Station Details page where you will see the instrument listed in the table at the bottom of the page.

Personal Space Weather Station Central Control System Add Instrument Form

Home	Stations	Observations	Users	Log In	Register your station	About
------	----------	--------------	-------	--------	-----------------------	-------

Instrument: *

Date instrument added:

 format YYYY-MM-DD

Nickname:

SerialNo:

Status:

Instrumenttype: *

After you have filled out this form and clicked the “Add Instrument” button, you will see a screen that shows you the details about this station, but you should see your just-added instrument in the list, like so:

[Back to your stations](#)

[Edit Station Details \(this station\)](#)

[Add New Instrument to this station](#)

Details of this station:

- ID: N000017
- Token: 8888970c6da3ab9a37e9
- User: ab4ej-3
- Latitude: 36.500
- Longitude: -87.000
- Grid: EM66
- Elevation: None
- Antenna 1:
- Antenna 2:
- Street Address:
- City:
- State:
- Postal Code:
- Date Created: July 15, 2022, 5:19 p.m.

Below is a list of instruments of this station. To edit details of an instrument, click on its ID.

ID	Instrument	Nickname	SerialNo	InstrumentType
27	my_new_instrument	instrument1	—	Magnetometer

[Edit Station Details \(this station\)](#) [Add New Instrument](#)

Using the Grape1-DRF Prebuilt Image

Introduction

This section focuses on the content of the Grape1-DRF pre-built image. The image is aimed mainly at our effort to collect good data for the 2023 and 2024 solar eclipse events. The intent of this document is to explain some things about how the Grape1-DRF package works and how the configuration is set up (note: the pre-built systems have configurations already set up for each participating user, so this is to help in case of errors or omissions in the prebuilt config file).

About the System and Its UI

The Grape1-DRF pre-built image has only the most rudimentary user interface; it is designed to be as simple as possible, to concentrate on its primary mission of collecting spectrum data for upcoming eclipses (although, this being solar maximum, you can be assured of seeing a lot of remarkable Doppler shifts on typical days).

The best way to start working with the system is to connect everything in advance:

- Antenna
- External disk drive (the system requires a powered hard drive which is formatted ext4 and named G1DRF)
- Sound card chip
- Leo Bodnar GPSDO with antenna mounted outside
- Wired network connection (preferred); WiFi will work but you will have to find your network and enter your wifi password
- Powered USB hub is preferred

When you power on the system, it will attempt to boot, and immediately start collecting data. You should see a display with two features: a slow-updating waterfall and a faster updating spectrum display. When WWV's signal is strong enough, you should see a spike in the middle of the spectrum display. (The waterfall updates one line every 110 seconds; the spectrum display updates once every 3 to 4 seconds).

If you try to boot and the system goes into “**Emergency Mode**,” this is usually because your disk drive is not plugged in, not working, powered off, not formatted **ext4** or not named **G1DRF**.

The time zone of the system is set to UTC and shown in 24 hour format. You may change this using Settings; the system will continue to collect data time-stamped with correct UTC.

Principles of Operation

The Grape1 Digital RF, “Grape1-DRF”, version (not to be confused with the original Grape1 which collected data using the fldigi application) collects data using the gnuradio application, saves it in DRF format and uploads the data to the Personal Space Weather Station Network.

All the prebuilt systems use the same Linux User-ID of “ubuntu” and the same password of “HamSCI2023!”³; you may change the system password if you like, but you must run the system under the preconfigured user-ID of “ubuntu”. (If you change the password and then forget it, we can’t help you; you will have to re-burn the SD card from the original system image). Also, please leave the structure and naming of the directories untouched: the software makes assumptions about where files are located based on hard-coded paths⁴.

The system has the necessary libraries installed. All the code necessary to run the system is in a single directory at: `/home/ubuntu/Documents/runGrape`. Here is a list of the files in that directory and what they do (how the package actually works is explained in the next section). This list is alphabetical, not related to the order in which things happen...

<code>__pycache__</code>	Directory to hold pseudocode for gnuradio, don’t touch this
<code>get_datetime.py</code>	called by gnuradio when gnuradio starts, this program gets the system date and time, plus configuration info like latitude and longitude to be put into DRF metadata when creating a DRF dataset
<code>gnuradio_schedule.py</code>	You run this program under python3 to start the system; it kicks everything off
<code>grape.log</code>	keeps a log file of what happens when the system starts to aid in debugging
<code>grape_DRF_spec.grc</code>	this is a flowgraph that can be opened in gnuradio to change how data collection works (don’t change this unless you know what you are doing). For the <code>get_datetime</code> program to work, you must start <code>gnuradio-companion</code> or <code>grape_DRF_spec.py</code> from the <code>runGrape</code> directory.
<code>grape_DRF_spec.py</code>	A python program created by gnuradio which runs data collection. It is the python equivalent of <code>grape_DRF_spec.grc</code>
<code>grape_restart.py</code>	Restarts the system after it is stopped at 00:00Z (UTC) to update the date, start a new observation, and upload data.

³ The exclamation point is part of the password.

⁴ Crude, yes, but this is Version 1.

grape_trigger.py	Sets up the day's observation and starts gnuradio
killprocess.py	Stops gnuradio in case it is running so that we don't have multiple copies running at once (which is harmless, except for eating CPU cycles)
readme.txt	A summary of how to use the system
uploader.config	Configuration file for Grape1-DRF
uploader_V3_Grape1.py	Program to upload DRF data in most recently acquired observation

Some key points about how the system works

- Every day, a new observation is automatically started at 00:00Z (UTC), and is named: OBS-YYYY-MM-DDTHH-MM. (This is to avoid generating multi-day observations which are unwieldy in size). The system will attempt to upload data from the previous observation to the server. The scheduler shows a continuously updating display of the system clock when it is running the schedule. If you halt the system manually by clicking the halt (X) button (upper right corner of waterfall), the scheduler will continue to run; click in the terminal window and use CTRL-C to quiesce the system.
- The uploader uses lftp to mirror the local data drive observations to the PSWS server. We recommend that you do not store other data on your G1DRF drive, because under some odd circumstances it may end up being mirrored up to the PSWS server. The mirror function usually works well because it handles it gracefully when you have starting and stopping of data collection and uploading; it knows if something is redundant and usually won't duplicate or reload files it has already uploaded.
- Data transfer to the server uses secure ftp (inside lftp), so your data is secure (as long as you keep your station token private to yourself).
- The system will synchronize time with NTP as soon as it is connected to a network.
- The data collection system **starts automatically** after power-up or reboot. This is so that it can recover after a power failure. If you want to start the system but not have data collection active, click on the terminal window and hit CTRL-C immediately after system auto-start That will halt the session.

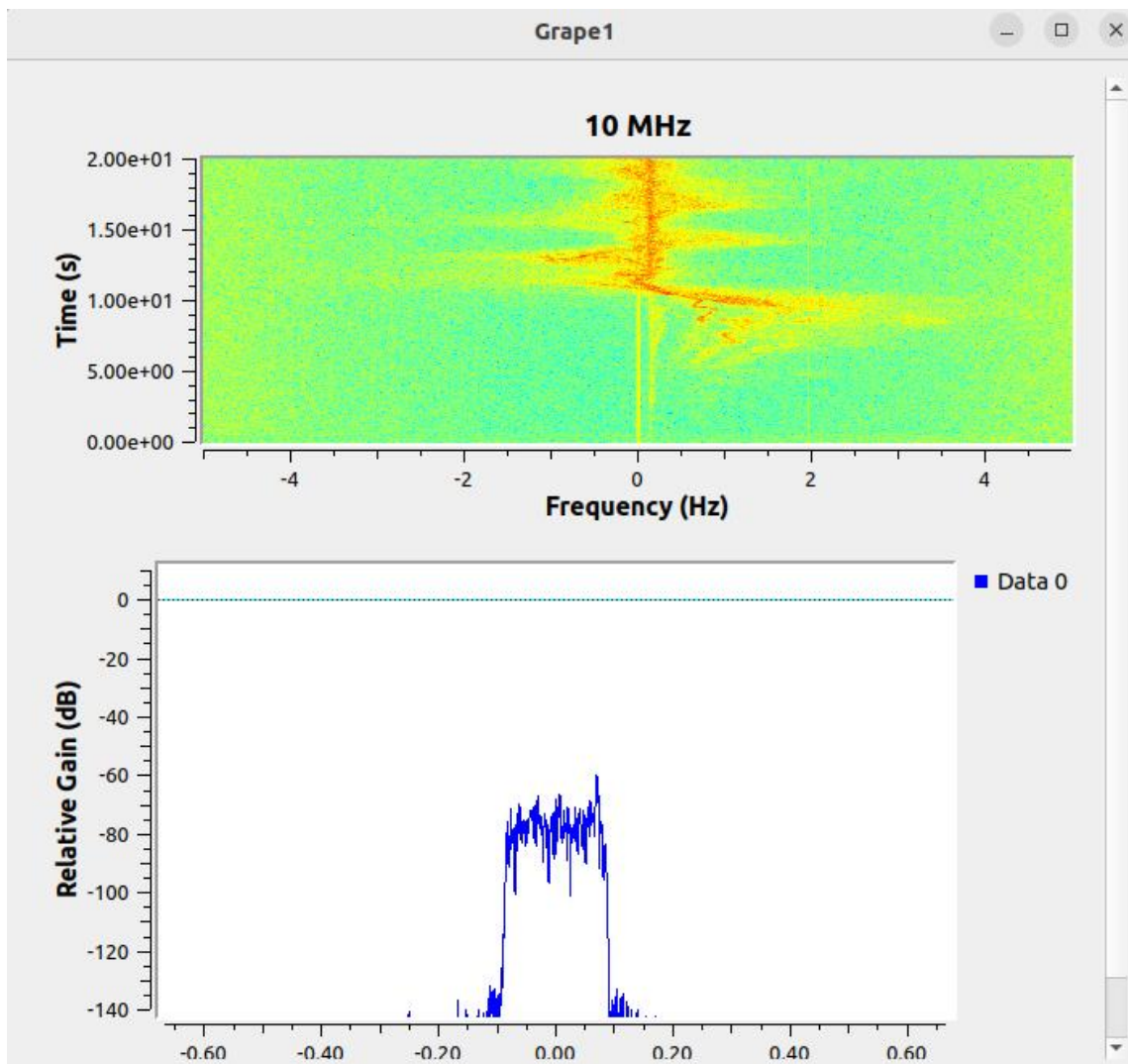
Using the Grape1 DRF System

The system should start automatically upon power-up. If you want to start it yourself, use:

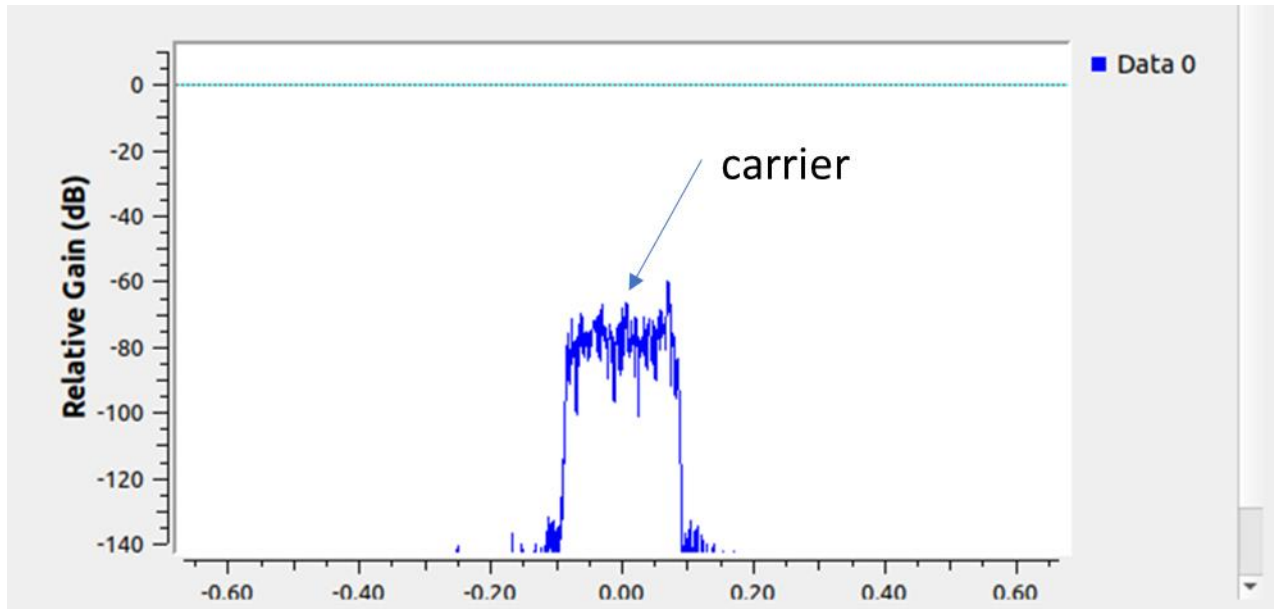
```
cd /home/ubuntu/Documents/runGrape  
  
python3 gnuradio_schedule.py
```

If you use this in a terminal window and the system is already running in a different window, the startup process will kill (halt) any other Grape data collection processes as part of the startup.

When the system starts, you will see a screen like the following. This figure shows a typical sunrise in the waterfall (top)



Note that the bottom trace (frequency spectrum) is 8kHz wide (although the x-axis labeling is not correct). There is often some noise present, which can make the WWV carrier (in the middle of the spectrum) hard to see, especially when the band you are monitoring does not have strong propagation. The figure below shows where the carrier is:



Errors During System Restart:

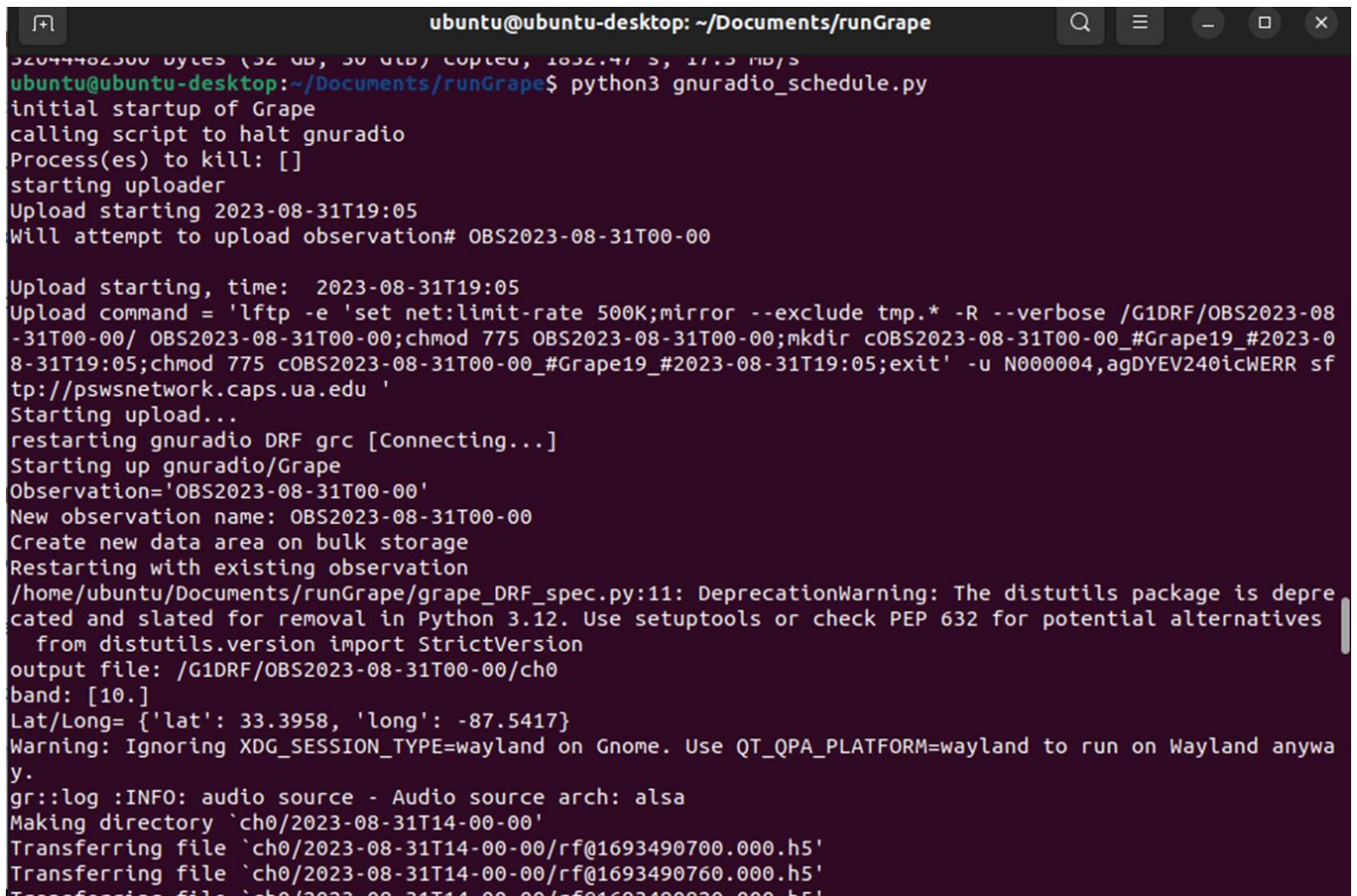
If you stop data collection and then try to restart it again *in the same minute*, you may get a Digital RF error message to the effect that it can't write over the file. (Recall that the system records data in files of one minute each). This is a feature⁵ included in DRF to minimize data loss during anomalous events such as power loss. If this happens, just wait a minute and start the system again.

⁵ Whether something is a feature or a bug has been a continuous topic of conversation about many things since the dawn of the computer age. I don't know, I just work here.

Uploading

The normal way of uploading collected data is that the system tries to upload after it restarts every day at 00:00Z. If you restart the system (reboot, or stop and start `gnuradio_schedule.py`) it will upload what it has collected so far this day since 00:00Z. This is a “mirroring” operation, so that further upload sessions after that will not re-send data that was already successfully sent.

The figure below shows where the system has been restarted and uploaded the pending data. At the top you see the invocation of `gnuradio_scheduler`, the start of the upload, the restart of `gnuradio`, and at the bottom the start of the list of files being uploaded. There will be a lot of files: the system stores one file for each minute of data (meaning 1,440 per day)⁶.



```
ubuntu@ubuntu-desktop: ~/Documents/runGrape
3204482300 bytes (32 MB, 30 GB) copied, 1032.47 s, 17.5 MB/s
ubuntu@ubuntu-desktop:~/Documents/runGrape$ python3 gnuradio_schedule.py
initial startup of Grape
calling script to halt gnuradio
Process(es) to kill: []
starting uploader
Upload starting 2023-08-31T19:05
Will attempt to upload observation# OBS2023-08-31T00-00

Upload starting, time: 2023-08-31T19:05
Upload command = 'lftp -e 'set net:limit-rate 500K;mirror --exclude tmp.* -R --verbose /G1DRF/OBS2023-08-31T00-00/ OBS2023-08-31T00-00;chmod 775 OBS2023-08-31T00-00;mkdir cOBS2023-08-31T00-00_#Grape19_#2023-08-31T19:05;chmod 775 cOBS2023-08-31T00-00_#Grape19_#2023-08-31T19:05;exit' -u N000004,agDYEV240icWERR sf tp://pswsnetwork.caps.ua.edu '
Starting upload...
restarting gnuradio DRF grc [Connecting...]
Starting up gnuradio/Grape
Observation='OBS2023-08-31T00-00'
New observation name: OBS2023-08-31T00-00
Create new data area on bulk storage
Restarting with existing observation
/home/ubuntu/Documents/runGrape/grape_DRF_spec.py:11: DeprecationWarning: The distutils package is deprecated and slated for removal in Python 3.12. Use setuptools or check PEP 632 for potential alternatives
  from distutils.version import StrictVersion
output file: /G1DRF/OBS2023-08-31T00-00/ch0
band: [10.]
Lat/Long= {'lat': 33.3958, 'long': -87.5417}
Warning: Ignoring XDG_SESSION_TYPE=wayland on Gnome. Use QT_QPA_PLATFORM=wayland to run on Wayland anyway.
gr::log :INFO: audio source - Audio source arch: alsa
Making directory `ch0/2023-08-31T14-00-00'
Transferring file `ch0/2023-08-31T14-00-00/rf@1693490700.000.h5'
Transferring file `ch0/2023-08-31T14-00-00/rf@1693490760.000.h5'
Transferring file `ch0/2023-08-31T14-00-00/rf@1693490820.000.h5'
```

At the end of uploading, the PSWS network detects the completion of the upload, whereupon it adds the upload to the database, and you can see it on the website under

⁶ The one file per minute is a trade-off between # files, system performance, and ability to recover from error conditions.

Observations (following figure). Observations are listed with the most recent at the top of the list.

Central Control System

Observations

Home	Stations	Observations	Analysis	Users	Log In	Register your station				
☰ Open Filter To download observation data, click on File/Observation link. Please be patient, it may take a while to										
Data rate	Center Frequency	Station	Instrument	Size (MB)	File/Observation					
1	—	Mag_kv0s4	Mag_rev_E	0.6628	OBS2023-09-01T00:00.zip					
1	—	W2NAF	GMAG	0.9346	OBS2023-09-01T00:00.zip					
1	—	AB4EJ-mag1	Magnetometer2	0.4340	OBS2023-09-01T00:00.zip					
10	—	AB1XB Node 33	Grape1.0	0.0876	OBS2023-09-01T00-00					
10	—	W7LUX	Grape66	0.1977	OBS2023-09-01T00-00					
10	—	AB1XB Node 33	Grape1.0	34.1548	OBS2023-08-31T00-00					
10	—	AB4EJ-Grape1	Grape19	27.3327	OBS2023-08-31T00-00					
10	—	KD1LE node 43	Grape1	34.1620	OBS2023-08-31T00-00					
1	2	3	4	5	6	7	8	...	162	next >

You can filter this list by clicking the Open Filter button (upper left).

(This illustration shows the Center Frequency as just dashes, which is a bug that should be corrected before the system goes into full production).

Select options below to filter:

Instrument Type:

- Grape
- Tangerine
- Magnetometer

Center Frequency:

- 10.000 MHz
- 5.000 MHz
- 2.500 MHz
- 15.000 MHz

Station Nickname:

Start Date (UTC):

mm / dd / yyyy

End Date (UTC):

mm / dd / yyyy

Latitude Range:

[-90, 90] - [-90, 90]

Longitude Range:

[-180, 180] - [-180, 180]

Filter

Here we see the filter sub-screen. If you click on a field such as the Instrument type, center frequency, station nickname, etc., then click the Filter button (bottom left), the system will show in the list only those Observations that meet the search criteria.

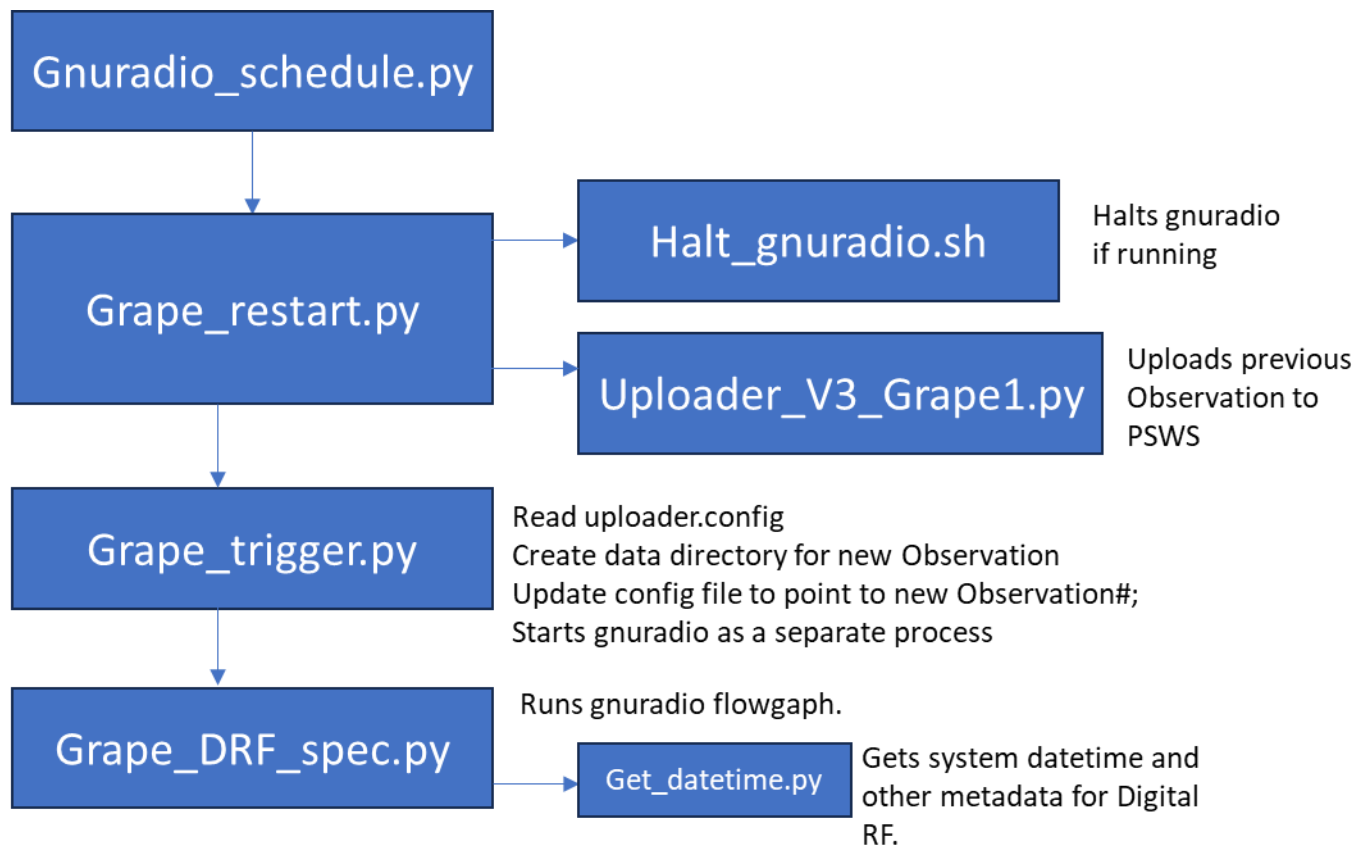
If you wish to download the data of an Observation, click on the File/Observation (blue link); the system will attempt to download that observation's data to your computer, where it will appear in the Downloads folder. Be aware that some of the Observations (especially from early days) can be enormous.

How the gnuradio scheduler works

From the directory `/home/ubuntu/Document/runGrape`

... you key in: `python3 gnuradio_schedule.py`

Here is the order in which things happen:



Configuration file (uploader.config)

In the configuration below, anything that is in Times font (like this sentence) is documentation and *is not to be included in the actual configuration file*.

For reference, here is the Station details screen showing where the token (token_value), station ID (thestationid) and the instrument ID (instrumentid) come from (used in the configuration file):

Central Control System
AB4EJ-mag1

Home Stations Observations Analysis Users Log In Register your station About

[Back to your stations](#)
[Edit Station Details \(this station\)](#)
[Add New Instrument to this station](#)

Details of this station:

- ID: N000003
- Nickname: AB4EJ-mag1
- Token: ymbd25[REDACTED]c3
- User: ab4ej-3
- Latitude: 33.354
- Longitude: -87.542
- Grid: EM63fi
- Elevation: 112.0
- Antenna 1: Hexbeam
- Antenna 2: Dipole
- Street Address: Carmel Bay Dr
- City: NORTHPORT
- State: AL
- Postal Code: 35475
- Date Created: March 29, 2022, 7:39 p.m.

Below is a list of instruments of this station. To edit details of an instrument, click on its ID.

ID	Instrument	Nickname	SerialNo	InstrumentType
1	Magnetometer2	MAG1	none	Magnetometer
72	Grape1.12	—	—	Grape

Here is a configuration file defining the settings:

[profile]

```
token_value = <your token from your station in PSWS network>
node = <Node number, if assigned – not used for anything now>
grid = EM63fj <your grid>
prefix = <station prefix, usually callsign, used only for stations with a
        magnetometer>
thestationid = <your station ID>
central_host = pswsnetwork.caps.ua.edu
```

[mag_settings] (ignore all this if not using magnetometer)

```
obs = OBS2023-03-22T21:52
instrumentname = Mag3
throttle = 100K
orig_path = /home/ab4ej/projects/rm3100-runMag/logs
temp_path = /home/ab4ej/projects/rm3100-runMag/xferData
upload_path = /media/ab4ej/seagate2/uploadTemp
targetdir = magData
sleeptime = 1800
```

[spectrum_settings]

```
obs = (observation#; starts out blank, then system-assigned)
band = 10.0 <data collection band; use 5.0, 10.0 or 15.0>
instrumentid = <instrument ID>
throttle = 100K <upload bandwidth limit in Kbits/sec>
spectrum_storage = /G1DRF <this is the standard disk name>
```

The obs starts out blank, then will be assigned according to the date-time the system is started. If you have trouble with this, you can blank it out and the system will recreate it.

Leave the band setting as is – to switch to a different band to monitor requires changing a jumper on the Grape1 board and changing the programming of the Leo Bodnar GPSDO.

Throttle: Set this to a lower setting if you find that uploads are interfering with your home internet response. If you are one of those lucky people with very fast broadband, you can set this to 0 (zero) which means unlimited.

Example working configuration file:

(Note: mag_settings are a bit of a work in progress.)

```
[profile]
token_value = Hy5Bftkiokliyu^(9675HF^&
grid = JO63fj
prefix = K2TWS
thestationid = N000704
central_host = pswsnetwork.caps.ua.edu

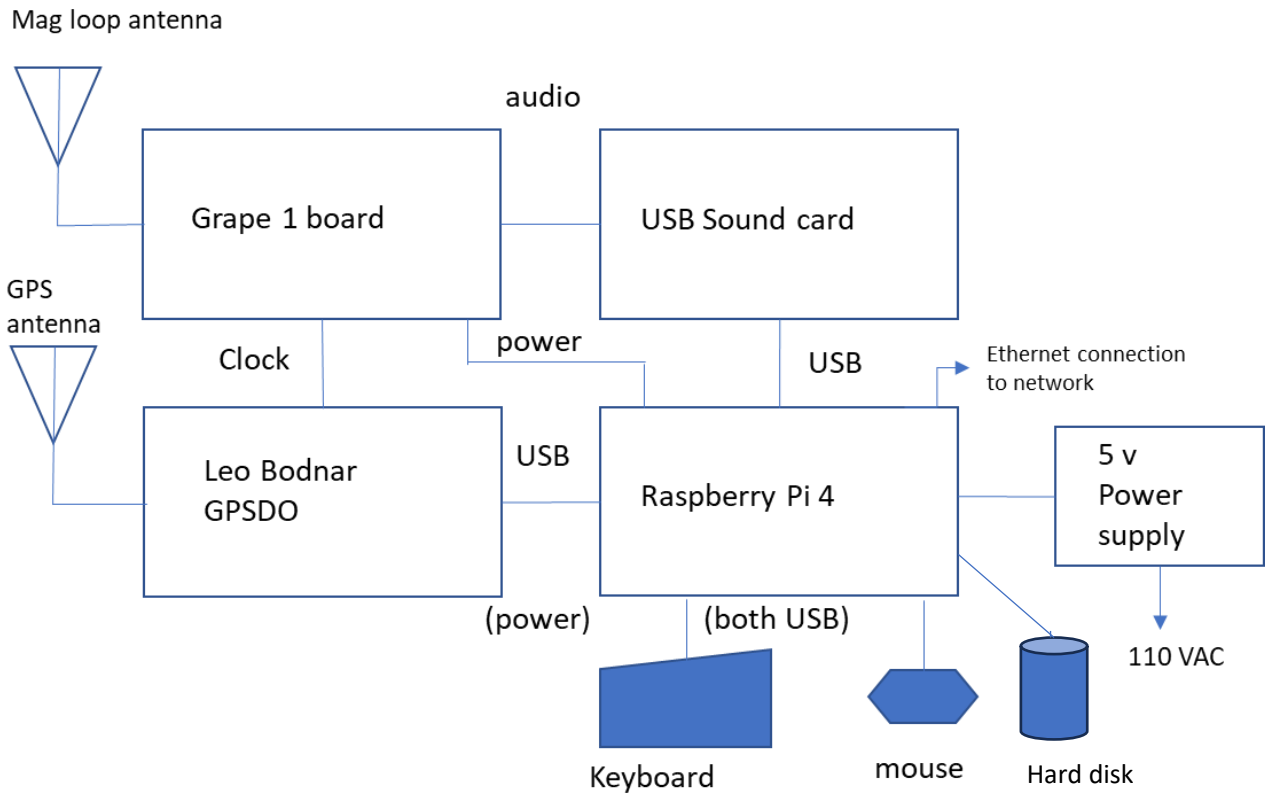
[mag_settings]
obs = OBS2023-03-22T21:52
instrumentname = Mag3
throttle = 100K
orig_path = /home/ab4ej/projects/rm3100-runMag/logs
temp_path = /home/ab4ej/projects/rm3100-runMag/xferData
upload_path = /media/ab4ej/seagate2/uploadTemp
targetdir = magData
sleeptime = 1800

[spectrum_settings]
obs =
band = 10.0
```

```
instrumentid = 2
throttle = 100K
spectrum_storage = /G1DRF
```

Hardware Architecture

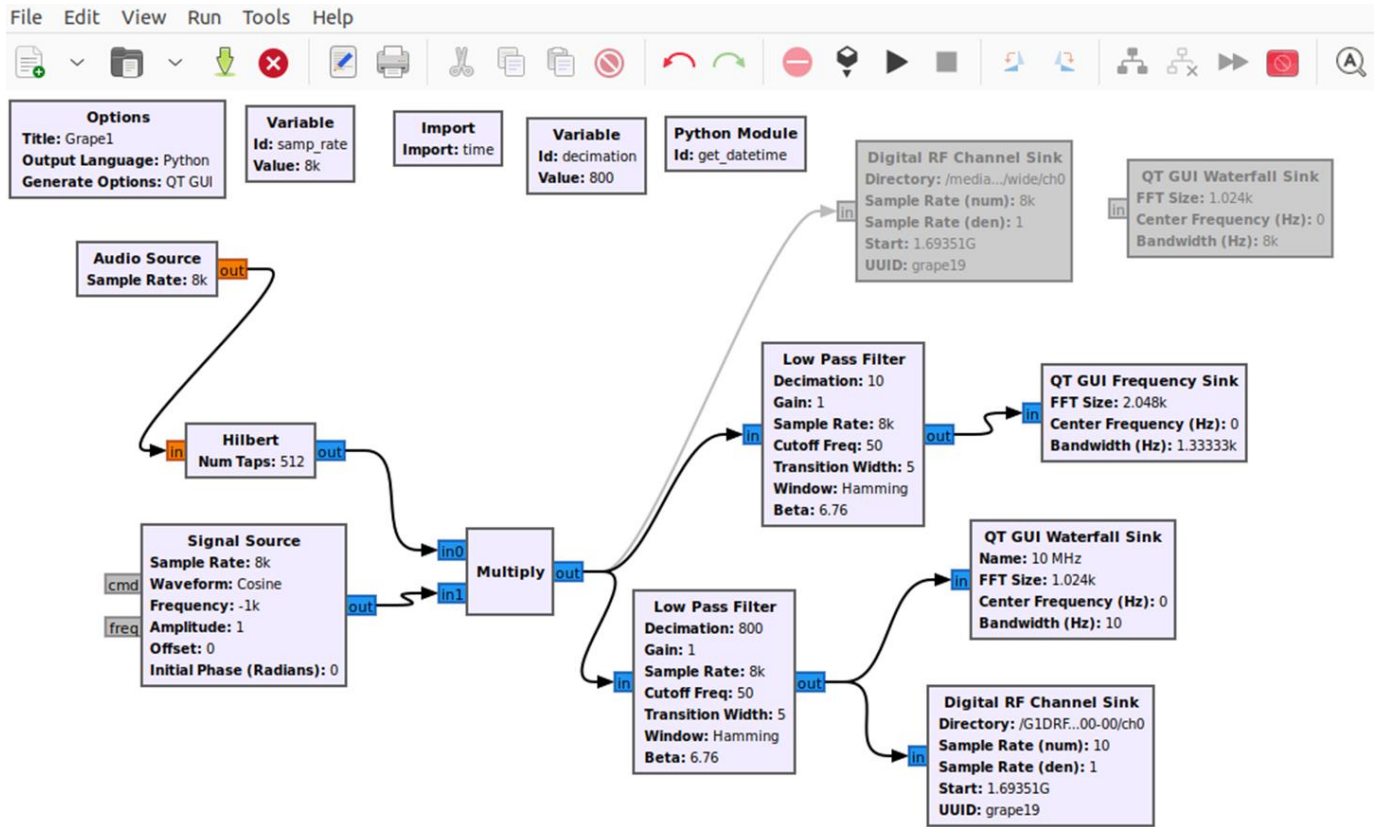
Here is a typical Grape1-DRF hardware layout. (Note that a number of different variations of this design have been successfully implemented.)



It is recommended to use a powered USB hub to connect everything to the Raspberry Pi.

How Gnuradio Is used

Gnuradio does the signal processing for the Grape1-DRF version. To look at the flowgraph, navigate to `/home/ubuntu/Documents/runGrape` and enter `gnuradio-companion`. You can then open the flowgraph `grape_DRF_spec.grc`. Don't change anything unless you *really* know what you are doing.



This flowgraph has the following blocks, starting at upper left:

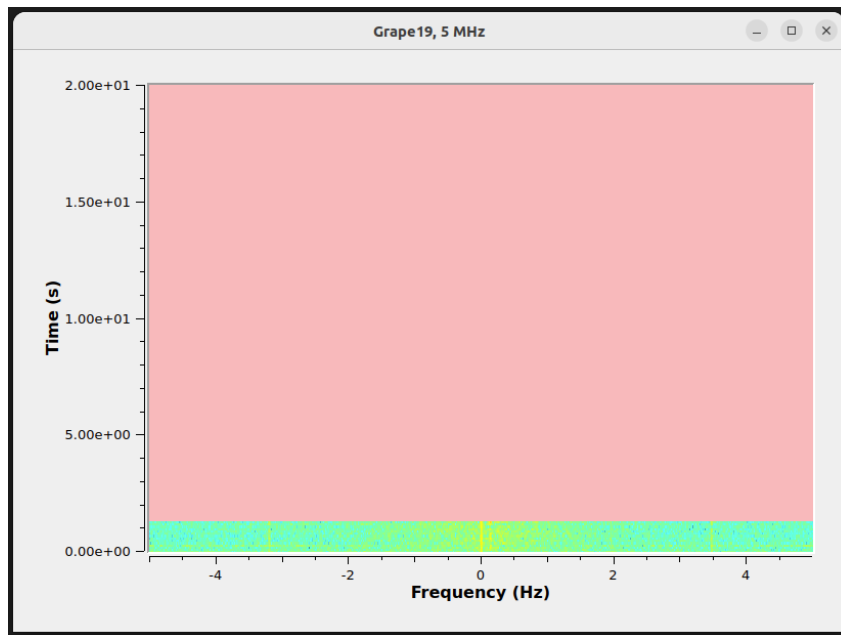
- Audio Source: connects to the USB sound card connected to the R-Pi and the Grape receiver board.
- Hilbert: a filter that shifts the phase angle of positive and negative frequency components so that a later FFT can split out frequencies above and below the center. (You don't have to understand what this is doing to use it).
- Signal Source and Multiply: this is a 8k sine wave use to mix with the 8 ksps signal to bring the frequency down to baseband
- Low Pass Filter: removes frequency components above the sample rate so that they don't show up as noise

- QT GUI waterfall sink displays the detected signal as a waterfall, where Doppler shift moves the detected frequency away from the zero point in the center. When the received signal matches the clock signal exactly (no Doppler shift), the display is centered (zero).
- Digital RF Channel Sink: writes the detected signal to disk as a DRF dataset. Values are saved as “I” and “Q” (real and imaginary components) using 32-bit floating point precision. This block saves the signal in a 10 Hz wide “narrow” band, which focuses on the parts of the signal right around the target frequency (2.5, 5, 10 MHz) for Doppler studies.
- Python Module (upper right) – reads the Adafruit Ultimate GPS board to get precise latitude, longitude and date-time (optional). If you are not using the Adafruit, you can delete this.
- There an added Frequency sink that displays a 8 kHz wide spectrum, which updates every 3 seconds or so. If you have a good signal, you can spot it as a pip in the middle of the spectrum. Note that this might not be easily visible in your selected band at some times of the day, depending on conditions.

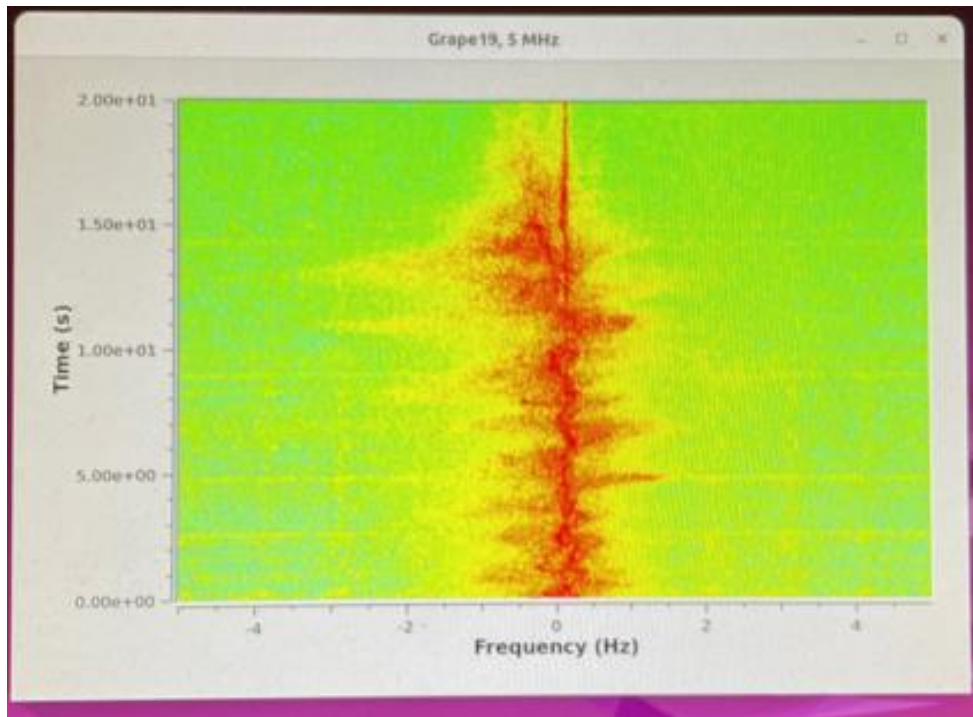
The flowgraph above also shows some blocks which are disabled (grayed out), which you can activate if you wish to save the full 8 kHz wide spectrum.

These functions should start automatically upon system boot-up.

You should see a waterfall like so (very slow; updates about once every 2 minutes)



This waterfall starts at the bottom and scrolls up. The yellow line you see in the center (above the 0 Frequency point) is the WWV signal; as the frequency varies away from zero due to Doppler, you will see this line move. The waterfall shows the bandwidth of 10 Hz, that is, +/- 5 Hz from the center point.



The system will automatically stop recording at 00:00Z (UTC) every day, start the next day's observation, and upload the previous day's data. You can see where a new observation has arrived in the server by looking at the Observations list; you can filter these various ways and see your observations.

Observations

Home
Stations
Observations
Users
Log In
Register your station
About

New observations show up at top of this list; you can use the filter to show only observations from your station.

Station Nickname: Start Date: End Date:

2.500 MHz
15.000 MHz

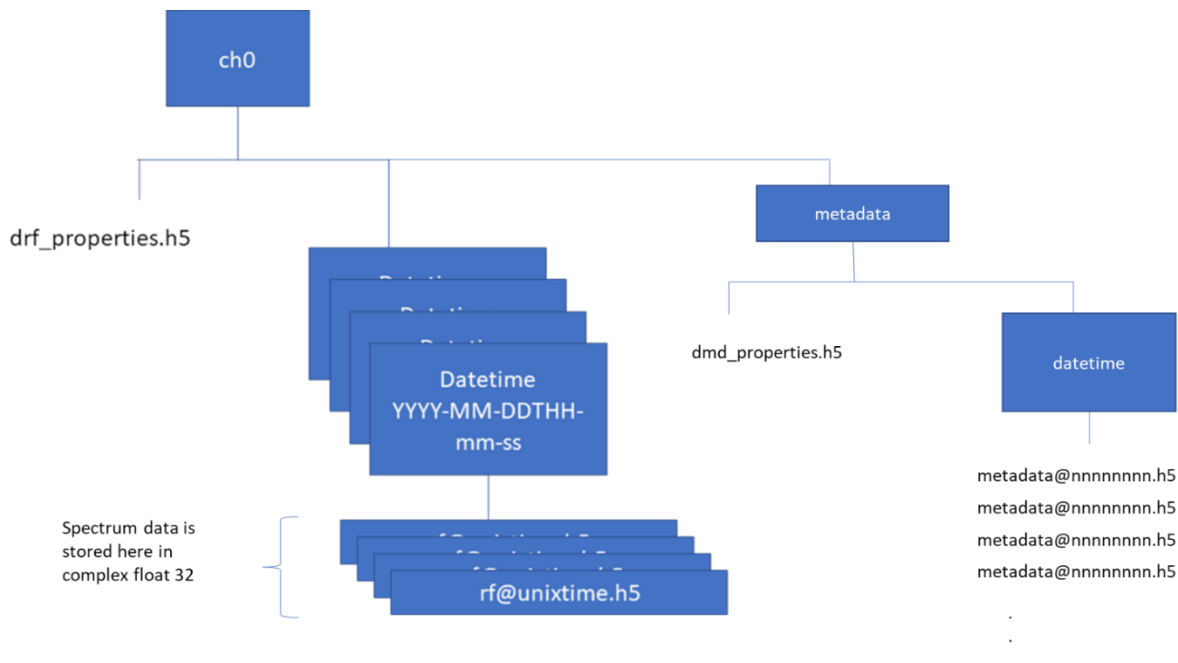
To download observation data, click on File/Observation link. Please be patient, it may take a while to zip a large observation.

Data File	Center Frequency	Station	Instrument	Size	File/Observation	Start (UTC)	End (UTC)
1	—	Mag_kv0s4	Mag_rev_E	0.7017	OBS2023-01-17T00:00.zip	2023-01-17 00:00:00	2023-01-17 15:50:00
1	—	N000003_A	Magnetometer2	0.2703	OBS2023-01-17T00:00.zip	2023-01-17 00:00:00	2023-01-17 15:49:00
10	5.000 MHz	Grape32	Grape32	2,615.8790	OBS2022-01-14T13:00	2022-08-04 19:11:00	2023-01-17 15:49:00
1	—	W2NAF	GMAG	1.0099	OBS2023-01-17T00:00.zip	2023-01-17 00:00:00	2023-01-17 15:46:00

An Introduction to Digital RF

Digital RF (DRF) is a data storage design that is optimized for saving radio spectrum data. Developed by MIT Haystack Observatory, it allows you to include metadata both about the entire package of data being saved, and also about individual measurements (if you need that level of detail) (for details, see https://github.com/MITHaystack/digital_rf) . Behind the scenes, it uses the HDF5 (“Hierarchical Data Format”) protocol (refer to <https://hdfgroup.org/>), a standard for saving scientific/mathematical data.

HDF (and DRF in particular) store data in a “dataset” format that is more like a database than a conventional file. This means that you can’t simply view/edit it using a text editor or Excel. Rather, you must use the necessary utilities that understand how to navigate through the binary file structure of the datasets. This minor inconvenience is compensated for the flexibility of the format, which allows you to store a variety of different data types, including lists, arrays, and images and associate metadata (descriptions of the data) with the stored information at every level. Software to navigate inside DRF files is available in several languages; for this project, we use Python. The following illustrates the logical architecture of a typical DRF dataset; the blue boxes are directories.



From upper left top, we have:

1. ch0 – by convention, the top level directory of a DRF dataset.
2. drf_properties.h5 – metadata pertaining to the entire dataset
3. Datetime directories – titled with datetime, these directories contain the detailed spectrum data files

4. Under the datetime directories, individual files contain the spectrum data. If this DRF dataset contains more than one channel (i.e., monitoring of multiple frequencies such as 5 MHz, 10 MHz, 15 MHz), these are stored as columns in these files. The data are complex (real and imaginary, each in a column), 32-bit floating point.
5. Metadata: under this directory is a properties file and one or more datetime directories, which contain a series of metadata files; these are used in cases where individual spectrum data packages or measurements have metadata associated with them at the detailed level. (Note that the gnradio DRF code stores metadata down at this level).

Needless to say, this sophisticated dataset format requires some special handling at every step. This is the reason for the relatively complex code for uploading your collected data to the PSWS Network Central Control System; Observations are stored in one-day packages (about 35 MB each for 10 Hz bandwidth monitoring).

If you wish to work with the data, sample code is available from sources such as:

- The Digital RF repository in github.com contains DRF sample code and documentation
- Sample code for analyzing and plotting spectrum data is available from the author
- For getting into the low level details, you can find documentation on the HDF internal format at the hdfgroup.org website

UNDER CONSTRUCTION: The PSWS software team is working on an enhancement to the web site to automatically make a waterfall plot of each uploaded spectrum observation.