

---

# **GBT Docs**

***Release 0.1***

**Green Bank Observatory**

**May 09, 2024**



# CONTENTS

<b>1</b>	<b>Tutorials</b>	<b>3</b>
1.1	Receiver-specific Tutorials	3
1.1.1	HI Position Switched (psw) Spectrum	3
1.1.1.1	1 Observation Preparation	4
1.1.1.2	2. Observations	6
1.1.1.3	3. Data Reduction	9
1.1.1.4	4. Data Reduction - scripted	27
<b>2</b>	<b>How-To Guides</b>	<b>31</b>
2.1	Infrastructure	31
2.1.1	How to connect remotely to the GBO network	31
2.1.1.1	FastX connection	31
2.1.1.2	VNC connection	37
2.1.1.3	What to do if the GBO network is down/slow	42
2.2	General Guides	48
2.2.1	GBT observations 101	49
2.2.1.1	Observing Preparation	49
2.2.1.2	Observing	51
2.3	Observing Mode Guides	52
2.3.1	How to derive a Mueller matrix	53
2.3.1.1	Observing Script	53
2.3.1.2	Data Reduction	56
2.4	Receiver specific Guides	56
2.4.1	How to observe using Argus	56
2.4.1.1	1. Turn on Argus	56
2.4.1.2	2. Primary Calibrator	57
2.4.1.3	3. Secondary Calibrator	58
2.4.1.4	4. Science Target	61
2.4.1.5	5. Subsequent observing procedure	61
2.4.2	MUSTANG-2 How-To Guides	61
2.4.2.1	How to Propose with MUSTANG-2	62
2.4.2.2	How to setup MUSTANG-2	63
2.4.2.3	How to observe with MUSTANG-2	71
2.4.2.4	MUSTANG-2 data	97
2.5	Software	98
2.5.1	How to process spectra using GBTIDL (examples)	98
2.5.1.1	Basic On/Off	98
2.5.1.2	Basic Frequency-Switched (fsw)	101
2.5.1.3	Basic Nodding	103
2.5.1.4	Advanced On/Off	105

<b>3</b>	<b>Explanation Material</b>	<b>115</b>
<b>4</b>	<b>Reference Guides</b>	<b>117</b>
4.1	Hardware . . . . .	117
4.1.1	Receivers . . . . .	117
4.2	Software . . . . .	117
4.2.1	Argus . . . . .	117
4.2.1.1	Introduction . . . . .	118
4.2.1.2	Configuration . . . . .	119
4.2.1.3	Observing . . . . .	119
4.2.1.4	Monitoring and Diagnostics . . . . .	122
4.2.1.5	IF Routing . . . . .	123
4.2.1.6	Troubleshooting Guide . . . . .	124
4.2.1.7	Data Reduction . . . . .	125
4.2.1.8	Documentation . . . . .	125
4.2.2	MUSTANG-2 . . . . .	126
4.2.2.1	MUSTANG-2 Overview . . . . .	126
4.2.2.2	MUSTANG-2 Instrument Team . . . . .	127
4.2.2.3	Technical Aspects . . . . .	128
4.2.2.4	Useful References . . . . .	132
4.2.3	Scheduling Block Commands . . . . .	134
	<b>Python Module Index</b>	<b>171</b>
	<b>Index</b>	<b>173</b>



The GBT User Documentation Hub is a prototype documentation system under active development. It is meant to consolidate all user documentation to use the GBT and process data obtained with the GBT.

---

### **Tutorials**

Learning-oriented lessons take you through a series of steps to complete a project.

Most useful when you want to get started with the GBT.

[Go to Tutorials](#)

### **How-To Guides**

Practical step-by-step guides to help you achieve a specific goal.

Most useful when you're trying to get something done.

[Go to How-To Guides](#)

### **Explanation**

Big-picture explanations of higher-level concepts.

Most useful for building understanding of a particular topic.

[Go to Explanation Material](#)

### **References**

Nitty-gritty technical descriptions of how the GBT works.

Most useful when you need detailed information about different GBT components or applications.

[Go to Reference Guides](#)

---

The hub follows the diataxis approach, where documentation is categorized with the user's need in mind. As a GBT user, you will be able to find Tutorials, How-to Guides, Explanations and Reference material that support your learning and the working aspects of your day-to-day tasks with either practical step-by-step instructions or knowledge-providing material.



## TUTORIALS

Learning-oriented lessons that take you through a series of steps to complete a project. Most useful when you want to get started with the GBT.

### 1.1 Receiver-specific Tutorials

HI Position Switched (psw) Data

- Setup scheduling blocks.
- Calibrate the data.

Keys: *L-Band*, *psw*, *gbtidl*

#### 1.1.1 HI Position Switched (psw) Spectrum

This tutorial will run you through the steps required to setup your scheduling blocks to execute an HI observation using position switching and how to calibrate the data.

---

##### What you should already know

In order to complete this tutorial, you need a GBO computing account. You should be relatively familiar with using the command line, and starting AstrID.

---

---

##### Data

This example is from the HI survey dataset <http://greenbankobservatory.org/~koneil/HIsurvey/index.shtml>

---

### 1.1.1.1 1 Observation Preparation

The instructions below will run you through the steps required to setup your scheduling blocks to execute an HI pointed observation (using position-switching) and how to calibrate the data.

## 1.1 Catalog

Before you start writing your scheduling block(s) it is helpful to prepare a source catalog in a separate file. This is especially advised if you have a long list of sources. For a short catalog it is also possible to add the sources directly in your scheduling block (not described here).

Here is an example of a RA/Dec coordinate system catalog with velocity, showing the header lines of the catalog and then 5 sources. The full catalog is available at `/home/astro-util/HISurvey/HI_survey.cat`.

```
# My source list with radial velocities
format=spherical
coordmode = J2000
HEAD= Name RA  Dec VEL
U8249      13:10:26.5    24:55:15.0    2541
U8503      13:30:44.9    32:45:38.0    4676
U8516      13:31:52.5    20:00:01.0    1020
U8896      13:58:38.5    07:12:58.0    4401
U8904      13:58:51.0    26:06:24.0    9773
```

Here Name is the name of the source (choose a name that uniquely identifies the source in your catalog), RA is the right ascension, Dec is the declination, and VEL is the source velocity in units of km/s.

We advise to save this catalog as a `.cat` file, in a known location. We will call it later from our scheduling block(s).

---

**Todo:** Add link to reference material for catalogs.

---

## 1.2 Configuration and Observing Scripts

At the GBT we use AstrID to prepare and execute scheduling blocks. AstrID is an application that you can use to edit and submit custom python-based scheduling blocks for GBT observations. AstrID is python-based and can incorporate custom user scripts. Here we show the full AstrID script, including reference to the HI catalog and the configurations. The script is available at `/home/astro-util/HISurvey/HI_survey.py`

```
# Reset configuration from prior observations.
ResetConfig()

# Import catalog of flux calibrators and user defined sources.
Catalog(fluxcal) #A catalog of standard flux calibrators known by astrid
Catalog('/home/astro-util/HISurvey/HI_survey.cat')

# Config for spectral line observations of HI using position switching.
psw_HI_config="""
receiver      = 'Rcvr1_2'      # L-Band receiver for HI
obstype       = 'Spectroscopy' # Spectral line observations
backend       = 'VEGAS'        # Spectral line backend
restfreq      = 1420.4058      # Rest frequency for HI (MHz)
```

(continues on next page)

(continued from previous page)

```

deltafreq      = 0.0          # Offsets for each spectral window (MHz)
bandwidth      = 23.44        # Defined by chosen VEGAS mode (MHz)
nchan          = 32768        # Number of channels in spectral window
vegas.subband  = 1           # Single/multiple spectral windows (1 or 8)
swmode         = 'tp'         # Switching mode: total power with noise diode
swtype         = None         # Type of switching, no switching
swper          = 1.0          # Length of full switching cycle (seconds)
swfreq         = 0, 0         # Frequency offset (MHz)
tint           = 6.0          # Integration time (s; int. mult. of swper)
vframe         = 'lsrk'       # Velocity reference frame
vdef           = 'Optical'     # Doppler-shifted velocity frame
noisecal       = 'lo'         # Level of the noise diode, 'lo' or 'hi'
pol            = 'Linear'      # 'Linear' or 'Circular' polarization
notchfilter    = 'In'         # 'In' to block 1200-1310 MHz RFI signal
""""

#First, lets go to a calibrator source, to check the telescope calibration
#If you are using a prime focus receiver, just use "AutoPeak
AutoPeakFocus('3C286')

# Break, to make sure the AutoPeakFocus looks good; If not, hit No and restart the_
↳script
Break('Everything ok?')

#Slew to your source of interest
Slew('U8503')

# If your flux calibrator was far from your source, then
# Perform position and focus correction on nearby calibrator
# Note that this step is often not necessary at low frequencies
AutoPeak()

# Break, to make sure the AutoPeakFocus looks good; If not, hit No and restart the_
↳script
Break('Everything ok?')

# Reconfigure after calibrator corrections.
Configure(psw_HI_config)

#Slew back to your source of interest
Slew('U8503')

# Balance the hardware.
Balance()

# Take an ON + OFF scan of the object
# The OFF scan follows the same sky area (Az, El) as the ON
OnOff('U8503', Offset('J2000', '00:05:00', 0.0, cosv=True), 300)

```

### 1.1.1.2 2. Observations

To learn how to execute your observing scripts, please follow the *GBT observations 101* guide.

## 2.1 AutoPeakFocus observations

Recall that after slewing near your source, the script looks for a nearby bright point source and runs a series of Peak (pointing) and Focus scans. When everything is working well, the scans look like this (switch to the DataDisplayTab in AstrID to see them).

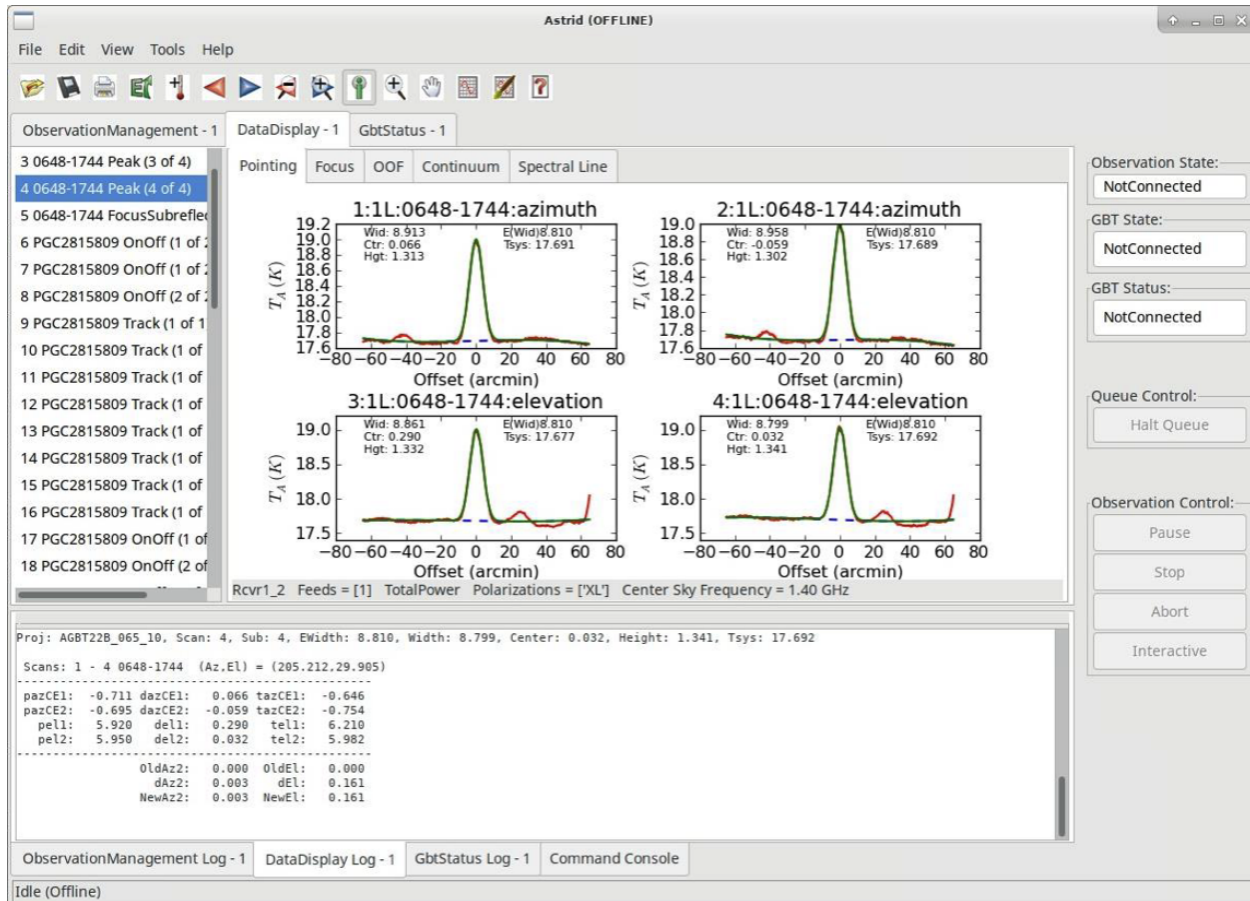


Fig. 1: L-Band peak observations. Four scans are taken, two scanning in Azimuth, two scanning in Elevation.

**Warning:** Sometimes, though, the focus curve fails to fit properly. This is typically due to the chosen source not having enough power. When this happens, you can look at the results and decide if they are close enough to simply have the operator put the focus offset into the system. This will also be true as long as the peak appears to be at  $-20 \pm 10$  mm.

A couple examples of this situation are shown below.

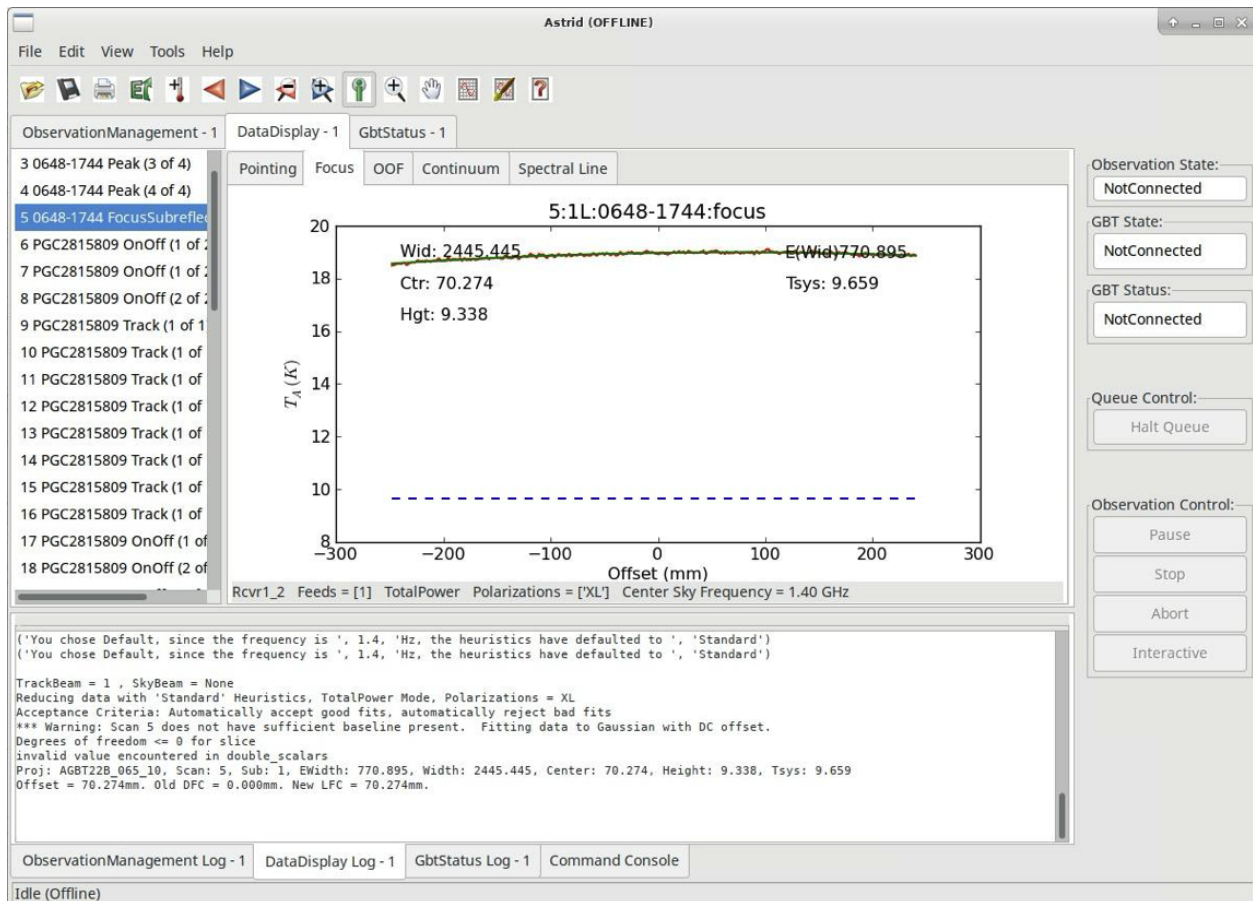
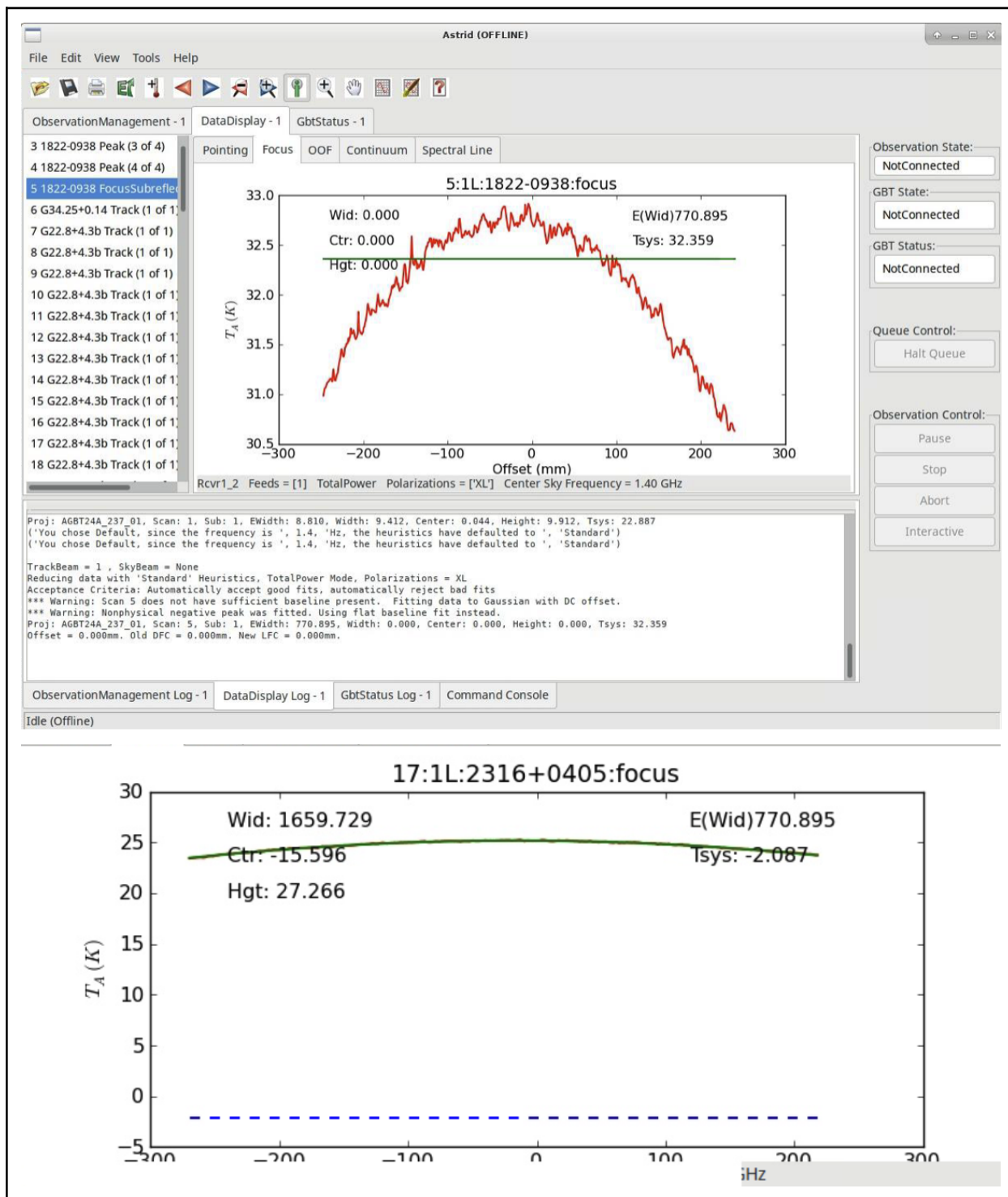


Fig. 2: L-Band focus observation.



**Danger:** When you're online (observing), you should never click on any of the scan numbers. This will cause AstrID to re-process the data and can change your pointing and focus values.



---

**Note:** You can always look at your observations after your observing session has ended. Start AstrID in offline mode and then under the DataDisplay Tab, open the ScanLog.fits file in the directory of your observations of interest.

---

## 2.2 Source observations

Once the AutoPeakFocus routines are complete, the telescope should configure for your spectral line observations, go to your source and balance the power, and then observe! You can check the status of your observations using the GBTStatus tab in AstrID.

---

**Note:** The DataDisplay - Spectral Line subtab is often not able to keep up with the incoming data, and so issues with not displaying data can be ignored.

---

Once your first set of data is written to disk (here when the first ON scan is complete, after 5 minutes), you should go into gbtidl and look at the data.

---

**Note:** You cannot look at the fully processed spectrum (ON+OFF) in gbtidl until the ON+OFF pair is complete (after about 10.5 minutes). You can, however, look to see that the data looks ok, using gbtidl and the `gettp, 5` command (where 5 is the scan number of interest).

---

More information on data processing in the [next section](#). Happy Observing!

### 1.1.1.3 3. Data Reduction

We will show two examples of how to process the obtained spectra:

---

#### Example 1

*Simple example*

Data taken with standard ONOFF command with cals (noise diodes) firing.

---

---

#### Example 2

*More complicated example*

Many observations were taken with the noise diodes turned off, followed by a short period where the noise diodes are firing. As a result you need to determine  $T_{sys}$  and then apply it to the on+off scans.

---

### 3.1 Simple Example

This simple example is from the HI survey dataset (<http://greenbankobservatory.org/~koneil/HIsurvey/index.shtml>). It also matches the observation technique in Observing, example 1.

#### 3.1.1 Start gbtidl

Open a terminal. At the prompt enter

```
gbtidl
```

You should see the following welcome screen:

```
IDL Version 7.1.1 (linux x86_64 m64). (c) 2009, ITT Visual Information Solutions
Installation number: 15269-3.
Licensed for use by: National Radio Astronomy Observatory
```

```
Starting GBTIDL
```

```
Display Device   : X
Visual Class     : TrueColor
Visual Depth     : 24-Bit
Color Table Size : 256
Number of Colors : 16777216
Decomposed Color : 0
```

```
-----
                        Welcome to GBTIDL v2.10.1
```

```
For news, documentation, enhancement requests, bug tracking,
discussion, and contributions, visit:
```

```
http://gbtidl.nrao.edu
```

```
For help with a GBTIDL routine from the command line, use
the procedure 'usage'. For example:
```

```
usage,'show'           ; gives the syntax of the procedure 'show'
usage,'show',/verbose  ; gives more information on 'show'
```

```
-----
GBTIDL ->
```

### 3.1.2 Load files of interest

```
GBTIDL -> dirin, '/home/astro-util/HIsurvey/Session02'
```

### 3.1.3 Look at file content

```
GBTIDL -> summary
```

Scan	Source	Vel	Proc	Seq	RestF	nIF	nInt	nFd	Az	El
220	3C286	0.0	OffOn	1	1.400	1	6	1	184.6	82.0
221	3C286	0.0	OffOn	2	1.400	1	6	1	186.6	82.0
222	3C286	0.0	OffOn	1	1.400	1	6	1	193.2	81.9
223	3C286	0.0	OffOn	2	1.400	1	6	1	195.1	81.8
224	3C286	0.0	OffOn	1	1.400	1	6	1	195.0	80.3
225	3C286	0.0	OffOn	2	1.400	1	5	1	199.5	81.6
226	3C286	0.0	OffOn	1	1.400	1	6	1	200.3	80.1
227	3C286	0.0	OffOn	2	1.400	1	6	1	205.4	81.3
228	B1328+254	0.0	OffOn	1	1.400	1	6	1	207.2	74.0
229	B1328+254	0.0	OffOn	2	1.400	1	6	1	210.6	75.2
230	B1345+125	0.0	OffOn	1	1.400	1	18	1	192.5	62.1
231	B1345+125	0.0	OffOn	2	1.400	1	18	1	194.9	63.4
244	B1345+125	0.0	OffOn	1	1.400	1	18	1	200.3	61.0
245	B1345+125	0.0	OffOn	2	1.400	1	18	1	202.8	62.2
246	B1345+125	0.0	OffOn	1	1.400	1	18	1	203.7	60.5
247	B1345+125	0.0	OffOn	2	1.400	1	18	1	206.3	61.6
248	B1345+125	0.0	OffOn	1	1.370	1	18	1	207.4	59.8
249	B1345+125	0.0	OffOn	2	1.370	1	18	1	210.1	60.8
250	B1345+125	0.0	Track	1	1.370	1	3	1	215.2	59.6
251	B1345+125	0.0	Track	1	1.370	1	3	1	215.9	59.4
263	U8091	213.0	OffOn	1	1.420	1	30	1	240.5	51.1
264	U8091	213.0	OffOn	2	1.420	1	30	1	240.3	51.2
265	U8091	213.0	Track	1	1.420	1	3	1	243.4	49.4
266	U8249	2541.0	OffOn	1	1.420	1	10	1	241.5	50.4
267	U8249	2541.0	OffOn	1	1.420	1	30	1	242.1	50.1
268	U8249	2541.0	OffOn	2	1.420	1	30	1	241.9	50.2
269	U8249	2541.0	Track	1	1.420	1	3	1	244.9	48.1
270	U8503	4676.0	OffOn	1	1.420	1	30	1	269.8	60.7
271	U8503	4676.0	OffOn	2	1.420	1	30	1	269.6	61.0
272	U8091	213.0	OffOn	1	1.420	1	30	1	253.1	41.9
273	U8091	213.0	OffOn	2	1.420	1	30	1	253.0	42.0
274	U8091	213.0	Track	1	1.420	1	3	1	255.5	39.6
275	U8249	2541.0	OffOn	1	1.420	1	30	1	266.1	47.3
276	U8249	2541.0	OffOn	2	1.420	1	30	1	266.0	47.5
277	U8249	2541.0	Track	1	1.420	1	3	1	267.9	45.2
278	U9965	4524.0	OffOn	1	1.420	1	30	1	220.3	68.0
279	U9965	4524.0	OffOn	2	1.420	1	30	1	219.9	68.1
280	U9965	4524.0	Track	1	1.420	1	3	1	222.8	67.4
281	U10351	891.0	OffOn	1	1.420	1	30	1	219.3	77.8
282	U10351	891.0	OffOn	2	1.420	1	30	1	218.4	77.9
283	U10351	891.0	Track	1	1.420	1	3	1	227.8	76.3

(continues on next page)

(continued from previous page)

284	U9007	4618.0	OffOn	1	1.420	1	30	1	248.1	38.8
285	U9007	4618.0	OffOn	2	1.420	1	30	1	248.0	38.9
286	U9007	4618.0	Track	1	1.420	1	3	1	250.5	36.7
287	U9007	5257.0	OffOn	1	1.420	1	30	1	233.6	48.4
288	U9007	5257.0	OffOn	2	1.420	1	30	1	233.5	48.5
289	U9803	5257.0	OffOn	1	1.420	1	30	1	264.5	58.5
290	U9803	5257.0	OffOn	2	1.420	1	30	1	264.3	58.7
291	U9803	5257.0	Track	1	1.420	1	3	1	266.5	56.3
292	U10351	891.0	OffOn	1	1.420	1	30	1	251.4	67.5
293	U10351	891.0	OffOn	2	1.420	1	30	1	251.1	67.8
294	U10351	891.0	Track	1	1.420	1	3	1	255.3	65.0
295	U10629	2980.0	OffOn	1	1.420	1	30	1	232.9	67.2
296	U10629	2980.0	OffOn	2	1.420	1	30	1	232.6	67.4
297	U10629	2980.0	OffOn	1	1.420	1	30	1	237.4	65.5
298	U10629	2980.0	OffOn	2	1.420	1	30	1	237.1	65.6
299	U10629	2980.0	Track	1	1.420	1	3	1	241.4	63.6
300	U11017	4644.0	OffOn	1	1.420	1	30	1	234.0	76.6
301	U11017	4644.0	OffOn	2	1.420	1	30	1	233.2	76.8
302	U11017	4644.0	OffOn	1	1.420	1	30	1	240.3	74.8
303	U11017	4644.0	OffOn	2	1.420	1	30	1	239.7	75.0
304	U11017	4644.0	Track	1	1.420	1	3	1	242.6	74.0
305	U11017	4644.0	Track	1	1.420	1	1	1	242.9	73.8
306	U11017	4644.0	Track	1	1.420	1	3	1	246.0	72.6
307	U11461	3122.0	OffOn	1	1.420	1	30	1	166.9	59.8
308	U11461	3122.0	OffOn	2	1.420	1	30	1	166.7	59.8
309	U11461	3122.0	OffOn	1	1.420	1	30	1	172.3	60.2
310	U11461	3122.0	OffOn	2	1.420	1	30	1	172.1	60.2
311	U11461	3122.0	Track	1	1.420	1	3	1	178.0	60.4
312	U11578	4601.0	OffOn	1	1.420	1	30	1	155.3	58.6
313	U11578	4601.0	OffOn	2	1.420	1	30	1	155.1	58.6
314	U11578	4601.0	Track	1	1.420	1	3	1	160.4	59.4
315	U11578	4601.0	Track	1	1.420	1	3	1	160.9	59.5
316	U11627	4864.0	OffOn	1	1.420	1	30	1	156.8	55.2
317	U11627	4864.0	OffOn	2	1.420	1	30	1	156.6	55.1
318	U11627	4864.0	OffOn	1	1.420	1	30	1	161.4	55.9
319	U11627	4864.0	OffOn	2	1.420	1	30	1	161.3	55.9
320	U11627	4864.0	Track	1	1.420	1	3	1	166.7	56.6
321	U11992	3592.0	Track	1	1.420	1	3	1	124.4	54.3
322	U11992	3592.0	OffOn	1	1.420	1	30	1	124.9	54.5
323	U11992	3592.0	OffOn	2	1.420	1	30	1	124.7	54.4

### 3.1.4 Empty the buffer memory

Make sure the buffer memory is empty

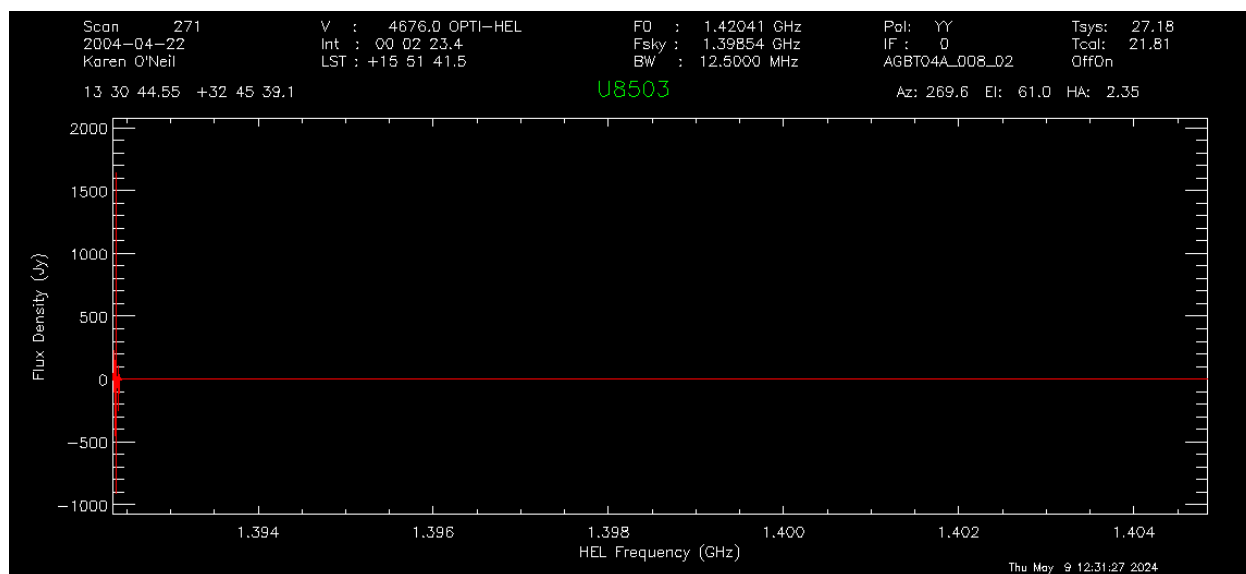
```
GBTIDL -> sclear
```

### 3.1.5 Process single on+off pair

#### 3.1.5.1 Process first polarization

Process the first polarization of the on+off pair (scan #270, 271), setting the intensity unit to Jansky.

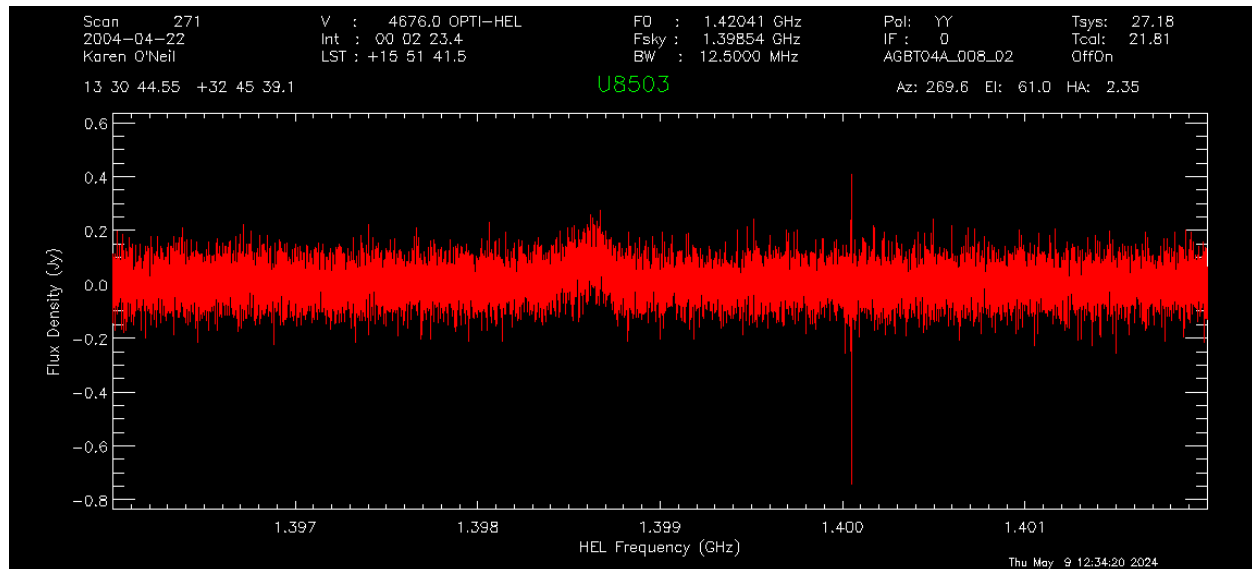
```
GBTIDL --> getps, 270, plnum=0, units='Jy'
```



#### 3.1.5.2 Adjust x-axis

Set the x-axis to an interesting frequency:

```
GBTIDL -> setx, 1.396, 1.402
```



### 3.1.5.3 Accumulate results

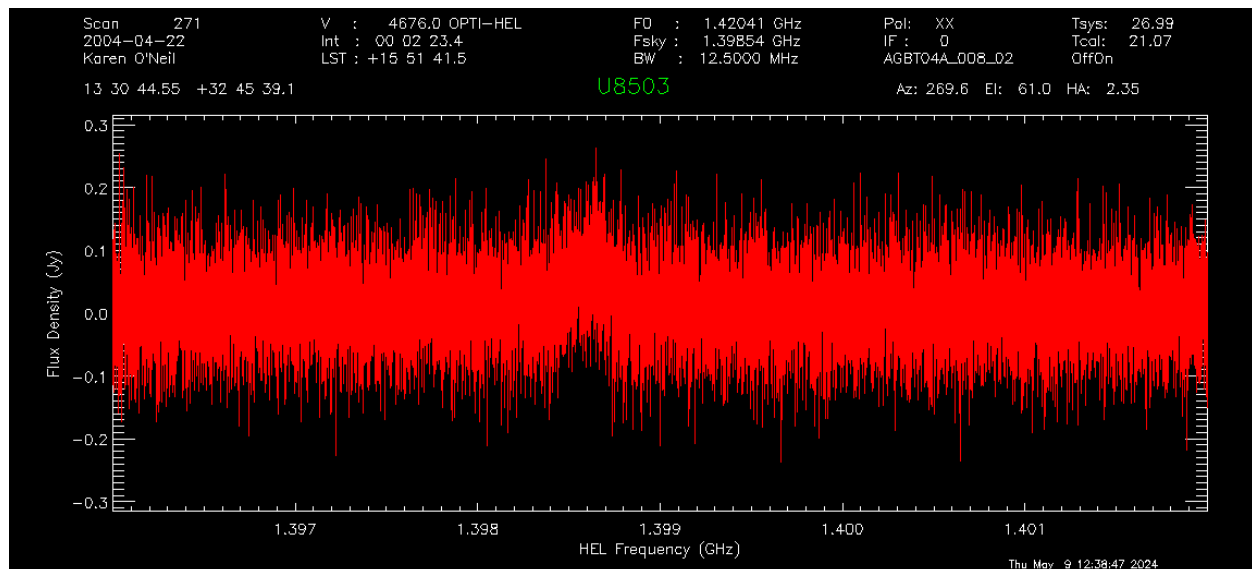
Accumulate the results from first polarization in the buffer.

```
GBTIDL -> accum
```

### 3.1.5.4 Process second polarization

Now process the second polarization of the same on+pf pair (scan #270, 271), again setting the intensity unit to Jansky.

```
GBTIDL -> getps, 270, plnum=1, units='Jy'
```



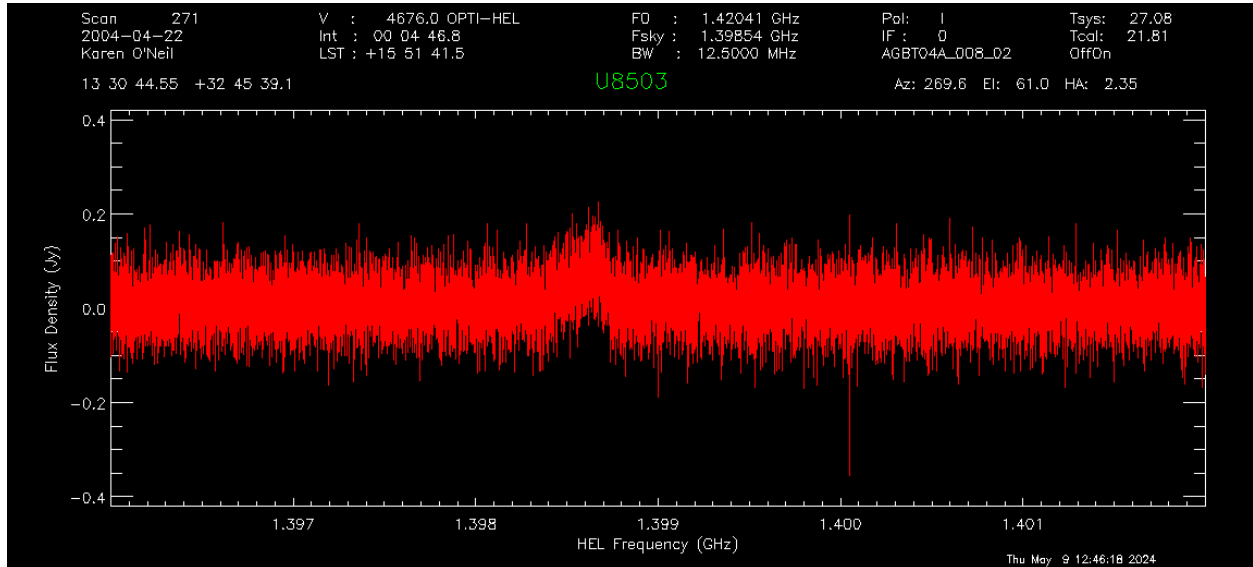
### 3.1.5.5 Accumulate results

Accumulate the results from second polarization in the buffer.

```
GBTIDL -> accum
```

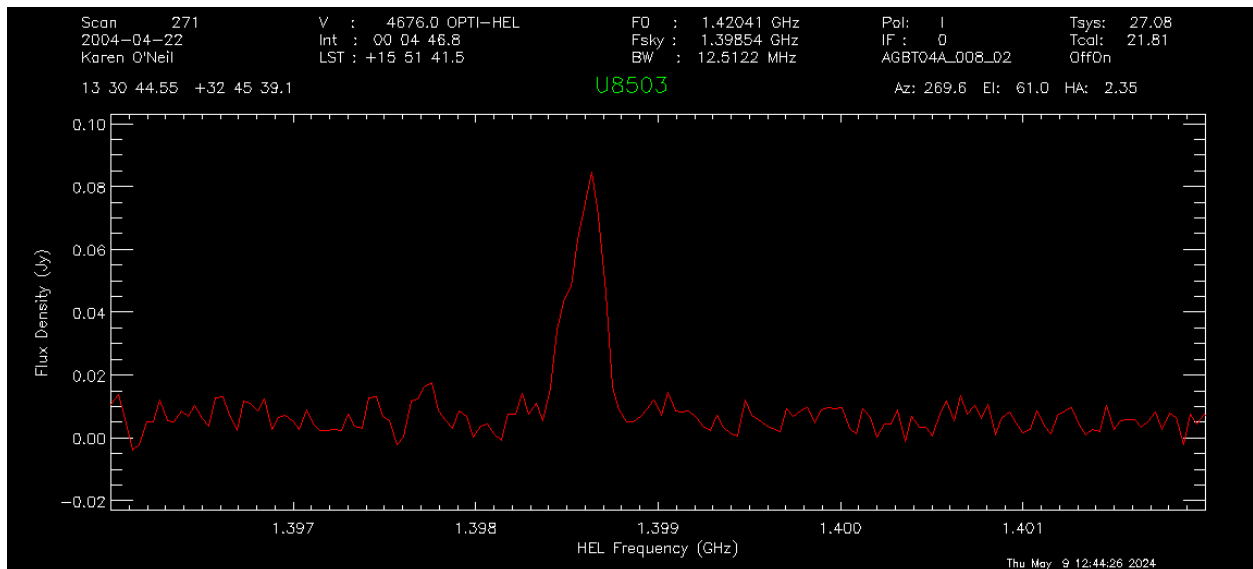
### 3.1.5.4 Average polarizations

```
GBTIDL -> ave
```



### 3.1.6 Smooth the result

```
GBTIDL -> gsmooth, 100, /decimate
```



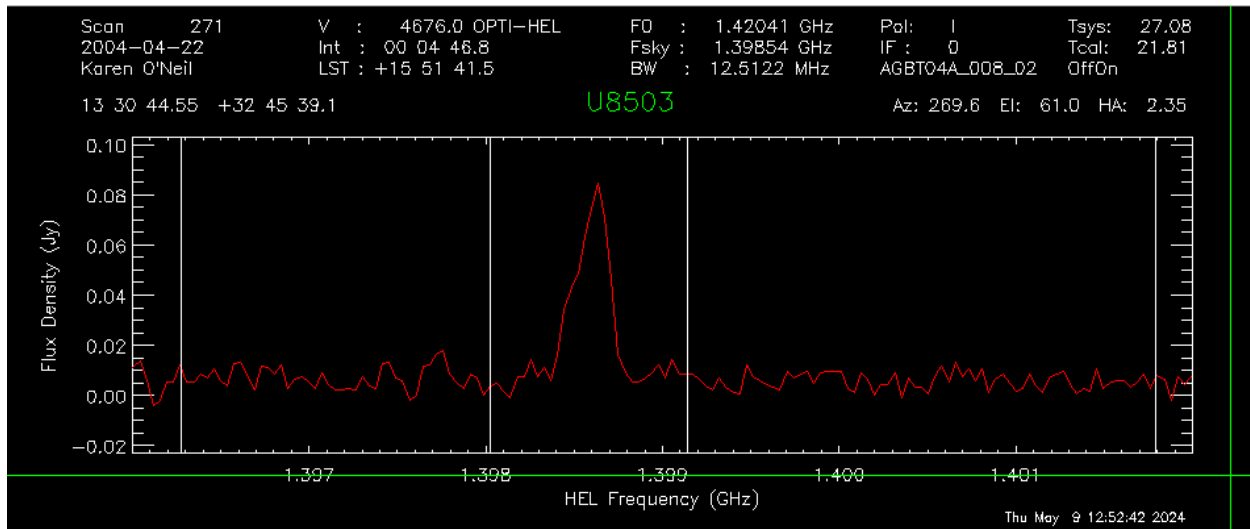
### 3.1.7 Fit the baseline

#### 3.1.7.1 Choose region

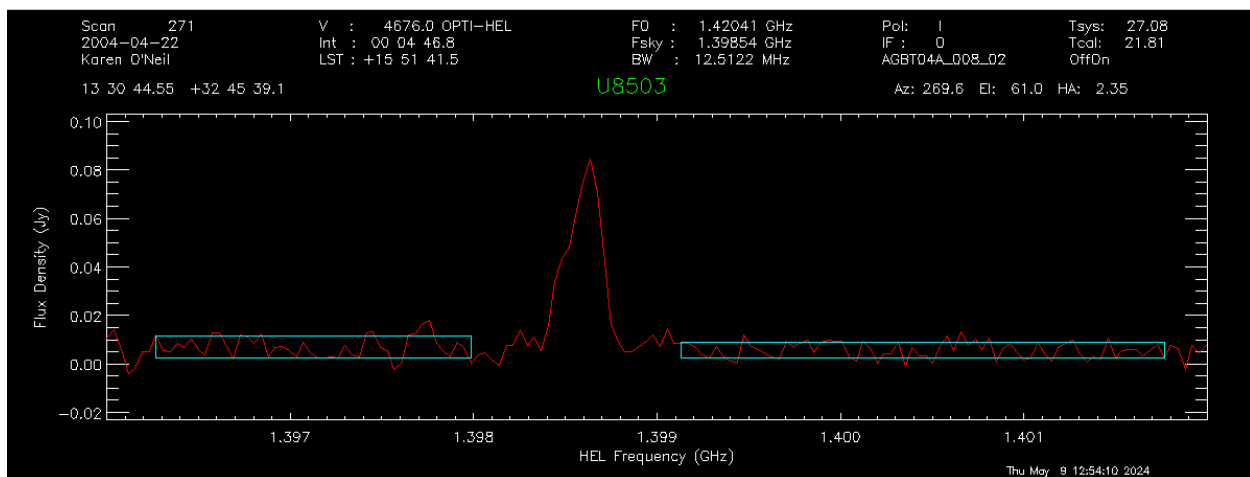
To choose the region for baseline fitting, mark the start and end of the baseline region on either side of the source interactively. Use the left button of your mouse to mark the region and the right button to exit.

```
GBTIDL -> setregion
```

This is what the plotter window will look like after marking the baseline regions (white vertical lines) while still in interactive mode (green crosshair cursor).



This is what the plotter window will look like after exiting the interactive mode. The selected regions are now marked with the cyan rectangular boxes.





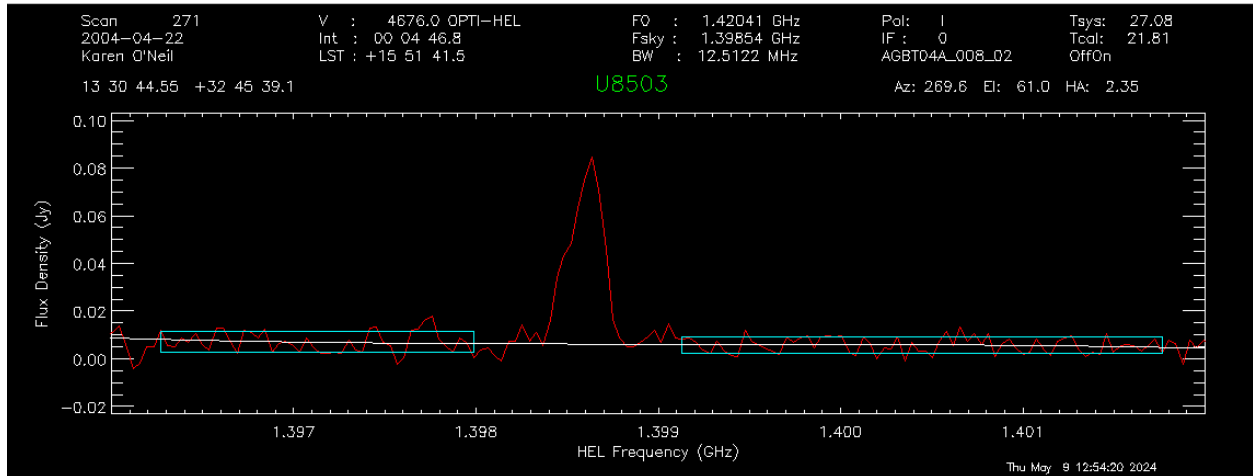
### 3.1.7.2 Set order of baseline fit

We will use a polynomial baseline fit of order 3.

```
GBTIDL -> nfit, 3
```

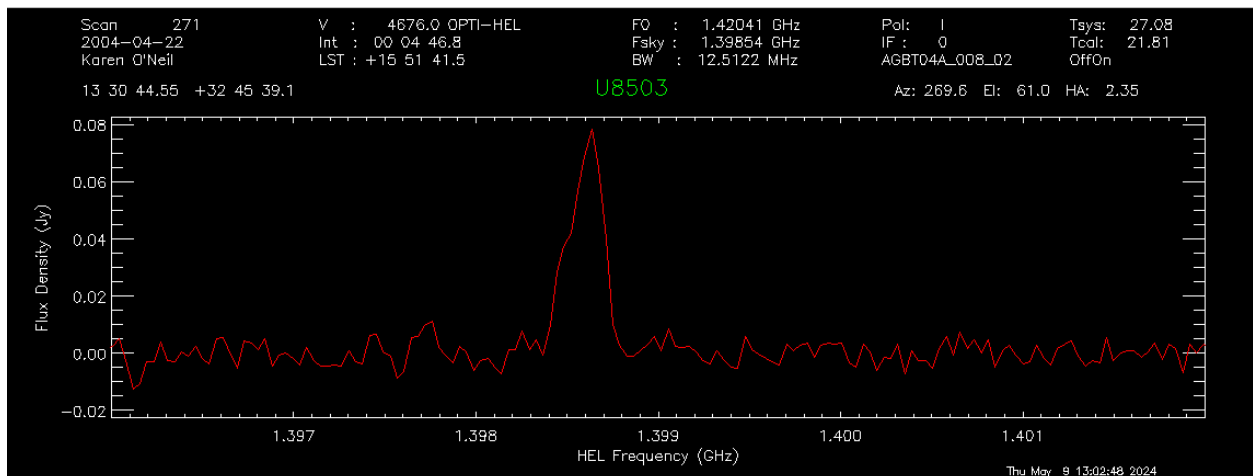
### 3.1.7.3 Show the fitted baseline

```
GBTIDL -> bshape
```



### 3.1.7.4 Subtract the fitted baseline

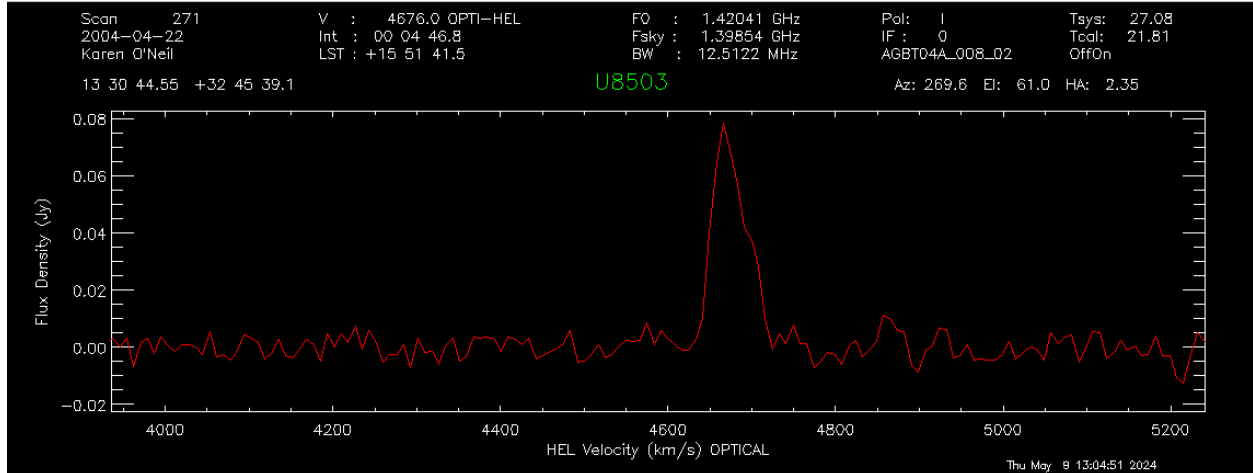
```
GBTIDL -> baseline
```



### 3.1.8 Switch the x-axis

Switch the x-axis from frequency to velocity

```
GBTIDL -> velo
```



### 3.1.9 Determine statistics

#### 3.1.9.1 RMS

Determine the statistics in the velocity ranges surrounding the galaxy, i.e. 4000 km/s - 4400 km/s and 4800 km/s - 5200 km/s.

```
GBTIDL -> stats, 4000, 4400
```

Chans	bchan	echan	Xmin	Xmax	Ymin	Ymax
49	197	245	4003.4	4400.8	-0.0069890	0.0075569
	Mean	Median	RMS	Variance	Area	
	1.9736E-06	0.00027831	0.0034322	1.1780E-05	0.00080165	

```
GBTIDL -> stats, 4800, 5200
```

Chans	bchan	echan	Xmin	Xmax	Ymin	Ymax
49	101	149	4799.2	5198.7	-0.0087689	0.011331
	Mean	Median	RMS	Variance	Area	
	-0.00011646	-0.00080158	0.0043761	1.9150E-05	-0.047555	

### 3.1.9.2 Line properties

Using the AWW (area, width, velocity) routine, determine the line's properties, including the velocity width at 20% of the line's peak velocity.

```
GBTIDL -> gmeasure, 1, 0.2, brange=4610, erange=4735, rms=0.0039
```

```
Area, Width, Velocity (followed by errors in same order) =    3.67    83.31 4677.81
->0.22    0.00    0.00
```

### 3.1.10 Compare with online database

Check your results against the online database available here: <http://greenbankobservatory.org/~koneil/HIsurvey/index.shtml>.

## 3.2 More Complicated Example

This example is from the HI survey dataset (<http://greenbankobservatory.org/~koneil/HIsurvey/index.shtml>).

It also matches the observation technique in Observing, example 1. Note, though, that not all the data in this dataset was taken using the standard ONOFF command with the cals (noise diodes) firing. Instead many of the observations were taken with the noise diodes turned off, followed by a short scan where the noise diodes are firing. As a result you need to determine the Tsys and then apply it to the on+off scans.

### 3.2.1 Start gbtidl

Open a terminal. At the prompt enter

```
gbtidl
```

You should see the following welcome screen:

```
IDL Version 7.1.1 (linux x86_64 m64). (c) 2009, ITT Visual Information Solutions
Installation number: 15269-3.
Licensed for use by: National Radio Astronomy Observatory
```

```
Starting GBTIDL
```

```
Display Device   : X
Visual Class     : TrueColor
Visual Depth     : 24-Bit
Color Table Size : 256
Number of Colors : 16777216
Decomposed Color : 0
```

```
-----
                        Welcome to GBTIDL v2.10.1
```

```
For news, documentation, enhancement requests, bug tracking,
discussion, and contributions, visit:
```

(continues on next page)

(continued from previous page)

<http://gbtidl.nrao.edu>

For help with a GBTIDL routine from the command line, use  
the procedure 'usage'. For example:

```
usage, 'show'           ; gives the syntax of the procedure 'show'
usage, 'show', /verbose ; gives more information on 'show'
```

-----

GBTIDL ->

### 3.2.2 Load files of interest

```
GBTIDL -> dirin, '/home/astro-util/HIsurvey/Session02'
```

### 3.2.3 Look at file content

```
GBTIDL -> summary
```

Here we're showing only the scans of interest for this example. We have two OffOn pairs (scans pairs 295+296, 297+298) and one track scan (scan 299).

Scan	Source	Vel	Proc	Seq	RestF	nIF	nInt	nFd	Az	El
295	U10629	2980.0	OffOn	1	1.420	1	30	1	232.9	67.2
296	U10629	2980.0	OffOn	2	1.420	1	30	1	232.6	67.4
297	U10629	2980.0	OffOn	1	1.420	1	30	1	237.4	65.5
298	U10629	2980.0	OffOn	2	1.420	1	30	1	237.1	65.6
299	U10629	2980.0	Track	1	1.420	1	3	1	241.4	63.6

### 3.2.4 Double-check the scans

```
GBTIDL -> list, scan=295
```

Here you can see that there is no noise diode (cal) firing during the on + off scans. We're only showing the first few lines of the output, when you run the command, the output list will be a bit longer.

#INDEX	SOURCE	SCAN	PROCEDURE	POL	IFNUM	FDNUM	INT	SIG	CAL
2472	U10629	295	OffOn	XX	0	0	0	T	F
2473	U10629	295	OffOn	YY	0	0	0	T	F
2474	U10629	295	OffOn	XX	0	0	1	T	F
2475	U10629	295	OffOn	YY	0	0	1	T	F
2476	U10629	295	OffOn	XX	0	0	2	T	F
2477	U10629	295	OffOn	YY	0	0	2	T	F
2478	U10629	295	OffOn	XX	0	0	3	T	F
2479	U10629	295	OffOn	YY	0	0	3	T	F
2480	U10629	295	OffOn	XX	0	0	4	T	F
2481	U10629	295	OffOn	YY	0	0	4	T	F

```
GBTIDL -> list, scan=296
```

#INDEX	SOURCE	SCAN	PROCEDURE	POL	IFNUM	FDNUM	INT	SIG	CAL
2532	U10629	296	OffOn	XX	0	0	0	T	F
2533	U10629	296	OffOn	YY	0	0	0	T	F
2534	U10629	296	OffOn	XX	0	0	1	T	F
2535	U10629	296	OffOn	YY	0	0	1	T	F
2536	U10629	296	OffOn	XX	0	0	2	T	F
2537	U10629	296	OffOn	YY	0	0	2	T	F
2538	U10629	296	OffOn	XX	0	0	3	T	F
2539	U10629	296	OffOn	YY	0	0	3	T	F
2540	U10629	296	OffOn	XX	0	0	4	T	F
2541	U10629	296	OffOn	YY	0	0	4	T	F

```
GBTIDL -> list, scan=299
```

#INDEX	SOURCE	SCAN	PROCEDURE	POL	IFNUM	FDNUM	INT	SIG	CAL
2712	U10629	299	Track	XX	0	0	0	T	T
2713	U10629	299	Track	XX	0	0	0	T	F
2714	U10629	299	Track	YY	0	0	0	T	T
2715	U10629	299	Track	YY	0	0	0	T	F
2716	U10629	299	Track	XX	0	0	1	T	T
2717	U10629	299	Track	XX	0	0	1	T	F
2718	U10629	299	Track	YY	0	0	1	T	T
2719	U10629	299	Track	YY	0	0	1	T	F
2720	U10629	299	Track	XX	0	0	2	T	T
2721	U10629	299	Track	XX	0	0	2	T	F
2722	U10629	299	Track	YY	0	0	2	T	T
2723	U10629	299	Track	YY	0	0	2	T	F

### 3.2.5 Determine $T_{sys}$

We will use the track scan (scan 299) to determine the system temperature,  $T_{sys}$  for each polarization.

```
GBTIDL -> gettp, 299, plnum=0
```

```
Scan: 299 (IF:0 FD:0 PL:0)   Tsys: 27.98
```

```
GBTIDL -> gettp, 299, plnum=1
```

```
Scan: 299 (IF:0 FD:0 PL:1)   Tsys: 27.40
```

### 3.2.6 Empty the buffer memory

Make sure the buffer memory is empty

```
GBTIDL -> sclear
```

### 3.2.7 Process on+off pairs

Here you have to use the getsigref command, and explicitly give the command the on and off scan numbers.

```
GBTIDL -> getsigref, 296, 295, plnum=0, tsys=27.98, unit='Jy'
```

```
SigScan: 296 RefScan: 295 (IF:0 FD:0 PL:0) units: Jy tau:0.008 ap_eff:0.710  ̣  
↪Tsys: 27.98
```

```
GBTIDL -> accum
```

```
GBTIDL -> getsigref, 296, 295, plnum=1, tsys=27.40, unit='Jy'
```

Blanked spectra: ignored 2 integrations

```
SigScan: 296 RefScan: 295 (IF:0 FD:0 PL:1) units: Jy tau:0.008 ap_eff:0.710  ̣  
↪Tsys: 27.40
```

```
GBTIDL -> accum
```

```
GBTIDL -> getsigref, 298, 297, plnum=0, tsys=27.98, unit='Jy'
```

```
SigScan: 298 RefScan: 297 (IF:0 FD:0 PL:0) units: Jy tau:0.008 ap_eff:0.710  ̣  
↪Tsys: 27.98
```

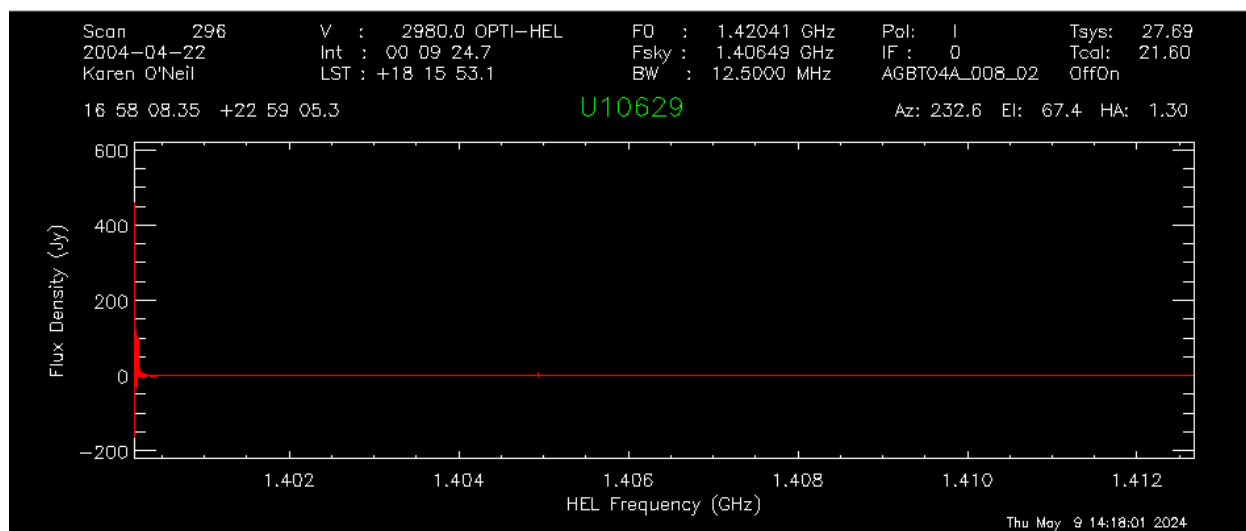
```
GBTIDL -> accum
```

```
GBTIDL -> getsigref, 298, 297, plnum=0, tsys=27.40, unit='Jy'
```

```
SigScan: 298 RefScan: 297 (IF:0 FD:0 PL:1) units: Jy tau:0.008 ap_eff:0.710  ̣  
↪Tsys: 27.40
```

```
GBTIDL -> accum
```

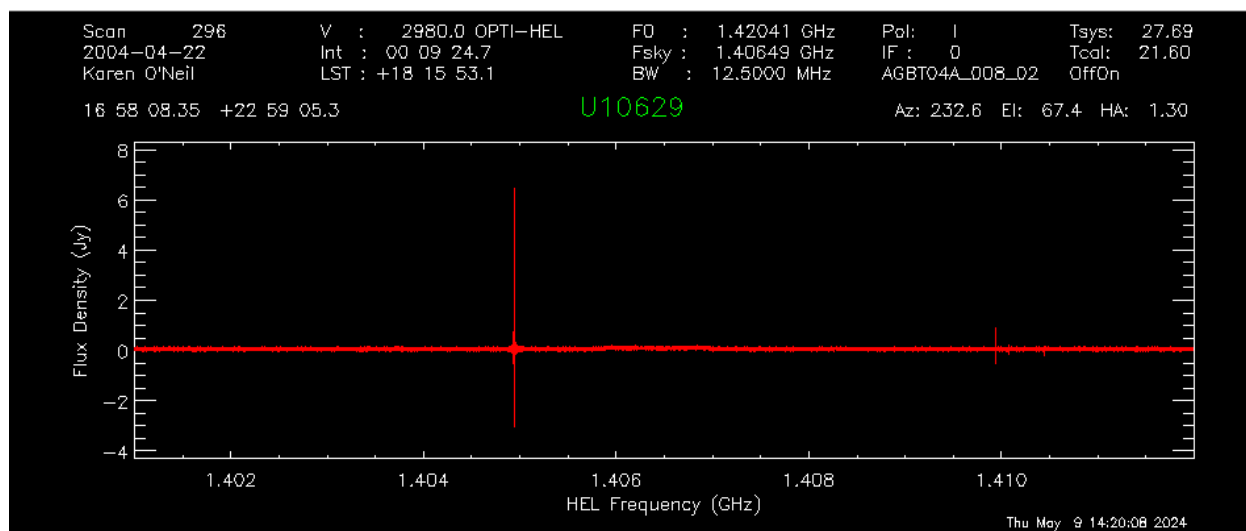
```
GBTIDL -> ave
```



### 3.2.8 Adjust x-axis

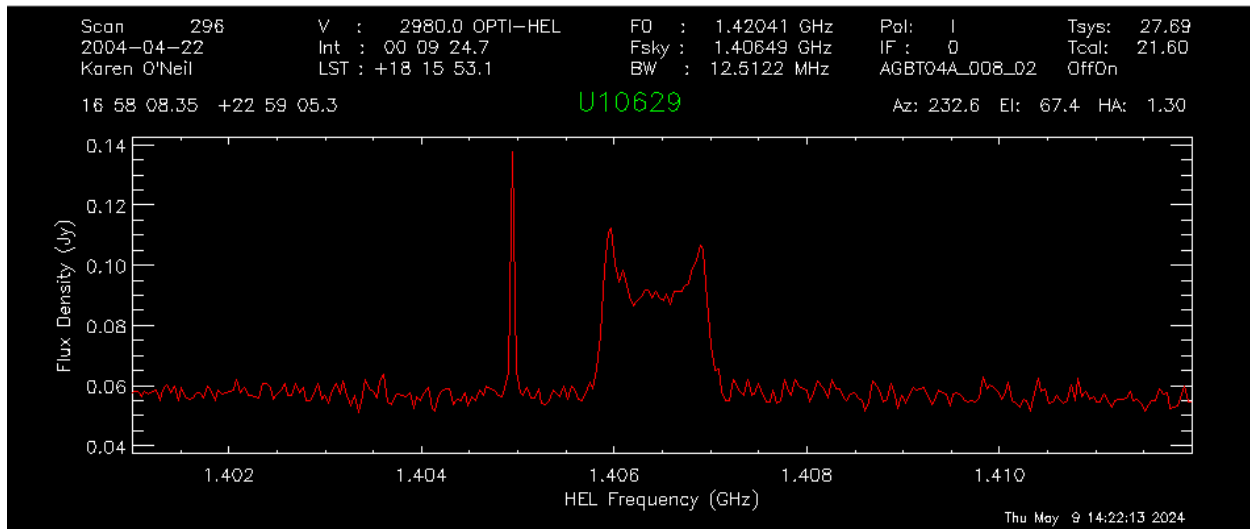
Set the x-axis to avoid the edges.

```
GBTIDL -> setx, 1.401, 1.412
```



### 3.2.9 Smooth the spectrum

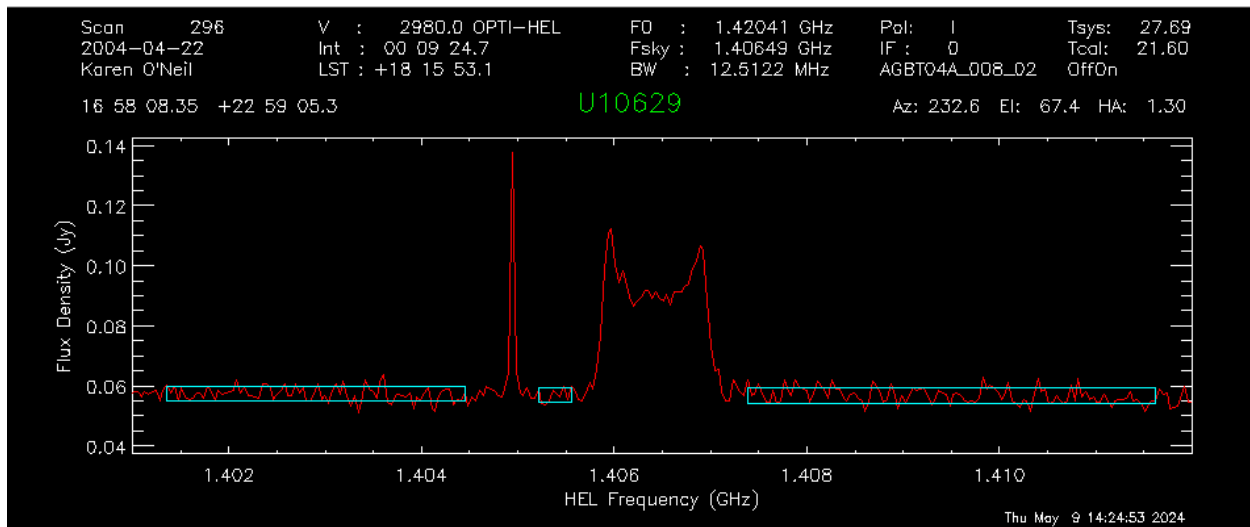
```
GBTIDL -> gsmooth, 100, /decimate
```



### 3.2.10 Remove baseline

Set regions avoiding the RFI spike.

```
GBTIDL -> setregion
```



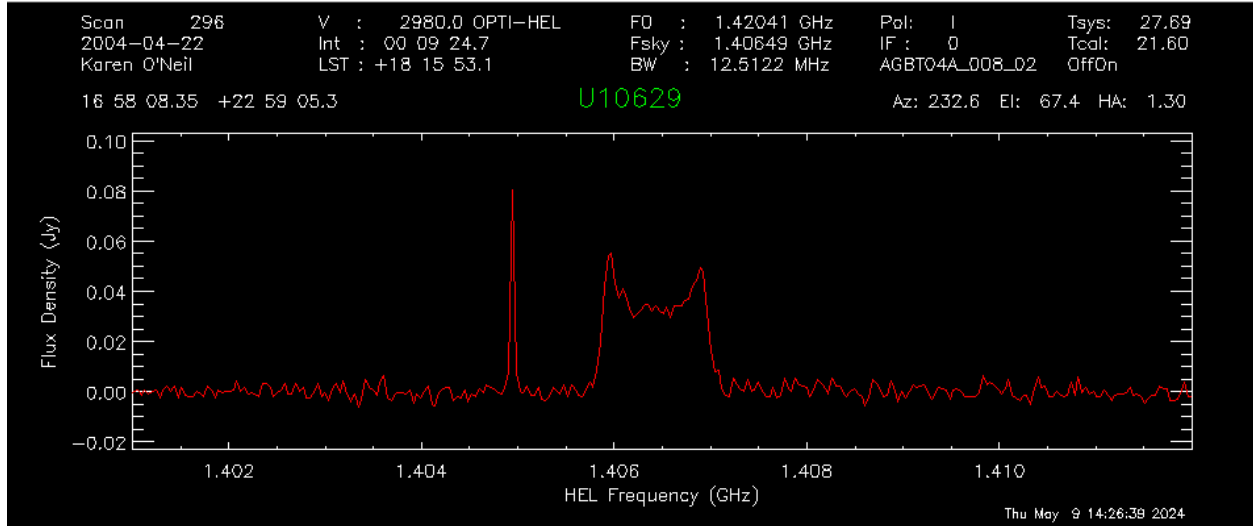
Remove baseline (after inspection) using first order polynomial (i.e. a line).

```
GBTIDL -> nfit, 1  

GBTIDL -> bshape  

GBTIDL -> baseline
```





### 3.2.11 Determine statistics

#### 3.2.11.1 RMS noise

Determine the statistics in the velocity ranges surrounding the galaxy, i.e. 2000 km/s - 2500 km/s and 3500 km/s - 4000 km/s.

```
GBTIDL -> stats, 2000, 2500
```

Chans	bchan	echan	Xmin	Xmax	Ymin	Ymax
63	222	284	1996.4	2503.1	-0.0050648	0.0063588
	Mean	Median	RMS	Variance	Area	
	0.00022812	0.00011783	0.0025809	6.6610E-06	0.11764	

```
GBTIDL -> stats, 3500, 4000
```

Chans	bchan	echan	Xmin	Xmax	Ymin	Ymax
62	40	101	3496.8	4000.3	-0.0062628	0.0065100
	Mean	Median	RMS	Variance	Area	
	-3.3634E-05	-0.00021309	0.0024638	6.0704E-06	-0.017239	

#### 3.2.11.2 Line properties

Using the AWW (area, width, velocity) routine, we can determine the line's properties, including the velocity width at 20% of the line's peak velocity. Here we will use a few methods:

##### 1. Fraction of mean

As a fraction of the mean within the region of interest. The mean of data from brange through erange is calculated. The edges are then those locations where the data values are greater than fract\*mean for 3 consecutive channels starting from the end points of the region of interest and searching towards the center (method #1), first using 50% of the peak intensity and then 20%

```
GBTIDL -> gmeasure, 1, 0.5, brange=2815, erange=3142, rms=0.00252
```

```
Area, Width, Velocity (followed by errors in same order) =    9.45  252.76 2979.
↪57    0.27    0.00    0.00
```

```
GBTIDL -> gmeasure, 1, 0.2, brange=2815, erange=3142, rms=0.00252
```

```
Area, Width, Velocity (followed by errors in same order) =    9.45  277.50 2974.
↪64    0.26    0.00    0.00
```

## 2. Fraction of maximum value

As a fraction of the maximum value within the region of interest. The peak of data from brange to erange is found. The edges are then those locations where the data values are greater than  $\text{fract} * (\text{peak} - \text{rms})$  for 3 consecutive channels starting from the end points of the region of interest and searching towards the center (method #2), first using 50% of the peak intensity and then 20%.

```
GBTIDL -> gmeasure, 2, 0.5, brange=2815, erange=3142, rms=0.00252
```

```
Area, Width, Velocity (followed by errors in same order) =    9.45  238.38 2980.
↪15    0.26    0.00    0.00
```

```
GBTIDL -> gmeasure, 2, 0.2, brange=2815, erange=3142, rms=0.00252
```

```
Area, Width, Velocity (followed by errors in same order) =    9.45  258.82 2979.
↪04    0.25    0.00    0.00
```

## 3. Fraction of each of two peaks – identified by user

You use the cursor to mark two peaks in the region of interest or those peaks are identified through the lefthorn and righthorn parameters. The maximum value within 10 channels of each user-supplied peak location is found. The left edge is where the data value falls below  $\text{fract} * (\text{peak} - \text{rms})$  for 3 consecutive channels searched from the location of the peak. The right-channel peak is similarly used to find the right edge (method #3), first using 50% of the peak intensity and then 20%.

```
GBTIDL -> gmeasure, 3, 0.5, brange=2815, erange=3142, rms=0.00252
```

click on the positions for the two peaks

```
Area, Width, Velocity (followed by errors in same order) =    9.49  239.90↪
↪2979.39    0.26    0.00    0.00
```

```
GBTIDL -> gmeasure, 3, 0.2, brange=2815, erange=3142, rms=0.00252
```

click on the positions for the two peaks

```
Area, Width, Velocity (followed by errors in same order) =    9.49  259.96↪
↪2978.47    0.25    0.00    0.00
```

#### 1.1.1.4 4. Data Reduction - scripted

This example is from the HI survey dataset (<http://greenbankobservatory.org/~koneil/HIsurvey/index.shtml>).

```
; Step 3.2.2
dirin, '/home/astro-util/HIsurvey/Session02'

jnk='jnk'    ; this is used later, to pause the observations
freeze      ; keep from plotting on the screen, to speed up the processing

; skipping steps 3.2.3, and 3.2.4 since you are running a script you should already
↳ understand your dataset.

; Steps 3.2.5 here we are running the commands quietly, and then
; getting the tsys values from the !g structure.

gettp,299,plnum=0, /quiet
tsys0=!g.s.tsys

gettp,299,plnum=1, /quiet
tsys1=!g.s.tsys

;Step 3.2.6 - clear the buffer
sclear

;Step 3.2.7; Here wer are going to loop over the scans of interest.
;Because this is a text file, we are using the $& at the end, to continue the lines
;appropriately. Again this is being run quietly

for i=295,297,2 do begin &$    ; scan numbers
  for p=0,1 do begin &$      ; polarizzation numbers
    if (p eq 0) then tsys=tsys0[0] else tsys=tsys1[0] &$
    getsigref,i+1,i,plnum=p,tsys=tsys,unit='Jy', /quiet &$
    accum &$
  endfor &$
endfor

ave

; Step 3.2.8 - fix the x-axis
setxunit,'GHz' ; make sure we have the expected units in the x-axis.
setx,1.401,1.412

; Step 3.2.9 - smooth the data to the desired resolution
gsmooth,100,/decimate

show
unfreeze

; Step 3.2.10 - Remove baseline
; set region - Here we are going to avoid clicking on the screen and set the regions for
↳ the
; baseline fitting
region=[1.402, 1.4045, 1.40506, 1.4054, 1.4072, 1.4115]
```

(continues on next page)

(continued from previous page)

```

reg1=xtochan(region)
Nregion,reg1
nfit,1
bshape

; this pauses the observations, to see the baseline shape
; this is only needed if this script is being read into gbtidl as a whole.
read,jnk,prompt='Push return to continue: '

;remove the baseline
baseline ; removes the baseline

; Step 3.2.11.1: determine (and print) the rms noise
velo
stats,2000,2500, ret=mystats
rms1 = mystats.rms
stats,3500,4000, ret=mystats
rms2 = mystats.rms

rms = (rms1 + rms2) / 2

; Step 3.2.11.2: determine the galaxy's mean intensity using method 1, and calculating
↳ the 20% and 50% velocity widths
gmeasure,1, 0.5,brange=2815,erange=3142,rms=rms
gmeasure,1, 0.2,brange=2815,erange=3142,rms=rms

```

Let's say you have the script saved as HI\_survey\_process.pro. You can execute the script doing the following in a regular terminal

```
gbtidl
```

```
GBTIDL -> @HI_survey_process.pro
```

Once done you should see this output from the code:

```

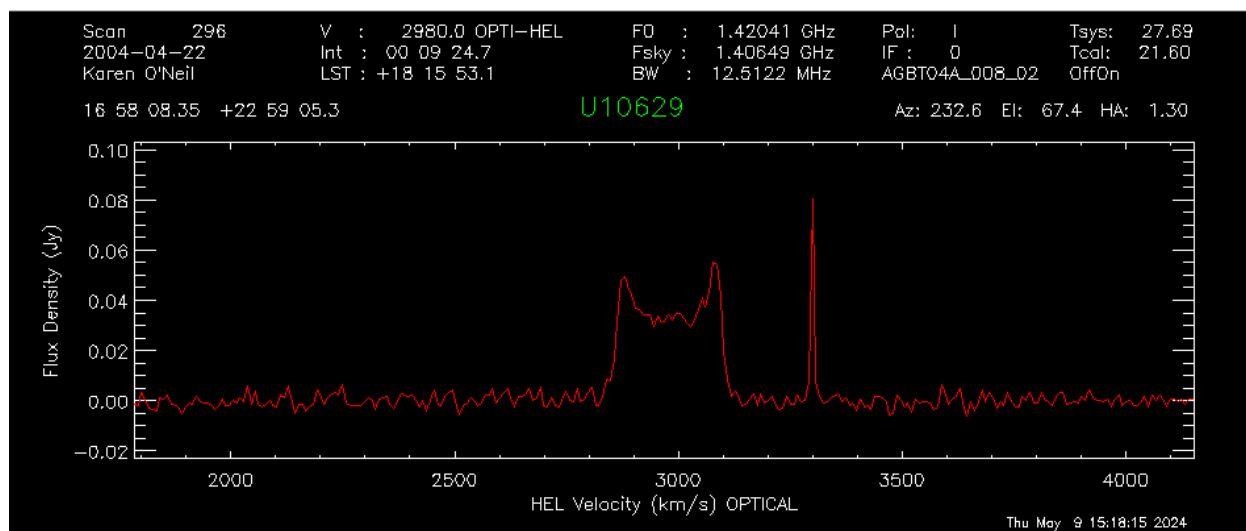
% ACCUMAVE: Average of :          4 spectra
Push return to continue:
% Program caused arithmetic error: Floating divide by 0
  Chans    bchan    echan      Xmin      Xmax      Ymin      Ymax
    63      222      284    1996.4    2503.1   -0.0052186    0.0062189

              Mean      Median      RMS      Variance      Area
      8.7379E-05 -4.4913E-05    0.0025822  6.6675E-06    0.045062
  Chans    bchan    echan      Xmin      Xmax      Ymin      Ymax
    62      40      101    3496.8    4000.3   -0.0062437    0.0065244

              Mean      Median      RMS      Variance      Area
     -1.8582E-06 -0.00018846    0.0024669  6.0857E-06 -0.00095242
Area, Width, Velocity (followed by errors in same order) =    9.43  252.71 2979.58
↳ 0.27    0.00    0.00
Area, Width, Velocity (followed by errors in same order) =    9.43  277.39 2974.65
↳ 0.26    0.00    0.00

```

As well as this spectrum in the GBTIDL Plotter (separate window)





## HOW-TO GUIDES

Practical step-by-step guides to help you achieve a specific goal. Most useful when you're trying to get something done.

---

### 2.1 Infrastructure

These guides provide information on how to use GBO infrastructure.

#### Remote Connection

How to connect remotely to the GBO network?

[Remote connections](#)

#### 2.1.1 How to connect remotely to the GBO network

This connection is needed for performing observations and is also available to process data.

There are multiple ways to connect remotely to the GBO network:

1. *FastX connection*
2. *VNC connection*


##### 2.1.1.1 FastX connection

We recommend the FastX connection using your browser. FastX is a commercial product that provides remote access to Linux systems by giving you a full graphical interface within your browser. No software is required on your machine except a browser. This service is available from anywhere that has a reasonable internet connection.


**Caution:** You should avoid accessing this service from public machines such as at hotels or conferences as these machines are highly likely to be compromised and will record your login credentials.

## Accessing the GBO network


Point your browser at <https://ssh.gb.nrao.edu:3443/>. You will be prompted for a username and password. This should be your NRAO/GBO username and your UNIX password.




Fast and Secure Connections to your Linux desktop



Username



Password



[Manage private keys](#)

SSH Login

Powered by [FastX](#)

Build: [3.3.63](#)

Once you have logged in, you will be greeted by a screen informing you to abide by the terms of the NRAO Computing Security and Computing Use Policies. Click “Proceed”.

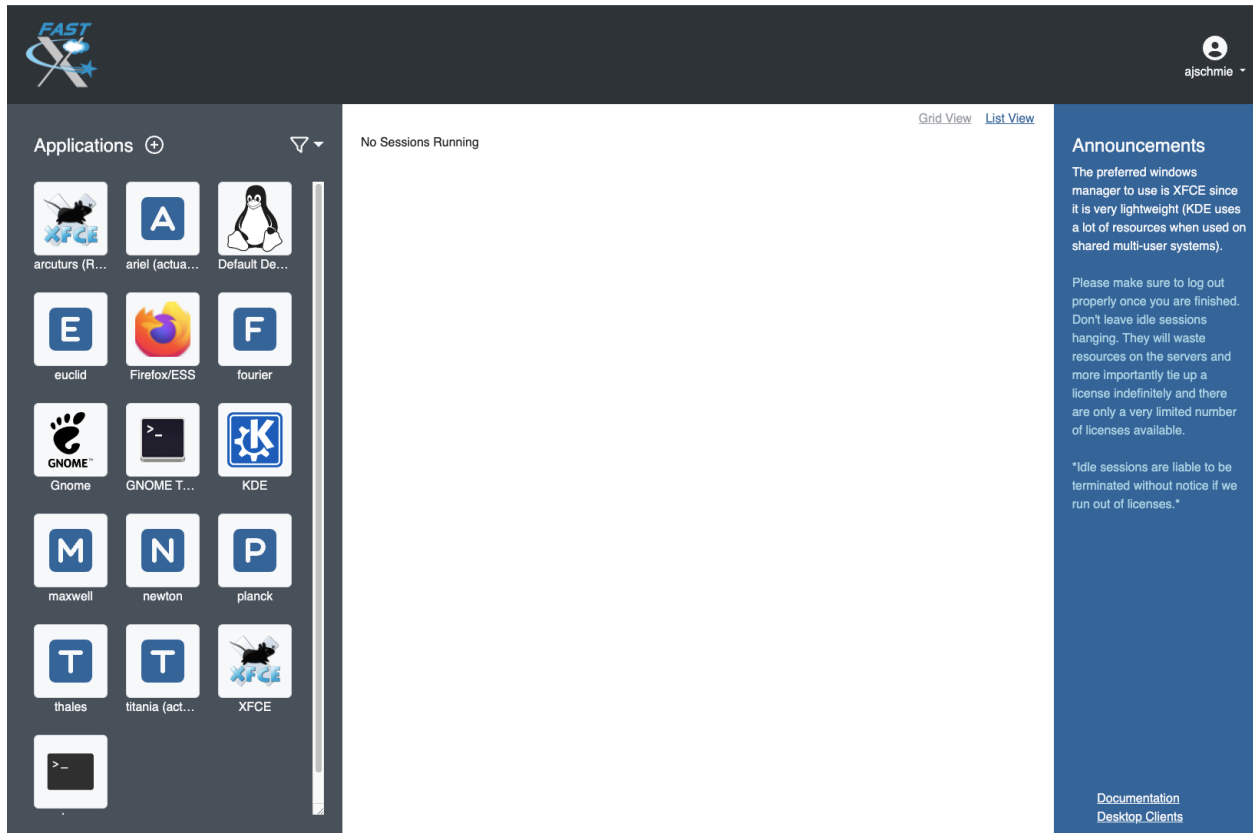


Green Bank Observatory and National Radio Astronomy Observatory computing facilities are exclusively for the use of authorized personnel, who are expected to abide by the terms of the [NRAO Computing Security and Computing Use Policies](#)

Proceed

Next you will see the FastX dashboard.






In the top right corner you can access your User Preferences, learn about FastX and Sign Out.


For your convenience shortcut connections to our different computing machines are provided in the left-hand panel. If **you're actually observing**, pick *ariel* or *titania*. If **you want to process your data**, please choose *euclid*, *fourier*, *maxwell*, *newton*, *planck*, or *thales*.

After selecting your target machine, and if your system supports it, you might see a page that allows you the choice between running the FastX session in the browser or via an app on your computer. For the latter to work, you are required to download and install the app.



How do you want to connect to your session?

 **Browser Client**  
Connect with the browser. No download required!

 **Desktop Client**  
Connect with the desktop client for the best user experience!

☐ Remember my selection

Need the desktop client? [Download the Mac Client here!](#)  
[Need a different client?](#)

You will be asked for your UNIX password one more time and then you should see your XFCE desktop on the machine of your choice. FastX session should resize your Linux screen automatically to adjust to the size of your browser.

### FastX session controls

While in a FastX session, there is a small drop-down menu in the top part of the window that appears when you hover over it and allows some control over the session.



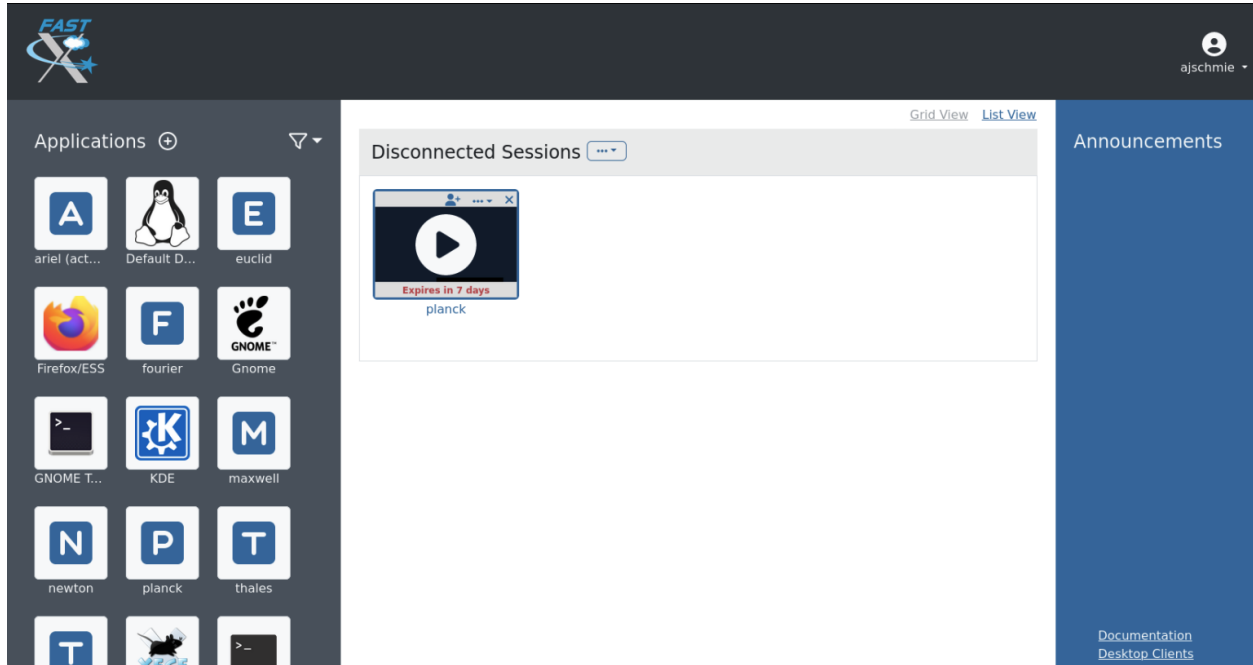
From left to right:

1. **House** brings you back to the FastX dashboard.
2. **Keyboard** allows you to set the keyboard.
3. **People** controls the sharing of the session (more [here](#)).
4. **Folder** allows to open files to the clipboard.
5. **Settings** allows you to change other settings such as the window size. Tinkering with these settings should not generally be required but on a very slow connection lowering refresh rates and picture quality may be helpful.

## Ending a FastX session

When you are finished, click the home button in the control menu (see above) and close the browser tab that contains the session you were working in. In the FastX starting page you will see a session in the center (Disconnected Sessions). This will expire automatically after 7 days, but if you don't need it anymore, please terminate the session to help save resources for your fellow observers.

You can reconnect to a disconnected session by clicking the “play” button.



**Note:** Be aware that if a host is rebooted the session will also disappear. Please also bear in mind that due to licensing only a limited number of concurrent sessions are possible.

## Is this secure?

Short answer: Yes! Access to this service is only available over the HTTPS protocol and as such all traffic between your browser and the server is encrypted.

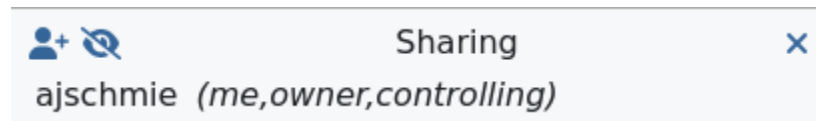
## What can't I do?

You cannot fire up a browser in a session and access the FastX service in that browser. This kind of recursion is not permitted. As usual the [NRAO Computer Use Policy](#) applies to all use of this service.

## Sharing a session

This software makes it possible to share a session in a similar manner to VNC but the sharing is accomplished through a web link containing the session key rather than sharing a VNC password. Sharing of sessions **should only occur where there is a genuine need to do so**, for instance when a member of staff is assisting an observer with actual observations or data reduction.

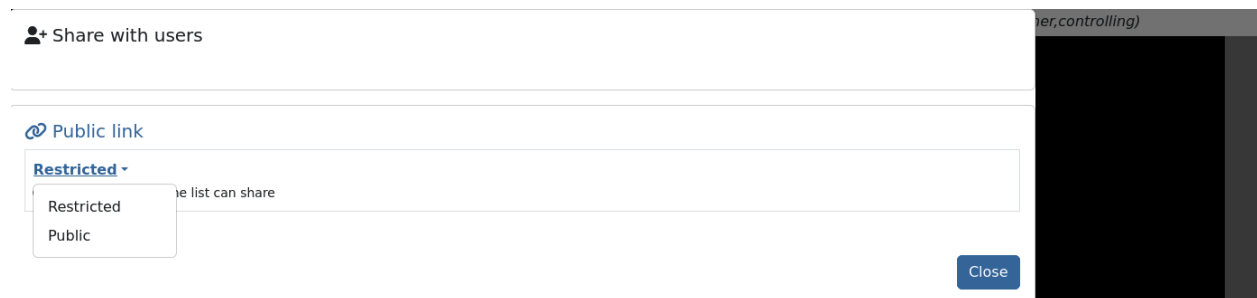
Sharing a session is accomplished by using the sharing icon in the session control drop menu. This will open a second menu in the top-right corner of the browser window.



Click the first icon in the top left (human figure with a plus sign), which will open yet another box.



Click on Public link and switch the options from Restricted to Public.



You can copy the link and share with e.g. the staff member. **This must be done in a secure manner** such as encrypted IM or through Green Banks talk and draw system. **Email is not a secure method**. As the dialogue states, anyone with this link can connect to the session.

**Attention:** Once a session has been shared it must be terminated after use and not left running to connect to later!

### 2.1.1.2 VNC connection

#### Quick VNC Reference

1. Login to GBO `ssh username@ssh.gb.nrao.edu`
2. `ssh` to either titania or ariel by `ssh titania`
3. Start vnc session with `vncserver -your_screen_geometry`. This will give you your session number.
4. **Open a new terminal on your computer with**

```
ssh -N -C -L 590n:titania.gbt.nrao.edu:590n username@ssh.gb.nrao.edu
```

where n is the number you were given in step 3. After entering your password, it will just sit there and that's good.

5. Use your vnc viewer on your computer to view your session.

#### How do I get the necessary software?

VNC comes with most Linux distributions and is easily set up, the same applies to an SSH client. Mac OS and Windows users should download and install a VNC Viewer.

Note Chicken of the VNC is no longer supported. Mac users can use the built-in VNC viewer, RealVNC or TigerVNC.

Mac OS includes a SSH client. You can open a terminal by launching either Terminal or X11, both of which are in your Mac's Applications > Utilities folder.

VNC for Windows is available from [TightVNC](#) or [RealVNC](#). Several commercial versions of VNC are available, but the free edition is suitable for remote GBT observations. For purposes of remote GBT observations, only the VNC viewer has to be installed on your computer. The VNC server has already been installed on the GBT control room computers and other appropriate machines in Green Bank.

You will also need an SSH client. An SSH client allows you to make a secure SSH connection from your work/home machine to the Linux machines in the GBT control room. That is, with SSH client software running on your computer, you can open a terminal window to the remote Linux computer. For Windows users, PuTTY is a freeware SSH client. It is available from [www.chiark.greenend.org.uk/~sgtatham/putty/download.html](http://www.chiark.greenend.org.uk/~sgtatham/putty/download.html). Although other SSH client software exists (e.g. SSH Secure Shell, Secure CRT), our instructions assume you are using PuTTY.

Thus, remote Window users should:

- Download and install VNC unless it is already installed. You only need to install the VNC viewer.
- Download and install PuTTY unless it is already installed on your machine.

#### VNC setup instructions

The example below uses the computer `titania.gb.nrao.edu` for the VNC server. `titania.gb.nrao.edu` is currently the computer usually available to host VNC sessions. As our computer infrastructure changes, this might not always be the case. You should always check with the operator the name of the computer that you should use, and if it is different substitute that computer's name for `titania.gb.nrao.edu` in the instructions below.

If this is your first session ever with VNC, you need to setup a VNC server password and specify which window manager you would like to use.

## Log into the gateway machine

Log into the GBO gateway machine. The steps for Linux and Mac OS are different from those for Windows users:

Open a terminal on your local computer and type

```
ssh stargate.gb.nrao.edu
```

If you are not using an SSH agent, you will be asked to enter your NRAO Linux account username and password.

Start up PuTTY on your Windows machine. A PuTTY configuration window will appear. In the configuration window, specify the host name (*stargate.gb.nrao.edu*) and click on ‘Open’ to obtain a terminal window to the host. After specifying the host name, one can choose ‘Save’ to save the session for future use. If the host name already appears among the ‘Saved Sessions’, double click on the host name to open a terminal window to that host.

---

**Todo:** Add a better quality screenshot here.

---

In the PuTTY terminal window to *stargate.gb.nrao.edu* log in to your GBO Linux account if you are not using an SSH agent.

## How do I provide a VNC password?

**Warning:** Choose a VNC password that is different from your NRAO Linux account as you may later wish to share your VNC password with other who can then watch your VNC session. For example, if you are having difficulties during a remote observing session, you might wish to share your VNC password with a Green Bank staff member who can access your virtual desktop and suggest solutions to your problems.

To create a VNC password, you must *log into a GBO computer*. Once you are logged into the GBO computer, and regardless of your remote computer OS, type at the Linux prompt in the terminal window on the remote computer

```
vncpasswd
```

Then enter your chosen VNC server password.

## How do I set up a Window Manager?

By default, you will be given *Tom’s Window Manager* (TWM) for VNC sessions, which is probably one of the most arcane and archaic window managers around. If you would like to use the more intuitive and modern K Desktop Environment (KDE) window manager, you can perform the following steps:

*Log into a GBO computer*. Once you are logged into the GBO computer, and regardless of your remote computer OS, type at the Linux prompt in the terminal window on the remote computer

```
mkdir ~/.vnc
```

---

**Note:** If you have already used VNC in the past, this step is not necessary as you will get a message stating that the ‘file’ already exists. This is normal.

---

If you want to use the *kdewindow* manager, at the Linux prompt, type:

```
cp /users/ccclark/xstartup-KDE ~/.vnc/xstartup
```

## VNC session instructions

Each time you observe remotely with the GBT, you will need to follow these instructions.

**Caution:** Please do not start more than one VNC session at a time! There is only a limited total number of sessions and ports available at any time. Likewise, don't forget to end your VNC server session when you have finished observing. For instructions on ending your VNC server, please read [How do I close my VNC session?](#).

## How do I start a VNC session?

To set up a session at the remote machine you must first [log into a GBO computer](#). Once you are logged into the GBO computer, and regardless of your remote computer, type at the Linux prompt

```
ssh titania.gb.nrao.edu
```

If you are not using an SSH agent, you will be asked to enter your GBO Linux account username and password. At the Linux prompt on *titania* type:

```
vncserver
```

**Note:** If this is your first session in Green Bank, you will be asked which password should protect your future sessions. As noted above, choose a VNC password that is different from your Linux account password as you may later wish to share this password with a Green Bank staff member for troubleshooting your problem.

Once you have typed `vncserver`, the system will reply with e.g.

```
New 'titania:2 (USERNAME)' desktop is titania:2
```

In this example the VNC session number is 2. Remember this number since you will need it later. In the instructions below, the VNC session number is designated as **n**.

Leave this connection to the server open, as it will be used later to kill processes on *titania* associated with the VNC session.

## How do I start a VNC viewer on my local machine?

You will need to establish a SSH tunnel and start a VNC viewer in order to view the session on your local machine (work/home). The process is different for each computer platform:

To establish an SSH tunnel, open a terminal on your Mac and type:

```
ssh -N -C -L 590n:titania.gb.nrao.edu:590n YOURLOGIN@stargate.gb.nrao.edu
```

Replace **n** with the desktop number from the previous step and YOURLOGIN with the name of your GBO Linux account. If you are not using an SSH agent, you will be prompted for your GBO Linux account password.

To start a VNC viewer, launch TigerVNC from your Mac's Applications folder.

---

**Todo:** Add screenshot of tiger vnc.

---

Set Host to localhost, set Display to **n**, enter your VNC password, check Remember Password, and check Allow other clients to connect and hit the Connect button. The VNC Viewer window to titania will now appear. In this window you can start astrid and cleo, open xterm, etc, almost as if you were sitting in front of a titania screen.

To establish an SSH tunnel AND start a VNC viewer, open a terminal on your local computer and type:

```
vncviewer -Shared -via YOURLOGIN@stargate.gb.nrao.edu titania.gb.nrao.edu:n
```

Replace **n** with the desktop number from the previous step and YOURLOGIN with the name of your GBO Linux account. The -Shared option allows support staff to “snoop” on your session when assisting you. If you are not using an SSH agent, you will be prompted for your GBO Linux account password. You will next be prompted for your VNC password, which should be different from your GBO Linux account password and sharable with support staff. You will be asked for this password regardless of whether or not you have an SSH agent running.

---

**Note:** If your Linux version does not support the -via option you might want to install [Tight VNC](#) or use the following equivalent to -via. Open a terminal on your local computer and type:

```
ssh -N -C -L 590n:titania.gb.nrao.edu:590n YOURLOGIN@stargate.gb.nrao.edu &
```

Open a terminal on your local computer and type:

```
vncviewer -Shared localhost:n
```

---

The VNC Viewer window to titania will now appear. In this window you can start astrid and cleo, open xterm, etc, almost as if you were sitting in front of a titania screen.

**To establish an SSH tunnel:**

Start another instance of PuTTY and bring up another PuTTY Configuration window. In this second PuTTY window, enter the following information under each Category (listed in the left panel of the window). This information can be saved for future use. Session - Host Name is stargate.gb.nrao.edu. (If stargate.gb.nrao.edu is already listed among the Saved Sessions, click on it and choose Load to place that name in the Host Name section of the window.)

Connection > Data - Enter your Green Bank Linux account login name as the Auto-loginusername.

Connection > SSH - Select Enable compression.

Connection > SSH > Tunnels - Remove any previously used ports with the Remove button. For Source port enter 590n, where n is the VNC session number reported in the first PuTTY window (the VNC server). For Destination, enter titania.gb.nrao.edu:590n Then choose Add, then Open.

A terminal screen will open to stargate.gb.nrao.edu. If you are not using an SSH agent, you will be prompted for your GBO Linux account password.

You need to type nothing else in this window except exit at the end of the VNC session. The existence of this window serves only to provide the tunnel from your Windows machine to the Green Bank system.

**To start the VNC viewer:**

Start the VNC viewer on your Windows machine. If using TightVNC, please select the viewer having the “best compression.” A popup window will appear, VNC Viewer: Connection Details. Click on Options. Search for and select the option for sharing the connection and then click OK. For Server enter localhost:n, where **n** is the VNC session number, as before.



Next a VNC Viewer: Authentication window will pop up. Enter your vncserver password (not your GBO Linux password).

The VNC Viewer window to titania will now appear on the screen of your Windows machine. In this window you can start astrid and cleo, open xterm, etc, almost as if you were sitting in front of a titaniascreen.

### What special features does the VNC viewer window have?

- Depressing the left mouse button inside the main VNC Viewer window brings up a menu, allowing you, for example, to open another xterm.
- Minimizing a window - Click on the upper left part of a window.
- Resizing a window - Use the icon at the upper right of a window.
- To configure your VNC Viewer, press F8 and a menu should appear.

### How do I close my VNC session?

**Important:** If you stop using your VNC viewer but don't kill the `vncserver`, your session stays alive and uses computing resources. If you run your VNC viewer a few hours later or from a different computer, you can continue where you left off. However, there is a limited number of sessions and ports available at any one time. Unless you know you will be using the session again within a few hours, please do not leave `vncserver` running. If you do, the GBT operator is likely to kill your `vncserver` session within 24 hours.

Also, please **do not start more than one VNC server at a time**.

To stop your VNC session, first close your VNC viewer, then kill your VNC server. You'll need a terminal window that is logged into an NRAO computer. In the terminal window on the remote GBO computer, at the Linux prompt type:

```
ssh titania.gb.nrao.edu
```

Once you are on *titania* type

```
vncserver -kill :n
```

where **n** is the VNC session numbers. Disconnect from *titania.gb.nrao.edu* by typing `exit` and then disconnect from *stargate.gb.nrao.edu* by typing `exit` again.

### Troubleshooting Information

There have been times when a local port is taken by another user or another VNC session. In these rare cases, the recommended port forwarding won't work. To determine if a port is used, the terminal command,

```
netstat -a | grep :59
```

This will list all used ports (there may be a delay of a few seconds before this list appears). In these cases, the tunnel has to be changed to `590m:stargate.gb.nrao.edu:590n` where `590m` is some unused port numbered somewhere above 5900. And wherever `localhost:n` occurs in the above instructions, substitute with `localhost:m`.

If you would like to check if you are already running a VNC server (really, there should only be one), use any terminal that is logged into an NRAO computer and type:

```
ls ~YOURLOGIN/.vnc | grep .pid
```

## Windows short instructions

Once you have done remote observing with VNC under Windows, the instructions below will serve as a reminder of the steps involved.

### 1. Start VNC server on titania

- Start PuTTY, Host Name is *stargate.gb.nrao.edu* (prospero on site), choose 'Open' to open terminal window, log in to Linux account.
- `ssh titania.gb.nrao.edu`
- `vncserver -geometry 1200x1000 -depth 8` (switches and numeric values may vary)
- Note VNC session number, **n**

### 2. Set up SSH tunnel

- Start PuTTY again, Host Name is *stargate.gb.nrao.edu* (prospero on site)
- *Connection > Data* - Set Auto-login username to linux account user name
- *Connection > SSH > Tunnels* - Source port is 590**n**, destination is *titania.gb.nrao.edu:590n*
- Choose 'Add', then 'Open' to open the second terminal window, enter GBO Linux account password

### 3. Start VNC viewer on Windows machine

- First popup window, enter `localhost:n` for Server
- Second popup window, enter VNC password

### 4. Closing down VNC session

- X out of VNC window after closing astrid, cleo etc.
- In first PuTTY terminal window (where `vncserver` was started) enter `vncserver -kill :n`
- exit out of second PuTTY window.

#### 2.1.1.3 What to do if the GBO network is down/slow

Occasionally we experience network issues where either the entire Green Bank network is down or intermittent which causes FastX and VNC sessions through Green Bank to be extremely slow.

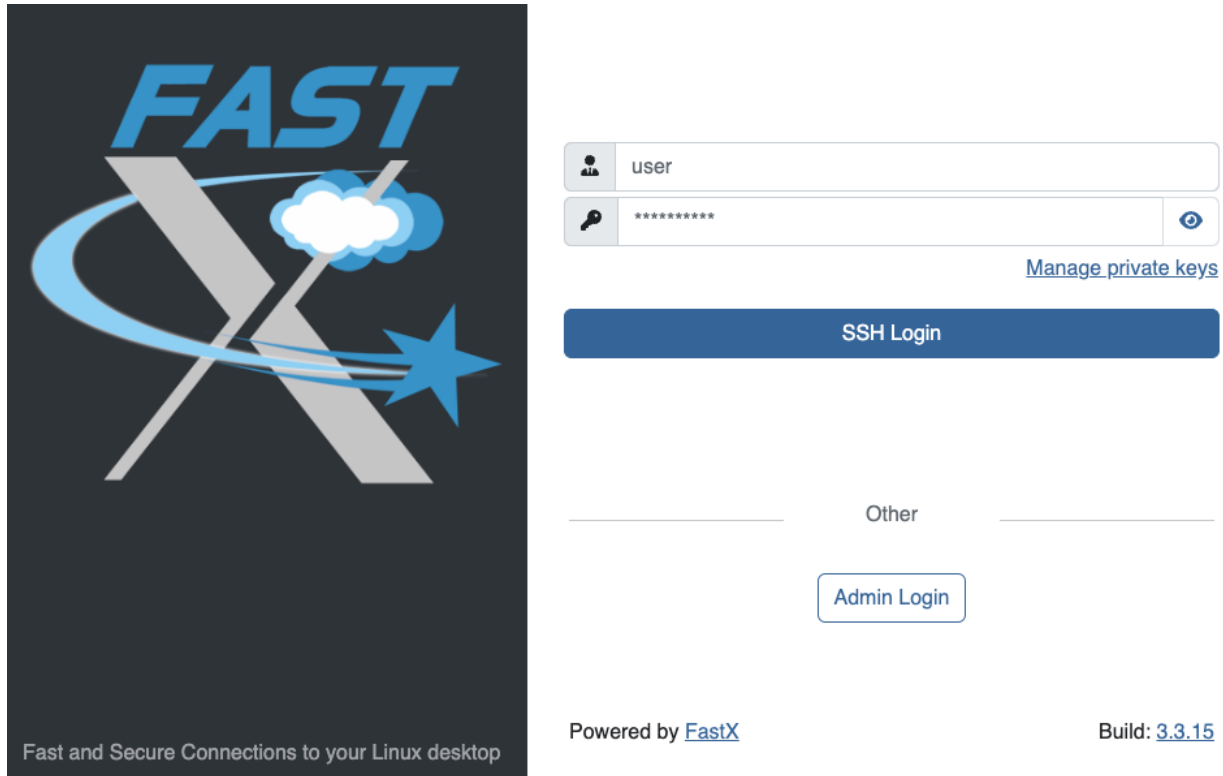
Suggested checks:

- If you are unsure which end of the internet is slow, try running a speed test on your network connection. If your speed levels are where they should according to your internet provider, the problem is likely on the Green Bank end of the network.
- Check [this status page](#) to see the status of the ssh gateways. Orange means that it was down for a small portion of that time slice. Red is down for a significant portion of the time span that slice of the bar covers. Each vertical bar is a day.

In the case that the internet issue is on the GBO side, you can run your FastX or VNC connection through Charlottesville. Or with FastX, if you get kicked out, just keep trying to re-connect.

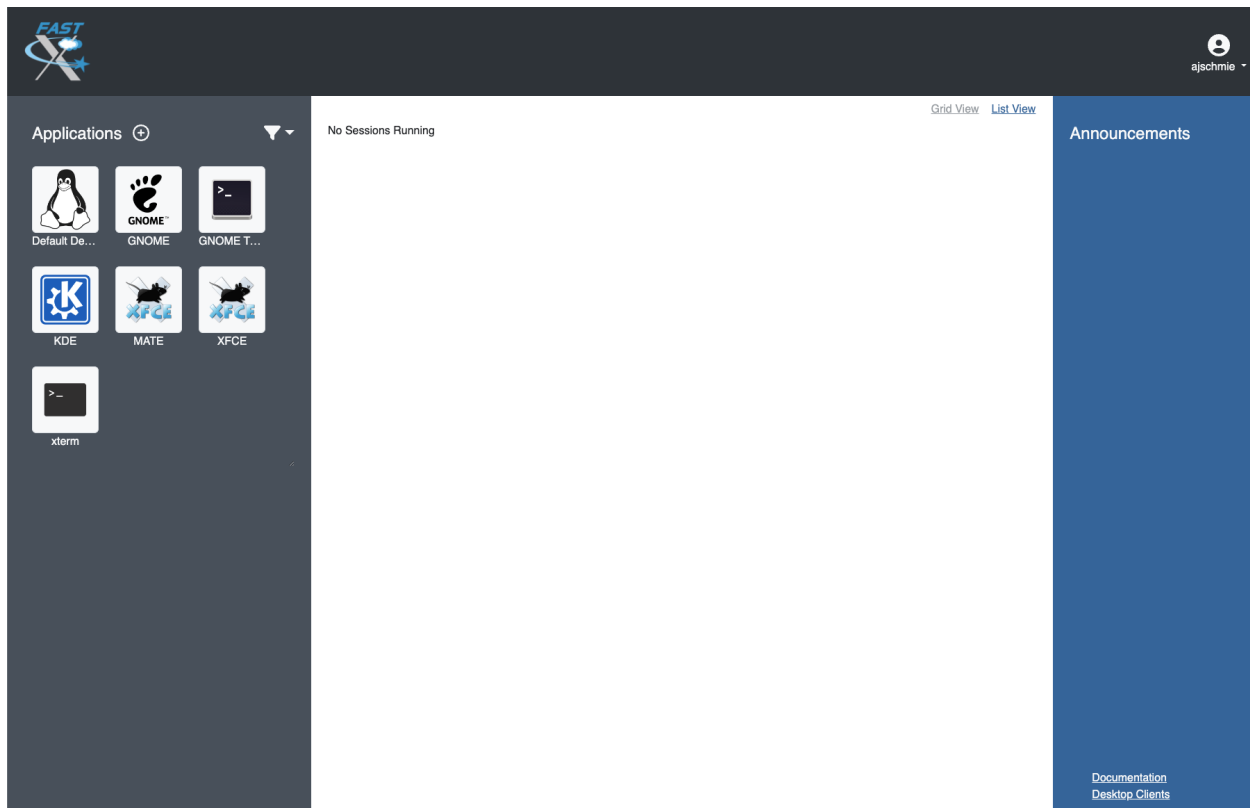
## FastX through Charlottesville

Point your browser to <https://fastx.cv.nrao.edu>. You will be prompted for a username and password. This should be your NRAO/GBO username and your UNIX password (the same you're using to connect to the GBO network described above).



The image shows the FastX login interface. On the left is a large graphic with the word "FAST" in blue, a large grey "X" with a blue swoosh and star, and the text "Fast and Secure Connections to your Linux desktop" at the bottom. On the right is the login form. It has two input fields: the first is for the username, with a user icon and the text "user"; the second is for the password, with a key icon, masked text "\*\*\*\*\*", and a toggle icon. Below the password field is a link "Manage private keys". A large blue button labeled "SSH Login" is below the inputs. Below that is a section titled "Other" with a button labeled "Admin Login". At the bottom, it says "Powered by [FastX](#)" and "Build: [3.3.15](#)".

Once you have logged in, you will be greeted by the same screen informing you to abide by the terms of the NRAO Computing Security and Computing Use Policies again. Click “Proceed” to continue to your FastX dashboard.



In the application panel on the left-hand side, you might not see a lot of preset connections yet. You can create your own connection by clicking the circled plus sign next to “Applications”. A pop-up will open. Provide the following details:

- Command \*: `ssh <machine-name>.gb.nrao.edu startxfce4`
- Run As User: `<your user name>`
- Name: `<name of the machine or whatever identifier you want to use>`

Available machine names are:

- **for observing only:** *ariel, titania*
- **for data processing:** *euclid, fourier, maxwell, newton, planck, thales*

Command \*

ssh ariel.gb.nrao.edu startxfce4

Window Mode

Single

1024

x

768

Profile

---- Default Profile ----

Run As User

ajschmie

Name

Ariel

Launch

You can permanently save the connection details, by clicking the down-arrow next to Launch and selecting “Create Application”. Then click Launch to start the connection.

Command \*

ssh ariel.gb.nrao.edu startxfce

Window Mode

Single

1024

x

768

Profile

---- Default Profile ----

Run As User

ajschmie

Name

Ariel

Launch

Create Application

Clear History

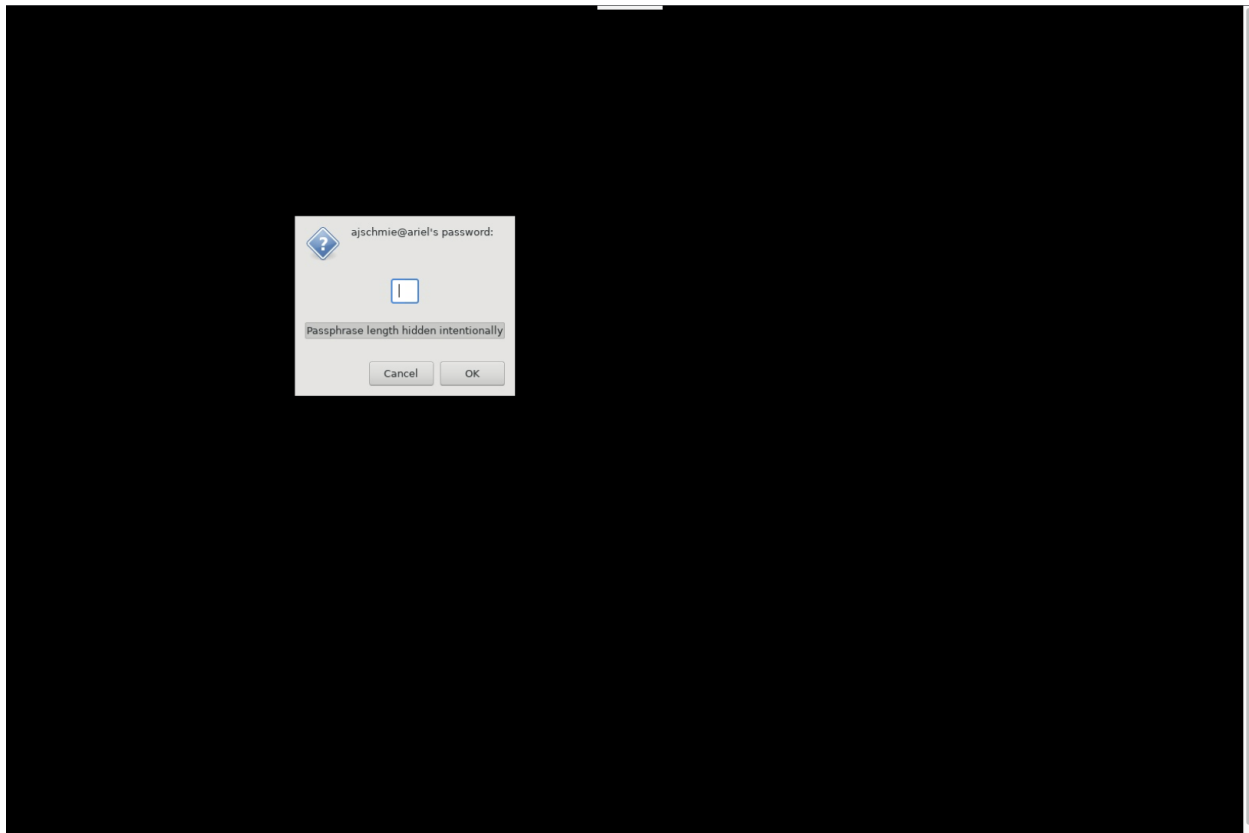
You will be asked for your password twice: (1) to establish the ssh connection to a machine called polaris and then (2) to log into the machine you choose (ariel in the example here). You may experience longer load times to start the session, but once you're logged in, things usually work well.

SSH Login: **ajschmie@polaris**

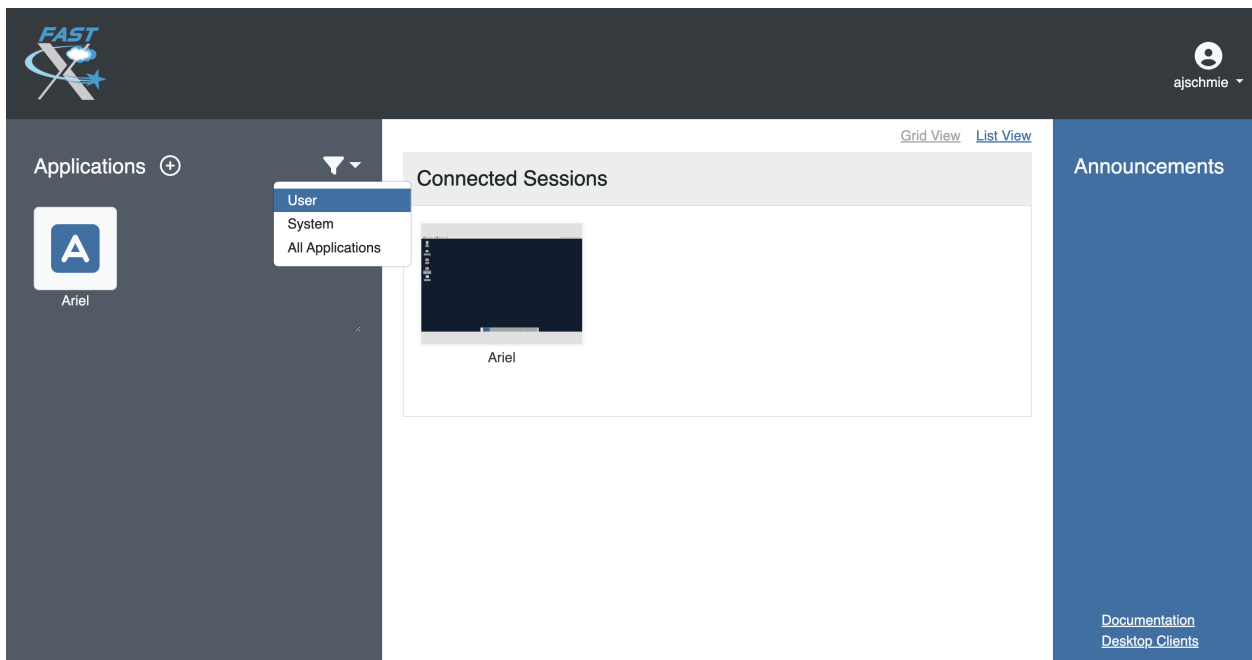
Password:

Cancel

Submit



If you don't see your newly created application in the dashboard, you might need to change the filter settings by clicking the filter icon and selecting "User".



## VNC through Charlottesville

If you cannot even connect to green bank to start your VNC session through Charlottesville instead:

```
ssh username@ssh.cv.nrao.edu
ssh titania
vncserver -geometry dimension1xdimension2 :n
```

---

**Hint:** Manually choose a higher number session to decrease the likelihood that that number is being used on either a GBO or Charlottesville computer.

---

Then make your tunnel

```
ssh -N -C -L 590N:titania.gbt.nrao.edu:590n username@ssh.cv.nrao.edu
```

And with your VNC viewer connect to localhost:590n

---

**Hint:** You may encounter an error when trying to create a tunnel through Charlottesville `bind [\\:\\\\#]:590\\#:` address already in use, `channel_setup_fwd_listener_tcpip: cannot listen to port: 590\\#`, could not request local forwarding This was thought to mean that that # port was already taken on a Charlottesville computer so you will have to choose a different VNC session number. Hence the earlier suggestion to manually choose a session number that is higher initially to avoid such issues.

---

---

### Acknowledgement

We would like to thank Joeri van Leeuwen (UBC), Tom Troland (U. Kentucky), and Jeff Mangum (NRAO CV) who kindly provided the entire content of these VNC instructions. We appreciate the time they took to make VNC easier for all remote observers.

---

## 2.2 General Guides

These guides provide information on how to perform general and common tasks with the GBT.

### Observing Instructions

Most basic steps to observe with the GBT

[GBT observing](#)



## 2.2.1 GBT observations 101

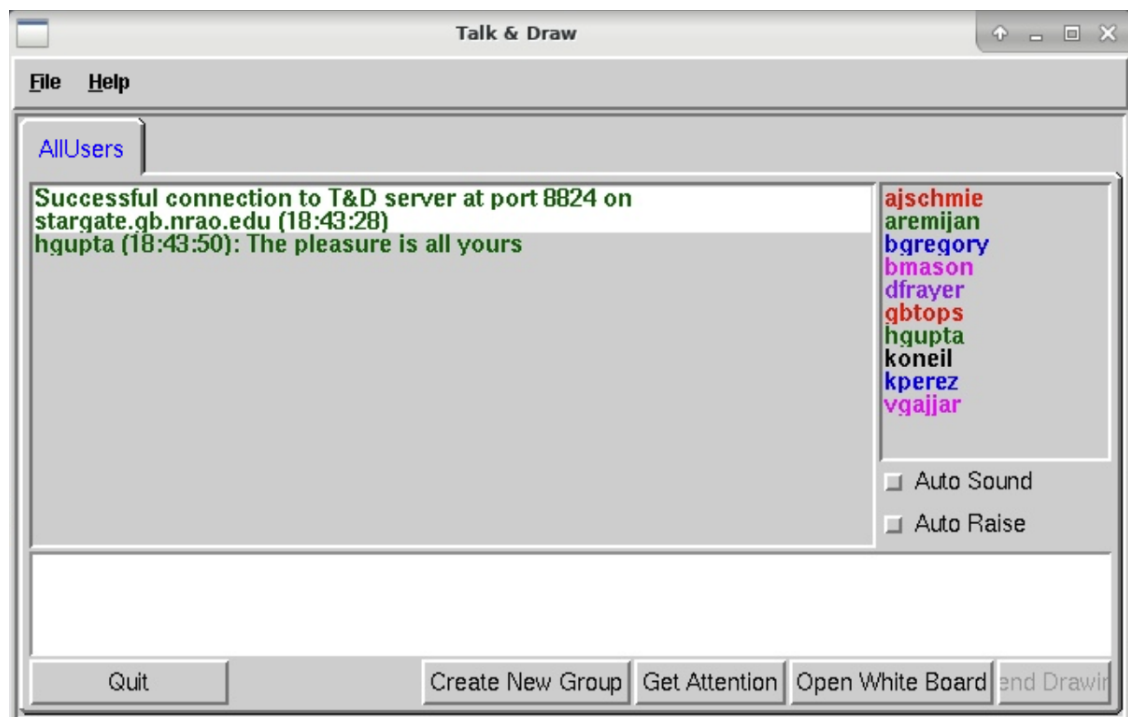
We assume you are either on-site or are using a *remote connection*.

### 2.2.1.1 Observing Preparation

Latest 30 minutes before the start of your observing session (give it more time if you're new to the GBT):

1. Log into *ariel* or *titania*, one of the two dedicated observing computers.
2. **Start CLEO's talk & draw application `cleo talk`**

**Note:** If the fontsize in talk&draw is unreadably small, close all open CLEO applications and run `xrandr --dpi 96` in a terminal. Then reopen your CLEO application.



3. **Contact the telescope operator 30 minutes before the start of your observing session**

- either via talk & draw or
- call via telephone: The contact number is +1-304-456-2346 or the Operator's direct line at +1-304-456-2341 (x2346 connects to a speaker phone and is the preferred number to use).
- Tell the operator who you, your project, and what computer you will be running your observations from (*ariel* or *titania*). They usually will tell you right away, if not, ask for the operator's name (you will need that later).

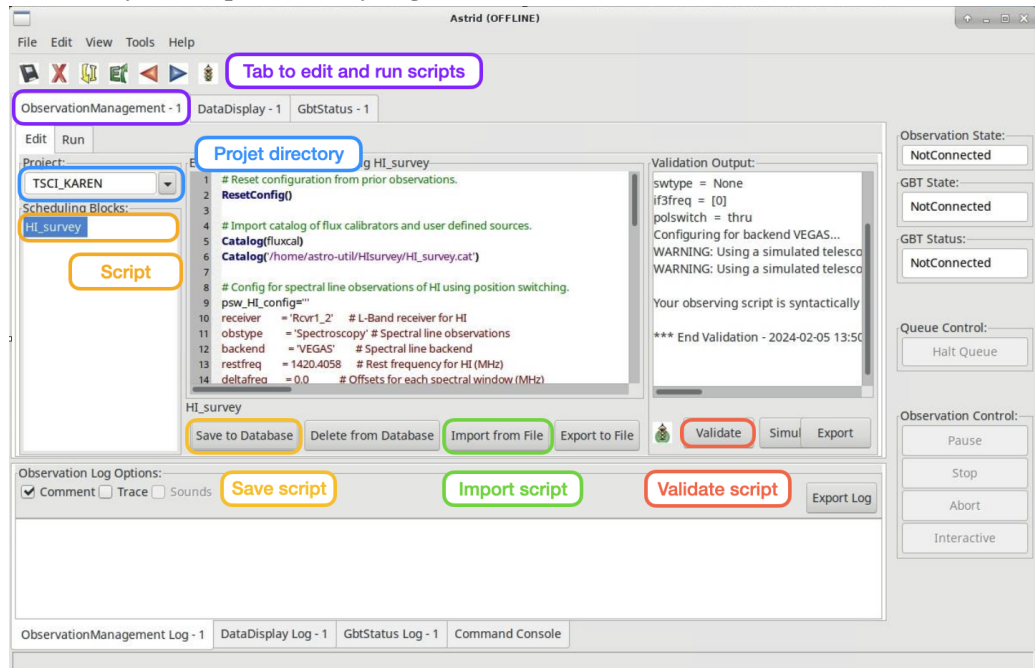
4. **Start AstrID, type `astrid` in a terminal**

- Choose "Work Offline" mode when prompted



## 5. In AstrID:

- Make sure all your scripts are ready to go:

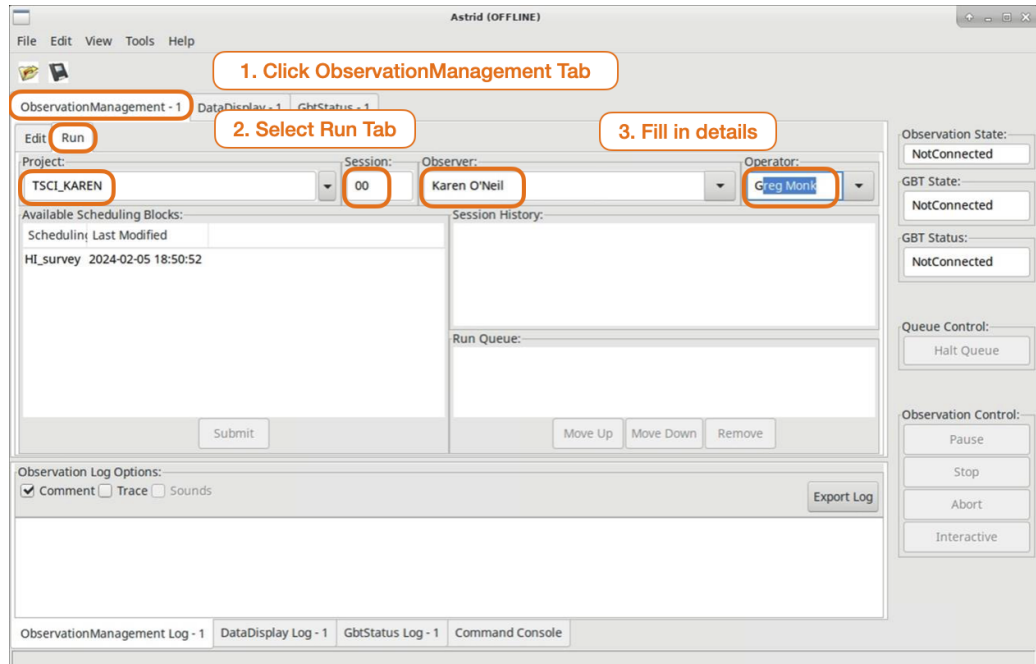


- Fill in your project details in the ObservationManagement Run Tab

- Enter your project (if not there already)
- Enter your name as observer.

**Note:** The dropdown menu is very long and impractical. Instead start typing your first name and it will bring up the options that match what is typed. Keep typing until you see your name.

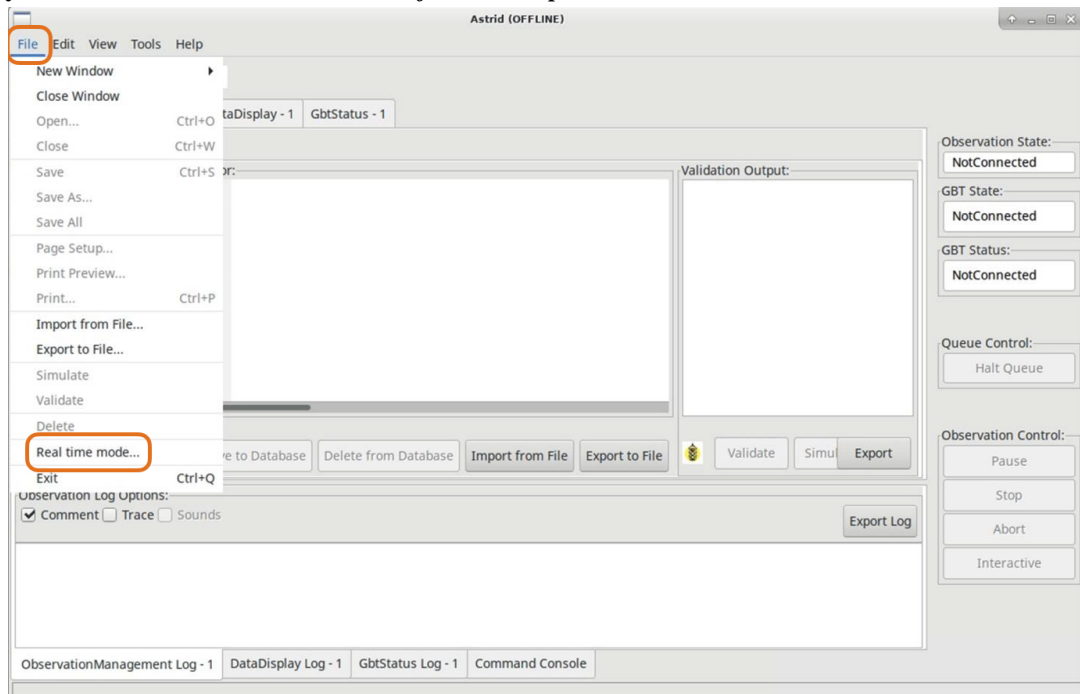
- Enter the telescope operator's name.

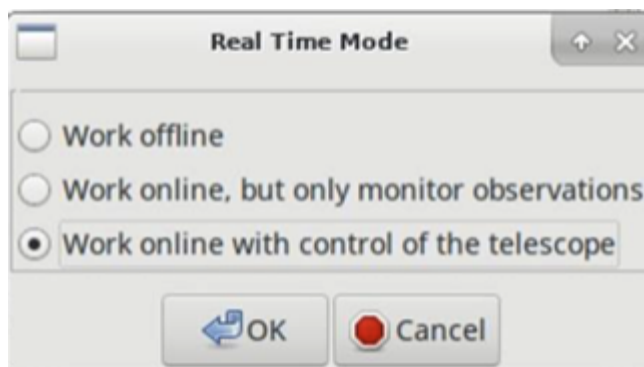


### 2.2.1.2 Observing

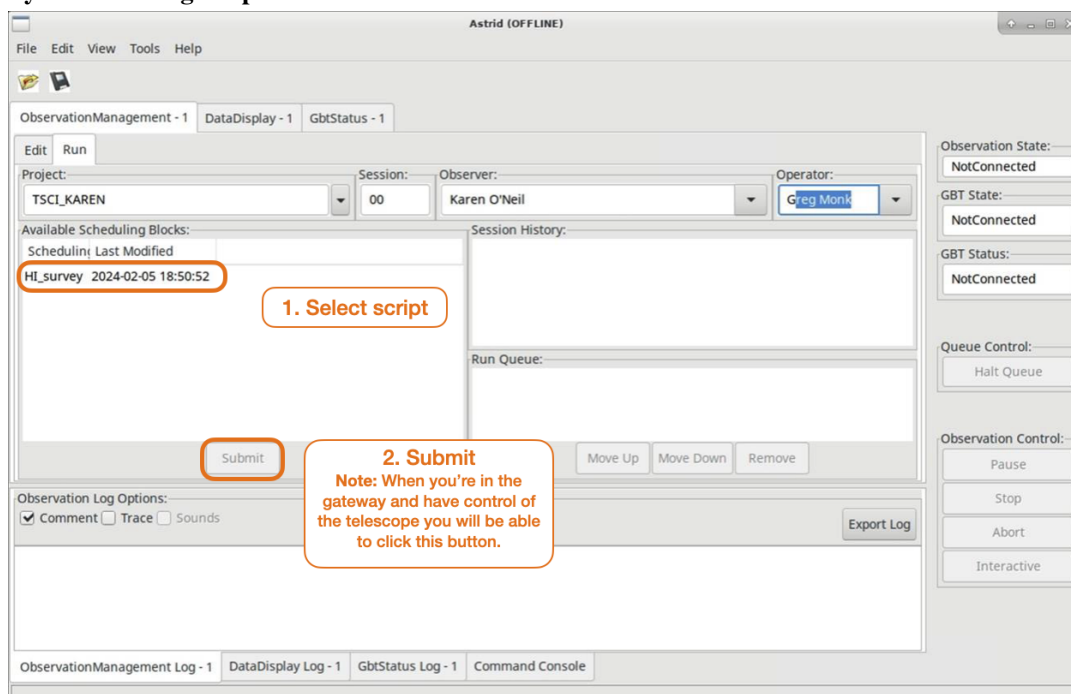
Once the operator lets you know your observing time has started and they give you security access (put you in the “gateway”), you should

1. Put your Astrid into *online with control of the telescope mode*





## 2. Load your observing script and submit it



## 2.3 Observing Mode Guides

These guides contain end-to-end instructions, i.e. how to set up specific observations and how to calibrate and post-process the obtained data.

### Polarization

How to derive a Mueller matrix?

[Mueller matrix](#)

### 2.3.1 How to derive a Mueller matrix

The instructions below will run you through the steps required to setup your scheduling blocks to execute the observations required to measure the Mueller matrix for the C-band receiver.

#### 2.3.1.1 Observing Script

At the GBT we use AstrID to prepare and execute scheduling blocks.

##### Catalog

Before you start writing your scheduling block it is helpful to prepare a source catalog. We will write the catalog in our scheduling block. For measuring a Mueller matrix it is important to select a polarized source. 3C286 and 3C138 are good choices.

```
source_catalog = """
format = spherical
coordmode = J2000
HEAD = NAME      RA      DEC
3C286      13:31:08.28  +30:30:32.9
3C138      05:21:09.88  +16:38:22.0
"""
```

##### Configuration

Next we will define the configuration for the observations.

```
config = """
receiver      = 'Rcvr4_6'          # C-band receiver
obstype       = 'Spectroscopy'    # Specifies spectral line
↳observations
backend       = 'VEGAS'            # Specifies spectral line backend
restfreq      = 4500, 5100, 5800, 6700, 7400 # Specifies rest frequencies (MHz)
bandwidth     = 1080               # Specifies spectral window width
↳(MHz)
nchan         = 16384              # Specifies number of channels in
↳spectral window
vegas.subband  = 1                 # Specifies single or multiple
↳spectral windows (1 or 8)
swmode        = 'tp_nocal'         # Specifies switching mode
↳switching power with no noise diode
swtype        = 'none'             # Specifies type of switching, no
↳switching
tint          = 3                 # Specifies integration time
↳(seconds; integer multiple of swper)
swper         = 3                 # Specifies length of full
↳switching cycle (seconds)
vframe        = 'topo'             # Specifies velocity reference frame
vdef          = 'Radio'           # Specifies velocity definition
noisecal      = 'hi'              # Specifies level of the noise
```

(continues on next page)

(continued from previous page)

```

↪diode, use 'hi' for measuring a Mueller matrix
pol                = 'Linear'                # Specifies 'Linear' or 'Circular'
↪polarization
vegas.vpol         = 'cross'                # Specifies recording of the full
↪Stokes polarization products
dopplertrackfreq   = 5800                  # Specifies the frequency used for
↪Doppler tracking (MHz)
"""

```

For measuring a Mueller matrix all four Stokes products have to be recorded (vegas.vpol='cross') using the high power noise diode (noisecal='hi') to have a higher signal-to-noise when fitting for the phase slope between polarizations. As will be discussed later, the integration time should be chosen so that enough samples are recorded over the beam with an appropriate signal-to-noise.

## Scheduling Block

Now we have the ingredients required to build a scheduling block. We will use the catalog and configuration we defined previously and also incorporate the observing procedures. For this observation we will use the polarized source as our calibrator.

For observations using the Spider scan the default behavior is that the telescope will cross over the specified location four times (slices=4), creating eight spider legs over the source. The time it takes to complete a leg is set in seconds by the scanDuration argument. The length of each leg is specified by the startOffset argument, and it defines half of the length of the leg. That is, if we use startOffset=Offset('AzEl', '00:40:00', '00:00:00', cosv=True), the leg will be 80' long. Between each leg the noise diode will be fired for 10 seconds (calDuration=10). From these parameters we can estimate how long each Spider procedure will last, including the 20 s overheads between scans

$$\text{spiderDuration} = (\text{scanDuration} + 20 \times 2 + 10 \times 2) \times 4$$

To determine the values for startOffset, scanDuration and tint you should think about your science goals, and remember that these parameters are related via

$$\text{tint} = \frac{\text{HPBW}}{(2 \times \text{startOffset}) / (\text{scanDuration})} \frac{1}{\text{samples}}$$

with HPBW the half power beam width at the lowest observing frequency, and # samples the number of data points over the beam. If you want to observe a point source, the full beam response might not be important, but if your source is extended you will want to map your beam. In this example we will assume you are interested in the full beam response, so we will set # samples=5. For the length of the spider legs you want to make sure that there will be enough blank sky to calibrate the observations. Here we will use a size of five times the HPBW (3.1' at 3960 MHz), so startOffset=Offset("AzEl", "00:15:30", "00:00:00", cosv=True). Then, the integration time is tint=0.02scanDuration, so if we use 150 s for scanDuration, then tint=3.

```

# Example scheduling block for carrying out Mueller matrix observations.

# Reset configuration from prior observation.
ResetConfig()

# Define a source catalog and tell the system about it.
source_catalog = ""
format = spherical
coordmode = J2000
HEAD = NAME      RA              DEC

```

(continues on next page)

(continued from previous page)

```

3C286          13:31:08.28      +30:30:32.9
3C138          05:21:09.88      +16:38:22.0
"""

Catalog(source_catalog)

# Define the configuration we will use.
config = """
receiver       = 'Rcvr4_6'
obstype        = 'Spectroscopy'
backend        = 'VEGAS'
restfreq       = 4500, 5100, 5800, 6700, 7400
bandwidth      = 1080
nchan          = 16384
vegas.subband  = 1
swmode         = 'tp_nocal'
swtype         = 'none'
tint           = 3
swper          = 3
vframe         = 'topo'
vdef           = 'Radio'
noisecal       = 'hi'
pol            = 'Linear'
vegas.vpol     = 'cross'
dopplertrackfreq = 5800
"""

# Perform position and focus correction on the calibrator.
AutoPeakFocus("3C286")

# Reconfigure after position and focus corrections.
Configure(config)

# Slew to your calibrator.
Slew("3C286")

# Balance the IF system.
Balance()

# Define the parameters for the Spider procedure.
scanDuration   = 150 # seconds
startOffset    = Offset('AzEl', '00:15:30', '00:00:00', cosv=True)

# Estimate how many complete Spiders we can fit into
# our observing session.
# How much time do we have?
# Assume 3 hours for this example.
totalTime      = 3 * 60 * 60.
spiderTime     = 4*(scanDuration + 20*2 + 10*2)
numberSpiders  = int(totalTime/spiderTime)
print("Will observe {0} Spiders".format(numberSpiders))

for i in range(numberSpiders):

```

(continues on next page)

(continued from previous page)

```
Spider("3C286", startOffset, scanDuration)
```

### 2.3.1.2 Data Reduction

To be done

---

## 2.4 Receiver specific Guides

These guides provide receiver specific information on how to achieve a specific goal.

### Argus

How-to observe with Argus (75-115 GHz)

[Argus](#)

### MUSTANG-2

Guides for MUSTANG-2 (bolometer camera)

[MUSTANG-2](#)

### 2.4.1 How to observe using Argus

This guide contains instructions for the recommended observing procedure for the 23B/24A winter observing season.

---

#### Change Log

##### 2024-01-04:

- Adjusted instructions to reflect change of available receivers
- Modified code snippet in [3.1 Pointing and Focus Correction](#) to apply  $\Delta_{\text{focus}}$  when switching from Argus to “peak/focus receiver”

##### 2023-11-13:

- Added code snippet in [3.1 Pointing and Focus Correction](#) to automatically apply  $\Delta_{\text{focus}}$ .
- 

#### 2.4.1.1 1. Turn on Argus

Run the script below and then make sure Argus is turned on.

```
ResetConfig()

# start up Argus by calling the 'presets' command
# this will turn on CIF/LNA if necessary

SetValues("RcvrArray75_115", {"presets": "on"})
print "CIF Power: ", GetValue("RcvrArray75_115", "cif_power")
print "LNA Power: ", GetValue("RcvrArray75_115", "lna_power")
```



In the “ObservationManagement Log” in Astrid, make sure you see the lines “CIF Power: on” and “LNA Power: on”. If not, reissue the startup script above.

```
[03:22:39]***** Begin Scheduling Block
[03:22:39]***** observer = Anika Schmiedeke, SB name = 00_argus_startup, project ID = TGBT23B_604, date = 26 Oct 2023

[03:22:39]*** Notice:Resetting the configuration. You must run another configuration before running a scan
[03:22:40]CIF Power:  on
[03:22:45]LNA Power:  on

[03:22:46]***** End Scheduling Block
```

ObservationManagement Log - 1	DataDisplay Log - 1	GbtStatus Log - 1	Command Console
-------------------------------	---------------------	-------------------	-----------------

### 2.4.1.2 2. Primary Calibrator

Go to your primary calibrator (the brightest calibrator available at the time of your observations).

#### 2.1 Set the surface

Run **AutoOOF** using *Ka-band* to set the surface (you can use Argus, if your calibrator is strong enough; e.g. 0319+4130 is a very strong calibrator that Argus can use for autooof).

```
Catalog('fluxcal')
Catalog('/home/astro-util/astridcats/wband_pointing.cat')

source = '0319+4130' # replace with your calibrator
Break("Ask the operator to switch to Ka-Band")
AutoOOF(source)
```

#### 2.2. Absolute flux calibration, focus reference

Run **AutoPeakFocus** and **AutoPeak** using *Argus* at your target frequency for absolute flux calibration. You can optionally use `elAzOrder=True`, which will run the elevation pointing scans first, followed by the azimuth pointing scans. We typically have larger offsets in elevation than in azimuth, so using this option facilitates finding the pointing solutions.

```
source = '0319+4130' # replace with your calibrator
freq_argus = 93173.0 # replace with your target frequency in MHz

Break("Ask the operator to switch to Argus. Click yes when Argus is in place.")
SetValues("ScanCoordinator", {"receiver": "RcvrArray75_115"})
SetValues("L01", {"restFrequency_A": freq_argus})

AutoPeak(source, frequency=freq_argus, elAzOrder=True)
Break("Check the pointing solution")
Focus(source)
Break("Check the focus solution")
AutoPeak(source, frequency=freq_argus, elAzOrder=True)
```

## 2.3 Determine focus offset

Run **AutoPeakFocus** using *Ka-Band* to determine the focus offset between Argus (at your frequency) and Ka-Band (at the standard Ka-Band pointing frequency).

```
source = '0319+4130' # replace with your calibrator
Break("Ask the operator to switch to Ka-Band. Click yes when Ka-Band is in place.")
SetValues("ScanCoordinator", {"receiver": "Rcvr26_40"})
SetValues("L01", {"restFrequency_A": 32000})

AutoPeakFocus(source)
```

Step 2.2 provides  $\text{focus}_{\text{Argus}}$  at your target frequency and Step 2.3 provides  $\text{focus}_{\text{Ka, primary}}$ . Using those two numbers we can calculate the focus offset,  $\Delta_{\text{focus}}$ , as  $\Delta_{\text{focus}} = \text{focus}_{\text{Argus}} - \text{focus}_{\text{Ka, primary}}$ . Determining the focus offset with a single decimal point is sufficient.

### Example

$\text{focus}_{\text{Argus}} = -4 \text{ mm}$

$\text{focus}_{\text{Ka, primary}} = -1 \text{ mm}$

$\Delta_{\text{focus}} = -4 \text{ mm} - (-1 \text{ mm}) = -3 \text{ mm}$

### 2.4.1.3 3. Secondary Calibrator

Go to your secondary calibrator (nearby your science target, i.e. within ~30 deg in Az and ~10-20 deg in El, the closer the better to minimize slew times).

## 3.1 Pointing and Focus Correction

Run **AutoPeakFocus** using *Ka-Band*, this script will at the end automatically apply your determined  $\Delta_{\text{focus}}$ . If you have the run the script more than once, please make sure you comment out line 14 `SetValues("Antenna", {"local_focus_correction,Y": new_lfc})` before re-issuing the script, to avoid adjusting the focus multiple times.

```
1 Catalog('/home/astro-util/astridcats/kband_pointing.cat')
2
3 source = '0336+3218' # replace with your calibrator
4 freq_argus = 93173.0 # replace with your target frequency in MHz
5 delta_focus = 4.0 # replace with your focus offset value in mm
6
7
8 Break("Ask the operator to switch to Ka-Band. Click yes when Ka-Band is in place.")
9 SetValues("ScanCoordinator", {"receiver": "Rcvr26_40"})
10 SetValues("L01", {"restFrequency_A": 32000})
11 # adjust focus for Ka-Band
12 lfc = float(GetValue("Antenna", "local_focus_correction,Y"))
13 new_lfc = lfc - delta_focus
14 SetValues("Antenna", {"local_focus_correction,Y": new_lfc})
15
16 AutoPeakFocus(source)
```

(continues on next page)

(continued from previous page)

```

17 Break("Ask the operator to switch back to Argus. Click yes when Argus is in place.")
18 SetValues("ScanCoordinator", {"receiver": "RcvrArray75_115"})
19 SetValues("L01", {"restFrequency_A": freq_argus})
20
21
22 # adjust the focus for next Argus observations
23 lfc = float(GetValue("Antenna", "local_focus_correction,Y"))
24 new_lfc = lfc + delta_focus
25 SetValues("Antenna", {"local_focus_correction,Y": new_lfc})
26
27 print ""
28 Comment("-----")
29 Comment("LFC-Y changed from %f to %f    (shift of %f mm)" % (float(lfc), float(new_lfc),
↪ float(delta_focus)))
30 Comment("-----")
31 print ""

```

### 3.2 Add the focus correction factor manually

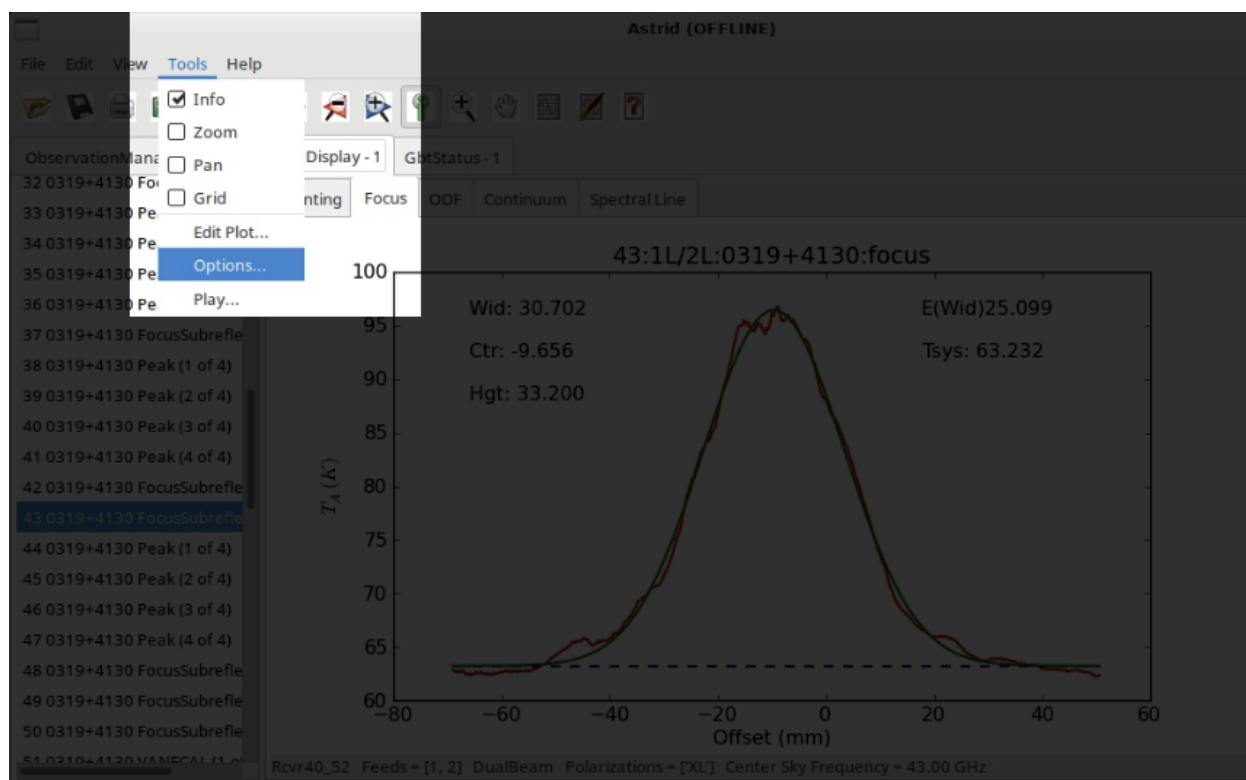
If you don't use the code snippet provided in *3.1 Pointing and Focus Correction* to add the focus offset,  $\Delta_{\text{focus}}$  you calculated in step 2.3 to the determined focus correction, LFC, you will need to do it manually.

#### Example

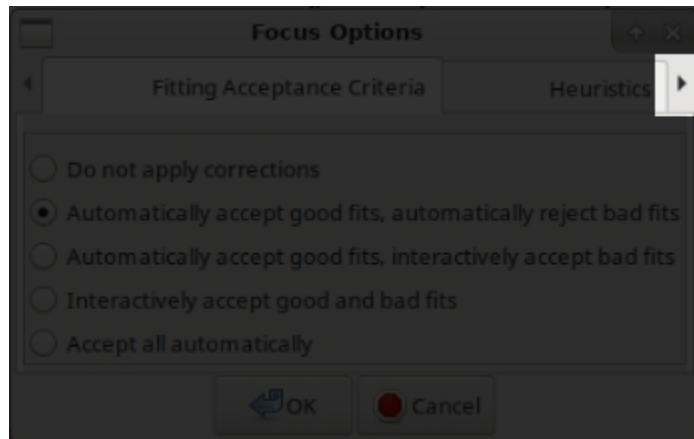
$\text{focus}_{\text{Ka, secondary}} = +2 \text{ mm}$

$\text{LFC} = \text{focus}_{\text{Ka, secondary}} + \Delta_{\text{focus}} = +2 \text{ mm} + (-3 \text{ mm}) = -1 \text{ mm}$

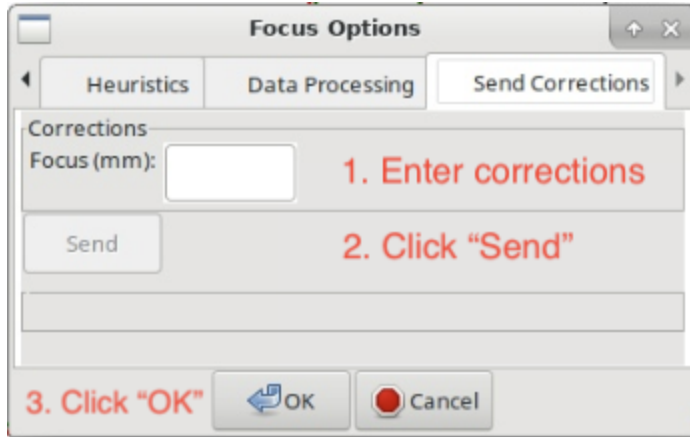
To add this LFC value in the system, you have to be in the “DataDisplay” Tab in Astrid, and there in the subtab “Focus”. Then click “Tools” in the top left menu of the Astrid applications and choose “Options”.



A pop-up window “Focus Options” will open. Click the right arrow a few times, to switch to the last tab “Send Corrections”.



In the “Send Corrections” tab enter your determined LFC, click the send button and then click OK after you have confirmed that the corrections have been send to the telescope, e.g. by checking the LFC value in the CLEO Status page.



Alternatively you can ask the Operator to enter the LFC value for you. You will also want to adjust the focus when you switch from Argus to Ka before executing peak/focus calibration.

#### 2.4.1.4 4. Science Target

Go to your science target, configure Argus for your science observations, check the YIG power. Run a vanecal and execute your observations. We recommend to verify your configuration/setup by executing short track or OnOff observation on a known source at the beginning of each observing run.

#### 2.4.1.5 5. Subsequent observing procedure

Alternate between observations of your science target (*Step 4 above*) and observations of the secondary calibrator (*Step 3 above*) every 30-40 min, depending on weather conditions.

Remember to re-run an AutoOOF every 3-6 hours, depending on weather (and more often if you've been scheduled at or shortly after sunset while the temperature was still dropping).

### 2.4.2 MUSTANG-2 How-To Guides

Practical step-by-step guides to help you using MUSTANG-2.

#### Proposal Information

Instructions and information for proposers

[Proposal Information](#)

#### Setup - Tuning and Biasing

How to setup MUSTANG-2 for observations

[Setup](#)

#### Observing

How to setup MUSTANG-2 for observations

[Observe](#)

#### Data

Information on data products and data processing

## Data products

### 2.4.2.1 How to Propose with MUSTANG-2

General properties (such as FOV, resolution, bandpass information) can be found on [the MUSTANG-2 overview page](#).

If you have general questions about feasibility or initial questions feel free to reach to the members listed under the Contact section on [MUSTANG-2 instrument team](#) webpage. However, we ask that you please do this in a reasonable amount of time before a proposal deadline as the team will be busy with many proposals right before the deadline.

## Proposal Requirements

### Team Approval

All MUSTANG-2 proposals are shared-risk and must be approved by the [MUSTANG-2 instrument team](#). Furthermore, the entire MUSTANG-2 instrument team **must** be included as co-investigators on the proposal.

In order to get your proposal approved by the MUSTANG-2 team, contact one or all of the members listed under the Contact section on [MUSTANG-2 instrument team](#) webpage with a draft of your proposal at least a week in advance of the proposal deadline. The MUSTANG-2 team will principally focus on the technical feasibility of a proposal and make suggestions accordingly. The technical justification on a proposal should reference publicly available mapping speeds (e.g. from the MUSTANG-2 mapping [webpage](#) and/or MUSTANG-2 mapping speeds memo). The GBT sensitivity calculator does not currently incorporate MUSTANG-2 mapping speeds.

### Overhead observing constraints

The overhead for MUSTANG-2 is dominated by initial setup and calibration. We generally recommend a minimum session length of 2 hours. Allowing for weather and calibration and observing overheads, observers should conservatively allow an observing efficiency of 50% (or 100% overheads relative to on-target time). Thus in the end to account for setup and calibration time to get thier final time request the user should multiply thier on-target time by a factor of 2.

### Source visibility considerations

Daytime observing at 90 GHz is currently not advised. The changing solar illumination gives rise to thermal distortions in the telescope structure which make calibrating 90 GHz data extremely difficult. Useful 3mm observations are currently only possible between 3h after sunset and a half hour past sunrise. Further cooler temperatures are required for observing at 90 GHz thus the high-frequency observing season for MUSTANG-2 is typically ~October - late April/early May. Thus your target must be visible to the GBT 3h after sunset and a half hour past sunrise in ~October - late April/early May.

## Other things of note

### Data and Observing

Though the entire MUSTANG-2 instrument team will be involved in the proposal process, conversely, the MUSTANG-2 team will reduce the data and provide appropriate data products (principally a calibrated map, transfer function, and beam characterization) to the proposal team (see *the list of possible data products*). End-to-end data reduction is currently fairly involved. We will work to provide documentation on data processing and hope to eventually allow proposers to process their own data.

MUSTANG-2 instrument team also asks that the PI and team get trained to observe with MUSTANG-2 and observe for their project whenever possible.

### Calculating sensitivities

MUSTANG-2 is NOT included in the GBT sensitivity calculator thus sensitivities are typically calculated using simulations or previous observations.

For galaxy clusters, one can run simulations or use the tables in this memo to compute the expected Compton Y or peak and corresponding required sensitivity. Then reference the table on the *mapping webpage* and use the following relation to compute required integration time.

**As a general rule one can use the relationship between integration time (t) and sensitivity ( $\sigma$ ) where  $t \propto 1/\sigma^2$  and the values in the table above to calculate the required integration time or desired sensitivity. For example, if one would like to calculate the required integration time corresponding to a desired sensitivity:**

- From the radiometer equation  $t \propto 1/\sigma^2$
- set up in a proportional relationship  $t_2/t_1 \propto (\sigma_1/\sigma_2)^2$  where  $t_2$  is the required integration time that you are solving for,  $t_1$  is 1 hour,  $\sigma_1$  is the sensitivity corresponding to the map size from the table on the *mapping webpage*, and  $\sigma_2$  is the desired sensitivity that you have calculated
- $t_2 \propto (\sigma_1/\sigma_2)^2 \times t_1$  and thus  $t_2$  is your integration time

#### 2.4.2.2 How to setup MUSTANG-2

### Tuning

Detailed M2 instrument team instructions on tuning and biasing that can be used as a reference can be found here: <https://safe.nrao.edu/wiki/bin/view/GB/Pennarray/OnGbtOps>.

---

### What is Tuning?

Tuning is MUSTANG's rough equivalent to balancing. It has multiple channels of data and tuning is trying to ensure that the attenuation levels are all roughly equivalent. You do not need to be anywhere specific on the sky nor does MUSTANG need to be in the GBT turret for this step to be done.

Mustang has ~256 channels, ~215 feed horns, of which about ~211 are populated. It has 4 roaches, which are a type of Field Programmable Gate Array (FPGA, which is essentially a specialized CPU for data acquisition and manipulation). These 4 roaches process 64 channels each, adding up to the 256 channels. However, since not all the feed horns are populated, some of the channels in these roaches are already unused.

The IQ plots and many graphs for each roach represent all the different channels for each roach. Since some of those channels are unused, it's not abnormal to see some channels that don't look right. As long as the majority look okay,

then things are working fine.

---

**Note:** All tuning steps should take place in time before your allotted observing time. That is, before you are given control of the GBT. Allow at least one hour of time before your allotted observing time to allow for time for things to go. You can do tuning up to several hours before the allotted observing time.

---

## 1. Preparation

1. Start an observing VNC
2. Ask the operator to put you in the gateway for M2

## 2. Run startup + tuning script

### 1. Check that the array is cool and not cycling

Go the CLEO M2 housekeeping tab and check that the array temperature is ~400mK. Also check that the “Cycle State” on the left side of the window is IDLE and not in stages 1-5. If it is in a “stage” this means that it is cycling. If either the temperature of the array is warm or it is cycling, the array is not ready and/or not cool enough for tuning yet.

### 2. Change directories to where the startup + tuning script is

```
cd /users/penarray/Public
```

### 3. Run script

```
./startMUSTANG.bash projectCode_sessionNumber
```

where `projectCode` is your proposal number (e.g. AGBT18A\_014) and `sessionNumber` is your current (AstrID) session number (e.g. 01). The session number is **not** the source code from the DSS e-mail. It is very important to get this right or data reduction will fail to pick up the tuning which in turn affects focusing. Check <https://safe.nrao.edu/wiki/bin/view/GB/Pennarray/NewRunNotes> to see how many sessions have already been observed for this project.

**Note:** It’s important when typing out your session number that the session number has two digits. For example, if this is your 9th session, type out AGBTProjectID\_09 NOT AGBTProjectID\_9. If you do the latter option then you will not be able to take data correctly.

If you make a mistake and get the session number wrong, after tuning is done it can be corrected by creating soft links in the same way as you do when changing projects. If you’ve accidentally mislabeled your Project ID and session number, ssh to egret using lmonctrl and type this command:

```
cd /home/gbtlogs/Rcvr_MBA1_5tuning/
```

In this directory you should be able to see your projectID and session number. Simply rename it by typing:

```
mv WRONG_ProjectID_sessionNumber CORRECT_ProjectID_sessionNumber
```

This will fix any issues with data acquisition.

### 4. Follow process of script



1. The script will ask you if you really want to do this: Use the arrow keys to select yes.
2. Then it will ask you to enter the lmonctrl password on egret

---

**Hint:** Ask the M2 instrument team if you don't know it.

---

3. Then the script will do the following:

1. Log into egret and restart the manager
2. Telnet into the iboot bar and turn on the roaches, function generator, and HEMTs
3. Start one xterm as lmonctrl@egret and one gnome-terminal with 5 tabs running ipython sessions: one general and 4 labeled as M1-4 which tune each roach. Note that sometimes gnome-terminal fails in which case it will bring up separate x-terms.
4. During tuning it will ssh into each roach every 5 seconds
5. After tuning has finished it will bring up the tuning plots
6. Then it will set the manager into observing mode and check if data are flowing - if not it will attempt to fix this.

1. Troubleshooting - Roach Not Awake

If one of the roaches is not awake you may see an error in the tuning process like this:

```

IPython: monctrl/new_umux (on egret)
File Edit View Search Terminal Tabs Help
IPython: monctrl... x IPython: monctrl... x IPython: monctrl... x IPython: monctrl... x IPython: monctrl... x
help -> Python's own help system.
object? -> Details about 'object', use 'object??' for extra details.
('rootdir=', '/export/home/egret/monctrl/MUSTANG/mustangr1-1/')
('saveDir=', '/home/gbtlogs/Rcvr_MBA1_5tuning/TGBT23B_608_01/mustangr1-1/')
('proj=', 'TGBT23B_608_01')
('tuneKwds=', {'atten_out': [5.0, 10.0, 15.0], 'df': 50000.0, 'plot2screen': 'eog', 'equalizePower': False, 'loSpan': 10
00000.0})
/export/home/egret/monctrl/new_umux/umuxlib.py:411: VisibleDeprecationWarning: Reading unicode strings without specifying
the encoding argument is deprecated. Set the encoding, use None for the system default.
  out=np.genfromtxt(StringIO(StringIO(ipstr)),dtype=None,names=['roachNum','roachIP','newIFcard'],delimiter=' ')
mustangr1-1 : connecting to roach mustangr1-1.gbt.nrao.edu
mustangr1-1 : Failed to connect to roach mustangr1-1.gbt.nrao.edu
mustangr1-1 : Roach is not connected, run connect()
mustangr1-1 : Roach is not connected, run connect()
startTones:Roach is not connected - no action taken
mustangr1-1 : Starting to tune Roach mustangr1-1.gbt.nrao.edu
mustangr1-1 : created new directory /home/gbtlogs/Rcvr_MBA1_5tuning/TGBT23B_608_01/mustangr1-1//2023_10_11_07:44/ for
saving tuning results.
mustangr1-1 : Loaded frequencies from /export/home/egret/monctrl/MUSTANG/mustangr1-1/default_freqs.txt
mustangr1-1 : Roach is not connected, run connect()
mustangr1-1 : Roach is not connected, run connect()
mustangr1-1 : setting iq centers in firmware to zero.
mustangr1-1 : Roach is not connected, run connect()
mustangr1-1 : Roach is not connected, run connect()
mustangr1-1 : Adjusting atten_in to bring ADC counts to a std of 0.2 of 12 bit ADC range =819.2
mustangr1-1 : Roach is not connected, run connect()
Error occurred in roach setup - details follow:
Traceback (most recent call last):
  File "/export/home/egret/monctrl/new_umux/umuxlib.py", line 46, in startDAQ
    um.tunePhase(**tuneKwds)
  File "/export/home/egret/monctrl/new_umux/umuxlib.py", line 1517, in tunePhase
    dPhi,stdAmp,stdPhase = self._phaseResponse(atten_in,atten_out,loSpan,df,default_freqs)
  File "/export/home/egret/monctrl/new_umux/umuxlib.py", line 1439, in _phaseResponse
    self.autoAdcRange()
  File "/export/home/egret/monctrl/new_umux/umuxlib.py", line 921, in autoAdcRange
    i,q = self.getSnapAdc()
TypeError: 'NoneType' object is not iterable
Feel free to finish tuning by hand, or re-run startDAQ
In [1]:

```

with the main error being “Roach is not connected.”

If one of the roaches will not wake up, first ssh to it:

```
ssh root@mustangr1-#
```

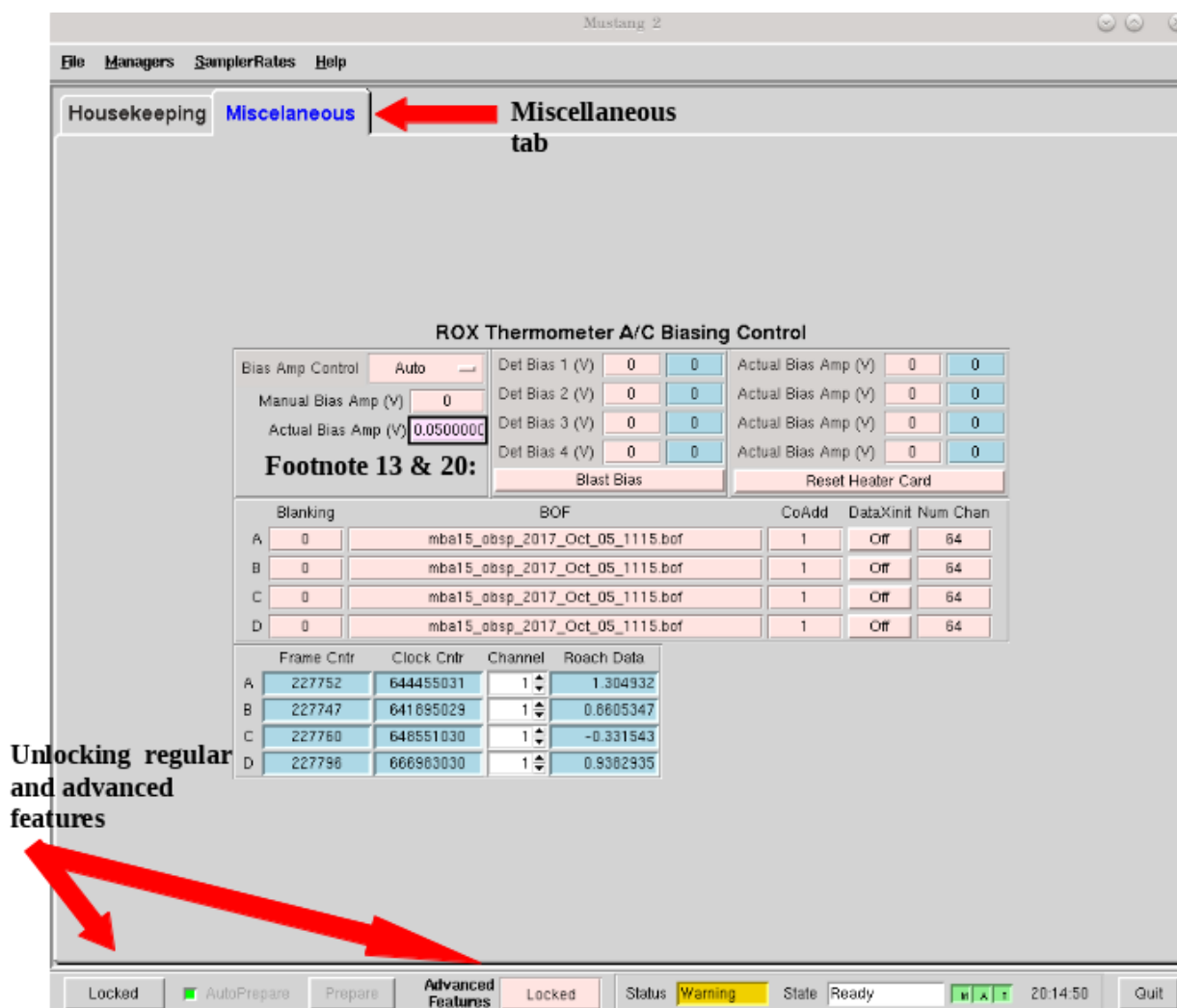
where # is the roach number. It may take a while for the ssh to go through (several minutes). Once the ssh goes through, in the ipython session for that roach you can redo the tuning by doing the `um1=startDAQ()` command. Simply type in `um1=` and do the up arrow to find the command; you are looking for a command that looks like `um1=startDAQ(rootdir=rootdir, SaveDir=SaveDir, project=proj, doVNA=False, logLevel="DEBUG", tuneKwds=tuneKwds,...)`. If that does not work, you will have to do restart the tuning process using the same project code and session number.

## 2. Check the IQ, Flux Ramp, and Phase Response plots output by the script.

See <https://safe.nrao.edu/wiki/bin/view/GB/Pennarray/TuningResults> for explanations and examples of good and bad tuning results.

## 3. Check that data is flowing

Go to the Mustang Manager in CLEO. Click the miscellaneous tab, and click the “Locked” on the bottom left of the window to unlock the regular features, then also unlock advanced features by clicking the “Locked” next to Advanced Features.



Click through the channels and look for:

- The “Frame Cntr” numbers should be changing and not be really low or 0.
- The “Roach Data” numbers changing.

- The Frame and Clock Cntr columns next to the Channel and Roach data should be similar across the 4 roaches (if they finished tuning at the same time).

If there is no data flowing in one or more roaches, you have a few potential solutions:

- You can try resending the channel numbers by going to “Num Chan” -> enter 64 -> press enter.
- Or turn “DataXinit” off then on.
- SSH to the problematic roach(es), e.g. “ssh root@mustangr1-1”.
- If these steps do not solve the problem, you may need to either restart the manager, or worst case, turn off data streaming (zero biases if you notice a problem after biasing the detectors) and power cycle the roaches (in the ibootbar). If after restarting the manager, the problem persists, restart the roaches (in ibootbar). If you restart the ROACHes, you will need to redo the tuning steps.

**Note:** Be sure to lock the Mustang Manager back when you are done to prevent any accidental miss-clicks. You press the same “Locked” buttons as you did in the beginning of this step, only now they will appear as “unlocked” until you click them again.

## Biasing

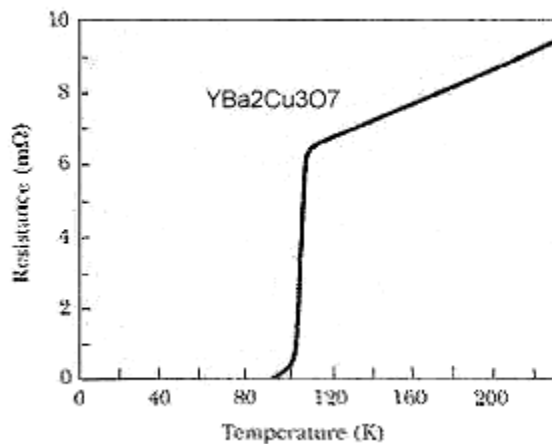
### What is Biasing?

Biasing is finding the voltage that puts the TES detectors on the transition from superconducting to normal, the point at which the resistance of the superconductor is changing with temperature (and makes a good thermometer) for measuring the power landing on the bolometer.

### More background information

The MUSTANG-2 receiver is a continuum receiver that uses a bolometric thermometer to make its measurements. Essentially, it is a highly sensitive thermometer with a filter for its bandwidth. Therefore, any photons in the bandwidth hitting the receiver will raise the temperature slightly.

It is able to be this sensitive by taking advantage of the science behind superconductors. This can be explained using the graph below:



This graph is for a specific superconductor, but the concept is the same, even if the exact temperature and resistance is different. Don't pay attention to the numbers, but rather the trends.

As you can see, the material is only superconducting at lower temperatures. Once it gets hot enough, it becomes a regular resistor, with higher resistance with higher temperatures. What the MUSTANG-2 receiver takes advantage of is the portion of the graph called the "transition edge," the area in between the material being a regular resistor and being a superconductor. Here, the resistance changes very rapidly with even a slight change in temperature.

What biasing does, is ensure that each roach, when observing blank sky, is set in such a way that the maximum number of channels are placed at this transition edge, in order to ensure maximum sensitivity of the receiver. You will be seeing graphs for each channel, and the point which the AI is choosing is what it believes to be the transition edge of that graph.

Because we are only able to choose one setting for each roach, hence the attempt to simply maximize the effectiveness of all the channels, usually at the expense of certain channels in that roach.

---

## 1. Run biasing

All previous steps (tuning) can take place in the 1-hour prep before your allotted observing time, i.e. before you are given control of the GBT. However, **biasing must be done on blank sky**, therefore **you must have control of the telescope** for this and all subsequent steps (else just seeing the subreflector/ground spill over). You must be on your own account, logged onto titania or ariel, and also have permission to be in the gateway from the operator.

### 1. Navigate to the proper directory to run biasing

```
cd /home/gbtlogs/Rcvr_MBA1_5tuning/detbias
```

### 2. Configure the appropriate bash profiles

```
source /home/gbt/gbt.bash
source /home/gbt/sparrow/sparrow.bash
```

### 3. Run the bias script

```
python new_detbiasV3.py projectCode_sessionNumber
```

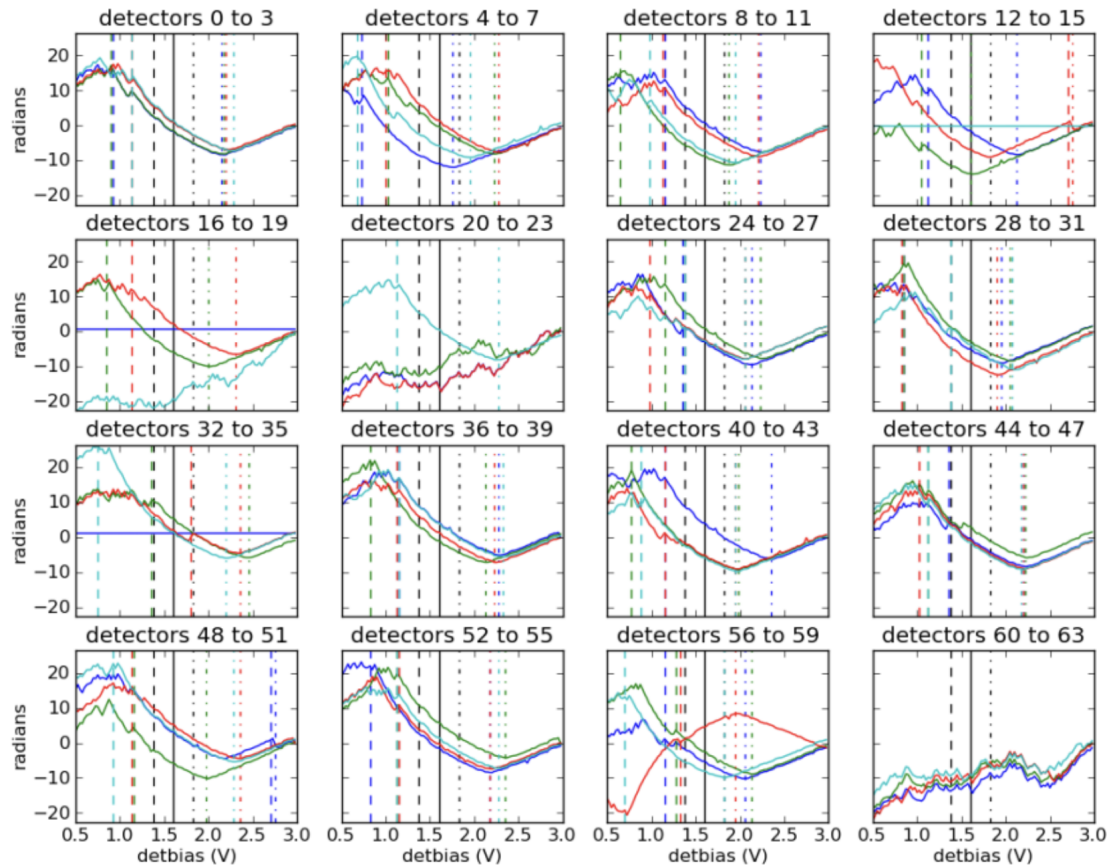
where projectCode\_sessionNumber is e.g. AGBT18A\_014\_01.

### 4. Inspect the bias plots

They are typically referred to as det bias files, as det bias is a shortened way of referring to the determined bias. You will see the speed of data coming going quickly and 'Det Bias' (in Misc tab) changing. After waiting a while (5 min or more), you will get a set of graphs.

A good set of biases will look like this:

Detector bias file for roach D taken 09/02/2019 at 20:37:36 normal=dot-dash,sc=dash



You will get 4 sets of graphs like this, one for each roach. This one is for roach D, or roach 4, as shown in the title. See <https://safe.nrao.edu/wiki/bin/view/GB/Pennarray/TuningResults> for some examples of bad detbias plots.

The solid black lines indicate the AI-decided detbias for each channel. It's okay to see some of the lines reversed in direction (like in detectors 56 to 59 in this example) however something is wrong with that detector when it doesn't have that general shape (such as in detector 20-23, or 60-63). Having a couple bad detectors isn't unheard of, it's more bothersome if a large percentage of detectors don't look right.

##### 5. Close plots

when you're done inspecting them

##### 6. Send Bias values to roaches

In the terminal, enter Y to send bias values to roaches and anything else to ignore calculated values.

##### 7. Note calculated values

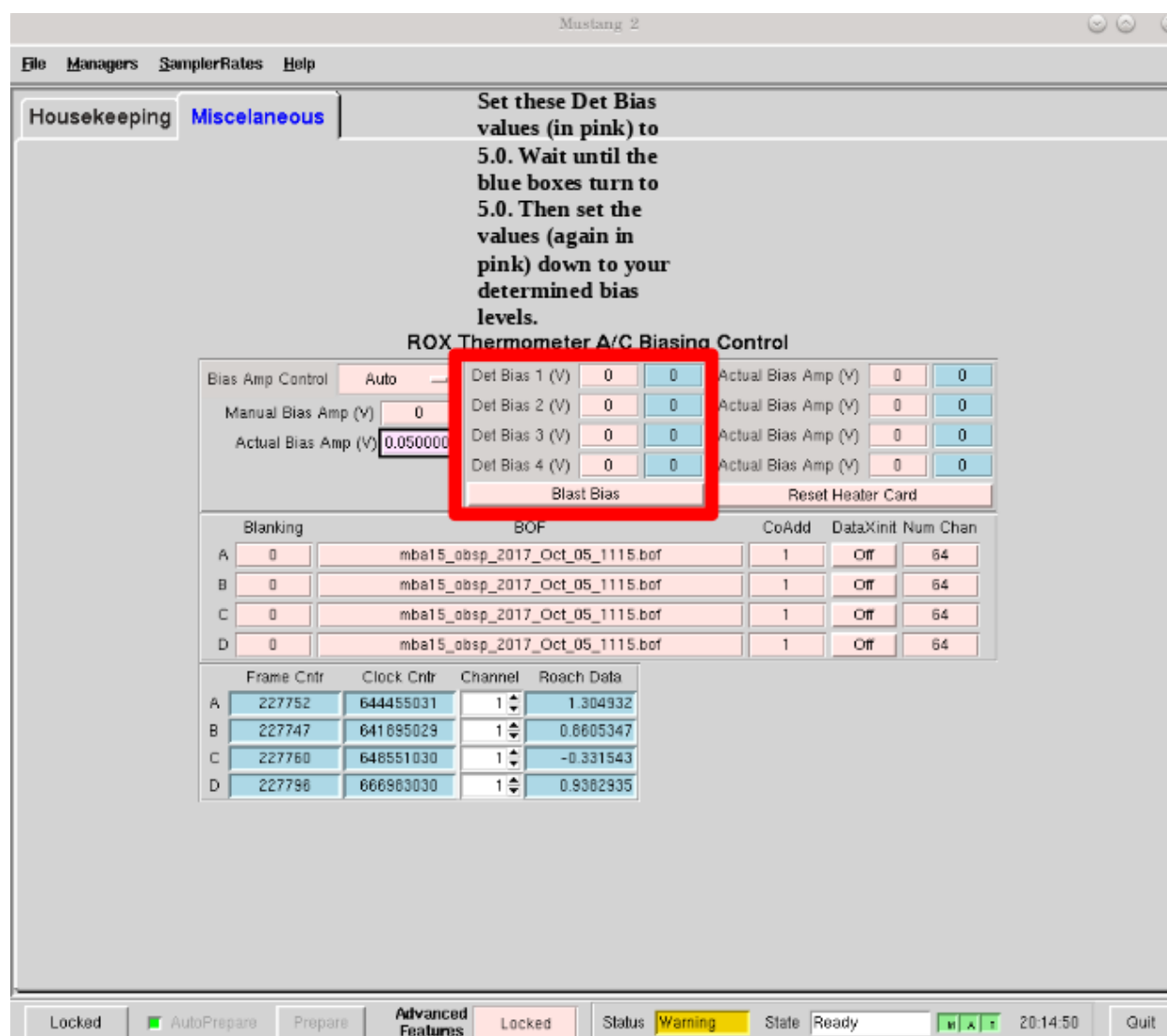
Record the calculated values are (good practice to put them in the log) by checking the Bias values in Misc! Then if the manager crashes, you know what values to re-enter.

## 2. Short session: Enter biases manually

If you have a short observing session, you can manually enter the biases to save some time. To do this

- unlock the manager
- **roach-by-roach**
  - set the DetBias to 5.0
  - press enter
  - wait until the blue box shows a DetBias of 5.0
  - then enter 1.2 into the DetBias

**Important:** Check with an experienced M2 team member as to what the current reliable Det Bias value is.



If the manager crashed and you need to re-enter the values that were previously calculated, follow the same process but put in your recorded values.

## Issues with Manager? Restart the Manager

If you are having issues with the manager or it crashed, you will need to restart the M2 manager. To do this, do the following:

1. Ask the operator to restart the MUSTANG manager using TaskMaster, even if you've been told how to do this yourself. Restarting machines through TaskMaster is a responsibility that is supposed to only be held by the operator.
2. **When the operator has told you that they have restarted the M2 manager:**
  1. Go to your Cleo Mustang Manager screen
  2. In the drop down menu go to Managers→Off and then again to click Managers->On to turn the manager off and back on.
  3. Re-check the daily cycle to make sure that it is turned off.
3. Restarting the manager *before* biasing: You're done!
4. **Restarting the manager *after* biasing:**
  1. Re-check that the det-biases are what you expected them to be.
  2. Check that the dataXinit buttons are on.

### 2.4.2.3 How to observe with MUSTANG-2

#### 1. Project Preparation

##### 1.1 Calibrator and Source Preparation

You are expected to have your pointing calibrators, flux calibrators, OOF sources, and science sources planned out **at least a few hours before** the time of your observation. You can use CLEO's Scheduler and Skyview to do this.

- **Flux calibrators:**

- **Find those that are closest to your source. You can use planets or any of the ALMA grid cal listed in the following catalog:**

```
/users/penarray/Public/Catalogs/alma_gridcal.cat
```

You will need to observe at least one of these during your observing session to ensure flux calibration. Preferably 2-3, but if you are ok with a 10-20% error in your flux measurement 1 calibrator is ok.

- **OOF sources**

- It is efficient to use the flux calibrators as your first OOF source of the night. For OOF sources, a general guide is that you want a bright source that is  $> 1$  Jy and  $25 < \text{elevation} < 60$ . Planets, if available, are often preferred as flux calibrators (especially Mars, Uranus, or Neptune), though potentially only Uranus is also a preferred OOF target. A more intricate way of thinking about an OOF source is to consider the elevation of your science target: if it will be “low” (average observing elevation is  $\sim 35$  or less), or “high” (average observing elevation  $\sim 60$  or higher) then one would prefer to OOF on a source with a similar elevation. If the science target is in between, then the OOF elevation will be less important.

- **Pointing calibrators**

- **You can find suitable calibrators using CLEO's Scheduler & Skyview**

\* Click Catalog... in the upper right-hand corner

- \* Click Add/Select/DeSelect Catalogs ...
- \* Select mustang\_pointing
- \* Click Apply
- **The goal is to find a calibrator that is 10-15 deg from your target and > 0.5 Jy (though if have good weather a better choice is something close that is 0.1 Jy). To find a source that is > 0.5 Jy fo the following in CLEO's Scheduler & SkyView:**
  - \* Go to the box in the right-hand corner that says Source Intensity Range and in the Min box put 0.5
  - \* Hit enter
  - \* Load your science source catalog
  - \* Enter the time you will be observing in the UT Date and Time box
  - \* Find a source that is showing and is 10-15 deg from your target.
- **Science source catalog**  
You will either need to make one or a M2 team member has already made one for you.

## 1.2 Observing Scripts

You are expected to have your scripts ready **several hours before** the time of your observation. Ideally at the start of the observing semester.

Template observing scripts are located in: .. code:: bash

```
/users/penarray/Public/m2_template_scripts/
```

Read the README for instructions on editing these scripts once you have them in your project directory.

**If you are creating the scripts for the first time for your project, you will want to copy**

### 1. Standard calibration scripts

- 1\_m2setup
- 2\_m2oof
- 3\_m2quickDaisy00F
- 4\_m2quickDaisyPC

### 2. One of the science scripts labeled 5\_XXX

---

**Note:** The radius of the daisy will depend on your science - reach out to the M2 instrument team for guidance.

---

The scripts m2quickDaisy and skydip are extra but can be of use.



## 1.3 Observing Log

During observing, you are expected to edit the MUSTANG-2 observing run notes wiki and take notes of what's occurred throughout the night.

1. Create a new page and entry at the bottom of the [observing logs wiki](#) by clicking “Edit Wiki text”
2. Follow the naming convention of entry above <AGBTsemester\_project-code\_session-number>, e.g. AGBT18A\_014\_01
3. On the new log page you have created you can put in the text from a [log template](#). On the log template page, go to “Edit”, copy that text, and paste it into your new log (also in the “Edit” mode). You will have to get rid of some extra spaces.

---

**Note:** When you are actually recording information during observing you can be in either the “Edit Wiki Text” or “Edit” modes. But for some reason copying the formatting from the log template to your log has to be done in the “Edit” mode.

---

---

**Note:** If you don't have permissions yet to edit the wiki and are observing, you can take notes in a text document and email them to the MUSTANG-2 team afterwards to upload to the wiki for you.

---

## 2. Observing Preparation

### 2.1 Connect

Open and connect to *VNC session* or start an XFCE *FastX session* on *titania* or *ariel* via FastX.

### Connection Issues?

The internet at GBO can be intermittent at times. Specifically there are days that the internet goes down for 30-60 seconds at a time quite often. Are you having issues with FastX or your VNC being really laggy? Check [this status page](#) to see the status of the ssh gateways. See [these instructions](#) for FastX and VNC workarounds using Charlottesville to potentially better your connection.

### 2.2 AstrID

Open an AstrID session and navigate to your corresponding MUSTANG-2 project. The MUSTANG-2 instrument team should have already populated your Astrid area with appropriate scripts.

## 2.3 CLEO

The following are suggested CLEO windows to have open during observing:

- **Launch** → **Receivers** → **Mustang2**
  - **To monitor specific thermometers, click the gray box next to the titles:**
    - \* PT Fridge 1
    - \* PT Fridge 2
    - \* Array
    - \* HE4 Fridge 1 Charcoal
    - \* He4 Fridge 2 Charcoal
    - \* He3 Charcoal
    - \* He4 Fridge 1 Evap
    - \* He4 Fridge 2 Evap
    - \* He3 Evap.

These thermometers are of interest because they can indicate that things are wrong before they affect the array, or help diagnose what is wrong and how to fix it if the array temperature starts to go up. Sometimes a cryocycle gets started by accident - in which case if you are looking at the charcoal you can hit abort quickly and no damage is done. Other times a helium4 might run out and that can pull up the array temperature - not much you can do but often you can still collect some good data for a while.

- **Launch** → **Status**
- **Launch** → **Antenna**
- **Launch** → **Observer Tools** → **Scheduler & Skyview**
  - Click on Real-time mode
  - **Load in catalogs:**
    - \* mustang\_pointing
    - \* your science target catalogs
- **Launch** → **Observer Tools** → **Talk and Draw**

## 3. Observing Procedure

### 3.1. Communicate with operator

A few minutes before your observing start time (say 15 minutes, better 30 minutes), get on Talk & Draw, tell the operator who you are and what project you are observing for. Also ask who the operator is.

### 3.2. Fill AstrID info

In Astrid under ObservationManagement, go to the Run tab and fill in the Observer and Operator information.

### 3.3. Take control

Once the member of the M2 instrument team has finished biasing and the operator tells you are in the gateway/gives you the go ahead, in Astrid → File → Real time mode ... → Select work online with control of the telescope.

### 3.4. Configure

Run the 1\_m2setup script in Astrid.

### 3.5. OOF

1. Make sure that you have changed mySrc in 2\_m2oof and run the 2\_m2OOF script in Astrid.
2. For the first OOF of the night, you need to have calSeq=True so that a skydip is done as a part of the OOFing process. An OOF will take ~20 minutes to run.
3. **Check the OOF results in Astrid → DataDisplay → OOF and re-rerun if necessary.**  
For M2, we typically apply the z5 corrections. When the corrections are available, press the green button that reads After selecting the Zernike solution above, click this green button to send the solutions to the telescope.

---

**Note:** Sometimes OOF may time out and you will get a red screen if this happens. If this happens, re-OOF as this will restart the calculations of the solutions.

---

---

**Hint:** While your OOF is running, it is a good time to:

- **Write down the weather conditions from the GbtStatus tab in Astrid in the observing log**
    - Pyrgometer - if working
    - Temperature
    - Humidity
    - IR Cloud Cover
    - Wind Velocity.
  - *Start the m2gui* which is used to check M2 data while observing.
  - **In the m2gui check**
    - the skydip (once this has been executed through the OOF process)
    - that you can see the OOF images
-

### 3.6 Quick daisy on OOF source

1. **Run the 2\_m2quickDaisyOOF script on your OOF/calibrator source**  
It's best if you can make your OOF source and your calibrator source the same.
2. **Use the m2gui and determine**
  - beam shape (WidthA & WidthB)
  - peak of the source (Peak\_Height)
3. Record these values in your observing log
4. It's a good idea to check the time streams (see the [check time streams section](#) for instructions and examples.)

### 3.7 Quick daisy on pointing calibrator

1. Run the 3\_m2quickDaisyPC script on your pointing source.
2. **Use the m2gui again and determine**
  - beam shape (WidthA & WidthB)
  - peak of the source (Peak\_Height)
3. Record these values in your observing log

---

**Note:** During this initial data acquisition (and to some extent, throughout the night) check your Mustang2 CLEO screen, and make sure that the numbers in sections such as `Frame Cntr` and `Roach Data` are continuing to change with time (if so, the boxes will mostly be blue). However, if they stop (indicated when the boxes turn lavender) then the Mustang2 manager has crashed, and you'll need to [restart it](#).

---

### 3.8 Take science data

Take ~30 minutes of science data followed by a quick daisy on your pointing calibrator. Often this is accomplished by submitting several science scripts (e.g., `5_science_rX`) in Astrid. For example, often for cluster science each individual science scan is ~8-9 minutes in length. So if you are submitting individual beauty scans (which `5_science_rX` are), you can submit 4 of the science scripts in a row followed by your pointing calibrator scan.

It's a good idea to check the time streams (see the [check time streams section](#) for instructions and examples.)

---

**Note:** If you try to look at the science data in the m2gui, make sure you choose the “faint science” option under `source type`.

---

#### What is `science_r2p5` and `science_r3`?

`Science_r2p5` and `science_r3` are the science scans of the observation. The difference between the two is the radius of the scans in arcminutes (one is 2.5' and one is 3' respectively). If you only see science scans, unlabeled otherwise, then they are likely 3' in diameter. Legacy M2 scripts will have labels like `beauty_r3`.

---

### 3.9 Continue to take science data

1. Continue to do ~30 minutes of science data followed by a quick daisy on the pointing calibrator for the rest of the night.
2. Monitor the beam size (`WidthA` and `WidthB`) and the `Peak_Height` using the `m2gui` to determine if you need to OOF again.

### 3.10 When to OOF?

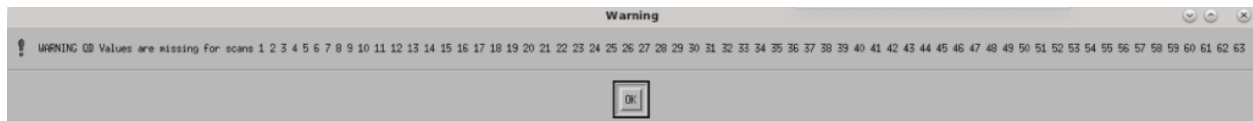
If the new `Peak_Height` is down by more than ~15%, or if `WidthA` and `WidthB` become very different from one another (indicating that the beam has become overly elliptical) you'll want to do an OOF.

#### Optional

If you don't have much observing time left, once the `PeakHeight` is down by more than 15%, instead of redoing the OOF scan, you can do another `m2QuickDaisy` on the pointing source to be sure that it is that low, and then do two more Beauty scans until the `PeakHeight` has gone down by another 15% (so a cumulative 30%).

### 3.11 Be aware - Issue with quadrant detector

In early 2023 it was discovered that over the past year or two the quadrant detector sometimes isn't working and doesn't write files to `/home/gbtdata/project_code_session/QuadrantDetector` as we expect. The GUI now will pop up a warning box (WARNING QD Values are missing for scans: ...) if it detects that the quadrant detector files are not being written.



If this happens during observing, press ok and ask the operator to restart the quadrant detector manager.

## 4. Checking data with the m2gui

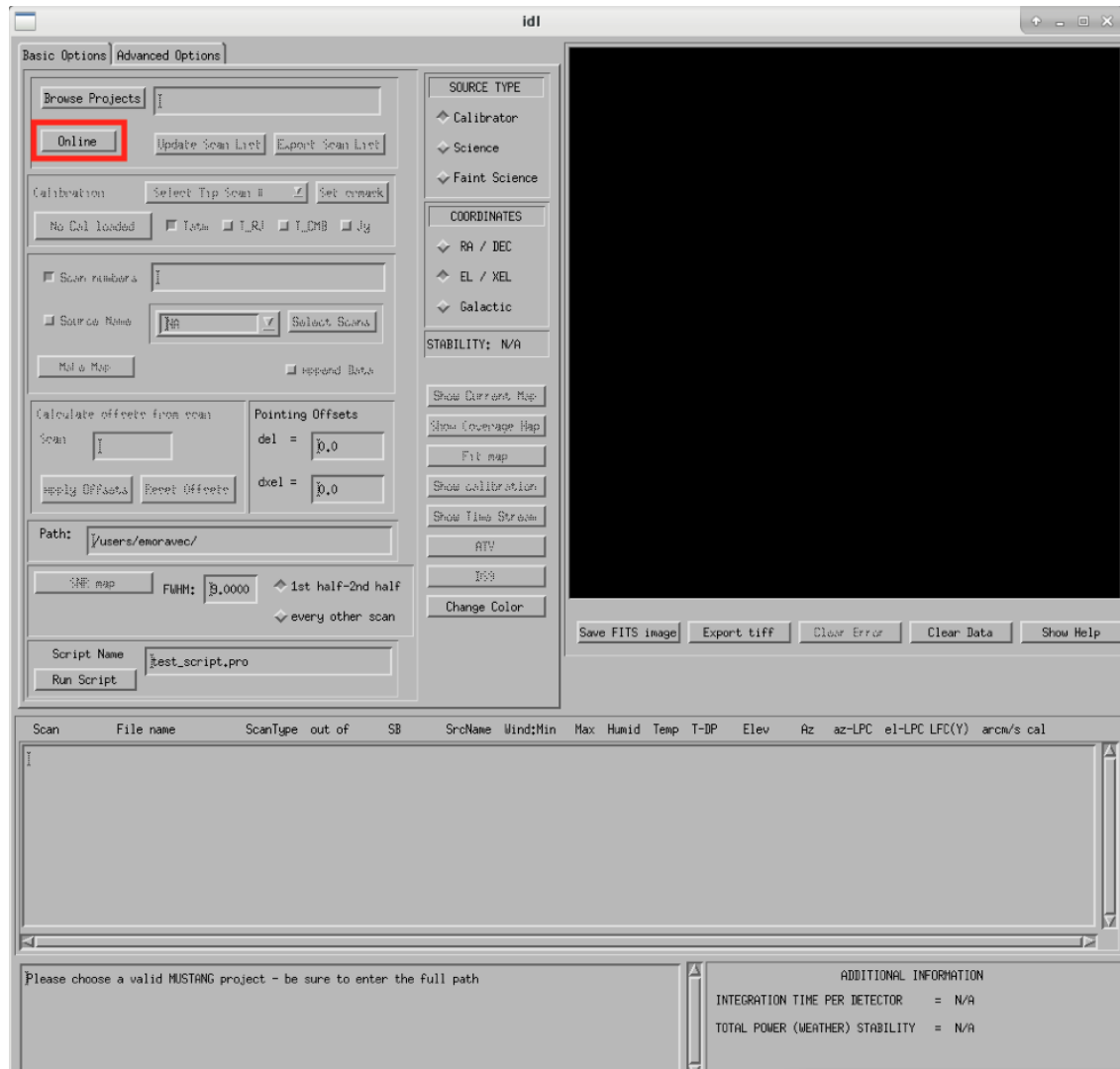
### 4.1 Start-up m2gui

To open up the `m2gui`, execute in a terminal (in a directory where you have write-access):

```
~penarray/Public/startm2idl
m2gui
```

After you have opened the `m2gui` follow these steps to check the tipping scan, monitor the beam shape (`width`, `widthA`, `widthB`) and peak of calibrators (`Peak_Height`), or to just check the data.

1. **Go online**  
Click the online button.



**Note:** If you want to open up a previous project that is not the current online project, click **Browse Projects**, find the project+session in the left hand column, and double click that folder to open it up.

## 4.2 Check Tipping Scan

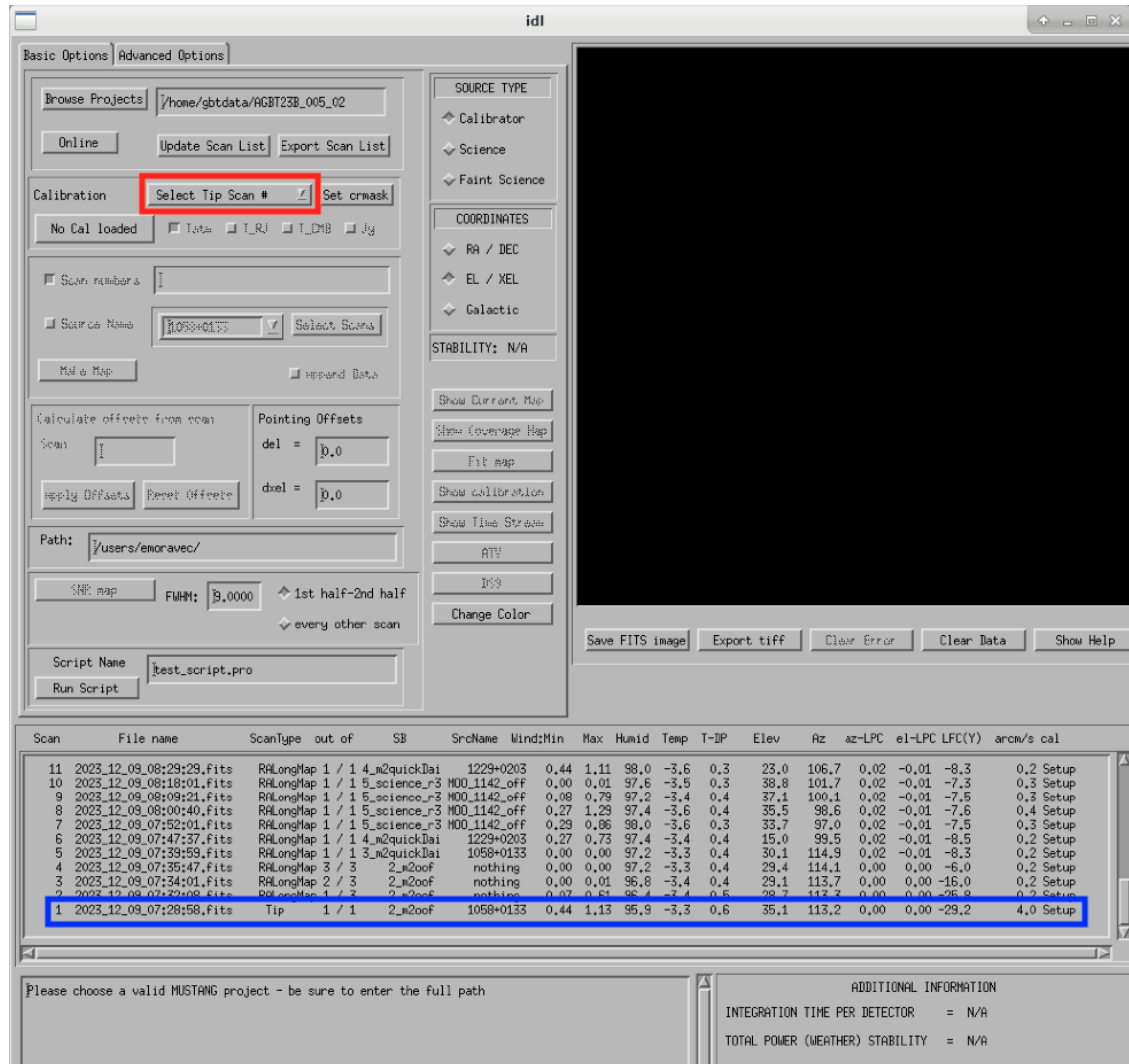
### What is a Skydip (Tipping Scan)?

What is a skydip? And what are the plots that we looking at? A skydip is a flat field. If you look at the detector bias curves some are inverted and even those with the same sign will have a different response to bias. We use the fact that the atmosphere is not transparent and has a  $\frac{1 - \exp^{-\tau}}{\cos(\text{elevation})}$  dependence. With a fair guess of the opacity  $\tau$ , you can do a fit on each detector to get them roughly Kelvin\_RJ. These calibrations are used to make maps of known sources and the results scaled to bring them to the correct amplitude.

#### 1. Select tipping scan

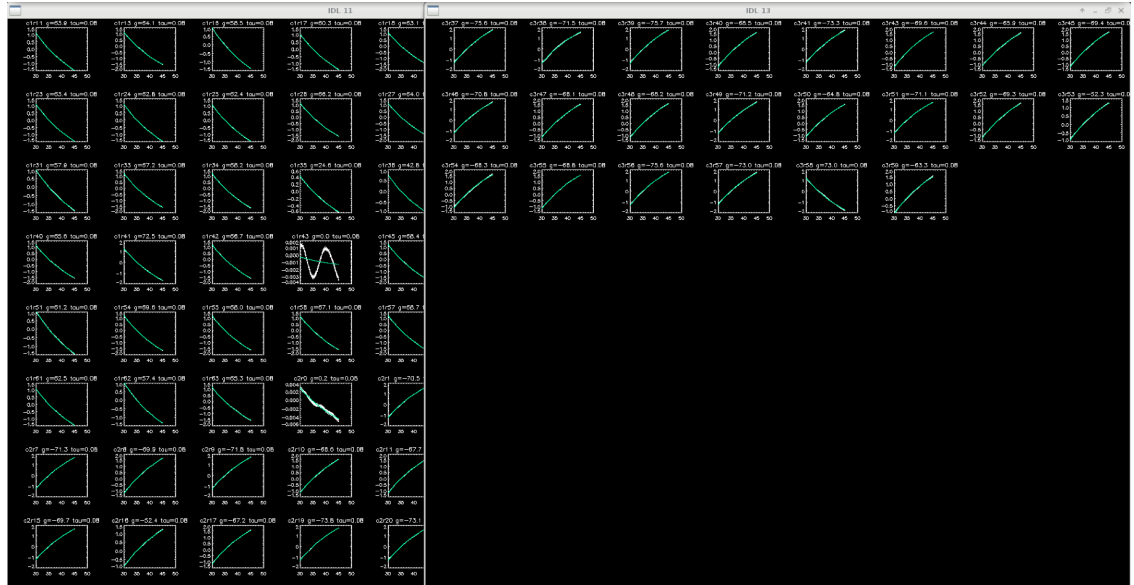
Under Calibration, click **Select Tip Scan** and choose the most recent scan number from the bottom

labeled Tip under scan type. At the beginning of the night, this should be from scan 1, before the 3 OOF scans (see below image - blue box).



## 2. Inspect plots

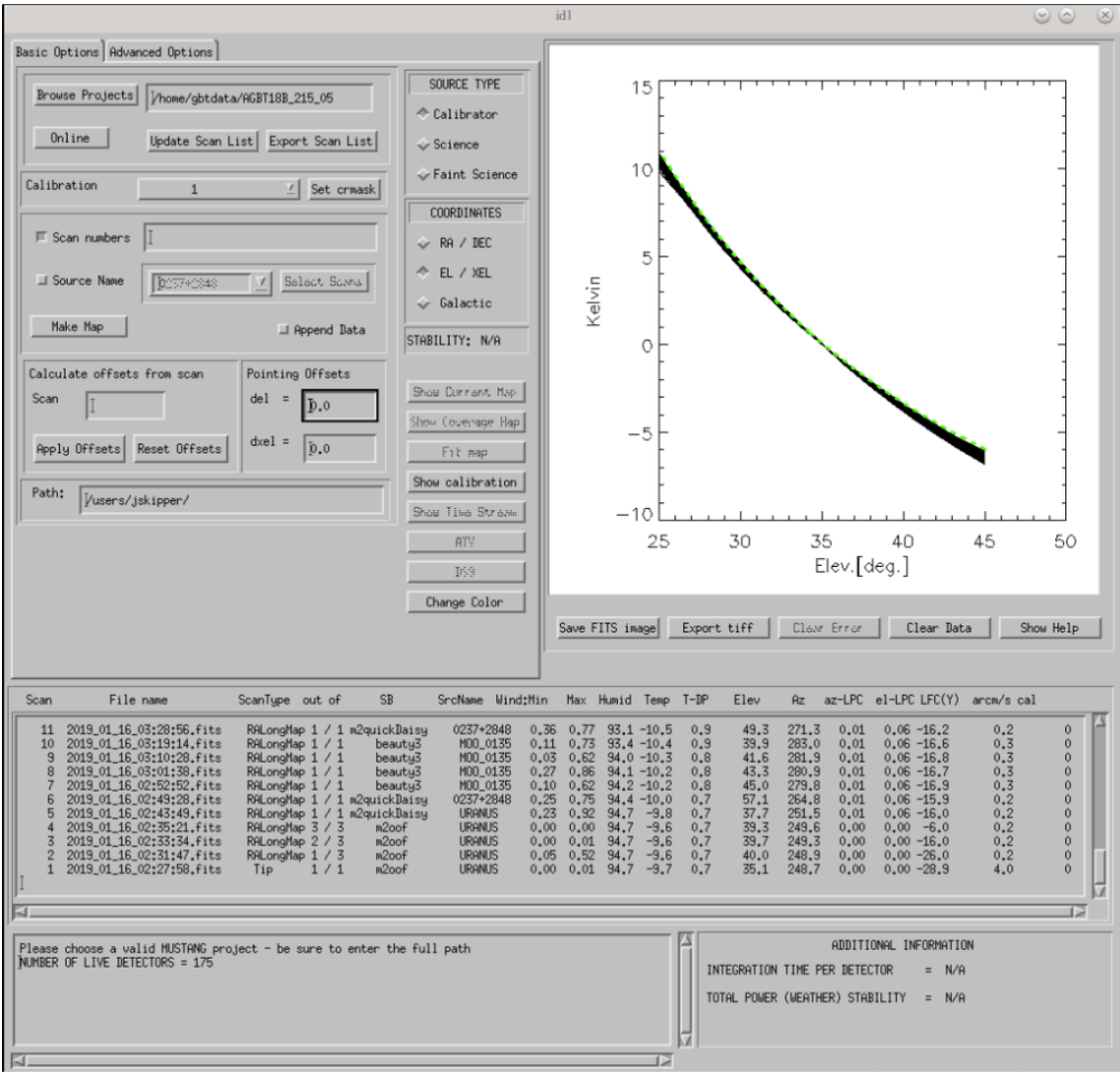
Many plots will pop up - one for each roach showing the results of the tipping scan for each roach. You can click out of these once they finish unless you are particularly curious about specific roaches. After these plots have been produced, you will see a graph to the right in the main gui window, showing the results of the tip scan - each roach is plotted in black with a fit in green. Check to make sure that it looks reasonable.



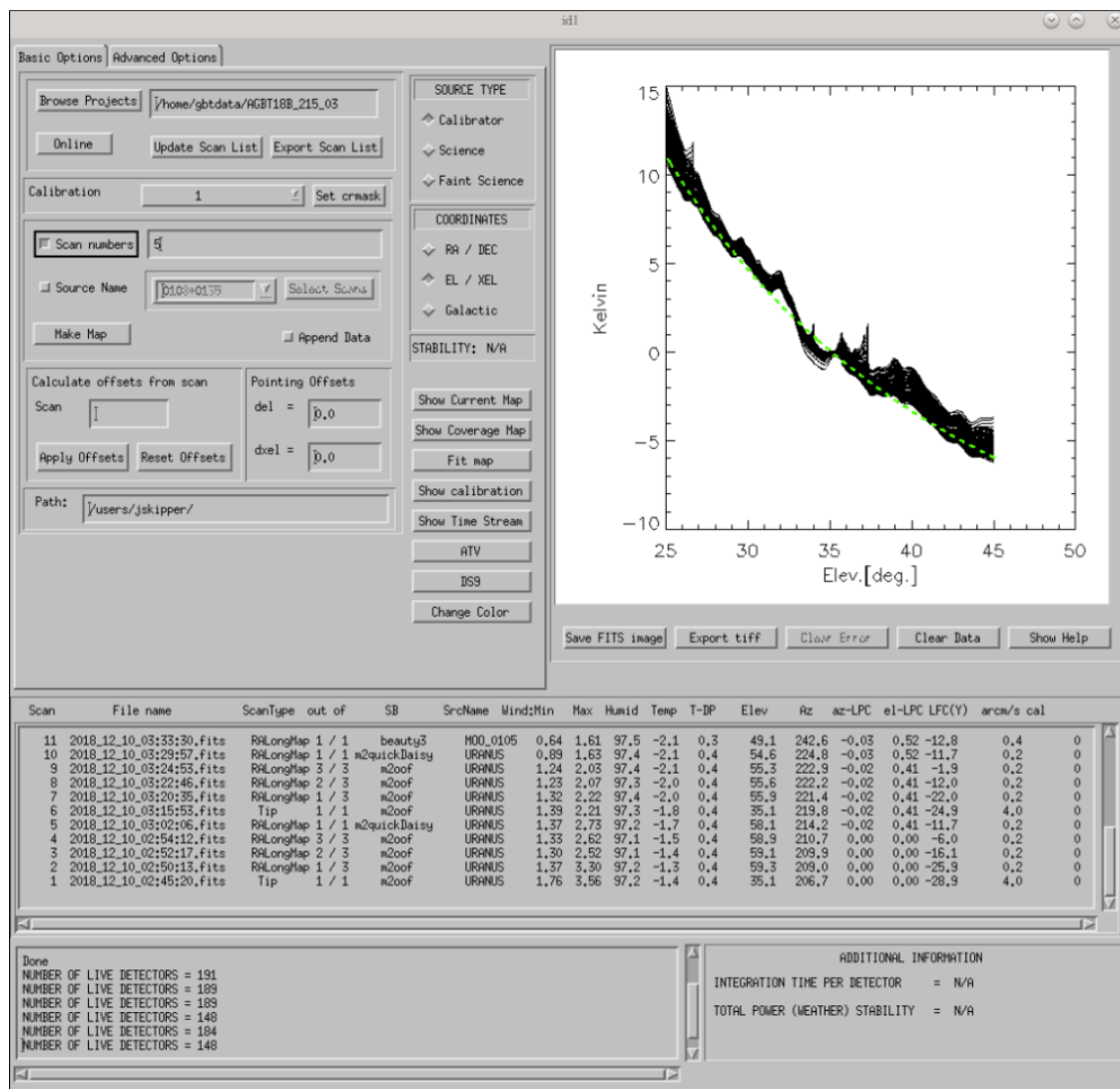
### Examples of tipping scans

A good weather skydip. The black lines (one for each roach) should be fairly free of wiggles and the dashed green line (which is the fit) should follow the black lines fairly closely.





A bad weather skydip. The black lines (one for each roach) are full of wiggles and the dashed green line (the fit) is not following the black lines well.

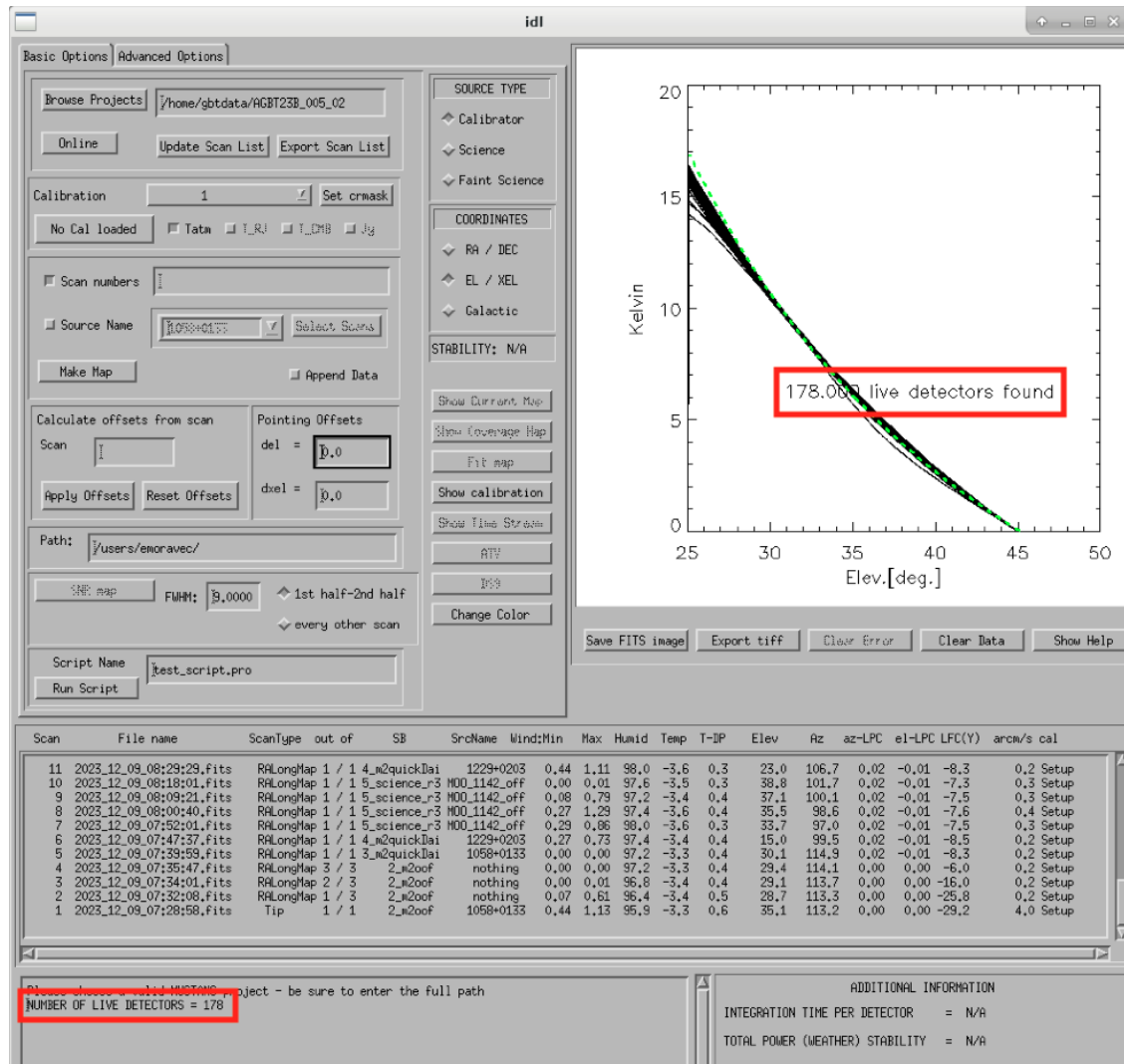


If the tipping scan doesn't look right (a lot of wiggles), try running the skydip script in AstrID. This reruns the tipping scan without having to redo the whole OOF. If it still looks bad, check the weather conditions in CLEO. The weather might not be good enough to observe. You can also call one of the M2 instrument team and get their advice.

### 3. Check the number of live detectors

At this stage, check the number of live detectors, as well as throughout the night. Record this in your observing log.

In the image below, you can see where to check the number of live detectors:



Generally it's good to have 170+ live detectors, however it can sometimes be as low as 160 if the tuning step didn't go very well. If you see this number as low as the 150s or 140s (especially if it's lower than that, which it shouldn't be) be sure to contact a M2 team member. You can also try re-tuning (see section A) and hope that that fixes it.

#### 4. Continue

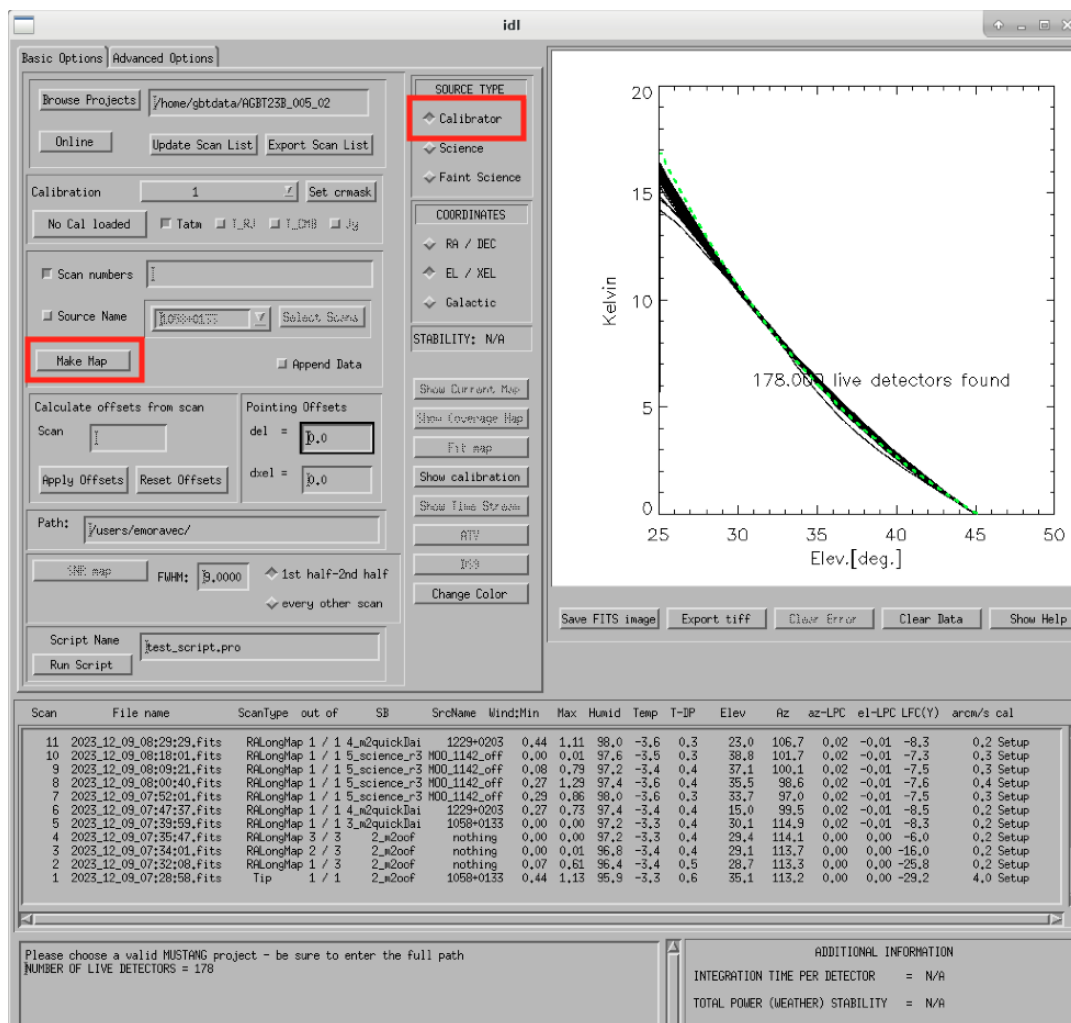
If the tipping scan and number of live detectors look good.

### 4.3 Checking Calibrator/Beam Parameters

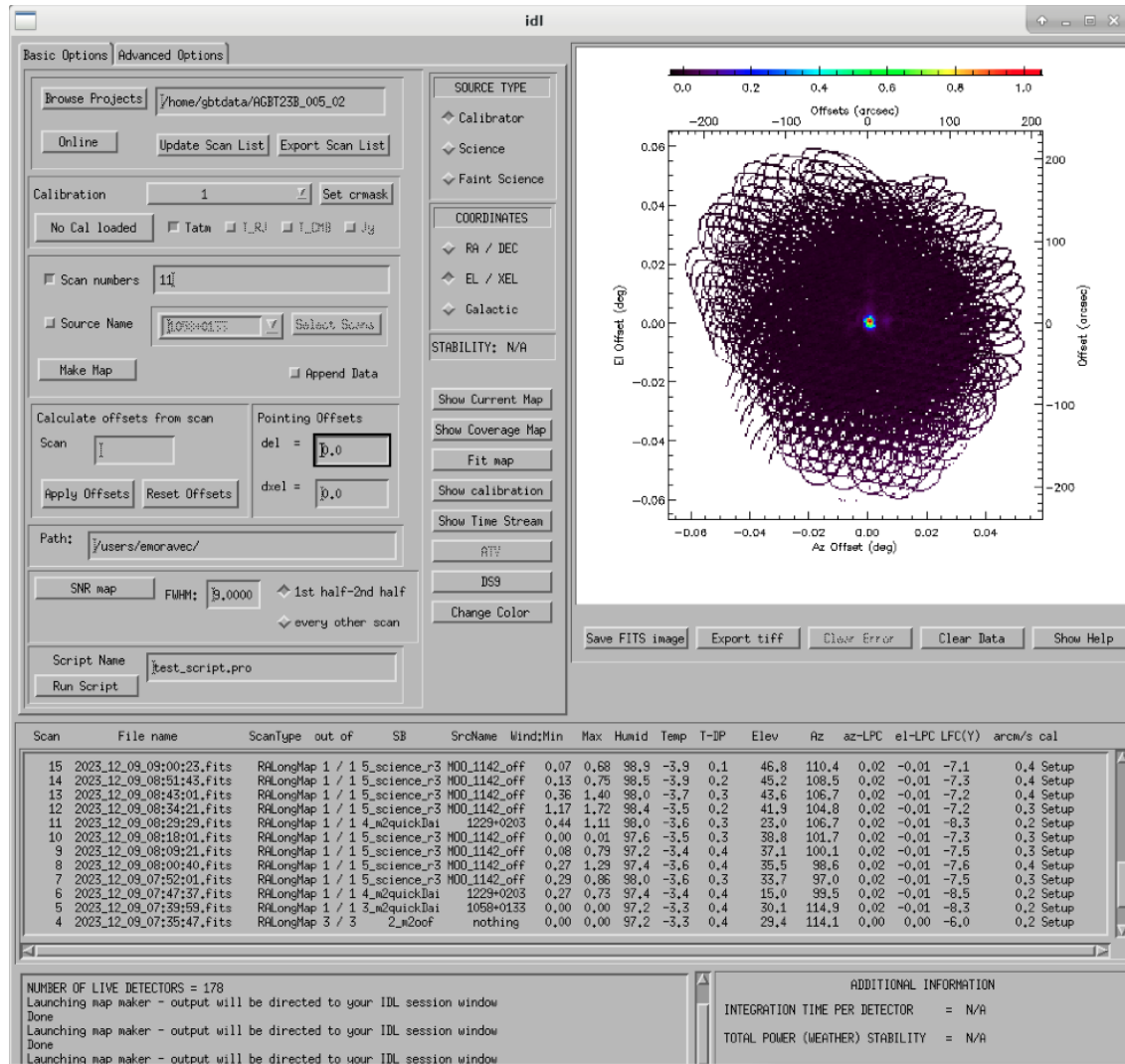
#### 1. Make map

To make a map of a calibrator, after you have run the m2quickDaisy script on a source in AstrID

- Click Update Scan List to find the source scan number of the source you just observed
- Set the Scan Numbers to the scan number of interest
- Set Source Type to Calibrator
- Click Make Map



This will open up an image of the daisy map that you selected. The map should look something like this:

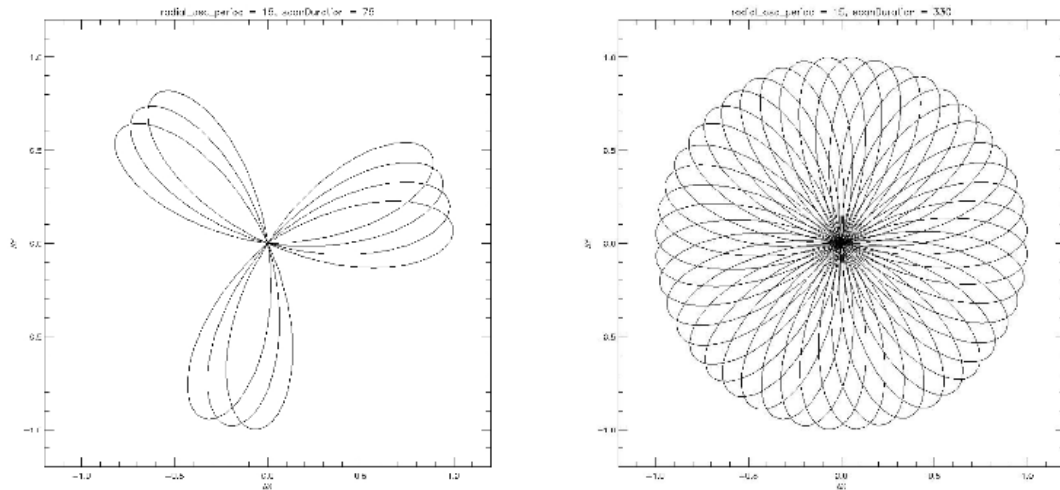


What you see at this stage is an image of the daisy scan. In the center is your calibrator source, visible because it is a bright source. Later, when looking at daisy scans of your science source, it's very likely that you will only see a flat map in the center because it's so much more faint.

The units of the color-coding of this map are in Kelvin of the forward beam. The forward beam is calibrated for the estimated sky temperature at that elevation that we gleaned from our tipping scan earlier on in the night. Therefore, the forward beam temperature should hover around zero if everything is calibrated correctly.

### What is a Daisy Map?

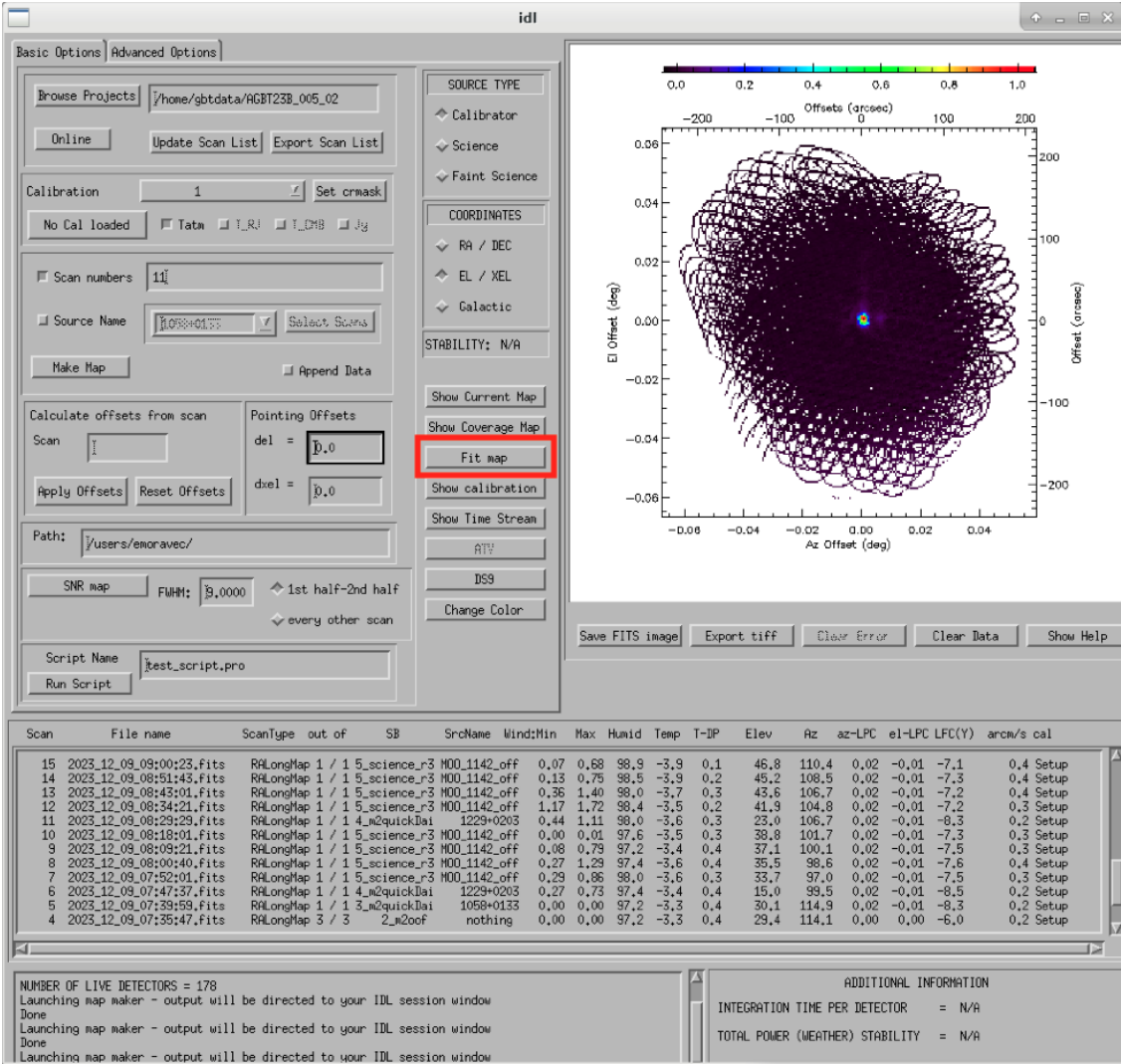
The maps that the M2 team makes are called daisy scans. This is because they loop many times around a central point, looking somewhat like daisy petals. This emphasizes exposure time on the center of the map, with less exposure on the outside edges of the map, making the center of the map more accurately calibrated. They then use the outside of the map to calibrate the sky temperature and remove these effects in the center of the daisy in later post-processing.

(a) Daisy scan with  $\text{scanDuration} = 5 \times \text{radial\_osc\_period}$ .(b) Daisy scan with  $\text{scanDuration} = 22 \times \text{radial\_osc\_period}$ .

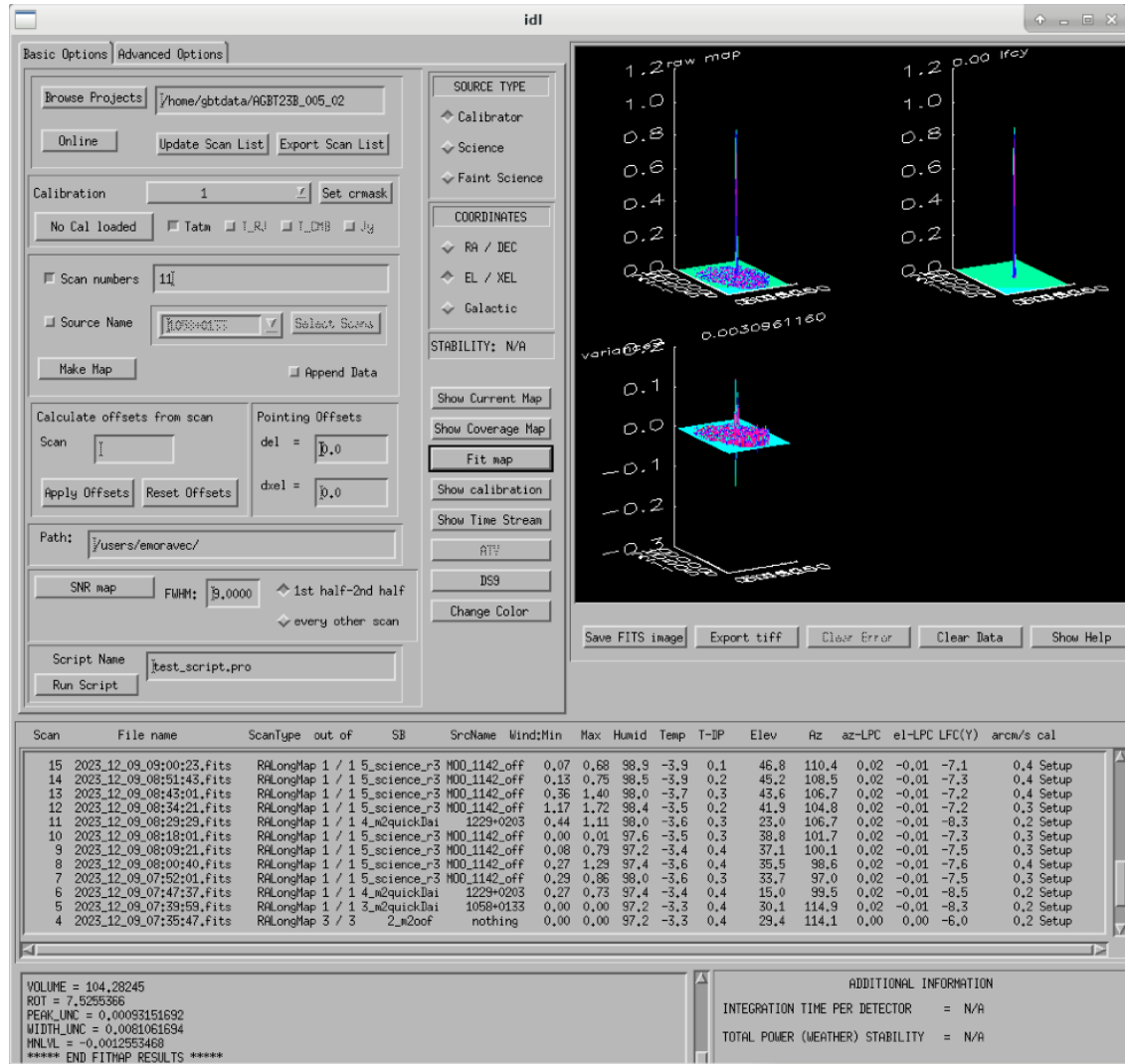
The lines drawn on the map designate the beam path of the GBT on the sky relative to your source. As you can see, each loop begins at the source, extends out, and then returns to the source. This is done throughout the space around your source. Because every loop returns to your source, this results in a higher exposure time on your source relative to the rest of the sky. However, because the units are in Kelvin of the forward beam, this does not mean a higher temperature, but instead simply less noise in the map.

## 2. Fit Map

Click Fit Map.



This will produce the following plots in the gui.



### 3. Check fitting parameters

The fit parameters will be printed out in your terminal.



```

Terminal - emoravec@euclid:~
File Edit View Terminal Tabs Help
Iter 25 CHI-SQUARE = 5577190.7 DOF = 21269
P(0) = -0.00125535
P(1) = 1.11429
P(2) = 3.73331
P(3) = 4.44568
P(4) = 246.033
P(5) = 242.274
P(6) = 5.95474
% Compiled module: EXTAST.
% Compiled module: REMCHAR.
% Compiled module: XYZAD.
% Compiled module: WCSXY2SPH.
% Compiled module: DETERM.
** Structure <1174218>, 15 tags, length=120, data length=116, refs=1:
LFCY FLOAT 0.00000
PEAK_HEIGHT DOUBLE 1.1142911
WIDTH DOUBLE 9.6300149
WIDTHA DOUBLE 10.468768
WIDTHHB DOUBLE 8.7912615
SD DOUBLE 0.0030961160
EL_OFFSET DOUBLE 0.51822513
XEL_OFFSET DOUBLE 1.7773469
XO DOUBLE 0.0010492627
YO DOUBLE 0.00069950895
VOLUME DOUBLE 104.28245
ROT DOUBLE 7.5255366
PEAK_UNC DOUBLE 0.00093151692
WIDTH_UNC DOUBLE 0.0081061694
MNLVL DOUBLE -0.0012553468
** Structure <1174218>, 15 tags, length=120, data length=116, refs=2:
LFCY FLOAT 0.00000
PEAK_HEIGHT DOUBLE 1.1142911
WIDTH DOUBLE 9.6300149
WIDTHA DOUBLE 10.468768
WIDTHHB DOUBLE 8.7912615
SD DOUBLE 0.0030961160
EL_OFFSET DOUBLE 0.51822513
XEL_OFFSET DOUBLE 1.7773469
XO DOUBLE 0.0010492627
YO DOUBLE 0.00069950895
VOLUME DOUBLE 104.28245
ROT DOUBLE 7.5255366
PEAK_UNC DOUBLE 0.00093151692
WIDTH_UNC DOUBLE 0.0081061694
MNLVL DOUBLE -0.0012553468
% Program caused arithmetic error: Floating underflow
IDL>

```

---

**Note:** The Floating underflow error you see in the output is **not** a concern.

---

#### 4. Record values

Write down the values for PEAK\_HEIGHT, WIDTHA, and WIDTHHB in the observing log to compare to later pointing scans to monitor the beam and decide if you need to re-OOF.

#### 4.4 Checking Science Scans

If you would like to make a map of of a science scan(s), you can do so by following the same steps as making a map of a calibrator with the following modification

- under Source Type select Faint Science

---

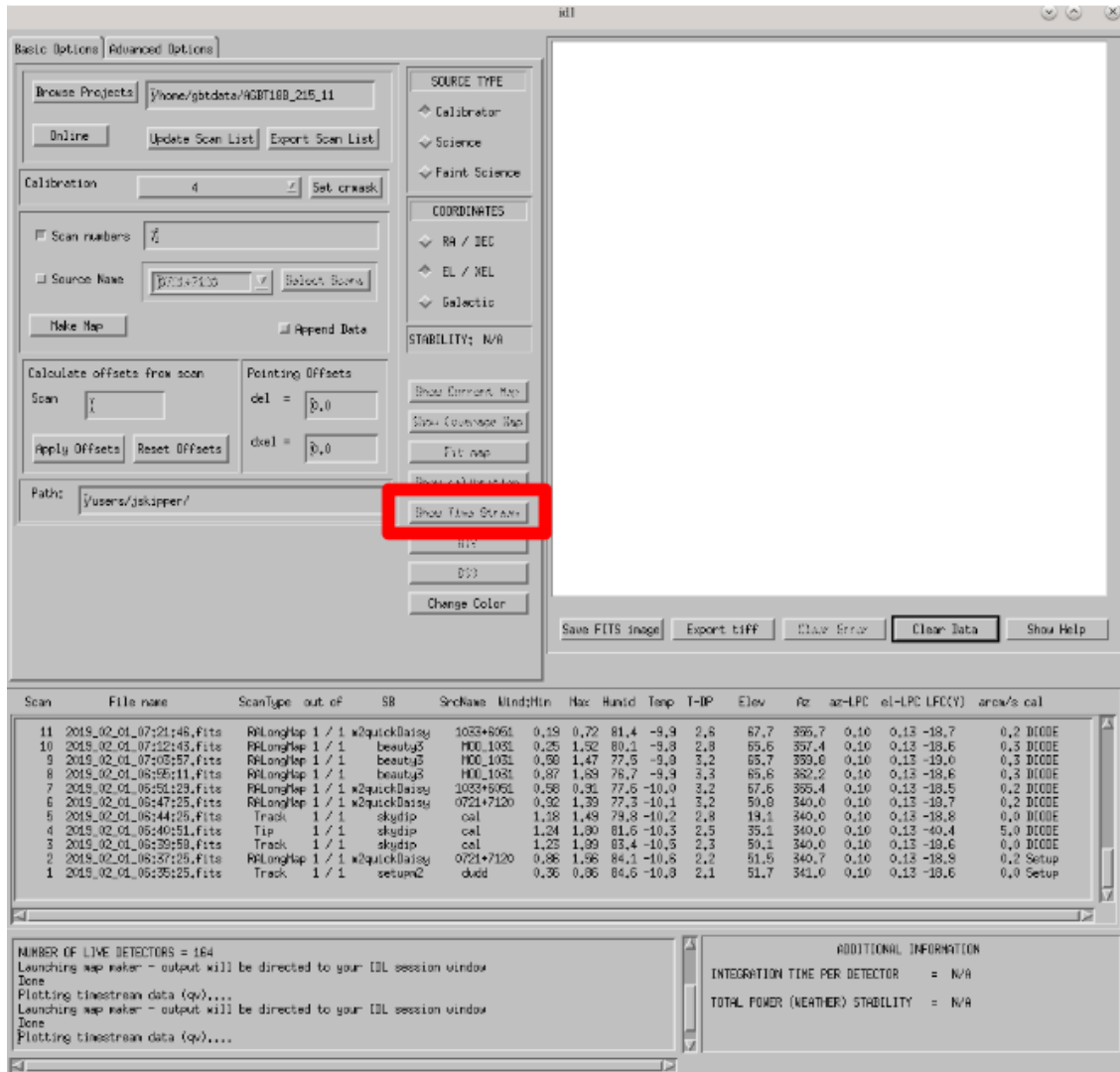
**Note:** You can add several science scans together by putting them all separated by commas in the scan list.

---

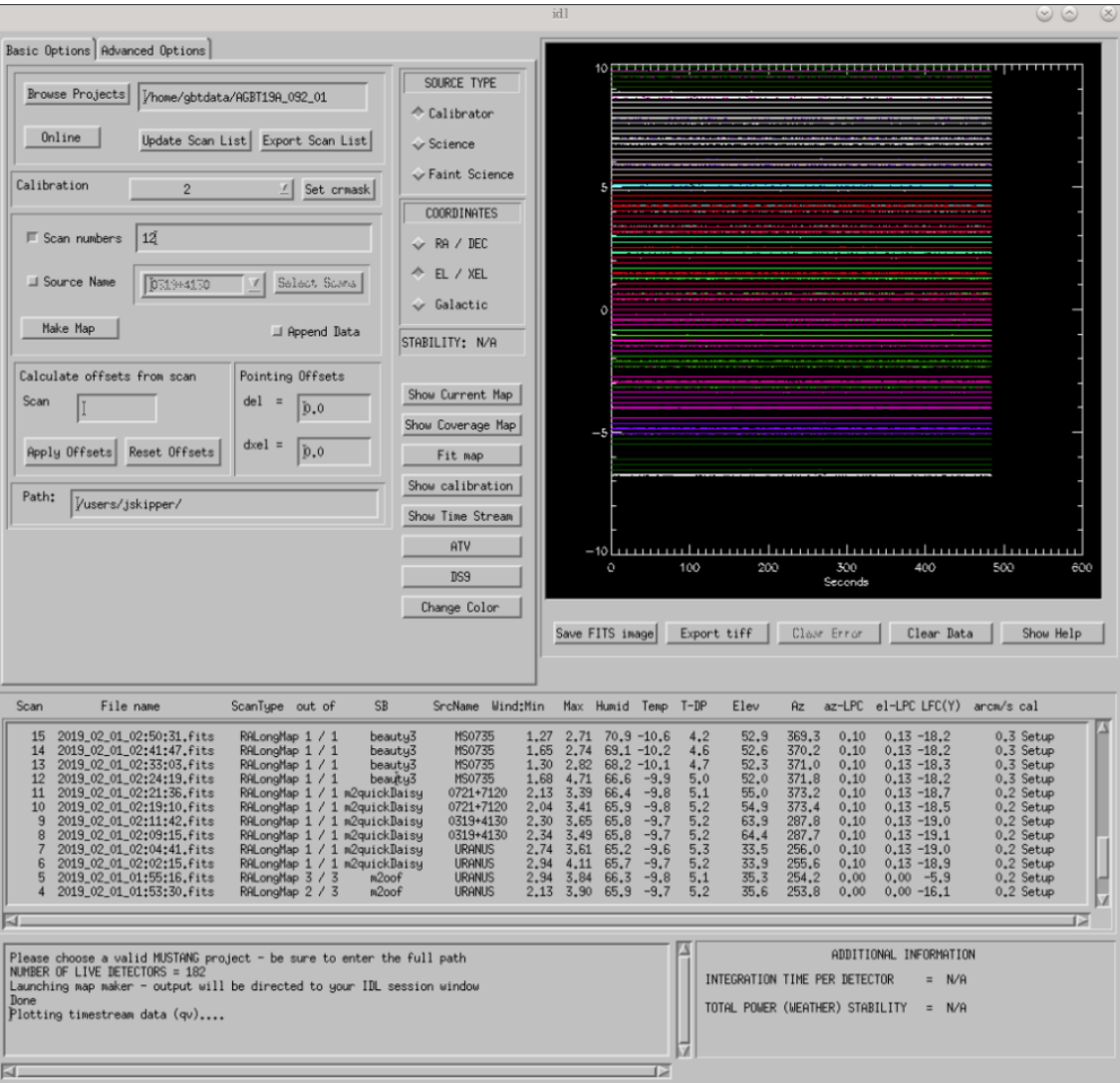
## 4.5 Checking Time Streams

It is a good idea to check the time streams (checking how the sky temperature is changing over time) as well as the maps. To do so:

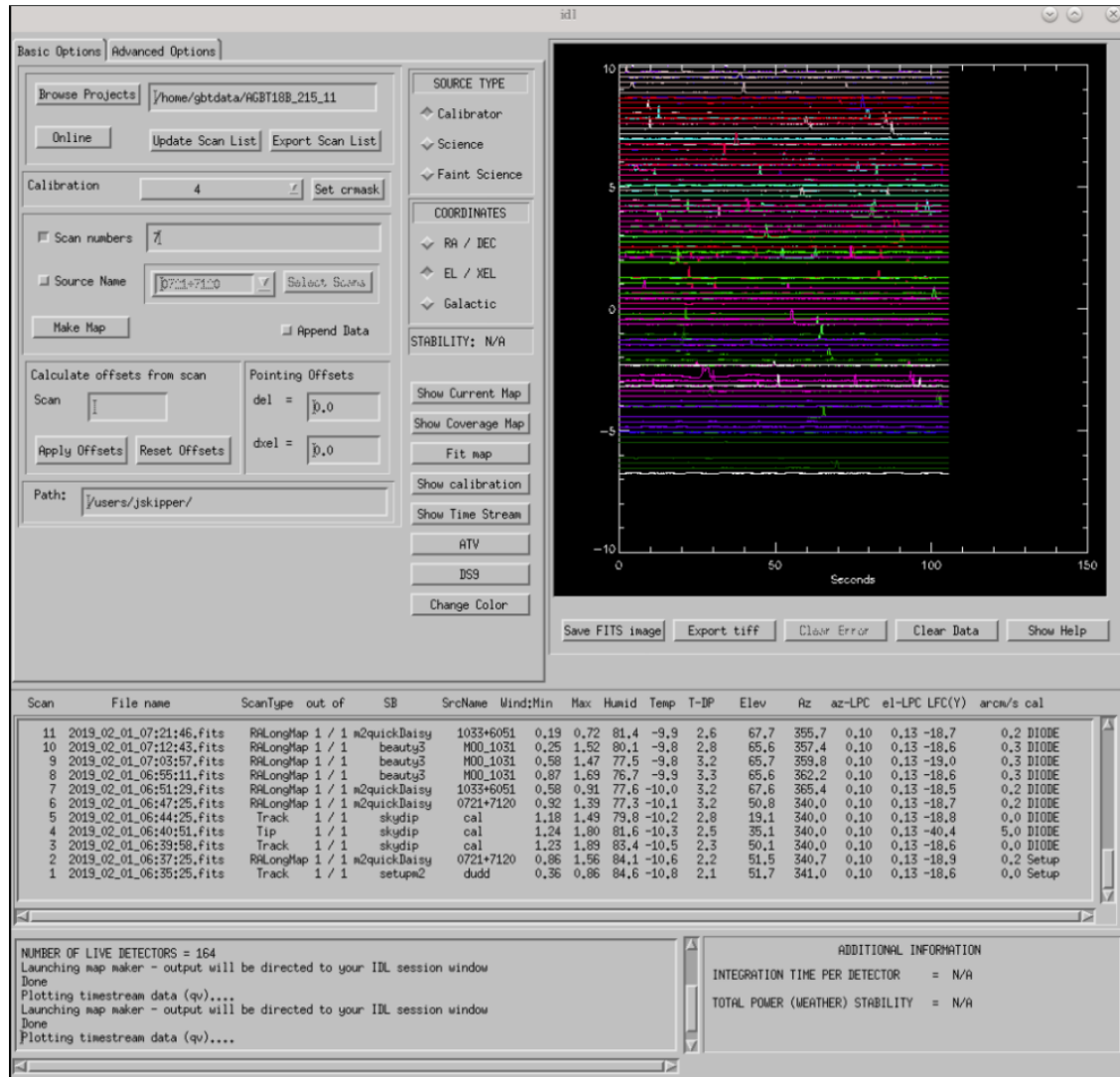
- Make your map (see [4.3 Checking Calibrator/Beam Parameters](#))
- Click **show time stream** button underneath the **Fit Map** button after making your map



## Examples



Faint science time streams (a cluster) in good weather.



Calibrator time streams in bad weather. Note that these calibrator time streams still look similar to calibrator calibrator time streams in good weather due to the bright nature of the calibrator sources.

#### 4.6 Troubleshooting: m2gui hangs

If your m2gui is hanging (won't quit) do the following in a terminal:

```
ps -u
```

Find the PIDs of startm2gui and idl and kill both.

```
kill -9 PID
```

## 5. General Advice for Determining “Bad Weather”

Once you have some indication of bad weather (bad skydip, bad time streams, or physical weather indication), you will want to make an educated guess as to what the trajectory of the weather/data is in order to determine whether or not to keep observing or give up the time. There are many tools that you can use to an assessment of this trajectory. Consider, do the following suggest that the remainder of your scans would be scientifically useful? (this can be used as a checklist of sorts)

- **Time streams**
  - Check the time streams of the science scans as laid out above in B3.4. Are they wiggly? How wiggly?
  - How many “bad” science scans have there been in a row?
- **Skydip(s)**
  - How does the first skydip of night look? How wiggly is it?
  - If you are seeing indications of bad weather and you decide to OOF again one could add a skydip in to test the weather (calSeq=True).
  - One could even do a one off skydip.
- **Beam**
  - Has the beam been deteriorating?
- **Weather forecast**
  - Check <https://www.gb.nrao.edu/~rmaddale/Weather/AllOverviews.html>.
  - Check another reputable weather forecaster (Weather underground, weather.forcast.gov, Windy, etc.)
- **Direct communication with the operator**
  - Ask the operator what the weather is like. Since you asked at the beginning of the observation you have one data point.
  - This also serves as a way to keep the operator in the loop and aware of a potentially imminent decision to relinquish telescope control.

---

**Note:** The observer should reach out to the operator once the concern of bad weather is identified to let them know that the weather is a concern. This could be as early as the first bad scan (time streams, whether a science scan or those from a skydip). A good practice is that if there are two consecutive scans with bad time streams, the operator should be notified and consulted at this point. That doesn’t mean a decision needs to be made this early on, but it lays the groundwork so that both parties are aware of a potentially imminent decision to relinquish telescope control. If the observer has doubts, reach out to an M2 team member after a second bad scan.

---

A few data/weather trajectories are as follows:

- **Improve**
  - Is it a one off? As in its just a cloud passing by?
  - Is the or will the weather improve?

- Stay the same. Is the weather staying bad and not improving?
- Get worse. Is the trajectory getting worse and worse?

You will need to monitoring the situation over time and over multiple scans in order to make a guess about the trajectory of the data. One note is the it is usually never sufficient to come across one bad scan and call it quits. There is usually always some nebulous time span (~half hour to an hour) to determine that things are bad and staying bad. If you think the weather will improve and the improvement should happen soon and give ample time for valuable science scans, then the suggestion is to try to endure the bad weather. However, for weather staying the same and getting worse, the advice is to rely on the other metrics to make a determination, except for the case that the operator identifies clear precipitation with no expectation for improvement. At that point, one can give up the time promptly if it's heavily raining or snowing.

**When making a judgment call as to whether to give up the time due to bad weather, consider the following cases:**

- How much time is left? If there is not much time left it is less likely that the weather will change.
- Are you observing a faint target? If you give up amount of time you have left, will that amount of time you have left make a difference for your science?
- How much time has been observed for the project and how much time is left in the project? We ask for a factor of 2 of overheads so maybe there is time to tolerate bad weather.

---

**Note:** ~30 minutes is a rough minimum amount of time to relinquish control, but the operator will need some time to prepare a backup project so this is why it is good to keep in touch with the operator throughout this process. So the general advice is that if you give up the time near the end of an observation, the minimum time left in an observing session would be ~45 minutes.

---

---

**Note:** The flip side of overheads (i.e. maybe the project can tolerate bad weather) is that if you are observing the last session (using up all awarded time), any rescheduled observing would all go to overheads. If it's not the last session, then the advice is to give up the remainder of time for bad weather (if all bad-weather items are checked).

---

Again, when in doubt you can always call an M2 team member to help you make the call of whether or not to give up the time.

## 6. Changing M2 Projects/Second M2 Project of the Night

If you are observing for an M2 project that is not the first M2 project of the night then before observing you will need to create a link for the tuning so that OOF & data reduction can find the right tuning.

### 6.1 Make symlink

Before you begin observing, login to egret and type:

```
cd /home/gbtlogs/Rcvr_MBA1_5tuning/  
ln -s <old_project_session> <new_project_session>
```

where `old_project_session` is the full name of the previous M2 project and `new_project_session` is the second M2 project of the night that you are observing for.

**Warning:** Be very careful to put in the right project and session ID or this step will not work and you won't get any data. You can ask the previous observer for the old project session ID, or look for it by typing:

```
ls -ltr /home/gbtdata/
```

The last modified file will tell you what the most recent project ID was.

## 6.2 Run `m2setup`

When the observing time for the second project starts, you need run `m2setup` in AstrID again. This is already outlined in the directions.

**Warning:** Some people think they can skip this step when changing from another MUSTANG-2 run. This is not the case. It's very important to still run `m2setup` at the beginning of your session.

## 6.3 Skydip/OOF

You can possibly skip OOFing at the beginning of this second project. You can ask the previous observer when they last did an OOF and what the progression of the beam was.

- **If you need to re-OOF**
  - make sure that `calSeq=True` to get a skydip
- **If you do not need to re-OOF**
  - do a stand-alone skydip and change `myAz` to the Azimuth of whatever your first source will be (calibrator, etc.). The telescope will slew to that Az.

## 6.4 Flux calibrator

You'll also want to still observe your flux calibrator using the `m2quickdaisy` script.

**Warning:** This is another thing people think they can skip, but it makes reduction later more difficult. Check the beam with this flux calibrator.

# 7. Observing Troubleshooting

## 7.1 MUSTANG-2 Manager

Sometimes the MUSTANG-2 manager refuses to start - you try to start it and you get a failure every time (using TaskMaster or asking the operator to do this for you).

**The solution is to**

- log onto egret
- shut the computer down
- log onto the iboot bar
- power off egret and the housekeeping
- leave it off for 30 seconds

- turn these back on

Egret may take a while to reboot but once it does you should be able to restart the manager. Assuming this works you should also make sure to press the `reset heater card` button on the manager twice.

## 8. Closing up for the night

### 8.1 Go offline

In AstrID, go from **working online** to **working offline**:

- File → Real time mode ... → work offline.

### 8.2 Shutdown M2

For the shutdown process you can either do this (a) **automatically** or (b) **manually**.

Run the following script in a terminal:

```
/users/penarray/Public/stopMUSTANG.bash
cd /users/penarray/Public
./stopMUSTANG.bash
```

#### 1. Set detector biases to zero

- Go to the Mustang Manager in CLEO
- Click on the miscellaneous tab
- In the top middle, you will see 4 rows of Det Bias 1-4, corresponding to the 4 roaches.
- Unlock the manager
- **roach-by-roach:**
  - type 0 in the left DetBias box
  - press enter
  - wait until the blue box (right DetBias box) shows a DetBias of 0
  - repeat this step for all 4 roaches.

#### 2. Turn off data transmission

- Mustang2 CLEO scan turn off `DataXinit` for all four roaches.

---

**Note:** You will need to be in gateway AND unlock both the `unlock` and `advanced features unlock` buttons to do this.

---

#### 3. Turn off components

In VNC session, go to <http://mustangboot.gbt.nrao.edu> and turn off the *roaches*, *HEMTs*, and *Function Generator* by checking those three boxes then go to left of the screen and click ‘Off’ (gray button).

#### 4. Turn on daily cycle

**Mustang2 CLEO window**

- go to Housekeeping



- unlock
- **recheck daily cycle to be on and put autocycle trigger to HE4**  
This means that if either of the He4 fridges run out it starts a cycle.
- **set the daily cycle time = 0.65 of a day in UT**  
This is the time of day that the daily cycle starts measured in fraction of a day (UT). 0.65 is a nice balance between ensuring the cycle is over by the time any observations are likely to come up, yet not so early that there is no time to work with the receiver in the morning.

### 8.3 Kill VNC session

Either kill your FastX session or your VNC session via the terminal.

---

### Congratulations!

You're all done! Now, let's do some science with that data!

---

#### 2.4.2.4 MUSTANG-2 data

Here are the current options for obtaining and working with MUSTANG-2 data. Unless the PI is well-versed in working with MUSTANG-2 data, most often PIs request one or more of the *deliverables* that the MUSTANG-2 team offers. One can also do a quick, simple reduction (see Basic Reduction below).

---

#### Deliverables

Description of deliverables.

[See Deliverables](#)

#### Basic Reduction

MUSTANG-2 wiki page that describes how to make preliminary images from MUSTANG-2 data.

[See Guide](#)

### MUSTANG-2 Deliverables

**The MUSTANG-2 team is able to produce the following data products for all MUSTANG-2 projects:**

- **A calibrated map either in Jy/beam or Kelvin (main beam).**
  - Each scan is gridded individually. Maps are stacked via weighted averaging.
- **An associated noise map**
  - The default noise map, for a single scan, flips every other detector. The combined noise map is the stack of all individual scans.
- **An associated SNR map**
  - The above map and noise map are smoothed by some amount (generally 9). The weight map is scaled according to the RMS in the noise map

**Additionally, the following products will help in using the above data products:**

- **A transfer function**

This accounts for the filtering we perform on the data. The calibrated map referenced above does not preserve signals on all scales; a transfer function (along with convolution of the beam shape) must be taken into account when modelling the intrinsic astronomical signal. More information on how this is calculated and can be used can be found at [https://safe.nrao.edu/wiki/bin/view/GB/Pennarray/MUSTANG\\_CLASH](https://safe.nrao.edu/wiki/bin/view/GB/Pennarray/MUSTANG_CLASH). A repository of transfer functions (ascii files) is available [here](#).

- **A stacked beam map**

From the calibrators, observed every ~30 minutes, we can compute an average beam for a given science target. The stacked beam is normalized to have a peak of 1. Double Gaussian (azimuthally symmetric) fits are included in the header (among the last cards in the header). The user can also produce their own fits to the stacked beam, if they so choose.

---

## 2.5 Software

### GBTIDL

How-to process spectra using GBTIDL (examples)

[GBTIDL](#)

#### 2.5.1 How to process spectra using GBTIDL (examples)

Below are a series of common data reduction GBTIDL processes with explanations and screenshots mixed in. To get an in-depth explanation for any gbtidl-specific procedure or function, you can type “usage” with the procedure/function name and the “/verbose” option:

```
usage, “getps”, /verbose
```

##### 2.5.1.1 Basic On/Off

Here is an example of a basic data reduction process on an L-band On/Off procedure.

---

#### Data

ngc2415.fits

---

Load the data into gbtidl.

```
filein, "data/ngc2415.fits"
```

Display a summary of its contents.

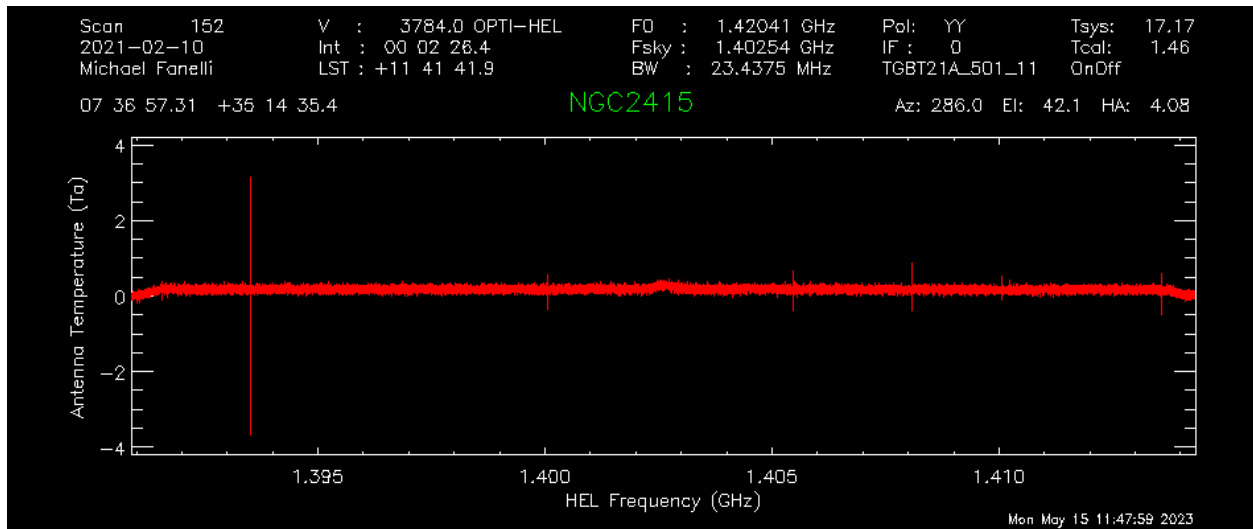
```
summary
```

```
GBTIDL -> summary
  Scan      Source      Vel      Proc Seq      RestF nIF nInt nFd      Az      El
  -----
    152      NGC2415    3784.0    OnOff  1      1.420   5  151   1  286.0  42.1
    153      NGC2415    3784.0    OnOff  2      1.420   5  151   1  286.6  41.6
```

We can see that this data set has two scans on the galaxy NGC 2415, with one slightly offset in position. The catalog used for this observation says the galaxy has a velocity of 3784 km/s, the rest frequency of the observed line in the first frequency window is 1420 MHz/1.420 GHz, there are 5 frequency windows, 151 integrations, 1 beam/feed, and lastly the azimuth/elevation values of the GBT at the time of the scan.

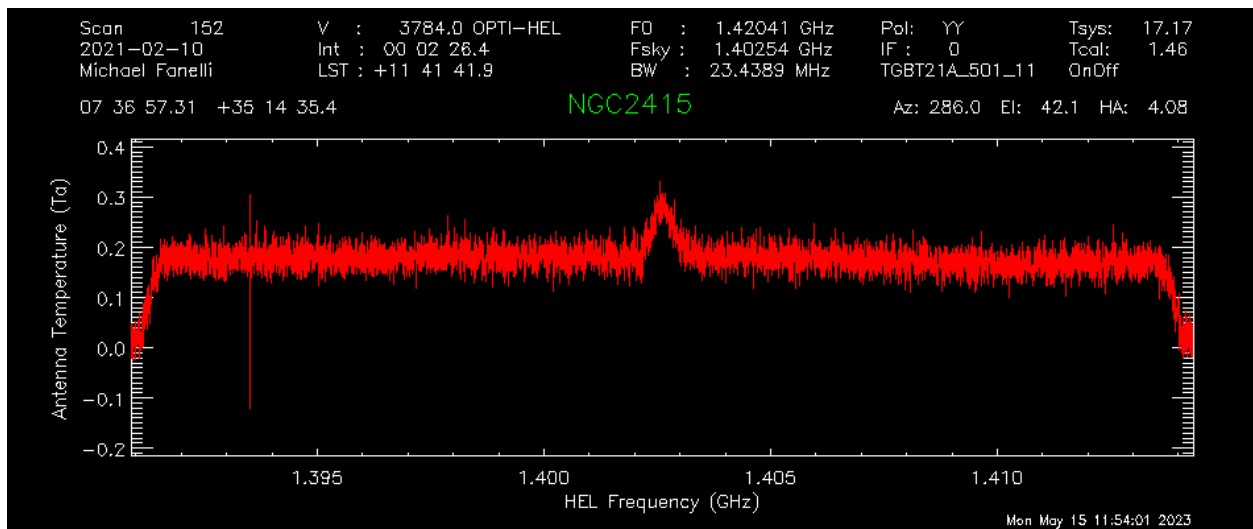
For a first look at the spectrum, we call the “getps” procedure, which calibrates the On vs. Off scans and averages up all 151 integrations. Without designating the polarization or spectral window, it will default to the first polarization (Linear YY for the L-band receiver) and first spectral window (centered on the redshifted HI line in this case).

```
getps, 152
```



In order to see anything useful, we may have to smooth the data. This is usually done with the procedures “gsmooth” or “boxcar”, with the latter using a gaussian smoothing kernel and the latter using a flat kernel. The first input “N” is the size of the kernel in channels and the second option keeps only every N-th channel.

```
gsmooth, 5, /decimate
```

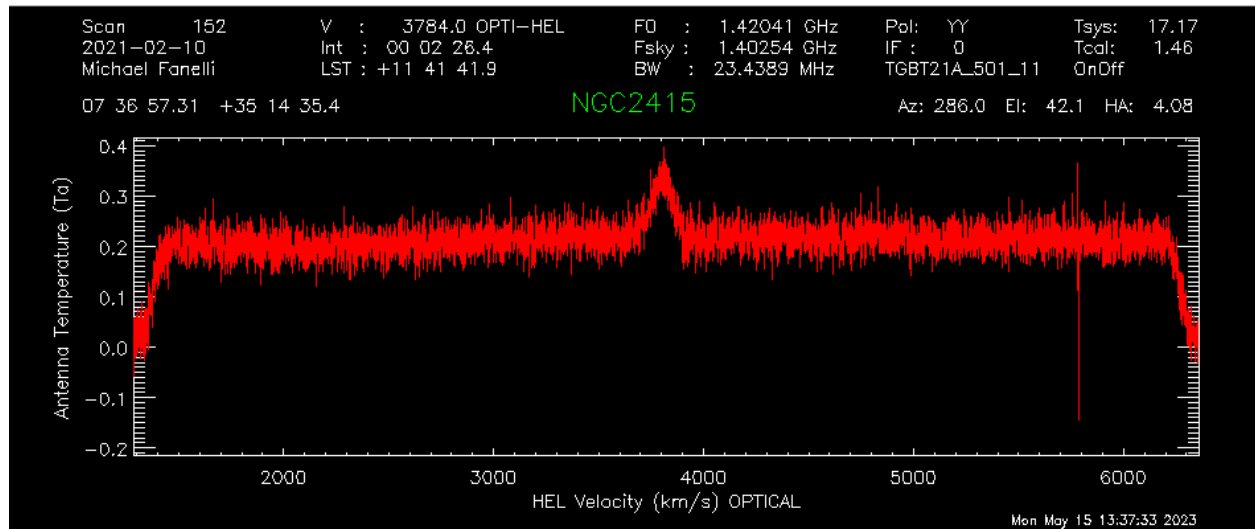


We can convert the data to the same velocity frame specified in the observing configuration with “setxunit”. By default this will apply a doppler correction. This can also be done by selecting “km/s” in the GUI - the fourth dropdown menu from the left.

```
setxunit, "km/s"
```

We may learn that the TCAL value in the database is out of date, and the signals are too weak by a factor of 1.2. “scale” removes this systematic offset.

```
scale, 1.2
```



Now we may want to save these results for posterity or further analysis. You can also use “write\_csv” to make a csv file instead of an ascii-columned text file, or “write\_ps” to output a “publication-quality” postscript image of the spectrum.

```
write_ascii,"NGC2415_HI.ascii"
```

gbitd also allows for multi-line input and loops in the same line. We can use this to quickly check the other spectral windows. The line breaks are made with the “&” character. First, let’s freeze the plotter so it doesn’t auto-update after each loop:

```
freeze
```

And change the x-axis back to frequency so that each spectral window is plotted correctly.

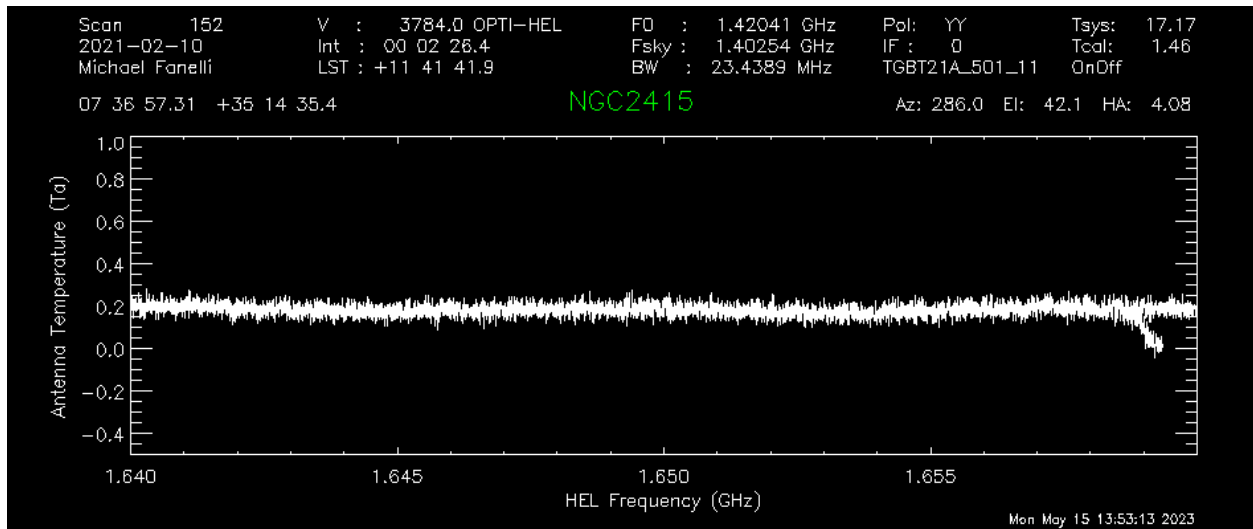
```
setxunit, "GHz"
```

Now finally type in the loop:

```
for k=1,4,1 do begin & getps,152,ifnum=k & gsmooth,5,/decimate & scale,1.2 & oshow &
endfor
```

Note that the third and fourth spectral windows overlap significantly. We’re not going to save these spectra. We can zoom into the OH line:

```
setxy,1.64,1.66,-0.5,1
```



And it doesn't look like there is anything there.

### 2.5.1.2 Basic Frequency-Switched (fsw)

Here is an example of a basic data reduction process on a nearby ammonia cloud.

#### Data

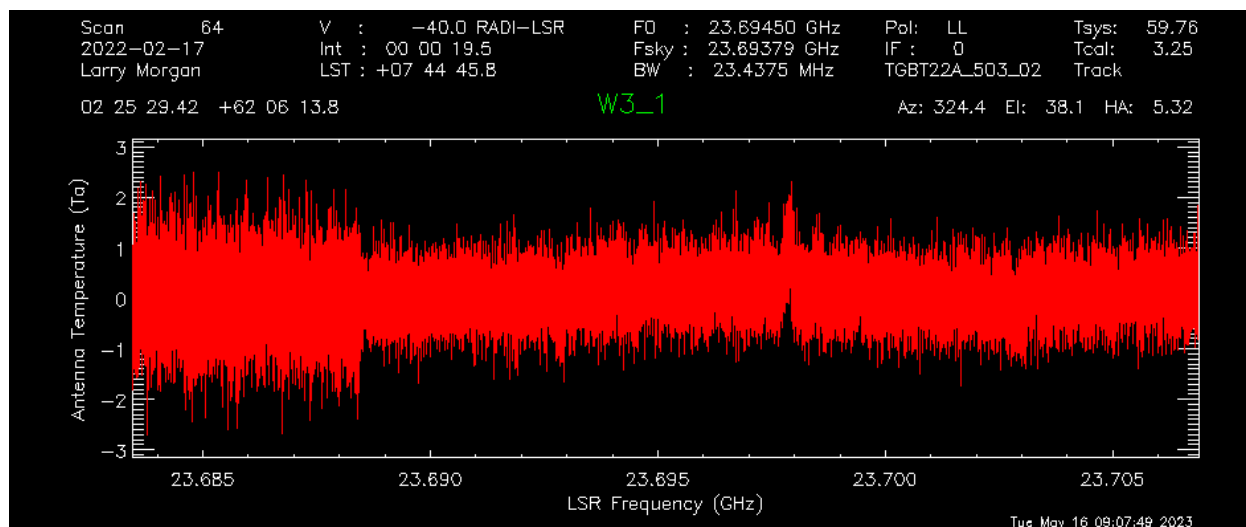
TGBT22A\_503\_02.raw.vegas

Load in the data. This is a directory, so either “dirin” or “filein” will work.

```
filein, "data/TGBT22A_503_02.raw.vegas"
summary
```

In here, there is one frequency-switched scan (#64), and two nodding scans (#62 and #63). For the KFPA, which is 7 beams arranged in a hexagon, the `Track()` command will use the central beam by default, so `fdnum=0`. The frequency-switched scan is calibrated with

```
getfs, 64, fdnum=0
```



It looks like there is a small detection at around 23.698 GHz. We can smooth the spectrum to see a little more clearly.

```
gsmooth,5,/decimate
```

There is a very nice detection of ammonia! Let's average the two polarizations to reduce the noise even further. First, put the current spectrum in the primary accumulation buffer;

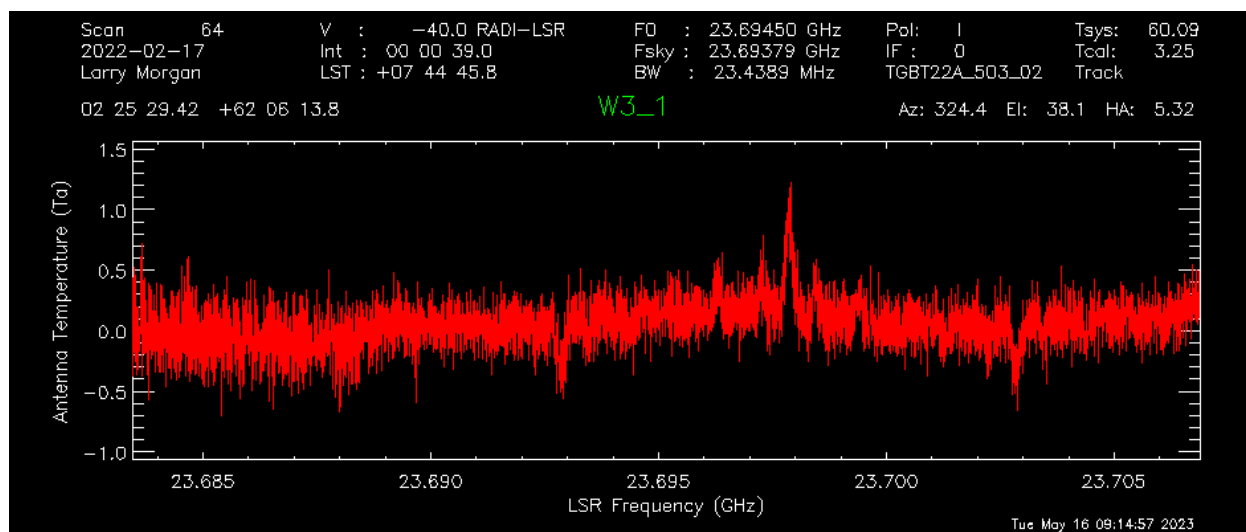
```
accum
```

And load the other polarization, smooth it to the same frequency resolution, and add it to the primary accumulation buffer. You can press the up arrow in GBTIDL for an input history.

```
getfs,64,fdnum=0, plnum=1
gsmooth,5,/decimate
accum
```

Now we average the two spectra in the accumulation buffer together, which will automatically drop the result in the primary data container.

```
ave
```



We can output this spectrum to an sdfits file with the “keep” procedure. We have to set a filename first.

```
fileout, "W3_1_NH3.fits"
keep
```

To compare with the nodding scans in the next example, we can save this to the next data container in GBTIDL.

```
copy, 0, 1
```

### 2.5.1.3 Basic Nodding

Here is an example of a basic data reduction process on a nearby ammonia cloud, from a nodding scan.

#### Data

TGBT22A\_503\_02.raw.vegas

Nodding scans are only done with multibeam receivers on the GBT. They are performed by tracking the source with one beam for a certain amount of time, then moving the whole telescope in azimuth/elevation so that a different beam tracks the source for the same amount of time. It effectively functions like a double On/Off scan. The same data file we used above has two nodding scans on the same ammonia cloud.

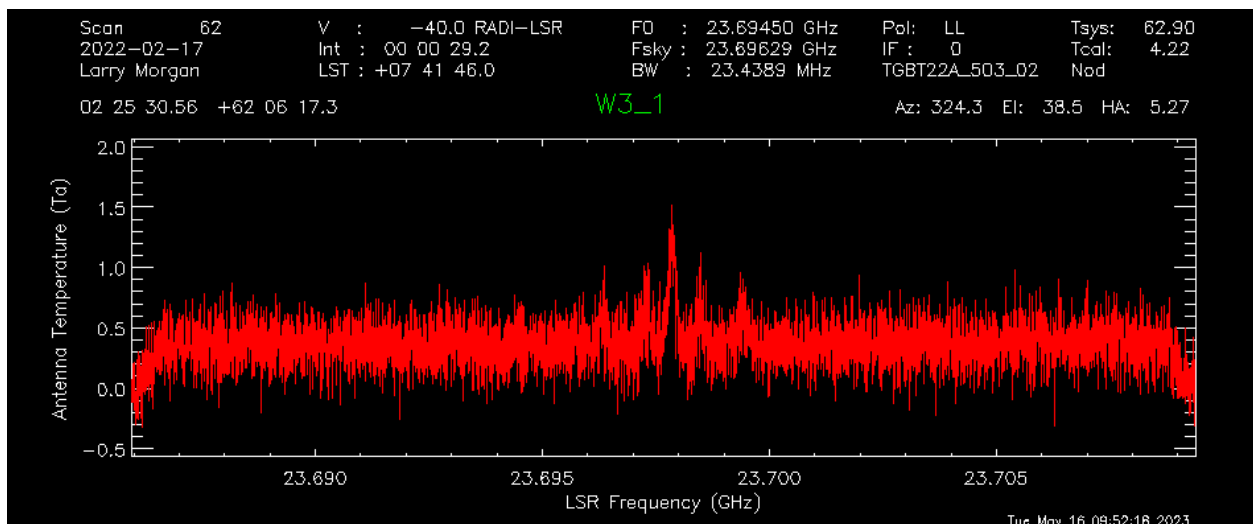
```
filein, "data/TGBT22A_503_02.raw.vegas"
```

Normally, nodding scans are calibrated with “getnod”, which is currently broken for KFPA data. We can use “getsigref” instead, which functions almost identically. We just have to define the signal and reference scans for each beam. For this data, the nodding was between beams 3 and 7, which correspond to fdnum values of 2 and 6. First, “sclear” makes sure the accumulation buffer from the previous example is cleared.

```
sclear
```

Beam 3 was on source in scan 62 (the “signal” scan) and offset in scan 63 (the “reference” scan).

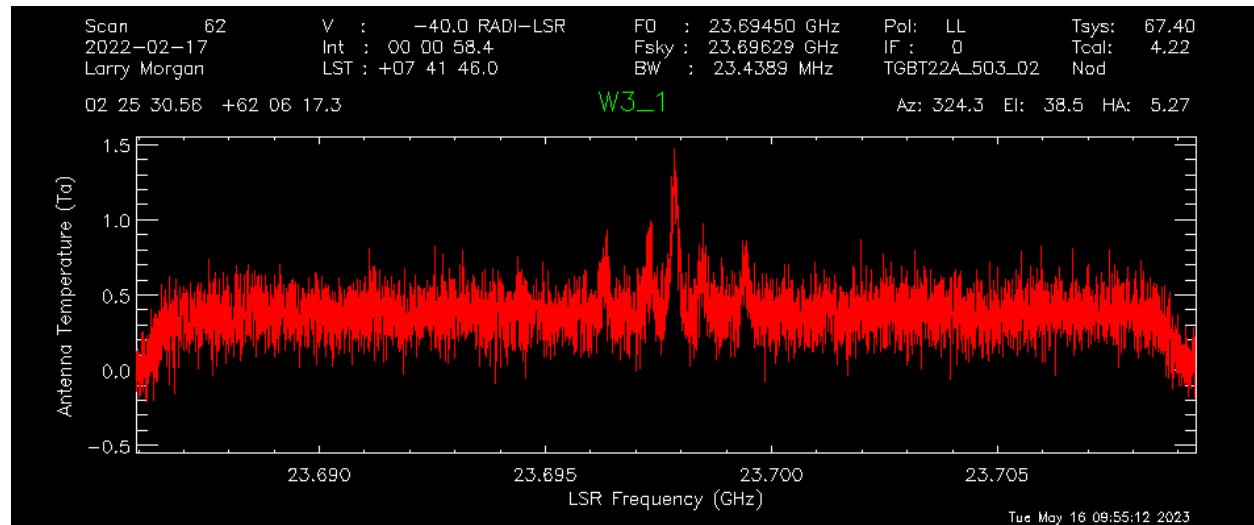
```
getsigref, 62, 63, fdnum=2
gsmooth, 5, /decimate
```



```
accum
```

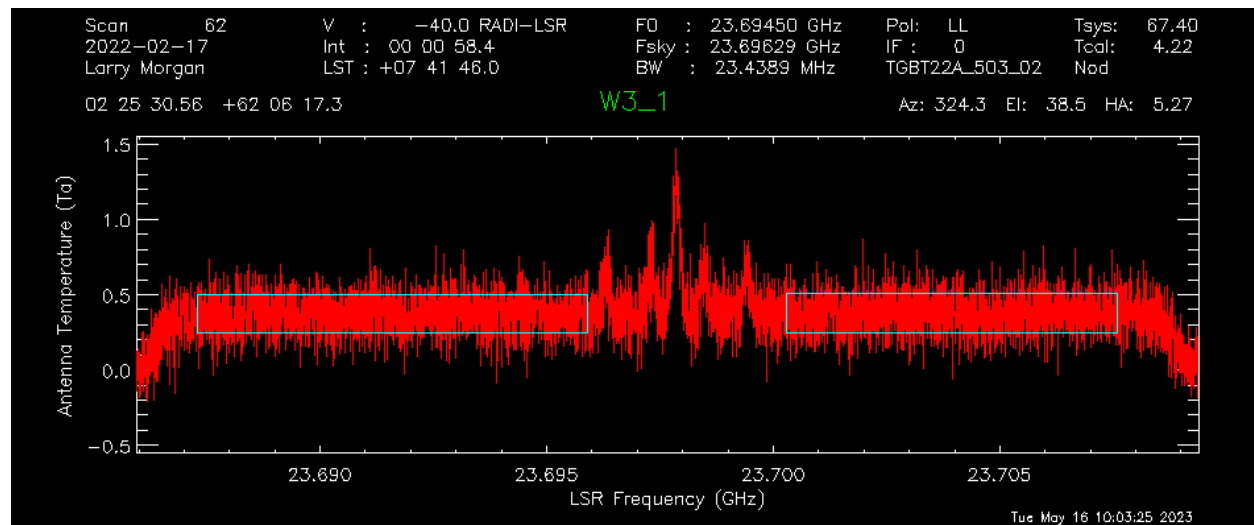
Beam 7 was also part of the nod, but was offset in the opposite way. So, scan 63 is now the signal scan, and 62 is the reference scan.

```
getsigref, 63, 62, fdnum=6
gsmooth, 5, /decimate
accum
ave
```



The continuum is slightly offset from 0, so we can use the baseline feature to subtract that out. “setregion” sets the areas the fitting procedure uses, and this can be done either on the GUI with the left/right mouse buttons or by designating a series of start/stop points in channel number from the command line input. The regions in this case should be everything except for the rolloff at the edges of the band and around the signal itself.

```
setregion
```



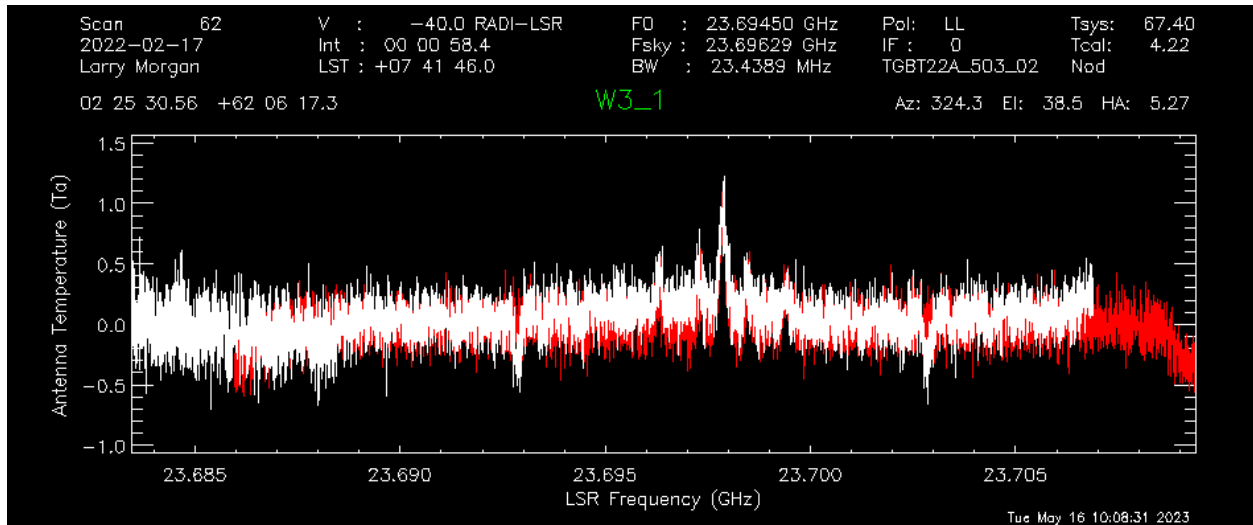
“baseline” by default uses a 0th order polynomial - a flat line - to fit. The continuum is already pretty flat, so this is all that’s needed.



```
baseline
```

Now the spectrum's baseline should be centered about  $T_a = 0$  Kelvin. Next, to compare with the frequency-switched data, we use “oshow” with the number of the data container we saved to.

```
oshow, 1
```



#### 2.5.1.4 Advanced On/Off

##### RFI excision and baselining

#### Data

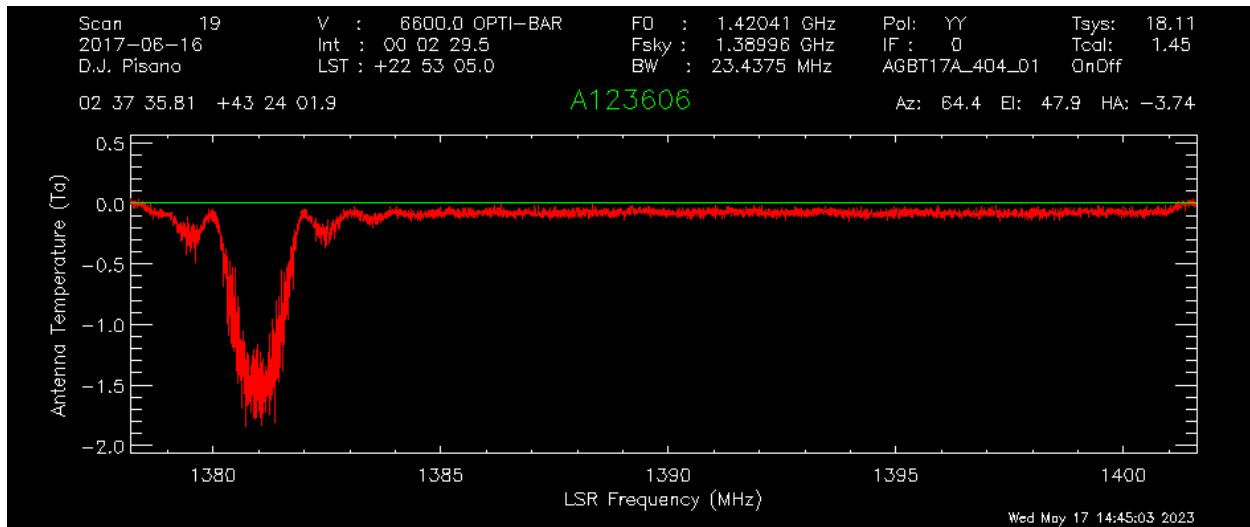
AGBT17A\_404\_01.raw.vegas

Load in the data:

```
filein, "data/AGBT17A_404_01.raw.vegas"
```

You can see there are two sets of position-switched L-band scans here. We will start with the latter two and see if we can find an HI detection:

```
getps, 19
zline
```



“zline” will help with modelling the baseline later. We can see there is a huge GPS-L3 RFI signal flooding out the left side of the band. We can step through one integration at a time (there are 60 total plus one blanked integration) to see how bad/pervasive the GPS is.

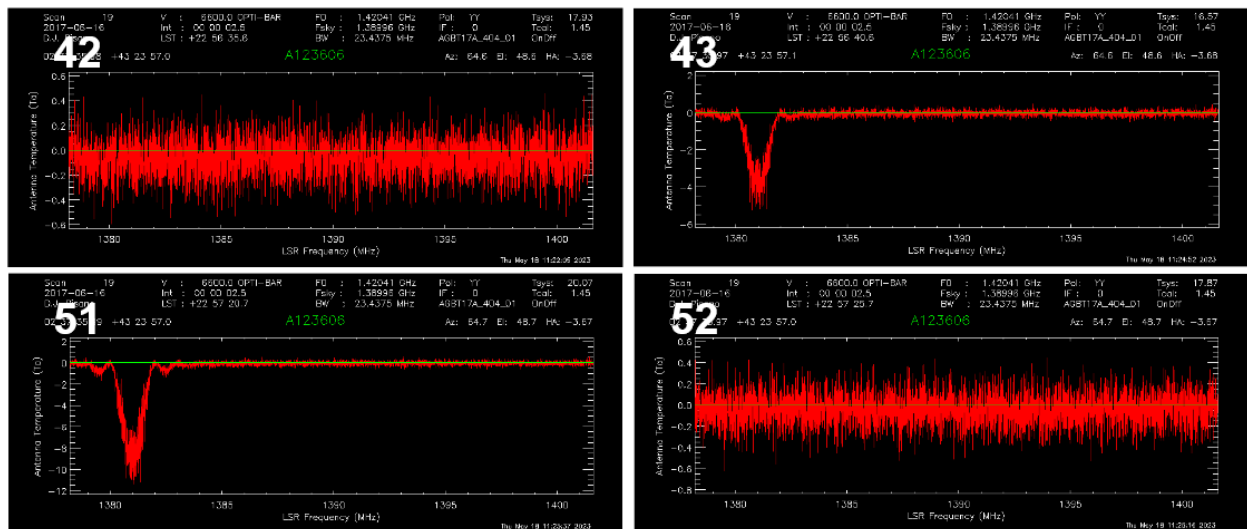
```
for i=0,61 do begin & getps, 19, intnum=i, plnum=0 & end
```

This will step through all 60 integrations as fast as your computer can calibrate and plot them. If you want to see it a little slower, you can add a wait statement:

```
for i=0,61 do begin & getps, 19, intnum=i, plnum=0 & wait, 0.3 & end
```

From this, we can see there is only a portion in the latter half of the OFF scan that is blocked by RFI. Stepping through the integrations manually, we can see the trouble starts in integration #43 and ends at integration #51.

```
getps,19, intnum=42
getps,19, intnum=43
getps,19, intnum=51
getps,19, intnum=52
```



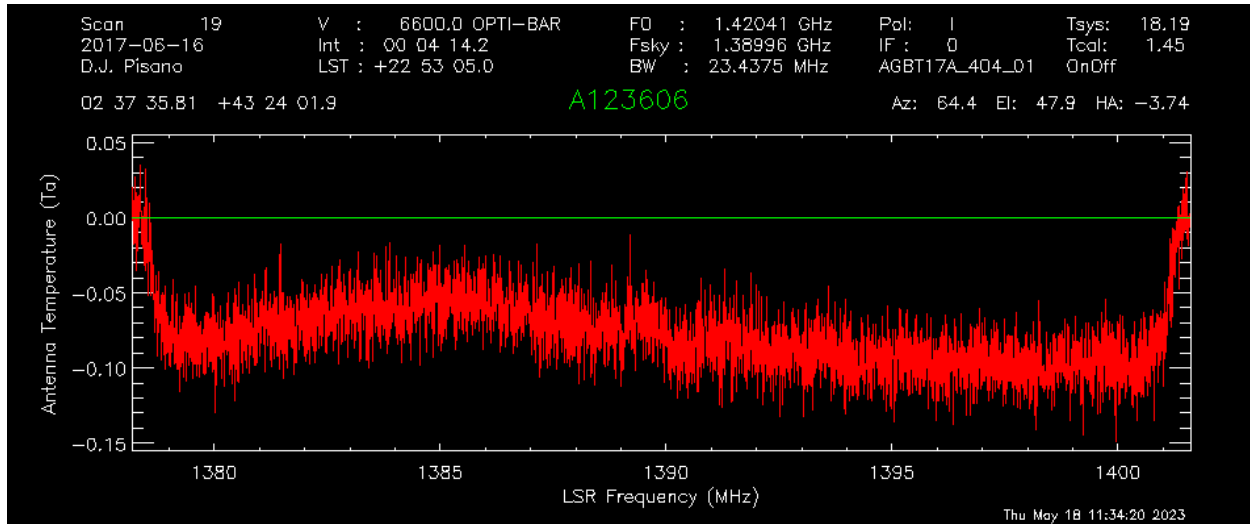
So let's accumulate all the clean integrations for both polarizations, and see if there's any HI detection. Keep in mind

the IDL for loops are inclusive on both ends.

```

sclear
for i=0,42 do begin & getps, 19, intnum=i, plnum=0 & accum & end
for i=0,42 do begin & getps, 19, intnum=i, plnum=1 & accum & end
for i=52,60 do begin & getps, 19, intnum=i, plnum=0 & accum & end
for i=52,60 do begin & getps, 19, intnum=i, plnum=1 & accum & end
ave

```

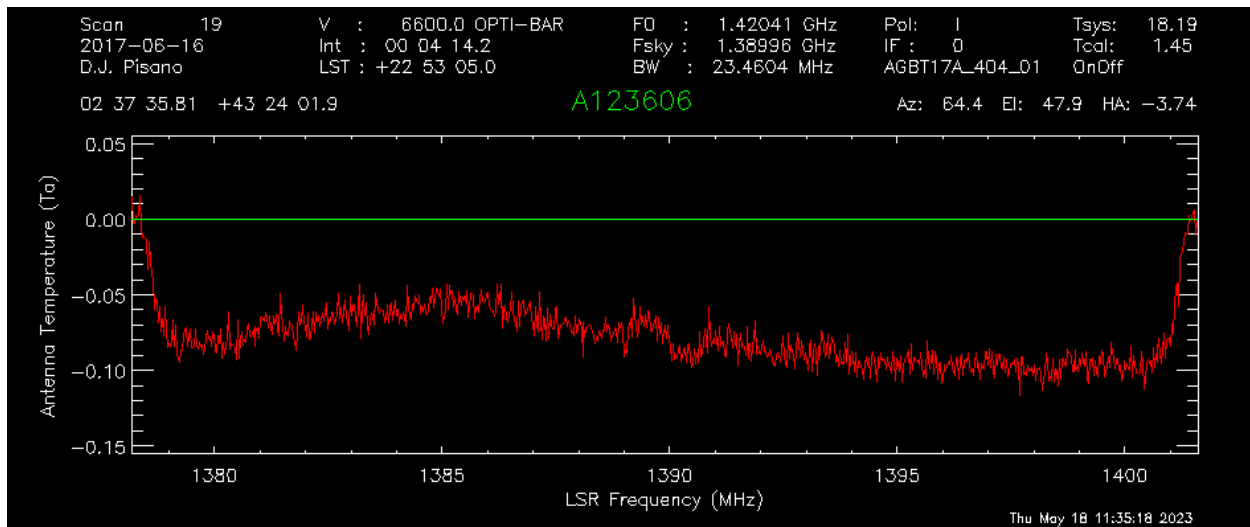


Smooth is:

```

boxcar, 5, /decimate

```

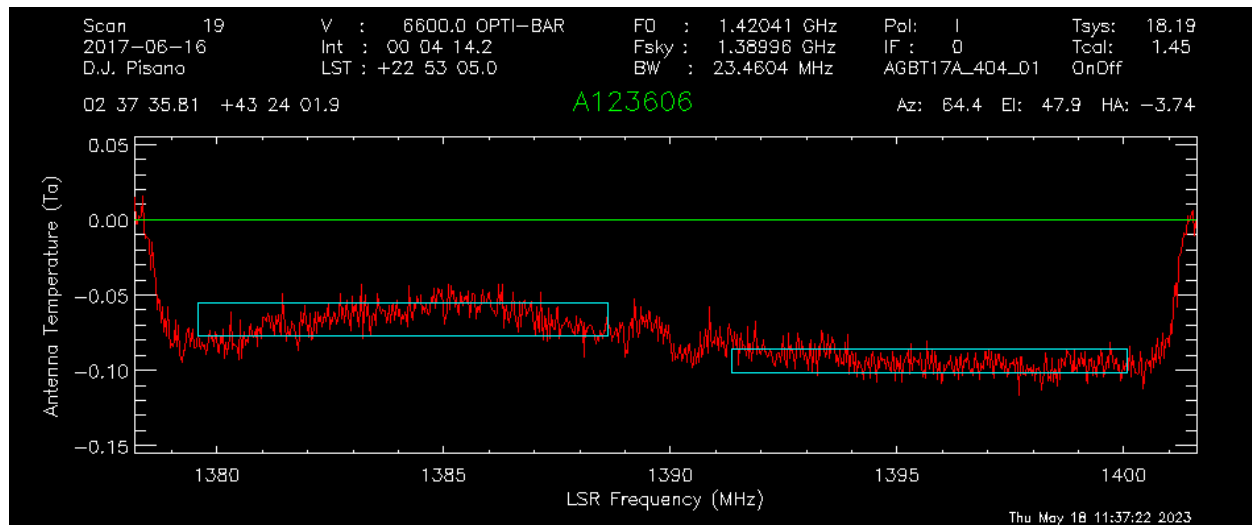


There may be a small detection at 1389.5 MHz. Let's try to fit a baseline - we may have to fit either a 2nd or 3rd order polynomial. First, we will set a checkpoint here by copying the current spectrum to the second data container so we can go back to this step. Then, setregion to everything but the bandpass edges and the possible signal in the middle:

```

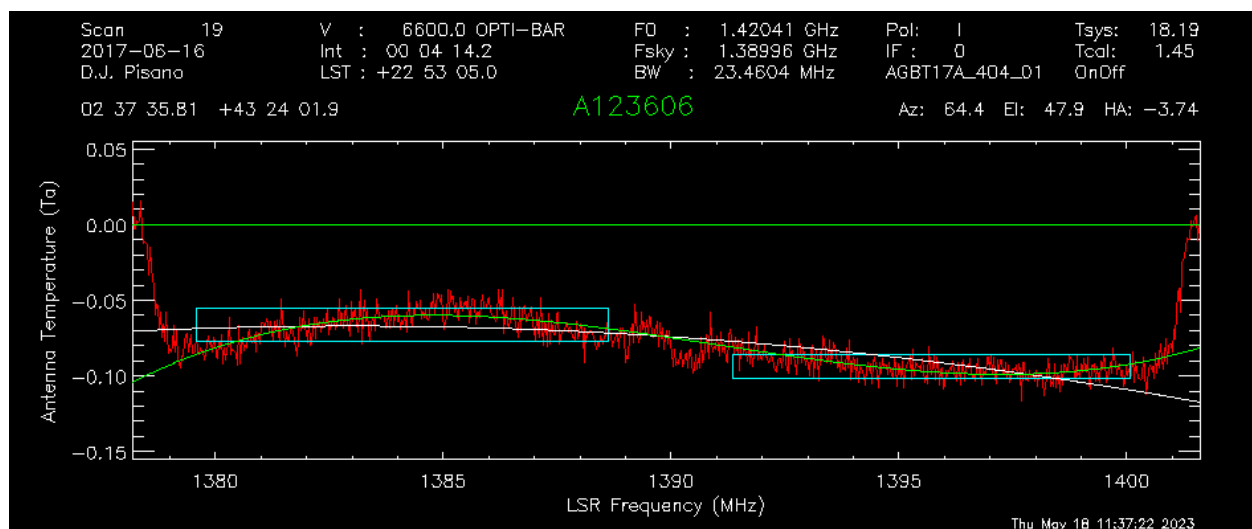
copy, 0, 1
setregion

```



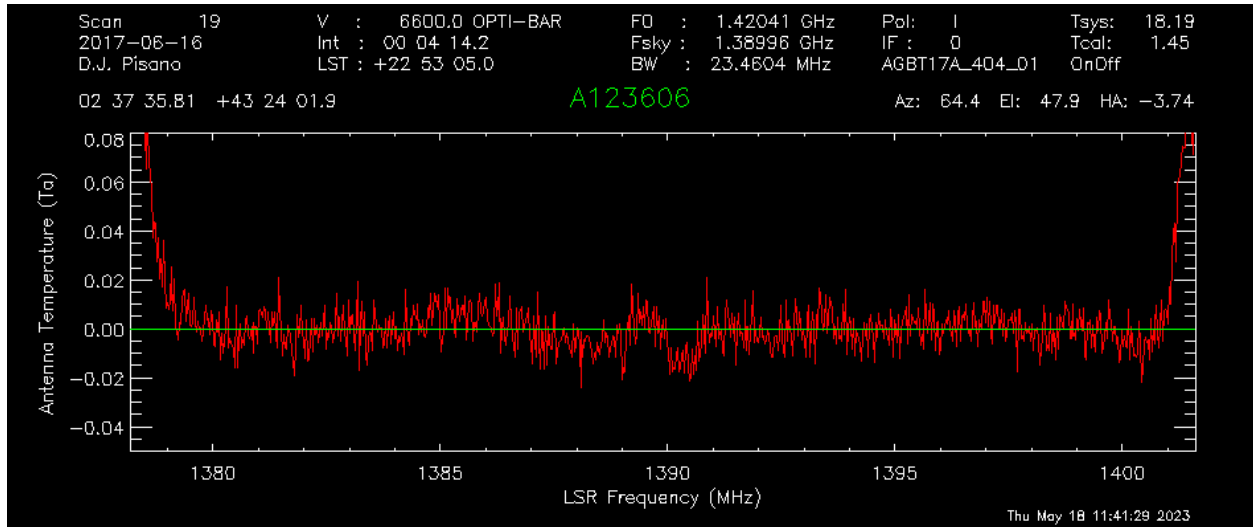
We can trial baseline fits with the “bshape” procedure.

```
bshape, nfit=2
bshape, nfit=3, color=!green
```



The 3rd order fit (green) looks much better than the 2nd order fit (white). Next, the “bsubtract” procedure applies the last fit computed and subtracts it from the data - in this case, our 3rd order fit.

```
bsubtract
sety, -0.05, 0.08
```



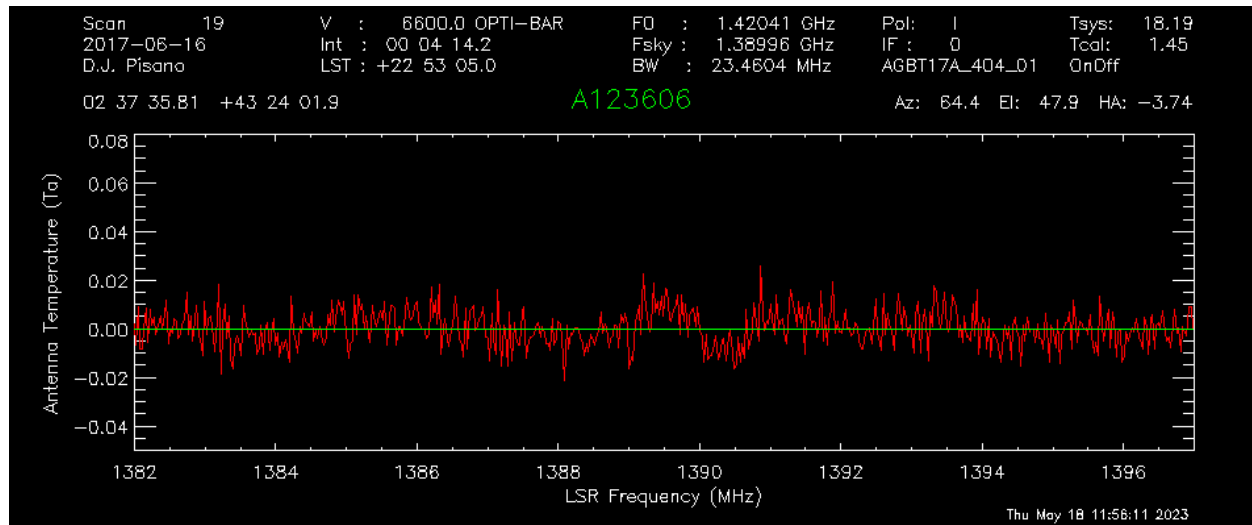
There may be a tiny detection, but the baseline fit is not the best, particularly noticeable in the 1384 - 1389 MHz range. We might go back and see if we can apply a more strict fit, setting the region to be closer in to our possible detection and avoiding more of the bandpass edge.

```
copy,1,0
setregion                ; see image below for the range I chose
bshape, nfit=3
bshape, nfit=4, color=!green
```



The fourth order fit looks to follow that hump at 1385 MHz a little better, so we might pick that despite the large divergence towards the edges of the band.

```
bsubtract
setxy, 1382,1397,-0.05,0.08
```

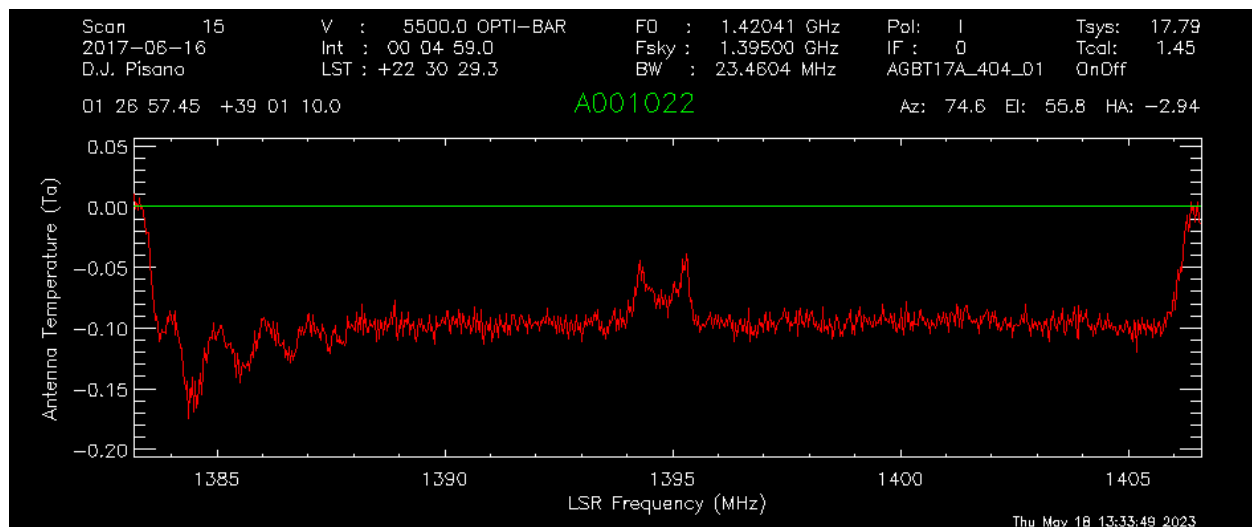


The possible signal looks slightly more significant, but maybe not quite enough to warrant a detection.

### Double Gaussian feature

Now let's turn our attention to scan 15. First, accumulate both polarizations together.

```
sclear
getps,15
accum
getps,15, plnum=1
accum
ave
boxcar, 5, /decimate
```



It does look like there is some GPS-L3 interference on the left side again, we can ignore that since it is far away. Let's grab some info about the spectrum and switch to velocity units.

```
header
```

```

-----
Proj: AGBT17A_404_01   Src   : A001022                   Obs : D.J. Pisano
Scan:      15          RADec : 01 26 57.4 +39 01 10       Fsky:  1.394995 GHz
Int :       0          Eqnx  : 2000.0                   Frst:  1.420406 GHz
Pol :       I          V     : 5500.0      OPTI-BAR       BW  : 23.460   MHz
IF  :       0          AzEl  : 74.621      55.767       delF: 28.610   kHz
Feed:      1          Gal   : 130.437      -23.334       Exp : 299.0    s
Proc:    OnOff        UT    : +10 10 19.0    2017-06-16   Tcal:   1.45    K
Seqn:      1          LST/HA: +22 30 29.3    -2.94       Tsys:  17.79   K
-----

```

The sky frequency,  $\nu_{\text{sky}}$ , is 1395 MHz and the smoothed frequency resolution,  $d\nu$ , is 28.61 kHz, which corresponds to a velocity resolution,  $dv$ , of 6.15 km/s.

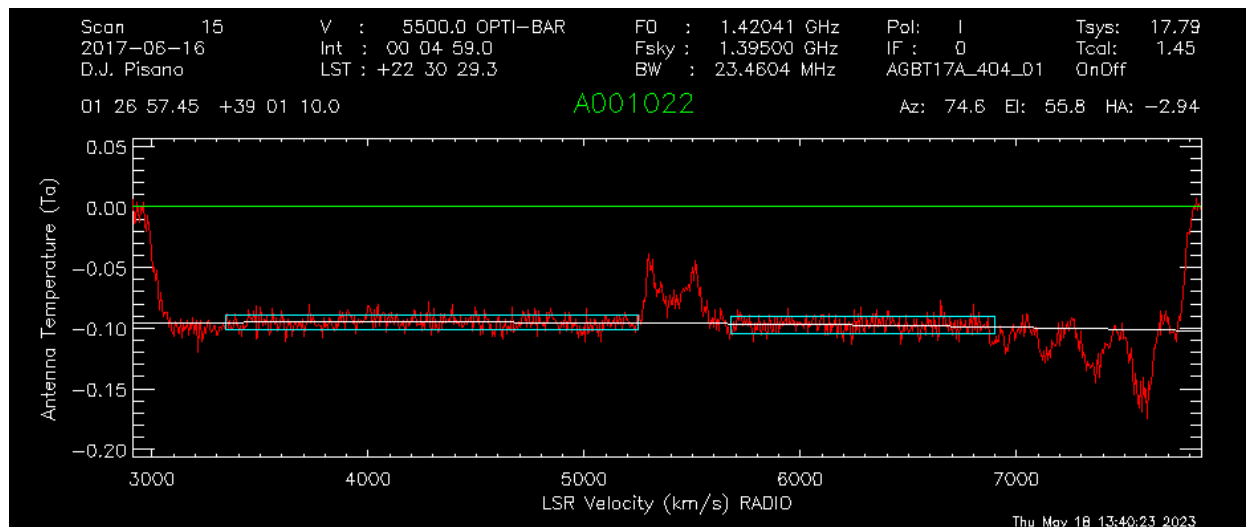
$$dv = c * \frac{(\nu_{\text{sky}} - (\nu_{\text{sky}} - dv))}{(\nu_{\text{sky}} - dv)}$$

There seems to be a slight downward curve in the baseline, so I will fit a 2nd order baseline.

```

velo
setregion
bshape, nfit=2

```



```

bsubtract
sety, -0.05, 0.08

```

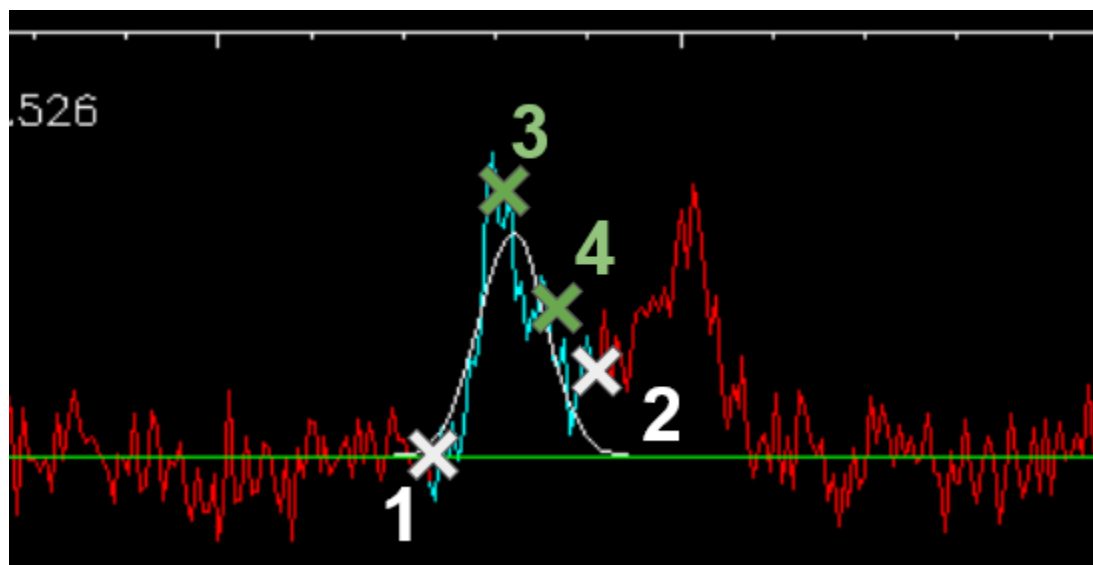
Now we will fit two gaussians to this detection. Since this is a rotating HI galaxy, the actual model should be a two-horn profile, but two gaussians should be enough to fit this. GBTIDL does not have a native two-horn profile fitting procedure.

```

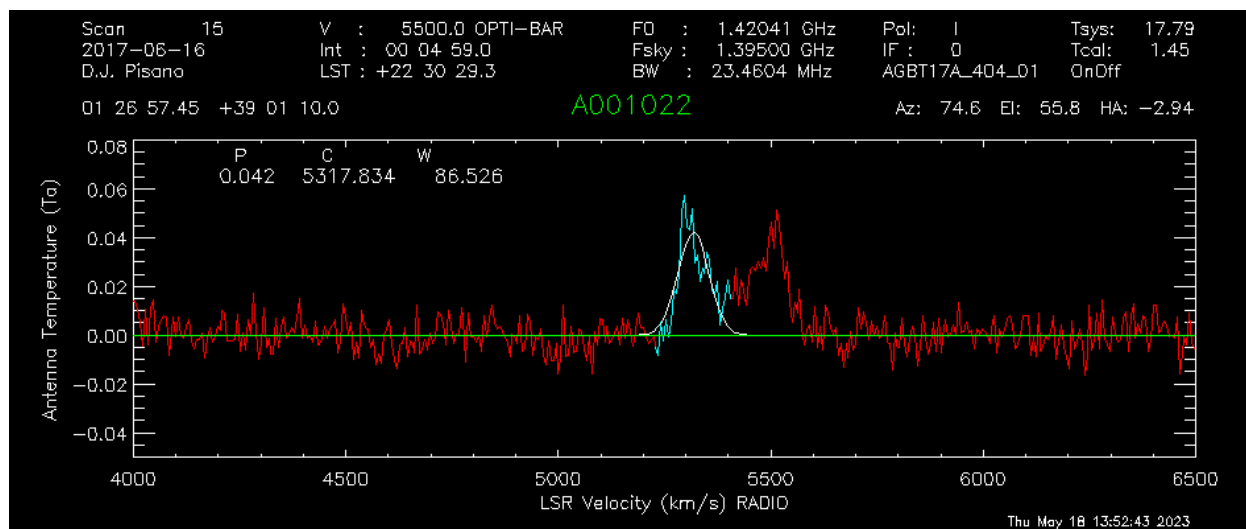
setx, 4000, 6500
fitgauss, modelbuffer=2

```

The program will tell you what to do, but the process involves left clicking the boundaries of the signal, then giving it guesses to model with the middle mouse button. In the zoomed in image below, I left click at the white X marks on either side, then use the middle mouse button to click at the top of the signal, then the half-power point in the order shown:



And finally, a right click tells GBTIDL to model the Gaussian:





```

GBTIDL -> fitgauss,modelbuffer=2
*****
Instructions for fitgauss procedure:
Left Mouse Button: to mark regions to be fit.
Center Mouse Button: to mark initial guesses (center then width)
Right Mouse Button: to calculate and show fits
*****
Limits accepted for region          1
Guesses accepted for gaussian       1
***** Initial Guesses
      G#      Height      Center (km/s)      FWHM (km/s)
Init:      1      0.04912      5308.9331      55.83

***** Fitted Gaussians
      Height      Center (km/s)      FWHM (km/s)
1      0.04194 ( 0.003969)      5317.8335 ( 4.008)      86.53 ( 9.610)
% Program caused arithmetic error: Floating underflow

```

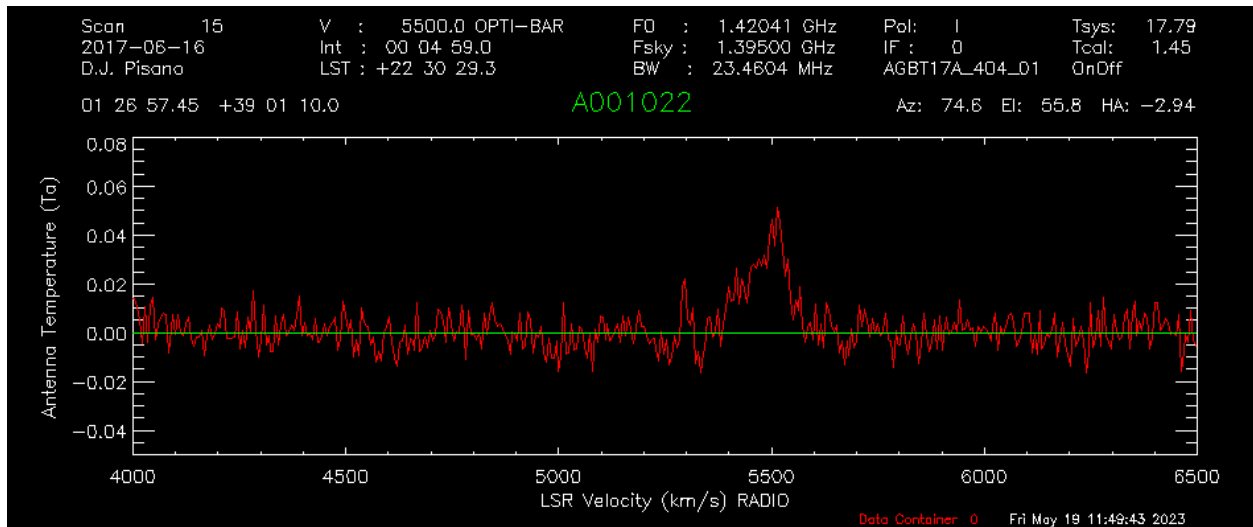
We'll copy the original spectrum to data container 4, then subtract this gaussian out so we can model the other one.

```

copy,0,4
subtract, 4, 2, 0

```

So now the primary data container has the results of DC4 - DC2.

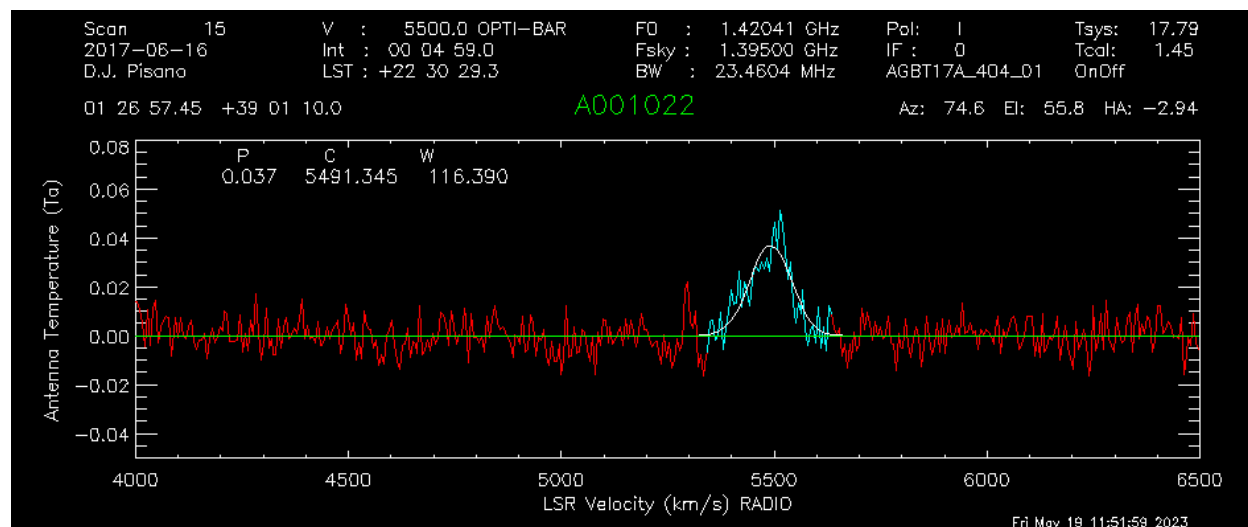


Fit the other Gaussian:

```

fitgauss, modelbuffer=3

```

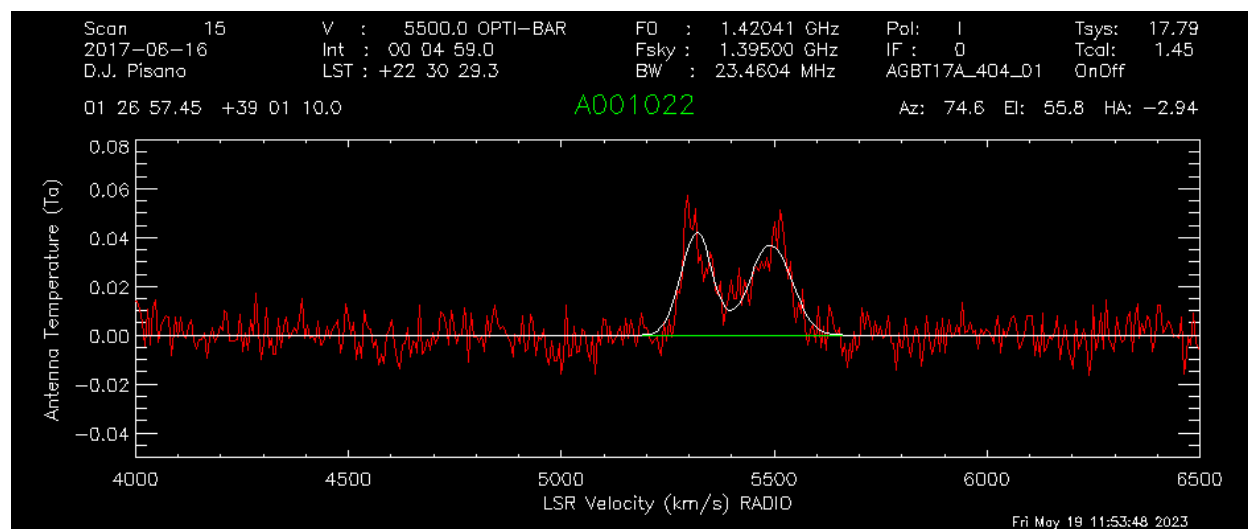


\*\*\*\*\* Fitted Gaussians

	Height	Center (km/s)	FWHM (km/s)
1	0.03694 ( 0.002464)	5491.3452 ( 3.806)	116.4 ( 8.978)

And then show the original spectrum with the two models overlaid:

```
copy, 4, 0
add, 2, 3, 5
oshow, 5
```



## **EXPLANATION MATERIAL**

Big-picture explanations of higher-level concepts. Most useful for building understanding of a particular topic.



## REFERENCE GUIDES

Nitty-gritty technical descriptions of how the GBT works. Most useful when you need detailed information about different GBT components or applications (receivers, observation procedures, data analysis tools).

---

### 4.1 Hardware

#### 4.1.1 Receivers

##### **Argus**

References for Argus (75-115 GHz)

[Argus](#)

##### **MUSTANG-2**

References for MUSTANG-2 (bolometer camera)

[MUSTANG-2](#)

---

### 4.2 Software

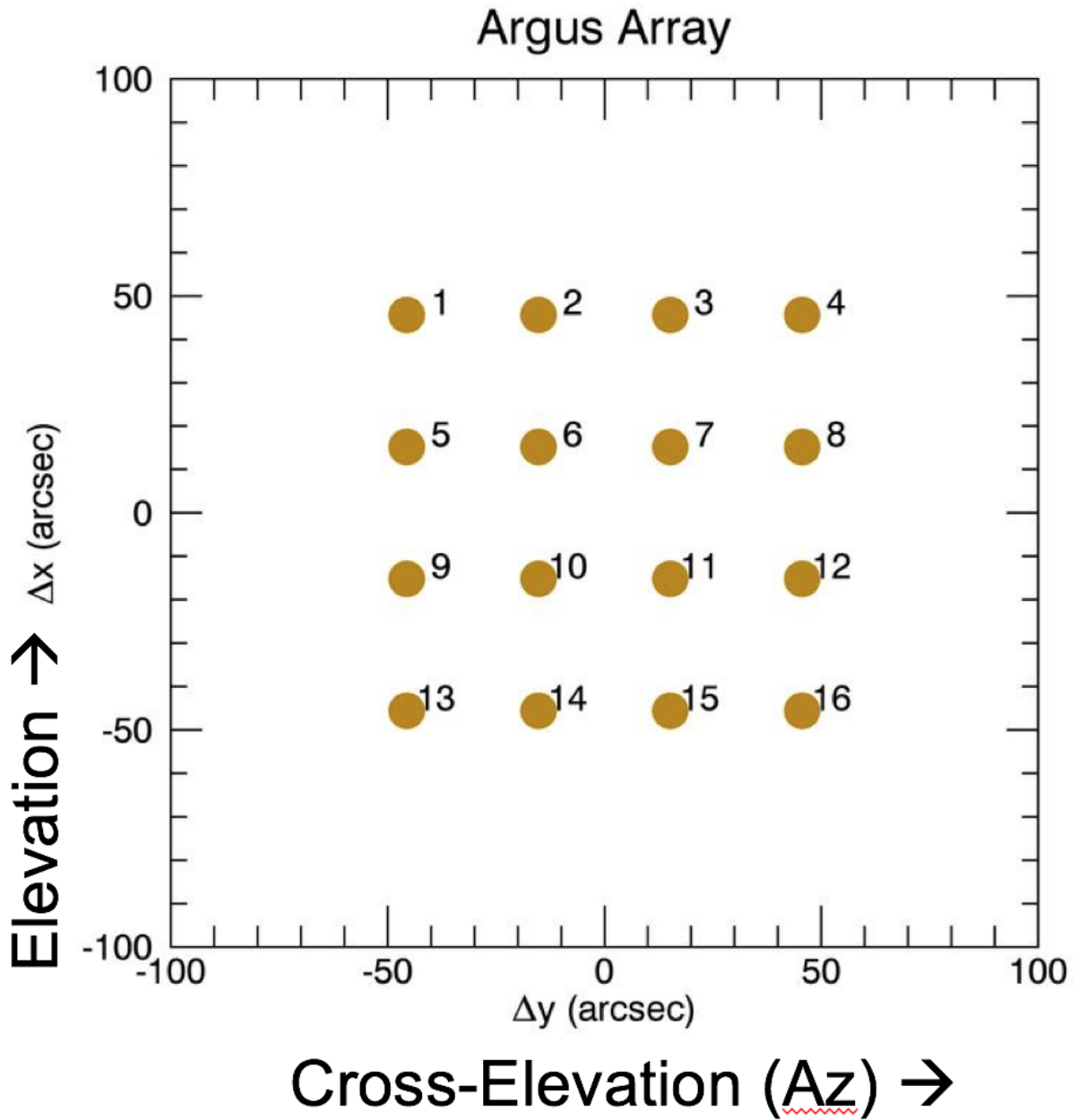
[AstrID Scheduling Block Commands](#)

#### 4.2.1 Argus

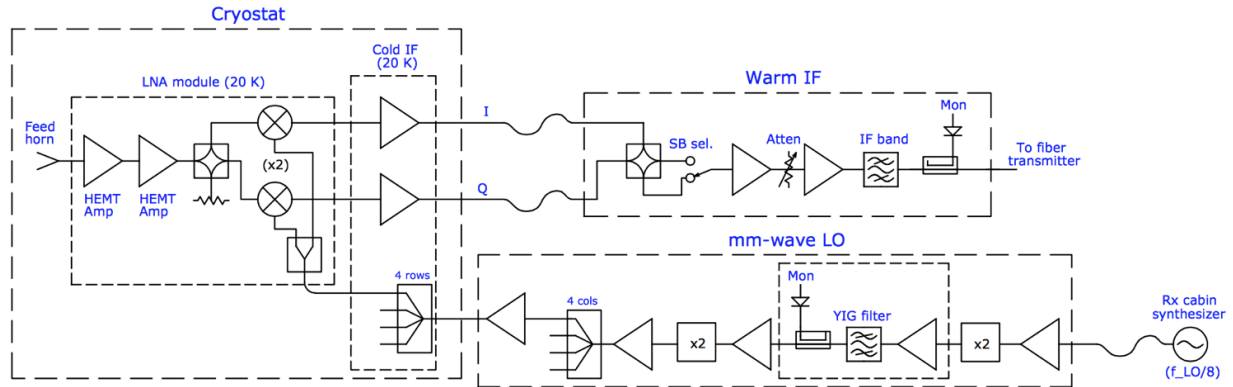
The documentation below is a copy of the content in the Observer Guide. This is still a big mix-match of Tutorial, How-To Guide, and Reference Guide and will need to get sorted out.

#### 4.2.1.1 Introduction

Argus is a 16 element focal-plane-array operating from 74 GHz o 116 GHz. The feeds are configured in a 4x4 array and are each separated by 30.4 arcsec on the sky.



Argus has an absorber vane that can be placed over the entire array for calibration. The 16 Argus beams can be connected to the 16 separate VEGAS channels (VEGAS has 8 bank, each of which have two channels [A1, A2, B1, B2, ..., H1, H2]). Each of the 16 beams only measure 1 polarization (linear, X). Argus uses an IQ-mixer scheme to separate the USB and LSB, and the side-band isolation is approximately 15-20 dB.



The instantaneous bandwidth for Argus is  $\sim 1.5$  GHz, which is similar to the VEGAS spectrometer bandwidth. Users can observe multiple lines simultaneously using the VEGAS sub-banding modes (modes 20-29) for lines separated by less than the  $\sim 1.25$  GHz effective bandwidth of an individual VEGAS bank. For spectral line mapping experiments, Argus is typically configured with each of the Argus beams connected to an individual VEGAS channel. BEams 9-16 use the regular GBT IF system and can be configured with multiple VEGAS banks, or the DCR for pointing, focus and OOF. Beams 1-8 have dedicated IF paths that are only connected to specific VEGAS banks.

For the chopper-vane calibration technique that Argus adopts, the natural temperature scale measures is  $T_A^*$  (GBT Memo 302). This temperature scale has the advantage of correcting for atmospheric attenuation while its derivation is nearly independent of opacity. Users need to take a vane calibration sequence whenever the configuration changes or whenever the IF system is balanced to calibrate the data.

Argus does not have noise diodes.

#### 4.2.1.2 Configuration

Argus uses the standard config-tool software that automatically configures the system based on user input (e.g., frequency and bandwidth). Example Argus configuration files are given in `/home/astro-util/projects/Argus/OBS`. The configuration keywords specific to Argus are the following:

```
receiver    = "RcvrArray75_115"
beam        = "all"                # or list the beams separately, e.g., beams="10,11"
swmode      = "tp_nocal"           # or "sp_nocal"
polarization = "linear"
sideband    = "LSB"                # for best performance: LSB < 112 GHz, USB >= 112 GHz
```

#### 4.2.1.3 Observing

The observing strategies for Argus are similar to those presented for the 4mm W-Band receiver. To maximize the telescope efficiency for targets smaller or similar in size to the beam ( $\sim 8$  arcsec), observations should be carried out during the night under stable thermal conditions. Depending on the science goal, successful daytime observations are possible for extended sources, where accurate beam shapes are not as crucial. Example Argus observing scripts are located at `/home/astro-util/projects/Argus/OBS`. The recommended observing procedures are the following

1. Startup Astrid and go online (with control of the telescope, when given permission by the operator)
2. Run the `argus_startup` script. This script checks the instrument status, turns ON the instrument if it is currently off and pre-configures Argus for the default 90 GHz parameters.
3. At the start of the observing session, run an AutoOOF to optimize the surface, unless the exact beam shape is not important for your science goals. This procedure will correct the surface for the current thermal conditions

and derive the initial pointing and focus corrections. For AutoOOF it is recommended to use the brightest point source in the sky between 25-80 degrees elevation. If the Ka-Band receiver is available, run the AutoOOF at Ka-Band instead of Argus for more accurate surface corrections. When running AutoOOF with Argus, it is recommended to avoid using the `calSeq=False` keyword, so the data will be properly calibrated in the fitting for the surface model. The Astrid Observation Management Log will report the system temperatures on the sky from the initial vanecal scans. The same Astrid command “AutoOOF(source)” can be used for any of the receivers that use AutoOOF (i.e., Ka, Q, W-Band, Argus, Mustang-2) and the software will adopt the appropriate defaults for each band.

- After the AutoOOF solutions are applied, run a point and focus with Argus to confirm the telescope collimation offsets.
- For Argus, run AutoPeakFocus() with a bright pointing source (>1.5 Jy) within ~ 30 degree of the target region; brighter sources are better than closer sources, since the GBT gregorian pointing model is fairly accurate. Choose a frequency that is the approximate frequency of your science frequency since the YIG filter system can take time to adjust to large frequency shifts. For the best science results, AutoPeakFocu() should be run every 30-50 minutes depending on conditions (point more often during the day and after sunrise and sunset). Avoid pointing in the keyhole (el>80deg). Since the elevation pointing offsets can be larger than those observed typically in azimuth, use the `elAzOrder=True` keyword to observe the elevatin Peak scans first. An example pointing command showing the usage of the frequency, `calSeq`, and `elAzOrder` keyword is `AutoPeak(source, frequency=90000., calSeq=False, elAzOrder=True)`.
- If pointing is problematic with Argus, e.g., observations during the day, or in periods of marginal weather, or in cases where the pointing source is too weak, observers can point and focus in X-band and use these telescope corrections for their Argus observations. Also, if there are no nearby bright sources to point with Argus and the telescope is at slow slew rate (at cold temperatures), it can be faster to switch receivers for pointing than to slew far away and point with Argus. Pointing and focus using Argus requires special attention, and users should not blindly accept the default solutions provided by the software system. Users can enter solutions manually as needed as discussed in Section~ref{sec:gfmSENDcorrections}. If you are unsure of the Argus pointing results, point in X-band (which is adjacent to Argus in the turret).
- After configuration and balancing VEGAS for science observations, check the power levels in the system. The VEGAS levels should be ~ -20+/i 3 dBm.

File Managers Help																			
Bank	Power	Mode	Filter BW MHz	Noise Source	Tone	Polarization	Subbands	Int Time sec	Switching Source	State	SubSys Select								
A	-19.88	1	1400	off	tone	self		1	internal	Ready		More Info...							
	-19.51																		
B	-20.04	1	1400	off	tone	self		1	internal	Ready		More Info...							
	-19.84																		
C	-19.89	1	1400	off	tone	self		1	internal	Ready		More Info...							
	-21.64																		
D	-18.88	1	1400	off	tone	self		1	internal	Ready		More Info...							
	-20.14																		
E	-19.76	1	1400	off	tone	self		1	internal	Ready		More Info...							
	-19.20																		
F	-20.68	1	1400	off	tone	self		1	internal	Ready		More Info...							
	-19.93																		
G	-20.40	1	1400	off	tone	self		1	internal	Ready		More Info...							
	-18.25																		
H	-33.19	1	1400	off	tone	self		1	internal	Ready		More Info...							
	-19.25																		
Phase Tables: Requested... Actual... Master Switching Bank A Blanking Source internal VEGAS Power Monitor...																			
VEGAS		Status	clear	State	Ready		M	A	T	Locked		AutoPrepare	Prepare	Quit					
Bank A		Status	clear	State	Ready		M	A	T	Bank E		Status	clear	State	Ready		M	A	T
Bank B		Status	clear	State	Ready		M	A	T	Bank F		Status	clear	State	Ready		M	A	T
Bank C		Status	clear	State	Ready		M	A	T	Bank G		Status	clear	State	Ready		M	A	T
Bank D		Status	clear	State	Ready		M	A	T	Bank H		Status	clear	State	Ready		M	A	T

- The target power levels in the IF rack (for beams 9-16) are 1.5 Volts. The YIG LO power level going into the



instrument should range from  $\sim 0.2$ – $0.6$  Volts (above 84 GHz). The power coming out of the warm electronics of Argus should read about  $\sim 1.0$ – $1.4$  for beams 1-8 and  $\sim 0.5$ – $0.9$  for beams 9-16 (under the TP column of the WIF section of the Argus pyCLEO window).

## Argus Cleo Window

**Annotations:**

- Turn manager off/on under Manger tab
- Vane state: obs vs cal
- TP out of Argus should be 1-1.4 for Ch1-8 and 0.5-0.9 for Ch9-16 when configured
- Click Prepare to get current instrument parameters
- LO power should be  $>0.2$  after configuration
- Click to unlock and issue commands
- LNA and CIF need to be on to use Argus
- Beam status green is good, red indicates there may be a problem

**GUI Sections:**

- Commands:** Startup, Vane Cal, Vane Obs, Shutdown, Reboot, Prepare, Manger Status.
- Vane:** State (obs), angle (3.37), temp [C] (23.5), current (0.0).
- Cryo:** 20K Cold Head (13.700), 20K Plate (23.700), 70K Cold Head (50.400), 70K Plate (74.700), Card Cage [K] (24.700), Pressure (Torr) (1.000e-05).
- Power Supplies:** Amp +15 V (15.14), Amp -15 V (-14.99), Digital +5 V (4.79), Cold IF V (1.78), Drains V (4.84), Chassis T [C] (34.5).
- WIF:** Table with columns Ch, TP, Atten, SB, Card T. Rows 1-16.
- Status codes:** LNA (on), CIF (on), system (32), power (0.0), IF power (16384.0), thermal (0.0), LNA, mixer bias errors (0.0).
- Beam Status:** 4x4 grid of colored squares (green for good, red for problem).
- Argus GUI log messages:** Log of system events.

9. Users must run the `argus_vanecal` procedure to calibrate the data (located in `/users/astro-util/projects/Argus/OBS/argus_vanecal`) after each configuration and/or balance for observations that need to be calibrated. The vane calibration is stable over longer periods than is needed for pointing and focusing, so only one `argus_vanecal` procedure is required for each set of VEGAS observations between the pointing and focus observations. Under stable temperature conditions, the vane calibration is consistent over several hours while it is recommended to point and focus every 30–50 minutes for Argus.
10. It is best to observe similar frequencies together in time since it can take a few minutes for the YIG system to adjust to large frequency jumps. If you need to switch by a large amount in frequency (e.g.,  $>4$  GHz), configure and wait a couple of minutes before observing. If the YIG LO power is low after a large frequency shift (e.g.,  $<0.2$ ), re-configure again.
11. Only beams 9-16 that go through the GBT IF Rack can be configured with the DCR. All 16 beams can be configured with VEGAS using the 8 dedicated optical fibers for Argus beams 1-8.
12. Beam-8 has very little sideband rejection and will show higher noise when using the LSB at high frequency (e.g., when the oxygen atmospheric line is in the USB).
13. The “Auto” procedures will run `vanecal` observations by default. For pointings/focus scans that do not need to be calibrated, observers can use the `calSeq=False` keyword, e.g., `AutoPeak(source, frequency=90000., calSeq=False)`. The use of the `calSeq=False` keyword will save a minute or two of time. However, it is recommended to run the `vanecal` observations while pointing between science blocks of observations in order to track the performance of the system from the calibrated peak scans. If your frequency is not specified, the default frequency for the Argus Auto procedures is 86000 MHz.
14. Beam 10 is the default signal beam and beam 11 is the default reference beam for pointing, focus, and OOF observations.
15. The instrument performance using VEGAS can be checked by running the `vanecal.pro` procedure within

GBTIDL. Example Argus data reduction scripts are located at /users/astro-util/projects/Argus/PRO. The vanecal.pro routine uses the getatmos.pro procedure which derives the opacities and atmospheric conditions from the Green Bank Weather database.

16. For absolute calibration carryout PEAK scans after applying good surface, pointing, and focus corrections for a source of known flux density (e.g., ALMA source catalog (<https://almascience.eso.org/sc/>)). The ALMA calibrator catalog can also be used to check the strength of your pointing/focus source. The calibration methods and performance of the telescope are presented in GBT Memo #302.

#### 4.2.1.4 Monitoring and Diagnostics

The Argus pyCleo page can be used to monitor the status of the instrument.

## Argus Cleo Window

**Annotations:**

- Turn manager off/on under Manger tab (pointing to the Managers tab in the top menu)
- Vane state: obs vs cal (pointing to the Vane State dropdown)
- Click Prepare to get current instrument parameters (pointing to the Prepare button)
- LO power should be >0.2 after configuration (pointing to the LO power field)
- Click to unlock and issue commands (pointing to the Locked button)
- LNA and CIF need to be on to use Argus (pointing to the LNA and CIF status fields)
- Beam status green is good, red indicates there may be a problem (pointing to the Beam Status grid)
- TP out of Argus should be 1-1.4 for Ch1-8 and 0.5-0.9 for Ch9-16 when configured (pointing to the TP field in the WIF table)

**WIF Table:**

Ch	TP	Atten	SB	Card T
1	1.286	7	lower	29.5
2	1.209	7	lower	28.5
3	1.154	6	lower	28.5
4	1.193	6	lower	28.0
5	1.268	8	lower	29.5
6	1.265	7	lower	30.0
7	1.270	4	lower	30.5
8	1.219	4	lower	30.5
9	0.761	18	lower	28.0
10	0.761	20	lower	28.5
11	0.744	19	lower	28.5
12	0.692	24	lower	27.0
13	0.742	21	lower	29.5
14	0.741	24	lower	30.0
15	0.669	21	lower	30.0
16	0.683	20	lower	30.5

**Beam Status Grid:**

1	2	3	4
5	6	7	8
9	10	11	12
13	14	15	16

**Argus GUI log messages:**

```

2019-09-09 10:04:36.967506: PyCLEO Argus panel started.
2019-09-09 10:11:43.042905: INFO -- panel is currently locked and will not accept commands.
2019-09-09 10:11:45.758639: Lock status changed to False.
2019-09-09 10:11:52.212183: manager requested to be off
2019-09-09 10:11:55.467415: manager requested to be on
2019-09-09 10:12:26.908613: startup requested
2019-09-09 10:12:45.783369: Lock status changed to True.

```

This tool can be started from the CleoLauncher under the Receivers tab labeled “RcvrArray75\_115”. The pyCleo can be used for running basic instrument commands, such as startup or the vane control. Before issuing a command, you must unlock the CLEO window by clicking the green “locked” button to unlocked (red). The instrument parameters shown by pyCleo for Argus are updated after a configuration, at the start of each scan during observing, and every 30 minutes when not observing. Updated instrument values can be obtained by issuing a “prepare” command, which is done under the top managers tab (prepare) or the prepare button under the reboot button.

The Beam Status buttons are color coded, where green means the signal associated with the beam is good and red indicates a potential issue with the beam. If a beam is read, the data may still be usable depending on the system temperature associated with the beam.

The Vane state is “obs” when Argus feeds are looking at the sky (with an angle of ~ 3.4) and “cal” when the vane is covering Argus during the Vane calibration scan (angle of ~ 1.6).

The LNA and CIF (low noise amplifiers and cold IF electronics) need to be in the on state to carry out observations. After configuration, the YIG LO power (listed under the YIG section) should be ~ 0.2-0.6 V. The total power levels of the WIF (warm IF electronics) should read ~ 1.0-1.4 for beams 1-8 and ~0.5-0.9 for beams 9-16 after configuration and while observing.

#### 4.2.1.5 IF Routing

The mapping between the VEGAS channels, Converter Modules, IF channels, and the VEGAS beams is shown below

### Mapping Argus Beams to VEGAS and IF Channels

VEGAS Bank	VEGAS (J)	Argus Beam	Converter Module CM	IFrack Optical Driver OD	Dedicated Fibers
A1	1	9	1	1	-
A2	2	11	5	3	-
B1	3	10	2	2	-
B2	4	12	6	4	-
C1	5	1	3	-	1
C2	6	3	7	-	3
D1	7	2	4	-	2
D2	8	4	8	-	4
E1	9	13	9	5	-
E2	10	15	13	7	-
F1	11	14	10	6	-
F2	12	16	14	8	-
G1	13	5	11	-	5
G2	14	7	15	-	7
H1	15	6	12	-	6
H2	16	8	16	-	8

Observers should verify the VEGAS power levels ~-20+/-3 dBm via the VEGAS Cleo window.

File Managers Help													
Bank	Power	Mode	Filter BW MHz	Noise Source	Tone	Polarization	Subbands	Int Time sec	Switching Source	State	SubSys Select		
A	-19.88	1	1400	off	tone	self		1	internal	Ready		More Info...	
	-19.51												
B	-20.04	1	1400	off	tone	self		1	internal	Ready		More Info...	
	-19.84												
C	-19.89	1	1400	off	tone	self		1	internal	Ready		More Info...	
	-21.64												
D	-18.88	1	1400	off	tone	self		1	internal	Ready		More Info...	
	-20.14												
E	-19.76	1	1400	off	tone	self		1	internal	Ready		More Info...	
	-19.20												
F	-20.68	1	1400	off	tone	self		1	internal	Ready		More Info...	
	-19.93												
G	-20.40	1	1400	off	tone	self		1	internal	Ready		More Info...	
	-18.25												
H	-33.19	1	1400	off	tone	self		1	internal	Ready		More Info...	
	-19.25												
Phase Tables: Requested... Actual... Master Switching Bank A Blanking Source internal VEGAS Power Monitor...													
VEGAS		Status	clear	State	Ready		Locked		AutoPrepare	Prepare	Quit		
Bank A		Status	clear	State	Ready		Bank E		Status	clear	State	Ready	
Bank B		Status	clear	State	Ready		Bank F		Status	clear	State	Ready	
Bank C		Status	clear	State	Ready		Bank G		Status	clear	State	Ready	
Bank D		Status	clear	State	Ready		Bank H		Status	clear	State	Ready	

As an example shown by Figure fig:argusvegas, the VEGAS channel H1 is  $-33$  dBm which is too low to yield useful data. The H1 VEGAS bank corresponds to VEGAS channel J15, converter module CM12, dedicated fiber “6”, and Argus beam 6. In this example, the data associated with Beam-6 from Argus would be bad and would show non-physical system temperatures.

#### 4.2.1.6 Troubleshooting Guide

The Argus CLEO window can be used to troubleshoot Argus issues, which can be launched from cleo under the Receivers tab by selecting “RcvrArray75-115”

- To control the instrument unlock the system by selecting the green button “Locked” on left to unlock the window (it turns red when unlocked).
- To get the current status of the instrument click the “Prepare” button which is the last Command listed in the upper left.
- When done, lock the system to avoid accidentally issuing a command by clicking the red unlocked button to green (locked).
- Confirm that the CIF and LNA are both on. If off, run the Argus startup script.
- Make sure the vane is in the desired position (e.g., obs for looking at the sky). If the vane is “stalled” or in an unknown state, click the Vane Obs button to move the vane to the obs position. If the vane is not already in the obs position, a configuration will also command the vane to the obs position. If the vane fails to move have the operator contact a support scientist.
- Confirm there is LO power from YIG after configuration.
- The status of the instrument is checked before each scan, and the scan will be aborted if there is not enough YIG LO power, or for other major issues. If the YIG power is too low, or the WIF power levels are low, and/or if one or more of the beam Status colors are red, reconfigure. If one or more the beams are bad, observations with the remaining beams can continue, but one must have sufficient YIG LO power to carry out Argus observations.

- Sometimes the GBT M&C system will report old Argus errors when everything is working. You can ignore and continue observing, or try to clear the lingering error messages with the following procedure.
  - “Prepare” to retrieve updated instrument parameters which may clear the error.
  - If the error persists, turn the manager off and back on from the Managers pull-down menu at the top of the CLEO window. Click off, wait a couple of seconds, and then click on.
  - This usually clears the error messages. Sometimes the error message(s) need to be cleared manually from the messageMux system by the software group.
- If Argus communication errors occur (e.g., Netburner time out error), then the recent commands given to Argus may not have been done and you may need to re-configure and re-issue your observing script. Within the Argus CLEO window, click the “Prepare” button to collect the current state of the instrument. If the LNA/CIF are off under the Status Codes, run the Startup script and then reconfigure. Turn the manager off and back on again to clear the Netburner errors.
- If Argus is in a “fault” state after configuration and you are unable to collect data after multiple attempts then:
  - Turn manager off and back on again (under the Managers tab at the top of the Argus CLEO window) and reconfigure.
  - If cycling the manager does not work, have the operator restart turtle and/or “grail” and reconfigure.
  - If neither of the above work, then have the operator contact a support scientist.

#### 4.2.1.7 Data Reduction

Argus is calibrated on the T<sub>A</sub>\* antenna temperature scale. Observations need to run a set of vanecal observations for each set of science data. For absolute flux calibration, a source of known flux density should be observed. The ALMA Calibrator Source Catalog has an extensive record of the flux density histories for many of the bright 3mm point sources (<https://almascience.eso.org/sc/>). By using ALMA flux density values as a function of time, ~10% absolute calibration uncertainties can be obtained for Argus data. The methods and equations for calibrating Argus data are presented in GBT Memo #302.

The standard GBTIDL scripts (getps, getnod, getfs) do not work with Argus data, since these assume a noise diode for calibration. Example Argus scripts for the reduction of spectral line data can be found at /home/astro-util/projects/Argus/PRO. Users can use the vanecal.pro procedure within GBTIDL to derive the T<sub>cal</sub> value and the system temperatures for each of the Argus beams. Frequency switched data can be reduced using argus\_fsw.pro, and position switch data can be reduced using argus\_onoff.pro.

#### 4.2.1.8 Documentation

- **Argus Web Page:** <http://www.gb.nrao.edu/argus/>
- **GBT Calibration Memo:** [https://library.nrao.edu/public/memos/gbt/GBT\\_302.pdf](https://library.nrao.edu/public/memos/gbt/GBT_302.pdf)
- **Argus configuration and observing scripts which are used in Astrid:** /home/astro-util/projects/Argus/OBS
  - **argus\_startup:** Script that turns Argus on if it is not already on. The script configures the instrument with the default settings. This is run at the start of an Argus observing session.
  - **argus\_vanecal:** Script that runs the vanecal observations. It observes with the vane over the array as well as blank sky scan with a default 6 arcmin offset from the commanded position to avoid a bright calibrator object. If observing the Moon or a very bright extended continuum source, one can use the argus\_vanecal\_bigoffset2 or argus\_vanecal\_bigoffset to observe blank sky.
  - **autooof:** Script that runs the AutoOOF observations. The sources listed are the brightest W-band sources in the sky.

- **autopeak\_focus**: Script that runs pointing and focus observations. The pointing observations are run first, and then the script issues a break to allow the user to enter the solutions manually into the system before the focus scan.
- **autopeak\_calibrate**: Calibration script to run on a known calibrator to compute the aperture and main-beam efficiency of the telescope after good pointing and focus corrections have been applied.
- **argus\_config\_example**: Example total power configuration (tp\_nocal) for Argus.
- **mapRA**: Example frequency switching (sp\_nocal) observing script for a RA/Dec map.
- *GBTIDL reduction scripts*: /home/astro-util/projects/Argus/PRO
  - **getatmos.pro**: Script that returns the atmospheric opacity and effective atmospheric noise and temperature for a specific time and frequency from the Green Bank Observatory weather database. This needs to be run on a Green Bank computer since using special code that only runs locally.
  - **vanecal.pro**: Script that computes the system temperature for each of the Argus beams from the vanecal observations. The script uses getatmos.pro to compute the Tcal value (see GBT Memo#302).
  - **argus\_fsw.pro**: Script that calibrates a frequency switched observation.
  - **argus\_onoff.pro**: Script that calibrates a position switched observation.
- **ALMA Source Catalog**: <https://almascience.eso.org/sc/>

## 4.2.2 MUSTANG-2

Below are various technical aspects and references that an observer or a MUSTANG-2 team member may want to know about MUSTANG-2.

---

### MUSTANG-2 Overview

A brief description of MUSTANG-2 and its capabilities.

[See Details](#)

### Instrument Team

A list of the MUSTANG-2 instrument team members.

[See Members](#)

#### 4.2.2.1 MUSTANG-2 Overview

MUSTANG-2 is a 223-feedhorn bolometer (continuum) camera with a bandpass from 75 to 105 GHz on the GBT.

MUSTANG-2 is inherently a mapping instrument and observes by scanning “on-the-fly” (OTF). Our most common scan pattern is that of a Lissajous Daisy that can observe objects from our resolution (9) to several arcminutes in size (more details on [mapping](#)). For scans of ~a degree, we have switched to a raster scan. If you would like to consider another mapping strategy, please contact a member of [the MUSTANG-2 instrument team](#) to inquire about it. The instantaneous field of view (FOV) is just over 4 arcminutes (4.2). For full bandpass details see [this page](#).

**MUSTANG-2 was commissioned on the GBT in spring 2016 (see development [timeline](#)), and has been offered for observations on a shared risk basis, in collaboration with the instrument team, since the 2018A GBO proposal call. Several features distinguish it from its predecessor, MUSTANG:**

- A new, microstrip-coupled detector design yields higher sensitivity and less susceptibility to environmental microphonics.

- Detectors are feedhorn coupled, with the sum of two linear polarizations measured by a single TES per feed.
- The instantaneous field of view is 4 arcminutes (vs 42 arcseconds for MUSTANG)
- The receiver design incorporates a tilted refrigerator and receiver rotator, resulting in much lower dependence of cooling performance on telescope elevation.
- The detector readout is the first astronomical use of microwave resonators to multiplex TES bolometers.

MUSTANG-2 has been developed by a collaboration including the University of Pennsylvania, NIST, NRAO, the University of Michigan, and Cardiff University. All critical MUSTANG-2 systems have already been proven in operation on the GBT in early 2015 during an engineering run using a partially populated version of the receiver (“MUSTANG-1.5”, which had 64 populated feed horns).

Many *papers* have been published using both MUSTANG and MUSTANG-2.

### The Status of Future GBT Open-Skies Observing

In order to cope with an evolving funding landscape the GBT is in the process of moving to a model which relies upon a larger fraction of private and collaborative (“pay-to-play”) partnerships. It is expected that a significant fraction of GBT time, depending on the future level of NSF funding, will remain open under something like the current proposable, open-skies arrangement; however the mix of observing capabilities that is available under this arrangement is TBD. **If you are interested in forming or contributing to such a partnership to ensure continued access to MUSTANG-2 on the GBT, please contact the director of the Green Bank Observatory, Jim Jackson.**

#### 4.2.2.2 MUSTANG-2 Instrument Team

The following are members of the MUSTANG-2 Instrument Team and should be included on all proposals:

- Simon Dicker
- Charles Romero
- Brian Mason
- Mark Devlin
- Tony Mroczkowski
- Jonathan Sievers
- Craig Sarazin
- Emily Moravec
- Luca Di Mascolo
- Tanay Bhandarkar
- Karen Perez Sarmiento
- Saianeesh Haridas
- Jack Orlowski-Scherer
- Julia Robe

## Contact Details

Please contact Simon Dicker at [simon.dicker@gmail.com](mailto:simon.dicker@gmail.com), Brian Mason at [bmason@nrao.edu](mailto:bmason@nrao.edu), or Emily Moravec at [emoravec@nrao.edu](mailto:emoravec@nrao.edu) for permission concerning proposal submission or any other inquiries concerning MUSTANG-2.

### 4.2.2.3 Technical Aspects

#### Bandpass

Description MUSTANG-2's bandpass.

[See Details](#)

#### Filtering

Description of MUSTANG-2 transfer functions with various filtering and scans sizes.

[See Details](#)

#### Mapping

Description of mapping parameters.

[See Details](#)

### MUSTANG-2 Bandpass Information

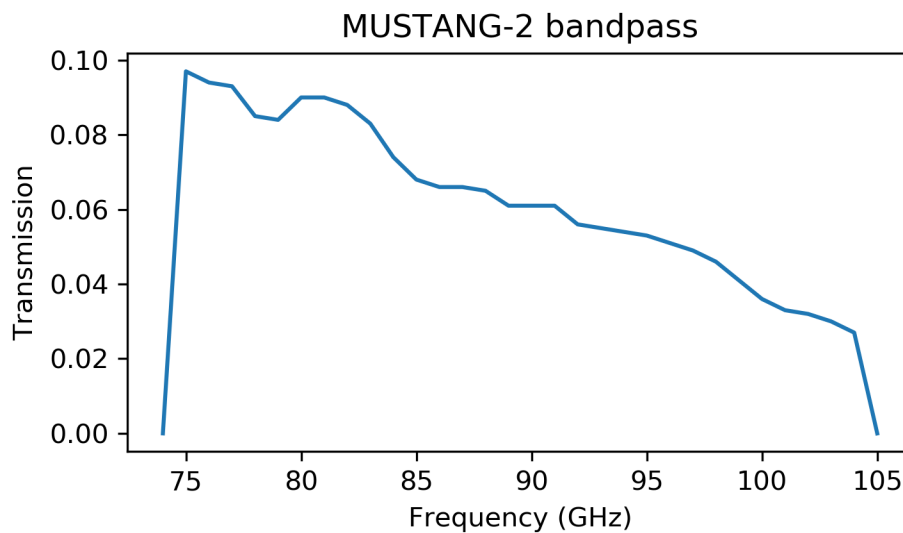




Table 1: Spectral Index Information

Spectral Index	Center Frequency
-4.0	85.453782
-3.5	85.822352
-3.0	86.199666
-2.5	86.585006
-2.0	86.977568
-1.5	87.376465
-1.0	87.780735
-0.5	88.189355
0.0	88.601248
0.5	89.015301
1.0	89.430375
1.5	89.845321
2.0	90.258998
2.5	90.670285
3.0	91.078096
3.5	91.481391
4.0	91.879194
4.5	92.270596
5.0	92.654768

You can download a text file with these spectral index and frequency values [here](#).

## MUSTANG-2 Filtering

Like any other ground-based millimeter continuum observations, our default data processing attempts to remove the Earth's atmosphere with either a common-mode subtraction or something quite similar to it (e.g. PCA).

There is an early memo on filtering which compared the results of observing simulated galaxy clusters using two different (Lissajous daisy) scan sizes and two different sets of filtering parameters. There is an executive summary [here](#).

The following figures show transfer functions for a broad range of (Lissajous) scan sizes (from 2.5 to 5.0) with a fairly gentle filtering (3 components subtracted from the timestreams via PCA and a windowed filter, keeping frequencies between 0.06 Hz and 41.0 Hz).

Below are the transfer functions from above with the same scan sizes but instead filtered with 5 components (via PCA) and a window between 0.08 Hz and 41.0 Hz. We note that the units of the wavenumber axes of all the plots on this page are 1/arcseconds.

Below we show the differences between these two reductions (PCA=3 with a Fourier window between 0.06 Hz and 41.0 Hz vs PCA=5 with a Fourier window between 0.08 Hz and 41.0 Hz) by splitting the scan sizes used above among two plots:

A filtering with a highpass of 0.06 Hz is, unfortunately, a bit more gentler than we find is necessary. Rather, a highpass at 0.07 or 0.08 Hz often results in acceptable noise in our maps. There are still other datasets which require still more aggressive filtering, either a highpass at 0.09 Hz, or even 0.1 Hz.

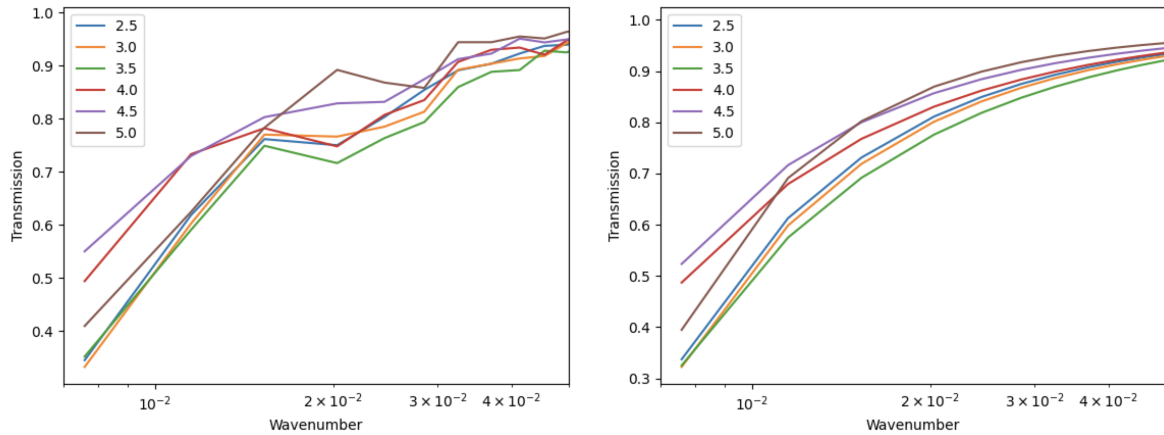


Fig. 1: Figure 1. Transfer functions for reduction parameters of PCA=3, and a Fourier window between 0.06 Hz and 41.0 Hz. **Left:** large-scale transmission by scan size (raw binned data). **Right:** fitted curves of the form  $\text{Transmission} = p[0] - (r/p[1])^p[2]$ , by scan size (in arcminutes).

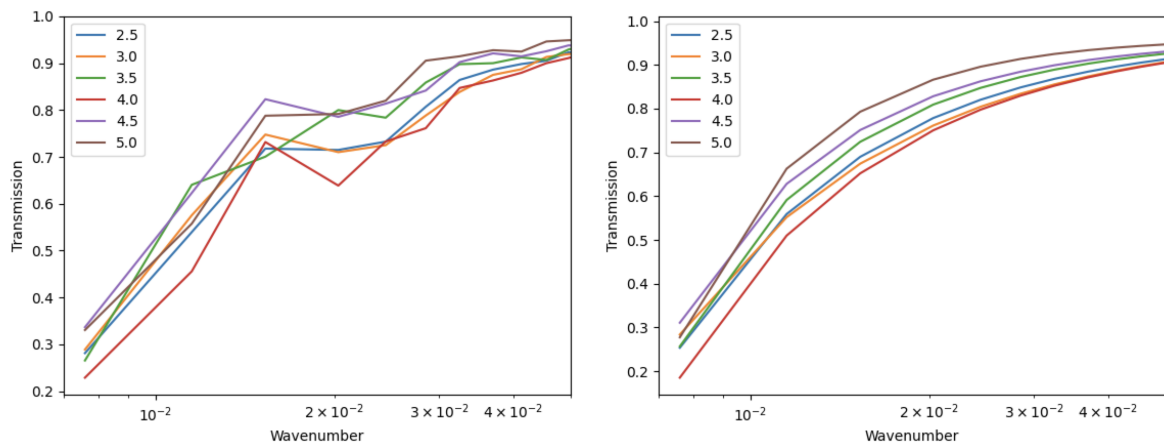


Fig. 2: Figure 2. Transfer functions for reduction parameters of PCA=5, and a Fourier window between 0.08 Hz and 41.0 Hz. **Left:** large-scale transmission by scan size (raw binned data). **Right:** fitted curves of the form  $\text{Transmission} = p[0] - (r/p[1])^p[2]$ , by scan size (in arcminutes).

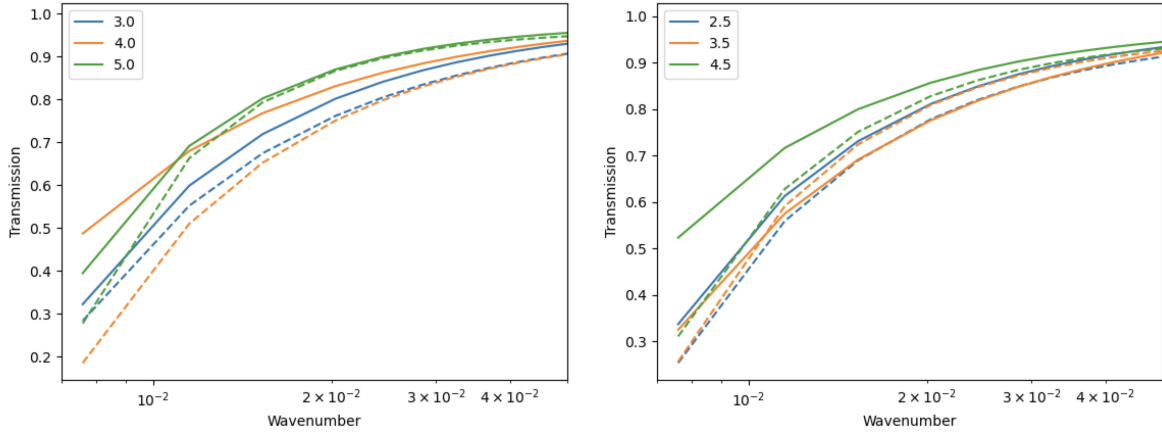


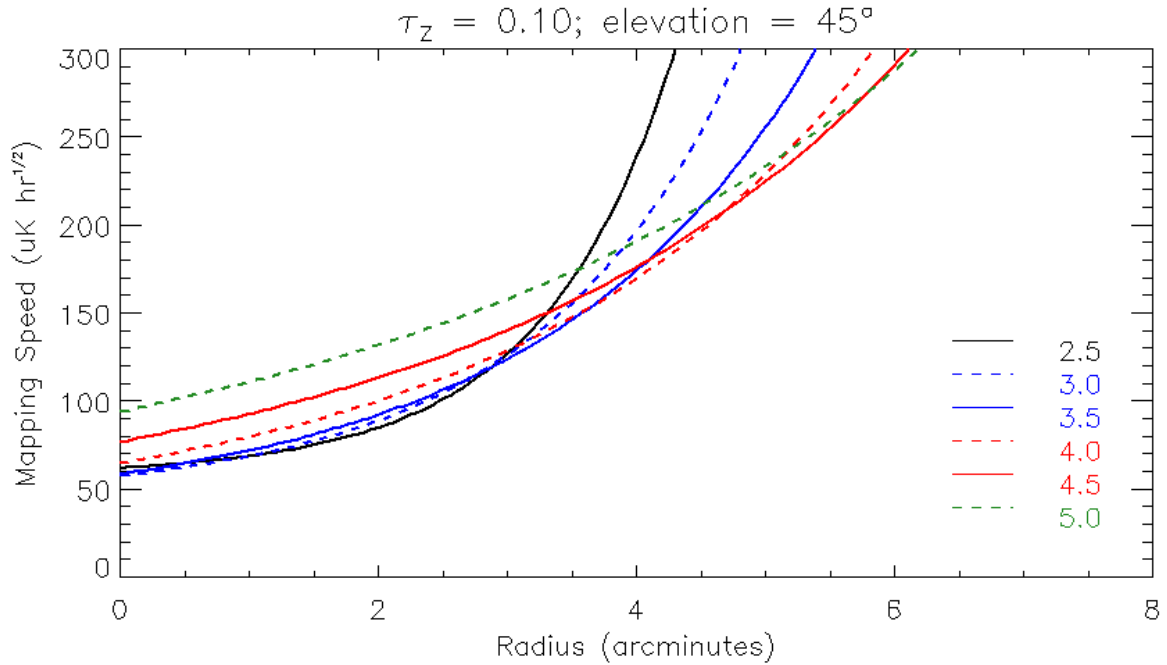
Fig. 3: Figure 3. Dashed lines indicate the harsher filtering (CA=5 with a Fourier window between 0.08 Hz and 41.0 Hz); solid lines are the gentler filtering (PCA=3 with a Fourier window between 0.06 Hz and 41.0 Hz).

### Transfer Functions for Download

A repository of transfer functions (ascii files) is available [here](#).

### MUSTANG-2 Mapping Information

MUSTANG-2's primary scanning method is that of a Lissajous Daisy of varying radii. The mapping speed profiles have been scaled to a zenith opacity of 0.1 and elevation of 45 degrees.



This sensitivity assumes an effective smoothing of 10.0'' FWHM. If heavier smoothing is acceptable the sensitivity is better by a factor of (FWHM/10.0 arcsec).

Historically, we have reported mapping speeds based on the RMS within a circle of radius 2 arcminutes. These values have now been extended to a range of scan sizes seen below.

Table 2: Mapping speeds and corresponding sensitivities

Scan size (radius, arcminutes)	Mapping speed in central 2' ( $\mu$ K hr <sup>1/2</sup> )	Sensitivity ( $\mu$ Jy/beam)	Sensitivity (Compton y)
2.5	73	56.2	2.0E-5
3.0	74	57	2.0E-5
3.5	77	59.3	2.1E-5
4.0	85	65.5	2.3E-5
4.5	98	75.5	2.7E-5
5.0	117	90.1	3.2E-5

#### Notes on the above table:

- The mapping speeds above are calculated based on the RMS in the central 2 arcminutes. For small scans, this region has roughly uniform noise.
- The conversion from  $\mu$  K to  $\mu$  Jy/beam is roughly 0.77; i.e. multiple the  $\mu$  K noise by 0.77 to get a noise estimate in  $\mu$  Jy/beam.
- The conversion from  $\mu$  K to Compton y is roughly 2.74E-7; i.e. multiple the  $\mu$  K noise by 2.74E-7 to get a noise estimate in Compton y.
- Extended signal on scales up to 5' should be imaged with reasonable fidelity, but faint signal more extended than this may be difficult to detect. Bright emission (20 mJy/beam or more) can be reconstructed over scales of many arcminutes. The angular resolution of MUSTANG on the GBT is typically 9" (FWHM) and the instantaneous field of view is a 4' diameter circle. Recovery of signal depends on data processing.

## Mapping Memo

Here is a MUSTANG-2 mapping speeds memo.

### 4.2.2.4 Useful References

#### Publications

List of MUSTANG-2 publications and press releases.

[See Publications](#)

#### Development

Description MUSTANG-2's historical development.

[See Timeline and Photos](#)

## MUSTANG-2 Publications

### Peer Reviewed Publications

- [Library](#) of MUSTANG-2 peer reviewed publications.
- [Library](#) of MUSTANG peer reviewed publications.

### Press Releases

- [Scientists Reveal Secrets to Burping Black Hole with the Green Bank Telescope](#) (Dec 2022) -
- [New image captures galaxy cluster emerging from cosmic web](#) (June 2021)
- [More than Meets the Eye: Complete Imaging of Cluster Collision](#) (Nov 2021)
- [MUSTANG Penetrates the Central Molecular Zone](#) (March 2019)
- [The Green Bank Telescope Snaps New Hot Photos of the Moon](#) (Aug 2019)

## MUSTANG-2 Historical Development

### Historical Updates

#### **Dec 2016: MUSTANG-2 on GBT to Commence Science Observing**

- MUSTANG-2 has been installed on the GBT with upgraded readout multiplexers and 211 detectors working. Observing on shared risk science proposals in collaboration with the instrument team have begun.

#### **May 2016: MUSTANG-2 First Light on GBT**

- The full MUSTANG-2 camera achieved first light on the GBT in May, 2016! It will return to the GBT in fall 2016 for shared-risk science through the 2017-A Green Bank Observatory proposal call.

#### **July 2015: MUSTANG-2 green-lighted for full production**

- The NSF ATI program has funded our proposal to fully populated the MUSTANG-2 focal plane with 215 dual polarization detectors. The completed MUSTANG-2 receiver is expected to arrive at the GBT for commissioning & early science observations early in 2016.

#### **May 2015: MUSTANG-1.5 completes successful engineering run on the GBT**

- MUSTANG-1.5 was successfully integrated with the GBT and first light observations of a variety of sources obtained. This run allowed the team to shake out final integration issues with the telescope and software systems, and constituted the first astronomical use of MKID-style microwave resonators to read out TES detectors.

#### **December 2, 2014: MUSTANG-1.5 Arrives at the GBT**

- MUSTANG-1.5 arrived in Green Bank and is preparing for first light on the GBT later this month!

#### **MUSTANG-2 is Under Construction!**

- MUSTANG-1.5 is currently being integrated and tested in the lab at U.Penn., with first light on the GBT scheduled for winter 2014.

#### **June 2014: MUSTANG-2 Detectors delivered to U.Penn!**

- In June 2014 72 packaged detectors were delivered to U.Penn for testing and characterization.

### MUSTANG retired at the end of 2013

- MUSTANG – the original 90 GHz, 64 pixel GBT bolometer camera– was retired at the end of calendar year 2013 in anticipation of the arrival of its successor, MUSTANG-1.5 MUSTANG proposals were accepted from June 2009 through February 2012.

### Photos

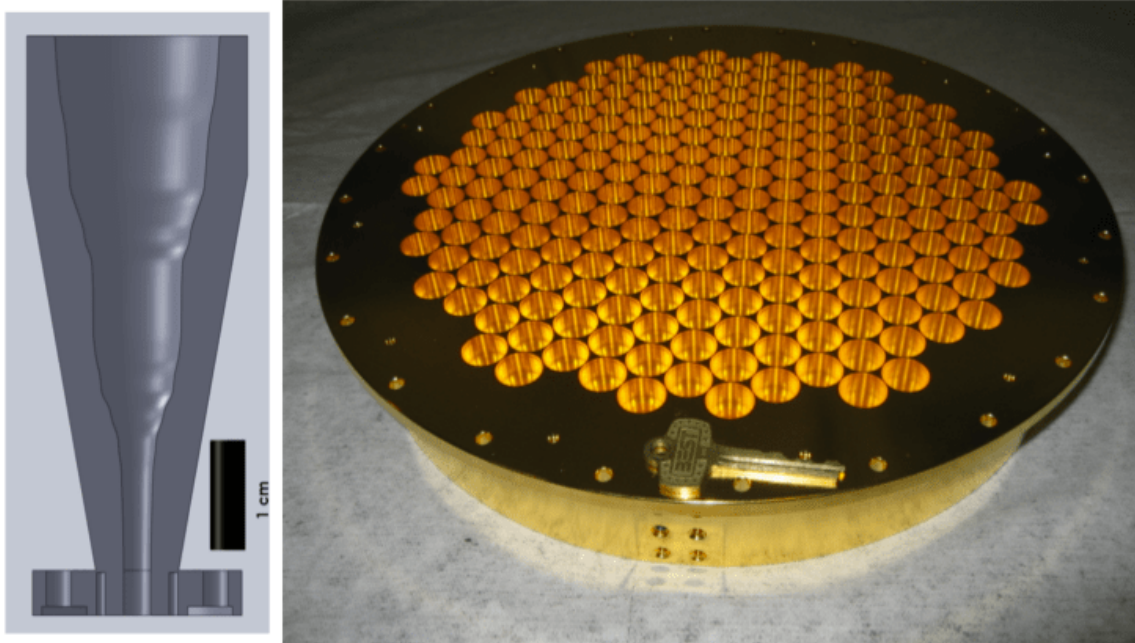


Fig. 4: Left: Cutaway view of tapered feed design. Right: the final MUSTANG-2 feed horn array as fabricated (223 feeds). The feed array weighs 8kg and has been successfully cooled to 300 mK in the MUSTANG-2 cryostat.

## 4.2.3 Scheduling Block Commands

**Annotation**(*key*, *value=None*, *comment=None*)

An utility function. It allows you to add any keyword and value to the GO FITS file. This could be useful if there is any information you would like to record about your observation for later data processing, or for record keeping.

---

**Note:** The information in a FITS KEYWORD created via the Annotation() function will be ignored by the standard GBT data reduction package GBTIDL.

---

### Parameters

- **key** (*str*) – A completely uppercase string of eight characters or less. Do not use any standard FITS keywords!

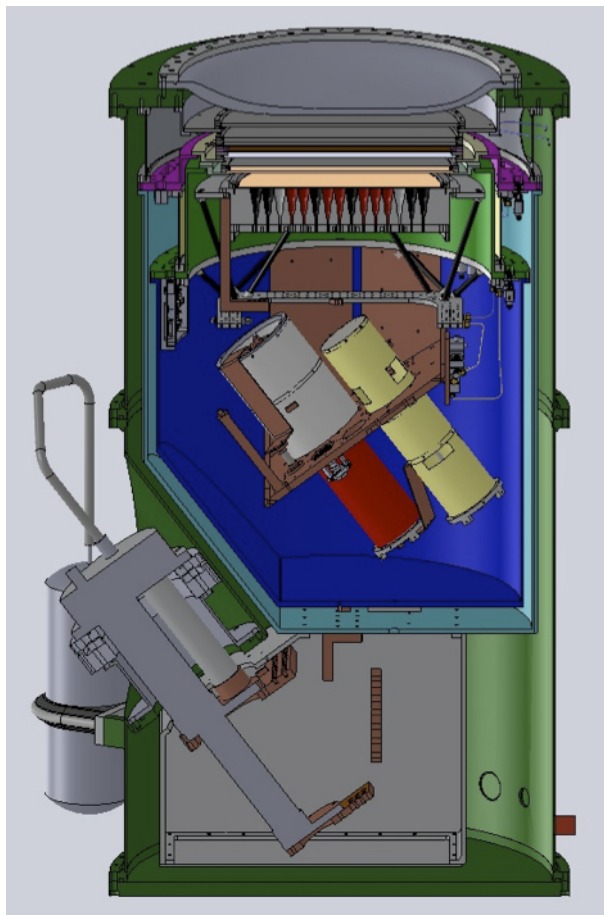


Fig. 5: Cutaway showing the design of the MUSTANG-1.5 receiver





Fig. 6: Receiver in the lab at U.Penn.





Fig. 7: Receiver in the lab at U.Penn.

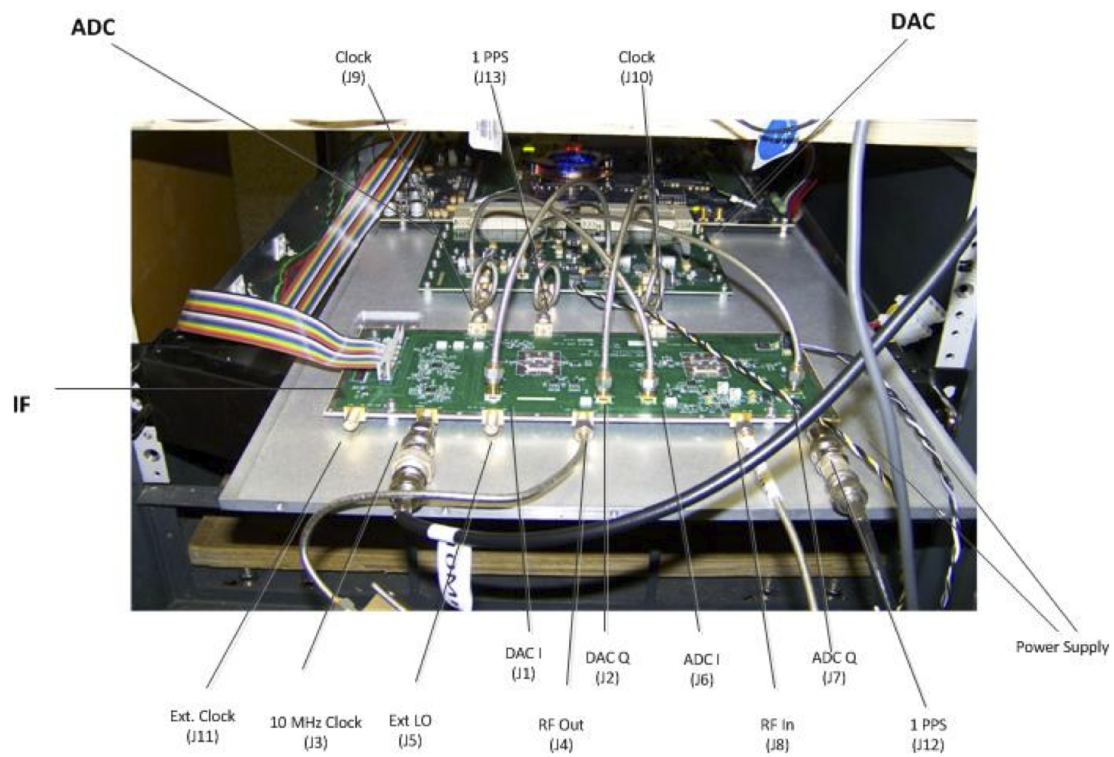


Fig. 8: One ROACH board with its custom daughter cards for the microwave resonator readout.

- **value** (*str*) – A value for the key.
- **comment** (*str*) – A comment to be added.

### Example

An example use of the Annotation() function is if you wish to specify what type of source you are observing. Your sources might include HII regions and Planetary Nebulae for example. You could specify each type with

```
Annotation('SRCTYPE',value='HII', comment='Type of source observed.')
Annotation('SRCTYPE',value='PNe', comment='Type of source observed.')
```

**Todo:** The Observer’s Guide does not mention comment as a parameter and does not explicitly state value in the command. Exact code usage needs to be checked.

**AutoFocus**(*source=None, location=None, frequency=None, flux=None, radius=10.0, balance=True, configure=True, beamName=None, refBeam=None, calSeq=True, gold=False*)

An utility scan. AutoFocus() is the same as AutoPeakFocus() except that it does not perform pointing scans. The intent of this scan type is to automatically focus the antenna for the current location on the sky and with the current receiver. Therefore it should not require any user input. However, by setting any of the optional arguments the user may partially or fully override the search and/or procedural steps as described below.

**Warning:** AutoFocus() should not be used with Prime Focus receivers. The prime focus receivers have pre-determined focus positions and there is not enough travel in the feed to move them significantly out of focus.

AutoFocus() will execute its own default continuum configuration unless “configure=False” is supplied as an optional argument, which is not recommended in general unless one knows the system well.

### Parameters

- **source** (*str*) – It specifies the name of a particular source in the pointing catalog or in a user-defined Catalog. Specifying a source bypasses the search process. Please note that NVSS source names are used in the pointing catalog. If the name is not located in the pointing catalog then all the user-specified catalogs previously defined in the Scheduling Block are searched. If the name is not in the pointing catalog or in the user defined catalog(s) then the procedure fails.
- **location** – A Catalog source name or Location object. It specifies the center of the search radius. The default is the antenna’s current beam location on the sky. Planets and other moving objects may not be used.
- **frequency** (*float*) – It specifies the observing frequency in MHz. The default is the rest frequency used by the standard continuum configurations, or the current configuration value if “configure=False”.
- **flux** (*float*) – It specifies the minimum acceptable calibration flux in Jy at the observing frequency. The default is 20 times the continuum point-source sensitivity.
- **radius** (*float*) – The routine selects the closest calibrator within the radius (in degrees) having the minimum acceptable flux. The default radius is 10 degrees. If no calibrator is found within the radius, the search is continued out to 180 degrees and if a qualified calibrator is found the user is given the option of using it [default], aborting the scan, or continuing the scheduling block without running this procedure.

- **balance** (*bool*) – Controls whether after slewing to the calibrator the routine balances the power along the IF path and again to set the power levels just before collecting data. Allowed values are True or False.
- **configure** (*bool*) – This argument causes the telescope to configure for continuum observing for the specified receiver.

---

**Note:** Because AutoFocus() is self-configuring when set to True, you must re-configure the GBT IF path for your science observations after the pointing and focus observations are done. If set to False, then no configuration is done, and the observer must ensure that the system is properly configured first before running the command.

---

- **beamName** (*str*) – It specifies which receiver beam will be the center of the cross-scan. beamName can be ‘C’, ‘1’, ‘2’, ‘3’, ‘4’, etc, up to ‘7’ for the KFPA receiver. This keyword should not be specified unless there is an issue with the default beams used for pointing (7.6).
- **refBeam** (*str*) – It specifies which beam will be the reference beam for subtracting the sky contribution for the pointing observations. The name strings are the same as for the beamName argument. This keyword should not be specified unless there is an issue with the default beams used for pointing (7.6). The KFPA and Argus have backup beam pairs that can be used for pointing and focus if there is an issue with one of the default beams. The valid beam pairs for the KFPA are beamName=‘4’, refBeam=‘6’ (default beam pair) or beamName=‘3’, refBeam=‘7’ (backup beam pair), while the valid beam pairs for Argus are beamName=‘10’, refBeam=‘11’ (default beam pair) or beamName=‘14’, refBeam=‘15’ (backup beam pair).

**calSeq: bool**

If True, then for ‘Rcvr68 92’ (W-Band) the observations will be preceeded by calibration calSeq observations, or for ‘RcvrArray75 115’ (argus) the calibration “vanecal” observations. If False, then the calibration vanecal or calSeq observations will be skipped. This keyword is only applicable for receivers operating above 66 GHz, and the associated calibration observations depend on the receiver (‘Rcvr68 92’, ‘RcvrArray75 115’, and ‘Rcvr PAR’) and the particular Auto utility procedure (see the individual receiver chapters for specifics). The default value is True.

**gold: bool**

If True then only “Gold standard sources” (i.e. sources suitable for pointing at high frequencies) will be used by AutoPeakFocus(). This parameter is ignored if the “source” parameter is specified.

AutoPeak will use the default scanning rates and lengths listed in Table 7.6.

---

**Todo:** Copy Table 7.6 from Observer’s Guide here.

---

---

**Sequence of events done by AutoFocus() in full automatic mode, i.e. with no arguments**

1. Determine the appropriate receiver based on the selection in the scan coordinator.
2. Determine the recommended beam, antenna/subreflector motions, and duration for peak and focus scans.
3. Get current antenna beam location from the control system.
4. Configure for continuum observations with the current receiver.
5. Run a balance (see § 7.4.1.9) to place the IF power levels appropriately.

6. Determine the source as specified by the user or as chosen by software using the minimum flux, observing frequency, location, and search radius. If no pointing source is found within the specified radius, then provide the observer the option to use a more distant source (default), and if none found either aborting (second default) or continuing the scheduling block.
7. Slew to source.
8. Run another balance to set the power levels at the location of the source.
9. Run a scan using the Focus command.

## Examples

The script below gives examples demonstrating the expected use of AutoPeakFocus.

```
#Configure for correct receiver at start of session...
execfile('/home/astro-util/projects/GBTog/configs/tp_config.py')
Configure(tp_config)

#Default (fully automatic)
AutoFocus()

#point and focus on 3C286
AutoFocus('3C286')

# find a pointing source near ra=16:30:00 dec=47:23:00
AutoFocus(location=Location('J2000', '16:30:00', '47:23:00'))

# Since AutoFocus() has executed its own configuration reconfigure for science_
↳ observations
Configure(tp_config)
```

**AutoOOF**(*source, location, frequency, flux, radius, balance, configure, beamName, refBeam, calSeq, gold, nseq='optional'*)

An utility scan. Out-Of-Focus (OOF) holography is a technique for measuring large-scale errors in the shape of the reflecting surface by mapping a strong point source both in and out of focus. The procedure derives surface corrections which can be sent to the active surface controller to correct surface errors. The procedure is recommended when observing at frequencies of 40 GHz and higher.

**AutoOOF() can only be used for observations above 26 GHz. Receiver choices are limited**

- 'Rcvr26\_40' (Ka-Band)
- 'Rcvr40\_52' (Q-Band)
- 'Rcvr68\_92' (W-Band)
- 'RcvrArray75\_115' (Argus)
- 'Rcvr\_PAR'(MUSTANG-2)

**Note:** AutoOOF is used in a similar manner to AutoPeakFocus. The command should normally be run without any arguments, since the default options are handled best by the OOF processing software. It is important to only OOF on bright sources (at least 3-4 Jy).

### Parameters

- **nseq** – Is an optional parameter for use with ‘Rcvr PAR’. It is used to specify the number of OTF maps made with AutoOOF and may take values of 3 or 5.
- **calSeq** (*bool*) – The default value is True which will run a set of calibration scans before the OOF map scans for ‘Rcvr68 92’ and ‘RcvrArray75 115’. This keyword is only applicable for receivers operating above 66 GHz and the associated calibration observations depend on the receiver (‘Rcvr68 92’, ‘RcvrArray75 115’, and ‘Rcvr PAR’; see the individual receiver chapters for specifics).

### Examples

```
# Specifying a source for AutoOOF
AutoOOF('2253+1608')
```

---

**Note:** The example OOF command above is applicable for all receivers, and users should refer to the individual receiver chapters to understand the specifics on how the OOF data are calibrated via the calSeq keyword. For Ka-band, the default backend for the AutoOOF procedure is the CCB. Besides the much higher sensitivity provided by the CCB (a fast beam-switching backend designed for continuum measurements), the larger beam at lower frequency makes the surface solutions less affected by winds. Since the Ka+CCB system provides the most accurate measurements of the surface parameters, it should be used whenever possible.

Users should refrain from using non-standard defaults for the AutoOOF measurements since the processing software is customized per receiver based on the default parameters (e.g., avoid running an AutoOOF at a non-default frequency or pre-configuring and running with the configuration=False keyword).

---

---

**Todo:** Add receiver specific instructions in the Hardware segment of the reference guides.

---

**AutoPeak**(*source=None, location=None, frequency=None, flux=None, radius=10.0, balance=True, configure=True, beamName=None, refBeam=None, elAzOrder=False, calSeq=True, gold=False*)

An utility scan. AutoPeak() is the same as AutoPeakFocus() except that it does not perform a focus scan. The intent of this scan type is to automatically peak the antenna for the current location on the sky and with the current receiver. Therefore it should not require any user input. However, by setting any of the optional arguments the user may partially or fully override the search and/or procedural steps as described below.

**Warning:** AutoPeak() should not be used with Prime Focus receivers. The prime focus receivers have pre-determined focus positions and there is not enough travel in the feed to move them significantly out of focus.

AutoPeak() will execute its own default continuum configuration unless “configure=False” is supplied as an optional argument, which is not recommended in general unless one knows the system well.

### Parameters

- **source** (*str*) – It specifies the name of a particular source in the pointing catalog or in a user-defined Catalog. Specifying a source bypasses the search process. Please note that NVSS source names are used in the pointing catalog. If the name is not located in the pointing catalog then all the user-specified catalogs previously defined in the Scheduling Block are searched. If the name is not in the pointing catalog or in the user defined catalog(s) then the procedure fails.

- **location** – A Catalog source name or Location object. It specifies the center of the search radius. The default is the antenna’s current beam location on the sky. Planets and other moving objects may not be used.
- **frequency** (*float*) – It specifies the observing frequency in MHz. The default is the rest frequency used by the standard continuum configurations, or the current configuration value if “configure=False”.
- **flux** (*float*) – It specifies the minimum acceptable calibration flux in Jy at the observing frequency. The default is 20 times the continuum point-source sensitivity.
- **radius** (*float*) – The routine selects the closest calibrator within the radius (in degrees) having the minimum acceptable flux. The default radius is 10 degrees. If no calibrator is found within the radius, the search is continued out to 180 degrees and if a qualified calibrator is found the user is given the option of using it [default], aborting the scan, or continuing the scheduling block without running this procedure.
- **balance** (*bool*) – Controls whether after slewing to the calibrator the routine balances the power along the IF path and again to set the power levels just before collecting data. Allowed values are True or False.
- **configure** (*bool*) – This argument causes the telescope to configure for continuum observing for the specified receiver.

---

**Note:** Because AutoPeak() is self-configuring when set to True, you must re-configure the GBT IF path for your science observations after the pointing and focus observations are done. If set to False, then no configuration is done, and the observer must ensure that the system is properly configured first before running the command.

---

- **beamName** (*str*) – It specifies which receiver beam will be the center of the cross-scan. beamName can be ‘C’, ‘1’, ‘2’, ‘3’, ‘4’, etc, up to ‘7’ for the KFPA receiver. This keyword should not be specified unless there is an issue with the default beams used for pointing (7.6).
- **refBeam** (*str*) – It specifies which beam will be the reference beam for subtracting the sky contribution for the pointing observations. The name strings are the same as for the beamName argument. This keyword should not be specified unless there is an issue with the default beams used for pointing (7.6). The KFPA and Argus have backup beam pairs that can be used for pointing and focus if there is an issue with one of the default beams. The valid beam pairs for the KFPA are beamName=‘4’, refBeam=‘6’ (default beam pair) or beamName=‘3’, refBeam=‘7’ (backup beam pair), while the valid beam pairs for Argus are beamName=‘10’, refBeam=‘11’ (default beam pair) or beamName=‘14’, refBeam=‘15’ (backup beam pair).
- **elAzOrder** (*bool*) – If True, the elevation peak scans will be done first before the azimuth peak scans. This can be helpful for high-frequency observations (> 40GHz) to provide more successful initial pointing solutions, since the elevation pointing offsets are typically larger than the azimuth offsets. The default is False, for which the azimuth pointing scans will be done before the elevation scans.
- **calSeq** (*bool*) – If True, then for ‘Rcvr68 92’ (W-Band) the observations will be preceded by calibration calSeq observations, or for ‘RcvrArray75 115’ (argus) the calibration “vanecal” observations. If False, then the calibration vanecal or calSeq observations will be skipped. This keyword is only applicable for receivers operating above 66 GHz, and the associated calibration observations depend on the receiver (‘Rcvr68 92’, ‘RcvrArray75 115’, and ‘Rcvr PAR’) and the particular Auto utility procedure (see the individual receiver chapters for specifics). The default value is True.

- **gold** (*bool*) – If True then only “Gold standard sources” (i.e. sources suitable for pointing at high frequencies) will be used by `AutoPeakFocus()`. This parameter is ignored if the “source” parameter is specified.

`AutoPeak` will use the default scanning rates and lengths listed in Table 7.6.

---

**Todo:** Copy Table 7.6 from Observer’s Guide here.

---



---

### Sequence of events done by `AutoPeak()` in full automatic mode, i.e. with no arguments

1. Determine the appropriate receiver based on the selection in the scan coordinator.
  2. Determine the recommended beam, antenna/subreflector motions, and duration for peak and focus scans.
  3. Get current antenna beam location from the control system.
  4. Configure for continuum observations with the current receiver.
  5. Run a balance (see § 7.4.1.9) to place the IF power levels appropriately.
  6. Determine the source as specified by the user or as chosen by software using the minimum flux, observing frequency, location, and search radius. If no pointing source is found within the specified radius, then provide the observer the option to use a more distant source (default), and if none found either aborting (second default) or continuing the scheduling block.
  7. Slew to source.
  8. Run another balance to set the power levels at the location of the source.
  9. Run a set of four pointing scans using the Peak command.
- 

### Examples

The script below gives examples demonstrating the expected use of `AutoPeakFocus`.

```
#Configure for correct receiver at start of session...
execfile('/home/astro-util/projects/GBTog/configs/tp_config.py')
Configure(tp_config)

#Default (fully automatic)
AutoPeak()

#point and focus on 3C286
AutoPeak('3C286')

# find a pointing source near ra=16:30:00 dec=47:23:00
AutoPeak(location=Location('J2000', '16:30:00', '47:23:00'))

# Since AutoPeak has executed its own configuration Reconfigure for science_
→ observations
Configure(tp_config)
```

**AutoPeakFocus**(*source=None, location=None, frequency=None, flux=None, radius=10.0, balance=True, configure=True, beamName=None, refBeam=None, elAzOrder=False, calSeq=True, gold=False*)

An utility scan. The intent of this scan type is to automatically peak and focus the antenna for the current location



on the sky and with the current receiver. Therefore it should not require any user input. However, by setting any of the optional arguments the user may partially or fully override the search and/or procedural steps as described below.

**Warning:** AutoPeakFocus() should not be used with Prime Focus receivers. The prime focus receivers have pre-determined focus positions and there is not enough travel in the feed to move them significantly out of focus.

AutoPeakFocus() will execute its own default continuum configuration unless “configure=False” is supplied as an optional argument, which is not recommended in general unless one knows the system well.

### Parameters

- **source** (*str*) – It specifies the name of a particular source in the pointing catalog or in a user-defined Catalog. Specifying a source bypasses the search process. Please note that NVSS source names are used in the pointing catalog. If the name is not located in the pointing catalog then all the user-specified catalogs previously defined in the Scheduling Block are searched. If the name is not in the pointing catalog or in the user defined catalog(s) then the procedure fails.
- **location** – A Catalog source name or Location object. It specifies the center of the search radius. The default is the antenna’s current beam location on the sky. Planets and other moving objects may not be used.
- **frequency** (*float*) – It specifies the observing frequency in MHz. The default is the rest frequency used by the standard continuum configurations, or the current configuration value if “configure=False”.
- **flux** (*float*) – It specifies the minimum acceptable calibration flux in Jy at the observing frequency. The default is 20 times the continuum point-source sensitivity.
- **radius** (*float*) – The routine selects the closest calibrator within the radius (in degrees) having the minimum acceptable flux. The default radius is 10 degrees. If no calibrator is found within the radius, the search is continued out to 180 degrees and if a qualified calibrator is found the user is given the option of using it [default], aborting the scan, or continuing the scheduling block without running this procedure.
- **balance** (*bool*) – Controls whether after slewing to the calibrator the routine balances the power along the IF path and again to set the power levels just before collecting data. Allowed values are True or False.
- **configure** (*bool*) – This argument causes the telescope to configure for continuum observing for the specified receiver.

---

**Note:** Because AutoPeakFocus() is self-configuring when set to True, you must re-configure the GBT IF path for your science observations after the pointing and focus observations are done. If set to False, then no configuration is done, and the observer must ensure that the system is properly configured first before running the command.

---

- **beamName** (*str*) – It specifies which receiver beam will be the center of the cross-scan. beamName can be ‘C’, ‘1’, ‘2’, ‘3’, ‘4’, etc, up to ‘7’ for the KFPA receiver. This keyword should not be specified unless there is an issue with the default beams used for pointing (7.6).
- **refBeam** (*str*) – It specifies which beam will be the reference beam for subtracting the sky contribution for the pointing observations. The name strings are the same as for the beamName argument. This keyword should not be specified unless there is an issue with

the default beams used for pointing (7.6). The KFPA and Argus have backup beam pairs that can be used for pointing and focus if there is an issue with one of the default beams. The valid beam pairs for the KFPA are beamName='4', refBeam='6' (default beam pair) or beamName='3', refBeam='7' (backup beam pair), while the valid beam pairs for Argus are beamName='10', refBeam='11' (default beam pair) or beamName='14', refBeam='15' (backup beam pair).

- **e1AzOrder** (*bool*) – If True, the elevation peak scans will be done first before the azimuth peak scans. This can be helpful for high-frequency observations (> 40GHz) to provide more successful initial pointing solutions, since the elevation pointing offsets are typically larger than the azimuth offsets. The default is False, for which the azimuth pointing scans will be done before the elevation scans.
- **calSeq** (*bool*) – If True, then for 'Rcvr68 92' (W-Band) the observations will be preceded by calibration calSeq observations, or for 'RcvrArray75 115' (argus) the calibration "vanecal" observations. If False, then the calibration vanecal or calSeq observations will be skipped. This keyword is only applicable for receivers operating above 66 GHz, and the associated calibration observations depend on the receiver ('Rcvr68 92', 'RcvrArray75 115', and 'Rcvr PAR') and the particular Auto utility procedure (see the individual receiver chapters for specifics). The default value is True.
- **gold** (*bool*) – If True then only "Gold standard sources" (i.e. sources suitable for pointing at high frequencies) will be used by AutoPeakFocus(). This parameter is ignored if the "source" parameter is specified.

AutoPeakFocus will use the default scanning rates and lengths listed in Table 7.6.

---

**Todo:** Copy Table 7.6 from Observer's Guide here.

---

---

#### Sequence of events done by AutoPeakFocus() in full automatic mode, i.e. with no arguments

1. Determine the appropriate receiver based on the selection in the scan coordinator.
  2. Determine the recommended beam, antenna/subreflector motions, and duration for peak and focus scans.
  3. Get current antenna beam location from the control system.
  4. Configure for continuum observations with the current receiver.
  5. Run a balance (see § 7.4.1.9) to place the IF power levels appropriately.
  6. Determine the source as specified by the user or as chosen by software using the minimum flux, observing frequency, location, and search radius. If no pointing source is found within the specified radius, then provide the observer the option to use a more distant source (default), and if none found either aborting (second default) or continuing the scheduling block.
  7. Slew to source.
  8. Run another balance to set the power levels at the location of the source.
  9. Run a set of four pointing scans using the Peak command.
  10. Run a scan using the Focus command.
-

## Examples

The script below gives examples demonstrating the expected use of AutoPeakFocus.

```
#Configure for correct receiver at start of session...
execfile('/home/astro-util/projects/GBTog/configs/tp_config.py')
Configure(tp_config)

#Default (fully automatic)
AutoPeakFocus()

#point and focus on 3C286
AutoPeakFocus('3C286')

# find a pointing source near ra=16:30:00 dec=47:23:00
AutoPeakFocus(location=Location('J2000', '16:30:00', '47:23:00'))

# Since AutoPeakFocus has executed its own configuration Reconfigure for science.
↪ observations
Configure(tp_config)
```

## Balance()

An utility scan. The Balance() command is used to balance the electronic signal throughout the GBT IF system so that each device is operating in its linear response regime. Balance() will work for any device with attenuators and for a particular backend. Individual devices can be balanced such as Prime Focus receivers, the IF system, the DCR, and VEGAS. The Gregorian receivers lack attenuators and do not need to be balanced. If the argument to Balance() is blank (recommended usage), then all devices for the current state of the IF system will be balanced using the last executed configuration to decide what hardware will be balanced.

### Advanced Syntax

Use only if you really know what you're doing.

```
Balance('DeviceName', {'DeviceKeyword': Value})
```

## Examples

```
# example showing the expected use of Balance()

# load configuration
execfile('/home/astro-util/projects/GBTog/configs/tp_config.py')
Configure(tp_config)

# Slew to target so that you may balance "on source"
Slew('3C286')

# Balance IF and devices for specified configuration
Balance()
```

## BalanceOnOff(location, offset=None, beamName='1')

An utility scan. When there is a large difference in power received by the GBT between two positions on the sky, it is advantageous to balance the IF system power levels to be at the mid-point of the two power levels. Typically this is needed when the “source position” is a strong continuum source. This scan type has been created to handle

this scenario; one should consider using it when the system temperature on and off source differ by a factor of two or more.

The `BalanceOnOff()` slews to the source position and then balances the IF system. It then determines the power levels that are observed in the IF Rack. Then the telescope is slewed to the off position and the power levels are determined again. The change in the power levels is then used to determine attenuator settings that put the balance near the mid-point of the observed power range. Note that the balance is determined only to within  $\pm 0.5$  dB owing to the integer settings of the IF Rack attenuators.

#### Parameters

- **location** – A Catalog source name or Location object. It specifies the source of which the telescope should slew. The default is the current location in “J2000” coordinate mode.
- **offset** (*Offset object*) – It moves the beam to an optional offset position that is specified relative to the location specified in the location parameter value.
- **beamName** (*str*) – It specifies the receiver beam to use for the scan. beamName can be ‘C’, ‘1’, ‘2’, ‘3’, ‘4’, or any valid combination for the receiver you are using such as ‘MR12’.

#### Examples

```
# example showing the expected use of Balance()

# load configuration
execfile('/home/astro-util/projects/GBTog/configs/tp_config.py')
Configure(tp_config)

# Balance IF and devices for specified configuration
BalanceOnOff('3C48', Offset('J2000', 1.0, 0.0))
```

**Break**(*message*='Observation paused.', *timeout*=300.0)

An utility function. It inserts a breakpoint into your SB and gives you the choice of continuing or terminating the SB. When a breakpoint is encountered during execution, your SB is paused and a pop-up window is created. The SB remains paused for a set amount of time or until you acknowledge the pop-up window and tell Astrid to continue running your script.

The `Break()` function can take two optional arguments, a message string and a timeout length.

#### Parameters

- **message** (*str*) – Displayed in the pop-up dialog with a default of “Observation paused”.
- **timeout** (*float*) – The number of seconds to get user-input before continuing the SB. If you wish for the timeout to last forever then use `None`. The default is 300 seconds, or 5 minutes.

Why have a timeout? If an observer walks away from the control room during his or her observing session (e.g. to go to lunch or the bathroom) and a breakpoint is reached, it would be counterproductive to pause the observation indefinitely. This will help save valuable telescope time.

### Example

```
Break('This will time out in 5 minutes, the default.')
Break('This will time out after 10 minutes.', 600)
Break('This will never time out.', None)
```

### ChangeAttenuation(*devicename*, *attnchange*)

An utility function. It allows the observer to change all the attenuators in the IF Rack or the Converter Rack by the same amount.

#### Parameters

- **devicename** (*str*) – Can be either 'IFRack' or 'ConverterRack'. This specifies the device in which the attenuators will be changed.
- **attnchange** (*float.*) – This specifies how much the attenuators should be changed. This value can be either positive or negative.

---

**Note:** If any new attenuator setting is less than zero or exceeds the maximum value, 31 for the IF Rack and 31.875 for the Converter Rack, then the attenuator setting is made to be the appropriate limiting value.

---

### Examples

The following examples adds 1 to the attenuation value in the IF rack and subtracts 0.5 from the attenuation value in the converter rack.

```
ChangeAttenuation('IFRack', 1.0)
ChangeAttenuation('ConverterRack', -0.5)
```

### Comment(*message*)

An utility function. It allows you to add a comment into the Astrid observing process which will be echoed to the observation log during the observation. What's the difference between this, and just writing comments with the pound (#) sign in your SB? When you use the pound sign to write your comments, they will not appear in the observation log when your SB is run. Using the Comment() function directs your comment to the output in the observation log.

#### Parameters

- **message** (*str*) – Text to display during the observation.

### Example

```
# now slew to the source
Comment('Now slewing to 3C 286')
Slew('3C286')
```

**Daisy**(*location*, *map\_radius*, *radial\_osc\_period*, *radial\_phase*, *rotation\_phase*, *scanDuration*, *beamName*='1', *cos\_v*=True, *coordMode*='AzEl', *calc\_dt*=0.1, *scantype*='MAP', *nseq*=1, *annons*={}, *procseq*=1)

A mapping scan. The Daisy scan type performs an OTF scan around a central point in the form of daisy petals. It is a useful observing mode for focal plane arrays, allowing more integration time in the central field of view. The Daisy scan will produce an approximately closed circular pattern on the sky after 22 radial oscillation periods.

---

**Todo:** Add figure 7.6 from the Observer’s Guide above. Same issue with the x-axis here as in the figures for RALongMap, DecLatMap etc.

---

For beam-sizes of 20 arcsec FWHM or so, the circular area mapped will be fully sampled if the map radius is less than 6 arcmin. It is not an especially useful observing mode for general-purpose single-beam mapping, since the largest “hole” in the map is  $\sim 0.3\times$  the map radius.

Trajectories are generated according to

$$\Delta\hat{x}(t) = \frac{r_0 \sin(2\pi t/\tau + \phi_1) \cos(2t/\tau + \phi_2)}{\cos(\hat{y}_0)}$$

and

$$\Delta\hat{y}(t) = r_0 \sin(2\pi t/\tau + \phi_1) \sin(2t/\tau + \phi_2),$$

where  $\hat{x}$  and  $\hat{y}$  are the major and minor coordinates of a sperical coordinate system,  $t$  is the time,  $r_0$  is the map radius,  $\tau$  is the radial oscillation period,  $\phi_1$  and  $\phi_2$  are the radial and rotational phases, and  $\hat{y}_0$  is the minor coordinate of the map center.

### Parameters

- **location** – A Catalog source name or Location object. It specifies the center of the map.
- **map\_radius** (*float*) –  $r_0$  in equation above. It specifies the radius of the map’s “daisy petals” in arcmin.
- **radial\_osc\_period** (*float*) –  $\tau$  in the equations above. It specifies the period of the radial oscillation in seconds.

**Caution:** Not to be less than  $15 \text{ sec} \times \sqrt{r_0/1.5 \text{ arcmin}}$  for radii  $> 1.5$  arcmin and in no case under 15 seconds.

- **radial\_phase** (*float*) –  $\phi_1$  in equations above. It specifies the radial phase in radians.
- **rotation\_phase** (*float*) –  $\phi_2$  in equations above. It specifies the rotational phase in radians.
- **scanDuration** (*float*) – It specifies the length of the scan in seconds.
- **beamName** (*str.*) – It specifies the receiver beam to use for both scans. beamName can be ‘C’, ‘1’, ‘2’, ‘3’, ‘4’ or any valid combination for the receiver you are using such as ‘MR12’.
- **cos\_v** (*bool*) – It specifies whether secant minor corrections (the  $\cos(\hat{y}_0)$  term in the first equation above should be used for the major axis of the coordinate system.
- **coordMode** (*str*) – It specifies the coordinate mode for the radius that generates the map.
- **calc\_dt** (*float*) – It specifies time sampling used by the control system to calculate a path. Values should be between 0.1 and 0.5. Calculating many points for a long daisy scan can significantly increase overhead at scan startup.

---

**Note:** It takes approximately 22 radial oscillation periods to complete a closed Daisy pattern. However, radial oscillation period is typically set to be in the range of 15–60 seconds depending on the radius being used. As an example, 22 oscillations of 20 seconds would take 440 seconds. If a long trajectory such as this is sent to the antenna manager, intrinsic inefficiencies in the array handling mechanism can significantly increase overhead at the start of a scan. Therefore one should try to keep individual scans to 5 minutes or less.

---

## Example

The script below will do 22 radial periods over 5 scans lasting 110 seconds each. The rotation phase and radial phase arguments are used so that each scan starts where the previous scan finished. This will produce the closed Daisy pattern. The entire SB should take approximately 10 minutes to complete.

```
nosc = 22.0                                # 22 radial oscillations for
↪closed Daisy pattern
map_radius = 2.8                            # in arcmin
radial_osc_period = 25.0                    # in seconds
n_scans = 5                                # split 22 oscillations over 5
↪scans
scanDuration = nosc * radial_osc_period / n_scans
phi2 = 2.0 * nosc / n_scans
phi1 = 3.14159265 * phi2

# NOTE:
#       - increment rotation_phase by phi2 each scan
#       - increment radial_phase by phi1 each scan

for i in range(n_scans):
    Daisy('3C123', map_radius, radial_osc_period, i * phi1, i * phi2, scanDuration,
          beamName='1', coordMode='J2000', cos_v=True, calc_dt=0.2)
```

**Todo:** Add figure 7.7 from Observer's Guide here. Same issue with the x-axis here as in the figures for RA-LongMap, DecLatMap etc.

**DecLatMap**(location, hLength, vLength, hDelta, scanDuration, beamName='1', unidirectional=False, start=1, stop=None)

A mapping scan. A Declination/Latitude map, or DecLatMap, performs an on-the-fly (OTF) raster scan centered on a specific location on the sky. Scans are performed in declination, latitude, or elevation coordinates depending on the desired coordinate system. This procedure does not allow the user to periodically move to a reference location on the sky, please see DecLatMapWithReference for such a map. The starting point of the map is at  $(-hLength/2, -vLength/2)$ .

### Parameters

- **location** – A Catalog source name or Location object. It specifies the center of the map.
- **hLength** – An Offset object. It specifies the horizontal width of the map, i.e., the extent in the longitude-like direction. hLength values may be negative.
- **vLength** – An Offset object. It specifies the vertical height of the map, i.e., the extent in the latitude-like direction. vLength values may be negative.
- **hDelta** – An Offset object. Similar to vDelta in RALongMap. It specifies the horizontal distance between map columns. hDelta values must be positive.
- **scanDuration** (float) – It specifies the length of each scan in seconds.
- **beamName** (str) – It specifies the receiver beam to use for the scan. beamName can be 'C', '1', '2', '3', '4' or any valid combination for the receiver you are using such as 'MR12'.
- **unidirectional** (bool) – It specifies whether the map is unidirectional (True) or boustrophedonic (False; from the Greek meaning 'as the ox plows, i.e. back and forth').

- **start** (*int*) – It specifies the starting row for the map. This is useful for doing parts of a map at different times. For example, if map has 42 rows, one can do rows 1-12 by setting “start=1, stop=12”, and later finish the map using “start=13, stop=42”.
- **stop** (*int*) – It specifies the stopping row for the map. The default is None, meaning “go to the end”.

---

**Note:** Observers should limit scanDuration, so that no more than 2 scans (or accelerations) are performed per minute. Overhead is ~20 seconds per scan.’

---

### Example

The following script produces a map with 41 rows each 120’ long, using a row spacing of 3’ and scan rate of 20’/min with beam 1 (default).

```
DecLatMap('NGC4258',                                # center of the map
          Offset('J2000', 2.0, 0.0, cosv=True),      # 120' width
          Offset('J2000', 0.0, 2.0, cosv=True),      # 120' height
          Offset('J2000', 0.05, 0.0, cosv=True),     # 3' column spacing
          360.0)                                     # 6 minutes per column
```

---

**Todo:** Add a plot showing the actual trajectory of the antenna on the sky. Figure 7.4 in the Observer’s Guide unfortunately shows the inverted RA axis, providing the impression the scan is obtained in negative RA direction.

---

**DecLatMapWithReference**(*location, hLength, vLength, hDelta, referenceOffset, referenceInterval, scanDuration, beamName='1', unidirectional=False, start=1, stop=None*)

A mapping scan. A Declination/Latitude map, or DecLatMap, performs an On-the-fly (OTF) raster scan centered on a specific location on the sky. Scanning is done in the declination, latitude, or elevation coordinate depending on the desired coordinate mode. This procedure allows the user to periodically move to a reference location on the sky. Please see DecLatMap if no reference point is required. The starting point of the map is at  $(-hLength/2, -vLength/2)$ .

#### Parameters

- **location** – A Catalog source name or Location object. It specifies the center of the map.
- **hLength** – An Offset object. It specifies the horizontal width of the map, i.e., the extent in the longitude-like direction. hLength values may be negative.
- **vLength** – An Offset object. It specifies the vertical height of the map, i.e., the extent in the latitude-like direction. vLength values may be negative.
- **hDelta** – An Offset object. Similar to vDelta in RALongMap. It specifies the horizontal distance between map columns. hDelta values must be positive.
- **referenceOffset** – An Offset object. It specifies the position of the reference source on the sky relative to the Location specified by the first input parameter.
- **referenceInterval** (*int*) – It specifies when to do a reference scan in terms of map columns. For example, setting referenceInterval=4 will periodically perform one scan on the reference source followed by 4 mapping scans.
- **scanDuration** (*float*) – It specifies the length of each scan in seconds.



- **beamName** (*str*) – It specifies the receiver beam to use for the scan. beamName can be ‘C’, ‘1’, ‘2’, ‘3’, ‘4’ or any valid combination for the receiver you are using such as ‘MR12’.
- **unidirectional** (*bool*) – It specifies whether the map is unidirectional (True) or boustrophedonic (False; from the Greek meaning ‘as the ox plows, i.e. back and forth’).
- **start** (*int*) – It specifies the starting row for the map. This is useful for doing parts of a map at different times. For example, if map has 42 rows, one can do rows 1-12 by setting “start=1, stop=12”, and later finish the map using “start=13, stop=42”.
- **stop** (*int*) – It specifies the stopping row for the map. The default is None, meaning “go to the end”.

### Example

The following script produces a map 120 arcmin long and 60 arcmin wide using a column spacing of 3 arcmin and scan rate of 4 arcmin/min. A reference position will be observed once before every 3 columns. The sequence of scans will be: reference → columns 1-3 → reference → columns 4-6 ...

```
DecLatMapWithReference('CygA',                                # center of map
                      Offset('J2000', 1.0, 0.0, cosv=True),  # 60 arcmin width
                      Offset('J2000', 0.0, 2.0, cosv=True),  # 120 arcmin length
                      Offset('J2000', 0.05, 0., cosv=True),   # 3 arcmin column
    ↪ spacing
                      Offset('J2000', 2.0, 0.0, cosv=True),   # 2 degree ref.
    ↪ offset in RA
                      3,                                       # reference before
    ↪ every 3 columns
                      30.0)                                    # 30 second scan
    ↪ duration
```

**Focus**(*location*, *start=None*, *focusLength=None*, *scanDuration=None*, *beamName=None*, *refBeam=None*)

An utility scan. Focus scan type moves the subreflector or prime focus receiver (depending on the receiver in use) through the axis aligned with the beam. Its primary use is to determine focus positions for use in subsequent scans.

#### Parameters

- **location** – Catalog source name or Location object. It specifies the source upon which to do the scan.
- **start** (*float*) – Specifies the starting position of the subreflector (in mm) for the focus scan.
- **focusLength** (*float*) – Specifies the ending position of the subreflector relative to the starting location (in mm).
- **scanDuration** (*float*) – Specifies the length of the scan in seconds. The default value is the recommended value for the receiver.
- **beamName** (*str*) – Specifies the receiver beam to use for measuring the focus. Make sure that you configure with the same beam with which you focus.
- **refBeam** (*str*) – Specifies the reference receiver beam to use for the receivers with more than one beam.

## Examples

```
# Focus using default settings and calibrator 0137+3309
Focus('0137+3309')

# Focus from -200 to 200mm at 400mm/min with beam 1
Focus('0137+3309', -200.0, 400.0, 60.0, '1')
```

### GetLST()

An utility function.

#### Parameters

**value** (*Return*) – The current Local Sidereal Time in decimal hours.

**Warning:** If Astrid is in “offline” mode, then GetLST() will return a value of None. Attempting to validate the script below without checking the return value is not equal to None while “offline” will result in an infinite loop.

## Example

The following example will repeatedly perform Track scans on the source “1153+1107” until the LST is past 13.5 hours when the source “1712+035” will be observed once.

```
while GetLST() < 13.5 and GetLST() != None:
    Track('1153+1107', None, 600.)

Track('1712+036', None, 600.)
```

### GetUTC()

An utility function.

#### Parameters

**value** (*Return*) – The current UTC time in decimal hours since midnight.

**Warning:** If Astrid is in “offline” mode, then GetUTC() will return a value of None. Attempting to validate the script below without checking the return value is not equal to None while “offline” will result in an infinite loop.

## Example

```
while GetUTC() < 12.0 and GetUTC() != None:
    Track('0353+2234', None, 600.)
```

### Horizon(*elevation=5.25*)

A scheduling block object. Observing Scripts allow an observer to specify a definition of the horizon. The user defined horizon can be used to begin an observation when an object “rises” and/or end the observation when it “sets” relative to the specified elevation of the “horizon”. The Horizon object may be used to obtain the initial time that a given source is above the specified horizon (including an approximate atmospheric refraction correction).

**Parameters**

- **location** – A Catalog source name or Location object using a spherical coordinate mode. Horizon() will not work with planets and ephemeris tables.
- **elevation** (*float.*) – The Horizon elevation in degrees. The default is 5.25 (the nominal GBT horizon limit).

**Return Value:**

A UTC time object containing the UTC time and date.

**Horizon(elevation).GetRise(location)**

Will return the most recent rise time if the source is currently above the horizon, or the next rise time if the source has not yet risen. GetRise(source) will return None if the source never rises and the current time if the source never sets.

**Horizon(elevation).GetSet(location)**

Will return the next set time of the source. GetSet(source) will return None if the source never sets and the current time if the source never rises.

**Examples**

Any Horizon object may be substituted as a start or stop time in scan types, such as Track(). The script below will display the time when VirgoA rises above 20° elevation. Depending on the position of the source at the time of execution, the SB would then either begin a Track() scan immediately or wait for VirgoA to rise above 5.25° elevation before beginning the scan. In both cases, the SB would terminate the next time VirgoA sets below 5.25° elevation.

```
print Horizon(20.0).GetRise('VirgoA')

h = Horizon()                                     #default horizon of 5.25 degrees elevation
Track('VirgoA', None, startTime=h, stopTime=h)
```

**Location**(*coordinateMode*, *value1*, *value2*)

A scheduling block object. It is used to represent a particular location on the sky.

**Parameters**

- **coordinateMode** (*str*) –

The following modes are allowed:

- 'J2000'
- 'B1950'
- 'RaDecOfDate'
- 'HaDec'
- 'ApparentRaDec'
- 'Galactic'
- 'AzEl'

- 'Encoder'
- **value1** (*float or str*) – A location must be specified by this value, the meaning depends on both the chosen coordinate mode and value type of the unit:
  - **float value:**  
Will always denote units in degrees of arc, regardless of the coordinate mode. This should not be confused with decimal use in Catalogs which denote decimal hours for RA and HA, and degrees of arc for all other angles.
  - **sexagesimal string:**  
Represents units of time for J2000, B1950, ApparentRaDec, and RaDecOfDate (i.e. 'hh:mm:ss.s') and degrees of arc for HaDec, Galactic, AzEl and Encoder (i.e. 'dd:mm:ss.s')
- **value2** (*float or str*) – A location must be specified by this value, the meaning depends on both the chosen coordinate mode and value type of the unit:
  - **float value:**  
Will always denote units in degrees of arc, regardless of the coordinate mode. This should not be confused with decimal use in Catalogs which denote decimal hours for RA and HA, and degrees of arc for all other angles.
  - **sexagesimal string:**  
Represents degrees of arc (i.e. 'dd:mm:ss.s')

## Examples

```
# RA is in units of *time*, Dec is in degrees
location = Location('J2000', '16:30:00', '47:15:00')

# Same location - RA is in degrees, Dec is in degrees
location = Location('J2000', 247.5, 47.25)

# Az is in degrees, El is in degrees
location = Location('AzEl', '45:00:00', '72:30:00')
```

**Nod**(*location, beamName1, beamName2, scanDuration*)

An observing scan. The Nod procedure does two scans on the same sky location with different beams.

**Caution:** Nod should only be used with multi-beam receivers.

### Parameters

**location** – A Catalog source name or Location object. It specifies the source upon which to do the Nod.

### beamName1: str

It specifies the receiver beam to use for the first scan. beamName1 can be '1' or '2' or any valid beam for the receiver you are using.

### beamName2: str

It specifies the receiver beam to use for the second scan. beamName2 can be 'C', '1', '2' or any valid beam for the receiver you are using.

**scanDuration: float**

It specifies the length of each scan in seconds.

**Examples**

```
# Nod between beams 3 and 7 with a 60s scan duration
Nod('1011-2610', '3', '7', 60.)
```

**Now()**

An utility function.

**Parameters**

**value** (*Return*) – A UTC time object containing the UTC time and date.

**Warning:** If Astrid is in “offline” mode, then Now() will return a value of None. Attempting to validate the script below without checking the return value is not equal to None while “offline” will result in an infinite loop.

**Example**

The following example will repeatedly perform Track scans on the source “1153+1107” until 09:54:12 UTC on 12 June 2016.

```
while Now() < '2016-06-12 09:54:12' and Now() != None:
    Track('1153+1107', None, 600.)
```

**OffOn**(*location*, *referenceOffset*, *scanDuration=None*, *beamName='1'*)

An observing scan. The OffOn scan type is the same as the OnOff scan except that the first scan is offset from the source location.

**Parameters**

- **location** – A Catalog source name or Location object. It specifies the source upon which to do the “On” scan.
- **referenceOffset** (*Offset object*) – Is specifies the location of the “Off” scan relative to the location specified by the first parameter.
- **scanDuration** (*float*) – It specifies the length of each scan in seconds.
- **beamName** (*str*) – It specifies the receiver beam to use for the scan.

**Examples**

```
# OffOn scan with reference offsets of 1 degree in RA and 1 deg in Dec with a 60 s
↳ scan duration using beam 1.
OffOn('0137+3309', Offset('J2000', 1.0, 1.0, cosv=False), 60, '1')
```

**Offset**(*coordinateMode*, *value1*, *value2*, *cosv=True*)

A scheduling block object. An Offset is a displacement from the position of a source or from the center position of a map. Offset objects may be added to other offset objects with the same coordinate mode and cosv correction. Offset objects may be added to Location objects with the same coordinate mode. Note that such addition is not commutative and must be of the form (Location+Offset). Offset+Location will produce a validation error.

### Parameters

- **coordinateMode** (*str*) –

The following modes are allowed:

- 'J2000'
- 'B1950'
- 'RaDecOfDate'
- 'HaDec'
- 'ApparentRaDec'
- 'Galactic'
- 'AzEl'
- 'Encoder'

- **value1** (*float or str*) – A location must be specified by this value, the meaning depends on both the chosen coordinate mode and value type of the unit:

- **float value:**

Will always denote units in degrees of arc, regardless of the coordinate mode. This should not be confused with decimal use in Catalogs which denote decimal hours for RA and HA, and degrees of arc for all other angles.

- **sexagesimal string:**

Represents units of time for J2000, B1950, ApparentRaDec, and RaDecOfDate (i.e. 'hh:mm:ss.s') and degrees of arc for HaDec, Galactic, AzEl and Encoder (i.e. 'dd:mm:ss.s')

- **value2** (*float or str*) – A location must be specified by this value, the meaning depends on both the chosen coordinate mode and value type of the unit:

- **float value:**

Will always denote units in degrees of arc, regardless of the coordinate mode. This should not be confused with decimal use in Catalogs which denote decimal hours for RA and HA, and degrees of arc for all other angles.

- **sexagesimal string:**

Represents degrees of arc (i.e. 'dd:mm:ss.s')

- **cosv** (*bool*) – It specifies whether secant minor corrections in equation the equation below should be used for the major axis of the coordinate system (i.e.  $h/\cos(v)$  is the offset value in the direction of  $h$ ). Since coordinate distances and angular separations are not equivalent for spherical coordinate systems, the following approximations may be used for small separations:

$$\Delta v = v_1 v_2$$

and

$$\Delta h = (h_1 h_2) \times \cos(v),$$

where  $h$  is the value of the major coordinate axis and  $v$  is the value of the minor coordinate axis.

For example, setting `cosv=True` with J2000 coordinate offsets will apply a `cos(Dec)` term from the second equation above to make maps appear rectangular if plotted with  $\Delta RA$  vs.  $\Delta Dec$  relative to a central location.

## Examples

The script below gives examples of adding Offset objects to Location and other Offset objects. The resulting coordinates are printed to screen.

```
start_location = Location('J2000','12:00:00','45:00:00')

offset1 = Offset('J2000','00:04:00','01:00:00',cosv=False)
offset2 = Offset('J2000', 2.0 , 2.0 ,cosv=False)
offset3 = offset1 + offset2

loc1 = start_location + offset1 #loc1 (RA,Dec) = (12:04:00, 45:00:00)
loc2 = start_location - offset2 #loc2 (RA,Dec) = (11:52:00, 43:00:00)
loc3 = start_location + offset3 #loc3 (RA,Dec) = (12:12:00, 48:00:00)

print 'RA,Dec of loc3 = (%s,%s)'%(loc3.GetH(),loc3.GetV())
```

**OnOff**(location, referenceOffset, scanDuration=None, beamName='1')

An observing scan. The OnOff scan type performs two scans. The first scan is on source, and the second scan is at an offset from the source location used in the first scan.

### Parameters

- **location** – A Catalog source name or Location object. It specifies the source upon which to do the “On” scan.
- **referenceOffset** (*Offset object*) – It specifies the location of the “Off” scan relative to the location specified by the first parameter.
- **scanDuration** (*float*) – It specifies the length of each scan in seconds.
- **beamName** (*str*) – It specifies the receiver beam to use for the scan.

## Examples

```
# OnOff scan with reference offsets of 1 degree in RA and 1 deg in Dec with a 60 s
↳ scan duration using beam 1.
OnOff('0137+3309', Offset('J2000', 1.0, 1.0, cosv=False), 60, '1')
```

**OnOffSameHA**(location, scanDuration, beamName='1')

An observing scan. The OnOffSameHA scan type performs two scans. The first scan is on the source and the second scan follows the same HA track used in the first scan.

### Parameters

- **location** – A Catalog source name or Location object. It specifies the source upon which to do the “On” scan.
- **scanDuration** (*float*) – It specifies the length of each scan in seconds.
- **beamName** (*str*) – It specifies the receiver beam to use for both scan.

## Examples

```
# OnOffSameHA scan with 60s scan duration using beam 1.
OnOffSameHA('0137+3309', 60, '1')
```

**Peak**(*location*, *hLength*=None, *vLength*=None, *scanDuration*=None, *beamName*=None, *refBeam*=None, *elAzOrder*=False)

An utility scan. Peak scan type sweeps through the specified sky location in the four cardinal directions. Its primary use is to determine pointing corrections for use in subsequent scans.

### Parameters

- **location** (*str*) – Catalog source name or Location object. It specifies the source upon which to do the scan.
- **hLength** (*Offset object*) – Specifies the horizontal distance used for the Peak. hLength values may be negative. The default value is the recommended value for the receiver.
- **vLength** (*Offset object*) – Specifies the vertical distance used for the Peak. vLength values may be negative. The default value is the recommended value for the receiver.
- **scanDuration** (*float*) – Specifies the length of each scan in seconds. The default value is the recommended value for the receiver.
- **beamName** (*str*) – Specifies the receiver beam to use for the scan. Make sure that you configure with the same beam with which you Peak.
- **refBeam** (*str*) – Specifies the reference receiver beam to use for the receivers with more than one beam.
- **elAzOrder** (*bool*) – If True, the elevation peak scans will be done first before the azimuth peak scans. This can be helpful for high-frequency observations (>40 GHz) since the elevation pointing offsets are typically larger than the azimuth offsets.

**Caution:** hLength, vLength and scanDuration should be overridden as a unit since together they determine the rate.

## Examples

```
# Peak using default settings and calibrator 0137+3309
Peak('0137+3309')

# Peak using encoder coordinates with scans of 90' length in 30s
Peak('0137+3309', Offset('Encoder', '00:90:00', 0), Offset('Encoder', 0, '00:90:00
→'), 30, '1')
```

**PointMap**(*location*, *hLength*, *vLength*, *hDelta*, *vDelta*, *scanDuration*, *beamName*='1', *start*=1, *stop*=None)

A mapping scan. A PointMap constructs a map by tracking fixed positions layed out on a grid. This procedure does not allow the user to periodically move to a reference location on the sky, please see PointMapWithReference for such a map. The starting point of the map is defined as (-hlength/2,-vLength/2).

### Parameters

- **location** – A Catalog source name or Location object. It specifies the center of the map.



- **hLength** – An Offset object. It specifies the horizontal width of the map. hLength values may be negative.
- **vLength** – An Offset object. It specifies the vertical height of the map. vLength values may be negative.
- **hDelta** – An Offset object. It specifies the horizontal distance between points in the map. hDelta values must be positive.
- **vDelta** – An Offset object. It specifies the vertical distance between points in the map. vDelta values must be positive.
- **scanDuration** (*float*) – It specifies the length of each scan in seconds.
- **beamName** (*str*) – It specifies the receiver beam to use for the scan. beamName can be 'C', '1', '2', '3', '4' or any valid combination for the receiver you are using such as 'MR12'.
- **start** (*int*) – It specifies the starting point for the map. Note in PointMap this counts points, not stripes.
- **stop** (*int*) – It specifies the stopping point for the map. The default is None, meaning “go to the end”.

### Example

The script below produces a 9 point map using a 3x3 grid. Points are separated by 10 arcsec in RA and 10 arcsec in Dec. Each point will be observed for 30 seconds using beam 1 (default).

```
PointMap('W75N',                                     # center of map
        Offset('J2000', 20.0 / 3600.0, 0.00, cosv=True), # 20 arcsec width
        Offset('J2000', 0.00, 20.0 / 3600.0, cosv=True), # 20 arcsec height
        Offset('J2000', 10.0 / 3600.0, 0.00, cosv=True), # 10 arcsec
        ↪horizontal spacing
        Offset('J2000', 0.00, 10.0 / 3600.0, cosv=True), # 10 arcsec
        ↪vertical spacing
        30.0)                                           # 30 second scan
        ↪length
```

**Todo:** Add a plot showing the actual trajectory of the antenna on the sky. Figure 7.5 in the Observer’s Guide unfortunately shows the inverted RA axis, providing the impression the scan is obtained in negative RA direction. The figure itself is correct, but I think it can be misleading to observers who may not pay attention to the x-axis.

**PointMapWithReference**(*location, hLength, vLength, hDelta, vDelta, referenceOffset, referenceInterval, scanDuration, beamName='1', start=1, stop=None*)

A mapping scan. A PointMap constructs a map by tracking on fixed positions layed out on a grid. This procedure allows the user to periodically move to a reference location on the sky. Please see PointMap if no reference location is required.

#### Parameters

- **location** – A Catalog source name or Location object. It specifies the center of the map.
- **hLength** – An Offset object. It specifies the horizontal width of the map. hLength values may be negative.
- **vLength** – An Offset object. It specifies the vertical height of the map. vLength values may be negative.

- **hDelta** – An Offset object. It specifies the horizontal distance between points in the map. hDelta values must be positive.
- **vDelta** – An Offset object. It specifies the vertical distance between points in the map. vDelta values must be positive.
- **referenceOffset** – An Offset object. It specifies the position of the reference source on the sky relative to the Location specified by the first input parameter.
- **referenceInterval** (*int*) – It specifies when to do a reference scan in terms of map points. For example, setting referenceInterval=4 will periodically perform one scan on the reference source followed by 4 pointed scans.
- **scanDuration** (*float*) – It specifies the length of each scan in seconds.
- **beamName** (*str*) – It specifies the receiver beam to use for the scan. beamName can be 'C', '1', '2', '3', '4' or any valid combination for the receiver you are using such as 'MR12'.
- **start** (*int*) – It specifies the starting point for the map. Note in PointMap this counts points, not stripes.
- **stop** (*int*) – It specifies the stopping point for the map. The default is None, meaning “go to the end”.

### Example

The script below produces a 4×4 point map using beam 1 (default). A reference position will be observed before every 2 points. The sequence of scans will be: reference (r) → points 1 and 2 (P1,2) → r → P3,4 → r → P5,6 → r → P7,8 → r → P9,10 → r → P11,12 → r → P13,14 → r → P15,16.

```
PointMapWithReference('2023+2223',                                # center of
↪map                                                             Offset('J2000', 90. / 60., 0.0, cosv=True),      # 90 arcmin
↪width                                                           Offset('J2000', 0.0, 90. / 60., cosv=True),      # 90 arcmin
↪height                                                         Offset('J2000', 30. / 60., 0.0, cosv=True),      # 30 arcmin
↪horizontal step spacing                                       Offset('J2000', 0.0, 30. / 60., cosv=True),      # 30 arcmin
↪vertical step spacing                                         Offset('J2000', 3.0, 0.0, cosv=True),           # 3 degree
↪ref offset in RA                                              2,                                               # reference
↪before every 2 points                                         2.0)                                           # 2 second
↪scan duration
```

**RALongMap**(*location*, *hLength*, *vLength*, *vDelta*, *scanDuration*, *beamName*='1', *unidirectional*=False, *start*=1, *stop*=None)

A mapping scan. A Right Ascension/Longitude (RALong) map performs an on-the-fly (OTF) raster scan centered on a sky location. Scans are performed along the major axis of the selected coordinate system. One can map in a variety of coordinate systems, including J2000, Galactic, and AzEl. The selected coordinate system is defined by the coordinateMode keyword for the Offset object. The starting point of the map is defined as (-hLength/2, -vLength/2) from the specified center location.

#### Parameters

- **location** – A Catalog source name or Location object. It specifies the center of the map.

- **hLength** – An Offset object. It specifies the horizontal width of the map (i.e., the extent in the longitude-like direction). hLength values may be negative.
- **vLength** – An Offset object. It specifies the vertical height of the map (i.e., the extent in the latitude-like direction). vLength values may be negative.
- **vDelta** – An Offset object. It specifies the distance between map rows. vDelta values must be positive.
- **scanDuration** (*float*) – It specifies the length of each scan in seconds.
- **beamName** (*str*) – It specifies the receiver beam to use for the scan. beamName can be ‘C’, ‘1’, ‘2’, ‘3’, ‘4’ or any valid combination for the receiver you are using such as ‘MR12’.
- **unidirectional** (*bool*) – It specifies whether the map is unidirectional (True) or boustrophedonic (False; from the Greek meaning ‘as the ox plows, i.e. back and forth’).
- **start** (*int*) – It specifies the starting row for the map. This is useful for doing parts of a map at different times. For example, if map has 42 rows, one can do rows 1-12 by setting “start=1, stop=12”, and later finishing the map using “start=13, stop=42”.
- **stop** (*int*) – It specifies the stopping row for the map. The default is None, meaning “go to the end”.

---

**Note:** Observers should limit scanDuration so that no more than 2 scans (or accelerations) are performed per minute. Overhead is 20 seconds per scan.

---

### Example

The following script produces a map with 41 rows each 120 arcmin long, using a row spacing of 3 arcmin and scan rate of 20 arcmin/min with beam 1 (default).

```

RALongMap('NGC4258',
    Offset('J2000', 2.0, 0.0, cosv=True),
    Offset('J2000', 0.0, 2.0, cosv=True),
    Offset('J2000', 0.0, 0.05, cosv=True),
    360.0)
# center of the map
# 120 arcmin width
# 120 arcmin height
# 3 arcmin row spacing
# 6 minutes per row

```

**Caution:** Observers should ensure that they are sampling sufficiently in the scanning direction when using OTF mapping. In this example data were recorded every 5 seconds (tint=5.0 in the configuration). This results in one sample every 1.67arcmin in the scanning direction using the above scan rate of 20arcmin/min. This is suitable for observations at 1420 MHz, where the FWHM of the beam is 8.8 arcmin. as the beam will be sampled at least 5 times.

**RALongMapWithReference**(*location, hLength, vLength, vDelta, referenceOffset, referenceInterval, scanDuration, beamName='1', unidirectional=False, start=1, stop=None*)

A mapping scan. A right ascension/longitude (RALong) map performs an on-the-fly (OTF) raster scan centered on a sky location. Scans are performed in the right ascension, longitude, or azimuth coordinate depending on the desired coordinate system. This procedure allows the user to periodically move to a reference location on the sky. If no reference location is needed, please use RALongMap instead.

#### Parameters

- **location** – A Catalog source name or Location object. It specifies the center of the map.

- **hLength** – An Offset object. It specifies the horizontal width of the map (i.e., the extent in the longitude-like direction). hLength values may be negative.
- **vLength** – An Offset object. It specifies the vertical height of the map (i.e., the extent in the latitude-like direction). vLength values may be negative.
- **vDelta** – An Offset object. It specifies the distance between map rows. vDelta values must be positive.
- **referenceOffset** – An Offset object. It specifies the position of the reference source on the sky relative to the Location specified by the first input parameter.
- **referenceInterval** (*int*) – It specifies when to do a reference scan in terms of map rows. For example, setting referenceInterval=4 will periodically perform one scan on the reference source followed by 4 mapping scans.
- **scanDuration** (*float*) – It specifies the length of each scan in seconds.
- **beamName** (*str*) – It specifies the receiver beam to use for the scan. beamName can be ‘C’, ‘1’, ‘2’, ‘3’, ‘4’ or any valid combination for the receiver you are using such as ‘MR12’.
- **unidirectional** (*bool*) – It specifies whether the map is unidirectional (True) or boustrophedonic (False; from the Greek meaning ‘as the ox plows, i.e. back and forth’).
- **start** (*int*) – It specifies the starting row for the map. This is useful for doing parts of a map at different times. For example, if map has 42 rows, one can do rows 1-12 by setting “start=1, stop=12”, and later finishing the map using “start=13, stop=42”.
- **stop** (*int*) – It specifies the stopping row for the map. The default is None, meaning “go to the end”.

### Example

The following script a map that is 120 arcmin long and 60 arcmin wide, using a row spacing of 3’ and scan rate of 4 arcmin/s. A reference position will be observed once before every 3 rows. The sequence of scans will be: reference → rows 1-3 → reference → rows 4-6 ...

```

RAlongMapWithReference('CygA',                                # center of map
                        Offset('J2000', 2.0, 0.0, cosv=True),  # 120 arcmin_
↪length
                        Offset('J2000', 0.0, 1.0, cosv=True),    # 60 arcmin_
↪height
                        Offset('J2000', 0.0, 0.05, cosv=True),   # 3 arcmin row_
↪spacing
                        Offset('J2000', 2.0, 0.0, cosv=True),    # 2 degree ref_
↪offset in RA
                        3,                                       # reference_
↪before every 3 rows
                        30.0)                                    # 30 second_
↪scan duration

```

**Slew**(location=None, offset=None, beamName='1', submotion=None)

An utility scan. Slew moves the telescope beam to point to a specified locaiton on the sky without collecting any data.

#### Parameters

- **location** – Catalog source name or Location object. It specifies the source to which the telescope should slew. The default is the current location in “J2000” coordinate mode.

- **offset** (*Offset object*) – Moves the beam to an optional offset position that is specified relative to the location specified in the location parameter value.
- **beamName** (*str*) – Specifies the receiver beam to use for the scan. beamName can be ‘C’ (center), ‘1’, ‘2’, ‘3’, ‘4’ or any valid combination for the receiver you are using such as ‘MR12’ (i.e. track halfway between beams 1 and 2).

---

**Note:** Once Slew() is complete, the location will continue to be tracked at a sidereal rate until a new command is issued.

---

## Examples

```
# Antenna slews to 3C48 with beam 1
Slew('3C48', beamName='1')

# Slew to 3C48 plus offset
Slew('3C48', ADD OFFSET)

# Slew to offset from current location
Slew(ADD OFFSET)
```

**SubBeamNod**(*location, scanDuration, beamName, nodLength, nodUnit='seconds'*)

An observing scan. For multi-beam receivers SubBeamNod causes the subreflector to tilt about its axis between two feeds at the given periodicity. The primary mirror is centered on the midpoint between the two beams. The beam selections are extracted from the scan’s beamName, i.e. ‘MR12’. The “first” beam (‘1’) performs the first integration. The periodicity is specified in seconds per nod (half-cycle). A subBeamNod is limited to a minimum of 4.483 s for a half cycle.

### Parameters

**location** – A Catalog source name or Location object. It specifies the source upon which to do the Nod.

### scanDuration: float

It specifies the length of each scan in seconds.

### beamName: str

It specifies the receiver beam pair to use for nodding. beamName can be ‘MR12’.

### nodLength: depends on unit of nodUnit (int for ‘integrations’, float or int for ‘seconds’)

It specifies the half-cycle time which is the time spent in one position plus move time to the second position.

### nodUnit: str

Either ‘integrations’ or ‘seconds’.

## Examples

```
# nodLength units in second
SubBeamNod('3C48', scanDuration=60.0, beamName='MR12', nodLength=4.4826624)

# nodLength units are 'tint' as set in the configuration
SubBeamNod('3C48', scanDuration=60.0, beamName='MR12', nodLength=3, nodUnit=
↪ 'integrations')
```

---

**Hint:** The scan will end at the end of the scanDuration (once the current integration is complete) regardless of the phase of the nod cycle. When the subreflector is moving the entire integration during which this occurs is flagged. It takes about 0.5 seconds for the subreflector to move between beams plus additional time to settle on source (total time is about ~1.5 second).

For example, if we had previously configured for Rcvr26 40 and an integration time of 1.5 seconds (tint=1.5 in the configuration), example 2 in script 7.49 would blank roughly one out of every three integrations in a half-cycle (nodLength=3) while the subreflector was moving between beams. If nodLength=5, then only one in five integrations would be blanked. A reasonable compromise in terms of performance and to minimize the amount of data blanked is to use subBeamNod with an integration time of 0.2s and a nodLength=30 (6 sec nodding between beams). It is important to use a small tint value to avoid blanking too much data (e.g., 0.5 sec or less).

The subBeamNod mode is useful to produce good baselines for the measurement of broad extra-galactic lines. For Ka-band, Q-band, and Argus, the performance of subBeamNod is significantly better than Nod observations. For W-band and the KPFA, the beams are farther apart and the subBeamNod technique does not work as well, and users are recommended to use Nod observations.

The antenna uses the average position of the two beams for tracking the target, and SDFITS reports the positions of the beams relative to the tracking position. Although the SDFITS header position will not match the target position, SubBeamNod successfully nods between the two beams during the scan. Control of the subreflector may be done with any scan type using the submotion class. This should only be done by expert observers. Those observers interested in using this class should contact their GBT “Friend”.

---

## TimeObject()

The Time Object is primarily used for defining scan start or stop times. The time may be represented as either a sexagesimal string or in a python mxDateTime object. You can learn more about mxDateTime at <http://www.egenix.com/files/python/mxDateTime.html>.

---

**Note:** One must access the python DateTime module directly from an observation script to generate time objects, i.e., using from mx import DateTime.

---

The Time Object can be expressed in either UTC or LST. The time can be either absolute or relative. An absolute or dated time specifies both the time of day and the date. An absolute time may be represented by either a sexagesimal string, i.e., “yyyy-mm-dd hh:mm:ss” or by a DateTime object. Relative or dateless times are specified by the time of day for “today”. “WaitFor” will treat a dateless time that is more than 30 minutes in the past as being in the future, i.e., the next day. Relative times may be represented by either a sexagesimal string, i.e., “hh:mm:ss” or a DateTimeDelta object. For UTC times, the sexagesimal representation may include a “UTC” suffix. Note that mxDate-Time objects are always UTC. LST time may only be used with relative times and the sexagesimal representation must include a “LST” suffix. Time Objects can have slightly varying formats and can be created in a few different ways.

Some examples are

- Absolute time in UTC represented by a string:  
“2006-03-22 15:34:10”
- Relative time in UTC as a `mxDateTime` object:  
`DateTime.TimeDelta(12, 0, 0)`
- Absolute time in UTC represented by a string:  
“2006/03/22 15:34:10 UTC”
- Relative time in LST as a string:  
“22:15:48 LST”
- Absolute time in UTC as a `mxDateTime` object:  
`DateTime.DateTime(2006, 1, 21, 3, 45, 0)`

## Examples

In this example we will continue to do one minute observations of `srcA` until Feb 12, 2016 at 13:15 UTC when we will then do a ten minute observations of `srcB`.

```
from mx import DateTime

switchTime=DateTime.DateTime(2016,2,12,13,15,0) # Feb 12, 2016, 13:15 UTC

while Now() < switchTime and Now() != None:
    Track(srcA, None, 60)

Track(srcB, None, 600)
```

**Tip**(*location*, *endOffset*, *beamName*='1', *scanDuration*=None, *startTime*=None, *stopTime*=None)

An utility scan. Tip scan moves the beam on the sky from one elevation to another elevation while taking data and maintaining a constant azimuth. It is recommended to tip from 6 deg to 45 deg as the atmosphere will not change significantly above 45 deg.

### Parameters

- **location** – Catalog source name or Location object. It specifies the source upon which to do the scan.
- **endOffset** (*Offset object*) – Specifies the beam’s final position for the scan, relative to the location specified in the first parameter. The offset also must be in AzEl or encoder coordinates.
- **beamName** (*str*) – Specifies the receiver beam to use for the Tip. `beamName` can be ‘C’ (center), ‘1’, ‘2’, ‘3’, ‘4’ or any valid combination for the receiver you are using such as ‘MR12’ (i.e. track halfway between beams 1 and 2).
- **scanDuration** (*float*) – Specifies the length of the scan in seconds.
- **startTime** (*time str 'hh:mm:ss'*) – Allows the observer to specify a start time for the Tip.
- **stopTime** (*time str 'hh:mm:ss'*) – Allows the observer to specify a stop time for the Tip.

**Note:** The scan time may be specified by either a `scanDuration`, a `stopTime`, a `startTime` plus `stopTime` or a `startTime` plus `scanDuration`.

## Examples

```
# Tip the GBT from 6 deg in elevation to 45 deg over a period of 5 min using beam 1
Tip(Location('AzEl', 1.6, 6.0), Offset('AzEl', 0.0, 39.0), 300.0, '1')
```

**Track**(location, endOffset, scanDuration=None, beamName='1', submotion=None, startTime=None, stopTime=None, fixedOffset=None)

An observing scan. The Track scan type follows a sky location while taking data.

### Parameters

- **location** – A Catalog source name or Location object. It specifies the source which is to be tracked.
- **endOffset** (*Offset object*) – Supplying an endOffset object with a value other than None will track the telescope across the sky at constant velocity. The scan will start at the specified location and end at (location+endOffset) after scanDuration seconds. If you wish to only track a single location rather than slew the telescope between two points, use None for this parameter.
- **beamName** (*str*) – It specifies the receiver beam to use for the scan. beamName can be 'C', '1', '2', '3', '4', or any valid combination for the receiver you are using such as 'MR12'.
- **startTime** (*Time object*) – This specifies when the scan begins in the Universal Time (UT). If startTime is in the past then the scan starts as soon as possible with a message sent to the scan log. If (startTime + scanDuration) is in the past, then the scan is skipped with a message to the observation log. The value may be.
- **stopTime** (*Time object*) – This specifies when the scan completes. If stopTime is in the past then the scan is skipped with a message to the observation log.
- **fixedOffset** (*Offset object*) – Track follows the sky location plus this fixed Offset. The fixedOffset may be in a different coordinate mode than the location. If an endOffset is also specified, Track starts at (location+fixedOffset), and ends at (location+fixedOffset+endOffset). The fixedOffset and endOffset must be both of the same coordinate mode, but may be of a different mode than the location. The fixedOffset parameter may be omitted.

---

**Note:** Scan timing must be specified by either a scanDuration, a stopTime, a startTime plus stopTime, or a startTime plus scanDuration.

---

## Examples

```
# Track 3C48 for 60 seconds using the center beam
Track('3C48', None, 60.0)

# Track a position offset by 1 degree in elevation
Track('3C48', None, 60.0, fixedOffset=Offset('AzEl', 0.0, 1.0))

# Scan across the source from -1 to +1 degrees in azimuth
Track('3C48', Offset('AzEl', 2.0, 0.0), 60.0, fixedOffset=Offset('AzEl', -1.0, 0.0))
```

**WaitFor**(time)

An utility function. It pauses the SB until the specified time is reached. The expected wait time is printed in the



observation log including a warning if the wait is longer than 10 minutes. `WaitFor()` will immediately return if the specified time has already passed and is within the last 30 minutes. While `WaitFor()` has the SB paused, it does not prevent the user from aborting. However if the user chooses to continue once the abort is detected, then the `WaitFor()` abandons the wait and returns immediately.

#### Parameters

**time** – A valid time object.

---

**Note:** If a value of `None` is used as an argument to `WaitFor()`, the SB will abort with a message to the observation log. This can occur when passing a value from `Horizon().GetRise()` or `Horizon().GetSet()` when such an event may never occur, such as the rise time for a circumpolar source.

---

#### Examples

The following example will pause the SB until a Local Sidereal Time of 15:13, then wait for the source “1532 3421” to rise above 10 deg elevation, and finally wait for the Sun to set below 5 deg elevation.

```
#Wait for 15:13 LST
WaitFor('15:13:00 LST')

#Wait until source is above 10 deg elevation
WaitFor(Horizon(10.0).GetRise('1532+3421'))

#Wait for the Sun to set below 5 deg elevation
WaitFor(Horizon(5.0).GetSet('Sun'))
```

*DecLatMap()*

*show a tooltip*



## PYTHON MODULE INDEX

### a

`astrid_commands`, [134](#)



## A

Annotation() (in module *astrid\_commands*), 134  
*astrid\_commands*  
 module, 134  
 AutoFocus() (in module *astrid\_commands*), 139  
 AutoOOF() (in module *astrid\_commands*), 141  
 AutoPeak() (in module *astrid\_commands*), 142  
 AutoPeakFocus() (in module *astrid\_commands*), 144

## B

Balance() (in module *astrid\_commands*), 147  
 BalanceOnOff() (in module *astrid\_commands*), 147  
 Break() (in module *astrid\_commands*), 148

## C

ChangeAttenuation() (in module *astrid\_commands*),  
 149  
 Comment() (in module *astrid\_commands*), 149

## D

Daisy() (in module *astrid\_commands*), 149  
 DecLatMap() (in module *astrid\_commands*), 151  
 DecLatMapWithReference() (in module  
*astrid\_commands*), 152

## F

Focus() (in module *astrid\_commands*), 153

## G

GetLST() (in module *astrid\_commands*), 154  
 GetUTC() (in module *astrid\_commands*), 154

## H

Horizon() (in module *astrid\_commands*), 154

## L

Location() (in module *astrid\_commands*), 155

## M

module  
*astrid\_commands*, 134

## N

Nod() (in module *astrid\_commands*), 156  
 Now() (in module *astrid\_commands*), 157

## O

OffOn() (in module *astrid\_commands*), 157  
 Offset() (in module *astrid\_commands*), 157  
 OnOff() (in module *astrid\_commands*), 159  
 OnOffSameHA() (in module *astrid\_commands*), 159

## P

Peak() (in module *astrid\_commands*), 160  
 PointMap() (in module *astrid\_commands*), 160  
 PointMapWithReference() (in module  
*astrid\_commands*), 161

## R

RALongMap() (in module *astrid\_commands*), 162  
 RALongMapWithReference() (in module  
*astrid\_commands*), 163

## S

Slew() (in module *astrid\_commands*), 164  
 SubBeamNod() (in module *astrid\_commands*), 165

## T

TimeObject() (in module *astrid\_commands*), 166  
 Tip() (in module *astrid\_commands*), 167  
 Track() (in module *astrid\_commands*), 168

## W

WaitFor() (in module *astrid\_commands*), 168