ATA Observing Procedures

Jon Richards

August 28, 2019

Table of Contents

Overview 2

Coordinating the Observation 2

Step 1 Coordinate with the team 2

Step 2 Send out a system email 2

Step 3 Reserving antennas with fxconf.rb 3

Single dish using the SNAPs 4

Plan the observation 4

Prepare the dish and feed 4

Beamformer 6

Get the code 6

Plan the observation 6

Performing the observation 7

Where’s the data? 7

Recording the data 7

Viewing live in Baudline 8

Ending a beamformer observation 8

Correlator 8

SNAP On/Off Observations 9

Processing the On/Off Data 9

Long SNAP Observing 10

Processing the Long SNAP Data 11

Useful ATA commands 11

A few of the most useful commands 11

Logging into a feed control board 12

RFSwitch/SNAP Assignments 12

Acronyms 12

# Overview

This documents explains the steps necessary to perform various types of observing at the ATA.

There are three types of observing methods addressed in this document:

1. Single dish using the SNAPs
2. Beamformer
3. Correlator

It is useful to understand there are four independent tuning options available on the ATA:

* LOA -> Connected to the DSA/FRB SNAPs and the NSG RF switches connected to the NSG SNAPs.
* LOB -> Connected to FXA
* LOC -> Connected to FXC, Used by BF1
* LOD -> Used by BF2 and BF3

It is assumed the observer has access to obs@tumulus. In general all command

# Coordinating the Observation

There is a certain amount of coordination required to reserve observing time. This is necessary so multiple users of the array do not try to observe at the same time.

## Step 1 Coordinate with the team

The operator should be in conversation with Elin and Mark regarding their observing or maintenance requirements. Also, the observer should post their intentions on the mailing list [atauser@list.seti.org](mailto:atauser@list.seti.org) in case anyone wants to comment or otherwise use the array for some other purpose.

## Step 2 Send out a system email

An email should be sent to let others know you are using the array.

As obs@tumulus: “atasetalarm –u <user name> -m <message>”

Example:

atasetalarm –u jrichards –m “I am using the array”

“atagetalarm –l” will show the last atasetalarm information

## Step 3 Reserving antennas with fxconf.rb

There is a utility “fxconf.rb” which allows an observer to reserve antennas for use, and also inform anyone using the system which antennas are currently being used.

First, list the groups and the dishes assigned to each of them. The “sals” means “subarray list”

> fxconf.rb sals

bad

bfa

collision 3f 3g 3h

eng 0a

fxa

fxc

fxd

maint 1c 1e 2g 3c 4f

nofeed 1j 2h 2m 3d 4e 4h

none 1a 1b 1d 1f 1g 1h 1k 2a 2b 2c 2d 2e 2f 2j 2k 2l 3e 3j 3l 4g 4j 4k 4l 5b 5c 5e 5g 5h

reserved

As a general matter of practice the dishes are assigned to

* The “none” group if not currently in use.
* The bfa group if to be used by the beamformer of SNAPs
* The fxa group if to be used by the correlator
* Move a dish into the maintenance group if it is having tracking issues
* All other groups are not used

To move a dish to another group “fxconf.rb sagive none bfa 1a 1b 4g”

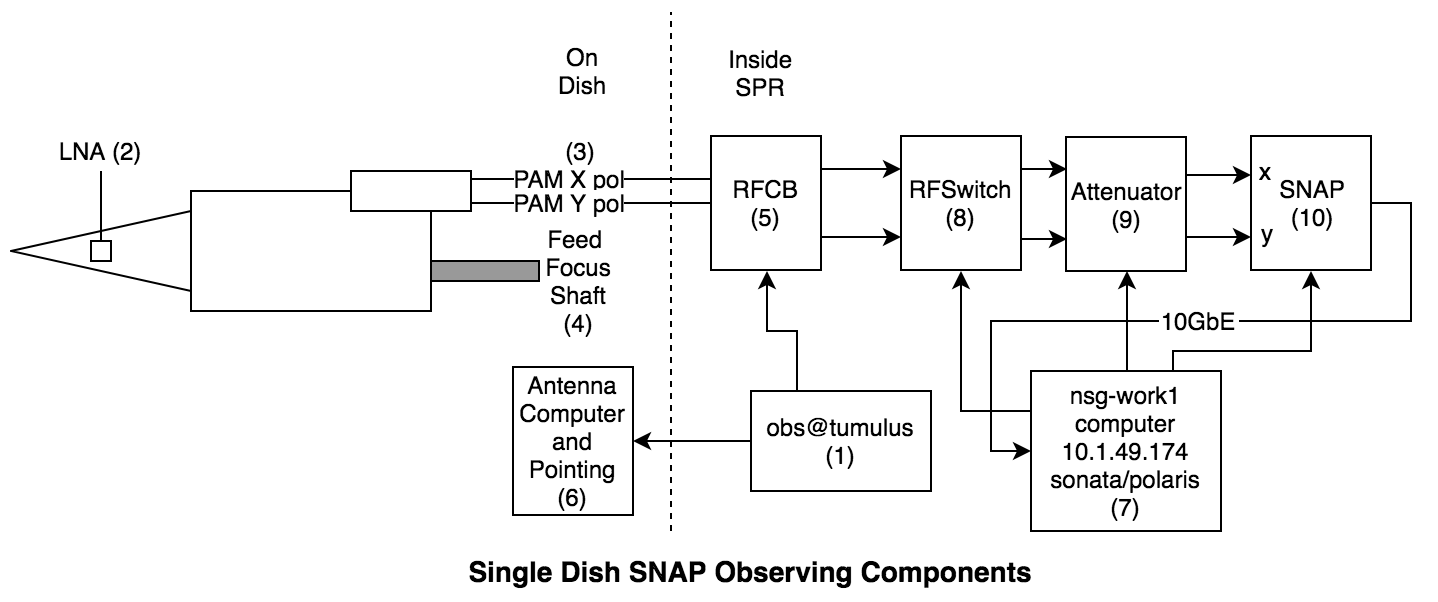
To move the dish back into none after observing “fxconf.rb sagive bfa none 1a 1b 4g”

Note: You can position all the dishes to the parked position (az: 180, el: 18) with one command “park.csh”. You must be obs@tumulus.

# Single dish using the SNAPs

This is an explanation of how to setup observing with one dish so the SNAPs can ingest and process the RF. These steps address the antenna feed all the way to the desired RF output of LOA of the RFCB.

This diagram shows the various components involved in observing with a SNAP and one antenna. Numbers in ( ) are referenced in the steps below.



## Plan the observation

1. What target to observe (goes-16)
2. What frequency (1680MHz)
3. What dish (2a)
4. Which LO (the SNAPs are on LOA)
5. Which SNAP? 2a is on Snap0 (see the **RFSwitch/SNAP Assignments** section)

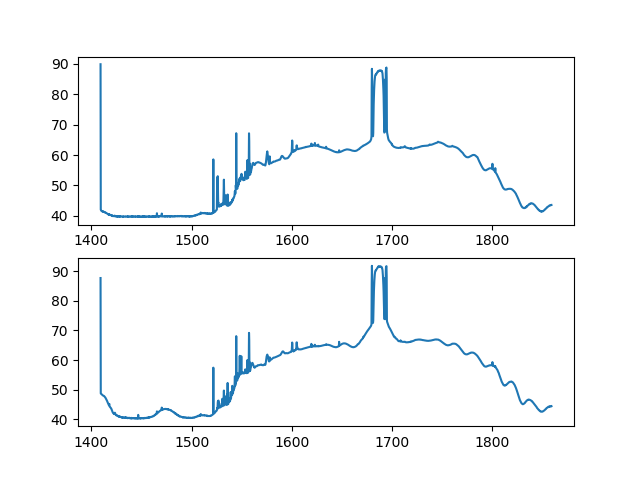
## Prepare the dish and feed

* Log into obs@tumulus (1). All these steps are performed as obs@tumulus.
* Complete the steps outlined in the **Coordinating the Observation** section. Reserve one dish and move it to the “bfa” group. “fxconf.rb sagive none bfa 2a”
* Turn on the feed LNA “atalnaon 2a” (2)
* Set the PAMs to default “atasetpams 2a” (3)
* Focus the feed “atasetfocus 2a 1680” (4)
* Set the frequency “atasetskyfreq a 1680” (5)
* Create the ephemeris “ataephem goes-16”. This creates a file goes-16.ephem. (1)
* Track the target “atatrackephem 2a goes-16.ephem –w” (1 to 6)

Now the dish is pointed and the RFCB LOA output.

* Log into nsg-work1 “ssh -Y [sonata@10.1.49.174](mailto:sonata@10.1.49.174) “, (polaris) (7)
* Tell the RFSwitch to route ant 2a to snap0 “rfswitch 2a” (8)
* Set the attenuation: “atten 20,23 2ax,2ay” (9)
* (7) snap\_plot\_spectra\_live.py snap0 /home/sonata/dev/ata\_snap/snap\_adc5g\_spec/outputs/snap\_adc5g\_spec\_2018-07-07\_1844.fpg -a auto
* In another window adjust the attenuation “atten 20,23 2ax,2ay” (9) to get the levels just right while viewing the live spectra plot. (7)

You should see something like:



# Beamformer

Observing with the beamformer requires a more complicated setup, initialization and calibration procedure. Luckily this has been distilled into a script that handles most of the work. For this reason the observing components diagram is sparse, the observing script takes care of much of the details of setting up the observing.

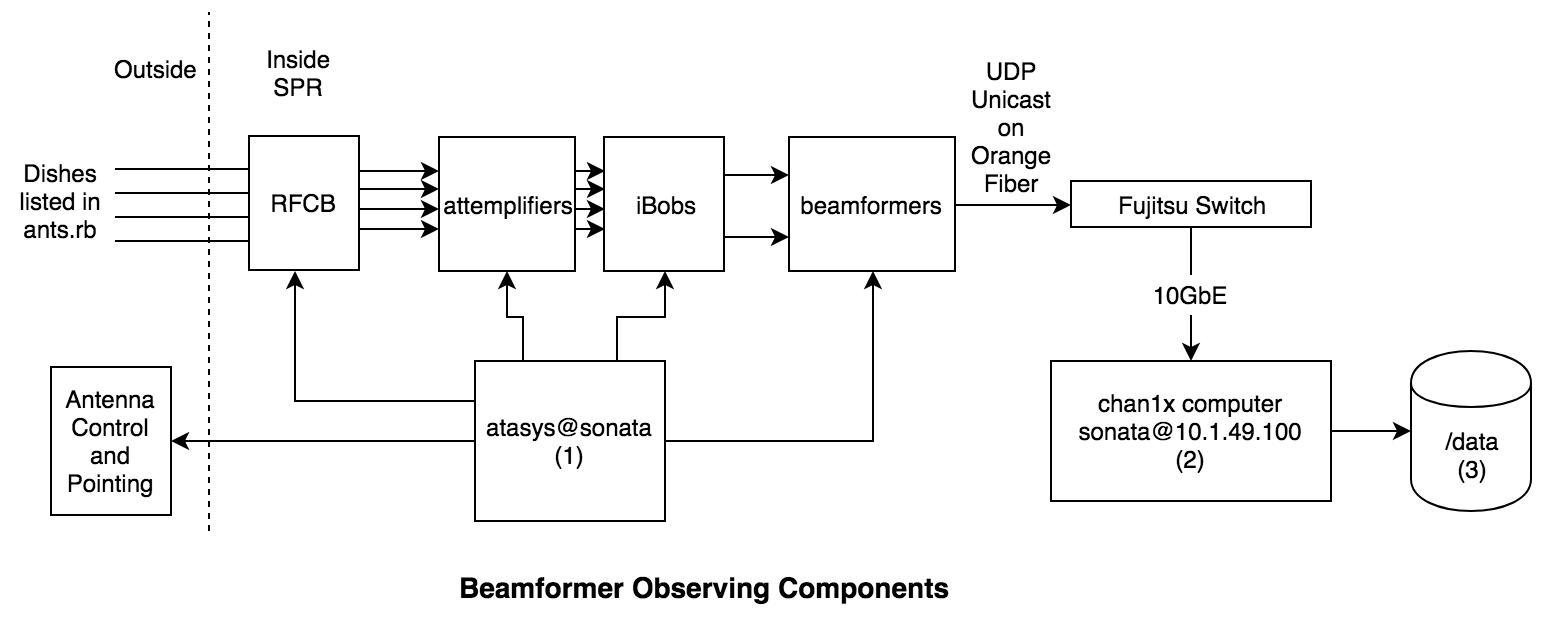
Beamformer observing uses one or more antennas and produces 6 Ethernet UDP data streams. One Ethernet data stream is produced for each beam pol. There are three beams and each beam contains two pols, x and y. That makes a total of six data streams.

Note: The beamformer observing has traditionally been run from atasys@sonata. “sonata” is a computer that has been used for SonATA observing to control the beamformers. Billy says the same scripts can be run on obs@tumulus, but this has not been verified. So log in as atasys@sonata and run from there.

## Get the code

The scripts to use are at <https://github.com/SETIatHCRO/ATA-Utils/tree/master/BeamformerHelpers>

Clone this from github if necessary, or just go to atasys@sonata:jrichards/bf where they currently reside.



## Plan the observation

1. What target(s) to observe (goes-16)
2. How many beams (1 for this tutorial)
3. What frequency (1680MHz)
4. What dishes (1a,2a)
5. What is the target IP address for the resulting beamformer data streams? (10.1.50.100, which is chan1x)
6. What delay calibrator to use. CasA is the best
7. What phase calibrator to use (3c48)

## Performing the observation

1. As usual, follow the steps in the “Coordinating the Observation” section.
2. Edit ants.rb to reflect the antenna pols we wish to use. (1)
3. Create the ephemeris for goes-16 “ataephem goes-16” (1)
4. (1) ./bf\_testing.rb casa 3c48 1680.0 1680.0 goes-16 goes-16 goes-16 testlog1
5. Wait for the script to complete calibrations and say the beamformer is observing the target. This may take 30 minutes or more

## Where’s the data?

At this point a UDP stream should be streaming to the computer sonata@10.1.50.100 (2) on port 50000 for the x pol, 50001 for the y pol. This computer is called “chan1x”

* Log into sonata@10.1.49.100 passwd: polaris. cd to “/home/sonata/chan1x-bcakup/src” (2)
* Verify the data is actually flowing to this computer. “./mrecv –p 50000 –bf” will print out the header details of the first packet it sees. Success! (2)

## Recording the data

Once the data is flowing in you can record the data to disk. There is a large RAID0 disk mounted called “/data” (3).

* cd to /home/sonata/chan1x-bcakup/src/beam2disk (2)
* Start recording the data to disk “sudo ./beam\_rw.sh 50000 /data/<whatever you want the file to be called>.bf”. This will record the x pol. If you want to record the y pol use port 50001. (2)
* Be careful to not overwrite existing data files by entering a previously used data file name.
* Control-C to stop the recording

Keep in mind that this data file will grow at the rate of 13GB per second!

## Viewing live in Baudline

Baudline is installed on chan1x (10.1.49/50.100) and chan2x (10.1.49/50/104)

Once beam data is flowing into the computer:

* cd to /home/sonata/baudline (2)
* “./dxfilter -pin 50001 -stdout | ./baudline\_1.09\_beta6m\_linux\_x86\_64 -lion -channels 2 -format s8 -quadrature –samplerate (2)
* 104857600 -record -fftsize 262144 -overlap 100 -basefrequency 168000000 -flipcomplex -stdin” (2)

## Ending a beamformer observation

* On atasys@sonata, Control-C the bf\_testing.rb script (2)
* Stop the beamformer packetizer so it does not output data over the fiber. “bfspk.rb -b 1 –stop”, “bfspk.rb -b 2 –stop”, “bfspk.rb -b 3 –stop” (2)
* Use “fxconf.rb sagive” to move the antenna back into “none”
* Use atasetalarm to inform the other users that you are finished.

# Correlator

Observing with the correlator is simpler than with the SNAPs or beamformer. There are tried and true scripts that do all the work for you.

Get the necessary script at <https://github.com/SETIatHCRO/ATA-Utils/tree/master/Correlator>

* cd to the location you wish to record the data
* Create a new directory to place the data. Cd to that directory
* copy mosfx-A+C and run into this directory
* edit the run file to include the appropriate values
* ./run

All correlator observations are performed on obs@tumulus.

In the run script you specify the time for the script to end, in decimal hours.

Note: On obs@tumulus there is a /setidata disk to record any correlator data for SETI Institute purposes. Cd to the /setidata disk and run the observation from there.

# SNAP On/Off Observations

The On/Off observing with the SNAPs is easy, it is automated with a script called snap\_obs.py. Download the project <https://github.com/SETIatHCRO/SNAPonoff>

Edit snap\_obs.py to specify which antennas you wish to use. See a dictionary called “snaps” and define the antennas you wish to use. You will need to know which SNAP the antennas are assigned to. See the “RFSwitch/Snap Assignments” section below.

snaps = {

"snap0" : ['2a','2b'],

"snap1" : ['2j','3e'],

"snap2" : ['1c','2h']

}

Then run snap\_obs.py:

* ./snap\_obs.py snap0,snap1,snap2 -o 0,10 -f 1400,2500,3500,4500,5500,6500,7500,8500,9500 -p moon,casa

The –p option specifies which targets you wish to observe. The available options are moon,casa,cyga,taua. The order of the list is important, the target first in the list that is up will be the one observed. If none of the targets are up at the moment the script will wait till the next one is up. This way, you can run snap\_obs.py for weeks on end without having to worry about what targets are up at the moment.

There are several things that you may wish to change:

* **In ata\_control.py an email address is defined. If you give this your email you will get updates whenever the observing automatically changes frequency or target. You may wish to disable this.**
* The database used is on the somata1 computer ([sonata@10.1.49.80](mailto:sonata@10.1.49.80)). You may wish to change this to the Google SQL instance.

Note: If you wish to stop the observing press Control-C. Wait 10 or 20 seconds and it should stop observing and assign all the ants back to “none”. Note that you will need to park the dishes.

## Processing the On/Off Data

The script sefd\_graphs.py in <https://github.com/SETIatHCRO/ATA-Utils/tree/master/SEFDCalc> is used to process the On/Off data and push the graphs and other information to <http://antfeeds.setiquest.info>. After observing, run sefd\_graphs.py and when it finished you can see the SEFD graphs in your browser. This should be run on NSG-head, and is ready to run in /home/sonata/ATA-Utils/SEFDCalc. Look at the code, there are comments at the beginning for customizing the processing.

## Long SNAP Observing

Occasionally long (perhaps hours days) SNAP observations are performed with dishes pointed to a fixed location. This is useful to determine the power stability of a feed over time. It is also useful to determine RFI that repeats over periods of seconds, minutes, hours, or even days.

There is a script called snap\_long\_obs.py located in <https://github.com/SETIatHCRO/SNAPonoff>. Before executing this script you need to point the dish, tune it, etc. The script does not automate any of this for you. This script basically just calls snap\_record.py.

PKL files are created, similar to snap\_obs.py, but each PKL file contains 1000 one second integrations. As a comparison, snap\_obs.py defaults to creating 16 one second integrations.

To prepare on tumulus:

* Point the dish to azimuth 330, elevation 30. This is the standard pointing, but may different depending on your needs.
* Tune LOA
* Set the feed focus
* Turn on the feed LNAs
* Set the feed PAM setting

Now log into NSG-head:

* Create a screen session
* Call rfswitch on NSG-head to route the RF to the SNAP
* Make a data directory, such as ~/data/20190817. You must make a directory.

Now execute snap\_long\_obs.py, for example:

./snap\_long\_obs.py snap1 400 2j ~/data/20190817 10000009 north weekend

This will create PKL files named the following, for example:

~/data/20190817/1566239096\_rf400.00\_n1000\_north\_2h\_on\_weekend\_obsid10000009\_400mhz\_1.pkl

Every 1000 seconds the PKL file is created and populated with the data. So over a night you will get a lot of PKL files.

To stop the long observing, kill the process. Note that you will have to move the dishes into “none” and park the dishes.

## Processing the Long SNAP Data

The Jupyter notebook <https://github.com/SETIatHCRO/ATA-Utils/tree/master/notebooks/OvernightSNAP> shows how to create one big graph of a long SNAP observation.

# Useful ATA commands

As user obs@tumulus there are a lot of ATA control commands. To show a list of the available commands: “atacmds”

## A few of the most useful commands

* atacmds – lists ALL the ata related commands
* ataasciistatus - show the status of all the antennas
* ataephem – creates an ephemeris file
* atacheck – checks the availability of targets
* atatrackephem – sends an ephemeris to antennas
* atasetazel – move a dish to an azimuth and elevation
* atasetskyfreq – sets the frequency for an LO. In MHz.
* atasetfocus – focuses the feed
* atasetpams – sets the PAM values for a feed
* atalnaon – turn on the LNAs of a feed
* [obs@tumulus - park.csh](mailto:obs@tumulus:park.csh) - parks the antennas
* atasetalarm – inform users of your observing status.
* atalistcatalog – list all targets in the ATA RA.Dec catalog. Use grep.
* Get the PAX detector power: “atagetitem 1c DetX” and “atagetitem 1c DetY”

Note: the “set” commands also have an associated “get” command. For instance; atagetskyfreq.

## Logging into a feed control board

* ssh obs@tumulus
* ssh ataant@antcntl
* ssh 1c netcat -v -i 2 rimbox 1518
* You now issue commands to the feed controller board. Try “TC”<return> or “getaccel”<return>

The feed controller board commands are documented at: [https://github.com/SETIatHCRO/antonio-feed-controller-board/blob/master/manuals/ATA%20Cooled%20Feed%20Manual%20Control%20Commands%20Ver%205a.pdf](https://github.com/SETIatHCRO/antonio-feed-controller-board/blob/master/manuals/ATA%20Cooled%20Feed%20Manual%20Control%20Commands%20Ver%205a.pdf" \t "_blank)

## RFSwitch/SNAP Assignments

For reference, here is a list of the available SNAPs and which antennas are connected to them.

NOTE: As of Aug 09, 2019 added 3c, 1c, 1k to SNAP2, replacing 2m, 3d, 4e which have no feeds. Here is the new assignment:

* Snap0: 2a, 2b, 2e, 3l, 1f, 5c, 4l, 4g
* Snap1: 2j, 2d, 4k, 1d, 2f, 5h, 3j, 3e
* Snap2: 1a, 1b, 1g, 1h, 2k, 1c, 3c, 4j, 5e, 2c, 1k, 2l, 2h, 1g, 5g

# Acronyms

NSG – New SETI Group

RFCB – Radio Frequency Converter Board

LO – Local Oscillator. There are 4, named A through D. LOA…LOD

SPR – Signal Processing Room, the main computer room

LNA – Low Noise Amplifier

PAM – Programmable Attenuation Module

RF Switch – The USB controlled RF switch that allows us to choose when antenna pols to input into a SNAP.