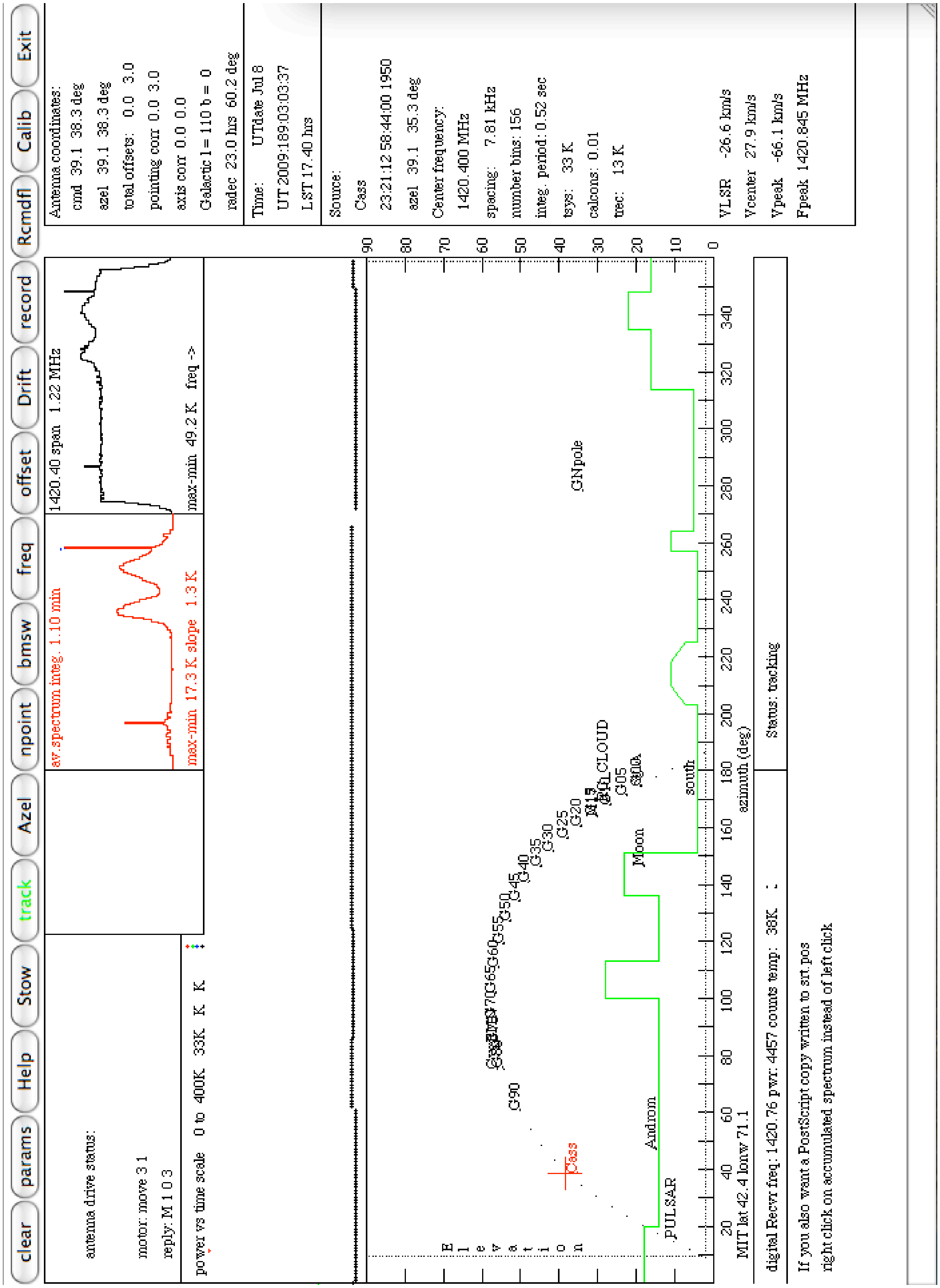
Supplement to the Small Radio Telescope Manual



Socket Version of the Small Radio Telescope Operating Program

The SRT program has two parts: (i) a server program that runs on the computer in 26-630 and communicates with the radio telescope and (ii) a user interface GUI program that communicates with the server via TCP and can, in principle, run on any computer a user wants to run the telescope from. The GUI is written in *java* and should run on any machine with a *java* run-time engine.

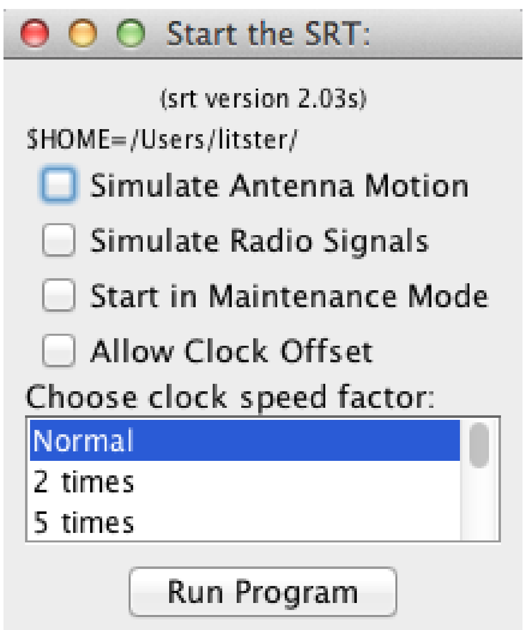
The computer in 26-630 runs Ubuntu Linux and has a Debathena overlay. You log into it with your Atena user name and password as if you were logging into a machine in an Athena cluster; authentication is taken care of by the Athena server and your password is not seen by or stored on the machine in 26-630. When you have logged in, you will find yourself in your home directory on Athena, but files (i.e., programs) on the Debathena machine will be available to you. This first discussion will be about how to run the SRT if you are logged into the computer in 26-630.

Starting the server:
The SRT server should be runs as a service daemon and you should not need to be concerned about it. If you want to check if it is running, there is a shell script to let you know. Just type this command

```
prompt> /usr/local/bin/server-status
```

If the server is not running, an appendix explains how to start it.

The User GUI Interface Program:
Next start the user interface program using the Junior Lab submenu of the Accessories menu at the top left of the screen. When you start the program a small window will open to allow you to choose the startup mode.



Normally you will just click the “Run Program” button to open the main window. (A typical main window is the cover illustration for this document.) Serious errors will be written to the the error logging file, and some errors will also be reported in the main window.

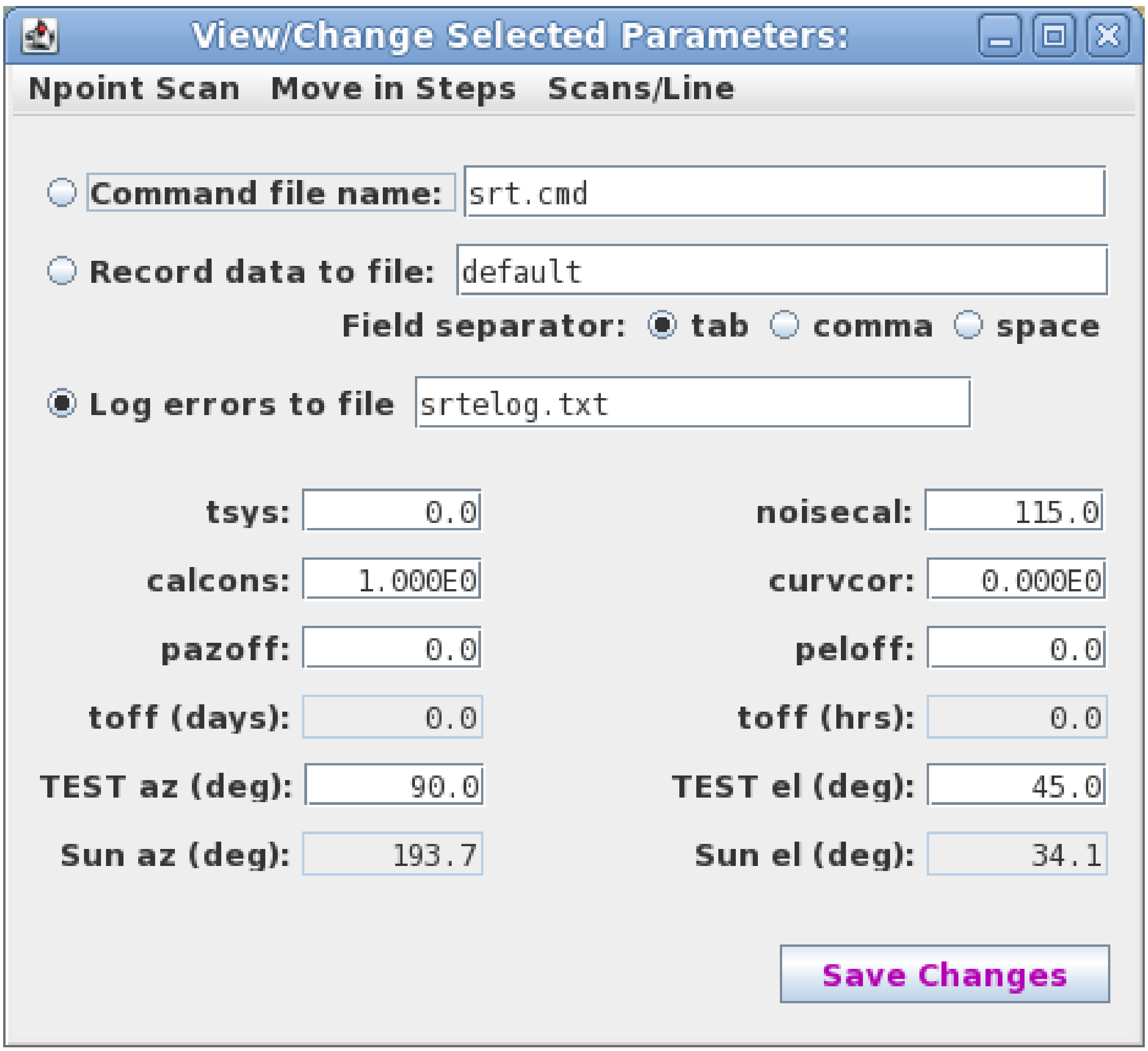
However, the GUI can be run with a simulated connection to the antenna and the radio receiver. In that case the interface will not try to connect to the server and you can run the program without an Internet connection. That can be useful to experiment with how to use the features of the program and to test command files—when it might be useful to offset the clock or run it faster.

To view the SRT TV camera output and see what the antenna is doing, start *Firefox* and open the link `192.168.1.4:80`. (You may need to log in as “`srtuser`” with password “`21cm`”.)

As the user interface (GUI) program will run on any computer it is possible that, for example, a user in the Junior Lab (4-361) may try to run the telescope while it is being run by a user in 26-630 (or vice versa). In that case, the would-be user will be informed the telescope is already in use and the domain name of the computer using the SRT.

The GUI program can do everything that the original Haystack SRT program could do and so you should read the original *SRTmanual* for things that are not discussed in this supplement. How to make and use command files is one particularly important topic.

The current program (it is version 2.03s) has some additional features that should be useful. There is a "parameter" window to make it easier for you to choose record and command file names, turn recording on and off, and run command files. You can do more with an "npoint" scan than was possible before. There are several telescope parameters that you can change in the parameter window. To open the parameter window, click the "params" button in the top row; it should turn green and a window like the one below should open.



The window shows the file names for the command, record and error log files. You may edit the fields if you want to change the files used. (The "default" record file name will use a file with the name `YYDDDHH.rad`.) In the window above, the radio button for the error log file is selected—indicating that error logging is turned on. The radio buttons for the two other files are not selected, indicating that SRT output is not being recorded and a command file is not in use. You may stop/start any of these processes by clicking the appropriate radio button to get the setting you want and closing the window by clicking the `Save Changes` button.

If you close the parameter window by clicking the `params` button on the main window or the $\boxed{\text{X}}$ on the parameter window, none of the changes you may have made in the parameter window will be implemented by the SRT program; to implement the changes close the window with the `Save Changes` button.

The `record` and `Rcmdfl` buttons on the main window still work to start and stop recording output or using command files, but you will have to choose the file names in the parameter window.

Normally the files will all be stored in your Athena home directory. If you would like to store them in a subdirectory, say `srt`, in your home directory, just put a `srt/` in front of the file name you choose.

The parameter window also displays several numerical parameters that you may change, as well as the current coordinates of the Sun—which you may not change. There are three pull-down menus at the top of the window; this is what you can do with them.

Npoint Scan:
With this menu you have some control over what happens when you start an npoint scan with the `npoint` button on the main menu. The default behavior is a 5×5 grid scan centered around the catalog target you have pointed the antenna at; the points in the scan are spaced 1/2 of the antenna beamwidth (the antenna beamwidth is 7° for our telescope at MIT). You may use the menu to change this to 3×3, 7×7, 9×9 or 11×11 grid scans. It seems unlikely that you would want to do a 9×9 or 11×11 grid scan, but further down the menu you have the choice to do either an Azimuth or an Elevation line scan through the target; then you may find the larger dimensions useful.

In all cases, the points are spaced by 1/2 beamwidth, and the program will tell you if the scan would run into a limit switch.

Normally the antenna returns to point at the astronomical object after the npoint scan; the last item on the menu allows you to choose to have it point to the peak of the scan.

Move in Steps:
On the sky plot a red cross hair usually indicates the current antenna pointing spot. When you move the antenna, a blue cross hair shows the target you have sent it towards. Normally, these are not updated until the antenna is pointed at the target, when it will turn red. With this menu, you may move the antenna to the target in various step sizes, and the red crosshair will be updated after each step. This has mostly cosmetic value, but it adds little to the time for the antenna to get to the target and you might like to have a progress report.

Scans/Line:
The SRT normally measures the signal at the antenna for a period of time (about 0.52 s in mode 1), takes a Fourier transform and displays the result in a series of equally spaced channels (typically 7.8125 kHz per channel) on the black plot of the GUI. This takes about 0.52 s in mode 1. I call this process a "single scan" and it will be saved in a single line in the `.rad` output file if you are recording data. Mode 4 takes three single scans and patches them together to get a scan over a wider spectrum; a mode 4 single scan takes about 2 s.

With this menu, you can choose to have the program to make several single scans (1, 2, 5, 10, 20, 50 or 100, to be precise) and display and save only the average of the number of single scans chosen. If you are averaging a number of single scans and tracking a source, the computer will check the tracking every third scan.

The number of scans/line will be recorded as a comment in the `.rad` file and it may also be set in a command file.

Antenna Pointing:
The antenna is pointed by running motors to rotate it about the azimuthal axis or vary the length of the push rod to change the elevation. As the motors move, they send electrical pulses back to the ground controller box on the table beside the computer. (The azimuthal motor, for example, sends back 12 pulses for each revolution of the motor shaft.) The computer will move the antenna by asking the appropriate motor to move a certain number of pulses in the appropriate direction, and it knows where the telescope is pointing by keeping track of the number of pulses returned by each motor. (This is not the best way to do this, but it is significantly cheaper than better methods.)

The antenna orientation is calibrated when the SRT program starts up; it reduces the azimuth and the elevation until the limit switches open at the stow point. (The azimuth and elevation of the stow point are set in the `srt.cat` file, hopefully to correspond to the actual antenna orientation when the limit switches open.)

You may find `pazoff` and `peloff`, which you can set in the parameter window, useful as a private correction for pointing errors of the antenna. When the SRT program starts up `pazoff` and `peloff` are set to zero, but once you set them in the parameter window they will keep the values you set until you change them or the SRT program is restarted.

One way to see if you should set `pazoff` and `peloff` would be to do an npoint grid scan of the sun. You can see if the peak signal position is in the center of the scan and the program will also estimate the offsets of the peak from the center. If you set `pazoff` and `peloff` to these offsets, the signal peak should be in the center of subsequent npoint scans. (Clicking the `offset` button on the main window will also open or close the parameter window.)

If you click on the sky map to ask the telescope to point to an astronomical object, such as the Sun, the program will calculate the azimuth and elevation from the time and date of the computer's clock and the information in the `srt.cat` file; then it will *add* the values of `pazoff` and `peloff`, respectively, to the destination azimuth and elevation coordinates it calculated—which were found using the coordinates of the stow point as a reference. This should correct for pointing errors; then clicking on targets in the sky map should point the antenna in the right direction.

If you set `pazoff=0` and `peloff=3.0`, for example, the red cross that shows the antenna position will appear 3° above the astronomical object that you have selected—as the cross represents where the computer thinks the antenna is pointing from counting motor pulses. The coordinates that appear in the data output files will still be the ones calculated from the time and date information, but they should now be where the antenna is actually pointing. The values of `pazoff` and `peloff` do not appear in the record file, as they were the corrections needed to make the reported coordinates correct.

However, whenever you change `pazoff` or `peloff` in the parameter window, that is recorded as a comment in the record file. The offsets you sometimes see in a record file data output line are the temporary offsets needed to point the antenna during a beamsw or npoint scan of an astronomical object.

Your imagination may suggest other uses of `pazoff` or `peloff`, such as an azumuthal or elevation scan that passes near an astronomical object.

Clock Offset:
When the antenna motion and radio signal are being simulated, you can offset the clock time if you want to see what the sky will look like at a different time or test a command file that runs some time in the future. Just type the offsets you want in the appropriate fields in the params window.

Private Startup:
The GUI program reads its startup information from the file `srt.cat` in `/usr/local/bin`. However if you would like to use a modified `srt.cat` the program will find it and use it instead if you place it in your Athena home directory. A `srt.cat` file that you may copy as well as the compiled *java* files used by the SRT and pdf copies of the original Haystack *SRTManual* and the *SRTManualSupplement* (this file) may be found in the Student WIKI (link on the 8.13 home page).

Remote Operation:

If you need to check to see if the server is running, connect to the computer (jlab-24.mit.edu or 18.109.0.104) using *ssh* (on Windows machines, *Putty* or *SecureCRT*). If you find you cannot log in to jlab-24 remotely, ask your instructor to add you to the group that is permitted to do this.

To check if the server is running

```
prompt> /usr/local/bin/server-status
```

It's a good idea to see who else might be logged in.

```
prompt> who
```

and to check if the SRT GUI program might already be running on the jlab-24 computer

```
prompt> ps ax | grep java
```

Remote Machine Setup:
To run the SRT from a remote computer, say your laptop or one in 4-361, that computer must have a *java* run-time engine, and a copy of the compiled *java* user interface program. For Windows and Linux machines this will be the file `srt.jar`. The program can be started with the command, for example:

```
prompt> java -jar /usr/local/bin/srt.jar
```

It can also be installed to run on Windows machines with a double-click; see the Student Wiki. For Macintosh OS-X machines there is a file "srt.app.zip" which will decompress into an Apple application package that you can run by double-clicking. All of these are available on the Student Wiki.

There should also be a `srt.cat` file in the user's home directory on the machine where the interface program is run. It should have HOST `18.109.0.104` and PORT `1421`) entries.

What *java* thinks is the user's home directory will be shown at the top of the startup window when you start the GUI.

Data record files, log files, and so on will be saved on the computer that is running the user interface program, also in the user's home directory. Otherwise, the procedure is the same as described for the computer in 26-630.

If you are running from a remote location, you can use the TV camera over the Internet to see what the SRT antenna is doing. The URL is `18.109.0.104:80` or `http://jlab-24.mit.edu` and should work with any browser on any computer.

Note: TCP is designed to be a reliable protocol; it's what you use in encrypted form for on-line banking and other transactions. It is unlikely to have problems if it runs on another computer at MIT. However, if the Internet is very busy and you are trying to run the SRT from an off-campus computer, which is normally forbidden, communication with the server could fail. This is more likely to happen when you are using the TV camera, as it uses more bandwidth than the communication between the server and the user interface does. The camera normally sends five frames/sec, and there is a button below the picture that you can click. It will stop the camera sending pictures until you click it again.

If the TCP communication fails, the server program should recognize that, return the telescope antenna to stow position, and then wait for another request from a user interface program—which probably should not be on the computer and at the time when TCP communication failed.

WiFi: the router in 26-630 is there so that your laptop or iPod can have a strong WiFi signal to access the MIT network. The SSID is `JLAB-24`, the encryption is `WEP` and the 64-bit (10-digit hex) key or password is `AE19051921`. (Since Fall 2009, a good MIT WiFi connection is also available.)

Command Files: Examples of command file use are given in the manual for the original SRT program and they also work with this user interface program.

There are two new commands
`: grid [n] [a || e]`
where n is the number of steps (3, 5, 7, 9 or 11) in $\frac{1}{2}$ beamwidths and a or e specify azimuth or elevation line scans. The arguments can be given in any order. If n is omitted, the default will be 5; if both a and e are omitted, you will get a grid scan.

`: repeat n`
where n is the number of scans averaged for each output line in the `.rad` file; see the discussion under the parameter window on page 5 for more information. The number n must be 1, 2, 5, 10, 20, 50 or 100; if anything else it will be set to 1.

Off Campus Use: The server will normally only connect to computers that are on campus (i.e., in the `mit.edu` domain). If you get an error message that says "MIT Only" that is the reason. Your instructor can add your computer's IP address to a special list that should allow you to connect.

Exporting to *Matlab*:
If you try to export the `.rad` files directly to *matlab* as data files, it will not like the comment lines. You can load them into *Excel* or use a text editor to replace the "`*`" with "`%`" characters. I wrote a *java* program called *ParseSRT* to help. It is installed on the computer in 26-630 and there is an icon on the desktop to run it. It does not do much; but under the "`Documentation`" menu there is an item called "`About ParseSRT`" that will tell you what it can do. Two of the capabilities you might find useful are: (i) scan a `.rad` file to find all of the n-point grid scans and plot them (it uses the same algorithm as the GUI interface does and can export the plot as a `.eps` file) and (ii) step through a `.rad` file and plot any scan you choose, similar to the black plot near the top right of the GUI interface, with an option to save the plot as a `.eps` file.

The program will run under Linux, Macintosh, and Windows if you have a *java* run time engine installed. If you'd like a copy for your computer, you can get one from the Student Wiki. Take a copy of the file called `ParseSRT.jar` for Linux or Windows machines. For a Mac, take the file `ParseSRT.zip`.

Cautionary Notes:

- Use care with the buttons at the top of the main window. The computer's response is often determined by which button the mouse pointer passed over last, so it may not be what you expected.
- The digital receiver will saturate (respond nonlinearly to changes in received power) if the count approaches 50 000. The count is shown near the bottom left of the main window, below the 90° point on the azimuth scale.

Appendix: Starting/Stopping the Server

You must be logged into the computer in 26-630 as *root* or use *sudo* to do these things. The protocol is

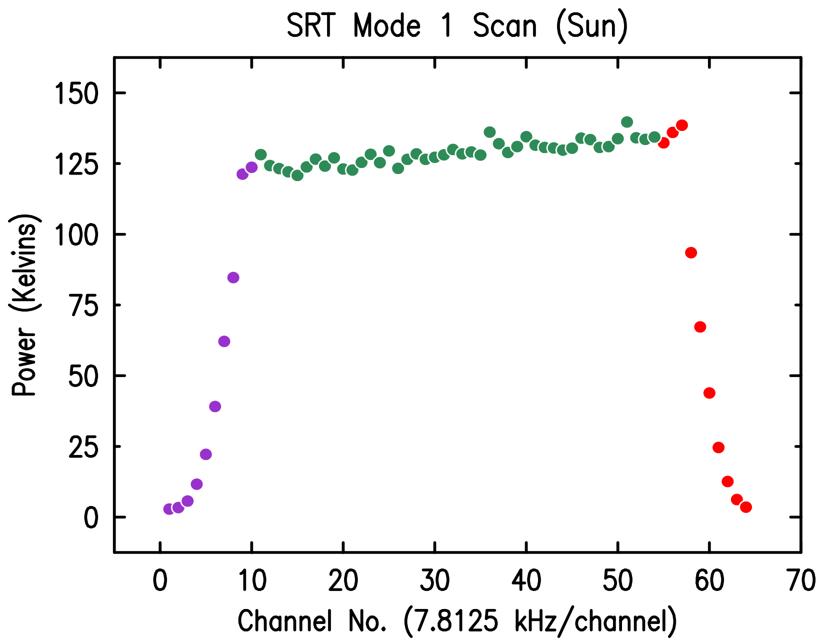
```
prompt> service srtdaemon command
```

where "command" is one of "start", "stop", "status", "restart" or "reload". What they do seems obvious except for the "reload" command. It causes the server to read again the list of special (non-`mit.edu`) IP addresses that the server will connect to, so that the list may be changed without restarting the server.

The server keeps a log of its activities in the file `/usr/local/var/log/serverlog.txt` on the computer in 26-630.

Appendix: Roll-off Channels

The first and last few frequency channels in a measurement are not usable because the receiver gain rolls off. A typical mode 1 scan looks like this.



Normally the SRT GUI program is compiled to ignore the first and last 10 channels when calculating the average power and also when patching three measurements together to make a mode 4 scan. However, if more than the first/last 10 channels were to have low gain then the patching in mode 4 scans could show steps where the measurements join or even a sawtooth shape; some users have seen this type of thing happen. Recent versions of the GUI program *srt* allow the number of roll-off channels to be set by an entry in the `srt.cat` file. The entry should be a line that says `ROLLOFF N` in the file. It will be ignored by the older *srt* program; if it is missing the new program (version 1.90s) will use the default 10 channels. If you experience strange behavior on mode 4 scans, you might want to try changing the number of roll-off channels to 11 or 12. The three measurements that make up a mode 4 scan are taken about a second apart, and if the signal to the receiver were fluctuating rapidly enough there might still be problems patching them together.

The number of channels in a mode 4 scan depends on the number of roll-off channels. The table below gives these numbers.

`ROLLOFF`	Channels Used (`spec[]`)	Mode 4 Bandwidth
10	0–55, 74–119, 138–191	156 channels (1.219 MHz)
11	0–54, 75–118, 139–191	154 channels (1.203 MHz)
12	0–53, 76–117, 140–191	148 channels (1.156 MHz)
13	0–52, 77–116, 141–191	144 channels (1.115 MHz)
14	0–51, 78–115, 142–191	140 channels (1.094 MHz)

The channel numbers given in the center column are those used from each of the three measurements. The first (low frequency) measurement channels are numbered 0–63, the middle (center frequency) measurement channels are numbered 64–127, and the third (high frequency) measurement channels are numbered 128–191. They are patched together so that all channels end up spaced by 7.8125 kHz.

Matching Scans:
The GUI program (version 1.90s) offers three ways to join the mode 4 measurements together, with a `srt.cat` line of the form `MATCH N`, where N is 0, 1 or 2. If N is 0 (the default and original Haystack method) the center frequency powers for each measurement are left unchanged and the three measurements are each tilted about an axis through their center frequencies to try to join them smoothly. If N is 1, the middle measurement powers are not changed and the top and bottom measurements each have a (possibly different) number added to all frequency bins to join them smoothly to the middle measurement. When N is 2, the result is similar except that all power measurements in the top and bottom measurements are multiplied by a constant factor chosen to join them smoothly to the middle measurement.