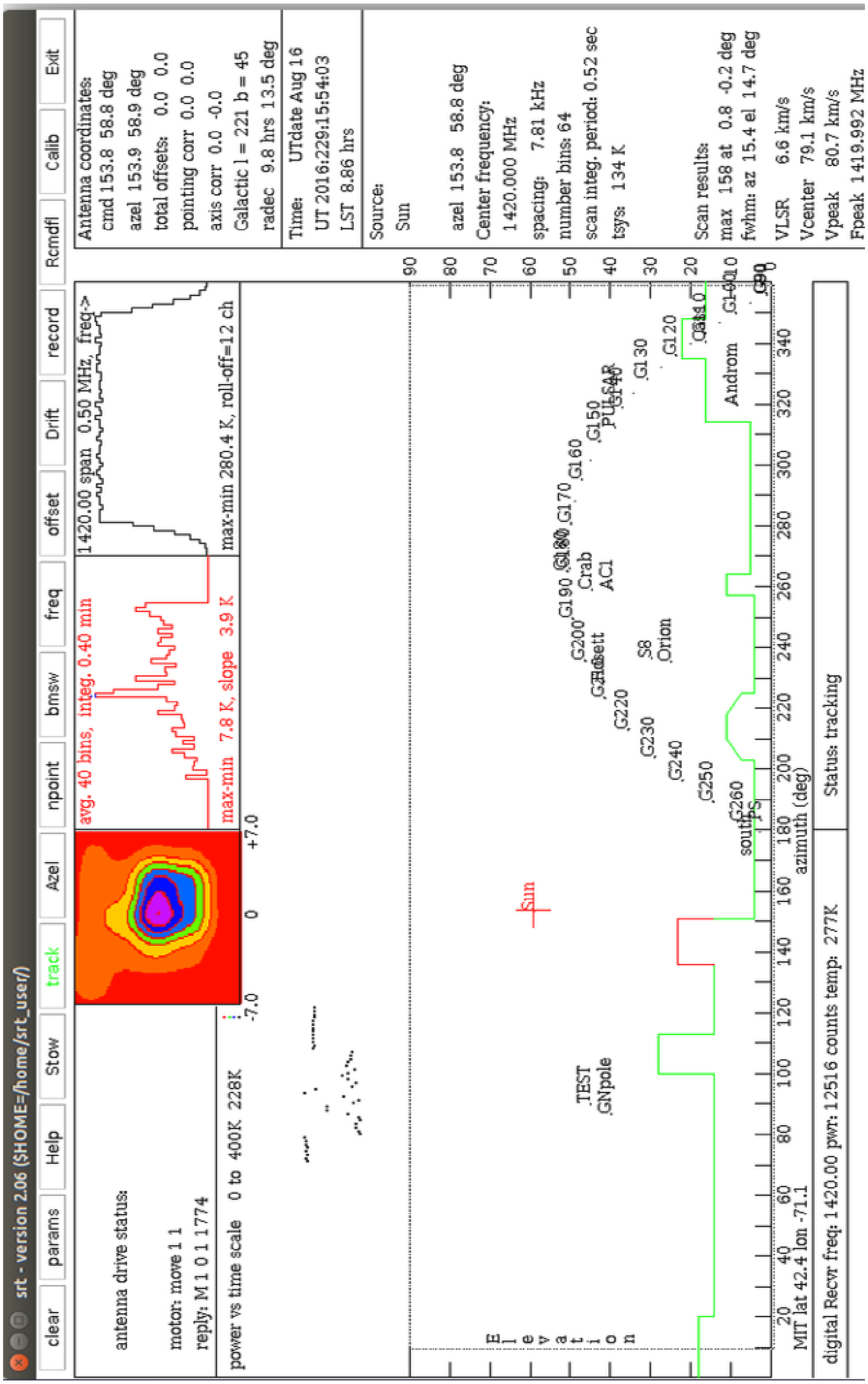
SRT Student Operator Manual



User's Guide to the Small Radio Telescope Operating Program

The SRT program has two parts: (1) a server program that runs on the computer in building 36 and communicates with the radio telescope and (2) a graphical user interface (GUI) program that communicates with the server via TCP (Transmission Control Protocol). The GUI can run on any computer from which a user wants to run the telescope (more on this later). It is written in *java* and should run on any machine with a *java* run-time engine.

The computers that control the telescopes run Fedora Linux. You can log into them through the secure shell with the username "`srt_user`", password "`1420.4MHz`" (see the section on remote access for more).

You can use the telescope camera to see what the SRT is doing. The URL to the camera is `18.108.0.86:80` or `http://jlab-24.mit.edu` and the login is `srtuser` with password `21cm`.

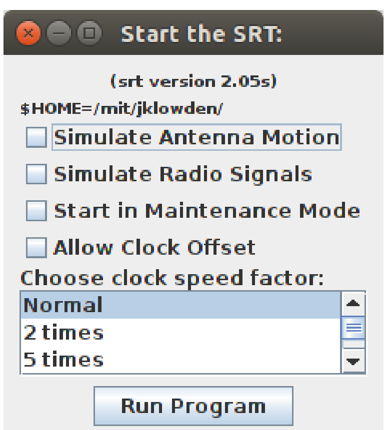
The User Interface Program:

The user interface program is a Java archive file named srt.jar. The most current version is "srt-2.05.jar". On the computer in lab, you can start the program from the command line with the following command, for example:

```
prompt> java -jar /usr/local/bin/srt-2.05.jar
```

The path to the .jar file will depend on where it is on the machine.

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.

However, the GUI can be run with a simulated connection to the antenna and the radio receiver. 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.
Serious errors will be written to the the error logging file, ∼`/srtelog.txt` on the computer running the telescope. Some errors will also be reported in the main window.

Using the Program:

The best way to familiarize yourself with the telescope program is to use it. This section should provide the necessary information to operate the telescope successfully for the 21cm experiment.

The program can do everything that the original Haystack SRT program could do. The user interface is an interactive JAVA-generated window. The window consists of a tool-bar, a text entry command bar, an information side bar listing; times, coordinates, and source information, observing frequency and system temperatures, and a sky-map showing antenna travel limits, azimuth and elevation tick marks, as well as source plots and galactic coordinates. A typical main window is the cover illustration for this document.

Toolbar:
The toolbar is the main way you will control the telescope through the GUI. Use care with the buttons: the computer's response is often determined by which button the mouse pointer passed over last, so it may not be what you expected.

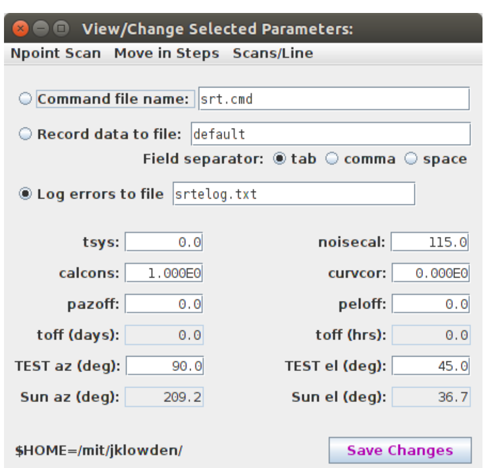
To get started, you'll need to be able to point the telescope and calibrate it. Clicking on a source in the sky map will select that source and tell the telescope to point to it, or you can specify coordinates for pointing.
Azel: Allows the user to enter azimuth and elevation coordinates in the command entry text box. The azimuth and elevation should be separated by a space. The telescope will then slew to that pointing.

Stow: Returns the telescope to the "normal" stow position at 0 Azimuth (pointing North) and 0 Elevation.

Calib: Starts an automatic calibration sequence during which a noise source, located at the apex of the antenna surface, is enabled for approximately 1 second. The system then takes a data sample without the noise source enabled. The resulting system noise temperature is reported in the information side bar as "Tsys". Users will point the telescope at an empty area of the sky for calibration.

You'll also want to do a scan of the sun.
Npoint: Initiates a 25-point scan/map of the source selected. The map is 1/2 beamwidth spaced and when finished displays a false-color, gaussian plot to the left of the accumulated spectrum plot.

Control of further telescope functionality can be found in the other toolbar buttons.
Params: Opens "parameter" window to edit program parameters such as names for the command file, recorded data and error log files (The "default" record file name will use a file with the name `YYDDDHH.rad`.)



The radio buttons next to the file names toggle the use of those files: if the corresponding button is selected, data or errors will be recorded to the file, or the command file will be executed.The `record` and `Rcmdfl` buttons on the main window also work to start and stop recording output or using command files, but the file names are available for editting in the Parameter window.

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.

There are three pull-down menus at the top of the window; this is what you can do with them.

Npoint Scan: Change what happens when you start an npoint scan with the `npoint` button: the default is a 5×5 grid scan centered around the catalog target, but the menu allows users to change this to 3×3, 7×7, 9×9 or 11×11 grid scans, or to do an Azimuth or Elevation line scan through the target. In all cases, the points are spaced by 1/2 beamwidth (the antenna beamwidth is 7° for our telescope), and the program will tell you if the scan would run into a limit switch.

Move in Steps: On the sky plot a red cross hair indicates the current antenna pointing spot, and blue cross hair shows the target you have sent it towards. Normally, these are not updated until the antenna has finished slewing. 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 only a 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.

Rcmdfl: Initiates reading of the default command file (srt.cmd) and begins data recording.

Record: Toggles output file recording on and off. By default, output data files are labeled in the form: `yydddhh.rad`. The file name can be speicified using the `params` window.

Track: Toggles tracking of a source such that the antenna moves over time to correct for Earth's rotation. Track will automatically be on in the latest version of the software; the text should be green. Tracking will also automatically work when a source is selected from a command (.cmd) file.

Drift: Offsets the pointing in Right Ascension and Declination to allow the selected source to drift through the 7-degree beam of the telescope.

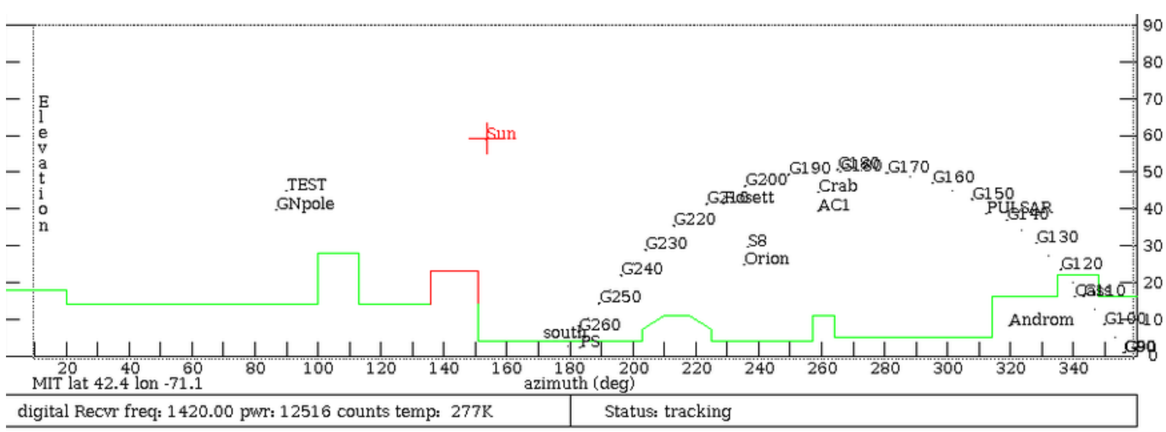
Clear: Clears the control console display of accumulated spectral-line data and n-point scan data.

Freq: Prompts the user to enter the desired settings for center frequency and observing mode. The two observing modes relevant to Junior Lab students are modes 1 and 4. Mode 1 is a single scan with a bandwidth of approximately 0.5MHz. Mode 4 is three Mode 1 scans patched together to give a bandwidth of approximately 1.2MHz. Scans of galactic hydrogen should be made using Mode 4 due to the higher bandwidth.

Offset: Enables the user to enter any az/el offset pair desired. Left click the offsetbutton and enter the offsets in the command entry box. Usage: Azimuth_offset Elevation_offset, default sign is positive.

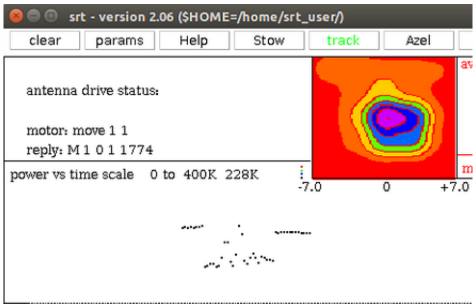
Bmsw: Initiates a continuous "off/on/off" beamswitched comparison observation of the selected source at the frequency settings entered with "freq". The off/on/off measurements are a set and are complimentary, meaning that the two off-source positions switch sign within each set. The off-source measurements are spaced at +/- 1 beamwidth in azimuth (offset = +/- beamwidth/cos EL). "BEAMWIDTH" is set in the srt.cat file. A left mouse click on the bmsw button after the scan has started will abort the beamswitch operation. Results of the observation appear in the left (red) spectral plot window.

Sky Map:
The sky map is the most obvious feature of the GUI. It shows full sky coverage in azimuth and elevation. The elevation scale is exagerated 33% from the azimuth scale. The sources in `srt.cat` are automatically plotted and labeled on the map. Disregard the building outlines plotted on the sky map; they were made for the previous telescope placement on 26.

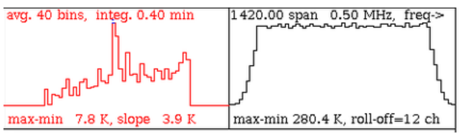


Receiver output and Telescope status:
Two small text boxes just below the map contain information from the receiver.
The left box displays the current: frequency sweep (set in "freq"), receiver "counts" (uncorrected power level detected by the receiver), and the real-time system temperature. The right box briefly describes the status of the telescope: stowed, slewing, or tracking.

Drive Status and Plots:
In the upper right of the control window, a text box displays the antenna drive status and some messages. To the right of the drive status is the plotted result of a 25-point scan of the sun.

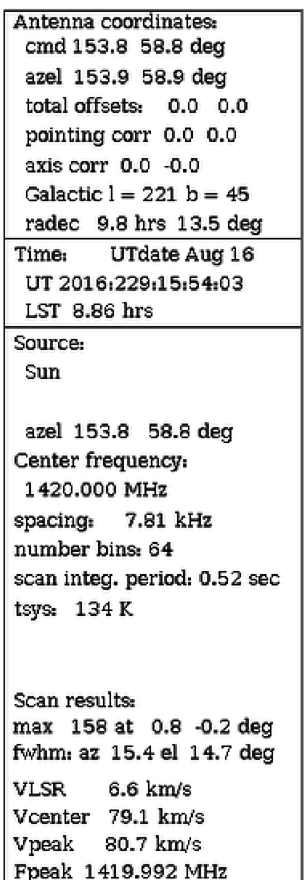


Below this, a continuously running chart recorder shows relative power of the signal received vs. time. The plot will continually overwrite as it plots from left to right and will reset to the bottom of the chart when the plot value reaches the vertical scale maximum. The dots representing power level are color-coded: black 0-400, blue 400-800, green 800-1200, red 1200-1600.



To the right of the sun scan plot, we see the spectral plots. The discrete plot (black, right) is the plot of each individual spectrum as it finishes the user input span in MHz. The accumulated plot (red, left) is the accumulated spectra since the selected observation began.

Information Sidebar:



The information sidebar lists nearly all of the pertinent information the user needs to monitor real time observing with the telescope. The sidebar displays from top to bottom:

Antenna Coordinates:
- Command (computer command)
- Azel (actual position)
- Total Offsets (User input, mapping, etc)
- Pointing Corrections (degrees)
- Axis Corrections (mechanical, user set)
- Galactic Coordinates
- RA and DEC (Hrs and Deg)

Time:
- Universal Time (UT)
- Local Sidereal Time (LST)

Source:
- Source Name
- RA and DEC
- Azimuth and Elevation
- Center Frequency: Input Frequency (MHz)
- Spacing: Bin spacing (MHz)
- Number of bins: From input (Integer)
- Scan integration period (seconds)
- Tsys: degrees K
- Scan Results

Using Command Files:

Command files are text files containing series of commands for the telescope. They allow users to give the telescope a sequence of commands, which it will then automatically execute. They are useful for taking data when you are not actively able to monitor the telescope (e.g.

at odd hours). The command file is ASCII and can accept instruction lines (those that are read and take some action), blank lines, and comment lines (both ignored by the system).

Command file rules:

- Each line must start with a time or : or $*$
- Reads one line at a time skipping blank lines, lines with past times, and lines which start with $*$
- Stops at and executes any instruction line with current or future time.

Time format options:

- UT: line starts with `yyyy:ddd:hh:mm:ss cmd`
- Alternate Local Sidereal Time: `LST:hh:mm:ss cmd`
- Current time: `:  cmd`
- Current time plus n seconds: `:n cmd` (Note the lack of a space between the colon and delay in seconds)

Commands:
```
:  sourcename [n | b]
```
where *sourcename* is any source listed in srt.cat, and the optional mode argument is n for a 25-point scan, or b for beamswitch.

```
:  radec hh:mm:ss | [sign]dd:mm:ss [epoch] [mode]
:  azel az el
:  galactic glat glon
:  offset azoff eloff
```
where coordinates are given in degrees.

```
:  stow
:  noisecal
:  record [filename][0 | 1 | 2 | 3 | 4 ]
```
where the second argument is the recording mode (0 = normal, 1 = short, 2 = add vlsr, 3 = special, 4 = summary only). If filename is ommitted, records will be labeled `yymmddhh.rad` by default.

`:  roff` (turns off record)
`:  short` (suppresses recording cmds)
```
:  freq frequency digitalmode
:  grid [n] [a | e]
```
where n is the number of steps (3, 5, 7, 9 or 11) in 12 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.

Output Data Files:

The output data file (yydddhh.rad) is an ASCII text file. **Comment lines:**
Start with an asterisk and list the STATION LAT and LONG(E/W) on the first line.

```
*  STATION LAT= 42.50 DEG LONG= 71.50
```

When the output data is the result of an input command file, the next comment line could be the listing of the first command line in the input file.

```
* filename.cmd: line 1 : command
```

Calibration results will also follow an asterisk:

```
* tsys 215 calcons 0.98 trecvr 195 tload 300 tspill 20
```

Data lines:
Start with a time mark (yyyy:ddd:hh:mm:ss), then list: azimuth, elevation, offsets, galactic coordinates, start frequency, frequency spacing and data points.

```
2011:205:22:30:38  239.9  20.0   0.0   0.0  249.1  39.3

1419.84 0.00781250  4  152  214 ...
```

Data points listed in the .rad file are multiplied by the calibration constant produced prior to the data taking.

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. Professor Litster wrote a *java* program called *ParseSRT* to help. It is available on the 8.13 webpage for the experiment. 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`.

Remote Operation:

There are two ways to run the experiment remotely: through a copy of the Java GUI on a remote machine communicating with the server on the machine in 36, or by logging into the telescope computer remotely through the secure shell. The former option is the more typical case, so we will address it first.
The 21cm computer in 4-361 runs Debathena and is set up with the most recent version of srt.jar. The program can be started with the command, for example:

```
prompt> java -jar /path/to/the/file/srt-2.05.jar
```

where the path will depend on the computer.

Remote Machine Setup:
To set up another computer to run the SRT remotely (e.g. your laptop) that computer must have a *java* run-time engine, and a copy of the compiled *java* user interface program (srt.jar). There should also be a `srt.cat` file in the user's home directory (what *java* thinks is the user's home directory will be shown at the top of the startup window when you start the GUI). It should have HOST `18.108.0.106` and PORT `1421` entries. A `srt.cat` file that you may copy as well as the compiled *java* files used by the SRT can be found on the Junior Lab webpage for the experiment.

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. Srt.jar can be run as above from the command line. It can also be installed to run on Windows machines with a double-click; see the Student Wiki.

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.

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.

Remote Operation through the Secure Shell:
If you don't want to download the SRT program and set up a remote machine, you can still run the telescope through the computer in 36. You'll need to have the `secure shell` and `X-Windows` installed. On Linux or Macintosh systems, use the Terminal application. On Windows, *Putty*, *SecureCRT*, or *X-Win32* will allow you to use the secure shell.

Once you have your terminal, the command to connect to the computer running the SRT remotely is

```
prompt> ssh -X -p 222 srt_user@jlab-24.mit.edu
```

You will be prompted for the password `srt_user`, which is `1420.4MHz` (this user has sudo privileges, so be careful). The "`-X`" is an option allowing X-forwarding so you will see and control the GUI on the machine you connect from. The port for the SRT computer is specified with "`-p 222`". Run srt.jar as above from the command line of jlab-24; the program is in the home directory.

The important thing to remember is that the data files will be saved to the computer running the GUI, which in this case is the telescope computer. You may wish to retrieve these files, which can be done over the command line with `scp`. Suppose there was a data file `foo.rad` in the directory `/home/srt_user/mydata` on the SRT machine and you were running a terminal window on OS-X or Linux and wanted to transfer it to the directory you were in on your machine.
Here is the command:

```
prompt> scp -P 222 srt_user@jlab-24.mit.edu:/home/srt_user/mydata/foo.rad .
```

Note the dot, which indicates the current directory, and that `scp` specifies the port number with a capital "`P`".
Alternatively, you might want to copy a command file `foo.cmd` on your machine to the SRT computer:

```
prompt> scp ./foo.cmd -P 222 srt_user@jlab-24.mit.edu:/home/srt_user/mydata
```

Again, the dot indicates the current directory. It is best to clear out your files from the SRT computer when you are done with the experiment to preserve its storage space.

Since the user interface (GUI) program will run on any computer it is possible that two users may try to run the telescope simultaneously. 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. This can cause the server to stop responding, necessitating a server restart.

Starting/Stopping the Server:
You must be logged into the computer in building 36 (jlab-24.mit.edu or 18.108.0.86) to do these things, using the above `ssh` login.

To check if the server is running

```
prompt> server-status
```

The protocol to control the server is

```
prompt> sudo systemctl command srtserver.service
```

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 building 36 (JLAB-24).