

Socket Version of the Small Radio Telescope Operating Program

| | |
|--|---|
| Running the SRT server program | 2 |
| Running the SRT user interface program | 4 |
| Simulation Modes | 7 |
| Running a Command File | 7 |
| Exporting Data to <i>Matlab</i> | 7 |
| Cautionary Notes | 7 |

Over the summer of 2009 a new program to run the SRT was installed. It seems to work well enough that I think you should use it. If you find any bugs or have features to suggest, send an email message to litster@mit.edu.

The original SRT program is still available and you should become familiar with its manual, as most of the commands are the same and therefore will not be discussed here. This document is just a supplement that discusses features of the new program.

The new SRT program has two parts; the first part is a server that talks to the telescope over the RS-232 port and runs on the computer in 26-630. The second part is a user program that is similar to the original SRT program: it provides a GUI, stores data record files, logging files, and runs command files. It communicates with the server by TCP, is written in *java*, and can therefore run on any computer equipped with a *java* run-time engine.

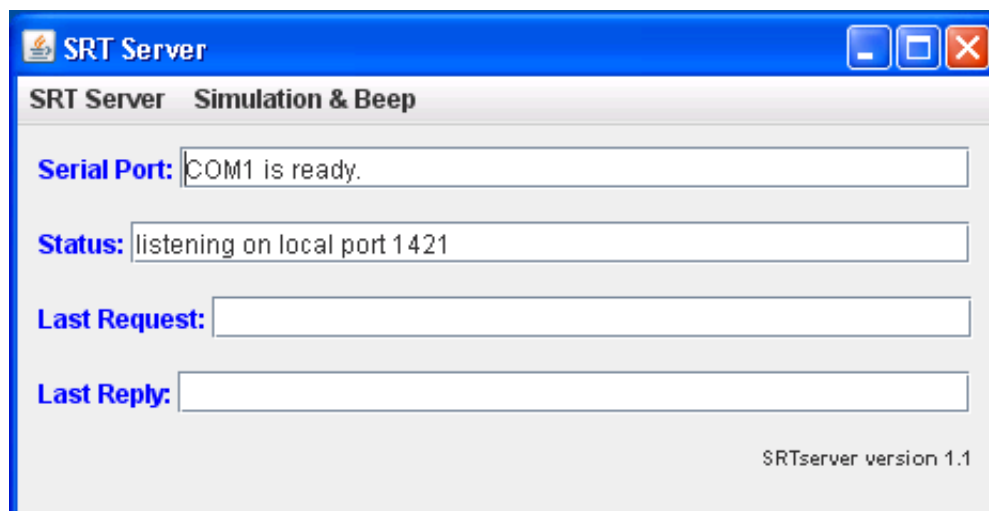
The program has some features that should be useful. There is a “parameter” window to make it easier for you to choose record and command file names, turn recording on and off, and run command files. You can do a lot more with an “npoint” scan than was possible before. There are several telescope parameters that you can change in the parameter window. Finally, most errors are saved in a log file so that you do not have to write them down and I can, *en principe*, identify and fix them.

Starting the server:

The first step is to start the server program if it is not running. To do that, you need to be logged into the computer in 26-630 and the controller box on the table beside the computer must be turned on. Open a Command Prompt window and type the commands

```
prompt> cd C:\SKTcassi\server
prompt> java SRTserver
```

The server should start and, after it has opened the serial port, should give a window like this



As the user interface program will run on any computer, it is possible that 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), for example. In that case, the would-be user will be told the telescope is already in use and the domain name of the computer using the SRT. The actual user's computer will also "beep" if the beep capability of the server is turned on.

The server gives you some information in its window and a menu lets you place it in simulation mode for the antenna pointing and/or radio functions (for testing the link between user interface program and the server without an actual RS-232 connection to the SRT) if you should want that. The menu also enables the "beep" option to be turned on or off. Server error messages are written to the Command Prompt window, and a log of the IP addresses of permitted and rejected client computers is maintained in a log file. You may minimize the Command Prompt window (and the server window, if you wish).

The next step is to start the user control and interface program. The instructions on the next page assume you will be running it on the computer in 26-630.

To run it from a different computer, say one in 4-361, that computer must have a *java* run-time engine, a copy of the compiled *java* code (take all of the `.java` and `.class` files in `C:\SKTcassi`) for the user interface program, and the appropriate configuration files (especially `srt.cat` with the `HOST 18.109.0.104` and `PORT 1421`). Data record files, log files, and so on will be saved on the computer that is running the user interface program. Otherwise, the procedure is the same as described on following pages. Apple *java* is sometimes a bit idiosyncratic. If you have problems, try removing all of the `.class` files and compiling (`javac *.java`). I found that it then works fine when run from a *Terminal* window.

When in 26-630 use the TV monitor to see the antenna, look through the hole in the wall, or send a partner outside and use the walkie-talkies. If you are running from a remote location, you can use the TV camera over the Internet. The IP is `18.109.1.176` and, as it requires *ActiveX*, you can only see it with *Internet Explorer*.

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 will give no problems if the user interface program is on the computer in 26-630 and is very 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 far more bandwidth than the communication between the server and the user interface does. 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`. (As of Fall 2009, a good MIT WiFi connection is also available.)

Running the user interface program:

The program lives in folder `C:\SKTcassi`, which is also the default folder where files will be saved, command files read from, and so on. To run the program, open another `Command Prompt` window and type

```
prompt> cd C:\SKTcassi
prompt> java srt 0
```

That should start the program, which will look and work much like the original one. (A typical main window is the cover illustration for this document.) Serious errors will be written to the `Command Prompt` window and to the error logging file. Some errors will also be reported in the main window. If the main window covers the `Command Prompt` window, you can see it by minimizing the main window.

You will notice a control button called `params` in the top row of the main window. Clicking this button will open a window that allows you to view and change some operating parameters for the SRT. If you click the `params` button it should turn green and a window like the one below will open.

View/Change Selected Parameters:

Npoint Scan Move in Steps

Command file name: srt.cmd

Record data to file: default

Field separator: tab comma space

Log errors to file: srtelog.txt

tsys: 0.0 noisecal: 115.0

calcons: 1.000E0 curvcor: 0.000E0

pazoff: 0.0 peloff: 0.0

toff (days): 0.0 toff (hrs): 0.0

Sun az (deg): 288.4 Sun el (deg): -22.4

Save Changes

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 `YYDDDDHH.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 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. If you want a file that is not in the `C:\SKTcassi` directory, you should specify the full path name for it: e.g., `C:\My Documents\galaxyscan.cmd.` Once you have chosen a record file name you should be able to turn recording to it on and off with the **record** button on the main window.

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 two 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.

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 azimuthal or elevation scan that passes near an astronomical object.

Simulation:

The user interface program offers several simulation modes. To see what they are, start it with no command line arguments:

```
prompt> java srt
```

When the antenna motion and radio signals are both simulated, the program will not try to communicate with a server program. For that, start it as

```
prompt> java srt 1 n
```

where n is an integer. When $n \neq 0$, the program will simulate both antenna and radio. The clock will run faster by a factor $|n|$ and if $n < 0$ the time can be offset from the computer's time. You can specify the offset (days and hours) in the parameter window. This can be useful if you want to see what 21 cm sources will be visible in the sky at future times.

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 is one new command

```
: 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.

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 “Parse SRT” top left menu there is an item called “About ParseSRT” that will tell you what it can do.

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, look in the directory `C:\SKTcassi` on the computer in 26-630. Take a copy of the file called `ParseSRT.jar` for Linux or Windows machines. For a Mac, take the file `ParseSRT.app.tar.gz`.

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 last passed over, 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.

Server Test Mode:

Don't try this unless you know what you are doing.

If you start the server as

```
prompt> java SRTserver -t
```

it will run in test mode. This is intended for maintenance and testing of the SRT antenna motion. In test mode, the server will not listen for TCP socket connections, but it will respond to certain commands typed into the “Last Request” field. The commands it will respond to are `stow` and `move m n` where `m` is the motor number and `n` is the number of steps.

When the server is in test mode a **Move Motor** button will be visible at the lower right of the window. Click this button to ask the SRT to execute the command that you typed in.

To run the server in its normal mode, you will have to restart it. In the normal mode, the server will try to send the antenna to stow position when it starts up and also when you stop the program using the `Quit` item of the pull-down menu under “SRT Server” in the main menu bar. In test mode, it only does the antenna motions that you explicitly ask it to do.