1 Components

The recycling system consists of three programs. These are the server, the graphical client (GC) and the text-based client (TC). This allows the server to always be running, whether recycling is in progress or not. Furthermore, multiple people can view the progress simultaneously. The TC allows nearly anyone to connect to the system, regardless of the computer they are using, while the GC provides a format that facilitates spotting of anomalies and other problems that may occur.

1.1 The Server

The server should always be running whenever SHARC II is expected to be in use. The server is the only component of the recycling system that deals with the instrument. It is responsible for the logging of the temperatures and the control of the Neocera LTC-21.

The server can only be started or halted by a small number of people. In the event of a server crash, one of these people will have to be notified to restart it. Other than the starting and stopping of the server, all commands the server can carry out may be ordered by anyone. Current plans call for access to the server to be via telnet, limited to incoming connections from observatory computers only. (By access, I do not mean the starting and stopping.)

Logging of the temperature data to a file is continuous while the server is running. At the start of a new recycling process, a new file is started and that file is written to until the server is shut down or a new process is started. While not recycling, the server takes readings every 5 minutes. During a recycling, the readings occur every 15 seconds. The log file header contains the time at which the recycling was started and the name of the user to instigate it. Possible additional information is what type of cycle was ordered, but analysis of the logs should provide that information, should it not be included in the header. All logs will be saved to the same directory. The files/directory are world readable, but only the server and those with permissions to start/stop it will have write access.

1.2 The Text-Based Client

The text-based client is essentially a telnet session. It is a shell script that runs "netcat," a program that opens telnet connections, directing the i/o to standard input/output. The main purpose of writing the script is so that the user need not worry about the port number to connect to. The client does no processing of it's own. The user interface to be described is handled solely by the server.

1.2.1 Text-Based User Interface

Upon connection to the server a menu is displayed. The menu is as follows:

- ? - Display Menu

- 1 - Actively Display Readings
- 2 - Display Logged Readings
- 3 - Display Recycling Status
- he3 - Instigate a Helium 3-Only Cycle
- he4 - Instigate a Helium 4-Only Cycle
- full - Instigate a Full Recycle
- halt - Halt the Current Cycle
- help <command> - Display Help on *command*
- q - Quit

Choosing "1" causes the client to receive readings from the server as it takes them. The "readings" consist of two lines. The first line contains column headers, much as the log file does at the top. The second line is the actual reading, as it would appear in the log file. These readings will continue as they are taken until another option is selected. Finally, the current state, along with state description is displayed every time a new state is entered by the server.

Choosing "2" causes the log file to be displayed as if it had been viewed, piped through "more". The difference is that the header of the file appears at the top, but the rest of the file is in reverse order. This is done because the most recent readings are probably the ones that are important. Should the user type q before the entire file has been displayed, the command prompt will return and the viewing of the log will abort.

Choosing "3" gives the status of the fridge recycling. The status can be "logging," or any one of the states of the state machine the server runs to do fridge recycling. Should the status be anything other than "logging," a description of the current state is given after the state number, as well as the name of the user who initiated the recycle. The most recent temperature reading is also displayed in the format used by menu option 1, although this is a one time display.

Choosing "he3" initiates a helium 3-only cycle. Should the server be in any state other than the logging state, the user will be informed that the machine is currently recycling and cannot comply with their command. This command takes an optional parameter, which is the time, in hours, to delay the start of the recycle by. Without this parameter, the recycle starts immediately.

Choosing "he4" behaves exactly the same as "he3," save that it initiates a helium 4-only cycle, rather than helium 3.

Choosing "full" behaves as do the previous two commands, save that it initiates a full recycle. The full cycle consists of a helium 3 pre-cool cycle, should the coldhead be too hot, followed by a helium 4 cycle and then a helium 3 cycle.

Choosing "halt" displays an error message if the server is in logging mode. Otherwise, it confirms that there is a cycle in progress and asks the user to confirm the halt. Upon receipt of a "n," nothing happens. Receipt of a "y" causes the current cycle to stop. Heating of pumps will stop and the server will go into logging mode.

1.3 The Graphical Client

The graphical client consists of 3 waveform displays, labelled "He 3 Pump," "He 4Pump," and "Coldhead." It also contains a few components that display which state the recycling process is in (including the logging state). Finally, it also displays the time at which the current log file was started and the user who started it. Buttons exist to send the "halt," "he3," "he4," and "full" commands, as described in the text-based client section. When one of these other than the "halt" button is clicked, a window appears asking for the time to delay before starting the recycling. The default value is zero and should that value be unchanged, they recycling will start immediately upon clicking upon the continue button. When the halt button is pressed, a window asking for confirmation appears.

Upon starting up, the waveforms are filled in as much as possible from the log file. As new readings are taken in, the waveforms are updated to reflect them. The state components change as the state changes. This client behaves much like the text-based client does when in mode 1, save that the displays are graphical.