EGSE Software
A Guide to the Source

Nicholas Bonham and Roy Smart

June 5, 2017

Contents

1 Revision History

Revision	Date	History	Initial
0.1	06-05-2017	Created	NPB

2 Introduction

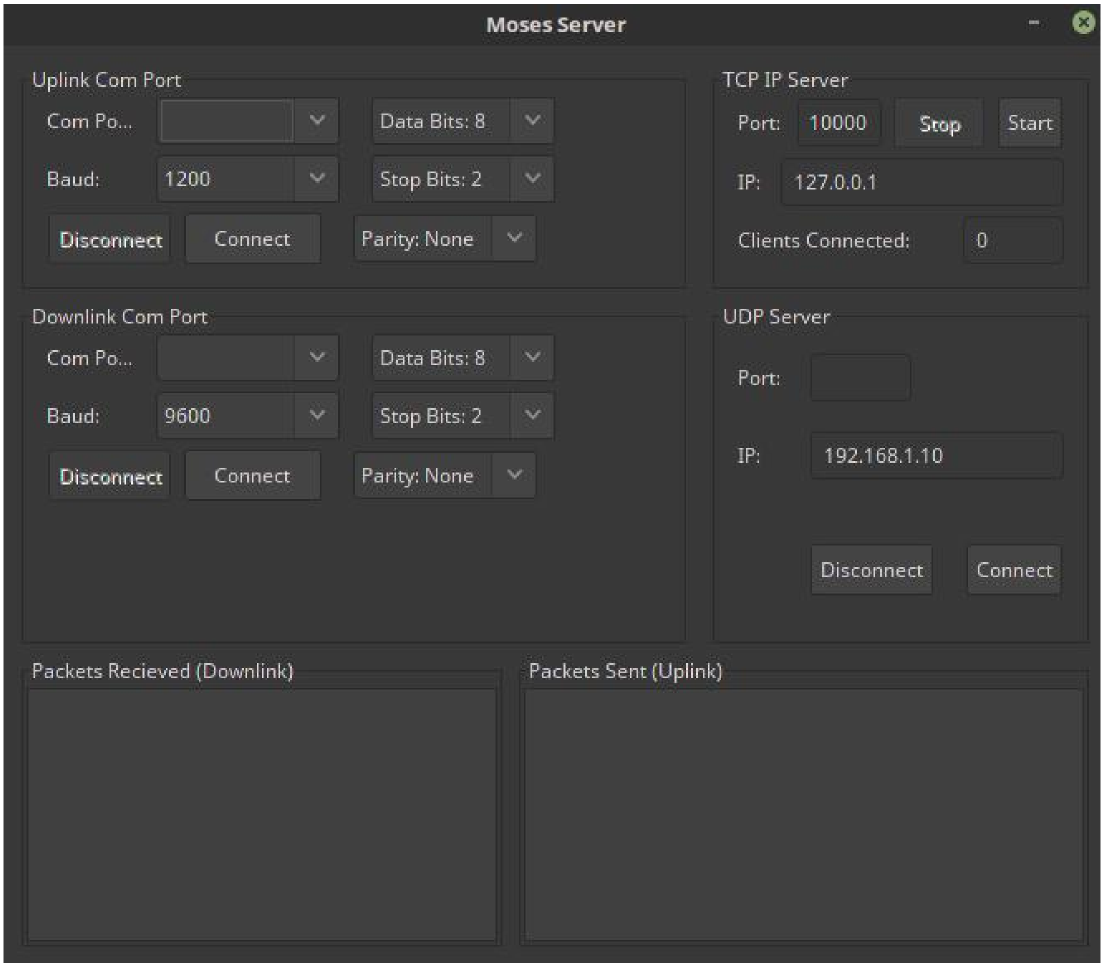
In the summer of 2018, the Kankelborg research group will launch a sounding rocket containing the instruments ESIS and MOSES. Each instrument is designed to image the sun in the Extreme Ultraviolet wavelengths. In order to communicate with the instrument and send commands to and from the instrument during flight, a flight software that operates rapidly needs to be implemented. This flight software is controlled from the ground and sends commands to the instrument via an RS-232 radio connection.

The EGSE server and client are programs written in Java and will be run on EGSE 2, a computer running Linux Mint 17 in the EGSE tower. The EGSE tower consists of multiple computers which run different programs in order to communicate with the instrument as well as display data sent back to the ground. The server and client are used to exchange HLP packets with the flight computer while the instrument is on the ground. The flight computer will be interfaced with EGSE 2 via an Ethernet connection.

The existing EGSE software was written by Matthew Handley with contributions from Roy Smart, Jackson Remington, and David Keltgen. While the existing software was successful, updates were necessary due to the changes in the payload. Rather than sending only the MOSES instrument, ESIS will also be in the payload. The two instruments will be sharing the same up-link interface. To ensure successful communication between the flight computer and both instruments, modifications to multiple sections of the EGSE software were required.

3 EGSE Server

The EGSE server is operated through a graphical user interface (GUI) that contains a slew of panels corresponding to different functions of the EGSE server. The primary function of the server is a gateway for communication between the EGSE client and the MOSES instrument.



Uplink Com Port

Com Po...

Baud: 1200

Data Bits: 8

Stop Bits: 2

Parity: None

TCP IP Server

Port: 10000

IP: 127.0.0.1

Clients Connected: 0

Downlink Com Port

Com Po...

Baud: 9600

Data Bits: 8

Stop Bits: 2

Parity: None

UDP Server

Port:

IP: 192.168.1.10

Figure 1: Screenshot of the EGSE Server GUI.

3.1 Uplink/Downlink Com Port

The Uplink and Downlink Com port panels are designed to detect port interfaces and control the data flow rate, size, and stop bit size. When a port is active, it is displayed as an option that can be selected in the drop-down box. The baud rate drop-down box contains a number of baud rates that can be selected. Baud rate is defined as the rate information is being relayed in a communication channel. Once all of the options are configured appropriately, the `Connect` button establishes the connection to the specified port location.

3.2 TCP IP Server

The TCP IP server is responsible for relaying information sent as TCP packets. This panel gives the user the ability to specify which port and IP address will be used for TCP packets transmission. The `Start` button initiates the connection; the `Stop` button terminates this connection. The `Clients Connected` window displays the number of successful connections to the specified port and IP address.

3.3 UDP Server

The UDP server is responsible for linking a connection for UDP packets to be transmitted. These packets are identical to the commands being sent to the server from the EGSE client, but are transmitted as UDP packets, rather than TCP packets. The UDP port is specified in the `Port` text field. The IP address to be sent to is specified in the `IP` text field.

3.4 Packets Received/Sent Windows

The windows at the bottom of the EGSE server display the packets of information that are either being sent or received by the server.

4 EGSE Client

4.1 Timer

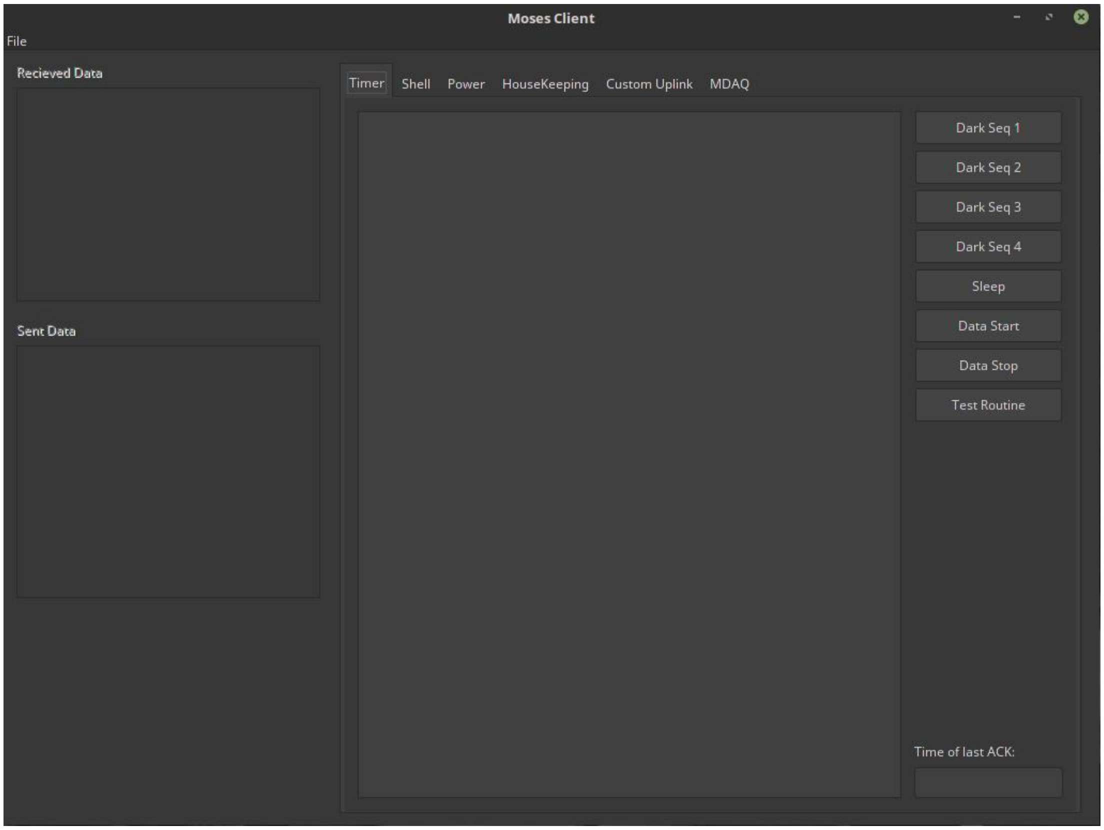


Figure 2: Screenshot of the EGSE Client GUI.

The timer tab in the EGSE client is a section in which the user may send timing commands to the EGSE server, and then to the instrument. The timing commands are as follows:

1. `Dark Seq 1-4`: These links instruct the MOSES flight computer to begin the dark sequence time line which records exposures from the CCDs in the instru-

ment. These sequences take multiple exposures with the shutter closed at varying exposure lengths. This is a necessary task in order to model the dark current from the ROE. Dark sequences are taken at various stages of the mission including while the instrument is on the ground, during the up-leg of the flight, and after the instrument records data with an open shutter. A fourth dark sequence is an optional feature that may be necessary based on the user's discretion.

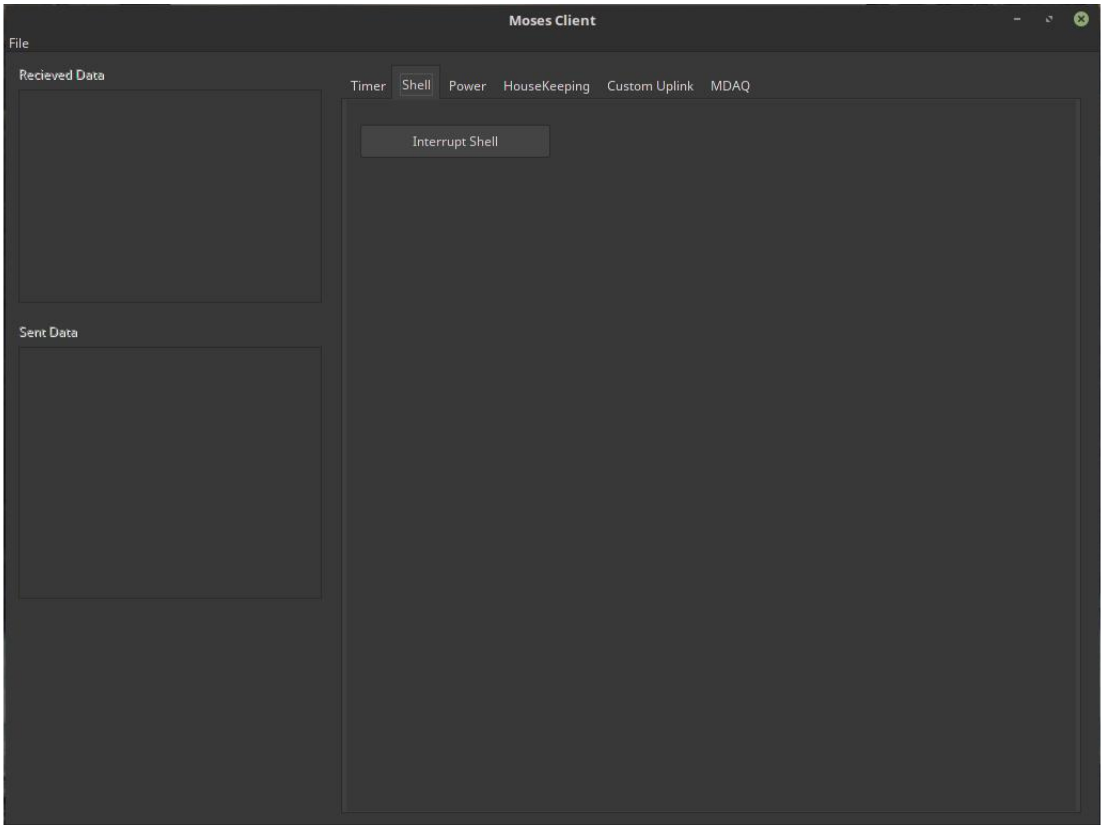
2. `Sleep`: This instructs the flight computer to go into sleep mode, which shuts off some of the components of MOSES until they are needed again.

3. `Data Start`:

4. `Data Stop`:

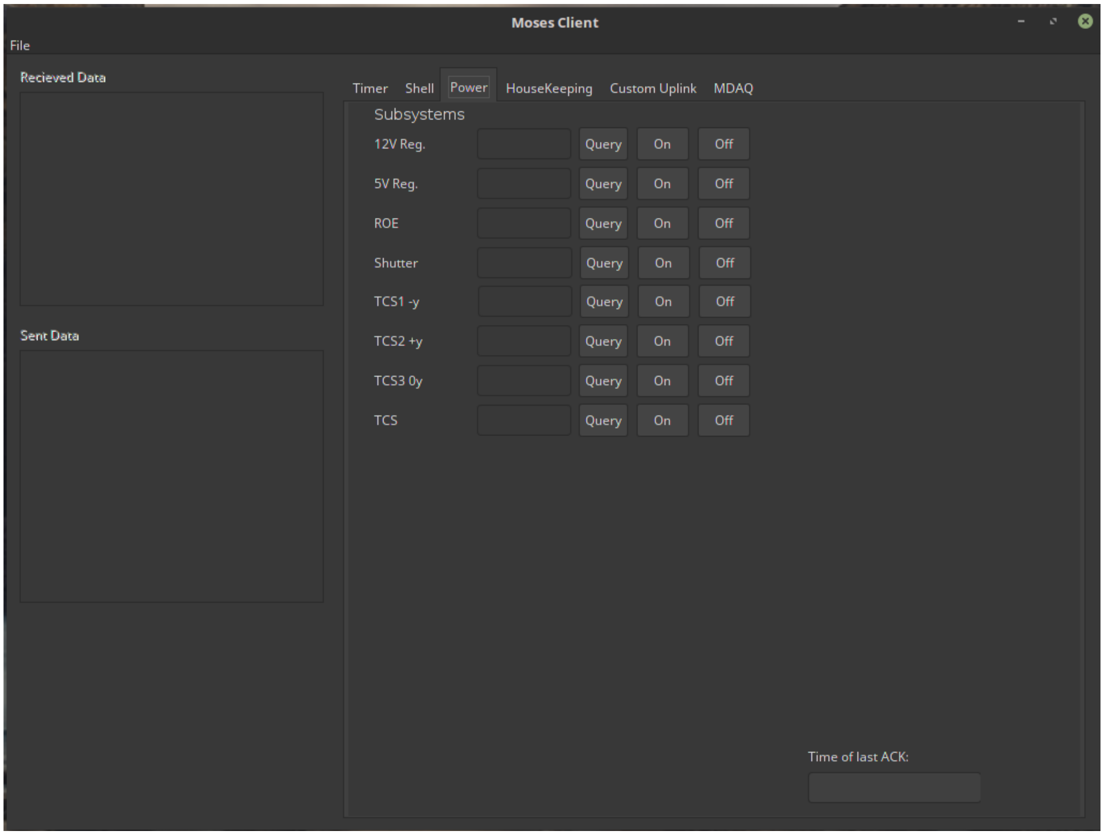
5. `Test Routine`: The test routine effectively pings the flight computer to make sure that it is communicating with the EGSE software.

4.2 Shell



Sometimes, it is necessary to terminate a process, such as if the process is stuck in a loop, or if you suddenly realize you accidentally typed a very dangerous linux command! The `Interrupt Shell` button allows the current process to stop running on the shell.

4.3 Power



The power tab contains commands for the power subsytems. The `On` button turns a specific system on. Likewise, the `Off` button turns a specific system off. Each of the electrical components on the instrument can be controlled from here. The purpose of each electrical system is as follows:

1. `12V Reg.`: The 12 V regulated supply powers many components of the instrument, including...

2. `5V Reg.`: The 5 V regulated supply powers...

3. `ROE`: The read out electronics for the CCDs in the MOSES instrument. These record voltage values corresponding to intensity measurements which form an

image. The ROE contains an analog-digital converter allowing the data to be saved and accessible later on.

4. `Shutter`: The shutter allows light to pass through the instrument. Once the shutter power system is turned on, the shutter can be opened, allowing MOSES to record data.

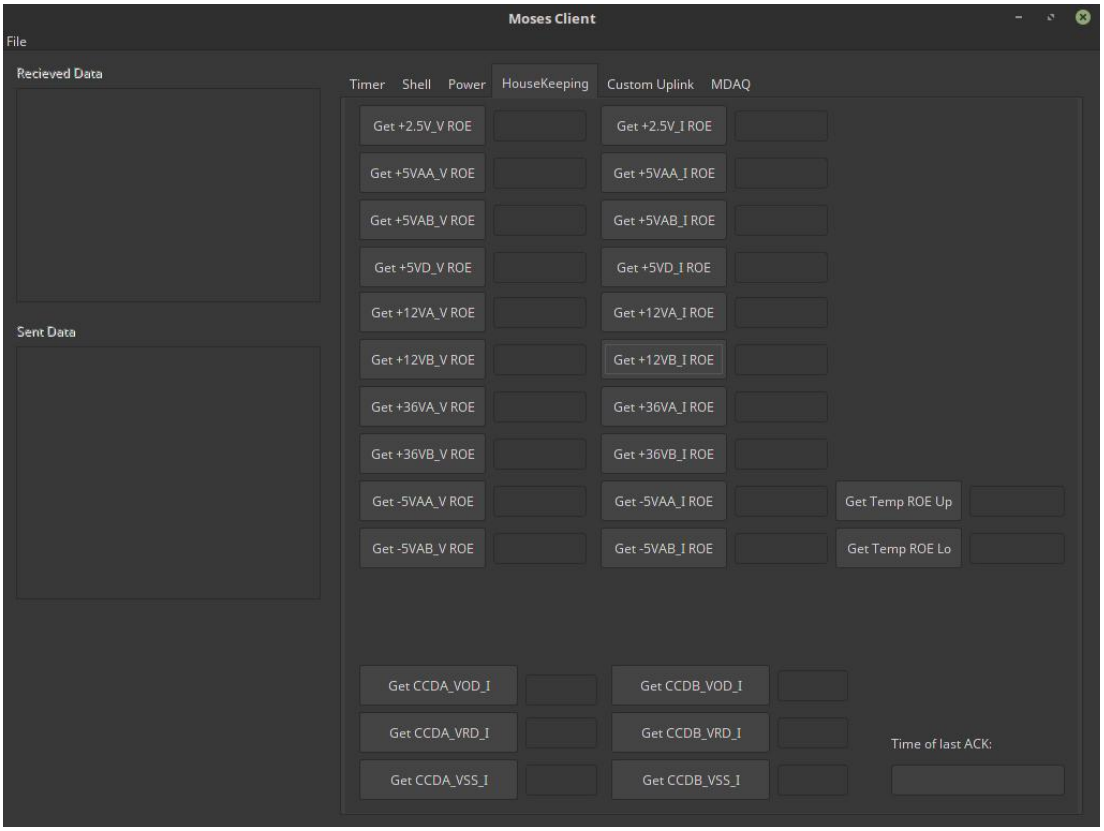
5. `TCS1 -y`:

6. `TCS2 +y`:

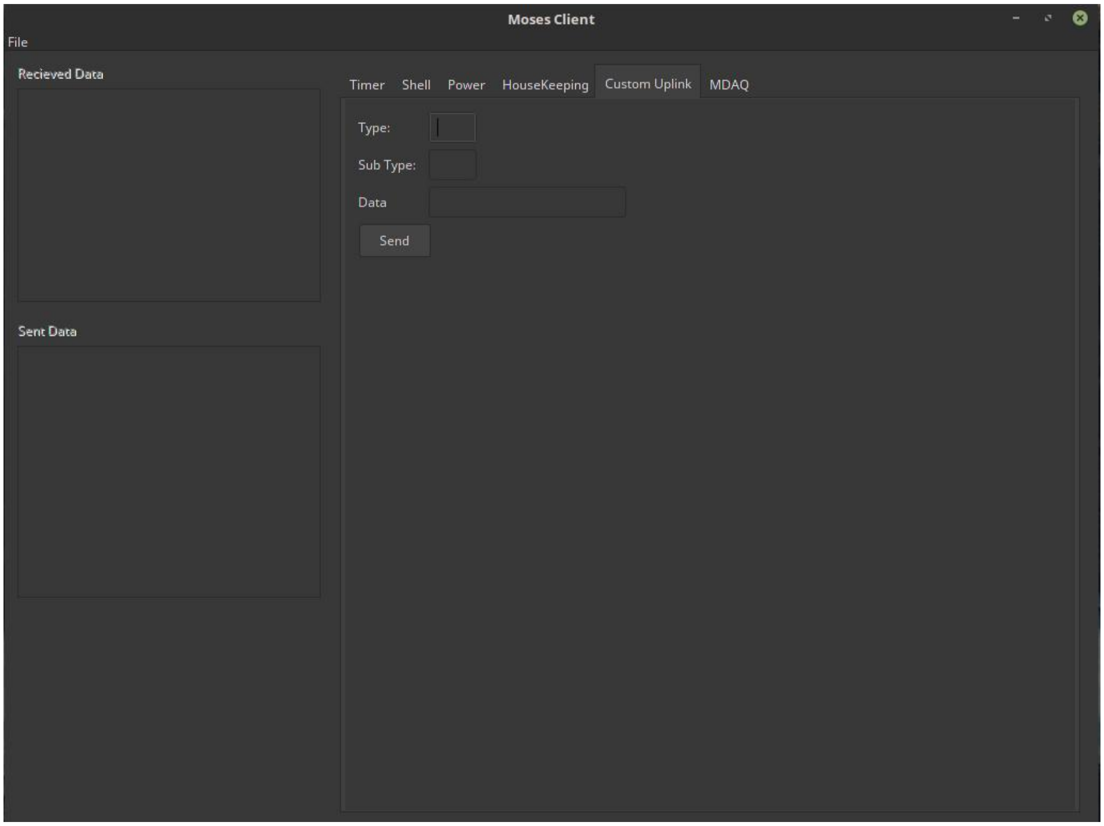
7. `TCS3 0y`:

8. `TCS`:

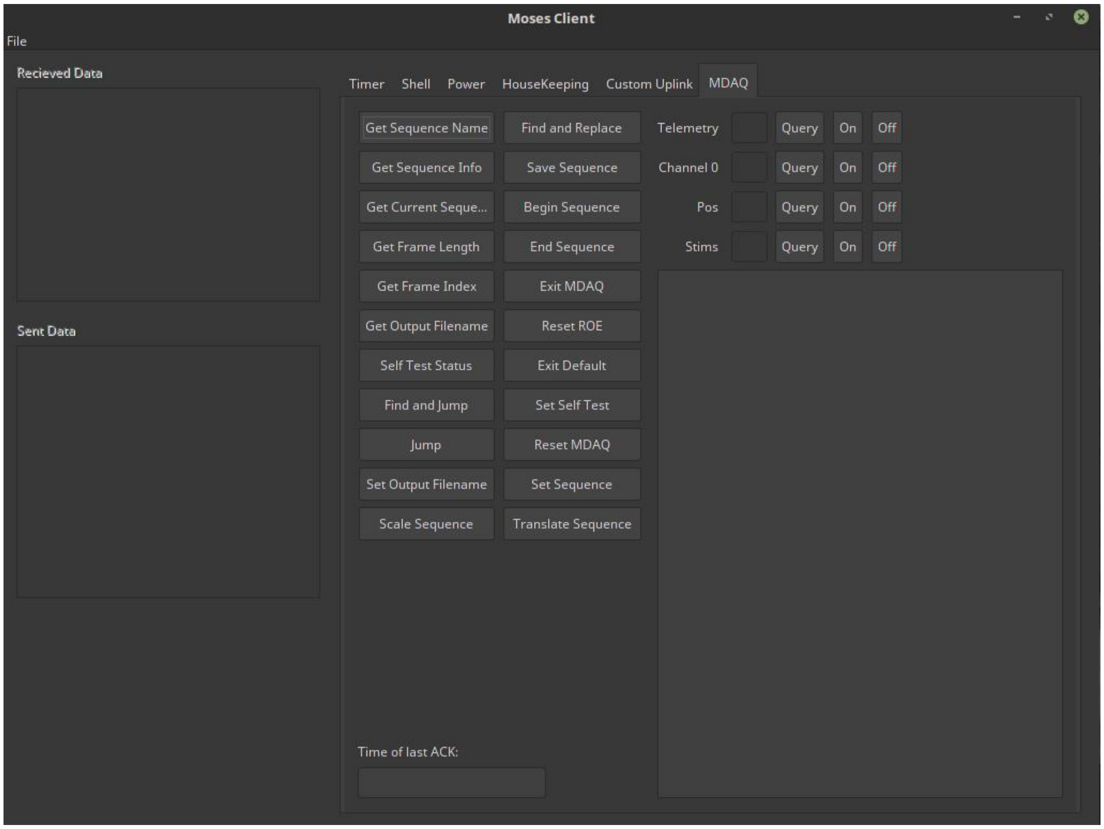
4.4 Housekeeping



4.5 Custom Uplink



4.6 MDAQ


Moses Client
File
Recieved Data
Sent Data
Timer
Shell
Power
HouseKeeping
Custom Uplink
MDAQ
Get Sequence Name
Find and Replace
Telemetry
Query
On
Off
Get Sequence Info
Save Sequence
Channel 0
Query
On
Off
Get Current Seque...
Begin Sequence
Pos
Query
On
Off
Get Frame Length
End Sequence
Stims
Query
On
Off
Get Frame Index
Exit MDAQ
Get Output Filename
Reset ROE
Self Test Status
Exit Default
Find and Jump
Set Self Test
Jump
Reset MDAQ
Set Output Filename
Set Sequence
Scale Sequence
Translate Sequence
Time of last ACK:

5 Setup

5.1 Operation

In order to startup the EGSE software, both the EGSE server and EGSE client need to be started. This can be done through an IDE such as NetBeans, or may be done through the command line in the terminal. Once those are up and running, the TCP transmission on the EGSE server must be initiated. This is done by clicking `Start` under the TCP IP Server panel on the server. Make sure the appropriate port and IP address are specified.

To connect the client to the server, go to `File` and click `Connect`. This selection will bring up a ne window with options for connecting to a specific port and IP address. These must match the specified port and address on the server. A successful connection will print, ”Connected to ⟨IP address⟩:⟨port⟩”. Conversely, an attempt to connect to the server before starting the server, or typing a mismatch in the port or IP fields will result in an error message, ”Couldn’t get I/O for the connection to: ⟨IP address⟩:⟨port⟩”.

Once a successful connection is established, the communication between the server and client may commence. To terminate the connection, press the `Stop` button in the TCP IP Server panel. This will print, ”TCP socket closed”.

6 Dictionary of Abbreviations

1. EGSE - Electronic Ground Station Equipment
2. MOSES - Multi-Order Slit-less EUV Spectrograph
3. ESIS - EUV Snapshot Imaging Spectrograph
4. HLP - Housekeeping Link Protocol
5. TCP - Transmission Control Protocol
6. UDP - User Datagram Protocol
7. CCD - Charge-Coupled Device
8. ROE - Readout Electronics
9. MDAQ - Mission Data Acquisition