



Space Traffic Management Conference

2016 Emerging Dynamics

Nov 16th, 5:30 PM

The Design and Capabilities of OSCOM System

Patrick Rupp

Student, Embry-Riddle Aeronautical University, ruppp@my.erau.edu

Joseph Stroup

Student, Embry-Riddle Aeronautical University, stroupj@my.erau.edu

Yevginy Lischuk

Student, Embry-Riddle Aeronautical University, lischuky@my.erau.edu

Sergei J. Balardi

Student, Embry-Riddle Aeronautical University, bilardis@my.erau.edu

Forrest W. Gasdia

Student, Embry-Riddle Aeronautical University, gasdiaf@my.erau.edu

See next page for additional authors

Follow this and additional works at: <https://commons.erau.edu/stm>

Rupp, Patrick; Stroup, Joseph; Lischuk, Yevginy; Balardi, Sergei J.; Gasdia, Forrest W.; and Barjatya, Aroh, "The Design and Capabilities of OSCOM System" (2016). *Space Traffic Management Conference*. 5. <https://commons.erau.edu/stm/2016/posters/5>

This Event is brought to you for free and open access by the Conferences at Scholarly Commons. It has been accepted for inclusion in Space Traffic Management Conference by an authorized administrator of Scholarly Commons. For more information, please contact commons@erau.edu.

Presenter Information

Patrick Rupp, Joseph Stroup, Yevginy Lischuk, Sergei J. Balardi, Forrest W. Gasdia, and Aroh Barjatya

Patrick Rupp
(ruppp@my.erau.edu)

Yevgeniy Lischuk
(lischuky@my.erau.edu)

Joseph Stroup
(stroujp@my.erau.edu)

Sergei Bilardi
(bilardis@my.erau.edu)

Forrest Gasdia
(forrestgasdia@gmail.com)

Aroh Barjatya
(aroh.barjatya@erau.edu)

Abstract

Optical tracking and Spectral characterization of CubeSats for Operational Missions (OSCOM) is a system designed at Embry-Riddle Aeronautical University that uses custom software and inexpensive commercial-off-the-shelf (COTS) equipment to perform space situational awareness tasks for satellites ranging in size from the International Space Station (ISS) to CubeSats and small orbital debris. The OSCOM system is primarily designed to perform high frame rate photometry but also is capable of resolved imaging of larger satellites. In this poster we present the current hardware and software system overview, as well as our plans for fully automating the tracking and data acquisition for multiple systems deployed around the world.

System Overview

OSCOM's current system consists of a variety of instrumentation that includes multiple telescopes, telescope mounts, machine vision cameras and several computers to track and image satellites. The system is portable in a car's trunk and therefore is capable of being field deployed to perform simultaneous multi-point observations. Custom software is used to interface with the telescope mount to track satellites and later process image data and perform photometry. The current version of the tracking software is being upgraded to provide more functionality and maintainability.



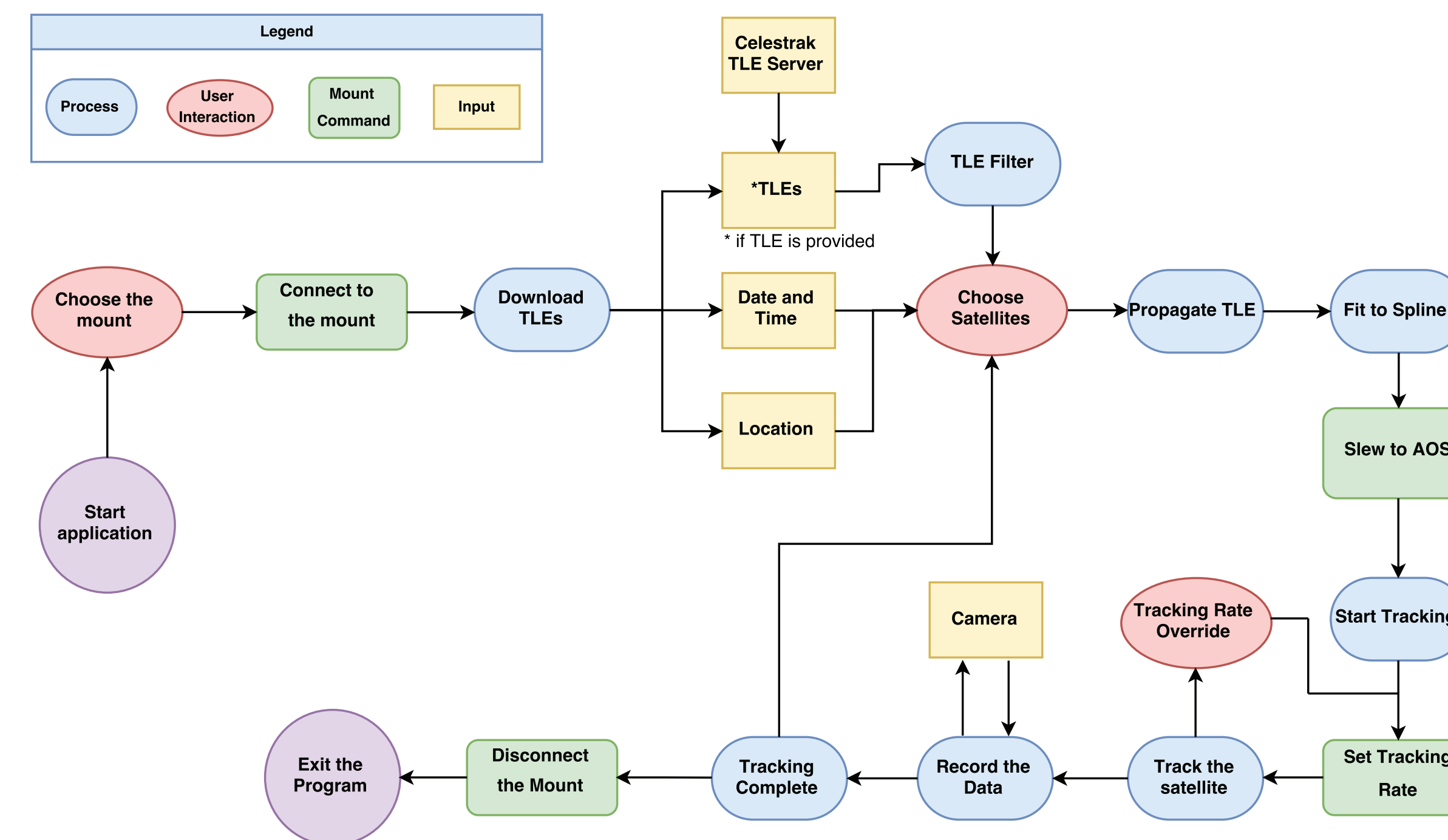
The image on the left shows the OSCOM system mounted to a pier on the veranda of the College of Arts and Sciences building at Embry-Riddle Aeronautical University's Daytona Beach campus while the image to the right shows the OSCOM system deployed in the suburbs of Massachusetts.

Hardware

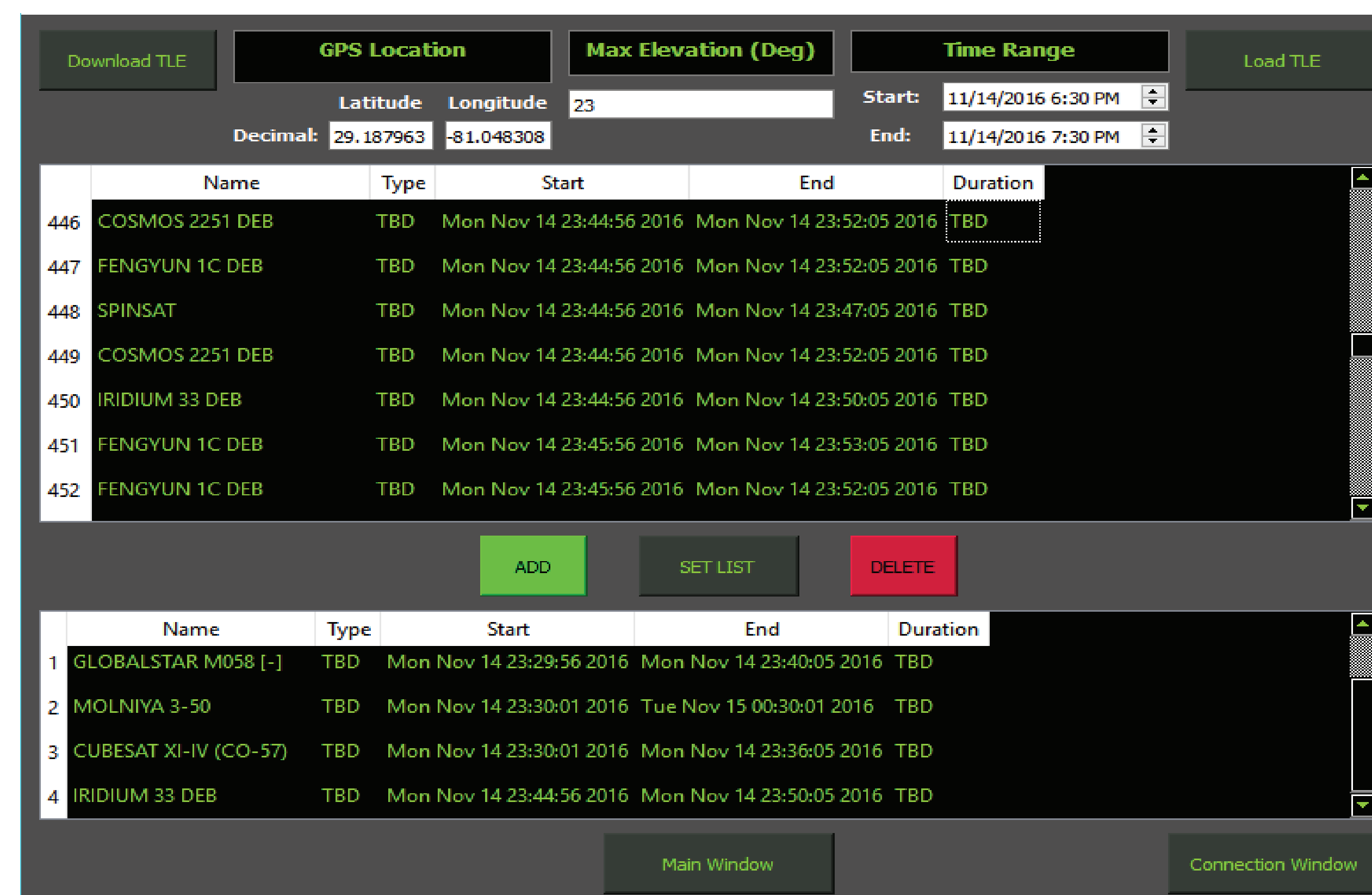
OSCOM's hardware elements include carefully chosen COTS equipment to maximize the throughput while yielding a relatively low, total cost system compared to other systems that perform similar functions. OSCOM's primary optical system consists of a Celestron 11" Rowe-Ackerman-Schmidt Astrograph (RASA) with a machine vision CMOS camera from Allied Vision. The camera is capable of sub-frame capture that results in high frame rate, time resolved photometry of satellites. Telescopes such as the CDK 20 have also been used to perform resolved imaging of objects such as the International Space Station (ISS). The 11" $f/2.2$ RASA provides high throughput and wide field imaging, allowing dim objects to be detectable and enabling easier object tracking than what could be performed on telescopes with smaller field of views. For bright objects, filters can be used to provide photometry of a specific spectral band. The optical system is installed onto a computerized telescope mount. This enables custom software to send commands to the mount to track satellites. OSCOM primarily operates on a Bisque Paramount, but other mounts, such as the Losmandy G11, have also been used for satellite tracking. The OSCOM system can be used for portable operation, shown in the images above, and placed either on the veranda of the College of Arts and Sciences building at Embry-Riddle Aeronautical University's Daytona Beach campus or in the field.

Software

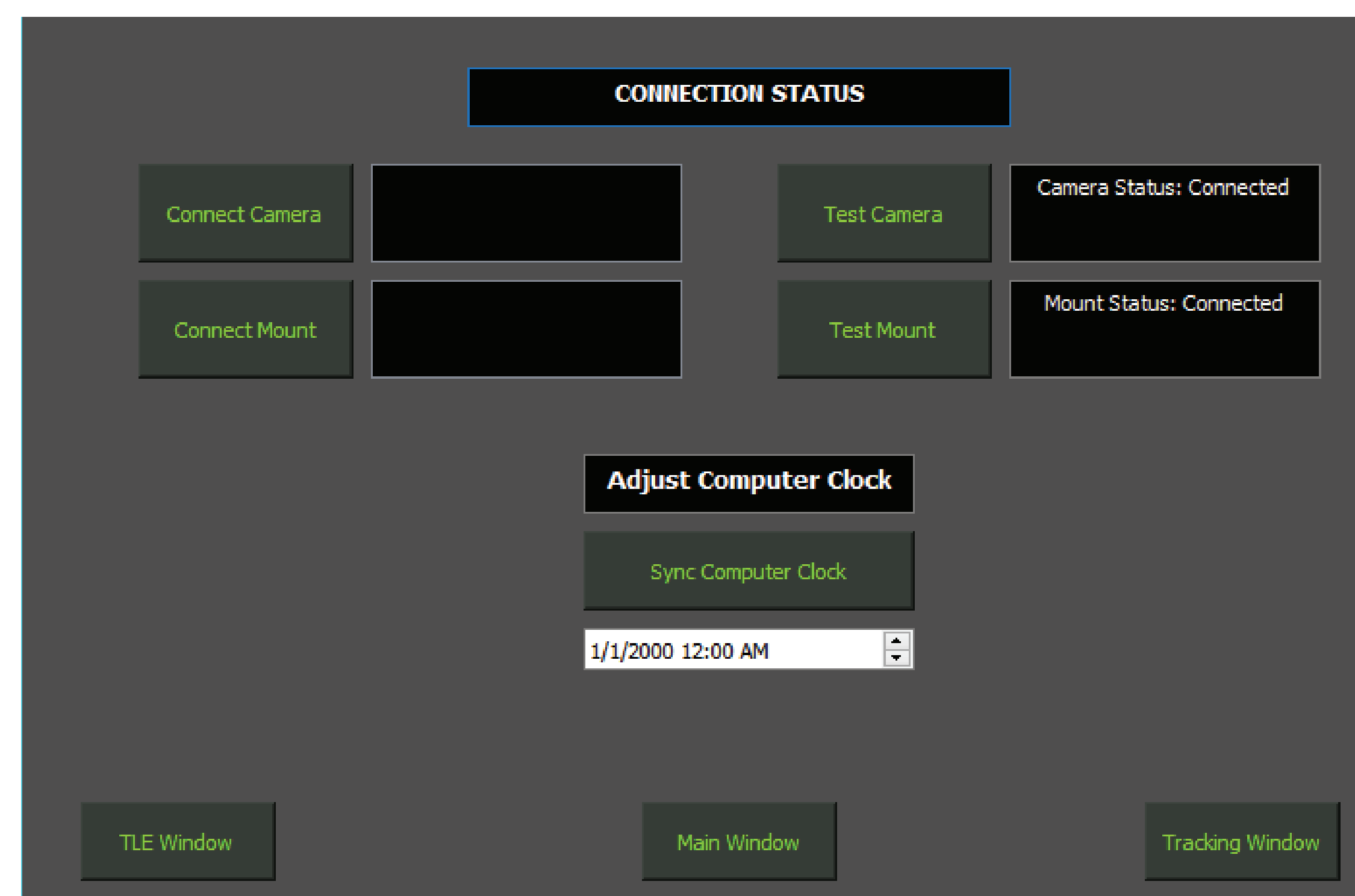
OSCOM utilizes both commercial and custom software to perform satellite tracking and imaging, image processing and photometry. The main custom program used to perform tracking is named Auriga and is written in C++. Auriga propagates a two-line orbital element (TLE) for a specific satellite, performs a spline fit and takes a derivative to yield a velocity function which is used to send rate commands to the telescope mount during tracking. Improvements to Auriga are continuously under progress to make it more user friendly, adaptable and easily maintainable. Current improvements include the addition of a graphical user interface, integration of new mount control algorithms, retrieval of TLE's from Celestrak, and TLE selection options for performing multiple, consecutive observations efficiently. Commercial software such as TheSkyX and Genika Astro are used to communicate with Paramount telescope mounts and cameras, respectively. Another custom software, Optical Satellite Analysis Tools (OSAT), assists with processing image data and performing photometry to produce light curves which can be later analyzed.



OSCOM System Diagram: Shows a processing flow overview of the latest version of the tracking software.



TLE List Window: The TLE window has two windows concerning satellite tracking. The first is populated with a list of satellites passing over the user specified GPS location with a minimum pass elevation angle within the desired observing time interval. The satellites are either downloaded from Celestrak's satellite database using the *Download TLE* button or provided by the user in the form of a text file and loaded, using the *Load TLE* button. The second list contains the user selected satellites to be tracked.



Hardware Connection Window: Includes functionality to connect to the mount and camera, test whether the mount is responding to slew commands, test data acquisition from the camera and sync the computer time to allow for accurate TLE based object tracking.

Graphical User Interface

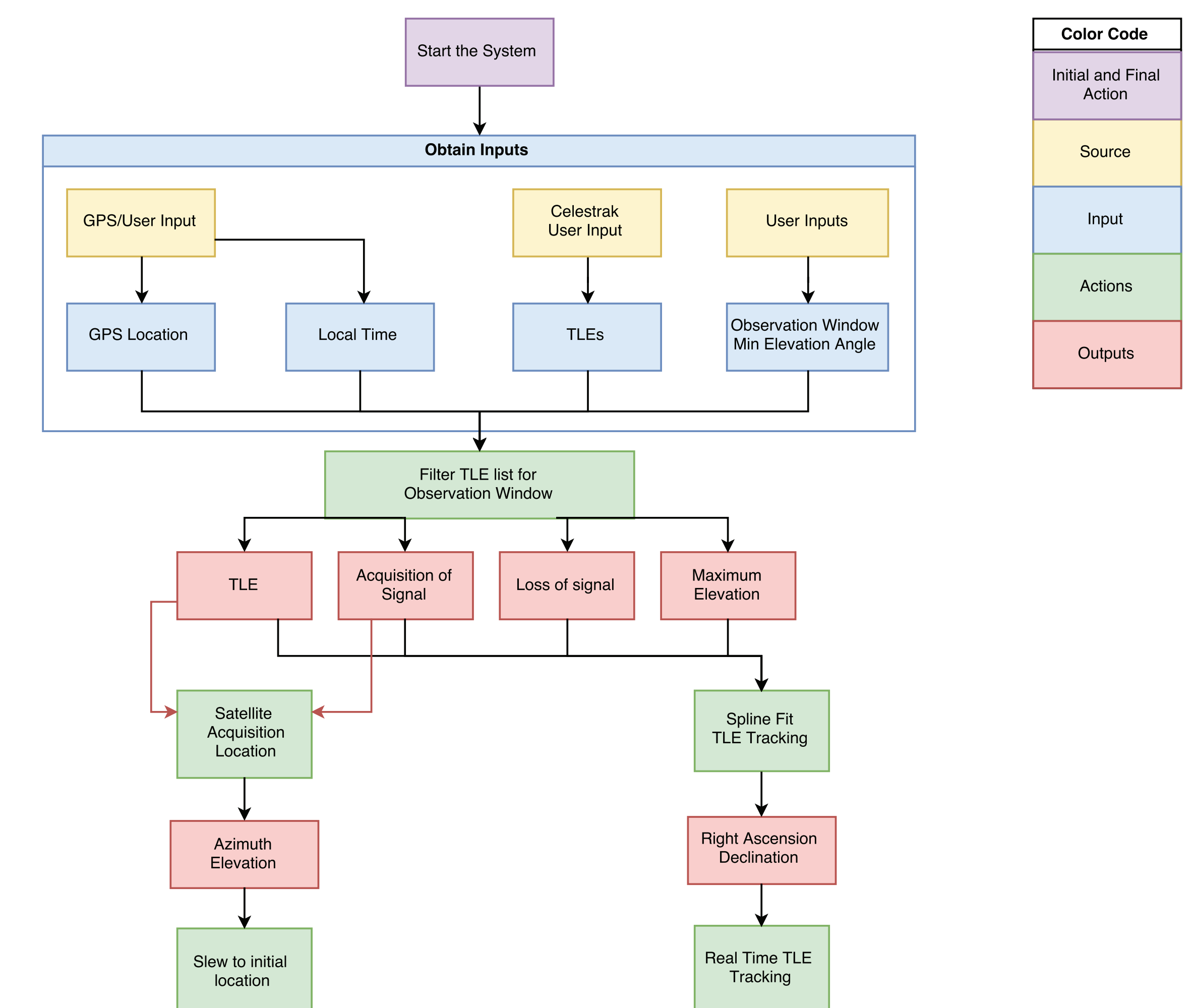
The graphical user interface (GUI) is being designed in C++, utilizing the QT package. Four main windows provide basic GUI functionality, the Start Window, TLE List Window, Hardware Connection Window and Satellite Tracking Window. The Start Window links to all other functional windows. The TLE List Window displays user provided satellites or satellites downloaded from Celestrak's database, which pass over a specified location with a minimum specified elevation angle and in a desired time frame. The Hardware Connection Window allows the user to select a mount and camera from a variety of different types, while the Satellite Tracking Window allows initialization of tracking, track rate adjustments, and a window with a live feed from the camera.

Communication Architecture

The communication architecture, written in C++, requires a modularized structure to allow support from various mount and camera types which have vastly different operating protocols. Windows Network Sockets communicate with the individual mounts, which accept strings of JavaScript as commands. Currently supported mounts are Bisque Paramounts and Losmandy mounts with Gemini II. Genika Astro is used to communicate with the camera. It enables high speed image capture, image capture adjustments, histogram stretch and real-time image display.

Tracking Overview

Current improvements to the tracking software feature obtaining satellite TLEs from a Celestrak database by a user, initiated download. Those orbital elements are then propagated to determine whether a pass occurs over a given location. The list of orbital elements goes through several filters such as an illumination filter, which determines whether a satellite is in Earth's shadow, operational status filter, and mission type filter. The user can select which satellites to track from the simplified, filtered list. Once a satellite is selected, its orbit is propagated and a spline is fit. Using the spline and its derivative, the mount is slewed to its acquisition location and waits for the satellite to enter the field of view. When the time for the satellite to pass occurs, the TLE-based tracking begins. If the satellite TLE is slightly inaccurate, a manual correction can be applied to keep the satellite in the field of view of the image.



Satellite Tracking Flow Diagram: Shows the process by which satellite tracking is performed and the inputs and outputs associated with the current tracking program.

Future Goals

There are several major improvements to the software planned. The software will have support for several different telescope mounts. The user will be able to select the desired mount from within the GUI. Also, interfacing with the CMOS camera will be done so the user can see images on the GUI while tracking. After this feature is implemented, optical feedback will be added to allow for closed-loop tracking of satellites. The new feature will contain two modes. In user-assisted mode, the user will select an area of interest an unresolved satellite is located. The tracking algorithm will then find and center the satellite in the image. In automated mode, an area of interest is found automatically using a streak detection method for finding satellites. This will be accomplished by using a satellite's TLE to determine an approximate angle and length for the streak caused by the satellite as it passes over the acquisition location. Following acquisition, the same method of optical feedback tracking is used as in user-assisted mode.