1. GUI
    a. First Time Usage
        i. The GUI uses the C# .NET 3.0 framework. Installing Microsoft's Visual Studio 2015+ IDE will ensure that the correct version of C# .NET is installed, as well as supply the correct development environment for the GUI. This can be found at https://www.visualstudio.com/downloads/.
        ii. The .exe file will need to be built using Visual Studio; load the solution file and then click Build -> Build Solution. You can then run the GUI directly from Visual Studio via the debug tool.
    b. Subsequent Usage
        i. The GUI can be run by double-clicking the .exe file within the bin\Debug folder.
        ii. Alternatively, the GUI can be run by building the solution file from Visual Studio and running its debugger.
    c. GUI Information
        i. The GUI is a Windows C# form that allows the user to enter in a TLE and GPS coordinates via the drop down file menu.
        ii. Upon entering a valid TLE/GPS data and selecting the "Start Tracking" button, the TLE/GPS data will be forwarded to the software control system where it can be transformed into a path for the mount to follow.
2. Plane Tracking Module (Python -> PlaneTracking) (Uses Python 2.7 on Windows 10)
    a. First Time Usage
        i. Plug in the RTL-SDR and antenna into the computer into a USB that you particularly like
        ii. DO NOT install any drivers that Windows suggests for the RTL
        iii. Run zadig_2.2.exe, select the RTL from the drop-down menu, and install the driver
        iv. Run ntp-4.2.8p8-win32-setup.exe and follow the instructions
        v. Restart device, probably (will allow zadig/RTL installation to take effect as well as NTP installation)
        vi. Open Task Manager, go to Services tab, stop the W32Time service, start the NTP service
    b. Subsequent Usage
        i. Plug in RTL-SDR into the USB port that you originally used
        ii. Verify in script.py (lines 32, 33, 70) that the path is correct (if using absolute path, change to current directory)
        iii. Run script.py
        iv. Enjoy!
    c. RTL and Antenna Information
        i. RTL Name: RTL-SDR R820T2
        ii. Amazon link (provides sufficient documentation): https://www.amazon.com/gp/product/B011HVUEME/ref=oh_aui_detailpage_o00_s00?ie=UTF8&psc=1

   iii. The antenna is a half-wave dipole antenna that was made out of a wire hanger cut into two pieces, bent, and pierced through half of a champagne cork. To replicate the antenna you can simply build something that is nearly visually identical to the antenna provided. The only precise measurement is the end-to-end length of the final antenna, which should be 137mm. The connection to the RTL-SDR unit is an SMA female input. It is convenient to solder the wire ends of the antenna to an SMA male-to-male adapter to make it easier to connect to the RTL unit. The particular adapters used in the project were found here: https://www.amazon.com/gp/product/B00AUB5ODM/ref=oh_aui_detailpage_o07_s00?ie=UTF8&psc=1
   iv. The plane tracking module is important for full system testing and validation. By implementing plane tracking we can cut our testing cycle and test the product more frequently. Further, our customer's satellite is non-functional in orbit so he wants to use our project to track a high-altitude balloon for the time being. These balloons output messages very similar to planes and they move in similar (albeit slower) trajectories. The plane tracking component will be very useful in developing a high-altitude balloon tracking component. The prediction algorithm developed for predicting future plane position proves invaluable for this application as well. The plane tracking module is a completely standalone module and does not rely on any other components. It can be accessed by changing settings in the GUI.

3. GPS Module (Python -> GPS) (Uses Python 2.7 on Windows 10)
   a. First Time Usage
      i. Unzip driver folder and follow driver installation process
      ii. Plug in GPS unit to a USB port
      iii. Right-click Windows icon in the bottom left of the desktop; select Device Manager
      iv. Expand "Ports (COM & LPT)" and identify the COM number of "Prolific USB-to-Serial Comm Port"
      v. Open gps.py and change the COM port number on line 25 to match that of your machine
      vi. Run "pip install pyserial" in a command shell (the GPS module uses the package PySerial)
   b. Subsequent Usage
      i. Copy gps.py into any directory where you have code that requires a call to the GPS module
      ii. In the Python scripts that require GPS info, add "from gps.py import poll"
      iii. Make calls to the GPS as "gps = poll()"
      iv. gps is a list of 4 elements where element 0 is UTC time, element 1 is latitude in degrees, element 2 is longitude in degrees, and element 3 is altitude in meters

c. GPS Receiver Information
    i. Name: US GlobalSat BU-353-S4 USB GPS Receiver
    ii. Website: http://usglobalsat.com/p-688-bu-353-s4.aspx
    iii. The GPS unit is important in calculating look angle for the mount. By automatically and accurately learning the current time and the current position of the mount we can properly point the mount towards our target. The GPS module is a completely standalone module and does not rely on any other components. It can be accessed as described in the "Subsequent Usage" section.

4. Path Predict Module
    a. First Time Usage
        i. Install Python 2.7
        ii. Install Pyephem Python 2.7 package
    b. Subsequent Usage
        i. Input TLE into GUI
        ii. Verify GPS module can receive data
        iii. Verify satellite will pass overhead within the next two hours
        iv. Verify alpha, beta, time, velocity, and acceleration print
    c. Pyephem Package Information
        i. http://rhodesmill.org/pyephem/
        ii. Pathpredict.py is designed to take a TLE inputted from the GUI and call current GPS location and time. From that information it will use the pyephem library to compute the satellites location over an assigned time period by an assigned interval of time. The position and time information will then be used to find velocity and acceleration. The position information will be passed to the motor code to execute the path smoothly. The velocity and acceleration will be used to determine the values to step the motors' velocity and acceleration limits.
5. Motor Control Code
    a. First-time use
        i. Download python 2.7
        ii. Install numpy python package
        iii. Download Phidgets Drivers for your OS from http://www.phidgets.com/docs/Operating_System_Support
        iv. Download Phidgets Python libraries from http://www.phidgets.com/docs/Language_-_Python#Libraries_and_Drivers
        v. Run "python setup.py install"
        vi. At the top of your python script, import the following:

    "from ctypes import *from time import sleep
    from Phidgets.PhidgetException import PhidgetErrorCodes, PhidgetException

from Phidgets.Events.Events import AttachEventArgs, DetachEventArgs, ErrorEventArgs, InputChangeEventArgs, CurrentChangeEventArgs, StepperPositionChangeEventArgs, VelocityChangeEventArgs
from Phidgets.Devices.Stepper import Stepper
from Phidgets.Phidget import PhidgetLogLevel”

   b. Subsequent Use
      i. Import Libraries into Code as in a.

6. Image Processing Module
   a. First-time use
      i. Run .exe file from GUI on Windows 10
   b. Subsequent usage
      i. Run .exe file from GUI on Windows 10
      ii. To edit source code and build a new executable, install OpenCV according to the following instrucitons: https://github.com/MicrocontrollersAndMore/OpenCV_3_Windows_10_Installation_Tutorial
7. MinnowBoard Turbot
   a. First-Time Use (Windows 10 Pro)
      i. Assemble Minnowboard Turbot with Silverjaw Lure and msata drive
      ii. Download bootable Windows 10 media to a USB drive
      iii. Download Turbot Firmware from Intel: https://firmware.intel.com/projects/minnowboard-max
      iv. Plug the firmware USB into the Turbot
      v. Plug Turbot into HDMI-ready display, keyboard and mouse, and power
      vi. Follow the instructions at https://minnowboard.org/tutorials/updating_your_firmware
      vii. Power off, plug Windows 10 bootable media into Turbot, power on
      viii. Follow Windows prompts to install Windows 10 Pro
      ix. Restart Turbot
      x. Create Username and Password or log in to MS Account
   b. Subsequent Use
      i. Power on and log into Turbot
8. Star Field Module
   a. Clone https://github.com/scivision/astrometry_azel
   b. Follow instructions in https://github.com/scivision/astrometry_azel/blob/master/README.rst