Python Telescope Automation User Manual for Modern Eddington Experiment 2024

Kyle Gourlie and Austyn Moon

Faculty Advisor: Heather Hill

Linn-Benton Community College

Oregon NASA Space Grant Consortium

May 19, 2023

Contents

1 Background

In the astronomy community, external software such as *Prism*, *Astrometry*, *Sharpcap*, and *SkyX* is used for astrophotography and data analysis of images. This method results in expensive software costs and the complexity of integrating all operations asynchronously. *ASCOM* Alpaca fixes these issues by being a bridge between the hardware operating systems and uniting their operations under a single private network. The benefit of utilizing the astronomical hardware through *ASCOM Alpaca* is the ability to automate the operation process through the programming language Python. The connection is made through a module called *Alpyca* that communicates to *ASCOM Alpaca* through its A.P.I. (Application Programming Interface). The goal of this project is to fully automate the process of the Arthur Eddington Experiment which was initially performed in the year 1919 to verify Einstein's General Theory of Gravitation. The modernization of the experiment through software and electronics has been done before in years prior, but with each new data set, the result has had higher accuracy. This experiment is unique compared to previous ones due to its free and open-source nature. The automation program will be available to anyone for free and will be used to gather data during the 2024 solar eclipse.

The automation process consists of three parts: slew mount controls, camera controls, and image processing. The telescope mount controls will slew to the desired positions on the day of the eclipse and track the Sun's movement to have direct exposure to the correct location in the sky. The camera controls will control the exposure rate, gain, and number of exposures. All the data will be stored in the form of a FITS file and analyzed after the eclipse. The image processing Python script will stack the FITS file images to decrease the amount of atmospheric disturbance.

i Einstein's Theory of Gravity

Albert Einstein is a famous theoretical physicist from the early 20th century. His main contributions to Physics include his theories of Special and General Relativity. The Theory of Special Relativity discusses the unification of space and time into one entity: space-time. The theory surrounds the observations that nothing in the observable universe can go faster than the speed of light, no matter the frame of reference. For this to be true, time and space must be united and can be contracted and stretched (time dilation and length contraction). Time dilation is the concept that time behaves slower and faster in space, not at the same rate. Length contraction is similar but in regards to measurements of length. The Theory of General Relativity is where Einstein developed his theory on gravitation. In the universe, there are four fundamental forces: strong nuclear, weak nuclear, electromagnetic, and gravitational forces. Current theories have discovered that three out of four of the theories are explained via Quantum Mechanics, but gravity is not. Einstein's Theory of Gravitation states that gravity is a result of the bending of space-time. Einstein also theorized that the curvature of space-time can be experimentally measured by light. This is where Arthur's Eddington Experiment of 1919 comes in.

ii Arthur Eddington's 1919 Experiment

During a total solar eclipse on May 29th, 1919, Arthur Eddington and Frank Watson Dyson observed the apparent deflection, called gravitational lensing, of light from a star passing near the Sun predicted by Einstein's Theory of General Relativity (Dyson et al., 1920). Eddington's idea was that through this experiment he would be able to confirm Einstein's Theory of General Relativity. Since Eddington's 1919 experiment, many 20th century scientists have failed to reproduce this remarkable experiment using an optical telescope (Grant, 2018). In 2017, Don Bruns was able to conduct the Modern Eddington Experiment (MEE) and produce results, but not without multiple external and expensive software packages and fairly complex processing steps. The 1919 Eddington Experiment has been renamed the Modern Eddington Experiment (MEE) because of the use of modern digital cameras, and most recently wide-angle digital cameras (Dittrich, 2022). In addition, very large numbers of stars can be imaged and quickly downloaded onto computers (Dittrich, 2022). The work that Don Bruns has done is the foundation of work that our team at Linn-Benton Community College wishes to start from and synthesize into usable and affordable software for educational use.

It is also worth mentioning that Eddington's Experiment has come under criticism from the modern scientific community (Kennefick, 2009). This criticism and inability to reproduce Eddington's Experiment using an optical telescope has created a strong motivation to better understand the eclipse expeditions of Eddington and to reproduce his results from 1919. While the results of Eddington's Experiment are under scrutiny, Einstein's Generalized Theory of Relativity has been confirmed by hundreds of other experiments.

iii Introduction to the Modern Eddington 2024 Experiment

Ever since the 1919 Eddington eclipse expeditions measured the gravitational deflection of starlight at a total solar eclipse, thereby supporting Einstein's General Relativity, numerous astronomers have attempted to replicate and improve on their results. We propose to conduct a Modern Eddington Experiment (MEE) in April 2024 in order to:

1. Conduct the experiment more accurately than ever done before.
2. Offer the use of four MEE research stations to be used by Portland Community College, Linn-Benton Community College, Oregon State University, and Sunriver Nature Center and Observatory.
3. Create a project website using the acquired domain name of ModernEddingtonExperiment.org.
4. Recruit teams to use the MEE stations in an international educational experience in El Salto (Durango) Mexico.

With the high precision of the equipment for these stations, and proper procedures and training, a very large data set can be obtained driving down the uncertainty error, hopefully below 1%. This accuracy will then approach the accuracy of gravitational lensing experiments performed by the global collaboration of radio telescopes. We will be measuring the Einstein Coefficient, which is the theoretical deflection of a light photon which passes exactly by the limb of the sun. The hope is to perform the Modern Eddington Experiment more accurately than ever before with as many as 12 research teams and as many as 36 students.

Early in the 20th Century, Einstein had predicted a gravitational Einstein Coefficient of 1.7512 arc-seconds which is the deflection of a photon if it passed by the Sun very close to the limb. Following the 1919 eclipse, Dyson and Eddington reported deflection coefficients of 1.98 ± 0.12 arc-sec and 1.61 ± 0.30 arc-sec from their two stations based on just seven stars. Between 1919 and 2017, about ten parties performed the experiment with varying results but collected limited numbers of stars on their glass plates. The experiment now is called “Modern” because with the advent of modern digital cameras, and most recently wide-angle digital cameras, very large numbers of stars can be imaged and quickly downloaded onto computers.

For example, at the solar eclipse of August 2017, twelve parties attempted the MEE and two were successful:

1. Dr. Donald Bruns, using a small refractor and CCD camera at a high-altitude site in Wyoming, found a value of 1.75 $\pm$ 0.05 arc-sec, an uncertainty of 3% from air turbulence. He collected about 400 data points. This was the most accurate performance of the experiment in history. For this extraordinary accomplishment, AAS awarded him the Chambliss Amateur Achievement Award.

2. A Portland Community College team comprised of Professors Rod Lee and Toby Dittrich, with the skilled mentoring of amateur astronomer Richard Berry, four Portland Community College undergraduate students obtained a value of 1.68 $\pm$ 0.8 arc-sec despite severe air turbulence at their low-altitude site in western Oregon due in part to forest fires. The PCC team collected about 800 data points.

Presently, both Don Bruns and Richard Berry are project advisors and mentors for the faculty and students of the MEE 2024 experimental effort. Because of the continued improvement of the CMOS cameras, and the acquisition of ultrafast computers planned for in this grant, during the 2024 eclipse in Mexico our teams could potentially acquire tens of thousands of data points, making this performance of the MEE the most precise execution in history by more than a factor of ten!

2 Installation of Software

The computer program that automates the telescope processes is written in a computer programming language called Python. There are many computer languages such as C, C++, Java, Fortran, etc but they all share the same goal: they are NOT machine code. Machine code is the computer language that the hardware can directly understand. Coding in machine code is very slow and tedious, which is why other languages were developed that use an interpreter to translate whatever computer language into machine code. Not only does the Python program need to communicate with the computer's hardware, but also needs to communicate with the external telescope and camera hardware. The additional astronomy hardware means that not only does the computer need a Python Compiler, an application that can run the code, and an Interpreter, it needs a separate bridge that allows for communication between the computer and the telescope/camera hardware. This is where *ASCOM Alpaca* comes into play.

i Prerequisites

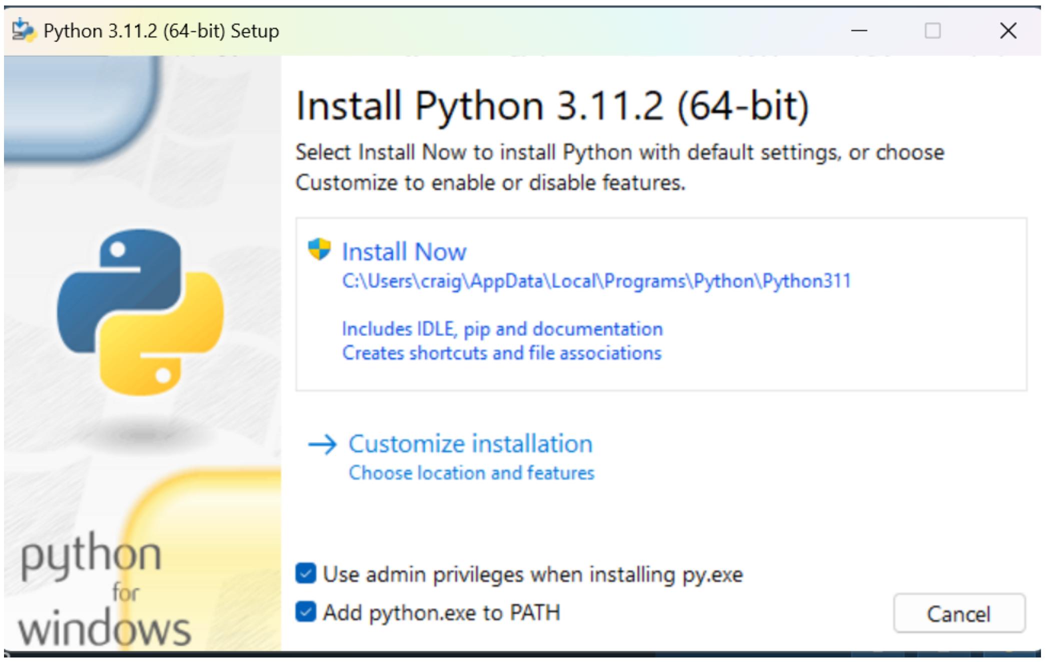
In order to successfully implement this Python program with your Astronomical Hardware, you will need a few things. The most important piece of equipment that you will need is an ASCOM-compatible telescope and camera setup. ASCOM will be fully discussed in a later section, but what it means to you right now is that you will need to look at the specifications of your equipment. A quick internet search of the make and model of your telescope mount and camera should provide enough information to come to a conclusion. For the purposes of this User Manual, the camera in use will be a ZWO ASI 1600M Pro CMOS camera with a Celestron AVX Telescope Mount (#91519). While these models are highly recommended due to confirmation that the equipment is compact-able with the program, similar models made by ZWO and Celestron should also work just as well.

Another software-related requirement is computer storage. Our team recommends 50 Giga-Bytes (GB) of available storage for amateur use of the program. This leaves room for the large sizes of software installations that will be needed, and the large file sizes of your astronomical objects.

All resources and information regarding our Python program can be found here at our GitHub Repository.

ii Python Installation

In order for your computer to be able to read and run Python programs, you will need to install a Python Compiler. A very common one is through Python.org. You can either navigate the website to install the proper version of Python or open the executable file from this shared Google Drive. Once the ZIP file has been downloaded and extracted, there is a file titled "Python IDE.exe." Opening this file will begin the installation of Python onto your computer. The first window prompted will appear as this:


Python 3.11.2 (64-bit) Setup

Install Python 3.11.2 (64-bit)

Select Install Now to install Python with default settings, or choose Customize to enable or disable features.

Install Now
C:\Users\craig\AppData\Local\Programs\Python\Python311

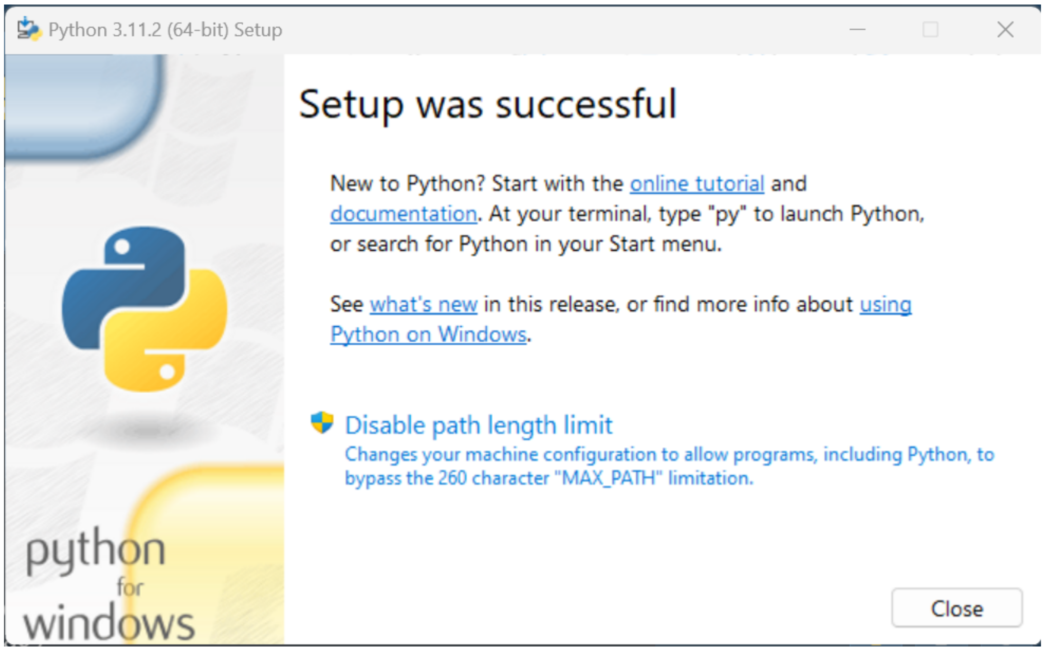
Includes IDLE, pip and documentation
Creates shortcuts and file associations

Customize installation
Choose location and features

☑ Use admin privileges when installing py.exe
☑ Add python.exe to PATH

Note the two check marks selected at the bottom, most importantly "Add python.exe to PATH." It is vital that this option is selected because it allows the use of the PIP to install Python packages. PATH is important because it bridges the connection between the Python Compilers to other resources. This "bridge" is how modules can be downloaded over the computer's terminal and via the internet instead of physically downloading the file and trying to bridge it manually to Python.

It is recommended to select the default installation location, not the custom installation option. Don't select the custom installation option for installation. Select the latter option. After the selection "Install Now," another window will appear:



Setup was successful

New to Python? Start with the online tutorial and documentation. At your terminal, type "py" to launch Python, or search for Python in your Start menu.

See what's new in this release, or find more info about using Python on Windows.

Disable path length limit
Changes your machine configuration to allow programs, including Python, to bypass the 260 character "MAX_PATH" limitation.

At this moment, the Python installation is now complete. If you want to explore the Python IDE, you can search for the application and it will appear.

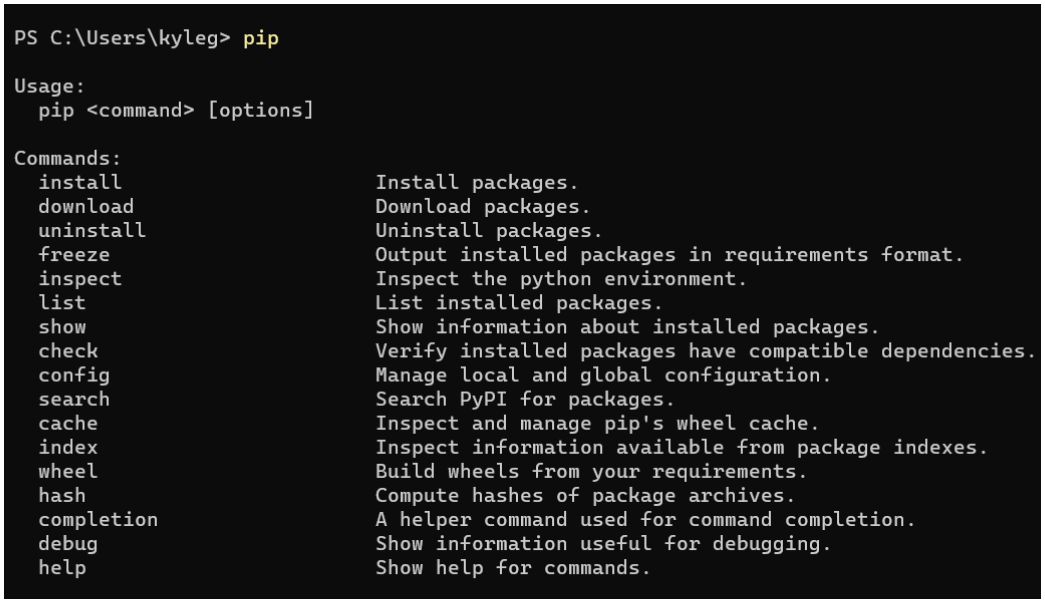
iii PIP Installation

PIP is a Python package manager that allows the installation of Python packages with ease. If the Python IDE installation was performed correctly, PIP should already be on your computer. To check to see if PIP was installed, you can open the computer's terminal and enter the following command:

```
pip
```

followed by pushing `Enter`.

As shown in the figure below, if the installation is correct, many options involving PIP will appear:



```
PS C:\Users\kyleg> pip

Usage:
  pip <command> [options]

Commands:
  install                     Install packages.
  download                    Download packages.
  uninstall                   Uninstall packages.
  freeze                      Output installed packages in requirements format.
  inspect                     Inspect the python environment.
  list                        List installed packages.
  show                        Show information about installed packages.
  check                       Verify installed packages have compatible dependencies.
  config                      Manage local and global configuration.
  search                      Search PyPI for packages.
  cache                       Inspect and manage pip's wheel cache.
  index                       Inspect information available from package indexes.
  wheel                       Build wheels from your requirements.
  hash                        Compute hashes of package archives.
  completion                  A helper command used for command completion.
  debug                       Show information useful for debugging.
  help                        Show help for commands.
```

If any issues occur at this portion of the installation, refer to **??**.

Now that PIP has been correctly installed, it is time to install the individual Python packages that will be needed for the telescope and camera automation program. To install a particular PIP package, open the computer's terminal and type:

```
pip install
```

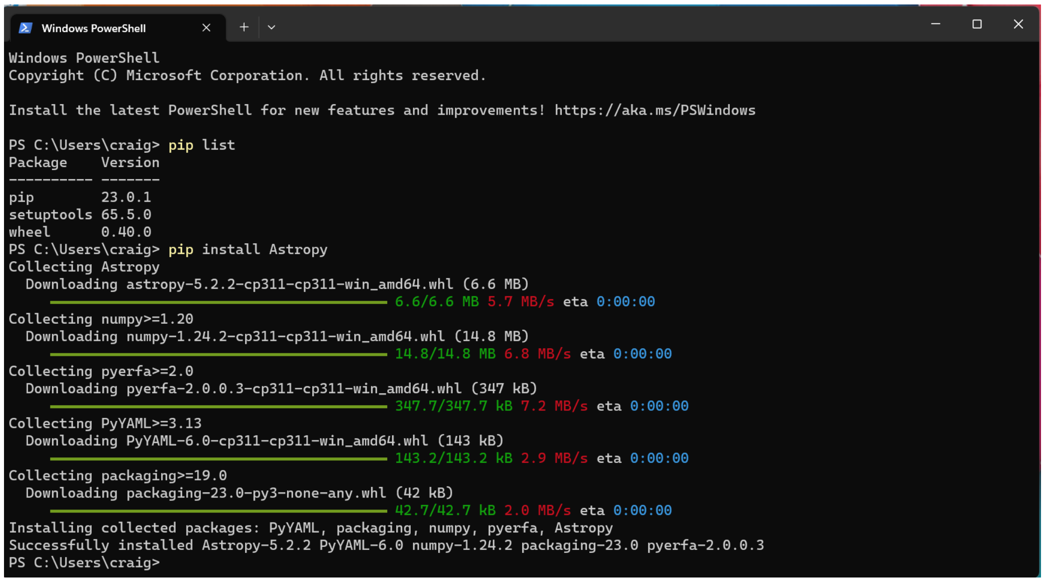
followed by the name of the package. For example, to install astropy, type:

```
pip install astropy
```

The package names are case sensitive. The user will want to install packages in this order:

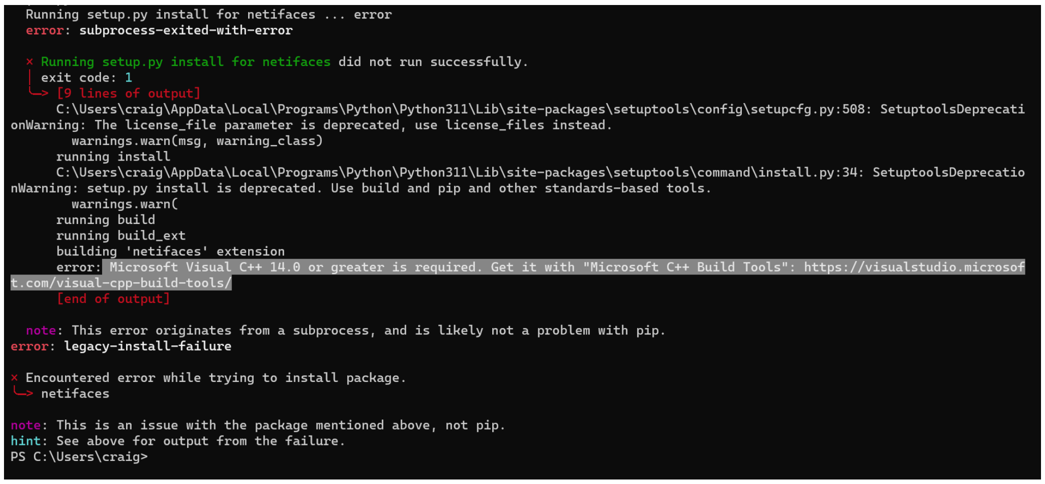
1. astropy
2. matplotlib
3. PySide6
4. alpyca

Here is an example of implementing it with `astropy`:



```
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Install the latest PowerShell for new features and improvements! https://aka.ms/PSWindows

PS C:\Users\craig> pip list
Package    Version
---------- -------
pip        23.0.1
setuptools 65.5.0
wheel      0.40.0
PS C:\Users\craig> pip install Astropy
Collecting Astropy
  Downloading astropy-5.2.2-cp311-cp311-win_amd64.whl (6.6 MB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 6.6/6.6 MB 5.7 MB/s eta 0:00:00
Collecting numpy>=1.20
  Downloading numpy-1.24.2-cp311-cp311-win_amd64.whl (14.8 MB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 14.8/14.8 MB 6.8 MB/s eta 0:00:00
Collecting pyerfa>=2.0
  Downloading pyerfa-2.0.0.3-cp311-cp311-win_amd64.whl (347 kB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 347.7/347.7 kB 7.2 MB/s eta 0:00:00
Collecting PyYAML>=3.13
  Downloading PyYAML-6.0-cp311-cp311-win_amd64.whl (143 kB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 143.2/143.2 kB 2.9 MB/s eta 0:00:00
Collecting packaging>=19.0
  Downloading packaging-23.0-py3-none-any.whl (42 kB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 42.7/42.7 kB 2.0 MB/s eta 0:00:00
Installing collected packages: PyYAML, packaging, numpy, pyerfa, Astropy
Successfully installed Astropy-5.2.2 PyYAML-6.0 numpy-1.24.2 packaging-23.0 pyerfa-2.0.0.3
PS C:\Users\craig>
```

When you get to the last PIP installation, 4. `alpyca`, expect an error to occur. In the terminal, an error message will occur saying something along the lines of: Microsoft Visual C++ 14.0 or greater is required. Get it with "Microsoft C++ Build Tools." as shown below.



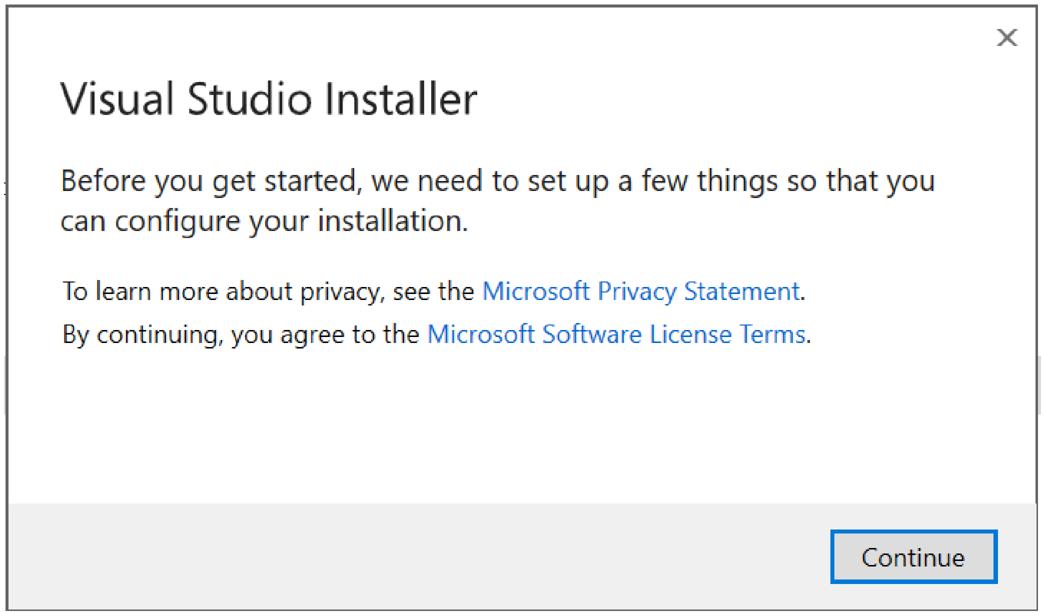
```
  Running setup.py install for netifaces ... error
  error: subprocess-exited-with-error

  × Running setup.py install for netifaces did not run successfully.
  │ exit code: 1
  ╰─> [9 lines of output]
      C:\Users\craig\AppData\Local\Programs\Python\Python311\Lib\site-packages\setuptools\config\setupcfg.py:508: SetuptoolsDeprecati
onWarning: The license_file parameter is deprecated, use license_files instead.
        warnings.warn(msg, warning_class)
      running install
      C:\Users\craig\AppData\Local\Programs\Python\Python311\Lib\site-packages\setuptools\command\install.py:34: SetuptoolsDeprecatio
nWarning: setup.py install is deprecated. Use build and pip and other standards-based tools.
        warnings.warn(
      running build
      running build_ext
      building 'netifaces' extension
      error: Microsoft Visual C++ 14.0 or greater is required. Get it with "Microsoft C++ Build Tools": https://visualstudio.microsof
t.com/visual-cpp-build-tools/
      [end of output]

  note: This error originates from a subprocess, and is likely not a problem with pip.
error: legacy-install-failure

× Encountered error while trying to install package.
╰─> netifaces

note: This is an issue with the package mentioned above, not pip.
hint: See above for output from the failure.
PS C:\Users\craig>
```

In order to solve this problem, we will have to go through the Visual Studio Setup Installation.

iv Visual Studio Setup Installation

In order to install `alpyca`, you will need to install Microsoft Visual Studio Setup. If you have heard of Microsoft Visual Studio Code (VS Code), Microsoft Visual Studio Setup is something different. VS Code is the application you use to write code, while VS Setup is the application used to install the Python compiler so you can run the code.

To install VS setup, you can find the executable file inside the Installation Package under the name of "VisualStudioSetup.exe". Once you open the `.exe` file, a window will appear stating that a few additional things will need to be installed:


Visual Studio Installer

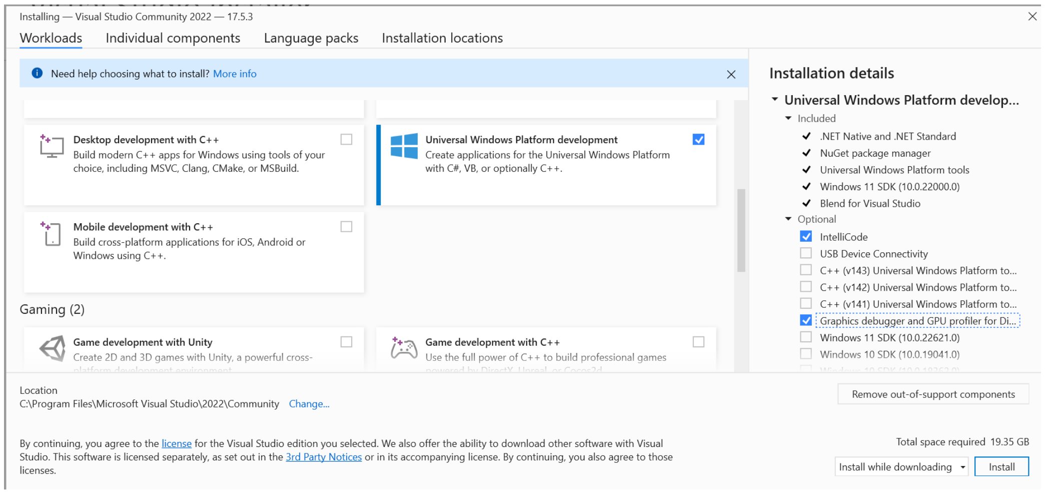
Before you get started, we need to set up a few things so that you can configure your installation.

To learn more about privacy, see the Microsoft Privacy Statement.

By continuing, you agree to the Microsoft Software License Terms.

Once you select "Continue", the rest of the application installation will be completed.

Once this is done, another window will be prompted asking you to select the packages that you wish to you. As shown below, the package that needs to be installed is titled "Universal Windows Platform Development." It is not visible as the first option, so you will need to scroll down within the window until it appears. Once selected, make sure that "Intellicode" and the "Graphics Debugger" boxes are selected as shown below.



Installing — Visual Studio Community 2022 — 17.5.3

Workloads | Individual components | Language packs | Installation locations

Need help choosing what to install? More info

Desktop development with C++ ☐
Build modern C++ apps for Windows using tools of your choice, including MSVC, Clang, CMake, or MSBuild.

Universal Windows Platform development ✓
Create applications for the Universal Windows Platform with C#, VB, or optionally C++.

Mobile development with C++ ☐
Build cross-platform applications for iOS, Android or Windows using C++.

Gaming (2)

Game development with Unity ☐
Create 2D and 3D games with Unity, a powerful cross-

Game development with C++ ☐
Use the full power of C++ to build professional games

Installation details

Universal Windows Platform develop...

Included
- ✓ .NET Native and .NET Standard
- ✓ NuGet package manager
- ✓ Universal Windows Platform tools
- ✓ Windows 11 SDK (10.0.22000.0)
- ✓ Blend for Visual Studio

Optional
- ✓ IntelliCode
- ☐ USB Device Connectivity
- ☐ C++ (v143) Universal Windows Platform to...
- ☐ C++ (v142) Universal Windows Platform to...
- ☐ C++ (v141) Universal Windows Platform to...
- ✓ Graphics debugger and GPU profiler for Di...
- ☐ Windows 11 SDK (10.0.22621.0)
- ☐ Windows 10 SDK (10.0.19041.0)

Location
C:\Program Files\Microsoft Visual Studio\2022\Community Change...

By continuing, you agree to the license for the Visual Studio edition you selected. We also offer the ability to download other software with Visual Studio. This software is licensed separately, as set out in the 3rd Party Notices or in its accompanying license. By continuing, you also agree to those licenses.

Remove out-of-support components

Total space required 19.35 GB

Install while downloading | Install

Notice the large space required of 19.35 GB. This is why it is suggested to have at least 50 GB of space available. Depending on the speed of your computer, this installation can take a while (∼15 minutes).

After this, the application will ask you to **Sign in** to Visual Studio. You can skip this step. Finally, it will ask you to start Visual Studio. You can open the app, but there is no need to do anything with it, so you can just exit out of it once it opens.

Return to PIP Installation and complete the installation of `alpyca`. If the PIP installation of `alpyca` still isn't working, restart your computer and try again. Sometimes your computer will need a restart when massive amounts of files and software are downloaded so the new information can be recognized.

v ASCOM Installation

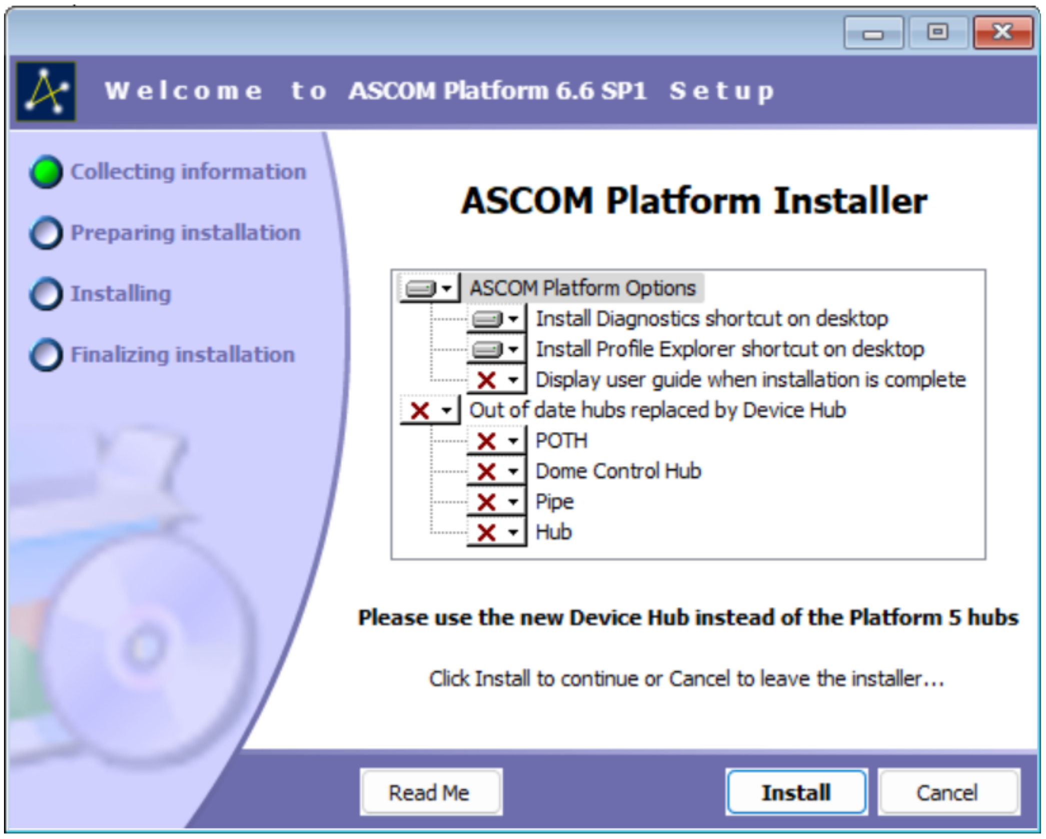
In the past, experiments like this one required software for each piece of hardware. This was due to each piece of equipment having a different operating system. The use of ASCOM bypasses the need for external software due to its driver compatibility. All equipment is operated under the ASCOM Remote Server application. While the primary operation will be under this application, drivers are needed for proper communication between the hardware and the computer. Each driver is unique to the piece of hardware, so it is imperative that the correct driver is installed. In the installation package folder, there are examples of the drivers that one will need to install on their home PC. The computer will now be referred to as "home PC" due to the ability to execute the code from any device connected to a private network that will be operating ASCOM Alpaca. There needs to be a main computer that will have the drivers installed and have the hardware plugged into the COM ports (USB ports on the computer) and "home PC" will be designated as this device. The installation package folder contains the necessary drivers for a Celestron telescope mount and a ZWO ASI CMOS camera.

The ASCOM executable files are located in the Installation Package:

1. `AscomPlatform661.3673.exe`
2. `ASCOMRemote.6.6.8048.20286.setup.exe`

Let's start with the Ascom Platform Installer.

When your computer asks to make changes to your device, the application is named "ASCOM Platform 6.6 SP1." The difference in name makes doesn't affect anything. When the file is opened, your screen will display this:



Welcome to ASCOM Platform 6.6 SP1 Setup

ASCOM Platform Installer

- ASCOM Platform Options
  - Install Diagnostics shortcut on desktop
  - Install Profile Explorer shortcut on desktop
  - ✗ Display user guide when installation is complete
- ✗ Out of date hubs replaced by Device Hub
  - ✗ POTH
  - ✗ Dome Control Hub
  - ✗ Pipe
  - ✗ Hub

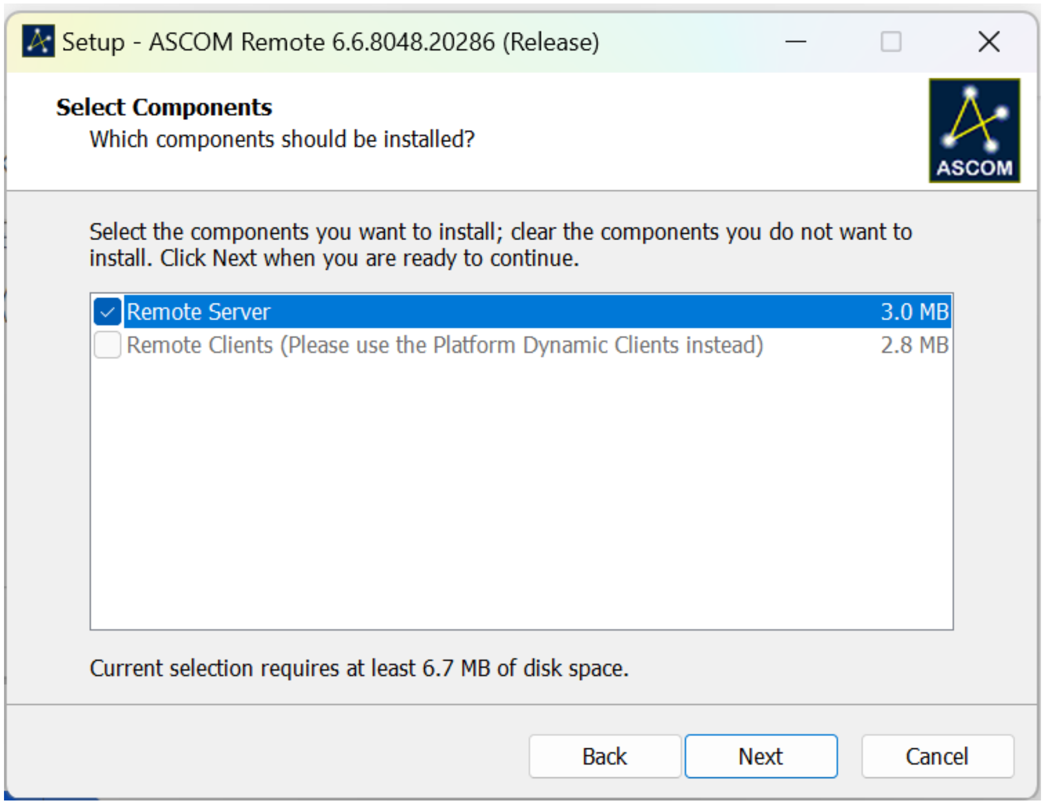
Please use the new Device Hub instead of the Platform 5 hubs

Click Install to continue or Cancel to leave the installer...

You do not need to mess with any of the settings, just click "Install". After this, two applications will be installed: "ASCOM Diagnostics" and "ProfileExplorer".

The next installation will be the ASCOM Remote Installer.

After opening the file, the application will ask you which components you want to install. Make sure to select the box labeled “Remote Server” as shown below.



Setup - ASCOM Remote 6.6.8048.20286 (Release)

Select Components

Which components should be installed?

Select the components you want to install; clear the components you do not want to install. Click Next when you are ready to continue.

✓	Remote Server	3.0 MB
	Remote Clients (Please use the Platform Dynamic Clients instead)	2.8 MB

Current selection requires at least 6.7 MB of disk space.

The rest of the installation steps are just selecting “Next” and “Install”. If all the downloads were performed correctly, there will be four ASCOM apps on your computer:

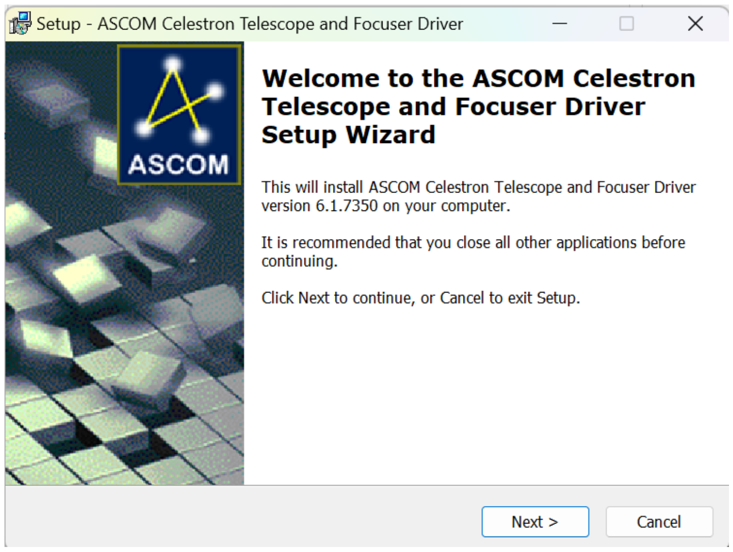
1. ASCOM Diagnostics
2. ASCOM ProfileExplorer
3. ASCOM Device Hub
4. ASCOM Remote Server

The ASCOM Diagnostics app is used in modifying any of the ASCOM software and for troubleshooting. The ASCOM ProfileExplorer is the application that would be used if not using the ASCOM Alpaca method for the connection of hardware. The ASCOM Device Hub is the User Interface of ASCOM. Last, the ASCOM Remote Server will be used to create a private server for the telescope hardware to communicate with each other and communicate with ASCOM and other devices connected to the server.

vi Driver Installations

The purpose of a computer driver is to allow your computer's COM ports to be used for third-party applications. A COM port is the same as the USB ports on your computer that you can plug in a flash drive or a wired computer mouse into. Now, these types of media don't require the additional step of installing a specific driver because your computer comes stock with various drivers that allow you to use them. Other equipment such as a ZWO ASI CMOS Camera and a Celestron Telescope Mount require additional driver installations.

The first one that will be installed is the Celestron Telescope Mount driver. Inside the installation package folder, the driver is labeled as `Celestron Telescope and Focuser Setup (6.1.7350).exe`. If you selected the correct driver, a window will pop up looking like this:


Setup - ASCOM Celestron Telescope and Focuser Driver

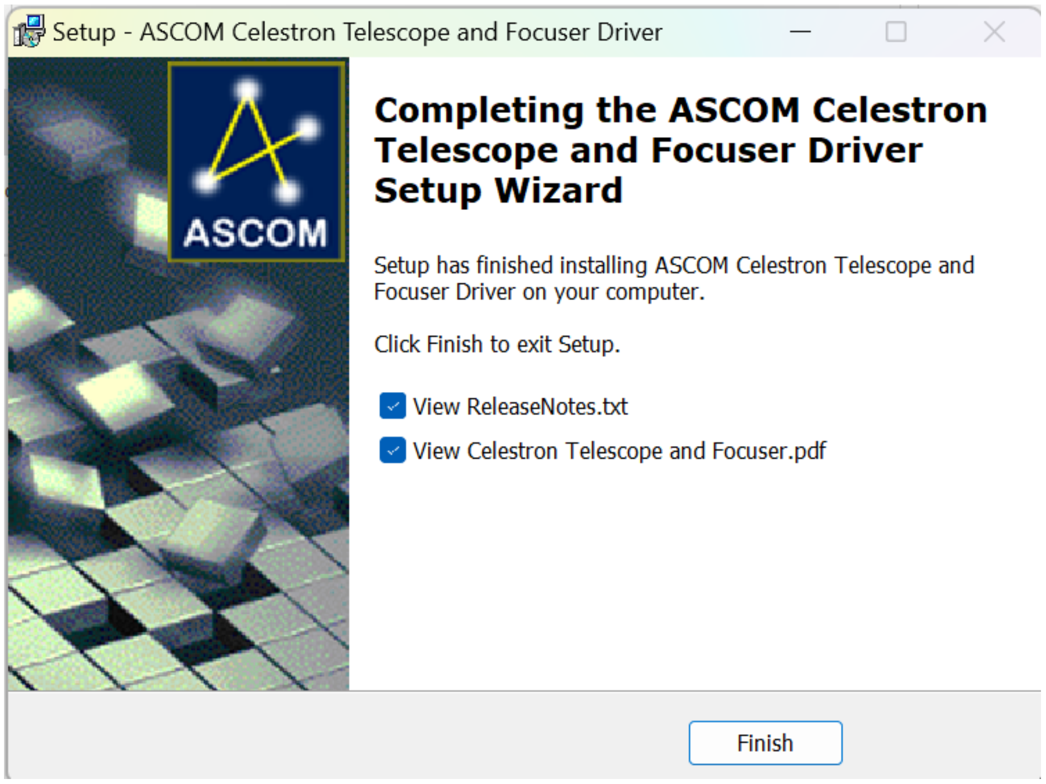
Welcome to the ASCOM Celestron Telescope and Focuser Driver Setup Wizard

This will install ASCOM Celestron Telescope and Focuser Driver version 6.1.7350 on your computer.

It is recommended that you close all other applications before continuing.

Click Next to continue, or Cancel to exit Setup.

After selecting “Next” a new window will emerge allowing for the option to have a release note text file and a driver user manual pdf to open as shown below. These choices are entirely optional and won’t affect our installation.


Setup - ASCOM Celestron Telescope and Focuser Driver

Completing the ASCOM Celestron Telescope and Focuser Driver Setup Wizard

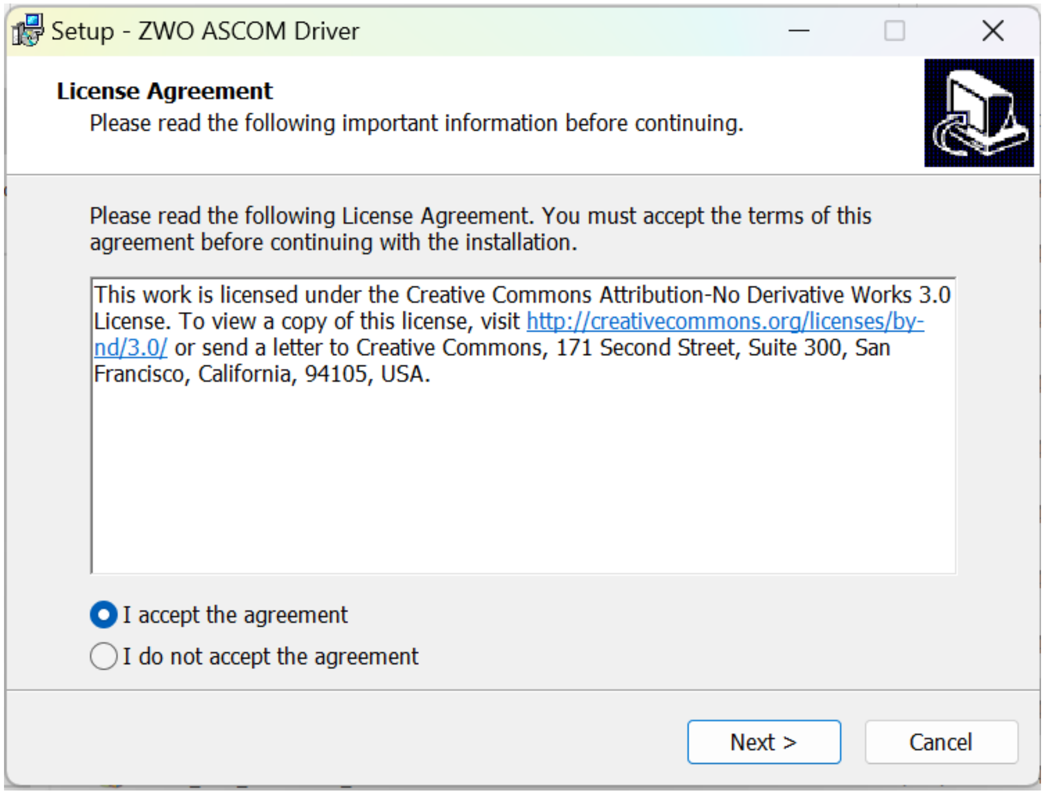
Setup has finished installing ASCOM Celestron Telescope and Focuser Driver on your computer.

Click Finish to exit Setup.

✓ View ReleaseNotes.txt

✓ View Celestron Telescope and Focuser.pdf

The last installation will be the ZWO ASI camera driver. In the installation package folder, it is called `ZWO_ASCOM_Setup_V6.5.11.exe`. What makes the installation of this driver slightly different is that you will need to agree to a User Agreement as shown below.



License Agreement

Please read the following important information before continuing.

Please read the following License Agreement. You must accept the terms of this agreement before continuing with the installation.

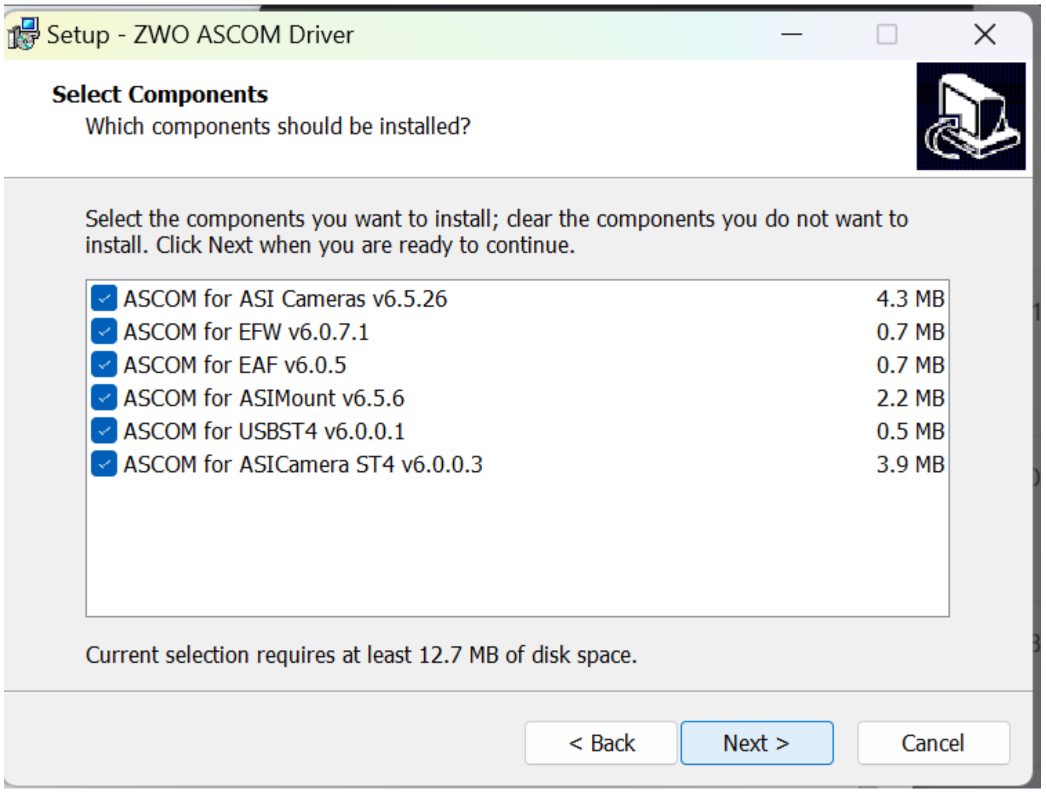
This work is licensed under the Creative Commons Attribution-No Derivative Works 3.0 License. To view a copy of this license, visit http://creativecommons.org/licenses/by-nd/3.0/ or send a letter to Creative Commons, 171 Second Street, Suite 300, San Francisco, California, 94105, USA.

[✓ I accept the agreement]

I do not accept the agreement

In order to use the driver, you must accept the License Agreement.

Once the User Agreement has been accepted, the ZWO ASCOM Driver will prompt you to choose what components you want to download. Select all the options as shown below.



Setup - ZWO ASCOM Driver

Select Components

Which components should be installed?

Select the components you want to install; clear the components you do not want to install. Click Next when you are ready to continue.

✓	ASCOM for ASI Cameras v6.5.26	4.3 MB
✓	ASCOM for EFW v6.0.7.1	0.7 MB
✓	ASCOM for EAF v6.0.5	0.7 MB
✓	ASCOM for ASIMount v6.5.6	2.2 MB
✓	ASCOM for USBST4 v6.0.0.1	0.5 MB
✓	ASCOM for ASICamera ST4 v6.0.0.3	3.9 MB

Current selection requires at least 12.7 MB of disk space.

Next, an installation window will display stating all the things you have selected. The clicking of "Install" will finish the process and all the installations required for the use of the Python program will be complete. Note that this driver installation will also download an application called "ASIMount".

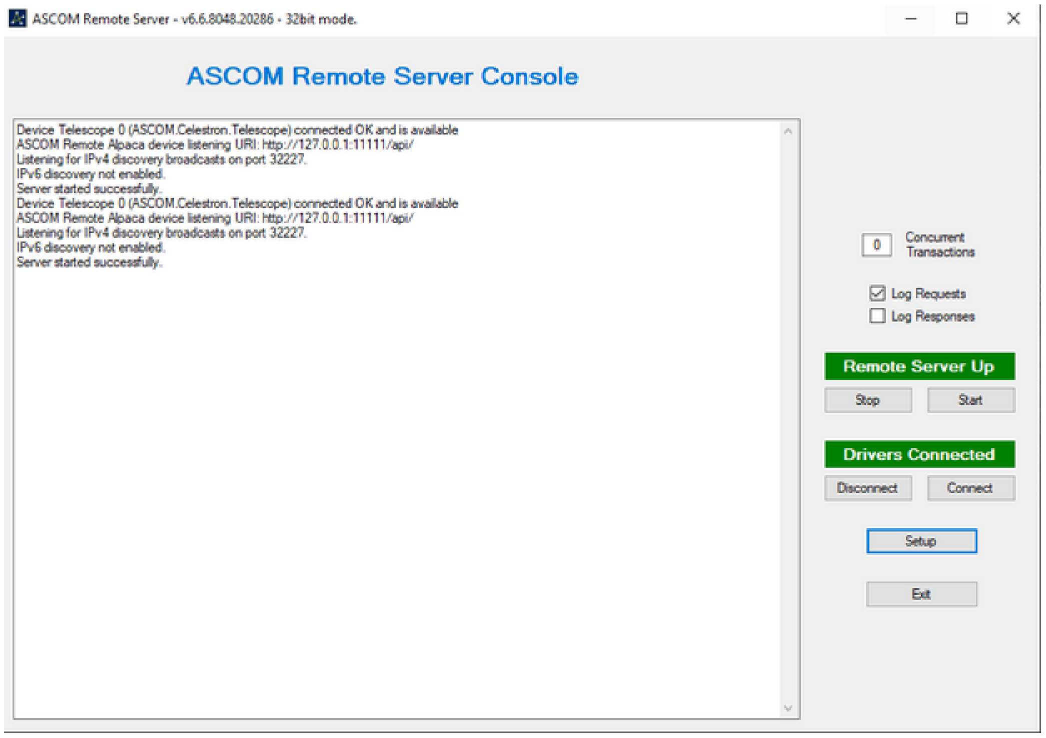
vii Checklist of Software Installations

There were many steps that went into this installation process, so here is a checklist of all the Software Installations that were completed. Make sure that everything listed here was installed.

- ☐ Python IDE
- ☐ VisualStudioSetup
- ☐ PIP
- ☐ PIP Packages
- ☐ ASCOM Diagnostics
- ☐ ASCOM ProfileExplorer
- ☐ ASCOM Device Hub
- ☐ ASCOM Remote Server
- ☐ Celestron Telescope ASCOM Driver
- ☐ ZWO CMOS ASCOM Camera Driver

3 ASCOM Alpaca Server Operations

The only ASCOM application that you will be operating is the ASCOM Remote Server. This application won't be found as a Shortcut on your desktop, so you will have to search for it using the keywords "ASCOM Server". Before opening the app, make sure that all of your astronomical equipment, such as the ZWO ASI camera and Celestron Telescope Mount, are plugged into the computer. The app will look like this:



```
Device Telescope 0 (ASCOM.Celestron.Telescope) connected OK and is available
ASCOM Remote Alpaca device listening URI: http://127.0.0.1:11111/api/
Listening for IPv4 discovery broadcasts on port 32227.
IPv6 discovery not enabled.
Server started successfully.
Device Telescope 0 (ASCOM.Celestron.Telescope) connected OK and is available
ASCOM Remote Alpaca device listening URI: http://127.0.0.1:11111/api/
Listening for IPv4 discovery broadcasts on port 32227.
IPv6 discovery not enabled.
Server started successfully.
```

Directly below the title will be a box of white space that is possibly populated with text. This is the ASCOM Alpaca API terminal. Once in operation, commands will be prompted there and text will be displayed. The text will be a series of `GET` and `JSON` functions. These commands are helpful in understanding possible errors that occur during operation. To better understand these commands, you can read the user manual named, "ASCOM Alpaca API Reference v1" which is inside the "Important Documentation" folder within the Installation Package folder. `JSON` stands for JavaScript

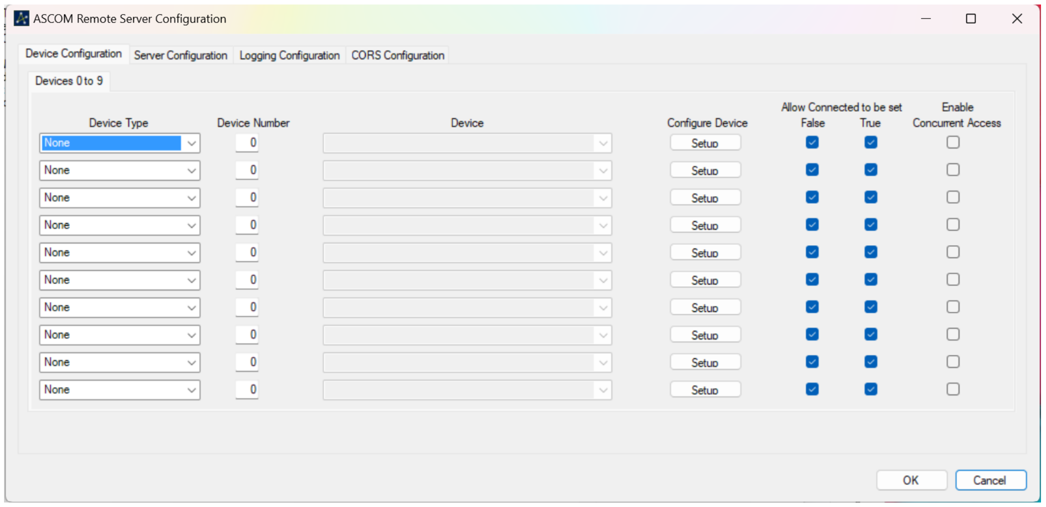
Object Notation.

To better understand how the private server works, try to think about the `GET` and `JSON` functions as input and output commands. For example, if you went out to a restaurant to order food, you have to tell the waiter or waitress what you want to eat. This information on what you want to eat will be transmitted from the waiter or waitress to the cook. This is similar to how the GET function works. The Python program is asking the server to tell the hardware to do something. The JSON is like the response from the cook. Hopefully, the meal that you ordered. From this, you can see that any transmission of data comes in pairs: a GET functions and a JSON function. Now, input commands can be more than just GET functions, such as PUT. The API reference goes more into detail if you are interested.

On the right side of the window are the “Remote Server Up” and the “Drivers Connected” sections The first one turns the server On and Off while the second section will connect or disconnect the drivers. This option can join or sever the connection of the astronomy hardware. Directly below the “Drivers Connected” section are two buttons titled “Setup” and “Exit”. The “Setup” button will prompt you to a Device Configuration tab.

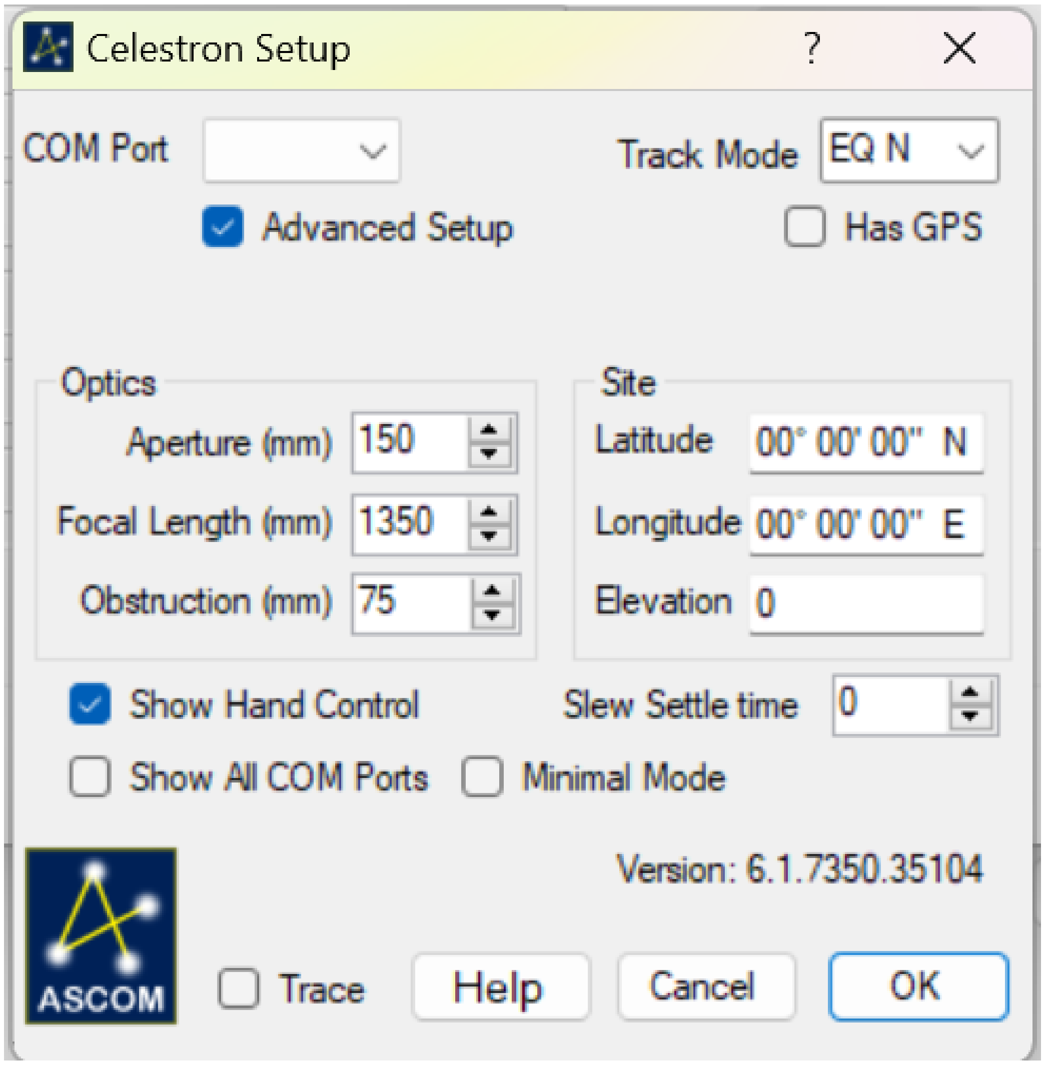
i Device Setup and Connection

This tab of the ASCOM server should appear as this:



Device Type	Device Number	Device	Configure Device	Allow Connected to be set False	True	Enable Concurrent Access
None	0		Setup	✓	✓	☐
None	0		Setup	✓	✓	☐
None	0		Setup	✓	✓	☐
None	0		Setup	✓	✓	☐
None	0		Setup	✓	✓	☐
None	0		Setup	✓	✓	☐
None	0		Setup	✓	✓	☐
None	0		Setup	✓	✓	☐
None	0		Setup	✓	✓	☐
None	0		Setup	✓	✓	☐

One of the wonderful things with the use of ACOM's server is the many options for the types of astronomical devices and the number of devices that can be connected. Under the "Device Type" section, you can select the type of device that will be connected to your computer. Lets make the first device be the Celestron Telescope Mount. This will fall under the Telescope device type. At this point, make sure that your camera and telescopes are properly connected. If they are not, your computer won't recognize them and won't be able to connect them to the server. Under the "Devices" part, you can now select how your device will be connected to the server. If all the drivers were installed correctly, there will be no issues here. The Device that needs to be selected will be "Celestron Telescope Driver." The last step will be to configure the Device. This section will have a lot of information displayed and will appear as this:


Celestron Setup

COM Port

Track Mode EQ N

[✓] Advanced Setup

[ ] Has GPS

Optics

Aperture (mm) 150

Focal Length (mm) 1350

Obstruction (mm) 75

Site

Latitude 00° 00' 00" N

Longitude 00° 00' 00" E

Elevation 0

[✓] Show Hand Control

Slew Settle time 0

[ ] Show All COM Ports

[ ] Minimal Mode

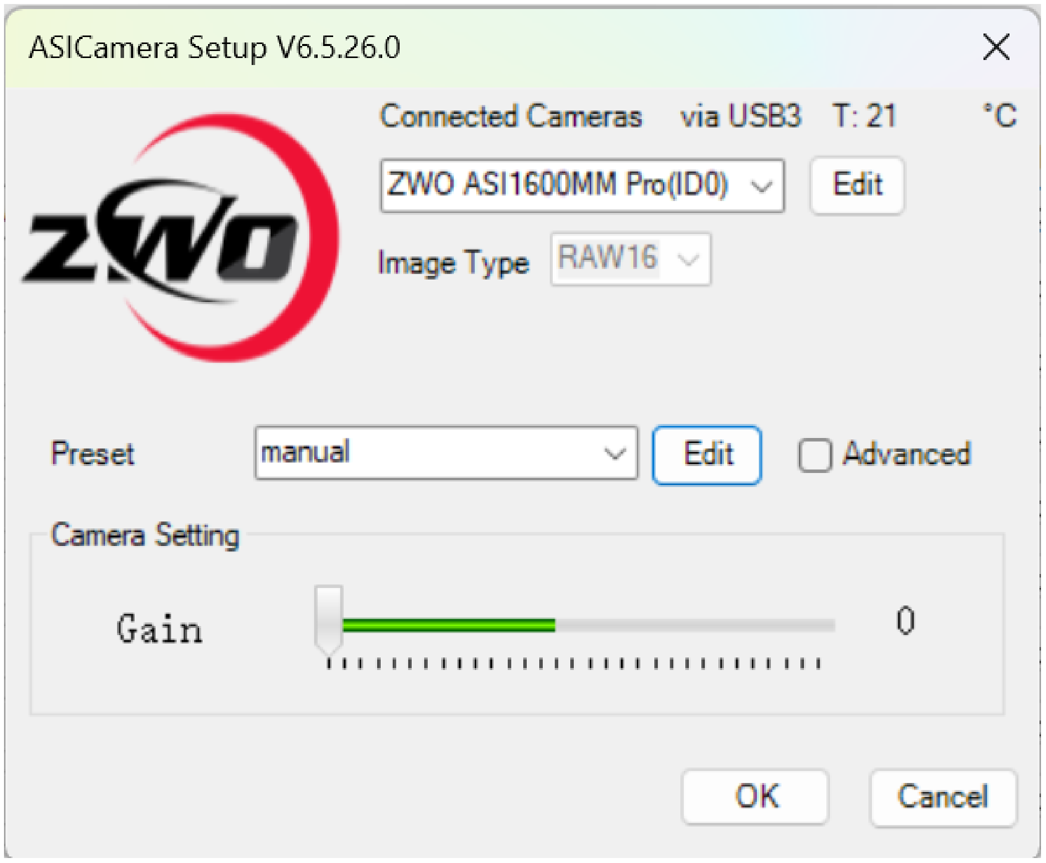
Version: 6.1.7350.35104

[ ] Trace

The COM Port option is the location of where the Telescope Mount is plugged into the computer's USB port. If no COM port options are available, this means that your computer isn't recognizing your telescope mount device. Check to make sure that a proper connection is made (no loose wires) and try again. If the issue persists, restart your computer. If the issue still develops, look at the troubleshooting section.

The Track Mode portion is the coordinate system that the telescope mount will track with. There is no need to change it from Equatorial North, but there are other options such as Equatorial South, and Altitude/Azimuth. The "Optics" and "Site" options won't apear unless you select "AdvaappearSetup." The information regarding the optics of the physical telescope that you are using won't affect the movement of the mount, but the location in which you are located is important for automatic tracking purposes.

The process is exactly the same to connect your ZWO ASI Camera to the server, except you need to choose Camera for your Device Type and ASI Camera(1) for the Device. The device configuration will appear as this:



ASICamera Setup V6.5.26.0

Connected Cameras via USB3 T: 21 °C

ZWO ASI1600MM Pro(ID0)

Image Type RAW16

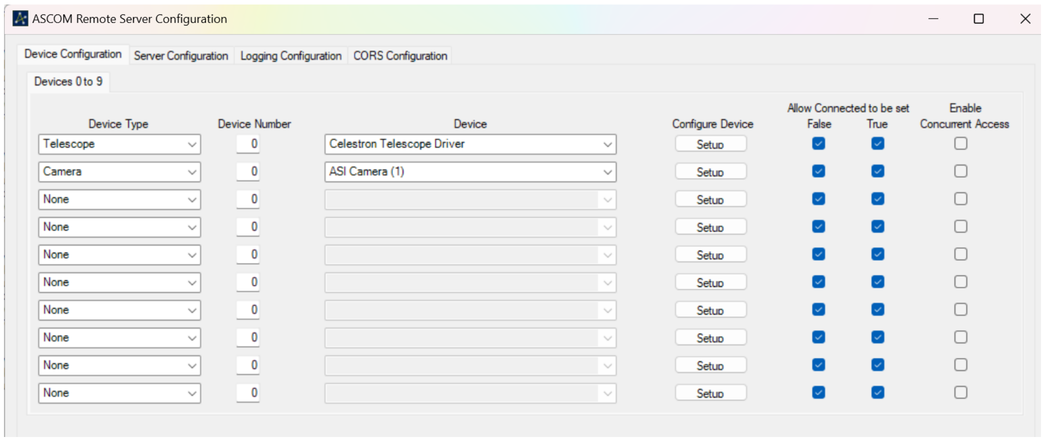
Preset manual

Advanced

Camera Setting

Gain 0

The Device Configuration tab should look like this in the end:


ASCOM Remote Server Configuration

Device Configuration | Server Configuration | Logging Configuration | CORS Configuration

Devices 0 to 9

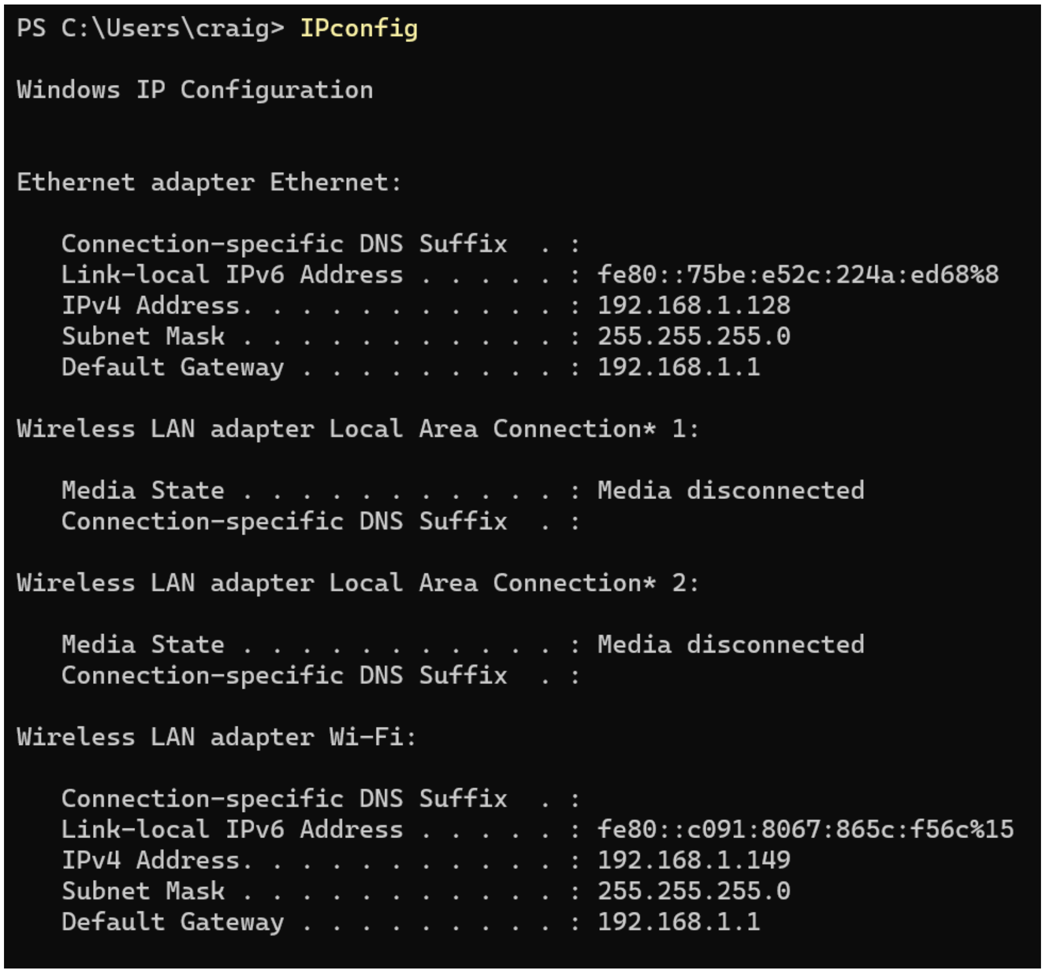
Device Type	Device Number	Device	Configure Device	Allow Connected to be set False	Allow Connected to be set True	Enable Concurrent Access
Telescope	0	Celestron Telescope Driver	Setup	✓	✓	☐
Camera	0	ASI Camera (1)	Setup	✓	✓	☐
None	0		Setup	✓	✓	☐
None	0		Setup	✓	✓	☐
None	0		Setup	✓	✓	☐
None	0		Setup	✓	✓	☐
None	0		Setup	✓	✓	☐
None	0		Setup	✓	✓	☐
None	0		Setup	✓	✓	☐
None	0		Setup	✓	✓	☐

ii Server Configuration

The first part of the creation of the server is complete. The second part is linking the server to a legitimate network. A wireless network is NOT the same as the internet. An internet connection can be wireless (WiFi) or wired (Ethernet connection). A very common source of a wireless connection is a WiFi Router. You can use a WiFi Router with an internet connection, but it will be safer to use a wireless connection that is not connected to the internet. This is what is considered a private network. Only people directly connected to the Router can access the network. "Network" encompasses both the wired and wireless aspect of it.

Server via Network

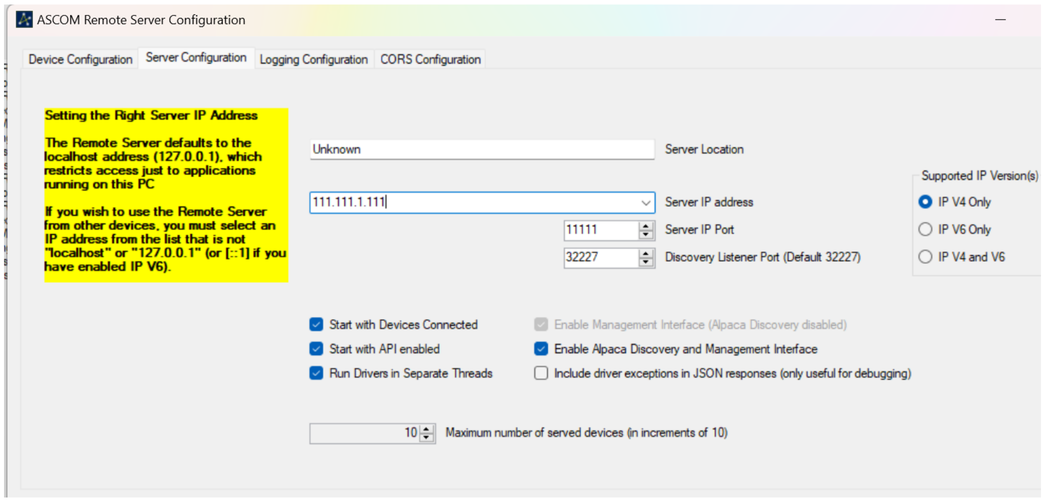
The first step to linking the created server to a network is finding out the network's IP address. Luckily, all you need to do for this is to make sure you are connected to your network by whatever means (via Internet or Ethernet). It is recommended that you use an Ethernet connection because a wired connection has the fastest speed for the transmission of data. Large astronomy images will be transmitted through the network, so speed is key. Once the connection is made, you can go into the computer's terminal and type “IPconfig”



```
PS C:\Users\craig> IPconfig

Windows IP Configuration


Ethernet adapter Ethernet:

   Connection-specific DNS Suffix  . :
   Link-local IPv6 Address . . . . . : fe80::75be:e52c:224a:ed68%8
   IPv4 Address. . . . . . . . . . . : 192.168.1.128
   Subnet Mask . . . . . . . . . . . : 255.255.255.0
   Default Gateway . . . . . . . . . : 192.168.1.1

Wireless LAN adapter Local Area Connection* 1:

   Media State . . . . . . . . . . . : Media disconnected
   Connection-specific DNS Suffix  . :

Wireless LAN adapter Local Area Connection* 2:

   Media State . . . . . . . . . . . : Media disconnected
   Connection-specific DNS Suffix  . :

Wireless LAN adapter Wi-Fi:

   Connection-specific DNS Suffix  . :
   Link-local IPv6 Address . . . . . : fe80::c091:8067:865c:f56c%15
   IPv4 Address. . . . . . . . . . . : 192.168.1.149
   Subnet Mask . . . . . . . . . . . : 255.255.255.0
   Default Gateway . . . . . . . . . : 192.168.1.1
```

Multiple options occur regarding the sources of an IP address. In the above example, two forms of an network connection are present. The “Ethernet adapter Ethernet” is the Ethernet connection while the “Wireless LAN adapter Wi-Fi” is the WiFi connection. Both sources of network have similar attributes, but the one we care about is the IPv4 Address. This should be four sets of numbers connected by a period like “111.111.1.111”. You can directly copy this number by highlighting the text, and press ctrl+shift+c. When copying and pasting things in the terminal, you must also press shift.

Now that you know your IPv4 address, you can now use this piece of information to create the server with this network. If you go back the the ASCOM Remote Server Setup and go to the "Server Configuration" tab, there is a spot for you to paste the IPv4 address into. It is labeled as "Server IP address."



Setting the Right Server IP Address

The Remote Server defaults to the localhost address (127.0.0.1), which restricts access just to applications running on this PC

If you wish to use the Remote Server from other devices, you must select an IP address from the list that is not "localhost" or "127.0.0.1" (or [::1] if you have enabled IP V6).

Server Location: Unknown

Server IP address: 111.111.1.111

Server IP Port: 11111

Discovery Listener Port (Default 32227): 32227

Supported IP Version(s): [✓ IP V4 Only] IP V6 Only IP V4 and V6

✓ Start with Devices Connected

✓ Start with API enabled

✓ Run Drivers in Separate Threads

✓ Enable Management Interface (Alpaca Discovery disabled)

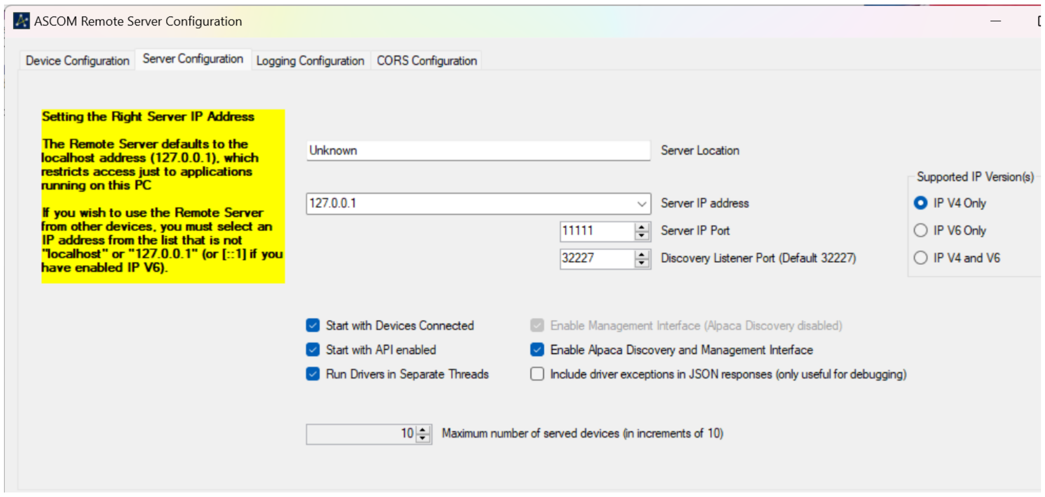
✓ Enable Alpaca Discovery and Management Interface

☐ Include driver exceptions in JSON responses (only useful for debugging)

Maximum number of served devices (in increments of 10): 10

Server via Local Host

If you wanted to avoid the hassle of using a Network to create a server and you just want your computer to be the server, this is the choice for you. This option will also give you the highest performance since it is just running through your computer. By default, the ASCOM Remote Server Configuration should set the Server IP address to be “147.0.0.1”. This limits every interaction occurring between the astronomical hardware, and the Python Program to your computer and nowhere else. This ensure top security and fast performance.



ASCOM Remote Server Configuration

Device Configuration | Server Configuration | Logging Configuration | CORS Configuration

Setting the Right Server IP Address

The Remote Server defaults to the localhost address (127.0.0.1), which restricts access just to applications running on this PC

If you wish to use the Remote Server from other devices, you must select an IP address from the list that is not "localhost" or "127.0.0.1" (or [::1] if you have enabled IP V6).

Server Location: Unknown

Server IP address: 127.0.0.1

Server IP Port: 11111

Discovery Listener Port (Default 32227): 32227

Supported IP Version(s): [✓ IP V4 Only] IP V6 Only IP V4 and V6

- [✓] Start with Devices Connected
- [✓] Start with API enabled
- [✓] Run Drivers in Separate Threads
- [✓] Enable Management Interface (Alpaca Discovery disabled)
- [✓] Enable Alpaca Discovery and Management Interface
- [ ] Include driver exceptions in JSON responses (only useful for debugging)

Maximum number of served devices (in increments of 10): 10

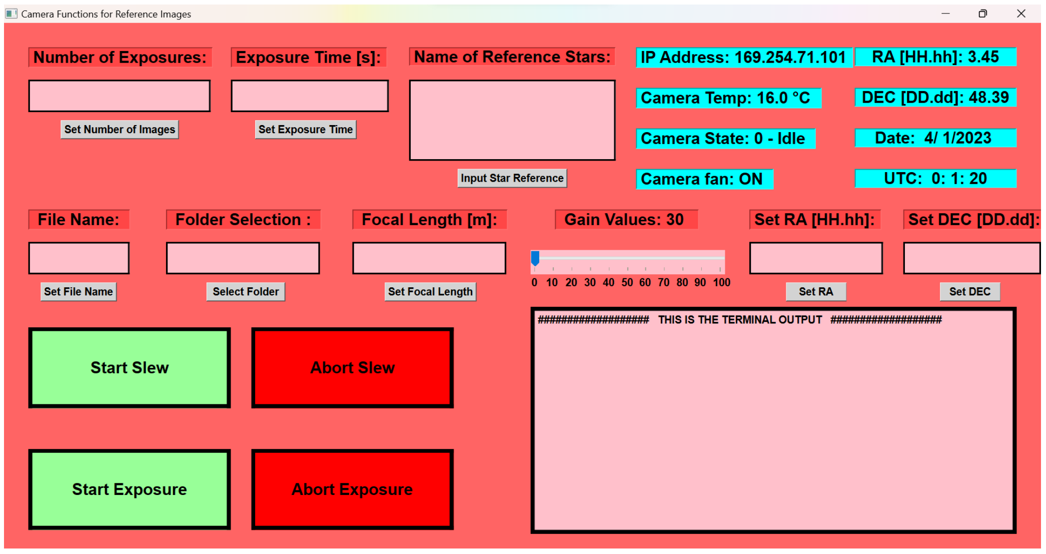
iii Network vs Local Host

The ease of using the latter option for the server choice prompts an initial question of why would someone choose to use a real network as a choice for their server. A reason to use a network is the ability to use more than one computer. The use of a network allows for the computer connected to all the hardware to not have to run the Python Program. You can treat this computer as your “Home Base” PC (personal computer). The Home Base PC can have all the ASCOM applications, drivers, and hardware plugged into it, and you can have a separate computer running only the Python Program. This User Manual is phrased to where there isn’t a separation between the Home Base PC and the PC running the Python Program to try to keep it simpler. In order to run the Python Program on another computer (we will call it the PPC for Python Personal Computer), you would only need to install the Python IDE and the PIP packages (Python, PIP, and Visual Studio Setup Installations) while the Home Base PC would only need to install the ASCOM stuff (ASCOM, and Driver Installations).

4 The Python Program

The Python program actually consists of two different programs. The first one focuses on the the astronomical equipment automation process while the other consists of image processing. You will only be having to interact with the astronomical equipment automation process so that will be the main focus.

The physical Python script consists of many moving parts. The explanation of the program will be focused on an audience with no programming or astronomy experience. The Python Program's goal is to educate and introduce you to astronomy. As you can see, the program requires a lot of astronomical inputs and displays so an in-depth explanation is needed.



IP Address: 169.254.71.101
RA [HH.hh]: 3.45
Camera Temp: 16.0 °C
DEC [DD.dd]: 48.39
Camera State: 0 - Idle
Date: 4/ 1/2023
Camera fan: ON
UTC: 0: 1: 20
Gain Values: 30

```
##################   THIS IS THE TERMINAL OUTPUT   ##################
```

i Displayed Information

Eight pieces of information are displayed in the program. The first is the IP address in which the server is linked to. Next is information on the camera such as its temperature of operation, whether the fan is on, and its state. The last displays is the telescopes celestial coordinates (Right Ascension and Declination), and current date/time. The camera state and fan operation are the only displayed values that will not be stored in the image data.

5 References

Bruns, Donald G. “Gravitational starlight deflection measurements during the 21 August 2017 total solar eclipse.” Classical and Quantum Gravity 35.7, 075009 (6 March 2018). https://doi.org/10.1088/1361-6382/aaaf2a.

Dittrich, William A. (Toby). Proposal to Oregon Space Grant Consortium Affiliate Faculty Research Incubator Program (AFRIP). 27 May 2022.

Dyson, Frank Watson, et al. “IX. A determination of the deflection of light by the sun’s gravitational field, from observations made at the total eclipse of May 29, 1919.” The Royal Society 220, 291-333 (1 Jan. 1920). https://doi.org/10.1098/rsta.1920.0009.

Grant, Andrew. “A record-setting Eddington experiment.” Physics Today (21 Aug. 2018). DOI:10.1063/PT.6.3.20180821a.
http://physicstoday.scitation.org/do/10.1063/PT.6.3.20180821a/full/

Kennefick, Daniel. ”Testing relativity from the 1919 eclipse—a question of bias.” Physics Today 62, 37-42 (2009). https://doi.org/10.1063/1.3099578.