

HOLOGRAM CREATION PYTHON

Version 4.0

ABSTRACT

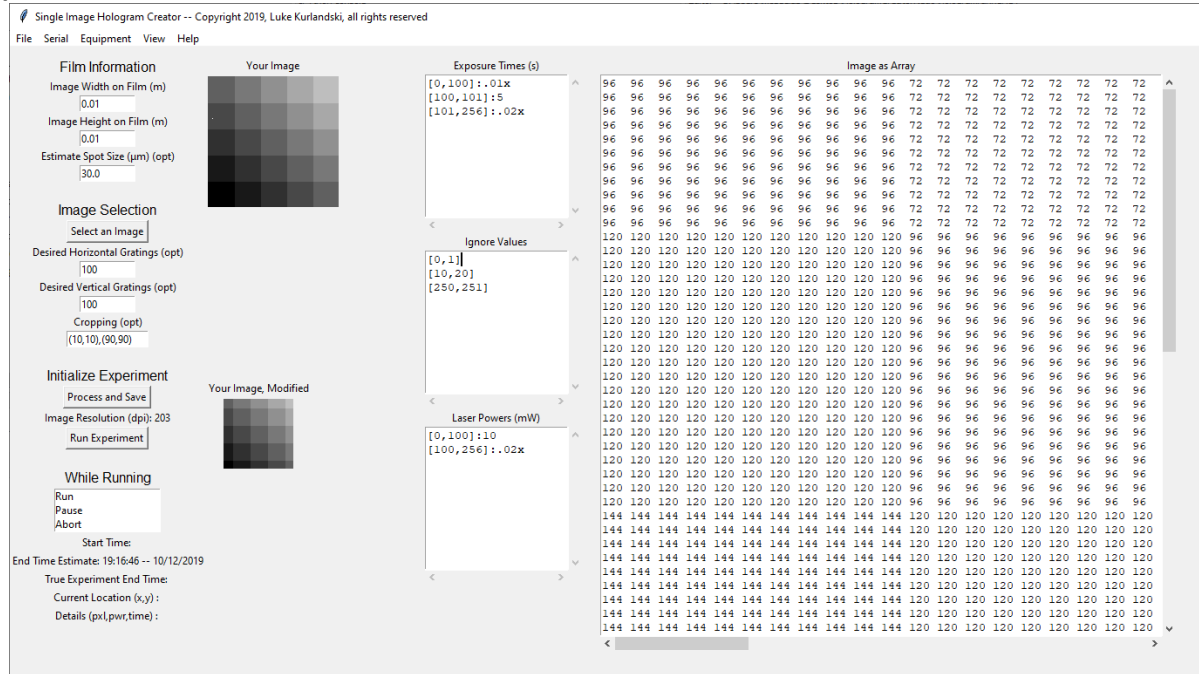
Hologram creation in Python involves intelligent programming in a beautiful language and a smooth, reliable user interface.

Luke Kurlandski,
December 2019

TCNJ Department of
Physics, Dr. McGee
Laser Lab

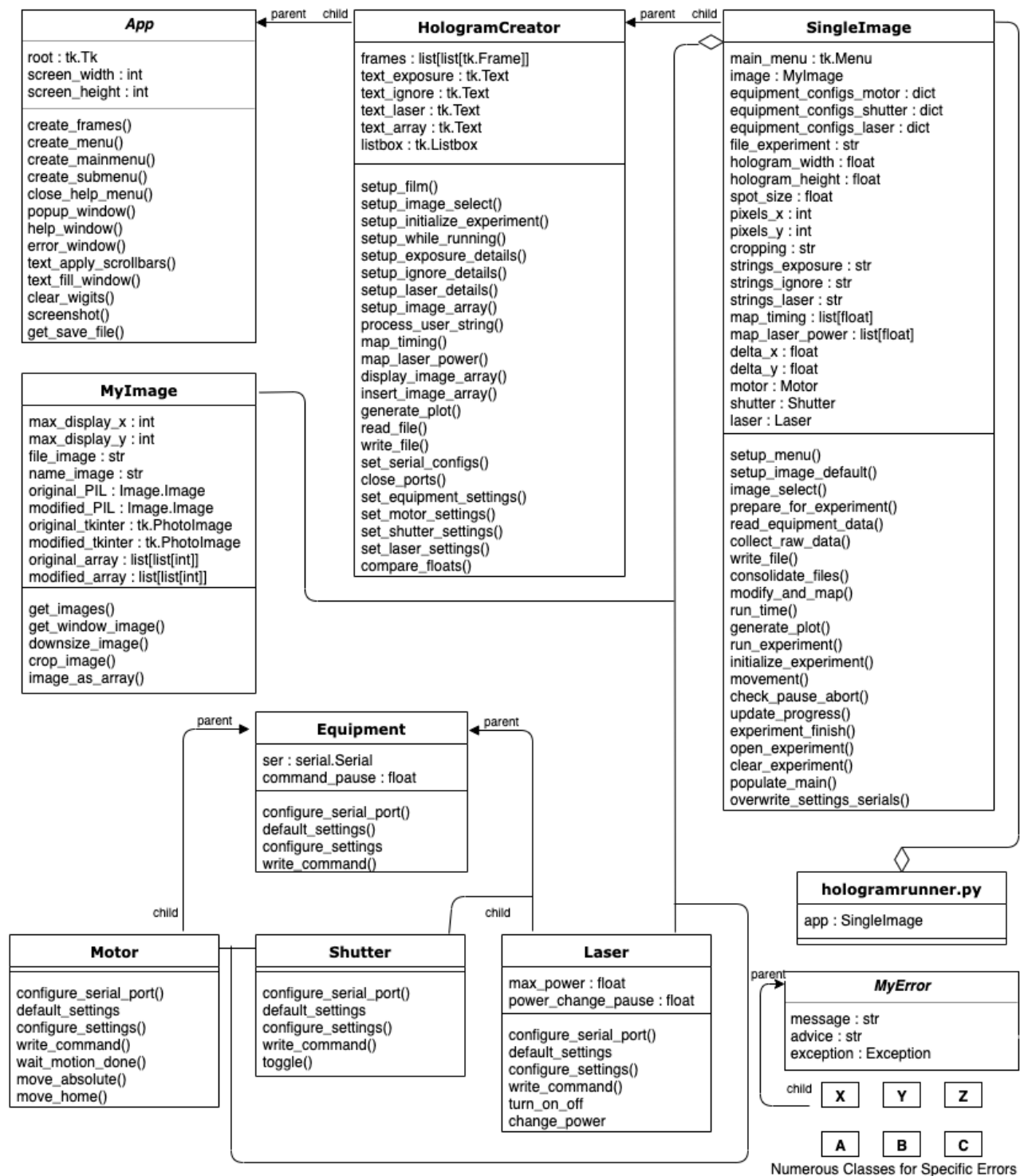
Table of Contents

Software	2
UML Diagram	2
Class Descriptions	3
SingleImage Class	4
Equipment Class	5
Usage at Glance	6
Physical Setup	6
Program Usage	6
Detailed Usage	7
Film Information	7
Image Selection	7
Exposure Information	7
Initialize Experiment	8
While Running	8
File Menu	9
Serial Menu	9
Equipment Menu	9
View Menu	10
Help Menu	10



Part I: Software

UML diagram



Class Descriptions

App

This class provides a basic framework to create a variety of graphical user interfaces for any experiment needed. The capabilities of this class include sophisticated exception handling, menu creation, and popup windows. When making a tkinter program, have it inherited from App to write less code. Note, this class is abstract and should not be instantiated.

GenericCreator

This class provides the functionality to perform many operations that would be needed for any sort of image-creation experiment. At the time of writing, McGee lab focusses on creating a single image upon a hologram. In the future, we may choose to alter this experiment significantly. However, much of the programming functionality we would need for a different kind of experiment is already included in GenericCreator. Thus, inherit from GenericCreator to take advantage of many of the functions created, such as processing user strings into mappings, writing python dictionaries to .txt files, and configuring machinery. Note, this class is abstract and should not be instantiated.

SingleImage

This class is designed to provide a simple, user interface to facilitate the creation of single-image holograms upon a film. The class takes user input, processes it, and drives machinery to print an image hologram. This class is the first non-abstract class discussed. This class will be discussed in great detail in a further section.

MyImage

This class is used to provide image processing across two different python imaging libraries: PIL and tkinter. The class has the capability to downsize, convert, and crop images. Each MyImage object has an original image and a modified image. Each image is represented in three different forms: as a PIL image, a tkinter PhotoImage, and as an array.

Equipment

This class allows easy use of machinery that uses serial ports. Inheriting from this class provides a solid framework for how any serial-port-using device should operate. Its child classes are detailed below. Each child uses method overriding to raise precise Errors when operations go wrong. Each child class should be expanded as needed to include more equipment configurations and more functionality. Note, this class is abstract and should not be instantiated. The concrete child classes are as follows:

1. Motor: used to configure and operate Newport motors.
2. Shutter: used to configure and operate the shutter.
3. Laser: used to configure and operate lasers.

MyError

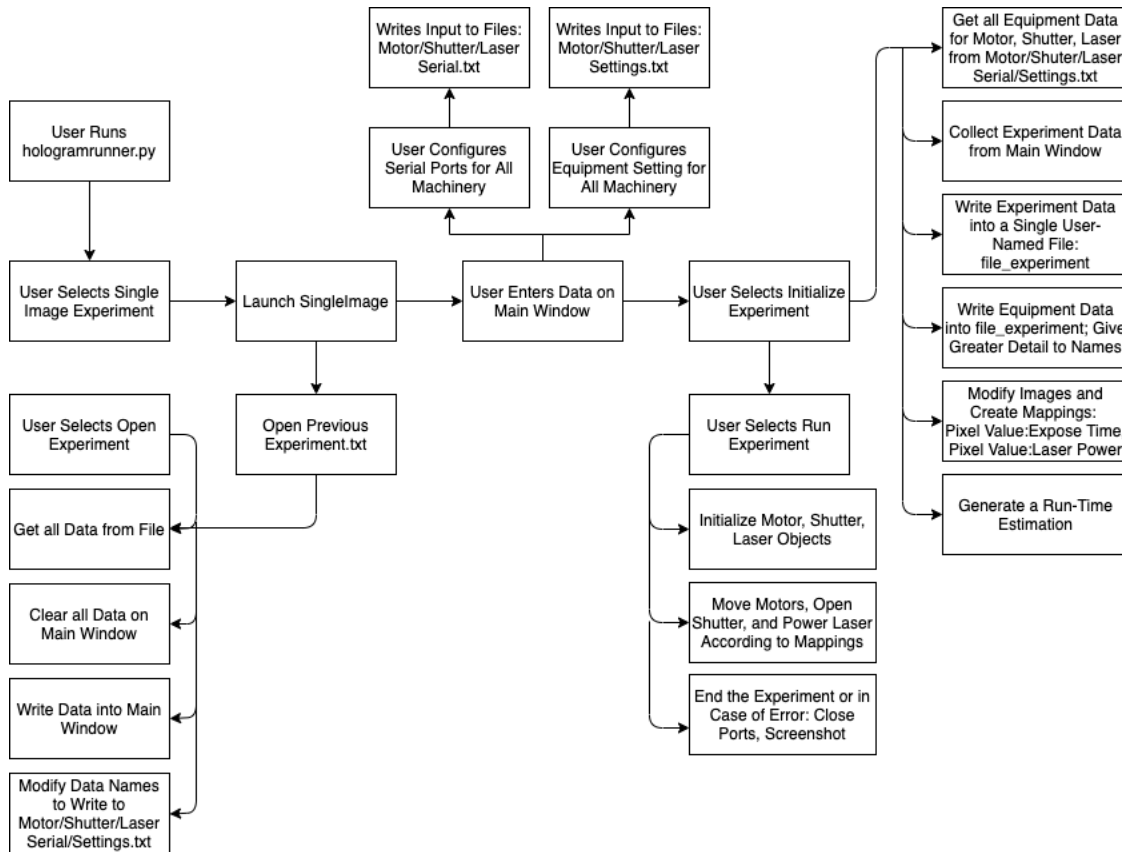
This is an exception class, used to raise precise exceptions for various procedures that can arise. Its child classes are: FileFormatError, InputError, NoFileError, MissingDataError, UnknownError, UserInterruptError, EquipmentError, MotorError, ShutterError, and LaserError.

hologramrunner.py

This is not a class, simply a file. This is the file which should be run to select which type of experiment the user wishes to run. As more hologram creation classes are created, presumably inheriting from GenericCreator, more choices for experiment should be incorporated into this runner file.

SingleImage Class

SingleImage is the driving class behind the single image creation process (as of now). This does not mean it is “the most important one,” rather, it simply holds the reins to the process. Below is a flow chart of the processes that occur to create an image hologram.



Throughout this whole process, SingleImage does very little heavy lifting. It uses the methods of App, GenericHologram, MyImage, and Equipment to do most of the complicated work for it. This results in a class that is simple to read, maintain, and expand. This is the art of object-oriented programming.

The reader may wonder why there are two sets of files being written to. This is by far the most complicated aspect of SingleImage. The reason for this is to ensure data is compatible with the Equipment class.

In the `Motor/Shutter/Laser Serial/Settings.txt` files, data is written with a simple title. This title becomes the key in a python dictionary, which is compatible with the Equipment class. For example, the `Laser Serial.txt` file has an entry that looks like this:

```
Baudrate::
19200
#####
```

When the data is written into `file_experiment`, it must have a more descriptive name, since three devices all have a 'Baudrate'. The entry is processed and modified to look like this:

```
Laser Serial Baudrate::
19200
#####
```

The descriptive form allows the experimenters and the program to understand exactly what parameters are being used. However, these phrases are not compatible with the Equipment class and must be altered before they can be used by machinery.

The Motor/Shutter/Laser Serial/Settings.txt files are re-written every time the user opens up the window to enter this information. When the user selects the button to initialize the experiment, these files are read, given more descriptive titles, and are then appended to the file_experiment.txt file.

When an experiment is opened, such as in the initial start of the program, the opposite of this process occurs. The file_experiment.txt file is read, its data titles are trimmed into a less descriptive form, and they are stored in the correct Motor/Shutter/Laser Serial/Settings.txt files. The experiment data not concerning equipment is loaded into the main window.

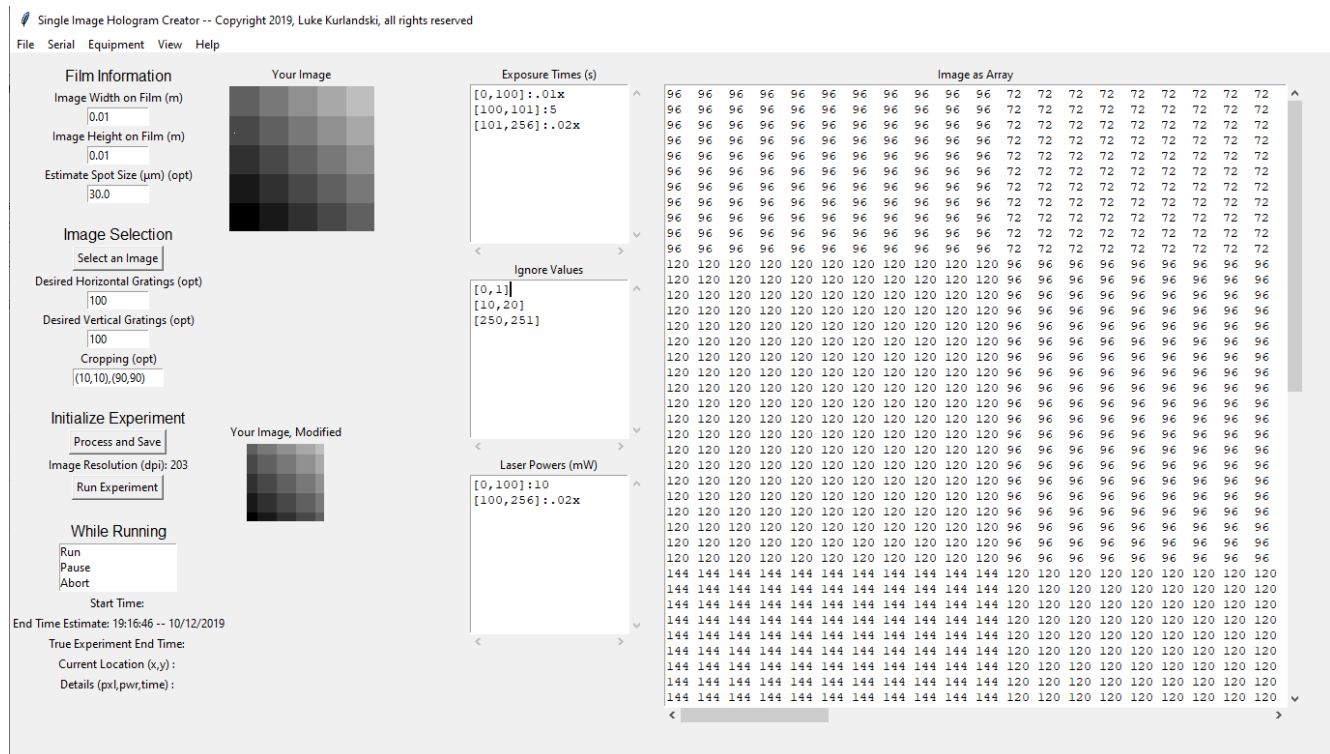
Equipment Class

The Equipment, Motor, Shutter, and Laser classes are used to control the respective equipment associated with their name. These devices all use a serial port to read and write commands to them. The Equipment class provides a framework for how the Motor, Shutter, and Laser classes should appear and provides some useful functions which may be used by all kinds of equipment.

Below is a snippet at how one should create a Motor object to control the Motor. For the full capabilities, read the documented code.

```
1  """Test various stuff file"""
2
3  #Import the module that contains the Motor class.
4  from serialcontrol import Motor
5
6  #Create a dictionary that stores the configurations.
7  serial_and_setting_configurations = {
8
9      #Describe the serial configurations for the device.
10     'Port' : 'COM7',
11     'Baudrate' : 19200,
12     'Timeout' : .1,
13     'Stopbits' : 1,
14     'Bytesize' : 8,
15     'Parity' : 'N',
16
17     #Describe the settings for the device
18     'Command Pause' : .1, #Pause between writing to port
19     'Velocity' : 1, #Speed of motor in mm/s
20     'Acceleration' : 4, #Acceleration of motor in mm/s/s
21     'Deceleration' : 4, #Deceleration of motor in mm/s/s
22     'Axes' : [1,2] #List of axes that will be used
23 }
24
25 #Create an object with above configurations.
26 myMotor = Motor(serial_and_setting_configurations)
27
28 #Using methods of the object, do whatever you want!
29 myMotor.move_home(axis=1) #Move axis 1 home.
30 myMotor.move_home(axis=2) #Move axis 2 home.
31 myMotor.move_absolute(axis=1, go_to_position=10) #Axis 1 to 10.
32 myMotor.move_absolute(axis=2, go_to_position=-4.6) #Axis 2 to -4.6.
```

Part II: Usage at a Glance



Above: main window of SingleImage

Physical Setup

1. Set "home" position on motor: manually move all axes to 0; turn motor off
2. Turn on shutter
3. Turn on laser; give time to warm up
4. Put the hologram in cradle

Program Usage

1. Film Information: how large the image should appear on the hologram
2. Image Selection: choose an image and modify it as desired
3. Exposure Times: how long a certain pixel values should be exposed
4. Ignore Values: which pixel values should be ignored
5. Laser Powers: what power to use for certain pixel values
6. Serial: enter correct serial port details for the shutter, motor, laser
7. Equipment: enter desired equipment settings for shutter, motor, laser
8. View: view full screen array representation of image; chart of pixel mappings
9. Initialize Experiment: process the data; run experiment
10. While Running: pause or abort an experiment as desired; view experiment progress

Part III: Detailed Usage

Film Information

- Image Width/Height on Film (float)
 - How wide the image should appear on the hologram, in meters
 - Used to determine the path of motion for the motor
- Estimate Spot Size (float)
 - Estimate the diameter of a typical grating, in microns
 - Optional, is not used in any computations

Image Selection

- Select an Image (.png/.jpeg)
 - Press this button to open up a file explorer to select an image for printing
 - Tested thoroughly with .png and .jpeg images
- Desired Horizontal/Vertical Gratings (int)
 - How many gratings/pixels should appear on the image in the horizontal/vertical direction
 - Optional, if blank, the program will use original dimensions
- Cropping (string)
 - Coordinates to crop, after downsizing image
 - Format:
(x1,y1),(x2,y2)
 - Program will use portion of image lying within these coordinates:
(x1,y1) ; (x1,y2) ; (x2,y1) ; (x2,y2)

Exposure Information

- Exposure Details (string)
 - A mapping of pixel value to exposure time
 - Format:
 - [{start range},{end range}]:{open time}
 - [{start range},{end range}]:{scalar multiple}x
 - Notes:
 - {} – indicates where the user should insert a value
 - start range (int) – pixel value beginning the range, included in range
 - end range (int) – pixel value ending the range, not be included in range
 - open time (float) – seconds the shutter should remain open
 - scalar multiple (float) – multiplies this number times pixel value to produce a variable exposure time
- Ignore Details (string)
 - Override the exposure mapping with pixel values that should be ignored
 - Optional, if blank, does not alter experiment
 - Format:
 - [{start range},{end range}]
 - Notes:
 - {} – indicates where the user should insert a value
 - start range – pixel value beginning the range, included in range
 - end range – pixel value ending the range, not be included in range

- Laser Details (string)
 - A mapping of pixel value to laser power
 - Format:
 - `{{start range},{end range}}:{open time}`
 - `{{start range},{end range}}:{scalar multiple}x`
 - Notes:
 - `{}` – indicates where the user should insert a value
 - start range (int) – pixel value beginning the range, included in range
 - end range (int) – pixel value ending the range, not be included in range
 - open time (float) – seconds the shutter should remain open
 - scalar multiple (float) – multiplies this number times pixel value to produce a variable exposure time
 - Laser does not change power immediately; set laser pause period to 4-5 seconds

Initialize Experiment

- Initialize Experiment
 - Select to:
 - Pull data from main window
 - Pull data from equipment files
 - Downsize and crop image
 - Create pixel value mappings
 - Generate run time estimate, dots per inch
 - Store all data into user-named file
- Image Resolution
 - Number of gratings in one horizontal inch of the hologram
- Run Experiment
 - Select to:
 - Establish connection with machinery
 - Read array representation of image
 - Drive motors, open shutter, alter laser power according to pixel value
 - Take a screenshot at conclusion of experiment

While Running

- List box
 - Run – program will run
 - Pause – program will pause; press run to un-pause
 - Abort – program will close serial devices and cancel experiment in progress
- Start Time
 - Time user started the experiment, when the run button was pushed.
- Experiment End Time
 - Records when the experiment terminates.
- End Time Estimate
 - Performs an estimation as to how long the experiment will take.
- Current Location/Details
 - Describes the current pixel being exposed.
 - Describes the current pixel value, laser power value, and exposure time for current pixel.

File Menu

- Quit
 - Safely close the application
 - If running experiment, abort first, then select 'Quit'
- Open
 - Open any correctly formatted previous experiment
- Open Example Experiment
 - Open a sample experiment to display how to use this program
- Open Previous Experiment
 - Open the previous experiment

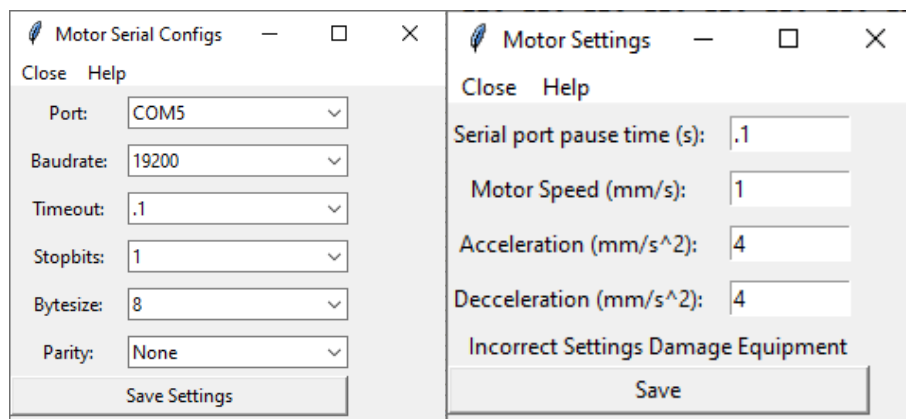
Serial Configurations Menu

- Allows the user to configure the serial connections with equipment
- Stored in Motor/Shutter/Laser Serial.txt
- For correct configurations, read the user manual for piece of equipment

Equipment Settings Menu

- Configure settings for each piece of equipment used
- Carefully read user manual for equipment; incorrect entry may damage equipment
- Motor
 - Serial Command Pause Period (float): waiting period between writing commands to serial port; dependent upon the quality of cable used in experiment, in seconds
 - Velocity (float): speed of stages, in mm/s
 - Acceleration (float): acceleration of stages, in mm/s/s
 - Deceleration (float): deceleration of stages, in mm/s/s
- Shutter
 - Serial Command Pause Period (float): waiting period between writing commands to serial port; dependent upon the quality of cable used in experiment, in seconds
 - Operating Mode (int): describes the operating mode to use
- Laser
 - Serial Command Pause Period (float): waiting period between writing commands to serial port; dependent upon the quality of cable used in experiment, in seconds
 - Maximum Laser Power (float) : maximum power capable of the laser, in mW
 - Power Change Pause (float) : waiting period between changes of laser power, in seconds

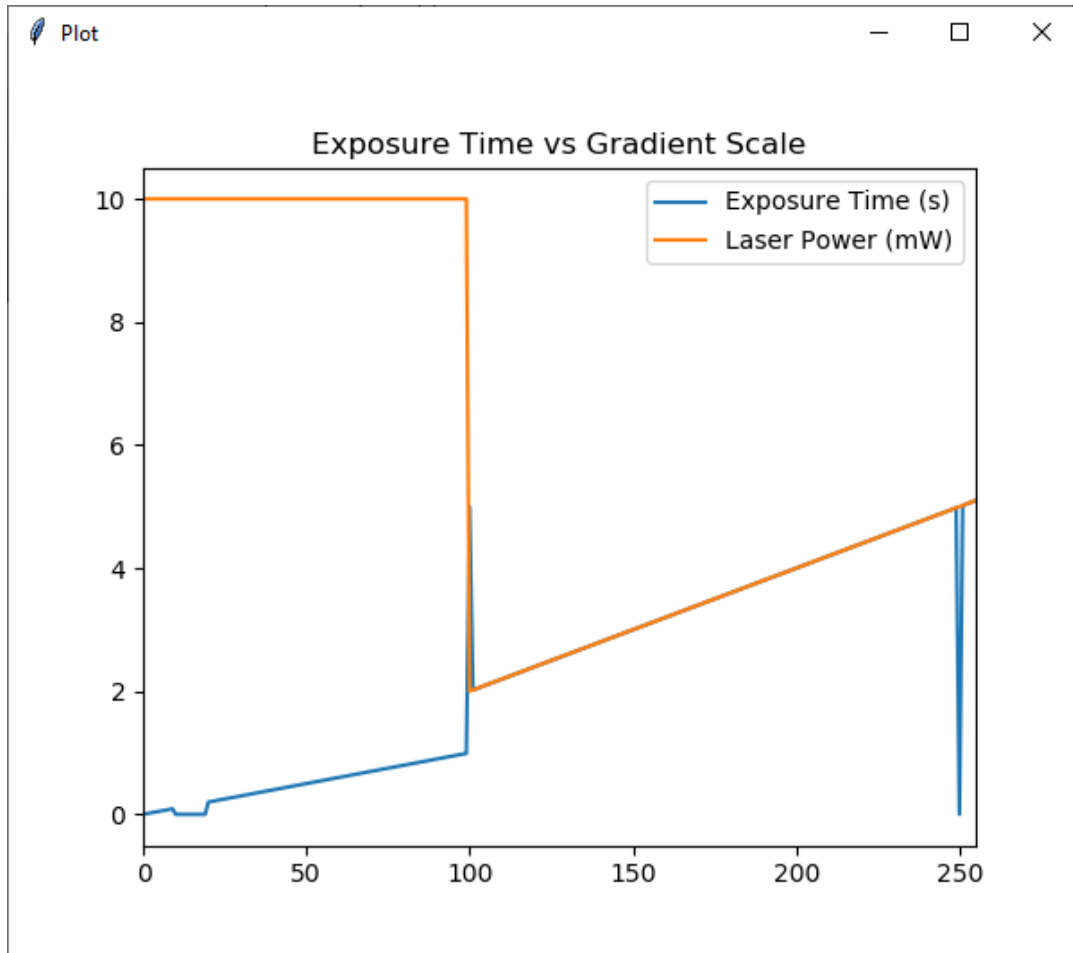
Below: serial configurations window for motor (left); equipment settings window for motor (right)



View Menu

- Image as Array: launch a popup window of the image represented as an array
- Mappings Chart: launch a popup window with a graph displaying mappings

Below: chart of mappings vs pixel value



Help Menu

- Launch a popup window containing related information
- Contains a save option, which will save any changes the user made

