General Guide for Experimental Build

TL;DR: Run "Experiment Control (python).py" in Spyder 32-bit to execute an experiment.

What's in this "Main Python Experimental Build folder?":

1. 3 Python Programs
   1. Experiment Control (Python) Ver2.py
      1. Calls the other 2 Python files to run the whole GUI and execute experiments
   2. Connections\_Classes\_Ver2.py
      1. Contains two classes:
         1. Camera – a class for managing camera connections, taking pictures, processing pictures, and the part of the GUI related to the camera.
         2. allNonCamConnectClass – a class for managing all connections in the build other than the camera as well as the part of the GUI related to the non-camera connections (which are all basically serial connections). It is used for controlling:
            1. The stepper motor aka PGen aka Arduino Leonardo
            2. The shutter motor aka Arduino Uno
            3. The voltage generator aka Agilent aka VGen
   3. Experiment\_RunningLoadingDesign\_Classes\_Ver2.py
      1. Contains two classes:
         1. experimentClass – a class for loading and running experiments. It manages the part of the GUI related to these aspects of the experiment. It also can call the creation of a designWindow instance.
         2. designWindow – a class that builds a window containing a text editor with aids to design an experiment preset file.
2. 5 Folders
   1. \_\_pycache\_\_ - not important. Generated automatically by Python.
   2. Experiment\_Info\_and\_Pictures – contains folders which contain raw pictures and information regarding the parameters used to run the experiment. Inside these folder can be a folder that contains the processed pictures if they have been processed using the GUI.
   3. Image\_Processor\_ARGV – contains an executable for processing raw picture txt files into png files. The multitude of files and folders in this folder are needed for the executable to run quickly, so make sure to not delete those dependencies, or it won’t work. This executable is a compiled version of Image\_Processor\_ARGV.py, found in the same folder. You can just call the .py from the command line to process an individual file if you want.
   4. Sample Experimental Design Files – Some experiment design txt files that we’ve tested as examples. This folder is not needed for running, but it’s nice.
   5. Temporary Image Files for GUI – When we take pictures to update the image preview in the GUI, we save a txt file and a png file in this folder. You should have this folder with this name, but it doesn’t matter what’s already in it.
3. 1 .dll – QcamApi\_AlexTakeAPic\_Ver2.dll – Alex-constructed library in C++ that is a wrapper for calling the SDK from Python. This is 100% necessary for running, as it contains the functions for controlling the camera. This is compiled in 32-bit, so we cannot run Python in the 64-bit.
4. 1 .lib – QcamDriver.lib – The SDK used to contain most of the drivers for the camera. 100% necessary for controlling the camera. This is compiled in 32-bit, so we cannot compile the .dll in 64-bit, which means that we must have the Python in 32-bit.
5. This Readme – duh.

Gain and Exposure

Take standard image, several frames in sequence and see how consistent the intensity is. As fast as we can and see if we get consistent image.

Seee how long it takes to snap a photo using timing.

Look into processing binary files with imread

Finish documentation -> Alter library to add gain and exposure as arguments -> Alter python to add gain and exposure as arguments and change experimental design and loading function -> test and confirm -> command line arguments executable