Software Modules:

MainV1.py	Main program that calls all other modules
Settings	Attempts to read configuration from Magnitude.io server, if that fails reads the configuration from a locally stored file. Should that fail as well, it uses “hard-coded” parameters contained within.
StoreData	Stores data in local text file (ExoLab_Data.csv)
Device Interfaces:	
Camera	Controls the fisheye camera
LCD	Controls the 16x2 LCD display
LightPanel	Controls the LED light matrix
Sensors	Reads the temperature, humidity, carbon dioxide and light sensors
Network Interfaces:	
AdafruitIO.py	Interfaces to Adafruit.IO
AWS	Interfaces to Amazon Web Services for storage of JPG images
MIOWS	Interfaces to Magnitude.io’s server
Data and logfiles:	
ExoLab_Data.csv	ExoLab data file
ExoLab.log	Log file for debugging
DefaultSettingsV2.json	Default parameters in same JSON format as received from Magnitude.io server.

While the software implementation enables interfacing to the Magnitude.io network which requires a license, it can also be used standalone. Configuration parameters for lighting and timings are stored on the SD card in an editable JSON file, data measurements in a CSV file, and JPG images in a folder.

Additionally, this implementation enables use of Adafruit.io, a web-based platform for storing and visualizing data from Adafruit.com. The current implementation fits within the limits of the free version, but could be extended to enable more command and control functions with the paid version (currently $99 per year).

Software is coded in Python with the appropriate Adafruit and Raspberry libraries. Magnitude.io’s servers do not store the JPG images, just the URL to this image. In this implementation, images are stored on Amazon Web Services (S3) using a free account.