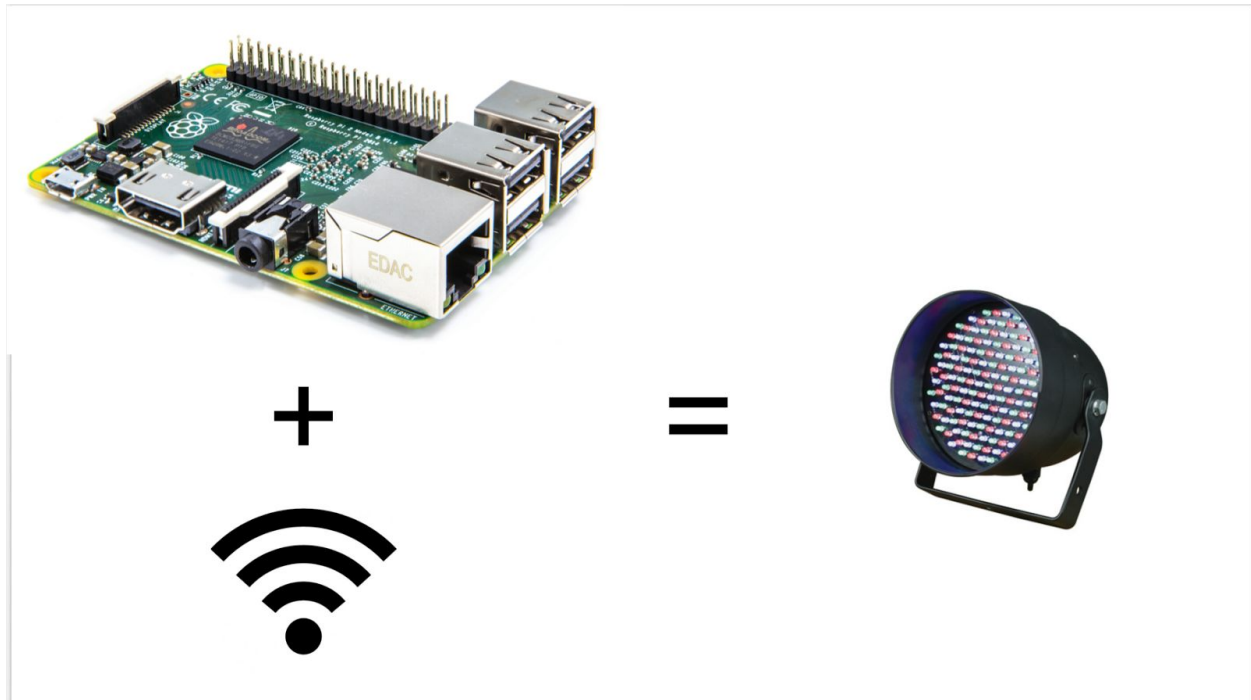


Initially, it was decided to control the DMX lights using a Raspberry Pi, Wifi, and Art-Net (a communications protocol for controlling DMX lights). The reasoning here was that Raspberry Pi natively uses Python on its operating system. A Python module that grabs API data from sensors across Carnegie Mellon campus had already been built, so the hope was that it would be a simple integration.



Soon, however, it became clear that the Raspberry Pi was not the best solution. Firstly, it became very difficult to connect the Raspberry Pi to the CMU Wifi network.

Network Registration

Welcome to the Network Registration System

Release 1.1.9

NOTE: As of 2/24/2014, activations should be done via cando.net.cmu.edu.

Users with Carnegie Mellon Andrew IDs can use this system to register machines to be connected to the campus network and activate network outlets.

All machines must be registered through this system prior to use on the campus network. [About Network Registration](#).

Important

By using this system, you agree to abide by the policies and guidelines concerning the use of computer, network, and telecommunications resources. These policies and guidelines govern the use of such resources. Please read the [Carnegie Mellon Computing Policies and Guidelines](#).

If you [do not agree to abide by these guidelines](#), you may not use this system.

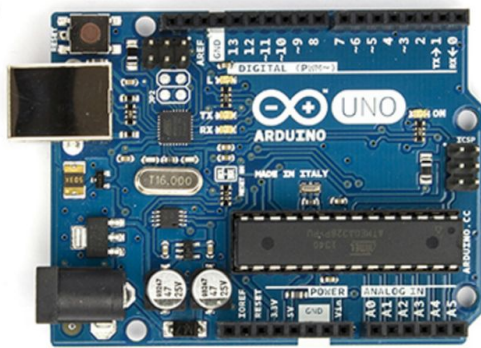
By clicking "Enter", you indicate that you have read, understand, and agree to the policies and guidelines.

Enter

The first step in connecting to the network was registering the Raspberry Pi through CMU. The Raspberry Pi we were using did not have native Wifi, so we initially tried connecting it via an ethernet cord. This proved to be difficult, and one phone call to technical support later we realized that we would have to purchase a Wifi adapter for the Raspberry Pi.

Two days later when the part had arrived, we attempted to connect the Raspberry Pi directly to the Wifi after registering it. This, too, proved to not work. What's more, there was still the matter of figuring out how to control the overhead lights using Art-Net. Through Python could be used to grab specific API data, an Art-Net Python module had not yet been developed.





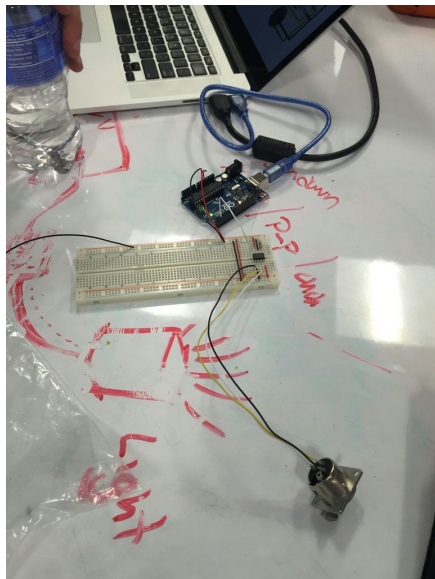
+

=



These two factors in mind, we decided to switch to Arduino. The switch was beneficial for multiple reasons. Firstly, processed API data could be sent directly to Arduino via serial, so there was no need to develop modules for the networking component. In addition, Arduino could communicate directly to the DMX lights using a DMX cable and already written Arduino code. This took off the strain of trying to figure out how to use Art-Net.

To communicate from Python to Arduino via serial, an easy to use Python module called PySerial was used and already developed Arduino code was modified. Working directly with Python made it easy to also utilize the Python module used for grabbing API data.



With demo Arduino code uploaded and DMX cables connected, we were met with success. The lights could be successfully controlled using Python on a computer, an Arduino, and a DMX cable.



Green was meant to signify *good* energy consumption and Red was meant to signify *bad* energy consumption.

However, the issue here was that these lights did not seem organic enough. From here, we decided to switch instead to incandescent light bulbs.



+

=



In Python, a demo showing the full range of the lights was developed. In addition to this, we developed code that actually used the live API data. Finally, Lumofracto was created.