|  |
| --- |
|  |
| Auroral Activity Indicator (AAI) |
| Implementation Guide and Project Documentation |
|  |
| **Dr. Richard Hechter**  **Karen Latimer** |
|  |

|  |
| --- |
|  |

Table of Contents

[Project Scope 2](#_Toc786576)

[Materials and Technical Requirements 3](#_Toc786577)

[Materials: 3](#_Toc786578)

[Technical Requirements: 3](#_Toc786579)

[Downloading Canopy: 3](#_Toc786580)

[Connecting To Philips Hue: 3](#_Toc786581)

[Security 4](#_Toc786582)

[Task Breakdown 4](#_Toc786583)

[1. Connect lights to Wi-Fi 4](#_Toc786584)

[2. Code the python application 4](#_Toc786585)

[3. Testing 4](#_Toc786586)

[4. Assemble and Install 4](#_Toc786587)

[KP Scale 5](#_Toc786588)

[Development / Coding 6](#_Toc786589)

[Troubleshooting 8](#_Toc786590)

[Challenges 8](#_Toc786591)

[User Guide 8](#_Toc786592)

# Project Scope

In this project, we used Wi-Fi enabled coloured lights and KP values from NOAA, used to measure auroral activity to indicate probability of visible auroral displays. Retrieving data from NOAA the colour of the lights will change in accordance to the average value that is calculated and flashes 3 times. The process is repeated every 20 minutes.

The Philips Hue Bridge is connected to a Wi-Fi router and the code accesses the bridge through an IP address and a userid that will need to be generated.

The application is coded in Python 2 using the Canopy development environment.

# Materials and Technical Requirements

## Materials:

* Philips Hue multi-coloured light strip
* Philips Hue Bridge

## Technical Requirements:

* Python development environment (Canopy)
* Wi-Fi connection/ router access
* Philips Hue app

## Downloading Canopy:

* <http://store.enthought.com/downloads>
* Pick canopy for operating system (i.e. windows 10 64 bit)
* Create userid and password with enthought (optional)

## Connecting To Philips Hue:

The Philips Hue python package must be installed into the development environment. In a python command prompt use the following command:

*pip install phue*

To find the IP address of the Philips Hue Bridge it must be connected to the network. In a web browser go to

*Meethue.com/api/nupnp*

Make note of the IP address and add to the python code. Every new router will create a new IP address for the bridge

To connect to the bridge through the code, you have to create a user name in the bridge itself. Follow the instructions at

[*https://developer.meethue.com/develop/get-started-2/*](https://developer.meethue.com/develop/get-started-2/)

Copy the created username into the Python code.

# Security

The Philips Hue Bridge must be connected directly to the Wi-Fi router. If you are working on a secure network, a network professional for your organization should set it up.

# Task Breakdown

## Connect lights to Wi-Fi

* Download the Philips Hue app from the app store or google store.
* Connect the lighting system to the Wi-Fi using the directions in the app.

NOTE: on a secure network, an ICT or IST professional may need to set this up

## Code the python application

* Download Python development environment (canopy)
* Import phue module from github.com using Python command prompt
* Find bridge IP address (meethue.com) and create user name in the bridge (API debugger tool)
* Code using python for acquiring data, calculating average KP value, and controlling lights
* Debug

## Testing

* Create test data in order to test the entire data range and colour range
* Set the code to use the real data set

## Assemble and Install

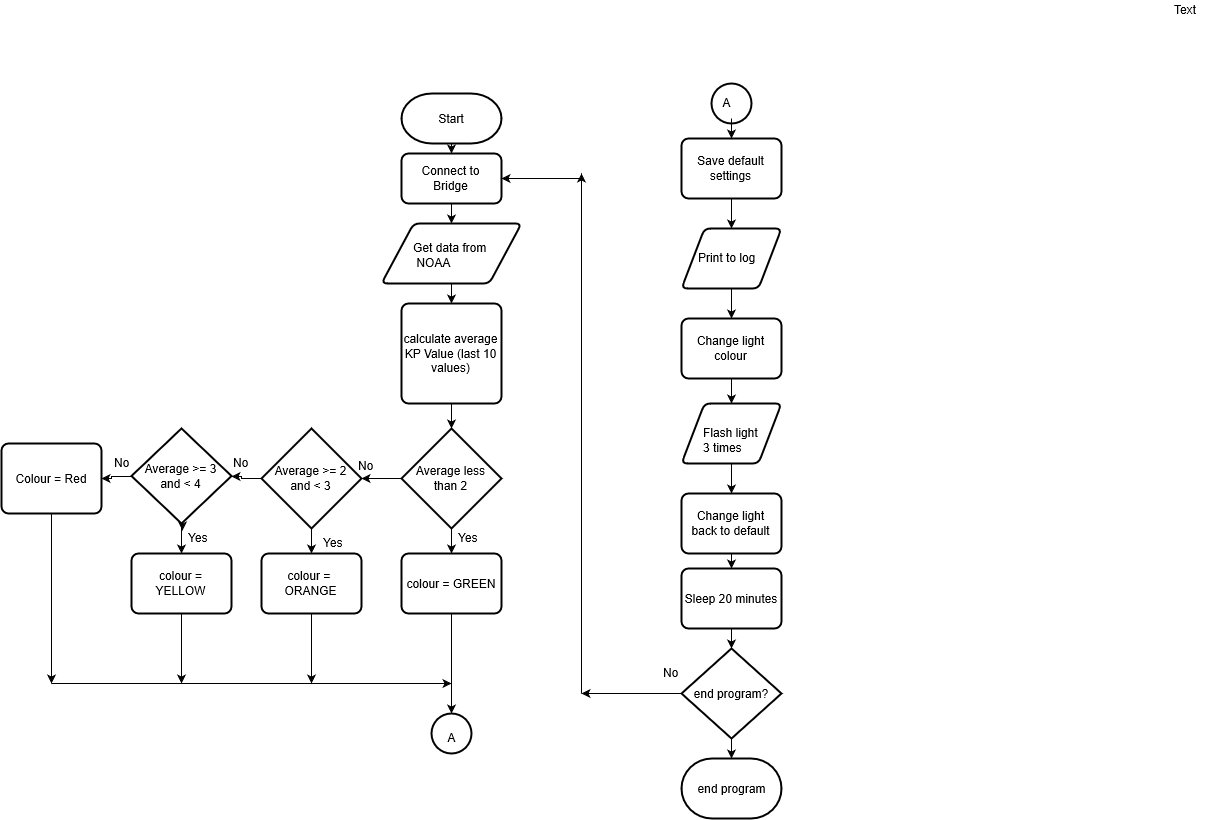
* Install lights in its desired location using included adhesive strips
* Connect bridge to Wi-Fi router and install in desired location

# KP Scale

|  |  |  |
| --- | --- | --- |
| **KP Value** | **Colour** | **Meaning** |
| 0 | Green | No chance of auroral activity |
| 1 | Green | Slight chance of auroral activity |
| 2 | Yellow | Fair chance of auroral activity |
| 3 | Orange | Moderate chance of auroral activity |
| 4 + | Red | Go Outside!  High probability of auroral activity |

# Development / Coding

**Flowchart:**

****

**Pseudo Code:**

*Use the python module phue from github.com for the official Philips Hue protocol.*

Connect light bridge to code through Wi-Fi

Select light source

Repeat every 20 minutes:

Retrieve JSON data from NOAA

Calculate average of last 10 entries

If the KP value is less than 2 colour = green

If the KP value is between 2 -3 colour = yellow

If the KP value is between 3-4 colour =orange

If the KP value is 4 or higher colour = red

Blink light three times

Write values to log

End repeat

End process with keyboard interrupt (control C)

# Troubleshooting

|  |  |
| --- | --- |
| My lights aren’t working | Check the power supply  Check the Wi-Fi connection |
| There is no data | Check NOAA source. If the website is down, wait and try again. If NOAA changed data file format, may require change in source code. |
| Program is not connecting to the bridge | Check the IP address and user name  (User name is created using the Philips Hue api debugger tool; IP address can be found at [www.meethue/api/nupnp](http://www.meethue/api/nupnp)) |
| Program not stopping/ Ctrl-C not working | Shutdown the canopy environment |

# Challenges

The challenging part of this project was connecting the python code to the Philips Bridge. Each new Wi-Fi router assigns the bridge a new IP address, so coding it off site requires a change in the IP address within the code itself. Also, procedures for generating the userid in the bridge that allows the code to access the API was not intuitive or well documented.

Dealing with the secure network on campus, and not having direct access to the Wi-Fi router was concerning. Cooperating with the IST department to gain access to a network connection was key to the successful final implementation on campus. If this is an issue for future implementations, consult with your organization’s network professionals.

# User Guide

1. Open Canopy and the AAI file
2. From the Run menu, select “Run File”
3. To end program, type Ctrl-C. The program must finish the sleep cycle before ending.

\*NOTE: In the future, an executable file may be available.