

Instructions for Silicon Stem Academy WiFi Lamp

Below are instructions for downloading and installing the code for the SSA WiFi lamp that was built for the Winter 2017 session.

Installing the Arduino IDE

In order to program the lamp, you'll first need to install the program that lets you write the code. This program is called an "Integrated Develop Environment", or IDE. Arduino provides a free, open-source IDE for programming your Arduino. To install it, go to <https://www.arduino.cc/en/Main/Software>, and download the IDE version applicable to your system. After the download, follow the instructions for your specific system for installing programs.

Installing the ESP8266 board definitions into the IDE

The WiFi microcontroller we used on the lamp is not an official Arduino. Because of this, there are a couple things that we need to do to allow the Arduino IDE to program the lamp microcontroller. There are clear instructions here: <https://learn.adafruit.com/adafruit-feather-huzzah-esp8266/using-arduino-ide>.

The two main steps are:

1. Install the required CP2104 USB driver to have the COM/Serial port appear properly. The drivers can be found here: <https://www.silabs.com/products/development-tools/software/usb-to-uart-bridge-vcp-drivers>. Find the driver for your system and download it. This will download a small program, that when run will install the driver onto your system for you. After you've run the program, restart your computer for it to take effect.
2. Install the ESP8266 board package. This is important for letting the IDE know how to talk to the WiFi chip. Open the Arduino IDE and follow these instructions:
 - a. Go to File->Preferences
 - b. In the box for "Additional Boards Manager URLs:" type: http://arduino.esp8266.com/stable/package_esp8266com_index.json
 - c. Go to Tools->Board->Boards Manager
 - d. search for "ESP8266 by ESP8266 Community"
 - e. Install the package once you've found it.
 - f. Once the package is installed, restart the Arduino IDE.

Once those steps are complete, you should be able to find the "Adafruit HUZZAH ESP8266" under Tools->Boards. This will be the chip you want to use.

Downloading the code

Go to www.github.com/SiliconStemAcademy/DeskLampWinter2017

Click on the “Clone or Download” button on the right side of the screen. A small window appears. Click on the “Download ZIP” button. Depending on how your system is set up, this will either automatically download the ZIP folder into your “Downloads” folder, or it will prompt you to save it in a location of your choice.

Once downloaded, go to the location of the download, and unzip the folder using the tool of your choice.

Move the folder and all its contents to a location that you’ll remember, so you’ll know where to find it in the future.

Installing the Libraries

Most of the libraries required by the code are already installed on your system. There are a few that are not, though. We have included them in the github folder, to make them easier to install. To install these libraries, follow these instructions:

1. Open the IDE
2. Go to “Sketch->Include Library->Add .ZIP Library”
3. Navigate to the “Required Libraries” folder, inside the SSA_DeskLamp_Student folder you downloaded.
4. Select one of the zipped library folders inside.
5. The program will tell you the library has been installed successfully on the blue bar near the bottom.
6. Once that has happened, repeat for the other zipped folders.

Setting up your lamp with your specific home network information

Note if you don’t already have line numbers in your Arduino IDE, go to File->Preferences and check the “Display line numbers” box. Then follow the instructions. Note that the line numbers will change as you add custom functions to the code, so these are approximate line locations.

1. Line 34: Uncomment “#define DEBUG” (remove the // in front of it)
2. Line 56: Change the SSID to the name of your home router
3. Line 57: Change the password to your router’s password
4. Upload the code using the procedure outlined below.
5. Once uploaded, open the Serial Monitor by selecting Tools->Serial Monitor
6. The Arduino will display the device’s assigned IP address on the serial monitor. Note every router will assign a different number, so you can’t skip these steps and copy your friend’s number.
7. Once you have the number, go to line 620 and replace the current IP address with the address of YOUR lamp.
8. Line 34: comment “#define DEBUG” (add the // in front of it)

Uploading the code onto the lamp

1. From the folder, open the SSA_DeskLap_Student.ino file. This will open the code in the Arduino IDE.
2. Go to Tools->Boards and make sure "Adafruit HUZZAH ESP8266" is selected
3. Go to Tools->Port and select the COM port for your device.
4. Press the Upload button on the top bar of the IDE.

Making your own state functions

If you want to make your own functions for how the lights behave, you need to follow the instructions below. Again note that the line numbers might change as you add functions, so these are approximate line numbers.

1. Write your custom function just above setup() in the code, near where all the other LED control functions are written, following the template below. The template can also be found at the very bottom of the code, so it can be copy/pasted for easier use.

```
1202 void your_function_name(void)
1203 {
1204     static uint8_t red_value = 0;
1205     static uint8_t green_value = 0;
1206     static uint8_t blue_value = 0;
1207
1208     //type your code here
1209
1210     for(int ii = 0; ii < NUMPIXELS; ++ii)
1211     {
1212         strip.setPixelColor(ii, strip.Color(red_value, green_value, blue_value));
1213     }
1214 }
1215
1216
1217
```

2. On line 73, add a #define with the name of your function, following the style on line 62 through 72 (#define NAMEOFOURFUNCTION)

```
71 #define NIGHT_NIGHT 9
72 #define CUSTOM_COLOR 10
73 #define NAMEOFOURFUNCTION 11
74
75
```

3. In the get_state() function on line 756, change one of the current state assignments to your custom state name. EG "state = BOUNCE" would become "state = NAMEOFOURFUNCTION"

```

761     if (xyz.z > 8000)
762     {
763         if (xyz.y > 0)
764         {
765             change_delay = 10;
766             state = NAMEOFOURFUNCTION;
767         }
768     else
769     {

```

4. In the switch statement between lines 1162 and 1194, add a new case just above the default case. The case will look like this:

```

1197         case NAMEOFOURFUNCTION:
1198             your_function_name();
1199             break;
1200         default:
1201             break;
1202     }
1203 } //end last_change_time
-----

```

Bill of Materials

Note that some of these parts are custom made, and can't be bought. Specifically the metal rings. Luckily those should never be damaged, so you won't need to replace them. However if you want to build a new lamp, you'll have to go to a metal shop and have them roll you two rings, 9" in diameter, and have them weld on 6 equally spaced 1/4" x 3/8" bolts to each ring. All the locally sourced parts can be bought at Ace hardware for sure. Most likely Lowes and Home Depot carry them as well.

The below parts are all required:

Adafruit Feather Huzzah (quantity 1) Get the one with "Loose headers"

<https://www.adafruit.com/products/2821>

NeoPixel strip 1m (60 LEDs per meter)

<https://www.adafruit.com/products/1461>

Accelerometer (quantity 1)

<https://www.adafruit.com/products/2809>

Power Supply (AC Adapter) (quantity 1)

https://www.amazon.com/gp/product/B00MRGKPH8/ref=oh_aui_detailpage_o00_s03?ie=UTF8&psc=1

Schottky Diodes. (Quantity 4).

<http://www.mouser.com/Search/ProductDetail.aspx?R=SB540E-Gvirtualkey59960000virtualkey750-SB540E-G>

Resettable Fuse. (Quantity 1).

<http://www.mouser.com/Search/ProductDetail.aspx?R=RGEF400virtualkey65000000virtualkey650-RGEF400>

Wires (length ~1 foot) Can be bought locally or online. 18 awg to 22 awg is sufficient.

1/4" nuts (quantity 16) (Locally sourced)

1/4" x 3/8" bolts (quantity 6) (Locally Sourced) for metal rings

1/4" x 1" bolts (quantity 4) (Locally Sourced) for carriage.

¼" wire terminal (quantity 6). (Locally Sourced)

3D printed carriage (quantity 1). File found in online repository

3D printed half-circle (quantity 2). File found in online repository