ESE 150 – Lab 11: Internet-of-Things (Servo and WiFi)

LAB 11

In this lab we will see how our computers can manipulate the physical world:

1. Communicate with our ItsyBitsy over WiFi.
2. Use our ItsyBitsy to control a servo.
3. Use the servo to actuate a light switch.

Background:

Concatenate (cat) – cat is the unix command-line tool for combining (concatenating) the output from several files and sending them somewhere. Often this somewhere is the terminal. So, one way to read a file on your computer screen is to simply cat the file. The output of cat can be sent to another unix program using the pipe construct that we saw in Lab 9.

Netcat (nc) – nc is a unix command-line tool that can send and receive data over the network. The "cat" part is by analogy with the cat routine above. Instead of sending data to the console, it can send its data over a network connection. As such, it takes arguments for the IP address of the destination host (when sending) and the port on the destination host. It can specify the protocol (e.g. UDP, TCP). It can also listen on a port, in which case it specifies the port that it should receive data from. Rather than reading from files, it typically takes its input from (and produces its output to) another program using the pipe construct.

Echo (echo) – is the unix command-line tool for printing an output to somewhere. That somewhere can be the terminal, but it can also be redirected (like any unix command) to somewhere else or used in a pipe to feed input into another unix command or program.

Prelab: Prepare background and setup

- Read background materials and make sure you're ready for lab.

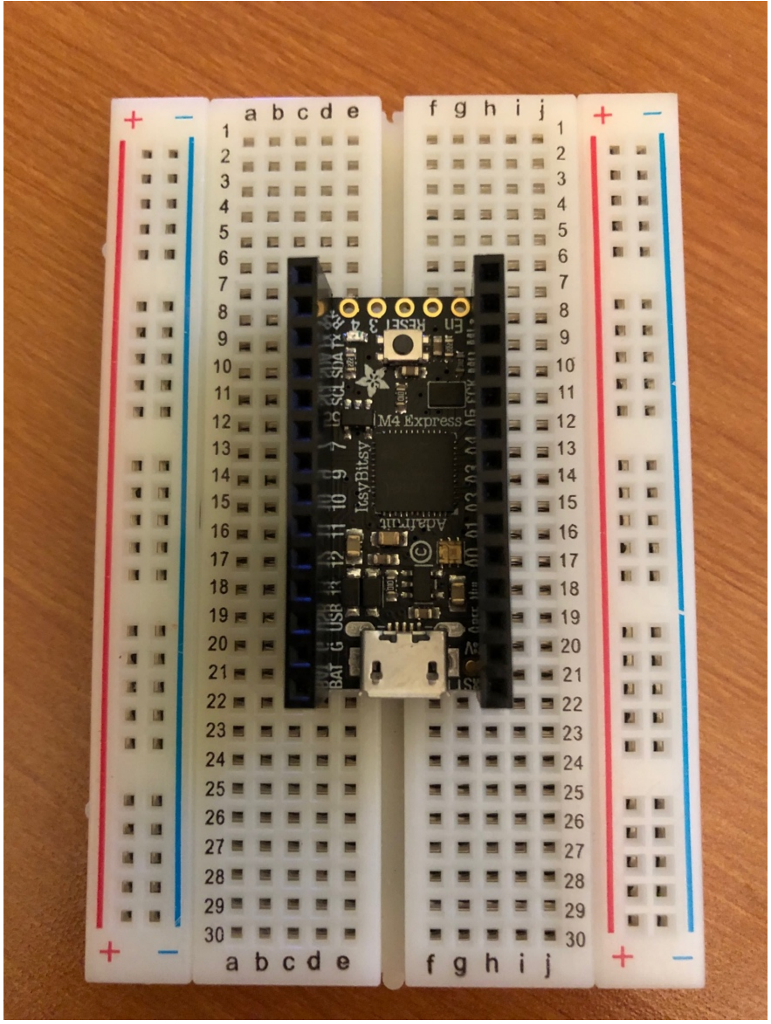
1. We assume you have already downloaded the Arduino IDE and configured it for the Itsy Bitsy. Instructions are in Lab 1, Prelab, Step 3.
2. Read through Lab.
3. Read how to control servos with the Arduino server library: https://learn.adafruit.com/using-servos-with-circuitpython/arduino
   a. What are the units for specifying position using the Arduino server library?
   b. What is maximum and minimum range for servo position?
4. Read about AirPennNet-Device: https://www.isc.upenn.edu/how-to/using-wireless-penn#For-devices
   a. How many devices can you register (total)?
   b. How many have you already registered (before this lab)?
   c. **We strongly recommend completing Section 1 of the lab before lab starts if you are operating from your dorm.** This is because there may be a15 minute delay between registering your device and the having the device recognized on AirPennNet-Device. You will require the 'AirPennNet-Device' network for connecting your ItsyBitsy.
5. Submit your answers to questions in 3 and 4 to the associated Prelab Quiz.

Lab Procedure:

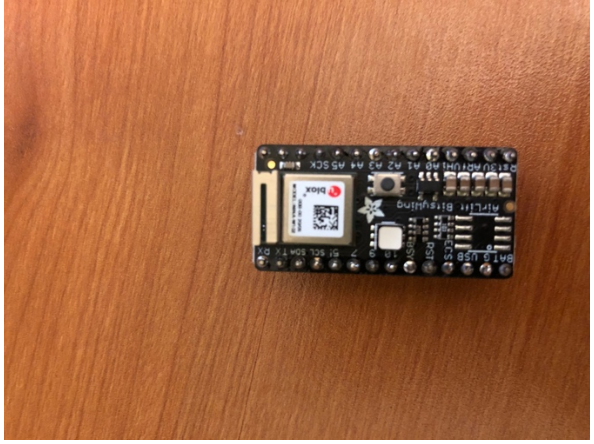
Lab – Section 1: Add AirLift Modem to ItsyBitsy

- In this section we'll setup the ItsyBisty for Wireless Communication.

1. Place the ItsyBitsy on the middle of the breadboard with the USB connection pointed down:
    a. Place the RX pin in 8d.

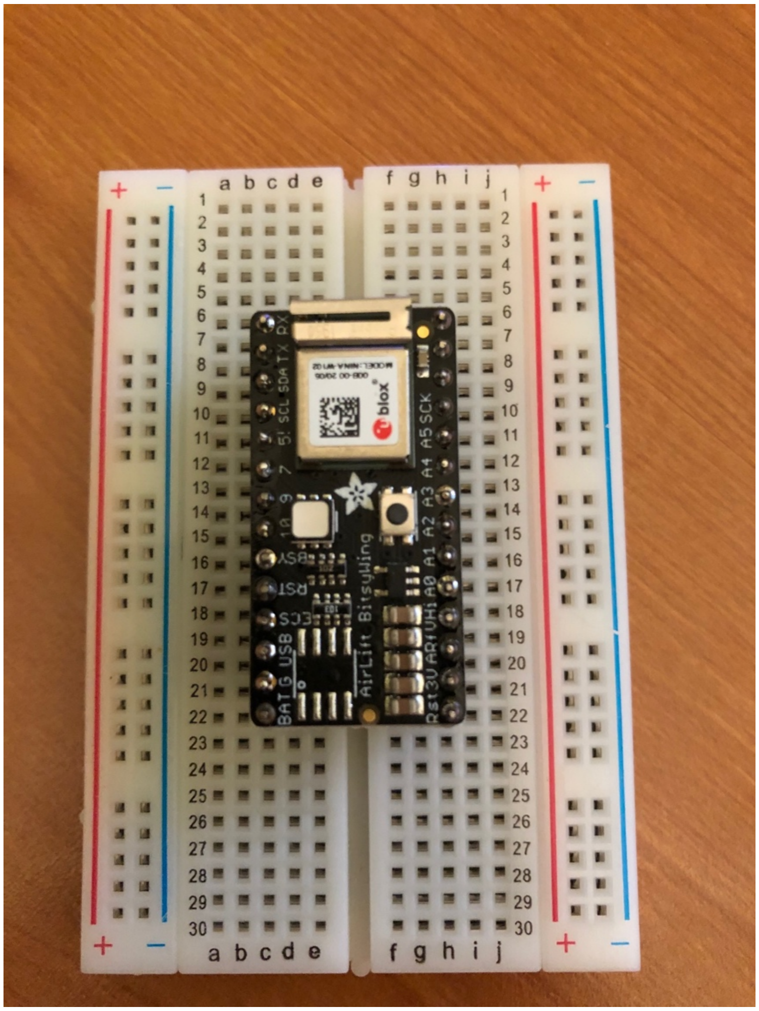


2. Find the AirLift BitsyWing



Note the antenna (shown on right in this diagram) is fragile. Use care not to put force on it when inserting and handling the AirLift.

3. Place the AirLift on top of the ItsyBitsy matching pins (e.g. BAT on top of BAT pin, RX on top of RX pin)

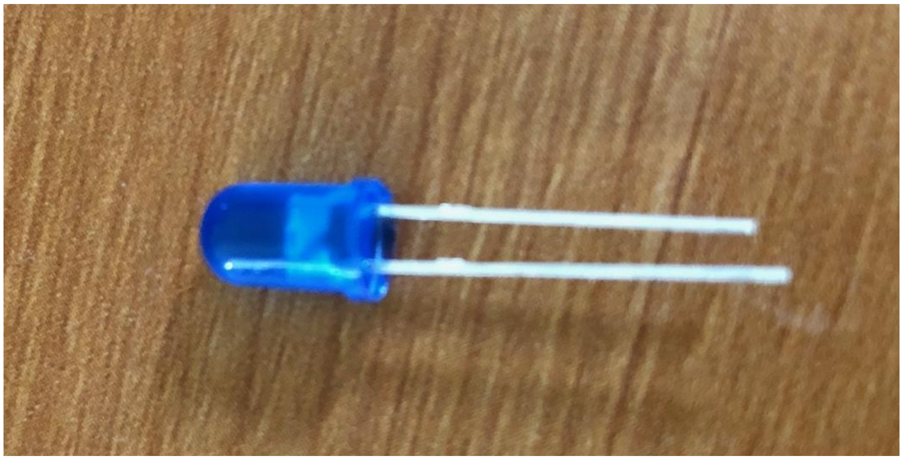

a b c d e
f g h i j
+
−
RX TX SDA SCL 5! 7 9 10 BSY RST ECS USB G BAT
SCK A5 A4 A3 A2 A1 A0 VHi AR 3V Rst
AirLift BitsyWing
blox

4. Plug in the USB cable to connect your ItsyBitsy to your computer.
5. Follow the instructions on https://learn.adafruit.com/adafruit-airlift-bitsy-add-on-esp32-wifi-co-processor/arduino through WiFi Connection Test and (if necessary for your home WiFi) Secure Connection Example:
    a. Library Install
    b. First Test
        i. For the #defines for pins, the one that matches the web page was under #elif !defined(SPIWIFI_SS)
        ii. Running the ScanNetworks you should see a list of WiFi networks
            1. Similar to what you see from your smartphone or laptop
            2. Record the MAC address for your AirLift that this prints out.
    c. If on Penn Campus
        i. Register your MAC address for AirPennNet-Device
    d. WiFi Connection Test
        i. At home, you will want to use the same network (SSID) as your laptop since you will be communicating with the ItsyBits/AirLift later in the lab.
        ii. On Penn Campus, you will want to use AirPennNet-Device and the password you identified during registration.

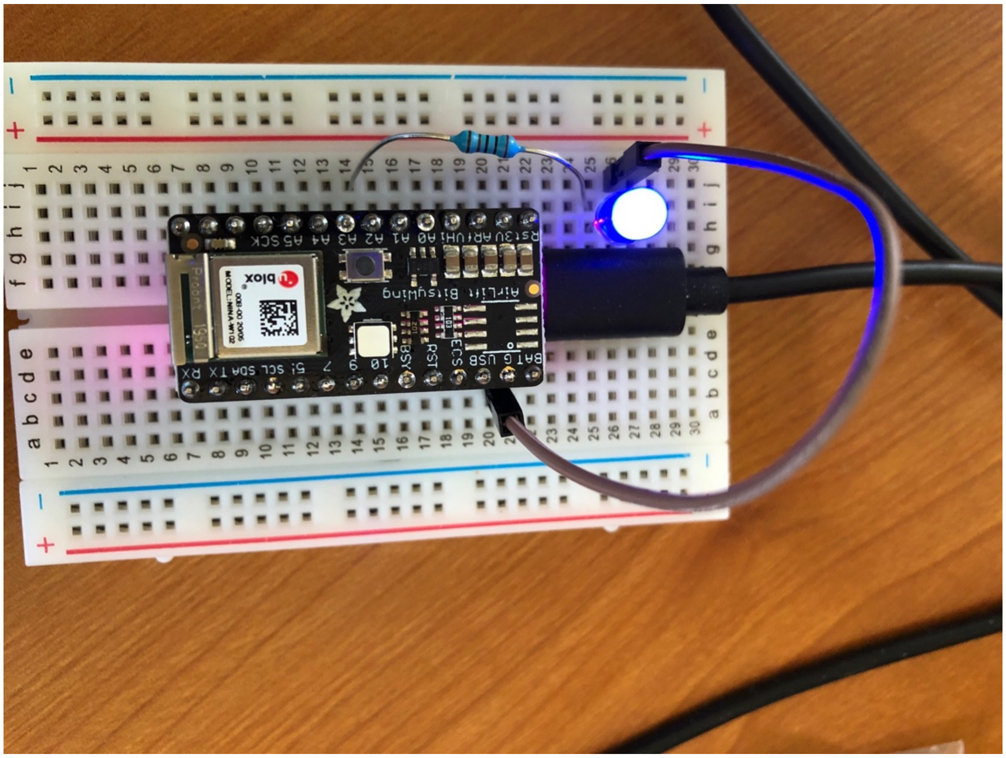
Lab – Section 2: Wifi Communication

- In this section, you'll start communication via WiFi with your ItsyBits.

1. Load the WiFiUdpSendReceiveString example:
    a. Files/Examples/WiFiNINA/WiFiUdpSendReceieveString
    b. Update arduino_secrets.h as you did for WiFiWebClient in Step 5.c in previous section
    c. Upload to ItsyBitsy
    d. Note the IP Address that the ItsyBitsy/AirLift acquired:
       Attempting to connect to SSID: SD2T3
       Connected to wifi
       SSID: SD2T3 (subject to change according to your network)
       IP Address: 192.168.1.30 (subject to change according to your network)
       signal strength (RSSI):-70 dBm
2. Use nc on your computer to send a string to the ItsyBitsy/AirLift
    a. Say Hello: `echo "Hello" | nc -u 192.168.1.30 2390`
        i. 192.168.1.30 is the IP Address my ItsyBitsy/AirLift acquired; yours will probably be different
        ii. The -u option to nc tells it to send UDP packets
        iii. 2390 is the port; look through the WiFiUdpSendReceiveString to see where that is set and used.
    b. On Windows, netcat (nc) may not be installed by default. If nc is not found, you can install it with: `pacman -S netcat`
    c. Note on the Arduino console that the "Hello" was received.
    d. On your computer, you should receive an "acknowledgement" back
    e. You may need to type Control-C to end the `nc` command after you get a response.
3. Add a discrete LED on pin A3.
    a. Select a discrete LED from your parts kit.



b. Wire a 2K Ω resistor from A3 to the positive (long) lead on the LED.



c. Wire the short lead to Ground (G).

d. You can test your LED wiring with a simple program like:

```
void setup() { } // nothing to do here
void loop() {
  digitalWrite(A3,LOW);
  delay(1000);
  digitalWrite(A3,HIGH);
  delay(1000);
}
```

While there is a built-in LED on the board on PIN 13, do **not** plan on using that. PIN 13 has a special function for the AirLift, so turning it to low will interfere with communications. So, we add this separate LED as an indicator.

4. Refine WiFiUdpSendReceiveString to have it turn on or off an LED based on the packet you send.

   a. When the received packet begins with the character ‘0’, have it turn the LED off.

   b. When the received packet begins with the character ‘1’, have it turn the LED on.

   c. This should take 2 lines of code.

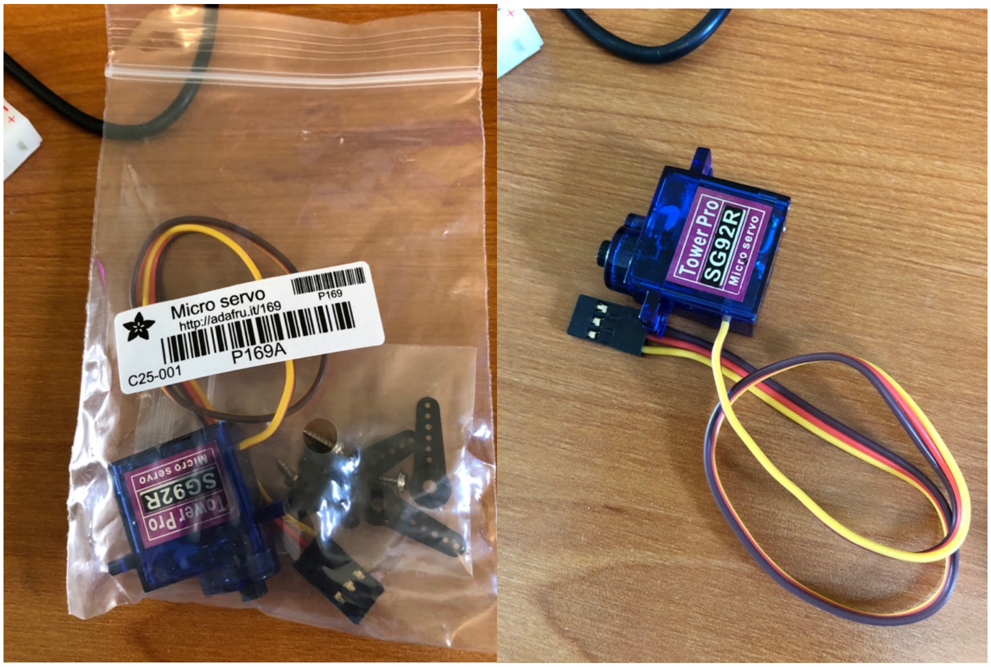
   d. Save this revised LED-Control program and include it with your lab writeup.

      i. Do not include your `arduino_secrets.h` in the writeup.

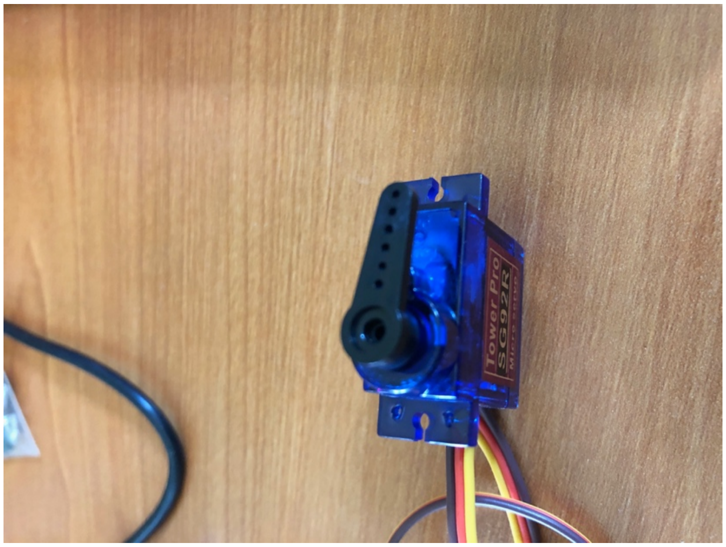
Lab – Section 3: Servo

- In this section you will setup your ItsyBitsy to control a Servo.

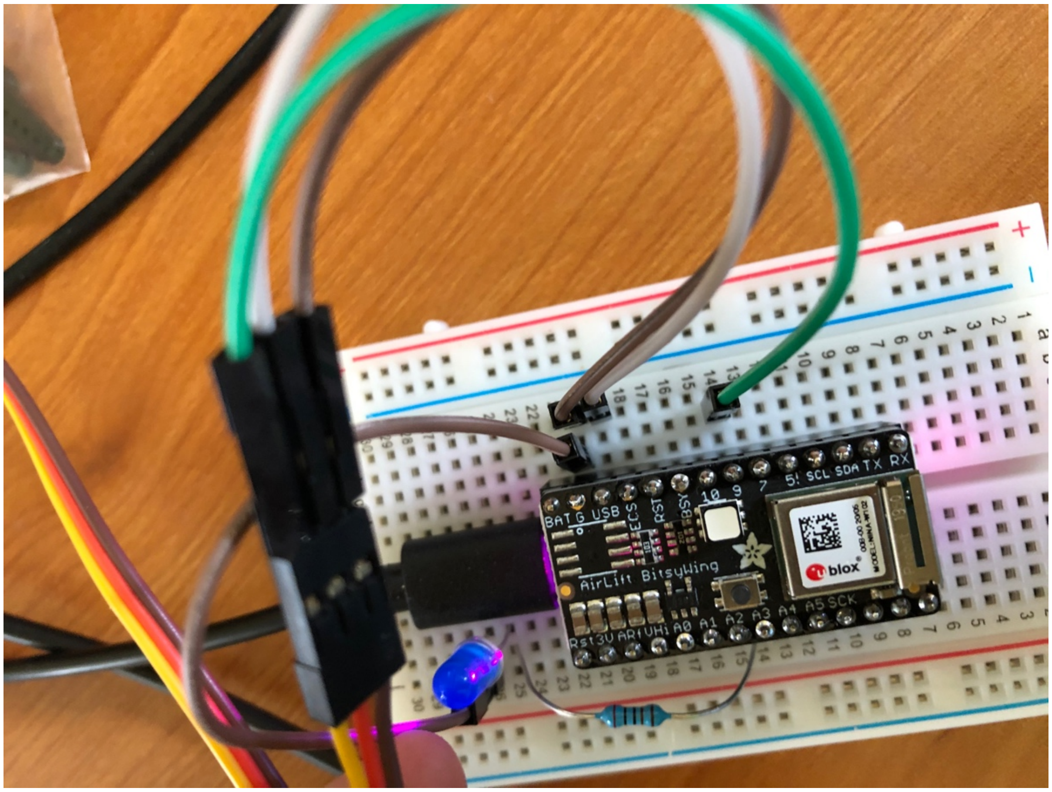
1. Locate Servo.



2. Press-fit the one-sided arm.



3. Wire to your Pin 9 on your ItsyBitsy
   a. Wire
      i. Yellow line to pin 9
      ii. Brown line to Ground (G)
      iii. Red line to 5V (USB)

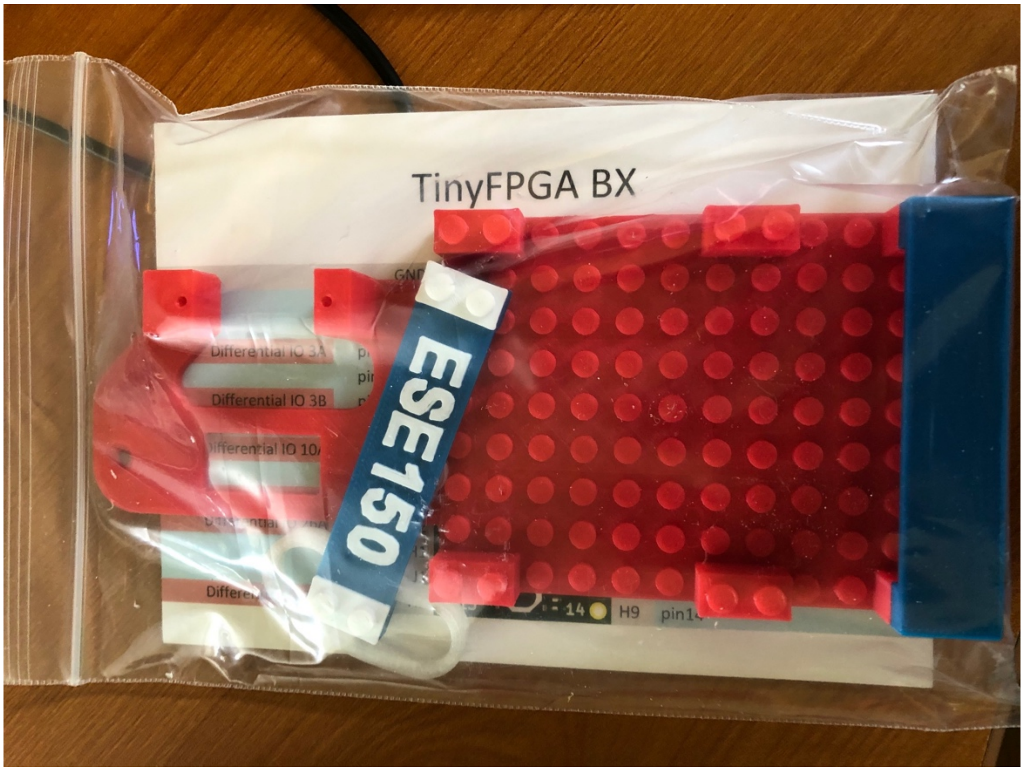


4. Program the ItsyBitsy to Control your Servo:
   a. Remember you read about using the Arduino Servo Library in the prelab.
   b. Program your servo to move between 0 degrees and 180 degrees 5 times with 10 seconds of delay between movements.
   c. For this section, this is purely an ItsyBitsy task; you will combine with WiFi in next section.
5. Include your servo control code in your writeup.

Lab – Section 4: Light Control

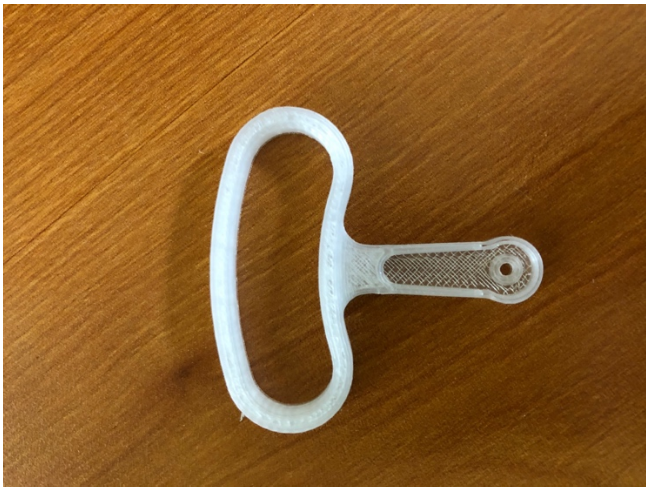
- In this section you will put together the WiFi and Servo Control to remotely control a light switch.

1. Integrate Servo Control with WiFiUdpSendReceiveString:
    a. In addition to turning on or off the LED also actuate the servo into the up or down position.
    b. Test and tune operation.
2. Install the Servo into the provided Switch Housing:
    a. Find printed housing parts.



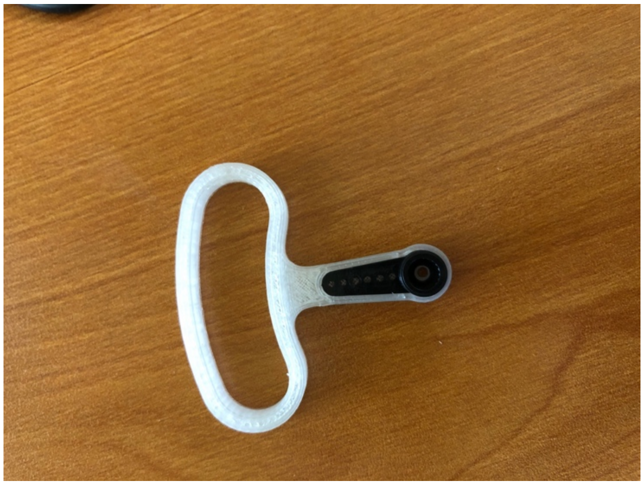
b. Add printed switch arm and secure with small screw

   i. Switch ARM

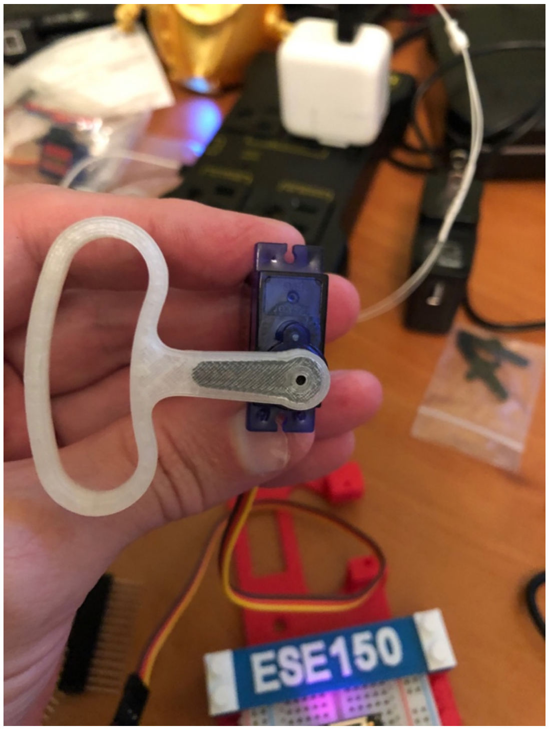
   ii.

   iii. Press fit into single-sided ARM from servo kit

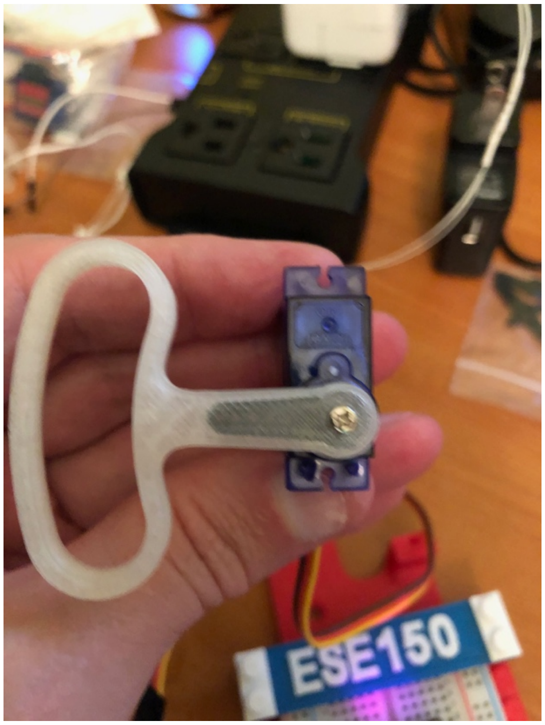
   

   iv. Test range of motion for servo. It may be necessary to reset how the ARM is coupled to the servo drive gear (where is arm at 0 degrees? 180 degrees?) to give full range of motion for switch arm.

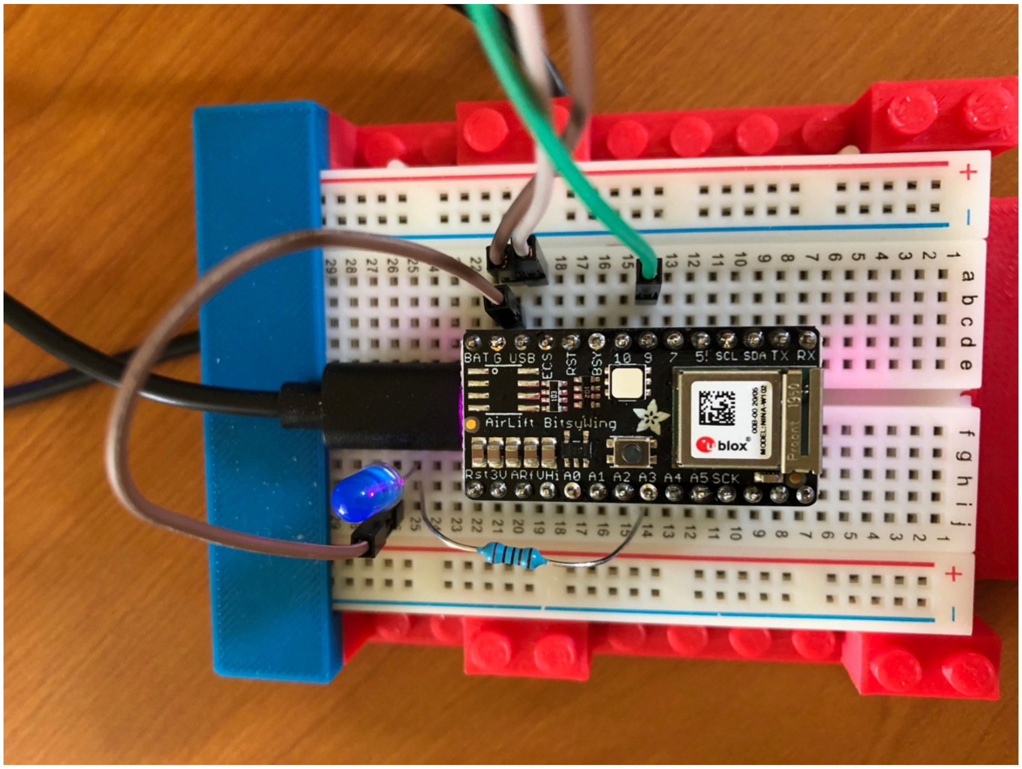
   v. Make sure the arm is pointed to the left when the servo gear is at the bottom.



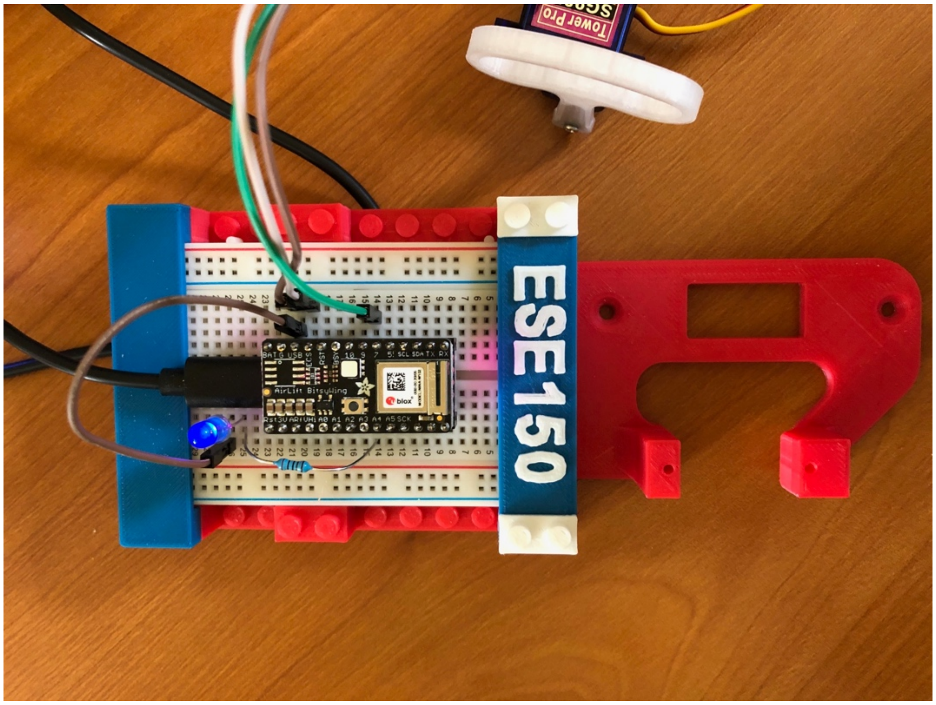
vi. Add screw to hold printed and servo arm onto serve drive gear.



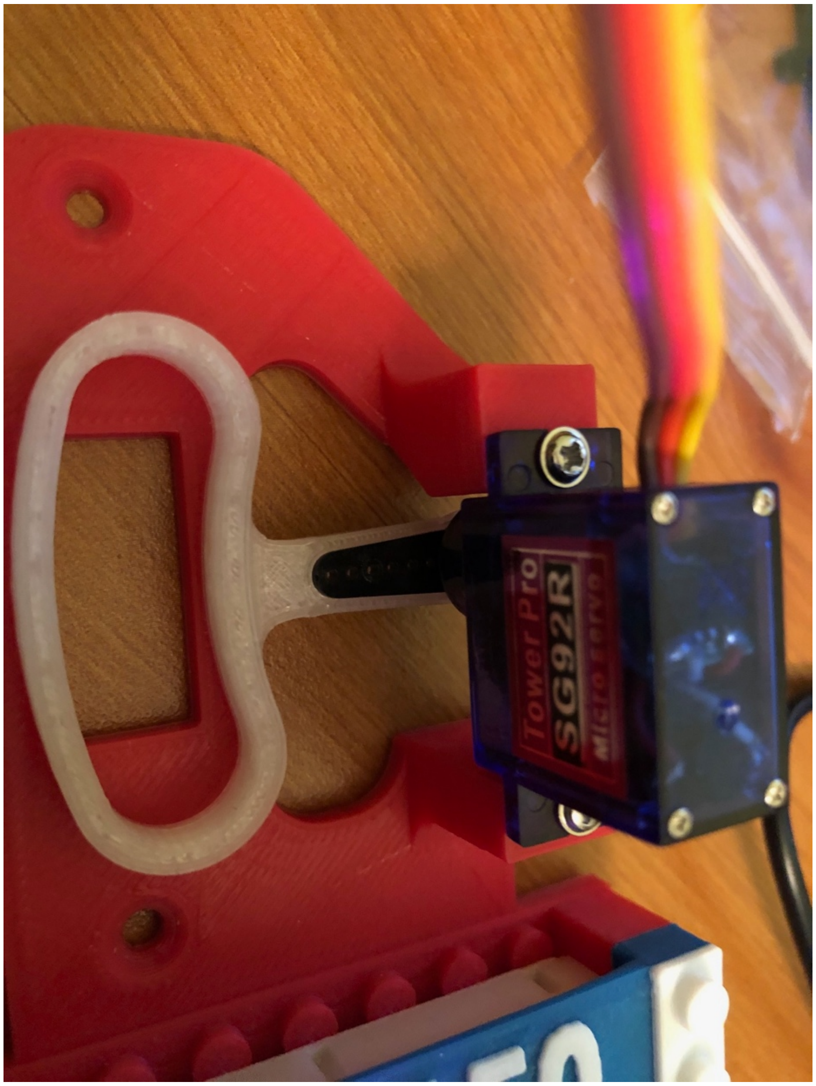
c. Place bread board in Switch Housing.



d. Use ESE150 retainer to hold top of bread board.



e. Mount servo in Switch Housing and secure with larger screws. The servo gear should be close to the top of the housing, not the bottom.

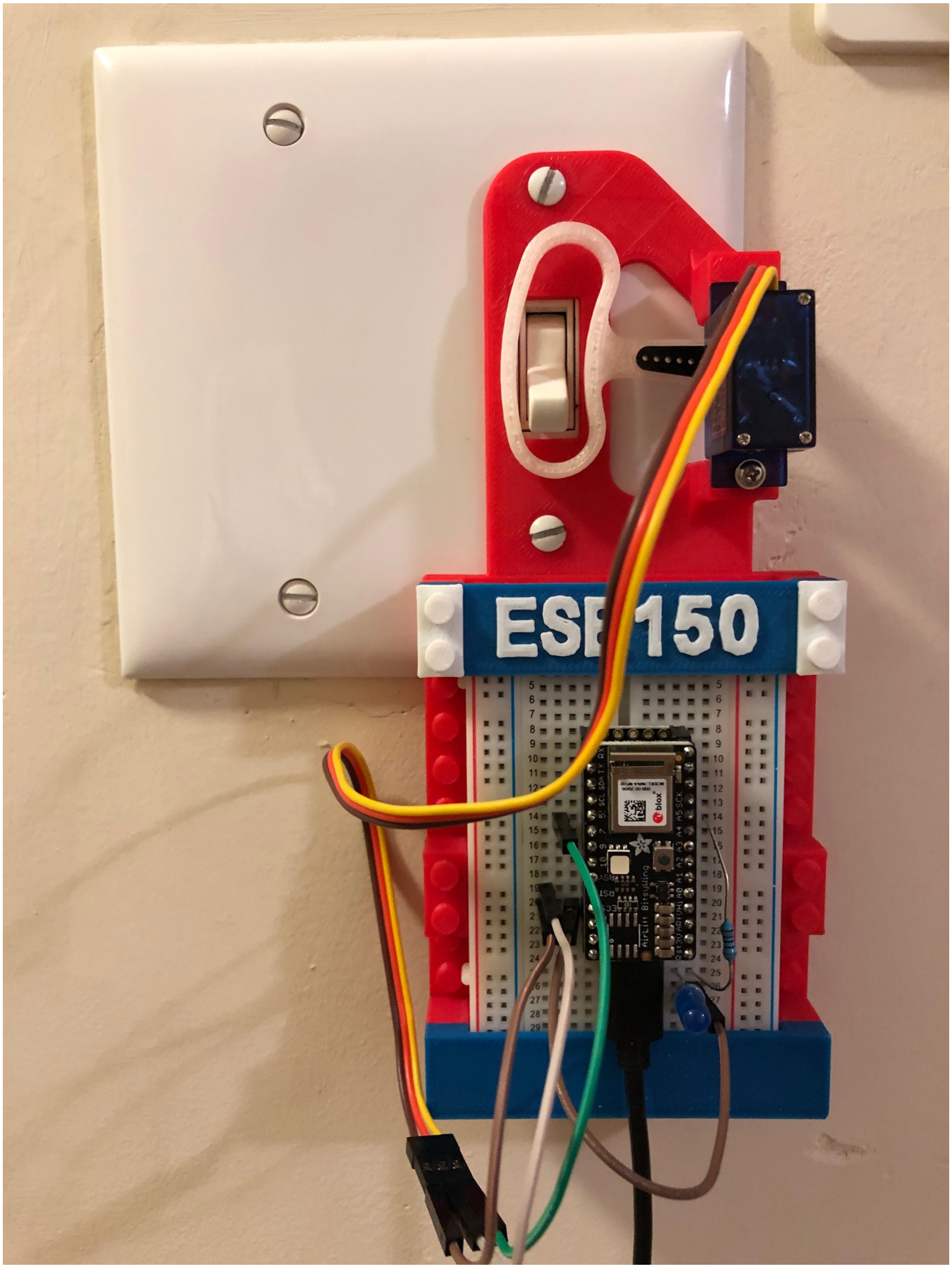


   i. When placed in the Switch Housing, mechanically, it won't have the full 180 degree range of motion.
   ii. Here and/or later, you will likely need to change the upper and lower limit range you program the servo to switch among.

3. Install the Light Control module in the Switch Housing on a Light Switch:
    a. Remove existing screws holding plate.

b. Supplied screws: Some of these screws are longer than typical wall plate screws in case you need them to accommodate the additional thickness of the Switch Housing.

c. Use existing screws or one of the longer supplied screws to hold our Switch Housing plate on top of existing plate (screw goes through both new plate and original).



4. Tune your code and arm placement as necessary
    a. You want to make sure the arm sweeps far enough up to move the light switch into the ON position and far enough down to switch it into the OFF position.
    b. As noted above, the range of motion will likely need to be restricted short of 0 to 180 degrees.
    c. You may want to home the position to something in the middle after switching ON or OFF so that the servo isn't continually being driven past the allowed mechanical range.
5. Demonstrate that you can remotely control your light using your computer.
    a. Show your TA for your exit checkoff.
6. Take a picture of your light setup
    a. Post in a private message on piazza.
        i. Let us know if we have permission to use for ESE publicity (not required, but appreciated).
    b. Include with report.
7. Include your final light control code in your report.
8. (optional) You can operate your ItsyBitsy connected only to a USB Power plug (including the one provided in your kit) rather than to your computer. That will allow you to separate your computer from the light.
    a. You will need to change your Arduino code to not wait to see a live Serial Port
        i. Comment out or remove the loop that begins: `while (!Serial)`
        ii. It may be useful to turn the LED on when the server starts so that you will know when the ItsyBitsy is ready to receive commands now that you no longer have serial output telling you that it has started.
    b. Use the USB Plug.

9. (optional) You can communicate to your light control module with any device that can send appropriate WiFi packets.
    a. If you want to control the device using Android, we recommend using this app.
        i. https://play.google.com/store/apps/details?id=com.jca.udpsendreceive&hl=en&gl=US
        ii. Make sure to specify the correct IP address and port number. This app only does UDP, so you don't need to specify which protocol to use.
    b. If you want to control the device using iOS, we recommend this app.
        i. https://apps.apple.com/us/app/tcp-udp-client/id906830843
        ii. Make sure to choose the UDP protocol, correctly specify the IP address and use port 2390. Put your encoded message in the message field and send!
10. Keep your Light Control module for use with next week's lab.

Postlab

1. Your ItsyBitsy/AirLift unit could control multiple devices.
   1.1. Assuming one pin per device, about how many devices can you control?
   1.2. How would you revise the encoding in the packet in your WiFiUdpSendSendReceiveString to support this multiple-device control?
2. How would you revise the encoding in the packet in your WiFiUdpSendSendReceiveString if you wanted to specify the servo position rather than ON/OFF?
3. If you add a button and used an ItsyBitsy pin as input, sketch how you could use a second ItsyBitsy/AirLift pair as an ON/OFF switch for your Light Controller?
4. Anyone who knows your AirLift IP address and port (and how you format your packet) could control your light switch (or someone who mistook your IP address for theirs). How could you add some level of security to prevent unintended access?

HOW TO TURN IN THE LAB

- Each student should assemble an individual writeup and upload as a PDF document to canvas, containing:
  - Answers to pre-lab questions
  - Arduino code
    - Do not include your `arduino_secrets.h` in the lab report.
  - Pictures of final light installation
  - Answers to post-lab questions