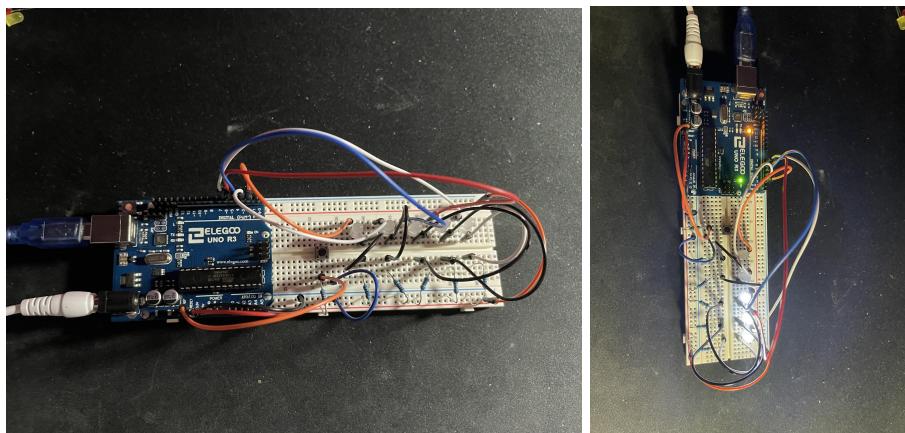


Etude 1: Perceptron-P

part one

I followed the reference image in the document to build the circuit, but I did not really color code the wires. I realized the usefulness of it afterward as I was analyzing my board to make sure all the wires are in the correct hole. After placing every component, I tested to see if the lights effectively work and they did. I noticed the button, but I was somewhat afraid to interact with it since I wasn't sure what it does or if it could break the build, so I did not press it. I proceeded to plug the device into my computer and uploaded the code that was provided for the assignment. I noticed it flashing, so I presumed it worked and plugged the device elsewhere and started attempting to take pictures with my phone while waving it around.

set up + set up lit up



various shots



part two

At first glance, I noticed that the resistors in the built circuit are replaced with wires in the alternate circuit. The LED lights are now linked to each other as their cathode pins (-) are connected with a wire. The second build is a series circuit as the components are connected end-to-end in a line which forms a single path. Instead of having every current go through a single resistor (like in the alternate circuit), the built circuit has each branch pass through its own individual resistor. The resistors in the first build are parallel to each other. And so, the built circuit, the first option, is a parallel circuit.

The model we built is more reliable since it should still function after a broken part (eg. one of the bulbs dying). The LED lights are independent in a way that if one branch stops functioning, the rest of the setup still works. For the second build, when an LED light stops working, all the other lights won't be lighting up since the current is no longer able to flow.

Also, the components in a parallel circuit share the same voltage, making that LED lights light up with the same intensity. The series circuit share the power and may not distribute the strength equally, meaning that the LED lights may look dimmer as the power is unequally being shared to all the components on the board.

part three

When I pressed the button, the current seemed to have stopped momentarily and lit up again after a few seconds. Until it is released, holding the button keeps the LED lights off. Following that observation, I figured that the button, when pressed, somehow prevents the circuit from traveling through the parallel paths set up. Since the current won't circulate in that area, the lights do not receive power and cannot light up until the button is released.

I assume that interacting with the button changes the path of the circulation. As electricity wants to go to ground (negative) as quickly and effortlessly as possible, it will take the path with the least resistance so it won't lose too much power. Pressing the button creates a new path. Since the new electric circuit created is a series circuit that has less resistance than the parallel setup, the current will go through the temporary path instead.

custom message

After some retakes, I learned that some special characters, such as the colon and semicolon (; :) didn't work and the board did not light up after uploading the code which contains "unreadable" characters. I also noticed that capitalized letters do not show up. Looking through the code again made me realize that the code automatically capitalizes letters and the special characters I tried to use were not listed (in the displayChar function).

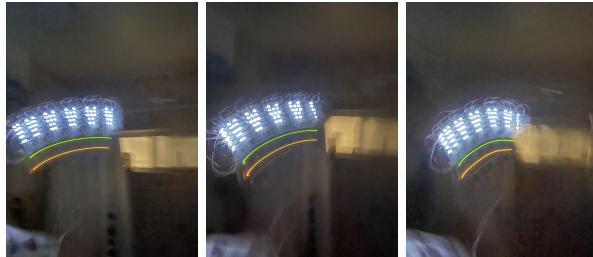
Arduino code

```
PERCEPTRON_P_ETUDE_ONE
{
    if (c == 'v') {for (int i = 0; i < 5; i++) {displayLine(v[i]); delay(delayTime); displayLine(0);}}
    if (c == 'w') {for (int i = 0; i < 5; i++) {displayLine(w[i]); delay(delayTime); displayLine(0);}}
    if (c == 'x') {for (int i = 0; i < 5; i++) {displayLine(x[i]); delay(delayTime); displayLine(0);}}
    if (c == 'y') {for (int i = 0; i < 5; i++) {displayLine(y[i]); delay(delayTime); displayLine(0);}}
    if (c == 'z') {for (int i = 0; i < 5; i++) {displayLine(z[i]); delay(delayTime); displayLine(0);}}
    if (c == '!') {for (int i = 0; i < 5; i++) {displayLine(excl[i]); delay(delayTime); displayLine(0);}}
    if (c == '?') {for (int i = 0; i < 5; i++) {displayLine(ques[i]); delay(delayTime); displayLine(0);}}
    if (c == ',') {for (int i = 0; i < 5; i++) {displayLine(cos[i]); delay(delayTime); displayLine(0);}}
    if (c == '.') {for (int i = 0; i < 5; i++) {displayLine(space[i]); delay(delayTime); displayLine(0);}}
    delay(charBreak);
}

void displayString(char* s)
{
    for (int i = 0; i <= strlen(s); i++)
    {
        displayChar(s[i]);
    }
}

void loop()
{
    displayString("b o o p");
}
```

Initial Result



"Ow" but the capitalized O did not appear

Final Result



attempt to illustrate the word "boop"