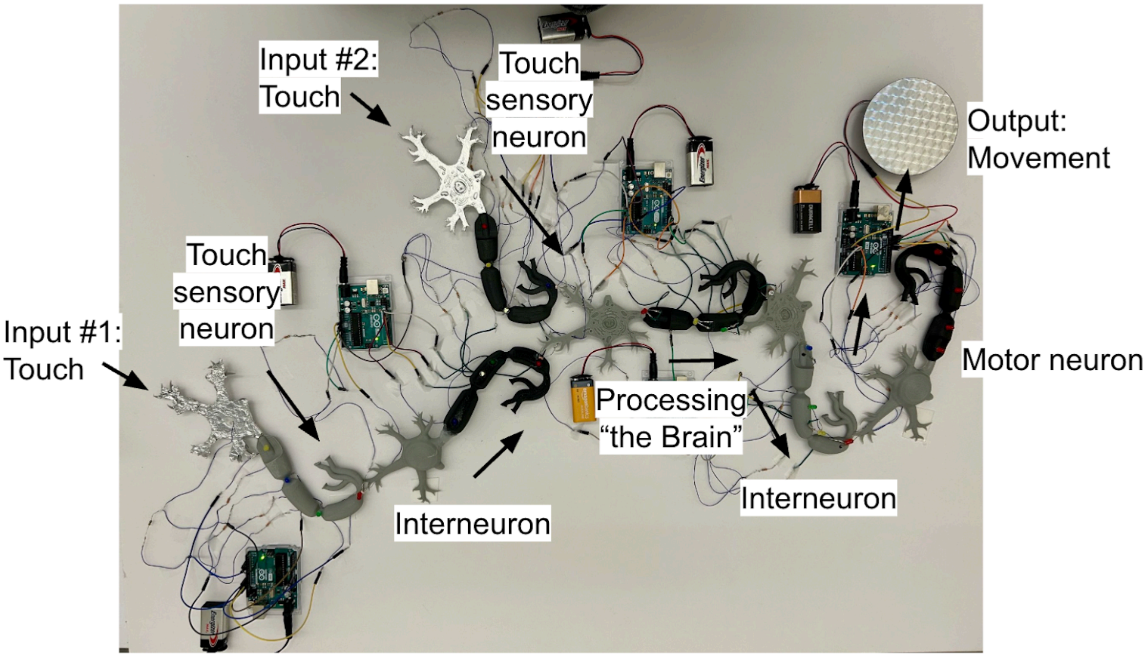
Project Title: Sensory Circuit Simulation

Overview

Project Summary: In this project, students will design and build an electronic circuit that mimics the touch sensory system, exploring how neurons communicate through electrical and chemical signaling. Using 3D-printed neuron models, students will practice soldering and transfer their designs from breadboards to sculptural circuitry, blending biology, engineering, and art to deepen their understanding of sensory pathways to deepen their understanding of sensory pathways.



Student Understandings

Learning Prerequisites:

Basic Breadboard Circuitry

- Understanding how to connect components on a breadboard
- Familiarity with series and parallel circuits
- Ability to read simple circuit diagrams

Introductory Arduino Skills

- Uploading code to an Arduino
- Using digital and analog pins for input and output
- Working with basic sensors and actuators (e.g., buttons, LEDs, buzzers)

Neuronal Function

- Structure and function of neurons (dendrites, axon, synapse)
- Electrical and chemical signaling in neural communication

Sensory System Overview

- How touch stimuli are received and transmitted to the brain
- The role of sensory neurons, interneurons, and motor neurons

Core Neuroscience Dogma

- Understanding the input → processing → output framework
- Ability to map this framework onto both biological systems and electronic models

Learning Objectives:

Students will know:

- The basic structure and function of neurons, including how they transmit signals through electrical impulses and chemical synapses.
- How the human touch sensory system works, from sensory receptors to neural pathways.
- The components of a simple electronic circuit and how they can be used to model biological systems.

Students will be able to:

- Design and build an electronic circuit that simulates the behavior of a touch-sensitive neural pathway.
- Solder components and transfer a prototype from breadboard to a sculptural, 3D-printed neuron model.

- Explain how sensory information is processed and transmitted in the nervous system using both biological and electronic analogies.
- Collaborate across disciplines to merge scientific knowledge with creative fabrication techniques.

Materials

Material	Description	Quantity
Neuron Model by SpaceKnight	3D model of a neuron downloaded from Thingiverse; used to build sculptural circuitry	6
Arduino Uno	Microcontroller board used to program and control the circuit	6
Adafruit MPR121	Capacitive touch sensor breakout board for detecting touch input	2
Photoresistor (LDR)	Light-sensitive resistor used to simulate sensory input	4
10kΩ resistor	Resistors used to limit current or form voltage dividers in the circuit	4
DC motor	Small motor used to simulate neural output (e.g., movement or feedback)	1
LEDs	Light-emitting diodes for visual output or signal indication	24
220Ω resistor	Resistors used to protect LEDs and limit current	25
NPN transistor (2N2222)	Transistor used to control power to components like motors or LEDs	1
Diode (1N4007)	Used to protect components from back current when using motors	1
Breadboard Jumper Wires	Male-to-male wires for connecting components on a breadboard	many
30 AWG wire	Tin-plated copper wire; solid core, used for permanent connections in sculpture	1 spool
Breadboards	Solderless boards used to prototype electronic circuits	6

Soldering kit	Includes soldering iron, solder, and tools for creating permanent connections	1
Drill	Used to drill holes in 3D neuron models for component placement	1

Teacher Preparation

1. Print 3D Neuron Models

- Download the 3D neuron model from Thingiverse (Neuron Model by SpaceKnight).
- Print 6 models, one per group
- *Optional:* Pre-drill pilot holes in key locations (axon terminals, dendrites) for easier component mounting.

2. Organize Materials by Neuron Type

Create labeled kits for each group with the following suggested components:

Neuron Type	**Key Components**
Sensory Neuron	MPR121 touch sensor, 4x LEDs, jumper wires, 5x 220Ω resistors
Interneuron	Photoresistor, 4x LEDs, jumper wires, 10kΩ resistor, 5x 220Ω resistors,
Processing Neuron	2x Photoresistors, 4x LEDs, jumper wires, 10kΩ resistor, 5x 220Ω resistors
Motor Neuron	Photoresistor, 5x 220Ω resistors, DC motor, diode, transistor, 4x LEDs, jumper wires

3. Prepare Workstations

- Set up soldering station(s) with irons, solder, and safety equipment.
- Provide each group with a breadboard, Arduino Uno, and a power source (USB or battery pack).
- Lay out shared bins with resistors, jumper wires, 30 AWG wire, and tools (wire strippers, pliers, etc.).
- Have printed circuit diagrams and step-by-step breadboard guides ready for each neuron type.

—-----------------------------------

A. Introduction

1. Start by asking: *“Think of a time you touched something hot—what happened, and how fast did you react?”* Let a few students share their experiences.

2. Review the central dogma of neuroscience and definitions:

Briefly explain each step:

Input = Sensory information (external: touch, sound; internal: body temp, hunger)

Processing = Brain or spinal cord analyzing information

Output = Response (movement, speech, change in heart rate, etc.)

3. Introduce three neuron types with quick examples:

Neuron Type	**Function**	**Example**
Sensory Neuron	Detects stimuli and sends signals to the brain/spinal cord	Touch receptor in skin
Interneuron	Connects neurons within the brain or spinal cord; processes information	Reflex pathways in the spinal cord
Motor Neuron	Sends signals from brain/spinal cord to muscles or glands	Telling your hand to pull away

4. Class Simulation to model a neural circuit. Act out a simplified sensory pathway using students as neurons:

i) **Sensory Neurons (2 students):**
Each receives a different input stimulus—for example, one reacts to a touch, the

other to light. These students pass a physical object (e.g., a small ball or light-up prop) as a signal toward the next step.

ii) **Processing Neuron (1 student):**
This student represents the brain. They receive input from both sensory neurons and decide how to respond. For example, if both inputs are received, they may trigger a more urgent response; if only one is active, a different or no output might occur.

You can ask this student to “think aloud” (e.g., “Both inputs are active—send a strong signal!”) to model processing.

iii) **Interneuron (1 student):**
This student simply passes the signal from the processing neuron to the motor neuron—highlighting that interneurons can act as relays, but they aren't necessarily "thinking" themselves.

iv) **Motor Neuron (1 student):**
Receives the output signal and carries out a physical response (like jumping, clapping, or shouting “ouch!”).

5. Wrap-up and transition to the project. Explain to the students that they’ll be building electronic circuits to simulate neurons. This will help them understand how circuits can model the sensory processing pathways in the brain.

B. Neuron Model Procedure

1. Prototype on Breadboard

- Use the provided circuit diagrams to build your assigned neuron type (e.g., sensory, interneuron, motor neuron) on a breadboard.
- Test your circuit thoroughly to ensure all inputs and outputs function as expected (e.g., LED lights up when touched).
- Ask for support if your circuit isn’t responding—debug before moving to the next step.

2. Prepare the 3D Neuron Model

- Identify where your components (LEDs, wires, etc.) will be embedded on the model.
- Drill or carefully punch holes in your 3D neuron model to fit your LEDs or allow wire pass-through.

3. Transfer Circuit to Neuron Model

- Use 30 AWG wire to begin transferring your circuit from the breadboard to your neuron sculpture.

- Solder all connections securely and test each component as you go.
- Route wires neatly along the model to highlight the neuron's structure (axon, dendrites, etc.).

4. Final Testing & Debugging

- Once soldering is complete, connect power and test your neuron model.
- Check for loose connections, unlit LEDs, or unexpected behavior.

5. Connect and Communicate Between Neurons

- Link neuron models using jumper wires or breadboard connectors to simulate signal transmission.
- Start with the provided pathway:
  Touch Input #1 → Interneuron → Processing Neuron → Interneuron → Motor Neuron
  Touch Input #2 → Processing Neuron → Interneuron → Motor Neuron

6. Explore and Modify

- Once the basic communication chain is working, experiment!
  - Try adding delay, changing outputs (LED to motor), or rerouting connections.
  - Ask: How does adding or removing a neuron affect the circuit's behavior?

C. Student Assessment

1. After students have completed the design, testing, and integration of their neuron circuits, assign a written and visual explanation task. Each group will write a detailed write-up and submit a circuit diagram that documents their project.

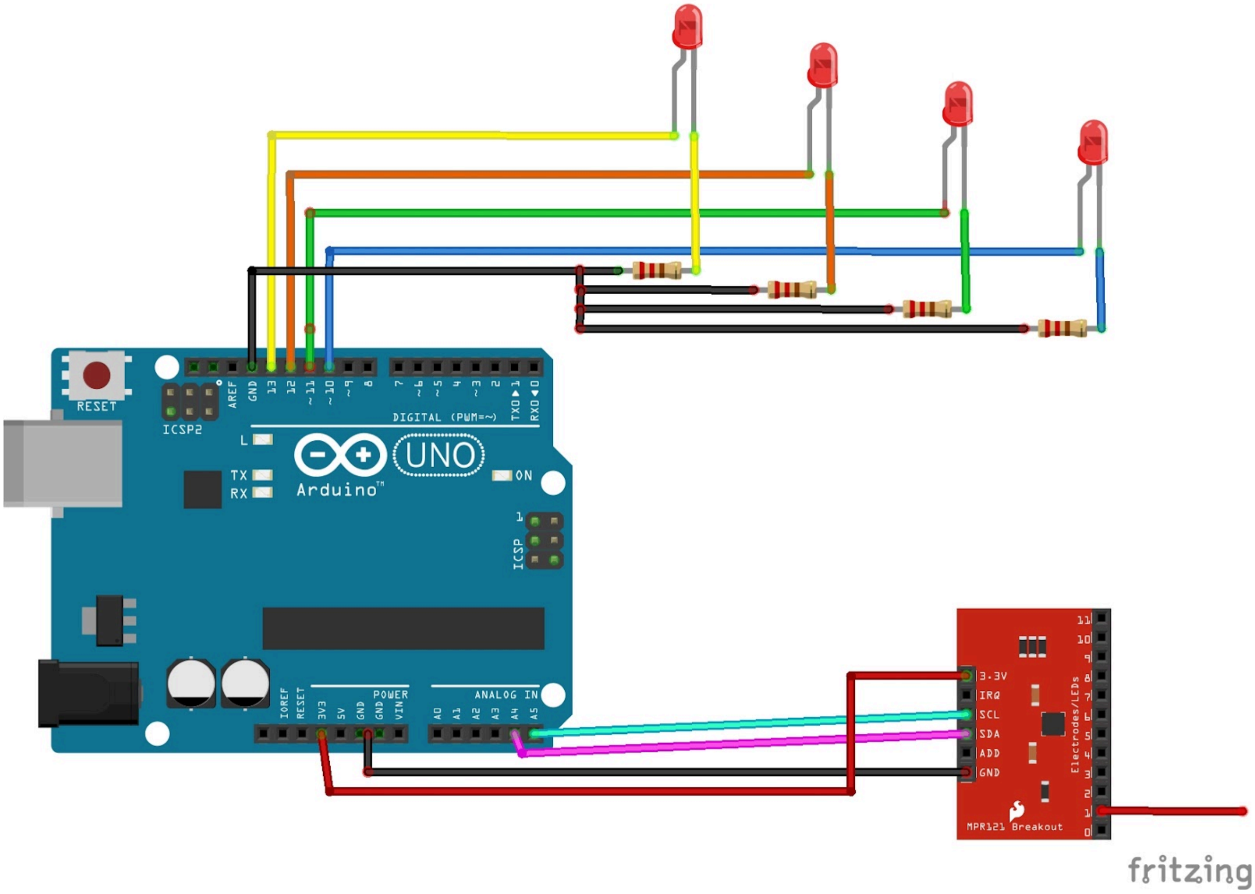
   Students should include the following in their write-up:

   - Identification of the neuron type they modeled (sensory neuron, processing neuron, interneuron, or motor neuron).
   - A description of the components used (e.g., sensors, LEDs, resistors, Arduino, 30 AWG wire) and the function of each within the circuit.
   - An explanation of how the circuit simulates the behavior of the selected neuron type—specifically how it detects, processes, or transmits a signal.
   - A brief reflection on any technical or conceptual challenges faced during the prototyping or soldering process, and how they addressed them.
   - A description of how their neuron connects to and functions within the class-wide neural network, emphasizing communication between neurons.

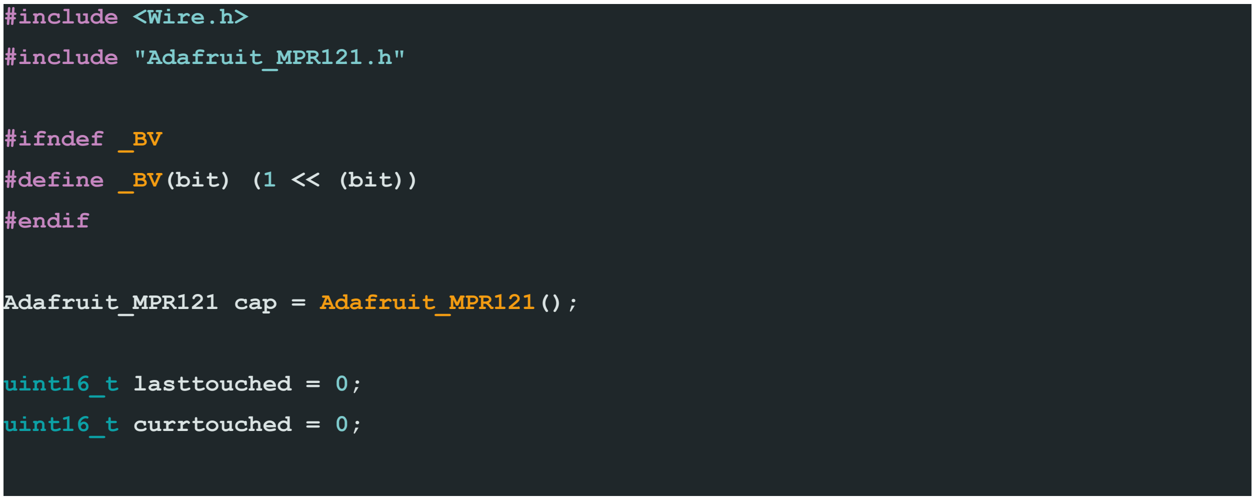
Encourage clear, concise writing using accurate scientific and technical vocabulary. Use this reflection as both a content check and a communication exercise. You may choose to assess based on clarity, completeness, circuit accuracy, and integration of biological concepts.

Input: Touch

Circuit Diagram:



Code:

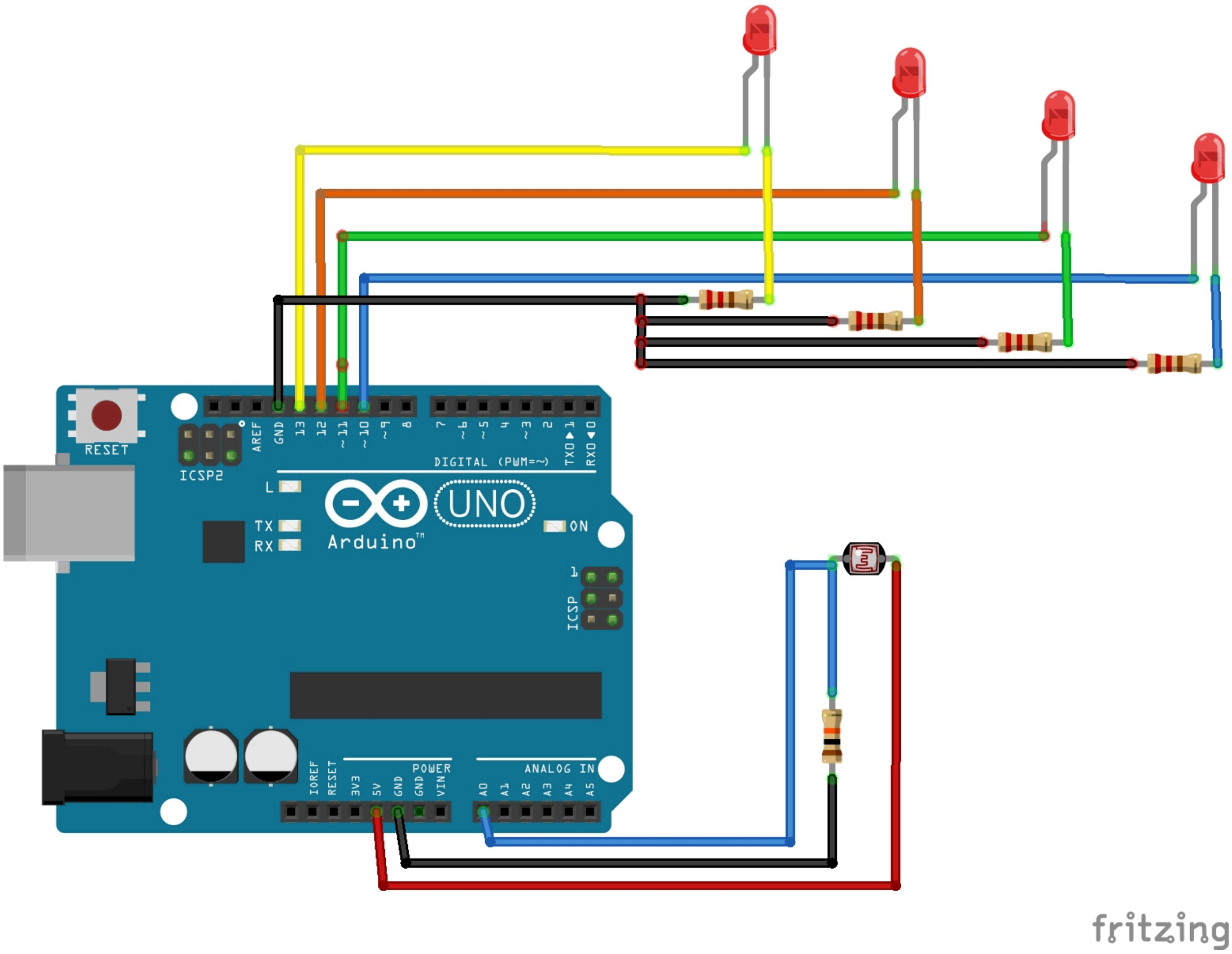


```
#include <Wire.h>
#include "Adafruit_MPR121.h"

#ifndef _BV
#define _BV(bit) (1 << (bit))
#endif

Adafruit_MPR121 cap = Adafruit_MPR121();

uint16_t lasttouched = 0;
uint16_t currtouched = 0;
```

```
// LED pins
const int ledPins[] = {10, 11, 12, 13};

void setup() {
 Serial.begin(9600);

 while (!Serial) {
   delay(10);
 }
  Serial.println("Adafruit MPR121 Capacitive Touch sensor test");
  if (!cap.begin(0x5A)) {
   Serial.println("MPR121 not found, check wiring?");
   while (1);
 }
 Serial.println("MPR121 found!");

 // Initialize LED pins
 for (int i = 0; i < 4; i++) {
   pinMode(ledPins[i], OUTPUT);
   digitalWrite(ledPins[i], LOW);
 }
}

void loop() {
 currtouched = cap.touched();

 for (uint8_t i = 0; i < 12; i++) {
   if ((currtouched & _BV(i)) && !(lasttouched & _BV(i))) {
     Serial.print("Pad ");
     Serial.print(i);
     Serial.println(" touched");
   }
   if (!(currtouched & _BV(i)) && (lasttouched & _BV(i))) {
     Serial.print("Pad ");
     Serial.print(i);
     Serial.println(" released");
   }
 }

 // If touch just started
 if (currtouched != 0 && lasttouched == 0) {
   Serial.println("Touch detected - starting LED sequence");
   for (int i = 0; i < 4; i++) {
     Serial.print("Turning on LED ");
     Serial.println(i);
     digitalWrite(ledPins[i], HIGH);
     delay(1000);
     digitalWrite(ledPins[i], LOW);
   }

   Serial.println("LED sequence complete");
 }

 lasttouched = currtouched;

 delay(50); // debounce and avoid flooding Serial output
}
```

Interneuron

Circuit Diagram:

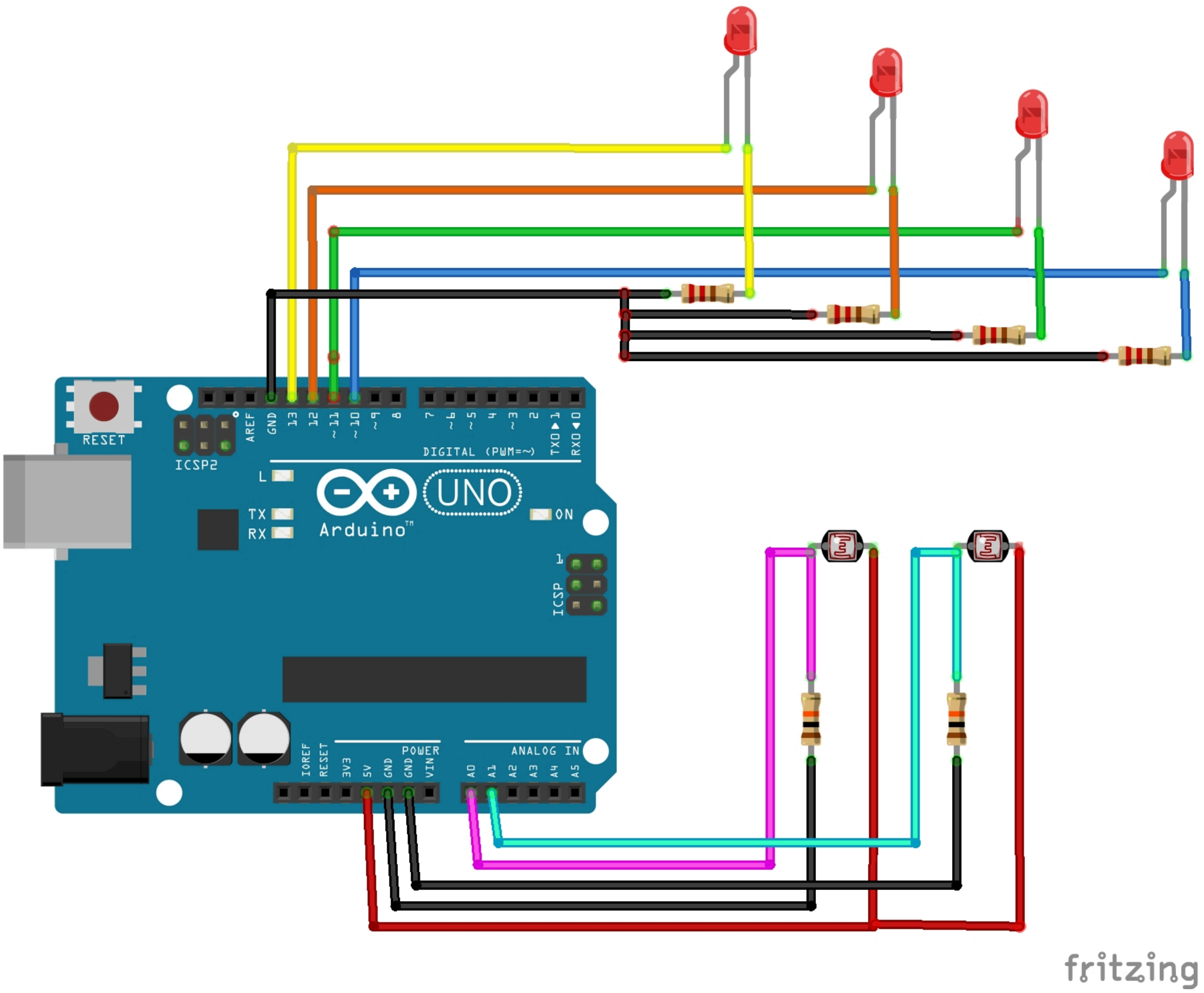


Code:

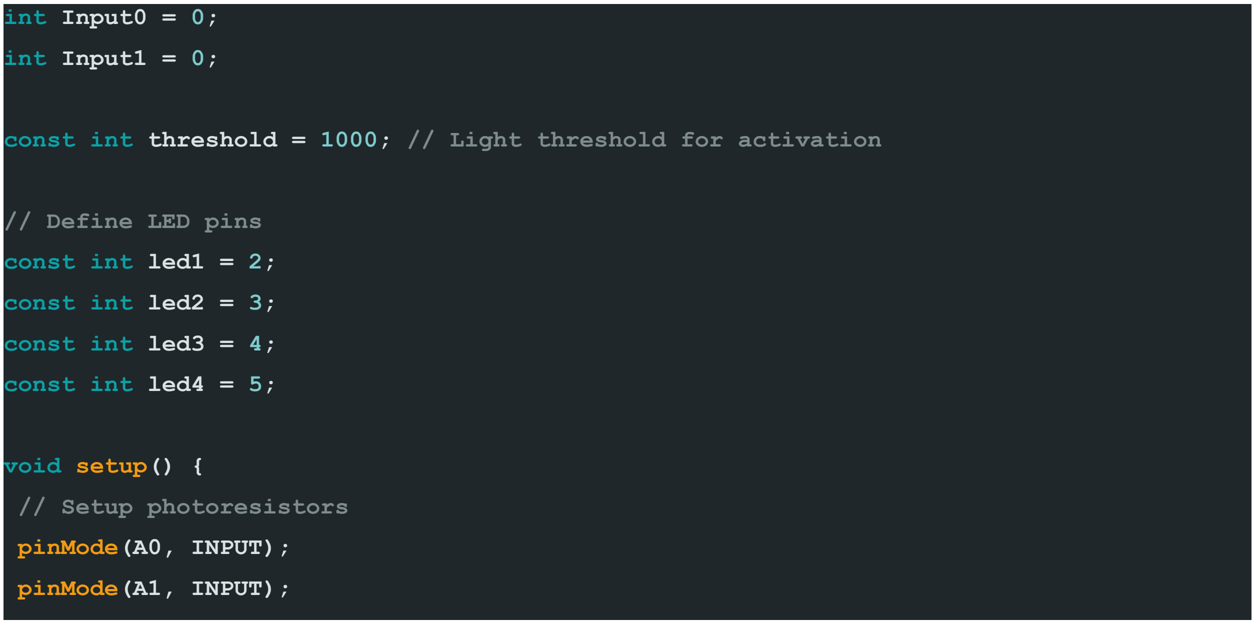
```
int animationSpeed = 400;    // Speed for LED blinking
int sensorValue = 0;         // Value read from the photoresistor
const int sensorPin = A0;    // Analog pin connected to the photoresistor
const int ledPins[] = {13, 12, 11, 10}; // LED pins in sequence
const int controlPin = 9;    // Output pin for analog control (optional)

void setup() {
 // Set LED pins as outputs
 for (int i = 0; i < 4; i++) {
   pinMode(ledPins[i], OUTPUT);
 }
  pinMode(sensorPin, INPUT);      // Set the photoresistor pin as input
 pinMode(controlPin, OUTPUT);    // Set the control pin as output
 Serial.begin(9600);             // Initialize Serial Monitor
}

void loop() {
 // Read the value from the photoresistor
 sensorValue = analogRead(sensorPin);

 // Print the sensor value to the Serial Monitor
 Serial.print("Light Sensor Value: ");
 Serial.println(sensorValue);

 // Map sensorValue to control brightness (optional)
 analogWrite(controlPin, map(sensorValue, 0, 1023, 0, 255));

 // If sufficient light is detected, start LED animation
 if (sensorValue > 250) { // Adjust threshold as needed
   Serial.println("Light detected! Starting LED animation..."); // Indicate light
detection
   for (int i = 0; i < 4; i++) {
     digitalWrite(ledPins[i], HIGH); // Turn on the LED
     delay(animationSpeed);          // Wait for the animation speed
     digitalWrite(ledPins[i], LOW);  // Turn off the LED
     delay(animationSpeed);          // Wait before the next LED
   }
 } else {
   // If light is below the threshold, turn off all LEDs
   Serial.println("Light level too low, LEDs are off."); // Indicate low light
   for (int i = 0; i < 4; i++) {
     digitalWrite(ledPins[i], LOW);
   }
 }

 delay(500); // Delay for stability and readable Serial Monitor output
}
```

Processing Neuron

Circuit Diagram:



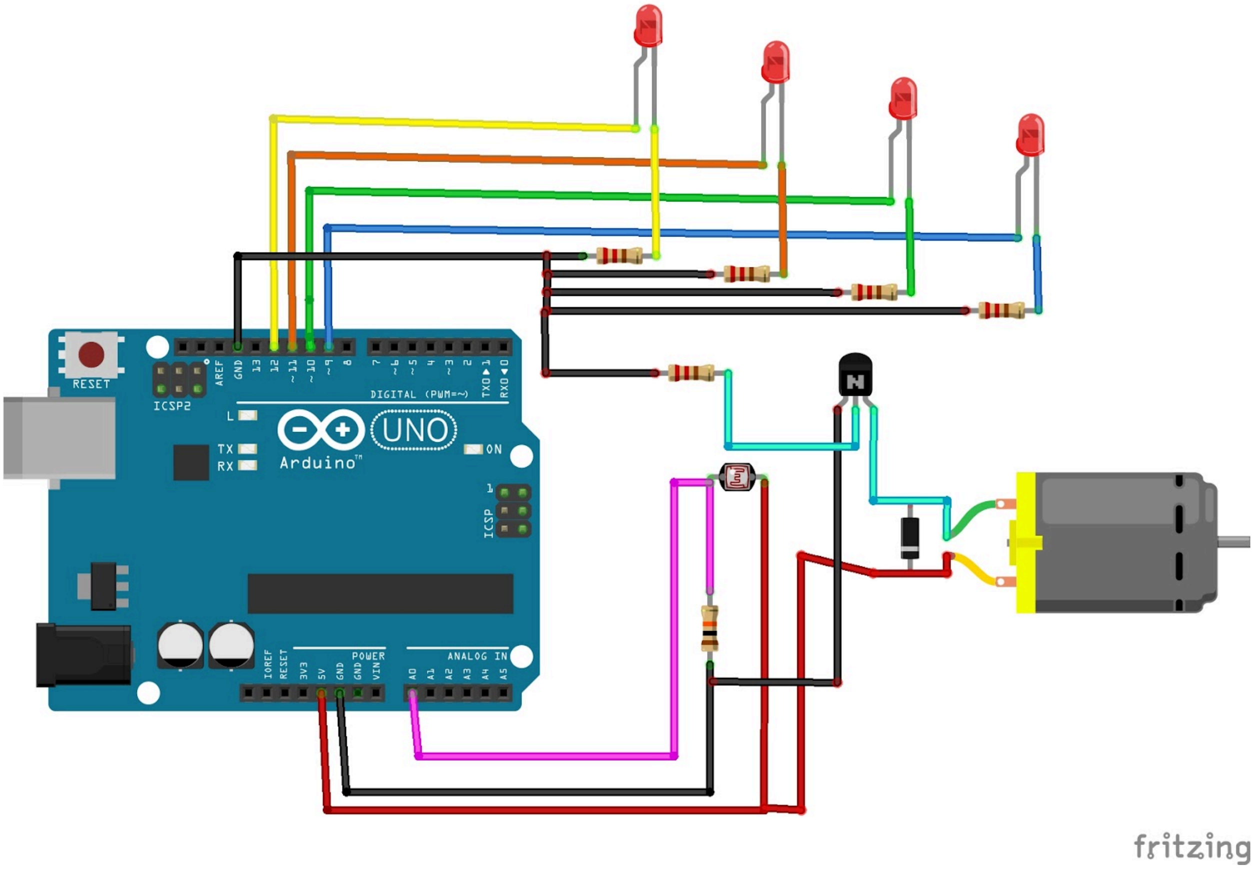
Code:



```
int Input0 = 0;
int Input1 = 0;

const int threshold = 1000; // Light threshold for activation

// Define LED pins
const int led1 = 2;
const int led2 = 3;
const int led3 = 4;
const int led4 = 5;

void setup() {
 // Setup photoresistors
 pinMode(A0, INPUT);
 pinMode(A1, INPUT);
 // Setup LEDs as outputs
 pinMode(led1, OUTPUT);
 pinMode(led2, OUTPUT);
 pinMode(led3, OUTPUT);
 pinMode(led4, OUTPUT);

 // Ensure LEDs are initially off
 digitalWrite(led1, LOW);
 digitalWrite(led2, LOW);
 digitalWrite(led3, LOW);
 digitalWrite(led4, LOW);

 // Start Serial Monitor
 Serial.begin(9600);
}

void loop() {
 // Read photoresistor values
 Input0 = analogRead(A0);
 Input1 = analogRead(A1);

 // Print sensor values for debugging
 Serial.print("Photoresistor 1: ");
 Serial.print(Input0);
 Serial.print(" | Photoresistor 2: ");
 Serial.println(Input1);

 // Check if either sensor detects light above the threshold
 if (Input0 > threshold || Input1 > threshold) {
   digitalWrite(led1, HIGH);
   delay(500);
   digitalWrite(led2, HIGH);
   delay(500);
   digitalWrite(led3, HIGH);
   delay(500);
   digitalWrite(led4, HIGH);
   delay(5000); // Keep LEDs on for 5 seconds
 }
 else {
   // Turn off all LEDs if neither sensor detects light
   digitalWrite(led1, LOW);
   digitalWrite(led2, LOW);
   digitalWrite(led3, LOW);
   digitalWrite(led4, LOW);
 }
}
```

Motor Neuron

Circuit Diagram:



Code:

```


int sensorState = 0;


void setup() {
// put your setup code here, to run once:
pinMode(13, OUTPUT);
pinMode(12, OUTPUT);
pinMode(11, OUTPUT);
pinMode(10, OUTPUT);
pinMode(9, OUTPUT);
pinMode (A0, INPUT);


Serial.begin(9800);
}

void loop()
{
sensorState = analogRead(A0);
Serial.print("Sensor State: ");
Serial.println(sensorState);


;if(sensorState>300 && sensorState<1023)
{
digitalWrite(9, HIGH);
delay(500);
digitalWrite(10, HIGH);
delay(500);
digitalWrite(11, HIGH);
delay(500);
digitalWrite(12, HIGH);
delay(500);
digitalWrite(9, LOW);
delay(100);
digitalWrite(10, LOW);
delay(100);
digitalWrite(11, LOW);
delay(100);
digitalWrite(12, LOW);
delay(100);
digitalWrite(13, HIGH);
delay(10000);
digitalWrite(13, LOW);
}
else
{
}
}
```

References

MPR121
https://learn.adafruit.com/adafruit-mpr121-12-key-capacitive-touch-sensor-breakout-tutorial/overview

Neuron Model https://www.thingiverse.com/thing:3170054

Photoresistor (Analog Input)
https://www.tinkercad.com/learn/overview/OXQL7IEL26F9H0U?type=circuits

Multiple LEDs & Breadboards
https://www.tinkercad.com/learn/overview/O94BCSIL26F9EV3?type=circuits

Project 13: Touchy-Feely Lamp
https://www.tinkercad.com/learn/overview/ONY3E9VISCC2VBZ?type=circuits

Digital Input/Analog Inputhttps://www.tinkercad.com/learn/overview/OH411VKJH0TNG79?type=circuits