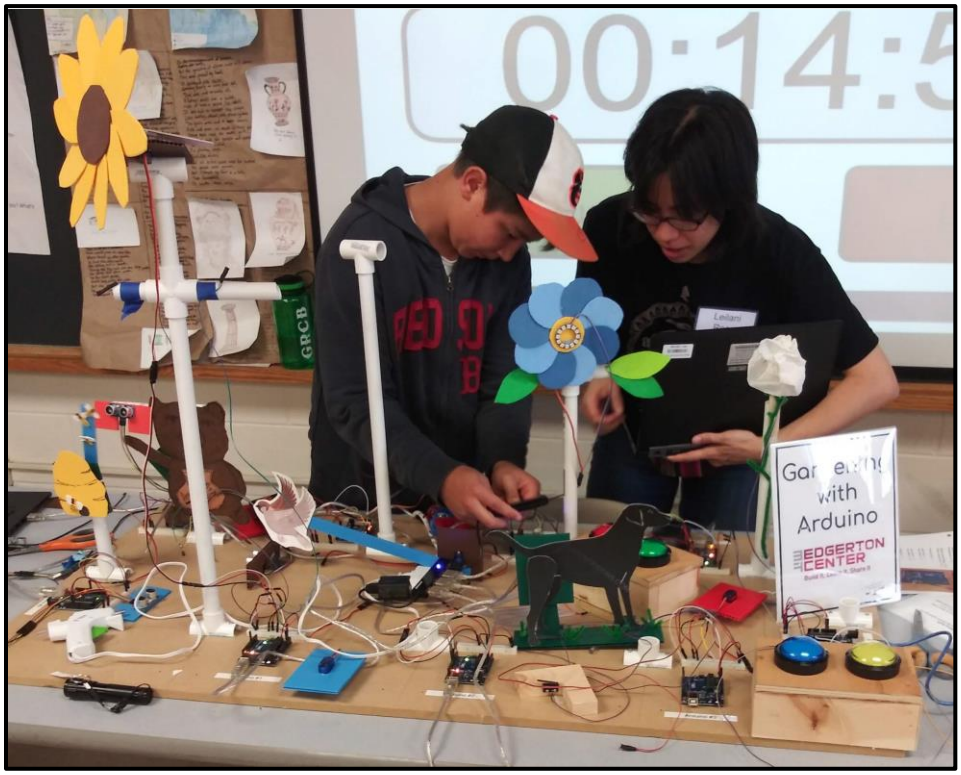
EDGERTON CENTER
Massachusetts Institute of Technology
77 Massachusetts Avenue, 4-408, Cambridge, MA 02139
Web: edgerton.mit.edu

Arduino Garden

A Do-It-Yourself Guide to Promote STEM Skills and Awareness



Summary
Introduce Physical Computing with the Arduino Garden! Use electronic devices to sense the environment and create physical actions. Students brainstorm, build, and code characters and objects they might find in a garden - buzzing bees, smiling flowers, gnomes – get creative! This activity with grades 6 and Although this activity requires considerable fabrication in advance, it can be used over and over again. Suitable for grades 6 and up.

Authors: Diane Brancazio, dianeb@mit.edu
Contributors: Leilani Roser, lroser@mit.edu

Date: February 5, 2020
Revision Number: 1

Why We Use Creative Commons Licensing:

Our mission at the Edgerton Center is to uphold the legacy of Harold "Doc" Edgerton by providing the necessary resources, encouragement, and support for students to gain an education in engineering both in the conventional classroom and particularly outside of it. We promote an overarching ethos of building, learning, and sharing.

As a non-profit entity committed to "building, learning, and sharing" resources for students, we want to make our materials available for free whenever possible, and encourage others to share. Publishing our materials under a **Creative Commons Attribution-NonCommercial-ShareAlike 4.0 International License** allows us to do this by requiring users and contributors to redistribute at no charge, and by helping others to trace our materials back to their original source**.**
More information: http://creativecommons.org/licenses/by-nc-sa/4.0/

You are free to use, modify, and share our materials under these terms:

BY: Give proper **Attribution** - this requires that when you use or share our materials, you provide the following: the name of the creator (MIT Edgerton Center) and other parties that collaborated or provided funding for the work; a copyright notice; this license notice; a disclaimer notice; a link to the material; and indications of changes that were made. You may do so in any reasonable manner, but not in any way that suggests the licensor endorses you or your use.

NC: You may only use the material for **NonCommercial** purposes.

SA: You may remix, transform, or build upon the material - and if you do, you must **ShareAlike** - distribute your contributions under the same license as the original. When sharing or adapting, there are to be **no additional restrictions** — you may not apply legal terms or technological measures that legally restrict others from doing anything the original license permits.

In your advertising materials:
You may say that your program uses curriculum developed by the MIT Edgerton Center, and where appropriate, provide a link to our home page: http://edgerton.mit.edu/.
You may not describe your program as a partnership or collaboration with MIT or the MIT Edgerton Center.

Help others to find us by citing the MIT Edgerton Center website, where a variety of curriculum materials are available for download: http://edgerton.mit.edu/k-12

Contents:

1. Teachers Guide
    a. Summary of project
    b. Learning Goals
    c. Time and estimate
    d. Materials
    e. How to use Starter Code

Sections 2 → 6 can be printed and used as a guide for students

2. Garduino Challenge Overview
3. Computer and Wiring Guide
4. Code Library for components (Sensors and Actuators)
5. Code Syntax and Structure
6. Garduino Examples with Code
    a. Sunflower
    b. Runaway Bird
    c. Colored Flower
    d. Buzzy Bees
    e. Dog Lifting Leg
    f. Friendly Critter

7. Build Guide: Garden base, Sensors, and Actuators
    a. Garden Base
    b. Pushbutton Switch
    c. Lever Switch
    d. Photoresistor with pull-down resistor
    e. Neopixel Ring
    f. Servo Motor and Mount
    g. Ultrasonic Distance Sensor and Mount

Teachers Guide

Summary of project

This project is designed to be an introduction to the Arduino platform, physical computing, and coding.
Physical computing is the use of a computer to get input from the physical environment, process that information using computer logic, then control physical actuators based on the logic.

The computer used is the Arduino Uno which is programmed in C. Starter code is provided so that students can focus on the essential concepts of physical computing rather than coding syntax.

There are Sensor and Actuator components available. Sensors get information from the environment. Actuators "do" something. In this activity students create an animated character/object with 1 or 2 sensors and 1 or 2 actuators.

Students may optionally create a manually controlled passive character to interact with the animated character, or craft auxiliary pieces to embellish the Garden scene.

Learning Goals

Students will be introduced to physical computing. They will use the Arduino IDE, (Interactive Development Environment), connect electronic devices, and debug their projects.

Time estimate

This activity can be completed in 1 hour. Add more time to refine the projects.

Materials

Overall structure: Plywood with PVC fittings and PVC pipe

Craft Materials

Foam Sheets
Cardstock
Pushpins
Pipe cleaners
Eyes-googly
Markers
Craft sticks
Hot glue guns
Corrugated plastic sheets
PVC plastic angle strips
PVC pipe and fittings
Scissors, Utility snips,
Wire - solid core, (~22 gauge)

Electronics

Computers and equipment on Garden board

(10) Arduino Uno, power bus and USB A- B cable
(10) USB power sources for all Arduinos

Sensors (quantities)

(4) Push button - colored, Set of 2
(6) Lever switch, on wood block
(10) Distance sensor assembly
(12) Light sensor assembly

Actuators

(10) 12-neopixel ring (addressable LEDs)
(10) Servo motors

How to use Starter Code

1. Install the Arduino IDE from https://www.arduino.cc/en/Main/Software
2. Install these libraries NewPing.h and Adafruit_NeoPixel.h
    a. Follow instructions at https://www.arduino.cc/en/Guide/Libraries

3. Download the Arduino Garden starter code in a zip file at: https://www.dropbox.com/s/vyc8sopnkf18uw3/GardenCodeStarter%2Bsamples.zip?dl=0

4. Extract the files and put in your Arduino folder.

5. Create new code Using GardenCodeStarter of code.
    a. Make sure to rename the files using Save As from within the Arduino IDE
    b. We use **tabs** to make the coding easier, that is, the main tab only has the logic, all the other functions and setting are in other tabs and do not need to be touched except by an advanced programmer. For more info on tabs
        i. see this video https://www.youtube.com/watch?v=dSAb3LZBHp0
        ii. or this documentation https://www.arduino.cc/en/guide/environment#toc8

6. Use the coding guidelines and library provided here to create your own code.

“Garduino” Challenge

CHALLENGE: Use code, Arduino boards, lights, motors, and sensors to plant something special in our computer controlled garden!

What even IS Arduino?

Arduino boards are programmable circuit boards that can use computer logic to control electronic parts. It’s a very popular device for physical computing.

Wait, so what’s “Physical Computing?”

Physical computing, the computer gathers information from its surroundings, processes that information using a code, and then uses the information to control other electronic parts. To do physical computing, you need

- **Sensors** - we brought light sensors, button switches, and a lever switch.
- **Actuators** - actuators “do” something - we brought colored lights and servo motors.
- A **programmable controller** or computer - we’re using the Arduino Uno
- **Code** that tells the Arduino how the actuators should react to the sensors

Computer and Wiring Guide

- Use jumper wires to connect components to the Arduino board and the power bus (a strip from a breadboard).
- All components have **5V** (red wire) and **GND/Ground** (Black wire) connections. Connect these to the 5V/GND power bus. 5V is indicated by a red (+), GND is indicated by a blue (-). Connect these wires to the plastic power bus.
- Each component also has 1 or 2 other wires for data (input or output). Connect these other colored wires to the numbered Arduino pin sockets as follows:

Photoresistor 1 --- A1
Photoresistor 2 --- A2

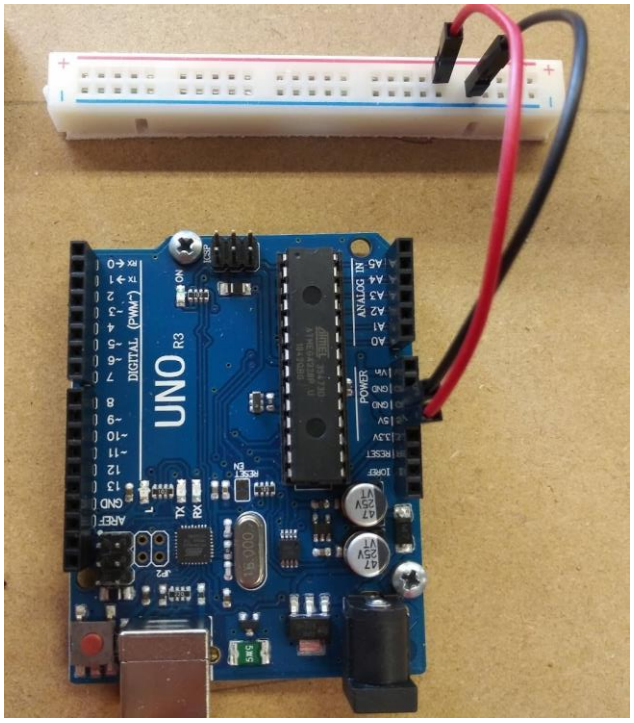
Distance sensor Trigger --- Pin 2
Distance sensor Echo --- Pin 3

Neopixel Ring --- 6

Servo Motor 1 --- 7
Servo Motor 2 --- 8

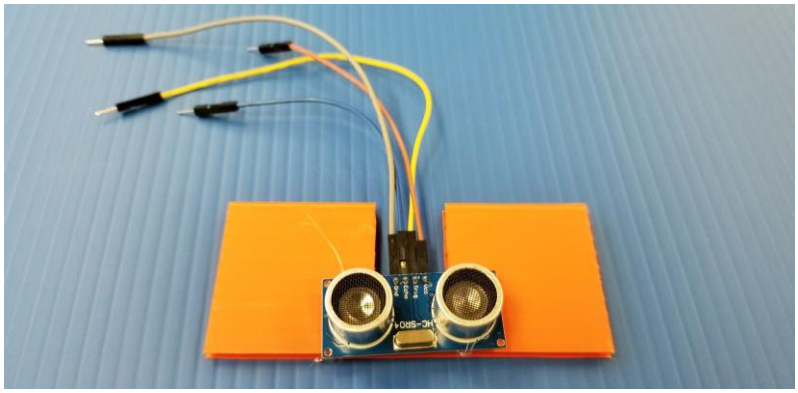
Lever Switch 1 --- 9
Lever Switch 2 --- 10

Pushbutton1 --- 11
Pushbutton2 --- 12



Code Library: Sensors (input devices)

Distance Sensor



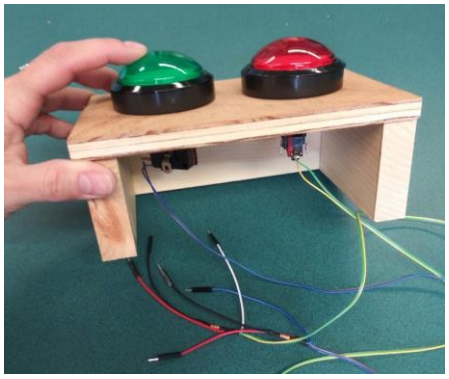
	COMMANDS	VARIABLES	NOTES
	ReadDistance();	Distance: integer from 0 to 1000. Valid distances are between 4 and 200	Get the distance in centimeters (cm) to a perpendicular surface, and put the result in the variable “distance”. Use conditional statements to check if the data is in the valid range (4 to 200 cm)

Light Sensors (photoresistors)

	COMMANDS	VARIABLES	NOTES
	ReadLightSensor1(); ReadLightSensor2();	Lightlevel1, Lightlevel2: integer value between 0 and 1023. The expected range is 400 - 900.	Get the light level from the specified Light Sensor, and put the result in the appropriate variable

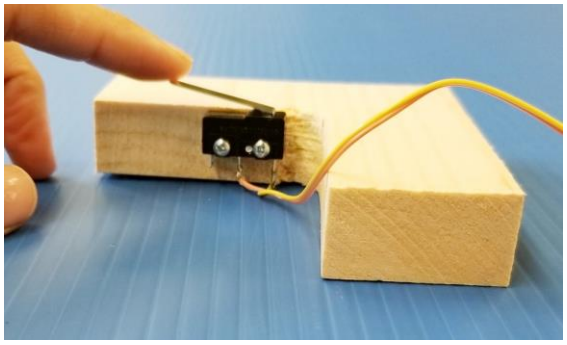
Code Library: Sensors (input devices)

Pushbutton switches



COMMANDS	VARIABLES	NOTES
ReadButton1(); ReadButton2();	button1, button2: HIGH (switch pressed) or LOW (switch loose)	Determine if a specified pushbutton switch is pressed or not. Get information from the switch and put the result in the appropriate variable

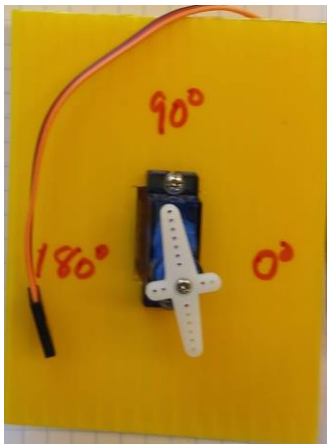
Lever switches



COMMANDS	VARIABLES	NOTES
ReadLeverSwitch1(); ReadLeverSwitch2();	lever1, lever2: HIGH (switch pressed) or LOW (switch loose)	Determine if a specified lever switch is pressed or not. Get information from the switch and put the result in the appropriate variable

Code Library: Actuators (output devices)

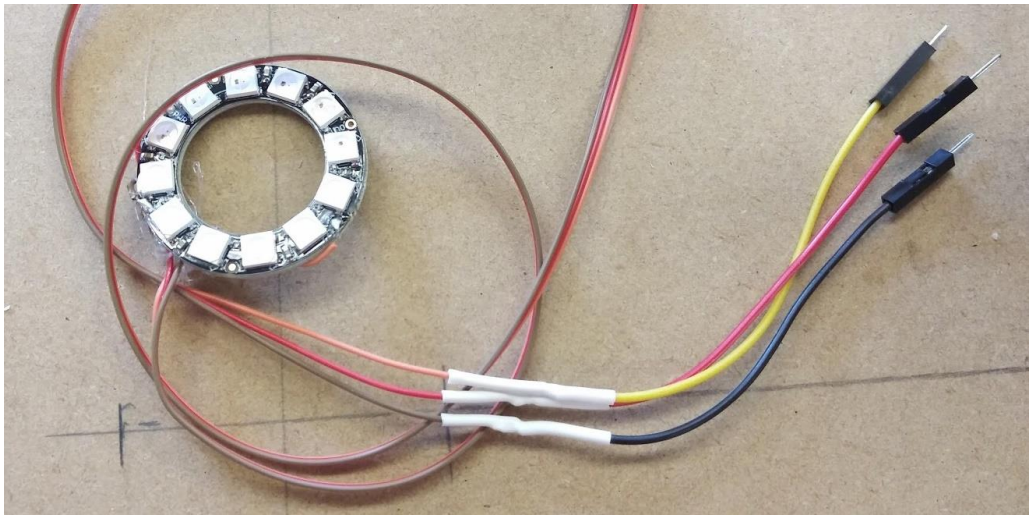
Servo Motors



	COMMANDS	VARIABLES	NOTES
	TurnServo1(angle); TurnServo2(angle);	angle: integer from 0 to 180	The servo motor will turn its arm to the angle in the parentheses. The variable “angle” is a degree value from 0 to 180 as shown on the servo motor mount
	ServoSpeed = speed;	speed: 1, 2, or 3.	The servo motor speed is set by the variable “speed”. Set it to 1, 2, or 3 where 1 = slow, 2 = medium, 3 or higher = fast.
	delay(delaytime);	delaytime: integer from 0 to 5000	Use delays to pause the code and slow the sensing and acting. The variable “delaytime” represents milliseconds, delay(1000) is 1 second

Code Library: Actuators (output devices)

NeoPixel Ring



	COMMANDS	VARIABLES	NOTES
	LightCircle(); LightHalfCircle(); LightSmile(); ClearCircle();	red: integer from 0 to 100 green: integer from 0 to 100 blue: integer from 0 to 100	Set the Neopixel color using variables “red”, “green”, and “blue”. Set each variable to any integer value from 0 (off) to 100 (bright)

Code Syntax and Structure

Code lines for loop()			
	COMMANDS	SYNTAX	NOTES
	Statements to set variables	variable = value;	The variable must go on the left of the = sign. Say it as “variable is set to value”

	Functions	function();
		OR
		function(value);
			Functions call a set of commands to do a specific task.
			All functions have (); after the name.
			Some functions require a value in the ()

	Conditionals
	
	A “*condition*” is evaluated. The “*something*” is done if the *condition* is TRUE
		if (*condition*)
		{
		do something
		}
		else (*condition*)
		{
		do something
		}
			Use these operators in the *condition*
			== Is Equivalent to
			!= Is NOT Equivalent to
			> Greater Than
			>= Greater Than or Equivalent to
			< Less Than
			<= Less Than or Equivalent to
			
			The “do something” statement must be in curly brackets { }, which can also be called “begin” and “end”
			
			The “else” part is optional

	Logic
	
	Use logic to evaluate 2 or more conditions
		if ((*condition*) && (*condition*))
		{
		do something
		}
			Use these operators for logic
			&& AND
			|| OR
			! NOT

	Comments	// notes in the code	Comments are notes in the code to help people understand it All words after // are not processed by the Arduino. Another way to make comments is like this /* comment */

Garduino Examples with Code

- Sunflower turning to the sunlight on either side
- Bird in a nest flies away when cat approaches
- Colored Flower - lights up different colors when colored buttons are pressed.
- Buzzy Bees - buzz around when a character approaches
- Critter lights up a smile when a friend sits down or a plate of sweets is set near it

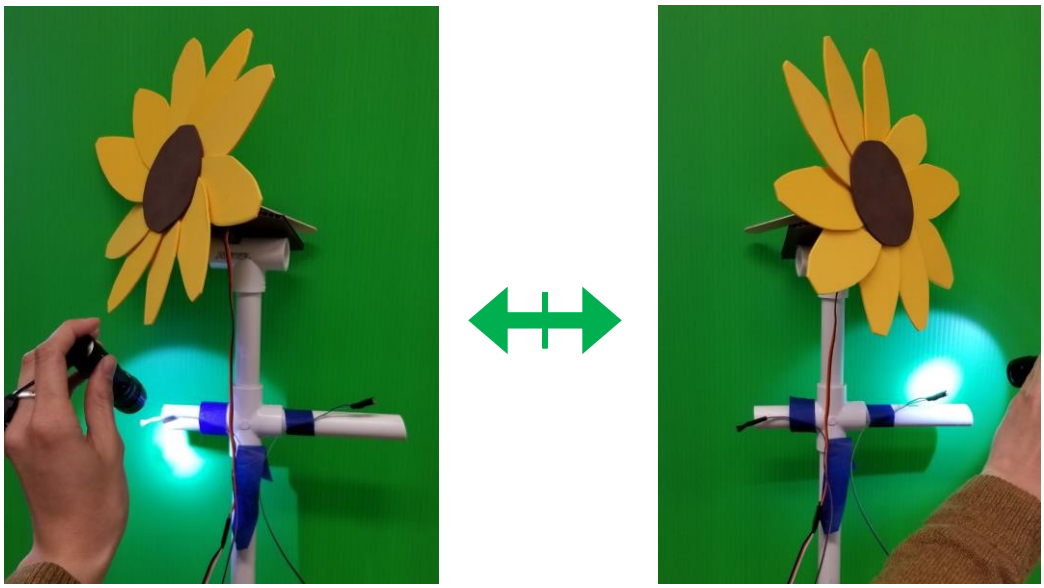
Garduinos ideas

- Rabbit chomping on plants
- Woodchuck chewing
- Watering Can tipping
- Squirrel throwing a nut
- Flamingo
- Sun/Clouds/Rain moving around
- Gnome or critter peeking out from behind a rock
- Bird Bath
- ?? What can you add to the Garden?

Sunflower

Turns to face a light source

Uses 2 Light Sensors, 1 Servo Motor



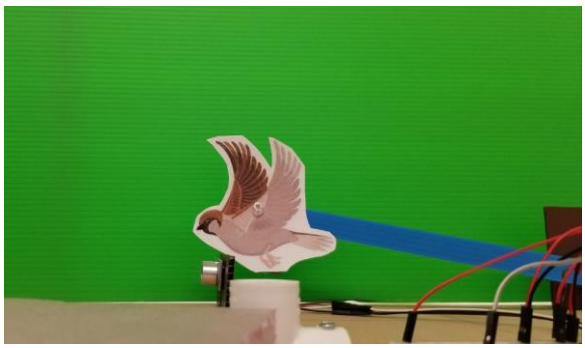
WHAT TO TYPE:

```
/************************* your code here ************************/
ReadLightSensor1();       // get light level from sensor 1 on the left
ReadLightSensor2();       // get light level from sensor 2 on the right

if (lightlevel1 > 850)              // check for light on left
   {
     ServoSpeed = 2;
     TurnServo1(30);
     delay(1000);
   }

if (lightlevel2 > 850)              // check for light on right
   {
     ServoSpeed = 2;
     TurnServo1(150);
     delay(1000);
   }
/*****************************************************************************/
```

Runaway bird

Flies away when predator approaches

Parts: 1 Distance Sensor, 1 Servo Motor



WHAT TO TYPE:

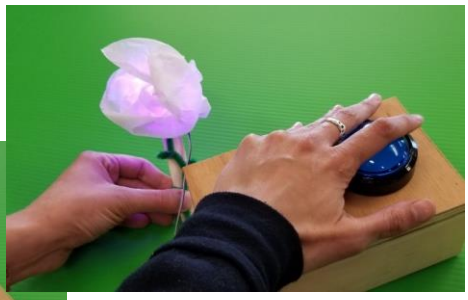
```
/************************* your code here ************************/
ReadDistance();
  if ((distance > 3) && (distance < 10))    // too close! make bird fly away
    {
    ServoSpeed = 3;
    TurnServo1(90);
    delay(1000);
    }
  else                // nothing nearby, safe for bird to land
    {
    ServoSpeed = 2;
    TurnServo1(170);
    }
/*****************************************************************************/
```

Colored Flower

Lights up different colors when colored buttons are pressed.

Optional: blink the flower every second

Parts: 2 colored pushbuttons, 1 neopixel ring



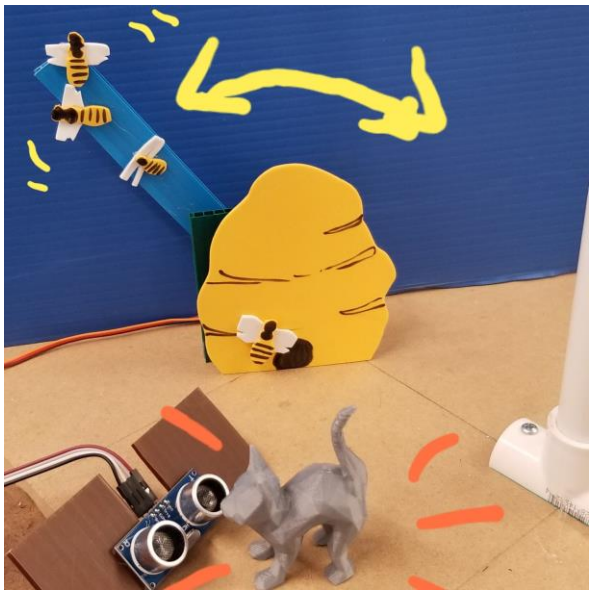
WHAT TO TYPE:

```
/************************* your code here ************************/
  red = 0; green = 0; blue = 0;
  ReadButton1();         // red button
  ReadButton2();         // green button

  if (button1 == HIGH)    // check if red button is pushed
     {
      red = 50;
      }
  if (button2 == HIGH)    // check if green button is pushed
     {
      green = 50;
      }

  LightCircle();            // blink code
  delay (1000);
  ClearCircle();
  delay (1000);
/*****************************************************************************/
```

Buzzy bees

Fly around when something approaches

Parts: 1 Distance Sensor, 1 Servo Motor



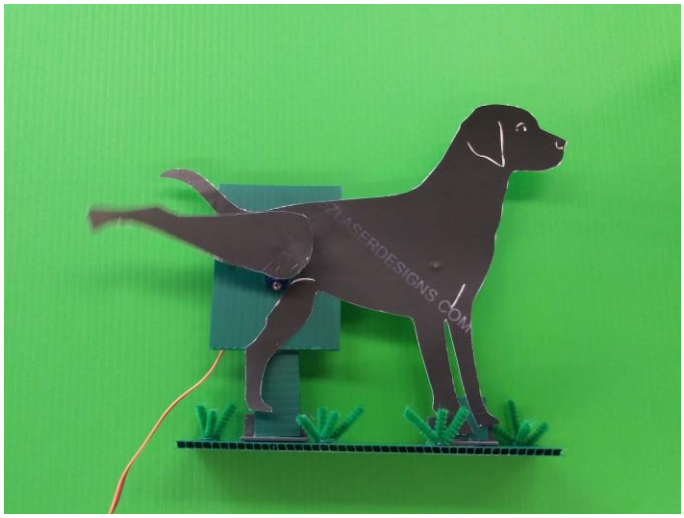
WHAT TO TYPE:

```
/************************* your code here ************************/
ReadDistance();
  if ((distance > 3) && (distance < 20))          // “Something got too close! Time to swarm!”
    {
    ServoSpeed = 3;
    TurnServo1(0);       // left wing out
    TurnServo2(180);   // right wing out
    delay(200);

    }
/*****************************************************************************/
```

Dog Lifting Leg

Press the lever switch and see the dog pee!

Parts: 1 Lever Switch, 1 Servo Motor



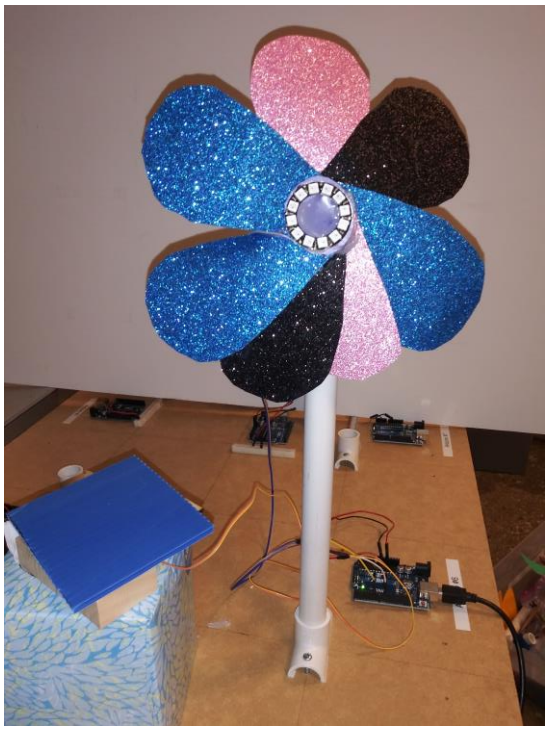
WHAT TO TYPE:

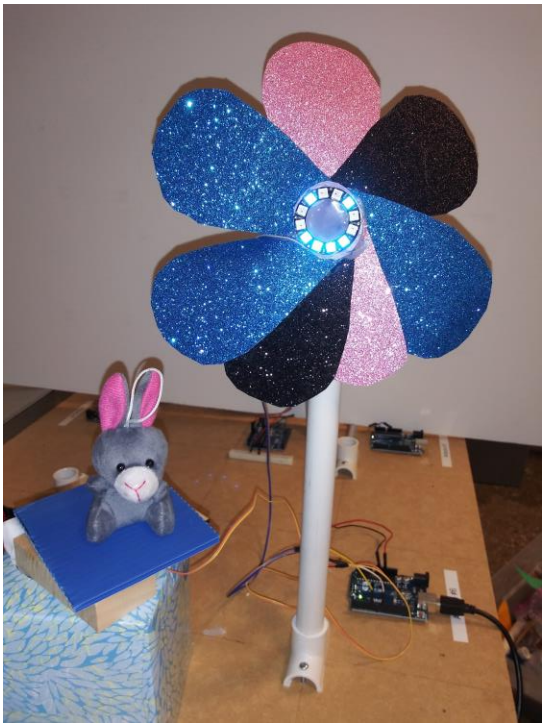
```
 /********************* your code here ********************************/
ReadLeverSwitch1();
ServoSpeed = 3;
if (lever1== HIGH)
{
  TurnServo1(0);
  delay(500);
  }
TurnServo1(90);
delay(500);

/*******************************************************/
```

Friendly critter

Lights up a smile when someone sits next to it

Parts: 1 Lever Switch, 1 Neopixel ring





WHAT TO TYPE:

```
/************************* your code here ************************/

  red = 0; green = 50; blue = 100;              //combine 3 colors of light to color the smile
  delay(250);              // wait ¼ second
  ReadLeverSwitch1();
  if (lever1 == HIGH)
     {
       LightSmile();
      }
  else
     {
       ClearCircle();
      }
/*****************************************************************************/
```

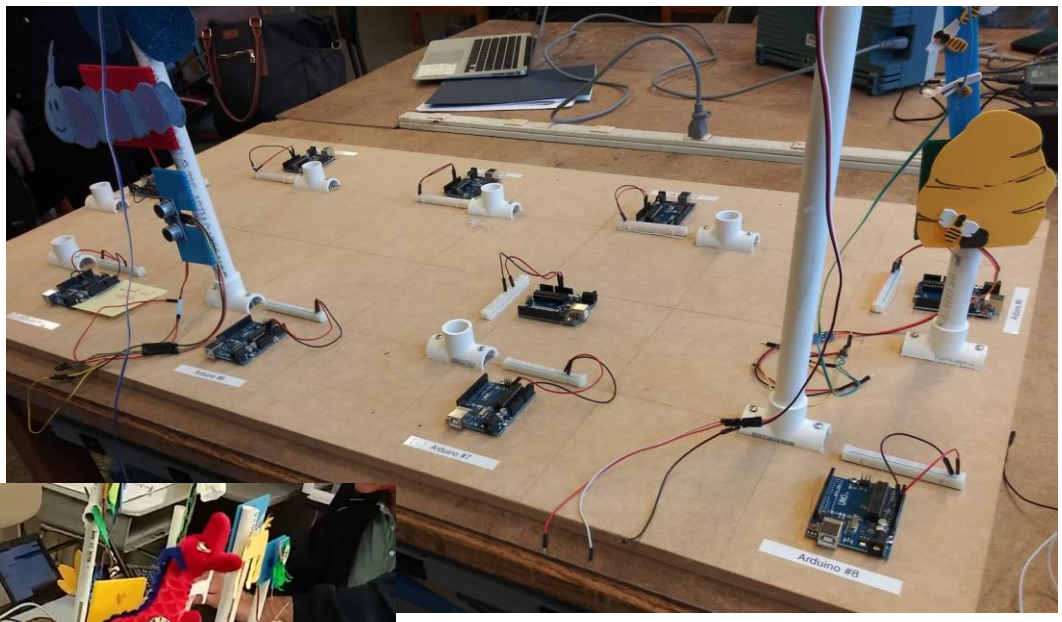
Build Guide: Garden base, Sensors, and Actuators

Garden base

Material suggestion: 2’ x 4’ sheet of ½” MDF

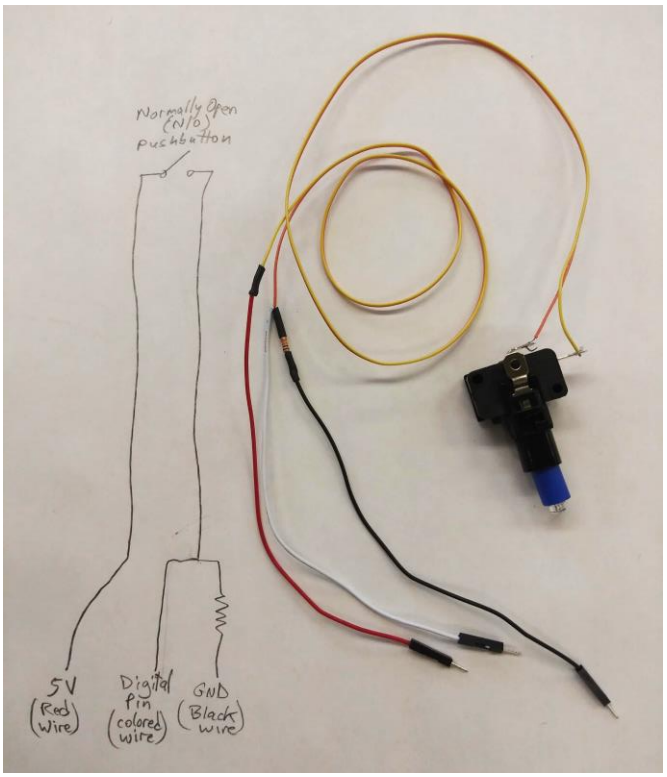
For each station

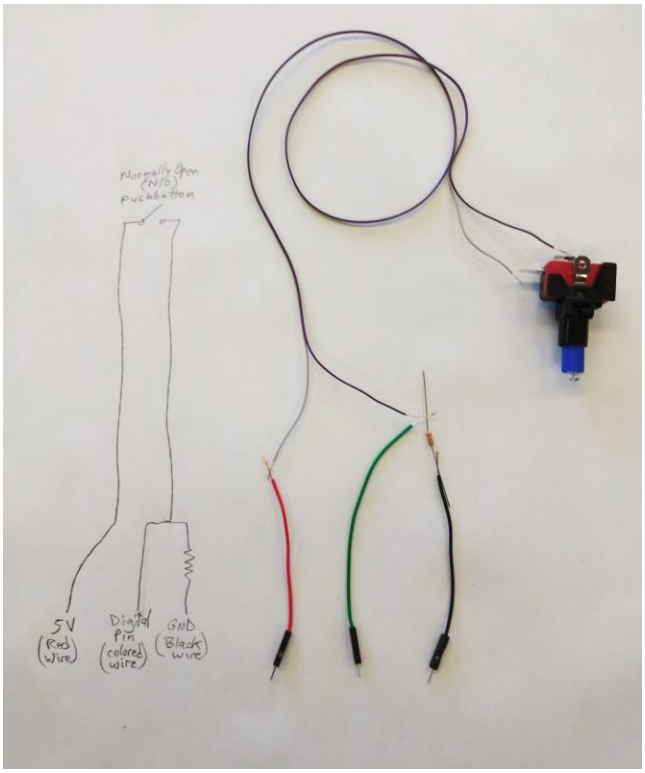
- 1 Arduino Uno, screwed in with wood screws
- 1 power bus from a 400 hole solderless breadboard (each board has 2, just snap them off)
- 1 ½” PVC T connector, cut as shown and with 2 clearance holes. Screw to MDF board with wood screws.
- 1 red and 1 black M-M jumper (or solid hook-up wire, 22 AWG) to connect Arduino to power bus
- USB hubs (not shown) to power all Arduinos when not connected to computers

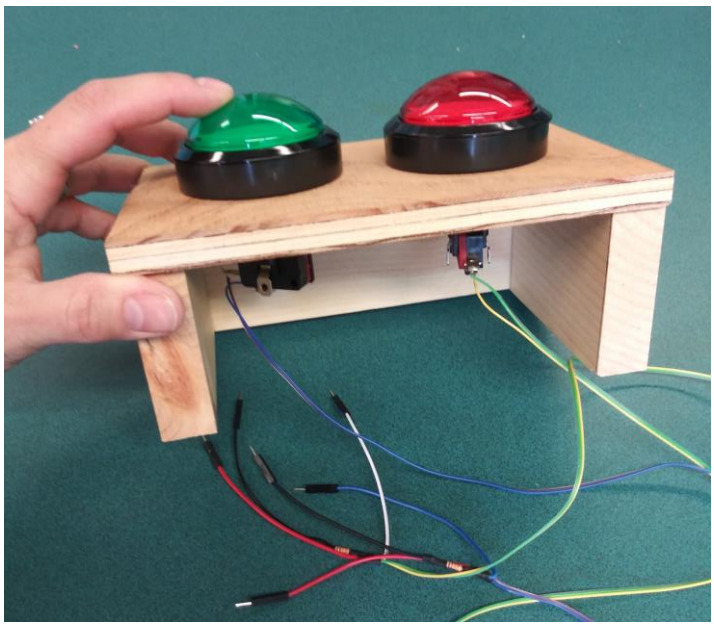


Sensors and Actuators

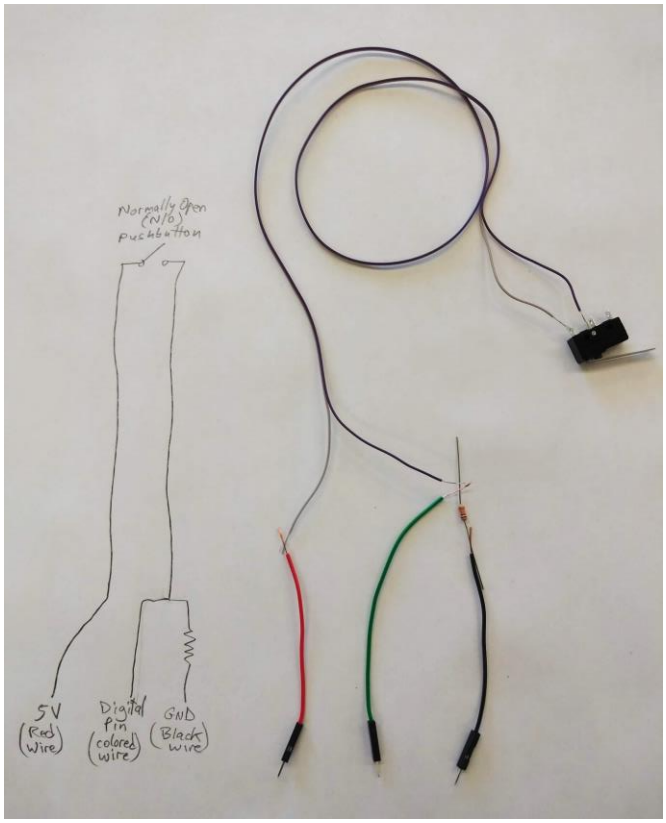
Pushbutton Switch

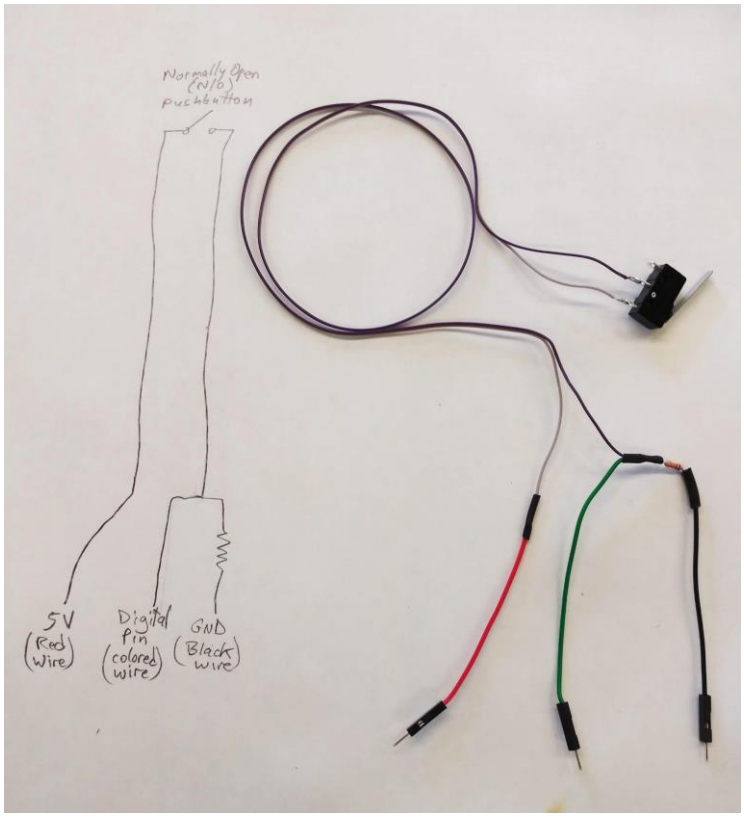


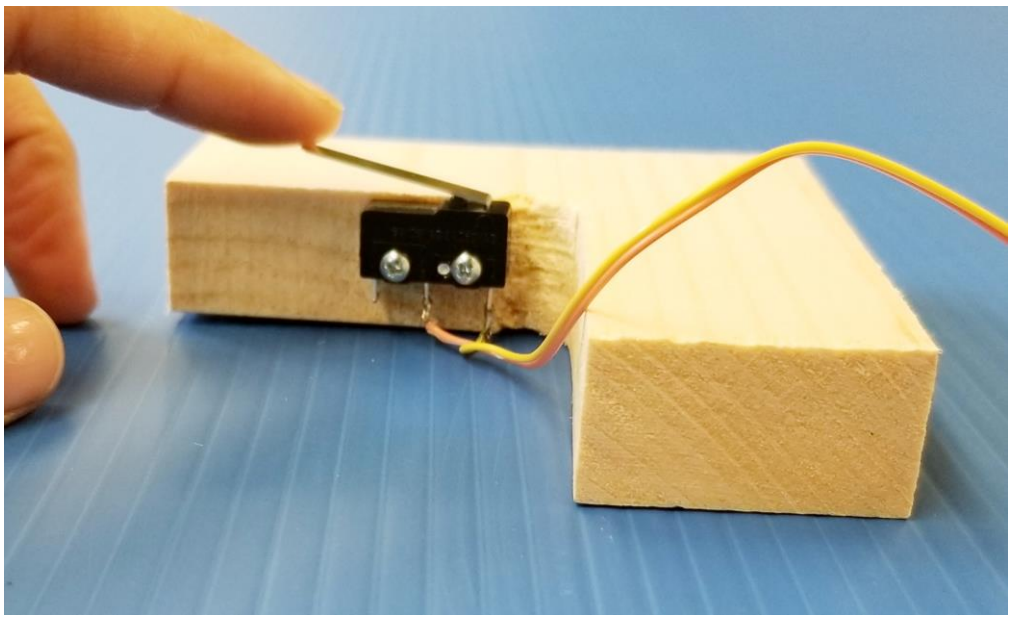




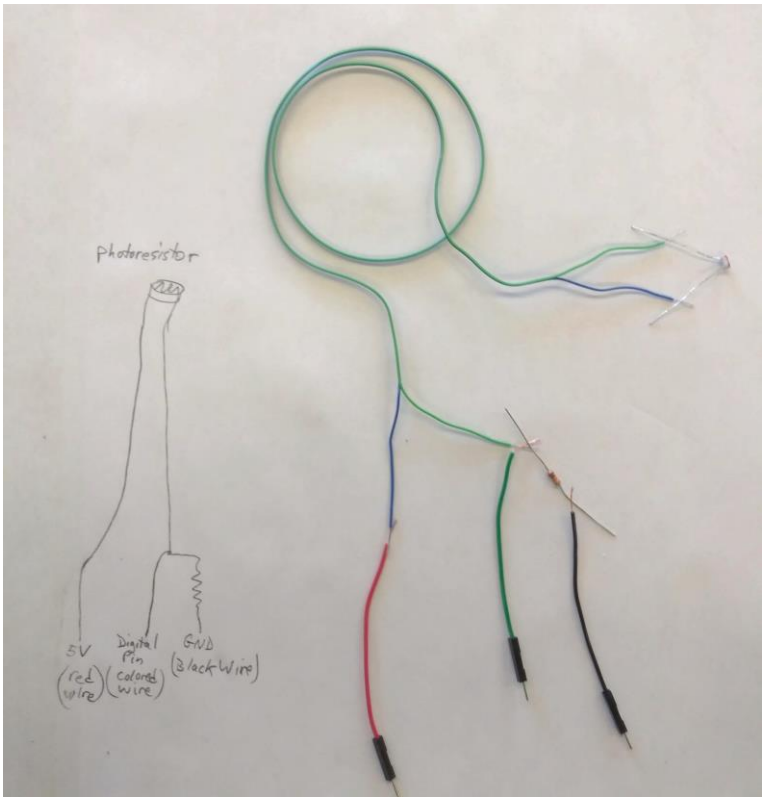
Lever Switch

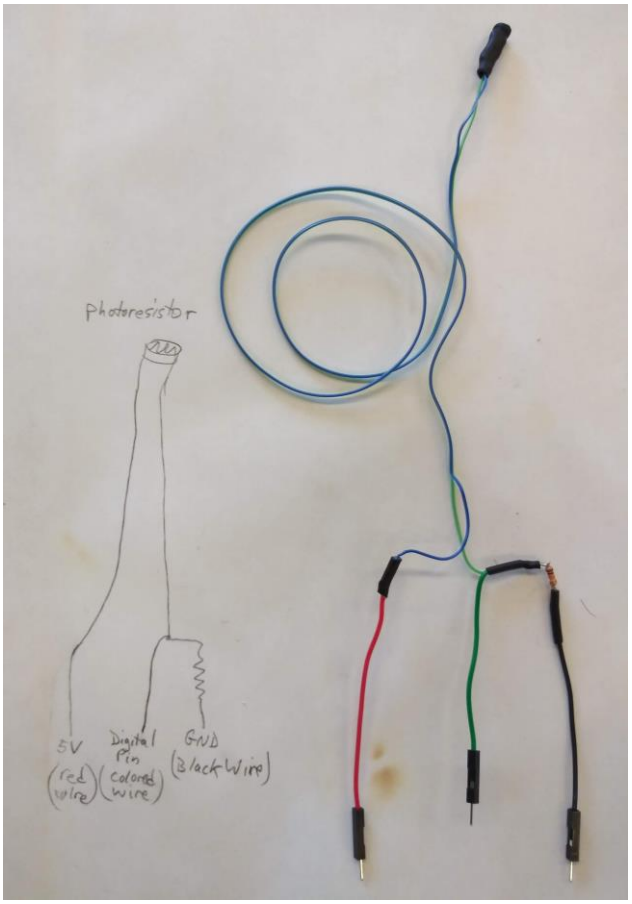


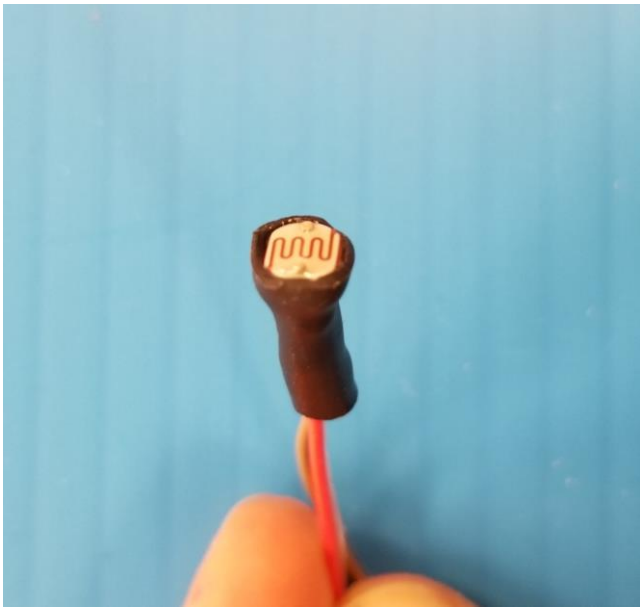


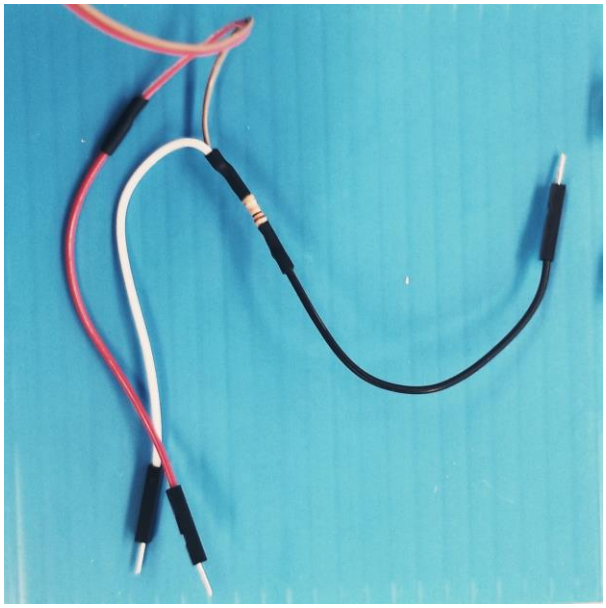


Photoresistor with pull-down resistor

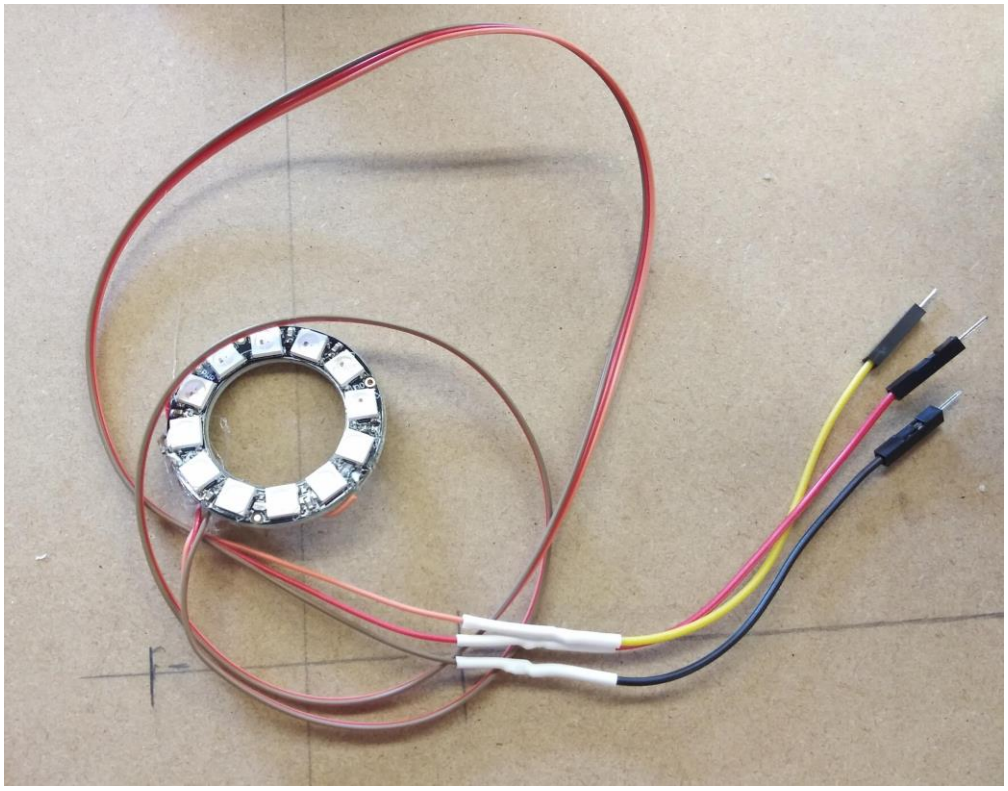


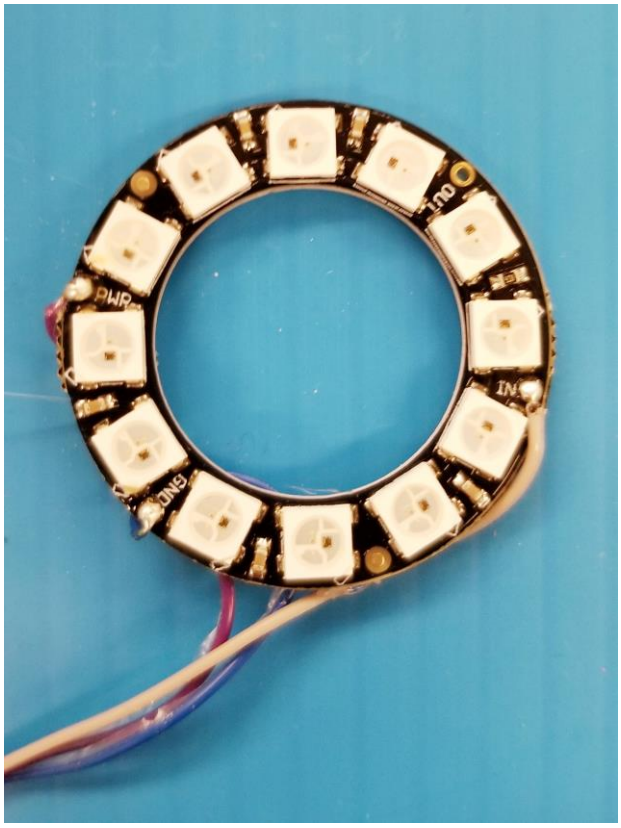




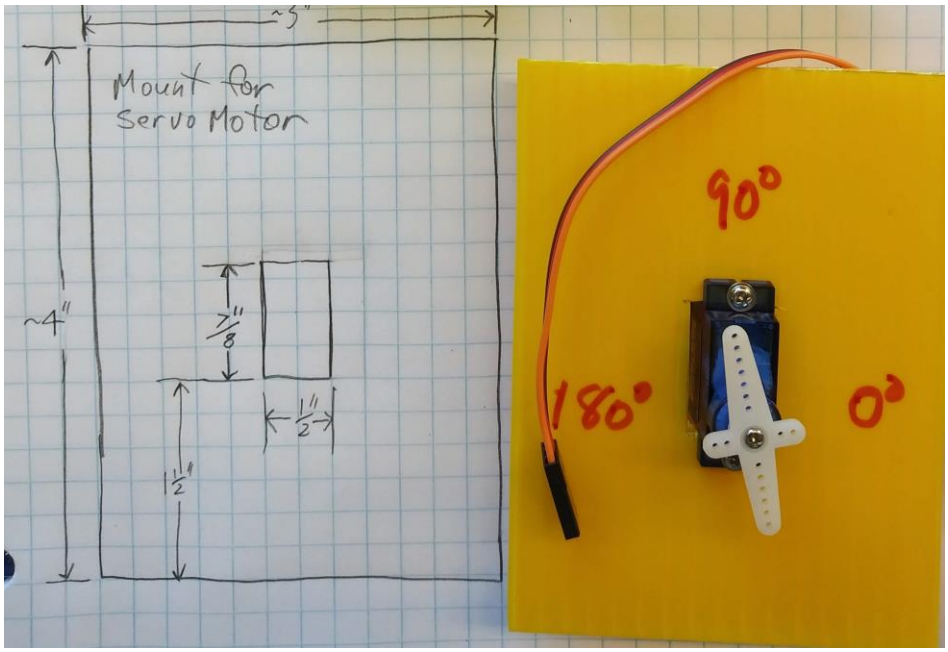


Neopixel Ring





Servo Motor and Mount



Ultrasonic Distance Sensor and Mount