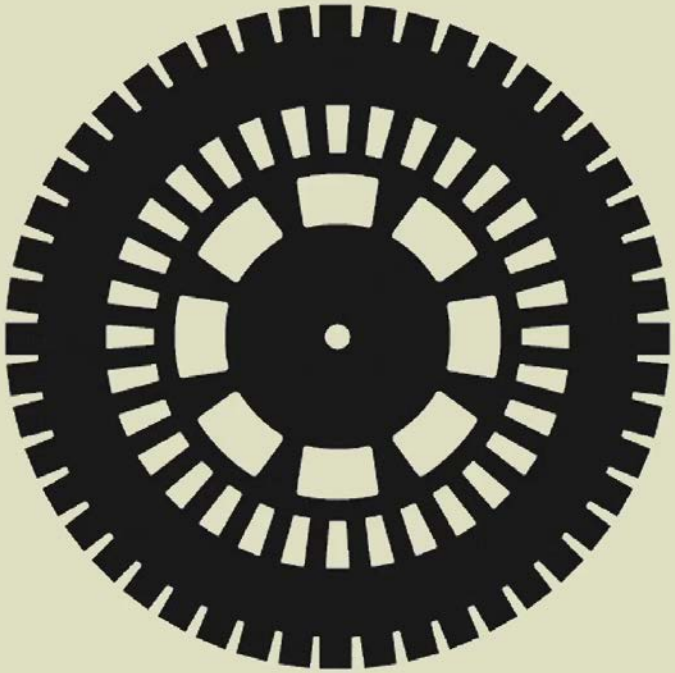
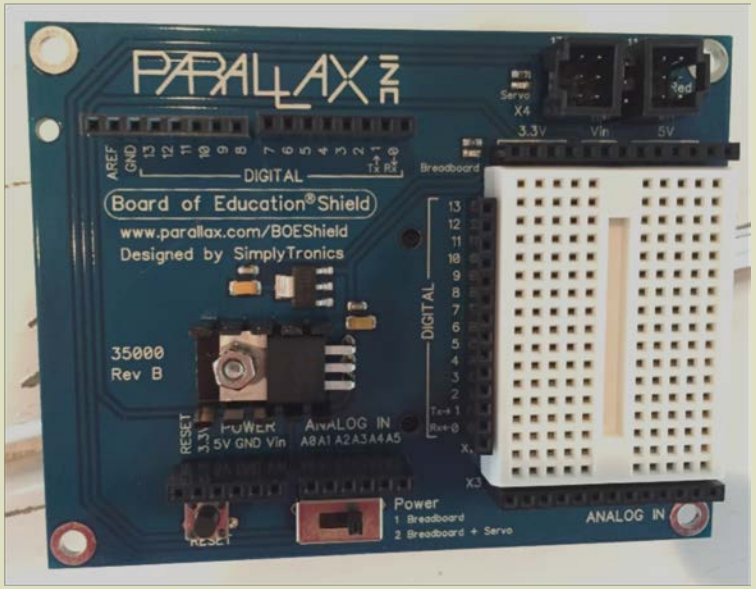
Welcome!



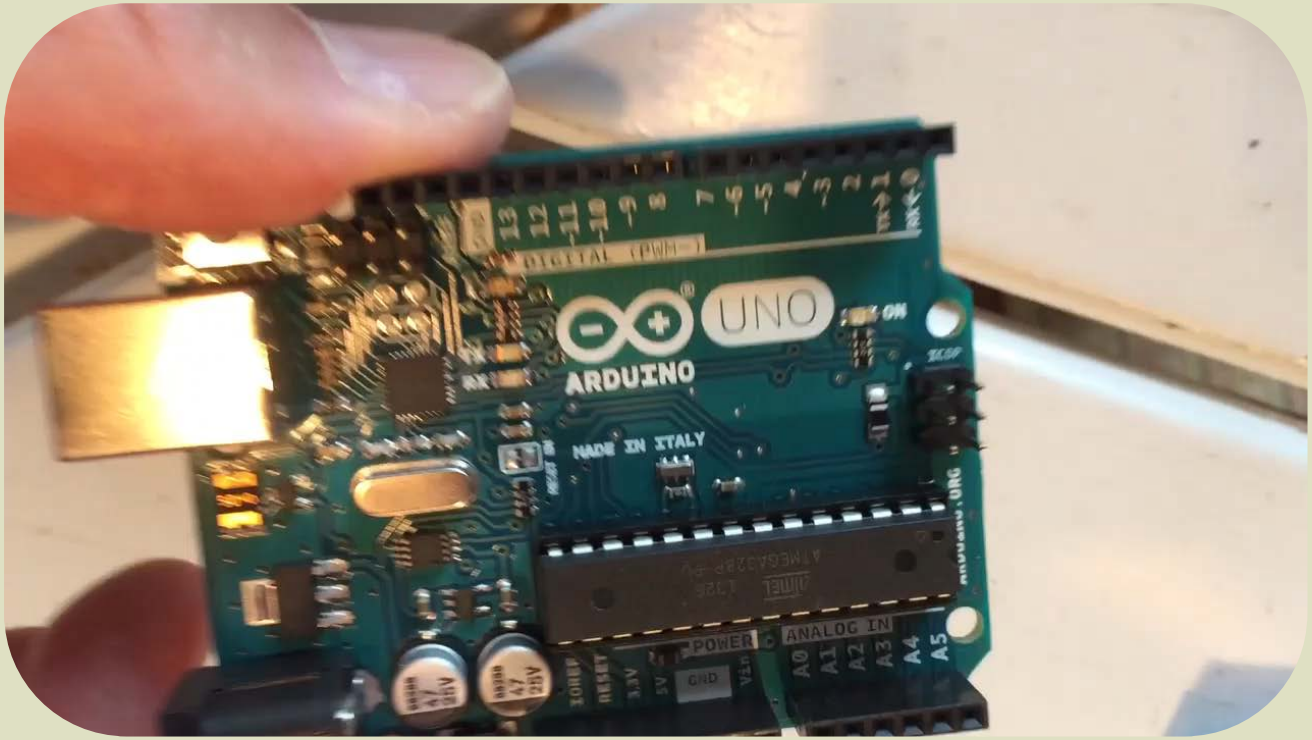
gizmo garden©

Two Boards



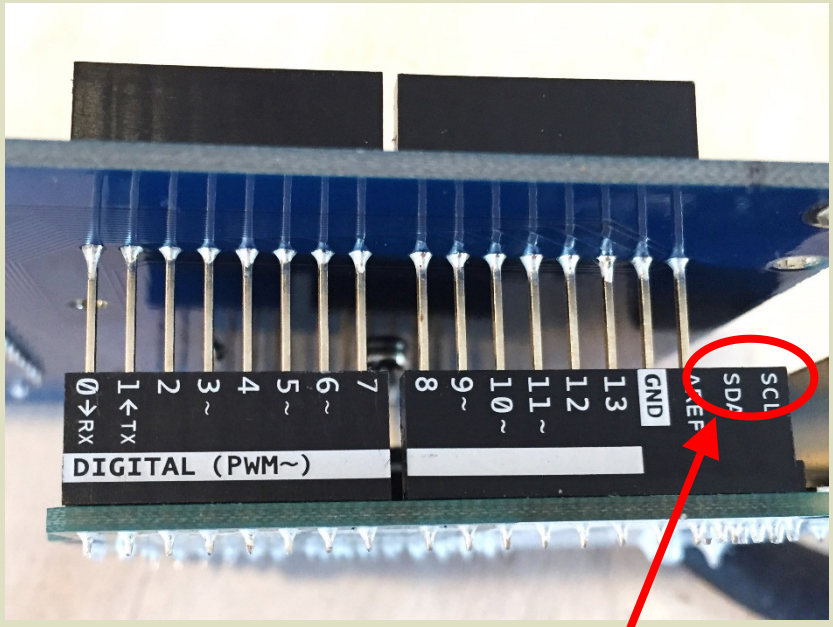
gizmo garden©

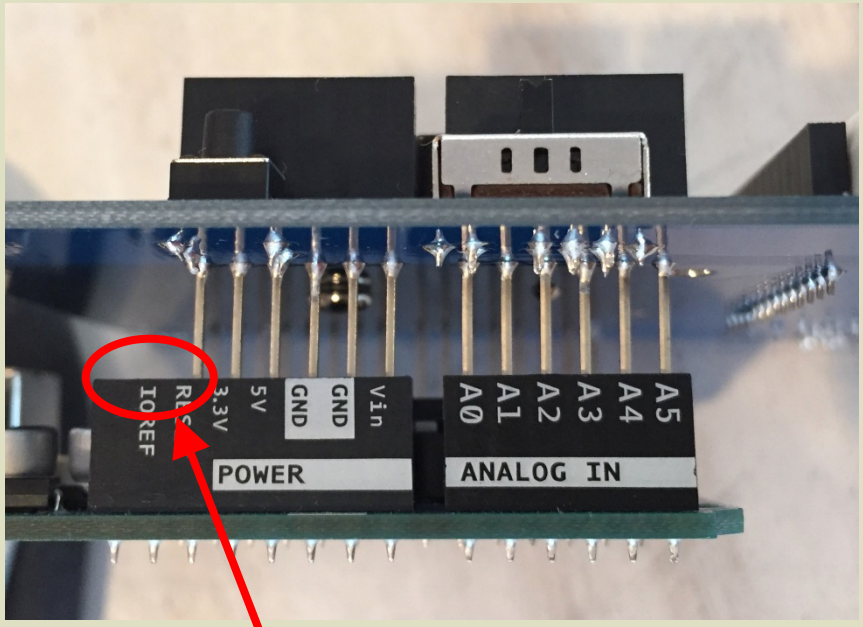
Mating Boards



gizmo garden©

Line Up the Pins





two empty holes

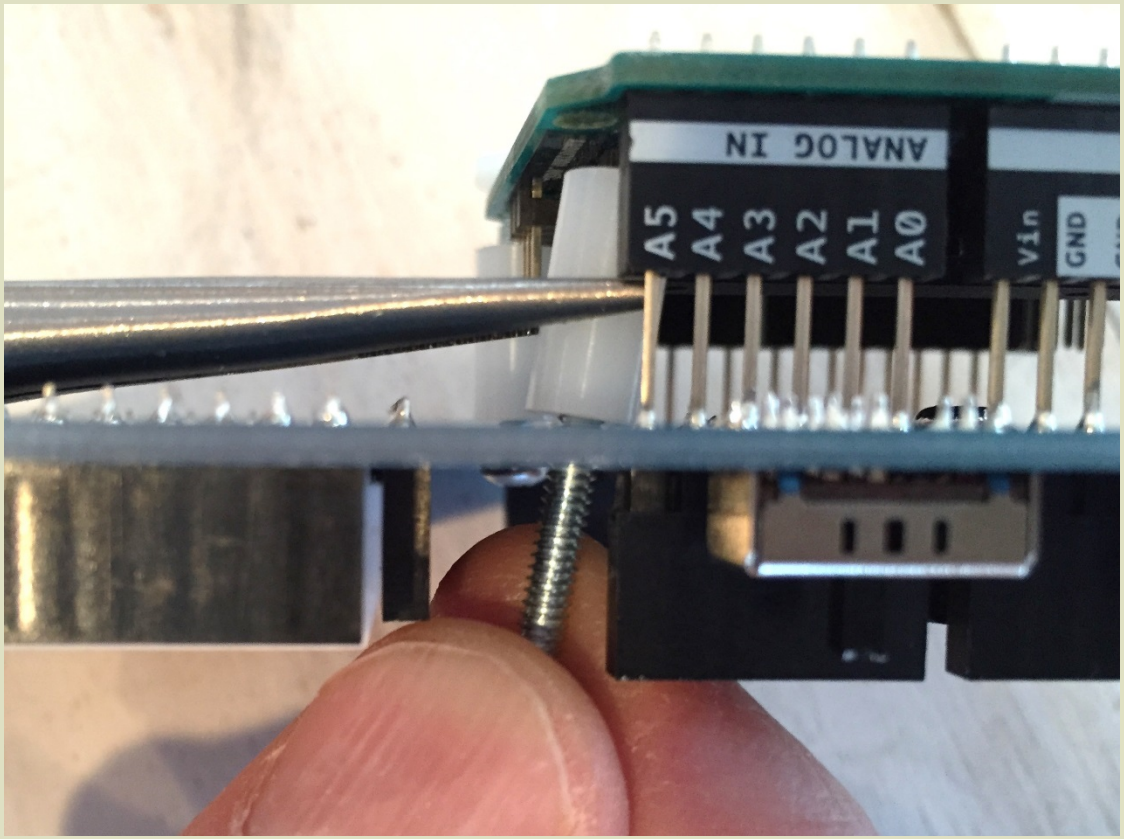
one empty hole

Don’t push the boards all the way together yet

gizmo garden©

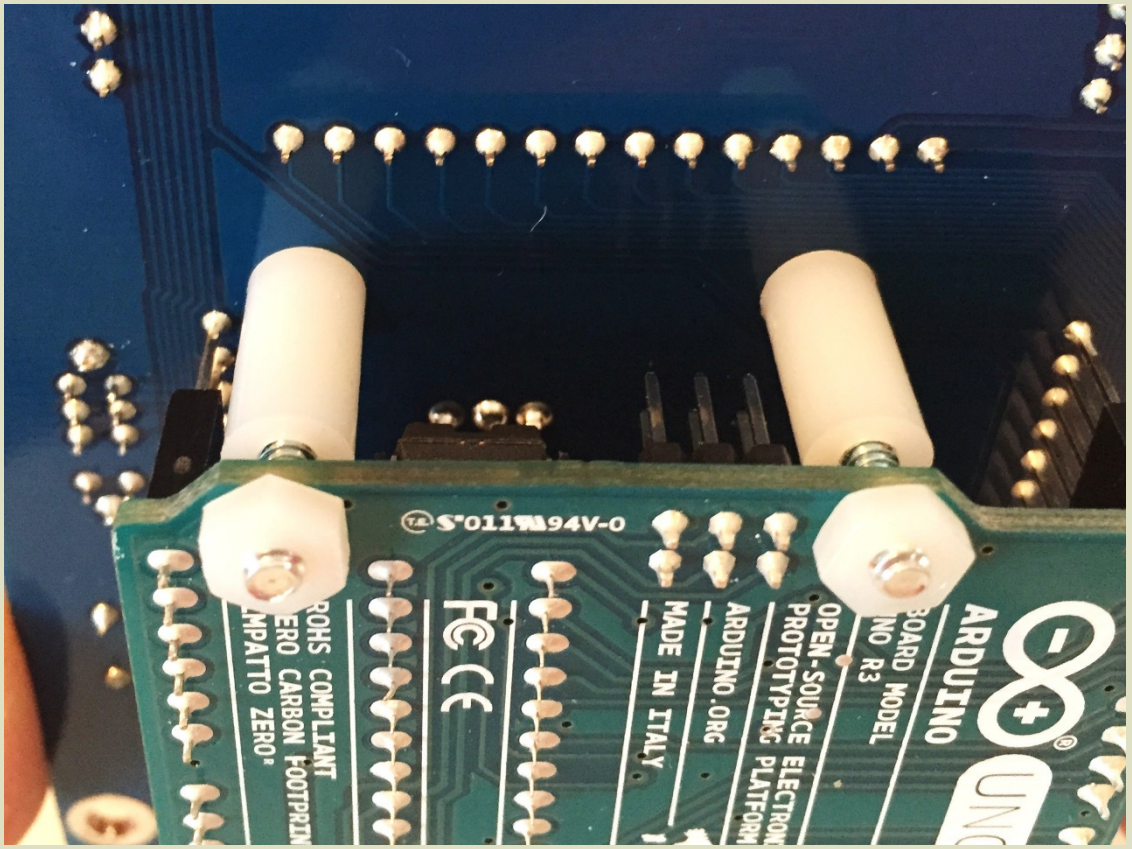
Insert 2 Nylon Spacers

Secure with 2 1” screws, 2 nylon nuts



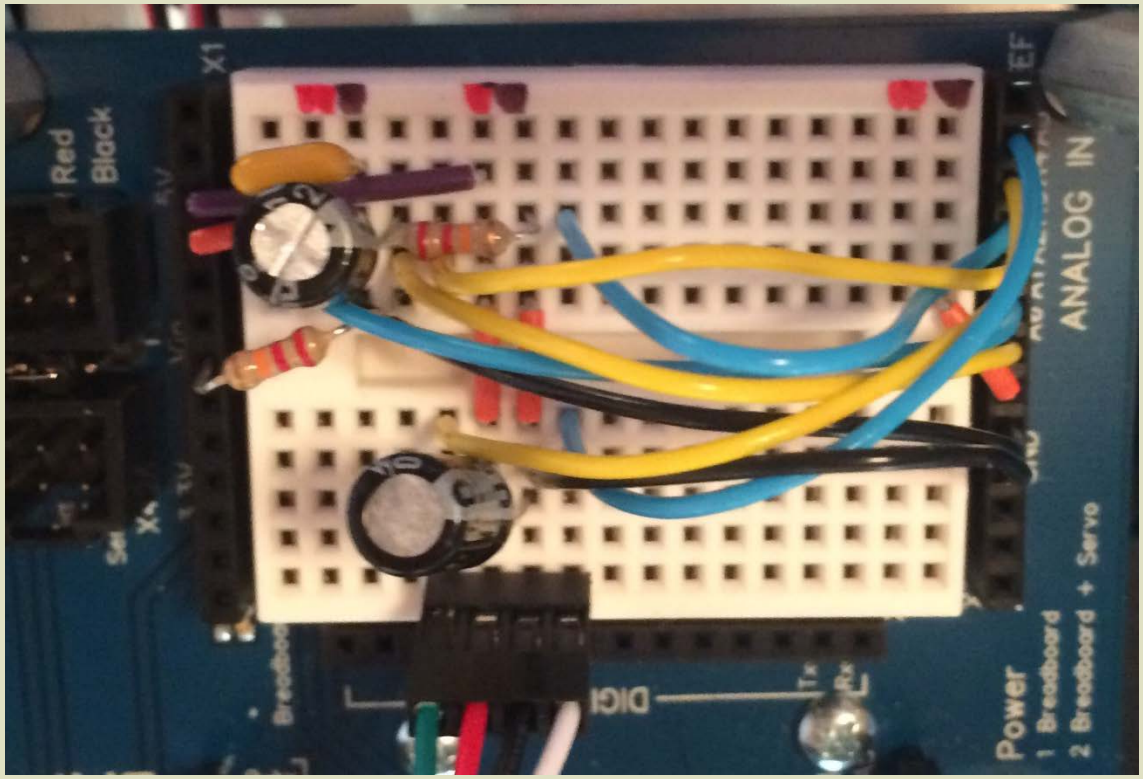
gizmo garden©

Nylon Spacers Installed



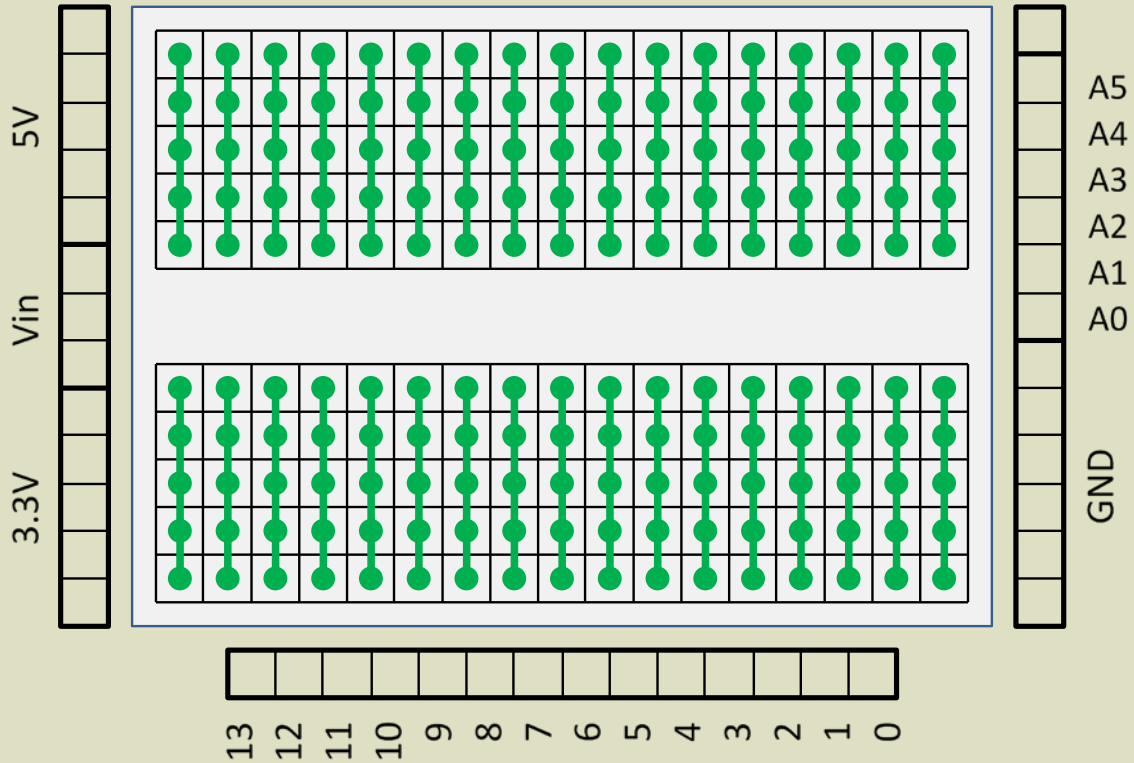
gizmo garden©

Main Breadboard



gizmo garden©

Breadboard Connections



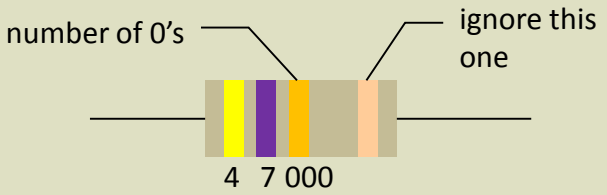
gizmo garden©

Safety Glasses

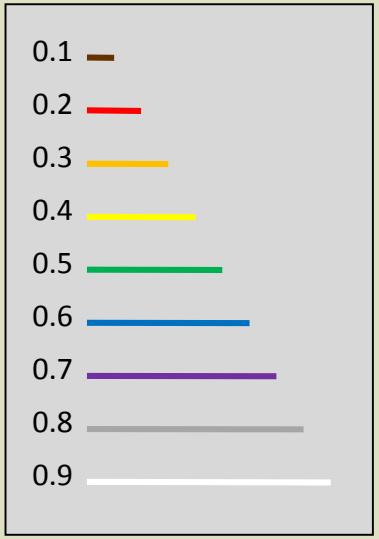
Color Codes for Numbers

	Color	Value
Colors of the Rainbow	black	0
	brown	1
	red	2
	orange	3
	yellow	4
	green	5
	blue	6
	violet	7
	grey	8
	white	9

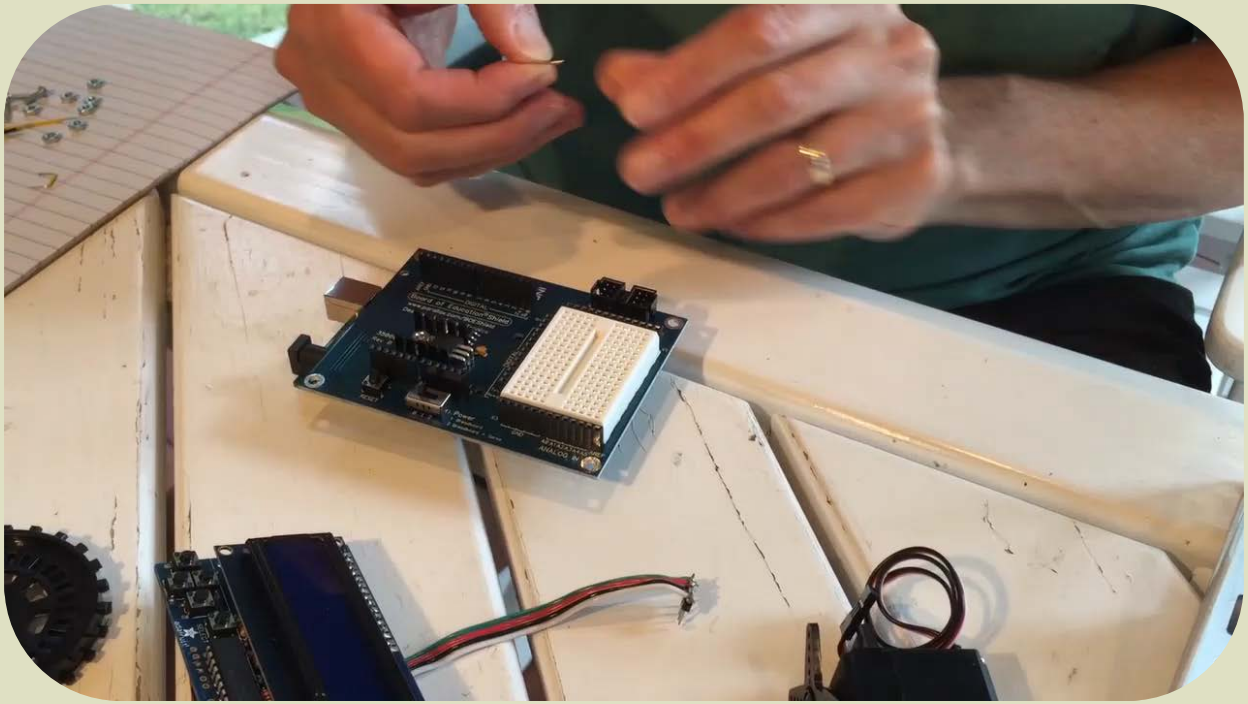
A 47,000 ohm resistor (47 KΩ)



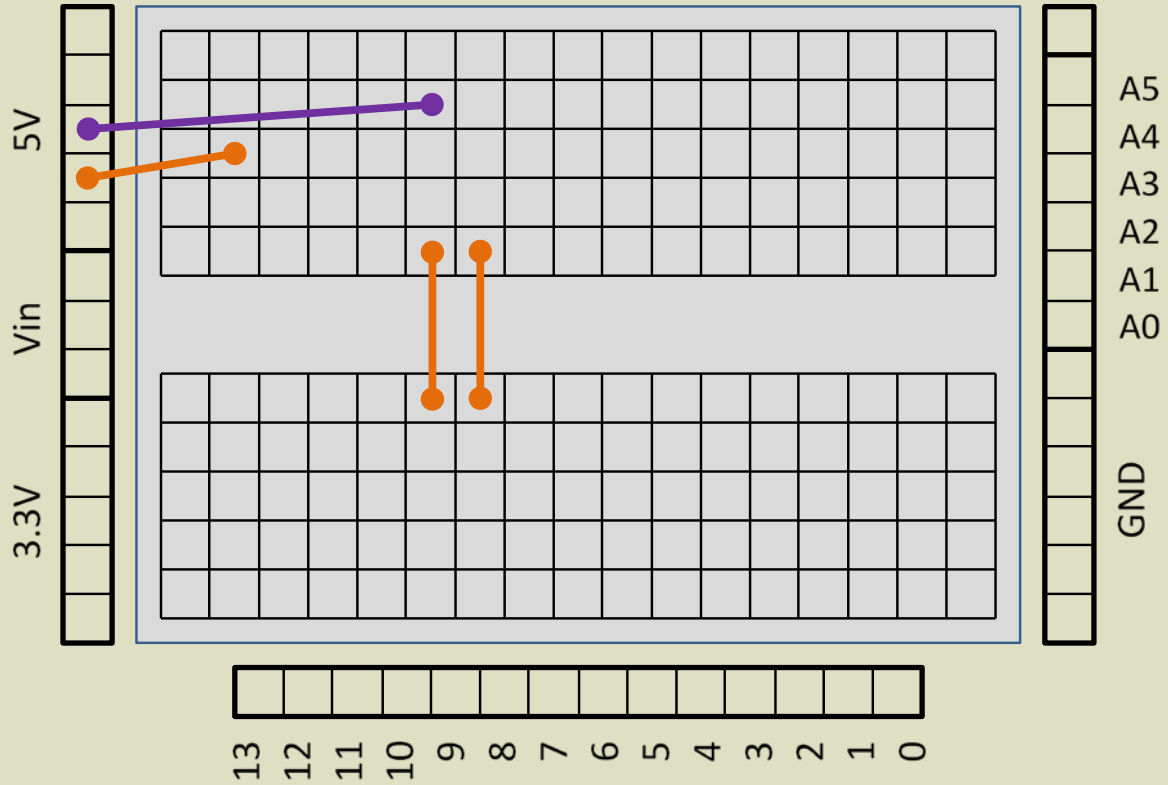
Wire Lengths



Bread Boarding Wires

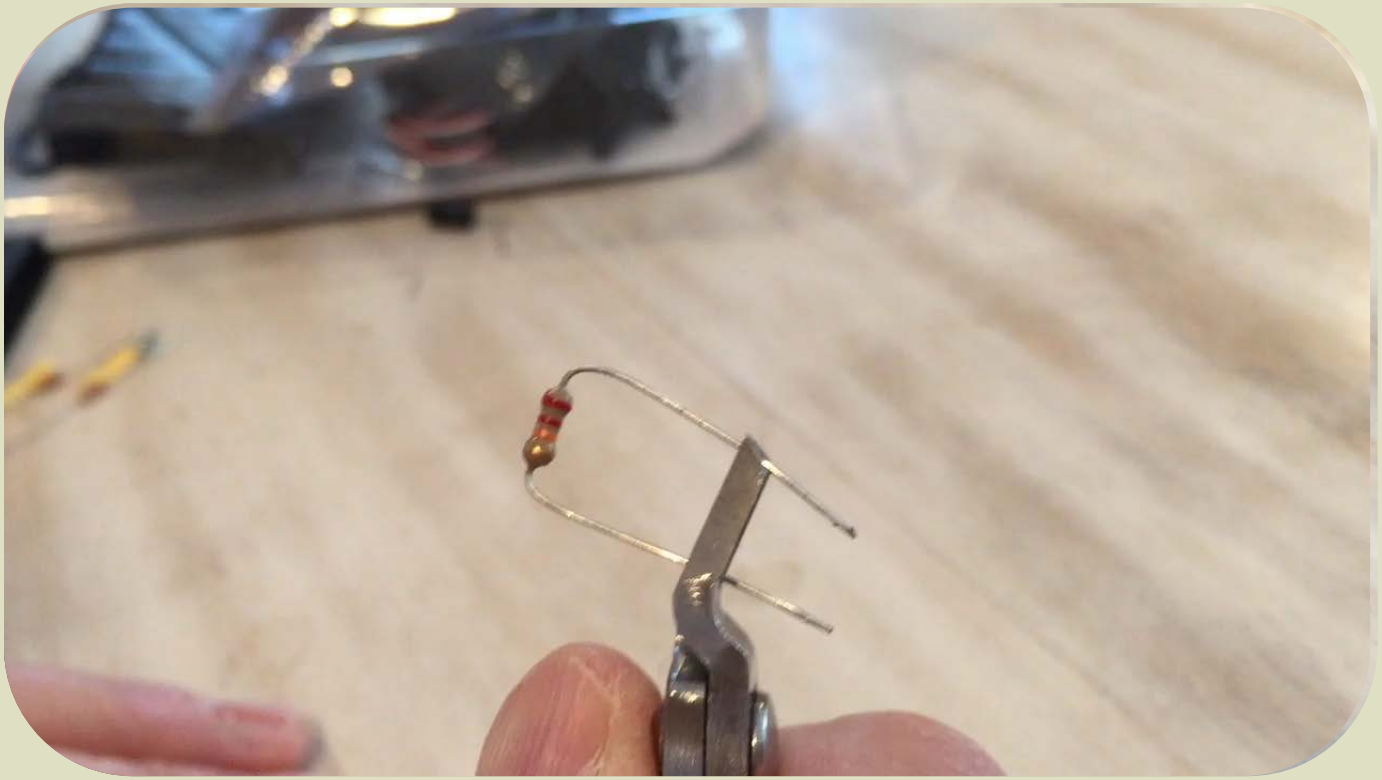


Short Wires

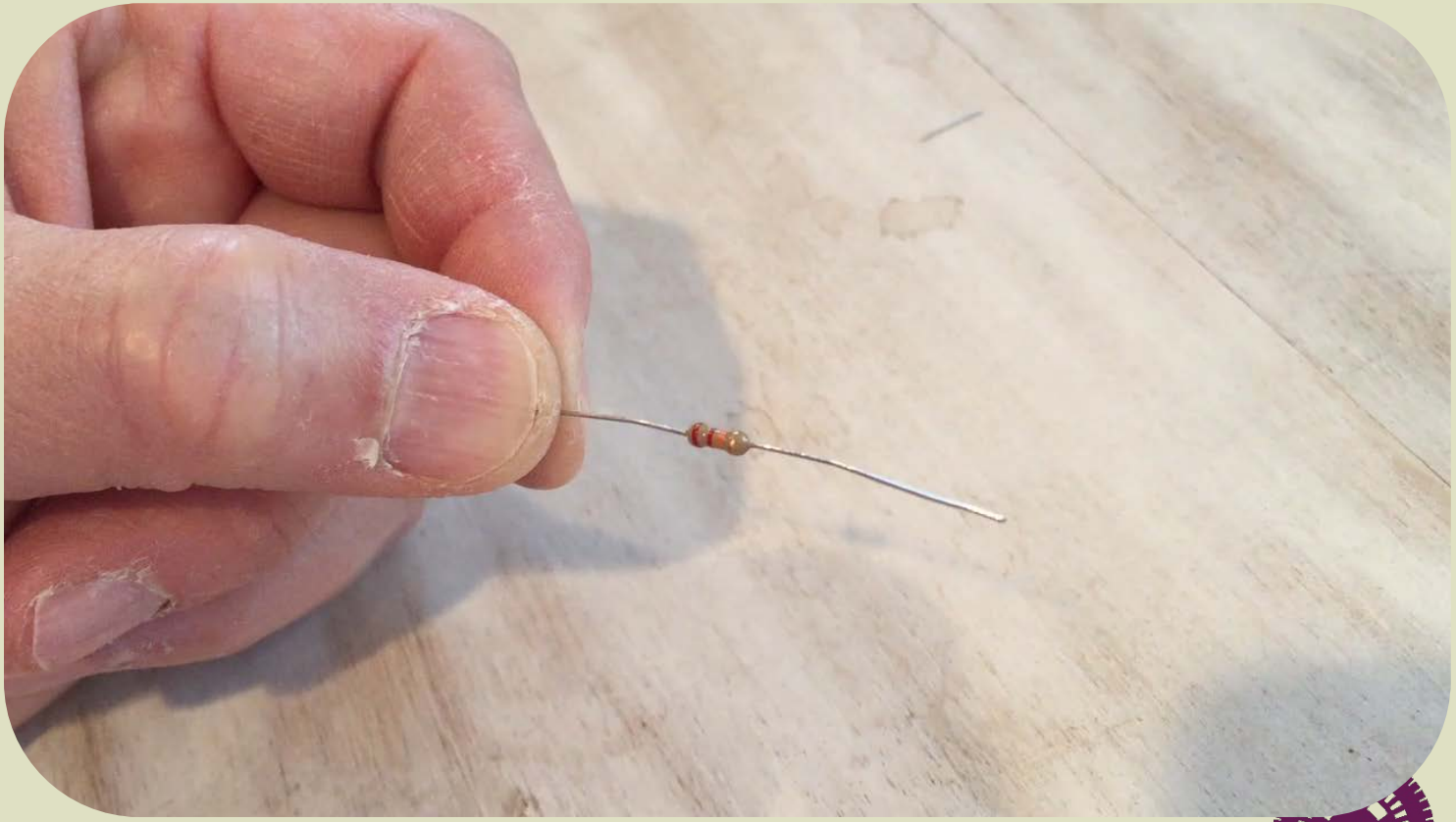


gizmo garden©

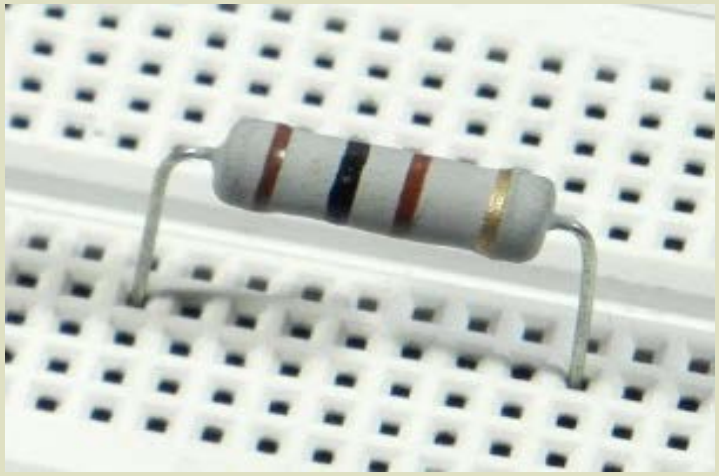
Resistors

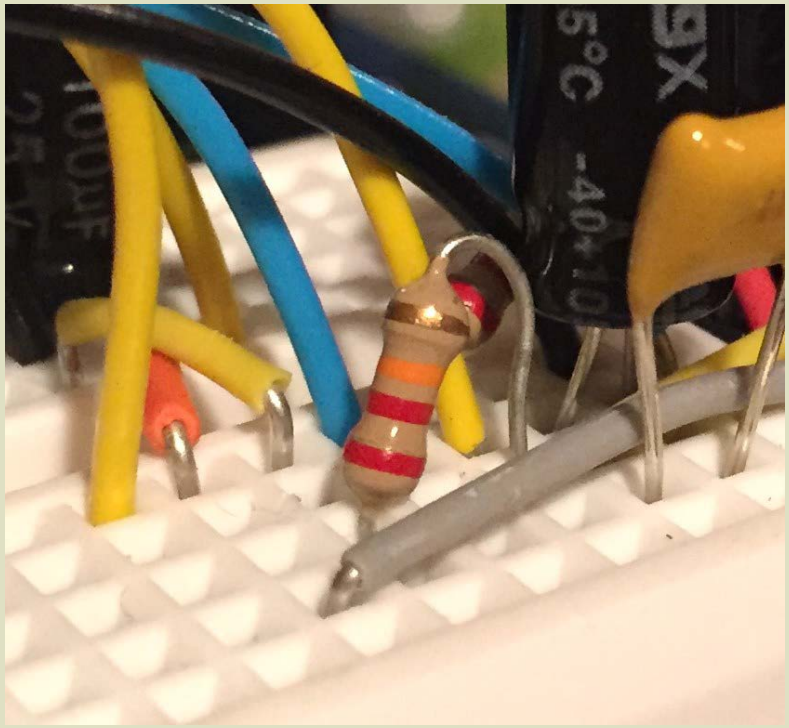


Resistors



Resistors

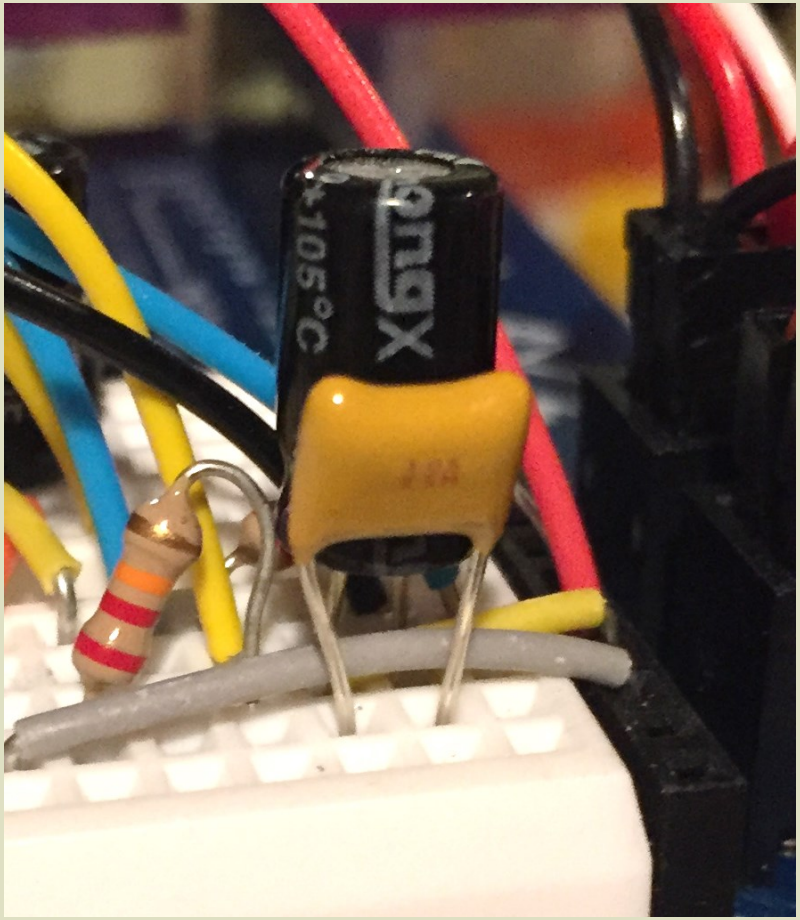




Capacitors

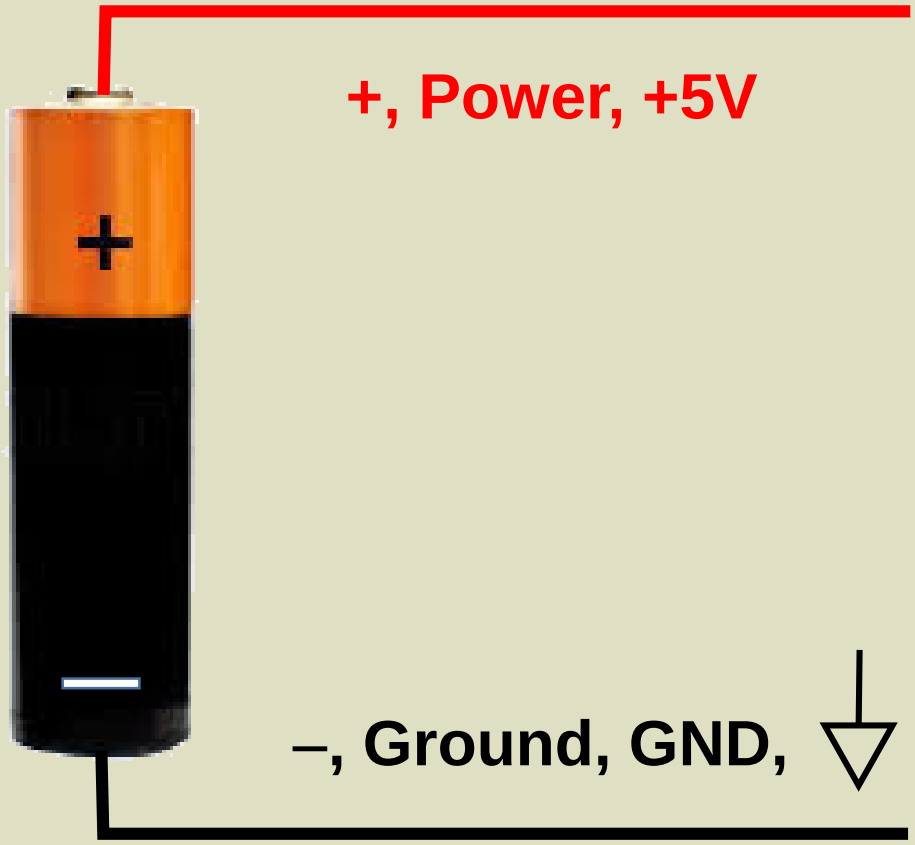
gizmo garden©

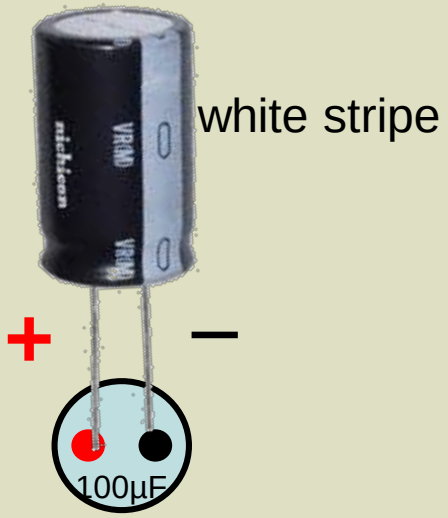
Another Kind of Capacitor



gizmo garden©

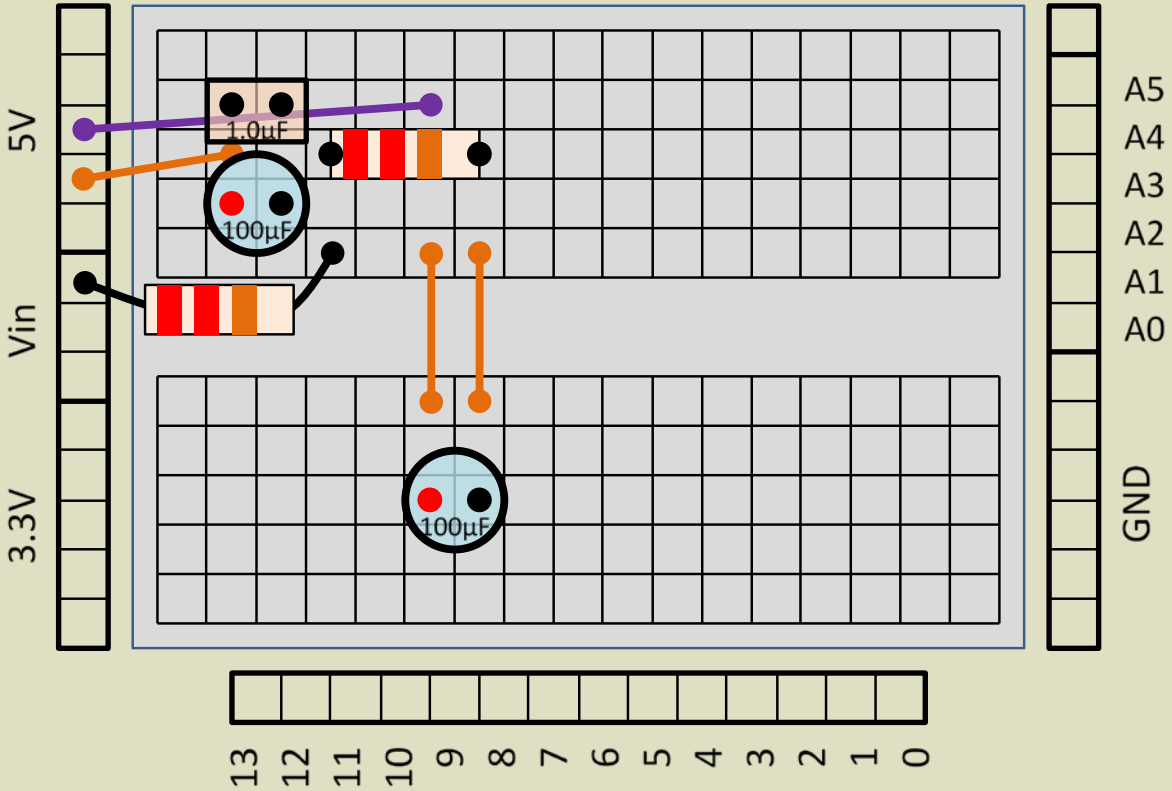
Color Codes for Power





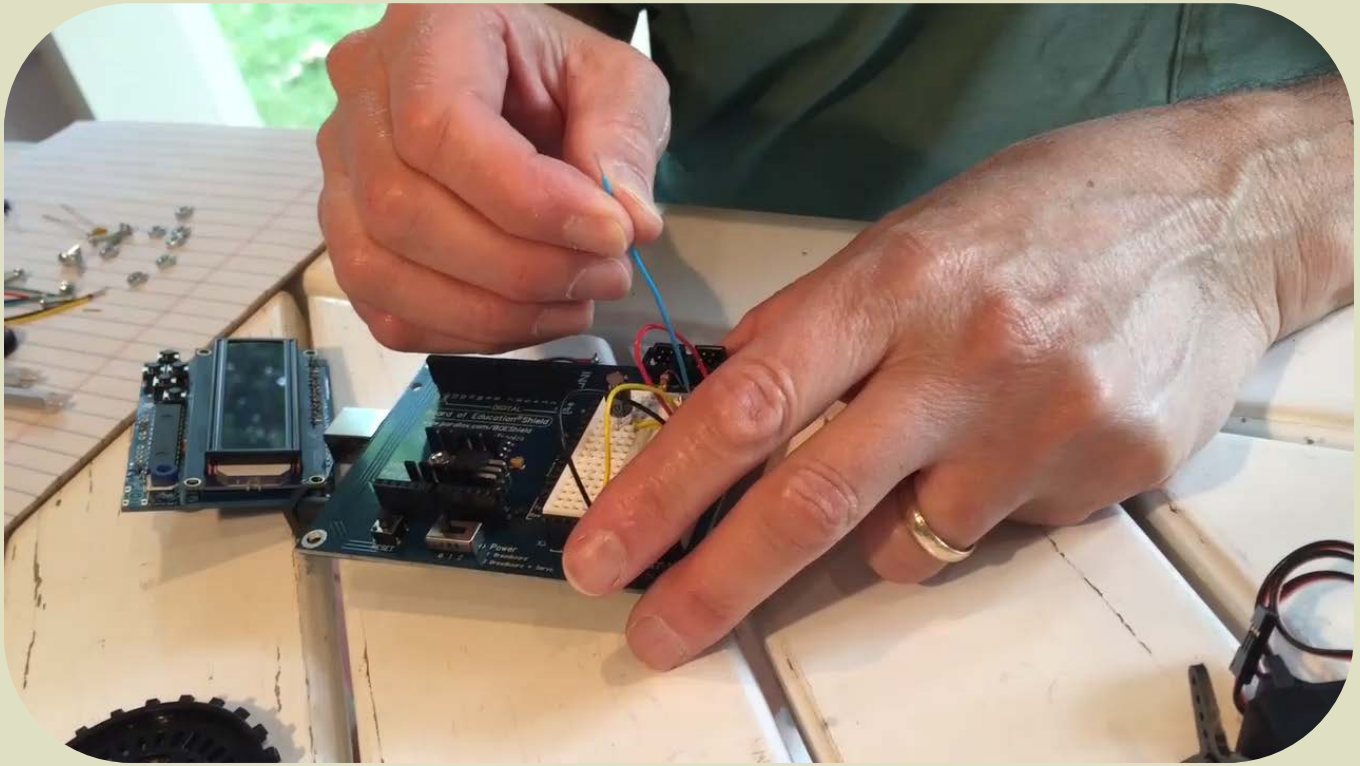
gizmo garden©

Components

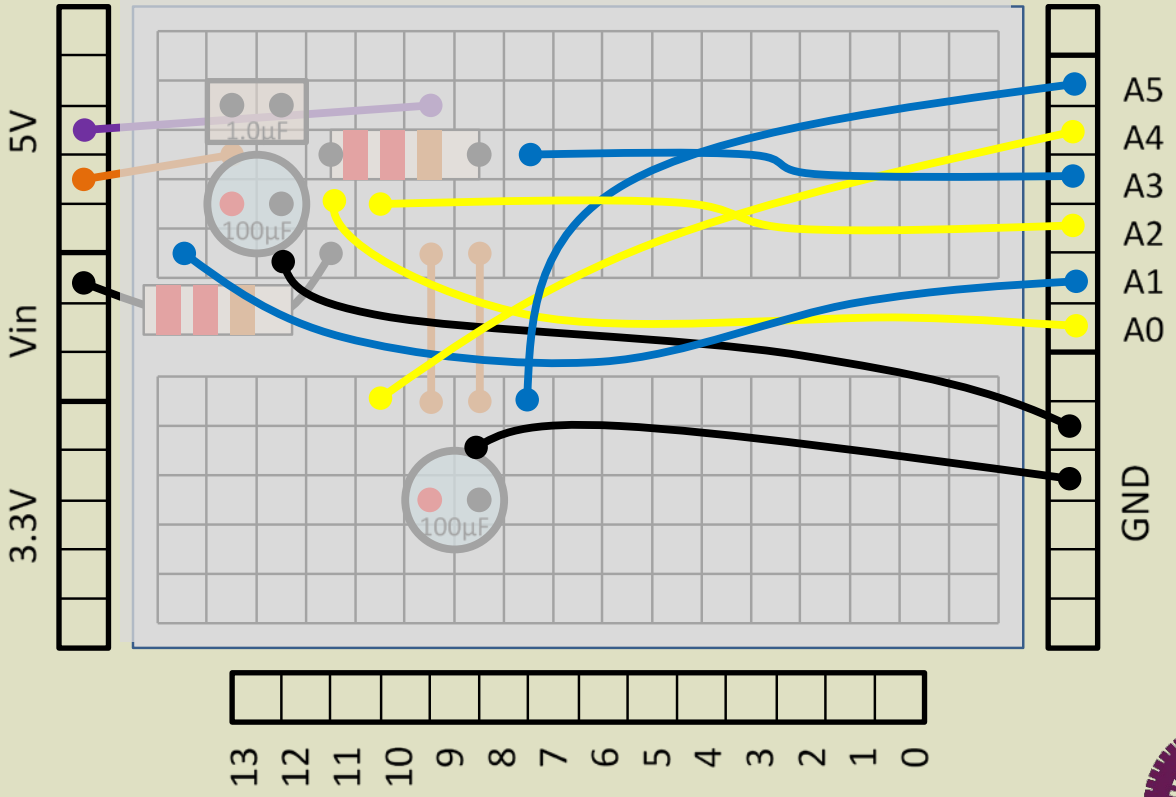


gizmo garden©

Long Wires

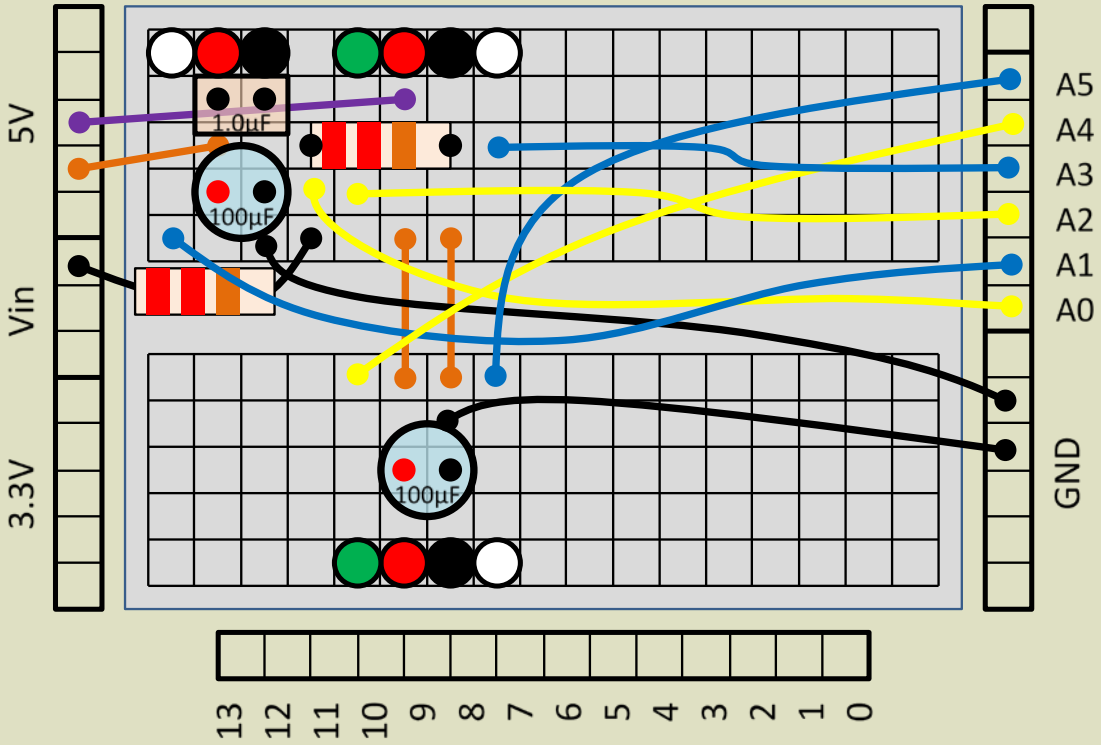


Long Wires

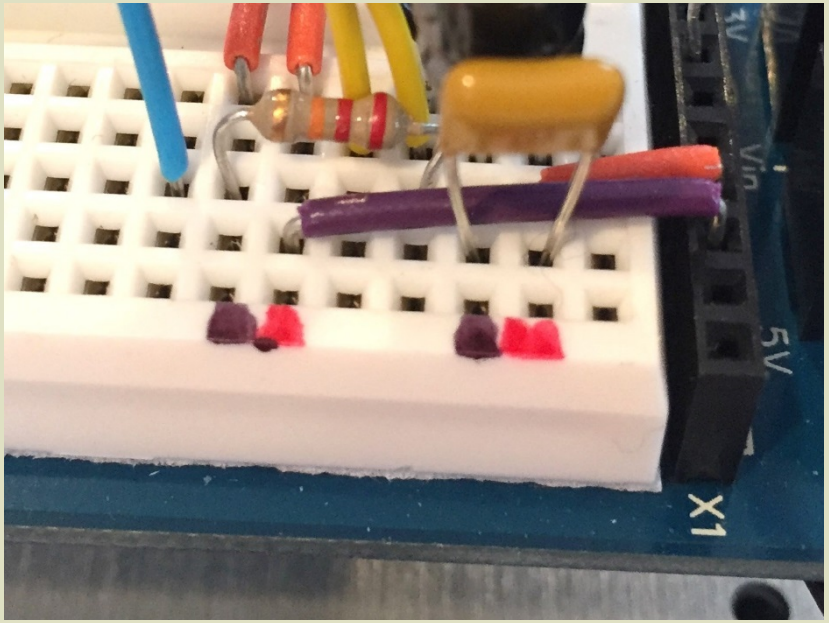


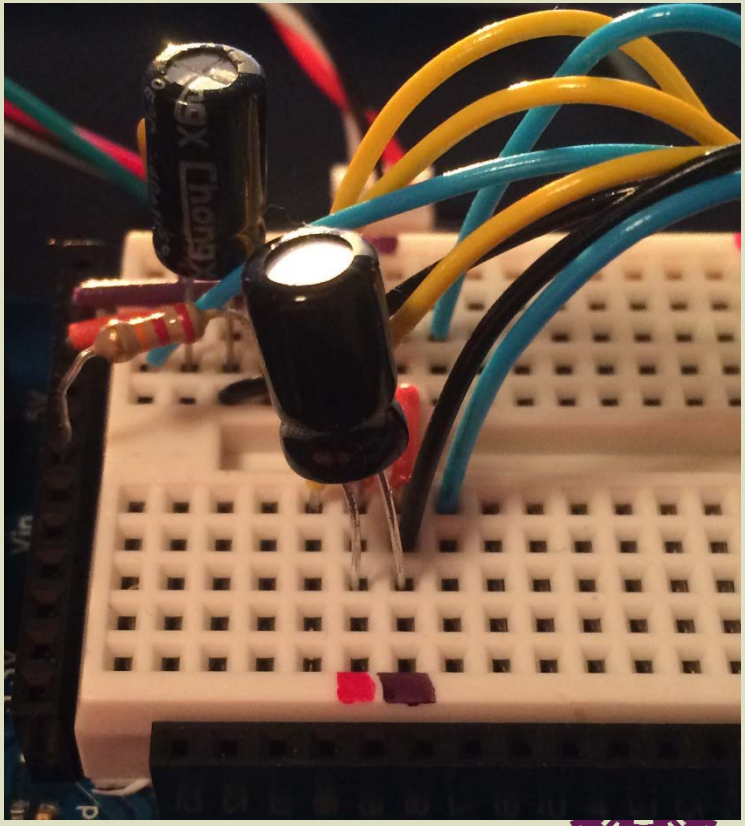
gizmo garden©

Connections



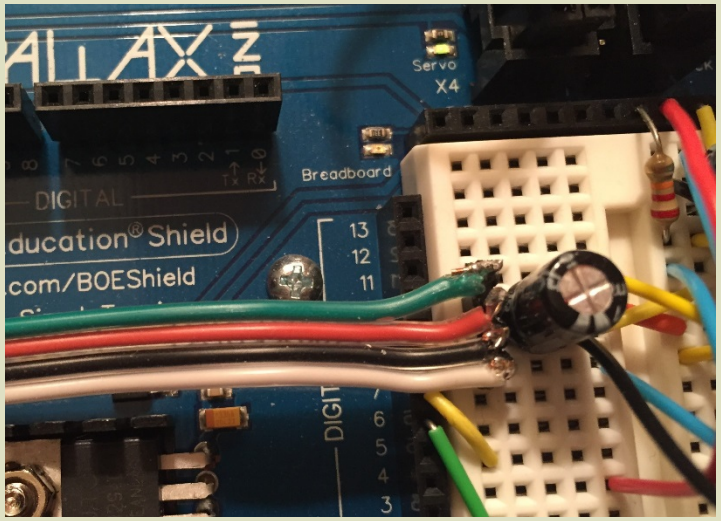
Mark Power and Ground

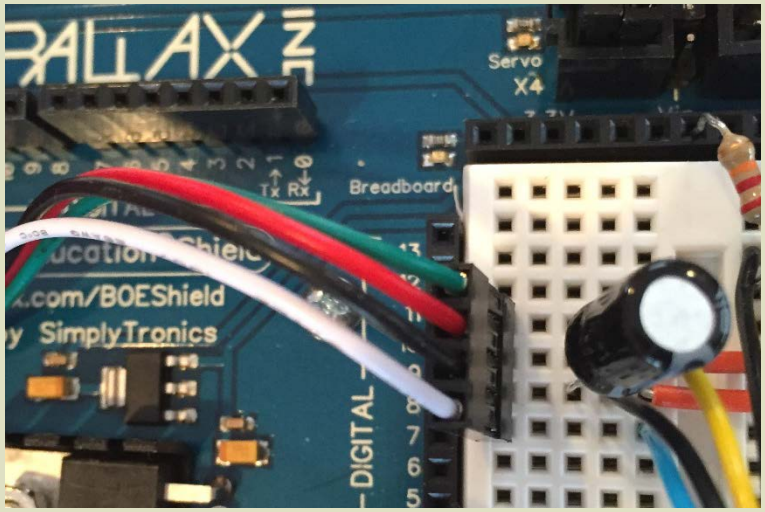




Remove Glasses

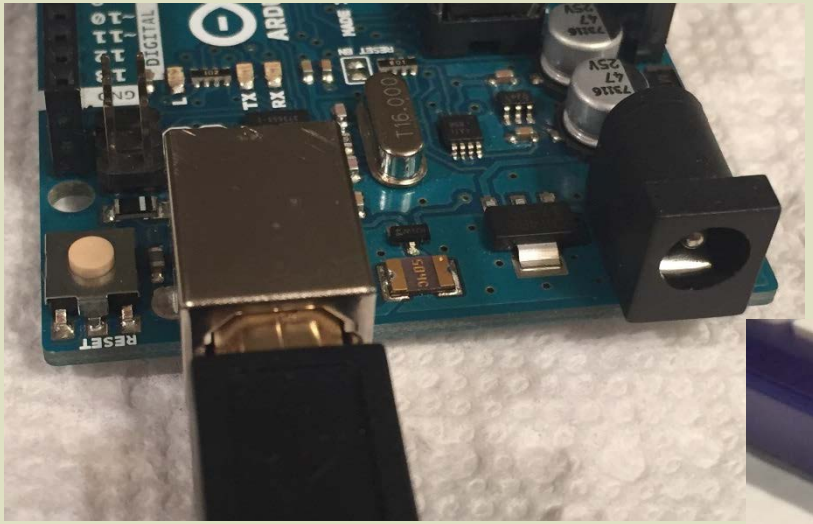
Connect Dashboard





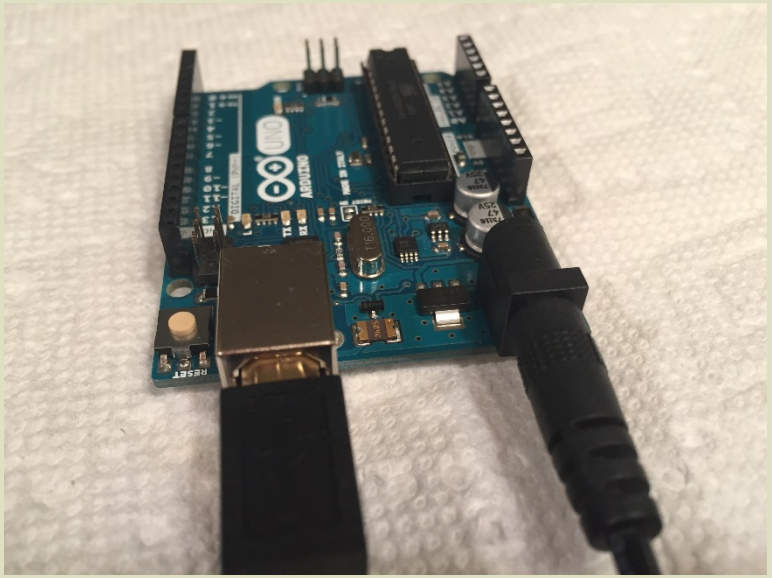
gizmo garden©

Connect Gizmo to Computer

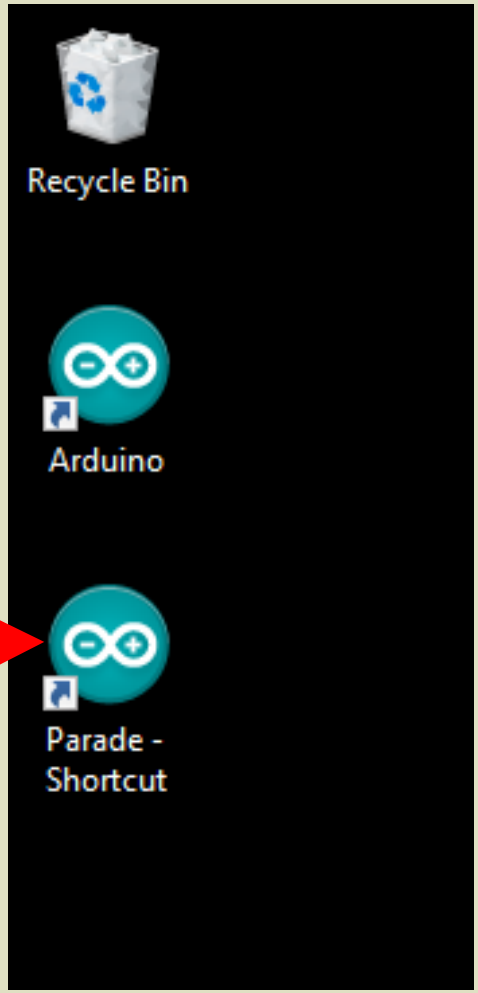


gizmo garden©

Plug in Gizmo

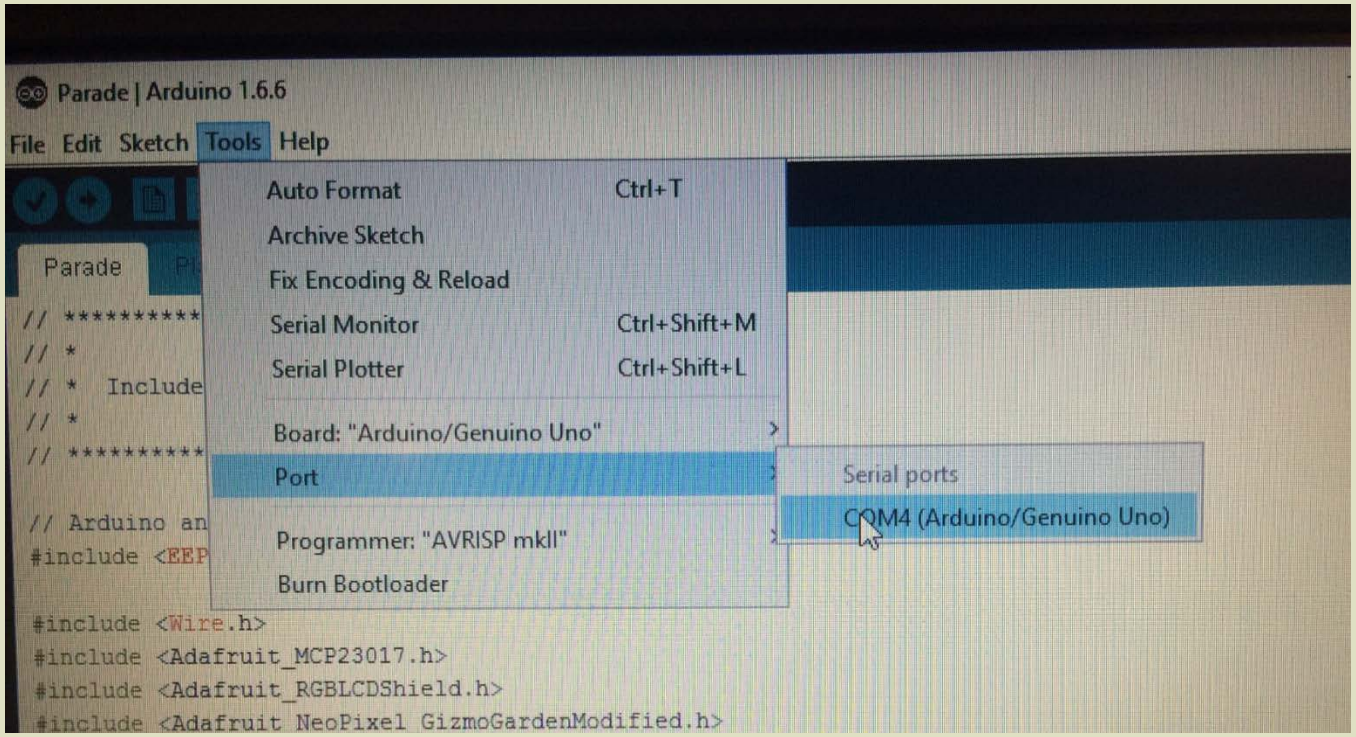


Open Arduino App



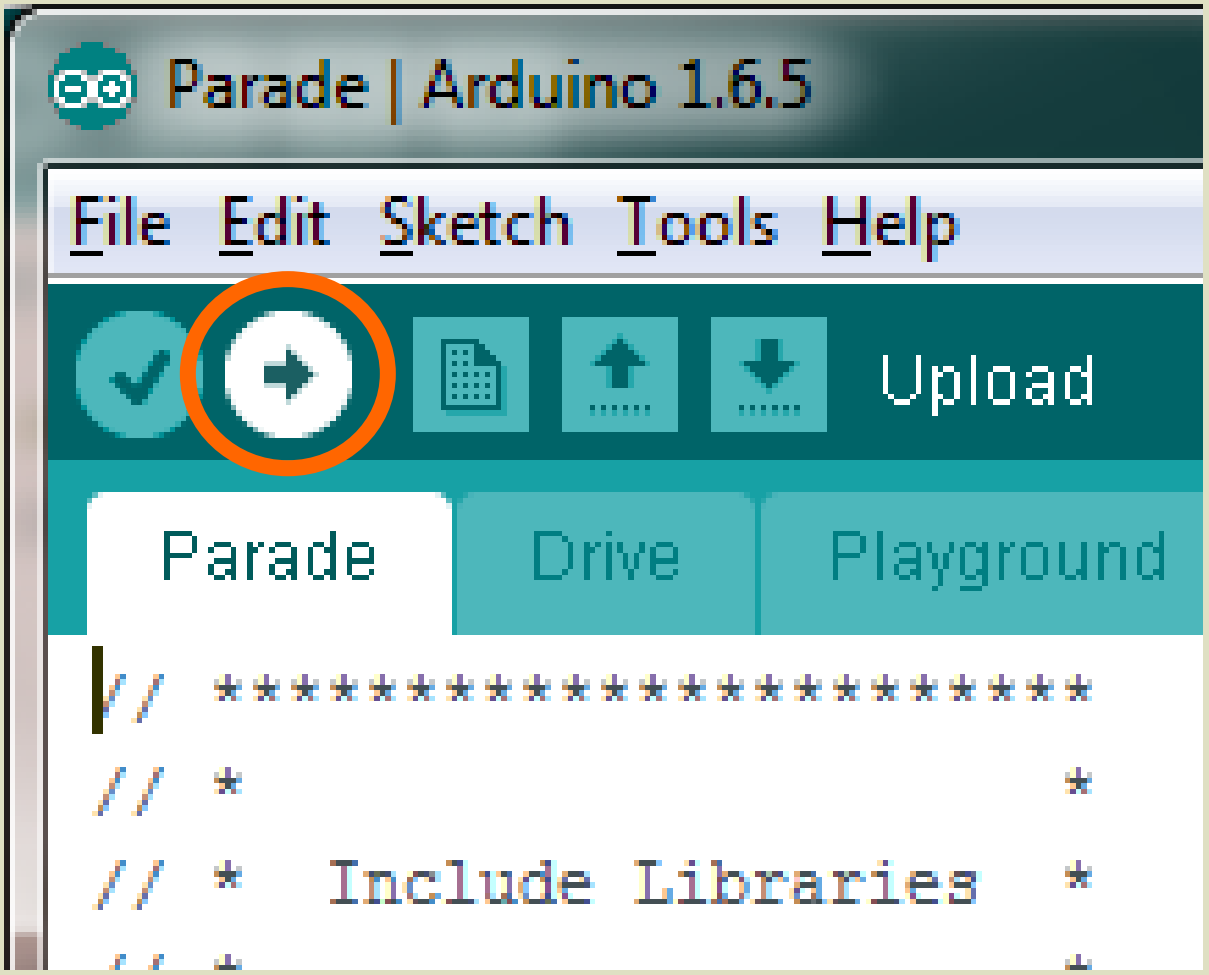
Double-click

Connect the Port



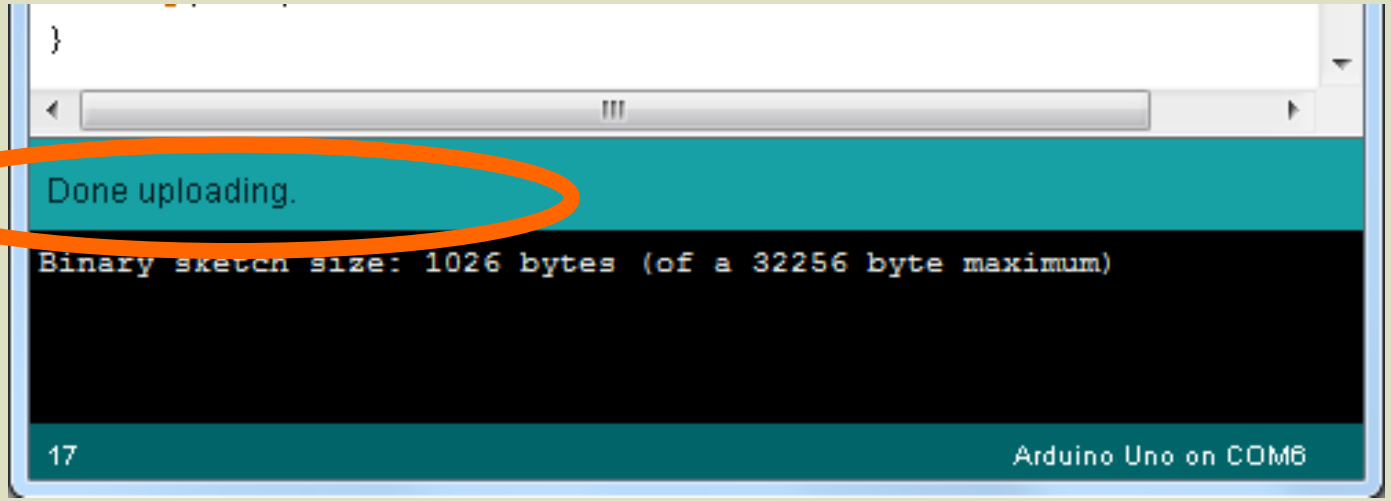
gizmo garden©

Upload Parade



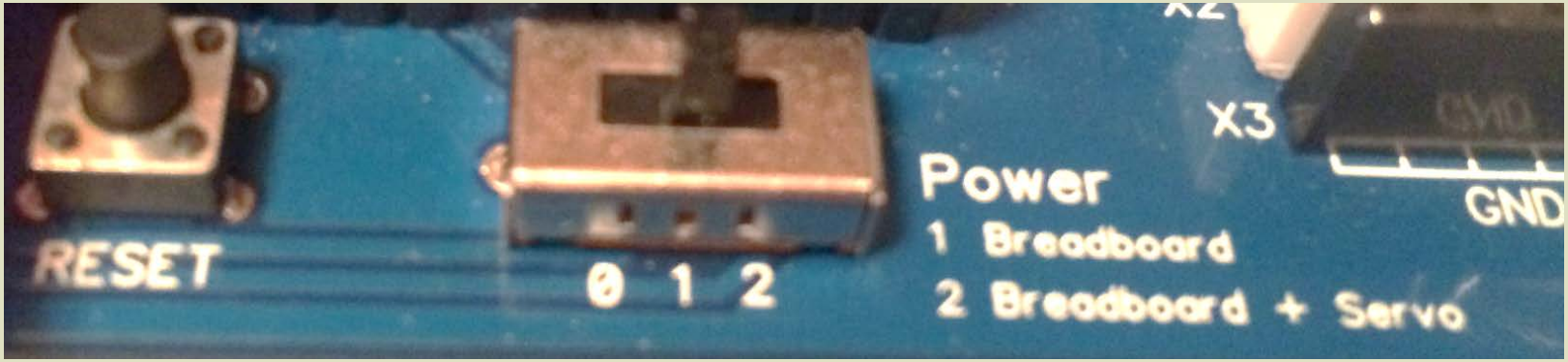
gizmo garden©

Look for “Done”



gizmo garden©

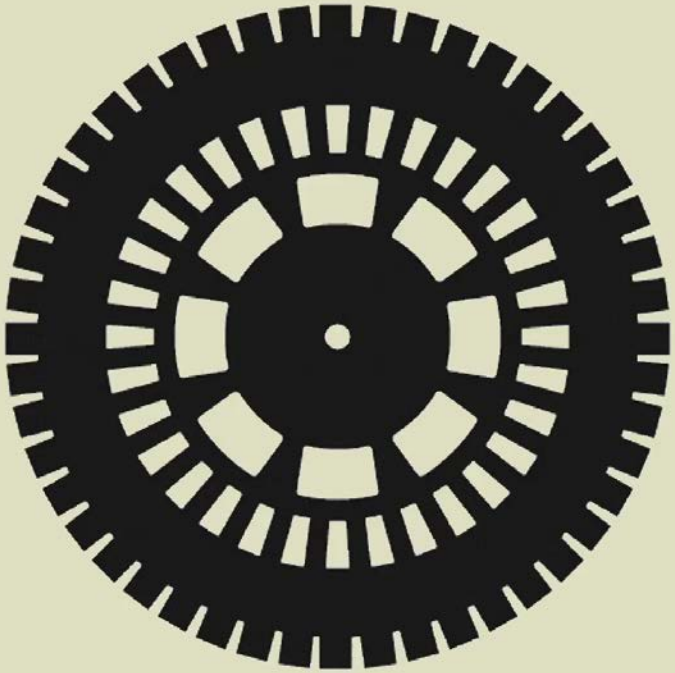
Switch Power & Push Reset



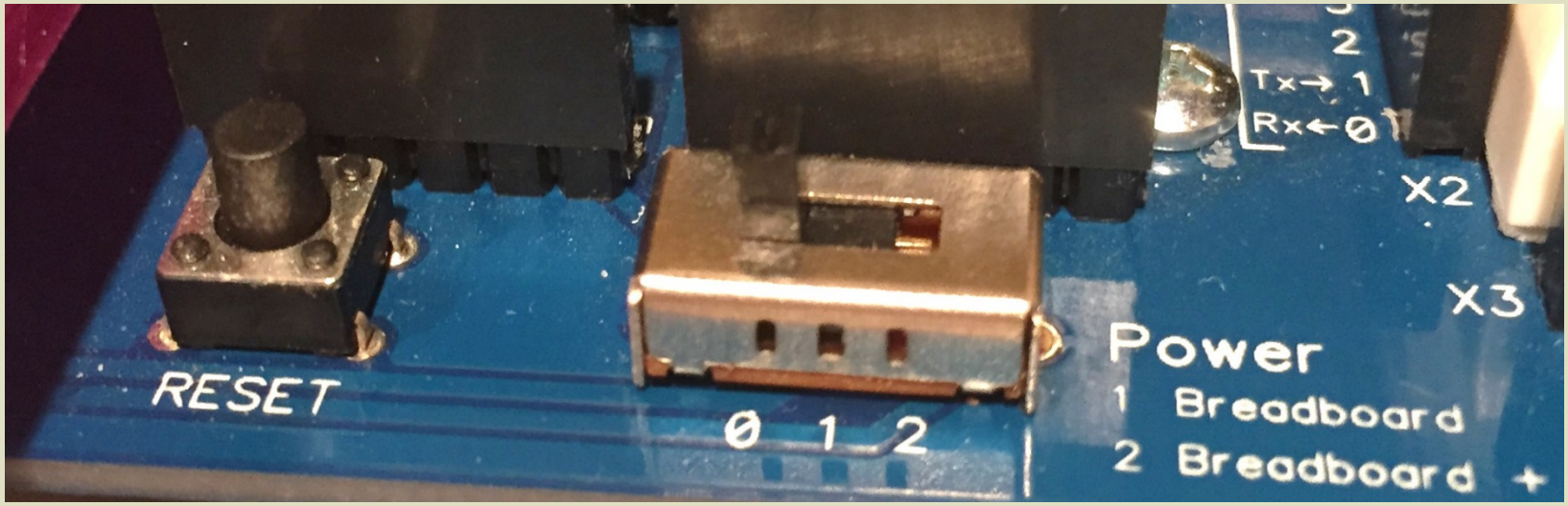
gizmo garden©

gizmo garden©

Well Done!

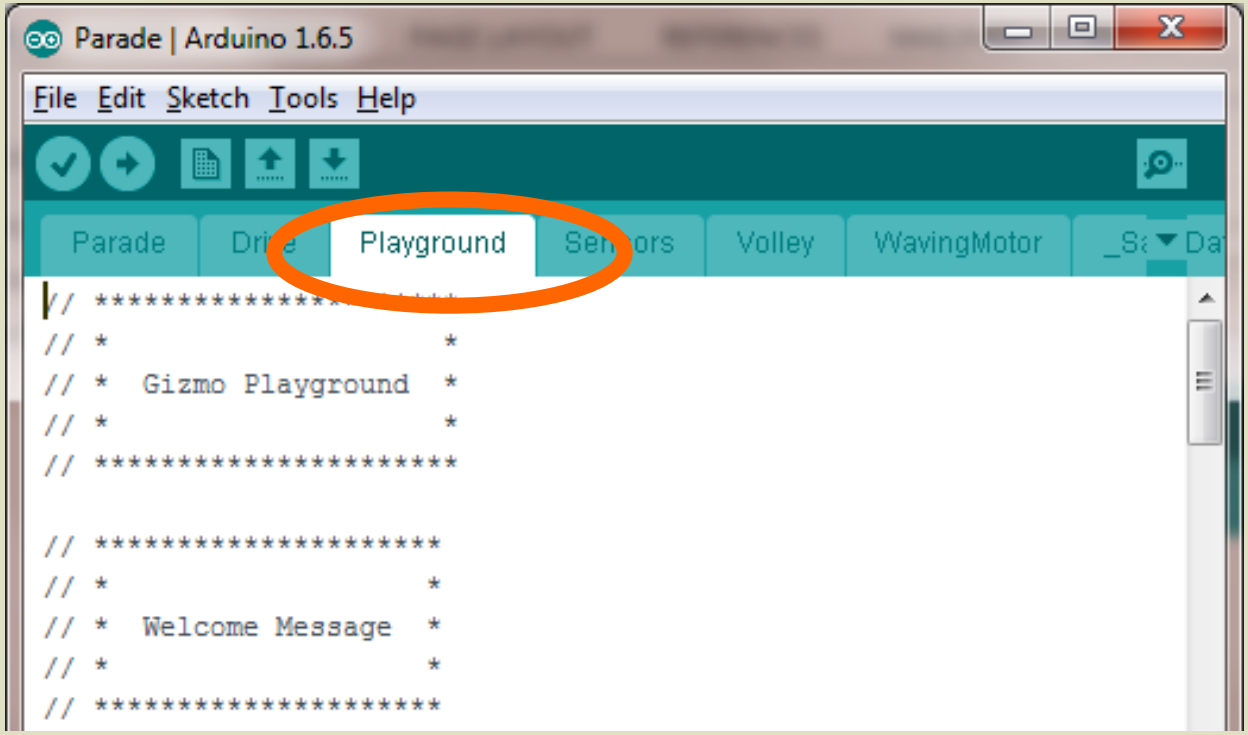


Power Off



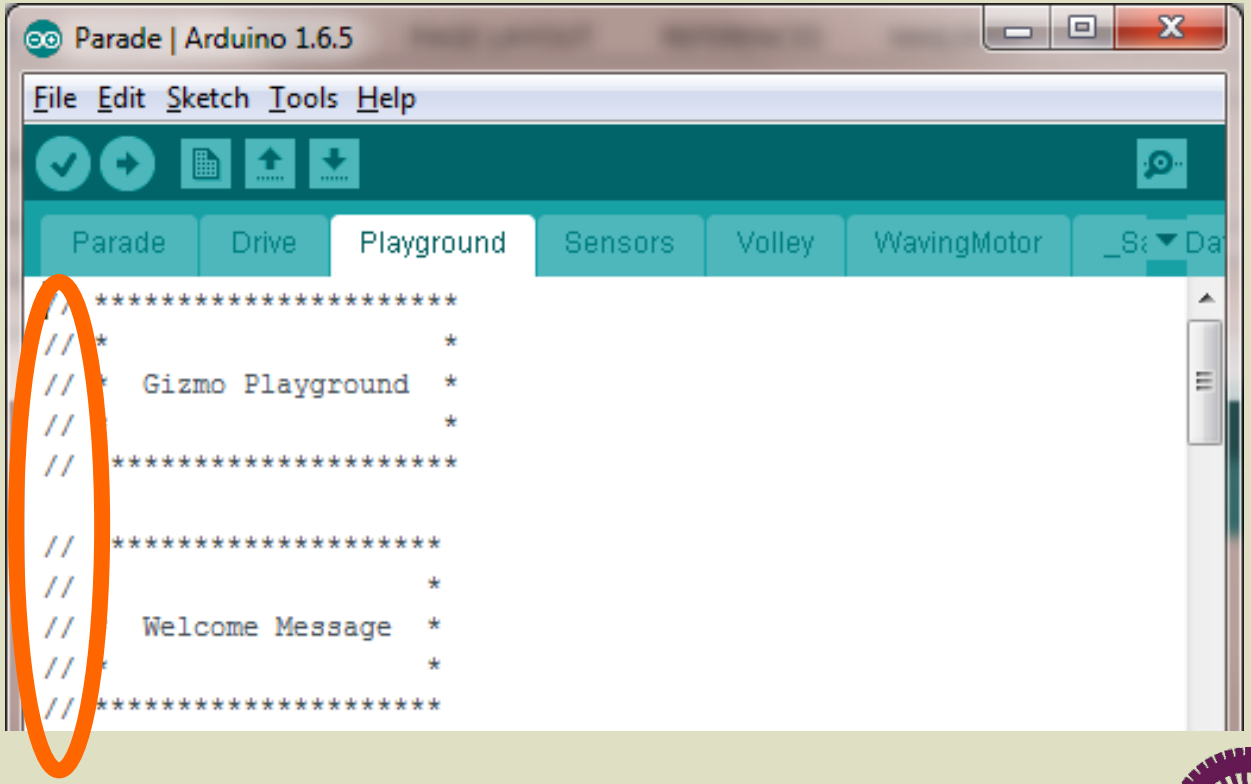
gizmo garden©

Playground Tab



gizmo garden©

Comments

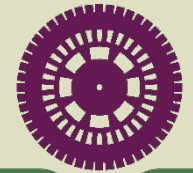


gizmo garden©

Code Reference #1

Comments

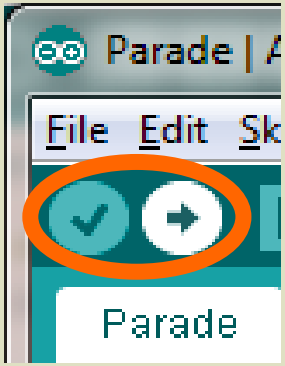
```
// Tells the compiler to ignore up to
// the end of the line

/* Tells the compiler to ignore up to
   a matching end marker */
```



What's a "Compiler"?

C++ code

```
void welcome()
{
  lcd.setCursor(4, 0);
  lcd.print(F("Welcome!"));
  lcd.setCursor(2, 1);
  lcd.print(F("Gizmo Garden"));
  delay(2000);
}
```

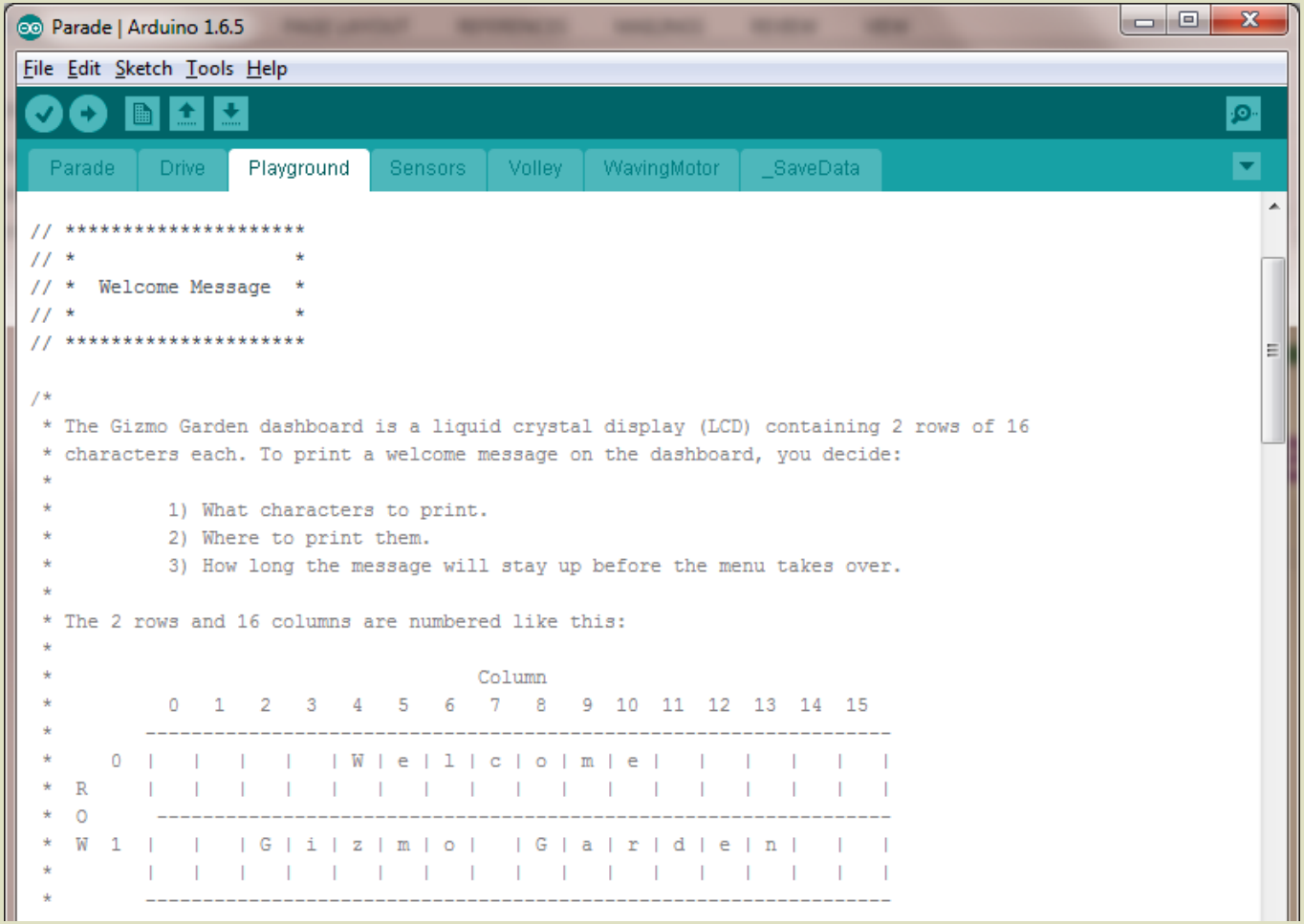


Machine Language

gizmo garden©

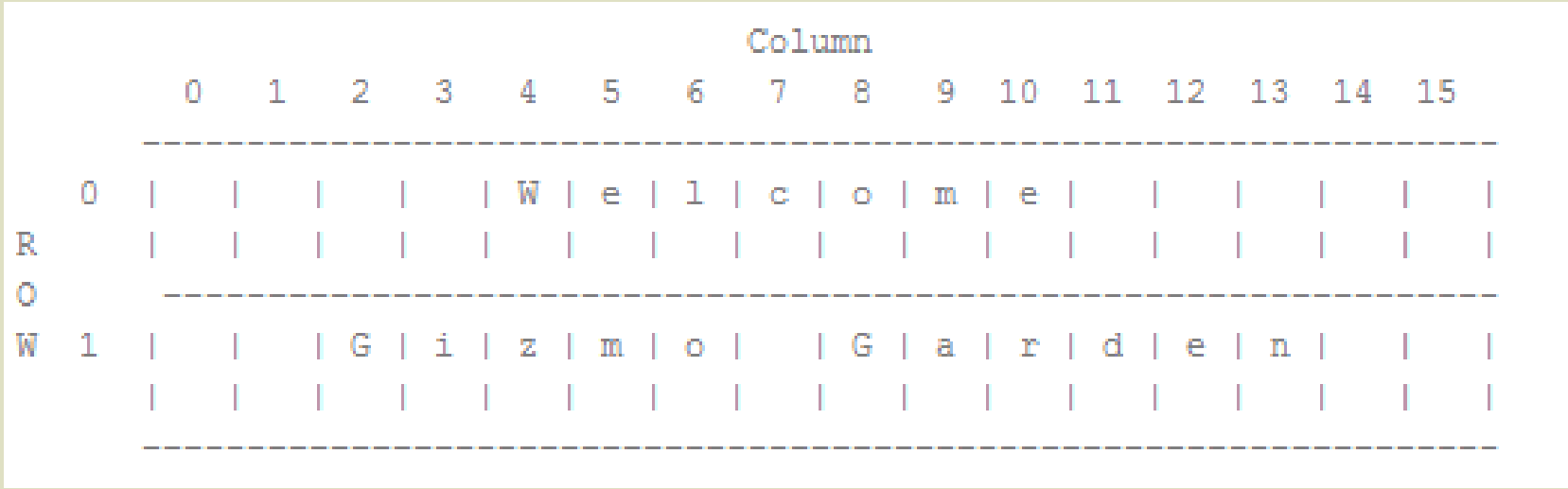
Amazing Grace

Rear Admiral Grace Hopper created the first compiler in 1952. She was known as "Amazing Grace".



gizmo garden©

The Dashboard
a Liquid Crystal Display (LCD)



gizmo garden©

Welcome Function

```
void welcome()
{
  lcd.setCursor(4, 0);
  lcd.print(F("Welcome!"));
  lcd.setCursor(2, 1);
  lcd.print(F("Gizmo Garden"));
  delay(2000);
}
```

Welcome Function

```
void welcome()
{
  lcd.setCursor(4, 0);
  lcd.print(F("Welcome!"));
  lcd.setCursor(2, 1);
  lcd.print(F("Gizmo Garden"));
  delay(2000);
}
```

Welcome Function

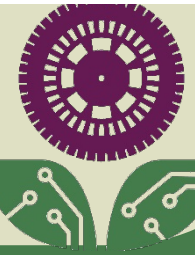
```
void welcome()
{
  lcd.setCursor(4, 0);
  lcd.print(F("Welcome!"));
  lcd.setCursor(2, 1);
  lcd.print(F("Gizmo Garden"));
  delay(2000);
}
```

gizmo garden©

Welcome Function

Statements: actions to take in the order given

```
void welcome()
{
  lcd.setCursor(4, 0);
  lcd.print(F("Welcome!"));
  lcd.setCursor(2, 1);
  lcd.print(F("Gizmo Garden"));
  delay(2000);
}
```

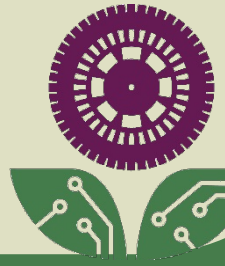


Code Reference #2

Function Format

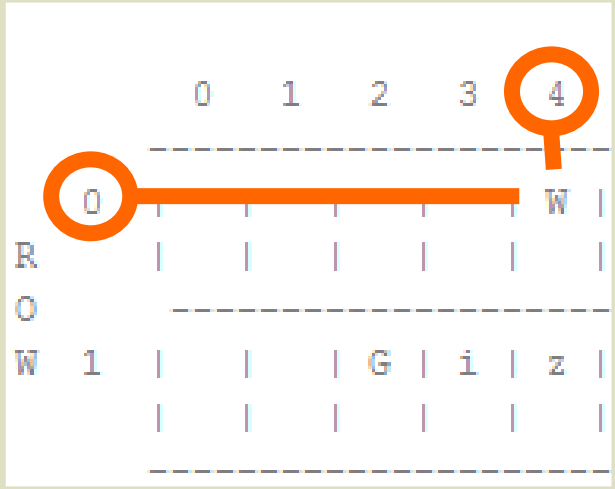
- For function type with no input/output:

```
void functionName ()
{
   statements
}
```



Welcome Function

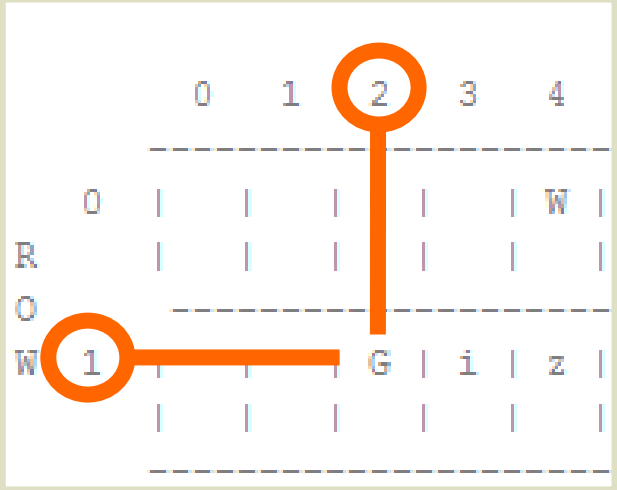
```
void welcome()
{
  lcd.setCursor(4, 0);
  lcd.print(F("Welcome!"));
  lcd.setCursor(2, 1);
  lcd.print(F("Gizmo Garden"));
  delay(2000);
}
```



gizmo garden©

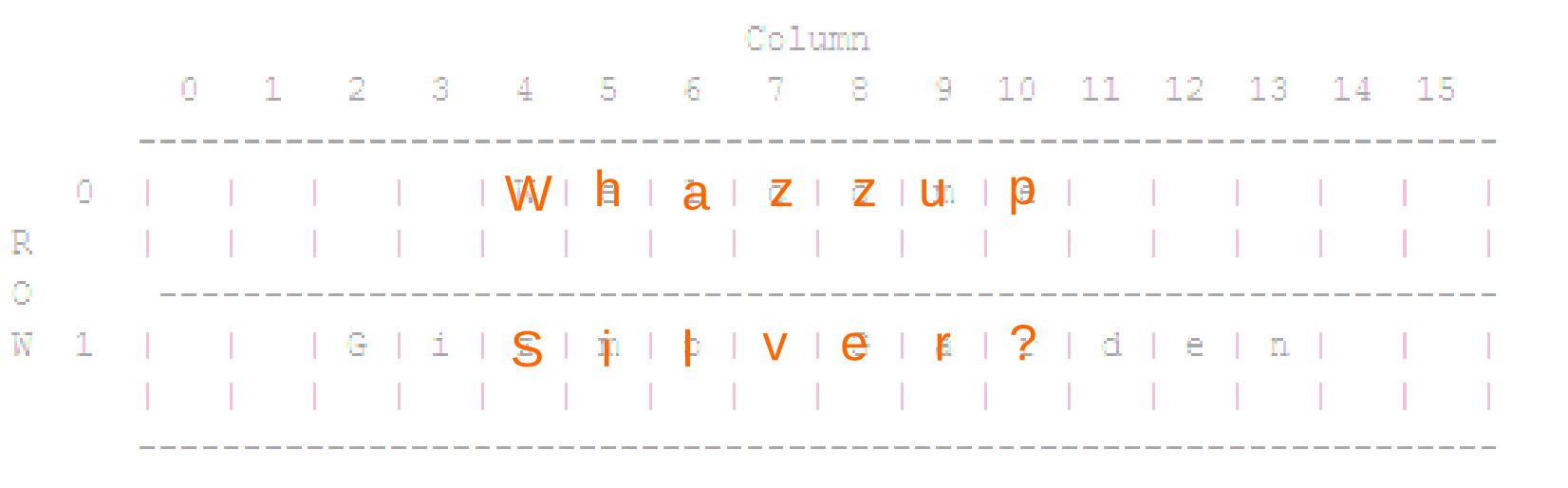
Welcome Function

```
void welcome()
{
  lcd.setCursor(4, 0);
  lcd.print(F("Welcome!"));
  lcd.setCursor(2, 1);
  lcd.print(F("Gizmo Garden"));
  delay(2000);
}
```



gizmo garden©

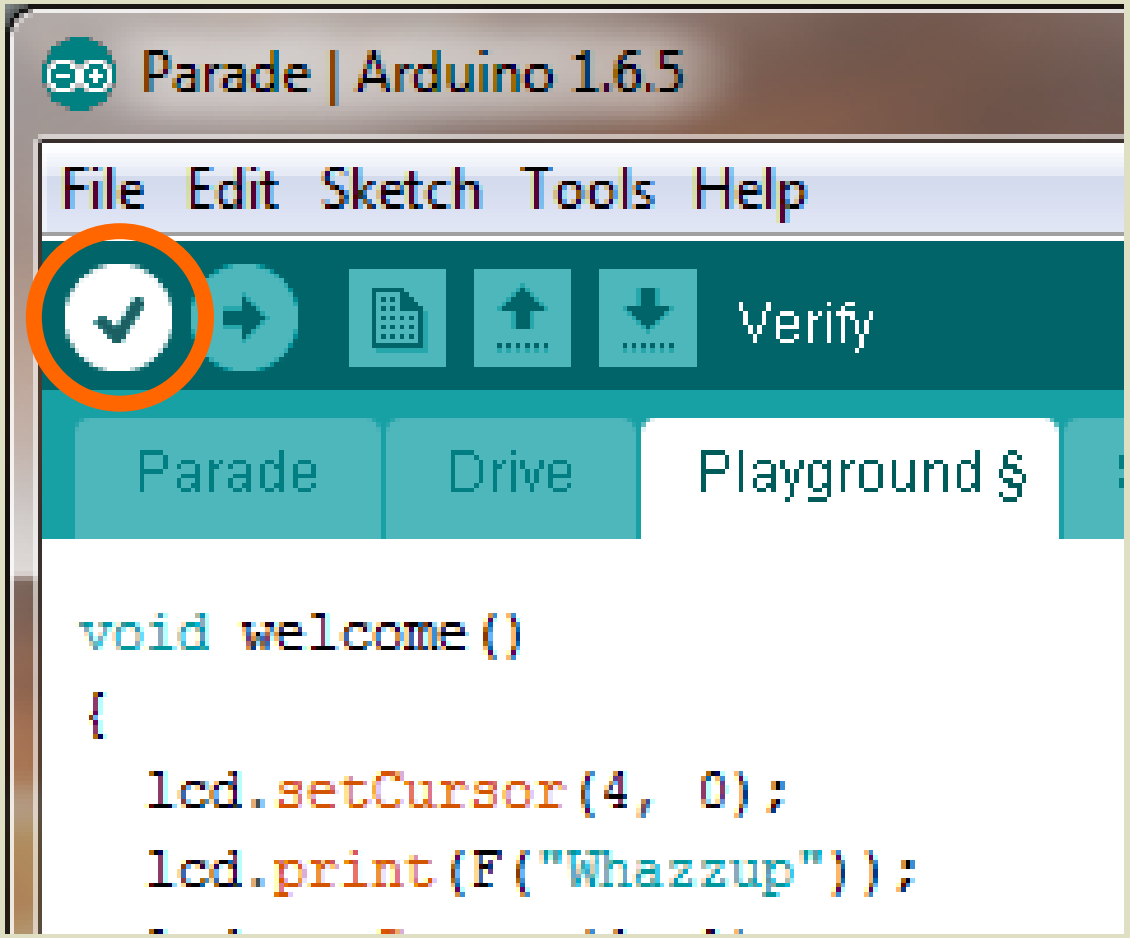
Your Message



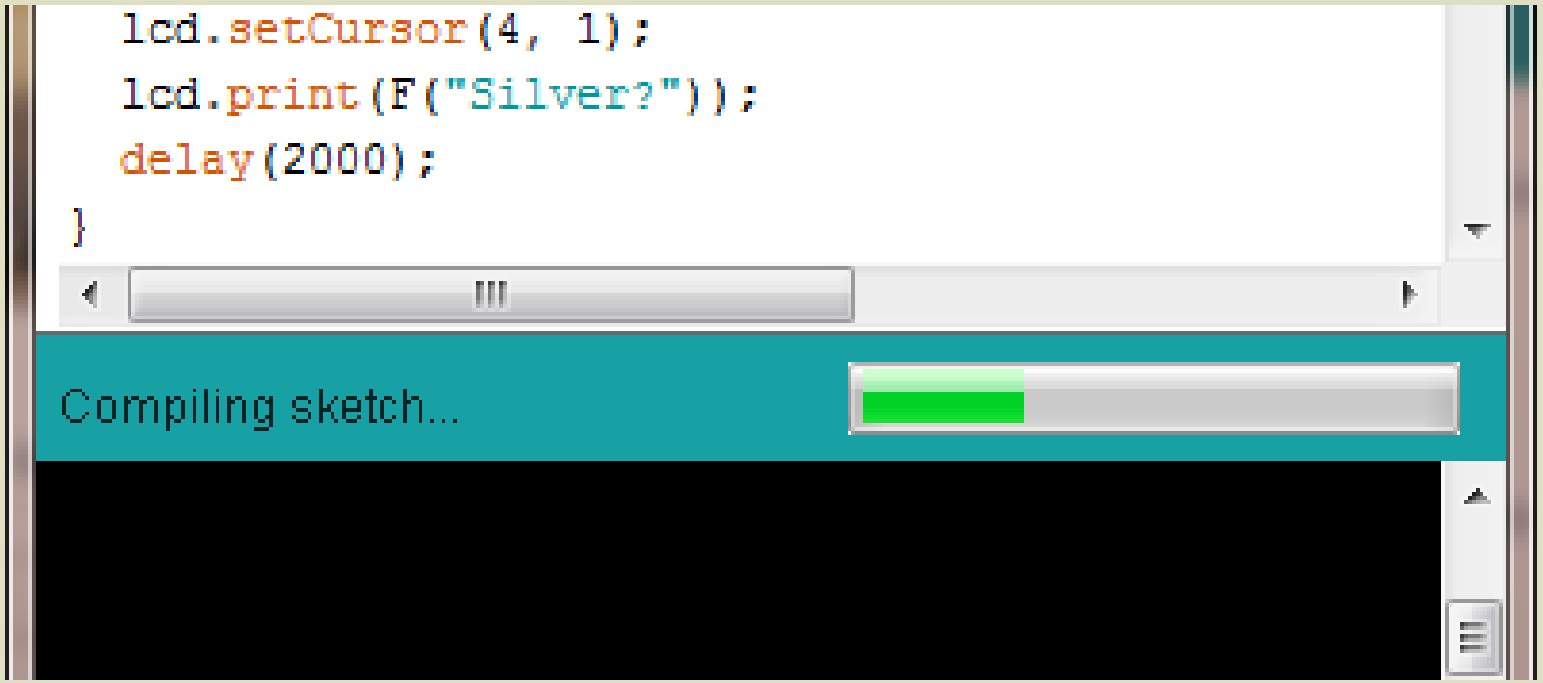

```
void welcome()
{
  lcd.setCursor(4, 0);
  lcd.print(F("Whazzup"));
  lcd.setCursor(4, 1);
  lcd.print(F("Silver?"));
  delay(2000);
}
```

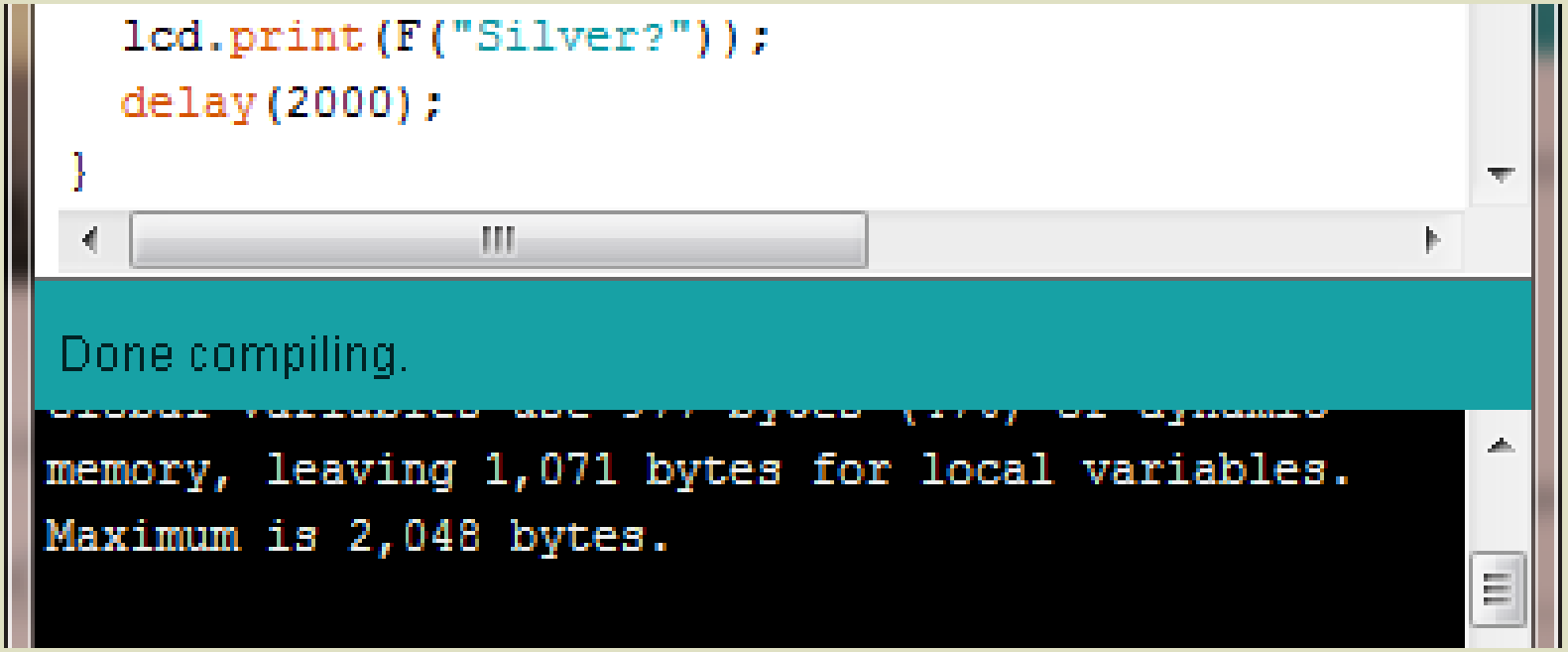

gizmo garden©

Verify

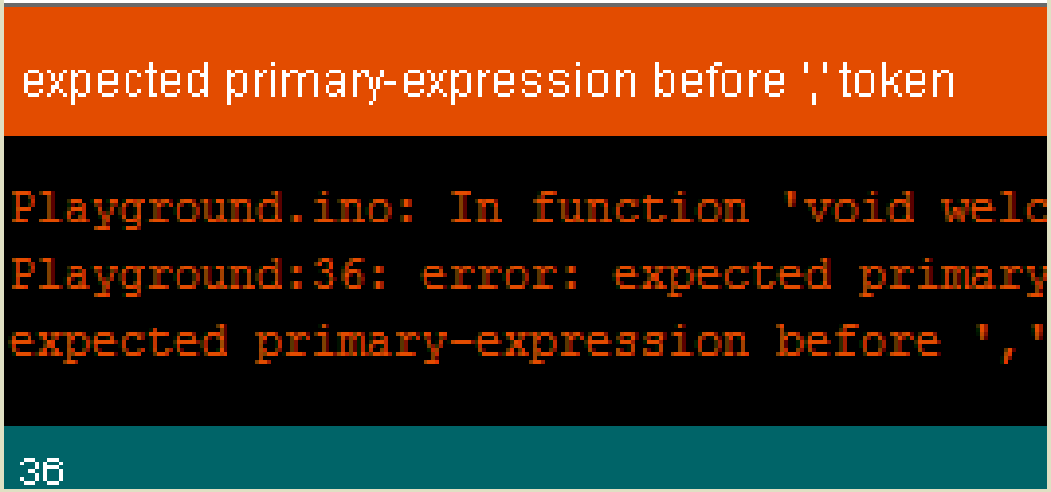


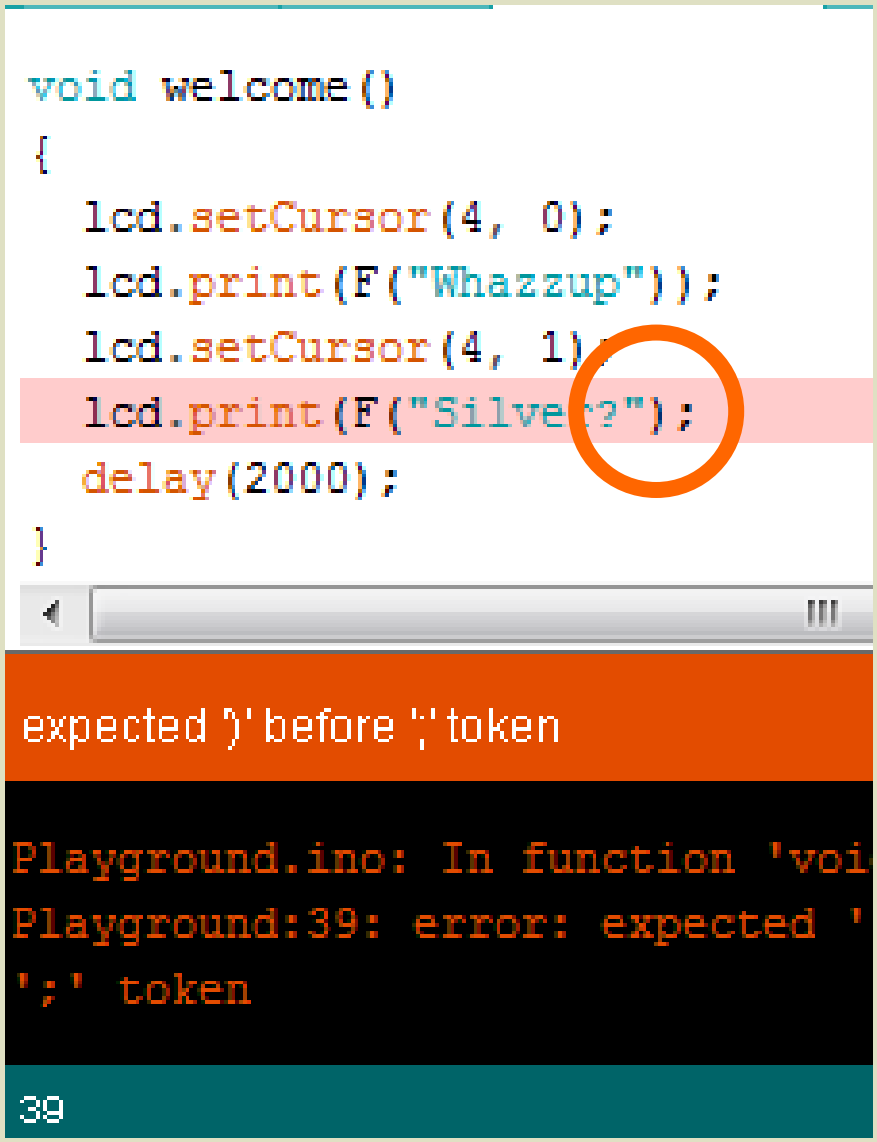
gizmo garden©

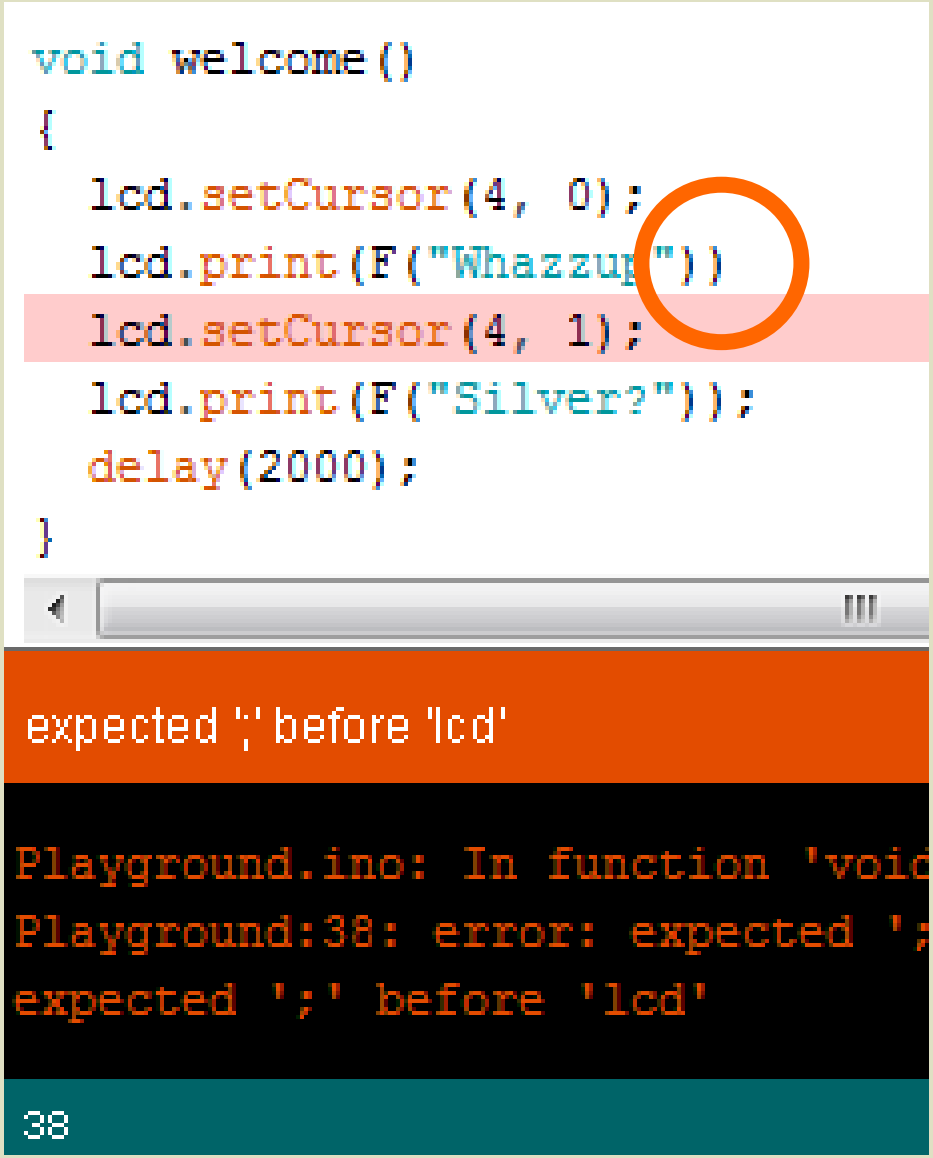




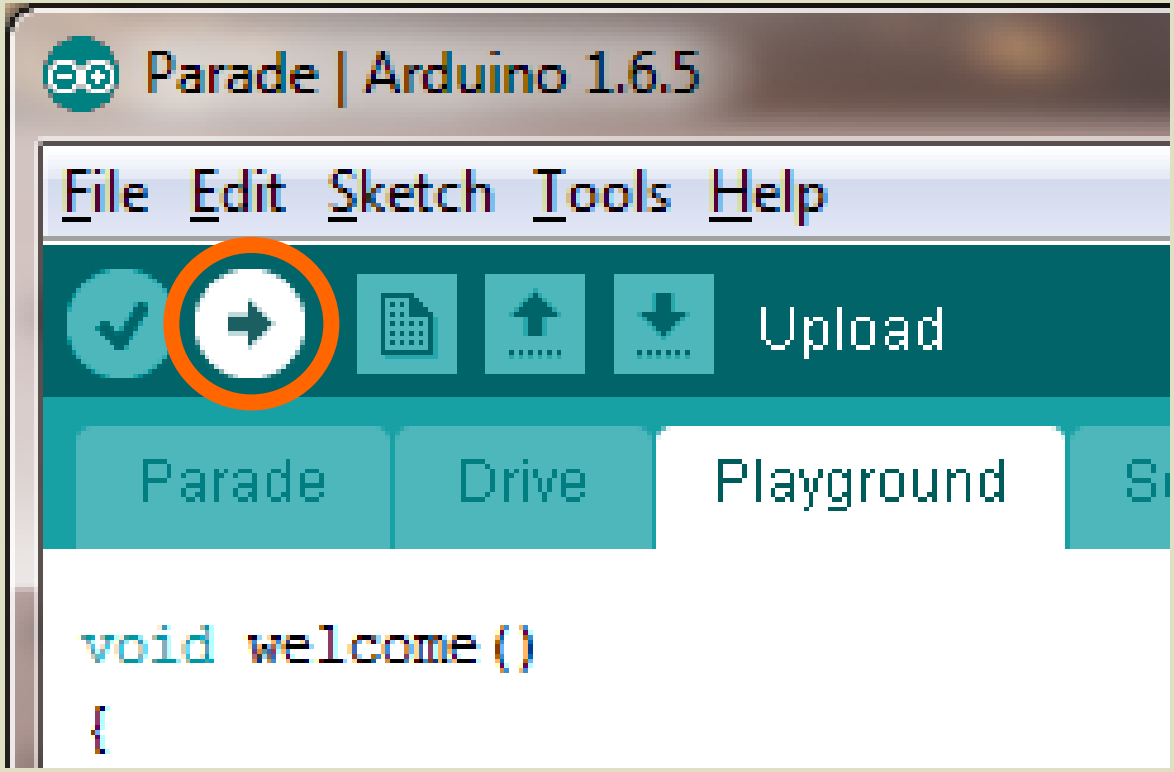
```
void welcome()
{
  lcd.setCursor(, 0);
  lcd.print(F("Whazzup"));
  lcd.setCursor(4, 1);
  lcd.print(F("Silver?"));
  delay(2000);
}
```





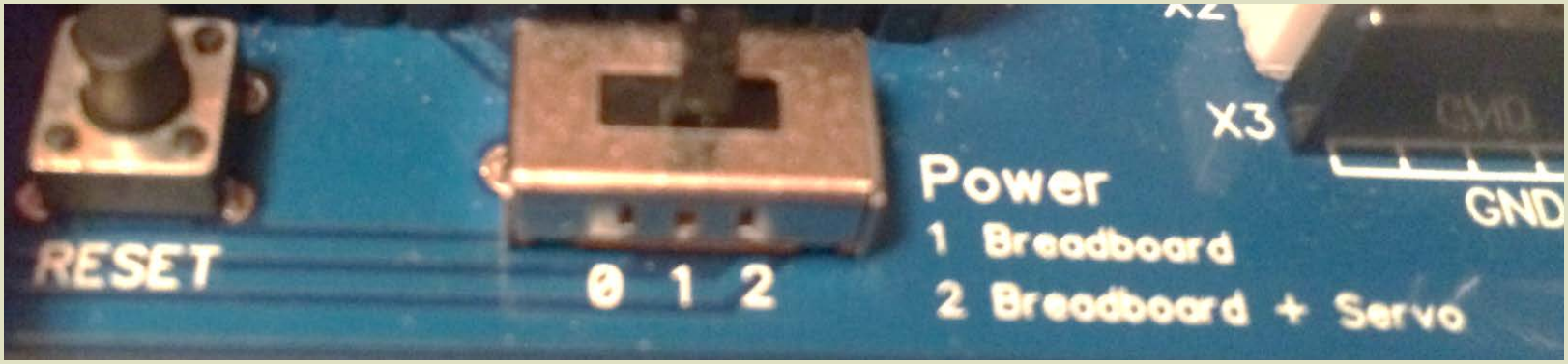


Upload



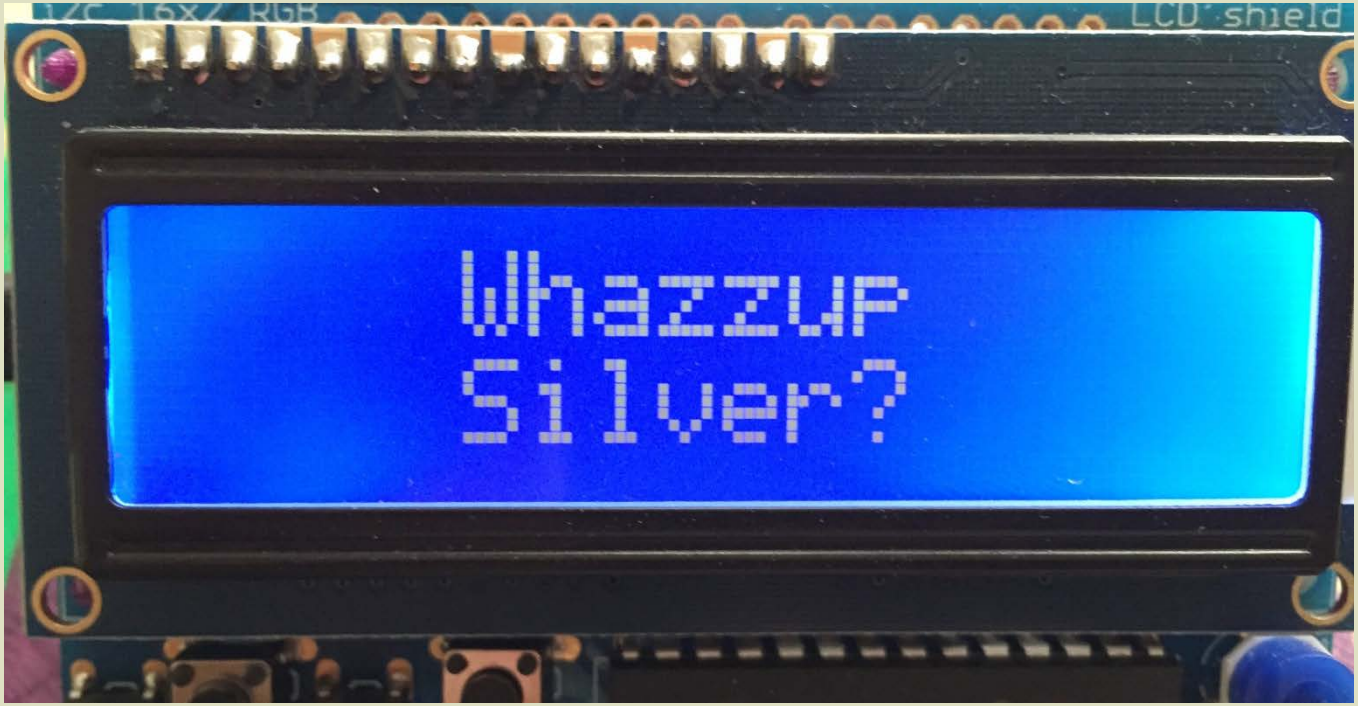
gizmo garden©

Power On & Reset



gizmo garden©

Your Message



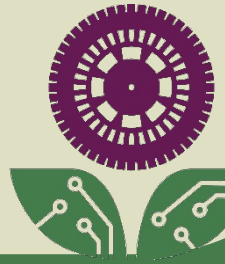
gizmo garden©

Code Reference #3

LCD Functions

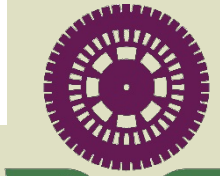
- Writes stuff to the dashboard

```
lcd.setCursor(column#, row#);
lcd.print(F("message"));
```



Delay

```
void welcome()
{
  lcd.setCursor(4, 0);
  lcd.print(F("Welcome!"));
  lcd.setCursor(2, 1);
  lcd.print(F("Gizmo Garden"));
  delay(2000);
}
```

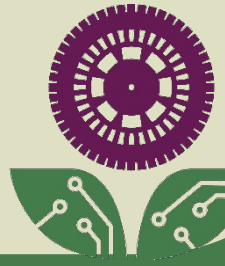


Code Reference #4

Delay

- Tells the Arduino to wait before doing the next thing

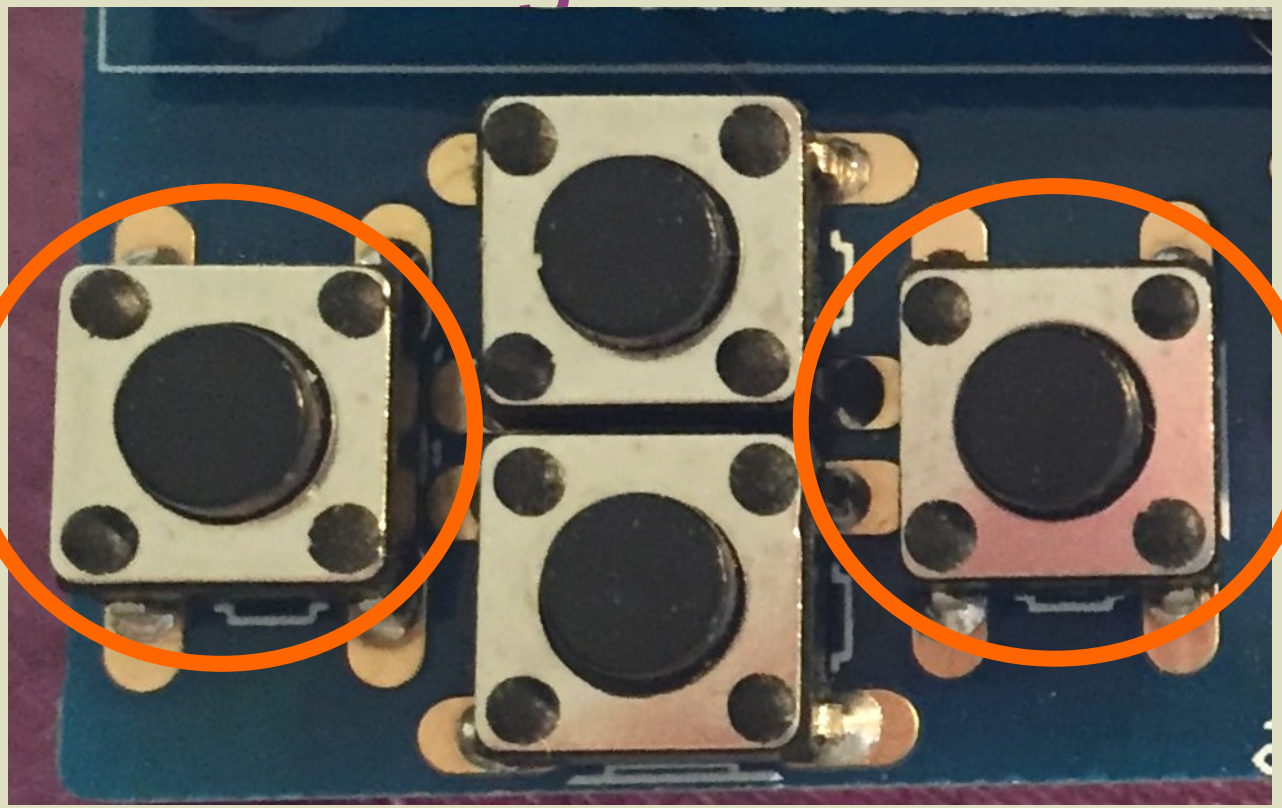
```
delay(milliseconds);
```



Dashboard Buttons

gizmo garden©

Left-Right Scrolls

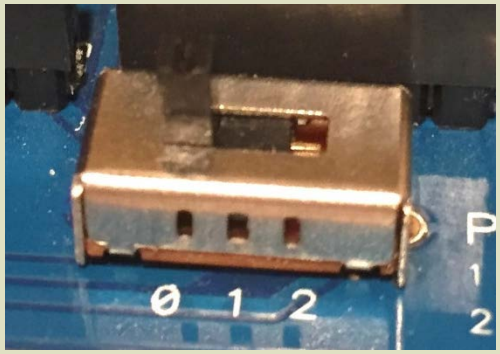


Horizontal Scroll

Monitoring a Sensor

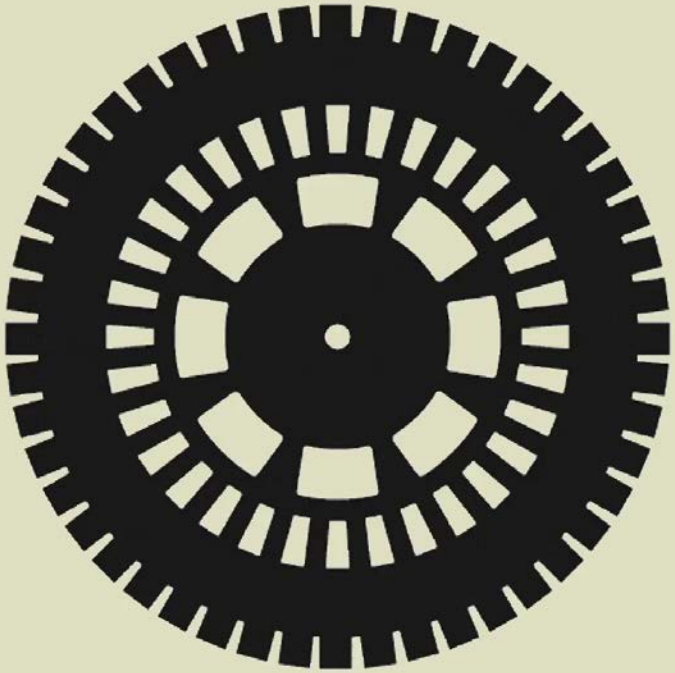
gizmo garden©

Power Off

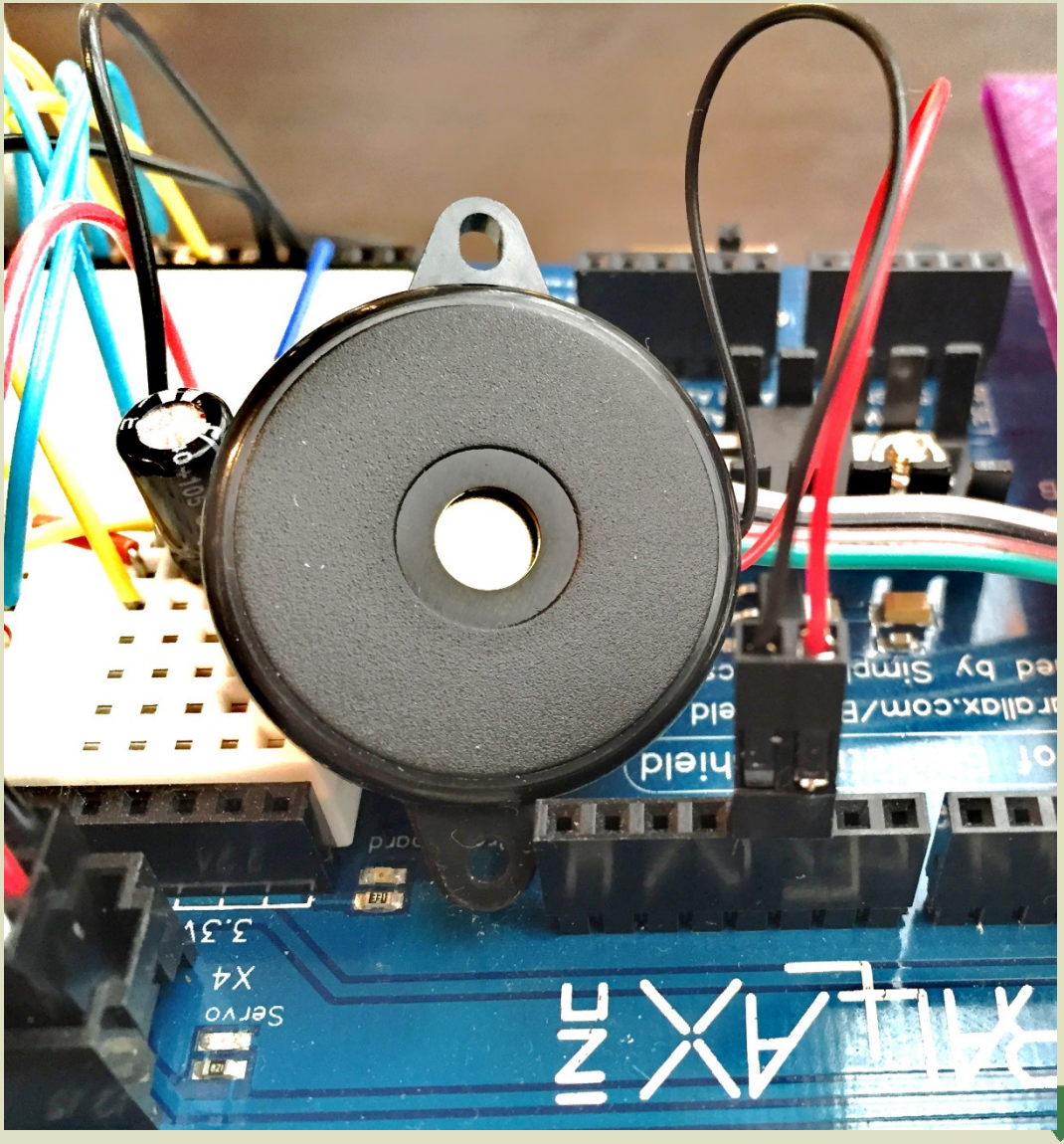


gizmo garden©

You Can Edit Code!

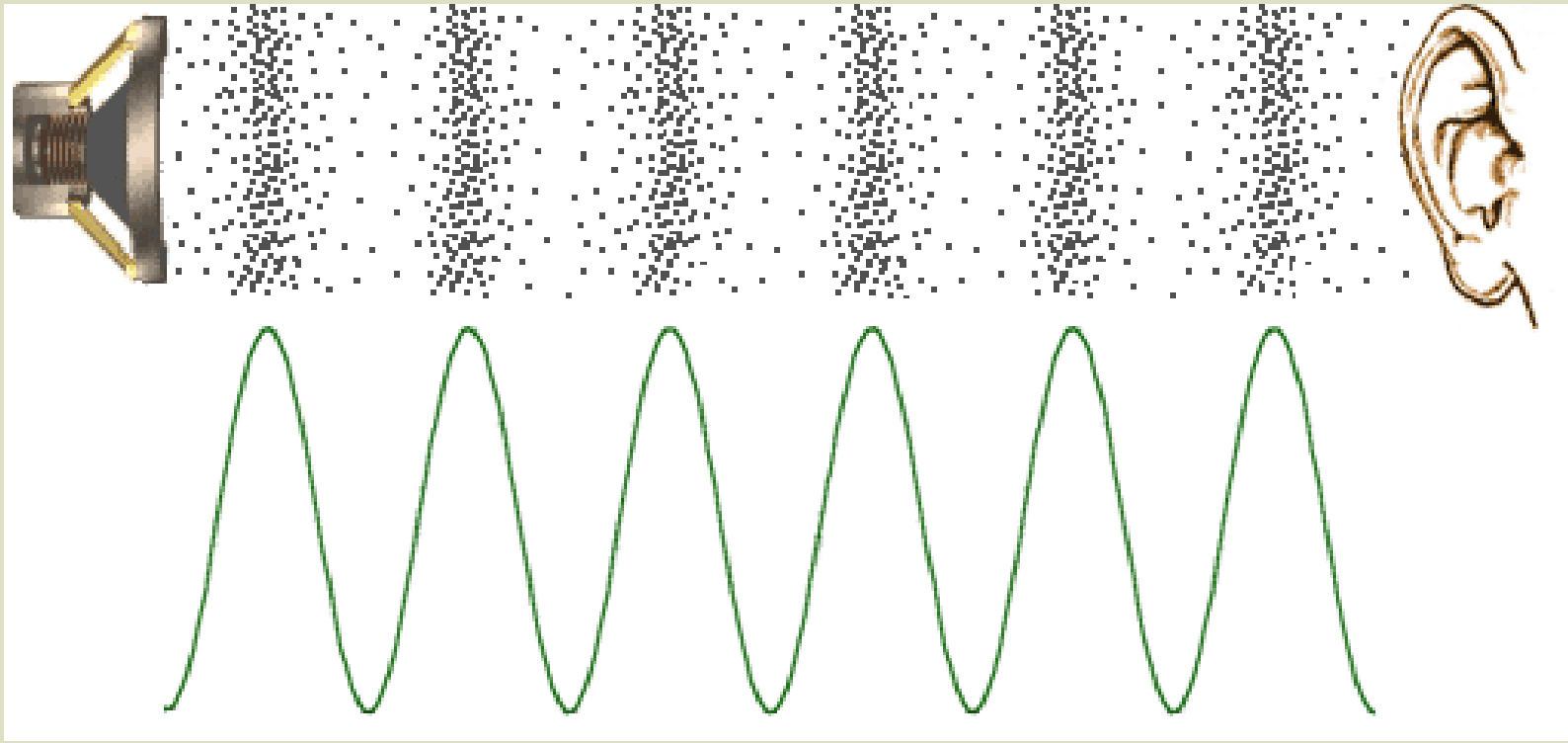


Musical Horns

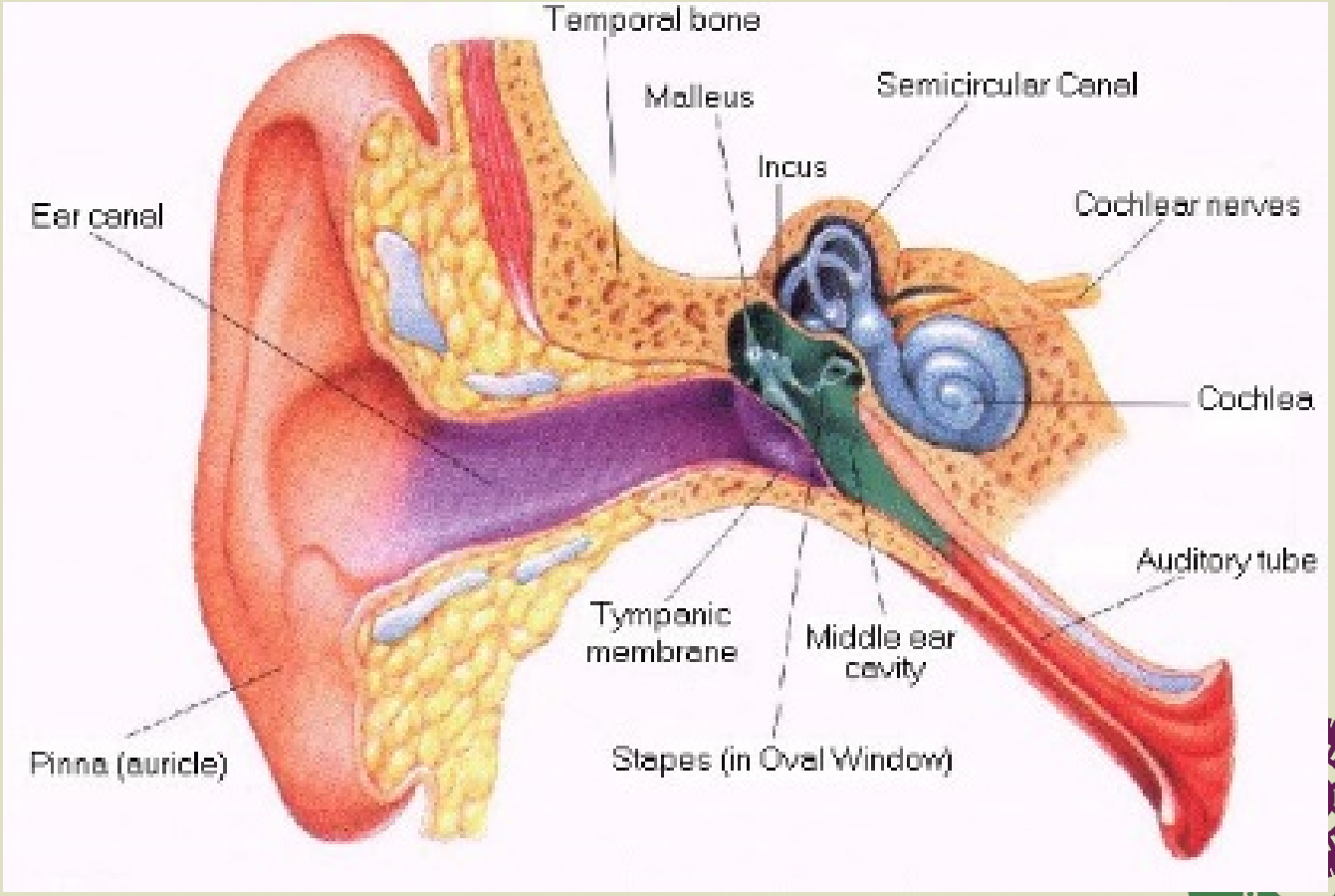


gizmo garden©

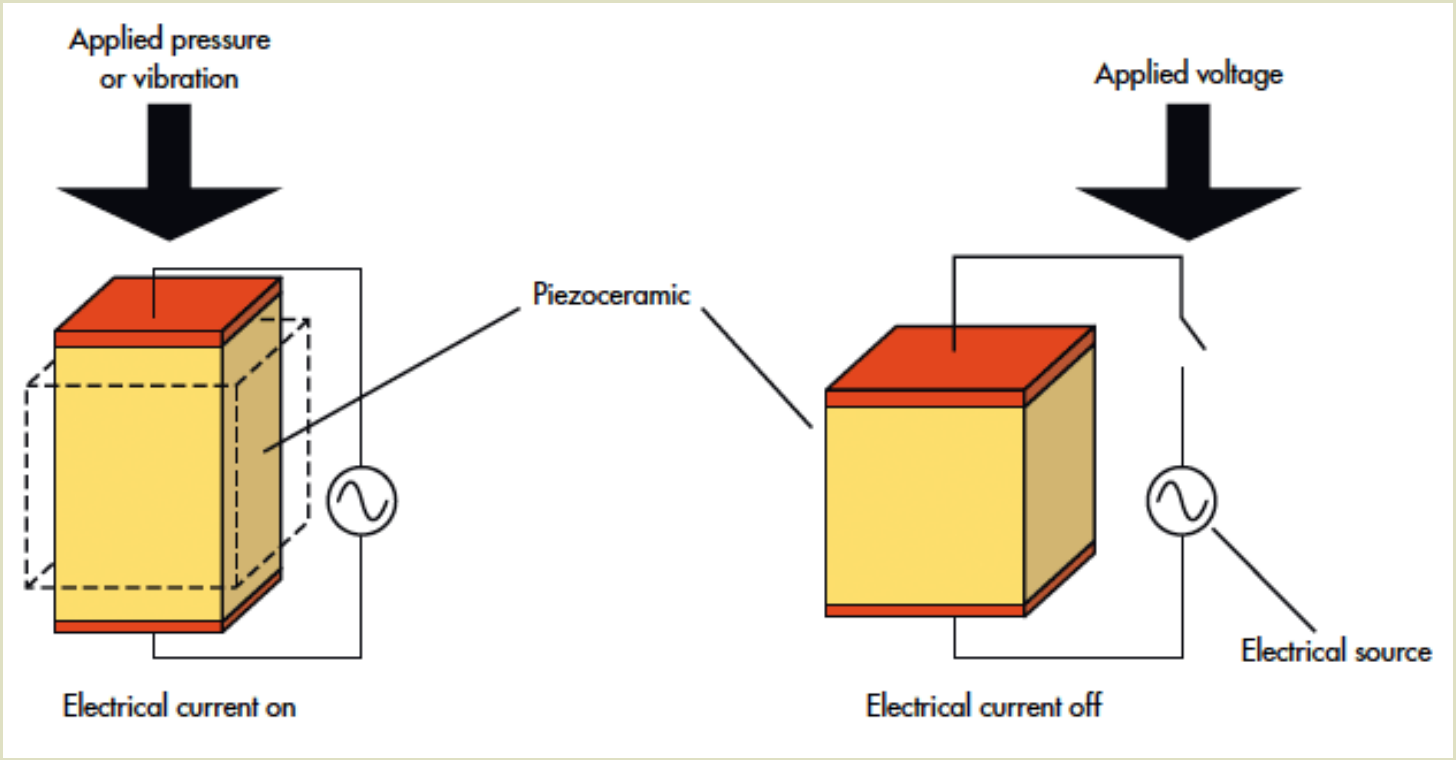
Sound



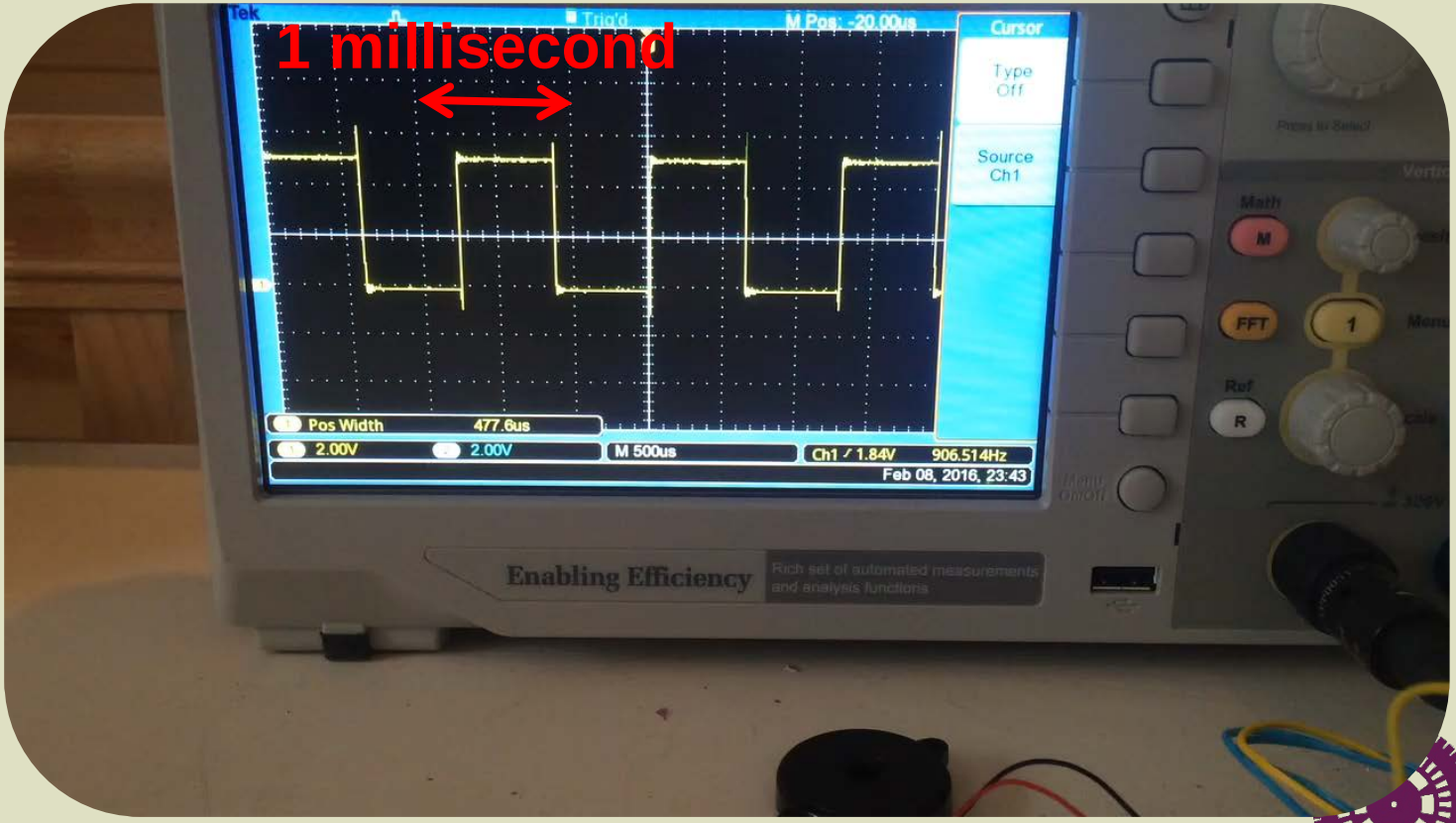
Ear converts sound waves to nerve signals



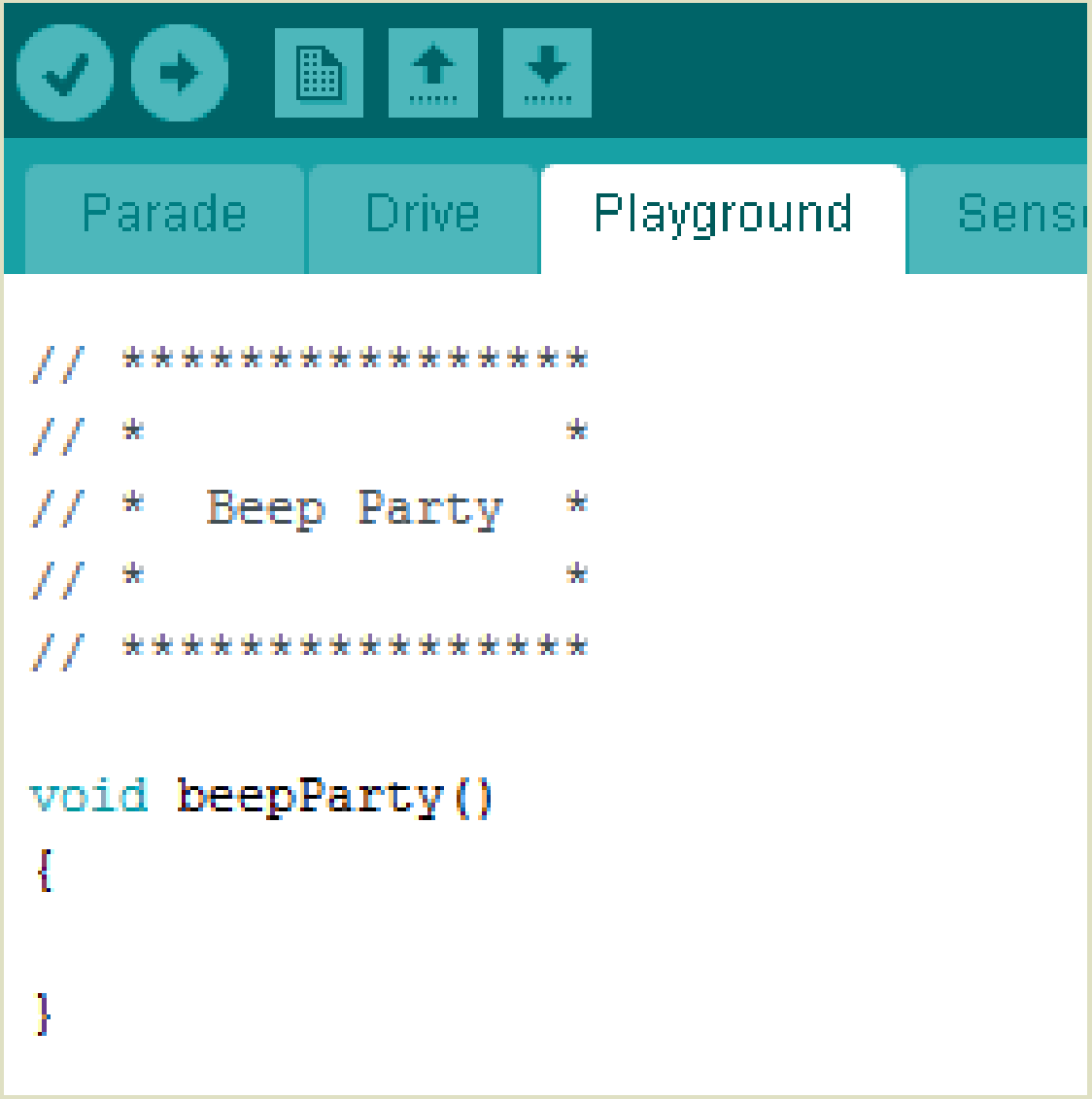
Piezoelectrics



Pitch and Frequency

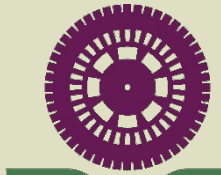


gizmo garden©

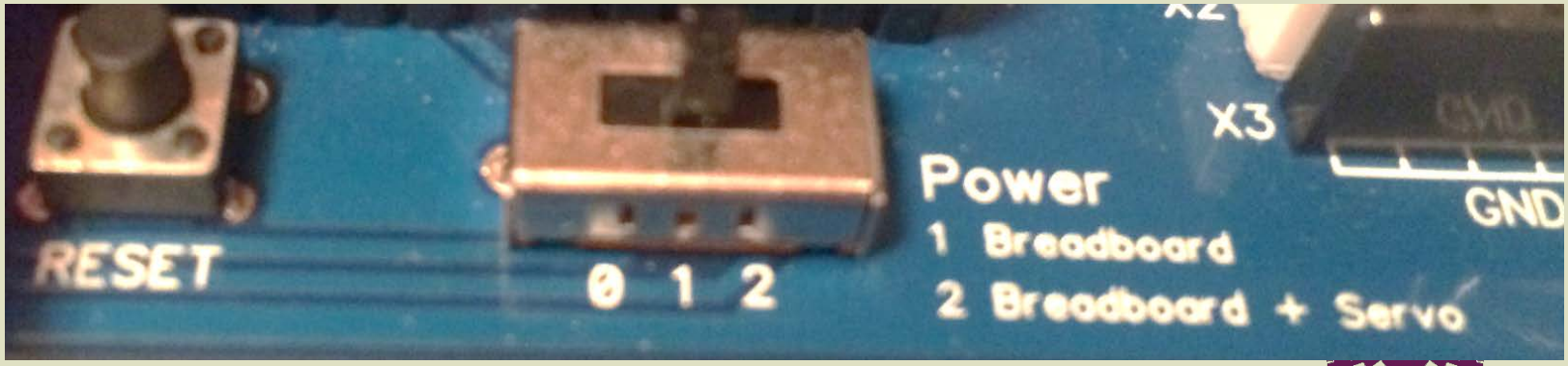


Play & Delay Code

```
void beepParty()
{
   GizmoGardenTone(2400, 250);
   delay(400);
}
```



Upload, Power #1, Reset if Dashboard is Blank

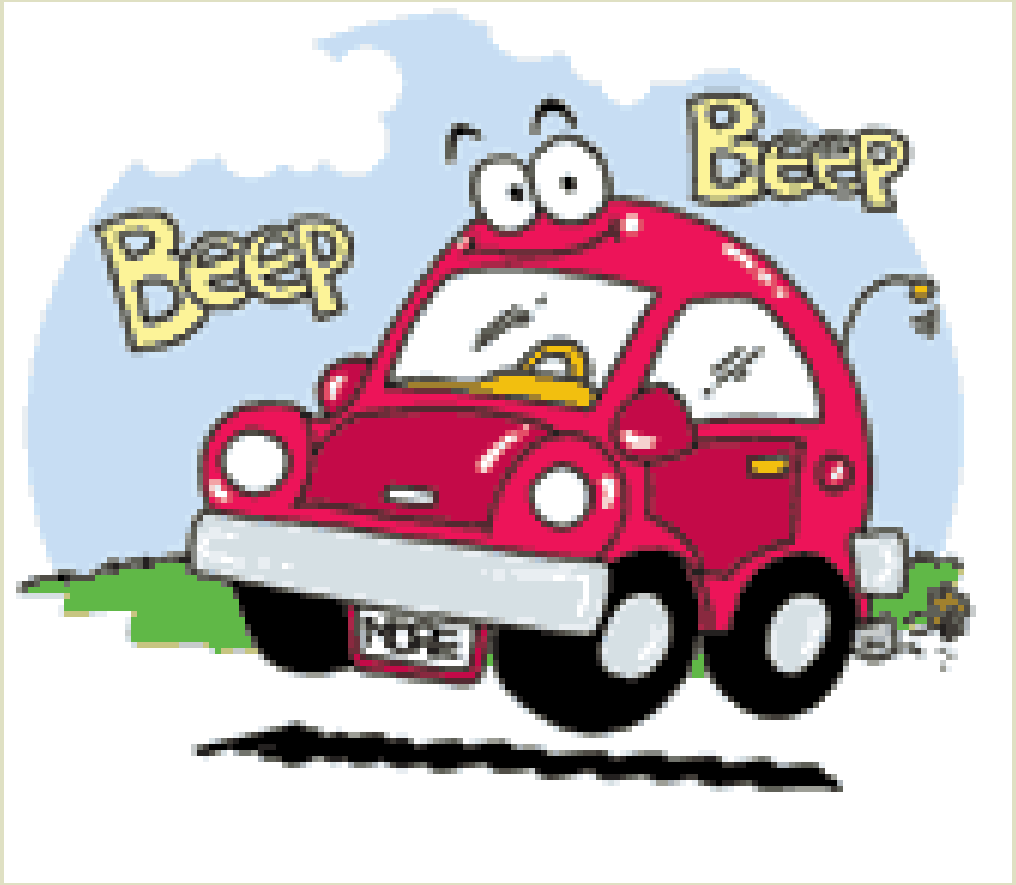


gizmo garden©

Scroll Left-Right to Beep Party

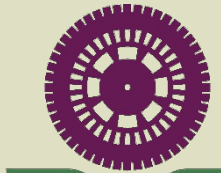
Push Select

gizmo garden©

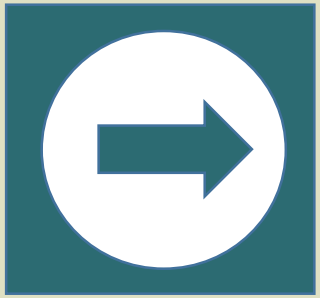


Try higher pitch, longer lasting

```
void beepParty()
{
    GizmoGardenTone(3500, 1000);
    delay (400);
}
```



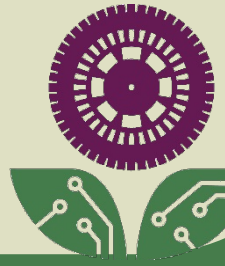
Upload, BeepParty, Select



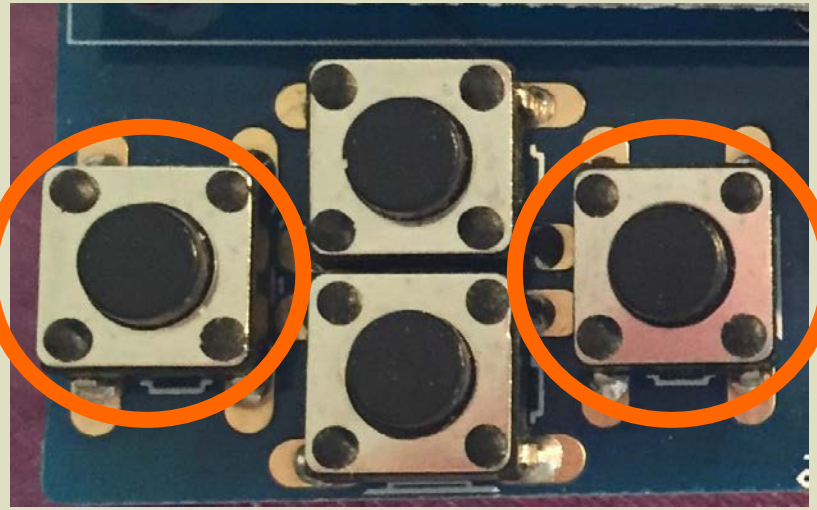
Code Reference #5

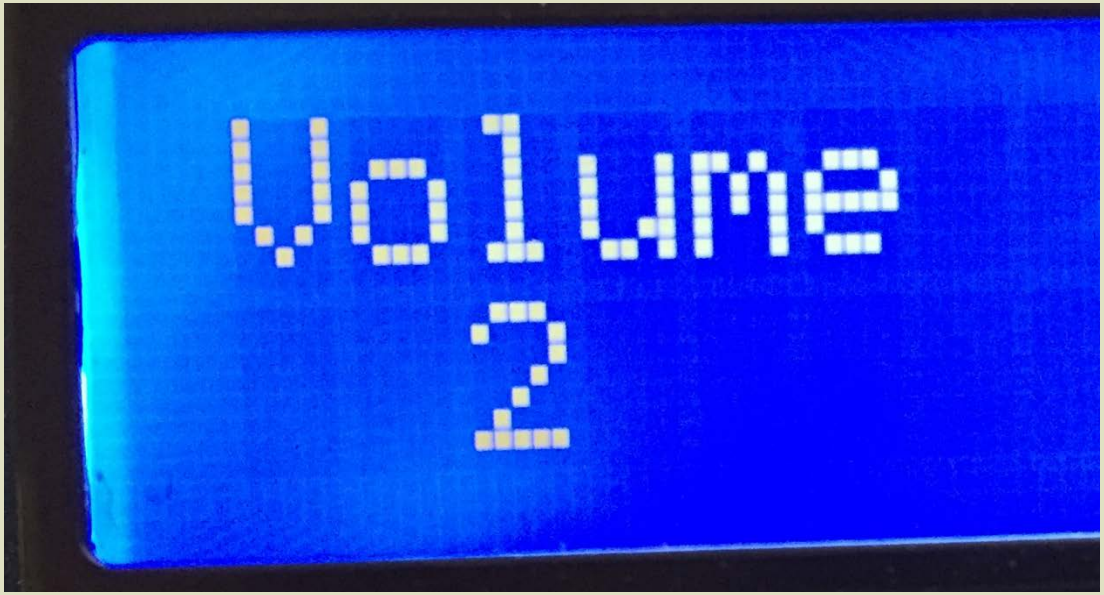
Beep Horn Function

```
GizmoGardenTone(pitch, milliseconds);
```



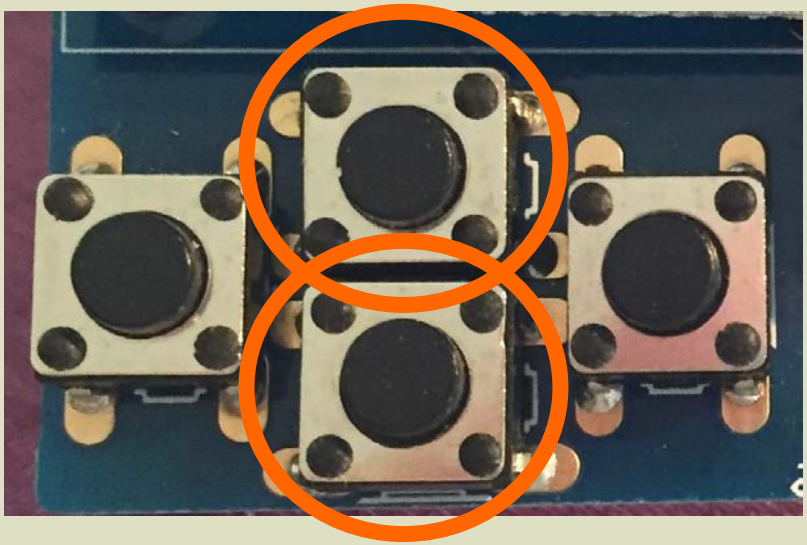
L-R Scroll to Volume





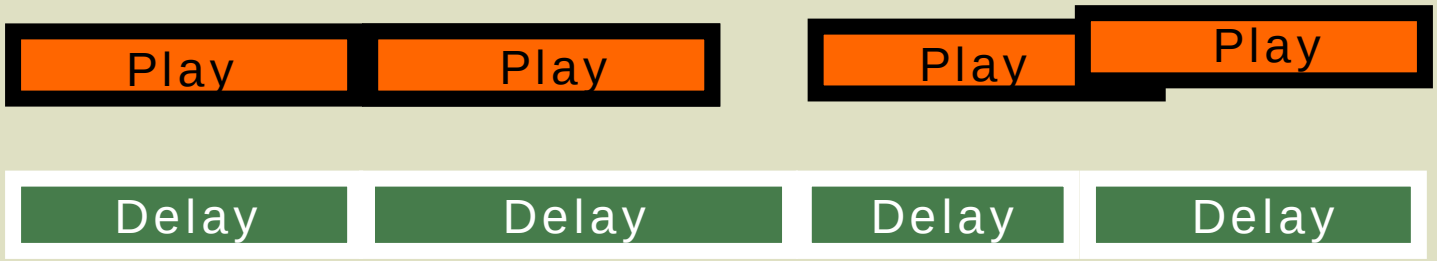
gizmo garden©

Up-Down Lower Vol



gizmo garden©

Play & Delay



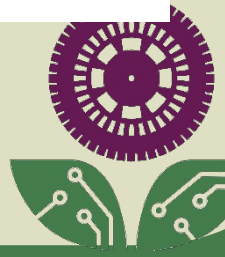
Your Beeping Party

```
void beepParty()
{
```

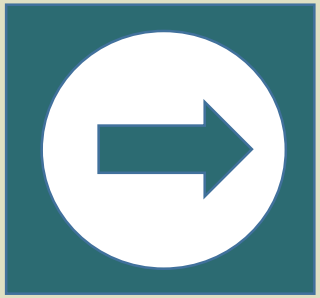
Your Code Here

Cut/Paste Works!

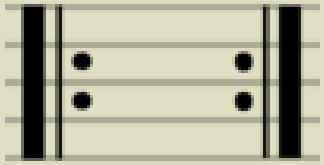
```
}
```



Upload, BeepParty, Select



How Do I Repeat?



Editing Code to Make a “For loop”

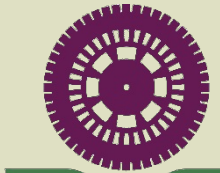
```
void beepParty()
{
  GizmoGardenTone(2400, 250);
  delay(400);
}
```

```
void beepParty()
{
  for (int i = 0; i < 5; ++i)
  {
    GizmoGardenTone(2400, 250);
    delay(400);
  }
}
```


gizmo garden©

Click to Set the Cursor

```
void beepParty()
{
  GizmoGardenTone(2400, 250);
  delay(400);
}
```

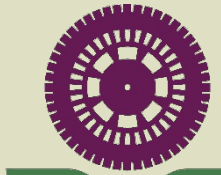


Enter

```
void beepParty()
{

  GizmoGardenTone(2400, 250);
  delay(400);
}
```

Type: for ()

```
void beepParty()
{
  for ()
    GizmoGardenTone(2400, 250);
    delay(400);
}
```



Enter

```
void beepParty()
{
  for ()
  |
  GizmoGardenTone(2400, 250);
  delay(400);
}
```

Type: {

```
void beepParty()
{
  for ()
  {
  GizmoGardenTone(2400, 250);
  delay(400);
}
```

gizmo garden©

Enter

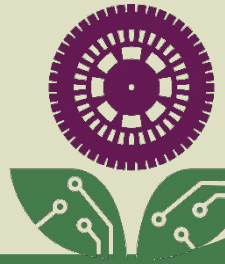
```
void beepParty()
{
   for ()
   {

   }
   GizmoGardenTone(2400, 250);
   delay
}
```

Added automatically

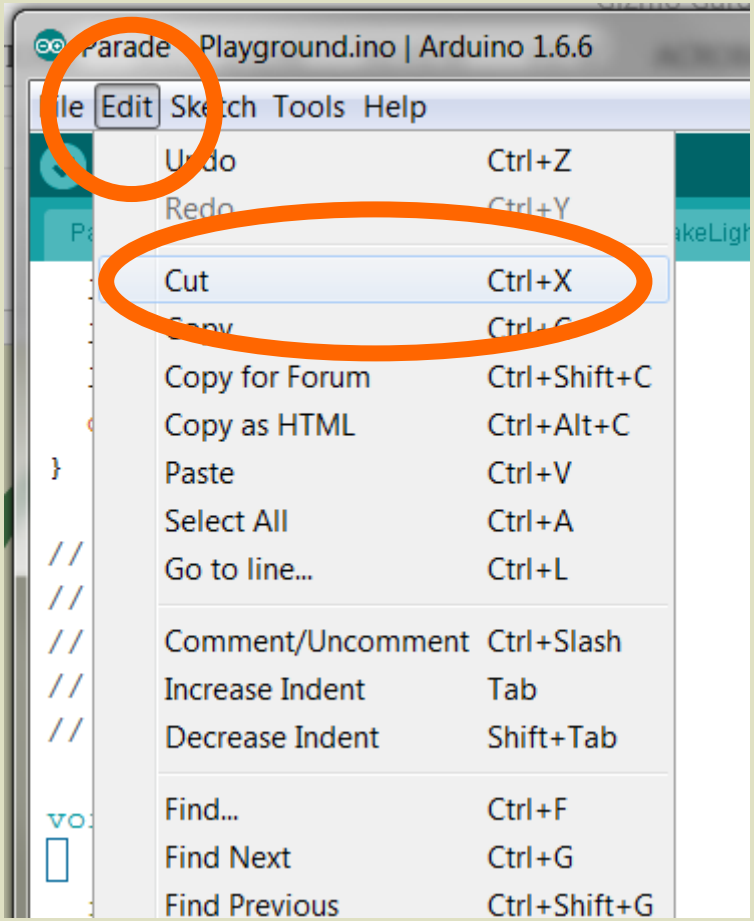
gizmo garden©

Click and Drag to Select

```
void beepParty()
{
  for ()
  {

  }
  GizmoGardenTone(2400, 250);
  delay(400);
}
```

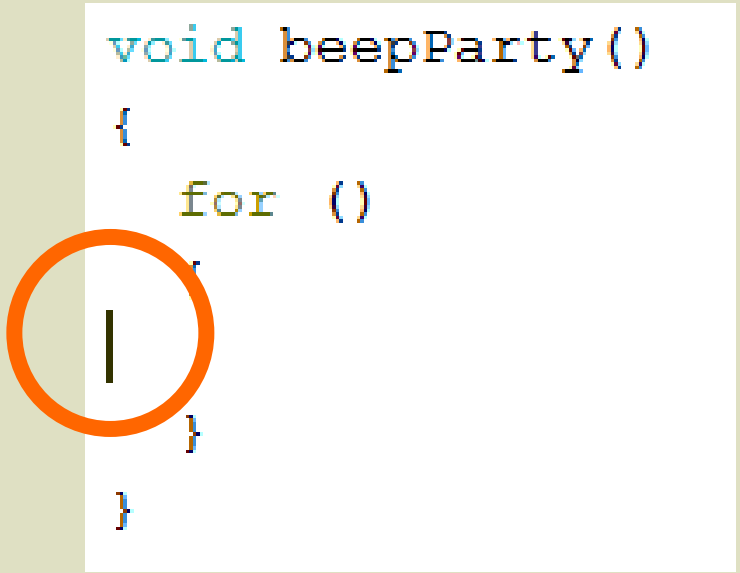


Edit→Cut, or Crtl-X

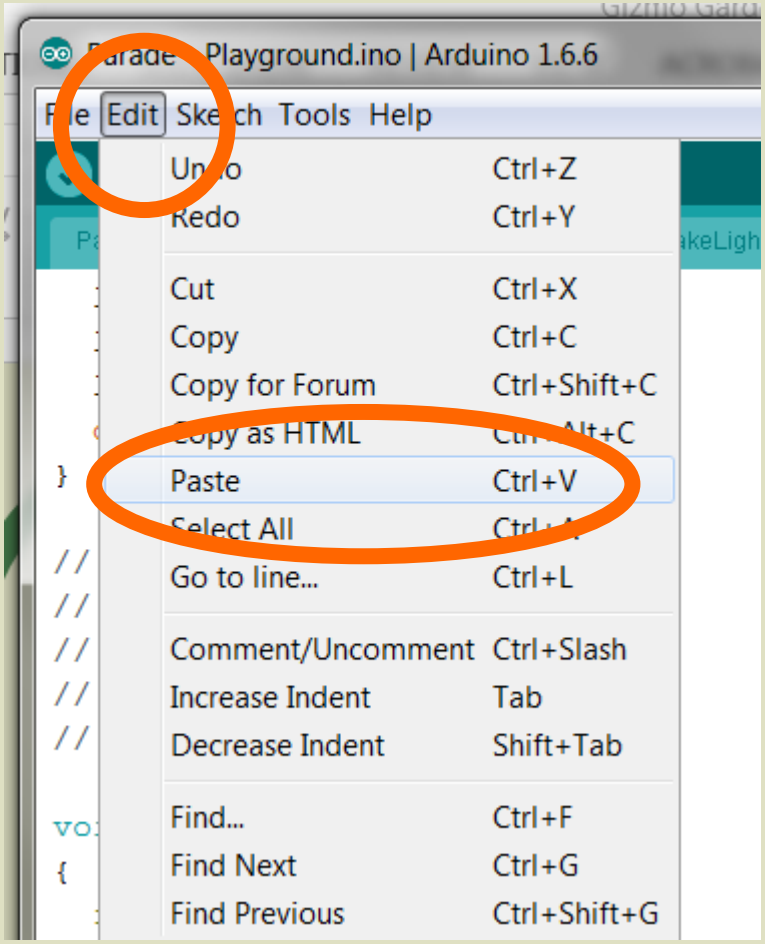


gizmo garden©

Click to Set the Cursor



Edit→Paste, or Crtl-V

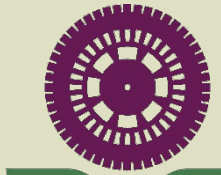



```
void beepParty()
{
  for ()
  {
  GizmoGardenTone(2400, 250);
  delay(400);

  }
}
```


gizmo garden©

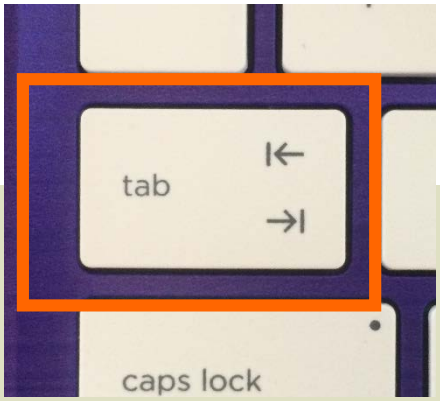
Click and Drag to Select

```
void beepParty()
{
   for ()
   {
   GizmoGardenTone(2400, 250);
   delay(400);

   }
}
```



Tab to Indent

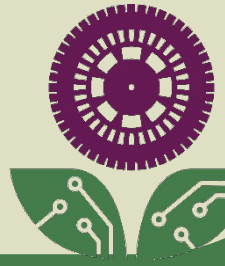
```
void beepParty()
{
   for ()
   {
      GizmoGardenTone(2400, 250);
      delay(400);

   }
}
```



Tip: Shift-Tab undoes indent

Remove Blank Line

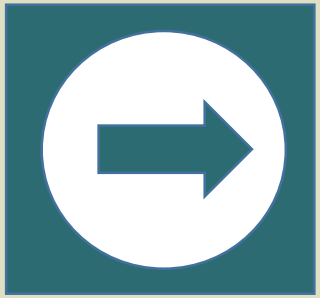
```
void beepParty()
{
  for ()
  {
    GizmoGardenTone(2400, 250);
    delay(400);
  }
}
```



Type this to tell the **for** loop what to do

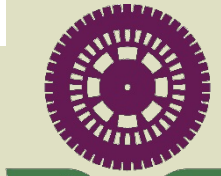
```
void beepParty()
{
   for (int i = 0; i < 5; ++i)
   {
      GizmoGardenTone(2400, 250);
      delay(400);
   }
}
```

Upload, BeepParty, Select

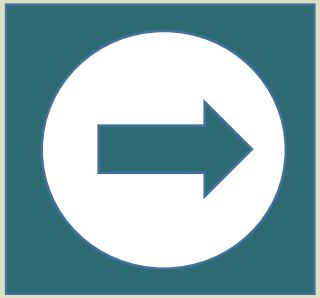


Change # of Repeats

```
void beepParty()
{
   for (int i = 0; i < 5; ++i)
   {
      GizmoGardenTone(2400, 250);
      delay(400);
   }
}
```



Upload, BeepParty, Select



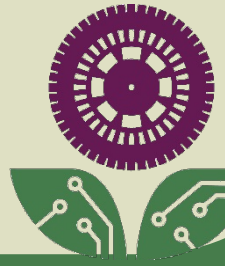
Code Reference #6

for Loop

(most basic)

- Repeats whatever is within the curly braces

```
for (int i = 0; i < #ofLoops; ++i)
{

}
```



Deeper Dive #1

(in Beep Party section of Playground Tab)

- What happens when you change the pitch from the constant value of 2400 to a mathematical expression? Try it!

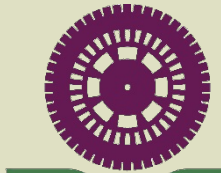
```
void beepParty()
{
  for (int i = 0; i < 5; ++i)
  {
    GizmoGardenTone(2400 + 250 * i, 250);
    delay(400);
  }
}
```

Code Reference #7

Math Expressions

- You can use the following math operations:

+	add
-	subtract
*	multiply
/	divide



Code Reference #8

Math Expressions

- Like algebra, multiply and divide are done before add and subtract
- () can change the order

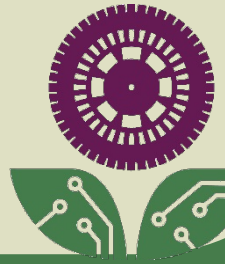
```
 2400 + 250  * i // multiply then add
(2400 + 250) * i // add then multiply
```

Deeper Dive #2

(in Beep Party section of Playground Tab)

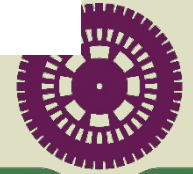
- Let's break down the pieces of the "How To" within the parenthesis of the loop. There's a set-up, an ending test, and a next step.

```
void beepParty()
{
  for (int i = 0; i < 5; ++i)
  {
    GizmoGardenTone(2400, 250);
    delay(400);
  }
}
```



Set-Up Creates a "Variable"

```
void beepParty()
{
   for (int i = 0; i < 5; ++i)
   {
      GizmoGardenTone(2400, 250);
      delay(400);
   }
}
```



Code Reference #9

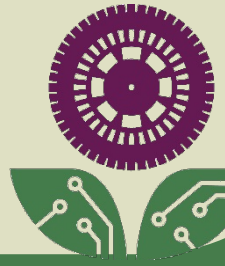
Create Variable

```
int i = 0;
```

Say what type you want. One type of variable is a whole number, which we write as **`int`**, short for integer.

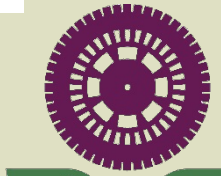
Pick a name for your variable. Usually you pick something descriptive, but for simple loop counting we just use **`i`**.

Say what value your variable will have when it is first created. Later code can change the value.



Ending Test

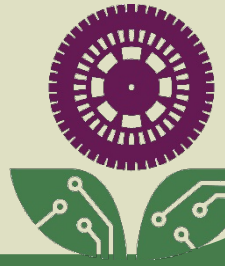
```
void beepParty()
{
   for (int i = 0; i < 5; ++i)
   {
      GizmoGardenTone(2400, 250);
      delay(400);
   }
}
```



Code Reference #10

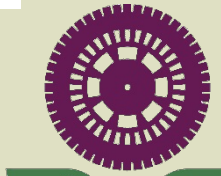
Comparisons

- You can make comparisons using these symbols
  - **>** (greater than)
  - **<** (less than)
  - **>=** (greater than or equal to)
  - **<=** (less than or equal to)
  - **==** (equal to)
  - **!=** (not equal to)



Do Next

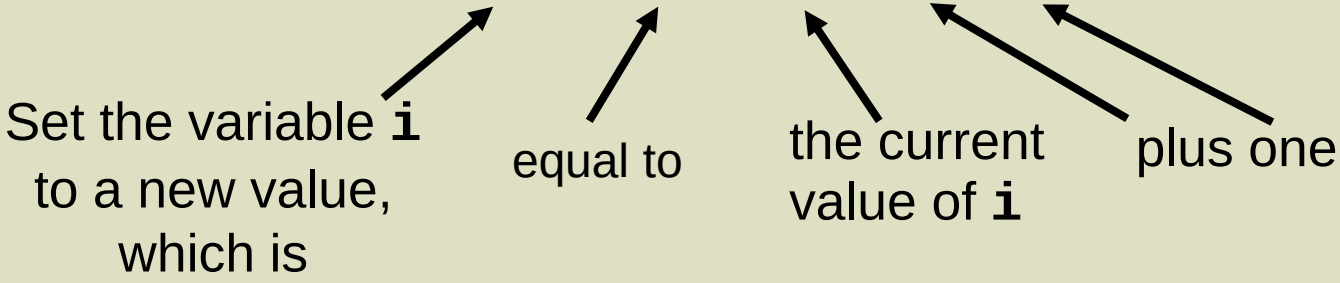
```
void beepParty()
{
   for (int i = 0; i < 5; ++i)
   {
      GizmoGardenTone(2400, 250);
      delay(400);
   }
}
```



Code Reference #11

Counting

```
i = i + 1
```




```
++i
```

This is an abbreviation so you don't have to type so much just to add 1

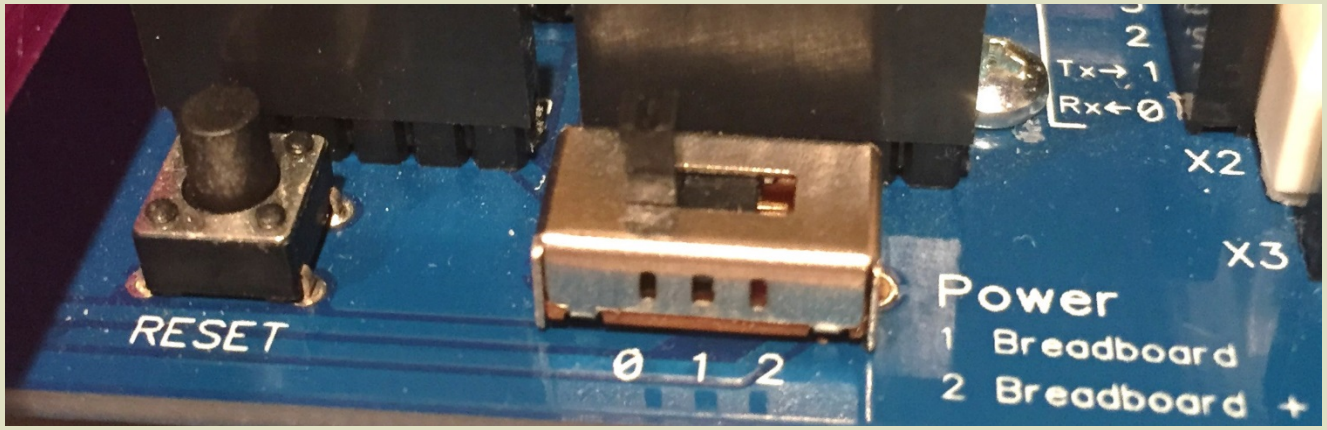
Deeper Dive #3

(in Beep Party section of Playground Tab)

- In the GizmoGardenTone function, give the variable the name **`pitch`** with a starting value of **`2400`**.
- Express the ending test and next step in terms of pitch.
- Does the beep party change?

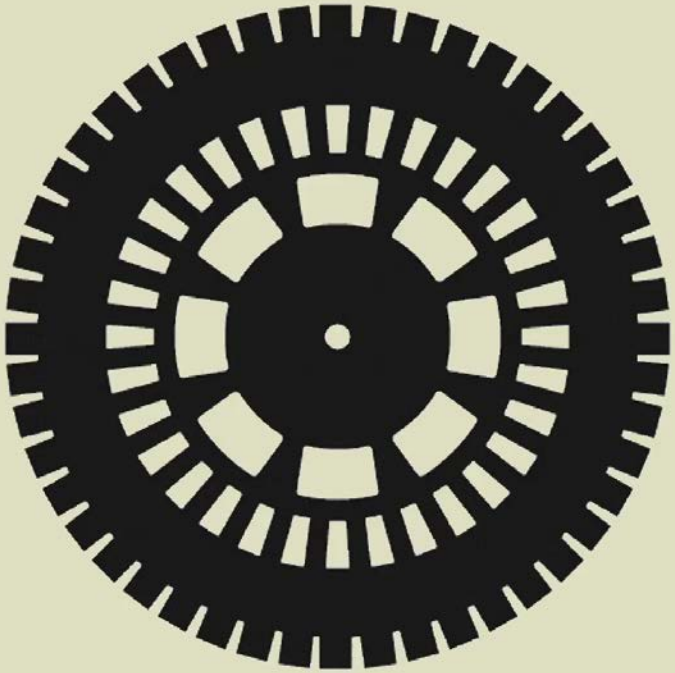
```
void beepParty()
{
  for (int pitch = 2400; pitch <= 3400; pitch = pitch + 250)
  {
    GizmoGardenTone(pitch, 250);
    delay(400);
  }
}
```

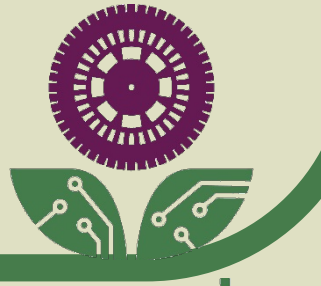
Power Off



gizmo garden©

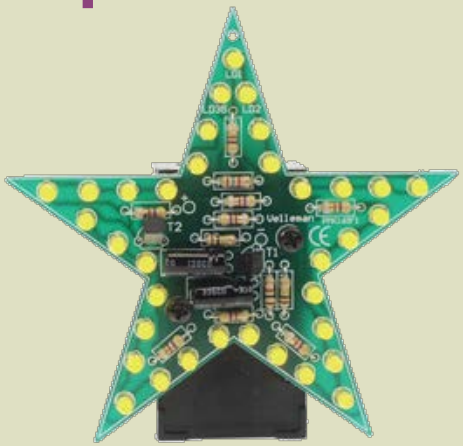
You are a C++ Coder!

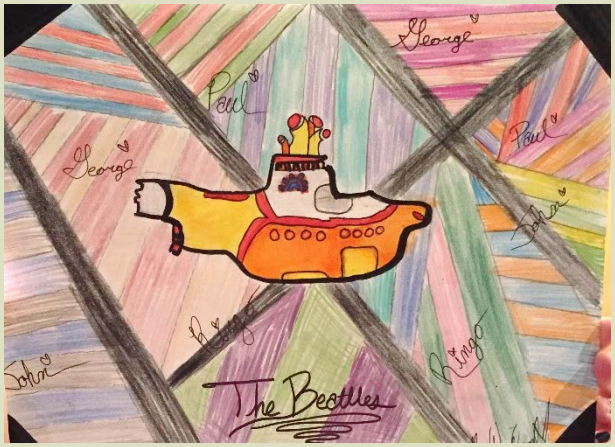




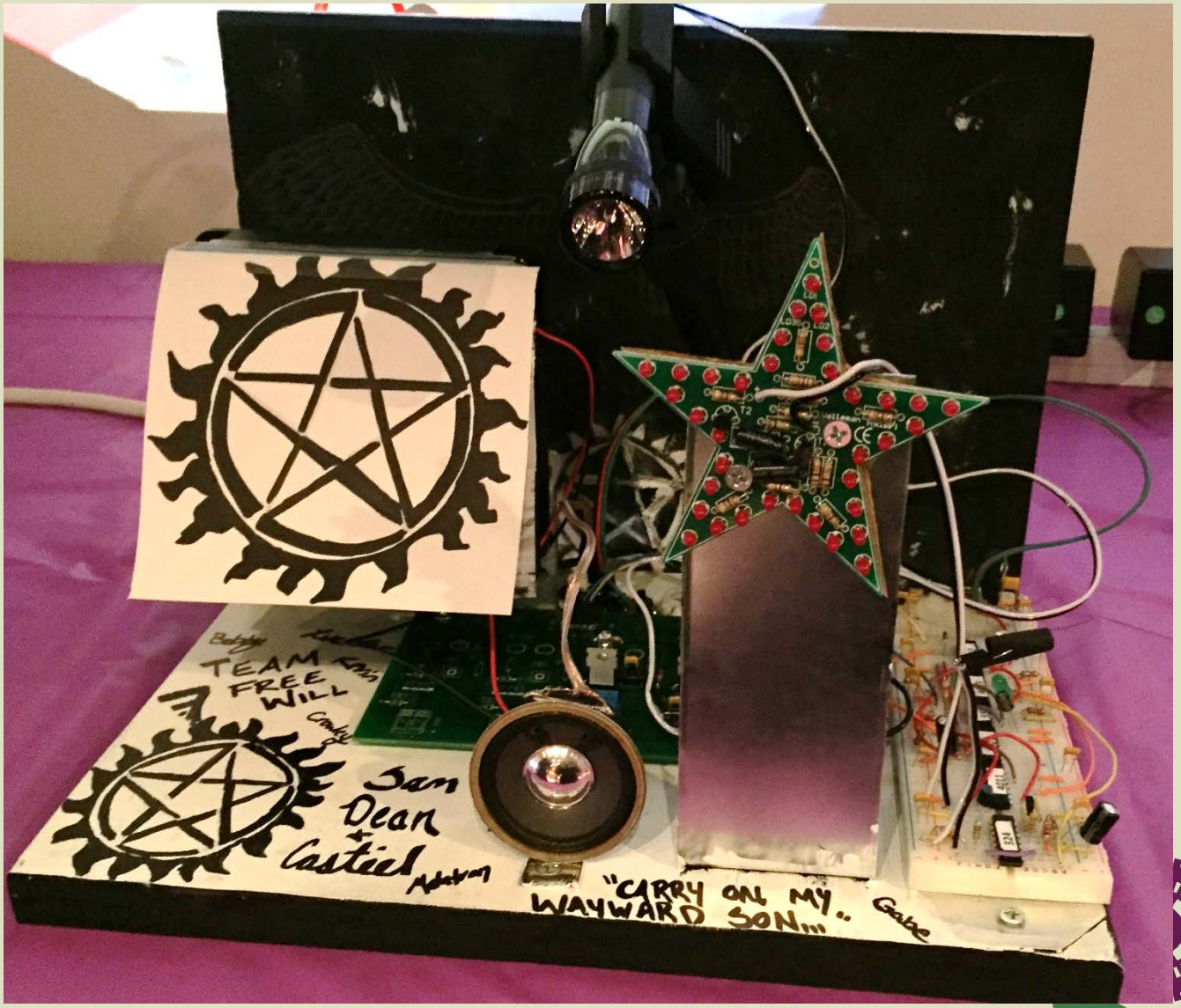
Gizmo Parade!

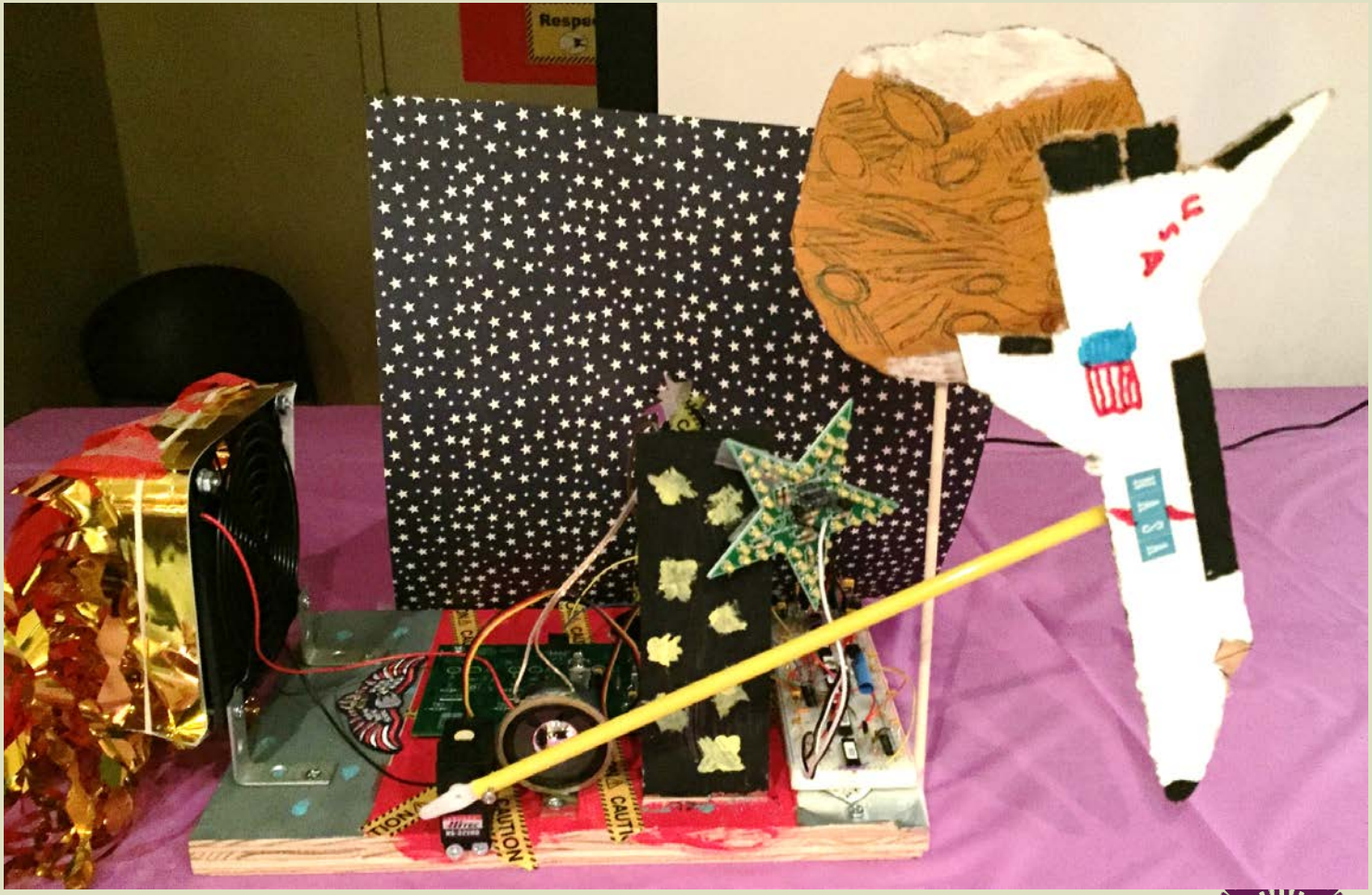
Last Year's Options

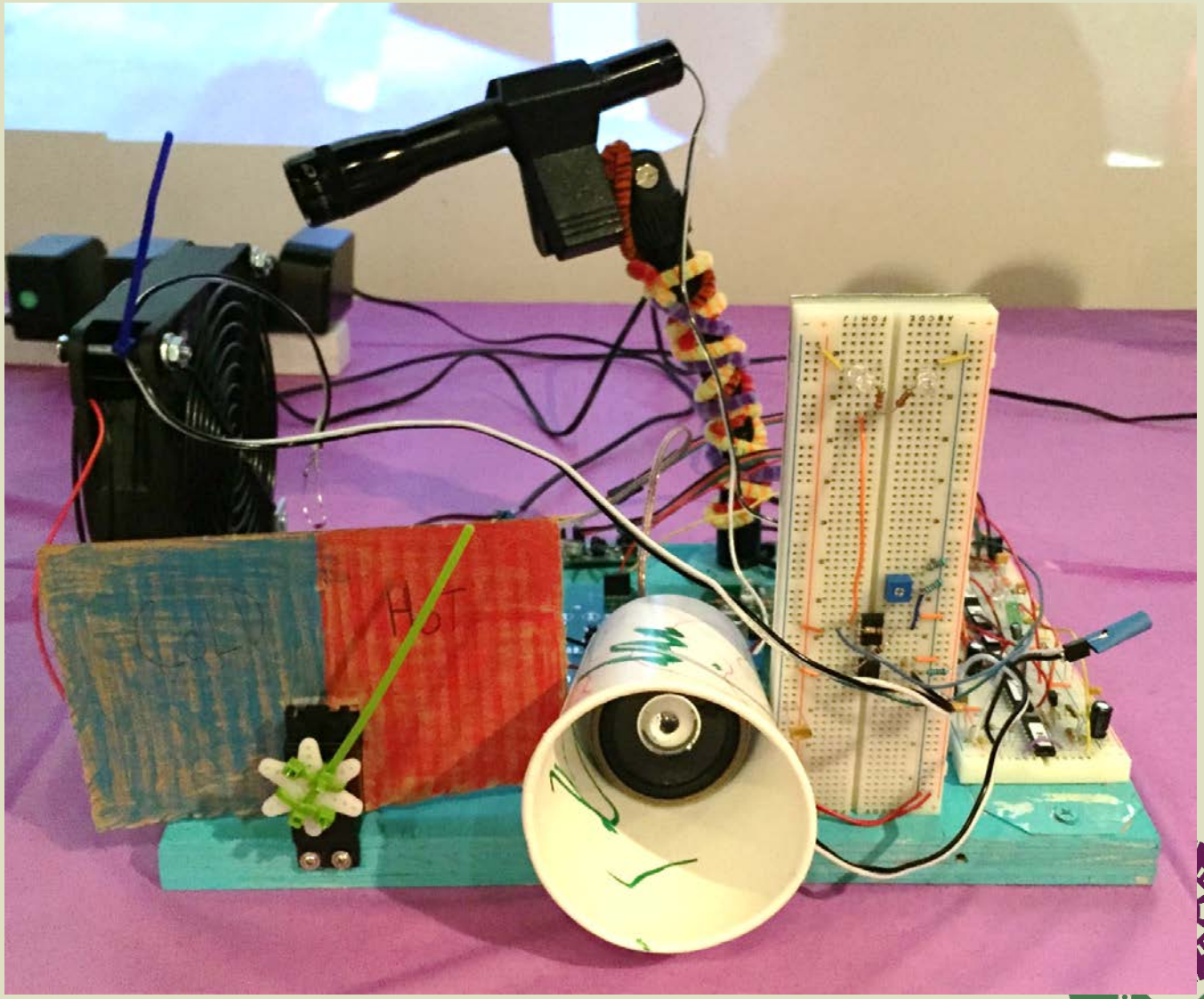




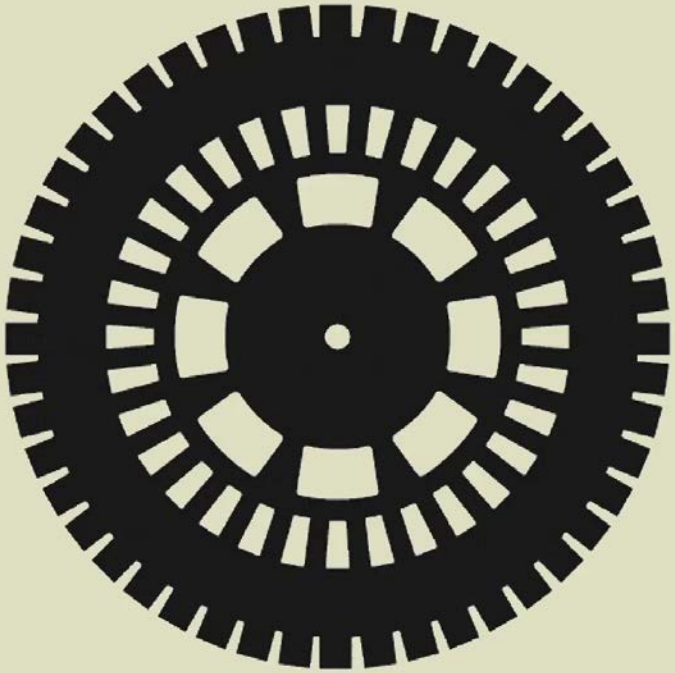
gizmo garden©



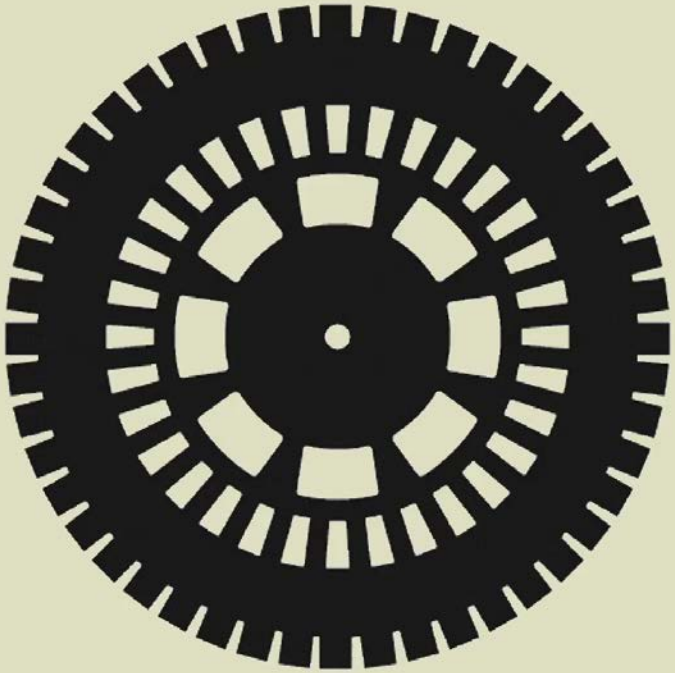




Great Gizmos!

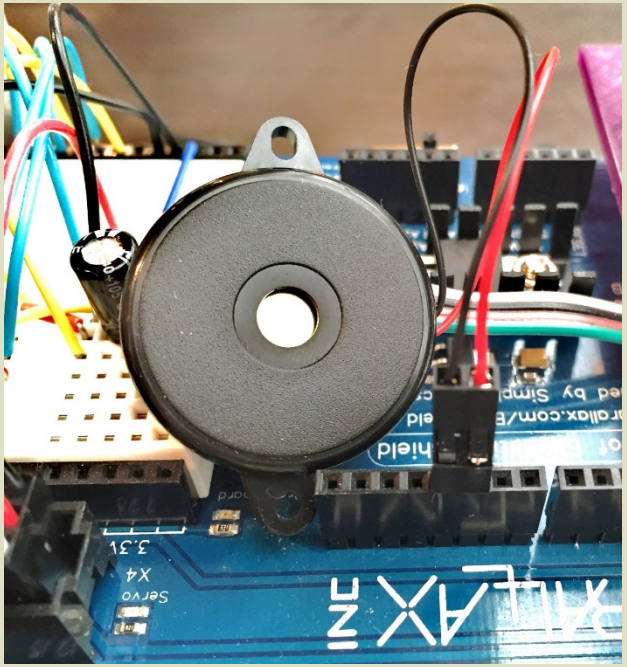


Let's Move!

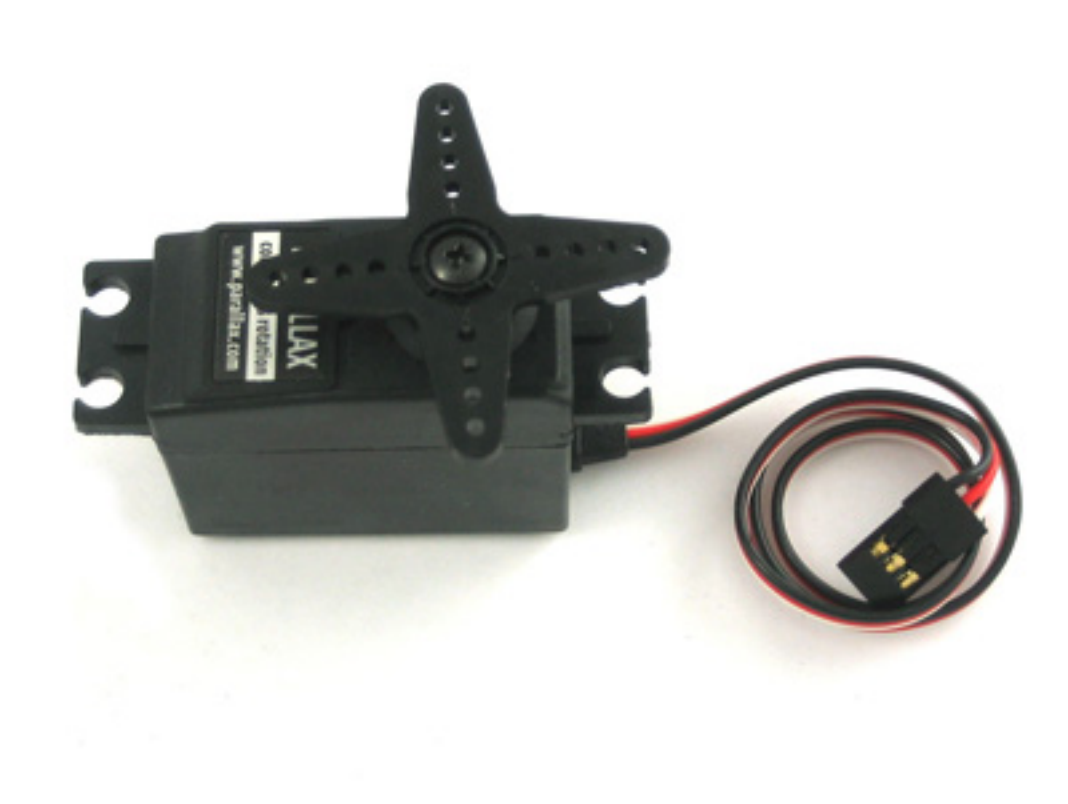


Remove Horn Beeper

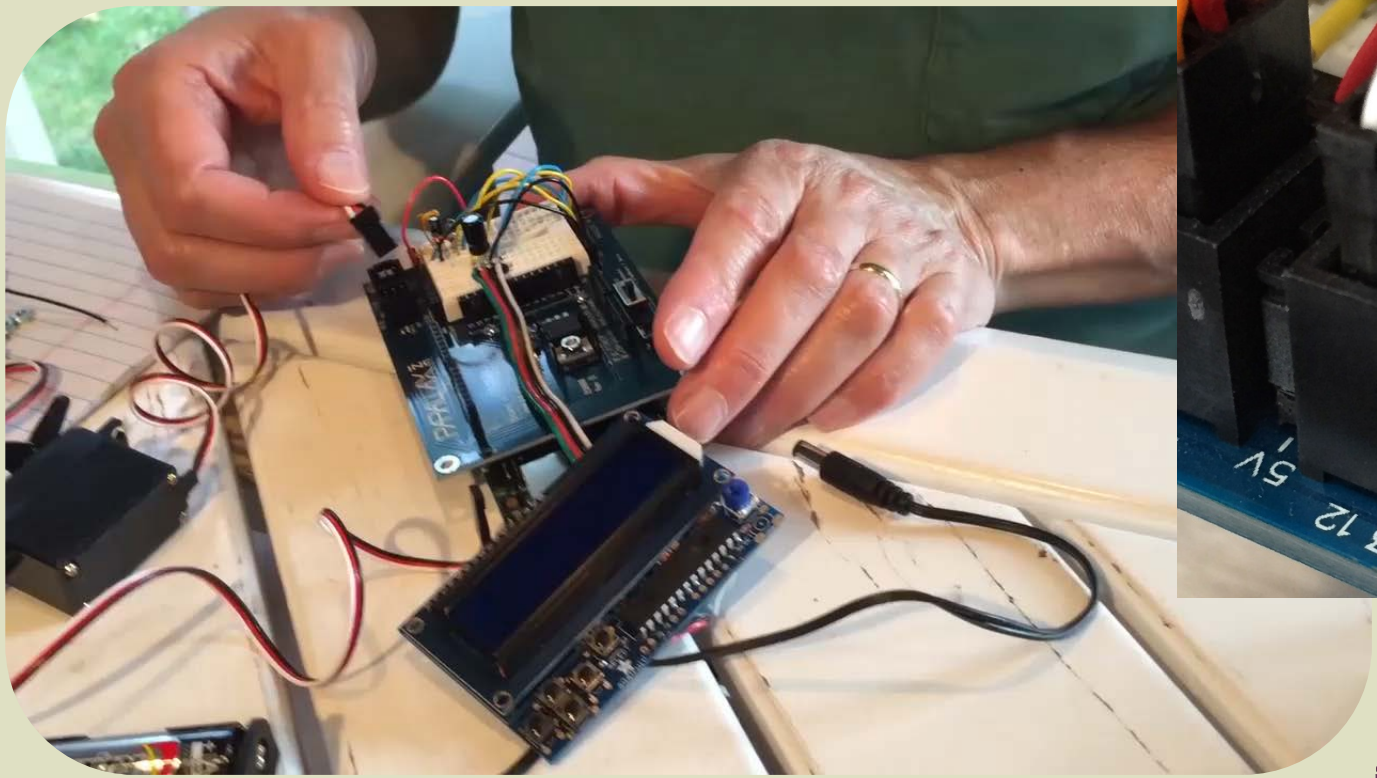
Confirm power unplugged from gizmo

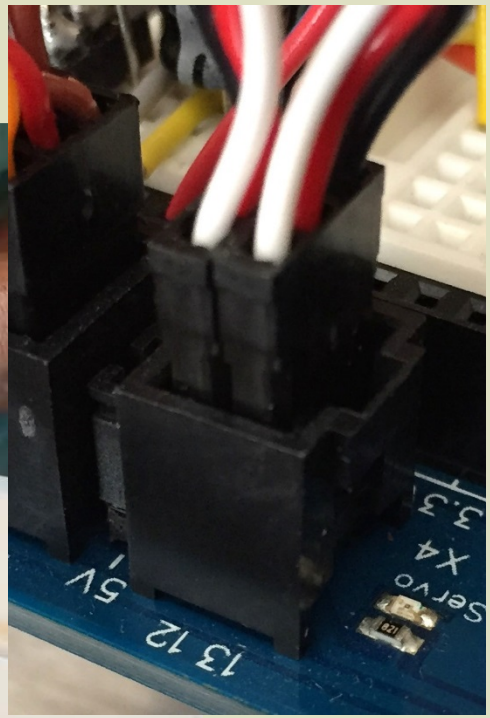


Motors



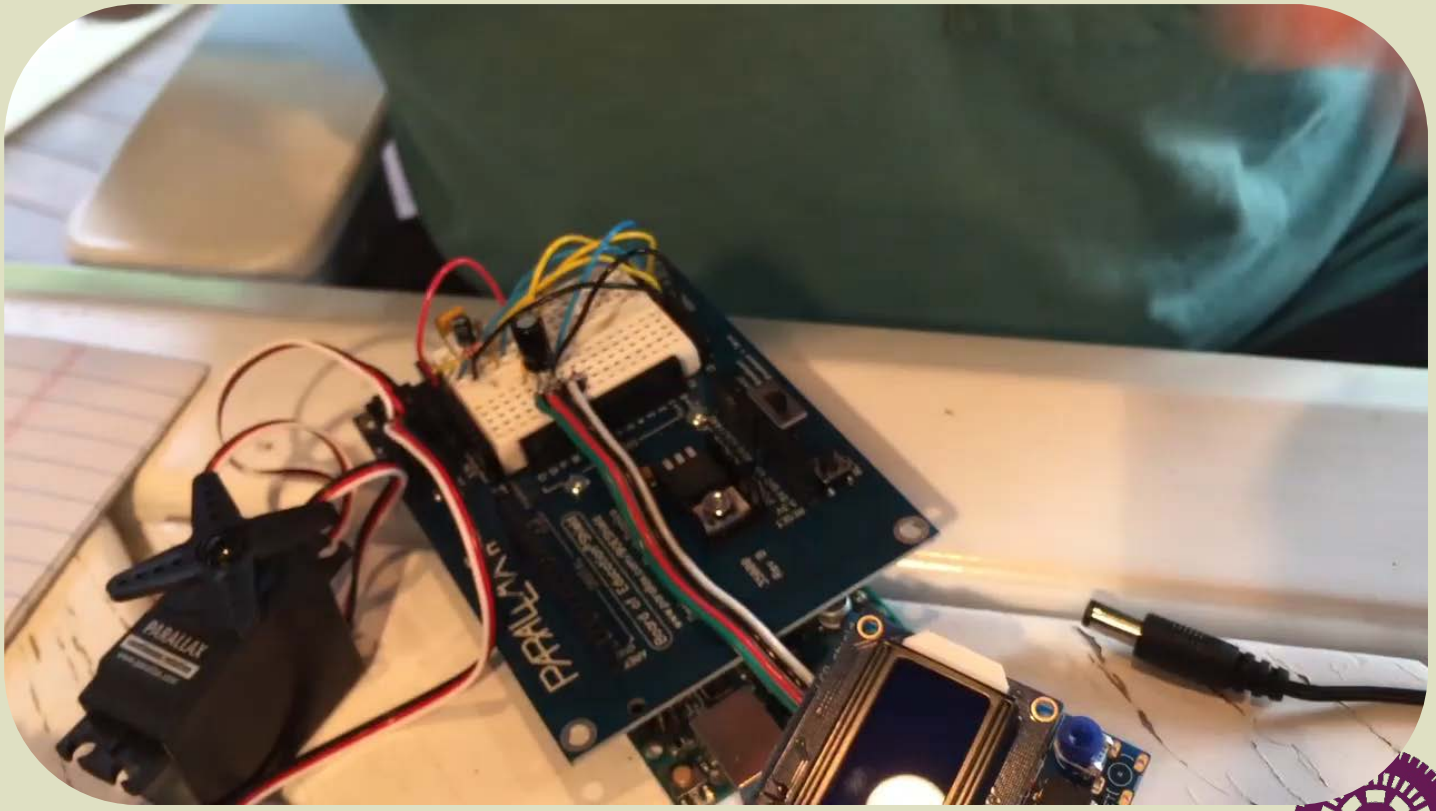
Plug in Motors



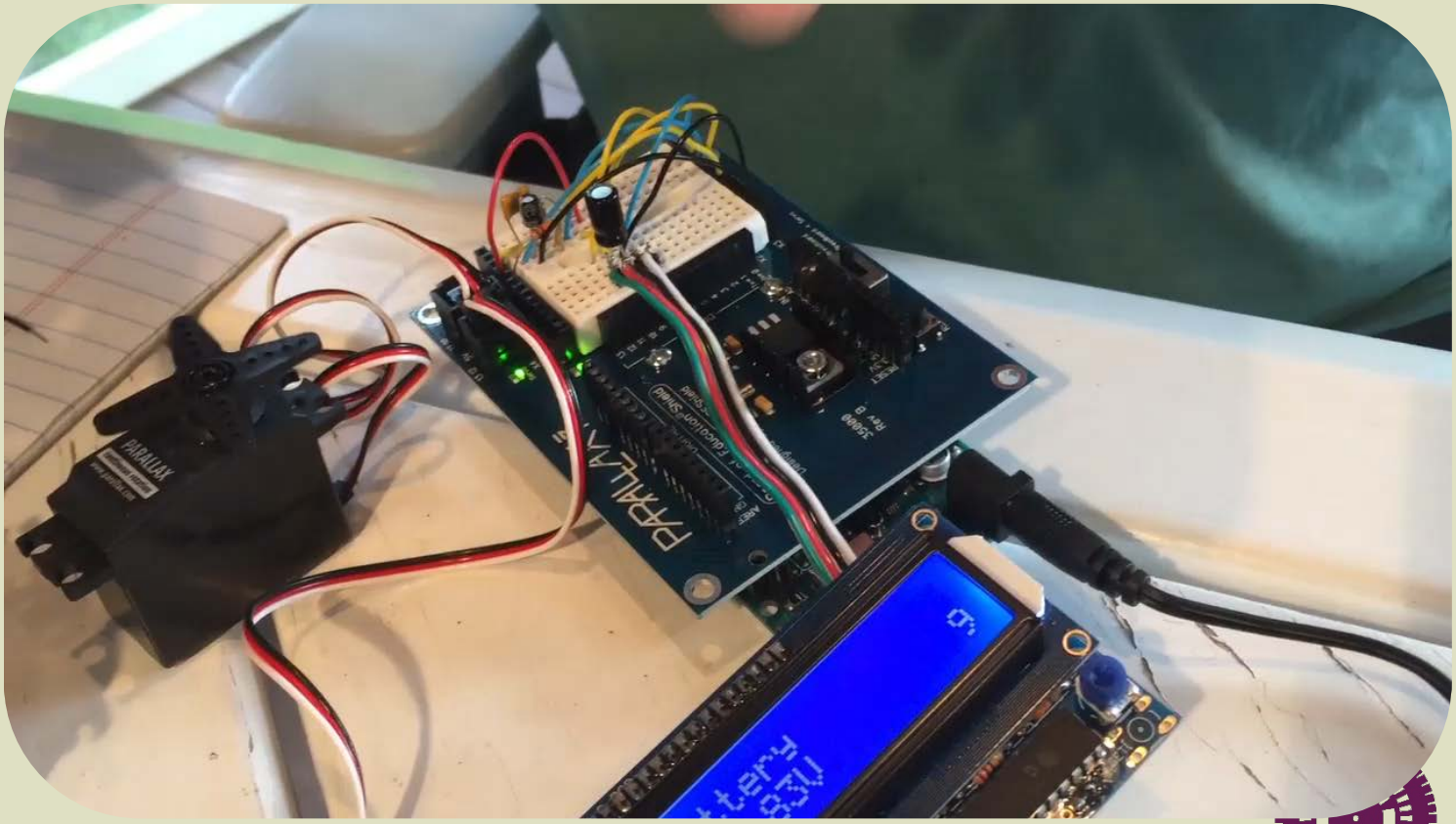


gizmo garden©

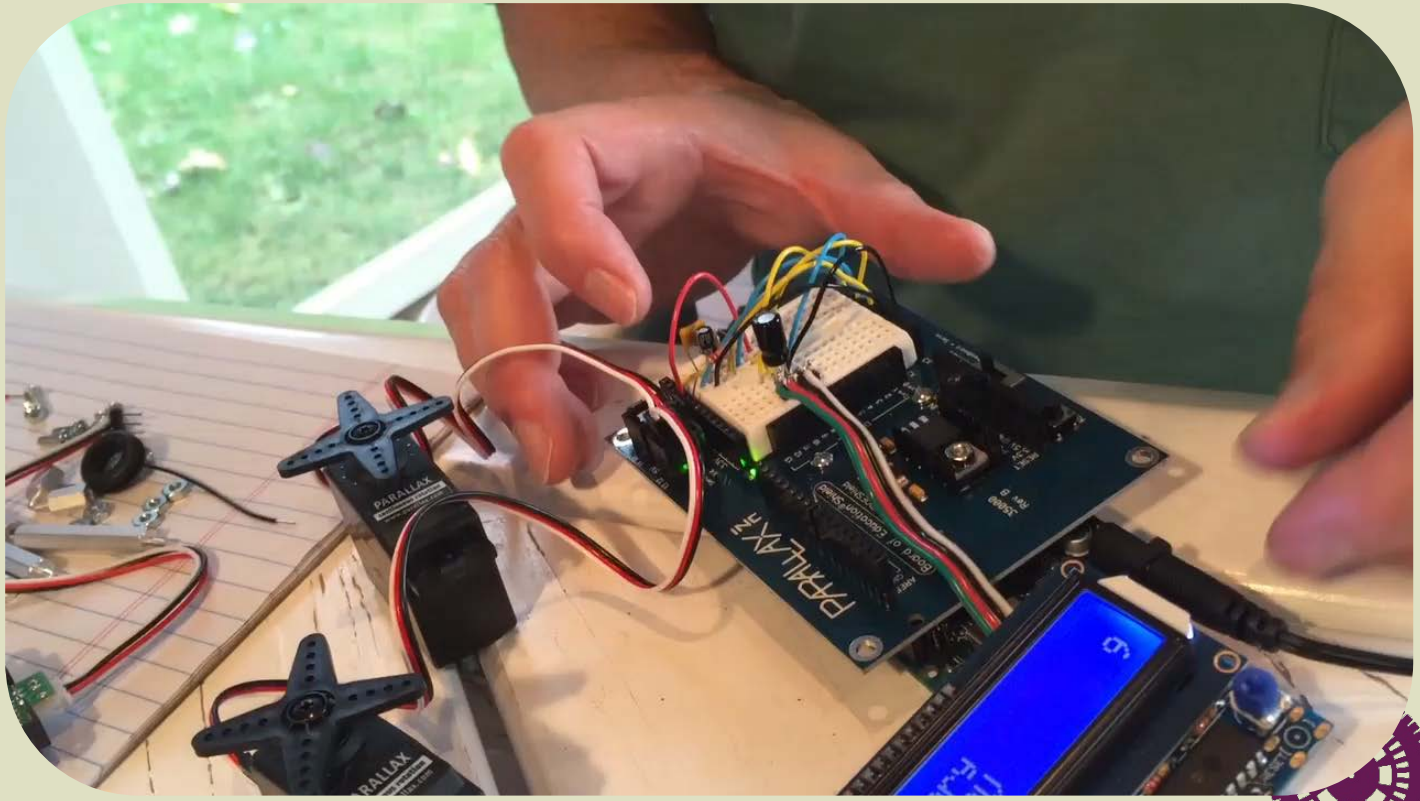
Power Up



Zero Motors

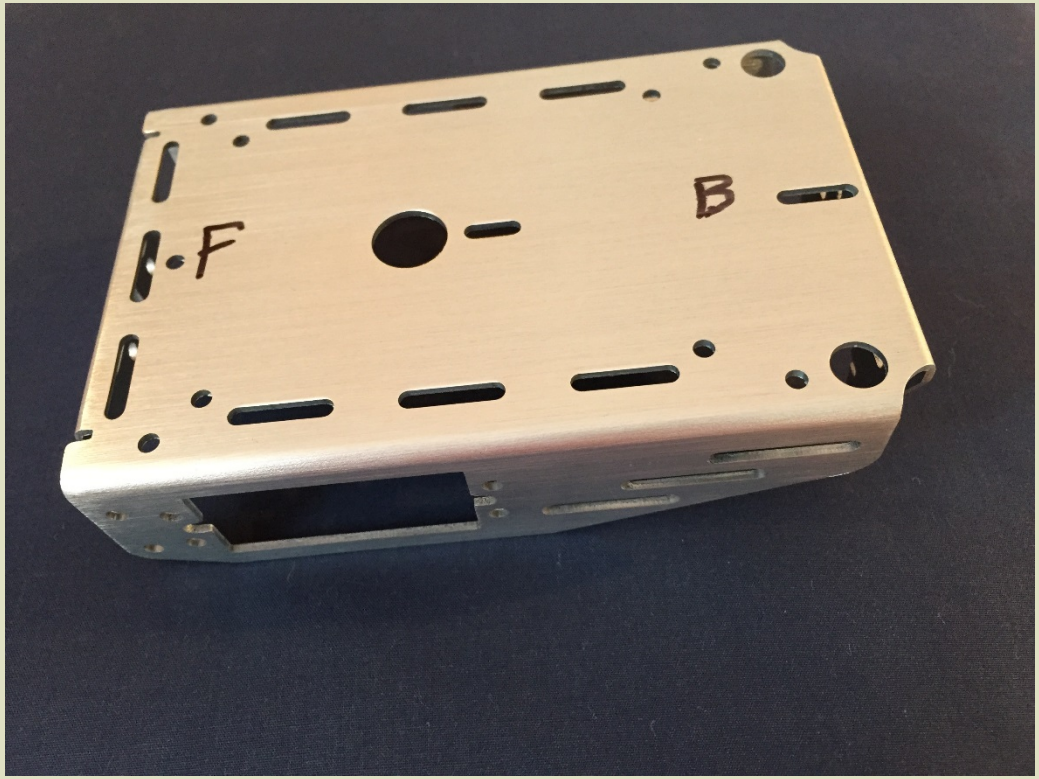


Unplug Power, Dashboard & Motors

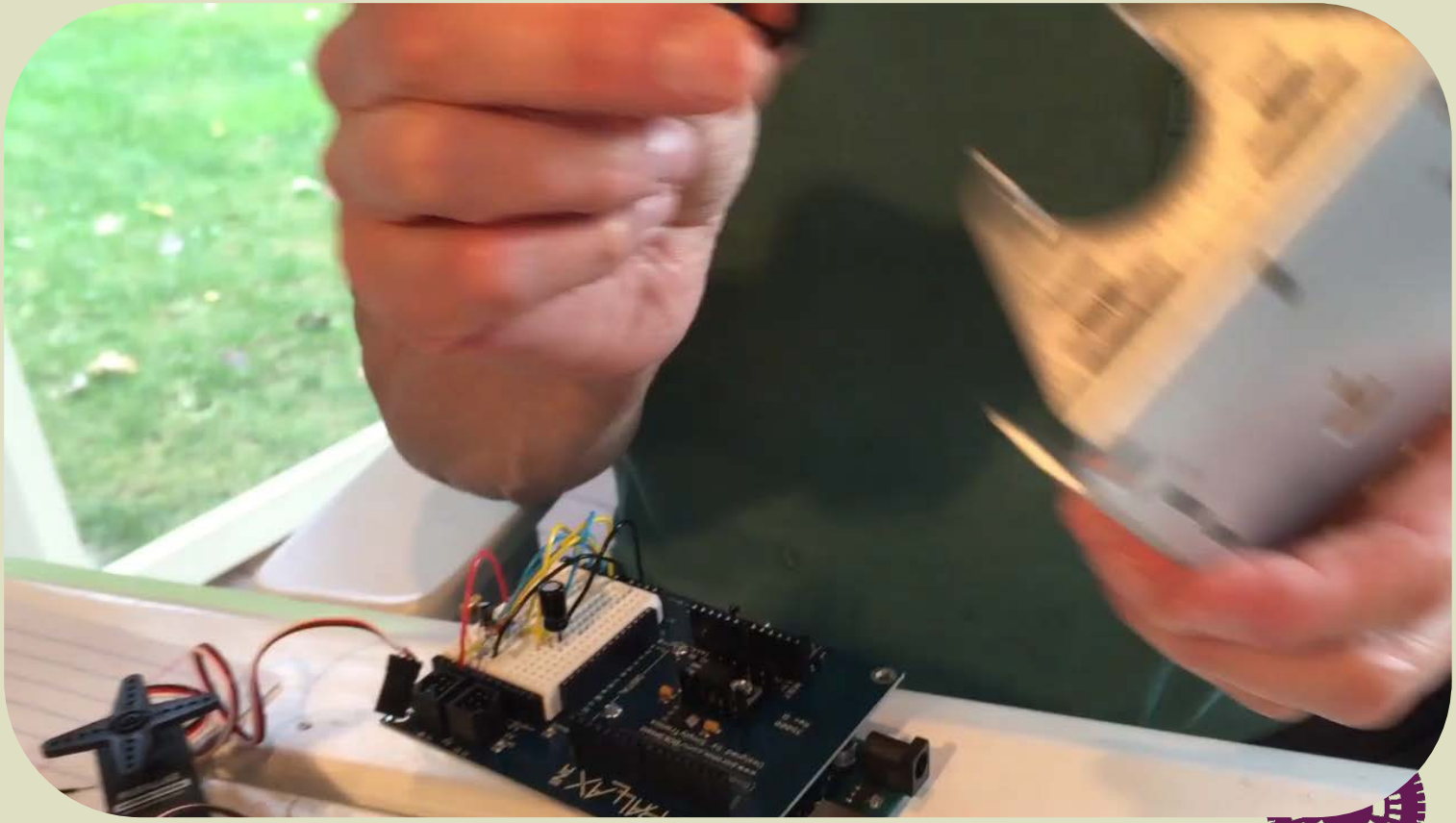


gizmo garden©

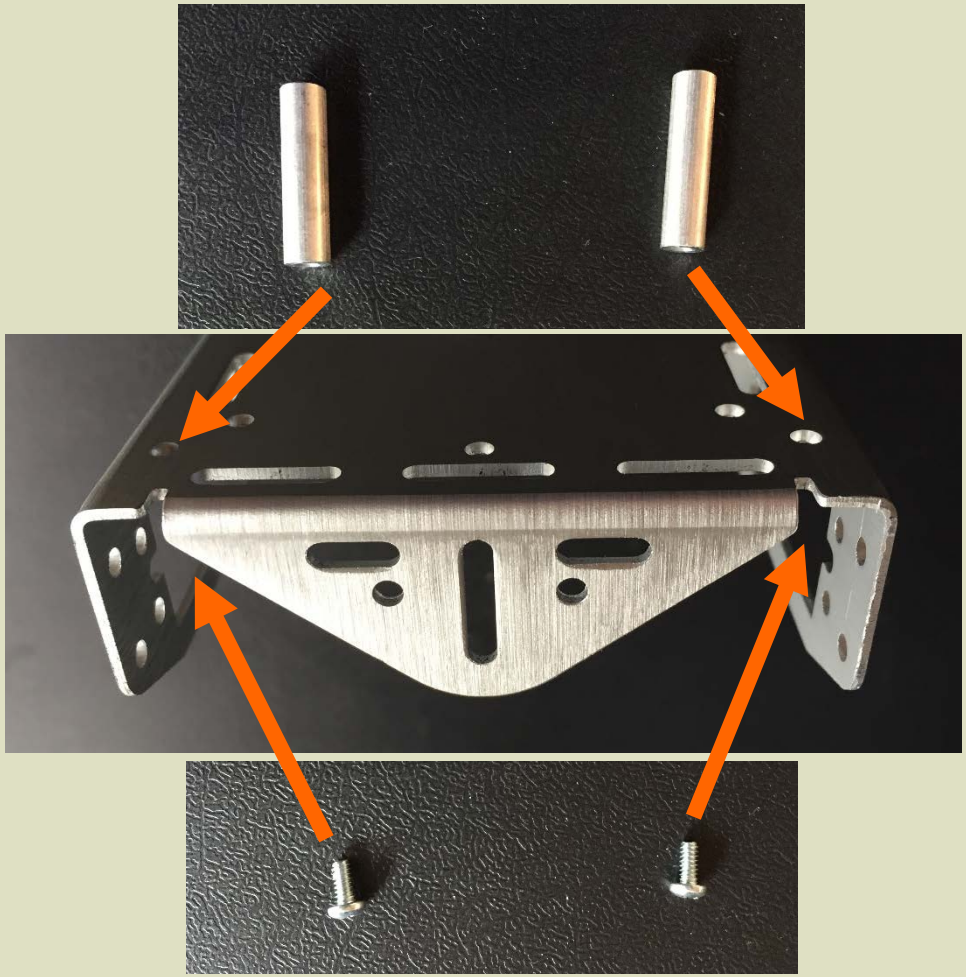
Mark Front and Back of Chassis



Insert Grommet



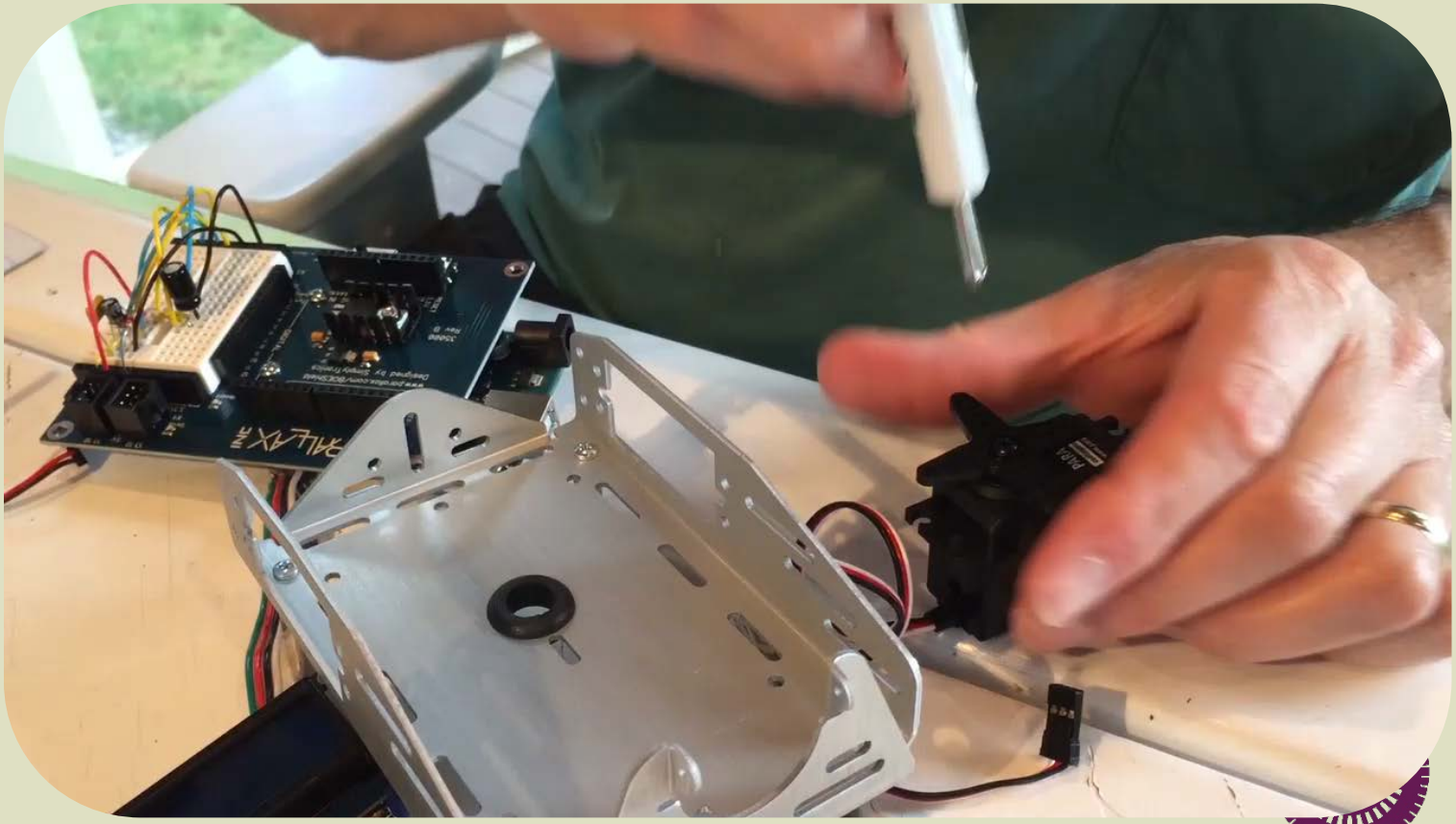
Install **Two 1”** Threaded Round Posts at Chassis Front **using Two ¼” (smallest) Screws**



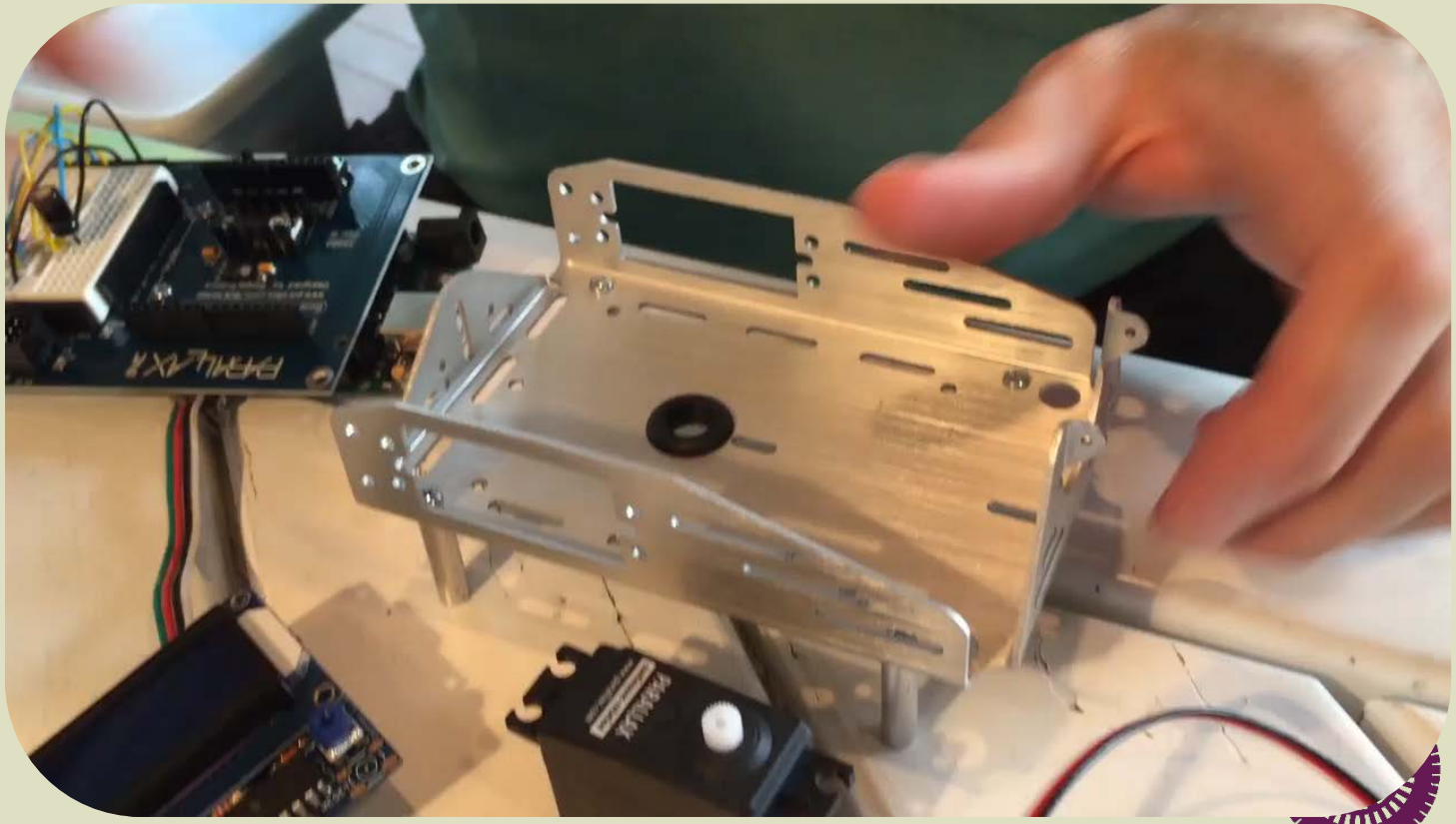
gizmo garden©

Attach 2 Front Posts

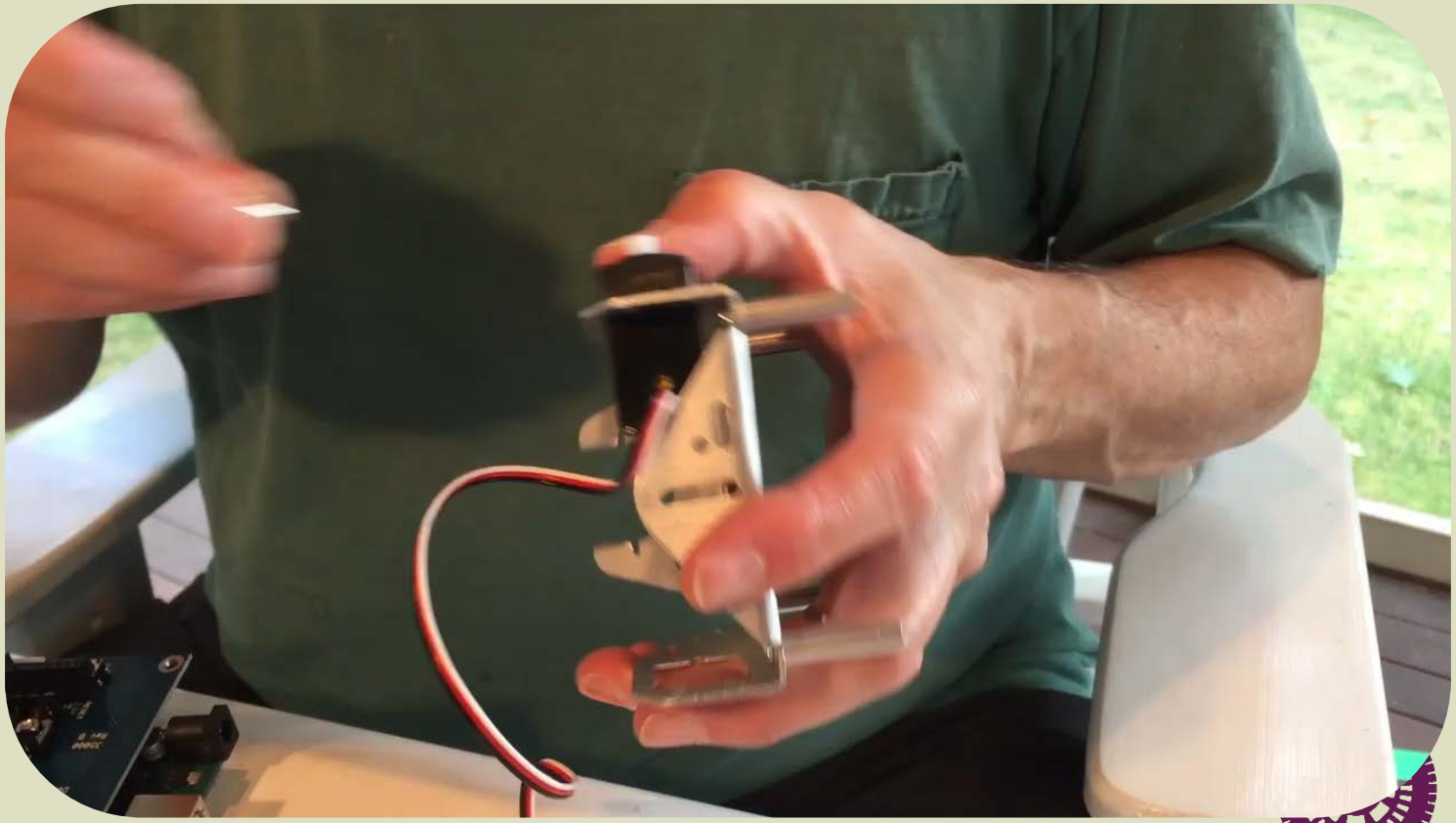
Unscrew & Remove Motor Arms



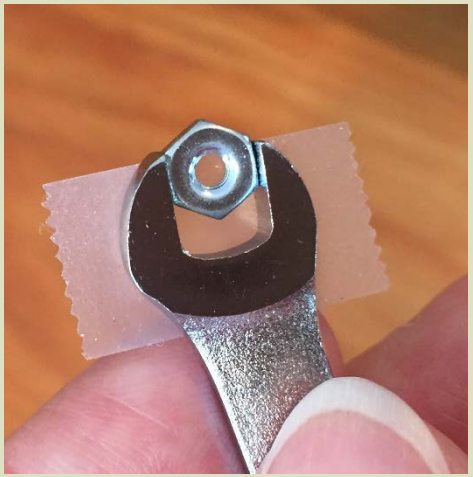
Insert motor wires, then motor, with wires toward front



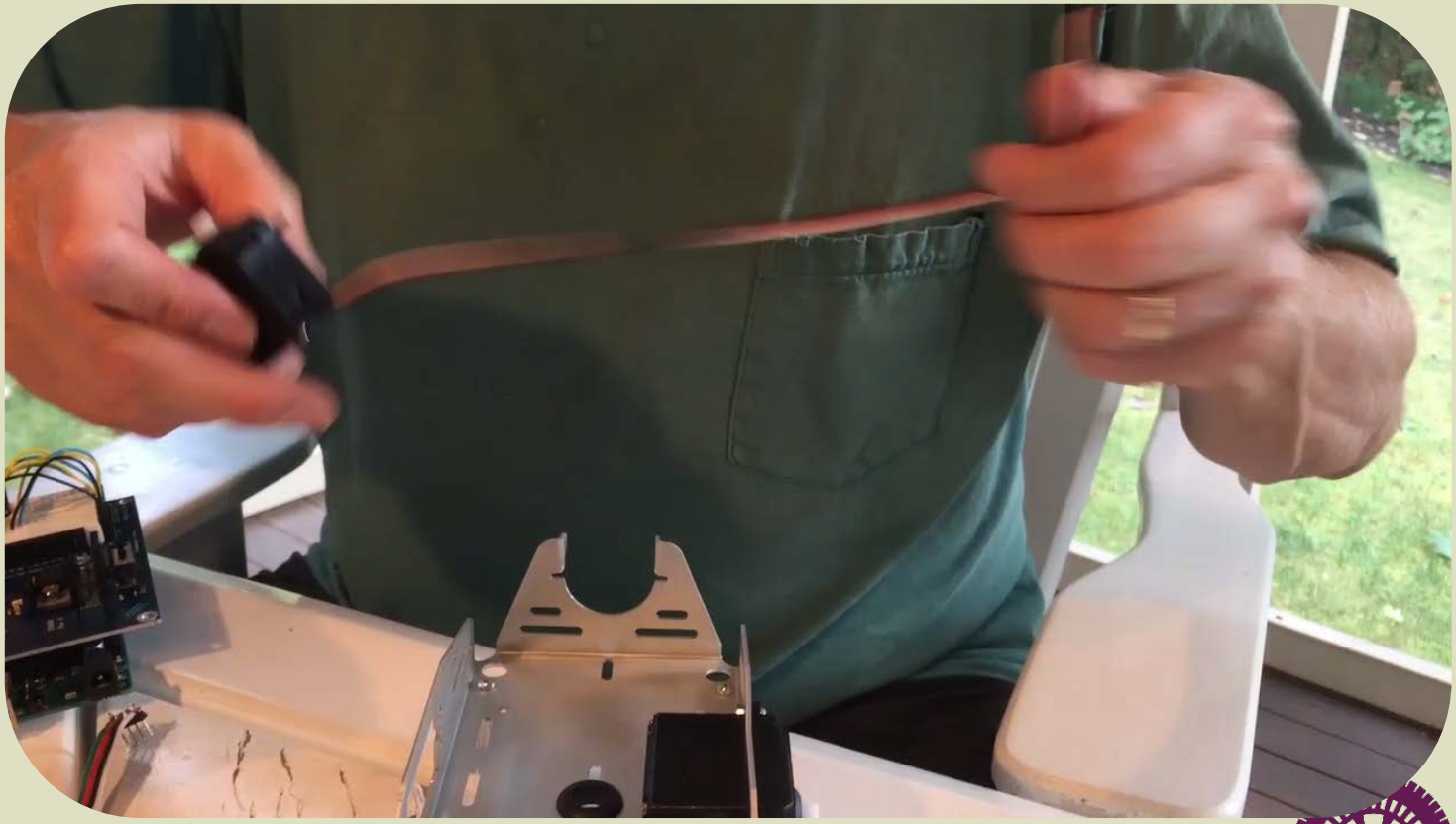
Secure motor with four 3/8” screws (second smallest) and four nuts



4 Ways to Hold Nuts



Repeat for 2nd Motor



Insert Battery Pack

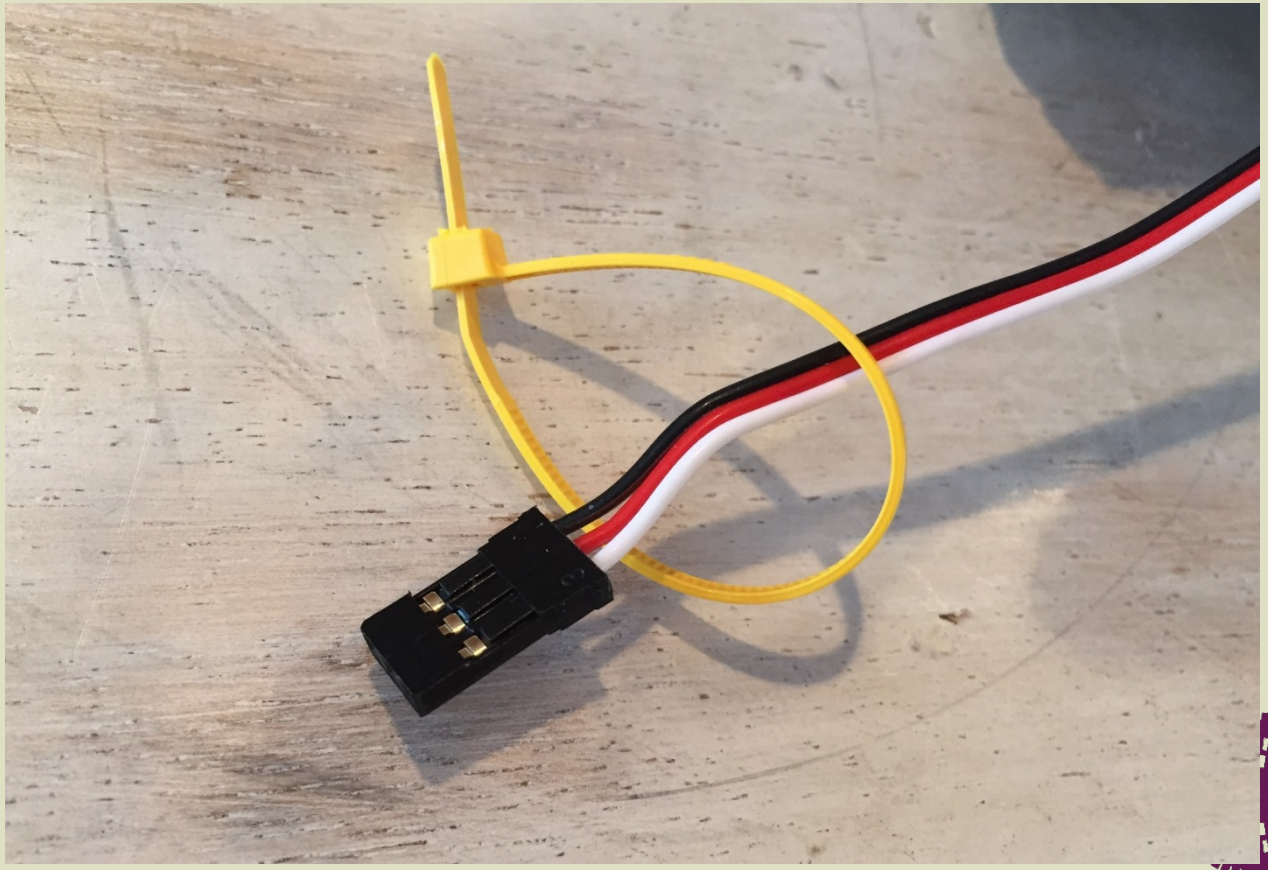
wire toward center

Secure Battery Pack using two 3/8” screws into two 1” threaded round posts

Load Batteries

(1st, middle, & last point to center)

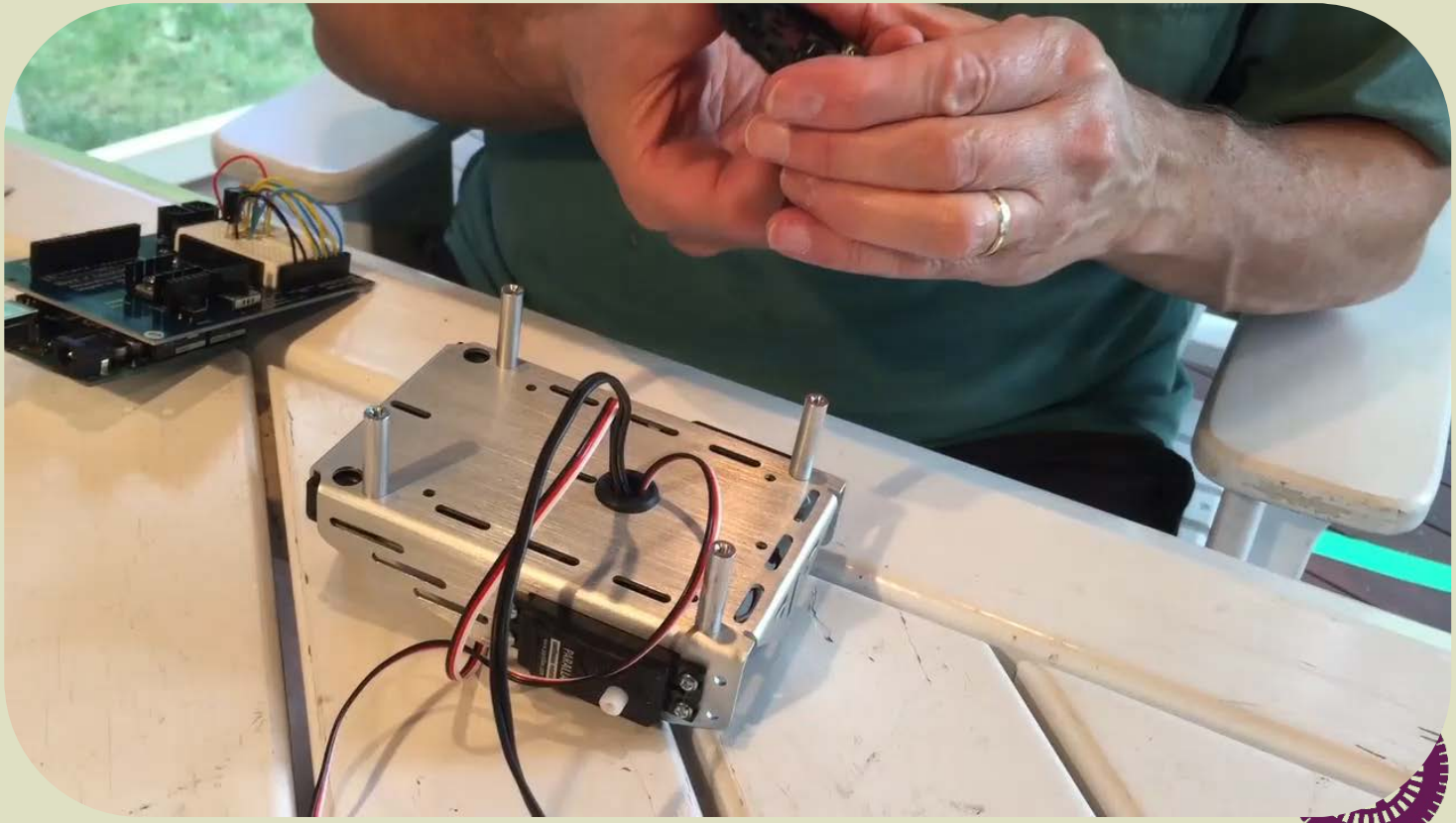
Mark the Left Motor's Wire



Thread Three Cables through Center Hole

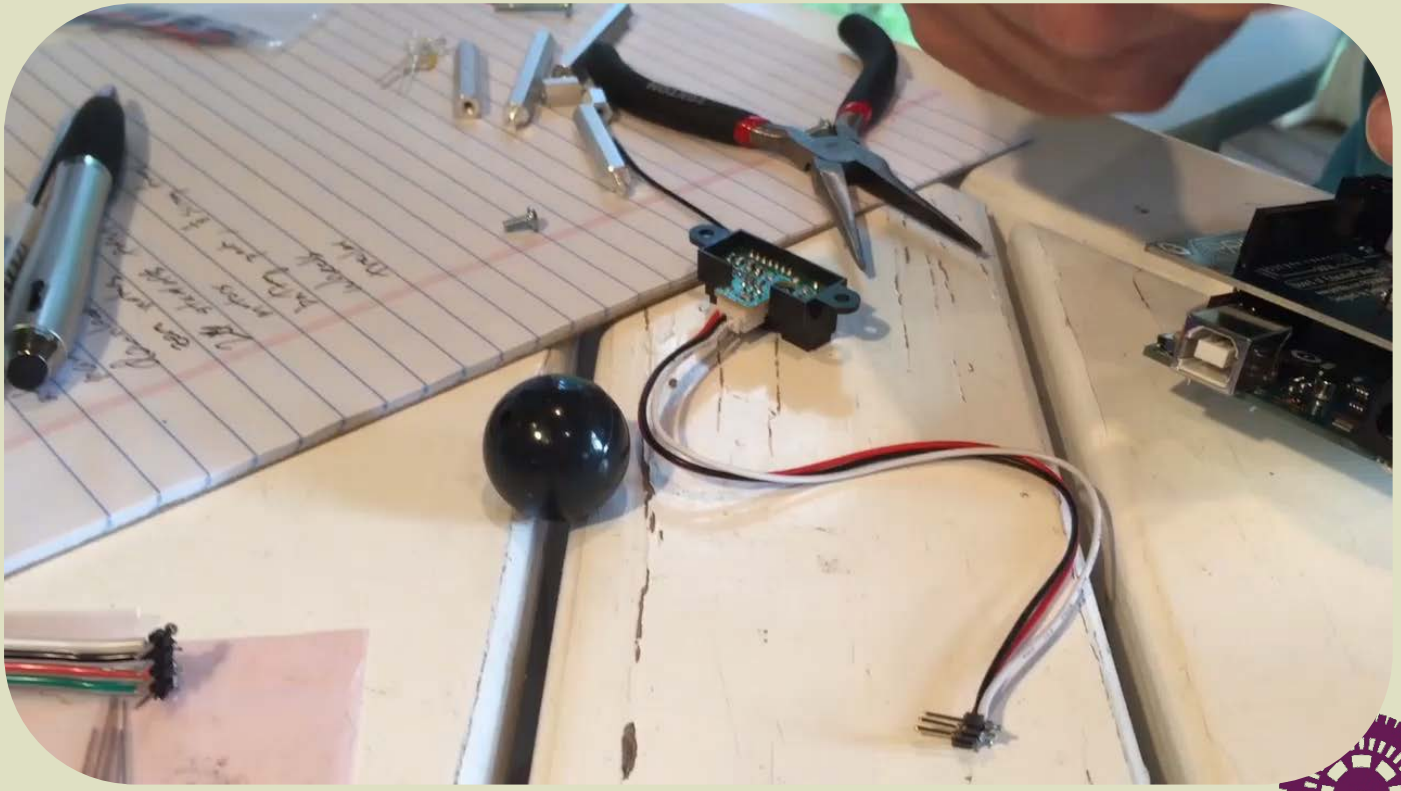
Tires onto Wheels

Pop Wheels onto Motor Shafts

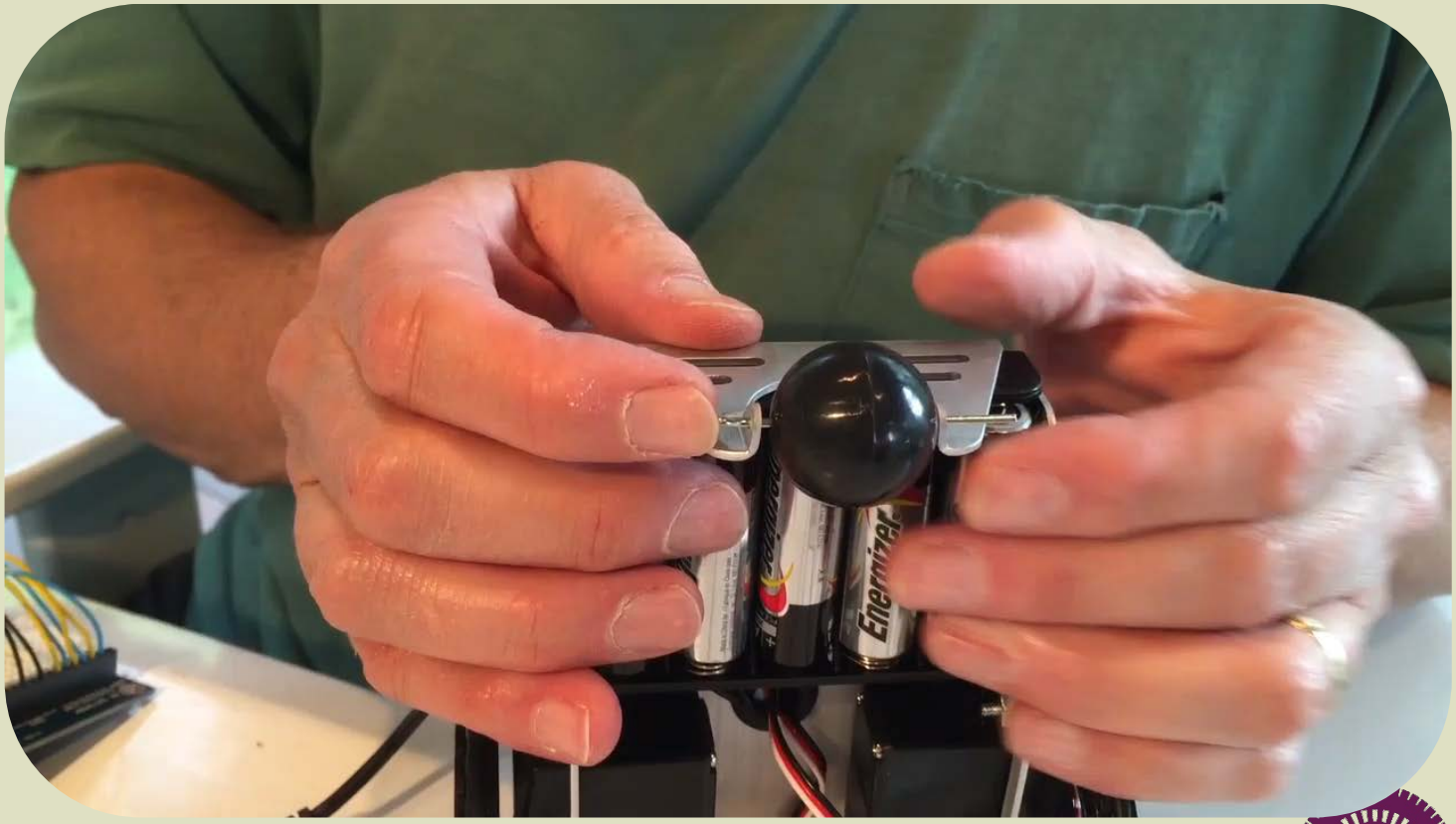


Secure wheels using screws that held arms onto motors

Thread Pin Through Rear Chassis & Ball

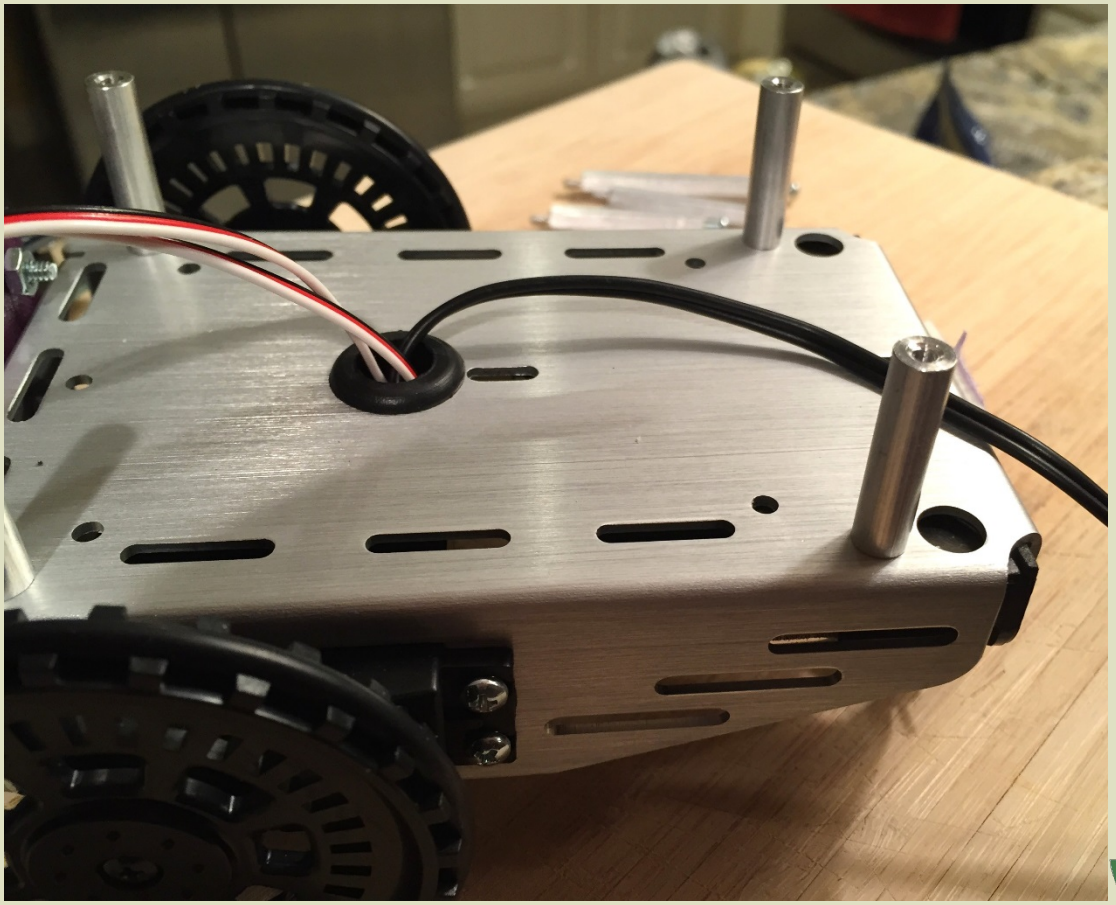


Bend Pin Legs Out

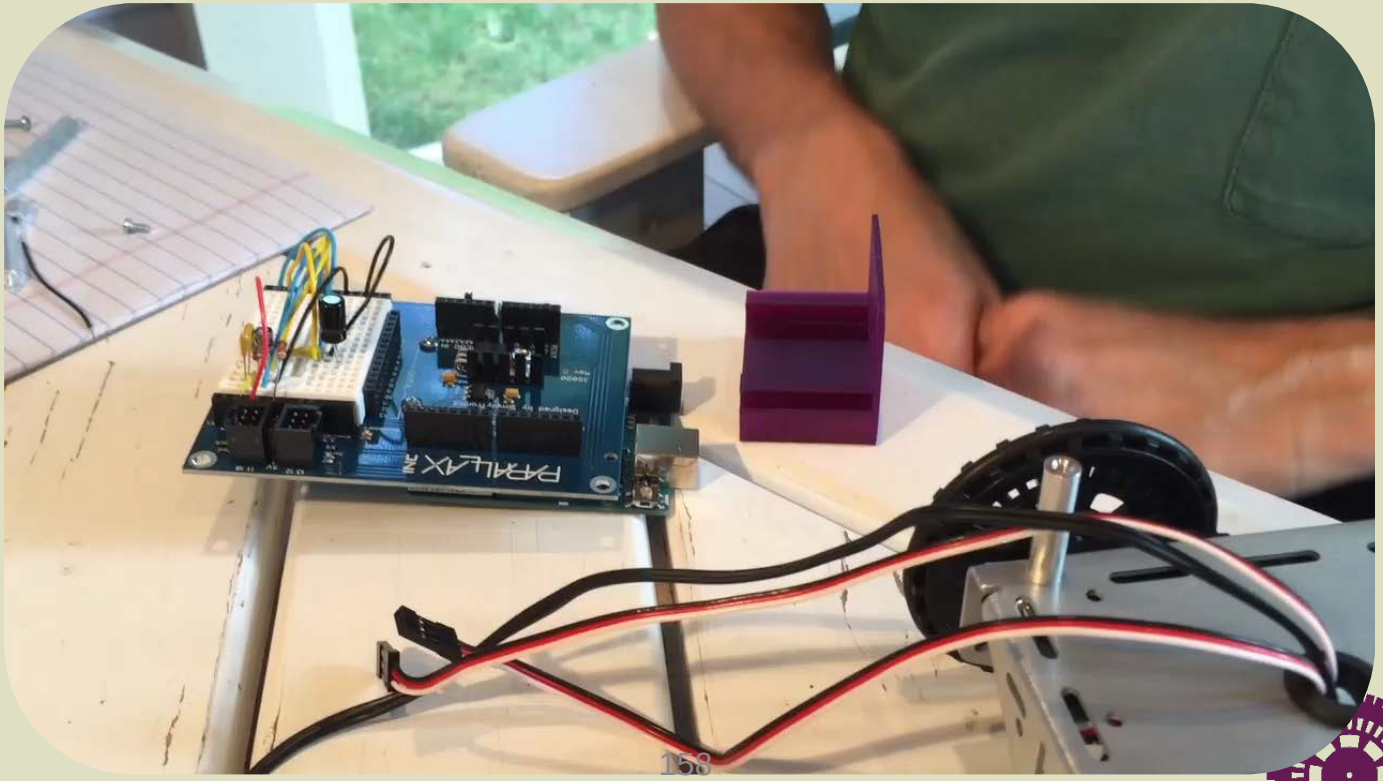


Drape red/white/black motor cables out between front wheels.

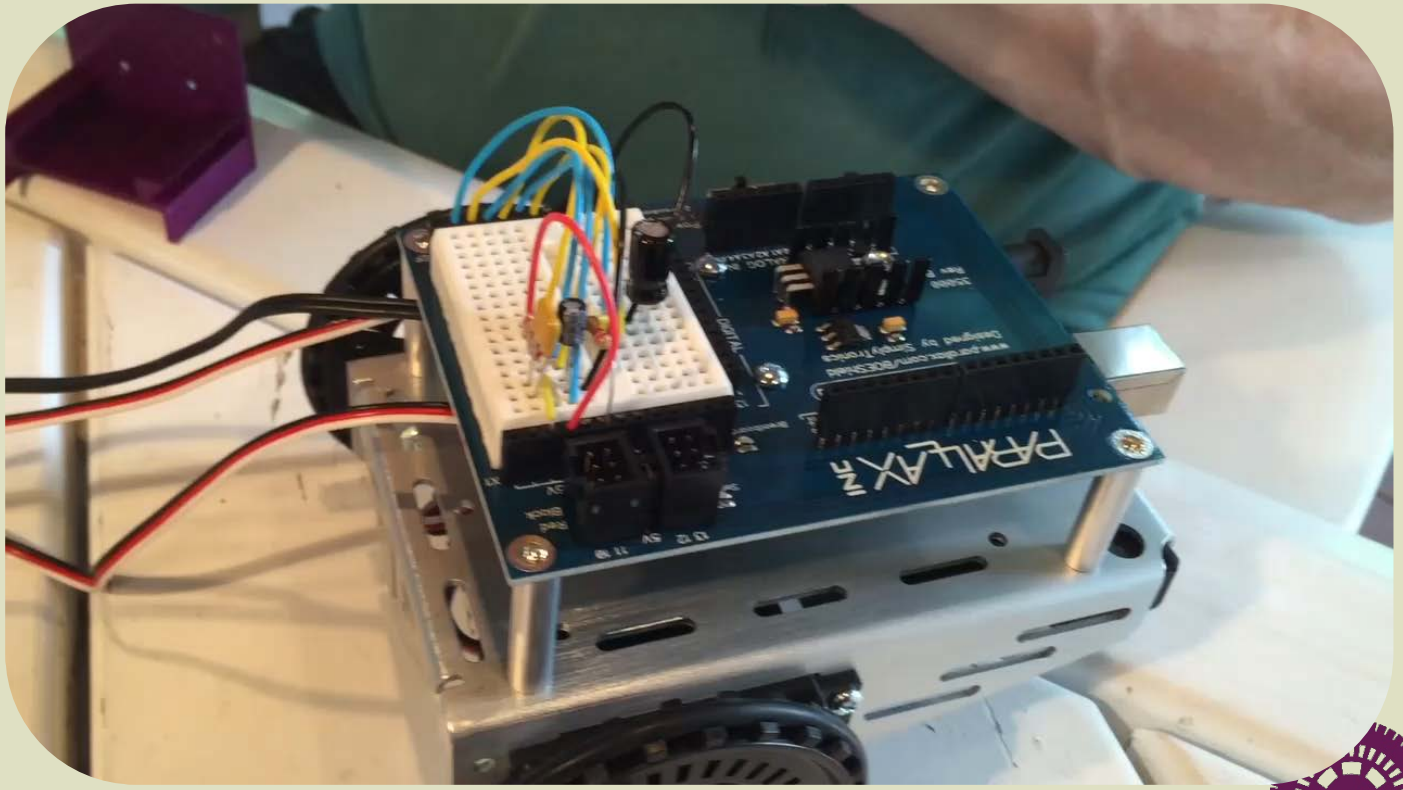
Drape black battery cord toward the back ball.



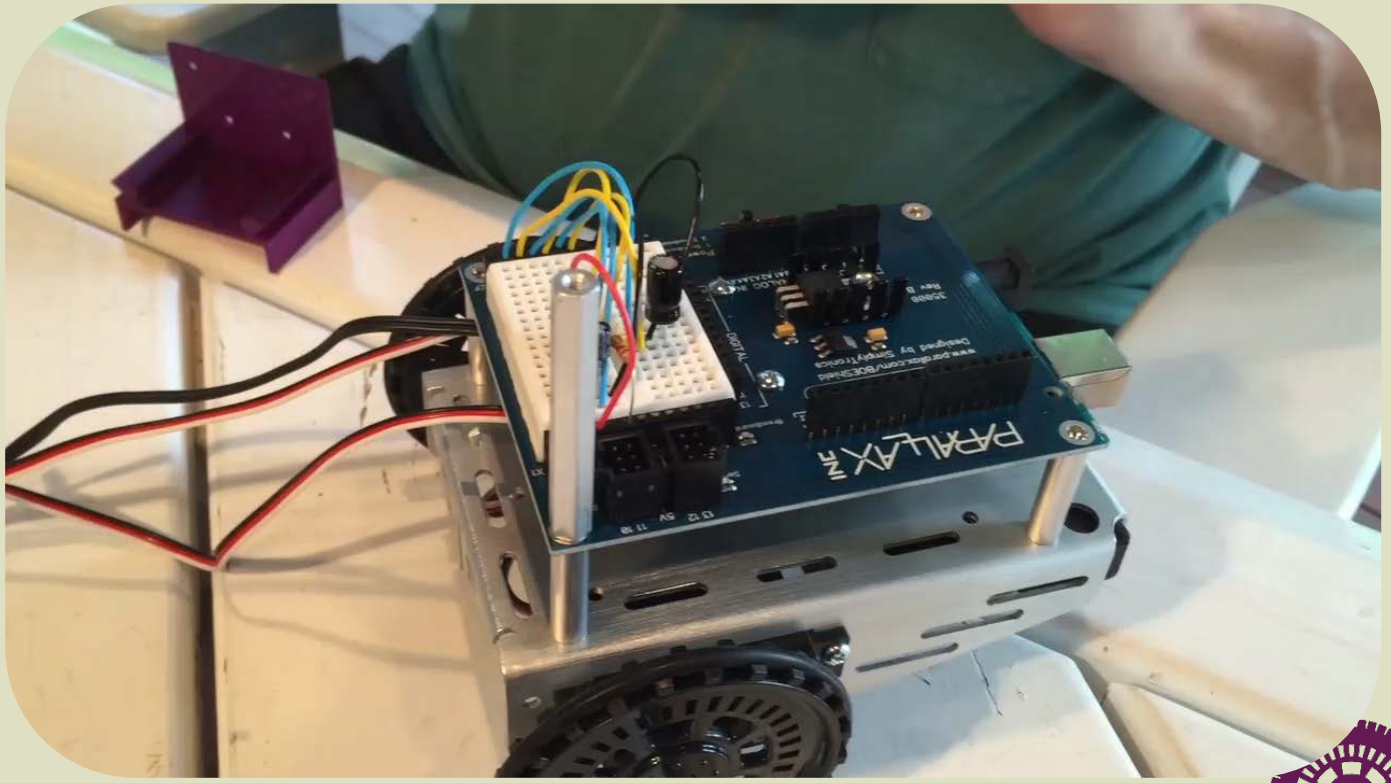
Set Electronics on Posts
(breadboard toward front wheels)



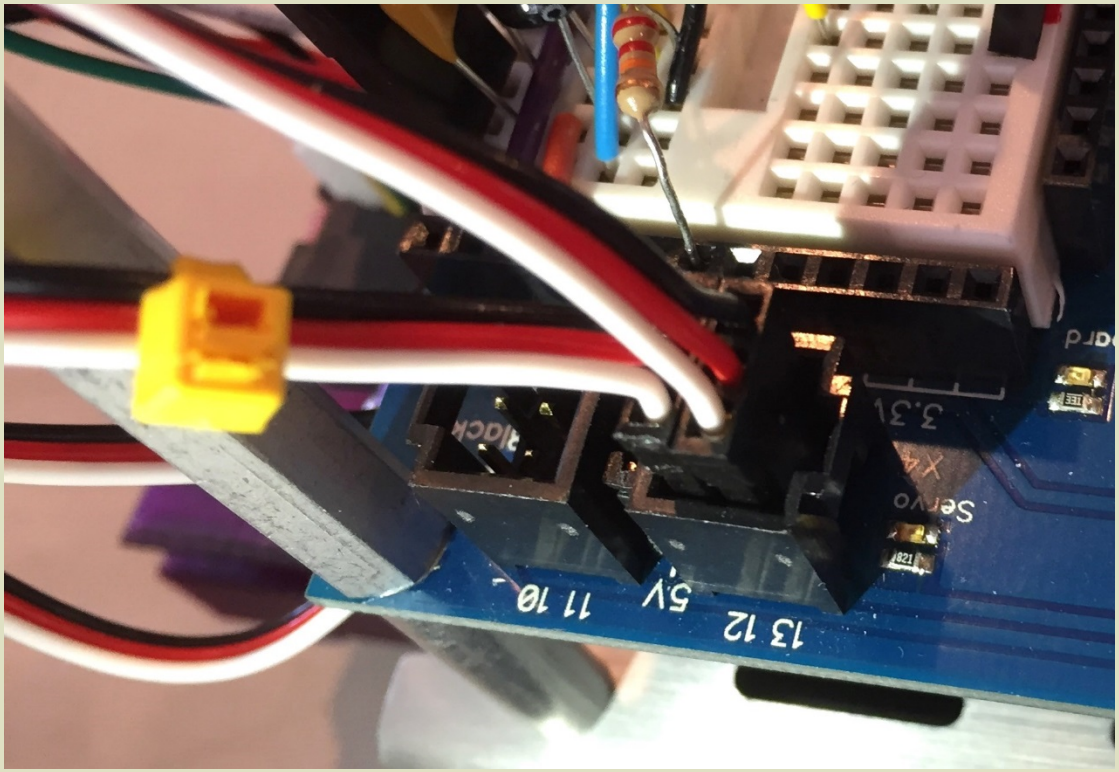
Screw in two long octagonal posts over front wheels



Add Mini Octagonal Posts
in rear corners



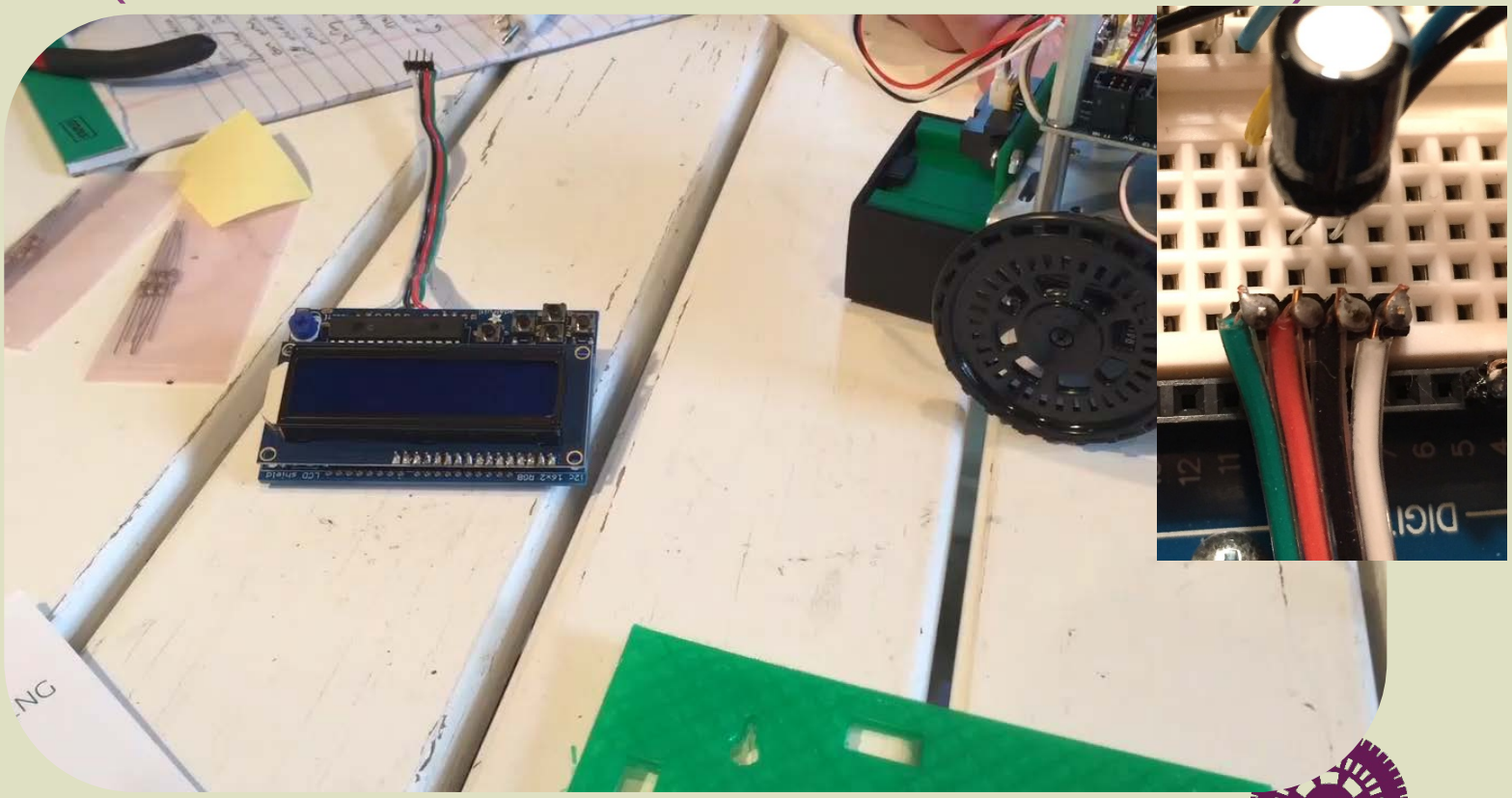
Re-Plug Motors, Left Motor in #12



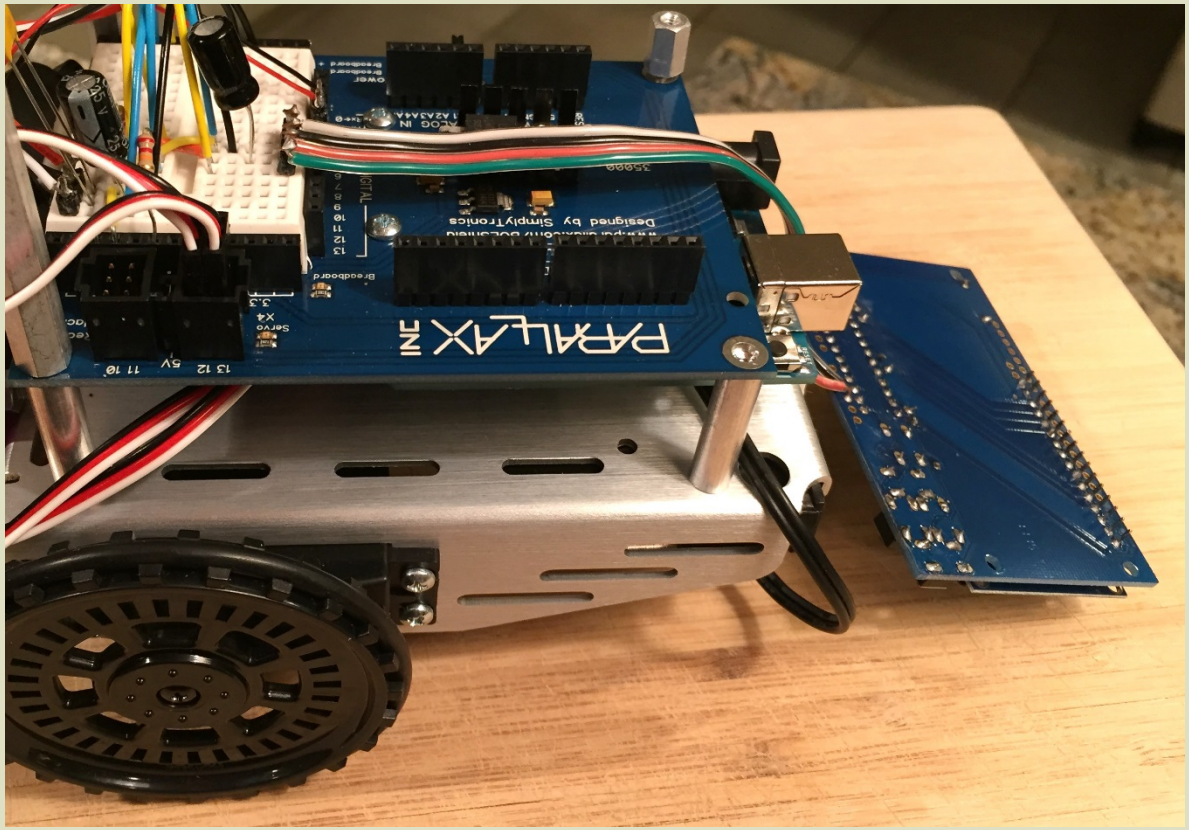
gizmo garden©

Plug in Dashboard

(Mind the Power and Ground marks)



Ensure Dashboard is resting face down toward the rear

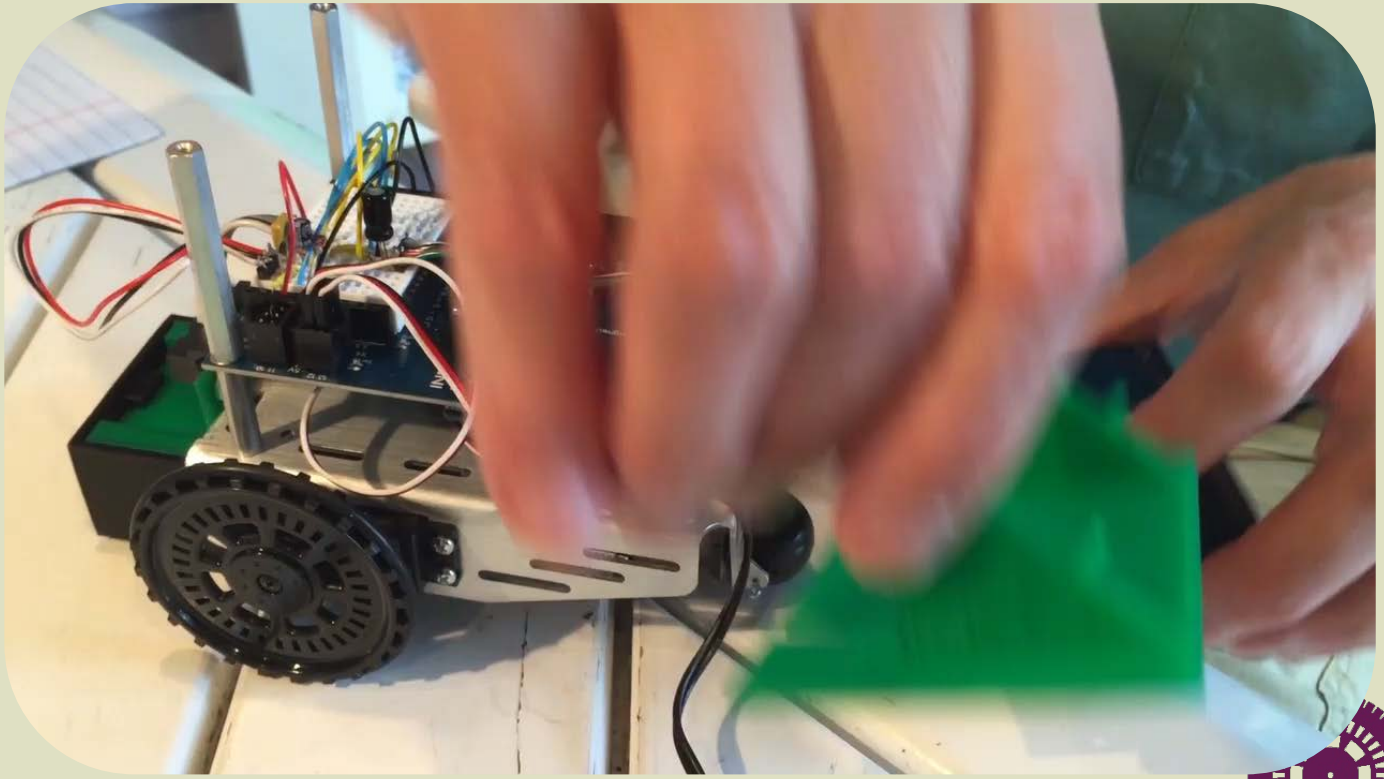


gizmo garden©

Select Dash Holder

Mount Dash Holder

using medium octagonal posts

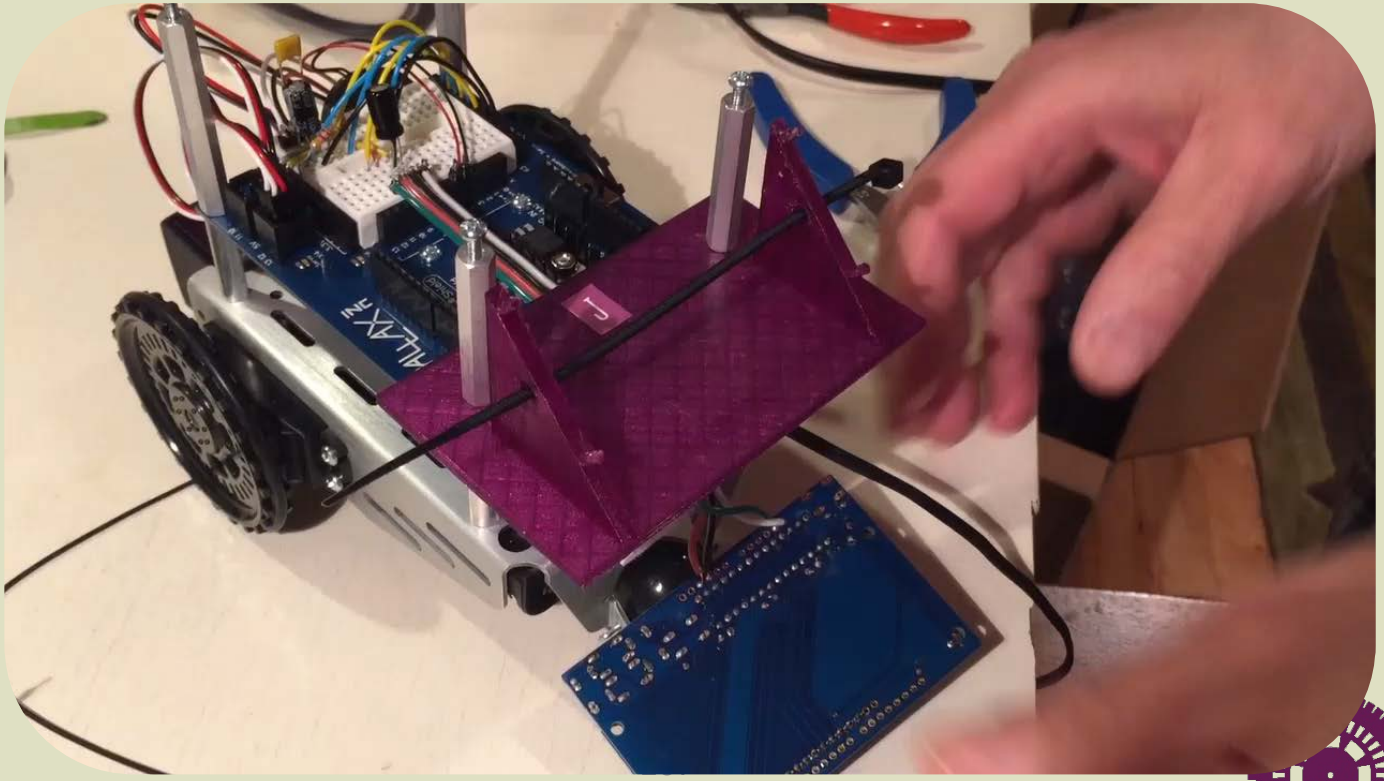


Thread a tie wrap through side holes in dash holder

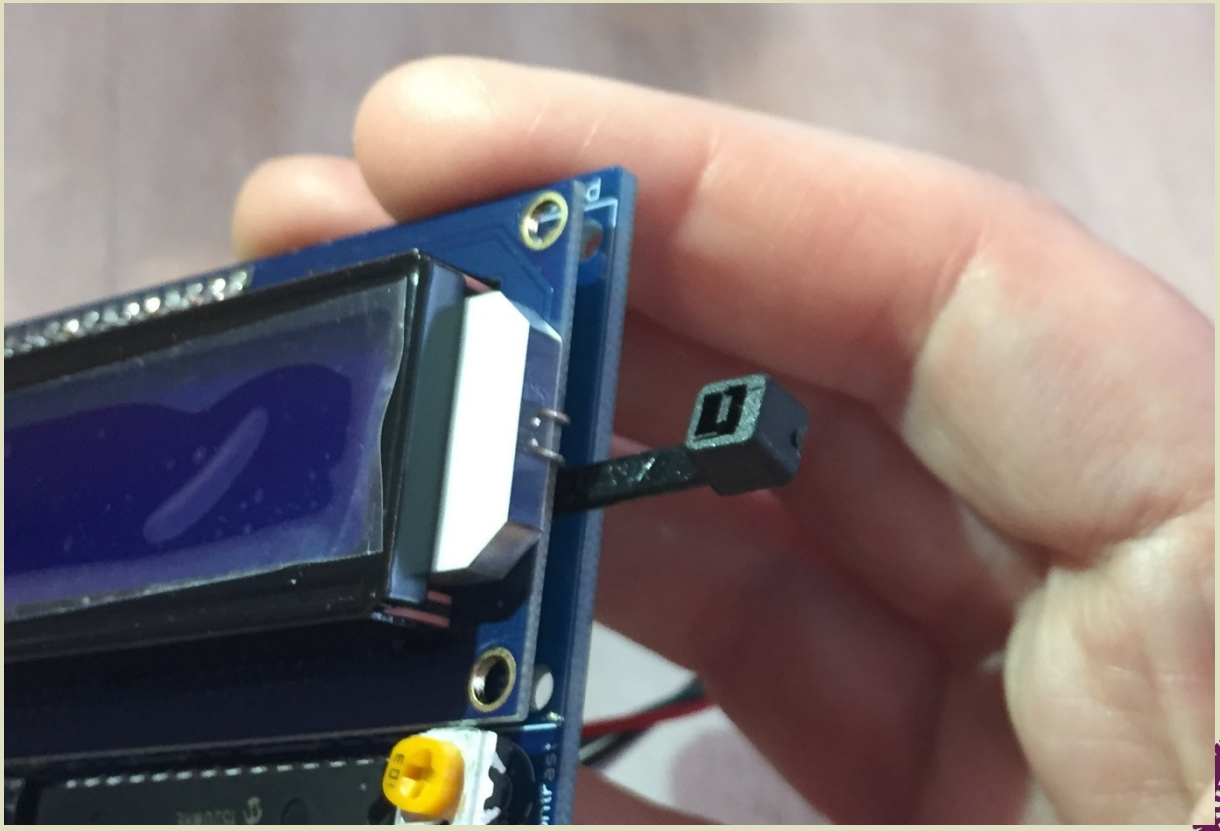
Correct: ridges on top

Incorrect: writing on top

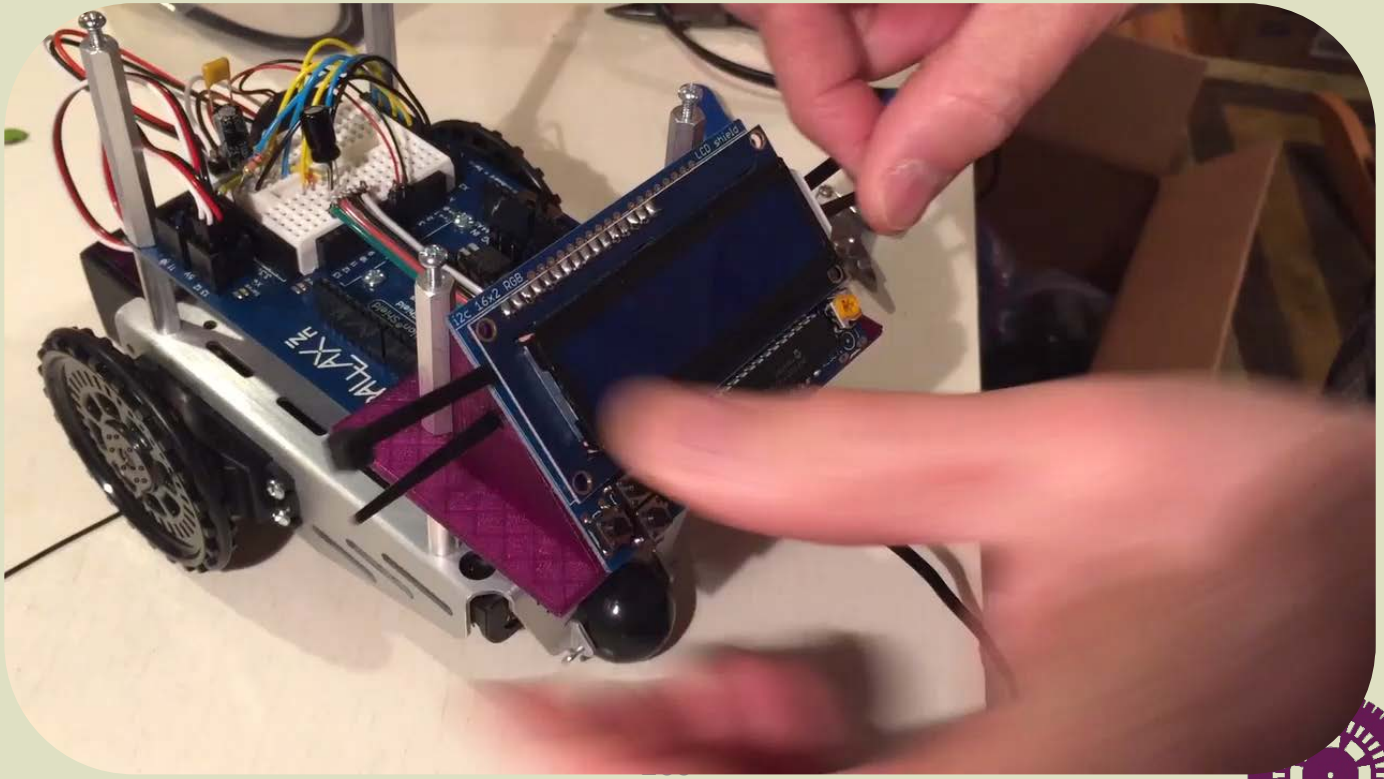
Snap Dashboard onto plastic pegs



Slide an opposite-orientation tie-wrap between the display and the circuit board

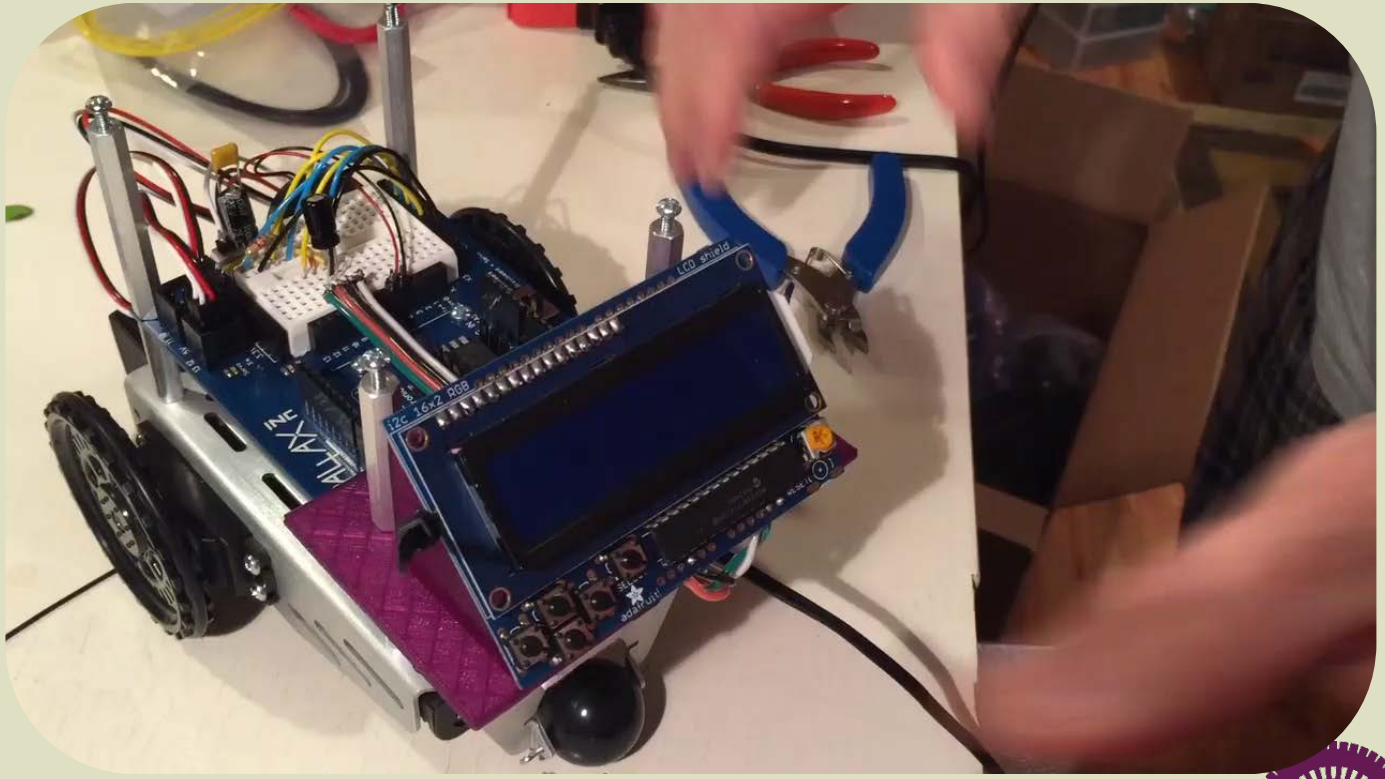


Tighten Tie Wraps

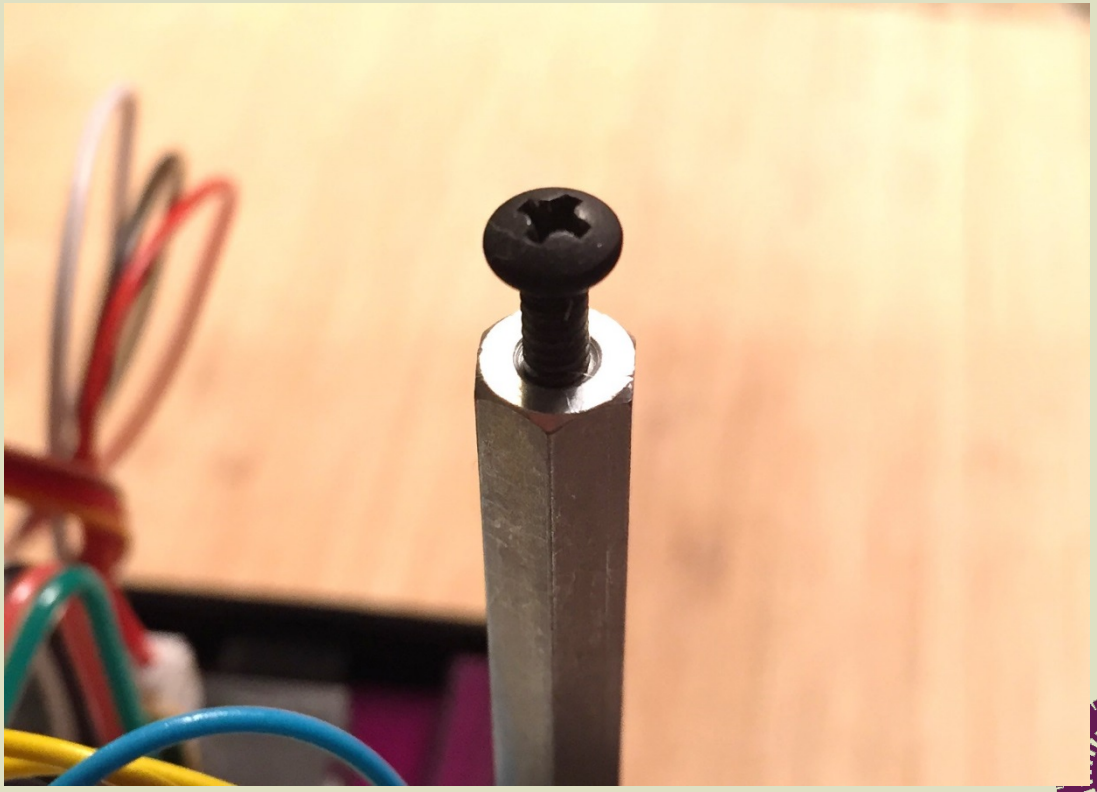


gizmo garden©

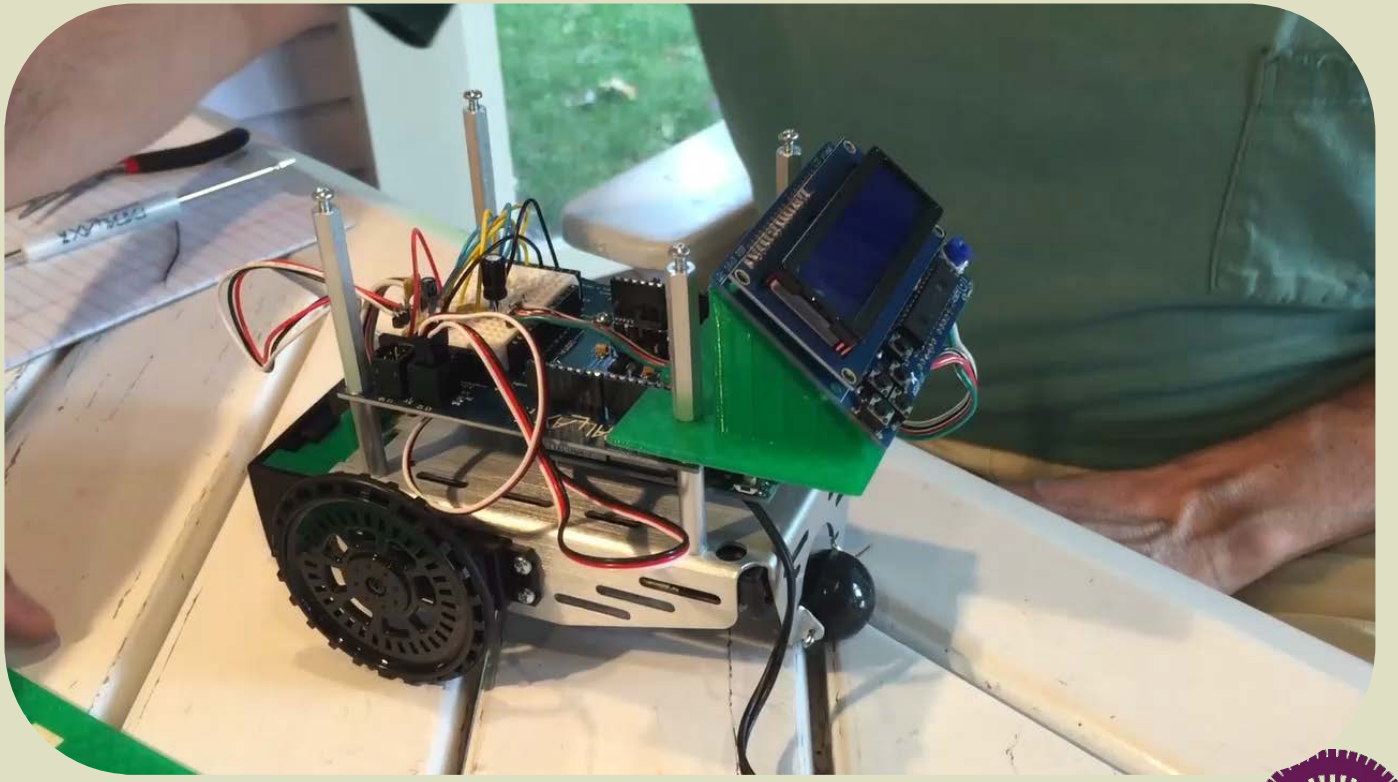
Clip Tie Wrap Ends



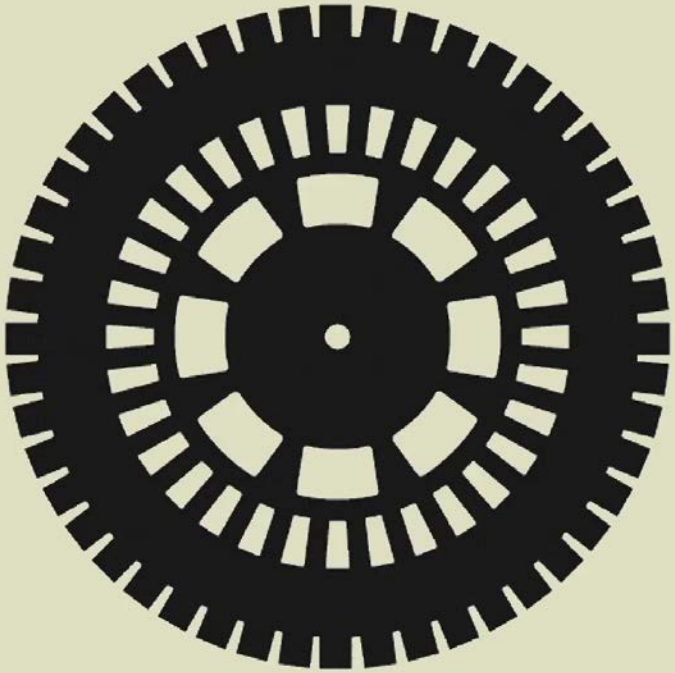
Insert black screws part way into posts



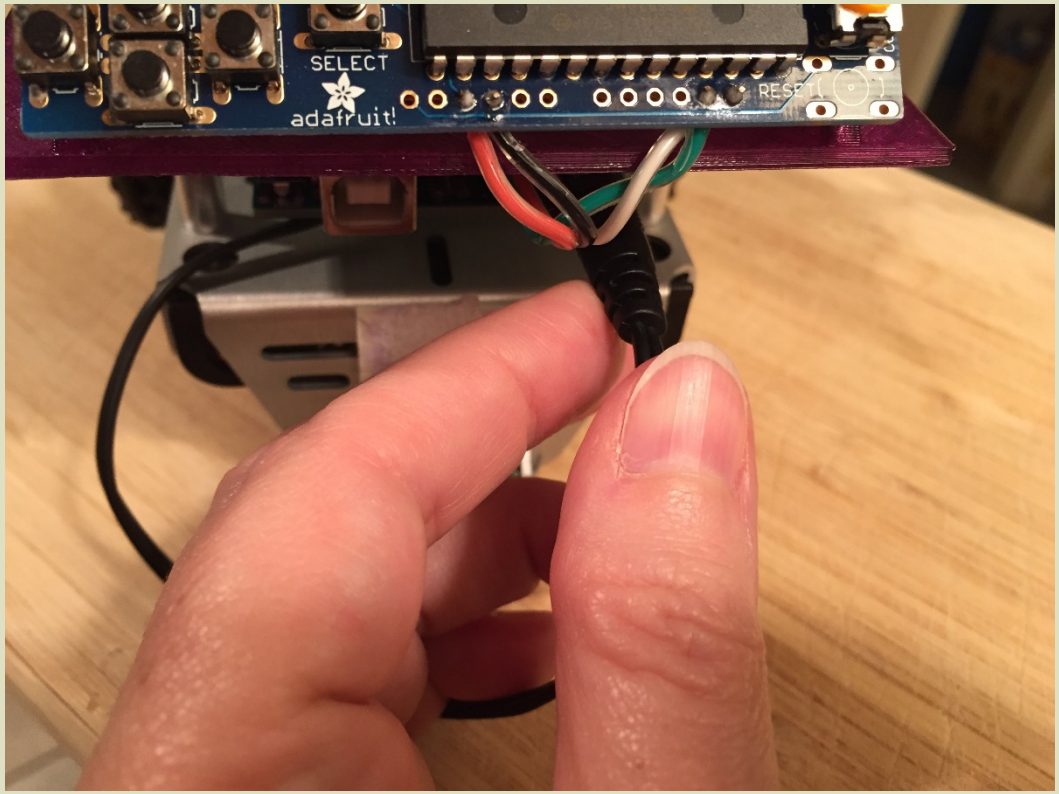
Mount Platform



Ready to Roll!

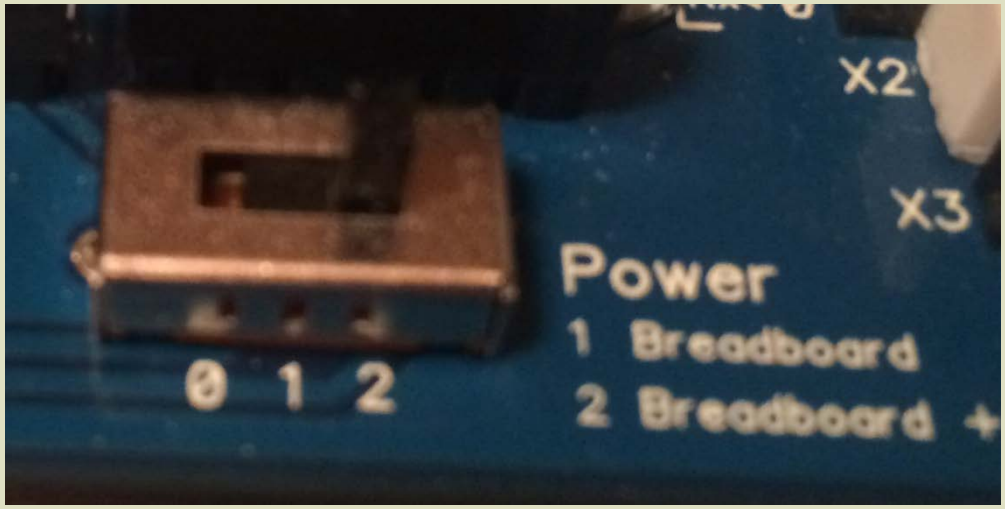


Plug in the Gizmo battery pack



gizmo garden©

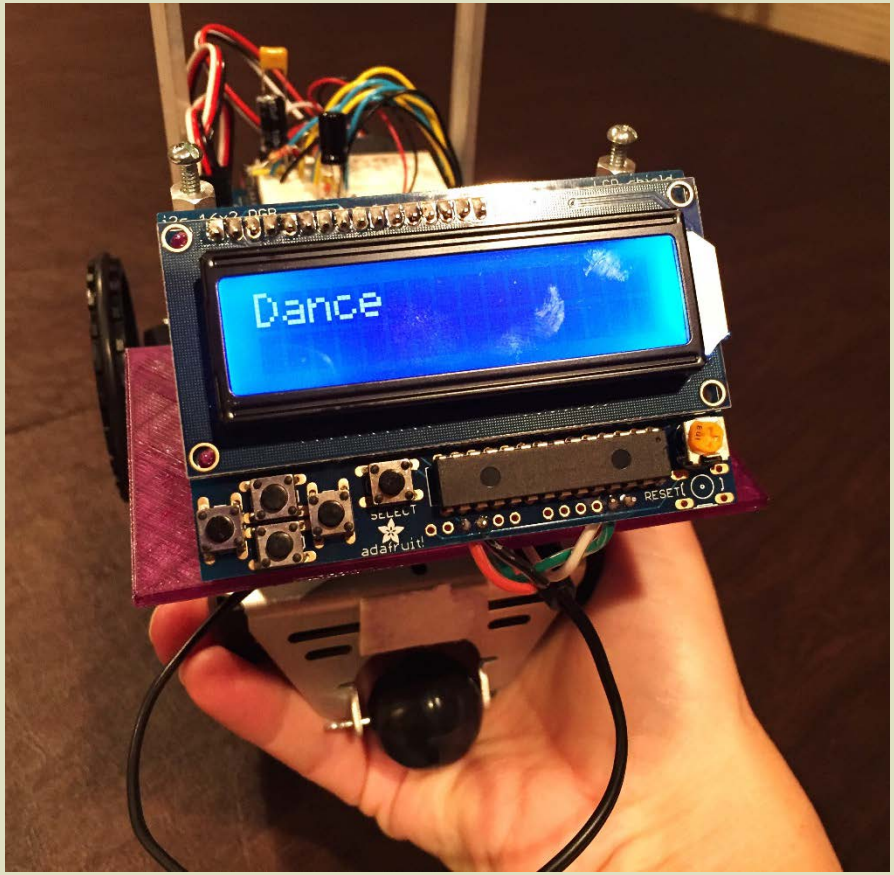
Power to “2”



Scroll Left-Right to Dance Party

gizmo garden©

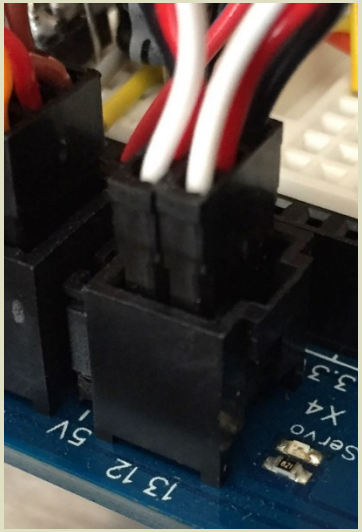
Hold Gizmo so front drive wheels can Spin



On Dashboard, Push Select

Did Both Wheels go Forward?

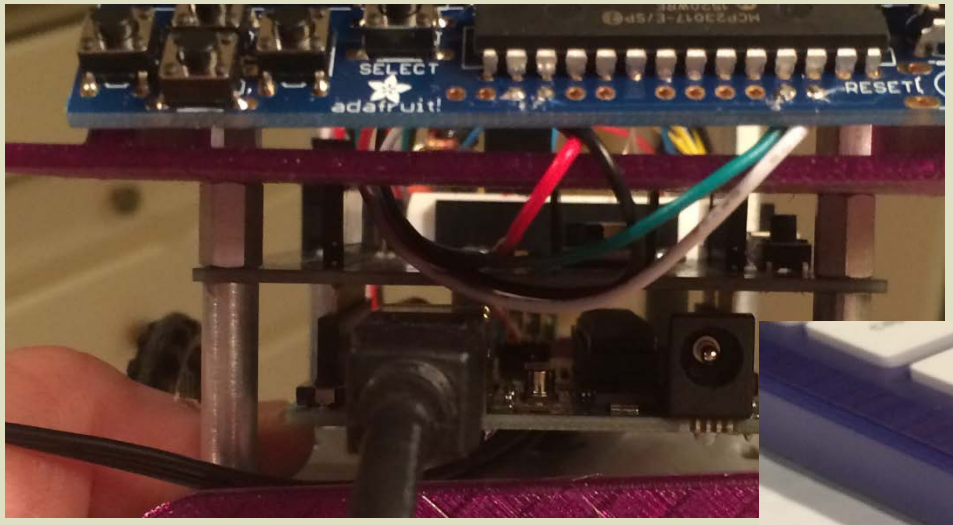
- If yes, you're good to go.
- If both wheels went backward, transfer the cable going into input 12 to the input 13 slot and vice versa.



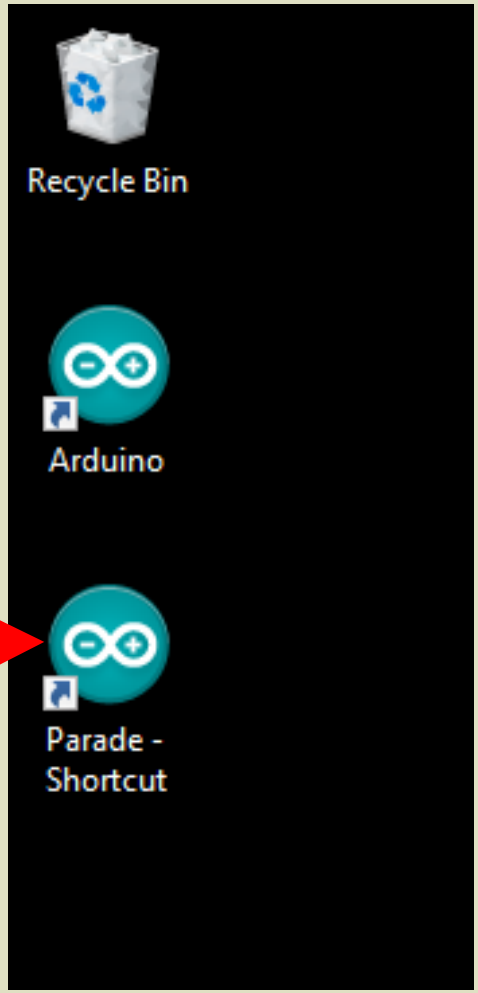
gizmo garden©

Clear desk space. Hit select with gizmo on desk

Connect Gizmo to Computer

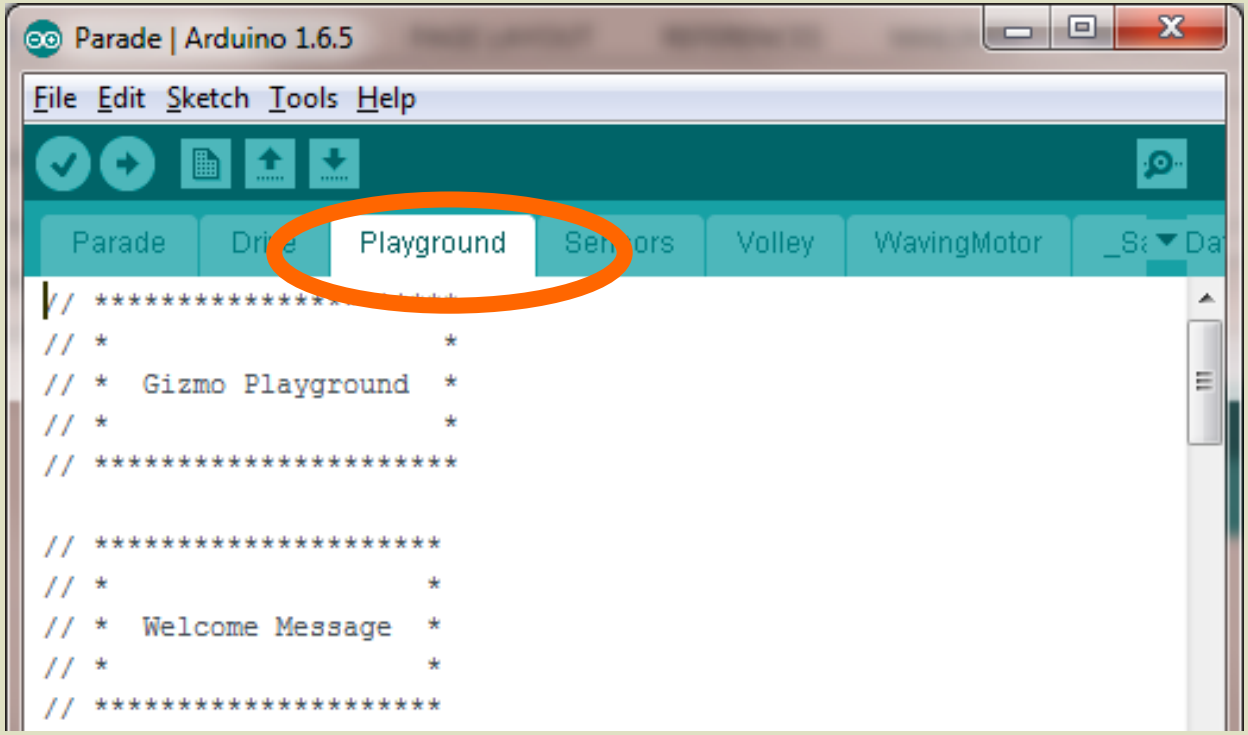


Open Arduino App



Double-click

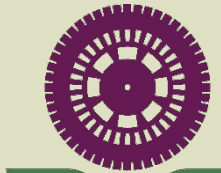
Playground Tab



gizmo garden©

Scroll down to Dance Party

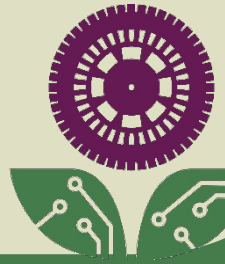
```
// ******************
// *                *
// *  Dance Party   *
// *                *
// ******************

void danceParty()
{
  leftWheel.setSpeed(75);
  rightWheel.setSpeed(75);
  delay(250);

  leftWheel.setSpeed(0);
  rightWheel.setSpeed(0);
}
```



Dance/Beep Similar Function Definition

```
void beepParty()
{
  GizmoGardenTone(2400, 250);
  delay(400);
}
```

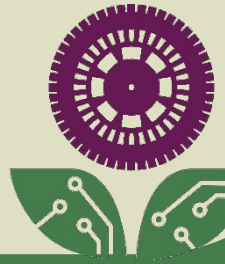
```
void danceParty()
{
  leftWheel.setSpeed(75);
  rightWheel.setSpeed(75);
  delay(250);

  leftWheel.setSpeed(0);
  rightWheel.setSpeed(0);
}
```



Dance/Beep Similar Function Curly Braces

```
void beepParty()
{
  GizmoGardenTone(2400, 250);
  delay(400);
}
```

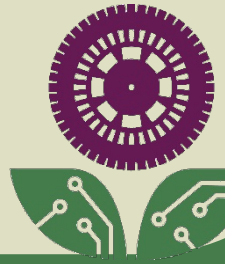
```
void danceParty()
{
  leftWheel.setSpeed(75);
  rightWheel.setSpeed(75);
  delay(250);

  leftWheel.setSpeed(0);
  rightWheel.setSpeed(0);
}
```



Dance/Beep
Similar use of delay

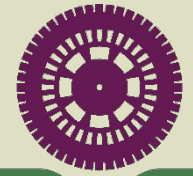
```
void beepParty()
{
  GizmoGardenTone(2400, 250);
  delay(400);
}
```

```
void danceParty()
{
  leftWheel.setSpeed(75);
  rightWheel.setSpeed(75);
  delay(250);

  leftWheel.setSpeed(0);
  rightWheel.setSpeed(0);
}
```



Dance has two new functions

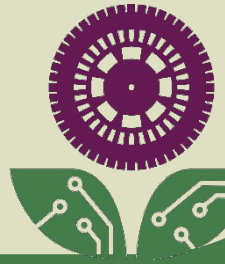
```
void danceParty()
{
  leftWheel.setSpeed(75);
  rightWheel.setSpeed(75);
  delay(250);

  leftWheel.setSpeed(0);
  rightWheel.setSpeed(0);
}
```



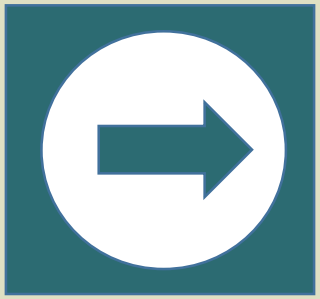
Try Making % Speed Negative

- Place gizmo near front of desk, rear dashboard facing you

```
void danceParty()
{
  leftWheel.setSpeed(-75);
  rightWheel.setSpeed(-75);
  delay(250);

  leftWheel.setSpeed(0);
  rightWheel.setSpeed(0);
}
```



Upload, Scroll Left-Right to Dance Party

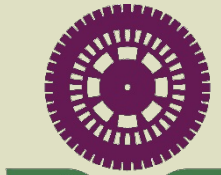


Push Select

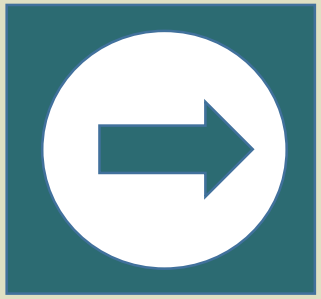
- Make sure your gizmo has room to back up!

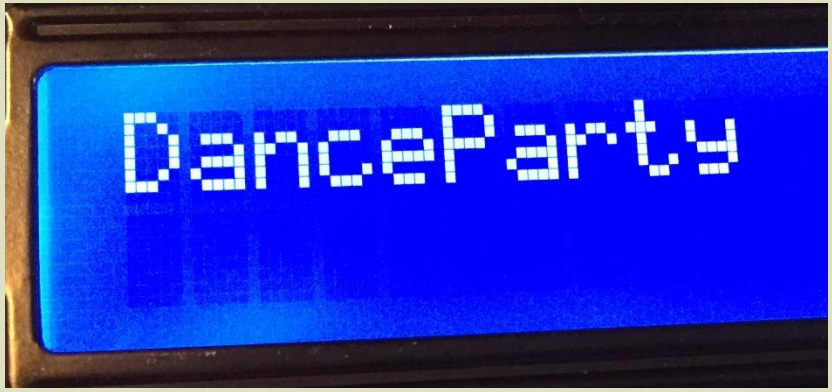
Try Making One Speed Positive, One Negative

```
void danceParty()
{
  leftWheel.setSpeed(-75);
  rightWheel.setSpeed(75);
  delay(250);

  leftWheel.setSpeed(0);
  rightWheel.setSpeed(0);
}
```



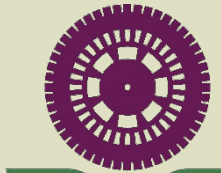
Upload, DanceParty, Select





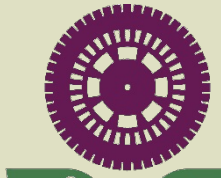
Unlike Beepers, Wheels must be halted

```
void danceParty()
{
  leftWheel.setSpeed(-75);
  rightWheel.setSpeed(75);
  delay(250);

  leftWheel.setSpeed(0);
  rightWheel.setSpeed(0);
}
```

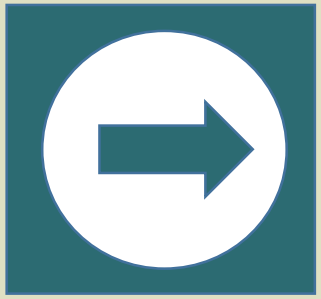


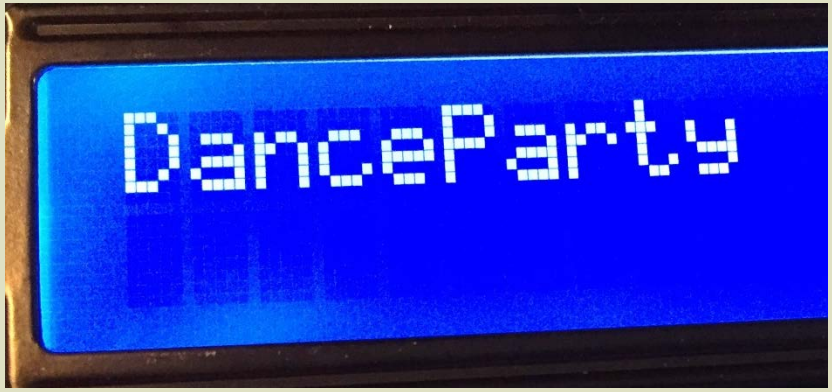
Insert 3 lines to twist back the other way

```
void danceParty()
{
  leftWheel.setSpeed(-75);
  rightWheel.setSpeed(75);
  delay(250);
  leftWheel.setSpeed(75);
  rightWheel.setSpeed(-75);
  delay(250);

  leftWheel.setSpeed(0);
  rightWheel.setSpeed(0);
}
```



Upload, DanceParty, Select





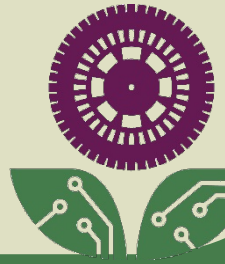
Code Reference #12

Motion

- Turns the Gizmo wheels at a percentage of their maximum speed

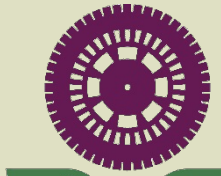
```
leftWheel.setSpeed (%)
rightWheel.setSpeed (%)
```

- To stop the wheels, set their speed to zero

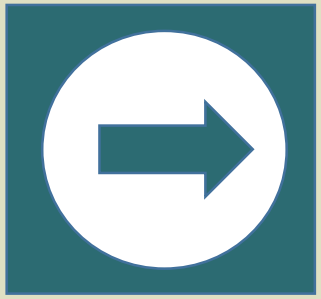


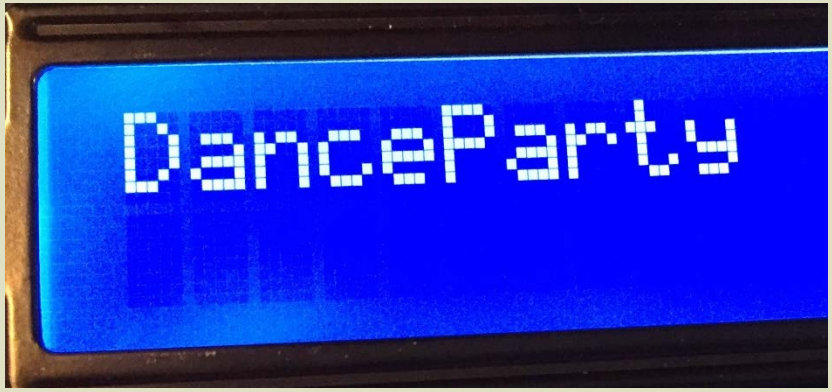
Loop it

```
void danceParty()
{
  for (int i = 0; i < 3; ++i)
  {
    leftWheel.setSpeed(-75);
    rightWheel.setSpeed(75);
    delay(250);
    leftWheel.setSpeed(75);
    rightWheel.setSpeed(-75);
    delay(250);
  }
  leftWheel.setSpeed(0);
  rightWheel.setSpeed(0);
}
```



Upload, DanceParty, Select

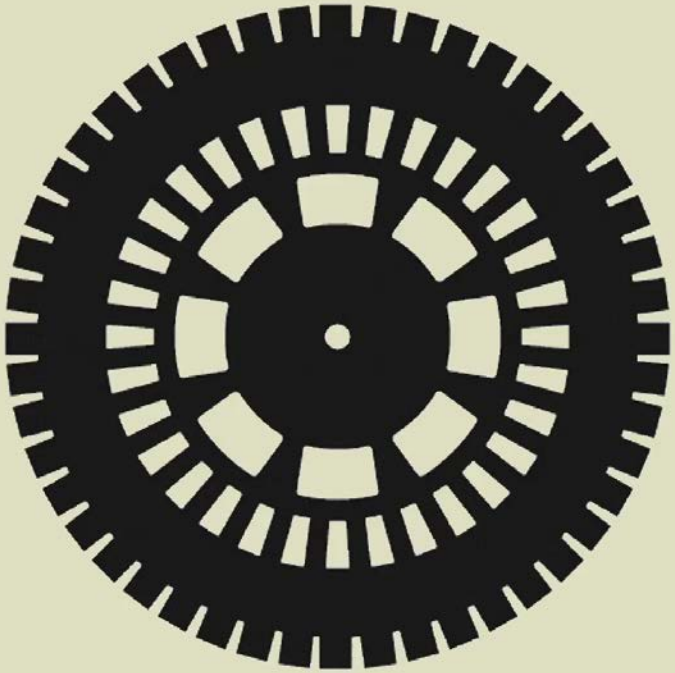




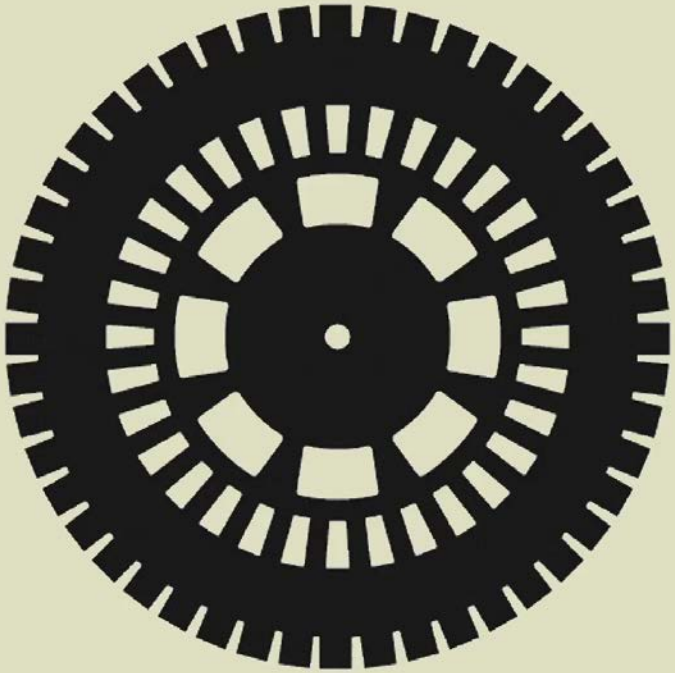
Take your computer & gizmo to a space on the floor & Create your gizmo Dance Party

gizmo garden©

Dancing Machine!

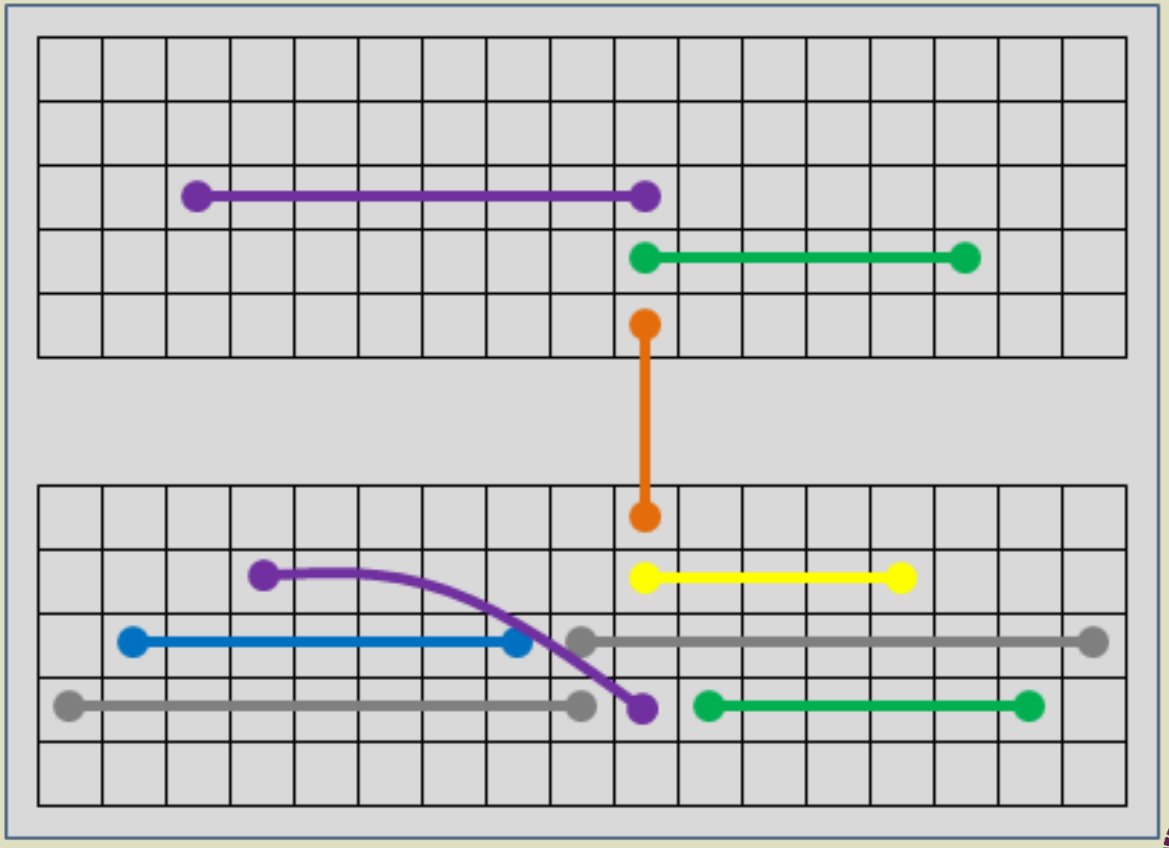


I Sense a Great Day!

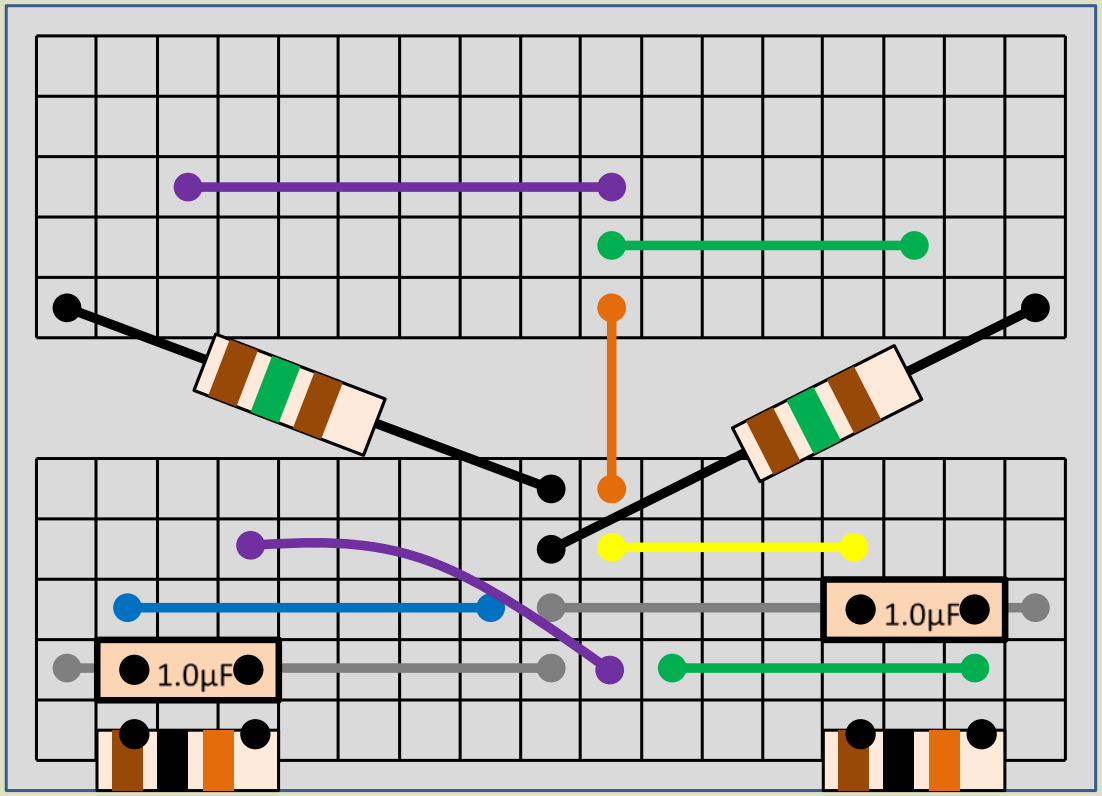


Safety Glasses

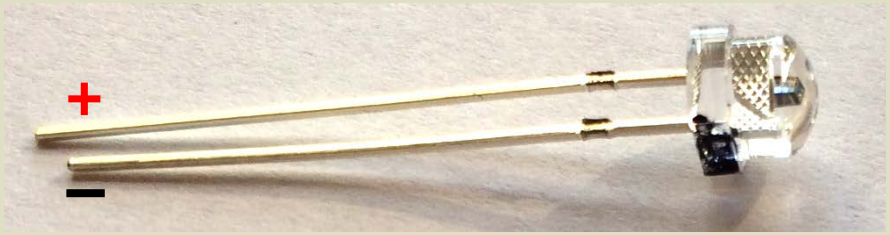
Sensor Board Wires



Sensor Board Components



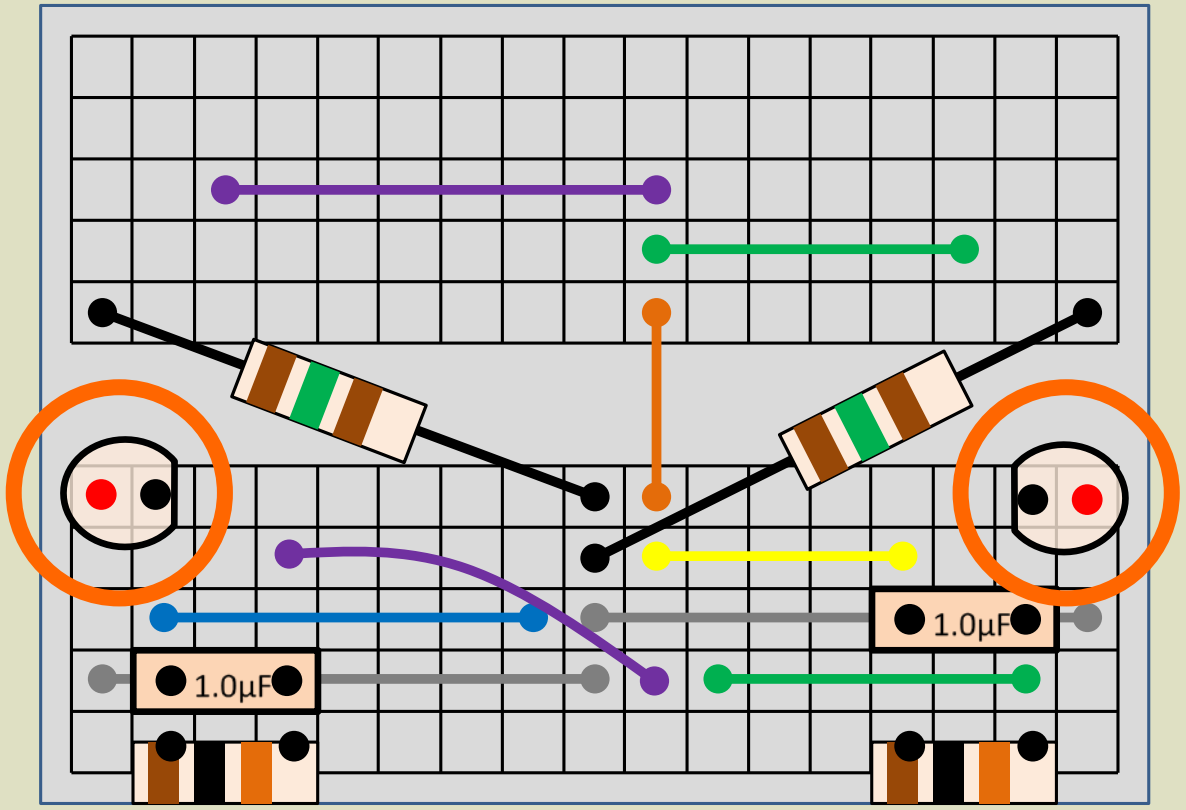
Mark Phototransistors



Use your black Sharpie to put a mark on the side with the shorter lead. This is the – side, and we mark it so that we know what side is – when we clip the leads to the same length.

Clip Phototransistor Leads

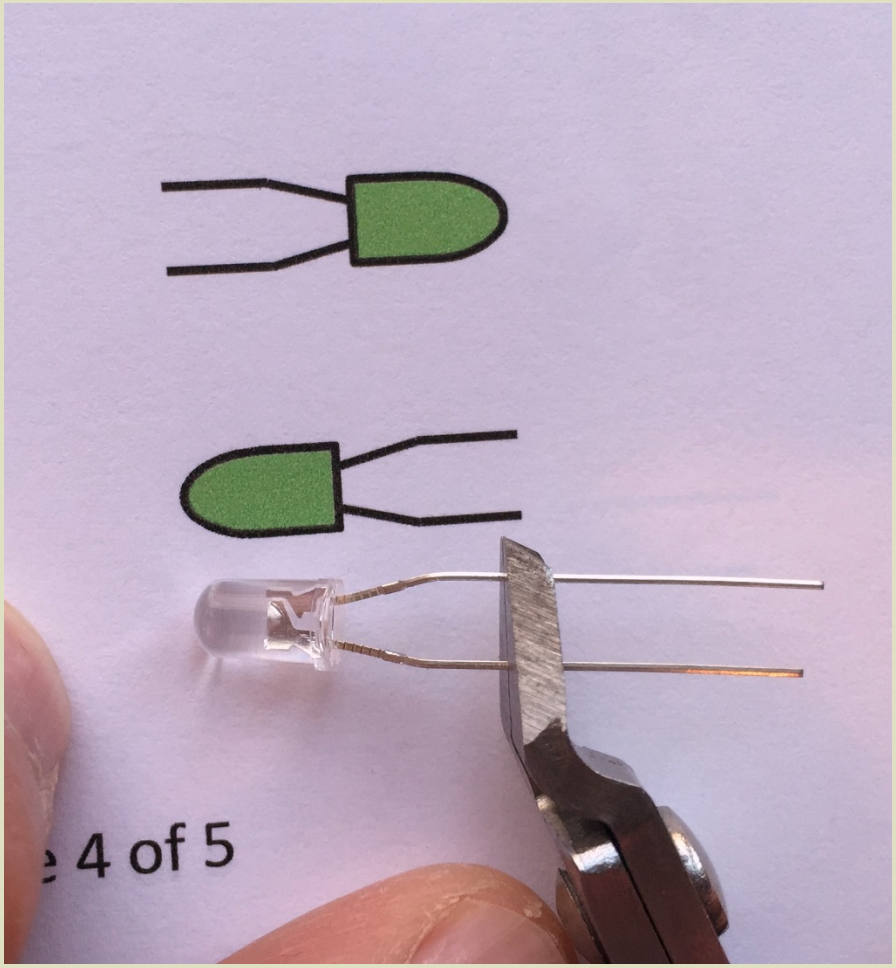
Insert Phototransistors



Mark LEDs

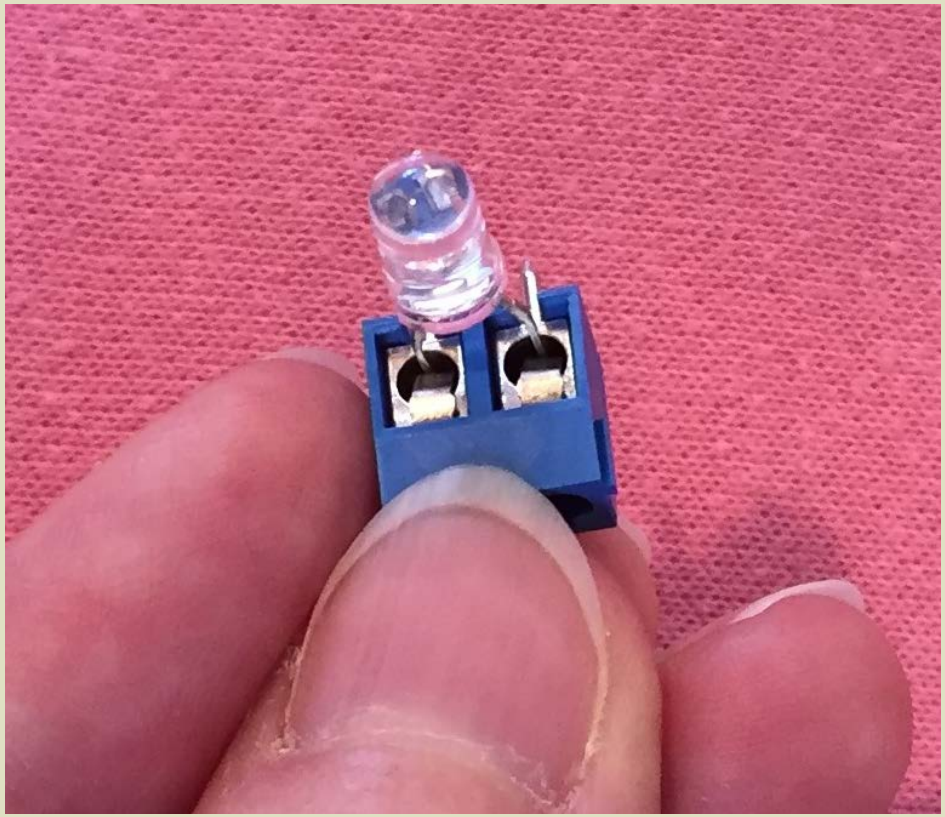
Use your black Sharpie to put a mark on the side with the shorter lead. This is the – side, and we mark it so that we know what side is – when we clip the leads to the same length.

Bend and Clip LED Leads

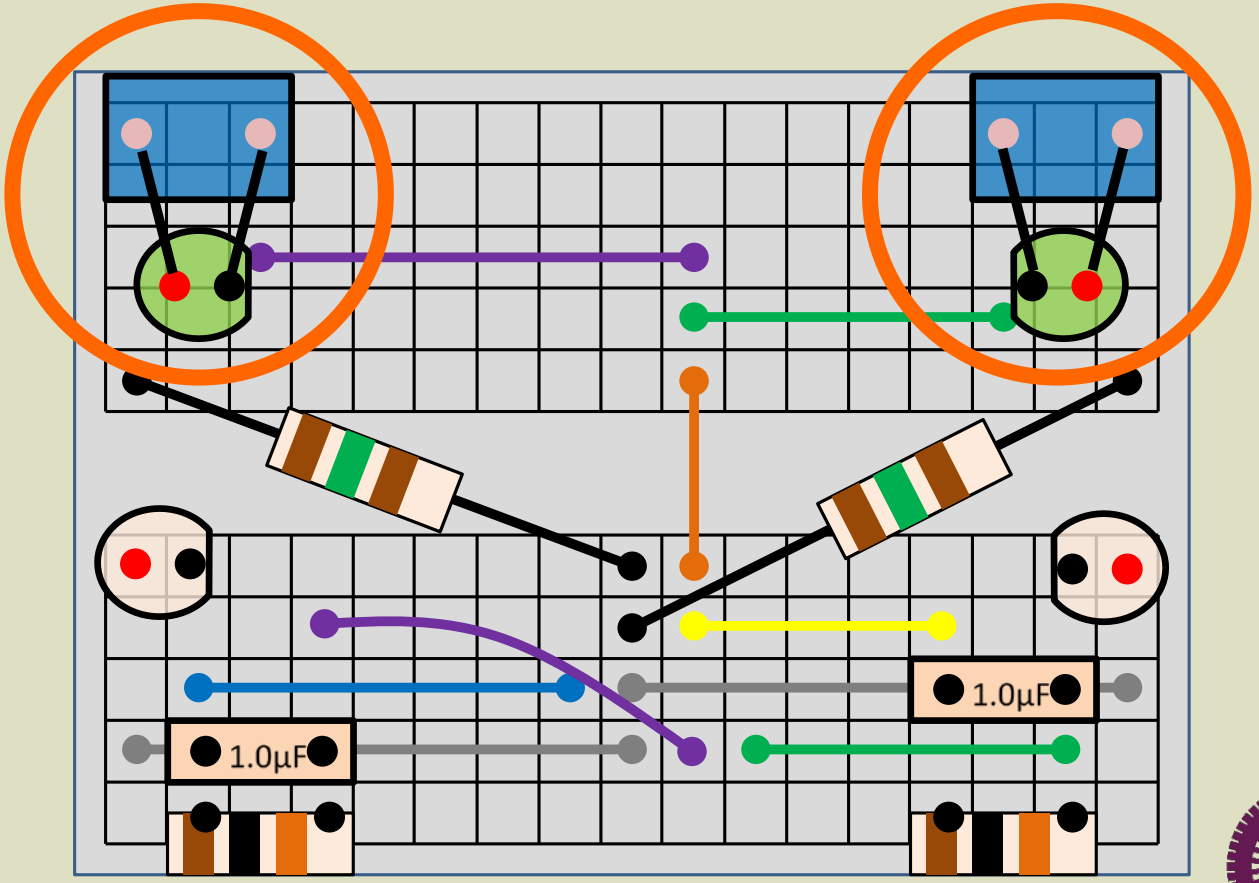


Insert LEDs into Terminals

Insert the two LEDs in opposite directions in their terminals



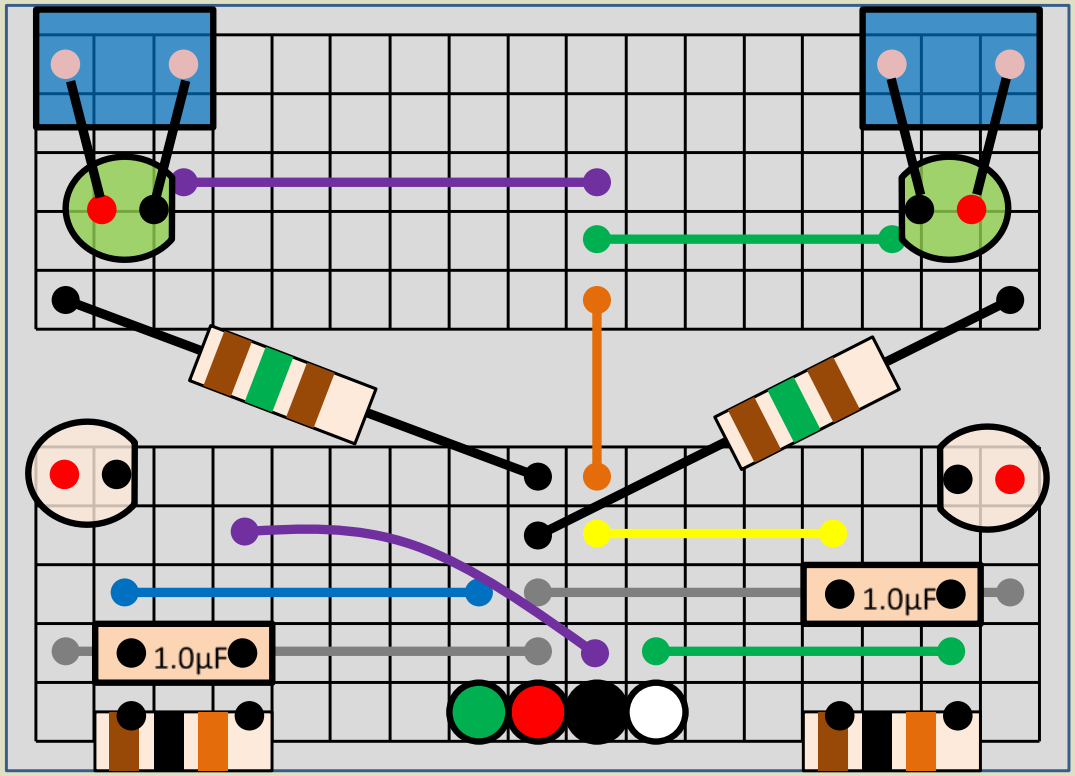
Insert Terminals



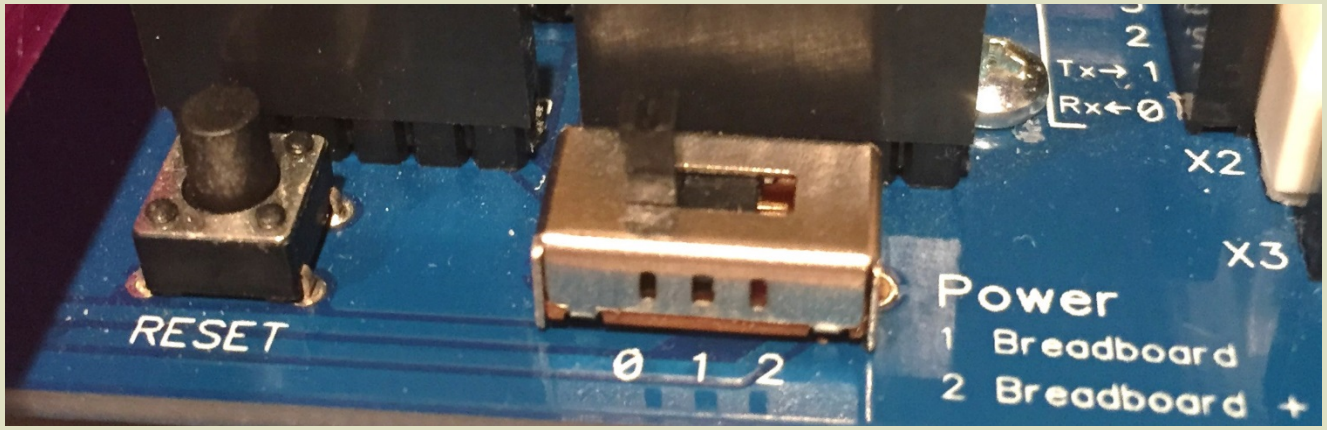
Remove Glasses

Connections

(Mark Power and Ground)

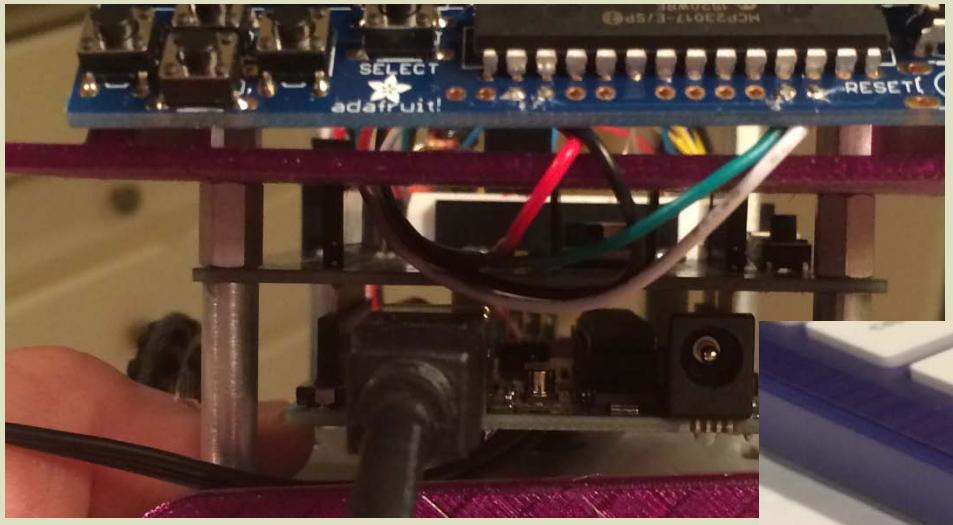


Power Off



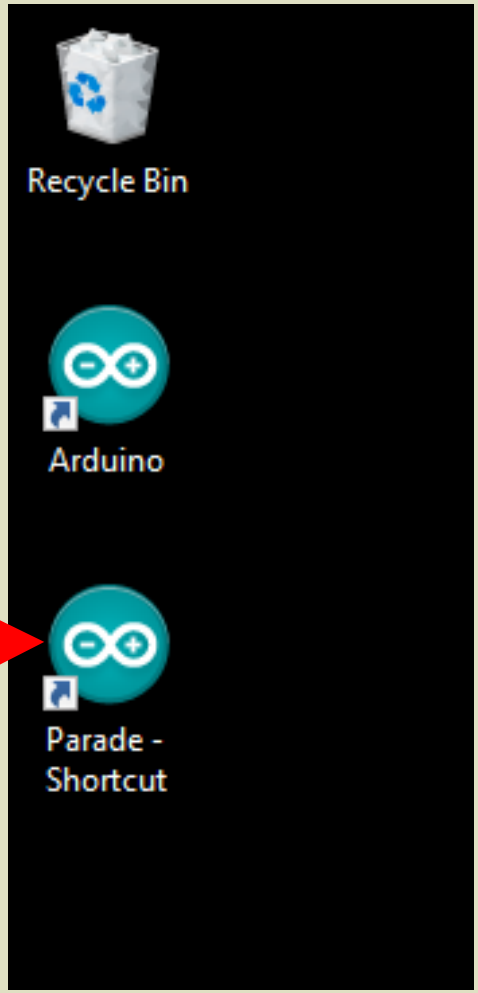
gizmo garden©

Connect Gizmo to Computer



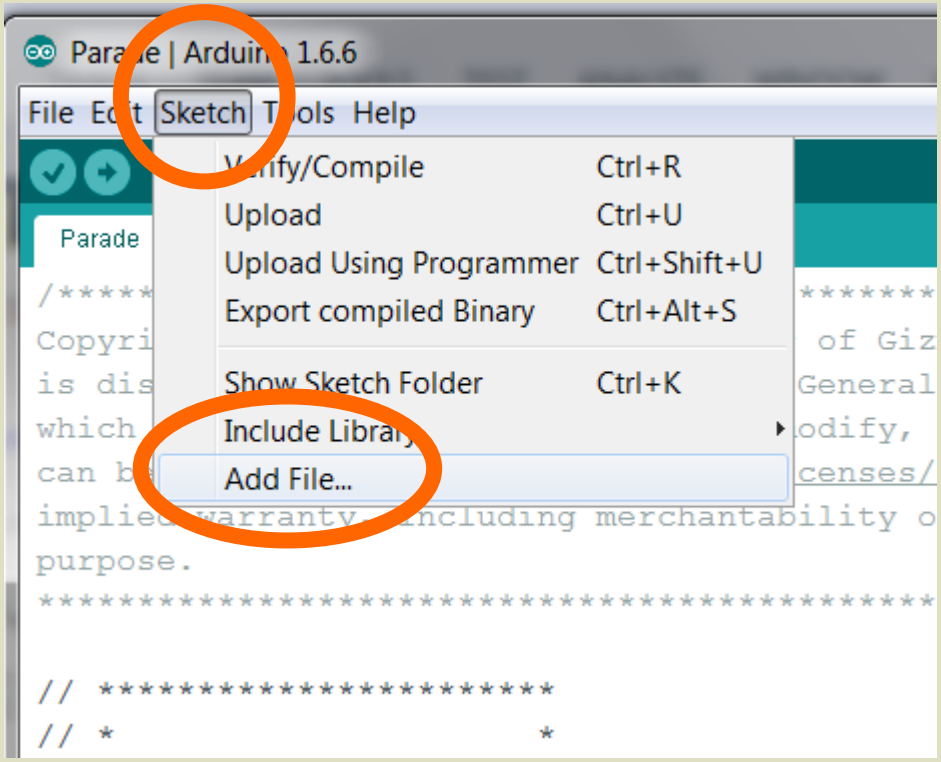
gizmo garden©

Open Arduino App



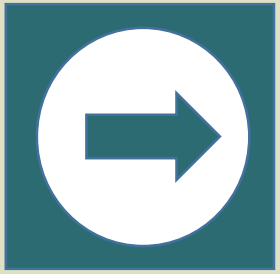
Double-click

Add Sensors from the Code Catalog

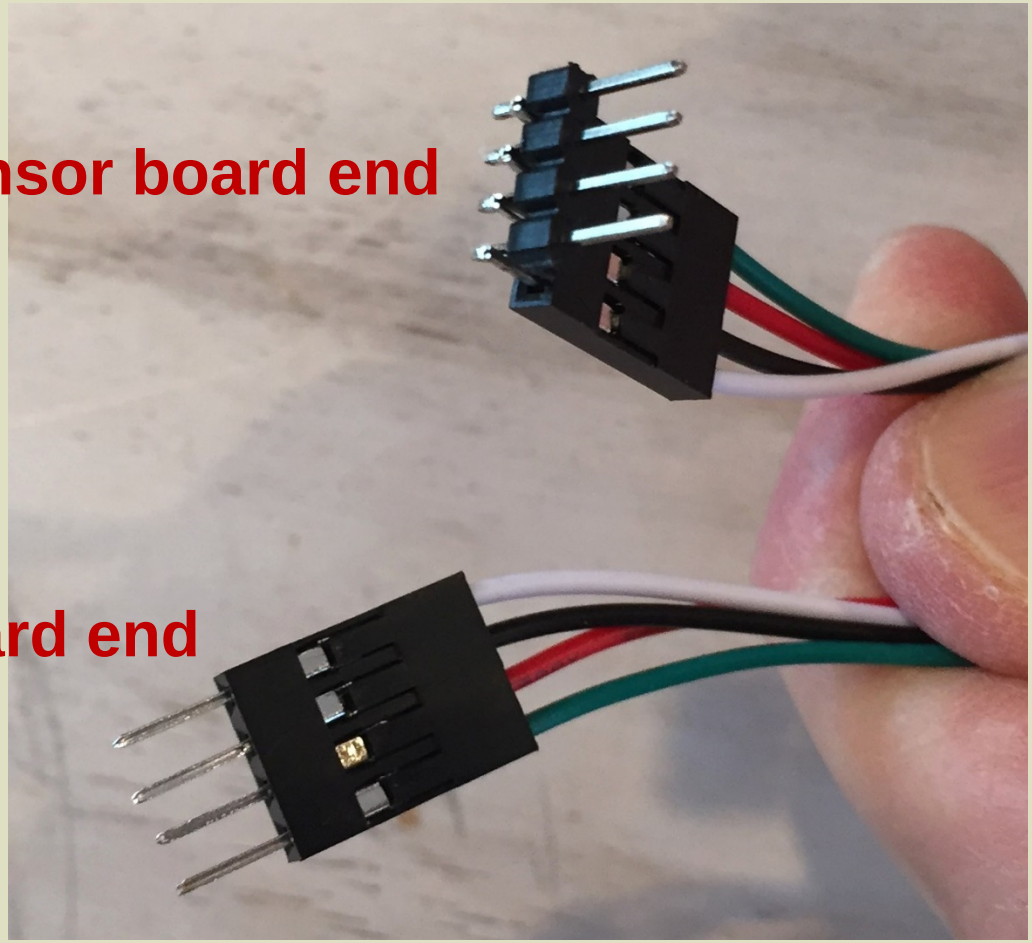


gizmo garden©

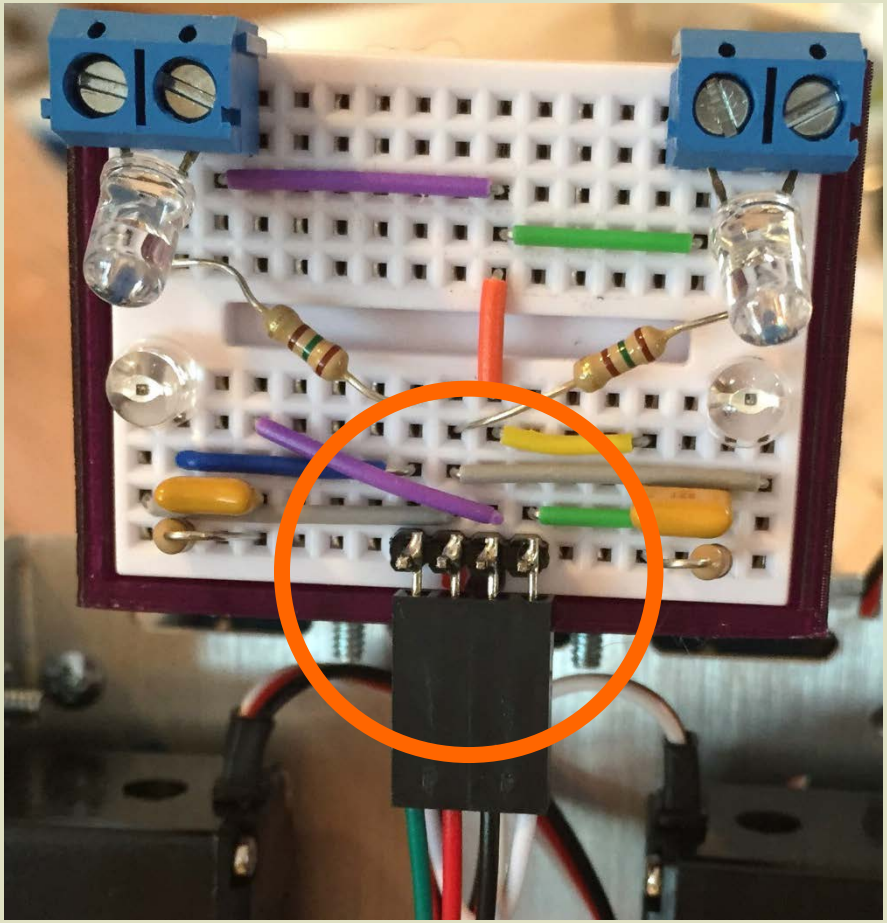
Upload, Keep Power off For Now



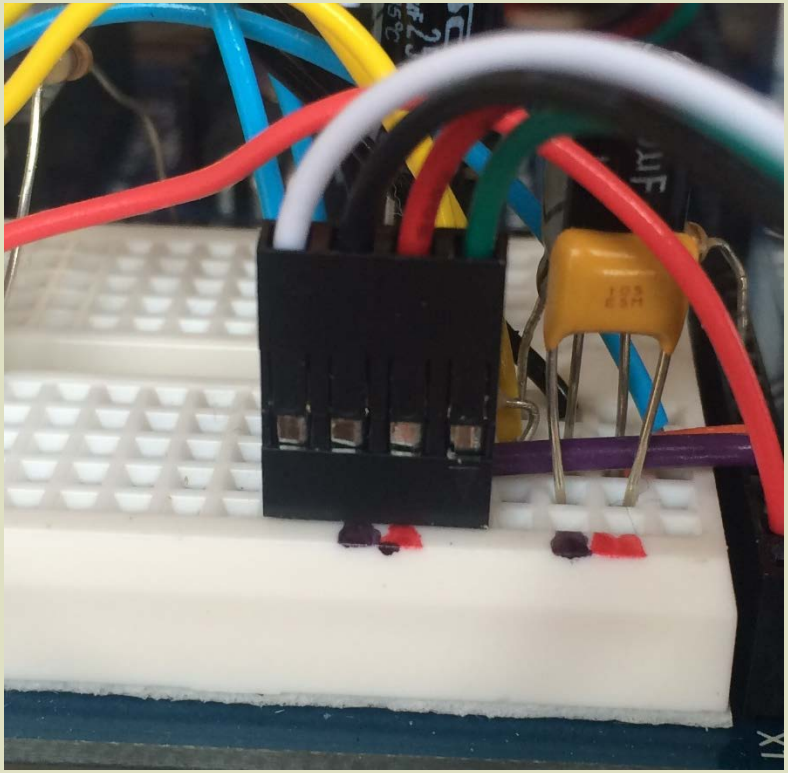
Sensor Cable



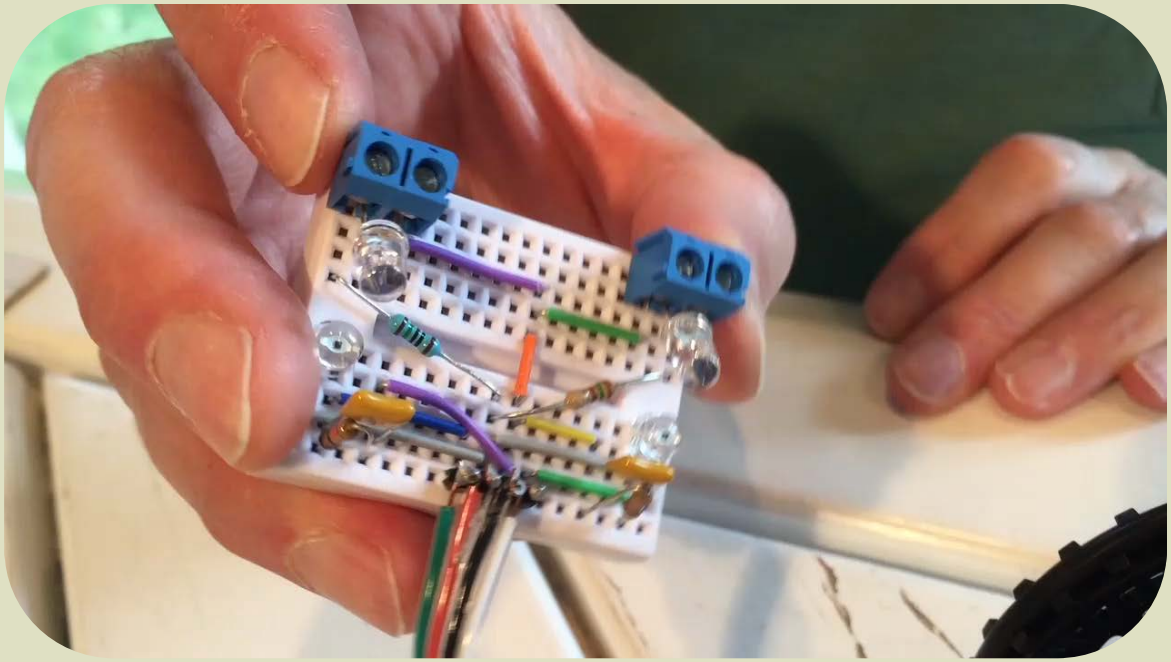
Connect to Sensor Board

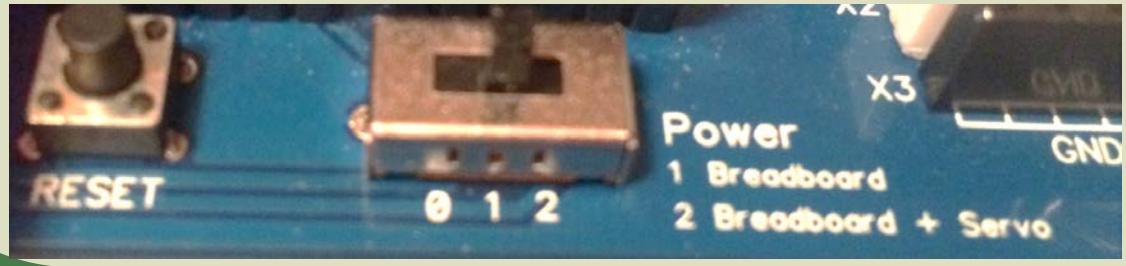


Connect to Main Board



Power to Position 1
Do LEDs Light up?



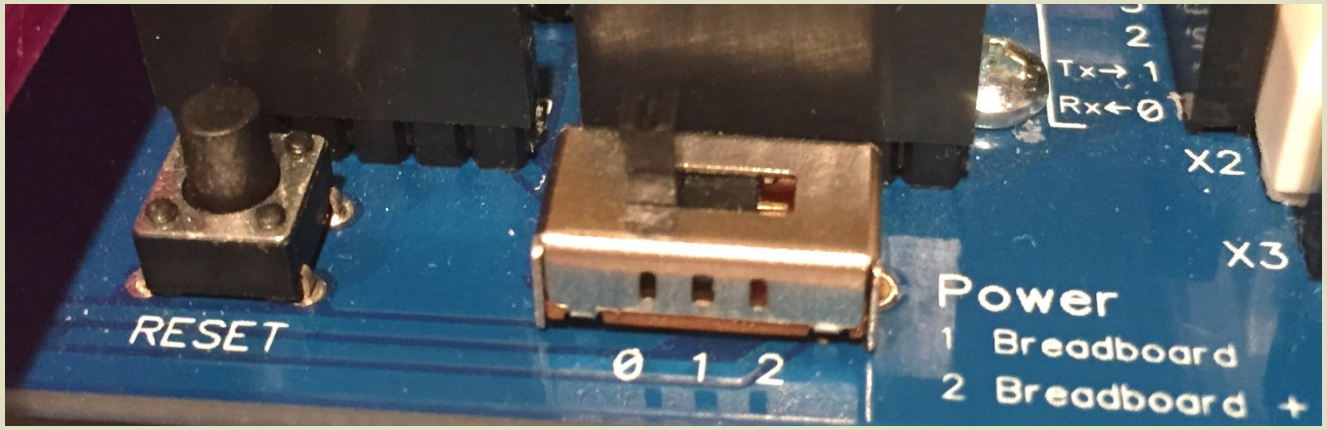


gizmo garden©

Scroll to RoadSensors
Do they respond to light?

May need to reset if dashboard is blank

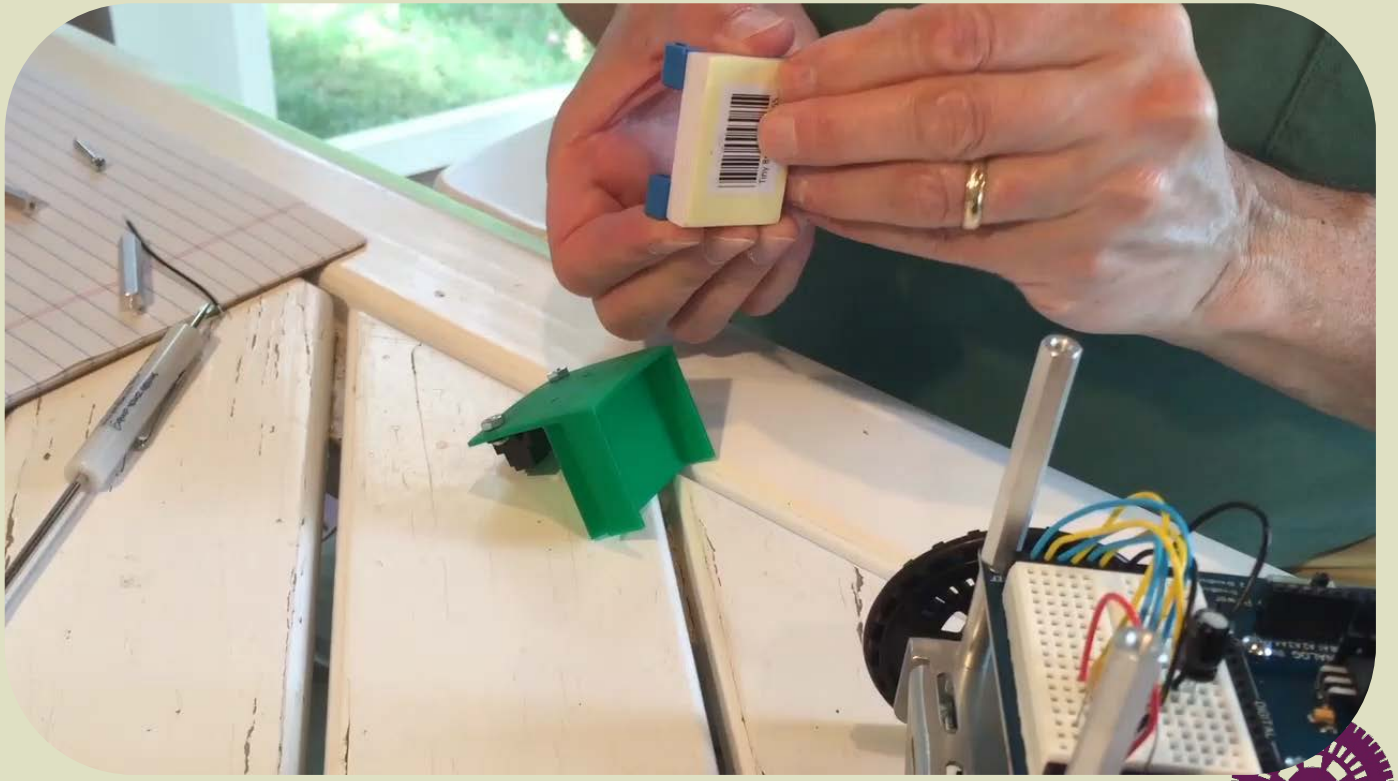
Power Off



gizmo garden©

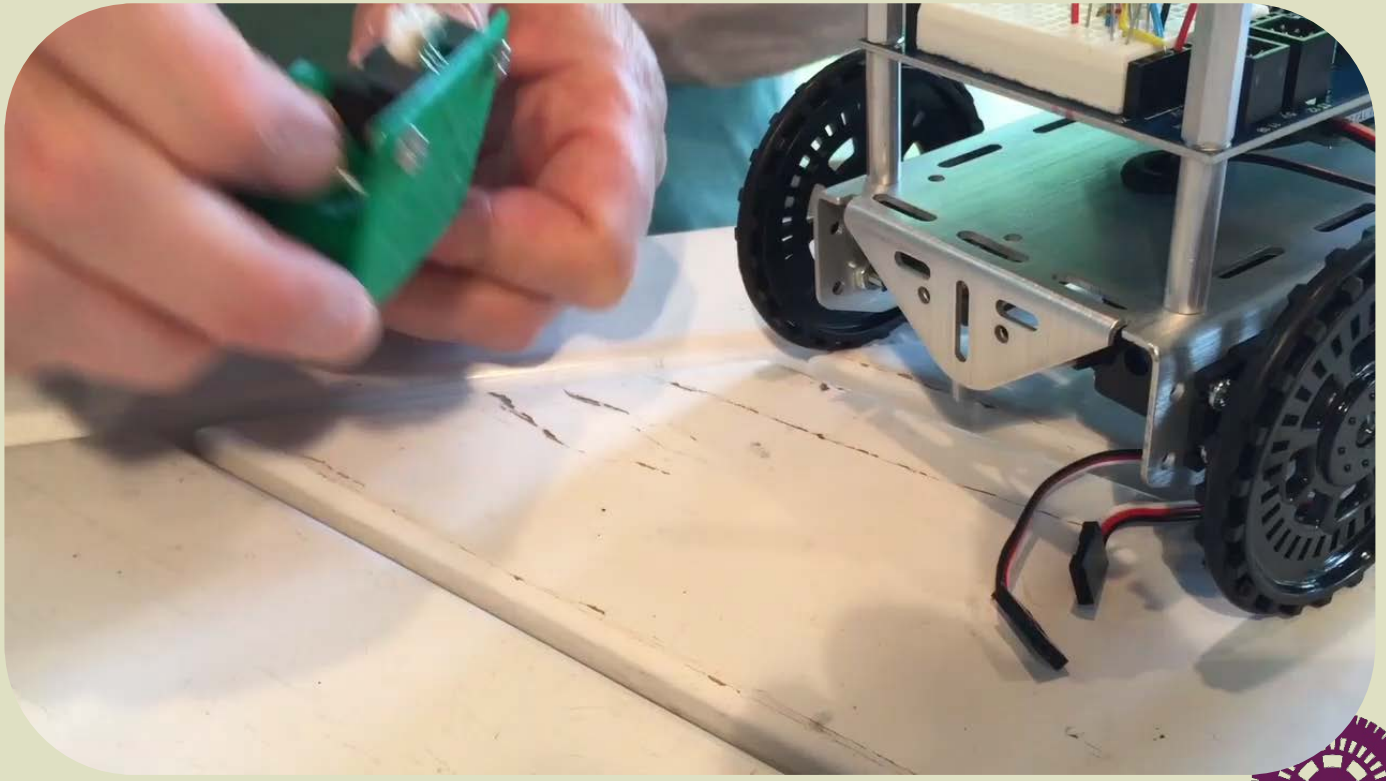
Attach Proximity Sensor to Bracket using two ¼” screws (smallest) and two nuts

Stick Breadboard onto Bracket



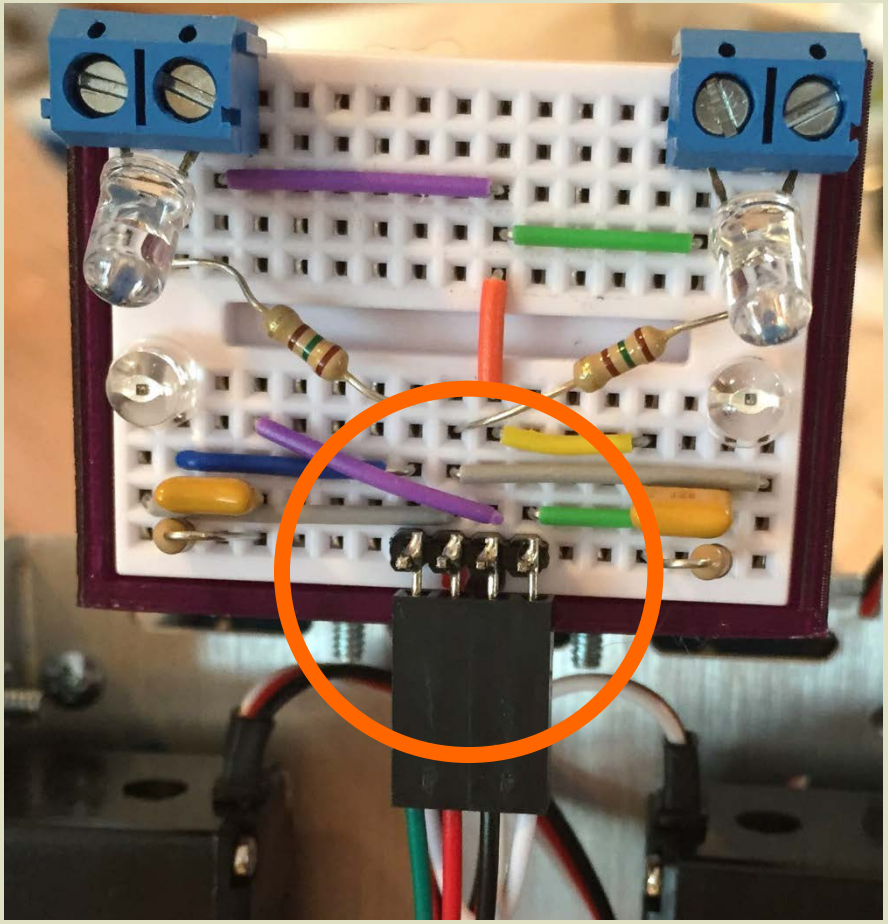
Be careful, once stuck it can’t be moved.

Attach bracket to round holes in chassis front using two 3/8” screws (second smallest) and two nuts

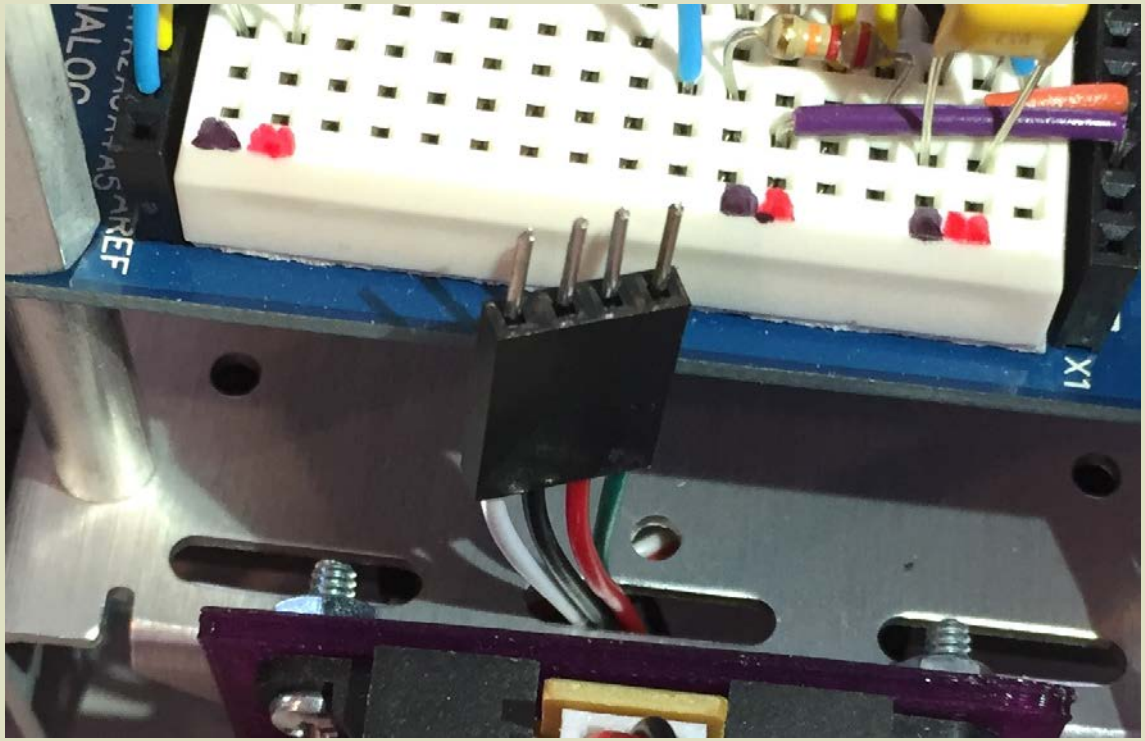


The screws go through the round holes, not the oblong slots

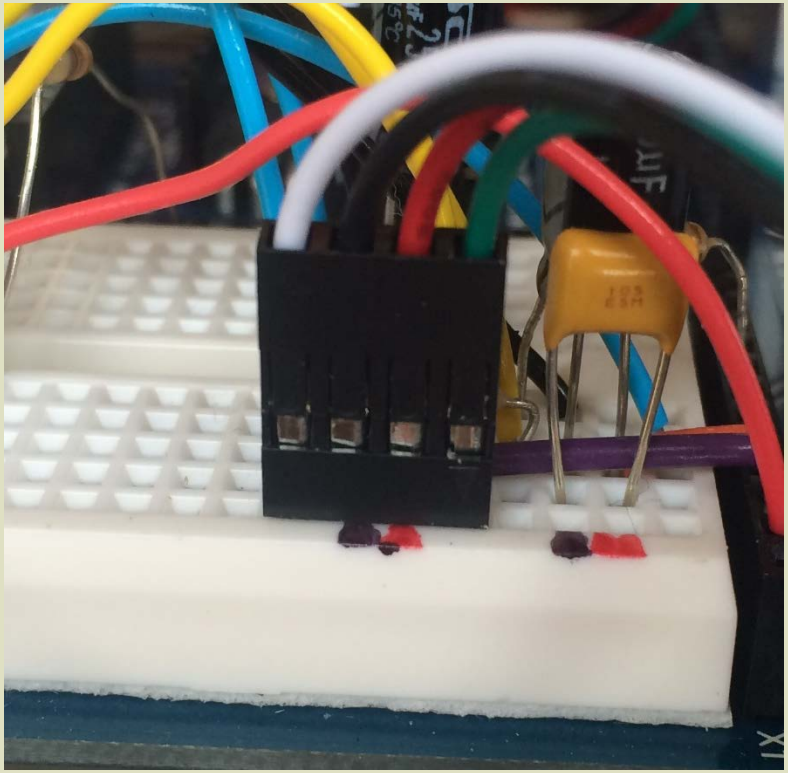
Connect to Sensor Board



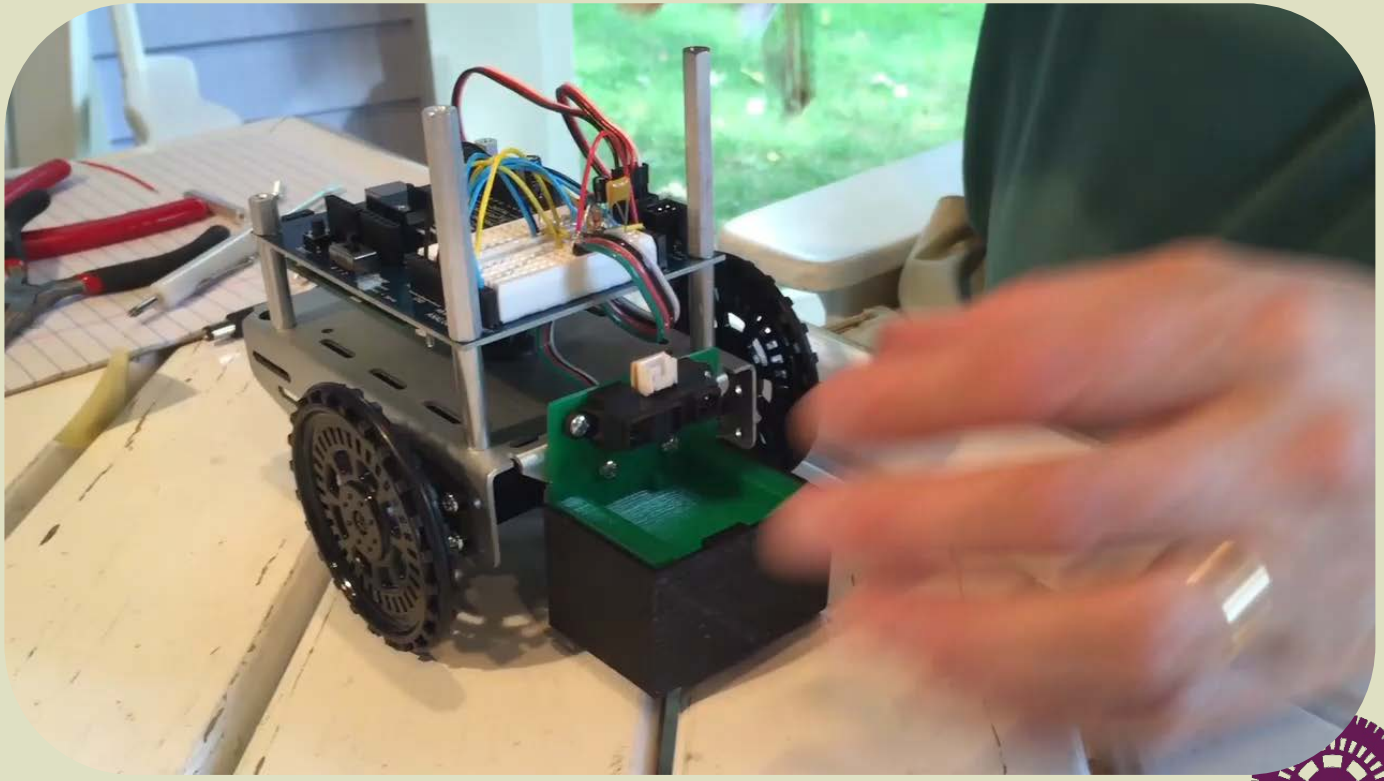
Push Cable Through Center Slot



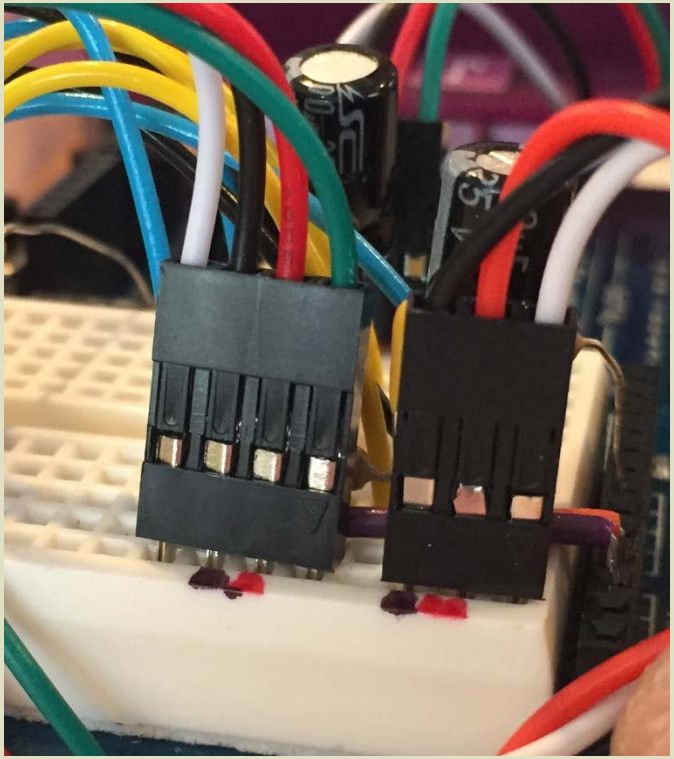
Connect to Main Board



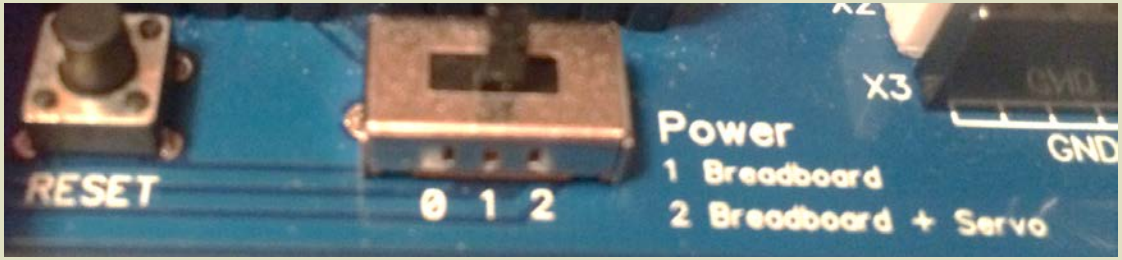
Attach Proximity Sensor Cable



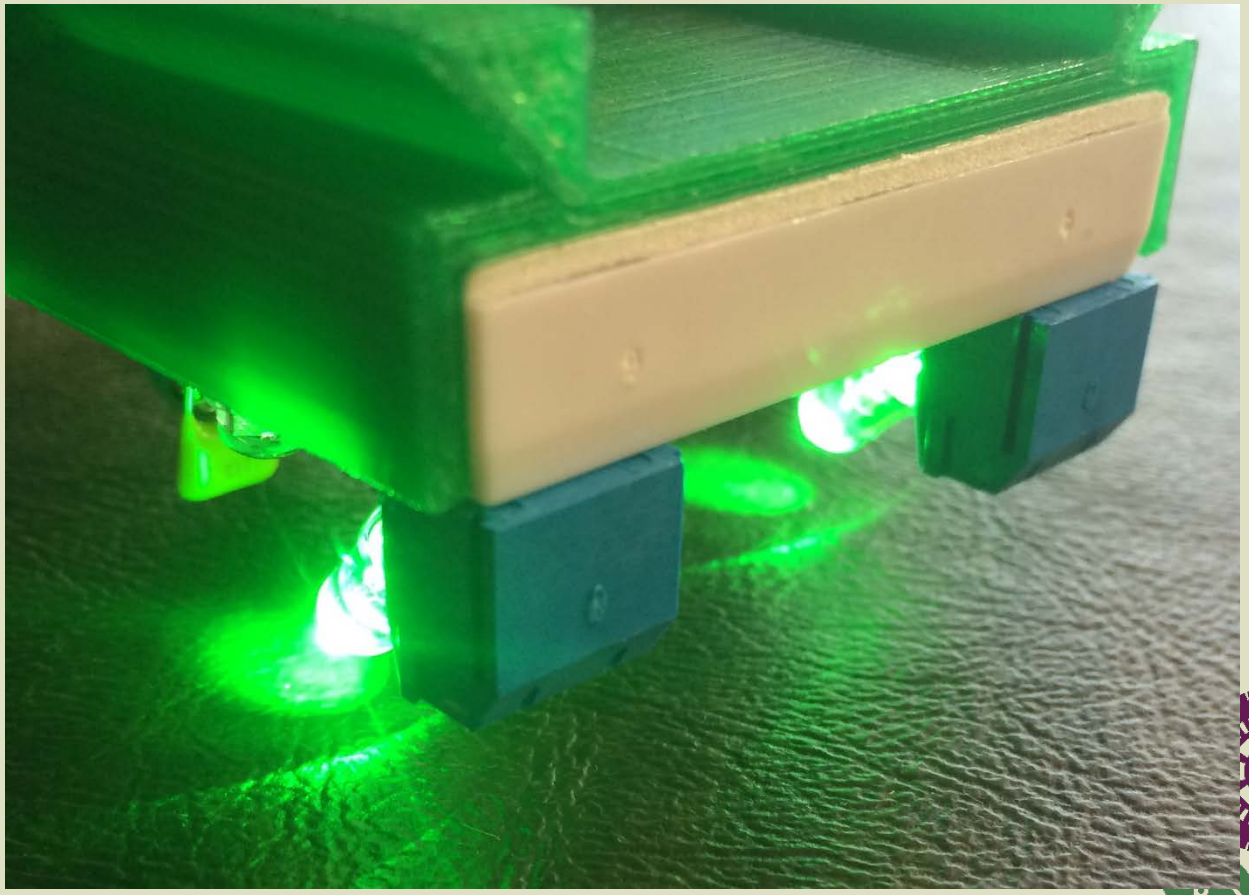
Both Connectors in Place



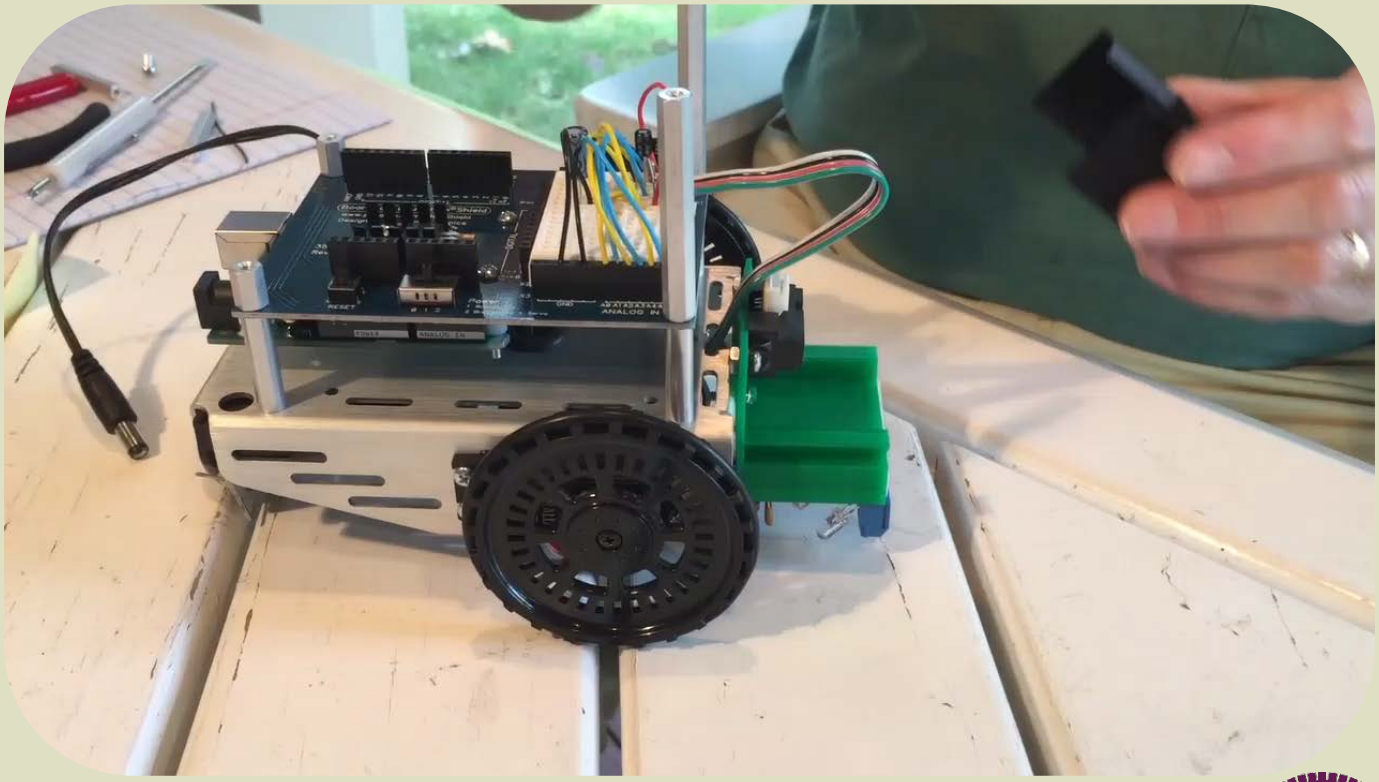
Power to Position 1



Bend LEDs to get light spots below phototransistors



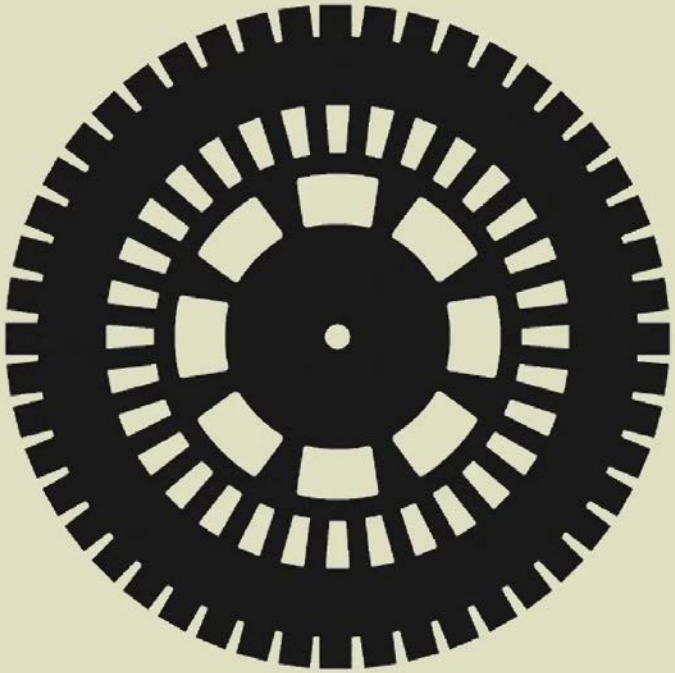
Slide on Shroud



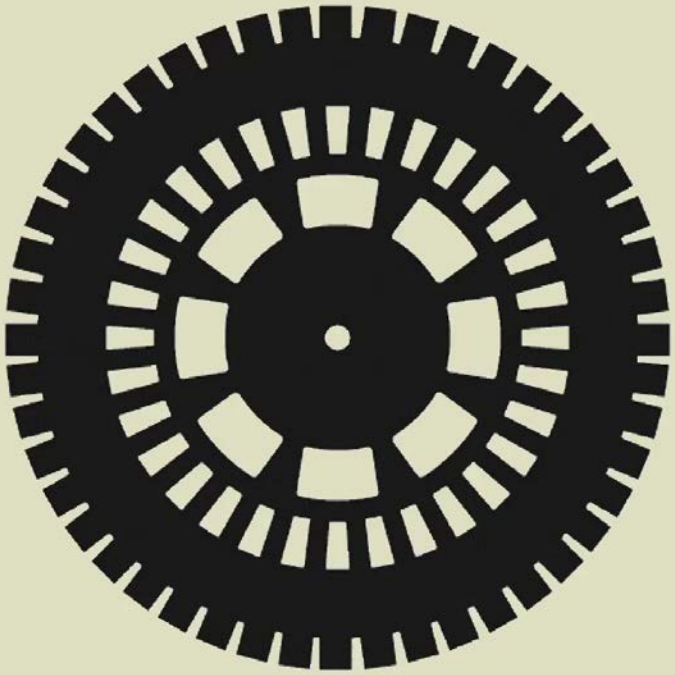
Scroll to Proximity
Does it respond to your hand?

May need to reset if dashboard is blank

I Can See!

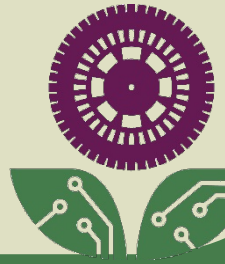


Learning to Drive
Part 1: Speed Control

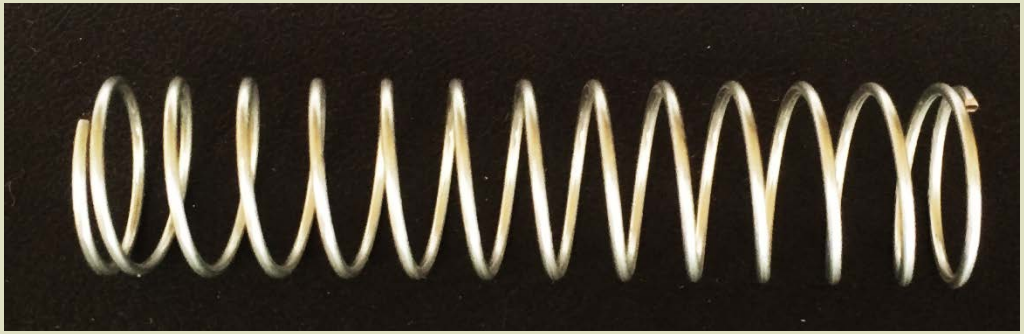


Rules of the Road

- All clear, go fast
- Getting closer, slow down
- Close enough, stop
- Too close, back up



Use an Imaginary Spring!



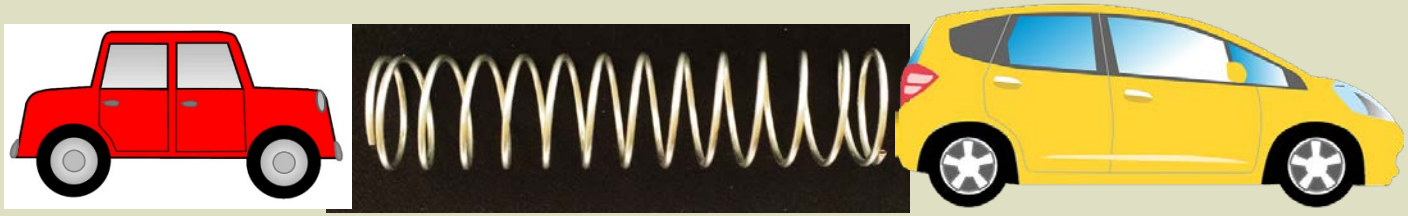
All Clear

Cars are far apart. Spring is stretched and pulling hard. Trailing car goes fast.

Getting Closer, Slow Down

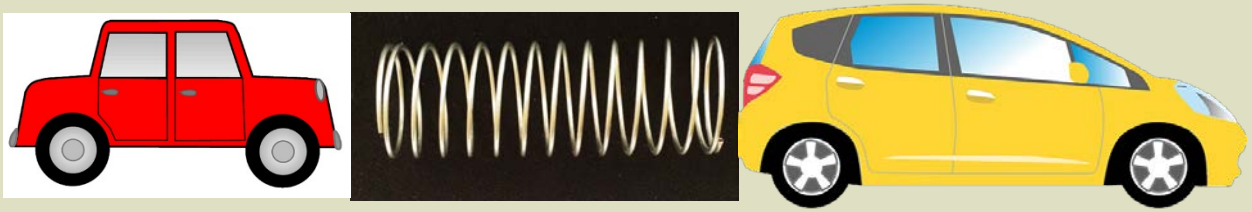
Cars are closer. Spring is still stretched but not as much. It's pulling a little, trailing car goes slower.

Close Enough



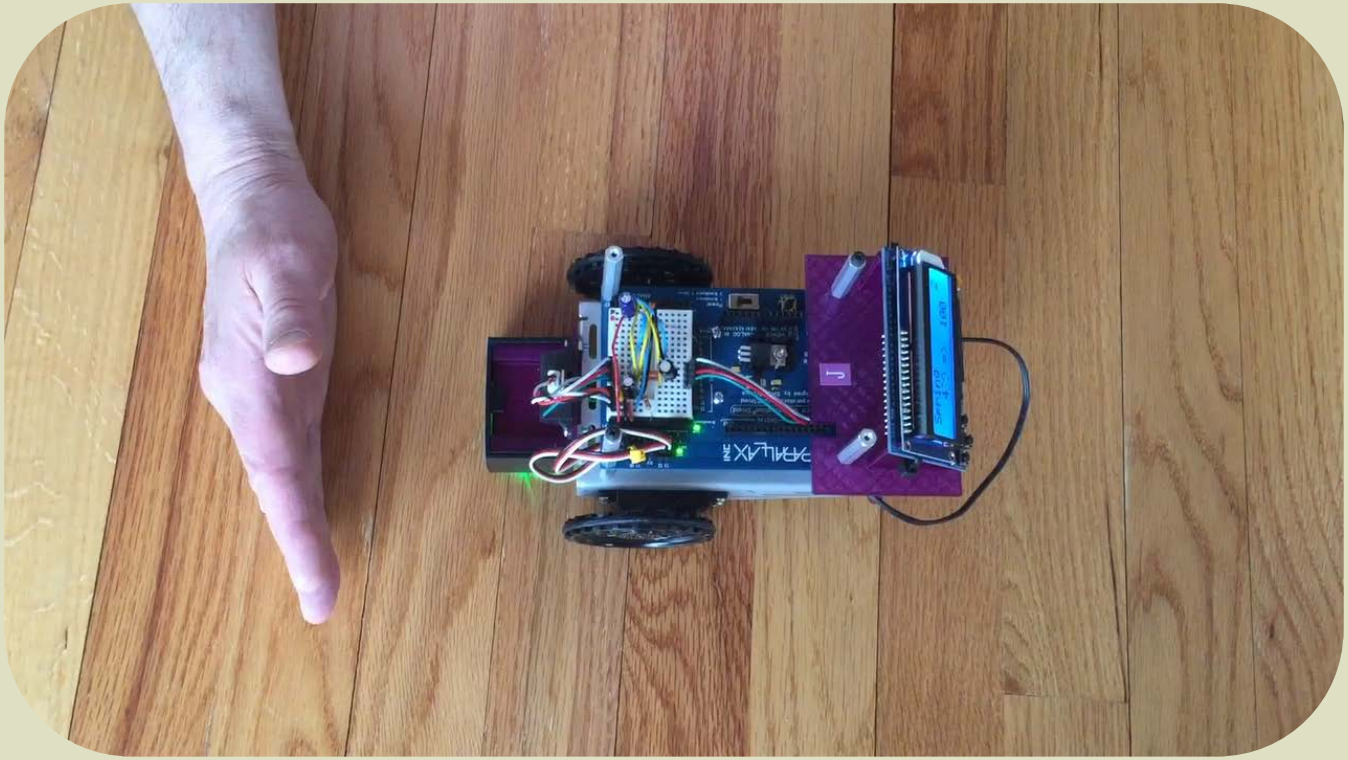
Spring is at its natural length, no pushing or pulling on trailing car so it is stopped

Too Close, Back Up

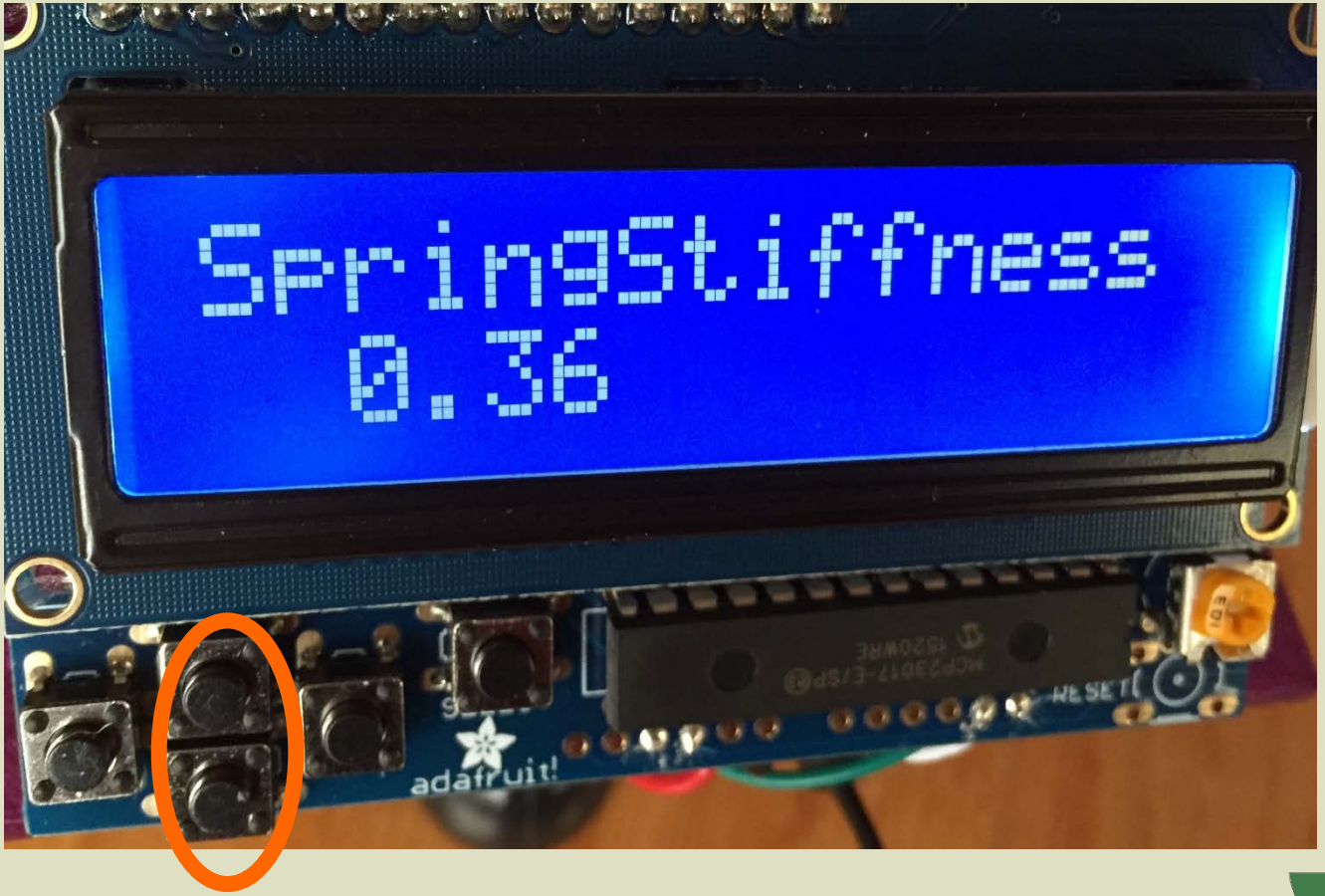


Spring is compressed, pushing on trailing car so it backs up.

Let's See It

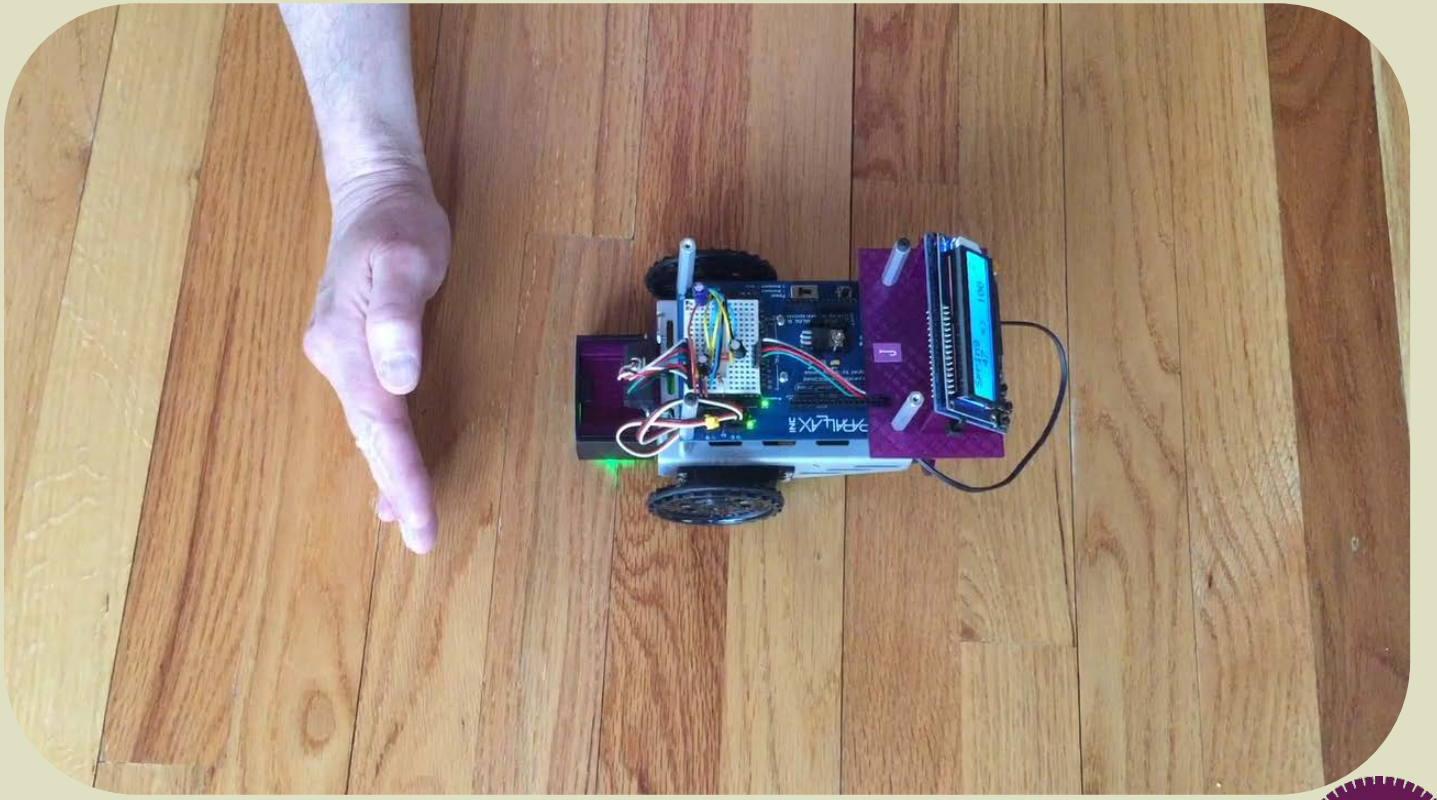


We can change the stiffness of the spring

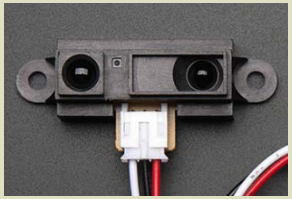


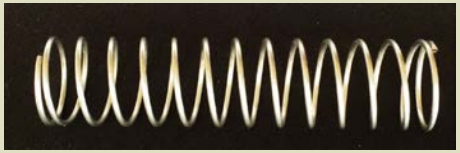
Let's try a stiffer spring

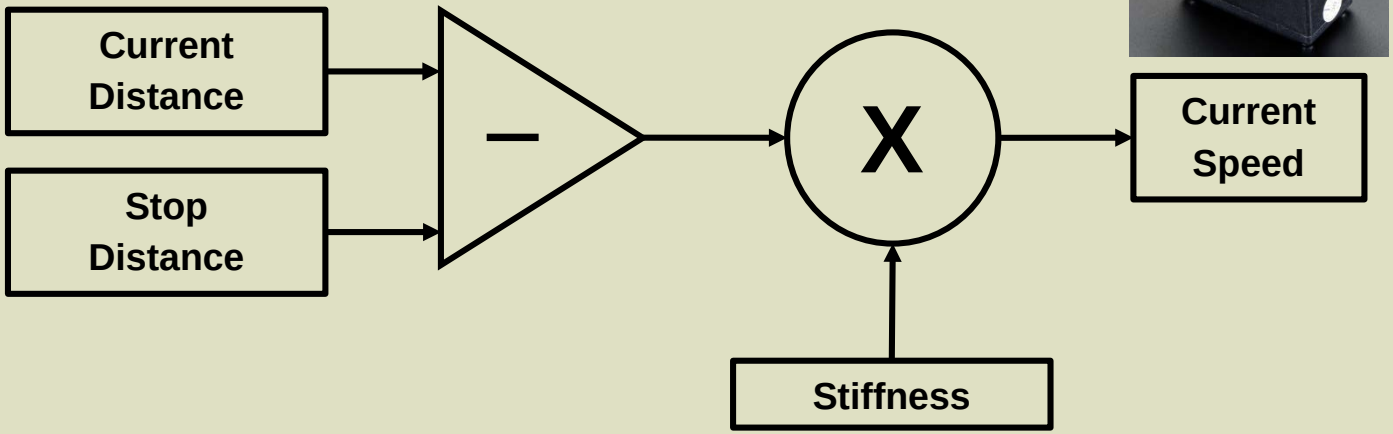
Stiff Spring



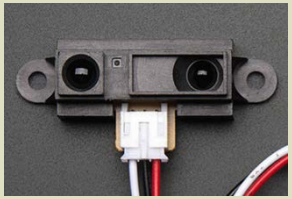
A Software Spring

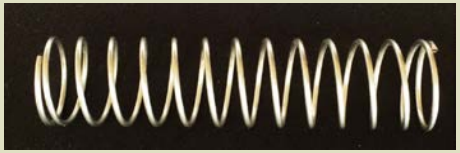


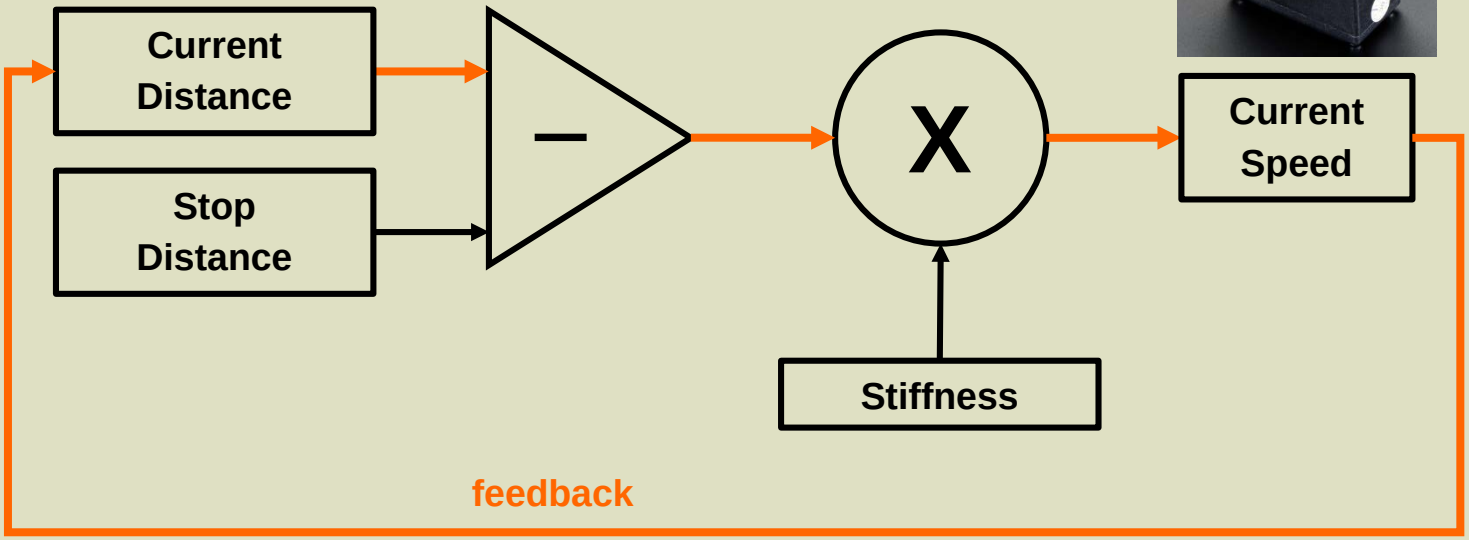




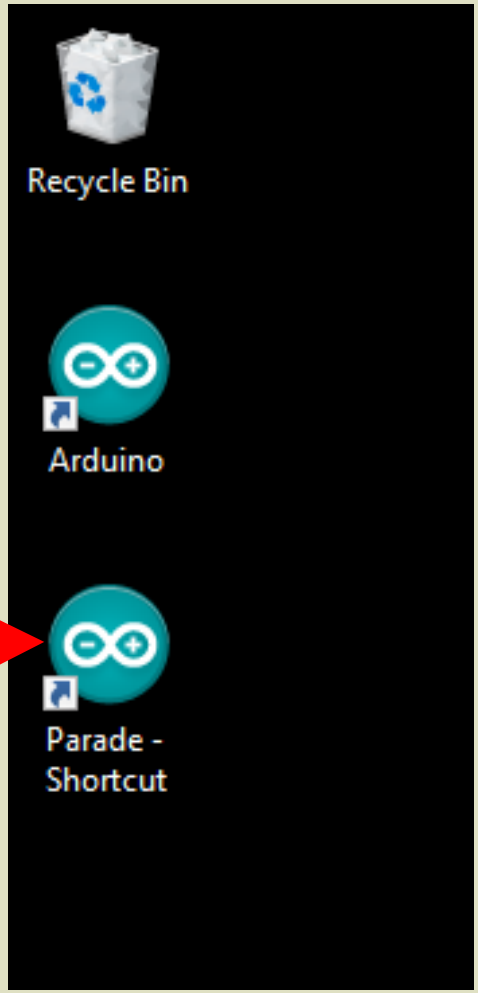
In the Loop





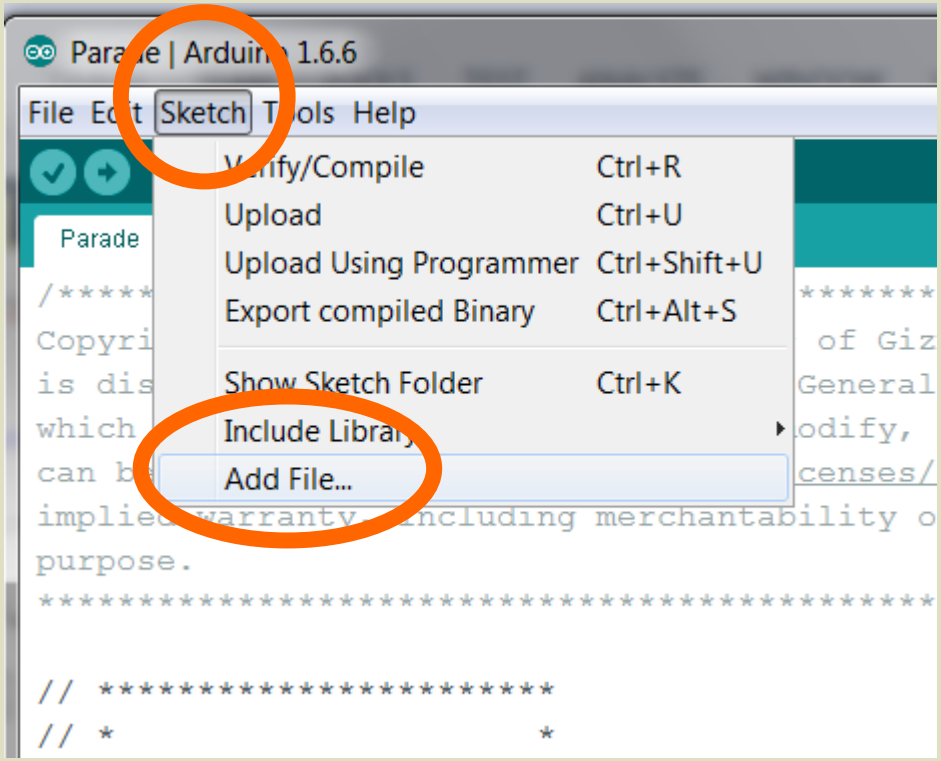


Open Arduino App



Double-click

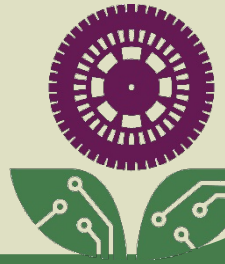
Add Spring from the Code Catalog



gizmo garden©

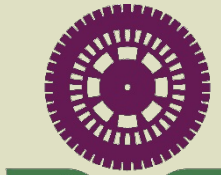
Find the **gasPedal** function

```
//  ****************
//  *                *
//  *   Gas Pedal    *
//  *                *
//  ****************

void gasPedal()
{
}
```



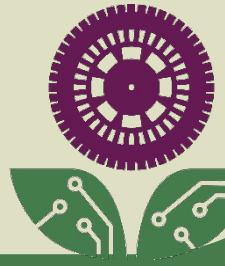
Get the Current Distance

```
void gasPedal()
{
   float distance = readProximitySensor();
}
```



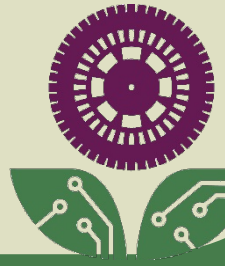
Compute Speed

```
void gasPedal()
{
  float distance = readProximitySensor();
  float driveSpeed = (stopDistance - distance) * springStiffness;
}
```

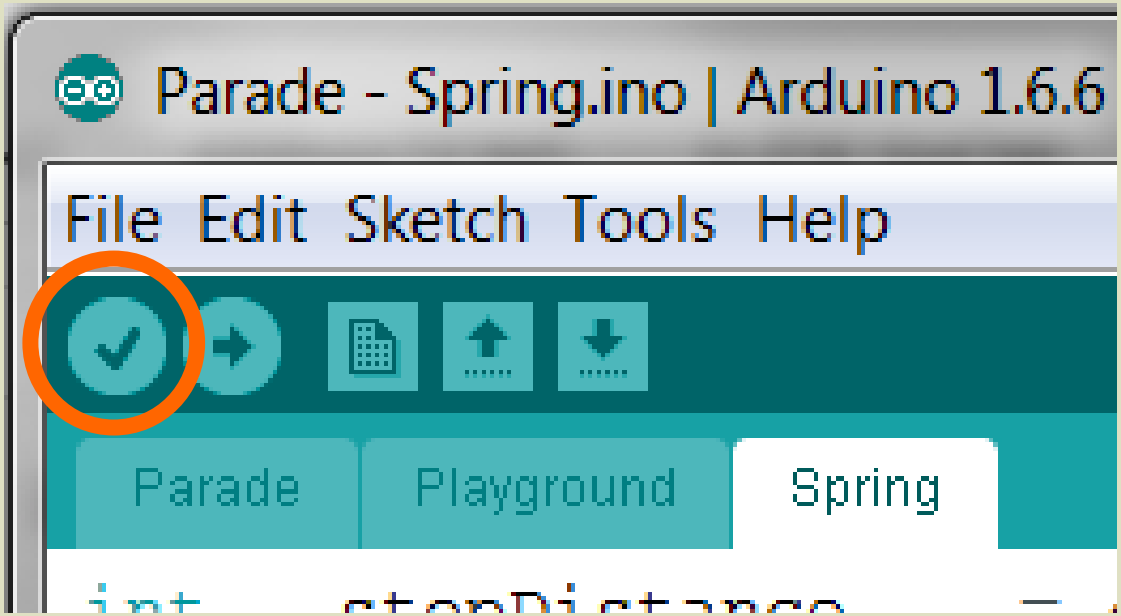


Tell Motors to Go

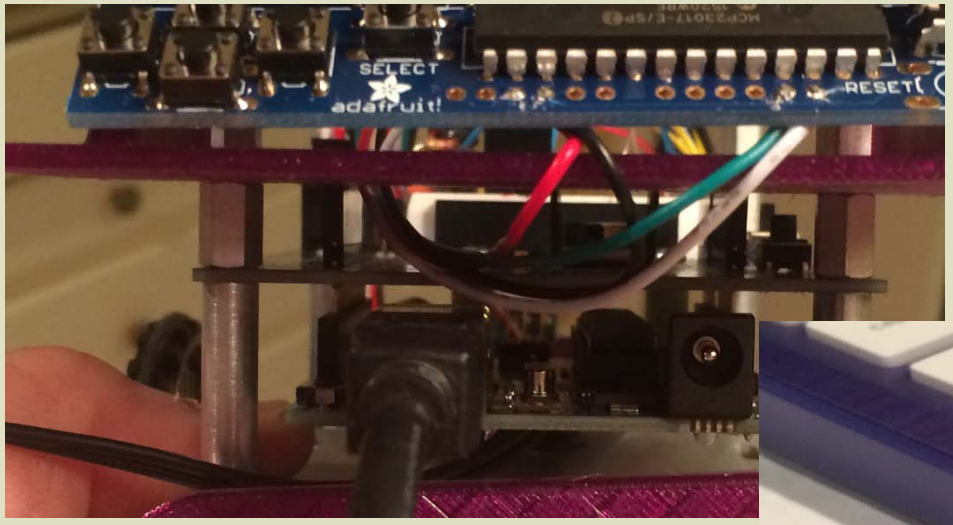
```
void gasPedal()
{
  float distance = readProximitySensor();
  float driveSpeed = (stopDistance - distance) * springStiffness;
  driver.setFullSpeed(driveSpeed);
}
```



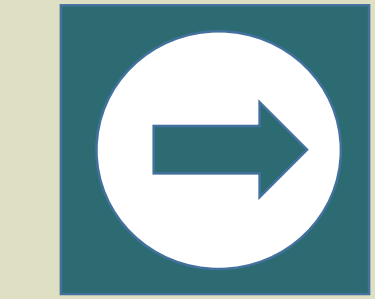
Verify



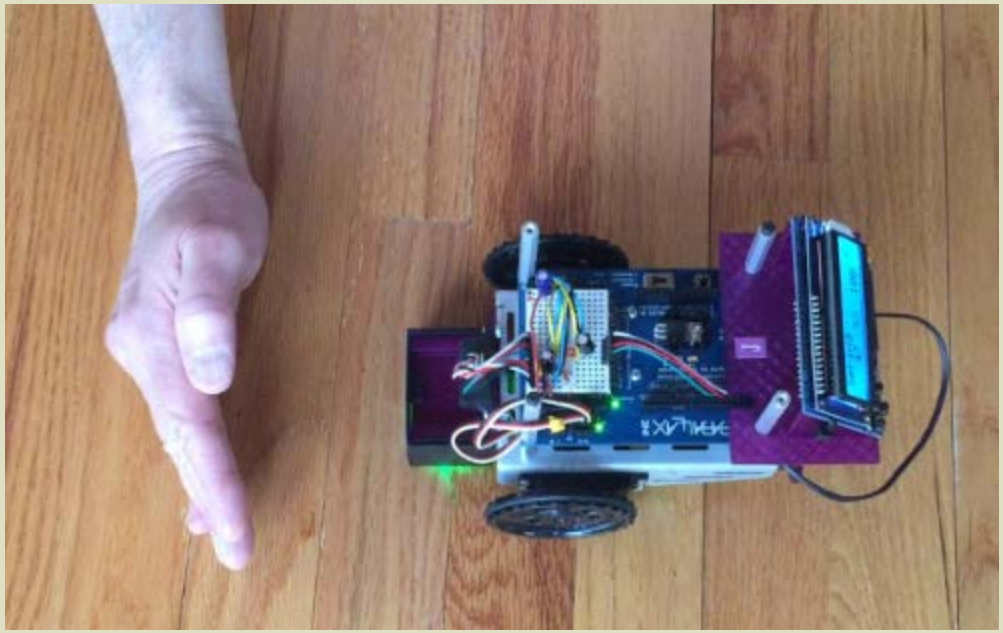
Connect Gizmo to Computer



Upload, Power #2, Scroll Left-Right to Spring

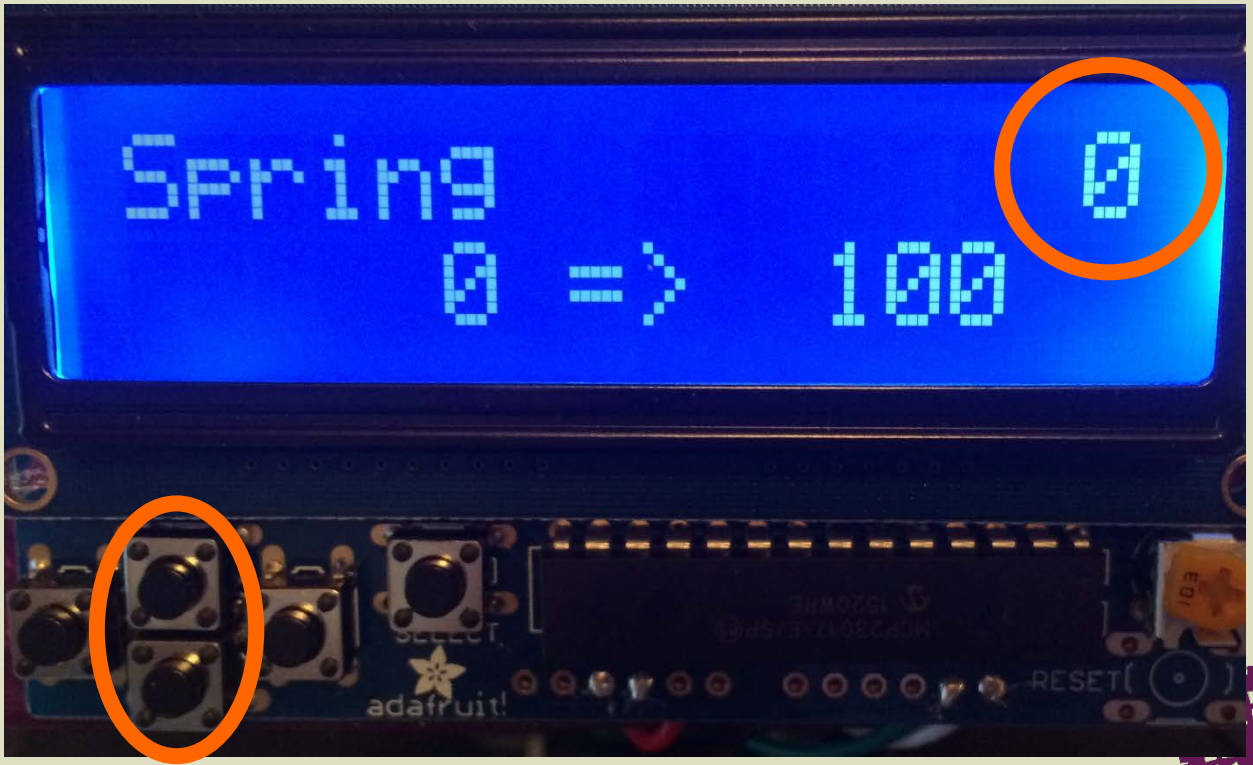


Be in Control



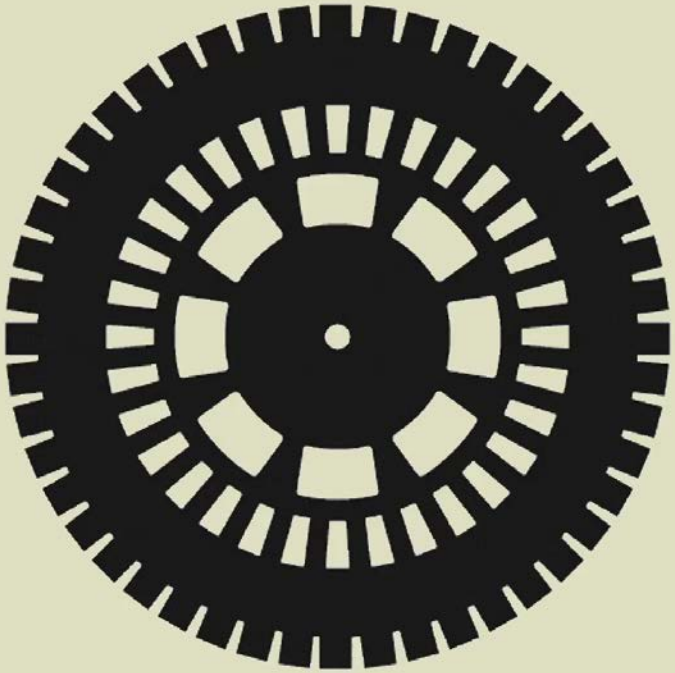
Select to Start/Stop

Up/Down to Balance

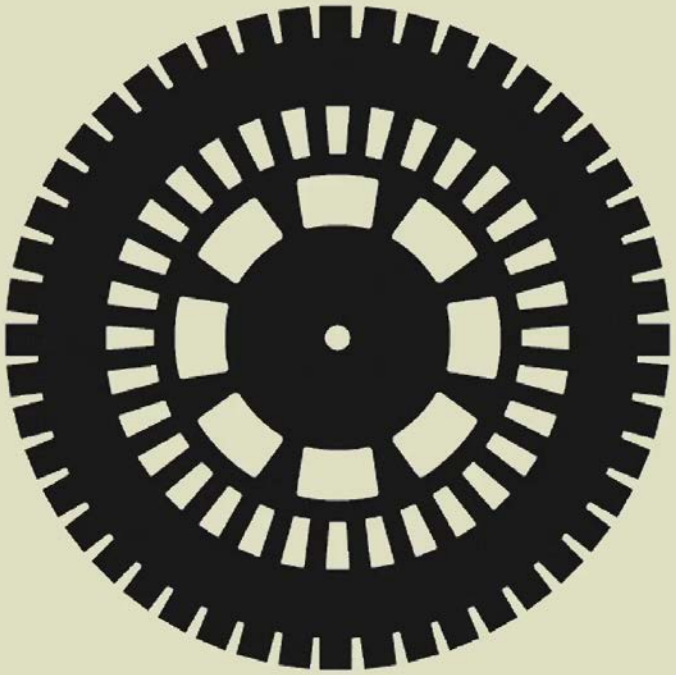


Play Around

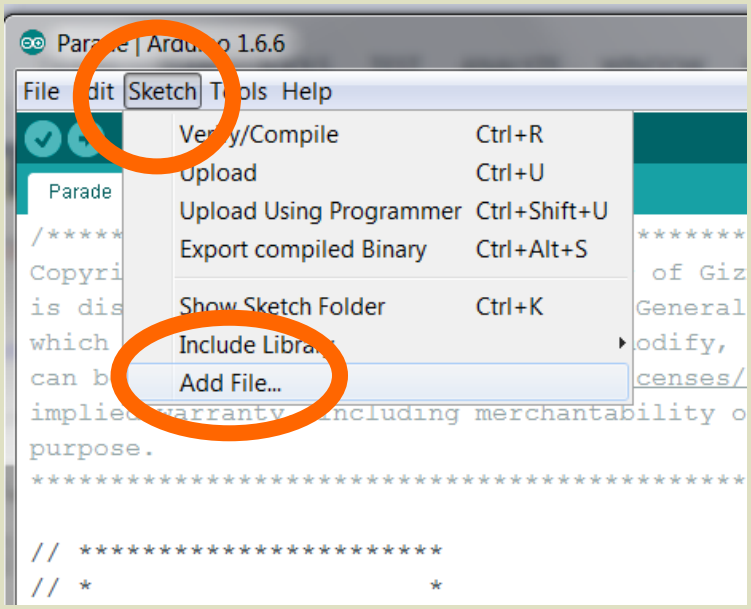
In Control!

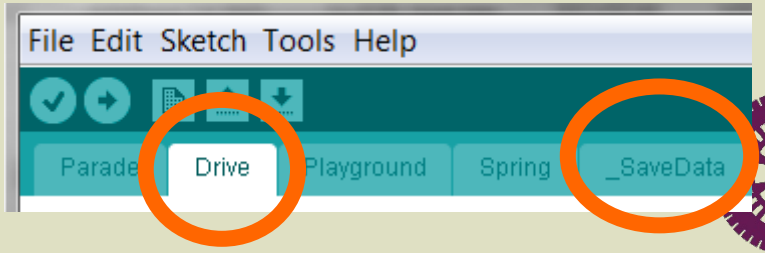


Learning to Drive
Part 2: Steering



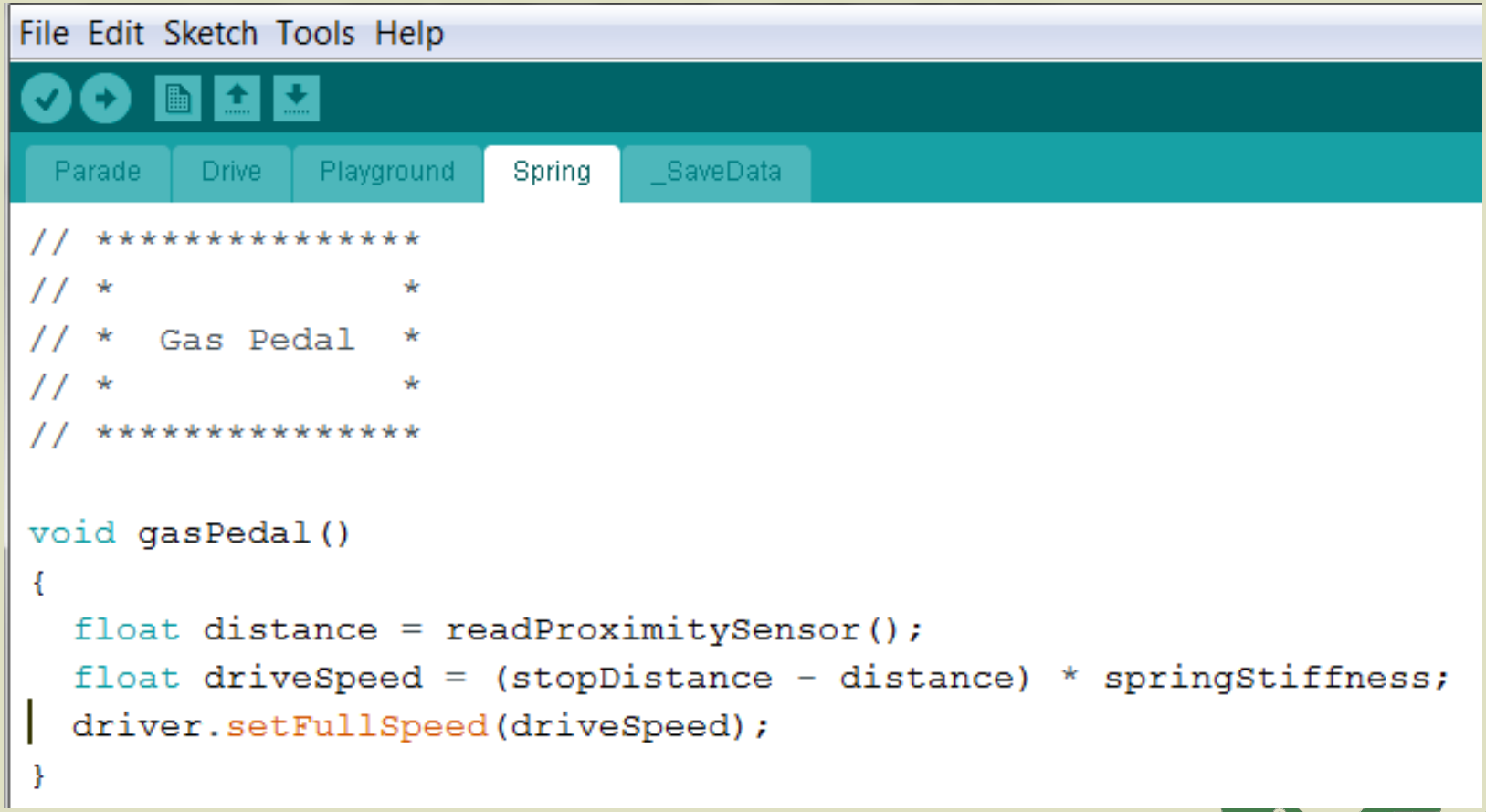
Add Drive and SaveData from the Code Catalog





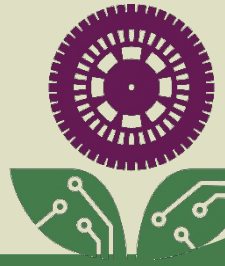
gizmo garden©

Find **gasPedal** in the Spring tab

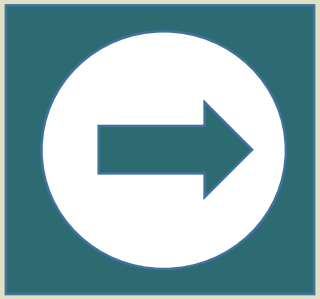


The parade leader goes a little slower

```
void gasPedal()
{
  float distance = readProximitySensor();
  float driveSpeed = (stopDistance - distance) * springStiffness;
  if (leadMode && driveSpeed > 80)
    driveSpeed = 80;
  driver.setFullSpeed(driveSpeed);
}
```

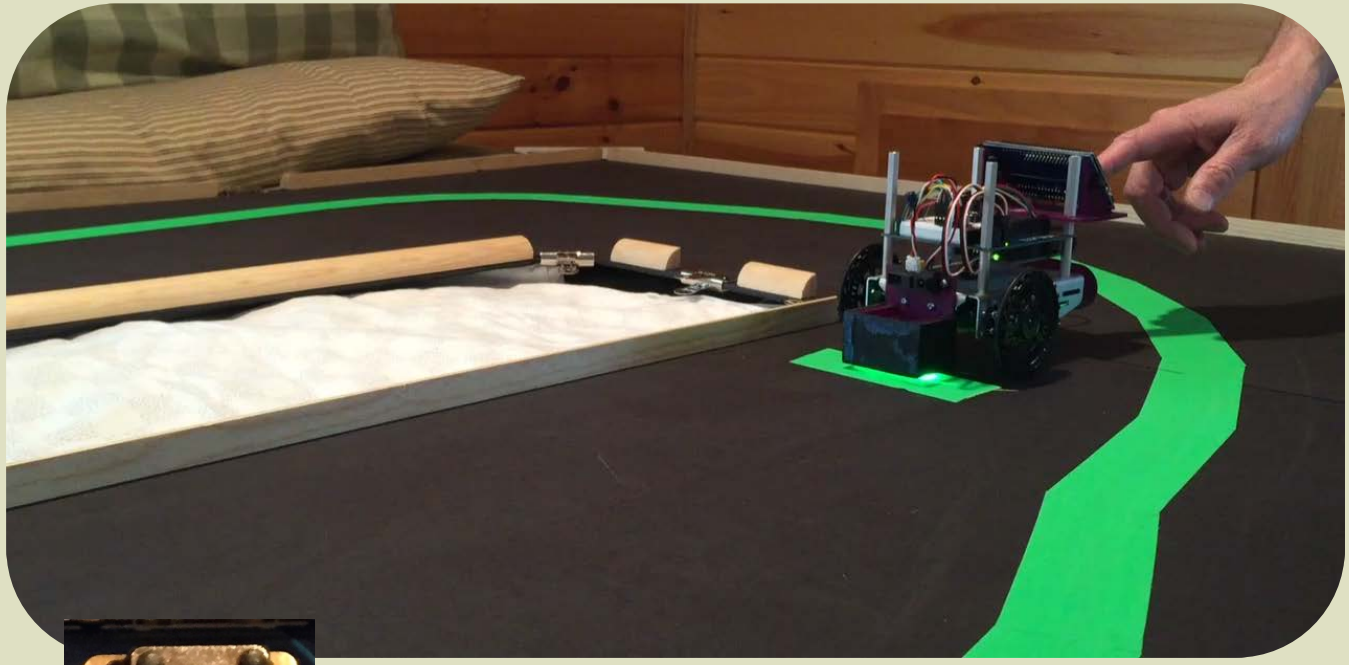


Upload, Power #2, Scroll Left-Right to Calibrate



gizmo garden©

Calibrate Road Sensors



Select to start

Results OK?

- These numbers should be at least 300
- If not, check the aim of the green LEDs

273

Save the Calibration

Scroll here

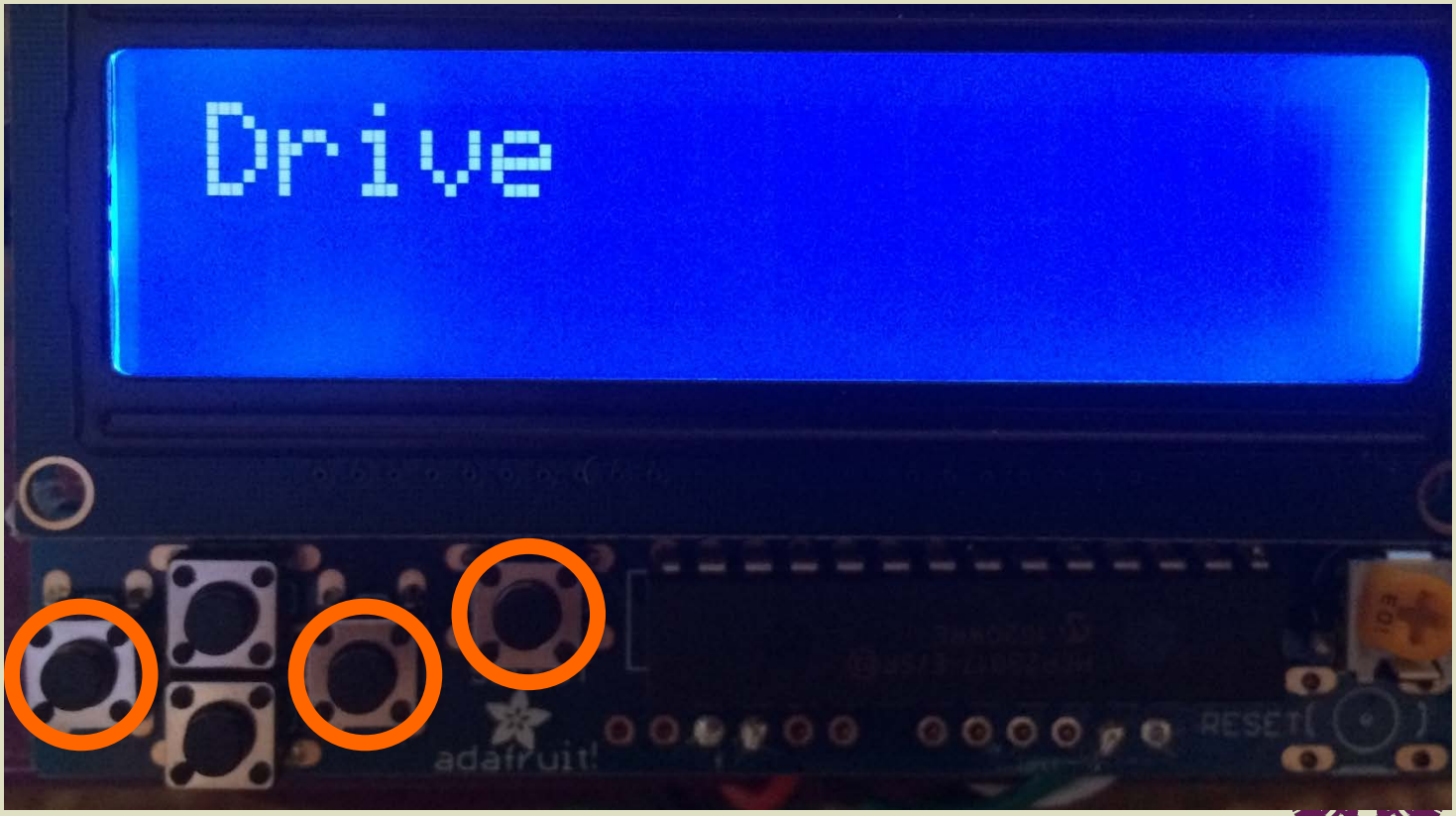
Click Select
to save

Here is what is saved

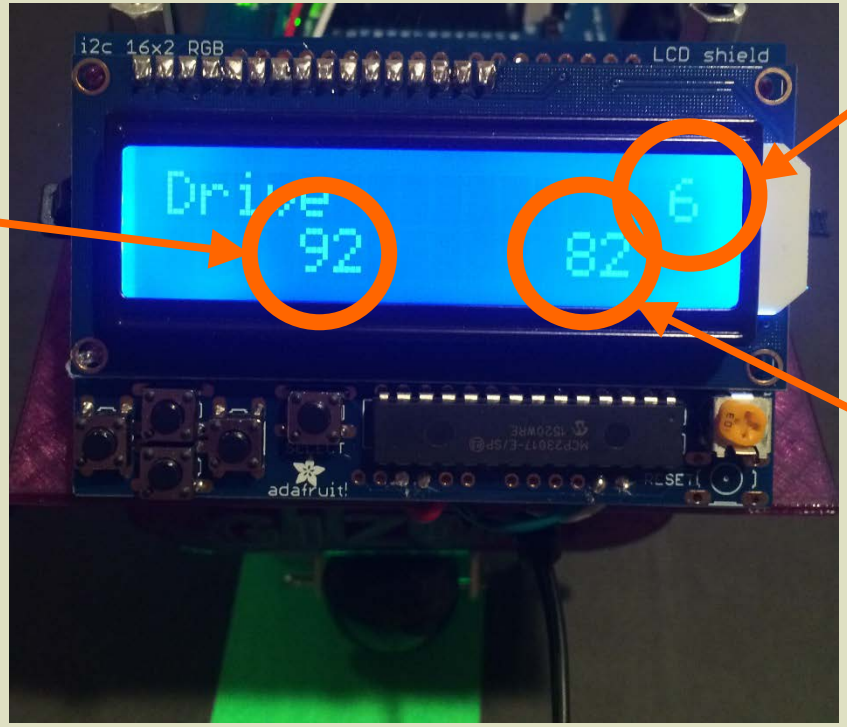
- Calibration data
- Spring stiffness
- Stop distance
- Drive stiffness
- Leader/follower setting
- Left-right balance (Spring menu)
- Beeper volume

275

Scroll to Drive
Select to Start/Stop



Dashboard While Driving



Left sensor
0 – 100
0: off road
100: on road
in between for
partially on road

Speed
0: stopped
> 0: going forward
< 0: backing up

Right sensor
0 – 100
0: off road
100: on road
in between for
partially on road

gizmo garden©

Play Around

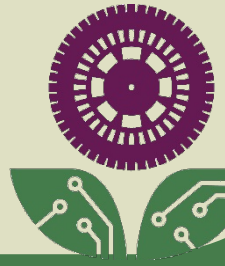
- What happens if DriveStiffness is too low?
- What happens if DriveStiffness is high?

Self-Driving!

Traffic

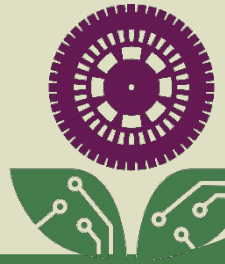
What if the Lead Gizmo Stops Short?

Test	SpringStiffness	StopDistance
Safest?	0.36	475
Least safe?	1.50	475
Safer?	1.50	350
What do you want to try?		



Aggressive Drivers

- All gizmos have StopDistance = 475
- Aggressive gizmo: SpringStiffness = 1.50
- Other gizmos: SpringStiffness = 0.36
- What happens when the aggressive driver is in the middle?
- What happens if there are two aggressive drivers?



Drive Safe!