



Virtual Activities with TinkerCAD Software using Arduino

**Dr. Murat Kuzlu
Jorge Ramirez III
Department of Engineering Technology**

Outline

- Arduino Introduction
- Component Overview
- Virtual Activities (1-5)



<https://medium.com/@otavioguastamacchia/creating-a-simple-iot-case-8102f22908a7>

Why Arduino ?

- It is an **open-source project**, software/hardware is extremely **accessible** and very flexible to be customized and extended
- It is **flexible**, offers a variety of digital and analog inputs, *SPI* and serial interface and digital and *PWM* outputs
- It is **easy to use**, connects to computer via USB and communicates using standard serial protocol, runs in standalone mode and as interface connected to PC/Macintosh computers
- It is **inexpensive**, and comes with free authoring software
- Arduino is backed up by a growing **online community**, lots of source code is already available and we can share and post our examples for others to use, too.

Autodesk TinkerCAD



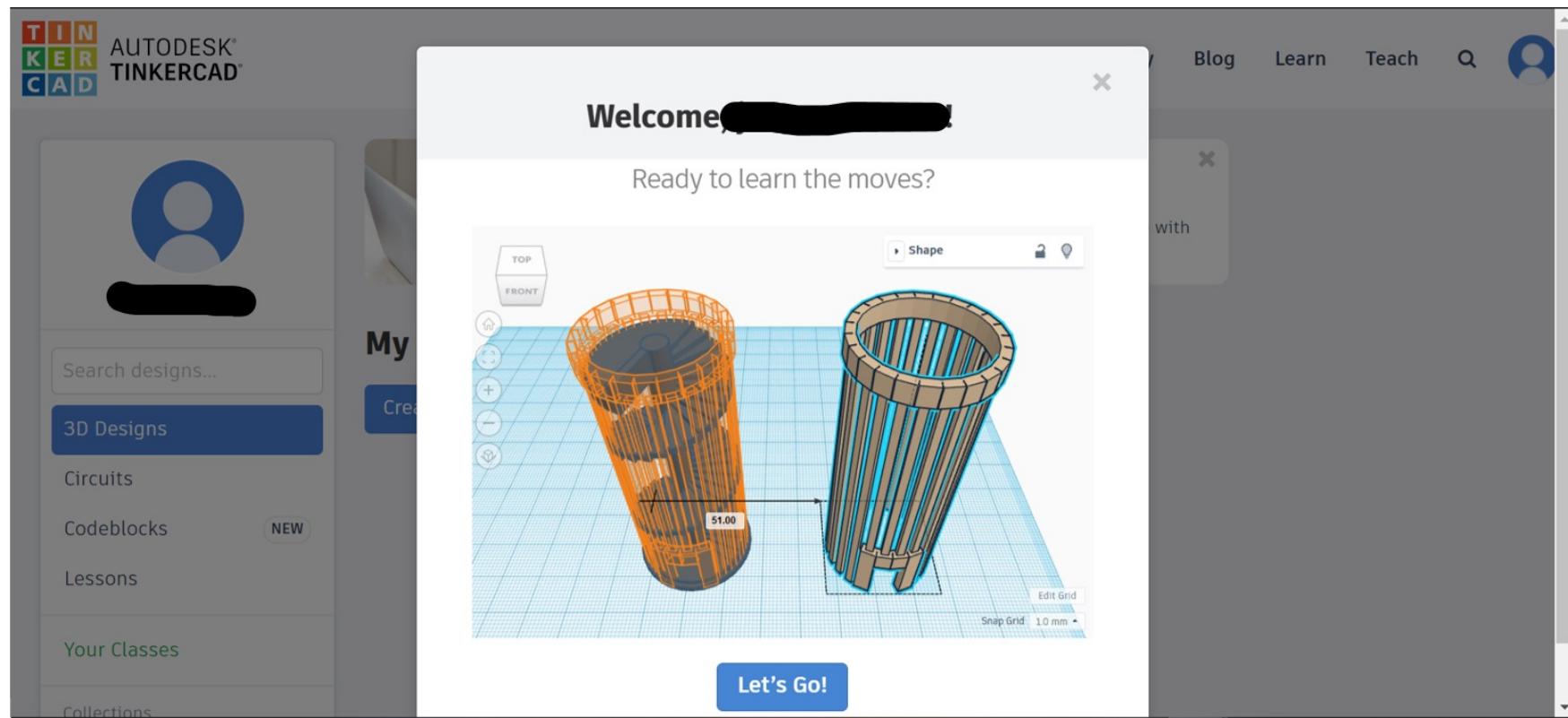
- Free online collection of software tools that help people all over the world think, create and make. A leader in 3D design, engineering and entertainment software.
- Arduino Simulation environment
- <https://all3dp.com/2/tinkercad-arduino-how-to-design-simulate-circuits/>

Creating an account with TinkerCAD

1. Go to <https://www.tinkercad.com/>
2. Top right corner, click on "Join Now"
3. At the bottom, click on "Create a personal account"
4. Select an email option that suits you
 - You may be asked to fill in extra information ex: Country, birthdate.
5. Enter email address and a password you wish to use
6. Click "Create account"

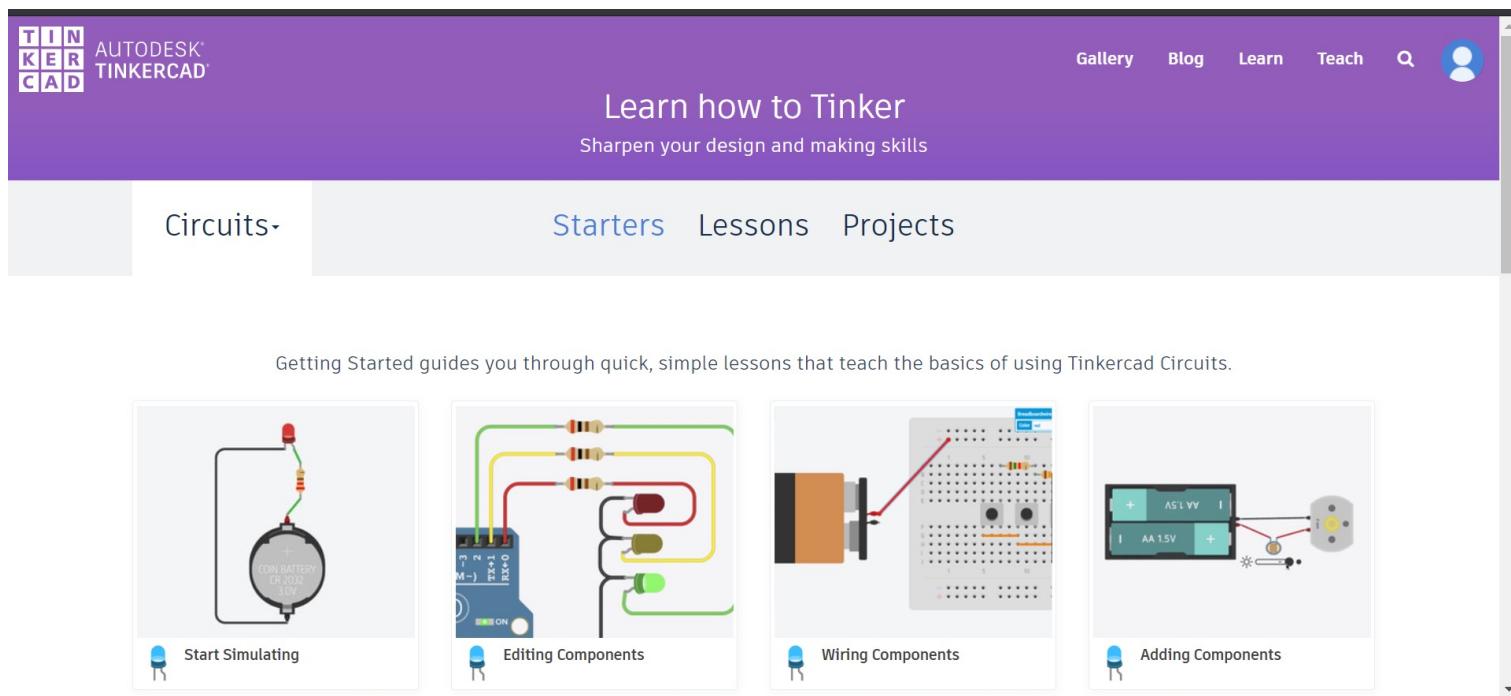
Creating an account with TinkerCAD

1. You should now see this screen, go ahead and close the welcome window and you'll be at your home profile page.



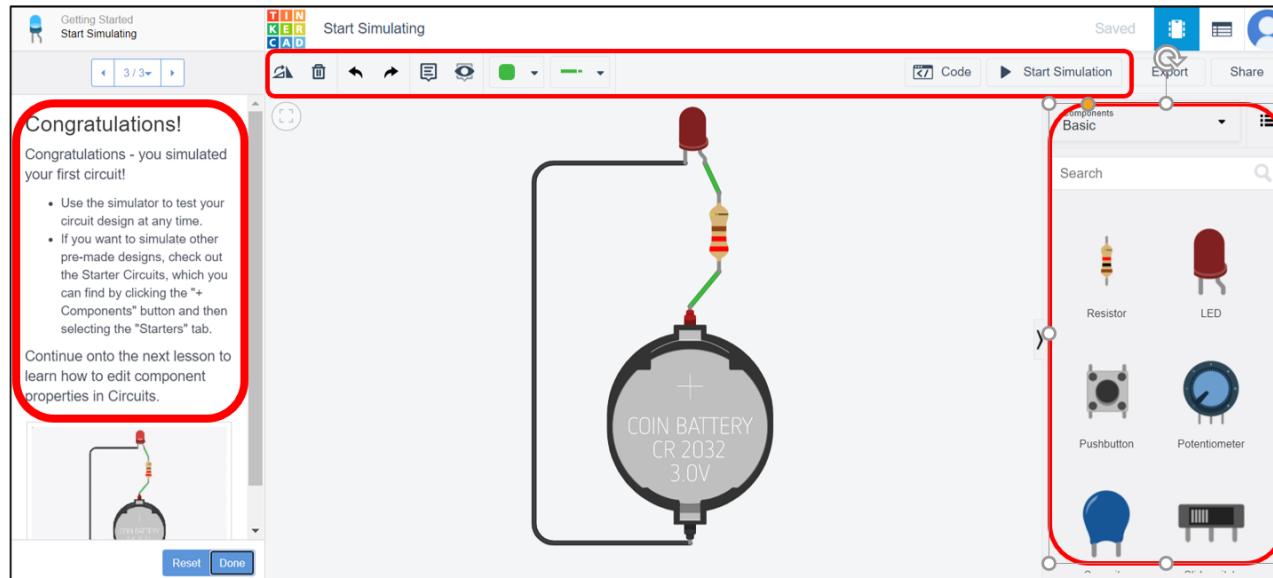
Getting Started with TinkerCAD

1. Top right corner click on "Learn"
2. On the left, click on the drop down menu "3D"
 - From the drop down menu select Circuits
 - Follow this link directly if needed: <https://www.tinkercad.com/learn/circuits/learning>



Getting Started with TinkerCAD

- Review four introduction lessons on TinkerCAD
 1. Start Simulating
 2. Editing components
 3. Wiring components
 4. Adding components
- Be sure to read through each step carefully on the left and explore as much as you want to get comfortable!



TinkerCAD Environment

Copy of Arduino-Basic-Project-#2

All changes saved

Code Start Simulation Export Share

1 (Arduino Uno R3)

The screenshot shows the TinkerCAD interface for an Arduino project. On the left, an Arduino Uno board is connected to a breadboard. Wires connect pin 4 to a green LED, pin 3 to a red LED, and pin 2 to a yellow LED. The breadboard has columns labeled a through j and rows labeled 1 through 24. On the right, the Arduino code is displayed:

```
//Date:09/27/2021
//Name:
int greenLed = 4;
int redLed = 3;
int yellowLed = 2;

void setup()
{
  pinMode(yellowLed, OUTPUT);
  pinMode(redLed, OUTPUT);
  pinMode(greenLed, OUTPUT);
}

void loop()
{
  digitalWrite(greenLed, HIGH);
  digitalWrite(redLed, HIGH);
  digitalWrite(yellowLed, HIGH);
  delay(1000); // Wait for 1000 millisecond(s)
  digitalWrite(greenLed, LOW);
  digitalWrite(redLed, LOW);
  digitalWrite(yellowLed, LOW);
  delay(1000); // Wait for 1000 millisecond(s)
}
```

Below the code is a "Serial Monitor" window.

Overview of components

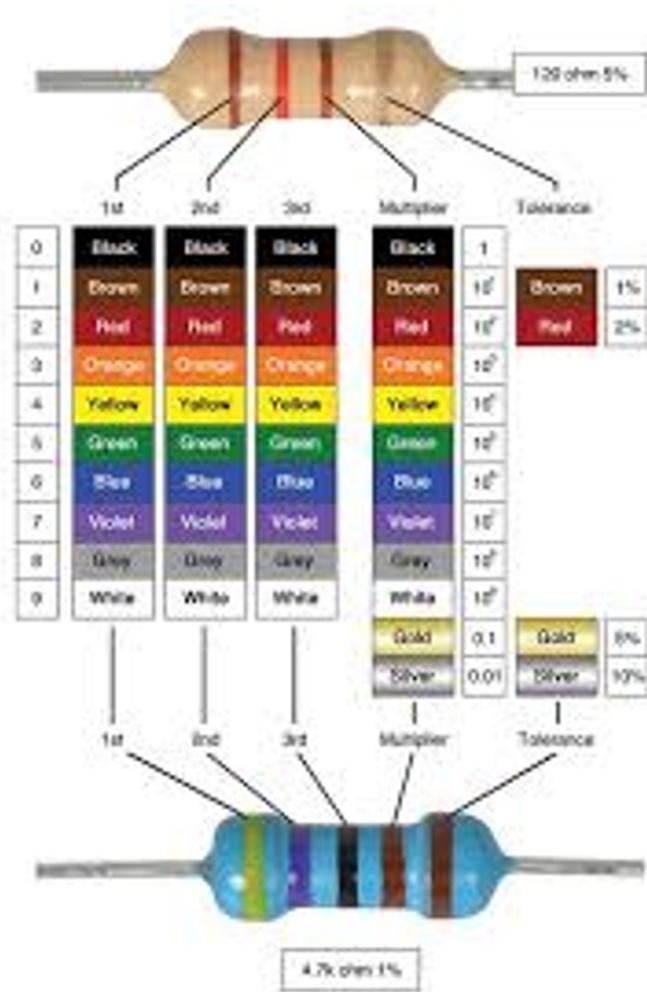
- The resistor
- The LED
- Arduino
- Breadboard

The resistor

Resistors are the **workhorse** of electronics

Resistance is **measured** in Ω (Ohm) and adds up in series; a resistors orientation doesn't matter

A resistors Ω value is **color-coded** right on it

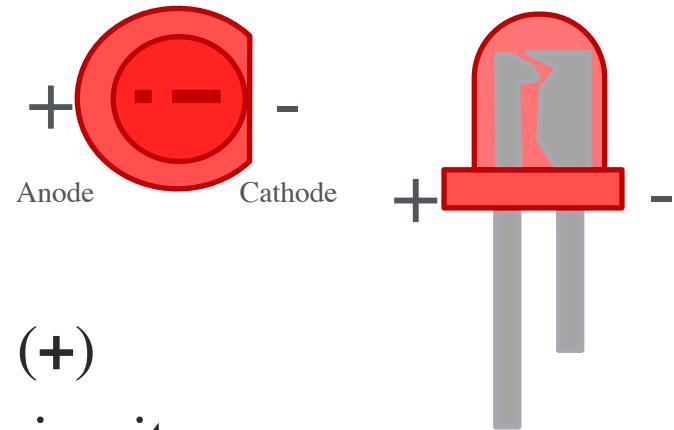


The LED

The **LED** (Light Emitting Diode)
is a simple, digital **actuator**

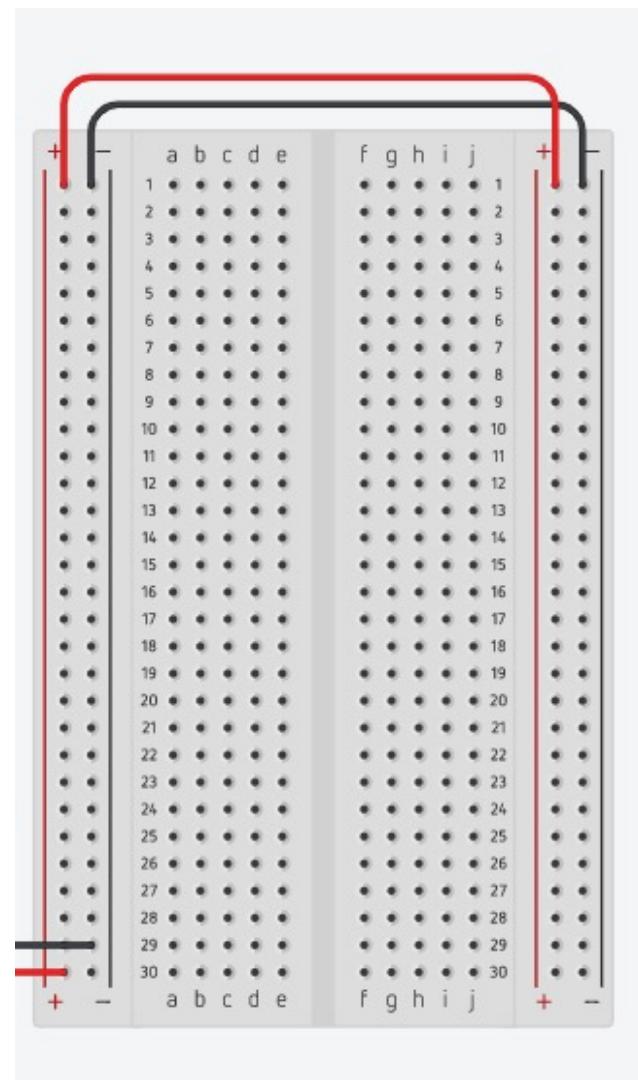
LEDs have a **short leg (-)** and a **long leg (+)**
and it matters how they are oriented in a circuit

To prevent damage, LEDs are used together with a $1K\Omega$
resistor (or anything from 300Ω to $2K\Omega$)



Breadboard

- The power bus
- Horizontal connections
- Vertical connections



Virtual Activity - 1

Blinking LED

Starting a new circuit

Dashboard | Tinkercad [x](#) [+](#) tinkercad.com/dashboard?&type=tinkercad&collection=designs [🔍](#) [☆](#) [ABP](#)

**TIN
KER
CAD** AUTODESK® TINKERCAD®

Log in to www.tinkercad.com

Tinkercad Lesson Plans
Tinkercad lesson plans are ready to use online or in the classroom. Discover curriculum developed in partnership with teachers. [Learn more](#)

My recent designs

Create new design

3D Designs **Circuits** **Codeblocks** **Lessons**

Click "Circuits."

Ben Finio

Search designs...

Cool Snicket 5 months ago Private

Glorious Kieran-Sango 8 months ago Private

Stunning Luulia 5 years ago Private

Powerful Amberis-Wolt 5 years ago Private

Project 1

Create project

Tweets Follow

Tinkercad Retweeted

Allison Debellis @adebellis8

15

Starting a new circuit



Gallery Blog Learn Teach Q

Ben Finio

Search designs...

3D Designs

Circuits

Codeblocks

NEW

Lessons

Your Classes

Projects

Project 1

Create project

Tweets

Follow

Tinkercad Retweeted

Allison Debellis
@adebellis8

Tinkercad Lesson Plans

Tinkercad lesson plans are ready to use online or in the classroom. Discover curriculum developed in partnership with teachers. [Learn more](#)

Circuits

Create new Circuit

Click "Create new Circuit."

Example Arduino Light Show (With R...

5 hours ago
Private

My First Arduino Circuit

7 hours ago
Public

Brave Wolt

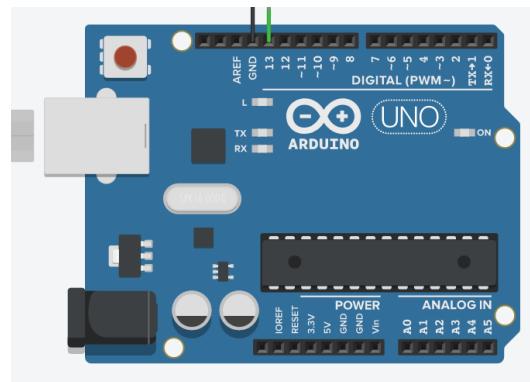
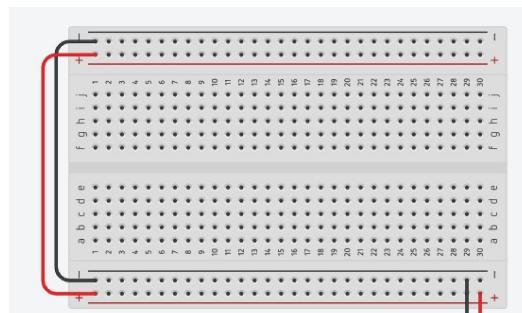
9 hours ago
Private

Spectacular Stantia-Jaiks

5 days ago
Private

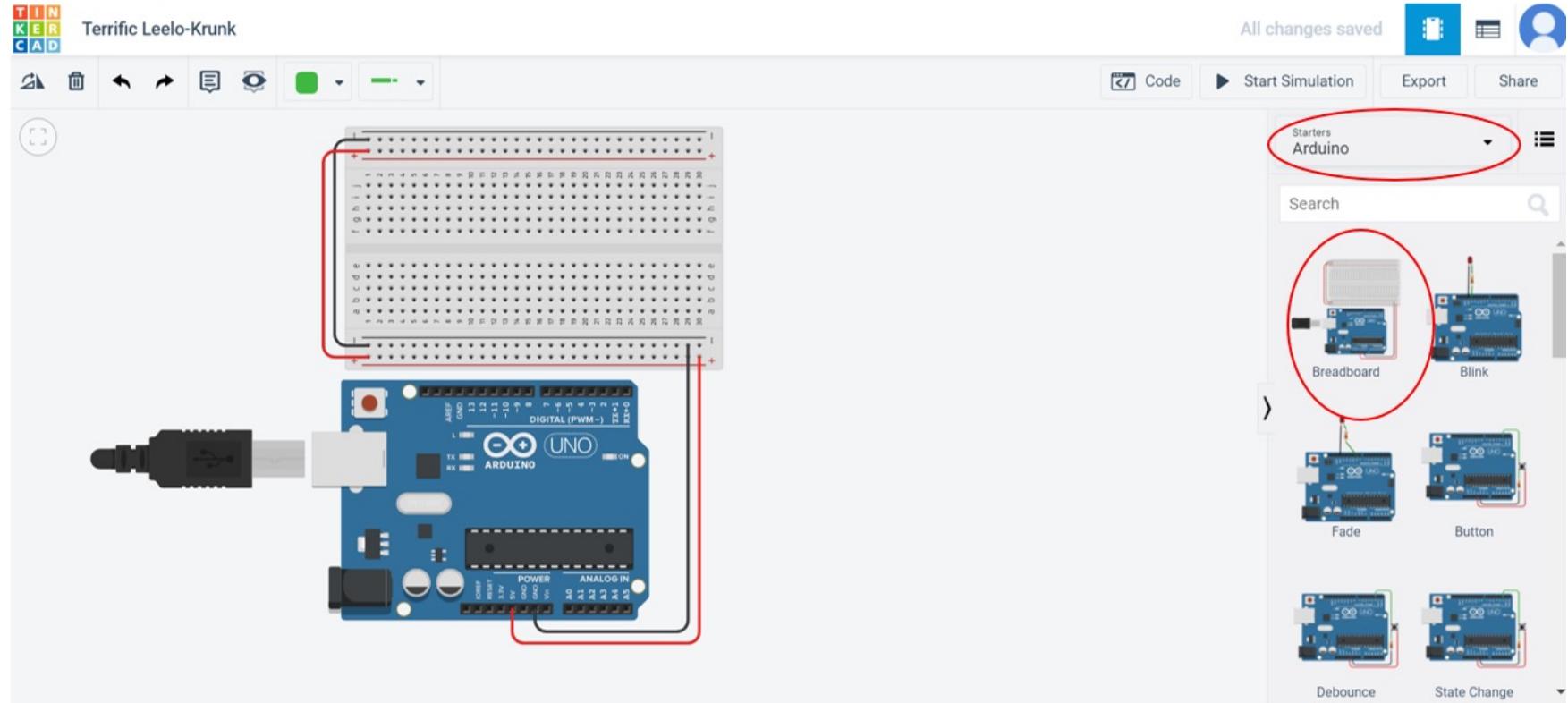
Components to be used

- LED
- Resistor
- Arduino
- Breadboard



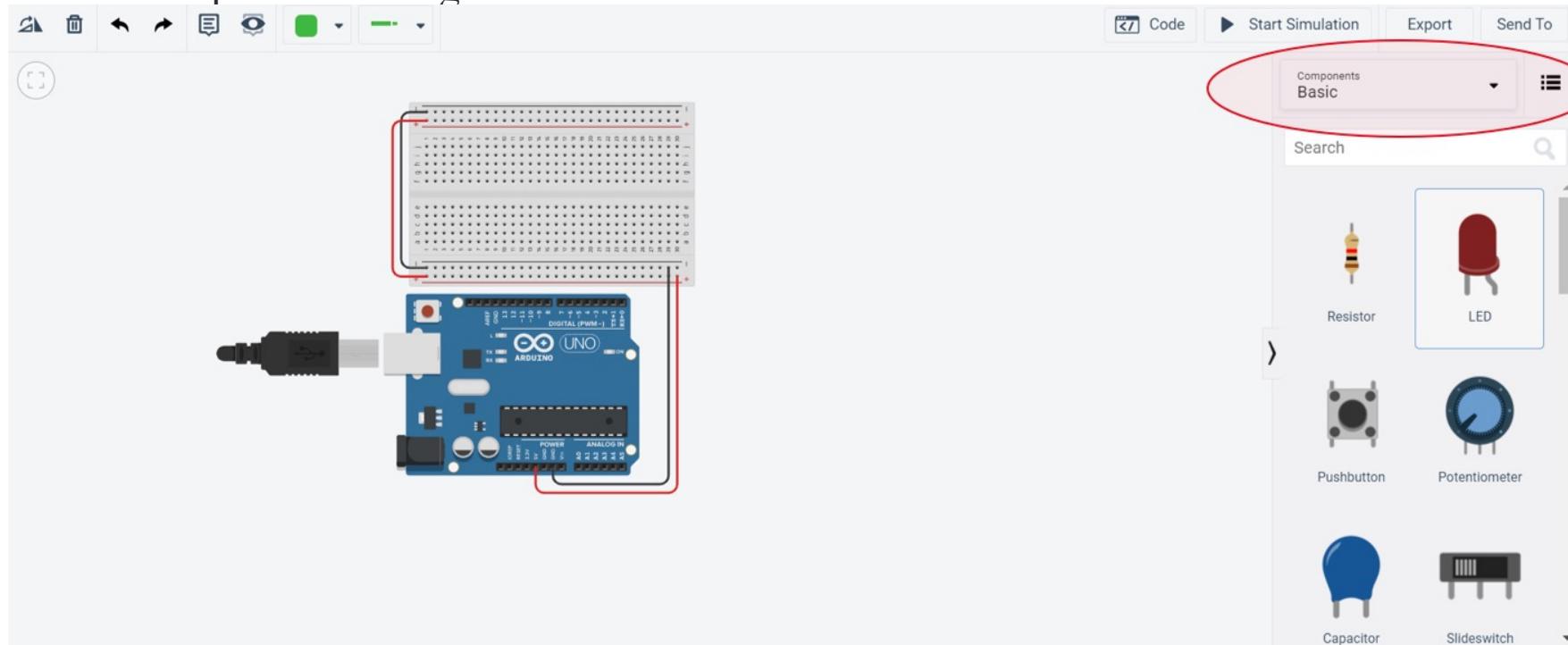
Setting up our Workplace

1. Under Starters/Components: click Arduino.
2. Click and drag on the first component, place it on the workplace and let go.



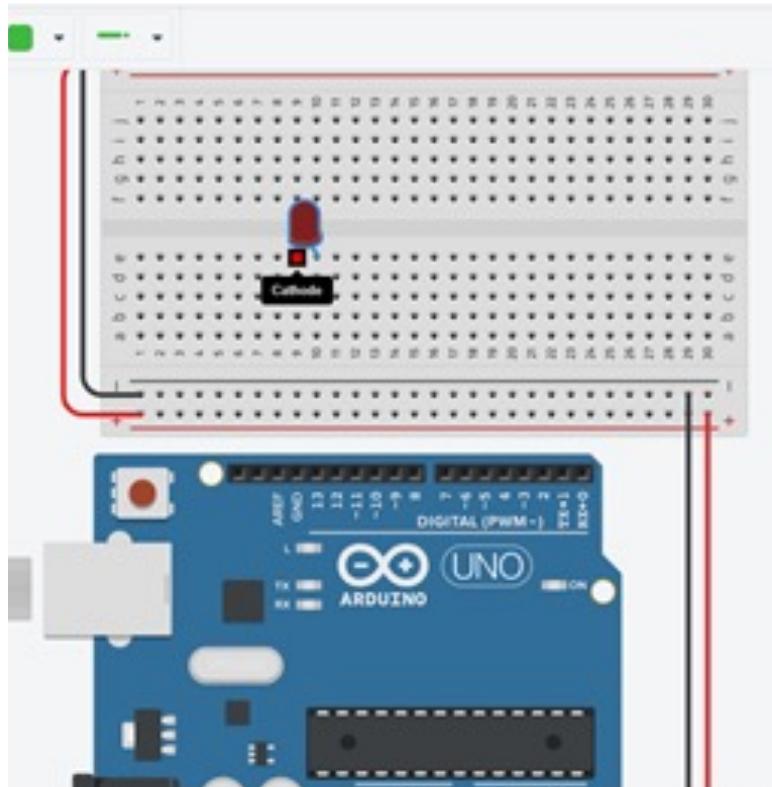
Setting up our Workplace

1. Under Basic/Components: click on the LED.
2. Click and drag on the first component, place it on the workplace and let go.

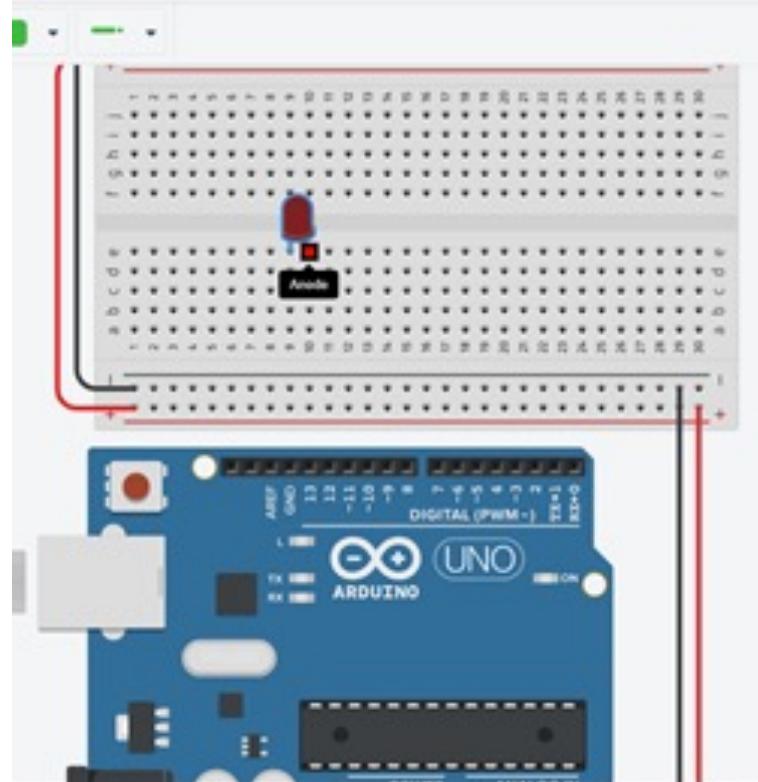


Setting up our Workplace

1. Under Basic/Components: click on the LED.
2. Click and drag on the first component, place it on the workplace and let go.



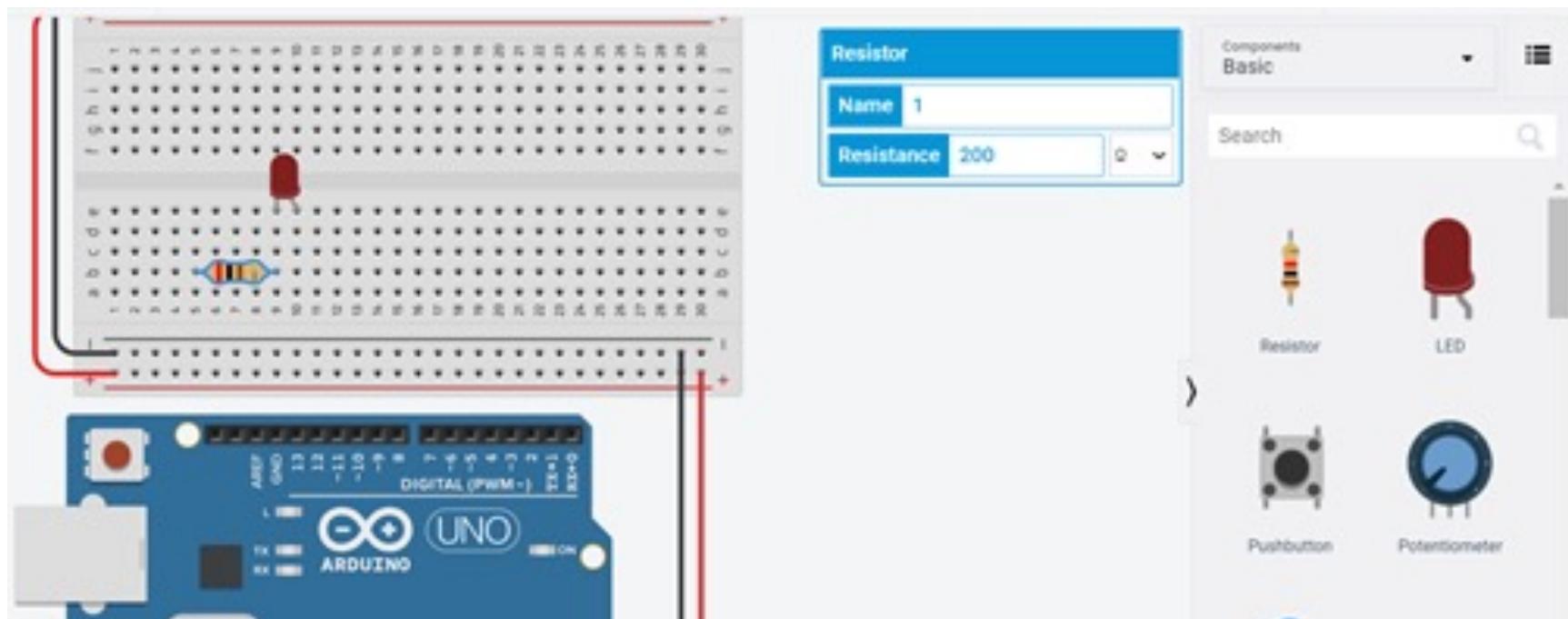
*Notice the Cathode is on Row 9, Column e. And the Anode is on Row 10, Column e.



Setting up our Workplace

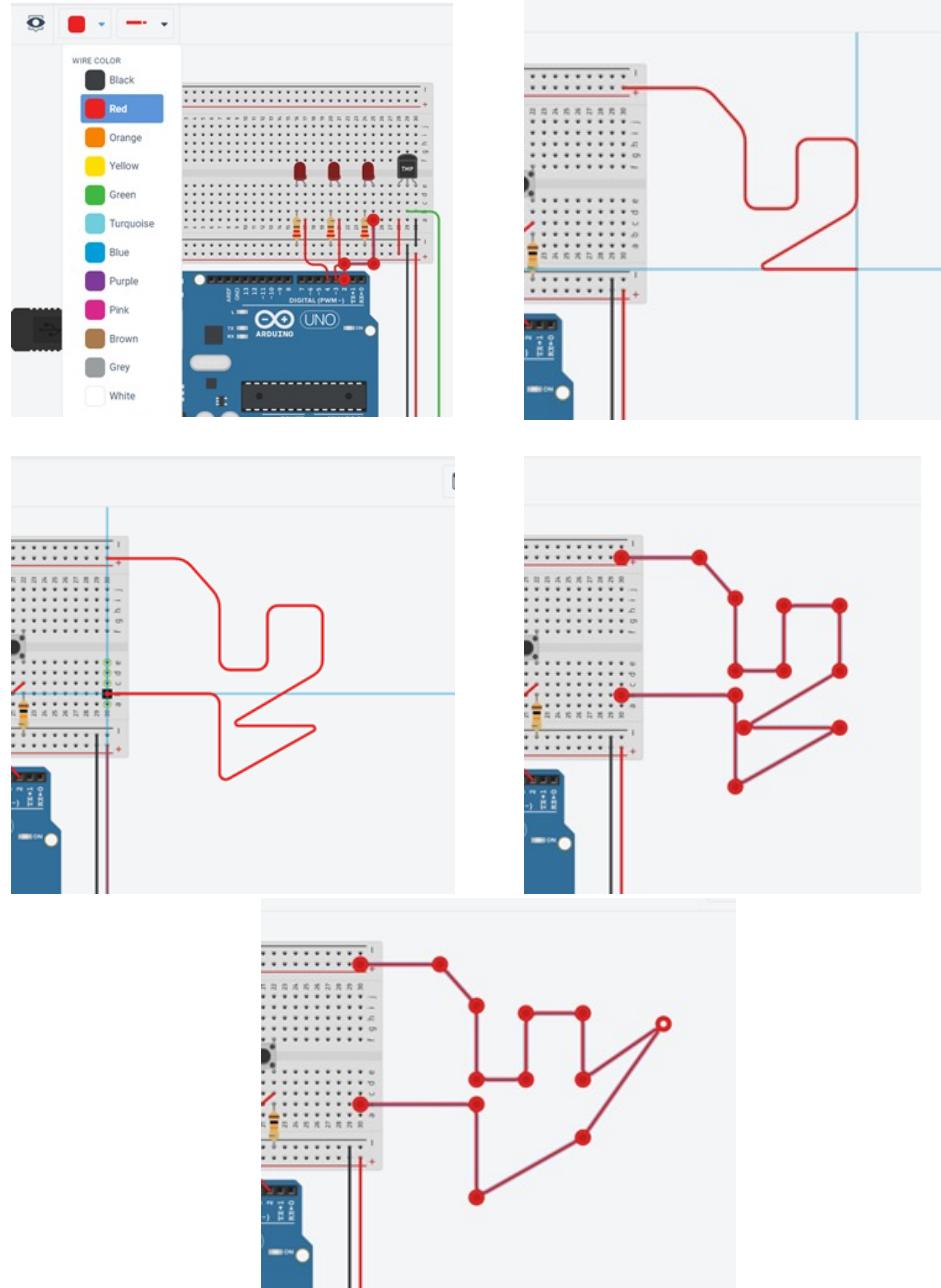
1. Under Basic/Components: click on the Resistor.
2. Click and drag on the component, place it on the workplace and let go.

- Notice one end is on Row 5, Column b.
And the other end is on Row 9, Column b.
- You can rotate the resistor by press "R" on your keyboard.
- Set your resistor to 200 Ohms



Wiring Orientation

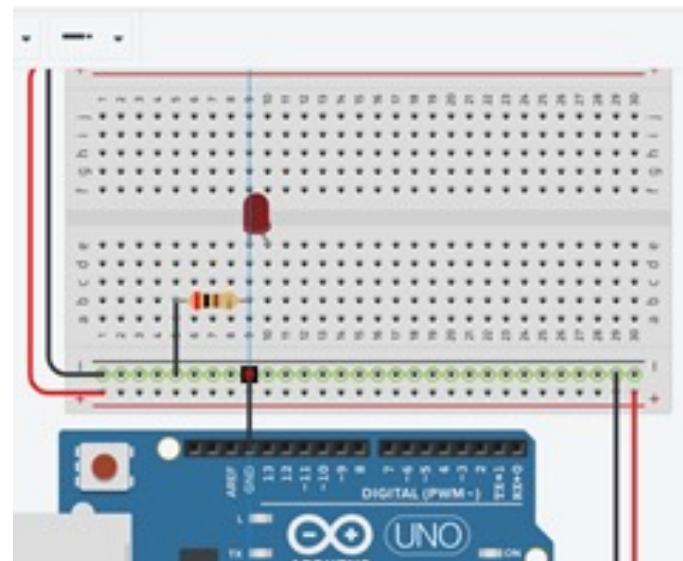
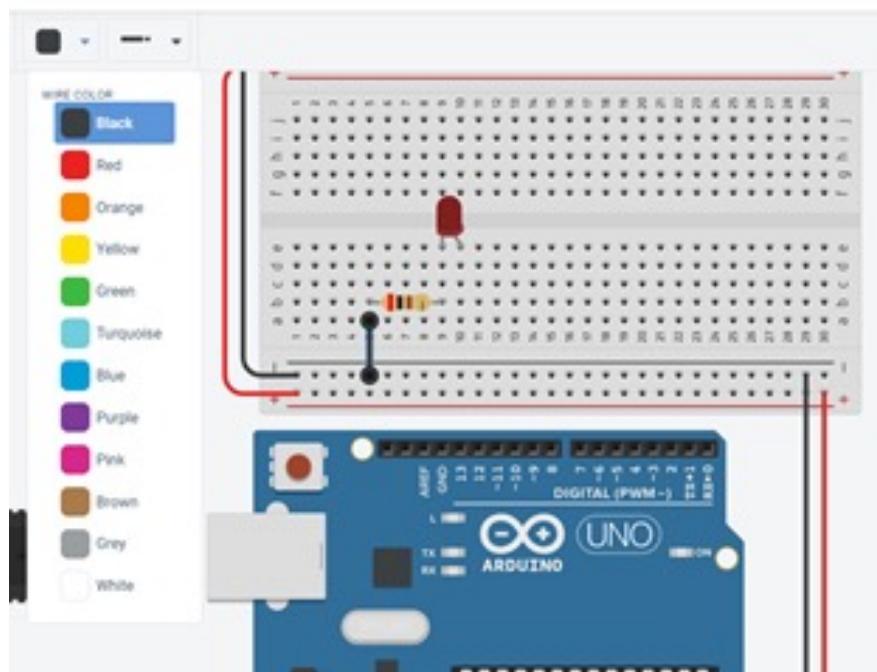
1. Choose a color for the wire.
Commonly, your ground will be black, and power will be red.
2. Next choose a start point and click once, you must choose a port or a location on the bread board. You will now have a wire that can be stretched almost at any angle.
3. In order to make your wire "curve" at different points, you can right click, as long as you do not click when hovering over a port or else it'll assumingly connect to that port.
4. You add as many turning points as you want, and they will appear after you connect to a port.
5. After which you can stretch the wire through its' turning points.



Wiring a LED with Arduino

Set up

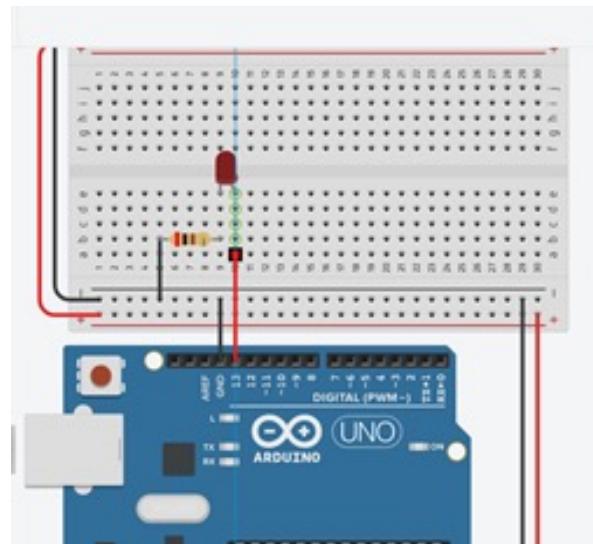
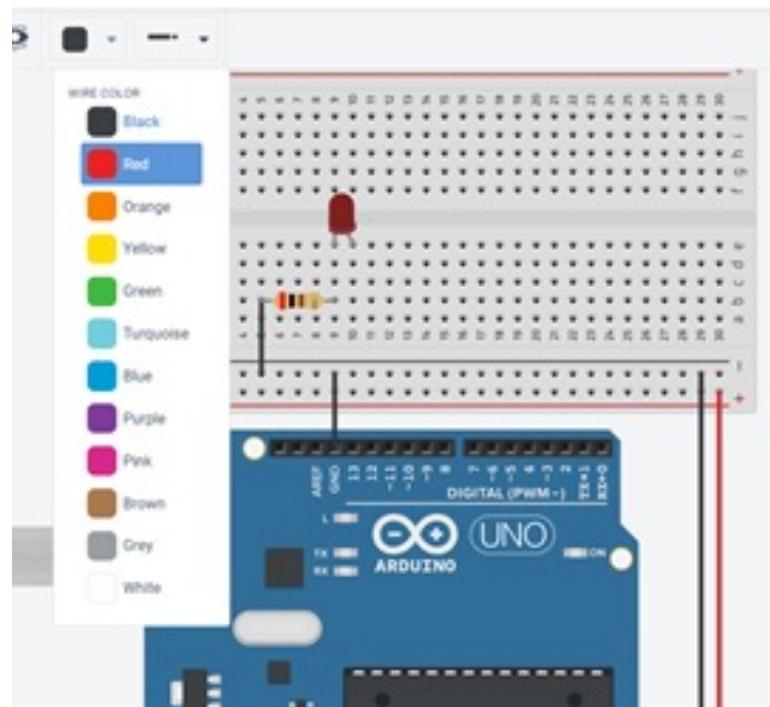
1. Select the color black for the following wires as seen in the image.
2. One end starts at the Ground bus (row 5). The other end connects to Row 5, Column a.
3. The second wire connects from the GND port on the Arduino to the Ground bus, aligned with row 9. This provide a return path to ground while using a pull down resistor on the Cathode side of the LED.



Wiring a LED with Arduino

Set up

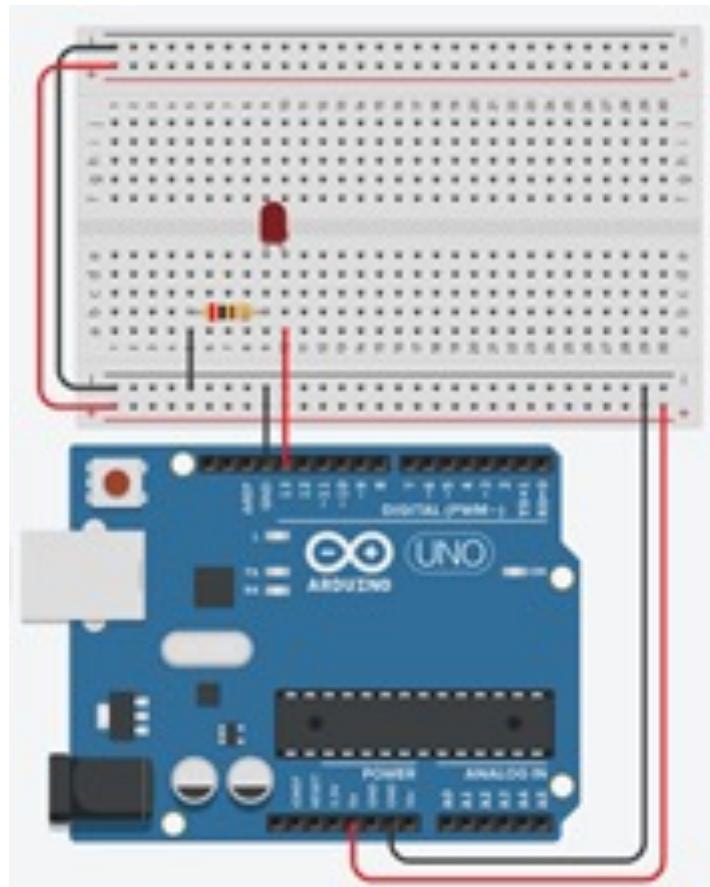
1. Select the color red for the following wire as seen in the image.
2. One end starts at the port on the Arduino, Port 13.
3. The other end of the wire connects to Row 10, Column a. This provides power to the Anode side of the LED.



Wiring a LED with Arduino Cont.

Set up

- Ensure hardware is wired as shown on the image.
- Next will be the coding process.



C++ Review

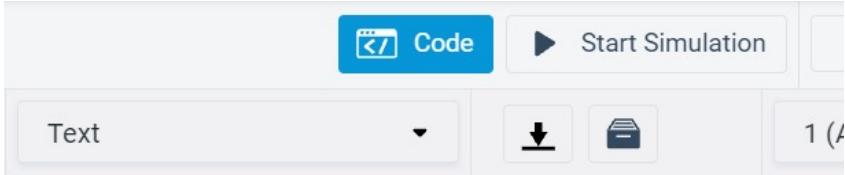
Code

1. Click on "Code" button to open the coding window.
2. Review code format:

```
// Comments for the user
```

```
void setup()
{
    variables; // here we setup our variables
}
```

```
void loop()
{
    predefined variables; // here we setup our loop
    delay(1000); // wait for 1 second
}
```



```
1 // C++ code
2 //
3 void setup()
4 {
5     pinMode(13, OUTPUT);
6 }
7
8 void loop()
9 {
10    digitalWrite(13, HIGH);
11    delay(1000); // Wait for 1000 millisecond(s)
12    digitalWrite(13, LOW);
13    delay(1000); // Wait for 1000 millisecond(s)
14 }
```

Copy code C++

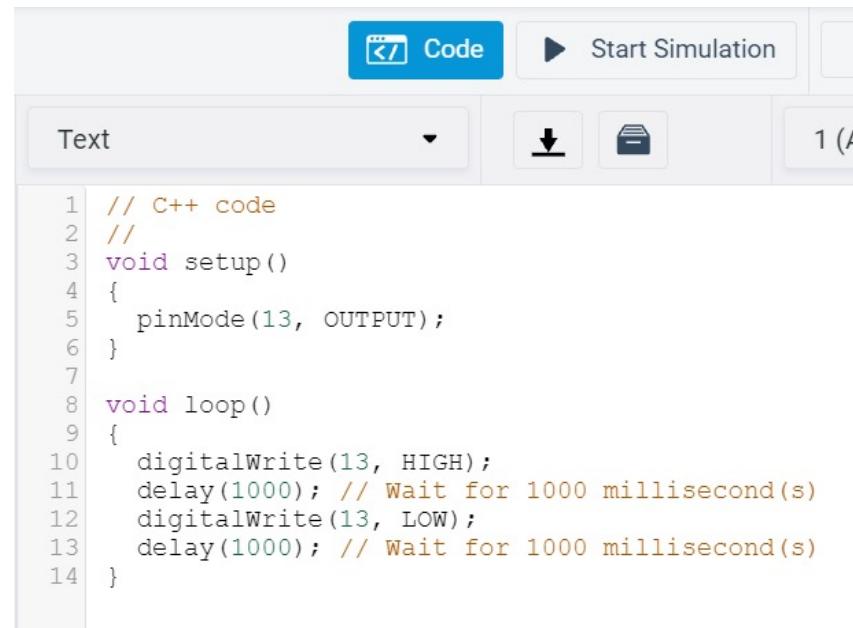
Code

1. When code is similar to the image on the right then go ahead and Click "Start Simulation".

2. Copy Code:

```
// C++ code  
//  
void setup()  
{  
    pinMode(13, OUTPUT);  
}
```

```
void loop()  
{  
    digitalWrite(13, HIGH);  
    delay(1000); // Wait for 1000 millisecond(s)  
    digitalWrite(13, LOW);  
    delay(1000); // Wait for 1000 millisecond(s)  
}
```

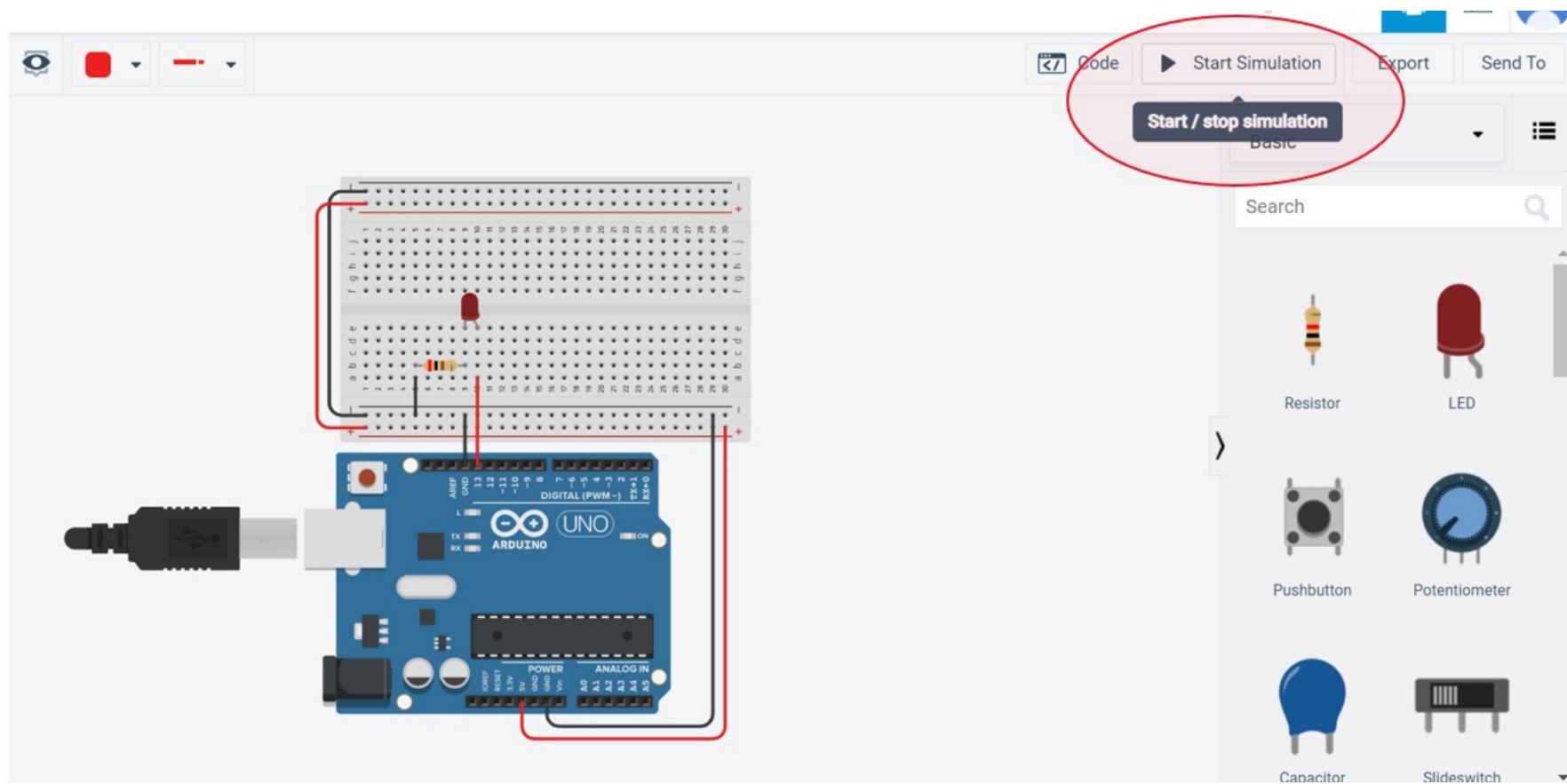


```
1 // C++ code  
2 //  
3 void setup()  
4 {  
5     pinMode(13, OUTPUT);  
6 }  
7  
8 void loop()  
9 {  
10    digitalWrite(13, HIGH);  
11    delay(1000); // Wait for 1000 millisecond(s)  
12    digitalWrite(13, LOW);  
13    delay(1000); // Wait for 1000 millisecond(s)  
14 }
```

Begin Simulation

Blink LED

Click on Start Simulation to begin simulation and you'll see the LED start to blink.



Simulation

Blink LED

TIN KER CAD Arduino Circuit All changes saved

If your LED does not blink, carefully double-check your breadboard wiring. For example, can you tell what is wrong with this circuit? (go to next slide for answer)

```
void loop()
{
    digitalWrite(13, HIGH);
    delay(1000); // Wait for 1000 millisecond(s)
    digitalWrite(13, LOW);
    delay(1000); // Wait for 1000 millisecond(s)
}
```

Simulation

Blink LED

TIN KER CAD Arduino Circuit All changes saved Start Simulation Export Share

The resistor is not in the same row as the short leg of the LED. They are not connected. This is an open circuit, so the LED will not light up.

```
digitalWrite(13, HIGH);
delay(1000); // Wait for 1000 millisecond(s)
digitalWrite(13, LOW);
delay(1000); // Wait for 1000 millisecond(s)
```

Simulation

Blink LED

ARDUINO CIRCUIT

All changes saved

Code Start Simulation Export Share

Text

```
1 // C++ code
2 //
3 void setup()
4 {
5   pinMode(13, OUTPUT);
6
7   loop()
8
9   digitalWrite(13, HIGH);
10  delay(1000); // Wait for 1000 millisecond(s)
11  digitalWrite(13, LOW);
12  delay(1000); // Wait for 1000 millisecond(s)
13
14 }
```

How about this circuit? Can you tell what is wrong here? (Go to next slide for answer.)

The diagram shows an Arduino Uno R3 microcontroller connected to a breadboard. A red LED is connected between pin 13 (labeled 'D13' on the board) and ground. The breadboard has two columns of 30 rows each, labeled 1 through 30. The red line from the Arduino's pin 13 goes to the top rail of the breadboard, then to the leftmost column of pins, and finally to the positive terminal of the red LED. The negative terminal of the LED is connected to the bottom rail of the breadboard, which is connected to ground. A red box highlights the connection from pin 13 to the LED's positive terminal. A callout box contains the text: "How about this circuit? Can you tell what is wrong here? (Go to next slide for answer.)".

Simulation

Blink LED

TIN KER CAD Arduino Circuit All changes saved

Code Start Simulation Export Share

Text

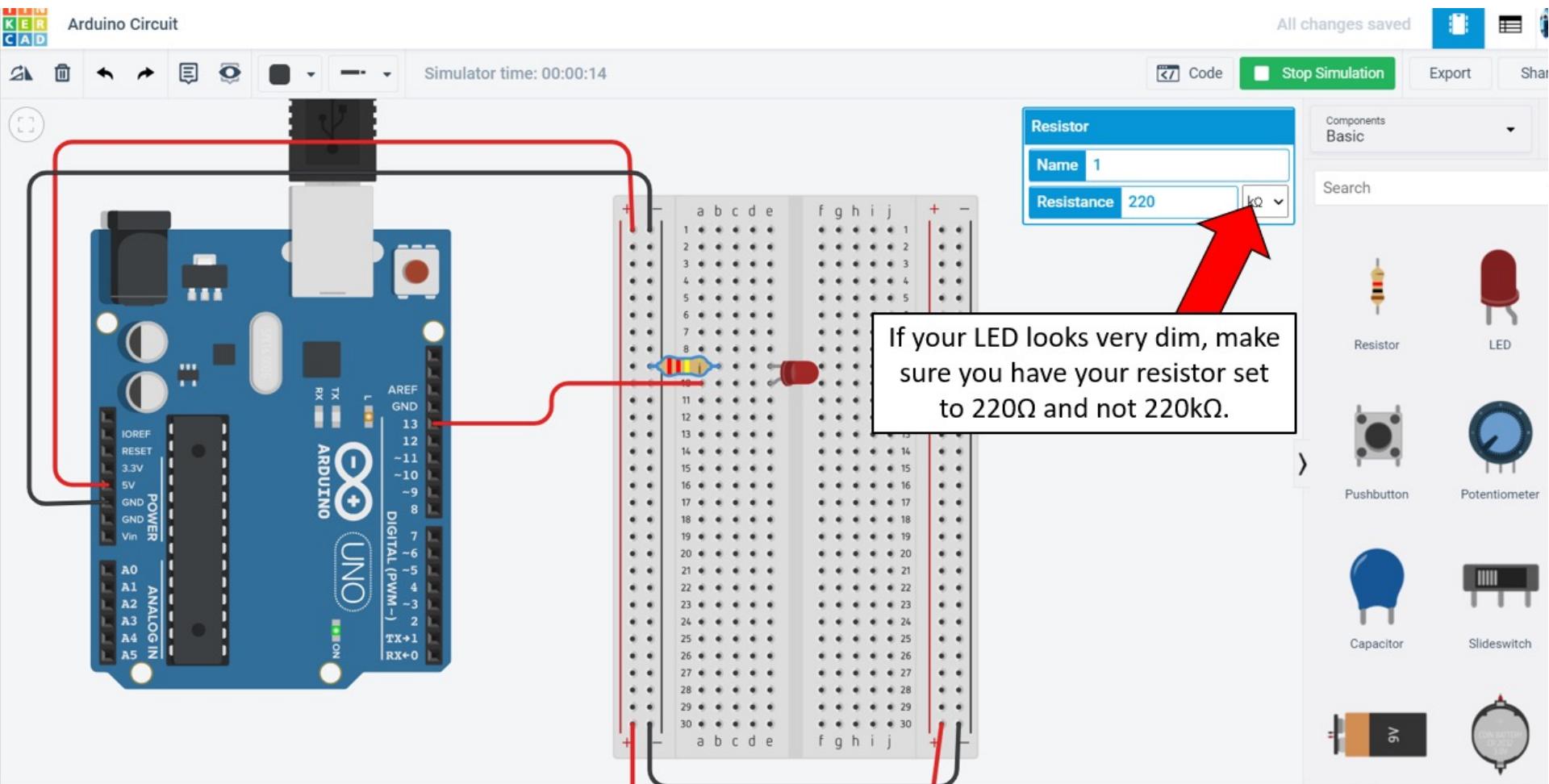
1 (Arduino Uno R3)

Both legs of the LED are in the same breadboard row. These holes are connected to each other, so this creates a short circuit. The LED will not light up.

```
1000 millisecond(s)
digitalWrite(13, LOW);
delay(1000); // Wait for 1000 millisecond(s)
}
```

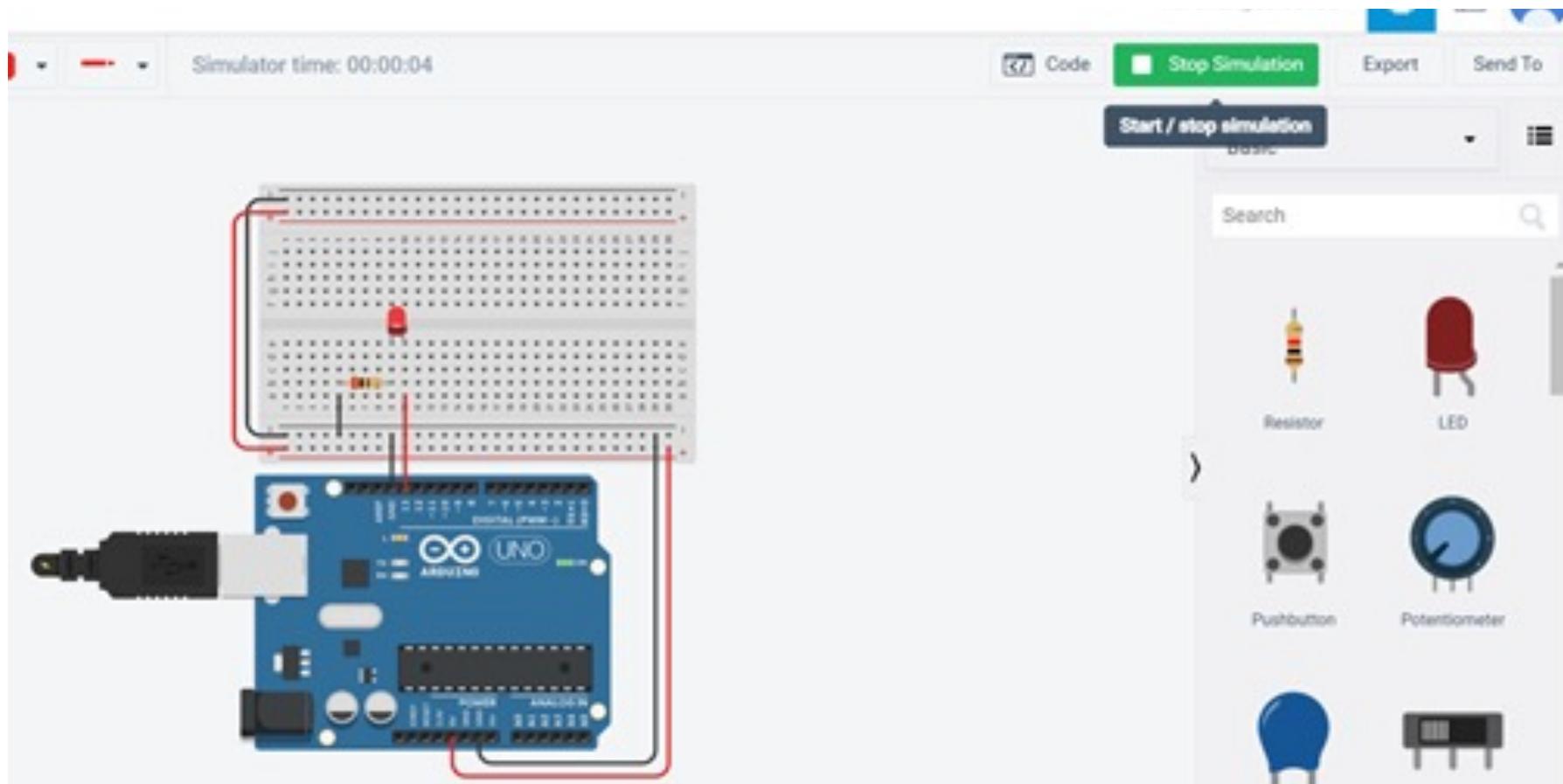
Simulation

Blink LED



Stop simulation

Click on Stop Simulation to end simulation and you'll see the LED stop blinking.



What did we witness?

- Did you notice the small LED flashing on the board itself? This built in LED is also connected to pin 13, and is meant to be used for testing purposes without the need to connect any external components. It even has its own tiny resistor, soldered directly to the surface of the board

Virtual Activity – 2

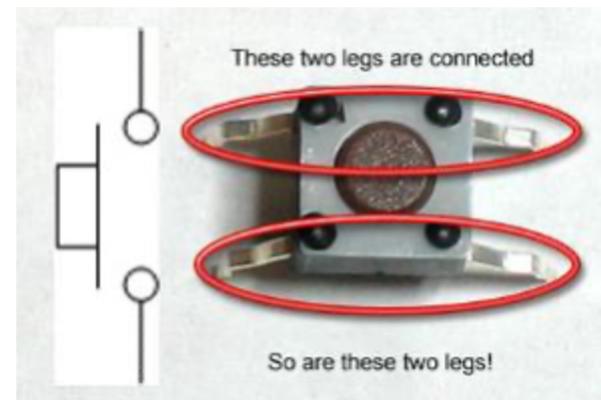
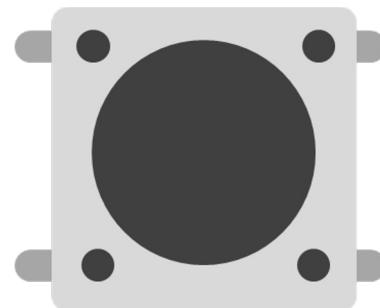
Arduino Switch

The switch

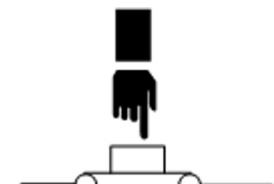
A switch is a simple, digital **sensor**

Switches come in different forms, but all of them in some way **open** or **close** a gap in a wire

The **pushbutton** switch has four legs for easier mounting, but only two of them are needed



Normally Open
When not pressed



Closed
When pressed

Starting a new circuit

Dashboard | Tinkercad x + tinkercad.com/dashboard?&type=tinkercad&collection=designs ...

 **AUTODESK® TINKERCAD®**

Log in to www.tinkercad.com

Tinkercad Lesson Plans
Tinkercad lesson plans are ready to use online or in the classroom. Discover curriculum developed in partnership with teachers. [Learn more](#)

My recent designs

Create new design

3D Designs ←

Circuits Click "Circuits." NEW

Codeblocks

Lessons

Your Classes

Projects

Project 1

Create project

Tweets Follow

Tinkercad Retweeted

 **Allison Debellis** @adebellis8

Ben Finio

Search designs...

Cool Snicket 5 months ago Private

Glorious Kieran-Sango 8 months ago Private

Stunning Luulia 5 years ago Private

Powerful Amberis-Wolt 5 years ago Private

3D Designs

Cool Snicket 5 months ago Private

Glorious Kieran-Sango 8 months ago Private

Stunning Luulia 5 years ago Private

Powerful Amberis-Wolt 5 years ago Private

Circuits

Codeblocks NEW

Lessons

Your Classes

Projects

Project 1

Create project

Tweets Follow

Tinkercad Retweeted

 **Allison Debellis** @adebellis8

3D Designs

Cool Snicket 5 months ago Private

Glorious Kieran-Sango 8 months ago Private

Stunning Luulia 5 years ago Private

Powerful Amberis-Wolt 5 years ago Private

Circuits

Codeblocks NEW

Lessons

Your Classes

Projects

Project 1

Create project

Tweets Follow

Tinkercad Retweeted

 **Allison Debellis** @adebellis8

Starting a new circuit



Gallery Blog Learn Teach Q

Ben Finio

Search designs...

3D Designs

Circuits

Codeblocks

NEW

Lessons

Your Classes

Projects

Project 1

Create project

Tweets

Follow

Tinkercad Retweeted

Allison Debellis

@adebellis8

Tinkercad Lesson Plans

Tinkercad lesson plans are ready to use online or in the classroom. Discover curriculum developed in partnership with teachers. [Learn more](#)

Circuits

Create new Circuit

Click "Create new Circuit."

Example Arduino Light Show (With R...

5 hours ago
Private

My First Arduino Circuit

7 hours ago
Public

Brave Wolt

9 hours ago
Private

Spectacular Stantia-Jaiks

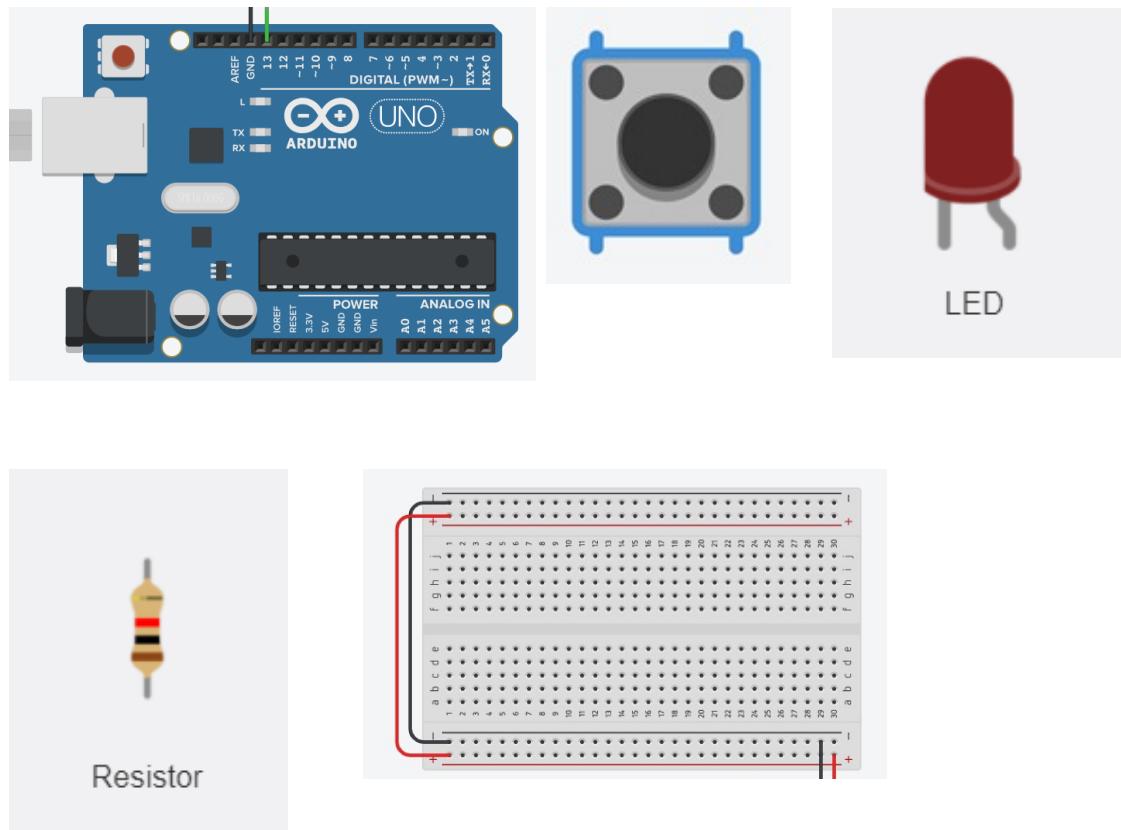
5 days ago
Private

Components List

We will introduce how to blink the on-board LED and how to blink an external LED with a switch.

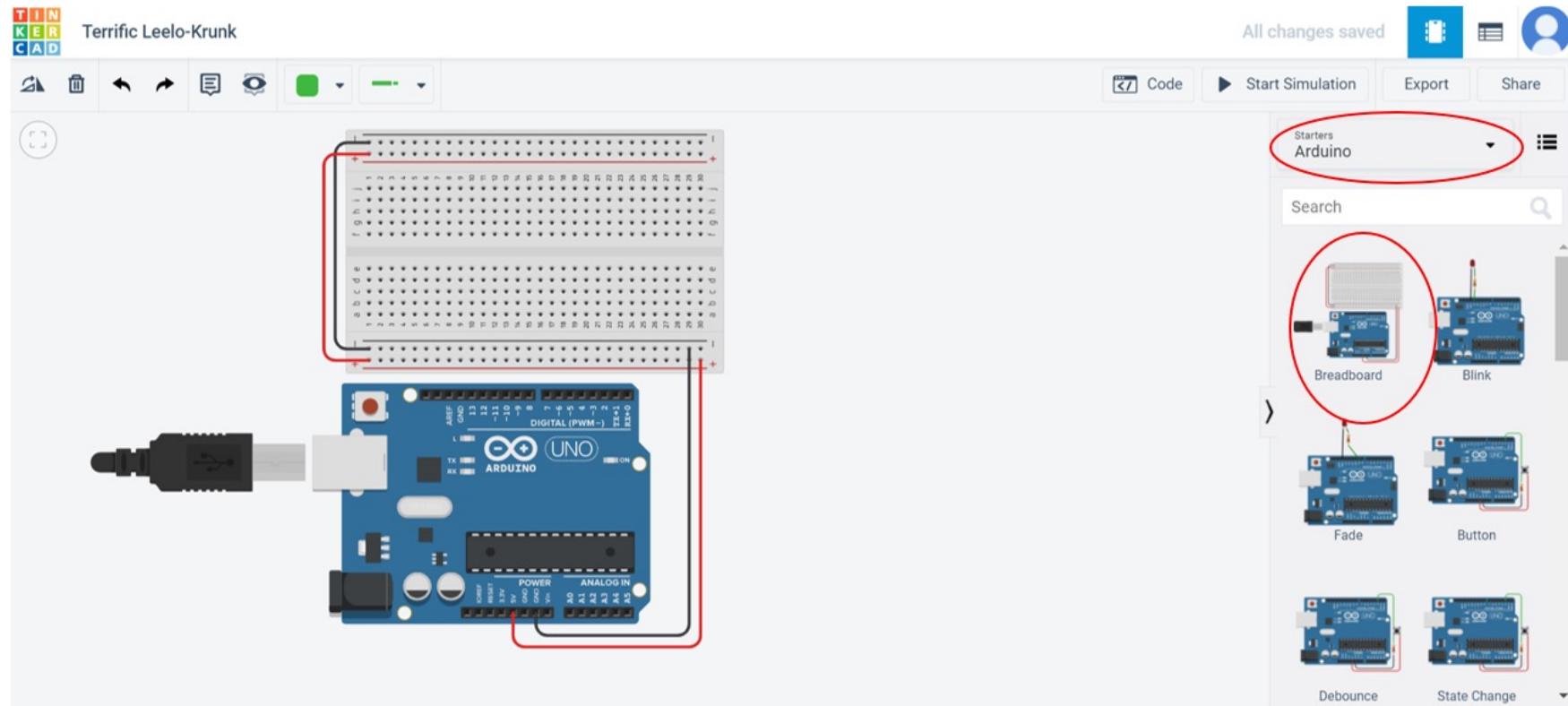
Components

- Arduino x1
- Push Button x1
- LED x1
- $10\text{ K}\Omega$ Resistor x1
- $200\ \Omega$ Resistor x1
- Bread Board x1



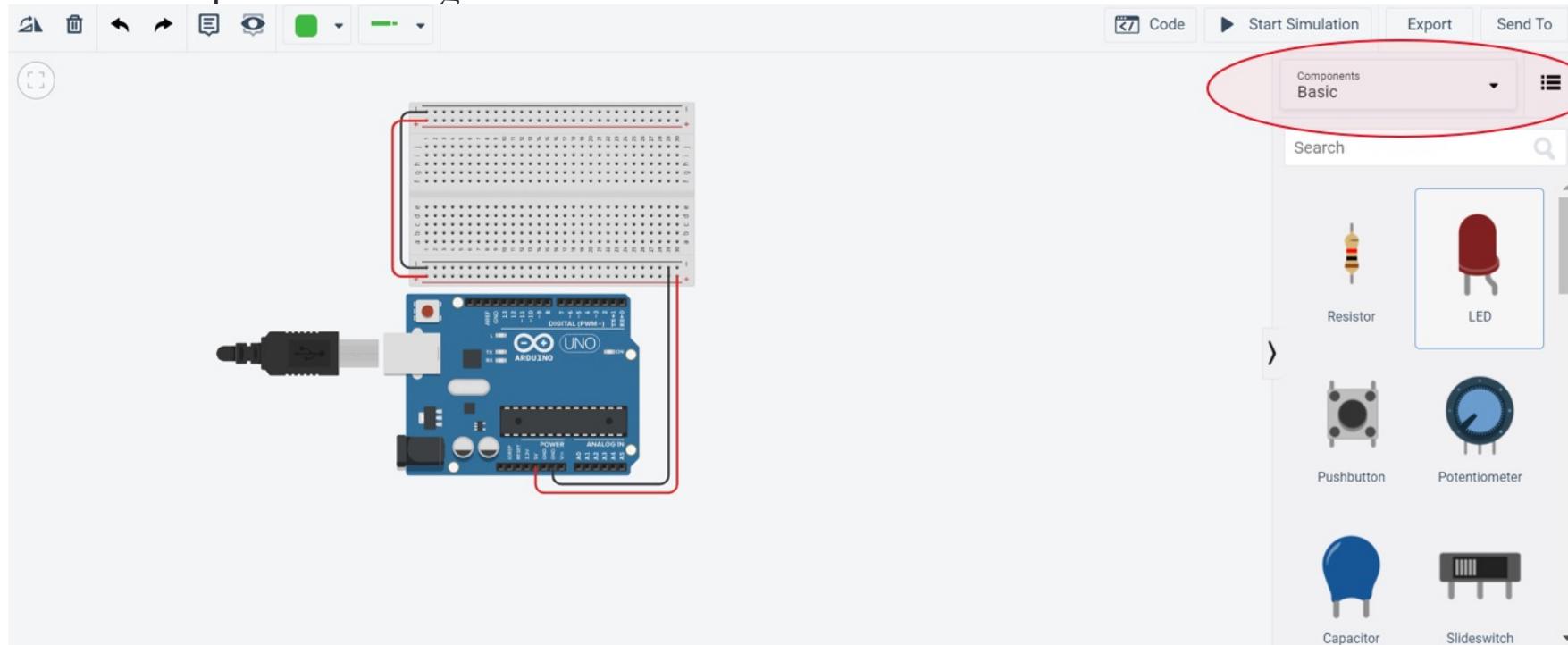
Setting up our Workplace

1. Under Starters/Components: click Arduino.
2. Click and drag on the first component, place it on the workplace and let go.



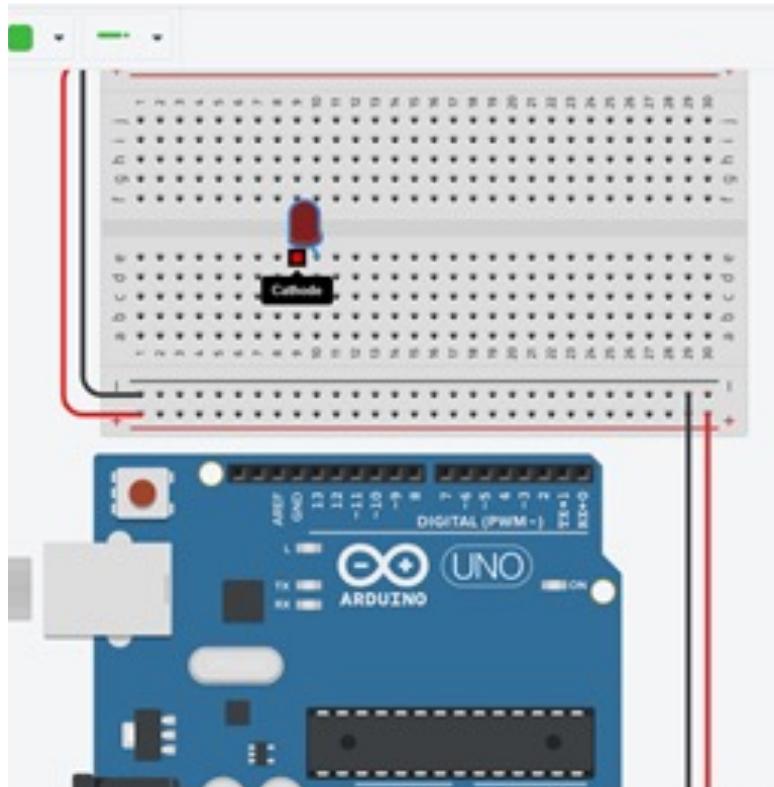
Setting up our Workplace

1. Under Basic/Components: click on the LED.
2. Click and drag on the first component, place it on the workplace and let go.

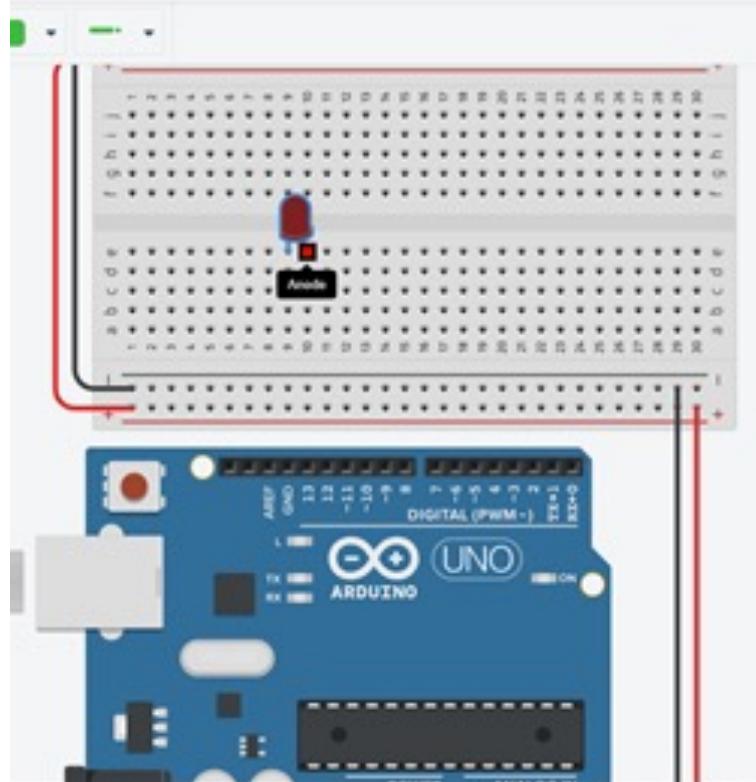


Setting up our Workplace

1. Under Basic/Components: click on the LED.
2. Click and drag on the first component, place it on the workplace and let go.



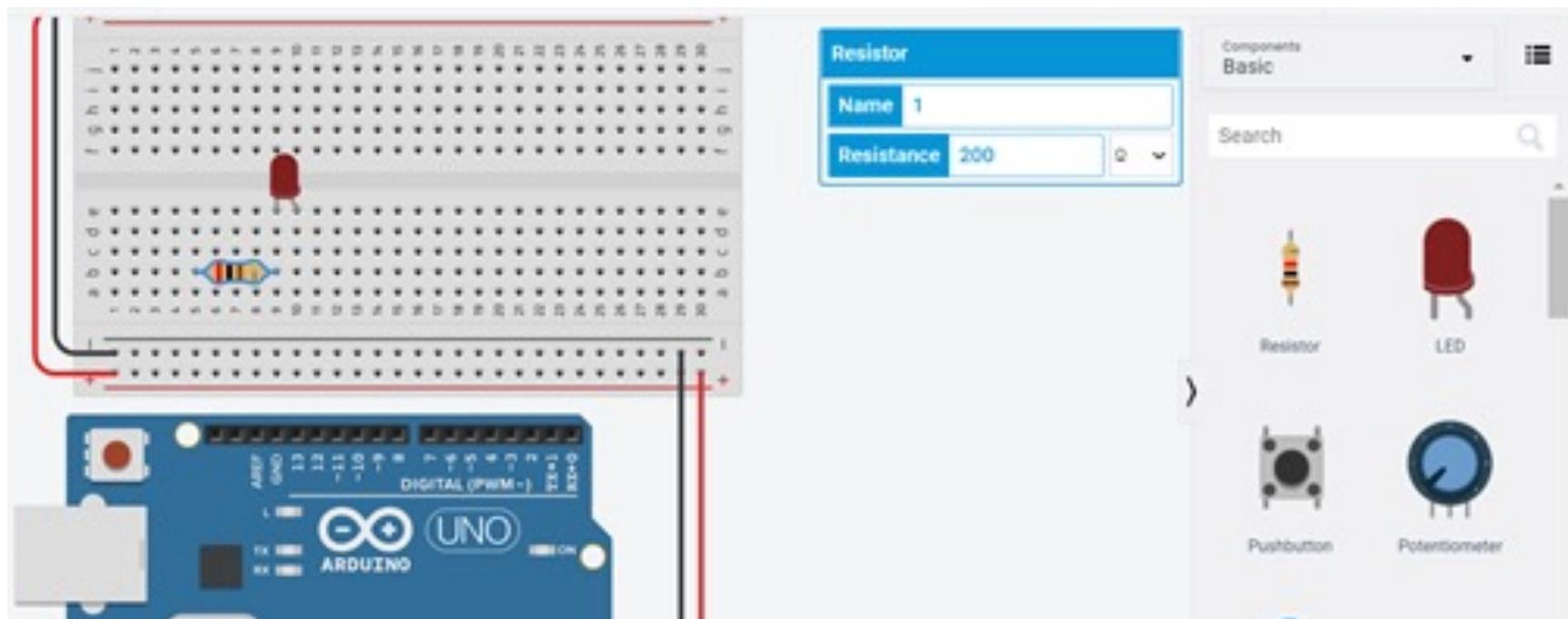
*Notice the Cathode is on Row 9, Column e. And the Anode is on Row 10, Column e.



Setting up our Workplace

1. Under Basic/Components: click on the Resistor.
2. Click and drag on the component, place it on the workplace and let go.

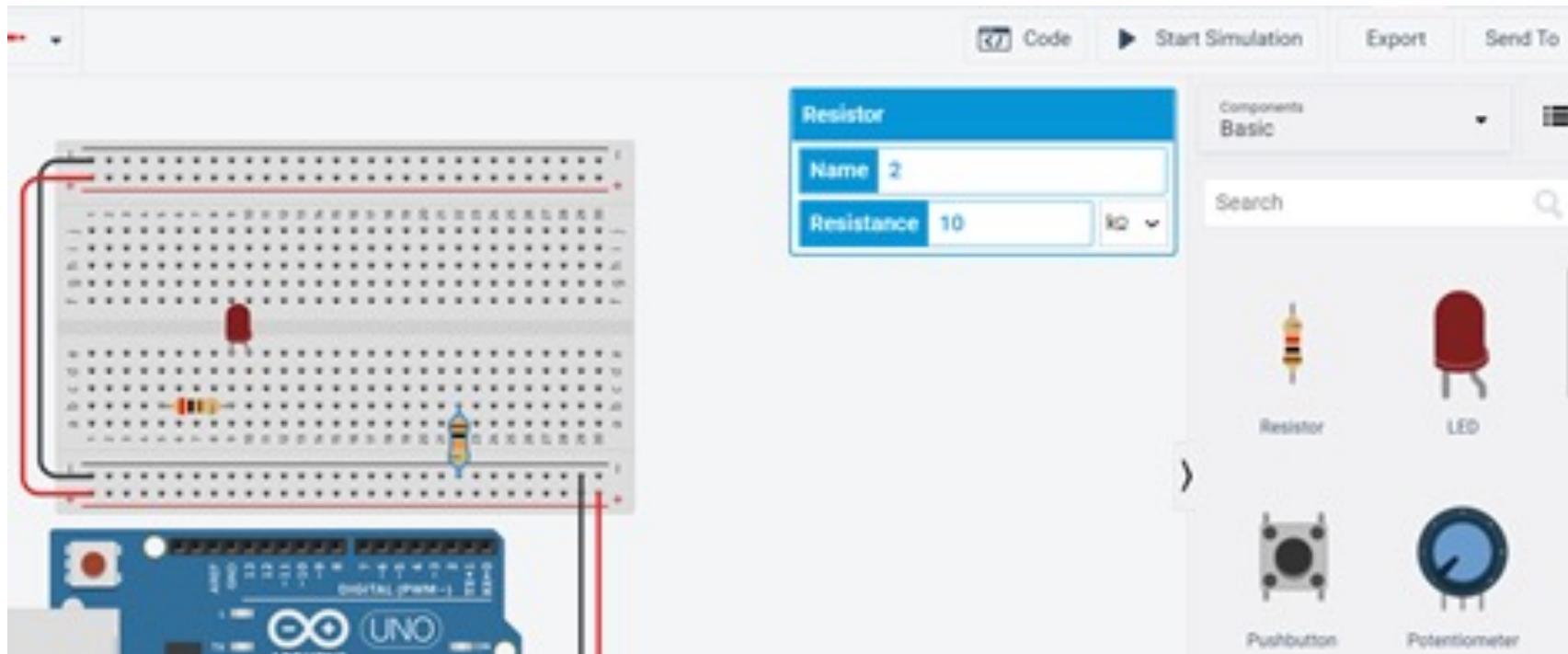
- Notice one end is on Row 5, Column b.
And the other end is on Row 9, Column b.
- You can rotate the resistor by press "R" on your keyboard.
- Set your resistor to 200 Ohms



Setting up our Workplace

1. Under Basic/Components: click on the Resistor.
2. Click and drag on the component, place it on the workplace and let go.

-Notice one end is on Row 22, Column b.
And the other end is aligned to Row 22 on the Ground Bus.
-You can rotate the resistor by press "R" on your keyboard.
-Set your resistor to 10 kOhms

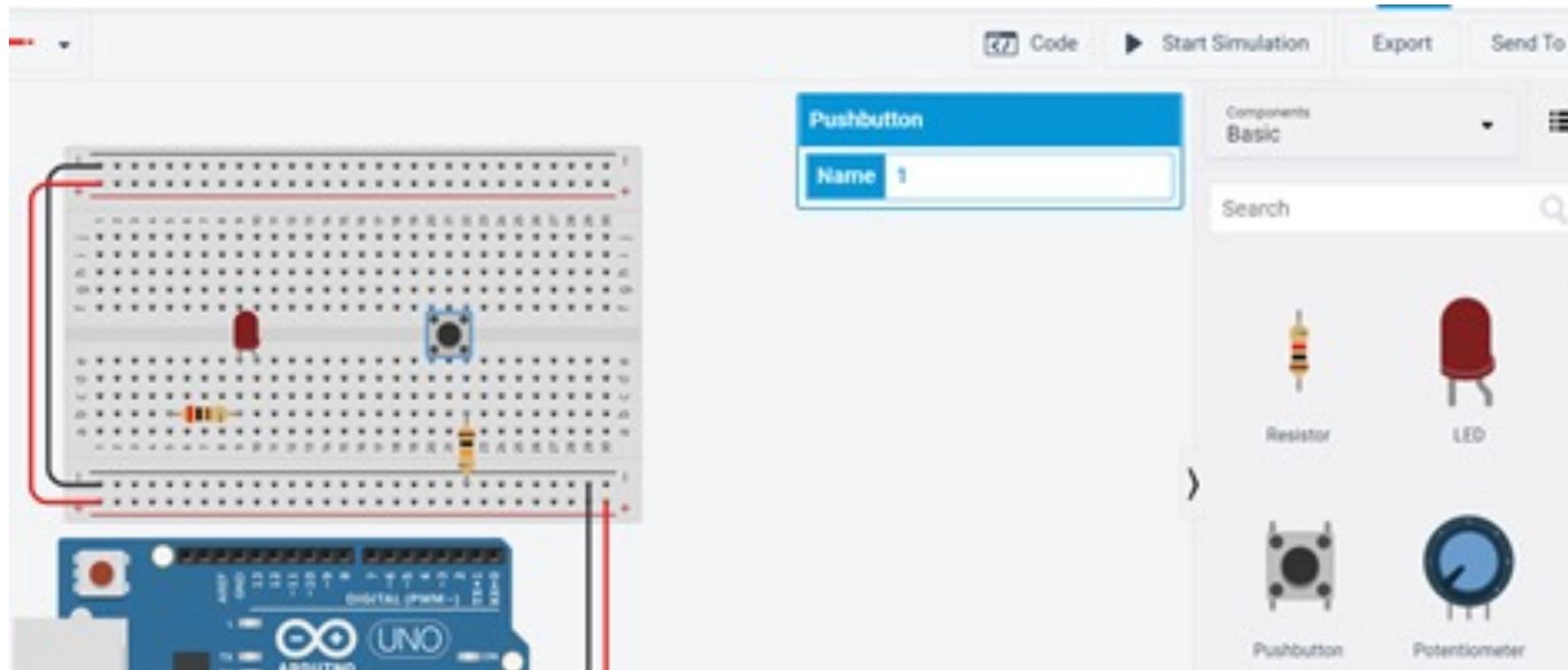


Setting up our Workplace

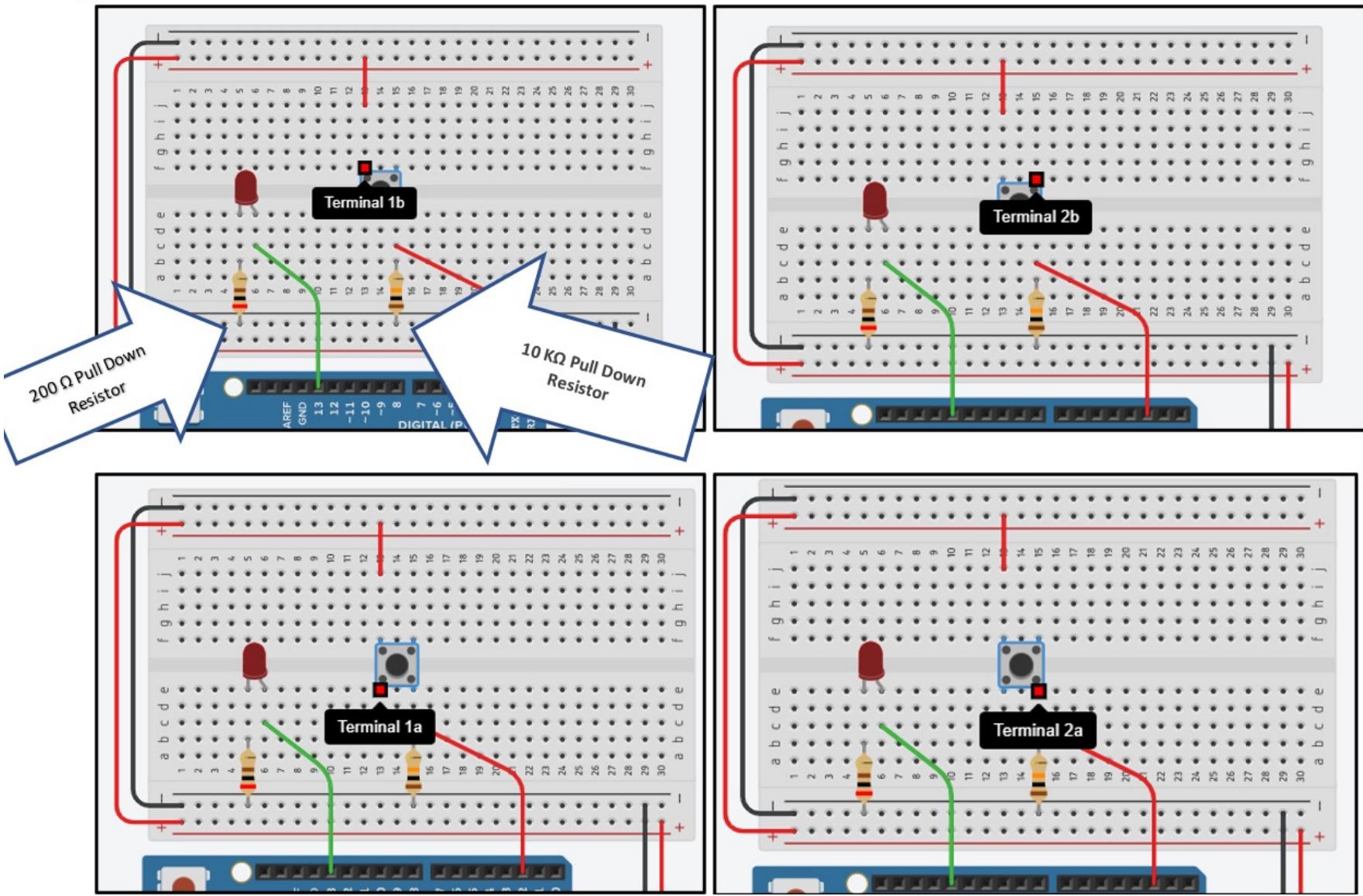
1. Under Basic/Components: click on the Pushbutton.
2. Click and drag on the component, place it on the workplace and let go.

-Terminals for the Pushbutton: Row 20, Column f (Terminal 1b) . Row 22, Column f (Terminal 2b). Row 20, Column e (Terminal 1a). Row 22, Column e (Terminal 2a)).

-You can rotate the resistor by press "R" on your keyboard.

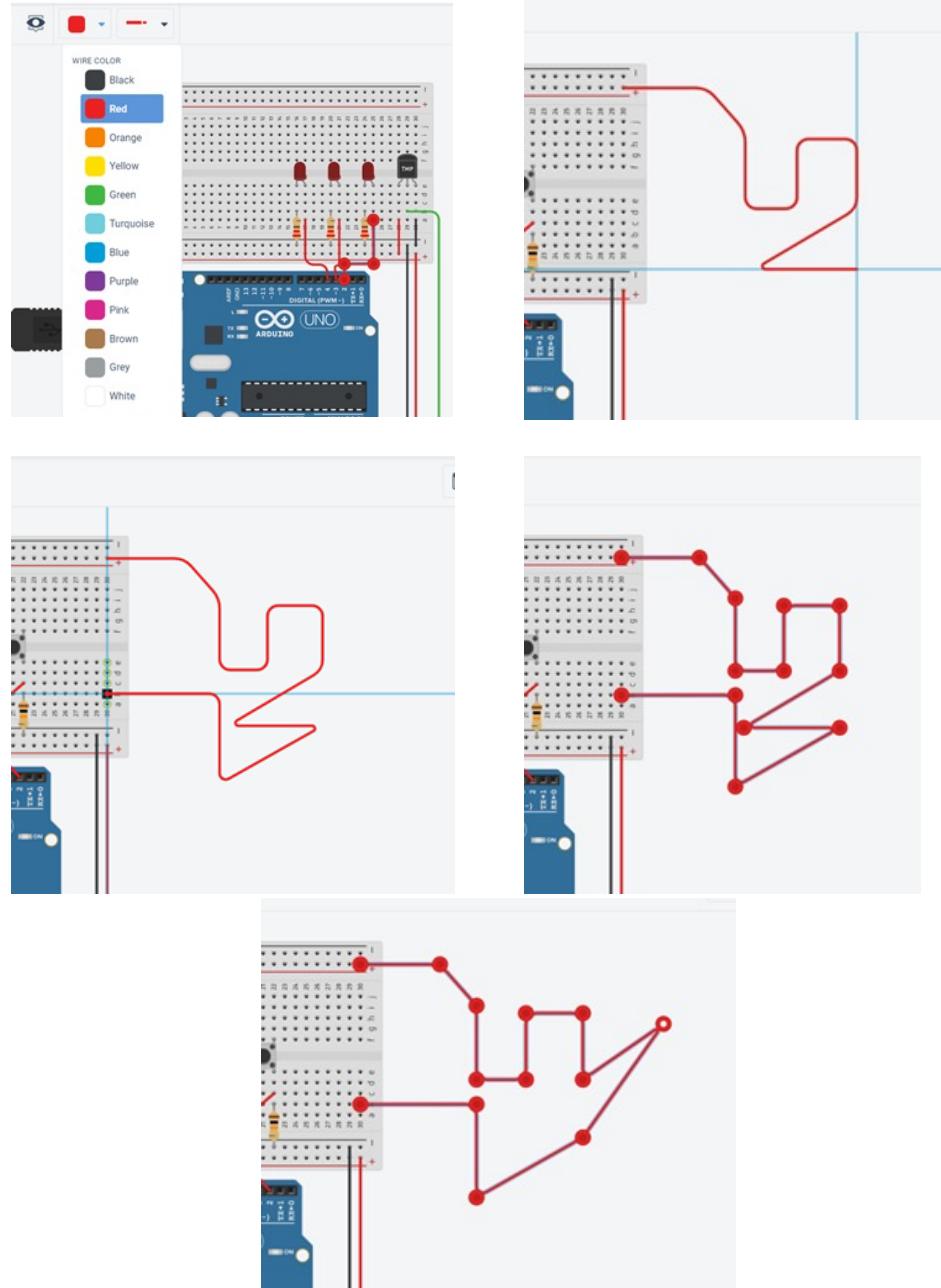


Switch Terminals



Wiring Orientation

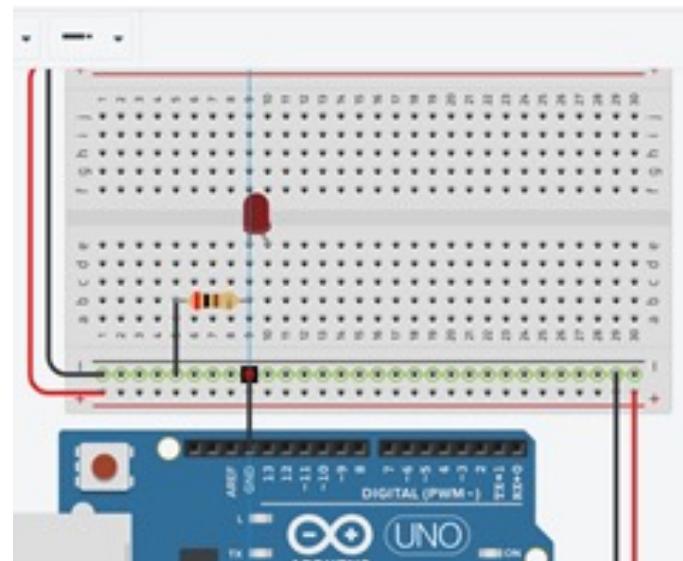
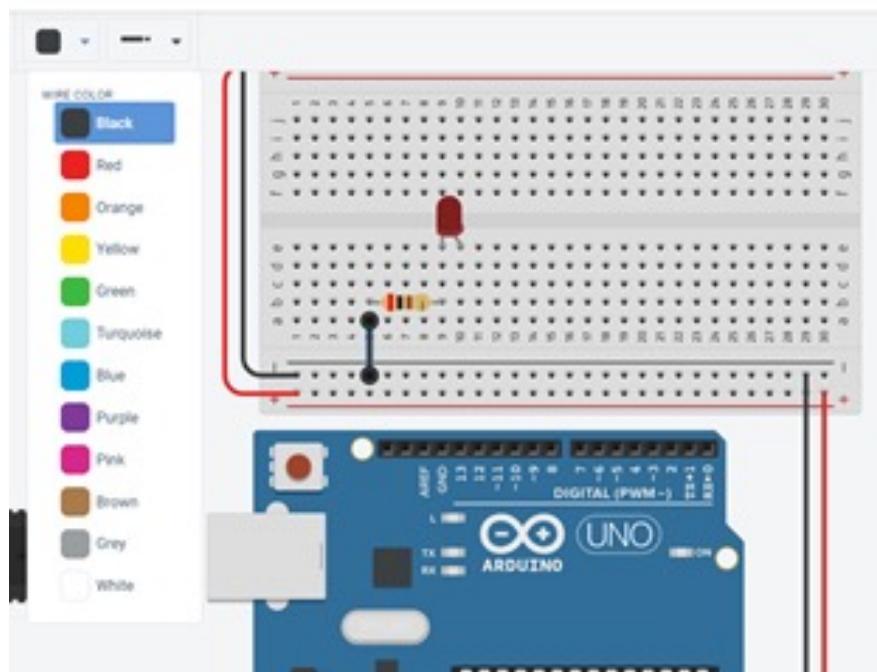
1. Choose a color for the wire.
Commonly, your ground will be black, and power will be red.
2. Next choose a start point and click once, you must choose a port or a location on the bread board. You will now have a wire that can be stretched almost at any angle.
3. In order to make your wire "curve" at different points, you can right click, as long as you do not click when hovering over a port or else it'll assumingly connect to that port.
4. You add as many turning points as you want, and they will appear after you connect to a port.
5. After which you can stretch the wire through its' turning points.



Wiring a LED with Arduino

Set up

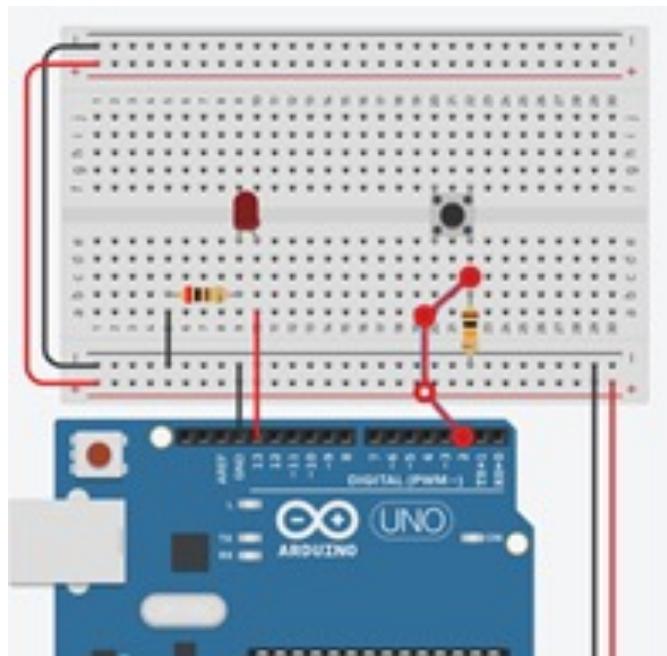
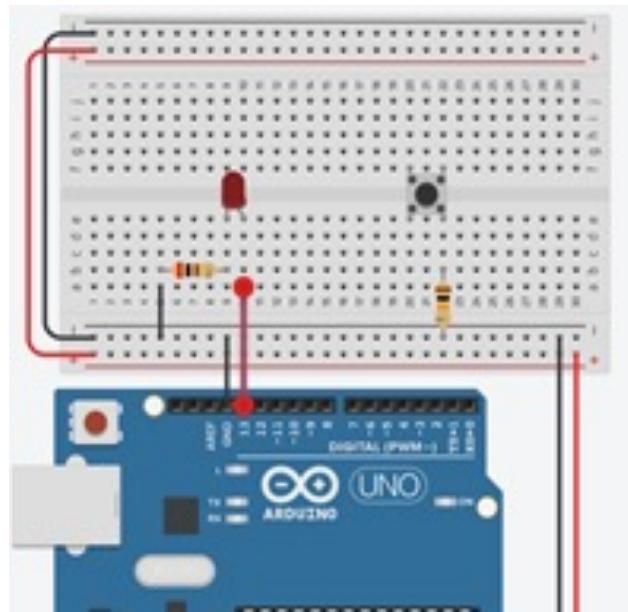
1. Select the color black for the following wires as seen in the image.
2. One end starts at the Ground bus (row 5). The other end connects to Row 5, Column a.
3. The second wire connects from the GND port on the Arduino to the Ground bus, aligned with row 9. This provide a return path to ground while using a pull down resistor on the Cathode side of the LED.



Wiring a LED with Arduino

Set up

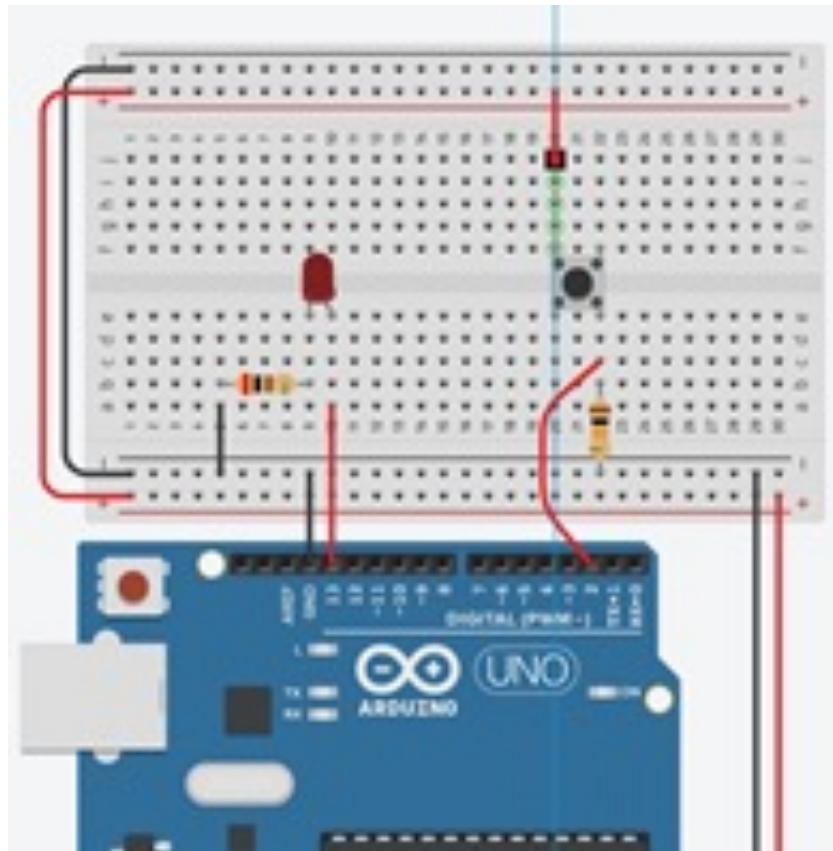
1. Select the color red for the following wires as seen in the image.
2. One end starts at Port 13 on the Arduino. The other end connects to Row 10, Column a.
3. The second wire connects from Port 2 on the Arduino to Row 22, Column c.



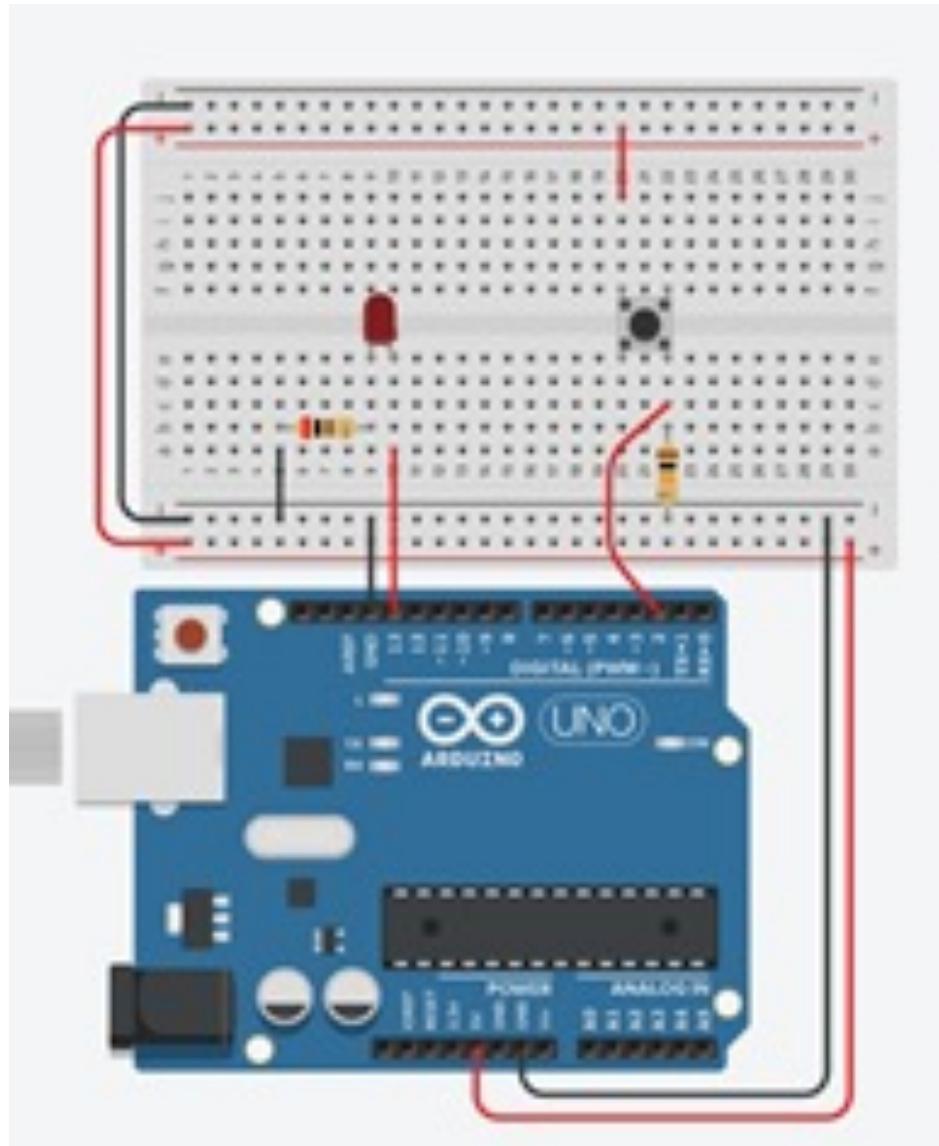
Wiring a LED with Arduino

Set up

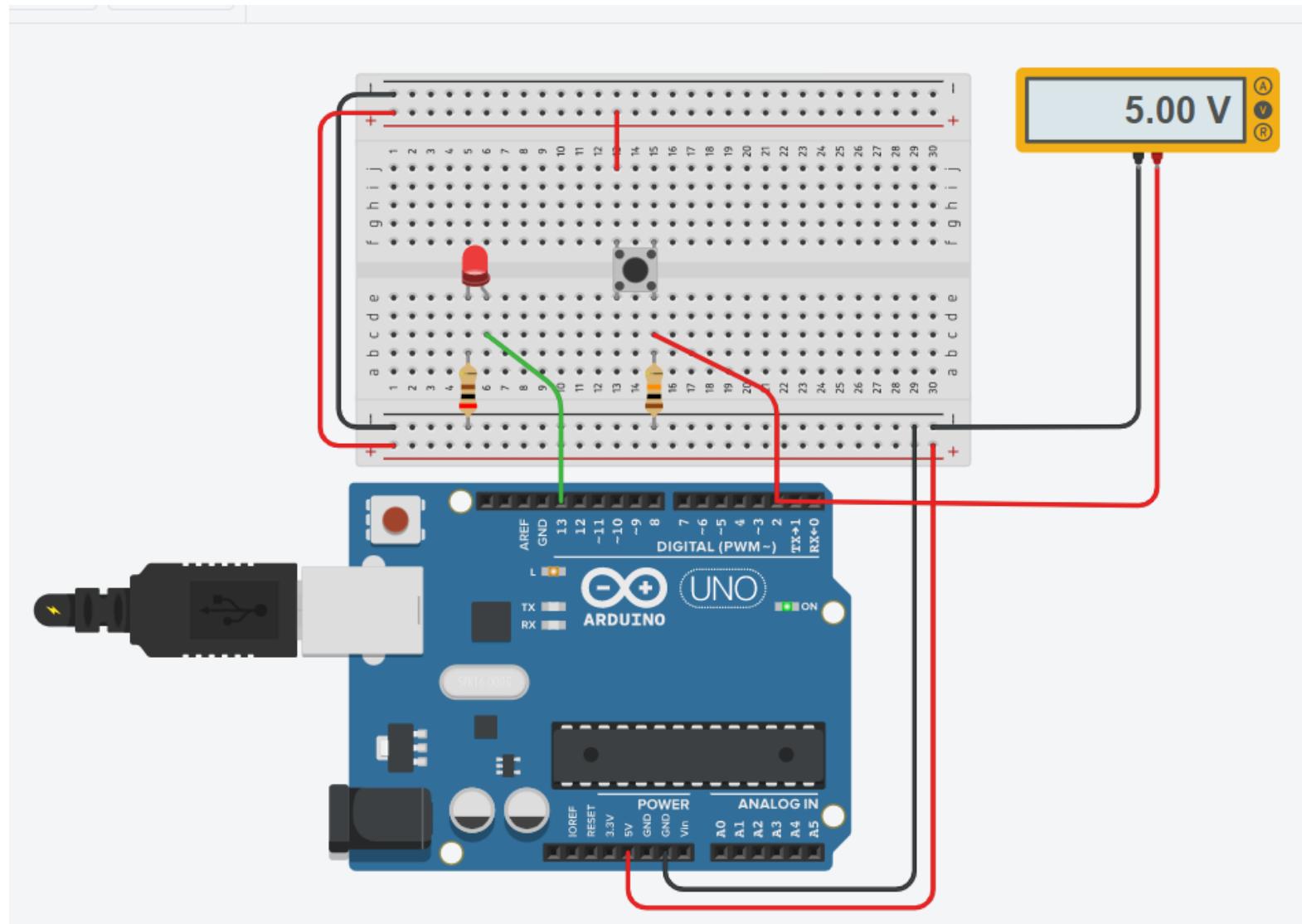
1. One end starts at Positive Bus aligned with Row 20. The other end connects to Row 20, Column j.



Final Product



Analyzing your Switch



C++ First Half of Code

```
//C++
//variable (buttonState), is set to digital value
zero,
 //to ensure the state od the button begins at
zero, OFF.
int buttonState = 0;//variables

void setup()//Pin setup
{
 //pinMode allows us to set up individual pins to
input
 //or output. We set Pin 2 as input. And the built
in
 //LED as output.
pinMode(2, INPUT);
pinMode(LED_BUILTIN, OUTPUT);
}
```

The screenshot shows the Arduino IDE interface. The top bar indicates the board is set to "1 (Arduino Uno R3)". The main area displays the C++ code for a push button example. The code includes comments explaining the purpose of each section: setting up a variable for the button state, defining the setup() function to set pin 2 as an input and the built-in LED as an output, and the loop() function which reads the button state, checks if it's HIGH, and then toggles the built-in LED. A "Serial Monitor" tab is visible at the bottom.

```
1 //variable (buttonState), is set to digital value zero,
2 //to ensure the state od the button begins at zero, OFF.
3 int buttonState = 0;//variables
4
5 void setup()//Pin setup
6 {
7 //pinMode allows us to set up individual pins to input
8 //or output. We set Pin 2 as input. And the built in
9 //LED as output.
10 pinMode(2, INPUT);
11 pinMode(LED_BUILTIN, OUTPUT);
12 }
13
14 void loop()//Loop setup
15 {
16 // read the state of the push button and store the
17 //information gathered in the variable buttonstate.
18 buttonState = digitalRead(2);
19 // check if pushbutton is pressed. if it is, the
20 // button state is HIGH
21 if (buttonState == HIGH) {
22   digitalWrite(LED_BUILTIN, HIGH);//Write HIGH to built
23   //in LED.
24   // If pushbutton is not HIGH.
25 } else {
26   digitalWrite(LED_BUILTIN, LOW);//Write LOW to the built
27   //in LED.
28 }
29 delay(10); // Delay a little bit (0.01 seconds) to improve
30 //simulation performance.
```

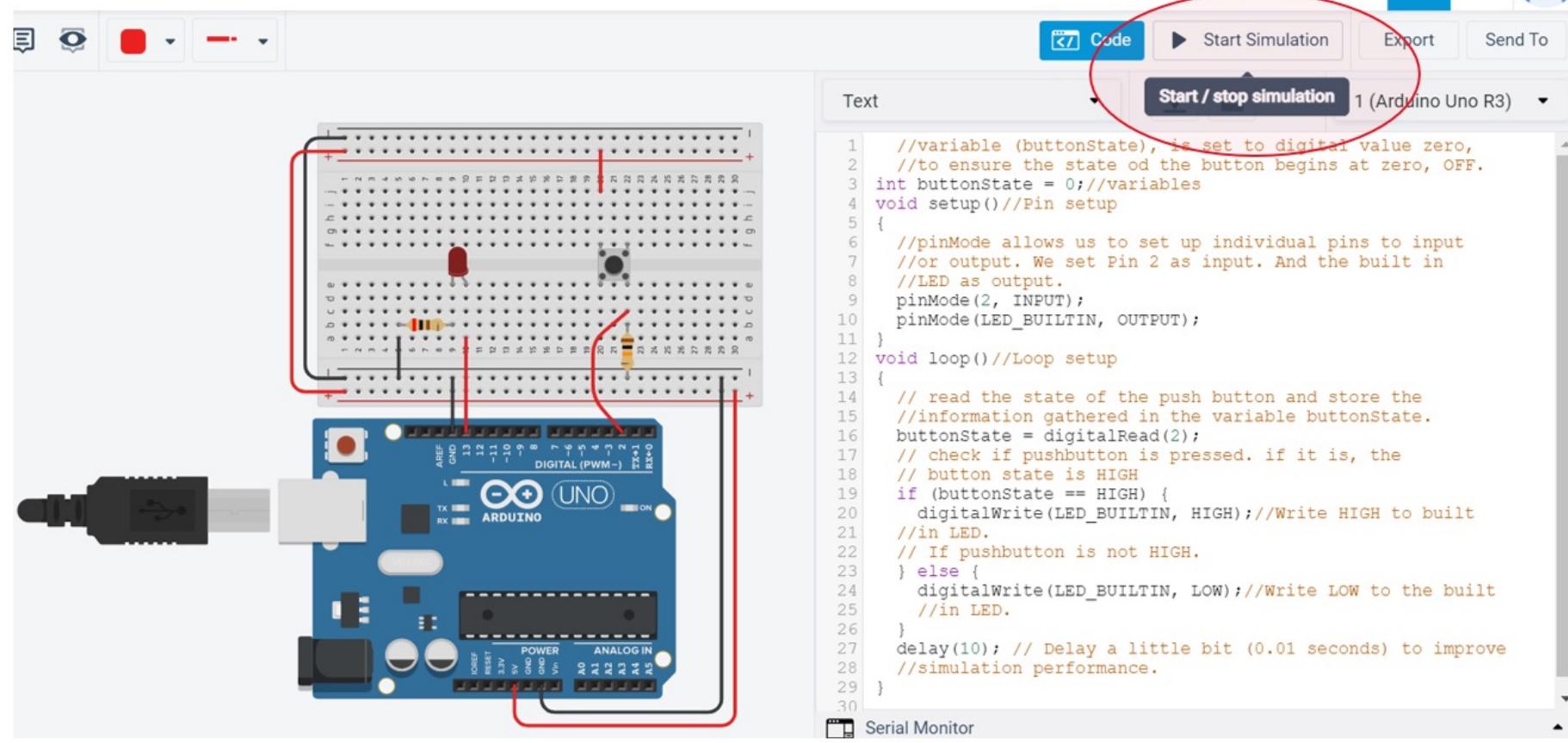
C++ Second Half of Code

```
void loop()//Loop setup
{
    // read the state of the push button and
    store the
    //information gathered in the
    variable buttonState.
    buttonState = digitalRead(2);
    // check if pushbutton is pressed. if it is, the
    // button state is HIGH
    if (buttonState == HIGH) {
        digitalWrite(LED_BUILTIN,
HIGH);//Write HIGH to built
        //in LED.
        // If pushbutton is not HIGH.
    } else {
        digitalWrite(LED_BUILTIN, LOW);//Write
        LOW to the built
        //in LED.
    }
    delay(10); // Delay a little bit (0.01 seconds)
    to improve
    //simulation performance.
}
```

The screenshot shows the Arduino IDE interface. The top bar indicates the board is set to "1 (Arduino Uno R3)". The main window displays the C++ code for a sketch. Below the code area, there is a "Serial Monitor" tab.

```
1 //variable (buttonState), is set to digital value zero,
2 //to ensure the state od the button begins at zero, OFF.
3 int buttonState = 0;//variables
4
5 void setup()//Pin setup
{
7 //pinMode allows us to set up individual pins to input
8 //or output. We set Pin 2 as input. And the built in
9 //LED as output.
10 pinMode(2, INPUT);
11 pinMode(LED_BUILTIN, OUTPUT);
}
13
14 void loop()//Loop setup
{
16 // read the state of the push button and store the
17 //information gathered in the variable buttonstate.
18 buttonState = digitalRead(2);
19 // check if pushbutton is pressed. if it is, the
20 // button state is HIGH
21 if (buttonState == HIGH) {
22     digitalWrite(LED_BUILTIN, HIGH);//Write HIGH to built
23     //in LED.
24     // If pushbutton is not HIGH.
25 } else {
26     digitalWrite(LED_BUILTIN, LOW);//Write LOW to the built
27     //in LED.
28 }
29 delay(10); // Delay a little bit (0.01 seconds) to improve
30 //simulation performance.
```

Begin Simulation



The screenshot shows a simulation environment for an Arduino Uno R3. On the left, the physical hardware is shown with a USB cable connected. The central part is a breadboard with various components: a red LED, a push button, and two resistors. Wires connect these components to the Arduino pins. A red box highlights the breadboard area. On the right, the software interface includes a toolbar with icons for file, eye, and print. Below the toolbar is a menu bar with 'Code', 'Start Simulation', 'Export', and 'Send To'. A red circle highlights the 'Start / stop simulation' button. The main window has tabs for 'Text' and 'Serial Monitor'. The 'Text' tab contains the following code:

```
1 //variable (buttonState), is set to digital value zero,
2 //to ensure the state od the button begins at zero, OFF.
3 int buttonState = 0;//variables
4 void setup()//Pin setup
5 {
6     //pinMode allows us to set up individual pins to input
7     //or output. We set Pin 2 as input. And the built in
8     //LED as output.
9     pinMode(2, INPUT);
10    pinMode(LED_BUILTIN, OUTPUT);
11 }
12 void loop()//Loop setup
13 {
14     // read the state of the push button and store the
15     //information gathered in the variable buttonState.
16     buttonState = digitalRead(2);
17     // check if pushbutton is pressed. if it is, the
18     // button state is HIGH
19     if (buttonState == HIGH) {
20         digitalWrite(LED_BUILTIN, HIGH);//Write HIGH to built
21         //in LED.
22     // If pushbutton is not HIGH.
23     } else {
24         digitalWrite(LED_BUILTIN, LOW);//Write LOW to the built
25         //in LED.
26     }
27     delay(10); // Delay a little bit (0.01 seconds) to improve
28     //simulation performance.
29 }
30 }
```

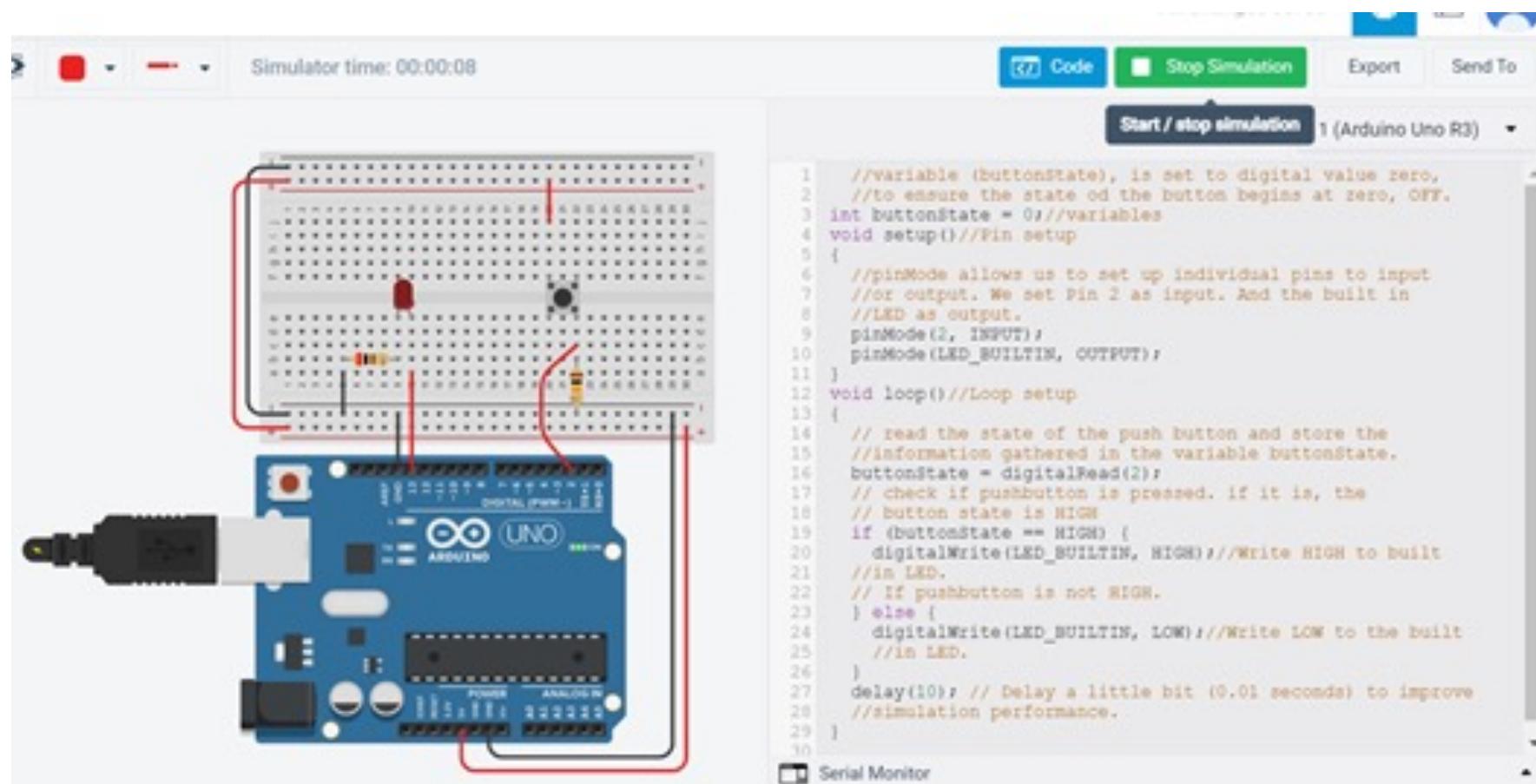
Simulation and Review

- Click on the Start Simulation to review your results. Your push button should turn on the LED, and turn off when you let go.
- Can you figure out a way to swap the logic in your circuit, so that the LED is lit by default and goes dark when the button is pressed?

Stop Simulation

Simulator time: 00:00:08

Code Stop Simulation Export Send To Start / stop simulation 1 (Arduino Uno R3)



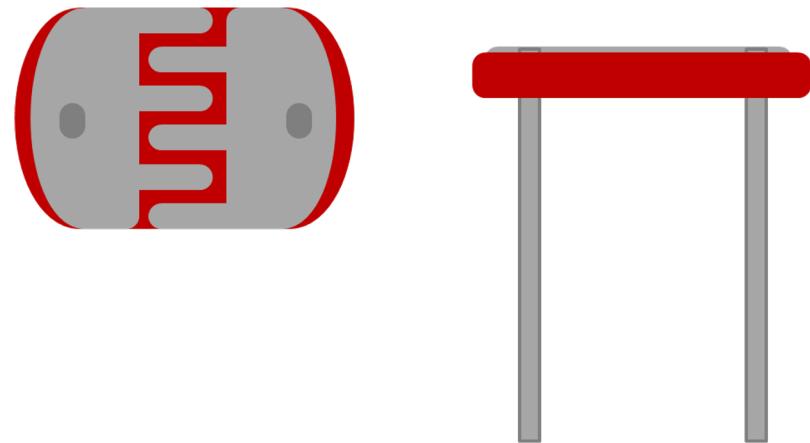
```
//variable (buttonstate), is set to digital value zero,  
//to ensure the state of the button begins at zero, OFF.  
int buttonState = 0; //variables  
void setup() //Pin setup  
{  
    //pinMode allows us to set up individual pins to input  
    //or output. We set Pin 2 as input. And the built in  
    //LED as output.  
    pinMode(2, INPUT);  
    pinMode(LED_BUILTIN, OUTPUT);  
}  
void loop() //Loop setup  
{  
    // read the state of the push button and store the  
    //information gathered in the variable buttonState.  
    buttonState = digitalRead(2);  
    // check if pushbutton is pressed. If it is, the  
    //button state is HIGH  
    if (buttonState == HIGH) {  
        digitalWrite(LED_BUILTIN, HIGH); //Write HIGH to built  
        //in LED.  
        // If pushbutton is not HIGH.  
    } else {  
        digitalWrite(LED_BUILTIN, LOW); //Write LOW to the built  
        //in LED.  
    }  
    delay(10); // Delay a little bit (0.01 seconds) to improve  
    //simulation performance.  
}
```

Virtual Activity - 3

Photoresistor

Photoresistor (LDR)

A photoresistor or **LDR** (light dependent resistor) is a resistor whose resistance depends on light intensity



An LDR can be used as a simple, **analog sensor**

The orientation of an LDR does not matter

Starting a new circuit

Dashboard | Tinkercad [x](#) [+](#) tinkercad.com/dashboard?&type=tinkercad&collection=designs [🔍](#) [☆](#) [ABP](#)

**TIN
KER
CAD** AUTODESK® TINKERCAD®

Log in to www.tinkercad.com

Tinkercad Lesson Plans
Tinkercad lesson plans are ready to use online or in the classroom. Discover curriculum developed in partnership with teachers. [Learn more](#)

My recent designs

Create new design

3D Designs **Circuits** **Codeblocks** **Lessons**

Click "Circuits."

Ben Finio

Search designs...

Cool Snicket 5 months ago Private

Glorious Kieran-Sango 8 months ago Private

Stunning Luulia 5 years ago Private

Powerful Amberis-Wolt 5 years ago Private

Project 1

Create project

Tweets Follow

Tinkercad Retweeted

Allison Debellis @adebellis8

61

Starting a new circuit



Gallery Blog Learn Teach Q

Ben Finio

Search designs...

3D Designs

Circuits

Codeblocks

NEW

Lessons

Your Classes

Projects

Project 1

Create project

Tweets

Follow

Tinkercad Retweeted

Allison Debellis

@adebellis8

Tinkercad Lesson Plans

Tinkercad lesson plans are ready to use online or in the classroom. Discover curriculum developed in partnership with teachers. [Learn more](#)

Circuits

Create new Circuit

Click "Create new Circuit."

Example Arduino Light Show (With R...

5 hours ago
Private

My First Arduino Circuit

7 hours ago
Public

Brave Wolt

9 hours ago
Private

Spectacular Stantia-Jaiks

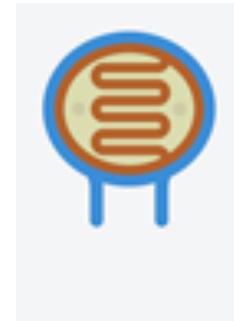
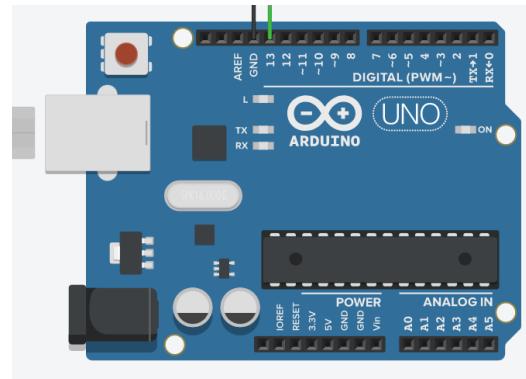
5 days ago
Private

Components

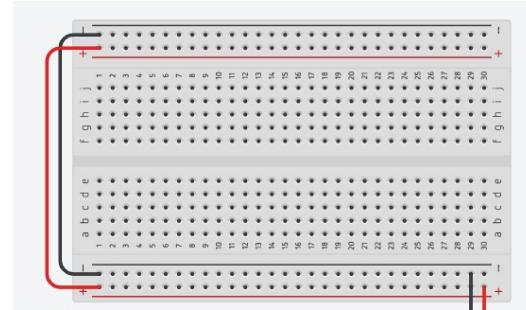
We will introduce how to blink the on-board LED and how to blink an external LED using an LDR.

Components

- Arduino
- LDR / photoresistor
- Two resistors
- Breadboard
- LED



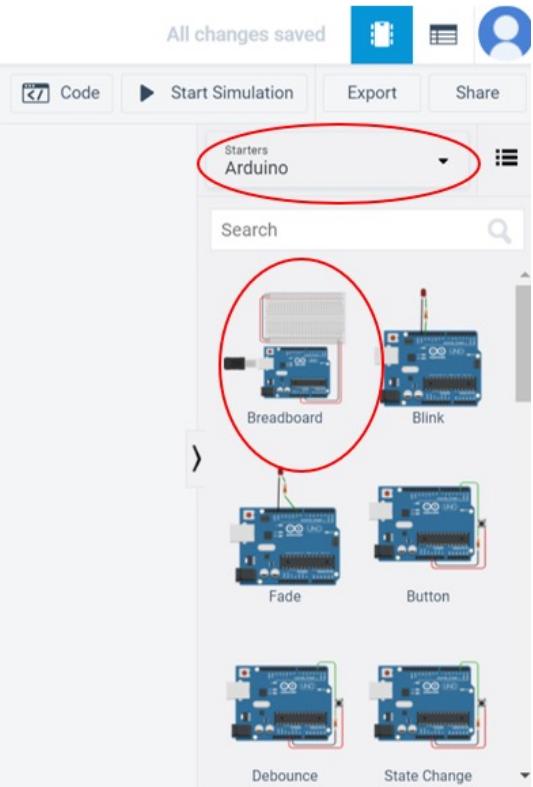
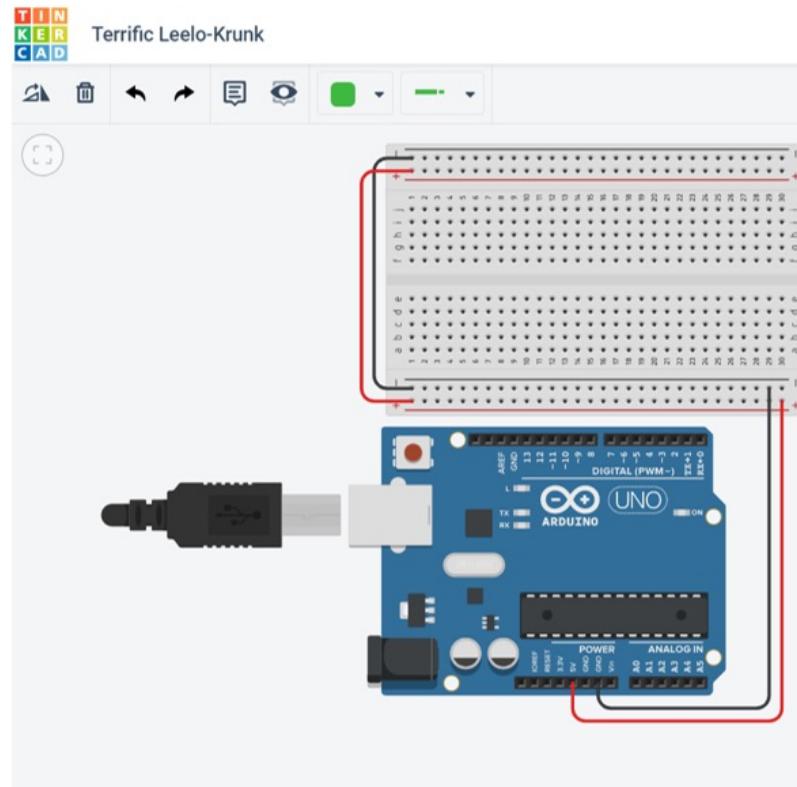
Resistor



LED

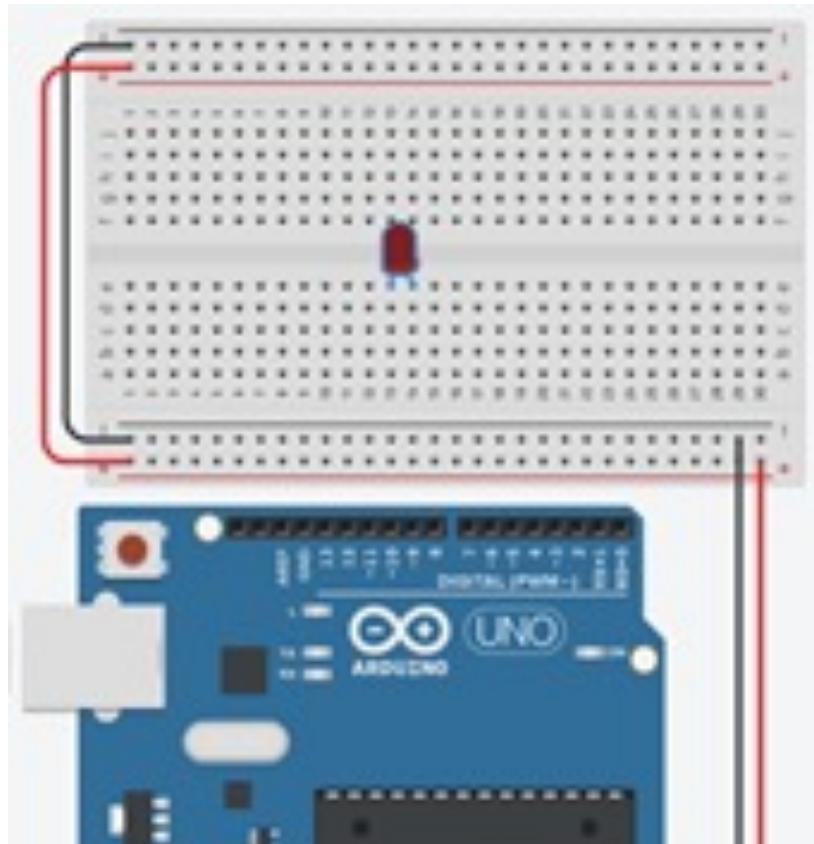
Setting up our Workplace

1. Under Starters/Components: click Arduino.
2. Click and drag on the first component, place it on the workplace and let go.
3. Repeat step two for the following components: two resistors, one LDR, and one LED.



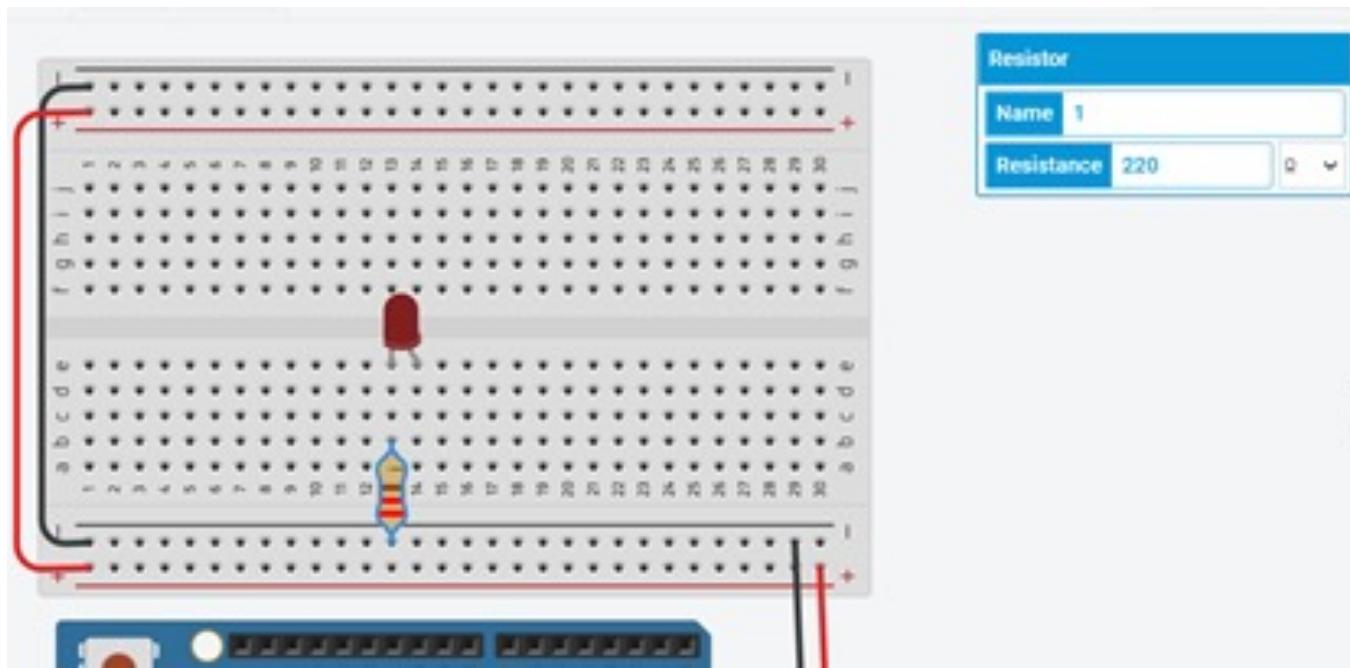
Setting up our Workplace

1. Under Starters/Components: click LED.
2. Click and drag on the first component, place it on the workplace and let go.
3. Cathode will be on Row 13, Column e. Anode will be on Row 14, Column e.



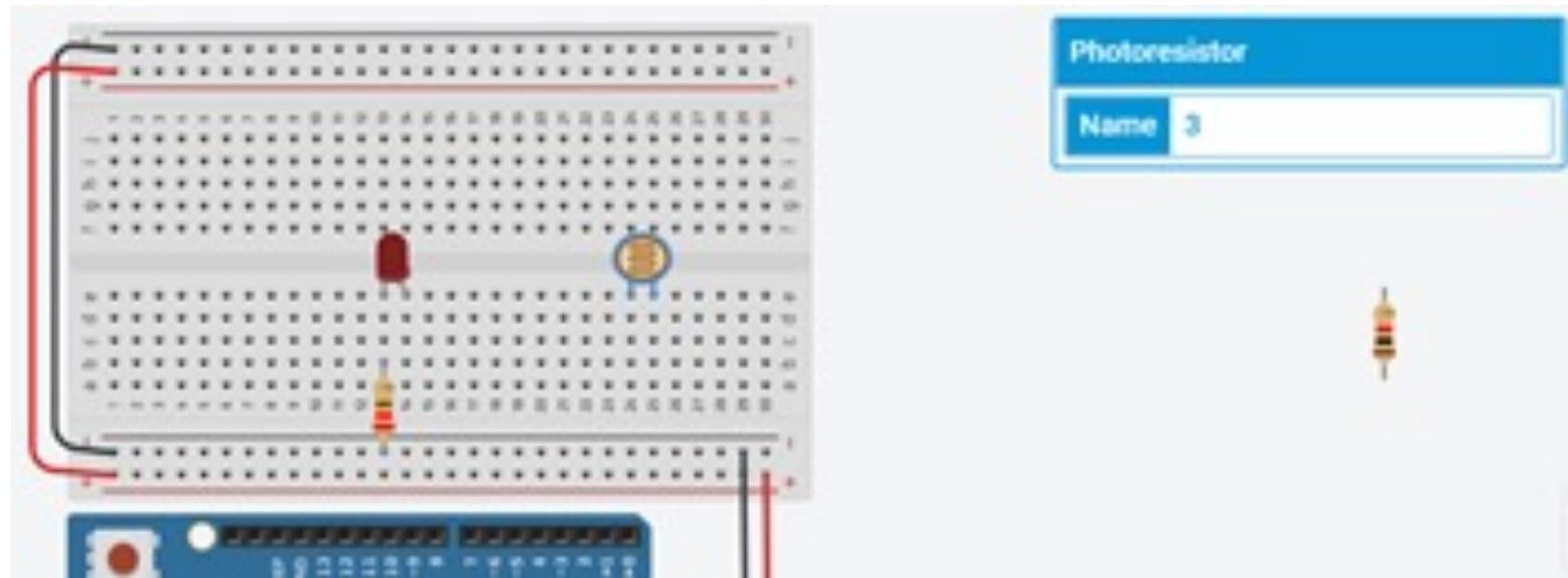
Setting up our Workplace

1. Under Starters/Components: click Resistor.
2. Click and drag on the first component, place it on the workplace and let go.
3. One end is on Row 13, Column b. The other end is aligned to Row 13 on the Ground Bus.
4. Set this Resistor to 220 Ohms



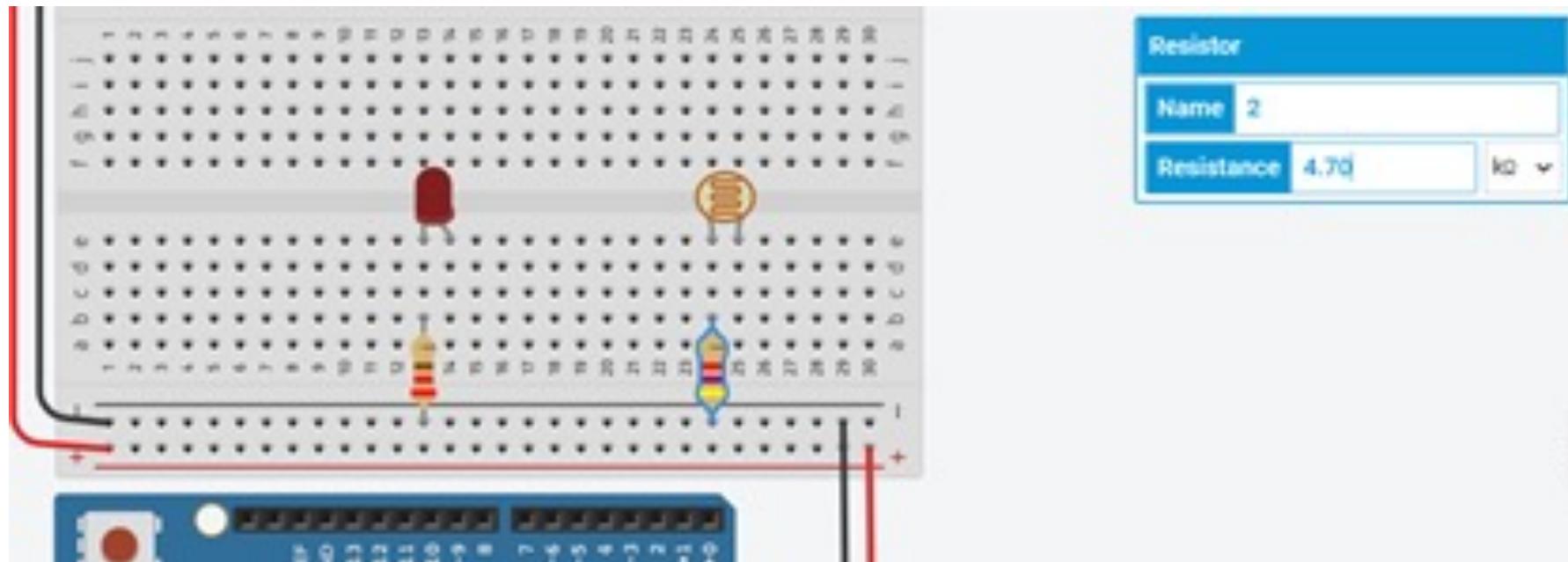
Setting up our Workplace

1. Under Starters/Components: click Photoresistor.
2. Click and drag on the component, place it on the workplace and let go.
3. One end is on Row 24, Column e. The other end is on Row 25, Column e.
4. You can rotate the resistor by press "R" on your keyboard.



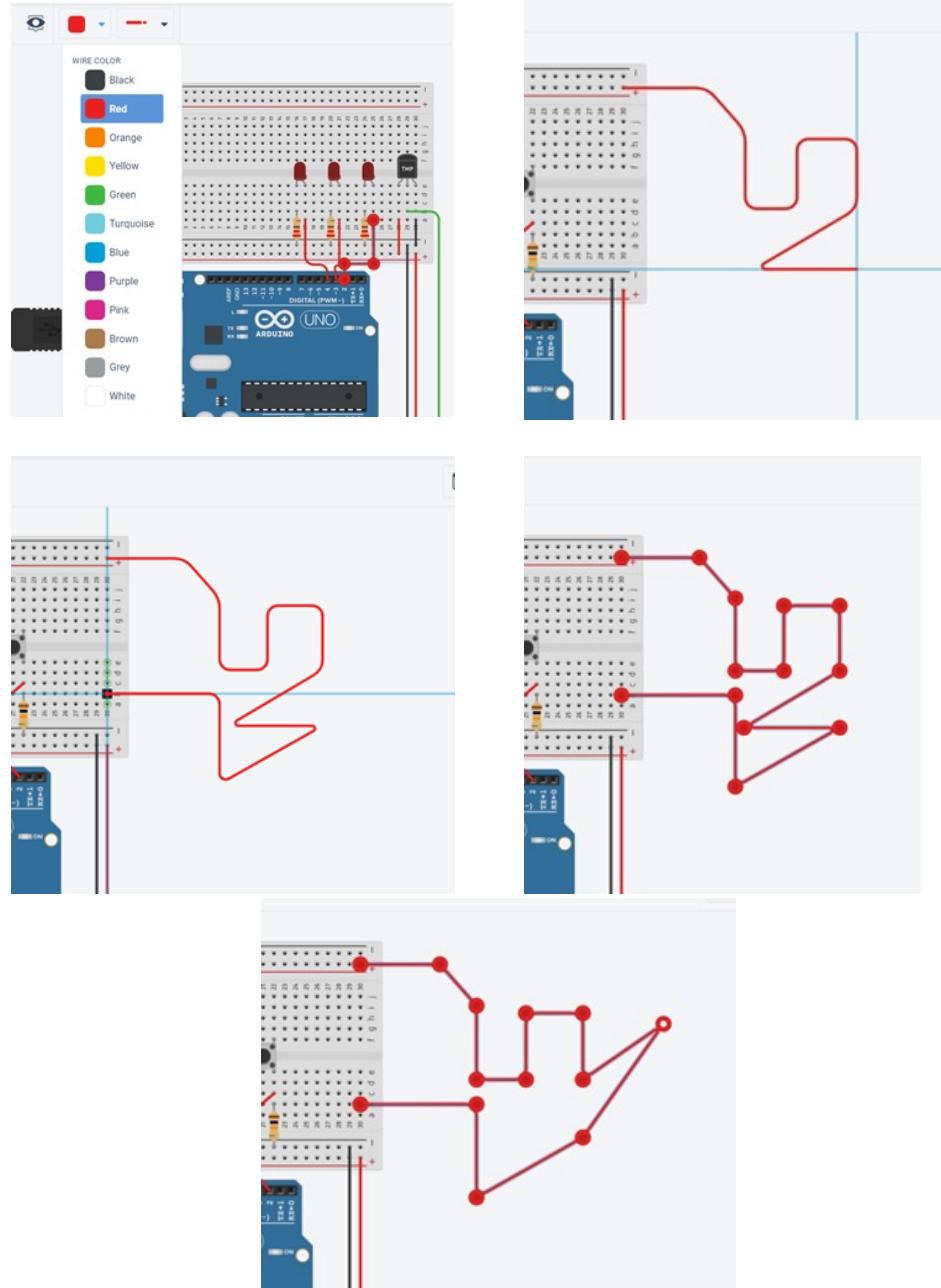
Setting up our Workplace

1. Under Starters/Components: click Resistor.
2. Click and drag on the first component, place it on the workplace and let go.
3. One end is on Row 24, Column b. The other end is aligned to Row 24 on the Ground Bus.
4. Set this Resistor to 4.70 kOhms



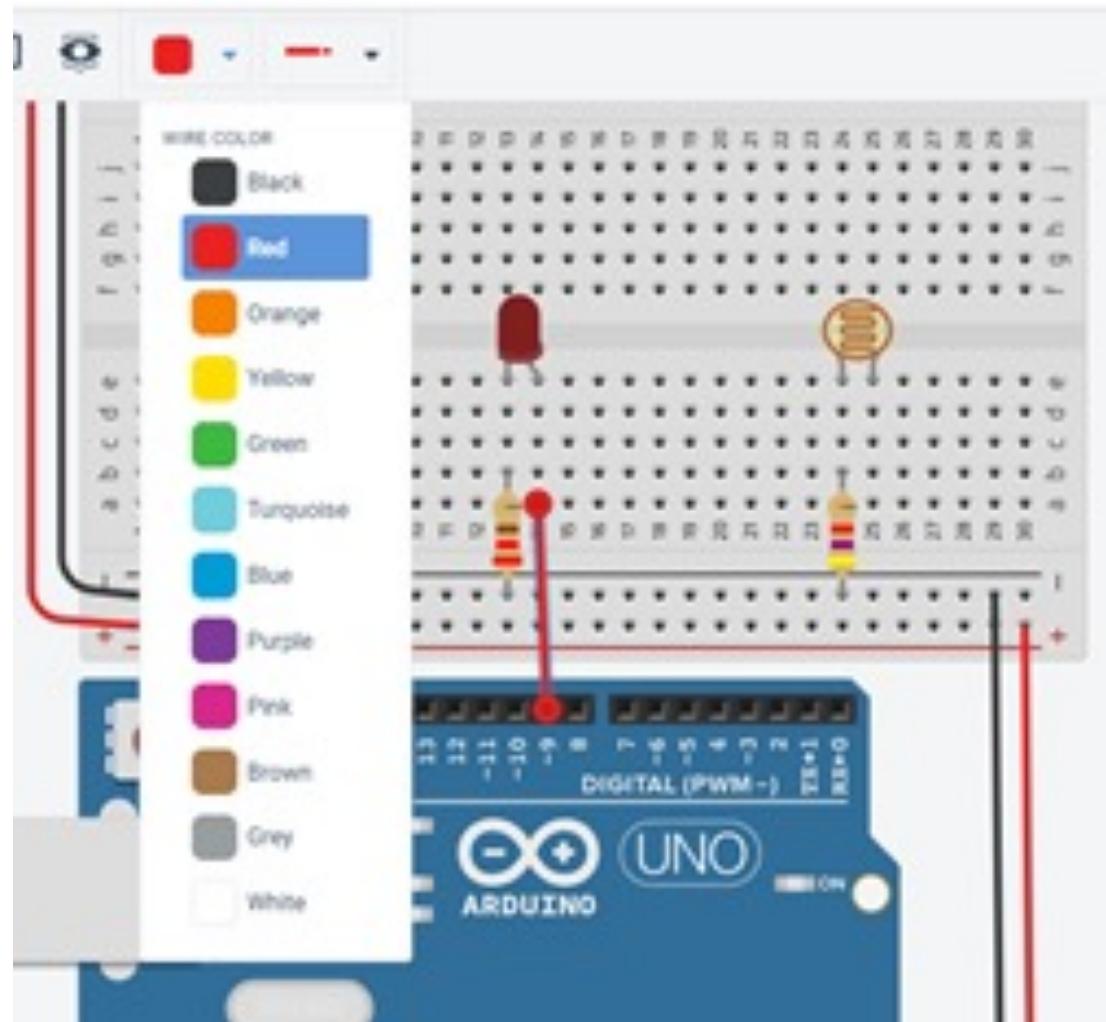
Wiring Orientation

1. Choose a color for the wire.
Commonly, your ground will be black, and power will be red.
2. Next choose a start point and click once, you must choose a port or a location on the bread board. You will now have a wire that can be stretched almost at any angle.
3. In order to make your wire "curve" at different points, you can right click, as long as you do not click when hovering over a port or else it'll assumingly connect to that port.
4. You add as many turning points as you want, and they will appear after you connect to a port.
5. After which you can stretch the wire through its' turning points.



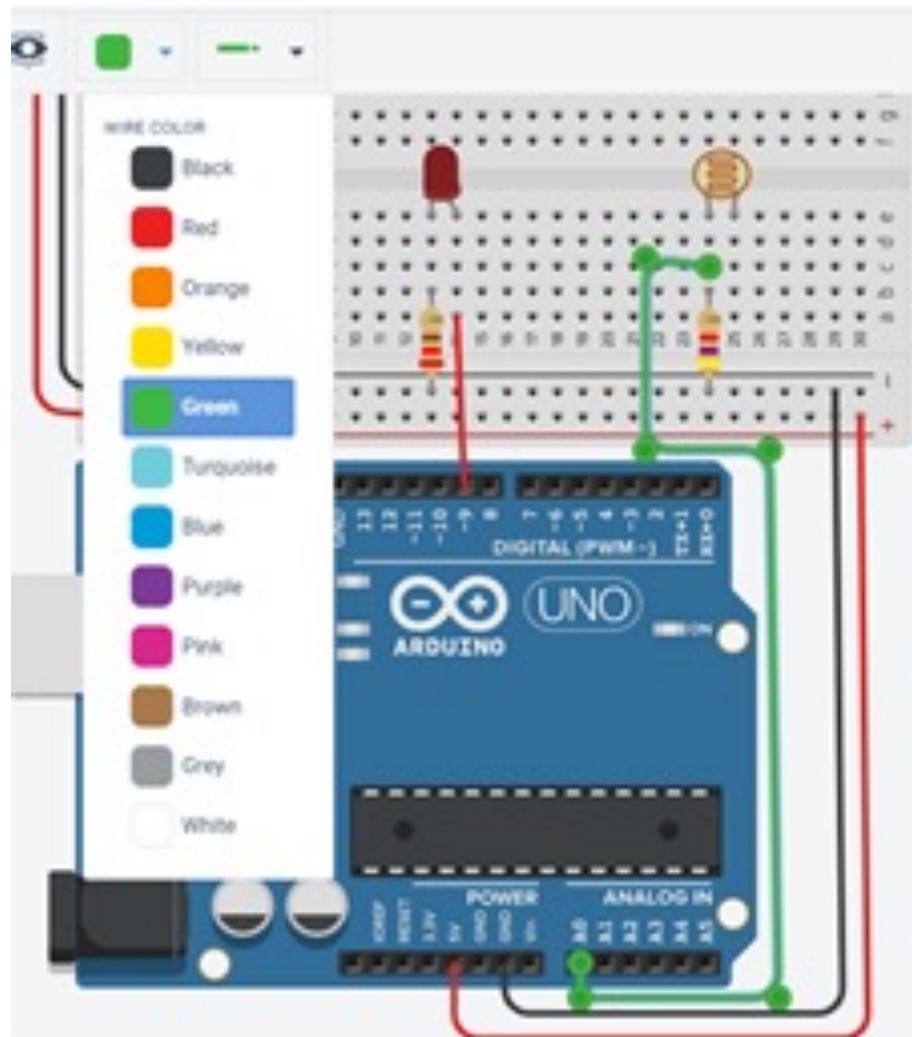
Wiring components

1. For this wire select the color red.
 2. One end will begin at the Arduino Port 9. The other end will connect to Row 14, Column a.



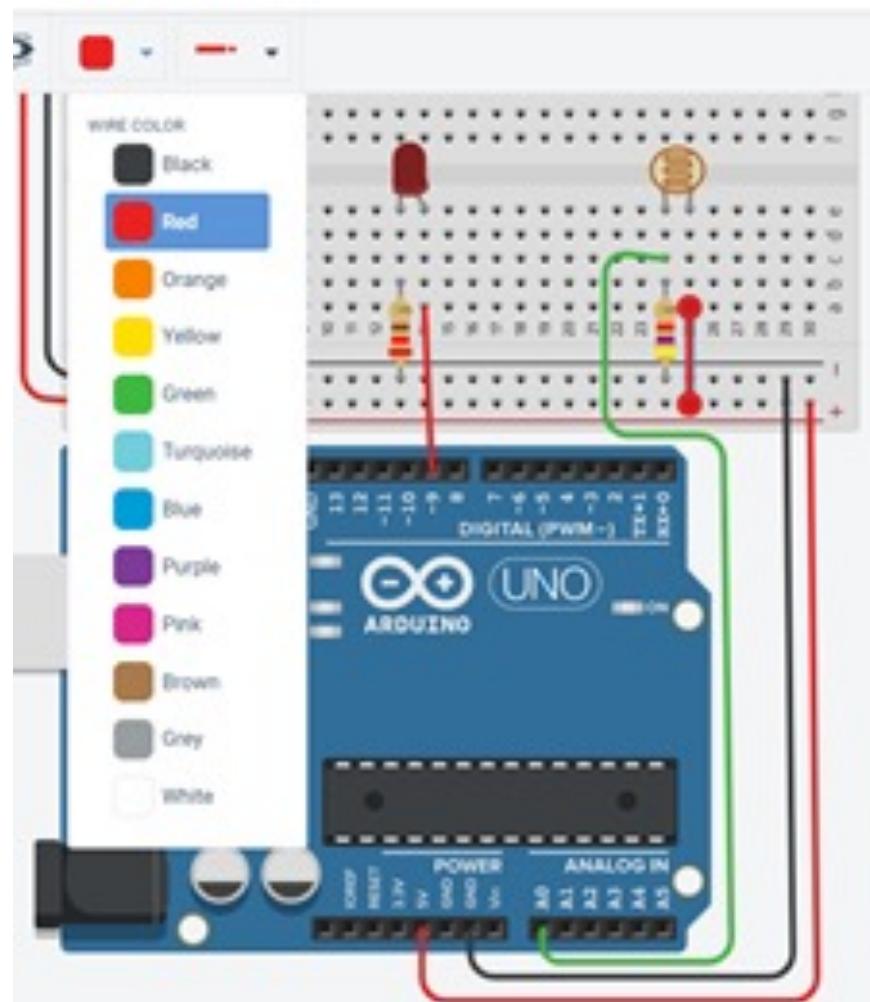
Wiring components

1. For this wire select the color green.
2. One end will begin at the Arduino Analog Port 0. The other end will connect to Row 24, Column c.

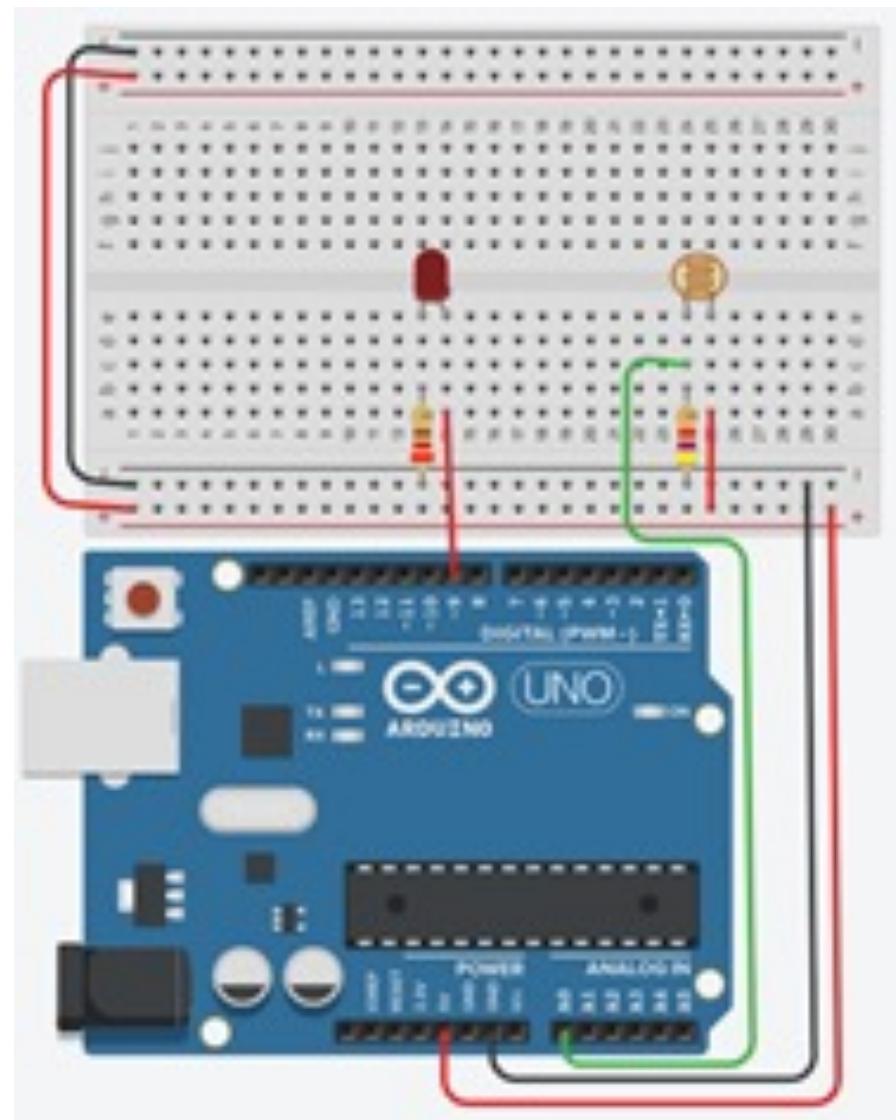


Wiring components

1. For this wire select the color red.
2. One end will begin at the positive bus. The other end will connect to Row 25, Column a.



Final Product

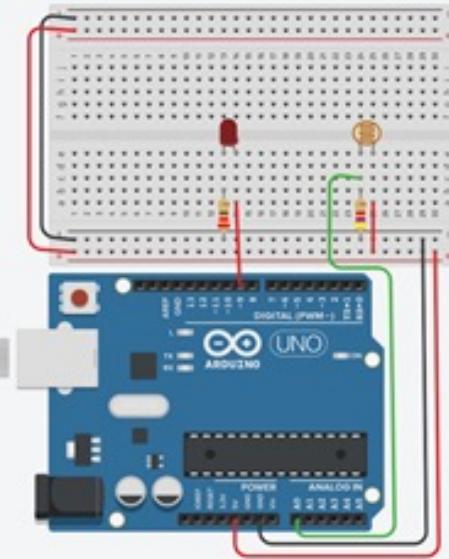


Analog input with Arduino

Code

→ Copy the following code

```
// C++ code  
//  
int sensorValue = 0;  
  
void setup()  
{  
    pinMode(A0, INPUT);  
    Serial.begin(9600);  
  
    pinMode(9, OUTPUT);  
}  
  
void loop()  
{  
    // read the value from the sensor  
    sensorValue = analogRead(A0);  
    // print the sensor reading so you know its range  
    Serial.println(sensorValue);  
    // map the sensor reading to a range for the LED  
    analogWrite(9, map(sensorValue, 0, 1023, 0, 255));  
    delay(100); // Wait for 100 millisecond(s)  
}
```



Code

Start Simulation Export Send To

1 (Arduino Uno R3)

Text

```
1 // C++ code  
2 //  
3 int sensorValue = 0;  
4  
5 void setup()  
6 {  
7     pinMode(A0, INPUT);  
8     Serial.begin(9600);  
9  
10    pinMode(9, OUTPUT);  
11 }  
12  
13 void loop()  
14 {  
15     // read the value from the sensor  
16     sensorValue = analogRead(A0);  
17     // print the sensor reading so you know its range  
18     Serial.println(sensorValue);  
19     // map the sensor reading to a range for the LED  
20     analogWrite(9, map(sensorValue, 0, 1023, 0, 255));  
21     delay(100); // Wait for 100 millisecond(s)  
22 }
```

Serial Monitor

Begin Simulation

The screenshot shows the Arduino IDE interface. On the left is a breadboard simulation window displaying a circuit with an Arduino Uno board. The circuit includes a red LED connected to digital pin 9 through a current-limiting resistor. A green wire connects digital pin 9 to analog pin A0. Two yellow potentiometers are connected between analog pins A0 and A1. The Arduino Uno board is shown with its pins labeled, and a USB cable is connected to its port. On the right is the main IDE window with the following C++ code:

```
// C++ code
//
int sensorValue = 0;

void setup()
{
    pinMode(A0, INPUT);
    Serial.begin(9600);

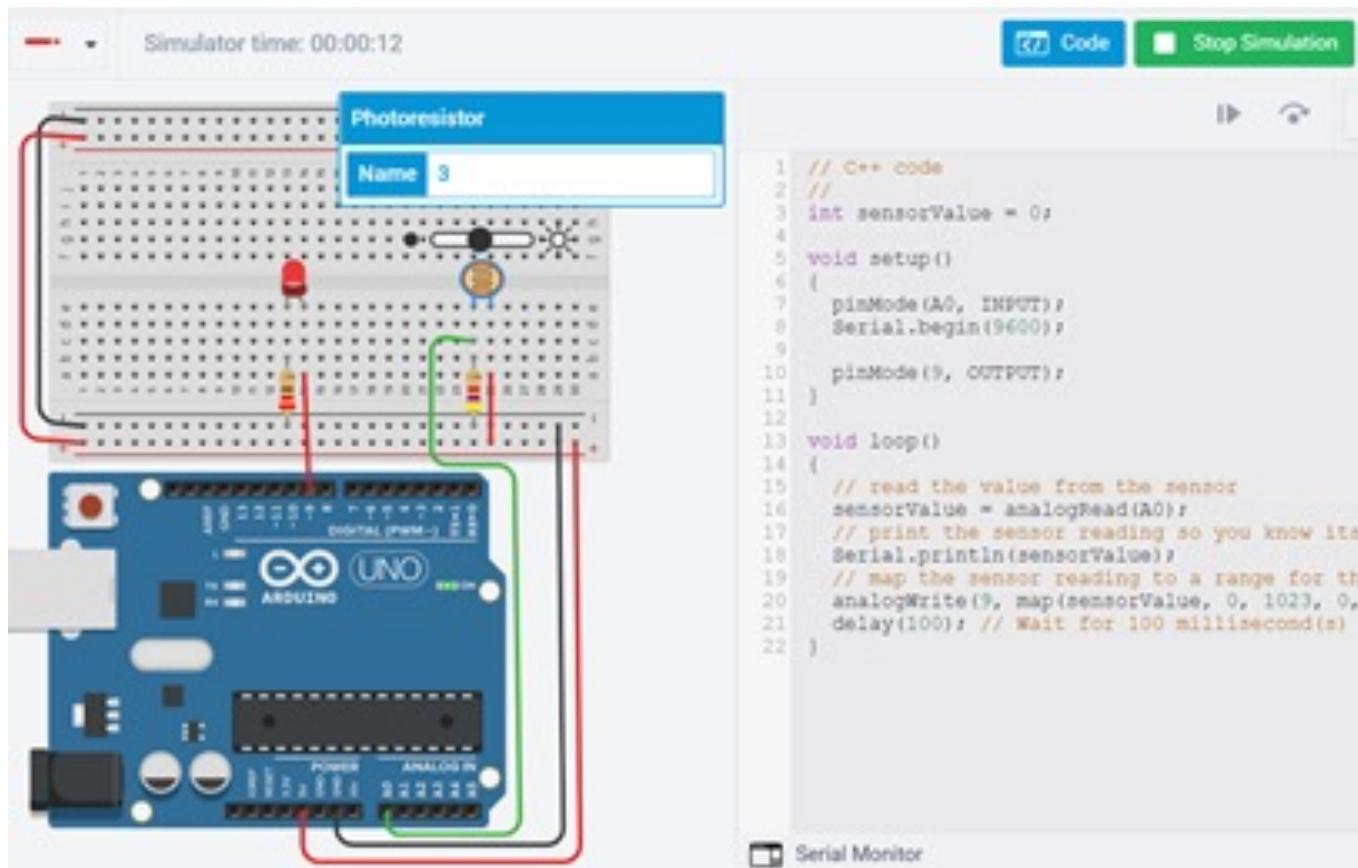
    pinMode(9, OUTPUT);
}

void loop()
{
    // read the value from the sensor
    sensorValue = analogRead(A0);
    // print the sensor reading so you know its range
    Serial.println(sensorValue);
    // map the sensor reading to a range for the LED
    analogWrite(9, map(sensorValue, 0, 1023, 0, 255));
    delay(100); // Wait for 100 millisecond(s)
}
```

A red oval highlights the "Start / stop simulation" button in the toolbar above the code editor. The status bar at the bottom indicates "Serial Monitor".

Simulation and Review

- Click on the Start Simulation to review your results. Your Photoresistor will display a dimming scroll for you to change from left to right. By scrolling from left to right you can see the LED turn on and off at your control.



Stop Simulation

Simulator time: 00:00:12

Code Stop Simulation

Photoresistor
Name 3

```
// C++ code
//
int sensorValue = 0;

void setup()
{
  pinMode(A0, INPUT);
  Serial.begin(9600);

  pinMode(9, OUTPUT);
}

void loop()
{
  // read the value from the sensor
  sensorValue = analogRead(A0);
  // print the sensor reading so you know its
  Serial.println(sensorValue);
  // map the sensor reading to a range for the
  analogWrite(9, map(sensorValue, 0, 1023, 0,
  delay(100); // Wait for 100 millisecond(s)
}
```

Serial Monitor

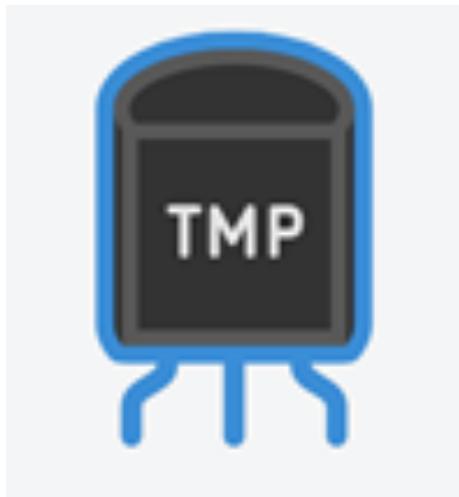
Virtual Activity - 4

Temperature sensor

Temperature Sensor TMP36

Temperature Sensor (Analog Input):

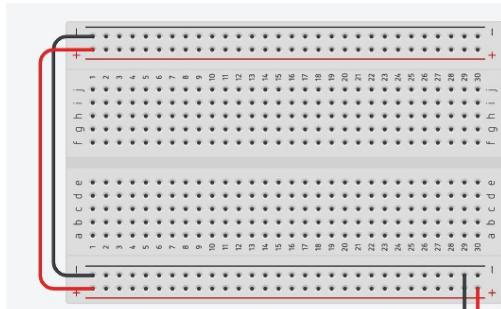
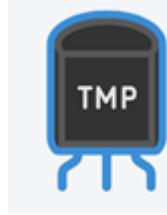
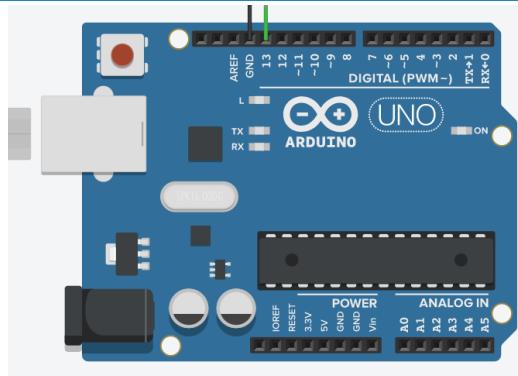
In this project, you will turn the Arduino into a thermometer. Use a temperature sensor to measure your skin temperature, and register the output with three LEDs. Even though the Arduino is a digital tool, it can interpret signals from an analog input, like the temperature sensor, using the built in Analog-to-Digital (ADC) converter, accessed through the analog pins.



Components

Components

- Arduino
- TMP36 Temperature sensor
- Breadboard
- Three Resistors
- Three LEDs



Resistor



LED

Starting a new circuit

Dashboard | Tinkercad [x](#) [+](#) tinkercad.com/dashboard?&type=tinkercad&collection=designs [🔍](#) [☆](#) [ABP](#)

**TIN
KER
CAD** AUTODESK® TINKERCAD®

Log in to www.tinkercad.com

My recent designs

Create new design

3D Designs **Circuits** **Codeblocks** **Lessons**

Click "Circuits."

Ben Finio

Search designs...

Cool Snicket 5 months ago Private

Glorious Kieran-Sango 8 months ago Private

Stunning Luulia 5 years ago Private

Powerful Amberis-Wolt 5 years ago Private

Allison Debellis @adebellis8

Project 1

Create project

Tweets Follow

Tinkercad Retweeted

81

Starting a new circuit



Gallery Blog Learn Teach Q

Ben Finio

Search designs...

3D Designs

Circuits

Codeblocks

NEW

Lessons

Your Classes

Projects

Project 1

Create project

Tweets

Follow

Tinkercad Retweeted

Allison Debellis

@adebellis8

Tinkercad Lesson Plans

Tinkercad lesson plans are ready to use online or in the classroom. Discover curriculum developed in partnership with teachers. [Learn more](#)

Circuits

Create new Circuit

Click "Create new Circuit."

Example Arduino Light Show (With R...

5 hours ago
Private

My First Arduino Circuit

7 hours ago
Public

Brave Wolt

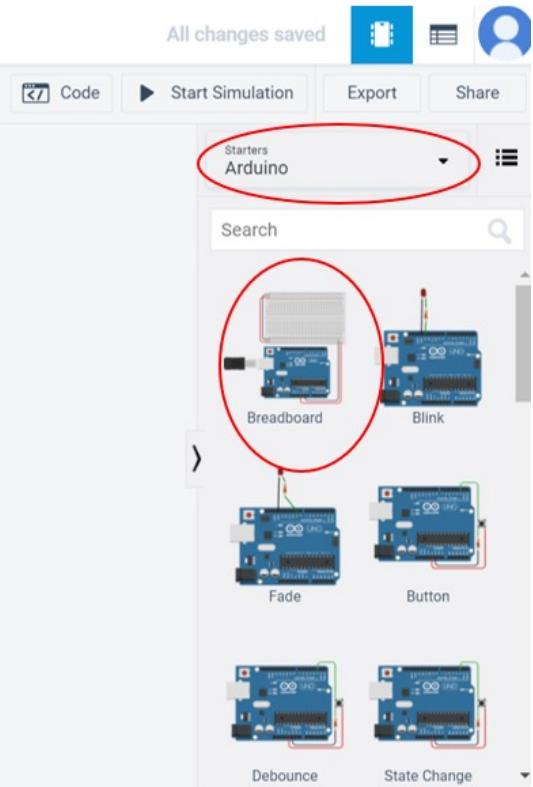
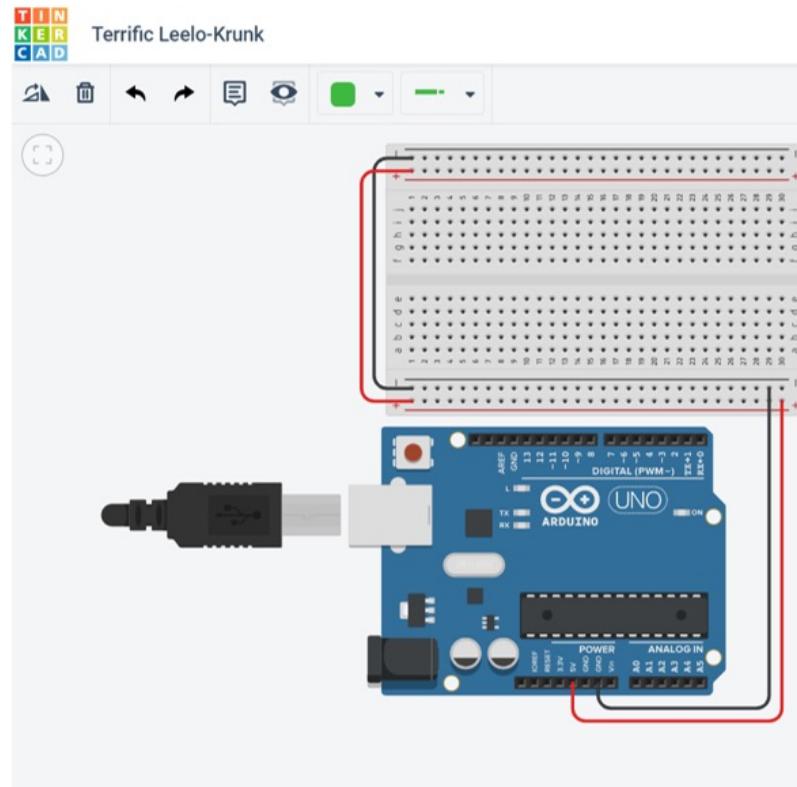
9 hours ago
Private

Spectacular Stantia-Jaiks

5 days ago
Private

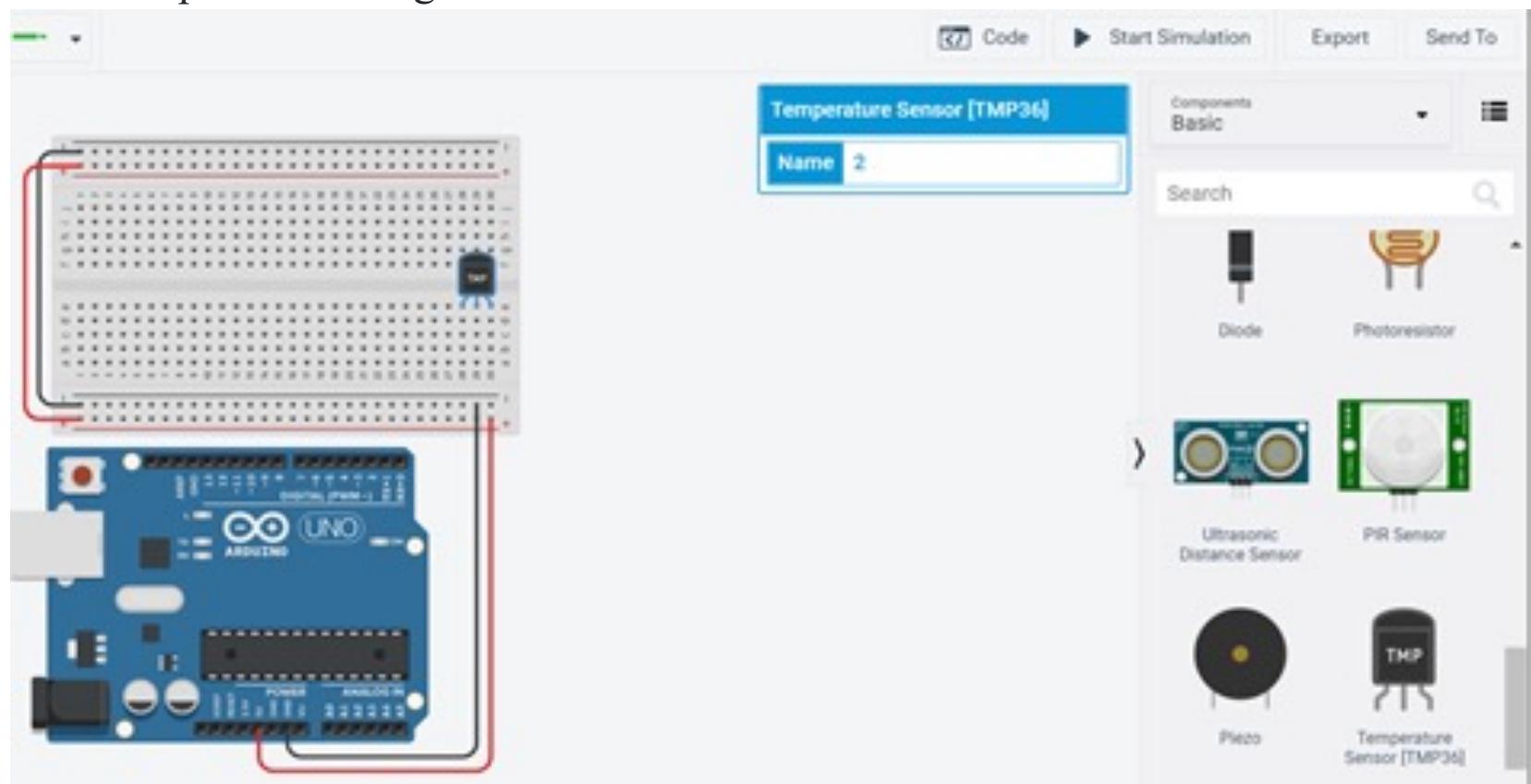
Setting up our Workplace

1. Under Starters/Components: click Arduino.
2. Click and drag on the first component, place it on the workplace and let go.
3. Repeat step two for the following components: two resistors, one LDR, and one LED.



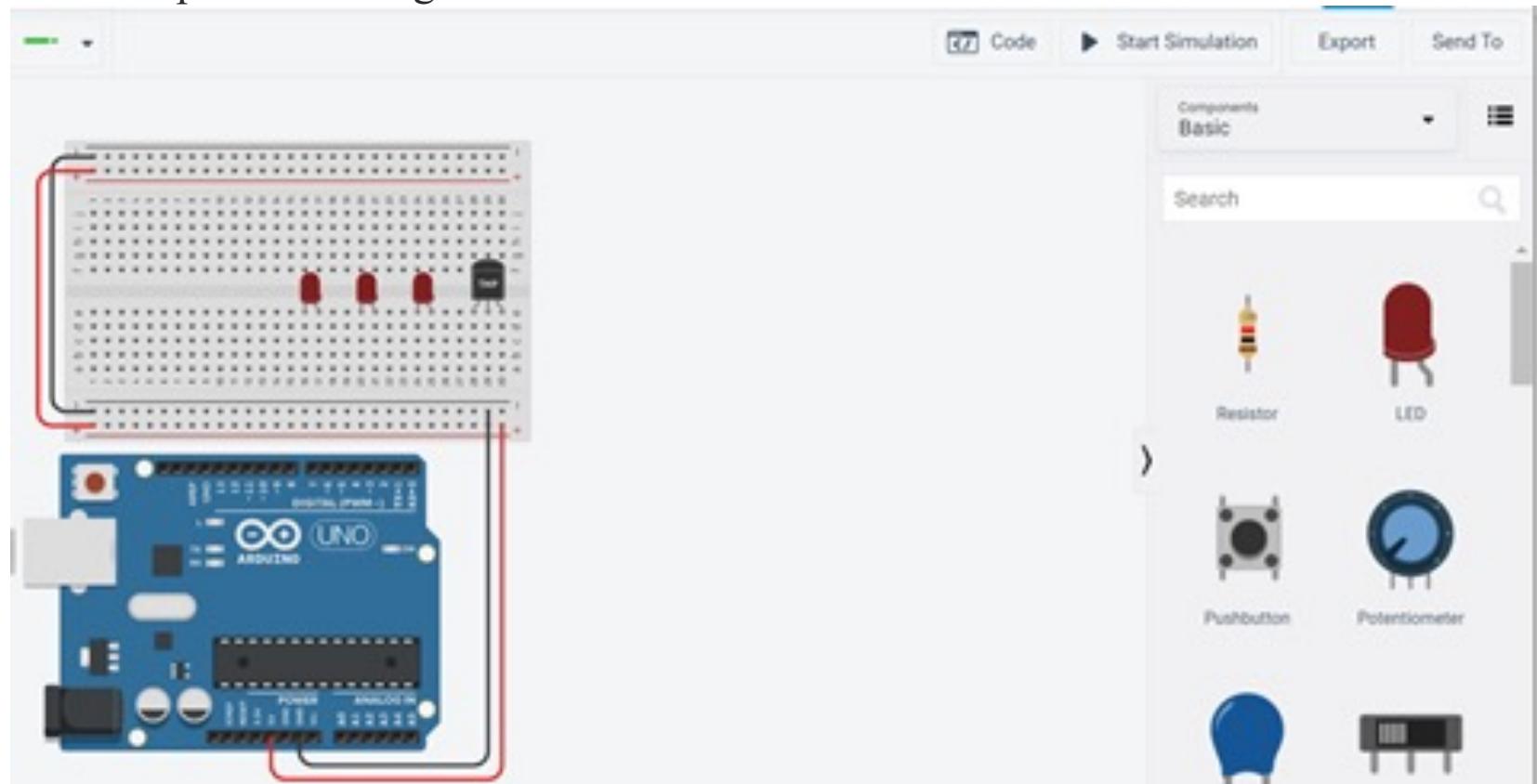
Setting up our Workplace

1. Under Basic/Components: click Temperature Sensor.
2. Click and drag on the component, place it on the workplace and let go.
3. The following three terminal on the temperature sensor will be connected:
Rows 28, 29, 30. Column e.



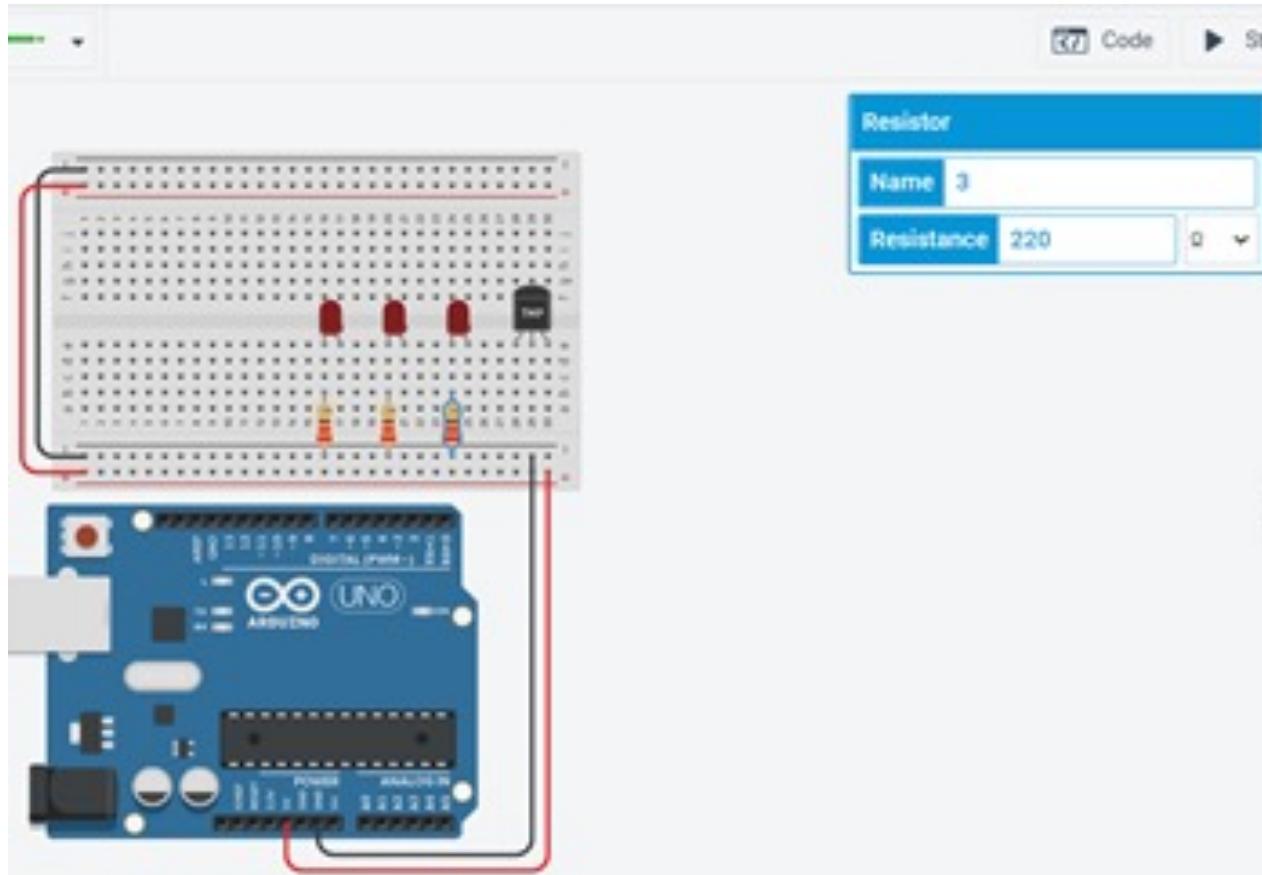
Setting up our Workplace

1. Under Basic/Components: click LED.
2. Click and drag on the component, place it on the workplace and let go.
3. The following three LEDs will be connected to: Rows 16, 17, Column e. Rows 20, 21, Column e. Rows 24, 25, Column e.



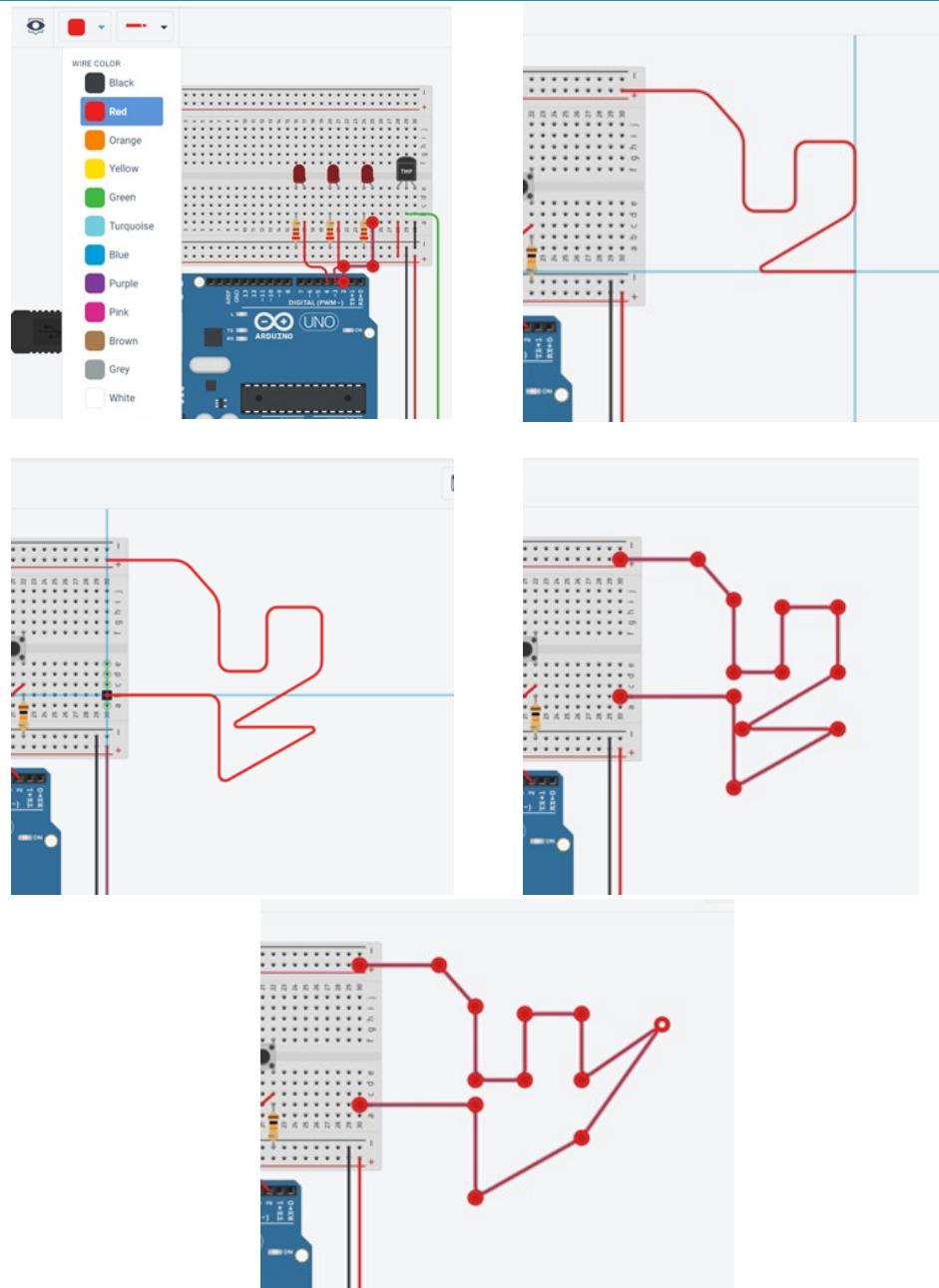
Setting up our Workplace

1. Under Basic/Components: click Resistor.
2. Click and drag on the component, place it on the workplace and let go.
3. The following three Resistors will be connected to: (The top ends) Row 16, Column b. Row 20, Column b. Row 24, Column b. The bottom ends will connect to Ground bus, aligned with Rows 16, 20, 24. Set resistors to 220 Ohms.



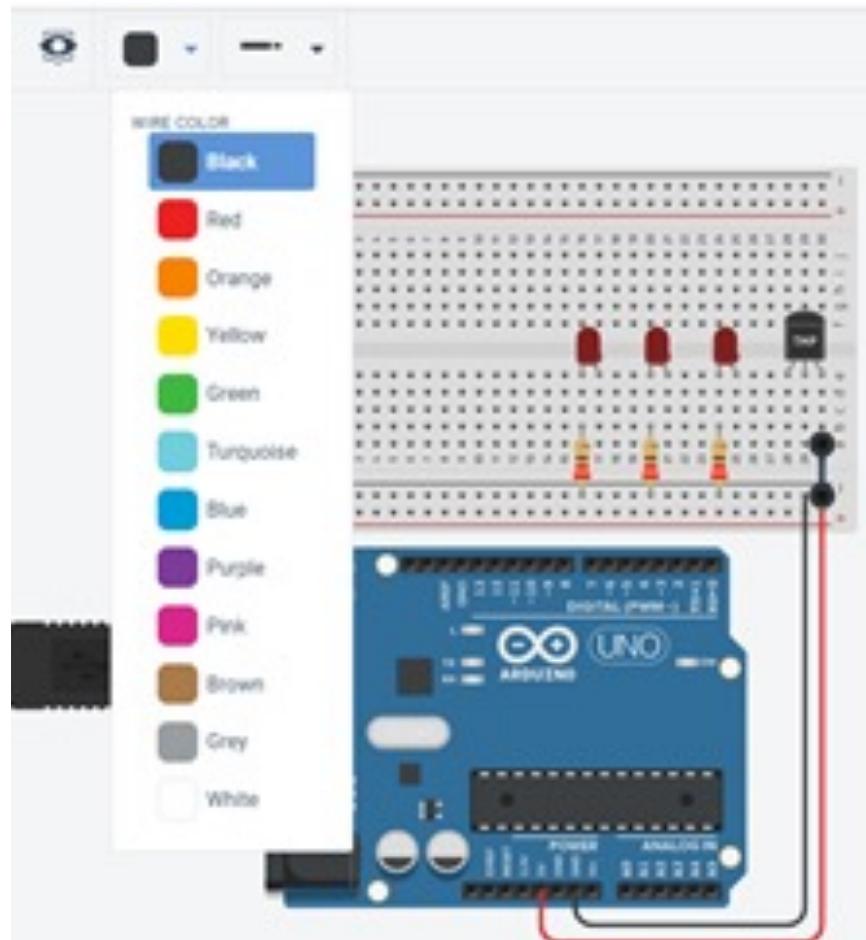
Wiring Orientation

1. Choose a color for the wire.
Commonly, your ground will be black, and power will be red.
2. Next choose a start point and click once, you must choose a port or a location on the bread board. You will now have a wire that can be stretched almost at any angle.
3. In order to make your wire "curve" at different points, you can right click, as long as you do not click when hovering over a port or else it'll assumingly connect to that port.
4. You add as many turning points as you want, and they will appear after you connect to a port.
5. After which you can stretch the wire through its' turning points.



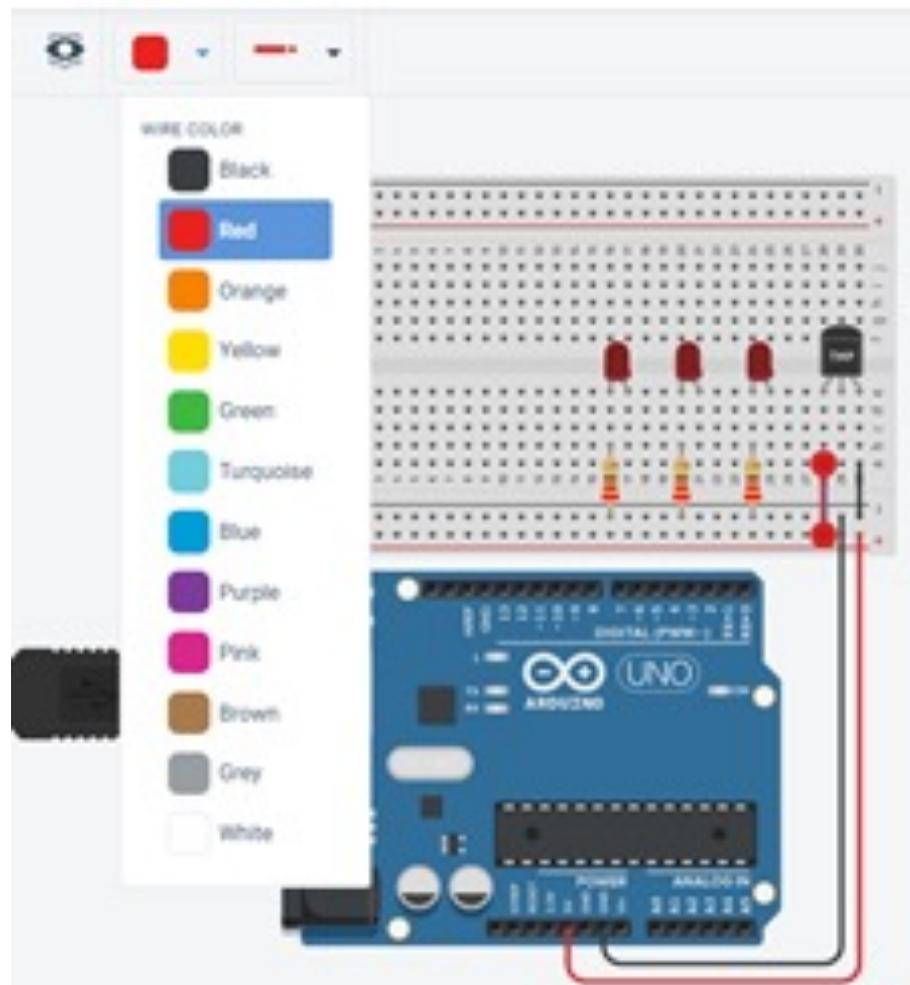
Wiring components

1. For this wire select the color black.
2. One end will begin on the Ground bus aligned to Row 30. The other end will connect to Row 30, Column a.



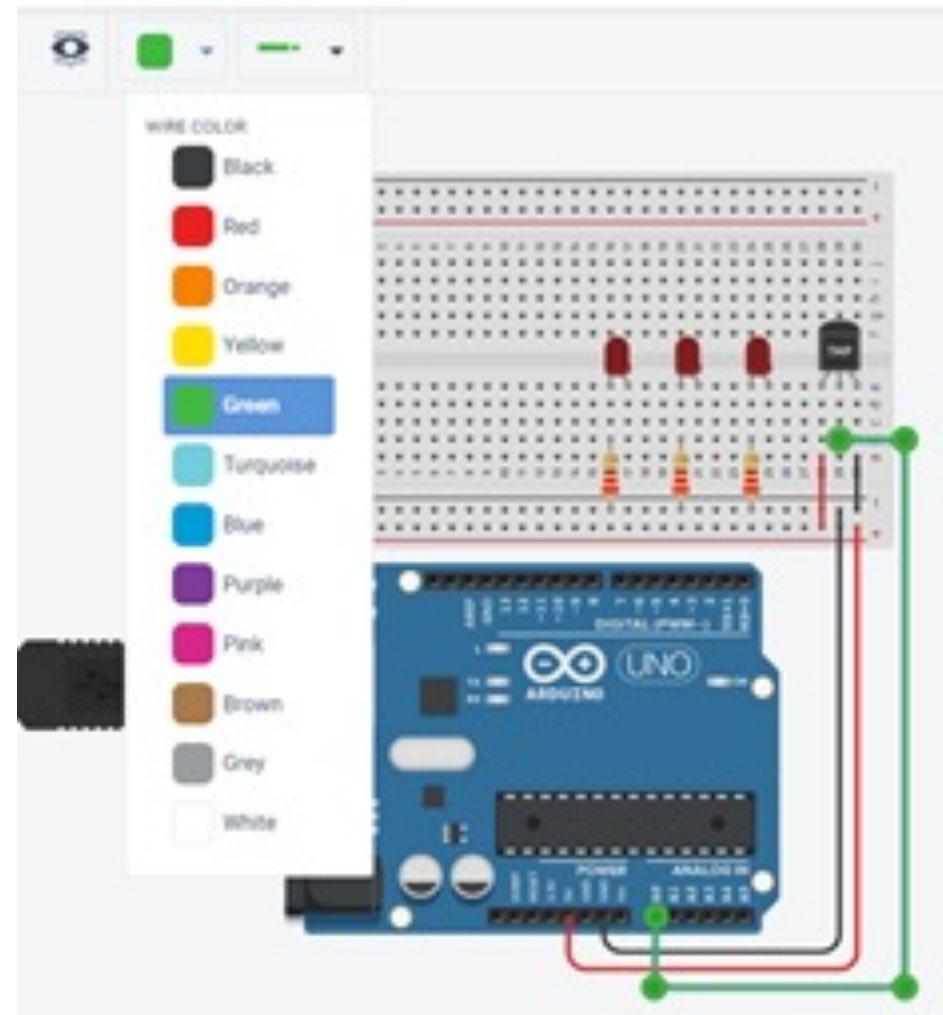
Wiring components

1. For this wire select the color red.
2. One end will begin on the Positive bus aligned to Row 28. The other end will connect to Row 28, Column a.



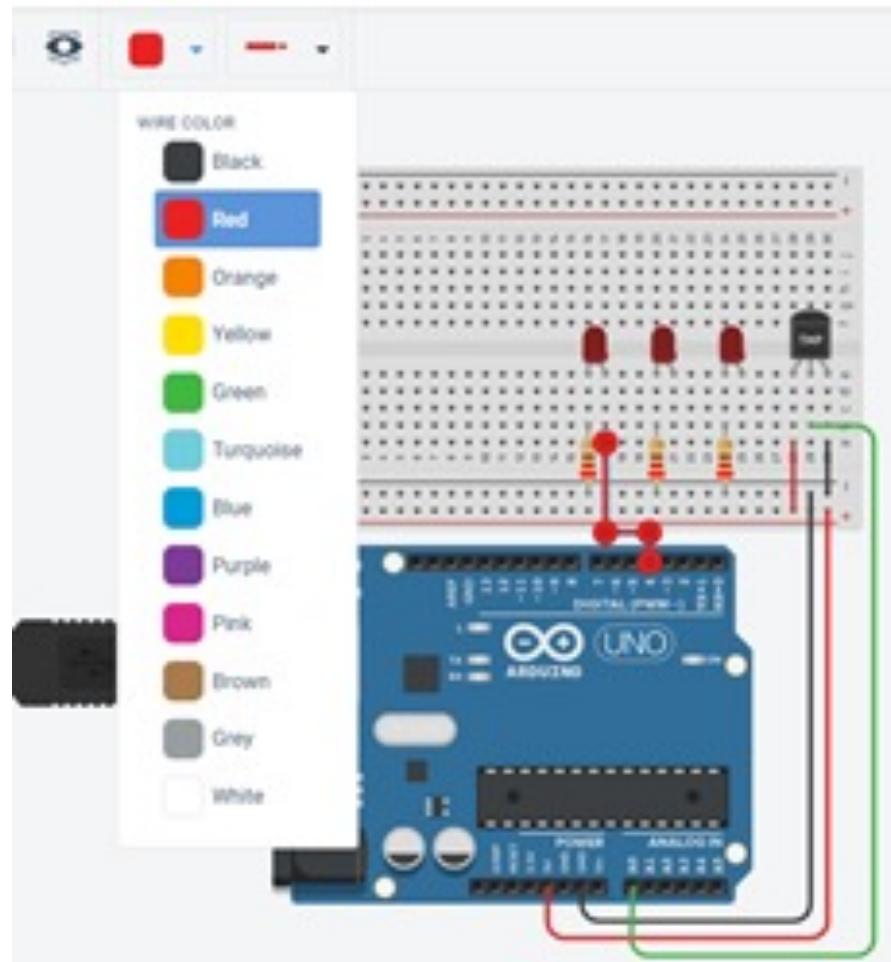
Wiring components

1. For this wire select the color green.
2. One end will begin on the Analog Port A0. The other end will connect to Row 29, Column b.



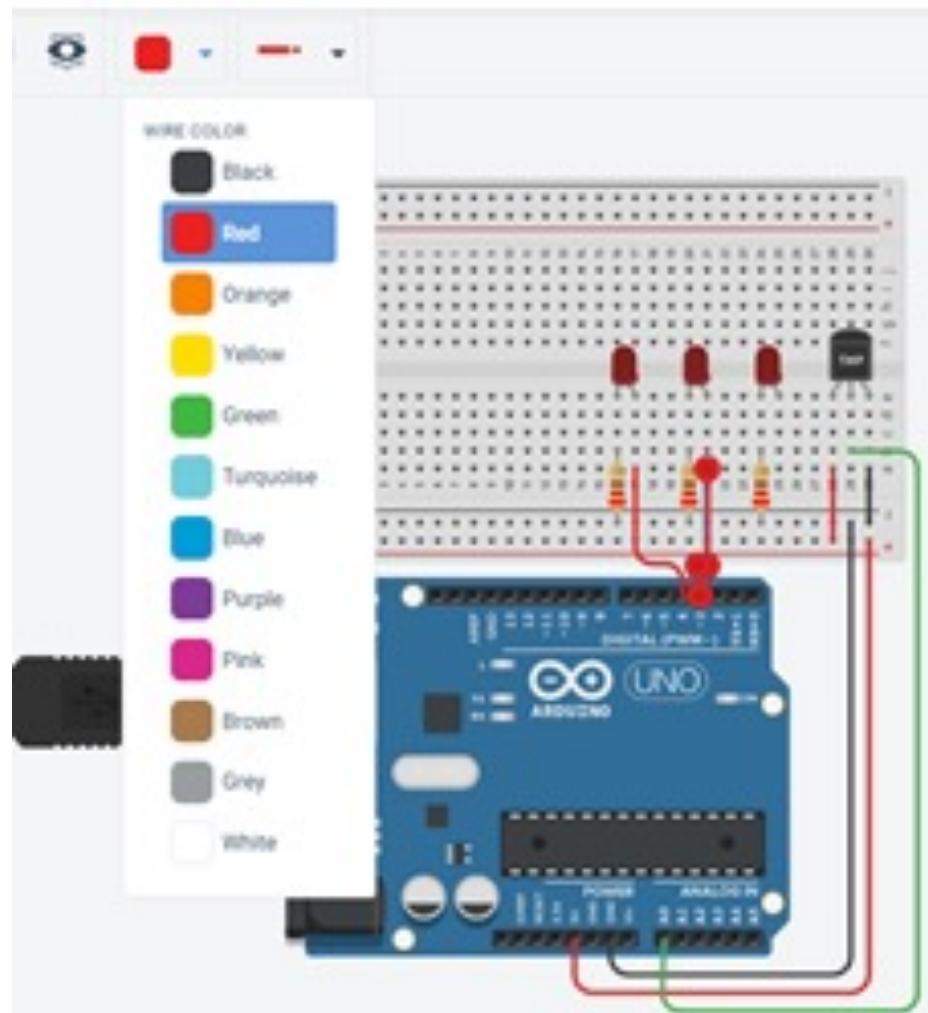
Wiring components

1. For this wire select the color red.
2. One end will begin on Port 4. The other end will connect to Row 17, Column a.



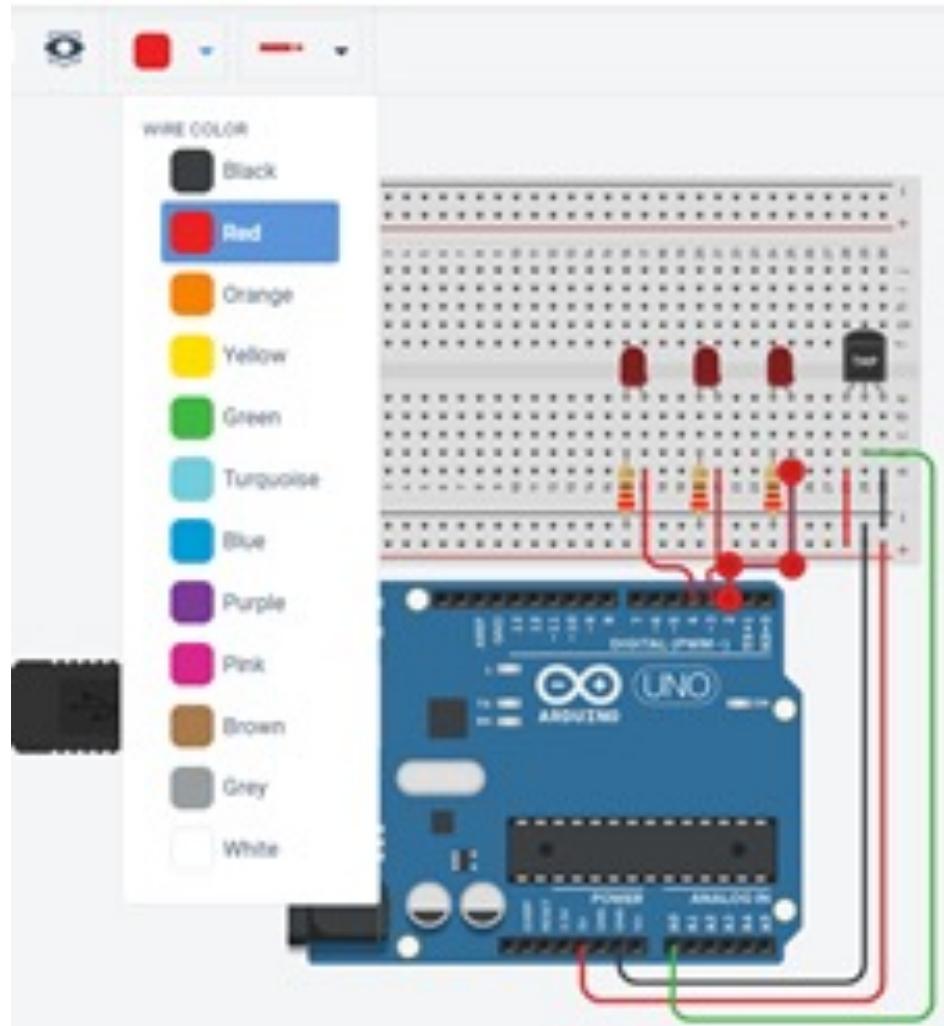
Wiring components

1. For this wire select the color red.
2. One end will begin on Port 3. The other end will connect to Row 21, Column a.

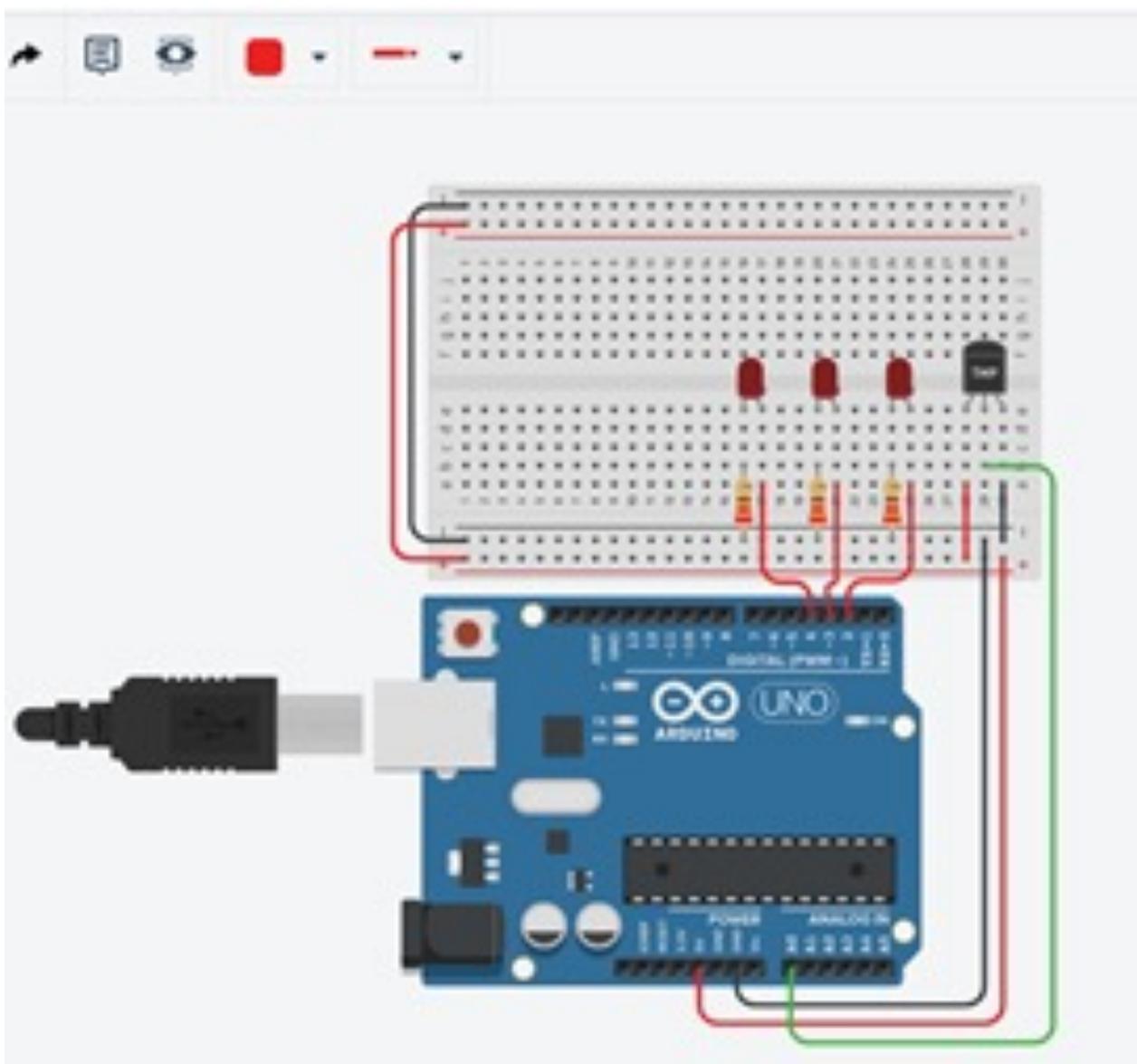


Wiring components

1. For this wire select the color red.
2. One end will begin on Port 2. The other end will connect to Row 25, Column a.



Final Product



Analog input with Arduino

Code

- Copy the following code
Part 1 of 3:

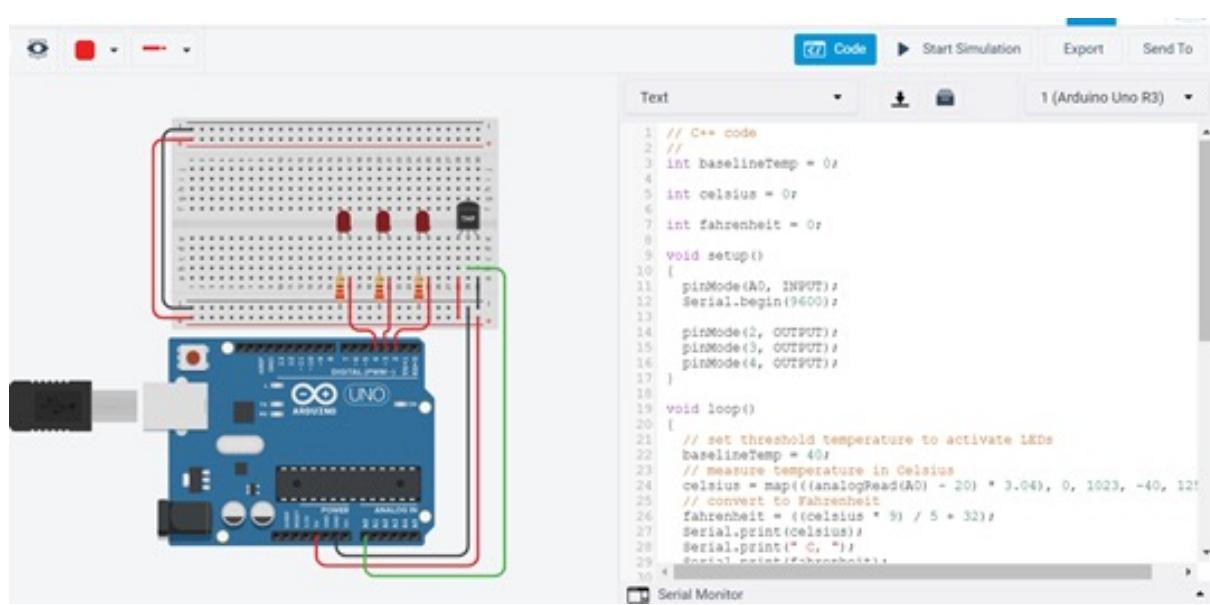
```
// C++ code
//
int baselineTemp = 0;

int celsius = 0;

int fahrenheit = 0;

void setup()
{
    pinMode(A0, INPUT);
    Serial.begin(9600);

    pinMode(2, OUTPUT);
    pinMode(3, OUTPUT);
    pinMode(4, OUTPUT);
}
```



Analog input with Arduino

Code

→ Copy the following code

Part 2 of 3:

```
void loop()
{
    // set threshold temperature to activate LEDs
    baselineTemp = 40;
    // measure temperature in Celsius
    celsius = map((analogRead(A0) - 20) * 3.04), 0, 1023, -40, 125);
    // convert to Fahrenheit
    fahrenheit = ((celsius * 9) / 5 + 32);
    Serial.print(celsius);
    Serial.print(" C, ");
    Serial.print(fahrenheit);
    Serial.println(" F");
    if (celsius < baselineTemp) {
        digitalWrite(2, LOW);
        digitalWrite(3, LOW);
        digitalWrite(4, LOW);
    }
    if (celsius >= baselineTemp && celsius < baselineTemp + 10) {
        digitalWrite(2, HIGH);
        digitalWrite(3, LOW);
        digitalWrite(4, LOW);
    }
}
```

Analog input with Arduino

Code

→ Copy the following code

Part 3 of 3:

```
}
```

```
if (celsius >= baselineTemp + 10 && celsius < baselineTemp +
```

```
20) {
```

```
    digitalWrite(2, HIGH);
```

```
    digitalWrite(3, HIGH);
```

```
    digitalWrite(4, LOW);
```

```
}
```

```
if (celsius >= baselineTemp + 20 && celsius < baselineTemp +
```

```
30) {
```

```
    digitalWrite(2, HIGH);
```

```
    digitalWrite(3, HIGH);
```

```
    digitalWrite(4, HIGH);
```

```
}
```

```
if (celsius >= baselineTemp + 30) {
```

```
    digitalWrite(2, HIGH);
```

```
    digitalWrite(3, HIGH);
```

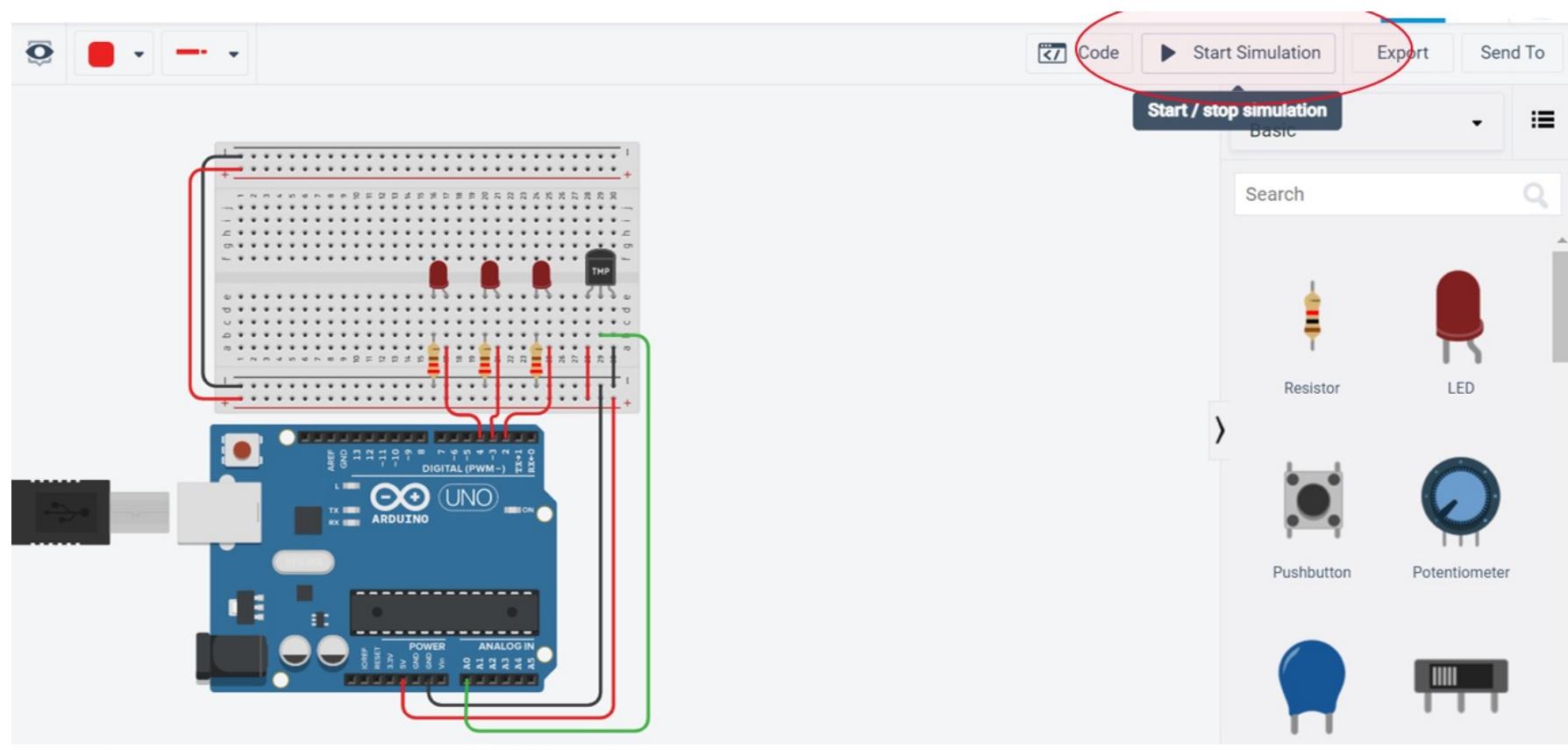
```
    digitalWrite(4, HIGH);
```

```
}
```

```
delay(1000); // Wait for 1000 millisecond(s)
```

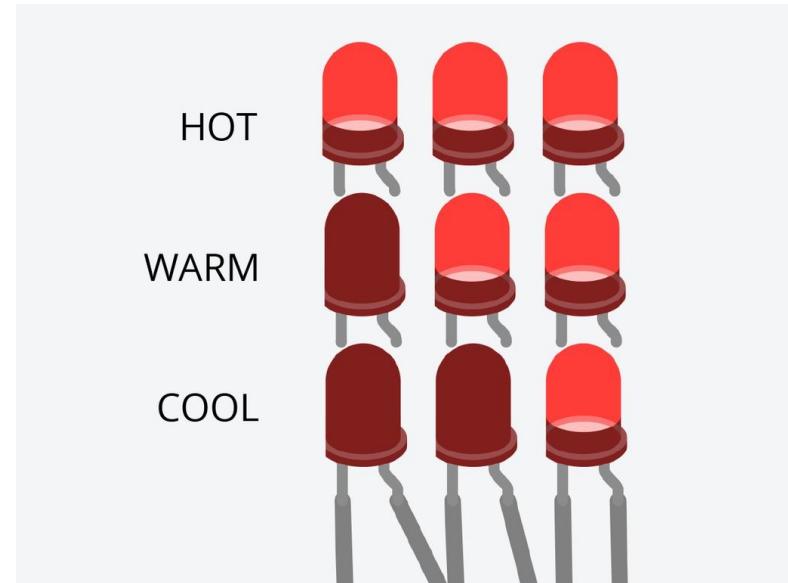
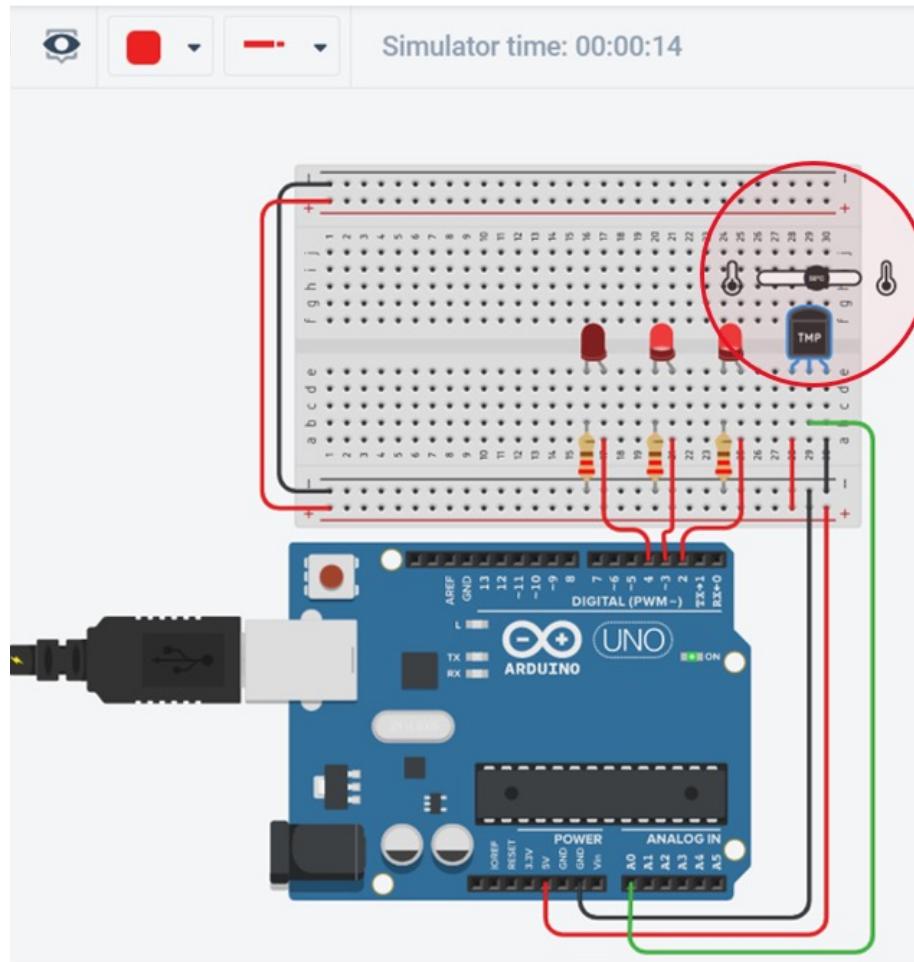
```
}
```

Begin Simulation

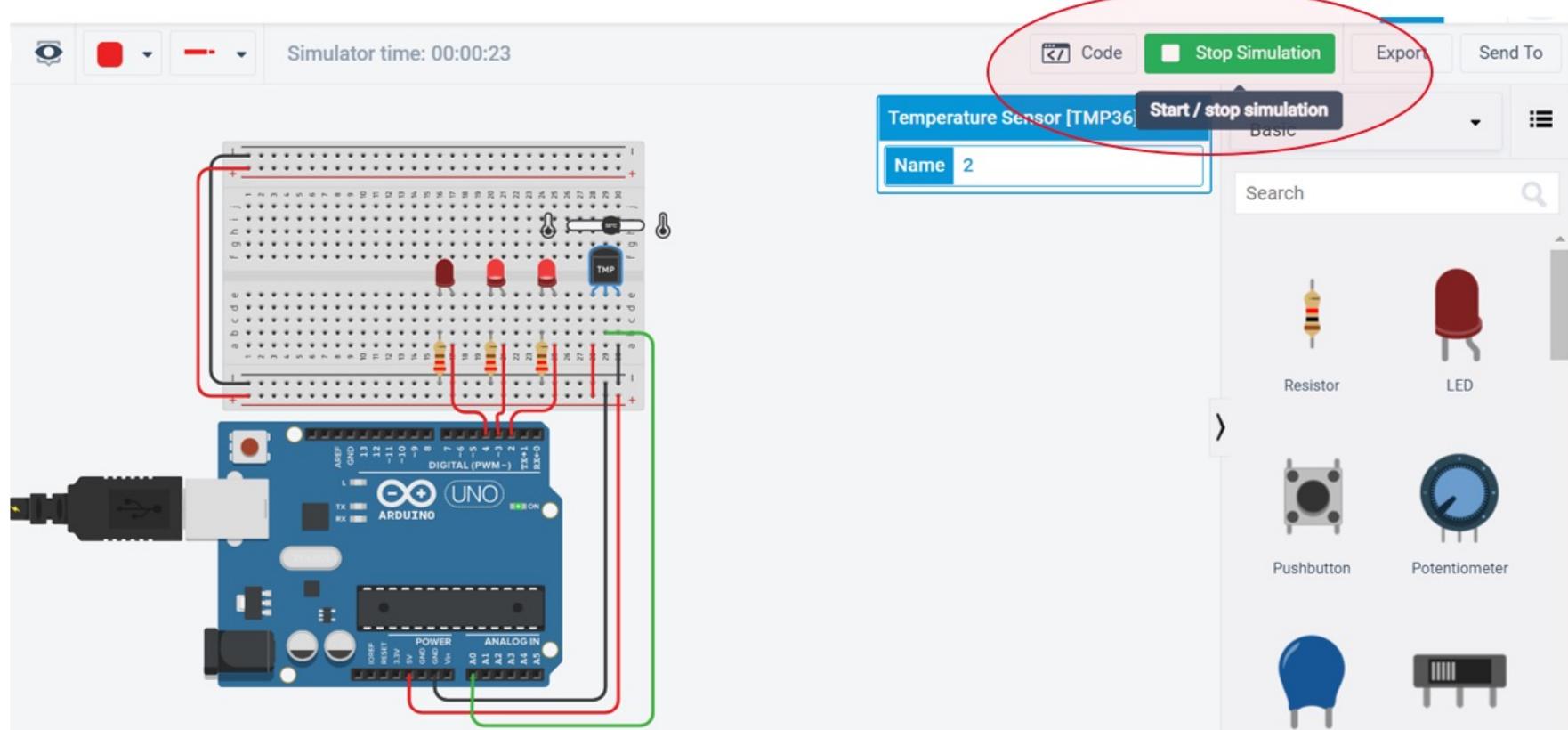


Review Simulation

Play with the temperature dial. This change in temperature represents the change in temperature in real life, like the temperature outside or your body temperature.



End Simulation



Virtual Activity - 5

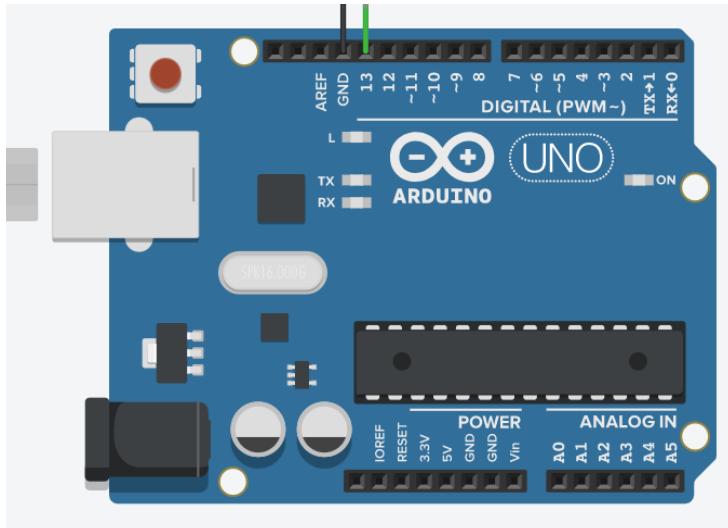
Keypad with Arduino

4x4 Keypad with Arduino

We will introduce how to use a 4x4 matrix keypad.

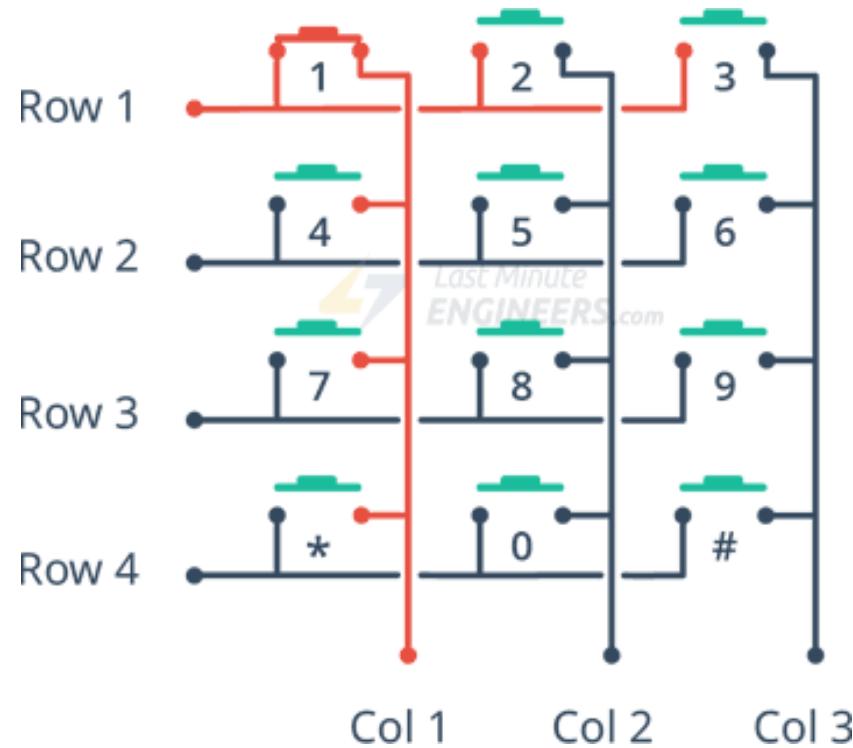
Components

- Arduino
- Keypad



The Keypad

Shown below is a gif of a 4x3 keypad,
we will be using a 4x4 keypad.



Starting a new circuit

The screenshot shows the Tinkercad Dashboard. At the top left is the Autodesk Tinkercad logo. The top right features links for 'Gallery', 'Blog', 'Learn', 'Teach', and a search icon. A red arrow points from the top center to a black box containing the text 'Log in to www.tinkercad.com'. Another red arrow points from the left sidebar to the 'Circuits' button, which is highlighted with a blue background and white text. A third red arrow points from the bottom left to the 'Follow' link. On the left sidebar, there are sections for '3D Designs' (selected), 'Circuits', 'Codeblocks' (NEW), and 'Lessons'. Below that is a 'Your Classes' section with 'Projects' (Project 1) and a 'Create project' button. At the very bottom, it shows 'Tweets' and a 'Follow' link. The main content area displays 'My recent designs' with cards for various 3D models like a hammer, a green pyramid, and a red cube. A modal window titled 'Tinkercad Lesson Plans' is open, showing two children working on a computer.

Starting a new circuit



Gallery Blog Learn Teach Q



Ben Finio

Search designs...

3D Designs

Circuits

Codeblocks

NEW

Lessons

Your Classes

Projects

Project 1

Create project

Tweets

Follow

Tinkercad Retweeted



Allison Debellis
@adebellis8



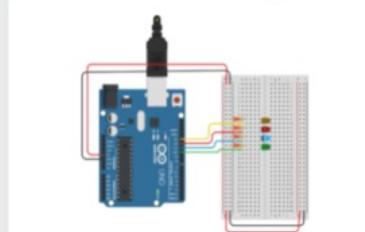
Tinkercad Lesson Plans

Tinkercad lesson plans are ready to use online or in the classroom. Discover curriculum developed in partnership with teachers. [Learn more](#)

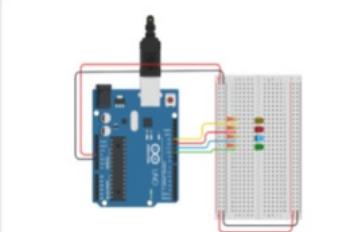
Circuits

Create new Circuit

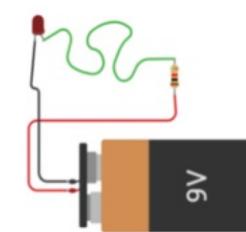
Click "Create new Circuit."



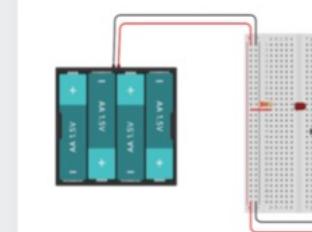
Example Arduino Light Show (With R...
5 hours ago
Private



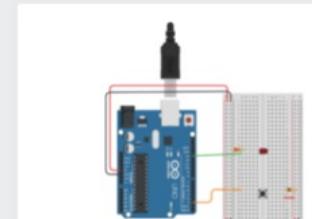
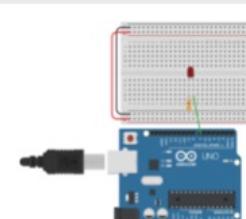
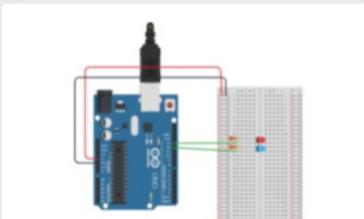
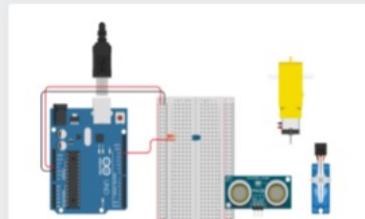
My First Arduino Circuit
7 hours ago
Public



Brave Wolt
9 hours ago
Private

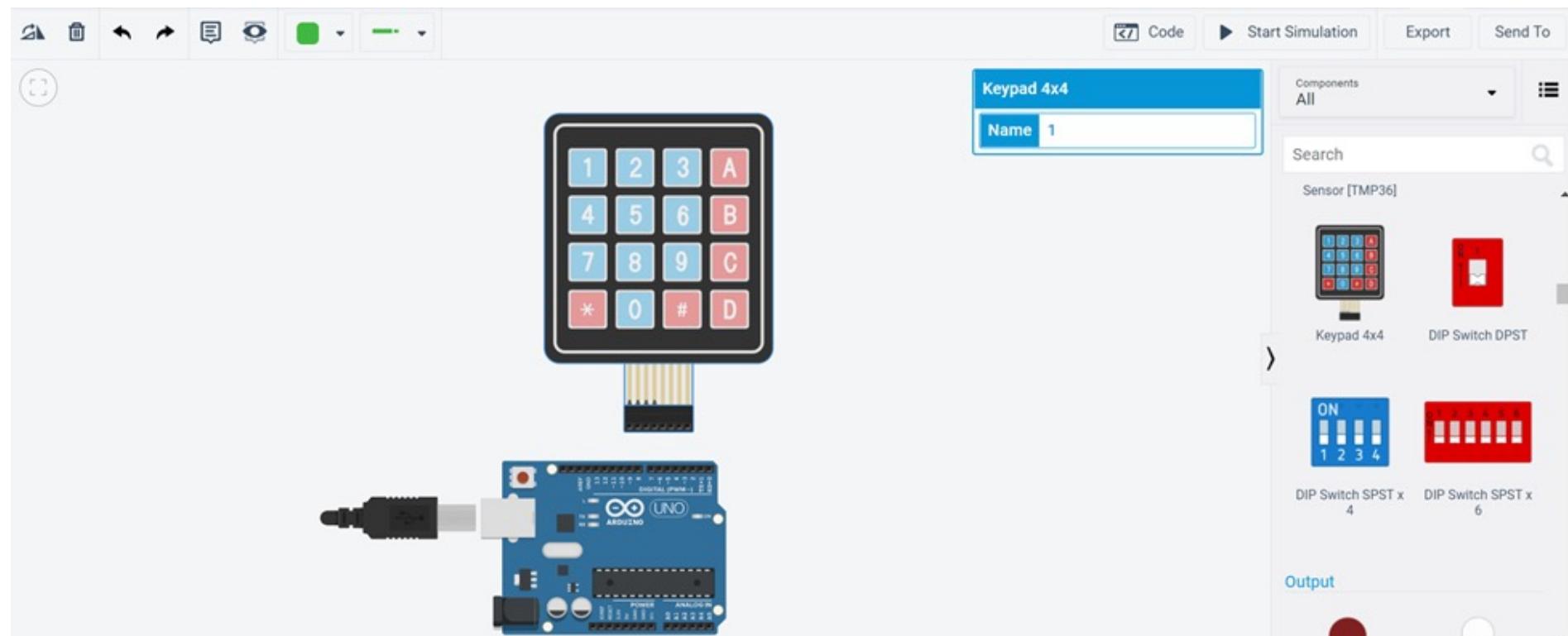


Spectacular Stantia-Jaiks
5 days ago
Private



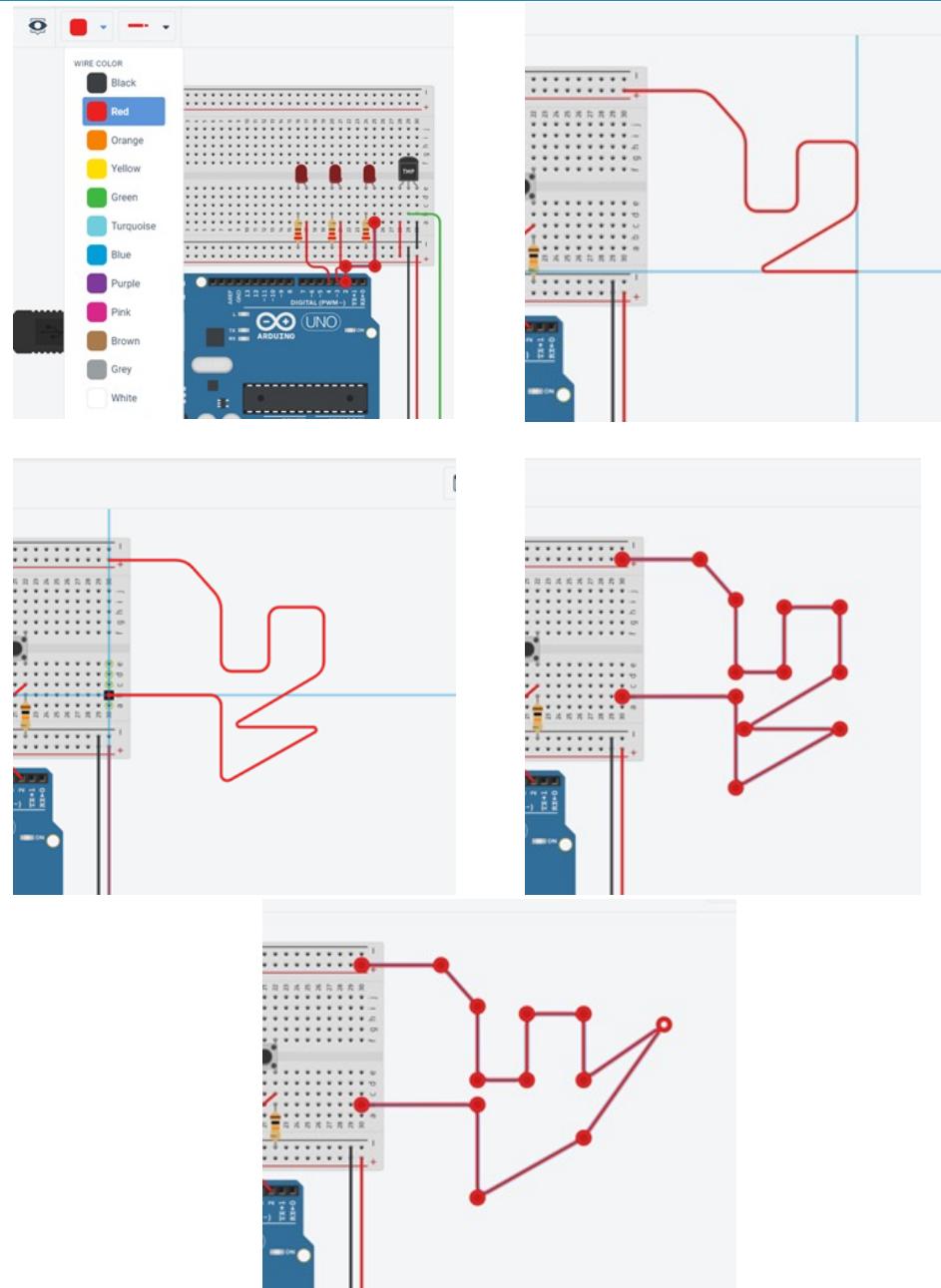
Setting up our Workplace

1. Under Starters/Components: click Arduino.
2. Click and drag on the first component, place it on the workplace and let go.
3. Repeat step two for the following components: 4x4 Keypad.



Wiring Orientation

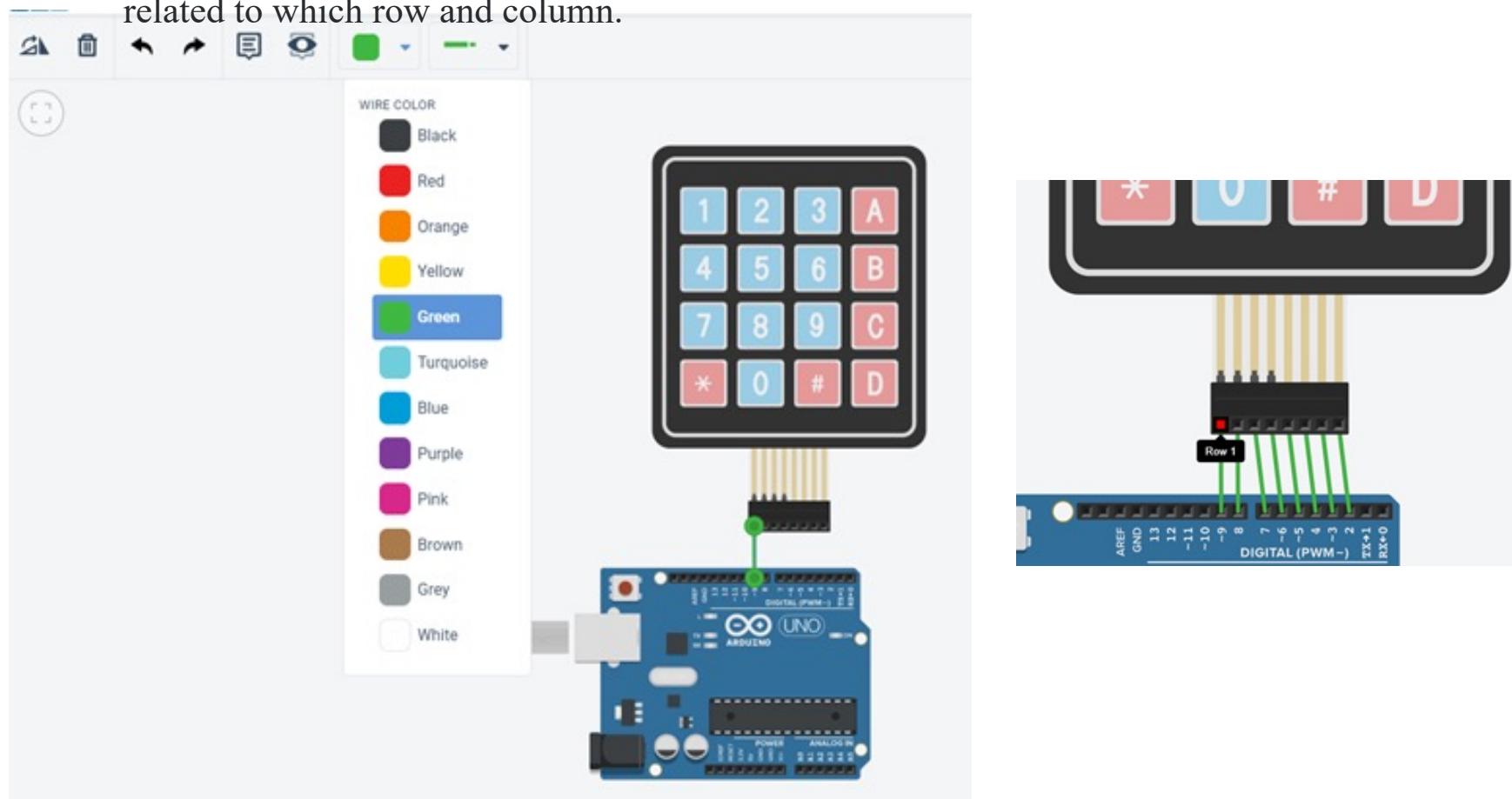
1. Choose a color for the wire.
Commonly, your ground will be black, and power will be red.
2. Next choose a start point and click once, you must choose a port or a location on the bread board. You will now have a wire that can be stretched almost at any angle.
3. In order to make your wire "curve" at different points, you can right click, as long as you do not click when hovering over a port or else it'll assumingly connect to that port.
4. You add as many turning points as you want, and they will appear after you connect to a port.
5. After which you can stretch the wire through its' turning points.



Wiring a 4x4 Keypad with Arduino

Set up

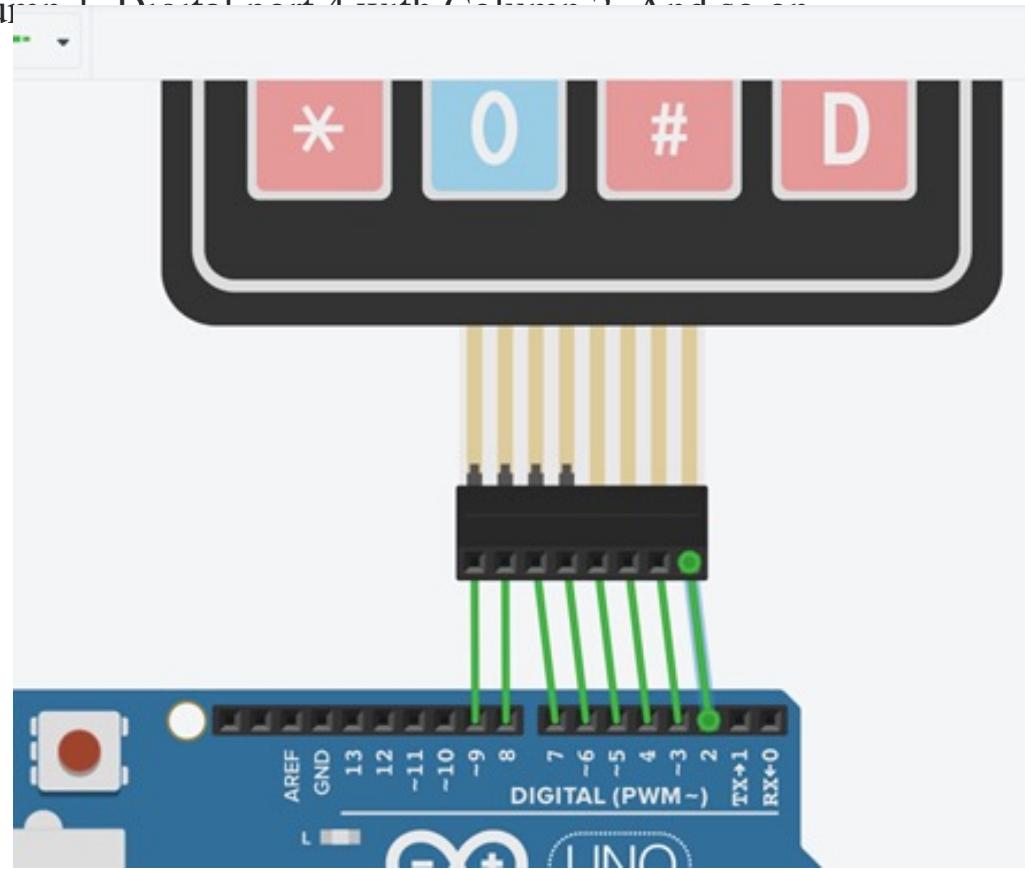
1. For this wire select the color green.
2. One end will begin on the Digital Port 9 and the other end will connect to "Row 1" port. If you hover your mouse over each port on the keypad you can see which port is related to which row and column.



Wiring a 4x4 Keypad with Arduino

Set up

1. For this wire select the color green.
2. Continue the process from the last slide and wire from left to right. Notice how the digital ports on the Arduino align with keypad port. Digital port 9 with Row 1. Digital port 8 with Row 2. Digital port 7 with Row 3. Digital port 6 with Row 4. Digital port 5 with Column 1. Digital port 4 with Column 2 and so on.



Final Product



Keypad Code

Code

→ Copy the following code

Part 1 of 2:

```
#include <Keypad.h>

const byte ROWS = 4; //global variable that represents four rows
const byte COLS = 4; //global variable that represents four columns

char keys[ROWS][COLS] = {
    {'1','2','3','A'},
    {'4','5','6','B'},
    {'7','8','9','C'},
    {'*','0','#','D'}
};

byte rowPins[ROWS] = {9, 8, 7, 6}; //connect to the row pinouts of the keypad
byte colPins[COLS] = {5, 4, 3, 2}; //connect to the column pinouts of the keypad
```

Keypad Code

Code

- Copy the following code
Part 2 of 2:

```
//Create an object of keypad
Keypad keypad = Keypad( makeKeymap(keys), rowPins, colPins, ROWS, COLS );

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

void loop(){
    char key = keypad.getKey();// Read the key

    // Print if key pressed
    if (key){
        Serial.print("Key Pressed : ");
        Serial.println(key);
    }
}
```

Keypad Code

Code

→ Final Code Product:

The screenshot shows a code editor interface with the following details:

- Toolbar:** Includes "Code" (highlighted in blue), "Start Simulation", "Export", and "Send To".
- Text Area:** Shows the Arduino code for a keypad. The code defines a 4x4 keypad matrix, initializes row and column pins, creates a Keypad object, and implements setup() and loop() functions to read keys and print them to the Serial Monitor.
- Status Bar:** Shows "1 (Arduino Uno R3)" and a dropdown menu.
- Bottom Bar:** Includes "Serial Monitor" and other standard IDE icons.

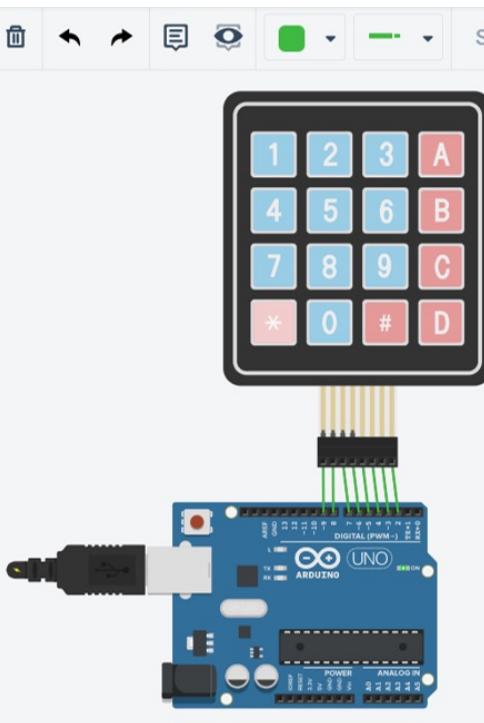
```
1 #include <Keypad.h>
2
3 const byte ROWS = 4; //global variable that represents four rows
4 const byte COLS = 4; //global variable that represents four columns
5
6 char keys[ROWS][COLS] = {
7     {'1','2','3','A'},
8     {'4','5','6','B'},
9     {'7','8','9','C'},
10    {'*','0','#','D'}
11 };
12
13 byte rowPins[ROWS] = {9, 8, 7, 6}; //connect to the row pinouts of the keypad
14 byte colPins[COLS] = {5, 4, 3, 2}; //connect to the column pinouts of the keypad
15
16 //Create an object of keypad
17 Keypad keypad = Keypad( makeKeymap(keys), rowPins, colPins, ROWS, COLS );
18
19 void setup(){
20     Serial.begin(9600);
21 }
22
23 void loop(){
24     char key = keypad.getKey(); // Read the key
25
26     // Print if key pressed
27     if (key){
28         Serial.print("Key Pressed : ");
29         Serial.println(key);
30     }
31 }
```

Begin Simulation



Review Simulation

Notice how the Serial Monitor displays and tracks what has been pressed on the keypad.



The diagram shows an Arduino Uno microcontroller connected to a 4x4 keypad. The keypad is represented by a 4x4 grid of buttons. The top row contains buttons labeled 1, 2, 3, and A. The second row contains buttons labeled 4, 5, 6, and B. The third row contains buttons labeled 7, 8, 9, and C. The bottom row contains buttons labeled *, 0, #, and D. A ribbon cable connects the keypad to the Arduino Uno. The Arduino Uno has its USB port connected to a computer.

Simulator time: 00:00:30

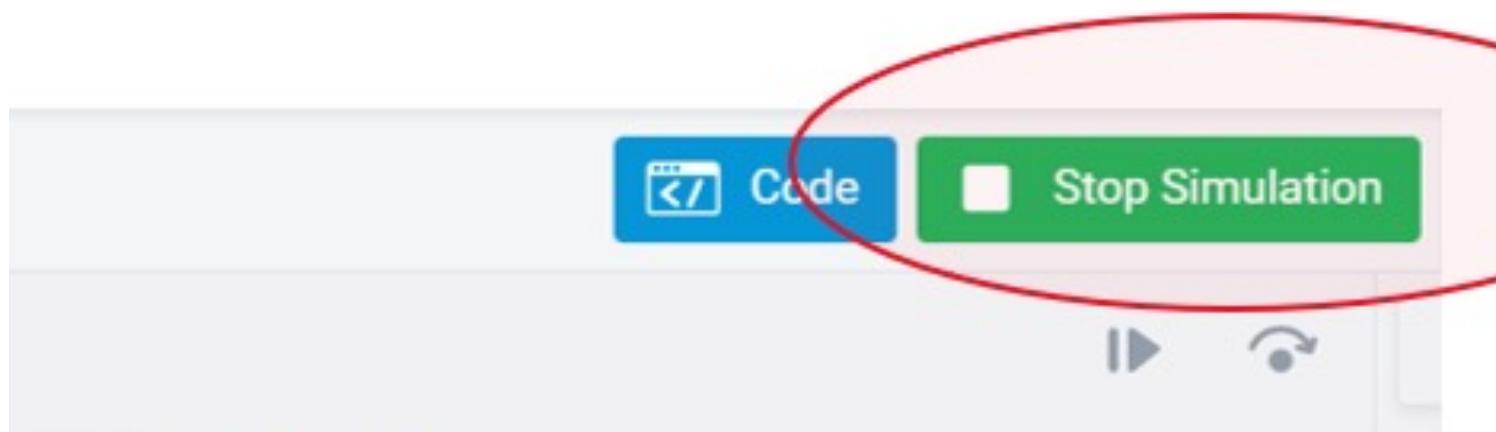
Code Stop Simulation Export Send To 1 (Arduino Uno R3)

```
1 #include <Keypad.h>
2
3 const byte ROWS = 4; //global variable that represents four rows
4 const byte COLS = 4; //global variable that represents four columns
5
6 char keys[ROWS][COLS] = {
7     {'1','2','3','A'},
8     {'4','5','6','B'},
9     {'7','8','9','C'},
10    {'*','0','#','D'}
11 };
12
13 byte rowPins[ROWS] = {9, 8, 7, 6}; //connect to the row pinouts of the keypad
14 byte colPins[COLS] = {5, 4, 3, 2}; //connect to the column pinouts of the keypad
15
16 //Create an object of keypad
17 Keypad keypad = Keypad( makeKeymap(keys), rowPins, colPins, ROWS, COLS );
```

Serial Monitor

```
Key Pressed : 3
Key Pressed : D
Key Pressed : 7
Key Pressed : *
```

Stop Simulation



More to Explore!

There are a few useful functions you can use with Keypad object. Few of them are listed below:

- `char waitForKey()` Waits forever until someone presses a key. Warning – It blocks all other code until a key is pressed. That means no blinking LED's, no LCD screen updates, no nothing with the exception of interrupt routines.
- `KeyState getState()` Returns the current state of any of the keys. The four states are IDLE, PRESSED, RELEASED and HOLD.
- `boolean keyStateChanged()` Let's you know when the key has changed from one state to another. For example, instead of just testing for a valid key you can test for when a key was pressed.
- `setHoldTime(unsigned int time)` Sets the amount of milliseconds the user will have to hold a button until the HOLD state is triggered.
- `setDebounceTime(unsigned int time)` Sets the amount of milliseconds the keypad will wait until it accepts a new keypress/keyEvent.
- `addEventListener(keypadEvent)` Triggers an event if the keypad is used.



Dr. Murat Kuzlu
mkuzlu@odu.edu

Jorge Ramirez III
jrami013@odu.edu