Basics of Arduino (TINKERCAD)

by Mukesh_Sankhla

Arduino and Tinkercad are two powerful tools that are gaining popularity in the world of electronics and engineering. While both of these tools are commonly used by hobbyists and professionals alike, many people may not have heard of them or may not fully understand what they are.

Arduino is an open-source electronics platform that is designed for building and programming electronic devices. It consists of a programmable microcontroller, or computer chip, that can be used to control electronic components such as LEDs, motors, sensors, and more. Arduino boards come in a variety of sizes and shapes, and they can be programmed using a variety of programming languages, including C and C++. Arduino is often used by hobbyists, artists, and designers to create interactive art installations, wearable technology, and other innovative projects.

Tinkercad is an online platform that allows users to design and simulate 3D models. It is a free and easy-to-use tool that is designed for beginners and experts alike. With Tinkercad, users can create 3D models of objects, buildings, and more, and then export them for 3D printing or use in other projects. Tinkercad also has a range of features that make it ideal for educational use, such as lesson plans, tutorials, and project ideas.

One of the most exciting things about Arduino and Tinkercad is that they can be used together. Tinkercad has a range of electronic components that can be used to build circuits and control devices, and it also has a built-in Arduino simulator that allows users to program and test their circuits in a virtual environment. This makes it easy to prototype and test new ideas before building them in the real world.

Overall, Arduino and Tinkercad are two powerful tools that are transforming the world of electronics and engineering. Whether you are a hobbyist, artist, designer, or educator, these tools offer endless possibilities for creating and innovating.

Thank You NextPCB:

This Demonstrations are successfully completed because of the help and support from NextPCB. NextPCB is one of the most experienced PCB manufacturers in Global, has specialized in the PCB and assembly industry for over 15 years. Not only could NextPCB provide the most innovative printed circuit boards and assembly technologies in the highest quality standards, the fastest delivery turnaround as fast as 24 hours.

Guys if you have a PCB project, please visit their website and get exciting discounts and coupons.

Only 0$ for 5-10pcs PCB Prototypes Nextpcb.com/pcbprototype

Register and get $100 from NextPCB: Nextpcb.com/coupon

Supplies:

- DFRobot Beginner Kit for Arduino

Or

- Arduino Uno
- LED Kit

- Push Button Kit
- Potentiometer Kit
- Servo Motors
- Buzzer
- Soil Moisture Sensor
- I2C LCD
- Force Sensor

To get high-quality electronic components at an extremely low cost, hqonline.com is the perfect site. They have labels from all the international brands, so you can find the best components for your device without spending a lot of money. This offers you the best value for your money.

Step 1: About Arduino Uno

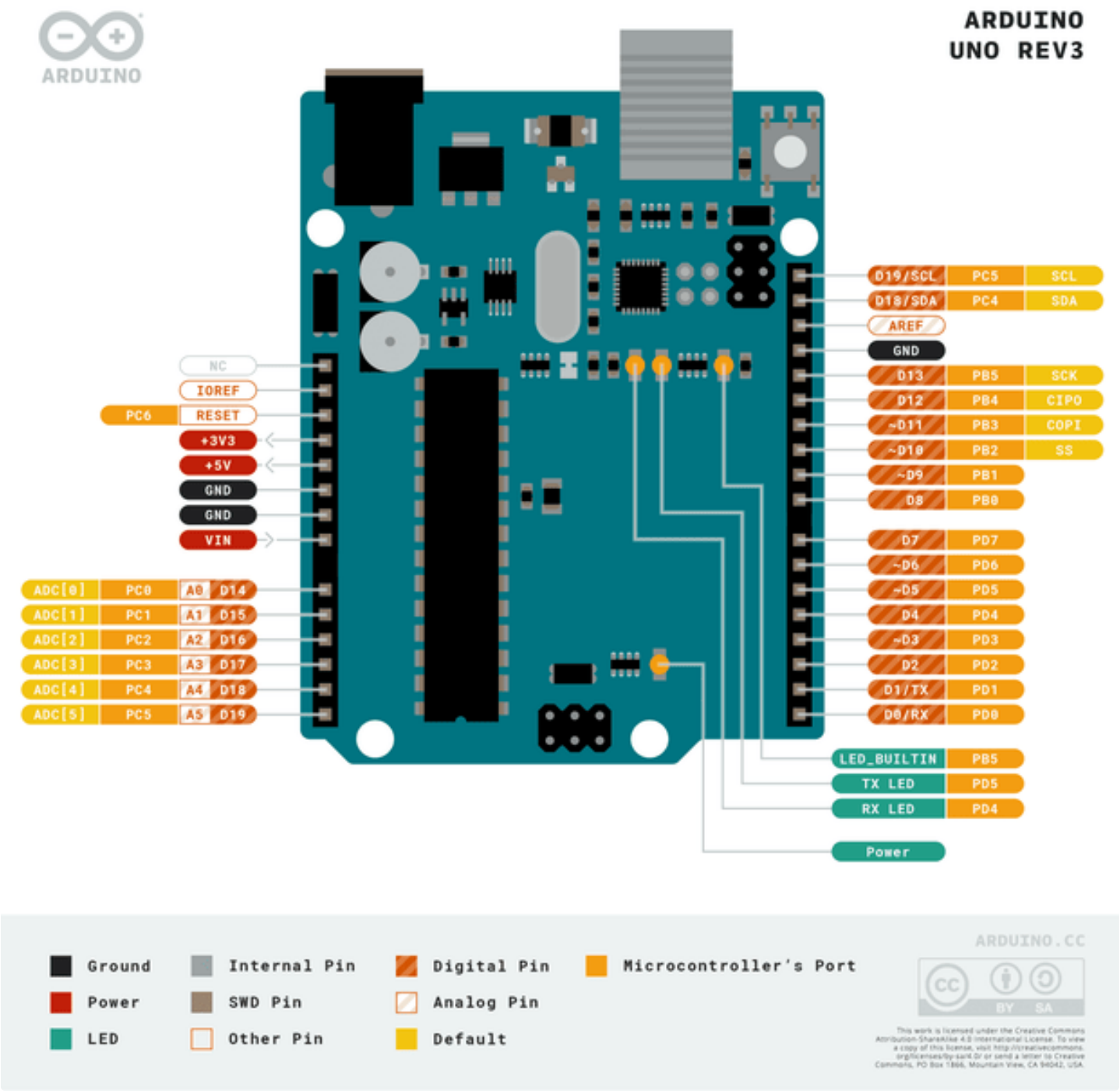
Arduino UNO is a microcontroller board based on the ATmega328P microcontroller. It is one of the most popular boards in the Arduino family, known for its simplicity and versatility, and is widely used by makers, hobbyists, and professionals for a wide range of projects.

The Arduino UNO board has a range of features that make it ideal for prototyping and experimentation. It has 14 digital input/output pins, 6 analog input pins, a 16 MHz quartz crystal, a USB connection for programming and power, and a power jack. It also has a reset button, a power LED, and a built-in LED that can be used for testing and debugging.

The board can be programmed using the Arduino IDE, a software development environment that makes it easy to write and upload code to the board. The Arduino IDE is based on the C++ programming language and has a library of pre-written code, known as "sketches," that can be easily modified and adapted for different projects.

Some common applications of the Arduino UNO include robotics, home automation, sensor networks, and interactive art installations. It is also used in educational settings to teach programming and electronics to students of all ages.

Overall, the Arduino UNO is a powerful and flexible microcontroller board that is well-suited for a wide range of projects and applications. Its simplicity and ease of use make it an ideal choice for beginners and experts alike.



Step 2: About Autodesk Tinkercad Circuit

Autodesk Tinkercad Circuit is a powerful online tool for designing and simulating electronic circuits. It is part of the Tinkercad suite of tools that allows users to create 3D designs, circuits, and code all in one platform. Tinkercad Circuit is ideal for beginners who want to learn electronics and programming, as well as professionals who want to quickly prototype and test their designs before building them in the real world.

Tinkercad Circuit provides a simple drag-and-drop interface for designing circuits. It has a library of electronic components such as resistors, capacitors, LEDs, and microcontrollers that can be easily added to the circuit. The components can be connected together using virtual wires to create a complete circuit. The tool also provides a

simulation feature that allows users to test their circuit before building it in the real world. The simulation feature can be used to measure voltage, current, and resistance, and it can also be used to troubleshoot problems with the circuit.

In addition to designing circuits, Tinkercad Circuit also provides a code editor that allows users to program their circuits. It supports programming languages such as C++, JavaScript, and Python. Users can write code to control their circuits and create interactive projects such as robots, alarms, and other devices. The code editor also includes a built-in serial monitor that allows users to send and receive data from their circuit.

Tinkercad Circuit is an excellent tool for educational use, as it provides a fun and engaging way for students to learn about electronics and programming. It has a range of lesson plans and tutorials that cover topics such as circuit design, programming, and robotics. The tool is also ideal for makers who want to quickly prototype and test their designs before building them in the real world.

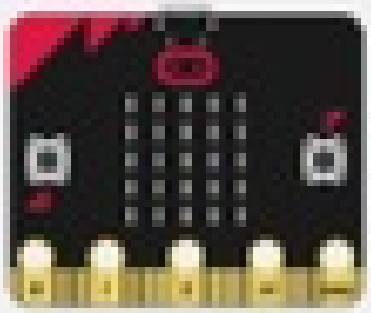
In conclusion, Autodesk Tinkercad Circuit is a powerful online tool for designing and simulating electronic circuits. It provides a user-friendly interface, a library of electronic components, and a simulation feature that allows users to test their circuit before building it in the real world. With its built-in code editor and support for multiple programming languages, Tinkercad Circuit is an ideal platform for both beginners and professionals.

9V Battery

Coin Cell 3V Battery

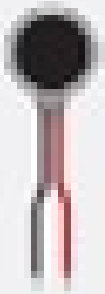
1.5V Battery

Breadboard Small



micro:bit

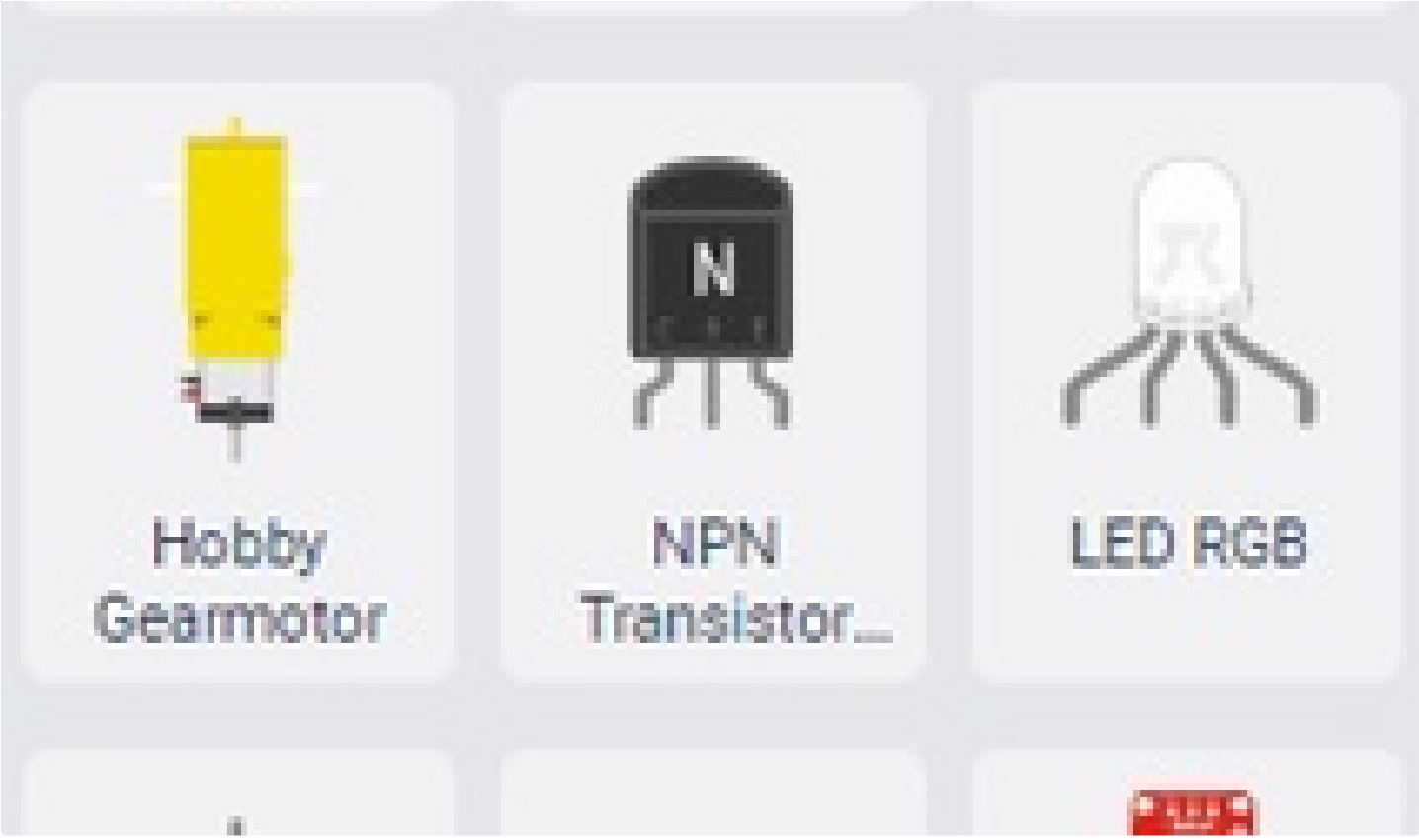
Arduino Uno R3



Vibration Motor

DC Motor

Micro Servo



Step 3: Analog and Digital Signals

Analog signals and digital signals are two types of signals used to transmit information in electronic communication systems.

Analog signals are continuous signals that represent physical quantities such as sound, light, and temperature. An analog signal can take on any value within a range of values, and it changes smoothly over time. An example of an analog signal is a human voice when speaking into a microphone. The microphone converts the sound waves into an electrical signal that varies in amplitude and frequency, which can then be transmitted through a wire or over the airwaves.

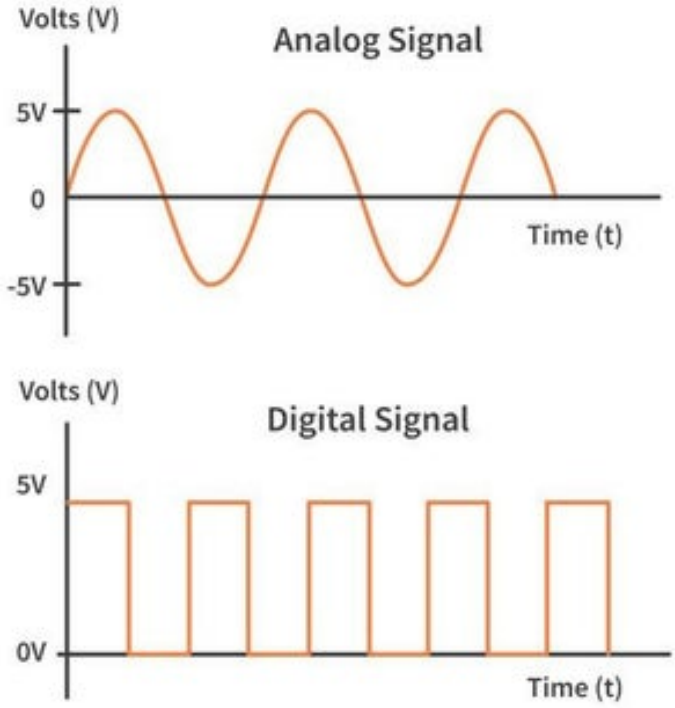
In Arduino we have 10 bit analog, analog signals are represented using analog inputs, which are capable of reading voltage values from 0 to 5 volts. These analog inputs can be used to read analog signals from sensors such as temperature sensors, light sensors, and accelerometers. For example, a temperature sensor may output an analog voltage signal that corresponds to the temperature of its environment, and the Arduino can use its analog input to read this signal and convert it into a digital value that can be processed by the microcontroller.

On the other hand, digital signals are discrete signals that represent information using a sequence of 0s and 1s. Digital signals are used to represent numerical values, text, images, and other types of data. Unlike analog signals, digital signals can only take on specific values, and they change abruptly in discrete steps. An example of a digital signal is a computer keyboard. When a key is pressed, it sends a digital signal to the computer that represents the corresponding character.

Digital signals in Arduino are represented using digital inputs and outputs. Digital inputs are used to read the state of a switch or a button, while digital outputs are used to control the state of a LED, a motor, or any other actuator. For example, a button can be connected to a digital input of the Arduino, and when the button is pressed, the digital input

will read a high value (1), indicating that the button is pressed. Similarly, a LED can be connected to a digital output of the Arduino, and when the output is set to a high value (1), the LED will turn on, and when it is set to a low value (0), the LED will turn off.

In Arduino, analog signals are read using analog inputs(Pins A0 to A5) and digital signals are read using digital inputs(Pins D0 to D19) and controlled using digital outputs. These signals are used to interact with the physical world and control various devices and sensors.



Step 4: LED Blink

- In this step, we are going to make a LED blink(ON and OFF) with 0.5Sec(500 Mili Seconds) interval.

- To add the components we just need to drag and drop the components on the work area as shown.
- Drag an Arduino Uno and a LED from the components list, you can see that we have lot many components in this list to play with, we are going to use few of this further.
- Now connect Anode of the LED to Pin 2 of Arduino and Cathode of the LED to GND of Arduino.
- I personally represent the GND/-Ve with black colored wire, 5V/3V/+Ve with Red and other signal wires with multiple colors.
- Now you have successfully made your first Electric Circuit Connection, Now let's code the Arduino

- Click on Code on the Top-Right side here you will find different blocks which you can drag and drop to build your logic.
- In Arduino, we have two main sections i.e on start/void setup() and forever/void loop()

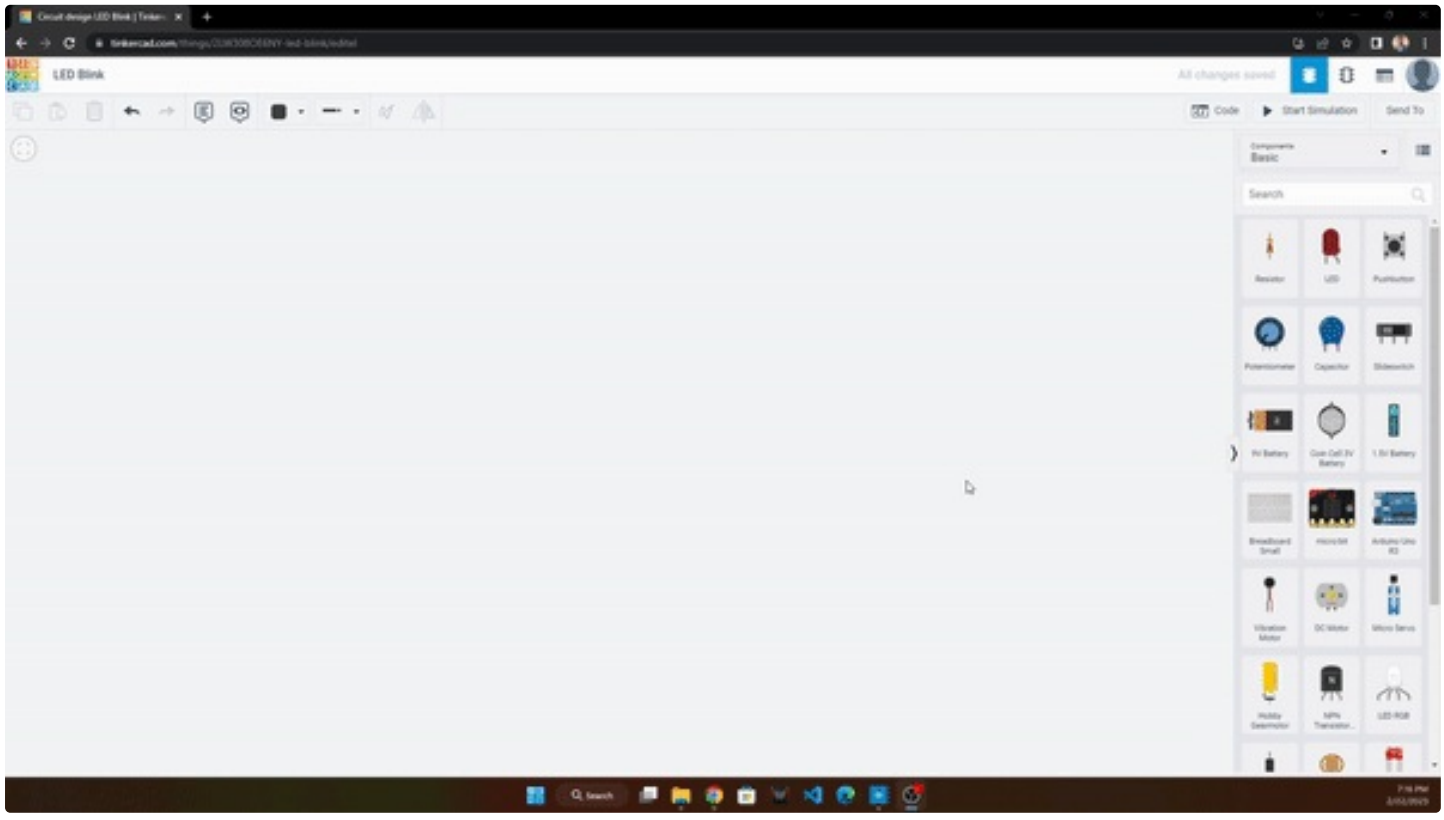
- on start/void setup() - in this section whatever the logic we write will execute only once during the system boot(Power On).
- forever/void loop() - in this whatever the logic we write will be executed again and again repeatedly unless the power is off.

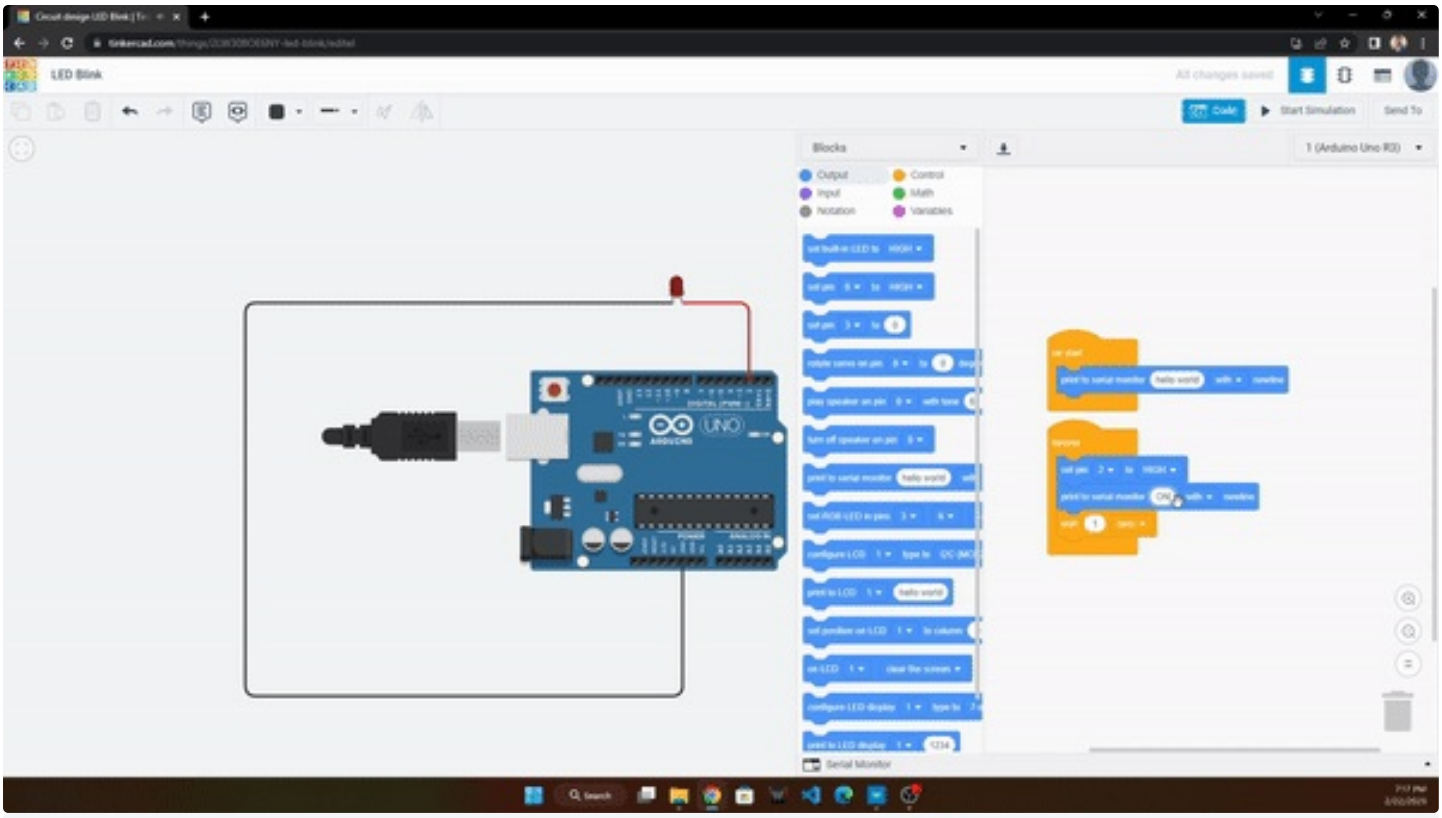
- Now in the on start I have dragged a print to serial monitor block, with this block we can print whatever the messages we want on our PCs screen via a serial monitor, you can see at the bottom we have Serial Monitor, we will see how it works.
- In forever I have dragged the set pin block and set the pin 2 to Hight following it I have added a serial monitor message block that says that the LED is ON.
- Then I added a Wait block which makes the flow(loop) wait for 0.5 seconds, and then set pin 2 to low followed by OFF message and wait for 0.5 seconds before turning LED ON.

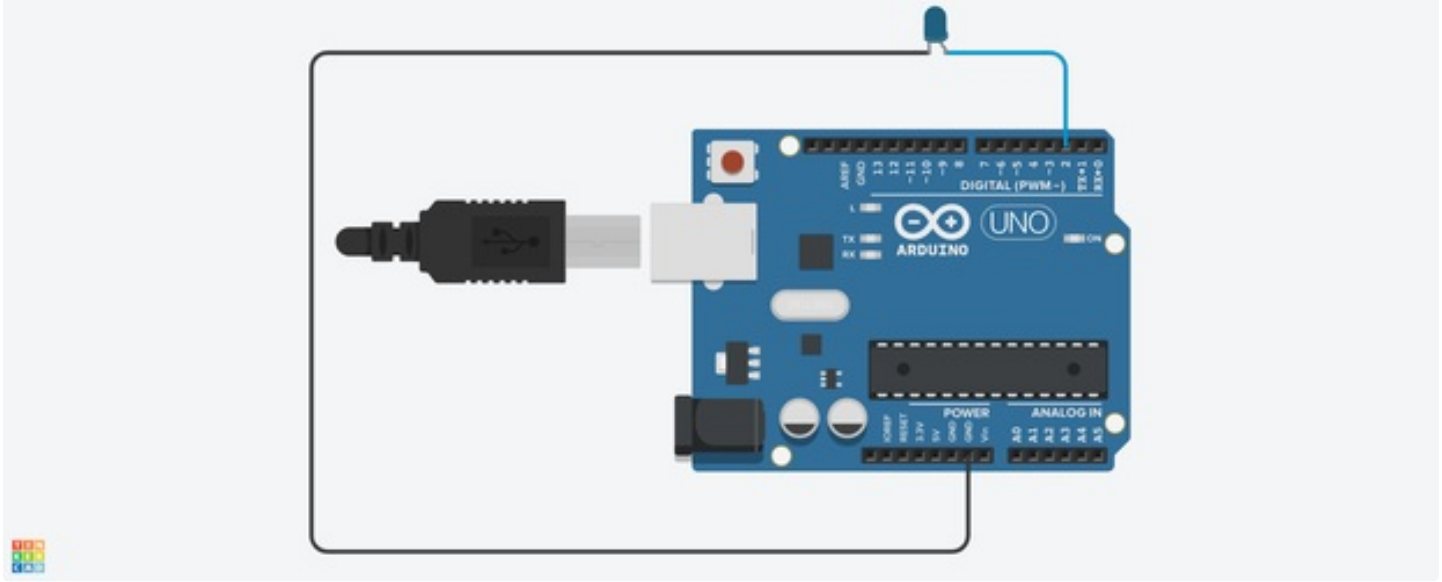
- This code will first Print a messaage hello world(on start) and enters the forever loop.
- In forever loop first, Turn On the LED, print the message ON on the serial monitor wait for 0.5 sec(500 ms), and then turn off the LED, print OFF message and wait for 0.5 sec and the cycle gets repeated.

Click to see My LED Blink

Note: In every section, I have added screen recorded demo of every demonstration in the form of .gif(Low quality) and .mov(HD Quality) you can refer them.




LED Blink
Blocks
Output
Control
Input
Math
Notation
Variables
Serial Monitor


DIGITAL (PWM~)
UNO
ARDUINO
POWER
ANALOG IN

Download

https://www.instructables.com/FUM/CKVX/LEJZ4BC7/FUMCKVXLEJZ4BC7.mov

Step 5: Traffic Signal

In this step, we are going to make a traffic light containing Red, Orange, and Green lights. The traffic light stays in the Red and Green states for 5 sec and takes 1 sec to change from Red to Green i.e Orange state.

- First, drag all the components as shown, but before that do you know how to use a Breadboard? if not

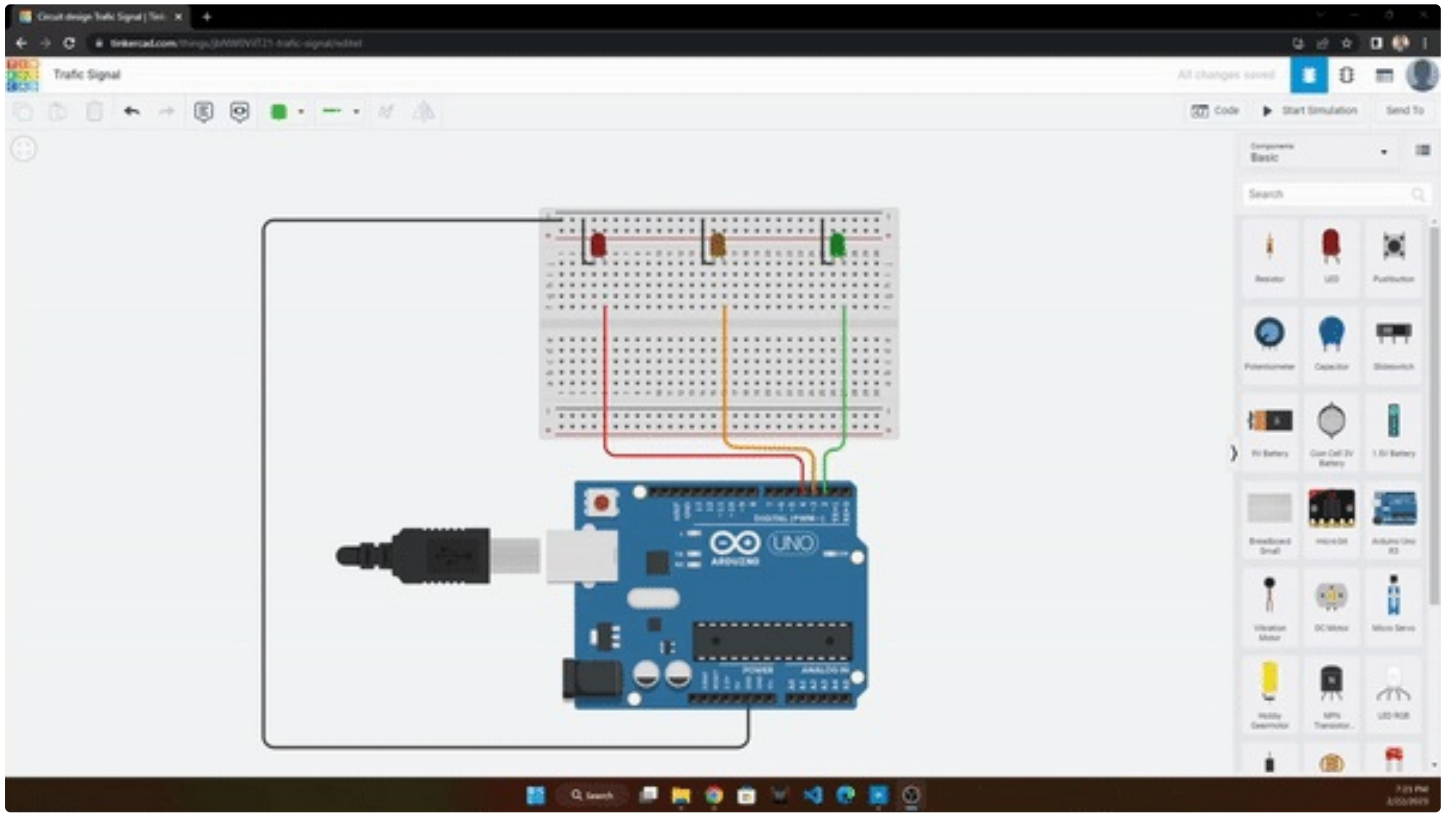
click to check these Instructables.

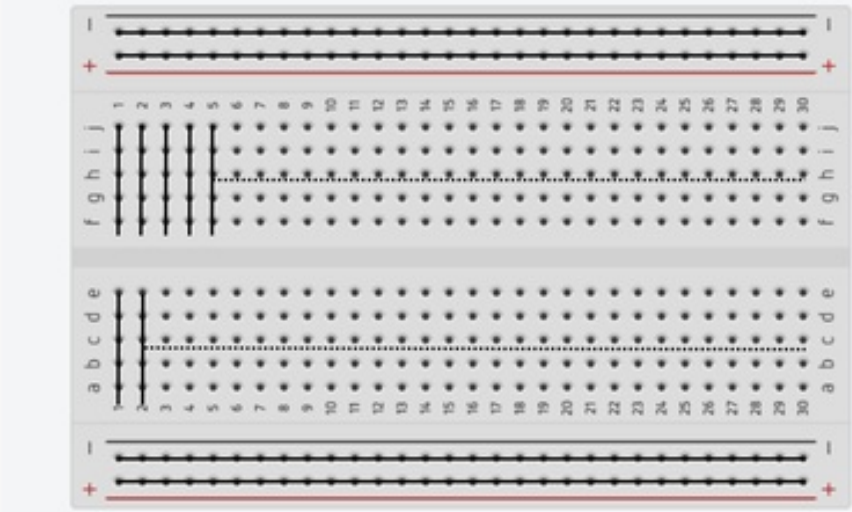
- Now click on the LEDs you will find a small pop-up window, here you can change the color of the LEDs, change the colors as per the demonstration.
- Connect
- Red LED to Pin 4 (Anode of LEDs)
- Orange LED to Pin 3
- Green to Pin 2 and
- All cathodes to GND

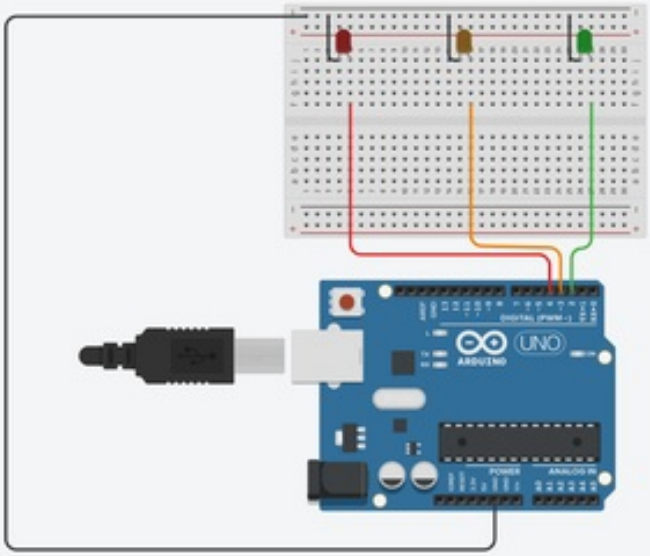
Now let's code it

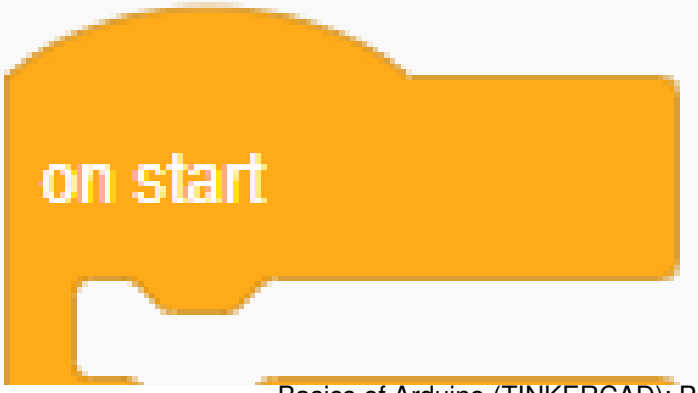
- I have not used on start block but you can if you want to print any message.
- In the forever logic as you can see I have set different LEDs as per the demonstration statement
- First Turn on Red Light(Pin 4 - High)
- Wait for 5 sec
- Turn Off Red Light(Pin 4 - Low)
- Turn On Orange LED(Pin 3 - High)
- wait for 1 sec
- Turn Off Orange LED(Pin 3 - Low)
- Turn On Green LED(Pin 2 - High)
- Wait for 5sec
- Turn Off Green LED(Pin 2 - Low)
- and repeat .....

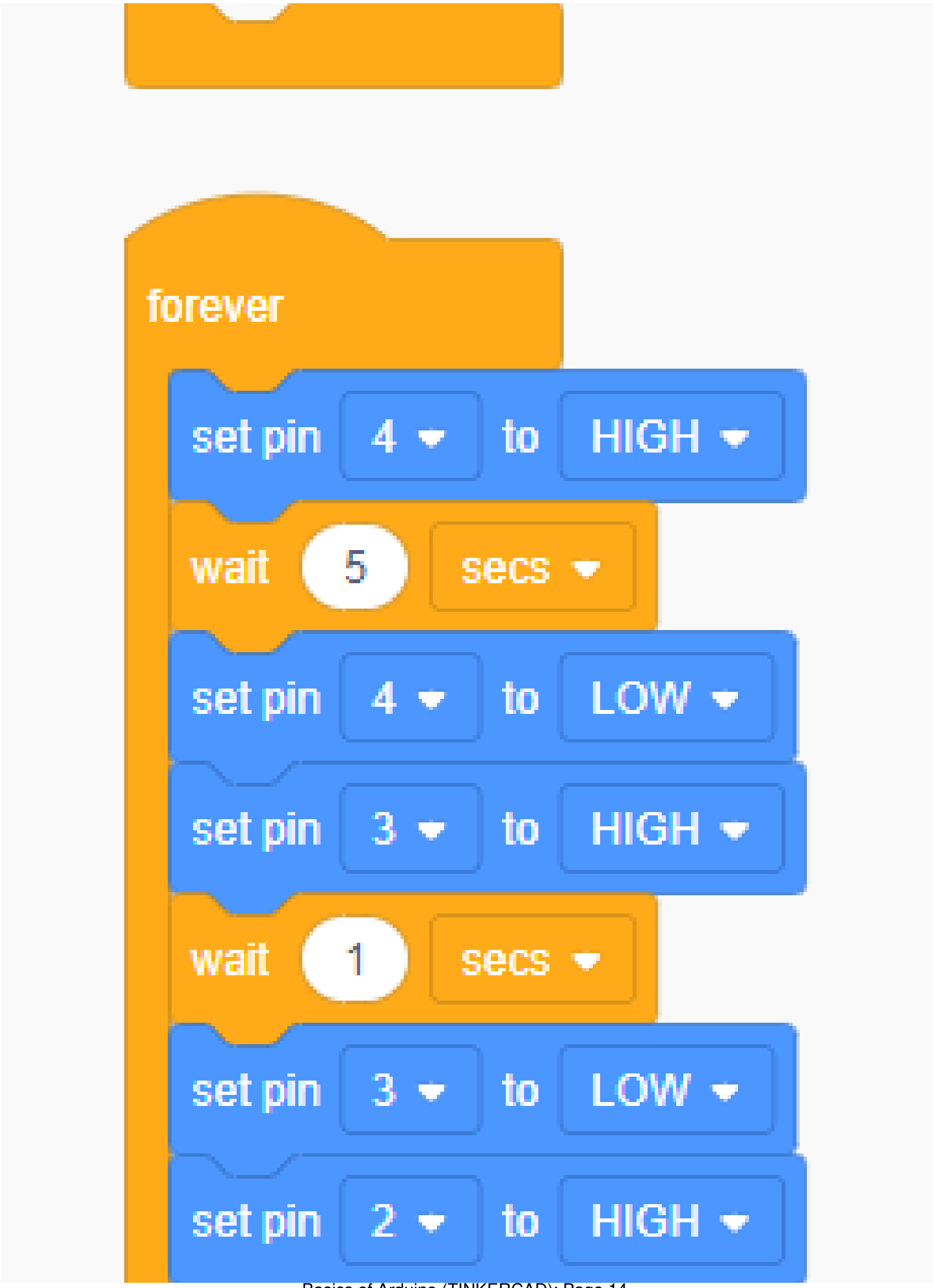
Click to view My Traffic-Signal

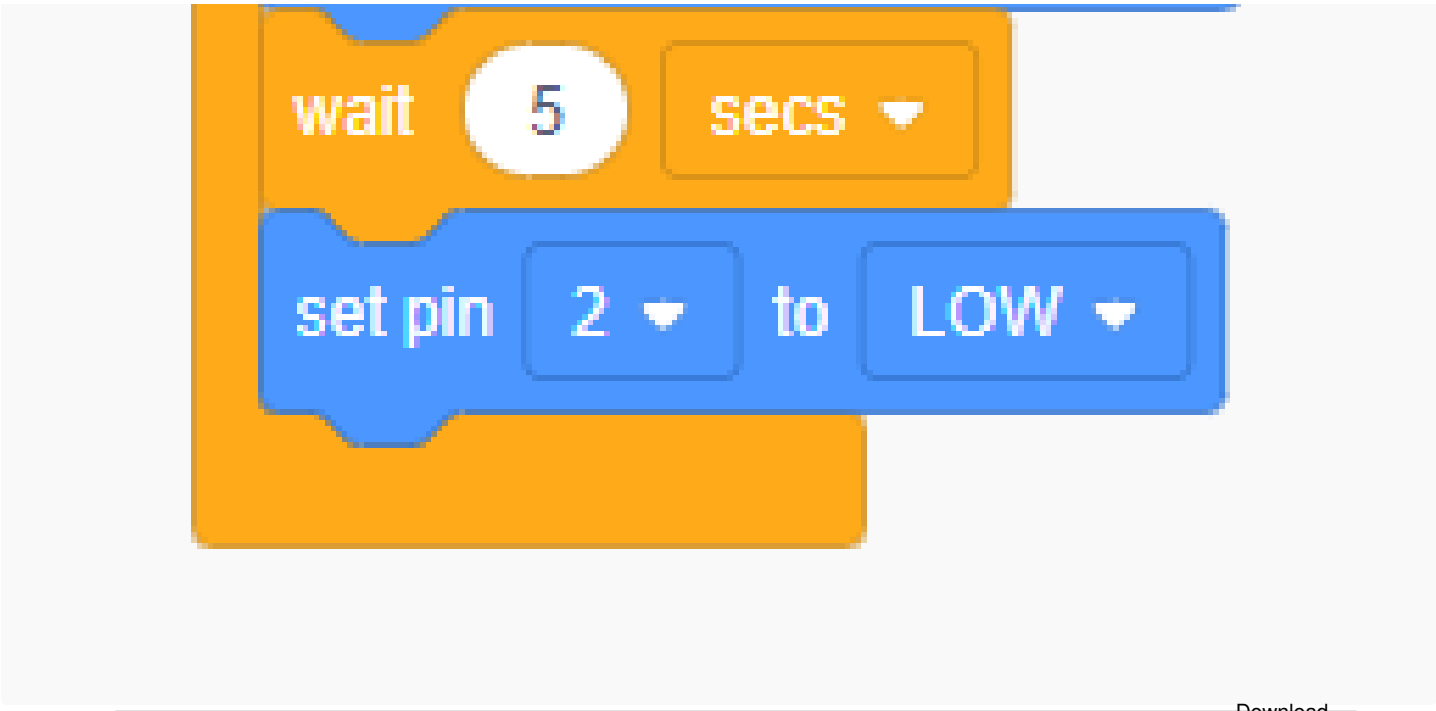







on start


forever
set pin 4 to HIGH
wait 5 secs
set pin 4 to LOW
set pin 3 to HIGH
wait 1 secs
set pin 3 to LOW
set pin 2 to HIGH



Download

https://www.instructables.com/FA8/5OKH/LEJZ4BEF/FA85OKHLEJZ4BEF.mov

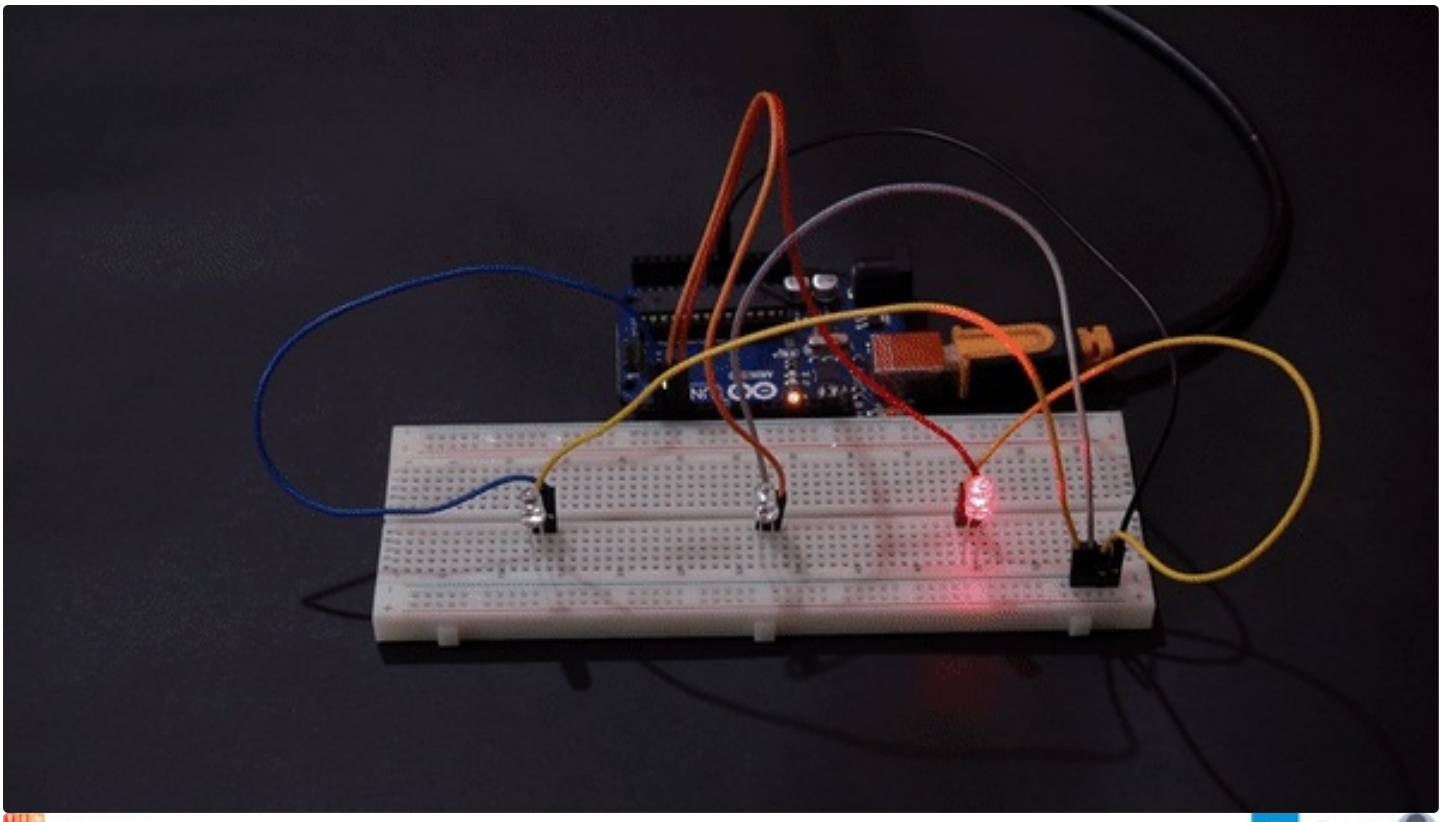
Step 6: Use Physical Hardware

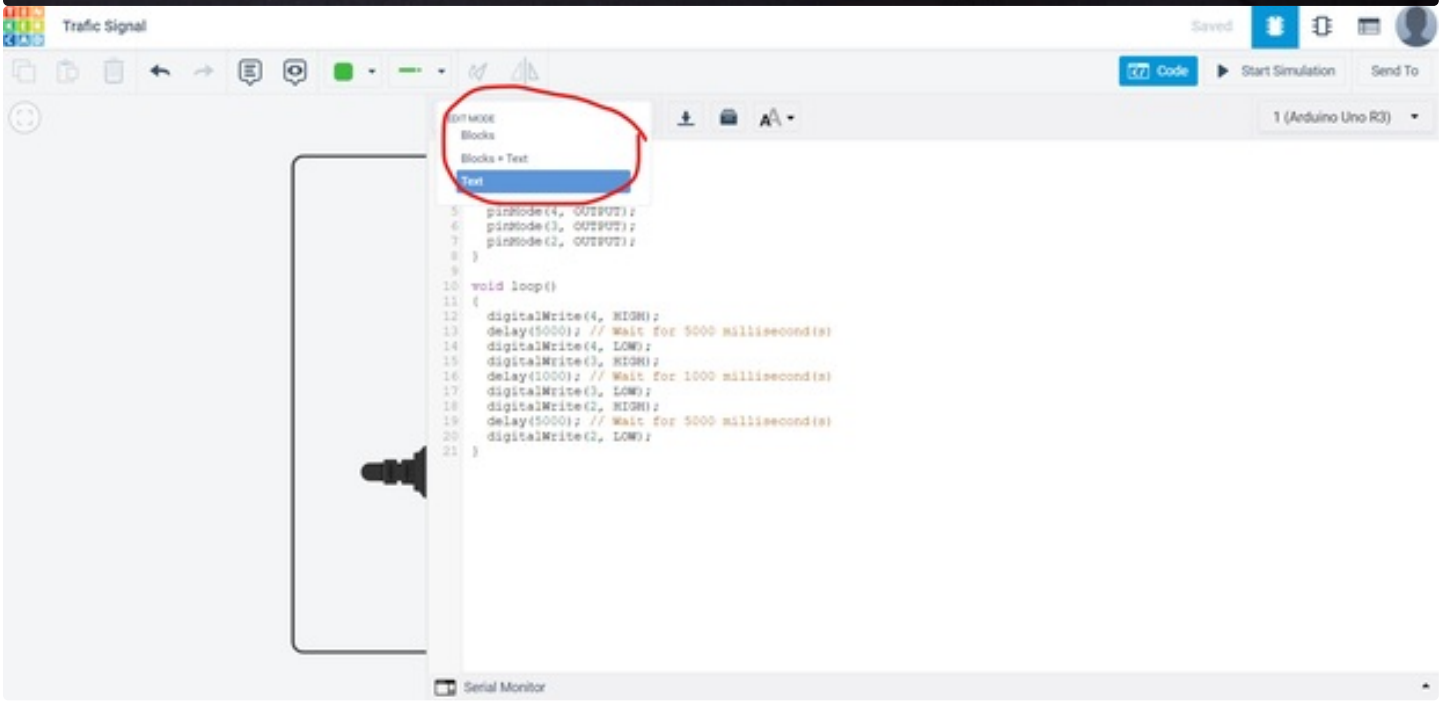
- Now we will use physical hardware and upload the code by downloading it from Tinkercad.
- You can get the components used in these demonstrations from the above supplies.
- Connect the components as per shown in step 5.
- In the codes section, you can also write the Arduino code in C++ by changing the Edit Mode. Tinkercad has the flexibility to build logic with Blocks, or Text code, or hybrid of both Blocks+Text Code.
- When you build logic using block you can get code for that from Text, we can just copy or click on the download icon to download the Arduino code.

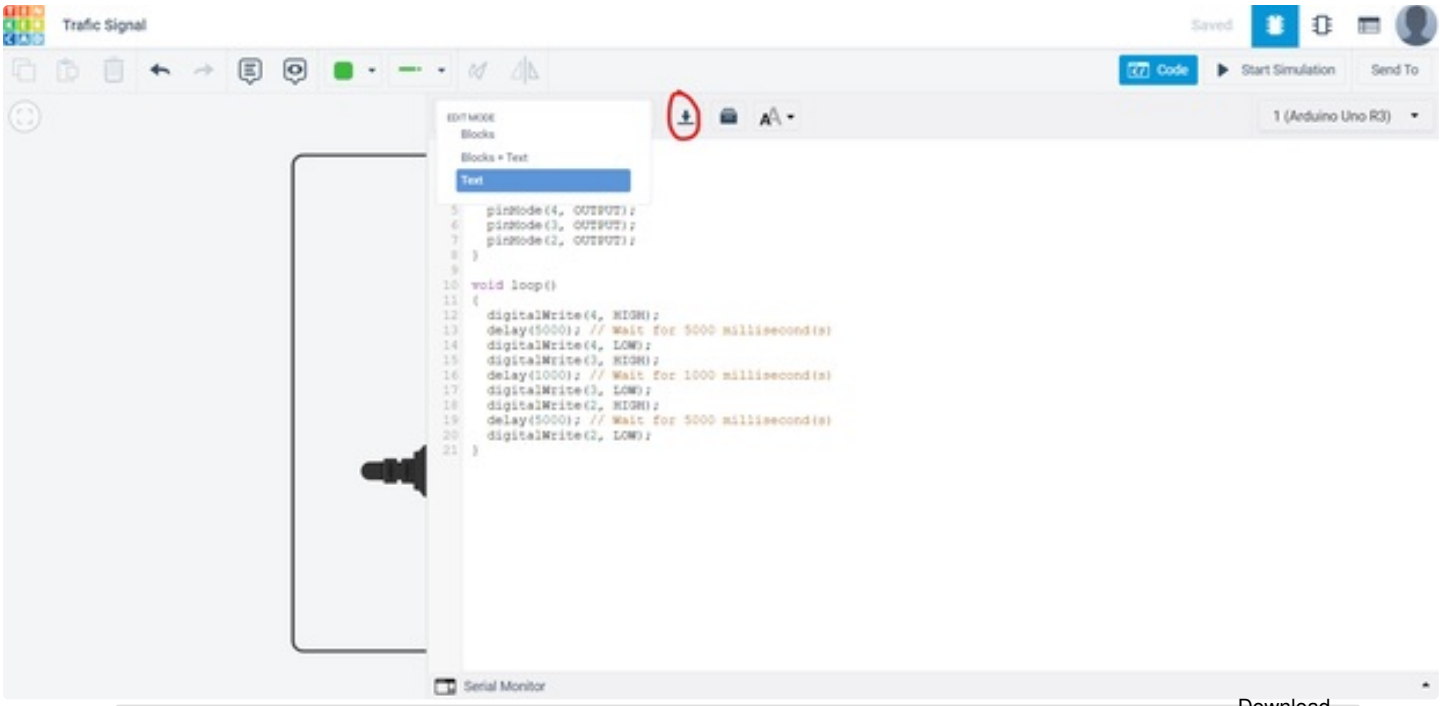
- Now we need Arduino IDE, Click To Download.
- Open the downloaded code in Arduino IDE, Select the Board Type as Arduino Uno and com port as shown in the below video.
- Now you are all set, just click on the upload icon, if you have connected all the connections properly the lights will blink as per step 5.

That's how simple Arduino is!

You can upload any Arduino-compatible code in this way and make your own projects.




Trafic Signal
Saved
Code
Start Simulation
Send To
1 (Arduino Uno R3)
Blocks
Blocks + Text
Text
pinMode(4, OUTPUT);
pinMode(3, OUTPUT);
pinMode(2, OUTPUT);
}
void loop()
{
digitalWrite(4, HIGH);
delay(5000); // Wait for 5000 millisecond(s)
digitalWrite(4, LOW);
digitalWrite(3, HIGH);
delay(1000); // Wait for 1000 millisecond(s)
digitalWrite(3, LOW);
digitalWrite(2, HIGH);
delay(5000); // Wait for 5000 millisecond(s)
digitalWrite(2, LOW);
}
Serial Monitor



https://www.instructables.com/F8Z/JUAQ/LEJZ4BEN/F8ZJUAQLEJZ4BEN.mov Download

https://www.instructables.com/F8Q/Z1OM/LEJZ4BEM/F8QZ1OMLEJZ4BEM.mov Download

Step 7: Push Button

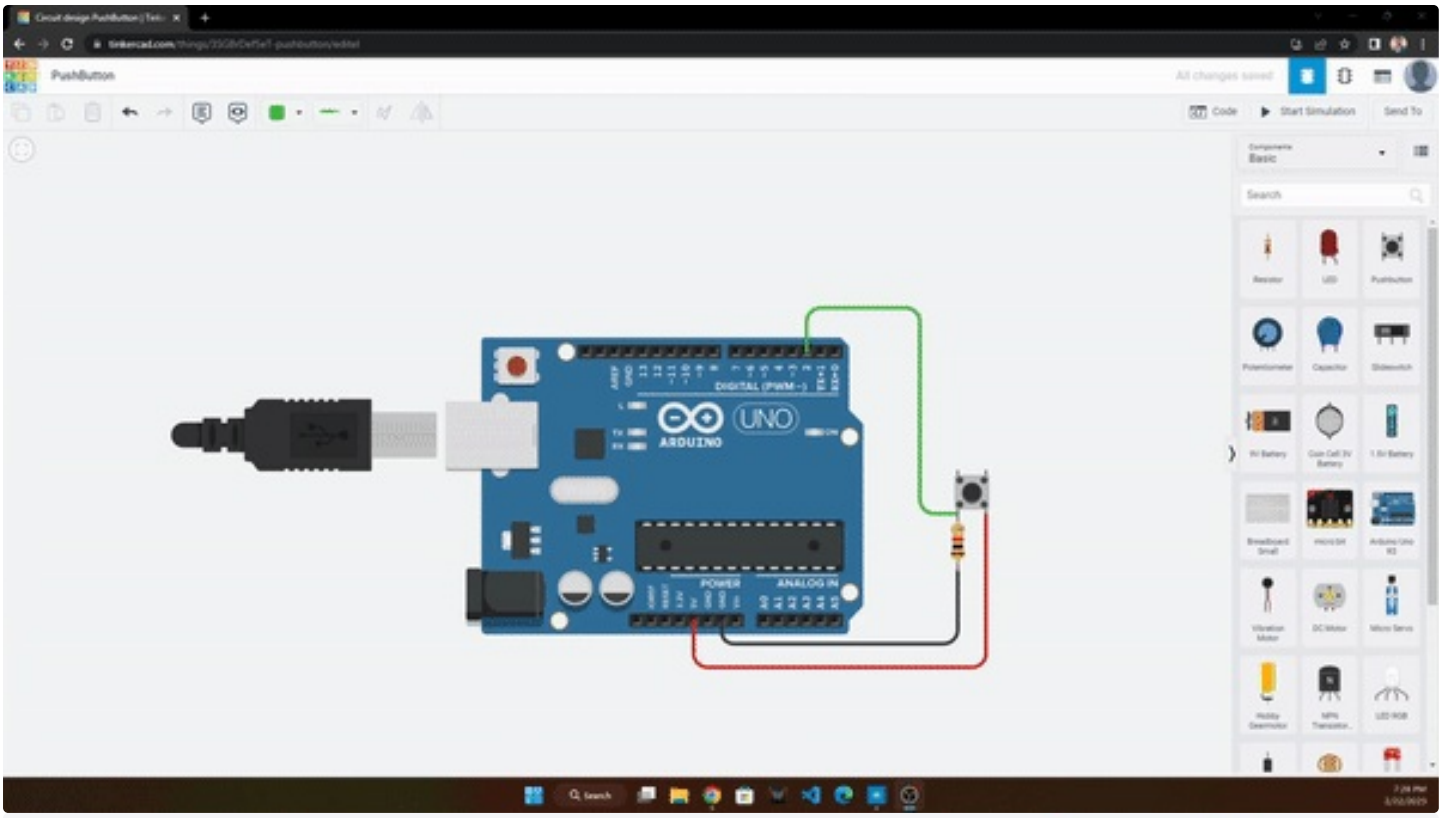
The above demonstration shows how we can use digital output but now we are going to see how we are going to use digital input.

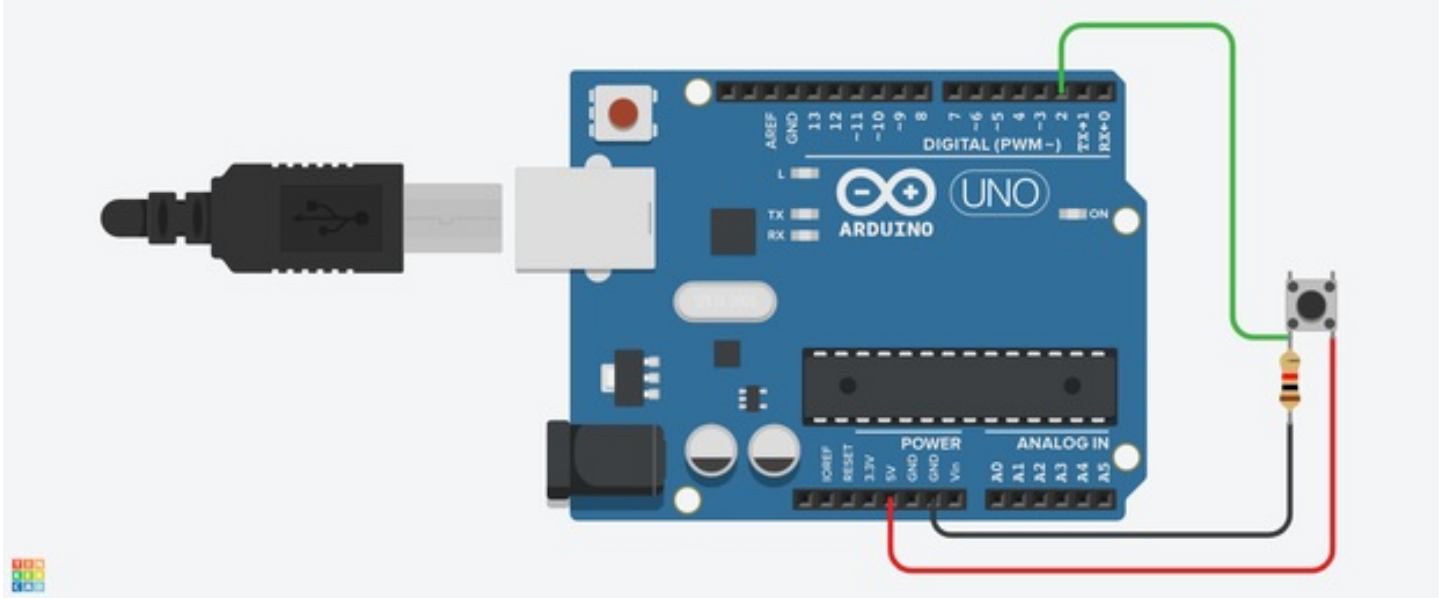
- Connect the push button and Arduino as shown with a resistor of 10KOhm.
- One end of the push button to Pin 2
- Same end with a resistor and GND
- Another end to 5V
- We can change the resistance value by clicking the resistor.
- Here this resistor is going to act as a Pull-Down resistor, confused? Let me explain
- Suppose we press the push button, the input pin goes to High state i.e 5V, when we release it, the input stays on High state, which means always ON/High, that's why we connected 10K resistor between input and GND. Now the input always will be low except when the button is pushed.
- Similarly, we also have a Pull-Up resistor connection there we connect a resistor between the input and 5V.

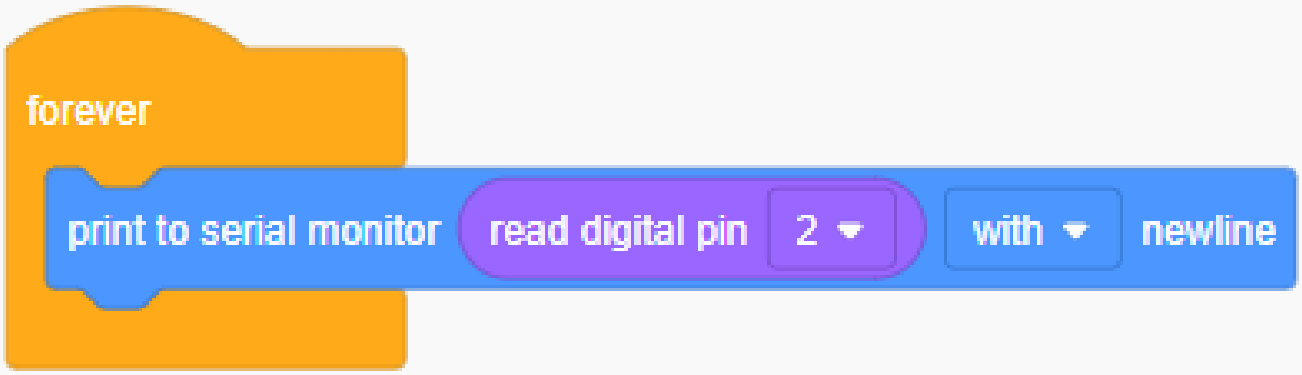
Block Code

- In the forever section as you can see I have taken a read digital pin block, set it to read pin 2 and print the value on Serial Monitor.

Click to see my Pushbutton Demo


PushButton
Code
Start Simulation
Send To
ARDUINO
UNO
DIGITAL (PWM~)
POWER
ANALOG IN


ARDUINO
UNO
DIGITAL (PWM~)
POWER
ANALOG IN


forever
print to serial monitor
read digital pin
2
with
newline

Step 8: 4-Button Piano

In this step we are going to make a 4-key piano, here we will take 4 push buttons and a buzzer. When we press any one key(Push button) we should get a tune that we have set to that key, different keys will have a different tune.

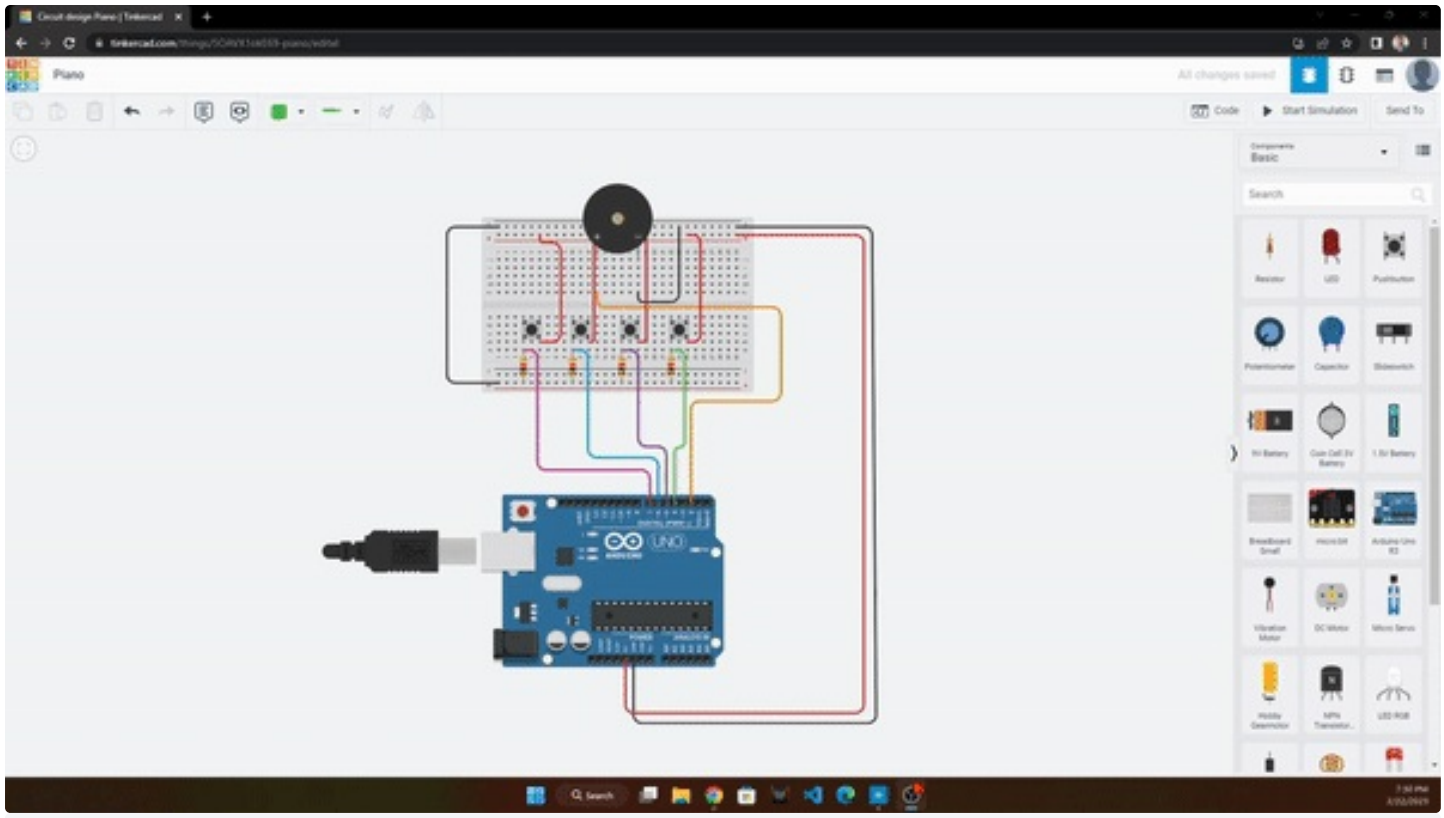
Connect the components as shown

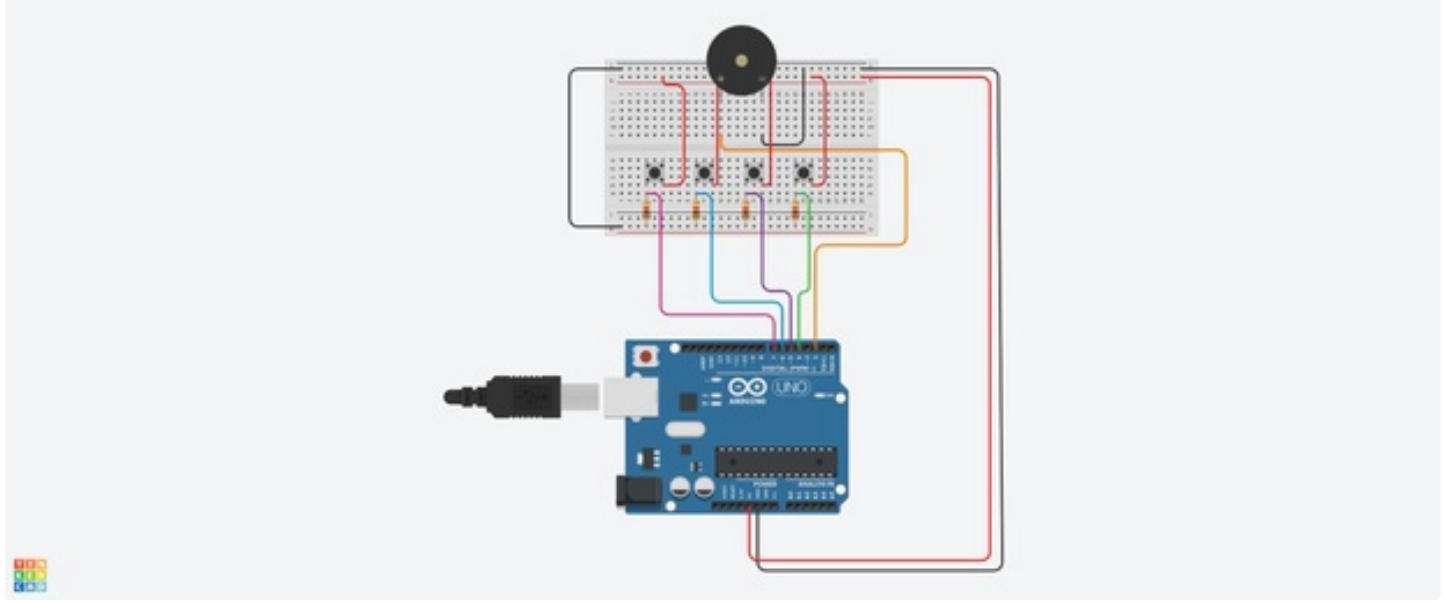
- Buzzer to Pin 2
- Push Button 1 to Pin 4
- Push Button 2 to Pin 5
- Push Button 3 to Pin 6
- Push Button 4 to Pin 7
- GND, Resistors and 5V

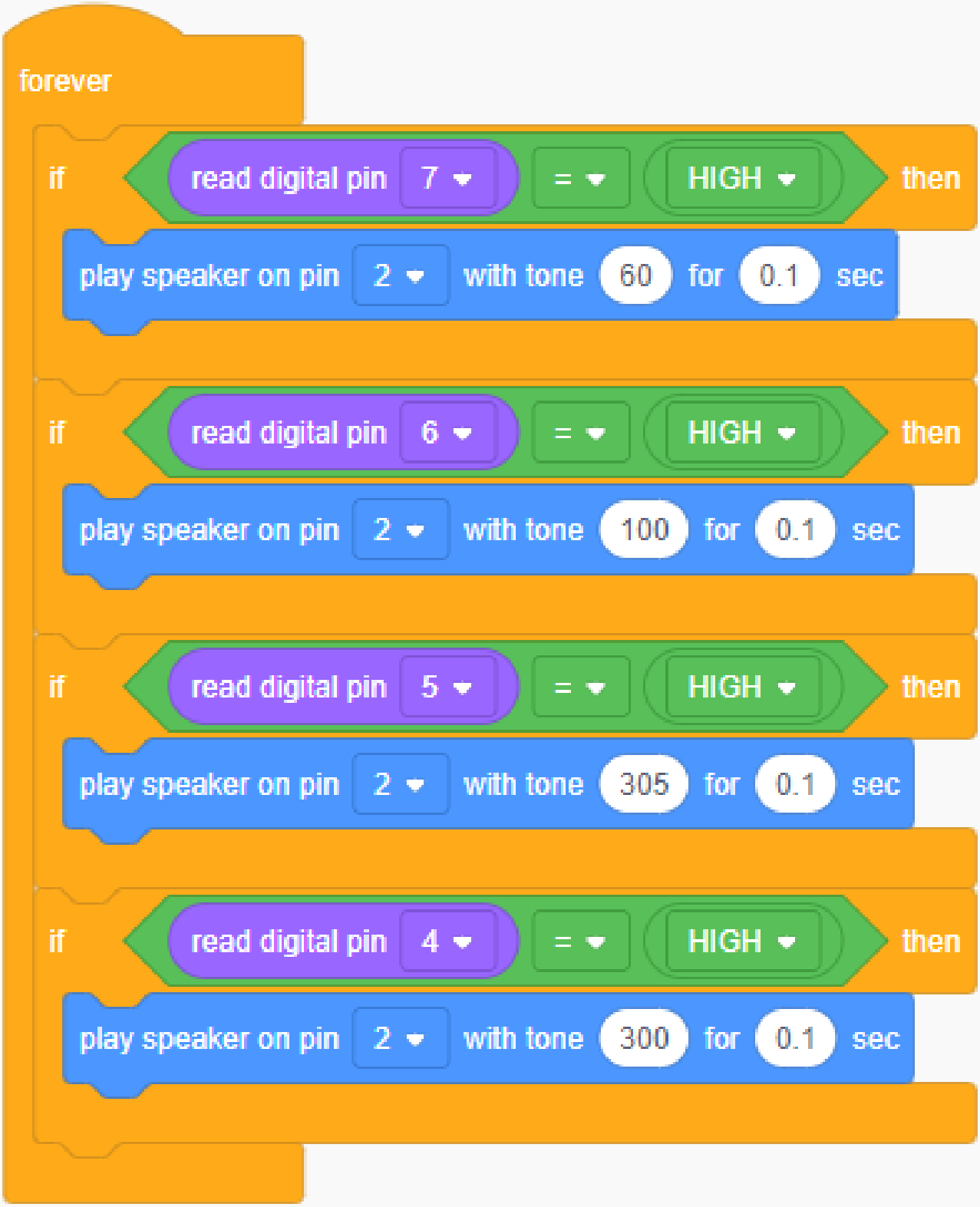
code

- In the Blocks, I am comparing the digital inputs of each pin 4, 5, 6, and 7 using if blocks, if the pin is High then the play speaker on pin block plays a specific set tune.
- You can change the tone value and duration.

Click to see my Piano Demo


Piano
Code
Start Simulation
Send To
Basic
Search





Download

https://www.instructables.com/FSI/64G3/LEJZ4BHO/FSI64G3LEJZ4BHO.mov

Step 9: Variable Potentiometer

Now let's see how we can use Analog input, for this demonstration we are going to use a variable resistance potentiometer.

Connect the potentiometer to Arduino as shown

- Middle terminal to Pin A0
- Left terminal to GND
- Right terminal to 5V

Code

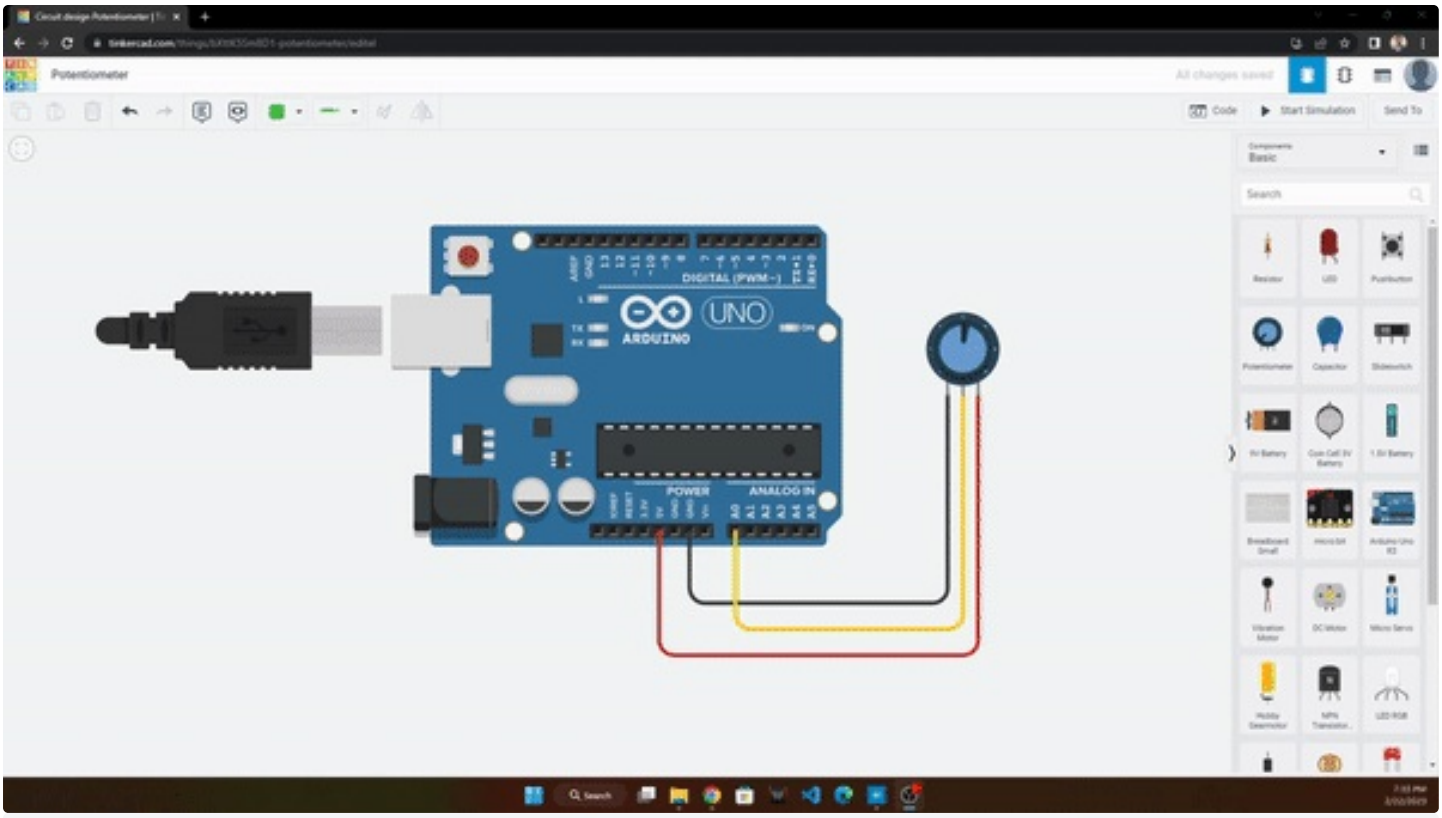
- In the forever section drag and drop the Analog read pin, and set it to A0, this block is placed inside the print serial monitor block.
- These will continuously read the analog signal from pin A0 and print its value on the serial monitor.

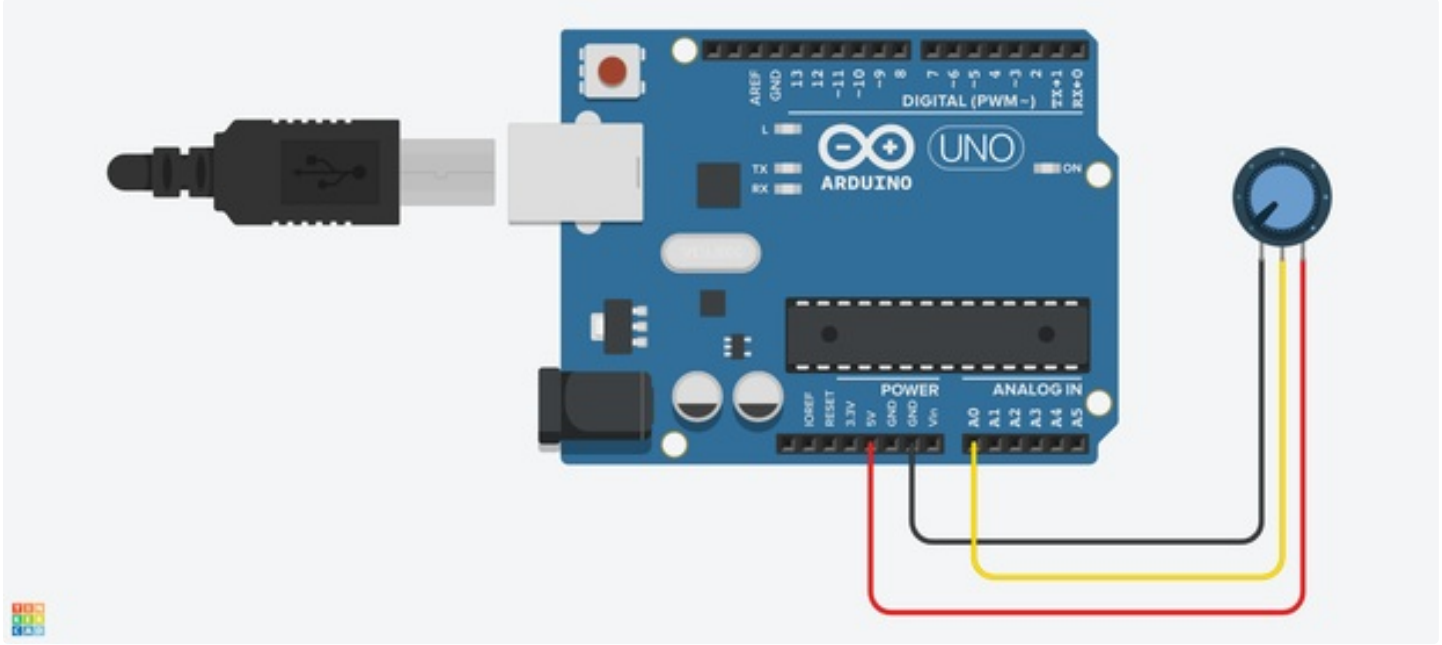
Start simulating, as you can see in the serial monitor as we rotate the knob the value changes between 0 to 1023.

Can you answer why the max value is 1023??

It's because Arduino has 10-bit Analog, 2 to the power 10 = 1024.

Click to see My Potentiometer Demo


Potentiometer
Code
Start Simulation
Send To
DIGITAL (PWM~)
UNO
ARDUINO
POWER
ANALOG IN


AREF
GND
DIGITAL (PWM~)
UNO
ARDUINO
TX
RX
ON
POWER
ANALOG IN
IOREF
RESET
3.3V
5V
GND
GND
Vin
A0
A1
A2
A3
A4
A5

Download

https://www.instructables.com/F6R/ZEMO/LEJZ4BJ6/F6RZEMOLEJZ4BJ6.mov

Step 10: RGB Lamp

In this step, we are going to make a Multicolor RGB Lamp.

Let's use 3 potentiometers to change the value of Red, Green, and Blue and a Multicolor LED.

Connect the circuit diagram as shown

- LED cathode to GND
- LED Red to 11
- Led Green to 10
- Led Blue to 9
- Potentiometer 1 t0 A0
- Potentiometer 2 t0 A1
- Potentiometer 3 t0 A2

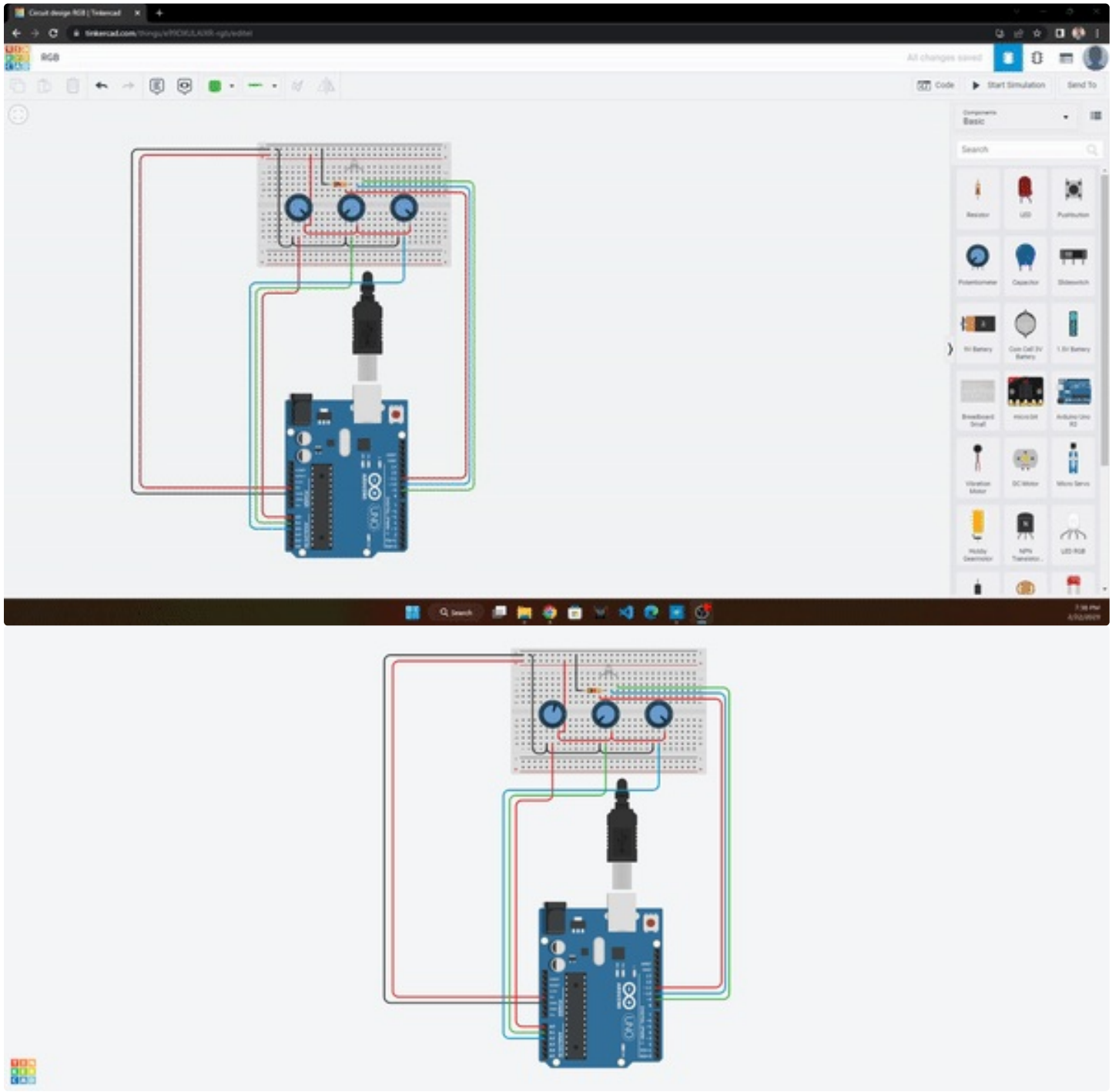
Code

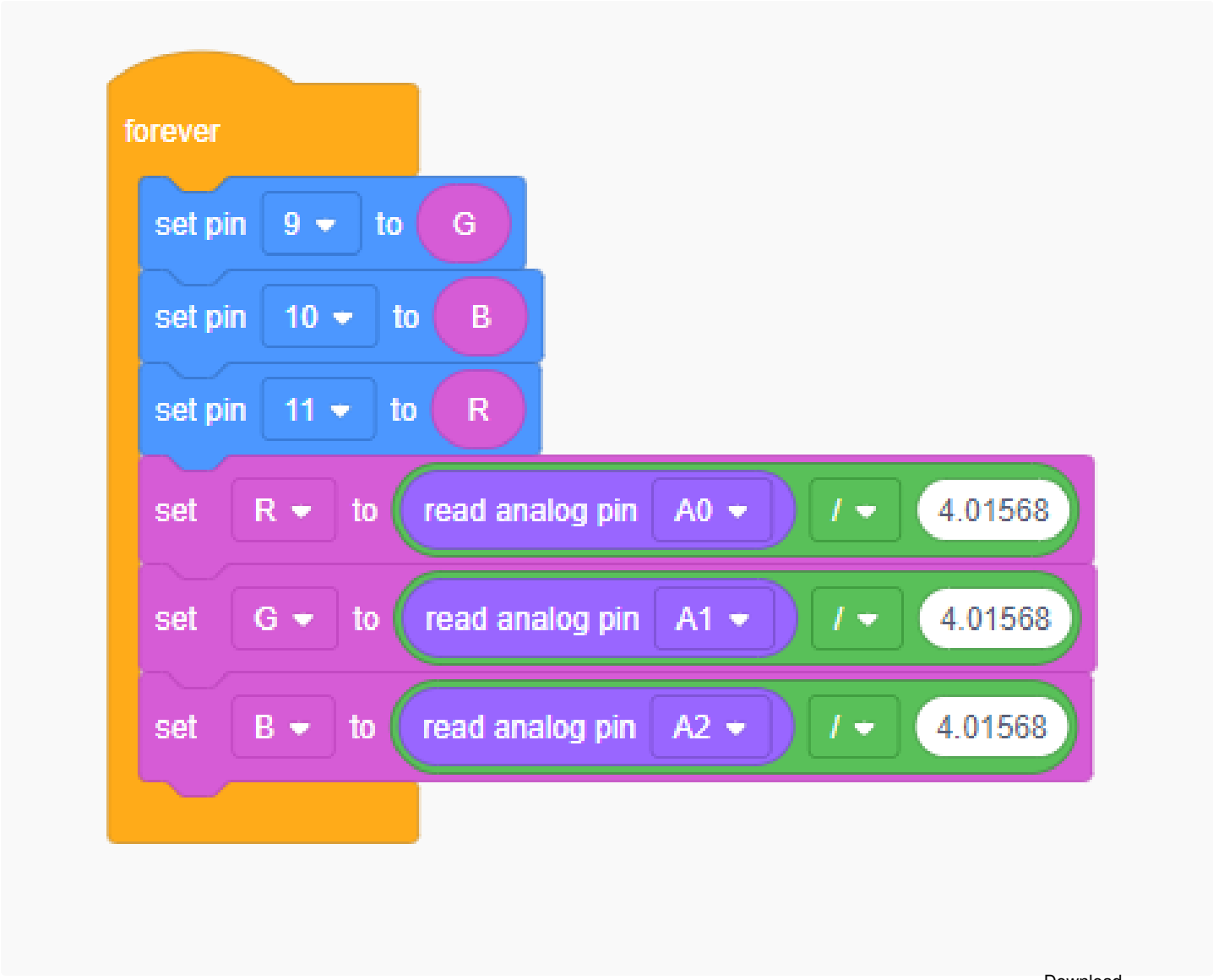
- First create 3 variables named R, G, and B or Red, Green, and Blue
- Then drag the set pin and set the corresponding pins 9, 10, and 11 aligned with B/Blue, G/Green, and R/Red.
-
- In the block as you can see we are setting the values of the pins to the respective variables.
- reading the analog signal of Red, Green, and Blue - Potentiometers and setting(Storing) them to R, G, and B variables.
-
- But as you can see before storing the variable value I used mathematic block division - /.
- Since we get an analog value between 0 - 1024, but the RGB value must be between 0-255

- i.e R, G, B = 0-255, 0-255, 0-255
-
- So I have divided the analog value with the coefficient 4(from 1024/256).

Start simulating and rotating the different knobs, you will see a different color on different combinations of RGB knobs.

Click to see My RGB Lamp





Download

https://www.instructables.com/F3W/7VKC/LEJZ4BKP/F3W7VKCLEJZ4BKP.mov

Step 11: Force Sensor

In this demonstration, we are going to connect a force sensor with Arduino.

As shown drag and drop all the components and connect

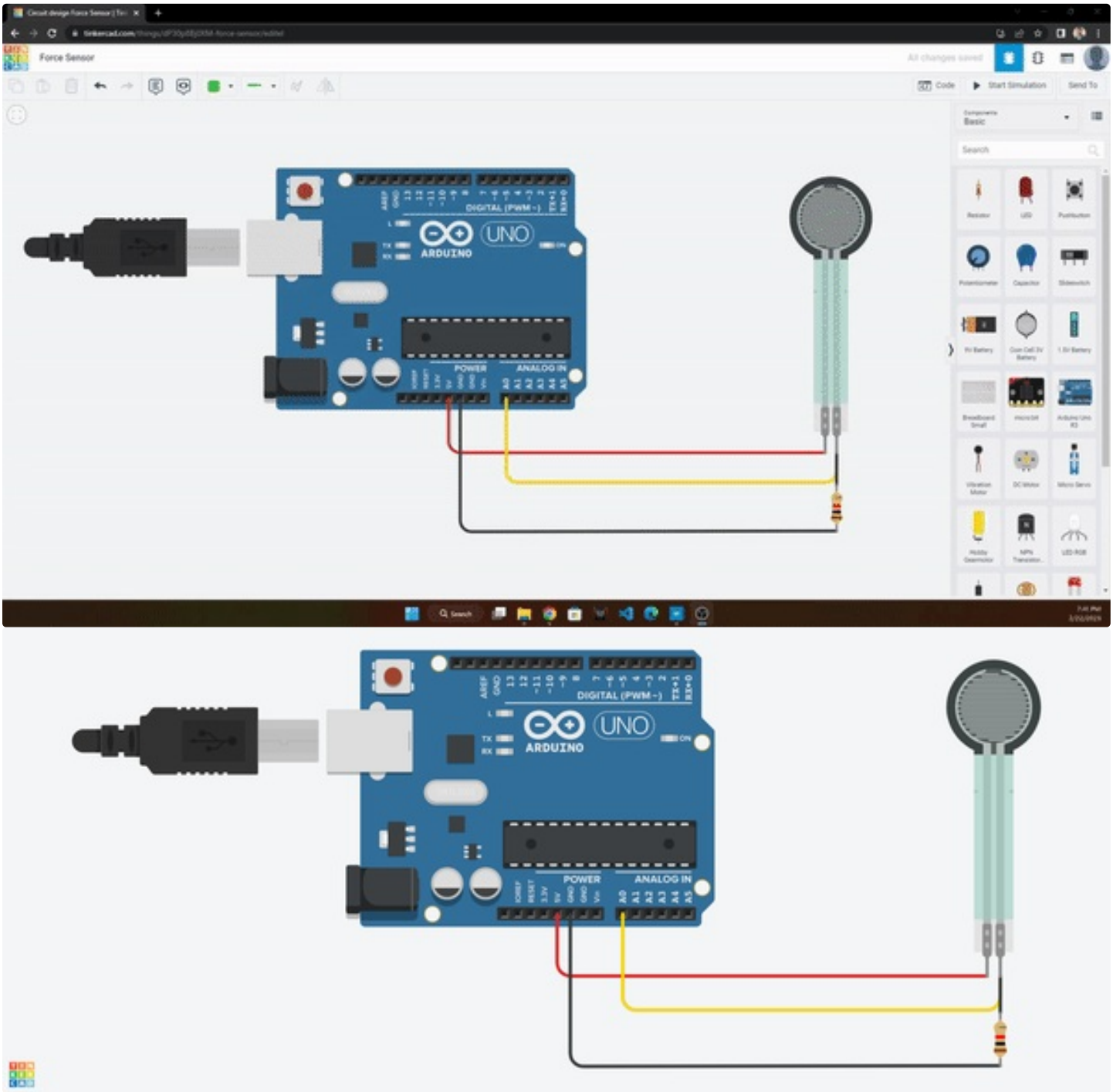
- One terminal of force sensor to A0
- The same terminal should be connected via a 10KOhm resistor to GND
- and another terminal to 5V

Can you name the resistor circuit in the comments, is it Step-Up or Step-Down?

Code

- It's a simple analog read set to pin A0 and prints on serial monitor.
- Start simulating, and click on the force sensor to change the value.

Click to see My Sensor Connection.



Download

https://www.instructables.com/FJ0/8P7O/LEJZ4BM9/FJ08P7OLEJZ4BM9.mov

Step 12: Moisture Sensor

In this step, we are going to make an Automatic Plant Watering System.

Let's drag and drop a Soil Moisture Sensor, a DC Pump(DC Motor), and an Arduino

Connect

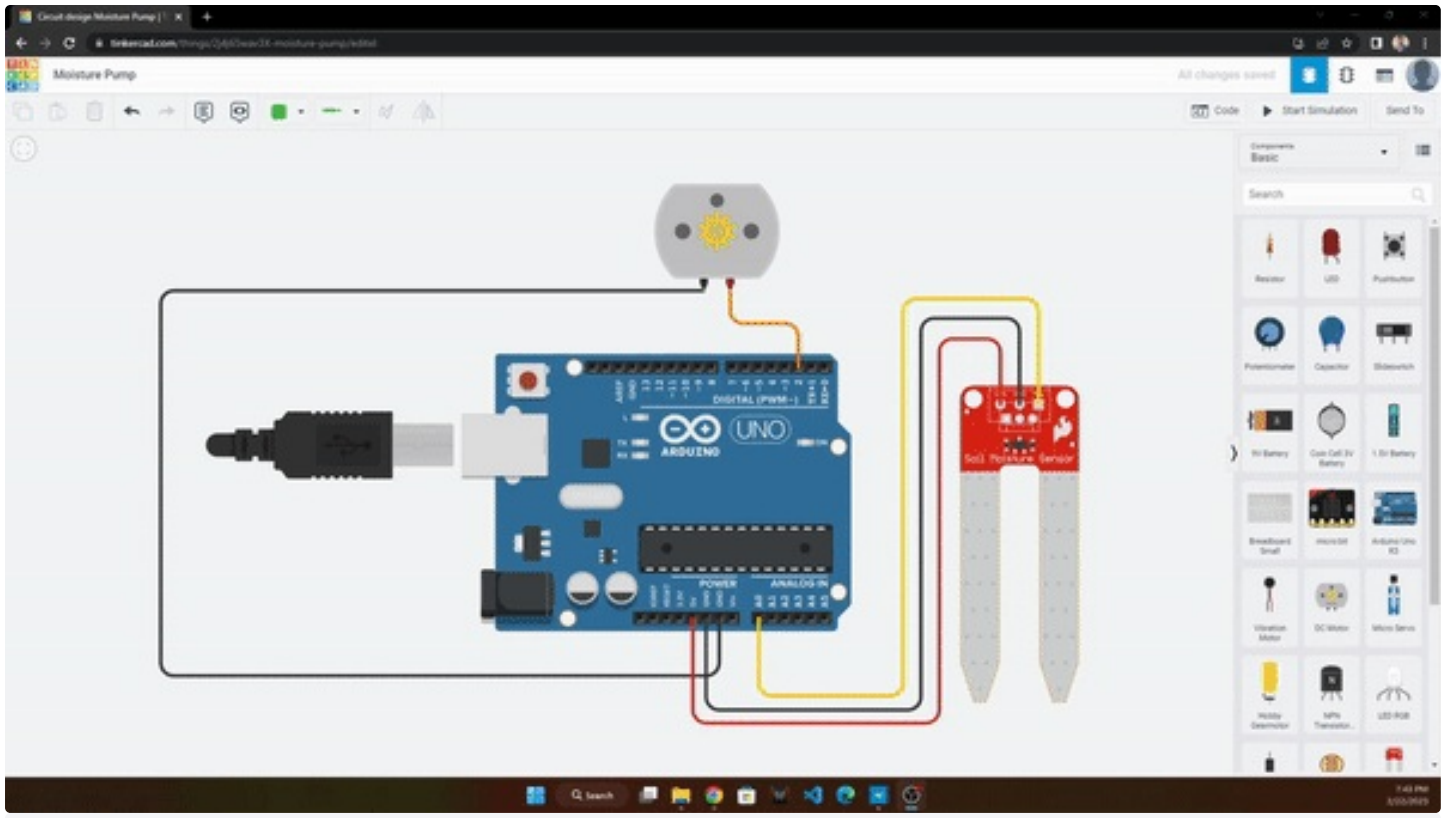
- Moisture sensor to Pin A0
- VCC to 5V
- GND to GND
- Terminal 1 of motor/pump to GND
- Terminal 2 of motor/pump to Pin 2

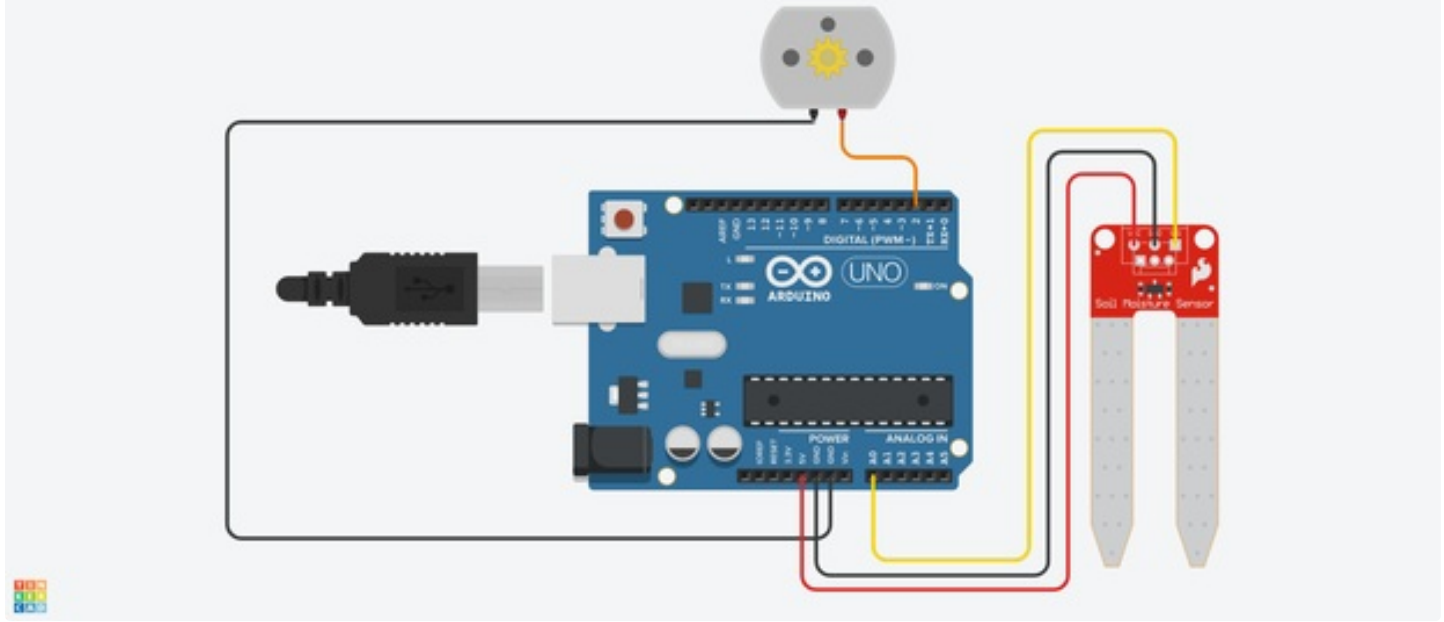
Code

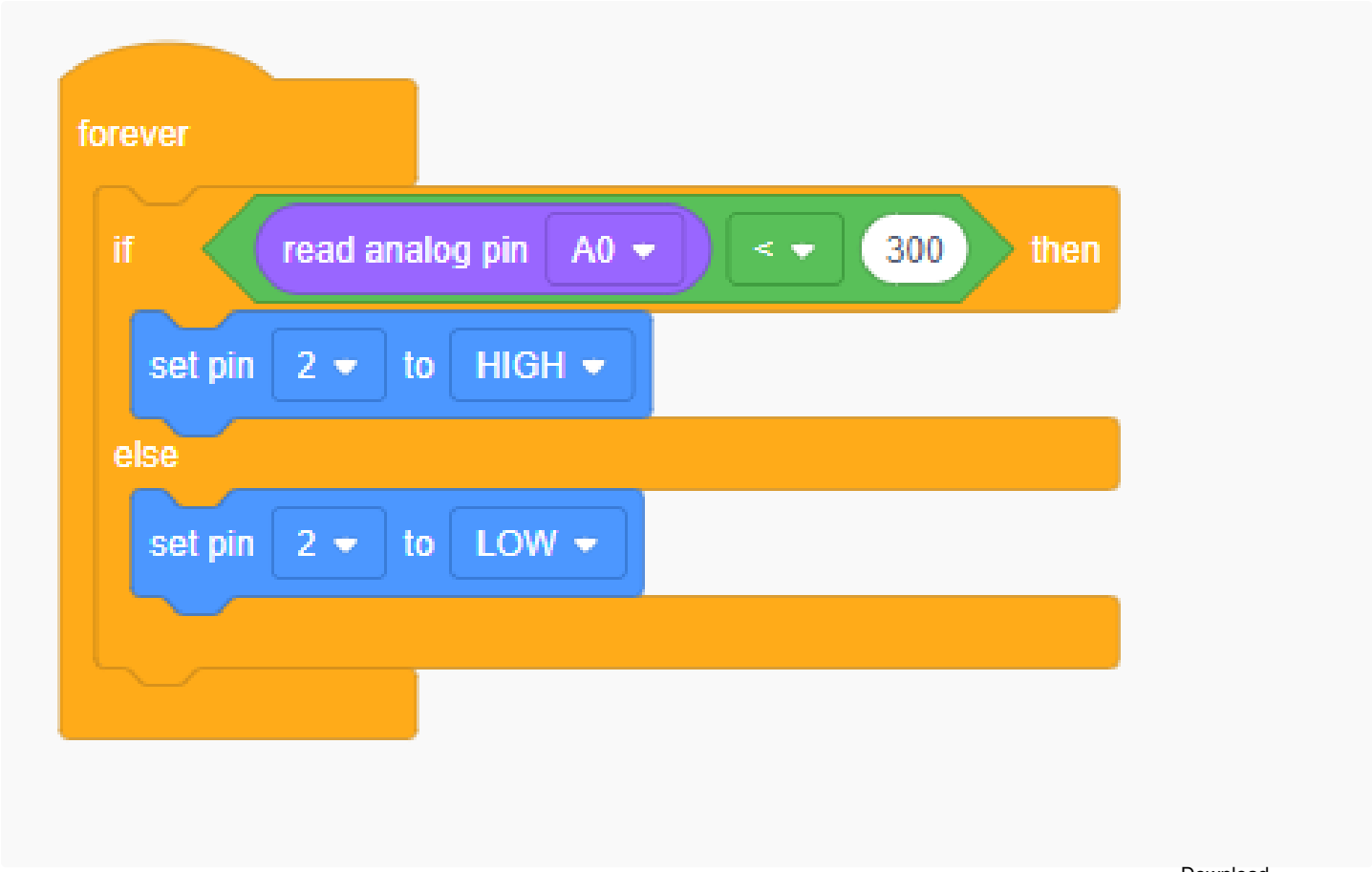
- In the code as you can see we have analog read set to Pin A0
- Comparing its value as < 300 (Means dry, less moisture less value)
- You can change this value as per your requirement
- and the if the value is less than 300 turn on the Pump(Pin 2 High)
- else turn off pump(Pin 2 Low)

Start simulating and click on the sensor to change the value, you will see the motor starts on at low moisture and turns off at high moisture.

Click to see My Automatic Watering System


Moisture Pump
Code
Start Simulation
Send To
Components
Basic
Search
DIGITAL (PWM~)
UNO
ARDUINO
POWER
ANALOG IN
Soil Moisture Sensor


DIGITAL (PWM~)
UNO
ARDUINO
POWER
ANALOG IN
Soil Moisture Sensor



Download

https://www.instructables.com/F49/KS66/LEJZ4BNU/F49KS66LEJZ4BNU.mov

Step 13: Ultrasonic Sensor

In this section, we are going to make a digital scale using an Ultrasonic sensor, LCD Display, and Arduino Uno.

Drag all the components and connect them as shown

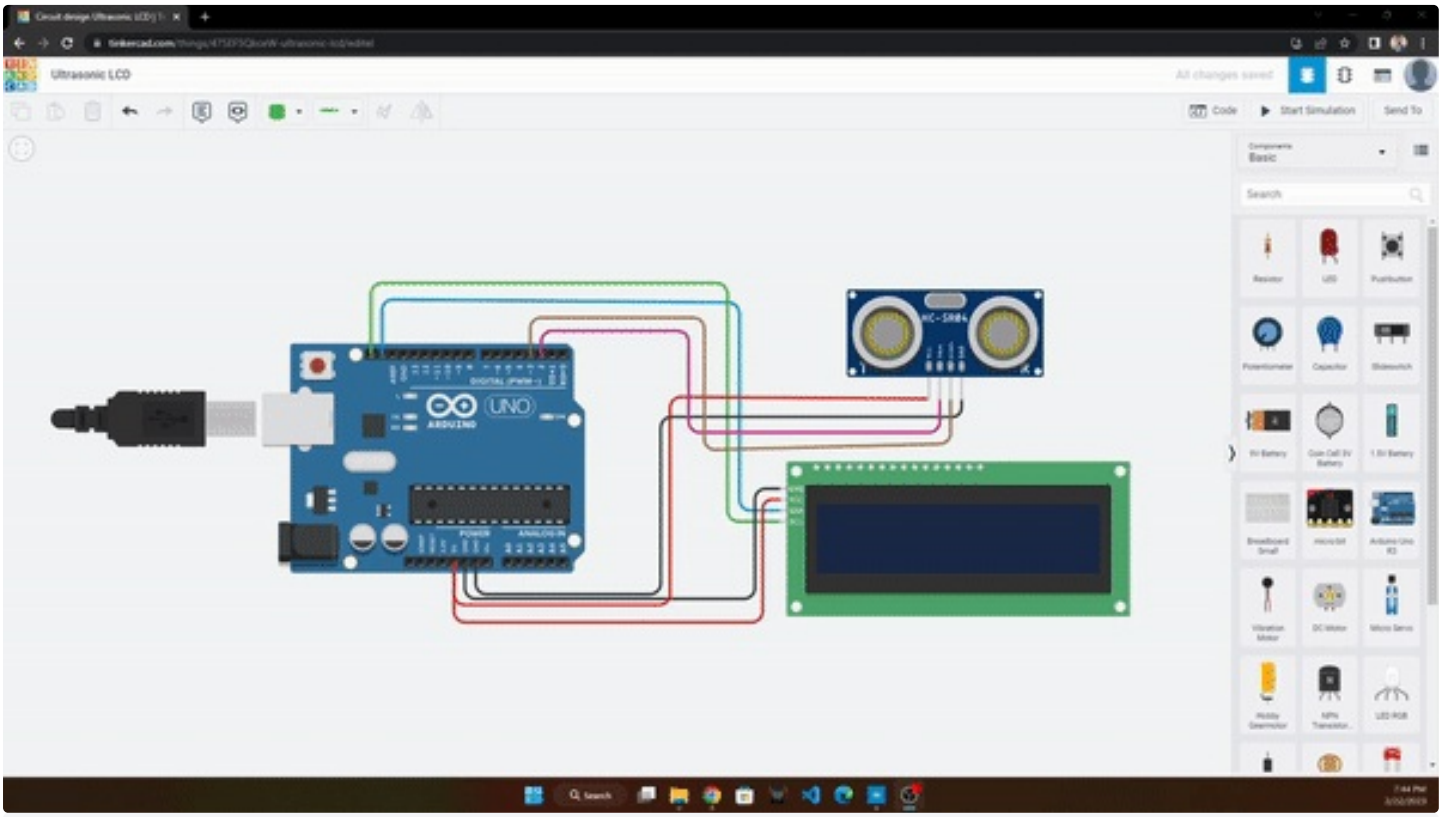
- GNG of Sensor to GND
- Echo to Pin 3
- Trig to Pin 2
- Vcc to 5V
- GND of Display to GND
- VCC to 5V
- SDA to Pin SDA of Arduino
- SCL to Pin SCL of Arduino

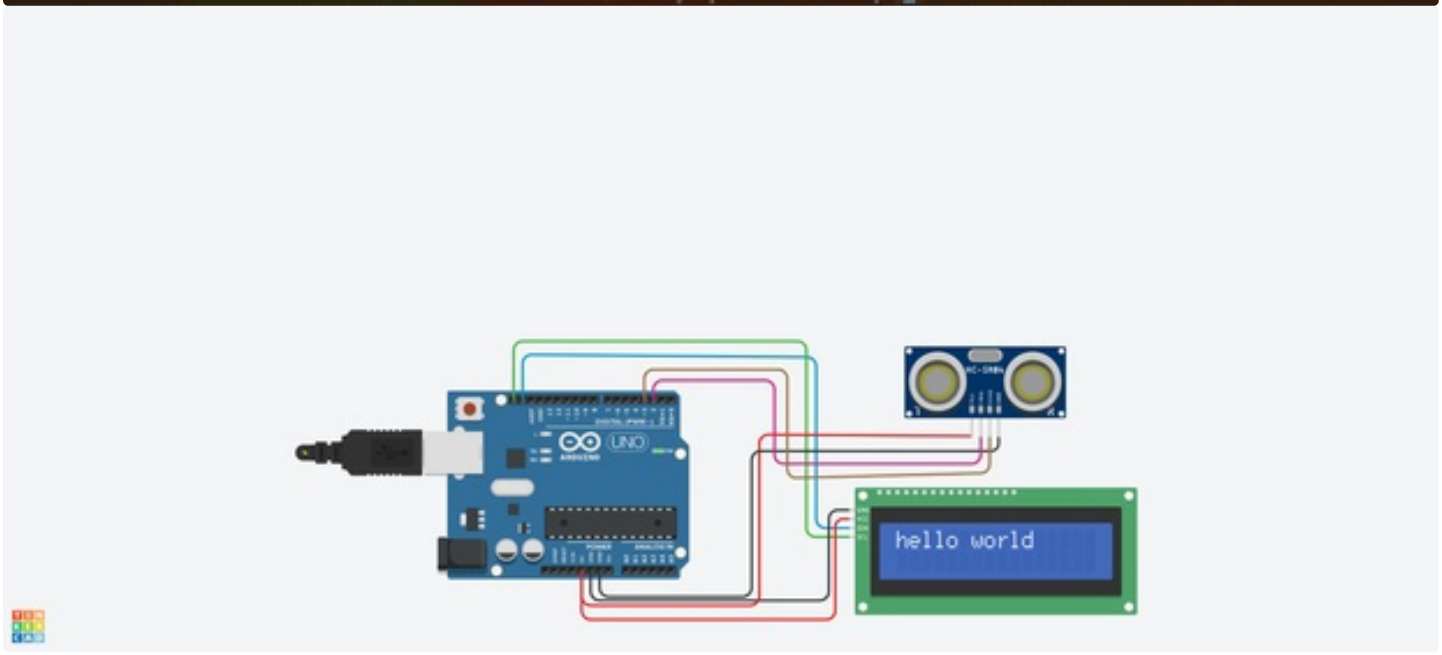
Code

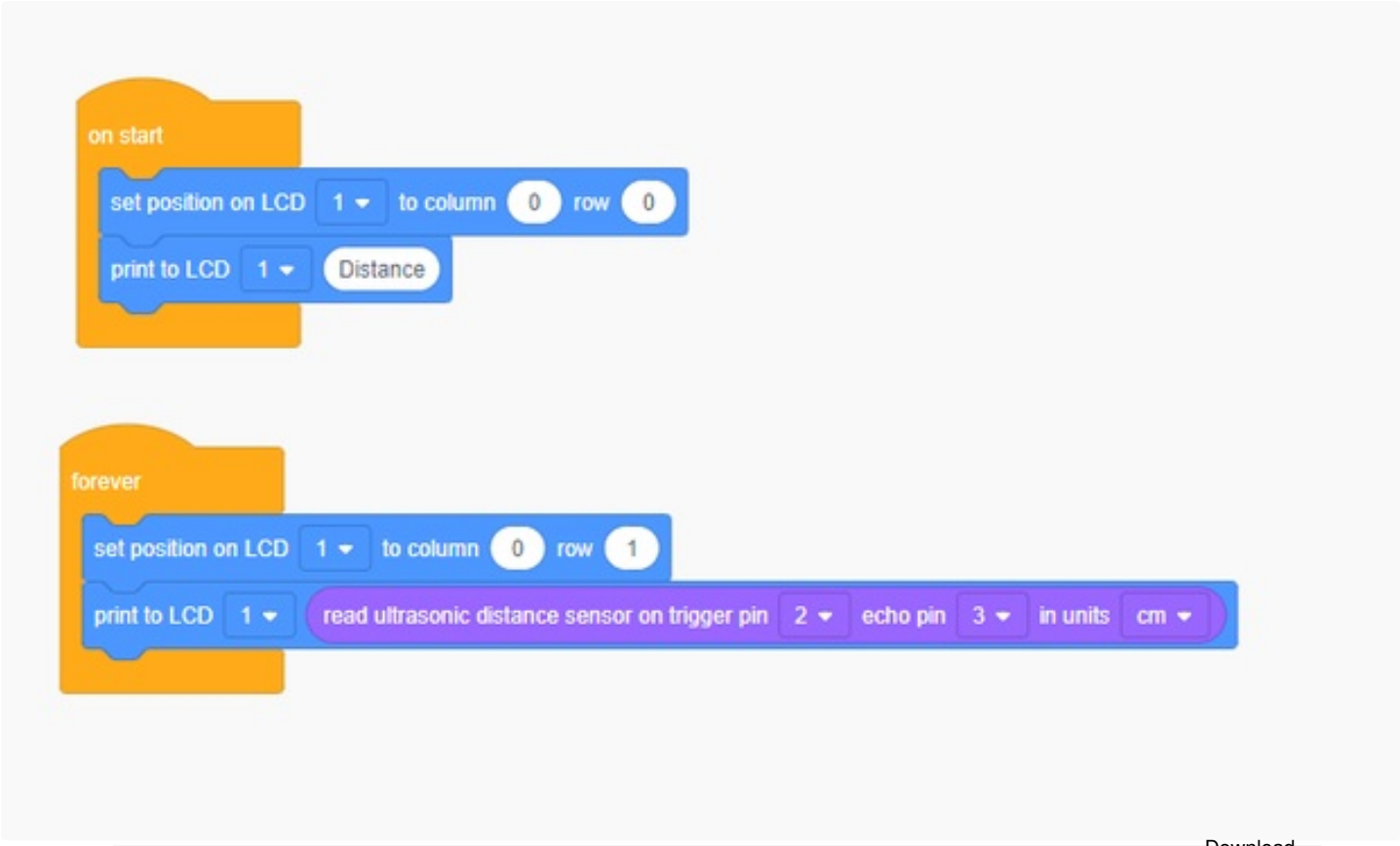
- In on start section, we are printing the message 'Distance' on the LCD display at position 0th column and 0th row.

- Then in forever section, we are setting the position to the 0th column, 1st row, and printing the ultrasonic value using read ultrasonic distance sensor block set to)Trigger Pin 2 and Echo Pin 3.
- Start simulating and click on the sensor to change the distance, you will see the changing values to the LCD display.

Click to see My Digital Scale.







Download

https://www.instructables.com/F6H/ZFSG/LEJZ4BPE/F6HZFSGLEJZ4BPE.mov

Step 14: Servo Motor

In this step, we are going to see how we can control a Servo motor using an Arduino Uno.

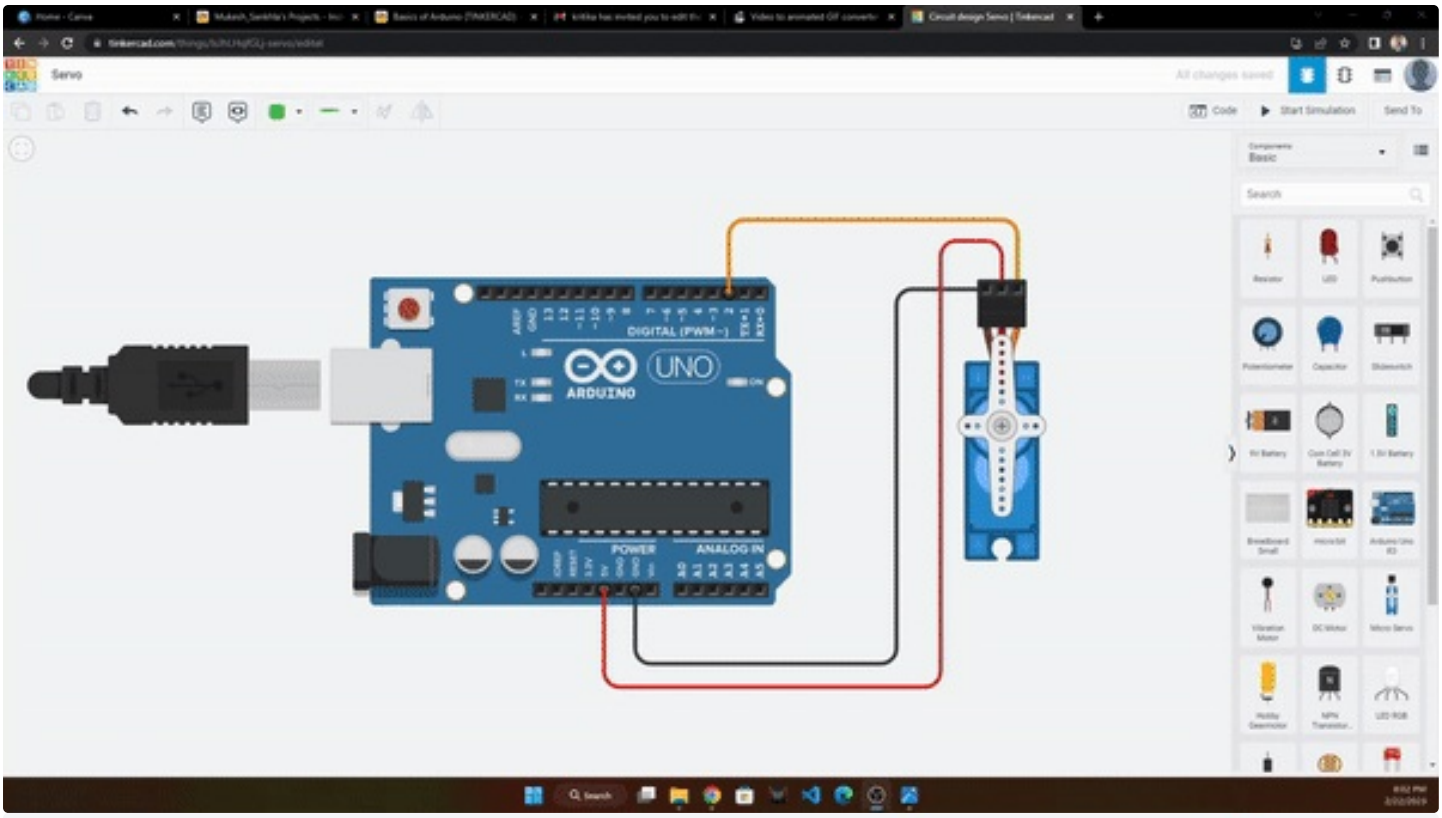
Drag and drop all the components and connect

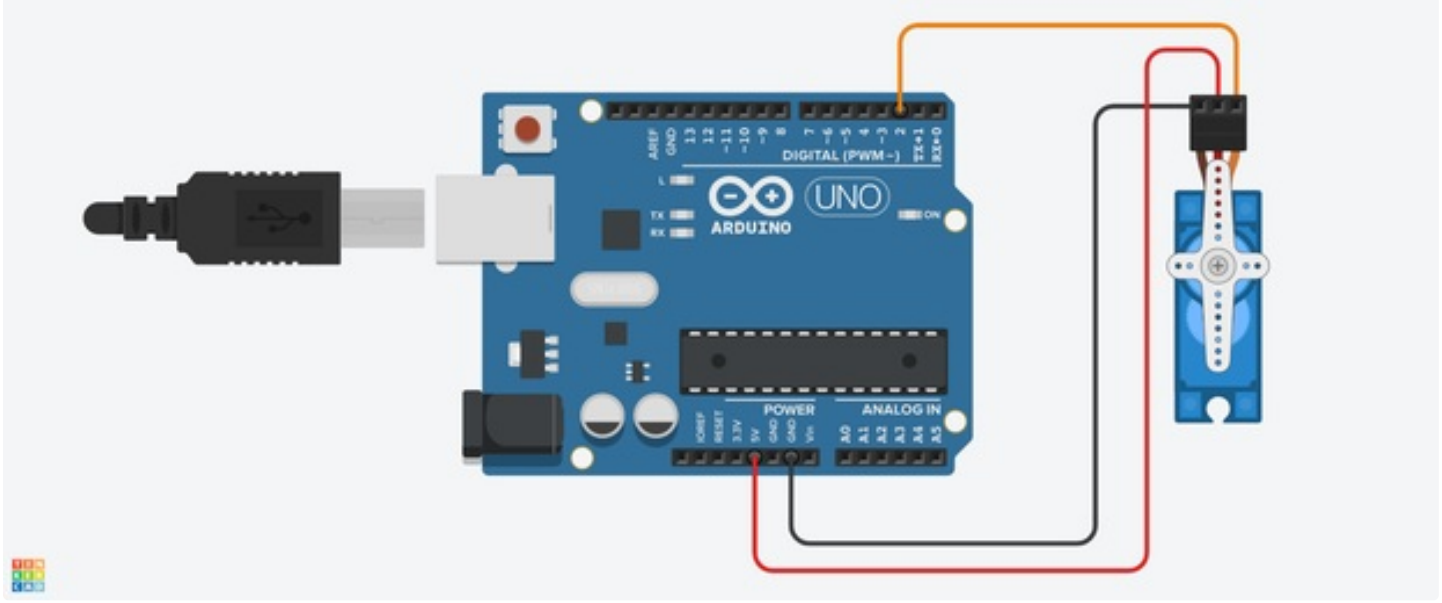
- Signal to Pin 2
- Power to 5V
- GND to GND

Code

- In this, I am rotating the motor from 0 degrees to 180 deg.
- Then from 180 to 120 deg and back to 0 deg.

Click to see My Servo Interface.


Servo
Code
Start Simulation
Send To
DIGITAL (PWM~)
UNO
ARDUINO
POWER
ANALOG IN


DIGITAL (PWM~)
UNO
ARDUINO
POWER
ANALOG IN

Download

https://www.instructables.com/F8E/G44Z/LEJZ4BQZ/F8EG44ZLEJZ4BQZ.mov

Step 15: Activity

Here is a project that I have built back in 2020, It's a 4 by 4 Tic Tac Toe game, click to see full-built instructions and get 3D Models.

Task: Below is the modified code of it, which works in Tinkercad, I want you to make a similar 3 by 3 Tic Tac Toe Game in Tinkercad and share it with me in the below I Made It Section.

Flow:

On Start

- Animate the RGB LEDs In Random Colors for 4 cycles.

Forever

- Player 1(Say Green) and Player 2(Say Red) have a specific color, when a player clicks on any button it glows with the corresponding player color.

- Say if a player wins all the LEDs should animate with that player's specific color.
- If all the boxes were filled then the match should draw, showing another color animation.
- For more details you can watch My 4 by 4 Tic Tac Toe game Instructables.

Click to see My 4x4 Tinkercad Version Tic Tac Toe.

```
#include <Keypad.h>
#include <Adafruit_NeoPixel.h>


#define PIN       3
#define NUMPIXELS 16
Adafruit_NeoPixel pixels(NUMPIXELS, PIN, NEO_GRB + NEO_KHZ800);
#define DELAYVAL 500


int player=1;
int n,i,e=0,j=0;
char a[16]={'1','2','3','4','5','6','7','8','9','10','11','12','13','14','15','16'};
char key;
int w=0;


const byte ROWS = 4; //four rows
const byte COLS = 4; //four columns
//define the cymbols on the buttons of the keypads
char hexaKeys[ROWS][COLS] = {
  {'1','2','3','A'},
  {'4','5','6','B'},
  {'7','8','9','C'},
  {'*','0','#','D'}
};
byte rowPins[ROWS] = {4,5,6,7}; //connect to the row pinouts of the keypad
byte colPins[COLS] = {8,9,10,11}; //connect to the column pinouts of the keypad
Keypad customKeypad = Keypad( makeKeymap(hexaKeys), rowPins, colPins, ROWS, COLS);


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


  int s =0;
  while(s != 4)
  {
 for(int i=0; i<NUMPIXELS; i++) {
      pixels.setPixelColor(i, pixels.Color(255, 0, 0));
    }
    pixels.show();
    delay(200);
    for(int i=0; i<NUMPIXELS; i++) {
      pixels.setPixelColor(i, pixels.Color(0, 255, 0));
    }
    pixels.show();
    delay(200);
    s++;
  }
  for(int i=0; i<NUMPIXELS; i++) {
      pixels.setPixelColor(i, pixels.Color(0, 0, 0));
    }
  pixels.show();
}

void loop()
{
  key = customKeypad.getKey();
  if (key && player==1)
  {
    Serial.println("\nP1");
    Serial.println(key);
```

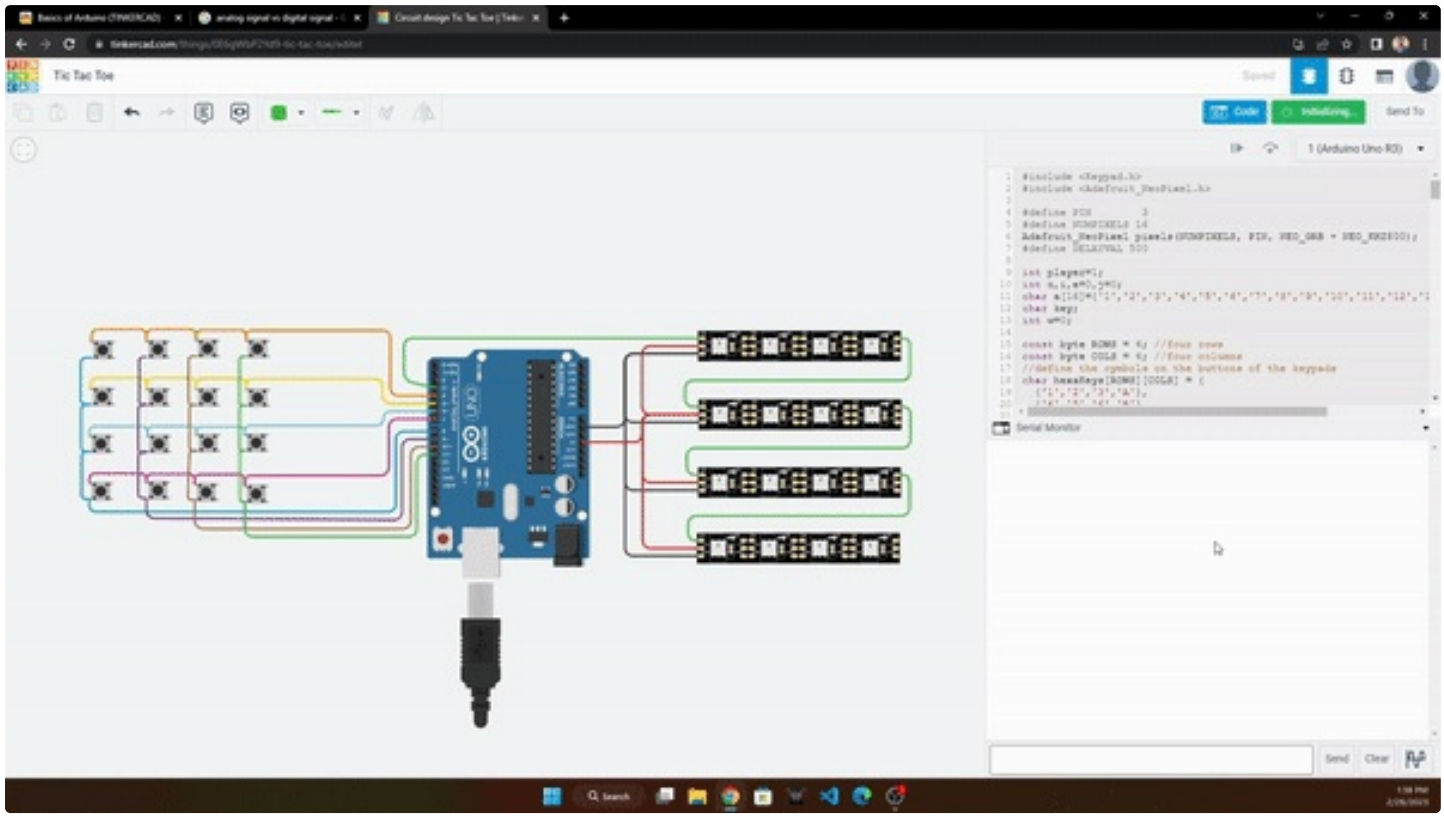
```
  n=number(key);
  Serial.println(n);
  a[n]='X';
  Serial.println(a[n]);
  for(int i=0; i<NUMPIXELS; i++) {
    if(a[i]=='X'){
      pixels.setPixelColor(i, pixels.Color(0, 255, 0));
      pixels.show();
    }
    if(a[i]=='O'){
      pixels.setPixelColor(i, pixels.Color(255, 0, 0));
      pixels.show();
    }
  }
  w=check(a,n);
  if(w==1)
  {
    Serial.println("Player 1 Wins!");
    while(1)
    {
     for(int i=0; i<NUMPIXELS; i++) {
       pixels.setPixelColor(i, pixels.Color(0, 255, 0));
   }
      pixels.show();
      delay(200);
      for(int i=0; i<NUMPIXELS; i++) {
       pixels.setPixelColor(i, pixels.Color(0, 0, 0));
      }
      pixels.show();
      delay(200);
    }
  }
  player++;
  delay(1000);
}
delay(20);
key = customKeypad.getKey();
if (key && player==2)
{
  Serial.println("\nP2");
  Serial.println(key);
  n=number(key);
  Serial.println(n);
  a[n]='O';
  Serial.println(a[n]);
  for(int i=0; i<NUMPIXELS; i++) {
    if(a[i]=='X'){
      pixels.setPixelColor(i, pixels.Color(0, 255, 0));
      pixels.show();
    }
    if(a[i]=='O'){
      pixels.setPixelColor(i, pixels.Color(255, 0, 0));
      pixels.show();
    }
  }
  w=check(a,n);
  if(w==2)
  {
    Serial.println("Player 2 Wins!");
    while(1)
    {
     for(int i=0; i<NUMPIXELS; i++) {
       pixels.setPixelColor(i, pixels.Color(255, 0, 0));
   }
      pixels.show();
      delay(200);
      for(int i=0; i<NUMPIXELS; i++) {
       pixels.setPixelColor(i, pixels.Color(0, 0, 0));
      }
      pixels.show();
      delay(200);
    }
  }
  e++;
 player--;
 delay(1000);
```

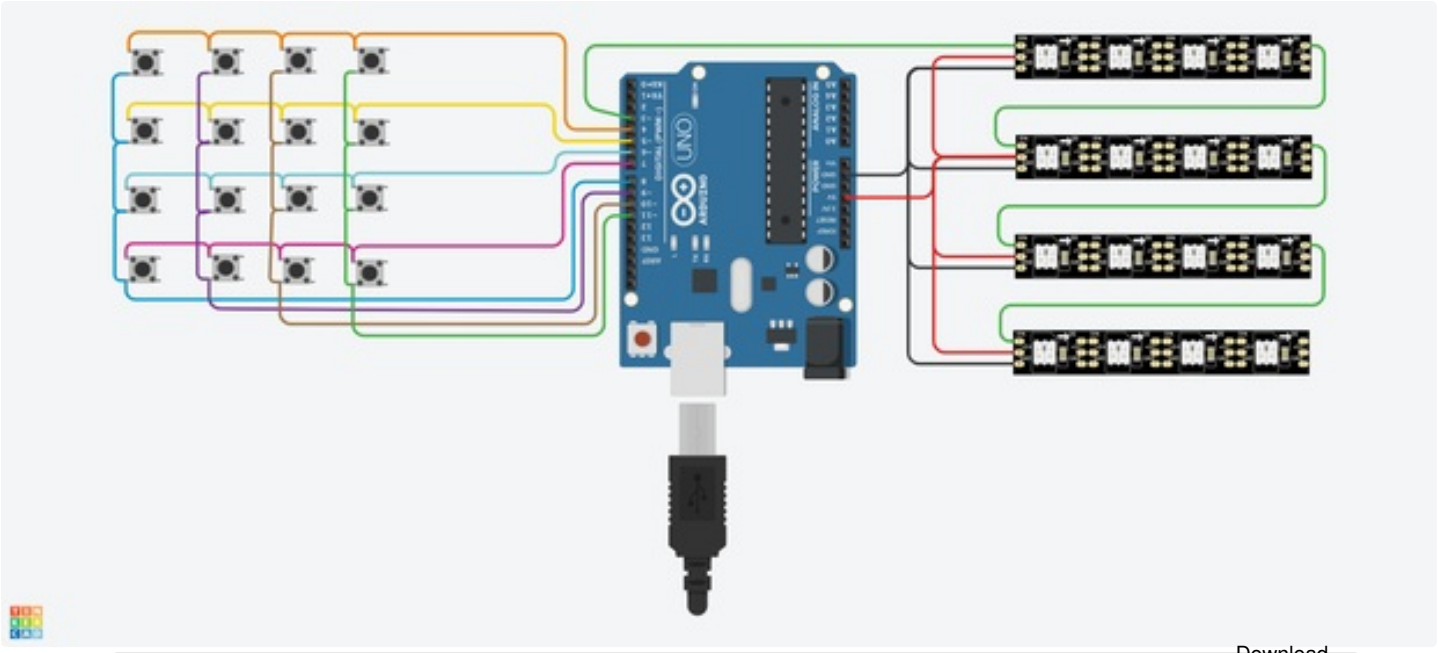
```
   delay(1000);
  }
  if(e==8)
  {
    Serial.println("Game Draw");
    while(1)
    {

  }
  }
  delay(20);
}


int number(char keyn)
{
  if(keyn == '1')
    return 0;
  if(keyn == '2')
    return 1;
  if(keyn == '3')
    return 2;
  if(key == 'A')
    return 3;
  if(keyn == '4')
    return 4;
  if(keyn == '5')
    return 5;
  if(keyn == '6')
    return 6;
  if(keyn == 'B')
    return 7;
  if(keyn == '7')
    return 8;
  if(keyn == '8')
    return 9;
  if(keyn == '9')
    return 10;
  if(keyn == 'C')
    return 11;
  if(key == '*')
    return 12;
  if(keyn == '0')
    return 13;
  if(keyn == '#')
    return 14;
  if(keyn == 'D')
    return 15;
}


int check(char a[16],int n)
{
if(a[0]=='X' && a[1]=='X' && a[2]=='X' && a[3]=='X')
    return 1;
  if(a[0]=='O' && a[1]=='O' && a[2]=='O' && a[3]=='O')
    return 2;
  if(a[0]=='X' && a[4]=='X' && a[8]=='X' && a[12]=='X')
    return 1;
  if(a[0]=='O' && a[4]=='O' && a[8]=='O' && a[12]=='O')
    return 2;
  if(a[0]=='X' && a[5]=='X' && a[10]=='X' && a[15]=='X')
    return 1;
  if(a[0]=='O' && a[5]=='O' && a[10]=='O' && a[15]=='O')
    return 2;
  if(a[1]=='X' && a[5]=='X' && a[9]=='X' && a[13]=='X')
    return 1;
  if(a[1]=='O' && a[5]=='O' && a[9]=='O' && a[13]=='O')
    return 2;
  if(a[2]=='X' && a[6]=='X' && a[10]=='X' && a[14]=='X')
    return 1;
  if(a[2]=='O' && a[6]=='O' && a[10]=='O' && a[14]=='O')
    return 2;
  if(a[3]=='X' && a[7]=='X' && a[11]=='X' && a[15]=='X')
    return 1;
  if(a[3]=='O' && a[7]=='O' && a[11]=='O' && a[15]=='O')
```

```
  return 2;
 if(a[3]=='X' && a[6]=='X' && a[9]=='X' && a[12]=='X')
  return 1;
 if(a[3]=='O' && a[6]=='O' && a[9]=='O' && a[12]=='O')
  return 2;
 if(a[4]=='X' && a[5]=='X' && a[6]=='X' && a[7]=='X')
  return 1;
 if(a[4]=='O' && a[5]=='O' && a[6]=='O' && a[7]=='O')
  return 2;
 if(a[8]=='X' && a[9]=='X' && a[10]=='X' && a[11]=='X')
  return 1;
 if(a[8]=='O' && a[9]=='O' && a[10]=='O' && a[11]=='O')
  return 2;
 if(a[12]=='X' && a[13]=='X' && a[14]=='X' && a[15]=='X')
  return 1;
 if(a[12]=='O' && a[13]=='O' && a[14]=='O' && a[15]=='O')
  return 2;
}
```

Thank You!

I Hope you found these to be helpful for you to start tinkering with Arduino.





Download

https://www.instructables.com/FNA/W6FG/LEJZ4BSL/FNAW6FGLEJZ4BSL.mov

Your presentation is good, but I suggest you go and look at real-life traffic light sequences. In most countries the sequence after green is yellow, then back to red. In some countries the sequence after red is red-and-yellow, then green

Excellent, thank you

It's my pleasure

well written mate.

Thank You