

A COLLECTION OF TINKERCAD-ABLE PROJECTS: VOLUME I





A COLLECTION OF TINKERCAD-ABLE PROJECTS: VOLUME I

WRITTEN BY
PENGIRAN HAFFEY IZZAT B ABDUL GHAFOR
PHELYCIA FIZZY BINTI WELLING
SYAZWANI BINTI ABDUL KADIR

PUBLISHED BY:

POLITEKNIK MUKAH KM 7.5 JALAN OYA, 96400 MUKAH, SARAWAK

TEL: +6084-874001 FAX: +6084-874005 HTTPS://PMU.MYPOLYCC.EDU.MY/

A COLLECTION OF TINKERCAD-ABLE PROJECTS: VOLUME I

AUTHOR: PENGIRAN HAFFEY IZZAT B ABDUL GHAFOR PHELYCIA FIZZY BINTI WELLING SYAZWANI BINTI ABDUL KADIR

COPYRIGHT @ 2024 EDITION

ALL RIGHT RESERVED. NO PART OF THIS
PUBLICATION MAY BE REPRODUCE,
DISTRIBUTED OR TRANSMITTED IN ANY FORM
OR BY ANY MEANS, INCLUDING PHOTOCOPYING,
RECORDING OR OTHER ELECTRONIC OR
MECHANIC METHODS, WITHOUT THE PRIOR
WRITTEN PERMISSION OF THE WRITER, EXPECT
IN THE CASE OF BRIEF QUOTATION EMBODIED
REVIEW AND CERTAIN OTHER NONCOMMERCIAL AND EDUCATION USE PERMITTED
BY POLITEKNIK MUKAH

PREFACE

THE EMBEDDED INTERNET OF THINGS (IOT) COURSE AT POLYTECHNIC HAS AN **ACCOMPANYING EBOOK** CONTAINING A COLLECTION OF ARDUINO PROJECTS, THE FROOK IS INTENDED TO GUIDE STUDENTS THROUGH THE CREATION OF THESE PROJECTS USING **AUTODESK'S TINKERCAD** APPLICATION, THE PRIMARY AIM OF THIS RESOURCE IS TO **FAMILIARIZE STUDENTS** WITH THE DEVELOPMENT OF ARDUINO PROJECTS THROUGH SIMULATION TOOLS.

ABOUT THE AUTHORS



- PENGIRAN HAFFEY IZZAT B ABDUL GHAFOR
- POLITEKNIK MUKAH, SARAWAK

- PHELYCIA FIZZY BINTI WELLING
- POLITEKNIK MUKAH, SARAWAK





- SYAZWANI BINTI ABDUL KADIR
- POLITEKNIK MUKAH, SARAWAK

PAGE I

COPYRIGHT

PAGE ii

PREFACE

PAGE iii

ABOUT THE AUTHORS

PAGE 1

INTRODUCTION TO ARDUINO UNO

PAGE 4

INTRODUCTION TO TINKERCAD

PAGE 9

PROJECT I: LED BLINK (AUTO)

PAGE 21

PROJECT II : LED BLINK (PUSH BUTTON)

S **—** Z H **(** Ш

PAGE 35

PROJECT III: LED FLOW (AUTO)

PAGE 47

PROJECT IV : LED FLOW (PUSH BUTTON)

PAGE 61

PROJECT V : RGB LED BLINK

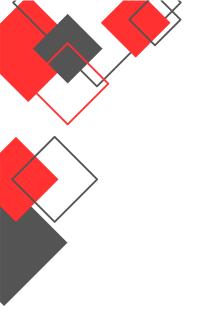
COLOUR CHANGE

(AUTO)

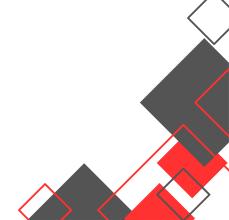
PAGE 74

PROJECT VI : RGB LED BLINK
COLOUR CHANGE
(PUSH BUTTON)

PAGE 87 REFERENCES

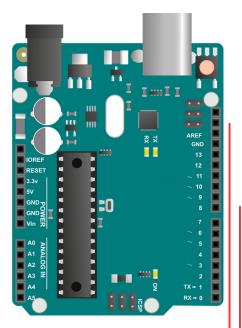


INTRODUCTION TO ARDUINO UNO





WHAT IS ARDUINO UNO



- The Arduino Uno is a widely embraced microcontroller board.
- It is favored for its versatility and accessibility in the realm of electronics.
- Developed by Arduino, it serves as an entry point for novices and a reliable tool for seasoned developers in the maker community.
- It features a user-friendly USB interface, allowing seamless connection to a computer for programming and power.
- With a variety of input/output pins, it enables the interfacing of sensors, actuators, and other electronic components.



Abundance of Resources:

With a vast array of tutorials, libraries, and community forums available, the Arduino Uno R3 provides ample support for users at every skill level.

WHY ARDUINO UNO

Open-Source Nature: Both its hardware and software are open source, encouraging collaboration, innovation, and customization within the community.

Ease of Use: The Uno R3 is renowned for its userfriendly interface, making it an excellent choice for beginners diving into the world of programming and electronics.

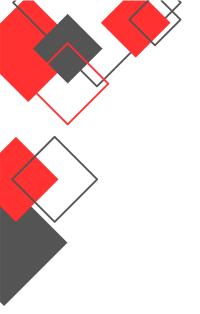


Affordability:

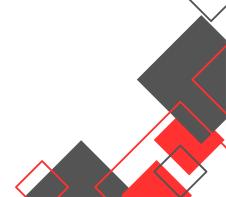
Compared to many other microcontroller platforms, the Arduino Uno R3 is costeffective, allowing enthusiasts to experiment and innovate without a hefty investment.

Cross-Platform Compatibility:

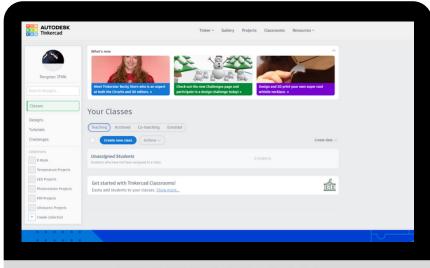
Its compatibility with various operating systems -Windows, Mac, and Linux makes it accessible to a wide range of users



INTRODUCTION TO TINKERCAD







TINKERCAD.COM



Tinkercad is an intuitive, browser-based 3D design and modeling tool created to make the world of 3D printing and CAD (Computer-Aided Design) accessible to everyone.

Launched in 2011, Tinkercad quickly gained popularity for its user-friendly interface, making it easy for people with little to no prior experience in CAD to design and create 3D models.

The platform is not limited to 3D modeling; it also allows users to simulate and create circuits through its Tinkercad Circuits feature, enabling experimentation with electronics virtually.

Beyond 3D Modeling:

- Not confined to 3D modeling; includes Tinkercad Circuits for virtual electronics experiment.
- Allows users to simulate and create circuits within the platform.
- The simulation includes simulation of Arduino Uno.



Precision Tools:

 It provides tools for accurate measurements, alignment, and grouping objects, simplifying the design process.

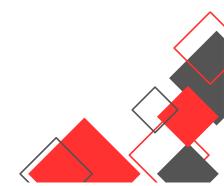


Versatility and Integration:

- Tinkercad's versatility, ease of use, and integration with 3D printing services.
- Popular among educators, hobbyists, and professionals for accessible yet powerful design capabilities.

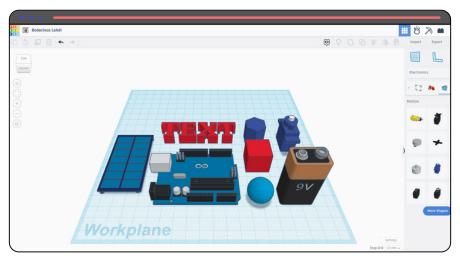
Simplicity In Design:

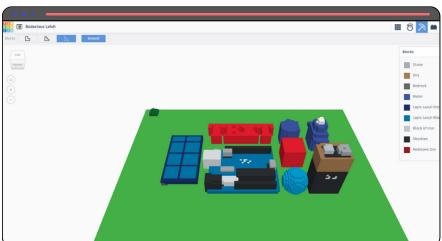
- Simple drag-and-drop method with basic shapes and geometric forms.
- Enables easy manipulation and combination for crafting designs.





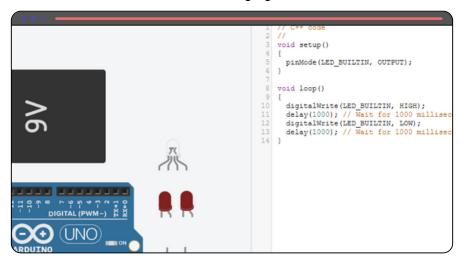
3D DESIGN

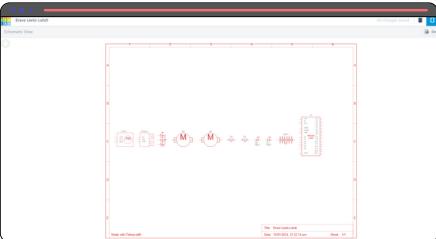


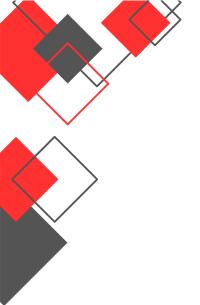




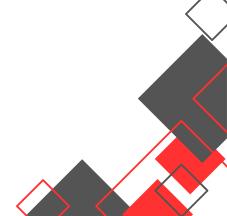
CIRCUIT DESIGN







PROJECT I LED BLINK (AUTO)





DIFFICULTY LEVEL	BEGINNER

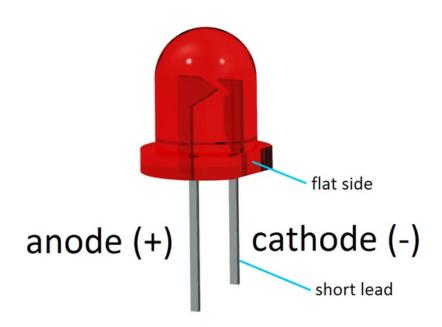
HARDWARE NEEDED	AMOUNT	
Single Coloured Light Emitting Diode (LED)	1	
Arduino UNO R3	1	THE TAX ANALOGY AND ANALOGY AN
Breadboard	1	<u></u>
200 Ohm Resistor	1	

INPUT	OUTPUT
NONE	LED Blinking With Intervals of 1 Seconds



WHAT TO KNOW ABOUT LED:

- Also known as Light Emitting Diode (LED)
- Two legs/points [1 Anode (+) & 1 Cathode (-)]
- The colour of an LED is determined by the material used in the semiconducting element
- Most common colours (Red, Blue, Yellow, Orange, Green, White)
- The more current is being supply, the brighter it is
- Without a resistor to reduce the current, the shorter its lifespan will be





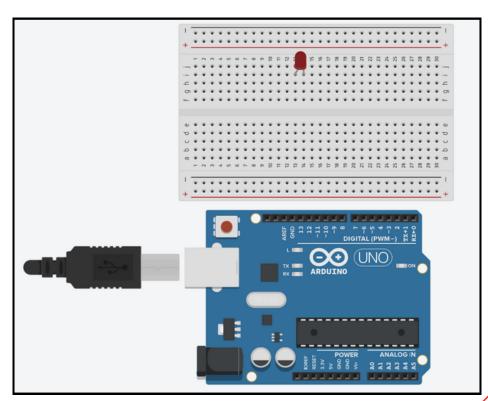






STEP 1:

- Arrange Arduino UNO and Breadboard as below
- Insert an LED into the breadboard with the Anode (positive leg) on the left and the Cathode (negative leg on the right)

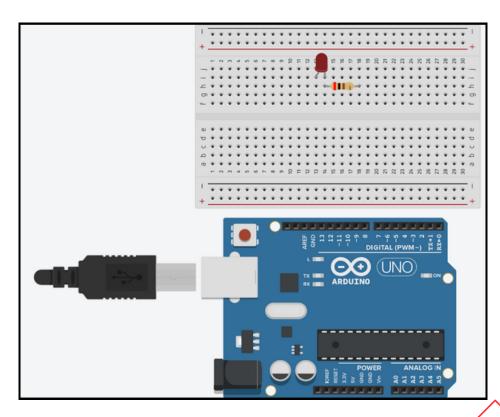






STEP 2:

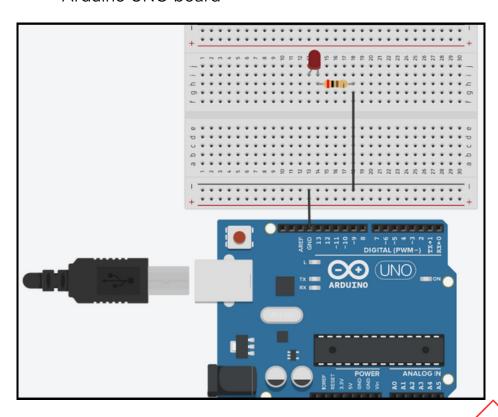
- Connect a 200 Ohm resistor into the Cathode leg of the LED
- Resistors are not polarised, so orientation does not matter
- Resistor will help to limit current going into the LED





STEP 3:

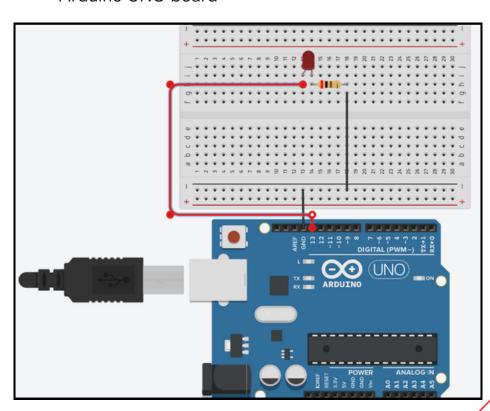
 Connect the resistor to the Ground (GND) of the Arduino UNO board





STEP 4:

 Connect the anode leg of the LED to PIN 13 of the Arduino UNO board





STEP 5:

 Insert the following code (next page) inside code tab of Tinkercad project

```
₹/ Code

    Start Simulation

                                                               Send To
                                                   1 (Arduino Uno R3) -
 Text
 1 const int LED OUTPUT PIN = 13;
 2 void setup() {
     pinMode (LED OUTPUT PIN, OUTPUT);
 6 void loop() {
     digitalWrite(LED_OUTPUT_PIN, HIGH); // turn LED ON
                                            // Wait 1 seconds
     delay(1000);
                                            // turn LED OFF
     digitalWrite(LED_OUTPUT_PIN, LOW);
     delay(1000);
                                            // Wait 1 seconds
11 }
" Serial Monitor
```



STEP 5:

• The code as shown in previous page:

```
const int LED_OUTPUT_PIN = 13;
void setup() {
   pinMode(LED_OUTPUT_PIN, OUTPUT);
}

void loop() {
   digitalWrite(LED_OUTPUT_PIN, HIGH); // turn LED ON
   delay(1000); // Wait 1 seconds
   digitalWrite(LED_OUTPUT_PIN, LOW); // turn LED OFF
   delay(1000); // Wait 1 seconds
}

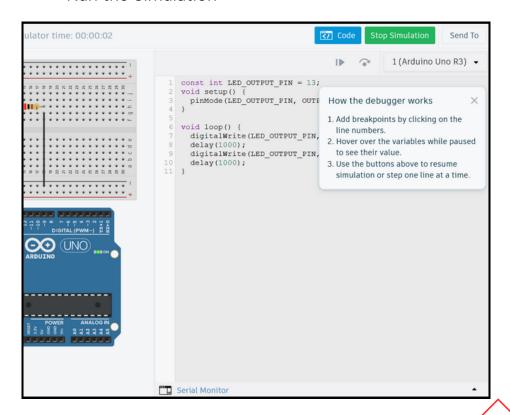
// Wait 1 seconds
// Wait 1 seconds
// Wait 1 seconds
// Wait 1 seconds
```





STEP 6:

· Run the Simulation





STEP 6:

 Please scan the QR Code below to see a simulation video of the project

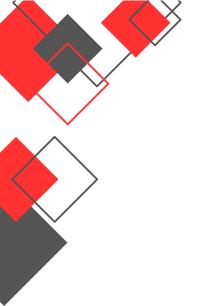




CONCLUSION:

As shown in the video demonstration of the simulation, the LED will blink with intervals of 1 seconds (1 seconds ON, 1 seconds OFF). Students are encourage to tinker with the code and add some additional LEDs.





PROJECT II LED BLINK (PUSH BUTTON)





DIFFICULTY LEVEL BEGINNER

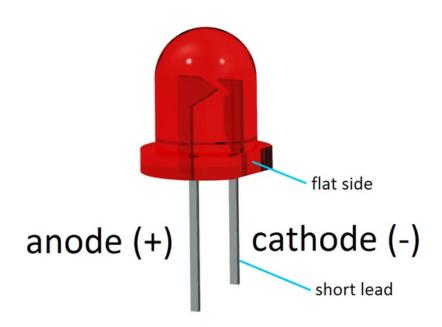
HARDWARE NEEDED	AMOUNT	
Single Coloured Light Emitting Diode (LED)	1	
Arduino UNO R3	1	THE RESERVE THE PARTY OF THE PA
Breadboard	1	
200 Ohm Resistor	1	
Push Button	1	

INPUT	OUTPUT
Button Pressed	LED Blinking When Button Is Pressed



WHAT TO KNOW ABOUT LED:

- Also known as Light Emitting Diode (LED)
- Two legs/points [1 Anode (+) & 1 Cathode (-)]
- The colour of an LED is determined by the material used in the semiconducting element
- Most common colours (Red, Blue, Yellow, Orange, Green, White)
- The more current is being supply, the brighter it is
- Without a resistor to reduce the current, the shorter its lifespan will be

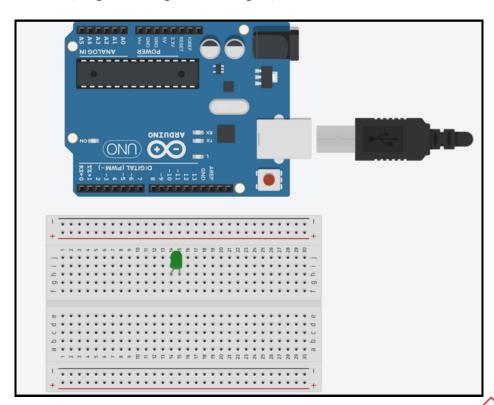






STEP 1:

- Arrange Arduino UNO and Breadboard as below.
- Insert an LED into the breadboard with the Anode (positive leg) on the left and the Cathode (negative leg on the right).

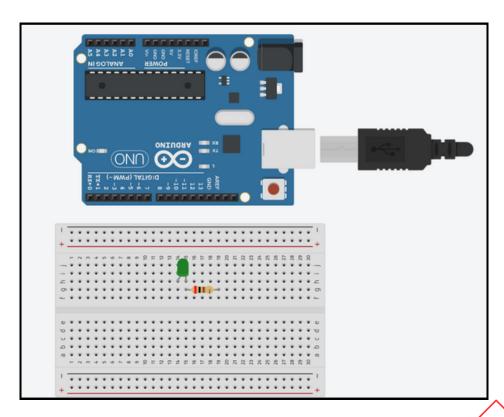






STEP 2:

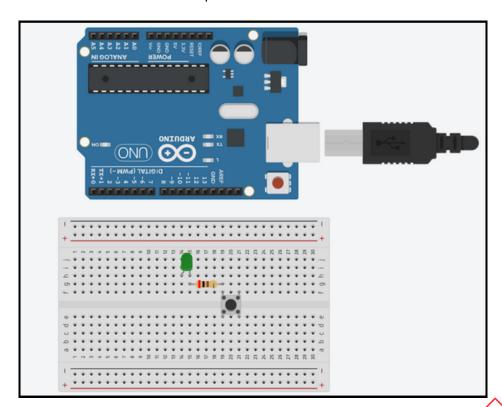
- Connect a 200 Ohm resistor into the Cathode leg of the LED.
- Resistors are not polarised, so orientation does not matter.
- Resistor will help to limit current going into the LED.





STEP 3:

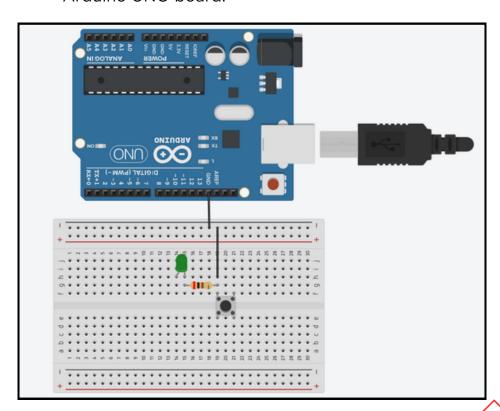
- · Connect a Push Button to the breadboard
- Make sure that one pin of the button are connected to the pin of the resistor





STEP 4:

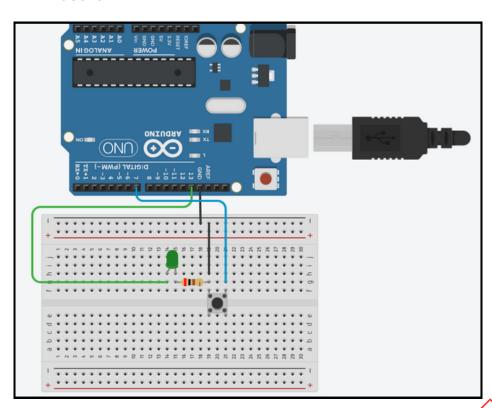
 Connect the resistor to the Ground (GND) of the Arduino UNO board.





STEP 5:

- Connect the anode leg of the LED to PIN 13 of the Arduino UNO board
- Connect the button to PIN 7 of the Arduino UNO board





STEP 6:

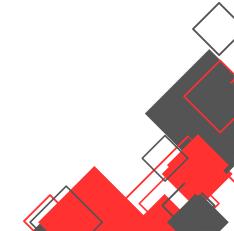
 Insert the following code (next page) inside code tab of Tinkercad project



STEP 6:

The code as shown in previous page:

```
1 const int Led_Pin = 13;
   const int Button_Pin = 7;
 45
   void setup()
 6
    pinMode(Led_Pin,OUTPUT);
 7
    pinMode(Button_Pin,INPUT_PULLUP);
 8
 9
10
   void loop()
11
    int digitalVal = digitalRead(Button_Pin);
12
13
    if(HIGH == digitalVal)
14
15
      digitalWrite(Led_Pin,LOW);
16
    else
17
18
19
      digitalWrite(Led_Pin,HIGH);
20
21 }
```





STEP 7:

• Run the simulation

```
₹7 Code
                                             Start Si
Text
 1 const int Led_Pin = 13;
 2 const int Button Pin = 7;
 4 void setup()
 5 {
     pinMode (Led Pin, OUTPUT);
      pinMode (Button_Pin, INPUT_PULLUP);
 8
   void loop()
      int digitalVal = digitalRead(Button_Pin);
      if (HIGH == digitalVal)
14
       digitalWrite(Led_Pin,LOW);
     else
       digitalWrite(Led_Pin, HIGH);
21 }
```



STEP 7:

 Please scan the QR Code below to see a simulation video of the project

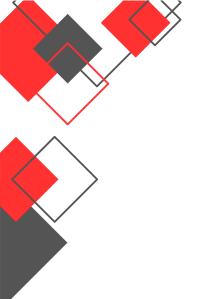




CONCLUSION:

As shown in the video demonstration of the simulation, the LED will blink when the button is push/pres. Students are encourage to tinker with the code and add some additional LEDs.





PROJECT III LED FLOW (AUTO)





DIFFICULTY LEVEL	BEGINNER
DIFFICULTY LEVEL	BEGINNER

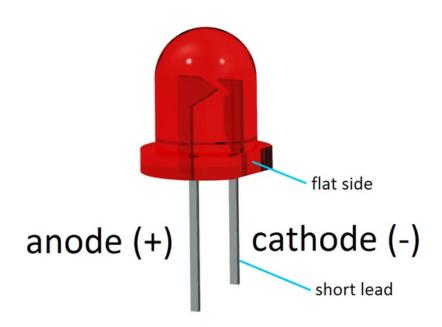
HARDWARE NEEDED	AMOUNT	
Single Coloured Light Emitting Diode (LED)	1	
Arduino UNO R3	1	IN THE PROPERTY AND THE
Breadboard	1	1
200 Ohm Resistor	1	

INPUT	OUTPUT
NONE	LED Colour Start to Flow Automatically for 2 Different Sequences



WHAT TO KNOW ABOUT LED:

- Also known as Light Emitting Diode (LED)
- Two legs/points [1 Anode (+) & 1 Cathode (-)]
- The colour of an LED is determined by the material used in the semiconducting element
- Most common colours (Red, Blue, Yellow, Orange, Green, White)
- The more current is being supply, the brighter it is
- Without a resistor to reduce the current, the shorter its lifespan will be

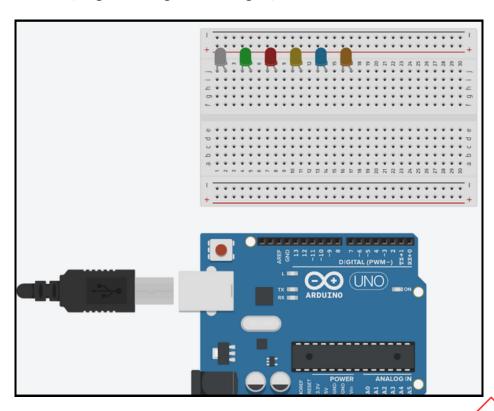






STEP 1:

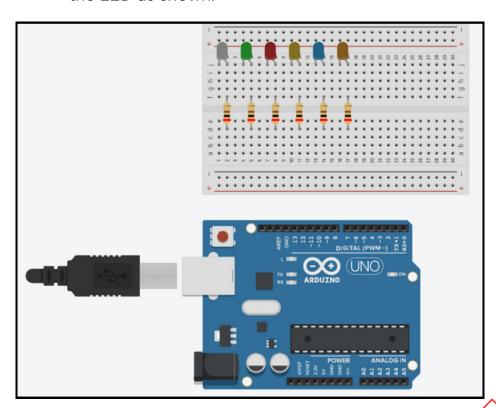
- Arrange Arduino UNO and Breadboard as below.
- Insert an LED into the breadboard with the Anode (positive leg) on the left and the Cathode (negative leg on the right).





STEP 2:

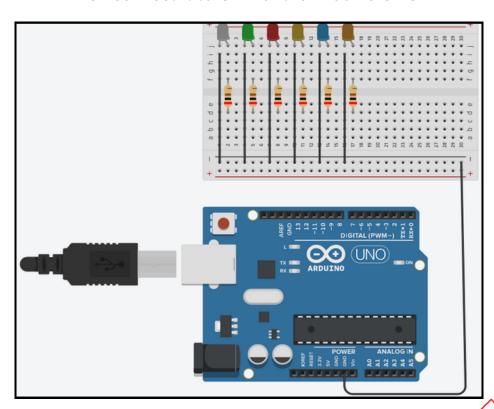
• Connect the 200 Ohm resistor to the Anode of the LED as shown.





STEP 3:

- Connect the Cathode of LED to -VE point of breadboard.
- Then connect it to GND of the Arduino UNO.

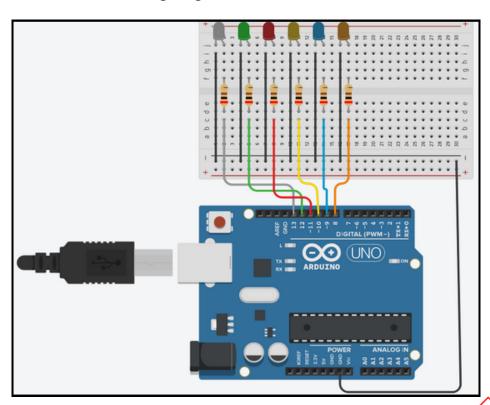






STEP 4:

- Connect the LEDs to PIN 13 to PIN 8 via the resistors as shown.
- Make sure it connect to the resistor to control the current going into the LEDs later on.





 Insert the following code (next page) inside code tab of Tinkercad project

```
Start Simulation
                                                                                                                                                         Send To
                                                                                                                         A△ -
                                                                                                                                      1 (Arduino Uno R3) •
                                                                                                        ± 📾
                                                            int leds[] = (13, 12, 11, 10, 9, 8);
int numLeds = 6;
                                                               void setup() {
  for (int i = 0; i < numLeds; pinMode(leds[i++], OUTPUT));</pre>
                                                               void loop() {
   // Run the first LED sequence: Flow Left to Right to Left
                                                                  flowLeftToRight();
                                                                   // Run the second LED sequence: Blink Odd to Even
                                                                  blinkOddToEven();
                                                                  // Increment the counter after every second run of the second sequesequenceCounter = (sequenceCounter + 1) % 3; // Reset counter aft // Increment array counter here if needed
                                                          void flowLeftToRight() {

22  // LED sequence: Flow Left to Right to Left

23  for (int i = 0; i < numLeds; i++) {
                                                                    digitalWrite(leds[i], HIGH);
                                                                    delay(50);
digitalWrite(leds[i], LOW);
                                                                     delay(50);
® Ի Ψ ϒ Ψ ™ N 및 Ç
DIGITAL (PWM~) H N
                                                                   delay(50);
                                                                   for (int i = numLeds - 1; i >= 0; i--) {
  digitalWrite(leds[i], HIGH);
                                                                     digitalWrite(leds[i], LOW);
                                                                     delay(50);
                                                                   delay(50);
                                                              void blinkOddToEven() {
   // LED sequence: Blink Odd to Even
   for (int j = 0; j < 2; j++) {
      for (int i = 0; i < numleds; digitalWrite(leds[i], HIOS), i += 2)
}</pre>
                                                                      for (int i = 0; i < numLeds; digitalWrite(leds[i], LOW), i += 2);
                                                                     for (int i = 1; i < numLeds; digitalWrite(leds[i], HIGH), i += 2
```



The code as shown in previous page:

```
1 int leds[] = {13, 12, 11, 10, 9, 8};
   int numLeds = 6;
 3
    int sequenceCounter = 0;
 4
 5
    void setup() {
     for (int i = 0; i < numLeds; pinMode(leds[i++], OUTPUT));
 7
 8
 9
    void loop() {
10
                     t LED sequence: Flow Left to Right to Left
 11
     flowLeftToRight();
12
13
      // Run the second LED sequence: Blink Odd to Even
14
     blinkOddToEven();
15
     // Increment the counter after every second run of the second sequence sequenceCounter = (sequenceCounter + 1) % 3; // Reset counter after running twice
16
17
18
     // Increment array counter here if needed
19
20
21
    void flowLeftToRight() {
22
                                   ft to Right to Left
23
24
     for (int i = 0; i < numLeds; i++) {
       digitalWrite(leds[i], HIGH);
25
       delay(50);
26
       digitalWrite(leds[i], LOW);
27
       delay(50);
28
29
     delay(50);
30
31
     for (int i = numLeds - 1; i >= 0; i--) {
32
       digitalWrite(leds[i], HIGH);
33
       delay(50);
34
       digitalWrite(leds[i], LOW);
35
       delay(50);
36
37
     delay(50);
38
```



39



Continue

```
40
    void blinkOddToEven() {
                               Odd to Even
41
42
43
     for (int j = 0; j < 2; j++) {
       for (int i = 0; i < numLeds; digitalWrite(leds[i], HIGH), i += 2);
44
       delay(50);
45
46
       for (int i = 0; i < numLeds; digitalWrite(leds[i], LOW), i += 2);
47
       for (int i = 1; i < numLeds; digitalWrite(leds[i], HIGH), i += 2);
48
49
       delay(50);
50
51
52
53
54
       for (int i = 1; i < numLeds; digitalWrite(leds[i], LOW), i += 2);
       delay(50);
55
```





STEP 6:

Run the simulation

```
00:01.355
                                                                                                                                                   Start / stop simulation
                                                                                                void loop() {
   // Run the first LED sequence: Flow Left to Right to Left
                                                                                                   flowLeftToRight();
                                                                                                    // Run the second LED sequence: Blink Odd to Even
                                                                                                   blinkOddToEven();
                                                                                                   // Increment the counter after every second run of the s
sequenceCounter = (sequenceCounter + 1) % 3; // Reset o
// Increment array counter here if needed
                                                                                            20
21 void flowLeftToRight() (
                                                                                                   // LED sequence: Flow Left to Right to Left
for (int i = 0; i < numLeds; i++) {
    digitalWrite(leds[i], HIGH);
    delse(S);</pre>
                                                                                                      delay(50);
                                                                                                      digitalWrite(leds[i], LOW);
                                                                                                      delay(50);
                                                                                                   delay(50);
                                                                                                   for (int i = numLeds - 1; i >= 0; i--) {
   digitalWrite(leds[i], HIGH);
                  aaabbhhhh aaaaaa
                                                                                                      digitalWrite(leds[i], LOW);
                                                                                                      delay(50);
                                                                                                   delay(50);
                      TX ARDUINO
                                                                                                void blinkOddToEven() {
                                                                                                   for int j = 0; i < numLeds; digitalWrite(leds[i], MIGR</pre>
                                                                                                      delay(50);
                                                                                                     for (int i = 0; i < numLeds; digitalWrite(leds[i], LOW)
                                                                                                      for (int i = 1; i < numLeds; digitalWrite(leds[i], HIGH
                                                                                                      delay(50);
                                                                                                      for (int i = 1; i < numLeds; digitalWrite(leds[i], LOW
                                                                                                      delay(50);
```



STEP 6:

 Please scan the QR Code below to see a simulation video of the project

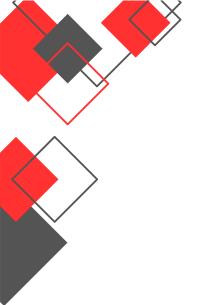




CONCLUSION:

As shown in the video demonstration of the simulation, the LED will run a flow of light show for 2 different sequences automatically. Students are encourage to tinker with the code and add some additional LEDs.





PROJECT IV LED FLOW (PUSH BUTTON)





DIFFICULTY LEVEL BEGINNER

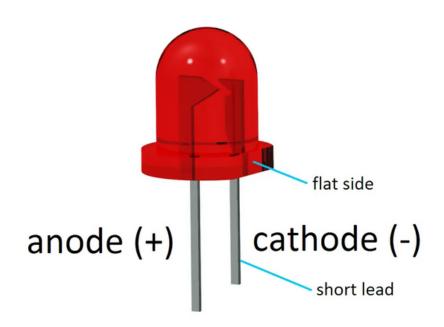
HARDWARE		
NEEDED	AMOUNT	
Single Coloured Light Emitting Diode (LED)	6	
Arduino UNO R3	1	THE RESERVE TO SERVE THE RESERVE THE RESER
Breadboard	1	
200 Ohm Resistor	6	
Push Button	1	

INPUT	OUTPUT
Button Pressed	 1st sequence of LED flow when button is pressed. 2nd sequence of LED flow when button is released.



WHAT TO KNOW ABOUT LED:

- Also known as Light Emitting Diode (LED)
- Two legs/points [1 Anode (+) & 1 Cathode (-)]
- The colour of an LED is determined by the material used in the semiconducting element
- Most common colours (Red, Blue, Yellow, Orange, Green, White)
- The more current is being supply, the brighter it is
- Without a resistor to reduce the current, the shorter its lifespan will be

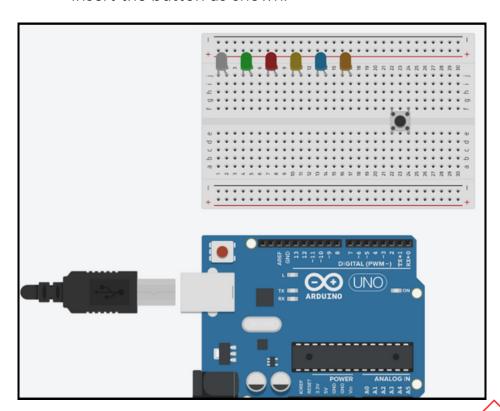






STEP 1:

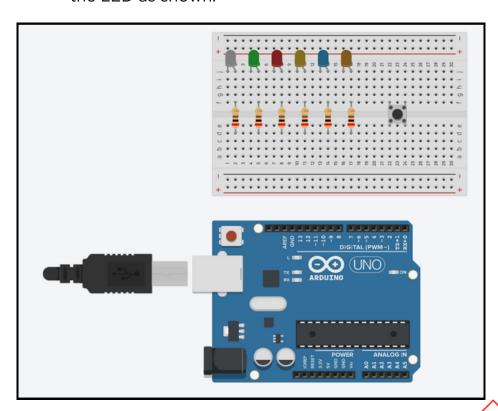
- Arrange Arduino UNO and Breadboard as below.
- Insert an LED into the breadboard with the Anode (positive leg) on the left and the Cathode (negative leg on the right).
- Insert the button as shown.





STEP 2:

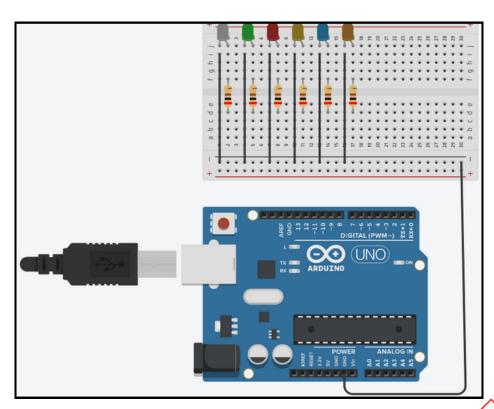
• Connect the 200 Ohm resistor to the Anode of the LED as shown.





STEP 3:

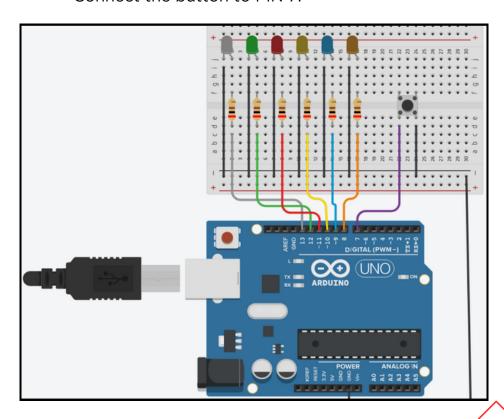
- Connect the Cathode of LED to -VE point of breadboard.
- Connect the button to -VE point of the breadboard.
- Then connect it to GND of the Arduino UNO.





STEP 4:

- Connect the LEDs to PIN 13 to PIN 8 via the resistors as shown.
- Make sure it connect to the resistor to control the current going into the LEDs later on.
- Connect the button to PIN 7.





 Insert the following code (next page) inside code tab of Tinkercad project

```
Start Simulation
                                                                            A4 -
                                                                                             1 (Arduino Uno R3)
                                                         <u>+</u>
       int leds[] = {13, 12, 11, 10, 9, 8};
       int numLeds = 6;
const int Button Pin = 7;
       void setup() {
  for (int i = 0; i < numLeds; pinMode(leds[i++], OUTFUT));
  pinMode(Button_Pin, INFUT_FULLUP);</pre>
          int buttonState = digitalRead(Button_Pin);
if (buttonState == LOW) {
              flowLeftToRight();
              blinkOddToEven():
19 void flowLeftToRight() {
20  for (int i = 0; i < numLeds; i++) {
21   digitalWrite(leds[i], HIGH);
22   digitalWrite(leds[i], HIGH);
             delay(100);
digitalWrite(leds[i], LOW);
delay(100);
         delay(100);
          for (int i = numLeds - 1; i >= 0; i--) {
              digitalWrite(leds[i], HIGH);
              delay(100);
digitalWrite(leds[i], LOW);
              delay(100);
         delay(100);
       void blinkOddToEven() {
          old Diamodologeen() (
for (int) = 0; j < 2; j++) {
for (int) = 0; j < numLeds; digitalWrite(leds[i], HIGH), i
delay(ISO);
for (int i = 0; i < numLeds; digitalWrite(leds[i], LOW), i
for (int i = 1; i < numLeds; digitalWrite(leds[i], HIGH), i
for (int i = 1; i < numLeds; digitalWrite(leds[i], HIGH), i
              delay(150);
for (int i = 1; i < numLeds; digitalWrite(leds[i], LOW), i</pre>
              delay(150);
```



The code as shown in previous page:

```
1 int leds[] = {13, 12, 11, 10, 9, 8};
2 int numLeds = 6;
   const int Button_Pin = 7;
45
   void setup() {
6
    for (int i = 0; i < numLeds; pinMode(leds[i++], OUTPUT));
7
8
    pinMode(Button_Pin, INPUT_PULLUP);
9
10
   void loop() {
    int buttonState = digitalRead(Button Pin);
11
12
    if (buttonState == LOW) {
13
     flowLeftToRight();
14
    } else {
     blinkOddToEven();
15
16
17 }
18
```





· Continue:

```
19 void flowLeftToRight() {
     for (int i = 0; i < numLeds; i++) {
20
21
       digitalWrite(leds[i], HIGH);
22
       delay(100);
23
24
25
26
27
28
29
30
31
       digitalWrite(leds[i], LOW);
       delay(100);
     delay(100);
     for (int i = numLeds - 1; i \ge 0; i \ge 0
       digitalWrite(leds[i], HIGH);
       delay(100);
       digitalWrite(leds[i], LOW);
32
33
34
       delay(100);
     delay(100);
35 } 36
```





Continue:

```
void blinkOddToEven() {
37
38
     for (int j = 0; j < 2; j++) {
      for (int i = 0; i < numLeds; digitalWrite(leds[i], HIGH), i += 2);
39
      delay(150);
40
41
      for (int i = 0; i < numLeds; digitalWrite(leds[i], LOW), i += 2);
      for (int i = 1; i < numLeds; digitalWrite(leds[i], HIGH), i += 2);
42
43
      delay(150);
      for (int i = 1; i < numLeds; digitalWrite(leds[i], LOW), i += 2);
44
45
      delay(150);
46
47 }
48
```





STEP 6:

Run the simulation

```
1 (Arduino Uno R
                                                                                                                I > 0°
                                              int leds[] = {13, 12, 11, 10, 9, 8};
int numLeds = 6;
                                                   const int Button_Pin = 7;
                                                 void setup() {
  for (int i = 0; i < numLeds; pinMode(leds[i++], CUTFUT));
  pinMode(Button_Pin, INFUT_FULLUF);
}</pre>
                                                  void loop() {
  int buttonState = digitalRead(Button_Pin);
  if (buttonState == LOW) {
   flowLeftToRight();
}
                                                         blinkOddToEven();
3 *
                                           18
19 void flowLeftToRight() {
                                                     for (int i = 0; i < numLeds; i++) {
   digitalWrite(leds[i], HIGH);</pre>
                                                         delay(100);
digitalWrite(leds[i], LOW);
                                                         delay(100);
                                                     delay(100);
                                                     for (int i = numLeds - 1; i >= 0; i--) {
   digitalWrite(leds[i], HIGH);
                                                        delay(100);
digitalWrite(leds[i], LOW);
                                                         delay(100);
                                                     delay(100);
                                                 void blinkOddToEven() {
for (int j = 0; j < 2; j++) {
for (int i = 0; i < numLeda; digitalNrite(leds[i], HIGH),
delay(150);
for (int <math>i = 0; i < numLeda; digitalNrite(leds[i], LOW), i
for (int <math>i = 1; i < numLeda; digitalNrite(leds[i], HIGH),
delay(150);
}
                                                         delay(150);
for (int i = 1; i < numLeds; digitalWrite(leds[i], LOW), i</pre>
                                                         delay(150);
                                            47 ]
                                          Serial Monitor
```



STEP 6:

 Please scan the QR Code below to see a simulation video of the project

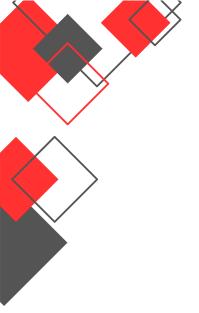




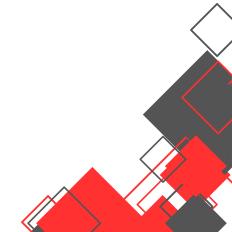
CONCLUSION:

As shown in the video demonstration of the simulation, the LED will run a flow of light show for 2 different sequences automatically. 1st sequence will play when the button is pressed and when the button is released, 2nd sequence will play. Students are encourage to tinker with the code and add some additional LEDs.





PROJECT V LED RGB BLINK COLOUR CHANGE (AUTO)





HARDWARE NEEDED	AMOUNT	
RGB LED	1	
Arduino UNO R3	1	THE TAX AMAZON A
Breadboard	1	1
200 Ohm Resistor	3	

INPUT	OUTPUT
NONE	RGB LED will blink 3 different colours randomly with interval or 1 seconds.



WHAT TO KNOW ABOUT RGB LED:

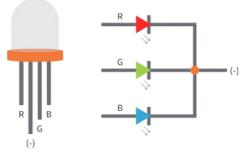
- They are two types of RGB LED:
 - Common Cathode
 - Common Anode
- It comes 4 legs/pin. 1 for Red, 1 for Green, 1 Blue and 1 for either Cathode (-) or Anode (+).
- User can basically either create Red colour light, or green coloured light or blue coloured light.
- User can also create combination colour from two or three of the basic light (Red, Green, Blue) to make a whole new colour of light.
- The combination of light is endless.





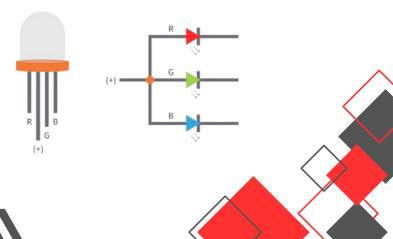
COMMON CATHODE RGB LED

- In an RGB LED with a common cathode configuration, the internal LEDs share a common cathode connected to the external cathode lead.
- To individually control each color, you must provide a HIGH signal or connect to VCC for the red, green, and blue leads, while connecting the anode lead to the negative terminal of the power supply.



COMMON CATHODE RGB LED

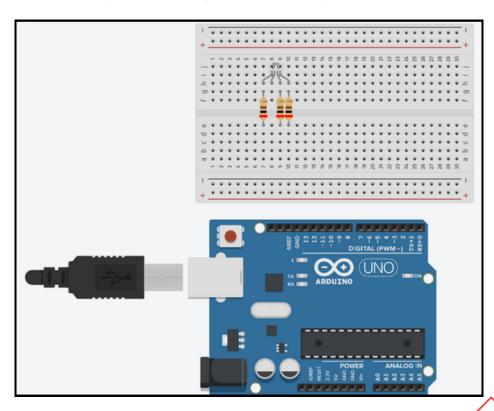
- In an RGB LED with a common anode configuration, the internal LEDs share a common anode connected to the external anode lead.
- To individually control each color, you must provide a LOW signal or ground for the red, green, and blue leads, while connecting the anode lead to the positive terminal of the power supply.





STEP 1:

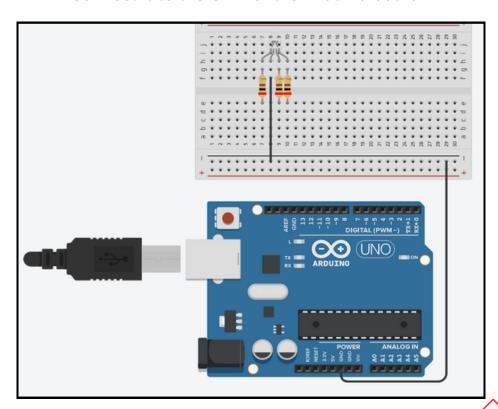
- Arrange Arduino UNO and Breadboard as below.
- Insert an RGB LED into the breadboard with the.
- Connect 200 Ohm resistor on each of the RGB LED pin excluding the common pin.





STEP 2:

- Connect the RGB LED common pin to the -VE of the breadboard.
- Connect it to the GND of the Arduino board.

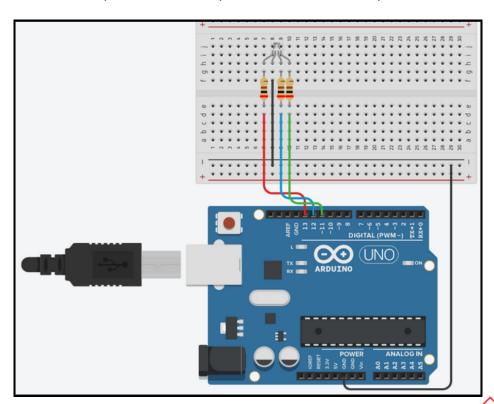






STEP 3:

- Connect the RGB LED pins to PIN 13, 12 and 11 on the Arduino board.
- Red pin to 12, Blue pin to 12 and Green pin to 11.





 Insert the following code (next page) inside code tab of Tinkercad project

```
    Start Simulation

                                                                                                                                                                                                   Send To
                                                                                                                                                                            1 (Arduino Uno R3) •
                                                                                                                                          +
                                                                                                                                                          A^ -
                                                                                           // Define pin numbers for Red, Green, and Blue LEDs #define R 13 #define G 12
                                                                                           // Define RGB array with Red, Green, Blue, Yellow, Purple, and White int rgb[] = (255, 0, 0, // Red 0, 255, 0, 0, // Green 0, 0, 255, // Blue 255, 255, 0, // Yellow 255, 0, 255, // Purple 255, 255, 255, // White
                                                                                           void setup() {
                                                                                              // Set the pin modes to OUTPUT for RGB LEDs
pinMode(R, OUTPUT);
                                                                                               pinMode(G, OUTPUT);
pinMode(B, OUTPUT);
                                                                                      21 void loop() (
                                                                                              // Loop through the RGB array and set each color with a 100ms inter
for (ant i = 0; i < 10; i + 3) (setColor(rgb[i), rgb[i + 1], rgb[i + 2]);
                                                                                                   delay(500);
                                                                                      29 // Function to set RGB color
30 void setColor(int red, int green, int blue) {
                                                                                              analogWrite(R, red);
                                                                                              analogWrite(G, green);
analogWrite(B, blue);
2388 82222
```



The code as shown in previous page:

```
Define pin numbers for R=Red, G=Green, and B=Blue LEDs
   #define R 13
   #define G 12
 3
4
5
   #define B 11
 6
    // Define RGB array with Red, Green, Blue, Yellow, Purple, and White
     ralues
    int rgb[] = {255, 0, 0, // Red
 8
            0, 255, 0,
            0, 0, 255,
 9
            255, 255, O, <mark>/</mark>
10
11
            255, 0, 255, //
            255, 255, 255}; // White
12
13
14
   void setup() {
     // Set the pin modes to OUTPUT for RGB LEDs pinMode(R, OUTPUT);
15
16
     pinMode(G, OUTPUT);
17
     pinMode(B, OUTPUT);
18
19 }
20
```





Continue:

```
21 void loop() {
22
23
24
      // Loop through the RGB array and set each color with a 100ms
    interval
25
26
27
28
29
      for (int i = 0; i < 18; i += 3) {
       setColor(rgb[i], rgb[i + 1], rgb[i + 2]);
       delay(500);
30
31 // Function to set RGB color32 void setColor(int red, int green, int blue) {
33
      analogWrite(R, red);
      analogWrite(G, green); analogWrite(B, blue);
34
35
36 }
37
```









STEP 5:

Run the simulation

```
Code Stop Simulation
                                                                                                                                                                    IÞ
                                                                                                                                                                                       1 (Arduino Uno
                                                                                                            1 // Define pin numbers for Red, Green, and Blue LEDs
2 #define R 13
#define G 12
                                                                                                               // Define RGB array with Red, Green, Blue, Yellow, Purple, a int rgb[] = {255, 0, 0, // Red 0, 255, 0, // Green 0, 0, 255, // Blue 255, 255, 0, Yellow 255, 0, 255, // Furple 255, 255, 255, // White
                                                                                                          14 void setup() {
                                                                                                         pinMode(R, OUTPUT);
pinMode(G, OUTPUT);
pinMode(B, OUTPUT);
pinMode(B, OUTPUT);
                                                                                                                 // Set the pin modes to OUTPUT for RGB LEDs
pinMode(R, OUTPUT);
                                                                                                              void loop() {
   // Loop through the RGB array and set each color with a 10
for (int i = 0; i < 18; i += 3) {
   setColor(rgb[i], rgb[i + 1], rgb[i + 2]);
}</pre>
                                                                                                                      delay(500);
           REPORT SERVICES
                // Function to set RGB color

void setColor(int red, int green, int blue) {
    analogNrite(R, red);
    analogNrite(G, green);
    analogNrite(B, blue);
}
                TX ARDUINO UNO
                           POWER ANALOGIN
                                                                                                         Serial Monitor
```



STEP 6:

 Please scan the QR Code below to see a simulation video of the project

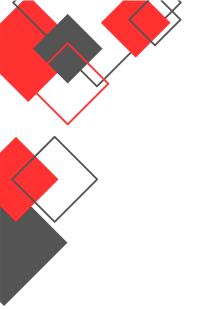




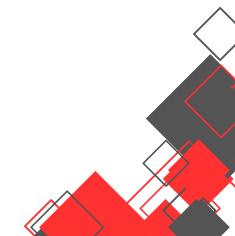
CONCLUSION:

As shown in the video demonstration of the simulation, the RGB LED will blink colour from Red, Green, Blue, Purple, Yellow and White. Students are encourage to tinker with the code and add some additional LEDs.





PROJECT VI LED RGB BLINK COLOUR CHANGE (PUSH BUTTON)





DIFFICULTY LEVEL BEGINNER

HARDWARE NEEDED	AMOUNT	
RGB LED	1	
Arduino UNO R3	1	THE RESERVE THE PARTY OF THE PA
Breadboard	1	
200 Ohm Resistor	3	
Push Button	1	

INPUT	OUTPUT
NONE	RGB LED will blink 3 different colours randomly with interval or 1 seconds.



WHAT TO KNOW ABOUT RGB LED:

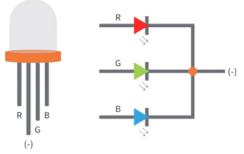
- They are two types of RGB LED:
 - Common Cathode
 - Common Anode
- It comes 4 legs/pin. 1 for Red, 1 for Green, 1 Blue and 1 for either Cathode (-) or Anode (+).
- User can basically either create Red colour light, or green coloured light or blue coloured light.
- User can also create combination colour from two or three of the basic light (Red, Green, Blue) to make a whole new colour of light.
- The combination of light is endless.





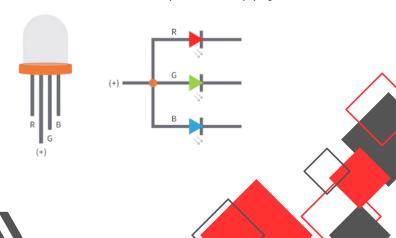
COMMON CATHODE RGB LED

- In an RGB LED with a common cathode configuration, the internal LEDs share a common cathode connected to the external cathode lead.
- To individually control each color, you must provide a HIGH signal or connect to VCC for the red, green, and blue leads, while connecting the anode lead to the negative terminal of the power supply.



COMMON CATHODE RGB LED

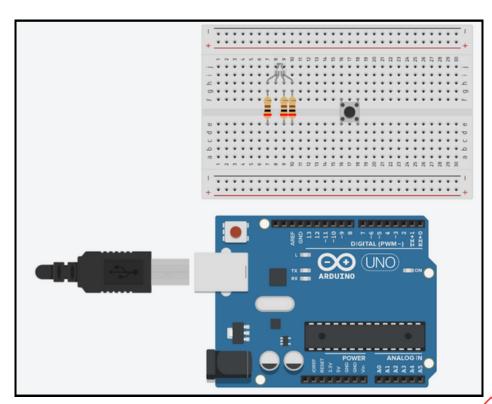
- In an RGB LED with a common anode configuration, the internal LEDs share a common anode connected to the external anode lead.
- To individually control each color, you must provide a LOW signal or ground for the red, green, and blue leads, while connecting the anode lead to the positive terminal of the power supply.





STEP 1:

- Arrange Arduino UNO and Breadboard as below.
- Insert an RGB LED into the breadboard with the.
- Connect 200 Ohm resistor on each of the RGB LED pin excluding the common pin.
- Add push button on the breadboard.



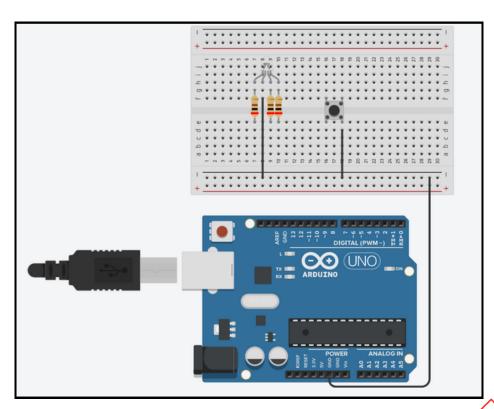






STEP 2:

- Connect the RGB LED common pin to the -VE of the breadboard.
- Connect push button to -VE of the breadboard.
- Connect it to the GND of the Arduino board.

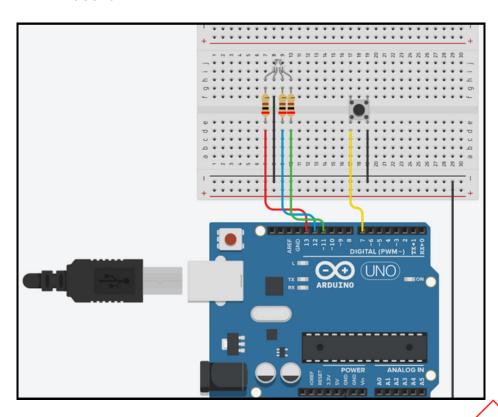






STEP 3:

- Connect the RGB LED pins to PIN 13, 12 and 11 on the Arduino board.
- Red pin to 12, Blue pin to 12 and Green pin to 11.
- Connect push button to PIN 7 on the Arduino board.





 Insert the following code (next page) inside code tab of Tinkercad project

```
1 (Arduino Uno R3) •
                                                                                                               +
                                                                                                                               AQ =
                                                                       // Define pin numbers for Red, Green, and Blue LEDs #define R 13 \,
                                                                      #define G 11
                                                                    4 #define B 12
                                                                       // Define the pin for the button
const int buttonPin = 7;
                                                                         // Set the pin modes to CUTPUT for RGB LEDs
pinNode(R, CUTPUT);
pinNode(G, CUTPUT);
pinNode(B, CUTPUT);
                                                                  15 // Set button pin as INPUT PULLUP; use internal pull-up resistor 17 pinKode(buttonFin, INPUT_PULLUP);
                                                                  y void loop() {

// Check if the button is pressed

if (digitalRead(buttonPin) == LOW) {
                                                                             // If the button is pressed, toggle the color
toggleColor();
delay(500); // Debounce delay to avoid multiple readings for a
                                                                 25 26 }
DIGITAL (PWM-)
                                                                  28 // Function to toggle RGB color on and off
                                                                  29 void toggleColor() (
30 for (int i = 0; i < 10; ++i) { // Blink 10 times
                                                                                  Generate random values for Red, Green, and Blue components
                                                                            int redValue = random(256);
int greenValue = random(256);
                                                                             int blueValue = random(256);
                                                                             // Set RGB LED to the random color and make it blink
                                                                             analogWrite(R, redValue);
                                                                             analogWrite(G, greenValue);
analogWrite(B, blueValue);
                                                                             delay(500); // On time
                                                                             analogWrite(R, 0);
analogWrite(G, 0);
analogWrite(B, 0);
                                                                             delay(500); // Off time
```



The code as shown in previous page:

```
1 // Define pin numbers for R=Red, G=Green, and B=Blue LEDs
 2 #define R 13
 3 #define G 11
 4 #define B 12
 5
   // Define the pin for the button
 7
   const int buttonPin = 7;
 8
 9 void setup() {
            he pin modes to OUTPUT for RGB LEDs
10
    pinMode(R, OUTPUT):
11
12
    pinMode(G, OUTPUT);
13
    pinMode(B, OUTPUT);
14
    // Set button pin as INPUT_PULLUP; use internal pull-up resistor
15
    pinMode(buttonPin, INPUT_PULLUP);
16
17 }
18
19
   void loop() {
         neck if the button is pressed
20
21
    if (digitalRead(buttonPin) == LOW) {
22
23
        If the button is pressed, toggle the color
     toggleColor();
24
     delay(500); // Debounce delay to avoid multiple readings for a
   single press
25
26 }
27
```





Continue:

```
28 // Function to toggle RGB color on and off
29 void toggleColor() {
    for (int i = 0; i < 10; ++i) { // Blink 10 times
30
     // Generate random values for Red, Green, and Blue components
int redValue = random(256);
31
32
33
      int greenValue = random(256);
34
      int blueValue = random(256);
35
     // Set RGB LED to the random color and make it blink
36
37
     analogWrite(R, redValue);
38
     analogWrite(G, greenValue);
39
      analogWrite(B, blueValue);
40
41
      delay(500); // On time
42
      // Turn off the RGB LED
43
44
     analogWrite(R, 0);
45
     analogWrite(G, 0);
46
      analogWrite(B, 0);
47
     delay(500); // Off time
48
49
50 }
51
```





STEP 5:

Run the simulation

```
Code Stop Simulation
                                                                                                                                        I ► 1 (Arduino Uno R
                                                                                         // Define pin numbers for Red, Green, and Blue LEDs
                                                                                          #define R 13
                                                                                          #define B 12
                                                                                          // Define the pin for the button
const int buttonPin = 7;
                                                                                          void setup() {
                                                                                            // Set the pin modes to CUTPUT for RGB LEDs
pinMode(R, CUTPUT);
pinMode(G, CUTPUT);
                                                                                            pinMode(B, OUTPUT);
                                                                                     15 // Set button pin as INPUT_FULLUP; use internal pull-up resi

15 pinMode(buttonPin, INFUT_FULLUP);

17 }
                                                                                         void loop() {
    // Check if the button is pressed
if (digitalRead(buttonFin) == LOW) {
    // If the button is pressed, toggle the color
    toggleColor();
    delay(500); // Debounce delay to avoid multiple readings
26 }
27
                                                                                    TX ARDUINO UNO
                                                                                               // Set RGB LED to the random color and make it blink
                                                                                               analogWrite(R, redValue);
analogWrite(G, greenValue);
analogWrite(B, blueValue);
                                                                                               delay(500); // On time
                                22222
                                                                                               // Turn off the RGB LED
                                                                                               analogWrite(R, 0);
analogWrite(G, 0);
analogWrite(B, 0);
                                                                                               delay(500); // Off time
```



STEP 6:

 Please scan the QR Code below to see a simulation video of the project





CONCLUSION:

As shown in the video demonstration of the simulation, the RGB LED will blink random colour for ten times once the push button was pressed. Students are encourage to tinker with the code and add some additional LEDs.



REFERENCES

https://www.tinkercad.com/

https://www.arduino.cc/

https://randomnerdtutorials.com/ electronics-basics-how-do-rgbleds-work/

https://byjus.com/physics/lightemitting-diode/

https://projecthub.arduino.cc/se msemharaz/interfacing-rgb-ledwith-arduino-b59902

A COLLECTION OF TINKERCAD-ABLE PROJECTS: VOLUME I

e ISBN 978-967-2097-87-7



POLITEKNIK MUKAH (online)