

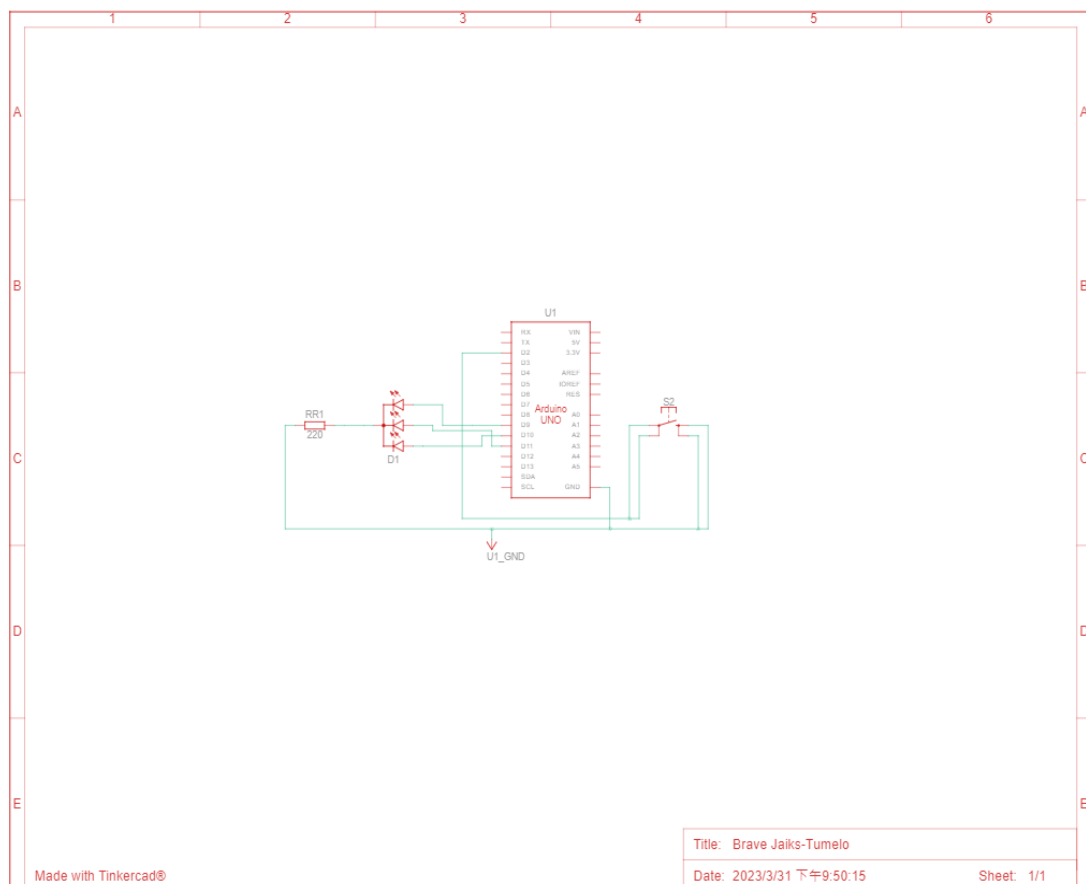
1. 使用一個 RGB 全彩 LED 接腳分別指定到 PWM 接腳 9,10,11 然後使用一個觸控開關來控制，每按一次就隨機出現一種顏色燈光。除了輸出到 LED，同時也要將值列印在串列埠監控視窗。

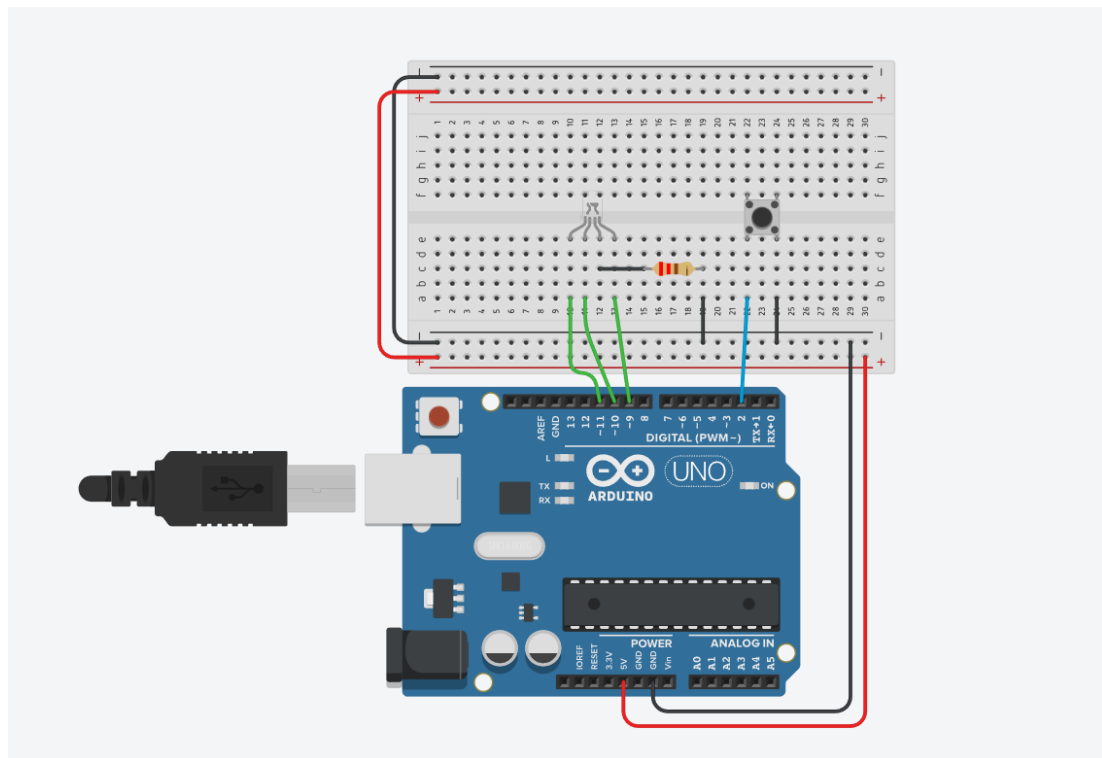
## ● 實驗材料

## Component List

Name	Quantity	Component
U1	1	Arduino Uno R3
D1	1	LED RGB
RR1	1	220 Ω Resistor
S2	1	Pushbutton

## ● 電路圖





## ● 程式碼

Code

Start Simulation

Send To

Text

1

2

3

4

5

6

7

8

9

10

11

12

13

14

15

16

17

18

19

20

21

22

23

24

25

26

27

28

29

30

31

32

33

34

35

```

/** 範例5.7.1(RGB全彩LED) */
const int Rpin=9, Gpin=10, Bpin=11; //將接腳分別指定到PWM接腳9,10,11
const int SW=2;
int i,j,k;

void setup() {
  Serial.begin(9600);
  pinMode(SW, INPUT_PULLUP);
  pinMode(Rpin, OUTPUT);
  pinMode(Gpin, OUTPUT);
  pinMode(Bpin, OUTPUT);
  randomSeed(analogRead(A0));
  color(0,0,0);
}

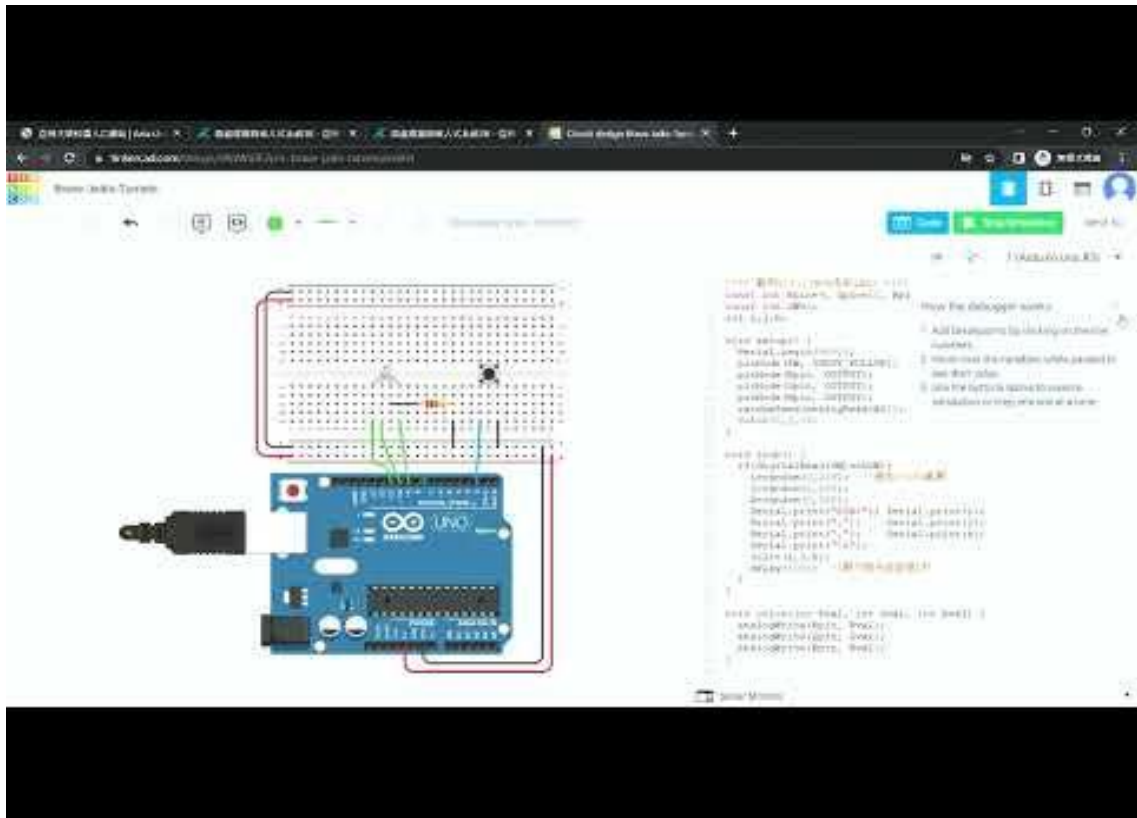
void loop() {
  if(digitalRead(SW)==LOW){
    i=random(0,256); //產生0-255亂數
    j=random(0,256);
    k=random(0,256);
    Serial.print("RGB="); Serial.print(i);
    Serial.print(","); Serial.print(j);
    Serial.print(","); Serial.print(k);
    Serial.print("\n");
    color(i,j,k);
    delay(500); //顯示燈光並延遲1秒
  }
}

void color(int Rval, int Gval, int Bval) {
  analogWrite(Rpin, Rval);
  analogWrite(Gpin, Gval);
  analogWrite(Bpin, Bval);
}

```

Serial Monitor

- 錄製影片



2. 使用一位七段顯示器顯示數字0~9 然後使用一個觸控開關來控制，每按奇數次就遞加偶數次就遞減。

- 實驗材料

#### Component List

Name	Quantity	Component
U1	1	Arduino Uno R3
Digit1	1	Anode 7 Segment Display
S1	1	Pushbutton
R1	1	330 $\Omega$ Resistor

●



Date: 2023/4/1 下午5:06:58

Sheet: 1/1



- 程式碼

Code

Start Simulation

Send To

Text

1 (Arduino Uno R3)

```
1  int i;
2  int KeyData;
3  int numKeys=0;
4  int val=0;
5  const int debounceDelay=20;
6  const byte num[10]={
7      B11000000,    //0
8      B11111001,    //1
9      B10100100,    //2
10     B10110000,    //3
11     B10011001,    //4
12     B10010010,    //5
13     B10000010,    //6
14     B11111000,    //7
15     B10000000,    //8
16     B10010000    //9
17 };
18 const int seg[]={2,3,4,5,6,7,8,9}; //abcdefgp
19 const int sw=10;
20 void setup()
21 {
22     pinMode(sw,INPUT_PULLUP);
23     for(i=0;i<8;i++)
24         pinMode(seg[i],OUTPUT);
25 }
26 void loop()
27 {
28     KeyData=digitalRead(sw);
29     if(KeyData==LOW)
30     {
31         delay(debounceDelay);
```

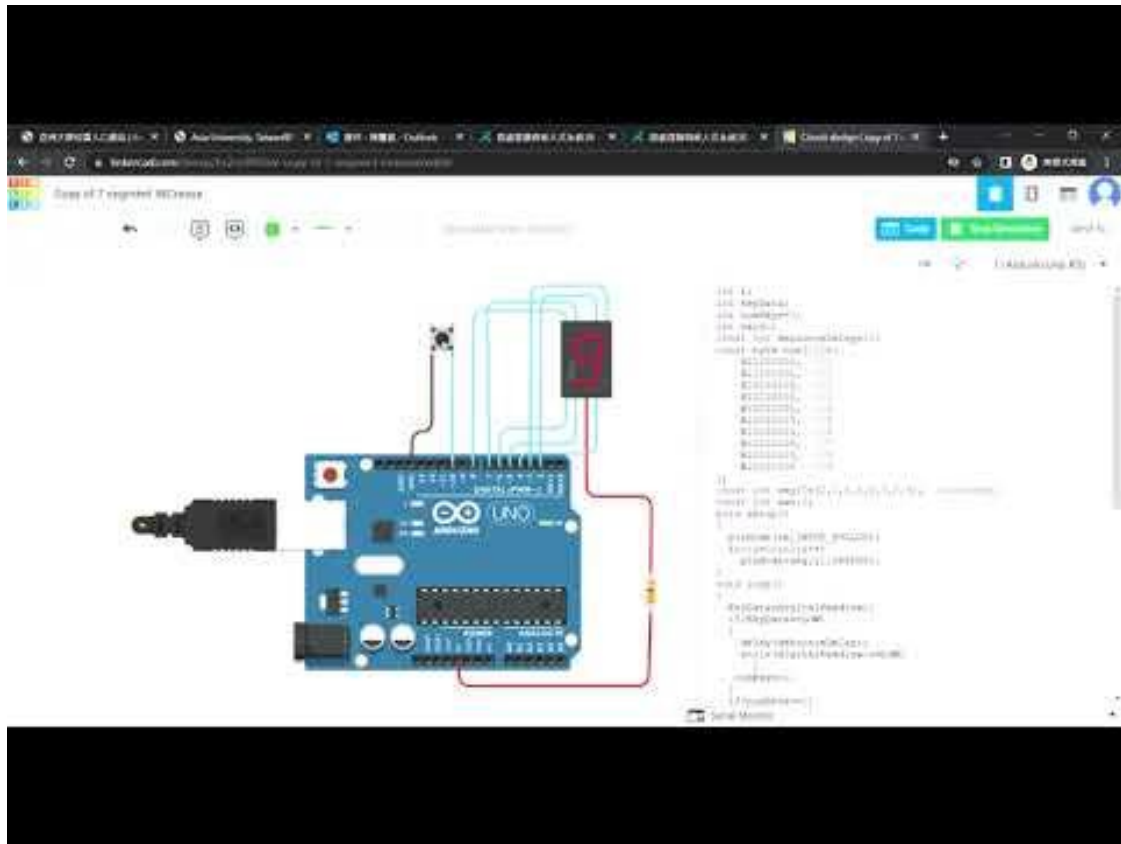
Text

1 (Arduino Uno R3)

```
32     while(digitalRead(sw)==LOW)
33     {
34         numKeys=1;
35     }
36     if(numKeys==1)
37     {
38         if (val==0)
39         {
40             val=9;
41         }else
42         {
43             val--;//
44         }
45     }
46
47     /* if (val==9)
48     {
49         val=0;
50     }else
51     {
52         val++;//
53     }*/
54
55 }
56
57 for(i=0;i<8;i++)
58 {
59     if(bitRead(num[val],i))
60         digitalWrite(seg[i],HIGH);
61     else
62         digitalWrite(seg[i],LOW);
63 }
64 delay(100);
65 numKeys=0;
66 }
67
```

Serial Monitor

- 錄製影片



- 反思(第一題)

這個項目旨在使用 Arduino 控制 RGB LED 燈，透過觸控開關來調整燈光的顏色和強度。RGB LED 由三個顏色通道組成，分別是紅色、綠色和藍色，可通過 PWM 調節各通道的亮度，從而產生不同的顏色。

首先，我們需要將 RGB LED 的三個引腳分別連接到 Arduino 的 PWM 引腳 9、10、11 上。接著，我們將使用觸控開關來控制燈光的顏色和強度。每次按下開關，燈光的顏色會隨機變化，我們可以使用 Arduino 的隨機數函數來實現這個功能。同時，在燈光變化時，我們還需要將 RGB 值輸出到 Serial Monitor 中，這可以使用 Serial.print() 函數來完成。

透過這個項目，我們能夠更深入地了解 Arduino 的 PWM 控制和隨機數函數的使用，同時學會了如何使用觸控開關來控制燈光。通過這些知識和技能的應用，我們能夠設計和實現更加複雜的控制系統，同時深入了解電路控制的原理和方法。總而言之，這個項目為我們提供了一個實用且有趣的學習平台，讓我們對於 Arduino 的控制能力更有自信，同時也擴展了我們的技能和知識。

- 反思(第二題)

在這個專案中，我學到了如何使用單獨七段顯示器顯示數字0~9，同時透過一個觸摸開關來控制顯示器顯示的數字。這個專案讓我更深入地了解了 **Arduino** 的控制能力，並學會了如何利用 **Arduino** 來控制電子元件的行為。

在這個專案中，我學到了如何使用七段顯示器來顯示數字。我需要將七段顯示器的不同接腳連接到 **Arduino** 的不同腳位上，然後利用程式來控制七段顯示器的行為。透過控制每一個腳位的高低電位，可以讓七段顯示器顯示不同的數字。我還學習了如何使用延遲函數來控制顯示器顯示每一個數字的時間，以達到更好的顯示效果。

在這個專案中，我還學習了如何使用觸摸開關來控制顯示器顯示的數字。我需要透過程式設計來判斷按下觸摸開關的次數，並根據奇偶性來遞增或遞減顯示器顯示的數字。這樣的設計可以讓使用者更加方便地控制顯示器顯示的數字，適用於不同的應用場景。

透過這個專案的實作，我也學會了如何運用不同的元件和函數來實現一個簡單的控制系統，同時也讓我對於電子電路有了更深入的了解。總而言之，這個專案讓我對於 **Arduino** 的控制能力更有信心，也讓我更加熟悉了解電路控制的知識。我相信這些知識和經驗對於我未來的學習和職業發展都會有所幫助。