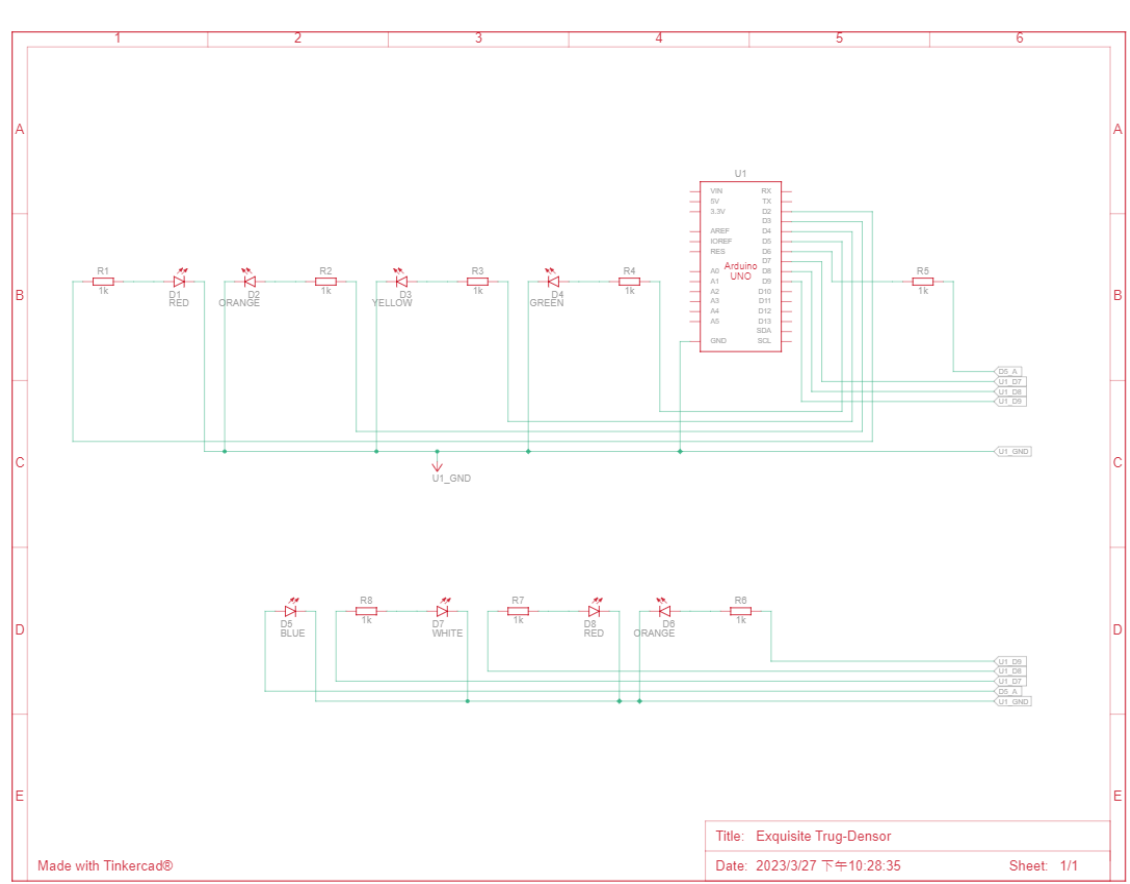


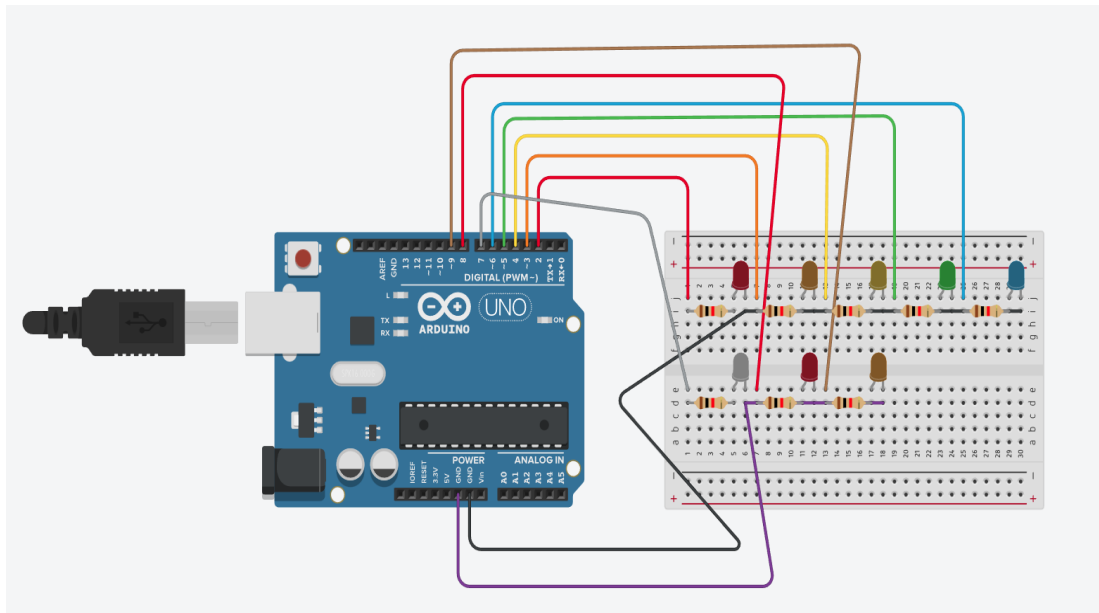
1. 使用 8 個 LED，控制 LED 能向左向右來回循環亮燈。每秒亮一顆。

● 實驗材料

Name	Quantity	Component
U1	1	Arduino Uno R3
D1 D8	2	Red LED
D2 D6	2	Orange LED
D3	1	Yellow LED
D4	1	Green LED
R1 R2 R3 R4 R5 R6 R7 R8	8	1 kΩ Resistor
D5	1	Blue LED
D7	1	White LED

● 電路圖





● 程式碼

Code
Start Simulation
Send To

Text
1 (Arduino Uno R3)

```

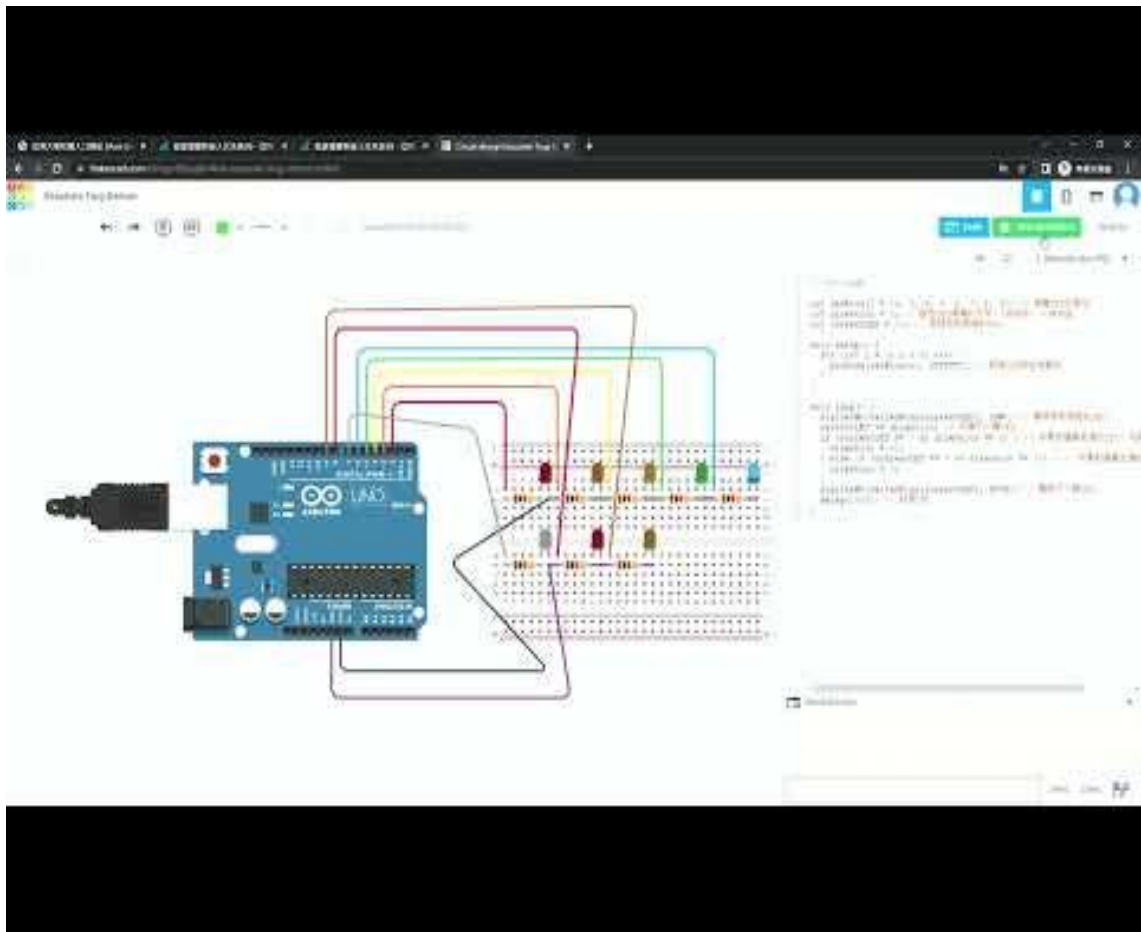
1 // C++ code
2
3 int ledPins[] = {2, 3, 4, 5, 6, 7, 8, 9}; // 定義LED的腳位
4 int direction = 1; // 設定LED循環的方向，1為向右，-1為向左
5 int currentLED = -1; // 記錄目前亮燈的LED
6
7 void setup() {
8     for (int i = 0; i < 8; i++) {
9         pinMode(ledPins[i], OUTPUT); // 設定LED腳位為輸出
10    }
11 }
12
13 void loop() {
14     digitalWrite(ledPins[currentLED], LOW); // 關閉目前亮燈的LED
15     currentLED += direction; // 切換下一顆LED
16     if (currentLED == 7 && direction == 1) { // 如果到達最右邊的LED，切換方向
17         direction = -1;
18     } else if (currentLED == 0 && direction == -1) { // 如果到達最左邊的LED，切換方向
19         direction = 1;
20     }
21     digitalWrite(ledPins[currentLED], HIGH); // 開啟下一顆LED
22     delay(1000); // 延遲1秒
23 }
24

```

Serial Monitor

Send
Clear

- 錄製影片

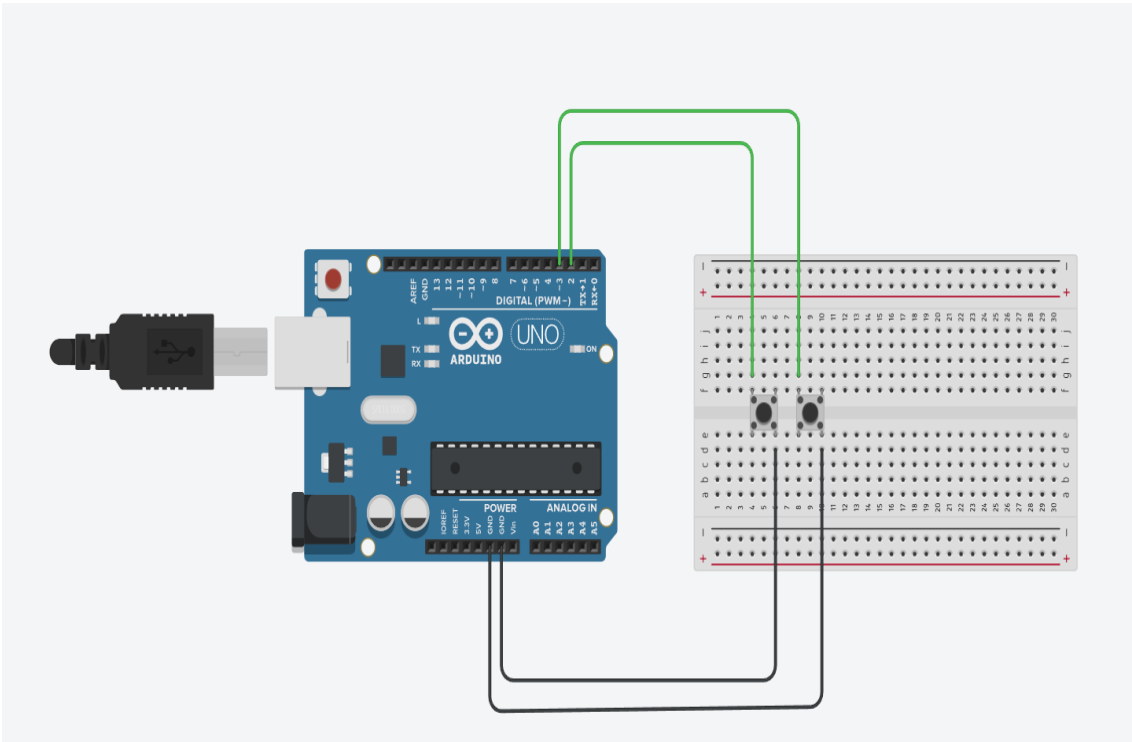
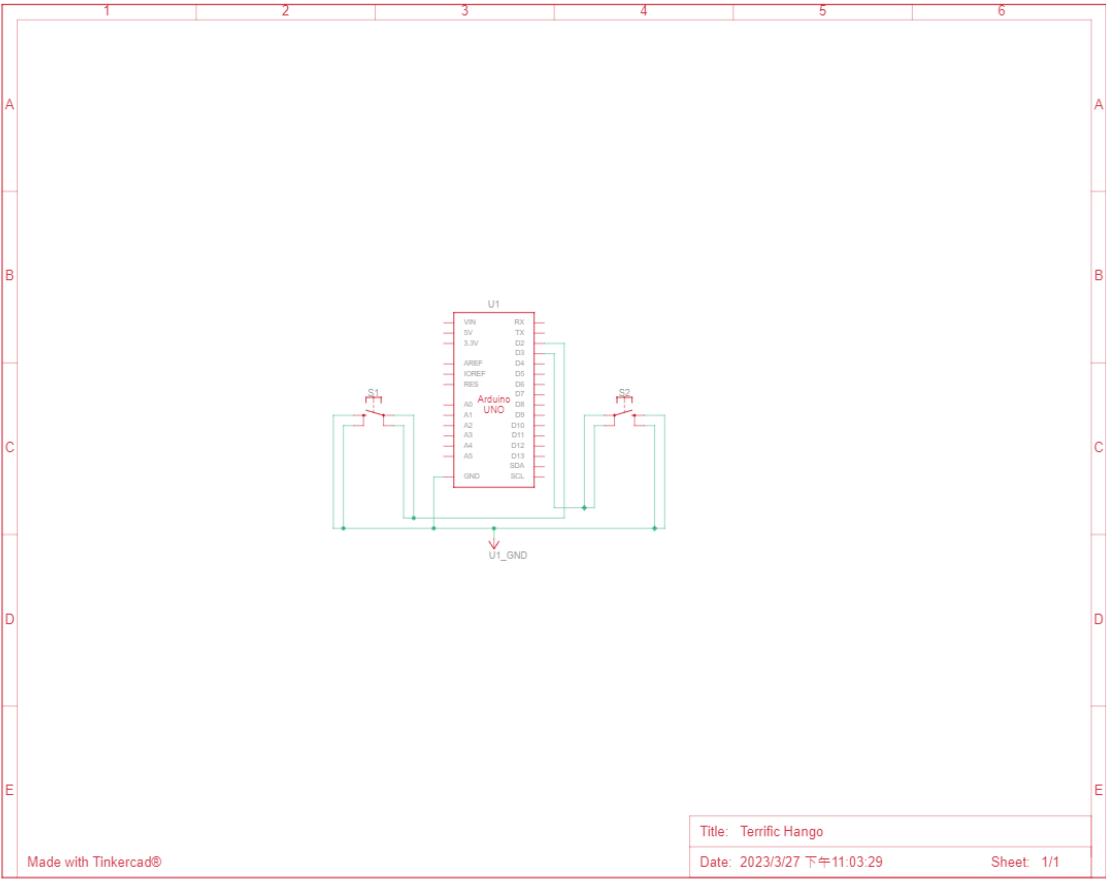


2. 使用二個按鈕開關 A 與 B，分別接到 D2 跟 D3 接腳，實做出一個具有上數跟下數功能的計數器，當使用者按壓開關 A 一次計數值就會加 1，按壓開關 B 一次計數值就會減 1，更新後的數值會即時的顯示在 PC 端的串列埠視窗。

- 實驗材料

Name	Quantity	Component
U1	1	Arduino Uno R3
S1 S2	2	Pushbutton

● 電路圖



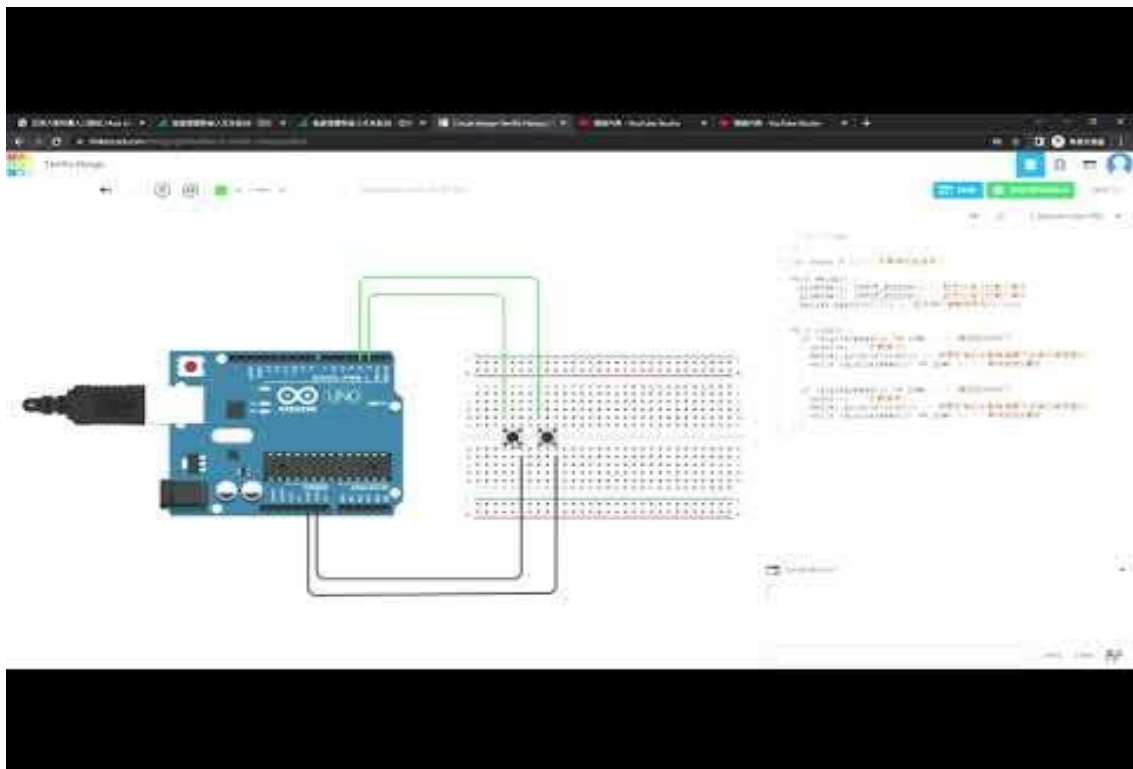
- 程式碼

```
Text [Download] [Save] [Font] 1 (Arduino Uno R3)
1 // C++ code
2 //
3
4 int count = 0; // 計數器初始值為0
5
6 void setup() {
7   pinMode(2, INPUT_PULLUP); // 設定D2為上拉輸入模式
8   pinMode(3, INPUT_PULLUP); // 設定D3為上拉輸入模式
9   Serial.begin(9600); // 設定串列傳輸速率為9600bps
10 }
11
12 void loop() {
13   if (digitalRead(2) == LOW) { // 當按鈕A被按下
14     count++; // 計數器加1
15     Serial.println(count); // 將更新後的計數器值顯示在串列埠視窗中
16     while (digitalRead(2) == LOW) {} // 等待按鈕A釋放
17   }
18
19   if (digitalRead(3) == LOW) { // 當按鈕B被按下
20     count--; // 計數器減1
21     Serial.println(count); // 將更新後的計數器值顯示在串列埠視窗中
22     while (digitalRead(3) == LOW) {} // 等待按鈕B釋放
23   }
24 }
25
```

Serial Monitor

[Send] [Clear] [Font]

- 錄製影片



- 反思(第一題)

透過這次的 **Arduino** 製作，我學到了很多關於電子裝置和程式設計的知識。使用 **Arduino** 可以幫助我們更好地了解各種電子元件的原理和應用，並學習如何以編程的方式將它們組合在一起以創建有用的裝置。

這次製作中，我學到了如何使用 **Arduino** 來控制8個 **LED** 燈，使它們能夠向左向右來回循環亮燈，每秒亮一顆。這個製作過程中，我需要精確控制每一個 **LED** 燈的亮滅時間和順序，並且需要熟悉 **Arduino** 的程式語言和函數。

在這個過程中，我也學到了一些有用的技巧和工具，例如如何使用面包板將電子元件連接在一起，以及如何使用 **Arduino IDE** 來編寫和上傳程式碼。這些技巧和工具將對我的未來學習和製作過程非常有幫助。

此外，透過這次製作，我也學到了一些重要的概念，例如控制流程、變量和陣列等。這些概念是編程中非常重要的基礎，對於進一步學習和應用編程技能非常有幫助。

總的來說，這次 **Arduino** 製作讓我更深入地了解了電子元件和程式設計，同時也為我提供了一個有趣且實用的工具，讓我能夠創建自己的電子裝置。我相信在未來，我會繼續學習和使用 **Arduino** 來實現更多有趣的製作。

- 反思(第二題)

這次使用 **Arduino** 製作計數器的過程，讓我更深入地了解了按鈕的原理和如何與 **Arduino** 搭配使用，同時也學到了如何使用串列通訊將計數值傳輸到電腦端，這是非常有用的技能。

在這個計數器的製作中，我使用了兩個按鈕分別接到 **D2**跟 **D3**接腳，並使用程式設計實現按下按鈕後計數器的功能。我學到了如何使用 **Arduino** 的 **digitalRead** 函數來讀取按鈕的狀態，並使用 **if** 語句實現按下按鈕後計數器的增加和減少。

此外，我還學到了如何使用 **Serial.print** 函數將計數器的值傳輸到電腦端，這樣就可以即時顯示計數器的值。這是一個非常實用的功能，可以讓我更方便地監控和控制計數器的運作。

在這個過程中，我也遇到了一些問題和挑戰，例如按鈕彈跳的問題和計數器溢出的問題。通過解決這些問題，我更深入地了解了 **Arduino** 的工作原理和如何優化程式碼。

總的來說，這次 **Arduino** 製作讓我更深入地了解了電子元件和程式設計，同時也為我提供了一個有趣且實用的工具，讓我能夠創建自己的電子裝置。我相信在未來，我會繼續學習和使用 **Arduino** 來實現更多有趣的製作。