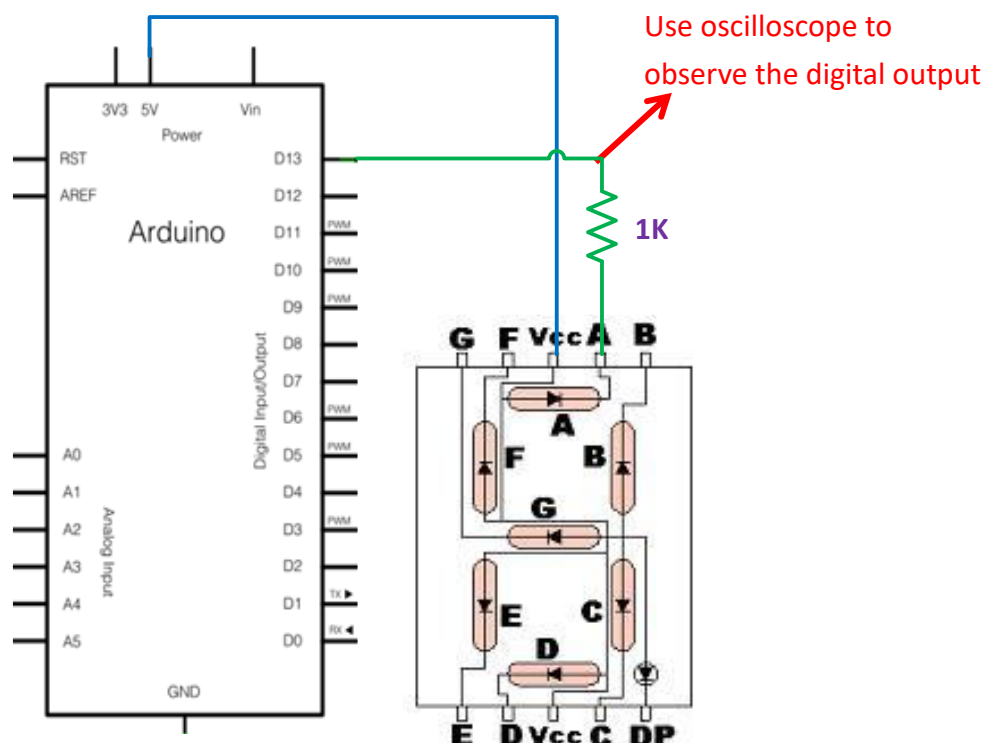


REPORT

Experiment 1: LED Blink.



NOTE: use DC coupling to observe the waveforms, and use cursor/measure to mark V_{high} and V_{low} .

Arduino 本實驗新函式介紹與程式分析:

函式介紹:

1. setup() / loop() :

當 Arduino 開始執行一個 sketch 的時候，會先呼叫 setup() 函式，進行包括變數的初始化、pinmode 的設定以及呼叫使用的函式庫。完成 setup() 函式後，會執行 loop() 函式，一般來說這個 loop() 可以看做 C 或 C++ 的 while(1)，在特殊情況外將持續執行直到 Arduino 板子重新執行 sketch 或是 reset。

2. pinMode(pin, mode):

設定 Arduino 板的 pin 腳狀態成 mode，mode 有三種選擇，INPUT、OUTPUT 和 INPUT_PULLUP，對應的狀態如下-

- INPUT 會將腳設為輸入端，此狀態下 pin 腳會有高輸入阻抗，相當於 100 兆歐姆的電阻，優點是只需微小電流即可改變 HIGH/LOW level，因此在讀取傳感器時非常有用，缺點是容易受到雜訊干擾，導致 "floating"。
- INPUT_PULLUP 相較於 INPUT 狀態，會將 pin 腳接到 Arduino 內部的上拉電阻，可以在不使用外部上拉或下拉電阻的狀態下，防止輸入 pin "floating"，導致不可預測的結果。

- **OUTPUT** 會將腳設為輸出端，也就是給予很低的阻抗，方便將訊號傳送出去，亦可以提供外接電路大量電流，除此之外也可以當作 **current source** 或是 **sink**，這使他非常適合用於驅動此實驗的 **LED**。由於低阻抗的原因，要避免將 **OUTPUT** 狀態下的 **pin** 腳短路或彼此串聯，否則會導致 **Arduino** 版燒毀。

3. **digitalWrite(pin, value):**

指定 **pin** 腳為 **HIGH**(通常為 5V)或 **LOW**(通常為 0V|地)。

如果將引腳配置為輸入(**INPUT**)，則 **digitalWrite()**將啟用 (**HIGH**) 或禁用 (**LOW**) 輸入引腳上的內部上拉電阻。

如果沒有將 **pinMode()**設置為 **OUTPUT**，直接把 **LED** 連接到引腳上，設定 **digitalWrite(HIGH)**時，**LED** 可能會顯得暗淡。因為沒有明確設置 **pinMode()**時，**digitalWrite()**將啟用內部上拉電阻，相當於一個大的電流限制電阻。

4. **delay(value):**

以 **value** 指定的時間（以毫秒為單位）暫停程式運行。（一秒中有 1000 毫秒。）

程式分析:

下三圖為 Lab1-1 的 **Arduino** 程式碼，有做更改的只有 **int led = 13** 與 **int led = 4** 以及 **delay()**的時間，來從不同的 **pin** 腳輸出以及輸出不同的頻率。

Lab1-1-1(pin13 OUTPUT& blink frequency 1Hz) sketch:

```
//lab1-1
int led = 13;

// the setup function runs once when you press reset or power the board
void setup() {
  // initialize digital pin LED_BUILTIN as an output.

  //pinMode(LED_BUILTIN, OUTPUT);

  pinMode(led, OUTPUT);
  //Serial.begin(9600);
}

// the loop function runs over and over again forever
void loop() {
  //digitalWrite(LED_BUILTIN, LOW); // turn the LED on (LOW is the voltage level)
  digitalWrite(led, LOW);
  delay(1000); // wait for a second
  //digitalWrite(LED_BUILTIN, HIGH); // turn the LED off by making the pin voltage HIGH
  digitalWrite(led, HIGH);
  delay(1000); // wait for a second
}
```

Lab1-1-2(pin4 OUTPUT& blink frequency 0.5Hz) sketch:

```
//lab1-2
int led = 4;

// the setup function runs once when you press reset or power the board
void setup() {
  // initialize digital pin LED_BUILTIN as an output.

  //pinMode(LED_BUILTIN, OUTPUT);

  pinMode(led, OUTPUT);
  //Serial.begin(9600);
}

// the loop function runs over and over again forever
void loop() {
  //digitalWrite(LED_BUILTIN, LOW); // turn the LED on (LOW is the voltage level)
  digitalWrite(led, LOW);
  delay(2000); // wait for 2 seconds
  //digitalWrite(LED_BUILTIN, HIGH); // turn the LED off by making the pin voltage HIGH
  digitalWrite(led, HIGH);
  delay(2000); // wait for 2 seconds
}
```

Lab1-1-3(pin4 OUTPUT& blink frequency 2Hz) sketch:

```
//lab1-3
int led = 4;

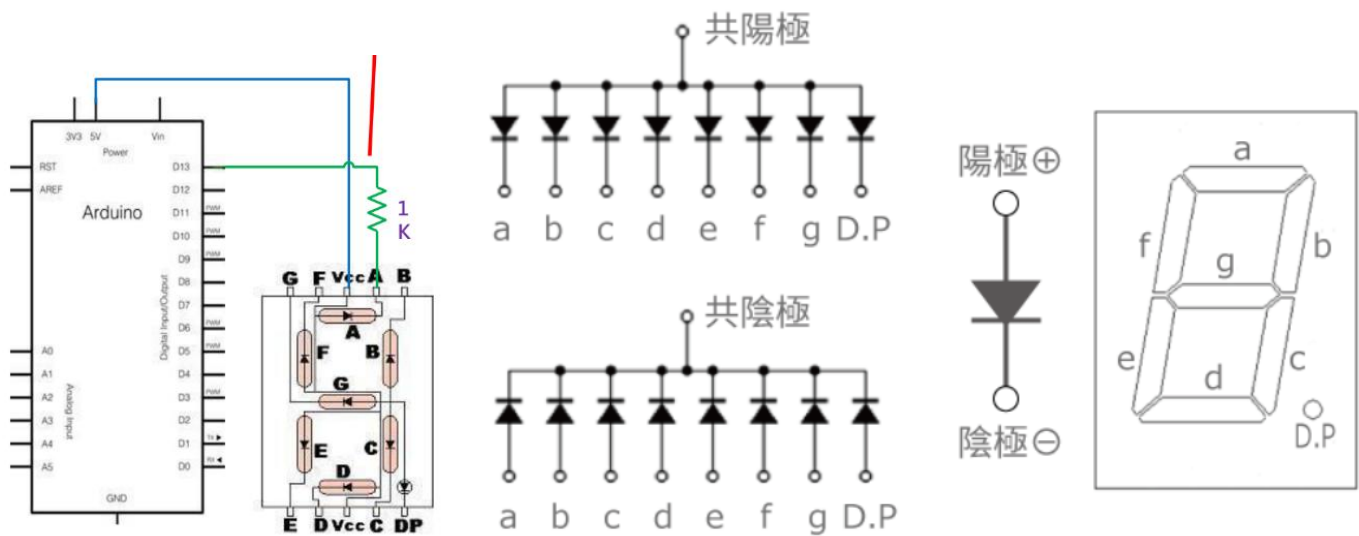
// the setup function runs once when you press reset or power the board
void setup() {
  // initialize digital pin LED_BUILTIN as an output.

  //pinMode(LED_BUILTIN, OUTPUT);

  pinMode(led, OUTPUT);
  //Serial.begin(9600);
}

// the loop function runs over and over again forever
void loop() {
  //digitalWrite(LED_BUILTIN, LOW); // turn the LED on (LOW is the voltage level)
  digitalWrite(led, LOW);
  delay(500); // wait for 0.5 second
  //digitalWrite(LED_BUILTIN, HIGH); // turn the LED off by making the pin voltage HIGH
  digitalWrite(led, HIGH);
  delay(500); // wait for 0.5 second
}
```

電路分析:

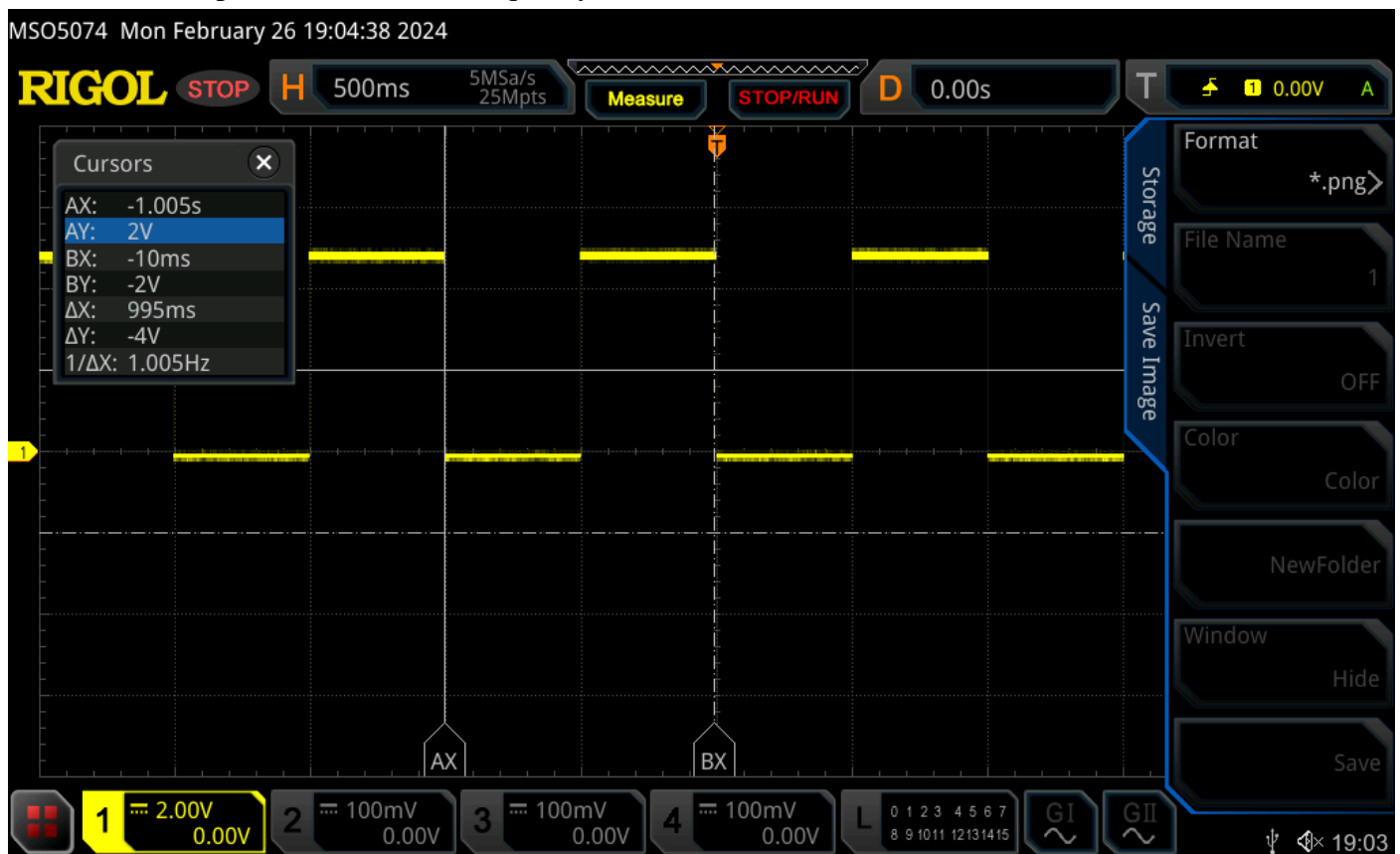


由實驗 ppt 提供的圖片(左上圖)，可以看出實驗中使用的是如右上圖展示的共陽極結構的七段顯示器，由上學期對 diode 的學習與操作，可以知道當 Vcc 端的電壓相對 A 點(實驗 1-1-1 的 pin13/實驗 1-1-2、1-1-3 的 pin4)高出 cut-in voltage，發光二極體就會發亮。

對照 Arduino 的程式碼，當 pin 腳被設置為 HIGH 時，LED 會熄滅，而 LOW 時則會發亮。

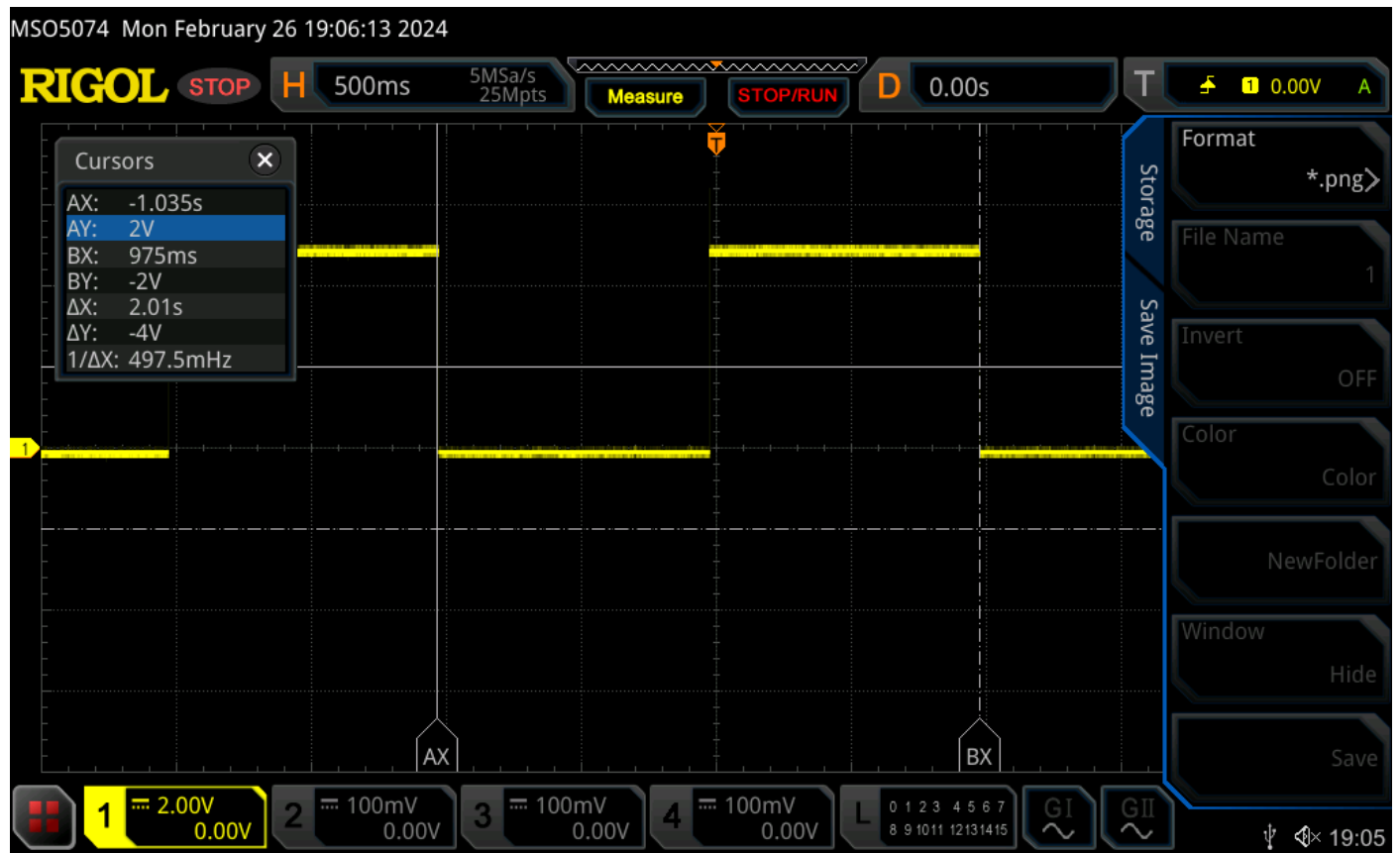
1.

Arduino D13 output waveform with frequency 1Hz.



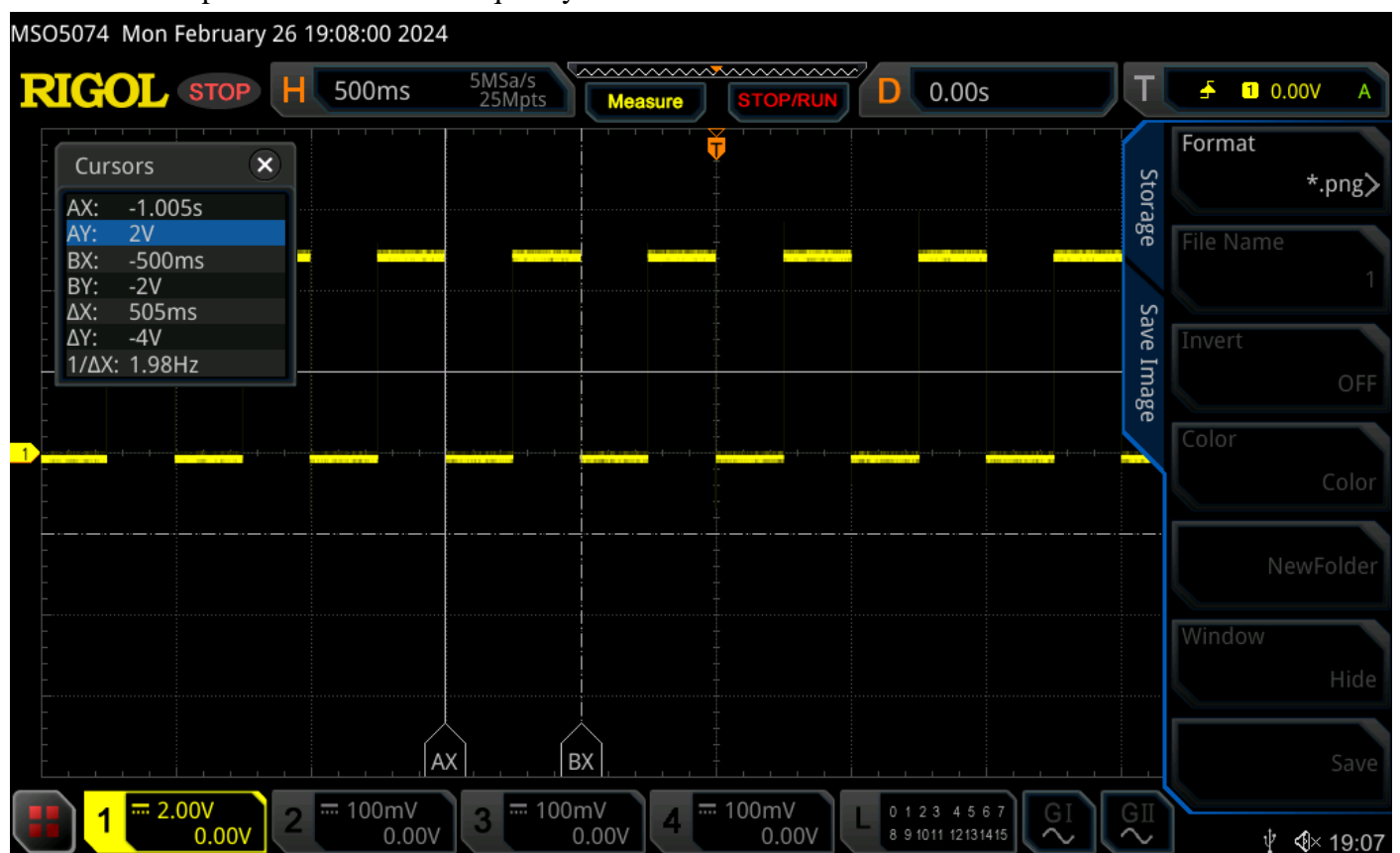
2.

Arduino D4 ouput waveform with frequency 0.5Hz.



3.

Arduino D4 ouput waveform with frequency 2Hz.



Question:

What output level (high/low) will make LED turn light or dark? Why?

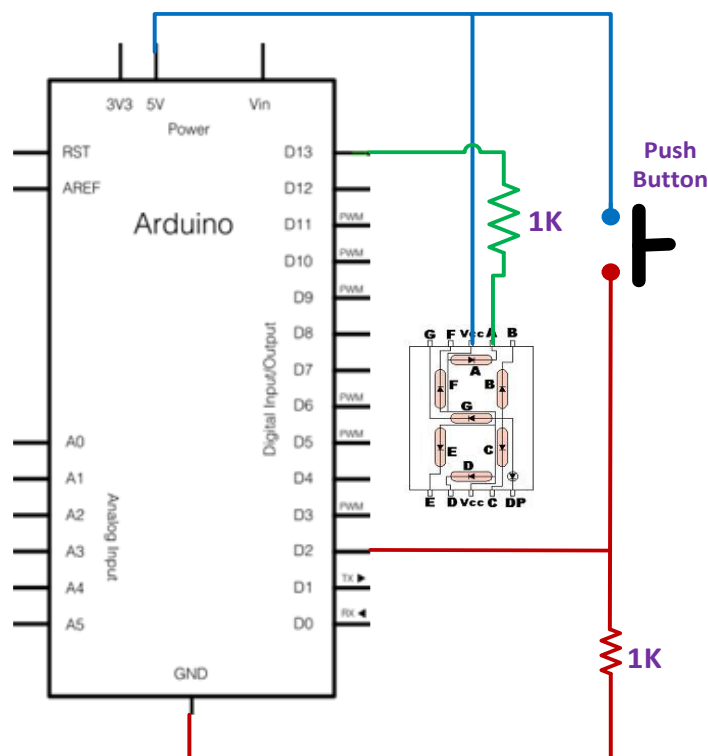
當 Output level 為 high 時 LED 不亮，為 low 時 LED 發亮。

由於七段顯示器為共陽極，因此當 pin 腳為高電位 HIGH 時，LED 兩端電位差小於 cut-in voltage 所以部會發光，反之 pin 腳為低電位 LOW 時，則會發光。(更詳細分析可見上方實驗 1-1 的電路分析)

若沒有在 setup()中設置 pinMode()，會發生甚麼事?

由於 Arduino 的 pin 腳預設為 INPUT，而上方(Arduino 本實驗新函式介紹與程式分析)有提到，INPUT 模式下的 pin 腳會有高的輸入阻抗(也可看作輸出阻抗，若訊號由內部發出)，造成輸出訊號很微弱，如此一來會導致 LED 不會發亮，或是亮度低。

Experiment 2: LED Blink with push button.



Arduino 本實驗新函式介紹與程式分析:

函式介紹:

1. Serial.begin(value) / Serial.println() / Serial.read():

當需要讓 Arduino 與連接的電腦溝通(傳送資料)時，就會用到 Serial 系列函式。

Serial.begin(value)開啟 Serial Port，value 為自定義的通訊速率 (baud rate)。

Serial.println()傳送資料到連接電腦，傳送的內容可以在序列埠監控視窗看到，此功能可以讓我們視覺化得到所需資訊，也可以方便 debug。

Serial.read()則可以接收來自連接電腦所給予的資料。

2. digitalRead(pin):

讀取指定數位腳位(pin)的值，回傳 int 值 1 高電位 (HIGH) 或 0 低電位 (LOW)。

程式分析:

下兩圖為 Lab1-2 的 Arduino 程式碼，值得注意的是兩者在 loop()函式中的條件判斷分別是 if(buttonState == LOW)與 if(buttonState == HIGH)，這是由於 pull-low 時，在按鈕沒有按下的時候 pin2，也就是 pushButton 的電位為地(0V)，所以 digitalRead 回傳值給予 buttonState 為 LOW，因此 if 判斷式須為 if(buttonState == LOW)，才會執行，反之 pull-high 時，按鈕沒按下時 pin2 的電位為 Vcc(5V)，buttonState 為 HIGH，因此 if 判斷式須為 if(buttonState == HIGH)，才會執行。

Lab1-2-1(pin 2 pull-low) sketch:

```
//lab1-2-1 pull low
int pushButton = 2;
int led = 13;

void setup() {
  // initialize serial communication at 9600 bits per second:
  Serial.begin(9600);
  // make the pushbutton's pin an input and led's pin an output:
  pinMode(pushButton, INPUT);
  pinMode(led, OUTPUT);
}

void loop() {
  // read the input pin:
  int buttonState = digitalRead(pushButton);

  // make LED blink and output buttonState.
  if (buttonState == LOW) {
    digitalWrite(led, LOW); // turn the LED on
    delay(1000); // wait for a second
    digitalWrite(led, HIGH); // turn the LED off by making the voltage LOW
    delay(1000);
    Serial.println(buttonState);
  } else { // turn LED on and output buttonState.
    digitalWrite(led, LOW);
    Serial.println(buttonState);
  }
}
```


Lab1-2-2(pin 2 pull-high) sketch:

```
//lab1-2-2 pull high
int pushButton = 2;
int led = 13;

void setup() {
  // initialize serial communication at 9600 bits per second:
  Serial.begin(9600);
  // make the pushbutton's pin an input and led's pin an output:
  pinMode(pushButton, INPUT);
  pinMode(led, OUTPUT);
}

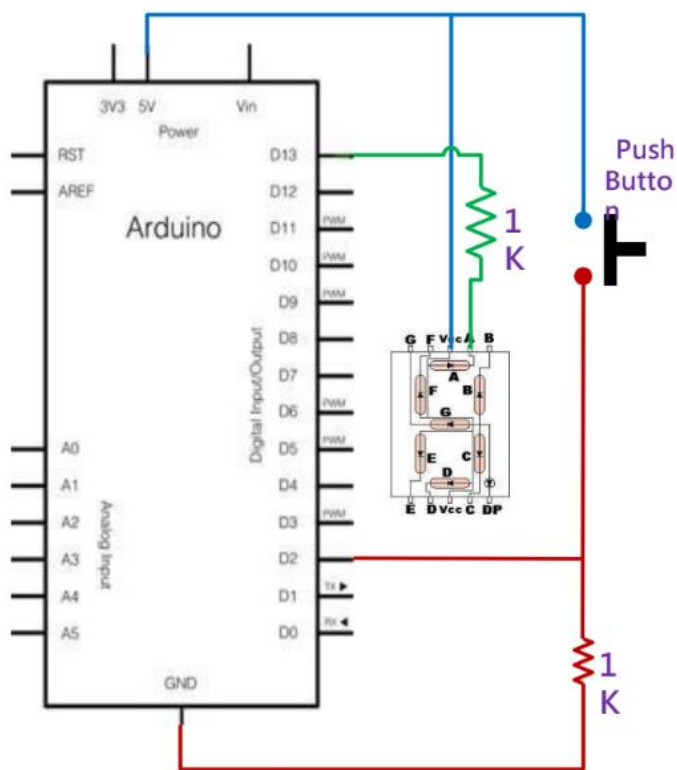
void loop() {
  // read the input pin:
  int buttonState = digitalRead(pushButton);

  // make LED blink and output buttonState.
  if (buttonState == HIGH) {
    digitalWrite(led, LOW); // turn the LED on
    delay(1000); // wait for a second
    digitalWrite(led, HIGH); // turn the LED off by making the voltage LOW
    delay(1000);
    Serial.println(buttonState);
  } else { // turn LED on and output buttonState.
    digitalWrite(led, LOW);
    Serial.println(buttonState);
  }
}
```

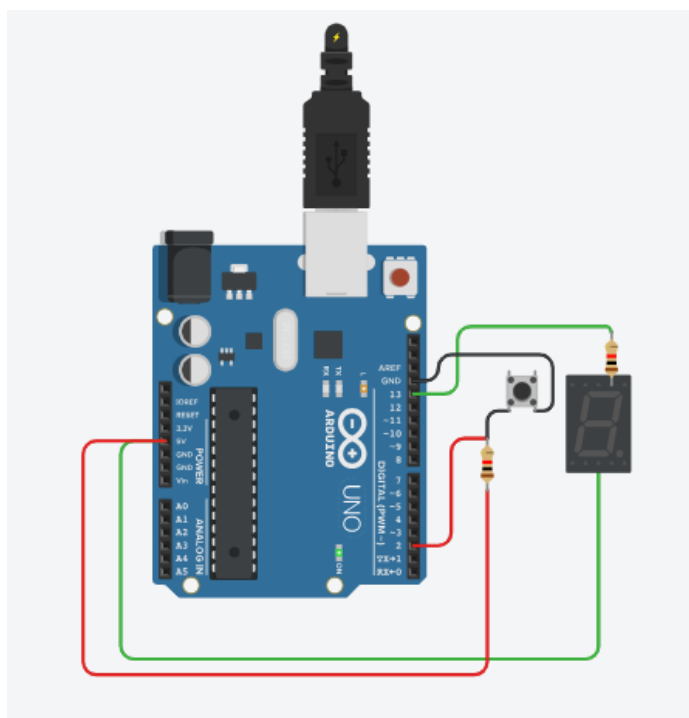
電路分析:

pull-high 與 pull-low 介紹:

上拉 (Pull-Up) 或下拉 (Pull-Down) 電阻 (兩者統稱為“拉電阻”)，最基本的作用是将狀態不確定的信號線通過一個電阻將其箝位至高電位 (上拉) 或低電位 (下拉)，以防止信號線因懸空而出現不確定的狀態，繼而導致系統出現不期望的狀態。



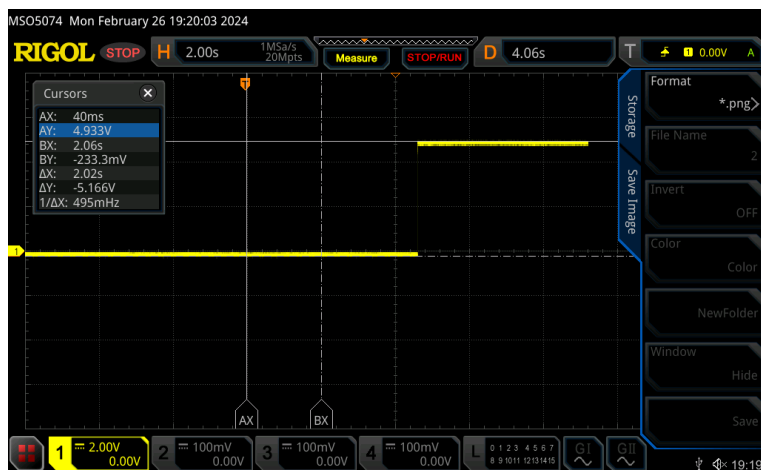
(pull-low)



(pull-high)

實驗 1-2-1 的電路圖如左上，在按鈕沒有按下時，D2 的電位是 0V(接地)，而當按鈕按下後，D2 的電位理想上則會變為 5V，因此由定義可以得到這是一個 pull-low 的電路型態，而實驗 1-2-2 的電路圖如右上角，在按鈕放開時，D2 的電位接近 5V(可由內電阻與 1k ohm 的電阻分壓得到)，而當按鈕按下時，則會因為與地短路而電位變成 0V。而 Arduino 板會根據上面分析的 sketch 程式碼，在 D13 腳位做出對應的輸出，使得 7 段顯示器發亮。

下圖為示波器量測 pull-low 時的 D2 腳位電壓波型，左側低電位為按鈕放開時，右側高電位為按鈕按下時，cursor 測量到 4.933V 左右。



1.

Is this circuit a pull-high or pull-low type? ___pull-low___.

2.

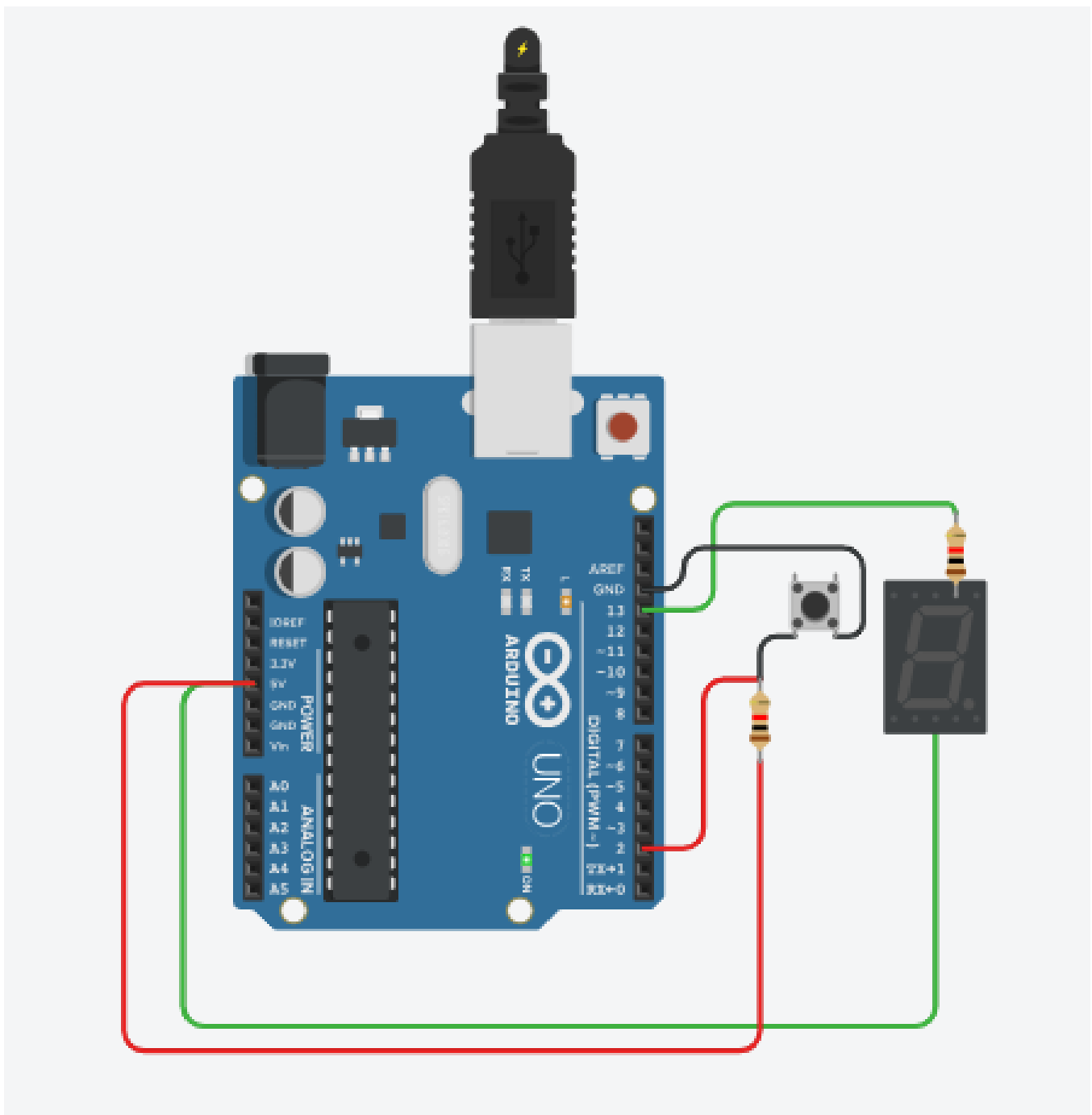
Measure pin D2 voltage using voltage meter:

Behavior	DC Voltage (V)
Pushbutton pressed	4.94
Pushbutton released	0.00

3.

Change your circuit to another type and modify your circuit and sketch to achieve the same requirement.

Modified circuit (pull-high):



Modified sketch:

```
//lab1-2-2 pull high
int pushButton = 2;
int led = 13;

void setup() {
  // initialize serial communication at 9600 bits per second:
  Serial.begin(9600);
  // make the pushbutton's pin an input and led's pin an output:
  pinMode(pushButton, INPUT);
  pinMode(led, OUTPUT);
}

void loop() {
  // read the input pin:
  int buttonState = digitalRead(pushButton);

  // make LED blink and output buttonState.
  if (buttonState == HIGH) {
    digitalWrite(led, LOW); // turn the LED on
    delay(1000); // wait for a second
    digitalWrite(led, HIGH); // turn the LED off by making the voltage LOW
    delay(1000);
    Serial.println(buttonState);
  } else { // turn LED on and output buttonState.
    digitalWrite(led, LOW);
    Serial.println(buttonState);
  }
}
```

4.

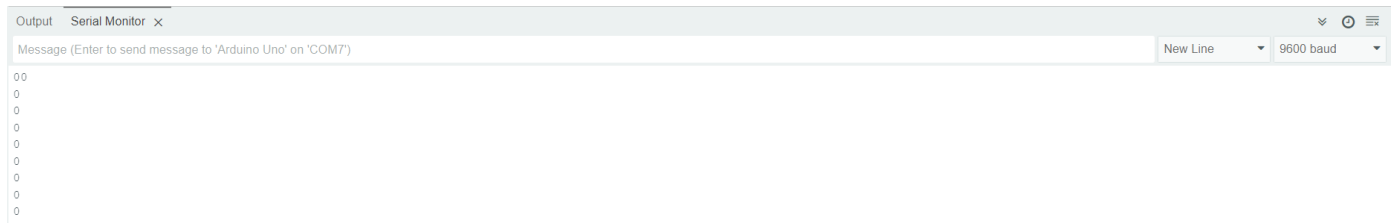
Measure pin D2 voltage using voltage meter after step 3:

Behavior	DC Voltage (V)
Pushbutton pressed	0.00
Pushbutton released	4.94

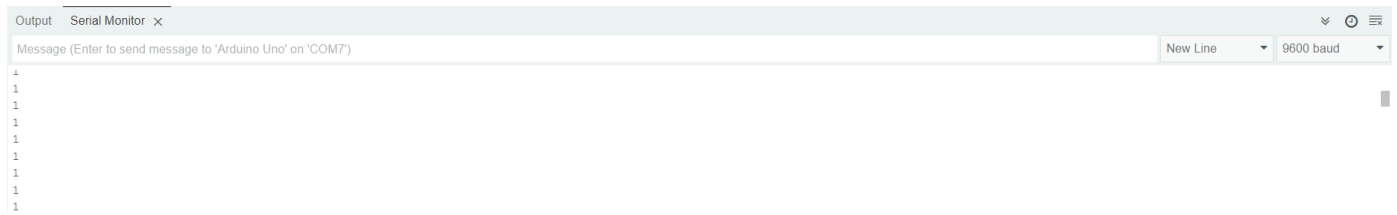
5.

Attach your serial monitor screen shot

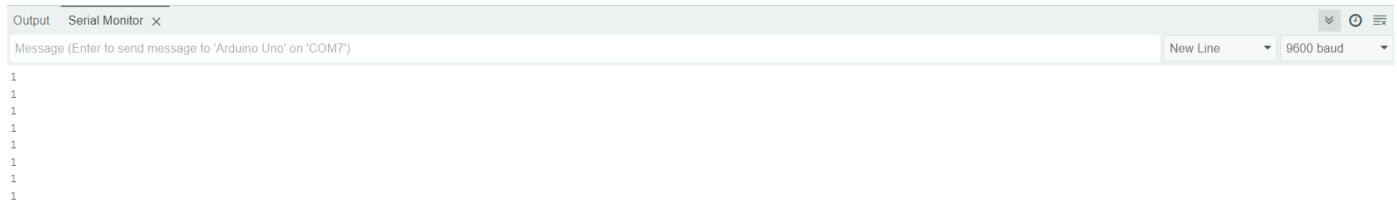
Pull-low&bottom released (輸出 0，低電位):



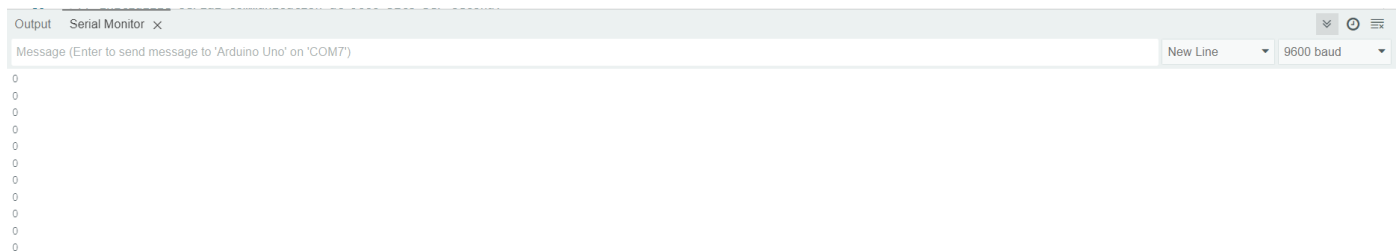
Pull-low&bottom pressed (輸出 1，高電位):



Pull-high&bottom released (輸出 1，高電位):



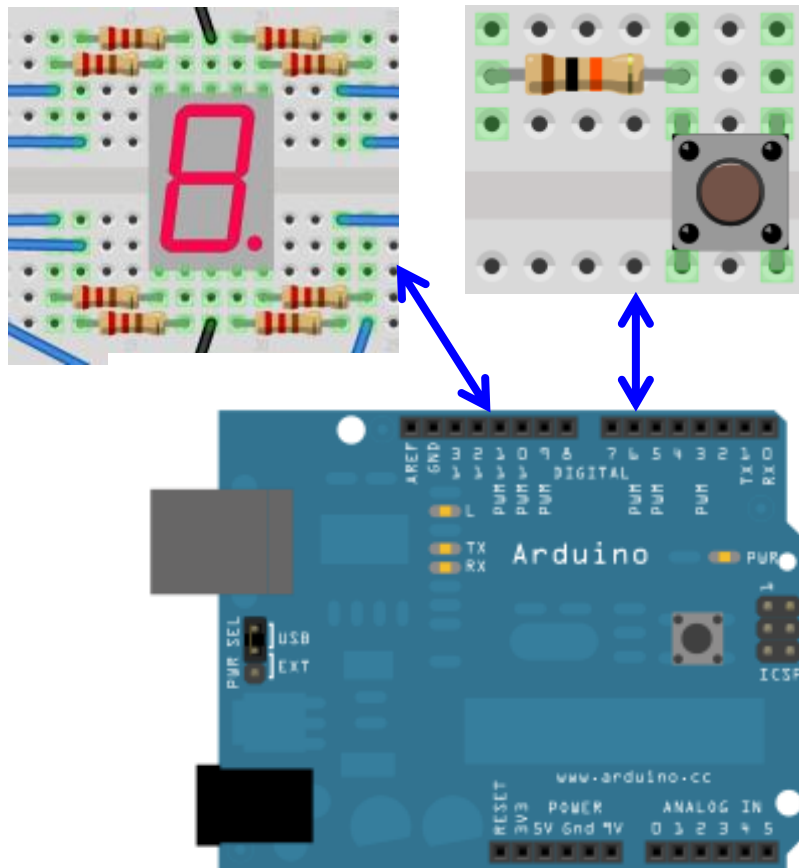
Pull-high&bottom pressed (輸出 0，低電位):



Question:

若 pull-low 與 pull-high 電路的 gnd 並非連接在 Arduion 板上的 gnd 接孔，而是其他的 gnd(如 function generator 的地線)會發生甚麼事？

由電子實驗(一)得知，地的概念並非為絕對的 0V，而是與所需輸出點電位差為要求位能的一個相對點，因此若將 gnd 連接至外部的地線，則有可能導致 D2 腳位實際的電位出現錯誤，導致 LED 的發光與否與想要的結果不同。

Experiment 3: LED Marquee.**Arduino 本實驗新函式介紹與程式分析:****函式介紹:****1. showSevenSeg(A, B, C, D, E, F, G, P):**

此函式是我們自己定義的一個 function，他會將吃到的 A~G 和 P 的值(byte)分別寫入我們在 setup()之前定義的 pinA 到 pinG 和 pinP 腳位的狀態(1 HIGH 或 0 LOW)。

程式分析:

首先在 setup()函式內定義了 pinA~pinG 和 pinP 腳位為 OUTPUT 腳位，pushButton 為 INPUT 腳位，接著在 loop()函式中使用 digitalRead 讀取 pushButton 的狀態給 buttonState 用來進行下方 if 和 else 的判斷，當 buttonState 為低電位時(按鈕放開)，則會依序亮起編號為 A-B-C-D-E-F 的發光二極體，而當 buttonState 為高電位時(按鈕按下)，則會亮起 7 段顯示器的所有發光二極體。

電路分析:

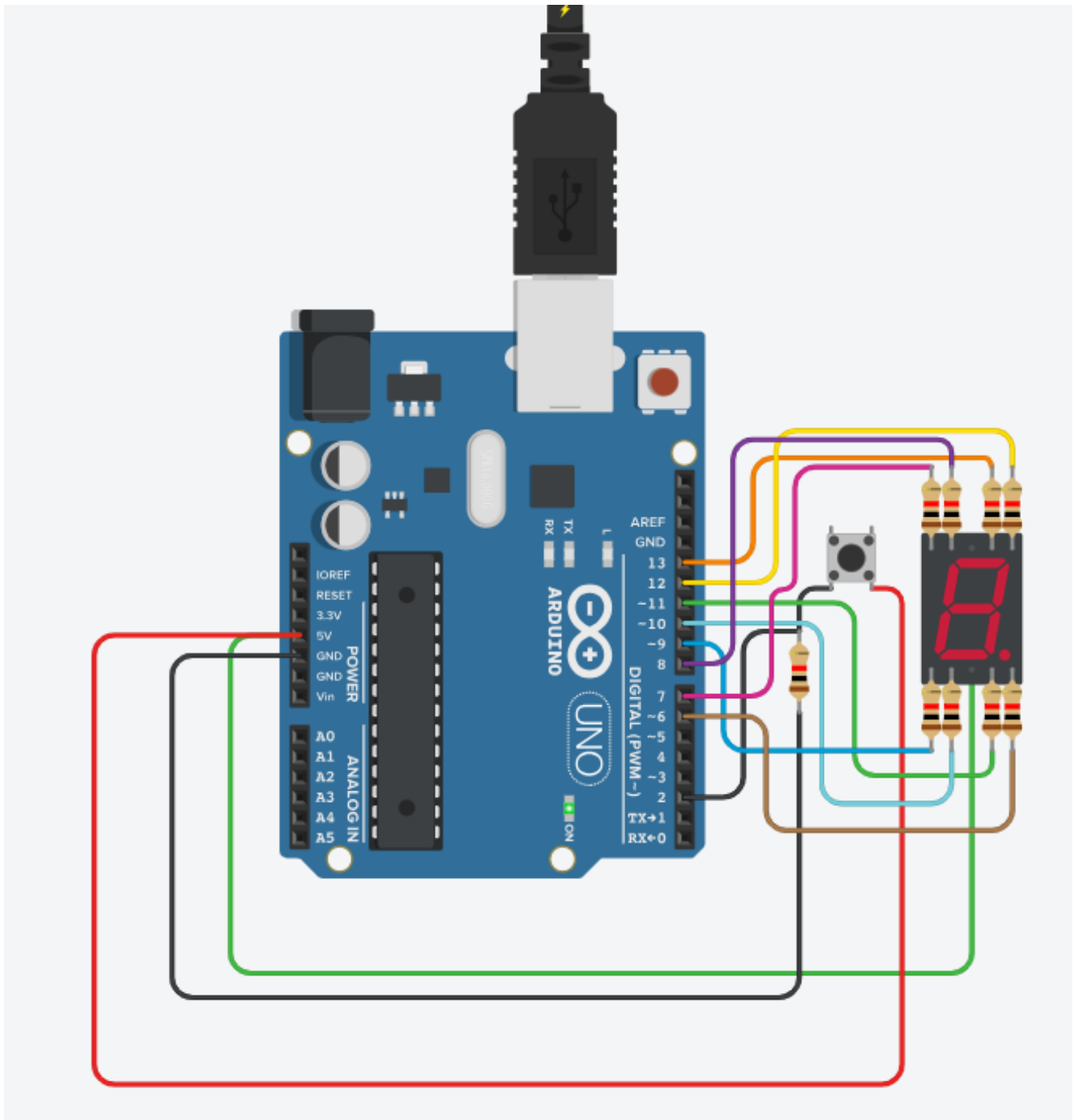
此實驗沿用 1-2-1 的 pull-low 按鈕電路去做 if-else 的條件判斷，所以腳位 D2 的狀態與實驗 1-2-1 相同，按鈕放開時為低電位，按下後則為高電位，而 7 段顯示器的各個接點，也透過一個限流電阻連接到 Arduino 的 pin 腳上，從而達到讓 Arduino 程式操控 7 段顯示器的發光效果。

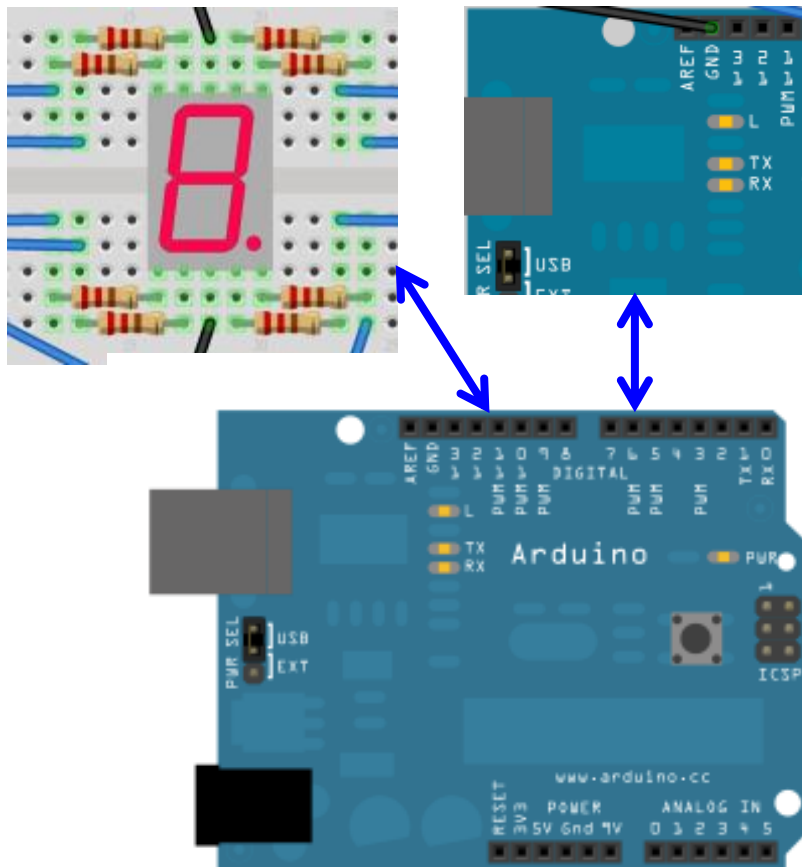
Attach your sketch and result pictures:

```

1
2 //lab1-3
3 byte pinA=13, pinB=12, pinC=11, pinD=10, pinE=9, pinF=8, pinG=7, pinP=6;
4 byte pushButton = 2;
5
6 void setup() {
7   pinMode(pinA, OUTPUT); pinMode(pinB, OUTPUT); pinMode(pinC, OUTPUT);
8   pinMode(pinD, OUTPUT); pinMode(pinE, OUTPUT); pinMode(pinF, OUTPUT);
9   pinMode(pinG, OUTPUT); pinMode(pinP, OUTPUT);
10  pinMode(pushButton, INPUT);
11 }
12
13 // show the specified LEDs of 7-segment
14 void showSevenSeg(byte A, byte B, byte C, byte D, byte E, byte F, byte G, byte P) {
15   digitalWrite(pinA, A); digitalWrite(pinB, B); digitalWrite(pinC, C);
16   digitalWrite(pinD, D); digitalWrite(pinE, E); digitalWrite(pinF, F);
17   digitalWrite(pinG, G); digitalWrite(pinP, P);
18 }
19
20 void loop() {
21   // read the input pin:
22   int buttonState = digitalRead(pushButton);
23   if (buttonState == LOW) {
24     showSevenSeg(0,1,1,1,1,1,1,1); // turn LED A of 7-segment on.
25     delay(500);
26
27     showSevenSeg(1,0,1,1,1,1,1,1); // turn LED B of 7-segment on.
28     delay(500);
29
30     showSevenSeg(1,1,0,1,1,1,1,1); // turn LED C of 7-segment on.
31     delay(500);
32
33     showSevenSeg(1,1,1,0,1,1,1,1); // turn LED D of 7-segment on.
34     delay(500);
35
36     showSevenSeg(1,1,1,1,0,1,1,1); // turn LED E of 7-segment on.
37     delay(500);
38
39     showSevenSeg(1,1,1,1,1,0,1,1); // turn LED F of 7-segment on.
40     delay(500);
41   }
42   else {
43     showSevenSeg(0,0,0,0,0,0,0,0); // turn all LEDs of 7-segment on.
44   }
45 }
46

```



Experiment 4: student ID Marquee.**Arduino 本實驗新函式介紹與程式分析:****函式介紹:**

無

程式分析:

大致與實驗 1-3 相同，只有在 loop()內的 if-else 判斷的 else 部分，從 7 段顯示器全部發亮變成依照自己的學號發光，我的學號對應的依序為 X-1-X-1-X-1-X-5-X-1-X-1-X-2-X-3-X-9

電路分析:

與實驗 1-3 相同。

Attach your sketch and result pictures:

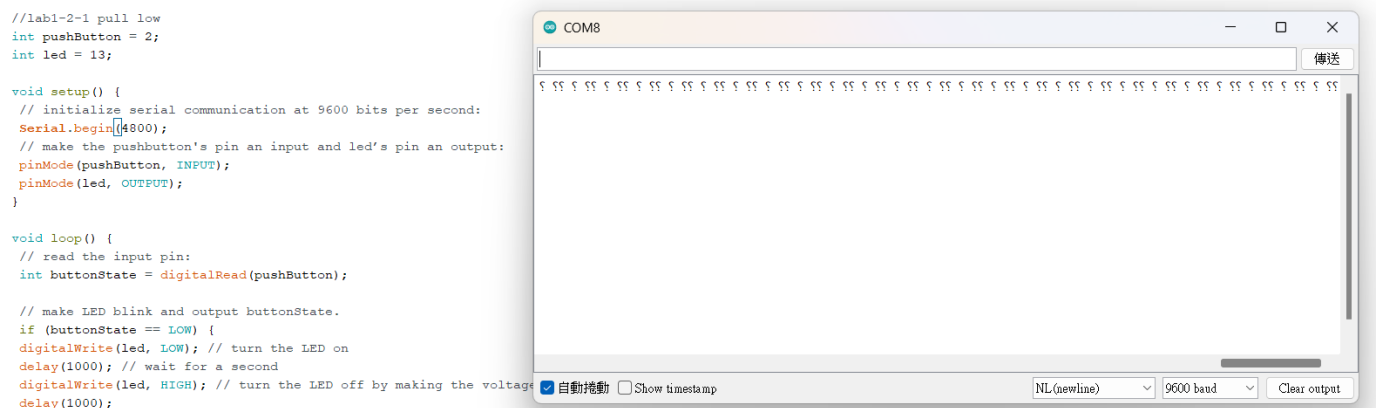
```

1
2 //lab1-3
3 byte pinA=13, pinB=12, pinC=11, pinD=10, pinE=9, pinF=8, pinG=7, pinP=6;
4 byte pushButton = 2;
5
6 void setup() {
7   pinMode(pinA, OUTPUT); pinMode(pinB, OUTPUT); pinMode(pinC, OUTPUT);
8   pinMode(pinD, OUTPUT); pinMode(pinE, OUTPUT); pinMode(pinF, OUTPUT);
9   pinMode(pinG, OUTPUT); pinMode(pinP, OUTPUT);
10  pinMode(pushButton, INPUT);
11 }
12
13 // show the specified LEDs of 7-segment
14 void showSevenSeg(byte A, byte B, byte C, byte D, byte E, byte F, byte G, byte P) {
15   digitalWrite(pinA, A); digitalWrite(pinB, B); digitalWrite(pinC, C);
16   digitalWrite(pinD, D); digitalWrite(pinE, E); digitalWrite(pinF, F);
17   digitalWrite(pinG, G); digitalWrite(pinP, P);
18 }
19
20 void loop() {
21   // read the input pin:
22   int buttonState = digitalRead(pushButton);
23   if (buttonState == LOW) {
24     showSevenSeg(0,1,1,1,1,1,1); // turn LED A of 7-segment on.
25     delay(500);
26
27     showSevenSeg(1,0,1,1,1,1,1); // turn LED B of 7-segment on.
28     delay(500);
29
30     showSevenSeg(1,1,0,1,1,1,1); // turn LED C of 7-segment on.
31     delay(500);
32
33     showSevenSeg(1,1,1,0,1,1,1); // turn LED D of 7-segment on.
34     delay(500);
35
36     showSevenSeg(1,1,1,1,0,1,1); // turn LED E of 7-segment on.
37     delay(500);
38
39     showSevenSeg(1,1,1,1,1,0,1); // turn LED F of 7-segment on.
40     delay(500);
41   }
42   else {
43     showSevenSeg(0,0,0,0,0,0,0); // turn all LEDs of 7-segment on.
44   }
45 }

```

Overall Question:

如果 `Serial.begin()` 括弧內的 **baud rate** 數值與序列埠監控視窗的 **baud rate** 不同會顯示什麼資訊？



這裡我選用 `Serial.begin(4800)` 與序列埠監控視窗 `baud rate=9600` 做觀察，發現顯示的所有數值都是亂碼，因此只要 `Serial.begin()` 括弧內的 **baud rate** 數值與序列埠監控視窗的 **baud rate** 不同，監控視窗就只會出現亂碼，而不會在公倍數的地方有正常的顯示資訊。

實驗心得:

這次的實驗算是在我大學裡第一次接觸到 **Arduino** 的應用，而這次的實驗內容也算幫自己找回國中和高中在操作 **Arduino** 板的記憶，起到複習的作用，大部分的 **code** 與電路都算是非常的簡單，很適合第一次接觸 **Arduino** 的人，也適合作為銜接較有難度實驗的一個開頭實驗，實驗中唯一有遇到一個比較麻煩的點就是杜邦線的母頭要自己接一個母轉公的針腳，我不小心被刺到手指，有點痛。

整體而言，此次實驗我認為是認識與摸索 **Arduino** 非常好的開頭，也期待後面幾周的 **Arduino** 課程與實驗。

實驗結論:

實驗 1-1 重點就是要記得在 **Arduino** 的腳位與 7 段顯示器之間串接一個電阻，由於 **OUTPUT mode** 的腳位本身是沒有高輸出阻抗的，因此若忘記串接一個限流電阻，有可能會導致 7 段顯示器的發光二極體燒壞。

實驗 1-2 重點為 **pull-high** 與 **pull-low** 的按鈕電路了解與設計，**pull-high** 與 **pull-low** 其實在很多的 **IC** 與電路中都有被使用，因此了解上拉與下拉電阻的運作其實是學習電路的必備知識。

實驗 1-3 重點有兩個，第一個是要使用第一個自定義的 **function showSevenSeg()**，要確認 7 段顯示器對應的 **pin** 腳與程式撰寫的是否正確，否則 7 段顯示器不會按照預期的情況發光，第二個重點是要確保實際電路有連接正確，因為 7 段顯示器的 **a~f** 點位和 **dp** 其實不是順時鐘轉一圈，而是有跳來跳去的，因此連接線路時要注意不要接錯。

實驗 1-4 由於大部分程式與全部電路都沿用實驗 1-3，因此重點部分在於顯示自己學號的程式碼，正常來說單獨使用 **showSevenSeg()** **function** 就好，但是助教有提到我們可以另外寫一個 **function**，吃

一個數值，並用該數值進行 `switch` 的比較來進行 `showSevenSeg` 的數字輸出，這樣可以精簡程式碼，也方便以後顯示數字的程式撰寫數值，以下是我後來完成的 `function`:

```
void showSevenSegnumber(int num) //根據num顯示對應數字
{
    switch(num)
    {
        case 0:
            showSevenSeg(0,0,0,0,0,0,1,1);
            delay(500);
            break;
        case 1:
            showSevenSeg(1,0,0,1,1,1,1,1);
            delay(500);
            break;
        case 2:
            showSevenSeg(0,0,1,0,0,1,0,1);
            delay(500);
            break;
        case 3:
            showSevenSeg(0,0,0,0,1,1,0,1);
            delay(500);
            break;
        case 4:
            showSevenSeg(1,0,0,1,1,0,0,1);
            delay(500);
            break;
        case 5:
            showSevenSeg(0,1,0,0,1,0,0,1);
            delay(500);
            break;
        case 6:
            showSevenSeg(0,1,0,0,0,0,0,1);
            delay(500);
            break;
        case 7:
            showSevenSeg(0,0,0,1,1,1,1,1);
            delay(500);
            break;
        case 8:
            showSevenSeg(0,0,0,0,0,0,0,1);
            delay(500);
            break;
        case 9:
            showSevenSeg(0,0,0,0,1,0,0,1);
            delay(500);
            break;
        case 10:
            showSevenSeg(1,1,1,1,1,1,1,1);
            delay(100);
            break;
    }
}
```

Reference:

Arduino. (2024) Arduino. Retrieved from: <https://www.arduino.cc/> (2024/02/27)

台部落。(2022/02/23) 【小常識】從按鈕開關看上拉 pull-up 電阻、下拉電阻。檢自:
<https://www.twblogs.net/a/5e51d5d4bd9eee21168065a8> (2024/02/27)

台部落。(2019/01/03) 上拉 (Pull Up) 或下拉 (Pull Down) 電阻詳解。檢自:
<https://www.twblogs.net/a/5c2de8acbd9eee35b21c4548> (2024/02/27)

ROHM semiconductor. (未知) 7 段 LED。檢自:
https://www.rohm.com.tw/electronics-basics/led/led_what7 (2024/02/27)