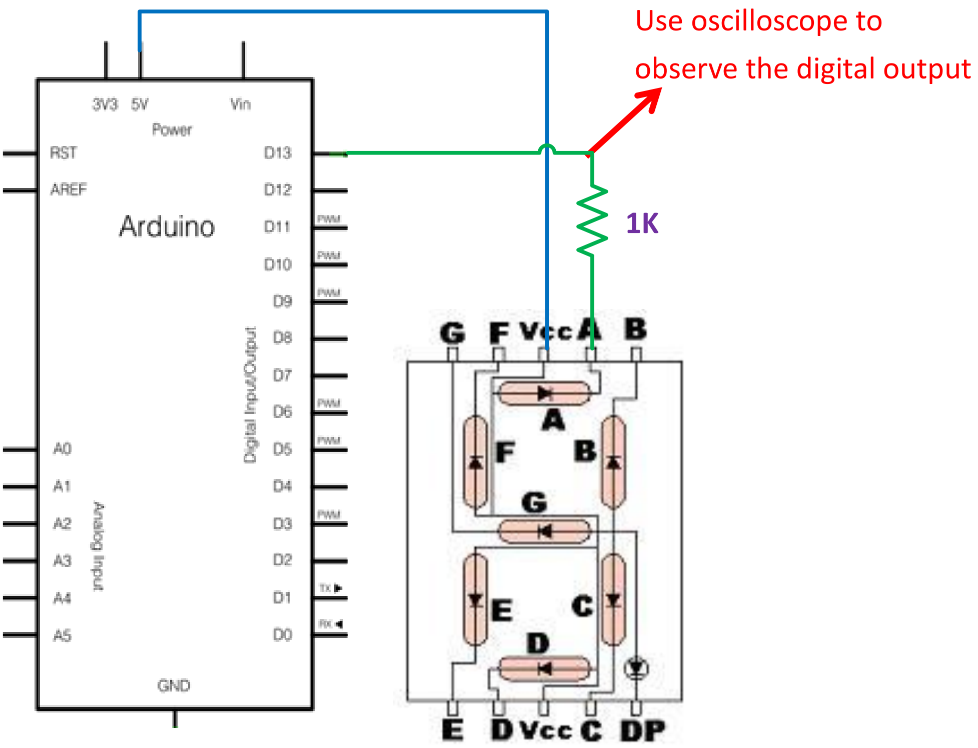
REPORT

Experiment 1: LED Blink.

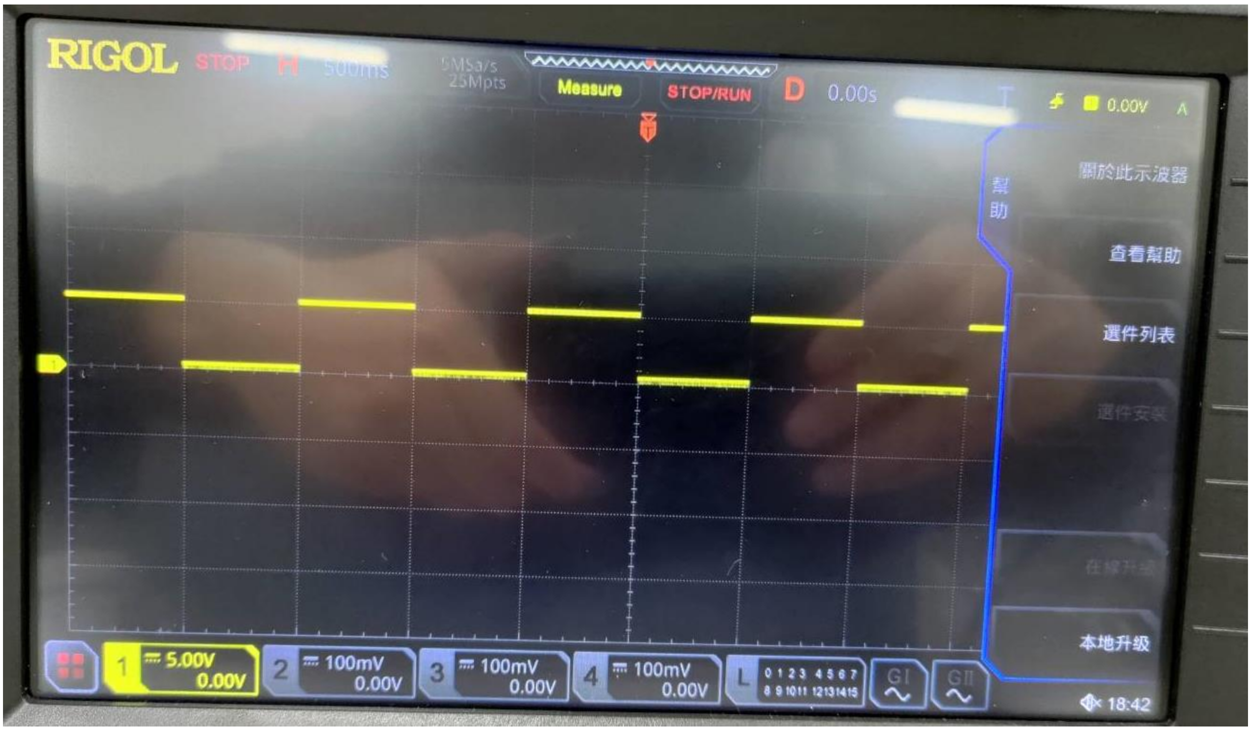


Use oscilloscope to observe the digital output

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

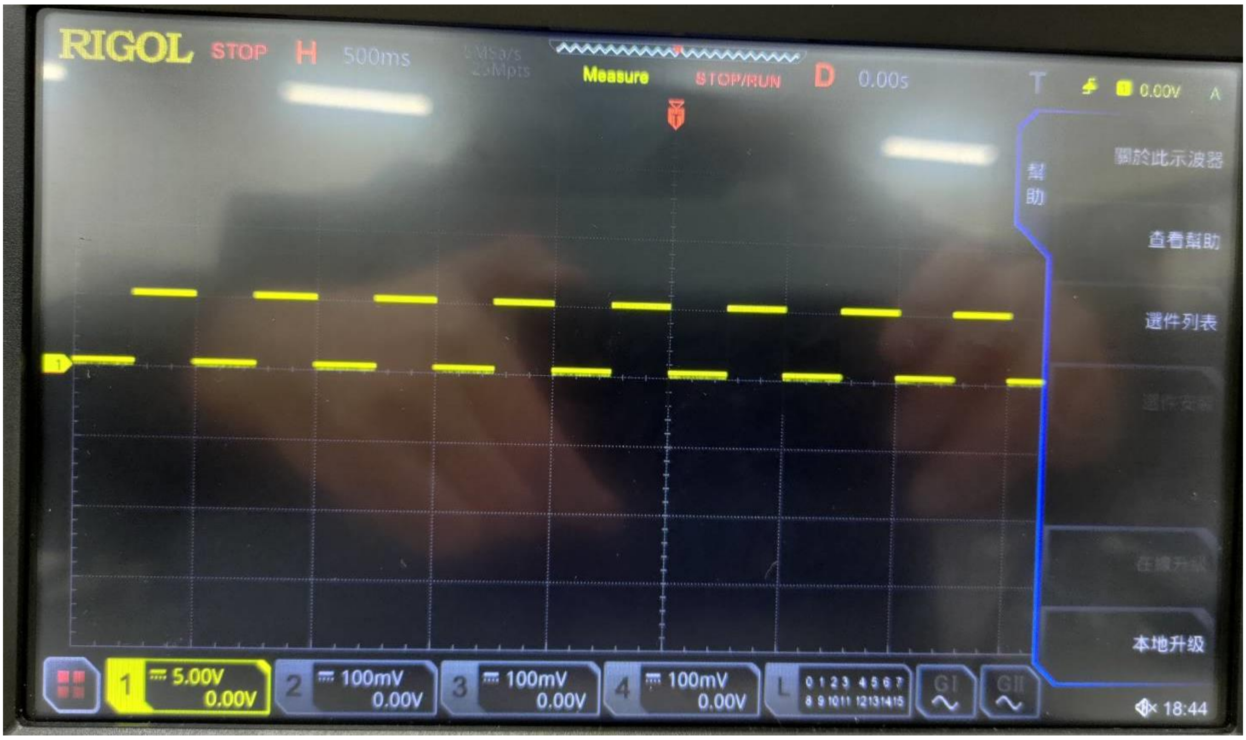
1.

Arduino D13 ouput waveform with frequency 1Hz.



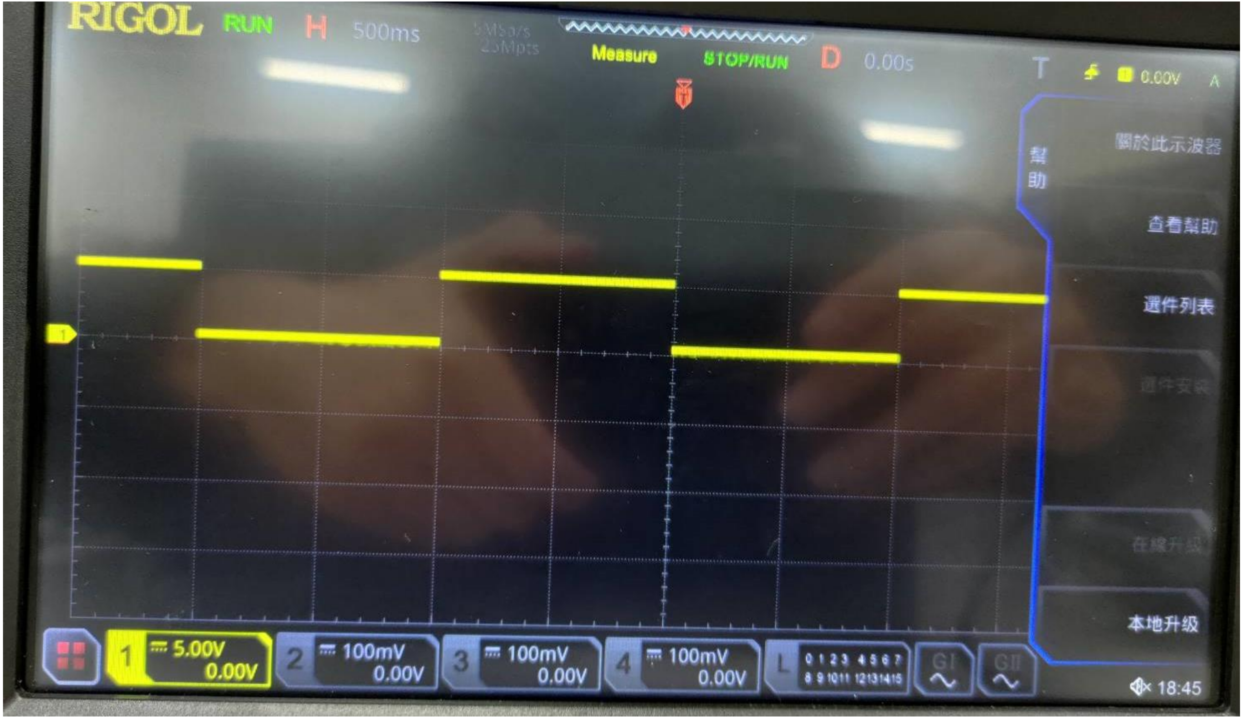
2.

Arduino D4 ouput waveform with frequency 0.5Hz.



3.

Arduino D4 ouput waveform with frequency 2Hz.

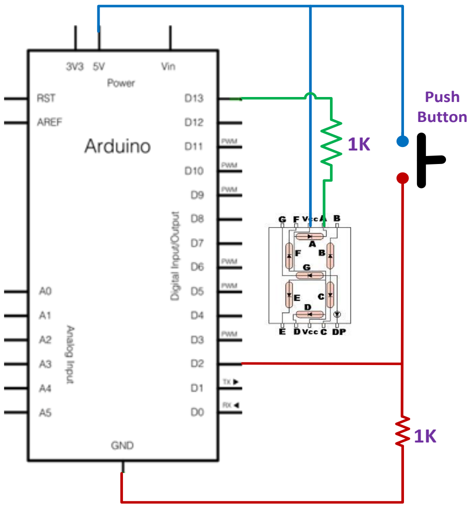


Question:

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

當 Output 為 low 時 LED 會亮，因為是採用共陽極的接法。

Experiment 2: LED Blink with push button.



1.

Is this circuit a pull-high or pull-low type? _pull low___________.

2.

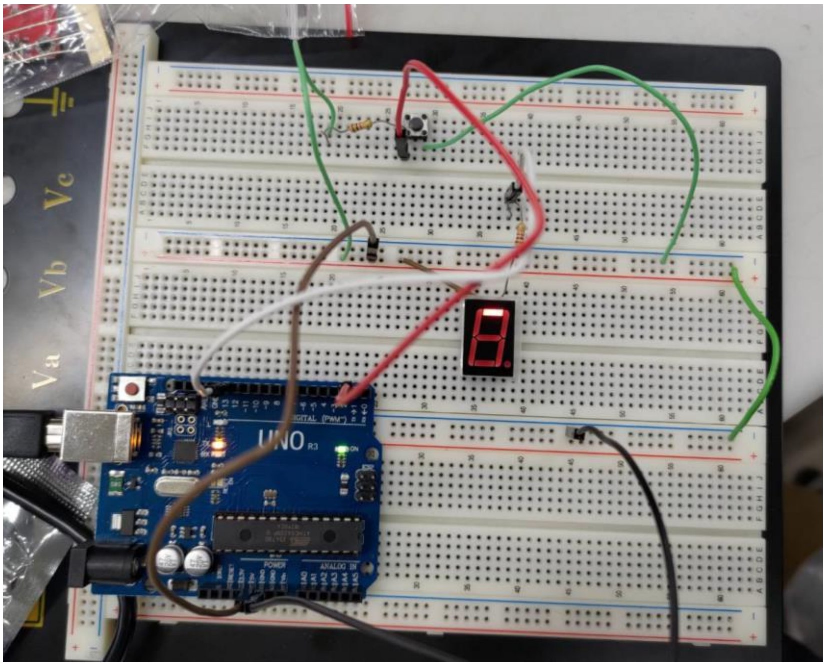
Measure pin D2 voltage using voltage meter:

Behavior	DC Voltage (V)
Pushbutton pressed	**4.95**
Pushbutton released	**0**

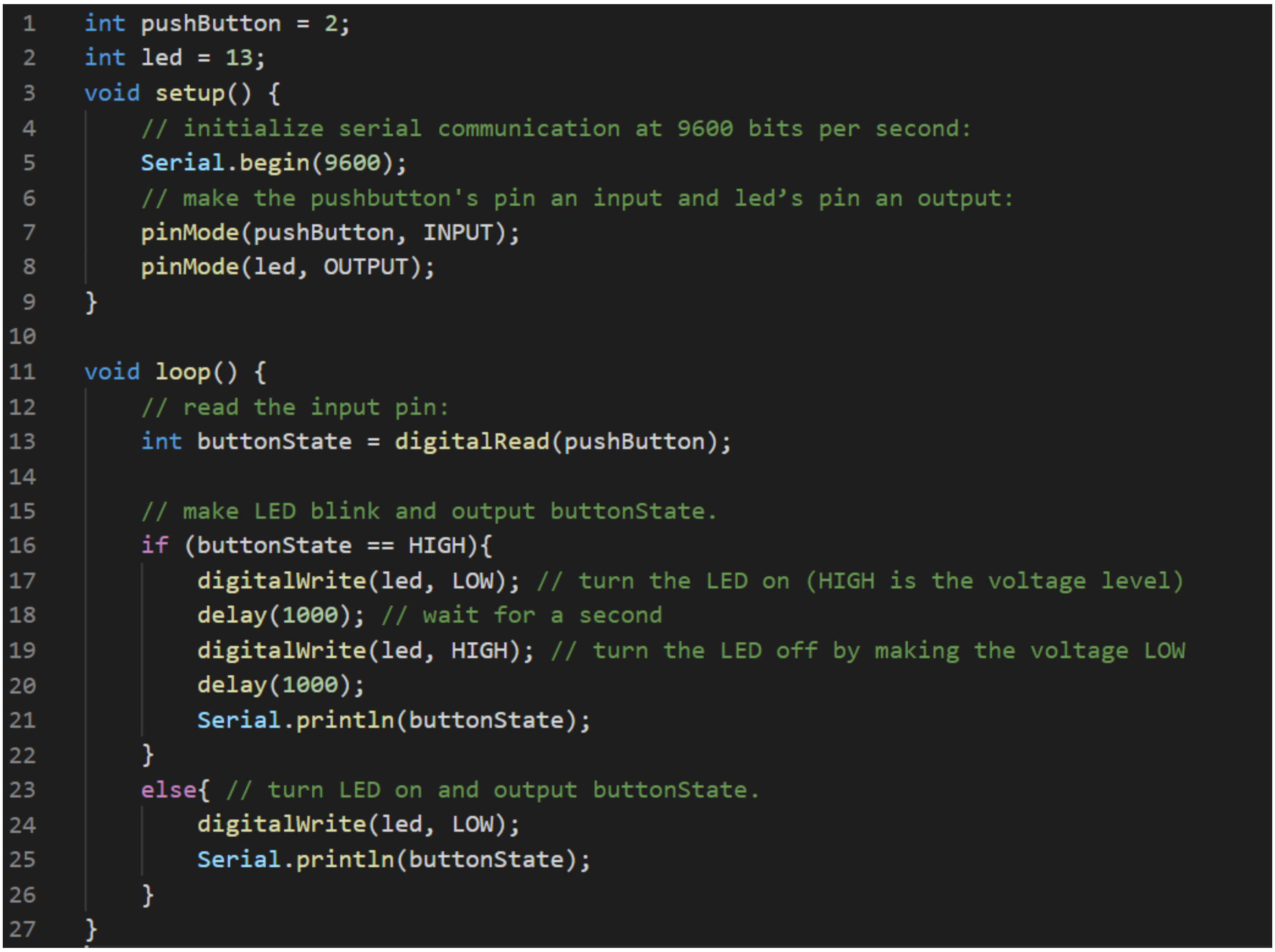
3.

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

Modified circuit:



Modified sketch:



```
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 (HIGH is the voltage level)
        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);
    }
}
```

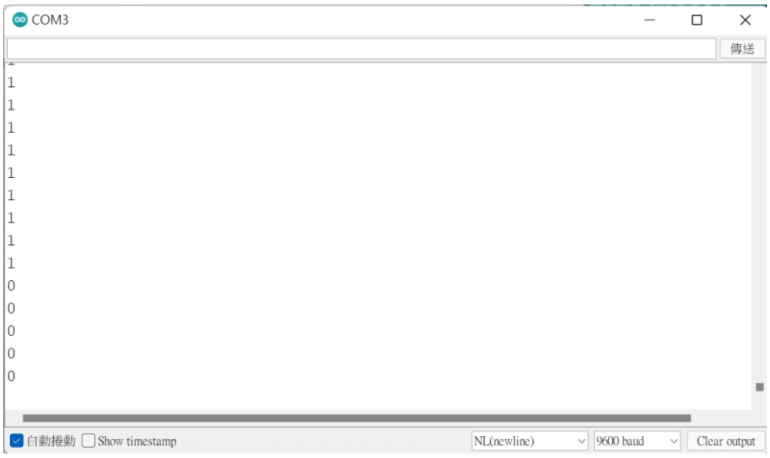
4.

Measure pin D2 voltage using voltage meter after step 3:

Behavior	DC Voltage (V)
Pushbutton pressed	**0**
Pushbutton released	**4.96**

5.

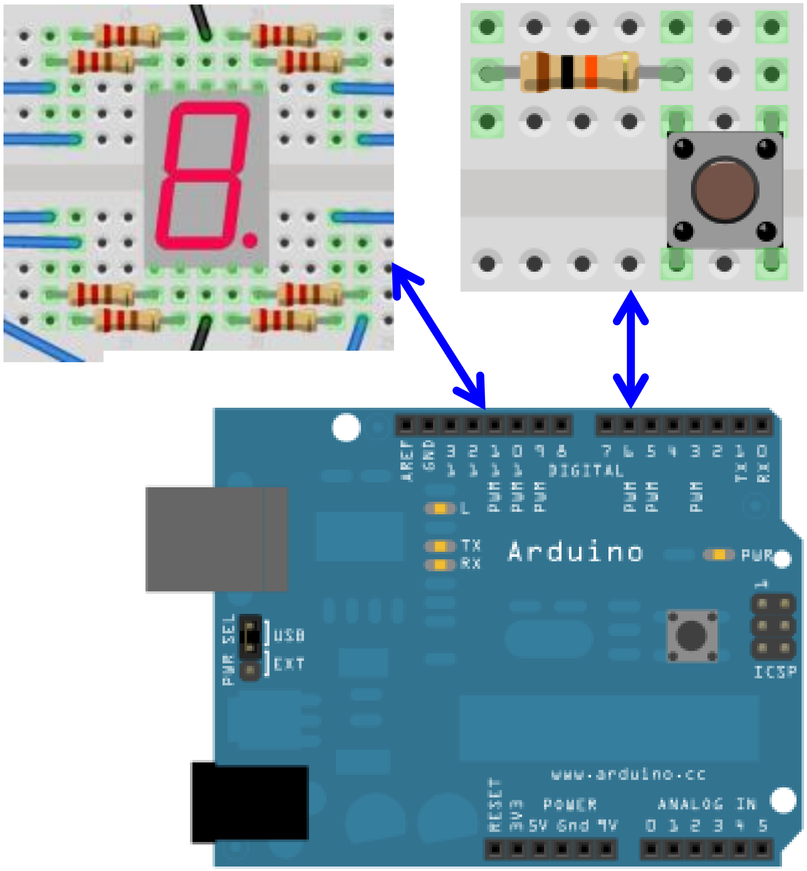
Attach your serial monitor screen shot



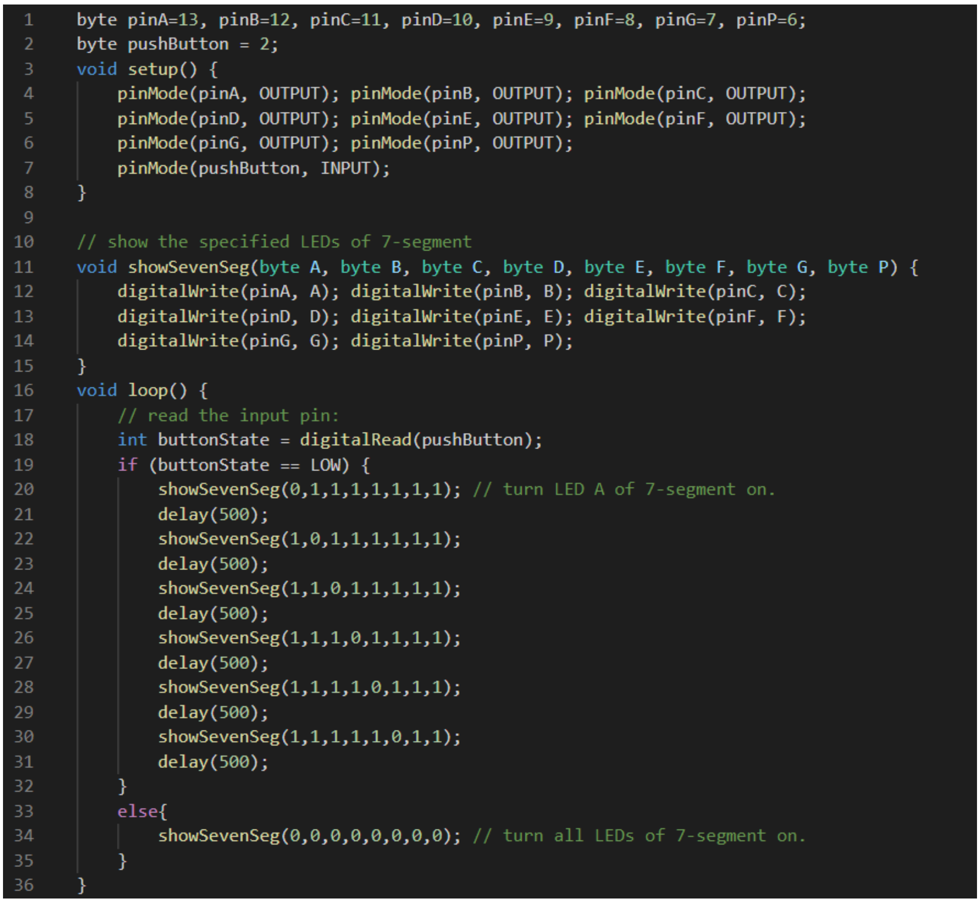
```
1
1
1
1
1
1
1
1
1
0
0
0
0
0
```

☑ 自動捲動 □ Show timestamp NL(newline) 9600 baud

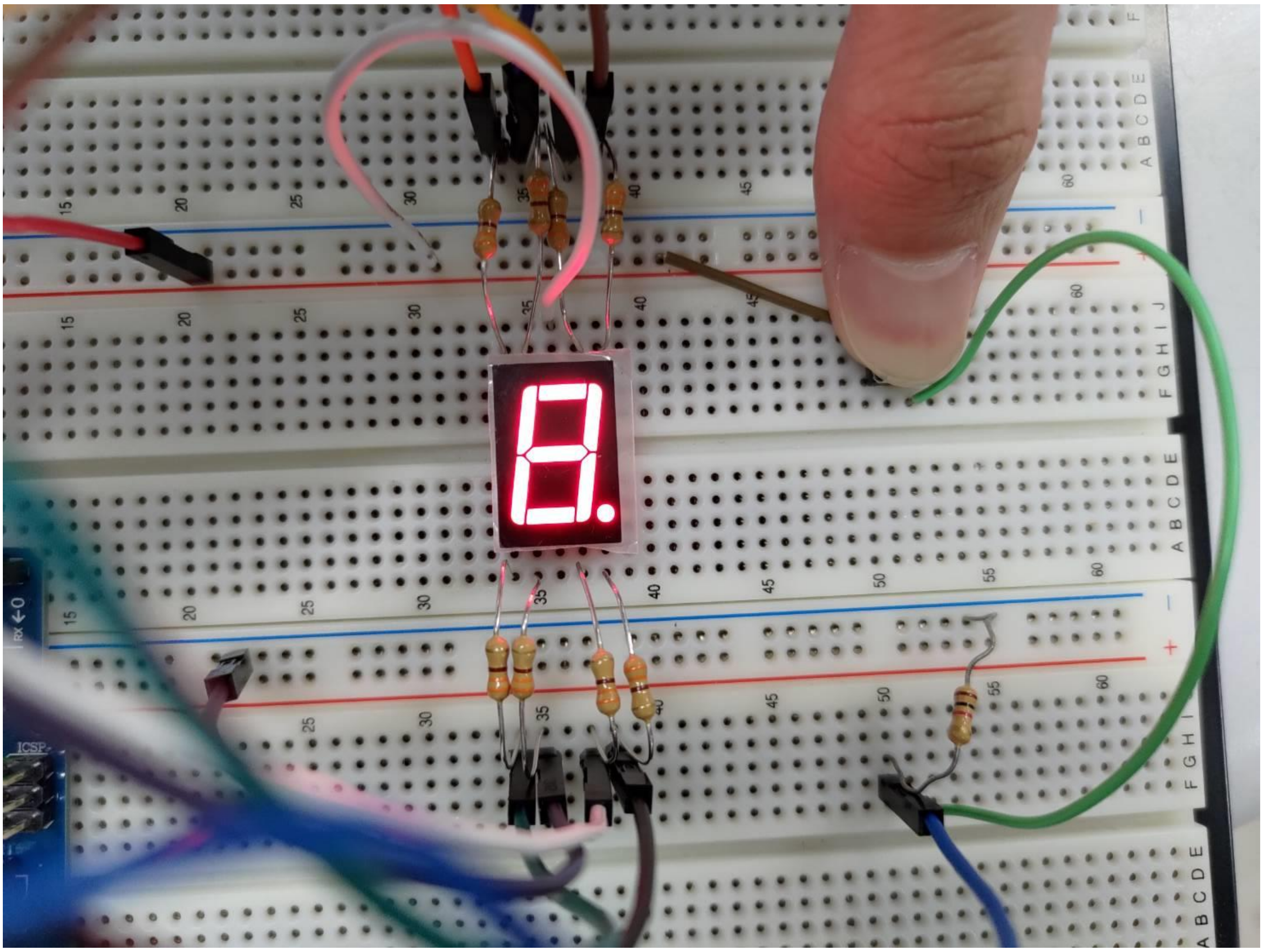
Experiment 3: LED Marquee.



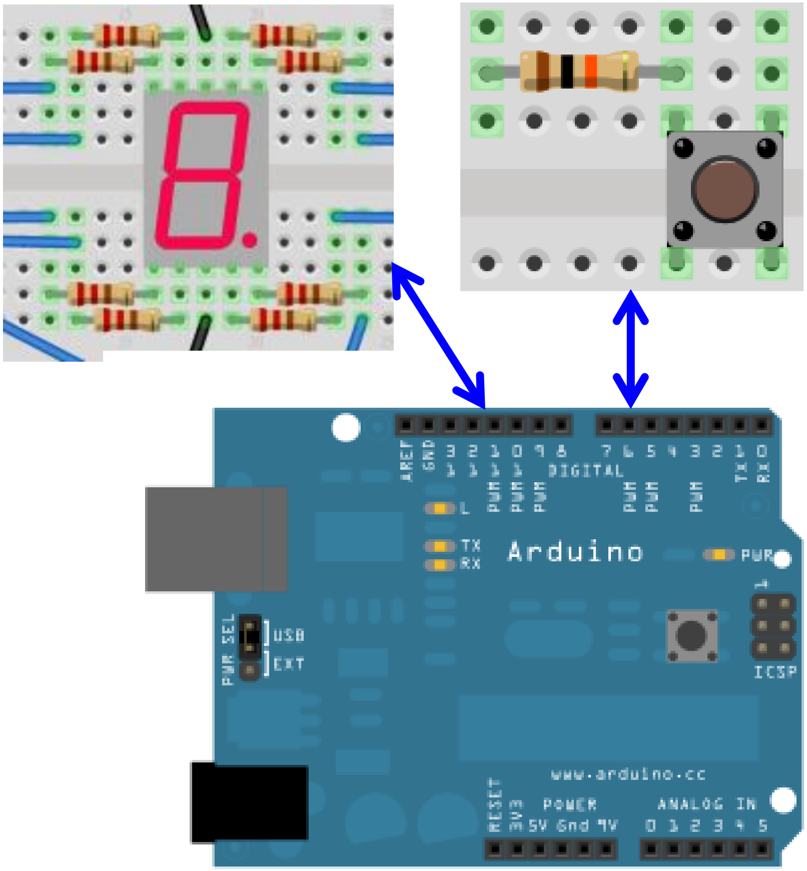
Attach your sketch and result pictures:



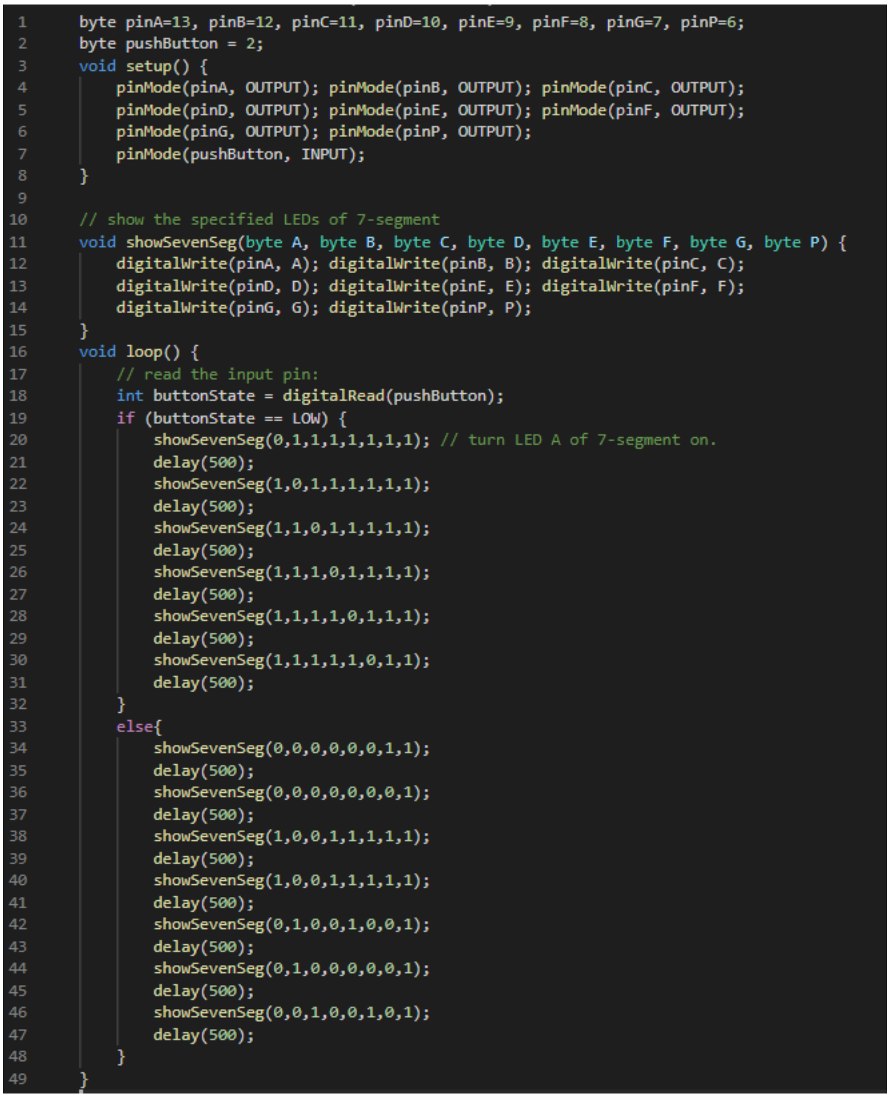
```
byte pinA=13, pinB=12, pinC=11, pinD=10, pinE=9, pinF=8, pinG=7, pinP=6;
byte pushButton = 2;
void setup() {
    pinMode(pinA, OUTPUT); pinMode(pinB, OUTPUT); pinMode(pinC, OUTPUT);
    pinMode(pinD, OUTPUT); pinMode(pinE, OUTPUT); pinMode(pinF, OUTPUT);
    pinMode(pinG, OUTPUT); pinMode(pinP, OUTPUT);
    pinMode(pushButton, INPUT);
}

// show the specified LEDs of 7-segment
void showSevenSeg(byte A, byte B, byte C, byte D, byte E, byte F, byte G, byte P) {
    digitalWrite(pinA, A); digitalWrite(pinB, B); digitalWrite(pinC, C);
    digitalWrite(pinD, D); digitalWrite(pinE, E); digitalWrite(pinF, F);
    digitalWrite(pinG, G); digitalWrite(pinP, P);
}
void loop() {
    // read the input pin:
    int buttonState = digitalRead(pushButton);
    if (buttonState == LOW) {
        showSevenSeg(0,1,1,1,1,1,1,1); // turn LED A of 7-segment on.
        delay(500);
        showSevenSeg(1,0,1,1,1,1,1,1);
        delay(500);
        showSevenSeg(1,1,0,1,1,1,1,1);
        delay(500);
        showSevenSeg(1,1,1,0,1,1,1,1);
        delay(500);
        showSevenSeg(1,1,1,1,0,1,1,1);
        delay(500);
        showSevenSeg(1,1,1,1,1,0,1,1);
        delay(500);
    }
    else{
        showSevenSeg(0,0,0,0,0,0,0,0); // turn all LEDs of 7-segment on.
    }
}
```



Experiment 4: student ID Marquee.



Attach your sketch and result pictures:



```
byte pinA=13, pinB=12, pinC=11, pinD=10, pinE=9, pinF=8, pinG=7, pinP=6;
byte pushButton = 2;
void setup() {
    pinMode(pinA, OUTPUT); pinMode(pinB, OUTPUT); pinMode(pinC, OUTPUT);
    pinMode(pinD, OUTPUT); pinMode(pinE, OUTPUT); pinMode(pinF, OUTPUT);
    pinMode(pinG, OUTPUT); pinMode(pinP, OUTPUT);
    pinMode(pushButton, INPUT);
}

// show the specified LEDs of 7-segment
void showSevenSeg(byte A, byte B, byte C, byte D, byte E, byte F, byte G, byte P) {
    digitalWrite(pinA, A); digitalWrite(pinB, B); digitalWrite(pinC, C);
    digitalWrite(pinD, D); digitalWrite(pinE, E); digitalWrite(pinF, F);
    digitalWrite(pinG, G); digitalWrite(pinP, P);
}
void loop() {
    // read the input pin:
    int buttonState = digitalRead(pushButton);
    if (buttonState == LOW) {
        showSevenSeg(0,1,1,1,1,1,1,1); // turn LED A of 7-segment on.
        delay(500);
        showSevenSeg(1,0,1,1,1,1,1,1);
        delay(500);
        showSevenSeg(1,1,0,1,1,1,1,1);
        delay(500);
        showSevenSeg(1,1,1,0,1,1,1,1);
        delay(500);
        showSevenSeg(1,1,1,1,0,1,1,1);
        delay(500);
        showSevenSeg(1,1,1,1,1,0,1,1);
        delay(500);
    }
    else{
        showSevenSeg(0,0,0,0,0,0,1,1);
        delay(500);
        showSevenSeg(0,0,0,0,0,0,0,1);
        delay(500);
        showSevenSeg(1,0,0,1,1,1,1,1);
        delay(500);
        showSevenSeg(1,0,0,1,1,1,1,1);
        delay(500);
        showSevenSeg(0,1,0,0,1,0,0,1);
        delay(500);
        showSevenSeg(0,1,0,0,0,0,0,1);
        delay(500);
        showSevenSeg(0,0,1,0,0,1,0,1);
        delay(500);
    }
}
```

Experiment 5: Up/down counter.

本題的電路其功能為一個一位數的可上數和下數的計數器。電路啟動後，七段顯示器與 serial monitor 會從 0 開始往上數，數到 9 後會自動從 0 再從頭開始數，例如：0123456789012345……，此時若按外接的按鈕一下（所謂的按一下，指的是先按下按鈕，接著放開按鈕後才開始反應，類似於點擊電腦滑鼠時的動作），則會改為下數，數到 0 後會自動從 9 再從頭開始數，例如原本從 0 開始往上數到 8，若此時你按一下按鈕，則七段顯示器與 serial monitor 會顯示 76543210987654321098765……，如果接下來你又按一下按鈕，則再次切換成上數，以此類推。

此題的電路與 Experiment 4 一樣無須改變，需要改的是程式，其中偵測按鈕是否按下的部份，請使用中斷指令，該指令的語法如下：

attachInterrupt(digitalPinToInterrupt(pin), ISR, mode);

其中：

pin：the pin number you want to interrupt

ISR：the ISR to call when the interrupt occurs.

This function must take no parameters and return nothing. This function is sometimes referred to as an interrupt service routine.

mode：defines when the interrupt should be triggered.

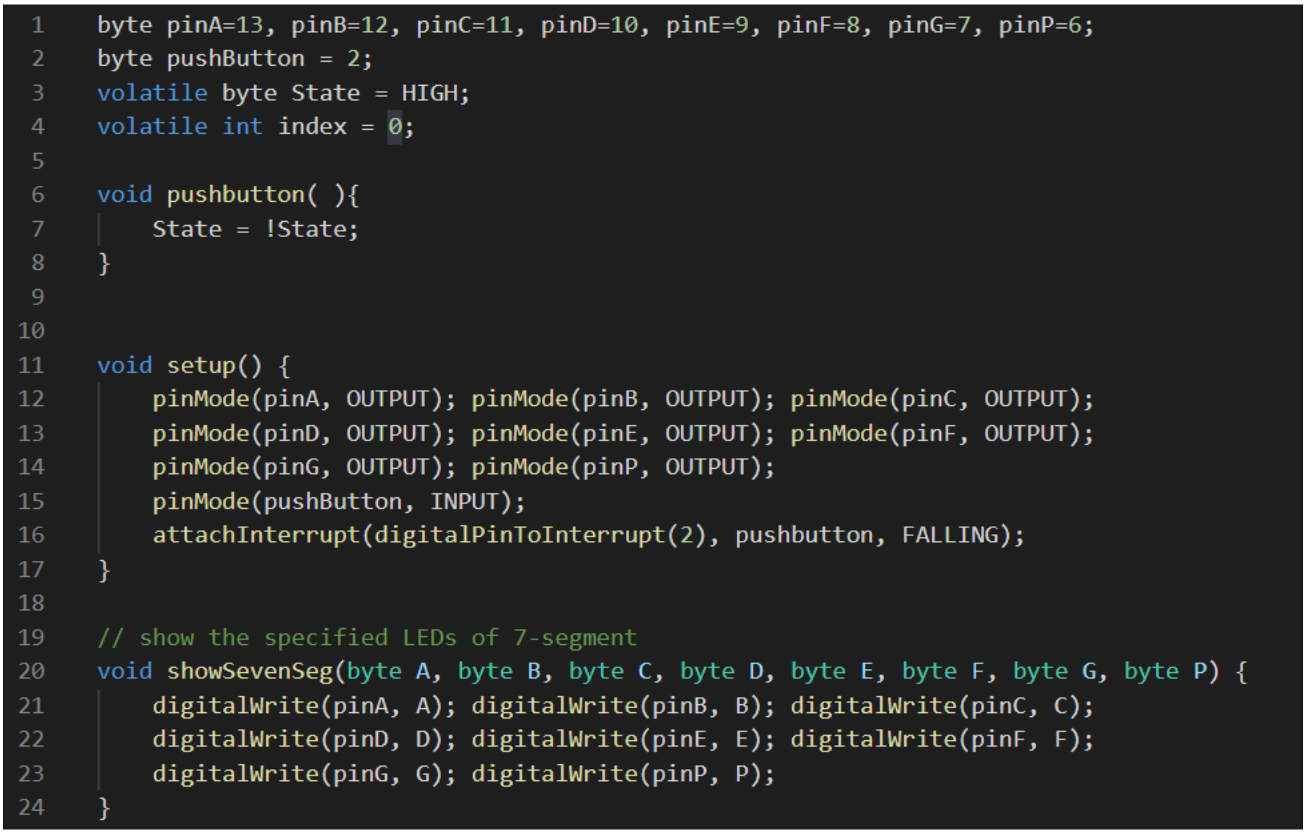
Four contstants are predefined as valid values：

LOW to trigger the interrupt whenever the pin is low,

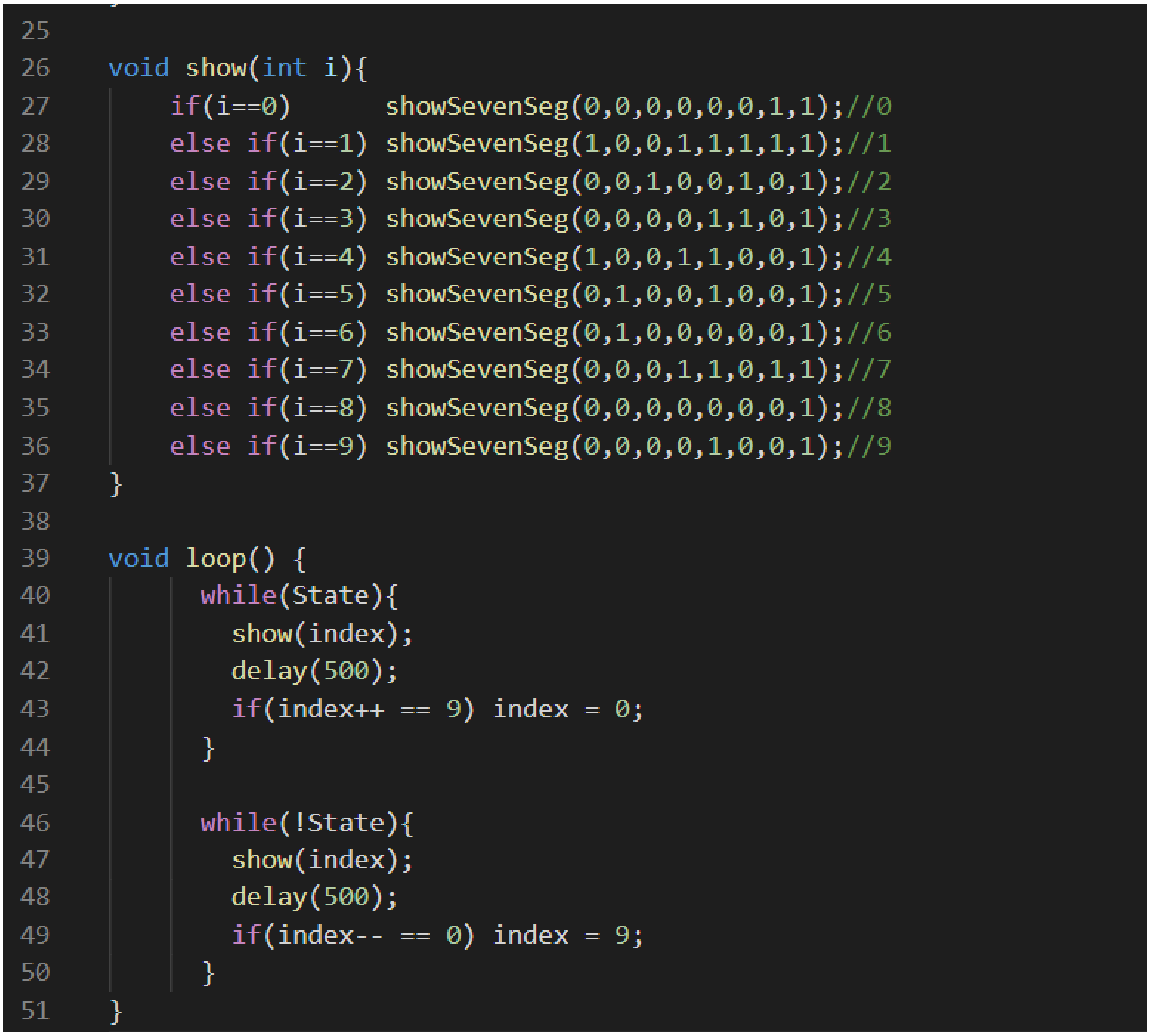
CHANGE to trigger the interrupt whenever the pin changes value,

RISING to trigger when the pin goes from low to high,

FALLING for when the pin goes from high to low.



```
byte pinA=13, pinB=12, pinC=11, pinD=10, pinE=9, pinF=8, pinG=7, pinP=6;
byte pushButton = 2;
volatile byte State = HIGH;
volatile int index = 0;

void pushbutton( ){
    State = !State;
}


void setup() {
    pinMode(pinA, OUTPUT); pinMode(pinB, OUTPUT); pinMode(pinC, OUTPUT);
    pinMode(pinD, OUTPUT); pinMode(pinE, OUTPUT); pinMode(pinF, OUTPUT);
    pinMode(pinG, OUTPUT); pinMode(pinP, OUTPUT);
    pinMode(pushButton, INPUT);
    attachInterrupt(digitalPinToInterrupt(2), pushbutton, FALLING);
}

// show the specified LEDs of 7-segment
void showSevenSeg(byte A, byte B, byte C, byte D, byte E, byte F, byte G, byte P) {
    digitalWrite(pinA, A); digitalWrite(pinB, B); digitalWrite(pinC, C);
    digitalWrite(pinD, D); digitalWrite(pinE, E); digitalWrite(pinF, F);
    digitalWrite(pinG, G); digitalWrite(pinP, P);
}
```



```
void show(int i){
    if(i==0)      showSevenSeg(0,0,0,0,0,0,1,1);//0
    else if(i==1) showSevenSeg(1,0,0,1,1,1,1,1);//1
    else if(i==2) showSevenSeg(0,0,1,0,0,1,0,1);//2
    else if(i==3) showSevenSeg(0,0,0,0,1,1,0,1);//3
    else if(i==4) showSevenSeg(1,0,0,1,1,0,0,1);//4
    else if(i==5) showSevenSeg(0,1,0,0,1,0,0,1);//5
    else if(i==6) showSevenSeg(0,1,0,0,0,0,0,1);//6
    else if(i==7) showSevenSeg(0,0,0,1,1,0,1,1);//7
    else if(i==8) showSevenSeg(0,0,0,0,0,0,0,1);//8
    else if(i==9) showSevenSeg(0,0,0,0,1,0,0,1);//9
}

void loop() {
      while(State){
        show(index);
        delay(500);
        if(index++ == 9) index = 0;
      }

      while(!State){
        show(index);
        delay(500);
        if(index-- == 0) index = 9;
      }
}
```

最後這邊的 bonus 我卡了好久，一開始連 interrupt 具體來說是如何實現的都不太清楚，因此就開始上網查詢這個 funtion 的定義及使用方式，一旦了解定義之後，我便想起之前邏輯設計和計算機組織寫 verilog 都有用到的 finite state machine，一旦定義好 state 與下一個 state 之間的關係及轉換條件後，便可以輕鬆的做出來了。雖然 Arduino 的語法跟 C 有點像，但實作起來的感覺卻是大相逕庭，也有可能是因為才剛入門而已，希望日後的實驗能夠順利許多!