

雲林科技大學

資訊工程系所

Week16

遊戲搖桿與 8×8 矩陣之應用

指導教授:陳木中 教授

授課學生:詹沐恩 學生

中華民國 113 年 6 月 13 日

A. 搖桿感測器的應用(Analog Joystick)

➤ 程式碼

(VERT_PIN、HORZ_PIN 和 SEL_PIN) 作為輸入。主迴圈中持續讀取這些引腳的模擬值並將其輸出到序列監視器，還檢查 SEL_PIN 的數位值是否為低電平來判斷按鈕是否按下。每次讀取和輸出之間會有 0.5 秒的延遲。

```
#define VERT_PIN A0
#define HORZ_PIN A1
#define SEL_PIN 3

void setup() {
  // put your setup code here, to run once:
  Serial.begin(9600);
  pinMode(VERT_PIN, INPUT);
  pinMode(HORZ_PIN, INPUT);
  pinMode(SEL_PIN, INPUT);
}

void loop() {
  // put your main code here, to run repeatedly:
  int vert=analogRead(VERT_PIN);
  int horz=analogRead(HORZ_PIN);
  int sel=analogRead(SEL_PIN);
  Serial.print("VERT_PIN = ");
  Serial.print(vert);
  Serial.print("  HORZ_PIN = ");
  Serial.print(horz);
  Serial.print("  SEL_PIN = ");
  Serial.println(sel);
  bool selPressed=digitalRead(SEL_PIN)==LOW;
  delay(500);}
```

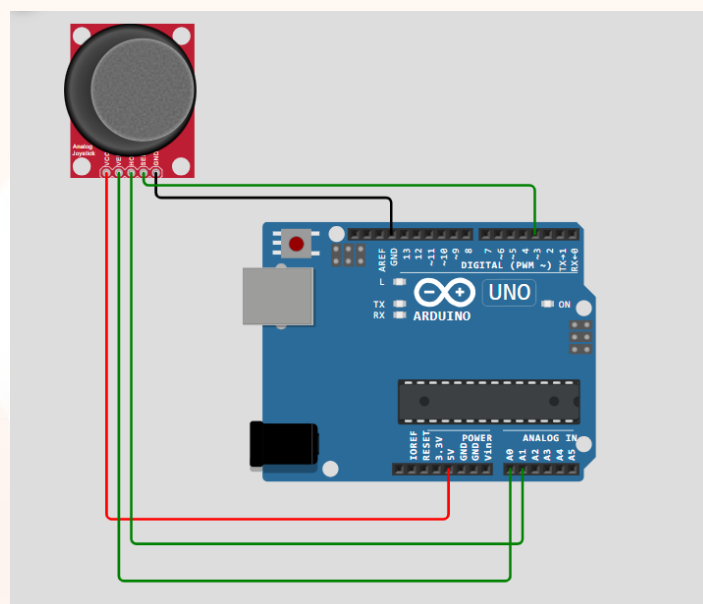
➤ 使用元件(Analog Joystick)

1. 模擬搖桿是一種輸入設備，具有兩個模擬輸出軸，用於測量水平和垂直方向的移動，並配有一個數位按鈕，當搖桿被按下時會觸發。這些特性使其在遊戲控制和機器人控制等應用中廣泛使用。通過讀取模擬輸出值和按鈕狀態，實現精確的方向和動作控制。



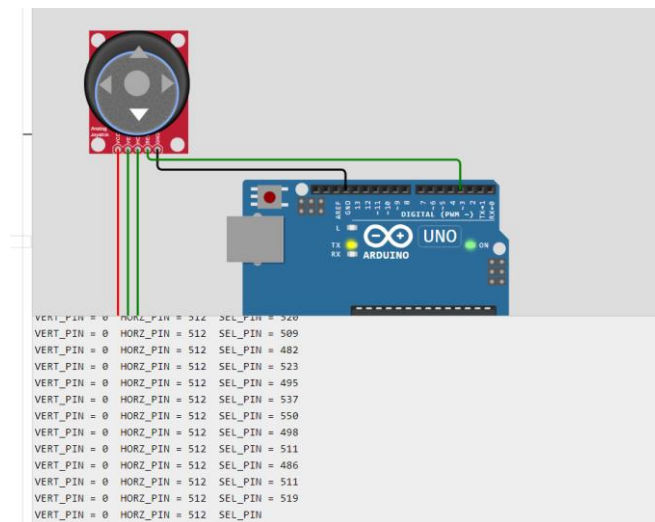
➤ 接線電路

1. 將 Analog Joystick 模組與 Arduino 做連接，Analog Joystick 接 3、A0、A1 接腳，+ 接 vcc - 接 GND

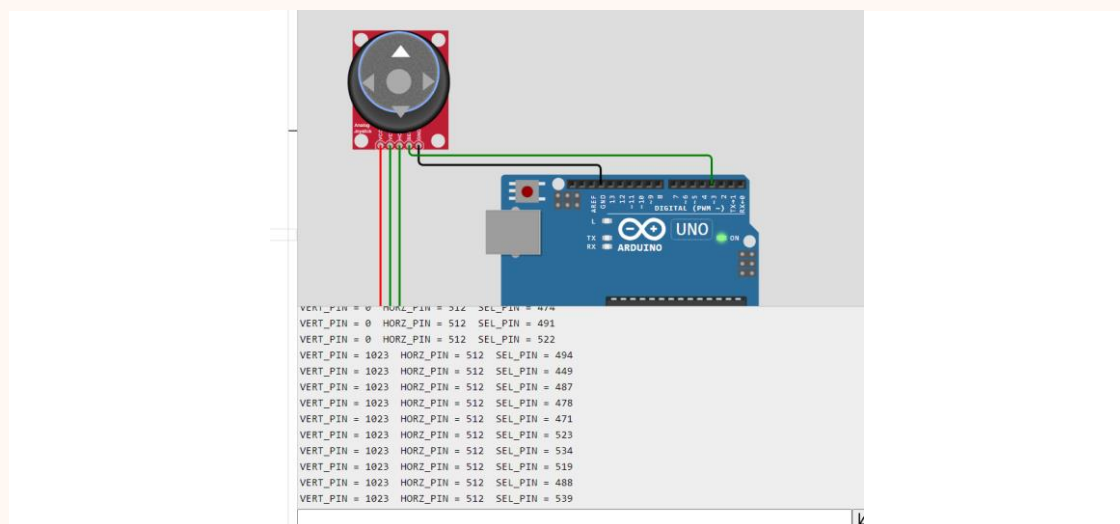


➤ 實驗結果

1. 模擬量角器輸出的垂直和水平數據能被成功讀取並顯示在串列監視器上。按鈕的狀態也被正確讀取並顯示，但使用了模擬輸入進行讀取，導致按鈕狀態的檢測不準確。程式每隔 500 毫秒讀取並顯示一次這些數據，證明了連續監測功能的可行性。

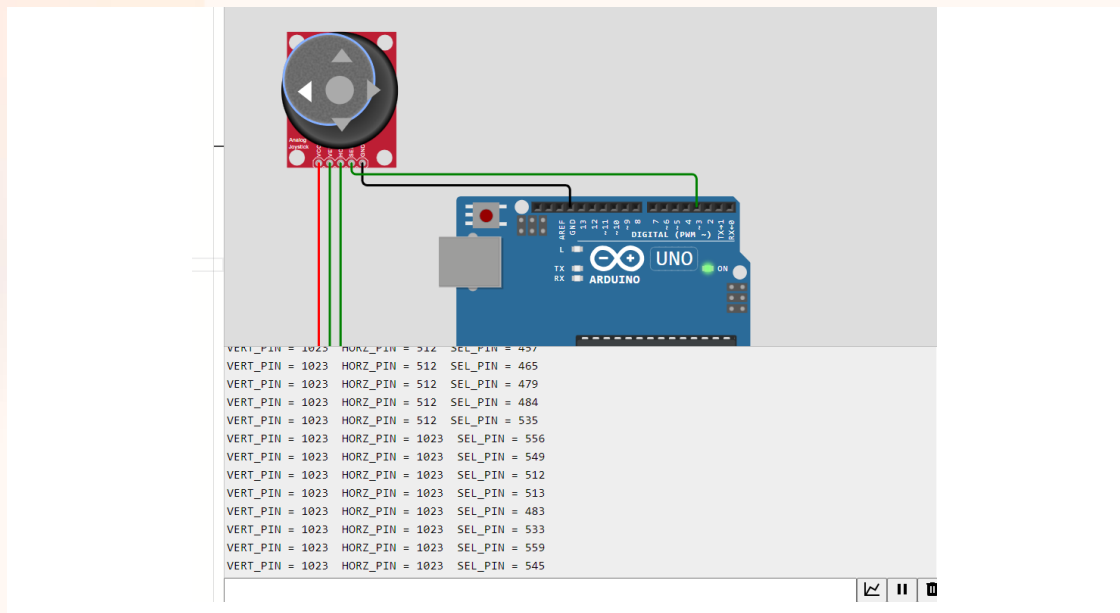


實驗結果: VERT=0(向下)

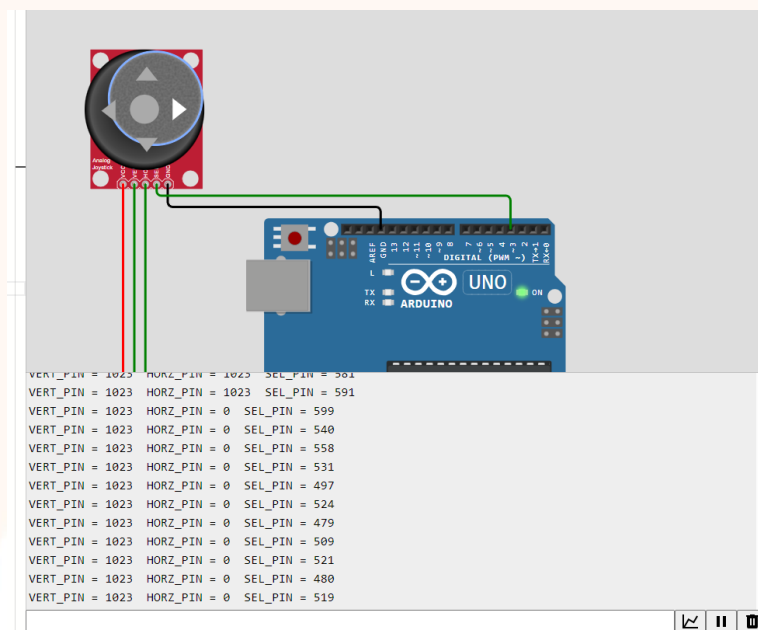


實驗結果: VERT=1023 (向上)

➤ 實驗結果



實驗結果: HORZ=1023 (向左)



實驗結果: HORZ=0 (向右)

B. 遊戲搖感測器控制 LED(Analog Joystick)

➤ 程式碼

第一個設置了一個 LED 並監視模擬數位 A0 的數據，並透過串行端口輸出。第二個程式定義了三個模擬輸入引腳並讀取它們的數值，然後根據這些數值控制數個輸出引腳。

```
void setup()
{
    pinMode(LED_BUILTIN, OUTPUT);
    Serial.begin(9600);
}

void loop()
{
    int data=analogRead(A0);
    Serial.print("A0= ");
    Serial.println(data);
    delay(1000);
}

#define VERT_PIN A0
#define HORZ_PIN A1
#define SEL_PIN 3

void setup() {
    // put your setup code here, to run once:
    Serial.begin(9600);
    pinMode(VERT_PIN, INPUT);
    pinMode(HORZ_PIN, INPUT);
    pinMode(SEL_PIN, INPUT);
}

void loop() {
    // put your main code here, to run repeatedly:
    int vert=analogRead(VERT_PIN);
```

```
int horz=analogRead(HORZ_PIN);
int sel=analogRead(SEL_PIN);
Serial.print("VERT_PIN = ");

Serial.print(ver);
Serial.print("  HORZ_PIN = ");
Serial.print(horz);
Serial.print("  SEL_PIN = ");
Serial.println(sel);
bool selPressed=digitalRead(SEL_PIN)==LOW;
delay(500);
if(ver>600)
{
    digitalWrite(11, HIGH);
    delay(500);
}
else if(ver<300)
{
    digitalWrite(10, HIGH);
    delay(500);
}
if(horz>600)
{
    digitalWrite(9, HIGH);
    delay(500);
}
else if(horz<300)
{
    digitalWrite(8, HIGH);
    delay(500);
}
digitalWrite(8,LOW);
digitalWrite(9,LOW);
digitalWrite(10,LOW);
digitalWrite(11,LOW);
```

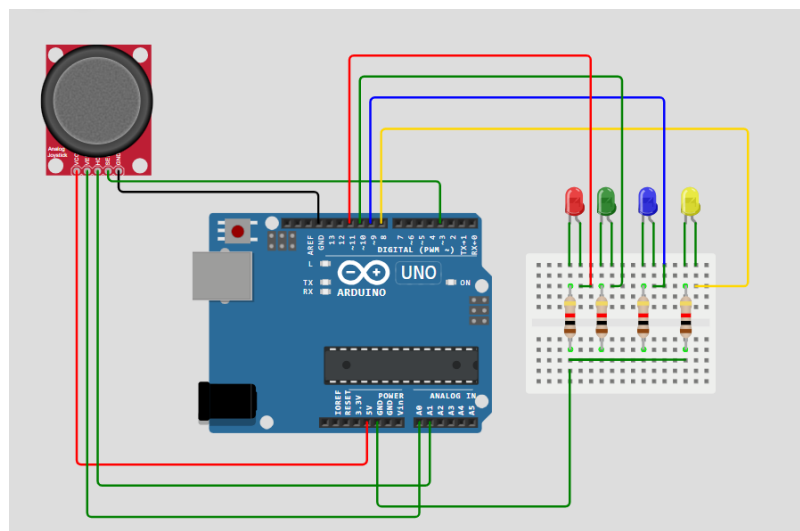
➤ 使用元件(Analog Joystick)

1. 模擬搖桿是一種輸入設備，具有兩個模擬輸出軸，用於測量水平和垂直方向的移動，並配有一個數位按鈕，當搖桿被按下時會觸發。這些特性使其在遊戲控制和機器人控制等應用中廣泛使用。通過讀取模擬輸出值和按鈕狀態，實現精確的方向和動作控制。



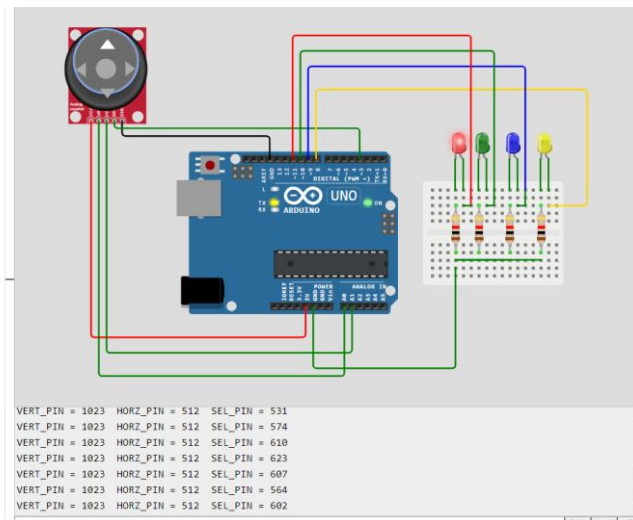
➤ 接線電路

1.將 Force Sensor 模組與 Arduino 做連接，Force Sensor 接 A0 接腳，+ 接 vcc -接 GND

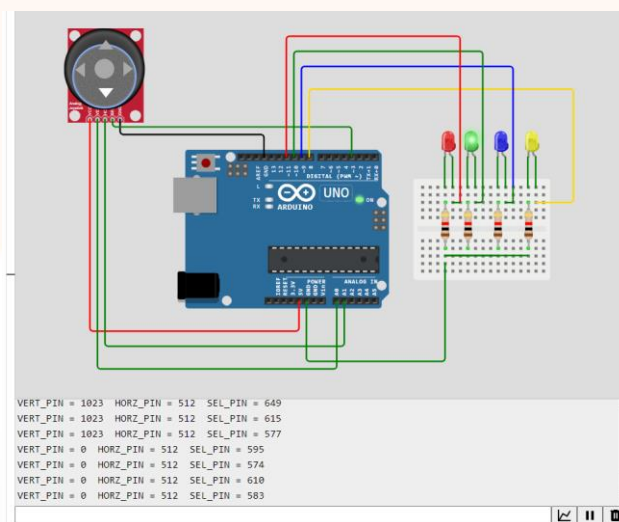


➤ 實驗結果

1. 當垂直引腳或水平引腳的模擬讀數超過 600 時，對應的 LED 燈會被點亮，且低於 300 時亦然。這些讀數會每 500 毫秒檢查一次，並在檢查後將所有 LED 燈關閉。每次讀取的數據會通過串行端口打印出來以供監控。

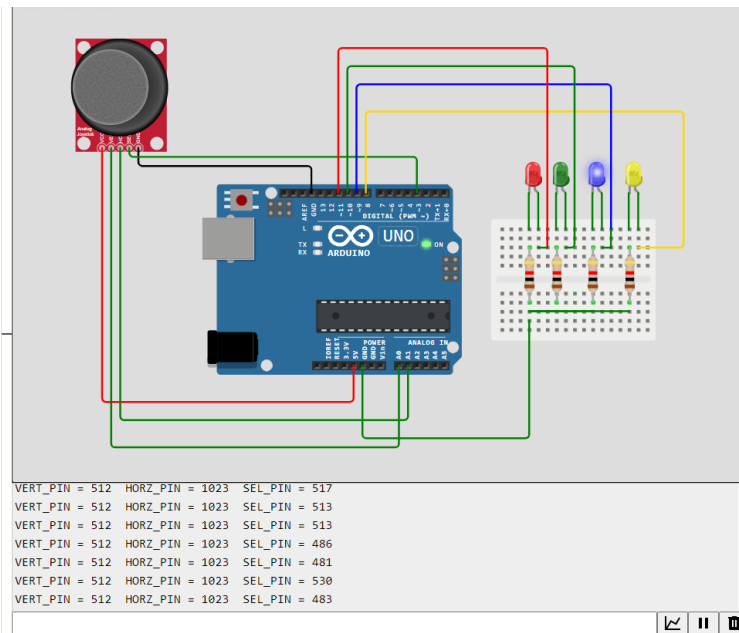


實驗結果: VERT=1023 (向上紅燈亮)

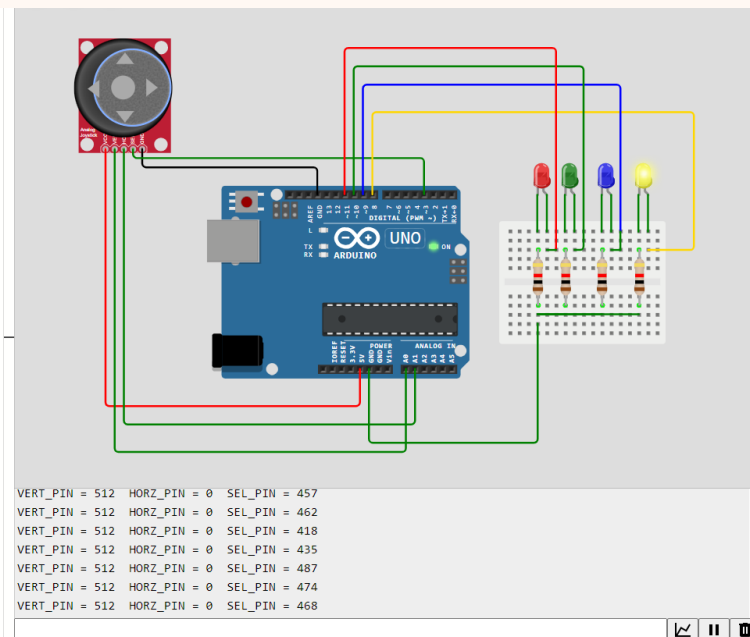


➤ 實驗結果: VERT=0 (向下綠燈亮)

➤ 實驗結果



➤ 實驗結果: HORZ=1023 (向左藍燈亮)



➤ 實驗結果: HORZ=0 (向右黃燈亮)

C. 8*8 矩陣感測器之應用(8*8matrix)

➤ 程式碼

使用 LedControl 庫控制 LED 矩陣顯示，初始化設定 LED 矩陣的引腳，設定亮度和清除顯示內容，然後在主迴圈中隨機點亮 LED 矩陣中的一個 LED。

```
#include <LedControl.h>

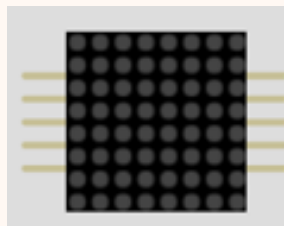
// Pin 11:Data in
// Pin 13: Clock
// Pin 10: CS(Load)
LedControl lc = LedControl(11, 13, 10, 1);

void setup() {
    // put your setup code here, to run once:
    lc.shutdown(0, false); // 關閉省電模式
    lc.setIntensity(0, 8); // 設定亮度為 8 (介於 0~15 之間)
    lc.clearDisplay(0);    // 清除螢幕
}

void loop() {
    // put your main code here, to run repeatedly:
    int randomNumber_row = 0;
    int randomNumber_col = 0;
    // 將 Led 的列 1 欄 3 設定為亮
    lc.setLed(0, randomNumber_row, randomNumber_col, 1); // 將 Led 的欄,列設定為亮
}
```

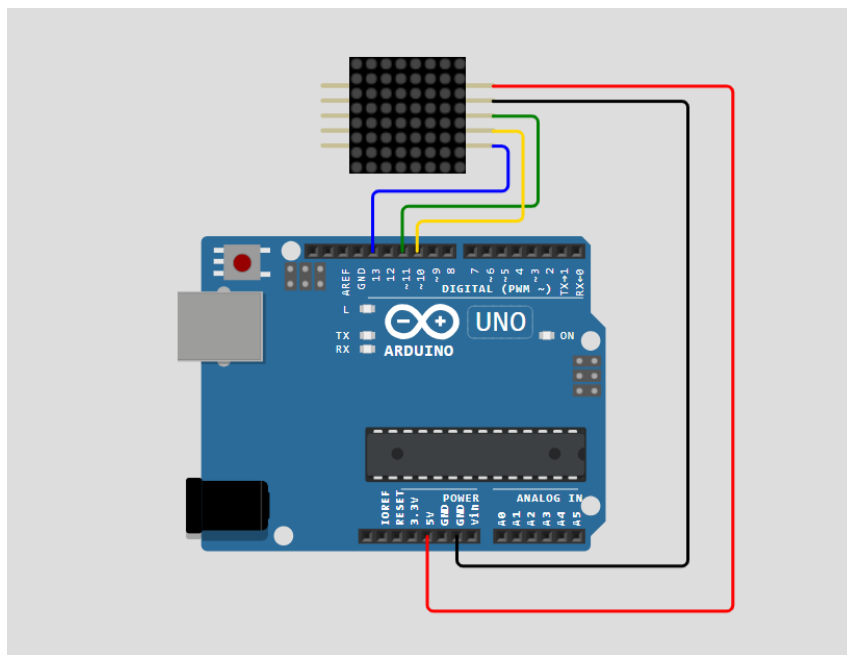
➤ 使用元件(8*8 矩陣感測器)

1. 8*8 矩陣感測器是一種具有 64 個感測元件的裝置，可用於檢測和測量各種物理量，如溫度、光照度或壓力等。它通常以行和列的形式排列，能夠同時探測多個位置或條件變化，廣泛應用於科學研究和工業檢測領域。



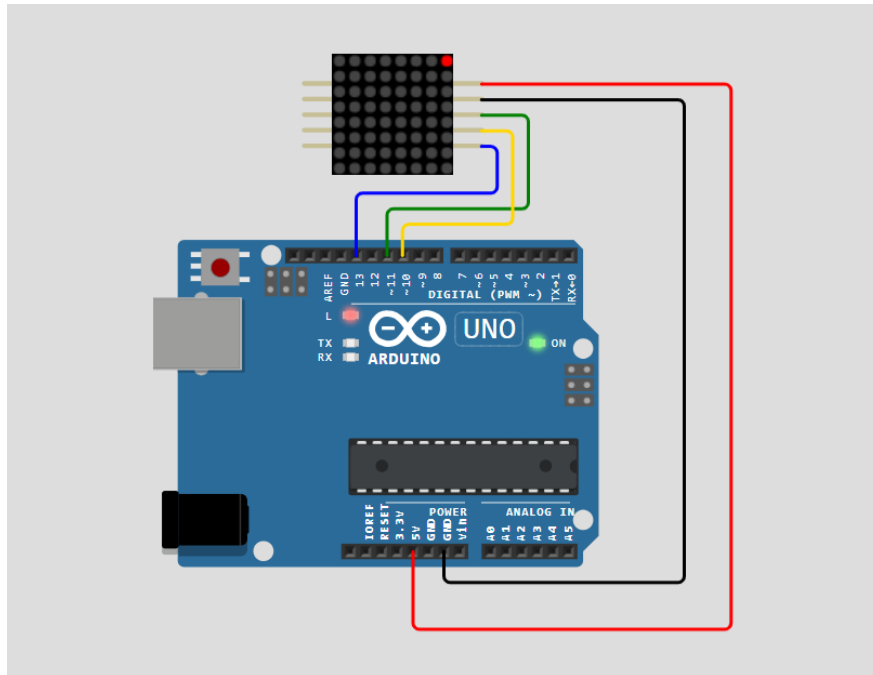
➤ 接線電路

- 1. 將(8*8 矩陣感測器模組與 Arduino 做連接，(8*8 矩陣感測器接 10,11,13 接腳，+ 接 vcc -接 GND



➤ 實驗結果

1. 利用 LedControl 庫控制 LED 矩陣顯示，設置亮度為 8 並清除螢幕，在每個循環中隨機點亮 LED 矩陣的特定位置。



實驗結果: 8*8 矩陣感測器(右上角亮燈)

D. 七段顯示器與曲率感測器的應用

E. 程式碼

控制七段顯示器的輸出。首先定義了兩個陣列，分別是 seg7 和 TAB，用於表示七段顯示器的腳位和顯示的數字對應的位元組。然後在 setup 函數中設置了七段顯示器的腳位為輸出模式。在 loop 函數中讀取 A0 引腳的模擬值，根據計算結果決定顯示的數字，並將對應的位元組傳送到七段顯示器。

```
#include <LedControl.h>

// Pin 11:Data in
// Pin 13: Clock
// Pin 10: CS(Load)
#define VERT_PIN A0
#define HORZ_PIN A1
#define SEL_PIN 3
LedControl lc = LedControl(11, 13, 10, 1);

void setup() {
    // put your setup code here, to run once:
    lc.shutdown(0, false); // 關閉省電模式
    lc.setIntensity(0, 8); // 設定亮度為 8 (介於 0~15 之間)
    lc.clearDisplay(0);    // 清除螢幕
    Serial.begin(9600);
    pinMode(VERT_PIN, INPUT);
    pinMode(HORZ_PIN, INPUT);
    pinMode(SEL_PIN, INPUT);
}
int randomNumber_row = 4;
int randomNumber_col = 4;
```

```

void loop() {
    int vert=analogRead(VERT_PIN);
    int horz=analogRead(HORZ_PIN);
    int sel=analogRead(SEL_PIN);
    Serial.print("VERT_PIN = ");

    Serial.print(vert);
    Serial.print("  HORZ_PIN = ");
    Serial.print(horz);
    Serial.print("  SEL_PIN = ");
    Serial.println(sel);
    bool selPressed=digitalRead(SEL_PIN)==LOW;
    // put your main code here, to run repeatedly:
    // 將 Led 的列 1 欄 3 設定為亮
    if(vert>600)
    {
        lc.setLed(0, 0, 1, 1); // 將 Led 的欄,列設定為亮
    }
    else
    {
        lc.setLed(0, 0, 1, 0); // 將 Led 的欄,列設定為亮

    }
    if(vert<300)
    {
        lc.setLed(0, 2, 1, 1); // 將 Led 的欄,列設定為亮
    }
    else
    {
        lc.setLed(0, 2, 1, 0); // 將 Led 的欄,列設定為亮

    }

    if(horz<300)
    {
        lc.setLed(0, 1, 0, 1); // 將 Led 的欄,列設定為亮  }
    }
    else

```

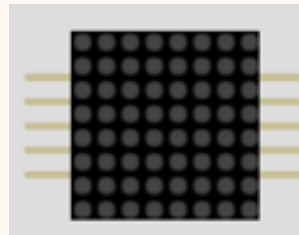
```

{
    lc.setLed(0, 1, 0, 0); // 將 Led 的欄,列設定為亮}
if(horz>600){
    lc.setLed(0, 1, 2, 1); // 將 Led 的欄,列設定為亮  }
    else{ lc.setLed(0, 1, 2, 0); // 將 Led 的欄,列設定為亮
}}

```

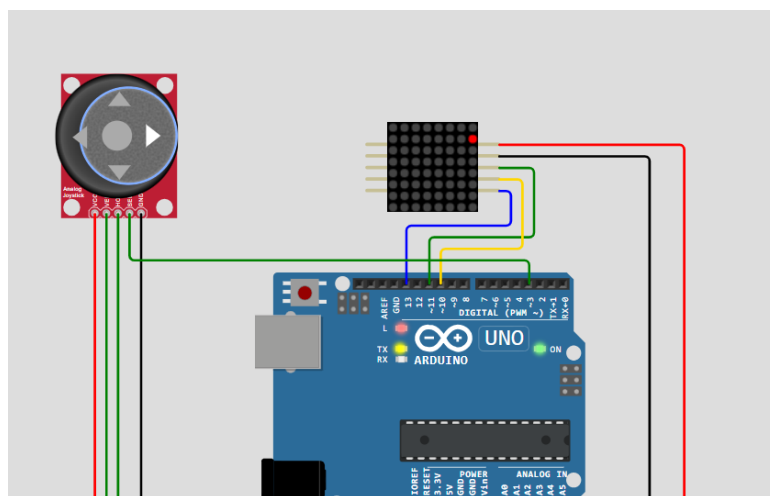
➤ 使用元件(8*8 矩陣感測器)

1. 8*8 矩陣感測器是一種具有 64 個感測元件的裝置，可用於檢測和測量各種物理量，如溫度、光照度或壓力等。它通常以行和列的形式排列，能夠同時探測多個位置或條件變化，廣泛應用於科學研究和工業檢測領域。



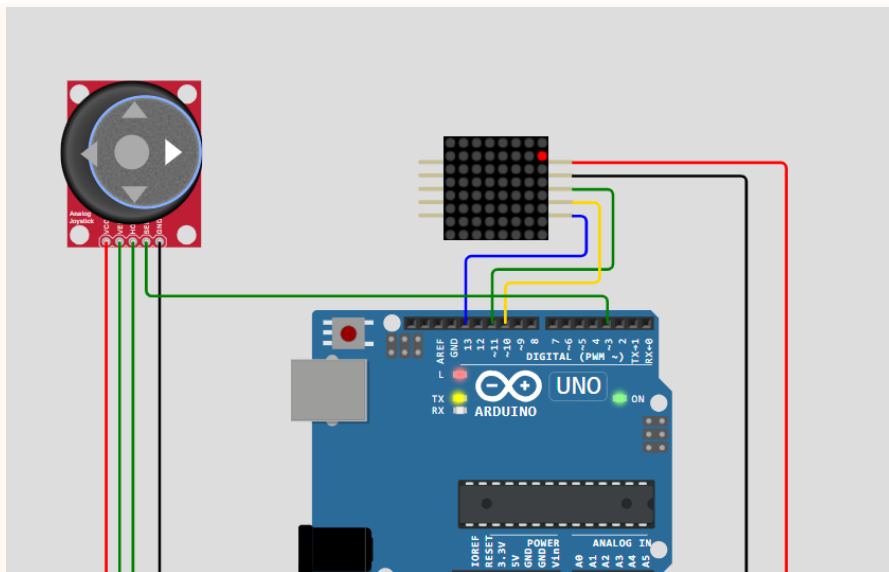
➤ 接線電路

- 1. 將七段顯示器與 Arduino 做連接，七段顯示器接 1-10 接腳，+ 接 vcc - 接 GND

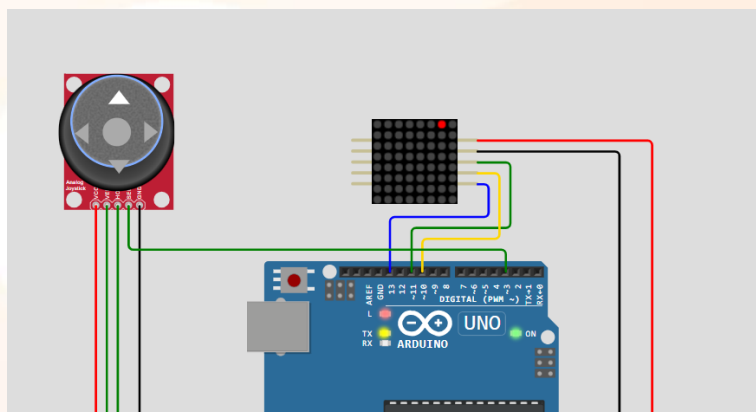


➤ 實驗結果

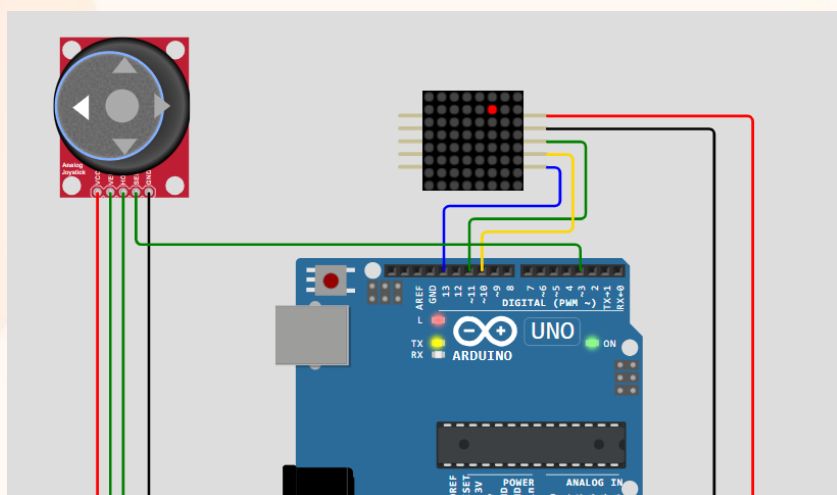
1. Tilt Sensor Tilt Sensor 是一種用來測量物體傾斜角度的裝置，常用於自動化系統、電子設備及安全系統中。它通過檢測重力的變化來確定物體的傾斜方向和角度，提供精確的角度信息。



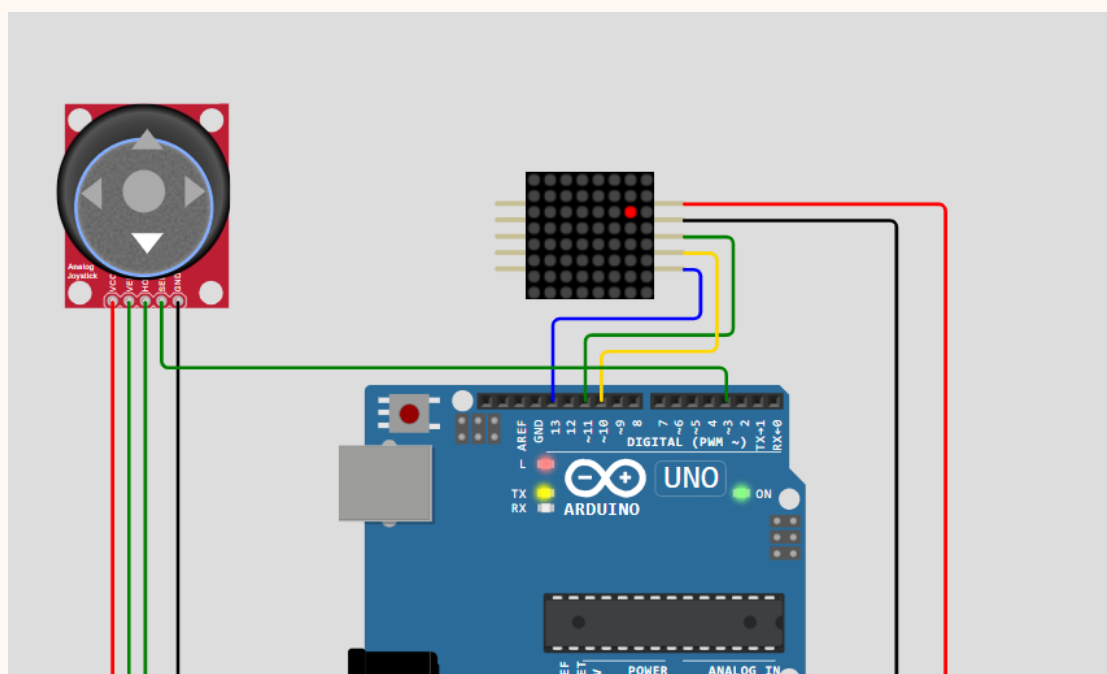
實驗結果: 8*8 矩陣感測器(右邊亮燈)



實驗結果: 8*8 矩陣感測器(上邊亮燈)



實驗結果: 8*8 矩陣感測器(左邊亮燈)



實驗結果: 8*8 矩陣感測器(下邊亮燈)

```

/ERT_PIN = 0  HORZ_PIN = 512  SEL_PIN = 460
/ERT_PIN = 0  HORZ_PIN = 512  SEL_PIN = 440
/ERT_PIN = 0  HORZ_PIN = 512  SEL_PIN = 418
/ERT_PIN = 0  HORZ_PIN = 512  SEL_PIN = 467
/ERT_PIN = 0  HORZ_PIN = 512  SEL_PIN = 435
/ERT_PIN = 0  HORZ_PIN = 512  SEL_PIN = 494
/ERT_PIN = 0  HORZ_PIN = 512  SEL_PIN = 487
/ERT_PIN = 0  HORZ_PIN = 512  SEL_PIN = 456
/ERT_PIN = 0  HORZ_PIN = 512  SEL_PIN = 504
/ERT_PIN = 0  HORZ_PIN = 512  SEL_PIN = 536
/ERT_PIN = 0  HORZ_PIN = 512  SEL_PIN = 489

```

額外實作: 7 段顯示器的應用

➤ 程式碼

使用 LedControl 庫控制 LED 矩陣，透過類比輸入設定 LED 的亮度和位置，並根據輸入控制 LED 的亮滅。

```
#include <LedControl.h>
#define VERT_PIN A0
#define HORZ_PIN A1
#define SEL_PIN 3
LedControl lc = LedControl(11, 13, 10, 1);

void setup() {
    // put your setup code here, to run once:
    lc.shutdown(0, false); // 關閉省電模式
    lc.setIntensity(0, 8); // 設定亮度為 8 (介於 0~15 之間)
    lc.clearDisplay(0);    // 清除螢幕
    Serial.begin(9600);
    pinMode(VERT_PIN, INPUT);
    pinMode(HORZ_PIN, INPUT);
    pinMode(SEL_PIN, INPUT);
}
int randNumber_row = 4;
int randNumber_col = 4;
void loop() {
    int vert=analogRead(VERT_PIN);
    int horz=analogRead(HORZ_PIN);
    int sel=analogRead(SEL_PIN);
    Serial.print("VERT_PIN = ");

    Serial.print(vert);
    Serial.print("  HORZ_PIN = ");
    Serial.print(horz);
    Serial.print("  SEL_PIN = ");
    Serial.println(sel);
}
```

```

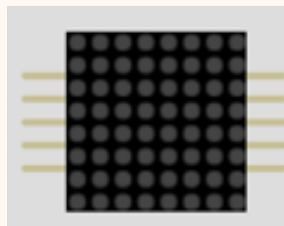
bool selPressed=digitalRead(SEL_PIN)==LOW;
// put your main code here, to run repeatedly:
// 將 Led 的列 1 欄 3 設定為亮
if(vert>600)
{lc.setLed(0, randNumber_row%8, randNumber_col%8, 0); // 將 Led 的欄,列設定
randNumber_row--;
}
else if(vert<300)
{
    lc.setLed(0, randNumber_row%8, randNumber_col%8, 0); // 將 Led 的欄,列設定為亮
    randNumber_row++;
}
if(horz>600)
{
    lc.setLed(0, randNumber_row%8, randNumber_col%8, 0); // 將 Led 的欄,列設定為亮
    randNumber_col++;
}
else if(horz<300)
{
    lc.setLed(0, randNumber_row%8, randNumber_col%8, 0); // 將 Led 的欄,列設定為亮
    randNumber_col--;
}
if(randNumber_row<0)
    randNumber_row=6;
if(randNumber_col<0)
    randNumber_col=6;

lc.setLed(0, randNumber_row%8, randNumber_col%8, 1); // 將 Led 的欄,列設定為亮}

```

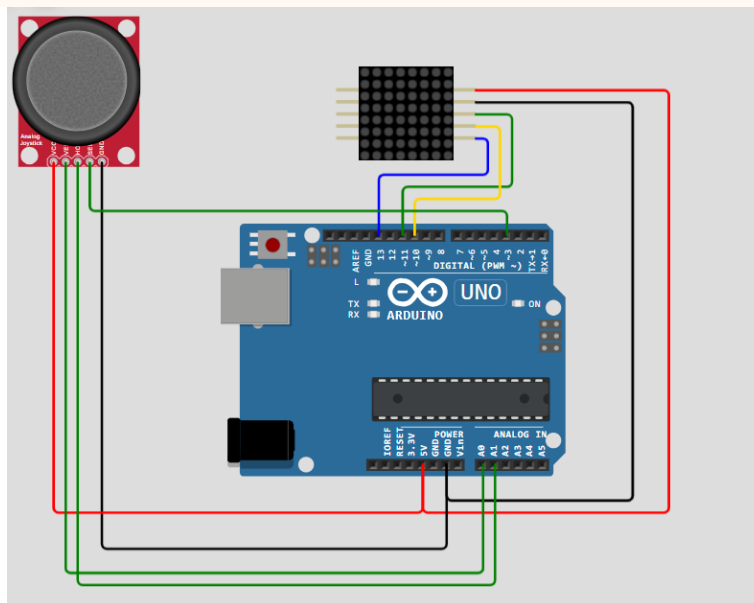
➤ 使用元件(8*8 矩陣感測器)

1. 8*8 矩陣感測器是一種具有 64 個感測元件的裝置，可用於檢測和測量各種物理量，如溫度、光照度或壓力等。它通常以行和列的形式排列，能夠同時探測多個位置或條件變化，廣泛應用於科學研究和工業檢測領域。



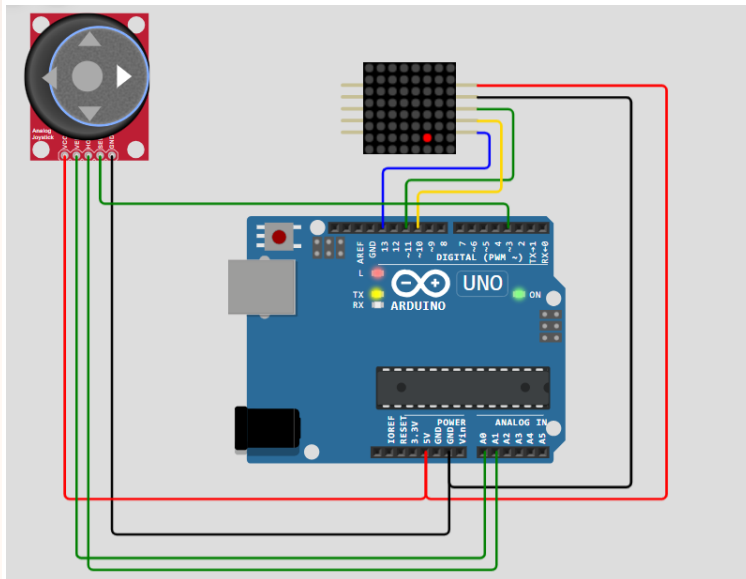
➤ 接線電路

1. 將 8*8 矩陣感測器與 Arduino 做連接，8*8 矩陣感測器接 10 接腳，+ 接 vcc - 接 GND



➤ 實驗結果

1. 模擬控制搖桿為主題的 LED 矩陣顯示。程式中使用 Arduino 來讀取模擬搖桿的輸入，並根據輸入的值控制 LED 的亮滅狀態，展示出基本的搖桿控制效果。



實驗心得:

我學到了如何利用不同類型的感測器（如搖桿、LED 矩陣、七段顯示器等）來實現各種電子裝置的功能和應用。從讀取模擬和數位輸入的基本原理開始，到如何通過 Arduino 控制和操作這些感測器，這些實驗不僅提供了技術上的學習，還深入探討了感測器在不同應用場景中的實際應用。

通過搖桿感測器的實驗，了解了如何讀取搖桿的模擬輸出值和數位按鈕的狀態，LED 矩陣的實驗展示了如何使用 LedControl 庫來控制和操作 8x8 LED 矩陣

七段顯示器的實驗則專注於如何將數字或字元顯示在七段顯示器上。透過讀取來自曲率感測器的數據，利用 Arduino 控制七段顯示器顯示相應的數字，這顯示了如何將感測器輸出的物理量轉換為易於理解和顯示的形式