

Lab2 Report

組員：0116327 鍾興璇 0116322 邱一豐 第 16 組

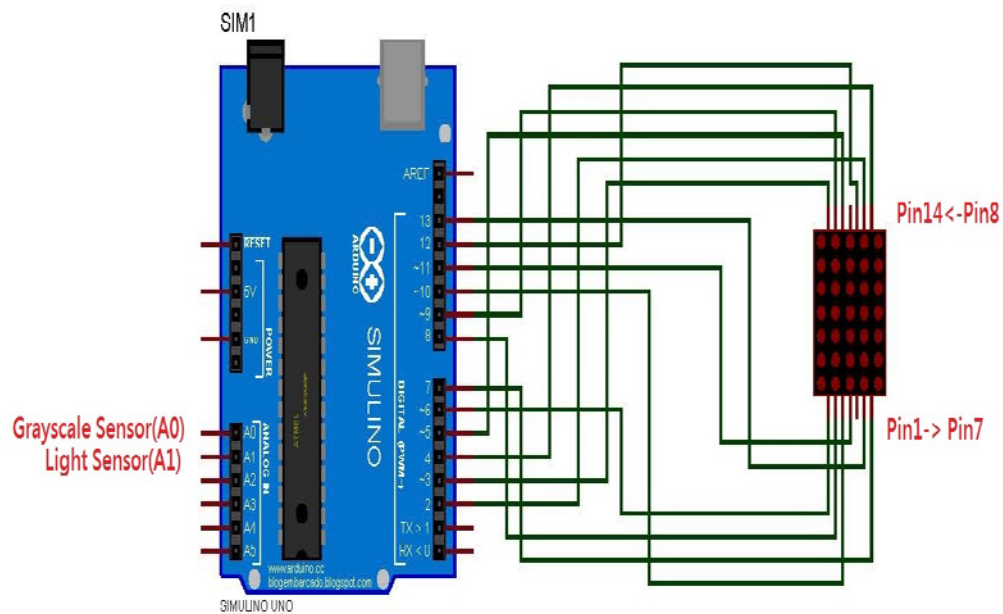
1. Lab Name: Arduino grayscale/light

2. Demo:

將 Arduino 分別接上灰度(L)和光線(R)的感應器，比較兩個感應器的值，如果灰度的值比光線的值大，就在請在 LED 陣列印出 L，相反的如果光線比灰度大，LED 陣列就印 R。

a. 實驗電路圖

電路圖



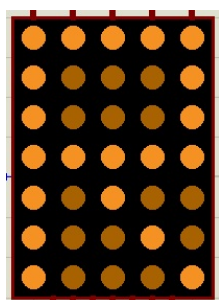
| LED 與 Arduino pin 對應 | |
|------------------------|---------------------|
| 5x7 LED Dot Matrix pin | Arduino digital pin |
| 1 | 6 |
| 2 | 8 |
| 3 | 9 |
| 4 | 10(擇一) |
| 5 | x |
| 6 | 12 |
| 7 | 7 |
| 8 | 4 |
| 9 | 2 |
| 10 | 11 |
| 11 | 10(擇一) |
| 12 | 5 |
| 13 | 8 |
| 14 | 3 |

(Arduino 上的 pin10 可選擇接到 5x7 LE 上的 pin4 or pin10 上)

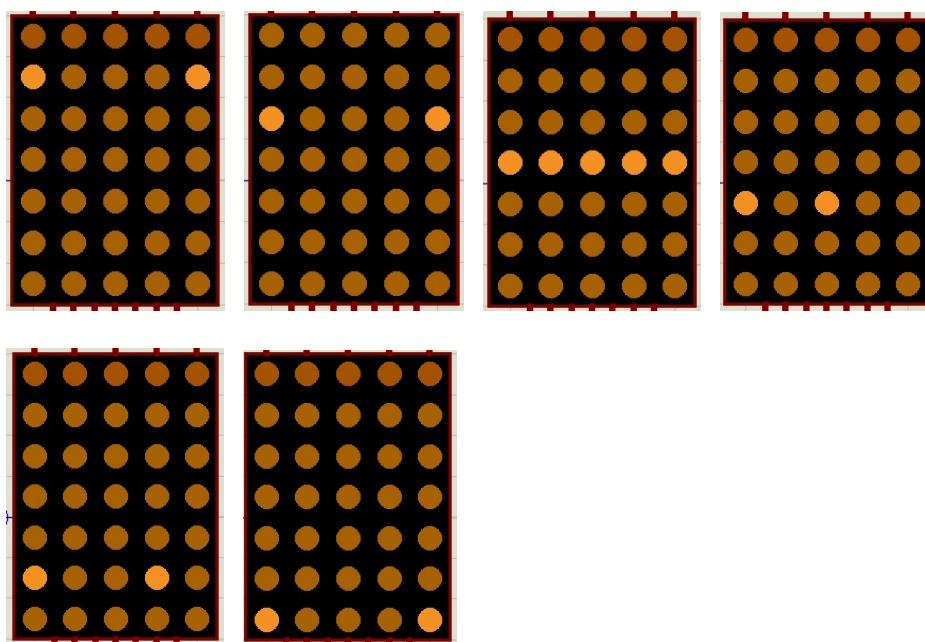
b. 請大概解釋 code(demo)的流程以及用意

首先由 lab2.1 及 lab2.2 中我們學會如何由灰度感測器及光線感測器讀入類比的數值，並將數值顯示在終端機上面。在 DEMO 的程式中我們只需要將讀入的值先暫存，再依據兩個值的大小關係判斷要在 LED 上顯示 L 或 R 即可。在程式中我們使用 val_L 來存 light sensor 所讀到的數值，而 val_R 來存 grayscale sensor 所讀到的數值，之後再比較兩者數值來決定 LED 該顯示的字母(L or R)。如果 val_L 小於 val_R 則表示亮度值小於灰度值，LED 顯示 L (程式中呼叫 showlow())，如果 val_L 大於 val_R 則表示亮度值大於灰度值，LED 顯示 R(程式中呼叫 showhigh())。在函式 showlow() 和 showhigh() 中，我們使用一系列一列(row by row)的掃描方式在 LED 陣列上顯示字母。

下圖以顯示字母 R 為例



(圖 1)肉眼所看到



以此方式循環搭配快速執行會使肉眼看起來如(圖 1)

程式碼

```
1 //pin 2~13 for the LED matrix7X5
2 //A0 A1 for the sensor (grayscale , light) to read data
3 const byte ROW = 7; //PIN:2~8 for row:9,14,8,12,1,7,2
4 const byte COL = 5; //PIN:9~13 for column:13,3,4,10,6
5
6 int arr_L[ROW][COL] = { //character L
7     {1,1,0,0,0 },
8     {1,1,0,0,0 },
9     {1,1,0,0,0 },
10    {1,1,0,0,0 },
11    {1,1,0,0,0 },
12    {1,1,1,1,1 },
13    {1,1,1,1,1 }
14 };
15
16 int arr_R[ROW][COL] = { //character R
17     {1,1,1,1,1 },
18     {1,0,0,0,1 },
19     {1,0,0,0,1 },
20     {1,1,1,1,1 },
21     {1,0,1,0,0 },
22     {1,0,0,1,0 },
23     {1,0,0,0,1 }
24 };
25
26 void showlow(){
27
28     for( int i = 0 ; i < ROW ; i++ ){
29
30         //to turn off all rows , clear the selection
31         for(int j = 0 ; j < ROW ; j++){
32             digitalWrite( j+2 , 1);
33         }
34         //select the one row
35         digitalWrite( i+2 , 0 ); //select the row
36
37         //write the data to row
38         for(int j = 0 ; j < COL ; j++){
39             digitalWrite( 9+j , arr_L[i][j] );
40         }
41
42         delay(1);
43     }
44 }
45
46
```

```

47
48 void showhigh(){
49
50     for( int i = 0 ; i < ROW ; i++ ){
51
52         //to turn off all rows , clear the selection
53         for(int j = 0 ; j < ROW ; j++){
54             digitalWrite( j+2 , 1);
55         }
56
57         //select the one row
58         digitalWrite( i+2 , 0 );
59
60         //write the data to row
61         for(int j = 0 ; j < COL ; j++){
62             digitalWrite( 9+j , arr_R[i][j] );
63         }
64
65         delay(1);
66     }
67 }
68
69

```

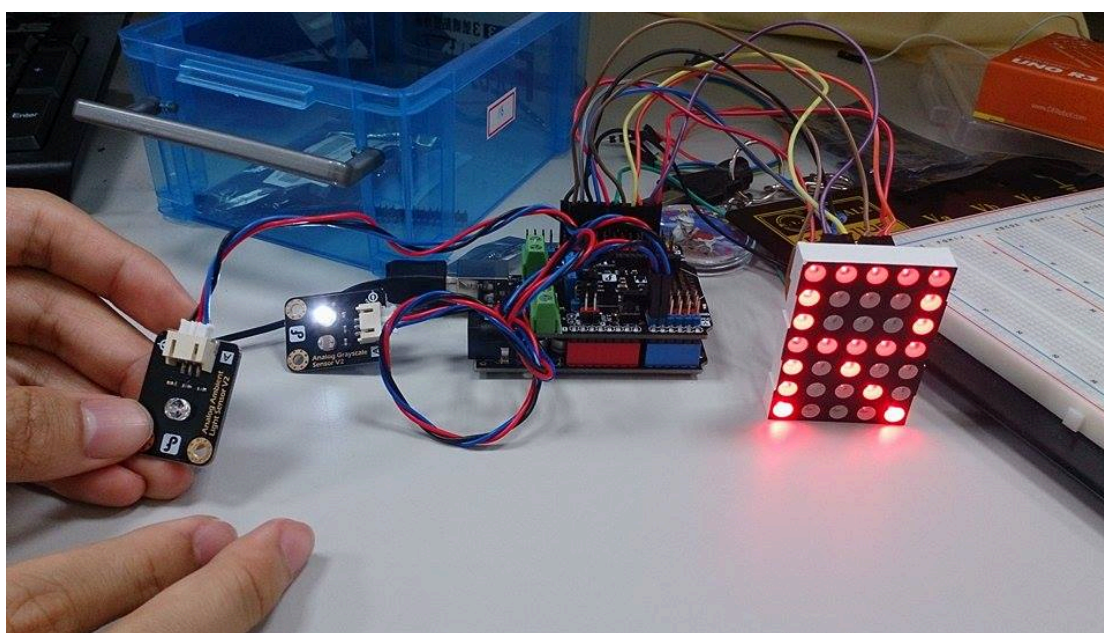
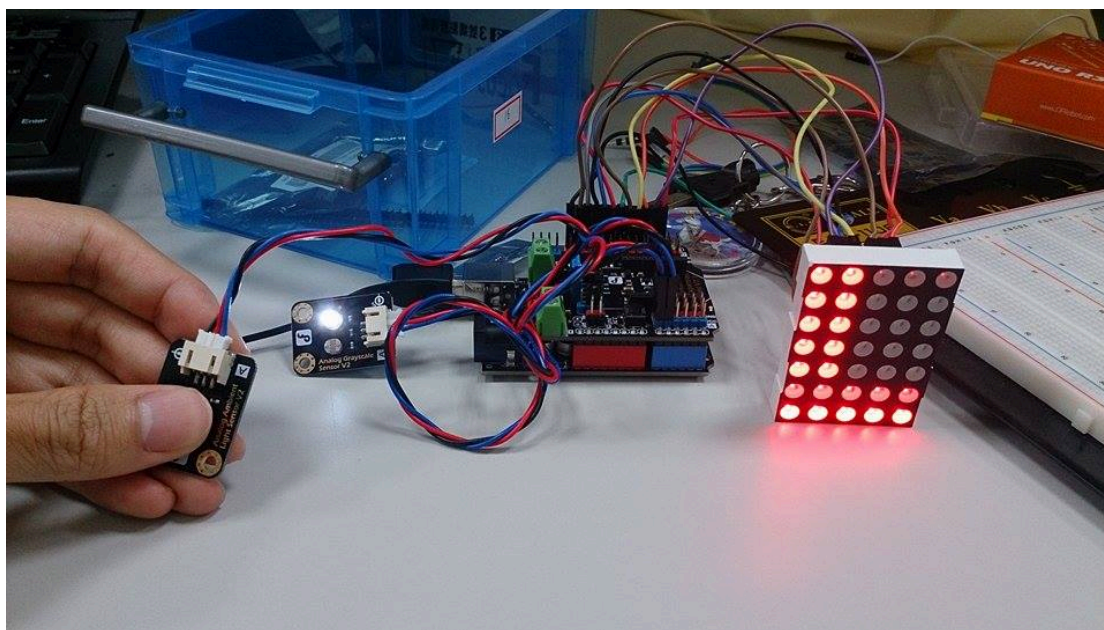
```

71
72 void clear(){
73
74     //turn off the row
75     for(int i = 2 ; i <= 8 ; i++){
76         digitalWrite( i , 1);
77     }
78
79     //turn off the column
80     for( int i = 9 ; i <= 13 ; i++){
81         digitalWrite( i , 0);
82     }
83
84 }
85
86 void setup(){
87
88     //set the pin mode
89     for( int i = 2 ; i <= 13 ; i++){
90         pinMode( i , OUTPUT );
91     }
92
93     clear(); //clear the display
94     Serial.begin(9600); //set the baud rate for the terminal
95
96 }
97

```

```
98
99 void loop(){
100     int val_L; //the variable store the value of grayscale sensor
101     int val_R; //the variable store the value of light sensor
102
103     val_R = analogRead(0); // read the grayscale sensor
104     val_L = analogRead(1); //read the light sensor
105
106
107     //show the sensor information on the screen
108     Serial.print("L:");
109     Serial.print(val_L,DEC);
110     Serial.print("R:");
111     Serial.println(val_R,DEC);
112
113     if(val_R > val_L){
114         showlow(); //show the L
115     }else{
116         showhigh(); //show the R
117     }
118
119 }
```


實驗圖片

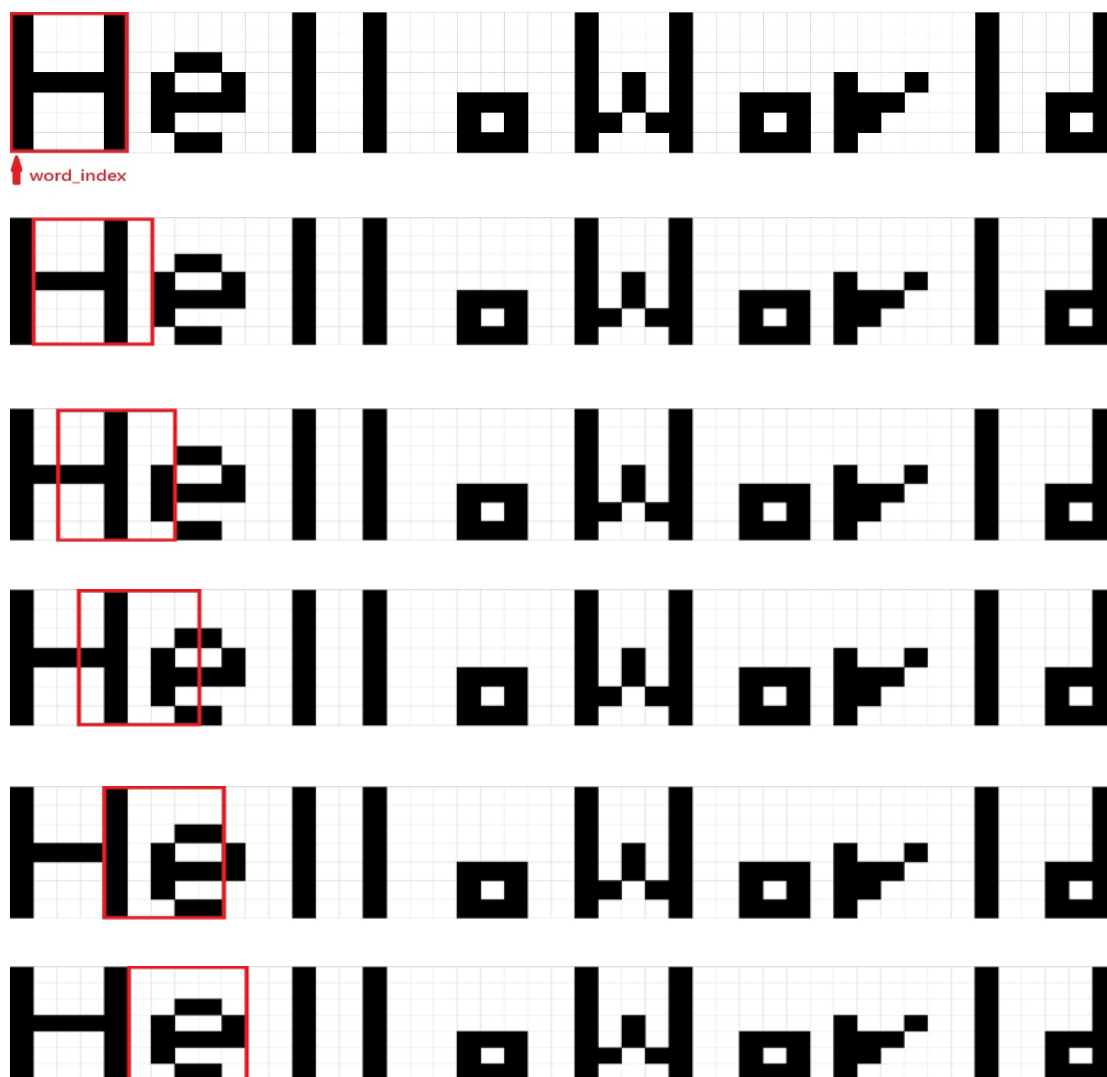


3. Bonus 部分

a. 說明

與上一題一樣使用掃描的方式來顯示字元，透過固定掃描同一個區域(在這裡稱為視窗格)多次來增加視覺暫留，使 LED 陣列看起來有移動的效果，為了要有動畫的感覺，在 bonus 的顯示部份我們採用與前一題不同的點在於，在這裡使用一行一行的方式顯示 (column by column)並在給定的範圍內重複多次的顯示。

以下為將字串左移 (掃描窗格右移) 圖為例



程式中我們使用 speed variable 來控制在該視窗格內該重複掃描的次數，重複執行 speed 次後就會將掃描窗格移動，經過我們的實驗，將 speed 設為 20 是較為適合的，不會移動太快也不會移動太慢，另外使用 word_index variable 來表示視窗格的起始位置左移或右移時只需要改變 word_index 的執來達到目的。

程式碼

```
1 //pin 2~13 for the LED matrix7X5
2 //A0 A1 for the sensor (grayscale , light) to read data
3 const byte ROW = 7; //PIN:2~8 for row
4 const byte COL = 5; //PIN:9~13 for column
5 int word_index; //the current word index
6 int arr_String[50][7] = {
7     //row1 -> row8
8     //H
9     {1,1,1,1,1,1,1 },
10    {0,0,0,1,0,0,0 },
11    {0,0,0,1,0,0,0 },
12    {1,1,1,1,1,1,1 },
13    {0,0,0,0,0,0,0 },
14    //e
15    {0,0,0,1,1,1,0 },
16    {0,0,1,0,1,0,1 },
17    {0,0,1,0,1,0,1 },
18    {0,0,0,1,1,0,0 },
19    {0,0,0,0,0,0,0 },
20
21    //I l
22    {0,0,0,0,0,0,0 },
23    {1,1,1,1,1,1,1 },
24    {0,0,0,0,0,0,0 },
25    {1,1,1,1,1,1,1 },
26    {0,0,0,0,0,0,0 },
27
28    //o
29    {0,0,0,0,0,0,0 },
30    {0,0,0,1,1,1,1 },
31    {0,0,0,1,0,0,1 },
32    {0,0,0,1,1,1,1 },
33    {0,0,0,0,0,0,0 },
34
35    //space
36    {0,0,0,0,0,0,0 },
37    {0,0,0,0,0,0,0 },
38    {0,0,0,0,0,0,0 },
39
40    //W
41    {1,1,1,1,1,1,1 },
42    {0,0,0,0,0,1,0 },
43    {0,0,0,1,1,0,0 },
44    {0,0,0,0,0,1,0 },
45    {1,1,1,1,1,1,1 },
46
47
48    //o
49    {0,0,0,0,0,0,0 },
50    {0,0,0,0,0,0,0 },
51    {0,0,0,1,1,1,1 },
52    {0,0,0,1,0,0,1 },
53    {0,0,0,1,1,1,1 },
54    {0,0,0,0,0,0,0 },
55
```

```

56 //r
57 {0,0,1,1,1,1,1 },
58 {0,0,0,1,1,0,0 },
59 {0,0,0,1,0,0,0 },
60 {0,0,1,0,0,0,0 },
61 {0,0,0,0,0,0,0 },
62
63 //l d
64 {1,1,1,1,1,1,1 },
65 {0,0,0,0,0,0,0 },
66 {0,0,0,1,1,1,1 },
67 {0,0,0,1,0,0,1 },
68 {1,1,1,1,1,1,1 },
69 {0,0,0,0,0,0,0 },
70 {0,0,0,0,0,0,0 },
71 {0,0,0,0,0,0,0 },
72 {0,0,0,0,0,0,0 },
73 {0,0,0,0,0,0,0 },
74 {0,0,0,0,0,0,0 }
75 };
76 //write a column data (7bit) toLED matrix column
77 void write_7bit_data( int pinstart , int arr_7bit[7] ){
78     for(int i = 0 ; i < 7 ; i++){
79         digitalWrite( i+pinstart , !arr_7bit[i] ); //need to inverse
80     }
81 }
82

```

```

84 //select a column
85 void write_5bit_sel(int pinstart , int pin ){
86     //clear
87     for(int i = 0 ; i < 5 ; i++){
88         digitalWrite( pinstart+i , 0);
89     }
90     digitalWrite(pin , 1 );
91 }
92
93
94 void setup(){
95     for( int i = 2; i <= 13 ; i++){
96         pinMode( i , OUTPUT );
97     }
98     Serial.begin(9600);
99 }
100

```

```

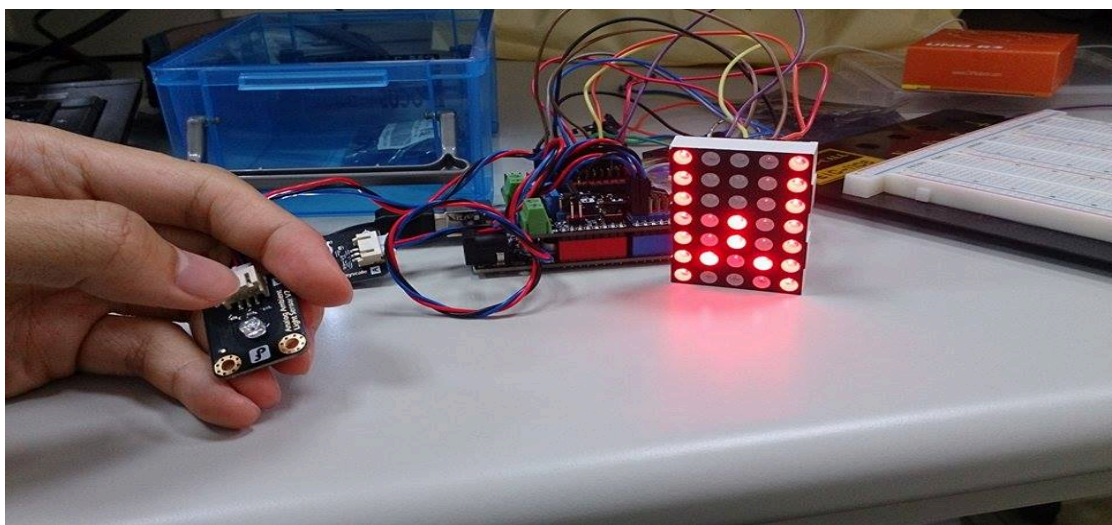
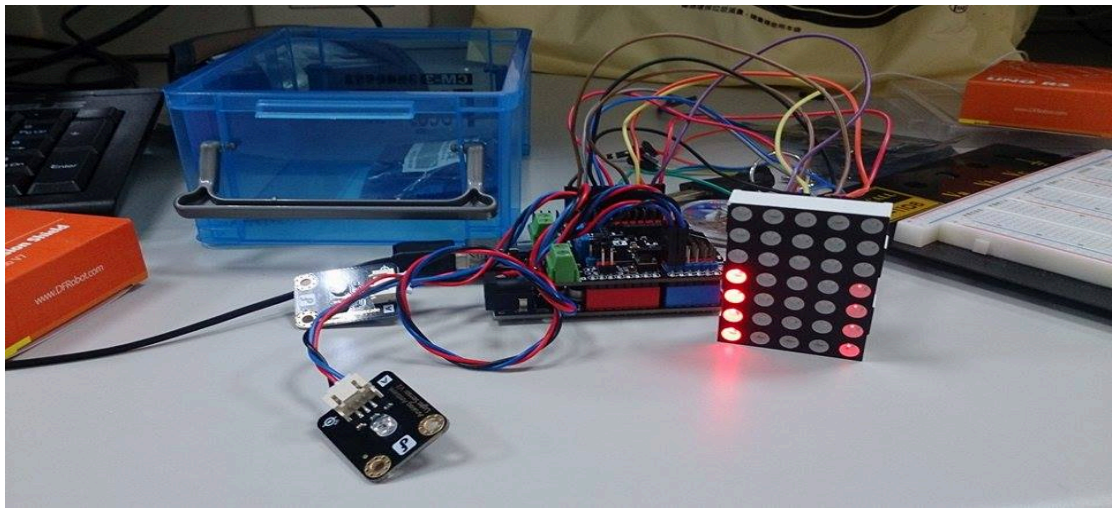
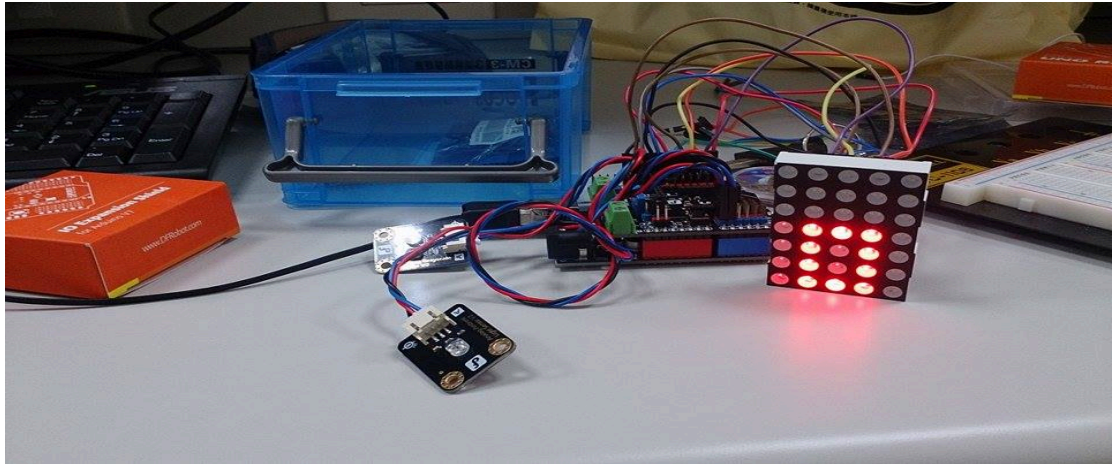
101 void loop(){
102     int val_L;
103     int val_R;
104
105
106     int speed = 20;
107     while(1){
108
109         val_L = analogRead(0);
110         val_R = analogRead(1);
111         Serial.print("L:");
112         Serial.print(val_L,DEC);
113         Serial.print("R:");
114         Serial.println(val_R,DEC);
115         if( val_L > val_R ){
116             word_index--;
117             if( word_index < 0 ){
118                 word_index = 45;
119             }
120
121         }else{
122             word_index++;
123             if( word_index > 45 ){
124                 word_index = 0;
125             }
126         }
127     }
128
129     //display the window content
130     for(int k = 0 ; k < speed ; k++){
131         for(int j = 4 ; j >= 0 ; j--){
132             write_5bit_sel( 9 , 9+j );
133             write_7bit_data( 2 ,arr_String[word_index+j] );
134             delay(1);
135         }
136     }
137 }

```

b. 如果程式有使用到 library，請大概解釋不使用 library 的寫法，相反的如果沒有使用到 library，請大概解釋使用 library 的寫法。

在這個部份我們沒有使用 FrequencyTimer2 library 的寫法，不過我們有去研究該寫法，在 FrequencyTimer2 library 中有 setPeriod 函式可以設定一段時間過後去執行 interrupt routine，會由 setOnOverflow() 這個函數其參數為 function pointer，並且該函數會去執行該 pointer 所指到的 function，因此只要將顯示的函數寫成 setOnOverflow() 的參數載入，就可以呈現出移動字母感覺，與我們不使用 library 相比而言可以減少不必寫重複掃描窗格的次數，因為該數值會由 setPeriod 函數達成。

實驗圖片



3. 本次實驗遇到的問題

- (1)一開始沒將 delay 的數值調整好，導致看到的結果與預期的不符
Sol:經過幾次調整 delay 的參數後終於成功顯示出 L 和 R。
- (2)程式結構問題，未考慮到讀取 sensor 的時機以及改變 word_index 的時機。
Sol:將結構改為先讀取 sensor 數值後，再做左移或右移的判斷，最後再做一組掃描視窗的顯示，如此反覆。
- (3)Bonus 部分沒將 array 中儲存的資料放對，導致顯示出來的字母左右相反
Sol:修改 array 中 data 擺放的位置即可。

4. 本次實驗心得

在這次的實驗中我們學會使用灰度感測器以及光線感測器，也學會在 Arduino 中使用陣列的技巧搭配到 LED matrix 上，在 bonus 的部分更學會了如何使用 FrequencyTimer2 library 中的 timer 來實作出 interrupt。這次的 bonus 其實在日常生活中是很常見的，在街道上經常可以看到跑馬燈出現在看板上，由此可見該應用的廣泛程度。經過這次的實驗後相信我們也可以做出自己想要的跑馬燈。