

第1組

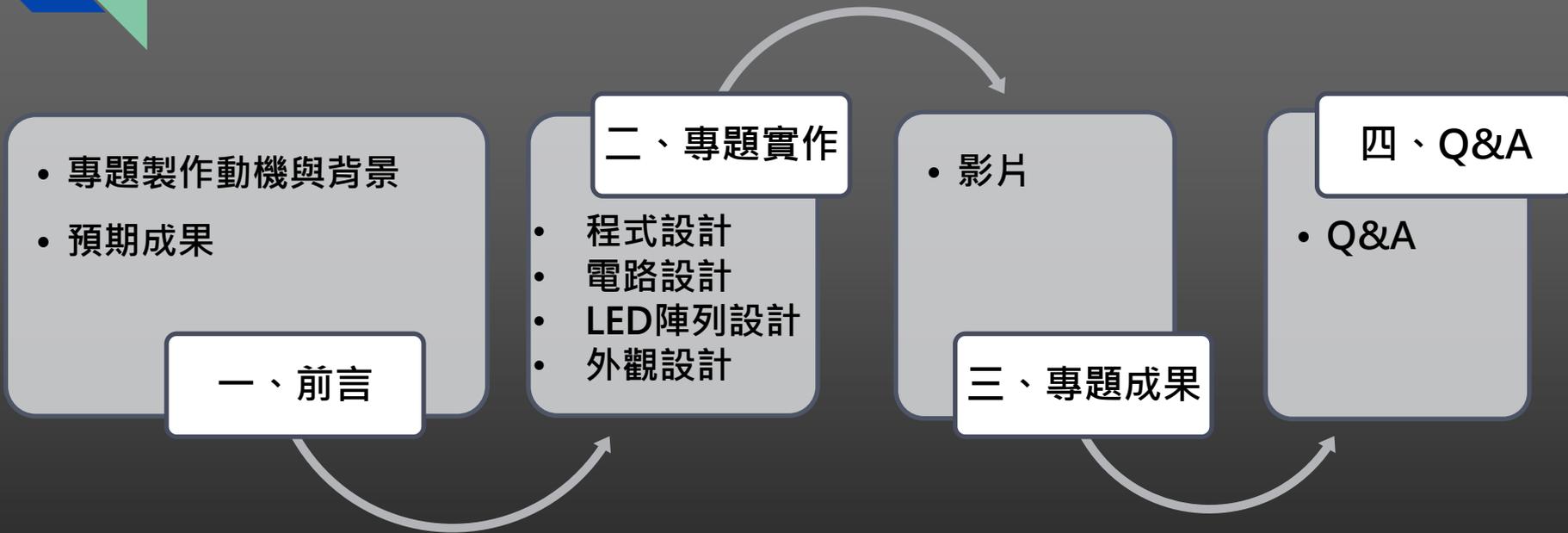
5x5x5立體貪食蛇小遊戲

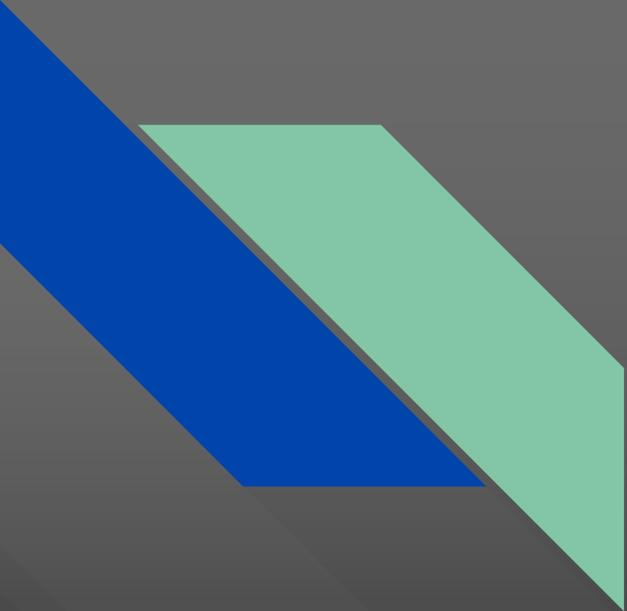
期末專題報告

組長: 李松濤
組員: 張智翔
組員: 趙宇晨
組員: 蔡佳融

指導老師: 黃建中 老師

大綱





一、前言

1. 組員介紹、合照
2. 專題製作背景及目的
3. 預期成果
4. 組員貢獻度

組員合照



前言

專題實
作

專題成
果

問題與
討論



專題製作背景及目的

幾年來科技發展迅速，隨著智慧型手機成為主流，各式各樣的手遊也推陳出新，而在提起經典的手機遊戲時，貪食蛇一定會出現在大家的腦海中

所以我們決定把貪食蛇，結合我們三年所學，製作出5x5x5立體貪食蛇，讓大家在回味經典的同時，也有創新的玩法

前言

專題實
作

專題成
果

問題與
討論



預期成果

- 外觀精美
- 可在立體矩陣中遊玩貪食蛇
- 可由USB Type-B供電
- 可使用LCD顯示分數等資訊

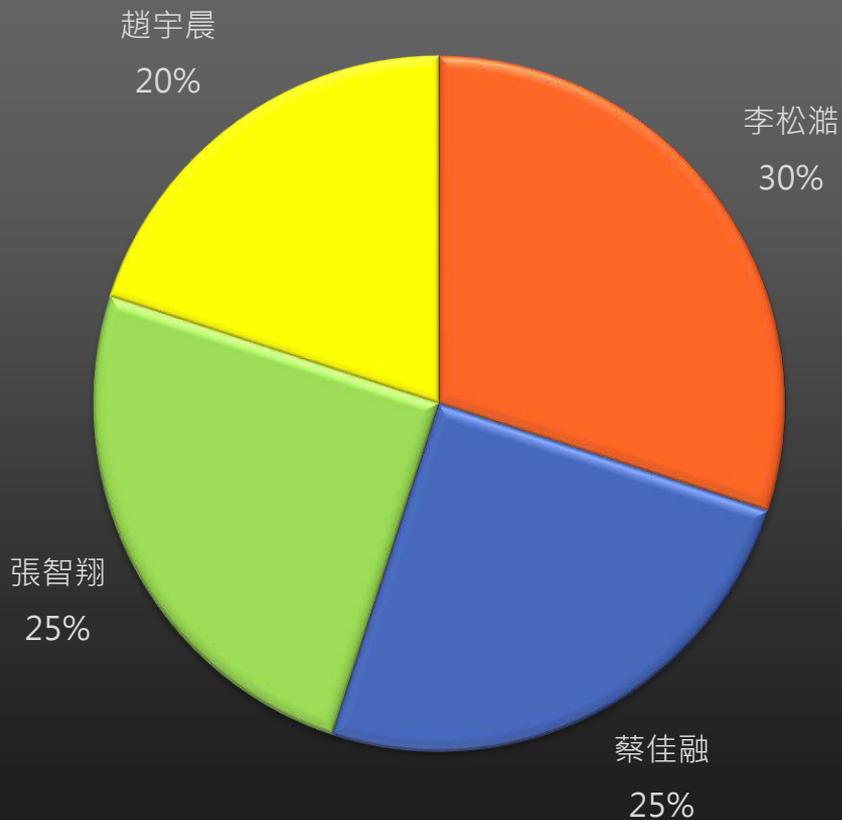
前言

專題實
作

專題成
果

問題與
討論

組員貢獻度

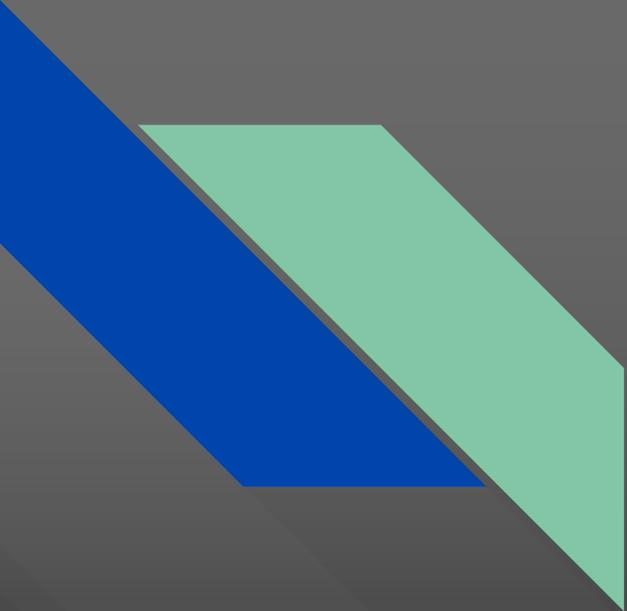


前言

專題實
作

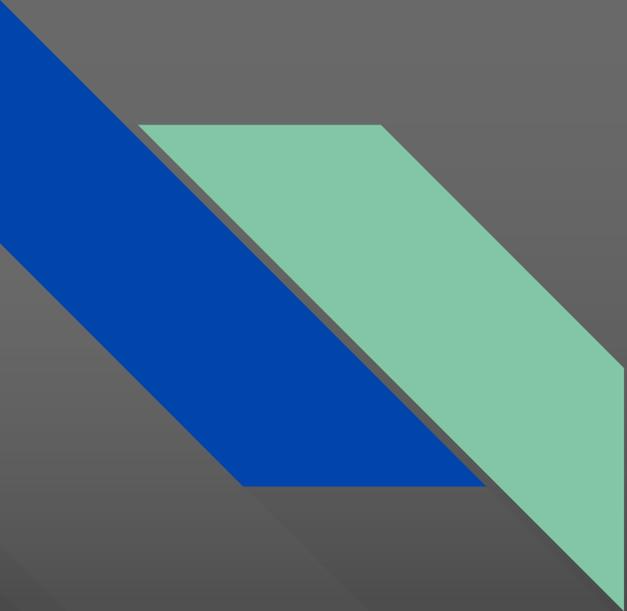
專題成
果

問題與
討論



一、專題實作

1. LED陣列設計
2. 電路設計
3. 程式設計
4. 外觀設計

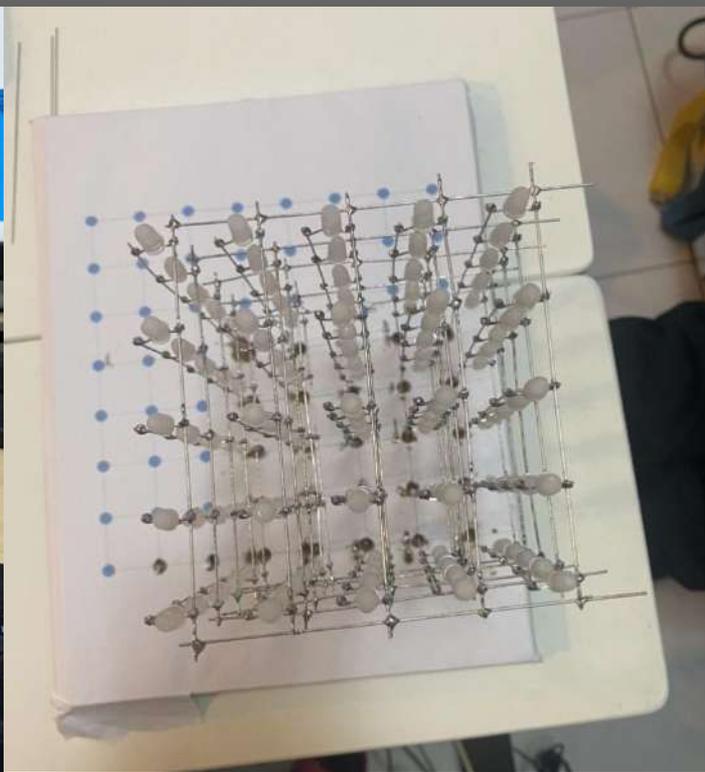
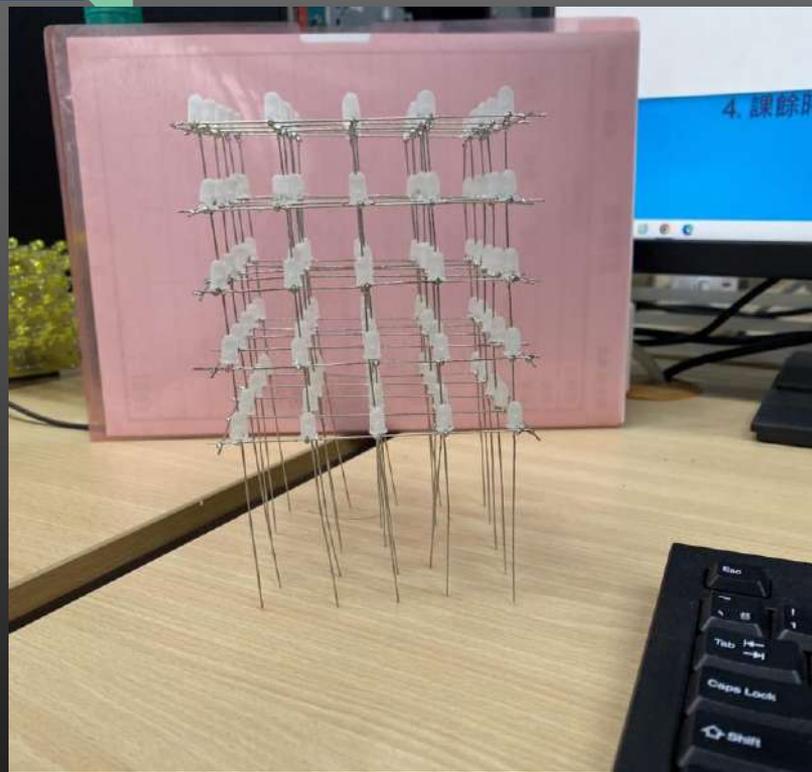


1. LED矩陣設計

1. 成品圖
2. 製作模具
3. LED處理
4. 組裝

負責組員:張智翔

成品圖



前言

專題實
作

專題成
果

問題與
討論

材料介紹

- 製作立體點矩陣的模具



前言

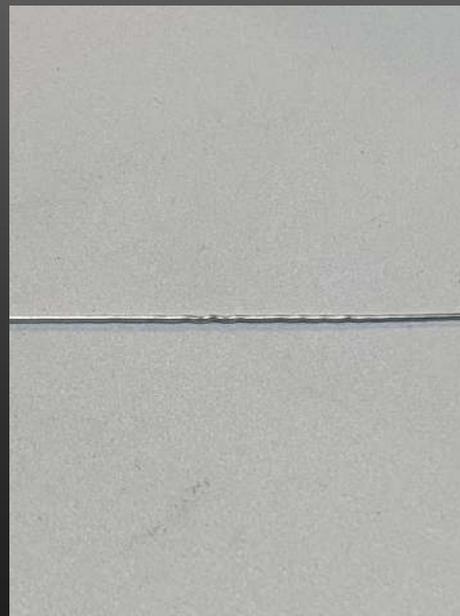
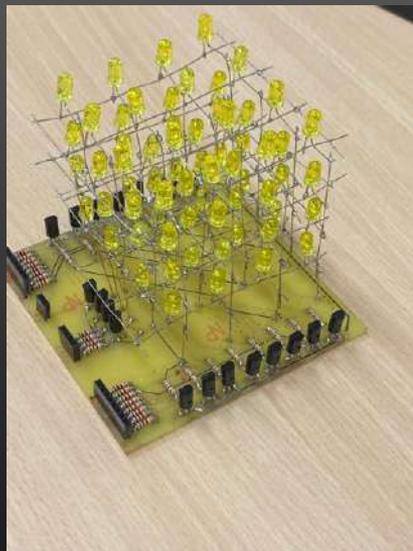
專題實
作

專題成
果

問題與
討論

材質較硬的鐵絲

- 為了讓鐵絲的支撐力變得更好以至於不會變形



前言

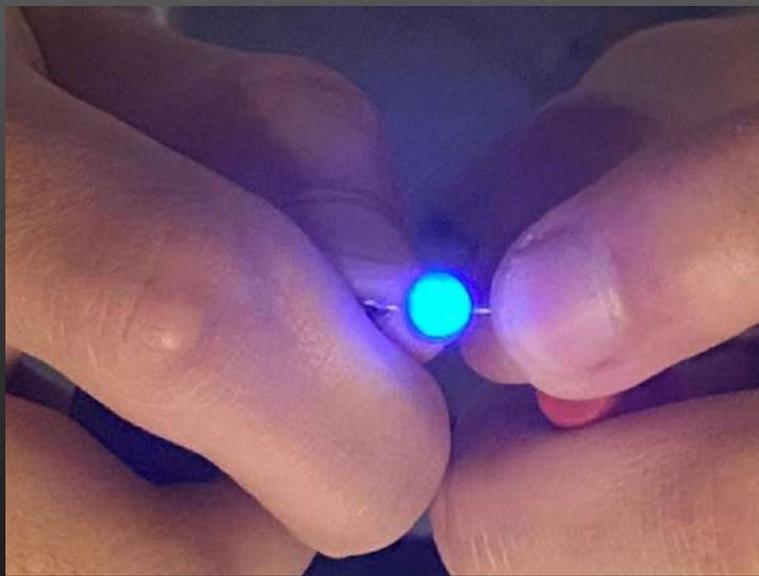
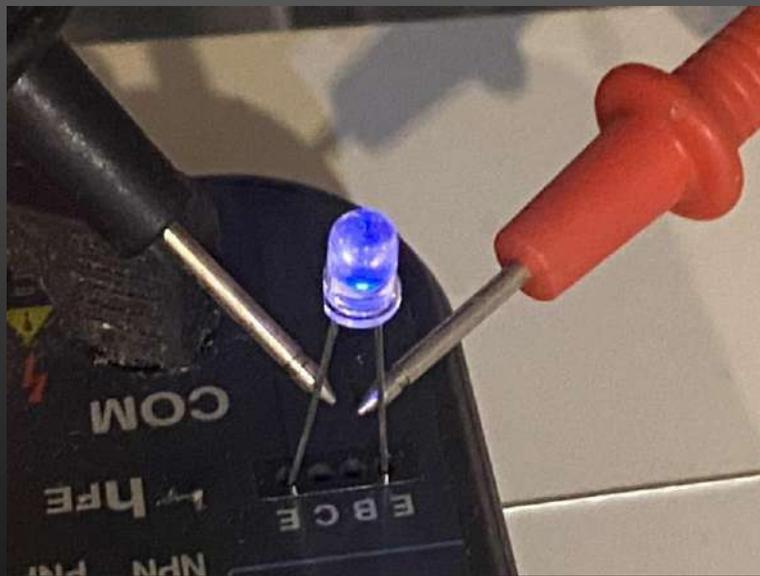
專題實
作

專題成
果

問題與
討論

LED處理

- 為了讓他可以顯示得更清楚，使用砂紙把表面磨霧，以免光線穿透



前言

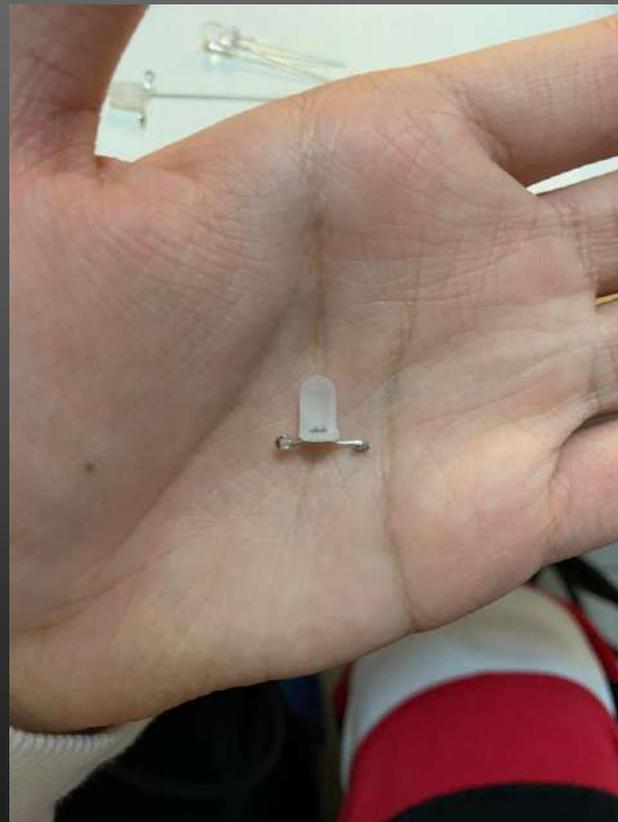
專題實
作

專題成
果

問題與
討論

LED接腳處理過程

- 為了讓焊點固定



前言

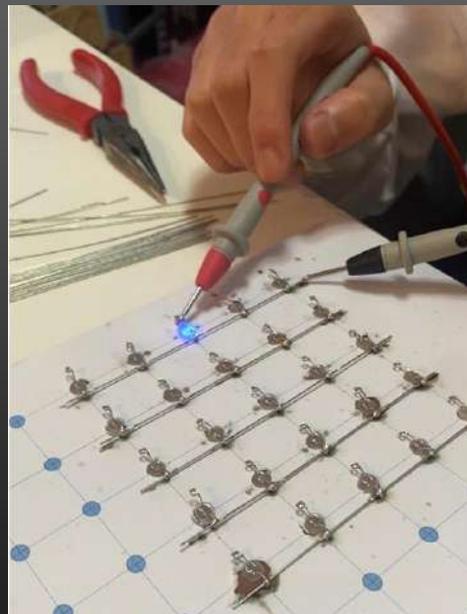
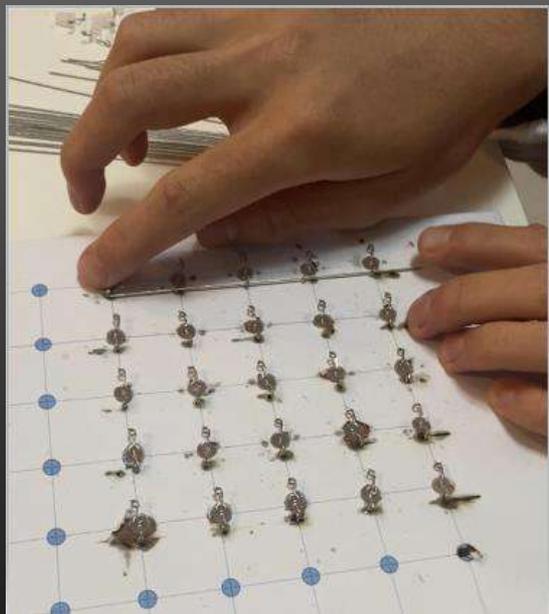
專題實
作

專題成
果

問題與
討論

每層的LED矩陣

- 每一橫面LED陰極接再一起



前言

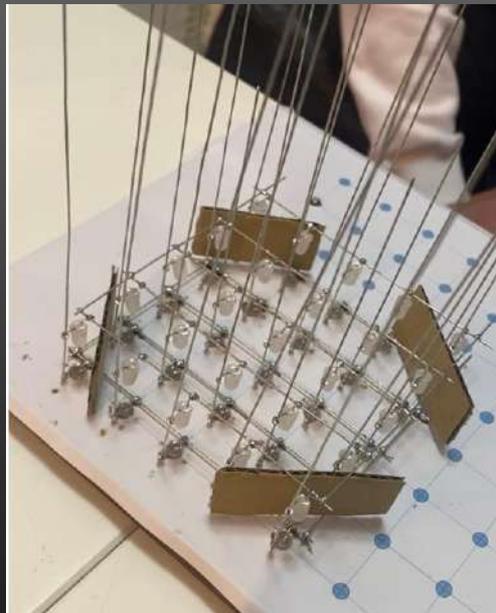
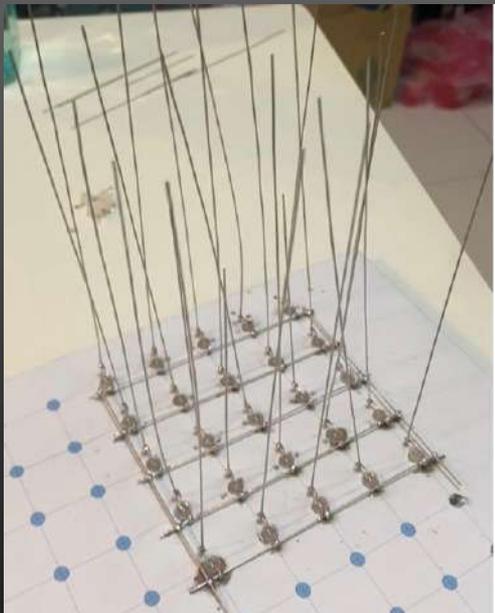
專題實
作

專題成
果

問題與
討論

LED矩陣組裝

- 把每一直條的陽極接再一起

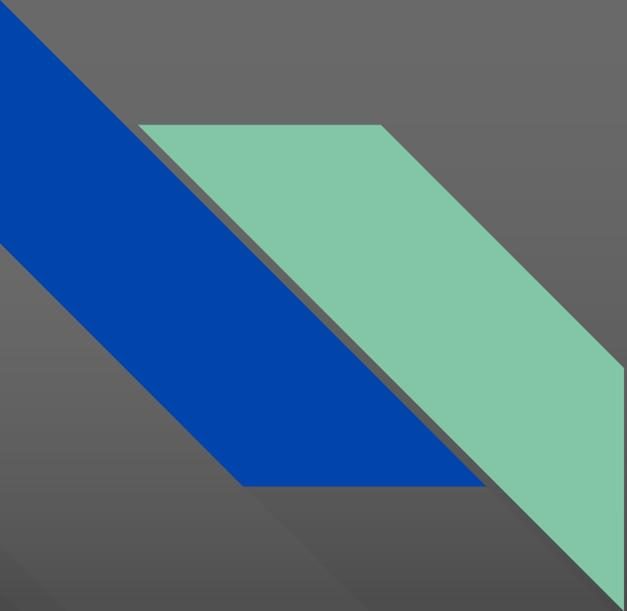


前言

專題實
作

專題成
果

問題與
討論



2. 電路設計

1. 元件介紹
2. 電路設計圖
3. 電路布局
4. 成品圖

負責組員:蔡佳融

Altium Designer

- Altium Designer(簡稱AD)
- 由澳大利亞的軟體公司Altium Ltd.開發
- 用於電路原理圖、PCB、FPGA設計
- 在高速電路板設計領域，它可進行差分對布線



前言

專題實
作

專題成
果

問題與
討論

2SC1384

- NPN BJT
- Sink邏輯
- 使用5顆

Transistor

Panasonic

2SC1383, 2SC1384

Silicon NPN epitaxial planer type

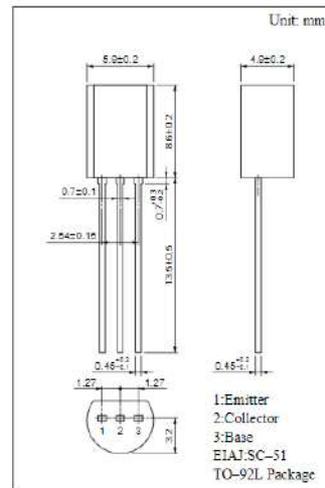
For low-frequency power amplification and driver amplification
Complementary to 2SA683 and 2SA684

■ Features

- Low collector to emitter saturation voltage $V_{CE(sat)}$.
- Complementary pair with 2SA683 and 2SA684.

■ Absolute Maximum Ratings (Ta=25°C)

Parameter	Symbol	Rated	Unit
Collector to base voltage	V_{CBO}	30	V
2SC1383		60	
Collector to emitter voltage	V_{CEO}	25	V
2SC1384		50	
Emitter to base voltage	V_{EBO}	5	V
Peak collector current	I_{CP}	1.5	A
Collector current	I_C	1	A
Collector power dissipation	P_C	1	W
Junction temperature	T_j	150	°C
Storage temperature	T_{stg}	-55 ~ +150	°C



前言

專題實作

專題成果

問題與討論

電路介紹

- 5x5x5矩陣

		cubic			
25	5-5				
24	5-4				
23	5-3				
22	5-2				
21	5-1				
20	4-5				
19	4-4				
18	4-3				
17	4-2	COM 5		30	
16	4-1				
15	3-5	COM 4		29	
14	3-4				
13	3-3	COM 3		28	
12	3-2				
11	3-1	COM 2		27	
10	2-5				
9	2-4	COM 1		26	
8	2-3				
7	2-2				
6	2-1				
5	1-5				
4	1-4				
3	1-3				
2	1-2				
1	1-1				

5*5*5_LED

前言

專題實
作

專題成
果

問題與
討論

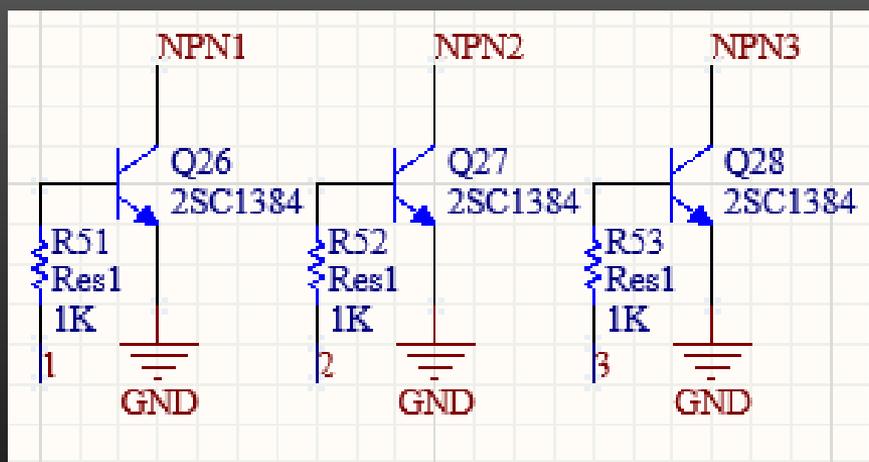
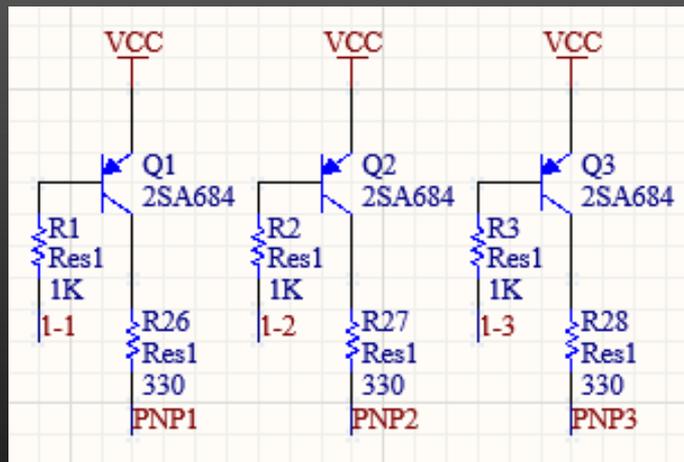
驅動電路

前言

專題實
作

專題成
果

問題與
討論



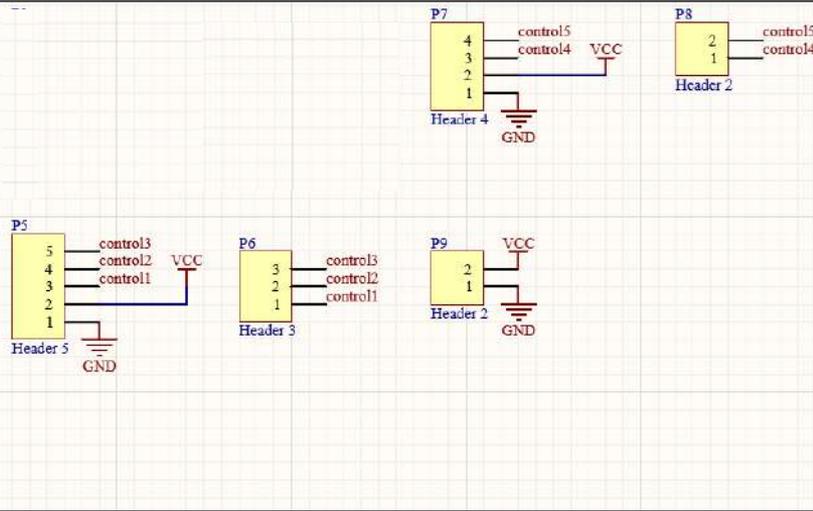
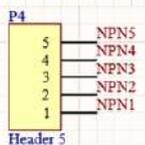
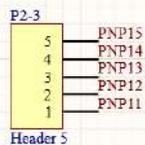
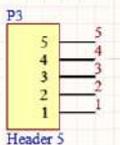
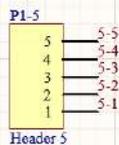
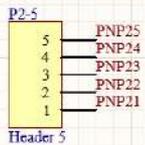
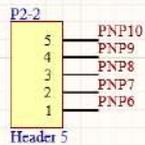
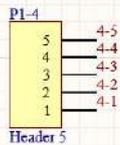
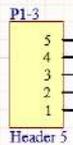
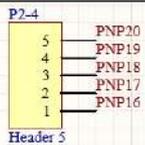
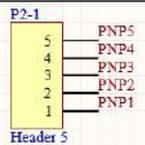
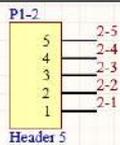
前言

資料線和控制線

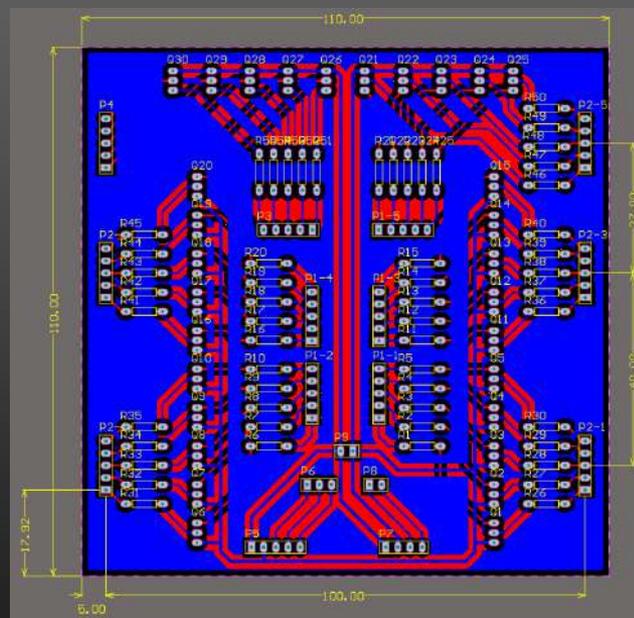
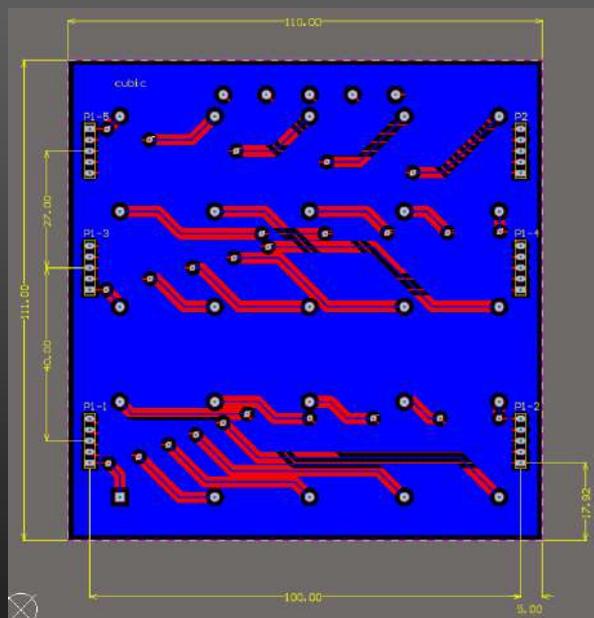
專題實作

專題成果

問題與討論



電路板佈局



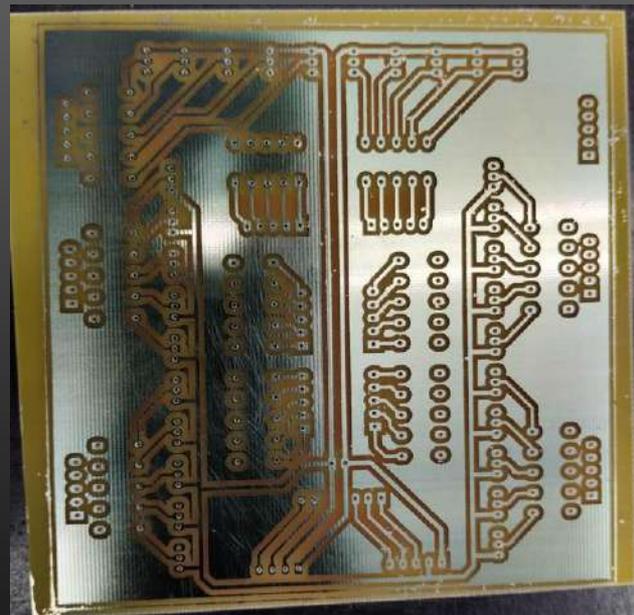
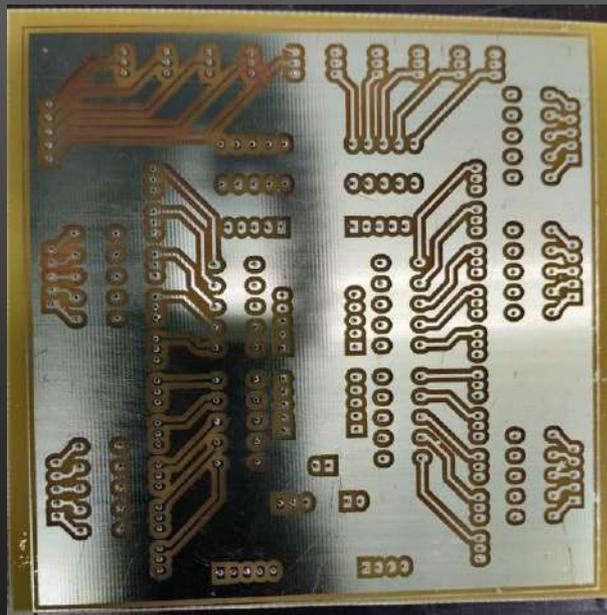
前言

專題實
作

專題成
果

問題與
討論

成品



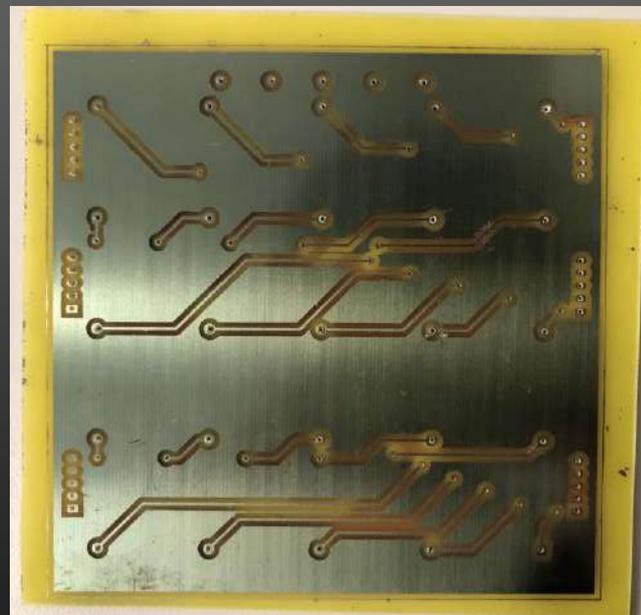
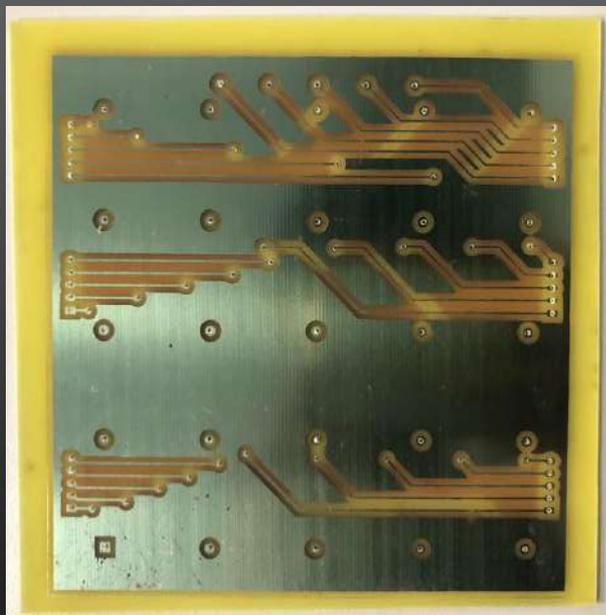
前言

專題實
作

專題成
果

問題與
討論

成品



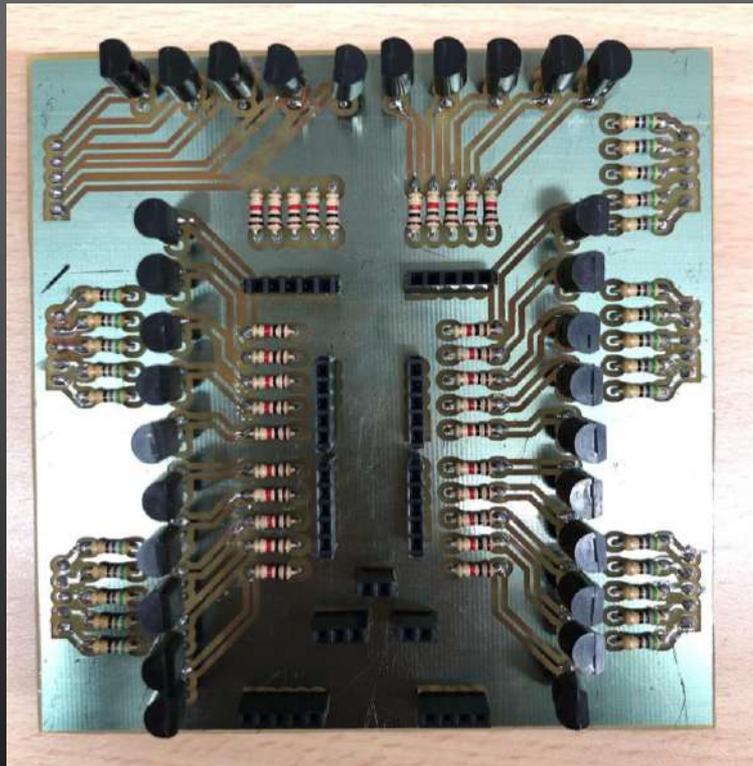
前言

專題實
作

專題成
果

問題與
討論

成品

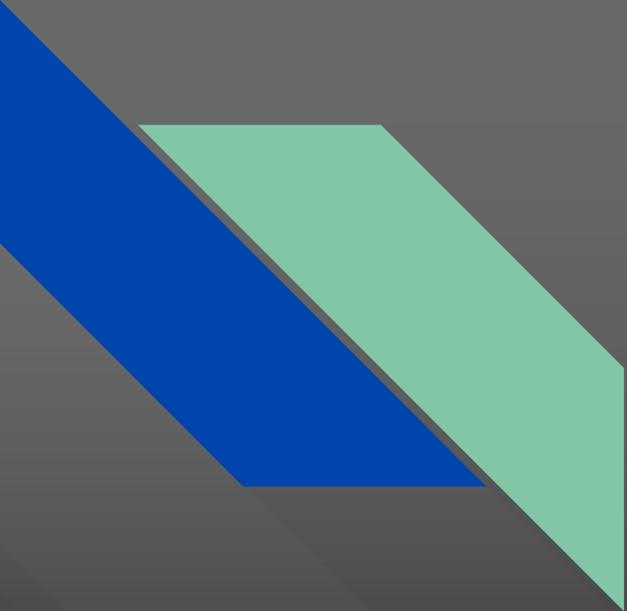


前言

專題實
作

專題成
果

問題與
討論



3. 程式設計

1. 編輯軟體介紹
2. 程式簡介

負責組員:李松濤

Arduino Mega 2560

- 數位I/O 接腳 * 54
- 類比輸入接腳 * 16
- Flash Memory : 256KB
- SRAM : 8KB
- EEPROM : 4KB



前言

專題實
作

專題成
果

問題與
討論

Arduino IDE

- 以Java編寫的跨平台整合開發環境
- 語法凸顯、括號匹配、自動縮排、一鍵編譯、燒入執行檔等功能
- 使用與C/C++相仿的程式語言



前言

專題實
作

專題成
果

問題與
討論

Visual Studio Code

- Visual Studio Code(簡稱VS code)
- 由微軟開發跨平台免費原始碼編譯器
- 支援語法凸顯、程式碼自動補全、程式碼重構
- 可透過內建擴充元件程式商店安裝擴充元件已加強軟體功能



前言

專題實
作

專題成
果

問題與
討論

LED矩陣相關變數

- LED矩陣顯示陣列 (matrix[data][z][y][x])
- 陣列顯示內容 (print_)
- 暫存器資料站存陣列 (led_port[5])

```
byte matrix[4][led_array_length][led_array_length][led_array_length] = //顯示陣列(x,y,z)
{
    //遊玩的畫面
    {{0, 0, 0, 0, 0, 0}, {0, 0, 0, 0, 0, 0}, {0, 0, 0, 0, 0, 0}, {0, 0, 0, 0, 0, 0}, {0, 0, 0, 0, 0, 0}},
    {{0, 0, 0, 0, 0, 0}, {0, 0, 0, 0, 0, 0}, {0, 0, 0, 0, 0, 0}, {0, 0, 0, 0, 0, 0}, {0, 0, 0, 0, 0, 0}},
    {{0, 0, 0, 0, 0, 0}, {0, 0, 0, 0, 0, 0}, {0, 0, 0, 0, 0, 0}, {0, 0, 0, 0, 0, 0}, {0, 0, 0, 0, 0, 0}},
    {{0, 0, 0, 0, 0, 0}, {0, 0, 0, 0, 0, 0}, {0, 0, 0, 0, 0, 0}, {0, 0, 0, 0, 0, 0}, {0, 0, 0, 0, 0, 0}},
    //倒數 1
    {{0, 1, 1, 1, 0, 0}, {0, 0, 0, 0, 0, 0}, {0, 0, 0, 0, 0, 0}, {0, 0, 0, 0, 0, 0}, {0, 0, 0, 0, 0, 0}},
    {{0, 1, 0, 0, 0, 0}, {0, 0, 0, 0, 0, 0}, {0, 0, 0, 0, 0, 0}, {0, 0, 0, 0, 0, 0}, {0, 0, 0, 0, 0, 0}},
    {{0, 0, 1, 0, 0, 0}, {0, 0, 0, 0, 0, 0}, {0, 0, 0, 0, 0, 0}, {0, 0, 0, 0, 0, 0}, {0, 0, 0, 0, 0, 0}},
    {{0, 1, 1, 0, 0, 0}, {0, 0, 0, 0, 0, 0}, {0, 0, 0, 0, 0, 0}, {0, 0, 0, 0, 0, 0}, {0, 0, 0, 0, 0, 0}},
    {{0, 0, 1, 1, 0, 0}, {0, 0, 0, 0, 0, 0}, {0, 0, 0, 0, 0, 0}, {0, 0, 0, 0, 0, 0}, {0, 0, 0, 0, 0, 0}},
    //倒數 2
    {{0, 1, 1, 1, 0, 0}, {0, 0, 0, 0, 0, 0}, {0, 0, 0, 0, 0, 0}, {0, 0, 0, 0, 0, 0}, {0, 0, 0, 0, 0, 0}},
    {{0, 1, 0, 0, 0, 0}, {0, 0, 0, 0, 0, 0}, {0, 0, 0, 0, 0, 0}, {0, 0, 0, 0, 0, 0}, {0, 0, 0, 0, 0, 0}},
    {{0, 1, 1, 1, 0, 0}, {0, 0, 0, 0, 0, 0}, {0, 0, 0, 0, 0, 0}, {0, 0, 0, 0, 0, 0}, {0, 0, 0, 0, 0, 0}},
    {{0, 0, 0, 1, 0, 0}, {0, 0, 0, 0, 0, 0}, {0, 0, 0, 0, 0, 0}, {0, 0, 0, 0, 0, 0}, {0, 0, 0, 0, 0, 0}},
    {{0, 1, 1, 1, 0, 0}, {0, 0, 0, 0, 0, 0}, {0, 0, 0, 0, 0, 0}, {0, 0, 0, 0, 0, 0}, {0, 0, 0, 0, 0, 0}},
    //倒數 3
    {{0, 1, 1, 1, 0, 0}, {0, 0, 0, 0, 0, 0}, {0, 0, 0, 0, 0, 0}, {0, 0, 0, 0, 0, 0}, {0, 0, 0, 0, 0, 0}},
    {{0, 0, 0, 1, 0, 0}, {0, 0, 0, 0, 0, 0}, {0, 0, 0, 0, 0, 0}, {0, 0, 0, 0, 0, 0}, {0, 0, 0, 0, 0, 0}},
    {{0, 1, 1, 1, 0, 0}, {0, 0, 0, 0, 0, 0}, {0, 0, 0, 0, 0, 0}, {0, 0, 0, 0, 0, 0}, {0, 0, 0, 0, 0, 0}},
    {{0, 0, 0, 1, 0, 0}, {0, 0, 0, 0, 0, 0}, {0, 0, 0, 0, 0, 0}, {0, 0, 0, 0, 0, 0}, {0, 0, 0, 0, 0, 0}},
    {{0, 1, 1, 1, 0, 0}, {0, 0, 0, 0, 0, 0}, {0, 0, 0, 0, 0, 0}, {0, 0, 0, 0, 0, 0}, {0, 0, 0, 0, 0, 0}},
};
```

前言

專題實
作

專題成
果

問題與
討論

矩陣顯示

- 使用掃描
- 使用暫存器控制I/O

```
ISR(TIMER1_OVF_vect) //掃描中斷 320micro  
{ //掃描LED三維陣列  
    TCNT1 = -52; //周中斷一次
```

```
    for(int j = 0; j < led_array_length; j++)  
    {  
        for(int i = 0; i < led_array_length; i++) //顯示  
        {  
            if(matrix[print_data][led_layer][j][i] == 1) //當顯示陣列內容為蛇的身體時  
            {  
                led_port[j] &= ~_BV(i);  
            }  
            else if(matrix[print_data][led_layer][j][i] == 2) //當顯示陣列內容為蘋果時  
            {  
                if(apple_flash/* || 1*/) //閃爍  
                {  
                    led_port[j] &= ~_BV(i);  
                }  
            }  
            else  
            {  
                led_port[j] |= _BV(i);  
            }  
        }  
    }  
}
```

```
PORTF = led_port[0];  
PORTB = led_port[1];  
PORTL = led_port[2];  
PORTC = led_port[3];  
PORTA = led_port[4];
```

```
PORTK = 0x00;  
PORTK = _BV(led_layer);
```

```
led_layer = (led_layer + 1) % led_array_length; //掃描計數+1  
}
```

前言

專題實
作

專題成
果

問題與
討論

搖桿相關變數

- 搖桿輸入方向: joystick_input
- 搖桿X、Y軸0準位
joystick_x_level、joystick_y_level
- 搖桿方向轉換陣列: dir[6][5]

前言

專題實
作

專題成
果

問題與
討論

抓取搖桿輸入

- 使用類比輸入接腳
- 使用判斷式判斷現在狀態

```
void joystick_direction() //取得搖桿方向***** (讀取程式) 前後左右 0123
{
    int x, y;
    x = analogRead(joystick_x);
    y = analogRead(joystick_y);
    //Serial.println(joystick_x_level);
    //Serial.println(joystick_y_level);

    if(x - joystick_x_level < -300)
    {
        joystick_input = 0;
    }
    else if(x - joystick_x_level > 300)
    {
        joystick_input = 1;
    }
    else if(y - joystick_y_level > 300)
    {
        joystick_input = 2;
    }
    else if(y - joystick_y_level < -300)
    {
        joystick_input = 3;
    }
    else
    {
        joystick_input = 4;
    }
}
```

```
void snake_direction() //判定貪食蛇移動方向 前後左右上下 012345
{
    snake_move_direction = dir[snake_move_direction][joystick_input];
}
```

前言

專題實
作

專題成
果

問題與
討論

LCD顯示

- 使用LiquidCrystal_I2C
- 區分4種顯示狀況

```
void lcd_display(byte m
```



```
else if(mod = 3)//finish_lose
{
    lcd.setCursor(0, 0);
    lcd.print("score:");

    for(int i = 0; i < 10 - count_digit(score); i++)
    {
        lcd.print(' ');
    }
    lcd.print(score);

    lcd.setCursor(0, 1);
    lcd.print("high record:");
    for(int i = 0; i < 4 - count_digit(score_highest); i++)
    {
        lcd.print(' ');
    }
    if(score_highest)
    {
        lcd.print(score_highest);
    }
    else
    {
        lcd.print('0');
    }
}
```

前言

專題實
作

專題成
果

問題與
討論

貪食蛇相關變數

- 貪食蛇座標陣列
snake_coordinate_x[]、
snake_coordinate_y[]、
snake_coordinate_z[]
- 貪食蛇移動方向
snake_move_direction

前言

專題實
作

專題成
果

問題與
討論

貪食蛇生成

- 設置初始位置
- 將Y作標遞增

```
void snake_reset()                //貪食蛇重製座標
{
    snake_coordinate_x.clear();    //清除貪食蛇座標陣列
    snake_coordinate_y.clear();    //清除貪食蛇座標陣列
    snake_coordinate_z.clear();    //清除貪食蛇座標陣列

    LinkedList<byte> tmp;          //新增一個暫時的陣列
    tmp.add(2); //設定x軸座標
    tmp.add(0); //設定y軸座標
    tmp.add(4); //設定z軸座標

    for(int i = 0; i < 3; i++)
    {
        snake_coordinate_x.unshift(tmp[0]); //將臨時陣列猜入座標陣列中
        snake_coordinate_y.unshift(tmp[1]); //將臨時陣列猜入座標陣列中
        snake_coordinate_z.unshift(tmp[2]); //將臨時陣列猜入座標陣列中
        tmp[1] += 1; //將y軸座標+1
    }
}
```

前言

專題實
作

專題成
果

問題與
討論

貪食蛇移動

- 新增暫時座標陣列
- 按照移動方向更改座標
- 將座標控制在0~4之間並加入座標陣列
- 呼叫碰撞程式

```
void snake_move() //貪食蛇移動
{
    LinkedList<int> tmp;//(xyz)
    tmp.clear();
    tmp.add(snake_coordinate_x[0]);
    tmp.add(snake_coordinate_y[0]);
    tmp.add(snake_coordinate_z[0]);

    case 0://前
        tmp[1] += 1;
        break;
    case 1://後
        tmp[1] -= 1;
        break;
    case 2://左
        tmp[0] -= 1;
        break;
    case 3://右
        tmp[0] += 1;
        break;
    case 4://上
        tmp[2] += 1;
        break;
    case 5://下
        tmp[2] -= 1;
        break;

    tmp[i] = (tmp[i] + led_array_length) % led_array_length;
    snake_coordinate_x.unshift(tmp[0]);
    snake_coordinate_y.unshift(tmp[1]);
    snake_coordinate_z.unshift(tmp[2]);
    snake_hit(tmp[0], tmp[1], tmp[2]); //蛇是否有碰撞
}
```

前言

專題實作

專題成果

問題與討論

貪食蛇碰撞

- 判斷是否碰到蘋果
- 判斷是否碰到身體

```
void snake_hit(byte x, byte y, byte z)//蛇是否有碰撞
{
    if(apple_coordinate[0] == x) && (apple_coordinate[1] == y) && (apple_coordinate[2] == z)//當頭撞到蘋果
    {
        score += 1;
        lcd_display(1);
        create_apple();
    }
    else//如果沒碰到蘋果
    {
        snake_coordinate_x.pop();
        snake_coordinate_y.pop();
        snake_coordinate_z.pop();
    }
}

for(int i = 1; i < snake_coordinate_x.size(); i++)//當頭撞到身體
{
    if( (snake_coordinate_x[i] == snake_coordinate_x[0])&&
        (snake_coordinate_y[i] == snake_coordinate_y[0])&&
        (snake_coordinate_z[i] == snake_coordinate_z[0])
        )
    {
        continued = 0;
    }
}
}
```

前言

專題實
作

專題成
果

問題與
討論

蛇的頭閃爍

- 使用Timer4的溢位中斷

```
ISR(TIMER4_OVF_vect)    //蛇頭閃爍
{
    TCNT4 = -763;
    if(continued)
    {
        if(matrix[0][snake_coordinate_z[0]][snake_coordinate_y[0]][snake_coordinate_x[0]] == 1)
        {
            matrix[0][snake_coordinate_z[0]][snake_coordinate_y[0]][snake_coordinate_x[0]] = 0;
        }
        else
        {
            matrix[0][snake_coordinate_z[0]][snake_coordinate_y[0]][snake_coordinate_x[0]] = 1;
        }
    }
}
```

前言

專題實作

專題成果

問題與討論

蘋果生成

- 隨機生成一個三維座標
- 判斷是否生成在蛇身上

```
void apple_create() //生成蘋果
{
    byte accept = 0; //重複
    byte coordinate[3]; //臨時蘋果座標陣列
    do
    {
        accept = 0; //重複0個軸座標

        for(int i = 0; i < 3; i++)
        {
            coordinate[i] = random(led_array_length); //座標隨機(0~3)
        }

        for(int i = 0; i < snake_coordinate_x.size(); i++)
        {
            if((snake_coordinate_x[i] == coordinate[0])&&
                (snake_coordinate_y[i] == coordinate[1])&&
                (snake_coordinate_z[i] == coordinate[2])) //如果x軸座標一樣
            {
                accept = 1;
            }
        }
    } while(accept); //如果重複，重隨機一個點

    for(int i = 0; i < 3; i++)
    {
        apple_coordinate[i] = coordinate[i]; //將臨時座標放入蘋果座標陣列
    }
    apple_to_matrix();
}
```

前言

專題實
作

專題成
果

問題與
討論

蘋果的閃爍

- 使用Timer3的溢位中斷

```
ISR(TIMER3_OVF_vect)           //蘋果閃爍 掃描LED三維陣列
{
    TCNT3 = -3906; //50周中斷一次
    if(continued)
    {
        if(matrix[0][apple_coordinate[2]][apple_coordinate[1]][apple_coordinate[0]] == 1)
        {
            matrix[0][apple_coordinate[2]][apple_coordinate[1]][apple_coordinate[0]] = 0;
        }
        else
        {
            matrix[0][apple_coordinate[2]][apple_coordinate[1]][apple_coordinate[0]] = 1;
        }
    }
}
```

前言

專題實
作

專題成
果

問題與
討論



遊戲數據

- 遊戲進行變數(continued)
- 現在分數(score)
- 最高分紀錄(score_highest)

前言

專題實
作

專題成
果

問題與
討論

主程式-初始化設定

- 初始化貪食蛇移動方向
- 將得分歸0
- 將最高分重新從EEPROM中讀出
- 所有變數初始化
- 呼叫副程式初始化貪食蛇、重設顯示陣列生成第一顆蘋果

```
snake_move_direction = 0;
int jx, jy;
score = 0;

lcd_display(0);          //LCD顯示開始畫面

joystick_input = 4;
step_delay_time = step_delay_array[1];

while(digitalRead(bottom1_pin) == 0) {

if(step_delay_time == 600)
{
    score_highest = EEPROM.read(score_highest_hard_eeeprom_address); //把最高紀錄從EEPROM中取出
}
if(step_delay_time == 1000)
{
    score_highest = EEPROM.read(score_highest_normal_eeeprom_address); //把最高紀錄從EEPROM中取出
}
if(step_delay_time == 1200)
{
    score_highest = EEPROM.read(score_highest_easy_eeeprom_address); //把最高紀錄從EEPROM中取出
}

lcd_display(1);          //顯示遊玩畫面
display_array_reset();

snake_reset();           //重設貪食蛇
apple_create();          //生成第一顆蘋果
snake_to_matrix();

count_down();           //倒數321
continued = 1;
}
```

前言

專題實
作

專題成
果

問題與
討論

主程式-調整難度

- 進入遊戲前選擇難易度

```
while(digitalRead(botton1_pin) == 0)
{
    joystick_direction();

    if(joystick_input == 0 || joystick_input == 1)
    {
        step_delay_time = step_delay_array[1];

        lcd.setCursor(0, 0);
        lcd.print(" normal level ");
    }
    else if(joystick_input == 3)
    {
        step_delay_time = step_delay_array[0];

        lcd.setCursor(0, 0);
        lcd.print(" easy level ");
    }
}
```

```
else if(joystick_input == 2)
{
    step_delay_time = step_delay_array[2];

    lcd.setCursor(0, 0);
    lcd.print(" hard level ");
}

lcd.setCursor(0, 1);
lcd.print("press the joycon");

unsigned long tmp_delay = millis();
while(millis() - tmp_delay < 1){}
//等待按鈕按下
```

前言

專題實作

專題成果

問題與討論

主程式-遊戲中

- 將貪食蛇和蘋果的位子放入顯示陣列
- 使用while迴圈每1ms讀取一次搖桿輸入

```
while(continued)
{
    display_array_reset();
    snake_to_matrix();
    apple_to_matrix();
    unsigned long joystick_delay = millis();
    while(millis() - joystick_delay < step_delay_time)
    {
        joystick_direction();
        unsigned long tmp_delay = millis();
        while(millis() - tmp_delay < 1){}
    }
    snake_direction();

    snake_move();
}
```

前言

專題實
作

專題成
果

問題與
討論

前言

主程式-遊戲結束

- 判斷是有創造新的紀錄
- 等按鈕按下後等待，避免誤觸

```
if(score > score_highest) //當分數比最高分高
{
    lcd_display(2); //顯示結束畫面(贏)
    Serial.println("finish display1");
    if(step_delay_time == 600)
    {
        EEPROM.update(score_highest_hard_eeeprom_address, score); //把最高紀錄從EEPROM中取出
    }
    if(step_delay_time == 1000)
    {
        EEPROM.update(score_highest_normal_eeeprom_address, score); //把最高紀錄從EEPROM中取出
    }
    if(step_delay_time == 1200)
    {
        EEPROM.update(score_highest_easy_eeeprom_address, score); //把最高紀錄從EEPROM中取出
    }
}
else
{
    lcd_display(3); //顯示結束畫面(一班)
    //Serial.println("finish display2");
}

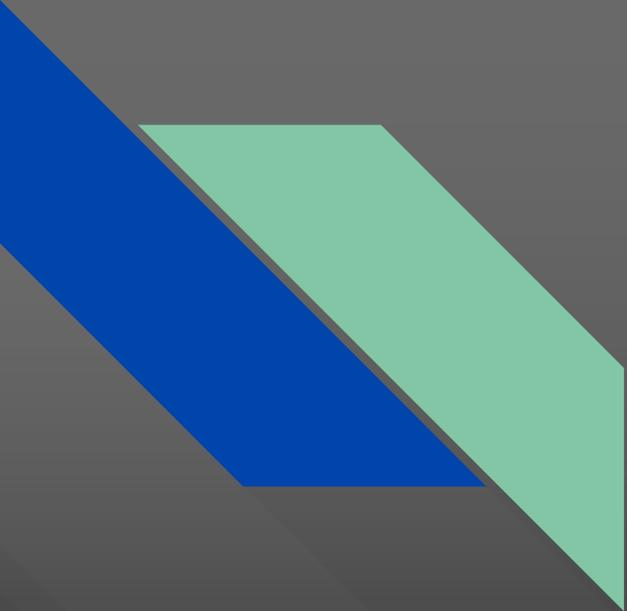
Serial.println("finish\n\n");
display_array_reset();

while(digitalRead(botton1_pin) == 0){ //等待按鈕按下
    unsigned long time_last = millis();
    while(millis() - time_last < 500){
        Serial.println("end");
    }
}
```

專題實
作

專題成
果

問題與
討論



4.外觀設計

1. 電路設計
2. 程式設計
3. LED陣列設計
4. 外觀設計

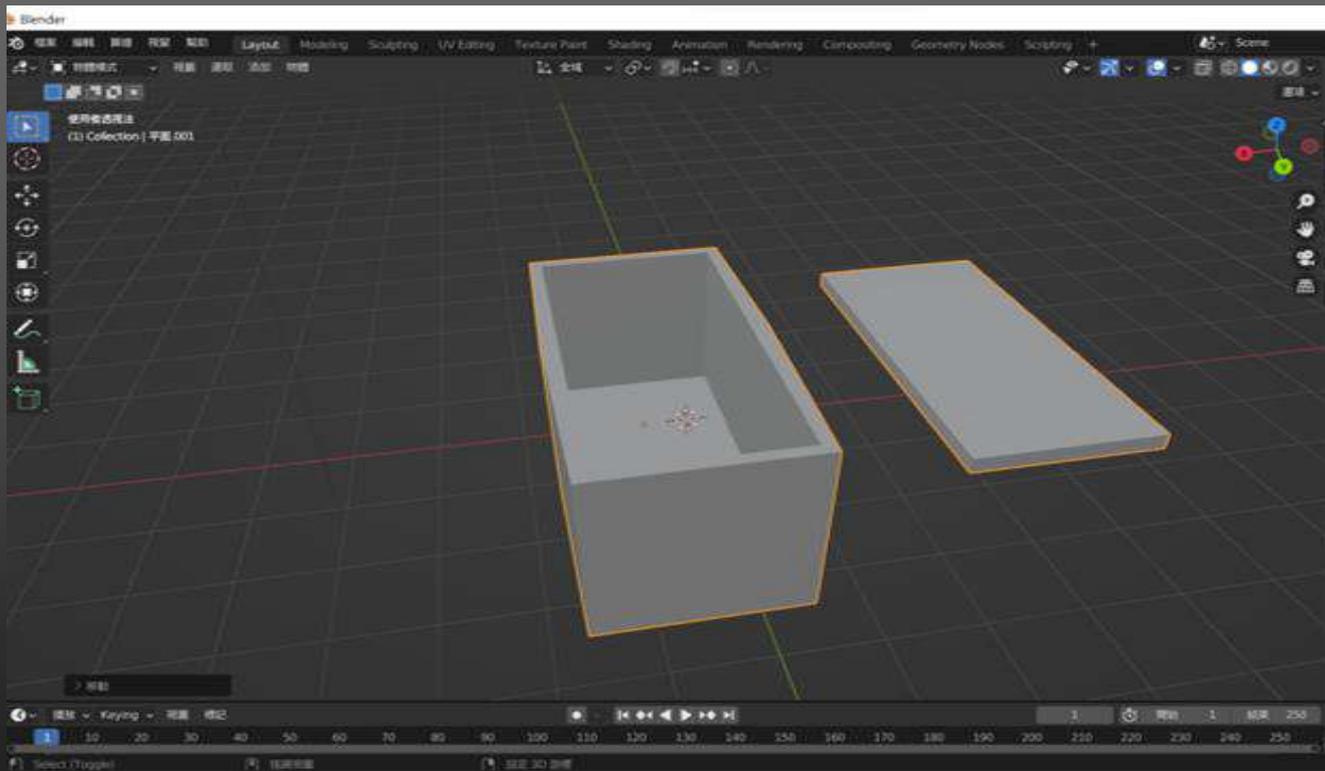
負責組員:趙宇晨

前言

專題實
作

專題成
果

問題與
討論



試作紙盒模型



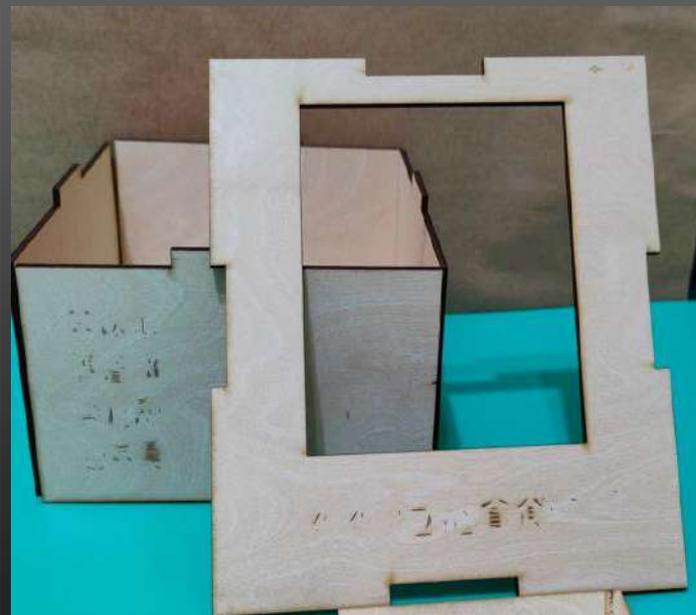
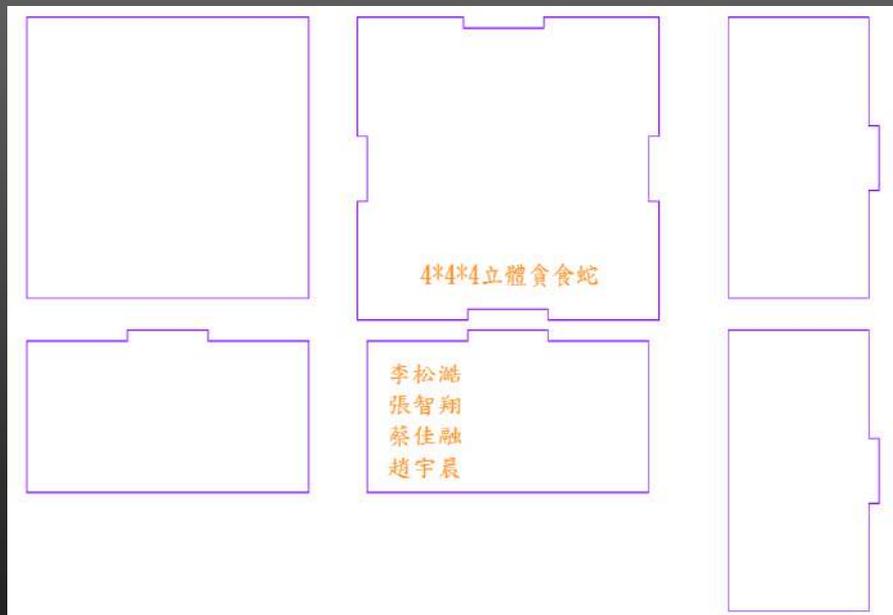
前言

專題實
作

專題成
果

問題與
討論

雷射切割



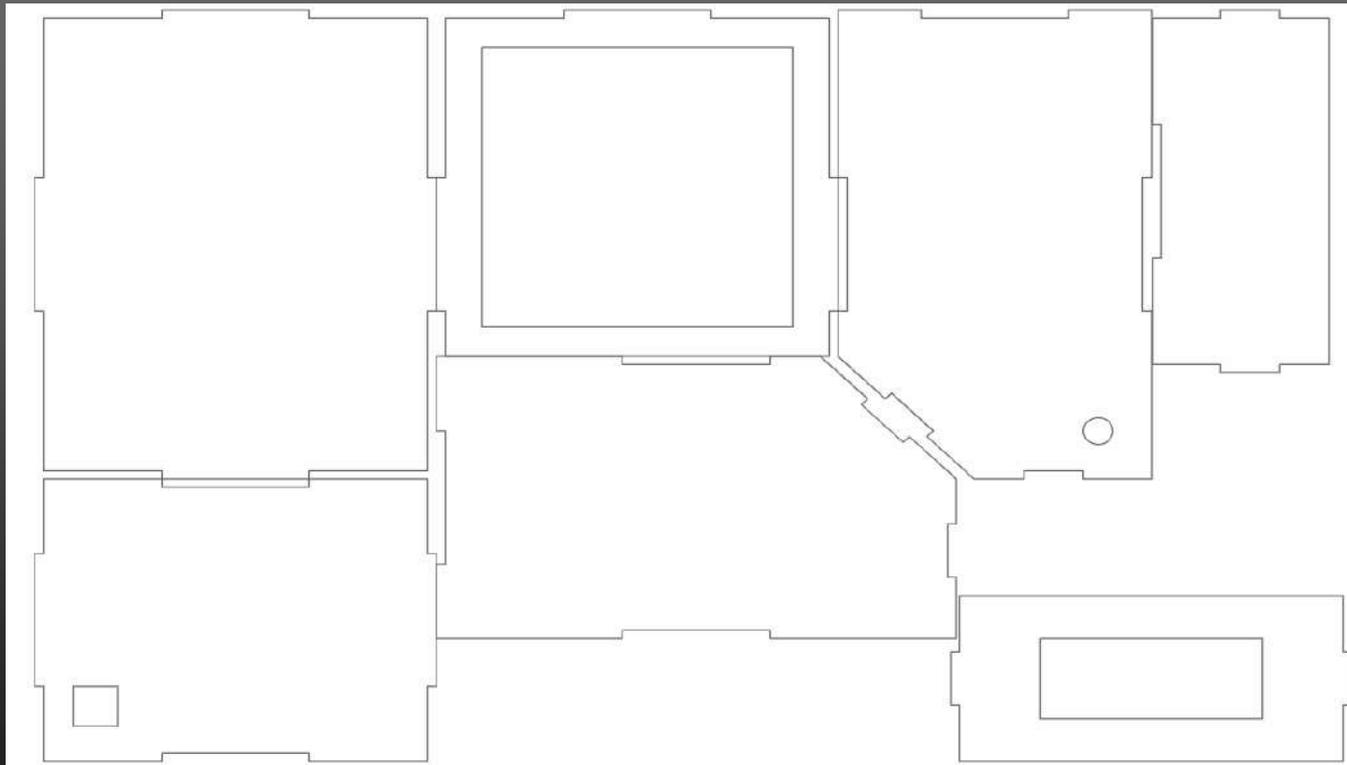
前言

專題實
作

專題成
果

問題與
討論

成品設計圖



前言

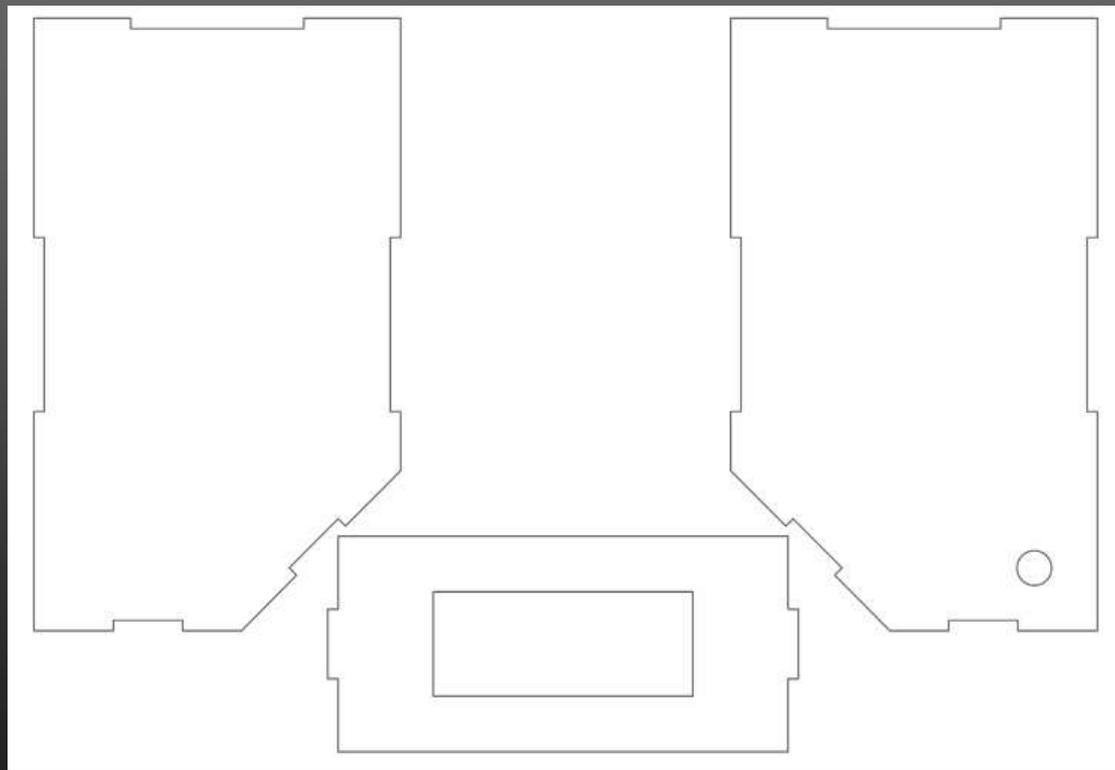
專題實
作

專題成
果

問題與
討論



成品設計圖

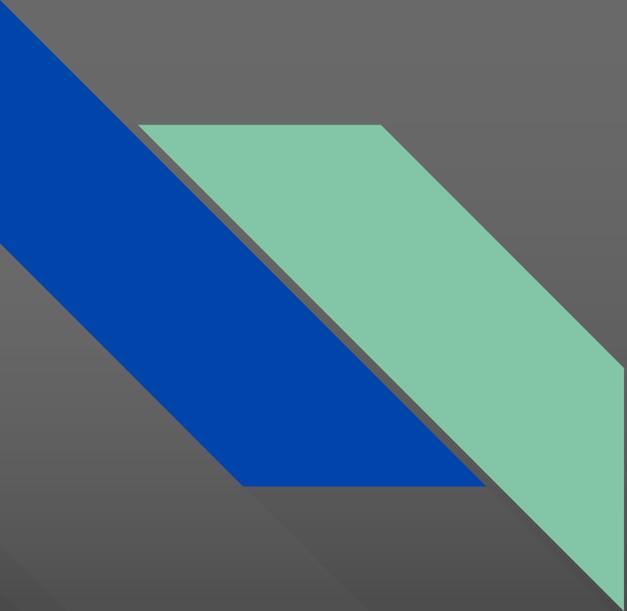


前言

專題實
作

專題成
果

問題與
討論



三、專題成果

- 成果影片

成果影片

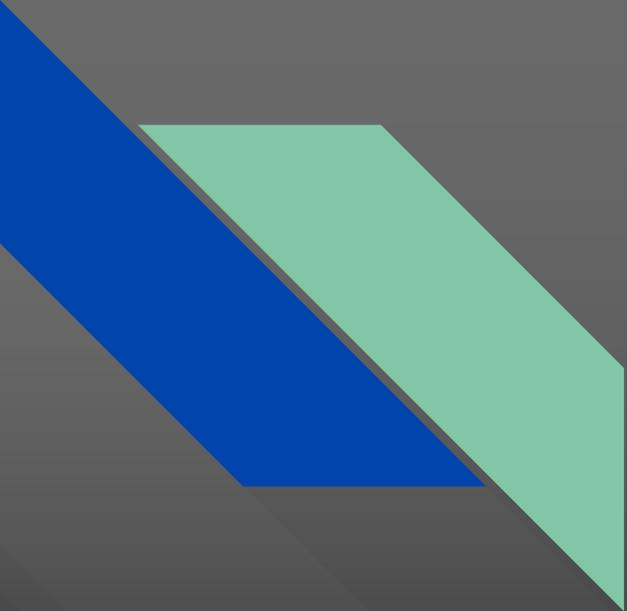
前言

專題實
作

專題成
果

問題與
討論

CyberLink
PowerDirector



四、Q&A

前言

專題實
作

專題成
果

問題與
討論

感謝大家的聆聽