

```

1  #include <Adafruit_NeoPixel.h>
2
3  /*<1>紅色按鈕-外部中斷腳位PB2 (INT0)：具照明及警示作用
4     <2>綠色按鈕-PB3內含上拉電阻：遊戲鍵\開始及設定鍵
5     <3>環燈訊號腳-PB0
6     <4>無源蜂鳴器-PB1
7
8  */
9  //-----
10 #include <Adafruit_NeoPixel.h>//載入第三方程式控制環燈
11 #define LED_PIN 0 //環號串列訊號腳位PB0
12 #define KEY_PUSH 3//按鈕控制腳位(內建上拉電阻,平時為HIGH)
13 #define NUMLEDS 24//環燈=總顆數
14 #define BRIGHTNESS 50//燈亮度在15~50即可,最大亮度為255
15 #define buzzerPin 1 //無源蜂鳴器訊號腳位PB1
16 int period = 1000; //計時器延時週期
17 int Int_period = 700;//中斷延時週期
18 int GameState = 0; //遊戲初始狀態=0 1(第1關-3燈) 2(第2關-2燈) 3(第3關-1燈)
19 byte RingSpeed[] = {110, 100, 60, 40}; //共有4個等級的遊戲
20 byte Speed=0;//速度盒編號 RingSpeed陣列編號
21 byte Set_Led[]={0, 0, 0}; //用陣列設定號燈(3燈=3個陣列.2燈2個陣列.1燈1個陣列)
22 byte led_run_at = 0; //跑馬燈號現在位置(led跑到那個位置)
23 byte Setting_Led = 0; //燈號設定點(亂數值存放)
24 byte LEVEL=0;//關卡
25 byte Score=0;//計分用
26 unsigned long Time_Get = 0;//擷取時間點,開機55天會溢位(歸0)
27 unsigned long IntTime_Get = 0;//中斷擷取時間點
28 boolean Go_Play = false;//遊戲處於等待狀態(布林值)
29 volatile byte Int state = 0; //定義輸入狀態
30 //宣告環燈 顆數 腳位
31 Adafruit_NeoPixel RingLED = Adafruit_NeoPixel(NUMLEDS, LED_PIN, NEO_GRB + NEO_KHZ800);
32 //-----
33 //-----重置函式
34 void(* resetFunc)(void)=0;
35 //-----環燈初始化動作
36 void setup() {
37     RingLED.begin(); //初始化(環燈)
38     RingLED.setBrightness(BRIGHTNESS); //環燈亮度
39     Time_Get = millis();//擷取開機時間
40     pinMode(KEY_PUSH, INPUT_PULLUP);//按鈕輸入腳3(KEY_PUSH),內建上拉電阻
41     pinMode(buzzerPin, OUTPUT);//
42     // Serial.begin(9600);//串列傳輸啟動
43     randomSeed(analogRead(KEY_PUSH));//(擷取亂數因子)解決亂數不亂的問題
44     //#####以下是將GIMSK中斷暫存器的INT0啟用,MCUCR控制暫存器設為負緣觸發
45     pinMode(2, INPUT_PULLUP);//紅色按鈕,外部中斷腳位
46     GIMSK |= (1<<INT0); // INT0啟用
47     MCUCR |= (1<<ISC01); // MCUCR控制暫存器設為負緣觸發
48     sei(); //enabling global interrupt
49 }
50 //-----
51 //-----無窮迴圈
52 void loop() {
53     if (GameState == 0){
54         //Go_Play == false等待按鈕被按下去,則往下做3關,每關4種變速
55         if (digitalRead(KEY_PUSH) == LOW && Go_Play == false) {
56             delay(1000);//除火花,延遲 1000 毫秒,也就是 1 秒鐘。
57             PlayStar();//奏音樂
58             led_run_at = 0;//跑馬燈現在位置
59             Go_Play = true;//變數為true,可以取設定點(亂數)
60             //Serial.print("!");//*判斷按鈕準不準
61             GameState=1; //開始玩第1關
62         }
63         Z_Random();//亂數顯示燈號(等待綠鍵被按!)
64     }
65     if (GameState == 1){Level_1();} //往第1關開始遊戲(每關4種變速)
66     if (GameState == 2){Level_2();} //往第2關開始遊戲(每關4種變速)
67     if (GameState == 3){Level_3();} //往第3關開始遊戲(每關4種變速)
68 }
69 //-----
70 //-----第1關(3顆設定燈):-----
71 void Level_1(){
72     RingLED.clear();//清除燈號
73     led_run_at=0;//跑馬燈歸零

```

```

74 while (GameState == 1) {
75     period = RingSpeed[Speed]; // [0][1][2][3] 存放不同速度值
76     //====取亂數置入設定點-->再延伸成3個設定燈====
77     if (millis()-Time_Get > period){ //延時 (亮燈週期過了,再重設週期)
78         Time_Get = millis(); // 擷取當下時間放入Time_Get
79         if (Go_Play==true){ //Go_Play為true,可以取設定點(亂數)
80             Setting_Led = random(NUMLEDS); //隨機-取設定點
81             Go_Play = false; //隨機設定號燈完畢,等待按鈕!
82         }
83     }
84     //----設定3個點-----
85     //以下亮3個號燈(跨14-15-0)
86     Set_Led[0]=Setting_Led; //陣列0=放入亂數值
87     if (Setting_Led == (NUMLEDS-2)){ //第14顆:(若16顆為例)
88         Set_Led[1]=(NUMLEDS-1); //第15顆
89         Set_Led[2]=0; //第0顆
90     }
91     //以下亮3個號燈(跨15-0-1)
92     else if (Setting_Led == (NUMLEDS-1)){
93         Set_Led[1]=0;
94         Set_Led[2]=1;
95     }
96     else{
97         Set_Led[1]=Set_Led[0]+1;
98         Set_Led[2]=Set_Led[0]+2;
99     }
100    //---點亮3個燈[ ][ ]---
101    RingLED.setPixelColor(Set_Led[0], 0, 255, 0); // (綠)
102    RingLED.setPixelColor(Set_Led[1], 0, 255, 0); // (綠)
103    RingLED.setPixelColor(Set_Led[2], 0, 255, 0); // (綠)
104    RingLED.show();
105    LED_Runing(); //呼叫跑馬燈
106    Check_Push(GameState); //檢查按鈕是否被按下?
107 }
108 }
109 //-----第2關(2顆設定燈):-----
110 void Level_2(){
111     RingLED.clear(); //清除燈號
112     led_run_at=0; //跑馬燈歸零
113     while (GameState == 2) {
114         period = RingSpeed[Speed]; // (陣列中的數字愈大愈慢)
115         //====取亂數置入設定點-->再延伸成2個設定燈====
116         if (millis()-Time_Get > period){ //延時 (亮燈週期過了,再重設週期)
117             Time_Get = millis(); // 擷取當下時間放入Time_Get
118             //====亂數設定OK了嗎? Yes(改false)
119             if (Go_Play==true){ //Go_Play為true,可以取設定點(亂數)
120                 Setting_Led = random(NUMLEDS);
121                 Go_Play = false; //隨機設定號燈完畢,等待按鈕!
122             }
123         }
124         //---設定2個點-----
125         //====以下亮2個號燈(跨15-0)====
126         Set_Led[0]=Setting_Led; //陣列0=放入亂數值
127         if (Setting_Led==(NUMLEDS-1)){ Set_Led[1]=0; }
128         else{ Set_Led[1]=Set_Led[0]+1; }
129         RingLED.setPixelColor(Set_Led[0], 0, 255, 0); // (綠)
130         RingLED.setPixelColor(Set_Led[1], 0, 255, 0); // (綠)
131         RingLED.show();
132         LED_Runing(); //呼叫跑馬燈
133         Check_Push(GameState); //檢查按鈕是否被按下?
134     }
135 }
136 //-----第3關(1顆設定燈):-----
137 void Level_3(){
138     RingLED.clear(); //清除燈號
139     led_run_at=0; //跑馬燈歸零
140     while (GameState == 3) {
141         period = RingSpeed[Speed]; // (陣列中的數字愈大愈慢)
142         //====取亂數置入設定點====
143         if (millis()-Time_Get > period){ //延時 (亮燈週期過了,再重設週期)
144             Time_Get = millis(); // 擷取當下時間放入Time_Get
145             //====亂數設定OK了嗎? Yes(改false)
146             if (Go_Play==true){ //Go_Play為true,可以取設定點(亂數)
147                 Setting_Led = random(NUMLEDS);

```

```

147         Go_Play = false; //隨機設定號燈完畢,回歸,等待按鈕!
148     }
149     //-----
150     //====以下亮1個號燈====
151     Set_Led[0]=Setting_Led;
152     RingLED.setPixelColor(Set_Led[0], 0, 255, 0); //(綠)
153     RingLED.show();
154     LED_Runing(); //呼叫跑馬燈
155     Check_Push(GameState); //檢查按鈕是否被按下?
156 }
157 }
158 }
159 //-----
160
161 //-----
162 //----跑馬燈----
163 void LED_Runing(){
164     led_run_at++; //循環燈...往下累加
165     if (led_run_at >= NUMLEDS) { //如果循環紅點位置=60符合則歸零
166         RingLED.setPixelColor(led_run_at-1, 0, 0, 0); //最後一顆歸0
167         led_run_at = 0; //歸零
168     }
169     RingLED.setPixelColor(led_run_at, 255, 0, 0); //亮燈
170     RingLED.setPixelColor(led_run_at-1, 0, 0, 0); //往後關燈
171     RingLED.show();
172 }
173 //-----
174 //----壓下去(檢查是否得分?)-----
175 void Check_Push(byte LV){ //Go_Play == false 為等待的意思
176     if (digitalRead(KEY_PUSH) == LOW && Go_Play == false ) { //按鈕被按下去,往下做
177         delay(1000);
178         Go_Play=true;
179         //Serial.print("S"); //判斷按鈕準不準
180         //-----
181         //----得分-----
182         if(LV==1){
183             if (Set_Led[0]==led_run_at || Set_Led[1]==led_run_at || Set_Led[2]==led_run_at) {
184                 Score++;
185                 ScreLed(Score,550); //呼叫計分燈號函式(累計得分,高音頻)
186             }
187             else{ScreLed(Score,150);} //呼叫計分燈號函式(不累計,低音頻)
188         }
189         else if(LV==2){
190             if (Set_Led[0]==led_run_at || Set_Led[1]==led_run_at) {
191                 Score++;
192                 ScreLed(Score,550); //呼叫計分燈號函式(累計得分,高音頻)
193             }
194             else{ScreLed(Score,150);} //呼叫計分燈號函式(不累計,低音頻)
195         }
196         else if(LV==3){
197             if (Set_Led[0]==led_run_at) {
198                 Score++;
199                 ScreLed(Score,550); //呼叫計分燈號函式(累計得分,高音頻)
200             }
201             else{ScreLed(Score,150);} //呼叫計分燈號函式(不累計,低音頻)
202         }
203         //---換速度檔次-----
204         Speed++; //換下個速度檔次
205         //Serial.print(Speed); //查看勝利或失敗 */
206         //Serial.print(">");
207         //若4種檔次完成,換第二關
208         if(Speed==4){
209             //Serial.println("-");
210             GameState++; //下一大關
211             if(GameState==4){GameState=0; Go_Play=false; Score=0;} //3大關全部完成
212             Speed=0;
213             PlayFamily();
214         }
215     }
216 }
217 //-----
218 //----計分燈號+鳴----
219 void ScreLed(byte Gain,int Freq){ //接收得分值,及音頻

```

```

220 RingLED.clear();//清除燈號
221 for(byte i=0;i<Gain;i++){RingLED.setPixelColor(i, 255 , 255, 0);}
222 RingLED.show();
223 //--鳴--
224 for (byte j = 0; j <2; j++){
225     tinyTone(Freq, 400);}
226 delay(450);
227 RingLED.clear();//清除燈號
228 }
229 //-----
230 //----亂數顯示燈號----
231 void Z_Random(){
232 for (byte i = 0; i < NUMLEDS; i++) { //環燈號訊號
233     RingLED.setPixelColor(i, random(0, 255), random(0, 255), random(0,
234         255)); //黃色
235     delay(15);
236     RingLED.show();
237 }
238 //-----
239 //----星際爭霸----
240 void PlayStar(){
241 int Star[] =
242 {440,440,440,349,523,440,349,523,440,659,659,659,698,523,415,349,523,440};
243 int StarDur[] = {400, 400, 400, 250, 50, 400, 250, 50, 900, 400, 400, 400, 250, 50,
244 400, 250, 50, 900};
245 for (int thisNote = 0; thisNote < 18; thisNote++) { //18個音符
246     tinyTone(Star[thisNote],StarDur[thisNote]);
247     delay(30); // 音符之間的間隔
248 }
249 //-----
250 //----全家音樂----
251 void PlayFamily(){
252 int FamilyMusic[]={330, 262, 196, 262, 294, 392, 0, 0, 294, 330,0, 294, 196, 262,0};
253 for (int thisNote = 0; thisNote < 15; thisNote++) { //14個音符
254     tinyTone(FamilyMusic[thisNote],200);
255     delay(30); // 音符之間的間隔
256 }
257 //-----
258 //----中斷函式----(處理第一次中斷無效問題)
259 ISR (INT0_vect){
260 if ( millis() - IntTime_Get > Int_period ){
261     IntTime_Get = millis();
262     Int_state++; //累計按下次數
263     //Serial.print(Int_state);
264     while(Int_state==1){ //按1次-->手電筒模式
265         for (byte i = 0; i < NUMLEDS; i++) { //環燈號訊號
266             RingLED.setPixelColor(i,255,255,255); //白色
267             RingLED.setBrightness(180); //環燈亮度
268             RingLED.show();
269         }
270     }
271     while(Int_state==2){ //按2次-->閃白光模式
272         for (byte i = 0; i < NUMLEDS; i++) { //環燈號訊號
273             RingLED.setPixelColor(i,255,255,255); //白色
274             RingLED.setBrightness(180); //環燈亮度
275             RingLED.show();
276         }
277         delay(400);
278         for (byte i = 0; i < NUMLEDS; i++) { //環燈號訊號
279             RingLED.setPixelColor(i,0,0,0); //黑色
280             RingLED.show();
281         }
282         delay(400);
283     }
284     while(Int_state==3){ //按3次-->閃紅光模式
285         for (byte i = 0; i < NUMLEDS; i++) { //環燈號訊號
286             RingLED.setPixelColor(i,255,0,0); //紅色
287             RingLED.setBrightness(180); //環燈亮度
288             RingLED.show();
289         }

```

```

290         delay(400);
291         for (byte i = 0; i < NUMLEDS; i++) { //環燈號訊號
292             RingLED.setPixelColor(i,0,0,0); //黑色
293             RingLED.show();
294         }
295         delay(400);
296     }
297     while(Int_state==4){ //按4次-->閃紅光+警示音模式
298         for (byte i = 0; i < NUMLEDS; i++) { //環燈號訊號
299             RingLED.setPixelColor(i,255,0,0); //紅色
300             RingLED.setBrightness(180); //環燈亮度
301             RingLED.show();
302         }
303         tinyTone(650, 200);
304         delay(300);
305         for (byte i = 0; i < NUMLEDS; i++) { //環燈號訊號
306             RingLED.setPixelColor(i,0,0,0); //黑色
307             RingLED.show();
308         }
309         tinyTone(900, 200);
310         delay(300);
311     }
312     if(Int_state==5){ //按5次-->重置模式
313         Int_state=0;
314         resetFunc();
315     }
316 }
317 }
318 //-----
319 //----音樂播放函式(傳音符[頻率],聲音長度)
320 void tinyTone(int frequency, int length){
321     unsigned long period;
322     unsigned long n;
323     period=1000000/frequency;
324     n=1000*(long)length/period;
325     for(int j=1;j<n;j++){
326         digitalWrite(buzzerPin,HIGH);
327         delayMicroseconds(period/2);
328         digitalWrite(buzzerPin,LOW);
329         delayMicroseconds(period/2);
330     }
331 }
332

```