

```

1  /*-----01-Arduino系列-Attiny85 遊戲及照明兩用---
2  <1>紅色按鈕PB2腳-外部中斷腳位PB2-外部中斷腳位(INT0)：具照明及警示作用
3  <2>綠色按鈕PB3腳-內含上拉電阻：遊戲鍵\開始及設定鍵
4  <3>環燈訊號腳-PB0腳
5  <4>無源蜂鳴器-PB1腳
6  <5>可使用的腳位共6支,可再增加一個按鈕調亮度
7  */
8  //-----02-C語言外部宣告-----
9  #include <Adafruit_NeoPixel.h> //載入第三方程式控制環燈WS2812
10 #define LED_PIN 0 //環號WS2812訊號腳位接PB0
11 #define KEY_PUSH 3 //按鈕(綠)控制腳位(內建上拉電阻,平時為HIGH)
12 #define NUMLEDS 24 //依環燈WS2812顆數修正
13 #define BRIGHTNESS 30 //燈亮度在15~50即可,最大亮度為255
14 #define buzzerPin 1 //無源蜂鳴器訊號腳位PB1
15
16 int period = 1000; //計時器延時週期
17 int Int_period = 700; //中斷延時週期
18
19 byte 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)*
20 byte RingSpeed[] = {110, 100, 60, 40}; //速度陣列-共有4個速級的遊戲
21 byte Speed=0; //速度盒編號 RingSpeed陣列編號
22 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)
23 byte led_run_at = 0; //跑馬燈號現在位置(led跑到那個位置)
24 byte RANled_run_at = 0; //跑馬亂數燈
25 byte Setting_Led = 0; //燈號設定點(亂數值存放)
26 byte LEVEL=0; //關卡
27 byte Score=0; //計分用
28
29 unsigned long Time_Get = 0; //擷取時間點,開機55天會溢位(歸0)
30 unsigned long IntTime_Get = 0; //中斷擷取時間點
31 boolean Go_Play = false; //遊戲處於等待狀態(布林值)
32 volatile byte Int_state = 0; //定義輸入狀態
33
34 //-----03定義WS2812環燈：顆數 腳位 頻率
35 Adafruit_NeoPixel RingLED = Adafruit_NeoPixel(NUMLEDS, LED_PIN, NEO_GRB + NEO_KHZ800);
36
37 //-----04 重置函式 -----
38 void(* resetFunc)(void)=0; //如果有呼叫 resetFunc函式,則軟體重啟
39
40 //-----05 程式開始-1.進行WS2812環燈初始化動作 2.按鈕宣告輸入 3.蜂鳴器宣告輸出
41 //4.外部中斷腳位宣告
42 void setup() {
43     RingLED.begin(); //初始化(環燈)
44     RingLED.setBrightness(BRIGHTNESS); //環燈亮度
45     Time_Get = millis(); //擷取開機時間
46     pinMode(KEY_PUSH, INPUT_PULLUP); //按鈕輸入腳3(KEY_PUSH),內建上拉電阻
47     pinMode(buzzerPin, OUTPUT); //
48     // Serial.begin(9600); //串列傳輸啟動
49     randomSeed(analogRead(KEY_PUSH)); // (擷取亂數因子)解決亂數不亂的問題
50     //中斷腳位:以下將GIMSK中斷暫存器的INT0啟用,MCUCR控制暫存器設為負緣觸發
51     pinMode(2, INPUT_PULLUP); //紅色按鈕,外部中斷腳位
52     sei(); //enabling global interrupt
53     GIMSK |= (1<<INT0); // INT0啟用
54     MCUCR |= (1<<ISC01); // MCUCR控制暫存器設為負緣觸發
55     //亂數測試燈號3次
56     for(byte i=0;i<=3;i++){
57         LED_Random(); //按鈕尚未觸發,則顯示"亂數燈(彩色燈)"
58     }
59     //慢速跑馬
60     period = RingSpeed[0];
61     Time_Get = millis(); //擷取當前時間
62 }
63
64 //-----06 進入無窮迴圈 1.等待綠按鈕被觸發 2.執行"亂數燈(彩色燈)"-----
65 void loop() {
66     if (GameState == 0){ // 如果第0關-->等待按鈕(且)Go_Play == false遊戲等待中
67         if (digitalRead(KEY_PUSH) == LOW && Go_Play == false) {
68             delay(1000); //除火花,延遲 1000 毫秒,也就是 1 秒鐘。
69             PlayStar(); //奏音樂
70             led_run_at = 0; //跑馬燈現在位置
71             Go_Play = true; //遊戲開始
72             GameState=1; //開始玩第1關
73         }
74     }
75 }

```

```

73 //---慢速跑馬-----
74 if (millis()-Time_Get > period){ //延時 (亮燈週期過了,再重設週期)
75     Time_Get = millis(); // 擷取當下時間放入Time_Get
76     LED_Runing(); //呼叫跑馬燈
77 }
78 //-----
79 }
80 if (GameState == 1){Level_1();} //往第1關遊戲(每關4種變速)
81 if (GameState == 2){Level_2();} //往第2關遊戲(每關4種變速)
82 if (GameState == 3){Level_3();} //往第3關遊戲(每關4種變速)
83 }
84
85 //-----07 第Level=1關(3顆設定燈):-----
86 void Level_1(){
87     RingLED.clear(); //清除燈號
88     led_run_at=0; //跑馬燈歸零
89     //-----第1關遊戲迴圈
90     while (GameState == 1) {
91         period = RingSpeed[Speed]; //取速度值陣列內容
92         //---由燈數中,隨機取1顆--->再延伸左右各1形成3個燈
93         if (millis()-Time_Get > period){ //延時 (亮燈週期過了,再重設週期)
94             Time_Get = millis(); // 擷取當下時間放入Time_Get
95             if (Go_Play==true){ //遊戲中,可以取設定點(亂數)
96                 Setting_Led = random(NUMLEDS); //隨機取設定點
97                 Go_Play = false; //隨機設定號燈完畢,等待按鈕!
98             }
99             //----由隨機取1點,再延伸為3個點-----
100             //以下亮3個號燈(以16個燈為例:處理跨界14-15-0)
101             Set_Led[0]=Setting_Led; //陣列0=放入亂數值
102             if (Setting_Led == (NUMLEDS-2)){ //第14顆:(若16顆為例)
103                 Set_Led[1]=(NUMLEDS-1); //第15顆
104                 Set_Led[2]=0; //第0顆
105             } //以下亮3個號燈(以16個燈為例:處理跨界15-0-1)
106             else if (Setting_Led == (NUMLEDS-1)){
107                 Set_Led[1]=0;
108                 Set_Led[2]=1;
109             }
110             else{
111                 Set_Led[1]=Set_Led[0]+1;
112                 Set_Led[2]=Set_Led[0]+2;
113             } //---點亮3個燈[][][]---
114             RingLED.setPixelColor(Set_Led[0], 0, 255, 0); // (綠色)
115             RingLED.setPixelColor(Set_Led[1], 0, 255, 0); // (綠色)
116             RingLED.setPixelColor(Set_Led[2], 0, 255, 0); // (綠色)
117             RingLED.show(); //顯示設置燈
118             LED_Runing(); //呼叫跑馬燈
119             Check_Push(GameState); //檢查按鈕是否被按下?
120         }
121     }
122
123 //-----08 第Level=2關(2顆設定燈):-----
124 void Level_2(){
125     RingLED.clear(); //清除燈號
126     led_run_at=0; //跑馬燈歸零
127     //-----第2關遊戲迴圈
128     while (GameState == 2) {
129         period = RingSpeed[Speed]; //取速度值陣列內容
130         //---由燈數中,隨機取1顆--->再延伸左右各1形成2個燈
131         if (millis()-Time_Get > period){ //延時 (亮燈週期過了,再重設週期)
132             Time_Get = millis(); // 擷取當下時間放入Time_Get
133             if (Go_Play==true){ //Go_Play為true,可以取設定點(亂數)
134                 Setting_Led = random(NUMLEDS);
135                 Go_Play = false; //隨機設定號燈完畢,等待按鈕!
136             }
137             //---設定2個點[][]-----
138             //====以下亮2個號燈(以16個燈為例:處理跨界15-0)====
139             Set_Led[0]=Setting_Led; //陣列0=放入亂數值
140             if (Setting_Led==(NUMLEDS-1)){ Set_Led[1]=0; }
141             else{ Set_Led[1]=Set_Led[0]+1; }
142             RingLED.setPixelColor(Set_Led[0], 0, 255, 0); // (綠)
143             RingLED.setPixelColor(Set_Led[1], 0, 255, 0); // (綠)
144             RingLED.show(); //顯示設置燈
145             LED_Runing(); //呼叫跑馬燈

```

```

146         Check_Push(GameState); //檢查按鈕是否被按下?
147     }
148 }
149 }
150
151 //-----09 第Level=3關(1顆設定燈):-----
152 void Level_3(){
153     RingLED.clear(); //清除燈號
154     led_run_at=0; //跑馬燈歸零
155     //-----第3關遊戲迴圈
156     while (GameState == 3) {
157         period = RingSpeed[Speed]; // (陣列中的數字愈大愈慢)
158         //=====取亂數置入設定點=====
159         if (millis()-Time_Get > period){ //延時 (亮燈週期過了,再重設週期)
160             Time_Get = millis(); // 擷取當下時間放入Time_Get
161             //====亂數設定OK了嗎? Yes(改false)
162             if(Go_Play==true){ //Go_Play為true,可以取設定點(亂數)
163                 Setting_Led = random(NUMLEDS);
164                 Go_Play = false; //隨機設定號燈完畢,回歸,等待按鈕!
165             }
166             //---設定2個點[][]-----
167             //====以下亮1個號燈====
168             Set_Led[0]=Setting_Led;
169             RingLED.setPixelColor(Set_Led[0], 0, 255, 0); // (綠)
170             RingLED.show();
171             LED_Runing(); //呼叫跑馬燈
172             Check_Push(GameState); //檢查按鈕是否被按下?
173         }
174     }
175     //遊戲結束--亂數測試燈號3次
176     for(byte i=0;i<=3;i++){
177         LED_Random(); //按鈕尚未觸發,則顯示"亂數燈(彩色燈)"
178     }
179     period = RingSpeed[0];
180     Time_Get = millis(); //擷取當前時間
181 }
182
183 //-----10 跑馬燈函式 -----
184 void LED_Runing(){
185     led_run_at++; //循環燈...往下累加
186     if (led_run_at >= NUMLEDS) { //如果循環紅點位置=燈數(符合則歸零)
187         RingLED.setPixelColor(led_run_at-1, 0, 0, 0); //最後一顆歸0
188         led_run_at = 0; //歸零
189     }
190     RingLED.setPixelColor(led_run_at, 255, 0, 0); //亮燈
191     RingLED.setPixelColor(led_run_at-1, 0, 0, 0); //往後關燈
192     RingLED.show();
193 }
194
195 //-----11 檢查按鈕是被按(檢查是否得分?)-----
196 void Check_Push(byte LV){ //Go_Play == false為等待的意思
197     if (digitalRead(KEY_PUSH) == LOW && Go_Play == false ) { //按鈕被按下去,往下做
198         delay(1000); //延時
199         Go_Play=true;
200         //-----得分-----
201         if(LV==1){ //第1關:3個燈選中其1,則得分
202             if (Set_Led[0]==led_run_at || Set_Led[1]==led_run_at || Set_Led[2]==led_run_at) {
203                 Score++;
204                 ScreLed(Score,550); //呼叫計分燈號函式(累計得分,高音頻)
205             }
206             else{ScreLed(Score,150);} //呼叫計分燈號函式(不累計,低音頻)
207         }
208         else if(LV==2){ //第2關:2個燈選中其1,則得分
209             if (Set_Led[0]==led_run_at || Set_Led[1]==led_run_at) {
210                 Score++;
211                 ScreLed(Score,550); //呼叫計分燈號函式(累計得分,高音頻)
212             }
213             else{ScreLed(Score,150);} //呼叫計分燈號函式(不累計,低音頻)
214         }
215         else if(LV==3){ //第3關:1個燈選中,則得分
216             if (Set_Led[0]==led_run_at) {
217                 Score++;
218                 ScreLed(Score,550); //呼叫計分燈號函式(累計得分,高音頻)

```

```

219     }
220     else{ScreLed(Score,150);} //呼叫計分燈號函式(不累計,低音頻)
221 }
222     //---切換速度陣列[ ][ ][ ][ ]-----
223     Speed++; //換下個檔速
224     //若4種檔速完成,換下一關
225     if(Speed==4){
226         //Serial.println("-");
227         GameState++; //下一關
228         if(GameState==4){GameState=0; Go_Play=false;Score=0;} //3大關全部完成
229         Speed=0;
230         PlayFamily();
231     }
232 }
233 }
234
235 //----12 計分燈號(黃燈)+鳴(函式)----
236 void ScreLed(byte Gain,int Freq){ //接收(得分值,音頻)
237     RingLED.clear(); //清除燈號
238     for(byte i=0;i<Gain;i++){RingLED.setPixelColor(i, 255 , 255, 0);}
239     RingLED.show(); //秀"黃燈數"
240     //--鳴--
241     for (byte j = 0; j <2; j++){
242         tinyTone(Freq, 400);}
243     delay(450);
244     RingLED.clear(); //清除燈號
245 }
246 //-----
247 //----13 秀"亂數燈(彩色燈)"----
248 void LED_Random(){
249     for (byte i = 0; i < NUMLEDS; i++) { //環燈號訊號
250         RingLED.setPixelColor(i, random(0, 255), random(0, 255), random(0,
251             255)); //0~255隨機取數
252         delay(15);
253         RingLED.show();
254     }
255 }
256 //----14 演奏"星際爭霸"音樂----
257 void PlayStar(){
258     int Star[] =
259     {440,440,440,349,523,440,349,523,440,659,659,659,698,523,415,349,523,440};
260     int StarDur[] = {400, 400, 400, 250, 50, 400, 250, 50, 900, 400, 400, 400, 250, 50,
261         400, 250, 50, 900};
262     for (int thisNote = 0; thisNote < 18; thisNote++) { //18個音符
263         tinyTone(Star[thisNote],StarDur[thisNote]);
264         delay(30); // 音符之間的間隔
265     }
266 }
267 //----15 演奏"全家音樂"----
268 void PlayFamily(){
269     int FamilyMusic[]={330, 262, 196, 262, 294, 392, 0, 0, 294, 330,0, 294, 196, 262,0};
270     for (int thisNote = 0; thisNote < 15; thisNote++) { //14個音符
271         tinyTone(FamilyMusic[thisNote],200);
272         delay(30); // 音符之間的間隔
273     }
274 }
275 //----16 外部中斷PB2腳(紅)按鈕被觸發1.白光 2.閃白光 3.閃紅光 4.閃紅光+鳴 5.軟體重啟
276 ISR (INT0_vect){
277     if ( millis() - IntTime_Get > Int_period ){
278         IntTime_Get = millis();
279         Int_state++; //累計按下次數
280         //Serial.print(Int_state);
281         while(Int_state==1){ //按1次-->手電筒模式
282             for (byte i = 0; i < NUMLEDS; i++) { //環燈號訊號
283                 RingLED.setPixelColor(i,255,255,255); //白色
284                 RingLED.setBrightness(180); //環燈亮度
285                 RingLED.show();
286             }
287         }
288         while(Int_state==2){ //按2次-->閃白光模式

```

```

289         for (byte i = 0; i < NUMLEDS; i++) { //環燈號訊號
290             RingLED.setPixelColor(i,255,255,255); //白色
291             RingLED.setBrightness(180); //環燈亮度
292             RingLED.show();
293         }
294         delay(400);
295         for (byte i = 0; i < NUMLEDS; i++) { //環燈號訊號
296             RingLED.setPixelColor(i,0,0,0); //黑色
297             RingLED.show();
298         }
299         delay(400);
300     }
301     while(Int_state==3){ //按3次-->閃紅光模式
302         for (byte i = 0; i < NUMLEDS; i++) { //環燈號訊號
303             RingLED.setPixelColor(i,255,0,0); //紅色
304             RingLED.setBrightness(180); //環燈亮度
305             RingLED.show();
306         }
307         delay(400);
308         for (byte i = 0; i < NUMLEDS; i++) { //環燈號訊號
309             RingLED.setPixelColor(i,0,0,0); //黑色
310             RingLED.show();
311         }
312         delay(400);
313     }
314     while(Int_state==4){ //按4次-->閃紅光+警示音模式
315         for (byte i = 0; i < NUMLEDS; i++) { //環燈號訊號
316             RingLED.setPixelColor(i,255,0,0); //紅色
317             RingLED.setBrightness(180); //環燈亮度
318             RingLED.show();
319         }
320         tinyTone(650, 200);
321         delay(300);
322         for (byte i = 0; i < NUMLEDS; i++) { //環燈號訊號
323             RingLED.setPixelColor(i,0,0,0); //黑色
324             RingLED.show();
325         }
326         tinyTone(900, 200);
327         delay(300);
328     }
329     if(Int_state==5){ //按5次-->軟體重啟模式
330         Int_state=0;
331         resetFunc();
332     }
333 }
334 }
335
336 //----17 音樂播放函式(傳音符[頻率],聲音長度)
337 void tinyTone(int frequency, int length){
338     unsigned long period;
339     unsigned long n;
340     period=1000000/frequency;
341     n=1000*(long)length/period;
342     for(int j=1;j<n;j++){
343         digitalWrite(buzzerPin,HIGH);
344         delayMicroseconds(period/2);
345         digitalWrite(buzzerPin,LOW);
346         delayMicroseconds(period/2);
347     }
348 }
349

```