

(第二冊)

洪國勝、蔡懷文、王丹君 編著

國立高雄師範大學工業科技教育系 溫嘉榮教授推薦



目錄

第一冊

第1單元	準備工作	1
第2單元	電腦運算思維與資料數位化	23
第3單元	資料的運算	
第4單元	霹靂燈	53
第5單元	8*8 點陣 LED	63
第6單元	跑馬燈與告白板	69
第7單元	人工智慧的開始 - 比較運算子、邏輯運算子、	
	決策指令	80
第8單元	任勞任怨的迴圈指令 -for 與 while	105

第二冊

第9單元	七段顯示器與擲骰子	119
第10單元	指撥開關與表決器	134
第 11 單元	按壓開關與叫號器	149
第 12 單元	『基本資訊應用』組試題解答	158
第 13 單元	博奕程式與倒數計時器	174
第 14 單元	搶答器與選秀表決器	180
第 15 單元	蜂鳴器與電子琴	187
第 16 單元	變頻原理與變頻調光器	196
第 17 單元	可變電阻與電子調光器	201
第 18 單元	光敏電阻與小夜燈	204
第 19 單元	四位數七段顯示器	207
第 20 單元	電子時鐘	222
第 21 單元	溫濕度計	229
第 22 單元	紅外線遙控器	233
第 23 單元	電子琴教學機的製作	238
第 24 單元	智慧電子琴	242





因應教育部科技領域課綱「科技領域之教學,實作活動時數 官占整體課程時數的二分之一至三分之二」,翻閱目前中學生「生 活科技」課本,課綱內容非常精彩目豐富,包含科技的本質、機 構的繪圖、設計與製作、材料的選擇與加工、木工機具的保養與 操作、能源與發電動力機械的原理與操作、電子電路的原理與操 作、新興科技的發展與操作等。木工與手工具還停留在40年前 工藝課的操作,當時我們的工藝課可以製作木工、鐵管小椅子、 書架,拿來當作學校週會或家裡用。但現在環境不同了,塑膠一 體成形的家具很便宜,所以已經不流行 DIY,連帶家庭常備的木 工、鐵工的小機具也慢慢消失;能源與動力設備則太大,無法在 每個學校購置;電子電路原是不錯的科技實作選項,但目前已經 被單晶控制取代。以筆者資訊教學35年的經驗而言,本人大力推 廣機電整合的 Arduino 作為生活科技的實作課程。因為 Arduino 所占空間最小,設備費最便宜,最適合全班教學。可完成的生活 科技產品如字幕機、告白板、叫號器、電子鐘、計時器、溫度 計、紅綠燈、霹靂燈、遙控器、電子琴等,都與生活息息相關。 做完的產品可以帶回家用,可說是非常真實的生活科技之產品; 日本項操作沒有觸電、刀具運轉等危險性,沒有粉塵、也沒有噪 音,又可體驗與學習程式設計等運算思維。其次,我們也開發上 課用教具,而使用教具教學的優點如下:

- 教具可放在學校工場重複使用,這樣老師不用向學生收錢, 學生也不會因為忘了帶材料而影響進度,學生依序探索後, 可依自己的興趣,自己購置材料,完成所需專題。
- 2. Arduino 具有自我保護電路,學生插錯也不會壞。
- 操作電路電壓僅為 5V,任意觸摸、接錯電路等都不會有任何 危險。

筆者雖然程式教學與著作超過35年,快樂與熱情不變。儘 管不斷的重複讀稿,力求完善,但仍難免有疏漏及錯誤,尙祈各 位先進不吝指正,將不當或錯誤詞句回傳給我,如下表,本人不 勝感激。每年會選出與抽出精彩回函,寄發紀念品。本書感謝 高師大工教系學妹王丹君同學、學弟蔡懷文老師主動加入共筆創 作、完稿於蘆洲國中研習時,也感謝王漢卿老師主動幫忙改正 很多錯誤,在此致謝。其次,筆者也會在泉勝出版網站(www. goodbooks.com.tw)刊登本書勘誤表。

洪國勝、蔡懷文、王丹君 謹識

www.goodbooks.com.tw

	Arduino 程式設計									
頁數	行數	錯誤或不當詞句	建議修改為							
其他批評	平與建議:									

▶ 讀者意見、内文觀念與詞句修正回函表

以上意見、指正回函表,請郵寄至 aa163677@yahoo.com.tw,每 年會選出與抽出精彩回函,寄發紀念品。



推薦序

目前國民小學是採用圖形化介面的 Scratch 當作程式設計工 具,高中、大學、業界則採用文字化的程式設計介面,如 C/C++ 與 Python。國民中學就有點尷尬,到底要採用哪一種介面?看過 洪國勝老師的『Arduino 程式設計』,我也認同國中、高中生都應 該使用文字式程式設計,這樣才能快速連結產業需求。

用Arduino學程式

國中與高一學生因爲數學基礎還太少,老師教程式還要先教 數學,真的很費力,所以我也認同使用 Arduino 的 LED 實物來學 習迴圈與陣列,這樣更能激發與維持學生學習程式設計熱度。其 次,Arduino 是生活科技、資訊科技最好的補充教材。因為單晶 Arduino 是目前最新科技主流,簡單的電路與程式就可嵌入目前 工業與家用電器,進而簡化控制電路與節省開發的成本,如 ABS 防鎖死剎車、字幕機、電子琴、溫度計、遙控器、叫號器、電子 時鐘等。

目前校園常用的單晶控制晶片有 Macro:Bit、Arduino, Macro:Bit 比較著重在圖形開發介面,主要是來讓小學生玩簡單 的聲光控制遊戲; Arduino 則是文字形式的開發介面,也是真實 工業的嵌入式晶片,可真實改善許多生活與工業控制,適合中學 生以上與社會人士學習嵌入式系統控制。Arduino 之所以能異軍 突起,我認為是因為它主張開源,且軟硬體都是開放的,使用者 可以站在巨人的肩膀,繼續接力開發新產品。其次,Arduino 輸 出電流變大、腳位也變多,這樣可以簡化生活電器的四位數七段

(上) VI Arduino程式設計

與點陣 LED 顯示。既然單晶電路已經簡化、成本也降低,那麼 Arduino 程式設計也就更適合拿來當作國高中的生活科技課程的 實作教材,讓國中與高中生活科技也能與世界潮流平行接軌。學 生學的都是目前新興科技,而且都是簡單的接線與簡單的程式, 就可改善與實作目前生活科技產品,這樣的學習可說非常生動與 實用,當然可以激發學生自造更多創新科技產品。

洪國勝老師是我 76 年高師大工教系畢業高徒,國高中雲縣 與省賽工藝競賽科展優勝,所以立志讀工教系。大二全校程式設 計第一名、大四全國大學含研究所『微電腦應用創作』優勝第一 名,畢業分發高雄市立海青工商資訊科任教。退休後教學熱情不 變,自力成立『泉勝出版有限公司』,自費推廣國高中的科技與程 式教育,詳見泉勝出版網站。每本著作都是洪國勝老師累積超過 40 年實作與教學心得。所以,本人不斷推薦其 C/C++、Python、 APCS 與 Arduino 等程式設計著作,也在此『Arduino 程式設計』 寫公開序言推薦。

國立高雄師範大學工業科技教育系 溫嘉榮教授

大力推薦

2022/5/12

第9單元

七段顯示器與擲骰子

七段顯示器是由 8 個 LED 排列成「日」字, 8 個 LED 照理應有 16 隻腳,但為了簡化腳位(腳位簡化的優點是省元件的體積)與 配合低或高電位驅動,廠商是將所有 LED 的陽極或陰極共接再拉 出,所以分為共陽極或共陰極七段顯示器,兩者原理都相同。但 是,因為 Arduino 高電位的電流有 20mA,可以直接驅動 LED, 所以本書就介紹共陰極如下(共陰極的驅動電路比共陽極簡單, 這讀者以後就會明瞭):

共陰極七段顯示器

共陰極七段顯示器的內部電路與腳位圖如下圖左(大部分的共陰 與共陽七段顯示器腳位排列都相同)。

測量檢驗方式

- 請自行使用傳統指針型三用電表測量,先撥歐姆檔×10,例 如黑棒(+)接a,紅棒(-)接com點,則上面的a棒亮。
- 2. 或用附錄 A 自製的接點連通計,正極接 a,負極接 com,那 a 棒亮。



本書教學實驗板並未準備一位數七段顯示器,請將八位數七段 顯示器的最右邊的 com4 接地,那就可以當作一位數七段顯示器 用,如下圖,其次,a,b,c,d,e,f,g,dp的腳位我已經按順序由左 而右排列。



以下程式,可讓8個LED全亮。

```
void setup() {
  DDRF=0xFF; //指派PORTF腳位功能是輸出
}
void loop() {
  PORTF=0xFF; //指派PORTF腳位電壓為高電位,電壓5V
}
```

其次,若使用內部接線的實驗版,其電路同第20單元,如下圖:

(h) **120**



「下我們加入

以下我們加入以下底線字的程式,以軟體設定的方式,僅讓最右邊的 com4 為低電位,其餘均為高電位,這樣功能也同上,本書 往後程式,也都加上這兩行,這樣就可將以上 8 位數七段顯示 器,拿來當作一位數使用。

```
void setup() {
    DDRF=0xFF; //指派PORTF腳位功能是輸出
    DDRK=0xFF; //指派PORTK腳位功能是輸出
    PORTK=B1111110; //0xFE,僅最右邊一隻腳com4為低電位
    //DDRK=0xFF; PORTK=0xFE; //將四位數七段顯示器當作一位數
}
void loop() {
    PORTF=0xFF; //指派PORTF腳位電壓全為高電位,電壓5v
}
```

請鍵入以下程式,並觀察此七段顯示器 a,b,c,d,e,f,g,dp 等 LED 的 排列。

```
void setup() {
    DDRF=B11111111;
    DDRK=0xFF;PORTK=0xFE;//將四位數七段當作一位數
}
void loop() {
    byte a[]={0x01,0x03,0x07,0x0f,0x1f,0x3f,0x7f,0xFF,0};//8
```

```
byte b[]={0x01,0x02,0x04,0x08,0x10,0x20,0x40,0x80,0};//
 byte c[]={0x3f,0x6,0x5b,0x4f,0x66,0x6d,0x7d,0x7,0x7f,
            0x6f,0};//10
  for (byte i=0;i<=8;i++) {</pre>
      PORTF=a[i];
      delay(300);
  }
  for (byte i=0;i<=8;i++) {</pre>
      PORTF=b[i];
      delay(500);
  }
  for (byte i=0;i<=10;i++) {
      PORTF=c[i];//0,1,2,3,4,5,6,7,8,9,空白
      delay(800);
  }
}
```

由以程式可知,共陰極要顯示0到9,所要輸出的電壓如下表, 1代表亮,0代表不亮。例如,要顯示0,則a,b,c,d,e,f等腳位 要給高電壓(亮),g與dp要給低電位(不亮),所以資料要輸出 $0x3F \circ$

數字	dp	g	f	е	d	С	b	а	十六進制 數値
0	0	0	1	1	1	1	1	1	0x3F
1	0	0	0	0	0	1	1	0	0x6
2	0	1	0	1	1	0	1	1	0x5B
3	0	1	0	0	1	1	1	1	0x4F
4	0	1	1	0	0	1	1	0	0x66
5	0	1	1	0	1	1	0	1	0x6D
6	0	1	1	1	1	1	0	1	0x7D
7	0	0	0	0	0	1	1	1	0x7
8	0	1	1	1	1	1	1	1	0x7F
9	0	1	1	0	1	1	1	1	0x6F

(<u>____</u>)122

例如,以下程式可以重複循環顯示 0,1,2 各一秒鐘。

```
void setup() {
    DDRF=B1111111;
    DDRK=0xFF;PORTK=0xFE;//將四位數七段當作一位數
}
void loop() {
    PORTF=0x3f;
    delay(1000);
    PORTF=0x6;
    delay(1000);
    PORTF=0x5b;
    delay(1000);
}
```

範例 7a

查表顯示數字。

₩ 運算思維

前面我們要顯示 0,就輸出 0x3f;要顯示 1,就輸出 0x6;要顯示 2 就輸出 0x5b,這樣程式有點冗長,因為若要顯示 10 個數字,那 程式更長了,所以通常將以上數字以陣列存放如下:

```
byte a[]={0x3f,0x6,0x5b,0x4f,0x66,0x6d,0x7d,0x7,0x7f,0x6f};
```

那只要指派陣列索引,就可以得到對應輸出碼,此即為陣列索引 查表功能。例如,要顯示1,程式如下:

PORTF=a[1]; delay(1000);

要顯示 5,程式如下:

```
PORTF=a[5];
delay(1000);
```

(合 124 Arduino程式設計

以下程式,配合迴圈,就可以重複顯示0到9各1秒。

```
int a[]={0x3f,0x6,0x5b,0x4f,0x66,0x6d,0x7d,0x7,0x7f,0x6f};
int i=0;
void setup() {
    DDRF=B11111111;
    DDRK=0xFF;PORTK=0xFE;//將四位數七段當作一位數
}
void loop() {
    PORTF=a[i]; //PORTF
    delay(1000);
    i=(i+1) %10; //共10個
}
```

♡自我練習

1. 同上題,那如何於兩個數字之間,顯示一個空白呢?

範例 7b

連續數字的顯示。例如,顯示學號是 825710。

₩運算思維

 本例假設學號是 825710,那要分別送出 825710 對應的數值 0x7f, 0x5b, 0x6d, 0x7, 0x6, 0x3f,所以將以上資料放在陣列 如下:

byte b[]={0x7f,0x5b,0x6d,0x7,0x6,0x3f};

2. 根據上一範例,將b陣列指派給 PORTF 即可,所以程式如下:

```
byte b[]={0x7f,0x5b,0x6d,0x7,0x6,0x3f};
byte len=6;
byte i=0;
void setup() {
```

```
DDRF=B11111111;
DDRK=0xFF;PORTK=0xFE;//將四位數七段當作一位數
}
void loop() {
PORTF= b[i];
delay(1000);
i=(i+1) %len;
}
```

 前面程式是自己將 825710 查表,轉為七段顯示器的輸出碼, 其實應該讓電腦自己查表,這樣程式才會簡單。例如,將 825710 放在陣列,如下:

```
byte b[]={8,2,5,7,1,0};
```

 然後程式如下,此稱為陣列中的陣列。(善用陣列,您的程式 才會虎虎生風)

PORTF= a[b[i]];

例如,i=0,b[0]=8,那a[8]=0x7F,也就是送出0x7f,那就 顯示『8』。

5. 全部程式如下:

```
byte a[]={0x3f,0x6,0x5b,0x4f,0x66,0x6d,0x7d,0x7,0x7f,0x6f};
byte b[]={8,2,5,7,1,0};
byte len=6;
byte i=0;
void setup() {
    DDRF=B1111111;
    DDRK=0xFF;PORTK=0xFE;//將四位數七段當作一位數
}
void loop() {
    PORTF= a[b[i]];
    delay(1000);
    i=(i+1) %len;
}
```

(20126 Arduino程式設計

 有時候若是數字相同,例如,88255,每顯示完一個數字,若 沒有先熄掉,再輸出下一個數字,那看起來會是 825,所以 程式改善如下:

```
void loop() {
    PORTF= a[b[i]];
    delay(800);
    PORTF=0;
    delay(200);
    i=(i+1) %len;
}
```

7. 也可以將資料以字串型態儲存,全部程式如下:

```
byte a[]={0x3f,0x6,0x5b,0x4f,0x66,0x6d,0x7d,0x7,0x7f,0x6f};
String b="825710";byte len=b.length();
void setup() {
    DDRF=B11111111;    DDRK=0xFF;PORTK=0xFE;//將四位數七段當作一位數
}
byte i=0;
void loop() {
    PORTF= a[byte(b[i])-48];//逐一轉字元
    //請留意每一字元逐一轉爲byte爲其Ascii値,所以要減48
    delay(1000); i=(i+1) %len;
}
```

₩ 補充說明

 程式設計的領域,很多都是重複循環,請多利用陣列索引從 0開始的特性,且配合取餘(%)運算,這樣才能簡化程式的 撰寫。其次,陣列索引從0開始,只是給使用者方便,不用 0開始也沒關係。

₩ 自我練習

 請將今天的日期,用一個七段顯示器逐一顯示。提示:先將0 到9、"-"(0x40)以陣列存放,『"-"(0x40)』放在索引10,如下: byte a[]={0x3f,0x6,0x5b,0x4f,0x66,0x6d,0x7d,0x7,0x7f,0x6f,0x40}; 假設今天的日期是『2019-04-26』,也以陣列存放如下:(10 就會對應到 0x40,將顯示 "-"。)

byte b[]={2,0,1,9,10,0,4,10,2,6};

請將現在的時間,以陣列儲存,並用一個七段顯示器逐一顯示。

範例 9c

如何輸出1個四位數整數。例如, int a=1028, 那程式如何撰寫?

🗑 思考步驟

您要發揮國小數學能力,充分利用整數除法與取餘運算,將
 1個4位數,逐一分解成4個位數。程式如下:

```
int c=1028;//1028已經超過255,所以要宣告為int
d=c/1000;//千
c=(c-d*1000);
d=c/100;//百
c=c-d*100;
d=c/10;//十
d=c%10;//個位數
```

再將此四位數一一查表,找出對應輸出碼(陣列的查表功 能),再輸出,全部程式如下:

```
void setup() {
    DDRF=B1111111;
    DDRK=0xFF;PORTK=0xFE;
    Serial.begin(9600);
}
void loop() {
```

(<u>128</u>) Arduino程式設計

}

```
byte a[]={0x3f,0x6,0x5b,0x4f,0x66,0x6d,0x7d,0x7,0x7f,0x6f,0x40};
int c=1028;//1028已經超過255,不能宣告為byte, 要宣告為int
byte d=c/1000;//整數除法,得到千位數
PORTF=a[d];//顧示千位數
delay(1000);
c=(c-d*1000);
d=c/100;//百位數
PORTF=a[d];//顯示百位數
delay(1000);
c=c-d*100;
d=c/10;//十位數
PORTF=a[d];//顏示十位數
delay(1000);
d=c%10;//個位數
PORTF=a[d];//顏示個位數
delay(1000);
```

2. 以上是數字長度固定,題目限定四位數,所以應該善用迥 圈,才能簡化程式,因為 for 迴圈通常用在題目設計階段就 已經知道迴圈次數,所以程式如下:

```
byte a[]={0x3f,0x6,0x5b,0x4f,0x66,0x6d,0x7d,0x7f,0x6f};
void setup() {
   DDRF=B11111111;
   DDRK=0xFF; PORTK=0xFE; //將四位數七段當作一位數
   Serial.begin(9600);
}
void loop() {
  int c=1028;//1028已經超過255,不能宣告為byte, 要宣告為int
  int d;
 int e=1000;
  for (int i=3;i>=0;i--) {
   d=c/e;
   Serial.println(d);
    c=c-d*e;
   PORTF=a[d];
   e=e/10;
   delay(500) ;
 }
}
```

4. 以下程式就不行,因為 Arduino 的實數運算有誤差。

```
void loop() {
    int c=1028;//1028已經超過255,不能宣告爲byte,要宣告爲int
    int d;
    int e;
    for (int i=3;i>=0;i--){
        e=pow(10,i);
        Serial.println(e);
```

☑ 自我練習

- 1. 同上題,數字出現後,請閃爍一下。
- 同上題,可以顯示3位數,且含有1個小數點。例如, 103.2。
- 於單向紅綠燈的系統中(假設僅有紅黃綠三個燈,且秒數均 小於10),請加上紅燈與綠燈倒數秒數。

擲骰子

電腦常常需要模擬一些執行結果,例如,樂透開獎或擲骰子,此時就需要產生亂數,Arduino產生亂數的方法是使用 random()函式,其語法如下:

random(min,max);

產生含 min(含)與 max(不含)之間的整數亂數,例如,以下 程式可產生 10 到 19 的整數亂數。

int r = random(10, 20);//10..19,不含20

其中 min 若省略,則視為0,例如,以下程式可產生0到299的 整數亂數。

int r = random(300); //0..299

(二)130 Arduino程式設計

其次,因為亂數的產生是一種查表的動作,為了讓每次都能有不同的起始點,那就要使用 randomSeed()函式。例如,以下程式將A0 腳位空接,即可隨機讀取一個雜訊,且將此雜訊作為一個不同 亂數起點的依據。

randomSeed(analogRead(A0));//用0或AO都表示AO腳位

範偷 9d

示範以上程式。

🗑 程式列印

```
void setup(){
   Serial.begin(9600);
   randomSeed(analogRead(A0));//以雜訊作爲產生不同亂數起點的依據
}
void loop() {
   int r;
   r = random(10, 20); // produce a random number from 10 to 19
   Serial.println(r);
   delay(1000);
   r = random(300); // produce a random number from 0 to 299
   Serial.println(r);
}
```

☆自我練習

請將以上範例的 randomSeed(analogRead(A0)) 去掉,並觀察執行 結果,請留意每次執行程式時(『將序列埠監控視窗』關閉,再重 新開啓序列埠視窗,即可重新執行程式),其亂數順序是否都相 同。

擲骰子遊戲

有了 random() 函式,就可完成擲骰子、撲克牌、樂透開獎等程式。

範例 9e

示範產生3個亂數,且同時於序列埠視窗與七段顯示器輸出。

🖓 程式列印

```
//PORTF A0,A1…A7 連接到seven-segment a,b,c,d,e,f,g,dp
byte a[]={0x3f,0x6,0x5b,0x4f,0x66,0x6d,0x7d,0x7,0x7f,0x6f};
void setup() {
  Serial.begin(9600);
  randomSeed(analogRead(0));
  DDRF=B11111111;
  DDRK=0xFF; PORTK=0xFE; //將四位數七段當作一位數
}
void loop() {
  int r;
  r = random(1, 7); // produce a random number from 1 to 6
  Serial.println(r);
 PORTF=a[r];
  delay(1000);
  r = random(1, 7);
  Serial.println(r);
  PORTF=a[r];
  delay(1000);
  r = random(1, 7);
  Serial.println(r);
  PORTF=a[r];
  delay(1000);
  delay (1000);
}
```

(合 132 Arduino程式設計

但是,數字很容易重複,不易觀察,所以應該先熄掉,再顯示下 一個數字,程式如下:

```
//PORTF A0,A1…A7 連接到seven-segment a,b,c,d,e,f,g,dp
byte a[]={0x3f,0x6,0x5b,0x4f,0x66,0x6d,0x7d,0x7,0x7f,0x6f};
void setup() {
  Serial.begin(9600);
  randomSeed(analogRead(0));
  DDRF=B11111111;
  DDRK=0xFF; PORTK=0xFE; //將四位數七段當作一位數
}
void loop() {
  int r;
  r = random(1, 7); // produce a random number from 1 to 6
  Serial.println(r);
  PORTF=a[r];
  delay(1000);
  PORTF=0;
  delay(200);
  r = random(1, 7);
  Serial.println(r);
  PORTF=a[r];
  delay(1000);
  PORTF=0;
  delay(200);
  r = random(1, 7);
  Serial.println(r);
  PORTF=a[r];
  delay(1000);
  PORTF=0;
  delay(200);
  delay (1000);
}
```

for迴圈

上一範例,產生3次亂數,這3次亂數的程式其實相同,電腦有 一個指令 for,它可以簡化程式的撰寫,所以我們可以用一個 for 迴圈替代,如以下程式:

```
//setup() 同上
void loop() {
    int r;
    for (byte i=1;i<=3;i++) {
        r = random(1, 7);// print a random number from 1 to 6
        Serial.println(r);
        PORTF=a[r];
        delay(1000);
        PORTF=0;
        delay(200);
    }
    delay (1000);
}
```

閃爍

連續明滅稱為閃爍,以下程式,我們再用一個j迴圈讓它連續明滅2次。這樣迴圈裡面有迴圈,就稱為巢狀迴圈。

```
//setup()同上
void loop() {
  int r;
  for (byte i=1;i<=3;i++) {</pre>
    r = random(1, 7); // print a random number from 1 to 6
    Serial.println(r);
    PORTF=a[r];
    delay(600);
    for (byte j=1 ;j<=2;j++) {</pre>
      PORTF=a[r];
      delay(100);
      PORTF=0;
      delay(100);
    }
    delay(200);
  }
  delay (1000);
}
```

第10單元

指撥開關與表決器

前面的 LED 與七段顯示器都是輸出設備,本單元起要開始介紹一些輸入設備,例如,指撥開關與按壓開關。

數位輸入

Arduino 當輸入時,有兩種可能,一種是接收其他 IC 的輸出,這 時只要設定其為 INPUT 即可,但若要接收按壓開關或指撥開關等 開關裝置,則可設定其具有上拉電阻 (INPUT_PULLUP),這樣可 以簡化外部電路。開關的標準電路如下圖左,此 20kΩ 稱為上拉 電阻 (使用 10~50k 都可以),按鍵沒按壓是高電位,壓下去是低 電位,這樣就可以判斷開關有沒有被按。其次,Arduino 為了讓 使用者簡化電路,此上拉電阻可用軟體指派,所以使用者只要接 一個開關就好,如下圖右。(補充說明,沒上拉電阻可以嗎?當然 不可以,因爲開關沒按壓時,Arduino 輸入腳位浮接(懸空未接 稱爲浮接),此時就無法判定其電壓的高與低了。



指撥開關

指撥開關的內部結構如下圖。開關可上下滑動,下圖左是下滑, 電路處於斷路,下圖中往上滑動,則電路接通。往後為了簡化電 路,均以下圖右表示指撥開關。



範例 10a

示範使用指撥開關。

₩操作步驟

- 1. 完成下圖左電路,其中 LED 同上範例 2a。
- 2. 指撥開關的實驗板實體圖如下圖右,杜邦線從J5連接到 Arduino 微控板對應腳位即可, Gnd 內部已經連接。



(二)136 Arduino程式設計

digitalRead()一次僅讀取一個指定腳位電壓,請鍵入以下程式,並觀察執行結果。請留意開關 ON,其值是 LOW, LED 滅掉。

```
void setup() {
    Serial.begin(9600);
    pinMode(49,INPUT_PULLUP);//PL0
    DDRB=B1111111;PORTB=0;
    DDRC=0xFF;PORTC=0xFE;//0xFE is B1111110 將點陣LED當作8個LED使用
}
void loop() {
    byte b=digitalRead(49);//讀一個位元
    Serial.println(b);
    digitalWrite(53,b); //寫入一個位元
}
```

本例指撥開關接到 PORTL,可以用 PINL 讀取 PORTL 暫存器 8 隻腳全部電壓。請鍵入以下程式,並觀察執行結果,請留意開關 ON,其值是 LOW,LED 滅掉。

```
void setup() {
    Serial.begin(9600);
    pinMode(42,INPUT PULLUP);//PL7
    pinMode(43, INPUT PULLUP);//PL6
    pinMode(44, INPUT PULLUP);//PL5
    pinMode(45, INPUT PULLUP);//PL4
    pinMode(46,INPUT PULLUP);//PL3
    pinMode(47, INPUT PULLUP);//PL2
    pinMode(48,INPUT_PULLUP);//PL1
    pinMode(49, INPUT PULLUP);//PL0
    DDRB=B11111111; PORTB=0x00;
    DDRC=0xFF; PORTC=0xFE; //0xFE is B11111110 將點陣LED當作8個LED使用
}
void loop() {
    byte b=PINL;//讀8個位元
    Serial.println(b);
    PORTB=b;
}
```

5. 請將以上程式的 8 個 INPUT_PULLUP 通通改為 INPUT,程 式如下,並觀察執行結果。

pinMode(37,INPUT);//PC0

 指撥開關通常可用來作開關,或運作模式的選擇,請看以下 範例說明。

範例 10b

開關控制。指撥開關可以用來當開關,以下電路使用3位元指 撥開關,分別控制3個LED的明滅,且ON時才亮。

₩操作步驟

- 1. 請完成以下電路。
- 2. 使用 digitalRead() 函式,每次讀取1位元,且逐一控制對應 LED,程式如下:

```
void setup() {
   Serial.begin(9600);
   pinMode(47,INPUT PULLUP);//PL2
   pinMode(48,INPUT PULLUP);//PL1
   pinMode(49,INPUT PULLUP);//PL0
   DDRB=0xFF;
                PORTB=0;
   DDRC=0xFF; PORTC=0xFE; //將點陣LED當作8個LED使用
}
void loop() {
   byte b;
   b=digitalRead(47);//讀一個位元
   Serial.println(b);//輸出看看,確認是否正確
   if (b==0) / /== 比較是否相等運算子
     digitalWrite(51, HIGH); //用常數符號表示高電位5V
   else
     digitalWrite(51,LOW); //表示低電位OV
   b=digitalRead(48);//讀一個位元
```

}

```
Serial.println(b);//輸出看看,確認是否正確
if (b==0) //== 是比較是否相等運算子
 digitalWrite(52, HIGH); //用常數符號表示高電位5V
else
 digitalWrite(52,LOW); //表示低電位OV
b=digitalRead(49);//讀一個位元
Serial.println(b);//輸出看看,確認是否正確
if (b==0) / /== 是比較是否相等運算子
 digitalWrite(53,HIGH); //用常數符號表示高電位5V
else
 digitalWrite(53,LOW); //表示低電位OV
```

3. 以上我們用腳位編號寫程式,但有可能您要改變腳位,此時 您就要到處修改,萬一沒有完全修改,那就造成程式執行不 一致,所以我們通常將這些腳位,通通在程式最前面,用常 數符號(以下簡稱常數)代替,這樣當您要改變腳位時,只 要在最前面修改就行,不用到處修改。現在,將以上腳位定 義常數,程式如下:(這是單晶程式的習慣,因為您有可能要 換到不同型號的單晶執行,不同型號的可用腳位不同,或電 路變大,不同元件的腳位衝突等,都要改變腳位編號。)

```
const byte s2=47;//switch2
const byte s1=48;
const byte s0=49;
const byte 12=51;//led2
const byte 11=52;//led1
const byte 10=53;//led0
void setup() {
    Serial.begin(9600);
   pinMode(s2,INPUT PULLUP);//PL2
   pinMode(s1,INPUT PULLUP);//PL1
   pinMode(s0, INPUT PULLUP);//PL0
    DDRB=B11111111;
                       PORTB=0;
    DDRC=0xFF; PORTC=0xFE; //將點陣LED當作8個LED使用
}
```

(<u>138</u>)

```
void loop() {
   byte b;
   b=digitalRead(s2);//讀一個位元
   Serial.println(b);//輸出看看,確認是否正確
   if (b==0) //== 是比較是否相等運算子
     digitalWrite(12, HIGH); //用常數符號表示高電位5V
   else
     digitalWrite(12,LOW); //表示低電位OV
   b=digitalRead(s1);//讀一個位元
   Serial.println(b);//輸出看看,確認是否正確
   if (b==0) / /== 是比較是否相等運算子
     digitalWrite(l1,HIGH); //用常數符號表示高電位5V
   else
     digitalWrite(l1,LOW); //表示低電位OV
   b=digitalRead(s0);//讀一個位元
   Serial.println(b);//輸出看看,確認是否正確
   if (b==0) / /== 是比較是否相等運算子
     digitalWrite(10,HIGH); //用常數符號表示高電位5V
   else
     digitalWrite(10,LOW); //表示低電位OV
}
```

4. 使用 PINL 讀取 3 位元。本例一次讀 8 位元,但是我們僅用 3 個位元,沒有用到的 bit7, 6, 5, 4, 3 等 5 位元,請用位元運算子的 and 運算子『&』強制遮罩掉。(這是單晶控制的技巧,將不要的位元,用 and 0 去掉,要的位元用 and 1 留著。)

```
const byte s2=47;//switch2
const byte s1=48;
const byte s0=49;
const byte 12=51;//led2
```

```
const byte 11=52;//led1
const byte 10=53;//led0
void setup() {
    Serial.begin(9600);
    pinMode(s2,INPUT PULLUP);//PL2
    pinMode(s1,INPUT PULLUP);//PL1
    pinMode(s0,INPUT PULLUP);//PL0
    DDRB=B11111111;
                      PORTB=0;
    DDRC=0xFF; PORTC=0xFE; //將點陣LED當作8個LED使用
}
void loop() {
   byte b;
   b=PINL; //讀8個位元
    Serial.print(b);//輸出看看,確認是否正確
    Serial.print(',');
   b=b & B00000111;//遮罩沒用到的位元7,6,5,4,3
    Serial.println(b);//輸出看看,確認是否正確
    switch (b) {
      case (B000):
        digitalWrite(12,HIGH);
        digitalWrite(l1,HIGH);
        digitalWrite(10,HIGH);
       break; //語法, 不可漏掉
      case (B001):
        digitalWrite(12,HIGH);
        digitalWrite(l1,HIGH);
        digitalWrite(10,LOW);
       break;//不可漏掉
      case (B010):
        digitalWrite(12,HIGH);
        digitalWrite(l1,LOW);
        digitalWrite(10,HIGH);
       break; //不可漏掉
      case (B011):
        digitalWrite(12,HIGH);
        digitalWrite(l1,LOW);
        digitalWrite(10,LOW);
       break; //不可漏掉
      case (B100):
        digitalWrite(12,LOW);
```

(<u>-</u>) 140

```
digitalWrite(l1,HIGH);
   digitalWrite(10,HIGH);
   break;//不可漏掉
 case (B101):
   digitalWrite(12,LOW);
   digitalWrite(l1,HIGH);
   digitalWrite(10,LOW);
   break;//不可漏掉
 case (B110):
   digitalWrite(12,LOW);
   digitalWrite(l1,LOW);
   digitalWrite(10,HIGH);
   break;//不可漏掉
 case (B111):
   digitalWrite(12,LOW);
   digitalWrite(l1,LOW);
   digitalWrite(10,LOW);
   break;//不可漏掉
}
```

₩ 自我練習

}

- 請用3位元指撥開關,當作2進位的輸入,OFF代表0, ON代表1,且用七段顯示器輸出其值。例如,000,輸出0; 110,輸出6。
- 運作模式的選擇。請寫一個程式,讓您的單向紅綠燈可以用 1位元指撥開關設定2種運作模式,2種運作模式如下:

模式	緑燈時間	紅燈時間
0	5	5
1	閃黃燈	閃紅燈

(合 142 Arduino程式設計

3. 運作模式的選擇。請寫一個程式,讓您的單向紅綠燈可以用
 2 位元指撥開關設定4種運作模式,4種運作模式如下:

模式	緑燈時間	紅燈時間
0	5	5
1	7	3
2	4	6
3	閃黃燈	閃紅燈

範例 10c

輸入數字。

 指撥開關可用來輸入一個數字,本例使用3個指撥開關,如 下圖。第1個指撥開關代表1,第2個指撥開關代表2,第3 個指撥開關代表4,這樣就可以輸入0到7,且只要改變指撥 開關值,七段顯示器亦同步顯示輸入值。



 2. 單晶片的程式都要養成一個好習慣,那就是使用的腳位,要 取一個名稱。例如,本例使用腳位49,且將腳位取一個名 稱。

```
const byte s0=49;//1
```

這樣,往後的程式都使用 s0 表示腳位 49。例如:

```
pinMode(s0,INPUT_PULLUP);
byte s0val=digitalRead(s0);
```

這樣程式的移植性才會高,因為別人拿到程式時,有可能在 不同型號的晶片執行,因為不同型號的腳位數目不同,或有 些腳已經另有用途,此時只要在最前面改腳位編號就好,不 用在程式中到處改變腳位。

3. 以下是完整程式。

```
//指派腳位
```

```
const byte s0=49;//1
const byte s1=48;//2
const byte s2=47;//4
byte b[]={0x3f,0x6,0x5b,0x4f,0x66,0x6d,0x7c,0x7,0x7f, 0x67,0x0};//資料
void setup() {
  DDRF=0xFF;
  DDRK=0xFF; PORTK=0xFE;//將四位數七段當作一位數使用
  pinMode(s0, INPUT PULLUP); //指派腳位功能爲具有上拉電阻的輸入
 // pinMode(s0,INPUT);//這樣要自己接上拉電阻,不然不行
  pinMode(s1,INPUT PULLUP);
  pinMode(s2,INPUT PULLUP);
}
int t=0;
void loop() {
   t=0;
   byte s0val=digitalRead(s0);
   if (s0val==LOW)
       t=t+1;
   byte s1val=digitalRead(s1);
   if (s1val==LOW)
       t=t+2;
   byte s2val=digitalRead(s2);
   if (s2val==LOW)
       t=t+4;
    Serial.print(t);//顯示指撥開關所設定的初值
```

(合 144 Arduino程式設計

PORTF=b[t];//查表的動作,顧示O就要輸出Ox3f,顧示1就要輸出Ox6 PORTK=0xFE;

範例 10d

}

同上範例,但是程式一執行,還能倒數計時,結束時停留3 秒,然後重複以上動作。

🖗 程式列印

```
int t=0;//請留意不能用byte,因爲byte 沒有-1
void loop() {
   t=0;
   byte s0val=digitalRead(s0);
   if (s0val==LOW)
       t=t+1;
   byte s1val=digitalRead(s1);
   if (s1val==LOW)
       t=t+2;
   byte s2val=digitalRead(s2);
   if (s2val==LOW)
       t=t+4;
   Serial.print(t);//顯示指撥開關所設定的初值
   while (t>=0) { //只要t>0,就重複
     PORTF=b[t];//查表的動作,顧示O就要輸出Ox3f,顧示1就要輸出Ox6
     delay(1000);
     t=t-1;
   ł
   delay(3000);
}
```

以上是停留3秒,那把

```
delay(3000);
```

換成

while(true){}

就可等待到使用者按一下單晶片內置的『reset』鍵,待下一單元 介紹按壓開關,就可換成自己的『重置』按鈕。以下程式,計時 結束,還可持續閃爍。

```
while(true){
    PORTF=b[10];
    delay(500);
    PORTF=b[0];
    delay(500);
}
```

₩ 自我練習

 博奕程式。請使用6個指撥開關,僅撥第2個代表2…,僅撥 第6個代表6。程式一執行,就產生1到6的亂數,若相等 則使用LED輸出8個燈,若不相等也輸出一個燈。本例因為 沒有開關,所以每一次請用3秒鐘隔開。

範例 10e

表決器。假設有一項評審工作,有3位評審,當其中兩人或以 上同意,則表示過關且燈亮,請設計此電路。

🗑 電路設計

 本例先用指撥開關當作評審按鈕, 當有2個或以上評審on時,表示 通過,LED亮,電路設計如下: (亦可用按壓開關比較方便,請看 下一單元)



```
2. 以上想法的程式列印如下:
```

```
const byte sw1=49;
const byte sw2=48;
const byte sw3=47;
const byte led=13; //預植的LED
byte a;
byte b;
void setup() {
   pinMode(sw1, INPUT PULLUP);
   pinMode(sw2,INPUT PULLUP);
   pinMode(sw3,INPUT PULLUP);
   pinMode(led,OUTPUT);
   digitalWrite(led,LOW);
   DDRF=B111;//24,23,22
    Serial.begin(9600);
}
void loop() {
   a=PINL;
    a=~a;//逐位元反相
    a=a & B00000111;//將不要的位元遮罩
   PORTF=a;
   Serial.print(a);Serial.print(":");
   b=0;
   b=b+bitRead(a,0);//僅讀取位元0
   b=b+bitRead(a,1); //僅讀取位元1
   b=b+bitRead(a,2); //僅讀取位元2
   Serial.println(b);
    if (b>=2)
       digitalWrite(led,HIGH);
    else
        digitalWrite(led,LOW);
   delay(500);
}
```

 以上作法,程式較簡單,但是評審完,評審還要自己將指撥 開關撥回來,有點不方便。待下一單元介紹按壓開關,再用 按壓開關重做本範例。

(h) 146

🗑 自我練習

- 1. 同上範例,但每個人增加一個燈號,顯示自己的指撥狀態。
- 2. 假如有5位評審,三位以上通過就通過,請設計電路與程式。

範例 10f

選秀表決器。同上範例,但每個人才只有5秒鐘可以表演,兩 個人(含)以上按鈕,燈號亮,才表示入選。5秒鐘到了,沒 有通過也亮另外一個燈。(本例5秒只是方便測試,實務上,請 自行調整表演時間)

🗑 程式列印

```
const byte sw1=49;
const byte sw2=48;
const byte sw3=47;
const byte led=13; //預植的LED
byte a;
byte b;
void setup() {
   pinMode(sw1,INPUT PULLUP);
   pinMode(sw2,INPUT PULLUP);
   pinMode(sw3,INPUT PULLUP);
   pinMode(led,OUTPUT);
   digitalWrite(led,LOW);
   Serial.begin(9600);
   DDRB=B11111111;
   DDRC=0xFF; PORTC=0xFE; //0xFE is B11111110 將8*8點陣LED當作8個LED使用
}
long t1,t2;
void loop() {
  t1=millis();//取系統時間,單位是ms
  digitalWrite(led,LOW);
  do{ //開始計時
    a=PINL;
    a=~a;//逐位元反相
```

```
a=a & B00000111;//將不要的位元遮罩
   PORTF=a;
   Serial.print(a);Serial.print(":");
   b=0;
   b=b+bitRead(a,0); //讀取位元0,且累加此位元
   b=b+bitRead(a,1); //讀取位元1,且累加此位元
   b=b+bitRead(a,2); //讀取位元2,且累加此位元
   Serial.println(b);
   if (b>=2)
       digitalWrite(led,HIGH);
   else
       digitalWrite(led,LOW);
   t2=millis();
 }while( (t2-t1)<5000);//時間到
 if (b>=2){
       digitalWrite(led,HIGH);
 }
 else{
    while(true){//無窮迴圈,持續閃爍,直到使用者按單晶片的『reset』
         digitalWrite(led,HIGH);
         delay(200);
         digitalWrite(led,LOW);
         delay(200);
    }
 }
}
```

148

第11單元

按壓開關與叫號器

前面的指撥開關是用手指滑動開關,控制 ON 與 OFF,但是按 壓開關則是按下去時 ON,放開時,則有彈簧協助,又自動回到 OFF 狀態,按壓開關示意圖如下圖左,實體圖如下圖右。還有, 按壓開關有兩隻腳,也有四隻腳,兩隻腳如下圖中,兩隻腳的是 插麵包板用,請買腳長一點的,這樣可將腳撐開,如下圖中,剛 好可插到麵包板的地線,這樣佈線較簡單;四隻腳的如下圖右, 內部 2,3 兩兩相通,1,4 也兩兩相通,請用三用電表檢驗,以免使 用到成雙連接那兩隻,那就永遠是 ON,沒有按壓效果了。通常 1,3 為 1 組,或 2,4 為 1 組,沒按不導通,按鍵按下去則導通。四 隻腳的沒辦法插麵包板,若買到了,只好剪掉多餘的兩隻腳。下 圖右則是本書實驗板實體圖。



範例 11a

示範使用按壓開關。

₩操作步驟

- 1. 完成以下電路,其中 LED 同範例 2a、按壓開關麵包板接法 如下圖右。
- 2. 若使用本書實驗板,請連接 J9 到微控板即可, Gnd 內部已經 連接。





3. 鍵入以下程式,並觀察執行結果,請留意開關 ON,其值是 LOW, LED 滅掉。

```
void setup() {
   pinMode(22,INPUT_PULLUP);//PA0
   DDRB=B1111111;
   DDRC=0xFF;PORTC=0xFE;
   Serial.begin(9600);
}
void loop() {
   byte a=digitalRead(22);
   Serial.println(a);
   digitalWrite(53,a);
}
```

 本例 29,28,27,26,25,24,23,22 稱為 PORTA,也可使用 PINA 一次讀取 8 位元,請鍵入以下程式,並觀察執行結果,請留 意開關 ON,其值是 LOW,LED 滅掉。待會介紹蜂鳴器,也 是利用此電路製作電子琴。

```
void setup() {
 pinMode(29,INPUT_PULLUP);//PA7
 pinMode(28,INPUT PULLUP);//PA6
 pinMode(27, INPUT PULLUP);//PA5
 pinMode(26,INPUT PULLUP);//PA4
 pinMode(25, INPUT PULLUP);//PA3
 pinMode(24,INPUT_PULLUP);//PA2
 pinMode(23,INPUT PULLUP);//PA1
 pinMode(22,INPUT PULLUP);//PA0
  DDRC=0xFF; PORTC=0xFE;
  Serial.begin(9600);
}
void loop() {
   byte a=PINA;
   Serial.println(a);
   PORTB=a;
}
```

🗑 自我練習

- 1. 請用1個按壓開關,分別控制1個LED的明滅,且按下去 LED才亮。
- 2. 請用 8 個按壓開關,分別控制 8 個 LED 的明滅,且按下去 LED 才亮。

(合 152 Arduino程式設計

範例 11b

計數器。請寫一個程式,每當按壓開關被按一下,計數值加1。

₩ 思考步驟

1. 使用七段顯示器輸出計數結果,如下圖。



2. 程式設計初步如下:

```
const byte pb=22;
byte a[]={0x3f,0x6,0x5b,0x4f,0x66,0x6d,0x7d,0x7,0x7f,0x6f};
void setup() {
   pinMode(pb,INPUT PULLUP);
    Serial.begin(9600);
    DDRF=B11111111;
    DDRK=0xFF; PORTK=0xFE; //將四位數七段當作一位數
}
byte num=0;
byte b;
void loop() {
   b=digitalRead(pb);
    if (b==LOW)
        num++;
    Serial.println(num);
    PORTF=a[num];
}
```

 每按一下按鍵,計數值卻遞增很多, 那是因為單晶速度太快了,當您按下 按鈕的短短時間內,單晶已經重複偵 測很多次,而且開關接觸瞬間,開關 接點還會有彈跳現象(on與off之間 連續快速反覆變化),如右圖所示。



在此彈跳期間單晶也會重複偵測到此按壓開關很多次的 on 與 off,有些負載會吃不消,很容易壞掉。本例使用 while 迴圈 延遲一段時間,跳過這些彈跳電壓,程式如下:

```
const byte pb=22;
byte a[]={0x3f,0x6,0x5b,0x4f,0x66,0x6d,0x7d,0x7,0x7f,0x6f};
void setup() {
    pinMode(pb,INPUT PULLUP);
    Serial.begin(9600);
    DDRF=B11111111;
    DDRK=0xFF; PORTK=0xFE; //將四位數七段當作一位數
}
byte num=0;
byte b;
void loop() {
    b=digitalRead(pb);
    if (b==LOW) {
        while (digitalRead (pb) == LOW)
            delay(10);
        num++;
    }
    Serial.println(num);
    PORTF=a[num];
}
```

 也就是當按壓開關未被放開時,使用 while 迴圈繼續等待, 直到按鍵被釋放,程式再繼續執行,這樣計數值才不會一直 被灌水。本例的

```
while (digitalRead (pb) ==LOW)
       delay(10);
```

和以下程式效果相同,只是 while 迥圈僅執行一個敘述,所 以大括號省略。

```
while(digitalRead(pb) ==LOW) {
    delay(10);
}
```

- 5. 本例使用軟體克服電路彈跳問題。但是,非單晶的傳統數位 電路,則要用電容與電阻作一個微分電路,以便排除此一彈 跳電壓。
- 6. 以下程式,我同時用 LED 與七段顯示計數值。(本例請於 PORTB 接 LED 輸出)

```
const byte pb=22;
byte a[]={0x3f,0x6,0x5b,0x4f,0x66,0x6d,0x7d,0x7,0x7f,0x6f};
void setup() {
   pinMode(pb,INPUT PULLUP);
    Serial.begin(9600);
   DDRF=B11111111;
    DDRK=0xFF; PORTK=0xFE; //將四位數七段當作一位數
    DDRB=B11111111;
    DDRC=0xFF; PORTC=0xFE; //將8*8點陣LED當作8個LED使用
}
byte num=0;
byte b;
void loop() {
   b=digitalRead(pb);
    if (b==LOW) {
        while (digitalRead (pb) ==LOW)
            delay(10);
        num++;
    }
    Serial.println(num);
    PORTF=a[num];//七段顯示器
   PORTB=num;
                 //8個LED
}
```

(<u>5</u>) 154

🗑 自我練習

- 請安排2個按壓開關,其功能分別是遞增一、遞減一。(本題 使用序列埠視窗輸出)
- 2. 同上題,但輸出請用八個 LED 顯示 a 的值,計數值 1 就亮 1 個 LED,2 就亮 2 個 LED,依此類推。
- 同上題,但使用一個七段顯示器顯示其值,此即為叫號器。
 沒有印表機也沒關係,那請用一疊紙先寫好數字,消費者就可拿號碼牌了。等待 19 單元,再繼續介紹 4 位數叫號器。

範例 11c

請使用一個按壓開關控制 LED 的明亮,按一下時亮,且自保持,再按一下時滅,且自保持。

🗑 思考步驟

 本例為了能記憶目前的狀態且自保持,那我們需要一個變數 state,記錄且保留目前的狀態。

bool state=false;

2. 當按壓開關被按時,狀態改變。程式如下:

```
state=!state;//此爲布林的反相,!true =false,請看下一單元
```

₩程式列印(以下程式使用腳位編號 13 預置 LED)

```
const byte pb=22;//按壓開關
const byte led=13;//微控板預植LED
bool state=false;
byte b;
void setup() {
    pinMode(pb,INPUT_PULLUP);
    pinMode(led,OUTPUT);
```

```
digitalWrite(led,state);
}
void loop() {
    b=digitalRead(pb);
    if (b==LOW)
                       {
        while (digitalRead (pb) ==LOW)
             delay(10);
        state=!state;
    }
    digitalWrite(led,state);
}
```

₩ 補充說明

1. 請鍵入以下程式,並觀察執行結果。

```
void loop() {
   b=digitalRead(pb);
                 {
    if (b==LOW)
       state=!state;
    }
   digitalWrite(led,state);
}
```

₩ 補充說明

1. 本例與上一範例相同,因為單晶速度太快,而且有接點彈跳 問題,都要延遲一段時間。

∀自我練習

- 1. 請用兩個按壓開關控制一個 LED 的明滅,其中一個按下去 亮,且自保持;另一個按下去滅掉,且自保持。
- 2. 請用3個按壓開關控制3個LED,通通是按一下亮,且自保 持,再按一下滅,且自保持。
- 3. 請用兩個按壓開關控制 8 個 LED 的左旋或右旋。

(<u>5</u>) 156

- 請用一個按壓開關控制8個LED的左旋或右旋,每按一下, 改變狀態。
- 請寫一程式,使用一個按壓開關,控制三個 LED,讓使用者 在一個短暫的時間內,按一下時亮一個燈;按兩下時亮兩個 燈;按三下時亮三個燈;按四下時全熄滅。
- 6. 假設單向紅綠燈有兩種運轉模式,一種是紅燈、綠燈各5 秒,另一種是一邊閃紅、一邊閃黃,請寫一個程式,使用一 個按壓開關,控制單向紅綠燈的模式,每按一下按壓開關, 就轉為另一模式。
- 打地鼠。請利用 8 個 LED 與 8 個按壓開關,可以寫一個打地 鼠遊戲,得分由電腦螢幕輸出。
- 8. 同上題,得分用一個七段顯示器輸出。
- 9. 密碼鎖。假設您的密碼鎖有 8 個按鈕, 6 個號碼, 分別是 1 到 6, 1 個 Enter, 1 個更新密碼。本例預設密碼是 3,6,2 與 7 代表 Enter, 則 LED 全亮,代表門開,否則亮 1 個燈。
- 10. 同上題,連錯3次,則蜂鳴器發出警報聲,蜂鳴器請看第15 單元。
- 同上題,但按8可進行密碼更新,本列預設新密碼僅能3個 號碼,且連續輸入兩次,都相同,則可更新密碼。

第12單元

新北市110年國中技藝競賽電機電子群

『基本資訊應用』組試題解答

請先在麵包板完成以下電路,其中 Arduino 所使用腳位,於考試時才公告。此為 110 年新北市國中技藝競賽,賽前所公布的題目試題。



LED控制

以 S1、S2、S3 按鈕完成現場指定的三個功能(現場抽題,五個 功能抽三個),當 S1 按下時完成現場指定的的第一個功能;當 S2

第十二單元 新北市110年國中技藝競賽電機電子群『基本資訊應用』組試題解答 159

按下時,完成現場指定的第二個功能;當 S3 按下時,完成現場指定的第三個功能。現場待抽選五個功能要求如下:

 按下現場指定之按鈕時,首先是 LED0 發光,其餘的 LED 不 發光,其亮燈時間為1秒,之後則 LED1 發光,接著 LED2 發光。其順序 LED0 → LED1 → LED2 → LED3 → LED4 → LED 5 → LED6 → LED7,最後全部熄滅,每一 step 時間為1秒,若 再按下同一按鈕,則可再次循環。如下圖所示:



本例參考解答如下:

(1) 資料數位化如下:(本例將 LED 接到 PORTB)

時序	LED0	LED1	LED2	LED3	LED4	LED5	LED6	LED7	値
PORTB	13	12	11	10	50	51	52	53	
0	1	0	0	0	0	0	0	0	0x80
1	0	1	0	0	0	0	0	0	0x40
2	0	0	1	0	0	0	0	0	0x20
3	0	0	0	1	0	0	0	0	0x10
4	0	0	0	0	1	0	0	0	0x08
5	0	0	0	0	0	1	0	0	0x04
6	0	0	0	0	0	0	1	0	0x2
7	0	0	0	0	0	0	0	1	0x1
8	0	0	0	0	0	0	0	0	0

(合 160 Arduino程式設計

(2) 程式如下:本例將 LED 接到 PORTB。

```
void setup() {
    DDRB=B1111111;
    DDRC=0xFF;PORTC=0xFE;//將8*8點陣LED當作8個LED使用
}
void loop() {
    byte a[]={0x80,0x40,0x20,0x10,0x8,0x4,0x2,0x1,0};
    for (int i=0 ;i<=8 ;i++) {
        PORTB=a[i];
        delay(1000);
    }
}</pre>
```

 按下現場指定之按鈕時,首先是 LED7、6 發光,其餘的 LED 不發光,其亮燈時間為1秒,之後則 LED6、5 發光,接著 LED5、4 發光。其順序為 LED7、6→LED6、5→LED5、 4→LED4、3→LED3、2→LED2、1→LED1、0,最後全 部熄滅,每一 step 時間為1秒,若再按下同一按鈕,則可再 次循環。如下圖所示:



本例參考解答如下:

(1) 資料數位化如下:(本例將 LED 接到 PORTB。)

時序	LED0	LED1	LED2	LED3	LED4	LED5	LED6	LED7	値
PORTB	13	12	11	10	50	51	52	53	
0	0	0	0	0	0	0	1	1	0x3
1	0	0	0	0	0	1	1	0	0x6
2	0	0	0	0	1	1	0	0	0xc
3	0	0	0	1	1	0	0	0	0x18
4	0	0	1	1	0	0	0	0	0x30
5	0	1	1	0	0	0	0	0	0x60
6	1	1	0	0	0	0	1	1	0xc0
7	0	0	0	0	0	0	0	0	0x0

(2) 程式如下:本例將 LED 接到 PORTB。

```
void setup() {
    DDRB=B1111111;
    DDRC=0xFF;PORTC=0xFE;//將8*8點陣LED當作8個LED使用
}
void loop() {
    byte a[]={0x3,0x6,0xc,0x18,0x30,0x60,0xc0,0};//7
    for (int i=0 ;i<=7 ;i++){
        PORTB=a[i];
        delay(1000);
    }
}</pre>
```

3. 按下現場指定之按鈕時,首先是 LED0 發光,其餘的 LED 不發光,其亮燈時間為1秒,之後 LED0、1發光,接著 LED0、1、2發光。順序點亮各 LED,其點亮之順序為 LED 0→LED1→LED2→LED3→LED4→LED5→LED6→LED7,至 全部亮起,最後全部熄滅,每一 step 時間為1秒,若再按下 同一按鈕,則可再次循環。如下圖所示:



本例參考解答如下:

(1) 資料數位化如下:(本例將 LED 接到 PORTB。)

時序	LED0	LED1	LED2	LED3	LED4	LED5	LED6	LED7	値
PORTB	13	12	11	10	50	51	52	53	
0	1	0	0	0	0	0	0	0	0x80
1	1	1	0	0	0	0	0	0	0xc0
2	1	1	1	0	0	0	0	0	0xe0
3	1	1	1	1	0	0	0	0	0xf0
4	1	1	1	1	1	0	0	0	0xf8
5	1	1	1	1	1	1	0	0	0xfc
6	1	1	1	1	1	1	1	0	0xfe
7	1	1	1	1	1	1	1	1	0xff
8	0	0	0	0	0	0	0	0	0

(2) 程式如下:本例將 LED 接到 PORTB。

```
void setup() {
    DDRB=B11111111;
    DDRC=0xFF;PORTC=0xFE;//將8*8點陣LED當作8個LED使用
}
void loop() {
    byte a[]={0x80,0xc0,0xe0,0xf0,0xf8,0xfc,0xfe,0xff,0};//8
    for (int i=0 ;i<=8 ;i++){</pre>
```

162

```
PORTB=a[i];
delay(1000);
}
}
```

 按下現場指定之按鈕時,首先是左右兩側 LED0、7 點亮, 其亮燈時間為1秒,接著往中間 LED 點亮,其點亮之順序為 LED0、7→LED1、6→LED2、5→LED3、4,依序由兩側 往中間點亮,最後全部熄滅,每一 step 時間為1秒,若再按 下同一按鈕,則可再次循環。如下圖所示:



本例參考解答如下:

(1) 資料數位化如下:(本例將 LED 接到 PORTB。)

時序	LED0	LED1	LED2	LED3	LED4	LED5	LED6	LED7	値
PORTB	13	12	11	10	50	51	52	53	
0	1	0	0	0	0	0	0	1	0x81
1	0	1	0	0	0	0	1	0	0x42
2	0	0	1	0	0	1	0	0	0x24
3	0	0	0	1	1	0	0	0	0x18
8	0	0	0	0	0	0	0	0	0

(2) 程式如下:本例將 LED 接到 PORTB。

```
void setup() {
DDRB=B11111111;
DDRC=0xFF;PORTC=0xFE;//將8*8點陣LED當作8個LED使用
}
```

```
)164 Arduino程式設計
```

```
void loop() {
   byte a[]={0x81,0x42,0x24,0x18,0};//4
   for (int i=0 ;i<=4;i++) {
      PORTB=a[i];
      delay(1000);
   }
}</pre>
```

5. 按下現場指定之按鈕時,首先是由右向左點亮LED,其 燈亮時間為1秒,依序為LED7→LED6→LED5→LED4→ LED3→LED2→LED1→LED0,接著再由左向右依序點亮 LED0→LED1→···→LED7,最後全部熄滅,每一 step時間為1秒,若再按下同一按鈕,則可再次循環。如下 圖所示:



(1) 參考程式如下:本例將 LED 接到 PORTB。

void setup() {
 DDRB=B11111111;

164

七段顯示器控制

 按下 S4 按鈕時,首先是七段顯示器的 a 段發光,其餘各段不 發光,其亮燈時間為 0.5 秒,之後 a、b 段發光,接著 a、b、 c 段發光。順序點亮各段,其點亮之順序為 a 段→b 段→c 段→d 段→e 段→f 段→g 段→dp 段,至全部亮起,最後全 部熄滅,每一 step 時間為 0.5 秒,若再按下同一按鈕,則可 再次循環。



本例參考解答如下:

(1) 資料數位化如下:(本例將七段顯示器接到 PORTF)

時序	LEDdp	LEDg	LEDf	LEDe	LEDd	LEDc	LEDb	LEDa	値
PORTF	A7	A6	A5	A4	A3	A2	A1	A0	
0	0	0	0	0	0	0	0	1	0x1
1	0	0	0	0	0	0	1	1	0x3
2	0	0	0	0	0	1	1	1	0x7
3	0	0	0	0	1	1	1	1	0xf
4	0	0	0	1	1	1	1	1	0x1f

時序	LEDdp	LEDg	LEDf	LEDe	LEDd	LEDc	LEDb	LEDa	値
PORTF	A7	A6	A5	A4	A3	A2	A1	A0	
5	0	0	1	1	1	1	1	1	0x3f
6	0	1	1	1	1	1	1	1	0x7f
7	1	1	1	1	1	1	1	1	0xff
8	0	0	0	0	0	0	0	0	0

(2) 程式如下:本例將七段顯示器接到 PORTF。

```
void setup() {
    DDRF=B1111111;
    DDRK=0xFF;PORTK=0xFE;//將四位數七段當作一位數
}
void loop() {
    byte a[]={0x01,0x03,0x07,0x0f,0x1f,0x3f,0x7f,0xFF,0};//8
    for (byte i=0;i<=8;i++) {
        PORTF=a[i];
        delay(500);
    }
}</pre>
```

 按下 S5 按鈕時,七段顯示器可依序顯示 0→1→2→3→ 4→5→6→7→8→9,等十個數字,每個數字顯示 0.5 秒 後熄滅,再顯示下一個數字,顯示到9之後全部熄滅;在依 序顯示過程中,若按壓一次 S5 按鈕,則會停止下一位數的顯 示,直到再按壓一次 S5 按鈕時才會再繼續顯示停止前的下一 位數,在七段顯示器上顯示的 0~9 十個數須與下圖相同。

$$\begin{bmatrix} \mathbf{I} \\ \mathbf{I} \\ \mathbf{I} \\ \mathbf{0} \end{bmatrix} \begin{bmatrix} \mathbf{I} \\ \mathbf{I} \\ \mathbf{I} \end{bmatrix} \begin{bmatrix} \mathbf{I} \\ \mathbf{I} \end{bmatrix} \end{bmatrix} \begin{bmatrix} \mathbf{I} \\ \mathbf{I} \end{bmatrix} \end{bmatrix} \begin{bmatrix} \mathbf{I} \\ \mathbf{I} \end{bmatrix} \begin{bmatrix} \mathbf{I} \\ \mathbf{I} \end{bmatrix} \begin{bmatrix} \mathbf{I} \\ \mathbf{I} \end{bmatrix} \end{bmatrix} \begin{bmatrix} \mathbf{I} \\ \mathbf{I} \end{bmatrix} \begin{bmatrix} \mathbf{I} \\ \mathbf{I} \end{bmatrix} \begin{bmatrix} \mathbf{I} \\ \mathbf{I} \end{bmatrix} \end{bmatrix} \begin{bmatrix} \mathbf{I} \\ \mathbf{I} \end{bmatrix} \begin{bmatrix} \mathbf{I} \\ \mathbf{I} \end{bmatrix} \end{bmatrix} \begin{bmatrix} \mathbf{I} \\ \mathbf{I} \end{bmatrix} \begin{bmatrix} \mathbf{I} \\ \mathbf{I} \end{bmatrix} \end{bmatrix} \begin{bmatrix} \mathbf{I} \\ \mathbf{I} \end{bmatrix} \begin{bmatrix} \mathbf{I} \\ \mathbf{I} \end{bmatrix} \end{bmatrix} \begin{bmatrix} \mathbf{$$

166

數字	LEDdp	LEDg	LEDf	LEDe	LEDd	LEDc	LEDb	LEDa	十六進制 數値
PORTF	A7	A6	A5	A4	A3	A2	A1	A0	
0	0	0	1	1	1	1	1	1	0x3f
1	0	0	0	0	0	1	1	0	0x6
2	0	1	0	1	1	0	1	1	0x5b
3	0	1	0	0	1	1	1	1	0x4f
4	0	1	1	0	0	1	1	0	0x66
6	0	1	1	1	1	1	0	0	0x7c
7	0	0	0	0	0	1	1	1	0x7
8	0	1	1	1	1	1	1	1	0x7f
9	0	1	1	0	0	1	1	1	0x67
10	0	0	0	0	0	0	0	0	0

(2) 程式如下:本例將七段顯示器接到 PORTF。

硬體測試

(所有單晶電路都要先測試硬體電路,先確認硬體對了,再繼續)

1. 測試 8 個 LED 與七段顯示器是否正常

```
void setup() {
   DDRB=B11111111;
   DDRC=0xFF; PORTC=0xFE; //將8*8點陣LED當作8個LED使用
   DDRF=B11111111;
   DDRK=0xFF; PORTK=0xFE; //將四位數七段當作一位數
}
void loop() {
  PORTB=0xff;
  PORTF=0xff;
 delay(1000);
  PORTB=0;
 PORTF=0;
  delay(1000);
}
```

2. 測試按鍵輸入。本例將按壓開關接到 22,23,24,25,26,27 等腳 位,且於螢幕顯示按鍵是否正常工作。

```
const byte s1=22;
const byte s2=23;
const byte s3=24;
const byte s4=25;
const byte s5=26;
void setup() {
   pinMode(s1,INPUT PULLUP);
  pinMode(s2,INPUT PULLUP);
  pinMode(s3,INPUT PULLUP);
   pinMode(s4, INPUT PULLUP);
  pinMode(s5,INPUT PULLUP);
   Serial.begin(9600);
}
int c=0;
void loop() {
 byte b1=digitalRead(s1);
 byte b2=digitalRead(s2);
 byte b3=digitalRead(s3);
 byte b4=digitalRead(s4);
 byte b5=digitalRead(s5);
  if (b1==LOW) {
    while (digitalRead(s1) == LOW)
```

(<u>168</u>)

```
delay(10);
    c=c+1;
    Serial.println(c);
  }
  if (b2==LOW) {
    while (digitalRead(s2) == LOW)
      delay(10);
    c=c+1;
    Serial.println(c);
  }
 if (b3==LOW) {
    while (digitalRead(s3) == LOW)
      delay(10);
    c=c+1;
    Serial.println(c);
  }
 if (b4==LOW) {
    while (digitalRead(s4) == LOW)
      delay(10);
    c=c+1;
    Serial.println(c);
  }
 if (b5==LOW) {
    while (digitalRead(s5) == LOW)
      delay(10);
    c=c+1;
    Serial.println(c);//將結果輸出在序列埠視窗。
  }
}
```

3. 完整程式如下:

```
const byte s1=22;
const byte s2=23;
const byte s3=24;
const byte s4=25;
const byte s5=26;
void setup() {
    DDRB=B11111111;
    DDRC=0xFF;PORTC=0xFE;//將8*8點陣LED當作8個LED使用
```

```
DDRF=B11111111;
   DDRK=0xFF; PORTK=0xFE; //將四位數七段當作一位數
   pinMode(s1,INPUT PULLUP);
   pinMode(s2,INPUT PULLUP);
   pinMode(s3,INPUT PULLUP);
   pinMode(s4, INPUT PULLUP);
   pinMode(s5,INPUT PULLUP);
}
void loop() {
  byte b1=digitalRead(s1);
  byte b2=digitalRead(s2);
  byte b3=digitalRead(s3);
  byte b4=digitalRead(s4);
  byte b5=digitalRead(s5);
  if (b1==LOW) {
    while (digitalRead(s1) == LOW)
      delay(10);
    byte a[]={0x80,0x40,0x20,0x10,0x8,0x4,0x2,0x1,0};//8
    for (byte i=0;i<=8;i++) {</pre>
      PORTB=a[i];
      delay(1000);
    }
  }
  if (b2==LOW) {
    byte a[]={0x3,0x6,0xc,0x18,0x30,0x60,0xc0,0};//7
    for (byte i=0;i<=7;i++) {</pre>
      PORTB=a[i];
      delay(1000);
    }
  }
  if (b3==LOW) {
    byte a[]={0x80,0xc0,0xe0,0xf0,0xf8,0xfc,0xfe,0xff,0};//8
    for (byte i=0;i<=8;i++) {</pre>
      PORTB=a[i];
      delay(1000);
    }
  ł
  if (b4==LOW) {
    byte a[]={0x01,0x03,0x07,0x0F,0x1F,0x3F,0x7F,0xFF,0};
    for (byte i=0;i<=8;i++) {</pre>
      PORTF=a[i];
```

(<u>-</u>) 170

麵包板

剛買回來的麵包板如下圖左,可將微控板放在中間,並用螺絲輕 輕拴在麵包板(麵包板的洞洞剛好可鎖螺絲,但請不要拴太緊, 以免破壞麵包板),如下圖左。完成以上,接線如下圖右。



每個洞洞下面都有彈簧銅片,可以用來插入電子零件並固定與 連接這些電子零件,這些洞洞每5個一組,編號ABCDE內部相 通,FGHIJ內部也相通。周圍紅色與藍色長線條用來連接正負電 源。這些線條不連續,不連續的地方代表沒有相通,那是因為有

(合 172 Arduino程式設計

些電路需要有不同電壓大小的多電源,所以才會分區分開,但是 我們這種單晶數位電路都只有一種單電源,所以請您使用短的導 線,如下圖,將斷點自己補齊,補完後如上圖。特別留意的地方 是,廠商所準備的導線,其裸線都有0.5公分,這樣插下去,才 能確保插的夠深、夠牢靠。自己剪的千萬不能低於0.5公分,如 下圖左,若只有0.3公分,因為插的不夠深、當然不牢靠,回家 的路上搖一搖,很容易就鬆脫了,且不容易用肉眼除錯,那就會 很悽慘。導線粗細也要一致,不然彈簧被粗的撐開,細的就接觸 不良了。還有,每次電路做完,最好養成習慣,將導線拆光,因 為導線插太久,彈簧就鬆了,下次很可能接觸不良。麵包板不是 便宜就好,便宜的容易彈性疲乏,很容易接觸不良。

導線

因為 Arduino 微控板 I/O 腳位的洞洞比麵包板深、下圖右是 Arduino 專用杜邦彩虹線,線徑幾乎有 0.8mm,這樣才夠硬,才 能插的下去,而且這種彩虹線顏色有意義,較不容易插混了。中 間是麵包板專用導線,線徑有 0.6mm,勉強還可用,但右邊的單 心線太細,不到 0.5mm,那會太軟,根本差不進去;其次,多心 蕊線也都太軟插不進去。杜邦彩虹線,請將依白黑兩條拆開一組 (不要全撕開),用來當作電源的正負;棕、紅、橙、黃、綠、藍、 紫、灰 8 條一組,編號是 1,2,3,4,5,6,7,8 (不要全撕開)。使用時 要依照顏色來用,這樣才不會亂掉。



第十二單元 新北市110年國中技藝競賽電機電子群『基本資訊應用』組試題解答 173

導線如何剝皮

若有剝線鉗當然方便,但是因為單晶實習線路較簡單,沒有也沒關係,請照以下 SOP 練習。

- 左手拿尖嘴鉗,夾住導線。請用尖嘴鉗光滑的地方夾導線, 而不要用鋸齒的部分夾,這樣才不會刺破導線。
- 2. 右手拿斜口鉗,斜面部分平貼尖嘴鉗。
- 用斜口鉗斜面尖端貼住尖嘴鉗尖端當支點,且輕輕夾住導線,向外推開,就可將導線剝皮。