

目錄

第1章LED 閃爍	2
第2章按鈕	6
第3章For 迴圈	10
第4章.類比電壓訊號輸入	14
第5章.蜂鳴片訊號輸入	18
第6章.七段顯示器	24
第7章.直流馬達-PWM 控制輸出	28
第8章.伺服馬達控制	39
第9章.音樂旋律(蜂鳴器或喇叭)	45
第10章.LCD顯示	51
第11章.數位輸出擴充	56



第1章. LED 閃爍

介紹

LED 也稱發光二極體,主要拿來當作指示燈和數字顯示器等,廣泛使用在儀器、消費性產品和科學儀器上。加正向電壓時,發光二極體能發出單色、不連續的光;正 負電源如果接反,則不發亮。一般常用的 LED 樣式如圖 1-1,其中 LED 較長的接腳 為正極,較短的接腳負極。



圖 1-1.LED 接腳

本範例將學習用 Arduino 內定程式控制板上數位接腳 12 的 LED 燈閃爍。

使用材料

項次	材料名稱	數量
1	Arduino 載板	1
2	USB 連接埠	1
3	LED 燈	1
4	220 歐姆電阻	1
5	跳線	若干

表格 1. 使用材料

-----www.aroboto.com------

(TEL) 02-25576923 (FAX) 02-25577529

Email: support@aroboto.com



接線與電路

電路配線:LED 有極性,正極接數位腳 P2,負極跨接一個 220 歐姆的電阻後接地。 如圖 1-2。



圖 1-2.LED 電路

-----www.aroboto.com------

(TEL) 02-25576923 (FAX) 02-25577529 Email: support@aroboto.com



接線方式:LED 在 Arduino 的實際配線法,如下圖 1-3



圖 1-3. LED 在 Arduino 的配線

程式

int ledPin = 2;	// LED 連接數位接腳 2
<pre>void setup(){ pinMode(ledPin, OUTPUT);</pre>	// LED 接腳設為輸出
}	
void loop(){	
digitalWrite(ledPin, HIGH);	// LED 亮
delay(1000);	// 延遲1秒
digitalWrite(ledPin, LOW);	// LED 滅
delay(1000);	// 延遲1秒
}	



執行上述程式後 LED 會亮 1 秒然後滅,並如此重複亮滅;當 P2 給 High 時 LED 亮, 給 LOW 時滅;程式中 delay()的時間單位是毫秒,所以 delay(1000)即延遲 1 秒。 若 delay(500)即延遲半秒。

上述範例中 LED 接在 P2 接腳,並跨接 220 歐姆電阻後接地。如果使用數位接腳 P13, LED 可直接插在 P13 腳位和接地間,不必再另外接電阻,因為 Arduino 裡面已有 1K 歐姆電阻。

-----www.aroboto.com------

(TEL) 02-25576923 (FAX) 02-25577529 Email: support@aroboto.com



第2章。按鈕

介紹

按鈕也稱為按鍵,用來控制機械或程式的某些功能。在人機介面的使用上,按鈕是個不可或缺的電子產品。

一般電子零件的按鈕樣式如下圖 2-1。按鈕的設計,接腳 1 和 3 相連,接腳 2 和 4 也是相連的。



圖 2-1. 按鈕

下圖 2-2 說明按鈕接腳使用的狀態。當按鈕處於未按壓狀態時,1、3與2、4 沒有 導通,若按鈕處於按壓狀態時,則接腳1、3與2、4 才會導通。

若按鈕按壓時 1 和 3、2 和 4 才導通



若按鈕未按壓時 1 和 3、2 和 4 不導通

圖 2-2. 按鈕腳位

本範例將學習利用按鈕控制 Arduino 板上 LED 燈的閃爍。



使用材料

項次	材料名稱	數量
1	Arduino 載板	1
2	USB 連接埠	1
3	LED 燈	1
4	10K 歐姆電阻	1
5	按鈕	1
6	麵包板	1
7	跳線	若干

表格 2-1. 材料清單

接線與電路

按鈕部分:一端接+5V,另一端跨接一個 10K 歐姆電阻後接地,同時也接 Arduino 數位腳 P2。

LED 部分: LED 正極接 Arduino 數位腳 P3, 負極跨接一個 220 歐姆的電阻後接地。 如下圖 2-3。





接線方式:按鈕在 Arduino 的實際配線法,如圖 2-4。



圖 2-4. 按鈕在 Arduino 的配線

程式

int buttonPin = 2 ;	//宣告按鈕狀態的讀取腳位為2
int ledPin = 3;	//宣告 LED 控制腳位為 3
int buttonState $= 0;$	//宣告 buttonState 變數・初始值為 0
void setup(){	
pinMode(ledPin, OUTPUT);	//設定 LED 腳位是輸出
pinMode(buttonPin, INPUT);	//設定按鈕腳位是輸入
}	
void loop(){	
// 讀取按鈕的狀態,並將狀態	長存在 buttonState 變數中
buttonState = digitalRead(butt	onPin);



// 如果 buttonState=1 ・ 也就是按鈕指	安下而導通	
if (buttonState == HIGH){		
digitalWrite(ledPin, HIGH);	// 打開 LED	
}		
// 如果 buttonState=1 不成立・也就是	buttonState=0 按鈕未按下不導	Ē
else{		
digitalWrite(ledPin, LOW);	//關掉 LED	
}		
}		

說明

上述程式碼目的是當 P2 收到高電位時 LED 燈亮,收到低電位時,LED 燈滅。因此, 當壓下按鈕時 LED 將亮起;鬆開後 LED 燈將熄滅。

按鈕未按下時 (圖 2-2) 在按鈕的兩個接腳間 (點 3 與點 4) 是沒有導通。此時 P2 點經 10K 電阻接地偵測到低電位。當按下按鈕時,它的兩個接腳間 (點 3 與點 4) 則會導 通,此時 P2 腳偵測到 5V 電壓,為高電位。

如果數位接腳 P2 不連接任何東西 (不經 10K 電阻接地),此時 LED 會閃爍不定。這 是因為 P2 偵測到的輸入訊號在飄移,隨意徘徊在電位 High 和 Low 間,解決方法是 插入你拔出的電阻 (回復到原來的接法)。

本單元所使用的電阻接法為下拉電阻 (pull-down resistor) ,你也可以嘗試提升 電阻 (pull-up resistor)的接法,如下圖 2-5。





第3章. For 迴圈

介紹

本範例將藉由 Arduino 多組數位 1/0 接腳控制 LED 燈,來學習 for 迴圈的使用。

使用材料

項次	材料名稱	數量
1	Arduino 載板	1
2	USB 連接埠	1
3	LED 燈	6
4	220 歐姆電阻	6
5	麵包板	1
6	跳線	若干

表格 3-1 材料清單

接線與電路

電路配線:LED 正極接 Arduino 上數位 1/0 腳位(從 P2 到 P7)。LED 負極跨接一個 220 歐姆的電阻後接地。(如下圖 3-1)。







接線方式:LED 與 Arduino 的實際接線,如下圖 3-2



圖 3-2. Arduino 與多組 LED 的配線

程式





```
void loop() {
    //依序控制 I/O 腳位(從 thisPin=2 開始、每次遞增、到 thisPin=7 結束)
    for (int thisPin = 2; thisPin < 8; thisPin++) {
        digitalWrite(thisPin, HIGH); //控制 輪到的腳位輸出高電壓
        delay(timer); //延遲 0.1 秒
        digitalWrite(thisPin, LOW); //控制 輪到的腳位輸出低電壓
    }
//依序控制 I/O 腳位(從 thisPin=7 開始、每次遞減、到 thisPin=2 結束)
    for (int thisPin = 7; thisPin >= 2; thisPin--) {
        digitalWrite(thisPin, HIGH); //控制 輪到的腳位輸出高電壓
        delay(timer); //延遲 0.1 秒
        digitalWrite(thisPin, LOW); //控制 輪到的腳位輸出高電壓
        delay(timer); //延遲 0.1 秒
        digitalWrite(thisPin, LOW); //控制 輪到的腳位輸出低電壓
    }
}
```

說明

for 迴圈的語法使用說明如圖 3-3。for 後面的小括號內有 3 個變數條件:初始值、 測試條件、更新運算;當測試條件成立時,就會執行迴圈內的敘述程式(大括號內 的程式碼)。

執行 For 迴圈程式時, 會先預設一組初始值, 執行迴圈內的敘述程式前, 會判斷是 否滿足測試條件,條件成立時執行敘述程式,執行完後更新運算(通常為增量或減 量),然後再次判斷是否滿足測試條件,如此重複執行直到測試條件不符時,則跳出 此迴圈,繼續向下執行後面的程式。

> for (初始值 ; 測試條件 ; 更新運 算) { 控制過程敘述 ; }

圖 3-3.For **迴**圈



執行上述程式時,第一段 for 迴圈,會讓 LED 會依序讓 thisPin 遞增,由 P2 開始 亮 0.1 秒後熄滅,P3 亮 0.1 秒後熄滅…P7 亮 0.1 秒後熄滅,然後因為 thisPin=8, 不符合 thisPin < 8 的條件,所以跳出第一段 for 迴圈,進入第二段 for 迴圈。

第二段 for 迴圈,會讓 LED 會依序讓 thisPin 遞減,由 P7 開始亮 0.1 秒後熄滅, P6 亮 0.1 秒後熄滅…P2 亮 0.1 秒後熄滅,然後因為 thisPin=1,不符合 thisPin>=2 的條件,所以跳出第二段 for 迴圈,執行 loop,再回到第一段 for 迴圈。如此一直 循環下去。

而亮滅的動作是來當 1/0 腳位給 High 時 LED 亮,給 LOW 時 LED 滅;程式中 de lay () 的時間單位是毫秒,所以 de lay (1000) 即程式延遲 1 秒。若 de lay (100) 即延遲 0.1 秒。所以 LED 會亮持續 o. 1 秒後滅。

-----www.aroboto.com------

(TEL) 02-25576923 (FAX) 02-25577529 Email: support@aroboto.com



第4章. 類比電壓訊號輸入

介紹

本文介紹類比電壓訊號輸入。利用可變電阻 Potentiometer 改變所產生的電壓變化, 來調整類比電壓的輸入值。<u>可變電阻</u>樣式具有三個端子,如下圖 4-1。右圖接腳 1 和接腳 3 分別接地和電源(相反亦可),接腳 2 傳輸可變電阻造成的電壓變化。當旋 鈕順時鐘旋轉接腳 1 和接腳 2 的電阻值會升高,接腳 2 和接腳 3 的電阻值則降低。 常見的可變電阻分兩種特性:電阻成指數遞增,代號 A;電阻成線性遞增,代號 B。



圖 4-1. 可變電阻

本範例將學習利用可變電阻控制 Arduino 板上 LED 燈的閃爍的快慢。

使用材料

項次	材料名稱	數量	
1	Arduino 載板	1	
2	USB 連接埠	1	
3	LED 燈	1	
4	2K 歐姆電阻	1	
5	220 歐姆電阻	1	
6	麵包板	1	
7	跳線	若干	



接線與電路

電路配線:可變電阻一端接到+5V,另一端接地,中間的接 Arduino 類比輸入接腳 P0。同時,LED 正極接 Arduino 數位 1/0 腳位 P2,負極跨接 220 歐姆電阻後接地, 如圖 4-2。



接線方式: 可變電阻與 Arduino 的實際配線法, 如下圖 4-3





```
int sensorPin = 0;
                         // 宣告可變電阻輸入腳位為 0
int ledPin = 2;
                         // 宣告 LED 控制 腳位為 2
int sensorValue = 0;
                         // 宣告 sensorValue 變數,初始為0
void setup() {
 pinMode(ledPin, OUTPUT);
                        //設定 LED 腳位為輸出
void loop() {
   //讀取感測器的數值,儲存在 sensorValue 變數中
 sensorValue = analogRead(sensorPin);
   //LED 接腳輸出高電壓,LED 亮
 digitalWrite(ledPin, HIGH);
   //以 sensorValue 的讀值為延遲時間
 delay(sensorValue);
   //LED 接腳輸出低電壓, LED 滅
 digitalWrite(ledPin, LOW);
  //以 sensorValue 的讀值為延遲時間
 delay(sensorValue);
```

說明

執行程式,並且轉動可變電阻的旋鈕,LED 燈亮滅時間會有長短變化,這是受到可 變電阻控制的影響。如圖 4-3,當可變電阻的旋鈕向右轉,LED 燈亮滅間隔時間就 會變長,因為類比接腳 PO 所接收到的電壓值增加,sensorValue 值變大;相反的當 可變電阻的旋鈕向左轉,LED 燈亮滅間隔時間會縮短,因為類比接腳 PO 所接收到的 電壓降低,sensorValue 值變小。



由 Arduino 提供的 5V 電壓,可透過可變電阻,改變輸入到類比接腳 P0 的電壓大小。 由於 Arduino本身具有 10bit (2 的" 10" 次方=1024) 解析度的 ADC 轉換,也就是 A (類 比) 轉 D (數位),因此可以將 0~5V 的類比電壓值,轉換成 0~1023 的數位數值,方便 程式作運算。如果類比接腳接收到電壓是 0V,轉換成的數值是 0;如果接收到電壓 是 5V,轉換成的數值是 1023;如果接收到電壓是 2.5V,轉換成的數值是 511。

程式中 delay()的時間單位是毫秒,1000 毫秒即1秒。類比腳位讀取訊號的範圍值 是 0~1023,換算成 LED 的亮滅延遲時間是在 0~1.023 秒間。Arduino 的類比接腳接 收到電壓值越高,從電壓轉換數位數值越高,也就代表的延遲時間越高,LED 亮滅 相隔時間越長,亮滅越慢。

-----www.aroboto.com------

(TEL) 02-25576923 (FAX) 02-25577529 Email: support@aroboto.com



第5章. 蜂鳴片訊號輸入

介紹

本文介紹蜂鳴片訊號輸入,由於蜂鳴片的震動會產生電壓變化,因此可透過類比電 壓腳位可讀取產生的電壓值。蜂鳴片樣式,如下圖 5-1;蜂鳴片由兩種不同壓電材 質金屬片構成:接腳1焊在內層金屬片上,用紅線表示以連接正極;接腳2焊在外 層金屬片上,用黑線表示以連接負極。



圖 5-1 蜂鳴片

本範例將學習利用蜂鳴片控制 Arduino 板上 LED 燈的閃爍快慢。

使用材料

項次	材料名稱	數量
1	Arduino 載板	1
2	USB 連接埠	1
3	LED 燈	1
4	蜂鳴片	1
5	1M 歐姆電阻	1
6	麵包板	1
7	跳線	若干

表格 5-1



電路配線:蜂鳴片與一個 1MB 歐姆電阻的一端相接,再接上類比輸入接腳 PO,另一端接地。如圖 5-2



圖 5-2. 蜂鳴片與線路圖

接線方式:蜂鳴片與 Arduino 的實際配線法,如下圖 3





圖 5-3. 蜂鳴片與 Arduino 的配線

-----www.aroboto.com-----

(TEL) 02-25576923 (FAX) 02-25577529 Email: support@aroboto.com



```
const int ledPin = 13;
                               //宣告 LED 接腳為 13
const int knockSensor = 0;
                                //宣告感測器接腳為0
//宣告 threshold 為常數,並將數值設為 200,作門檻值。
const int threshold = 200;
                  //宣告 sensorReading 變數,初始值 0
int sensorReading = 0;
                      //宣告 ledState 變數・初始值 0
int ledState = LOW;
void setup() {
pinMode(ledPin, OUTPUT);
                             //設定 LED 腳位是輸出
 Serial.begin(9600);
                             //設定使用鮑率是 9600bps
void loop() {
 //讀取蜂鳴片數值,並儲存至 sensorReading 變數中
 sensorReading = analogRead(knockSensor);
 //判斷 sensorReading 數值是否超過門檻值
 //如果超過切換 LED 狀態,並在監控視窗中顯示訊息"Knock!"
 if (sensorReading>=threshold) {
  ledState = !ledState;
                                   //改變 ledState 變數
   digitalWrite(ledPin, ledState);
                                   //切換 LED 狀態
   Serial.println("Knock!"); //在監控螢幕顯示訊息"Knock!"
  }
 delay(100); //延遲 0.1 秒, 避免資料傳輸至電腦端過快, 而當機
```



說明

蜂鳴片是壓電材料,受到敲擊或震動時線路兩端會產生電壓差,類比接腳 P0 就會 讀取電壓並轉換成數值訊號,透過程式判斷來決定 LED 燈的亮滅。

執行程式後, 敲擊蜂鳴片, LED 燈會出現亮滅的變化, 會依據蜂鳴片敲擊次數來改 變 LED 狀態。如圖 5-3, 初始值 ledState = LOW, 因此一開始 LED 不亮燈, 當蜂鳴 片受到第一下敲擊時, 蜂鳴片受到震動而送出電壓, 此時電壓訊號將高於門檻值而 觸發 ledState 改變, 由 LOW 切換為 HIGH, 使 LED 燈亮; 敲擊第二下時, 此時電壓 訊號又高於門檻值再次觸發 ledState 改變, 由 HIGH 切換為 LOW, 造成 LED 燈滅。

if 程式運作如圖 5-4, if 後面小括號內是變數判斷;條件成立時執行敘述程式(大括號內的程式碼),若不符合判斷就執行下面敘述。



另外,在本範例的程式中我們使用到 Serial.println()指令。Serial.println() 可將資料顯示在 Arduino 軟體的監控視窗中;使用 Serial.println()指令前,必先 設定輸出的鮑率,這部分由 Serial.begin()指令來處理,更多的 Serial.println() 等相關使用會在串列通訊章節裡做更詳細的說明。監控視窗的使用如下說明:

打開 Arduino 軟體,載入程式碼後,執行程式上傳,並打開監控螢幕(Serial Monitor),如下圖 5-5:





本範例中鮑率設定為 9600·程式內的鮑率設定必須和監控螢幕上的鮑率設定一致, 不然會產生亂碼。當敲擊蜂鳴片時,監控視窗會在螢幕上跳行顯示「Knock!」。

有時候敲擊蜂鳴片時,LED 燈不會受到敲擊次數而控制亮滅,而且面板會顯示超過 1 次以上的「Knock」文字。這是因為蜂鳴片受到敲擊振動頻率太快,產生另一個電 壓數值訊號,迫使 LED 腳位狀態立刻更新發生瞬間亮滅。你可以調整門檻值來改善 這樣的情形。

-----www.aroboto.com------

(TEL) 02-25576923 (FAX) 02-25577529 Email: support@aroboto.com



第6章. 七段顯示器

介紹

本文介紹七段顯示器,並利用顯示器製造數字閃爍的效果。七段顯示器樣式,如下 圖 6-1。顯示器由八顆 LED 燈構成,內定配線牽引著 LED 接腳和導線,形成十條接 腳裸露在外。顯示器外接腳編號從接腳 a 到 g 以及接腳 dp,對應顯示器內 LED 燈編 號 a 到 g 以及 dp;中間兩端的接腳編號為 com。

七段顯示器分兩種規格,共陽跟共陰:共陽七段顯示器上面接腳 com 接電源;共陰 七段顯示器下面接腳 com 則接地。



圖 6-1. 七段顯示器

本範例將學習用 Arduino 範例程式控制七段顯示器上的 LED 閃爍。



使用材料

項次	材料名稱	數量
1	Arduino 載板	1
2	USB 連接埠	1
3	共陰七段顯示器	1
4	220 歐姆電阻	8
5	麵包板	1
6	跳線	若干

表格 2. 使用材料

接線與電路

電路配線:七段顯示器 LED 的接腳 a 到 g 以及接腳 dp 跨接 220 歐姆電阻後接到 arduino 數位 1/0 腳,從 P1 到 P8;共陰的接腳 com 為負極接地。電路如下圖 6-2。



圖 6-2. 七段顯示器的共陰接線



接線方式:共陰七段顯示器與 Arduino 的實際配線法,如下圖 6-3



圖 6-3. 共陰七段顯示器與 Arduino 的配線



```
int timer = 100;
                              //宣告 timer 變數・初始值=100
void setup(){
 //用一個 for 迴圈設定 P1 到 P7 每個接腳為輸出
 for (int thisPin = 1; thisPin < 8; thisPin++){
     pinMode(thisPin, OUTPUT);
  }
void loop(){
  //利用 for 迴圈,依序控制 P1~P7(遞增)輸出高電壓
  //造成 共陰七段顯示器 LED 從 a 到 g,皆亮燈
  for (int thisPin = 1; thisPin < 8; thisPin++) {
  digitalWrite(thisPin, HIGH);
   }
  delay(1000);
                                    //延遲1秒
  //利用 for 迴圈,依序控制 P7~P1(遞減)輸出低電壓
  //造成 共陰七段顯示器 LED 從 g 到 a,皆滅燈
  for (int thisPin = 1; thisPin < 8; thisPin++) {
  digitalWrite(thisPin, LOW);
  }
  delay(1000);
                                    //延遲1秒
```

說明

執行上述程式後七段顯示器會顯示數字 8 並且亮 1 秒然後滅 1 秒,如此重複亮滅; 從 P1 接腳到 P7 接腳給 High 時 LED 亮,給 LOW 時滅。

上述範例,從 P1 到 P7,每個 LED 的接腳,都跨接 220 歐姆電阻後接地。透過 for 迴圈一起控制每個 LED 亮減,因為一次控制顯示器上的七個 LED,看起來就像是顯 示數字 8。



第7章. 直流馬達-PWM 控制輸出

介紹

本文介紹如何控制直流馬達。利用 H 橋驅動晶片來控制馬達,包含使用 PWM 來控制直流 馬達的轉速變化、兩組數位 1/0 腳位控制馬達轉向。一般的馬達由永久磁鐵或電磁鐵、 電刷、整流子等元件構成,外觀樣式如下圖 7-1;馬達接腳 1 用黑線表示連接負極;接 腳 2 用紅線表示連接正極。



圖 7-1. 直流馬達

一般要做馬達驅動控制,會使用一個驅動晶片來控制,主要原因是馬達控制所需電 壓較高、電流較高、且需要控制方向與轉速。此晶片的作用可以讓馬達和電源形成 H橋接的方式,如圖 7-2 左邊 H的形狀。

H橋接可控制馬達的極性切換和運動方向,運作方式說明如下圖 7-2 左圖:如果開 關1和開關4接上,電流會從1經過馬達到4,如圖 7-2 中間;如果開關2和開關 3接上,電流會從2經過馬達到3,如圖 7-2 右邊。本實驗範例使用的L293D 晶片 就是使用H橋接。





L293D **晶片接腳樣式,如下圖** 7-3;



接腳 1 (1, 2EN): 1, 2EN 腳位用來啟動 1Y、2Y 腳位的功能。在控制馬達時,此腳位 會接上控制器的 PWM 輸出腳位,藉此來做轉速的控制。

接腳 2 (1A): 輸入控制訊號 (HIGH 或 LOW), 控制 1Y 腳位的狀態。

接腳3(1Y):接馬達紅線正極。

接腳 4 - 5: 接地。

接腳 6 (2Y): 接馬達黑線負極。

接腳7(2A):輸入控制訊號(HIGH或LOW),控制2Y腳位的狀態。

接腳 8 (VCC2): 接 3V 直流電。VCC2 對馬達作供電。

接腳 16 (VCC1): 接 5V 電源,對 L293D 作供電。



L293D 晶片會切換 H 橋接的開關。晶片腳 1 (1, 2EN) 接收 HIGH 準位時啟動 1Y、2Y 腳 位的功能;而 1A 為 LOW 且 2A 為 HIGH,馬達就正轉;相反的當 1A 為 HIGH 且 2A 為 LOW,馬達就反轉。晶片腳 1 (1, 2EN) 接收 LOW 準位時停止 1Y、2Y 腳位的功能。表格 3 用來說明 H 橋和馬達旋轉方向關係。

EN	1 A	2A	1Y	2Y	馬達旋轉方向
Н	L	Н	L	Н	正轉
Н	Н	L	Н	L	反轉
L	L	Н	Х	Х	無反應
L	Н	L	Х	Х	無反應

表格 3. H 橋接開關邏輯值

本範例將學習利用可變電阻、L293D 晶片和 Arduino 控制直流馬達的轉速。

使用材料

項次	材料名稱	數量
1	Arduino 載板	1
2	USB 連接埠	1
3	直流馬達	1
4	L293D 晶片	1
5	1K 歐姆電阻	1
6	可變電阻 10K 歐姆	1
7	按鈕	1
8	麵包板	1
9	1.5V 電池	2
10	電池盒	1
11	跳線	若干

表格 4. 使用材料



接線與電路

電路配線:可變電阻一端接到+5V,另一端接地,中間的接 Arduino 類比輸入接腳 A0。1,2EN 腳位接上 arduino 的 P11 腳位 (arduino 的 P3.5.6.9.10.11 都可以提供 PWM 訊號),提供 PWM 訊號給 L293。1A、2A 分別接上 arduino 數位 I/O 腳 P4、P5。 1Y、2Y 分別接上馬達兩端。

電源供應部分,控制電源與馬達電源,建議分開處理,Arduino 提供的 5V 的控制電 源給L293D,馬達所需要的 3V 電力另外由電池盒提供。在使用兩組供電源時,要注 意一點,電源供應端分開,但是兩組電源一定要作"共地",才會有相同的電位基 準。在接地時,線路配線是共點,電路如下圖 7-4。





接線方式:馬達與 Arduino 的實際配線法,如圖 7-5



圖 7-5. 馬達與 Arduino 的配線

程式

int motor 1 Pin = 4;	// 數位腳 4 接 1A 接腳
int motor $2Pin = 5;$	// 數位腳 5 接 2A 接腳
int enablePin = 11;	// PWM 接腳 11 接 EN 接腳
int resistorPin = 0 ;	// 可變電阻產生的電壓變化接類比輸入腳位 0
int resistorValue $= 0;$	//宣告 resistorValue・初始值 0



```
void setup() {
    pinMode(motor1Pin, OUTPUT); //設定 1A 腳位為輸出
    pinMode(motor2Pin, OUTPUT); //設定 2A 腳位為輸出
}
void loop() {
    //讀取 A0 腳位,並存在 resistorValue 變數中
    resistorValue = analogRead(resistorPin);
    //對 P11 腳位作 PWM 輸出,輸出值為 resistorValue/4
    analogWrite(enablePin, resistorValue/4);
    digitalWrite(motor1Pin, LOW); // 給 H 橋接腳 1 A, LOW
    digitalWrite(motor2Pin, HIGH); // 給 H 橋接腳 2A, HIGH
```

PWM 說明

在了解程式前,必須先了解 PWM 的用法。由於一般 MCU 控制器的世界哩,都是 0、1 的數位訊號,並無法提供類比電壓的輸出,因此常有人透過 PWM 以數位訊號的方式 來"模擬"類比電壓的輸出。

PWM 又稱做脈波寬度調整 (Pulse Width Modulation),藉由調整脈波寬度以數位訊 號模擬類比輸出。使用 PWM 時,會快速持續的產生出具有週期性的方波,圖 7-6 針 對方波來作介紹,週期裡的 HIGH 和 LOW 在方波內各佔不同的比例;方波 HIGH 的寬 度時間對應週期的時間比稱為責務週期 (Duty Cycle)。例如:HIGH 在方波寬度所佔 的時間是 1.5ms,方波週期為 2.0ms,所以責務週期為 (1.5/2.0) ×100% = 75%。 而這個 PWM 的頻率為 500Hz(每秒 500 個)。





透過改變方波的寬度,數位訊號可以模擬近似類比的訊號。如圖 7-7 的 1,責務週 期為 0%,模擬類比輸出電壓是 0V;圖 7-7 的 2,責務週期為 25%,模擬類比輸出電 壓接近 1.25V;圖 7-7 的 3;責務週期為 50%,模擬類比輸出電壓接近 2.5V;如圖 7-7 的 4,責務週期為 75%,模擬類比輸出電壓接近 3.75V;如圖 7-7 的 5,責務週期為 100%,模擬類比輸出電壓接近 5V。詳細的對應如表格 5。

Arduino 的 analogWrite()指令,設計用來作 PWM 的輸出(P5、P6 頻率為 1kHz;P9、 P10、P11、13 頻率為 490Hz;使用較高頻率,馬達控制的表現會較好)。使用者可 以很方便直接使用,不需要自行作處理。analogWrite()指令提供 8bit 的解析度, 也就是能將責務週期切分為 256 等份(2 的 8 次方)作使用。如果要輸出 75%的責務 週期,數值應該要填入 256×75%-1=191,因為從 0 開始算起。詳細的對應請參考表 格 5。



圖 7-7. 責務週期與類比電壓輸出關係圖



編號	責務週期 (%)	輸出電壓(V)	analogWrite()
1	0%	0	0
2	25%	1.25	63
3	50%	2.5	127
4	75%	3.25	191
5	100%	5	255

表格 5. 責務週期與類比輸出關係

程式說明

執行程式,然後轉動可變電阻,馬達轉速會有增減的變化,這是受到 PWM 輸出的控制影響。如圖 7-5,當可變電阻的旋鈕向左轉,resistorValue/4 的數值增加,PWM 的輸出增加,馬達的轉速增加;相反的當可變電阻的旋鈕向右轉,resistorValue/4 的數值減少,PWM 的輸出減少,馬達的轉速減少。

為何不直接使用 resistorValue 的數值, 而要使用 resistorValue/4?

這是因為 resistor Value 的數值來自於 arduino 類比電壓輸入(10bit ADC),該數 值範圍 0~1023;而 analogWrite 解析度只有 8bit→0~255,為了避免數值超過,因 此透過除以四來讓可變電阻產生的電壓值,等比例的縮小到適合的範圍中。

用按鈕控制馬達的方向

了解用 PWM 控制馬達轉速後,可以嘗試用按鈕切換馬達旋轉的方向。操作電路圖如 下圖 7-8。

電路配線:按鈕一端接到+5V,另一端跨接一個 1K 歐姆的電阻後接地,同時也接 Arduino 數位腳 P12;其他接腳的接法如圖 7-4 一樣。





附加按鈕後的操作程式如下:





```
void loop() {
```

```
//讀取可變電阻值,存放在 resistorValue 變數中
resistorValue = analogRead(resistorPin);
```

//**以** PWM 方式,設定馬達速度 analogWrite(enablePin, resistorValue/4);

//讀取按鈕的狀態,純放在 buttonValue 變數中 buttonValue = digitalRead(buttonPin);

```
if(buttonValue!=last_buttonVal){
```

```
//假如按鈕狀態改變,由LOW->HIGH,則 buttonValue為HIGH,
//此時執行內部程式
if(buttonValue==HIGH){
```

```
//切換 turn 變數的狀態,由 HIGH->LOW 或者 LOW->HIGH
turn =!turn;
//透過 LED 燈表示 turn 目前的狀態
```

```
digitalWrite(LED, turn);
```

```
}
```

//假如按鈕狀態改變,由 HIGH->LOW,則 buttonValue 為 LOW, //此時執行內部程式

```
else{
```

```
//不改變 turn 的狀態
```

```
//透過 LED 燈表示 turn 目前的狀態
```

```
digitalWrite(LED, turn);
```

```
}
```

```
//更新 last_buttonVal 的狀態
```

```
last_buttonVal= buttonValue;
```



	if(turn==HIGH){ digitalWrite(motor1Pin, HIGH)	// 如果 turn=HIGH, ;	控制馬達正轉
	digitalWrite(motor2Pin, LOW); }else{	; // 如果 turn=LOW	,控制馬達反轉
	digitalWrite(motor1Pin, LOW digitalWrite(motor2Pin, HIGH); I);	
	}		
//==			
}			

//===========控制馬達轉向=

上述程式中接腳 4 和接腳 5 是用來控制馬達的方向,當接腳 4 輸出 LOW、接腳 5 輸出 HIGH 時馬達正轉;當接腳 4 輸出 HIGH、接腳 5 輸出 LOW 時馬達反轉。

在程式中"//=====馬達方向切換判斷====="以下的程式碼是要透過按鈕來切 換馬達的方向:當讀取的按鈕狀態改變(與前一次按鈕狀態比較)且為新的按鈕狀態 為 HIGH 時則令馬達方向改變;其他條件下,馬達方向維持不變。

注意:按"按鈕"動作包含兩次狀態改變。第一次改變,按鈕按下時(手還未放開), LOW->HIGH;第二次改變,手放開按鈕彈回時,HIGH->LOW。

> ------www.aroboto.com------(TEL) 02-25576923 (FAX) 02-25577529

> > Email: support@aroboto.com



第8章. 伺服馬達控制

介紹

本文介紹伺服馬達,伺服馬達亦稱伺服機,一般用在遙控汽車、船、飛機等角度控 制機構中。其外觀構造如圖 8-1 所示。通常伺服機出力軸上會接「伺服擺臂」,如 圖 8-1 中伺服機上方的白色機構。

常用的伺服機有三條接線,在圖 8-1 中黑線接地線(GND),紅線接電源(6V),白線 為訊號線。伺服機內部由齒輪組(圖 8-2 的 1)和馬達、電路(圖 8-2 的 2)所構成。 馬達的運動由電路所控制再透過齒輪組輸出,可達到位置定點的控制。

伺服機的主要規格,除了尺寸大小外,扭力與速度也是選購伺服機時重要依據,使 用者須先了解自己所需要規的規格,再來決定伺服機的種類。以伺服馬達來說,扭 力與速度大大影響了價格,通常大扭力的價格是以指數型成長。



圖 8-1. 伺服馬達外觀





2 圖 8-2. 伺服馬達機構



伺服機的控制方式有別於一般的直流馬達,必須送出週期為 20ms (頻率為 50Hz)的 PWM 訊號,利用脈波寬度來控制伺服機轉動的角度,也就是前一章節所介紹 PWM 訊 號,但要注意的是周期為 20ms。Arduino 的 analogWrite()指令所輸出的 PWM,預 設的周期約為 2ms (P3、P9、P10、P11)或者 1ms (P5、P6),因此控制伺服機並不能 直接用 analogWrite()指令,而得透過 servo.h函式庫。

控制伺服馬達訊號的 PWM 脈波週期是 20ms,訊號的脈波寬度在 1.0ms 到 2.0ms 之間, 脈波的寬度決定伺服擺臂轉動的方向和角度;當輸出脈波寬度在 1.5ms 時,伺服馬 達會轉動到 90 度位置,如圖 8-3 左邊;若脈波寬度超過 1.5ms 會往一個方向轉動, 如:脈寬為 2.0ms 時,伺服馬達會轉動到 180 度位置,如圖 8-3 中間;若脈波寬度 小於 1.5ms 會往另一個方向轉動,如:脈寬為 1.0ms 時,伺服馬達會旋轉到 0 度位 置,如圖 8-3 右邊。



圖 8-3. 伺服馬達與訊號控制方式

一般的伺服馬達轉動方式如上文所述,但不同製造商的伺服馬達規格和馬達特性也 會有些差異,例如有些伺服馬達擺臂轉動的角度範圍超過 180 度甚至到 270 度,有 些則可能只有 90 度;另外,有些伺服機能夠回授伺服機的狀態等。通常伺服馬達 的操作電壓為 4.8V 到 6.0V,但也有伺服馬達操作電壓為 12V,本範例使用常用的 標準型伺服馬達 (6V 電源)與 Arduino 搭配的使用方式。



使用材料

項次	材料名稱	數量
1	Arduino 載板	1
2	USB 連接埠	1
3	伺服馬達	1
4	麵包板	1
5	1.5V 電池	4
6	跳線	若干

表格 8-1. 材料清單

以下範例將學習用 Arduino 內的 servo.h 函式庫控制伺服馬達旋轉。

接線與電路

電路配線:白線接 PWM 接腳 P9,紅線接 6V 的電池正極,黑線接電池負極(記得與控制器 Arduino 要共地)。 電路如圖 8-4。



圖 8-4. 伺服馬達與線路圖



接線方式:伺服馬達與 Arduino 的實際配線法,如下圖 5



圖 8-5. 伺服馬達在 Arduino 的配線



```
#include <Servo.h> //呼叫控制伺服馬達的函式庫
Servo servo_motor; //宣告伺服馬達變數
void setup(){
   servo_motor.attach(9); //伺服馬達接 PWM 腳位 9
   servo_motor.write(90); //控制伺服馬達到 90 度的位置
   delay(5000); // 延遲 5 秒
}
void loop(){
   servo_motor.write(0); //控制伺服馬達轉到 0 度位置
   delay(5000);
   servo_motor.write(180); //控制伺服馬達轉到 180 度的位置
   delay(5000);
```

說明

執行上述程式,伺服馬達的擺臂會先轉到 90 度位置等待 5 秒,接著轉到 0 度位置 等待 5 秒,再轉到 180 度等待 5 秒,如此重複擺動。

使用伺服馬達的程式指令需用到函式庫,所以在撰寫程式時要先呼叫伺服馬達的函 式庫,用#include <Servo.h>表示。#include 後面接的是標頭檔 Servo.h,代表所 呼叫的函式庫。函式庫在呼叫後,需要宣告命令集的名稱,上述程式裡 Servo 後面 接的是命令集的名稱 servo_motor。servo.h的命令有 servo_motor.attach()和 servo_motor.write(),其中 servo_motor.attach()是設定所使用的 PWM 腳位, servo_motor.write()是設定伺服馬達轉動的角度。有關 Servo 函式庫放在 Arduino-0018\ libraries\ Servo 資料夾中。



使用 servo_motor.write()可以使伺服馬達轉動到指定的角度,另外伺服馬達函式 庫裡也有其他程式指令可使伺服馬達到達類似的指定角度,例如 servo_motor.writeMicroseconds()也是其中一個控制命令,差別在於此命令是給 予脈波寬度大小進而使伺服馬達轉動到所要求的位置,參考程式如下。(此命令可 以控制的精度較高)。

#include <servo.h></servo.h>	
Servo servo_motor;	
void setup(){	
servo_motor.attach(9);	
servo_motor.writeMicroseconds(1500);	//指定脈波寬度為 1.5ms
delay(5000);	
}	
void loop(){	
servo_motor.writeMicroseconds(1020);	// 指定脈波寬度為 1.02ms
delay(5000);	
servo_motor.writeMicroseconds(1980);	//指定脈波寬度為 1.98ms
delay(5000);	
1	

-----www.aroboto.com-----

(TEL) 02-25576923 (FAX) 02-25577529 Email: support@aroboto.com



第9章. 音樂旋律(蜂鳴器或喇叭)

介紹

本文介紹如何透過蜂鳴器或喇叭來產生音樂旋律。本範例使用喇叭來產生音樂旋律, 喇叭如下圖 9-1;電路接線時,須注意喇叭背面標示接腳正極和負極。當給予喇叭 電壓,喇叭會產生震動而發出聲音;由於喇叭震動的頻率不同,因此音色也不會相 同。



圖 9-1. 喇叭

本範例將學習使用 Arduino 內建程式,使用數位接腳 P8 控制喇叭產生音樂。

使用材料

項次	材料名稱	數量
1	Arduino 載板	1
2	USB 連接埠	1
3	喇叭	1
4	220 歐姆電阻	1
5	麵包板	1
6	跳線	若干

表格 9-1. 材料清單



接線與電路

電路配線:喇叭接腳屬於正極的一端,跨接 220 歐姆電阻後接數位腳 P8;接腳屬於 負極的一端接地。如下圖 9-2



接線方式:喇叭與 Arduino 的實際配線法,如下圖 9-3。





```
Melody
  Plays a melody
  circuit:
 * 8-ohm speaker on digital pin 8
 created 21 Jan 2010
 by Tom Igoe
 This example code is in the public domain.
 http://arduino.cc/en/Tutorial/Tone
*/
#include "pitches.h"
                                     //呼叫 pitches.h 標頭檔
// 設定音符配合的旋律:
int melody[] = {
  NOTE_C4, NOTE_G3, NOTE_G3, NOTE_A3, NOTE_G3,0,
NOTE_B3, NOTE_C4};
// 節拍設定:4 = 四分音符,8 = 八分音符,以此類推
int noteDurations[ ] = {
  4, 8, 8, 4, 4, 4, 4, 4 };
void setup() {
  for (int thisNote = 0; thisNote < 8; thisNote++) {</pre>
    int noteDuration = 1000/noteDurations[thisNote];
/設定單一音符的節拍
//設定單一音符輸出腳位,順序,節拍
    tone(8, melody[thisNote],noteDuration);
    int pauseBetweenNotes = noteDuration *1.30;
    delay(pauseBetweenNotes);
  //每個音符相隔的時間
void loop() { }
```



pitches.h

#define NOTE_B0 31 #define NOTE_C1 33 #define NOTE_CS1 35 #define NOTE_D1 37 #define NOTE_DS1 39 #define NOTE_E1 41 #define NOTE_F1 44 #define NOTE_FS1 46 #define NOTE_G1 49 #define NOTE_GS1 52 #define NOTE_A1 55 #define NOTE_AS1 58 #define NOTE_B1 62 #define NOTE_C2 65 #define NOTE_CS2 69 #define NOTE_D2 73 #define NOTE_DS2 78 #define NOTE_E2 82 #define NOTE F2 87 #define NOTE_FS2 93 #define NOTE_G2 98 #define NOTE_GS2 104 #define NOTE_A2 110 #define NOTE_AS2 117 #define NOTE_B2 123 #define NOTE_C3 131 #define NOTE_CS3 139 #define NOTE_D3 147 #define NOTE_DS3 156 #define NOTE_E3 165 #define NOTE_F3 175 #define NOTE_FS3 185



#define NOTE_G3 196 #define NOTE_GS3 208 #define NOTE_A3 220 #define NOTE_AS3 233 #define NOTE_B3 247 #define NOTE_C4 262 #define NOTE_CS4 277 #define NOTE_D4 294 #define NOTE_DS4 311 #define NOTE_E4 330 #define NOTE_F4 349 #define NOTE_FS4 370 #define NOTE_G4 392 #define NOTE_GS4 415 #define NOTE_A4 440 #define NOTE_AS4 466 #define NOTE_B4 494 #define NOTE_C5 523 #define NOTE_CS5 554 #define NOTE_D5 587 #define NOTE_DS5 622 #define NOTE_E5 659 #define NOTE F5 698 #define NOTE_FS5 740 #define NOTE_G5 784 #define NOTE_GS5 831 #define NOTE_A5 880 #define NOTE AS5 932 #define NOTE_B5 988 #define NOTE_C6 1047 #define NOTE CS6 1109 #define NOTE_D6 1175 #define NOTE DS6 1245 #define NOTE_E6 1319 #define NOTE_F6 1397 #define NOTE_FS6 1480



#define NOTE_G6 1568 #define NOTE_GS6 1661 #define NOTE_A6 1760 #define NOTE_AS6 1865 #define NOTE_B6 1976 #define NOTE_C7 2093 #define NOTE_CS7 2217 #define NOTE_D7 2349 #define NOTE_DS7 2489 #define NOTE_E7 2637 #define NOTE_F7 2794 #define NOTE_FS7 2960 #define NOTE_G7 3136 #define NOTE_GS7 3322 #define NOTE_A7 3520 #define NOTE_AS7 3729 #define NOTE_B7 3951 #define NOTE_C8 4186 #define NOTE_CS8 4435 #define NOTE_D8 4699 #define NOTE_DS8 4978

說明

執行上述程式,喇叭會播放一段音樂後停止。程式中使用命令 tone (),括號內是設 定輸出腳位、音符的頻率、音符持續的時間。程式中運用時間延遲命令 delay ();, 為控制每個音符相隔時間的長短。



第10章. LCD 顯示

介紹

本文介紹 LCD 顯示器・並利用 LCD 顯示文字。本範例使用 LCD 顯示器規格為 16×2・ 藍底白字有背光的 LCD 顯示器・接腳圖樣如下圖 10-1;接腳使用方式如下表格 2。



圖 10-1.LCD 顯示器接腳樣式 (16×2LCD 顯示器)

接腳編號	接腳標示	使用方式
1	VSS	接地
2	VDD	電源 5V
3	V0	信號顯示與螢幕對比強度
4	RS	暫存器選擇 HIGH 或 LOW
5	R/W	讀/寫信號 HIGH 或 LOW
6	E	Enable 接腳接收 HIGH 或 LOW
7	DB0	接收訊號 HIGH 或 LOW
8	DB1	接收訊號 HIGH 或 LOW
9	DB2	接收訊號 HIGH 或 LOW
10	DB3	接收訊號 HIGH 或 LOW
11	DB4	接收訊號 HIGH 或 LOW
12	DB5	接收訊號 HIGH 或 LOW
13	DB6	接收訊號 HIGH 或 LOW
14	DB7	接收訊號 HIGH 或 LOW
15	А	+4.2V 供應 LED 電源
16	К	供應 LCD 背光 (接地)

表格 2.LCD 接腳的使用方式

-----www.aroboto.com-----

(TEL) 02-25576923 (FAX) 02-25577529

Email: support@aroboto.com



本範例將學習用 Arduino 控制 16×2 的 LCD 顯示器,讓文字依序閃爍。

使用材料

項次	材料名稱	數量
1	Arduino 載板	1
2	USB 連接埠	1
3	LCD 顯示器	1
4	10K 可變電阻	1
5	麵包板	1
6	跳線	若干

表格 3. 材料清單

接線與電路

電路配線如下圖 10-2,其中 V0 為調整顯示器的對比,接可變電阻後接地;另外圖 10-2 中 K (接+5V)、A (接地)是用來供應螢幕的背光。中間的部分 DB0 到 DB3 在本範 例中不使用,空接。



圖 10-2.LCD 與 Arduino 電路圖



接線方式:LCD 顯示器與 Arduino 的實際配線,如下圖 10-3



圖 10-3. LCD 顯示器與 Arduino 的配線

程式

#include <liquidcrystal.h></liquidcrystal.h>	//使用函式庫	
int time=500;	//宣告字母顯示的延遲時間	
//=設定 LCD 對應到 Arduino 的	り接腳 (rs, rw, enable, d4, d5, d6, d7)==	
LiquidCrystal lcd(12, 11, 10, 5,	4, 3, 2);	
void setup(){		
lcd.begin(16,2);	//設定 LCD 螢幕規格	
<pre>lcd.print("Character print");</pre>	//在 LCD 上顯示文字	
delay(2000);		
}		



```
void loop(){
 //顯示字元 A 到 Z
 for(int Character=65; Character<=90; Character+=1){</pre>
   if(Character \le 80){
                             //螢幕第1列顯示A到P
    lcd.clear();
                              //清除螢幕
    lcd.setCursor(Character-65,0); //文字顯示從第1行到第16行
    lcd.write(Character);
    delay(time);
   }else{
                              //螢幕第 2 列顯示 Q 到 Z
    lcd.clear();
    lcd.setCursor(Character-81,1); //文字顯示從第1行到第16行
    lcd.write(Character);
    delay(time);
```

說明

執行上述程式,LCD 顯示器會顯示文字「Character print」,接著列印英文字母從 A 到 Z。每個字母會顯示然後消失,每隔 0.5 秒顯示在下一個位置上,如此重複顯示。

本範例使用到 LCD 顯示器,在撰寫程式時要先呼叫 LCD 的函式庫,以#include <LiquidCrystal.h>表示。



有關 LCD 的控制命令如下:

LCD 的控制命令	說明
LiquidCrystal()	設定 LCD 與 Arduino 接腳的對應關係
lcd.begin()	設定 LCD 顯示器螢幕尺寸寬和高(欄和列)
lcd.clear()	清除 LCD 螢幕上字幕的顯示
lcd.home()	設定游標在螢幕的左上方
lcd.setCursor()	設定 LCD 上文字的顯示位置
lcd.write()	將字元顯示在 LCD 螢幕上
lcd.print()	在 LCD 上顯示文字
lcd.cursor()	在 LCD 螢幕上顯示游標
lcd.noCursor()	在 LCD 螢幕上取消游標
lcd.lcd.blink()	游標在 LCD 螢幕上閃爍
lcd.noBlink()	游標在 LCD 螢幕上不閃爍
lcd.display()	打開 LCD 的顯示
lcd.noDisplay()	關掉 LCD 的顯示
lcd.scrollDisplayLeft()	文字向左轉動
lcd.scrollDisplayRight()	文字向右轉動
lcd.autoscroll()	打開自動轉動(初始設定由右向左)
lcd.noAutoscroll()	打開自動轉動(初始設定由左向右)
lcd.leftToRight()	LCD 寫入字元從到右
lcd.rightToLeft()	LCD 寫入字元從右到左
lcd.createChar()	控制單一字元方格內每個小點的顯示,產生新的字元

-----www.aroboto.com-----

(TEL) 02-25576923 (FAX) 02-25577529 Email: support@aroboto.com



第11章.數位輸出擴充

介紹

本範例介紹數位輸出擴充,利用 74HC595 晶片透過 3 組 10 腳位,控制 8 組 LED 燈 閃爍。Arduino 有 13 組數位輸出腳,扣除 UART 埠,剩下 11 組;如果要使用超過 11 個數位訊號,就必須使用晶片來擴充數位輸出,如 74HC595 晶片。74HC595 晶片 接腳如下圖 11-1,腳位定義如下表格 11-1;如果需要控制 8 組以上的 LED,可以 透過串接 74HC595 晶片的方式,控制 16 組、24 組等更多組的數位輸出。



符號	晶片腳位	功能	接腳
Q0 to Q7	15, 1~7	數位輸出腳	LED
GND	8	0V 接地	接地
Q7 [*]	9	串列資料輸出	晶片串接時使用
MR	10	Master Reset	+5V
SH_CP	11	shift register clock input	P12
ST_CP	12	storage register clock input	P8
0E	13	輸出啟動 En	GND
DS	14	串列資料輸入	P11
VCC	16	操作電源	+5V

表格 11-1.75HC595 腳位

------www.aroboto.com------(TEL) 02-25576923 (FAX) 02-25577529

Email: support@aroboto.com



本範例透過 74HC595 晶片, 讓 Arduino 只使用 3 組 10 腳位, 卻可以控制 8 組 LED 燈閃爍。

使用材料

項次	材料名稱	數量
1	Arduino 載板	1
2	USB 連接埠	1
3	74HC595 晶片	2
4	麵包板	1
5	LED	16
6	470 歐姆電阻	16
7	10K 電阻	1
8	跳線	若干

表格 11-2. 材料清單

接線與電路

電路配線: 74HC595 晶片上, VCC 接 Arduino 所提供的 5V, 電路如下圖 11-2。



Email: support@aroboto.com



接線方式: 74HC595 與 Arduino 的實際配線, 如下圖 11-3。



圖 11-3.74HC595 與 Arduino 的配線

程式

int latchPin = 8;	//數位腳 P8 接 74HC595 的接腳 ST_CP				
int clockPin = 12;	//數位腳 P12 接 74HC595 的接腳 SH_CP				
int dataPin = 11;	// 數位腳 P11 接 74HC595 的接腳 DS				
void setup() {					
pinMode(latchPin, OUTPUT);					
pinMode(clockPin, OUTPUT);					



```
void loop() {
  for (int numberToDisplay = 0; numberToDisplay <255;
numberToDisplay++) {
    digitalWrite(latchPin, LOW);
    shiftOut(dataPin, clockPin, MSBFIRST, numberToDisplay);
    digitalWrite(latchPin, HIGH);
    delay(500);
  }
}</pre>
```

pinMode(dataPin, OUTPUT);

說明

執行上述程式,74HC595 晶片會接收 Arduino 板輸出的數位訊號,LED 燈會依據 numberToDisplay 的數值作呈現。

程式中的 numberToDisplay 是以十進位的格式來表示,其實並不好理解哪一顆 LED 會亮,哪一顆 LED 會滅,如果改以二進位的方式來看,就一目了然。請參考以下表 格 11-3。

當 numberToDisplay=1 時,二進位為 0000001,所以 LED0 亮起。

當 numberToDisplay=2 時,二進位為 0000010,所以 LED1 亮起。

當 numberToDisplay=3 時,二進位為 00000011,所以 LED0、LED1 都亮起。

當 numberToDisplay=4 時,二進位為 00000100,所以 LED2 亮起。

當 numberToDisplay=8 時,二進位為 00001000,所以 LED3 亮起。

以此類推,應該就可以了解控制 LED 亮滅的依據。

A State	藝科資訊工作室
	Aroboto.com

numberToDisplay	LED7	LED6	LED5	LED4	LED3	LED2	LED1	LED0
0	0	0	0	0	0	0	0	0
1	0	0	0	0	0	0	0	1
2	0	0	0	0	0	0	1	0
3	0	0	0	0	0	0	1	1
4	0	0	0	0	0	1	0	0
5	0	0	0	0	0	1	0	1
6	0	0	0	0	0	1	1	0
7	0	0	0	0	0	1	1	1
8	0	0	0	0	1	0	0	0
9	0	0	0	0	1	0	0	1
10	0	0	0	0	1	0	1	0
2	0	0	0	0	0	0	1	0
4	0	0	0	0	0	1	0	0
8	0	0	0	0	1	0	0	0
16	0	0	0	1	0	0	0	0
32	0	0	1	0	0	0	0	0
- 64	0	1	0	0	0	0	0	0
128	1	0	0	0	0	0	0	0
255	1	1	1	1	1	1	1	1

表格 11-3. 十進位與二進位對照表

程式中使用命令 shiftOut(),括號內是設定資料輸出腳位、輸出時脈的接腳、資料 傳輸的方向 (MSBFIRST 或 LSBFIRST)、資料的數值。

-----www.aroboto.com-----

(TEL) 02-25576923 (FAX) 02-25577529 Email: support@aroboto.com



如果要使用多組 74HC595 晶片做擴充輸出,電路如下圖 11-4。



圖 11-4. 多組擴充 IC 與 Arduino 結合電路圖



用 74HC595 晶片做數位擴充輸出後的操作程式如下:

```
int latchPin = 8;
                      //數位腳 P8 接 74HC595 的接腳 ST_CP
int clockPin = 12;
                      //數位腳 P12 接 74HC595 的接腳 SH_CP
int dataPin = 11;
                      //數位腳 P11 接 74HC595 的接腳 DS
byte D1=B1000000;
                    //以二進位來表示 D1
byte D2=B00000001;
                    //以二進位來表示 D2
void setup() {
  pinMode(latchPin, OUTPUT);
  pinMode(clockPin, OUTPUT);
  pinMode(dataPin, OUTPUT);
  shiftOut(dataPin, clockPin, MSBFIRST, D1);
  shiftOut(dataPin, clockPin, MSBFIRST, D2);
  digitalWrite(latchPin, HIGH);
 delay(300);
void loop() {
  for (int i =1 ; i<3;i++) {
                                     //設定移動方向
    for(int j=1; j<8; j++){
                                     //LED 燈亮滅的順序
     if(i==1){
       D1=D1>>1;
                                     //D1 向右移動 1 個位元
       D2=D2<<1:
                                     //D2 向左移動 1 個位元
      }
      else{
       D1=D1<<1;
                                     //D1 向左移動 1 個位元
       D2=D2>>1;
                                     //D2 向右移動 1 個位元
      digitalWrite(latchPin, LOW);
      shiftOut(dataPin, clockPin, MSBFIRST, D1);
      shiftOut(dataPin, clockPin, MSBFIRST, D2);
      digitalWrite(latchPin, HIGH);
      delay(300);
```