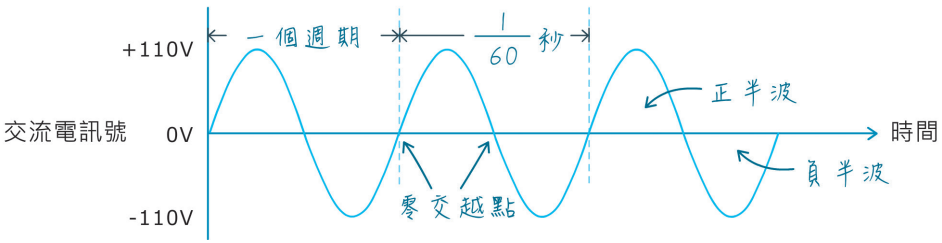
APPENDIX

C

交流電調光器製作

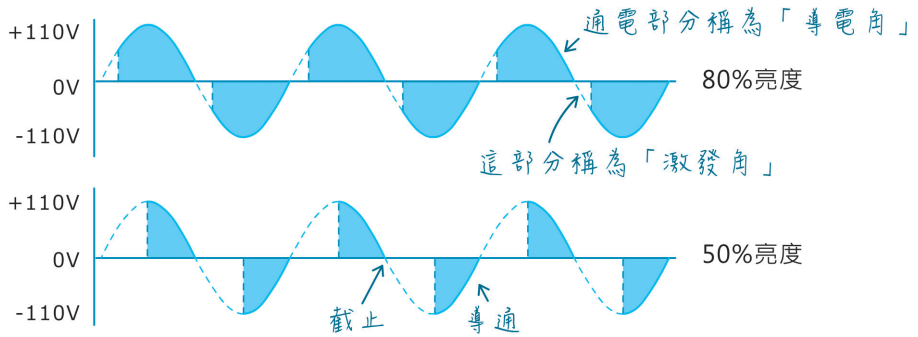
C-1 調整交流電的輸出功率

住家牆壁上的電源插座所提供的是**交流電**（Alternating Current，簡稱 AC），這種電源的大小與正負方向都會週期性地變化。台灣家庭的交流電有 110V 和 220V 兩種，頻率週期則是 60Hz（亦即，每秒鐘變化 60 次）。



從上圖可以看出，一個完整的交流電波形（稱為「全波」）是由正半波和負半波構成，波形和 0V 的交會點，稱為**零交越點（zero cross）**，稍後介紹的交流電控制需要使用「零交越點」當做參考點，並且透過一個電路元件偵測零交越點。

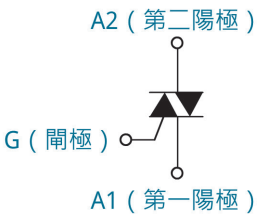
用 Arduino 或者其他微電腦裝置來控制交流負載（如：電燈泡）的開關，採用繼電器就行了。但若要調整交流負載的輸出功率（如：燈泡的亮度或者電風扇的轉速），則需採用如下圖一般，類似 PWM 的相位（或稱為「截波」）控制來調整供電的比例：

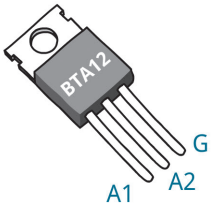


讀者可觀察到，不論正、負半波，**電流總是在零交越點截止**。「導電角」所佔的比例越高，代表開啟負載的時間越長（燈泡也更亮）。

使用 TRIAC 元件控制交流電設備

相較於 Arduino 微電腦的電源（5V, 0.5A），交流電負載的電壓和電流通常都比較大（如：110V, 5A），我們採用的控制元件稱為 **TRIAC（Tri-Electrode AC Switch**，中文譯名為「三極交流開關」或「雙向性三極閘流體」）。TRIAC 的外觀和電晶體相同：





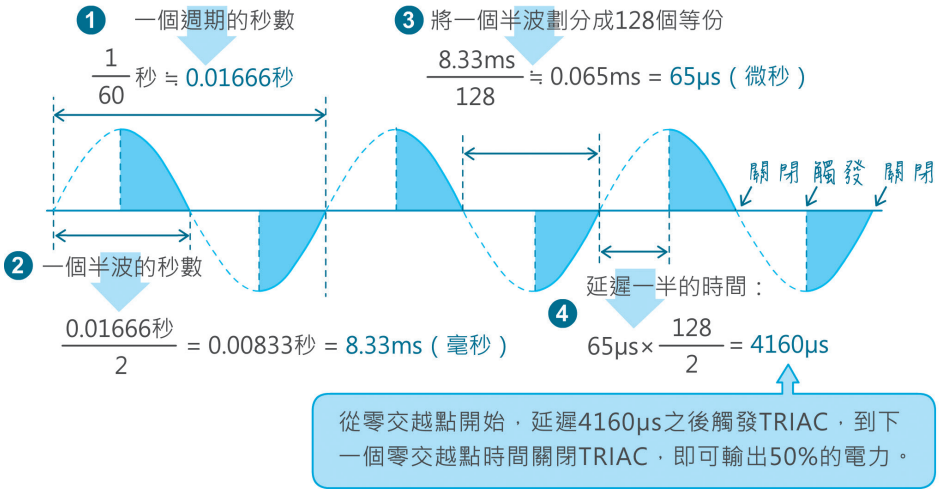
請將 TRIAC 看待成控制交流負載用的電子開關。因為交流電包含正負電流，所以 TRIAC 的符號由兩個不同方向的二極體組成，代表能讓正反向的電流通過。**G（閘極）是控制訊號輸入端，控制 A1 和 A2 是否導通**；在 G 極輸入正電位或者負電位，都能讓 A1 和 A2 導通。導通之後，它將維持導通狀態，直到 A1 和 A2 的電流低於某個臨界值或者 0（相當於力道無法推開二極體的閘門），TRIAC 就自動截止。

本文採用的 TRIAC 型號為 **BTA12-600B**，根據原廠的技術文件指出，它能容許 12A 的電流通過，用它來控制一般的白熾燈泡游刃有餘（註：筆者使用 20W 的燈泡測試；本文的相位控制電路無法控制普通的 LED 節能燈泡，也不能控制「電感」型負載，像交流馬達、電風扇）。

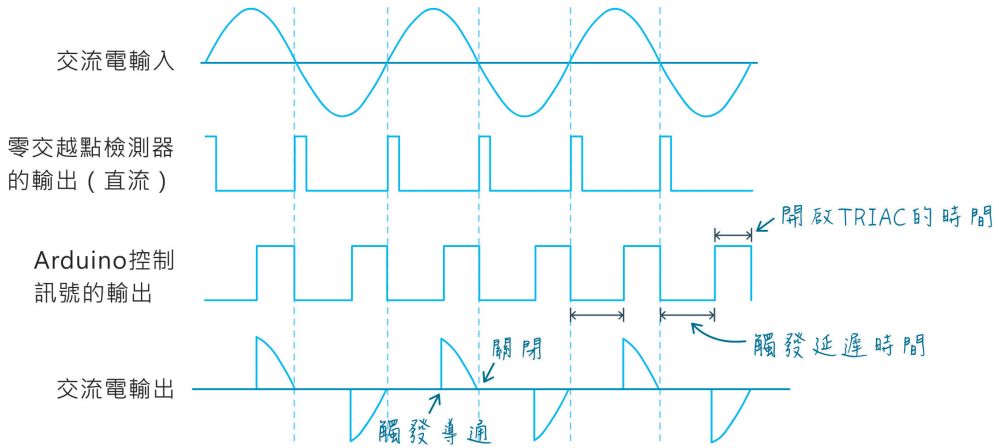
控制交流電的輸出電壓時，每次都要以**零交越點**為基準，來調整截止時間（激發角）和導通（導電角）的比例。如果不這麼做，被控制的燈泡將只會閃爍，而不是亮度產生變化。

交流電調光器程式的運作原理

假設我們要製作一個具備 128 段（0~127）的調光器，並預設讓它輸出 50% 的電力。從底下的計算式可得知，我們需要在每個**零交越點之後**延遲 4160 微秒，再觸發 TRIAC 導通：



調光器每**調高**或降低一段，延遲時間就要**減少**或增加 65 微秒（**延遲時間越短，電力輸出越高**）。如果把所有輸出 / 入訊號分開來看的話，它們的波形長像這樣：



第 3 章〈延遲與凍結時間〉一節提到，延遲微秒的 delayMicroseconds() 指令，超過 16383 µs 就不準確了，而且誤差隨著時間推移增大。若用它來實作本單元的調光器程式，燈光會在調整過程中發生閃爍的現象。因此，我們必須採用其他延遲觸發執行程式的方法。

C-2 定時觸發執行的 TimerOne 程式庫

UNO R3 的 ATmega328 微控器內部有三個計時器（timer），TimerOne 程式庫集合了一組用於設置和運用微控器 Timer1 計時器的程式碼，最基本的用法就是讓程式定時去觸發執行某一項工作。

TimerOne 程式庫快速上手：定時點滅 LED

本節將以 TimerOne 程式庫提供的 LED 閃爍程式，來說明此程式庫的使用方式。請先在**程式庫管理員**搜尋並安裝 "TimerOne"：

使用 TimerOne 程式庫所提供的各項指令之前，必須先執行底下的敘述，進行初始化：

```
Timer1.initialize(微秒);
```

其中的「微秒」參數，**最大可能值是 8388480（約 8.3 秒）**，若不設定參數，則採用預設值 1000000（1 秒）。初始化之後，即可透過 attachInterrupt()

指令，設定要定時觸發的中斷常式，第二個「微秒」參數是選擇性的，可不填寫。

```
Timer1.attachInterrupt(中斷常式, 微秒);
```

請選擇 Arduino IDE 主功能表的『**檔案 / 範例 /TimerOne/ISRBlink**』範例程式，其主程式片段如下：

```
#include <TimerOne.h>

void setup() {
  pinMode(13, OUTPUT);
  Timer1.initialize(100000);
  Timer1.attachInterrupt( timerIsr );
}

void loop(){
}
```

必須先執行這個指令（初始化）

0.1秒

每0.1秒觸發執行這個函數

透過 XOR（互斥或）來達成切換開關功能

ISRBlink 範例的 timerIsr() 自訂函式當中，包含一段開、關第 13 腳 LED 的敘述，它把目前第 13 腳的狀態（0 或 1），和 1 做 **XOR 運算（指令寫法：^）**，因此每一次執行這個敘述，第 13 腳的輸出就會和上一次相反。底下是自訂函式 timerIsr() 的內容說明：

```
void timerIsr() {
   digitalWrite( 13, digitalRead( 13 ) ^ 1 );
}
```

執行XOR運算

讀取13腳的狀態

輸入		XOR輸出
0	1	1
1	1	0

補充說明，**Timer1 計時器也負責控制數位 9 和 10 腳的 PWM 頻率**，所以，採用此程式碼時，不要將控制輸出接在這些數位腳。

上傳此程式碼到 UNO R3 板，第 13 腳的 LED 將快速閃爍。

動手做 C-1 交流電調光器電路

實驗說明：交流電實驗有危險性，所以本單元並未納入實作。筆者已組裝並驗證本單元的電路無誤，不過，在你組裝好電路、通電之前，請先確實採用電錶的歐姆檔，查看電源的輸入、輸出接腳是否有不該短路的地方（若**電錶顯示 0 歐姆，代表短路**）。

此外，建議讀者使用具有保險絲的電源延長線來連接本單元的電路，避免因為短路或其他狀況而發生危險。

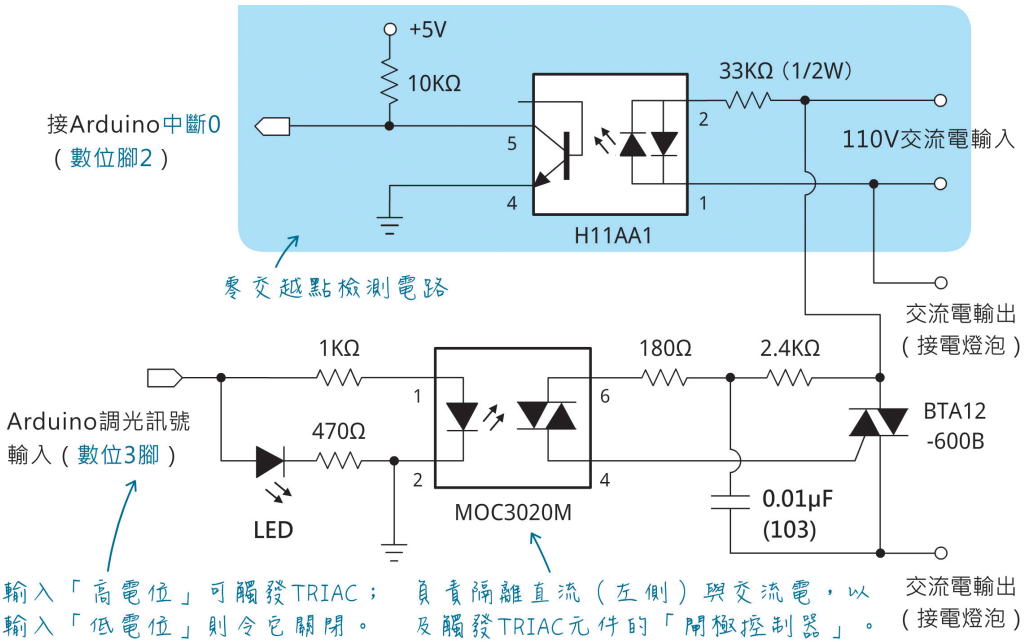
實驗材料：

10KΩ 可變電阻	1 個
180Ω（棕灰棕）1/4W 電阻	1 個
10KΩ（棕黑橙）1/4W 電阻	1 個
2.4KΩ（紅黃紅）1/4W 電阻	1 個
1KΩ（棕黑紅）1/4W 電阻	1 個
470Ω（黃紫棕）1/4W 電阻	1 個
33KΩ（橙橙橙）1/2W 電阻	1 個
0.01μF (103) 耐電壓 400V 的塑膠電容	1 個
LED（顏色不拘）	1 個
H11AA1 零交越檢測元件 （或者 4N25，搭配橋式整流器，請參閱下文說明）	1 個
MOC3020M 閘極控制元件	1 個
BTA12-600B TRIAC	1 個

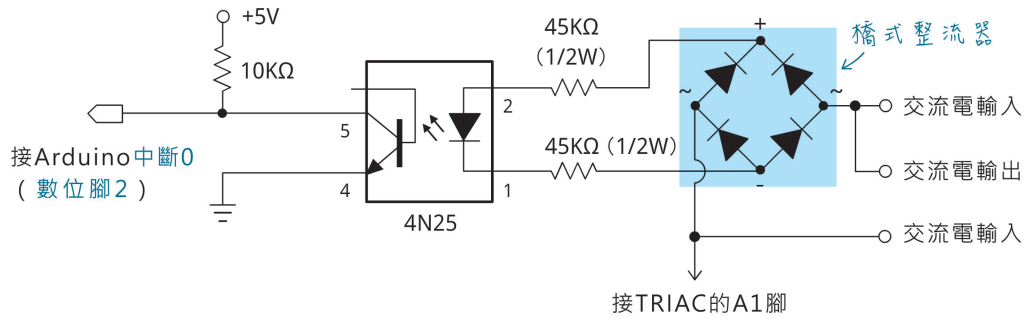
此外，你還需要自行剪裁一對 110V 的電源線和插座（五金行有售）。

實驗電路：底下是典型的交流相位控制電路，讀者可在網路上搜尋關鍵字 "arduino AC dimmer"（註：dimmer 代表「調光器」）便能找到其他類似的電

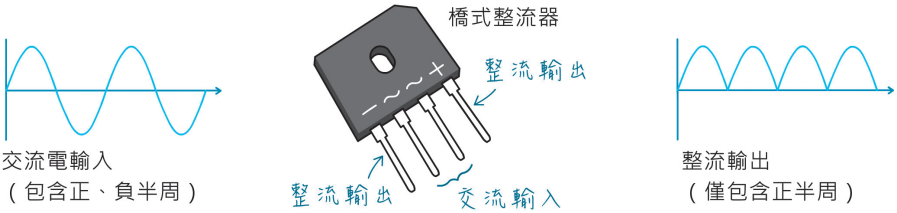
路。這個電路分成兩個部分，上半部採用 H11AA1 元件偵測零交越點，**每當偵測到零交越點，H11AA1 會輸出「高電位」**，平時則維持在低電位。



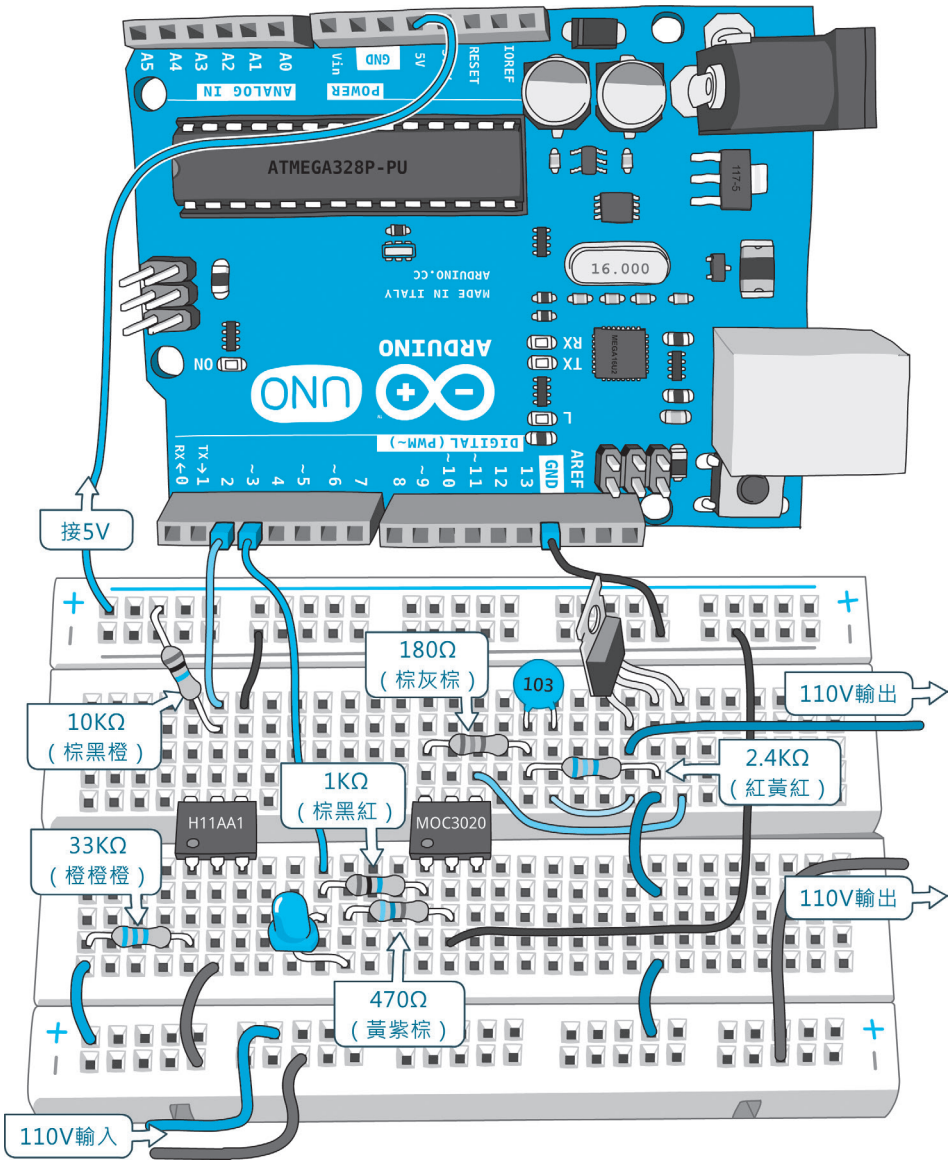
底下是另一種零交越點檢測電路，採用 4N25 代替 H11AA1 元件，TRIAC 控制電路和上圖的下半部相同。



由於 4N25 光隔離器只接收順向電流輸入，所以輸入訊號經過橋式整流器處理。橋式整流器內部由 4 個二極體組成，能把包含正、負電位的交流電訊號全轉變成正電位；你也可以用 4 個二極體（如：1N1004）代替一個橋式整流器。

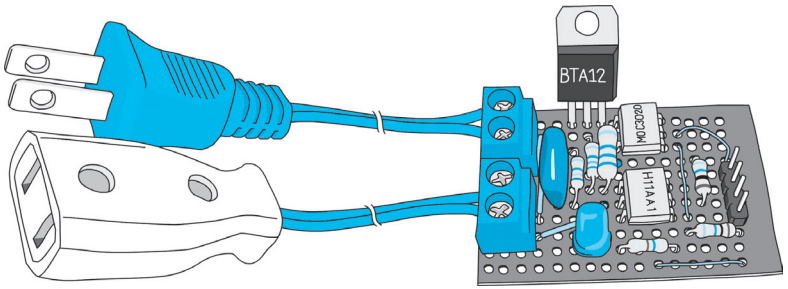


底下是採用 H11AA1 元件的麵包板電路連接示範：



另外，請在 A0 類比腳接一個 10KΩ 可變電阻來調整亮度。

110V 交流電輸入端就是一般的電源插頭，而 110V 輸出則是接電燈泡（或者如下圖的電源插座）。筆者直接把這個電路焊接在萬用 PCB 板：



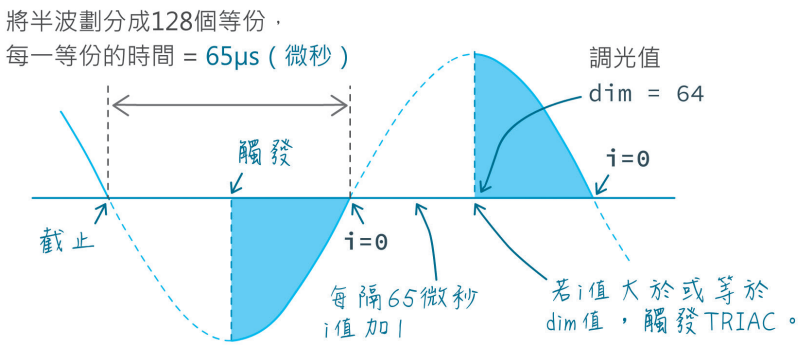
實驗程式：一開始先引用 TimerOne 程式庫，並宣告下列變數：

```
#include <TimerOne.h>    // 引用 TimerOne 程式庫

int dim = 64;            // 調光器的階段值（0-128），128 代表關閉
const byte acPin = 3;       // TRIAC 訊號輸出接腳
const byte potPin = A0;     // 可變電阻的接腳

volatile bool zeroCross=0; // 儲存零交越狀態的變數
volatile int i=0;          // 計算關閉 TRIAC 的延遲時間的計數器
```

本單元程式的原理如下，我們將設置一個每隔 65 微秒觸發執行的程式，每執行一次，就將變數 i 值加 1，並且判斷 i 值是否等於或大於調光變數 dim。



如果 i 值大於或等於 dim 變數值，隨即觸發 TRIAC，否則，TRIAC 維持在關閉狀態。本體程式碼如下：

```
void setup() {
   pinMode(acPin, OUTPUT);     // TRIAC 的控制輸出腳
   attachInterrupt(0, zeroCrossISR, RISING); // 偵測零交越訊號
   /*
     初始化 TimerOne 程式庫的 Timer1 定時觸發程式，
     參數 65 代表定時器的運作週期是 65 微秒。
   */
   Timer1.initialize(65);
   // 設定讓定時器每隔 65 微秒，自動執行 dim_check 函式
   Timer1.attachInterrupt(dim_check);
}

// 每當偵測到零交越點，底下的函式就會被執行
void zeroCrossISR() {
   zeroCross = true;
   i=0;
   digitalWrite(acPin, LOW);  // 關閉 TRIAC
}

// 底下的函式將每隔 65 微秒觸發一次
void dim_check() {
   if(zeroCross) {              // 若已經過零交越點....
      if(i>=dim) {              // 判斷是否過了延遲觸發時間...
         digitalWrite(acPin, HIGH); // 開啟 TRIAC
         i=0;                       // 重設「計數器」
         zeroCross=false;
      } else {
         i++;     // 增加「計數器」
      }
   }
}

void loop() {
   // 讀取可變電阻的值（0~1023），除以 8 可得到 128 階段值
   dim = analogRead(potPin) / 8;
}
```

實驗結果：編譯並上傳程式碼之後，插上交流電的插座和電燈泡，即可透過可變電阻來調整燈泡的亮度。再次叮嚀，記得要注意用電安全哦！

MEMO