

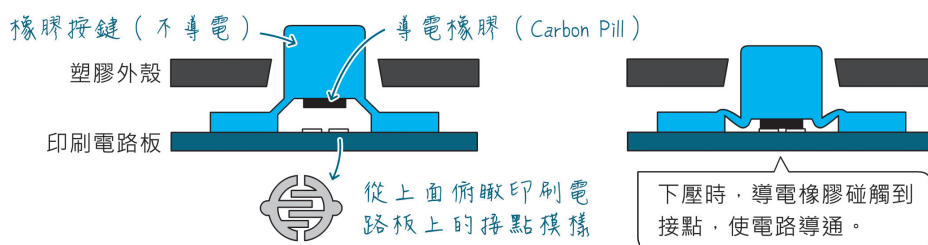
APPENDIX

E

改造 3C 小玩意 的控制鈕

許多 3C 產品越做越小價格也越來越低廉，像錄音筆和鑰匙圈形式的微型攝影機，如果你想要進行小小的改造，從 Arduino 控制這些裝置，例如，偵測到有人靠近時啟動微型攝影機錄影。其實，只要修改這些裝置的開關迴路即可。

這些小玩意的控制鈕的結構大致相同，如果將它們拆解（註：拆解之後，可能會失去保固），你將看到類似下圖的開關結構，當按鈕被按下時，按鍵底部的**導電橡膠**會讓印刷電路板上的接點導通：



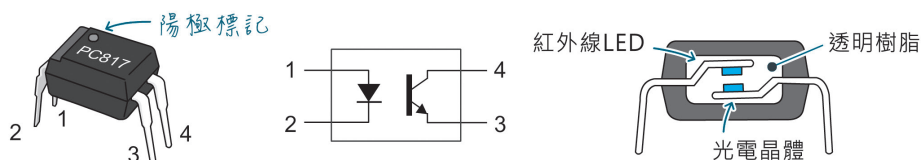
我們並不需要用馬達或其他機械裝置來按下或觸發開關。印刷電路板上的銅箔接點，相當於電路上的一個斷路，因此，若在它兩邊各焊接一條導線出來，再連接開關，即可取代原有的按鍵：



第 13 章提到**電晶體相當於數位開關**，包含之前提到的光電晶體在內，本單元將採用稱為「光電耦合」的元件來取代控制器原有的按鍵。

E-1 認識光電耦合元件

光電耦合（optical coupler）元件，又稱為**光隔離**元件，外型、電路符號與結構如下：



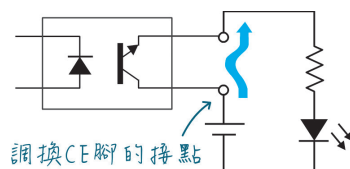
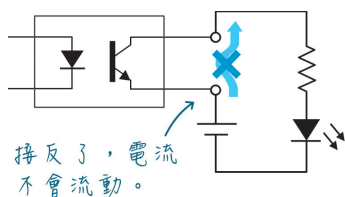
它也算是一種**光電開關**，當左邊的紅外線二極體導通、發光時，右邊的光電晶體也將導通。由於元件兩端的信號，全透過光線傳遞，沒有直接相連，因此元件兩端的電路相當於被**隔離**開來，不互相干擾。

以底下的應用為例，光電晶體一端接微型攝影機的按鈕接點，不管這小裝置採用 1.5V 或 9V 電源，都跟 Arduino 無關：



常見的光耦合元件型號有 NEC 的 PS2501、夏普的 PC817 以及台灣億光的 EL817，這三種型號的元件可以互換，外型與電路符號也相同。

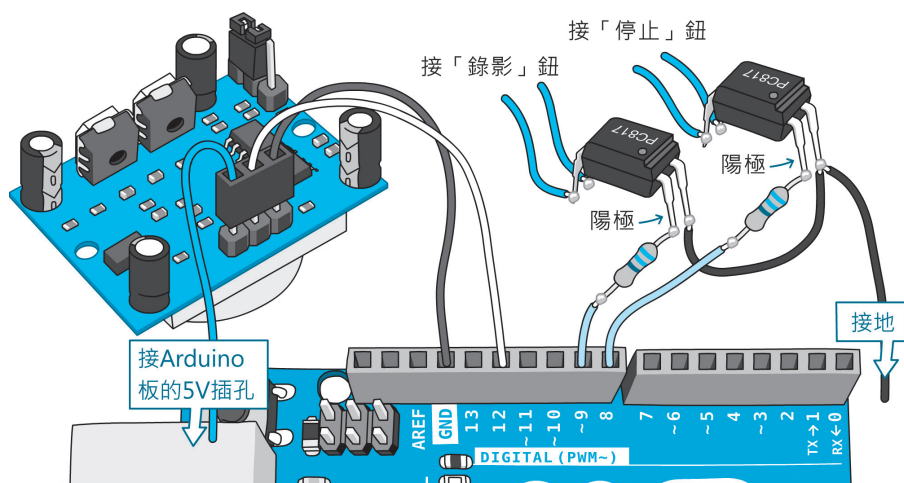
以夏普的 PC817 為例，規格書標示光電晶體的集極輸入電流（IC），最大可承受 50mA，對於一般電路裡的控制按鈕絕對夠用。不過，**光電晶體 C 和 E 腳的電流往單方向流動**，光是用眼睛看，可能無法判定光耦合的兩個接腳；如果無法確定 C, E 焊接的方向，就先隨便接，若測試後發現不能導通，再將焊接腳位對調即可：



除了光耦合元件，也可用第 18 章介紹的**繼電器（relay）**，不過，繼電器屬於機械式開關，反應速度比光耦合慢（但仍舊比人類快很多），而且繼電器消耗比較多的電量也比較容易故障，主要用於大電壓/大電流的電路。

光耦合元件的控制程式

底下是本單元的示範硬體，採用人體移動偵測模組，加上兩個光耦合元件（註：也有一個 IC 裡面包含兩組或 4 組光耦的型式），用來控制微型攝影機的「開始錄影」和「停止錄影」按鍵，讀者可自行變換，銜接其他 3C 設備的控制鈕。



光耦合元件的紅外線發射 LED 的陽極腳，要連接一個 **220Ω**（紅紅棕）的限流電阻，保護紅外線 LED。

範例程式碼如下，當 PIR 感測器偵測到人體移動時，它將點亮 Arduino 第 13 腳的 LED，並啟動「錄影」；過了 10 分鐘之後停止錄影。首先宣告程式變數，儲存**時間毫秒值**的變數類型，通常使用 **long**（長整數），以免變數容量不足而導致程式執行錯誤：

```
const byte pirPin = 12; // 紅外線感測器訊號腳位
const byte ledPin = 13; // LED 腳位
const byte recPin = 9;  // 錄影鈕
const byte stopPin = 8; // 停止鈕

long oldTime; // 暫存當前時間
/* 10 分鐘的毫秒數：1000 × 60 × 10
   底下這一行可改寫成：
   long delayTime = 1000L * 60L * 10L; */
long delayTime = 600000;
long diffTime; // 儲存時間差
bool turnOn = false; // 代表是否點亮 LED 的變數，預設為「否」
```

接著設定接腳的輸出與輸入狀態：

```
void setup() {
    pinMode(pirPin, INPUT);    // 感測器訊號腳位設定成「輸入」
    pinMode(ledPin, OUTPUT);   // LED 腳位設定成「輸出」
    pinMode(recPin, OUTPUT);   // 「錄影」腳設定成「輸出」
    pinMode(stopPin, OUTPUT);  // 「停止」腳設定成「輸出」
}
```

主程式迴圈本體內容如下，只要感測到人體移動，程式將記下啟動和點亮 LED 的那一刻，並不停地判斷自從點亮 LED 後，是否已經過 10 分鐘：

```
void loop() {  
    // 讀取感測器值，類型為布林 (0 或 1)  
    bool val = digitalRead(pirPin);  
  
    if (turnOn == false && val == true) {  
        // LED 尚未點亮且感測值為 1...  
        turnOn = true;           // 設定為「已點亮」  
        oldTime = millis();      // 暫存當前時間的毫秒值  
        digitalWrite(ledPin, HIGH); // 點亮 LED  
    }  
  
    if (turnOn) { // 如果 LED 目前是點亮的...  
        // 比較現在時間與之前記錄的時間  
        diffTime = millis() - oldTime;  
        if (diffTime >= delayTime) {  
            // 如果時間差大於或等於延遲時間 (10 分鐘)...  
            turnOn = false; // 設定為「關閉 LED」  
        }  
    } else { // 若設定為「關閉 LED」  
        digitalWrite(ledPin, LOW); // 關閉 LED  
    }  
}
```