

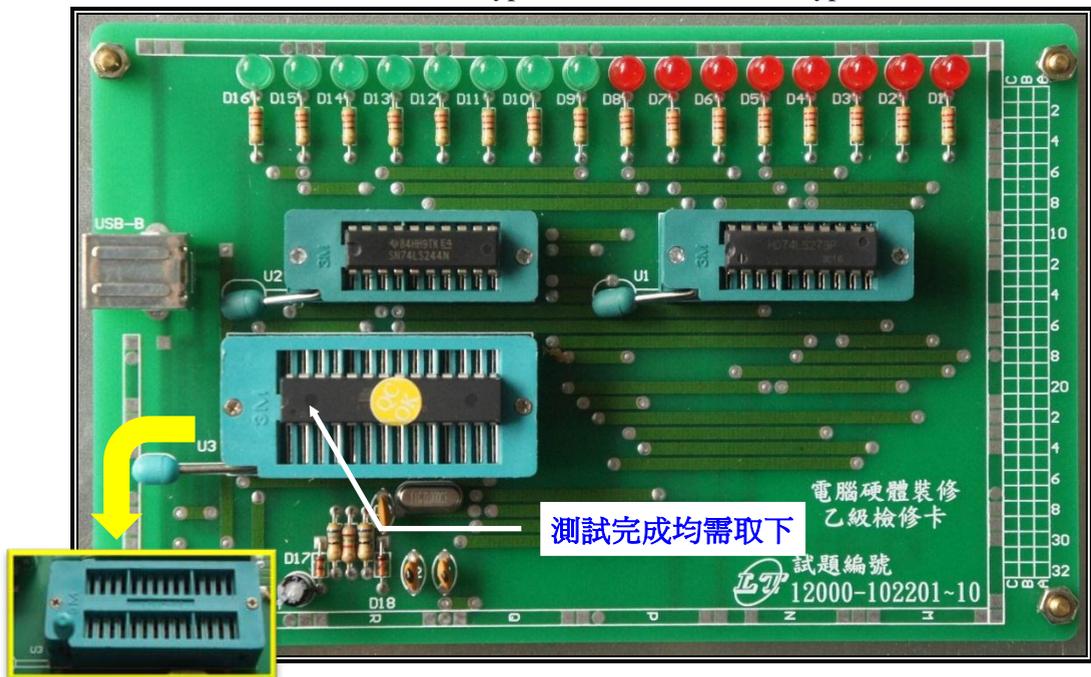
第2章

個人電腦介面卡製作

- 2-1 個人電腦介面卡製作流程
- 2-2 USB（通用串列匯流排）基本介紹
- 2-3 ATMEGA8-16PU 晶片基本介紹
- 2-4 IC 74LS244 與 74LS273 零件基本介紹
- 2-5 個人電腦介面卡電路圖介紹
- 2-6 Visual Basic 6.0 程式介紹
- 2-7 焊接技巧補充說明與注意事項
- 2-8 各式故障狀況排除問與答（Q&A）

2-1 個人電腦介面卡製作流程

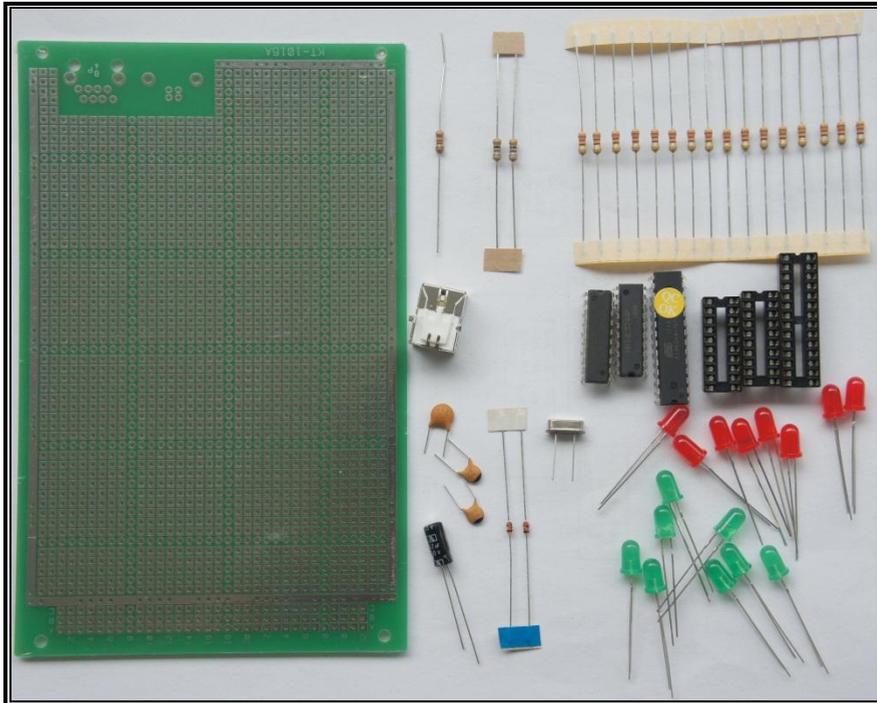
- 一、第一站試場每個工作崗位考前均會提供**個人電腦介面卡成品**，如下圖 2-1 所示，目的是讓應檢人檢查個人電腦 USB 設備以及三顆 IC 是否正常(※試場會另外提供測試程式)，若有任何 IC 或設備問題須立即提出，確認均無誤之後，試場會再將個人電腦介面卡成品收回，**內含韌體 ATMEGA8-16PU 取出留下**。介面卡成品均符合本試題提供之「個人電腦介面卡參考電路圖」、「個人電腦介面卡零件配置參考圖」位置製作，其中 ATMEGA8-16PU (內含韌體)、74LS244(左側)及 74LS273(右側)等 IC 之腳座需使用 **ZIF (Zero-Insertion Force)** 腳座，以方便將**測試 OK 的三顆 IC 留下使用**。每個工作崗位亦提供一條 USB 電纜線作為測試之用，一端為 Type A 接頭，另一端為 Type B 接頭。



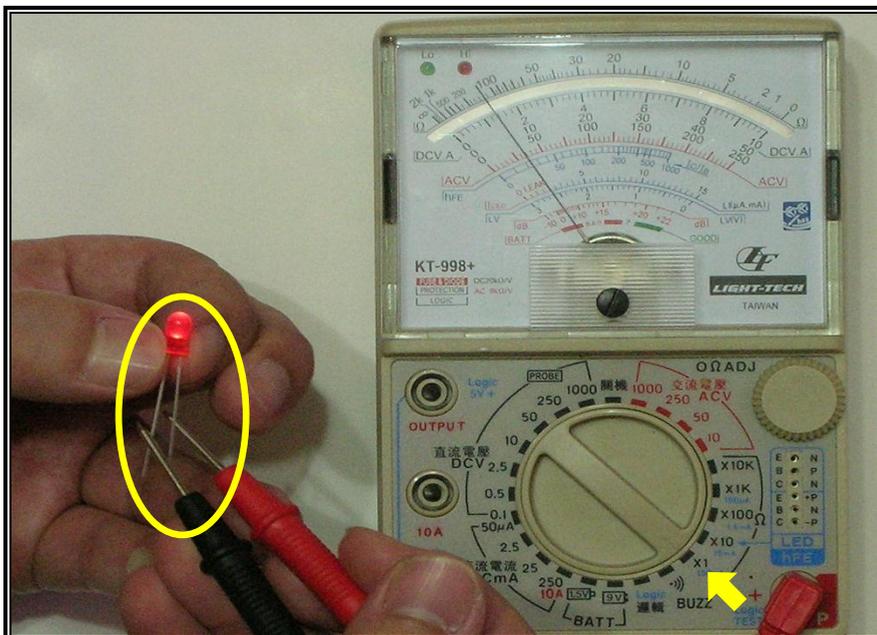
▶ 圖 2-1 試場提供的個人電腦介面卡成品 (※ 測試專用，每人均有)

- 二、第一站考試時間 150 分鐘，內含 20 分鐘檢查材料時間，依照試場所提供的第一站檢定材料表檢查是否缺件，若有缺損或故障時，得予更換，其餘時間更換器材，依評審表項目扣「10 分」，考試材料請參閱圖 2-2 所示(還有焊錫與 OK 單心線、4 根銅柱)。材料當中尤其是 16 顆的發光二極體 LED (極性為：長正/短負)務必事先量測好壞，否則等待會拆掉壞的 LED 再重焊的時間，反而會花更久時間喔！測量方式是利用三用電表的 X1 或 X10 歐姆檔，將三用電表帶「正電」的黑棒接 LED 長腳，帶「負電」的紅棒接 LED 短腳，則 LED 應點亮表示正常，如下圖 2-3 所示。

註：若有需要購買檢定相關材料或介面卡成品，請洽 LIGHT-TECH 堃喬股份有限公司，網址 <http://www.ltc.com.tw>，連絡電話：2999-2993。

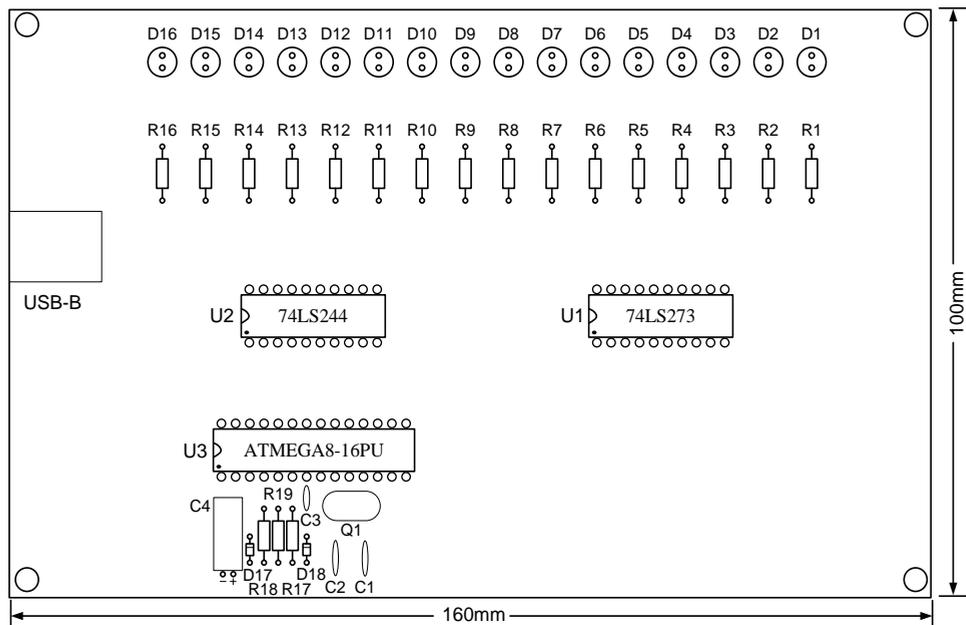


▶ 圖 2-2 第一站試場應提供的檢定材料（※ 另含焊錫、OK 單心線、銅柱）



▶ 圖 2-3 檢測發光二極體 LED 好壞方式

- 三、利用試題所提供的個人電腦介面卡『零件配置圖』，如圖 2-4 所示，將零件分別擺設到適當位置，電阻的誤差**統一朝下方**。所有零件安裝順序建議依序是：
1. 電阻 R1~R16 (※ 中間均間隔 2 格，並盡量靠上方，和 LED 也間隔 2 格)
 2. IC 座*3 (※ 缺口均朝左側 USB 連接頭方向)
 3. 電阻 R17 與 R18 (※ R17、R18 分別對齊 ATMEGA8 的 pin6、pin4)
 4. 電阻 R19 (※ 對齊 ATMEGA8 的 pin5，電阻值和 R17、R19 不同，勿裝錯)
 5. D17~D18 (※ 稽納二極體有標記的負端均朝上，並置於 R17 與 R18 兩側)
 6. Q1 (※ 石英晶體左邊腳位對齊 ATMEGA8 的 pin9)
 7. C1~C2 (※ 標記 20 表示 20pF，分別對齊石英晶體 Q1 的兩隻腳位)
 8. C3 (※ 標記 104 表示 0.1 μ f，對齊 ATMEGA8 的 pin7)
 9. C4 (※ 電解質電容負端朝左側，對齊 ATMEGA8 的 pin1、pin2)
 10. USB 連接頭 (※ 插入口要朝外側端)
 11. LED1~16 (※ 負端均朝上，右側 8 顆是紅色 D1~D8，左側 8 顆是綠色 D9~D16)
- ※ 其中 LED、稽納二極體、電解質電容器 C4，因為有極性區分，所以方向應特別留意，例如：所有 LED 的短腳應朝上（共陰），且 LED 顏色不可弄錯，最多可扣 60 分，那就不及格。當裝上 IC 時，記得左邊是 74LS244，控制綠色 LED，許多考生常因為緊張而弄相反。



▶ 圖 2-4 個人電腦介面卡零件配置圖 (※ 試場均有提供)

- 四、參閱試場所提供的『個人電腦介面卡參考電路圖』，如圖 2-5 所示，再利用材料包內的 OK 單心線連接所有電路，各 IC 的第一隻腳要特別留意，在焊接之前務必先參閱第 2-7 節焊接技巧補充說明與注意事項。介面卡實際完成圖照片（僅供參考），如圖 2-6、2-7 連接方式焊接。

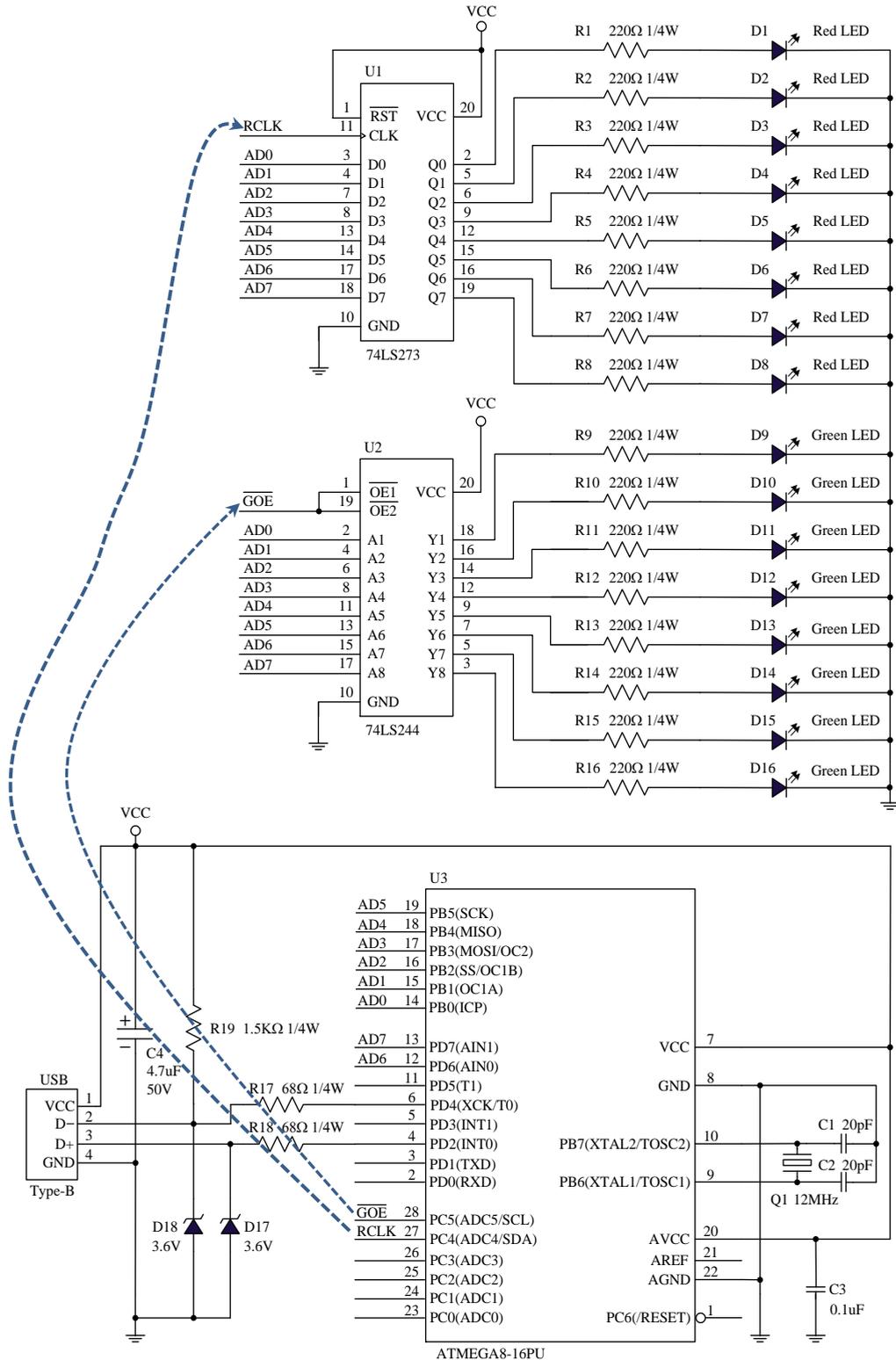
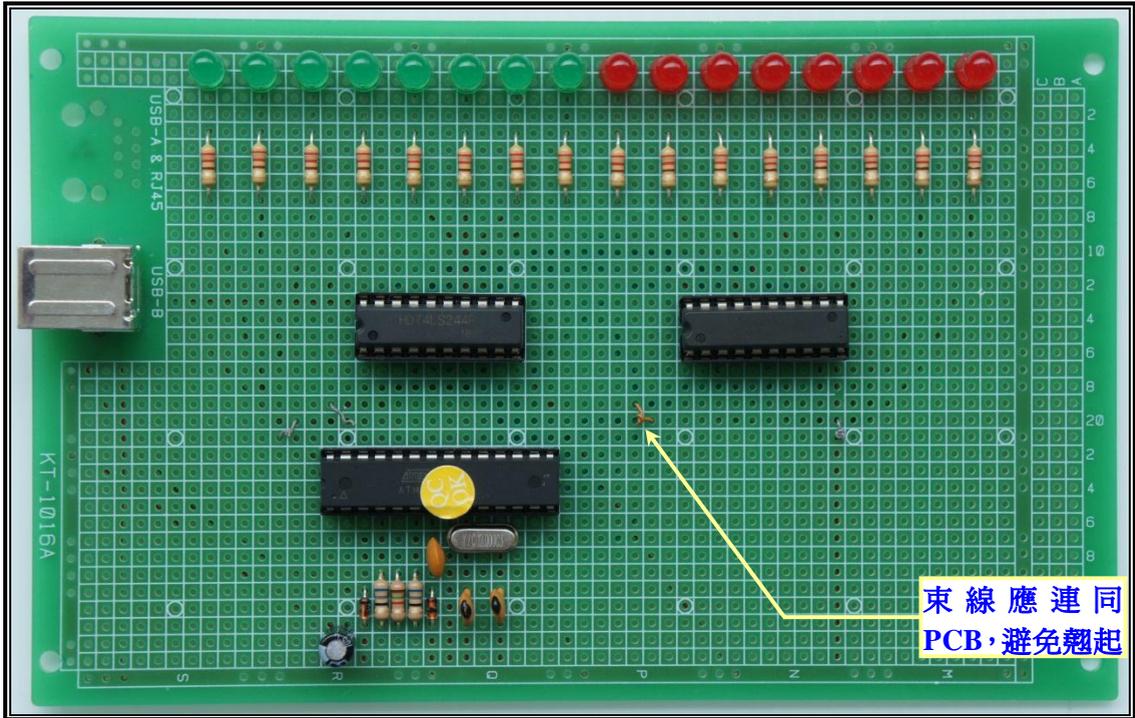
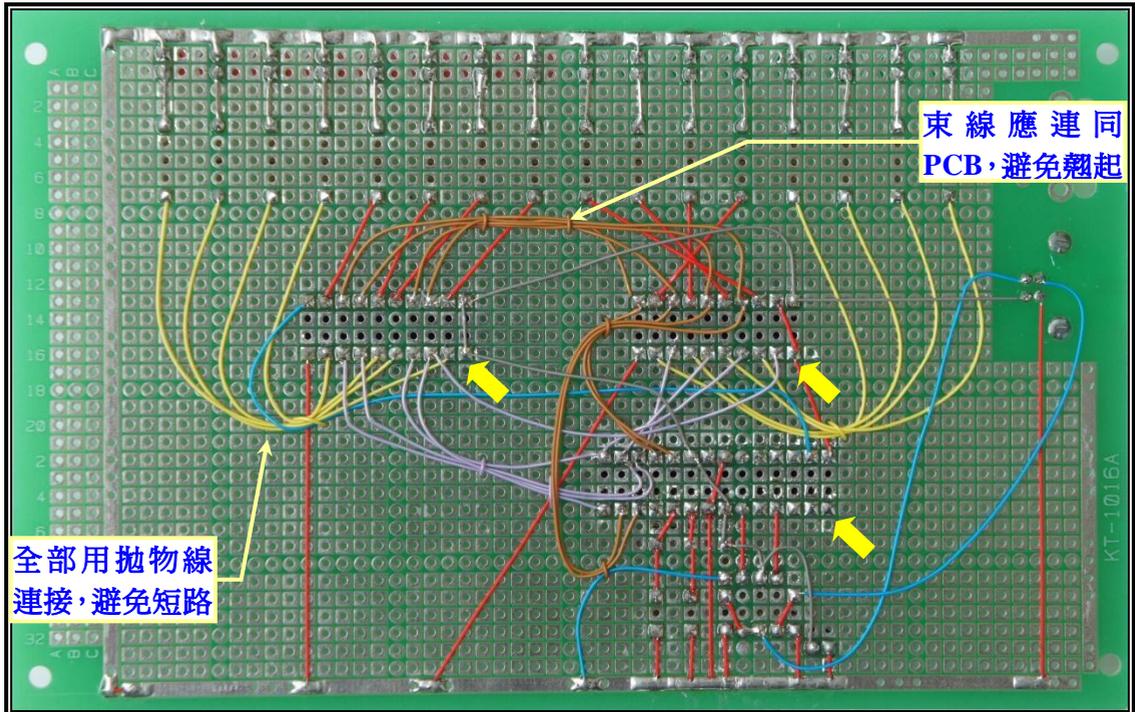


圖 2-5 個人電腦介面卡參考電路圖 (※ 試場均有提供，應會自行分析電路)



▶ 圖 2-6 個人電腦介面卡-元件正面 (※請依照試題提供位置擺設，大約位置即可)



▶ 圖 2-7 個人電腦介面卡-焊接背面 (※最建議的焊接，可避免短路；注意第一隻腳位置)



原始電路不需要再進行修改，且以功能為優先，評分雖然不看外觀，但是若外觀表現太差，則故障的機會也會跟著增加，並且不容易找出錯誤的所在。

五、由於試題規定考生的介面卡在通電檢驗時，若發生短路現象是以『不及格』方式處理，因此在測試之前，利用三用電表 **X1K Ω** 檔測量 IC 的 Vcc 與 GND 之間是否發生『**短路 0 Ω** 』，如圖 2-8 與 2-9 擇一測試，只要**超過 1.5K Ω (R19)**即表示沒有**短路**，所以有沒有裝上 IC 再測試均可，若沒有發生短路，即可將介面卡直接連接到電腦 USB 測試（※先利用試場提供程式測試硬體好壞）。

註 由於 Vcc 與 GND 之間存在 R19 的 **1.5K Ω** 串聯稽納二極體 **D18**，所以**不可能完全開路**。另外將三用電表帶『正電』黑棒接 Vcc，帶『負電』紅棒接 GND，因為稽納二極體 **D18 逆偏**，阻抗**較大**，約 7K Ω ；若將帶『正電』黑棒改接 GND，帶『負電』紅棒接 Vcc，則稽納二極體 **D18 順偏**，阻抗**較小**，約 3K Ω ，反正只要**確認有阻值即表示沒有短路**，同時 **USB 的 D+與 D-**也要**確認沒有短路**。

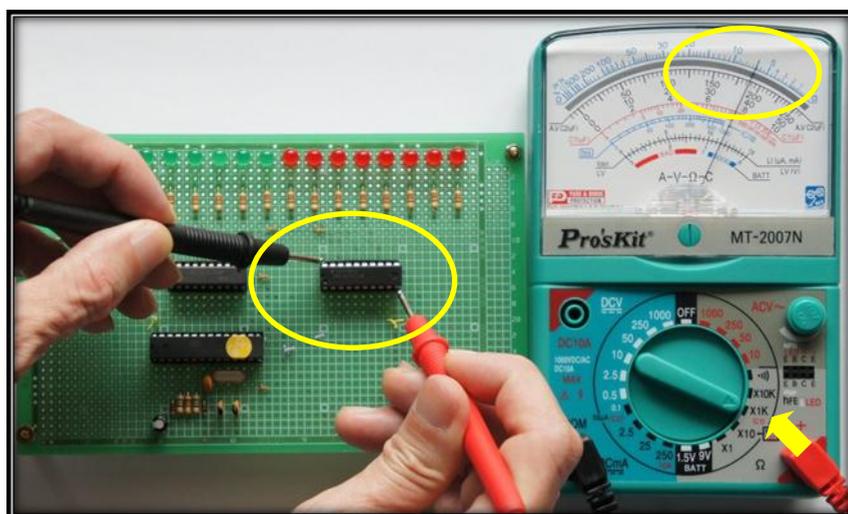


圖 2-8 測量電源之間是否沒有『**短路**』（※ D18 逆偏，阻抗較大）

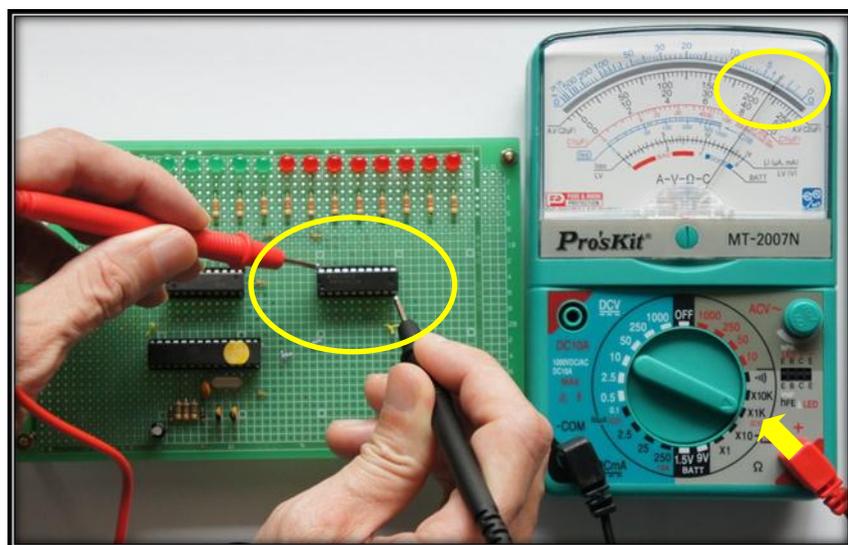


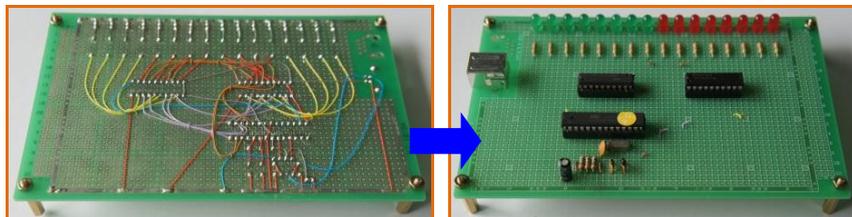
圖 2-9 測量電源之間是否沒有『**短路**』（※ D18 順偏，阻抗較小）

- 六、完成硬體焊接之後，記得先用試場提供的測試程式確認剛製作的介面卡硬體好壞？若是無法正常依序顯示，此時切勿緊張失措，應保持冷靜將問題逐一過濾檢查，而經常遭遇到的問題請自行參閱第 2-8 節 Q&A 說明，在平時練習過程中，不論你是否曾遭遇問題，都務必了解可能發生的問題，方有能力解決，以避免考試當天不知所措！當硬體功能均正常，再開始 VB 程式撰寫。
- 七、撰寫方式包括 Visual Basic 或 C 語言程式（任選一種，本書範例是以 Visual Basic6.0 語言為主），試場在寄發通知單另外還會提供 USB 函數庫或動態鏈結程式庫（DLL，Dynamic Link Library）以及呼叫的範例提供應檢人知道，但實際考試的 DLL 呼叫程式碼需自行輸入，考場並不會再提供。
- 八、開始進行 Visual Basic 程式的測試，因為剛才硬體已經確認正常，所以若是無法正常依序顯示，則一定是 VB 程式某地方輸入錯誤(含大小寫弄錯)，或是 USBIO.DLL 檔案放置在錯誤位置，或是還沒做『儲存→關閉→重新開啟』，此時切勿緊張失措，應保持冷靜將程式逐一再過濾檢查！
- 九、當自行測試所有的功能都正常，方可舉手請監評老師進行評分（請先將桌面稍作整理）。



1. 每間術科承辦單位所提供的 USB 函數庫或動態鏈結程式庫 DLL、呼叫範例程式等相關資料並不一定相同，但考生會於考前二週之前收到相關資料（光碟或提供術科測試辦理單位之網址自行下載），請考生要特別留意，若有任何疑義需事先打電話到術科承辦的單位進行查詢確認；本書光碟內附試場比較經常使用的『USBIO.DLL』檔，提供讀者練習。
2. 試場可能是提供 Visual Basic 6 或 Visual Basic 2005 程式語言軟體，兩者操作介面並不太相同，所以在收到術科試場寄來的相關資料時應詳閱確認，若想更早知道，可在公布學科成績時，網站應也會同時告知術科考試地點，屆時打電話或上術科單位的網站查詢比較快速，方便提早準備。※ 本書不再提供 Visual Basic 2005 解題，因為 Visual Basic 2005 沒有控制項陣列可使用，程式碼將變得冗長與複雜，不建議使用。
3. 實際考試個人電腦介面卡僅以功能為優先考慮，並不看整體的外觀，透過平時勤加練習，除了講求速度之外，每個焊點均應要求確實，也不要接錯的狀況發生，儘可能第一次就要成功，否則考生進行偵錯(Debug)的時間，反而會比重做的時間還要更長。
4. D1~D16 LED 擺設的位置並非固定不變，因為各家萬用版的設計稍有不同，所以測驗當天須稍作調整，以避免發生短路或 LED 沒接到 GND。
5. D1~D16 LED 顏色與排列順序不可以弄錯，否則扣 60 分而不及格。
6. 74LS244(左側)與 74LS273(右側)、ATMEGA8 方向不可相反。
7. OK 單心線若不敷使用或不好剝皮時，應該舉手詢問監評老師是否可使用自己攜帶的線材，以防止被當成作弊而論處。

8. 若是學校集體輔導，建議每次焊接練習完畢，**僅需將 OK 線全數拆除**，其他元件繼續保留原位，反覆練習 OK 單心線材的焊接技術，且應要求在 **50 分鐘之內完成 67 條接線較佳**。
9. 在平時練習時，**若遭遇部分功能失敗，強烈建議應找出問題，藉此提升檢修能力**，因為作者輔導學生多年，老師最擔心平時都做一次就成功的人因太自信，反而到了實際考試，當遭遇問題因無經驗，反而**自亂陣腳**！
10. 若是考試當天真的一直找不出問題，則是**建議拆掉 OK 單心線重做**，應該都還來得及；由於一塊萬用電路板至少可以反覆練習十幾次，讀者只要反覆多練習幾遍，相信第一站焊接部分一定沒有問題。
11. **為避免在焊接過程左右搖晃**，務必裝上考場提供的銅柱，並裝在元件面，如左下圖所示；一旦焊接完畢強烈建議將四根銅柱**反接**，如右下圖所示，然後才開始進行量測評分，這樣可**避免因為不小心壓到 OK 單心線，而臨時造成短路現象發生**。因此考試當天若時間不夠，可不用束線，功能為優先考量！



12. 第一次使用安裝會比較慢是因為需安裝驅動程式，拔掉再插入就會很快使用，但若 USB 不確定是否有被正確偵測，可利用觀察**裝置管理員**或**右下角圖示**的變化，人性化介面裝置新增的『**HID-compliant device**』即是介面卡。每台裝置設備並不相同，且顯示的種類順序也不一定相同，所以應先開啟裝置管理員的『人性化介面裝置』，確認 HID-compliant device 目前個數，當插上 USB 裝置重新觀察 HID 個數是否有增加。PS1:HID(Human Interface Device 人機介面裝置)參閱 P2-11 說明。PS2:compliant 應允的、符合規定/規格的。



圖 2-10 如何判斷是否正確偵測到 USB 介面卡裝置

2-2 USB (通用串列匯流排) 基本介紹

在早期電腦外接式裝置的傳輸介面都不相同，例如：印表機接 LPT 並列埠、數據機接 RS-232 串列埠、滑鼠與鍵盤接 PS/2 埠等，而且還必須要再安裝驅動程式，然後重新開機才能使用的不便性，因此，創造出一個『統一』介面標準，便成為無可避免的趨勢。

USB 是 Universal Serial Bus 的縮寫，中文是『通用串列(序列)匯流排』，最早是由英代爾 Intel、微軟 Microsoft 提出，後來又有許多電腦與電子科技大廠陸續加入研發，用來連接電腦系統和外部裝置的一種『串列』埠匯流排標準。

USB 利用一條傳輸線上來串接各類型的周邊設備，例如可以取代印表機之 LPT 並列埠、RS-232 數據機之串列埠、滑鼠與鍵盤接 PS/2 埠等，且資料傳輸速度都比原來更快。USB 允許把外面的設備與電腦連接時，不必重新配置規劃系統，也不必打開機殼。而當連接上電腦時，電腦會自動識別這些周邊設備，並且給予適當的驅動程式，無需再由使用者另外重新設定。底下簡述 USB 重要的特性與優點：

1. USB 界面統一了各種周邊設備的連接頭，包含通訊界面、印表機界面、音效裝置、儲存設備等，都可採用相同的 USB 界面規格，所以 USB 界面就像是『萬用接頭』，只要將插頭插入即可使用。
2. 具有『隨插即用』(Plug-and-Play)的特性，能自動偵測與配置系統的資源。
3. 具有『熱插拔』(Hot plugging)的特性，在作業系統已開機的狀態中，隨時可以插入或拔除 USB 裝置，而不需再另外關閉電源。
4. USB 最多可以連接 127 個周邊設備同時運作。
5. USB 1.1 的周邊設備應用兩種不同的速度：12Mbps(全速)和 1.5Mbps(慢速)。其中，慢速主要是應用於人機界面 HID (Human Interface Device)上。這是一種用於連接滑鼠、鍵盤等設備的 USB，2014 年開始的新版電腦硬體裝修乙級試題就是採用 1.5Mbps(慢速)標準，所以當插入介面卡時，裝置管理員會新增『HID-compliant device』裝置。至於傳輸速度更快的 USB2.0、USB3.0 因為與本試題無關，所以不多作介紹。

USB 接頭本身提供一組 4.4~5.25 伏特的電壓，電流供應約 100mA~500mA，可作為連接 USB 設備的電源，訊號腳位除了 V_{BUS} (紅色)與 GND(黑色)之外，還包括 D+(綠色)和 D-(白色)串列資料傳輸，兩條資料線的訊號是相反的，當其中一條的訊號為 High，則另一條的訊號就為 Low，因為該差動訊號能提升訊號抗雜訊抗干擾的能力，來達成高速傳輸的目的。而資料編碼方式並不是採高電位為 1、低電位為 0 的方式，而是採用 NRZI (Non return to zero, inverted 翻轉不歸零制)，NRZI 的發送訊號有兩種準位，並且在準位訊號內也已經夾帶了同步時脈訊號於內，當電位『保持(不變)』時傳送

邏輯 1，電位『**翻轉(改變)**』時傳送**邏輯 0**，如圖 2-11 所示，主要目的是要**抵消長導線**的電磁干擾，以便可以提高傳輸速度。

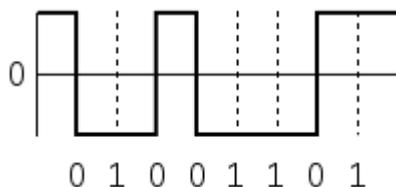


圖 2-11 NRZI 編碼方式

常見的 USB 接頭主要分成 TypeA 與 TypeB 型，兩者均具有**防呆**設計，避免插錯裝置，試場提供每位考生一條 USB 電纜線，一邊是採 TypeA 型，用來連接電腦 USB 埠，而另一端採 TypeB 型，以連接自行製作的個人電腦介面卡，關於 USB 的 TypeB 型腳位定義與外觀(※**務必熟記 4 支腳位定義**)、試場提供的 USB 傳輸線如圖 2-12 所示。

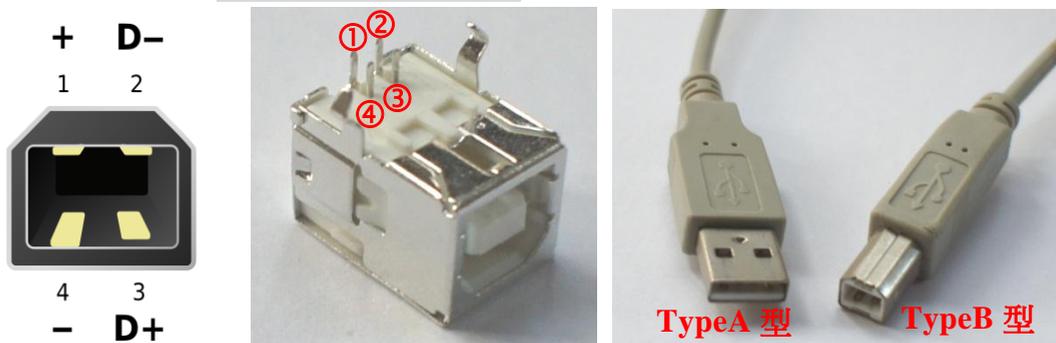


圖 2-12 焊接在印刷電路板上的 TypeB 型 USB 插座腳位定義

至於 USB 軟、硬體架構、匯流排階層架構、封包傳輸的通信協定等方式，因為已經超出高職生程度以上，所以試題並未列入，在此不多加介紹，若有興趣不妨參閱全華許永和編著『USB2.0 高速週邊裝置設計之實務應用』有更詳細說明。欲透過 USB 介面進行串列傳輸控制，需要進行底下三個步驟：

1. 撰寫 HID (Human Interface Device 人機介面裝置) 韌體的部分，包括定義資料傳輸方式，例如考場須先將 main.hex 燒錄到 ATMEGA8-16PU 裡面，這樣當我們插入 USB 界面裝置才可以被電腦偵測到，因此，若測試過程一直出現『**USB 裝置無法辨識**』錯誤訊息，發生原因就是 **ATMEGA8-16PU 相關線路沒有完全接好**或者是 main.hex 燒錄過程可能有錯誤所造成，但第二種錯誤不可能，因為考前均先測試過 IC，所以原因都是個人在焊接 ATMEGA8-16PU 線路過程，發生短路或空焊造成。

註 1：若有興趣自行燒錄 main.hex 檔案，可自行到中部辦公室網站，下載網址為 <http://www.labor.gov.tw/home.jsp?pageno=201110210028>，找到『12000 - 電腦硬體裝修(乙級)』圖示下載壓縮檔，裡面還包括**試場提供的測試程式**

執行檔，如圖 2-13 所示，還有 **USBIO.DLL** 檔案以及呼叫方式的說明。



圖 2-13 中部辦公室網站提供的測試程式執行檔範例畫面

- PC 端 HID API 函式的撰寫，如果將 API 函式先做成動態連結檔 DLL，這樣會有助於程式語言的撰寫，例如考場所提供的 **USBIO.DLL** 動態連結檔即是。如何查看 DLL 裡面內容，利用微軟所提供的 **Dependency Walker** 軟體，結果參考下圖 2-14 所示，下載該軟體網址為 <http://www.dependencywalker.com/>，下載解壓縮後，內有三個檔案，只要點選 **depends.exe** 執行即可看到。

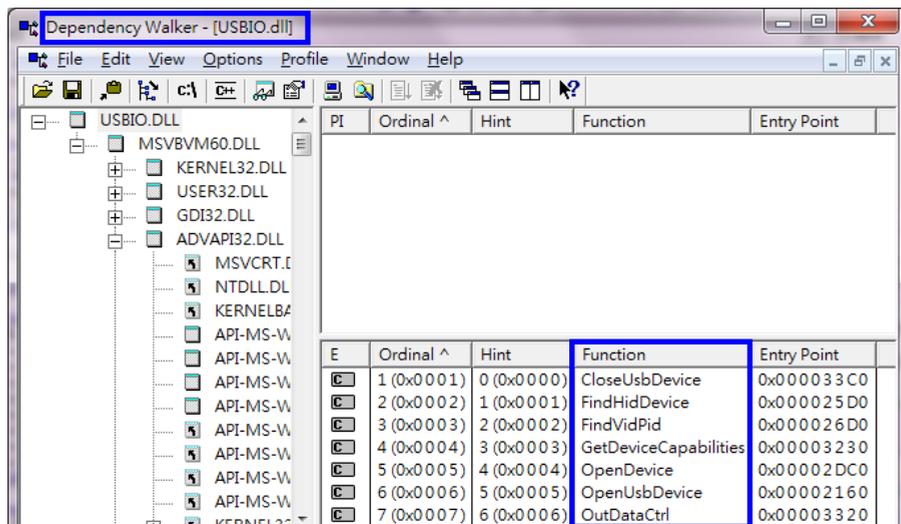


圖 2-14 查看 USBIO.DLL 裡面內容包括哪些函數

- 前兩個步驟考場都已經幫忙完成**，接著考生僅需撰寫 Visual Basic 程式去控制輸出即可，所以雖然在 2014 年將使用多年的 Printer 並列埠改為 USB 串列傳輸控制，對 Visual Basic 程式撰寫內容並無太大改變，**僅呼叫使用方式不同**。

2-3 ATMEGA8-16PU 晶片基本介紹

Atmel AVR 系列是一種基於改進的**哈佛結構** Harvard architecture，並採用精簡指令集（Reduced Instruction Set Computing，**RISC**）的微控制器，由 Atmel 公司於 1996 年研發。AVR 系列是首次採用快閃記憶體(Flash Memory)以作為數據儲存介質的單晶片微控制器之一。

其中 AVR 處理器的 ATMEGA8 系列，是採用低功耗 **8 位元** CMOS 微控制器結構，可提供許多嵌入式控制應用，且低成本的解決方案。ATMEGA8-16PU 微控制器外觀與腳位定義與內部結構圖請參閱圖 2-15 與圖 2-16，**數字 8 表示內建 8KB 的 Flash Memory** 用來撰寫程式、**數字 16 表示可以支援最高為 16MHz** 的系統時脈(註：試題採用 12MHz 石英晶體振盪)，英文**字母 P 表示為 DIP 封裝**，英文**字母 U 表示為無鉛工業等級**，另外 ATMEGA8-16PU 功能強大，底下僅列出較重要的幾項規格：

1. 內建 8K 位元組的可程式設計 Flash(具有同時讀寫的能力)，可以重複燒錄 10,000 次使用
2. 512 位元組 EEPROM
3. 1K 位元組 SRAM
4. 32 bits 通用 I/O 埠
5. 32 個 8bits 通用型工作暫存器
6. 當工作於 16 MHz 時性能高達 16 MIPS(Millions of Instructions Per Second)

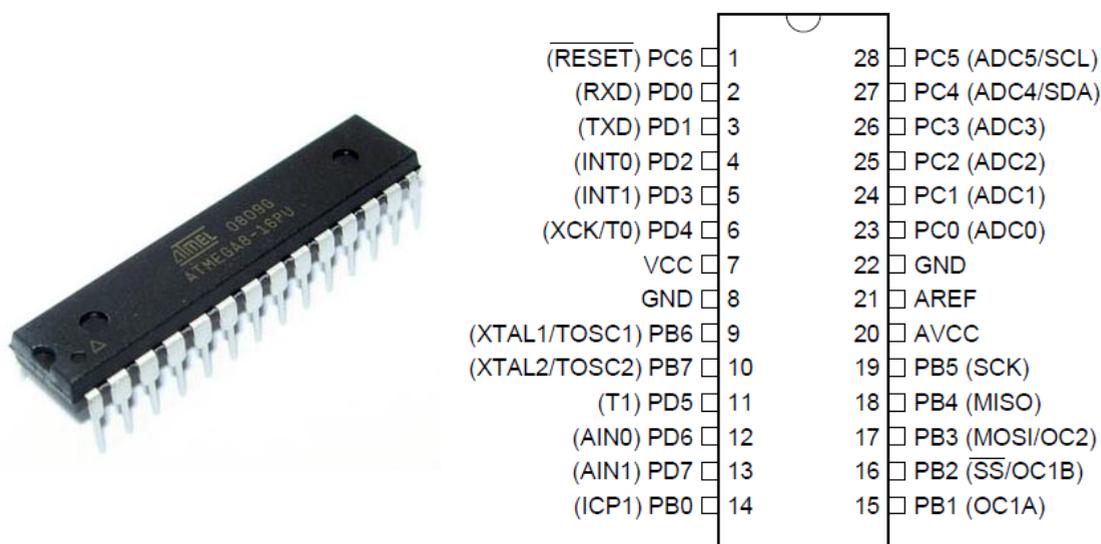


圖 2-15 ATMEGA8-16PU 微控制器外觀與 DIP 各腳位定義

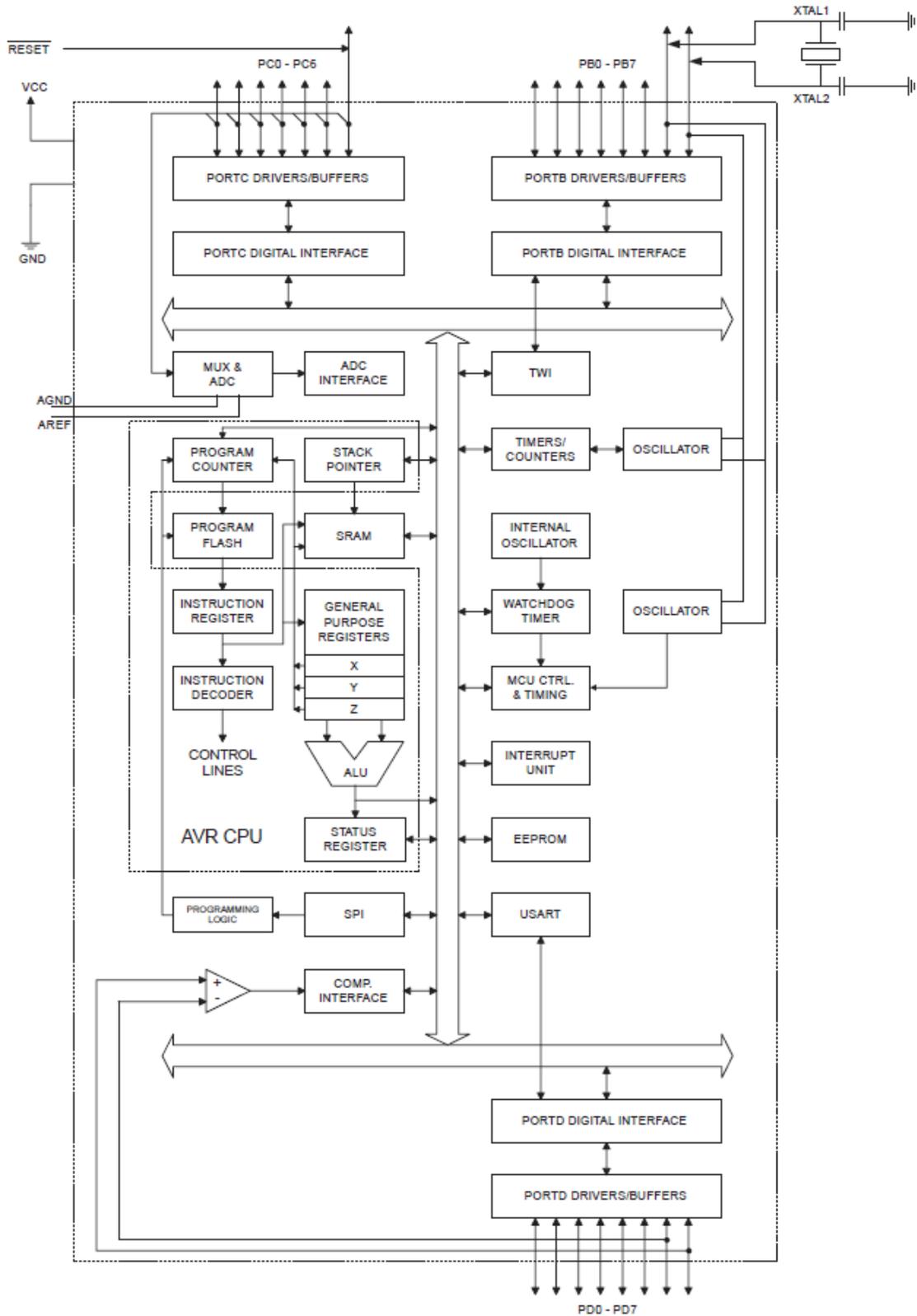


圖 2-16 ATMEGA8-16PU 晶片內部結構圖

2-4 IC 74LS244 與 74LS273 零件基本介紹

74LS244 內部包括兩組的三態閘 (Tri-State Gate)，每組又包括四個三態閘，內部構造與接腳電路如下圖 2-17 所示，第一隻腳固定在缺口的正下方，並依照『逆時鐘』的方向進行編號，且幾乎所有 74 系列 TTL IC 的 V_{cc} 都在『左上角』，而 GND 則是固定在『右下角』的地方。(※ ATMEGA8 的 V_{cc} 與 GND 是在 pin7、pin8)

所謂三態閘是指輸出除了『0』與『1』之外，還有『高阻抗』等三種狀態，當輸出呈現高阻抗的狀態時，即相當於『開路』狀態，所以連接到輸出端的 LED 就無法正常動作（熄滅）。

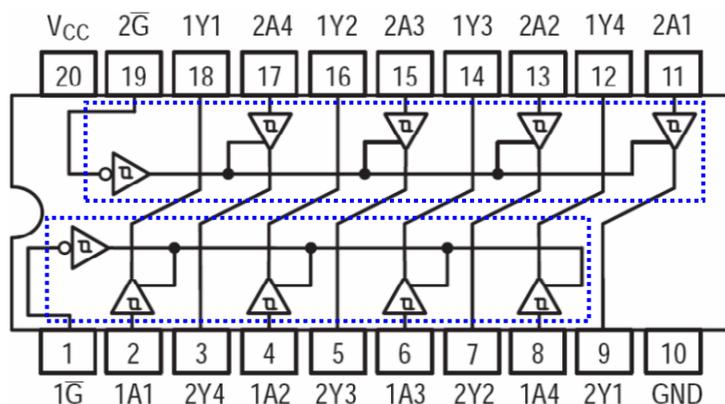


圖 2-17 IC 74LS244 三態閘內部構造與接腳圖

從上圖可觀察到 Pin1 與 Pin19 分別控制兩組三態閘的輸出，即所謂的致能 (Enable) 端，當 $1\bar{G} = 2\bar{G} = 0$ 時，則三態閘相當於緩衝器 (Buffer)，即輸出等於輸入訊號，若 $1\bar{G} = 2\bar{G} = 1$ 時，則三態閘呈現高阻抗輸出狀態，即所謂的『開路』，所以連接輸出端的 LED 就無法正常動作，即所有連接的 LED 都會熄滅，茲將三態閘的狀態表整理如下表 2-1 所示。

Input 輸入端		Output 輸出端
$1\bar{G}$ 、 $2\bar{G}$	A	Y
L L H	L H X (L 或 H)	L H 高阻抗

表 2-1 IC 74LS244 三態閘狀態表

74LS273 內部包括八組的 **D 型正反器** (Flip Flop 簡稱 FF)，正反器是指雙穩態零件，其中第 11 隻腳為時脈 (Clock Pulse, CP) 端，且屬於**正緣** (上升緣) 觸發式的正反器，而 D 是表示 Data (資料) 傳送的意義，**當 Clock 發生正緣的觸發訊號之後，輸入端 D 的訊號才會傳送到輸出端 Q**，但是若 Clock 沒有被觸發時，則會一直持續保持目前的狀態。另外 \overline{MR} 相當於清除 (Clear) 端，所以當 $\overline{MR} = 0$ 則輸出 Q 全部為 0。

74LS273 的接腳圖與內部構造圖如下圖 2-18 與 2-19 所示，而 D 型正反器的狀態表整理如下表 2-2 所示。

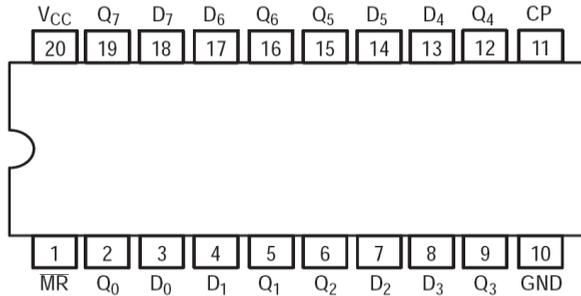


圖 2-18 IC 74LS273 D 型正反器接腳圖

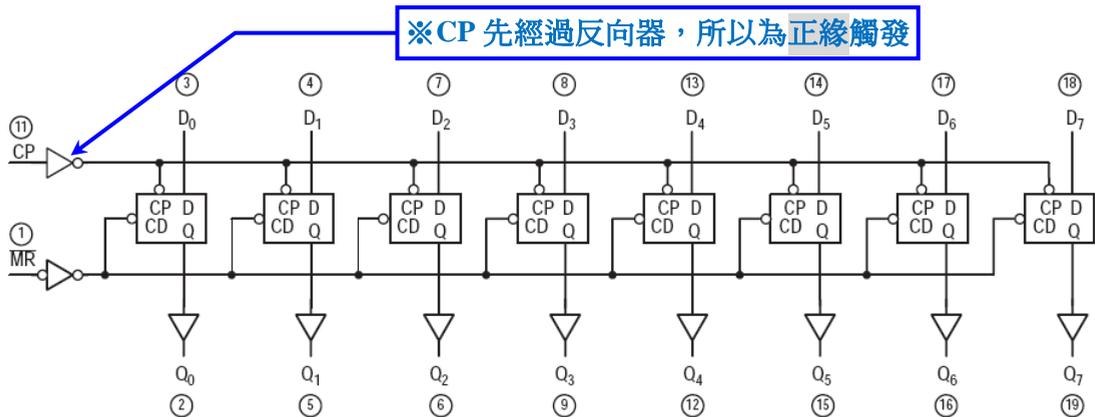


圖 2-19 IC 74LS273 D 型正反器內部構造圖

Input 輸入端			Output 輸出端
\overline{MR}	CP	D	Q
L	X (L 或 H)	X (L 或 H)	L
H	↑ (正緣)	L	L
H	↑ (正緣)	H	H

表 2-2 IC 74LS273 D 型正反器狀態表

2-5 個人電腦介面卡電路圖介紹

在 2008~2013 年所修訂的試題是利用印表機埠（又稱為並列埠 Parallel Port）控制 74LS244 與 74LS273 IC 傳送資料，使 15 顆 LED 亮滅，而 2014 年新版試題改用 USB 進行控制，由於需透過 ATMEGA8 控制 74LS244 與 74LS273 IC 傳送資料，而且**多一顆 LED16**，同時電路會因為**新增 ATMEGA8 控制電路而較為複雜**，如下圖 2-19，該電路十分重要，若接錯或焊接不確實，就無法控制 74LS244 與 74LS273 IC，甚至全部都不亮。茲將較重要的補充說明如下：

1. 在剛開始放置元件時，就應先參考試場提供的電路圖，例如 R17 應直接對齊 pin6、R18 應對齊 pin4、C3 應直接對齊 pin7、石英振盪 Q1 應對齊 pin9 等。
2. ATMEGA8 的 **pin27(PC4)**負責控制 74LS273 的『時脈』腳位，當 pin27 腳位送出 **0 到 1 的正緣觸發**，則可以開始控制**紅色的 LED1~LED8**，若僅是紅色 LED 都不亮，一定是 pin27 腳位連接或焊接有誤。
3. ATMEGA8 的 **pin28(PC5)**負責控制 74LS244 的『致能』腳位，當 pin28 腳位為**低態 0**，則可以開始控制**綠色的 LED9~LED16**，若僅是綠色 LED 都不亮，一定是 pin28 腳位連接或焊接有誤。
4. AD0~AD7 因為同時並接到 74LS244 與 74LS273 IC 的輸入端，若亮燈順序不正常，應是腳位弄錯，所以要特別小心。

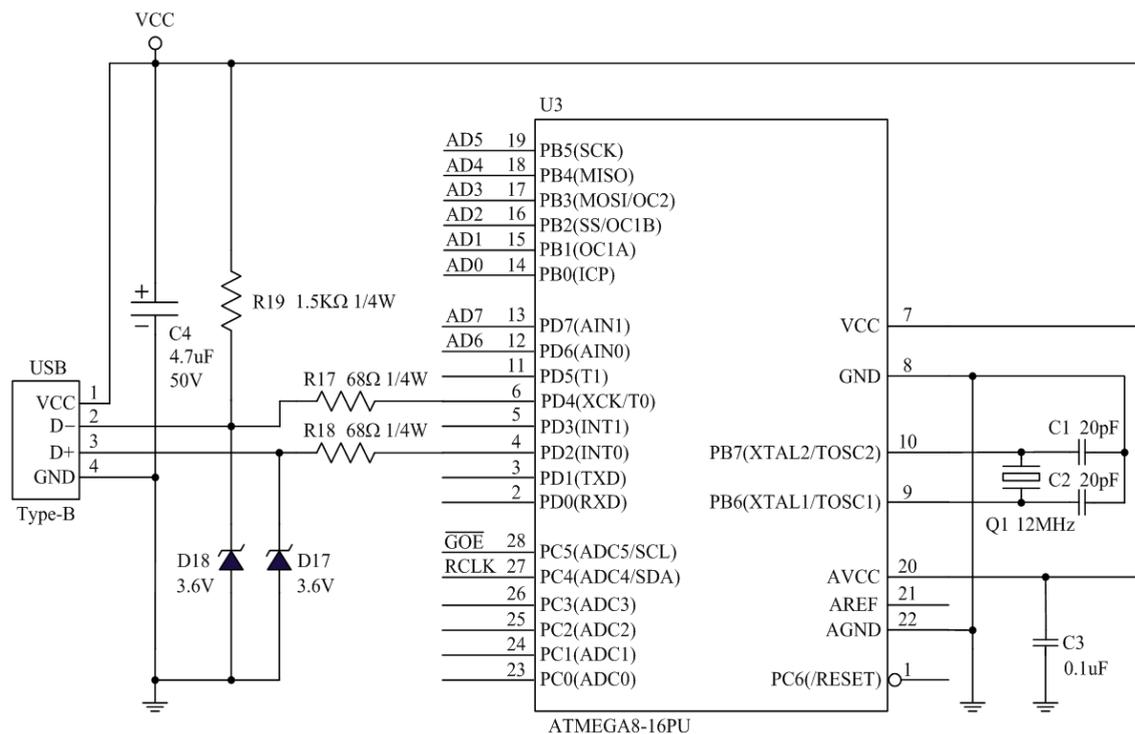


圖 2-19 個人電腦介面卡參考電路圖（※ ATMEGA8 控制電路區塊）

另外，其他的電路圖當中 **74LS273 控制紅色的 LED1~LED8 共 8 顆 LED**，而 **74LS244 則負責控制綠色的 LED9~LED16 共 8 顆 LED**(※2013 年之前舊試題僅 7 顆 LED)，若考試突然忘記，一定要習慣去看電路圖，該區塊電路相關重要說明如下：

1. 發光二極體 LED 是有分極性要特別留意，長腳為正端，**短腳為負端(共陰)**。
2. 74LS244 與 74LS273 IC 的第 10 Pin 腳接地，第 20 Pin 腳接+5V，但務必記得都要和 USB 的 Vcc 與 GND 分別連接在一起。
3. 電路圖 74LS273 的 Pin1 ($\overline{\text{RST}}$ ，Reset 重置)，即相當於 2-4 章節介紹的 $\overline{\text{MR}}$ 訊號，所以需接 Vcc，否則輸出資料永遠清除為 0，造成 LED 全不亮。
4. 74LS273 的 Pin11 (RCLK) 即相當於 2-4 章節介紹的 CP 正緣觸發訊號，該正緣觸發訊號連接到 ATMEGA8 的 **Pin27(PC4)**，若沒有正確連接，則**紅色 LED1~LED8 等 8 顆 LED** 因無正緣觸發而無法控制狀態。
5. 74LS244 的 Pin1 ($\overline{\text{OE1}}$) 與 Pin19 ($\overline{\text{OE2}}$)，即相當於 2-4 章節介紹的 $\overline{1\text{G}}$ 與 $\overline{2\text{G}}$ 訊號，該致能 (Enable) 訊號是一起連接到 ATMEGA8 的 **Pin28(PC5)**，若沒有正確連接，則**綠色 LED9~LED16 等 8 顆 LED** 因無致能而無法控制狀態。

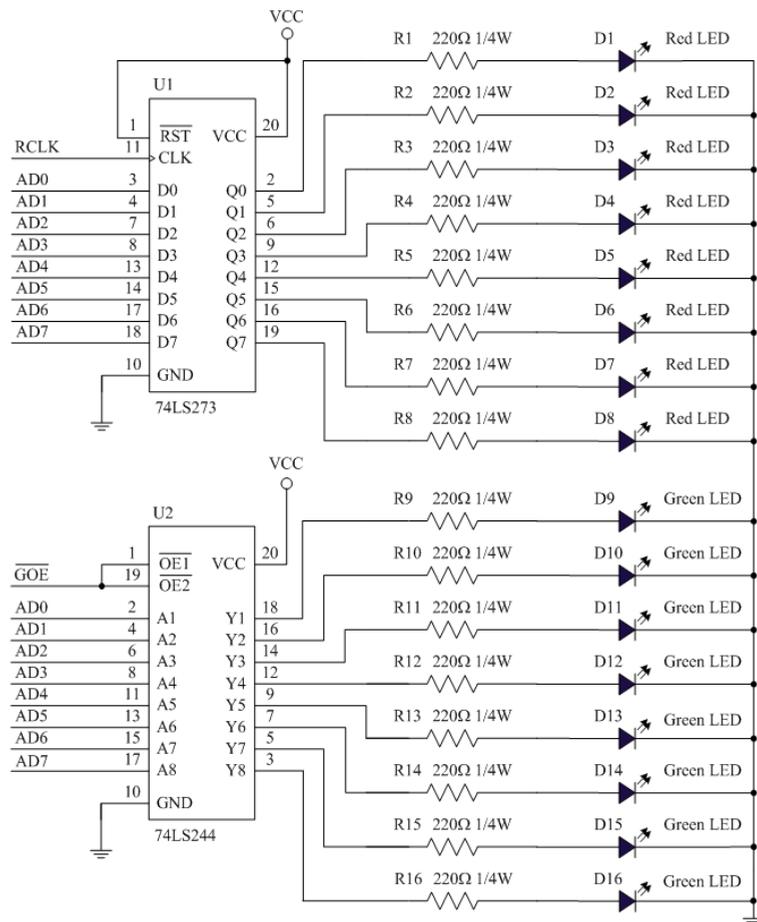


圖 2-20 個人電腦介面卡參考電路圖 (※ 試場均有提供，應會自行分析電路)

- 由於電路當中的 Vcc 節點連接數目比 GND 少，所以直接用一條 OK 單心線就將 Vcc 全部串接在一起，例如：**USB_Vcc → 74244IC Pin20 → 74273IC Pin20 → 74273IC Pin1 → ATMEGA8-16PU Pin20 → ATMEGA8-16PU Pin7 → C3 → R19 → C4+**，請自行參閱第 2-7 節，焊接補充說明。
- 由於 16 顆 LED 的負端與 USB 的 GND，還有許多元件的 GND 都要連接在一起，所以強烈建議**善用萬用板周圍均當作 GND 焊點**，但每間術科試場所提供的萬用電路板並不一定相同，**四周圍不一定都接通**，需自行串接使用，如圖 2-21，考生應特別留意，否則會**因為沒有共同參考電位而無法正常動作**。

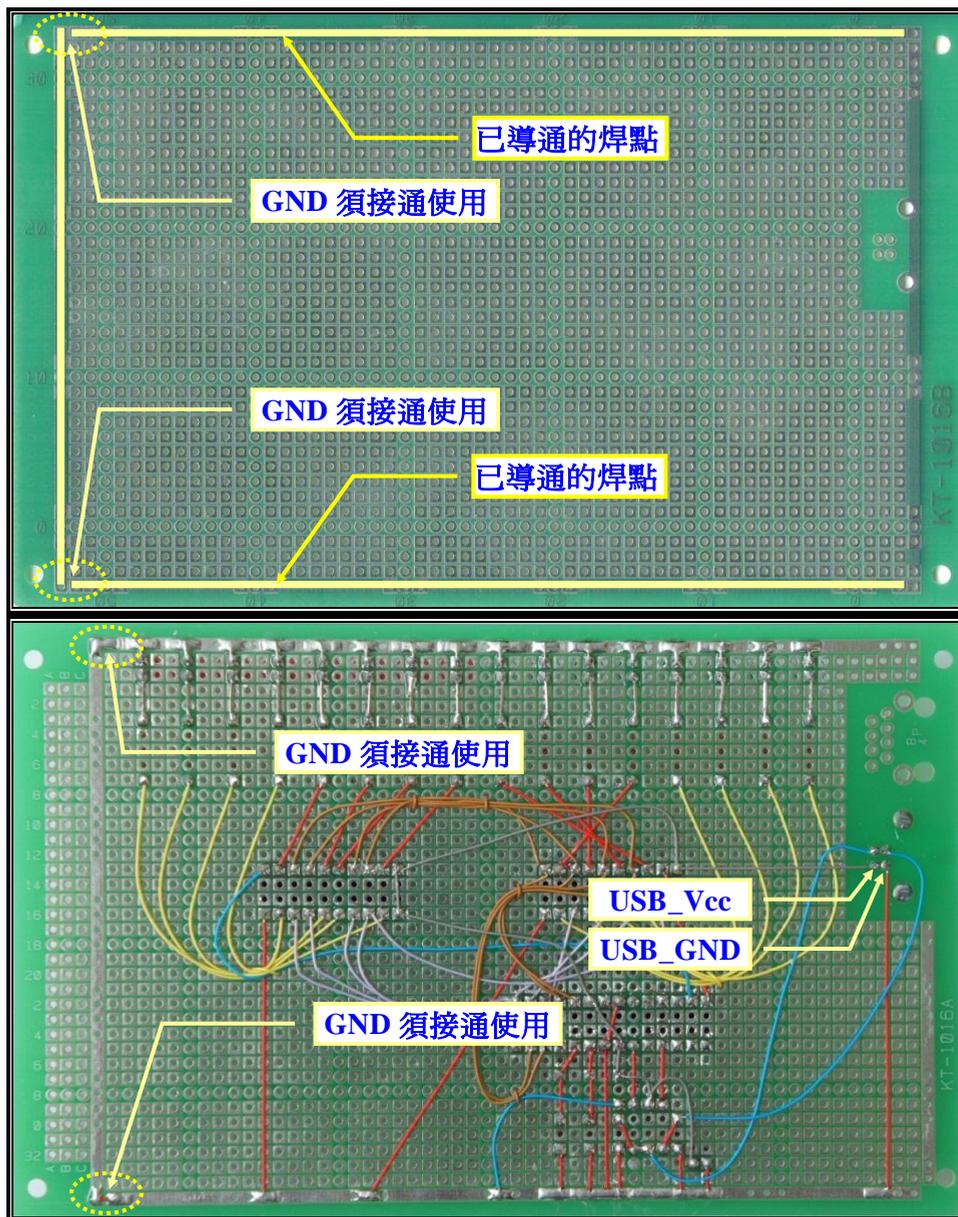


圖 2-21 個人電腦介面卡萬用板（※ 各試場提供的版本並未統一）

2-6 Visual Basic 6.0 程式介紹

接下來我們要設計Visual Basic 6.0程式，用以控制16顆LED燈號不同亮燈的順序，由十題當中抽一題進行測驗，雖然有十道題目，但程式碼幾乎完全相同，僅有綠色LED9~16顯示狀態的宣告資料不同而已，若考生對Visual Basic 6.0操作環境不熟悉，請自行參閱附錄C的內容說明與光碟全程錄製的多媒體教學影片。

註：本書不再提供Visual Basic 2005 版本，因為無控制項陣列，程式將更複雜與冗長。

當電腦執行程式時，螢幕顯示畫面與物件名稱如圖2-22所示，另需使用Timer1定時器物件，但它在執行時畫面不會顯示，文字方塊Text1負責顯示系統現在時間，如圖2-23所示（註：需點擊兩下系統視窗右下角的時間，將時間調整到現在的正確時間才行！），分別代表『時：分：秒』，時間格式不限，畫面字型字體、按鈕初始狀態與顏色由應檢人自行決定，但物件相對位置不可改變，而表單與各按鈕的標題需『完全』依照試題要求輸入，否則會被扣分。



圖 2-22 電腦執行時的顯示畫面



圖 2-23 調整電腦系統的日期與時間的畫面

第一站個人電腦介面卡實際LED尚需配合表單，對應顯示目前介面卡的LED亮滅狀態，這是2014年試題新增功能要求，電腦螢幕詳細動作要求說明如下：

- 當尚未完成介面卡與電腦USB連接，則電腦螢幕LED1~LED16均以『中空』顯示。



- 當介面卡與電腦USB連接成功，則電腦螢幕LED1~LED16均『填滿對應顏色』顯示。



- 當按下GreenLED按鈕，則電腦螢幕對應綠色LED9~LED16『同步點亮』顯示。



- 當按下RedLED按鈕，則電腦螢幕對應紅色的LED1~LED8『同步點亮』顯示。



本試題共需使用**22**個物件，含表單Form1，**表單標題Caption**屬性須完全依照試題的範例輸入，茲將上述其它各物件的屬性整理如表2-3所示。

Text1 文字方塊	Command1(1) 按鈕物件	Command1(2) 按鈕物件	Command1(3) 按鈕物件	G(7)~G(0) R(7)~R(0)	Timer1 計時器物件
Alignment=2 置中對齊	Caption = Green LED	Caption = Red LED	Caption = EXIT	Shape=3 圓形	Interval=1000 (計時基本單位為千分之一秒)
Text 內容為空白	Index=1 (自動產生)	Index=2 (自動產生)	Index=3 (自動產生)	Fillstyle=0 填滿	
Font (字型可省略)	Font (可省略)	Font (可省略)	Font (可省略)	1.填入的顏色利用程式設定。 2.配合 LED 顯示，需將左邊的 Index=7，右邊的 Index=0	
	Style=1 圖片外觀 (可省略，但背景顏色才會改變)	Style=1 圖片外觀 (可省略)	Style=1 圖片外觀 (可省略)		註：平時不要因為方便檢查，而自行調快顯示速度，實際考試若不是間隔一秒，可是會被扣分。
	BackColor 背景顏色 (可省略)	BackColor 背景顏色 (可省略)	BackColor 背景顏色 (可省略)		

表 2-3 各物件屬性總覽 (※ 綠色與紅色 LED 的按鈕不要放錯位置)



- 為了加快測驗時間，以及簡化程式內容，建議先設定好第一個 **Command1** 屬性後，再利用 **Ctrl+C** 複製物件，然後再按 **Ctrl+V** 貼上物件，此時會出現如圖 2-24 的詢問視窗，務必選擇 **是(Y)** 按鈕，利用控制項陣列會比較方便，此時原物件名稱已變為 **Command1(0)**與 **Command1(1)**，其 **Index** 屬性自動設為 0 與 1，當再按 **Ctrl+V** 貼上第三個按鈕物件，此時將不再詢問，直接會設定為 **Command1(2)**名稱。最後將 **Command1(0)→(3)**，這樣程式可少打一行，因為若 **Index=0** 表示尚未按任意按鈕。



圖 2-24 控制項陣列詢問視窗

- 另外 **G(7)~G(0)**與 **R(7)~R(0)**用來顯示 16 顆 LED 對應的狀態，記得先設定屬性後再開始進行複製，名稱 **name** 要修改，並也要利用控制項陣列方可，**Index** 順序不要弄錯，複製上去的位置大概就好，最後全部框選，利用格式→水平間距→相等，再用格式→對齊→主控項中間，即可將 16 個 **○** 物件一次對齊整，如圖 2-25 所示。

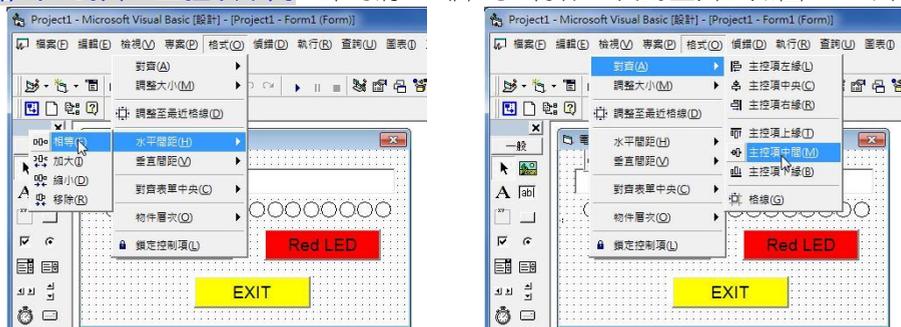


圖 2-25 將 16 個 **○** 物件對齊技巧

本書是利用 Visual Basic 6.0 程式來控制 16 顆 LED 燈號的順序，十道題目的程式碼幾乎完全相同，**僅儲存綠色 LED 狀態 b(99) 陣列宣告內容不同**而已，但每道試題都要再新增模組(module)，利用『**專案→新增模組**』，模組的預設檔名為 **module.bas**，以便自行在模組層次中宣告引用動態連結程式庫 (DLL) 內的外部程序，否則主程式就無法順利控制 LED，記得要『**儲存→關閉程式→重新開啟檔案**』方可正常使用。

本書光碟所提供的是『**USBIO.DLL**』檔案，**USB 函數庫及呼叫規範試場會於兩周前同術科通知單一併寄發，但是考試當天並不會提供給考生**，USBIO.DLL 宣告使用的方式如下：

- I. `Public Δ Declare Δ Function Δ OpenUsbDevice Δ Lib Δ "USBIO.DLL" (ByVal Δ MyUsbVendorID Δ As Δ Integer, Δ ByVal Δ MyUsbProductID Δ As Δ Integer) Δ As Δ Boolean`
- II. `Public Δ Declare Δ Sub Δ OutDataCtrl Δ Lib Δ "USBIO.DLL" (ByVal Δ OutData Δ As Δ Byte, Δ ByVal Δ OutControl Δ As Δ Byte)`
- III. `Public Δ Declare Δ Sub Δ CloseUsbDevice Δ Lib Δ "USBIO.DLL" ()`
- IV. `Public Δ Const Δ VendorID = &H1234`
- V. `Public Δ Const Δ ProductID = &H6789`



上述 Δ 並非程式碼，是表示**空白字元**，用來**區別指令與變數**，以方便讀者閱讀觀看，I~V 標題也不用輸入。並非所有試場所提供的 DLL 宣告方式都相同，且 DLL 檔案應放置於撰寫程式的**相同資料夾**，**並且要先儲存程式，關閉再重新開啟**才能正常執行控制。

當執行檔欲呼叫動態連結檔，會先到本身的資料夾去尋找，但若找不到便會到作業系統指定位置『C:\WINDOWS\system32』或是 VB6 發展環境之工作資料夾『C:\Program Files\Microsoft Visual Studio\VB98』內去尋找(Windows XP 作業系統)，在 Windows 7 作業系統下是在『C:\Program Files (x86)\Microsoft Visual Studio\VB98』，所以強烈建議將 **DLL 動態連結檔放置在撰寫程式相同的資料夾內比較方便**。

在模組層次中用來宣告引用 DLL 語法如下：

`[Public | Private] Δ Declare Δ Sub Δ name Δ Lib Δ "libname" [([arglist])]`

Declare 陳述式的語法具有下列幾個單元：

1. **Public**(公用)選擇性引數：用來宣告程序可在**所有模組**的其它程序中使用。
2. **Private**(私人)選擇性引數：用來宣告程序只可在宣告它的模組中使用。
3. **Function**(函數)選擇性引數：表示此程序**可傳回運算式中所使用的值**，例如：第一行 `Function Δ OpenUsbDevice`，因為有傳回是否有偵測到 USB 裝置的值，所以最後面要加 `As Δ Boolean` 當成布林值傳回。
4. **Sub**(副程式)選擇性引數：表示此程序不傳回任何值，例如：第 2~3 行的 `Sub Δ OutDataCtrl` 與 `CloseUsbDevice` **不需傳回任何值**，所以最後面僅用 () 即可。

CloseUsbDevice 是用來通知關閉移除 USB，不用傳遞任何參數，但 OutDataCtrl 因為還要告知資料與控制位置，所以需要傳遞參數，請參閱第 8 項說明。

- ※5. name 必要的引數：任何正確的**程序名稱**。請注意 DLL 之進入點的大小寫是有差別的，上述是以『OutDataCtrl』變數名稱命名，此名稱必須要和主程式呼叫名稱相同（大小寫需注意），雖然 VB 的程式碼會自動轉換大小寫，但對於使用者自訂的變數名稱並不會改變，造成在程式執行呼叫 DLL 時發生錯誤，這點考生需要特別留意！

VB 程式碼是大小寫均可，但除了在模組裡面的『OpenUsbDevice』、『OutDataCtrl』、『CloseUsbDevice』大小寫務必要一模一樣方可，其他的地方就無所謂，例如：模組程式碼原本應輸入『OpenUsbDevice』，因不小心改為『openUsbDevice』就會發生如圖 2-26 的錯誤，而在表單的程式碼內，若同樣輸入全部小寫，當你一執行程式時，會自動恢復大小寫狀態，所以沒有關係。請在參閱之前 P2-12 的圖 2-14，有提到利用 Dependency Walker 軟體查看 USBIO.DLL 裡面包含的函數，所以大小寫務必一致。



圖 2-26 因為大小寫不同而造成的錯誤呼叫訊息

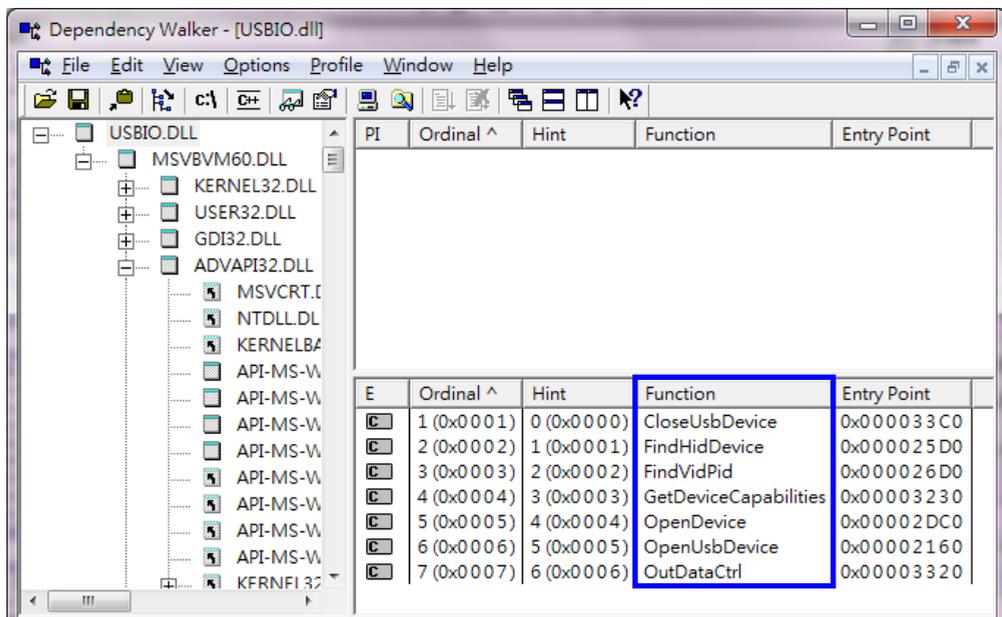


圖 2-14 查看 USBIO.DLL 裡面內容包括哪些函數 (同 P2-12)

6. **Lib** 必要的引數：表示含有宣告程序的 DLL 或程式碼來源，Lib 子句在所有的宣告裏皆為必要的引數。
7. **libname** 必要的引數：指含有宣告程序的 DLL 的名稱，須由試場提供告知，例如本書所使用的『**USBIO.DLL**』檔案，**大小寫均可**。
8. **arglist** 選擇性引數：當程序被呼叫時，所傳來當作引數的變數陣列。
詳細宣告請繼續看底下說明：

[ByVal | ByRef] ΔvarnameΔ[()] [AsΔtype]

上述 **arglist** 引數具有下列語法及單元：

- i. **ByVal** 選擇性引數：表示以『**傳值**』的方式來傳送引數。
 - ii. **ByRef** 選擇性引數：表示以『**傳址**』的方式來傳送引數。
 - iii. **varname** 必要的引數：傳給程序引數的變數名稱，由試場提供告知，**要背**。
 - iv. **type** 選擇性引數：傳給程序之引數的資料型態，可以是 **Byte**（位元組）、**Boolean**（布林值）、**Integer**（整數）、**Long**（長整數）、**Currency**（貨幣）、**Single**（單精準）、**Double**（倍精準）…。
9. 最後 **Const ΔVendorID=&H1234** 與 **ProductID=&H6789** 是用來識別 USB 裝置，**Const** 表示**常數**，&H 表示**16 進制**，**數字一定要相同才可識別**，**大小寫均可**。



DLL 的宣告是不是有一大堆的指令擔心輸入錯誤，或是根本背不起來呢？

本書建議各位考生可利用『**開始功能表** → **程式集** → **Microsoft Visual Basic 6.0** → **Microsoft Visual Basic 6.0 工具** → **API 檢視員**』，如圖 2-27 所示，然後選擇『**檔案** → **載入文字檔**』，選擇『**WIN32API.TXT**』檔案開啟。然後在『**可用的項目**』任意選擇一個範例，例如最後一個的 **Zero Memory** 上方點擊兩下，即可在最下面的地方找到基本範例，如圖 2-28 所示，其中 **Public**、**Declare**、**Sub**、**Lib**、**As**、**ByVal** 等基本指令都可以被參考使用，沒用到的宣告內容不要加入，您可以按右邊的『**複製**』鈕，或按『**Ctrl+C**』複製，再到 **Visual Basic 6.0** 按『**Ctrl+V**』貼上進行修改，但有些試場並不提供 **API 檢視員**，所以為了保險起見，建議讀者最好還是要全背起來，實際詳細操作內容請自行參閱本書教學光碟。



圖 2-27 API 檢視員視窗



圖 2-28 已載入範本 API 檢視員視窗

Visual Basic 6.0 第一題主程式碼 (註：程式碼不分大小寫，數字編號不用輸入)

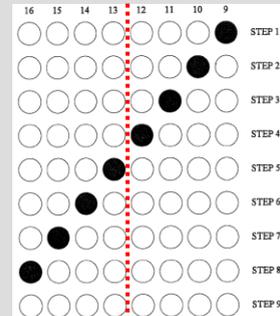
```

1. Dim a,b(99),cAsInteger
2. PrivateSubCommand1_Click(IndexAsInteger)
3. a=Index
4. c=0
5. EndSub
6. PrivateSubdisplay(no)
7. Fori=0To7
8. IfnoMod2=1Anda=1ThenG(i).FillColor=RGB(0,255,0)
   ElseG(i).FillColor=RGB(0,128,0)
9. IfnoMod2=1Anda=2ThenR(i).FillColor=RGB(255,0,0)
   ElseR(i).FillColor=RGB(128,0,0)
10. no=no\2
11. Nexti
12. EndSub
13. PrivateSubTimer1_Timer()
14. b(0)=1
15. b(1)=2
16. b(2)=4
17. b(3)=8
18. b(4)=&H10
19. b(5)=&H20
20. b(6)=&H40
21. b(7)=&H80
22. Text1.Text="Current Time:"&&Time$
23. Fori=0To7
24. G(i).FillColor=vbWhite
25. R(i).FillColor=vbWhite
26. Nexti
27. IfOpenUsbDevice(VendorID,ProductID) Then
28. Fori=0To7
29. G(i).FillColor=RGB(0,128,0)
30. R(i).FillColor=RGB(128,0,0)
31. Nexti
32. OutDataCtrl0,0
33. OutDataCtrl0,16
34. Ifa=1ThenOutDataCtrl a,b(c),0:display(b(c))
35. Ifa=2 And a<=7 Then
36. OutDataCtrl2^c,32
37. OutDataCtrl2^c,48
38. display(2^c)
39. EndIf
40. EndIf
41. Ifa=3ThenCloseUsbDevice:End
42. Ifc>15Thenc=15Elsec=c+1
43. EndSub

```

僅需點擊命令鈕物件，即可自動增加副程式頭尾的部分，無須親自撰寫。

指令與變數之間需空白區隔，按下Enter之後，等號兩側都會自動增加空白字元，例如：a= index



註：上面程式特別用藍色粗體表示要大小寫一致，其餘指令或變數則都可以。



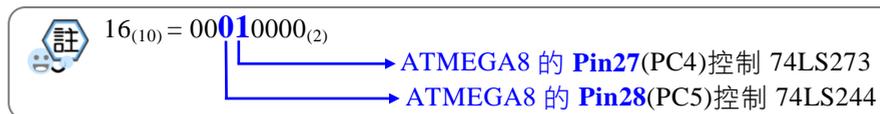
補充說明

Visual Basic 第一題程式說明 (※第二題到第十題程式說明省略)

- ※ 第 1 行：「`Dim a, b(99), c As Integer`」是同時宣告變數 a、b、c 當成整數型態。變數名稱均可自行修改，「`Dim`」用來宣告變數型態，「`As`」可翻譯為「當作」，而「`Integer`」表示整數型態。
變數 a 是用來記錄哪個按鈕被選擇；陣列 b 用來儲存綠色 LED 9~16 的資料，大小故意設大一點，是因為這樣十題都可以共用相同程式碼，十道題目狀態最多為 15 個，所以宣告的陣列大小值只要大於 14 即可(因為從 0 開始)；變數 c 是用來計數目前應顯示到第幾個狀態(步驟)。
- 第 3 行：當按下 Command1 陣列其中一個按鈕，它會利用「`Index`」變數傳回副程式，告知哪個按鈕被使用，所以要趕快利用變數 a 將目前的 `Index` 變數儲存，以便在副程式 `Sub Timer1_Timer()` 裡面判斷。
- 第 4 行：將計數變數 c 設為 0，每當按下 Command1 陣列其中一個按鈕，均從 b 陣列的第一個狀態開始顯示。同時為了配合第 36~38 行的 2^c 資料輸出，所以將初值設為 0，因為 $2^0=1_{(10)}=00000001_{(2)}$ ，使僅有第 1 顆紅色 LED1 點亮而已。
- 第 6 行：因為試題要在表單新增燈號同步顯示，故自行建立 `display` 副程式，該名稱可自訂，透過 `no` 變數傳遞資料即可，此行需自行輸入(第 12 行亦同)，可複製第 2~5 行再修改副程式名稱。
- 第 7~11 行：用來同步顯示表單對應的 LED 狀態，因為每種顏色 LED 都有 8 顆，所以 `For` 迴圈是 0~7，剛好可以配合 `G(0)~G(7)` 與 `R(0)~R(7)` 的 `Index` 值。
- ※ 第 8 行：`Mod` 指令是取餘數，配合第 10 行取商數，目的是做『十進制轉二進制』，並從 LSB 低位元開始轉換。若 `no Mod 2=1` 而且 `GreenLED` 綠色按鈕被按下，則對應的 `G(0)~G(7)` 填滿純綠色，否則填入淺綠色。指令 `RGB(紅色 Red 值,綠色 Green 值,藍色 Blue 值)`，每個顏色的數值介於 0~255 之間，所以要顯示純紅色就是 `RGB(255,0,0)`、純綠色就是 `RGB(0,255,0)`、純藍色就是 `RGB(0,0,255)`，若要顯示較淡顏色就給一半的值，例如：淺綠色為 `RGB(0,128,0)` 的值。
- 第 9 行：若 `no Mod 2=1` 而且 `RedLED` 紅色按鈕被按下，則對應的 `R(0)~R(7)` 填滿純紅色 `RGB(255,0,0)`，否則填入淺紅色 `RGB(128,0,0)` 的值。
- 第 10 行：將傳遞的值 `no` 除以 2，以取出整數，目的是做『十進制轉二進制』。

- ➡第 13~43 行：定時器 Timer1 預設值的 Interval=1000，所以每隔 1 秒鐘都會去檢查 USB 是否已經插入，若偵測到裝置才送出資料控制 LED。
- ※➡第 14~21 行：**&H** 表示數字內容為十六進制，亦可忽略**&H**，但需自行轉換為十進制(不建議)，該段內容是依照各試題綠色 LED9~16 顯示順序所設定的值，**LED 燈亮表示邏輯「1」**，**LED 熄滅表示邏輯「0」**，然後將對應的二進制轉成十六進制碼，**建議在右邊第 4、5 個位元中間自行加一條虛線，比較方便進行十六進制碼轉換，當數字內容不超過 10，則可以忽略&H**，因為兩者大小是相同，由於當未設定初值的內容皆為 0，所以最後一個全部熄滅的狀態可以忽略；而十道試題僅有該 b 陣列的宣告內容不同而已(參考 P2-31~33)，**考生應要會二進制轉換為十六進制，而非死背答案。**
- ➡第 22 行：依試題顯示目前系統的時間，Current Time 是表示目前時間，請參閱試題，**大小寫均需和試題範例一致**，不要輸入錯誤。而「&」是表示**連接符號，可將兩段內容結合起來**；「Time」是內建的副程式名稱，目的是顯示系統時間（**時間要正確**），**若忽略錢號「\$」則會多顯示上午或下午**，最好依據試題範例時間顯示格式顯示，否則會被扣分。
- ➡第 23~26 行：目的是先將表單對應的 G(0)~G(7)、R(0)~R(7)的 LED 燈號全部顯示中空白色，因為試題要求若 USB 尚未插入，應顯示中空白色，其中 vbWhite 是指**白色**，但亦可用 **RGB(255,255,255)**代替。
- ※➡第 27~40 行：偵測 USB 裝置是否插入？當 OpenUsbDevice(VendorID,ProductID)為**真 True**，表示 USB 裝置已經可以正常使用，所以開始控制 LED，由於該函數是 USBIO.DLL 裡面的函數，所以**OpenUsbDevice 大小寫均需一致，而且不可輸入錯誤。**
- ➡第 28~31 行：由於 USB 裝置插入，所以將表單對應的 G(0)~G(7)、R(0)~R(7) LED 燈號全部顯示「**不亮**」狀態，即只給一半的 128 數值。
- ➡第 32~33 行：目的是將介面卡的 16 顆 LED 全部熄滅。其中「**OutDataCtrl**」並非 Visual Basic 的內建指令，而是在之前宣告模組所設定的副程式名稱，所以**大小寫均需一致，而且不可輸入錯誤。**
- ※➡第 32 行：程式 OutDataCtrl 0,0 左邊的 0 是指**輸出資料 OutData**(0 表示全不亮)；右邊的 0 是指**控制訊號 OutControl**，因為 $0_{(10)}=000000_{(2)}$ ，所以可將 ATMEGA8 的 **Pin28(PC5)與 Pin27(PC4)設定為 0**，由於 ATMEGA8 的 Pin28(PC5)是連接到 74LS244 的致能端 $\overline{GOE}=0$ ，所以此時綠色 **LED9~16 全部熄滅**。但 74LS273 的 RCLK 時脈端還需正緣訊號方可觸發，目前僅是準備做觸發而已，所以此時紅色 **LED1~8 尚未熄滅**。

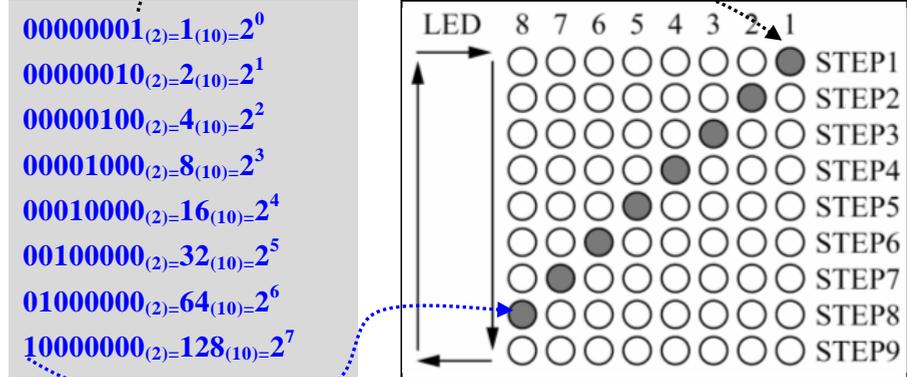
※第 33 行：因 $16_{(10)} = 0010000_{(2)}$ ，所以可將 ATMEGA8 的 Pin27(PC4)設定為 **1**，由於 ATMEGA8 的 Pin27(PC4)是連接到 74LS273 的 RCLK 時脈端，所以執行第 32 行與第 33 行的程式之後，其實是為了產生一個由 **0 到 1 的正緣觸發訊號**，方能將 0 的資料傳送到 74LS273 輸出端，此時所有紅色 LED1~8 全部熄滅。我們把清除 LED 燈號的動作放在每一秒鐘的動作，是為了當紅色 LED 顯示一半狀態時，若按下綠色 LED 按鈕，紅色 LED 會全部熄滅，這樣**才不會同時顯示兩色的 LED**。



第 34 行：**假如 (If)** $a=1$ ，表示目前『Green LED』按鈕被選擇，**然後 (Then)**就要將 b 陣列資料依序送出，並利用副程式 **display(b(c))** 同時更新表單的對應燈號。其中冒號是用來連接兩行程式，這樣可以省略一行 end if，OutDataCtrl b(c),0 右側的 0 可使 ATMEGA8 的 Pin28(PC5)為 0，使 74LS244 致能，以便能送出指定資料。

第 35~39 行：**假如 (If)** $a=2$ ，表示目前『Red LED』按鈕被選擇，**然後 (Then)**就要依序點亮紅色 LED1~8，並同時更新表單燈號。第 39 行 **If** 判斷句的結束(End)，在輸入時可直接輸入『**endif**』中間會自動新增一個空白，本段程式亦可都用冒號連接成一行程式。為何判斷還要加入『 **$c \leq 7$** 』，因為變數 c 會每秒自動遞增，當超過 2^8 會出現溢位錯誤，所以要停止送出資料。

※第 36~37 行：送出 74LS273 的正緣觸發訊號，為何右側的**控制訊號**不是同第 32、33 行的 0、16 數值，因為這樣**會同時點亮綠色 LED**，解決方式是改變控制值，因為 $32_{(10)} = 0010000_{(2)}$ 、 $48_{(10)} = 00110000_{(2)}$ ，由於 bit5=PC5=1，使 74LS244 一直失能，所以綠色 LED 都不會亮，而 bit4=PC4=0 到 1，使 74LS273 正緣觸發，紅色 LED 依序點亮。為何左側的資料固定為『 **2^c** 』，因為十道題目的要求都是將紅色 LED1 到 LED8 **依序點亮**，每一 LED 亮燈時間為一秒，其餘的 LED 熄滅，其中 0 表示熄滅，1 表示點亮，對應關係如下所示，其中**克拉 (“^”) 表示為次方 (乘冪) 的涵義**。由於為了簡化程式，當變數 c 大於 7 以後，因為第 35 行條件不再成立，所以紅色 LED 就全部熄滅(※ STEP9 即表示全滅)。



➡ 第 38 行：透過副程式 `display(2^c)` 同時更新表單的對應燈號。

➡ 第 41 行：假如(If) $a=3$ ，表示目前『EXIT』按鈕被選擇，然後(Then)就關閉 USB 裝置，同時結束(End)整個程式。

※➡ 第 42 行：由於計數器 c 會一直往上計數，當一直增加超過到 $2^c=2^{16}=65536$ (整數 16 位元的最大數字為 65535)，就會出現下圖 2-29 溢位 (Overflow) 的錯誤訊息視窗。所以當判斷變數 c 超過 15 時就要重新設定變數固定為 15，但若設 c 為 0，則是會從第一個狀態繼續再顯示一次，但因為試題是要求循環一次即可，最後要全部熄滅，因此，應將變數 c 重新設為 15 方可，若變數 c 還沒有超過 15，則自動加 1，以便變換下一個顯示的狀態。

註：為何不是令 $c=14$? 若是第 1~8 題都還不會出錯，但是因為第 9~10 題共有 $b(0)~b(14)$ 個狀態，令 $c=14$ ，畫面會停留在最後一個狀態 $b(14)$ ，而無法將全部 LED 熄滅。那為何令 $c=15$ 會全部熄滅，因為 $b(15)$ 並未設定初值，所以預設值為 0，因此，畫面會一直停留在 $b(15)$ 都不亮的畫面。

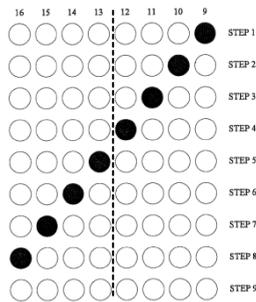


➡ 圖 2-29 溢位 (Overflow) 的錯誤訊息視窗

補充：底下為第 1~10 題程式碼，僅列出第 14 行開始的 b 陣列初值宣告，其餘均相同

Visual Basic 6.0 第一題主程式碼 (※ 僅 b 陣列初值設定)

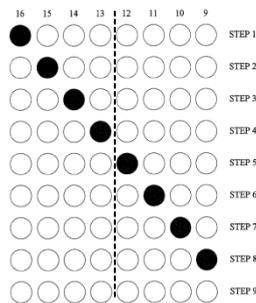
13. b(0)=1
14. b(1)=2
15. b(2)=4
16. b(3)=8
17. b(4)=&H10
18. b(5)=&H20
19. b(6)=&H40
20. b(7)=&H80



※b(8)以後沒設定，初值皆為 0，表示 LED 全滅

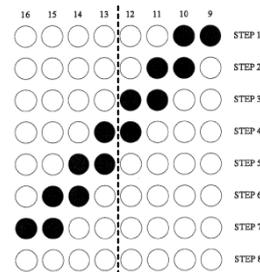
Visual Basic 6.0 第二題主程式碼 (※ 僅 b 陣列初值設定)

14. b(0)= &H80
15. b(1)= &H40
16. b(2)= &H20
17. b(3)= &H10
18. b(4)=8
19. b(5)=4
20. b(6)=2
21. b(7)=1



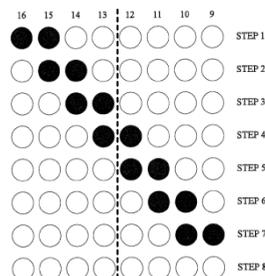
Visual Basic 6.0 第三題主程式碼 (※ 僅 b 陣列初值設定)

14. b(0)=3
15. b(1)=6
16. b(2)= &HC
17. b(3)= &H18
18. b(4)=&H30
19. b(5)=&H60
20. b(6)=&HC0



Visual Basic 6.0 第四題主程式碼 (※ 僅 b 陣列初值設定)

14. b(0)= &HC0
15. b(1)= &H60
16. b(2)= &H30
17. b(3)= &H18
18. b(4)=&HC
19. b(5)=6
20. b(6)=3



Visual Basic 6.0 第五題主程式碼 (※ 僅 b 陣列初值設定)

- | | | |
|---------------|-----------------|--------|
| 14. b(0)=1 | ○ ○ ○ ○ ○ ● | STEP 1 |
| 15. b(1)=3 | ○ ○ ○ ○ ● ● | STEP 2 |
| 16. b(2)=7 | ○ ○ ○ ○ ● ● ● | STEP 3 |
| 17. b(3)= &HF | ○ ○ ○ ○ ● ● ● ● | STEP 4 |
| 18. b(4)=&H1F | ○ ○ ○ ● ● ● ● ● | STEP 5 |
| 19. b(5)=&H3F | ○ ○ ● ● ● ● ● ● | STEP 6 |
| 20. b(6)=&H7F | ○ ● ● ● ● ● ● ● | STEP 7 |
| 21. b(7)=&HFF | ● ● ● ● ● ● ● ● | STEP 8 |
| | ○ ○ ○ ○ ○ ○ ○ ○ | STEP 9 |

Visual Basic 6.0 第六題主程式碼 (※ 僅 b 陣列初值設定)

- | | | |
|----------------|-----------------|--------|
| 14. b(0)= &H80 | ● ○ ○ ○ ○ ○ ○ ○ | STEP 1 |
| 15. b(1)= &HC0 | ● ● ○ ○ ○ ○ ○ ○ | STEP 2 |
| 16. b(2)= &HE0 | ● ● ● ○ ○ ○ ○ ○ | STEP 3 |
| 17. b(3)= &HF0 | ● ● ● ● ○ ○ ○ ○ | STEP 4 |
| 18. b(4)=&HF8 | ● ● ● ● ● ○ ○ ○ | STEP 5 |
| 19. b(5)=&HFC | ● ● ● ● ● ● ○ ○ | STEP 6 |
| 20. b(6)=&HFE | ● ● ● ● ● ● ● ○ | STEP 7 |
| 21. b(7)=&HFF | ● ● ● ● ● ● ● ● | STEP 8 |
| | ○ ○ ○ ○ ○ ○ ○ ○ | STEP 9 |

Visual Basic 6.0 第七題主程式碼 (※ 僅 b 陣列初值設定)

- | | | |
|----------------|-----------------|--------|
| 14. b(0)= &H18 | ○ ○ ○ ● ● ○ ○ ○ | STEP 1 |
| 15. b(1)= &H24 | ○ ○ ● ○ ○ ● ○ ○ | STEP 2 |
| 16. b(2)= &H42 | ○ ● ○ ○ ○ ○ ● ○ | STEP 3 |
| 17. b(3)= &H81 | ● ○ ○ ○ ○ ○ ○ ● | STEP 4 |
| | ○ ○ ○ ○ ○ ○ ○ ○ | STEP 5 |

Visual Basic 6.0 第八題主程式碼 (※ 僅 b 陣列初值設定)

- | | | |
|----------------|-----------------|--------|
| 14. b(0)= &H81 | ● ○ ○ ○ ○ ○ ● | STEP 1 |
| 15. b(1)= &H42 | ○ ● ○ ○ ○ ● ○ | STEP 2 |
| 16. b(2)= &H24 | ○ ○ ● ○ ○ ● ○ ○ | STEP 3 |
| 17. b(3)= &H18 | ○ ○ ○ ● ● ○ ○ ○ | STEP 4 |
| | ○ ○ ○ ○ ○ ○ ○ ○ | STEP 5 |

Visual Basic 6.0 第九題主程式碼 (※ 僅 b 陣列初值設定)

	16	15	14	13	12	11	10	9	
14. b(0)=1	○	○	○	○	○	○	○	●	STEP 1
15. b(1)=2	○	○	○	○	○	○	●	○	STEP 2
16. b(2)=4	○	○	○	○	○	●	○	○	STEP 3
17. b(3)=8	○	○	○	○	○	○	○	○	STEP 4
18. b(4)=&H10	○	○	○	○	○	○	○	○	STEP 5
19. b(5)=&H20	○	○	○	○	○	○	○	○	STEP 6
20. b(6)=&H40	○	○	○	○	○	○	○	○	STEP 7
21. b(7)=&H80	○	○	○	○	○	○	○	○	STEP 8
22. b(8)=&H40	○	○	○	○	○	○	○	○	STEP 9
23. b(9)=&H20	○	○	○	○	○	○	○	○	STEP 10
24. b(10)=&H10	○	○	○	○	○	○	○	○	STEP 11
25. b(11)=8	○	○	○	○	○	○	○	○	STEP 12
26. b(12)=4	○	○	○	○	○	○	○	○	STEP 13
27. b(13)=2	○	○	○	○	○	○	○	○	STEP 14
28. b(14)=1	○	○	○	○	○	○	○	○	STEP 15
	○	○	○	○	○	○	○	○	STEP 16

Visual Basic 6.0 第

十題主程式碼 (※ 僅 b 陣列初值設定)

	16	15	14	13	12	11	10	9	
14. b(0)= &H80	●	○	○	○	○	○	○	○	STEP 1
15. b(1)= &H40	○	●	○	○	○	○	○	○	STEP 2
16. b(2)= &H20	○	○	○	○	○	○	○	○	STEP 3
17. b(3)= &H10	○	○	○	○	○	○	○	○	STEP 4
18. b(4)=8	○	○	○	○	○	○	○	○	STEP 5
19. b(5)=4	○	○	○	○	○	○	○	○	STEP 6
20. b(6)=2	○	○	○	○	○	○	○	○	STEP 7
21. b(7)=1	○	○	○	○	○	○	○	○	STEP 8
22. b(8)=2	○	○	○	○	○	○	○	○	STEP 9
23. b(9)=4	○	○	○	○	○	○	○	○	STEP 10
24. b(10)=8	○	○	○	○	○	○	○	○	STEP 11
25. b(11)=&H10	○	○	○	○	○	○	○	○	STEP 12
26. b(12)=&H20	○	○	○	○	○	○	○	○	STEP 13
27. b(13)=&H40	○	○	○	○	○	○	○	○	STEP 14
28. b(14)=&H80	○	○	○	○	○	○	○	○	STEP 15
	○	○	○	○	○	○	○	○	STEP 16



補充說明 Visual Basic 6.0 程式比對軟體說明

有鑑於每次集體輔導學生遭遇的最大問題就是對 VB 程式不熟悉，有時僅是一個英文字母打錯，檢查了許久卻查不出問題所在，造成無法順利通過考試的遺憾！所以作者特地撰寫程式比對軟體，在撰寫區的程式碼會和 1.txt~10.txt 與 module.txt 逐行比對，若有內容不同會在解答區做 提示，當有要自行變更程式變數名稱可直接用『**筆記本**』編輯對應的 txt 檔即可，**程式除了指令與變數之間需留『一個空白字元』以區別(※其他地方不可輸入，但最前、後面的空白會自動刪除再比對)**，否則程式會判斷有錯誤，您必須一直反覆練習到完全沒有一次錯誤，因為實際考試並沒有人會提醒各位哪裡是否有輸入錯誤狀況的發生？



圖 2-30 VB 程式比對畫面 (※ 2014 年已經更新為 USB 介面版本)

茲將上述比對程式操作應留意的地方說明如下：

1. 在主程式與模組程式碼撰寫區已預先提供 4 行程式與 P2-25 提到的 API 檢視員所複製的資料，練習時僅需直接進行模組程式碼修改即可，**大小寫均可**。
2. **指令與變數之間才需加入「一個空白字元」**，其他地方均不可再加入空白。所有程式碼務必完全相同方可逐行進行文字比對，否則該行會視同錯誤！
3. 按下『**比對程式**』或勾選『**顯示解答**』才會看到解答區所有內容，以利練習。
4. 選擇不同題號，下方自動切換各題綠色 LED 狀態，以利程式撰寫。

2-7 焊接技巧補充說明與注意事項

在第一次焊接之前，請先使用本書光碟提供的模擬程式練習，如圖 2-31 所示，相信先熟悉所有接線之後再開始進行實際焊接練習，可以幫您達到事半功倍的效果。

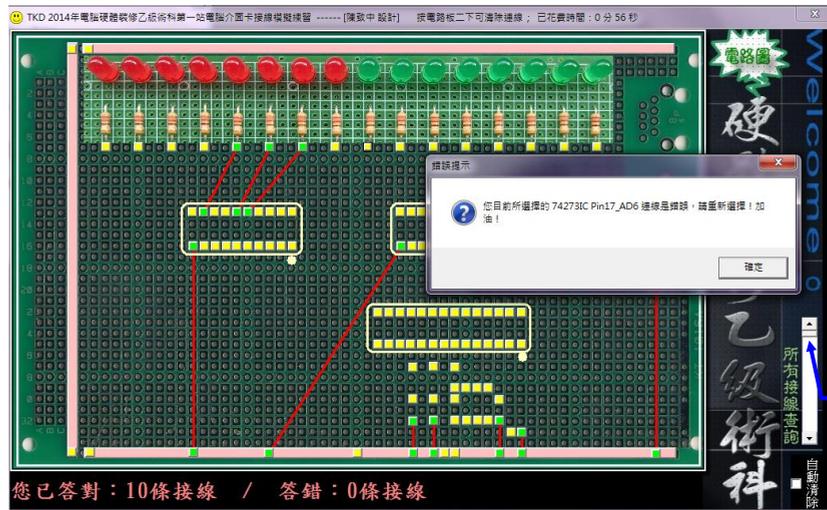


圖 2-31 模擬焊接程式畫面 (※ 2014 年已經更新 USB 介面版本)

茲將該模擬程式的幾項特殊功能說明如下：

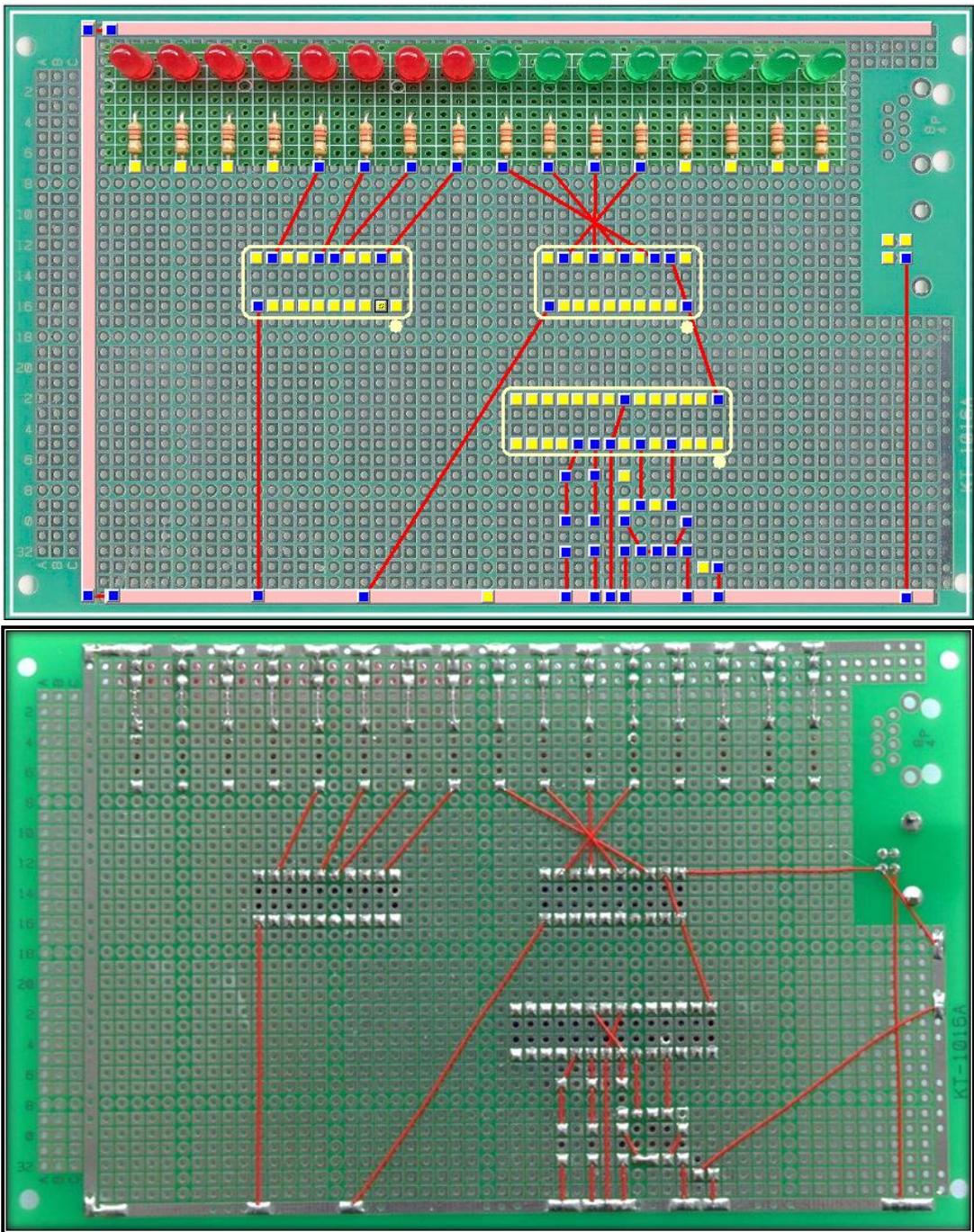
1. 表單標題會顯示測驗作答時間，且當滑鼠移到焊點上方，會自動顯示該腳位的定義名稱。
2. 若點選錯誤會自動出現錯誤訊息，並告知錯誤所在，同時該焊點會一直以紅、綠顏色交互閃爍方式告知，可提醒考生以後焊接該焊點時就要特別注意。
3. 在視窗最下方會顯示目前答對與答錯數，必須練習到完全沒有錯誤狀況，由於本程式會主動告知選擇錯誤，但實際焊接時，即使只有一條錯誤並不容易被發現，所以平時需反覆練習到沒有任何錯誤；若忘記電路連接方式，不妨點選右上方『電路圖』的圖示即可，方便觀察連接方式，該查詢不影響測驗的進行。
4. 若您是第一次練習，可點選右下方的垂直捲軸，它會先清除所有的接線，然後逐一手動顯示所有接線方式，包括文字訊息的提示，方便考生能夠快速記憶所有接線，一旦有查詢的記錄發生，會自動停止計時評分，這可以提供老師在指導學生時，方便進行焊接前學習的考核與記錄。
5. 所有 LED 負端共同接地，本模擬程式並未加入考慮這些焊點的模擬練習。加上利用萬用板周圍當成 GND，由於程式必須點對點連接判斷，所以其餘接地的連接方式須和範例相同才算正確，但實際考試只要確認有接到 GND 均可。
6. 74LS244 與 74LS273 所有腳位都有連接，ATMEGA8 則還有 10 隻腳沒有使用，所以總共『67』條接線要焊接，當全部答對才會出現恭喜的提示視窗。

可查看作者推薦的焊接順序

7. 實際焊接建議全部使用直線與拋物線，可避免在 IC 中間因為有接線而造成短路，茲將 67 條連線方式全部列出，強烈建議模擬連接順序就是實際焊接順序。

◎ STEP1:先焊 31 條「直線」

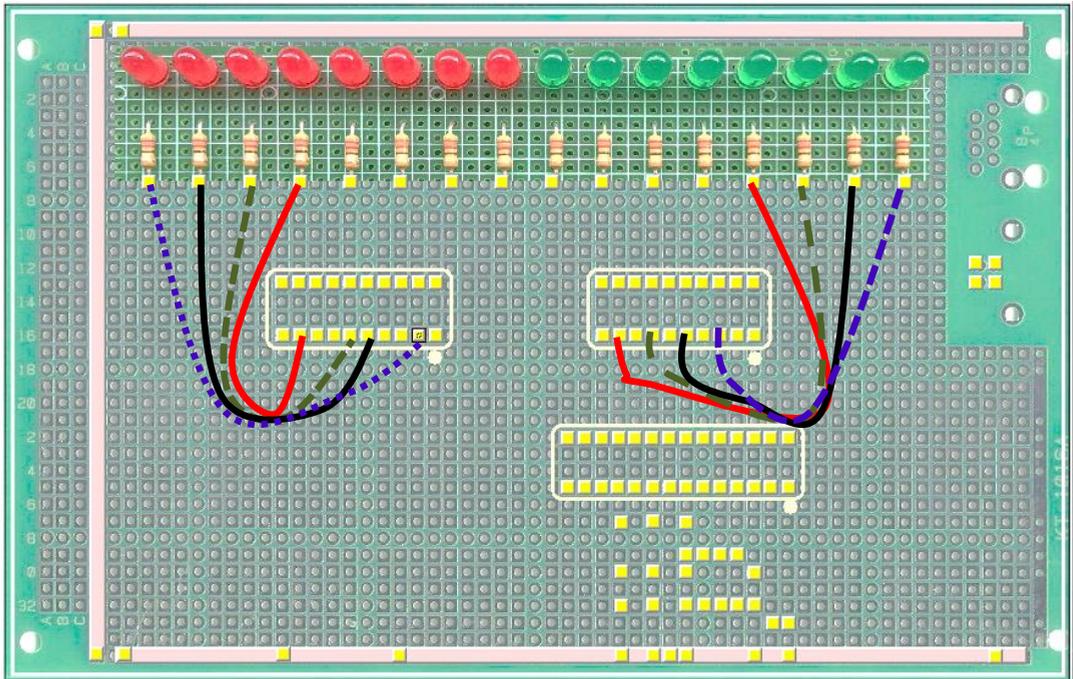
- ※ 第 1 條接線：USB_GND → GND
- ※ 第 2 條接線：上 GND → 左 GND
- ※ 第 3 條接線：左 GND → 下 GND
- ※ 第 4 條接線：74273IC Pin10 → GND
- ※ 第 5 條接線：ATMEGA8-16PU Pin10 → Q1 石英晶體
- ※ 第 6 條接線：Q1 石英晶體 → C1
- ※ 第 7 條接線：C1 → GND
- ※ 第 8 條接線：ATMEGA8-16PU Pin9 → Q1 石英晶體
- ※ 第 9 條接線：Q1 石英晶體 → C2
- ※ 第 10 條接線：C2 → GND
- ※ 第 11 條接線：ATMEGA8-16PU Pin22 → ATMEGA8-16PU Pin8
- ※ 第 12 條接線：ATMEGA8-16PU Pin8 → GND
- ※ 第 13 條接線：D18+ → GND
- ※ 第 14 條接線：ATMEGA8-16PU Pin6→R17 (※故 R17 要對齊 ATMEGA8 Pin6)
- ※ 第 15 條接線：ATMEGA8-16PU Pin4→R18 (※故 R18 要對齊 ATMEGA8 Pin4)
- ※ 第 16 條接線：D17+ → GND
- ※ 第 17 條接線：C4- → GND
- ※ 第 18 條接線：D18- → R17 (※直接利用電阻接腳焊接)
- ※ 第 19 條接線：R17 → R19
- ※ 第 20 條接線：D17- → R18 (※直接利用電阻接腳焊接)
- ※ 第 21 條接線：74244IC Pin19 → 74244IC Pin1
- ※ 第 22 條接線：74244IC Pin1 → ATMEGA8-16PU Pin28_ $\overline{\text{GOE}}$
- ※ 第 23 條接線：74244IC Pin10 → GND
- ※ 第 24 條接線：74273IC Pin12 → R5
- ※ 第 25 條接線：74273IC Pin15 → R6
- ※ 第 26 條接線：74273IC Pin16 → R7
- ※ 第 27 條接線：74273IC Pin19 → R8
- ※ 第 28 條接線：74244IC Pin12 → R12
- ※ 第 29 條接線：74244IC Pin14 → R11
- ※ 第 30 條接線：74244IC Pin16 → R10
- ※ 第 31 條接線：74244IC Pin18 → R9



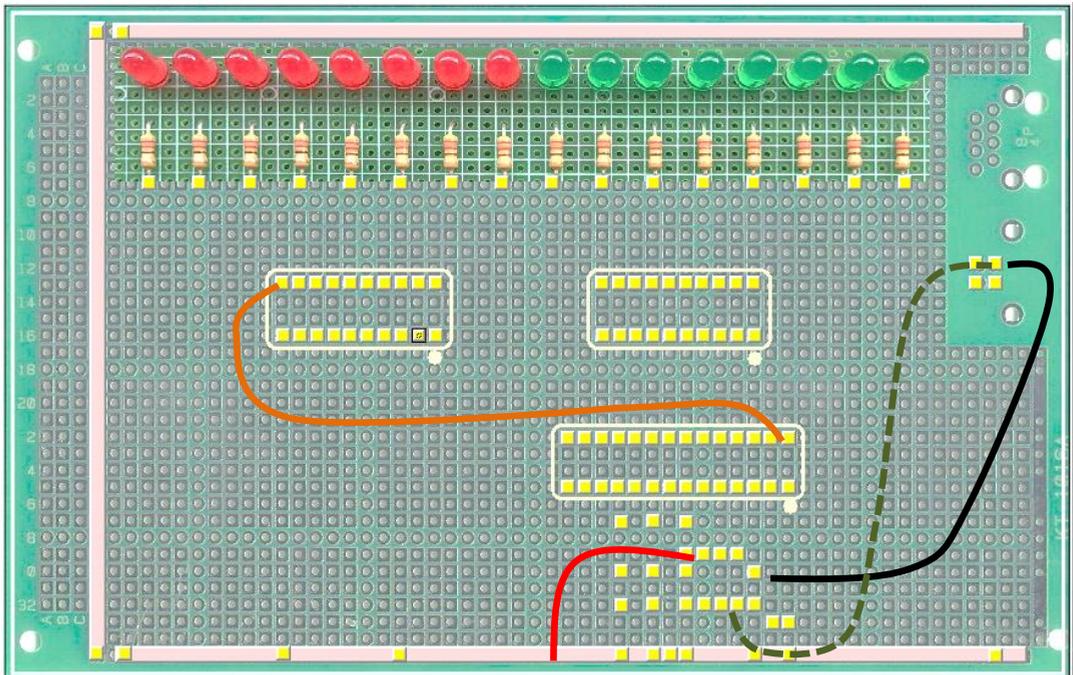
◎ STEP2:再焊 12 條「拋物線」(註：模擬程式僅提供點對點直線練習，無法呈現拋物線)

- ※ 第 32 條接線：R1 → 74273IC Pin2
- ※ 第 33 條接線：R2 → 74273IC Pin5
- ※ 第 34 條接線：R3 → 74273IC Pin6
- ※ 第 35 條接線：R4 → 74273IC Pin9
- ※ 第 36 條接線：R13 → 74244IC Pin9

- ※ 第 37 條接線：R14 → 74244IC Pin7
- ※ 第 38 條接線：R15 → 74244IC Pin5
- ※ 第 39 條接線：R16 → 74244IC Pin3



- ※ 第 40 條接線：C3 → GND
- ※ 第 41 條接線：74273IC Pin11_RCLK → ATMEGA8-16PU Pin27_RCLK
- ※ 第 42 條接線：USB_D+ → D17-
- ※ 第 43 條接線：USB_D- → R19



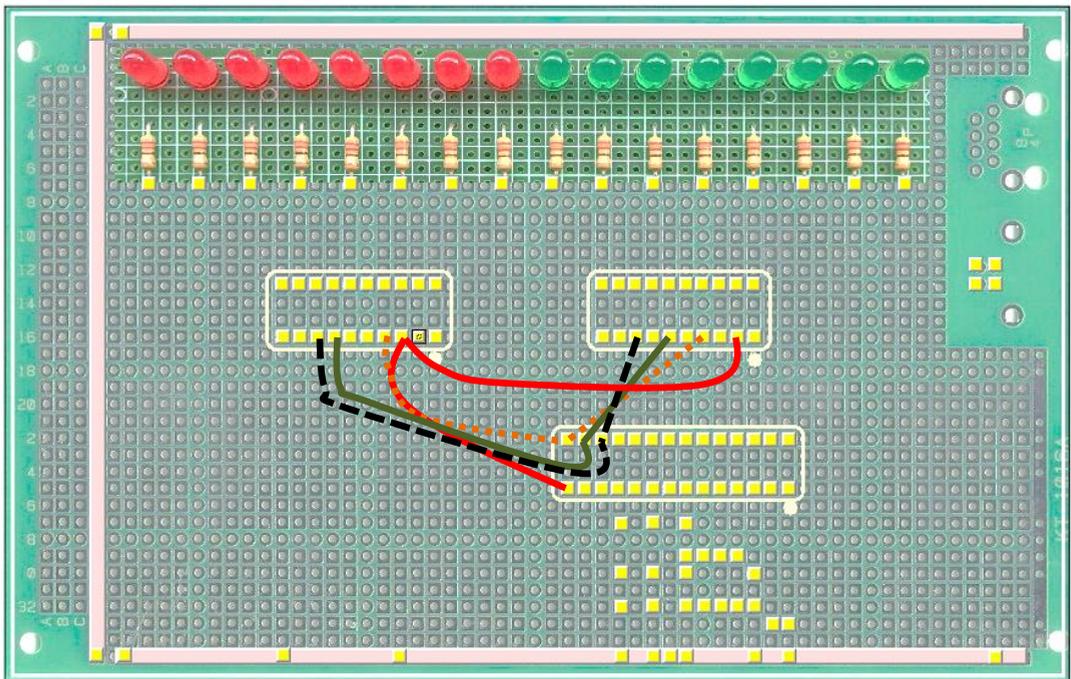
◎ STEP3: 焊 8 組「M 線」 (※ 中間不要剪斷)

- ※ 第 44 條接線：ATMEGA8-16PU Pin14_AD0 → 74273IC Pin3_AD0
- ※ 第 45 條接線：74273IC Pin3_AD0 → 74244IC Pin2_AD0

- ※ 第 46 條接線：74273IC Pin4_AD1 → ATMEGA8-16PU Pin15_AD1
- ※ 第 47 條接線：ATMEGA8-16PU Pin15_AD1 → 74244IC Pin4_AD1

- ※ 第 48 條接線：74273IC Pin7_AD2 → ATMEGA8-16PU Pin16_AD2
- ※ 第 49 條接線：ATMEGA8-16PU Pin16_AD2 → 74244IC Pin6_AD2

- ※ 第 50 條接線：74273IC Pin8_AD3 → ATMEGA8-16PU Pin17_AD3
- ※ 第 51 條接線：ATMEGA8-16PU Pin17_AD3 → 74244IC Pin8_AD3

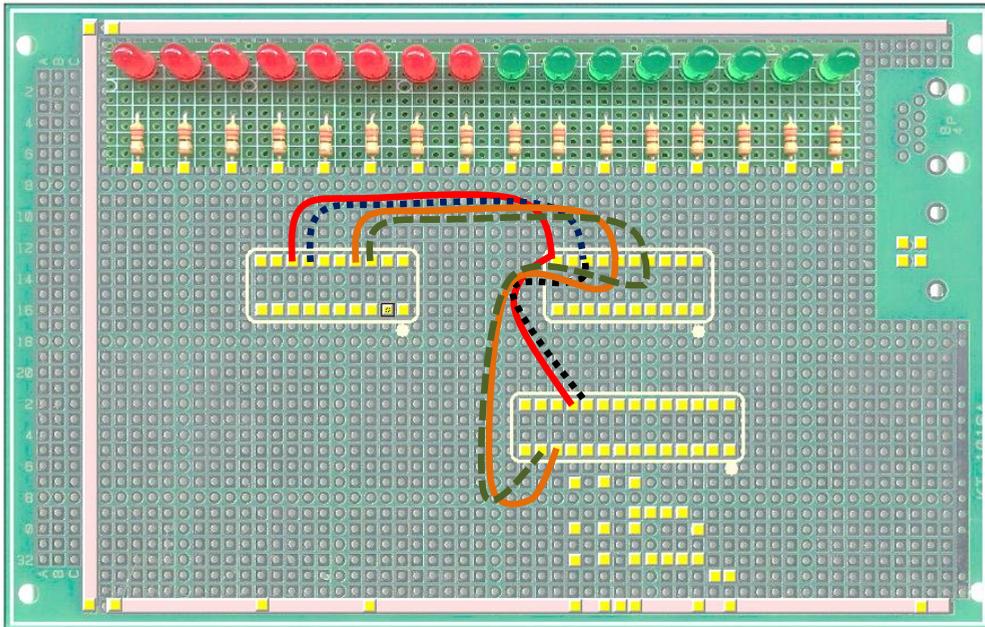


- ※ 第 52 條接線：74273IC Pin13_AD4 → 74244IC Pin11_AD4
- ※ 第 53 條接線：74244IC Pin11_AD4 → ATMEGA8-16PU Pin18_AD4

- ※ 第 54 條接線：74273IC Pin14_AD5 → 74244IC Pin13_AD5
- ※ 第 55 條接線：74244IC Pin13_AD5 → ATMEGA8-16PU Pin19_AD5

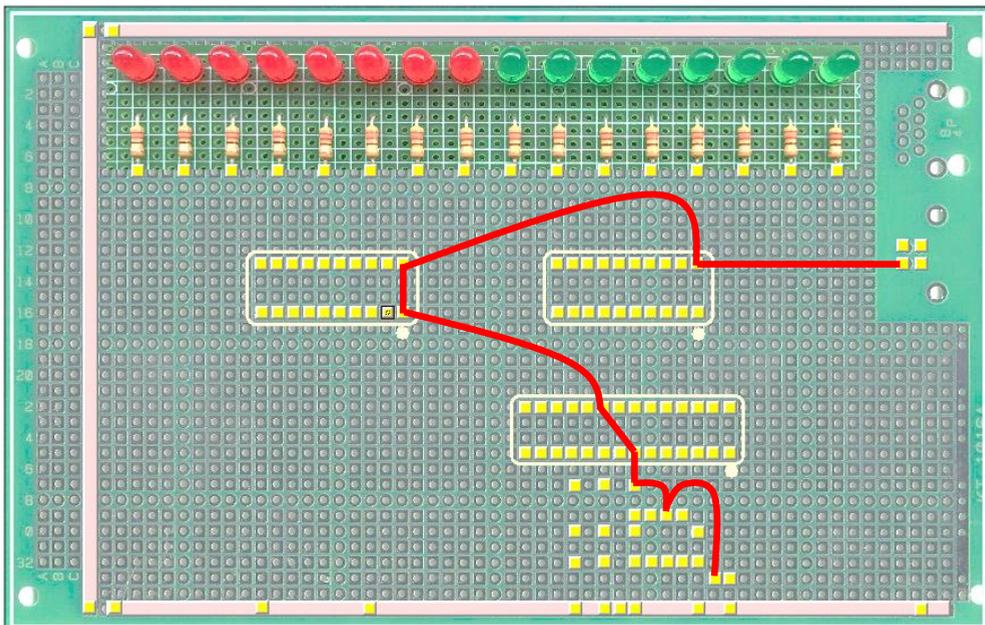
- ※ 第 56 條接線：74273IC Pin17_AD6 → 74244IC Pin15_AD6
- ※ 第 57 條接線：74244IC Pin15_AD6 → ATMEGA8-16PU Pin12_AD6

- ※ 第 58 條接線：74273IC Pin18_AD7 → 74244IC Pin17_AD7
- ※ 第 59 條接線：74244IC Pin17_AD7 → ATMEGA8-16PU Pin13_AD7



◎ **STEP4** : Vcc 順序也是採用 M 形線技巧，只要一條線直接串接「8 段」即可完成。

- ※ 第 60 條接線：USB_Vcc → 74244IC Pin20 (※ Vcc 共 8 條)
- ※ 第 61 條接線：74244IC Pin20 → 74273IC Pin20
- ※ 第 62 條接線：74273IC Pin20 → 74273IC Pin1
- ※ 第 63 條接線：74273IC Pin1 → ATMEGA8-16PU Pin20
- ※ 第 64 條接線：ATMEGA8-16PU Pin20 → ATMEGA8-16PU Pin7
- ※ 第 65 條接線：ATMEGA8-16PU Pin7 → C3
- ※ 第 66 條接線：C3 → R19
- ※ 第 67 條接線：R19 → C4+



在實際進行焊接時，除了按照上述步驟順序進行焊接，其他焊接的注意事項與技巧，歸納為底下幾項說明，若仍不清楚者可自行參閱本書精心錄製的多媒體教材光碟。

1. 請先確認萬用電路板的正面（零件面），**USB 埠的位置應在左側**，建議直接查看試場所提供介面卡成品的『**零件配置圖**』，請參閱本書 P2-4。



考試時因過於緊張，經常有考生將正、反面搞顛倒，請務必特別小心！

2. 零件擺設應依照『**由低到高**』順序擺設，而高度最矮的是**電阻 R1~R16**(※**稽納二極體先不要裝，等 U3 腳座確認再用**)，可先利用直尺將電阻腳位兩端『**一起折 90 度**』，以便折成**ㄇ**字形，但寬度須注意，再利用斜口鉗將電阻逐一剪下。
3. 將電阻 R1~R16『**從中間往兩側**』排列整齊，電阻腳位都先不要折到，利用任意紙張覆蓋在所有電阻上方，可避免電阻掉落，然後一起翻面進行焊接，焊接時有幾項技巧要特別注意，**首先旋轉 PCB 萬用板將欲焊接點（列）的方向，和我們的身體保持垂直**，在焊接時應隨時利用拿電烙鐵的手『**輕壓**』PCB 板上，這可使所有電阻能全部平貼於 PCB 萬用板上，以方便待會進行一連串的焊接。
4. 焊接時應先用電烙鐵將銅箔面加熱，再加入適當錒錫，焊點不宜太大否則容易短路，所以在適當時間內先將錒錫移走，最後才是移開電烙鐵，烙鐵頭應隨時保持清潔，但不是要在焊接之前才清潔，因為此時溫度會急遽下降，會比較不好焊接，而應該是在焊接之後就要先清潔，**全部焊完一邊就先利用斜口鉗剪掉多餘接腳**，不要等全部焊完再一起剪掉。
5. 第二個零件為 IC 座，先在欲擺設位置的其中一個焊點，暫時先加入一點錒錫，請注意 IC 座缺口方向應朝向 USB Port 同方向，然後用手指稍微頂住 IC 座，可避免在翻面時 IC 座脫落，重新加熱該焊點，在焊點尚未熔化之前，**切勿強行推入**，即可讓 IC 座暫時固定，千萬不要去折到 IC 腳位，否則待會不方便進行焊接，由於零件高度不同，**建議此時在四個角落加上銅柱**（※考場提供），**焊接時較不易左右搖晃**，再旋轉 PCB 板使 IC 座和我們身體保持垂直，這樣會比較方便進行其他焊點的焊接。**焊接時應隨時注意焊點不宜過大**，因為這樣會比較容易發生短路，但也不可以太少，因為待會 OK 單心線會不方便焊接，提供適當的錒錫是很重要的。
6. 接著開始將 ATMEGA8 相關零件擺至定位，詳細安裝順序自行參閱課本 P2-4。
7. 最後擺設零件是 16 顆 LED，請注意靠近 USB 埠端為綠色的 LED16，常犯的錯誤就是 LED 顏色左右弄相反，最多扣 60 分而不及格，同時也應注意 LED 腳位為『**長正短負**』，負端則可全部利用萬用板周圍已導通位置連接在一起。
8. 每放置一顆 LED 就先將接腳向外側扳開呈 90 度，LED 與接腳最好都能完全平貼於 PCB 面上，利用已彎曲 90 度的 LED 一隻接腳直接連接各電阻，斜口鉗的平面應朝向 LED 接腳末端，會比較方便精確剪到適當的長度，然後利用指

甲尖或一字起子壓住 LED 接腳，重新加熱該焊點即可，需稍微搖晃該接腳，以確定是否沒有空焊發生，當 16 顆 LED 全部固定後，再調整零件面的 LED 位置，避免上下位置歪曲變形，再將 LED 另一端接腳焊接固定，然後利用斜口鉗剪掉多餘長度。

9. 在進行單心線焊接之前，請先確認三顆 IC 的 Pin1 在哪裡，還有 USB 埠 4pin 腳位定義，千萬不要搞混囉！如圖 2-34 所示，在模擬軟體執行時，滑鼠游標若稍作停留，會自動出現各腳位的定義，包括之前 LED1 的位置都要熟記！

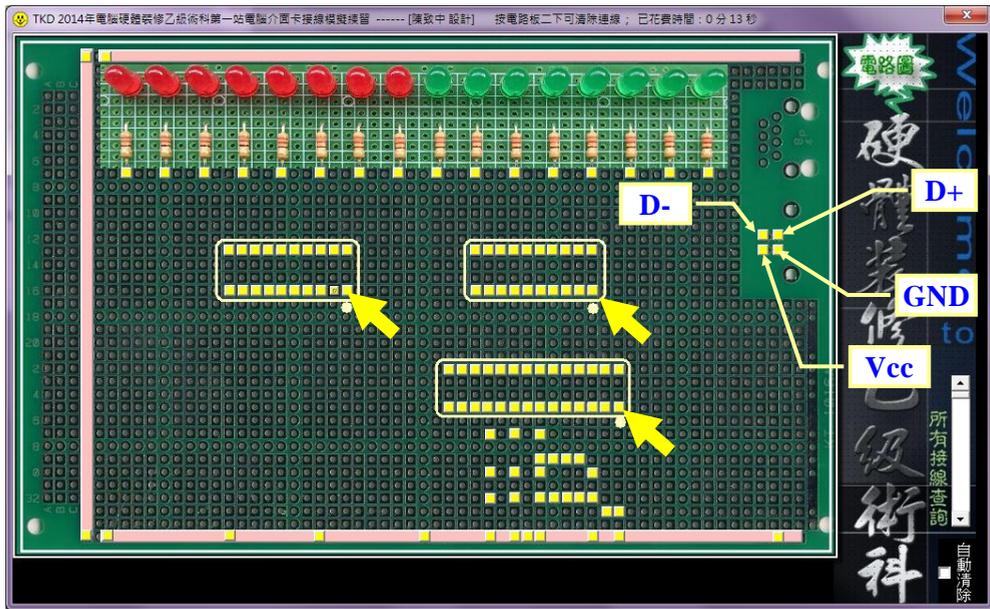


圖 2-34 萬用電路板背面（※ 需熟記各元件腳位位置）

10. 接下來是 AWG 30#單心線的焊接，其中 AWG 代表美國線規（American Wire Gauge），由於 OK 單心線非常細，若要利用斜口鉗剝皮，很容易因為太用力而全部剪斷，而裸露的長度「越短越好」，以避免和旁邊發生短路，當然你也可以特別去買一隻專用的「剝線鉗」，如圖 2-35 所示，在使用之前記得需先扳開開關，並使用 AWG 30#的位置，但為了加快考試速度，在這邊提供一種不是正統的焊接技巧，它是利用烙鐵頭的高溫直接將 OK 單心線外皮燙開，裸露的長度記得「越短越好」（※不可超過 2mm 為佳）。



圖 2-35 OK 單心線專用剝線鉗

11. 先剪一段適當長度的 OK 單心線，將一端的 OK 單心線剝皮，接著請特別注意底下焊接的分解動作：

- ❶ 將焊點銅箔面預先加熱
- ❷ 將 OK 單心線順勢推入
- ❸ 移開電烙鐵
- ❹ 等待兩秒鐘焊點冷卻，方可移動 OK 單心線
- ❺ 務必輕拉 OK 單心線，以檢查是否空焊，每次焊接完畢一定要順便檢查

另外一端先不要剪斷或剝皮，而是將 OK 單心線繞到該焊點側邊，並將 OK 單心線朝向另一邊方向輕拉，同時利用電烙鐵的最尖端將 OK 單心線稍微『往下、往內壓』，利用電烙鐵最尖的地方將 OK 單心線直接推入焊點內，如圖 2-36 所示，那是由於電烙鐵十分高溫，會同時將外皮剝開，即可同時對該焊點重新進行焊接；每焊一個點，最好都『輕拉一下 OK 單心線』，確定沒有空焊，否則待會很難從外觀再去檢查是否錯誤，所以這個動作真的非常地重要！

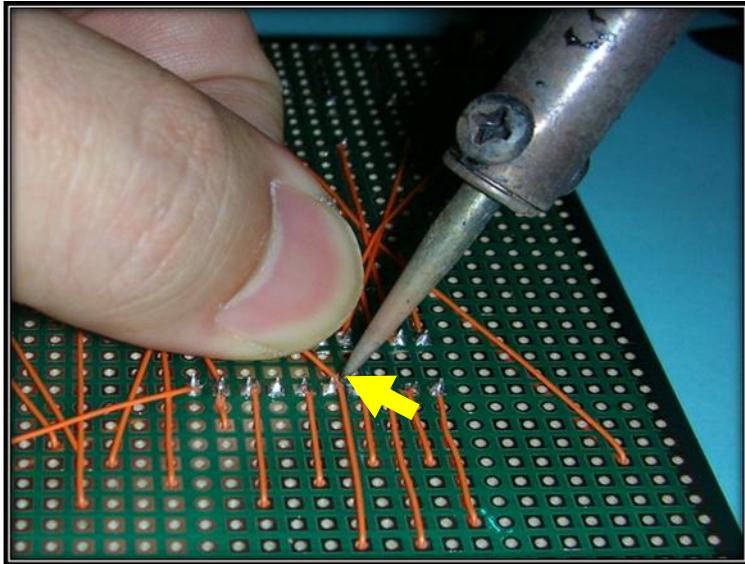


圖 2-36 焊接建議技巧

12. 在焊點冷卻之前切勿移動 OK 單心線，此時將多餘的 OK 單心線用斜口鉗剪掉，或利用左右搖擺 OK 單心線數次弄斷，因為這樣還可以同時檢查是否有空焊。
13. 若一開始 IC 座不小心裝錯方向，請勿驚慌，因為當 IC 插入時再特別留意正確方向即可，反正也沒人注意 IC 底下的 IC 座方向是否弄錯。
14. 若有三點以上要連接在一起，例如將 Vcc 全部直接串接在一起：USB_Vcc → 74244IC Pin20 → 74273IC Pin20 → 74273IC Pin1 → ATMEGA8-16PU Pin20 → ATMEGA8-16PU Pin7 → C3 → R19 → C4+，中間記得先不要剪斷也不要剝皮，然後利用同一條的 OK 單心線直接就再進行下一段的焊接，如圖 2-37 所示(圖片僅作參考)。

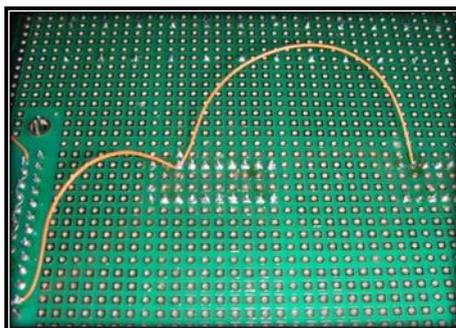


圖 2-37 避免分段進行焊接

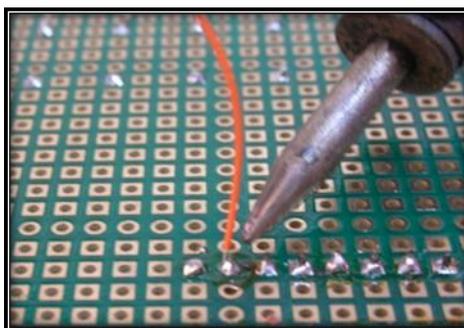


圖 2-38 將 OK 線與焊點保持『垂直』

15. 焊接最難的地方是如何避免發生短路，尤其焊到後面越麻煩，一不小心就將高溫電烙鐵頭碰到鄰近的 OK 線，因此**不要將 OK 單心線從尚未焊接的兩點中間直接通過，否則待會這兩點很難進行焊接**，根據筆者實際焊接經驗，建議依照本書 P36~P41 的焊接順序與方式會較不容易發生短路。
16. 焊接技巧採『**點對點拋物線**』進行，在進行點對點的連接時，每段 OK 線最好呈拋物線，所以**OK 線不宜太短**，使 OK 線端點可以與焊點呈『**垂直**』方式插入，如上圖 2-38 所示，這樣可避免與鄰近焊點造成短路現象，若是 OK 線使用太短會造成相鄰的焊點容易被擋住，將來會越來越難焊接，因為一不小心就會造成電烙鐵頭碰觸到其他 OK 線，而造成短路現象，成品如圖 2-6 與 2-7 所示(P2-6)，最後再將 OK 線平躺，**先利用考場提供軟體測試硬體電路是否正確，若無問題再作束線整理，千萬不要還未測試就先束線，萬一還要檢修，束線又要拆掉，浪費時間。**
17. 雖然考場有提供 IC 測試器，但無法測試 ATMEGA8 好壞，所以建議**利用考場在考試前提供的介面卡成品直接測試三顆 IC 好壞**，如圖 2-1(P2-2)，記得要取出 IC。
18. 第一次裝上 IC 是有技巧性，因為剛出產的 IC 腳位比較寬，無法直接插入 IC 座，所以需先插入一邊，然後再**用兩手指稍往外側推入一點**，等寬度和 IC 座相同時即可順利插上 IC，如圖 2-39 所示(圖片僅作參考)。

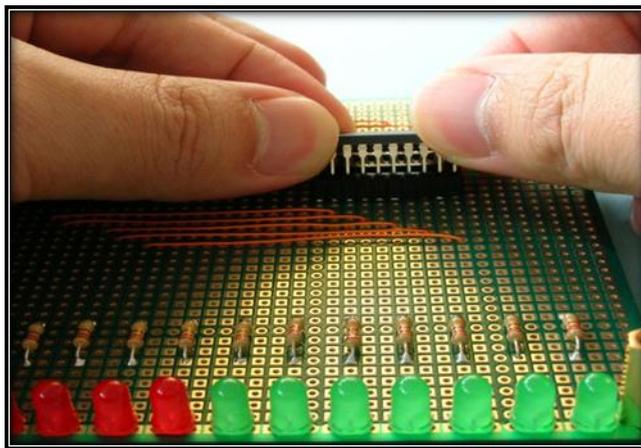


圖 2-39 第一次將 IC 置入的小技巧

2-8 各式故障狀況排除問與答 (Q&A)

建議各位在平時練習時，可以將軟體與硬體獨立分開練習，而硬體部分應是**加強 67 條單心線焊接的反覆練習**，在所有零件都不用拆除的情況下，目標應是練習在**50 分鐘之內完成「67」條單心線的焊接要求**。

由於在考試前，要先進行 ATMEGA8 韌體的測試，所以考場都會提供『介面卡測試程式執行檔』，所以硬體做完不要急著寫 VB 程式，而是先利用考場的程式進行測試，確認硬體沒問題再撰寫 VB 程式，將來若有問題，就可以知道是『**自行撰寫的程式**』錯誤。

在硬體部分，有部分考生因過於緊張，僅是將**IC 裝相反**而已，或者將**74LS244 與 74LS273 裝錯位置**。接著從外觀大略檢查是否有**短路現象**，尤其是交叉的地方，有時是因為**兩點之間有錫渣而造成短路**，還有檢查**所有元件腳是否有焊接**，整個電路僅 ATMEGA8 有**10 隻腳未使用**，這些都可從外觀先行判斷。

茲將一般考生常犯的硬體錯誤狀況整理如下，欲量測兩點之間是否導通（短路），可以利用**三用電表的歐姆檔進行測量**，而且記得是要從**零件面（正面）進行測量才對**，另外在測量電源之間是否**短路**，三顆 IC 其實可不用拆下，只要有**超過 1.5KΩ (R19) 阻抗即可**。

Q1：介面卡的 Vcc 與 GND 發生短路？

A1：先檢查 OK 單心線外觀，檢查是否 OK 單心線外層有被不小心燙到，而發生短路狀況，尤其是和 Vcc 與 GND 有連接經過的地方與 OK 單心線有交叉的地方都要特別留意。

Q2：LED1~LED8 其中有一顆紅色 LED 不亮，其他功能皆正常？

A2：因為 LED1~LED8 是由 74LS273 所控制，若部分 LED 正常，表示該 IC 的電源 Vcc 與 Pin1 ($\overline{\text{RST}}$ ，Reset 重置) 已連接、Pin11 (CLK) 也正常連接到 ATMEGA8 的 RCLK(pin27)，檢查方式應先參考電路圖，然後再倒追原來訊號的路徑，例如若 LED4 一直不亮，會發現 LED4 先經過 R4，再接到 74LS273 IC Pin9 的 Q3，而 Q3 的源頭為 Pin8 的 D3，而 Pin8 的 D3 接到 ATMEGA8 的 Pin17(AD3)或 74LS244 的 pin8(A4)，然後分別檢查剛才所經過的路徑是否都導通，若真有不通的狀況，直接再焊一條單心線或許會比較快。

Q3：LED9~LED16 其中有一顆綠色 LED 不亮，其他功能皆正常？

A3：因為 LED9~LED16 是由 74LS244 所控制，若部分 LED 正常，表示該 IC 的電源 Vcc 與 Pin1 ($\overline{\text{OE1}}$)、Pin19 ($\overline{\text{OE2}}$) 都正常連接到 ATMEGA8 的 pin28，檢查方式應先參考電路圖，然後倒追原來訊號的路徑，例如若 LED9 一直不亮，發現 LED9 會先經過 R9，再接 74LS244 IC Pin18 的 Y1，而 Y1 的源頭為 Pin2 的 A1，而 Pin2 的 A1 是接到 ATMEGA8 的 Pin14(AD0)或 74LS273 的 pin3(D0)，然後分別檢查剛才所經過的路徑是否都有導通，若真有不通的狀況，直接再焊一條單心線或許會比較快。

Q4：LED1~LED16 其中有任意連續兩顆 LED 一起全亮，其他功能皆正常？

A4：依照上述方式倒追檢查這兩顆 LED 原來路徑，檢查中間是否有地方發生短路，一般而言都是 OK 單心線裸露的**長度太長**，或是電烙鐵頭太粗，不小心將鄰近的 OK 單心線**外皮燒破而短路**，尤其是 OK 單心線交叉的地方最容易發生短路。

Q5：LED1~LED16 亮燈順序亂跳？

A5：一般都是 IC 的腳位弄錯，建議考生可先利用光碟內附的軟體，模擬練習到完全沒有任何錯誤再開始進行實際焊接，記得要將所有腳位先熟記。

Q6：僅有紅色 LED1~LED8 完全沒有亮燈反應，其他功能皆正常？

A6：可能是控制的 74LS273 IC 插錯方向，或是該 IC 電源或 Pin1 沒有接到 Vcc，或是 74LS273 IC 的 Pin11 (CLK) 並沒有接到 ATMEGA8 的 Pin27 的問題，因為若沒有正緣的觸發就不會有輸出訊號。

Q7：僅有綠色 LED9~LED16 完全沒有亮燈反應，其他功能皆正常？

A7：可能是控制的 74LS244 IC 插錯方向，或是該 IC 電源沒有接通，或是 74LS244 IC 的 Pin1 ($\overline{OE1}$) 與 Pin19 ($\overline{OE2}$) 致能訊號並沒有接到 ATMEGA8 的 Pin28 問題。

Q8：LED1~LED16 完全沒有亮燈反應？

A8：可能是 ATMEGA8 電路有錯，無法正常將串列資料(D+、D-)轉成並列資料(D7~D0)，也可能是部分 Vcc 與 GND 未確實連接，另外再**確認 IC 是否裝相反**，或是因為將 74LS244 與 74LS273 裝相反，記得 74LS244 應裝在**左側**，以控制綠色 LED。

Q9：僅兩側紅色 LED1 或綠色 LED16 一直沒有亮燈反應，其他功能皆正常？

A9：請檢查 LED1 或 LED16 的 GND 是否連接 (*不同電路板外圍已導通的配置並不同)。

Q10：僅某幾顆 LED 一直「恆亮」，其他功能皆正常？

A10：因為輸入端 **TTL 浮接(空焊)視為"1"**，所以會一直亮。請檢查對應的輸入端訊號接腳(*重新加熱焊接即可)，例如 LED2 有問題，請檢查 74LS273 的 Pin4(參閱電路圖)。

Q11：若紅色與綠色 LED 燈一直恆亮，且是「同一相對位置恆亮」？

A11：因為 TTL 浮接(空焊)視為"1"，且錯在同一地方，表示 ATMEGA8 的 AD0~7 空接。

Q12：若紅色 LED 燈一直跟著綠色 LED 燈一起亮，當綠燈亮時紅燈又跟著亮？

A12：表示 ATMEGA8 的 Pin27、28 中間發生短路，所以同時被觸發或致能。

Q13：僅某顆 LED 一直不亮，其他功能皆正常？

A13：直接用三用電表歐姆檔測量電路板上的 LED 不亮，但將 LED 拔下再測量又變好的，那是因為電路板 LED 兩端接腳短路，因此，若發生此狀況，不要急著又換一顆 LED，應先確定原因再決定是否更換，當然有可能真的是 LED 燒掉。(※此為真實案例)

Q14：快速判斷 R1~16 和 LED1~16 中間是否已正確連接？

A14：利用三用電表的 X1 或 X10 歐姆檔，先將帶『負電』的紅棒接於 IC 的 GND 端，然後再將三用電表帶『正電』的黑棒分別接在 R1~16，則對應的 LED 應點亮，這表示底下虛框內的 R1~16 和 LED1~16 連接都已正確，實際測試方式如下圖 2-40 所示，萬一對應的 LED 不亮，則從焊接面直接測量 LED 好壞，或接線不良造成。

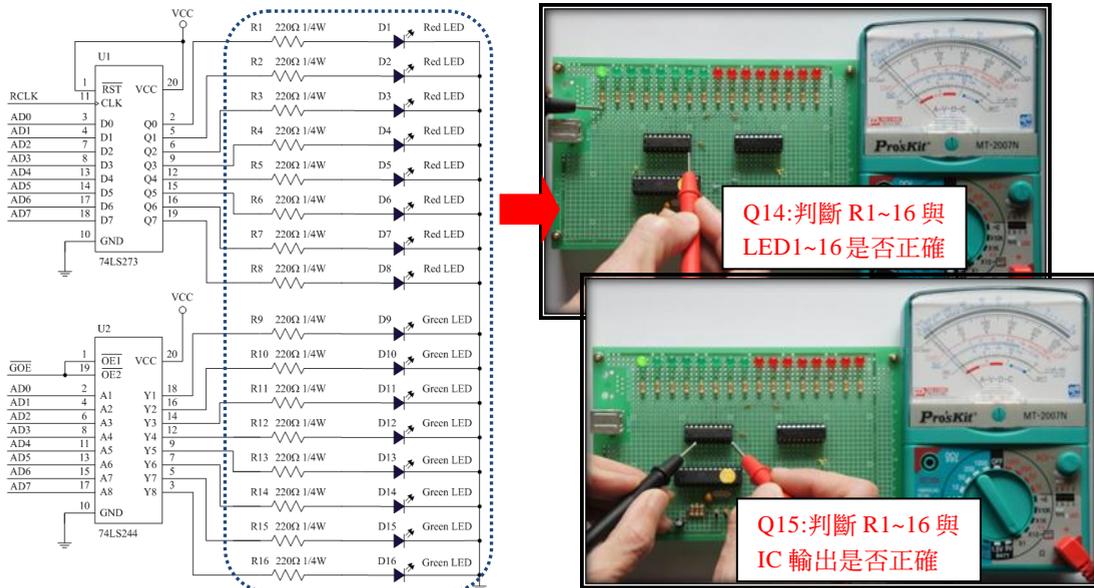


圖 2-40 快速檢查 LED1~LED16 → R1~R16 → IC 輸出腳位的技巧

Q15：快速判斷 R1~16 和兩顆 IC 的資料輸出是否已正確連接？

A15：三用電表紅棒仍然接於 IC 的 GND 端，但將帶『正電』的黑棒改接於 74LS273 的 Pin2、5、6、9、12、15、16、19 與 74LS244 的 Pin18、16、14、12、9、7、5、3，則對應的 LED 應點亮，這表示兩顆 IC 的輸出資料到 LED 之間的連接都正確完成。

Q16：USB 一直偵測不到介面卡裝置？

A16：先檢查 USB 四支腳位是否有短路發生，當接上 USB 後出現『USB 裝置無法辨識』錯誤訊息，在裝置管理員的通用序列匯流排控制器會出現 Unknown Device 未識別的裝置，主要發生原因是 ATMEGA8-16PU 線路沒有完全接好或燒錄內容有錯。
※由於考前已經測試確認韌體，但有可能在製作過程因短路而燒毀，當確認自己的線路『一定』無誤後，要求更換 IC 會被扣分。

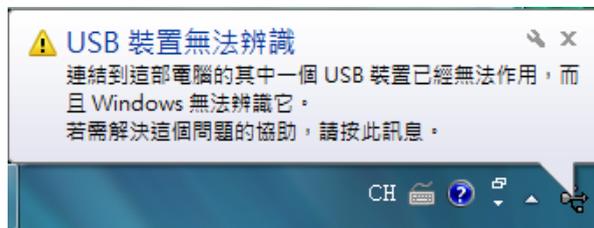


圖 2-41 USB 裝置無法辨識的錯誤訊息

Q17：電腦右下角視窗會一直出現『集線器連接埠發生電源爆衝』訊息？

A17：肯定是考生未量測是否短路就直接接上電腦 USB 測試，因為當發生短路時，則操作的電流肯定超過 500mA(USB 規格)，就會出現『集線器連接埠發生電源爆衝』訊息，倘若電腦設計不是很好，將有可能造成電腦當機或損毀，所以**每當有修正或重新焊接電路之後，務必重新量測是否短路方可再接上電腦 USB**，所以 Vcc 與 GND 有連接經過的地方與 OK 單心線有交叉的地方都要特別留意是否短路。



- A. 請**務必攜帶自己的三用電表**，勿因為過分有把握，認為一次就成功就不用帶，雖然考場有提供，但有人不會使用數位式三用電表，還是自己的電表最熟悉。
- B. 平常老師雖然要求焊接要『快』，那是希望可以爭取更多考試時間，實際考試應求平穩扎實，不要刻意凸顯自己速度，**每個焊點應確實輕拉一下檢查是否空焊較保險**。
- C. 若發生 IC 燒毀，務必拆掉 IC 量測 Vcc 與 GND 是否短路，問題排除後再找老師重拿 IC，**每顆 IC 扣 10 分**，當問題沒解決之前，若 IC 就先放上去，當然會一直燒毀。
- D. 若是 VB 程式問題一直找不到問題所在，不妨**重新啟動電腦**，全部再輸入一次，包含模組也是！倘若還是一直檢查不出來，可很有禮貌舉手請求再借測試板或更換電腦進行測試喔！千萬不要坐以待斃喔！
- E. 根據這幾年輔導經驗，第一站沒過的考生，反而大都是最有把握的人，因為平常這都做很順，幾乎一次就成功，就因為平時都很少遇到問題，考試一緊張反而亂了手腳，甚至連三用電表都不知怎樣找問題所在。**平時練習可靠檢修別人找不出問題的板子當作練習的機會**，真正厲害的人，不是做最快，而是會主動協助修理電路的人。
- F. 在程式軟體的部分，本書所附的範例已盡量精簡，方便考生熟記，但是由於有人考試時過於緊張，而**忘了將 Visual Basic 程式先儲存就開始執行**，以至於找不到對應的 DLL 檔，務必要特別小心。

