

Apple 界面實驗

這節實驗的目的是在讓您親自動手使用前面幾章所討論過的鎖住輸出口以及三態輸入口電路。您會發現，這些實驗以簡單的 SN7400 系列元件，將資料傳入與傳出 Apple。

6 - 1 實驗簡介

這一章必須將電路裝在麵包板上，完整的零件單摘列於附錄 B 中。本書假設您已有將簡單邏輯電路裝在麵包板上的經驗，並假設您熟悉基本的插麵包板技巧。實驗中將用到一些監聽邏輯狀態與產生邏輯狀態的輔助功能。通常，我們都可以燈泡或 LED 顯示邏輯 1（亮）與邏輯 0（暗），以開關產生邏輯準位，並以除跳訊脈衝器（簡稱脈衝器）產生邏輯準位轉態時無雜訊的邏輯準位。附錄中附有一點這些電路的簡單電路圖。若您不願意製作這些電路，則您可個別將之裝在麵包板上。一般而言，這本書的絕大部份實驗都可以簡單的電路完成。

我們舉了一個以解碼器電路作設備選取的實驗例子。雖然解碼器方法有許多，但我們認為就一個實驗即可闡明基本的原理。若您對其它的電路還有興趣，則請您參考諸如“8085A 手冊”以及“6502 程式設計與界面實驗”等有關書籍。事實上，各種計算機間之記憶器與輸入 / 輸出設備選取則是大同小異的。就絕大多數界面電路而言，界面麵包板上所用的解碼器電路就已經相當夠用了。

雖然這本書以一相當低的層次處理 Apple 界面，但您可能還希望了解一些其它重要的界面主題。這些主題在“TRS-80 界面實驗，第二冊”幾乎都有談到。有興趣的讀者希望您參考這本書。這本書上所討論的東西都很一般化，因此，其極其容易應用至 Apple 計算機系統上。這些主題包括：高電流、高電壓負載推動、數位至類比以及類比至數位轉換器，實用資料處理（平滑、濾波、平均、等等），串聯通信，與遙端控制。

圖 6-1 所示即為本章之實驗所使用之 Apple 計算機與麵包板的照片。麵包板與 Apple 計算機之間以如圖 6-2 所示之 40 號電纜連在一起。這個電纜在第 5 章時已介紹過。在將界面麵包板接至 Apple 時，特別記住，電纜的取向一定要對。電纜恒背對將界面連接至 Apple 之調適板的零件面。在電纜之界面一麵包板端，電纜必須推入 40 支接腳，以使電纜面向下或背向印刷電路板。假若電纜沒接好，則 Apple 將在螢幕上顯示出一堆雜亂文字，而不是像電源一打開時所顯示的 Apple II 標幟。這種錯接只要不要維持太久，Apple 或界面似乎不致造成任何永遠的傷害。

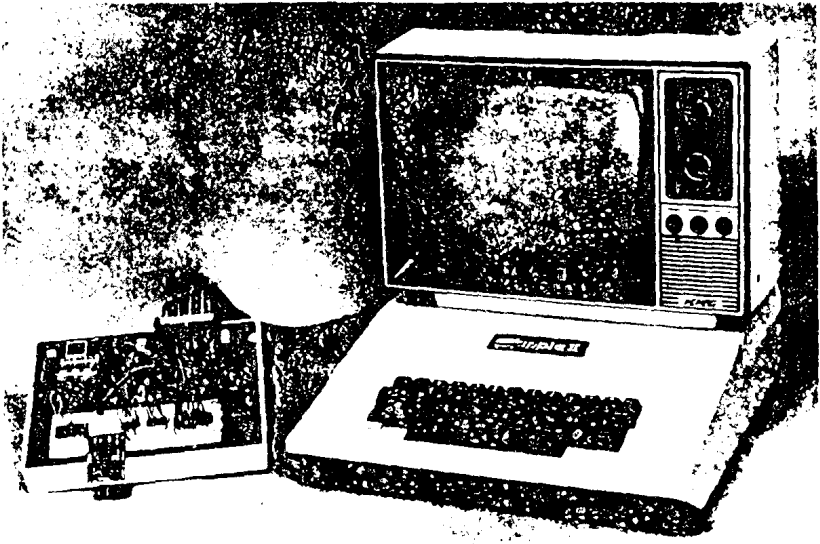


圖 6-1 實驗中所用的 Apple 計算機與麵包板。

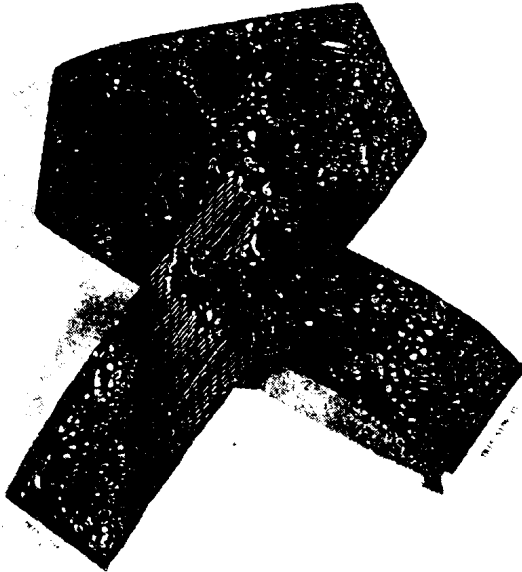


圖 6-2 界面電纜(注意接點方向在平軸電纜同一邊)。

有部份實驗會植基於或使用前面實驗所發展出的電路或程式。除非有教您這樣做，否則，請勿隨意將計算機的電源關掉。不然的話，您就得花費許多時間重新取入程式或重新裝配界面電路。有些實驗後面我們會加上諸如此類的提示語。

相信絕大多數讀者都會循序地做這些實驗，在這種情況下，實驗參考到前面的地方就沒什麼問題。倘若您是跳著做，有時您就會覺得有些東西不甚了解。爲了幫助您了解這些界面電路，在本章末了，我們特別又將最重要的輸入口、輸出口、與控制電路畫出於圖 6-27 中。爲了能隨手參閱，您可將這張圖複印一份，或直接將這幾頁拆下，放在手邊。除非特別聲明，否則，本章之絕大多數實驗所用的都是這張圖中所示的基本電路。必要時，您亦可以這些電路做成通用的輸入與輸出口。

假若您是老師，而且希望以這本書作 Apple 計算機的實驗教本，則您會發現，書中的程式很容易即可存在卡帶上。將程式存在卡帶後，學生即可隨手獲得這些程式，而不必浪費時間爲程式除錯。若您選擇使用卡帶，一定要使用高品質的卡帶，而且程式一旦錄在磁帶上後，卡帶背緣之“寫入保護”鈕就應拿掉。因爲，這樣可防止學生因意外寫入磁帶而破壞原有的程式。

學生或許發現自己亦應擁有一份卡帶程式，以便能隨時獲得實驗解與其它程式，不管是和其它人作交換或留作爾後參考。

本章的實驗已分成兩組，雖然沒有明顯的章節區分，但由許多小標題以及其它之註解，您將可發現這種分野。其中

，前 11 個實驗所做的都是基本界面與程式設計技術的探討。這幾個實驗可為初步的計算機界面或計算機電子學課程的實驗部份，提供一個基礎。

最後幾個實驗則對幾個更高深的課題作實驗的探討，其同時亦提供了可用以更充實前面之基本實驗的研究計劃。當然，所有這些實驗還是都可以做。

6-2 實驗

實驗 1

邏輯探測器之應用

目的

這個實驗的目的是要告訴您如何以裝在麵包板上的邏輯探測器電路，測知邏輯準位與脈衝。

討論

假設您所用的都是裝在麵包板上的邏輯探測器，當然，其它的邏輯探測器亦同。本實驗的步驟可幫助您熟悉麵包板

與現有可用的信號。

第 1 步

將 Apple 計算機接至視頻顯視器，且經 40 條導線之電纜接至界面麵包板。這個接線在實驗簡介時已說明過。

打開 Apple 計算機與麵包板的電源。計算機螢幕上應出現“APPLE II”的字樣以及一閃爍的方形光點（square cursor）。萬一沒有，則先關掉電源，仔細檢查一下接線，然後再重新開機。記得，40 號電纜一定要安穩地插入界面麵包板之接腳內以及在接至 Apple 之板緣上。電纜的面向也應檢查一下看是否正確。若您自己一個人無法找出問題所在，找個人來幫忙。

第 2 步

電源加至麵包板後，以一條跳線將 PROBE（探測器）插座之其中之一邏輯探測器輸入接腳 P，與電源插座處之其中之一 + 5 伏特電源接腳接起。看看邏輯探測器指示器會有什麼反應？

紅色 LED 亮，代表出現邏輯 1 狀態。

接著將跳線由 + 5 伏特之電源接腳移至同一插座之其中之一接地接腳。看看結果又如何？

綠色 LED 發亮，代表探測器電路之輸入出現邏輯 0 狀態。您可能已發現，在您將跳線接至 + 5 伏特或接地利那，脈衝測知（黃色）的 LED 會閃一下。這個閃亮即表示探測器已偵測到邏輯準位有所改變。不論遇到邏輯 1 變邏輯 0 或邏輯 0 變邏輯 1，黃色的 LED 都會閃亮一下。這個 LED 在測知脈衝與邏輯轉態上特別好用。

將探測器輸入接至 IC-19 的位址線 A0。接好之後，您看到什麼呢？所有 LED 都發亮，但亮度或許有所不同。這主要是因為 6502 微處理器晶片現在正在執行許多許多的組合語言指令並監聽著 ROM，因而，正以位址巴士選取許多不同記憶位置。將邏輯探測器之測試線再移至另外其它位址巴士線 A1, A2, A3……等等看看，您會發覺，在這些接脚上所測得的結果亦類似。事實上，不論那一條位址線，由於 6502 微處理器均不斷有新的位址資訊送出來，因此，LED 應迅速地不停閃亮著。

第 3 步

您或許希望以邏輯探測器測試麵包板上的一些其它點。譬如，資料巴士線與控制信號都可輕易地測試。不過，你要謹記，邏輯探測器僅接受麵包板及實驗中所使用之標準 TTL 所輸出的邏輯準位。因此，切勿以之測量這些準位以外的任

何東西。譬如，假若您將探測器接至一超出 0 至 5 伏特範圍的電壓，則探測器將會燒毀。

第 4 步

在使用探測器的過程中，您會發現，LED 發亮的情形有許多種組合情況。譬如，您可能看到紅色與黃色 LED 發亮，而綠色 LED 未亮。您知道這是什麼意思嗎？

這個意思是，探測器測到了一個脈衝，而且正被測試之電路的正常邏輯準位是邏輯 1。綠色 LED 很快地亮一下（您看不到），顯示邏輯 0 脈衝疾逝的出現。脈衝測知電路展延了脈衝，並使黃色 LED 發亮，致您可以“看到”探測器“捕住”了一個脈衝。

您或許又可看到綠色與黃色 LED 發亮，而紅色 LED 不亮。這又是什麼意思？

代表邏輯 0 準位，以及一短暫邏輯 1 脈衝。

亦可能所有 LED 都發亮。這種情形就表示邏輯探測器的輸入，不斷地在邏輯 1 與邏輯 0 準位間迅速變化。

在後面一些實驗中，我們將以邏輯探測器檢查電路之輸

出，並且測知邏輯狀態與脈衝。這時候我們會說“……以探測器檢查……”或“……以邏輯探測器測量……”。這個意思就是教你將邏輯探測器連接到受測的電路上，以便能“看出”結果。

關掉計算機之電源。（爾後將簡稱關機）。

實驗 2

設備位址解碼器之應用

目的

這個實驗使您能探知界面麵包板印刷電路板上之設備位址解碼電路的用法，這個解碼器由於在爾後所有的實驗中都會用得到，因此，您必須徹底了解其用法。

討論

這個實驗以 A15 ~ A0 之位址位元辨認輸入 / 輸出設備所使用的特定位址。我們將以位址開關設定某一範圍內的位址，並以邏輯探測器檢查解碼器電路的動作。實驗亦會用到一個 SN7402 NOR 閘 IC。

積體電路(IC)之接腳圖(圖 6-3)

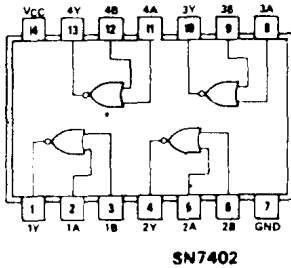


圖 6-3 SN7402 NOR 閘之接腳圖

第 1 步

現在麵包板上應未接任何電路。若有電路，請將之除去。在這個實驗內，界面之解碼器部份會用到整個 16 位元的位址巴士。記得，下半 (LO) 位址設定開關 (IC-6) 最下面一個開關必須置於“M”或“ON”的位置。

第 2 步

將 A15 ~ A14 之所有位址位元的浮降開關置於邏輯 1 的位置。記得，切勿動了“M”開關。您知道 SN74154 解碼器將解碼的是那些位址嗎？這個區間的位址有那些在 ADDRESS 輸出插座上有的呢？參考一下 5-4 圖。

區間內 65520 至 65535 之位址將為 4 線對 16 線的解碼器 (SN74154) 所解碼。由於解碼器僅提供了“下面”(較低) 八個位址，因此，唯有 65520 至 65527 之位址可得。

第 3 步

打開計算機之電源。若原來您正在執行程式，則請按 RESET 鍵。以邏輯探測器測試 ADDRESS 插座的八個位址輸出。其中有一個解碼器輸出動作(有脈衝)嗎？由於現在根本不執行程式，因此，這是您所想像的嗎？

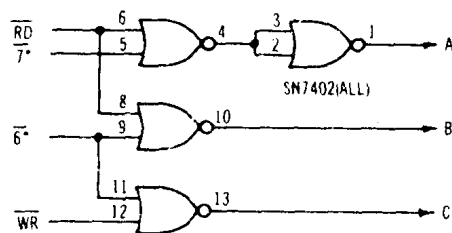
有兩個輸出應該動作，0 與 4，分別對應於位址 65520 與 65524。計算機未在執行 BASIC 程式時，其將執行一個一直監聽著鍵盤(看有無按鍵發生)的組合語言程式。因此，位址解碼電路應一直都在解碼位址。

第 4 步

將圖 6 - 4 所示的電路接好。記得，電源接腳，14 腳，一定要接至 + 5 伏特，而且接地接腳，7 腳，要接至電源

APPLE 界面實驗

接地。SN7402 的接腳圖請參閱圖 6 - 3。SN7402 您亦可換成 SN74LS02。此時，輸出 A，B，及 C 暫不接至任何電路。



* 解碼器插座位置

圖 6 - 4 功能脈衝產生電路

第 5 步

將 A15 ~ A4 位址位元的開關設定成位址 49312，寫成二進制即 11000000 10100000₂。記得，最低次四位元應忽略。當位址開關如此設定時，會產生那些位址呢？

49312 至 49327 之位址將被解碼，但只有 49312 至 49319 之位址可用。

第 6 步

將下列程式打入計算機，並教計算機開始執行：

```
10 A = PEEK(49318)
20 GOTO 10
```

以邏輯探測器監聽著解碼器輸出，並記下您所看到的一切：

您應看到“6”號輸出動作，而且其它一些輸出亦可能動作。

接著，監聽邏輯閘輸出 A，B，與 C，並記下邏輯探測器所測知的結果：

	邏輯 0	邏輯 1	脈衝
A			
B			
C			

這就是您所預料的嗎？您能做個解釋嗎？

是的，這就是我們所預期的。因為，程式中的輸入命令（

PEEK) 指明位址 49318 的設備作為輸入設備，而且經解碼的位址在解碼器的“6”號輸出上。因此，唯有“B”輸出該動作。程式中未再指及任何其它輸入設備，亦未指及任何輸出設備。

第 7 步

將程式 10 號一列中所含之位址改成 49325，使 10 號一列變成

```
10  A = PEEK ( 49325 )
```

然後再度執行程式且測試邏輯閘輸出 A，B，與 C。請問有那些輸出動作，表示脈衝出現嗎？何故？

不應有任何輸出動作，因為，設備位址 49325 並未裝在電路上。此外，位址 49325 在麵包板上亦沒有。在 49312 至 49327 的位址區間內，只有 49312 至 49319 幾個位址在 ADDRESS 插座上才有。

第 8 步

將 10 號一列再改成

```
10 A=PEEK(49318):B=PEEK(49319)
```

再執行程式時，脈衝出現於電路中之何處呢？

您應該發現輸出 A 與 B 動作。輸出 C 不動作，因為，其為一輸出控制脈衝，而程式中根本無輸出指令（POKE）。

第 9 步

再將程式修改。改變 10 號一列，使其能控制輸出設備 49318。此時，10 號一列的述句應如：

```
10 POKE 49318,0
```

資料值可使用 0 至 255（含）之間的任何值。執行程式，並測試輸出 A，B，及 C。您想那一個輸出應動作呢？事實亦然嗎？

由於 POKE 指令為輸出指令，且位址 49318 對應於解碼器之

“6”號輸出，故輸出C動作。看到B輸出亦動作您可能感覺得很驚訝！Apple之BASIC解釋程式在執行POKE指令時，計算機系統都會做一個“先讀取再寫入”（read-before-write）的動作。在設計界面電路的過程中記住這一點。

第10步

您能重新配組位址解碼部份之開關，使解碼器產生50944至50951之間的位址嗎？您如何做呢？這幾個位址真的出來了嗎？

是的，可以改變開關設定，使解碼器產生這些位址。首先，將位址換成二進制： $50944 = 11000111\ 00000000$ 。接著，重新設定A15～A8以及A7～A4的開關。這時候，解碼器之“6”與“7”輸出所對應的位址各為何呢？使用PEEK指令，以這個實驗一直在用的程式作個測試。您應可在邏輯閘之A與B輸出看到脈衝。

一旦測試過後，記得將位址開關還原成位址1100000010100000₂的狀態。

勿將電路拆掉，其還有用。不過，程式將不用了，因此，您可將麵包板與計算機的電源關掉。

實驗 3

使用設備選取脈衝

目的

在這個實驗您可看到以設備選取脈衝控制外部設備的用法。PEEK 與 POKE命令雖然通常均用以控制資訊流通，但事實上其亦可用以產生控制外部設備所用的脈衝。

討論

這個實驗以設備選取脈衝啓動或關閉一個簡單的設備。我們以邏輯探測器作“設備”，而且以兩個軟體產生的脈衝控制一個簡單的正反器。

IC接腳圖(圖 6-5)

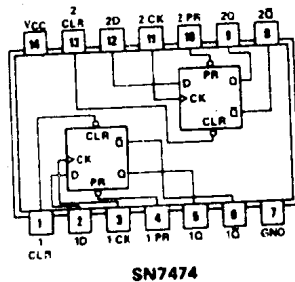


圖 6-5 SN7402與SN7474晶片之接腳圖