

APPLE II 實用手冊

上冊

蕭金林編著



目 錄



第一章 您，蘋果和這本書.....	1
需要什麼樣的系統.....	2
本書之內容.....	3
如何從本書獲得最大效果.....	6
第二章 文字幕顯示.....	9
標準文字幕格式.....	9
游標運作規則.....	10
以 PRINT 敘述控制游標.....	12
以 TAB 指令設定游標位置.....	18
運用游標位置暫存器.....	27
其他有關游標之控制.....	34
第三章 文字幕格式之變化.....	43
字形之變化.....	43
文字窗之變化.....	52

第四章 影像記憶之內容	71
文字顯示記憶區之組成	71
顯示畫面之字形碼	75
借助於系統監督程式	80
建立及使用字符串段落	86
文字畫面第二頁之操作	93
第五章 鍵盤輸入功能	101
以 INPUT 指令提示訊息	102
以 INPUT 指令控制程式執行之流程	106
以 PEEK 指令檢測按鍵	111
程式之單鍵控制	116
為執行控制而作按鍵解碼	125
第六章 低解度繪圖功能	135
操作之基本原理	135
在螢幕上顯示色彩	145
畫面型態之轉換	162
第七章 高解度繪圖功能	171
LOMEM和HIMEM之預估	172
啓動高解度繪圖操作	179
高解度畫面之色彩和格式	184
操作高解度繪圖功能	185
高解度畫面格式之變化	202
高解度繪圖造型表	205
高解度畫面影像記憶位址	226

第八章 機器語言和BASIC程式併用	239
一些有用的機器語言指令	240
輸入及執行機器語言副常式	249
叫用系統監督程式	254
將變數傳遞至機器語言常式	269
自機器語言副常式傳出變數	276
第九章 記憶範圍規劃	279
RAM記憶位址:\$0000~\$0BFF	280
較高的RAM記憶位址:\$0C00~\$BFFF	293
I/O位址:\$C000~\$CFFF	301
主要的ROM位址:\$D000~\$FFFF	314
第十章 迷你組合語言程式設計	317
組合語言程式設計要纜	319
使用迷你組合語言翻譯器	322
建立組合語言程式	328
藉系統監督程式載入程式	333
利用BRK指令除錯	335
附錄A 數值轉換	337
B 文字顯示之字形碼	349
C 文字幕 / 低解度繪圖之影像記憶位址	353
D 按鍵之字碼	355
E 低解度繪圖之色碼	359
F 高解度繪圖之影像位址	375
G 6502 指令表	387



第一章

您，蘋果和這本書



或許您還記得當第一次接通 Apple 電源時的感覺吧！對於大部份的人來說，那股興奮勁兒，將是畢生難忘。

能擁有一部自用的個人電腦，最大的好處是，從此您將踏上電腦之旅。而在這旅程中，永遠有許多嶄新的事物，等待着您去嘗試和學習，並且在這嘗試，學習和改進的過程中所帶來的樂趣，亦如初次開機時的喜悅，將使您永難忘懷。

當然，有時難免會因事與願違，而讓您恨不得將整部 Apple 丟出窗外，像這類的答非所問，或者結果無法如預期般地執行，更加深了操作者的挫敗感。不過，類似這般情景，對於每一個電腦使用者或程式設計者而言，不論其經驗之深淺，或是電腦系統之完美與否，總是難以避免的。

然而，個人電腦之操作，乃是依靠持續不斷地經驗累積。在學習過程中，您必須不斷地吸收新的知識，以作為日後應用的基礎，如此不斷地成長，學習的脚步永無休止，而樂趣也將無窮無盡。

欲維持這種對電腦喜好的熱度，最好的方法，莫過於挖掘該電腦的奧秘。如果一味地在舊事物堆中打轉，則不論該電腦有多能幹，遲早總會覺得乏味和厭倦，因此，唯有不斷地學習，才能維持盎然的生機。

本書之主要目的，即在幫助您從操作 Apple II 中，獲得更多的樂趣，並藉此促使您踏上不斷學習的旅程。

至於本書又將如何幫助您學習呢？基本而言，本書將對 Apple II 許多頗具效用，然而却被一般使用手冊所忽略的操作細節，作詳細之描述，如能熟悉這些操作細節，將使您的 Apple 更能發揮多元性的功效。所有這些特性，本書將以實例或示範，加以仔細闡明，以便您能了解，並引用到您自己設計的電腦程式中。

需要什麼樣的系統

由於目前的 Apple 電腦，已發展成極廣泛的應用形態，並各具有些微之差異，因此幾乎很難有某一著述，得以涵蓋並適用於這些差異。所以有必要在開始之前，先行界定使用之範疇，以便合乎大部份讀者之需要，並且期望其他之 Apple 電腦使用者，能自本書中擷取適用之資料，而有助於其操作實務。

所有本書中之程式和範例，皆可在具有 48K RAM 和 ROM 形式的整數 (Integer) BASIC 之 Apple II 上操作。不過，如果 RAM 之記憶容量少於 48K，即使在 16K 的情形下，除了必須犧牲高解析度作圖的第二頁畫面之外，也仍然能照常執行。

這些討論之內容，也同樣地適用於卡帶或磁片形式的 Apple 系統

。偶而，磁碟作業系統（DOS）會造成一些小小的困擾，尤其是當（DOS）啟動時，佔用了部份供其他用途的RAM記憶範圍，因此，磁碟作業系統操作時，使用者必須特別留意並避免因之造成的困擾。

如果您備有一部列表機，當然有助於對討論內容之理解，然而這並非絕對必要之設備。

雖然有關於彩色圖像之討論，若能配合使用彩色電視機或監督程式，更能相得益彰，不過，黑白螢幕也仍可勝任之。此外，本書中所使用之彩色名稱，將沿用Apple公司所制定之名稱，而您對這些色彩名詞之詮釋若有不同，也請一併留意。

本書之內容

大多數的人們，當他首次接觸到家用電腦時，即使僅試跑一些使用手冊上所列舉之BASIC程式，或其他套裝軟體——通常是一些遊戲程式，也能從中獲得少許之樂趣。

不過我們却要坦白承認：執行這些BASIC程式，或者卡帶和磁片的現成軟體，所帶來的樂趣乃是極為有限。當然，您也可以添購其他更精緻而且高價的既成軟體，或者引用書刊上所列舉之BASIC及機器語言程式，即使這些程式能合乎您的需求（很不幸的，大部份仍難以如願以償），但往往也已經是過時的產品，而少有裨益。

欲挽救這種熱忱消退的危機，最好的辦法，就是自行設計適用的程式。在設計的過程中，若能確實掌握需求之目標，將可提高您操作電腦的興緻。此外，學習並使用BASIC語言程式設計，也能使工作的成效持久不墜。

固然，以 BASIC 進程式設計，可提供給大多數的用者莫大的便利，不過，遲早您總會發覺 BASIC 語言的限制，而難以滿足。有經驗的 Apple 程式設計者，都會因 BASIC 語言本身的束縛太多，而感到難以發揮，往往在某些需求情況下，發覺 BASIC 無法充分而有效地達成任務。

例如，原本生動的圖像畫面，若以 BASIC 處理，則因某些指令的執行時間太長，而變得呆滯遲緩。又如字串變數中，若超過 250 個字長，在 BASIC 作業中，也會造成困擾。諸如此類的問題，則是起源於 BASIC 和硬體系統間溝通的障礙所致。

欲克服這些障礙，解決的方法之一是，閱讀有關的刊物，從中發掘素材，以便突破知識的瓶頸。或許您曾經這樣試過，但仍難免大失所望，這並非文章內容的錯誤所致；事實上，這一類的題材仍是有所裨益。只不過，大多數的刊物內容，所講述的有關 Apple 的操作技巧，涉及的層面頗為狹小，作者所提供的僅是針對某一特定情況的解決之道，而使得您在閱讀時，忽略了整體性的原則。

通常，一本書或一篇文章的精華所在，往往隱藏於一些程式列表或說明之內文中，如果只是概略的瀏覽，也許會誤認它只適於解決特定的難題而已。事實上，這些範例背後所隱藏的構想和思考方式，才是真正的無價之寶。

例如，講述如何以特定按鍵，操縱畫面上某一彩色點移動的技巧，也許並不適用於您；但是其中所涵蓋的檢鍵程序，以及繪出移動的彩色點這兩種操作，如果您能確實掌握它的原則，並加以引用，却是妙用無窮。

本書之主旨，即在講述這些重要之原則。您在閱讀時，不必費心

地從程式列表中挖掘所隱藏的寶藏，所有這些精華，都會詳盡的交待清楚。

雖然本書中列舉了許多程式，但其目的，僅用於輔助說明操作之原則。因此，程式本身，並沒有太大的意義；事實上，這些程式都是點到為止，談不上精緻，也不具備擴展的用途。正由於這種直截了當的特性，使得其中之主要關鍵能夠一目了然。

至於其他非教育性的程式，則由於顧慮太多的細微枝節，以致於喧賓奪主，反而掩飾了重要原則，這就不是本書編排的宗旨了。從這本書裏，您可以發掘很多有益於電腦操作的構想。

有一點必須特別聲明的是，本書並不急於推薦機器語言程式之設計。通常，當您渴望走出 BASIC 的範疇時，總會有人提醒您下一步該向機器語言進軍。

事實上這並不一定正確。電腦語言程式設計之能力，乃是循序漸進的；而學習的脚步，也應該是平穩踏實的。若是冒然地直接從 BASIC 躍進純粹機器語言的世界，則顯得太過突然而且生硬。更糟的是，如果未曾有過訓練和學習的機會，竟摒棄 BASIC 而走向機器語言，將更是難以想像。畢竟，這兩者在思考方式和設計技巧上，都有着顯著的差異。

因此，機器語言程式之設計，並非本書編寫的主要宗旨。一旦您從本書裏，了解到 Apple II 內部的許多操作特性，將可因而激起您對 BASIC 程式的興趣和信心。您也可以從 BASIC 語言裏，引用並執行系統常式，而突破一般程式設計的障礙。

這也正是本書編寫之緣由。

只要您能從 BASIC 語言的觀點，透視整個系統之結構，也正是

可以開始進入機器語言和組合語言的最佳時機。就算如此，也仍須將 BASIC 語言與之結合。這情形好比是：只要緊緊掌握住 BASIC 這條纜繩，您儘管放心大膽地涉入機器語言的急流之中。

在本書的後面章節裏，討論了一些機器語言及組合語言的程式設計。到那時候，由於您已熟習了 Apple 內部操作之細節，所以對於機器語言之設計，必能勝任愉快。

總而言之，如果您對 Apple II 個人電腦已開始覺得厭倦，本書將是極佳的一帖興奮劑。

如何從本書獲得最大效果

雖然本書中之部份內容，和其他書籍所載類似，但絕不是一般的使用者手冊。全書採取循序漸進的方式，因此，只要您按部就班地詳細研讀，必能獲致最大的效果。反之，如僅蜻蜓點水般地翻閱，則將事倍功半。

由於本書之前段內容，採取十進位的數據處理方式，所以有關之表格（Table）均使用十進位數值和位址（address），雖然這有點繁雜，但對於尚不熟悉十六進位表達方式的讀者而言，却屬必要。

當論及機器語言、組合語言之程式設計時，各項表格將附加十六進位值。對於不曾使用過十六進位者，附錄 A 之內容，將有助於數值轉換之學習。

如果您隨手翻閱書後的附錄，必可發現一些極為有用的對照表，其中每一數值和地址碼，皆以十六進位和十進位表達。

本書乃是電腦使用者自修的最佳指引。從閱讀中，您必可獲得許

多有助於學習程式設計的技巧和知識。在此，我們提出一項重要觀念，作為本書之寫照：

“假如您每天送一條魚給挨餓的人，或許可讓他多撐過一天；但如傳授給他釣魚的技巧和設備，必能使他活得更長久而且更充沛。”

既然您已擁有全部的設備，就讓本書告訴您全部的技巧吧！



第二章 文字幕顯示



在整個 Apple 電腦系統中，最搶眼也是最重要的部份，就是鍵盤輸入和畫面輸出單元，因為它們扮演着操作者和複雜、快速運作的系統內部間溝通的主要橋樑。

操作者下達給電腦的指示，大部份經由鍵盤輸入；而系統根據這些指示所運作之結果，則藉 CRT 畫面顯示出來。除非您執行的程式屬於影像圖畫，通常電腦所輸出的，大抵為文字訊息——如符號、字碼和數字等。

因此，本書即選擇從螢幕和鍵盤開始討論。

標準文字幕格式

Apple II 電腦系統所設定之螢幕畫面，是以每行 40 個字，全部共 24 行的方式顯示，因此，畫面上可同時顯示 960 個字形。當然，在必要時，也可以使其每行不足 40 個字，或者總行數小於 24。

圖 2-1 所繪的，即是依照這個方式，將畫面分成 960 個字形位置，圖中每一行、列位置，均分別標示各自的行編號或列編號。

其中每一小方塊空間（即一個字形位置），並非正方格，而是高度稍大於寬度的長方形。

接着，請特別留意，無論是行或列，其位置編號都從 0 開始。因此，40 個行的編號是 0 ~ 39，而全部 24 列的編號則為 0 ~ 23，這乃是因為在電腦的世界裏，0 是計數的一個單位。根據以上規則，在畫面左上角的第一個顯示位置，其編號是 0 行、0 列。而畫面右下角的最後一個位置之編號為 23 行、39 列。整個畫面上，最接近中央位置的，則是 11 行、19 列。

依此類推，所有畫面上每一字形方塊，皆可定出它的行、列座標編號，通常，畫面顯示之操作，都是從左到右，逐步輸出。當然，您也可以根據實際需要，而在任何特定的畫面位置顯示字形。

游標運作規則

在大部份的畫面顯示操作中，游標（cursor）指示下一個顯示字形的正確位置。Apple II 電腦在執行 BASIC 或監督程式（Monitor）時，以閃爍的方塊作為游標指示。整個系統程式，根據一組特定的記憶內容值（記載着現時游標的行、列位置），即可正確地推算出游標的下一個顯示位置。

當 Apple II 執行整數型 BASIC 之操作狀態時，——畫面以“>”符號作為提示，游標即顯示於橫行之前端。此時如按下任何字母，數字或符號鍵，則游標亦隨着向前移動一格，永遠正確地指出下一個

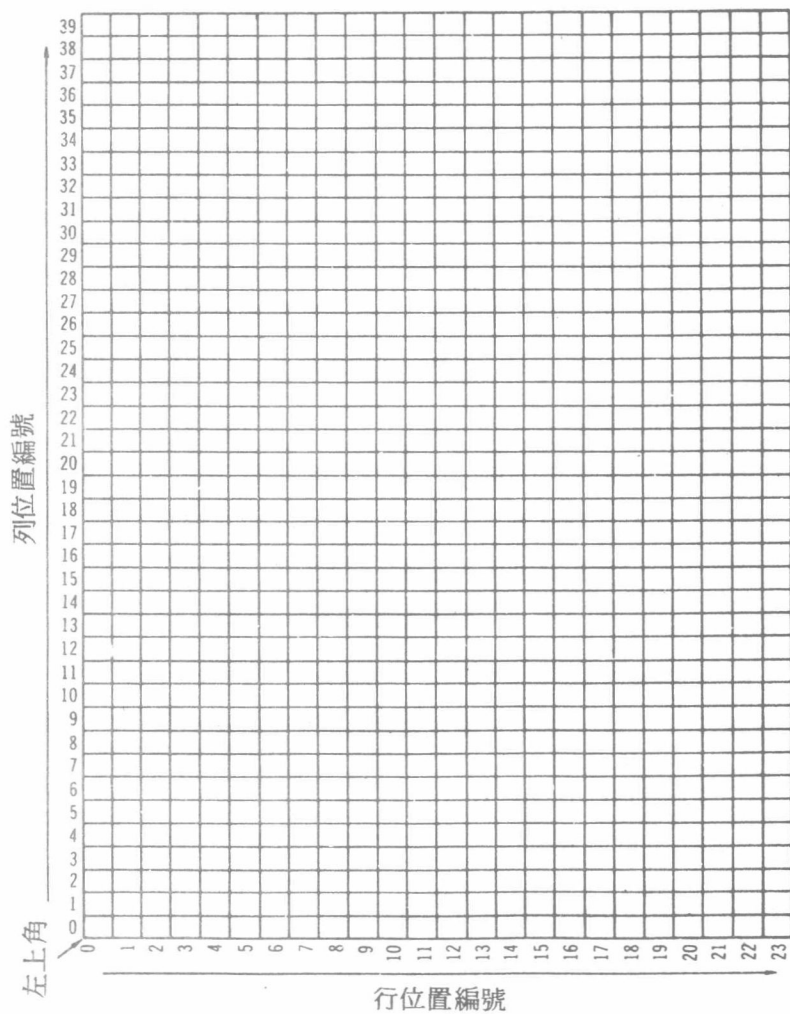


圖 2-1 螢幕畫面顯示格式

字形的顯示位置。若輸入之字形已填滿整個橫行（40 個字），則當繼續按鍵輸入時，游標將自動下移一行（Linefeed）並回到次行最左邊起點（Carriage return），而緊接着的按鍵輸入，即從此一位置繼續顯示。整個操作過程則如此週而復始，Apple 系統隨時會自動調整游標顯示的行、列位置。

當程式中某一敘述，將文字送往畫面顯示時，也仍遵守上述的運作規則。惟一的差別是，游標符號在印出的過程中不再顯示。因為當程式執行顯示操作時，實在沒有必要將不斷閃爍的游標方塊，也一併顯示，所以游標的功能，只是用來提醒操作者，當鍵盤輸入時，下一個字形的顯示位置。

假設在您所設計的 BASIC 程式中，含有一段 PRINT "HELLO" 敘述，則電腦執行這項敘述時，先在畫面上某一特定行、列位置顯示 "H"，然後使列位址編號遞增（即加一），以便顯示 "E"，接着又遞增以顯示 "L"，……直到顯示完畢。

這種游標定位的運作規則，在程式操作狀態或待命執行狀態，都同樣有效，不管其閃爍之方塊是否真正顯示，都永遠指向下一個字形位置。

以 PRINT 敘述控制游標

BASIC 語言的 PRINT 敘述，可命令系統在畫面上顯示文字符號或數值。除非指示電腦進行其他的操作，否則它將由一隱藏的游標導引，自左到右，逐次將各字形一一顯示。

● 簡單的PRINT敘述

簡單的PRINT敘述，可在顯示文字符號之後，自動執行換行（Linefeed）、回頭（Carriage return）之操作，除非加以設定，否則一律從各行的0列開始顯示。

例如執行下列BASIC程式：

```
10 FOR N=1 TO 4
20 PRINT "HELLO"
30 NEXT N
40 END
```

RUN（執行指令）以後，畫面顯示：

```
HELLO
HELLO
HELLO
HELLO
```

在行號20裏的PRINT敘述，被執行了四次，每次都從各行的0列開始顯示，並自動換行回頭。

除非另有指示，否則PRINT結束時，會自動換行、回頭。

這種自動換行、回頭的功能，無論所顯示的內容是文字、符號或數值，都同樣有效。例如下列程式：