

APPLE II 實用手冊

下冊

蕭金林編著

F — 顯示全文字或全圖像畫面

T — 設定為文字幕操作

G — 設定為繪圖操作

請仔細研究這段程式，即可了解如何運用表 6-6 之內容。

```

100 K= PEEK (-16384)
110 IF K<128 THEN 100
120 POKE -16368,0
130 IF K=208 THEN 230
140 IF K=211 THEN 240
150 IF K=212 THEN 250
160 IF K=199 THEN 210
170 IF K=198 THEN 200
180 IF K=205 THEN 220
190 GOTO 100
200 POKE -16302,0: GOTO 100
210 POKE -16304,0: GOTO 100
220 POKE -16301,0: GOTO 100
230 POKE -16300,0: GOTO 100
240 POKE -16299,0: GOTO 100
250 POKE -16303,0: GOTO 100

```

● 全畫面之繪圖操作

當系統設定為正常第一頁之全畫面文字幕操作狀態時，若執行 GR 指令，則轉變為文字幕和圖像混合顯示之狀態，並自動將圖像部分清除為黑色。假設此時您希望從全文字畫面，轉換為全圖像畫面，並且不想保留底端四行的文字顯示區域，而想將整個螢幕作為低解度繪圖之畫面，則可利用下列之 POKE 敘述達成。

```

POKE -16302,0
POKE -16304,0

```

在以一狀態下，畫面水平方向之繪圖範圍，和文字 / 圖像混合狀態時一樣，而垂直方向可資利用之範圍，則較為擴大，使得 PLOT 指令可繪圖之高度，從原來的 0 ~ 39 擴充為 0 ~ 47，這表示螢幕底端四行的文字顯示區域，也可作低解度繪圖之用。同樣的，VLIN 指令之垂直座標範圍，也隨着擴充為 0 ~ 47。

以下即為一全畫面繪圖操作之程式：

```
100 POKE -16302,0
110 POKE -16304,0
120 COLOR=4
130 FOR Y=0 TO 47
140 HLIN 0,39 AT Y
150 NEXT Y
160 COLOR=15
170 FOR Y=0 TO 47 STEP 47
180 HLIN 0,39 AT Y
190 NEXT Y
200 FOR X=0 TO 39 STEP 39
210 VLIN 0,47 AT X
220 NEXT X
230 GOTO 230
```

這段程式繪出一幅綠底白邊的圖案，而其範圍可及於整個畫面。雖然 HLIN 指令參數仍限制於 0 ~ 39，但 VLIN 指令則擴充到 0 ~ 47 的參數範圍。

當使用全畫面低解度繪圖操作時，必須特別留意，除非您打算結束操作，否則須避免讓程式停止下來。例如這段程式的 230 行，促使其進入永不休止的迴路，以防止程式執行 END 或類似的結束操作。如將 230 行刪除，或改為 END 敘述，重新執行之後，畫面顯示之結果就不一樣了。

若將程式後面幾行，修改並擴充如下：

```
230 FOR T=0 TO 1000: NEXT T
240 TEXT
250 CALL -936
260 END
```

則 RUN 以後，畫面上之彩色圖案將停留一段時間，（由 230 行之迴路決定時間之長短），接着 240 行使其恢復為文字幕狀態，然後 250 行清除畫面並移動游標至原點，而 260 行則結束操作。

經修改後的程式 240 行，說明了可從全畫面繪圖狀態，以 TEXT 指令使其恢復為正常之文字幕操作。此外，您也可以藉執行 POKE -16303,0 而獲得同樣之結果，為什麼？試將 240 行的 TEXT 指令改為 POKE -16303,0，並觀察其執行之結果。

總之，欲從全畫面繪圖狀態恢復為全文字操作，可有兩種方法，即 TEXT 或 POKE -16303,0。

正如可將資料碼載入低解度畫面一樣，當全畫面繪圖操作時，您也可以利用 POKE 指令，將資料碼載入螢幕底端原供文字顯示之區域，而顯示彩色之方塊。但假如您採用 $256 * \text{PEEK}(41) + \text{PEEK}(40) + \text{PEEK}(36)$ 的方式時，必須特別留意，應避免將螢幕右下角最後一格載入資料碼，否則將促使整個畫面內容上捲。

若從正常之文字幕操作狀態，轉換為全畫面繪圖操作時，系統並不自動將畫面清除為黑色。因此，如欲在繪圖之前，先將低解度之全畫面清除為黑色，則必須採取類似下列程式之步驟：

```
100 POKE -16302,0: POKE -16304,0
110 COLOR=0
120 FOR X=0 TO 39
130 VLIN 0,47 AT X
140 NEXT X
150 GOTO 150
```

希望您能自行分析這段程式之操作原理。

● 使用低解度繪圖之第二頁畫面

根據表 6-6 之記載，POKE-16299,0 可轉換為第二頁畫面，而 POKE-16300,0 則恢復為第一頁畫面。至於螢幕所顯示者是文字亦或低解度圖像，則以是否將 0 載入 -16303（文字），或 -16304（圖像）而定。除此之外，您尚可藉 POKE-16302,0 選擇全畫面文字 / 圖像操作，或者 POKE-16301,0 選擇混合顯示之操作。

在進行這些選擇之模擬前，您最好先複習一下有關第二頁畫面之操作功能，第四章所述者雖是文字幕狀態，但仍適用於低解度繪圖之操作。

首先，如果您打算在 BASIC 程式中，使用第二頁畫面時，必須先鍵入 LOMEM: 3072 之敘述，以便 BASIC 語言系統騰出這個範圍之記憶空間。（通常整數型 BASIC 將變數存放於第二頁畫面所屬的 RAM 記憶內。）

接着，由於無法以 PRINT 指令，直接將文字顯示於第二頁畫面，因此必須藉用 POKE 指令，將字形資料載入此一記憶空間之內。同

樣的道理，您也不能將 PLOT, HLIN, VLIN 或 SCRIN 指令，直接

應用於低解度繪圖之第二頁畫面，而必須以 POKE 指令，將資料碼載入。

最後，根據表 6-3 所記載，低解度繪圖之第二頁畫面和文字幕第二頁，均使用同樣的 RAM 記憶位址，因此，就螢幕上的同一位置而言，其在第二頁畫面之 RAM 位址，等於第一頁時的 RAM 位址加上 1024。

執行過 LOMEM: 3072 之後，輸入並執行下列程式：

```

10 REM ** THIS PROGRAM LATCHES UP IF YOU FAIL TO DO A LO
MEM:3071 FIRST **
15 POKE -16302,0: POKE -16304,0
20 VTAB 1: TAB 1
30 FOR N=0 TO 959
40 CP=256* PEEK (41)+ PEEK (40)+ PEEK (36)
45 CALL -1036
50 POKE CP,0: POKE CP+1024,0
60 NEXT N
70 END

```

程式可將第一頁和第二頁畫面都清除為黑色。執行時，您可從第一頁畫面上，觀察到它緩慢地將螢幕清除為黑色。如欲驗證它對第二頁畫面之執行效果，可從鍵盤輸入 POKE -16299,0，使之轉換到第二頁。程式之第 50 行，可同時清除第一頁和第二頁。

這段程式說明了兩點：第一，您可將第二頁視同第一頁畫面般的，只要將位址加上 1024，即可利用 POKE 指令，將圖形載入。第二，繪圖之過程極為緩慢，即使您將第 50 行修改成只用於一頁之範圍，執行之速度也快不了多少。而這種速度緩慢的感覺，可稍用點技巧，使得兩個畫面不同時進行清除，而將系統切換至先被清除完畢之畫

面，則另一個畫面清除之時間消耗，即可加以“隱藏”起來。

此外，還有一個技巧，就是以速度較快的 PLOT、HLIN 和 VLIN 指令，處理第一頁畫面，而只將速度慢的 POKE 指令用於第二頁畫面。

然而，實際上根本毋須如此大費周章，當以後論及機器語言繪圖程式時，您即可發現，它對第一頁或第二頁都極為快速。因此，大部分的動畫圖像，都以這種快速的機器語言程式處理操作。



第七章

高解度繪圖功能



Apple II 的高解度繪圖功能，是頗具趣味的程式設計項目，但也是一般電腦使用者很少觸及之範疇。假如您曾對於這項操作感到棘手，也毋須覺得忐忑不安，畢竟，大部分的 Apple 用者，都有過同樣的困擾。

爲了提供更方便的高解度繪圖操作，在整數型 BASIC ROM 裏的程式設計輔助 (Programmer's Aid, 即 Interger ROM 1)，含有一些副常式，可解決少部分 BASIC 程式設計之難題，然而，欲利用 BASIC 程式，充分發揮 Apple 的高解度功能，仍須藉用一些特殊之技巧。

儘管如此，Apple 的高解度繪圖操作，仍非深不可測，只要在程式設計時按步就班，並深思熟慮，一切之難題即可迎刃而解。

註：高解度等於高解析度；低解度等於低解析度

LOMEM和HIMEM之預估

欲在整數型 BASIC 之操作中，使用高解度繪圖，必須正確地估算 LOMEM 和 HIMEM 數值，這兩個指令，設定 BASIC 程式所使用的最低和最高記憶位址。

如何正確地設定 LOMEM 和 HIMEM 數值，取決於您的系統中所安置的 RAM 記憶之多寡，仔細研討有關 LOMEM 和 HIMEM 之特性，將有助於讓您了解它們和 RAM 空間及高解度繪圖操作之關係。

● 設定 HIMEM 位址

除非您特意安排，否則系統自動將 HIMEM 設定為可用之 RAM 的最高位址加一，此即正常 (default) 之 HIMEM 位值。換言之，系統中如有 16K RAM，則正常之 HIMEM 即設定為 16384；如具有 32K RAM，則 HIMEM 即為 32768；而在 48K RAM 時，正常之 HIMEM 則應為 -16384。Apple 系統在每次以 CTRL-B 進入整數型 BASIC 操作時，即自動設定其正常之 HIMEM 位址。

當 Apple 操作整數型 BASIC 時，總是從 HIMEM 減一之位址開始，逐一向下使用 RAM 記憶。因此除非您特意設定 HIMEM 等於其他非正常值，否則整數型 BASIC 總是從其最高之 RAM 位址開始，逐一向下使用記憶。

因此，假如您只有 16K RAM，則可供使用之 RAM 最高位址，恰等於高解度繪圖第一頁之最大位址，這一來問題就大了，不僅缺少高解度繪圖之第二頁畫面記憶範圍，而且所輸入之整數型 BASIC 程式，也正好寫入唯一的高解度繪圖記憶空間。因此，當只有 16K

RAM時，若保持HIMEM 為正常位址，則根本無法同時使用整數型 BASIC 程式和高解度繪圖操作。

欲解決這個問題，可在輸入程式之前，先設定HIMEM等於8192，即可將HIMEM移到高解度第一頁記憶範圍之底下，則BASIC 程式將從此一位址開始，逐次向下使用 RAM 記憶，而騰出高解度繪圖操作之空間。然而，您必須留意一點，每當您啟動Apple 系統，或按下CTRL-B時，都必須重新鍵入HIMEM: 8192。

至於32K或48K的系統，則HIMEM 之設定數值就不必太受限制了。在這種情形下，HIMEM 之正常值，總是遠大於高解度繪圖第二頁之記憶範圍。在32K RAM之系統裏，設定HIMEM 為正常值，還可留下8 K可自由使用之記憶，若犧牲高解度第二頁之畫面，則共有16K可用之RAM空間。而在48K RAM 之系統時，即使用到高解度第二頁畫面，HIMEM之正常設定值，也仍留下24K可自由運用之記憶空間，這對於一般之BASIC操作，已是足足有餘了。

● 設定LOMEM位址

不論Apple 系統所安插之RAM容量如何，LOMEM 之正常設定值，總是等於2048。此一位址即表示變數使用之起點位址，以便於BASIC 程式執行時，將數值變數由此位址開始向上存放。因此，當輸入整數型BASIC 時，程式之指令即自HIMEM位址，逐次向下存放；而在執程式時，所產生之變數，則從LOMEM位址開始，逐一向上堆積。如此一來，中間未使用的RAM 記憶範圍，則自兩端逐次縮小。

由於所設定之正常LOMEM 數值，遠低於任何高解度畫面所使用

之 RAM 範圍，因此，除非是極為特殊之整數型 BASIC 程式，否則將不致於使其變數使用記憶，佔用到高解度作圖之空間。

LOMEM還有一項極為重要之特性，它也是造形表之啓用位址。除非您特別指定將造形表存放於特定位址範圍，在一般情形下，它總是自 LOMEM位址開始堆積存放。

既然 BASIC 程式，將變數從 LOMEM 位址開始存放，而造形表也是從 LOMEM 開始堆積，那麼，這兩者是否會因而在同一記憶範圍裏混淆不清？

事實上，當造形表載入系統時，固然是從 LOMEM 位址開始向上堆積，然而一旦載入之後，系統即將 LOMEM 位址，從其正常數值移到造形表使用範圍之上端。因此，在使用造形表之情況下，LOMEM 數值已經調整到較高位址，而使得 BASIC 程式之變數，由此一新的較高 LOMEM 位址開始存放。

唯一的麻煩是，LOMEM 的正常設定值是 2048，它正好是文句 / 低解度畫面第二頁之起點位址，如果您計劃使用這第二頁畫面顯示，則必須鍵入 LOMEM: 3072，將 LOMEM 位址設定於第二頁畫面以外之位址。

一旦您將 LOMEM 重新設定在高於第二頁畫面之範圍，則在造形表載入時，即從此一 LOMEM 開始堆積，然後，再將 LOMEM 再度移到更高之位址。

● LOMEM和HIMEM之設定實例

表 7-1 和 7-2 列舉了 6 種繪圖畫面之組合，每一個表中的第一項組合是最單純的情況，只使用低解度第一頁畫面，而最後一項組合則

是最複雜之情況，使用高、低解度的第一頁和第二頁畫面。

表中註明“LOMEM設定值”和“HIMEM設定值”者，即是建議您在輸入 BASIC 程式前，針對畫面使用情況，所應設定之數值。而註明“可用之 RAM”一欄，則指操作整數型 BASIC 時，可供程式和造形表所使用之記憶空間。

表 7-1 適用於 16K RAM 之系統，而表 7-2 則適用於 32K 和 48K RAM 之系統。

表 7-1 16K LOMEM 和 HIMEM 之設定值

繪圖畫面之組合	LOMEM 設定值	HIMEM 設定值	可用之 RAM
低解度第一頁	正常值 ¹	正常值 ²	14 K
低解度第一頁和 第二頁	LOMEM: 3072	正常值	13 K
低解度第一頁 高解度第一頁	正常值	HIMEM: 8192	6 K
低解度第一頁和 第二頁 高解度第一頁	LOMEM: 3072	HIMEM: 8192	5 K
低解度第一頁 高解度第一頁和 第二頁	無 法 操 作		
低解度第一頁和 第二頁 高解度第一頁和 第二頁	無 法 操 作		

LOMEM 正常值¹ 是 2048

HIMEM 正常值² 是 16384

表 7-2 32K 和 48K 系統之 LOMEM 和 HIMEM 設定數值

繪圖畫面之組合	LOMEM 設定值	HIMEM 設定值	可用之 RAM
低解度第一頁	正常值 ^{1,3}	正常值 ^{2,4}	30 K / 46 K
低解度第一頁和 第二頁	LOMEM: 3072	正常值	29 K / 45 K
低解度第一頁 高解度第一頁	正常值	正常值	30 K / 46 K
低解度第一頁和 第二頁 高解度第一頁	LOMEM: 3072	正常值	29 K / 45 K
低解度第一頁 高解度第一頁和 第二頁	正常值	正常值	30 K / 46 K
低解度第一頁和 第二頁 高解度第一頁和 第二頁	LOMEM: 3072	正常值	29 K / 45 K

1 32K LOMEM 正常值是 2048

2 32K HIMEM 正常值是 32768

3 48K LOMEM 正常值是 2048

4 48K HIMEM 正常值是 -16384

例如您使用 16K RAM 之系統，而希望執行低解度第一頁和第二頁畫面之操作，並且在高解度之第一頁繪圖，則根據表 7-1 之記載，必須設定 LOMEM: 3072 和 HIMEM: 8192，因此，若在輸入 BASIC 程式之前，先設定這兩個數值，即可留下 5 K 之記憶空間以

供程式和高解度造形表使用。

如果您擁有 48K RAM 而希望使用低解度和高解度之第一頁、第二頁畫面，則只須將 LOMEM 設定為 3072，而不必更改 HIMEM 之正常設定數值，在此情況下，可提供大約 45K 之記憶空間供程式和造形表使用。（如為 32K RAM 系統，則有 29K 之可用空間）。

● 在程式中設定 LOMEM 和 HIMEM

整數型 BASIC 的 LOMEM 和 HIMEM 指令，僅可在待命狀態下使用，而無法在程式中執行此一敘述，這倒是很遺憾的事。如果 Integer BASIC 允許在程式中使用這項敘述，將可在程式之開頭處設定 LOMEM 和 HIMEM 數值，而不必在載入前先行鍵入。

實際上，是可能在整數型 BASIC 之程式開頭，設定 LOMEM 數值的。這就得充分利用 RAM 記憶中，存放 LOMEM 數值的位址 74 和 75，這兩個 byte 之位址，即被稱為 LOMEML 和 LOMEMH，合起來即是系統的 LOMEM 數值。假如您試驗如下：

1. 按下 CTRL-B 進入整數型 BASIC 操作。
2. 鍵入：

```
PRINT PEEK(74),PEEK(75)
```

即可在螢幕上顯示

```
0    8
```

此即目前用來設定 LOMEM 之兩個 byte，如將之合併轉換為十進位值，即等於 2048。（參考附錄 A 所述之十進位數值轉

換)

由此可見，當整數型 BASIC 啓動之初，的確是將 LOMEM 設定為 2048。

3. 鍵入：

```
POKE 74,0 : POKE 75,12
```

以便將數值載入 LOMEML 和 LOMEMH，而使得 LOMEM 被設定為 3072，這正是前面表 7-1 和表 7-2 所建議之 LOMEM 數值。

如此一來，您即可選擇在輸入程式之前，直接鍵入 LOMEM:3072 敘述，或者是在 BASIC 程式之第一行內，以 POKE 指令設定 LOMEM 數值。萬一您的 BASIC 程式裏，在尚未載入 LOMEM 數值之前，即先設定任何變數，則程式將無法掌握這些變數，而導致執行錯誤。

Apple 操作時之 LOMEM 數值，記存於 RAM 位址 74 和 75 之處。74 (LOMEML) 記存於低次位元，而 75 (LOMEMH) 則是高次位元。

至於 HIMEM 位址，則仍須在輸入 Integer BASIC 程式之前，以按鍵輸入設定其數值。如果在輸入程式之後才設定 HIMEM 數值，將造成系統之混亂，至少，所載入之 Integer BASIC 程式會有所漏失。

儘管如此，您仍然有必要了解 Apple 系統將 HIMEM 數值，暫存在記憶位址 76 和 77。

Apple 操作時之HIMEM數值，記存於RAM位址76和77兩處。76(HIMEML)記存於低次位元，而77(HIMEMH)記存於高次位元。

雖然您亦可讀取或修正HIMEM位址，但這項程序必須在輸入Integer BASIC程式之前進行。除非您甘願冒漏失BASIC程式之險，否則應避免在輸入程式後修正HIMEM數值。

啓動高解度繪圖操作

當您根據所須之繪圖操作，而設定LOMEM和HIMEM數值以後，即可進行兩個步驟，以便啓動繪圖作業程序。其第一個步驟即是設定一些高解度繪圖之變數，第二個步驟則是叫用高解度輔助常式。

●設定高解度繪圖變數

欲啓動高解度繪圖操作，必得先設定一些有關之變數，這項步驟亦須在輸入含有變數的BASIC程式之前進行。雖然BASIC程式執行中可由操作者定義一些變數，但此處用來啓動繪圖操作之變數，乃是指有關繪圖作業之特定變數。

在操作BASIC程式之前，所必須設定之高解度繪圖變數，共有6種之多，您可自行加以任意命名。這些變數之意義說明於表7-3。

雖然您可任意選用這些變數名稱，但命名時所使用的字數，却必須和其原來名稱相等。例如，您可以HX取代XX，或者也可用XO、

NN 等，但絕不能以單一字母，或多於兩個字的名稱代替之，例如 X 或 XPLACE 都不合規定。其他如 COLR 或 SHAPE 等變數名稱，也都必須遵守這項規則。

另外一項規則就是，設定這些變數時，必須依照表 7-3 所列之次序進行。如有違反此一順序，將使高解度繪圖操作錯誤。

此外，並非所有這 6 個變數都有必要使用。假如您在繪圖程式中並未使用造形操作，則根本不須設定 SHAPE、ROT 和 SCALE 三項變數。至於 XX、YY 和 COLR 變數，則在任何繪圖操作中，都絕對須要使用及設定。

表 7-3 高解度繪圖變數

變數名稱	定 義
XX	高解度畫面上點位置的橫向座標 (0~279)
YY	高解度畫面上，點位置的縱向座標 (0~159)
COLR	畫面上點或線描繪時的顏色碼 (如表 7-4)
SHAPE	繪圖造形表中所賦予之造形編號。造形表可存放 255 個造形及編號。
ROT	描繪造型圖案時之旋轉因素
SCALE	描繪造型圖案時之比例因素

如果您只希望描繪 (scale) 造形圖表，而不須要旋轉 (rotate)，那又該如何呢？在此情況下，仍須要設定 ROT 變數，才能使系統的變數表接受 SCALE 數值而完成操作。萬一您省略了 ROT 設定步驟，而在設定 SHAPE 後立即設定 SCALE，則 Apple 將誤以為