



全亞電腦

PA2000組合語言 操作實習手冊

駱德廉 編著

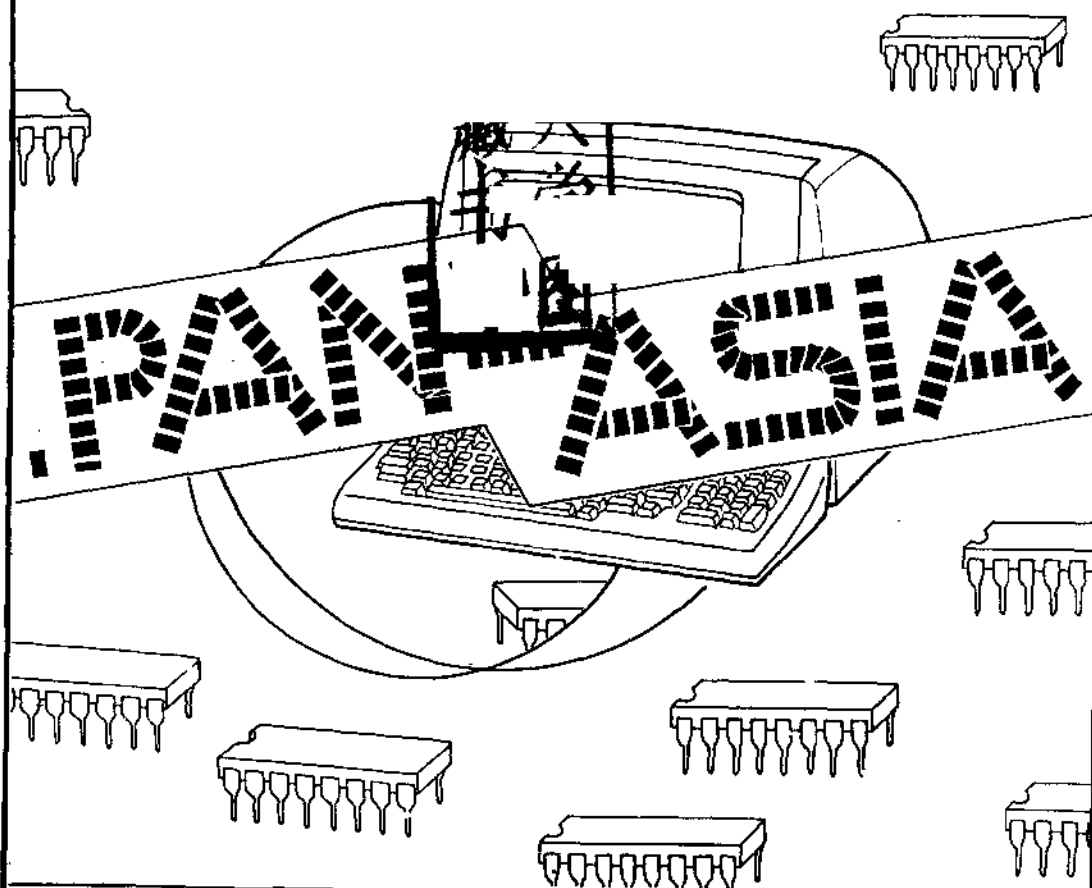


全華科技圖書股份有限公司 印行

 全亞電腦

PA2000組合語言 操作實習手冊

駱德廉 編著



全華科技圖書股份有限公司 印行



全華圖書 版權所有 翻印必究
局版台業字第0223號 法律顧問：陳培豪律師

**PA2000組合語言
操作實習手冊**

駱德廉 編著

出版者 全華科技圖書股份有限公司
北市龍江路76巷20-2號
電話：581-1300・541-5342
581-1362・581-1347
郵政帳號：100836
發行人 陳 本 源
印刷者 華一彩色印刷廠
定價 新臺幣 240 元
再 版 中華民國73年2月

感謝您

感謝您選購全華圖書！

希望本書能滿足您求知的慾望！

圖書之可貴 在其量也在其質

量指圖書內容充實、質指資料新穎够水準，我們就是本著這個原則，竭心盡力地為國家科學中文化努力，貢獻給您這一本全是精華的全華圖書。

序言

有人說八十年代是資訊的時代，也是電腦的時代；的確是如此大量資訊的收集與電腦快速的處理，已經把整個世界帶入到另一個新的境界了，今日世界各國莫不以發展電腦工業為其第一要務，尤其自從1971年微處理機加入了電腦的行列以後，更使得這項新興的行業對世界產生了震撼性的影響，因此有人稱它為第二次的工業革命。

全亞電子公司自從民國 65 年成立以來，為國內第一家微處理機專業化公司，積極從事國內微電腦應用的推廣；希望能藉此讓更多有志之士來加入微電腦的行列，共同推展我國的微電腦工業，多年來全亞公司無日不以此來期許，默默耕耘從事微電腦產品的開發與製造。

不斷的開發研究，推廣微電腦的應用，是全亞公司一貫的宗旨。平實的價格，專業化的服務是其努力的目標。從國內第一部國人自製的微型電腦學習機EDU80到PA2000多使用者的微型電腦，都足以顯示全亞公司的決心，更為國內電腦工業自力發展奠定了良好的基礎。

更難能可貴的，全亞公司為了讓廣大的使用者能以最短的時間達到最大的學習效果，編輯出一系列中文的使用手冊、實習手冊和軟硬體說明資料，希望能使每一個人（不論是理工或法商從業人員或學生）都能吸收此最新的科技，齊為今日的工業升級而努力。

本教材為PA2000組合語言的操作及實習教材，旨在以循序漸進的方式來教導各層面的學習者，讓讀者能在PA2000的微型電腦上充分發揮自己的能力，以組合語言配合著BASIC或FORTRAN高階語言來解決各方面的問題。

本教材共分兩篇，第一篇為組合程式的使用方法。旨在介紹如何在PA2000中使用ASMZ.COM、DEBUG.COM、LINK.COM與ED.COM等程式，以迅速有效的方式來進行組合格式

的設計及除錯工作。文中並以 CP/M 作業系統中的叫用方法為例，讓讀者更加能深層的運用微電腦。第二篇為 Z-80 組合語言的實習課程，教導讀者如何從頭學習組合語言，並輔以第一篇的操作方法而直接在 PA2000 系統上實習之，如此即可達事半功倍之效。

本教材編撰雖力求完美，但遺漏疏忽之處亦恐難免，盼望讀者於閱讀之後不吝指正謬誤之處，不勝感激。

駱德廉 謹識

民國七十二年七月



第一篇 組合程式的操作方法

第 一 章 組合語言程式設計入門	3
第 二 章 組合編譯程式的叫用方法	11
第 三 章 組合程式的結構欄	25
第 四 章 組合程式中假指令的使用方法	31
4-1 各種假指令的說明	31
4-2 形成各種機器碼的控制假指令	40
第 五 章 巨集和條件式組合語言	45
5-1 巨集組合語言	45
5-2 條件式組合語言	49
5-3 程式設計例題	50
第 六 章 組合程式的表列檔案	53
6-1 表列欄的說明	55
6-2 表列中的符號說明	56
第 七 章 連接 / 載入程式的使用方法	57
7-1 命令的格式	57
7-2 LINK 的控制開關	58
7-3 例題介紹	59

第 八 章 除錯程式的使用方法	63
8-1 簡 介.....	63
8-2 除錯程式的命令.....	65
8-3 除錯程式DEBUG, COM的命令表列.....	72
第 九 章 組合語言儲存庫ASM • LIB的使用方法	77
9-1 十進制數系的轉換.....	77
9-2 十六進制數系的轉換.....	78
9-3 字元輸入輸出副程式.....	79
第 十 章 BDOS系統叫用之組合語言設計方法	87
10-1 記憶體位置的分配.....	87
10-2 PDOS系統叫用的表列.....	88
第十一章 高階語言與組合程式的連接使用方法	111
11-1 FORTRAN 與組合語言.....	111
11-2 BASIC與組合語言.....	116



第二篇 PA2000組合語言程式設計實習

第 一 章 實習準備	123
第 二 章 基本算術與邏輯操作指令之運用	129
第 三 章 加減法練習	137
第 四 章 跳越指令程式環路	147
第 五 章 堆疊記憶與副程式	155
第 六 章 旋轉、移位指令和乘法程式	163

第 七 章 程式執行時間和記憶體位置作用	171
第 八 章 二進制除法程式	175
第 九 章 二進制對十進制轉換程式	181
第 十 章 十進制對二進制轉換程式	185
第十一章 開平方程式	193
第十二章 程式之中斷控制	201
第十三章 Z-80 CTC之使用	207
第十四章 外接SEVEN-SEGMENT之顯示練習	215
第十五章 紅、綠燈控制練習	221
第十六章 數位與類比資料之轉換	227
第十七章 數位IC功能測驗	241
第十八章 步進馬達控制	247
第十九章 數字時鐘	261
第二十章 PAB-PROG BOARD的使用	271
附錄一：Z-80 指令群	285
附錄二：PA-IDS 工業應用發展系統	297

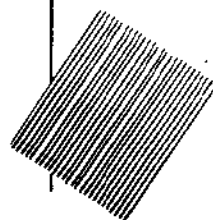
第 一 篇

組合程式的操作方法



組合語言程式

設計入門



組合語言是機器碼的助憶符號 (mnemonics)，它的目的地即是要提供一種較容易解釋的文字代號，來代表一計算機工作碼的指令。經由組合編譯程式的編譯，這些組合語言可形成各種的工作碼，載入記憶體中就可執行之。現在我們所要介紹的組合語言是 Z-80 的可重置巨集組合語言，它是一種 two pass (兩次處理) 的組合編譯程式，先從磁碟檔案中讀取原始的文字符號，將之組合編譯，再產生一個可重置的工作碼檔案和一個表列檔案。這些檔案可被送至任意的磁碟中，或控制台的螢幕上 (僅有表列檔案能在螢幕上顯示)。連接程式 / 載入程式 (LINKER / LOADER) 然後再將剛才組合編譯完的可重置碼，放入記憶體的任意指定位置中，所完成之機器碼可保存於磁碟的 COM 檔案中，做為命令程式的可執行檔案。

可重置組合編譯程式和連接程式是計算機用以產生機器語言程式的最有效方法之一。它可允許使用者建立許多個別的模組程式，分別的組合編譯之後，再使用連接程式將各個模組給連接起來。所以祇要各個模組間的界面定義清楚，一個大的程式可分成多個模組個別同時進行設計工作，在時間上可節省很多，效益上更因各個模組的工作量小而能考慮精密，而且這種方式下所產生的程式較易除錯。另外值得一提的是這種模組程式提供了一個資料檔案儲存庫 (LIBRARY)，使用者可以使用這個儲存庫中的已存在檔案來設計程式，而且這個儲存庫的叫用不一定要用組合語言，祇要界面符合定義，使用者可用其它的高階語言叫用之。使用者設計完的有效檔案，亦可加入或自己建立於新的

儲存庫中。

要使用組合語言的第一步即是將機器打開，將作業系統（CP/M）載入記憶體中，然後鍵入 `^DIR<RETURN>^` 命令查看檔案目錄中有沒有 `ASMZ.COM` 與 `ED.COM` 兩個檔案，若有的話就先用編輯程式（`ED.COM`）將所設計的程式輸入，形成原始的檔案，它的檔案種類（file type）必需使用 `.Z80`。等原始程式編輯完了之後，再使用編譯程式（`ASMZ.COM`）將原始程式翻譯，形成可重置的工作碼檔案（file type 為 `.REL`），以及表列檔案（file type 為 `.PRN`）。從表列檔案中我們可檢視原始程式中是否有句構上的錯誤，依據表列檔案的錯誤指示我們可重新使用編輯程式來修改之。

爲了要讓讀者瞭解上面所述的詳細步驟，現在以一個實際的操作例題來詳述之，在這個例題中我們要以 `Z-80` 的組合語言來設計程式，這個程式可促使控制台上的鈴聲每隔半秒中響一次。當然在進行編輯程式之前必須確定兩件事情，第一件即是 `CP/M` 作業系統是否已載入記憶體中，第二件即是系統組態是否具備從位址 `0` 開始之 `32k` 記憶體（`ASMZ.COM` 必須使用的最少組態），等兩個條件符合之後即用下列的指令將程式輸入。

```
A>ED TIMER.Z80
```

上面的指令輸入之後，編輯程式將會顯示出下列的訊息：

```
NEW FILE
: *
```

此時，編輯程式以“*”做為交談符號，我們可以用字母“`I<CR>`”來做為輸入的命令，這個命令的用意即是要將輸入資料插入游標所在的位置上。當“`I<CR>`”命令鍵入後，行數即顯示於前端，我們即可將下列的程式輸入：

```
NEW FILE
: *I<CR>
1:  ;THIS PROGRAM RINGS THE CONSOLE BELL AT
    APPROXIMATELY
2:  ; HALF-SECOND INTERVALS DETERMINED BY A
    TIMER LOOP
3:  ;
4:  BELL:      EQU      7
5:  WRITE:     EQU      2
6:  BDOS:      EQU      5
7:  TIMIT:     EQU      2FFH
8:  ;
9:  DURAT:     EQU      0FFH
10: ;
11: ;MAIN PROGRAM
12: ;
```

```

10:  START:   LD      SP,STACK
14:  LOOP:    LD      BC,TIMIT
15:  ;
16:  TIM2:     LD      A,DURAT
17:  TIM1:     DEC     A
18:           JR      NZ,TIM1
19:           DEC     C
20:           JR      NZ,TIM2
21:           DJNZ    TIM2
22:           LD      E,BELL
23:           LD      C,WRITE
24:           CALL    BDOS
25:           JP      LOOP
26:  ;
27:  ;STACK AREA
28:  ;
29:  BOTTOM:    DS      40H
30:  STACK:    EQU     $
31:  ;
32:           END
33:
: *E

```

在上面的資料輸入完之後，要跳出插入命令的模式祇需輸入 CTRL-Z 字元即可。若要檢查上面所輸入的資料是否有錯誤，則可以使用“B<CR>”命令將游標移至最頂端，再以T或P命令來逐行或逐頁檢視之。若有錯誤，則可以S命令來更改錯誤的資料，S命令的格式為

S<old text>^ [<newtext>^]<CR>

在這的“^ [”字元即是 CTRL-Z 鍵。

當編輯工作完成之後，我們可以用“E<CR>”命令來跳出編輯程式的控制，而剛才所輸入的資料會全部保存於磁碟檔案中，“TIMER.Z80”，如此的話交談符號又會恢復“A>”的字樣。

現在，我們可進行編譯之第二步工作了，輸入下列的命令即可：

A>ASMZ TIMER<CR>

編譯程式在這命令輸入之後即開始執行之，它會將現在磁碟中的原始檔案載入，經編譯後產生.REL工作檔案與.PRN表列檔案。當編譯工作完成之後，螢幕上即顯示下列的訊息，且顯示出“A>”交談標誌：

Errors 0

```
end of assembly
```

```
A>
```

如果上面所示之 Error 訊息不為 0，則需要再次使用編輯程式來修改原始程式，並再次以編譯程式來翻譯之。使用者可以“TYPE TIMER.PRN”命令來將表列檔案顯示於螢幕上，在按下< CR >鍵之前若先按下了 CTRL-P 鍵，則其資料會同時輸至印字機上。在這個表列的檔案中，第一行是指令的十六進位位址，第二行中可能是(1) 4 數元 (byte) 指令的工作碼，或(2) 4 數元資料的工作碼，或(3) 運算元表示式的相等數值。第三行是原始資料的行數號碼，第四、五、六行是原始檔案的組合語言程式。

當編譯完成而沒有任何錯誤之後，.REL 檔案必須要經過連接/載入程式的處理，才能形成.COM的可執行檔案。所以我們必須輸入命令：

```
LINK TIMER
```

連接程式執行時會出現“*”交談標誌，它的意思即是在等待更進一步的指令，於此，你可選擇執行連接的工作或執行跳回系統監督的工作，跳回系統之後可用 SAVE 命令將工作碼保存而形成一命令檔案，這個命令檔案即是所需要的可執行檔案。讓我們選用第二種方法（希望使用第一種方法的，祇需在“*”交談標誌後輸入/G）於“*”之後輸入/E命令即可跳出連接程式而進入系統中，這時LINK 程式會在控制台的螢幕上印出訊息：

```
[ 1000 105A 16 ]
```

這訊息中的第一個數是工作碼在記憶體中所佔的起始位址，第二個數是我們程式所佔用的最高位址，而最後一個數即是命令檔案所需保存的記憶點頁數。依據這些資料我們現在可輸入下列的命令來建立.COM檔案：

```
SAVE 16 TIMER.COM
```

建立完命令檔案之後，你祇要直接使用命令檔案的名稱就可以將程式載入記憶體中，從位址 100H 開始載入，並從 100H 位址開始執行之。所以LINK 程式必須將“JP 1000H”的指令轉換成 100H 位址可執行的指令，我們現在祇需輸入“TIMER”，就可以聽到控制台 CRT 每隔半秒的鈴聲了。

在組合語言的程式設計中，實際上經常要在設計的過程中使用除錯程式 (DEBUG.COM)，這個程式提供了許多特殊的功能，讓使用者以逐步對照暫存器內容和記憶體內容的方法來偵測程式設計上的錯誤。使用除錯程式的第一個步驟就是將所要被除錯的檔案名稱放在DEBUG 命令的後面，如此就可進入除錯程式中。其它的步驟和使用例題

將在後面詳述之。

例題：下列所示即是上面所述各個步驟的詳細操作表列：

A>ED TIMER.Z80

NEW FILE

```

: *I
1: ; THIS PROGRAM RINGS THE CONSOLE BELL
   AT APPROXIMATELY
2: ; HALFE-SECOND INTERVALS DETERMINED
   BY A TIMER LOOP
3: BELL: EQU 7
4: WRITE: EQU 2
5: BDOS: EQU 5
6: TIMIT: EQU 2FFH
7: ;
8: DURAT: EQU 0FFH
9: ;
10: ; MAIN PROGRAM
11: ;
12: START: LD SP, STACK
13: LOOP: LD BC, TIMIT
14: ;
15: TIM2: LD A, DURAT
16: TIM1: DEC A
17: JR NZ, TIM1
18: DEC C
19: JR NZ, TIM2
20: DJNZ TIM2
21: LD E, BELL
22: LD C, WRITE
23: CALL BDOS
24: JP LOOP
25: ;
26: ; STACK AREA
27: ;
28: BOTTOM: DS 40H
29: STACK: EQU $
30: ;
31: END
32:
: *ST
: *R
1: *ST
1: ; THIS PROGRAM RINGS THE CONSOLE BELL
   AT APPROXIMATELY
2: ; HALFE-SECOND INTERVALS DETERMINED
   BY A TIMER LOOP
3: BELL: EQU 7

```

```

4:  WRITE:  EQU      2
5:  BDOS:   EQU      5
1:  *3:
3:  *I
3:                ORG      100H
4:
4:  *B
1:  *10T
1:  ; THIS PROGRAM RINGS THE CONSOLE BELL
    AT APPROXIMATELY.
2:  ;  HALFE-SECOND INTERVALS DETERMINED
    BY A TIMER LOOP
3:                ORG      100H
4:  BELL:   EQU      7
5:  WRITE:  EQU      2
6:  BDOS:   EQU      5
7:  TIMIT:  EQU      2FFH
8:  ;
9:  DURAT:  EQU      0FFH
10: ;
1:  *B
1:  *FBELL^Z
1:  *T
AT APPROXIMATELY
1:  *FBELL^Z
4:  *5T
:      EQU      7
4:  WRITE:  EQU      2
5:  BDOS:   EQU      5
6:  TIMIT:  EQU      2FFH
7:  ;
4:  *E

```

A>ED TIMER

NEW FILE

: *Q

Q-(Y/N)?Y

A>ED TIMER.Z80

```

:  *4A
1:  *30T
1:  ; THIS PROGRAM RINGS THE CONSOLE BELL
    AT APPROXIMATELY
2:  ;  HALFE-SECOND INTERVALS DETERMINED
    BY A TIMER LOOP
3:                ORG      100H

```