

6502 系統程式

適用：OHIO SCIENTIFIC, APPLE,
ATARI 及 CBM

王 銘 傑 合 譯
蕭 金 林



科藝出版社

6 5 0 2 系統程式

適用： OHIO SCIENTIFIC, APPLE,
ATARI 及 CBM

王 銘 傑 合 譯
蕭 金 林

科藝出版社

6 5 0 2 系統程式

合譯者：王 銘 傑 · 蕭 金 林

出版者：科 藝 出 版 社

發行者：科 藝 出 版 社

地 址：九龍西洋菜街102號三樓

印刷者：達 華 印 刷 廠

地 址：香港柴灣工廠大廈10樓

定價：港幣貳拾捌元正

序 言

近年來由於政府的大力倡導以及電腦科技之進步，隨著其日漸普及，因此價格也下降到大多數人可接受的程度，因此現在談到電腦，已不再會令人感到神秘兮兮，遙不可及了，尤其微電腦應用系統已逐漸為國人所接受；這是一件可喜的現象；自從電動玩具遭取締後，仿間出現了大批仿製的電腦產品，一部在國外售價約2000美元的蘋果牌電腦，如今只需花費新台幣1500元甚或更低的價格即可在國內購得，表面看來是普及了，但從其一成不變的照抄，銷售者的缺乏專業知識，使用者在毫無保障，無人引導的情況下，大部份的人不是將之冷藏就只玩玩遊戲，徒呼“受騙”，誠然非國內資訊工業之福。

當從楊先生手中得到這本書時，第一眼的印象是真實用也很重要，爾後當楊先生提議翻譯時就一口答應下來，我們的動機乃希望提升使用者對電腦的認識及使用的層次，不但要能玩遊戲更能明白電腦的基本軟體作業系統，本書以OHIO SCIENTIFIC公司的CIPMF 650作業系統為基本架構，涵括了目前最著名的四套使用6502 CPU的個型電腦(OSI, APPLE, ATARI及CBM)之系統程式，深入淺出的敘述，將引導讀者建立作業系統的概念，這種屬於「紮根」的工作與知識，也是國內「電腦熱」所欠缺的，但願本書的出版能克盡棉薄之力。

最後我們願意揭櫫以下兩點供大家參考，學電腦不是紙上談兵要多操作實習，發展電腦無捷徑，COPY非正途，唯有從基礎腳踏實地做起。

王銘傑 謹識

目 錄

	頁數
介紹篇：概論與目的.....	1
第一章：認識電腦.....	4
第二章：組合程式語言.....	10
第三章：迴路及常式.....	21
第四章：算術及邏輯運算.....	33
第五章：螢幕顯示功能.....	44
第六章：系統監視能力.....	60
第七章：文句顯示功能.....	83
第八章：記憶內容展佈.....	97
第九章：解組合語言程式.....	114
第十章：搬移功能.....	135
第十一章：文句編組功能.....	145
第十二章：監視程式之擴充.....	159
第十三章：將程式載入系統記憶.....	162

附 錄

A 1.	十六進制轉換對照表	171
A 2.	ASII 字碼表	172
A 3.	6502 指令集—助憶碼表	173
A 4.	6502 指令集—動作碼表	175
A 5.	指令執行時間	179
A 6.	6502 助憶動作碼及定址法	181
B 1.	Ohio Scientific Challenger I-P 電腦	185
B 2.	PET 2001電腦	188
B 3.	APPLE II 電腦	195
B 4.	ATARI 800 電腦	203
C 1.	螢幕顯示應用程式	211
C 2.	螢幕監督程式(顯示常式)	223
C 3.	螢幕監督程式(更正常式)	233
C 4.	印字應用程式	243
C 5.	兩種十六進制展佈方法	257
C 6.	查表式解組程式(應用常式)	275
C 7.	查表式解組程式(定址常式)	287
C 8.	查表式解組程式(列表)	299
C 9.	搬移應用程式	317
C 10.	簡單之文句編輯程式(顯示常式)	329
C 11.	簡單之文句編輯程式(編輯常式)	337
C 12.	螢幕監督程式之功能擴充	349
C 13.	OHIO SCIENTIFIC C-IP 之系統資料集	355
C 14.	PET 2001 之系統資料集	361

C15	APPLE II 之系統資料集	367
C16	ATARI 800 之系統資料集	373
D1	螢幕顯示應用程式	389
D2	螢幕監督程式(應用常式)	390
D3	螢幕監督程式(更正常式)	391
D4	印字應用程式	392
D5	兩種十六進制展佈方法	393
D6	查表式解組合程式(應用常式)	395
D7	查表式解組合程式(定址常式)	396
D8	查表式解組合程式(列表)	397
D9	搬移應用程式	399
D10	簡單之文句編輯常式	400
D11	螢幕監督程式之功能擴充	401
E1	螢幕顯示應用程式	403
E2	螢幕監督程式(顯示常式)	405
E3	螢幕監督程式(更正常式)	407
E4	印字應用程式	409
E5	兩種十六進制展佈方法	411
E6	查表式解組合程式(應用常式)	413
E7	查表式解組合程式(定址常式)	415
E8	查表式解組合程式(列表)	417
E9	搬移應用程式	419
E10	簡單之文句編輯常式	421
E11	螢幕監督程式之功能擴充	423
E12	Ohio Scientific C-IP 之系統資料集	424
E13	PET 2001 之系統資料集	425
E14	APPLE II 之系統資料集	426
E15	ATARI 800 之系統資料集	427
	索引	429

介紹篇

概論與目的

平常我們總可以聽到人們談論電腦是多麼的精明能幹，無所不能，實際上一點也不是那麼一回事，真正電腦功能的發揮乃是拜程式設計人員所賜，由於程式設計人員的巧妙設計安排，使得電腦能扮演一部簡單的電子計算機，亦可成為登陸月球的要角，甚至為你完成薪資所得稅報表的準備，因此要電腦萬能必需要程式設計人員夠聰明，否則電腦只有一籌莫展的份了。

不可諱言地，良好的程式設計是使用電腦成功之基本要件，有鑑於這個重要性，本書中揭櫫的主要目標有二：第一希望介紹給初學者的技巧、名詞以及作用，第二，以組合語言程式提供一套軟體工具來發展 6502 CPU 的應用。

第一章中，我們以最快速精簡的方式將有關的系統軟硬體做一番瀏覽，第二章到第四章則以初學者為對象將組合語言加以複習介紹，其餘各章則分別以原始程式 (Source Listing)、機器碼 (Object Code) 及組合程式形態討論一些可能需要送入電腦執行的系統程式。

程式設計人員往往希望發展長度較短且執行起來較快的程式，但由於為了要縮短幾個位元或縮短執行時間，因此錯誤在所難免，而且程式閱讀不易或無法工作，當然若因受到實際記憶空間及運算時間之限制要求，那麼程式的易讀性 (Readability) 就不太重要了，但事實上遭遇這種壓力的情況並不多見；在正常的評估標準下，若各種狀況都正常的情況，一個程式的價值高低就由其是否精簡省時快速來評斷了。

但是否各種主客觀條件都可能是平等的呢？

但本書中所出現的程式，在設計的過程中，強調了一些在純粹考慮程式大小及執行速度快慢更要來得重要的幾個主題，以下幾點即是對程式設計的要求：

＊功用性：沒有只是用來表現特殊程式設計技巧的程式，在本書中所出現的所有程式都是有目的地，亦即一些自己不欲做而需交給電腦做的工作，例如用來監督電腦作業的監督程式 (Monitor)，將機器語言轉換成組合語言的反組程式 (Disassembler)，可以幫助我們輸入及修改鍵入文字及畫面的編輯程式 (Text Editor)，這些程式可謂之各有專長。

＊易用性：必需要能夠很容易就從螢幕上得知是什麼程式正在執行？處於何種操作模式？當執行過程中，需要其他資料時能加以獲得並允許更正錯誤的發生，有了此軟體程式，就可不需刻意去記程式的所在位置或變數名稱所代表的地址，甚至於可以各別的按鍵來對應不同的功能，或按照自己的習慣方式去指定功能對應的按鍵。

＊易讀性：對一位 6502 初入門的程式設計人員而言，必需要瞭解本書中各程式的工作原理，程式中的標記 (Label) 及備註 (Comment) 敘述，都適度的加入說明，使你明白各變數名稱，副程式 (Subroutine) 以及各指令所代表的意義，加了這些說明雖然對 6502 CPU 並沒有特殊作用，但却有助於對程式內容的瞭解。

＊方便性：本書中所討論的軟體程式，可適用在 OSI (OH IO SCIENTIFIC INC) 的 CIP 電腦，APPLE II，ATARI 400，800 及 PET 2001 上執行，只要將系統資料集 (System Data Block) 經過妥善的規劃，那麼即可在任一以 6502 CPU 為主的電腦配上鍵盤，記憶器及螢幕顯示器上加以運用。

＊共通性：本書各程式在設計時所佔用之位置乃從 1000 到 1FFF，屬於第二個 4KB 的位置，因此與你原先系統所有的程式是不相衝突的，尤其大部份的程式都保留了第零頁 (Zero Page)，因此你可隨心所欲的利用第零頁位置，也不需擔心在程式執行前後的資料儲

存問題。

＊擴充性：本書中的程式採高度模組式設計，所以你可加以延伸或重新安排組合以符合你自己的需要；系統特定的副程式採用間接呼叫定址，故可由其他程式來取代，此外數值大多以變數名稱來表示而不限制為定值，同時這些程式例既非單一用途亦非一成不變，因此你可將這些副程式加以組合，發揮其更大的功能。

＊精巧型：由於個人電腦（Personal Computer）的記憶容量不大，所以總會想把程式縮短個百分之十或二十，但這麼做的話，却又會破壞到上述的幾個原則如可讀性，方便性及擴充性，因此我們不擬如此去做。在許多情況下，可能會因為了省一個位元的記憶空間而使讀者不易明白程式的工作原理，最該考慮的是切勿因求簡省而在其他方面付出相當的代價。

＊快速：假設上述之目標皆已達到，本書中的程式已都儘可能被設計為最快速省時地，但在介乎快慢間的判定標準，設若一個快而不易懂的程式是沒有太大價值的，因此書中的例子絕不會令你抱怨要花太多額外的時間去瞭解。

有了上述的認識，那就靜下心來好好的閱讀吧！！

第一章

認識電腦

本書中的軟體程式可適用到相當多的電腦上，基於各電腦間只有極其微小的差異，所以在此我們只約略地加以說明。

6502微處理機

6502 微處理機是本書中的要角，當然若沒有其他元件的配合使用，是變不出什麼把戲的，其本身結構擁有三個暫存器（Register），即A，X及Y，各暫存器之長度為八個數元可儲存0到255間的數值資料，且各暫存器依功能不同具有相對的運算能力，例如A是疊算器（Accumulator）在此可進行數值255以下的加減運算，設若改成X或Y，那麼就只能夠做增減1的運算而已。

6502 微處理機能做的事可還多著呢！例如可設定某個暫存器相等於另一暫存器所存放的數值，暫存器與記憶器之間的資料相互擷取與存錄，諸如此類的動作即構成了整個系統的有效運轉。

記憶器之型式

電腦儲存資料的方式是以一串列的“0”與“1”來表示，其乃因電腦本身的記憶器可視為一連串精密的開關所組成，而任一開關的存在狀況非“開路”即為“閉路”，有了這兩種情況，就可代表了“ON”及“OFF”或“0”與“1”。

記憶器大體上可分為數類，一為僅讀記憶器ROM（為Read Only Memory之簡稱，此種記憶器的內含是固定而無法加以改變的；另一則為RAM（Random Access Memory）其具有供讀

(Read) 寫 (WRITE) 之功能，可隨意依需要而寫入不同之數值，圖 1 . 1 顯示出兩者之間的差異。

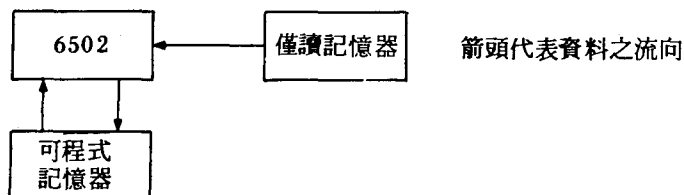


圖 1 . 1 6502 微處理機與記憶器間之關係

第三種記憶器與前二項由半導體構成的記憶器有所不同，而由一些外圍裝置所構成，如此的結構稱之為輸入埠 (INPUT PORTS)，透過這些輸入孔道，資料及信號可順利地傳送到 6502 微處理機，譬言鍵盤、終端機或語言產生器所產生的信號皆可經由由此輸入埠而被 6502 處理機在擷取各相關的埠別時加以讀入。

與第三種相對應的乃是輸出埠 (Output Port)，6502 處理機可將資料設定為“ 1 ”或“ 0 ”的狀況將之傳送到輸出埠，亦即透過上述輸入 / 出埠使得 6502 微處理機，得以成功地與外界溝通。

通常我們把這些經由輸入 / 出埠而與 6502 微處理機進行資料傳送接收的裝置稱之為週邊設備 (Peripheral Devices) 例如卡式錄音機，印字機，磁碟機甚或語音合成器，其中卡式錄音機及磁碟機具有資料儲存能力且容量亦大，又稱之為輔助儲存器 (MASS STORAGE 或 SECONDARY STORAGE)，圖 1 . 2 中描述了 6502 微處理機與上述記憶器間的相互關係。

假設有一螢幕顯示器 (Video Screen) 聯接到你的電腦上，我們可將螢幕視為 6502 記憶器的一部份，因此即可從此螢幕上讀出或寫入所需的資料 (譯者註：對螢幕上各顯像位置賦予相對應的記憶地址)；此外 6502 亦可採用掃描 (SCAN) 的方式將接在輸入 / 出埠的鍵盤按鍵信號加以讀入，此時鍵盤即可視之為 6502 上某一特定的

6 6502 系統程式

的記憶地址。如此一來，雖然 6502 實際上僅能直接處理記憶器的資訊，但因所有的輸出／入裝置都映對到某些特定的記憶位置，所以 6502 微處理機就等於是處於與使用者直接交談的作業方式（INTERACTIVE），請參照圖 1 . 3。

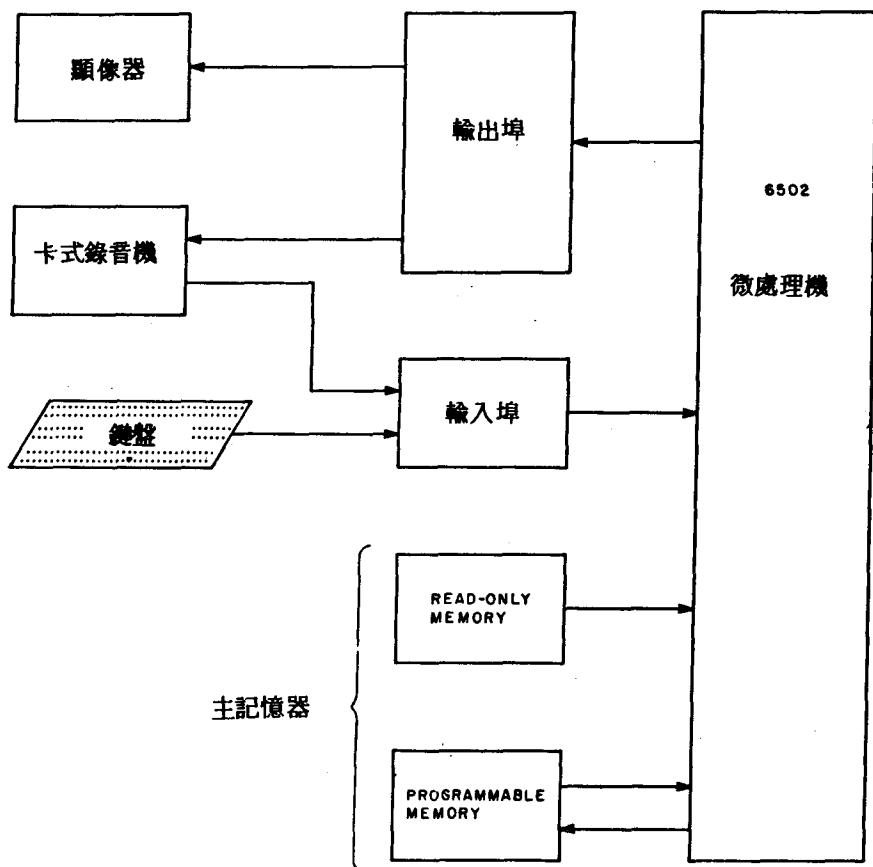


圖 1 . 2 6502 處理機與主記憶器及輸出入埠間資料傳送關係圖

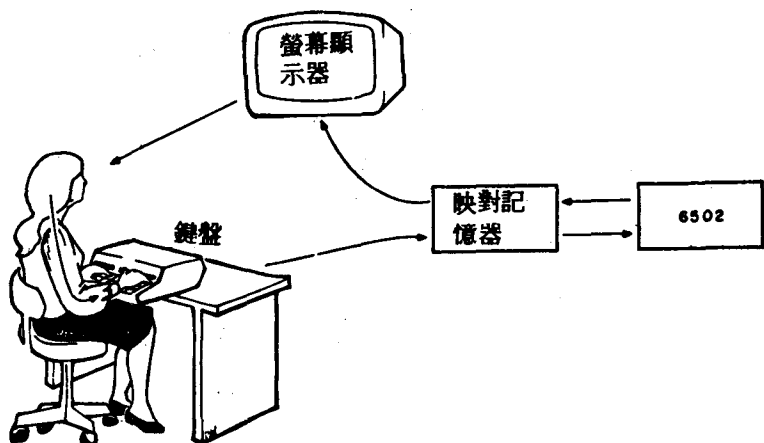


圖 1 . 3 6502 的人機交談方式，箭頭表資料流向

作業系統

到目前爲此，我們已大致對硬體結構有了粗略的描述，但事實上本書中所討論到的 OSI，ATARI 以及 PET 等電腦系統還都需要有其他軟體的支援才堪使用，例如每部機器必定會有一段儲存在 ROM 裡的作業系統程式去執行輸出／入的工作，以便配合螢幕或鍵盤的操作，相信你的電腦亦必有類似的這些工作程式，至於其原理則不擬多提。在一套電腦的軟體作業程式勢必包函了很多的副程式（Sub-routine），分別處理各相關作業，至於如何發揮其功能，那就需視自己對整個系統瞭解及熟悉的程度而定了，因此你若對自己的系統愈清楚那麼即可「熟能生巧」，爾後自然就可得心應手了。本書中所提及的系統軟體可適用於以上各系統而非特別侷限某一特殊機種，讀者可從中吸取重點，倘能再聯結自己原系統現行的有關副程式使用，則將如虎添翼，大大地發揮功能了。

BASIC 程式語言

人與人之間需有一套彼此瞭解的語言或文字才能加以溝通意見，人與電腦亦不例外，本書中所討論的這四種電腦系統，都具備有一項很重要的 BASIC 程式語言翻譯器 (INTERPRETER)，其儲存在系統中燒錄好的僅讀記憶器 (ROM) 一開機即可使用；所謂翻譯器亦屬程式的一種，其負責將你依 BASIC 語法所寫好的程式，變成可被電腦接受並加以執行所規定的工作。由於這是屬於口語化的高階電腦程式語言，所以只需假以時日並配合電腦實習，應可很容易就學會，至於其細部寫作設計技巧及指令應用，請詳細參考你所擁有的電腦所附送的手冊及相關書籍。

雖然 BASIC 已屬於通用的高階語言，但應用在不同的系統中亦會有些微的差異，所以奉勸讀者在使用機器之前，務必先行閱讀其專用之說明書，否則招數亦會有失靈的時候，切記!!

6502 機器碼

本書中所討論的 OSI，PET，APPLE 及 ATARI 等坊間最流行的個人型電腦，皆有一共同的心臟部位— 6502 處理機，當然必有一套完整的指令可用來加以執行，此處所謂的指令是指最原始而直接命令電腦動作的，我們稱之為機器碼，如予以符號化則提升為組合語言 (ASSEMBLY)，無論如何從使用層次而言是較 BASIC 為低，故亦稱為低階語言，例如在 BASIC 中我們只需以一行程式敘述完成的動作 (如 PRINT “HELLO”)，換成機器指令則要好幾個湊起來才能達成同樣的結果。(譯者註：在某些情況下亦有使用機器指令較 BASIC 便捷的情況，這全端視問題之要求以及對機器指令熟悉及靈活應用的程度而定了)。

既然機器指令直接命令 CPU 工作，因此不同的 CPU 就會有其不同的機器指令，而無法互通使用，所以本書所討論的程式自無法應

用以 8080 , Z-80 及 M6800 為基礎的系統（譯者註：雖無立竿見影之套用價值，但對觀念的建立亦有良好的學習參考需要）。

本書中所舉出的程式一律以機器指令表示，且皆控制在 4KB 的範圍完成，讀者可以保留下足夠的剩餘記憶空間去擺入自己的程式。

第二章

組合程式語言

有許多事情，在驟看時會覺得很複雜，困難甚至不可思議，但若
能充分冷靜下來，加以研究分析，瞭解其動作基本原理，那麼一切就
豁然開朗了。例如馬戲特技中所常見的「雙手三球」把戲，其實若以
慢動動作來分析只不過是三個簡單的連續動作，傳球→拋球→接球，
且必需恰到好處地保持一個球在空中，這個比譬與組合語言雖無直接
關係，但觀察 CPU 的動作亦有相似的動作原理，從記憶器中將資料
取出→運算→存回記憶器，有了這樣的認識，讀者該不會還百思不解
、望組合語言而却步了吧！！

我們若將 6502 處理機想像成特技表演人員，那麼你會發覺
6502 比人強太多了，表演人員才只有一雙手，但 6502 却有三個暫
存器（分別命名為 A，X 及 Y）以及無數個記憶位址可供利用（本書
中所討論的 4 種電腦系統至少都有 8KB 的記憶器），在 6502 的天
地裏，每一暫存器及記憶位置皆可存放一個位元（BYTE）的資料，
因此一次運算指令亦只牽涉到一個位元值的動作，由於其執行速度甚
快，所以在一秒鐘內就可以改變了數以千百計的位元資料。

（譯者註）以下本章所述至第四章乃就一些基本概念做一番很快
的複習與提示，若讀者原就不曾學過組合語言，那麼還是希望你能將
組合語言學好後，再來閱讀本書，否則吸收不了，恕不負責。

二進制（Binary）

由於在電腦裡，任何資料最終都以一連串的數元（BIT）形式來
儲存，這種以“0”或“1”來描述位元（BYTE）資料的表示法，