

適用於心理學實驗之 6502 改良軟體計時程式

吳 瑞 屯
國 立 臺 灣 大 學

爲了能適用於大部分的心理學實驗，作者將 Price (1979) 之微電腦軟體計時程式加以改良，使之可以滿足下列條件：1. 反應不一定只從鍵盤輸入，若程式需要，亦可指定任意 I/O 狀態的改變導致計時停止；2. 計時極限可長達 6 小時。本文提出改良後之組合語言程式，說明設計邏輯，並舉例闡明程式之呼叫方法，所有程式皆以機器可讀之型式存於磁片，以供需要之研究者複製。

自從 1975 年微電腦開始出現以後，在實驗心理學的領域裏，有逐漸增加的研究者開始使用微電腦進行實驗程序控制。由於電腦所具備的快速與精確特性，使得實驗時因爲程序或資料登錄過程中所產生的誤差得以減至最小，並且，由於程序的自動化，使得因子數目較多，或嘗試數多的複雜設計，可以在不增加受試者負擔的條件下順利進行，這意味著研究者對問題的探索在技術上能往更深更細的層面推進，某些較複雜理論的較深入驗證也逐漸變成可能。不管是作爲程序控制，或者透過界面 (interface) 連接測量儀器進行資料的自動蒐集，微電腦皆可大幅度改進實驗的品質。對於認知心理學者而言，由於螢幕就是一種呈現刺激材料的設備，因此，無論就研究或教學的角度而言，微電腦皆可取代大量的早期設備而成爲一種一般性而多目標的儀器，也因此可以斷言電腦已成爲今後從事較精細實驗之必備工具。

很多利用微電腦進行心理學實驗所需的軟體技術已陸續被發展出來，這些技術大致可分爲三方面：其一爲有關歷程操弄的技術，例如計時、刺激之隨機、特殊程序等 (Durso, 1984; Osaka, 1979; Owings & Fieldler, 1979; Reed, 1979)，其二爲有關界面與外部實驗儀器之控制的技術，例如幻燈機、測試儀器之控制 (Allen & Nicora, 1984; Loftus, Gillispie, Tigre, & Nelson, 1984)，其三則爲與資料分析有關之程式，例如統計計算、模擬，或一些特殊計算程序等 (Anderson, 1984; Campbell, 1984; Coulombe, 1984a, 1984b)。本文的目的係企圖討論在以 6502 微處理機作爲 CPU 的 8 位元微電腦上進行軟體計時的辦

法，之所以針對 6502，係因爲以 6502 爲 CPU 而設計的微電腦，如 APPLE II，目前已可在市面上以極低的價格購置，並且，APPLE II 過去的流行使得各種功能的界面被發展出來，如 VIA-6522 並行輸出入界面卡、中文漢卡等，這使得甚至以中文爲材料的心理學實驗也可以很容易地達成程序自動化。

APPLE II 與 Commodore PET 兩種 8 位元微電腦同樣是以 R6502 作爲 CPU，但後者由於利用了 1/60 秒硬體岔斷進行累次計算，因此能在不影響程式進行的情況下做 1/60 秒精確度的即時 (real time) 計時，但在 APPLE II 裏則將此等硬體岔斷保留給使用者自行運用，本身沒有計時功能。因此若欲使用 APPLE II 進行實驗程序控制或反應之計時時，在 APPLE II 上建立計時功能就成了首要之務。

在 APPLE II 上加上計時功能的方式有三種，一種是以界面的方式插入所謂的時間卡，另一種方法是模仿 PET 硬體岔斷的方式，以 1/60 秒的硬體岔斷，透過一個岔斷程式計時 (Mitterer & Osborne, 1982)，第三種方法就是以迴圈演算做爲基本邏輯，並小心計算機器時圈 (cycles) (Leventhal, 1979) 數目而設計的軟體計時程式 (Price, 1979)。前兩種方法的優點是可作即時計時，但缺點是需作硬體的附加或修改，在實作時，計時單位往往也難以更精細，甚至，有些時候硬體的岔斷會影響輸出入的運作，例如，使用第二種方法時，磁碟機的讀寫會受到干擾。相形之下，軟體計時的方式雖無法作即時計時，但在精確及使用方便的條件下，就顯得較易爲實驗者接受。

Price (1979) 的軟體計時組合程式有兩個很重要的限制，使得實際的運用上很不方便，其一為反應的偵測只能針對鍵盤為之，換言之，只有透過鍵盤按鍵方能使計時程式停止；另一個限制是計時極限僅為 83.75 秒，換言之，如果反應時間超過 83.75 秒，將導致溢位 (overflow) 而使得計時不準。這些限制使得該程式之可用程度不够廣泛，因此，本文擬改良 Price 之程式，使具有原程式的優點，且滿足如下二條件：1. 反應的偵測標的可由高階之呼叫程式指定為主機內任一記憶位址，該位址之內容變化即可使計時停止；2. 計時極限可長至 6 小時。

程 式 演 算 法

甚至在一般常用而複雜的週邊界面 (如漢卡) 工作的狀態之下，APPLE II 記憶位址中的 \$0300 至 \$03FF (符號 \$ 表 16 進位制) 段仍保留給使用者的機器語言程式，因此可在這段位址置入表一的機器語言副程式，而不至影響高階程式或輸出輸入的運作。

表一中的組合 (及機器) 語言程式基本上採用了 Price (1979) 的迴圈計算邏輯與「延宕因子」(delay factor) 的觀念，但在設計上加入如下的構想：1. 保留兩個記憶位址做為目標位址指位器，存於此兩記憶位址之內含將代表目標位址之地址，一旦目標位址之內容改變，則計時程式將停止；2. 由 3 個字元組成計數器。如表一所示，\$0300、\$0301 (以 BYTEL, BYTEH 代表) 為目標位址指位器，\$0302 (以 DF 代表) 為延宕因子記錄位置，\$0303 (以 FG 代表) 內存溢位指示，\$0304、\$0305、\$0306 (分別以 T1, T2, T3 代表) 為計數器，位址 \$0307 (以 AS 代表) 則存放導致計時程式停止之目標位址內含值。計時程式自 \$0310 (即十進制之 784) 位址開始，當該程式被呼叫時，將按照下述流程進行：

步驟 0：設定初值，首先清除位址 FG、T1, T2, T3 之內值使皆為 0，並根據位址 BYTEL 與 BYTEH 之內容找到目標位址，將目標位址內之初值抄入位址 AS 內。

步驟 1：即 TM0 起，檢查位址 FG 內之值，若其值不為 0，則跳至步驟 3。否則執行步驟 2。

步驟 2：將位址 DF 內之值抄入 Y 暫存器

(register)，並執行 Y 暫存器遞減迴圈 (即 TM1 起之迴圈)，直至 Y 暫存器內之值為 0 時離開迴圈，並將計數器內之值加 1，位址 T1, T2, T3 分別代表三個以 256 制做為進位之計數器，T1 表示較低位計數器，T3 則表示最高位計數器，計數器加 1 之運算係從位址 T1 之內容加起，當加至 256 時形成進位，使位址 T1 內之值變成 0 而位址 T2 內之值則加 1，同理，當位址 T2 內之值增至 256 時，亦形成進位，使位址 T2 歸零而位址 T3 內之值加 1。程式在必要之處加入空運作 (NOP) 之延宕時圈，使進位與不進位所需之時圈數得以平衡，換言之，不管進位與否，每一計時迴圈所需之時圈數為固定值，因此所需時間也等長。若位址 T3 之內容增至 256 而造成溢位，則位址 FG 之內容值將增加 1。

步驟 3：程式自 TM6 起，找出目標位址，並比較目標位址之內含與位址 AS 之內含是否一致，若是，則跳至步驟 1。否則執行步驟 4。

步驟 4：將目標位址之內含值抄入位址 AS 內，並離開本計時副程式，回到呼叫之主程式。

如果在呼叫副程式前，將代表鍵盤活動的位址 (\$C000，請查 APPLE II 手冊) 置入位址 BYTEL 及 BYTEH 內，位址 BYTEL 內存低位元，位址 BYTEH 內存高位元，例如在 BASIC 語言下執行動作 POKE 768, 0: POKE 769, 192，則呼叫副程式後，在鍵盤上按鍵將使計時程式停止；同理，可以選擇其他 I/O 位址作為目標位址，使透過其他 I/O 設備之活動可以使計時程式停止。

仔細地計算計時副程式內所需時圈數，可以發現每次計數器增 1 所需之時圈數受到兩個因素影響，其一為延宕因子 (即位址 DF 之內含值) 遞減至 0 所需時間，另一為進位之判斷、目標位址內含值比較所需之時間，前者受到位址 DF 內含值大小之影響，每減 1 需時 5 個時圈，而後者為一固定值 (即 49 時圈)。此外，進入與離開副程式所需的固定動作需時 57 時圈。由於 R6502 微處理機之一個時圈約需時 9.7779×10^{-7} 秒，若令 TI 表計時器內累計之值，即

$$TI = 256 * 256 * PEEK(T3) \\ + 256 * PEEK(T2) + PEEK(T1)$$

表一 R6502 軟體計時之機器語言與組合語言程式

Hex (Dec) Address	Machine Language	Assembly Language	Comments
0300 (768)		BYTEL EQU 768	target byte pointer (Low)
0301 (769)		BYTEH EQU 769	target byte pointer (High)
0302 (770)		DF EQU 770	store delay factor
0303 (771)		T1 EQU 771	counter 1
0304 (772)		T2 EQU 772	counter 2
0305 (773)		T3 EQU 773	counter 3
0306 (774)		AS EQU 774	store content of target byte
0307 (775)		BIT0 EQU 775	
0308 (776)		BIT1 EQU 776	
0310 (784)		TIME EQU 784	
0310 (784)		ORG TIME	**start of timing routine**
0310 (784)	A5 01	LDA \$01	initialization
0312 (786)	8D 08 03	STA BIT0	find target byte
0315 (789)	A5 02	LDA \$02	
0317 (791)	8D 09 03	STA BIT1	
031A (794)	AD 00 03	LDA BYTEL	
031D (797)	85 01	STA \$1	
031F (799)	AD 01 03	LDA BYTEH	
0322 (802)	85 02	STA \$2	
0324 (804)	A9 00	LDA #00	set T1, T2, T3, FG to 0
0326 (806)	8D 04 03	STA T1	
0329 (809)	8D 05 03	STA T2	
032C	8D 06 03	STA T3	
032F (815)	8D 03 03	STA FG	
0332 (818)	A0 00	LDY #00	
0334 (820)	B1 01	LDA (1), Y	put content of target byte
0336 (822)	8D 07 03	STA AS	into AS
0339 (825)	AD 03 03	TM0 LDA FG	check content of FG
033C (828)	D0 20	BNE TM6	if not 0, jump to TM6
033E (830)	AC 02 03	LDY DF	delay factor decrement loop
0341 (833)	88	TM1 DEY	
0342 (834)	D0 FD	BNE TM1	
0344 (836)	EE 04 03	INC T1	increase content of counters
0347 (839)	F0 04	BEQ TM2	
0349 (841)	EA	NOP	
034A (842)	EA	NOP	
034B (843)	D0 03	BNE TM3	
034D (845)	EE 05 03	TM2 INC T2	
0350 (848)	F0 04	TM3 BEQ TM4	
0352 (850)	EA	NOP	
0353 (851)	EA	NOP	
0354 (852)	D0 03	BNE TM5	

表一 (續)

Hex (Dec) Address	Machine Language	Assembly Language	Comments
0356 (854)	EE 06 03	TM4 INC T3	
0359 (857)	D0 03	TM5 BNE TM6	if not overflow, jump to TM6
035B (859)	EE 03 03	INC FG	otherwise set FG to 1
035E (862)	B1 01	TM6 LDA (1), Y	get content of target byte,
0360 (864)	CD 07 03	CMP AS	compare with that of AS.
0363 (867)	F0 D4	BEQ TM0	if match, jump to TM0
0365 (869)	8D 07 03	STA AS	otherwise update AS
0368 (872)	AD 08 03	LDA BIT0	
036B (875)	85 01	STA \$01	
036D (877)	AD 09 03	LDA BIT1	
0370 (880)	85 02	STA \$02	
0372 (882)	60	RTS	exit to calling program.

，且令位址 DF 之內存值為 PEEK (DF)，則所得反應時間為：

$$R. T. = (TI * (5 * PEEK(DF) + 49) + 57) \\ * (9.7779E - 7) \text{ 秒}$$

。改變位址 DF 之內存值可以控制計時之精確度及計時極限，例如，若在主程式內設定位址 DF 之內含值為 255，則可以以 1.295 msec 之精確度測量約 6 小時內之任何時間長度，當然，若位址 DF 之內存值較小，則計時精確度將提高（最高為 5.28×10^{-5} 秒），但計時長度將相應減小。

在 BASIC 下之載入與執行

假設表一之機器語言副程式已存於以

SOFTIME 為檔名之磁碟檔內，則可以以 BLOAD SOFTIME 之指令載入，或以表二內 BASIC 程式所述之方式，將機器語言副程式植於記憶體中。

表三所示則是一個用以測試計時副程式的可執行高階語言程式，該程式載入表一之機器語言計時副程式於記憶體內，並定義計時之目標位址為鍵盤，延宕因子值為 255，之後令使用者按鍵代表計時開始，並按任一鍵代表計時結束，然後在螢幕上輸出反應時間。值得注意的是，若以 9.7779×10^{-7} 秒作為一個時圈所需時間的估計，則當計時長度增長為數小時後，將會有些微低估的現象，因此表三的程式中以 9.793×10^{-7} 秒作為一個時圈所需時間之估計，以便得到比 Price 更正確的估計。讀者可自行修改此程式，使適用於其特殊之研究需要。

表二 建立軟體計時機器碼之 BASIC 程式範例

```

10 DATA 165, 1, 141, 8, 3, 165, 2, 141, 9, 3, 173, 0, 3, 133, 1, 173, 1, 3, 133
20 DATA 2, 169, 0, 141, 4, 3, 141, 5, 3, 141, 6, 3, 141, 3, 3, 160, 0, 177, 1, 141
30 DATA 7, 3, 173, 3, 3, 208, 32, 172, 2, 3, 136, 208, 253, 238, 4, 3, 240, 4
40 DATA 234, 234, 208, 3, 238, 5, 3, 240, 4, 234, 234, 208, 3, 238, 6, 3, 208
50 DATA 3, 238, 3, 3, 177, 1, 205, 7, 3, 240, 212, 141, 7, 3, 173, 8, 3, 133, 1
60 DATA 173, 9, 3, 133, 2, 96
70 FOR I=784 TO 882
80 READ X
90 POKE I, X
100 NEXT
110 END

```

表三 軟體計時之 BASIC 測試程式

```

10 D$=CHR$(4)
20 PRINT D$ "BLOAD SOFTIME"
30 POKE 768, 0: POKE 769, 192: REM TEST KEYBOARD
40 POKE 770, 255: REM DELAY FACTOR
50 FG=771: T1=772: T2=773: T3=774: AS=775: CT=9.793E-7
60 PRINT "HIT ANY KEY TO START TIMING";: GET AS: PRINT
70 PRINT "HIT ANY KEY TO STOP TIMING";
80 CALL 784: GET AS
90 TI=256*(256*PEEK(T3)+PEEK(T2))+PEEK(T1)
100 RT=(TI*(5*PEEK(770)+49)+57)*CT: TS=STR$(RT)+" SECONDS "
110 IF PEEK(FG)>0 THEN TS=" OVERFLOW "
120 PRINT: PRINT: PRINT TS: PRINT: PRINT: GOTO 60

```

結 論

綜使 16 位元微電腦已逐漸取代 8 位元微電腦原來的地位，但在心理學的實驗室裏，以 APPLE II 為主之 8 位元微電腦仍然扮演重要的角色，某些牽涉歷程控制與儀器界面的實驗，仍然以採用 8 位元微電腦主機較為方便，且較節省程式設計時間。特別是就採用中文材料的心理學實驗而言，由於以商用為主要目標而設計的 16 位元機器常常忽略了有關呈現方式或繪圖能力方面的考慮，因此 8 位元機器有時反而較能滿足複雜而多變化的實驗設計要求。基於此，本文將 Price (1979) 之計時程式加以修改，使能更容易適合這方面研究的需要。本文所提之各程式皆已存於磁片之上，若需本文所述程式者，請自備磁片，逕洽本文作者進行複製。

參 考 文 獻

- ALLEN, J.D. and NICORA, B. Interfacing PET/CBM with Coulbourn's DynaPort for experimental control and data acquisition. *Behavior Research Methods, Instruments and Computers*, 1984, 16, 492-494.
- ANDERSON, D.E. PSYCHO-STATS PC: A statistical package for the IBM PC. *Behavior Research Methods, Instruments and Computers*, 1984, 16, 482.
- CAMPBELL, D.J. A spectral-analysis program with graphics for the CBM/PET/C-64 microcomputers. *Behavior Research Methods, Instruments and Computers*, 1984, 16, 484-485.
- COULOMBE, D. Canonical correlation analysis: A BASIC system for microcomputers. *Behavior Research Methods, Instruments and Computers*, 1984, 16, 475-476.
- COLOMBE, D. Principal component analysis: A BASIC program for microcomputers. *Behavior Research Methods, Instruments and Computers*, 1984, 16, 477-478.
- DURSO, F.T. A subroutine for counterbalanced assignment of stimuli to conditions. *Behavior Research Methods, Instruments and Computers*, 1984, 16, 471-472.
- LEVENTHAL, L.A. 6502 Assembly language programming. Berkeley, California. Osborne/McGraw-Hill Inc., 1979.
- LOFTUS, G.R., GILLISPIE, S., Tigre, R.A. and NELSON, W.W. An Apple II-based slide-projector laboratory. *Behavior Research Methods, Instruments and Computers*, 1984, 16, 447-453.
- MITTERER, J.O. and OSBORNE, B. Improved real-time control on the Apple by mimicking an on-board 60-Hz clock. *Behavior Research Methods and Instrumentation*, 1982, 14, 412-413.
- OSAKA, N. A microprocessor-based real-time BASIC laboratory: A pulse motor-controlled visual stimulator. *Behavior Research Methods and Instrumentation*, 1979, 11, 549-552. (a)
- OSAKA, N. A microprocessor-based real-time BASIC random interstimulus interval generator with different probability density functions. *Behavior Research Methods and Instrumentation*, 1979, 11, 581-584. (b)
- OWINGS, R.A. and FIEDLER, C.H. Measuring reaction time with millisecond accuracy using the TRS-80 microcomputer. *Behavior Research Methods and Instrumentation*, 1979, 11, 589-591.
- PRICE, J.M. Software timing for 6500 series microcomputers. *Behavior Research Methods and Instrumentation*, 1979, 11, 568-571.
- REED, A.V. Microcomputer display timing: Problems and solutions. *Behavior Research Methods and Instrumentation*, 1979, 11, 572-576.

(1985年5月1日收件)

**A MODIFIED VERSION OF THE PRICE'S R6502
SOFTWARE TIMING PROGRAM FOR
PSYCHOLOGICAL EXPERIMENTS**

JEI-TUN WU

National Taiwan University

ABSTRACT

The software timing program developed by Price (1979) was modified so as to extend its range of applications. The modified version was also written in assembly language, and had the following advantages over Price's original one: (1) The input entry is acceptable not only from keyboard, but also from any I/O device with the corresponding byte address being pre-defined from the calling programs, (2) the maximum time length for timing can be extended to about 6 hours. The object code program and some other BASIC programs for testing it have been stored on a diskette and are available for reproductions.