

英雄一號機器人

高級程式設計與界面

原著者 Mark J. Robillard

譯者 陳廷光



英雄一號機器人

高級程式設計與界面

原著者

Mark J. Robillard

譯者

陳廷光

臺灣電力公司電腦課長

東華書局印行



版權所有・翻印必究

中華民國七十三年六月初版

大專
用書 英雄一號機器人

定價 新臺幣壹佰捌拾元整

(外埠酌加運費匯費)

譯者 陳 紅 光

發行人 卓 鑫 森

出版者 臺灣東華書局股份有限公司
臺北市博愛路一〇五號
電話：3819470 郵撥：6481

印刷者 合 興 印 刷 廠

行政院新聞局登記證 局版臺字字第零柒貳伍號
(73015)

序 言

恭喜！您現在所玩的，是個人用機器人中，少數最先進者之一。您所購買的 Heath ET-18 HERO 機器人訓練器，不但將增加您的知識而且將幫助您踏入機器人的領域。

筆者裝配好機器人並完成檢驗後，就發現急需更多的知識。隨機附送的技術手冊在操作方面，確已包羅相當豐富的資訊，但當使用到 HERO 時，却發現有待了解者仍然很多（書到用時方恨少！）。

本書的目的期在技術手冊之外，提供完整的硬體、軟體，及應用實例。但本書絕不是完整的，因為如何使您的機器人如虎添翼，完全局囿於個人的想像力而已。

第一章至第三章主要是針對沒有購買訓練課程而又需要機器語言和機器人語言之入門介紹者而寫的。這部份對 6808 的指令集、機器人語言，以及若干程式技巧（使用 HERO 作實例）等，有相當深入的討論；此外尚有幾個程式也羅列在內。本書背後附有 HERO Programmer's Reference Card，可供讀者方便參考。

第四章闡述 HERO 內部的韌體。本章除繪出 ROM 的記憶使用圖表外，也列出可供程式師使用的公用程式韌體。

第五章專注於程式設計課題。次第登場的有：HERO 操作系統的深入分析、如何加強操作系統功能的指引、另一種交替程式語言（可燒成 ROM 插於擴充插座）的設計，以及這種語言的程式設計方法。最後列有以培基寫成的遙控中介程式，以及支援培基指揮機器人動作的一個套裝副常式。

最後一章探討硬體的加強。加給 HERO 基本電路的中介，無論是在 CPU 板上或外加板，都是本章的範疇。為保持前後一致起見，裝配細則完全仿照機器人的安裝手冊編寫。我們在這一章分別討論到記憶

的加裝（CPU 板上及外加板二者）、遙控無限電、及光 / 音控制電路等。然後它們就可以應用於標準的培養控制程式。

機器有了，也沒有什麼好顧忌的，何不立刻就開始呢！

MARK J. ROBILLARD

目 錄

序 言	III
第一章 認識6808控制系統之功能	1
1-1 引言	1
1-2 6808 之內部結構	2
1-3 中斷與向量	10
1-4 埠地址之分派	15
第二章 6808之程式規劃	31
2-1 引言	31
2-2 定址模式	31
2-3 資料傳遞指令	34
2-4 算術與邏輯指令	40
2-5 跳躍與分支指令	43
2-6 資料測試與狀況碼指令	46
2-7 索引、堆疊、和中斷處理指令	48
第三章 機器人解譯器與機器人語言	50
3-1 移動命令	50
3-2 語言及感測指令	62
第四章 韌體與RAM之使用	76
4-1 核心執行常式	76
4-2 ROM分配圖	83
4-3 RAM變數	91

4-4 機器人公用程式	91
第五章 軟體之增強	94
5-1 擴充HERO之操作系統	94
5-2 增強機器人功能	102
第六章 硬體之增強	105
6-1 6808 之中介	105
6-2 HERO 1 電路板之接線	114
6-3 擴充記憶	125
6-4 增強程式輸入	136
6-5 遙控資料登錄	144
6-6 碰撞與接觸感測	162
6-7 擴充中介	168
6-8 串聯中介	179
附 錄	184

第一章

認識6808控制系統之功能

1-1 引言

本章和下一章是為沒有使用過 6800 系列微處理機的讀者而寫的。作者假設讀者已有一般電腦的知識，且對它們的程式設計已相當熟悉。對於尚無此基礎的讀者，坊間有許多適合這類題材的好書及自修課程可供研讀，本書將不涵蓋這些題材。

蘊涵於本書的是，有關 6800 適用於 HERO 機器人方面的程式設計。有經驗的程式師應能從本書所舉的實例，獲得有關於機器人的更多知識。對於 HERO 的擁有者而言，HERO 的控制器及其內部工作和程式方法等，除了基本文件（隨機附送者）外，仍有甚多可資學習者。

包括於本書的尚有技術手冊所刊載的機器人語言命令。但是讀者將可發現本書在每一道命令之後，附有相當豐富的使用者為導向的資訊，而不僅僅是參數的敘述而已。此外，使機器人“循規蹈矩”所需的一些技術與巧思，也涵蓋在內。

業已精通 6800 微處理機和機器人埠 (Port) 地址的熟手，可逕自跳到第三章開始探討機器人的語言。至於其餘的所有讀者，歡迎加入認識 6800 的行列。

1-2 6808 之內部結構

HERO 外殼內部的“腦”是 6808 微處理機。6808 是 6800 家族的一員，原為 Motorola 半導體產品公司的產品，現已有多家公司提供貨源。6808 是由 6802 裝置發展而成的。6802 在晶片內加入時脈產生器，這是 6800 晶片所無者。此外它還加入 128 拜 (byte，一般譯作位元組或數元組，本書取音譯) 的讀寫記憶，供堆疊指標 (stack pointer) 或一般存取用。可是，6802 的初期成品中，其記憶區域的收成率 (yield) 却很低。事實上，收成率糟得使 Motorola 決定把它當成只具有時脈產生器而不含記憶的商品上市；當時的商品編號是 SC 44301P。但是，由於使用量增加，該公司最後乃決定把它併入其生產線行列，於是，改變其產品編號而成為 6800 家族的一員，6808 於焉誕生。

與其他微處理機類似的是，6808 亦使用 16 秘 (bit，一般譯作位元、數元、或筆，本書採音譯) 寬的地址匯流排及 8 秘寬的資料匯流排跟其他的裝置通訊。其最大定址範圍為 65536 拜 (或 64 K)。它只要 +5V 一種電源就能工作，且如前所提者，晶片內部已製有時脈產生器。HERO 的主時脈頻率即係由接於 6808 的晶體輸出衍生而得。質言之，CPU 板上所用的是 3.579MHz 的彩色簪色 (color burst) 信號。6808 晶片本身把此頻率除以四，產生週期為 1.11 微秒的主振蕩信號；此週期稱為處理器週期時間。每一個指令的執行時間都以週期為單位，因此程式的執行時間可用諸指令執行週期總數來乘以週期時間 (1.11 μ s) 而得。我們將在討論個別指令時，再回到此題目進一步詳談。

6808 微處理機的內部結構如圖 1-1 所示。主要特徵如下：

1. 兩個累積器 (accumulator) A 及 B

2. 索引暫存器 (X)
3. 程式計數器 (PC)
4. 堆疊指標 (SP)
5. 狀況碼暫存器 (CC)
6. 72 個指令
7. 六種定址模式
8. 三種中斷

爲更深入了解 6808 起見，讓我們再個別詳參以上諸特徵。

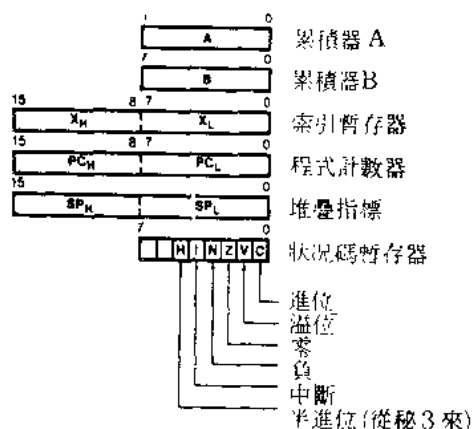


圖1-1 用來控制 HERO的6808微處理機的内部結構

1-2.1 累積器(A和B)

所有進出 6808 微處理機的資料，均經過 A 或 B 其中一個累積器。大多數的資料搬移指令，都會使二者互相複製（抄襲）。它們也作為算術與邏輯演算結果的暫時儲存場所。

1-2.2 索引暫存器(X)

此暫存器為 16 位寬。6808 通常利用此暫存器計算或儲存記憶地址。有一種稱為“索引”(indexed)的定址模式，允許使用此暫存器來直接定址一記憶位置。隨著本書的進行，讀者將會發現此種定址方法的使用非常普遍，因而十分重要。

1-2.3 程式計數器(PC)

如稍早所提者，6808 微處理機能定址到 65536 個記憶位置。以二進位法表示，等於是 16 位寬。因此程式計數器為 16 位寬。程式師無法直接觸及或控制此暫存器，只能用跳躍(Jump)或分枝(Branch)指令改變其內容，從而導引處理器的運算步向規定的記憶位置。

1-2.4 堆疊指標(SP)

堆疊指標也定址記憶，所以也為 16 位寬。它指向一記憶區域，俾於副常式呼叫或中斷呼叫時，保存現有程式計數器值、索引暫存器值。A/B 二累積器之值及狀況碼暫存器。實際上，存於堆疊指標的地址值是指向堆疊的第一個地址，然後諸值係依序存入第一個地址之前的區域，如表 1-1。

俟諸值存入堆疊後，堆疊指標將含 PC 的地址值減 7，或即狀況碼暫存器的堆疊地址再減一。副常式本身尚可呼叫別的副常式，而形成多層結構。呼叫的層數取決於堆疊可用的堆疊記憶容量而定。

當副常式執行完畢而回返程式時，再依照存入的相反次序從堆疊彈出諸值，恢復原狀態給程式。換言之，每從堆疊讀出一值，堆疊指標即自動加一。

表1-1 堆疊地址與內容

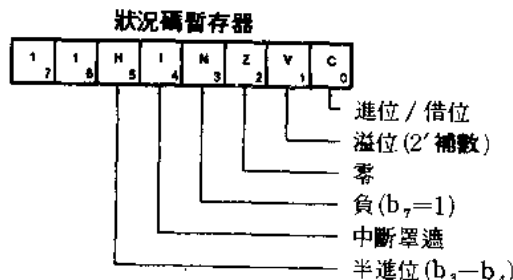
堆疊地址	堆疊內容
堆疊指標的內容 - 0	PC 的低位拜
堆疊指標的內容 - 1	PC 的高位拜
堆疊指標的內容 - 2	X 的低位拜
堆疊指標的內容 - 3	X 的高位拜
堆疊指標的內容 - 4	A
堆疊指標的內容 - 5	B
堆疊指標的內容 - 6	CC

表1-2 狀況碼暫存器諸秘說明

CC Bit	定 義
進位 - 借位(C)	執行加指令時，若累積器產生進位，進位旗誌即設為 1，否則即復歸為 0。
溢位(V)	若算術演算（2'補數）發生溢位時，V 秘設定，否則清除歸零。
零(Z)	當算術演算結果為 0 或當比較的兩數相等時，Z 秘將設定，否則 Z 秘為 0。
負(N)	當運算結果的秘 7 設定時，此秘即設定。故此秘用來指示負數的狀況。
中斷罩遮(I)	當 I 設定時，將抑制 IRQ 腳所發生的中斷請求。此中斷罩遮秘分別由 SEI 指令或 CLI 指令置定或清除。
半進位(H)	在執行 ABA、ADC、或 ADD 指令時若秘 3 產生進位至秘 4 時，即設定此秘，否則為 0。

1-2.5 狀況碼暫存器(CC)

程式在執行期中，常會執行資料的比較、算術運術、或其他的測試指令，俾獲得演算結果的狀態資訊；諸如兩數是否相等、兩數的大小關係、相加後有否進位等。狀態資訊係容納於狀況碼暫存器。有若干指令即係使用此暫存器的資訊，以作出邏輯決定。表 1-2 是暫存器諸秘的說明，諸秘的相關位置繪如圖 1-2。



狀況碼之置定與否隨先前運算結果而定

ABA: A = 1000 1000 B = 1000 1000	A + B = $\hat{1}0001\ 0000$	H = 1, Z = 0 C = 1, N = 0 V = 1
DEC A A = 0000 0001	A - 1 = 0000 0000	Z = 1, N = 0
LDA A = #80	A = 1000 0000	N = 1, Z = 0, V = 0
COM A A = 1000 0000 OR DEC A	$\bar{A} = 0111\ 1111$	N = 0, V = 0, C = 1, Z = 0 N = 0, V = 1, C = UNCHANGED, Z = 0
ABA: A = 1000 0010 = -126 ₁₀ B = 1000 0010 = -126 ₁₀	A + B = $\hat{1}0000\ 0100$ = +4 ₁₀	V = 1, Z = 0 C = 1, N = 0 H = 0

- 負 (N)** 當任何運算結果的秘 7 置定時，狀況碼暫存器的 N 秘 (秘 3) 即置定，反之，N 秘清除。
- 中斷罩遮 (I)** 當此秘置定 (I = 1) 時，將抑制 IRQ 中斷。若 I = 0 則處理器可被低電位之 IRQ 中斷。I 秘由 SEI 指令置定；由 RTI (若中斷前 I 秘係 0) 或 CLI 指令清除。
- 半進位 (H)** 在執行 ABA、ADC 或 ADD 指令時，若有進位從秘 3 進至秘 4，則狀況碼暫存器的半進位秘 H (秘 5) 將置定，反之則 H 秘清除。

註：狀況碼暫存器所容納之資訊，是最後影響狀況碼暫存器之指令的計算結果。

圖 1-2 6808 狀況碼暫存器一覽

在討論 6808 指令的敘述期間（第二章），將會常常參考到狀況碼暫存器。

1-2.6 中斷

如前所提 6808 MPU 有三種中斷：SWI、IRQ 及 NMI。

IRQ — 此是由硬體引發的中斷請求。亦即，當晶片的 IRQ 針腳感測到低電位時，6808 將檢查狀況碼暫存器的中斷罩遮秘，若此秘為 1，則不發生任何事，微處理機將繼續其原有工作。若此秘為 0（表示允許中斷），則程式計數器、索引暫存器、A B 累積器，及 CC 將依次存入堆疊。在此時，程式計數器將載入 FFF8 之值。該記憶位置含 IRQ 中斷服務常式開始地址之高位拜；而低位拜則含於位置 FFF9。此二拜組成向量位置指向服務常式之開始地址。服務常式的最後一個指令：從中斷回返（RTI），將從堆疊依序彈出諸值，使諸暫存器得以復原，一如副常式之呼叫程序。在處理中斷常式的期間，中斷罩遮秘將自動設定，俾在本次中斷服務完畢前，抑制其他的中斷請求。俟 RTI 指令執行後，中斷罩遮秘將自動回復到中斷發生前之原值。

NMI — 另一個外部中斷輸入是非罩遮中斷，它亦由中斷引發。顧名思義，6808 是無法罩遮或阻隔此中斷的。NMI 中斷服務常式之向量係存於 FFFC（高位拜）和 FFFD（低位拜）兩位置，其處理與 IRQ 請求相同。

SWI — 此是 6800 服務的最後一個中斷請求；又稱為軟體（SW 即係由 Soft Ware 而來）中斷。此請求是由 SWI 指令的執行而引發的。它也是非罩遮的；亦即它不會被 6808 MPU 所忽視。SWI 廣泛地應用於 HERO 機器人語言常式中。其向量地址是 FFFA（高位拜）及

FFFB (低位拜)。

此三種中斷的流程綜合如圖 1-3。從圖中上方中央靠左處，可

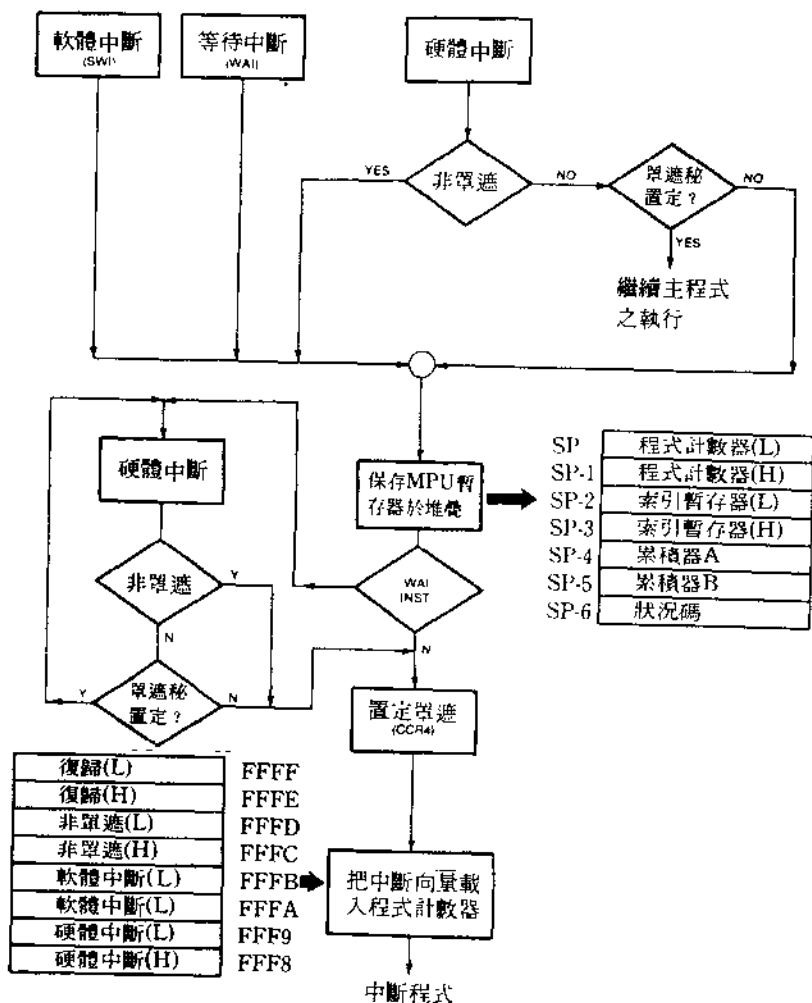


圖1-3 6808中斷結構之決策與指令流程圖

發現到一只標名為 等待中斷 (wait-for-interrupt, 簡作WAI) 的方盒。此亦指令之一。WAI 把處理器置於中止 (halt) 狀態, 此狀態只能靠硬體或外部中斷才能跳出。

6808 MPU的另外一個類似於中斷的外部請求, 是晶片上的復歸 (reset) 接腳。與一般微處理機類似者, 當引發復歸時, 所有一切都重新起始, 且程式將從一已知點開始擷取。復歸時將發生下列動作:

1. 中斷罩遮秘置定。
2. FFFE 位置的資料載入 PC (高位拜)。
3. FFFF 位置的資料載入 PC (低位拜)。
4. 從新的 PC 位置開始執行。

注意到復歸動作也抑制 IRQ 輸入端的中斷請求, 因此若欲使用中斷, 則在復歸之後應藉 CLI 指令清除中斷罩遮秘。

以上是有關 6808 的概略情形, 摘要如圖 1-4。在下一章討論

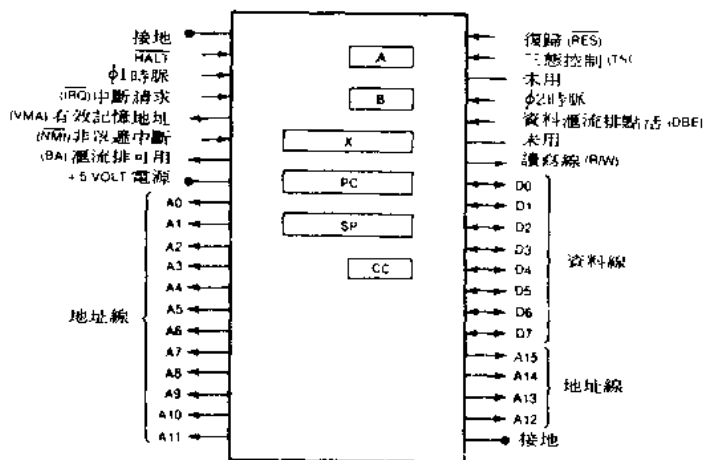


圖1-4 程式師眼中的6808模型

6808 指令及各種定址模式的期間，還會談到諸暫存器間的相互作用關係。

1-3 中斷與向量

前面我們討論到有關 6808 微處理機的中斷結構，以及當 IRQ 或 NMI 輸入起作用時，處理器的反應。我們也簡略提到軟體 SWI 中斷指令的使用及處理器的反應。下面將更深入探索 6808MPU 遇到中斷後的詳情。這一節的討論需要具有組合語言程式的知識。尚無組合語言程式經驗的讀者，或許可暫時先跳過這一小節。

1-3.1 向量 (Vectors)

如前所述，當 6808 微處理機受到外界的某種刺激時，具有組成向量以指向某一程式區域的能力。在中斷的情況下，6808MPU 具有暫時中止一作業以執行較高優先工作的能力。此即為“向量中斷”。另有一個外部輸入也會引發向量動作，而指向別處之程式。HERO 的復歸輸入除接自電源上升偵測電路外，尚接一按鈕開關。當開機（送電）或使用者按下該按鈕開關時，HERO 的 6808 將分從 FFFE 及 FFFF 兩位置擷取兩拜的資料，它們所組成的向量告訴 HERO 從 F37E 位置開始執行程式。這就是機器人開始動作之前的起始。在 F37E 位置，我們將可發現下列指令：

F37E 8E 0F DF	LDS	0FDF (power stack)
F381 8D 28	BSR	INITROB
F383 86 C2 80	LDAA	sleep port
F386 85 02	BITA	
F388 26 51	BNE	INITEXEC

第一個指令把 0FDF 載入堆疊指標。0FDF 地址只距 RAM 頂端幾