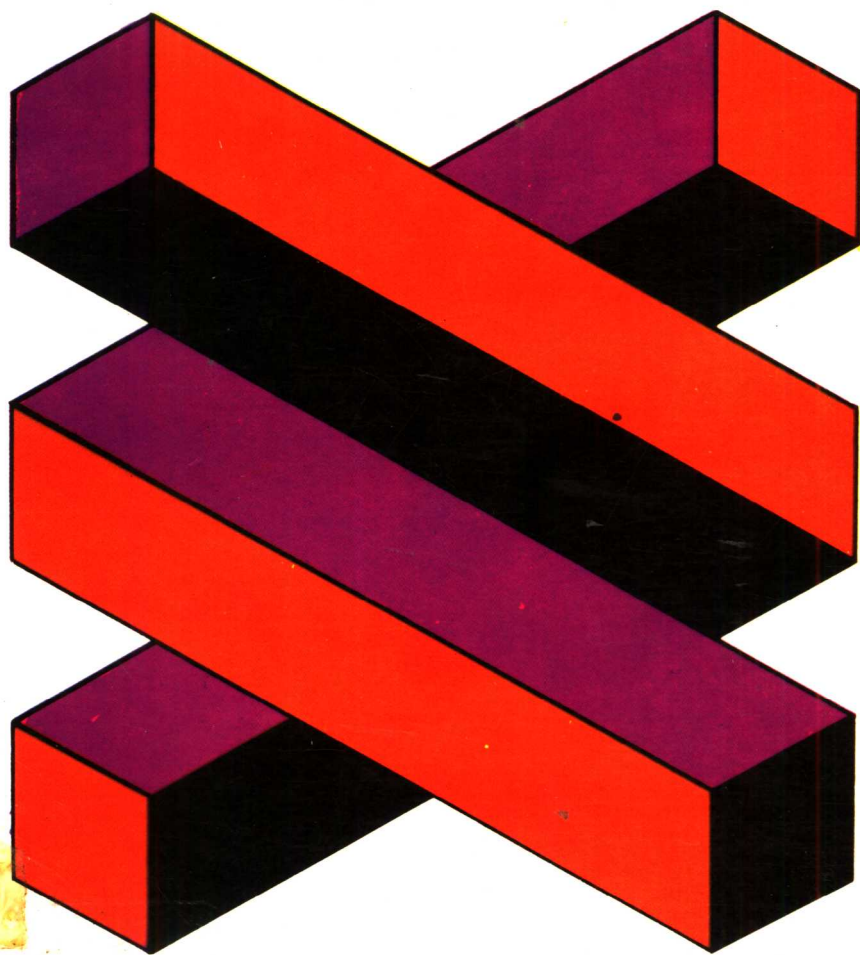


電腦與自動控制概論

COMPUTERS AND THE CYBERNETIC SOCIETY

MICHAEL A. ARBIB 原著 / 王嘉陵 ■ 編譯



羅拔書局印行

773
1237

電腦與自動控制概論

王嘉陵 ■ 編譯

羅拔書局印行

電腦與自動控制概論

編 者：王 嘉 陵

出版 兼：羅 拔 書 局
發行

澳門大馬路 381 號二樓 F 座

印 刷 者：振 興 印 刷 公 司

澳門龍崙 152 號地下

定價港幣 \$ 25.00

序

電腦的發展可說是日新月異，其應用範圍亦逐漸擴大，遍及農、工、商、學、軍、政、醫各界，用途已有數千種。幾乎任何繁雜問題，只要能寫成適當的程式，都可交由電腦處理，使問題立即迎刃而解。像銀行帳務、所得稅等皆可由電腦處理，不僅迅速、便捷，而且錯誤率極低，真可謂是劃時代的最重大發明。

由於電腦具有無窮的潛能，工業先進國家莫不競相探討，戮力開發，務期使電腦發揮最大效能。現在國際間甚至以電腦的發展情況作為衡量一個國家現代化程度的砝碼。而我中華民族素質優秀，歷史悠久，過去在科技上也有輝煌的成就，如今我們能否迎頭趕上與其他國家並駕齊驅，或躍居首位，端視我們自身的努力程度而定。

現代社會已邁入自動化的境地，以機械代替人工已觸目皆是。我們從某些機械的工作方式中可發現到它與人體的工作方式頗有類似之處，因此本書特將動物與機械的運動作一比較，以期使讀者能自其中瞭解人與電腦所發生的密切的交互作用，以及在自動控制社會中，如何將大量的資料，運用電腦處理，以增進人類生活的水準。所以讀者不僅可從本書中獲知電腦與自動控制的理論；更可洞悉電腦對未來社會所造成之影響。

筆者曾在國內外從事電腦工作多年，亦曾在大學教授電腦課程，深感此門科學對社會發展之助益極大。這種趨勢如繼續發展下去，無疑將會把現代社會帶入更高層次的境界。故特編譯本書，將電腦與自動控制作一具體而詳盡的介紹，使讀者能深切體會到在當今的自動化社會中，電腦與人類已達到息息相關的程度。並以由淺入深的方式和簡潔的字句寫出，使讀者易於瞭解，進而增進學習的興趣。

本書係編者利用公餘之暇所撰，倉促付梓，漏誤之處在所難免，尚祈各界先進不吝指正。

王嘉陵 謹識

HJ504/04

原 序

電腦對人類生活所產生的影響已日益增加，它可以使資訊得到溝通，儲存或檢索，也可以使各行各業的複雜問題得以解決。像我們的帳單、銀行帳戶、所得稅等皆可由電腦處理。政府機構或信用卡公司可以把國民的資訊儲存在資料庫裏，這種資料庫自然也是以電腦為基礎。電腦教學則對我們的學習方式有極大的影響，自動化對我們的工作方式也造成相當大的改變。政府機關常把電腦應用在經濟策劃上，環境保護者也利用電腦來研究生態系統。將微型電腦應用在特殊目的上已有增加的趨勢，例如數位電子表、汽車上點火控制等，真是不勝枚舉。

本書的目的就是要解釋電腦對現代人類所產生的影響，並且提供一些知識，使大家對電腦在未來之應用有一明確之認識。所以在本書中，我們將解釋電腦是什麼及如何進行程式設計，換句話說，我們如何給予電腦一些指令，使它能執行上述的各種功能。當我們對此有一基礎性之瞭解後，就可進一步探討電腦的各項應用，並能瞭解電腦在何處可作更廣泛的運用，在何處應加限制或控制。

本書既可用於自修，亦可充為各種電腦課程的教材，當作大專的電腦概論課程的用書非常適宜。

自動控制 (Cybernetics) 這個名詞的現代意義是美國麻省理工學院的諾伯韋納 (Norbert Wiener) 在 1948 年所引進。他發現某些機械的工作方式與人體的工作方式頗有類似之處。例如蒸氣引擎中有一項裝置，稱為調速機 (Governor)，可用於防止引擎運轉的太快或太慢。但是科學家也發現動物的腦部可以控制心肺的機能，使血液中始終保持適量的氧。韋納發現動物與機械間的比較關係很值得作一番系統性研究，於是他就把以動物與機械為比較，研究控制及通信的學問稱為自動控制學。這個名詞源於希臘文，原意為掌舵者，也就是控制船行進方向的人。調速機則源於拉

丁文，亦為控制之意。自從 1948 年以來，很多科學家都把自動控制原理應用在通信及控制上，所考慮的也不再是一個動物或一件機械，而是整體的人、動物、或機械。

在自動控制的社會中，由於有大量的資料受到處理，人與電腦因而發生密切的交互作用。本書就是要幫助大家瞭解這種交互作用，以及如何運用它來增進生活的水準。為達成此目的，我們必須要研究電腦問題，像我們如何使電腦執行一件指定工作等；也要研究社會及政治問題，像我們應讓電腦執行那些工作等。例如政府為了要處理所得稅、徵召適當人員服役等等，都會需要有關每個國民的資訊，當然這些資訊的份量一定相當可觀。這些資訊可以利用電腦儲存在一個龐大的電子檔案系統中，稱為資料庫。由於這種資料庫的存在，導致不少的政治及社會問題：像政府有權儲存那些資訊到資料庫中？每個國民如何更正或消除有關他本身的資訊？誰有權取得這些資訊？由誰來決定這些事情？在回答這些問題之先，我們必須要對下列問題的答案有一基本的瞭解：資料如何發生功能？如何了能保證只有已授權的人得以接取資料庫內的資訊？

在閱讀本書時，有一點必須要切記：電腦發展迄今不過只有三十年的歷史，但已使我們覺得經歷了極大的改變。但在三十年後，我們又將會發現在本書中所描述的一切已經趨於普遍，甚至於已經落伍。因此讀者從本書中所應學到的不只是目前的自動控制社會狀態而已，更重要的是瞭解電腦對我們未來的生活型態會造成多劇烈的改變。

邁可艾比柏

(Michael A. Arbib)

目 錄

第一章 電腦概論(上)

1 · 1	程式之構想	1
1 · 2	資訊的移動與儲存	18
1 · 3	為機器人領路	38
1 · 4	文句處理	56

第二章 電腦概論(下)

2 · 1	翻譯程式、作業系統、及分時	71
2 · 2	圖形與網路	84
2 · 3	日益縮小的機器	97
2 · 4	撰寫及修訂龐大的程式	104

第三章 模擬複雜的系統

3 · 1	模擬之構想	117
3 · 2	掠食者與犧牲品	131
3 · 3	世界之模擬	145

第四章 資料庫之興起

4 · 1	資料庫的管理	163
4 · 2	電腦、金錢、及犯罪	176
4 · 3	資料庫與私有權	185

第五章 人工智慧

2 電腦與自動控制概論

5 · 1 頭腦與電腦	193
5 · 2 能觀看及策劃的機器	206
5 · 3 能瞭解自然語言的機器	222
5 · 4 電腦在戰爭及和平方面的用途	236

第六章 學習與工作

6 · 1 電腦在教育上的應用	245
6 · 2 自動控制社會中的資訊流動	262
6 · 3 自動化的影響	270

第七章 網路與政治

7 · 1 策劃用之網路	283
7 · 2 電腦時代的民主政治	295
7 · 3 地球的主腦	304

第八章 機器語言

8 · 1 硬體如何發生功能	315
8 · 2 程式設計與編譯	336

索引	353
----	-----

第一章 電腦概論（上）

1 · 1 程式之構想

1 · 1 · 1 為簡單的機器人設計程式

電腦可以從事很多工作，像印製薪津支票，導引機器人（Robot）在室內行走，或是把太空船由月球收同等。對每件工作而言，必需要給電腦一個適當的程式（Program），告訴它這件工作應如何執行。當程式撰寫完成之後，就整體饋入電腦之中，電腦即能以高速執行其中的指令（Instructions），無需等待這些指令一一輸入。電腦所以有這種能力，是因為它有一個記憶體（Memory），整個程式可以儲存其中。每當它完成一個指令之後，它就自動由記憶體中取得下一個指令。

我們所以要使用電腦，是因為它的速率非常驚人，每秒鐘可以執行數百萬個，甚至數十億個運算。但是電腦完全遵從程式的指示，執行每一步的運算，然後使整個工作完成。這似乎造成一個很可怕的問題；我們如何趕上這部機器呢？我們如何才能給它數百萬個指令，使它一直保持忙碌？其實整個的概念完全建立在一句話上：「如果你第一次沒成功，那就再試一次。」

大部份的程式都含有迴路（Loop），也就是一序列的指令，電腦必須一再重複地執行它，直到某部份的工作完成為止。帶有迴路的程式也許只含有數百個指令，但它一旦進入電腦之後，卻可控制電腦相當長的時間。

我們現在要撰寫一串指令，告訴一個簡單的機器人如何走到門口。藉此我們可以瞭解一個帶有迴路的簡短程式如何能控制電腦的行動。

假設有一個機器人（圖 1 · 1 · 1）站在室內某處，它離四面的牆

2 電腦與自動控制概論

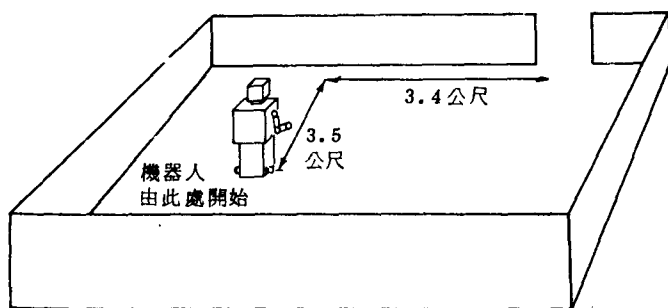


圖 1.1.1 機器人面對一堵牆，此牆之出入口在機器人之右手邊。

都有些距離，但面對其中某一面牆。在這面牆的某處有一出入口，目前它位於機器人的右手邊。這個出入口只是牆上的一個缺口而已，並未真正有個門存在。這個機器人可以轉動它底部的輪子，每次前進一步，其距離為 1 公厘。它也可以將一側的輪子固定，轉動另一側的輪子，達成轉彎的目的。由於這個機器人沒有眼睛，它並不知道出入口在那裏，所以無法直線移動過去。目前它只能先觸摸到一面牆之後，再辨別牆上的出入口在何處。

我們要撰寫一個程式，也就是一序列的指令，使其能儲存在電腦的記憶體中。因此機器人自動遵從這個程式的指示，由目前位置移動到出入口。假定我們知道機器人距離它面對的牆有 3.5 公尺，由「碰撞點」到出入口有 3.4 公尺，我們可以撰寫一個很簡單的程式（但在實際上相當繁瑣），使其完成這件工作。我們把這一系列寫在左邊，註解（Comments）寫在右邊，這些註解是用來說明這些指令的用途。

程式	註	解
前進一步	這個指令要重複 3500 次（因為 3.5 公尺等於 3500 公厘）使機器人由原地走到牆邊。	
前進一步		
前進一步		
前進一步		

右轉 90° 當它到達牆邊後，機器人必須向右旋轉 90° 使其能再沿牆前進。

前進一步	}	這個指令要重複 3400 次，使機器人沿牆走到出入口。
前進一步		
:		
前進一步		
前進一步		

在此程式中有 6901 個指令，如果要輸入到電腦的話，需要極長的時間。看起來把機器人推到門口比撰寫程式還容易一些！

此外我們還面臨另外一個問題。如果我們不知道機器人的起點，又當如何呢？或是我們想由其他地點開始，應該怎麼辦呢？如果每次都要撰寫一個程式，那未免太麻煩了，因此我們要撰寫另一種程式。對於這個機器人，我們實際要求它做三件事：

第一階段：直行到牆邊

第二階段：右轉 90°

第三階段：直行到出入口

我們把每一階段稱為是機器人應達成的次目標（Subgoal），當這些次目標逐個完成之後，它的最後目標也就達成。

現在我們來看看如何撰寫一個程式，使機器人能由室內任何一點開始，都能走到出入口。不過要記住一點，我們仍假設它面對有出入口的牆，而且出入口一定是在它右手邊某處。因此第一階段可分解成下列動作：

前進一步

測試：碰到牆了嗎？

假如答案為「否」：再前進一步

假如答案為「是」：跳到第二階段

為了使測試發生功能，機器人身上必須要有一個裝置，例如一個按鈕，當它碰撞到前方東西的時候，就會發出一個信號給電腦。

在圖 1 · 1 · 2 中就是上述動作的流程圖（Flow diagram），其中

4 電腦與自動控制概論

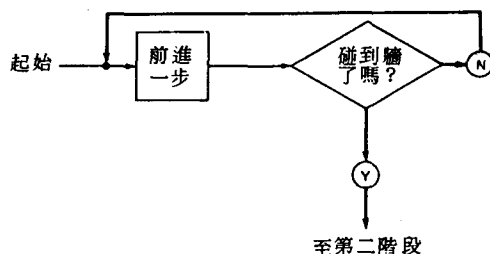


圖 1.1.2 第一階段的流程圖：「保持前進，直到牆邊為止」

每一方塊代表一個運算 (Operation)，每一菱形代表一個測試。菱形有兩個出口線，一個標記為 Y，代表測試結果為「是」；另一個標記為 N，代表測試結果為「否」。由於方塊是代表一個運算，所以只有一個輸出線，在此例中，它使電腦進入測試狀況。由於在第二階段中並無測試，所以它的流程圖



只有一個方塊而已 (見圖 1.1.3)。

圖 1.1.3 第二階段的流程圖：「右轉 90°」

第三階段的流程圖 (圖 1.1.4) 與圖 1.1.2 幾乎完全一樣，爲了要執行這部份的程式，機器人的側面必須要有發出信號的裝置，告訴電腦它在側面是否觸摸到牆。當測試的答案為「否」時，表示機器人已達成走到出入口的目標，所以它不再碰到牆。在此階段，電腦受到指示，由目前這個程式退出。它可能等待進一步的指示，也可能繼續遵從另一程式的指示，如果這個程式已儲存在它記憶中的話。

把這三個圖合併在一起，我們可以繪出整個過程的流程圖 (圖 1.1.5)

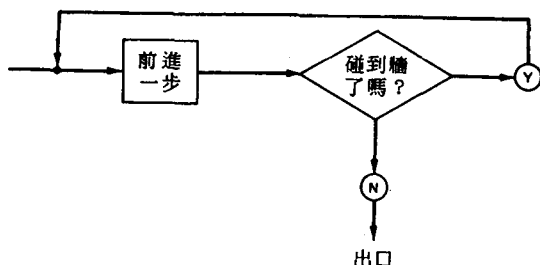


圖 1.1.4 第三階段的流程圖：「保持前進，到達出入口後才停止」

。其中有兩個迴路，一個在第一階段，另一個在第三階段。如此我們就不必為使機器人前進 3.5 公尺而給它 3500 個指令，在第一階段中我們只需要給它兩個指令：一個測試命令和一個前進命令。總而言之，原來有 6901 個指令的程式現在可用 5 個指令的程式取代。如果機器人離牆不足 3.5 公尺或碰撞點離出入口不足 3.4 公尺，則原來程式沒有絲毫用處。但是這五個指令的新程式却可應用在任何位置，只要機器人面對牆，而且出入口在它右手邊即可。

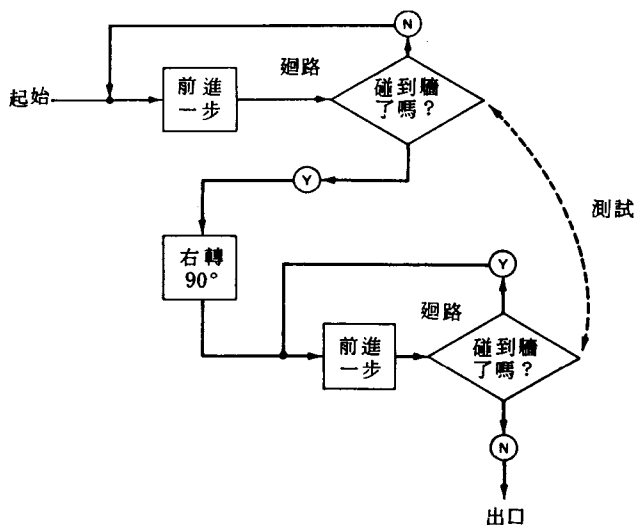


圖 1.1.5 使機器人走到出入口的全部流程圖

在我們撰寫程式時，流程圖十分有用，但流程圖本身却不便於輸入到電腦中。爲了要使程式進入電腦，我們必須一行地將指令鍵入電腦。有時爲了方便起見，我們在某些指令的左邊附上數字標記。

標 記	指 令
1	ADVANCE ONE STEP IF NOT TOUCHING WALL GO TO 1 TURN RIGHT 90°
2	ADVANCE ONE STEP IF TOUCHING WALL GO TO 2

6 電腦與自動控制概論

這個程式的構想是這樣的：在完成一個指令之後，電腦自動跳到下一行的指令上，除非有一個測試結果告訴它跳到別的地方。爲了要使它知道跳往何處，我們就把會跳入的指令附上一個標記。當它完成最後一個指令之後，它就停止動作，工作也全部完成。在此例中，我們有兩處附上標記 (Label)，也就是兩個「ADVANCE ONE STEP」指令。對於這兩個指令我們分別指定不同的標記，以期分辨它們。

由上面這個例子我們可以得到電腦程式的基本概念。我們使用兩種形式的指令：(1)運算指令，告訴電腦做某件事；(2)測試指令，告訴電腦進行一項測試，看看次目標是否已完成。在這種方式下，電腦可以將一組運算重複執行很多次，最後才達成次目標。

在複雜的程式中，電腦所重複執行的未必一定是單一的運算，它很可能一序列的運算。在一個主要的迴路中，可能包括很多個測試，並且達成數個次目標。無論如何，上述的5指令程式已能使電腦將機器人移動到門口，我們也可以用同樣方式撰寫少數幾個指令，但却可控制系統一段很長的時間。

程式設計最巧妙之處在於機器開始行動之前，我們可以先將整個程式輸入其內。由於程式在電腦的記憶體中，所以它可自動地執行下一指令，並且進行測試，決定下一個指令是什麼，它反覆這些動作，直到工作完成爲止。這個原理可以適用在任何情況中，無論這程式是用於控制機器人，或是回答一些複雜的數學問題，或是計算薪資等均可。

在這些電腦應用中有一共同性元素，那就是輸入到電腦中的資料 (Data)，資料就是供程式運算的數值和資訊，因而產生出可用的答案。在下一節我們將繼續討論資料應如何處理。在這個機器人的例子中，程式只使用一種資料，那就是機器人所發出的信號，藉以判斷它是否碰觸到牆。

1.1.2 程式之透視

我們已經見到一個程式的例子，它是一序列的指令，可以告訴電腦如

何執行一些工作。在我們的例子中，這項工作就是控制機器人的行動，使它移動到出入口。在我們討論其他程式的例子之前，先要說明幾項要點。

大部份電腦都是通用型的（General-purpose），這表示它們可用於各種不同的工作中，只要隨時更換程式就行了。通用型的電腦如果執行一個機器人控制程式，它就成為機器人控制器；如果它執行一個薪資程式（見2·2節），它就成為會計機器；如果它執行一個教學程式（見6·1節），它成為一位教師。

不過也有些電腦是專用型的（Special-purpose），這表示它們有特殊的電路，使它們最適合某些特定的工作。它不像通用型電腦在不同場合有不同的程式，它通常只有一個固定的程式，在機器製造時即裝設在內。例如數位電子表及控制汽車點火的新型電子裝置等，都是專用型電腦的例子。

由此我們也可以看出電腦硬體與軟體間的差別。硬體是廠商所供應的實際的電路，而軟體則是指供電腦使用的這些程式。這就導致一項有趣的問題：為一個電腦所撰寫的程式何時才能使用在另一部電腦上？假定你正告訴一位朋友如何修理車子，就如撰寫良好的程式一樣，你很仔細地一步步敘述如何進行。如果你說的是英語，而你的朋友不懂英語，那麼你的說明就完全沒有意義。就算你們都能說英語，但你的朋友連化油器和配電器都分不清楚，那他還是聽不懂你的指示。因此我們要在一個電腦上執行某個程式時，電腦首先必須瞭解這個程式所使用的語言。

在我們討論機器人控制程式的時候，曾提到電腦無法瞭解流程圖，但它可以瞭解五個指令的程式。也就是說，電腦能瞭解如何控制機器人，例如執行「ADVANCE ONE STEP」或「TURN RIGHT 90°」等指令。它也能瞭解下一步應執行那一個指令，例如執行「IF NOT TOUCHING WALL GO TO 1」等測試指令。有些電腦能瞭解這類的指令，有些則否。

像英文及中文等都是自然語言（Natural languages）的例子，也就是人與人之間對話所用的語言。中文和英文有不同的字彙，也有不同的組

8 電腦與自動控制概論

句法則。在程式設計語言中也有類似的情況。每種程式語言都有它獨特的指令格式，以及將這些指令串連起來的規則。

在通用型電腦的硬體中，機器能瞭解以某種特殊語言所撰寫的程式，這種語言稱為機器語言 (Machine language)。假如你能用機器語言把應進行的工作一步步說明出來，電腦就會完全遵照這些指令行事。不同廠商的電腦（有時甚至同一廠商所提供的不同系列電腦）都使用不同的機器語言。

但是大家常會發現用機器語言敘述工作應如何進行十分困難。由於這個緣故，電腦科學家就發展出高階語言 (High-level languages)。它們依然是程式設計語言，但却不是自然語言，其目的就是要使工作說明簡化。由於電腦硬體只能瞭解機器語言，所以需要一個特殊程式，稱為翻譯程式 (Translator)，電腦利用這個程式將一個高階語言程式翻譯成機器語言程式，以便讓機器瞭解。翻譯程式是電腦軟體的一部份，我們將在 2.1 節中詳細討論。

在一般情況下，我們並不需要知道如何利用機器語言撰寫程式，事實上大部份的電腦專家都不使用機器語言。因為他們發現用高階語言撰寫程式更有效率些，翻譯成機器語言的工作就讓電腦去進行。由於用高階語言所撰寫的程式比較容易讓人瞭解，所以寫起來快些，也比較不容易出錯。

由於大部份的讀者都不需要研究機器語言，所以我們只將它列入本書最後一章中。這部份的材料在前些章中並未使用到，而且僅供對電腦硬體細節有興趣的讀者參考之用，使他們能瞭解電腦如何一步步地執行程式中的指令。我們將在 1.3 節中研究一種簡單的高階語言，不過也是拿它作為一個例子而已，後面幾章的內容與這種語言並無關連。

因此我們發現一個重要事實：電腦如果要執行你的高階語言程式，它需要一種機器語言，但你却不必一定要瞭解這種機械語言。當我們對電腦作深一層研究的時候，就會見到很多種高階語言，像供商業應用的 COBOL，工程或科學應用的 FORTRAN，研究人工智慧的 LISP 或 PLANNER，以及 ALGOL、APL、PASCAL、PL/1 ………本書選擇一種高階語

言，稱為 LOGO，供讀者進一步研究之用。

在 1.3 節中，我們研究 LOGO 程式如何控制一個簡單的機器人，稱為 TURTLE，使其能沿着複雜的路徑前進。由於 1.3 節中與後文無關，跳過去也可以，但作一作它的習題將會很有幫助。在本書中我們經常會問起：「這種問題可由電腦解決嗎？」，換句話說，「撰寫程式讓電腦來執行這件工作是否有意義？」如果你從未寫過程式，對這些問題自然很難會有具體的意見。所以本書的目標就是要這方面提供一些知識，使讀者成為技術性社會中有用的一員。

1.1.3 數位電子表

大部份的手表都有時針、分針、和秒針（圖 1.1.6a），它們沿着表面一圈圈地旋轉，顯示出時間。這些指針是緩緩地在表面上轉動，並非由一個位置跳到另一個位置。但是新式的電子表（圖 1.1.6b）却將時間以數字形式表示出來，這些數字每秒鐘變化一次。所以表面並未顯示出時間逐漸增加的情況，因此與一般手表完全不同。

我們把一般手表（圖 1.1.6a）稱為類比裝置（Analog device），因為它的指針圓滑移動，與時間的圓滑流動非常類似（也就是說，它們具有共同的性質）。我們又把把電子表（圖 1.1.6b）稱為數位裝置（Digital device），因為它是以數字的方式表示出時間。

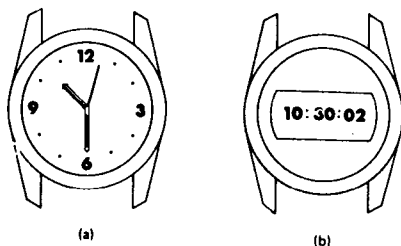


圖 1.1.6 在剛過 10 點 30 分 2 秒時，一般手表的秒針由 2 秒的位置一直向 3 秒的位置移動；但數位手表仍顯示着 10:30:02，直到第 3 秒過完，它才會換成 10:30:03。