
原序

先想想看，你為什麼要教導學生這一門主題，你想讓他們學到什麼？
最終的成果非關技術，只是一般的常識而已。
— 理查·費曼

幾乎所有時髦的電子設備都少不了嵌入式軟體。這些軟體藏在你的手錶、錄放影機、大哥大、冷氣機 ... 甚至微波爐與烤麵包機，嵌入式軟體幾乎是無所不在；軍事用途的嵌入式軟體可以導引飛彈侵襲敵軍的機場，而通訊衛星、深海探測機和先進醫療儀器的內部設計，也幾乎都是嵌入式軟體的天下。

有些人喜歡把所有的軟體一手包辦，像是電子工程師、電腦科學家、以及其它技術領域的專家，我本人就是其中之一。我從本身的經驗得知，這是的確是件吃力的工作，牽涉許多棘手而奧妙的技術。就我所知，目前似乎沒有哪一所學校專門開設嵌入式軟體設計的課程，而且在圖書館裡也找不到一本合適的書。

每一種嵌入式系統都是專屬的架構，它的硬體與其應用領域必須緊密的結合。對於各種嵌入式系統，其程式設計的過程可能迥然不同，需要長久的時間才能掌握其精髓。無論如何，開發嵌入式軟體有一個共通的工具，就是 C 語言。本書將告訴你如何使用 C 與 C++ 來設計嵌入式系統的軟體。

本書所提供的原始碼都有詳盡的解釋，讓你套用的現成的方式，來解決一般嵌入式軟體的問題。本書也涵蓋一些進階的議題，像是記憶體測試與驗證、驅動程式的設計與實作、即時作業系統概念與程式碼的最佳化技術等等。

我寫作本書的目的

曾經有一份評估報告：每一個美國人大約同時擁有八套內含微處理器的裝置。乍看之下，這個數字實在不可思議，我身邊哪來那麼多電腦？之後，當我仔細思考這個問題，我把周遭可能內含微處理器的裝置列出一份清單，不到五分鐘，清單上竟然出現十個嫌犯：電視機、收錄音機、錄放影機、煮咖啡機、鬧鐘、微波爐、洗碗機、遙控器、烤麵包機和電子錶。而這些只是我個人身邊的物品而已 — 如果把工作上的儀器算進去，數目將遠過於此。

事實上，不是光有處理器就夠了，是軟體賦予了產品的生命。我想把程式設計的技巧應用到開發嵌入式系統，但不知從何著手？當時我是大學的應屆畢業生，完全沒有修過任何相關的課程，學校也沒有相關的學科。

幸好，我畢業之後找到一份設計嵌入式軟體的工作，大部份的工作都是靠自己鑽研。公司裡面少數懂得嵌入式系統的人都忙得不可開交，根本顧不了我這個菜鳥，我只有自己去查資料。

現在，我決定還是自己來寫這本書。這份工作前無古人，完全是獨立創作。在寫作的過程中，最困難的就是不知該何時打住。每一個嵌入式系統都是獨立的，我也了解在每一個規則之下都有例外。儘管如此，我盡力闡釋各個主題的精髓，抽取實用層面的技術。

本書的對象

我假設讀者都具備基礎程式設計的經驗，至少要對於這兩種語言 (C/C++) 的語法有基本的認識。我會討論一些基本的資料結構，如鏈串列 (Linked List) ... 等等。你不必非常了解電腦硬體的架構，當然，透過本書你會學習關於硬體的知識。這也是嵌入式系統程式設計師工作的重點之一。

我心目中設定了兩種讀者群，一種是入門的菜鳥，如同我當年剛從學校畢業時一樣：具備電腦科學及工程的背景、有過幾年的程式設計經驗、對於撰寫嵌入式軟體興趣勃勃，但就是不知從何開始；本書的前五章算是你的一塊敲門磚，帶你進入嵌入式系統的殿堂；之後的章節才是你大展身手的所在。

第二種讀者則是針對嵌入式系統設計高手：精通嵌入式系統的硬體，並且知道如何去撰寫軟體，但是需要一本書來參考，進一步深入核心技術。大部分有經驗的嵌入式系統程式設計師都是用組合語言，對於把 C++ 應用到嵌入式系統可能就不甚了了。針對這一類讀者，本書算是一種另類選擇。

不論你屬於哪一種層次，本書都提供了必要的資訊，帶你進入這個極待開拓的蠻荒領域。

本書的架構

本書分為十章，一個附錄，一份相關資訊來源與專有名詞的定義。十個章節可拆成兩個部份，前五章主要針對入門的讀者，只要依照章節的順序閱讀，應該就能掌握基本的開發技巧。

第六章到第十章討論進階的議題，讓一些菜鳥逐漸提升到老鳥的層次。這些章節的內容雖然各自獨立，不過我建議你循序漸進，以得到整體性的觀念。這裡的範例程式對你未來的開發計畫大有幫助。

第一章，緒論：介紹、定義並列舉各種嵌入式系統，說明我選用 C 與 C++ 的理由。

第二章，初試身手：利用一個簡單的 C 程式帶你瀏覽嵌入式程式的設計流程。大部份程式語言的書籍開始會教你說一個 “Hello, World! ”，我也從善如流。

第三章，編譯、連結與定位：開發嵌入式處理器所執行的程式，將會用到一些軟體工具。

第四章，下載與偵錯：展示一些把程式載進嵌入式系統的技術；討論一些進行偵錯的技術和工具。

第五章，認識硬體：如何熟悉一個陌生的平台。

第六章，記憶體：嵌入式系統中關於記憶體的技術，包含測試記憶體的原始碼，與快閃記憶體的驅動程式實作。

第七章，週邊：解釋裝置驅動程式的設計與實作技術，並包含一個計時器（一種常見的週邊）的驅動程式範例。

第八章，作業系統：一個能應用在嵌入式系統的基本作業系統。讓你決定是否需要一個作業系統，該到哪裡購買，或乾脆自己寫一套。

第九章，系統整合：擴充之前所描述的裝置驅動程式與作業系統，如何控制更複雜的週邊，並提供一個完整的應用程式範例，以整合之前的技術。

第十章，最佳化：如何增進嵌入式軟體的執行效率與降低記憶體的需求。善用 C++ 的特性來改進系統的效能。

我盡量在一般的知識與特定範例之間取得一個平衡點。某些情況下，我會簡化瑣碎的細節以提高文章的可讀性；我把重點集中在整體的觀念，而非零碎的工具使用技巧。當你掌握了通盤的概念，任何的系統應該都難不倒你。仔細的研究範例程式，相信會得到不少啟發。

取得範例程式

本書的範例碼可透過 ftp 取得：

```
ftp://ftp.oreilly.com/pub/examples/nutshell/embedded_c/
```

這些範例碼依章節分類，也包含了 makefile 讓你自已重建執行檔。

歡迎指教

我們已就能力範圍所及，對本書的程式碼進行測試驗證，儘管如此，如果你對本書有任何的建言，或是發現有錯誤，請務必讓我們知道，我們將妥善處理你的建議，盡力讓這本書更完美，請寫信到：

美商歐萊禮股份有限公司台灣分公司

台北市杭州南路一段 131 巷 3 號一樓

電話：(02) 2396-8990

傳真：(02) 2396-8916

網址：<http://www.oreilly.com.tw>

電子郵件：mail@ora.com.tw

對於任何本書技術上的問題，請使用電子郵件：

bookquestions@oreilly.com (英文)

lin@oreilly.com.tw (中英文)

對於任何本書技術上的問題，請使用電子郵件：

<http://www.oreilly.com/catalog/embsys/>

或

<http://www.oreilly.com.tw/chinese/c/embed.htm/>

由衷的感謝

在我記憶中，我曾想要寫一兩本書，現在我做到了！我必須承認，一開始我的確一無所知，我不知道一本書的創作需要投入這麼多精力，需要這麼多人參與，更令人驚訝的事是：找這本書的出版商竟然這麼容易！我本來以為這是最困難的部份。

從提議出版算起，這本書耗時兩年才完成，因為我有一份正職工作，而且我也希望保持正常的社交生活，在我最後校稿的部份，我感到有一點抱歉，因為趕稿的關係，所以略嫌匆促。但是持續我的工作對於本書也是有幫助的，因為我可以和許多業界的高手討論，不論是硬體或軟體，其中許多人甚至還幫忙我校稿。

我要感謝我的編輯：Andy Oram。沒有他的熱忱、體諒和鼓勵，這本書不知道何年何月才會完成。

Michael Barr

mbarr@netri.no.com