

8051 記憶體의 規劃 (序)

看了一連串的算數運算，我們暫時休息一下，換個主題來討論。對於那些很期待看到除法運算的朋友們，先跟你們說聲抱歉。

在 8051 的領域裡，有人說學好中斷就等於學好 8051，我們認為這句話要加個但書，因為會說這句話的人，都是在單晶片的領域裡打滾了好些年，甚至是十數年的老經驗，對於一個新手來說，那是很遙遠的，因為不管怎麼說，中斷本來就有很多的細節需要考量，不單單是從程式的部分就可以看得出來的。它必須結合硬體運作的時序、電氣特性、與環境的干擾或特殊需求，適時地做不同的調整，沒有任何寫法是可以適用於所有場合的。

在我們的觀念裡，對於初學者而言，能夠掌握程式的動作，完整且仔細地規劃程式空間與記憶體空間，才是最重要的。因為要學好中斷所依賴的是經驗累積，但是學好掌握程式動作及空間規劃，靠的是正確觀念的建立。如果在學習的初期便能建立正確的撰寫程式技巧，日後撰寫程式時，不論是除錯或是擴充都能得心應手，事半功倍。下面的文章，是撰寫程式時應該要特別注意的細節與觀念，我們將其整理出幾個大綱，在爾後的文章中會陸陸續續向各位站友們介紹。

1. Data Memory 的規劃：

撰寫程式的過程中，不管是運算也好，資料讀寫也好，堆疊的存放也好，用到的都是 Data Memory 的空間。因此妥善規劃 Data Memory 的空間，不但能使系統的運作更為順暢，而動作產生錯誤的時候也能立即發覺問題點。規劃的重點如下：

a. 需要進行位元定址的資料，必須使用位址在 20H 2FH 之間的資料記憶體空間。

b. 運算量較大的計算式，儘量利用預設的暫存器空間 (R0 R7) 來做運算，可以減少記憶體空間的浪費，也可以加快運算的速度。

c. 用來儲存串列通信的接收資料時，如接收資料不加上長度的限制，請儘量將暫存空間規劃在所有的暫存變數之後，並預留一小段的空間才宣告堆疊，這樣才不會因接收過長的資料，導致其它的變數或堆疊的狀態值被外部資料所覆蓋，導致系統的誤動作。

d. 通信指令的設計必須簡潔有力，避免儲存指令的空間不敷使用。

2. Program Memory 的規劃：

8051 的程式記憶體空間有 4K Bytes 可用，而 8052 更多達 8K Bytes，除非您是使用 C 做為程式的撰寫工具，否則單純以 Assembler 所撰寫的程式應該是相當足夠的了。而我們最常使用的晶片，是比 8051 更精簡的 AT89C2051，空間有 2K Bytes，因此在程式空間的規劃上必須更具結構化。底下是筆者整理的撰寫重點：

- a. 變數的宣告必須區分為數值變數與位址變數兩個部分，儘量不要混在一起寫。
- b. 主程式撰寫之前儘量加入中斷起點的宣告，除非是測試用且不會用到中斷的小程式。
- c. 主程式必須包含系統中所有啟始狀態的設定。
- d. 主迴圈最後必須加入一個跳躍指令回到主迴圈的起點。
- e. 系統中所有的動作必須加以模組化。
- f. 由於 8051 的空間有限，在呼叫時必須隨時考慮堆疊區的空間有可能會不夠。

以上的觀念是寫好 8051 程式所必須具備的基本概念，您都記起來了嗎？詳細的內容，我們會在後續的文章中一一說明。