# 將變量靈活定義到外部內存的方法

今天學習了硬漢大哥的STM32H7教程第26章，對其中通過編輯鏈接器分散加載文件實現內存超方便使用的方法很感興趣，因此特意去學習了關於ARM C庫的知識，得到了將變量定義到外部內存的方法。正常定義一個變量到指定的地址范圍有兩種方法：

## 第一種 指定每個變量的實際物理地址

MDK AC6使用方法:在變量後面添加如下指令  
\_\_attribute\_\_((section(".ARM.\_\_at\_0X2407FDFC")));  
MDK AC5使用方法:在變量後面添加如下指令  
\_\_attribute\_\_((at(0X2407FDFC)));  
這種方法的優點為使用簡單，基本不會出現其他問題，缺點也很明顯，需要工程師規劃好內存分配，在變量較多時比較麻煩

## 第二種 修改.sct文件，將內存塊定義為一個節區

.sct文件修改內容

|  |
| --- |
| ; RW data 512KB AXI SRAM RW\_IRAM2 0x24000000 0x00080000 {  \*(.RAM\_D1) } |

MDK AC6與MDK AC5使用方法一致:在變量後面添加如下指令  
\_\_attribute\_\_((section(".RAM\_D1")));

使用此方法即可讓編譯器管理內存分配，使用起來非常方便，但僅使用此方法定義外部內存時，會出現程序卡死的情況，為什麼呢？

原來鏈接器會自動生成一個初始化節區的代碼，在進入main函數之前就會執行，而此時我們的外部存儲往往沒有初始化，這時對其訪問自然會產生錯誤，那麼如何抑制鏈接器的這種行為呢？方法分為兩步。

1. 首先，保證外部存儲所在的節區會被分配到ZI段，這樣相當於告訴C編譯器，內存可以不初始化。  
   MDK AC6保證保證編譯器認定節區為ZI段的方法  
   \_\_attribute\_\_((section(".bss.EXRAM")));  
   在AC6中，.bss.是一定要有的，保證節區為ZI段，後面的內容可以自己定義  
   MDK AC5保證保證編譯器認定節區為ZI段的方法  
   \_\_attribute\_\_((section("EXRAM"),zero\_init));  
   在AC5中，對節區的名字沒有要求
2. 其次，修改.sct文件，告訴鏈接器內存不要初始化  
   .sct文件修改內容

|  |
| --- |
| AC6  ;ZI data - 32MB EXRAM(0xC0000000)  ZI\_EXRAM 0xC0000000 UNINIT 0x02000000 {  \*(.bss.EXRAM)  } |

|  |
| --- |
| AC5  ;ZI data - 32MB EXRAM(0xC0000000)  ZI\_EXRAM 0xC0000000 UNINIT 0x02000000 {  \*(EXRAM)  } |

加入UNINIT關鍵字，告訴鏈接器不要初始化

經過上述的兩個步驟後，僅需要在變量後面加上  
AC6  
\_\_attribute\_\_((section(".bss.EXRAM")));  
AC5  
\_\_attribute\_\_((section("EXRAM"),zero\_init));

即可實現變量在外部內存中定義，另外注意一點，用這種方法定義的全局變量是無法賦初值的，賦初值編譯器會報錯。  
參考資料：《安富萊\_STM32-V7開發板\_用戶手冊，含BSP驅動包設計（V3.2）.pdf》第26章 《Arm&#174; Compiler Version 6.16 Reference Guide》  
 