

Condivisione memoria

Edoardo Di Pietrantonio
edoardo.dipietrantonio@raceup.it

March 5, 2022

Chapter 1

Sintassi di condivisione memoria

1.1 CPU1

- main.c Ciclo while in cui viene settato il registro MemCfgRegs.GSxMSEL.bit.MSEL_GSX dove X indica il numero del banco di memoria da condividere
- main.h `#pragma DATA_SECTION(var_to_share,"SHARERAMGSX")` va sempre inserito nel file .h dove viene definita e inizializzata la variabile `var_to_share`.

ATTENZIONE la stringa SHARERAMGSX indica in quale indirizzo inserire la variabile condivisa, la definizione del simbolo SHARERAMGSX si trova nel file 2837xD_FLASH_lnk_cpu2.cmd al path 2837xD_FLASH_lnk_cpu1.cmd al path `C:\ti\c2000\C2000Ware_3_04_00_00\device_support\f2837xd\common\cmd`. Di default sono definiti solo SHARERAMGS1 e SHARERAMGS2, ma è possibile aggiungere qualsiasi zona si voglia.

ATTENZIONE se si allocano due variabili alla stessa zona SHARERAMGSX si sovrascrivono a vicenda.

1.2 CPU2

Valgono le stesse considerazioni della CPU1 con la differenza che il ciclo while nel main attende che la memoria venga concessa dalla CPU1 e che il file di linking di riferimento 2837xD_FLASH_lnk_cpu2.cmd

Chapter 2

Sintassi di condivisione memoria

2.1 CPU1

- main.c Ciclo while in cui viene settato il registro MemCfgRegs.GSxMSEL.bit.MSEL_GSX dove X indica il numero del banco di memoria da condividere
- main.h `#pragma DATA_SECTION(var_to_share,"SHARERAMGSX")` va sempre inserito nel file .h dove viene definita e inizializzata la variabile var_to_share.

ATTENZIONE la stringa SHARERAMGSX indica in quale indirizzo inserire la variabile condivisa, la definizione del simbolo SHARERAMGSX si trova nel file 2837xD_FLASH_lnk_cpu2.cmd al path 2837xD_FLASH_lnk_cpu1.cmd al path C:\ti\c2000\C2000Ware_3_04_00_00\device_support\f2837xd\common\cmd. Di default sono definiti solo SHARERAMGS1 e SHARERAMGS2, ma è possibile aggiungere qualsiasi zona si voglia.

ATTENZIONE se si allocano due variabili alla stessa zona SHARERAMGSX si sovrascrivono a vicenda.

2.2 CPU2

Valgono le stesse considerazioni della CPU1 con la differenza che il ciclo while nel main attende che la memoria venga concessa dalla CPU1 e che il file di linking di riferimento 2837xD_FLASH_lnk_cpu2.cmd

Chapter 3

Aggiunta variabili memoria condivisa

La memoria effettivamente condivisa è organizzata attraverso una struct `Share_struct` sh che contiene diversi campi (array e struct interne, consiglio vivamente di controllare il codice). La seconda zona di memoria condivisa è dedicata al timestamp che è un semplice int.

IL SIMBOLO \Rightarrow INDICA COSA FARE SE SI VUOLE AGGIUNGERE UN VALORE DA LOGGARE

3.1 CPU1

- `main.h` Qui sono definite le struct per la condivisione \Rightarrow aggiungere il campo da condividere. Nel caso di aggiunta di una struct inserirla anche all-interno di `Share_struct`.
- `main.c` `Timer1`: qui avviene la scrittura effettiva nella scheda SD \Rightarrow aggiungere tutti i dati nuovi nella funzione `sprintf`
- `main.c` `Shared_Ram_dataRead_c1()`: aggiornamento memoria condivisa \Rightarrow non serve fare nulla

3.2 CPU2

- `utils.h` Qui sono definite le struct per la condivisione \Rightarrow aggiungere il campo da condividere. Nel caso di aggiunta di una struct inserirla anche all-interno di `Share_struct`.

- `main.c` (circa alla fine del `setup`) Qui avviene l'inizializzazione delle variabili condivise \Rightarrow solo se hai aggiunto una nuova `struct/array` all'interno di `Share_struct` devi inizializzarla
- `car_management.c` funzione `update_log_values()` Questa funzione aggiorna i valori della memoria da condividere \Rightarrow è assolutamente fondamentale aggiornare qualsiasi campo aggiunto in questa funzione.
- `car_management.c` funzione `update_shared_mem()` Questa funzione aggiorna i valori della memoria da condividere \Rightarrow solo se hai aggiunto una nuova `struct/array` devi tenerle aggiornate anche in questa funzione