


Assembleur dans du code C

On peut utiliser la directive `"__asm"` en C pour utiliser de l'assembleur.

```
Void main(void)
{
    __asm lds #$4000

    while(1)
    {

    }
}
```

Valeur de retour d'une fonction

la valeur de retour est retournée par les registres selon le tableau:

Size of Return Value	Type Example	Register
1 byte	char	B
2 bytes	int	D
3 bytes	far data pointer	X(L), B(H)
4 bytes	long	D(L), X(H)

Functions returning a result larger than two words are called with an additional parameter. This parameter is the address where the result should get copied to.

Accéder aux variables et fonctions assembleur depuis C

C → ASM

Il faut d'abord externaliser les variables que l'on veut rendre disponible aux autres fichiers du projet avec **XDEF**

```
;-----  
;Copyright 2003-20xx Haute école ARC Ingénierie, Switzerland.  
;All rights reserved.  
;-----  
;Nom du fichier : MonFichier.asm  
;Auteur et Date :  
  
;But :  
  
;Modifications  
;Date      Faite   Ctrl   Description  
;  
;-----  
  
; export symbols  
    XREF CData ; Déclarée de le fichier C  
    XDEF ASMData, ASMConst, AddVar  
  
;-----  
; data section  
;-----  
DEFAULT_RAM:      SECTION  
  
ASMData: DS.W 1      ;Déclaration d'une variable  
  
;-----  
; code section  
;-----  
DEFAULT_ROM:      SECTION  
ASMConst: DC.W $44A6 ; Déclaration d'une constant  
  
; Une fonction assembleur qui additionne CData au contenu de B (parameter  
; d'entrée)  
AddVar:  
    ADDB CData      ; add CData to the parameter in register B  
    STAB ASMData    ; result of the addition in ASMData  
    RTS
```

← externalise

Il faut créer un fichier header du même nom que le fichier qui contient le assembleur mais avec l'extension ".h" et contenant la définition des variable exportée

```
;-----  
;Copyright 2003-20xx Haute école ARC Ingénierie, Switzerland.  
;All rights reserved.  
;-----  
;Nom du fichier : MonFichier.h  
;Auteur et Date :  
  
;But :  
  
;-----  
#ifndef _MIXASM_H_  
#define _MIXASM_H_  
  
extern unsigned int  ASMData ;  
extern const unsigned int  ASMConst;  
void AddVar(unsigned char value);  
  
#endif
```

Et on peut utiliser tout ça dans notre code C

```
-----  
;Copyright 2003-20xx Haute école ARC Ingénierie, Switzerland.  
;All rights reserved.  
-----  
;Nom du fichier : MonFichier.c  
;Auteur et Date :  
  
;But :  
  
-----  
  
#include "MonFichier.h"  
  
static int Error = 0;  
const unsigned char CData = 12;  
  
void main(void)  
{  
    AddVar(10);  
  
    if (ASMDData != CData + 10)  
    {  
        Error = 1;  
    }  
    else  
    {  
        Error = 0;  
    }  
  
    for(;;); // wait forever  
}
```

Accéder aux variables et fonctions C depuis le code assembleur

ASM → C

Passage de paramètres à une fonction assembleur

Il existe une norme qui est respectée par les compilateurs pour le passage de paramètres aux fonctions. En C c'est celle du langage Pascal.

Les paramètres sont passés de gauche vers la droite par la pile sauf le dernier qui est passé par les registres si il est parmi les types ci-dessous sinon il va aussi dans la pile.

A, B, C, D

A, B, C → Stack

D → les registres si type, sinon Stack

Size of Last Parameter	Type Example	Register
1 byte	char	B
2 bytes	int, array	D
3 bytes	far data pointer	X(L), B(H)
4 bytes	long	D(L), X(H)

Parameters having a type not listed are passed on the stack (i.e., all those having a size greater than 4 bytes).

Importation des variable C

On importe les variables du C dans le code assembleur avec **XREF**

```
;-----  
;Copyright 2003-20xx Haute école ARC Ingénierie, Switzerland.  
;All rights reserved.  
;-----  
;Nom du fichier : MonFichier.asm  
;Auteur et Date :  
  
;But :  
  
;Modifications  
;Date      Faite   Ctrl   Description  
;  
;-----  
  
; Importation des symboles  
      XREF CData      ;External declaration of a variable  
      XREF CConst     ;External declaration of a constant  
      XREF Mfct_enC   ;External declaration of a function  
  
;-----  
; data section  
;-----  
DEFAULT_RAM:      SECTION  
  
;-----  
; code section  
;-----  
DEFAULT_ROM:      SECTION  
  
; Une fonction assembleur  
MFct:  
    LDAA CConst      ;  
    STAB CData       ;  
  
    ...  
  
    ; 1er parameter par la pile  
    LDAA #$33  
    PSHA  
    ; dernier paramètre par les registres  
    LDD #$112  
    JSR Mfct_enC  
    PULA  
    RTS
```

} import

dans le fichier C :

```
-----  
;Copyright 2003-20xx Haute école ARC Ingénierie, Switzerland.  
;All rights reserved.  
-----  
;Nom du fichier : MonFichier.c  
;Auteur et Date :  
  
;But :  
  
-----  
  
unsigned int CData;          /* Definition of a variable */  
unsigned const int CConst;   /* Definition of a constant */  
  
static int sSomme            /* La somme de 2 nombres */  
  
  
// Exemple de fonction C  
void MFct_enC(char aNb1, int aNb2)  
{  
    sSomme=aNb1+aNb2;  
}
```

} export implicite

On met le code C avec l'assembleur dans le même proj.

