

Tutoriel : μ VISION version 5

Etapes à suivre:

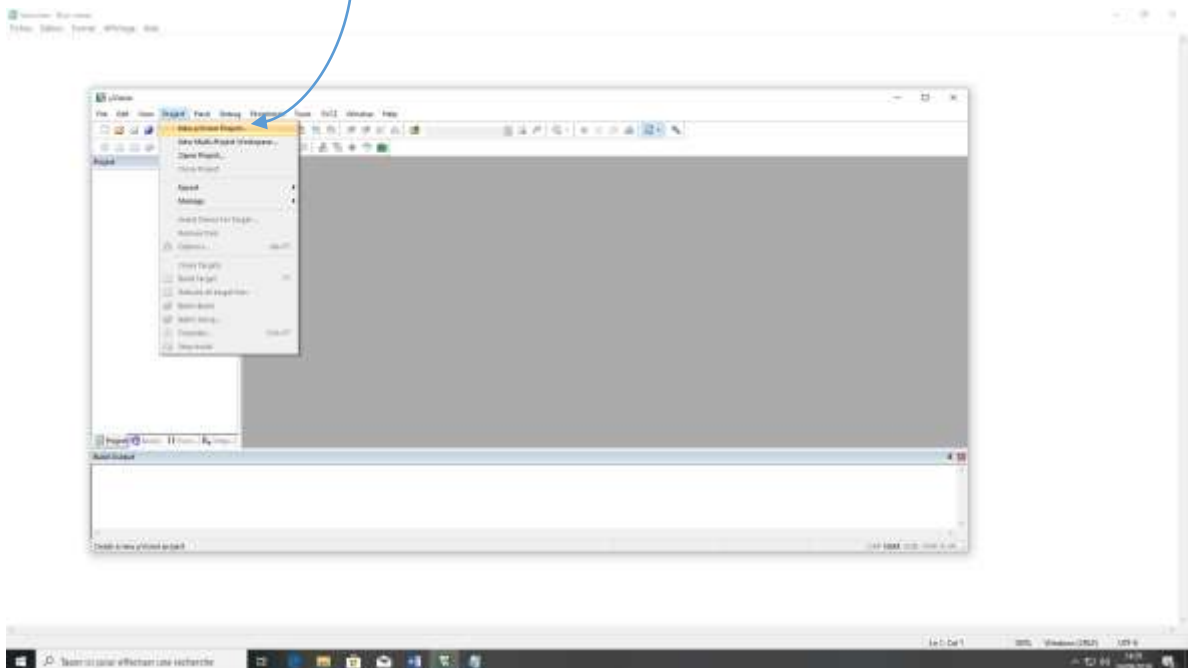
1. Sous l'explorateur Windows, créez un répertoire de travail dans le répertoire temporaire existant « C:\Temp » : par exemple on peut créer C:\Temp\TDm1.

N'oubliez pas de sauvegarder vos fichiers (usb, mail ...)

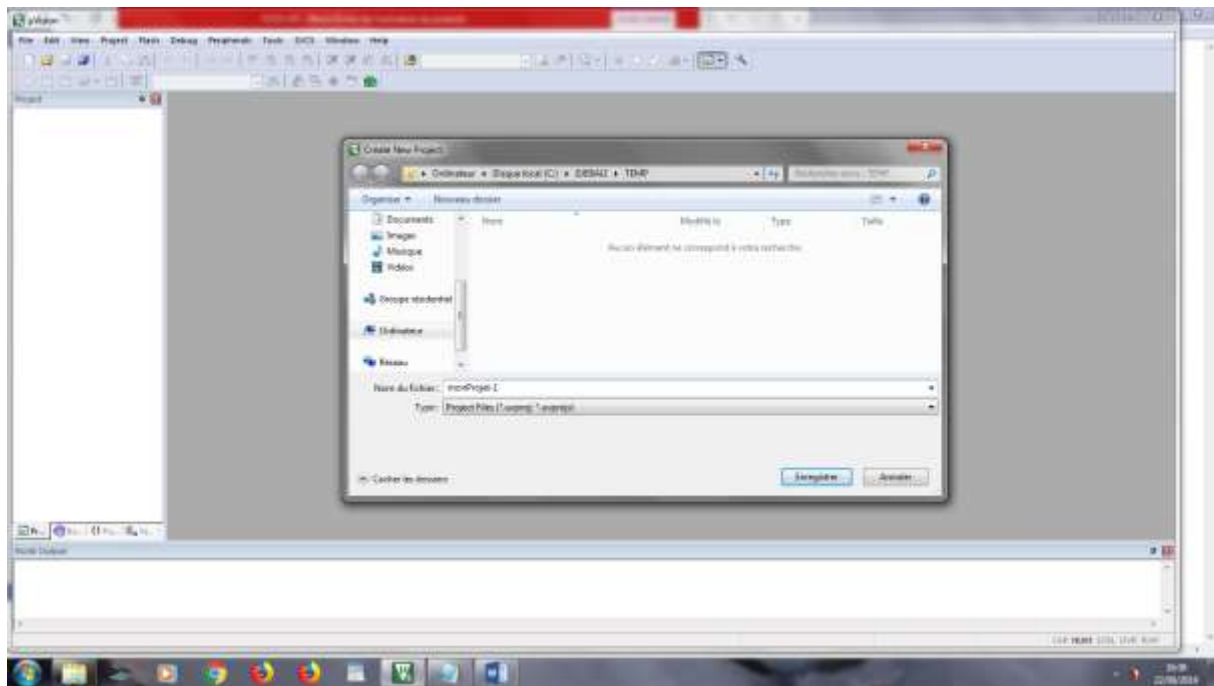
2. Lancez μ Vision version 5.

- Project → **new μ Vision Project** : pour créer un nouveau projet.
- En donnant un nom à votre projet : **monProjet-1**

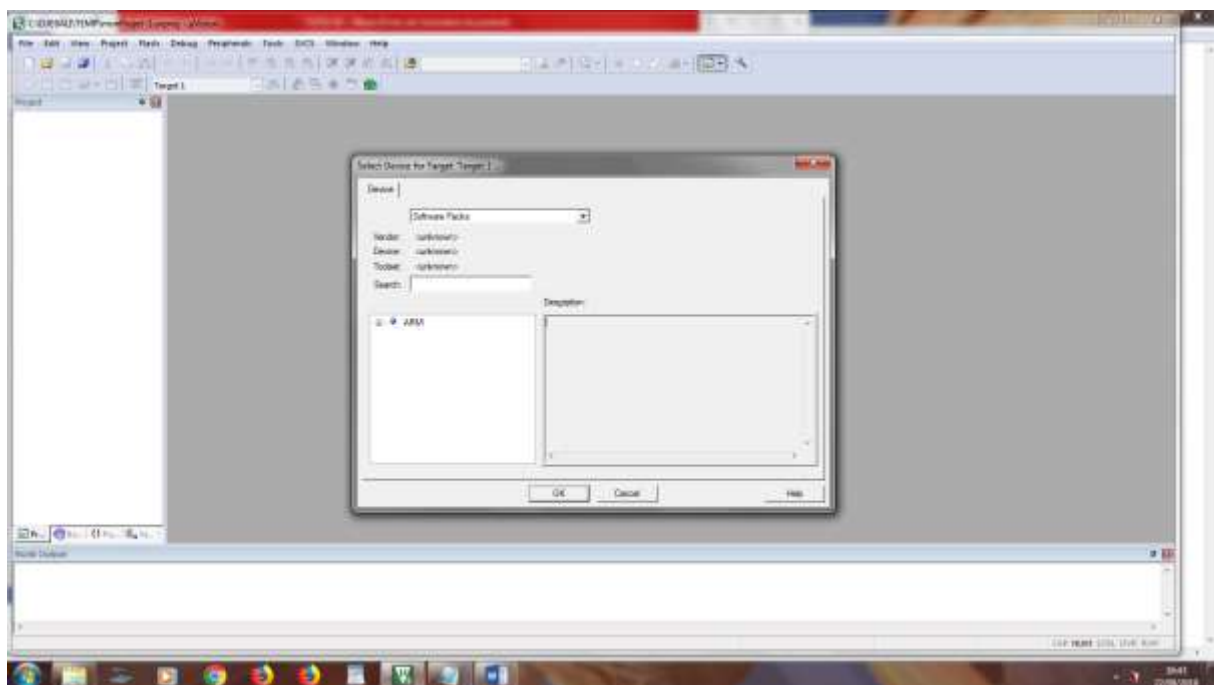
Choisir dans quel répertoire vous voulez placez le projet



En donnant un nom à projet : **monProjet-1** (sous c:\temp par exemple)

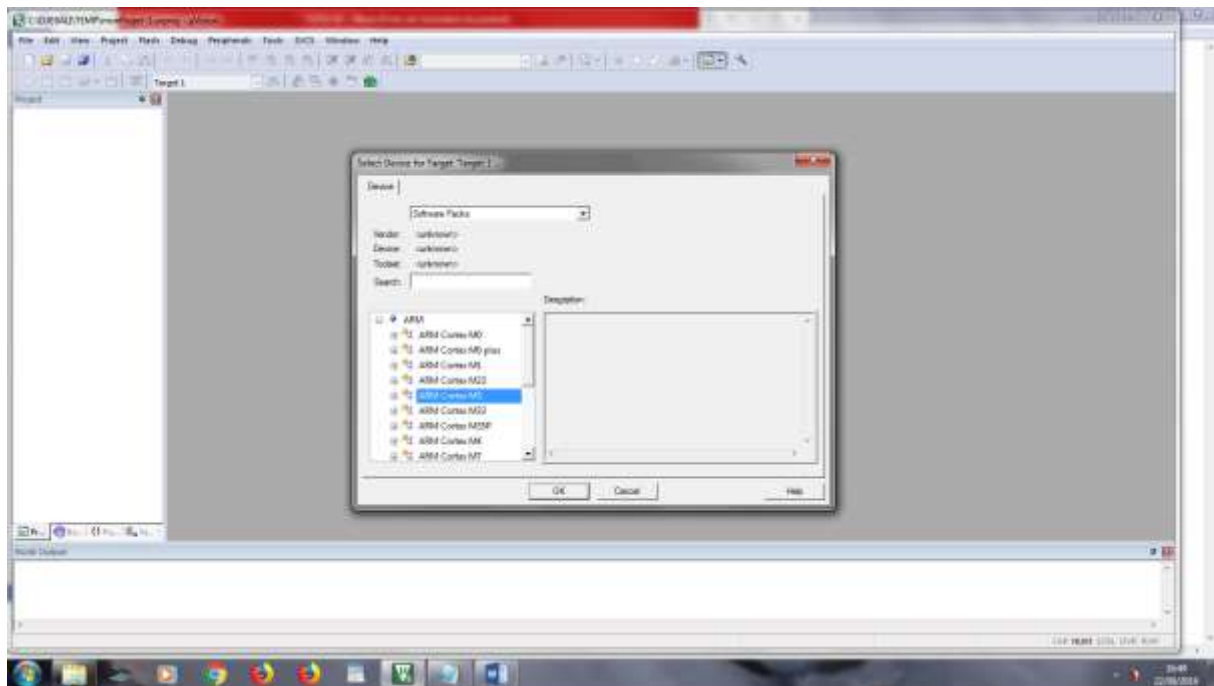


Après enregistrement on obtient le menu suivant :



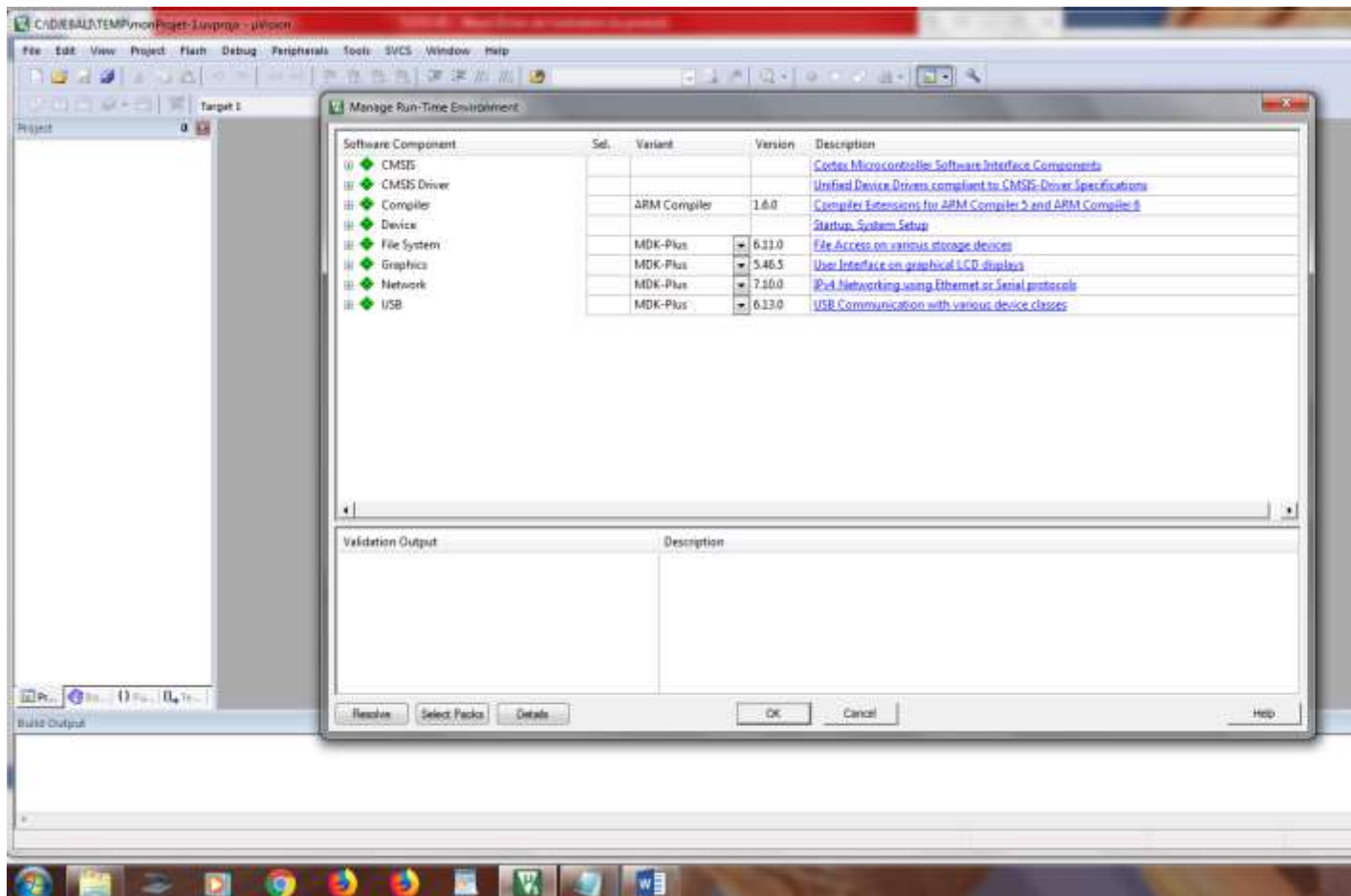
Dépliez ARM (en appuyant sur **+ARM**)

On obtient l'écran suivant :



Choisir **ARM Cortex M3**

On obtient l'écran suivant : **Manage Run-Time Environment**



Dans le menu **Manage Run-Time Environment**

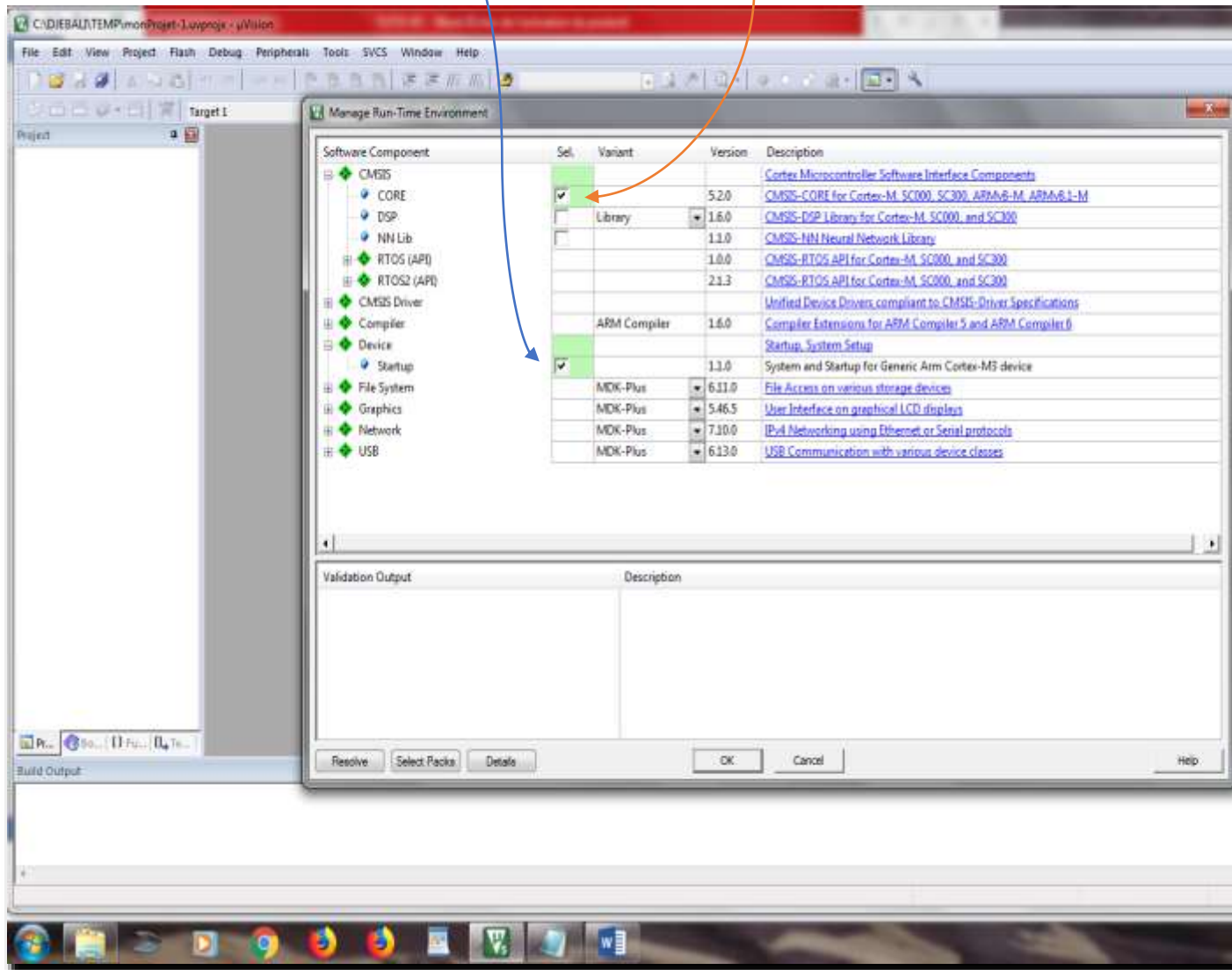
Dépliez **+ CMSIS** et cochez **CORE**

Dépliez **+ CMSIS DRIVER** :

Dépliez **+ DEVICE DRIVER**

Dépliez **Device** et cochez **Startup**

On obtient :



Puis tapez **OK**

Création d'un fichier assembleur (.s)

Créer un fichier (File New) par exemple de nom exo1 et extension « s » : exo1.s et le sauvegarder (File Save) dans votre répertoire de travail.

Ecrire le code de votre programme suivant en langage assembleur ARM dans un fichier **exo1.s** sous le répertoire Temp :

```

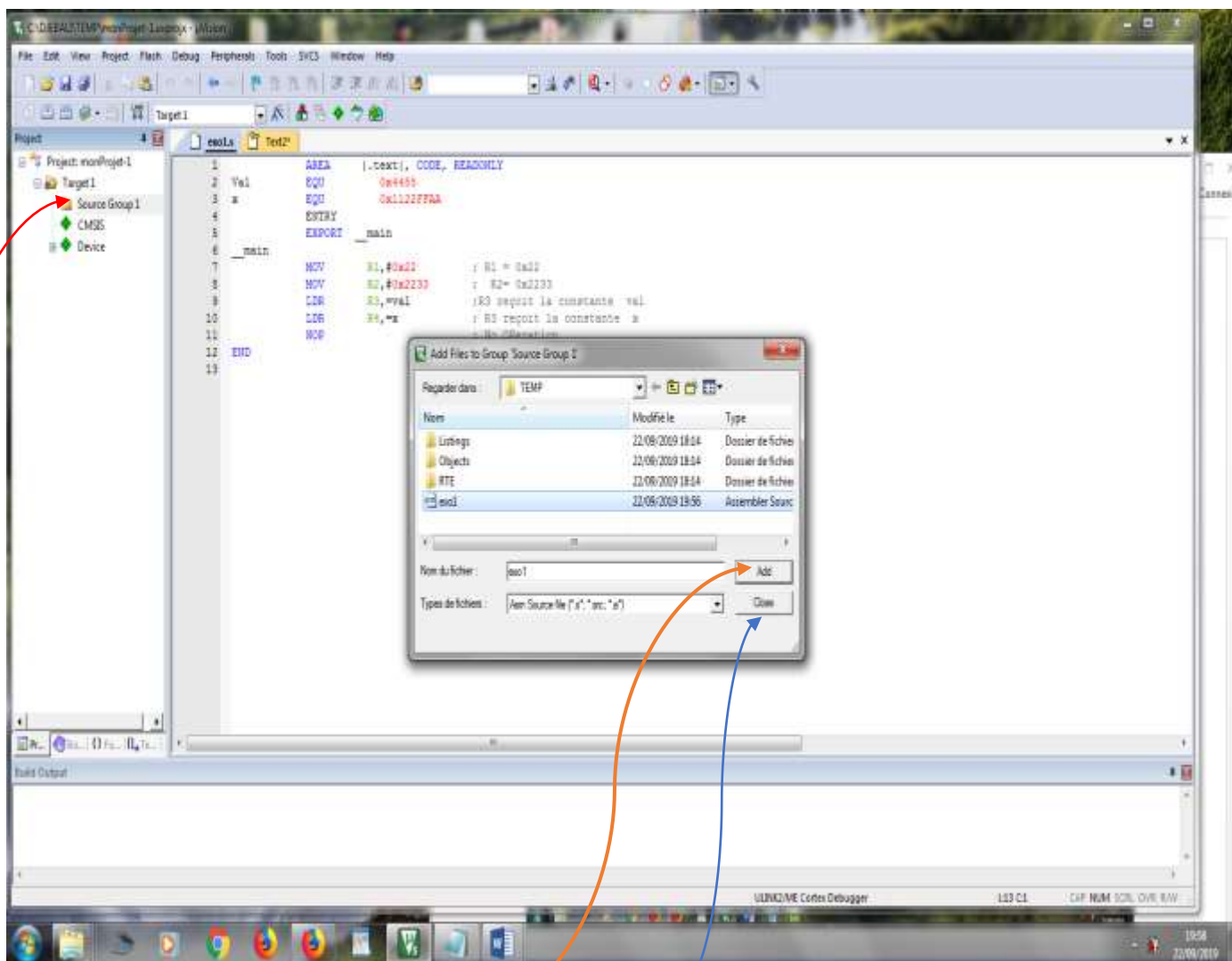
val      AREA |.text|, CODE, READONLY
x        EQU      0x4455
        EQU      0x1122FFAA
ENTRY
EXPORT   __main      ; le mot main est précédé par 02 soulignés (underscore)

__main
        MOV      R1,#0x22      ; R1 = 0x22
        MOV      R2,#0x2233    ; R2= 0x2233
        LDR      R3,=val      ; R3 reçoit la constante val
        LDR      R4,=x        ; R3 reçoit la constante x
        NOP          ; No Operation
        END

```

Ajouter un nouveau **Fichier** de type assembleur(.s). clic droit sur « Source Group 1

Selectionnez : Add Existing Files to Source Group1.



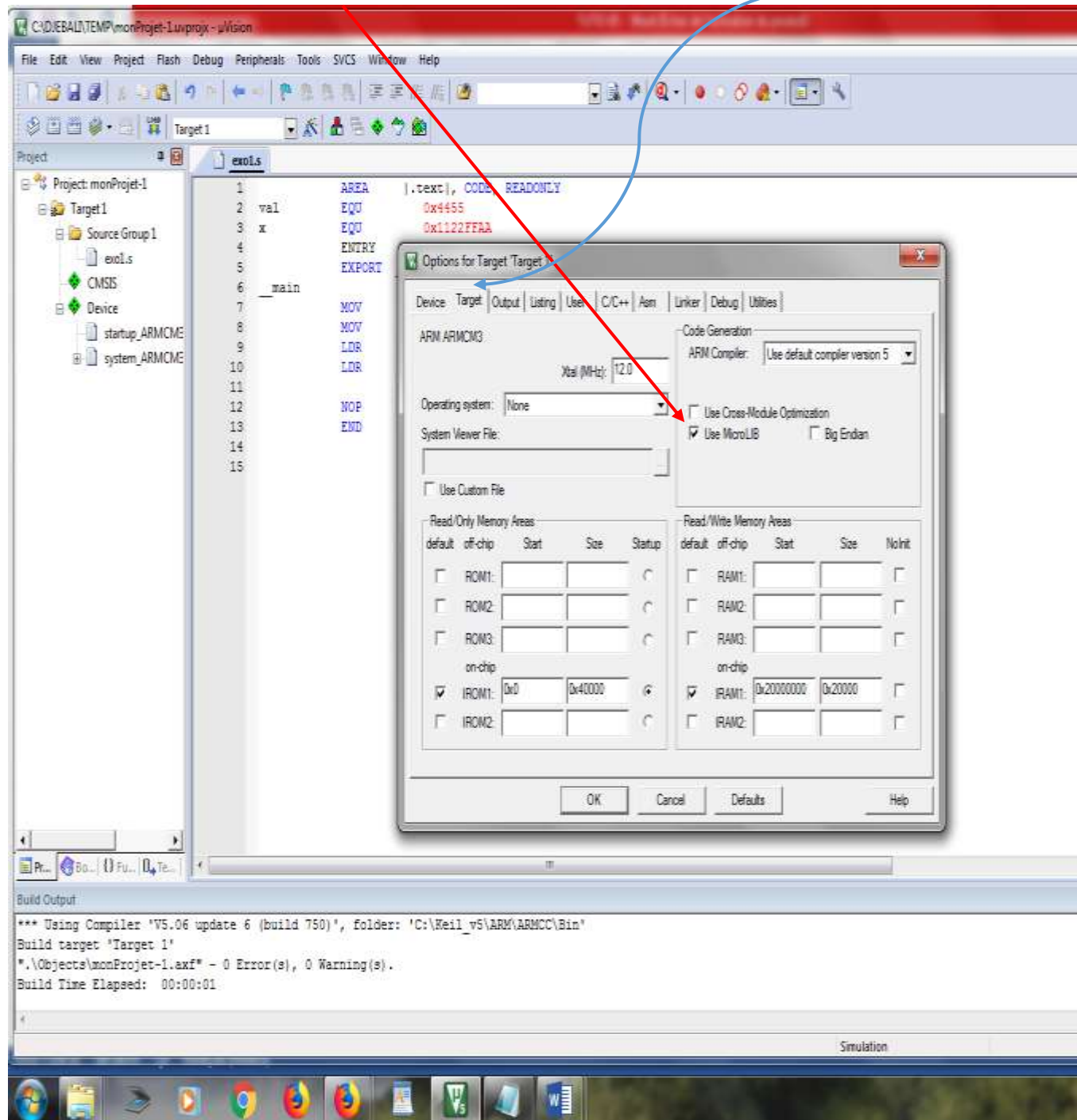
Selectionnez : **exo1.s** et Cliquez sur **add** puis **close**

Dépliez **+ source** votre source .s (ici exo1.s) fait maintenant partie du projet

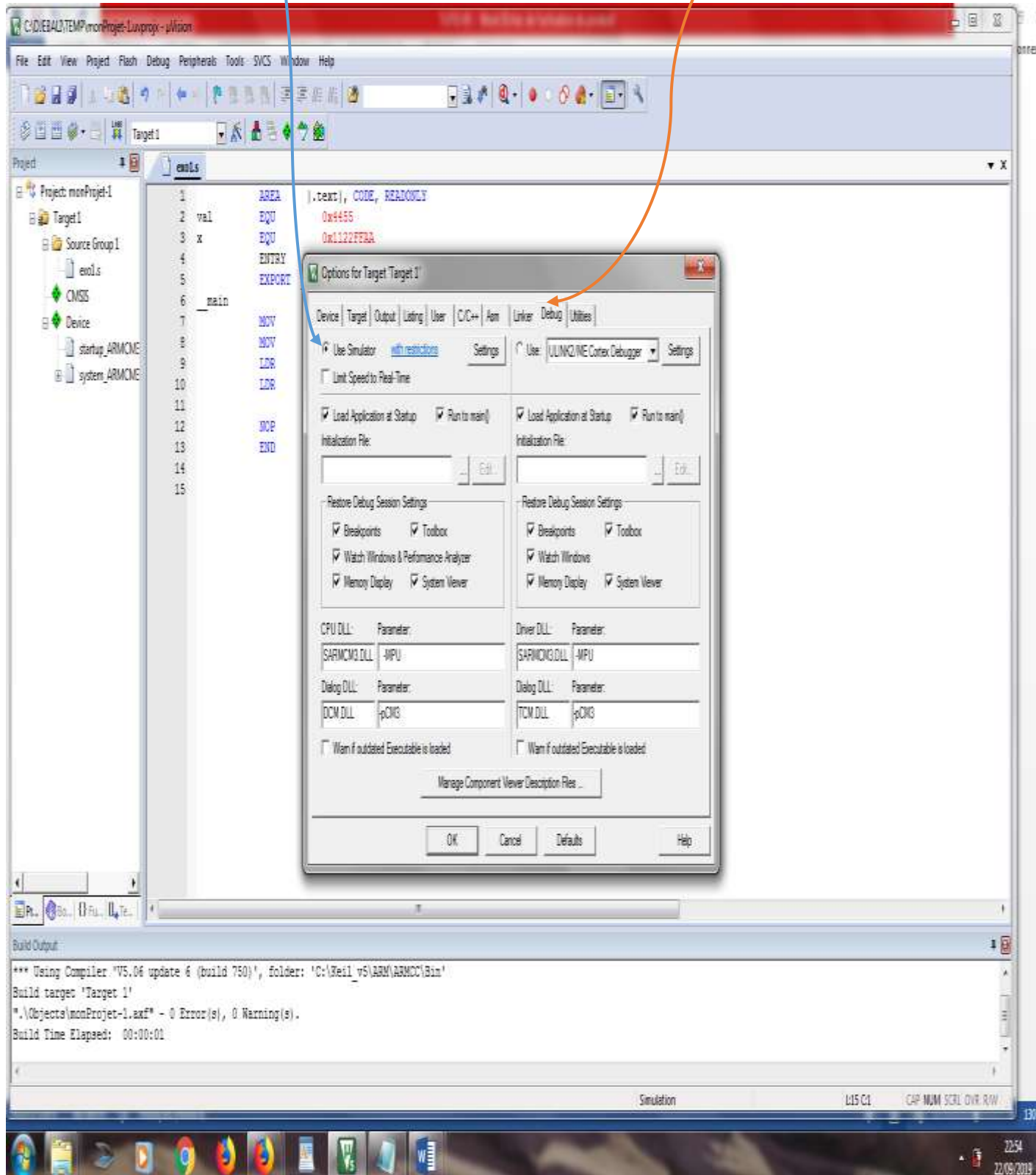
Aller dans le menu Project et ouvrir la fenêtre « Options for Target 'Target 1' »

Project > Options for Target 'Target 1'

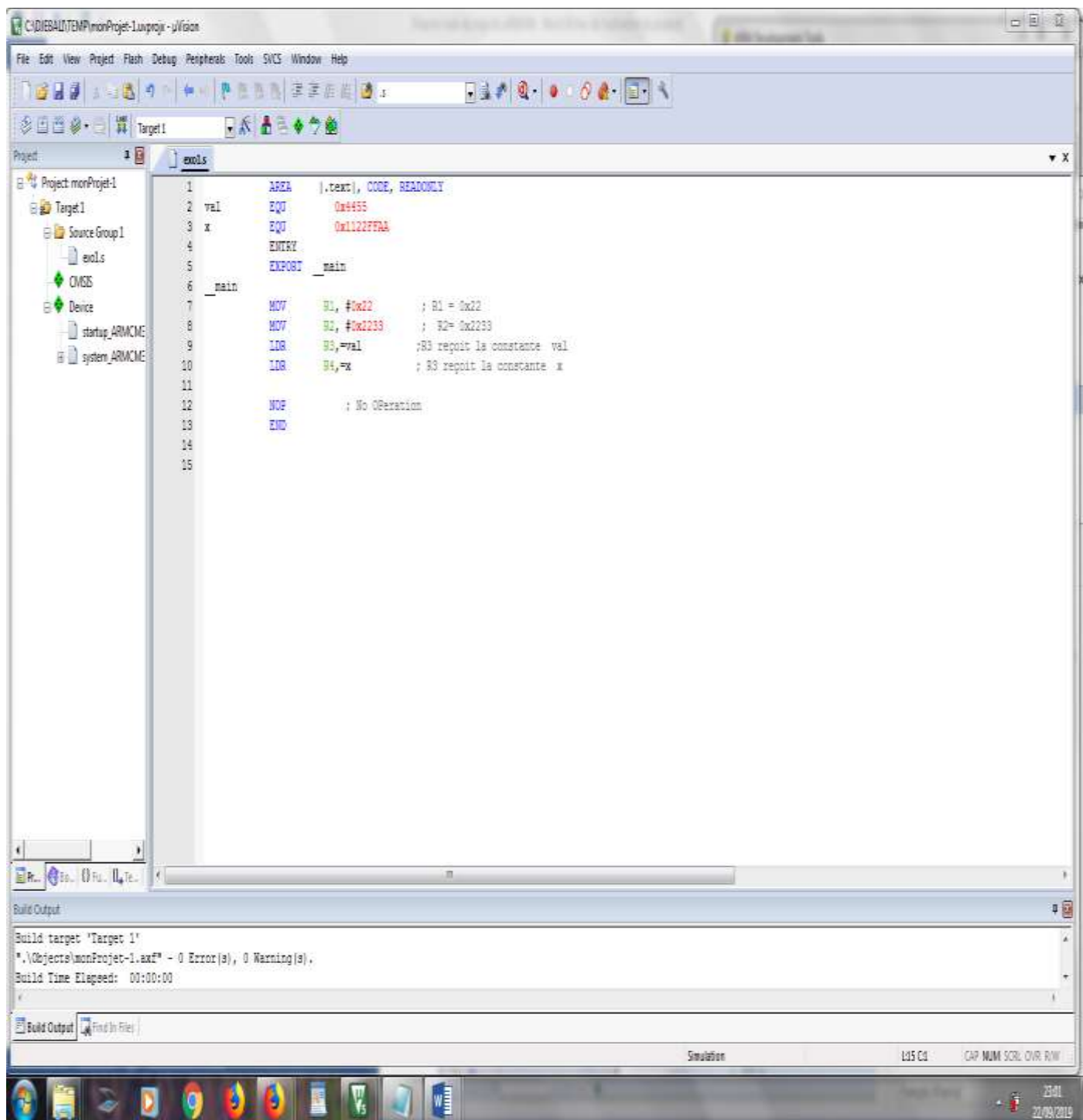
Cocher l'option « Use MicroLIB » de Options for Target 'Target : onglet Target



Cocher l'option « Use Simulateur » de Options for Target dans Onglet **debug**

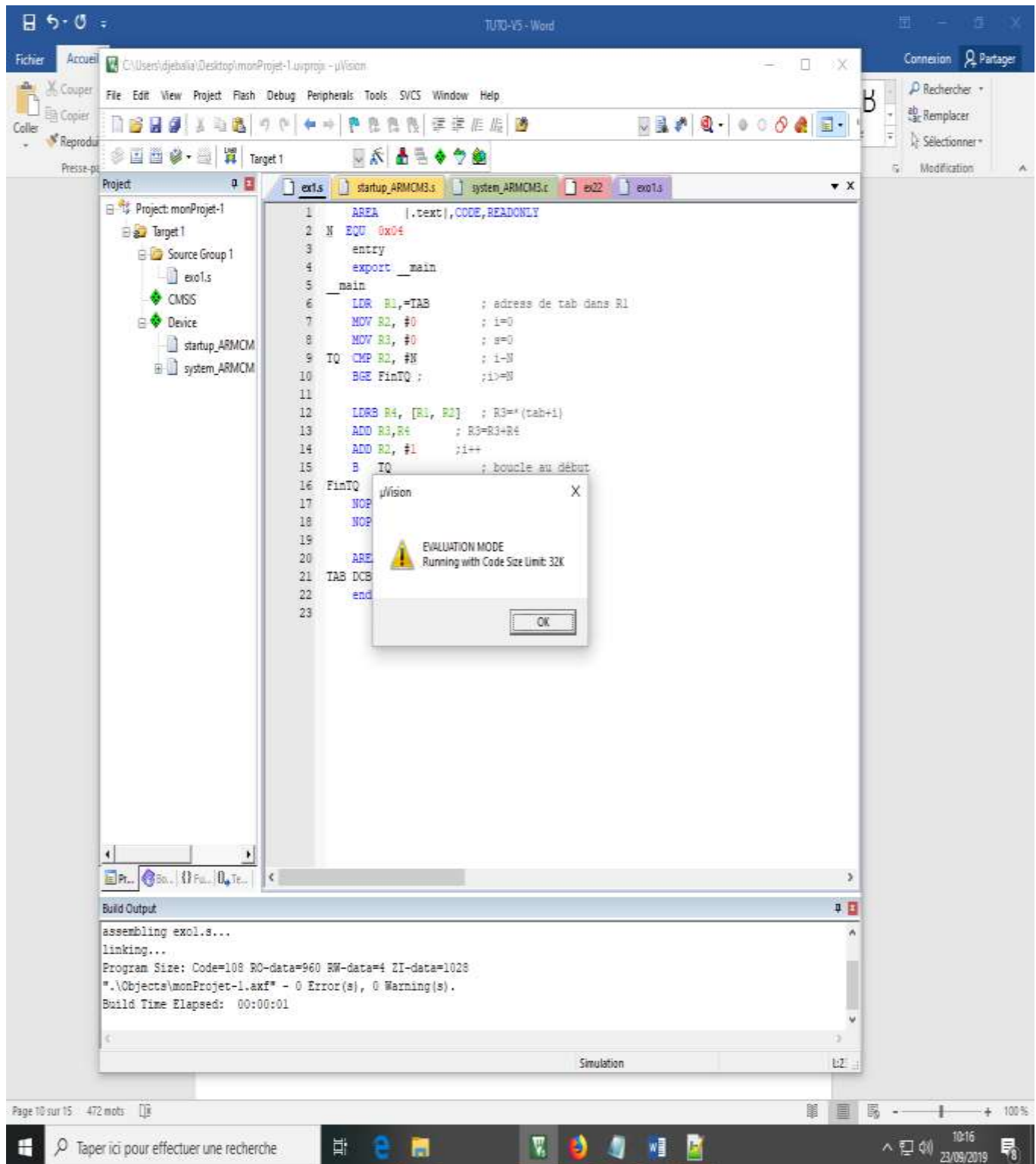


. Traduire le code source du projet exo1 (on dit « assembler ») à l'aide de la commande « **build target**» (F7).



S'il n'y a aucun message d'erreur Le programme est assemblé.

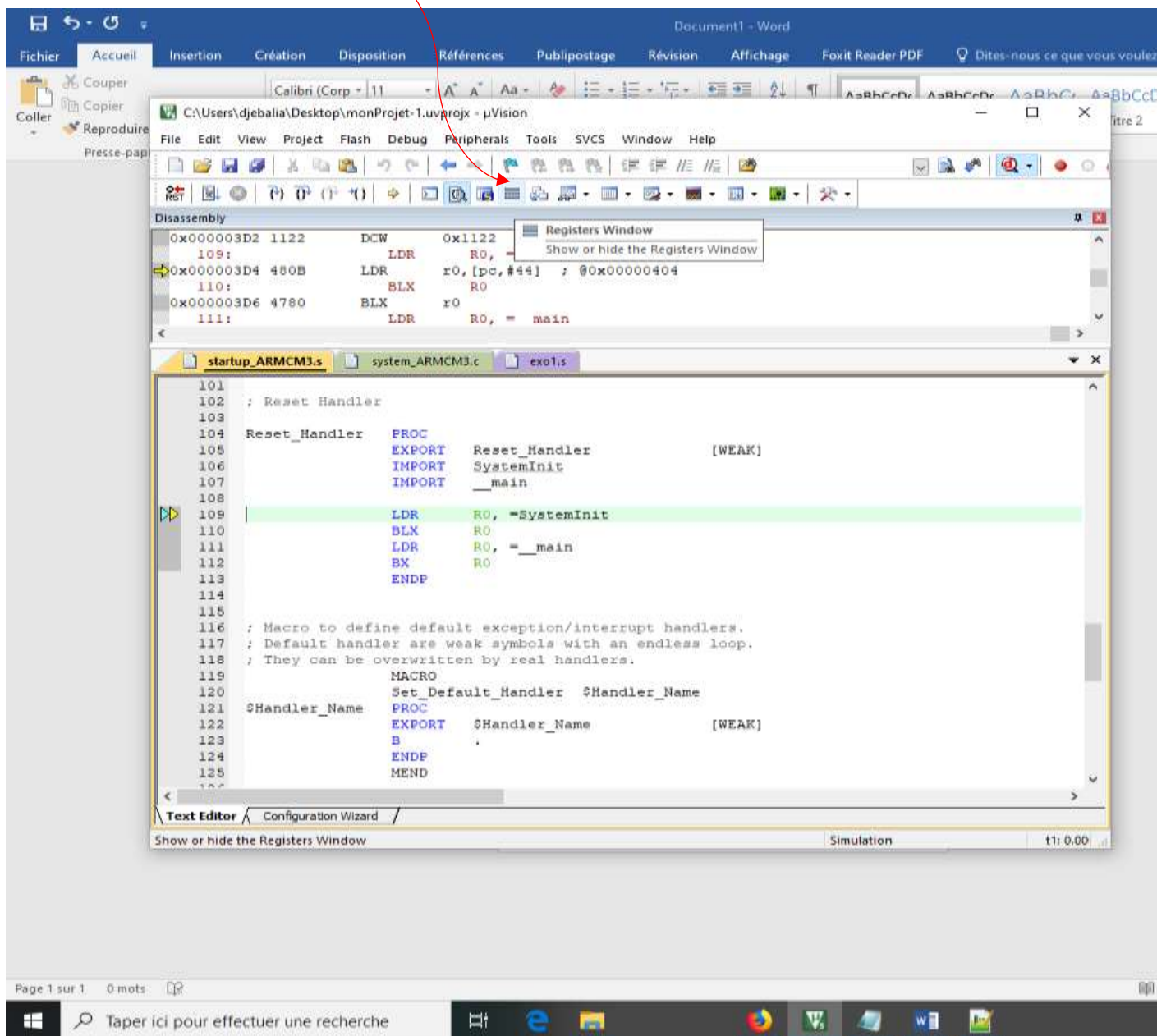
Exécution du programme : Tapez CTRL F5



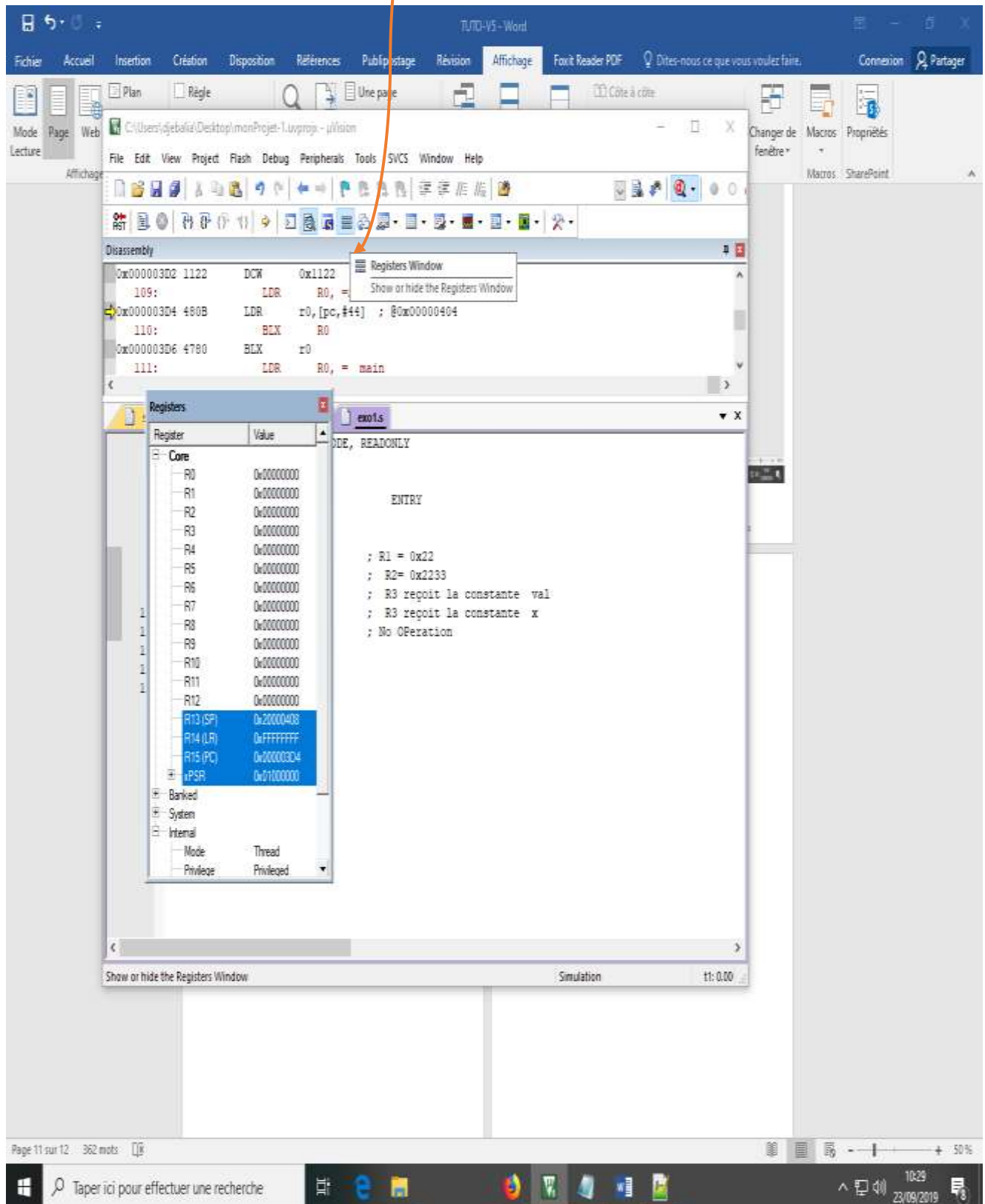
Tapez OK

On obtient l'écran suivant :

Cliquez sur **Registers Window**



Déplacer et élargir la fenêtre « Registers Window » permettant de visualiser les contenus des registres internes du microprocesseur.



Document1 - Word

Fichier Accueil Insertion Création Disposition Références Publipostage Révision Affichage Font Reader PDF Dites-nous ce que vous voulez faire... Connexion Partager

Calibri (Corp) 11 A Aa

Rechercher Remplacer Sélectionner Modification

C:\Users\djefalal\Desktop\monProjet-1\proj - uVision

File Edit View Project Flash Debug Peripherals Tools SVCS Window Help

Disassembly

Address	Disassembly	Commentary
0x000003BE 0000	DCW 0x0000	
7: 0x000003C0 F04F0122	MOV R1, #0x22	; R1 = 0x22
8: 0x000003C1 F04F0123	MOV R2, #0x2233	; R2 = 0x2233
9: 0x000003C4 F2422233	MOVW R2, #0x2233	
9: 0x000003C4 F2422233	LDR R3, =val	; R3 reçoit la constante

Registers

Register	Value
R0	0x000003C1
R1	0x000003C2
R2	0x00000000
R3	0x00000000
R4	0x00000000
R5	0x00000000
R6	0x00000000
R7	0x00000000
R8	0x00000000
R9	0x00000000
R10	0x00000000
R11	0x00000000
R12	0x00000000
R13 (SP)	0x20000408
R14 (LR)	0x000003D9
R15 (PC)	0x000003C4
xPSR	0x10000000

startup_ARMCM3.s system_ARMCM3.c ex01.s

```

1 AREA |.text|, CODE, READONLY
2 val EQU 0x4455
3 x EQU 0x1122FFAA
4
5 EXPORT __main
6 __main
7 MOV R1, #0x22 ; R1 = 0x22
8 MOV R2, #0x2233 ; R2 = 0x2233
9 LDR R3, =val ; R3 reçoit la constante
10 LDR R4, =x ; R4 reçoit la constante
11 NOP ; No Operation
12
13 END
14

```

Simulation t1: 0.00

Page 2 sur 2 · 0 mots

Taper ici pour effectuer une recherche

10:32 23/09/2019

- . **Exécuter pas à pas** (instruction par instruction) le programme à l'aide de la commande « Step Over » et observer le changement des contenus des registres, ainsi que l'état de chaque indicateur : **C, V, Z et N**.

Remarque :

Pour une exécution totale du programme, utiliser la commande « Run

