

0 Présentation du TP :

Pré-requis : Durée estimée :

Objectifs :

- \Rightarrow Connaissance du langage C ANSI.
- \Rightarrow 1 heures
 - \Rightarrow Ecrire, simuler et débugger un programme C

Ce TP vous permettra d'appréhender les outils de débogage intégrés dans ALTIUM associés au **langage C**. Cela vous permettra de simuler et mettre au point des programmes en **langage C**.

A l'issu de cette mise au point le code développé à pour finalité d'être descendu sur le processeur embarqué TSK3000A, ce qui nous renvoie vau TP3.

Sommaire du TP:

- 1 Création du projet embarqué
- 2 Ecriture du programme C « Hello World ! »
- 3 Mise au point du programme en langage C
 - \Rightarrow Simulation du programme avec un point d'arrêt
 - \Rightarrow Visualisation d'une variable
 - \Rightarrow Visualisation des registres SFR du processeur TSK3000A
 - \Rightarrow Exécution du programme en mode pas à pas

Dans ce TP nous nous servirons d'un programme simple affichant la chaîne de caractères « Hello World ! »

Ceci est un petit clin d'œil à Messieurs Ritchie, Thomson et Kernigham :

 \Rightarrow L'origine du langage C : Il a été développé par Dennis **R**itchie et **K**en Thompson dans les années 70. Ce langage C des origines est nommé **K&R C**.

Brian Kernighan aida à populariser le langage à l'aide du livre **« The C Programming** Language » décrivant le langage enfin stabilisé en 1978.

Dans son livre Kernighan propose un programme d'exemple : Le programme Hello World. Créer un programme affichant Hello World est depuis devenu l'exemple de référence pour présenter les bases d'un nouveau langage informatique.

```
Hello world tel qu'il est proposé en 1978:
#include <stdio.h>
main()
{
    printf("hello, world\n");
}
```

ANSI C est une évolution normalisée du C K&R qui reste extrêmement compatible.

1 Création du projet embarqué

1.1 Repartir d'un environnement vide :



1.2 Ajouter un projet embarqué à votre environnement :

 \Rightarrow Créer un nouveau projet \Rightarrow commande \Rightarrow File \Rightarrow New \Rightarrow Embedded Projet.

⇒ Un projet nommé « Embedded_Projet.PrjEmb » apparaît dans l'onglet gestion de projet

D <u>X</u> P	<u>F</u> ile	<u>E</u> dit	<u>V</u> iew	Proje <u>c</u> t	<u>P</u> lace	De	sign	<u>T</u> ools	<u>S</u> imulator	<u>R</u> eports				
🗋 💕 层		<u>N</u> ew				•		<u>S</u> chemat	ic					
	2	<u>O</u> pen			Ctrl+0		8	Open <u>B</u> u	s System Doc	ument				
Projects		<u>I</u> mport				•	119	<u>P</u> CB						
		<u>C</u> lose			Ctrl+F4		V Ì	<u>V</u> HDL Do	ocument					
	1	Open P	roject				V	Verilo <u>g</u> D	ocument					
		Open D	esign V	Vor <u>k</u> spac	e		C	<u>C</u> Source	Document					
		Check (Dut				C	<u>C</u> ++ Sou	irce Docume	nt				
		Save			Ctrl+S		Б	C/C++ <u>F</u>	<u>l</u> eader Docur	ment				
		Save As					A	<u>A</u> SM Sou	irce Docume	nt				
		Save Co	opv As					Softwar <u>e</u>	Platform Do	cument				
		Save Al	1					Text Doc	ument					
		Save Dr	oiect A	-		_	30	CA <u>M</u> Do	cument					
		Save Pr	oject A:	orkenace	٨		<u>a</u>	Outp <u>u</u> t J	ob File					
		Save De	sign w	orkspace	A3	_	Ð	Database	e Lin <u>k</u> File					
		Link Sh	eet to V	ault				Project			•		PC <u>B</u> Project	٦
		Compo	nent Ke	elease Ma	anager	_		<u>L</u> ibrary			•		EPGA Project	
		Page Se	et <u>u</u> p					Script Fil	es		•		Co <u>r</u> e Project	
	4	Print Pr	e <u>v</u> iew					Mi <u>x</u> ed-Si	gnal Simulat	ion	•		Integrated Library	
		<u>P</u> rint			Ctrl+P			<u>O</u> ther			•	1	Embedded Project	
	1	S <u>m</u> art P	DF				a	Design <u>V</u>	<u>V</u> orkspace			51	Script Project	

TP : Ecrire, simuler et débuguer un programme C

1.3 <u>Définir le processeur auquel est destiné le projet embarqué.</u>

Remarque : définir le processeur revient à définir le compilateur c associé.

\Rightarrow Cliquez droit sur le projet e	embarqué		
\Rightarrow Cliquez sur Project option		Project Options	
\Rightarrow Déroulez la liste des proces	sseurs disponibles		
Options for Embedded Project Embedded_Pro	ject1.PrjEmb		<u>8</u> X
Compiler Options Files With Options Configur	e Memory Locate Options Sections/Reser	ved Areas	
		Device: UNKNOWN	
		Device Selection	Sélectionner le processeur TSK3000A
Set To Installation Defaults	Description	×	Cancel

 \Rightarrow Vérifiez que les options de compilation suivantes sont sélectionnées :

Altium Designer Release 10 (Platform 10.700.22943) - Embedded_Project1.PrjEmb. Licensed to	Lycée Jean Moulin. Not signed in.		
DXP <u>File View Project Window H</u> elp			1	• 0 • 0 • 4
💕 🗶 👼				
ects 🔻 🖌	×			2 2
orkspace1 DsnWrk	Options for Embedded Project Embedd	ded_Project1.PrjEmb		
mbedded Project1.PriEmb Project	Compiler Options Files With Options	Configure Memory Locate Options Sections/Reserved Areas		
File View Structure Editor	51 51		Device: TASKING ARM	
	Build Options	🗆 General		
Embedded_Project1.PrjEmb	Processor	Keep temporary files that are generated during a co	ompile	
No Documents Added	Ber C++ Compiler	Stop build process on error	✓	
	C Compiler	Silent mode	✓	
	H-Assembler	Use absolute path names in generated makefile		
	⊞- Linker	Use additional make options		
		Additional make options		
		Use user-defined makefile		
		User-defined makefile		
		Directories		
		Executable files path		
		Include files path		
		Library files path		
		Library source files path	\$(PRODDIR)\carm\lib\src	
		Debug search path		
		Output directory (instead of project directory)	Output	
		Software Platform search path		
		Add default Software Platform location	✓	
		Default Software Platform Version		
sages				
ss Docu S Message Time Date M	4			
		Description		
	Set To Installation Defaults		UK	Lancel
	C.			
			System Design (Compiler Help Instruments VHDL
figure Project Options for Focused Project				
			1000	10:46
🧊 🖳 🖸 🚺 🖤 🗶				▲
				31/01/2012

Formation ALTIUM - 5	TP_Compliler_Simuler_Debugger_un fichier_C	03/04/2013	3/8	
----------------------	--	------------	-----	--

2 Ecriture du programme C « Hello World ! »

 \Rightarrow Cliquez droit sur le projet embarqué et adjoindre un fichier C :

 \Rightarrow Add New to Project \Rightarrow C File.



 \Rightarrow Dans ce fichier C recopiez le code ci-dessous :

```
#include <stdio.h>
void printloop(void)
{
    int loop;
    for (loop=0; loop<10; loop++)
    {
        printf("%i\n",loop);
    }
}
void main(void)
{
    printf("Hello World!\n");
    printloop();
}</pre>
```

 \Rightarrow Sauvegarder ce document : File \Rightarrow Save As « *main.c* » dans le dossier parent du projet. \Rightarrow . Sauvegarder le projet à nouveau : clic droit sur son nom \Rightarrow Save Projet.

ojects				▼ 🖉 ×
Workspace1.DsnWr	k		•	Workspace
Embedded_Project1	.PrjEn	ıb		Project
) File View 🔘 Stru	ucture	Editor		•ان 🔄
	Proje	ect1.PrjEmb *		B
	cumer	nts		-
		Compile <u>D</u> ocume	nt C_S	Source1.c
		Hide		
		<u>C</u> lose		
	(B)	Remove from Proi	iect	
		Save	Jeeum	
		<u>></u> ave		
		Save <u>A</u> s		

 $\Rightarrow \text{Compilez le projet embarqué}: \Rightarrow \text{Cliquez droit sur le projet} \\\Rightarrow \text{Cliquez sur Compile Embbeded Project}$

 \Rightarrow Si des erreurs apparaissent, corrigez les et recompiler !

3 Mise au point du programme en langage C :

3.1 Simulation du programme avec un point d'arrêt :

Lorsque vous ouvrez la fenêtre d'édition du programme Hello_World.c les fonctions de simulation sont implicitement actives.



⇒ Ouvrez les fenêtres **breakpoint** et **Debug** à partir de l'onglet **Embedded** (en bas à droite)

	System Design Compiler Embedded	<u>H</u> elp <u>I</u> nstru	uments
Formation ALTIUM - 5	TP_Compliler_Simuler_Debugger_un fichier_C	03/04/2013	5/8

Debug Console est la sortie par défaut des fonctions stdio donc de la fonction printf ();

 \Rightarrow Exécution du programme sans point d'arrêt : Run the embbeded program



Debug Console	▼ X
> C Hello World! 0 1 2 3 4	Le flux de sortie peut être visualisé dans la fenêtre Debug Console
5 6 7 8 9 _Exit: j _Exit > C	E
	✓ Send Halt

 \Rightarrow Arrêter l'exécution du programme bouton stop, puis effectuer un reset avec retour en début du programme C :



 \Rightarrow Pour ajouter un point d'arrêt ligne 10, (**printf**(« %i \n, loop), placez le curseur dessus

- \Rightarrow Cliquez droit
- \Rightarrow Toggle Breakpoint

<pre>#include <stdio.h></stdio.h></pre>									
void printloop(void)									
{ int loop;			Filename/A	s Address Line Nu	ımber	Count	Reset Count	Condition	In threads
\perp		\							
<pre>for (loop=0; loop<10 { printf("%i\n",lo</pre>	; 100	op++)		a					
<pre> for (loop=0; loop<10 { printf("%i\n", lo } </pre>	; 100	pp++) <u>T</u> oggle Breakpoint	F5						
<pre> for (loop=0; loop<10 { printf("%i\n",lo } } } </pre>	; loo	np++) Toggle Breakpoint Add <u>W</u> atch	F5 Ctrl+F5						
<pre>For (loop=0; loop<10 { printf("%i\n",lo } void main(void)</pre>	; loo	<u>T</u> oggle Breakpoint Add <u>W</u> atch <u>E</u> valuate	F5 Ctrl+F5 Ctrl+F7						
<pre>For (loop=0; loop<10 for (loop=0; loop<10 printf("%i\n", loop=0; } void main(void) for the printf("%i\n", loop=0; loop<10; for the printf("%i\n", loop=0; loop=0; loop=0; for the printf("%i\n", loop=0; loop=0; loop=0; loop=0; for the printf("%i\n", loop=0; l</pre>		Toggle Breakpoint Add Watch Evaluate Show Disassembly	F5 Ctrl+F5 Ctrl+F7						
<pre>For (loop=0; loop<10 for (loop=0; loop<10 for printf("%i\n",lo } void main(void) for printf("Hello World!</pre>	; 100	Toggle Breakpoint Add Watch Evaluate Show Disassembly Push And Export Fror	F5 Ctrl+F5 Ctrl+F7 m Hardware	-					

Breakpoints					-
Filename/Address	Line Number	Count	Reset Count	Condition	In threads
⊗ Hello_World.c	10	0	0		
∑ Debug Console → B	reakpoints /				

 \Rightarrow Exécution du programme avec les points d'arrêts : Run the embbeded program



ebug Console			▼ >
> C			
Hello World!			
printloop#10:	printt("%i\n",loop];		
u printloop#10:	printf("%) p" loop):		
	pinia(>8141 (100p))		
1			
printloop#10:	printf(''%i\n'',loop);		
> C			
2			
printloop#10:	printf(''%i\n'',loop);		
> U 2			Ξ
3 printloop#10:	printf("%i\n" loop):		
pinicop#10.	pinia(>=i'ii ,ioop),		
		 Send	Halt
Debug Consol	le Breakpoints		

3.2 Visualisation d'une variable :

$\Rightarrow A \text{ partir de l'ongle} \lor Watches}$	et Embedded sélectionnez Watches	Insérer une nouvelle image pour préciser comment on ajoute une expression
	Add Watch	Afaire < !!!!!
	Lexpression Loop ▼	
	Eormat OK Cancel	
Formation ALTIUM	- 5 TP_Compliler_Simuler_Debugger_un fichier_C	03/04/2013 7/8

- \Rightarrow Visualisation de la variable « loop ».
- \Rightarrow Placer un point d'arrêt dans la fonction **«printloop() ; »**.
- \Rightarrow Relancer le programme à chaque arrêt : la variable « **loop** » s'incrémente.

Watches		▼ X
Expression	Value	A
60^ [loop	6	
Debug Console	Watches Breakpoints	

3.3 Visualisation des registres SFR du processeur TSK3000A

\Rightarrow A partir de l'onglet **Embedded** sélectionnez **Registers**

Registers

Current CPU status: 0 ack vie ite itr umo ieo ump iep umc iec						
Register	Width	Decimal	Hexadecimal	Binary	Char	Τ.
0 Status	32	0	00000000	0000-0000-0000-00		
1 IEnable	32	0	00000000	0000-0000-0000-00		:
2 IPending	32	0	00000000	0000-0000-0000-00		
3 IMode	32	0	00000000	0000-0000-0000-00		
4 TBLO	32	6678	00001A16	0000-0000-0000-00		
5 TBHI	32	0	00000000	0000-0000-0000-00		
6 PIT	32	42949672	FFFFFFF	1111-1111-1111-11		
7 ER	32	0	00000000	0000-0000-0000-00		
8 EB	32	256	00000100	0000-0000-0000-00		
9 PC	32	64	00000040	0000-0000-0000-00	@	
10 HI	32	0	00000000	0000-0000-0000-00		
11 LO	32	0	00000000	0000-0000-0000-00		

3.4 Exécution du programme en mode pas à pas :

 \Rightarrow Touche F7 : Step into the current source line



