

Matière : Systèmes sur puces SoC

Compte rendu du TP3

Travail réalisé par :

Mokhles El Benna

Développement d'un système purement logiciel Cas d'un processeur communiquant avec une machine hôte (UART)

Table des matières

Objectif :	3
Logiciel nécessaire :	3
Matériel nécessaire :	3
Partie Théorique :	3
Première partie :	3
Description du système	4
Manipulation :	4
Deuxième partie :	14
Travail demandé :	21
Exercice 1 :	21
Exercice 2 :	21

Développement d'un système purement logiciel Cas d'un processeur communiquant avec une machine hôte (UART)

Objectif :

L'objectif de ce TP est de mettre en œuvre une communication entre un soft processeur (ARM Cortex-A9) inclus dans la carte FPGA ZYBOZ7-10 et un HyperTerminal sur une station hôte (PC). La communication est faite à travers un bloc UART instancié dans le FPGA en tant que softIP. Une application élémentaire va s'exécuter sur le processeur ARM Cortex-A9 et les données seront affichées sur le terminal par l'intermédiaire de la liaison RS232.

Vous serez en mesure de après la manipulation :

- Créer un projet Vivado pour un système Zynq
- Utilisez IP Integrator pour créer un système matériel
- Utilisez le SDK pour créer un projet de test de mémoire standard
- Exécutez l'application de test sur la carte

Logiciel nécessaire :

- Xilinx Vivado 2015.1 avec le package SDK

Matériel nécessaire :

- Carte de développement Digilent ZYBOZynq™-7000

- un câble micro USB pour la communication UART et la programmation JTAG.

Partie Théorique :

Première partie :

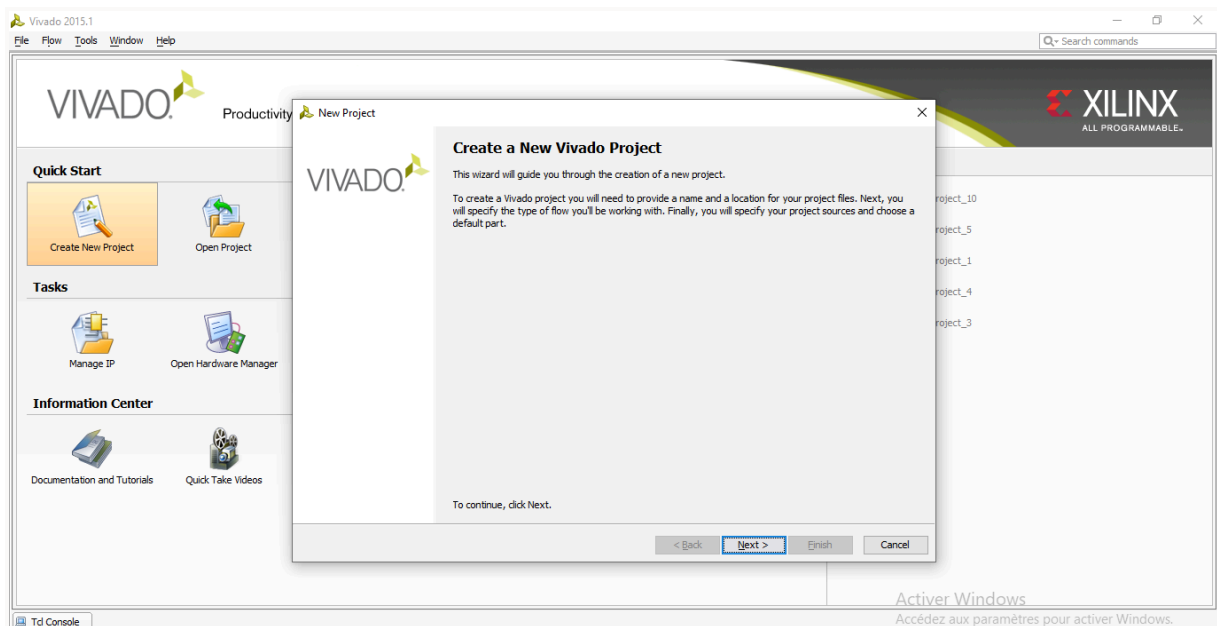
Ce TP est divisé en étapes qui consistent en des déclarations générales qui fournissent des informations sur les instructions détaillées qui suivent. Suivez ces instructions détaillées pour progresser dans cette manipulation. Cette Manipulation comprend 5 étapes principales: vous allez créer un projet de haut niveau à l'aide de Vivado, créer le Processing System à l'aide de Vivado IP Integrator, générer le HDL de niveau supérieur et exporter la conception vers le SDK, créer une application de test de mémoire dans le SDK, et enfin, testez dans le matériel.

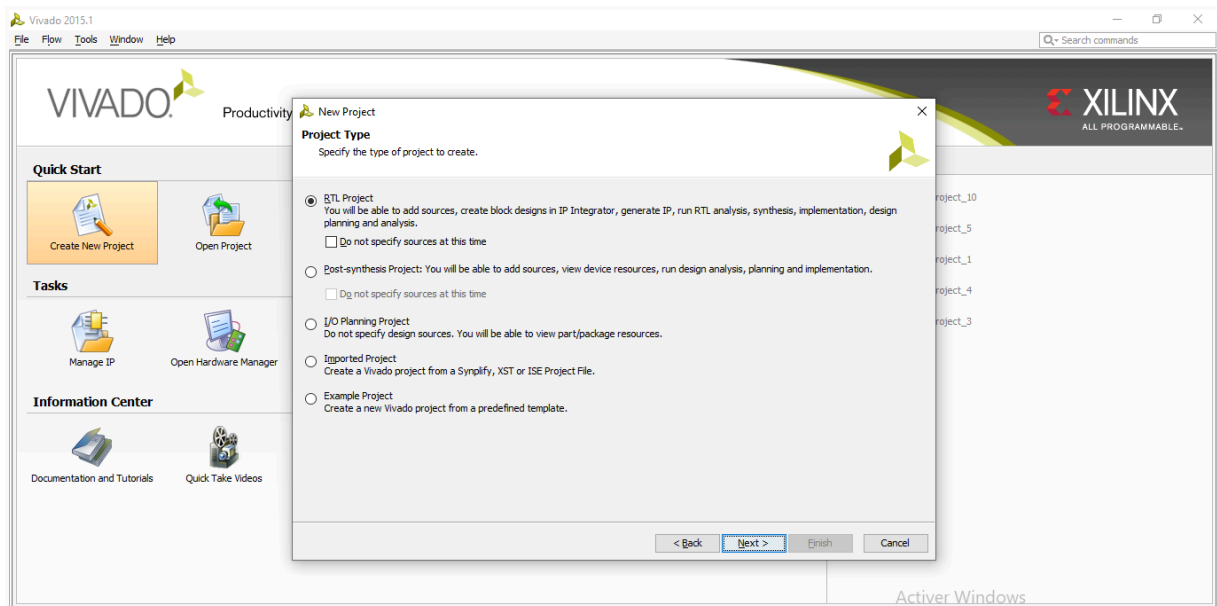
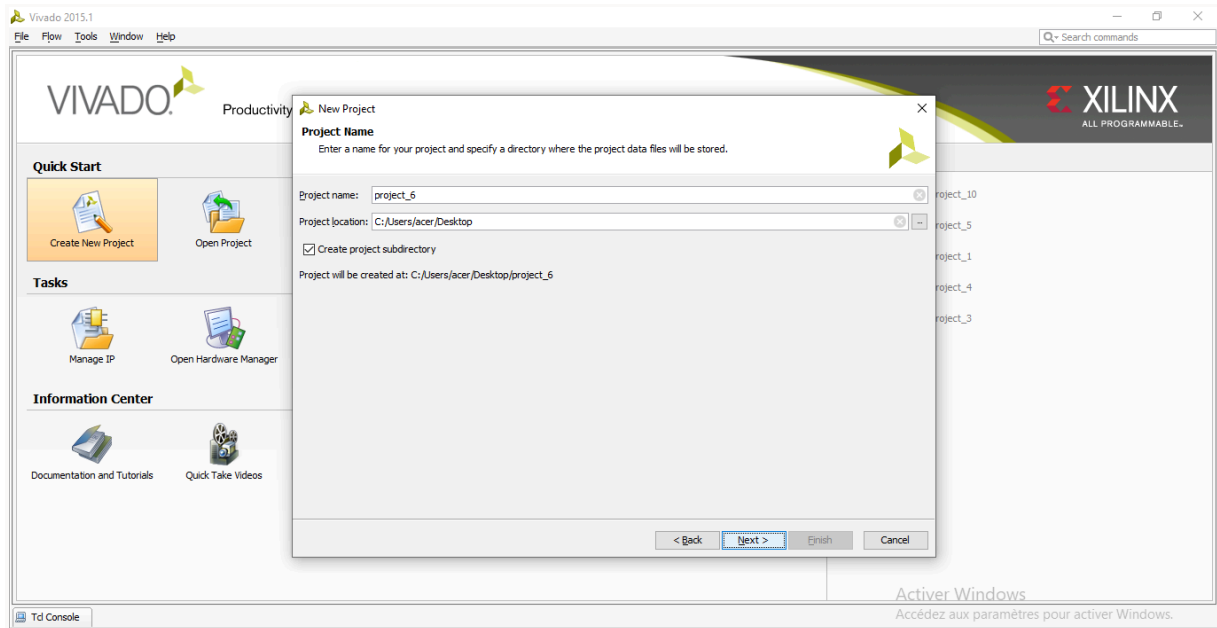
Description du système

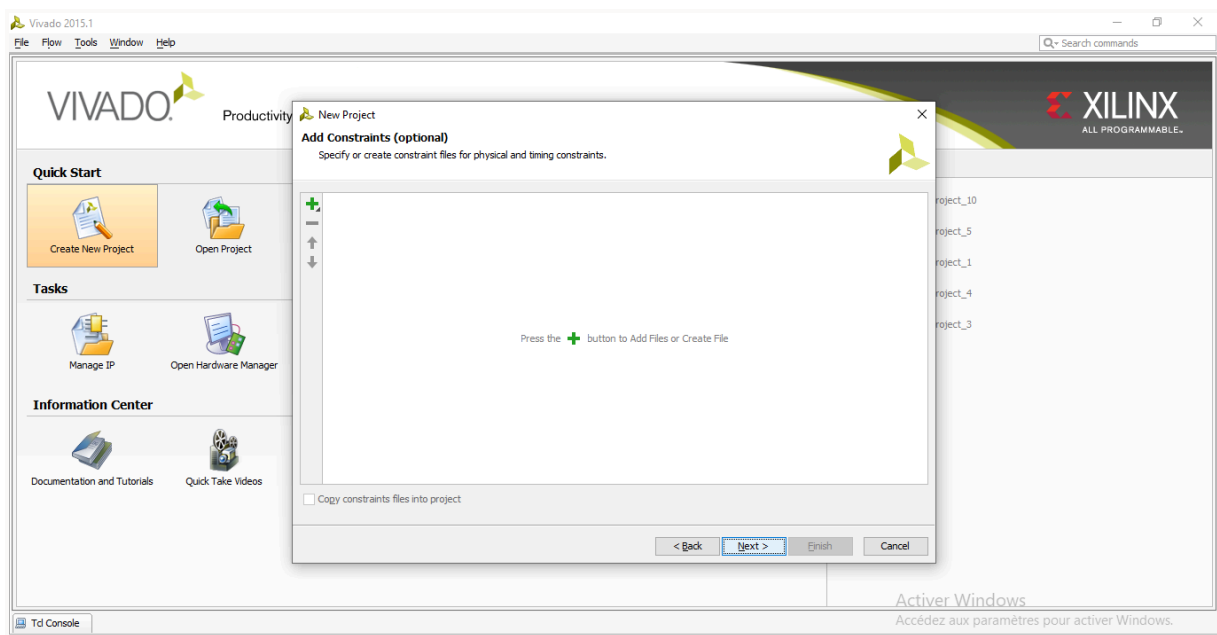
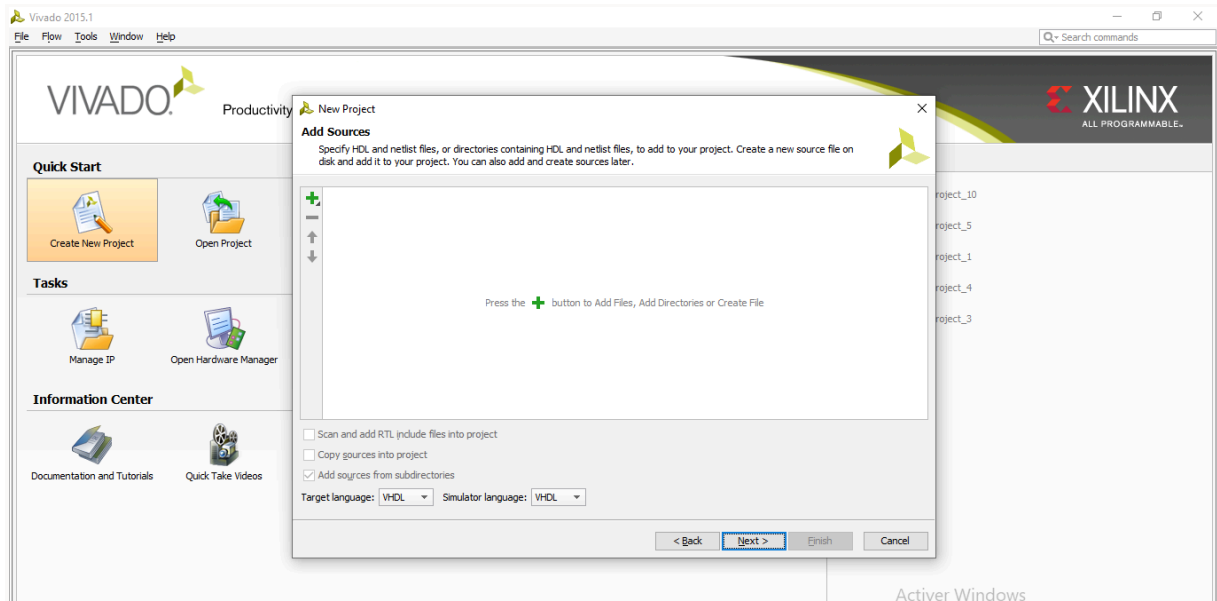
Le but des exercices de cette manipulation est de vous guider à travers une conception complète du système de processeur matériel et logiciel.

Manipulation :

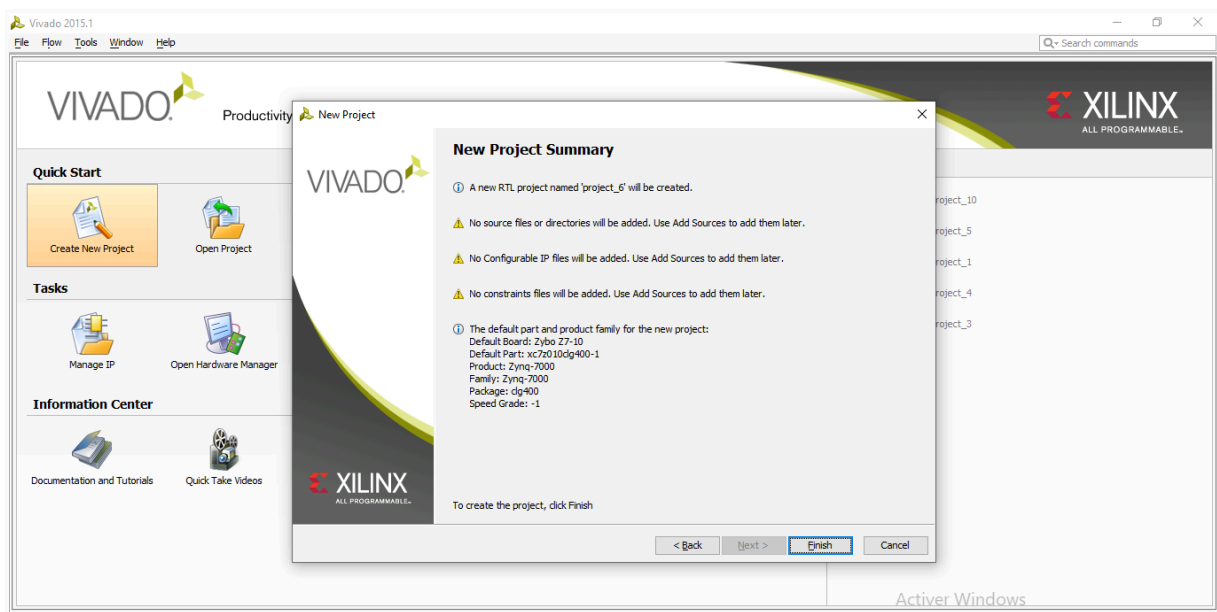
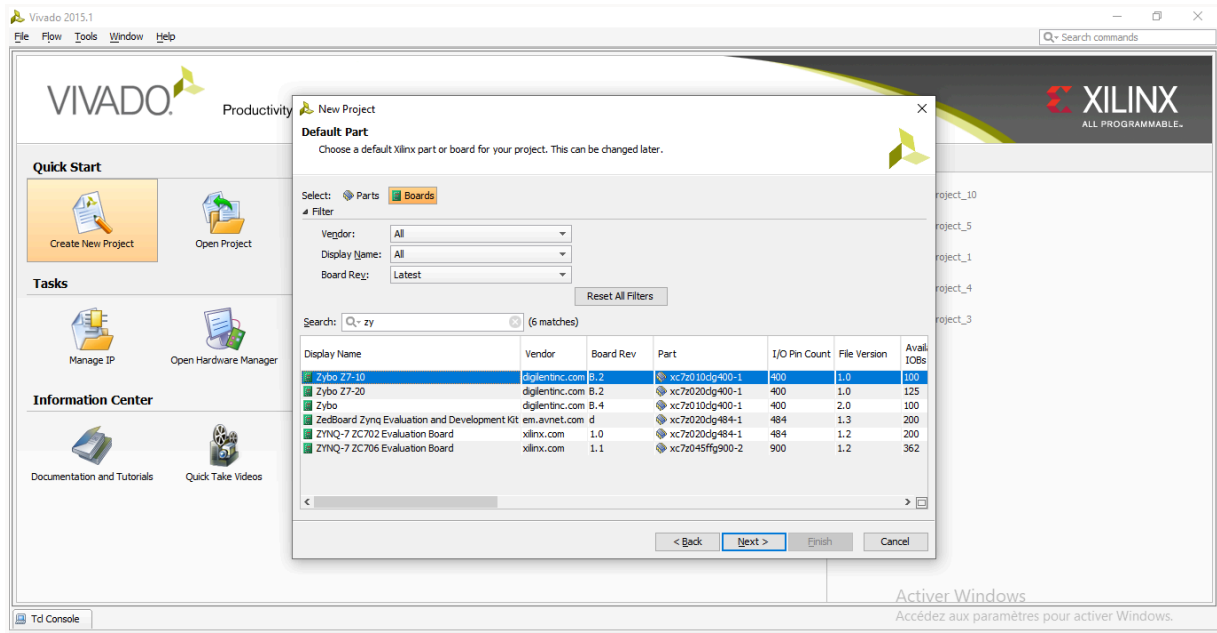
On va créer un nouveau projet



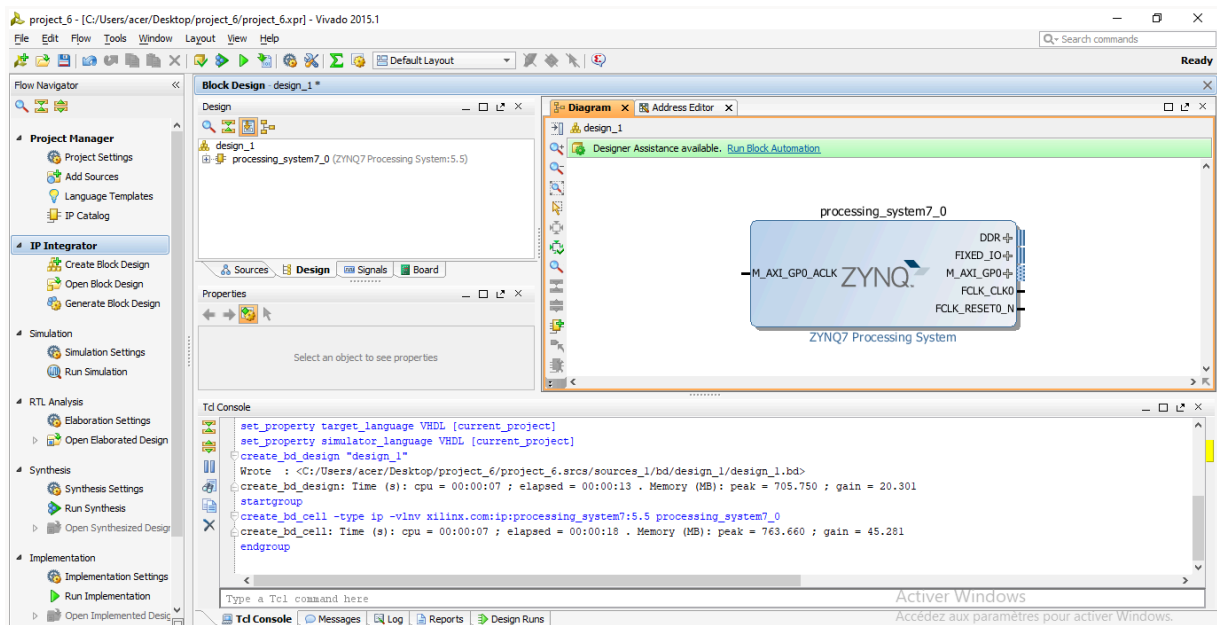
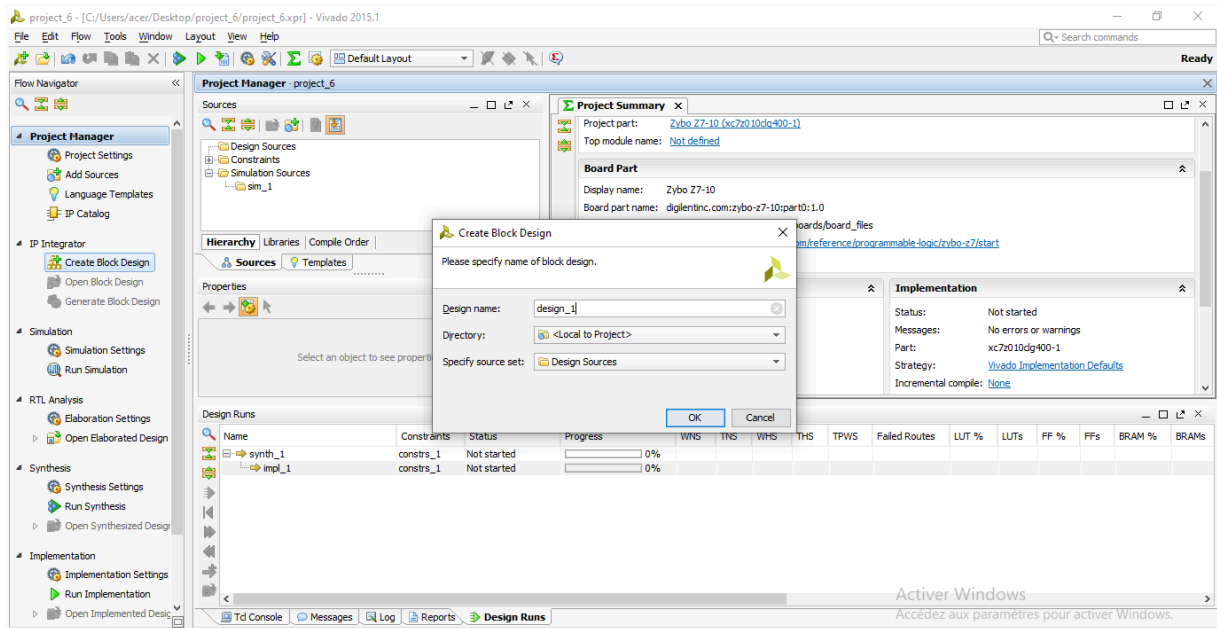




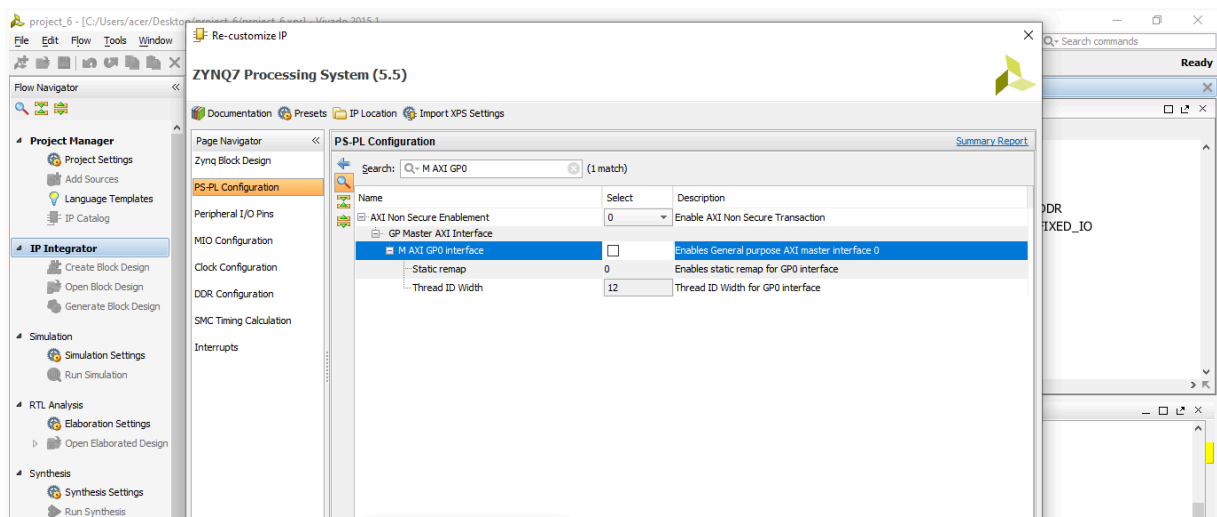
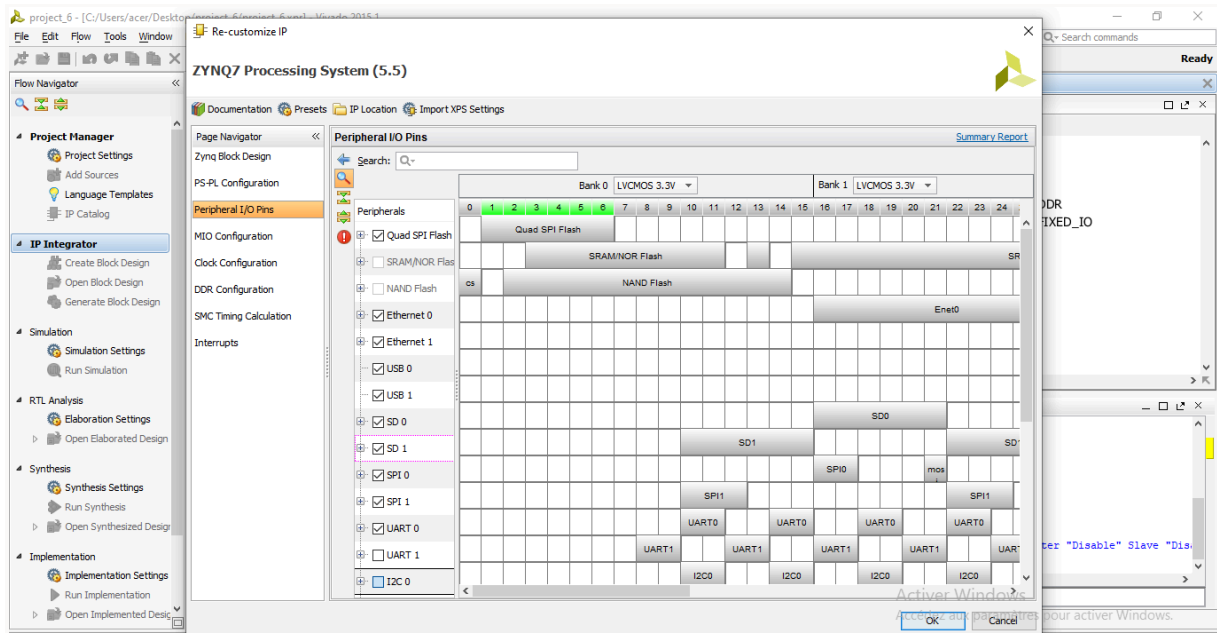
On a vérifié que « VHDL » est le langage sélectionné ».Puis, on clique sur le bouton « Boards » puis sur Zybo Z7-10. Enfin, on clique sur « Finish».

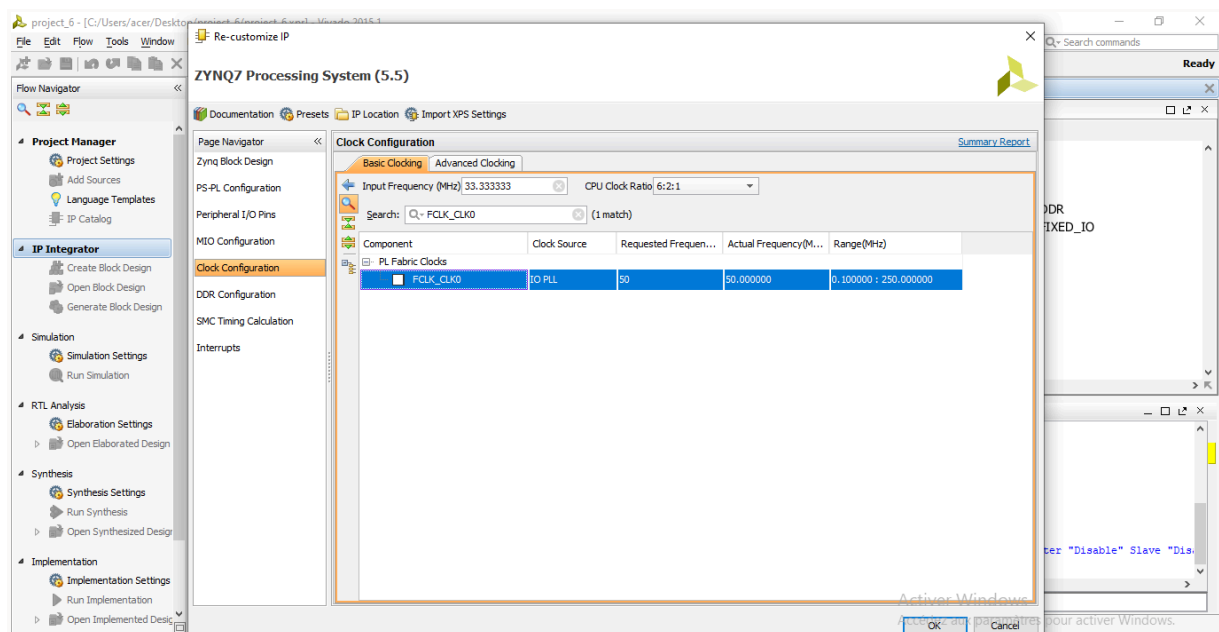
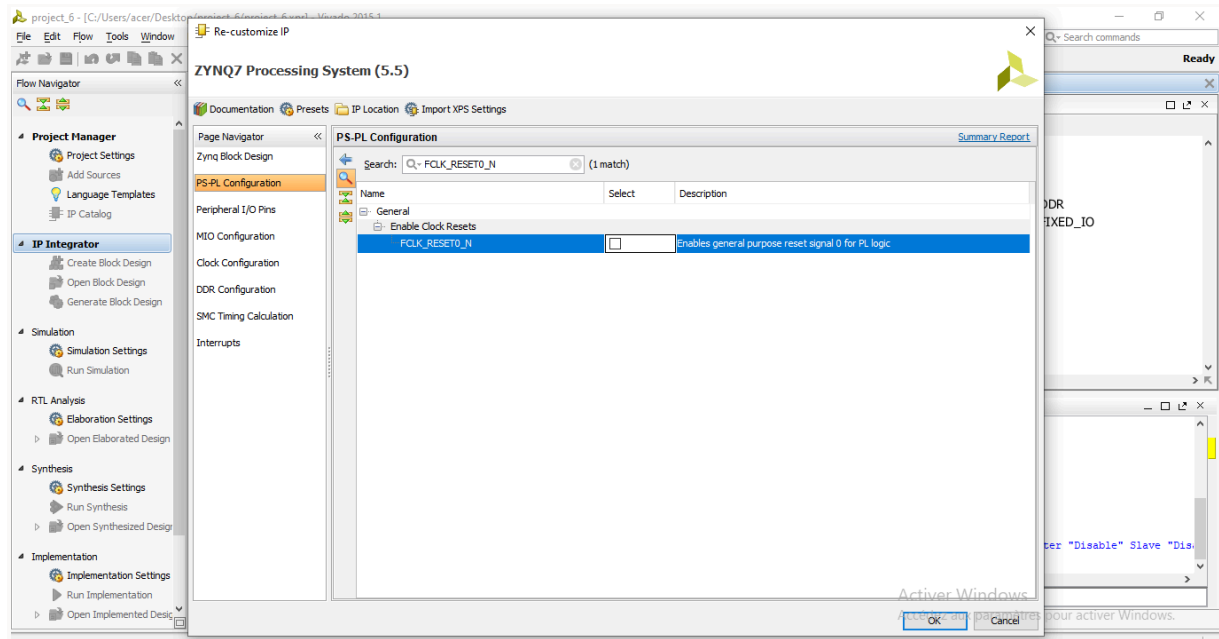


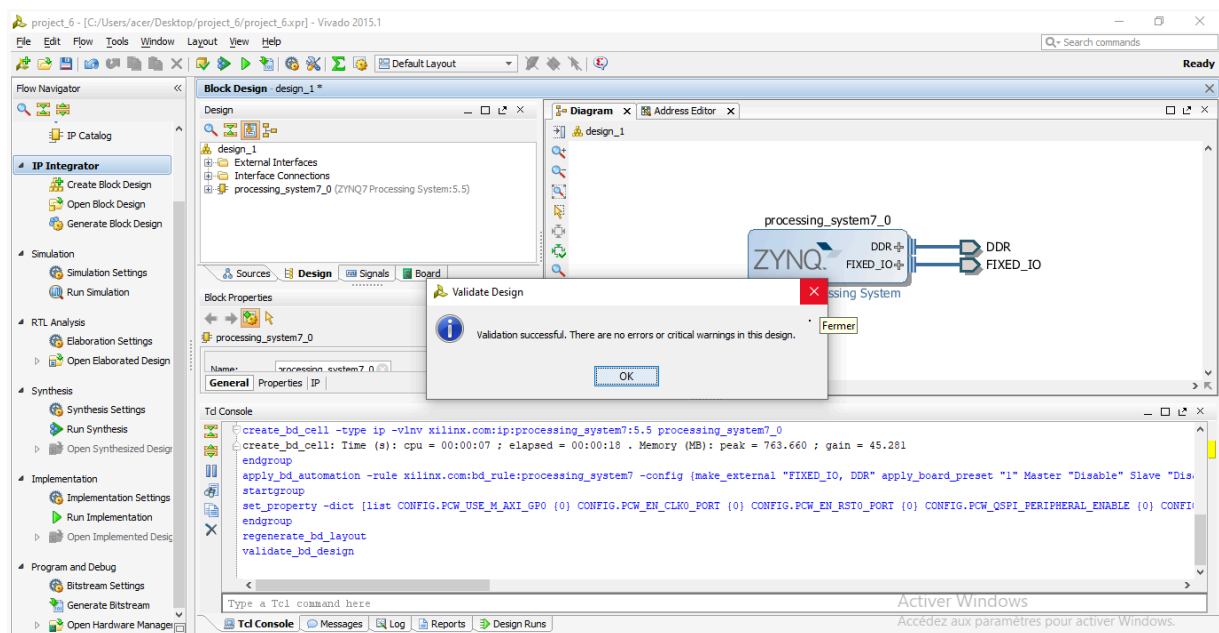
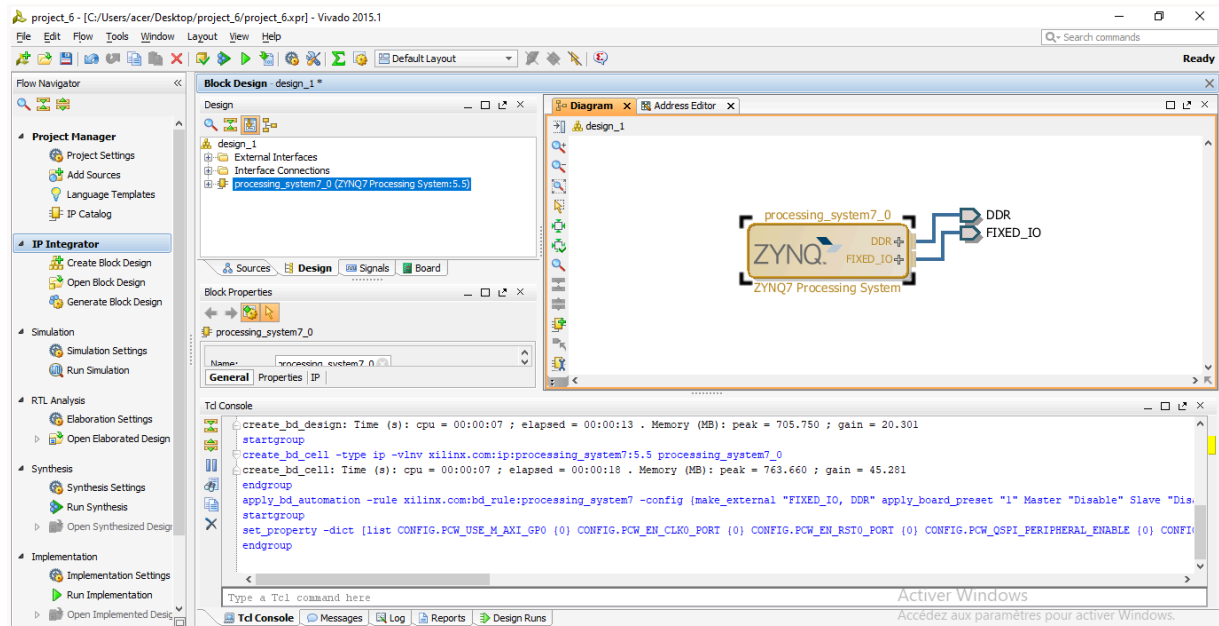
Dans un premier lieu, on va configurer le système ARM aussi bien que ses périphériques en créant un Block Design



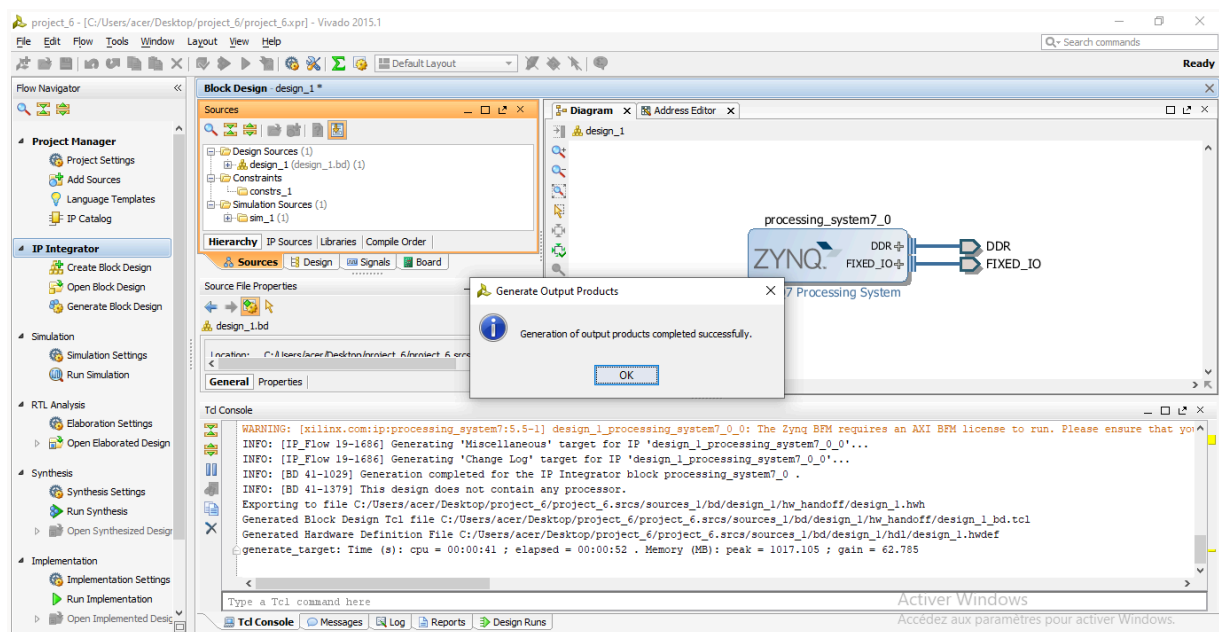
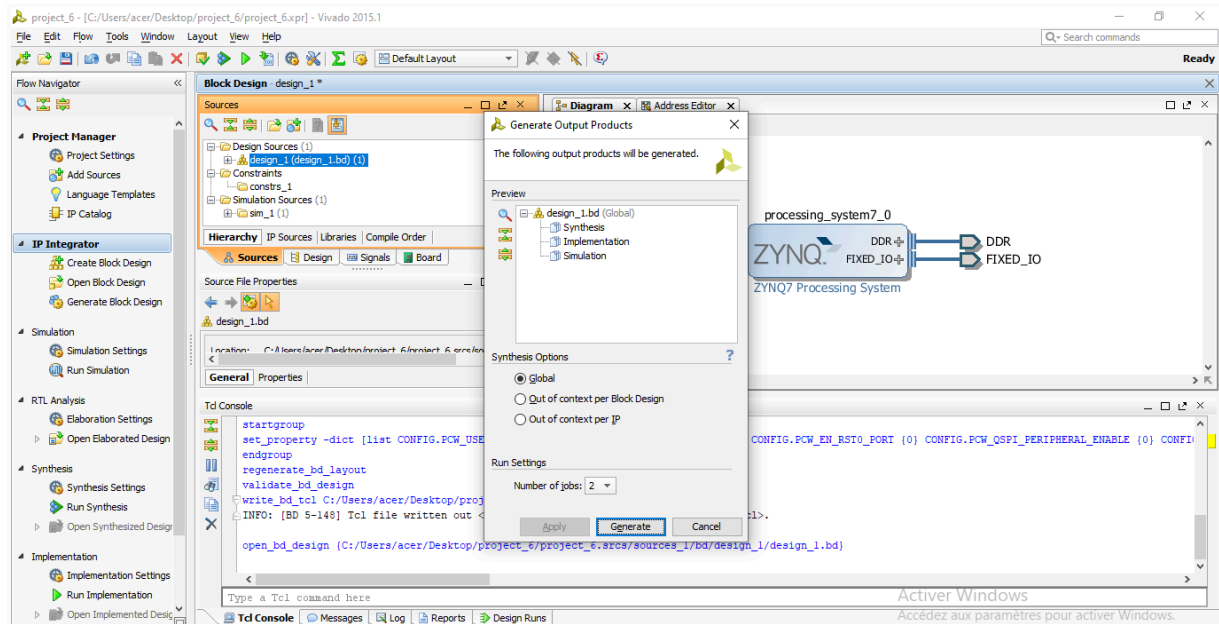
On Développe les périphériques d'E/S (I/O peripherals) et on désélectionne tous les périphériques d'E/S sauf UART 1.

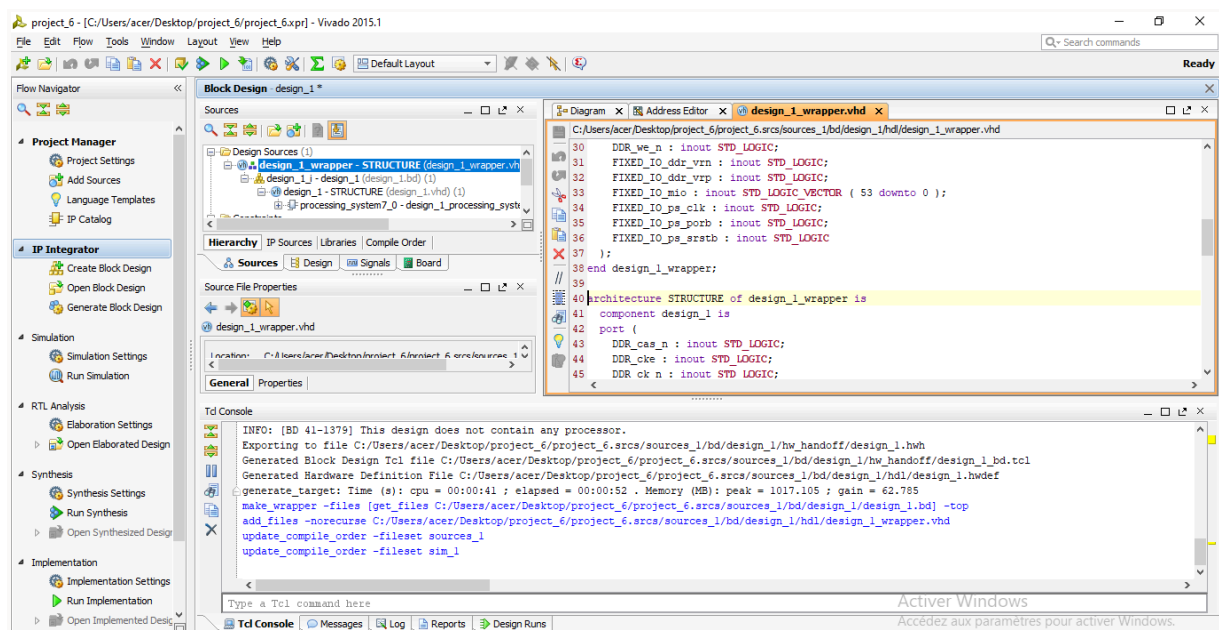
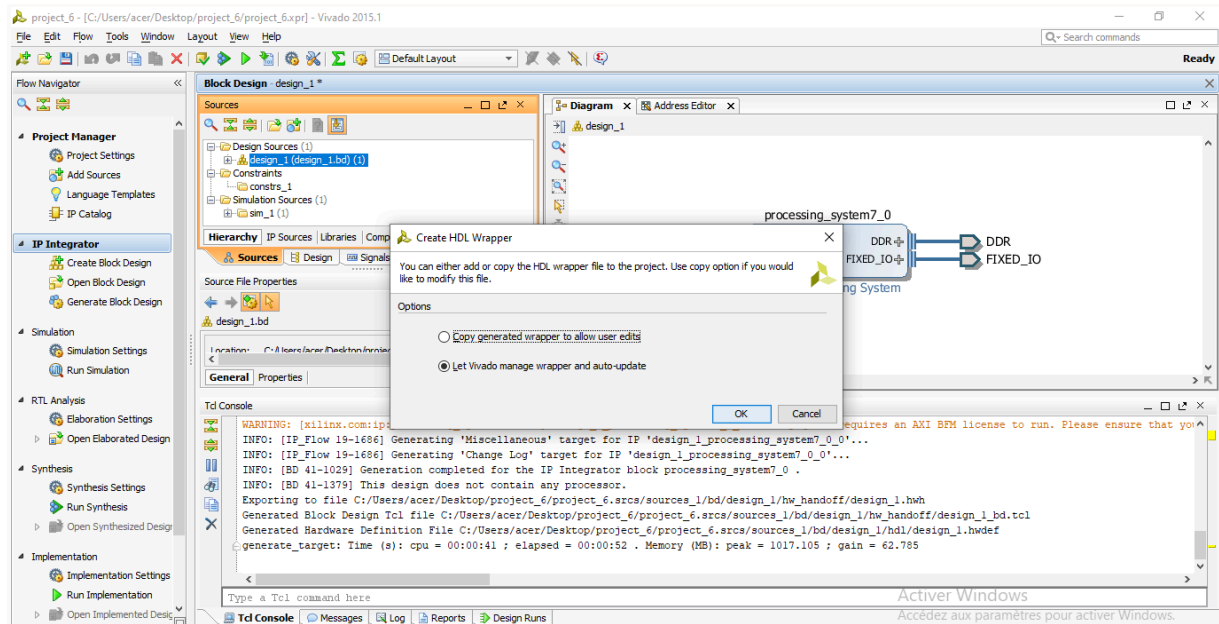




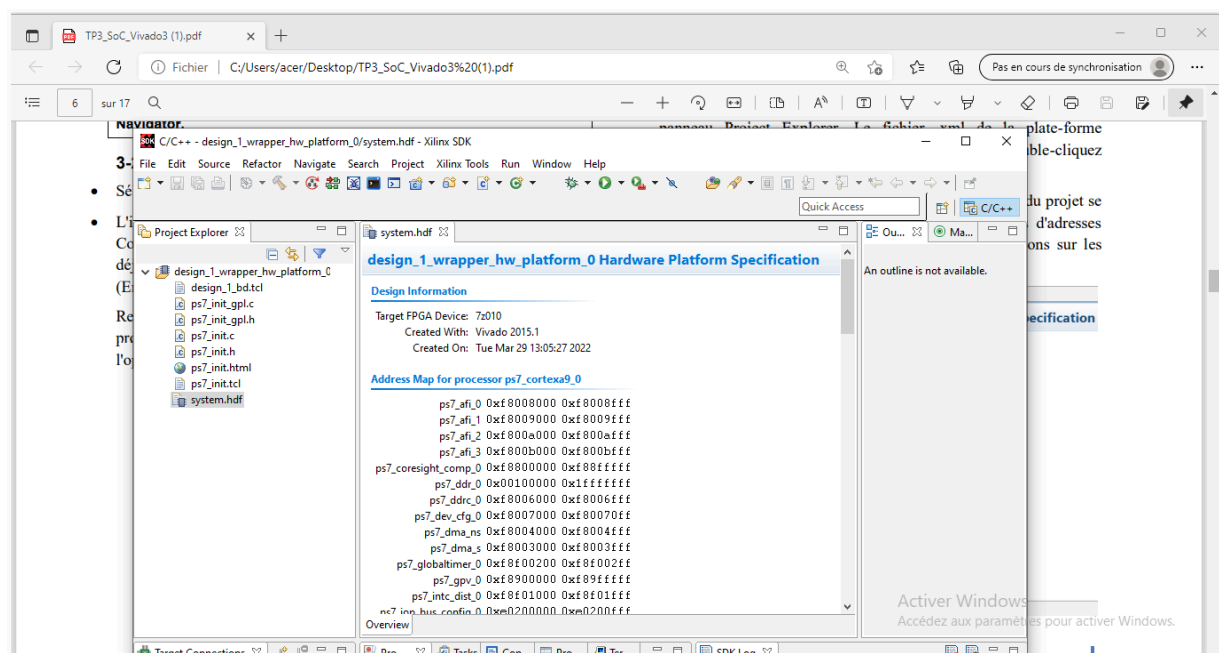
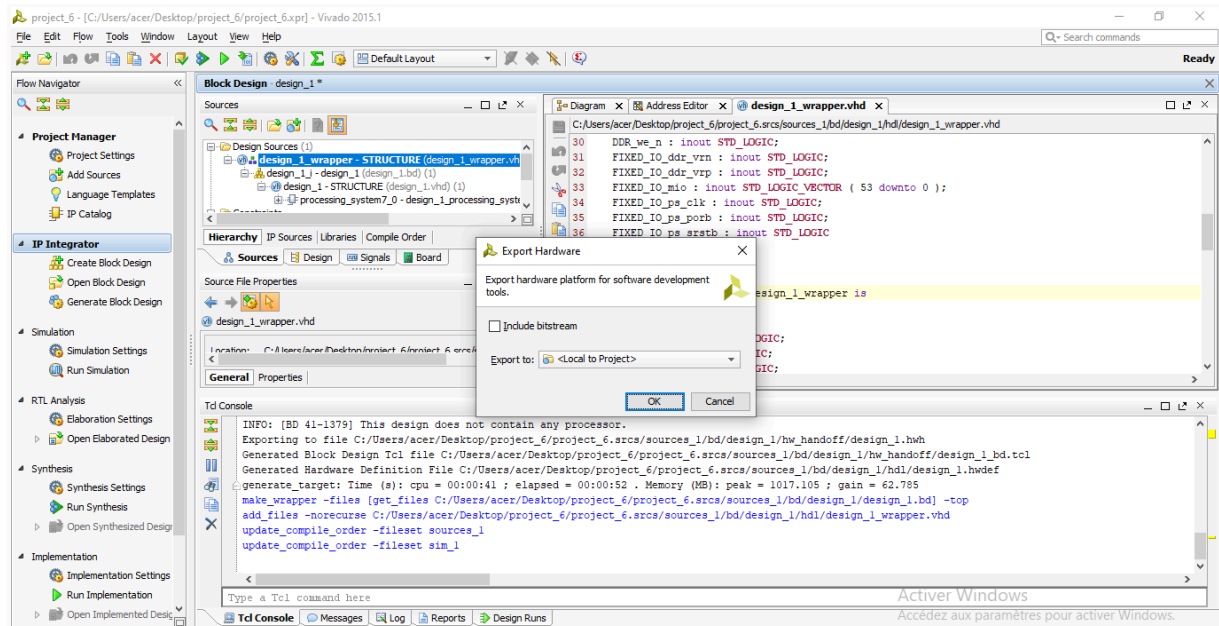


Dans le panneau des sources, on clique avec le bouton droit sur system.bd et on choisit Generate « Output Product » puis on clique sur Generate pour générer les fichiers d'implémentation, de simulation et de synthèse pour le design.



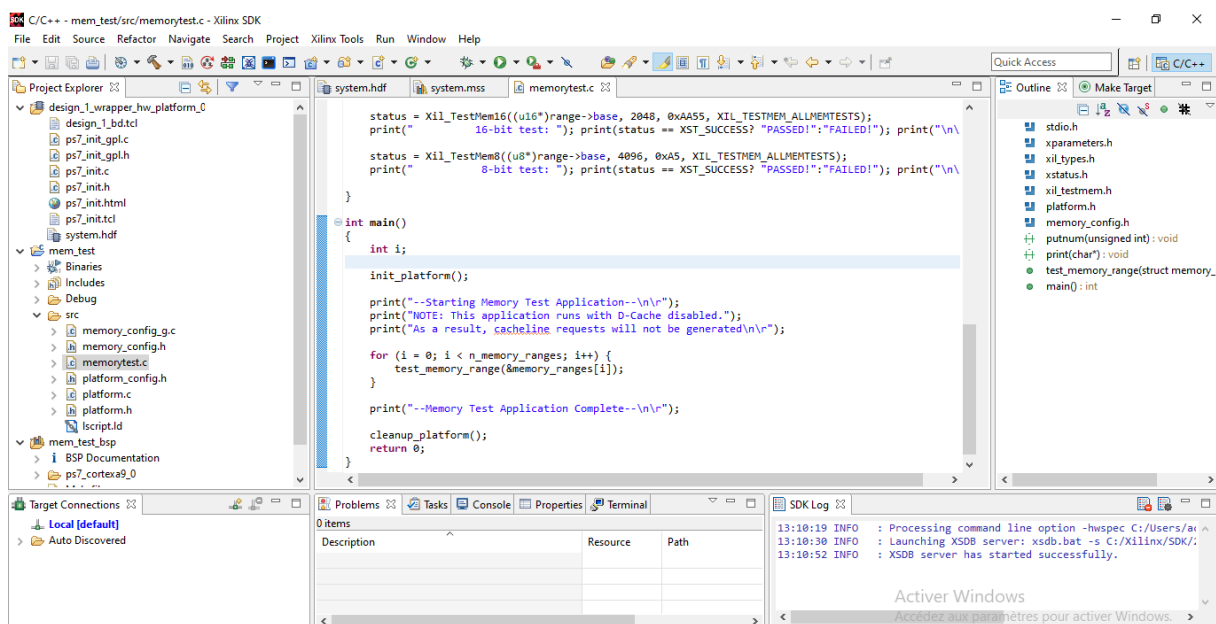
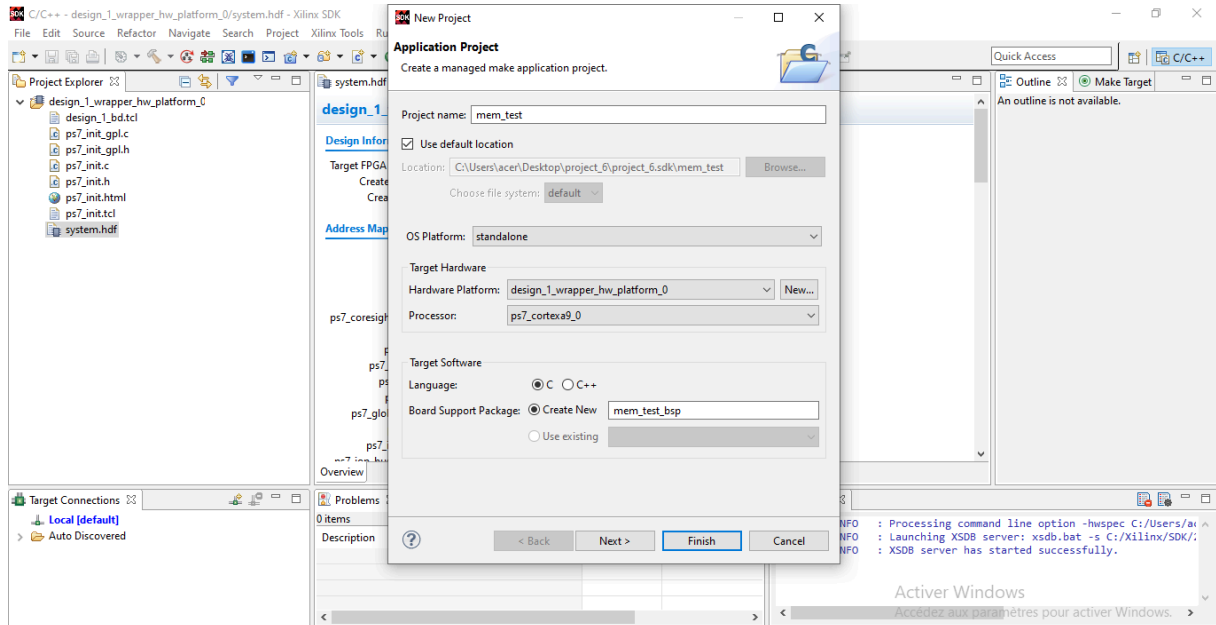


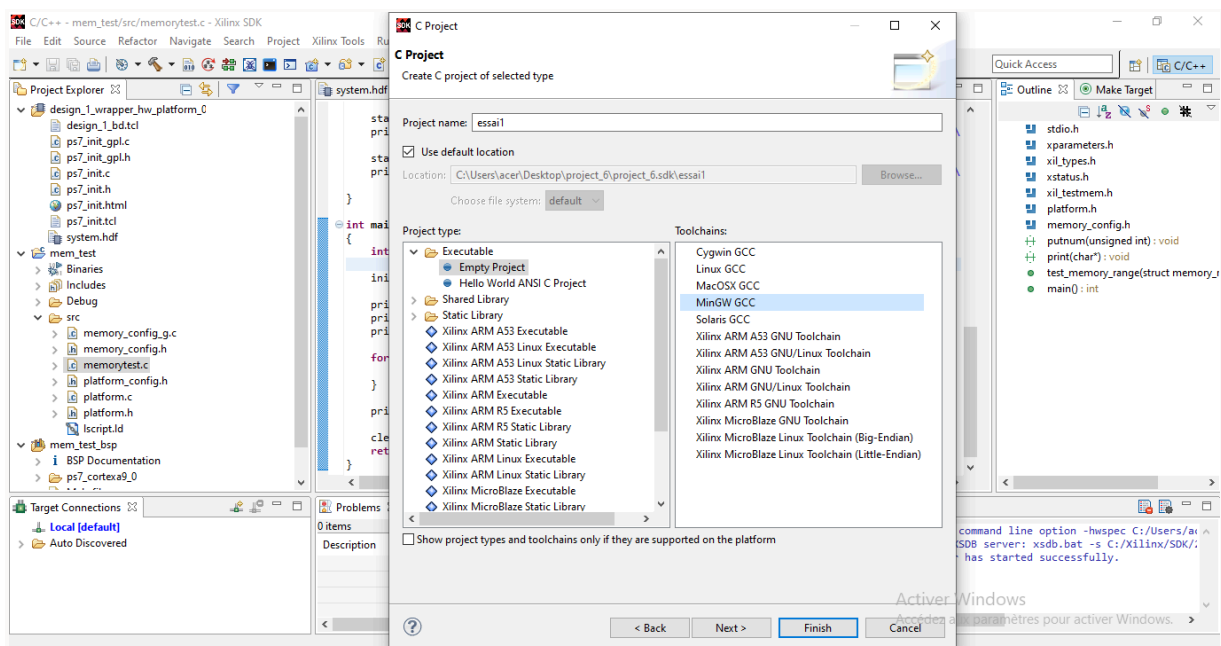
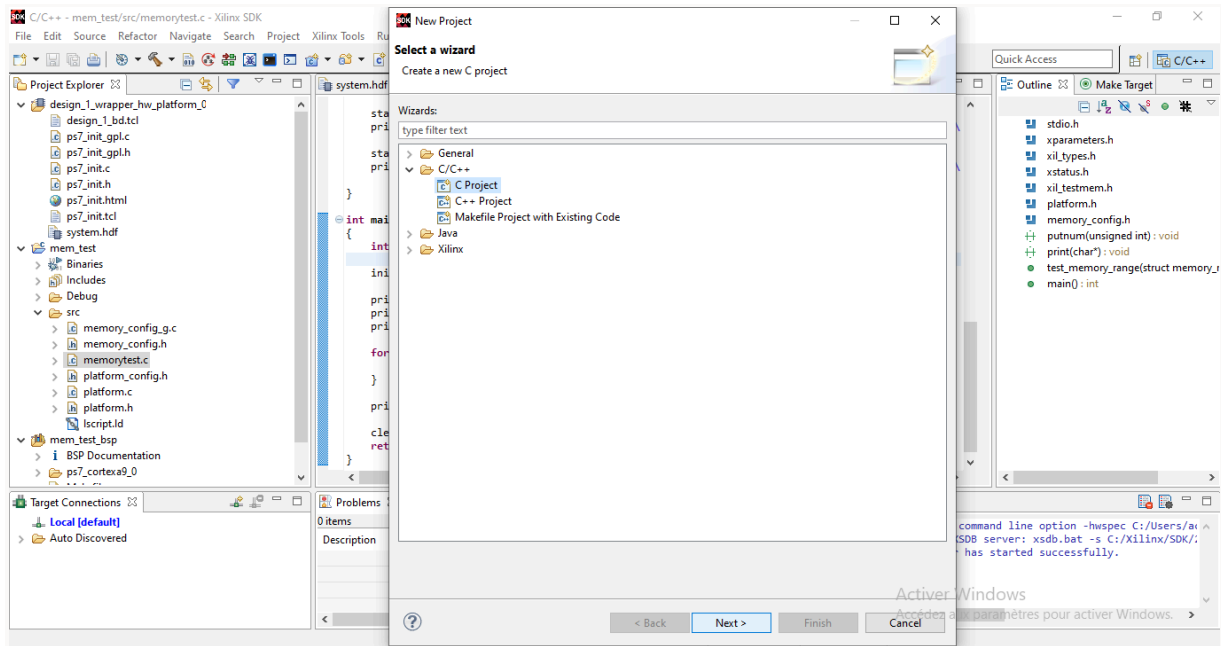
On Démarre le SDK en exportant le matériel

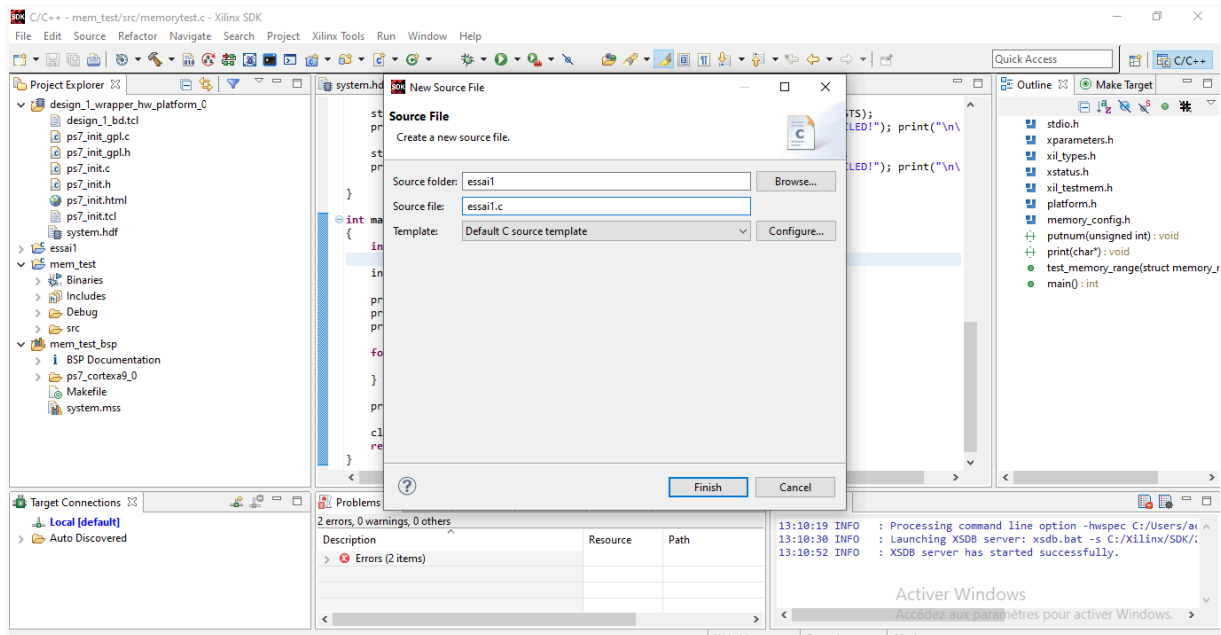


Deuxième partie :

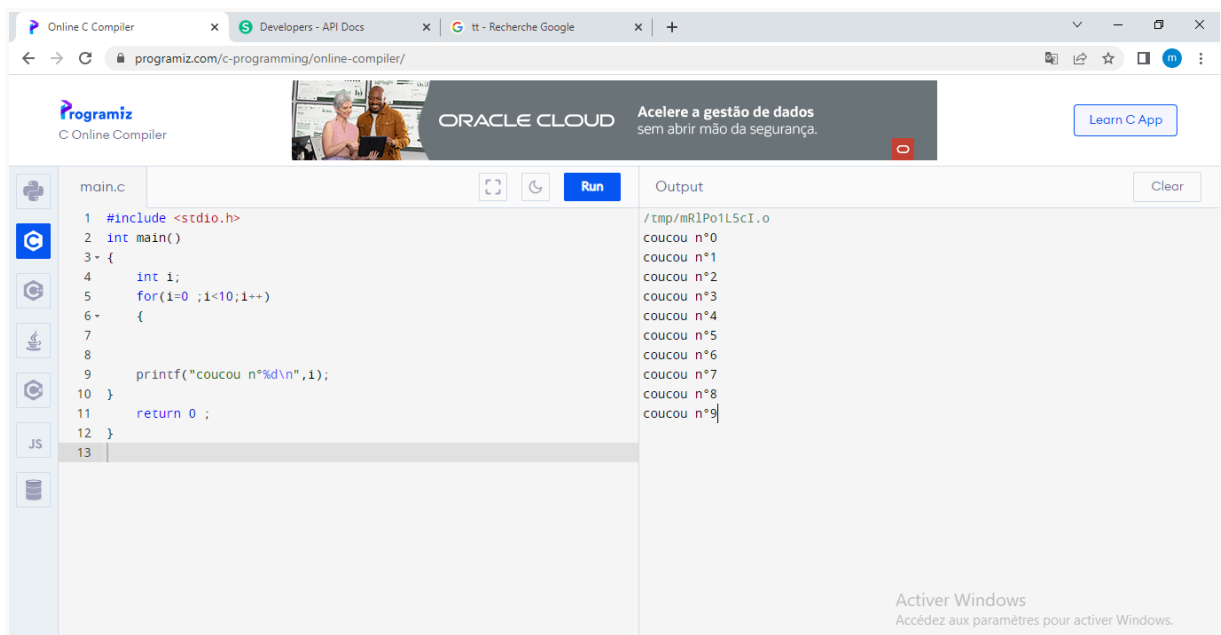
Dans Sdk on a crée un nouveau projet , Dans la fenêtre suivante, on clique sur Empty Project, on tape le nom du projet (essai1) et on sélectionne MinGW GCC

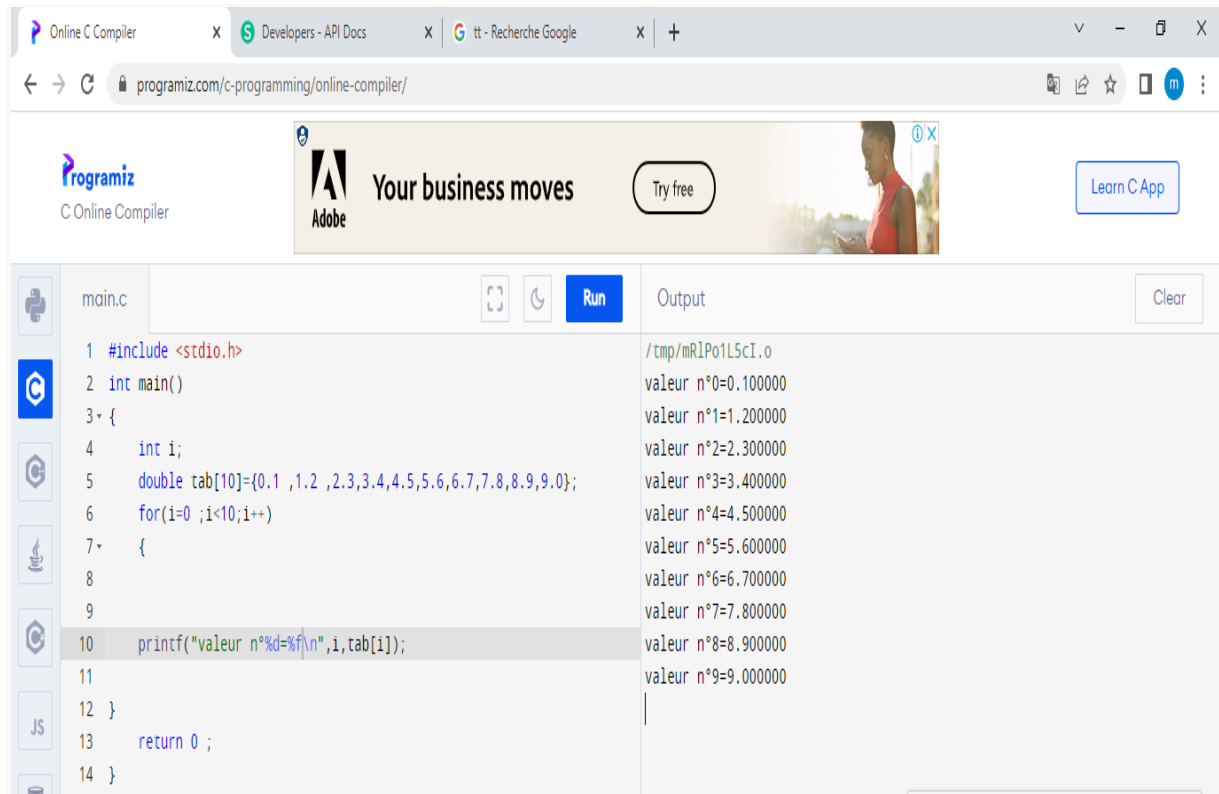






On va exécuter les programmes dans un compilateur C.





The screenshot shows the Programiz Online C Compiler web interface. The browser tabs include 'Online C Compiler', 'Developers - API Docs', and 'tt - Recherche Google'. The address bar shows 'programiz.com/c-programming/online-compiler/'. The main area is divided into a code editor on the left and an output window on the right. The code editor contains a C program that prints the values of an array. The output window shows the execution results.

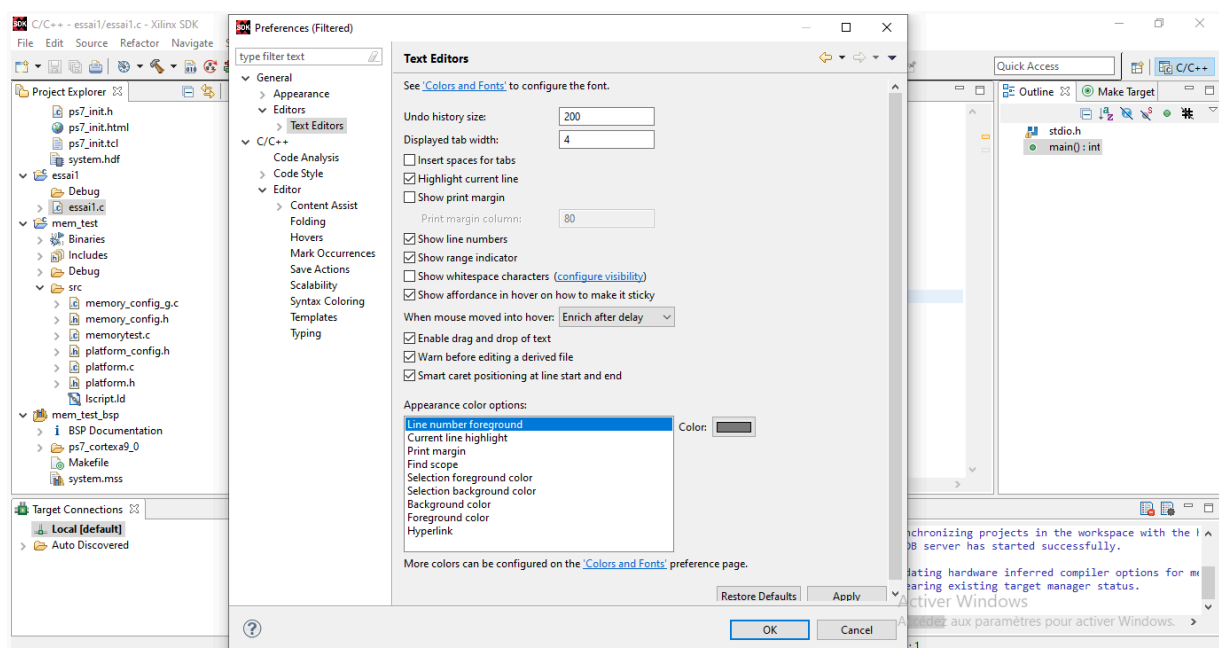
```

1 #include <stdio.h>
2 int main()
3 {
4     int i;
5     double tab[10]={0.1,1.2,2.3,3.4,4.5,5.6,6.7,7.8,8.9,9.0};
6     for(i=0 ;i<10;i++)
7     {
8
9
10    printf("valeur n°%d=%f\n",i,tab[i]);
11
12 }
13 return 0 ;
14 }
  
```

Output:

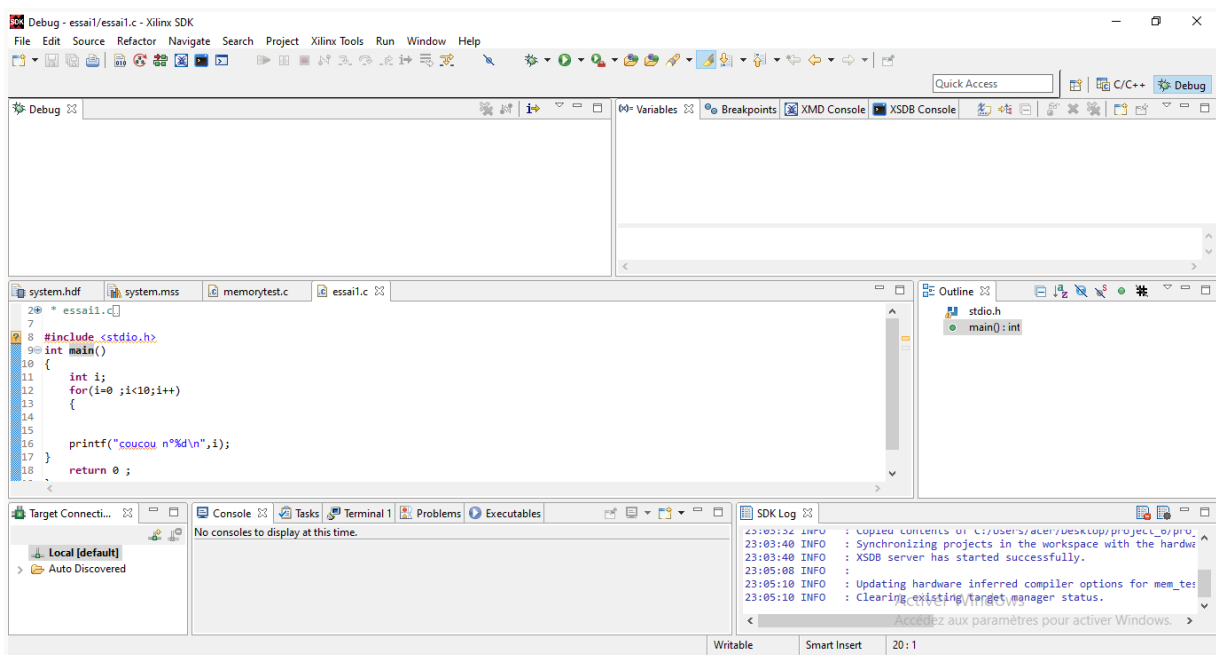
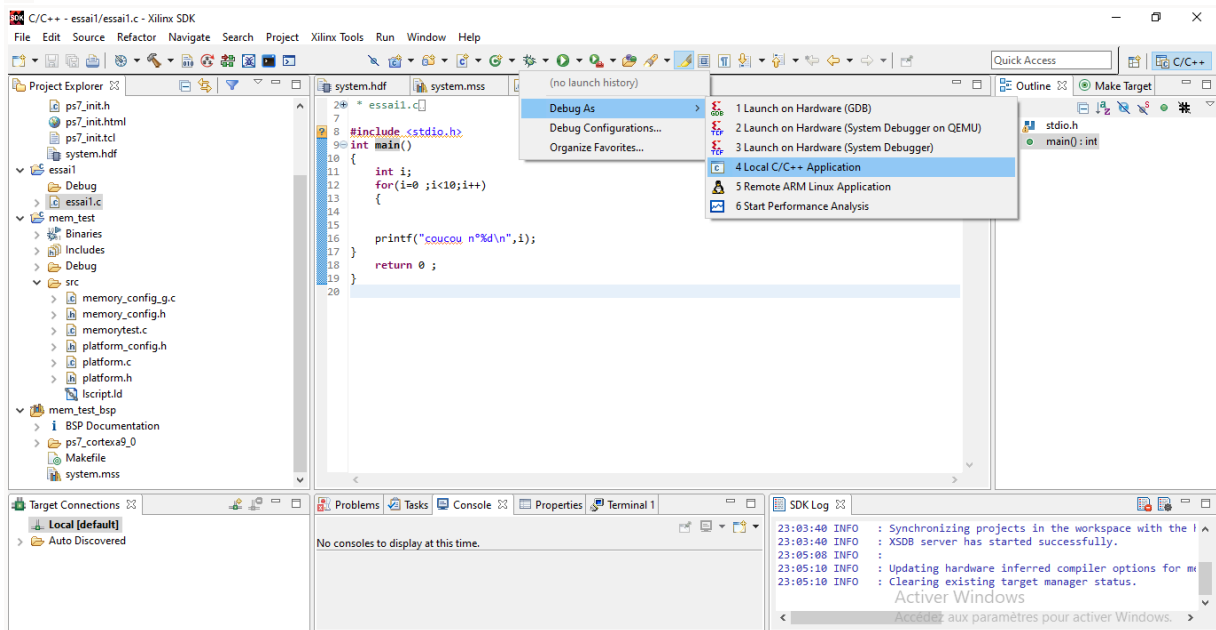
```

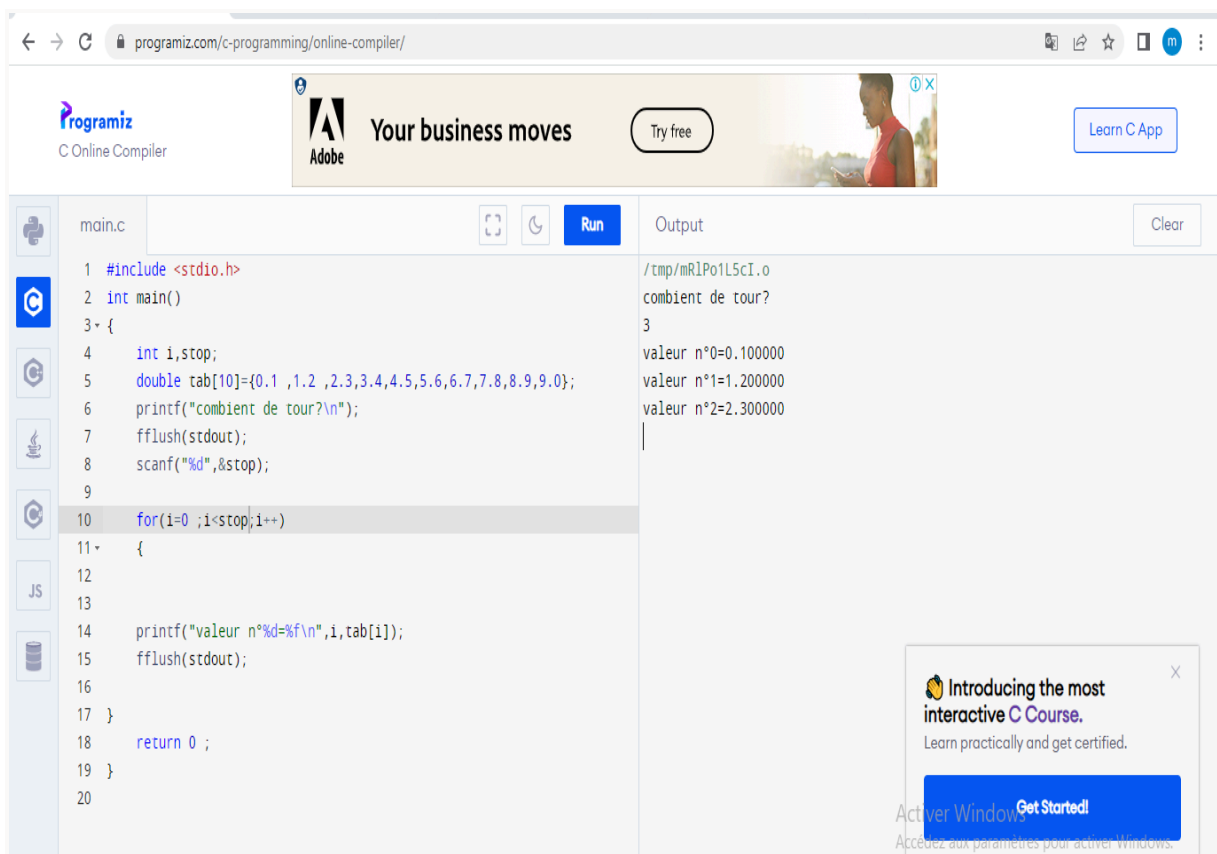
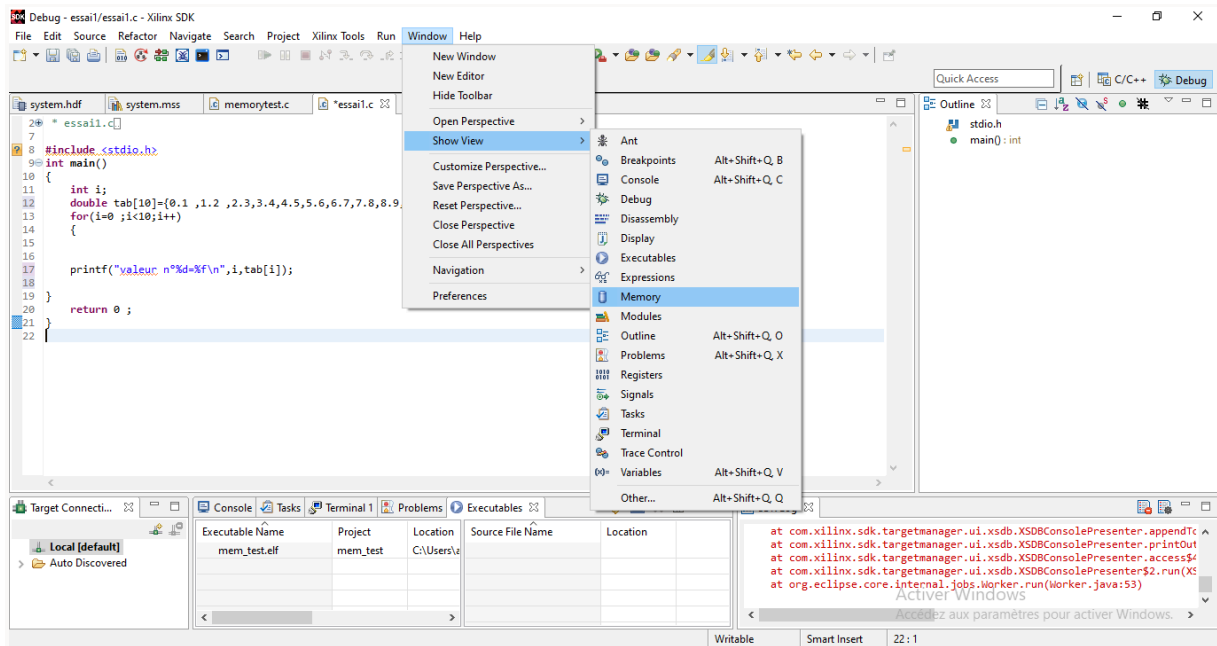
/tmp/mRlPo1L5cI.o
valeur n°0=0.100000
valeur n°1=1.200000
valeur n°2=2.300000
valeur n°3=3.400000
valeur n°4=4.500000
valeur n°5=5.600000
valeur n°6=6.700000
valeur n°7=7.800000
valeur n°8=8.900000
valeur n°9=9.000000
  
```



Relançons le Debug Mode

41





Travail demandé :

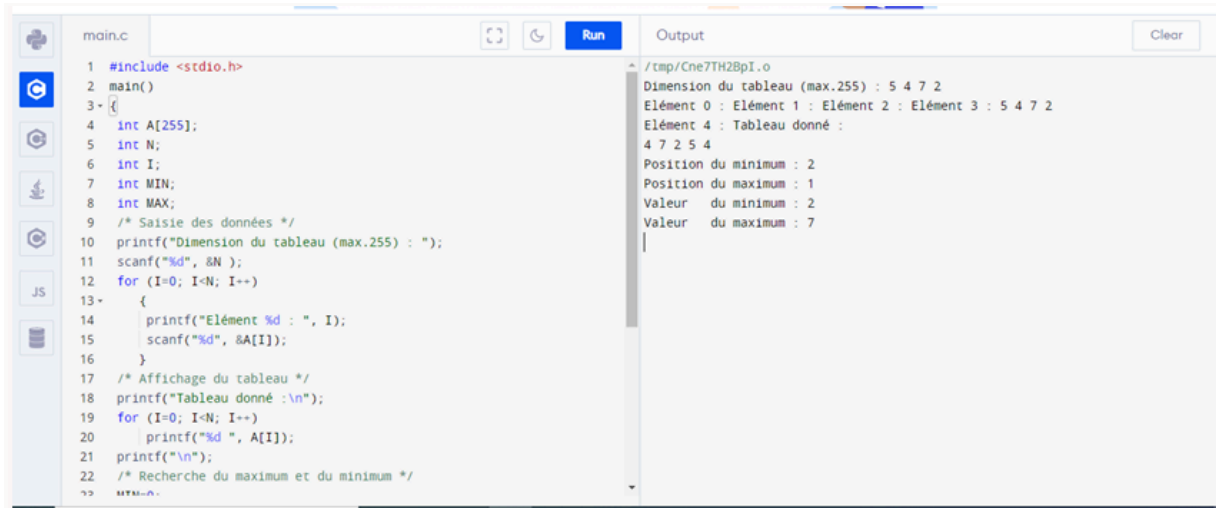
Exercice 1 :

```
main.c
1
2 #include <stdio.h>
3 int main() {
4     int T[255];
5     int N;
6     int I,J;
7     int AIDE;
8     printf("Dimension du tableau (max.255) : ");
9     scanf("%d", &N );
10    for (I=0; I<N; I++)
11    {
12        printf("Elément %d : ", I);
13        scanf("%d", &T[I]);
14    }
15    printf("Tableau donné : \n");
16    for (I=0; I<N; I++)
17        printf("%d ", T[I]);
18    printf("\n");
19    for (I=0, J=N-1 ; I<J ; I++,J--)
20    {
21        AIDE = T[I];
22        T[I] = T[J];
23        T[J] = AIDE;
24    }
25    printf("Tableau résultat : \n");
26    for (I=0; I<N; I++)
27        printf("%d ", T[I]);
28    printf("\n");
29 }
```

Output

```
/tmp/jVF1zBa4Hd.o
Dimension du tableau (max.255) : 7 20 100 1 5 6
Elément 0 : Elément 1 : Elément 2 : Elément 3 : Elément 4 : Elément 5 : 1 5
6 7 20 100
Elément 6 : Tableau donné :
20 100 1 5 6 1 5
Tableau résultat :
5 1 6 5 1 100 20
```

Exercice 2 :



```
1 #include <stdio.h>
2 main()
3 {
4     int A[255];
5     int N;
6     int I;
7     int MIN;
8     int MAX;
9     /* Saisie des données */
10    printf("Dimension du tableau (max.255) : ");
11    scanf("%d", &N );
12    for (I=0; I<N; I++)
13    {
14        printf("Elément %d : ", I);
15        scanf("%d", &A[I]);
16    }
17    /* Affichage du tableau */
18    printf("Tableau donné :\n");
19    for (I=0; I<N; I++)
20        printf("%d ", A[I]);
21    printf("\n");
22    /* Recherche du maximum et du minimum */
23    MIN=A[0];
24    MAX=A[0];
```

Output

```
/tmp/Cne7TH2BpI.o
Dimension du tableau (max.255) : 5 4 7 2
Elément 0 : Elément 1 : Elément 2 : Elément 3 : 5 4 7 2
Elément 4 : Tableau donné :
4 7 2 5 4
Position du minimum : 2
Position du maximum : 1
Valeur du minimum : 2
Valeur du maximum : 7
```