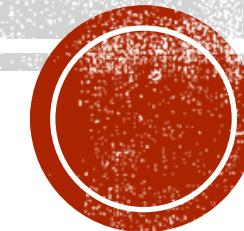
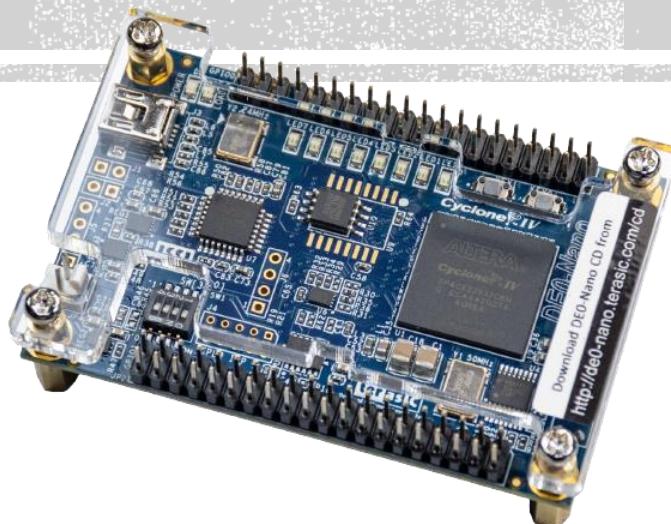


# PROGRAMMATION VHDL



By Damian Sal y Rosas

# OBJECTIF:

- Faire un premier programme VHDL sur Quartus Lite 18.1
- Prends en main le cible DE0 Nano

■ Outils: **Quartus Lite**



■ Cible: DE0 Nano



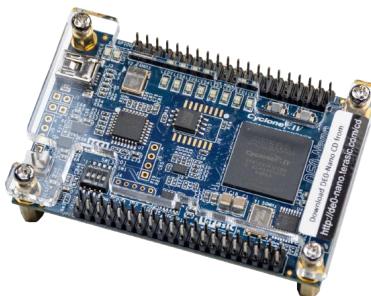
# PREMIER PROGRAMME VHDL



Visualiser le résultat sur une lumière LED



Programmation sur  
**Quartus Lite 18.1**



Validation sur la cible **DE0 Nano**

LET'S START ...



## Lancer le programme



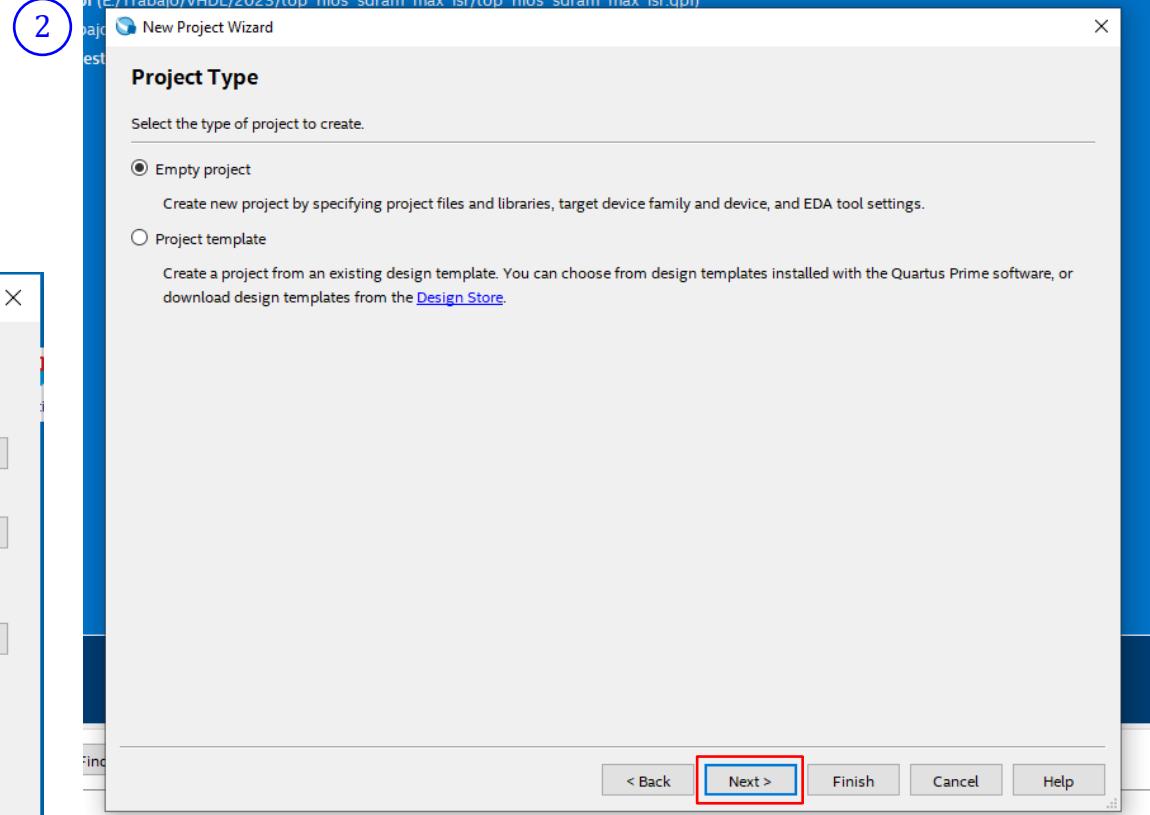
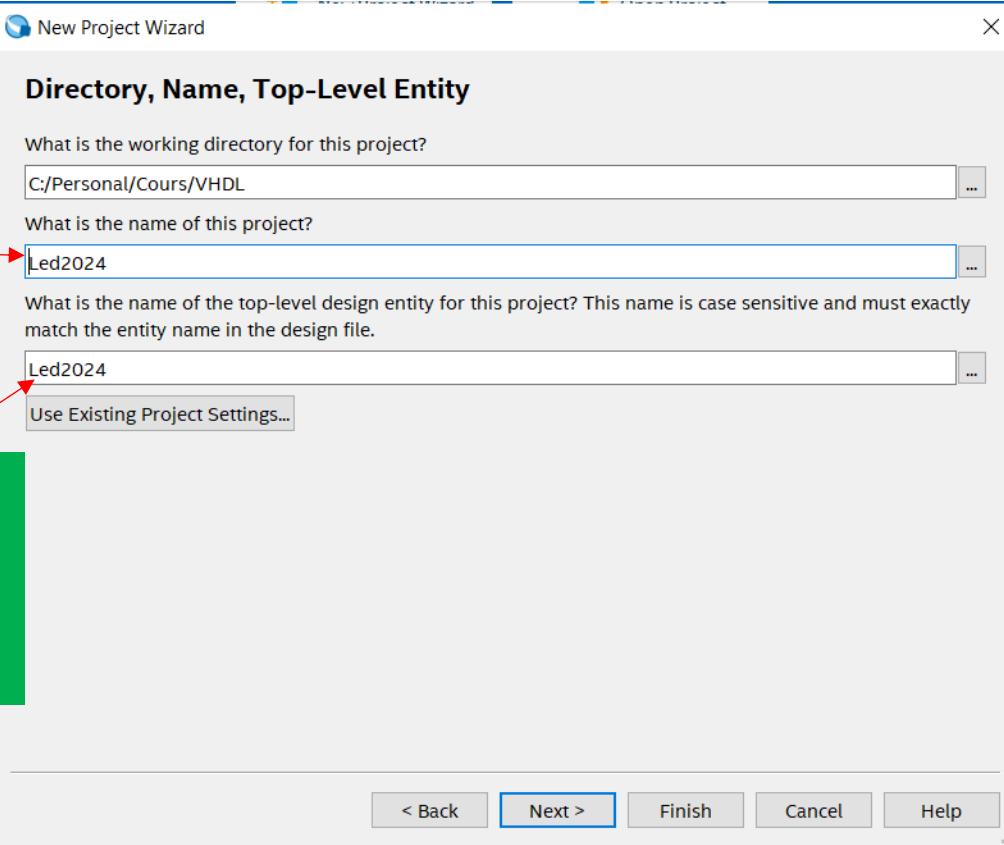
First: Il faut créer un projet  
→ New Project Wizard

The screenshot shows the Quartus Prime Standard Edition software interface. The window title is "Quartus Prime Standard Edition". The menu bar includes File, Edit, View, Project, Assignments, Processing, Tools, Window, and Help. The toolbar contains various icons for file operations like Open, Save, and Print. On the left, there's a "Project Navigator" pane showing a "Compilation Hierarchy" with several recent projects listed. Below it is a "Tasks" pane with a "Task" section containing items like "Compile Design", "Analysis & Synthesis", and "Fitter (Place & Route)". At the bottom, there's a "Messages" pane with tabs for "System" and "Processing". The main central area is titled "Home" and features a "Recent Projects" section with four project entries. Below this is a large button labeled "New Project Wizard" with a small icon of three squares and a plus sign, which is highlighted with a red rectangular box and an arrow pointing to it. To the right of this button is another button labeled "Open Project" with a similar icon. Further down are five smaller buttons: "Documentation" (book icon), "Training" (person icon), "Support" (person icon), "What's New" (globe icon), and "Notifications" (speech bubble icon). On the far right, there's an "IP Catalog" panel with sections for "Installed IP", "Project Directory" (which is currently empty), "Library" (with categories like Basic Functions, DSP, Interface Protocols, etc.), and a search bar for "Search altera.com". The Intel logo is visible at the bottom right of the main window.

Etape 1: Créer un fichier de travail

Etape 2: New Project Wizard → ... sélectionner le directoire → Next

1

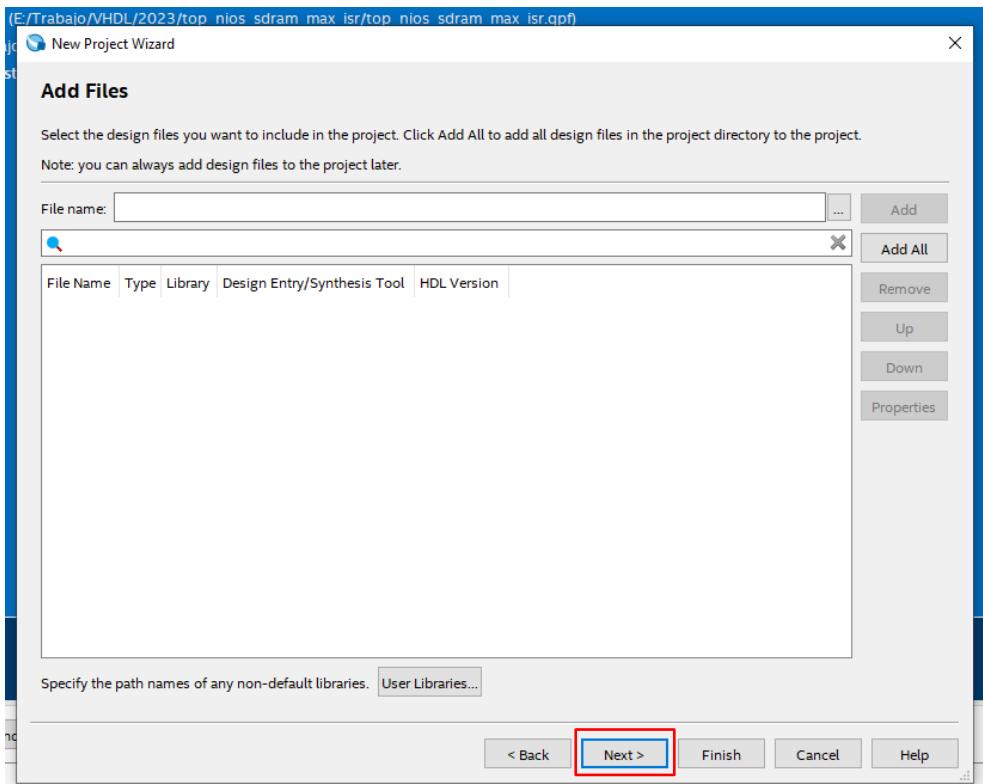


Sélectionner empty project → next

Considérations:

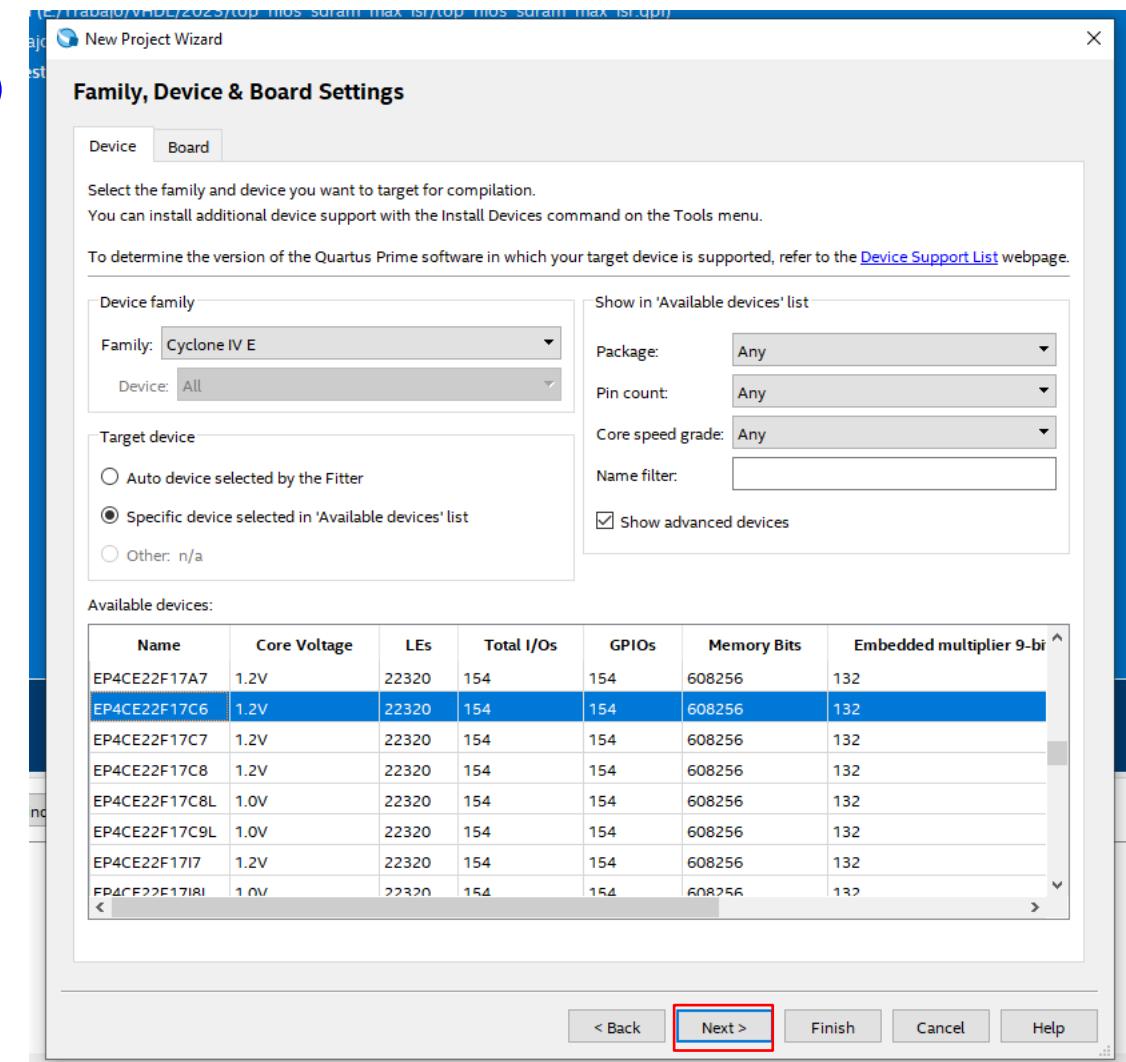
- **Short Name (le lien doit être court)**
- **Pas de espace dans le nom du projet**

3

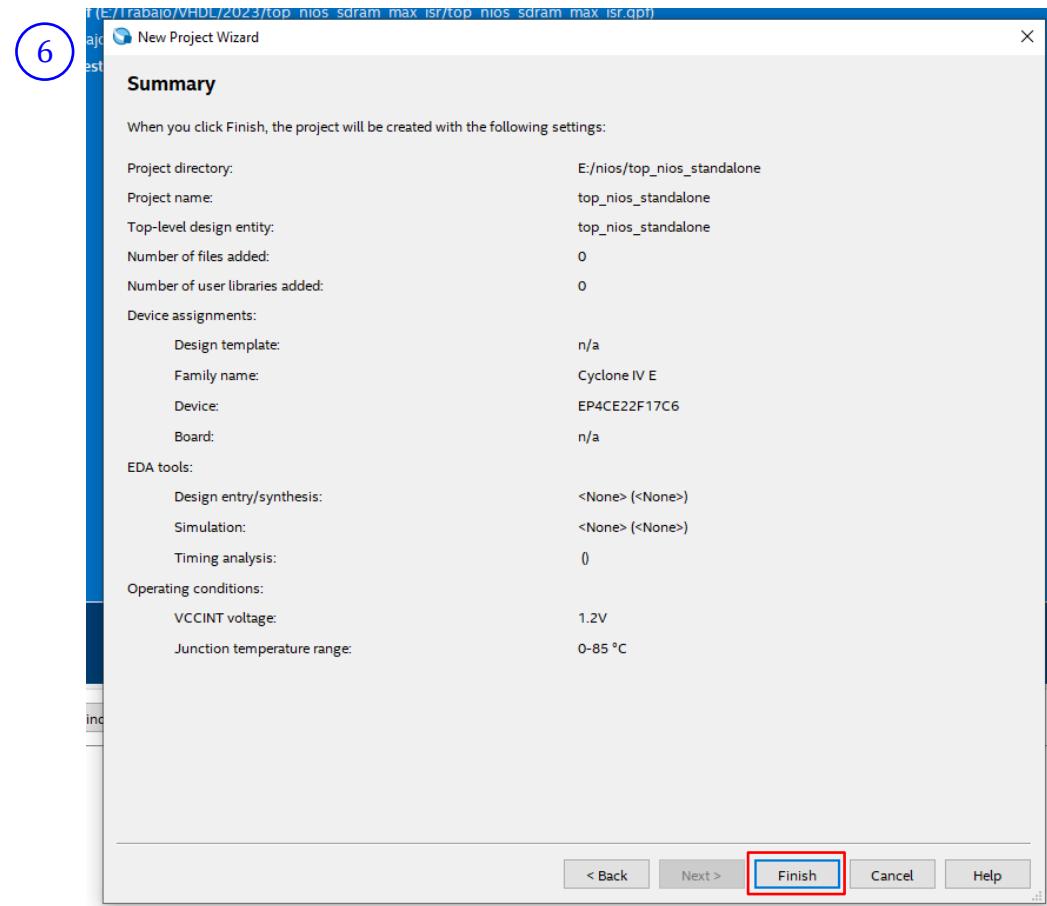
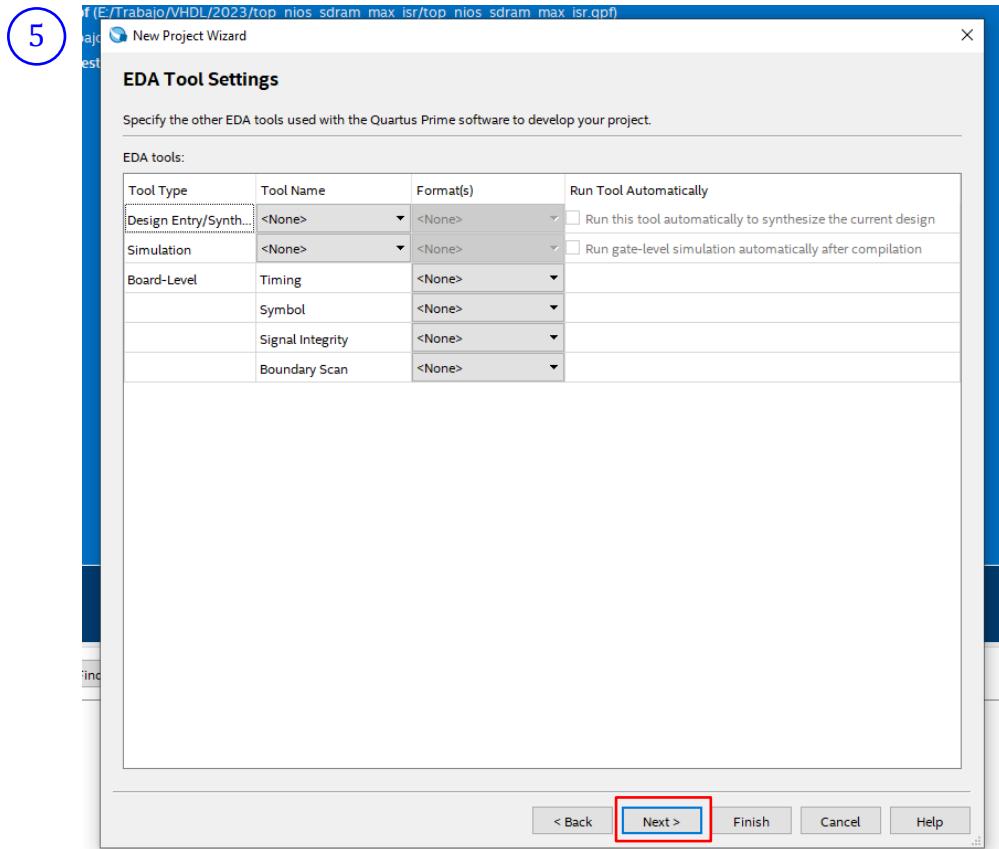


**Etape 3:**  
pas de fichiers à ajouter → Next

4



**Etape 4: sélectionner FPGA ... Altera  
Cyclone IV EP4CE22F17C6 → Next**



Etape 5: next → finish

\* Sélectionner en mode simulation Altera ModelSim – VHDL s'il faut simuler

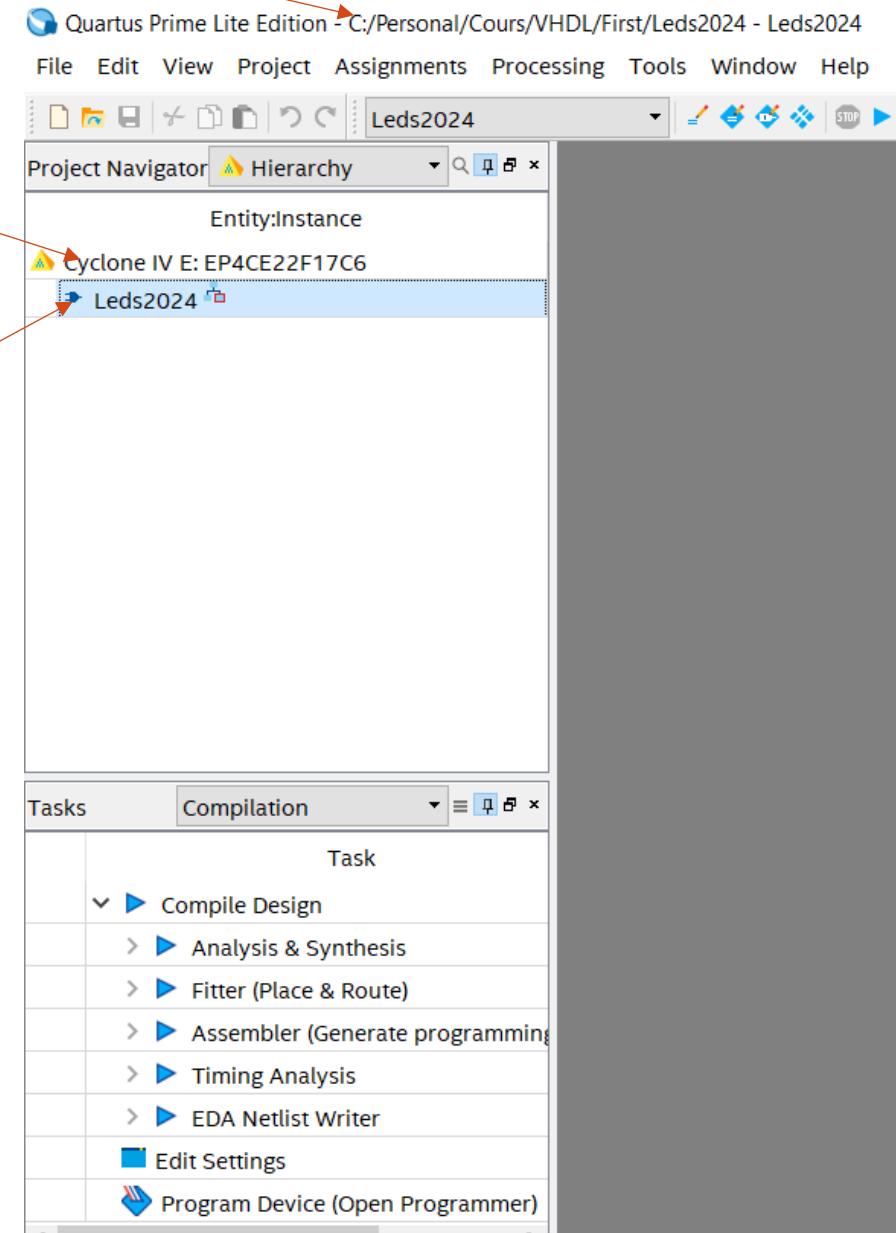
Directoire

# CONSOLE DE TRAVAIL

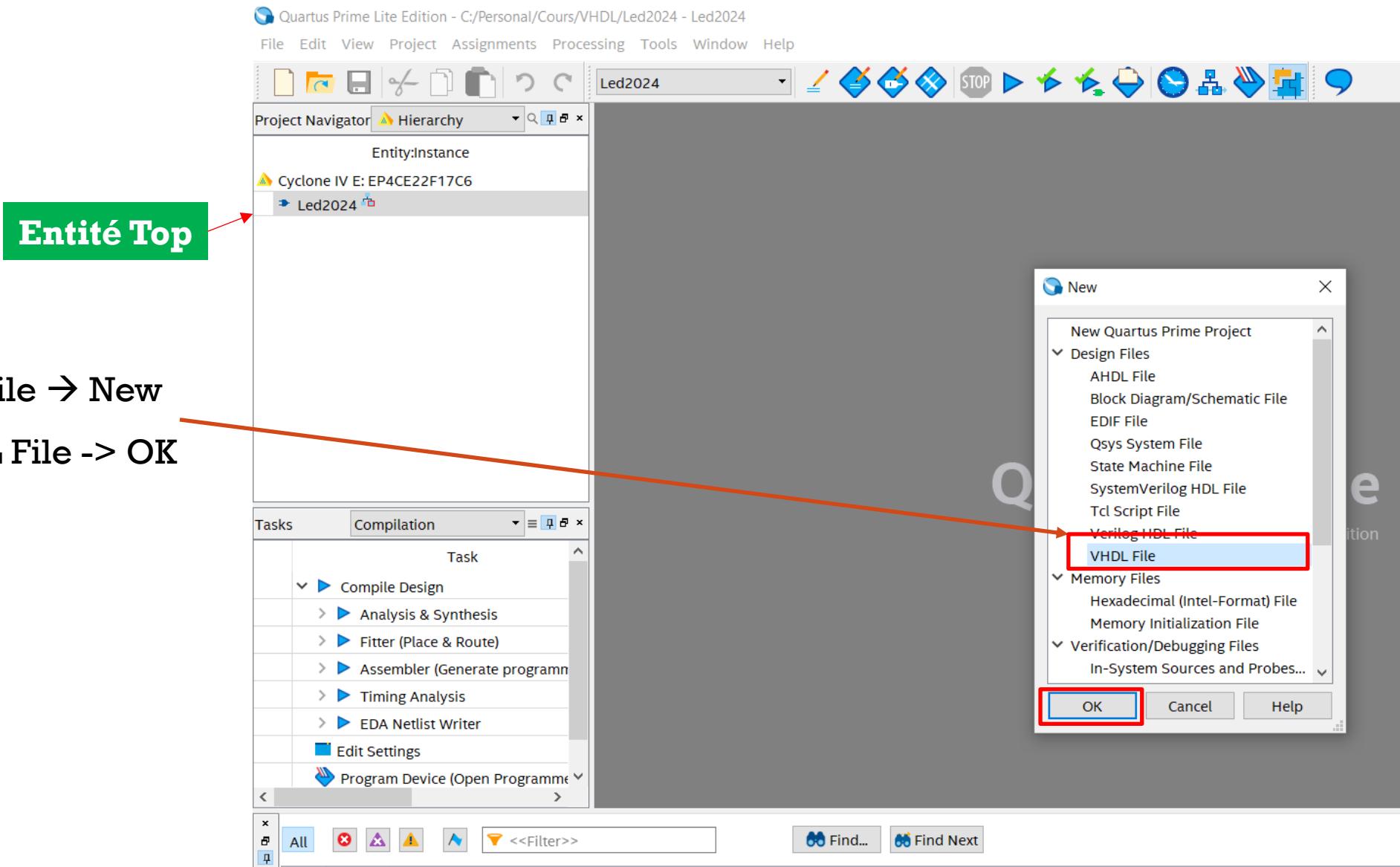
Code du FPGA

Entité Top

Il faut ajouter le  
fichier VHDL  
pour le design  
du circuit



# Ajouter un fichier VHDL sur la console de travail



# Ajouter un fichier VHDL sur la console de travail

Apparaît un nom par défaut:

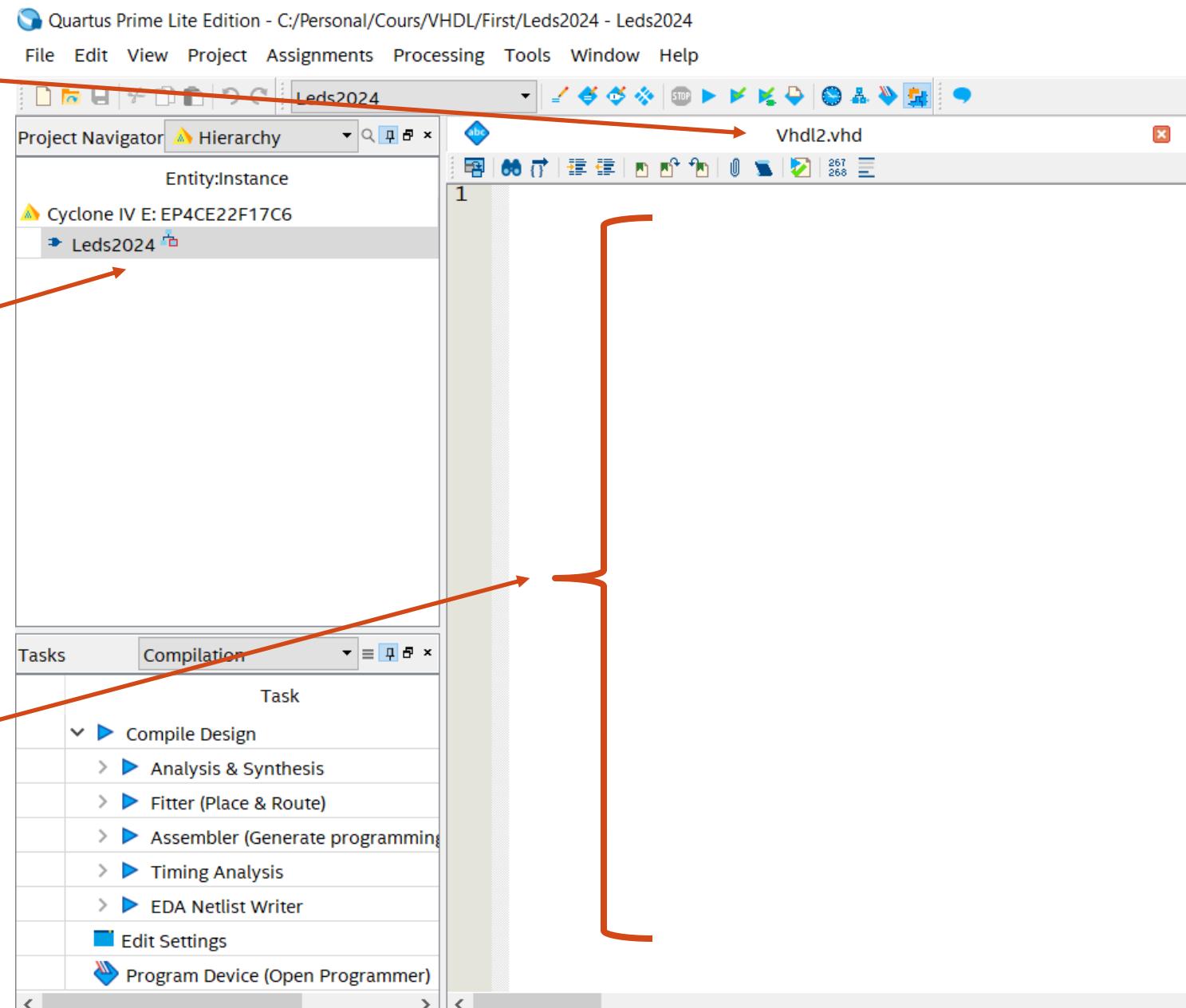
1. Il faut modifier le nom correspondant au nom de l'ENTITE

Entité Top

Après:

Il faut ajouter le code VHDL:

- library
- Entity
- architecture



# Ajouter un fichier VHDL sur la console de travail

## Etape 2:

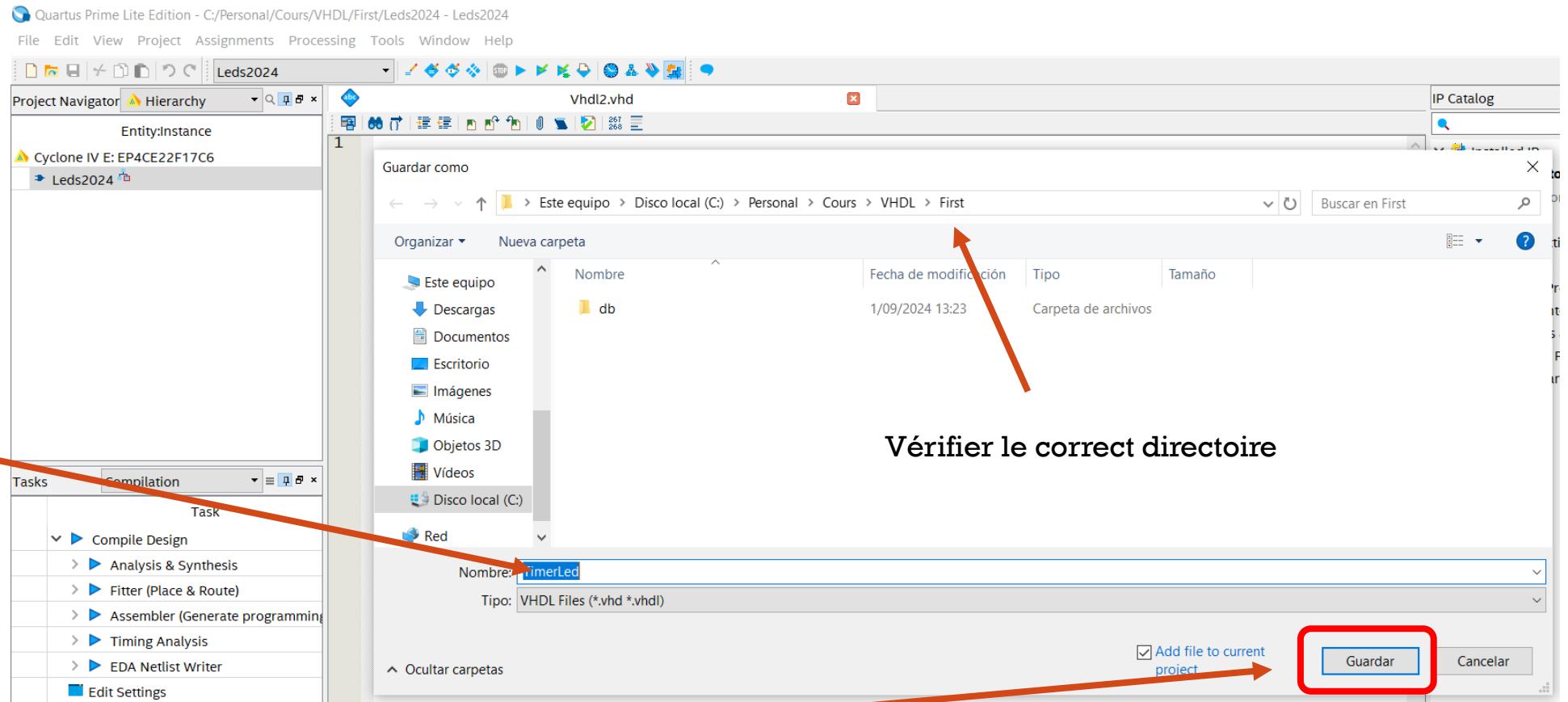
Garder le fichier avec le nom de l'entité

Exemple: TimerLed

File->SaveAs..

Indiquer le nom de l'entité

TimerLed

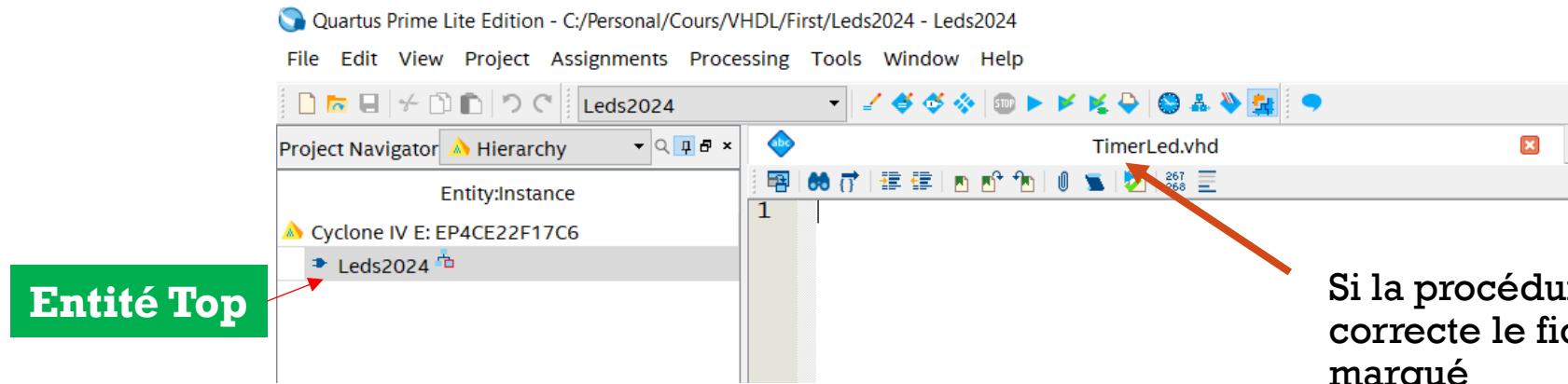


Finalement:

→ Save

**Attention: l'entité s'appelle « TimerLed » et doit être désignée comme entité TOP**

# Ajouter un fichier VHDL sur la console de travail



## Etape 3:

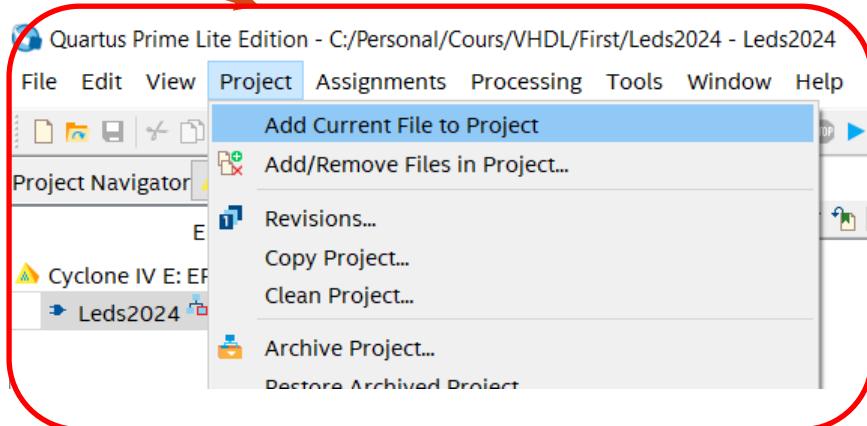
On doit ajouter le fichier **TimerLed.vhd** au projet:

→ Projet

→ Add current File to Projet

Si la procédure est correcte le fichier est marqué  
**TimerLed.vhd**

**Attention: l'entité s'appelle « TimerLed » et doit être désignée comme entité TOP**



# Ajouter le code « VHDL »

Quartus Prime Lite Edition - C:/Personal/Cours/VHDL/First/Leds2024 - Leds2024

File Edit View Project Assignments Processing Tools Window Help

Project Navigator Hierarchy

Entity:Instance

Cyclone IV E: EP4CE22F17C6

Leds2024

TimerLed.vhd

1 -- declaration des libraries  
2  
3  
4 library ieee;  
5 use ieee.std\_logic\_1164.all;  
6 use ieee.std\_logic\_unsigned.all;  
7 use ieee.std\_logic\_arith.all;  
8  
9 entity TimerLed is  
10  
11 -- generic (declaration de constants); --  
12 -- port(declaration de ports d'entrée et sortie);  
13 end TimerLed;  
14  
15 architecture Beh of TimerLed is  
16  
17 -- declaration de type de données, signals, components  
18  
19 begin  
20  
21 -- declaration du "comportement" du circuit --  
22  
23 end Beh;

Library

Entity

Architecture

Tasks Compilation

Task

> Compile Design

Edit Settings

Program Device (Open Programmer)

## Etape 4:

Déclarer le code VHDL

- Library
- Entity
- Architecture

**Erreurs 2: changer  
l'entité « Leds2024 » à  
l'entité « TimerLed »  
comme entité TOP**

## Etape 5:

Compléter:

- Library
- Entity
- Architecture

## Ajouter le code « VHDL »

```
-- declaration des librairies
library ieee;
use ieee.std_logic_1164.all;
use ieee.std_logic_unsigned.all;
use ieee.std_logic_arith.all;

entity TimerLed is
    generic(max: natural:=2500000);
    port(clk:in std_logic;
          rst:in std_logic;
          led: out std_logic);      -- signal led
end TimerLed;

architecture Beh of TimerLed is
begin
    process(clk,rst)
        variable temp:integer range 0 to max;
        variable tout:std_logic;
    begin
        if (rst='0') then
            led<='0';
            temp:=0;
            tout:='0';
        elsif (rising_edge(clk)) then
            temp:=temp+1;
            if (temp=max) then
                temp:=0;
                tout:=not tout;
            end if;
            led<=tout;
        end if;
    end process;
end Beh;
```

## Etape 6:

1. Clik dans:

→ Compile design

2. Résultat X: « erreurs dans le fichier»

Erreurs 1: assignation d'une variable

Erreurs 2: Entité top incorrecte

# Correction du code « VHDL »

**Etape 7:**

**Erreur 1:**

Correction du code VHDL

**temp:=temp+1;**

The screenshot shows the Quartus Prime interface. In the top right, there is a code editor window displaying VHDL code. The code includes a process block with an if-elsif structure. The elsif condition is highlighted in blue. Below the code editor is a messages window titled 'Messages' which lists several compilation errors and warnings.

```
18 begin
19   process(clk,rst)
20     variable temp:integer range 0 to max;
21     variable tout:std_logic;
22 begin
23   if (rst='0') then
24     led<='0';
25     temp:=0;
26     tout:='0';
27   elsif (rising_edge(clk)) then
28     temp:=temp+1;
29     if (temp=max) then
30       temp:=0;
31     tout:=not tout;
32 end begin
```

Messages window content:

- 18236 Number of processors has not been specified which may cause overloading on shared machines.
- 20030 Parallel compilation is enabled and will use 4 of the 4 processors detected
- 12019 Can't analyze file -- file Leds2024.vhd is missing
- 10500 VHDL syntax error at TimerLed.vhd(29) near text "="; expecting "(", or "'", or "."
- 12021 Found 0 design units, including 0 entities, in source file timerled.vhd
- 293001 Quartus Prime Analysis & Synthesis was unsuccessful. 1 error, 2 warnings
- 293001 Quartus Prime Full Compilation was unsuccessful. 3 errors, 2 warnings

**Etape 7:**

**Erreur 2:**

Indiquer l'entité top:TimerLed

The screenshot shows the Quartus Prime interface. In the top right, there is a code editor window displaying VHDL code. The code is identical to the one in the previous screenshot. Below the code editor is a messages window titled 'Messages' which lists several compilation errors and warnings, including one specific to the top-level entity.

```
18 begin
19   process(clk,rst)
20     variable temp:integer range 0 to max;
21     variable tout:std_logic;
22 begin
23   if (rst='0') then
24     led<='0';
25     temp:=0;
26     tout:='0';
27   elsif (rising_edge(clk)) then
28     temp:=temp+1;
29     if (temp=max) then
30       temp:=0;
31     tout:=not tout;
32 end begin
```

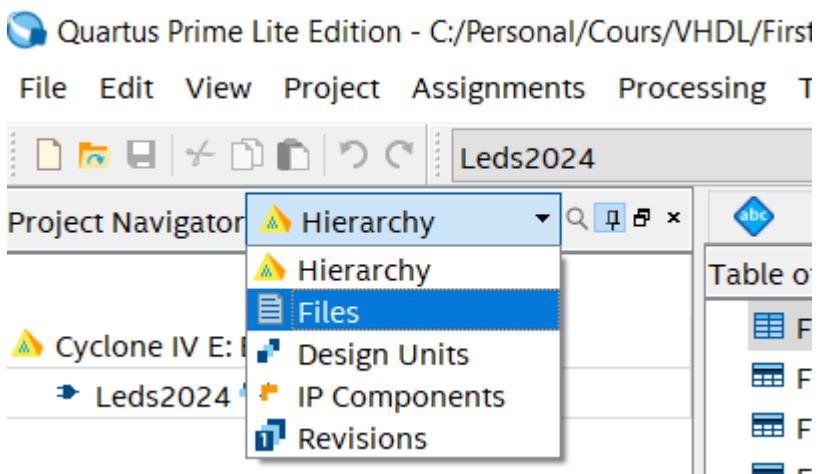
Messages window content:

- 18236 Number of processors has not been specified which may cause overloading on shared machines.
- 20030 Parallel compilation is enabled and will use 4 of the 4 processors detected
- 12019 Can't analyze file -- file Leds2024.vhd is missing
- 12021 Found 2 design units, including 1 entities, in source file timerled.vhd
  - 12007 Top-level design entity "Leds2024" is undefined
- 293001 Quartus Prime Analysis & Synthesis was unsuccessful. 1 error, 2 warnings
- 293001 Quartus Prime Full Compilation was unsuccessful. 3 errors, 2 warnings

# Correction du code « VHDL »

## Etape 7: Correction de l'Erreur 2:

1. Click Hierarchy → Files



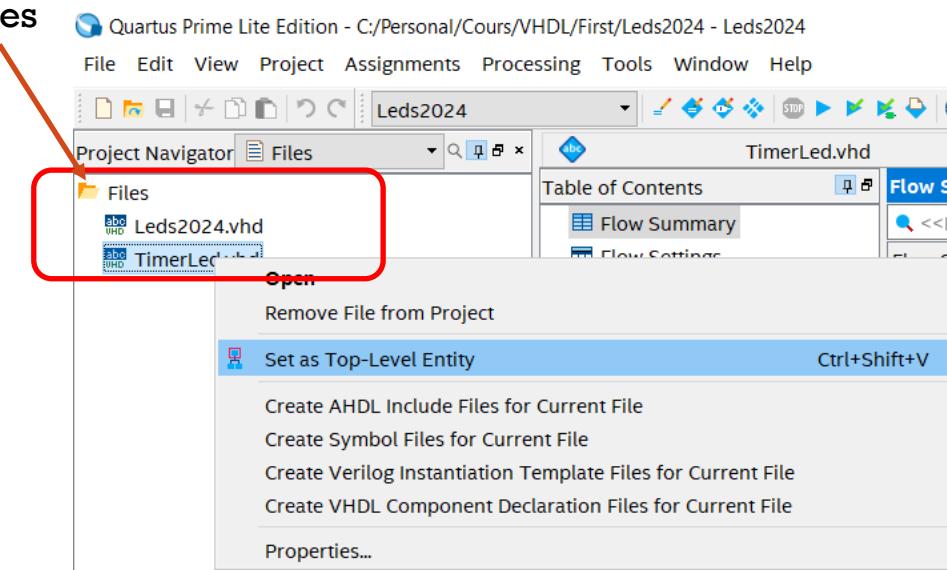
## Etape 7: Correction de l'Erreur 2:

La liste des entités VHDL sont visualisées

2. Click droite in Entité TimerLet

Selectionner:

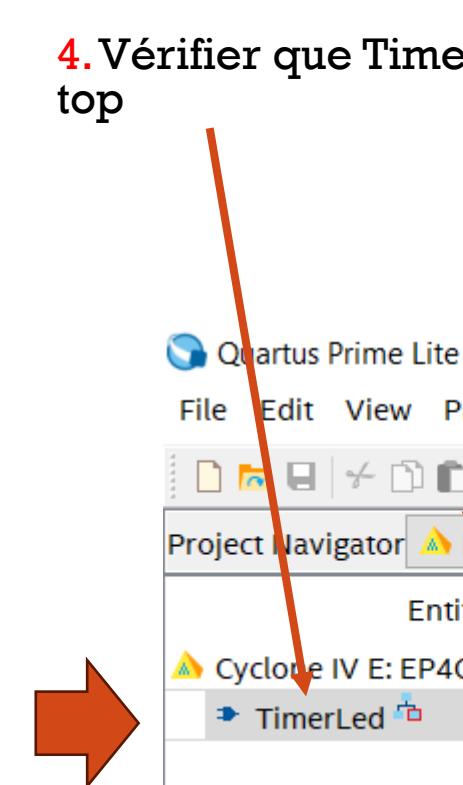
→ Set as Top-Level Entity



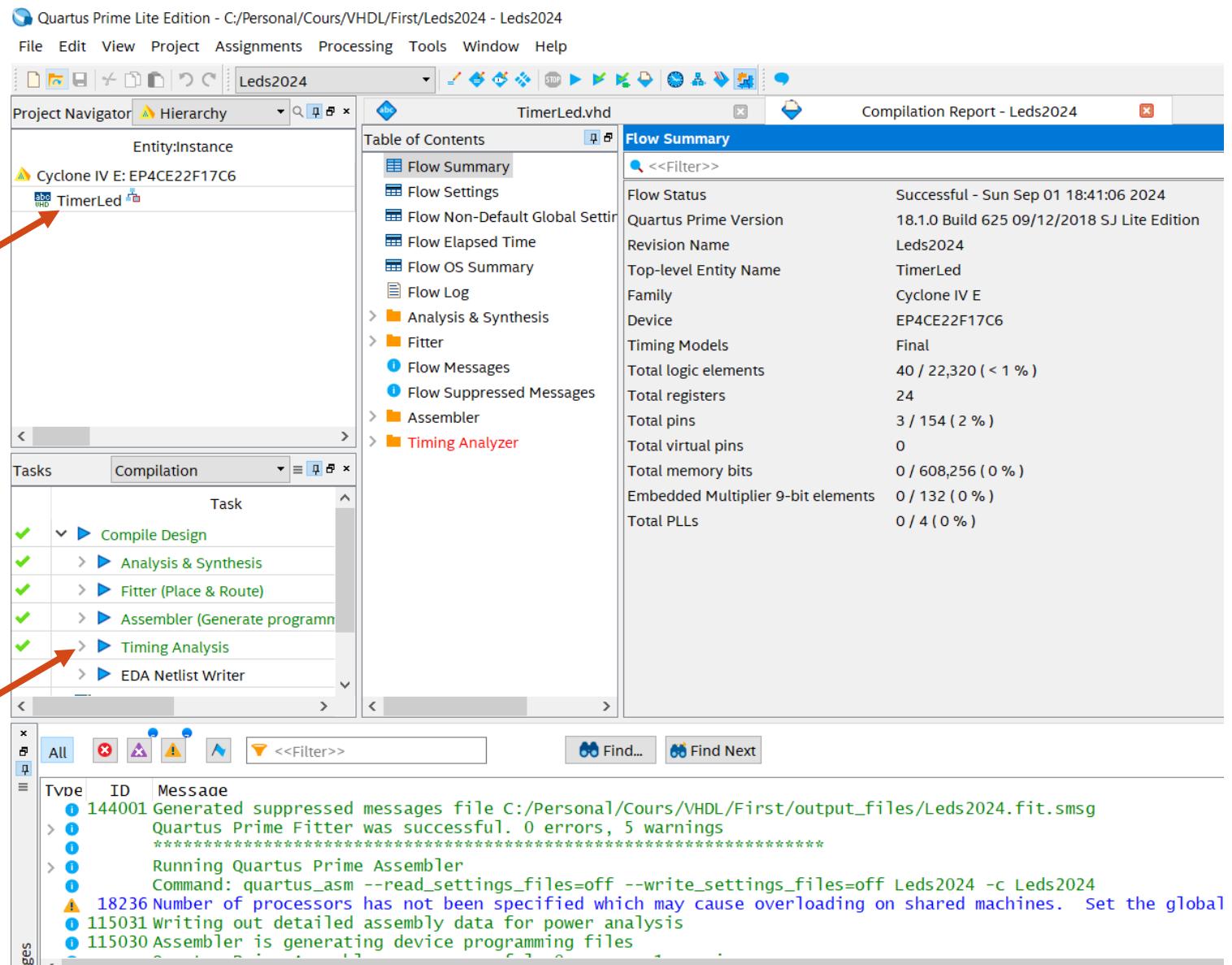
## Etape 7: Correction de l'Erreur 2:

3. Sélectionner Hierarchy

4. Vérifier que TimerLed est l'entité top



# Compilation réussie du programme VHDL



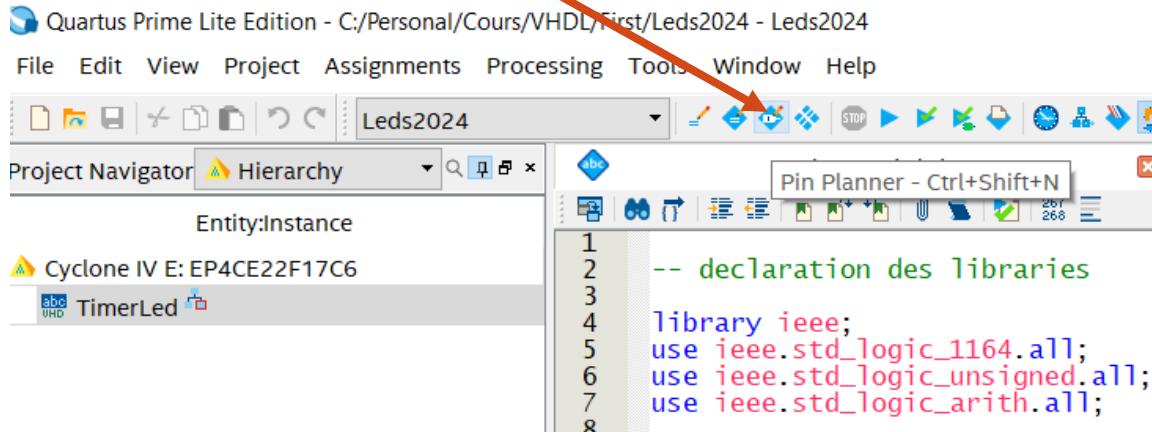
Entité top:  
**TimerLed.vhd**

Process de compilation  
réussi

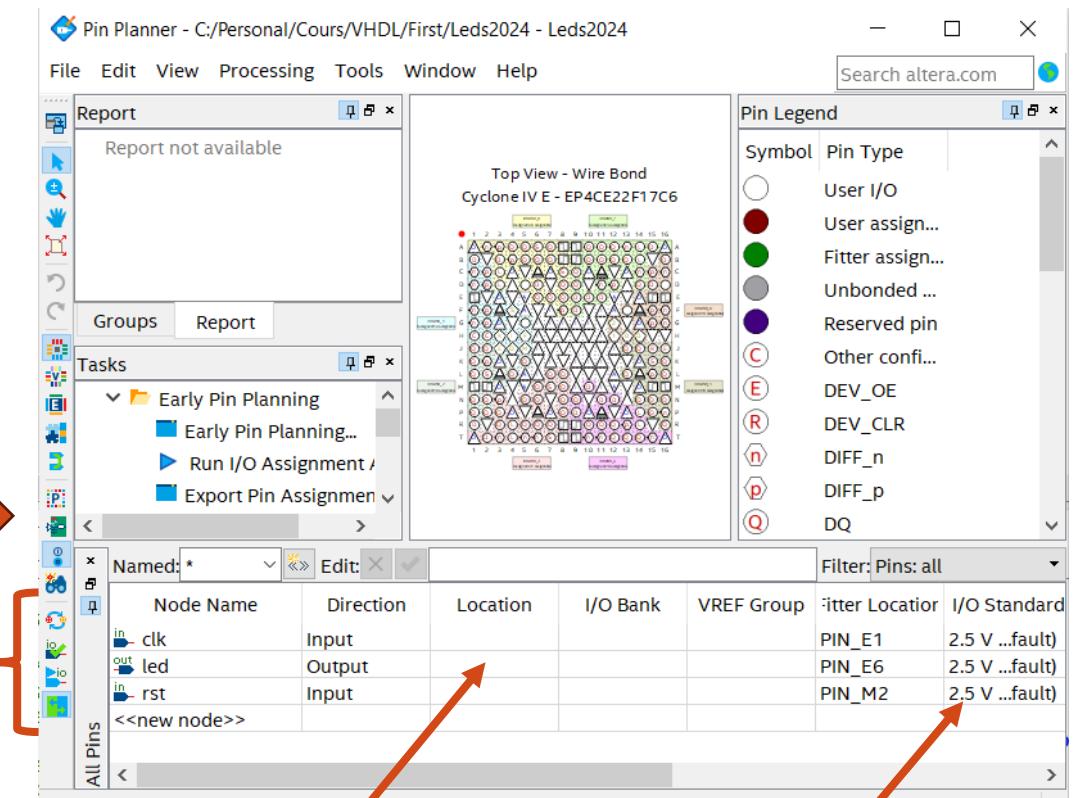
# Assignation de pins

- Une fois que la compilation est réussie, il faut assigner les entrées et sorties du FPGA (pins)
- L'assignation de pins est faite avec l'option **Pin Planner**

**Click → PinPlanner**



Entrées et sorties de l'entité



Indiquer le port du FPGA  
(voir datasheet DE0 Nano)

Indiquer la tension du port  
(voir datasheet DE0 Nano)

# Attribution de pins

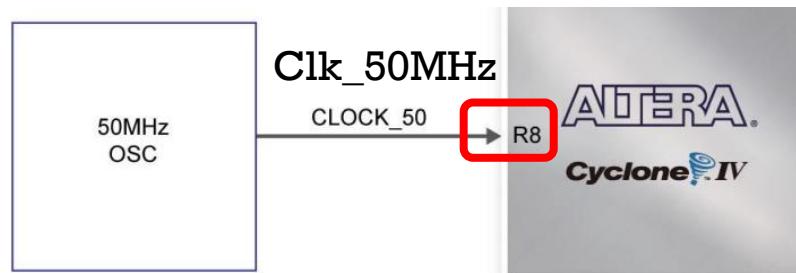
## Entrées et sorties du FPGA

Table 3-1 Pin Assignments for Push-buttons

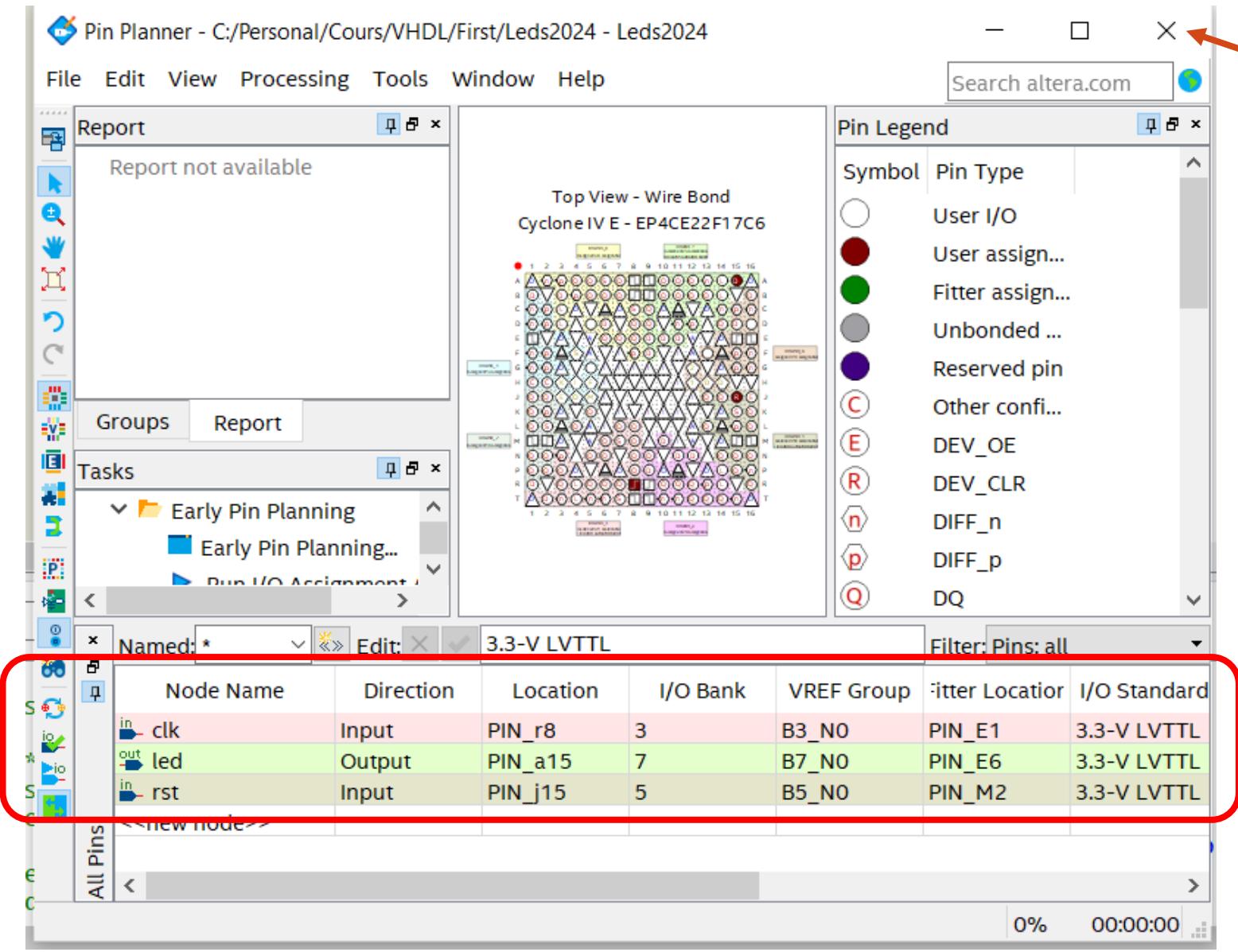
	Signal Name	FPGA Pin No.	Description	I/O Standard
rst	KEY[0]	PIN_J15	Push-button[0]	3.3V
	KEY[1]	PIN_E1	Push-button[1]	3.3V

Table 3-2 Pin Assignments for LEDs

	Signal Name	FPGA Pin No.	Description	I/O Standard
led	LED[0]	PIN_A15	LED Green[0]	3.3V
	LED[1]	PIN_A13	LED Green[1]	3.3V
	LED[2]	PIN_B13	LED Green[2]	3.3V
	LED[3]	PIN_A11	LED Green[3]	3.3V
	LED[4]	PIN_D1	LED Green[4]	3.3V
	LED[5]	PIN_F3	LED Green[5]	3.3V
	LED[6]	PIN_B1	LED Green[6]	3.3V
	LED[7]	PIN_L3	LED Green[7]	3.3V

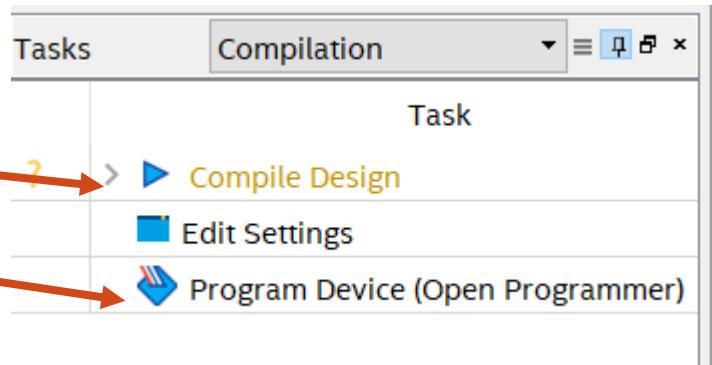


# Assignment de pins

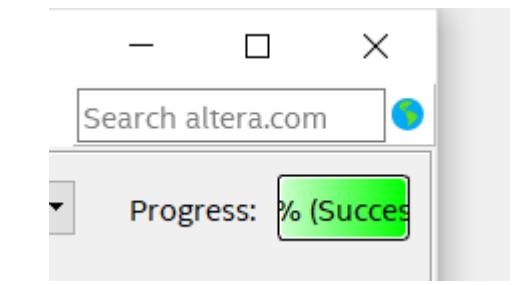
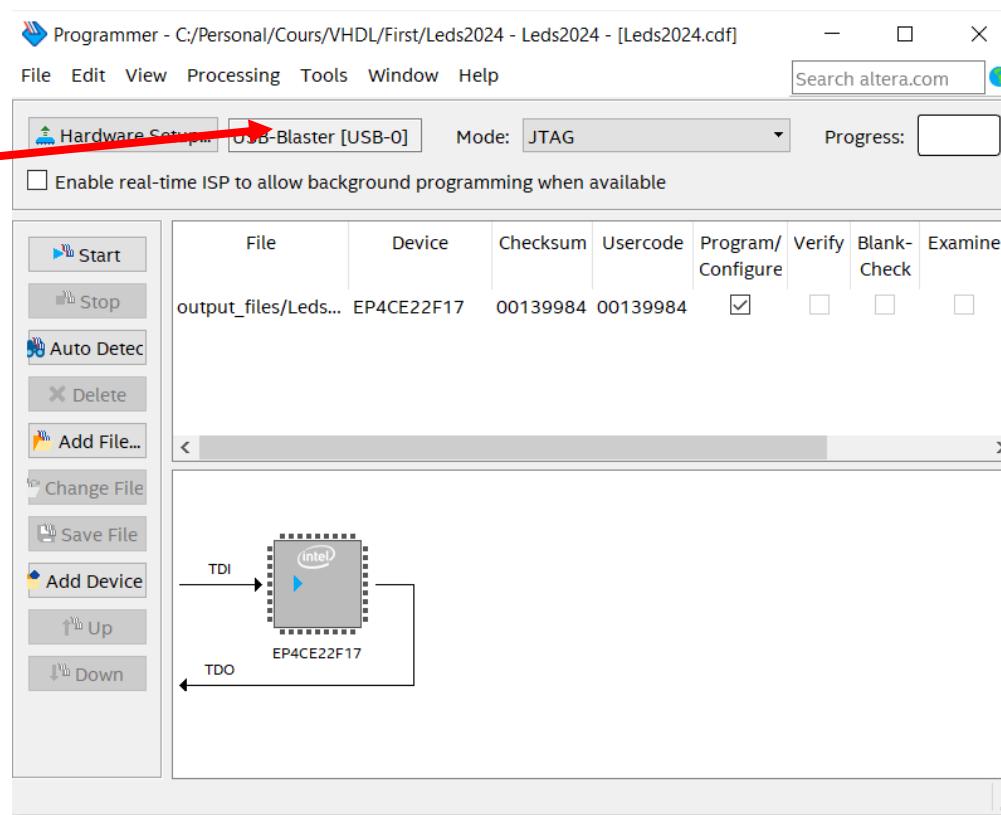


# Validation sur la carte

1. Compiler une 2eme fois (click)
2. Programmer la carte



3. Sélectionner USB-Blaster
4. Envoi du fichier .sof vers la carte



5. Programmation ok

