

Quickstart - Quartus II & ModelSIM

1) Télécharger et installer Quartus : <https://fpgasoftware.intel.com/?edition=lite>

Home > Downloads > Quartus Prime Lite Edition




Download Center for FPGAs

[Design Software](#)
[Embedded Software](#)
[Archives](#)
[Licensing](#)
[Programming Software](#)
[Drivers](#)
[Board System Design](#)
[Board Layout and Test](#)
[Legacy Software](#)

Quartus Prime Lite Edition

Release date: November, 2020
Latest Release: v20.1.1

Select edition:
Select release:

Operating System   Windows  Linux

✓ The Quartus Prime Lite Edition Design Software, Version 20.1.1 includes functional and security updates. Users should keep their software up-to-date and follow the [technical recommendations](#) to help improve security.

✓ The Quartus Prime Lite Edition Design Software, Version 20.1.1 is subject to removal from the web when support for all devices in this release are available in a newer version, or all devices supported by this version are obsolete. If you would like to receive customer notifications by e-mail, please subscribe to our [subscribe to our customer notification mailing list](#).

✓ The Quartus Prime Lite Edition Design Software, Version 20.1.1 supports the following device families: Arria II, Cyclone 10 LP, Cyclone IV, Cyclone V, MAX II, MAX V, and MAX 10 FPGA. [More](#)

[Combined Files](#) [Individual Files](#) [Additional Software](#)

Download and install instructions: [More](#)

[Read Intel FPGA Software v20.1.1 Installation FAQ](#)

[Quick Start Guide](#)

2) Installer le driver pour USB-Blaster :

https://www.terasic.com.tw/wiki/Altera_USB_Blaster_Driver_Installation_Instructions

3) Créer un nouveau projet

1. Choisir **New Project Wizard**
2. Choisir un dossier et un nom de projet
3. Choisir « Empty project » et cliquer sur Next
4. Dans le dialogue « Add files », Cliquer sur Next encore une fois
5. Dans le dialogue « Family, Device & board settings », choisir :
Family : **MAX 10 (DA/DF/DC/SA/SF/SC)**
Devices : **MAX 10 DA**
Available devices : **10M50DAF484C6GES**
6. Cliquer sur Next
7. Dans le dialogue EDA tool settings, cliquer sur Finish

4) Créer un nouveau fichier VHDL

- File -> New ... -> VHDL File
- Copier le contenu suivant

```
LIBRARY ieee;
USE ieee.std_logic_1164.all;

ENTITY quickstart IS
    PORT (
        SW : IN STD_LOGIC_VECTOR(9 DOWNT0 0);
        LEDR : OUT STD_LOGIC_VECTOR(9 DOWNT0 0)
    );
END quickstart;

ARCHITECTURE Behavior OF quickstart IS
BEGIN
    LEDR <= SW;
END Behavior;
```

- Sauvegarder le fichier (sous quickstart.vhdl)
- Compiler settings
 - a. Choisir « Hierarchy » dans Project Navigator,
 - b. Bouton de droite sur MAX10 :10M50DAF484C6GE5
 - c. Compiler Settings -> VHDL input -> VHDL 2008
- Définir comme Top-Level Entity.
 - a. Choisir Files dans Project Navigator,
 - b. Sélectionner le fichier quickstart.vhd
 - c. Bouton de droit, choisir « Set as Top-Level Entity »

5) Importer le fichier qsf pour assigner les pins

1. Télécharger le fichier .qsf correspondant à la plaquette DE10-lite fournis dans les <https://software.intel.com/content/www/us/en/develop/articles/fpga-academic-boards.html> (le fichier est aussi sur moodle)
2. Assignments -> import assignments...
3. Choisir le fichier .qsf

6) Compiler le programme

1. Processing -> Start compilation (ou ctrl-L)b

7) Programmer la plaquette

1. Tools->programmer
2. Hardware Setups... -> Currently selected hardware
 - a. choisir USB-Blaster [USB-0]
3. Start

Configurer ModelSIM

8) Modifiez votre code (remplacez votre code par celui-ci) :

```
library ieee;
use ieee.std_logic_1164.all;
package qs is
    component quickstart is
        PORT (
            SW : IN STD_LOGIC_VECTOR(9 DOWNT0 0);
            LEDR : OUT STD_LOGIC_VECTOR(9 DOWNT0 0)
        );
    end component;
end package;
```

```
-----

LIBRARY ieee;
USE ieee.std_logic_1164.all;
ENTITY quickstart IS
    PORT (
        SW : IN STD_LOGIC_VECTOR(9 DOWNT0 0);
```

```

        LEDR : OUT STD_LOGIC_VECTOR(9 DOWNTO 0)
    );
END quickstart;

ARCHITECTURE Behavior OF quickstart IS
BEGIN
    LEDR <= SW;
END Behavior;
-----
-- pragma translate_off
library ieee;
use ieee.std_logic_1164.all;
use ieee.std_logic_unsigned.all;
use work.qs.all;

entity testbench_quickstart is
end testbench_quickstart;

architecture test of testbench_quickstart is

    signal SW : STD_LOGIC_VECTOR(9 DOWNTO 0);
    signal LEDR : STD_LOGIC_VECTOR(9 DOWNTO 0);

begin
    DUT: quickstart port map(SW,LEDR);

    process begin

        SW <= "0000000001";
        wait for 10 ns;
        report "SW = " & to_string(SW) & "; LEDR = " & to_string(LEDR);

        SW <= "0000000010";
        wait for 10 ns;
        report "SW = " & to_string(SW) & "; LEDR = " & to_string(LEDR);

        SW <= "0000000100";
        wait for 10 ns;
        report "SW = " & to_string(SW) & "; LEDR = " & to_string(LEDR);

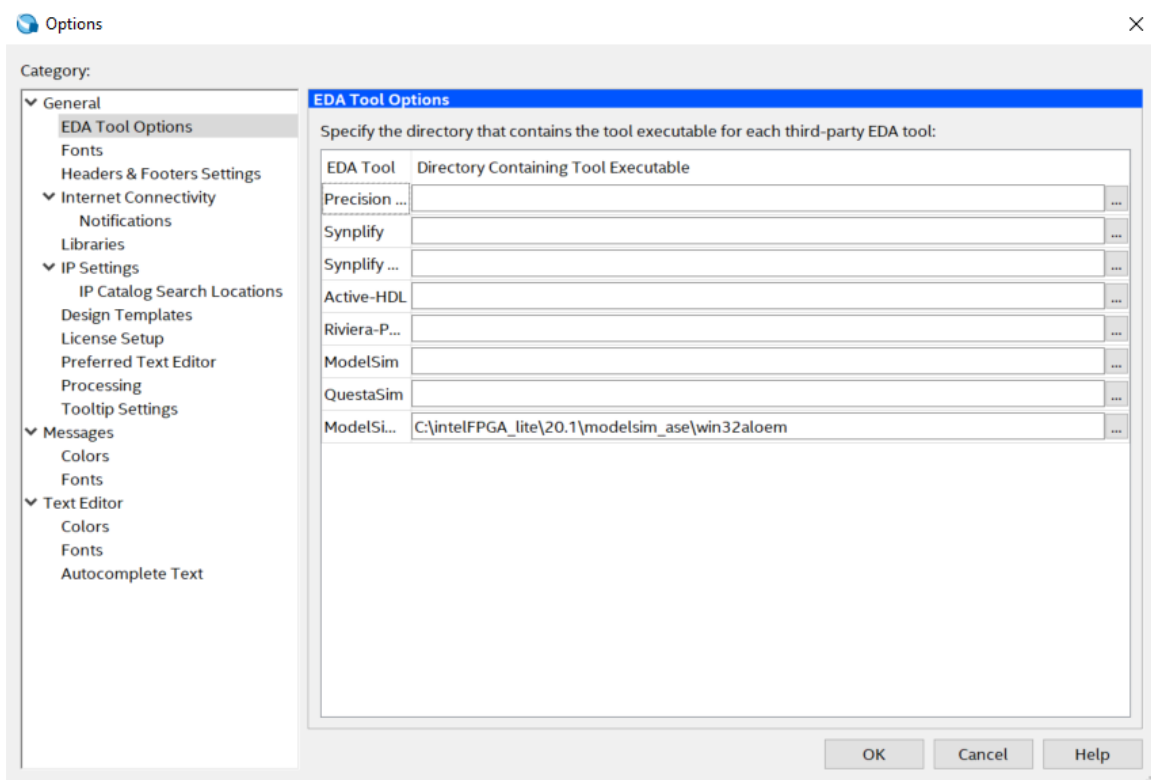
        wait;

    end process;
end architecture test;

-- pragma translate_on

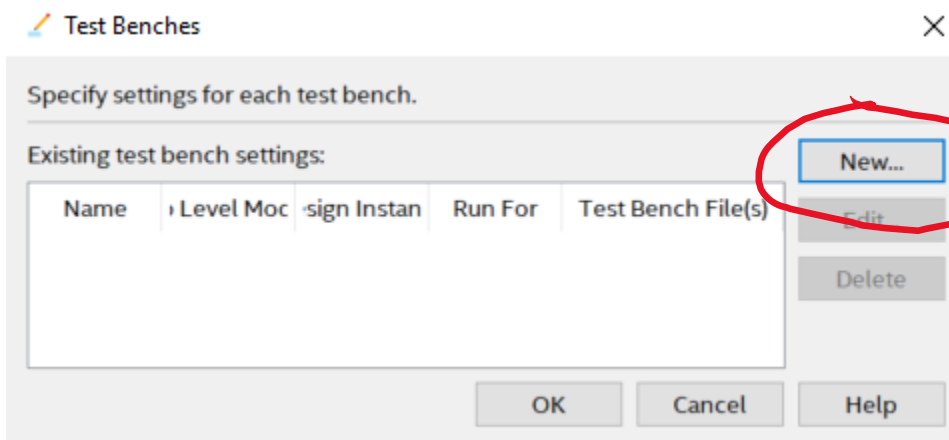
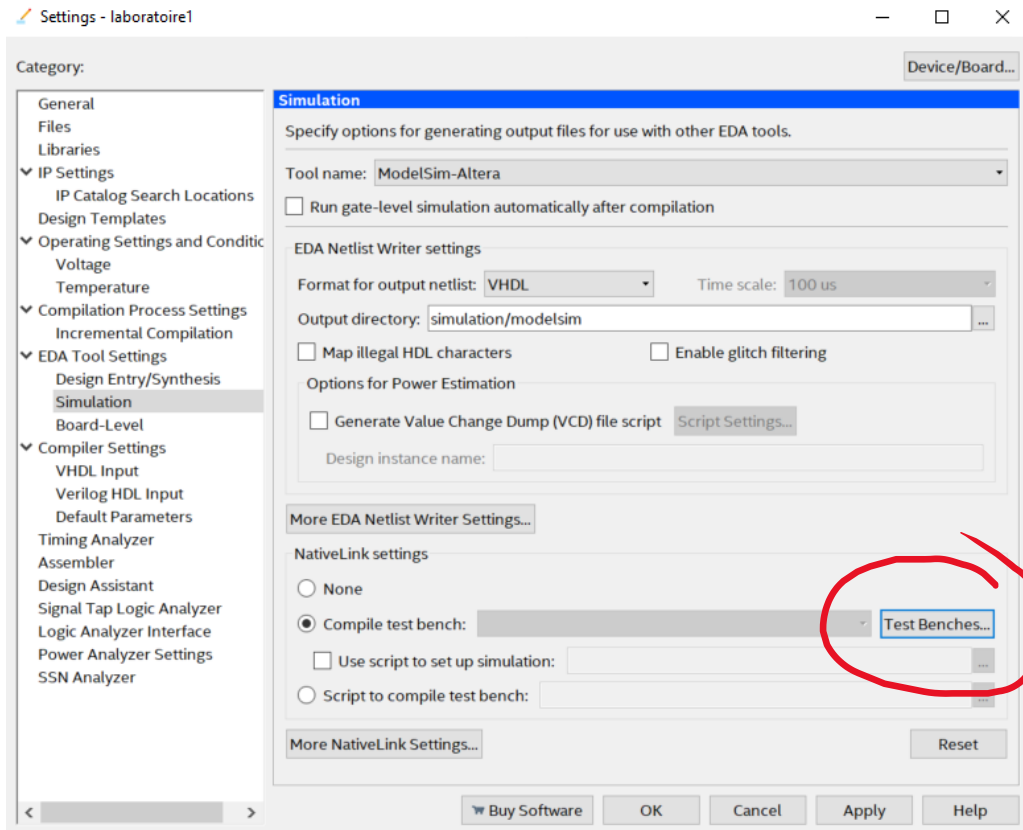
```

9) Configurer Quartus : Tools -> options



10) Ajoutez le testbench :

Assignments ► Settings ► EDA Tool Settings ► Simulation.



New Test Bench Settings

Create new test bench settings.

Test bench name: testbench_quickstart

Top level module in test bench: testbench_quickstart

☐ Use test bench to perform VHDL timing simulation

Design instance name in test bench: NA

Simulation period

☒ Run simulation until all vector stimuli are used

☐ End simulation at: S

Test bench and simulation files

File name: Add

File Name	Library	HDL Version
quickstart.vhd		Default

Remove Up Down Properties...

OK Cancel Help

11) Démarrez la simulation:

Tools ► Run Simulation Tool ► RTL Simulation

12) Le logiciel ModelSIM démarre.

Vous devriez voir ceci :

