Quartus II Software

Quick Start Guide

1. Introduction

Quartus II is an integrated design software for Altera's FPGAs and CPLDs. It's an easy-to-use platform including all the necessary tools for every stage of your FPGA design. Two editions(Subscription Edition and Web Edition) are provided with different performance and productivity. Quartus II Subscription Edition has higher performance, supporting more devices, but a valid license is required, users have to buy a license from Altera or start a 30-day free trial. While Quartus II Web Edition is a free, no license required version which is targeted for CPLD and medium-density FPGA. In addition, Altera provides a broad range of tool chains for FPGA hardware and software design. For more information, please visit Design Software page on Altera website.

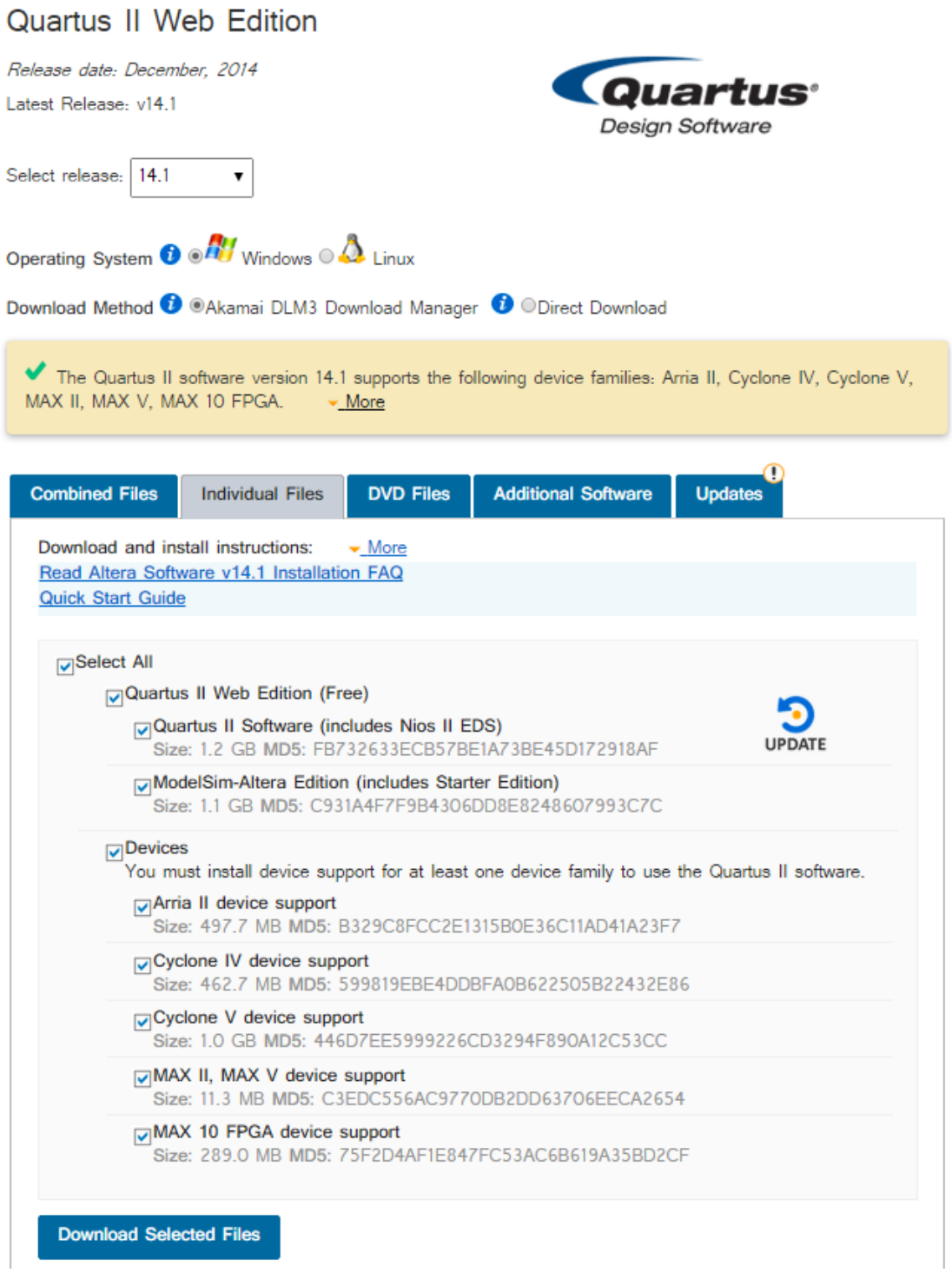
This guide will briefly introduce you how to use Quartus II design software to create a simple FPGA project, compile the project with an object file output and ultimately download the file to the internal RAM of target FPGA device. Also, the steps of programming an external serial configuration device via JTAG interface will be described.

Note: This guide is intended to give you a intuitive view of creating a simple FPGA project using Quartus II software. In fact, many important procedures of FPGA design are omitted. If you want to learn Quartus II software in depth, please read the Quartus II Handbook for help.

2. Get Tools Ready

2.1 Download Quartus II Web Edition Software

If you are a beginner for Altera FPGA and haven't installed any edition of Quartus software in your PC. Please go to Altera's download center and choose the best release version for you. You have to register an account on Altera website before downloading is allowed.



As the image shown above, the Quartus II software and the device packages are provided separately by Altera. Quartus II software is a must, but you can selectively download the device package. Usually, it's unnecessary to download all the device packages unless you need them all.

Note: Several release versions are provided by Altera, and different versions support different devices. Please notice that the latest version doesn't support some old devices, and vice versa. The table below shows the latest supported Quartus II software for the Cyclone series FPGA. You should decide which version is suitable according to the device on your board. Usually, the newer version always requires more hardware resource for your PC. Quartus II Web Edition 14.1(the latest release) is used for demonstration in this guide.

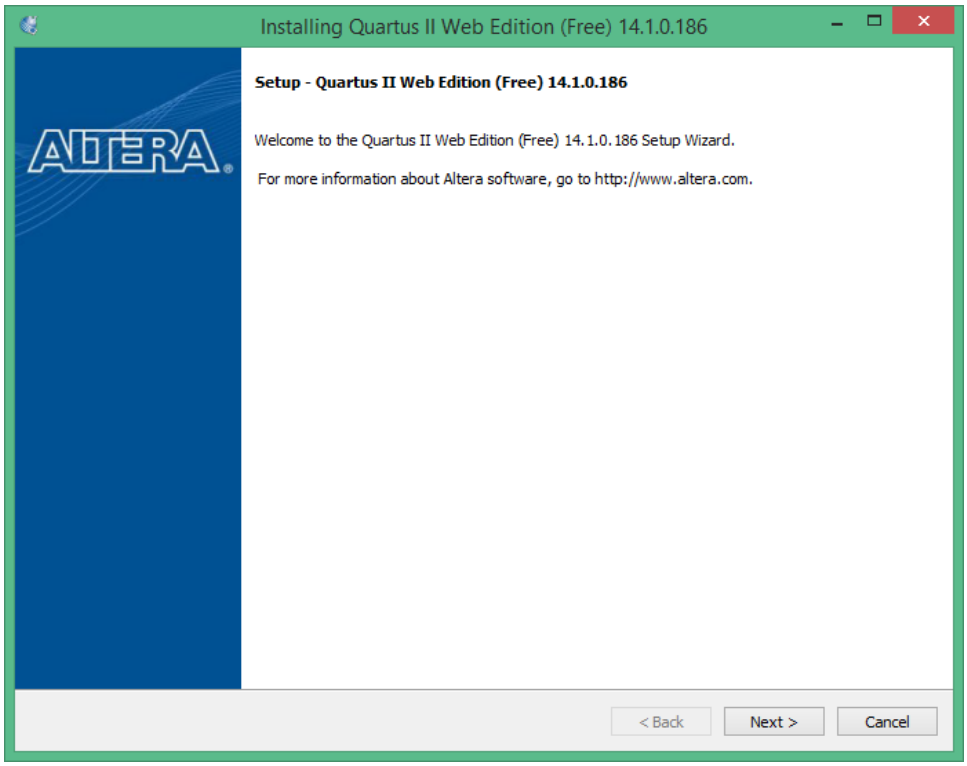
Cyclone Series

Family	Latest Supported Quartus II Version (Subscription)	Latest Supported Quartus II Version (Web)
Cyclone V	Latest Release	Latest Release
Cyclone IV GX	Latest Release	Latest Release
Cyclone IV E	Latest Release	Latest Release
Cyclone III LS	13.1	13.1
Cyclone III	13.1	13.1
Cyclone II	13.0sp1	13.0sp1
Cyclone	13.0sp1	11.0sp1

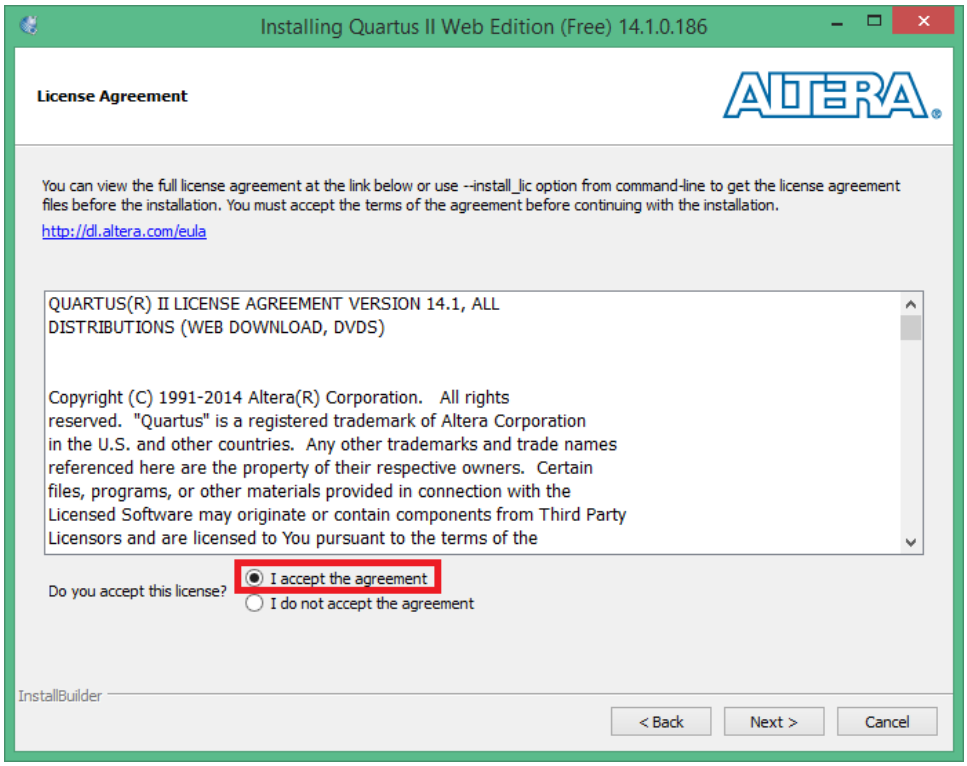
Note: *We have released the* ***EP1C3 Starter Board*** *based on the first generation Cyclone FPGA —EP1C3T144, so Quartus II Web Edition 11.0sp1 or older version is recommended for this board.*

2.2 Install Quartus II Software

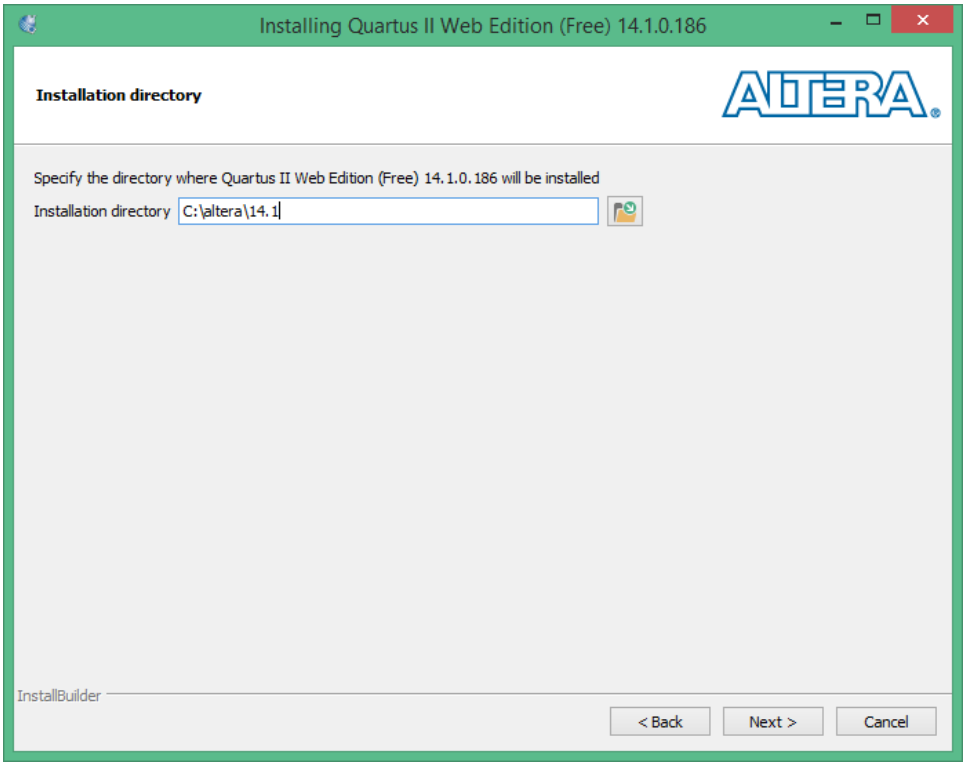
1) Double-click the setup executable file to start the installation wizard. Click the **Next** button to next window.



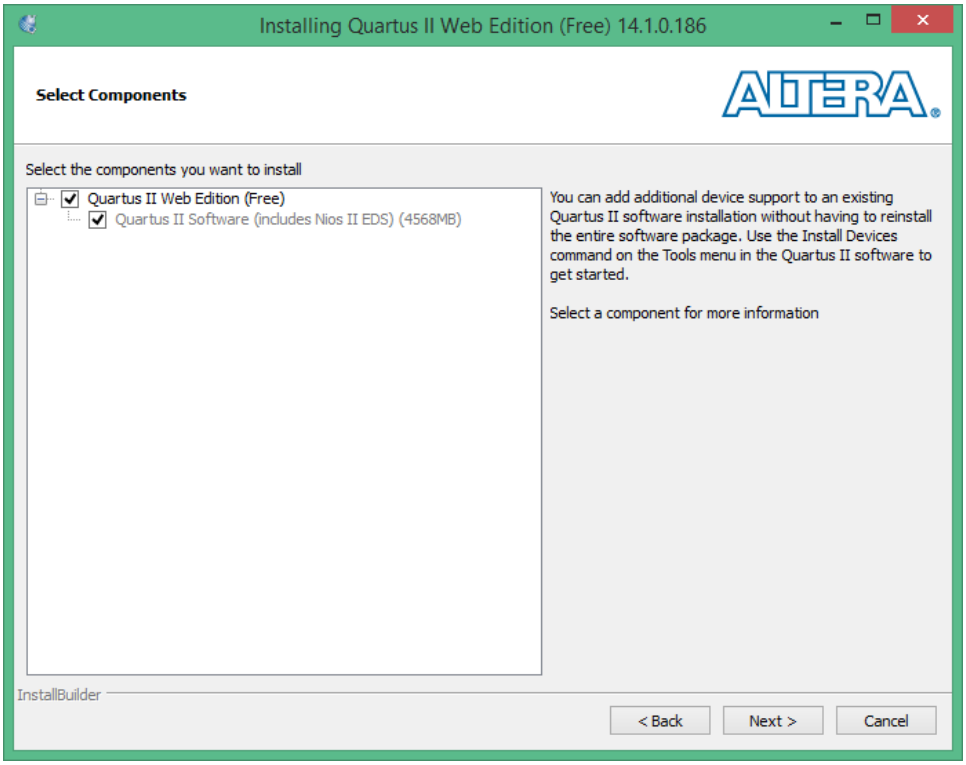
2) Click the "**I accept the agreement**" radio button, and click **Next** button to next window.



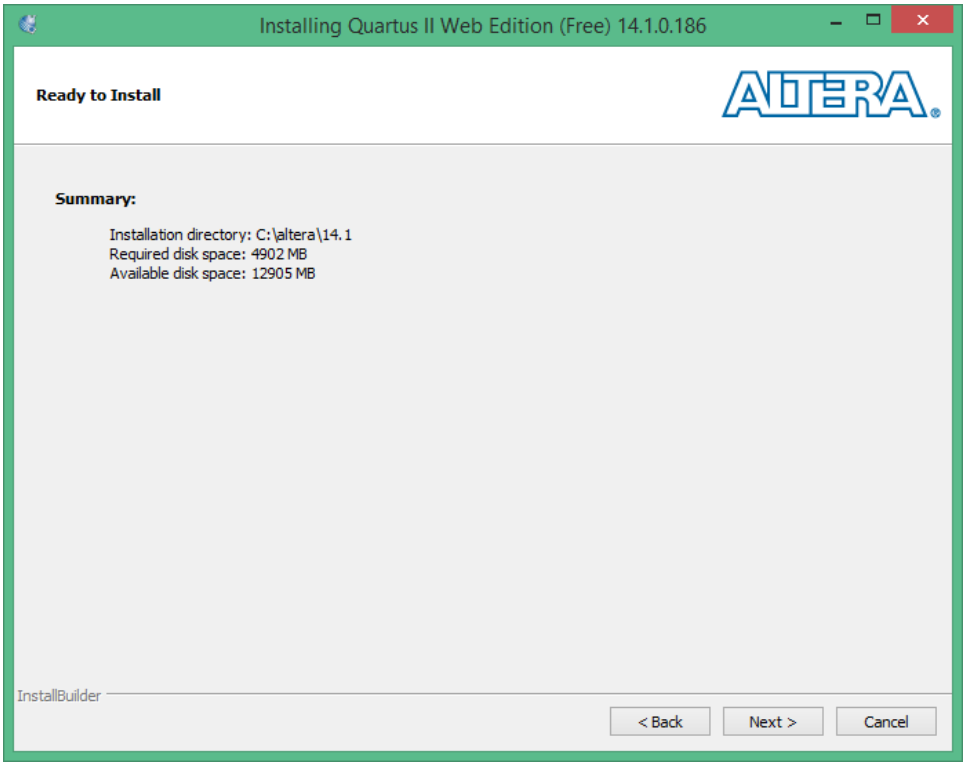
3) Specify the installation directory, click **Next** button to next window.



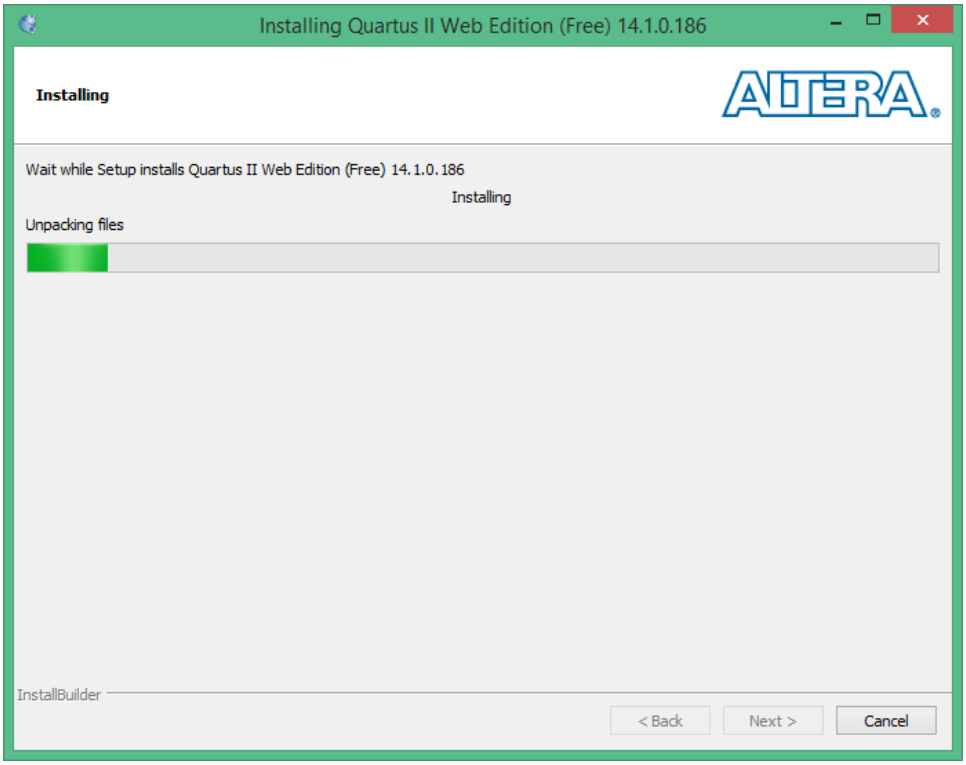
4) The check box is enabled by default. Click **Next** button to next window.



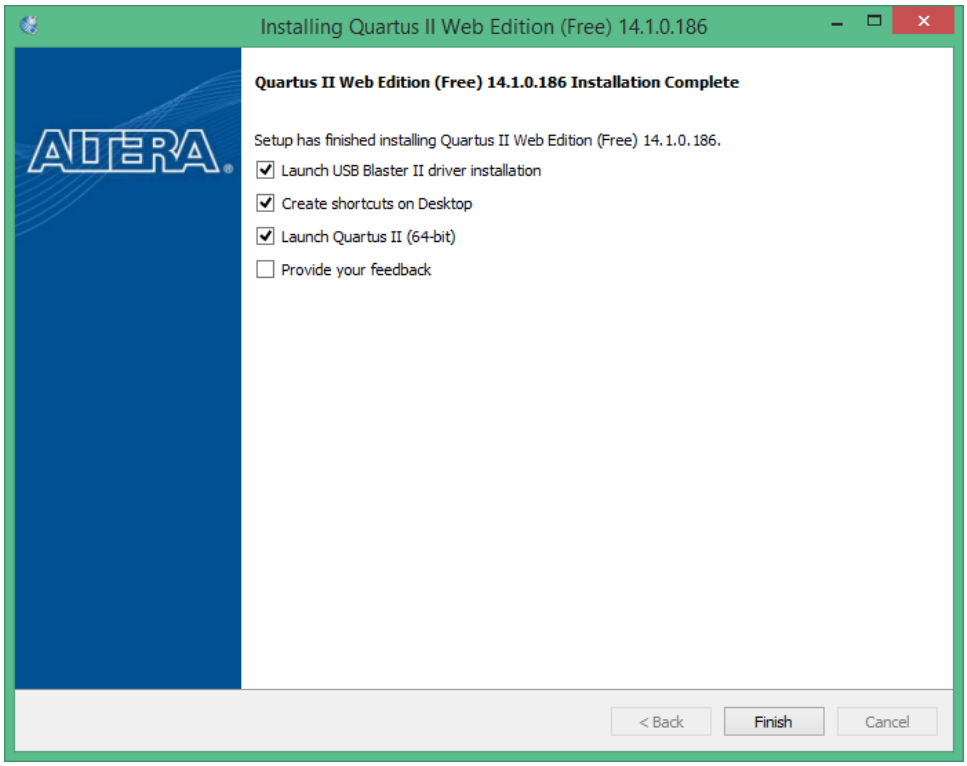
5) A summary window appears. Click **Next** button to start the installation process.



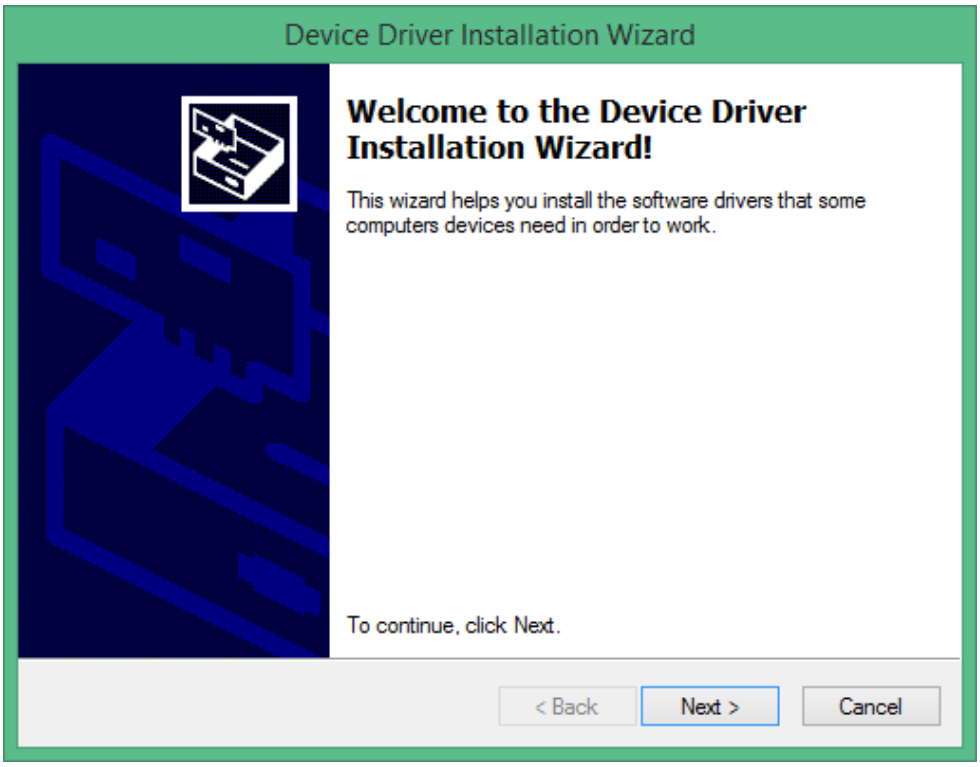
6) Please wait patiently. This will take several minutes.



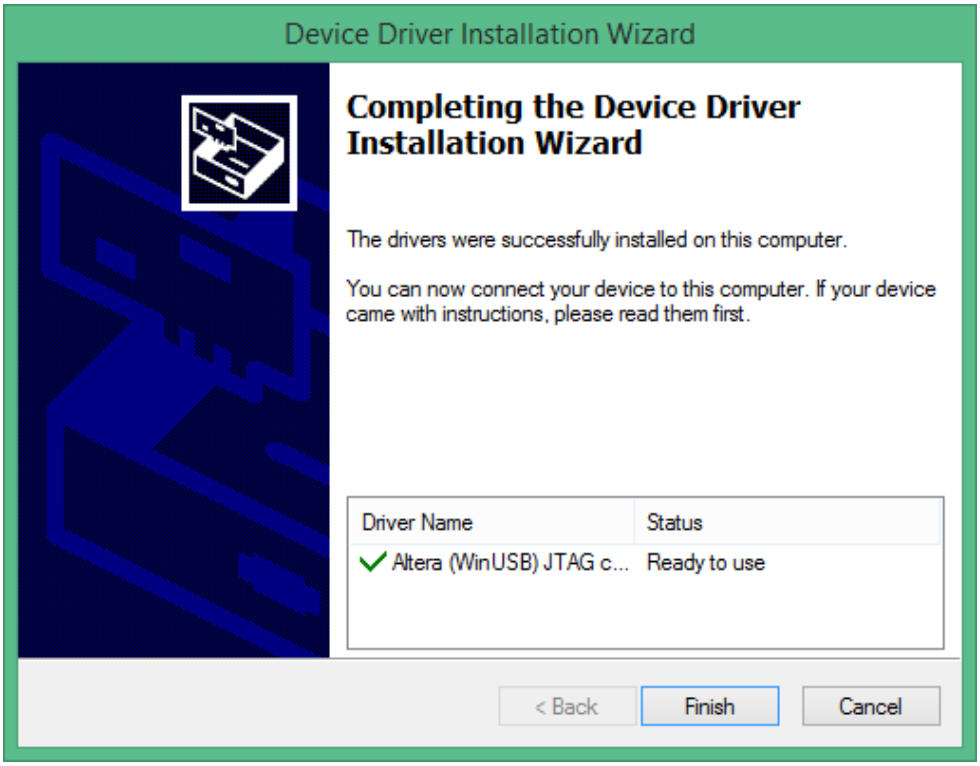
7) When the installation completed, a window appears and guides you to install the USB Blaster driver software. Click **Finish** button.



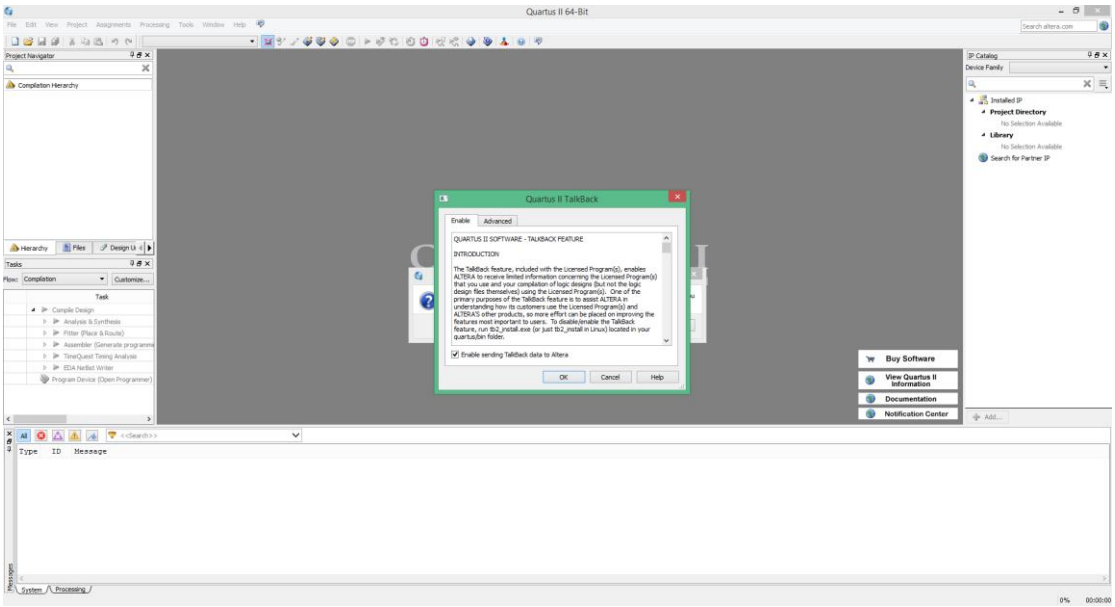
8) This is the driver installation wizard. Click **Next** button to install the driver.



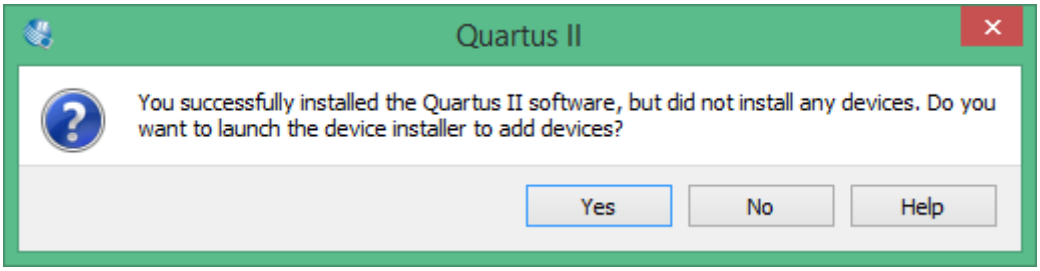
9) Click **Finish** button to complete the wizard.



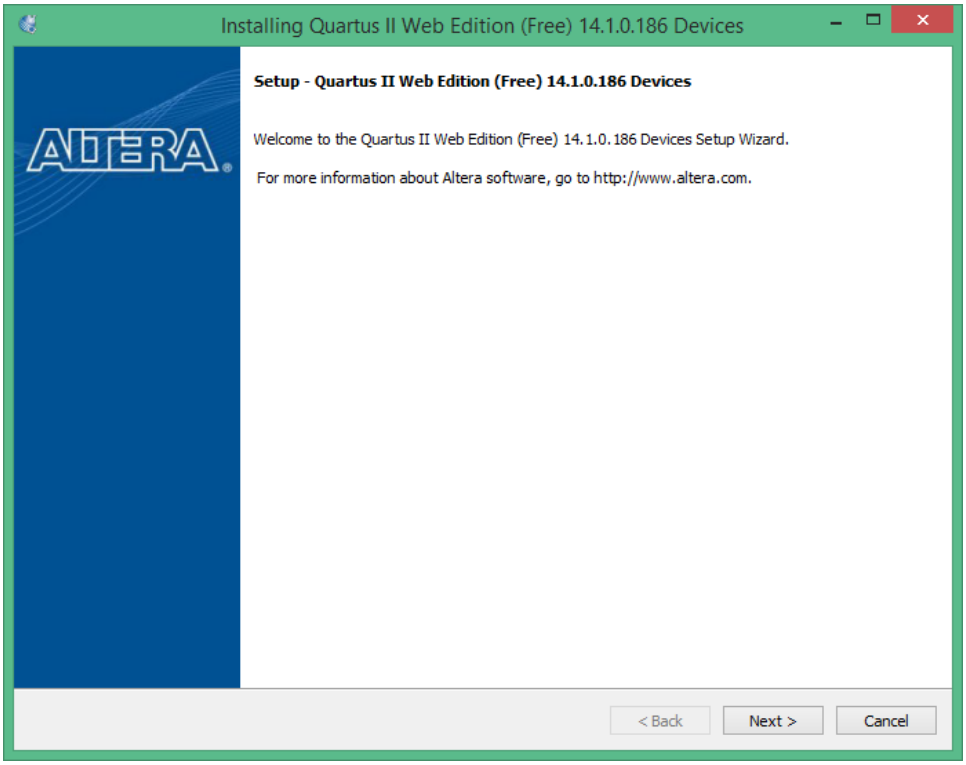
10) The Quartus II software will be automatically launched. Enable or disable the Quartus II Talkback function as you like.



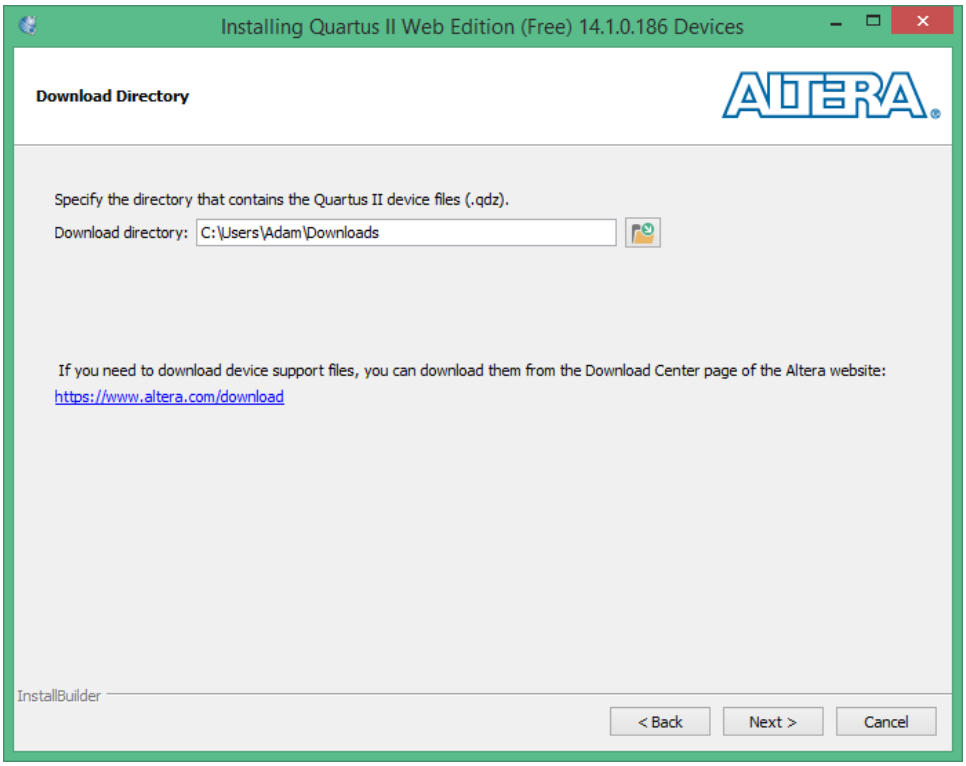
11) A message window appears warning you that no devices have been installed. Click **Yes** button to install the device package.



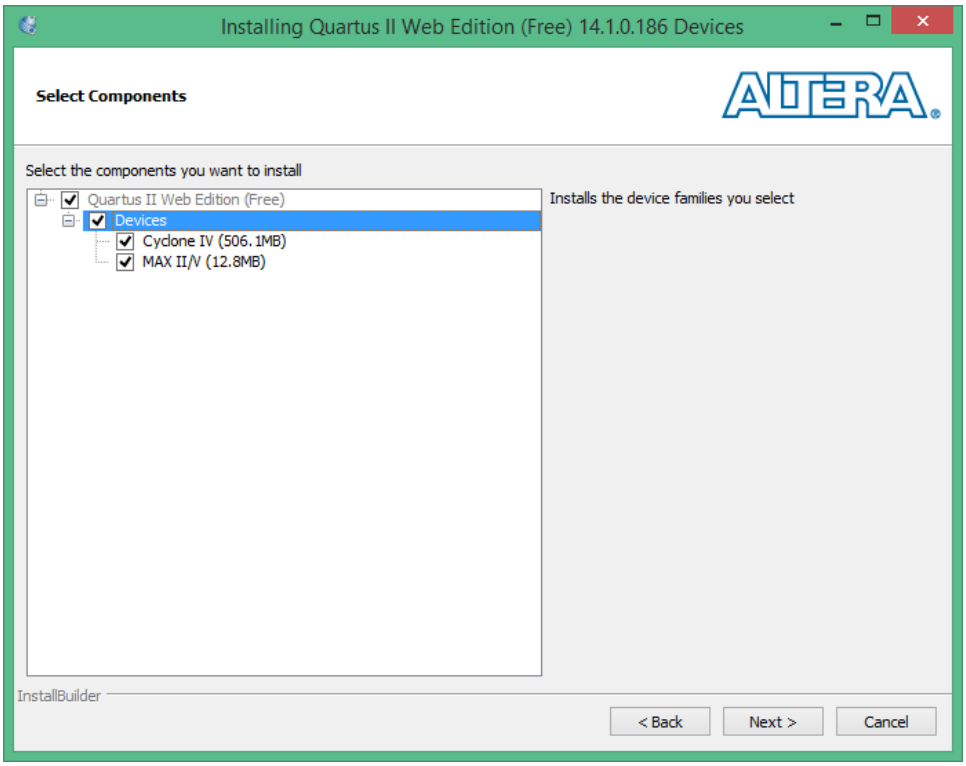
12) The devices setup wizard is launched. You must ensure that you have downloaded the device packages you need from Altera website. Click **Next** button to next window.



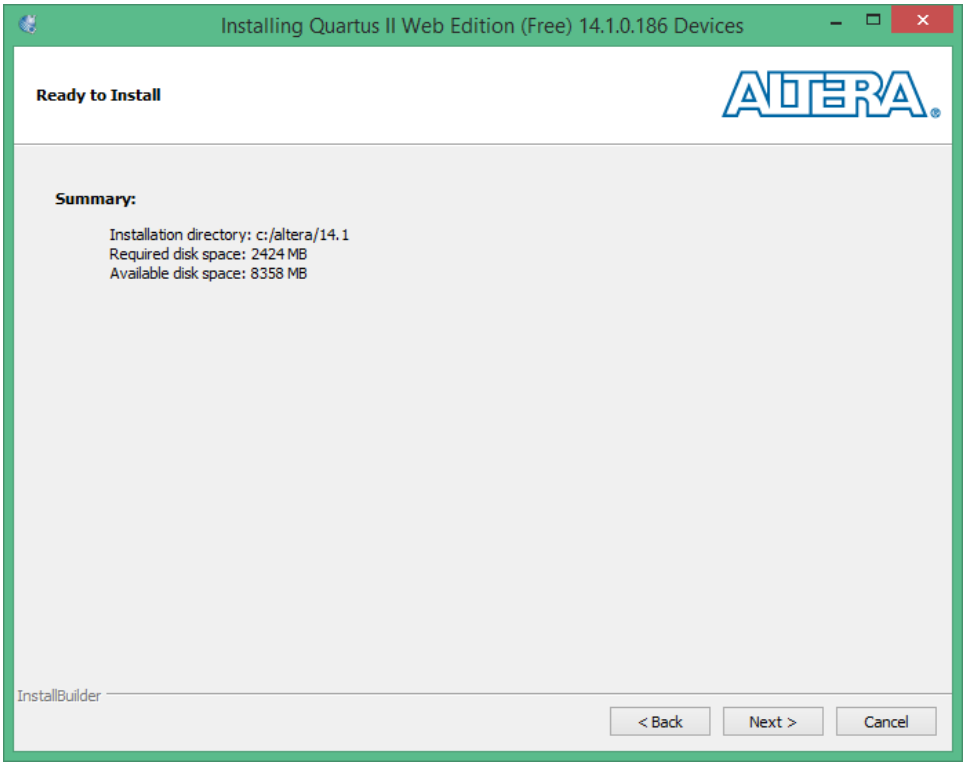
13) Specify the location where the device package is saved. Click **Next** button to next window.



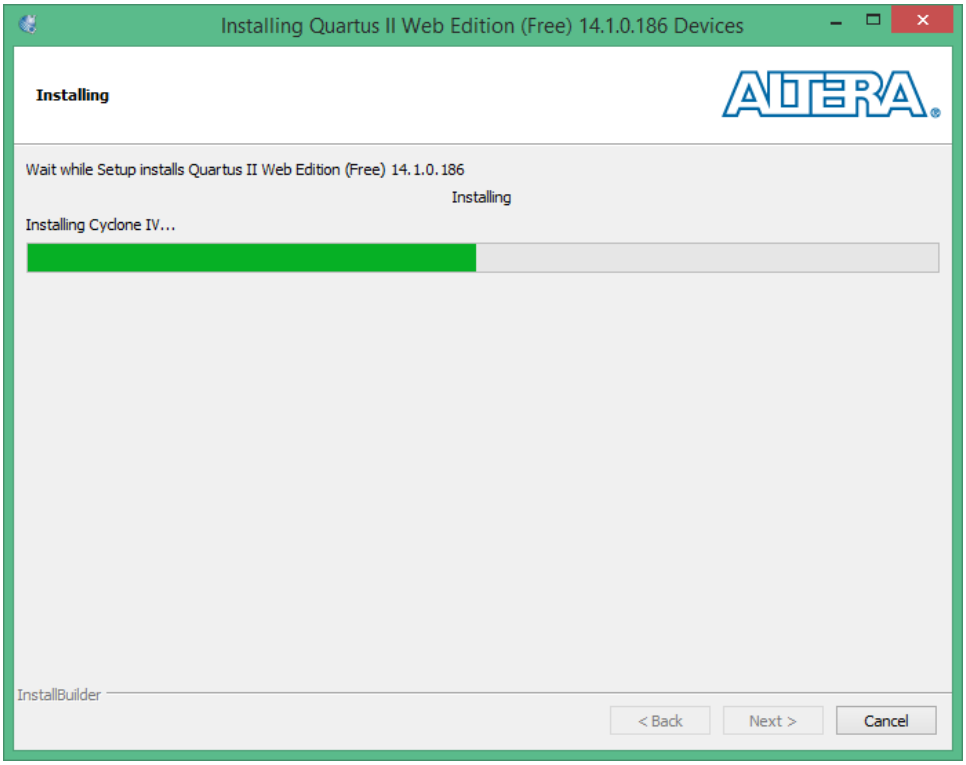
14) The devices available will be automatically listed in the window. Enable the relative check box if you want the device to be installed. Click **Next** button to next window.



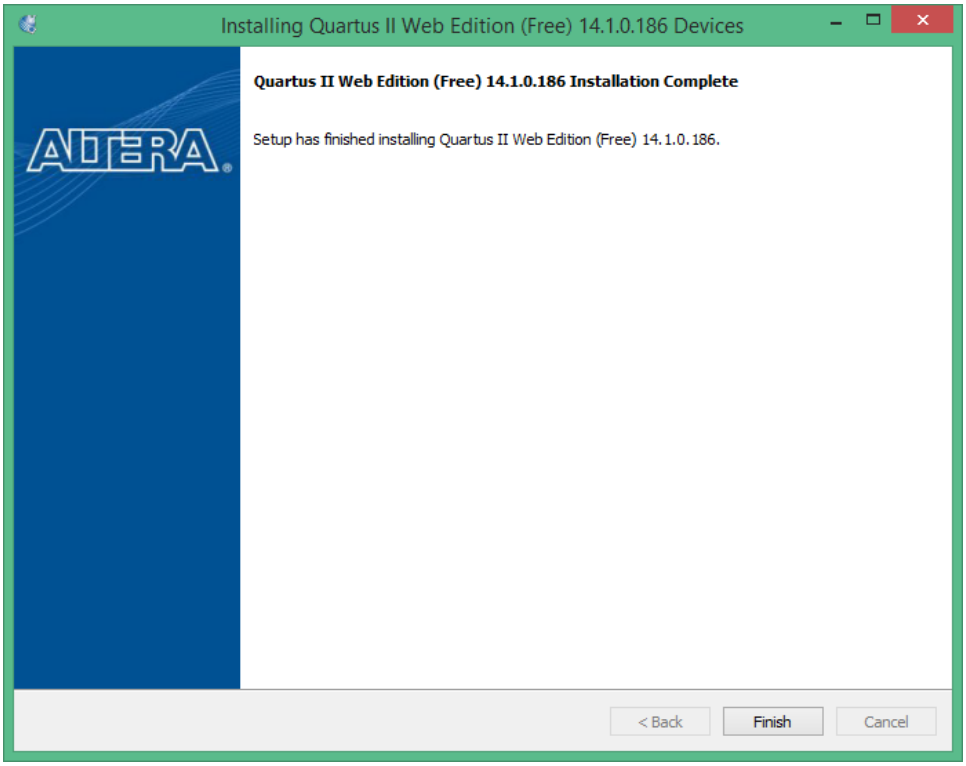
15) Click **Next** button to start installing.



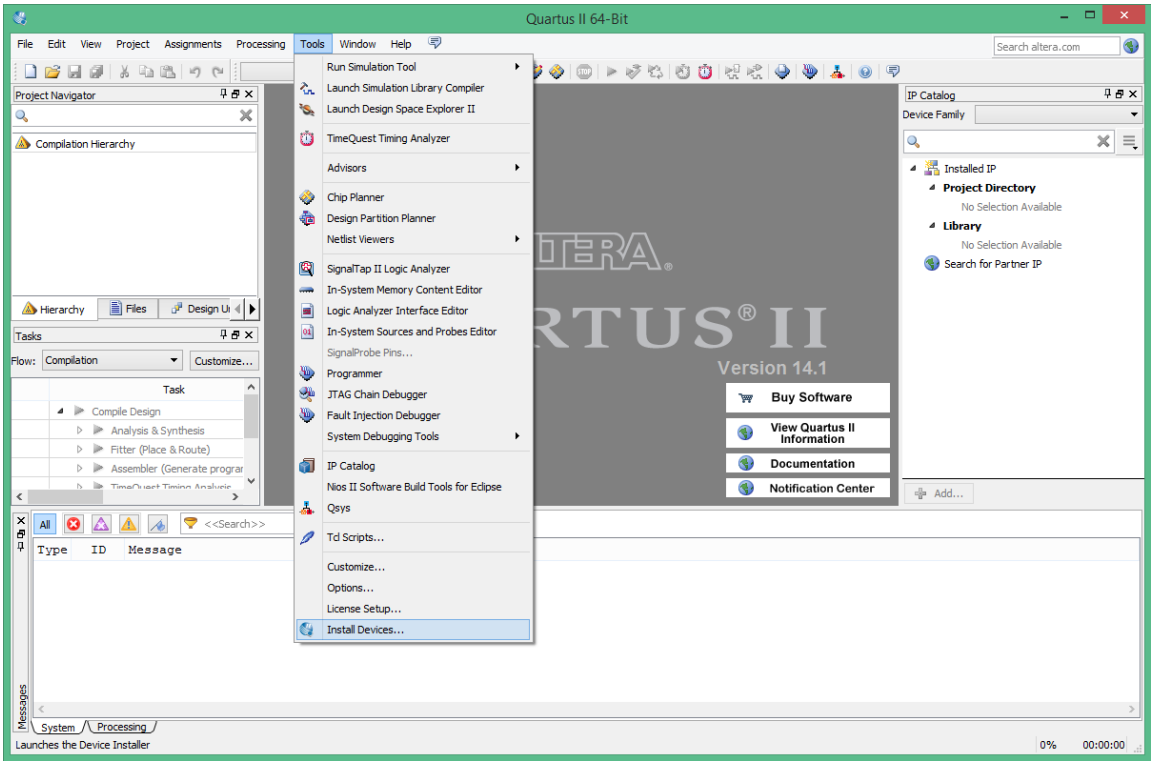
16) Please wait while the installation is in progress.



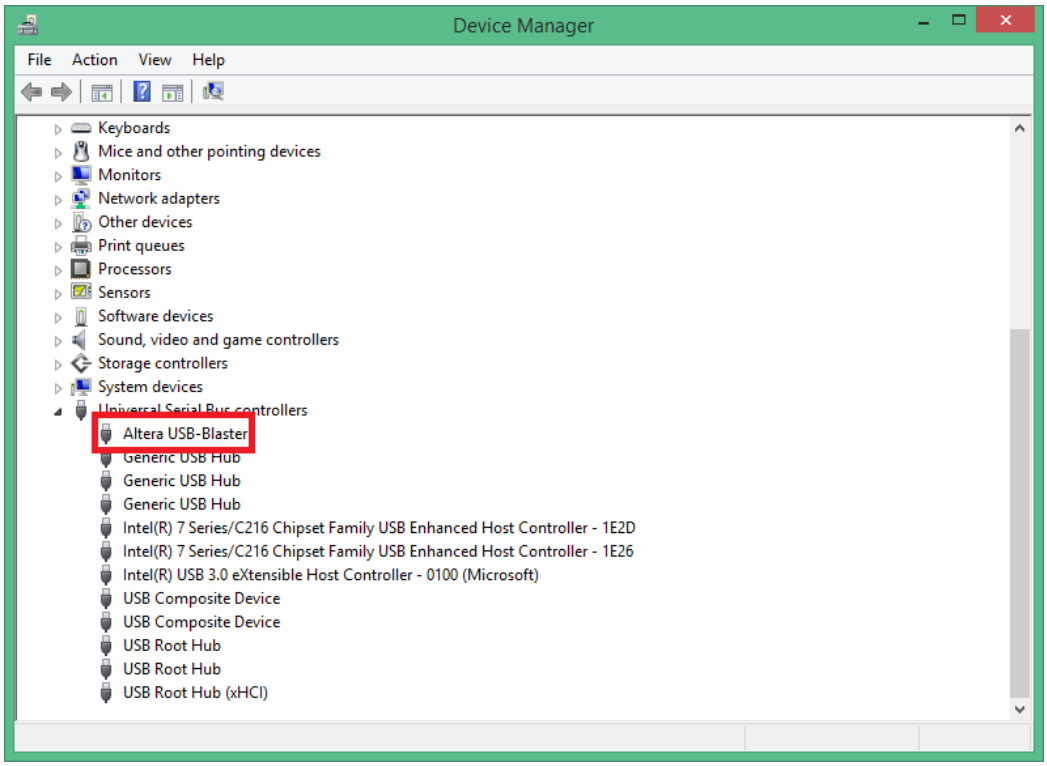
17) Click **Finish** button to exit the wizard.



18) If you want to install a new device in the future, please go to **Tools -> Install Devices...** in Quartus II main window. Quartus II also provides a standalone **Device Installer** software, you can find its shortcut in your Start Menu.



19) Right now, you have successfully installed the Quartus II software and the driver software for USB Blaster. You can plug USB Blaster to your PC, and check if the cable is successfully recognized.



20) You have to reinstall the USB Blaster driver if it was not properly installed in the previous steps. The driver software is located under **..\ altera\14.1\quartus\drivers** directory.

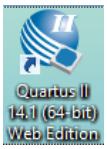
If you don't know how to install the USB Blaster driver. Please read USB-Blaster Download Cable User Guide first, or Google the subject for help.

3. Create a new project

Quartus II software uses the concept of project to manage the input and output files in FPGA design. It's easy to understand what a project is if you have some experience in C code development for a microcontroller. This section will walk you through the process of creating a new FPGA project. Verilog HDL is used to describe the function of the hardware. The project will be compiled with an object file output. The generated configuration file will be downloaded to the target device to check if the hardware is running as we expected. This is a very simple project and we name it LED_Blink. Just as its name indicates, one LED will blink with a fixed interval. Ultimately, the hardware will be verified on EP4CE6 Mini Board.

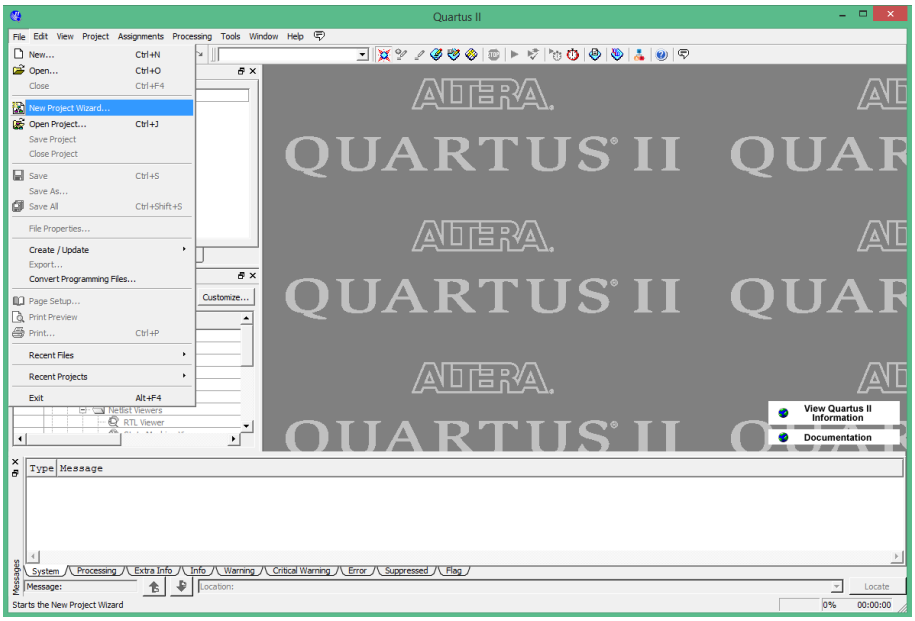
Note: In this guide, you don't need to understand the details of the example code if you don't have any knowledge of Verilog HDL.

1) Start Quartus II software by double-clicking the shortcut icon.

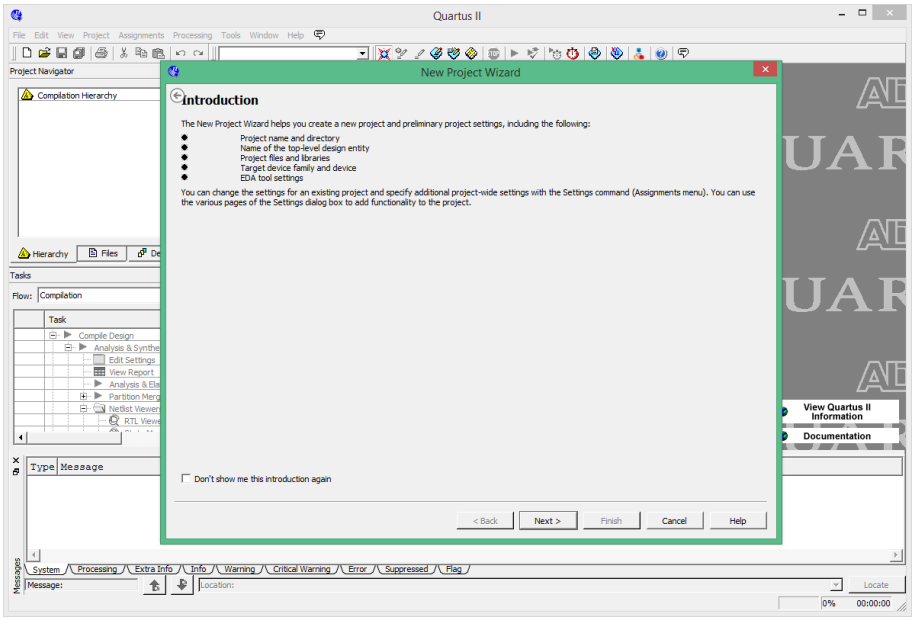


A floating window appears helping you get a quick start for designing a project or learning Quartus II software. You can disable this startup window by clicking the **Don't show this screen again** check box.

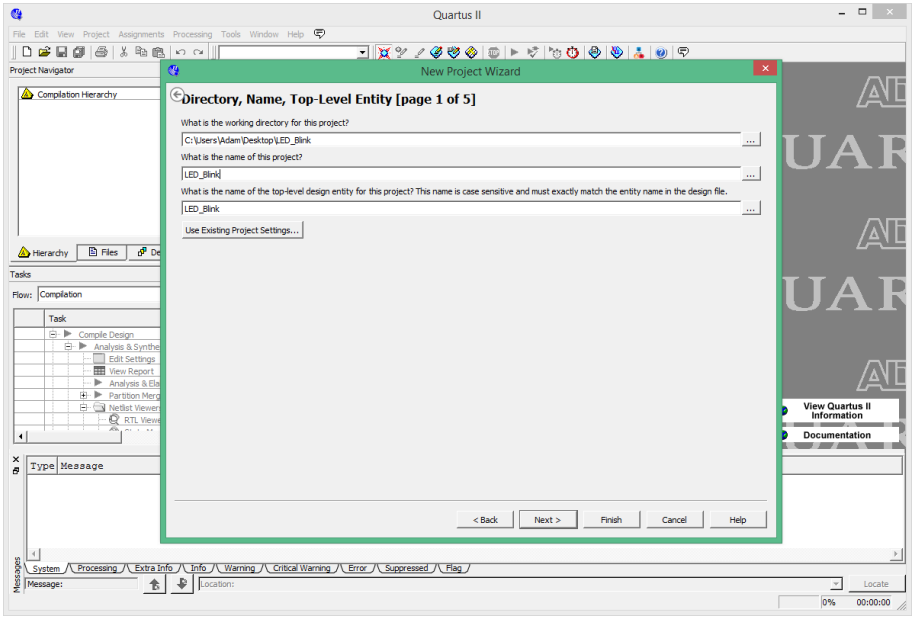
2) Go to **File -> New Project Wizard...**



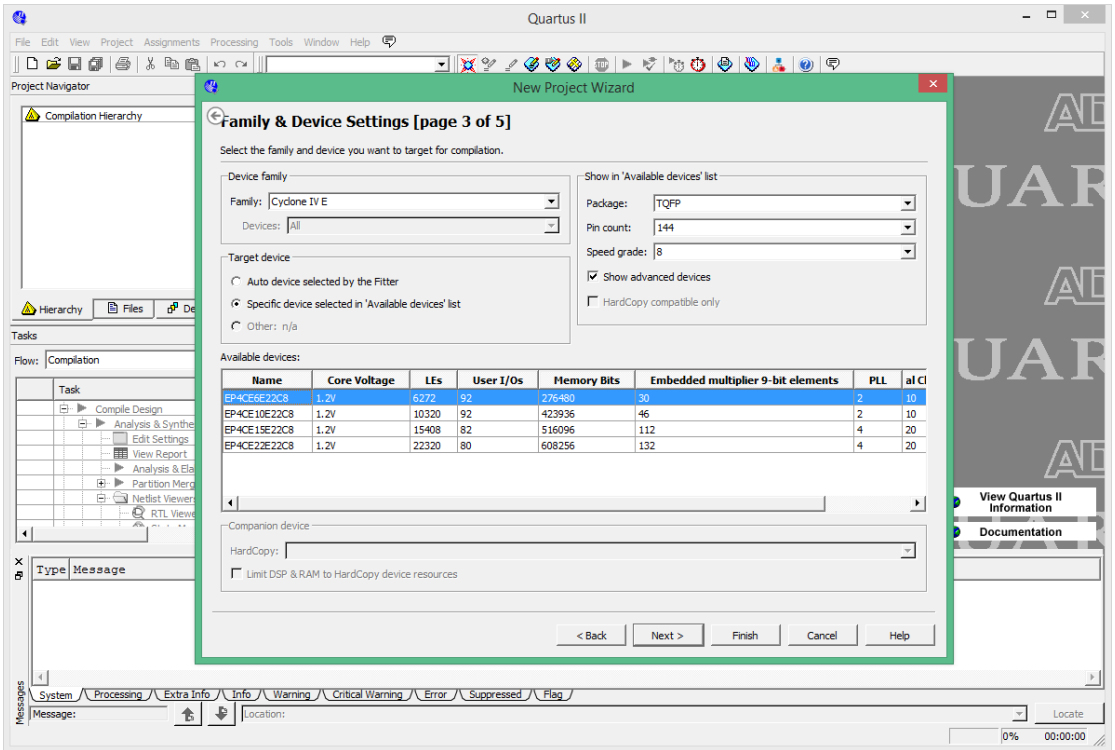
3) An introduction window tells you what you should do next when creating a project. Click **Next** button to next window.



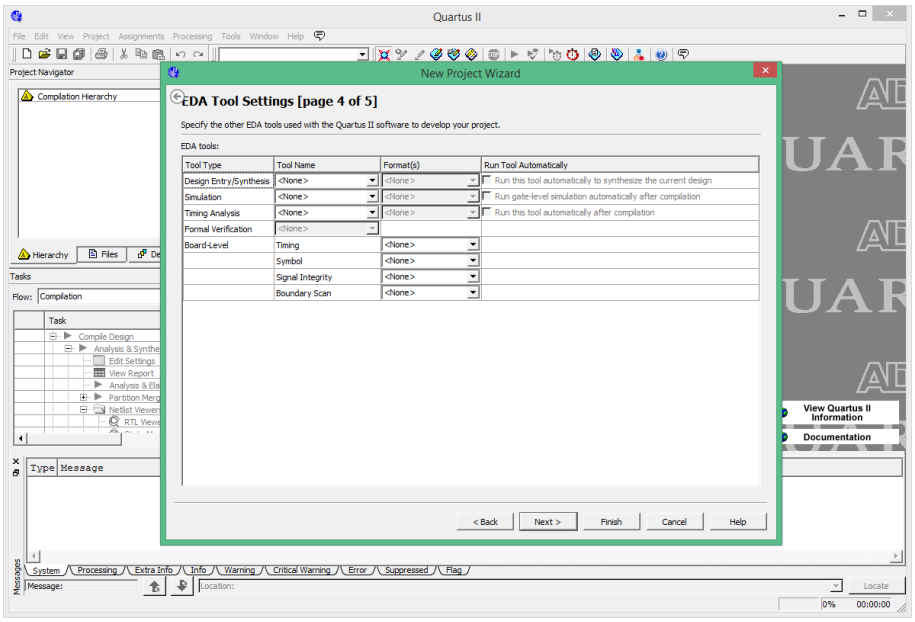
4) Specify the directory for this project and type the project name and top-level entity name as shown below. Click **Next** button to next window.



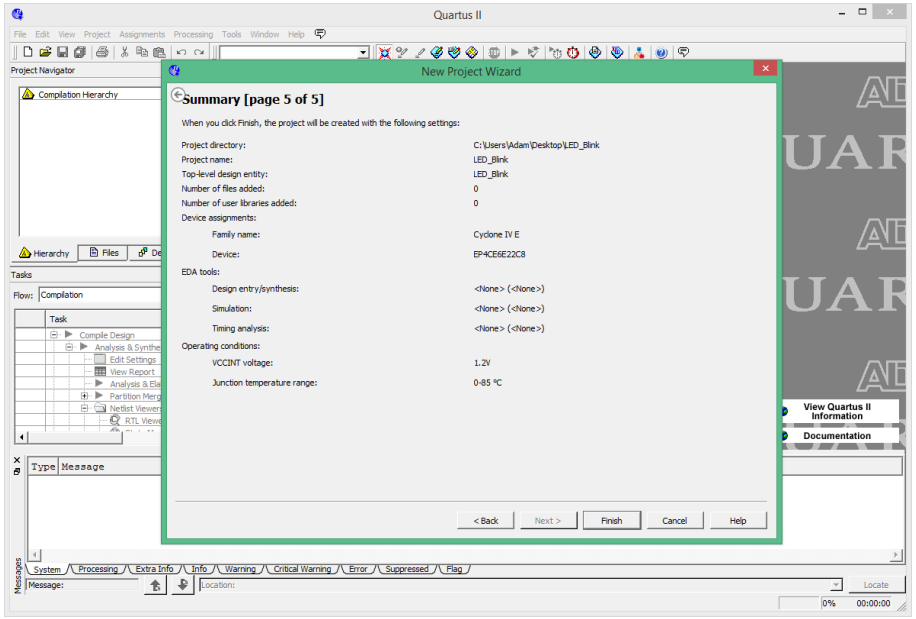
5) Select appropriate device for this project (EP4CE6E22C8N is stuffed on **EP4CE6 Mini Board**), click **Next** button to next window.



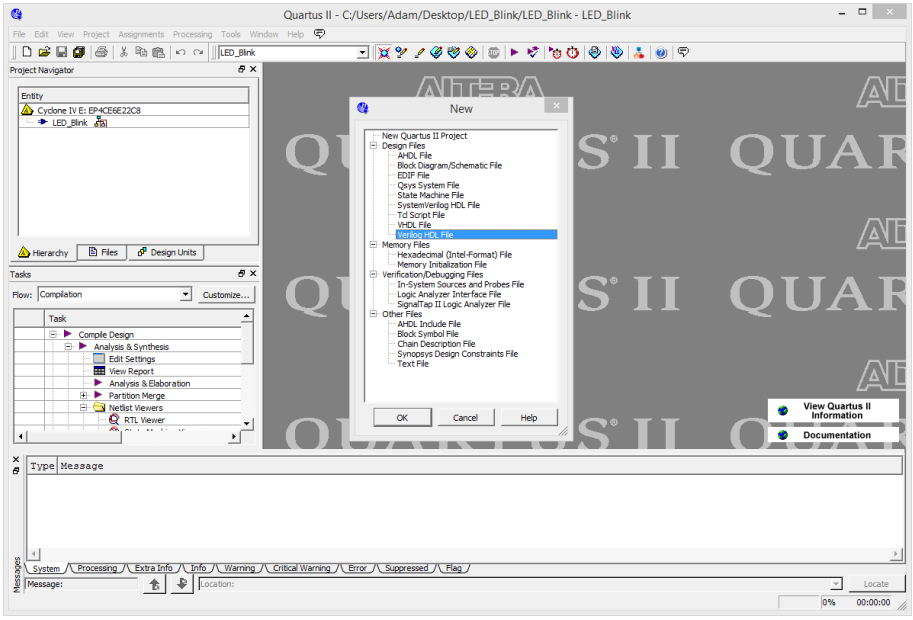
6) Ignore the EDA tool settings in this step, because we will not simulate the project. Click **Next** button to next window.



7) A summary window lists all the settings you specified in previous steps. Click **Finish** button to exit this wizard.



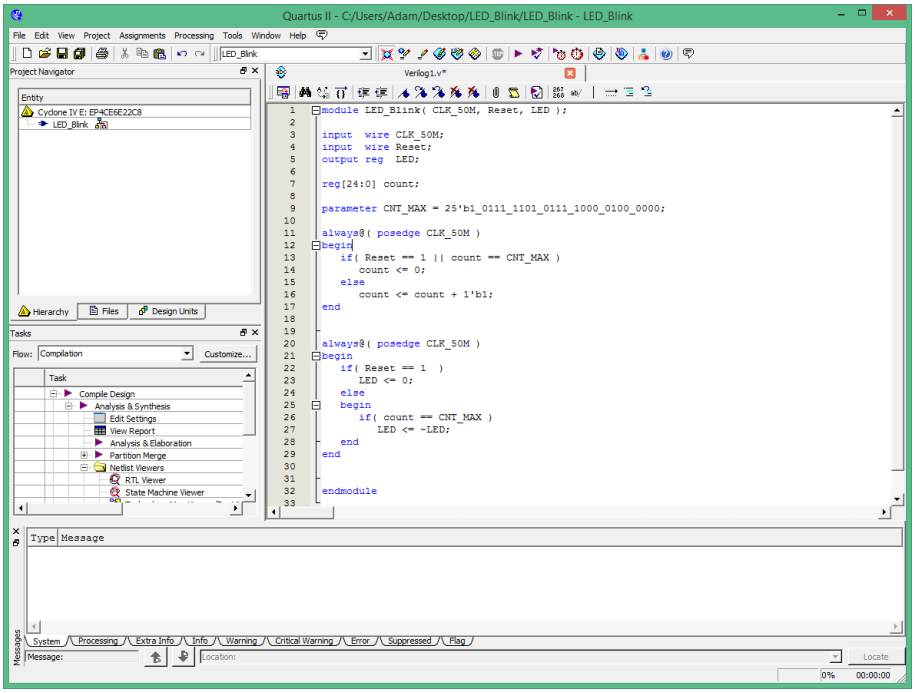
8) The main window of Quartus II software will appear in front of you. You can spend some time to tour around and get familiar with the layout of the user interface. Then click **File -> New...** ,and select **Verilog HDL file** to create a new HDL code file for this project.



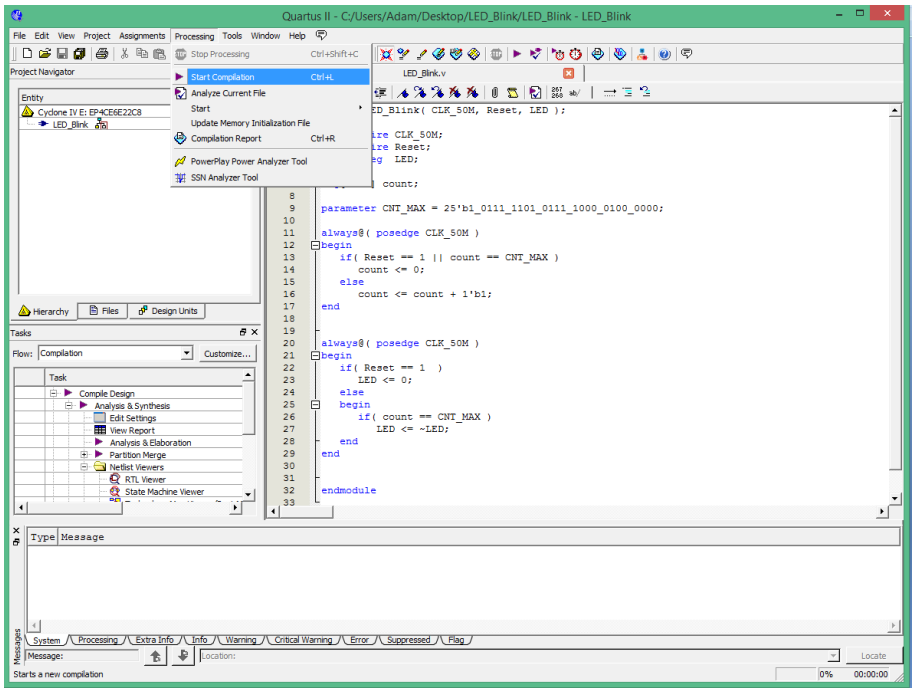
9) Copy the following code to the blank file you just created. Save the file as **LED_Blink.v**.

```
module LED_Blink( CLK_50M, Reset, LED );
input   wire CLK_50M;
input   wire Reset;
output reg LED;

reg[24:0] count;

parameter CNT_MAX = 25'b1_0111_1101_0111_1000_0100_0000;

always@( posedge CLK_50M )
begin
    if( Reset == 1 || count == CNT_MAX )
        count <= 0;
    else
        count <= count + 1'b1;
end

always@( posedge CLK_50M )
begin
    if( Reset == 1   )
        LED <= 0;
    else
    begin
        if( count == CNT_MAX )
            LED <= ~LED;
```

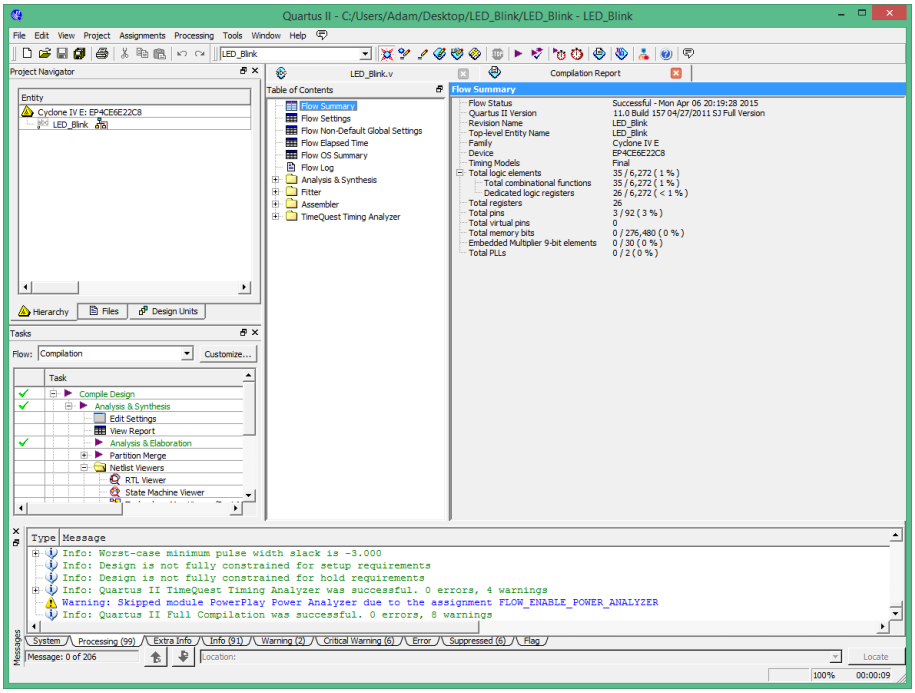
```
        end
    end
    endmodule
```



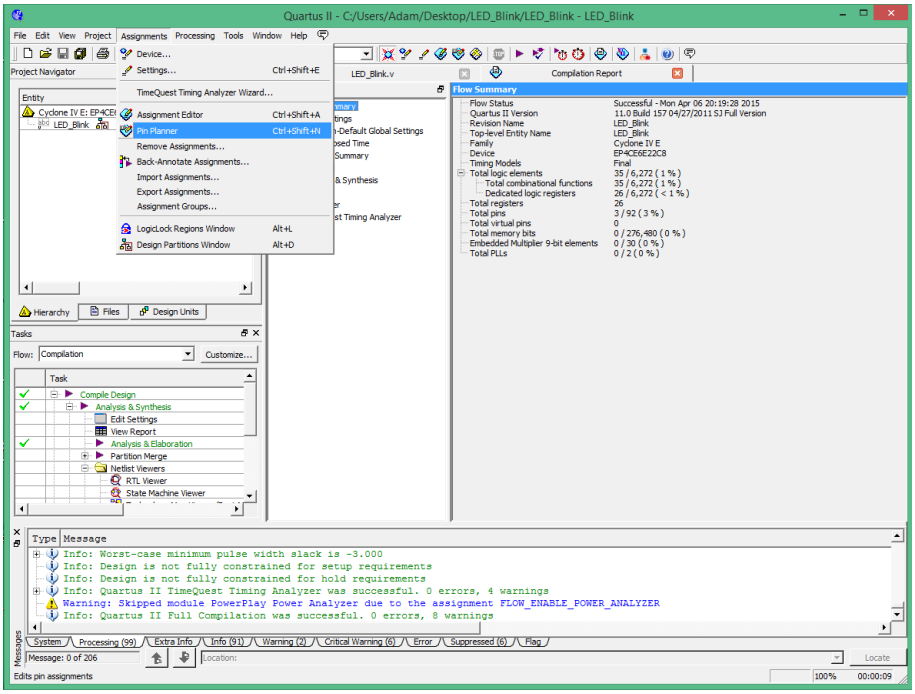
10) Click **Processing -> Start Compilation** to compile this project.



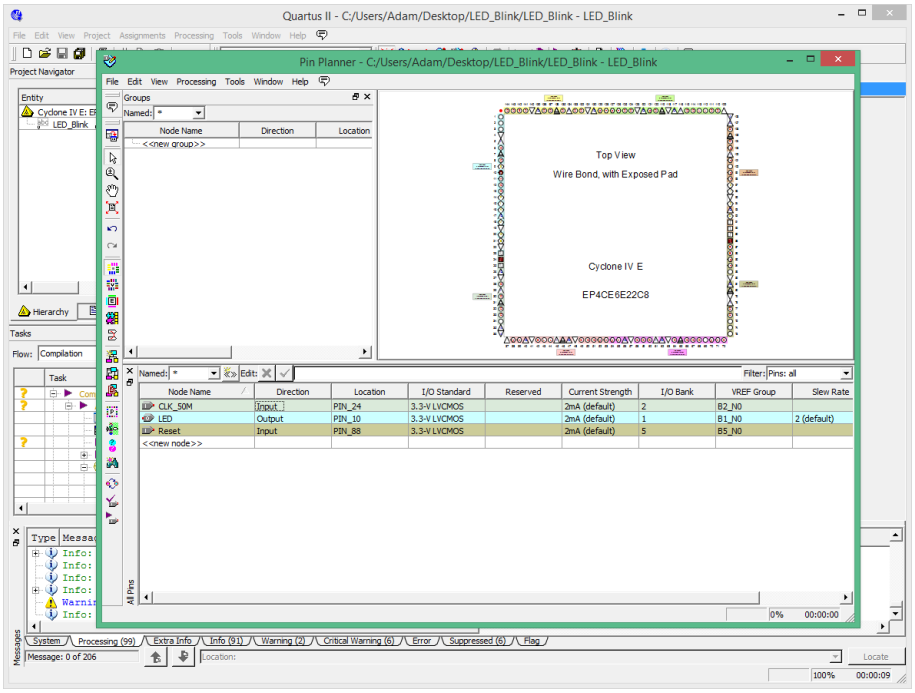
11) After compilation, a report window outlines the result of this compilation.



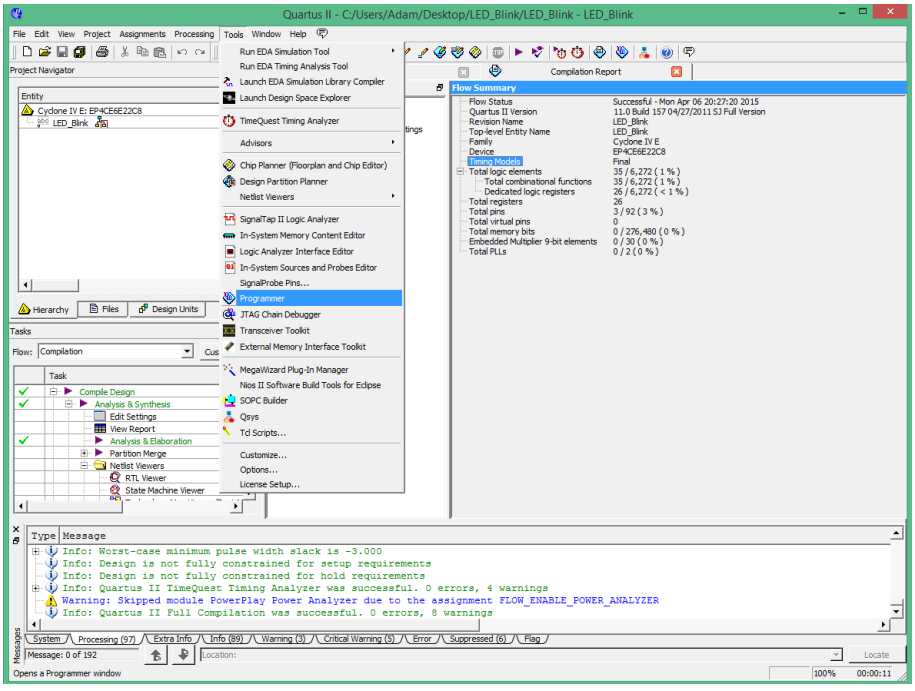
12) Click **Assignments -> Pin Planner**, to assign the input/output pins for the **LED_Blink** module.



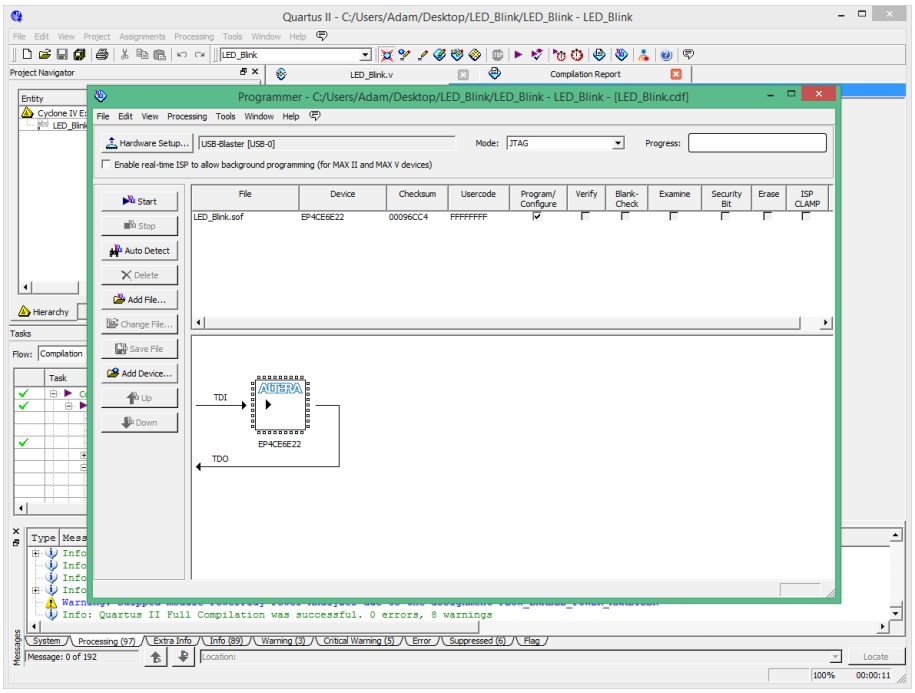
13) Enter the pin number for each signal in the **Location** column as shown below.



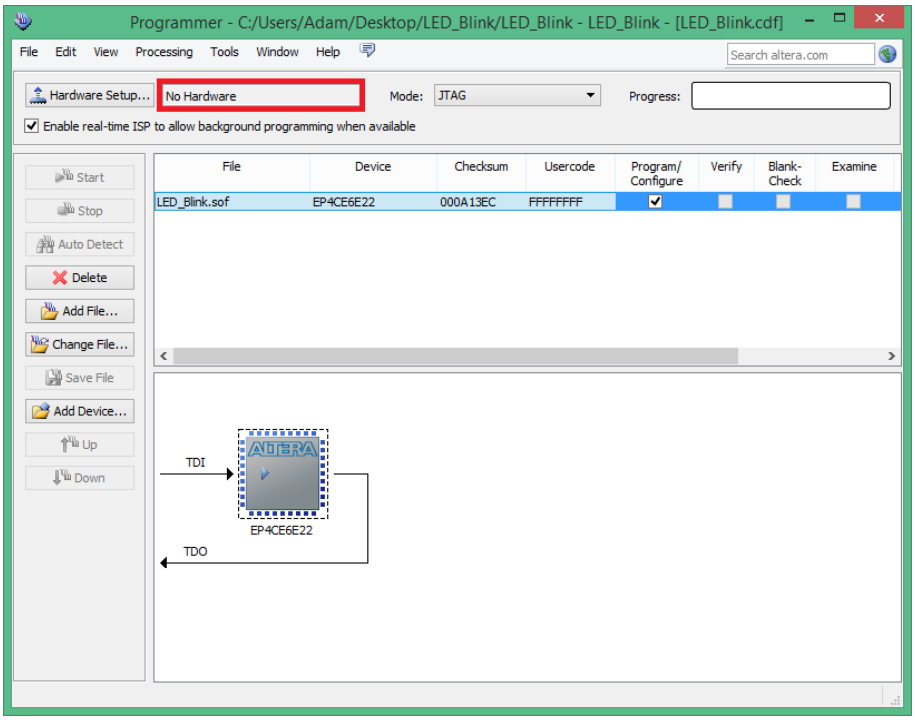
14) Recompile this project. This time the pin assignment information will be added during compilation. Power the EP4CE6 Mini module and connect the USB Blaster cable to the JTAG interface after the project is compiled successfully.
15) Click **Tools -> Programmer** to start the programming software.

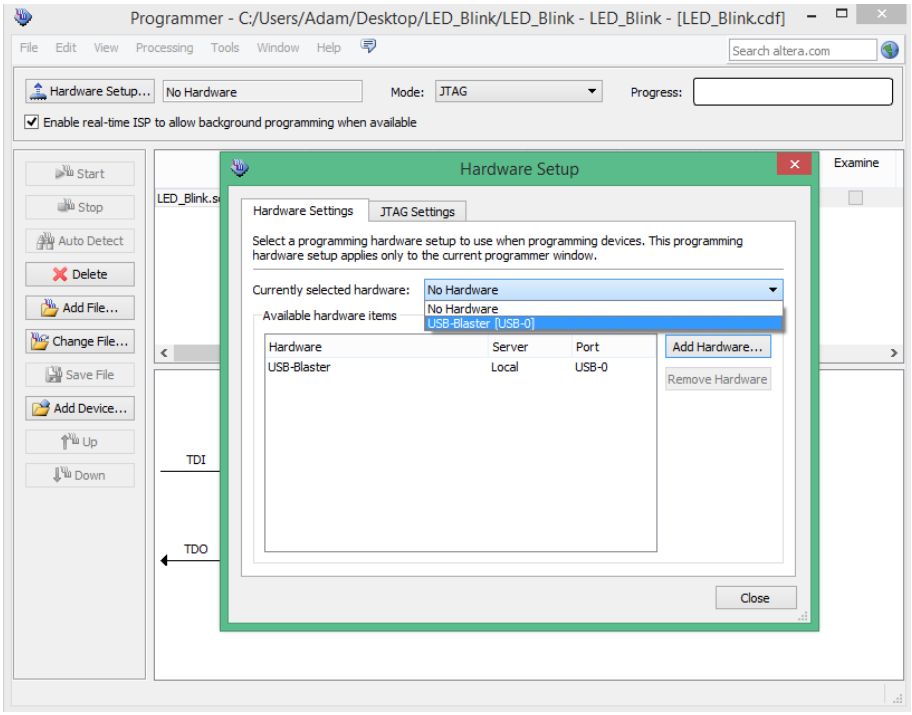


16) The programming software will automatically detect the available devices in the JTAG chain during startup and then list the result in the window.

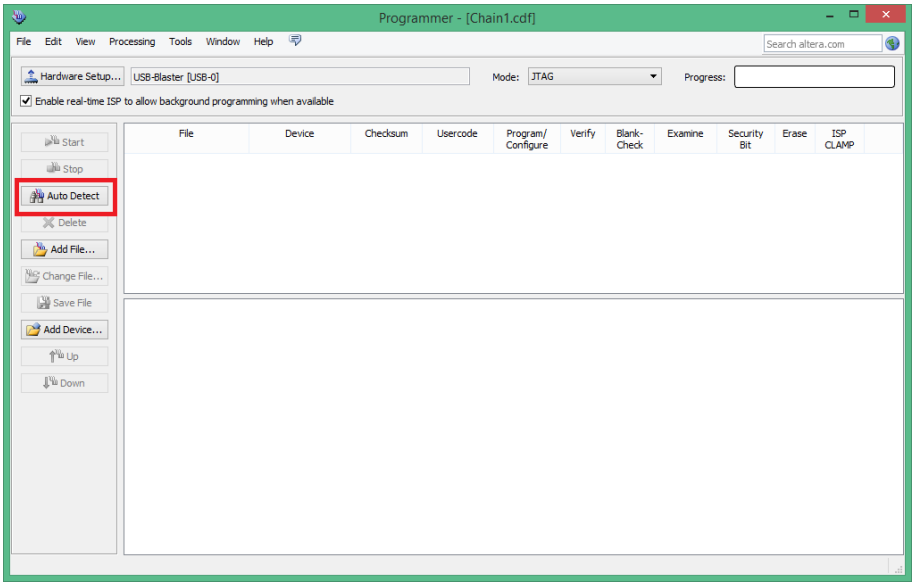


17) Sometimes you may encounter a problem that the USB download cable is not successfully detected, just as shown below. Click **Hardware Setup...** button, and select the **USB Blaster** in the dropdown list. Click **Close** button to exit.

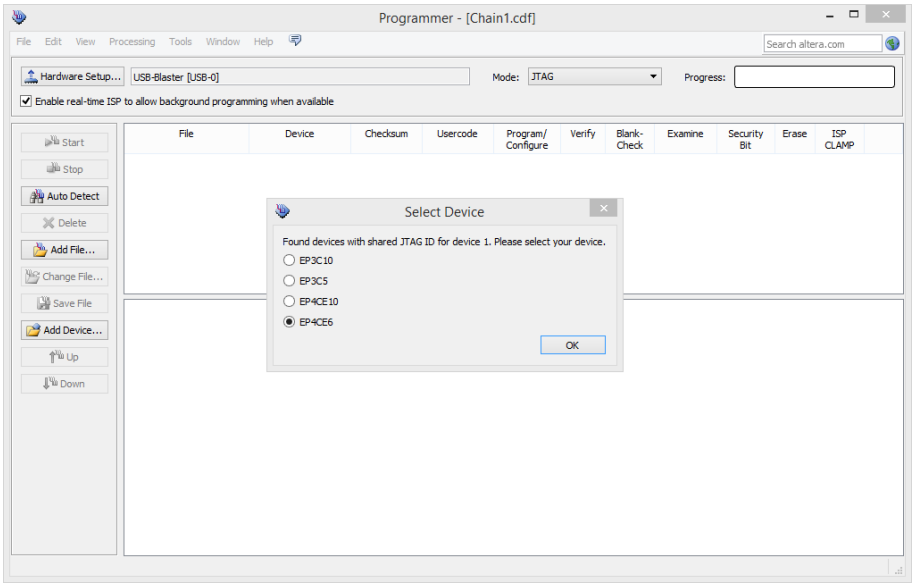




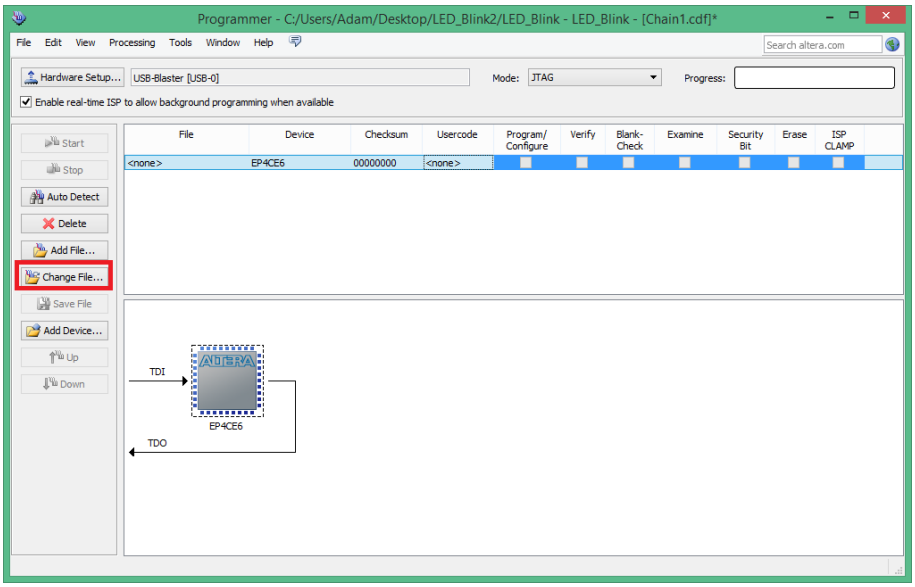
18) Click the **Auto Detect** button to start JTAG chain scan.

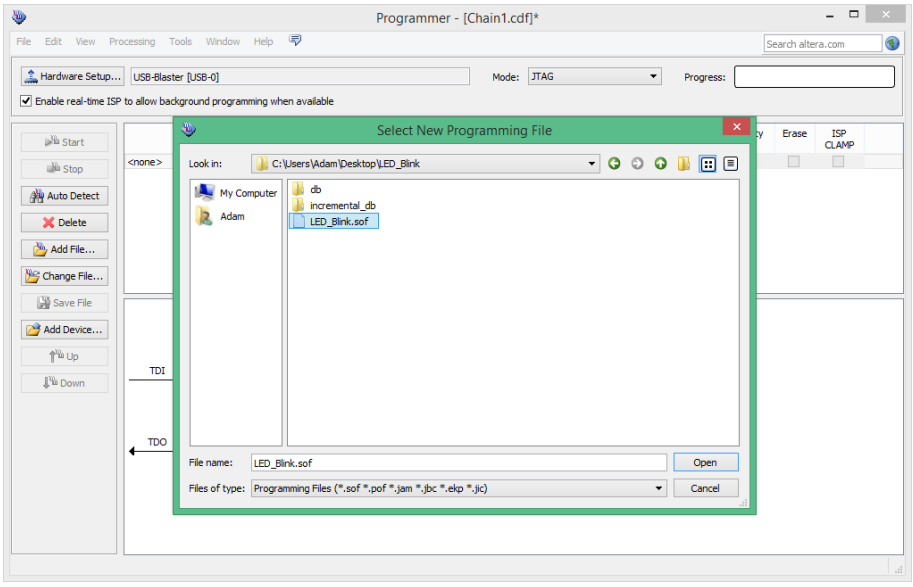


19) A tip window appears, select the device EP4CE6(The device name should match the one soldered on your board).

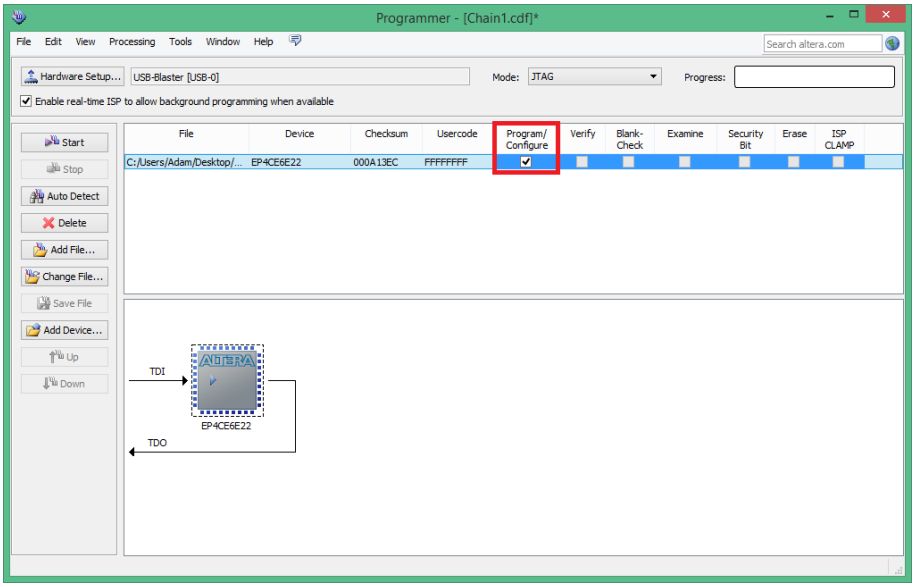


20) Click the **<none>** file to highlight it and then Click **Change File...** button, navigate to the **LED_Blink.sof** file and click **Open** button.

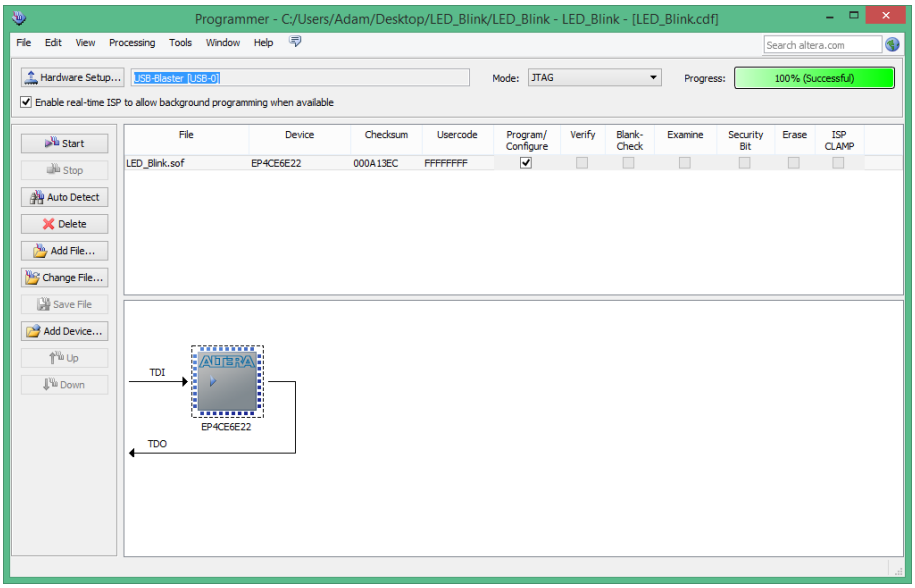




21) Enable the **Program/Configure** checkbox, and click **Start** button to program the device.

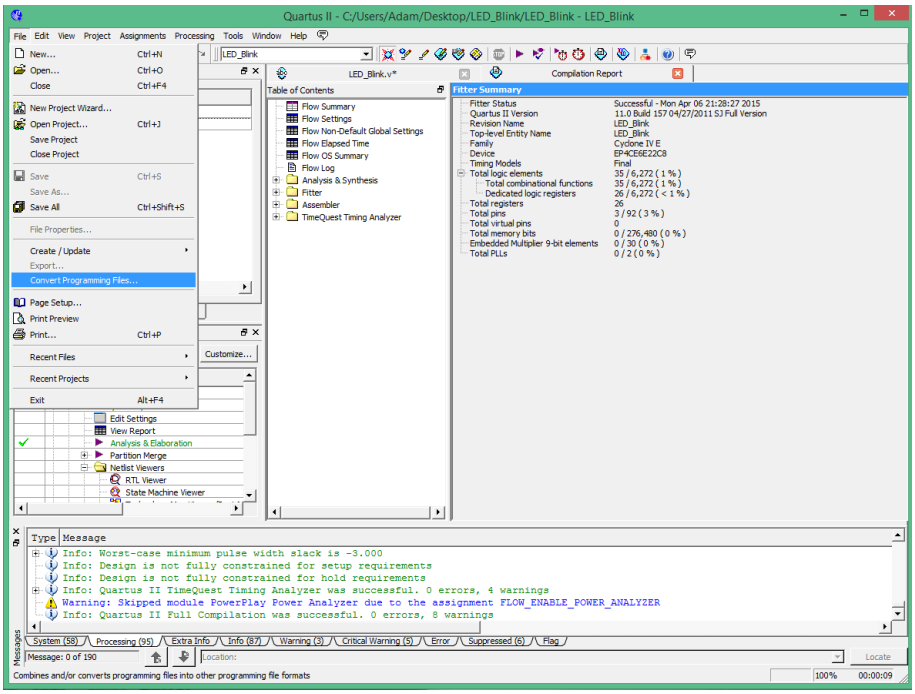


22) The status is displayed in the progress bar.



In the previous steps, the configuration file is downloaded to the internal RAM of FPGA device. Data will be lost after power off(Unlike the ROM device, power is required for RAM to keep data). In the following steps, we'll learn to program the nonvolatile serial configuration device EPCS4. Data stored in EPCS4 will be loaded into internal RAM of FPGA during power up or when CONFIG button is pressed. In this configuration scheme, the FPGA device works as a bridge/flash loader between JTAG download cable and configuration device.

23) Click **File -> Convert Programming Files...** in Quartus II main window.



24) Select the correct value for the properties as shown below.

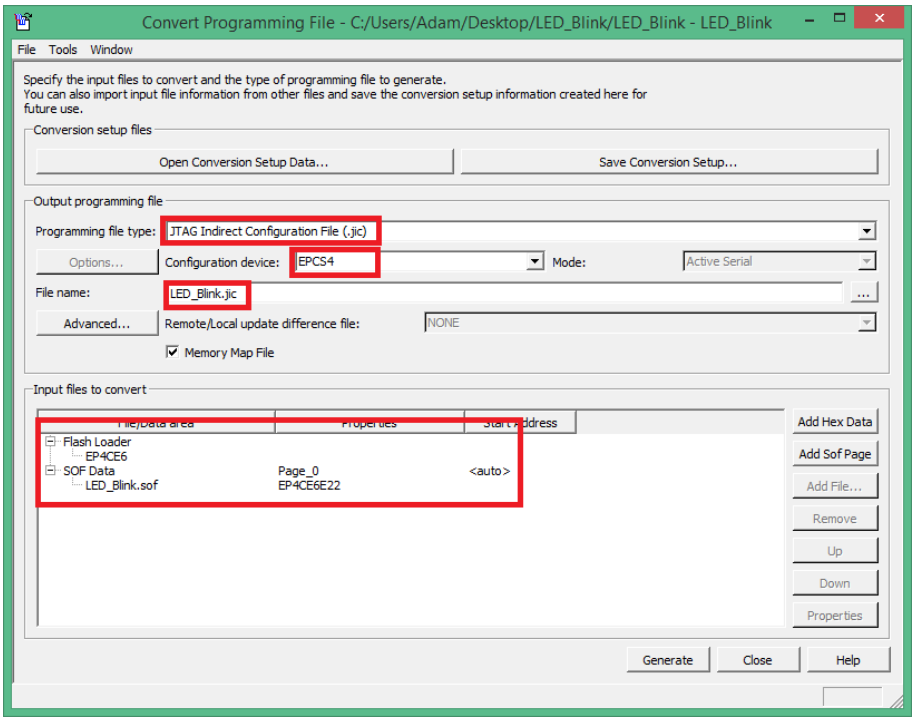
(1) Select **JTAG Indirect Configuration File(.jic)** as the programming file type.

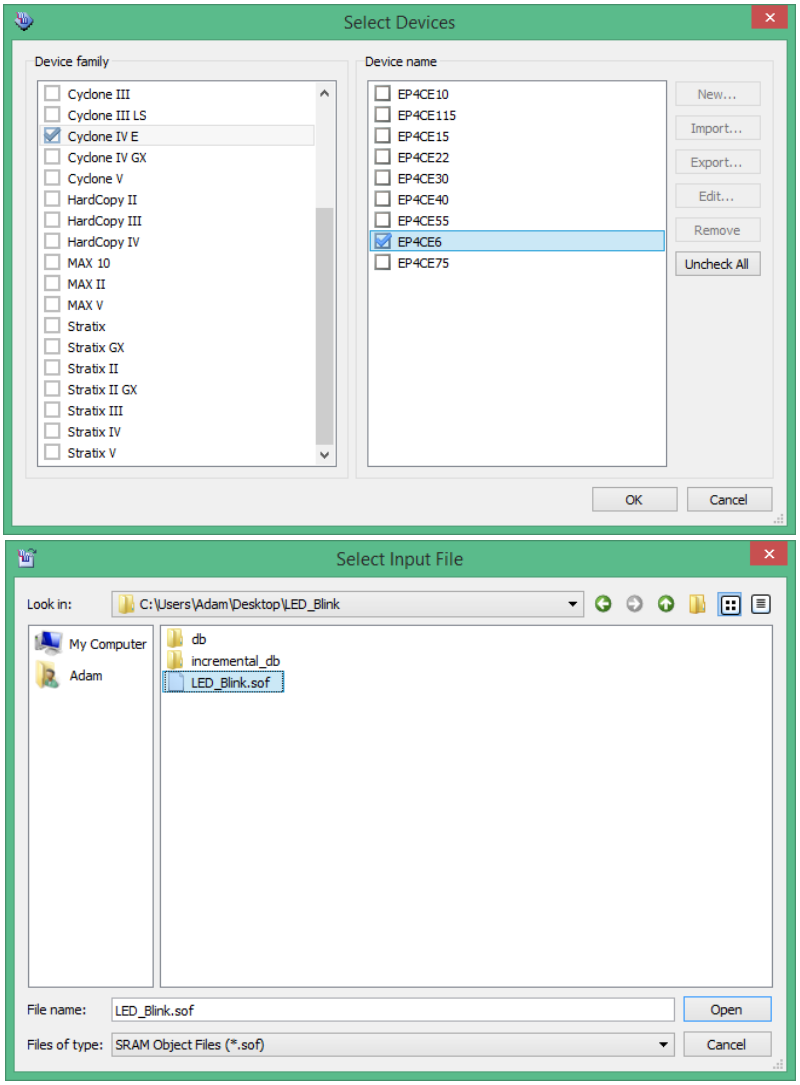
(2) Select **EPCS4** as the configuration device.

(3) Rename the output file as **LED_Blink.jic**.

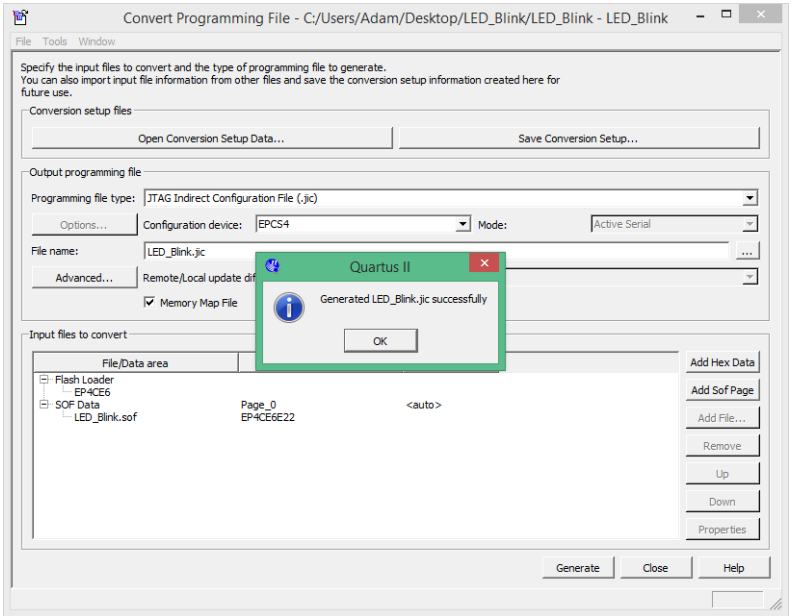
(4) Click **Flash Loader** and then click **Add Device...** button, select **Cyclone IV E** in device family and **EP4CE6** in device name. Click **OK** button to exit the window.

(5) Click **SOF Data**, and then click **Add File...** button, navigate to **LED_Blink.sof** file and click **Open** button.

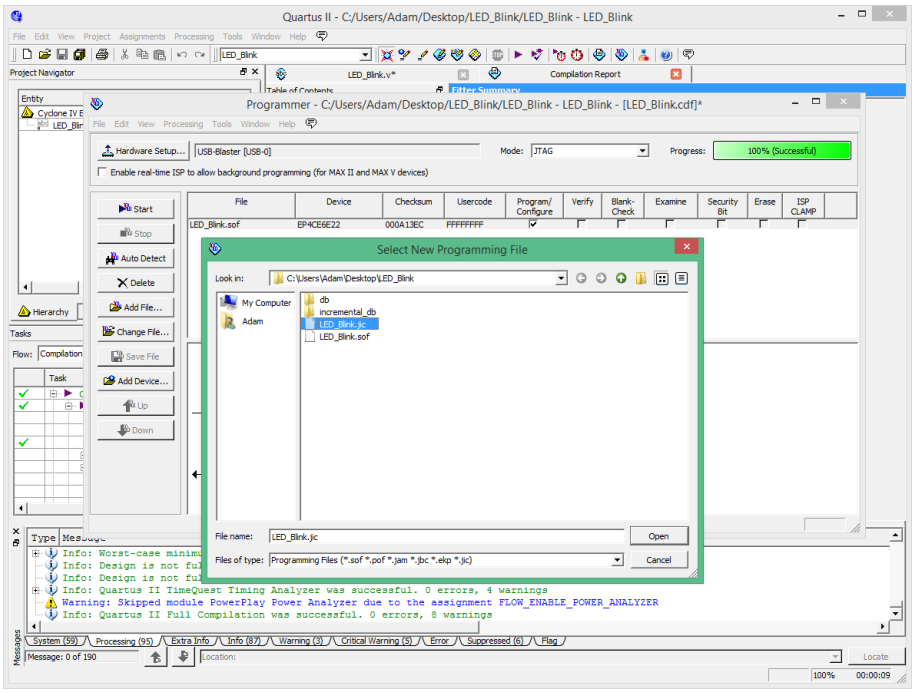




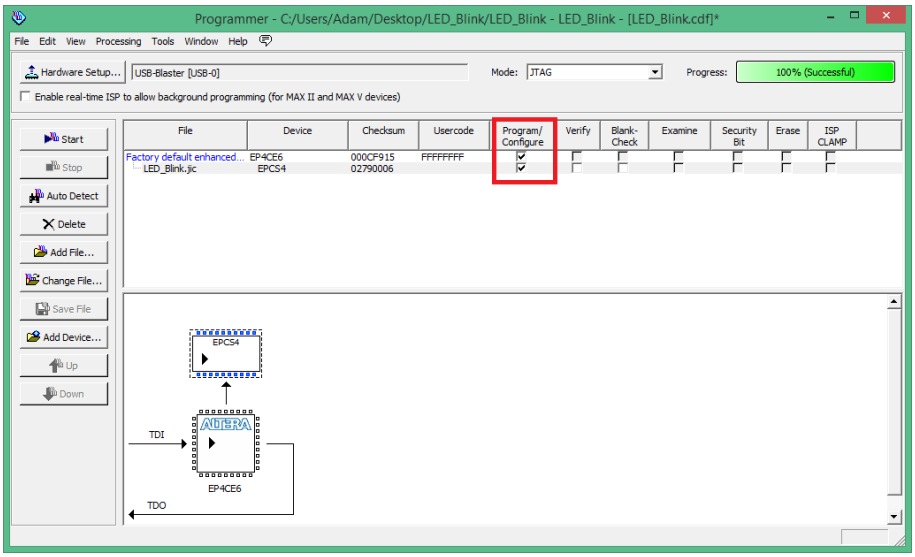
25) Click the **Generate** button located at the bottom-right of the window.



26) Go to **Programmer** window and click **Change File...** button, navigate to **LED_Blink.jic** file and click **Open** button.

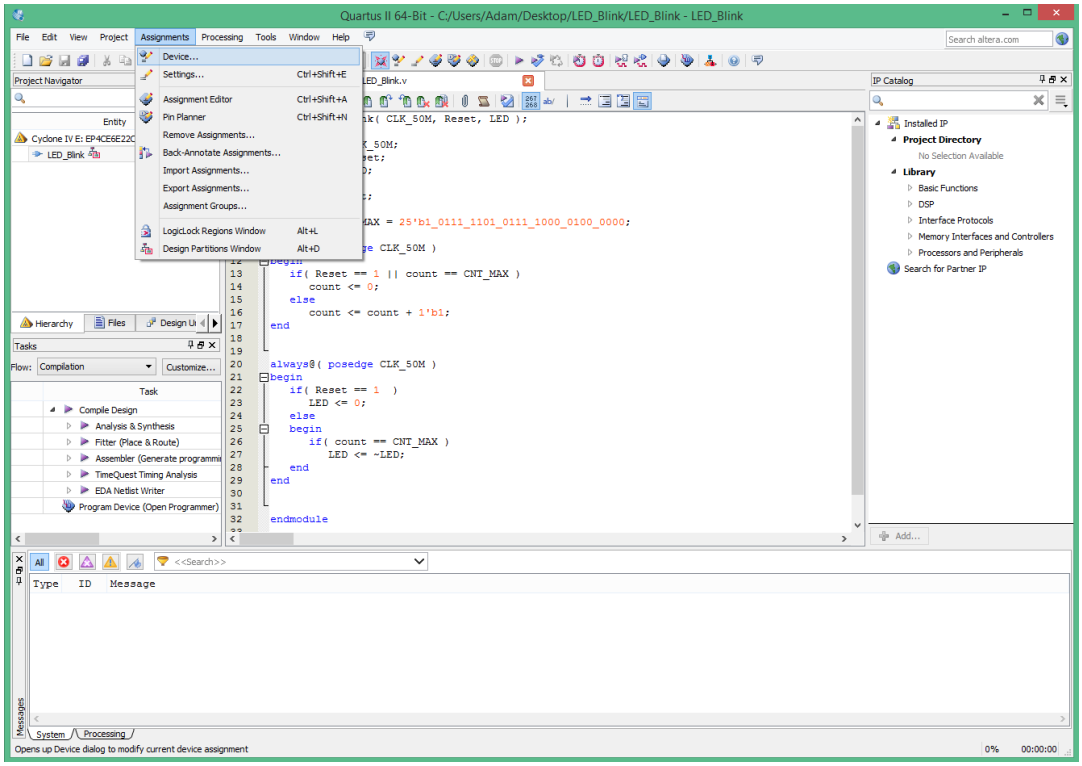


27) Enable the **Program/Configure** check box and click **Start** button to program the external serial flash EPCS4. After configuration, press the **CONFIG** button on the EP4CE6 Mini Board to force target device to reload the configuration from serial memory.

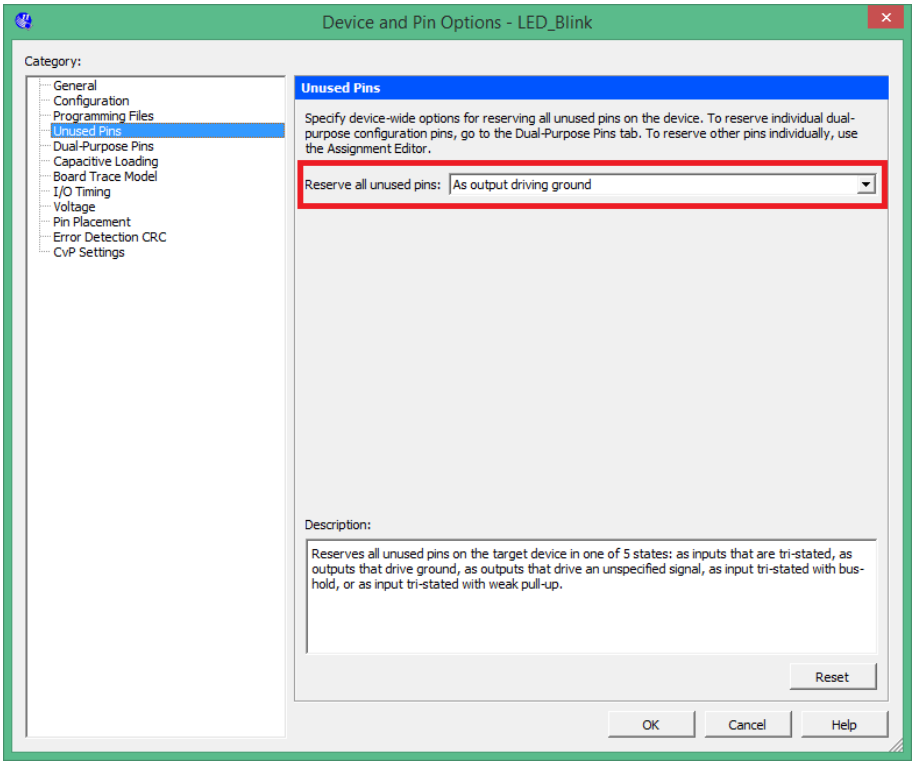


The EP4CE6 device has some dual-purpose pins. They could be used as general I/Os if the reserved functions are not needed. In addition, you could configure the state the of the unused I/O pins of a project. The steps are shown below.

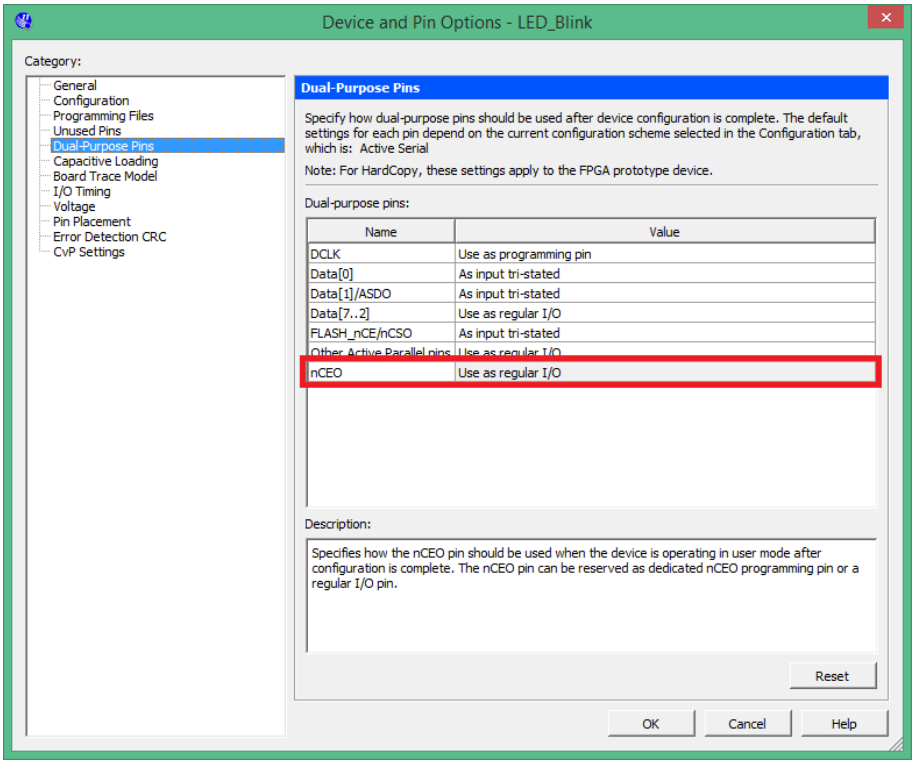
28) Go to **Assignments -> Device...**



29) You can configure the state of the unused pins under **Unused Pins** category.



30) Configure the dual-purpose pins under the **Dual-Purpose** Pins category.



Congratulations! You have learned the basic knowledge of how to use the Quartus II software to implement a FPGA design by yourself. Maybe the process is boring, but you'll be happy after finishing this guide step by step, because you'll be able to create more interesting projects easily. Thank you!

The End!

Revision History

REV.1	12/11 2013	Initial Release