



**FPGA**  
hackathon  
&  
conference

**25-27 November 2022**

**Conference**

(online)

**Hackathon**

(Kraków Technology Park + online)

[www.fpgahackathon.com](http://www.fpgahackathon.com)

**PATRONS**



Silesian University  
of Technology



**WAT** Military  
University  
of Technology

Powered by

**NOKIA**

*Instruction – using user PC*



## Table of Contents

1. Quartus installation .....	5
1.1. Links to download .....	5
1.2. Files to download .....	5
1.2.1. Multiple Download.....	5
1.2.2. Individual Files .....	6
1.3. Installation.....	7
2. Connection with board .....	12
2.1. Files to download .....	12
2.3. VirtualHere Client configuration .....	12
2.4. Enable ssh tunnel .....	13
2.5. Connection with USB device .....	14
2.6. Install JTAG Driver .....	15
3. Upload bitfile .....	20
3.1. Create empty project ( <i>for connection test</i> ).....	20
3.2. Import project .....	24
3.3. Compilation .....	25
3.4. Uploading bitfile .....	25
4. Additional actions.....	30
4.1. Files to download .....	30
4.2. UART Driver installation .....	30
5. Extras .....	32
5.1. Free Active-HDL Student Edition .....	32
5.2. Modelsim in quartus: .....	33
5.2.1. Linking modelsim to quartus.....	33
5.2.2. Adding testbench to project.....	34
5.2.3 Running simulation.....	34



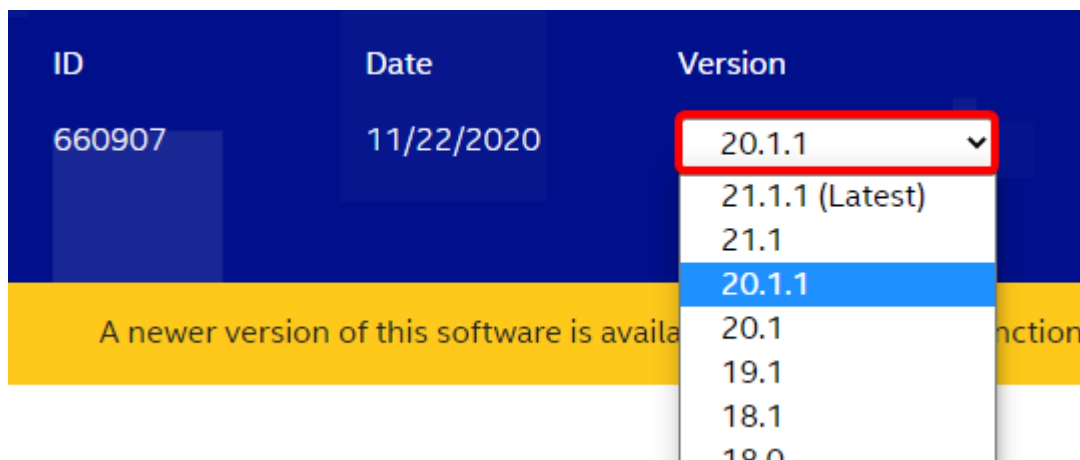
## 1. Quartus installation

### 1.1. Links to download

- Quartus ([windows link](#)) ([linux link](#))

### 1.2. Files to download

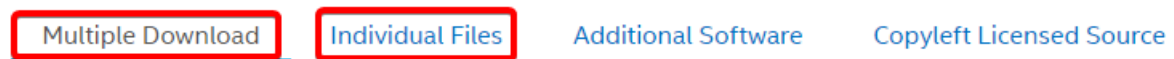
Select version of quartus you want:



You can use latest version, but we prefer version 20.1.1 – it is last version with ModelSim inside.

There are 2 options of downloading installation files: “Multiple Download” and “Individual Files”:

#### Downloads



“Multiple Download” is full quartus installation pack and “Individual Files” are separate files which can be selected if needed. “Individual Files” also takes less disk space.

#### 1.2.1. Multiple Download

All You need to download is single tar:

Intel® Quartus® Prime Lite Edition Software (Device support included)

Download  
Quartus-lite-20.1.1.720-windows.tar

Size: 5.9 GB  
SHA1: f1bec3a3bf03e7ab9106af5fac93475347b66e1e

When this file is downloaded you need to unpack and run “setup.bat” file.

### 1.2.2. Individual Files

Files required to download:

ModelSim-Intel® FPGA Edition (includes Starter Edition)

Download  
ModelSimSetup-20.1.1.720-windows.exe

Size: 1.2 GB  
SHA1: d484e4c7882fca584a9b0243cbdd74953a4aeb25

Intel® Quartus® Prime (includes Nios® II EDS)

Download  
QuartusLiteSetup-20.1.1.720-windows.exe

Size: 1.6 GB  
SHA1: 5edd76fcafa2a6a40077bc3eeed5bdc95cacdc8

Intel® Cyclone® V Device Support

Download  
cyclonev-20.1.1.720.qdz

Size: 1.3 GB  
SHA1: 95709f12822c4efed0f70516b1588d43e69ae85b

After download place all files in same directory and run QuartusLiteSetup-20.1.1.720-windows.exe on windows or QuartusLiteSetup-20.1.1.720-linux.run on linux.

If using Linux there is need to have proper access rights, therefore it is required to run chmod u+x on all downloaded files:

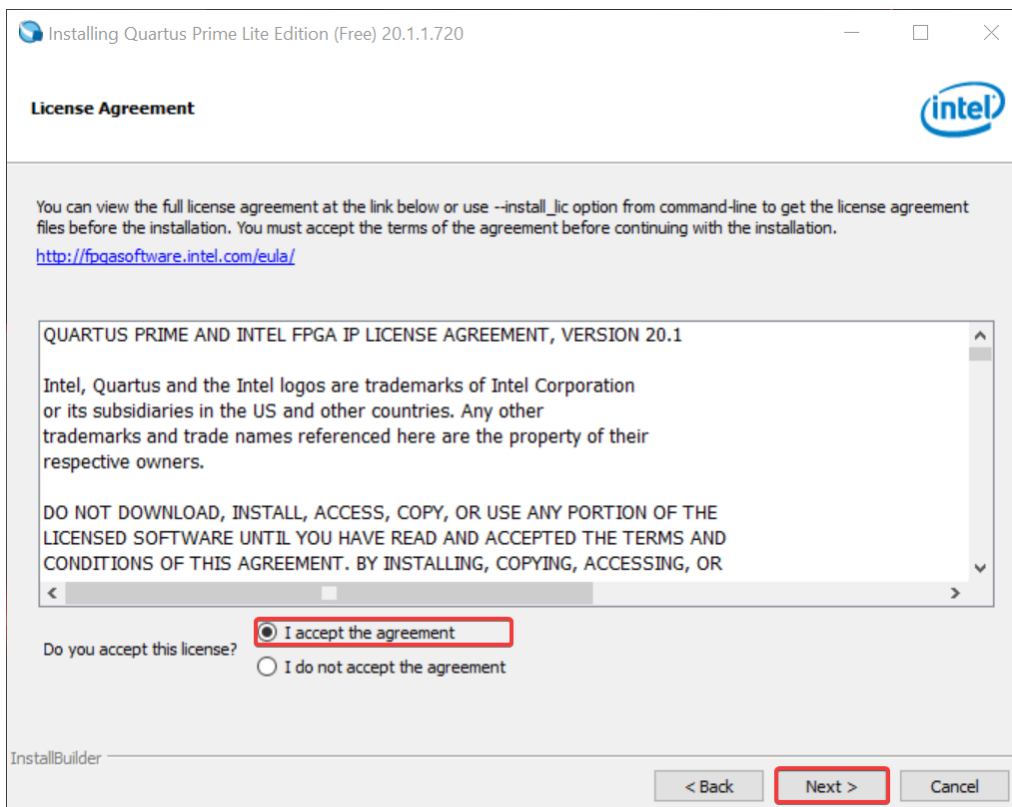
```
chmod u+x QuartusLiteSetup-20.1.1.720-linux.run
chmod u+x ModelSimSetup-20.1.1.720-linux.run
chmod u+x cyclonev-20.1.1.720.qdz
```

### 1.3. Installation

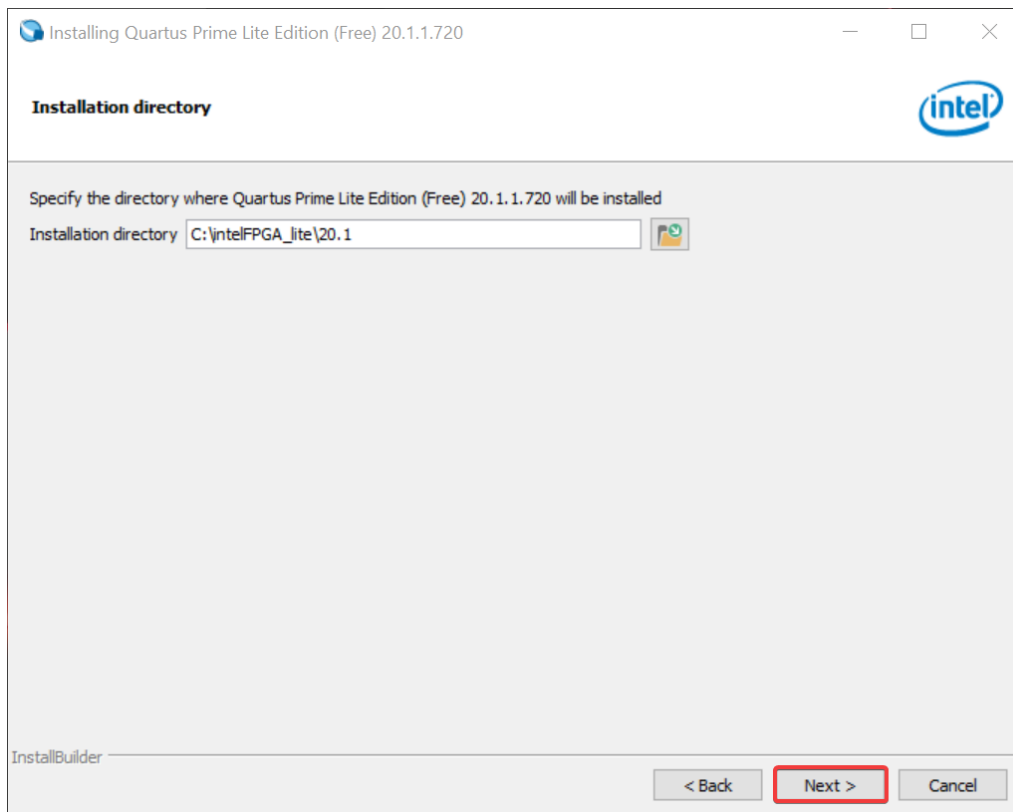
Installation process looks the same on windows and linux. Select “next”



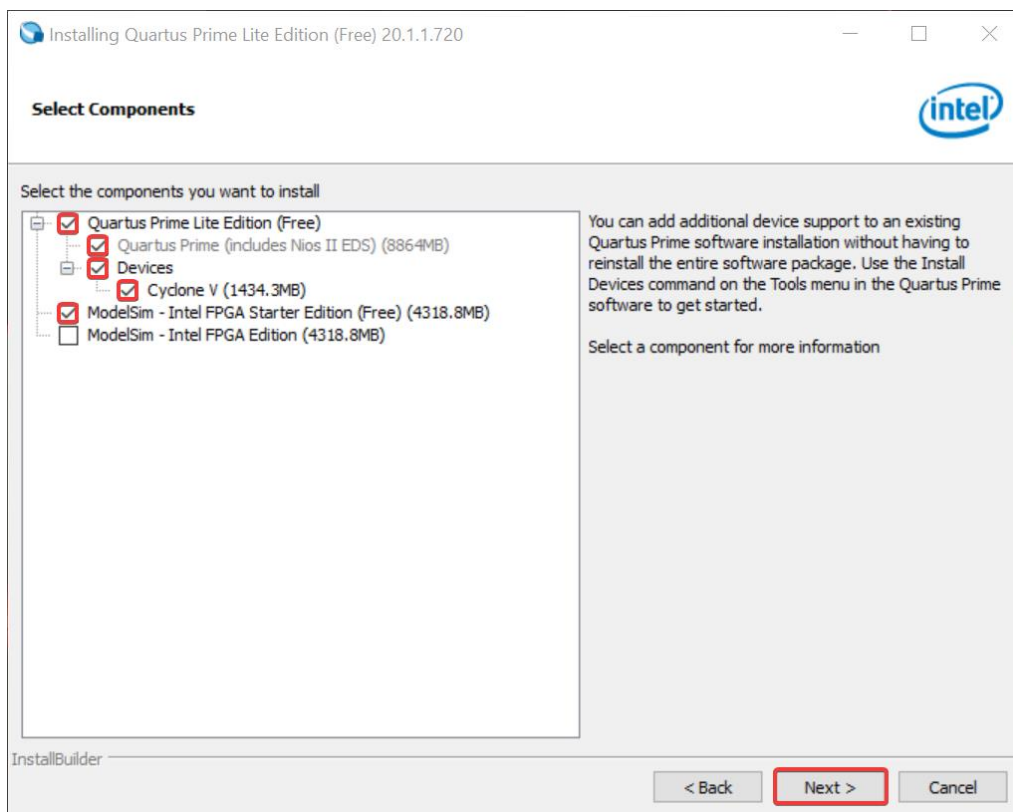
Accept the Agreement



## Choose path of installation

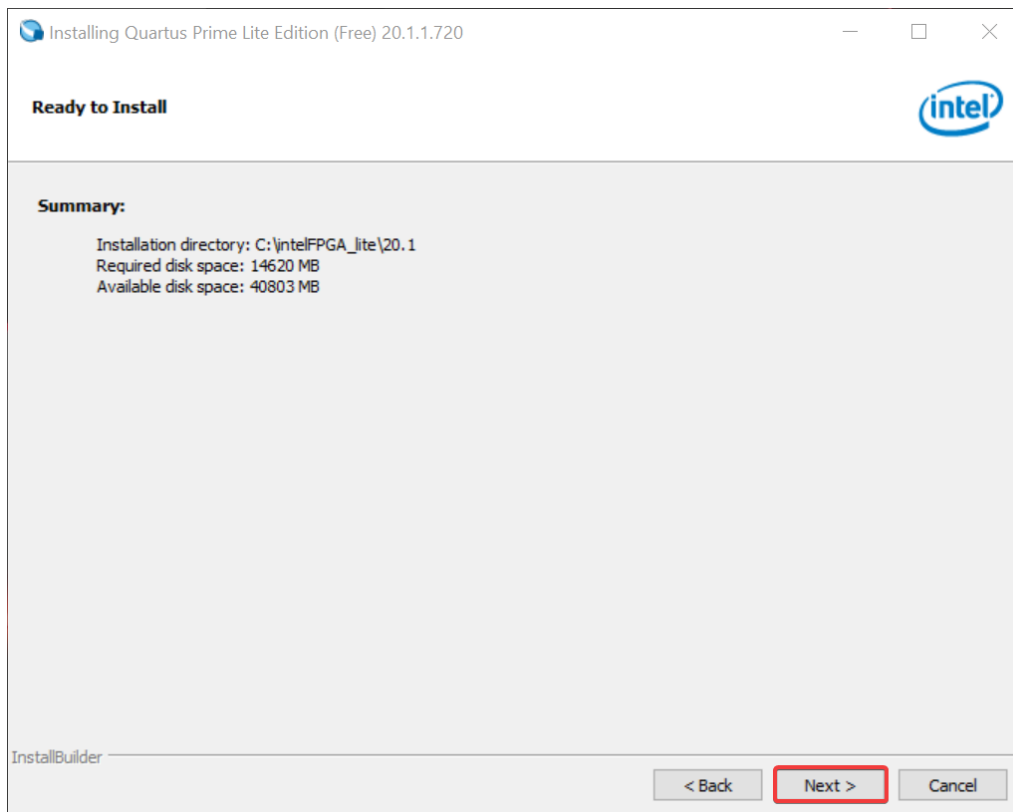


## Select all highlighted components

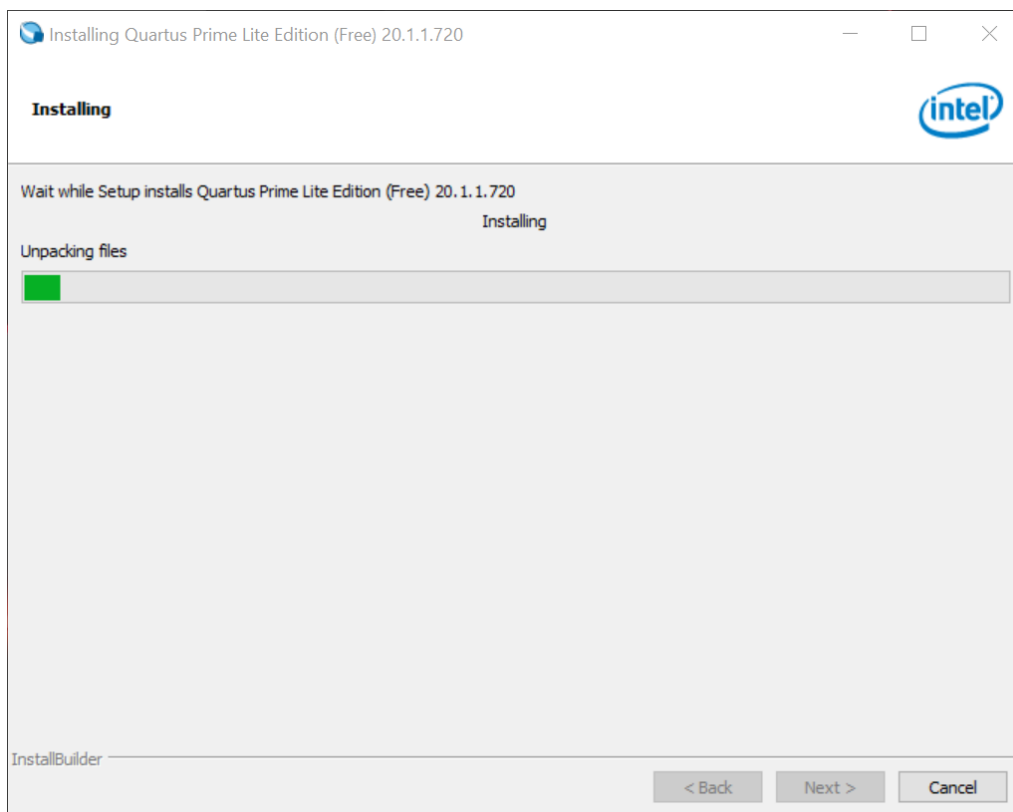




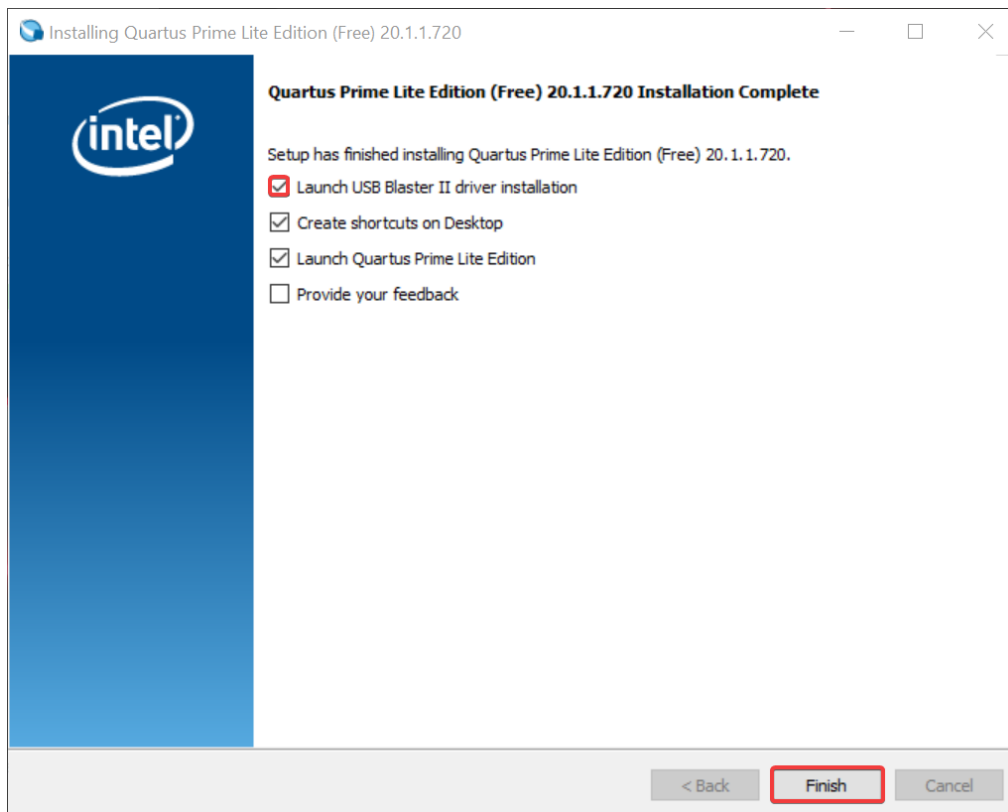
## Start installation



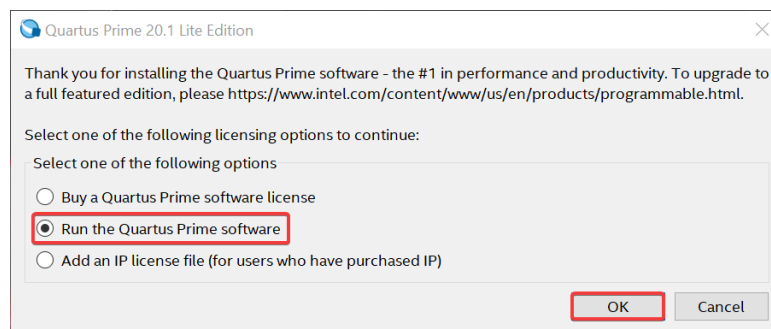
## Wait until the installation is complete



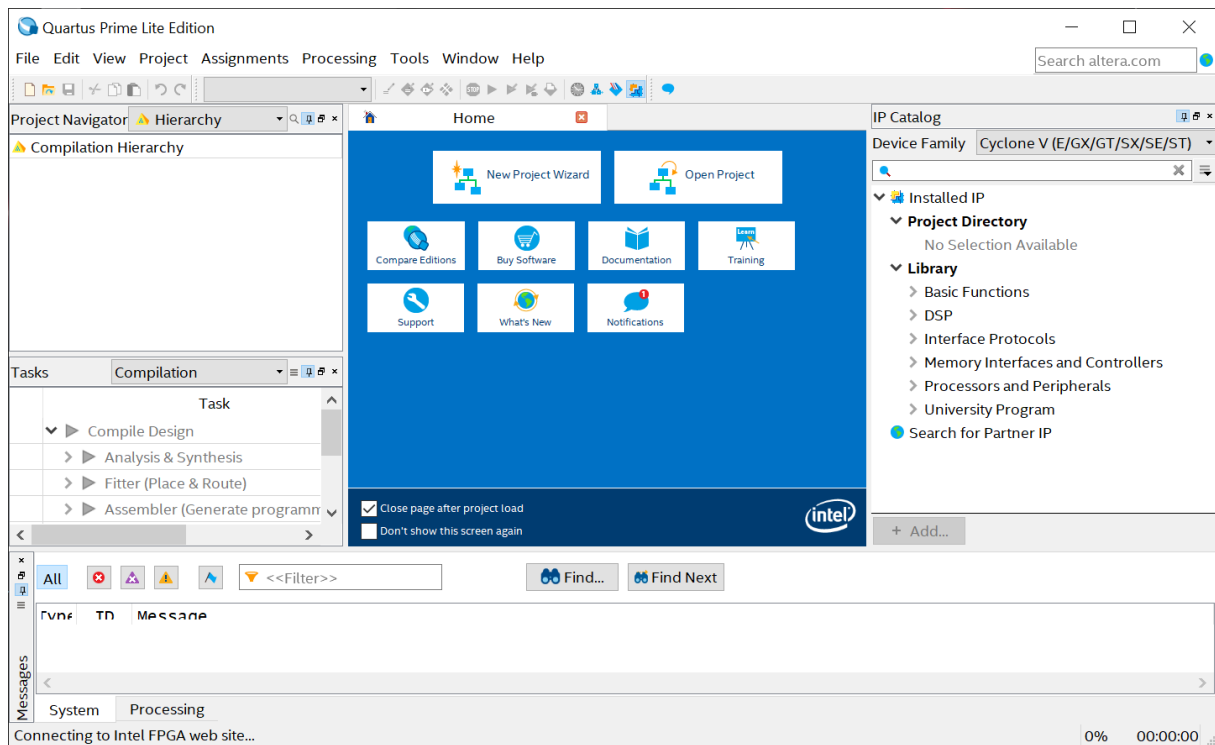
After installation select highlighted option (if it appears) and finish installation



Run quartus. While first run select highlighted option



## Quartus is running



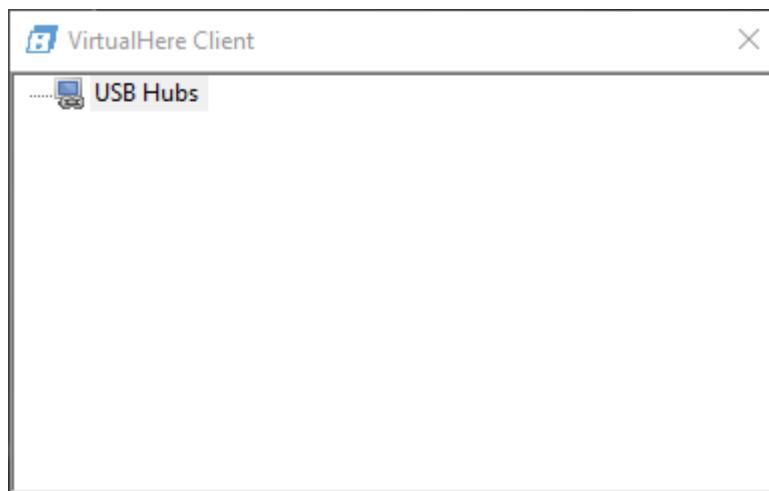
## 2. Connection with board

### 2.1. Files to download

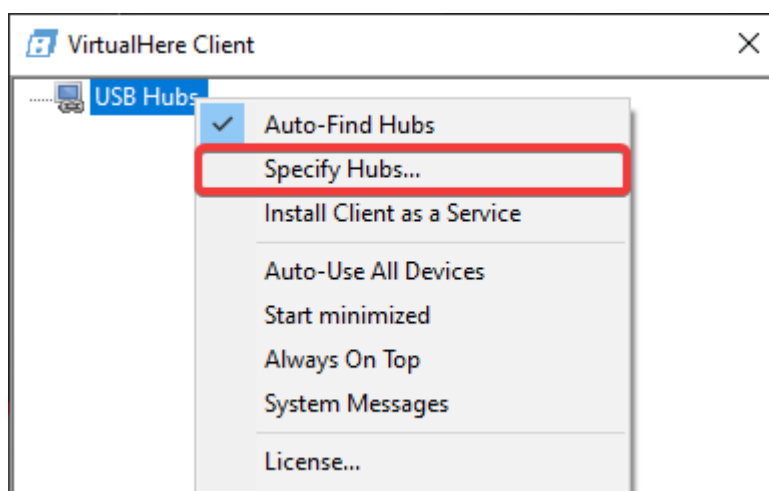
- VirtualHere Client [\(link\)](#)
- JTAG driver (Windows only) [\(link\)](#)

### 2.3. VirtualHere Client configuration

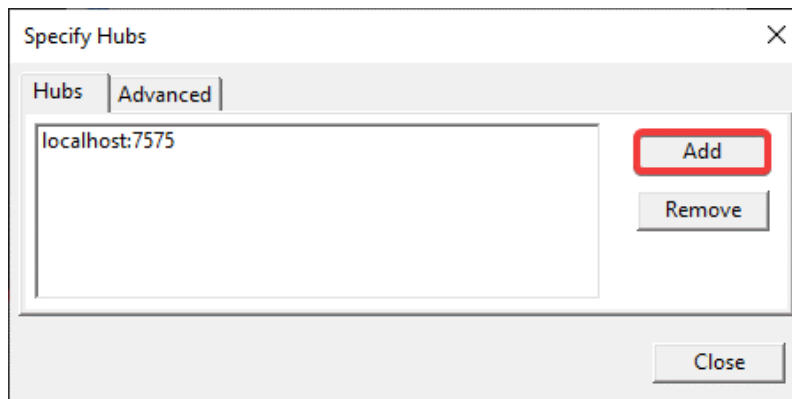
Open VirtualHere Client



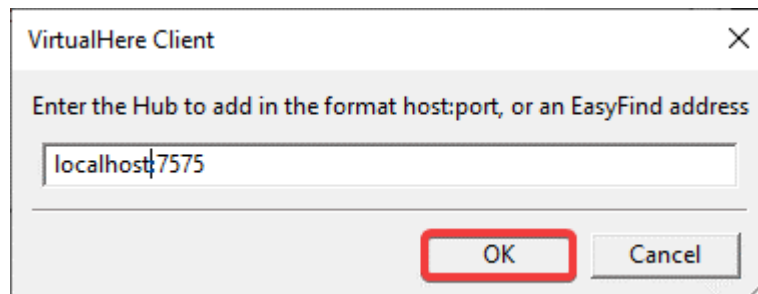
Click right mouse button on USB Hubs, than select "Specify Hubs" option



Select “Add” option



And type “localhost:7575”



After “OK” VirtualHere Client is configured. To see all devices there is need to create tunnel

## 2.4. Enable ssh tunnel

Open command prompt (on Windows) or terminal (on Linux), change directory to directory with ssh key and run command:

```
ssh -N -L 7575:localhost:7575 hackaton_user@31.172.183.106 -i <SSH_KEY_PATH>
```

SSH\_KEY\_PATH is path to ssk key received in the email, for example “SSH\_Key\_Team\_A1”. Path can be relative or absolute.

Console must be open all time while using connected usb devices.

\*If there is problem with key permissions on Windows - open cmd and use commands:

```
icacls <SSH_KEY_PATH> /inheritance:r
icacls <SSH_KEY_PATH> /grant:r "%username%": "(R)"
```

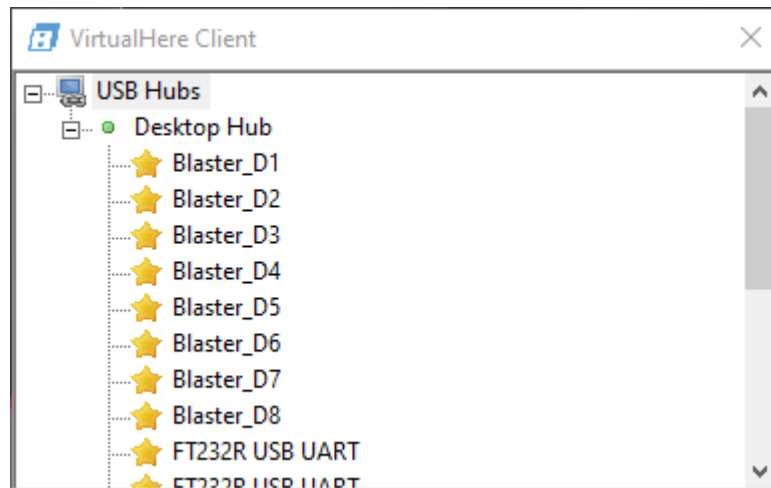
\*If there is problem with key permissions on Linux - open terminal and use command:

```
chmod 600 <SSH_KEY_PATH>
```

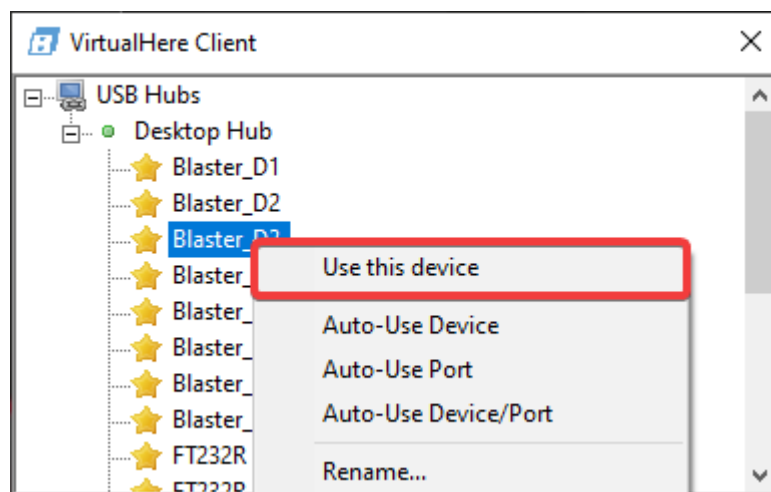
## 2.5. Connection with USB device

To obtain list of USB devices there is needed to open ssh tunnel.

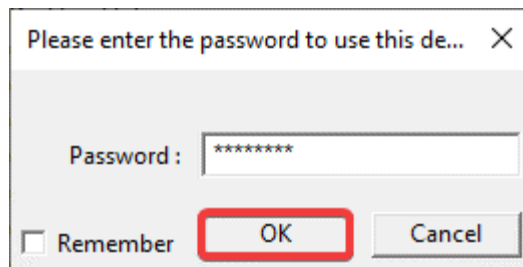
With configured VirtualHere Client and ssh tunnel opened Client automatically finds all shared USB devices



To connect use right mouse button on selected device and choose "Use this device option"

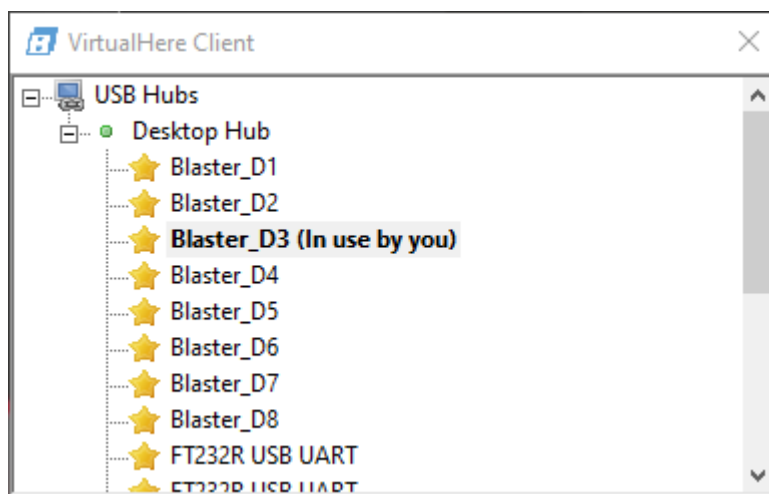


Enter the password which is 12 first characters of the 5<sup>th</sup> line in the team ssh private key file.



It is possible to use option “Remember”. If the remembered password was entered incorrectly, it can be canceled by selecting device and using shortcut Shift+F9.

If password for device is correct connection will be enabled



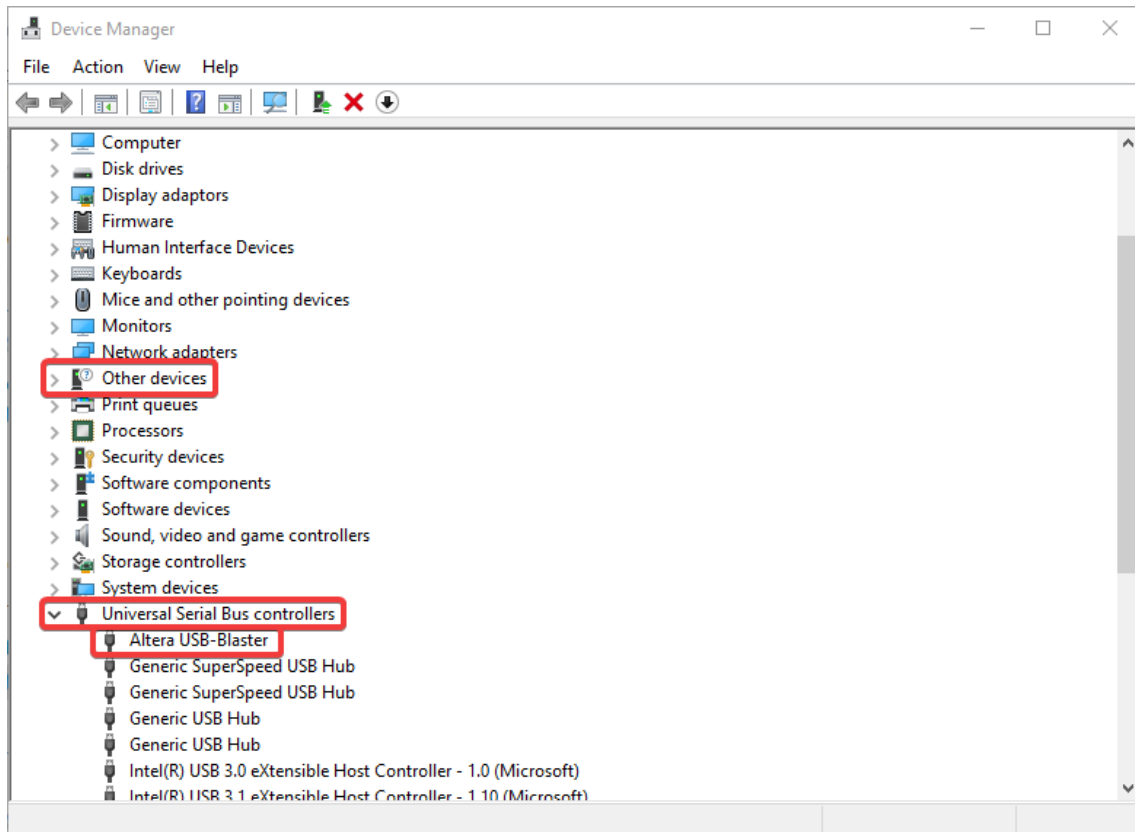
It is not possible to connect with usb device which is in being used by other user.

## 2.6. Install JTAG Driver

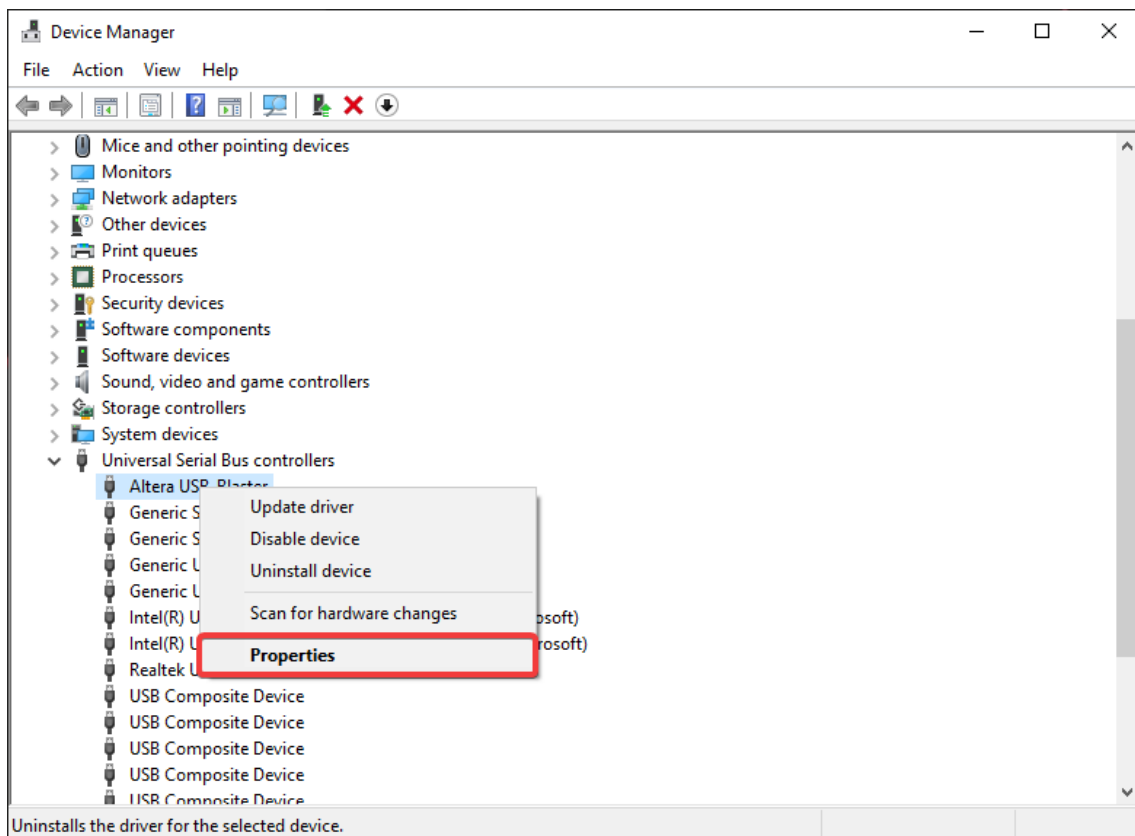
This part is needed only for windows users.

Unpack JTAG driver (Usb\_blaster\_q16.1.zip) downloaded from link in chapter 2.1, and open Device Manager from Control Panel.

Expand list of “Other devices” or “Universal Serial Bus controllers” and find “Altera USB-Blaster” or “USB-Blaster” device.

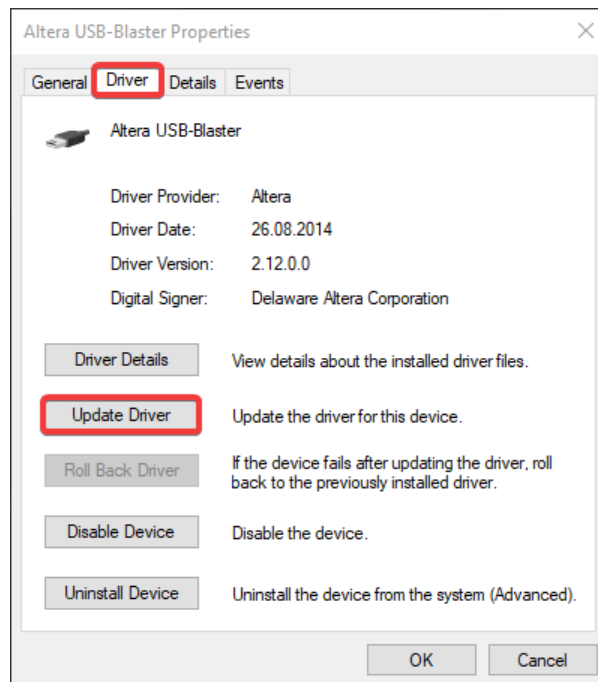


Click right mouse button on “Altera USB-Blaster” and select Properties

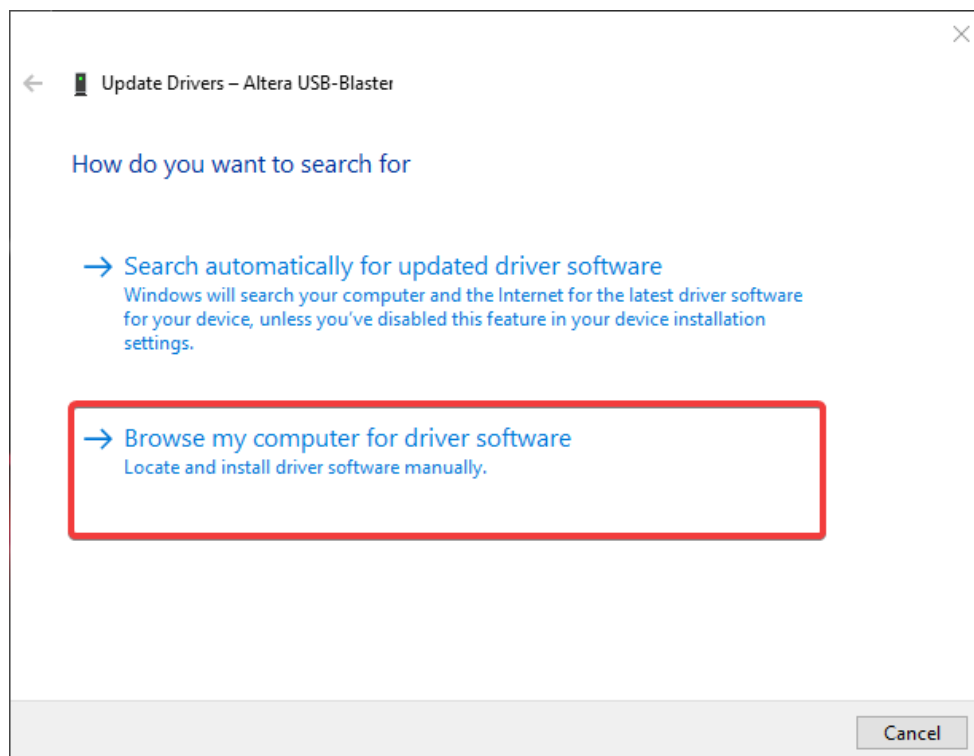




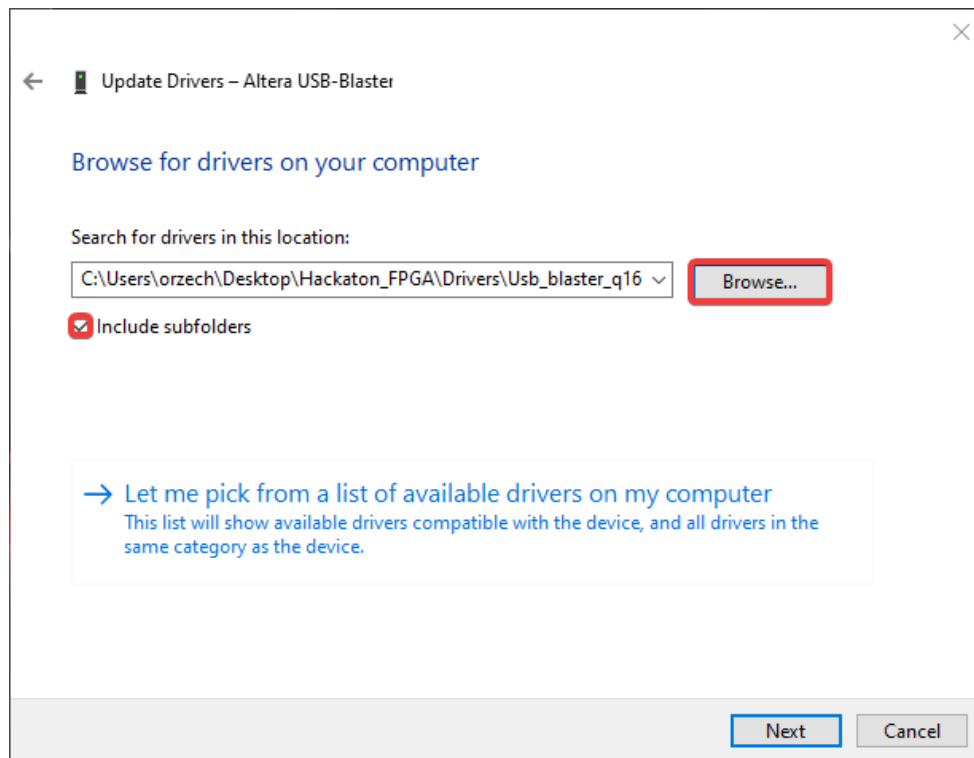
Choose “Driver” from top bar and select “Update Driver”



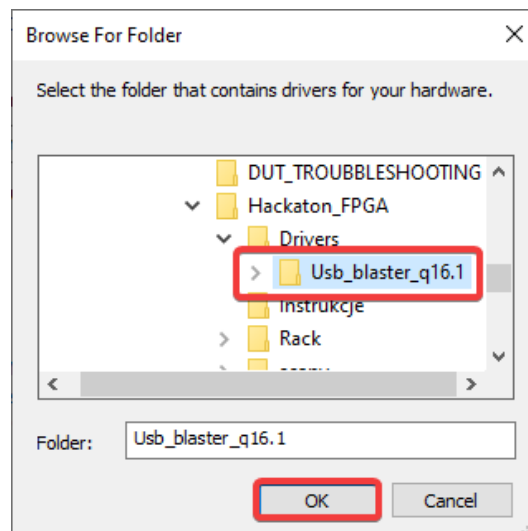
Select “Browse my computer for driver software”



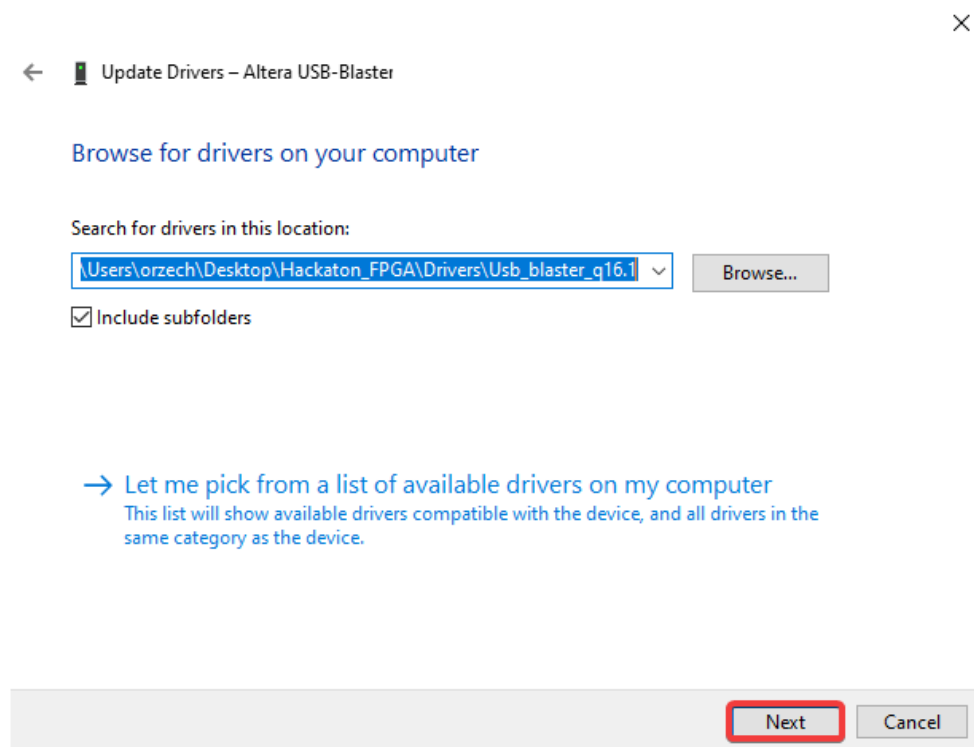
Mark "Include subfolders" and select "Browse" button



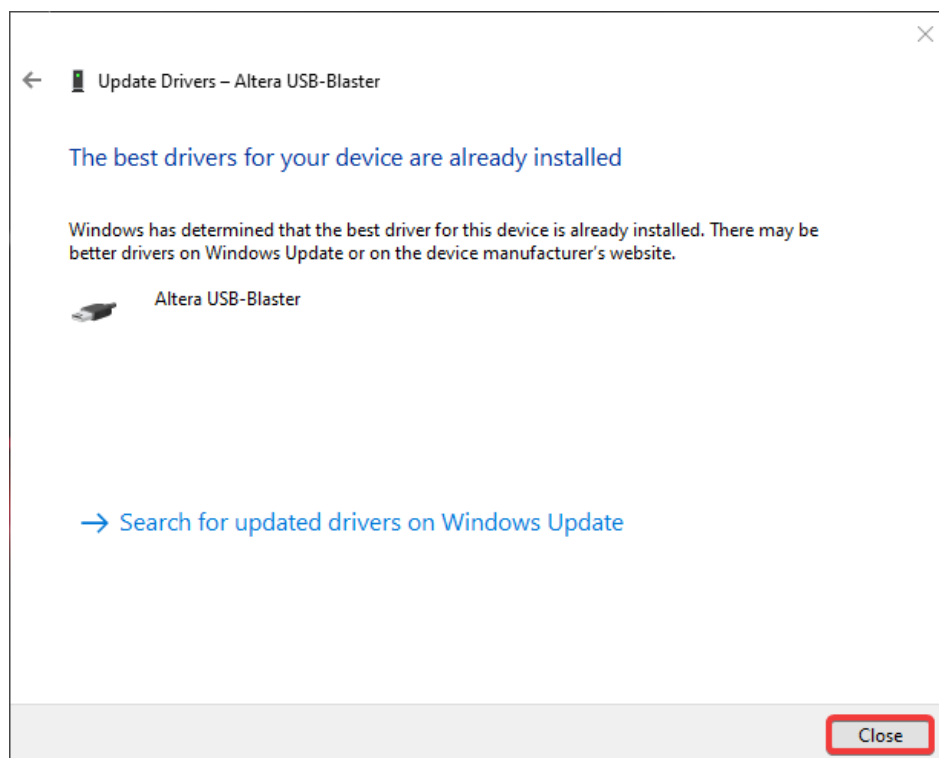
Choose path to unpacked driver. Select Usb\_blaster\_q16.1 directory (not x32 or x64 inside)



Select "Next"



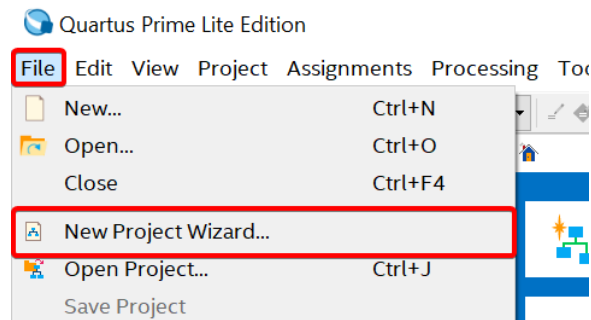
Drivers will be installed



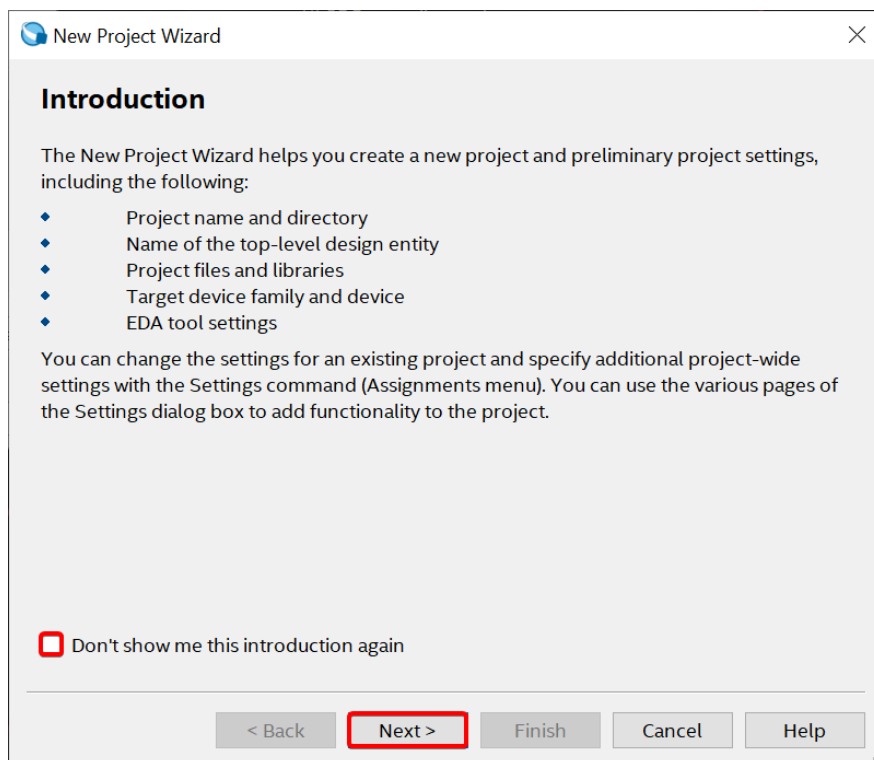
### 3. Upload bitfile

#### 3.1. Create empty project (*for connection test*)

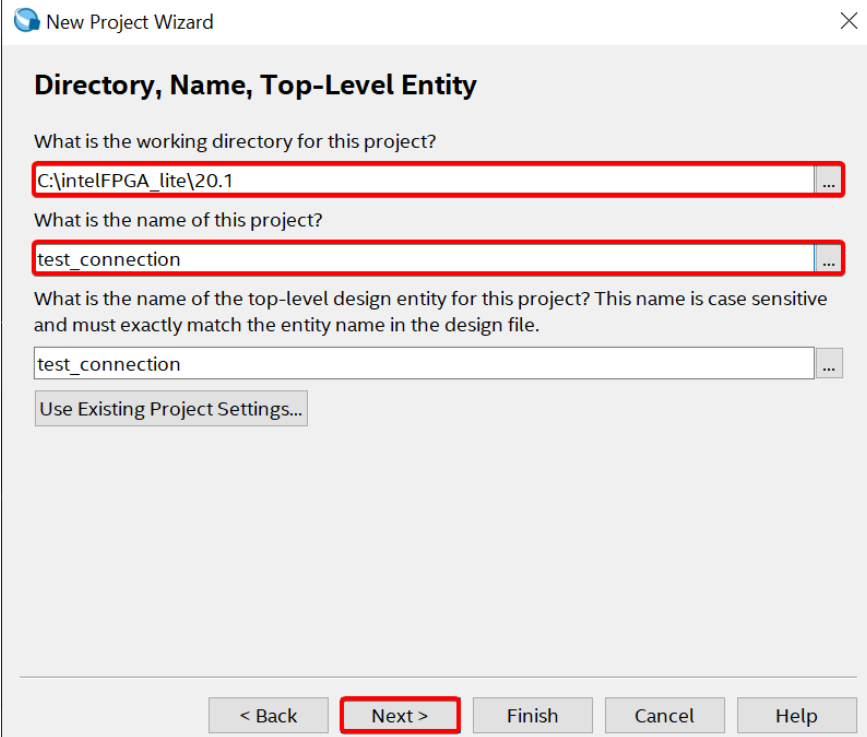
Select “New Project Wizard...” from “File” tab



Select “Next” and check “Don’t show me this introduction again” if you don’t want to see this Introduction again

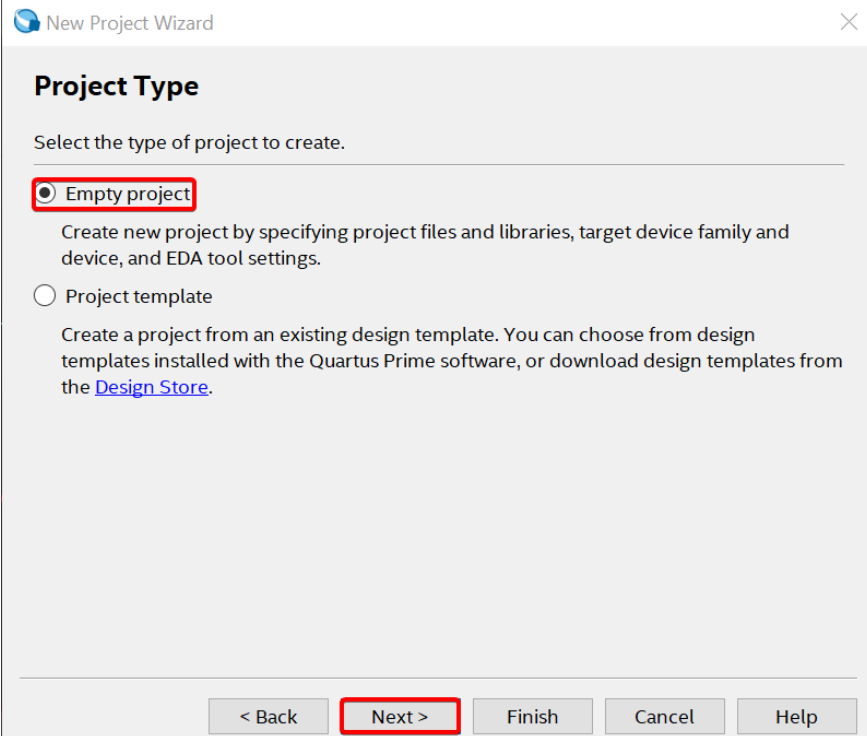


Enter correct project path and name

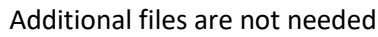


The screenshot shows the 'New Project Wizard' dialog box, specifically the 'Directory, Name, Top-Level Entity' step. The title bar reads 'New Project Wizard'. The main heading is 'Directory, Name, Top-Level Entity'. There are three text input fields, each with a red rectangular highlight around it. The first field is labeled 'What is the working directory for this project?' and contains the text 'C:\intelFPGA\_lite\20.1'. The second field is labeled 'What is the name of this project?' and contains the text 'test\_connection'. The third field is labeled 'What is the name of the top-level design entity for this project? This name is case sensitive and must exactly match the entity name in the design file.' and also contains the text 'test\_connection'. Below these fields is a button labeled 'Use Existing Project Settings...'. At the bottom of the dialog, there are five buttons: '< Back', 'Next >', 'Finish', 'Cancel', and 'Help'. The 'Next >' button is highlighted with a red rectangular box.

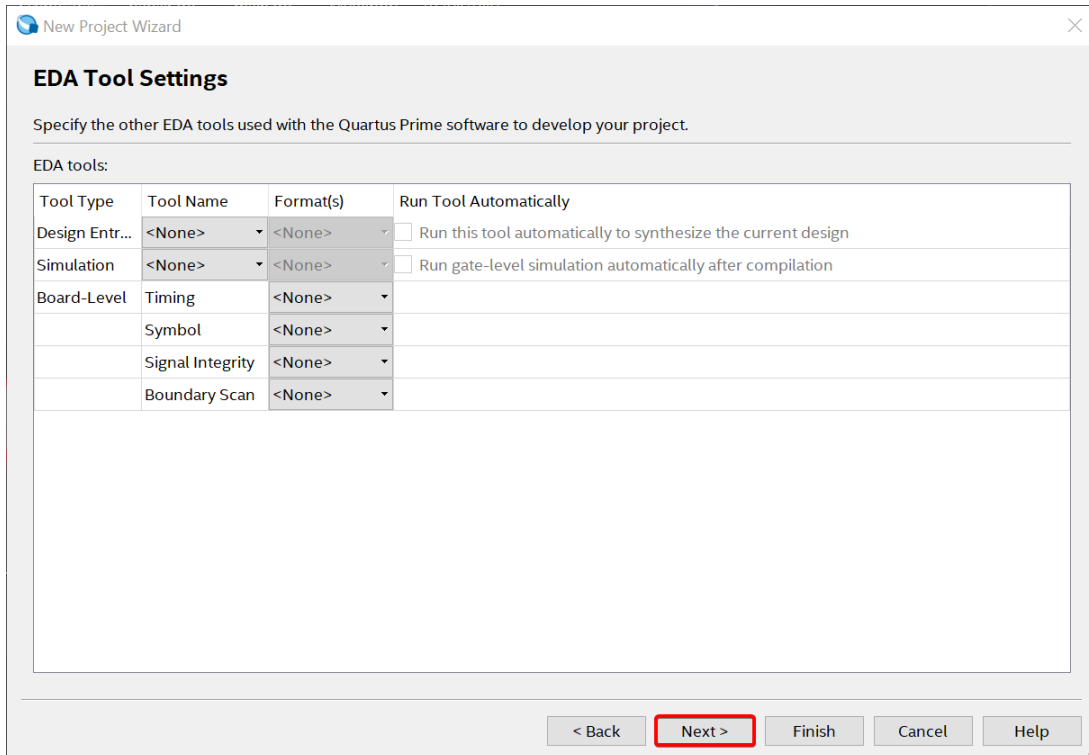
Select "Empty project"



The screenshot shows the 'New Project Wizard' dialog box, specifically the 'Project Type' step. The title bar reads 'New Project Wizard'. The main heading is 'Project Type'. Below the heading is the instruction 'Select the type of project to create.' There are two radio button options. The first option is 'Empty project', which is selected (indicated by a filled radio button) and has a red rectangular highlight around the text. Below this option is a description: 'Create new project by specifying project files and libraries, target device family and device, and EDA tool settings.' The second option is 'Project template', which is not selected (indicated by an empty radio button). Below this option is a description: 'Create a project from an existing design template. You can choose from design templates installed with the Quartus Prime software, or download design templates from the [Design Store](#).' At the bottom of the dialog, there are five buttons: '< Back', 'Next >', 'Finish', 'Cancel', and 'Help'. The 'Next >' button is highlighted with a red rectangular box.



For empty project nothing is needed here



The screenshot shows the 'EDA Tool Settings' window of the New Project Wizard. It contains a table for configuring EDA tools and a set of navigation buttons at the bottom.

**EDA Tool Settings**

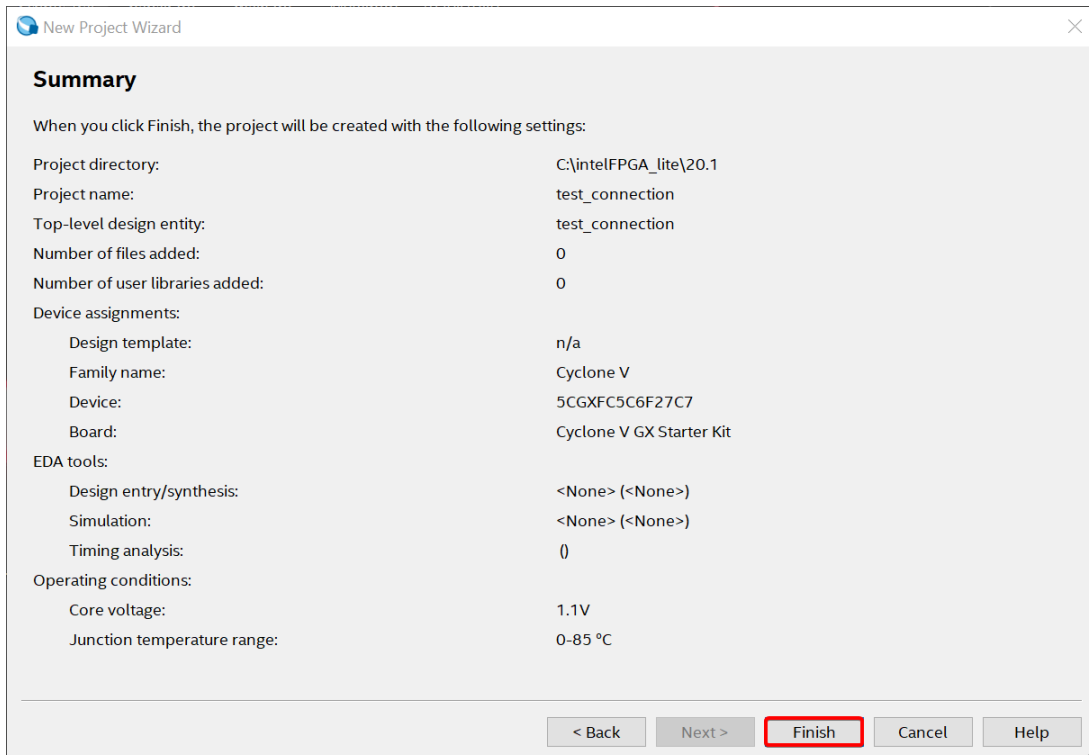
Specify the other EDA tools used with the Quartus Prime software to develop your project.

EDA tools:

Tool Type	Tool Name	Format(s)	Run Tool Automatically
Design Entr...	<None>	<None>	<input type="checkbox"/> Run this tool automatically to synthesize the current design
Simulation	<None>	<None>	<input type="checkbox"/> Run gate-level simulation automatically after compilation
Board-Level	Timing	<None>	
	Symbol	<None>	
	Signal Integrity	<None>	
	Boundary Scan	<None>	

Navigation buttons: < Back, **Next >**, Finish, Cancel, Help

Select "Finish" to create empty project



The screenshot shows the 'Summary' window of the New Project Wizard, displaying the configuration details for the project.

**Summary**

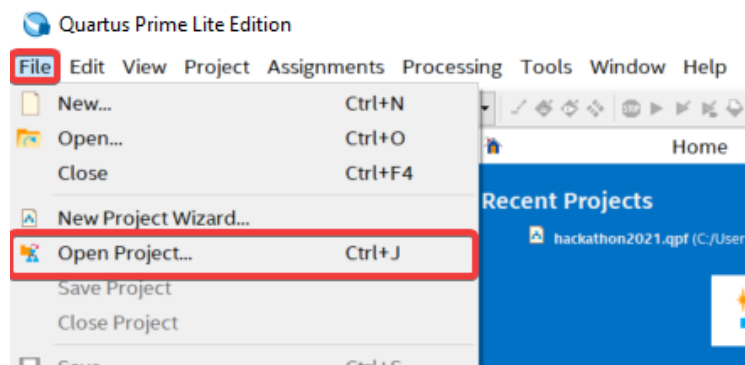
When you click Finish, the project will be created with the following settings:

Project directory:	C:\intelFPGA_lite\20.1
Project name:	test_connection
Top-level design entity:	test_connection
Number of files added:	0
Number of user libraries added:	0
Device assignments:	
Design template:	n/a
Family name:	Cyclone V
Device:	5CGXFC5C6F27C7
Board:	Cyclone V GX Starter Kit
EDA tools:	
Design entry/synthesis:	<None> (<None>)
Simulation:	<None> (<None>)
Timing analysis:	()
Operating conditions:	
Core voltage:	1.1V
Junction temperature range:	0-85 °C

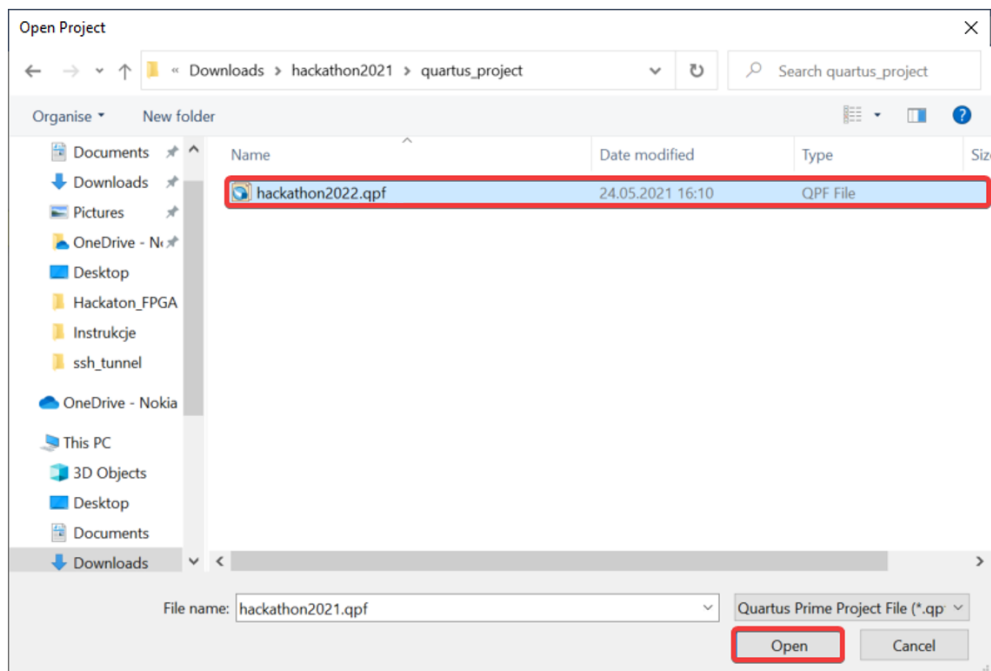
Navigation buttons: < Back, Next >, **Finish**, Cancel, Help

### 3.2. Import project

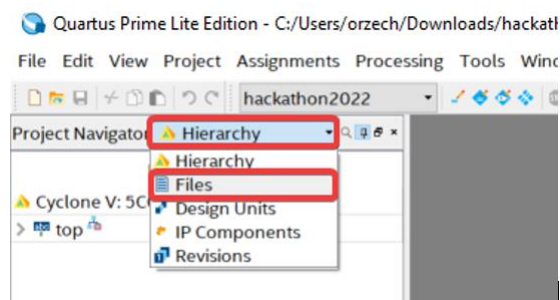
Select “Open Project” from “File” tab



Enter the path to project (\*.qpf file from “quartus\_project” directory)



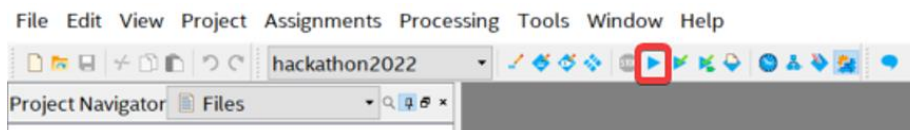
To see all files from project change view from “Hierarchy” to “Files”



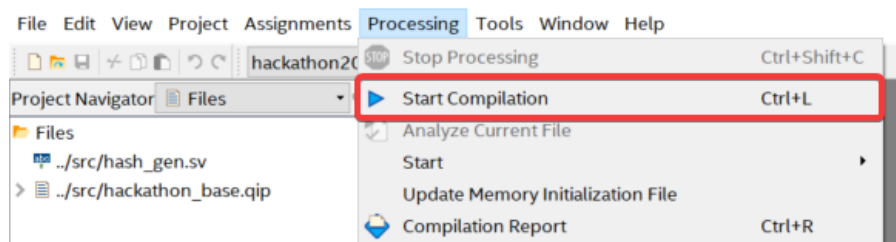


### 3.3. Compilation

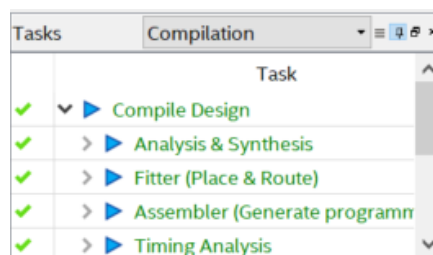
To compile project use button with triangle from tool bar



Or select "Start Compilation" from Processing tab



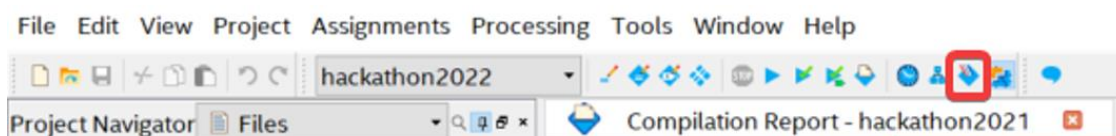
If compilation was successful all compilation phases should have green color



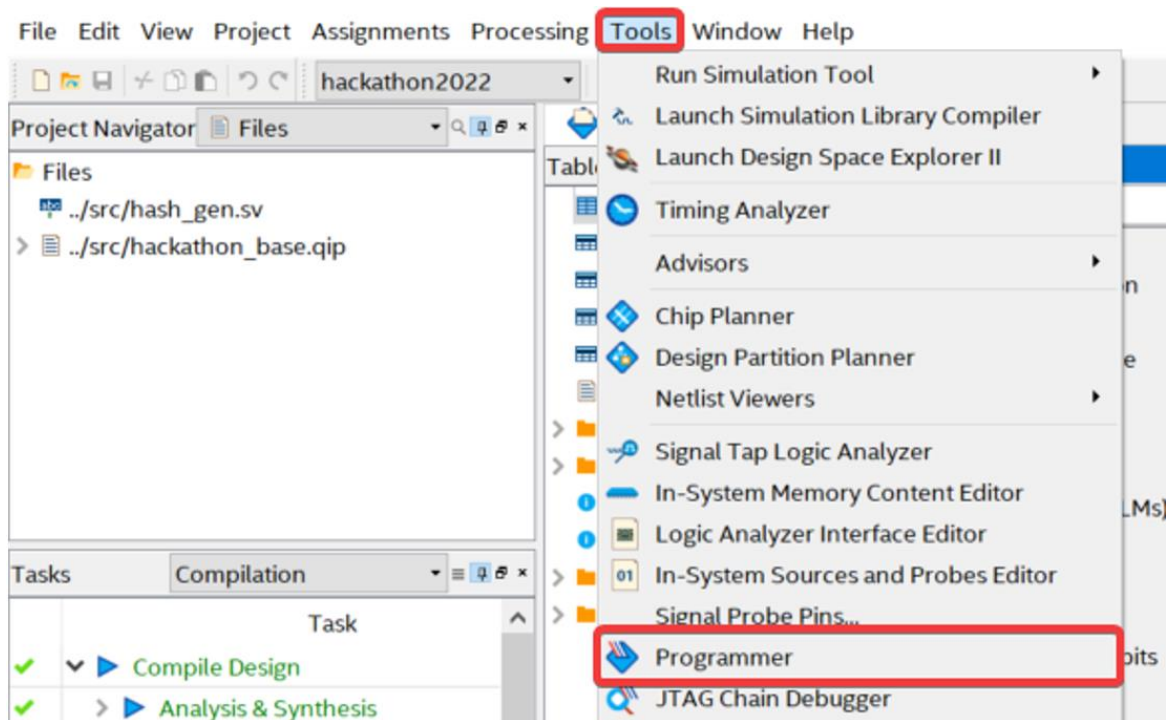
### 3.4. Uploading bitfile

To upload bitfile right usb device should be connected via virtualhere through the ssh tunnel.

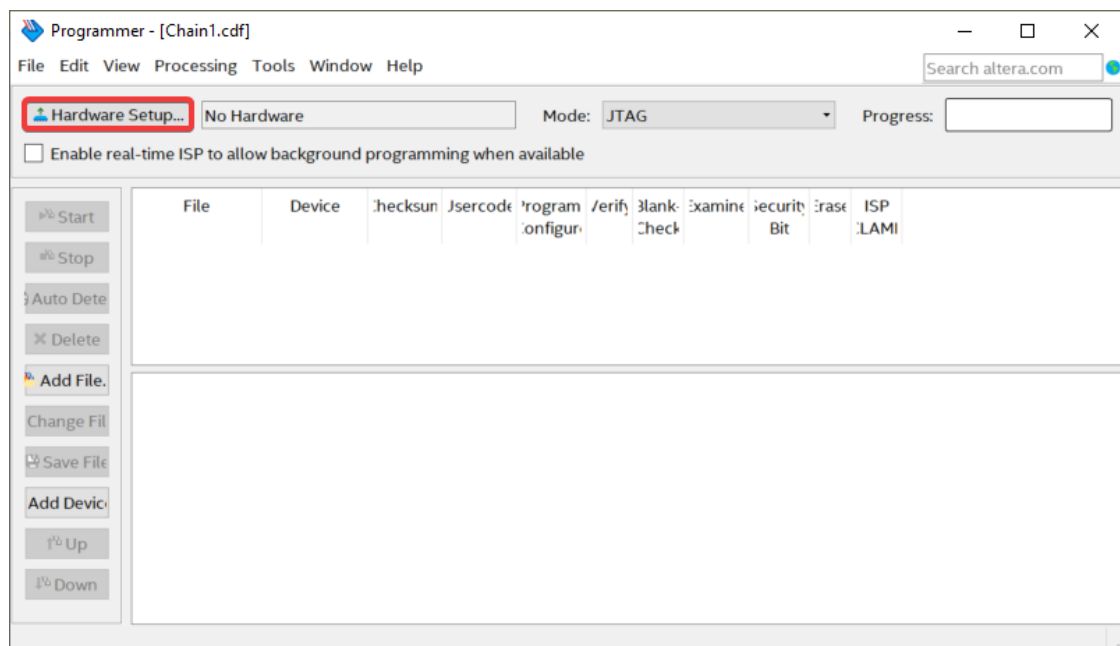
Run Programmer using symbol shown on picture below



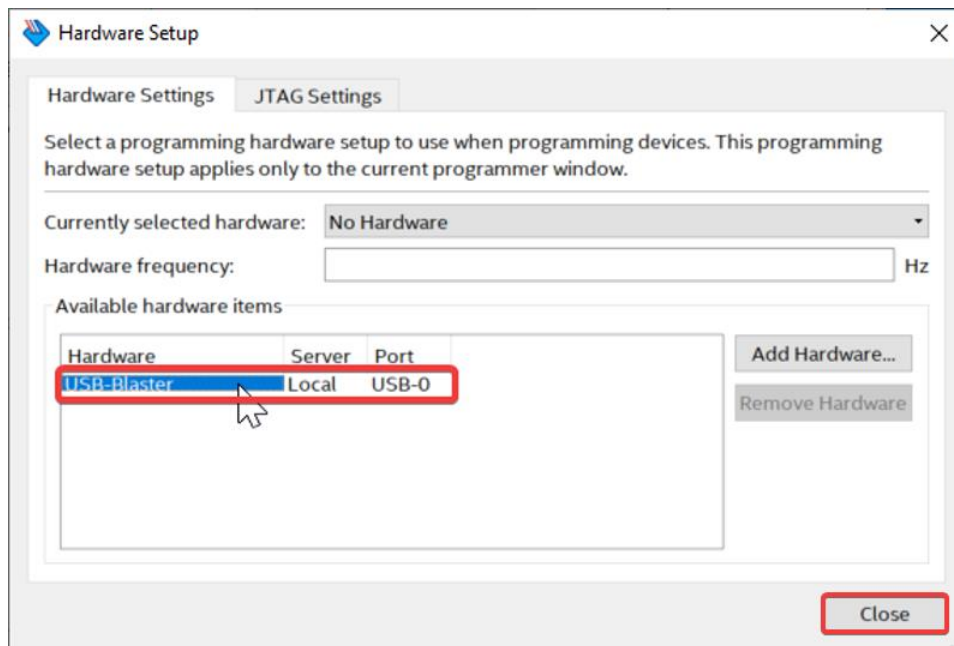
Or select "Programmer" option from "Tools" tab.



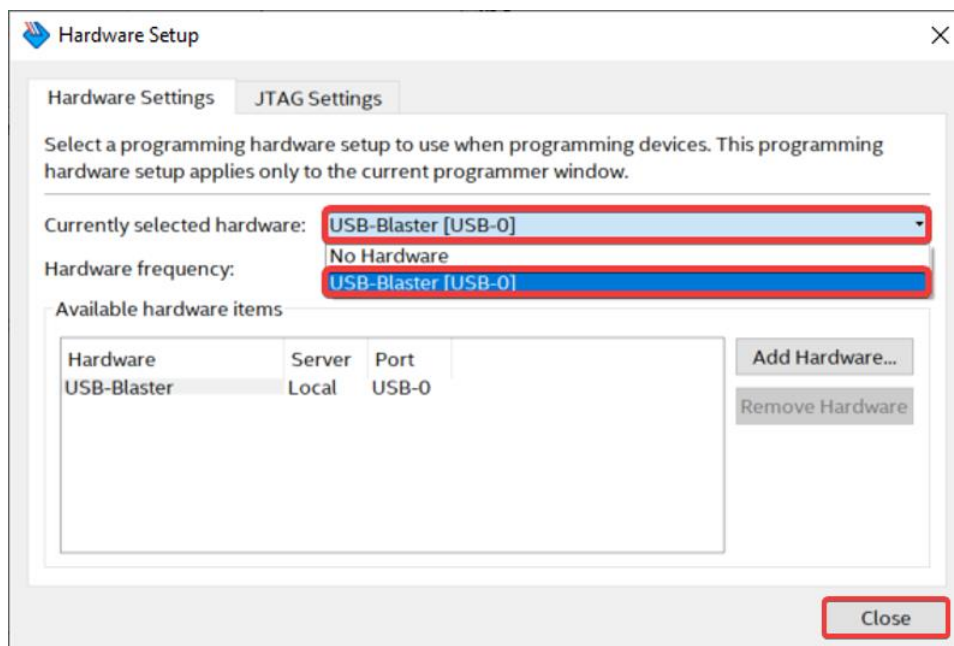
If “USB-Blaster” is not connected select “Hardware Setup”



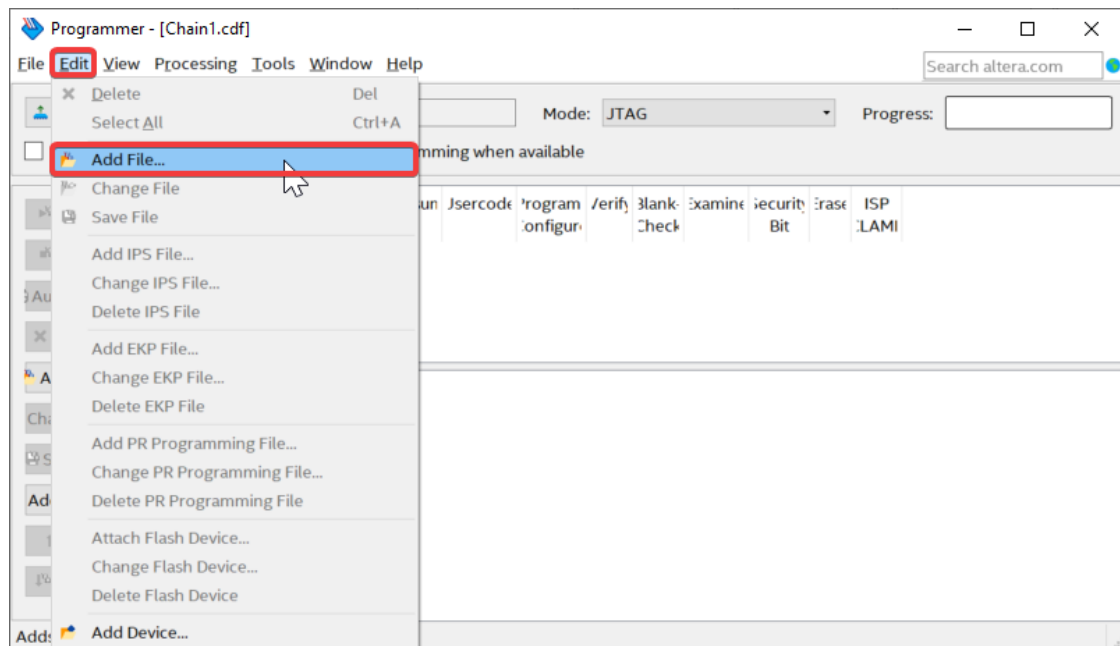
Double click on USB-Blaster connected



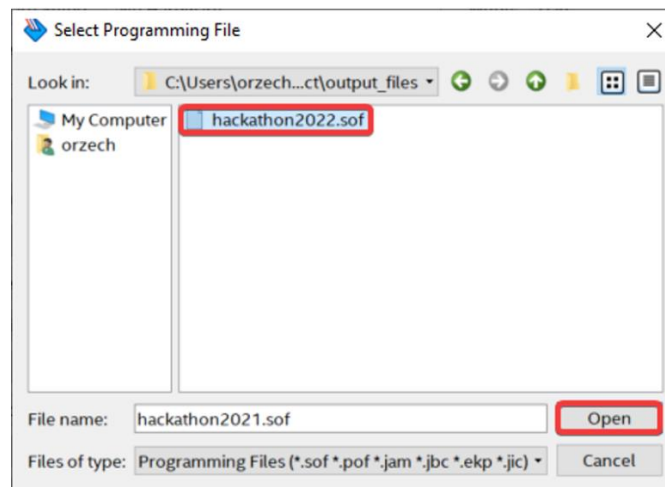
Or select USB-Blaster from drop-down list



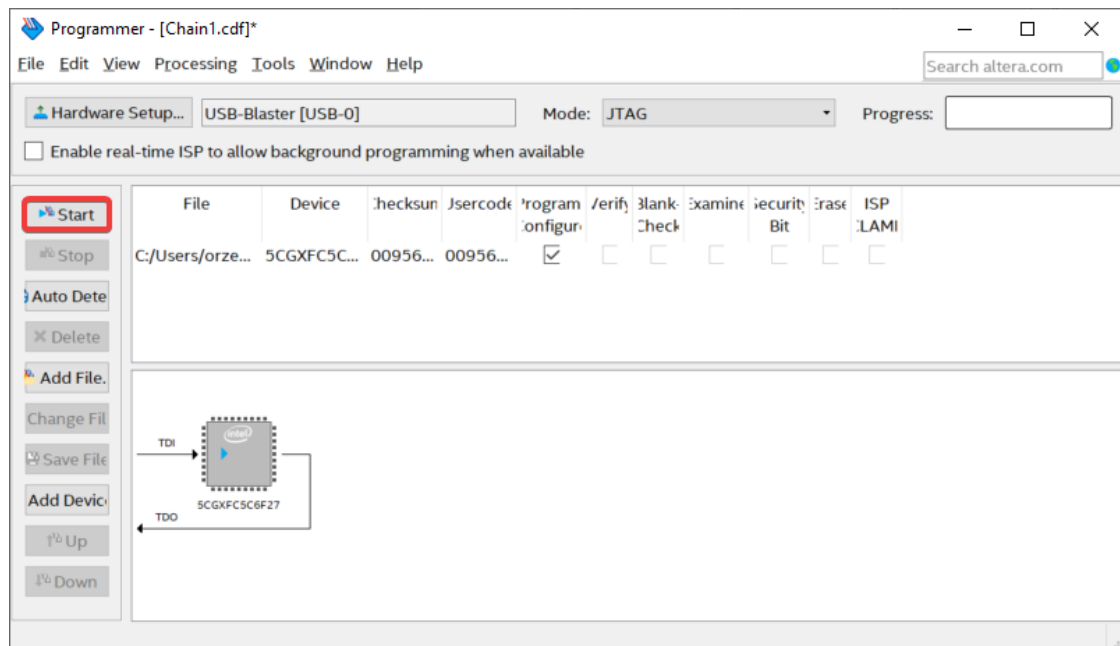
If there is need to choose bitfile expand “Edit” from toolbar and select “Add File”



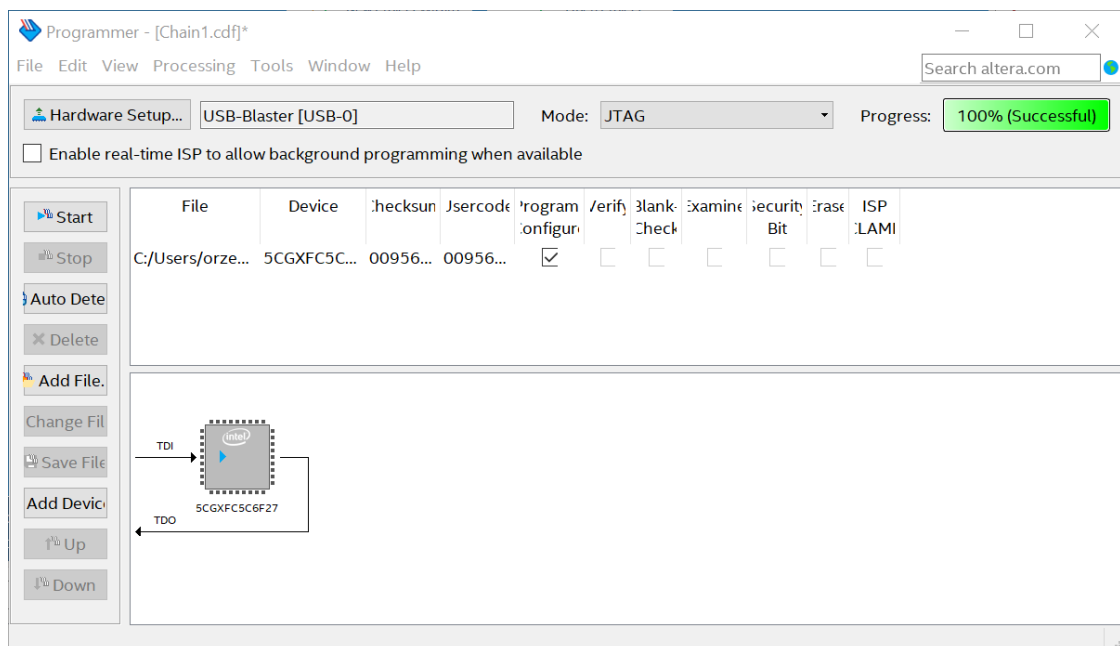
And select bitfile (file \*.sof from “output\_files” directory)



Use “Start” button to upload bitfile



If everything works bitfile will be uploaded with 100% Success



Sometimes it takes a while, so if “Progress” bar looks stuck - be patient.

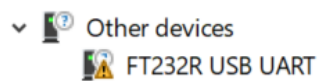
## 4. Additional actions

### 4.1. Files to download

- UART Driver ([link](#)) (For Windows users if needed)

### 4.2. UART Driver installation

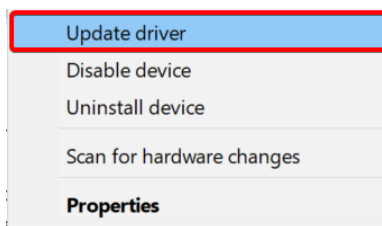
If using UART will be helpful and driver is not installed than in device manager you will see FT232R USB UART in Other devices.



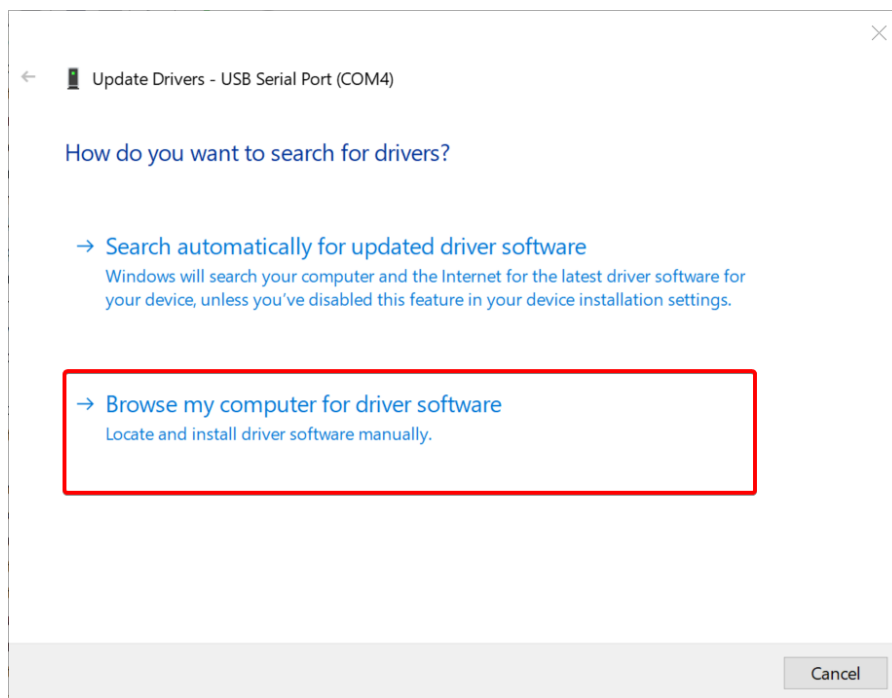
Download driver (link in 4.1). Choose the correct version for Your PC

Operating System	Release Date	Processor Architecture						
		x86 (32-bit)	x64 (64-bit)	PPC	ARM	MIPSII	MIPSIV	SH4
Windows*	2017-08-30	2.12.28	2.12.28	-	-	-	-	-

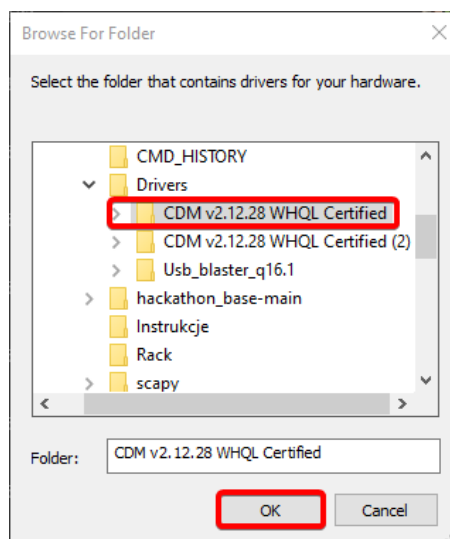
Unpack driver and right click on FT232R USB UART in device manager, select “Update driver” option.



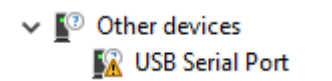
Select “Browse my computer for driver software”



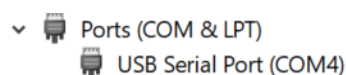
Select “Browse..” and find unpacked driver “CDM v2.12.28 WHQL Certified”



Select “Next”, driver should be installed and USB Serial Port should appear in Other devices.



Repeat all procedure for this device using same driver. Now USB Serial Port should appear and UART should work.



## 5. Extras

Additional actions that could be useful, but are not required to complete task

### 5.1. Free Active-HDL Student Edition (external simulation tool)

There is possibility to test Active-HDL from ALDEC - if needed to test software before using Riviera-PRO, which will be delivered on virtual machine

Link to download: [https://www.aldec.com/en/products/fpga\\_simulation/active\\_hdl\\_student](https://www.aldec.com/en/products/fpga_simulation/active_hdl_student)

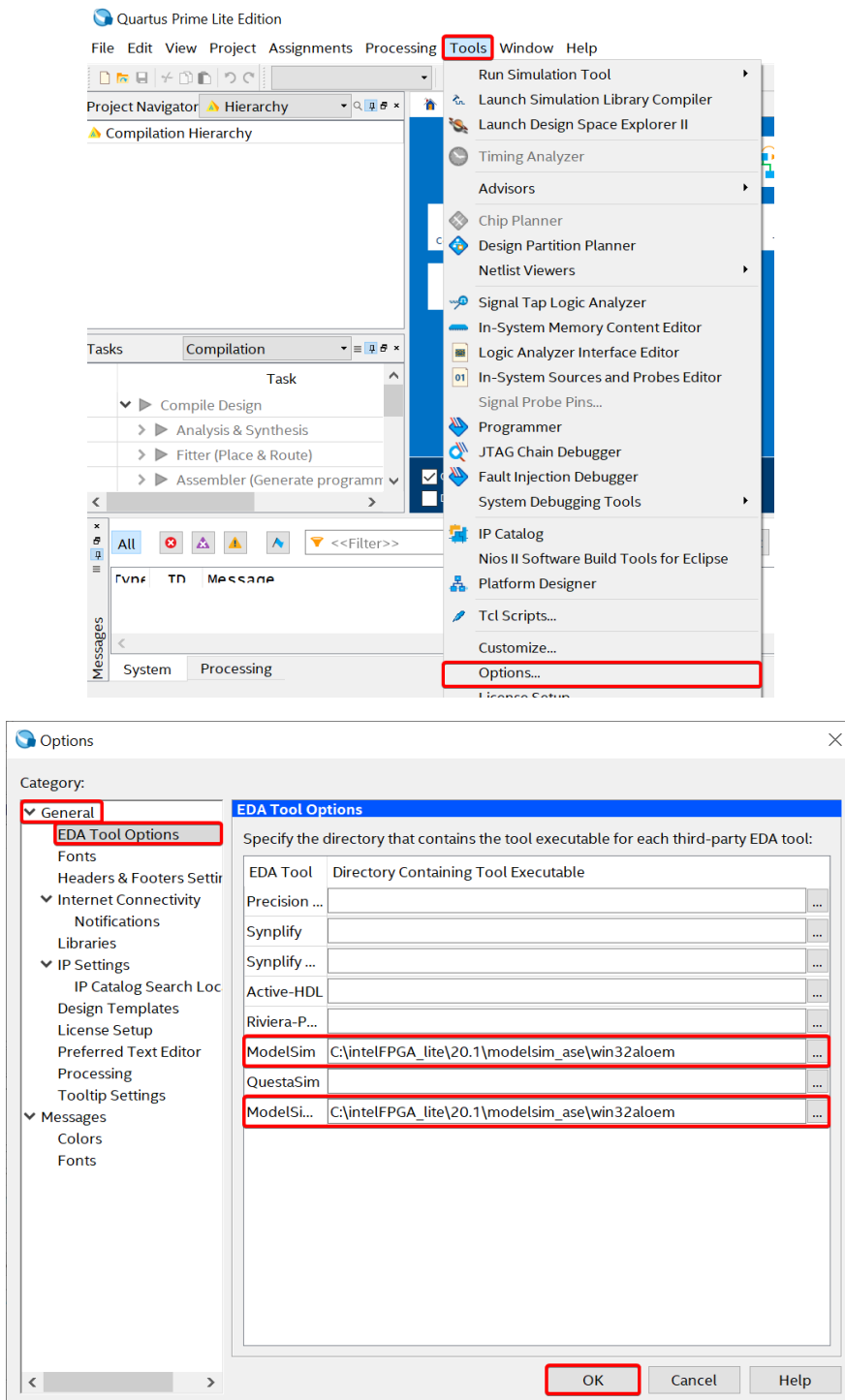


## 5.2. Modelsim (simulation tool integrated with this Quartus version)

### 5.2.1. Linking modelsim to quartus

Open **Tools** -> **Options**, go to section **General** -> **EDA Tool Options**, then enter valid paths under **ModelSim-Altera** (you can also fill **ModelSim** field) in Quartus installation directory – the path should end with:

`\modelsim_ase\win32aloem`

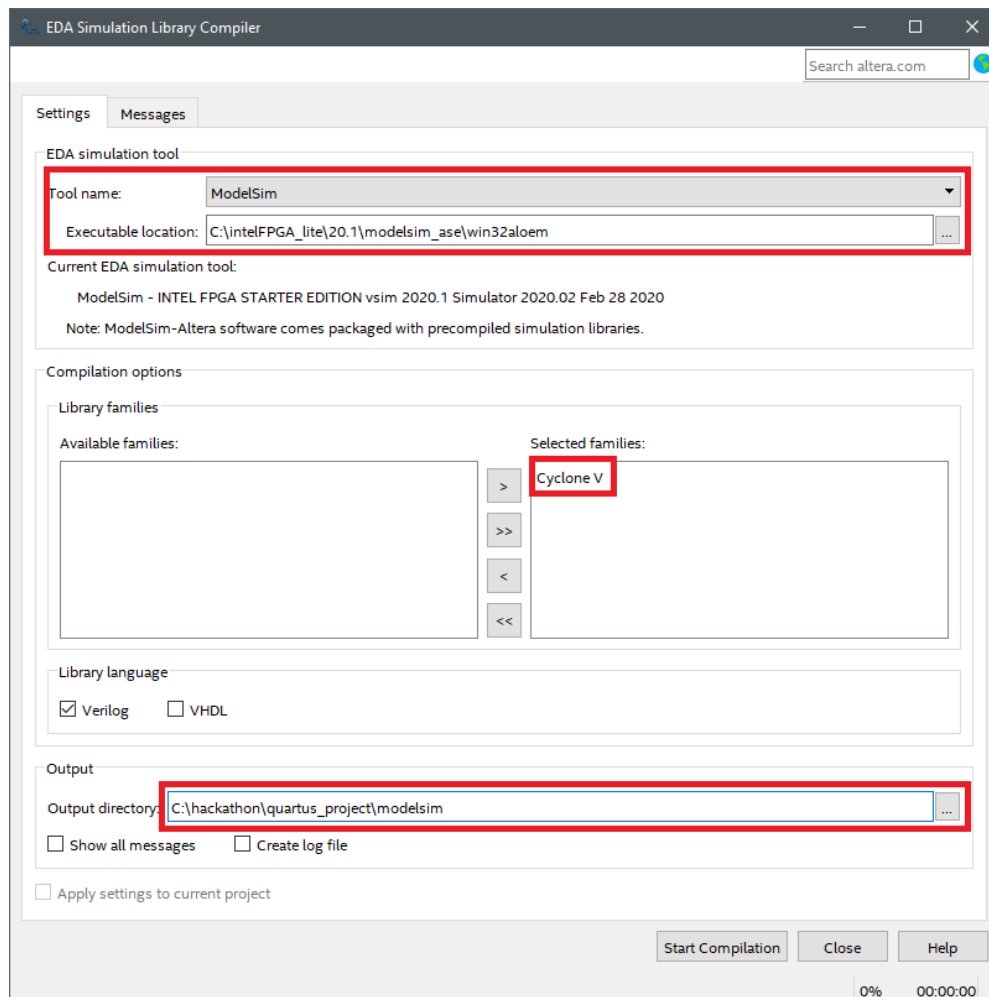


### 5.2.2. ModelSim Intel/Altera library compilation

Before running simulation, we need to compile Intel/Altera libraries.

Select **Tools -> Launch Simulation Library Compiler**. Make sure that selected tool and path matches with those entered in 5.2.1 and that **Cyclone V** is in selected families

Change output directory to your project directory with name **modelsim** – this directory will be created before compilation.



### 5.2.3 Running simulation

To run simulation using scripts delivered with base project select

**Tools -> Run Simulation Tool -> RTL Simulation**

This should open ModelSim, in ModelSim's transcript window use following commands to compile the design and run simulation

```
do ../../../../sim/comp.do      (to start compilation)
```

```
do ../../../../sim/sim.do       (to run simulation)
```