# Software Installation and Use

## Installation

### Quartus Prime Lite and ModelSim

Download the latest "Lite" edition of Quartus Prime from: `https://fpgasoftware.intel.com/`. From the "Individual Files" tab download:

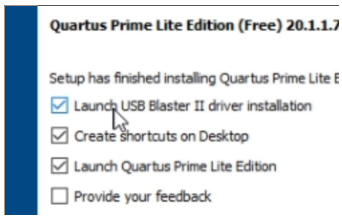
- Quartus Prime (includes Nios II EDS),
- ModelSim-Intel FPGA Edition (includes Starter Edition), and
- Cyclone IV device support

These three files are also available in the ELEX 2117 ShareOut folder. The BCIT ITS Knowledge Base has instructions on accessing ShareIn and ShareOut remotely.

Download all three files to the same folder and run the Quartus installer. Note that the downloads total about 3 GBytes and installation requires about 15 GBytes of free disk space.

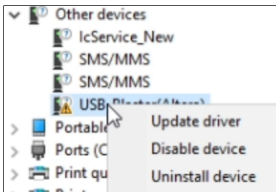
When you run the Quartus installer it should find the Modelsim and Cyclone IV device installer files in the same folder and offer to install them. Do so.

When the installation is complete the Quartus installer will show an option to install the drivers for the USB Blaster:

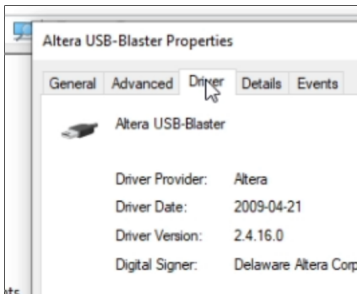



*Uncheck this box!* Windows will crash if you use the drivers in recent versions of Quartus for the "clone" USB Blaster in your parts kit.

Instead, download the `usb-drivers.zip` file available on the course website and extract the files. Run Device Manager and find the USB-Blaster device. Right-click on it and select "Update Driver":




and select the folder where you extracted drivers from the course website. Finally, check that the 2009 driver version was installed:




You may also be able to run Quartus and Modelsim from AppsAnywhere. This installs much faster but requires that you be on-line to use the software.

If you have problems when you run the programmer tool, see "Resolving USB-Blaster Problems" below.

There is a short video on the course website showing how to install the software.

## Logic Synthesis with Quartus

1. Run Quartus (on Windows select Intel FPGA ... > Quartus ...).

2. Select **File** > **New...** > **New Quartus Prime Project** > **OK**.

3. In the dialog boxes that follow: select a new folder for your lab (e.g. `C:\ELEX2117\lab1`), enter a project name (e.g. `lab1`), select an empty project, don't add any files, select the Cyclone IV Family, select the specific device `EP4CE6E22C8N`, and leave other settings at their defaults.

4. Add any existing design files using **Project** > **Add/Remove Files in Project...**, or create new ones using **File** > **New...** > **System Verilog HDL File**. One of the files must have a module with the top-level name specified above (in this example, `lab1`). The port names of this module will correspond to the names assigned to the FPGA's pins.

5. After all the design files have been created and added to the project, select **Processing** > **Start Compilation**. Correct any errors and recompile as necessary.

6. Select **Assignments** > **Pin Planner** and select the correct pin in the **Location** drop-down box for each I/O pin. Note that you must compile the project before the pin names are visible in Pin Planner. Recompile the project (**Processing** > **Start Compilation**) for the assignments to take effect.

7. Connect the FPGA board's micro-USB connector to a USB port. The power LED should go on.

8. Connect the "USB Blaster" to the JTAG port on the FPGA board and a free USB port. The POWER and ACT lights on the USB-Blaster should turn on.

9. Select **Tools** > **Programmer**, click on **Hardware Setup...**, select **USB-Blaster** from the drop-down and **Close**. **USB-Blaster** should appear next to **Hardware Setup...**

10. If necessary, click on **Add File...**, navigate to the location of the generated `.pof` file (typically in the `output_files` folder of the project folder) and select the `.pof` file.

11. Check that the **Program/Configure** checkboxes are checked and press **Start** to program the device. The progress bar should show 100%.

12. Test your design.

There is a short video on the course website showing how to use Quartus to synthesize a design and program the FPGA.

## SignalTap Logic Analyzer

This tool adds logic to your design that allows you to view signals in the FPGA. Once you have a project open and a design that compiles:

- select **Tools** > **Signal Tap Logic Analyzer**
- In the **JTAG Chain Configuration** box click on **Setup...** and select the USB Blaster.
- next to **SOF Manager** click on ... and select the correct `.sof` (FPGA serial programming) file from the `output_files` folder
- click on the download (  "Program Device") icon to verify that the device is connected and can be programmed.
- in the **Signal Configuration pane** under : **Clock:** click on ... and use the node finder to list the signals that can be monitored.

  Click on the  button ("Show more search options") beside the **List** button if necessary. Select `SignalTap:pre-synthesis` as the filter option and click on **List**. Select the clock input signal (e.g. `clock50`) as the clock by selecting it and clicking `>` to move it from the left to the right column.

  The icons in the node finder show if a signal is an input (`in`), output (`out`), register (`R`), or combinational logic (`C`). Buses can be expanded to select a subset of signals. Note that some signals may have been optimized away[1].

  Click **OK**.
- double-click in the Setup tab of the main window (it may say "Double-click to add nodes") to add the signals to be monitored. As before, list the pre-synthesis signals and add the one(s) you want to view.
- in the Trigger Condition column enter the signal values that you want to trigger on. For example, if you want to see what happens around the rising edge of a signal, set the trigger condition to "Rising Edge." There are many other ways to configure trigger conditions.

[1] If you really want to see these signals you'll need to add `keep` (for combinational logic) or `preserve` (for registers) directives to your code.

- recompile (  ) and reprogram (  ) the FPGA (answer yes when asked to save the `.stp` file and add it to the project)
- select an instance (typically `auto_signaltap_0`) and click on  ("Autorun Analysis") to start the logic analyzer
- the logic analyzer should display the signals being monitored in the Data tab each time the trigger condition is met
- you can right-click on a trace to change the display format (e.g. to binary) to make it easier to interpret the traces
- switch back to the Setup tab (below the waveform area) to make changes to the trigger conditions, sample depth, etc. Most changes will require recompilation.

## Simulation with ModelSim

1. Run ModelSim (on Windows select Intel FPGA ... > ModelSim ...).
2. (a) If this is the first time you simulate this design, select **File** > **New** > **Project**..., select the folder where your files are located as the Project Location, enter a suitable Project Name (e.g. `lab1`) and click OK. Select Add Existing File and select the file(s) that contain the entities you want to simulate, including your design files and testbench (if any), then select Close.

   (b) If you had already created a simulation project and it's not already open, select **File** > **Open**, select Files of type: Project Files (*.mpf) and select the project file,
3. Select **Compile** > **Compile All** to compile all the files in your project into the `work` library,
4. If there are syntax errors you will need to fix the error(s), save the file and go back to step 3,
5. Otherwise select **Simulate** > **Start Simulation**; select your testbench module from the `work` library and select **OK**.
6. Drag the signals you wish to view from the 'Sim' or 'Object' windows to the 'Wave' window (use the **View** menu to open windows).
7. Select **Simulate** > **Run -All**; this will run the simulation until it's complete.
8. The Transcript window will contain output from the testbench.
9. The Wave window will show the waveforms (select the Wave window, click on '+' and **Wave** > **Zoom** > **Full**); you can use a screen capture utility (e.g. Windows Snip tool) to save the waveforms.
10. If the results are not as expected, correct the errors, run **Compile** > **Compile All**, **Simulate** > **Restart**..., click **OK** and **Simulate** > **Run -All**.

## Resolving USB-Blaster Problems

If possible, check your USB-Blaster, cables and FPGA board at a BCIT lab session by programming one of the `.sof` files from the course web site. Ask the lab instructor for help if necessary.

If your hardware works with the lab PC but not with your own computer, follow the troubleshooting guide below.

You will need:

- a Windows 10 PC with Quartus Prime installed
- two free USB ports (or one port and one USB charger)
- a USB flash drive
- (in rare cases) a known-working USB-Blaster, cables, and FPGA board

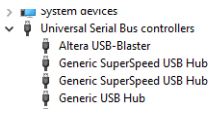
Do the following, *in order*:

1. Unplug any USB peripherals that are not required to run your computer, *including any Analog Discovery 2*, and restart your computer.
2. Check that the flash drive works in the USB port you plan to use for the USB-Blaster.[2] If other devices don't work when plugged into the ports you plan to use, shut down the computer (don't just restart it). If other USB devices still don't work, use a different USB port or computer.

[2] USB port power is turned off when you draw too much current (e.g. due to a short circuit on your breadboard).

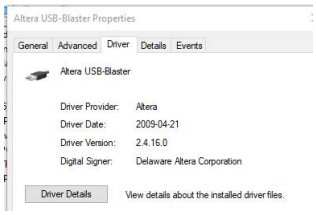
3. Disconnect the USB-Blaster from the FPGA and plug it into the USB port. Check the green POWER LED. If it's not on, try a known-good mini-USB cable and USB-Blaster.

4. Run Windows' Device Manager and expand the USB controllers section. You should see Altera USB-Blaster listed:

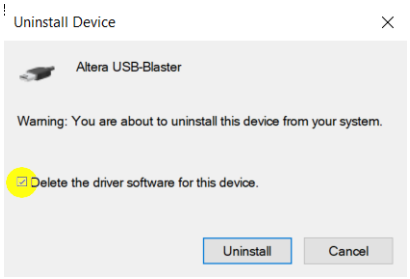



If not, (this is rare) try a known-good mini-USB cable and USB-Blaster.

5. Right-click on Altera USB Blaster, select Properties and click on the **Driver** tab. Check that Driver version 2.4.16.0 is installed as shown below.

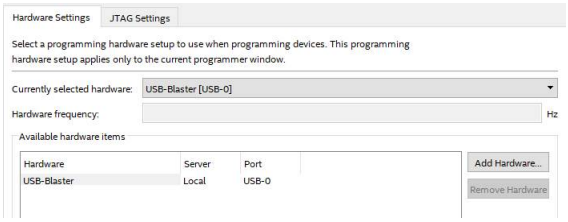



if you have a different version then you're probably using the drivers installed by Quartus rather than the ones from the course web site. Right-click on Altera USB Blaster and select **Uninstall device**. Check the box for Delete the driver software for this device.[3] and click **Uninstall**.




Install the Altera USB Blaster drivers from the course web site and check the version again. Do not proceed until the correct driver version has been installed.

6. In Quartus, run **Tools** > **Programmer** and click on **Hardware Setup**. Check that you can select the USB-Blaster:




If not, close the programmer application, unplug any USB devices that may be using an FTDI serial interface IC, *including the Analog Discovery 2*, and try again again. If the Quartus programmer shows in Device manager but the Quartus programmer still cannot detect the USB-Blaster, go to step 1 (and make sure all USB devices are unplugged before restarting).

7. At this point the USB-Blaster should still be disconnected from the FPGA. Remove all connections to the FPGA pins (to prevent issues due to short-circuited pins) and connect the micro-USB power connector to a USB power supply or USB port. If the red power LED does not light, check the USB port/charger and cable. Replace the micro-USB cable or FPGA board if the power LED does not light.

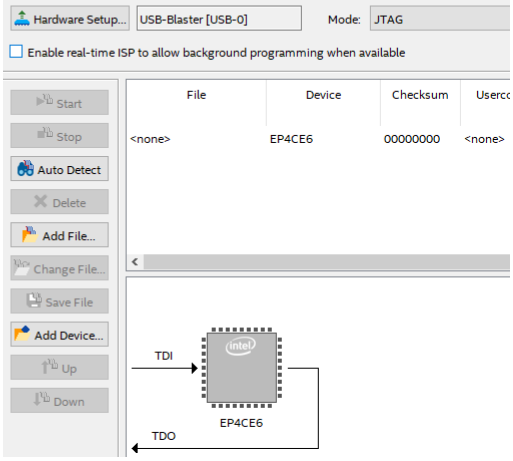
**Note:** The USB-Blaster supplies enough power to light the red power LED on the FPGA board but not enough for the FPGA to operate. Do not rely on the power LED to determine whether the FPGA is receiving power.

8. Connect the USB-Blaster to the FPGA board's JTAG connector using the ribbon cable. Note that the connectors are keyed.

9. Click on **Auto Detect** to check that the device is detected[4]:

[3] You must delete the newer driver so it does not get re-installed.

[4] If multiple devices are detected on the JTAG chain select the EP4CE6




10. If the device is not detected you should see a pop-up message saying "Unable to scan device chain." Run **Tools > JTAG Chain Debugger** or Click **Yes** if the message offers to run it for you.

11. Click on **Test JTAG Chain**.

    If the error is "Chain is in use." reboot the computer to restart the USB-Blaster driver.

    If the error is "No device detected." double-check the connection between the USB-Blaster and the FPGA board. Replace the USB-Blaster, ribbon cable, or FPGA board[5].

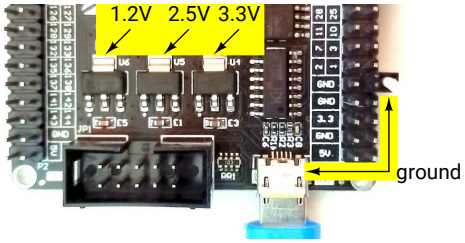
## Troubleshooting the Hardware

If substituting known-good hardware shows that the problem is with your FPGA board or USB-Blaster some additional checks may confirm the fault. In most cases the problem will be with a power supply or I/O pin.

Schematics for the FPGA board and the USB-Blaster are available on the course website if you want to do additional troubleshooting.

## FPGA Board

Disconnect everything from the header pins and the USB-Blaster from the JTAG port. Connect a known-good 5 V USB supply into the micro-USB connector:




If the configuration flash memory has not been re-programmed then the on-board LED's should show a flashing a sequence. If this is the case then the board's power supplies are likely not faulty.

But it's relatively easy to check these. Use a DMM to measure the voltage (with respect to a header pin marked GND or the micro-USB connector shield) on the heat sink tabs of the three voltage regulators (U6, U5 and U4). They should be within 5% of 1.2, 2.5 and 3.3 volts DC.

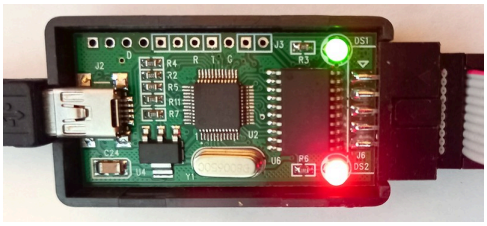
A very low or zero voltage, often accompanied by a burnt smell or an overheating component, indicates a short circuit or failed component. Identify the cause (but don't burn your fingers!). Remove the short, replace the component or replace the board.

A high voltage (5V) on the regulator output is less common, will probably have caused extensive component damage, and require replacing the board.

It's also possible for individual I/O pins to have failed as a result of exceeding their voltage or current ratings. In this case you may be able to complete your labs by substituting working I/O pins.

## USB-Blaster

There aren't many components in the STM-32-based ("clone") USB-Blaster. You can remove the top cover:



and check for 3.3 VDC between the USB shield and the heat sink tab on the voltage regulator (U4). On the bottom of the board you'll find ground and 3.3 VDC supplied from the FPGA board.

[5] Although the most common reason for this error is that there is no power connected to the FPGA board.