

Quartus Installation Instructions for Linux (Last Updated for Quartus v19.1)

This method was tested on Ubuntu 22.04 and Fedora Silverblue 36. I will note the differences between the two system installations when needed. I have not done as extensive testing on Ubuntu, so you may need to install more packages than are listed.

Prerequisites: Installing Dependencies

Quartus should work mostly out of the box with your system's 64 bit libraries. However, ModelSim requires 32-bit versions of libraries, which we must install.

Ubuntu 22.04

ModelSim dependencies:

```
sudo dpkg --add-architecture i386

sudo apt update

sudo apt install libc6:i386 libncurses5:i386 libstdc++6:i386 lib32ncurses6 libxft2 libxft2:i386 libxext6 libxext6:i386
```

Fedora/Fedora Silverblue

All dependencies should be installed inside of a toolbox container to avoid messing up your system with a bunch of old 32-bit libraries

```
sudo dnf install toolbox

toolbox create quartus

toolbox enter quartus
```

Quartus and ModelSim Dependencies:

```
sudo dnf remove libedit.x86_64

sudo dnf install libnsl libXrender unixODBC.i686 unixODBC-devel.i686 ncurses-compat-libs.i686 zeromq-devel.i686 libXext.i686 alsa-lib.i686 libXtst.i686 libXft.i686 libxml2.i686 libedit.i686 libX11.i686 libXi.i686 wget

mkdir ~/Downloads/quartus-deps

cd ~/Downloads/quartus-deps
```

Quartus Installation Instructions for Linux

(Last Updated for Quartus v19.1)

```
wget https://kojipkgs.fedoraproject.org/vol/fedora_koji_archive00/packages/freetype/2.4.12/6.fc20/i686/freetype-2.4.12-6.fc20.i686.rpm

wget https://kojipkgs.fedoraproject.org/vol/fedora_koji_archive00/packages/freetype/2.4.12/6.fc20/x86_64/freetype-2.4.12-6.fc20.x86_64.rpm

wget https://kojipkgs.fedoraproject.org/vol/fedora_koji_archive05/packages/fontconfig/2.12.6/4.fc28/i686/fontconfig-2.12.6-4.fc28.i686.rpm

wget https://kojipkgs.fedoraproject.org/vol/fedora_koji_archive05/packages/fontconfig/2.12.6/4.fc28/x86_64/fontconfig-2.12.6-4.fc28.x86_64.rpm

sudo dnf install *.rpm
```

Part 1: Installing Quartus Prime

1. Go to [Intel's Download Center](#) for Quartus Prime Lite 19.1 for Linux and download the following files: Quartus Lite Setup, ModelSim Setup, Cyclone V device support, MAX V device support, and MAX 10 device support. See the below screenshot.
2. Run the following commands in your Downloads folder:

```
chmod +x QuartusLiteSetup-19.1.0.670-linux.run

./QuartusLiteSetup-19.1.0.670-linux.run
```

3. Follow the prompts. Install all downloaded components in the default location, which should be `YourHomeDir/intelFPGA_lite/19.1`.

Part 2: Installing OOTB Programmer Support

The OOTB programmer is not automatically detected by the Quartus Programmer tool. We have to add drivers for it to Quartus and make sure that the programmer is controlled by the Quartus software drivers instead of the generic `ftdi_sio` kernel module.

1. [Download the MBFTDI driver from Marsohod](#) and unzip it.
2. Copy `libjtag_hw_mbftdi-blaster.so` to the Quartus library folder

```
cp libjtag_hw_mbftdi-blaster.so ~/intelFPGA_lite/19.1/quartus/linux64
```

3. Add a new udev rule that will unload the `ftdi_sio` kernel module every time the OOTB programmer is plugged into your system.

```
sudo nano /etc/udev/rules.d/51-usbblaster.rules
```

File contents:

```
SUBSYSTEM=="usb",ENV{DEVTYPE}=="usb_device",ATTR{idVendor}=="0403",ATTR{idProduct}=="6010",MODE="0666",RUN+="/usr/sbin/rmmod ftdi_sio"
```

When you are done editing your file in nano, use Ctrl+X to save your file. Press 'y' to confirm that you would like to save and then press enter to confirm the file name.

4. Modify the Quartus programmer shell script to always check for new JTAG programming hardware when the Quartus programmer is opened.

```
sudo nano ~/intelFPGA_lite/19.1/quartus/bin/quartus_pgmw
```

Add these lines at the top of the file after the line IFS=... Make sure to change "YOURHOMEDIR" to the name of your home directory.

```
#Check for connected programmers  
/home/YOURHOMEDIR/intelFPGA_lite/19.1/quartus/bin/jtagconfig
```

5. Connect your programmer to your PLD. Plug in your PLD's power USB. Finally, plug in your programmer's USB cable. Open Quartus and open up Tools>Programmer. Click "Add Hardware". Verify that MBFTDI-Blaster shows up in the hardware selection dialog.
 - If not, try `sudo rmmod ftdi_sio` manually to see if `ftdi_sio` is still loaded. you can also add "blacklist ftdi_sio" to `/etc/modprobe.d/ftdi-blacklist.conf` to permanently unload the `ftdi_sio` kernel module on system boot (not recommended). check the path to `rmmod` in your udev rule. (which `rmmod`)

Part 3: Making ModelSim Run

ModelSim Starter depends on old 32-bit libraries that we had to install above. To make it use these libraries, we have to modify the `vsim` shell script. Tip: use Ctrl+W to find text in nano.

```
sudo nano ~/intelFPGA_lite/19.1/modelsim_ase/bin/vsim
```

- Change `mode=${MTI_VCO_MODE:-""}` to `mode=${MTI_VCO_MODE:-"32"}`
- Change `vco="linux_rh60"` to `vco="linux"`

Part 4: Easy Access

Ubuntu

1. If you are on Ubuntu or not using a container, you can add the Quartus and ModelSim binaries to your PATH variable. Make sure to use your own home directory.

```
nano ~/.bashrc
```

Add to the bottom of the file:

```
export PATH="$PATH:$HOME/intelFPGA_lite/19.1/quartus/bin:$HOME/intelFPGA_lite/19.1/modelsim_ase/bin"
```

2. We will also make a .desktop file so that Quartus is accessible in your start menu. Alternatively, you can copy the .desktop placed on your Desktop from the Quartus installation to ~/.local/share/applications/

```
nano ~/.local/share/applications/quartus.desktop
```

File Contents: (make sure to replace **YOURHOMEDIRECTORY** with the name of your home directory)

```
[Desktop Entry]
Type=Application
Version=19.1
Name=Quartus Prime
Comment=Quartus (Quartus Prime 19.1) Lite Edition
Icon=/home/YOURHOMEDIRECTORY/intelFPGA_lite/19.1/quartus/adm/quartusii.png
Exec=quartus
Terminal=false
```

Fedora/Fedora Silverblue

1. Since we are using a container, we should not add Quartus to the system PATH variable because then the host system would try to run Quartus outside of the container and Quartus's dependencies would not be met. Instead, we will only create a desktop file for it so that it appears in the start menu of your desktop environment.

```
nano ~/.local/share/applications/quartus.desktop
```

File Contents: (make sure to replace YOURHOMEDIRECTORY with the name of your home directory)

```
[Desktop Entry]
Type=Application
Version=19.1
Name=Quartus Prime
Comment=Quartus (Quartus Prime 19.1) Lite Edition
Icon=/home/YOURHOMEDIRECTORY/intelFPGA_lite/19.1/quartus/adm/quartusii.png
Exec=/usr/bin/toolbox run -c quartus
/home/YOURHOMEDIRECTORY/intelFPGA_lite/19.1/quartus/bin/quartus
Terminal=false
```

Sources

<https://gist.github.com/vchernin/04b30f0eb72385592d4a2ece615476a8>

<https://marsohod.org/11-blog/335-mbftdi-quartus-ubuntu>

<https://gist.github.com/Razer6/cafc172b5cfae189b4ecda06cf6c64f>

https://mil.ufl.edu/3701/docs/quartus/linux/ModelSim_linux.pdf