

# How to program Nexys3 FPGA board on Linux using Digilent Adept software: A guide for beginners

📅 5. October 2012 / 👤 Uli Köhler / 📁 FPGA

Instead of having to buy expensive JTAG adapters like the Xilinx Platform Cable USB, Digilent FPGA boards like the Nexys3 support programming by using the Adept protocol being translation into JTAG internally. The Adept software is pretty easy to use if you know the basics of how to use the executables.

This is a beginner's tutorial, so if you already have some experience with FPGAs, you might want to skip some parts. Still you need to have some knowledge about how to use the shell.

## Prerequisites:

- You got a *.bit* file compiled for your specific FPGA model. To generate this from VHDL or Verilog code, you should use the Xilinx ISE (WebPack is free, but you need to register!) – if you use third-party software the calls to the Xilinx Toolchain may also be encapsulated in a Makefile or similar.
- You got your Nexys3 plugged into your computer using a standard micro-USB cable. Note that there are two micro USB ports on the Nexys3 board: One labeled *UART* and one labeled *USB PROG*. You need to insert the USB cable into the port labeled *USB PROG*
- The Nexys3 is turned on: You can turn it on and off using the switch adjacent to the USB PROG port. If the board is turned on, a red light will appear. Actually the switch doesn't turn the board off but sets it to receive power from the 5.5×2.1mm barrel adapter adjacent to the switch.

## Installation

- **Download** the *Digilent Adept2 runtime* and *Digilent Adept2 utilities* from <http://www.digilentinc.com/Products/Detail.cfm?Prod=ADEPT2>
- Extract both archives and (**as root**) **execute install.sh** for each of the directories being created. For most users selecting the default answer to the questions is sufficient.
- **Restart udev** to reload permissions. In some cases the udev rules don't work right away, so **if you get a permission error** on any command trying to access the FPGA board, just **execute it as root**.
- **Restart** your shell.
- Check if the software can find your board: Run `djtgcfg enum`

The output should look like this:

```
Found 1 device(s)

Device: Nexys3
Product Name: Nexys3
User Name: Nexys3
Serial Number: 210182392922
```

What you need to remember here is the string **after Device**: *If you use the Nexys3 board and you only got a single one connected to your computer at one time, it should usually just be Nexys3*. We call this string the **device identifier**.

## Identifying the correct device:

- JTAG is organized in chains in order to make it possible to communicate to more than one device with a single JTAG link. Because Adept is not designed for the Nexys and Atlys series alone but more as a versatile FPGA programming toolchain, the Adept tools need the JTAG ID of the device you want to program.
- To get a list of the devices for a single device identifier, run `djtgcfg init -d`. Therefore, in most cases, you want to run something like

```
djtgcfg init -d Nexys3
```

```
Initializing scan chain...
Found Device ID: 34002093

Found 1 device(s):
Device 0: XC6SLX16
```

The `djtgcfg` tool tells you that it has found only one JTAG device with **identifier 0**: The *XC6CSLX16* FPGA. On the Nexys3 board, there is only one device in the JTAG chain because the other programmable ICs like the PICs need to be programmed in other ways but on other boards you might need to check the names of the individual devices to see which identifier you want to use.

If you know the device identifier (e.g. `Nexys3`) and the JTAG identifier (e.g. `0`) you can easily run the programmer:

```
djtgcfg prog -d -i -f
```

Usually this boils down to something like this:

```
djtgcfg prog -d Nexys3 -i 0 -f my_awesome_program.bit
```

In case of success, you should get an output similar to the following:

```
Programming device. Do not touch your board. This may take a few minutes...
Programming succeeded.
```

That means you've successfully programmed your FPGA!

Note that the Nexys3 board doesn't save the program on any nonvolatile memory, so you need to reprogram your FPGA after you turned the power off and back on.

#### Alternative method of getting the program on the FPGA:

On the Nexys3, there is a USB host interface where you can plugin an any standard FAT-formatted USB drive. The software on the PIC microcontroller will then read the filesystem on the USB drive and if there is a single `.bit` file in the top level directory of the filesystem, it will program it into the FPGA. If this doesn't work, try another USB drive or renaming your file.

---

← C++11 : Iterate over smart pointers using foreach loop

How to fix node-waf "ImportError: No module named Scripting" on Linux →

## Search

 

## Categories

> Algorithms

> Allgemein

> Android

> Bioinformatics

> Build systems

> C/C++

> Cryptography

> Databases