

## Tarjetas de desarrollo FPGA

### Programming Guide

#### Alorium Sno

##### Steps

- Solder JTAG and power pins
- Install Quartus Prime Lite Edition
- Install the driver for the USB Blaster
  - Control Panel > Device Manager
  - Search for USB-Blaster
  - Update Driver > Browse > \intelFPGA\_lite\18.1\quartus\drivers
- With Quartus Prime Lite Edition software generate programming file [Assembler]
- Open Quartus Prime Lite Edition
- Go to Tools > Programmer
- Make sure that "USB-Blaster [USB-0]" is selected in Hardware Setup, otherwise click Hardware Setup and in "Currently selected hardware:" select it. Close
- Check the correct FPGA by clicking "Auto Detec" and select 10M16SA
- Click "Add File" and go to the project folder and search and open the programming file, either xxx.sof (lose at powerdown) or xxx.pof
  - project\_name/output\_files/xxx.pof
- Select file and delete it
- Check the two boxes "Program/Configure" and "Verify" of the main file
- Program the FPGA by clicking "Start"

##### Notes

- \*all steps were done using Windows 7
- When using on-chip memory IP cores change configuration mode to "Single Uncompressed Image with Memory Initialization" inside Device>Device and Pin Options...>Configuration mode

#### Alorium XLR8

##### Steps

- Solder JTAG pins
- Install Quartus Prime Lite Edition
- Install the driver for the USB Blaster
  - Control Panel > Device Manager
  - Search for USB-Blaster
  - Update Driver > Browse > \intelFPGA\_lite\18.1\quartus\drivers
- With Quartus Prime Lite Edition software generate programming file [Assembler]
- Open Quartus Prime Lite Edition
- Go to Tools > Programmer
- Make sure that "USB-Blaster [USB-0]" is selected in Hardware Setup, otherwise click Hardware Setup and in "Currently selected hardware:" select it. Close
- Check the correct FPGA by clicking "Auto Detec" and select 10M08SA
- Click "Add File" and go to the project folder and search and open the programming file, either xxx.sof (lose at powerdown) or xxx.pof
  - project\_name/output\_files/xxx.pof
- Select file and delete it
- Check the two boxes "Program/Configure" and "Verify" of the main file
- Program the FPGA by clicking "Start"

##### Notes

- \*all steps were done using Windows 7
- Notch of the USB Blaster's connector should face the inside of the board.

- For the use A0-5 pins there is 2 options: analog or digital, both connected to different pins on the FPGA. When used as digital, user has to enable this pins (buffer 3.3V-5V) through 6 pins also connected to the FPGA (see schematic).
- RST pin is active low.
- When using on-chip memory IP cores change configuration mode to "Single Uncompressed Image with Memory Initialization" inside Device>Device and Pin Options...>Configuration mode

## Papilio Pro

### Steps

- Install ISE Design Suite
- With ISE Design Suite software generate programming file [Generate Programming file]
- Download the file "Papilio-Loader-2.8-Setup.exe" (1) and install it
- Go to the project folder and search for the programming file xxx.bit
  - Double click and wait for the Papilio Loader to open
- File > Preferences > User Mode > Expert
- In "Target board" field select "Papilio One or Papilio Pro"
- Make sure the "Target .bit file" loaded the correct file
- Do not touch the other fields and in the "Operations" box check "Erase", "Write to" and "Verify"
- In the drop-down menu select either "FPGA" (lose at powerdown) or "SPI flash"
- Click "Do Selected Operations"

### Notes

- \*all steps were done using Windows 7
- (1) <http://forum.gadgetfactory.net/files/file/10-papilio-loader-gui/>
- if using VirtualBox make sure to enable USB: <https://www.techrepublic.com/article/how-to-enable-usb-in-virtualbox/>

## TinyFPGA BX

### Steps

- Install Lattice iCEcube2 (Windows 7)
- Install toolchains APIO and tinyprog from the terminal
  - \$ (sudo) pip install apio tinyprog
  - \$ apio install system scones icestorm iverilog
  - \$ apio drivers --serial-enable
- Add yourself to the dialout group in order for your user to be able to access serial ports
  - \$ sudo usermod -a -G dialout \$USER
- Connect your TinyFPGA BX board and make sure the bootloader is up to date
  - tinyprog --update-bootloader
- With Lattice iCEcube2 software generate programming file [Generate Bitmap] - Windows 7
- Plug the TinyFPGA BX if it is not already
- In a terminal window go to Lattice project folder and search for the programming file
  - project\_name/project\_name\_Implmnt/sbt/outpus/bitmap/xxx.bin
- Program the FPGA
  - \$ (sudo) tinyprog -p xxx.bin
- The configuration will remain until you press the reset button, when it would load the bootloader
- When connected to a dumb power supply the configuration will be reloaded after 1 second

### Notes

- \*Windows 7 was ONLY used for iCEcube2, the other steps were done in Ubuntu 14.04
- <https://tinyfpga.com/bx/guide.html>
- <https://www.instructables.com/id/Getting-Started-With-the-Tinyfpga/>
- <https://github.com/tinyfpga/TinyFPGA-BX>

## MAX1000

### Steps

### Notes

## Temperature Test

## Alorium Sno

**Alorium XLR8**

**Papilio Pro**

**TinyFPGA BX**

**MAX1000**