
apio Documentation

Release 0.2.1.2

Jesús Arroyo Torrens

February 06, 2017

| | | |
|----------|------------------------|----------|
| 1 | Contents | 3 |
| 1.1 | Installation | 3 |
| 1.2 | Quick Start | 5 |
| 1.3 | User Guide | 8 |
| 1.4 | Contribute | 28 |



Experimental open source micro-ecosystem for **open FPGAs**. Based on [platformio](#). Apio is a **multiplatform toolbox**, with static pre-built packages, project configuration tools and easy commands to verify, synthesize, simulate and upload your **verilog** designs.

Apio is used by [Icestudio](#).

1.1 Installation

Apio is written in [Python](#) and works on Linux (+ARM), Mac OS X, Windows.

Contents

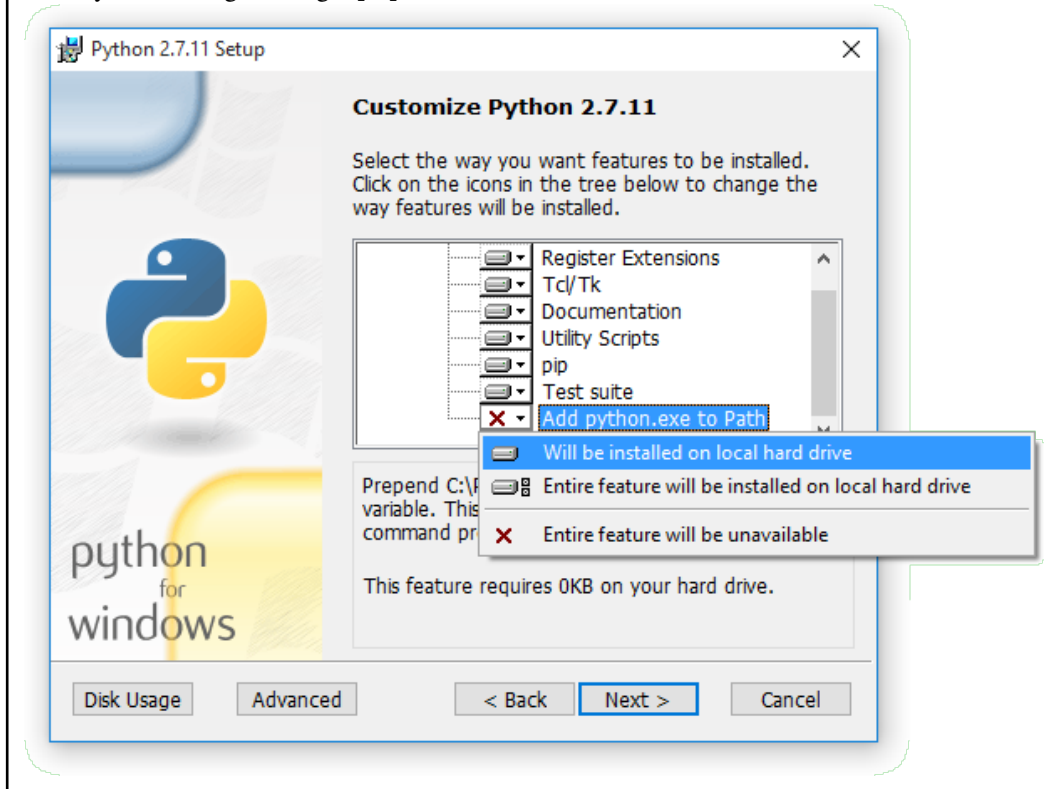
- *Installation*
 - *System requirements*
 - *Install Apio*
 - *Install FPGA FTDI drivers*
 - * *Using apio*
 - * *Manually*

1.1.1 System requirements

Operating System Linux (+ARM), Mac OS X or Windows

Python Interpreter Python 2.7

Attention: Windows Users: Please [Download the latest Python 2.7.x](#) and install it. **DON'T FORGET** to select Add python.exe to Path feature on the “Customize” stage, otherwise Python Package Manager pip command will not be available.



Terminal Application All commands below should be executed in [Command-line](#) application (Terminal). For Mac OS X and Linux OS - *Terminal* application, for Windows OS – `cmd.exe` application.

1.1.2 Install Apio

The latest stable version of Apio may be installed or upgraded via Python Package Manager ([pip](#)) as follows:

```
$ pip install -U apio
```

If `pip` command is not available run `easy_install pip`.

Note that you may run into permissions issues running these commands. You have a few options here:

- Run with `sudo` to install Apio and dependencies globally
- Specify the `pip install --user` option to install local to your user
- Run the command in a `virtualenv` local to a specific project working set.

Note: Debian users can also install the application and its packages by executing:

```
$ curl -sSL http://fpgalibre.sf.net/debian/go | sudo sh
$ sudo apt-get install apio
$ sudo apt-get install apio-scons apio-icestorm apio-iverilog apio-examples apio-system
```


1.1.3 Install FPGA FTDI drivers

Using apio

```
$ apio drivers --enable
```

To revert the FTDI drivers configuration

```
$ apio drivers --disable
```

Manually

Linux

Download [80-icestick.rules](#) and execute

```
$ sudo cp 80-icestick.rules /etc/udev/rules.d/  
$ sudo service udev restart
```

Mac OS X

Install [homebrew](#) and [libftdi](#)

```
$ brew install libftdi
```

Configure the drivers

```
$ sudo kextunload -b com.FTDI.driver.FTDIUSBSerialDriver -q  
$ sudo kextunload -b com.apple.driver.AppleUSBFTDI -q
```

To revert the drivers configuration

```
$ sudo kextload -b com.FTDI.driver.FTDIUSBSerialDriver -q  
$ sudo kextload -b com.apple.driver.AppleUSBFTDI -q
```

Windows

Go to this [tutorial](#) and follow its steps

1.2 Quick Start

Once apio has been installed and the drivers have been correctly configured is time to start playing with your FPGA!

1.2.1 Install packages

```
$ apio install --all
```

1.2.2 Create a project

Go to your project's directory or try the examples

```
$ apio examples -d leds  
$ cd leds
```

Configure your board

Find your board in the list

```
$ apio boards --list

Supported boards:

-----
Board          FPGA                      Type  Size  Pack
-----
Cat-board     iCE40-HX8K-CT256         hx    8k    ct256
go-board      iCE40-HX1K-VQ100         hx    1k    vq100
iCE40-HX8K    iCE40-HX8K-CT256         hx    8k    ct256
icestick      iCE40-HX1K-TQ144         hx    1k    tq144
icezum        iCE40-HX1K-TQ144         hx    1k    tq144
icoboard      iCE40-HX8K-CT256         hx    8k    ct256
kefir         iCE40-HX4K-TQ144         hx    8k    tq144:4k
```

Create an apio.ini file with your board

```
$ apio init --board icestick
```

1.2.3 Process the project

Verify

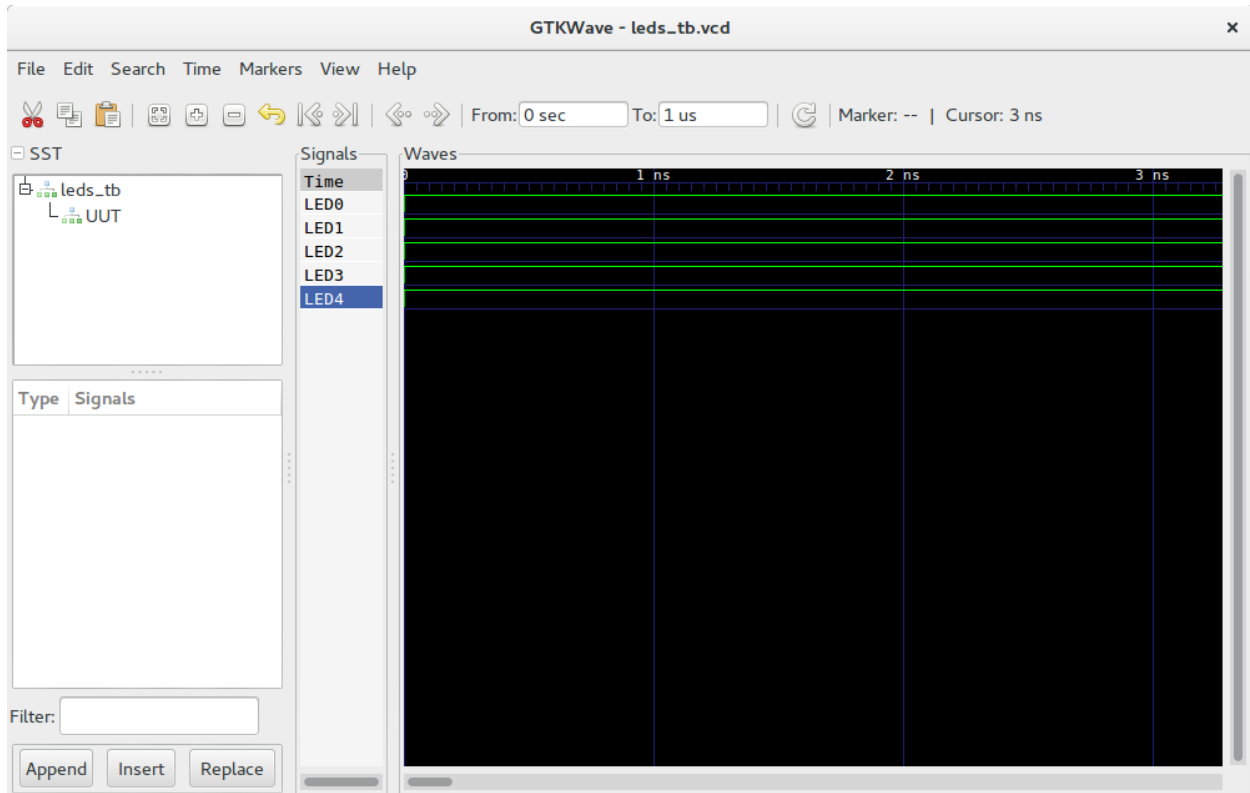
Check your verilog code using [Icarus Verilog](#)

```
$ apio verify
```

Simulate

Simulate your test bench using [Icarus Verilog](#) and [GTKWave](#)

```
$ apio sim
```



Note: GTKWave must be installed.

| | |
|---------|-------------------------|
| Debian | apt-get install gtkwave |
| Mac OSX | brew install gtkwave |
| Windows | apio install gtkwave |

Build

Synthesize your project using [Icestorm Tools](#)

```
$ apio build
```

Upload

Connect your FPGA board and upload the bitstream using [Icestorm Tools](#)

```
$ apio upload
```

All the leds should turn on after 3 seconds



Congrats! Now You have your fully open source FPGA toolchain ready!

1.3 User Guide

Contents

- *User Guide*
 - *Usage*
 - *Options*
 - *Code Commands*
 - *Environment Commands*

1.3.1 Usage

```
apio [OPTIONS] COMMAND
```

1.3.2 Options

--version

Show the version of Apio

```
$ apio --help
$ apio COMMAND --help
```

1.3.3 Code Commands

apio build

Contents

- *apio build*
 - *Usage*
 - *Description*
 - *Options*
 - *Examples*

Usage

```
apio build [OPTIONS]
```

Description

Synthesize the bitstream: generates a **bin** file from a **verilog** and a **pcf** files.

This command requires the `scons` and `icestorm` packages.

Options

-b, --board

Select a specific board.

--fpga

Select a specific FPGA.

--size `-type` `-pack`

Select a specific FPGA size, type and pack.

-p, --project-dir

Set the target directory for the project.

Note: All available boards, FPGAs, sizes, types and packs are showed in *apio boards*

Examples1. Process the *leds example*

```
$ apio build
Info: use apio.ini board: icezum
Using default SConstruct file
[] Processing icezum
-----
FPGA_SIZE: 1k
FPGA_TYPE: hx
FPGA_PACK: tq144
[ ... ]
After placement:
```

```
PIOs      3 / 96
PLBs      1 / 160
BRAMs     0 / 16

place time 0.00s
route...
pass 1, 0 shared.

After routing:
span_4    0 / 6944
span_12   2 / 1440

route time 0.01s
write_txt hardware.asc...
===== [SUCCESS] Took 0.99 seconds =====
```

apio clean

Contents

- *apio clean*
 - *Usage*
 - *Description*
 - *Options*
 - *Examples*

Usage

```
apio clean [OPTIONS]
```

Description

Clean the previous generated files: **blif**, **asc**, **bin**, **rpt** and **out**.

This command requires the `scons` package.

Options

-p, **--project-dir**

Set the target directory for the project.

Examples

1. Clean the *leds example*

```
$ apio clean
Using default SConstruct file
Removed hardware.blif
Removed hardware.asc
```

```
Removed hardware.bin
Removed hardware.out
===== [SUCCESS] Took 0.17 seconds =====
```

apio sim

Contents

- *apio sim*
 - *Usage*
 - *Description*
 - *Options*
 - *Examples*

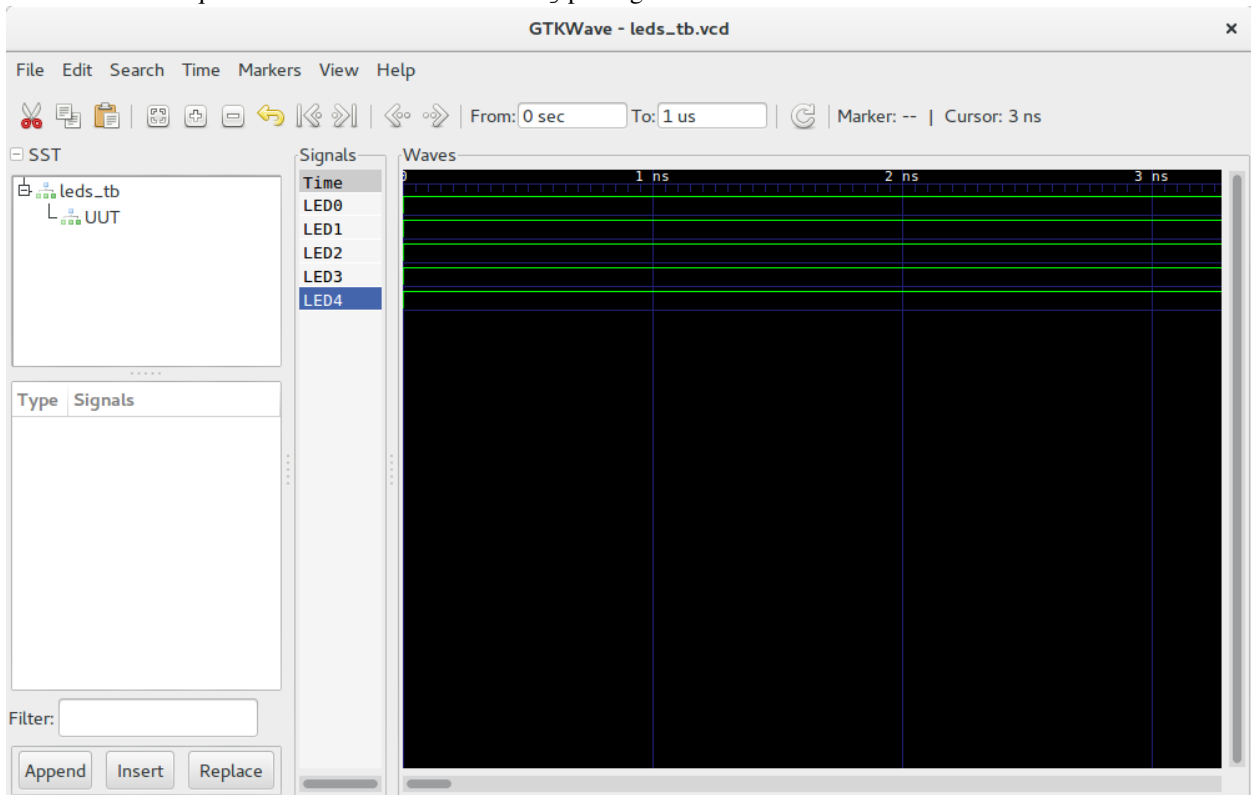
Usage

```
apio sim [OPTIONS]
```

Description

Launch the verilog simulation using [GTKWave](#) from a **verilog test bench**.

This command requires the `scons` and `iverilog` packages.



Note: GTKWave must be installed.

| | |
|---------|-------------------------|
| Debian | apt-get install gtkwave |
| Mac OSX | brew install gtkwave |
| Windows | apio install gtkwave |

Options

-p, --project-dir

Set the target directory for the project.

Examples

1. Simulate the *leds example*

```
$ apio sim
Using default SConstruct file
iverilog -B /path/to/lib/ivl -o leds_tb.out -D VCD_OUTPUT=leds_tb /path/to/vlib/system.v leds.v leds_
vvp -M /path/to/lib/ivl leds_tb.out
VCD info: dumpfile leds_tb.vcd opened for output.
End of simulation
gtkwave leds_tb.vcd leds_tb.gtkw

GTKWave Analyzer v3.3.66 (w)1999-2015 BSI

[0] start time.
[1000] end time.
WM Destroy
===== [SUCCESS] Took 1.96 seconds =====
```

apio time

Contents

- *apio time*
 - *Usage*
 - *Description*
 - *Options*
 - *Examples*

Usage

```
apio time [OPTIONS]
```

Description

Bitstream timing analysis: generates a **rpt** file with a topological timing analysis report, from a **verilog** and a **pcf** files.

This command requires the `scons` and `icestorm` packages.

Options

-b, --board

Select a specific board.

--fpga

Select a specific FPGA.

--size -type -pack

Select a specific FPGA size, type and pack.

-p, --project-dir

Set the target directory for the project.

Note: All available boards, FPGAs, sizes, types and packs are showed in *apio boards*

Examples

1. Timing analysis for the *leds example*

```
$ apio time
Info: use apio.ini board: icezum
Using default SConstruct file
[] Processing icezum
-----
FPGA_SIZE: 1k
FPGA_TYPE: hx
FPGA_PACK: tq144
[ ... ]
After placement:
PIOs      3 / 96
PLBs      1 / 160
BRAMs     0 / 16

place time 0.00s
route...
pass 1, 0 shared.

After routing:
span_4    0 / 6944
span_12   2 / 1440

route time 0.01s
write_txt hardware.asc...
// Creating timing netlist..
// Timing estimate: 0.24 ns (4161.98 MHz)
===== [SUCCESS] Took 1.10 seconds =====

$ cat hardware.rpt

icetime topological timing analysis report
```

```
=====
Warning: This timing analysis report is an estimate!
Info: max_span_hack is enabled: estimate is conservative.

Report for critical path:
-----

    pre_io_13_11_0 (PRE_IO) [clk] -> PADOUT: 0.240 ns
    0.240 ns io_pad_13_11_0_din

Total number of logic levels: 0
Total path delay: 0.24 ns (4161.98 MHz)
```

apio upload

Contents

- *apio upload*
 - *Usage*
 - *Description*
 - *Options*
 - *Examples*

Usage

```
apio upload [OPTIONS]
```

Description

Upload the bitstream to the FPGA. It builds the project if required.

It also performs an automatic discovery and validation of the FTDI chip depending on the selected board.

This command requires the `system`, `scons` and `icestorm` packages.

Note: FTDI driver configuration must be done before upload. More information in *apio drivers*.

Options

-d, --device

Select a specific device index. This numerical index is provided by `libftdi1`, that is different from `libftdi0`.

-b, --board

Select a specific board.

--fpga

Select a specific FPGA.

--size -type -pack

Select a specific FPGA size, type and pack.

-p, --project-dir

Set the target directory for the project.

Note: All available boards, FPGAs, sizes, types and packs are showed in *apio boards*

Examples

1. Upload the *leds example*

```
$ apio upload
Info: use apio.ini board: icezum
Number of FTDI devices found: 1
Checking device: 0
Manufacturer: Mareldem, Description: IceZUM Alhambra v1.1 - B01-020

Using default SConstruct file
[] Processing icezum
-----
FPGA_SIZE: 1k
FPGA_TYPE: hx
FPGA_PACK: tq144
DEVICE: 0
iceprog -d i:0x0403:0x6010:0 hardware.bin
init..
cdone: high
reset..
cdone: low
flash ID: 0x20 0xBA 0x16 0x10 0x00 0x00 0x23 0x51 0x85 0x32 0x13 0x00 0x54 0x00 0x29 0x10 0x06 0x15 0
file size: 32220
erase 64kB sector at 0x000000..
programming..
reading..
VERIFY OK
cdone: high
Bye.
===== [SUCCESS] Took 1.96 seconds =====
```

apio verify

Contents

- *apio verify*
 - *Usage*
 - *Description*
 - *Options*
 - *Examples*

Usage

```
apio verify [OPTIONS]
```

Description

Verify the **verilog** code. It is agnostic of the FPGA. It does not use the *pcf* file.

This command requires the `scons` and `iverilog` packages.

Options

-p, --project-dir

Set the target directory for the project.

Examples

1. Verify the *leds example*

```
$ apio verify
Using default SConstruct file
iverilog -B /path/to/lib/ivl -o hardware.out -D VCD_OUTPUT= /path/to/vlib/system.v leds.v
===== [SUCCESS] Took 0.17 seconds =====
```

1.3.4 Environment Commands

apio boards

Contents

- *apio boards*
 - *Usage*
 - *Description*
 - *Options*
 - *Examples*

Usage

```
apio boards [OPTIONS]
```

Description

Show FPGA boards information.

All supported boards:

- [Cat board](#)

- Nandland Go board
- iCE40-HX8K Breakout Board
- iCEstick Evaluation Kit
- IceZUM Alhambra
- icoBOARD 1.0
- Kéfir I

Note: All supported FPGAs are shown in [Project IceStorm web page](#)

Options

-l, --list

List all supported boards.

-f, --fpga

List all supported FPGAs.

Examples

1. Show all available boards

```
$ apio boards --list

Supported boards:

-----
Board          FPGA                      Type  Size  Pack
-----
Cat-board      iCE40-HX8K-CT256         hx    8k    ct256
go-board       iCE40-HX1K-VQ100        hx    1k    vq100
iCE40-HX8K     iCE40-HX8K-CT256         hx    8k    ct256
icestick       iCE40-HX1K-TQ144        hx    1k    tq144
icezum         iCE40-HX1K-TQ144        hx    1k    tq144
icoboard       iCE40-HX8K-CT256         hx    8k    ct256
kefir          iCE40-HX4K-TQ144        hx    8k    tq144:4k
```

2. Show all available FPGAs

```
$ apio boards --fpga

Supported FPGAs:

-----
FPGA                      Type  Size  Pack
-----
iCE40-HX1K-CB132          hx    1k    cb132
iCE40-HX1K-TQ144          hx    1k    tq144
iCE40-HX1K-VQ100         hx    1k    vq100
iCE40-HX4K-CB132         hx    8k    cb132:4k
iCE40-HX4K-TQ144         hx    8k    tq144:4k
```

| | | | |
|--------------------|----|----|----------|
| ice40-HX8K-CB132 | hx | 8k | cb132 |
| ice40-HX8K-CM225 | hx | 8k | cm225 |
| ice40-HX8K-CT256 | hx | 8k | ct256 |
| ice40-LP1K-CB121 | lp | 1k | cb121 |
| ice40-LP1K-CB81 | lp | 1k | cb81 |
| ice40-LP1K-CM121 | lp | 1k | cm121 |
| ice40-LP1K-CM36 | lp | 1k | cm36 |
| ice40-LP1K-CM49 | lp | 1k | cm49 |
| ice40-LP1K-CM81 | lp | 1k | cm81 |
| ice40-LP1K-QN84 | lp | 1k | qn84 |
| ice40-LP1K-SWG16TR | lp | 1k | swg16tr |
| ice40-LP4K-CM121 | lp | 8k | cm121:4k |
| ice40-LP4K-CM225 | lp | 8k | cm225:4k |
| ice40-LP4K-CM81 | lp | 8k | cm81:4k |
| ice40-LP8K-CM121 | lp | 8k | cm121 |
| ice40-LP8K-CM225 | lp | 8k | cm225 |
| ice40-LP8K-CM81 | lp | 8k | cm81 |

apio config

Contents

- *apio config*
 - *Usage*
 - *Description*
 - *Options*
 - *Examples*

Usage

```
apio config [OPTIONS]
```

Description

Apio configuration commands.

Options

-l, --list

List all configuration parameters.

-v, --verbose [0|1]

Verbose mode: *0* General, *1* Information.

-e, --exe [default|native]

Configure executables: *default* selects apio packages, *native* selects native binaries (except system package).

Note: In **debian** systems, if `/etc/apio.json` defines a new `APIO_PKG_DIR`, this new path will be used to load the packages.

| Mode | default | | native |
|-----------------------------|---------|--------|--------|
| | No | Yes | |
| <code>/etc/apio.json</code> | No | Yes | |
| Load installed packages | Yes | Yes * | No |
| Check installed packages | Yes | Yes ** | No |

* load `APIO_PKG_DIR` from `/etc/apio.json`

** Suggest message `apt-get install apio-[pkg]`

Examples

1. Show all configuration parameters

```
$ apio config --list
Executable mode: default
Verbose mode: 0
```

2. Enable native mode for executable binaries

```
$ apio config --exe native
Executable mode updated: native
```

3. Enable verbose mode 1

```
$ apio config --verbose 1
Verbose mode updated: 1
```

apio drivers

Contents

- *apio drivers*
 - *Usage*
 - *Description*
 - *Options*
 - *Examples*

Usage

```
apio drivers [OPTIONS]
```

Description

Enable/Disable the FTDI drivers.

- Linux: add the rules file. It may require a reboot or to unplug and reconnect the board.
- Mac OSX: configure `FTDIUSBSerialDriver` and `AppleUSBFTDI` keys and install `libftdi`.

- Windows: open zadig to replace the current driver by libusbK. It requires to unplug and reconnect the board. This command requires the `driver` package (only for Windows).

Note: More information in *Install FPGA FTDI drivers*

Options

-e, --enable

Enable FPGA drivers.

-d, --disable

Disable FPGA drivers.

Examples

1. Enable the FTDI drivers on Linux

```
$ apio drivers --enable
Configure FTDI drivers for FPGA
[sudo] password for user:
FPGA drivers enabled
```

1. Disable the FTDI drivers on Linux

```
$ apio drivers --disable
Revert FTDI drivers configuration
[sudo] password for user:
FPGA drivers disabled
```

apio examples

Contents

- *apio examples*
 - *Usage*
 - *Description*
 - *Options*
 - *Examples*

Usage

```
apio examples [OPTIONS]
```

Description

Manage verilog examples: <https://github.com/FPGAwards/apio-examples>

This command requires the `examples` package.

Options

-l, --list

List all available examples.

-d, --dir

Copy the selected example directory.

-f, --files

Copy the selected example files.

-p, --project-dir

Set the target directory for the examples.

-n, --sayno

Automatically answer NO to all the questions.

Examples

1. Show all available examples

```
$ apio examples --list
[ ... ]

leds
-----
Verilog example for Turning all the leds on (for the icestick/icezum boards)

wire
-----
Verilog example on how to describe a simple wire

[ ...]
```

2. Copy the *leds* example files

```
$ apio examples --files leds
Copying leds example files ...
Example files 'leds' have been successfully created!

$ ls
leds.pcf  leds_tb.gtkw  leds_tb.v  leds.v
```

3. Copy the *leds* example directory

```
$ apio examples --dir leds
Creating leds directory ...
Example 'leds' has been successfully created!

$ tree leds
leds
-- info
-- leds.pcf
```

```
-- leds_tb.gtkw
-- leds_tb.v
-- leds.v
```

apio init

Contents

- *apio init*
 - *Usage*
 - *Description*
 - *Options*
 - *Examples*

Usage

```
apio init [OPTIONS]
```

Description

Manage apio projects. In addition to the code, an apio project may include a configuration file **apio.ini** and a Scons script **SConstruct**.

Options

-s, --scons

Create a default SConstruct file. This file can be modified and it will be used instead of the default script.

-b, --board

Create a configuration file with the selected board. This will be the default board used in *apio build*, *apio time* and *apio upload* commands.

-p, --project-dir

Set the target directory for the project.

-y, --sayyes

Automatically answer YES to all the questions.

Examples

1. Create a SConstruct file.

```
$ apio init --scons
Creating SConstruct file ...
File 'SConstruct' has been successfully created!
```

2. Create an apio.ini file with the icezum board

```
$ apio init --board icezum
Creating apio.ini file ...
File 'apio.ini' has been successfully created!
```

apio install

Contents

- *apio install*
 - *Usage*
 - *Description*
 - *Options*
 - *Examples*

Usage

```
apio install [OPTIONS]
```

Description

Install packages. Automatically installs the latest version of the package. Also other versions can be installed using the following notation: **package@version**.

Available packages

| Package | Installation | Description |
|--------------------------|-----------------------|---|
| drivers | apio install drivers | Drivers tools (only for Windows) |
| examples | apio install examples | Verilog basic examples, pinouts, etc |
| gtkwave | apio install gtwave | Simulation viewer. GTKWave project (only for Windows) |
| icestorm | apio install icestorm | iCE40 FPGA synthesis, place & route and configuration tools. Icestorm project |
| iverilog | apio install iverilog | Verilog simulation and synthesis tool. Icarus Verilog project |
| scons | apio install scons | A software construction tool. Scons project |
| system | apio install system | Tools for listing the USB devices and retrieving information from the FTDI chips |

Options

-a, --all

Install all packages.

-l, --list

List all available packages.

-f, --force

Force the packages installation.

-p, --platform

Set the platform [linux_x86_64, linux_i686, linux_armv7l, linux_aarch64, windows, darwin] (Advanced).

Examples

1. Install system and scones packages:

```
$ apio install system scones
Installing system package:
Download tools-system-linux_x86_64-1.0.0.tar.gz
Downloading [#####] 100%
Unpacking [#####] 100%
Package 'system' has been successfully installed!
Installing scones package:
Download scones-2.4.1.tar.gz
Downloading [#####] 100%
Unpacking [#####] 100%
Package 'scones' has been successfully installed!
```

2. Install examples package version 0.0.8

```
$ apio install examples@0.0.8
Installing examples package:
Download apio-examples-0.0.8.zip
Downloading [#####] 100%
Unpacking [#####] 100%
Package 'examples' has been successfully installed!
```

3. Show all available packages

```
$ apio install --list

Installed packages:
-----
Name          Description          Version
-----
examples     Verilog examples    0.0.8
scones       Scones tool         2.4.1
system       System tools        1.0.0

Not installed packages:
-----
Name          Description
-----
icestorm     Icestorm toolchain
iverilog     Icarus Verilog toolchain
```

4. Install and update all packages

```
$ apio install --all
Installing examples package:
Already installed. Version 0.0.8
Installing icestorm package:
Download toolchain-icestorm-linux_x86_64-9.tar.gz
Downloading [#####] 100%
Unpacking [#####] 100%
```

```
Package 'icestorm' has been successfully installed!
Installing iverilog package:
Download toolchain-iverilog-linux_x86_64-1.0.0.tar.gz
Downloading [#####] 100%
Unpacking [#####] 100%
Package 'iverilog' has been successfully installed!
Installing scones package:
Already installed. Version 2.4.1
Installing system package:
Already installed. Version 1.0.0
```

5. Install the drivers package for windows in a linux platform

```
$ apio install drivers --platform windows
Installing drivers package:
Download tools-drivers-windows-1.0.0.tar.gz
Downloading [#####] 100%
Unpacking [#####] 100%
Package 'drivers' has been successfully installed!
```

apio system

Contents

- *apio system*
 - *Usage*
 - *Description*
 - *Options*
 - *Examples*

Usage

```
apio system [OPTIONS]
```

Description

System tools: <https://github.com/FPGAwards/tools-system>

This command requires the `system` package.

Options

--lsftdi

List all connected FTDI devices.

--lsusb

List all connected USB devices.

-i, --info

Show system information.

Examples

1. List connected FTDI devices

```
$ apio system --lsftdi
Number of FTDI devices found: 2
Checking device: 0
Manufacturer: Mareldem, Description: IceZUM Alhambra v1.1 - B01-020

Checking device: 1
Manufacturer: Lattice, Description: Lattice FTUSB Interface Cable
```

2. List connected USB devices

```
$ apio system --lsusb
1d6b:0003 (bus 3, device 1)
04ca:7049 (bus 2, device 4) path: 8
8087:0a2a (bus 2, device 3) path: 7
138a:0017 (bus 2, device 2) path: 6
0403:6010 (bus 2, device 16) path: 2
0403:6010 (bus 2, device 17) path: 1
1d6b:0002 (bus 2, device 1)
8087:8001 (bus 1, device 2) path: 1
1d6b:0002 (bus 1, device 1)
```

3. Show system information

```
$ apio system --info
Platform: linux_x86_64
```

apio uninstall

Contents

- *apio uninstall*
 - *Usage*
 - *Description*
 - *Options*
 - *Examples*

Usage

```
apio uninstall [OPTIONS]
```

Description

Uninstall packages. Before uninstalling a package, a confirmation is requested.

Available packages

| Package | Installation | Description |
|-----------------------|------------------------------------|---|
| <code>drivers</code> | <code>apio install drivers</code> | Drivers tools (only for Windows) |
| <code>examples</code> | <code>apio install examples</code> | Verilog basic examples, pinouts, etc |
| <code>gtkwave</code> | <code>apio install gtkwave</code> | Simulation viewer. GTKWave project (only for Windows) |
| <code>icestorm</code> | <code>apio install icestorm</code> | iCE40 FPGA synthesis, place & route and configuration tools. Icestorm project |
| <code>iverilog</code> | <code>apio install iverilog</code> | Verilog simulation and synthesis tool. Icarus Verilog project |
| <code>scons</code> | <code>apio install scons</code> | A software construction tool. Scons project |
| <code>system</code> | <code>apio install system</code> | Tools for listing the USB devices and retrieving information from the FTDI chips |

Options

-a, --all

Uninstall all packages.

-l, --list

List all installed packages.

-p, --platform

Set the platform [`linux_x86_64`, `linux_i686`, `linux_armv7l`, `linux_aarch64`, `windows`, `darwin`] (Advanced).

Examples

1. Uninstall `examples` package

```
$ apio uninstall examples
Do you want to continue? [y/N]: y
Uninstalling examples package
Package 'examples' has been successfully uninstalled!
```

2. Uninstall the `drivers` package for **windows** in a linux platform

```
$ apio uninstall drivers --platform windows
Do you want to continue? [y/N]: y
Uninstalling drivers package
Package 'drivers' has been successfully uninstalled!
```

apio upgrade

Contents

- *apio upgrade*
 - *Usage*
 - *Description*
 - *Examples*

Usage

```
apio upgrade [OPTIONS]
```

Description

Check latest Apio version in <https://pypi.python.org/pypi/apio>.

Examples

1. Check the Apio version

```
$ apio upgrade
You're up-to-date!
Apio 0.2.0 is currently the newest version available.
```

1.4 Contribute

1.4.1 Add a new board

In order to support a new board based on FPGA Lattice iCE40 family, follow these steps:

1. Find your FPGA in `fpgas.json`. This file contains all FPGAs supported by the Icestorm project.
2. Find or add your programmer in `programmers.json`
3. Add your board in `boards.json` with the following format:

- board key
 - fpga: FPGA key
 - programmer
 - * type: programmer key
 - * extra_args: more programmer args (optional)
 - check
 - * ftdi-desc: FTDI label description substring. Check ‘apio system –lsftdi‘
 - * platform: system platform. E.g. linux_armv7l, windows, etc. (optional)

For example:

```
"icoboard": {
  "fpga": "iCE40-HX8K-CT256",
  "programmer": {
    "type": "icoprogrammer"
  },
  "check": {
    "platform": "linux_armv7l"
  }
}
```



```
"kefir": {
  "fpga": "iCE40-HX4K-TQ144",
  "programmer": {
    "type": "iceprog",
    "extra_args": "-I B"
  },
  "check": {
    "ftdi-desc": "Milk JTAG:u"
  }
}
```

Note: For more detailed configuration [SConstruct file](#) can be edited. Also more apio packages and drivers' configuration methods can be added for a full-integration.

Symbols

- fpga
 - apio-build command line option, 9
 - apio-time command line option, 13
 - apio-upload command line option, 14
- lsftdi
 - apio-system command line option, 25
- lsusb
 - apio-system command line option, 25
- size -type -pack
 - apio-build command line option, 9
 - apio-time command line option, 13
 - apio-upload command line option, 14
- version
 - apio command line option, 8
- a, -all
 - apio-install command line option, 23
 - apio-uninstall command line option, 27
- b, -board
 - apio-build command line option, 9
 - apio-init command line option, 22
 - apio-time command line option, 13
 - apio-upload command line option, 14
- d, -device
 - apio-upload command line option, 14
- d, -dir
 - apio-examples command line option, 21
- d, -disable
 - apio-drivers command line option, 20
- e, -enable
 - apio-drivers command line option, 20
- e, -exe [default|native]
 - apio-config command line option, 18
- f, -files
 - apio-examples command line option, 21
- f, -force
 - apio-install command line option, 23
- f, -fpga
 - apio-boards command line option, 17
- i, -info
 - apio-system command line option, 25
- l, -list
 - apio-boards command line option, 17
 - apio-config command line option, 18
 - apio-examples command line option, 21
 - apio-install command line option, 23
 - apio-uninstall command line option, 27
- n, -sayno
 - apio-examples command line option, 21
- p, -platform
 - apio-install command line option, 23
 - apio-uninstall command line option, 27
- p, -project-dir
 - apio-build command line option, 9
 - apio-examples command line option, 21
 - apio-init command line option, 22
 - apio-time command line option, 13
 - apio-upload command line option, 15
 - command line option, 10, 12, 16
- s, -scons
 - apio-init command line option, 22
- v, -verbose [0|1]
 - apio-config command line option, 18
- y, -sayyes
 - apio-init command line option, 22

A

- apio command line option
 - version, 8
- apio-boards command line option
 - f, -fpga, 17
 - l, -list, 17
- apio-build command line option
 - fpga, 9
 - size -type -pack, 9
 - b, -board, 9
 - p, -project-dir, 9
- apio-config command line option
 - e, -exe [default|native], 18
 - l, -list, 18
 - v, -verbose [0|1], 18

apio-drivers command line option

- d, --disable, 20
- e, --enable, 20

apio-examples command line option

- d, --dir, 21
- f, --files, 21
- l, --list, 21
- n, --sayno, 21
- p, --project-dir, 21

apio-init command line option

- b, --board, 22
- p, --project-dir, 22
- s, --scons, 22
- y, --sayyes, 22

apio-install command line option

- a, --all, 23
- f, --force, 23
- l, --list, 23
- p, --platform, 23

apio-system command line option

- lsftdi, 25
- lsusb, 25
- i, --info, 25

apio-time command line option

- fpga, 13
- size --type --pack, 13
- b, --board, 13
- p, --project-dir, 13

apio-uninstall command line option

- a, --all, 27
- l, --list, 27
- p, --platform, 27

apio-upload command line option

- fpga, 14
- size --type --pack, 14
- b, --board, 14
- d, --device, 14
- p, --project-dir, 15

C

command line option

- p, --project-dir, 10, 12, 16