You are here: [Projects](https://rocketboards.org/foswiki/Projects/WebHome) » [OpenCV port](https://rocketboards.org/foswiki/Projects/OpenCVPort)

[OpenCV port](https://rocketboards.org/foswiki/Projects/OpenCVPort)

A port of OpenCV for Altera SoC and Arrow SoCKit boards

[01 Aug 2016 - 08:53](https://rocketboards.org/foswiki/diff/Projects/OpenCVPort?rev=18)|  Version 18 |  [Michael Daum](https://rocketboards.org/foswiki/Main/MichaelDaum) |  [Altera](https://rocketboards.org/foswiki/Projects/WebSearch#fq=tag%3AAltera), [Arrow](https://rocketboards.org/foswiki/Projects/WebSearch#fq=tag%3AArrow), [CycloneV](https://rocketboards.org/foswiki/Projects/WebSearch" \l "fq=tag%3ACycloneV), [Linux](https://rocketboards.org/foswiki/Projects/WebSearch#fq=tag%3ALinux), [SoC](https://rocketboards.org/foswiki/Projects/WebSearch" \l "fq=tag%3ASoC), [SocCKit](https://rocketboards.org/foswiki/Projects/WebSearch" \l "fq=tag%3ASocCKit)

|  |  |
| --- | --- |
| Board: | [Altera Cyclone V SoC Board](https://rocketboards.org/foswiki/Documentation/AlteraSoCDevelopmentBoard) |
| State: | running |
| Members: | [DarioPennisi](https://rocketboards.org/foswiki/Main/DarioPennisi) |
| Git: | <https://github.com/pnndra/altera-opencv/> |

Video

* [Introduction](https://rocketboards.org/foswiki/view/Projects/OpenCVPort#Introduction)
* [How to get the source and compile](https://rocketboards.org/foswiki/view/Projects/OpenCVPort#How_to_get_the_source_and_compile)
* [How to run the image](https://rocketboards.org/foswiki/view/Projects/OpenCVPort#How_to_run_the_image)
  + [Prerequisites](https://rocketboards.org/foswiki/view/Projects/OpenCVPort#Prerequisites)
  + [Getting some image displayed](https://rocketboards.org/foswiki/view/Projects/OpenCVPort#Getting_some_image_displayed)
  + [What to look for](https://rocketboards.org/foswiki/view/Projects/OpenCVPort#What_to_look_for)

Introduction

[opencv.250w.png](https://rocketboards.org/foswiki/pub/Projects/OpenCVPort/opencv.250w.png)This project is a port of OpenCV running on [Altera development board](https://rocketboards.org/foswiki/Documentation/AlteraSoCDevelopmentBoard) board (plus Bitec DVI HSMC) and [Arrow SocKit board](https://rocketboards.org/foswiki/Documentation/ArrowSoCKitEvaluationBoard). The project is based on an infrastructure that includes X11 with the matchbox window manager which provides a lightweight graphical user interface with a simple X terminal and support for USB keyboard and mouse.Video output is handled by the FPGA image provided for Altera and [SoCKit](https://rocketboards.org/foswiki/Projects/SoCKit) boards, through a custom frame buffer driver. Video input is (for now) handled via USB UVC.Kernel is based on the latest 3.10 with newest USB driver which allows connecting a USB hub with mixed full speed/high speed devices (earlier kernel builds won't work). The goal is to start from a software only port to finally get to a fully FPGA accelerated suite where individual kernels can be substituted with FPGA cores. ipTronix is currently developing a suite of IP blocks to ease implementation of hardware accelerated kernels which will range from full custom RTL to software programmed multiprocessor SIMD cores.

How to get the source and compile

1. clone the source from github - [https://github.com/pnndra/altera-opencv/](git://github.com/pnndra/altera-opencv/)
2. checkout the dario-iptronix branch
3. compile the image and u-boot

#clone the source from github

git clone git://github.com/pnndra/altera-opencv/

#go into the 'altera-opencv' directory

cd altera-opencv

#checkout current branch

git checkout dario-iptronix

#setup build environment and variables  
source ./iptronix-init  
  
#build image and bootloader (kernel will be built by iptronix-image)  
bitbake iptronix-image virtual/bootloader

How to run the image

Prerequisites

Since the kernel makes use of a custom frame buffer, in order to boot properly it's necessary to have the FPGA loaded before starting up the board ( [SoCKit.sof](https://rocketboards.org/foswiki/pub/Projects/OpenCVPort/SoCKit.sof) or [Altera.sof](https://rocketboards.org/foswiki/pub/Projects/OpenCVPort/Altera.sof)). at this time this has to be done manually through USB blaster; future builds will automatically load the FPGA from u-boot.Precompiled images can be downloaded from links below:

\* [Arrow SoCKit SD card image](https://rocketboards.org/foswiki/pub/Projects/OpenCVPort/sd_image_opencv_SoCKit.7z)

\* [Altera Eval Board SD card image](https://rocketboards.org/foswiki/pub/Projects/OpenCVPort/sd_image_yocto.zip)

Getting some image displayed

Once kernel is booted you'll see a terminal on the VGA output of [SoCKit](https://rocketboards.org/foswiki/Projects/SoCKit) boards or on DVI output of BITEC DVI board connected to HSMC of Altera boards. In case you don't want to use video output it is also possible to remote X windows display to a PC. in order to do this you either need a linux based PC running X11 (any graphical desktop should work) or a windows based PC with either Cygwin/X or any other X11 server application such as moba Xterm. on the PC side, if you use Cygwin/X or linux you have to issue (prefix it with sudo if you don't have privileges under linux):

xhost +

Note that this will enable any other PC or Xwindows application running on another PC to be displayed on your machine so this may cause security issues. it is possible to allow only a specific target (the altera board) to display applications and in this case just replace + with the IP address of the target. note also that firewall may block incoming connections so make sure you have it either disabled or set up to allow the X11 server to receive incoming connections.

On the target board you need to inform the operating system it should send its output to the host running X11 Server. assuming your X11 server has IP address 192.168.1.1 enter the following instruction (of course replace the IP address with the right one):

export DISPLAY=192.168.1.1:0.0

Once this is done any graphical application will send its output through the network to the X11 server. in order to test it just try to run xeyes, which opens a window with two eyes which follow your mouse pointer.

What to look for

OpenCV sample programs are present in the /usr/bin directory and some sample image files are present in the /usr/share/opencv/samples directory, finally data for classifiers is found under /usr/share/opencv/data Note that in addition to OpenCV this image also includes MPlayer so it will be possible to playback multimedia files or preview UVC webcam through it.

In order to test the webcam use:

mplayer -tv driver=v4l2:width=320:height=240 tv://

Note that as of today UVC webcam support is still very preliminary and with higher resolutions than 320x240 the image may get corrupted (sometimes it gets corrupted also at 320x240… this is under investigation). An extract of a few very nice examples that can be run is:

|  |  |  |
| --- | --- | --- |
| **Program** | **Description** | **Video** |
| edge | edge detection through canny algorithm |  |
| morphology2 | erode/dilate/open/close algorithms |  |
| fback | optical flow demo (requires camera) |  |
| retinaDemo | simulation of human retina behavior (requires camera) |  |
| laplace | laplacian kernels (requires camera) | [FPGA acceleration demo with real time coefficient change via IDE](http://youtu.be/8Mnqp6EXE7o) |
| video\_homography | track points of interest on a live image (requires camera) |  |
| camshiftdemo | track motion of a selected object (requires camera) |  |
| peopledetect | Find people in images (pedestrian detection) | [SoC only people detection in real time (standard opencv, 1 fps)](http://youtu.be/lrQqzxrX3o0)  [SoC only people detection in real time (iptronix optimizations,25 fps)](http://youtu.be/Sn_t-NxWDpI) |

* [Attachments (5)](https://rocketboards.org/foswiki/view/Projects/OpenCVPort)

[Show options](https://rocketboards.org/foswiki/view/Projects/OpenCVPort)

[x-office-document.png](https://rocketboards.org/foswiki/pub/Projects/OpenCVPort/Altera.sof)

[Altera.sof](https://rocketboards.org/foswiki/pub/Projects/OpenCVPort/Altera.sof?t=1615012903) (6.42 MB)

02 Sep 2013 - 08:46 | Version 1 | Dario Pennisi

Altera Eval Board FPGA image

[opencv.250w.png](https://rocketboards.org/foswiki/pub/Projects/OpenCVPort/opencv.250w.png)

[opencv.250w.png](https://rocketboards.org/foswiki/pub/Projects/OpenCVPort/opencv.250w.png?t=1615012903) (24.01 KB)

01 Aug 2016 - 08:51 | Version 1 | Michael Daum

[application-x-7z-compressed.png](https://rocketboards.org/foswiki/pub/Projects/OpenCVPort/sd_image_opencv_SoCKit.7z)

[sd\_image\_opencv\_SoCKit.7z](https://rocketboards.org/foswiki/pub/Projects/OpenCVPort/sd_image_opencv_SoCKit.7z?t=1615012903) (85.24 MB)

02 Sep 2013 - 20:56 | Version 1 | Dario Pennisi

Arrow SoCKit SD card image

[application-zip.png](https://rocketboards.org/foswiki/pub/Projects/OpenCVPort/sd_image_yocto.zip)

[sd\_image\_yocto.zip](https://rocketboards.org/foswiki/pub/Projects/OpenCVPort/sd_image_yocto.zip?t=1615012903) (147.20 MB)

31 Oct 2013 - 08:41 | Version 1 | Dario Pennisi

Altera Eval Board SD card image

[x-office-document.png](https://rocketboards.org/foswiki/pub/Projects/OpenCVPort/SoCKit.sof)

[SoCKit.sof](https://rocketboards.org/foswiki/pub/Projects/OpenCVPort/SoCKit.sof?t=1615012903) (6.42 MB)

02 Sep 2013 - 08:29 | Version 1 | Dario Pennisi

Arrow SoCKit FPGA image