OpenCV

Linux Installation Manual

Version 1.0

e-con Systems

4/6/2018

e-con Systems

Your Product Development Partner

Disclaimer

e-con Systems reserves the right to edit/modify this document without any prior intimation of whatsoever.

Contents

INTRODUCTION TO OPENCV 3

PREREQUISITES 3
DESCRIPTION 3

BUILDING OPENCV 4

STEP 1 - INSTALLING DEPENDENCIES 4
STEP 2 - CONFIGURING OPENCV 5

BUILDING SAMPLE CODE 6

TROUBLESHOOTING 7

SUPPORT 8

Introduction to OpenCV

Open Source Computer Vision Library (OpenCV) is an open source computer vision and machine learning software library. OpenCV libraries are used to communicate with Cameras. APIs introduced in the OpenCV can be supported with all e-Con systems cameras.

This document helps to install the OpenCV in Linux and build a sample code to access the camera with OpenCV.

Prerequisites

The prerequisites are as follows:

- Download OpenCV from here.
- Click clone or download option and copy the URL

```
$ git clone <OpenCV_URL>
$ cd opencv
$ git checkout <opencv_version(3.3.1 or 3.4.1)>
```

- GCC 4.4.
- CMake 2.8.7 or higher.
- GTK+2.x or higher, including headers (libgtk2.0-dev).
- Python 2.6.
- ffmpeg or libav development packages: libavcodec-dev, libavformat-dev, libswscale-dev.

Description

The following steps have been tested on Ubuntu 16.04 (Xenial xersus). OpenCV must work on any other relatively modern version of Linux OS.

Building OpenCV

The OpenCV is a sample command line application used to demonstrate some of the features of the e-con Cameras with OpenCV APIs. The steps to build OpenCV are as follows:

Step 1. Installing Dependencies
Step 2. Configuring OpenCV

Step 1 - Installing Dependencies

The below table lists the dependencies to be installed for using OpenCV.

Table 1: Installing Dependencies

Dependencies	Commands
Some general development libraries	`$ sudo apt-get install build-essential make cmake cmake-qt-gui g++`
Video4Linux Camera development libraries	`$ sudo apt-get install libv4l-dev`
OpenGL development libraries for creating graphical windows	`$ sudo apt-get install libglew-dev`
GTK development libraries for creating graphical windows	`$ sudo apt-get install libgtk2.0-dev`
Udev development libraries for accessing device information	`$ sudo apt-get install libudev-dev`
Libav video input or output development libraries	`$ sudo apt-get install libavformat-dev libavutil-dev libswscale-dev libavcodec-dev libavcodec-ffmpeg-extra56 libavformat-ffmpeg56 libavutil-ffmpeg54 libswscale-ffmpeg3 libdc1394-*`
Eigen3 math development libraries	`$ sudo apt-get install libeigen3-dev`

Step 2 - Configuring OpenCV

The steps to configure OpenCV are as follows:

1. To build and install OpenCV, you must run the following command.

```
$ mkdir sources
$ mv * sources/
```

2. Replace the **Videoio** folder with the folder provided by e-con Systems in this location with this path **OpenCV/sources/modules/**.

```
$ mkdir release
$ cd release
$ cmake -D BUILD_TIFF = ON -D WITH_CUDA = OFF -D
ENABLE_AVX = OFF -D WITH_OPENGL = OFF -D WITH_OPENCL = OFF -D
WITH_IPP = OFF -D WITH_TBB = OFF -D BUILD_TBB = OFF -D WITH_EIGEN =
OFF -D WITH_V4L = ON -D WITH_VTK = OFF -D BUILD_TESTS = OFF -D
BUILD_PERF_TESTS = OFF -D CMAKE_BUILD_TYPE = RELEASE -D
BUILD_opencv_world = ON -D CMAKE_INSTALL_PREFIX = /usr/local  <path
to the opencv source folder>
```

3. To build and install the OpenCV libraries in the location **/usr/local/lib/,** you must run the following command.

```
$ sudo make -j4 install
```

Building Sample Code

This section describes about how to build the sample code.

The steps to run sample application are as follows:

1. Download OpenCVCam sample command line application from here.
2. To build the sample code, run the following command.

```
$ sudo make
```

```
$ sudo ./OpenCVCam
```

3. Number of devices connected to the PC will be displayed just as below shown.

Troubleshooting

In this section, you can view the list of commonly occurring issues and their troubleshooting steps.

Camera Device connected, but the application does not displayed.

It seems like the camera is not properly detected by Linux, so check with the path **cd /sys/class/video4linux/** to detect the device.

CMake Error: The source directory does not appear to contain CMakeLists.txt.

1. Run cmake-gui from the terminal.
2. Provide the source and release folder path of the OpenCV project.
3. Configure and generate the solution.

Compiling command line application, libudev.so error adding symbols, DSO missing from command line.

Include the ludev dependency in the compilation command.

Camera displays in the command line but selecting the index does not end up with the preview.

Make sure the camera connected to the port starts with video0 **/sys/class/video4linux**.

/usr/bin/ld: cannot open output file OpenCVCam: permission denied. collect2: error: ld returned 1 exit status.

Provide *sudo* permission to the command.

Hid Settings is not shown in the output.

Provide **sudo** permission to the command.

```
sudo ./OpenCVCam
```

Support

Contact Us

If you need any support on OpenCV sample application, please contact us using the Live Chat option available on our website - https://www.e-consystems.com/

Creating a Ticket

If you need to create a ticket for any type of issue, please visit the ticketing page on our website - https://www.e-consystems.com/create-ticket.asp

RMA

To know about our Return Material Authorization (RMA) policy, please visit the RMA Policy page on our website - https://www.e-consystems.com/RMA-Policy.asp

General Product Warranty Terms

To know about our General Product Warranty Terms, please visit the General Warranty Terms page on our website - https://www.e-consystems.com/warranty.asp

Revision History

Rev	Date	Description	Author
1.0	10-April-2018	Initial Draft	Chandra Sekar. V