

OpenCV

# Linux Installation Manual

### **Disclaimer**

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

# Contents

<b>INTRODUCTION TO OPENCV</b>	<b>3</b>
<b>PREREQUISITES</b>	<b>3</b>
<b>DESCRIPTION</b>	<b>3</b>
<b>BUILDING OPENCV</b>	<b>4</b>
<b>STEP 1 - INSTALLING DEPENDENCIES</b>	<b>4</b>
<b>STEP 2 - CONFIGURING OPENCV</b>	<b>5</b>
<b>BUILDING SAMPLE CODE</b>	<b>6</b>
<b>TROUBLESHOOTING</b>	<b>7</b>
<b>SUPPORT</b>	<b>9</b>

# 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 (<https://github.com/opencv/opencv>)
- Click clone or download option and copy the URL

```
$ git clone <OpenCV_URL>
$ cd opencv
$ git checkout 4.7.0
```

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

## Description

The following steps have been tested on Ubuntu 18.04 (Bionic Beaver). It relatively works on other versions of Ubuntu 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	<code>\$ sudo apt-get install build-essential make cmake cmake-gui g++</code>
Video4Linux Camera development libraries	<code>\$ sudo apt-get install libv4l-dev</code>
OpenGL development libraries for creating graphical windows	<code>\$ sudo apt-get install libglew-dev</code>
GTK development libraries for creating graphical windows	<code>\$ sudo apt-get install libgtk2.0-dev</code>
Udev development libraries for accessing device information	<code>\$ sudo apt-get install libudev-dev</code>
Libav video input or output development libraries	<code>\$ sudo apt-get install libavformat-dev libavutil-dev libswscale-dev libavcodec-dev libavcodec-ffmpeg-extra56 libavformat-ffmpeg56 libavutil-ffmpeg54 libswscale-ffmpeg3 libdc1394-* libopencv-dev</code>
Eigen3 math development libraries	<code>\$ sudo apt-get install libeigen3-dev</code>

**In Ubuntu , the dependency libraries varies.**

Based on the Ubuntu versions, the development libraries will differ, for example in ubuntu 18.04 libavcodec version will be 57. Thus, install dependencies which supports the ubuntu versions.

For 20.04 OS, extra dependencies are needed. They include: pkg-config, gfortran, openexr, libatlas-base-dev, libtbb2, libopenexr-dev, libgstreamer-plugins-base1.0-dev, libgstreamer1.0-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\(https://github.com/econsystems/opencv/tree/master/Source\)](https://github.com/econsystems/opencv/tree/master/Source) with this path **OpenCV/sources/modules/**.

```
$ cd sources
$ mkdir release
$ cd release
$ cmake -D BUILD_TIFF=ON -D WITH_CUDA=OFF -D
WITH_GENERATE_PKGCONFIG=ON -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 WITH_GSTREAMER=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 commands.

```
$ sudo make -j4 && sudo make 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 (<https://github.com/econsystems/opencv/tree/master/Source>).
4. To build the sample code, run the following command.

```
$ sudo make  
$ sudo ./OpenCVCam
```

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

```
root@:/opt/OpenCV/source/opencv-3.3.1/sources/samples/cpp/tutorial_code/OpenCVCam# sudo ./OpenCVCam  
e-con's Sample OpenCV Application to Custom Formats  
Demonstrates the working of e-con's Custom Format cameras with the modified libraries of OpenCV  
Number of Camera Devices Connected to the Port : 1  
Camera Devices Connected to the PC Port :  
    0 - Exit  
    1 . See3CAM_30  
Pick a Camera Device to Explore : █
```

# 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.

Eg: `CMAKE_CXX_FLAGS="-ludev"`

## **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
```

## **Undefined reference to main:**

replace videoio module freshly ,delete opencv folders present in

`/usr/local/include/`



Rebuild opencv again.If problem still exist uncheck OpenCL,gstreamer and opencv test flags

#### **Error loading libopencv\_world.so**

```
sudo LD_LIBRARY_PATH=<path_to_the_libopencv_world.so>  
./OpenCVCam
```

## **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 Sekhar. V
1.1	3-August-18	ENABLE_AVX is deprecated	Chandra Sekhar. V
1.2	10-January-20	Removed the Device image.	Murali Mohan M
1.3	29-July-20	Added changes	Murali Mohan M
1.4	04-Nov-2020	Updated dependencies faq	M Vishnu Murali
1.5	29-July-2022	Updated document for 4.5.5 version	Nivedha Janarthanan
1.6	14-September-2023	Updated document for 4.6.0 version	Sushanth S
1.7	06-November-2023	Updated document for 4.7.0 version	Sushanth S