IMAGE DISPLAY ON HOST USING XSCOPE

This document describes the steps to be followed to execute host application on PC. The binaries are provided in the zip file.

Requirements : Linux 64/ Linux 32 bit OS
Packages : OpenCV 3.0
IDE : Eclipse C++, xTIMEcomposer studio 14

If OpenCV is installed , please skip this part.

Installing OpenCV:

OpenCV (Open Source Computer Vision Library) is an open source computer vision and machine learning software library. OpenCV was built to provide a common infrastructure for computer vision applications and to accelerate the use of machine perception in the commercial products. OpenCV Linux Installation Landing page.

Steps:
1. Install **aptitude** package manager.

```
sudo apt-get install aptitude.
```

2. Run the following commands in terminal to install packages to support opencv

```
sudo aptitude install build-essential

sudo aptitude install cmake git libgtk2.0-dev pkg-config libavcodec-dev
libavformat-dev libswscale-dev

sudo aptitude install python-dev python-numpy libtbb2 libtbb-dev libjpeg-dev
libpng-dev libtiff-dev libjasper-dev libdc1394-22-dev

sudo aptitude install libopencv-dev build-essential checkinstall cmake pkg-config
yasm libjpeg-dev libjasper-dev libavcodec-dev libavformat-dev libswscale-dev
libdc1394-22-dev libxine-dev libgstreamer0.10-dev libgstreamer-plugins-base0.10-
dev libv4l-dev python-dev python-numpy libtbb-dev libqt4-dev libgtk2.0-dev
libfaac-dev libmp3lame-dev libopencore-amrnb-dev libopencore-amrwb-dev libtheora-
dev libvorbis-dev libxvidcore-dev x264 v4l-utils ffmpeg cmake qt5-default
checkinstall
```

3. Download OpenCV installation file and extract to home folder.

4. Enter into the extracted OpenCV directory
```
cd /home/user_name/OpenCV
```

5. Create a New Directory named 'Release' an Enter into the new directory.
```
Cd ~/opencv
mkdir Release
cd Release
```

6. Build OpenCV files using CMAKE. Enter the following commands in terminal.(Inside Release folder)

```
cmake -D CMAKE_BUILD_TYPE=RELEASE -D CMAKE_INSTALL_PREFIX=/usr/local -D
WITH_TBB=ON -D BUILD_NEW_PYTHON_SUPPORT=ON -D WITH_V4L=ON -D
INSTALL_C_EXAMPLES=ON -D INSTALL_PYTHON_EXAMPLES=ON -D BUILD_EXAMPLES=ON -D
WITH_QT=ON -D WITH_OPENGL=ON -D  WITH_GTK=ON ..
```

7. If everything appears correct, enter the following to proceed.

```
make
sudo make install
```

8. To get OpenCV working properly

```
sudo /bin/bash -c 'echo "/usr/local/lib" > /etc/ld.so.conf.d/opencv.conf'
sudo ldconfig
```

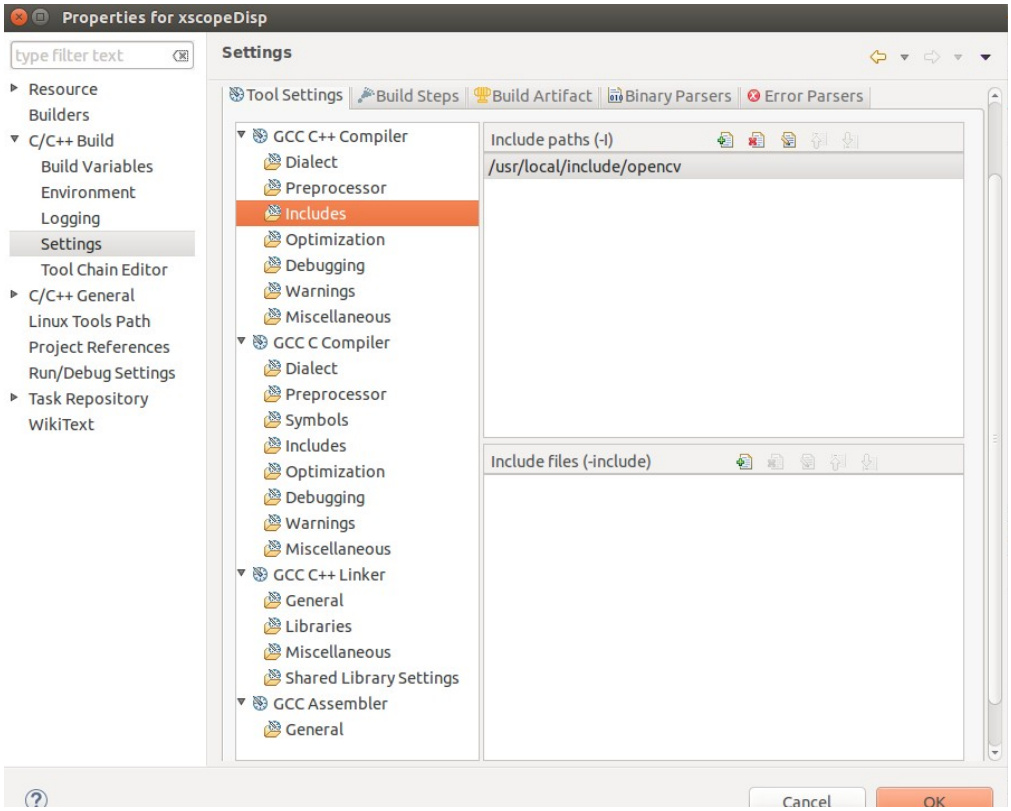
Once OpenCV is installed and configured, check whether libraries are installed by running the command and check output.

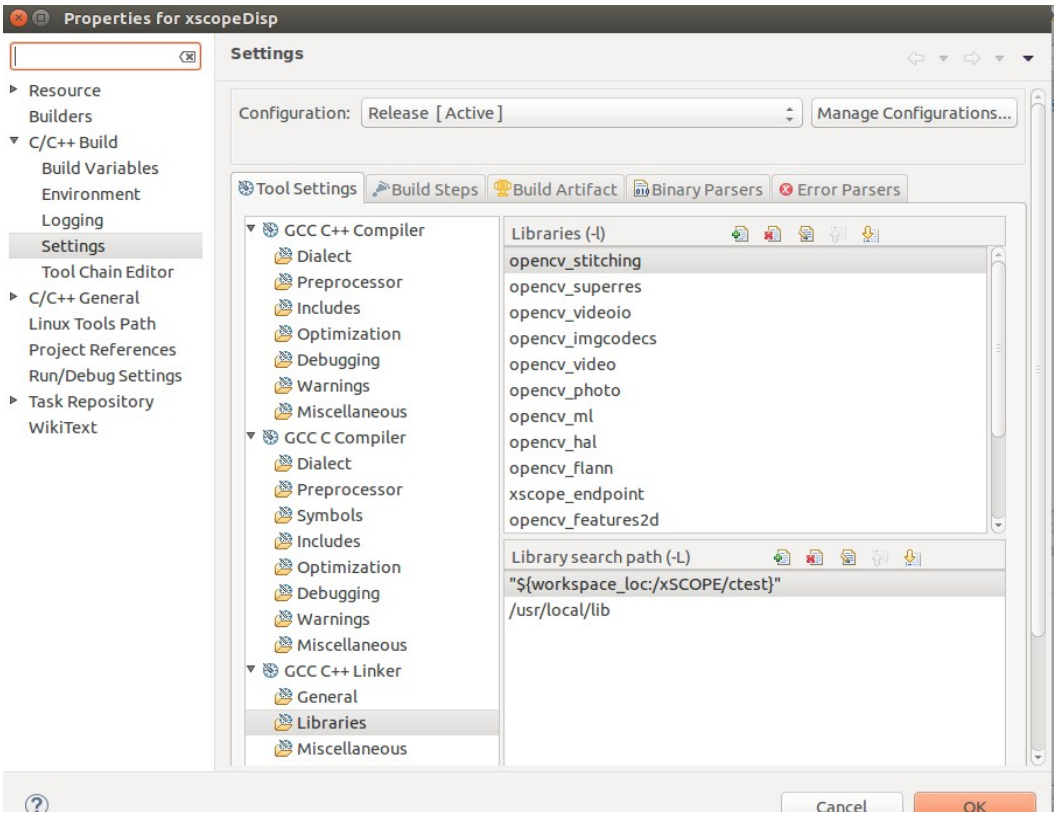
```
pkg-config --cflags opencv
output: I/usr/local/include/opencv -I/usr/local/include

pkg-config --libs opencv
Output:
-L/usr/local/lib -lopencv_core -lopencv_imgproc -lopencv_highgui -lopencv_ml
-lopencv_video -lopencv_features2d -lopencv_calib3d -lopencv_objdetect
-lopencv_contrib -lopencv_legacy -lopencv_flann
```

Running OpenCV in Eclipse C++ IDE

1. Download Eclipse C++ for Linux.
2. Run the binary.
2. Goto **File-->New--> C++ Project**
3. Enter project name. Select **Linux GCC**. Click **Next** , uncheck **Debug** select **Release**. Click **Finish**.
4. In the project explorer, Select project, Right click--> **New**-->**source folder**--> Name as **src.** For xSCOPE, copy header file xscope_endpoint.h and library xscope_endpoint.so into src.
5. Right click src-->**New**--> **source file**-->file_name.cpp
6. Wite code in the source file and save.
7. Add libraries and header files to the project
8. Right click on project-->**Properties**. In the left pane, expand **C/C++ build** select **Settings**.
9. Under **GCC C++ compiler** select **Includes** and in the right side under **Include paths(-I)**, add `/usr/local/include/opencv`
10. Under **GCC C++ linker** select **Libraries.** In the right side, Under **Library search path(-L)** add `/usr/local/lib` , To add xSCOPE library, click Add , select workspace and add the xscope_endpoint.so file.


Properties for xscopeDisp
type filter text
Resource
Builders
C/C++ Build
Build Variables
Environment
Logging
Settings
Tool Chain Editor
C/C++ General
Linux Tools Path
Project References
Run/Debug Settings
Task Repository
WikiText
Settings
Tool Settings
Build Steps
Build Artifact
Binary Parsers
Error Parsers
GCC C++ Compiler
Dialect
Preprocessor
Includes
Optimization
Debugging
Warnings
Miscellaneous
GCC C Compiler
Dialect
Preprocessor
Symbols
Includes
Optimization
Debugging
Warnings
Miscellaneous
GCC C++ Linker
General
Libraries
Miscellaneous
Shared Library Settings
GCC Assembler
General
Include paths (-I)
/usr/local/include/opencv
Include files (-include)
Cancel
OK


Properties for xscopeDisp
Resource
Builders
C/C++ Build
Build Variables
Environment
Logging
Settings
Tool Chain Editor
C/C++ General
Linux Tools Path
Project References
Run/Debug Settings
Task Repository
WikiText
Settings
Configuration:
Release [Active]
Manage Configurations...
Tool Settings
Build Steps
Build Artifact
Binary Parsers
Error Parsers
GCC C++ Compiler
Dialect
Preprocessor
Includes
Optimization
Debugging
Warnings
Miscellaneous
GCC C Compiler
Dialect
Preprocessor
Symbols
Includes
Optimization
Debugging
Warnings
Miscellaneous
GCC C++ Linker
General
Libraries
Miscellaneous
Libraries (-l)
opencv_stitching
opencv_superres
opencv_videoio
opencv_imgcodecs
opencv_video
opencv_photo
opencv_ml
opencv_hal
opencv_flann
xscope_endpoint
opencv_features2d
Library search path (-L)
"${workspace_loc:/xSCOPE/ctest}"
/usr/local/lib
Cancel
OK

11. Under **Libraries (-l)** in the top right side, add the libraries which are going to be used in the program. While entering library name, don't include 'lib' and'.so'. For ex: if library name is libopencv_core.so, enter under libraries as “opencv_core” only.
Enter the following under libraries(-l)

```
opencv_shape
opencv_highgui
opencv_imgproc
opencv_core
opencv_stitching
opencv_objdetect
opencv_superres
opencv_calib3d
opencv_features2d
opencv_videoio
opencv_imgcodecs
opencv_video
opencv_photo
opencv_ml
opencv_flann
opencv_hal
xscope_endpoint
```

12. After writing code, Build the project . Check for any errors and rectify it.
13. Run the program. Check output in console.

To Run the Binaries.
If OpenCV is installed properly in the system. The binaries can be run on the machine easily.

1. Open Terminal.
2. Enter into the host application folder.
   ```
   Cd ~/xscopeDisp
   cd Release
   ```
3. Run the binary file.
   ```
   ./xscopeDisp
   ```
4.In the console, the output would be
   ```
   init done
   opengl support available
   ```
5. Press Esc key to quit the application.