

環境構築手順書 ～Ubuntu編～

目次

開発環境について	...	1
開発環境のセットアップ	...	2
1、パッケージマネージャーの更新		
2、開発ツールのインストール		
NewTek NDI SDK v3.5のインストール	...	2
1、NewTek NDI SDK v3.5をダウンロード		
2、インストール		
OpenCV v3.2.0のインストール	...	3~4
1、画像表示ライブラリと画像フォーマットのインストール		
2、GUI表示用のライブラリのインストール		
3、ビデオストリーム及びフレームの処理に必要なライブラリのインストール		
4、OpenCVの最適化を行うライブラリのインストール		
5、GitHub上のOpenCV v3.2.0のリポジトリからクローンを行う		
6、Cmakeを実行する		
7、makeを行う		
8、OpenCVを適用させる		
RealSense SDK v2.0のインストール	...	5
1、実行用ツールのインストール		
2、開発者用ツールのインストール		
実行	...	6~7
1、実行前準備		
2、実行		
3、ソースの修正後再度実行するまで		

・開発/実行環境について

OS

Ubuntu 16.04 LTS

コンパイラ

gcc バージョン5.4.0

使用ライブラリ

Newtek NDI バージョン3.5

OpenCV バージョン3.2.0

RealSense バージョン 2.0

・開発環境のセットアップ

1、パッケージマネージャーの更新

ターミナルを起動し、root権限で以下のコマンドを打ちます。

```
# su -  
# sudo apt-get update  
# sudo apt-get upgrade
```

2、開発ツールのインストール

以下のコマンドを打ち、開発ツールをインストールします。

```
# sudo apt-get install build-essential cmake git pkg-config
```

・NewTek NDI SDK v3.5のインストール

1、NewTek NDI SDK v3.5をダウンロード

以下よりダウンロードを行います。

<https://jp.newtek.com/ndi/sdk/>

ダウンロード時に名前やメールアドレスなどの送信が必要です。

2、インストール

ダウンロードされたInstallNDISDK_v3_Linux.shを、NDIをインストールしたいフォルダにコピーします。

ターミナル上で以下のコマンドを入力します。

```
# cd “コピーしたディレクトリ”  
# ll InstallNDISDK_v3_Linux.sh  
# ./InstallNDISDK_v3_Linux.sh
```

・OpenCV v3.2.0のインストール

1、画像表示ライブラリと画像フォーマットのインストール

ターミナル上で以下のコマンドを実行します。

```
# sudo apt-get install libjpeg8-dev libtiff4-dev libjasper-dev  
libpng12-dev
```

2、GUI表示用のライブラリのインストール

ターミナル上で以下のコマンドを実行します。

```
# sudo apt-get install libgtk2.0-dev
```

3、ビデオストリーム及びフレームの処理に必要なライブラリのインストール

ターミナル上で以下のコマンドを実行します。

```
# sudo apt-get install libavcodec-dev libavformat-dev  
libswscale-dev libv4l-dev
```

4、OpenCVの最適化を行うライブラリのインストール

ターミナル上で以下のコマンドを実行します。

```
# sudo apt-get install libatlas-base-dev gfortran
```

5、GitHub上のOpenCV v3.2.0のリポジトリからクローンを行う

ターミナル上で以下のコマンドを実行します。

```
# cd ~  
# git clone https://github.com/ltseez/opencv.git  
# cd opencv  
# git checkout 3.2.0
```

```
# cd ~  
# git clone https://github.com/ltseez/opencv_contrib.git  
# cd opencv_contrib  
# git checkout 3.2.0
```

6、Cmakeを実行する

ターミナル上で以下のコマンドを実行します。

```
# cd ~/opencv
# mkdir build
# cd build
# cmake -D CMAKE_BUILD_TYPE=RELEASE ¥
-D WITH_V4L=ON ¥
-D CMAKE_INSTALL_PREFIX=/usr/local ¥
-D INSTALL_C_EXAMPLES=ON ¥
-D INSTALL_PYTHON_EXAMPLES=ON ¥
-D
OPENCV_EXTRA_MODULES_PATH=~/opencv_contrib/modules ¥
-D BUILD_EXAMPLES=ON ..
```

7、makeを行う

ターミナル上で以下のコマンドを実行します。

```
# make -j4
# sudo make install
```

8、OpenCVを適用させる

ターミナル上で以下のコマンドを実行します。

```
# echo usr/local/lib >> /etc/ld.so.conf.d/opencv.conf && cat ¥
/etc/ld.so.conf.d/opencv.conf
# ldconfig
# echo
PKG_CONFIG_PATH=$PKG_CONFIG_PATH:/usr/local/lib/pkgconfi
g >> /etc/bash.bashrc
# echo export PKG_CONFIG_PATH >> /etc/bash.bashrc && cat
/etc/bash.bashrc
```

・RealSense SDK 2.0のインストール

1、サーバーの公開鍵を登録

ターミナル上で以下のコマンドを実行します。

```
# sudo apt-key adv --keyserver keys.gnupg.net --recv-key  
C8B3A55A6F3EFCDE || sudo apt-key adv --keyserver  
hkp://keyserver.ubuntu.com:80 --recv-key  
C8B3A55A6F3EFCDE
```

2、リポジトリリストにサーバーの登録

ターミナル上で以下のコマンドを実行します。

```
# sudo add-apt-repository "deb http://realsense-hw-  
public.s3.amazonaws.com/Debian/apt-repo xenial main" -u
```

3、実行用ツールのインストール

ターミナル上で以下のコマンドを実行します。

```
# sudo apt-get install librealsense2-dkms  
# sudo apt-get install librealsense2-utils  
※インストールが成功すれば以下コマンドでRealsense Viewer  
を起動できるようになります。  
# realsense-viewer
```

4、開発者用ツールのインストール

ターミナル上で以下のコマンドを実行します。

```
# sudo apt-get install librealsense2-dev  
# sudo apt-get install librealsense2-dbg
```

・実行

1、実行前準備

ターミナル上でNDIをインストールしたディレクトリに移動します。

```
#cd “インストールしたディレクトリ”/”NDI SDK for  
Linux”/examples/C++
```

```
root@pss-VirtualBox:/home/pss# cd /home/pss/work/NDI_for_Linux/NDI_SDK_for_Linux/examples/C++/  
root@pss-VirtualBox:/home/pss/work/NDI_for_Linux/NDI_SDK_for_Linux/examples/C++#
```

以下のコマンドでC++フォルダの中にMakefile及びMakefile.projがあることを確認します。

```
#ll Makefile*
```

```
root@pss-VirtualBox:/home/pss/work/NDI_for_Linux/NDI_SDK_for_Linux/examples/C++# ll Makefile*  
-rw-rw-r-- 1 pss pss 918  8月 16  2017 Makefile  
-rw-rw-r-- 1 pss pss 692  8月 16  2017 Makefile.proj
```

Makefile.projをvimまたはテキストエディターなどで編集します。

3行目のCXXFLAGSを以下のように変更して下さい。

```
CXXFLAGS = -std=c++11 -pthread -MMD -I.././../include
```

5行目も以下のように変更して下さい。

```
LDLIBS = -lm -lndi -ldl -lpthread -lopencv_core -  
opencv_imgcodecs -opencv_highgui -opencv_videoio -  
opencv_imgproc -lrealsense2
```

```
CXXFLAGS = -std=c++11 -pthread -MMD -I.././../include
```

```
LDLAGS = -L.././../lib/$(NDILIB) -Wl,-rpath='$$ORIGIN'
```

```
LDLIBS = -lm -lndi -ldl -lpthread -lopencv_core -opencv_imgcodecs -opencv_highgui -opencv_videoio -opencv_imgproc -lrealsense2
```

続いて実行用のフォルダを作ります。

```
#mkdir NDI-Vidoe-Sender
```

Makefileを編集します。

8行目のPROJECTSを以下のように変更して下さい。

```
PROJECTS = ¥  
NDI-Video-Sender ¥
```

```
PROJECTS = \
    NDI-Video-Sender \  
.  
.PHONY: all  
all:
```


2、実行

変更後先程作ったフォルダに実行したいソース・ファイルをコピーします。

#cd “コピーしたいソースファイルのあるディレクトリ”

#cp * “インストールしたディレクトリ”/”NDI SDK for Linux”/examples/C++/ NDI-Video-Sender/

```
root@pss-VirtualBox:/home/pss/work/NDI_for_Linux/NDI SDK for Linux/examples/C++# cd /home/pss/work/source/
root@pss-VirtualBox:/home/pss/work/source# cp * /home/pss/work/NDI_for_Linux/NDI SDK for Linux/examples/C++/NDI-Video-Sender/
```

コピーが完了したら、再度C++フォルダに戻ります。

#cd “インストールしたディレクトリ”/”NDI SDK for Linux”/examples/C++

```
root@pss-VirtualBox:/home/pss/work/source# cd /home/pss/work/NDI_for_Linux/NDI SDK for Linux/examples/C++/
```

以下コマンドでMakeを実行します。

#Make

エラーが発生しなければビルド成功です。

```
root@pss-VirtualBox:/home/pss/work/NDI_for_Linux/NDI SDK for Linux/examples/C++# make
make[1]: Entering directory '/home/pss/work/NDI_for_Linux/NDI SDK for Linux/examples/C++/NDI-Video-Sender'
g++ -std=c++11 -pthread -MMO -I../include -L../lib/x86_64-linux-gnu -Wl,-rpath:'$ORIGIN' -o NDI-Video-Sender ConfigRead.cpp NdiCom.cpp NdiVideoSender.cpp -lasound -lnl -ldl -lpthread -lopencv_core -lopencv_imgcodecs -lopencv_highgui -lopencv_videoio -lopencv_imgproc
make[1]: Leaving directory '/home/pss/work/NDI_for_Linux/NDI SDK for Linux/examples/C++/NDI-Video-Sender'
make[1]: *** .PHONY:: No such file or directory. Stop.
make[1]: *** all: No such file or directory. Stop.
Makefile:12: recipe for target 'all' failed
make: [all] Error 2 (ignored)
```

ソースファイルをおいたフォルダに移動します。

#cd NDI-Video-Sender

以下コマンドで実行します。

#./NDI-Video-Sender X(Xには送信したいカメラ番号を指定)

```
root@pss-VirtualBox:/home/pss/work/NDI_for_Linux/NDI SDK for Linux/examples/C++# cd NDI-Video-Sender/
root@pss-VirtualBox:/home/pss/work/NDI_for_Linux/NDI SDK for Linux/examples/C++/NDI-Video-Sender# ./NDI-Video-Sender 1
Press <Ctrl> + <C> to exit...
Camera_1 SENDING...
```

NDI-Video-Receiverを実行する際も同様の流れ(NDI-Video-Senderの箇所をNDI-Video-Receiverに変更)となります。

3、ソースの修正後再度実行するまで

make済みのソースファイルを修正した場合、make clean後再度makeする必要があります。

```
root@pss-VirtualBox:/home/pss/work/NDI_for_Linux/NDI SDK for Linux/examples/C++# make clean
make[1]: Entering directory '/home/pss/work/NDI_for_Linux/NDI SDK for Linux/examples/C++/NDI-Video-Sender'
make[1]: Leaving directory '/home/pss/work/NDI_for_Linux/NDI SDK for Linux/examples/C++/NDI-Video-Sender'
make[1]: *** .PHONY:: No such file or directory. Stop.
make[1]: *** all: No such file or directory. Stop.
Makefile:19: recipe for target 'clean' failed
make: [clean] Error 2 (ignored)
root@pss-VirtualBox:/home/pss/work/NDI_for_Linux/NDI SDK for Linux/examples/C++# make
```