

# *CA378-AOIS for Jetson TX2*

## *ソフトウェアセットアップガイド*

*Version 1.0.3*

*Dated: 2018/01/10*

*Home Page <http://www.centuryarks.com/>*

日付	バージョン	コメント
2017/12/15	v1.0.0	新規リリース
2017/12/18	v1.0.1	「サンプルコードのビルド方法」を追記 wgetコマンドに「--no-check-certificate」を追記
2017/12/26	v1.0.2	複数カメラ接続2～5台に対応
2018/01/10	v1.0.3	インストール不具合修正

1. デモソフト環境設定
  2. ドライバのインストール
  3. デモソフトウェアインストール
  4. デモ実行方法
    4. 1. Focus & OIS デモ
    4. 2. 12M静止画撮影
    4. 3. 複数カメラ制御
  5. サンプルスクリプト
  6. サンプルコードのビルド方法
- Appendix
- A. 1. ファイル構成について
  - A. 2. 設定ファイルについて

★CA378-AOISのドライバをインストール前に以下の環境を構築を実施してください。

## 前提条件:

1. Jetpack 3.0のインストール (Linux for Tegra R27.1)
2. sudo権限の設定

## 手順1. Jetpack 3.0のインストール (Linux for Tegra R27.1)

(1) NVIDIA DEVELOPERサイトでユーザー登録します。

<https://developer.nvidia.com>

(2) Jetpack 3.0のダウンロードをします。

[https://developer.nvidia.com/embedded/jetpack-3\\_0](https://developer.nvidia.com/embedded/jetpack-3_0)

(3) 以下の10頁以降の手順に従ってインストールを行ってください。

[http://developer2.download.nvidia.com/embedded/L4T/r27\\_Release\\_v1.0/Docs/Jetson\\_X2\\_Developer\\_Kit\\_User\\_Guide.pdf?WZACmxm6jRHQtvToEuvEm4kspfaRaZJk8iX8mMEtn-YgwECKmEIn-GFEW5UPf9HIIdkALsnxIZX4qZxO43CE3crogni16PuviWZO4bLF23I12fMhJ1jxyn4qq5OZHaMImZrchFQAhDNjQln9rSK6fw0lGAfzB12fu5WXOU717PVLQW6slDJIKTg](http://developer2.download.nvidia.com/embedded/L4T/r27_Release_v1.0/Docs/Jetson_X2_Developer_Kit_User_Guide.pdf?WZACmxm6jRHQtvToEuvEm4kspfaRaZJk8iX8mMEtn-YgwECKmEIn-GFEW5UPf9HIIdkALsnxIZX4qZxO43CE3crogni16PuviWZO4bLF23I12fMhJ1jxyn4qq5OZHaMImZrchFQAhDNjQln9rSK6fw0lGAfzB12fu5WXOU717PVLQW6slDJIKTg)

## 手順2. sudo権限の設定

(1) 以下のコマンドを実行します。

```
$ sudo visudo
```

(2) 以下の赤字の行を追記します。

```
# User privilege specification
root  ALL=(ALL:ALL) ALL
nvidia ALL=(ALL:ALL) ALL

# Members of the admin group may gain root privileges
%admin ALL=(ALL) ALL

# Allow members of group sudo to execute any command
%sudo  ALL=(ALL:ALL) ALL
%nvidia ALL=(ALL:ALL) NOPASSWD: ALL
```

(3) Jetsonを再起動します。

```
$ sudo reboot
```

## 2. ドライバのインストール

### ・インストール手順

1. 以下のサイトから「CA378\_2L\_v1.0.3\_L4T27.1.tar.gz」をダウンロードします。

<https://github.com/centuryarks/CA378-AOIS/releases>

```
$ wget --no-check-certificate https://github.com/centuryarks/CA378-AOIS/releases/download/v1.0.3/CA378_2L_v1.0.3_L4T27.1.tar.gz
```

2. 「CA378\_2L\_v1.0.3\_L4T27.1.tar.gz」ファイルを解凍してください。

```
$ tar zxvf CA378_2L_v1.0.3_L4T27.1.tar.gz
```

3. 「/boot/extlinux/extlinux.conf」を編集し、以下の赤字の行を追記します。

```
$ sudo vim /boot/extlinux/extlinux.conf
```

```
TIMEOUT 30  
DEFAULT primary
```

```
MENU TITLE p2771-0000 eMMC boot options
```

```
LABEL primary
```

```
    MENU LABEL primary kernel
```

```
    LINUX /boot/Image
```

```
    FDT /boot/tegra186-quill-p3310-1000-c03-00-imx378.dtb
```

```
    APPEND fbcon=map:0 net.ifnames=0 console=tty0 OS=14t console=ttyS0,115200n8 memtype=0 video=tegrafb  
no_console_suspend=1 earlycon=uart8250,mmio32,0x03100000 gpt tegraid=18.1.2.0.0 tegra_keep_boot_clocks maxcpus=6  
android.kerntype=normal androidboot.serialno=0335115020673 vpr_resize root=/dev/mmcblk0p1 rw rootwait
```

## 2. ドライバのインストール

### 4. カメラ接続台数の設定

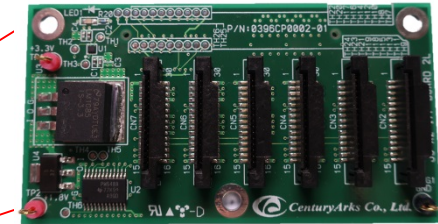
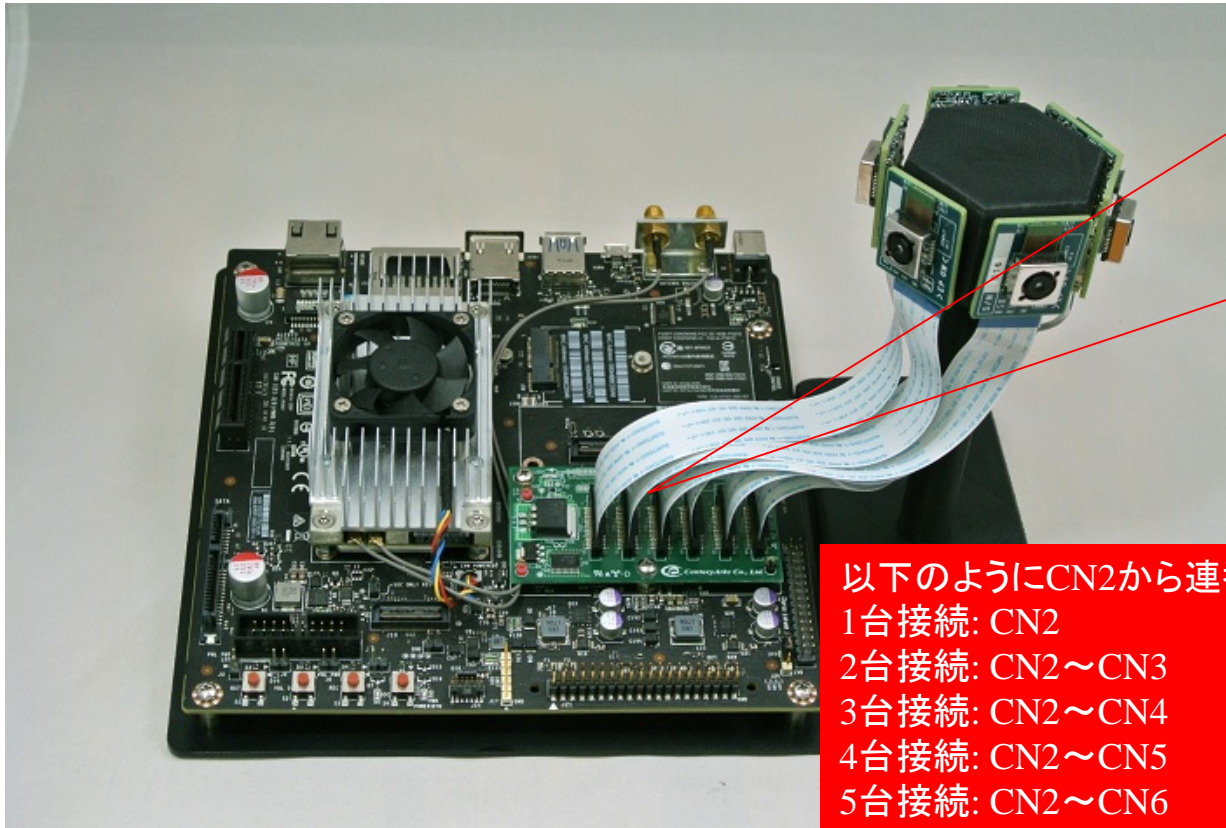
以下のコマンドを実行し、カメラ接続台数を入力します。

```
$ cd CA378_2L_v1.0.3_L4T27.1/  
$ ./Install.sh  
What is the number of camera connections? : 5  
CN2 : CH1 Camera module installed  
CN3 : CH2 Camera module installed  
CN4 : CH3 Camera module installed  
CN5 : CH4 Camera module installed  
CN6 : CH5 Camera module installed  
Cameras will be available after reboot.
```

### 5. Jetson TX2を終了し、電源をOFFにします。

## 2. ドライバのインストール

6. 以下の写真に示すようにJetson変換基板を接続します。



↑	↑	↑	↑	↑	↑
C	C	C	C	C	C
N	N	N	N	N	N
7	6	5	4	3	2

以下のようにCN2から連番になるように接続してください。

1台接続: CN2  
2台接続: CN2～CN3  
3台接続: CN2～CN4  
4台接続: CN2～CN5  
5台接続: CN2～CN6  
6台接続: CN2～CN7

7. Jetson TX2の電源をONにし、起動します。

※カメラ接続台数を変更したい場合は、手順4～7を繰り返してください。



## 2. デモソフトインストール

以下の手順でインストールを実行してください。

### ・インストール手順

1. 以下のサイトから「demo\_v1.0.0\_tx2.tar.gz」をダウンロードします。

<https://github.com/centuryarks/CA378-AOIS/releases>

```
$ wget --no-check-certificate https://github.com/centuryarks/Sample/releases/download/v1.0/demo_v1.0.0_tx2.tar.gz
```

2. 「demo\_v1.0.0\_tx2.tar.gz」ファイルを解凍してください。

```
$ tar zxvf demo_v1.0.0_tx2.tar.gz
```

3. 解凍できたフォルダ内にある「Install.sh」を実行してください。

```
$ cd demo  
$ ./Install.sh
```

4. デスクトップにショートカットが作成されます。

DEMO

# 4. デモ実行方法

- 4. 1. Focus & OIS デモ
- 4. 2. 12M静止画撮影
- 4. 3. 複数カメラ制御

# 4. 1. Focus & OIS デモ

Focus & OIS デモの起動:

1. デスクトップの「DEMO」アイコンをクリックします。
2. しばらくするとGUI画面が表示されます。
3. 被写体の距離を変更したり、カメラを動かしたりして、機能を確認してください。

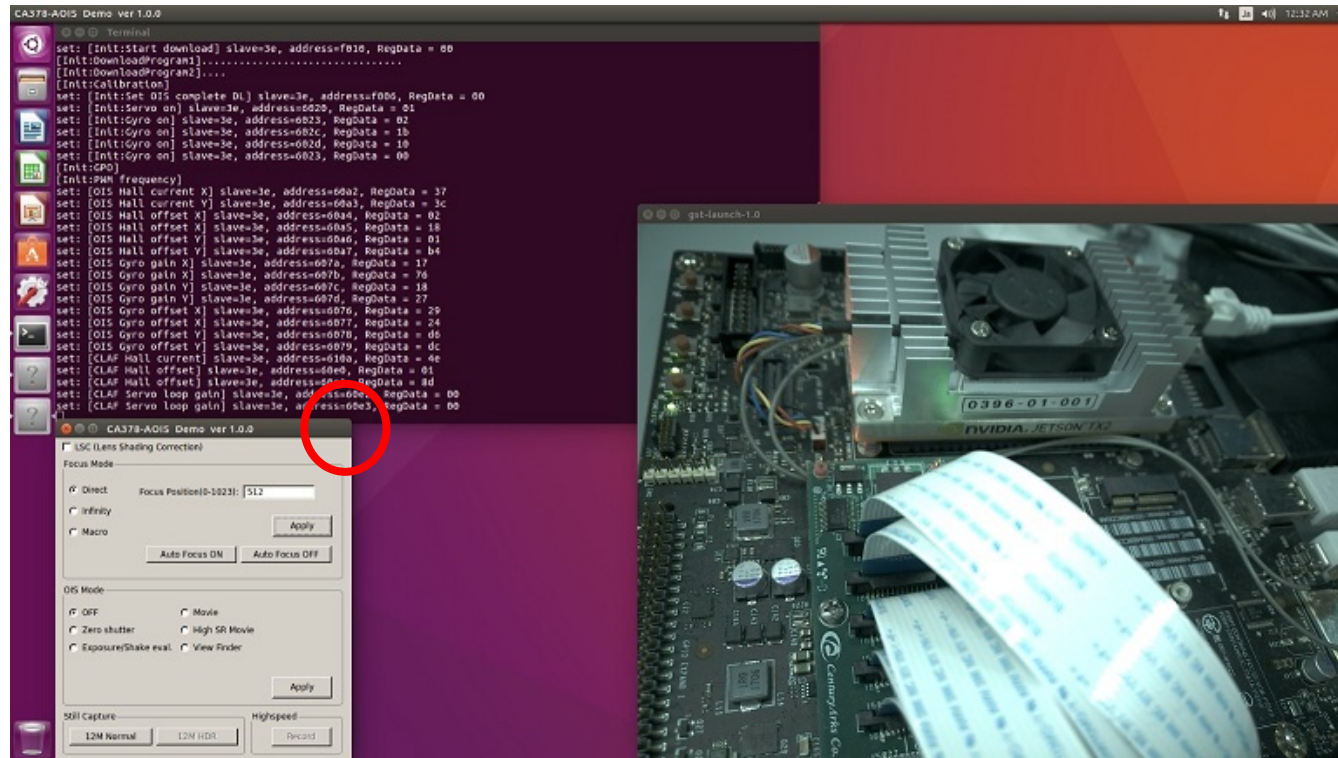
※機能詳細は、P12～P13を参照ください。



# 4. 1. Focus & OIS デモ

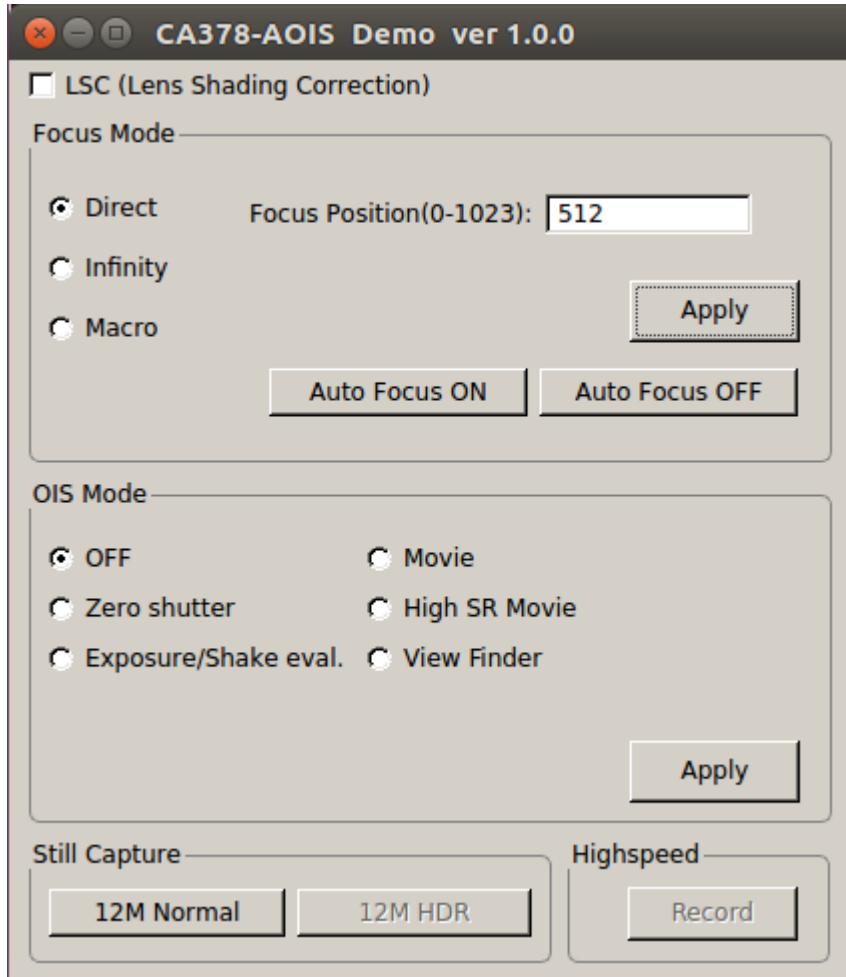
Focus & OIS デモの終了:

1. ×ボタンをクリックします。



# 4. 1. Focus & OIS デモ

Focus & OIS の各機能について説明します。



機能	説明
LSC	チェックするとシェーディング補正を有効にします ※理論値を設定してあります
Focus Mode	Direct: フォーカス位置を直接指定します Infinity: フォーカス位置を無限遠に設定します Macro: フォーカス位置を近距離に設定します Focus Position : フォーカス位置 Apply: 設定を適用します Auto Focus ON: オートフォーカスを有効にします Auto Focus OFF: オートフォーカスを無効にします ※現在デモ用にデバッグ制御されています。
OIS Mode	OFF: OISを無効にします。 各OISモードに対応します。 Zero Shutter Exposure/Shake eval. Movie High SR Movie View Finder Apply: 設定を適用します
Still Capture	12M Normal: 12M静止画撮影を行います。

# 4. 1. Focus & OIS デモ

スクリプトファイルの説明:

script/demo.shファイルについて説明します。

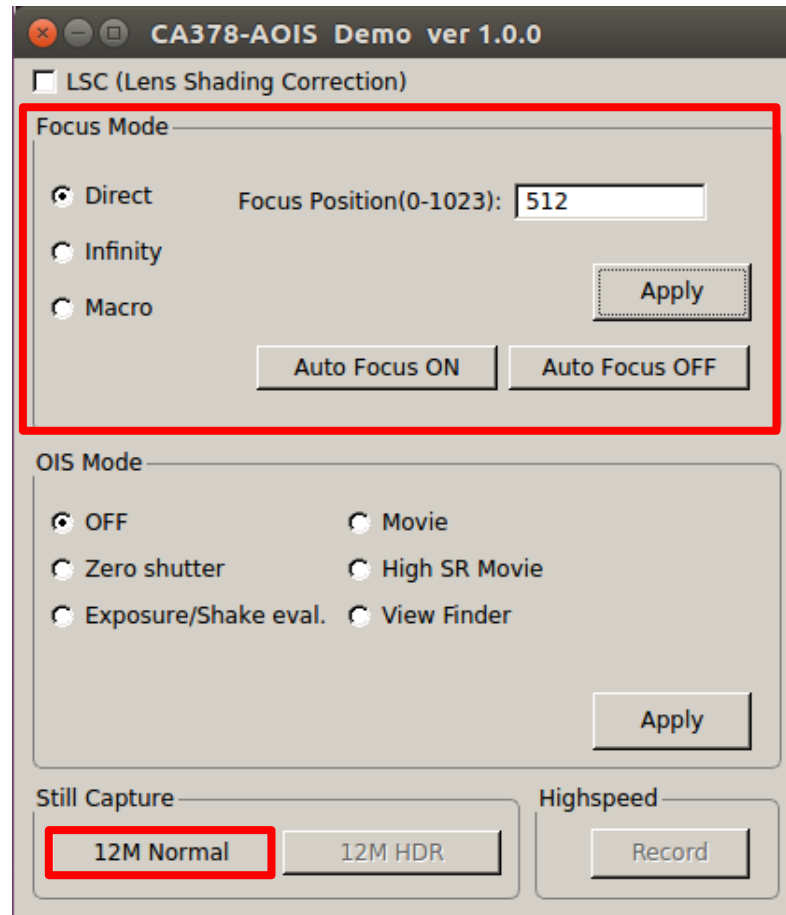
```
#!/bin/sh  
cd /home/nvidia/demo  
sudo ./bin/DemoGUI -id 0 -w 4056 -h 3040 -fps 30
```

機能	説明
-id	カメラモジュールが6台接続されている場合は0～5を指定します。
-w	幅を指定します。
-h	高さを指定します。
-fps	フレームレートを指定します。
	解像度とフレームレートの組み合わせは以下となります。 4056 x 3040 @ 30fps 3840 x 2160 @ 30fps 1920 x 1080 @ 30fps 1920 x 1080 @ 60fps 1920 x 1080 @ 120fps  モードを追加する場合はscript/preview.shを編集します。

## 4. 2. 12M静止画撮影

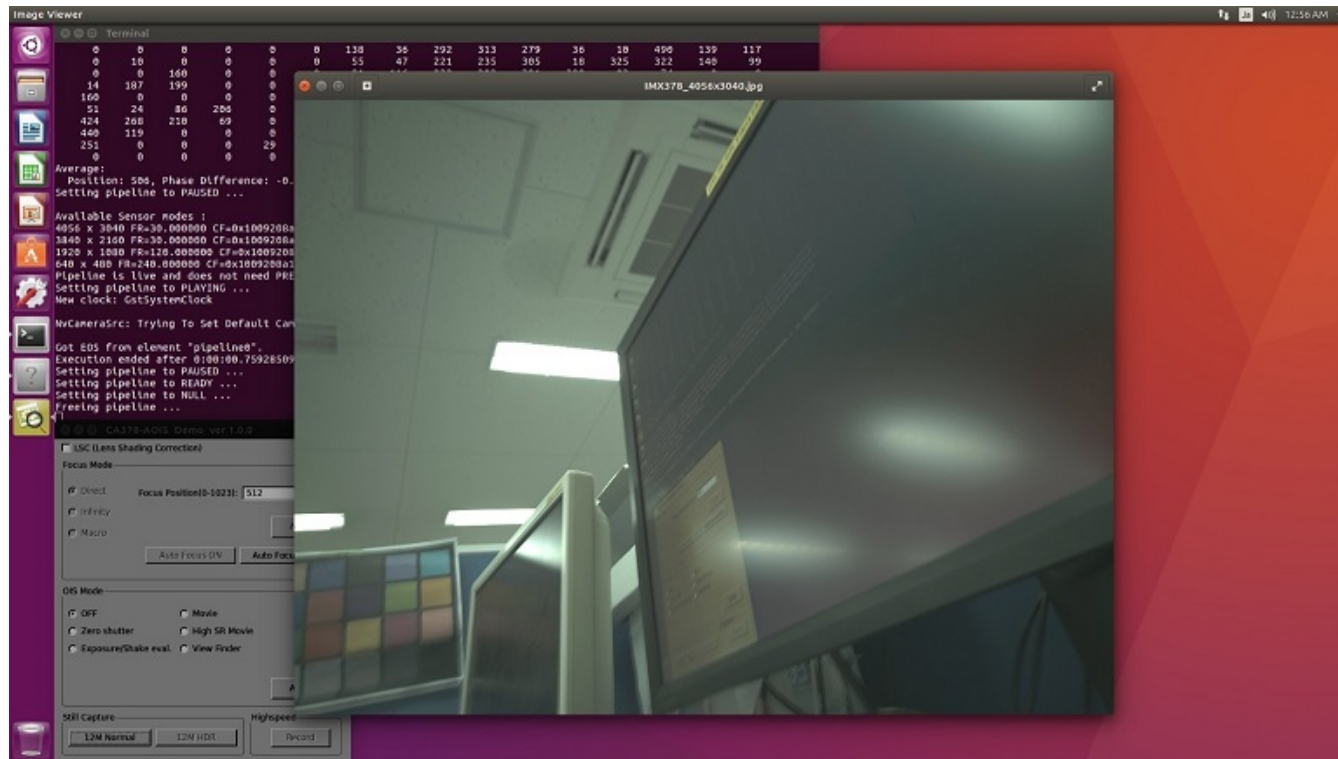
12M静止画撮影の実行:

1. フォーカスの調整をしておきます。  
(Auto Focus ONにし、焦点が合ったらAuto Focus OFFにすると便利です)
2. [12M Normal] ボタンをクリックします。



# 4. 2. 12M静止画撮影

3. JPEGとYUVフォーマットで撮像が可能です。  
(RAWフォーマットは近々対応予定です。)

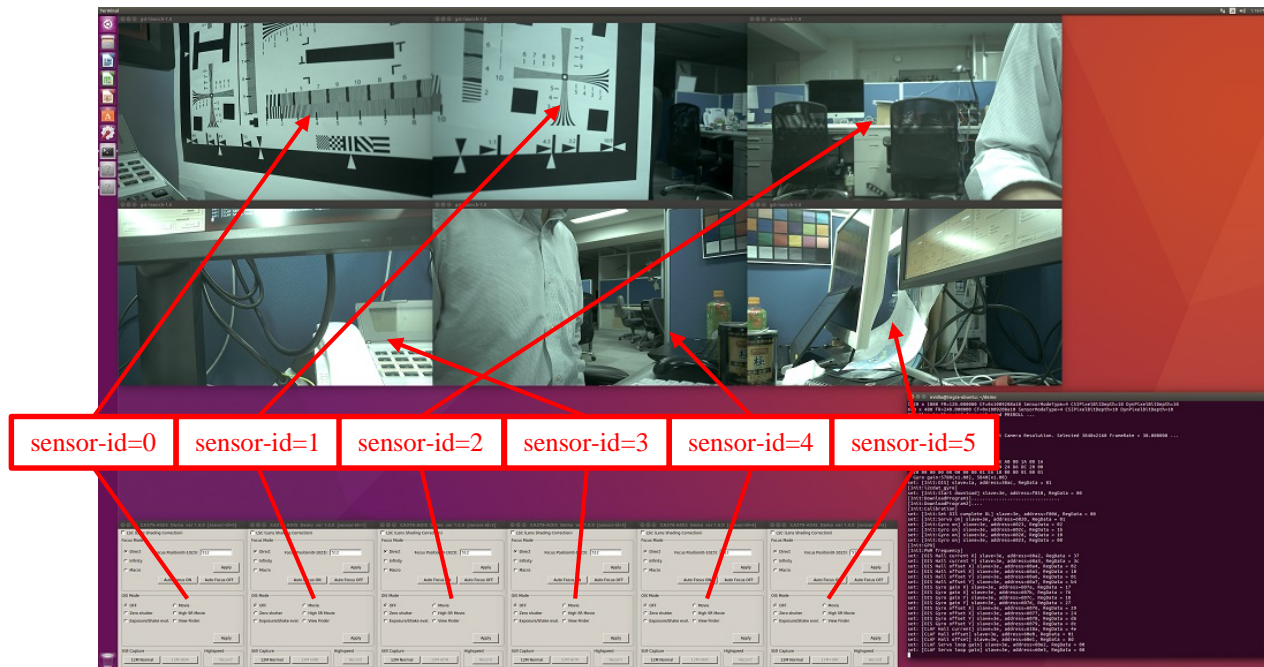




# 4. 3. 複数カメラ制御

しばらく時間がかかりますが、以下のコマンドで一度ずつ起動してください。

```
$ cd demo  
$ ./script/demo.sh 0 &  
$ ./script/demo.sh 1 &  
$ ./script/demo.sh 2 &  
$ ./script/demo.sh 3 &  
$ ./script/demo.sh 4 &  
$ ./script/demo.sh 5 &
```



Jetson TX2(1.4Gピクセル/秒)の処理能力を超えるため、環境や条件によっては、6台のカメラを表示することができない場合があります。

# 5. サンプルスクリプト

## 1. カメラ表示

以下のコマンドでカメラの映像を画面に映し出すことができます。

```
$ gst-launch-1.0 nvcamerasrc ! 'video/x-raw(memory:NVMM),width=4056,height=3040,framerate=(fraction)30/1' ! nvvidconv ! nvoverlaysink -e  
$ gst-launch-1.0 nvcamerasrc ! 'video/x-raw(memory:NVMM),width=3840,height=2160,framerate=(fraction)30/1' ! nvvidconv ! nvoverlaysink -e  
$ gst-launch-1.0 nvcamerasrc ! 'video/x-raw(memory:NVMM),width=1920,height=1080,framerate=(fraction)120/1' ! nvvidconv ! nvoverlaysink -e
```

## 2. JPEG静止画保存

以下のコマンドでJPEG静止画を保存することができます。

```
$ gst-launch-1.0 nvcamerasrc num-buffers=1 ! 'video/x-raw(memory:NVMM),width=3840,height=2160,format=I420,framerate=30/1' !  
! nvvidconv ! nvjpegenc ! filesink location=test.jpg -e
```

## 3. H.264動画保存

以下のコマンドでH.264動画を保存することができます。(最大60fps)

```
$ gst-launch-1.0 nvcamerasrc num-buffers=100 ! 'video/x-raw(memory:NVMM),width=1920,height=1080,format=I420,framerate=60/1' !  
! nvvidconv ! omxh264enc ! qtmux ! filesink location=test.mp4 -e
```

## 4. YUV動画保存

以下のコマンドでYUV動画を保存することができます。

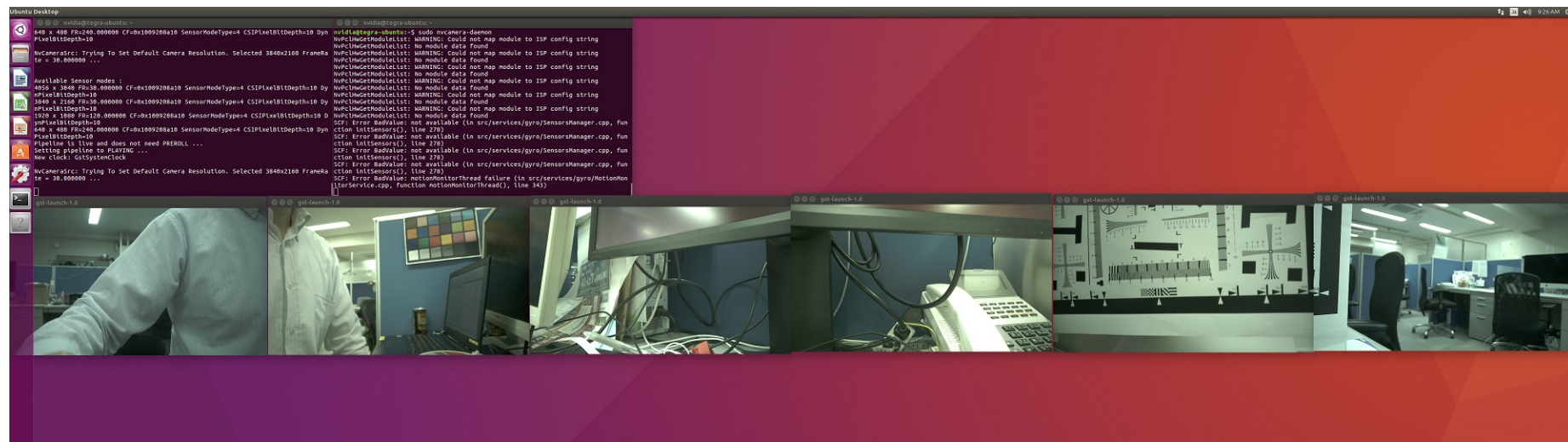
```
$ gst-launch-1.0 nvcamerasrc num-buffers=200 ! 'video/x-raw(memory:NVMM),width=1920,height=1080,format=I420,framerate=120/1' !  
! nvvidconv ! 'video/x-raw,width=1920,height=1080,format=I420,framerate=120/1' ! filesink location=test.yuv -e
```

# 5. サンプルスクリプト

## 5. 複数カメラ表示

以下のコマンドで複数カメラの映像を画面に映し出すことができます。

```
gst-launch-1.0 nvcamerasrc sensor-id=0 ! 'video/x-raw(memory:NVMM), width=(int)3840, height=(int)2160, framerate=(fraction)30/1' ! \
nvvidconv flip-method=2 ! 'video/x-raw, width=(int)640, height=(int)360, framerate=(fraction)30/1' ! queue ! xvimagesink \
nvcamerasrc sensor-id=1 ! 'video/x-raw(memory:NVMM), width=(int)3840, height=(int)2160, framerate=(fraction)30/1' ! \
nvvidconv flip-method=2 ! 'video/x-raw, width=(int)640, height=(int)360, framerate=(fraction)30/1' ! queue ! xvimagesink \
nvcamerasrc sensor-id=2 ! 'video/x-raw(memory:NVMM), width=(int)3840, height=(int)2160, framerate=(fraction)30/1' ! \
nvvidconv flip-method=2 ! 'video/x-raw, width=(int)640, height=(int)360, framerate=(fraction)30/1' ! queue ! xvimagesink \
nvcamerasrc sensor-id=3 ! 'video/x-raw(memory:NVMM), width=(int)3840, height=(int)2160, framerate=(fraction)30/1' ! \
nvvidconv flip-method=2 ! 'video/x-raw, width=(int)640, height=(int)360, framerate=(fraction)30/1' ! queue ! xvimagesink \
nvcamerasrc sensor-id=4 ! 'video/x-raw(memory:NVMM), width=(int)3840, height=(int)2160, framerate=(fraction)30/1' ! \
nvvidconv flip-method=2 ! 'video/x-raw, width=(int)640, height=(int)360, framerate=(fraction)30/1' ! queue ! xvimagesink \
nvcamerasrc sensor-id=5 ! 'video/x-raw(memory:NVMM), width=(int)3840, height=(int)2160, framerate=(fraction)30/1' ! \
nvvidconv flip-method=2 ! 'video/x-raw, width=(int)640, height=(int)360, framerate=(fraction)30/1' ! queue ! xvimagesink
```



Jetson TX2 (1.4Gピクセル/秒) の処理能力を超えるため、環境や条件によっては、6台のカメラを表示することができない場合があります。

# 6. サンプルコードのビルド方法

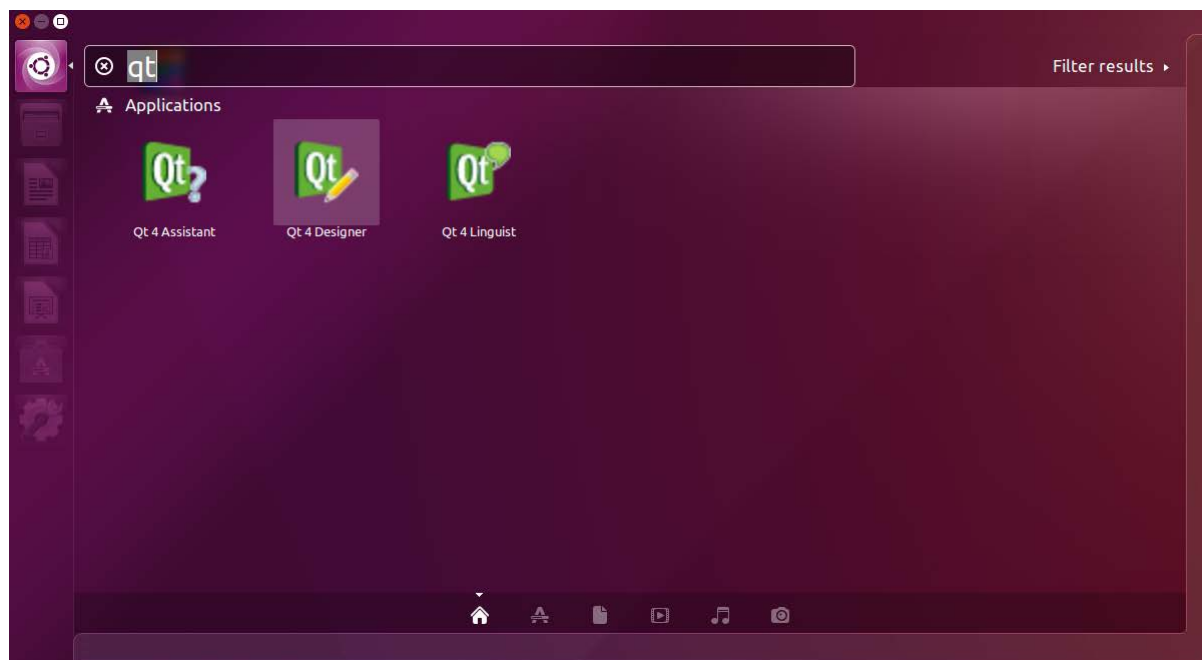
## 1. ビルド環境のインストール

以下のコマンドでビルド環境をインストールします。

```
$ sudo apt-get install qt4-dev-tools
```

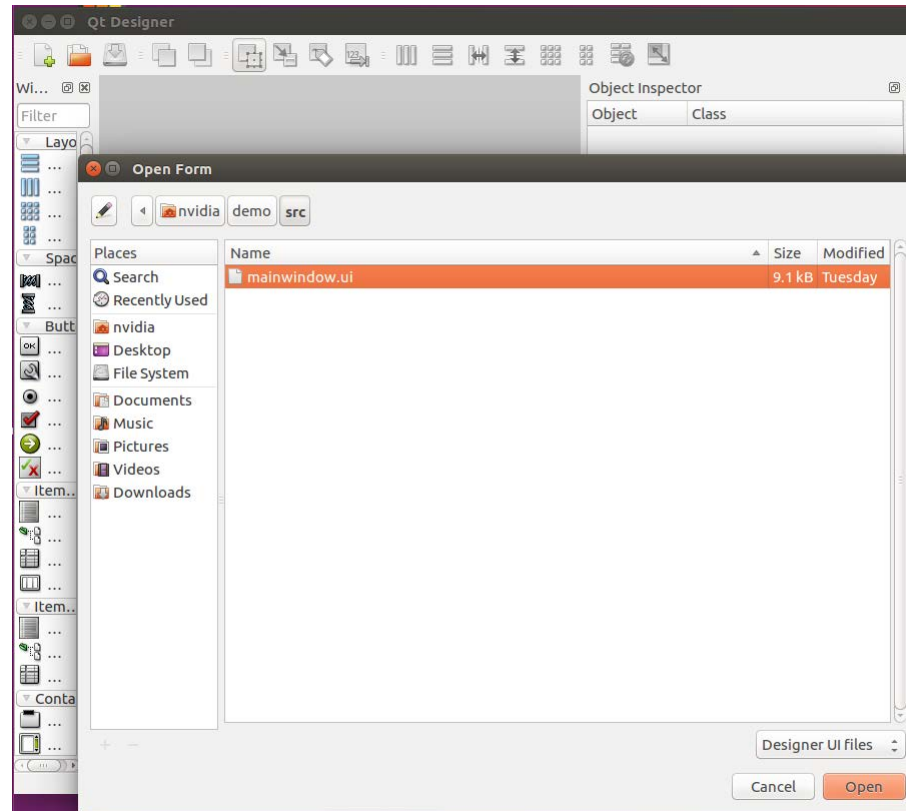
## 2. GUI デザイン

(1) 「qt」とサーチ欄に入力し、「Qt 4 Designer」を選択します。



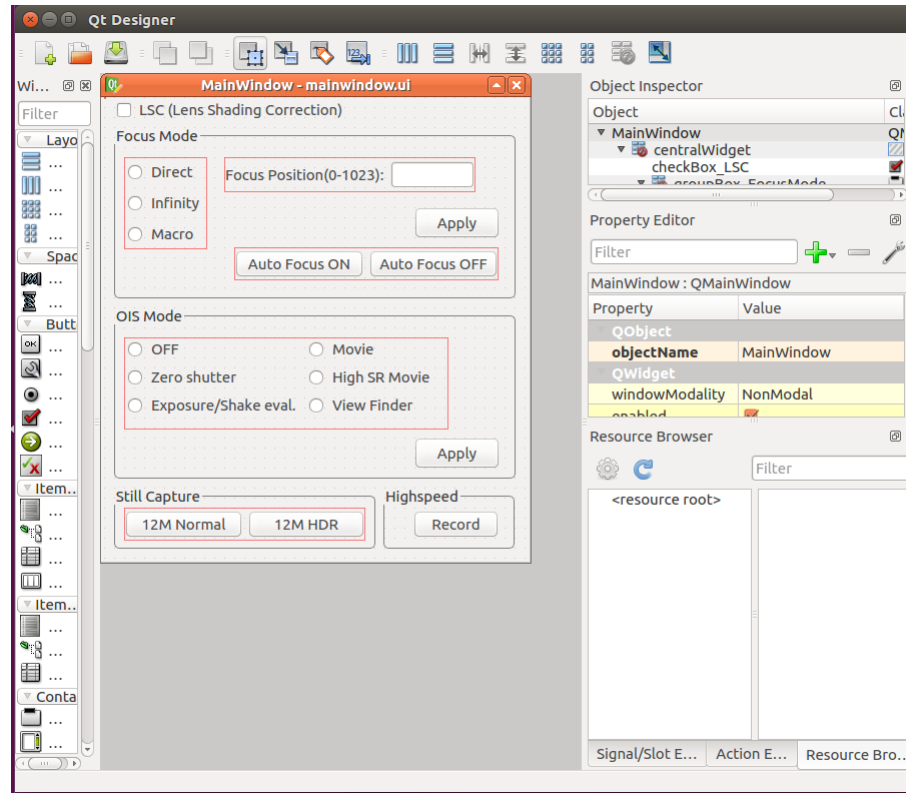
# 6. サンプルコードのビルド方法

(2) 「Open...」をクリック、「demo/src/mainwindow.ui」を選択して開きます。



# 6. サンプルコードのビルド方法

(3) デザインを変更し保存します。



## 3. ビルド

以下のコマンドでビルドします。

```
$ qmake  
$ make
```

# Appedix

# A. 1. ファイル構成について

デモソフトのファイル構成について説明します。

```
demo
├── Install.sh
├── bin
│   ├── demo.ini
│   └── DemoGUI
├── script
│   ├── demo.sh
│   ├── preview.sh
│   └── stillCapture12M_Normal.sh
└── src
    ├── af_control.c
    ├── af_control.h
    ├── communication.h
    ├── communication_jetson.c
    ├── debug_util.h
    ├── DemoGUI.pro
    ├── DemoGUI.pro.user
    ├── demo_control.c
    ├── demo_control.h
    ├── lsc_control.c
    ├── lsc_control.h
    ├── main.cpp
    ├── mainwindow.cpp
    ├── mainwindow.h
    ├── mainwindow.ui
    ├── Makefile
    ├── ois_control.c
    ├── ois_control.h
    ├── slave_address.h
    └── types_util.h
```

機能	説明
bin	DemoGUI: デモソフト Demo.ini: デモソフトの設定ファイル
script	スクリプトが記述されています。 仕様に応じてカスタマイズ可能です。 demo.sh preview.sh stillCapture12M_Normal.sh
src	デモソフトのソースコード一式です。



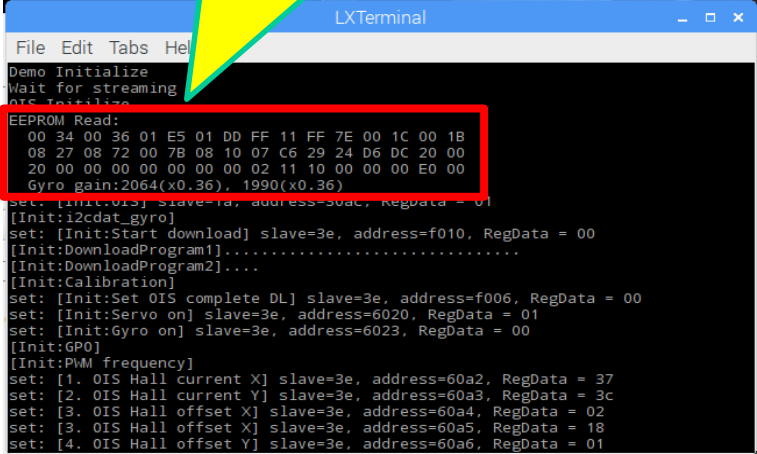
# A. 2. 設定ファイルについて

設定ファイルのdemo.iniについて説明します。

```
# DEMO Setting
preview = /home/pi/demo/script/preview.sh
stillCapture12M_Normal = /home/pi/demo/script/stillCapture12M_Normal.sh
gyroGainRateX=1.00
gyroGainRateY=1.00
autoFocusGain=2.0
autoFocusConfidenceThreshold=10
autoFocusMoveLimit=100
AutoFocusAverageNum=1
```

機能	説明
preview	プレビュー用のスクリプトPathです。
stillCapture12M_Normal	12M静止画撮影用のスクリプトPathです。
gyroGainRateX gyroGainRateY	EEPROMにOISキャリブレーション結果が書き込まれていた場合のみ有効です。
autoFocusGain	オートフォーカスのゲインを調整します。
autoFocusConfidenceThreshold	Phase Differenceの信頼度の閾値を指定します。
autoFocusMoveLimit	一度のフォーカス移動量を制限します。
AutoFocusAverageNum	オートフォーカスの平均量を調整します。

OISキャリブレーション済みの場合は、Terminalに以下のログが出力されます。



```
LXTerminal
File Edit Tabs Help
Demo Initialize
Wait for streaming
OIS Initialize
EEPROM Read:
00 34 00 36 01 E5 01 DD FF 11 FF 7E 00 1C 00 1B
08 27 08 72 00 78 08 10 07 C6 29 24 D6 DC 20 00
20 00 00 00 00 00 00 02 11 10 00 00 00 E0 00
Gyro gain:2064(x0.36), 1990(x0.36)
set: [Init:OIS] slave=3e, address=304c, RegData = 01
[Init:i2cdat_gyro]
set: [Init:Start download] slave=3e, address=f010, RegData = 00
[Init:DownloadProgram1].....
[Init:DownloadProgram2]....
[Init:Calibration]
set: [Init:Set OIS complete DL] slave=3e, address=f006, RegData = 00
set: [Init:Servo on] slave=3e, address=6020, RegData = 01
set: [Init:Gyro on] slave=3e, address=6023, RegData = 00
[Init:GPO]
[Init:PWM frequency]
set: [1. OIS Hall current X] slave=3e, address=60a2, RegData = 37
set: [2. OIS Hall current Y] slave=3e, address=60a3, RegData = 3c
set: [3. OIS Hall offset X] slave=3e, address=60a4, RegData = 02
set: [3. OIS Hall offset X] slave=3e, address=60a5, RegData = 18
set: [4. OIS Hall offset Y] slave=3e, address=60a6, RegData = 01
```