

# *CA378-AOIS for TinkerBoard* *ソフトウェアセットアップガイド*

*Version 1.0.1*

*Dated: 2019/12/04*

*Home Page* <https://www.centuryarks.com/products/sensor/cm>

日付	バージョン	コメント
2019/10/25	v1.0.0	新規リリース
2019/12/04	v1.0.1	ISPファイル更新、ストリーミング画像反転、画サイズ変更

1. 概要
2. デモソフト環境設定
3. ドライバのインストール
4. デモソフトウェアインストール
5. デモ実行方法
  5. 1. Focus & OIS デモ
  5. 2. 12M静止画撮影
  5. 3. 静止画撮影
  5. 4. 動画撮影
  5. 5. デモソフトウェアビルド
  5. 6. ストリーミング画像反転

## 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

本資料は、TinkerBoard上でカーネルドライバのビルド手順とソフトウェアのインストールについて説明します。

Hardware :TinkerBoard

OS :TinkerOS Debian9 v2.0.8

CSI Hardware :CenturyArks CA378-AOIS(Sony IMX378)

この資料についての問い合わせは、以下にメールしてください。

[ca-qa@centuryarks.com](mailto:ca-qa@centuryarks.com)

## 2. デモソフト環境設定

2種類の環境構築手順があります。

- 1:カーネル、ドライバ、デモソフトがインストールされたTinkerOSイメージを使用する手順
- 2:カーネル、ドライバ、デモソフトのマニュアルインストール手順

2.1. カーネル、ドライバ、デモソフトがインストールされたTinkerOSイメージを使用する手順

OS:TinkerOS Debian9 v2.0.8(Kernel v4.4.132)

Demo soft:v1.0.1

(1-1)以下のイメージをダウンロードしてください。

[https://www.centuryarks.com/images/product/sensor/2019-12-04-tinkeros-v1.0.1\\_CA378-AOIS.zip](https://www.centuryarks.com/images/product/sensor/2019-12-04-tinkeros-v1.0.1_CA378-AOIS.zip)

(1-2)イメージファイルをSDカードへ書き込んでください。Windowsであれば「win32diskimage」、Linuxであれば「balenaEtcher」等のアプリを使用します。

(1-3) Tinker boardを起動します。

ID:linaro

PASSWORD:linaro

(1-4)このイメージファイルはカーネル、ドライバ、デモソフトが含まれているため、4章までの手順は不要です。

## 2. デモソフト環境設定

### 2.2. カーネル、ドライバ、デモソフトのマニュアルインストール手順

(2-1) TinkerOSのイメージをダウンロードし、SDカードへ書き込みます。

[https://github.com/TinkerBoard/debian\\_kernel/releases/download/2.0.8/20181023-tinker-board-linaro-stretch-alip-v2.0.8.img.zip](https://github.com/TinkerBoard/debian_kernel/releases/download/2.0.8/20181023-tinker-board-linaro-stretch-alip-v2.0.8.img.zip)

(2-2) Tinker boardを起動します。

ID:linaro

PASSWORD:linaro

# 3. ドライバのインストール

## ・手順概要

1. カーネルソースの準備
2. カーネルのビルドとインストール

## 手順1. カーネルソースの準備

以下のサイトからファイルをTinkerBoardのホームディレクトリへダウンロードし、以下のコマンドを実行します。

[https://github.com/centuryarks/CA378-AOIS/releases/download/TINKER\\_v1.0.1\\_v4.4.132\(Debian9\\_v2.0.8\)/CA378\\_v1.0.1\\_TinkerOS\\_Debian\\_v2.0.8\\_src\\_build.tar.gz](https://github.com/centuryarks/CA378-AOIS/releases/download/TINKER_v1.0.1_v4.4.132(Debian9_v2.0.8)/CA378_v1.0.1_TinkerOS_Debian_v2.0.8_src_build.tar.gz)

```
# tar -zxvf CA378_v1.0.1_TinkerOS_Debian9_v2.0.8_src_build.tar.gz
# cd CA378_v1.0.1_TinkerOS_Debian9_v2.0.8_src_build
# ./PrepareKernelSources.sh
```

## 手順2. カーネルのビルドとインストール

以下のコマンドでカーネルをビルド、インストールします。

```
# ./BuildKernelSources.sh
```

## 再起動

```
# sudo reboot
```

# 4. 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\_v1.0.1\_tinker.tar.gz」をダウンロードします。

[https://github.com/centuryarks/Sample/releases/download/TINKER\\_v1.0.1\\_v4.4.132\(Debian9\\_v2.0.8\)/demo\\_v1.0.1\\_tinker.tar.gz](https://github.com/centuryarks/Sample/releases/download/TINKER_v1.0.1_v4.4.132(Debian9_v2.0.8)/demo_v1.0.1_tinker.tar.gz)

```
# wget --no-check-certificate ¥  
"https://github.com/centuryarks/Sample/releases/download/TINKER_v1.0.1_v4.4.132(Debian9_v2.0.8)/demo_v1.0.1_tinker.tar.gz"
```

2. 「demo\_v1.0.1\_tinker.tar.gz」ファイルを解凍してください。

```
# tar -zxvf demo_v1.0.1_tinker.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





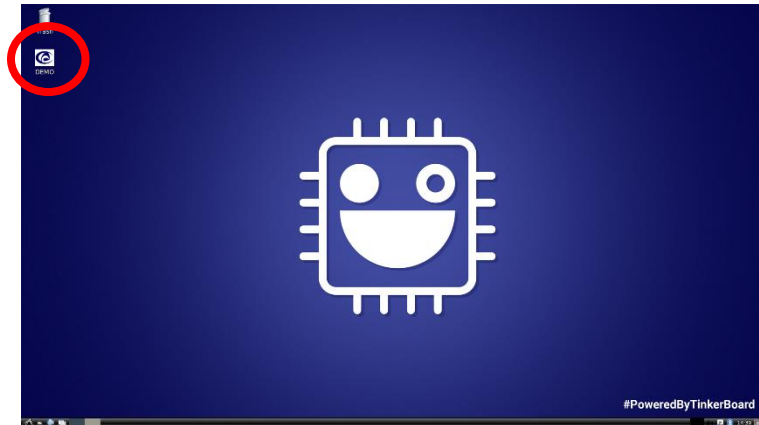
# 5. デモ実行方法

- 5. 1. Focus & OIS デモ
- 5. 2. 12M静止画撮影
- 5. 3. 静止画撮影
- 5. 4. 動画撮影
- 5. 5. デモソフトウェアビルド
- 5. 6. ストリーミング画像反転

# 5. 1. Focus & OIS デモ

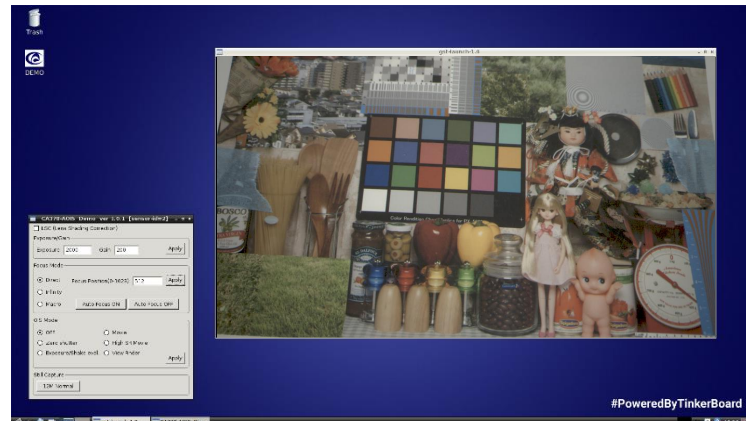
Focus & OIS デモの起動:

1. デスクトップの「DEMO」アイコンをクリックします。



2. 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。

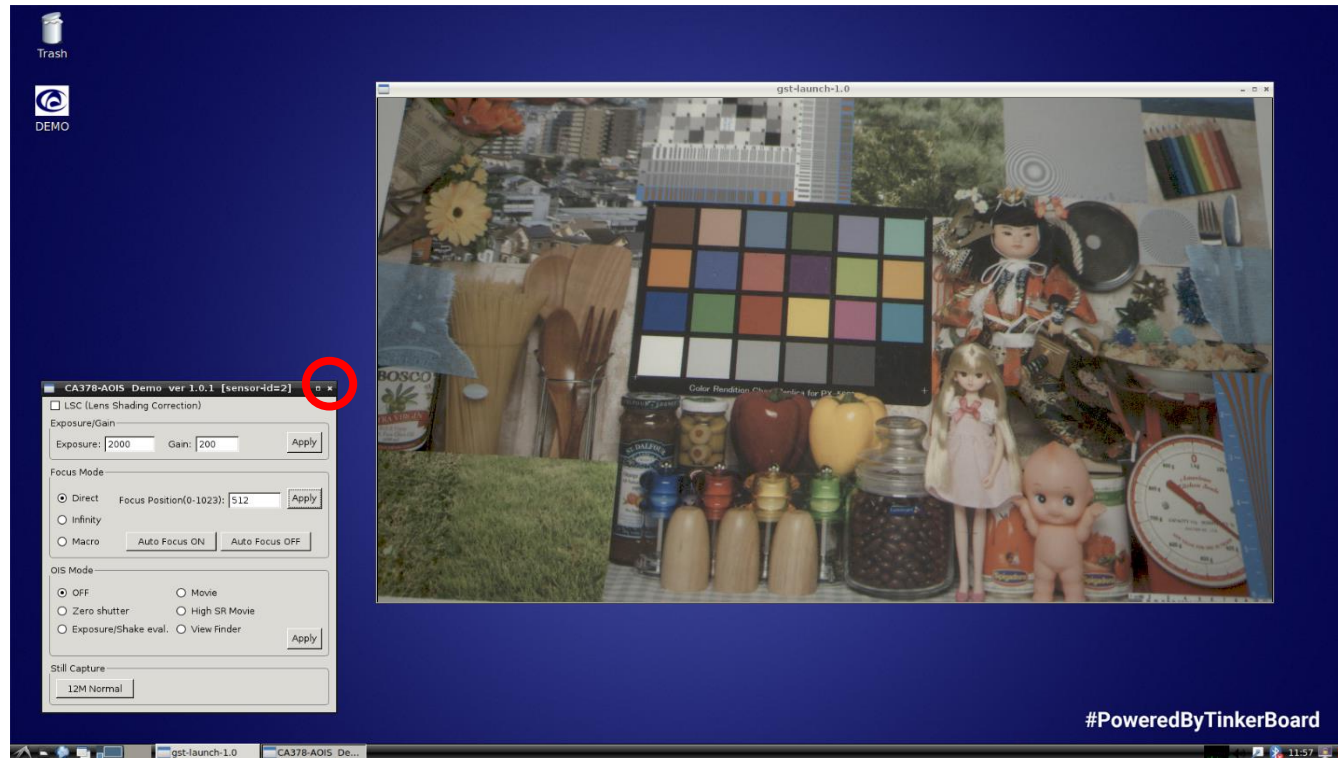
※機能詳細は、P12を参照ください。



# 5. 1. Focus & OIS デモ

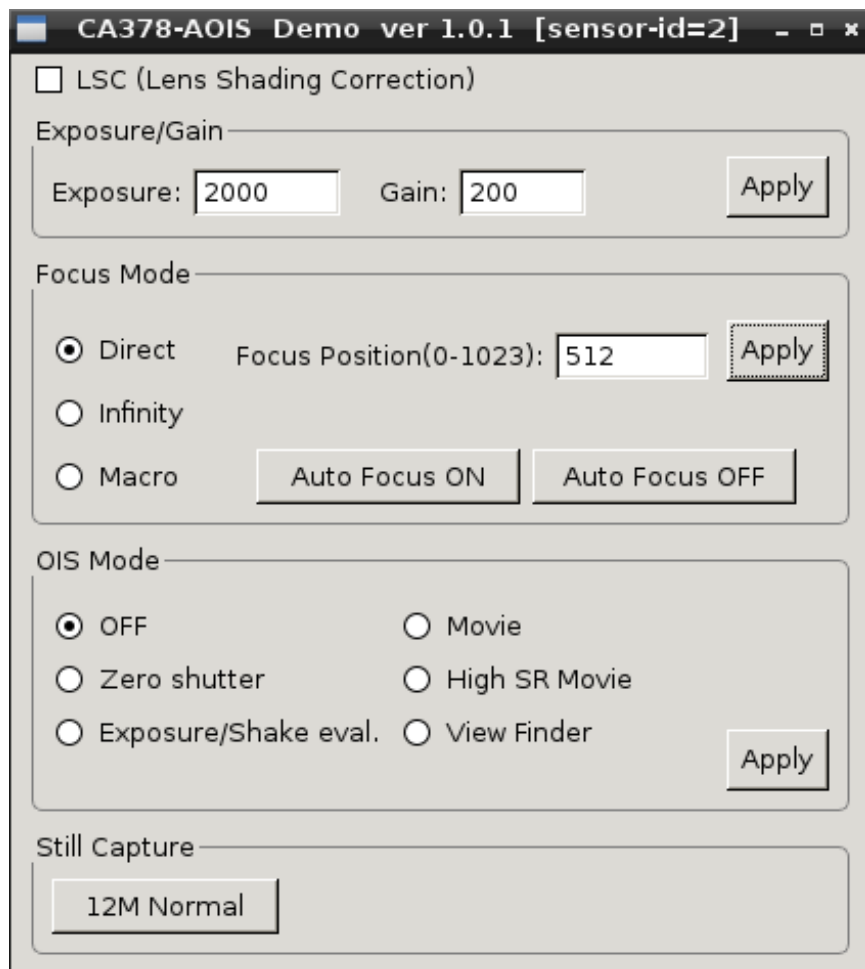
Focus & OIS デモの終了:

1. ×ボタンをクリックします。



# 5. 1. Focus & OIS デモ

Focus & OIS の各機能について説明します。

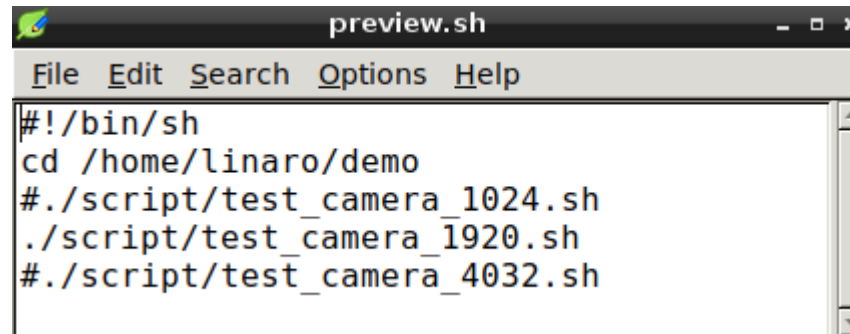


機能	説明
LSC	チェックするとシェーディング補正を有効にします ※理論値を設定してあります
Exposure/Gain	Exposure: 露光時間を設定します(1-65515) Gain: ゲインを設定します(100-2200) Apply: 設定を適用します
Focus Mode	Direct: フォーカス位置を直接指定します Infinity: フォーカス位置を無限遠に設定します Macro: フォーカス位置を近距離に設定します Focus Position : フォーカス位置 Apply: 設定を適用します Auto Focus ON: オートフォーカスを有効にします Auto Focus 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。

# 5. 1. Focus & OIS デモ

ストリーミング画像サイズを変更するには、以下のファイルのパラメータを変更してください。

~/demo/script/preview.sh



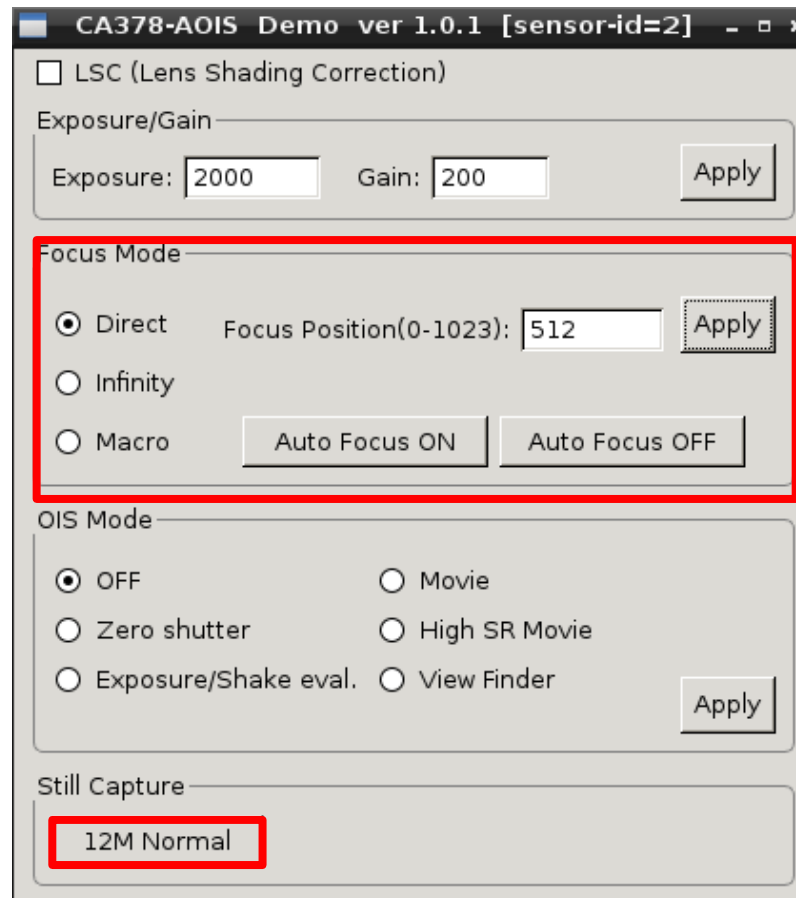
```
#!/bin/sh
cd /home/linaro/demo
#./script/test_camera_1024.sh
./script/test_camera_1920.sh
#./script/test_camera_4032.sh
```

Setting	Parameter
4032x3040	Width=4032, Height=3040, fps=15
1920x1080	Width=1920, Height=1080, fps=80
1024x768	Width=1024, Height=768, fps=210

## 5. 2. 12M静止画撮影

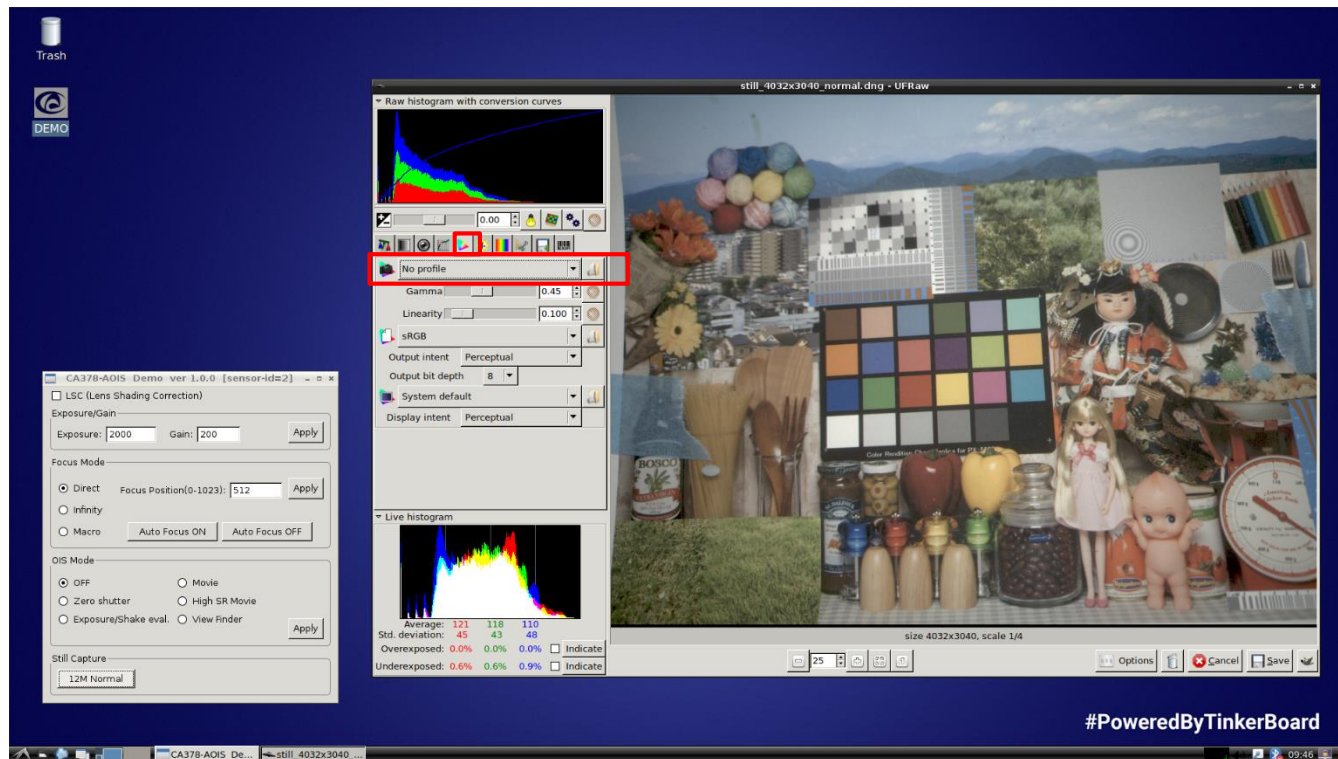
12M静止画撮影の実行:

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。



# 5. 2. 12M静止画撮影

3. RAWとDNGフォーマットで撮像が可能です。
4. カラーマネジメントのカメラプロファイル設定を「No profile」に設定してください。



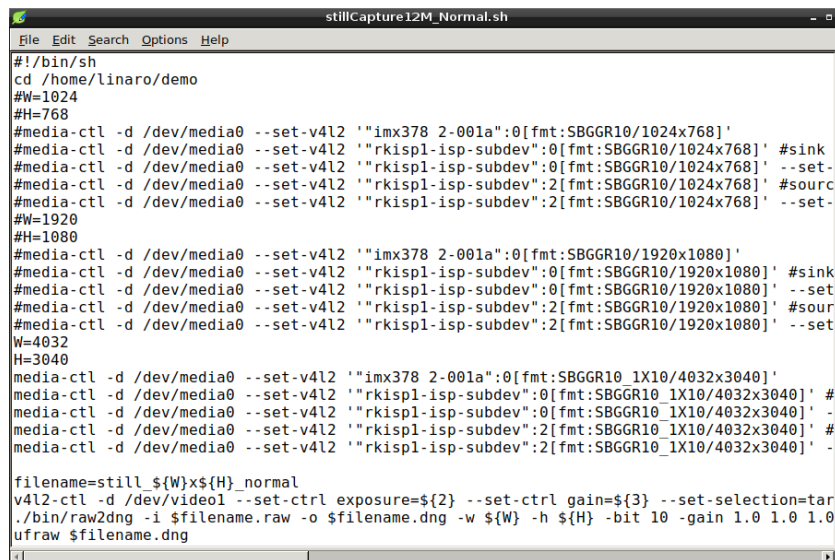


## 5. 3. 静止画撮影

指定画像サイズで撮影するには、Stillスクリプトを変更してアプリの撮影ボタンを押すか、キャプチャコマンドを実行してください。

### Stillスクリプト

~/demo/script/stillCapture12M\_Normal.sh



```
#!/bin/sh
cd /home/linaro/demo
#W=1024
#H=768
#media-ctl -d /dev/media0 --set-v4l2 "imx378 2-001a":0[fmt:SBGGR10/1024x768]'
#media-ctl -d /dev/media0 --set-v4l2 "rkisp1-isp-subdev":0[fmt:SBGGR10/1024x768]' #sink
#media-ctl -d /dev/media0 --set-v4l2 "rkisp1-isp-subdev":0[fmt:SBGGR10/1024x768]' --set-
#media-ctl -d /dev/media0 --set-v4l2 "rkisp1-isp-subdev":2[fmt:SBGGR10/1024x768]' #sourc
#media-ctl -d /dev/media0 --set-v4l2 "rkisp1-isp-subdev":2[fmt:SBGGR10/1024x768]' --set-
#W=1920
#H=1080
#media-ctl -d /dev/media0 --set-v4l2 "imx378 2-001a":0[fmt:SBGGR10/1920x1080]'
#media-ctl -d /dev/media0 --set-v4l2 "rkisp1-isp-subdev":0[fmt:SBGGR10/1920x1080]' #sink
#media-ctl -d /dev/media0 --set-v4l2 "rkisp1-isp-subdev":0[fmt:SBGGR10/1920x1080]' --set-
#media-ctl -d /dev/media0 --set-v4l2 "rkisp1-isp-subdev":2[fmt:SBGGR10/1920x1080]' #sourc
#media-ctl -d /dev/media0 --set-v4l2 "rkisp1-isp-subdev":2[fmt:SBGGR10/1920x1080]' --set-
#W=4032
#H=3040
#media-ctl -d /dev/media0 --set-v4l2 "imx378 2-001a":0[fmt:SBGGR10_1X10/4032x3040]'
#media-ctl -d /dev/media0 --set-v4l2 "rkisp1-isp-subdev":0[fmt:SBGGR10_1X10/4032x3040]' #
#media-ctl -d /dev/media0 --set-v4l2 "rkisp1-isp-subdev":0[fmt:SBGGR10_1X10/4032x3040]' -
#media-ctl -d /dev/media0 --set-v4l2 "rkisp1-isp-subdev":2[fmt:SBGGR10_1X10/4032x3040]' #
#media-ctl -d /dev/media0 --set-v4l2 "rkisp1-isp-subdev":2[fmt:SBGGR10_1X10/4032x3040]' -
filename=still ${W}x${H}_normal
v4l2-ctl -d /dev/video1 --set-ctrl exposure=${2} --set-ctrl gain=${3} --set-selection=target=crop,top=0,left=0,width=4032,height=3040
./bin/raw2dng -i $filename.raw -o $filename.dng -w ${W} -h ${H} -bit 10 -gain 1.0 1.0 1.0
ufraw $filename.dng
```

### キャプチャコマンド

```
# v4l2-ctl -d /dev/video1
--set-ctrl exposure=2000
--set-ctrl gain=200
--set-selection=target=crop,top=0,left=0,width=4032,height=3040
--set-fmt-video=width=4032,height=3040,pixelformat=BG10
--stream-mmap=8 --stream-to=temp.raw --stream-count=1
```

Setting	Parameter
4032x3040	Width=4032, Height=3040
1920x1080	Width=1920, Height=1080
1024x768	Width=1024, Height=768



## 動画メモリ領域の確保

```
# cd ~/demo/script/  
# ./ramdisk.sh
```

## 動画撮影(mp4)

```
# cd ~/demo/script/  
# ./mp4_recording.sh /mnt/ram/test 1920 1080 10 40
```

Argument	Description
arg1	Movie file name
arg2	Width
arg3	Height
arg4	Fps
arg5	Capture frame num

## 動画ビューワー(mp4)

```
# cd ~/demo/script/  
# mplayer -fps 10 /mnt/ram/test.mp4
```

# 5. 5. デモソフトウェアビルド

## デモソフトウェアビルド手順

```
# cd ~/demo/src/  
# ./MakeDomo.sh  
# mv GUI/DemoGUI /home/linaro/demo/bin
```

※qmakeでエラーが発生する場合には、以下のコマンドでインストールしてください。

```
# sudo apt install qt4-default
```

# 5. 6. ストリーミング画像反転

ストリーミング時の画像を反転するには、以下の手順を行います。

## 1. デモスクリプトファイル編集

- 以下の赤字部分のように編集します。(~/demo/script/test\_camera\_1920.sh)

```
#!/bin/sh
export DISPLAY=:0.0
media-ctl --set-v4l2 '"imx378 2-001a":0[fmt:SRGGB10_1X10/1920x1080]'
media-ctl -d /dev/media0 --set-v4l2 '"rkisp1-isp-subdev":0[fmt:SRGGB10_1X10/1920x1080]' #sink
media-ctl -d /dev/media0 --set-v4l2 '"rkisp1-isp-subdev":0[fmt:SRGGB10_1X10/1920x1080]' --set-v4l2 '"rkisp1-isp-subdev":0[crop:(0,0)/1920x1080]'
media-ctl -d /dev/media0 --set-v4l2 '"rkisp1-isp-subdev":2[fmt:YUYV8_2X8/1920x1080]' #source
media-ctl -d /dev/media0 --set-v4l2 '"rkisp1-isp-subdev":2[fmt:YUYV8_2X8/1920x1080]' --set-v4l2 '"rkisp1-isp-subdev":2[crop:(0,0)/1920x1080]'
v4l2-ctl --set-ctrl=vertical_flip=0 --set-ctrl=horizontal_flip=0
gst-launch-1.0 rkcamsrc device=/dev/video1 io-mode=4 isp-mode=2A tuning-xml-path=./script/IMX378_1022.xml ! videoconvert ! video/x-raw,format=NV12,width=1280,height=720 ! rkximagesink
```

## 2. ドライバファイル編集

- 以下の赤字部分のように編集します。
  - (~/CA378\_v1.0.1\_TinkerOS\_Debian9\_v2.0.8\_src\_build/debian\_kernel\_cp/drivers/media/i2c/imx378.c)

```
1450line
    fmt->format.code = MEDIA_BUS_FMT_SRGGB10_1X10;

1492line
    fmt->format.code = MEDIA_BUS_FMT_SRGGB10_1X10;
```

## 3. カーネルのビルドとインストール

- 以下のコマンドでカーネルをビルド、インストールします。

```
cd ~/CA378_v1.0.1_TinkerOS_Debian9_v2.0.8_src_build
./BuildKernelSources.sh
```

# 5. 6. ストリーミング画像反転

## 4. Raw2Dngファイル編集

- 以下の赤字部分のように編集します。
- (~demo/src/tool/raw2dng.c)

```
201line  
TIFFSetField (tif, TIFFTAG_CFAPATTERN, 4, "¥00¥01¥01¥02");
```

## 5. デモツールのビルドと更新

- 以下のコマンドでデモツールをビルド、更新します。

```
cd ~/demo/src  
./MakeDemo.sh
```

# Appendix

デモソフトのファイル構成について説明します。

```
demo
├── appicon.png
├── Install.sh
├── bin
│   ├── demo.ini
│   ├── DemoGUI
│   ├── raw2dng
│   └── raw2hdr
├── script
│   ├── demo.sh
│   ├── IMX378_I030_XGA_LSC.xml
│   ├── mp4_recording.sh
│   ├── preview.sh
│   ├── ramdisk.sh
│   ├── stillCapture12M_Normal.sh
│   ├── test_camera_I024.sh
│   ├── test_camera_I920.sh
│   └── test_camera_4032.sh
└── src
    ├── GUI
    │   ├── af_control.c
    │   ├── af_control.h
    │   ├── communication.h
    │   ├── communication_tinker.c
    │   ├── debug_util.h
    │   ├── demo_control.c
    │   ├── demo_control.h
    │   ├── DemoGUI.pro
    │   ├── lsc_control.c
    │   ├── lsc_control.h
    │   ├── main.cpp
    │   ├── mainwindow.cpp
    │   ├── mainwindow.h
    │   ├── mainwindow.ui
    │   ├── Makefile
    │   ├── ois_control.c
    │   ├── ois_control.h
    │   ├── slave_address.h
    │   ├── tools.h
    │   └── types_util.h
    ├── tool
    │   ├── libtiff.patch
    │   ├── Makefile
    │   ├── raw2dng.c
    │   ├── raw2hdr.c
    │   └── tools.h
    └── MakeDemo.sh
```

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。
tool	画像ファイル変換ツールが記述されています。

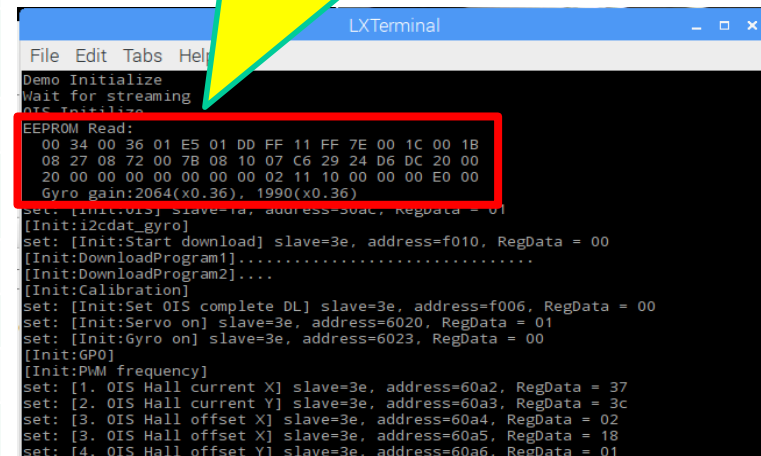
# A. 2. 設定ファイルについて

設定ファイルのdemo.iniについて説明します。

```
# DEMO Setting
preview = /home/linaro/demo/script/preview.sh
stillCapture12M_Normal = /home/linaro/demo/script/stillCapture12M_Normal.sh
gyroGainRateX=1.00
gyroGainRateY=1.00
autoFocusGain=2.0
autoFocusConfidenceThreshold=10
autoFocusMoveLimit=100
AutoFocusAverageNum=1
Exposure=1000
Gain=200
```

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。
Exposure	露光時間を調整します。
Gain	ゲインを調整します。

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=30ac, 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
```