

<VisionProcessingFramework(VPF)環境構築手順>

前提条件

- OSがRaspbianであること
 - SSPライブラリがインストール済であること
 - (SSPライブラリの使い方を参照してください)
- ダウンロードページの表のように、OSとSSPライブラリとVPFは動作する組み合わせで入手してください。

■wxWidgetsのインストール (VPFバージョン1.0.4以前を使う場合)

```
pi@raspberrypi: ~ $ sudo apt-get install libwxgtk2.8-dev
```

VPFはwxWidgetsを使用します。

■wxWidgetsのインストール (VPFバージョン1.0.5を使う場合)

Raspbian Stretch では上記の方法では動作しなくなりましたので、wxWidgetsをソースから作成します。

```
pi@raspberrypi:~$ wget https://github.com/wxWidgets/wxWidgets/releases/download/v2.8.12/wxWidgets-2.8.12.tar.gz
pi@raspberrypi:~$ tar -xvzf wxWidgets-2.8.12.tar.gz
pi@raspberrypi:~$ cd wxWidgets-2.8.12
pi@raspberrypi:~/wxWidgets-2.8.12 $ ./configure --with-gtk
pi@raspberrypi:~/wxWidgets-2.8.12 $ make -j3
pi@raspberrypi:~/wxWidgets-2.8.12 $ sudo make install
pi@raspberrypi:~/wxWidgets-2.8.12 $ sudo cp wx-config /usr/local/bin
pi@raspberrypi:~/wxWidgets-2.8.12 $ sudo chmod 755 /usr/local/bin/wx-config
```

makeの時間はRaspberry Pi 3 で約10分です。

■VPFのインストール

(1) VPFのソースをダウンロードページからダウンロードしてください。(Chromium では~/Downloads に保存されます) unzipコマンドで展開します。

```
pi@raspberrypi:~$ cd
pi@raspberrypi:~$ unzip ~/Downloads/VisionProcessingFramework/[-バージョン番号].zip
```

(2) Gtk-WARNINGの対応を行います。

```
pi@raspberrypi:~/VisionProcessingFramework/src/Plugins $ sudo mkdir -p /root/.local/share
```

<VisionProcessingFramework(VPF)操作手順>

■とりあえず試してみる

(1) VPFを起動します

```
pi@raspberrypi:~$ cd
pi@raspberrypi:~$ cd VisionProcessingFramework/src
pi@raspberrypi:~/VisionProcessingFramework/src $ sudo ./VisionProcessingFramework
```

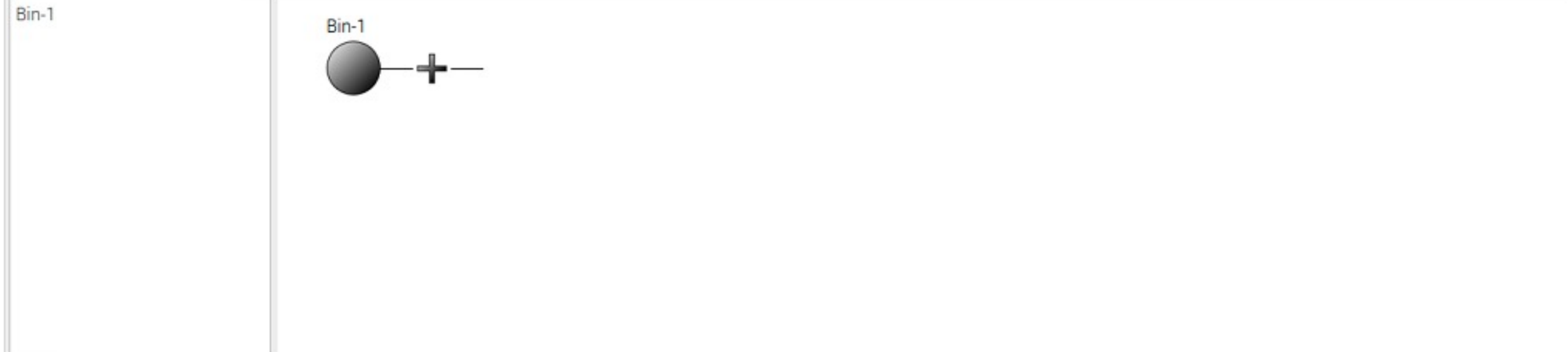
(2) Plugin managerを開きます

メニュー-> Plugin manager > Plugin manager を選択します。

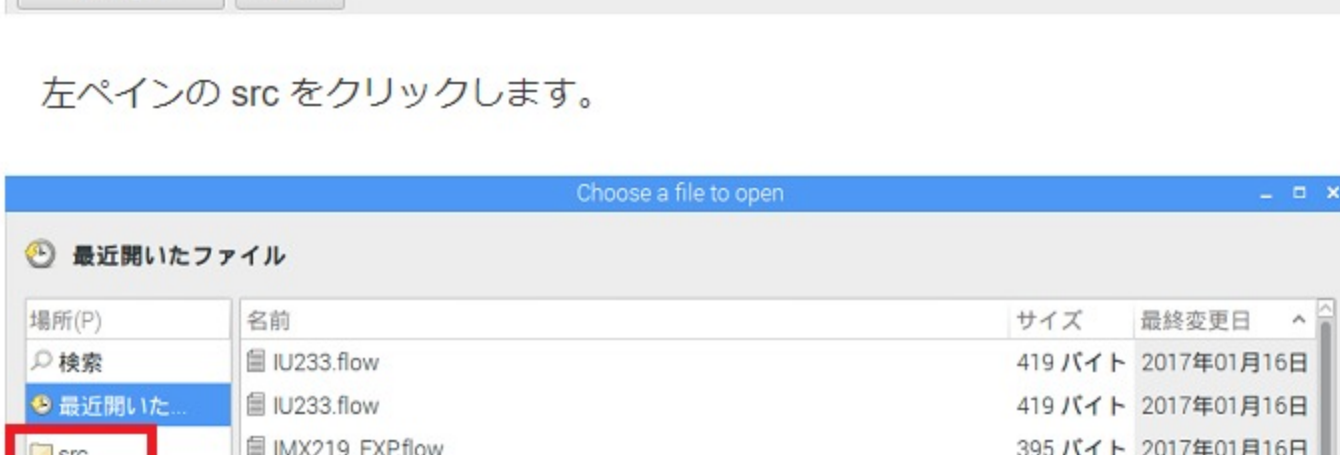


(3) フローをロードします

右下の Load Flow ボタンをクリックします。



左ページの src をクリックします。

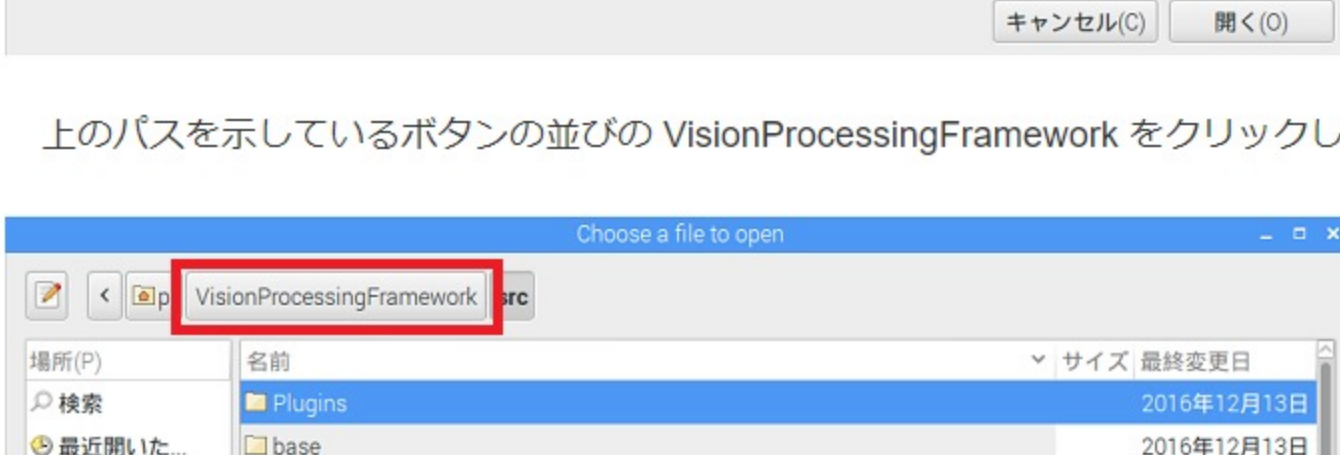


上のパスを示しているボタンの並びの VisionProcessingFramework をクリックします。



IMX219_EXP.flow を選択します。

Open ボタンをクリックします。



プラグインをつないだフローが読み込まれます。



(4) センサーを設定します

メニュー-> Sensor config > Sensor config を選択します。



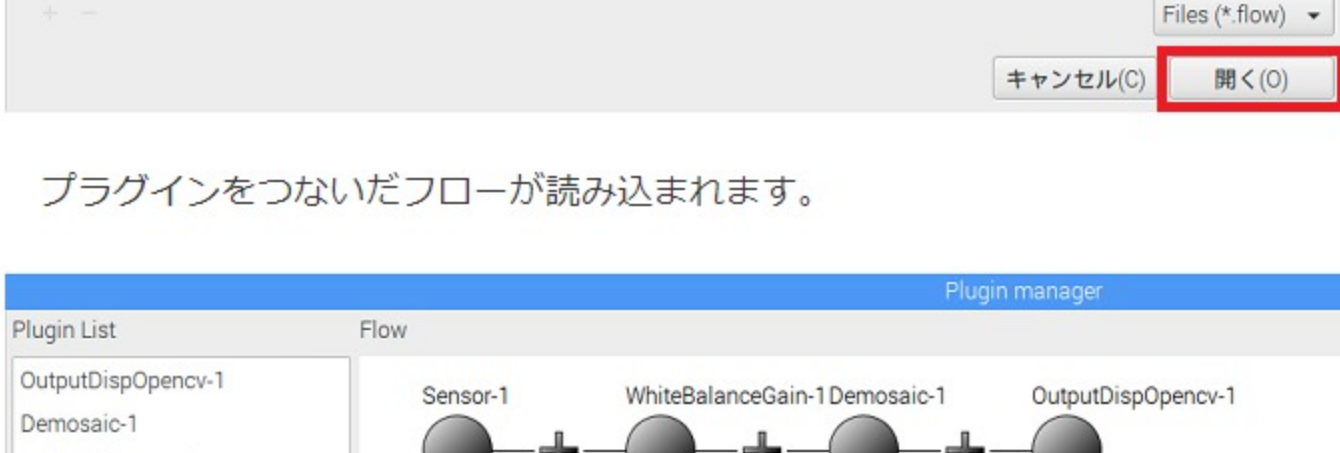
IMX219(640x480@60Hz).xml が選択されているはずですが、

Bit count が 10bit になっていることを確認します。

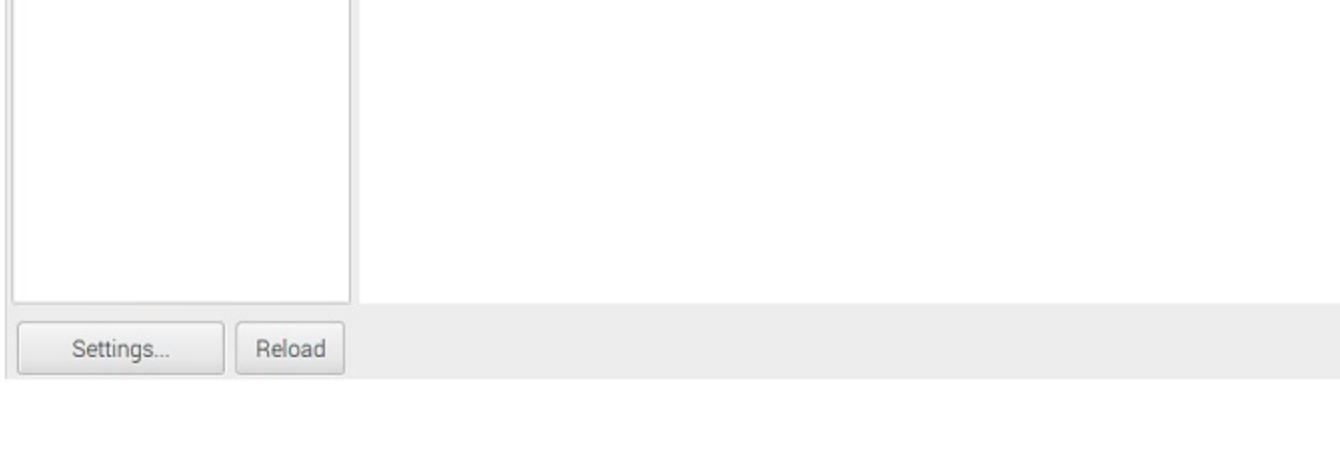
Apply をクリックします。



Sensor setting ボタンをクリックします。

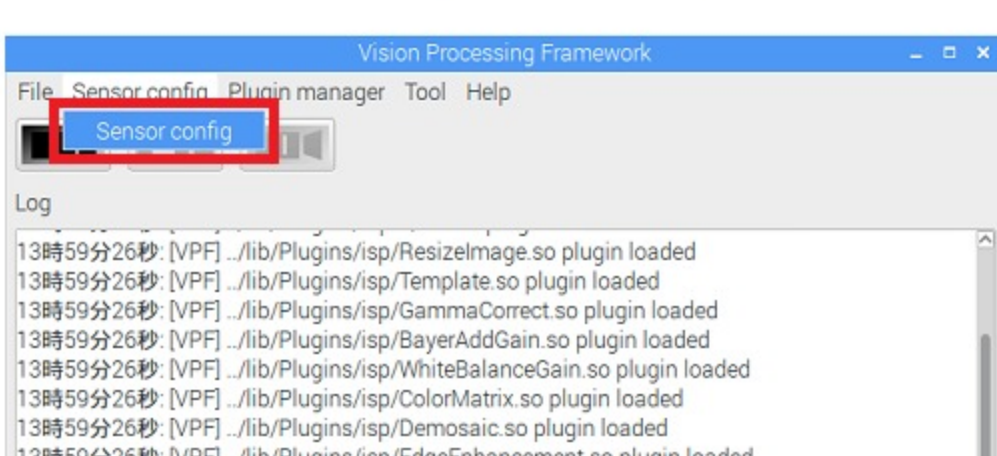


Sensor setting ウィンドウが開きます。



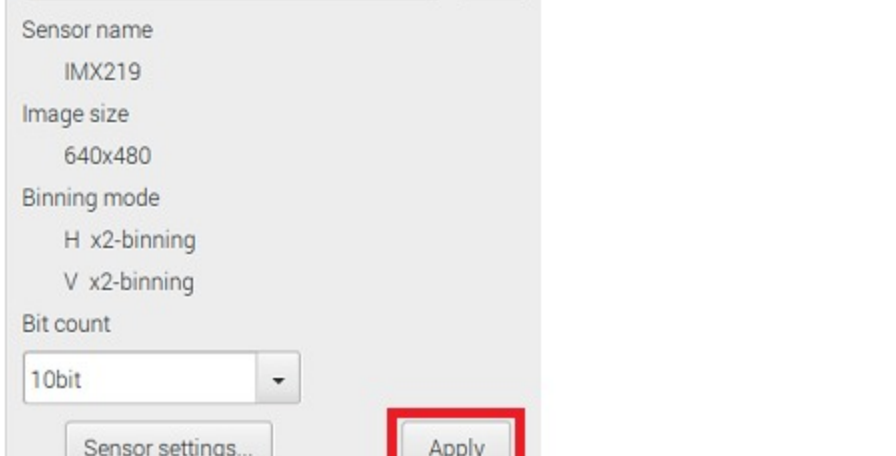
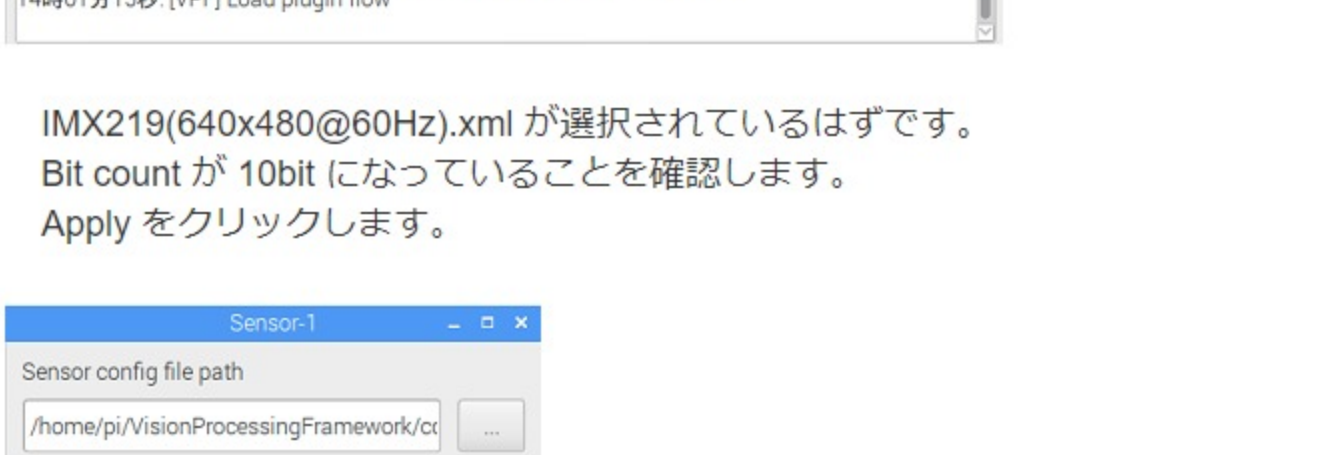
(5) カメラ表示します

VPF に戻って再生ボタンをクリックします。



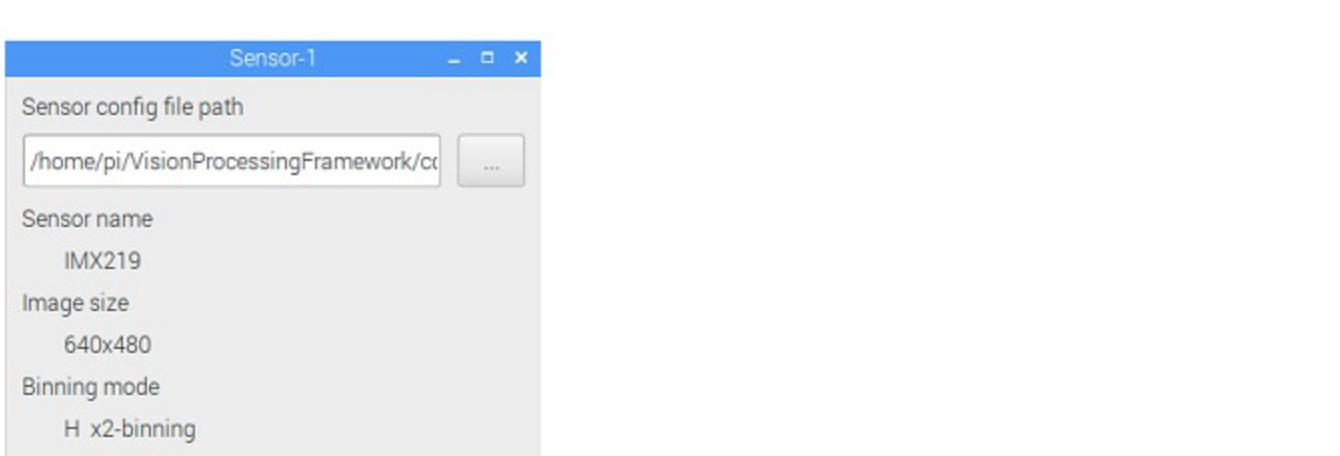
(6) Sensor setting し直す

(4) で開いた Sensor setting にある Analog gain や Exposure time をスライドしてOutputDispOpenCV-1の調整をします。



(7) Plugin manager を触ってみます

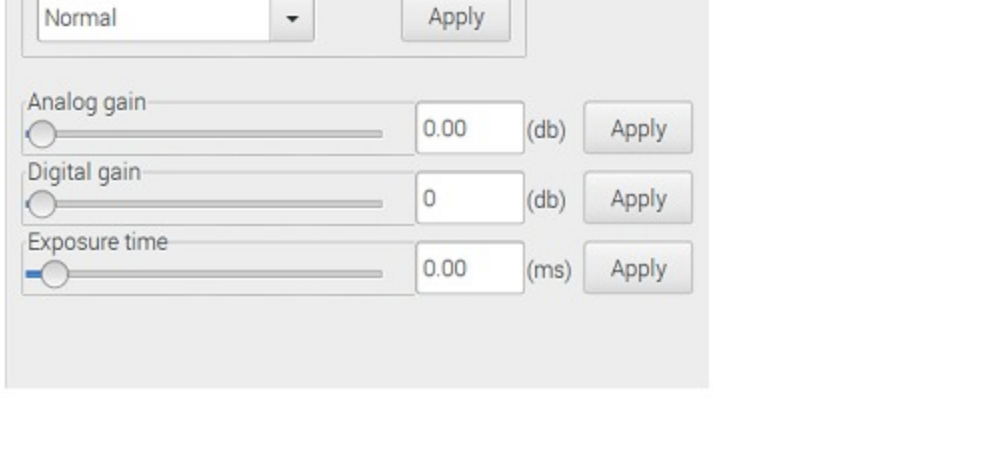
プラグイン(丸いアイコン)を右クリックすると設定ウィンドウがポップアップしますので設定値を変えて Apply してみてください。



各プラグインの機能はダウンロードページにある「プラグインの説明」を参照してください。

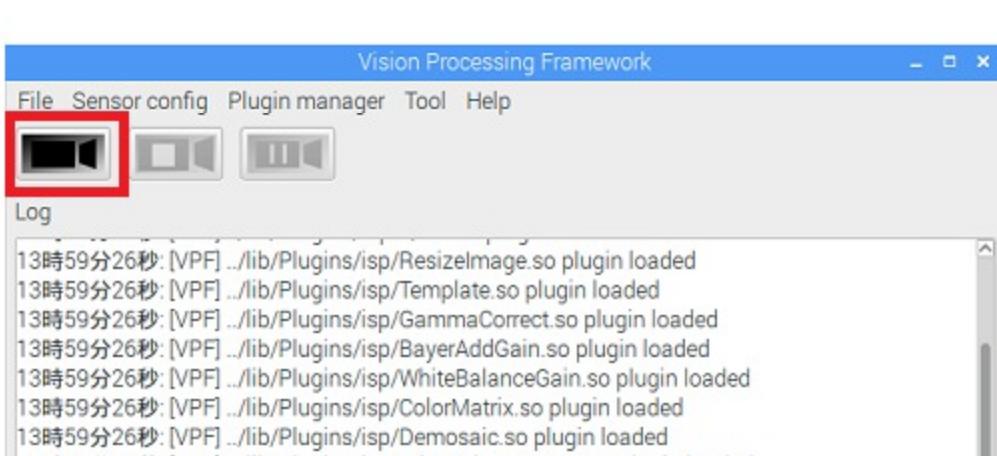
(8) 停止

VPFの真ん中のボタンをクリックして カメラ表示を停止します。



(9) 終了

VPFのウィンドウを閉じるボタンで終了します。



<VisionProcessingFramework(VPF)構築手順>

■VPFのビルド

VPFをソースから作成するときの手順になります。

(1) 共有ライブラリをコピー

動作組み合わせ表以外のを試す場合はSSPライブラリの共有ライブラリをコピーして下さい。

```
pi@raspberrypi:~$ cp libssp/[-バージョン番号]/lib/* VisionProcessingFramework/config/sensor/lib
```

(2) Pluginをビルドします。

```
pi@raspberrypi:~$ cd
pi@raspberrypi:~$ cd VisionProcessingFramework/src/Plugins
pi@raspberrypi:~/VisionProcessingFramework/src/Plugins $ make clean
pi@raspberrypi:~/VisionProcessingFramework/src/Plugins $ make
pi@raspberrypi:~/VisionProcessingFramework/src/Plugins $ chmod +x GetSo.sh
```

makeの時間はRaspberry Pi 3 で約20分です。

プラグインをプラグイン用ディレクトリに配置します。

```
pi@raspberrypi:~/VisionProcessingFramework/src/Plugins $ ./GetSo.sh
```

(3) VPFをビルドします。

```
pi@raspberrypi:~$ cd VisionProcessingFramework/src
pi@raspberrypi:~/VisionProcessingFramework/src $ make clean
pi@raspberrypi:~/VisionProcessingFramework/src $ make
```

makeの時間はRaspberry Pi 3 で約5分です。