

LEAG – SDK

Linux 版ユーザーズガイド

Ver.1.03 (2021/12/17)



リーグソリューションズ株式会社

〒277-0882 千葉県柏市柏の葉 5-4-6 東葛テクノプラザ 611

E-mail: info@leag.jp

目次

1. はじめに.....	3
2. 構成.....	4
3. 動作環境.....	7
4. インストール.....	8
4.1. サードパーティソフトウェアのインストール.....	8
4.2. LEAG-Library のインストール.....	10
5. サンプルコード.....	11
5.1. C++.....	11
5.2. Python.....	12
5.3. ROS ノード.....	12
5.3.1. 環境準備.....	14
5.3.2. カメラキャリブレーション.....	14
5.3.3. 実行方法.....	15
6. その他.....	16
6.1. 改訂履歴.....	16
6.2. 免責事項.....	16
6.3. ライセンス.....	16
6.4. 問い合わせ.....	16

1. はじめに

LEAG-SDK は、LEAG を搭載した AR コード付きマーカを撮像した画像を用い、マーカの位置及び姿勢情報を取得するためのソフトウェア開発キットです。

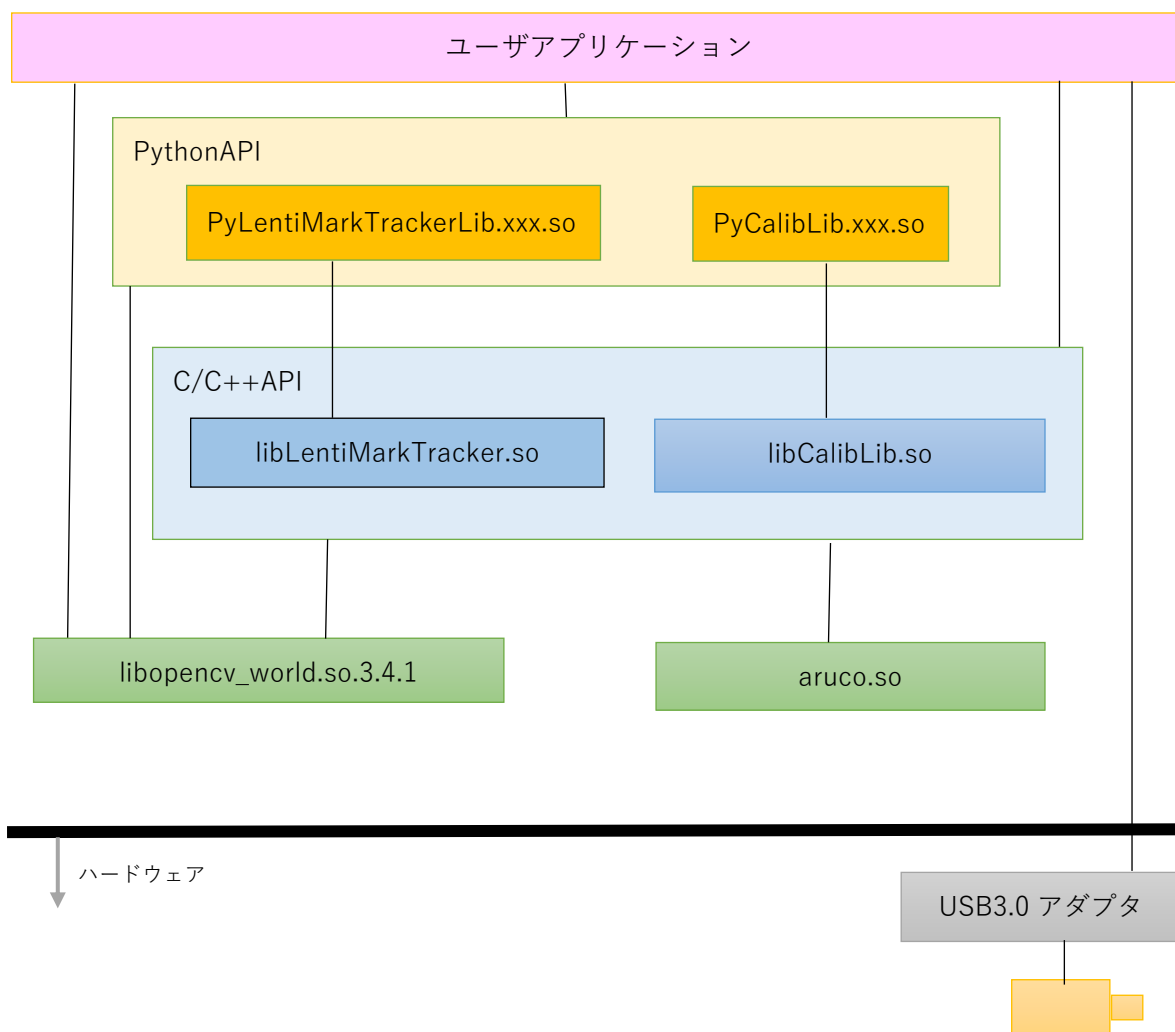
本ドキュメントは、LEAG-Library Linux 版の使用方法について記載します。

本ドキュメントは、Linux を使用したシステムを構築するソフトウェア技術者を対象に想定しています。

※LEAG-Library 使用方法については LEAG-Library ユーザーズガイドを参照ください。

2. 構成

LEAG-Library Linux 版のソフトウェア構成は以下の通りです。



Linux OS 用	内容
libLentiMarkTrackerLib.so	LEAG-Library 開発用関数ライブラリ本体
libCalibLib.so	カメラキャリブレーション用関数ライブラリ
PyLentiMarkTrackerLib.xxx.so	LEAG-Library Python ラッパー
PyCalibLib.xxx.so	カメラキャリブレーション用関数 Python ラッパー
aruco.so	AR 検出用ライブラリ
libopencv_world.so.3.4.1	画像処理ライブラリ

・フォルダ構成

LEAG_SDK

- Licenses	: ライセンス関連格納フォルダ
- install	: LEAG-Library インストーラ格納フォルダ
- log	: インストールログ格納フォルダ
- lib	: ライブラリファイル格納フォルダ
- develop	: 開発用ライブラリ格納フォルダ
- arm64	: ARM 64 ビット用
- arm86	: ARM 32 ビット用
- x64	: x64 用
- runtime	: ランタイム用ライブラリ格納フォルダ
- arm64	: ARM 64 ビット用
- arm86	: ARM 32 ビット用
- x64	: x64 用
- include	: ヘッダーファイル格納フォルダ
- sample	: ユーザサンプルコード
- cpp	: C++サンプルコード
- build	: ビルド用フォルダ
- conf	: 設定ファイル格納フォルダ
- img	: サンプル画像
- log	: ログ出力用フォルダ
- python	: Python サンプルコード
- JetsonNano	: JetsonNano による検出サンプルコード (ArduCAM を使用)
- Data	: 計測結果出力用フォルダ
- conf	: 設定ファイル格納フォルダ
- img	: サンプル画像
- RasPi	: RaspberryPi による検出サンプルコード (Pi カメラ HQ を使用)
- Data	: 計測結果出力用フォルダ
- conf	: 設定ファイル格納フォルダ
- img	: サンプル画像
- WebCam	: USB カメラによる検出サンプルコード
- Data	: 計測結果出力用フォルダ
- conf	: 設定ファイル格納フォルダ
- img	: サンプル画像
- ros	: ROS ノードサンプルコード
- lenti_mark	: マーカ検出ノードサンプルコード

- | - Data : 設定ファイル格納フォルダ
 - | - CameraParams : カメラ設定ファイル格納フォルダ
 - | - MarkerParams : マーカ設定ファイル格納フォルダ
- | - include : ヘッダーファイル格納フォルダ
- | - launch : ローンチファイル格納フォルダ
- | - msg : メッセージ定義ファイル格納フォルダ
- | - src : C++サンプルコード
- | - software : サードパーティソフトウェア格納フォルダ
 - | - ArduCAM : ArduCAM インストーラ格納フォルダ
 - | - JetsonNano : JetsonNano 用
- | - CodeMeter : コードメータランタイムインストーラ格納フォルダ
 - | - arm64 : ARM 64 ビット用
 - | - arm86 : ARM 32 ビット用
 - | - x64 : x64 用
- | - OpenCV : OpenCV インストーラ格納フォルダ
 - | - Common : 共通インストーラ
 - | - JetsonNano : JetsonNano 向けインストーラ
 - | - RasPi : RaspberryPi 向けインストーラ
- | - log : インストールログ出力用フォルダ

3. 動作環境

LEAG-Library Linux 版の動作確認は下記環境にて行っております。

動作確認環境：

別紙「LEAG-Library 動作確認環境」を参照ください。

必要ソフトウェア：

build-essential (g++, gcc)

CMake ver.3.10.2

OpenCV ver.3.4.1

サンカーラ(株) コードメータランタイムキット

ROS (※)

(※) ROS ノードのサンプルを使用される場合

4. インストール

インストール方法は、LEAG_SDK_x_x_x.tar.gz を任意のディレクトリにコピーし、アーカイブを展開して下さい。

※以下コピー先ディレクトリを“**userDir**”とします。

※LEAG_SDK_x_x_x.tar.gz の“x_x_x”はバージョンになります。

例) アーカイブファイルの展開

以下のコマンドを実行し、ファイルを展開して下さい。

```
$ sudo tar xfvz LEAG_SDK_x_x_x.tar.gz
```

4.1. サードパーティソフトウェアのインストール

LEAG-Libraryに必要なサードパーティソフトウェアを下記手順でインストールして下さい。

1) セキュリティドングルソフトウェアのインストール

下記コマンドを実行し、CodeMeter をインストールして下さい。

LEAG-SDK/software/CodeMeter/

install_CodeMeter.sh -c (CPU 種別：x64/arm86/arm64)

例)

```
$ cd userDir/LEAG_SDK/software/CodeMeter
```

• X64 の場合

```
$ sudo ./install_CodeMeter.sh -c x64
```

• ARM 32 ビット の場合

```
$ sudo ./install_CodeMeter.sh -c arm86
```

• ARM 64 ビット の場合

```
$ sudo ./install_CodeMeter.sh -c arm64
```


2) OpenCV のビルド・インストール

下記コマンドを実行し、OpenCV をインストールして下さい。

LEAG-SDK/software/OpenCV/

install_OpenCV.sh -h (ハード種別：Common/ JetsonNano/RasPi)

※OpenCV のダウンロードとビルド、インストールを行います。

※ハード種別について

- Common は共通のビルド設定
- JetsonNano は JetsonNano (ArduCAM) 向けのビルド設定
- RasPi は RaspberryPi (picamera ライブラリ) 向けビルド設定

例)

```
$ cd userDir/LEAG_SDK/software/OpenCV
```

- Common の場合

```
$ sudo ./install_OpenCV.sh -h Common
```

- JetsonNano の場合

```
$ sudo ./install_OpenCV.sh -h JetsonNano
```

- RaspberryPi の場合

```
$ sudo ./install_OpenCV.sh -h RasPi
```

- その他

コマンドの実行ログは下記に出力されます。

LEAG-SDK/software/log/コマンド名.txt

4.2. LEAG-LIBRARY のインストール

下記コマンドを実行し、LEAG-Library をインストールして下さい。

LEAG-SDK/install/

install.sh -c (CPU 種別 : x64/arm86/arm64) -t (ライセンス種別 : develop/runtime)

※ライセンス種別 : 開発ライセンスは develop、ランタイムライセンスは runtime

例)

```
$ cd userDir/LEAG_SDK/install
```

[X64]

- ・開発ライセンスの場合

```
$ sudo ./install.sh -c x64 -t develop
```

- ・ランタイムライセンスの場合

```
$ sudo ./install.sh -c x64 -t runtime
```

[ARM 32 ビット]

- ・開発ライセンスの場合

```
$ sudo ./install.sh -c arm86 -t develop
```

- ・ランタイムライセンスの場合

```
$ sudo ./install.sh -c arm86 -t runtime
```

[ARM 64 ビット]

- ・開発ライセンスの場合

```
$ sudo ./install.sh -c arm64 -t develop
```

- ・ランタイムライセンスの場合

```
$ sudo ./install.sh -c arm64 -t runtime
```

5. サンプルコード

C++、Python、ROS ノードのサンプルコードがあります。

LEAG-Library の詳細については「LEAG-Library ユーザーズガイド」を参照ください。

※LEAG-Library を実行するにはライセンスキー（USB ドングル）が必要です。

5.1.C++

画像ファイルからマーカ検出と計測結果をターミナルへ出力するサンプルコードです。

結果データはカメラを基準としたマーカの位置姿勢情報になります。

1) ビルド

下記コマンドを実行し、ビルドします。

```
$ cd userDir/LEAG_SDK/sample/cpp/build
```

```
$ sudo cmake ../
```

```
$ sudo make
```

2) 実行

下記コマンドを実行し、ターミナルに検出結果が出力されることを確認してください。

```
$ ./sample.out
```

出力結果例

```
LEAG-SDK Open. Detect Start!
-----
marker_id: 3
score: 100
contrast: 121
pos_x = -8.47843, pos_y = -45.34889, pos_z = 446.89485
angle_x = -163.59864, angle_y = 1.04374, angle_z = 1.87006
rot_0 = 0.99930, rot_1 = -0.03263, rot_2 = 0.01822
rot_3 = -0.03645, rot_4 = -0.95863, rot_5 = 0.28232
rot_6 = 0.00825, rot_7 = -0.28278, rot_8 = -0.95915
-----
```

5.2. PYTHON

Python サンプルの詳細は「LEAG-SDK サンプルコードガイド」を参照して下さい。

5.3. ROS ノード

lenti_mark ノードはカメラノードから画像を取得し、画像からマーカの検出を行い、計測結果をトピックと TF に出力するサンプルコードです。

・トピック

lenti_mark/msg/ markers.msg

データ型	データ名	内容
Header	header	ヘッダ情報（※1）
uint32	detect	マーカ検出数
lenti_mark/marker[]	markers	マーカ位置・姿勢情報群

lenti_mark/msg/ marker.msg

データ型	データ名	内容
Header	Header	ヘッダ情報
String	child_frame_id	frame_id（※2）
uint32	id	マーカ ID
uint32	score	信頼性スコア
uint32	contrast	コントラスト
geometry_msgs/Pose	pose	位置・姿勢情報

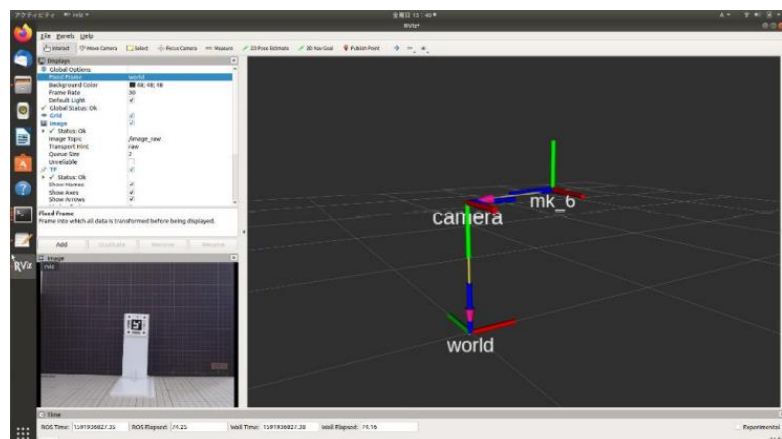
（※1） frame_id は “camera” になります。

（※2） child_frame_id は “mk_マーカ ID” になります。

・TF

RViz でカメラと高精度マーカの相対位置を確認できます。

表示例)



• ローンチファイル

lenti_mark ノードは launch ファイルから起動を行います。

lenti_mark/launch/lenti_mark.launch

パラメータ名	内容
mk_param_file	高精度マーカ設定ファイルパス
campara_type	カメラキャリブレーション情報取得方法 0：ファイル読み込み 1：トピック (CameraInfo)
cam_param_file	カメラキャリブレーションファイルパス
world_flg	原点 frame_id "world" の設定有無 (※1) true：設定有 false：設定無
cam_pos_x	原点(frame_id "world")からみたカメラ位置 (※2)
cam_pos_y	原点(frame_id "world")からみたカメラ位置 (※2)
cam_pos_z	原点(frame_id "world")からみたカメラ位置 (※2)

(※1) frame_id "world" を tf に設定します。

lenti_mark/msg/ markers.msg の Header に設定される frame_id は "camera" です。

(※2) world_flg が "true" の場合に有効。

5.3.1.環境準備

lenti_mark ノードの使用方法は下記になります。

※ROS は **melodic** がインストールされていることを想定しています。

※ROS のワークスペースを" **catkin_ws**"とします。

1) ROS ノードサンプルのコピー

下記コマンドを実行し、lenti_mark ディレクトリをコピーして下さい。

```
$ cd userDir/LEAG_SDK/sample/ros  
$ sudo cp -r lenti_mark catkin_ws/src/
```

2) ビルド

下記コマンドを実行し、lenti_mark ノードをビルドして下さい。

```
$ cd catkin_ws  
$ catkin_make  
$ source devel/setup.bash
```

3) UVC カメラノードのインストール (※1)

下記コマンドを実行し、UVC カメラノードをインストールしてください。

```
$ sudo apt install ros-melodic-uvic-camera  
$ sudo apt install ros-melodic-image-*
```

※1 UVC カメラノードを使用する場合の例です。

5.3.2. カメラキャリブレーション

カメラキャリブレーションは Python サンプルプログラムで行います。

※LEAG-SDK 開発キットのキャリブレーションボードを使用して下さい。

※USB カメラを使用した場合の例です。

※詳細は「LEAG-SDK サンプルコードガイド」を参照して下さい。

1) キャリブレーションプログラムを起動

下記コマンドを実行してプログラムを起動して下さい。

```
$ cd userDir/LEAG_SDK/sample/python/WebCam  
$ python3 10_SampleCalib.py
```

2) キャリブレーションボードの撮影

's'Key を押すと数秒おきに画像の取得が開始します。

「LEAG-SDK サンプルコードガイド」の「キャリブレーションのコツ」を参照の上、キャリブレーションボードを 30 回検出させて下さい。

conf ディレクトリに NewCameraCalibData.txt というファイルが出力されます。

3) ファイル名変更とコピー

conf フォルダに出力された NewCameraCalibData.txt をリネームして下さい。

リネームしたファイルを lenti_mark/Data/CameraParams/にコピーして下さい。

4) ローンチファイルの変更

lenti_mark/launch/lenti_mark.launch

cam_param_file の設定を作成したファイルパスに変更して下さい。

5.3.3. 実行方法

RViz を使用して lenti_mark ノードの動作確認を行います。

1) roscore 起動

ターミナルを起動し、下記コマンドを実行して下さい。

```
$ roscore
```

2) ローンチファイル実行

ターミナルを起動し、下記コマンドを実行して下さい。

```
$ roslaunch lenti_mark ros_sample.launch
```

※UVC カメラノード起動を含めたローンチファイルです。

3) RViz 起動

ターミナルを起動し、下記コマンドを実行して下さい。

```
$ rviz
```

4) マーカの撮影

マーカを撮影し、RViz にマーカの位置・姿勢が表示されることを確認して下さい。

6. その他

6.1. 改訂履歴

2020/3/16 Ver.1.0 初版

2020/6/12 Ver.1.01 LMTLibraryVer1.3.1 変更に伴い追記

2021/8/11 Ver.1.02 サンプルコード(C++,ROS ノード)変更に伴い修正

2021/12/17 Ver.1.03 LMTLibraryVer1.3.3 変更に伴い修正・追記・削除

6.2. 免責事項

このソフトウェアの免責事項は、別途付属の“/Documents/Licenses/License_Agreement.pdf”に記載されています。必ずご一読の上、ご利用されますようお願いいたします。

6.3. ライセンス

Microsoft、Windows、Windows XP、Windows Vista、Windows 7、Windows 8.1、Windows 10 及び、Visual C++は、Microsoft 社の商標もしくは登録商標です。

Intel、インテル、Intel ロゴ、Intel Inside、Intel Inside ロゴ、Intel Atom、Intel Atom Inside、Intel Core、Core Inside、Intel vPro、vPro Inside、Celeron、Celeron Inside、Itanium、Itanium Inside、Pentium、Pentium Inside、Xeon、Xeon Phi、Xeon Inside、Ultrabook、Iris は、アメリカ合衆国および/またはその他の国における Intel Corporation の商標です。

Linuxは、Linus Torvalds氏の日本およびその他の国における登録商標または商標です。

LEAG-SDKは第三者が著作権を所有するライブラリを使用しております。ライセンスについては、LEAG-SDK インストールフォルダ/Documents/Licenses フォルダ内を参照ください。

DXマーカは、大日本印刷社の商標もしくは登録商標です。

6.4. 問い合わせ

LEAG-SDK に関するご質問は、以下サポート窓口にお問合せ下さい。

リーグソリューションズ株式会社

Mail info@leag.jp