TOPPAN ToF
senSPure® SDK ライブラリ
API リファレンスマニュアル

TOPPAN 3D ToF Camera

TOPPAN ホールディングス株式会社

Revision 1.12

2025 年 9 月 9 日

© 2025 TOPPAN Holdings Inc. All rights reserved.
TOPPAN Holdings Inc. reserves the right to change contents without notice.

Contents

1. 概要 9
1-1. 本書の目的 9
1-2. 用語、略語などの定義 9
1-3. 関連ドキュメント 9
1-4. SDK 構成 10
1-5. 動作環境 10
1-6. ホスト PC 推奨環境 11
1-6-1. プログラミング言語 11
2. TOPPAN ToF SDK API 利用方法 12
2-1. 提供ファイル 12
2-2. 関連ライブラリファイル 12
3. TOPPAN ToF SDK 提供 API 13
3-1. クラス一覧 13
3-2. 共通定義 13
3-2-1. 定数定義 13
3-2-1-1. 定数定義一覧 13
3-2-1-2. 定数定義詳細 13
3-2-1-2-1. SDK_VERSION 13
3-2-1-2-2. INVALID_DEPTH 14
3-2-1-2-3. SATURATION_DEPTH 14
3-2-2. 列挙体定義 14
3-2-2-1. 列挙体定義一覧 14
3-2-2-2. 列挙体定義詳細 14
3-2-2-2-1. Result 14
3-2-2-2-2. ImageKind 15
3-2-2-2-3. PcdKind 15
3-2-3. 構造体定義 16
3-2-3-1. 構造体定義一覧 16
3-2-3-2. 構造体定義詳細 16
3-2-3-2-1. Version 16
3-2-3-2-2. Point2d 16
3-2-3-2-3. Point3d 17
3-2-3-2-4. ImageFormat 17
3-2-3-2-4-1. ImageFormat::set 18
3-2-3-2-4-2. ImageFormat::isExist 19
3-2-3-2-5. Range 19
3-2-3-2-6. FrameError 19
3-2-3-2-7. ConvState 20
3-2-3-2-8. FrameInfo 20
3-2-3-2-9. ImageData 21
3-2-3-2-9-1. ImageData::resize 21
3-2-3-2-10. PcdData 22

3-2-3-2-10-1. PcdData::resize 22
3-2-3-2-11. Frame 22
3-2-4. 型定義 23
3-2-4-1. 型定義一覧 23
3-2-4-2. 型定義詳細 23
3-2-4-2-1. ImageFormats 23
3-2-4-2-2. Images 23
3-3. PipelineFramework 23
3-3-1. 概要 23
3-3-2. 提供機能 24
3-3-3. PipelineFramework 内部クラス一覧 24
3-3-3-1. クラス関係 25
3-3-4. 制御シーケンス 25
3-3-4-1. 初期化シーケンス 25
3-3-4-2. 情報取得＆動作モード切り替えシーケンス 26
3-3-4-3. 画像受信シーケンス（closed_pl = false） 27
3-3-4-4. 画像受信シーケンス（Closed Pipeline） 27
3-3-4-5. イベント通知シーケンス 28
3-3-4-6. ステータス通知シーケンス 28
3-3-4-7. 終了シーケンス 28
3-3-5. Pipeline 処理構成例 29
3-3-5-1. Pipeline 処理構成例 1：カメラ出力のファイル保存 29
3-3-5-2. Pipeline 処理構成例 2：画像処理後の画像表示 29
3-3-5-3. Pipeline 処理構成例 3：画像処理後の検知処理 30
3-3-5-4. Pipeline 処理構成例 4：検知結果の外部出力 31
3-3-6. PlFw 31
3-3-6-1. 概要 31
3-3-6-2. メソッド一覧 31
3-3-6-2-1. 状態遷移 32
3-3-6-3. メソッド詳細 32
3-3-6-3-1. PlFw::PlFw 32
3-3-6-3-2. PlFw::~PlFw 33
3-3-6-3-3. PlFw::getCamDeviceList 33
3-3-6-3-4. PlFw::addPlProc 33
3-3-6-3-5. PlFw::wakeupPl 34
3-3-6-3-6. PlFw::shutdownPl 35
3-3-6-3-7. PlFw::getCamProperty 35
3-3-6-3-8. PlFw::setCamProperty 36
3-3-6-3-9. PlFw::startCapture 37
3-3-6-3-10. PlFw::stopCapture 37
3-3-6-3-11. PlFw::getEvent 37
3-3-6-3-12. PlFw::releaseBuf 39
3-3-6-3-13. PlFw::notifyEvent 39
3-3-6-4. PlFw クラス用定義 40
3-3-6-4-1. 定数定義 40
3-3-6-4-1-1. 定数定義一覧 40
3-3-6-4-1-2. 定数定義詳細 40
3-3-6-4-1-2-1. PROC_PIPELINE 40
3-3-7. FrameData クラス 40
3-3-7-1. 概要 40

3-3-7-2. メソッド一覧 40
3-3-7-3. メソッド詳細 41
3-3-7-3-1. FrameData::getFrame 41
3-3-7-3-2. FrameData::getFrameExt 41
3-3-7-3-3. FrameData::getUserBuf 41
3-3-7-4. FrameData クラス用定義 42
3-3-7-4-1. 構造体定義 42
3-3-7-4-1-1. 構造体定義一覧 42
3-3-7-4-1-2. 構造体定義詳細 42
3-3-7-4-1-2-1. FrameExtInfo 42
3-3-8. EvtThread クラス 42
3-3-8-1. 概要 42
3-3-8-2. メソッド一覧 42
3-3-8-3. メソッド詳細 43
3-3-8-3-1. EvtThread::EvtThread 43
3-3-8-3-2. EvtThread::~EvtThread 43
3-3-8-3-3. EvtThread::getKind 43
3-3-8-3-4. EvtThread::enableFrameDrop 44
3-3-8-3-5. EvtThread::getMaxQueuingFrames 44
3-3-8-3-6. EvtThread::recvChgProp 45
3-3-8-3-7. EvtThread::recvChgFmt 45
3-3-8-3-8. EvtThread::recvImage 45
3-3-8-3-9. EvtThread::recvReset 46
3-3-8-3-10. EvtThread::recvUserEvent 46
3-3-8-4. クラス内変数一覧 47
3-3-8-5. EvtThread クラス用定義 47
3-3-8-5-1. 列挙体定義 47
3-3-8-5-1-1. 列挙体定義一覧 47
3-3-8-5-1-2. 列挙体定義詳細 47
3-3-8-5-1-2-1. ProcKind 47
3-3-8-5-2. 構造体定義 48
3-3-8-5-2-1. 構造体定義一覧 48
3-3-8-5-2-2. 構造体定義詳細 48
3-3-8-5-2-2-1. CameraProperty 48
3-3-8-6. 型定義 48
3-3-8-6-1. 型定義一覧 48
3-3-8-6-2. 型定義詳細 48
3-3-8-6-2-1. PlCbFunc 48
3-3-9. RecordThread クラス 49
3-3-9-1. 概要 49
3-3-9-2. イベント一覧 49
3-3-9-2-1. EV_REC_START 49
3-3-9-2-2. EV_REC_STOP 49
3-3-9-3. ステータス通知 49
3-3-9-4. RecordThread クラス用定義 50
3-3-9-4-1. 列挙体定義 50
3-3-9-4-1-1. 列挙体定義一覧 50
3-3-9-4-1-2. 列挙体定義詳細 50
3-3-9-4-1-2-1. RecEvent 50
3-3-9-4-2. 構造体定義 50

3-3-9-4-2-1. 構造体定義一覧 50
3-3-9-4-2-2. 構造体定義詳細 50
3-3-9-4-2-2-1. RecordParam 50
3-3-10. LensConvThread クラス 51
3-3-10-1. 概要 51
3-3-10-2. メソッド一覧 51
3-3-10-3. メソッド詳細 51
3-3-10-3-1. LensConvThread::LensConvThread 51
3-3-10-4. イベント一覧 52
3-3-10-4-1. EV_LENS_PSBL_DIST 52
3-3-10-4-2. EV_LENS_DIST 53
3-3-10-4-3. EV_LENS_PCD_KIND 53
3-3-10-4-4. EV_LENS_PCD_ORG_POS 53
3-3-10-4-5. EV_LENS_PCD_COLOR 53
3-3-10-5. ステータス通知 53
3-3-10-6. LensConvThread クラス用定義 54
3-3-10-6-1. 列挙体定義 54
3-3-10-6-1-1. 列挙体定義一覧 54
3-3-10-6-1-2. 列挙体定義詳細 54
3-3-10-6-1-2-1. LensConvEvent 54
3-3-10-6-1-2-2. PcdColorKind 54
3-4. Camera クラス 55
3-4-1. 概要 55
3-4-2. 提供機能 55
3-4-2-1. デバイス制御機能 55
3-4-2-2. ファイル再生機能 56
3-4-2-2-1. ファイル再生対応フォーマットバージョン 57
3-4-3. メソッド一覧 57
3-4-3-1. 状態遷移 57
3-4-3-2. 状態遷移（ファイル再生） 58
3-4-4. 制御シーケンス 58
3-4-4-1. 初期化シーケンス 59
3-4-4-2. 画像受信シーケンス 60
3-4-4-3. 終了シーケンス 60
3-4-5. メソッド詳細 61
3-4-5-1. Camera 61
3-4-5-2. ~Camera 61
3-4-5-3. getDeviceList 62
3-4-5-4. openDevice 62
3-4-5-5. closeDevice 62
3-4-5-6. getProperty 63
3-4-5-7. setProperty 63
3-4-5-8. startCapture 64
3-4-5-9. stopCapture 65
3-4-5-10. capture 65
3-4-5-11. cancel 66
3-4-6. プロパティコマンド（カメラデバイス用） 66
3-4-6-1. プロパティコマンド一覧 66
3-4-6-1-1. CMD_DEV_INFO 67
3-4-6-1-2. CMD_FOV 67

3-4-6-1-3. CMD_EXT_TRG_TYPE 67
3-4-6-1-4. CMD_EXT_TRG_OFFSET 68
3-4-6-1-5. CMD_MODE_LIST 68
3-4-6-1-6. CMD_MODE 68
3-4-6-1-7. CMD_IMG_KINDS 68
3-4-6-1-8. CMD_IMG_FORMAT 69
3-4-6-1-9. CMD_POSTFILT_INFO 69
3-4-6-1-10. CMD_LENS_INFO 69
3-4-6-1-11. CMD_LIGHT_TIMES 69
3-4-6-1-12. CMD_AE_STATE 69
3-4-6-1-13. CMD_AE_INTERVAL 70
3-4-6-1-14. CMD_RAW_SAT_TH 70
3-4-6-1-15. CMD_IR_DARK_TH 70
3-4-6-1-16. CMD_INT_SUPP_INFO 71
3-4-7. プロパティコマンド（ファイル再生専用） 71
3-4-7-1. プロパティコマンド一覧 71
3-4-7-1-1. PlayBack::CMD_PLAY_TARGET 72
3-4-7-1-2. PlayBack::CMD_PLAY_TIME 72
3-4-7-1-3. PlayBack::CMD_PLAY_STATUS 72
3-4-7-1-4. PlayBack::CMD_PAUSE 73
3-4-7-1-5. PlayBack::CMD_FAST_PLAY 73
3-4-7-1-6. PlayBack::CMD_SLOW_PLAY 73
3-4-7-1-7. PlayBack::CMD_JUMP_FW 73
3-4-7-1-8. PlayBack::CMD_JUMP_BW 74
3-4-8. Camera クラス用定義 74
3-4-8-1. 列挙体定義 74
3-4-8-1-1. 列挙体定義一覧 74
3-4-8-1-2. 列挙体定義詳細 74
3-4-8-1-2-1. CameraType 74
3-4-8-1-2-2. PropCmd 75
3-4-8-1-2-3. OperationMode 76
3-4-8-1-2-4. ExtTriggerType 76
3-4-8-1-2-5. ImgOutKind 77
3-4-8-1-2-6. CamPrmKind 77
3-4-8-1-2-7. RegDevType 77
3-4-8-1-2-8. IntSuppModeType 78
3-4-8-2. 構造体定義 78
3-4-8-2-1. 構造体定義一覧 78
3-4-8-2-2. 構造体定義詳細 79
3-4-8-2-2-1. ConnDevice 79
3-4-8-2-2-2. CalbOpInfo 79
3-4-8-2-2-3. OpInfo 79
3-4-8-2-2-4. DeviceInfo 80
3-4-8-2-2-5. PostFiltInfo 80
3-4-8-2-2-6. LensInfo 81
3-4-8-2-2-7. CamFov 81
3-4-8-2-2-8. ModeInfo 82
3-4-8-2-2-9. LightTimesInfo 82
3-4-8-2-2-10. AEIntervalInfo 83
3-4-8-2-2-11. SignalThresholdInfo 83

3-4-8-2-2-12. CamPrmRequest 83
3-4-8-2-2-13. RegDevInfo 84
3-4-8-2-2-14. RegInfo 84
3-4-8-2-2-15. RegList 84
3-4-8-2-2-16. IntSuppParamInfo 85
3-4-8-2-2-17. IntSuppInfo 85
3-4-8-2-2-18. UsbPCAccKey 85
3-4-8-3. 型定義 86
3-4-8-3-1. 型定義一覧 86
3-4-8-3-2. 型定義詳細 86
3-4-8-3-2-1. ModeList 86
3-4-8-3-2-2. RegDevs 86
3-4-9. ファイル再生機能用定義 86
3-4-9-1. 列挙体定義 86
3-4-9-1-1. 列挙体定義一覧 86
3-4-9-1-2. 列挙体定義詳細 86
3-4-9-1-2-1. PlayBack::PlayBackCmd 86
3-4-9-1-2-2. PlayBack::PlayState 87
3-4-9-2. 構造体定義 87
3-4-9-2-1. 構造体定義一覧 87
3-4-9-2-2. 構造体定義詳細 88
3-4-9-2-2-1. PlayBack::ConfigParam 88
3-4-9-2-2-2. PlayBack::PlayTime 88
3-4-9-2-2-3. PlayBack::PlayStatus 88
3-5. Record クラス 89
3-5-1. 概要 89
3-5-2. 提供機能 89
3-5-2-1. 機能概要 89
3-5-2-1-1. 保存先データ構成 89
3-5-2-1-1-1. Rec_YYYYMMDD_HHMMSS ディレクトリ 89
3-5-2-1-1-2. RecInfo.json 90
3-5-2-1-1-3. RecImage.raw 91
3-5-2-1-1-3-1. RecImage.raw – Frame Info 92
3-5-3. メソッド一覧 93
3-5-3-1. 状態遷移 93
3-5-4. 制御シーケンス 94
3-5-5. メソッド詳細 94
3-5-5-1. Record 94
3-5-5-2. ~Record 94
3-5-5-3. openRec (1) 95
3-5-5-4. openRec (2) 95
3-5-5-5. closeRec 96
3-5-5-6. recFrame 96
3-5-6. Record クラス用定義 97
3-5-6-1. 構造体定義 97
3-5-6-1-1. 構造体定義一覧 97
3-5-6-1-2. 構造体定義詳細 97
3-5-6-1-2-1. Record::ConfigParam 97
3-5-6-1-2-2. Record::RecInfoParam 98

3-6. LensConv クラス 99
3-6-1. 概要 99
3-6-2. 提供機能 99
3-6-2-1. 機能概要 99
3-6-2-1-1. 歪曲補正機能 99
3-6-2-1-2. 点群変換機能 99
3-6-2-1-2-1. カメラ座標 100
3-6-2-1-2-1-1. 世界座標 100
3-6-3. メソッド一覧 101
3-6-3-1. 状態遷移 101
3-6-4. 制御シーケンス 101
3-6-5. メソッド詳細 102
3-6-5-1. LensConv 102
3-6-5-2. ~LensConv 102
3-6-5-3. setLensPrm 103
3-6-5-4. setFormat 103
3-6-5-5. setPosOrgRotation 103
3-6-5-6. correctDist 104
3-6-5-7. convPcdCamera 104
3-6-5-8. convPcdWorld 105
3-6-6. LensConv クラス用定義 105
3-6-6-1. 構造体定義 105
3-6-6-1-1. 構造体定義一覧 105
3-6-6-1-2. 構造体定義詳細 105
3-6-6-1-2-1. PosOrgRotation 105
4. 使用上の制約事項 107
4-1. データ出力に関する制約事項 107
4-2. 外部トリガ種別 Secondary(Slave, EXT_TRG_SLAVE)時の制約事項 107
5. 使用条件・免責事項 108
改定履歴 108

1. 概要

1-1. 本書の目的

本書は、TOPPAN 製 *senSPure*®カメラ向けに開発された専用 SDK を利用する際のアプリケーション・プログラム・インターフェイス（API）仕様について説明するものです。本書では専用 SDK を *senSPure*® SDK または TOPPAN ToF SDK と表記されていますが、それらは同一のものを指しています。

現在、以下のカメラを動作対象としています。

Table 1. 動作対象カメラ

Model	Product code	Camera firmware バージョン
C11U	TPSC1AS1Z	3.1.0 以上

1-2. 用語、略語などの定義

Table 2. 用語、略語などの定義

用語、略語	定義
SDK	Software Development kit
ToF	Time of Flight

1-3. 関連ドキュメント

本書を参照する際には、以下のドキュメントを合わせて参照してください。

Table 3. 関連ドキュメント

関連文書	内容
TOPPAN ToF *senSPure*® SDK 開発環境構築ガイド	TOPPAN ToF *senSPure*® SDK ソフトウェアの環境構築方法
TOPPAN ToF *senSPure*® SDK PostFilter ライブラリ リファレンスガイド	TOPPAN ToF *senSPure*® SDK と協調動作するポストフィルタライブラリ

1-4. SDK 構成

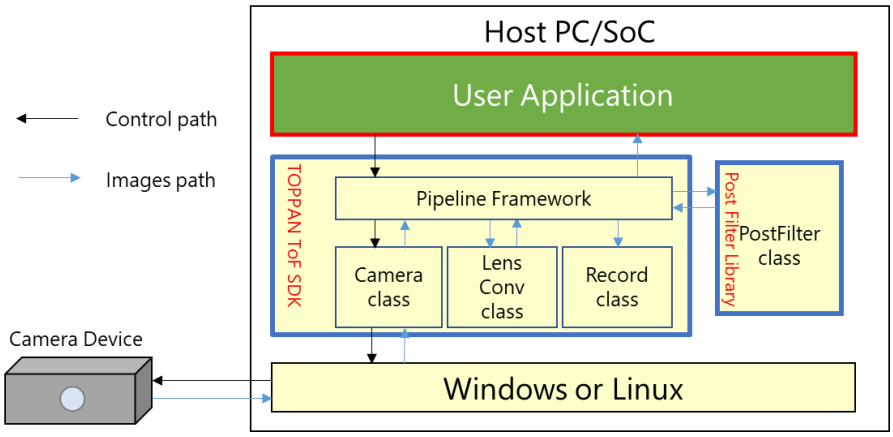


Figure 1. ソフトウェア構成図

Table 4. ブロック内容

ブロック		内容
User Application		本 SDK を制御するアプリケーション もしくは本 SDK 内に用意しているサンプルアプリケーション
TOPPAN ToF SDK (Library)		本 SDK として提供するライブラリ
	PipelineFramework	Camera クラス、Record クラス、LensConv クラスをまとめて管理し、これら以外の任意処理（OSS 連携処理や User 独自処理）を含めて Pipeline 処理を構築することができるフレームワーク
	Camera クラス	カメラデバイスの制御または保存されたファイルの再生を行う C++クラス
	LensConv クラス	カメラデバイスの出力画像に対してレンズに関わる変換処理を行う C++クラス
	Record クラス	カメラデバイスの出力画像をファイルに保存する C++クラス
PostFilter (Library)		本 SDK と協調動作するライブラリ 画像のフィルタ機能を提供する C++ クラス

1-5. 動作環境

本 SDK は下記の環境で動作確認を行っています。

Table 5. 動作環境

環境		OS 種類	バージョン
PC		Windows	Windows 10 64bit Windows 11 64bit
		Linux	Ubuntu 20.04LTS 64bit Ubuntu 22.04LTS 64bit
SoC	NVIDIA Jetson AGX Orin	Linux	JetPack 5.0.1 (Ubuntu 20.04LTS 64bit) JetPack 6.0 (Ubuntu 22.04LTS 64bit)

1-6. ホスト PC 推奨環境

本 SDK を動作させるホスト PC の推奨環境を以下に示します。

Table 6. 推奨環境

項目	推奨環境
CPU	4 コア 2GHz 以上, 64 ビット CPU
メモリ	8GB 以上
物理 I/F	専用の USB3.1(Gen1)ポート 搭載

PipelineFramework で複数のスレッド構成かつ高フレームレートでの処理が必要な場合は、CPU のコア数・周波数は上記よりも高スペックのホスト PC が必要になります。

1-6-1. プログラミング言語

本 SDK は C++（C++17 規格）で開発しています。ただし、MISRA-C++, CERT-C++には未対応です。

2. TOPPAN ToF SDK API 利用方法

2-1. 提供ファイル

本 SDK の API を利用する際には、使用するクラスに応じて、下記のヘッダファイルの include、ライブラリファイルのリンクを行ってください。

以下のファイルはビルド後に生成される include ディレクトリに格納されています。提供ディレクトリの構成は、"***1-3. 関連ドキュメント***"に記載している「TOPPAN ToF *senSPure*® SDK 環境構築ガイド」を参照してください。

Table 7. 提供ファイル

ファイル種別	ファイル名	
ヘッダファイル	共通定義	TpTofSdkDefine.h
	PlFw クラス	PlFw.h
	EvtThread クラス	EvtThread.h
	RecordThread クラス	RecordThread.h
	LensConvThread クラス	LensConvThread.h
	PostFilterThread クラス	PostFilterThread.h
	Camera クラス	Camera.h
	Record クラス	Record.h
	LensConv クラス	LensConv.h
	PostFilter クラス	PostFilter.h
ライブラリファイル	Windows	TpTofSdk.dll PostFilter.dll
	Linux	libTpTofSdk.so libPostFilter.so

2-2. 関連ライブラリファイル

本 SDK ライブラリ内部では次のライブラリファイルを参照しています。本 SDK ライブラリを使用するユーザプログラムでは下記ライブラリファイルのリンクが必要となります。

Table 8. 関連ライブラリ

OS 種類	関連ライブラリ
Windows	Windows SDK (DirectShow, WinUSB)
Linux	libusb
共通	OpenCV, boost

また、Windows 版には EWCLIB（http://insubaru.g1.xrea.com/ewclib/）を同梱して使用しています。

3. TOPPAN ToF SDK 提供 API

3-1. クラス一覧

本 SDK が API として提供するクラスの一覧を以下に示します。

*Table 9.*クラス一覧

クラス名	説明
PipelineFramework	Camera クラス、Record クラス、LensConv クラスをまとめて管理し、これら以外の任意処理（OSS 連携処理や User 独自処理）を含めて Pipeline 処理を構築することができるフレームワーク (参考: 3-3-3. PipelineFramework 内部クラス一覧)
Camera クラス	カメラデバイスの制御、 または保存されたファイルの再生を行う C++クラス ※PipelineFramework を使用する場合、本クラスは使用禁止です。
Record クラス	カメラデバイスの出力画像をファイルに保存する C++クラス ※PipelineFramework を使用する場合、本クラスは使用禁止です。
LensConv クラス	カメラデバイスの出力画像に対してレンズに関わる変換処理を行う C++クラス ※PipelineFramework を使用する場合、本クラスは使用禁止です。

3-2. 共通定義

TpTofSdkDefine.h に記載している本 SDK で共通して使用する定義を以下に示します。

3-2-1. 定数定義

3-2-1-1. 定数定義一覧

Table 10. 定数定義一覧

名称	説明
SDK_VERSION	SDK バージョン
INVALID_DEPTH	距離画像データ無効値（遠方）
SATURATION_DEPTH	距離画像データ無効値（近距離飽和）

3-2-1-2. 定数定義詳細

3-2-1-2-1. SDK_VERSION

Table 11. SDK_VERSION 定義

定義	const Version SDK_VERSION;
説明	・本 SDK のバージョン情報を示します。
参照	3-2-3-2-1. Version

3-2-1-2-2. INVALID_DEPTH

Table 12. INVALID_DEPTH 定義

定義	const uint16_t INVALID_DEPTH = 0xFFFFU;
説明	・距離画像データの無効値（遠方）を示します。
参照	3-2-2-2-2. ImageKind, 3-2-3-2-9. ImageData

3-2-1-2-3. SATURATION_DEPTH

Table 13. SATURATION_DEPTH 定義

定義	const uint16_t SATURATION_DEPTH = 0x0000U;
説明	・距離画像データの無効値（近距離飽和）を示します。
参照	3-2-2-2-2. ImageKind, 3-2-3-2-9. ImageData

3-2-2. 列挙体定義

3-2-2-1. 列挙体定義一覧

Table 14. 列挙体定義一覧

名称	説明
Result	戻り値
ImageKind	画像データの種別
PcdKind	点群データの種別

3-2-2-2. 列挙体定義詳細

3-2-2-2-1. Result

Table 15. Result 定義

定義	enum Result {
	SUCCESS = 0,
	CANCELED,
	REACH_EOF,
	ERR_INVALID_PTR,
	ERR_OVER_RANGE,
	ERR_BAD_ARG,
	ERR_BAD_STATE,
	ERR_NOT_EXIST,
	ERR_TIMEOUT,
	ERR_EMPTY,
	ERR_FULL,
	ERR_NOT_SUPPORT,
	ERR_SYSTEM
	};		

説明	・　各 API の戻り値を示します。		
値	名前	値	説明
	SUCCESS	0	成功
	CANCELED	1	待ち状態が解除された

	REACH_EOF	2	ファイル終端に到達
	ERR_INVALID_PTR	3	引数のポインタが不正
	ERR_OVER_RANGE	4	設定値が設定可能範囲外
	ERR_BAD_ARG	5	その他、不正引数
	ERR_BAD_STATE	6	状態遷移異常
	ERR_NOT_EXIST	7	デバイス・ファイル等が存在しない
	ERR_TIMEOUT	8	処理待ち中にタイムアウト発生
	ERR_EMPTY	9	バッファなどが空
	ERR_FULL	10	容量などが足りない
	ERR_NOT_SUPPORT	11	未サポート機能
	ERR_SYSTEM	12	システム異常（メモリ確保失敗など）
参照	All APIs		

3-2-2-2-2. ImageKind

Table 16. ImageKind 定義

定義	enum ImageKind : uint8_t {
	IMG_DEPTH = 0,
	IMG_IR,
	IMG_RAW1,
	IMG_RAW2,
	IMG_RAW3,
	IMG_RAW4,
	IMG_KINDS
	};		

説明	・ 画像データの種別を示します。		
値	名前	値	説明
	IMG_DEPTH	0	距離画像データ
	IMG_IR	1	IR 画像データ
	IMG_RAW1	2	センサ RAW G1 画像データ
	IMG_RAW2	3	センサ RAW G2 画像データ
	IMG_RAW3	4	センサ RAW G3 画像データ
	IMG_RAW4	5	センサ RAW G4 画像データ
	IMG_KINDS	6	画像種別数
参照	3-2-3-2-4. ImageFormats, 3-2-4-2-2. Images		

3-2-2-2-3. PcdKind

Table 17. PcdKind 定義

定義	enum PcdKind {
	PCD_XYZ = 0,
	PCD_RGBXYZ,
	PCD_IRXYZ,
	};		

説明	・ 点群データの種別を示します。		
値	名前	値	説明

	PCD_XYZ	0	色情報なし点群
	PCD_RGBXYZ	1	RGB 点群
	PCD_IRXYZ	2	IR 点群
参照	3-2-3-2-10. PcdData		

3-2-3. 構造体定義

3-2-3-1. 構造体定義一覧

Table 18. 構造体定義一覧

名称	説明
Version	バージョン情報
Point2d	画像平面上の画素位置情報
Point3d	3 次元空間内の座標情報
ImageFormat	画像データのフォーマット情報
Range	距離の範囲情報
FrameError	受信フレームの異常情報
ConvState	受信フレームの変換状態情報
FrameInfo	受信フレームの付加情報
ImageData	画像データ
PcdData	点群データ
Frame	受信フレーム情報

3-2-3-2. 構造体定義詳細

3-2-3-2-1. Version

Table 19. Version 定義

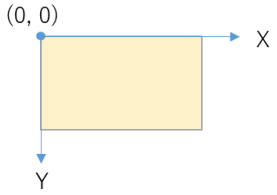
定義	struct Version {
	uint8_t major;
	uint8_t minor;
	uint16_t rev;
	};		

説明	・ バージョン情報を示します。		
引数	型	名前	説明
	uint8_t	major	Major version
	uint8_t	minor	Minor version
	uint16_t	rev	Revision
参照	3-2-1-2-1. SDK_VERSION, 3-4-8-2-2-4. DeviceInfo		

3-2-3-2-2. Point2d

Table 20. Point2d 定義

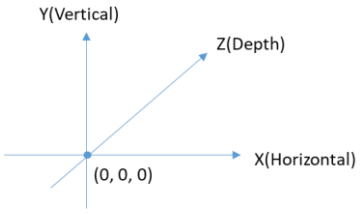
定義	struct Point2d { uint16_t x;

	uint16_t y; };		
説明	・ 画像平面上の画素位置を示します。 ・ 左上を原点とした画素位置となります。  		
引数	型	名前	説明
	uint16_t	x	X 軸方向座標 [pixel]
	uint16_t	y	Y 軸方向座標 [pixel]
参照	3-4-8-2-2-6. LensInfo		

3-2-3-2-3. Point3d

Table 21. Point3d 定義

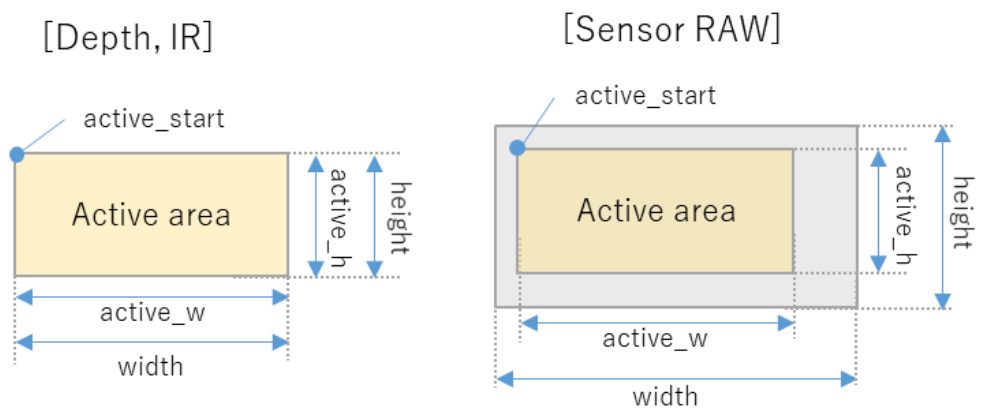
定義	struct Point3d {
	uint32_t color;
	float x;
	float y;
	float z;
	};		

説明	・ 3 次元空間内の座標情報を示します。  ・ RGB 点群または IR 点群の場合は、color が色情報を示します。		
引数	型	名前	説明
	uint32_t	color	UINT32_MAX の場合、無効点を示します。 RGB 点群の場合、上位 byte から 1byte 毎に Blue, Green, Red の値を示します。IR 点群の場合、16bit の IR 値を示します。
	float	x	X 軸方向座標 [mm]
	float	y	Y 軸方向座標 [mm]
	float	z	Z 軸方向座標 [mm]
参照	3-2-3-2-10. PcdData		

3-2-3-2-4. ImageFormat

Table 22. ImageFormat 定義

定義	struct ImageFormat { uint16_t width; uint16_t height; Point2d active_start; uint16_t active_w; uint16_t active_h;

<table>
<tr><td></td><td>uint32_t pixels;
uint8_t bpp;
size_t size;
};</td></tr>
<tr><td>説明</td><td>・ 画像データのフォーマット情報を示します。
・ Camera クラスからの出力としては、width, height で指定される全ての画素が出力されます。
画像種別がセンサ RAW（IMG_RAW1, IMG_RAW2, IMG_RAW3, IMG_RAW4）の場合、有効画素の情報（active_start, active_w, active_h）で指定した領域が有効画素領域として認識されます。
画像種別がセンサ RAW 以外の場合、active_start は常に(0, 0)に設定され、active_w, active_h はそれぞれ width, height と同じ値になります。すなわち、出力される全ての画素が有効画素となります。
（参考：3-4-2-1 デバイス制御機能）
 
</td></tr>
<tr><td>引数</td><td><table>
<tr><th>型</th><th>名前</th><th>説明</th></tr>
<tr><td>uint16_t</td><td>width</td><td>画像データ横幅 [pixel]</td></tr>
<tr><td>uint16_t</td><td>height</td><td>画像データ高さ [pixel]</td></tr>
<tr><td>Point2d</td><td>active_start</td><td>有効画素開始位置 [pixel]</td></tr>
<tr><td>uint16_t</td><td>active_w</td><td>有効画素横幅 [pixel]</td></tr>
<tr><td>uint16_t</td><td>active_h</td><td>有効画素高さ [pixel]</td></tr>
<tr><td>uint32_t</td><td>pixels</td><td>画像データ画素数（横幅×高さ）[pixel]</td></tr>
<tr><td>uint8_t</td><td>bpp</td><td>画像データ 1 画素のサイズ [byte]</td></tr>
<tr><td>size_t</td><td>size</td><td>画像データサイズ [byte]</td></tr>
</table></td></tr>
<tr><td>参照</td><td>3-2-4-2-1. ImageFormats</td></tr>
</table>

3-2-3-2-4-1.ImageFormat::set

Table 23. ImageFormat::set メソッド

<table>
<tr><td>機能</td><td>画像データフォーマット情報の設定</td></tr>
<tr><td>書式</td><td>inline void set (
uint16_t w = 0,
uint16_t h = 0,
uint8_t b = 0
);</td></tr>
<tr><td>説明</td><td>・ 画像フォーマット情報を設定します。
・ 引数省略時はフォーマット情報を初期化します。</td></tr>
</table>

	型	名前	in/out	説明
引数	uint16_t	w	in	画像データ横幅 [pixel]
	uint16_t	h	in	画像データ高さ [pixel]
	uint8_t	b	in	画像データ１画素のサイズ [byte]
戻り値	なし			
同期/非同期	同期型			

3-2-3-2-4-2.ImageFormat::isExist

Table 24. ImageFormat::isExist メソッド

機能	画像の有無取得			
書式	inline bool isExist (void);			
説明	・ 画像フォーマット情報が存在するか否かを取得します。			
引数	型	名前	in/out	説明
	なし			
戻り値	true	存在する		
	false	存在しない		
同期/非同期	同期型			

3-2-3-2-5. Range

Table 25. Range 定義

定義	struct Range {
	uint16_t min;
	uint16_t max;
	};		

説明	・ 距離の範囲情報を示します。		
引数	型	名前	説明
	uint16_t	min	最至近距離 [mm]
	uint16_t	max	最遠端距離 [mm]
参照	3-4-8-2-2-8. ModeInfo		

3-2-3-2-6. FrameError

Table 26. FrameError 定義

定義	struct FrameError {
	uint16_t drop : 1;
	uint16_t crc : 1;
	uint16_t reserved : 14;
	};			

説明	・ 受信フレームの異常情報を示します。 ・ 異常がある場合に各ビットに非 0 が入ります。			
引数	型		名前	説明
	uint16_t	1bit	drop	フレーム不連続情報 （0:連続、1:不連続）
	uint16_t	1bit	crc	CRC Error 情報 （0:非エラー、1:エラー）

	uint16_t	14bit	reserved	reserved for future
参照	3-2-3-2-8. FrameInfo			

3-2-3-2-7. ConvState

Table 27. ConvState 定義

定義	struct ConvState {
	uint8_t is_crct_dist : 1;
	uint8_t is_filt_med : 1;
	uint8_t is_filt_bil : 1;
	uint8_t is_filt_fly_p : 1;
	uint8_t reserved : 4;
	};

説明	・ 受信フレームの変換状態情報を示します。 ・ 歪曲補正されたデータの場合は is_crct_dist が 1 となります。 ・ Depth・IR 画像・点群データ以外の is_crct_dist は 0 となります。 ・ PostFilter のメディアンフィルタが適用されたデータの場合には is_filt_med が 1 となります。 ・ Depth・IR 画像・点群データ以外の is_filt_med は 0 となります。 ・ PostFilter のバイラテラルフィルタが適用されたデータの場合には is_filt_bil が 1 となります。 ・ Depth・IR 画像・点群データ以外の is_filt_bil は 0 となります。 ・ PostFilter のフライングピクセルフィルタが適用されたデータの場合には is_filt_fly_p が 1 となります。 ・ Depth 画像・点群データ以外の is_filt_fly_p は 0 となります。

引数	型		名前	説明
	uint8_t	1bit	is_crct_dist	歪曲補正済みか
	uint8_t	1bit	is_filt_med	メディアンフィルタを適用済みか
	uint8_t	1bit	is_filt_bil	バイラテラルフィルタを適用済みか
	uint8_t	1bit	is_filt_fly_p	フライングピクセルフィルタを適用済みか
	uint8_t	4bit	reserved	Reserved
参照	3-2-3-2-8. FrameInfo			

3-2-3-2-8. FrameInfo

Table 28. FrameInfo 定義

定義	struct FrameInfo {
	uint32_t number;
	timespec time;
	FrameError frm_err;
	int16_t temperature;
	uint32_t light_cnt;
	ConvState conv_stat;
	};

説明	・ 受信フレームの付加情報を示します。 ・ カメラデバイスでタイムスタンプ情報が付与されない場合は、本 SDK でフレームを受信した時間の情報が入ります。 ・ カメラデバイス内の温度情報が取得できない場合、temperature の値は UINT16_MAX となります。

	型	名前	説明
引数	uint32_t	number	フレーム番号
	timespec	time	タイムスタンプ情報
	FrameError	frm_err	フレーム異常情報
	uint16_t	temperature	カメラデバイス内温度情報 （固定小数点：整数部 10bit,小数部 6bit） （UINT16_MAX：無効温度値）
	uint32_t	light_cnt	発光回数値
	ConvState	conv_stat	変換状態情報
参照	3-2-3-2-6. FrameError, 3-2-3-2-7. ConvState, 3-2-3-2-9. ImageData, 3-2-3-2-10. PcdData		

3-2-3-2-9. ImageData

Table 29. ImageData 定義

定義	struct ImageData {
	FrameInfo info;
	std::vector<uint16_t> data;
	};		

説明	・ 下記画像種別の画像データを示します。 ・ Depth（IMG_DEPTH） ・ IR（IMG_IR） ・ センサ RAW （IMG_RAW1, IMG_RAW2, IMG_RAW3, IMG_RAW4） ・ 画像データのフォーマット情報は ImageFormat で定義されます。 ・ Depth の場合、無効画素には INVALID_DEPTH または SATURATION_DEPTH が設定されます。 ・ Depth の場合、値の単位は mm になります。		
引数	型	名前	説明
	FrameInfo	info	フレーム付加情報
	std::vector<uint16_t>	data	画像データ
参照	3-2-1-2-2. INVALID_DEPTH, 3-2-1-2-3. SATURATION_DEPTH, 3-2-2-2-2. ImageKind, 3-2-3-2-4. ImageFormat, 3-2-3-2-8. FrameInfo, 3-2-4-2-2. Images, 3-6-5-6. LensConv::correctDist, 3-6-5-7. LensConv::convPcdCamera, 3-6-5-8. LensConv::convPcdWorld		

3-2-3-2-9-1.ImageData::resize

Table 30. ImageData::resize メソッド

機能	画像データのリサイズ			
書式	inline void resize (const ImageFormat& format);			
説明	・ 画像フォーマット情報を元に画像データをリサイズします。			
引数	型	名前	in/out	説明
	const ImageFormat&	format	in	画像フォーマット
戻り値	なし			
同期/非同期	同期型			

3-2-3-2-10. PcdData

Table 31. PcdData 定義

定義	struct PcdData {
	FrameInfo info;
	PcdKind kind;
	std::vector<Point3d> data;
	};		

説明	・ 点群データを示します。 ・ 点群データ数の最大値は Depth と同じ画素数となりますが、実際に含まれる点群数は data.size()で取得して下さい。		
引数	型	名前	説明
	FrameInfo	info	フレーム付加情報
	PcdKind	kind	点群種別
	std::vector<Point3d>	data	点群データ
参照	3-2-3-2-8. FrameInfo, 3-2-2-2-3. PcdKind, 3-2-3-2-3. Point3d, 3-6-5-7. LensConv::convPcdCamera, 3-6-5-8. LensConv::convPcdWorld		

3-2-3-2-10-1.PcdData::resize

Table 32. PcdData::resize メソッド

機能	点群データのリサイズ			
書式	inline void resize (const ImageFormat& format);			
説明	・ Depth 画像フォーマット情報を元に点群データをリサイズします。			
引数	型	名前	in/out	説明
	const ImageFormat&	format	in	Depth 画像フォーマット
戻り値	なし			
同期/非同期	同期型			

3-2-3-2-11. Frame

Table 33. Frame 定義

定義	struct Frame {
	Images images;
	PcdData pcd;
	};		

説明	・ 受信フレーム情報（1 フレーム分）を示します。		
引数	型	名前	説明
	Images	images	全画像データ
	PcdData	pcd	点群データ
参照	3-2-4-2-2. Images, 3-2-3-2-10. PcdData, 3-4-5-10. Camera::capture, 3-5-5-6. Record::recFrame		

3-2-4. 型定義

3-2-4-1. 型定義一覧

Table 34. 型定義一覧

名称	説明
ImageFormats	全画像種別のフォーマット情報
Images	全画像種別の画像データ

3-2-4-2. 型定義詳細

3-2-4-2-1. ImageFormats

Table 35. ImageFormats 定義

定義	using ImageFormats = std::array<ImageFormat, IMG_KINDS>;
説明	・ 全画像種別のフォーマット情報を示します。
参照	3-2-2-2-2. ImageKind, 3-2-3-2-4. ImageFormat

3-2-4-2-2. Images

Table 36. Images 定義

定義	using Images = std::array<ImageData, IMG_KINDS>;
説明	・ 全画像種別の画像データを示します。
参照	3-2-2-2-2. ImageKind, 3-2-3-2-9. ImageData

3-3. PipelineFramework

3-3-1. 概要

Table 37. PipelineFramework 概要

提供ヘッダファイル	PlFw.h	PipelineFramework クラス定義
	FrameData.h	FrameData クラス定義
	EvtThread.h	EvtThread クラス定義
	RecordThread.h	RecordThread クラス定義
	RecordThreadEvent.h	RecordThread クラス用イベント型定義
	LensConvThread.h	LensConvThread クラス定義
	LensConvThreadEvent.h	LensConvThread クラス用イベント型定義
所属名前空間	krm	
説明	Pipeline 処理を構築するフレームワークおよび Pipeline で処理される各スレッド EvtThread クラスを継承した独自のクラスを用意することで独自処理を Pipeline 処理内の 1 スレッドとして組み込むことが可能となっています。	
スレッドセーフ	PipelineFramework クラスはスレッドセーフとなっています。 利用するユーザプログラム側で排他処理は必要ありません。	

3-3-2. 提供機能

PipelineFramework が提供する機能概要を以下に示します。

Table 38. PipelineFramework 内部提供機能

提供機能	機能概要
カメラデバイス制御機能	・ Camera クラスを使用したカメラデバイスの制御を行います。 ・ 本機能については下記の章を参照してください。 (3-4-2-1. デバイス制御機能)
ファイル再生機能	・ Camera クラス（PlayBack）を使用したファイル再生を行います。 ・ 本機能については下記の章を参照してください。 (3-4-2-2. ファイル再生機能)
ファイル保存機能	・ Record クラスを使用したファイル保存を行います。 ・ 本機能については下記の章を参照してください。 (3-5-2. 提供機能)
歪曲補正機能 点群変換機能	・ LensConv クラスを使用した歪曲補正、点群変換を行います。 ・ 本機能については下記の章を参照してください。 (3-6-2. 提供機能)
Pipeline処理機能	・ カメラデバイスから受信した画像データに対する各処理を Pipeline 処理として実行し、処理結果を出力する機能です。 ・ Pipeline 内の処理として実行する各処理の順番・構成は任意に決めることができます。ただし、実行順番に依存がある場合は処理順番を考慮して、使用するアプリケーションで Pipeline 処理の構成を決める必要があります。
ユーザスレッド追加機能	・ Pipeline 処理の中に任意の処理スレッドを追加することができる機能です。 ・ EvtThread を継承した任意処理を行うクラスを Pipeline 処理へ追加することができます。 ・ 画像データに対する画像処理の場合は LensConvThread より前、Point Cloud データに対する処理の場合は LensConvThread より後に配置してください。
イベント通知機能	・ Pipeline に追加した処理スレッドに対して、各処理スレッド独自のイベントを通知することができます。（パラメータの変更や処理の開始・停止など） ・ 処理スレッド毎の独自イベントの内容については各処理スレッドクラスの章を参照してください。
ステータス通知機能	・ 処理スレッド毎に異常発生や処理完了などのステータスを通知する機能です。 ・ 処理スレッド毎の通知内容については各処理スレッドクラスの章を参照してください。

3-3-3. PipelineFramework内部クラス一覧

PipelineFramework として使用する関連クラスの一覧を以下に示します。

Table 39. PipelineFramework 内部クラス一覧

クラス名	説明
PlFw	Camera クラス、Record クラス、LensConv クラスなどの単一機能を持つクラスを一元管理し、制御シーケンスや Pipeline による並列処理などをフレームワーク化した C++クラス 本クラスを制御することで、内包する各クラスの機能をまとめて制御することが可能です。

FrameData	１フレーム単位の画像・Point Cloud などのデータを管理する C++クラス
EvtThread	Pipeline 並列処理フレームワークを行うための基底クラスとなる C++クラス
RecordThread	Record クラスを用いた Pipeline 上の処理スレッドとなる C++クラス
LensConvThread	LensConv クラスを用いた Pipeline 上の処理スレッドとなる C++クラス

3-3-3-1. クラス関係

Pipeline Framework 内のクラスの関係を以下に示します。

Pipeline Framework(PlFw)クラスは Pipeline 処理として登録された各スレッド（LensConvThread, RecordThread や EvtThread を継承した独自 User Thread など）を制御して、カメラデバイスから受信した画像（FrameData）に対して、各種処理を Pipeline として実行します。

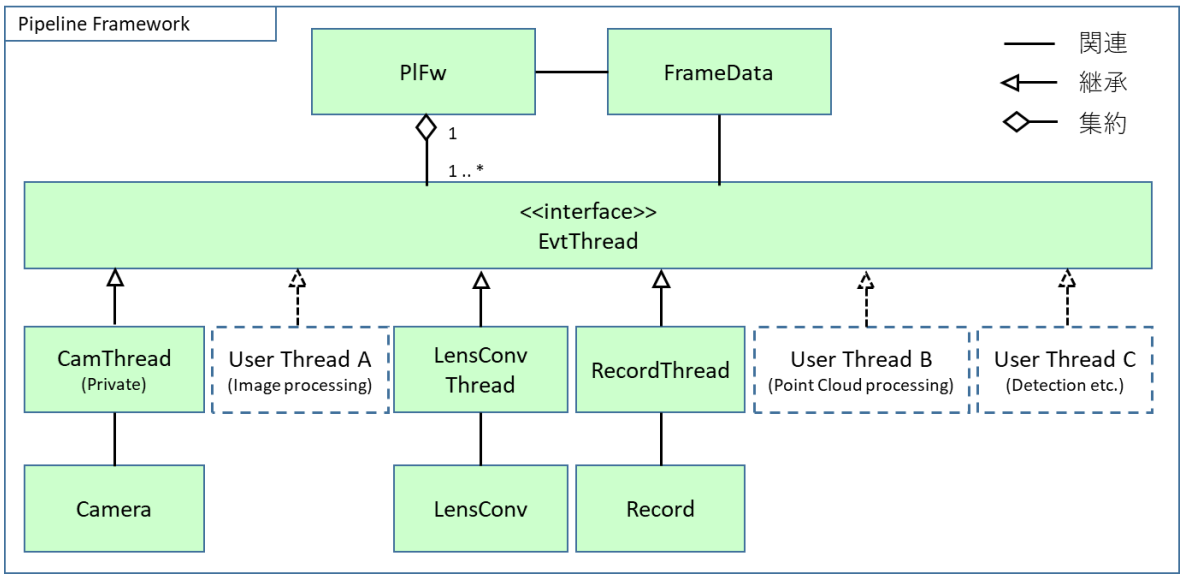


Figure 2. クラス関係図

3-3-4. 制御シーケンス

PipelineFramework クラスを用いたカメラ制御シーケンスを記載します。なお、以下のシーケンスに関しては関数の戻り値判定など異常系の判断は省略しております。

3-3-4-1. 初期化シーケンス

初期化時に、Pipeline 処理として、各スレッドを登録するシーケンスを以下に記載します。登録するスレッドについては使用用途に応じて変更してください。

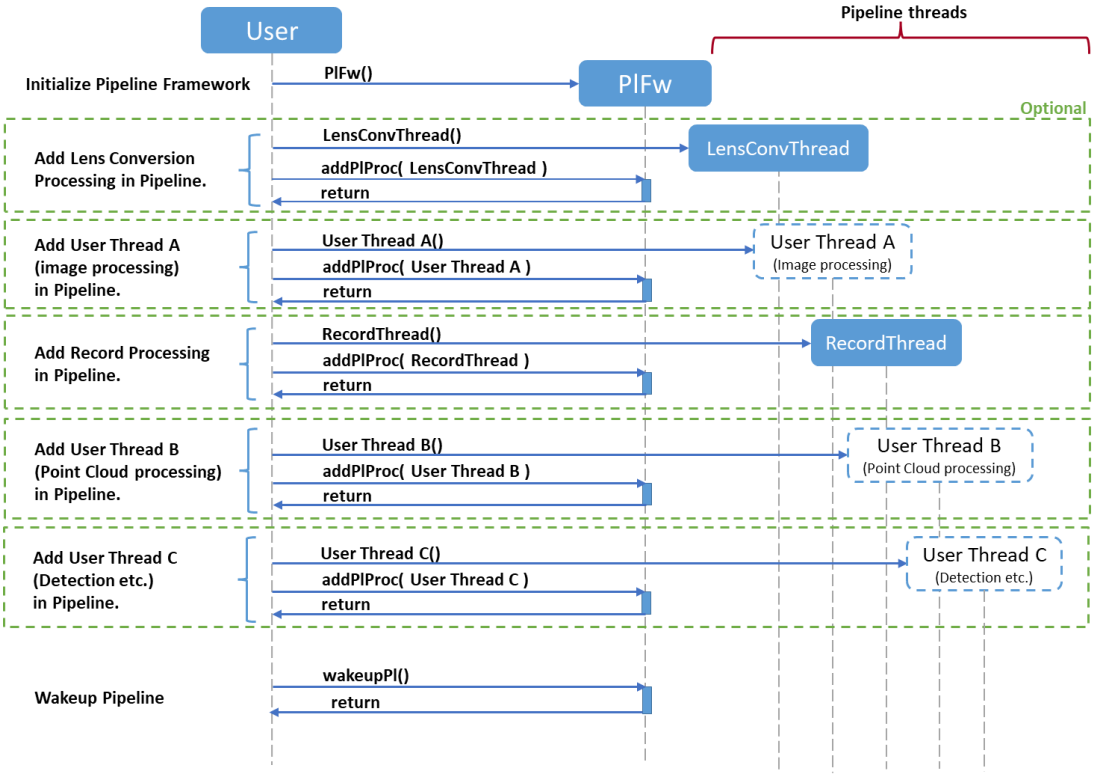


Figure 3. 初期化シーケンス図

3-3-4-2. 情報取得＆動作モード切り替えシーケンス

初期化後の情報取得、動作モード切り替えのシーケンスを以下に記載します。

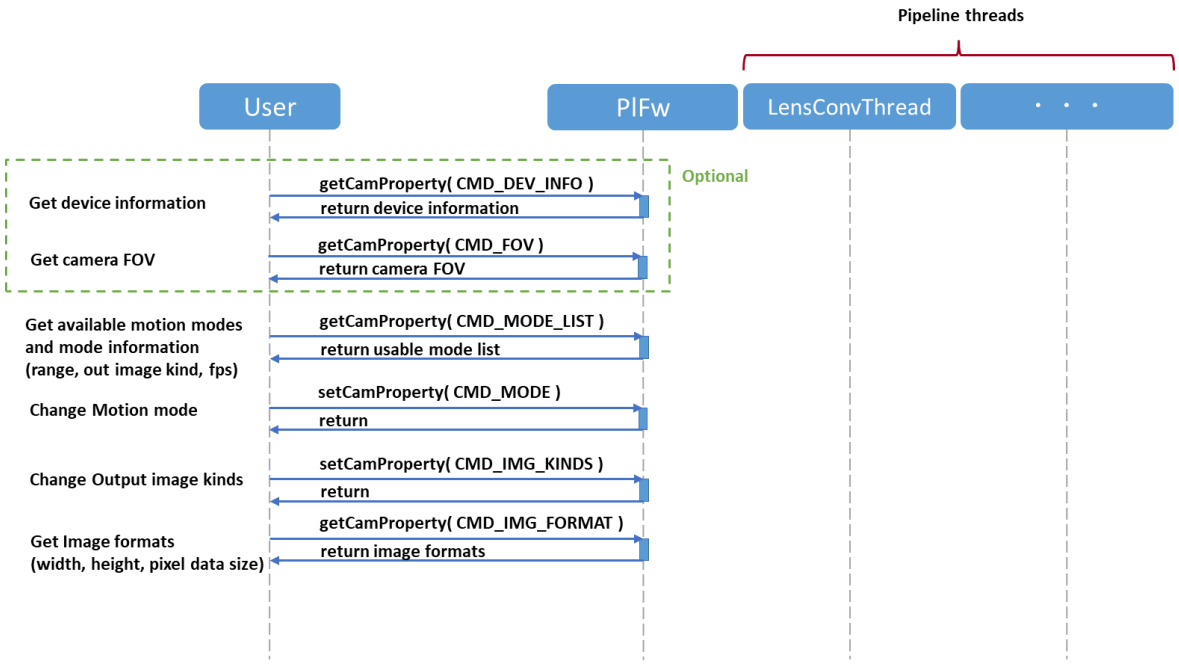


Figure 4. 情報取得＆動作モード切り替えシーケンス

3-3-4-3. 画像受信シーケンス（closed_pl = false）

PlFw::wakeupPl()の引数 closed_pl に false を設定した場合の画像受信シーケンスを以下に示します。このケースでは User Application は PlFw::getEvent()で画像の受け取りを行う必要があります。

なお、Pipeline として登録される各スレッドは全て同じメソッドが呼び出されるため、本シーケンス上は省略しています。

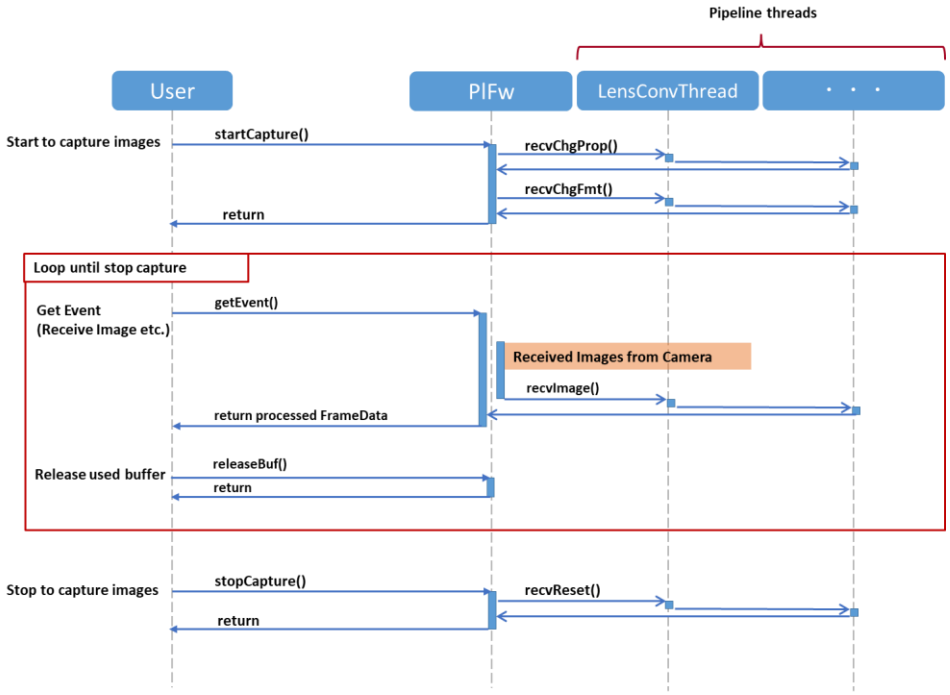


Figure 5. 画像受信シーケンス図（closed_pl = false）

3-3-4-4. 画像受信シーケンス（Closed Pipeline）

PlFw::wakeupPl()の引数 closed_pl に true を設定した場合の画像受信シーケンスを以下に示します。このケースでは User Application は PlFw::getEvent()で画像を受け取ることはできません。

Pipeline として登録される各スレッドは全て同じメソッドが呼び出されるため、本シーケンス上は省略しています。

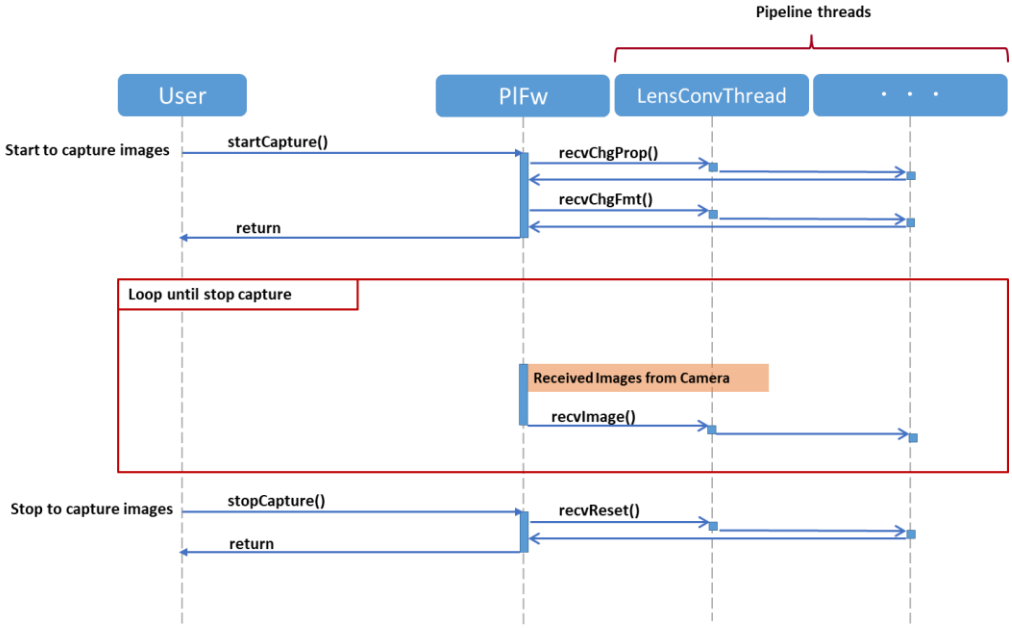


Figure 6. 画像受信シーケンス図（Closed Pipeline）

3-3-4-5. イベント通知シーケンス

PlFw::notifyEvent()を使用したイベント通知シーケンスを以下に示します。
例として RecordThread に対する保存開始・保存停止イベントの通知について記載します。RecordThread 以外も同様の形式になります。

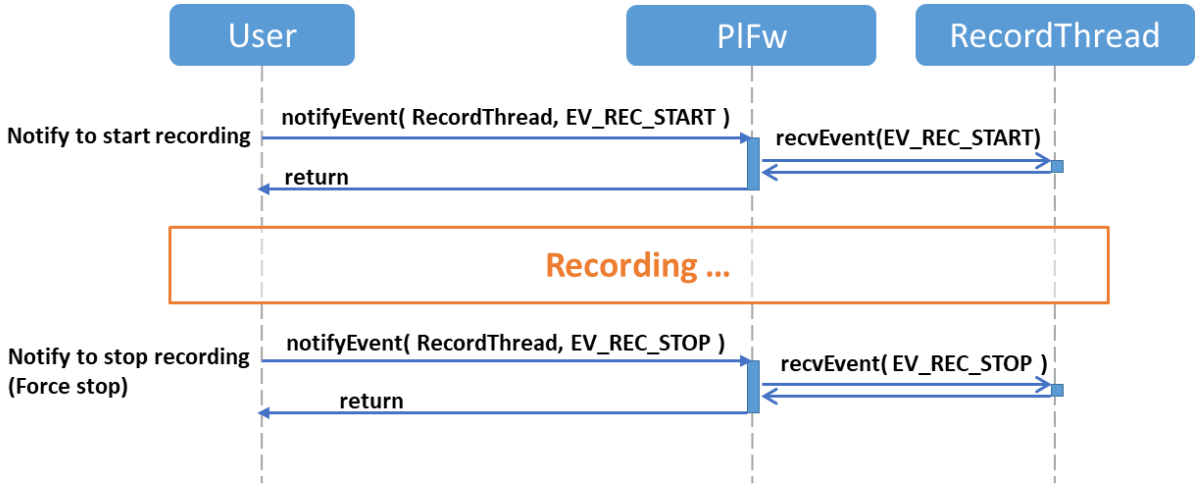


Figure 7. イベント通知シーケンス図（例：RecordThread）

3-3-4-6. ステータス通知シーケンス

Pipeline 処理に登録したスレッドからのステータス通知シーケンスを以下に示します。
例として RecordThread に対する保存開始後の保存完了のステータス通知について記載します。RecordThread 以外も同様の形式になります。

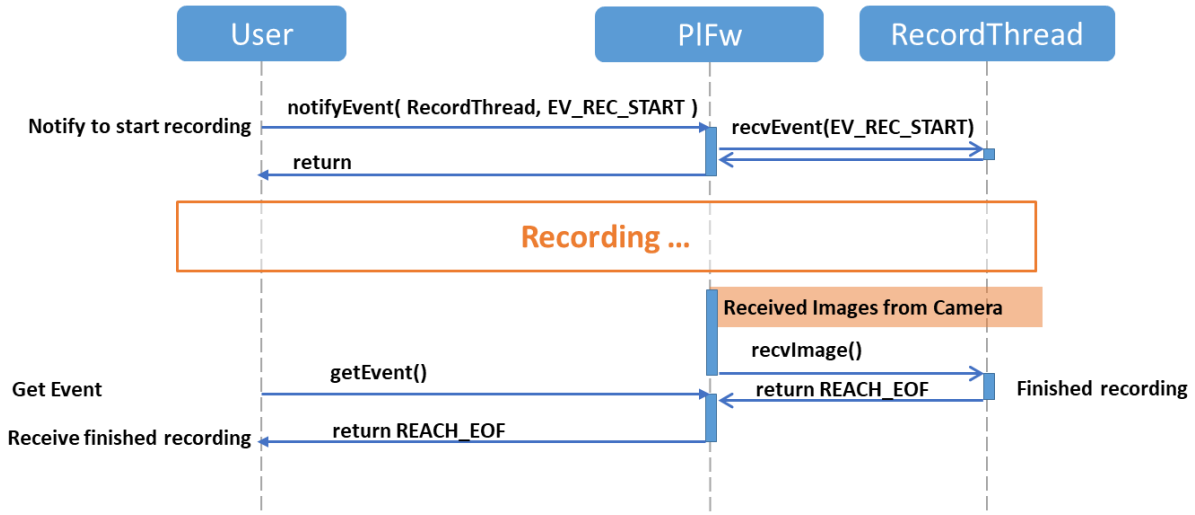


Figure 8. ステータス通知シーケンス図（例：RecordThread）

3-3-4-7. 終了シーケンス

終了時に、シーケンスを以下に記載します。Pipeline として登録される各スレッドは全て同じ形になるため、本シーケンス上は省略します。

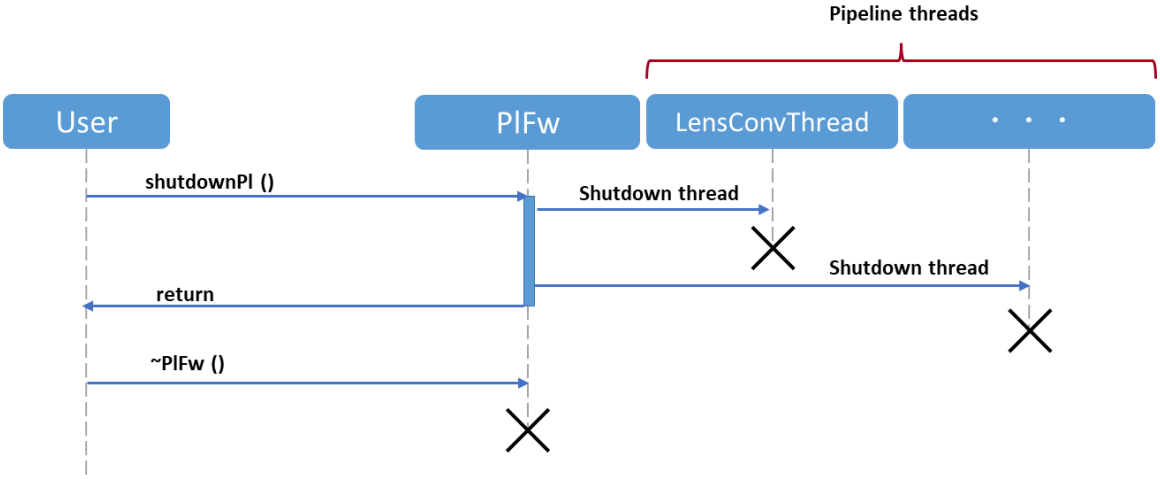


Figure 9. 終了シーケンス図

3-3-5. Pipeline処理構成例

Pipeline Framework では、用意されている RecordThread、LensConvThread や EvtThread を継承した独自の User Thread を組み合わせて Pipeline 処理を構築することができます。
構成例を以下に記載します。

3-3-5-1. Pipeline 処理構成例 1：カメラ出力のファイル保存

Pipeline 処理として RecordThread のみを登録し、Camera からの出力画像をファイルとして保存する機能のみを有した構成例です。
PlFw::wakeupPl()を実行する際に closed_pl=true とすることで、Application では画像データに対する処理を省くことができます。

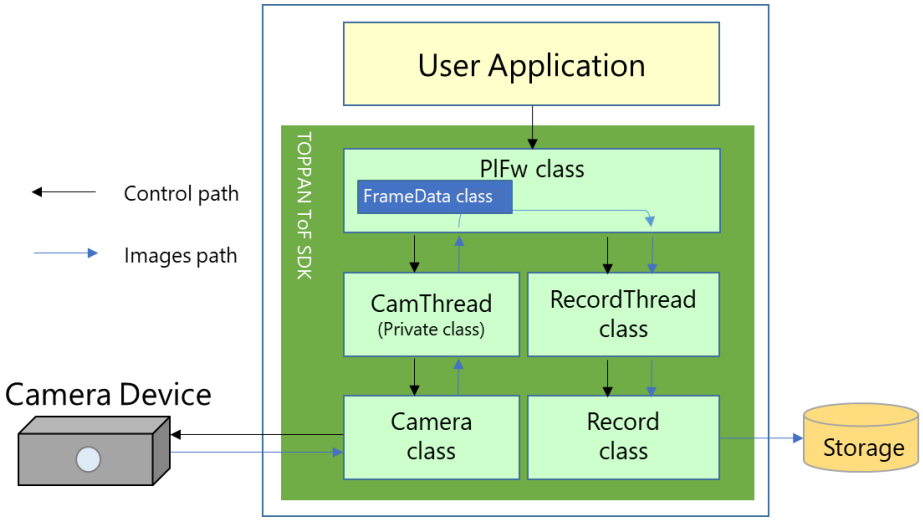


Figure 10. Pipeline 処理構成例 1：カメラ出力のファイル保存

3-3-5-2. Pipeline 処理構成例 2：画像処理後の画像表示

Pipeline 処理として LensConvThread のみを登録し、Camera からの出力画像に対して歪曲補正、PointCloud 変換を行った結果を画面に表示する構成例です。

PlFw::wakeupPl()を実行する際に closed_pl=false とすることで、Application で Pipeline 処理後の画像データを受け取ることができます。

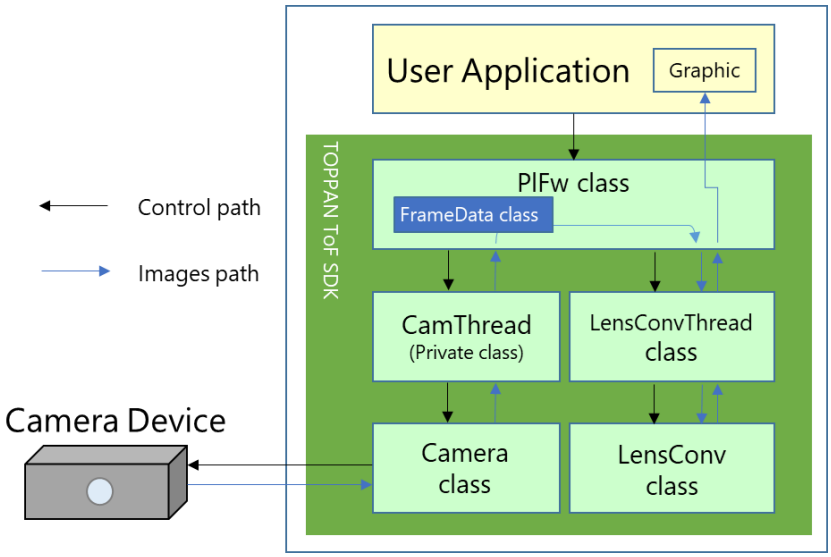


Figure 11. Pipeline 処理構成例 2：画像処理後の画像表示

3-3-5-3. Pipeline 処理構成例 3：画像処理後の検知処理

Pipeline 処理として独自の画像処理、LensConvThread、独自の検知処理を登録し、検知結果を画面に表示する構成例です。

EvtThread を継承した独自の画像処理を User Thread A（仮称）、独自の検知処理を User Thread B（仮称）として用意し、それぞれを Pipeline 処理へ登録することで、一連の処理を Pipeline としてまとめて制御することが可能です。

PlFw::addPlProc()を実行する際に追加バッファサイズを指定することで、画像データを管理している FrameData クラスに独自領域を用意することができ、画像データに合わせて検知結果などを受け渡すことができます。

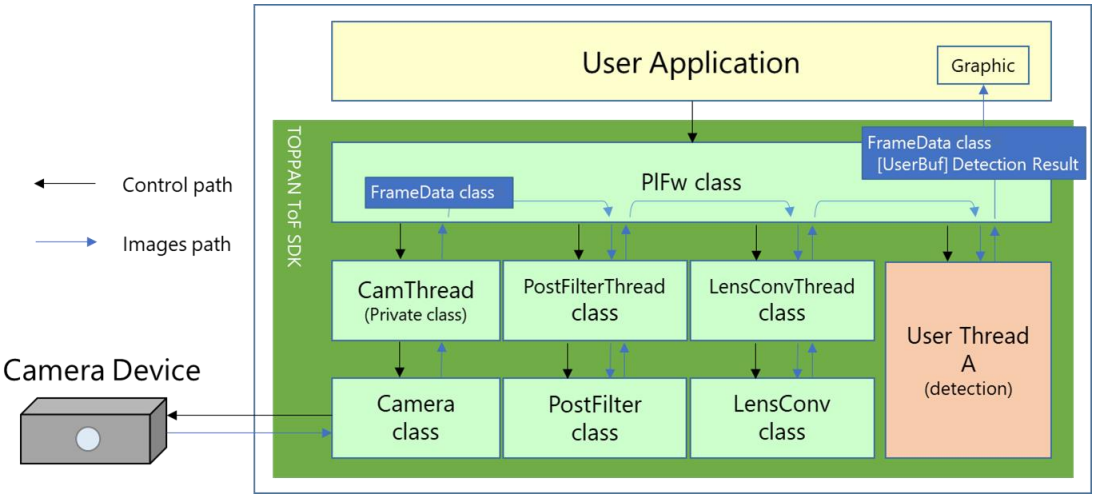


Figure 12. Pipeline 処理構成例 3：画像処理後の検知処理

3-3-5-4. Pipeline 処理構成例 4：検知結果の外部出力

「Pipeline 処理構成例 3：画像処理後の検知処理」の構成に対して検知結果を画面ではなく、外部機器へ出力する際の構成例です。

EvtThread を継承した外部機器へ送出する処理を User Thread C（仮称）として、Pipeline 処理へ追加することで、外部機器への出力までを一連の処理を Pipeline 処理でまとめて実行することができます。

なお、この構成時に外部機器からの制御信号の受信も必要な場合は、User Application で受信処理を行う必要があります。

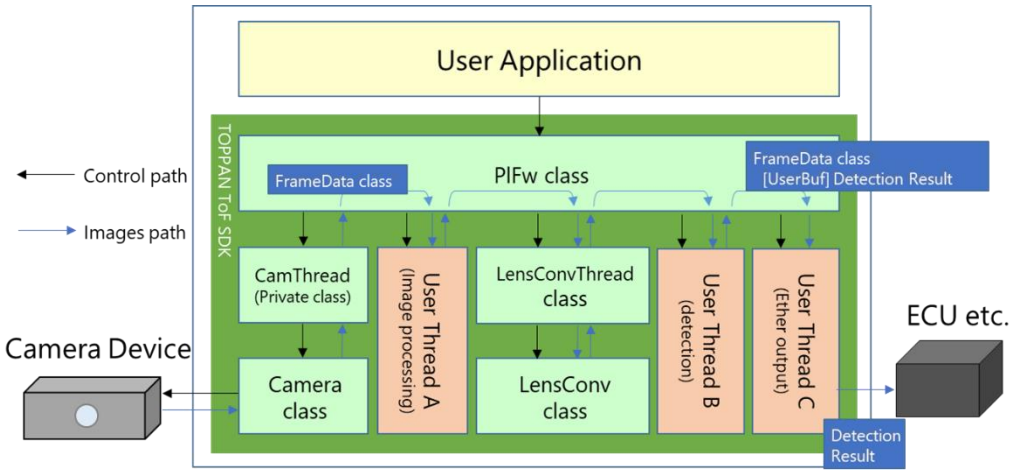


Figure 13. Pipeline 処理構成例 4：検知結果の外部出力

3-3-6. PIFw

3-3-6-1. 概要

Table 40. PIFw クラス概要

クラス名	PIFw
所属名前空間	Krm
説明	PipelineFramework の代表クラス

3-3-6-2. メソッド一覧

Table 41. PipelineFramework クラスメソッド

メソッド名	説明
PIFw::PIFw	コンストラクタ
PIFw::~PIFw	デストラクタ
PIFw::getCamDeviceList	接続しているカメラデバイスリストの取得
PIFw::addPIProc	Pipeline 処理の追加
PIFw::wakeupPI	Pipeline 処理の起動
PIFw::shutdownPI	Pipeline 処理の終了
PIFw::getCamProperty	カメラデバイスパラメータ取得
PIFw::setCamProperty	カメラデバイスパラメータ設定
PIFw::startCapture	画像出力開始

PlFw::stopCapture	画像出力停止
PlFw::getEvent	Pipeline 処理後の出力画像またはステータス通知の受信
PlFw::releaseBuf	出力画像の解放
PlFw::notifyEvent	Pipeline 処理への情報通知

3-3-6-2-1. 状態遷移

PlFw クラスを使用した PipelineFramework は以下の図に示す状態遷移となります。

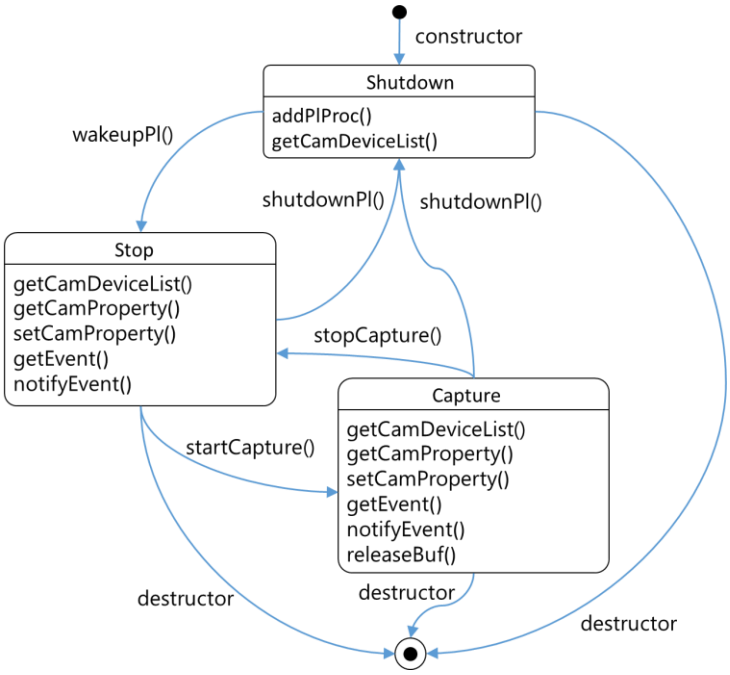


Figure 14. PipelineFramework 状態遷移

3-3-6-3. メソッド詳細

3-3-6-3-1. PlFw::PlFw

Table 42. PlFw::PlFw メソッド

機能	コンストラクタ			
書式	PlFw (CameraType type, const void* param, Result& res)			
説明	・ Pipeline framework の初期化および Camera 制御の初期化を行います。 ・ 引数 type が規定値外の場合、引数 res に ERR_OVER_RANGE が返ります。 ・ 3D ToF カメラ（C11U）を使用する際には引数 type に C11U_USB、引数 param に nullptr を指定してください。 ・ ファイル再生機能を使用する際には引数 type に PLAYBACK、引数 param に PlayBack::ConfigParam*型を指定してください。再生対象のディレクトリパスは PlFw::setCamProperty(PlayBack::CMD_PLAY_TARGET)でも再設定が可能です。			
引数	**型**	**名前**	**in/out**	**説明**
	CameraType	type	in	使用するカメラ種別
	const void*	param	in	デバイスパラメータ

	Result&	res	out	戻り値
戻り値	SUCCESS	0	成功	
	ERR_INVALID_PTR	3	引数のポインタが不正（PLAYBACK 指定時）	
	ERR_OVER_RANGE	4	カメラ種別が不正	
	ERR_SYSTEM	12	メモリ確保失敗	
同期/非同期	同期型			

3-3-6-3-2. PlFw::~PlFw

Table 43. PlFw::~PlFw メソッド

機能	デストラクタ			
書式	~PlFw (void)			
説明	・ Pipeline framework の初期化および Camera 制御の終了処理を行います。 ・ 本処理にて内部で確保したリソース開放を行うため、終了する際は必ずデストラクタが実行されるように設計してください。 ・ PlFw::wakeupPl()実行後、PlFw::shutdownPl()が実行されていない場合は本メソッド内で PlFw::shutdownPl()が実行されます。 ・ PlFw::startCapture()実行後、PlFw::stopCapture()が実行されていない場合は本メソッド内で PlFw::stopCapture()が実行されます。 ・ PlFw::getEvent()で受信した画像バッファが PlFw::releaseBuf()によって解放されていない場合は、本メソッドで PlFw::releaseBuf()が実行されます。			
引数	型	名前	in/out	説明
	なし			
戻り値	なし			
同期/非同期	同期型			

3-3-6-3-3. PlFw::getCamDeviceList

Table 44. PlFw::getDeviceList メソッド

機能	接続しているカメラデバイスリストの取得			
書式	Result getCamDeviceList (std::vector<ConnDevice>& dev_list)			
説明	・ 接続しているカメラデバイスのリストを取得します。 ・ 本メソッドは Camera::getDeviceList()と同一の内容になります。			
引数	型	名前	in/out	説明
	std::vector<ConnDevice>&	dev_list	out	接続している カメラデバイスリスト
戻り値	SUCCESS	0	成功	
	ERR_BAD_STATE	6	状態遷移異常	
	ERR_NOT_EXIST	7	対象デバイス未接続	
	ERR_SYSTEM	12	システム異常（メモリ確保失敗など）	
同期/非同期	同期型			

3-3-6-3-4. PlFw::addPlProc

Table 45. PlFw::addPlProc メソッド

機能	Pipeline 処理の追加

書式	Result addPlProc (std::shared_ptr<EvtThread>& proc, size_t ex_buf_size, uint16_t& proc_id)
説明	・ Pipeline 処理に任意の処理を追加します。 ・ 本メソッドで追加された順で Pipeline 処理が行われます。 ・ 引数 proc にはコンストラクタが実行された処理スレッドのオブジェクトを指定してください。 ・ 引数 ex_buf_size にはフレーム情報として追加するバッファのサイズを指定してください。追加バッファが不要な場合は 0 を指定して下さい。追加バッファは連続アドレスとして確保されるため、追加バッファ内で可変長データやポインタは使用できません。 ・ 成功時は、引数 proc_id に追加した処理 ID が返ります。この処理 ID は PlFw::getEvent()で通知されるイベントの通知元の判別、PlFw::notifyEvent()でイベントを通知する際の通知先、FrameData::getUserBuf()の引数として使用します。 ・ 登録された処理スレッドは本クラスのデストラクタで解放されます。 ・ PlFw::wakeupPl()を実行した後で本メソッドが呼ばれた場合は、ERR_BAD_STATE が返ります。 ・ 本メソッドで追加できる処理数は 65534 までです。上限を超える場合は ERR_FULL が返ります。

引数	型	名前	in/out	説明
	std:: shared_ptr<EvtThread>&	proc	in	追加する処理スレッド
	size_t	ex_buf_size	in	追加バッファサイズ
	uint16_t&	proc_id	out	処理 ID

戻り値			
	SUCCESS	0	成功
	ERR_INVALID_PTR	3	引数のポインタが不正
	ERR_BAD_STATE	6	状態遷移異常
	ERR_FULL	10	登録可能上限（65534）を超えた
	ERR_SYSTEM	12	システム異常（メモリ確保失敗など）

同期/非同期	同期型

3-3-6-3-5. PlFw::wakeupPl

Table 46. PlFw::wakeupPl メソッド

機能	Pipeline 処理の起動
書式	Result wakeupPl (uint16_t dev_id, bool closed_pl = false, const PlCbFunc notify_cb = nullptr)
説明	・ Pipeline 処理を起動し、カメラデバイスとの接続を確立します。 ・ 引数 dev_id には PlFw::getCamDeviceList()で取得したデバイス ID を指定してください。 ・ 引数 closed_pl が true の場合、本クラス内で閉じた Pipeline 処理となり、PlFw::getEvent()は使用できなくなります。 ・ 引数 notify_cb には closed_pl が true の場合のみコールバック関数を登録してください。受信状態に異常が発生した場合や各スレッドからのステータス通知が発生し

	た場合、登録されたコールバックに PlFw::getEvent()と同様の返り値が通知されます。 ・ 引数 closed_pl が false の場合、本クラス内で Pipeline 処理を行った後、PlFw::getEvent()に処理後のデータが通知されます。 ・ 本メソッド実行後、PlFw::shutdownPl()が実行されない状態で再度本メソッドが実行された場合、何も処理せずに SUCCESS が返ります。			
引数	型	名前	in/out	説明
	uint16_t	dev_id	in	デバイス ID
	bool	closed_pl	in	閉鎖 Pipeline フラグ true : closed Pipeline false : enable PlFw::getEvent()
	const PlCbFunc	notify_cb	in	状態通知コールバック
戻り値	SUCCESS	0	成功	
	ERR_NOT_EXIST	7	対象デバイス未接続	
	ERR_BAD_STATE	6	状態遷移異常	
	ERR_NOT_SUPPORT	11	動作対象カメラの Firmware バージョンが動作対象外	
	ERR_SYSTEM	12	システム異常（メモリ確保失敗など）	
同期/非同期	同期型			

3-3-6-3-6. PlFw::shutdownPl

Table 47. PlFw::shutdownPl メソッド

機能	Pipeline 処理の終了			
書式	Result shutdownPl (void)			
説明	・ カメラデバイスとの接続を切断し、Pipeline 処理を終了します。 ・ 本メソッド実行後、PlFw::getEvent()には CANCELED が返ります。 ・ PlFw::wakeupPl()が実行されていない状態で、本メソッドが実行された場合、何も処理せずに SUCCESS が返ります。 ・ PlFw::startCapture()実行後、PlFw::stopCapture()が実行されていない場合は本メソッド内で PlFw::stopCapture()が実行されます。			
引数	型	名前	in/out	説明
	なし			
戻り値	SUCCESS	0	成功	
	ERR_BAD_STATE	6	状態遷移異常	
	ERR_SYSTEM	12	システム異常（メモリ確保失敗など）	
同期/非同期	同期型			

3-3-6-3-7. PlFw::getCamProperty

Table 48. PlFw::getCamProperty メソッド

機能	カメラデバイスパラメータ取得
書式	Result getCamProperty (uint16_t prop_cmd, void* param)
説明	・ カメラデバイスのパラメータを取得します。

	・ 本メソッドは Camera::getProperty()と同一の内容になります。引数・戻り値については Camera クラスの getProperty()およびプロパティコマンドの章を参照してください。 ・ PlFw::wakeupPl()が実行されていない状態で本メソッドが呼ばれた場合は ERR_BAD_STATE が返ります。			
引数	型	名前	in/out	説明
	uint16_t	prop_cmd	in	プロパティコマンド（カメラデバイス用） プロパティコマンド（ファイル再生専用）
	void*	param	in,out	取得パラメータ
戻り値	SUCCESS	0	成功	
	ERR_INVALID_PTR	3	引数のポインタが不正	
	ERR_OVER_RANGE	4	設定値が設定可能範囲外（prop_cmd）	
	ERR_BAD_STATE	6	状態遷移異常	
	ERR_TIMEOUT	8	カメラデバイス間の通信でタイムアウト発生	
	ERR_NOT_SUPPORT	11	未サポート機能	
	ERR_SYSTEM	12	システム異常	
同期/非同期	同期型			

3-3-6-3-8. PlFw::setCamProperty

Table 49. PlFw::setCamProperty メソッド

機能	カメラデバイスパラメータ設定			
書式	Result setCamProperty (uint16_t prop_cmd, const void* param = nullptr)			
説明	・ カメラデバイスのパラメータを設定します。 ・ 本メソッドは Camera::setProperty()と同一の内容になります。引数・戻り値については Camera ライブラリ API 仕様書の setProperty()およびプロパティコマンドの章を参照してください。 ・ PlFw::wakeupPl()が実行されていない状態で本メソッドが呼ばれた場合は ERR_BAD_STATE が返ります。			
引数	型	名前	in/out	説明
	uint16_t	prop_cmd	in	プロパティコマンド（カメラデバイス用） プロパティコマンド（ファイル再生専用）
	const void*	param	in,out	設定パラメータ
戻り値	SUCCESS	0	成功	
	ERR_INVALID_PTR	3	引数のポインタが不正	
	ERR_OVER_RANGE	4	設定値が設定可能範囲外	
	ERR_BAD_ARG	5	その他、不正引数	
	ERR_BAD_STATE	6	状態遷移異常	
	ERR_NOT_EXIST	7	対象ディレクトリ・ファイルが存在しない	
	ERR_TIMEOUT	8	カメラデバイス間の通信でタイムアウト発生	
	ERR_NOT_SUPPORT	11	未サポート機能	
	ERR_SYSTEM	12	システム異常	

同期/非同期	同期型

3-3-6-3-9. PlFw::startCapture

Table 50. PlFw::startCapture メソッド

機能	画像出力開始			
書式	Result startCapture (void)			
説明	・ カメラデバイスの画像出力を開始します。 ・ PlFw::wakeupPl()で引数 closed_pl を false にしていた場合、画像出力開始後は、PlFw::getEvent()にて画像が受信されます。 ・ PlFw::wakeupPl()が実行されていない状態で本メソッドが呼ばれた場合は ERR_BAD_STATE が返ります。 ・ 既にカメラデバイスからの画像出力がされている状態で本メソッドが呼ばれた場合は何も処理せず SUCCESS が返ります。			
引数	型	名前	in/out	説明
	なし			
戻り値	SUCCESS	0	成功	
	ERR_BAD_STATE	6	状態遷移異常	
	ERR_NOT_EXIST	7	対象デバイス切断（再生対象ファイルがない）	
	ERR_SYSTEM	12	システム異常	
同期/非同期	同期型			

3-3-6-3-10. PlFw::stopCapture

Table 51. PlFw::stopCapture メソッド

機能	画像出力停止			
書式	Result stopCapture (void)			
説明	・ カメラデバイスの画像出力を停止します。 ・ PlFw::wakeupPl()が実行されていない状態で本メソッドが呼ばれた場合は ERR_BAD_STATE が返ります。 ・ PlFw::startCapture()が実行されていない状態で本メソッドが呼ばれた場合は何も処理せず SUCCESS が返ります。			
引数	型	名前	in/out	説明
	なし			
戻り値	SUCCESS	0	成功	
	ERR_BAD_STATE	6	状態遷移異常	
	ERR_SYSTEM	12	システム異常	
同期/非同期	同期型			

3-3-6-3-11. PlFw::getEvent

Table 52. PlFw::capture メソッド

機能	Pipeline 処理後の出力画像またはステータス通知の受信
書式	Result getEvent (uint16_t& proc_id, FrameData** frame, bool block = true)

説明

- カメラデバイスの出力画像に対して Pipeline で構成された処理を行った画像を取得します。
- 引数 proc_id が PROC_PIPELINE の場合は、Pipeline 処理後の出力画像が通知され、引数 frame に出力画像が格納されます。引数 frame の出力画像を使用した後は必ず PlFw::releaseBuf()で解放してください。
- 引数 proc_id が PROC_PIPELINE 以外の場合は、PlFw::addPlProc()で追加した処理スレッドの処理 ID が入り、該当処理スレッドからのステータス通知が戻り値として返ります。通知される戻り値については各処理スレッドの章を参照してください。
- PlFw::wakeupPl()が実行されていない状態で本メソッドが呼ばれた場合は ERR_BAD_STATE が返ります。
- PlFw::wakeupPl()の引数 closed_pl が true を指定されている場合、本メソッドは ERR_BAD_STATE が返ります。
- PlFw::startCapture()で画像出力を開始した後、本メソッドが呼ばれていない場合は、受信した新しいフレームから破棄されます。
- 引数 block の指定によって、下記のように受信待ち動作が変わります。

  【引数 block が true の場合】
  - 出力画像またはステータス通知が受信されるまで本メソッド内で受信待ち状態となります。
  - 受信待ち状態で PlFw::shutdownPl()が呼ばれた場合は、受信待ち状態を解除し、CANCELED を返します。
  - 外部トリガ種別が Standalone (EXT_TRG_STANDALONE)または Primary (Master, EXT_TRG_MASTER)である時、受信待ち状態で約 1 秒以内に画像が受信できない場合は内部で受信停止と開始を 5 回リトライします。リトライ後も画像が受信できない場合は受信待ちがタイムアウトし、ERR_TIMEOUT が返ります。
  - 外部トリガ種別が Secondary (Slave, EXT_TRG_SLAVE)である時、受信待ち状態で約 1 秒以内に画像を受信できない場合は、Camera::capture()から ERR_TIMEOUT が返ります。連続で ERR_TIMEOUT が返ってくる回数を内部でカウントし、最大カウント数に達した場合、ERR_TIMEOUT が返ります。最大カウント数は 86400 回（約 24 時間相当）です。受信待ち状態が解除された場合、カウント数をリセットします。
  - PlFw::startCapture()で画像出力が開始されていない状態でも受信待ち状態となります。この時は受信待ちのタイムアウトは発生しません。

  【引数 block が false の場合】
  - 出力画像またはステータス通知がない状態では本メソッド内で受信待ち状態にはならず、ERR_NOT_EXIST が返ります。

引数

型	名前	in/out	説明
uint16_t&	proc_id	out	処理 ID
FrameData**	frame	out	受信フレーム
bool	block	in	受信待ちフラグ true ：受信待ちする false ：受信待ちしない

戻り値

SUCCESS	0	成功
CANCELED	1	待ち状態が解除された
REACH_EOF	2	ファイル終端に到達（PLAYBACK のみ）
ERR_INVALID_PTR	3	不正ポインタ（引数 frame）
ERR_BAD_STATE	6	状態遷移異常

	ERR_NOT_EXIST	7	出力画像が存在しない
	ERR_TIMEOUT	8	受信待ち中にタイムアウト発生
	ERR_SYSTEM	12	システム異常
同期/非同期	同期型		

3-3-6-3-12. PlFw::releaseBuf

Table 53. PlFw::releaseBuf メソッド

機能	出力画像の解放			
書式	Result releaseBuf (FrameData** frame)			
説明	・ PlFw::getEvent()で取得した出力画像を解放します。 ・ 引数 frame は解放後、nullptr になります。 ・ PlFw::wakeupPl()が実行されていない状態で本メソッドが呼ばれた場合は ERR_BAD_STATE が返ります。 ・ PlFw::wakeupPl()の引数 closed_pl が true を指定されている場合、本メソッドは ERR_BAD_STATE が返ります。			
引数	**型**	**名前**	**in/out**	**説明**
	FrameData**	frame	in,out	受信フレーム
戻り値	SUCCESS	0	成功	
	ERR_INVALID_PTR	3	不正引数	
	ERR_BAD_STATE	6	状態遷移異常	
	ERR_SYSTEM	12	システム異常	
同期/非同期	同期型			

3-3-6-3-13. PlFw::notifyEvent

Table 54. PlFw::notifyEvent メソッド

機能	Pipeline 処理内の処理スレッドへのイベント通知
書式	Result notifyEvent (uint16_t proc_id, uint8_t event, void* param = nullptr)
説明	・ PlFw::addPlProc()で追加された処理スレッドに対してイベント通知を行います。 ・ 引数 proc_id には PlFw::addPlProc()に返る処理 ID を指定してください。 ・ 引数 event, param は PlFw::addPlProc()で追加した処理スレッド毎の独自イベントになります。内容については各処理スレッドクラスの章を参照してください。EvtThread クラスを継承して独自に用意されたユーザスレッドについては該当するユーザスレッドでイベント定義を行ってください。 ・ 引数 param の内容および領域は本メソッドの処理完了まで保持する必要があります。 ・ PlFw::wakeupPl()が実行されていない状態で本メソッドが呼ばれた場合は ERR_BAD_STATE が返ります。 ・ 通知先の処理スレッドの処理遅延などにより、イベント通知後、25 秒間応答がない場合は、ERR_TIMEOUT が返ります。

	型	名前	in/out	説明
引数	uint16_t	proc_id	in	処理 ID
	uint8_t	event	in	イベント ID
	void*	param	in,out	通知パラメータ
戻り値	SUCCESS	0	成功	
	ERR_BAD_ARG	5	処理 ID、イベント ID が不正	
	ERR_BAD_STATE	6	状態遷移異常	
	ERR_TIMEOUT	8	応答待ち中にタイムアウト発生	
	ERR_SYSTEM	12	システム異常	
同期/非同期	同期型			

3-3-6-4. PIFw クラス用定義

PIFw.h に記載している PIFw クラスで使用する定義を以下に示します。なお、PipelineFramework 内部のみが使用する定義については省略しています。

3-3-6-4-1. 定数定義

3-3-6-4-1-1.定数定義一覧

Table 55. 定数定義一覧

名称	説明
PROC_PIPELINE	画像受信を示す処理 ID

3-3-6-4-1-2.定数定義詳細

3-3-6-4-1-2-1.PROC_PIPELINE

Table 56. PROC_PIPELINE 定義

定義	static const uint16_t PROC_PIPELINE = 0;
説明	・PIFw::getEvent()において、画像受信を示す処理 ID
参照	3-3-6-3-11. PIFw::getEvent()

3-3-7. FrameDataクラス

3-3-7-1. 概要

Table 57. FrameData クラス概要

クラス名	FrameData
所属名前空間	krm
説明	PIFw::getEvent()及び EvtThread::recv()で受信する画像フレームを管理する C++クラス

3-3-7-2. メソッド一覧

Table 58. FrameData クラスメソッド

メソッド名前	説明
FrameData::getFrame	画像フレーム参照
FrameData::getFrameExt	フレーム付加情報取得
FrameData::getUserBuf	追加バッファ参照

3-3-7-3. メソッド詳細

3-3-7-3-1. FrameData::getFrame

Table 59. FrameData::getFrame メソッド

機能	画像フレーム参照			
書式	Frame* getFrame (void)			
説明	・ 画像フレームを参照します。			
引数	型	名前	in/out	説明
	なし			
戻り値	Frame*	画像フレーム		
同期/非同期	同期型			

3-3-7-3-2. FrameData::getFrameExt

Table 60. FrameData::getFrameExt メソッド

機能	フレーム付加情報取得			
書式	FrameExtInfo* getFrameExt(void)			
説明	・ フレーム付加情報として、受信フレームレート（実測値）と PlayBack 時の現在再生位置情報を参照します。 ・ FrameExtInfo.play_time は PlFw::PlFw()で PLAYBACK を指定していた場合のみ有効となります。			
引数	型	名前	in/out	説明
	なし			
戻り値	FrameExtInfo*	フレーム付加情報		
同期/非同期	同期型			

3-3-7-3-3. FrameData::getUserBuf

Table 61. FrameData::getUserBuf メソッド

機能	追加バッファ参照			
書式	void* getUserBuf (uint16_t proc_id)			
説明	・ PlFw::addPlProc()の時に追加したバッファを参照します。 ・ 引数 proc_id には PlFw::addPlProc()に返る処理 ID を指定してください。 ・ PlFw::addPlProc()の時にバッファを追加しなかった場合は nullptr が返ります。 ・ 引数 proc_id が不正の場合は nullptr が返ります。 ・ 戻り値を任意の型にキャストして使用してください。			
引数	型	名前	in/out	説明

	uint16_t	proc_id	in	処理 ID
戻り値	void*	追加バッファの先頭ポインタ		
同期/非同期	同期型			

3-3-7-4. FrameData クラス用定義

FrameData.h に記載している FrameData クラスで使用する定義を以下に示します。
なお、PipelineFramework 内部のみが使用する定義については省略しています。

3-3-7-4-1. 構造体定義

3-3-7-4-1-1.構造体定義一覧

Table 62. 構造体定義一覧

名称	説明
FrameExtInfo	PipelineFramework 内フレーム付加情報

3-3-7-4-1-2.構造体定義詳細

3-3-7-4-1-2-1.FrameExtInfo

Table 63. FrameData::FrameExtInfo 定義

定義	struct FrameExtInfo {
	uint16_t rcv_fps;
	PlayBack::PlayTime play_time;
	};		

説明	・ PipelineFramework 内で算出もしくは取得したフレーム付加情報を示します。		
引数	型	名前	説明
	uint16_t	rcv_fps	受信フレームレート [fps × 100]
	PlayBack::PlayTime	play_time	現在再生位置情報（PlayBack 動作時のみ）
参照	3-3-7-3-2. FrameData::getFrameExt		

3-3-8. EvtThreadクラス

3-3-8-1. 概要

Table 64. EvtThread クラス概要

クラス名	EvtThread
所属名前空間	krm
説明	Pipeline 処理の基底クラス Pipeline 処理として追加する処理スレッドは本クラスを継承して用意してください。

3-3-8-2. メソッド一覧

Table 65. EvtThread クラスメソッド

メソッド名	説明
EvtThread::EvtThread	コンストラクタ
EvtThread::~EvtThread	デストラクタ
EvtThread::getKind	処理種別の取得
EvtThread::enableFrameDrop	処理停滞時のフレームデータの廃棄許可
EvtThread::recvChgProp	カメラデバイス情報変化通知受信
EvtThread::recvChgFmt	画像フォーマット変化通知受信
EvtThread::recvImage	画像フレーム通知受信
EvtThread::recvReset	リセット通知受信
EvtThread::recvUserEvent	イベント受信

3-3-8-3. メソッド詳細

3-3-8-3-1. EvtThread::EvtThread

Table 66. EvtThread::EvtThread メソッド

機能	コンストラクタ			
書式	EvtThread (void)			
説明	・　処理スレッドクラスを初期化します。			
引数	型	名前	in/out	説明
	なし			
戻り値	なし			
同期/非同期	同期型			

3-3-8-3-2. EvtThread::~EvtThread

Table 67. EvtThread::~EvtThread メソッド

機能	デストラクタ			
書式	~ EvtThread (void)			
説明	・　処理スレッドクラスを破棄します。			
引数	型	名前	in/out	説明
	なし			
戻り値	なし			
同期/非同期	同期型			

3-3-8-3-3. EvtThread::getKind

Table 68. EvtThread::getKind メソッド

機能	処理種別の取得			
書式	ProcKind getKind (void)			
説明	・　処理種別を返します。 ・　EvtThread クラスを継承する場合は、本メソッドは override せずに使用してください。			
引数	型	名前	in/out	説明

	なし			
戻り値	ProcKind	処理種別		
同期/非同期	同期型			

3-3-8-3-4. EvtThread::enableFrameDrop

Table 69. EvtThread::enableFrameDrop メソッド

機能	処理停滞時のフレームデータの破棄許可			
書式	bool enableFrameDrop (void)			
説明	・ 画像受信時に Pipeline 処理が停滞している場合に、画像フレームを破棄するかを返します。 ・ EvtThread クラス継承時に画像フレームの破棄を許可しない場合は override して、false を返すようにしてください。ただし、画像フレームの破棄を許可しない場合、カメラデバイスからの画像受信に必要な画像フレームバッファを消費しきった場合、カメラデバイスからの画像受信が停止されます。 ・ 画像フレームの破棄を許可する場合、スレッド内の処理停滞時には EvtThread::getMaxQueuingFrames()が返す最大蓄積フレーム数を超えた場合、画像フレームは破棄されます。画像フレームを破棄した場合、次の画像フレームではフレーム不連続のフラグが立ちます。（FrameInfo.frm_err.drop =1 となる） ・ 本メソッドは PlFw::addPlProc()内で参照されます。			
引数	型	名前	in/out	説明
	なし			
戻り値	true	処理が停滞している場合に、画像フレーム破棄を許可する		
	false	処理が停滞している場合に、画像フレーム破棄を禁止する		
同期/非同期	同期型			

3-3-8-3-5. EvtThread::getMaxQueuingFrames

Table 70. EvtThread::getMaxQueuingFrames メソッド

機能	最大蓄積フレーム数取得			
書式	uint8_t getMaxQueuingFrames (void)			
説明	・ 画像受信時に Pipeline 処理が停滞している場合に、画像フレームの蓄積をする最大フレーム数を取得します。 ・ EvtThread::enableFrameDrop()が true を返している場合に、本メソッドが返す数値を超えた場合、溢れた画像フレームは破棄されます。 ・ 初期値は 3 フレームとなっています。初期値以外にしたい場合は EvtThread クラス継承時に override して、蓄積する最大フレーム数を返すようにしてください。 ・ 本メソッドは PlFw::addPlProc()内で参照されます。			
引数	型	名前	in/out	説明
	なし			
戻り値	1～255	最大蓄積フレーム数		
	0	初期値（3）を最大蓄積フレーム数とする		
同期/非同期	同期型			

3-3-8-3-6. EvtThread::recvChgProp

Table 71. EvtThread::recvChgProp メソッド

機能	カメラデバイス情報の受信			
書式	Result recvChgProp (const CameraProperty& property)			
説明	・ PlFw::startCapture()が実行されたときに呼びだされます。 ・ EvtThread クラスを継承時にカメラデバイス情報（測距レンジ、FPS など）を使用する場合は本メソッドを override してください。 ・ 本メソッドで SUCCESS 以外を返した場合、PlFw::startCapture()の戻り値として反映されます。 ・ 本メソッドの引数は本メソッド実行中のみ有効な変数です。ポインタ等へは登録しないでください。			
引数	型	名前	in/out	説明
	const CameraProperty&	property	in	カメラデバイス情報
戻り値	SUCCESS	0	成功	
	その他	–	継承したクラスで規定	
同期/非同期	同期型			

3-3-8-3-7. EvtThread::recvChgFmt

Table 72. EvtThread::recvChgFmt メソッド

機能	画像フォーマット情報の受信			
書式	Result recvChgFmt (const ImageFormats& formats)			
説明	・ PlFw::startCapture()が実行されたときに呼びだされます。 ・ EvtThread クラスを継承時に画像データのフォーマット情報を使用する場合は本メソッドを override してください。 ・ 本メソッドで SUCCESS 以外を返した場合、PlFw::startCapture()に戻り値して反映されます。 ・ 本メソッドの引数は本メソッド実行中のみ有効な変数です。ポインタ等へは登録しないでください。			
引数	型	名前	in/out	説明
	const ImageFormats&	formats	in	画像フォーマット
戻り値	SUCCESS	0	成功	
	その他	–	継承したクラスで規定	
同期/非同期	同期型			

3-3-8-3-8. EvtThread::recvImage

Table 73. EvtThread::recvImage メソッド

機能	画像フレーム通知受信
書式	Result recvImage (std::shared_ptr<FrameData>& frame)
説明	・ PlFw::startCapture()実行後、画像フレームが受信されたときに呼びだされます。

	・ EvtThread クラスを継承時に画像フレームを使用する場合は本メソッドを override してください。 ・ 画像フレーム内のデータを上書きすると、次の Pipeline 処理には上書きされた状態で通知されます。 ・ 引数 frame は所有権がある状態で渡されます。本メソッド内で所有権の解放は不要です。 ・ 本メソッドは 1 フレームの間隔以内に処理を完了させるようにしてください。 ・ 本メソッドで SUCCESS 以外を返した場合、PlFw::getEvent()もしくは PlFw::wakeupPl()で登録した callback 関数に戻り値が返されます。 ・ 本メソッドの引数は本メソッド実行中のみ有効な変数です。ポインタ等へは登録しないでください。			
引数	型	名前	in/out	説明
	std::shared_ptr<FrameData>&	frame	in,out	画像フレーム
戻り値	SUCCESS	0	成功	
	その他	–	継承したクラスで規定	
同期/非同期	同期型			

3-3-8-3-9. EvtThread::recvReset

Table 74. EvtThread::recvReset メソッド

機能	リセット通知受信			
書式	void recvReset (void)			
説明	・ PlFw::stopCapture()が実行されたときに呼びだされます。 ・ EvtThread クラスを継承時に停止処理での初期化処理などが必要な場合は本メソッドを override してください。			
引数	型	名前	in/out	説明
	なし			
戻り値	なし			
同期/非同期	同期型			

3-3-8-3-10. EvtThread::recvUserEvent

Table 75. EvtThread::recvUserEvent メソッド

機能	イベント受信			
書式	Result recvUserEvent (uint8_t event, void* param)			
説明	・ PlFw::notifyEvent()によるイベント通知が行われた際に呼び出されます。 ・ EvtThread クラスを継承時に独自イベント処理が必要な場合は本メソッドを override してください。 ・ 引数 event および param の内容については継承したクラスで規定してください。 ・ 本メソッドで返した戻り値が PlFw::notifyEvent()の戻り値として返されます。 ・ 引数 param の内容を変更した場合は、PlFw::notifyEvent()を呼び出した側の引数 param も更新されます。 ・ 本メソッドの引数は本メソッド実行中のみ有効な変数です。ポインタ等へは登録しないでください。			
引数	型	名前	in/out	説明

	uint8_t	event	in	イベント ID
	void*	param	in,out	通知パラメータ
戻り値	SUCCESS	0	成功	
	その他	–	継承したクラスで規定	
同期/非同期	同期型			

3-3-8-4. クラス内変数一覧

EvtThread クラスを継承したクラスで参照できる変数を以下に示します。

Table 76. EvtThread クラス変数

変数型	変数名	説明
uint16_t	proc_id_	処理 ID PlFw::addPlProc()で Pipeline 処理に追加した際の ID が格納されます。 FrameData::getUserBuf()で追加バッファにアクセスする際に使用してください。

3-3-8-5. EvtThread クラス用定義

EvtThread.h に記載している EvtThread クラスで使用する定義を以下に示します。
なお、PipelineFramework 内部のみが使用する定義については省略しています。

3-3-8-5-1. 列挙体定義

3-3-8-5-1-1.列挙体定義一覧

Table 77. 列挙体定義一覧

名称	説明
ProcKind	処理種別

3-3-8-5-1-2.列挙体定義詳細

3-3-8-5-1-2-1.ProcKind

Table 78. ProcKind 定義

定義	enum ProcKind : uint16_t {
	PROC_CAMERA = 0,
	PROC_RECORD,
	PROC_LENSCONV,
	PROC_POSTFILT,
	PROC_USER
	};		

説明	・ Pipeline 処理に追加された処理種別を示します。		
値	名前	値	説明
	PROC_CAMERA	0	Camera 受信処理
	PROC_RECORD	1	ファイル保存処理
	PROC_LENSCONV	2	Lens 系変換処理
	PROC_POSTFILT	3	PostFilter 処理
	PROC_USER	4	User 追加処理

参照	3-3-8-3-3. EvtThread::getKind

3-3-8-5-2. 構造体定義

3-3-8-5-2-1.構造体定義一覧

Table 79. 構造体定義一覧

名称	説明
CameraProperty	カメラ固有情報

3-3-8-5-2-2.構造体定義詳細

3-3-8-5-2-2-1.CameraProperty

Table 80. CameraProperty 定義

定義	struct CameraPropety { ModeInfo mode_info; LensInfo lens_info; PostFiltInfo post_filt_info; CamFov fov; } CameraProperty;		
説明	・ カメラ固有情報（動作モード情報、Lens 系変換処理パラメータ、PostFilter 処理情報、FOV）を示します。		
引数	型	名前	説明
	ModeInfo	mode_info	動作モード情報
	LensInfo	lens_info	Lens 系変換処理パラメータ
	PostFiltInfo	post_filt_info	PostFilter 処理情報
	CamFov	fov	カメラ視野角
参照	3-3-8-3-6. EvtThread::recvChgProp		

3-3-8-6. 型定義

3-3-8-6-1. 型定義一覧

Table 81. 型定義一覧

名称	説明
PlCbFunc	状態変化通知用のコールバック関数

3-3-8-6-2. 型定義詳細

3-3-8-6-2-1.PlCbFunc

Table 82. PlCbFunc 定義

定義	using PlCbFunc = std::function<void(uint16_t, Result)>;	
説明	・ 状態変化通知用のコールバック関数を示します。 ・ 引数には PlFw::addPlProc()で追加した処理スレッドの処理 ID、処理 ID が示す該当処理スレッドからのステータス通知が値として返されます。通知されるステータス通知については各処理スレッドの章を参照してください。	
引数	型	説明
	uint16_t	処理 ID
	Result	ステータス通知値

戻り値	なし
参照	3-3-6-3-5. PlFw::wakeupPl

3-3-9. RecordThreadクラス

3-3-9-1. 概要

Table 83. RecordThread クラス概要

クラス名	RecordThread
所属名前空間	krm
説明	Record クラスを使用したファイル保存処理を行うスレッド EvtThread を継承しているメソッドについては、EvtThread クラスの内容を参照してください。
追加バッファ	未使用

3-3-9-2. イベント一覧

PlFw::notifyEvent()で使用できるイベントを以下に示します。

Table 84. RecordThread クラスイベント

イベント名	説明
EV_REC_START	ファイル保存開始
EV_REC_STOP	ファイル保存停止

3-3-9-2-1. EV_REC_START

Table 85. EV_REC_START 概要

情報種別	ファイル保存開始
引数型	RecordParam
説明	・ ファイル保存を開始します。 ・ 戻り値として、Record::openRec()の戻り値が返ります。

3-3-9-2-2. EV_REC_STOP

Table 86. EV_REC_STOP 概要

情報種別	ファイル保存停止
引数型	なし
説明	・ ファイル保存を停止します。 ・ 戻り値として、Record::closeRec()の戻り値が返ります。

3-3-9-3. ステータス通知

RecordThread での処理中にステータス通知が発生した場合、PlFw::getEvent()もしくは PlFw::addPlProc()のコールバック関数に以下の値が通知されます。

Table 87. RecordThread クラスステータス通知

通知される値	説明
REACH_EOF	指定フレーム数分の保存完了
ERR_BAD_STATE	状態遷移異常による保存失敗
ERR_EMPTY	バッファなどが空になっていることによる保存失敗
ERR_FULL	容量などが足りないことによる保存失敗
ERR_SYSTEM	システム異常（メモリ確保失敗など）による保存失敗

3-3-9-4. RecordThread クラス用定義

RecordThreadEvent.h に記載している RecordThread クラスに対するイベント通知として使用する定義を以下に示します。

3-3-9-4-1. 列挙体定義

3-3-9-4-1-1.列挙体定義一覧

Table 88. 列挙体定義一覧

名称	説明
RecEvent	RecordThread クラスに対するイベント通知のイベント ID

3-3-9-4-1-2.列挙体定義詳細

3-3-9-4-1-2-1.RecEvent

Table 89. RecEvent 定義

定義	enum RecEvent : uint8_t {
	EV_REC_START,
	EV_REC_STOP,
	};		

説明	・ RecordThread クラスに対するイベント通知のイベント ID を示します。		
値	名前	値	説明
	EV_REC_START	0	ファイル保存開始
	EV_REC_STOP	1	ファイル保存停止
参照	3-3-6-3-13. PlFw::notifyEvent(), 3-3-8-3-10. EvtThread::recvUserEvent(), 3-3-9-2-1. EV_REC_START, 3-3-9-2-2. EV_REC_STOP		

3-3-9-4-2. 構造体定義

3-3-9-4-2-1.構造体定義一覧

Table 90. 構造体定義一覧

名称	説明
RecordParam	保存対象の設定情報

3-3-9-4-2-2.構造体定義詳細

3-3-9-4-2-2-1.RecordParam

Table 91. RecordParam 定義

定義	struct RecordParam {
	std::filesystem::path path;
	uint32_t save_frames;
	uint16_t packing_frames;
	bool is_filt_med;
	bool is_filt_bil;
	bool is_filt_fly_p;
	bool is_crct_dist;
	};		

説明	・ 保存対象の設定情報を示します。		
引数	型	名前	説明
	std::filesystem::path	path	保存先ディレクトリ
	uint32_t	save_frames	保存フレーム数
	uint16_t	packing_frames	１ファイル内に含めるフレーム数
	bool	is_filt_med	メディアンフィルタを適用済みか
	bool	is_filt_bil	バイラテラルフィルタを適用済みか
	bool	is_filt_fly_p	フライングピクセルフィルタを適用済みか
	bool	is_crct_dist	歪曲補正済みかどうか
参照	3-3-9-2-1. EV_REC_START		

3-3-10. LensConvThreadクラス

3-3-10-1. 概要

Table 92. LensConvThread クラス概要

クラス名	LensConvThread
所属名前空間	krm
説明	Lens 系変換処理を行うスレッド EvtThread を継承しているメソッドについては、EvtThread クラスの内容を参照してください。
追加バッファ	未使用

3-3-10-2. メソッド一覧

Table 93. LensConvThread メソッド一覧

メソッド名	説明
LensConvThread::LensConvThread	コンストラクタ

3-3-10-3. メソッド詳細

3-3-10-3-1. LensConvThread::LensConvThread

Table 94. LensConvThread::LensConvThread メソッド

機能	コンストラクタ			
書式	LensConvThread (bool enable_pcd = true, bool enable_distortion = true,)			
説明	・ Lens 系変換処理を行うスレッドの初期化を行います。 ・ 引数 enable_pcd を有効に設定した場合、FrameData::getFrame()で出力される画像フレームに点群データ（pcd）が出力されるようになります。点群の出力については PlFw::notifyEvent()での切り替えはできません。 ・ 引数 enable_distortion を有効にした場合、FrameData::getFrame()で出力される画像フレーム内の Depth 画像、IR 画像に歪曲補正がかかった画像が出力されるようになります。また、PlFw::notifyEvent(EV_LENS_DIST)でも切り替えをすることができます。 ・ 歪曲補正を無効に設定した場合であっても、点群変換に使用する Depth 画像に対しては歪曲補正が実施されます。 ・ 入力 Depth 画像が歪曲補正済みのデータの場合、enable_distortion の設定にかかわらず歪曲補正は無効になります			
引数	**型**	**名前**	**in/out**	**説明**
	bool	enable_pcd	in	点群変換の有効化 true ：有効（初期値） false ：無効
	bool	enable_distortion	in	歪曲補正の有効化 true ：有効（初期値） false ：無効
戻り値	なし			
同期/非同期	同期型			

3-3-10-4. イベント一覧

PlFw::notifyEvent()で使用できるイベントを以下に示します。

Table 95. LensConvThread クラスイベント

イベント名	説明
EV_LENS_PSBL_DIST	歪曲補正が可能かどうかの情報通知
EV_LENS_DIST	歪曲補正の ON/OFF
EV_LENS_PCD_KIND	点群変換の座標系切り替え
EV_LENS_PCD_ORG_POS	世界座標変換の原点位置、回転情報通知
EV_LENS_PCD_COLOR	点群変換後の色情報に対する設定

3-3-10-4-1. EV_LENS_PSBL_DIST

Table 96. EV_LENS_PSBL_DIST 概要

情報種別	歪曲補正が可能かどうかの情報取得
引数型	bool
説明	・ 歪曲補正が可能かどうかの情報を取得します。 ・ カメラデバイス内で歪曲補正を実施している場合は false となります。

3-3-10-4-2. EV_LENS_DIST

Table 97. EV_LENS_DIST 概要

情報種別	歪曲補正の ON/OFF
引数型	bool
説明	・ 引数が true の場合、歪曲補正を ON します。false の場合、歪曲補正を OFF します。 ・ 入力画像が歪曲補正変換済みのデータの場合、enable_distortion の設定にかかわらず歪曲補正は無効になります。 ・ 初期状態は LensConvThread::LensConvThread()の引数 enable_ distortion の値になります。

3-3-10-4-3. EV_LENS_PCD_KIND

Table 98. EV_LENS_PCD_KIND 概要

情報種別	点群変換の座標系切り替え
引数型	bool
説明	・ 引数が true の場合、世界座標系の点群変換を行います。false の場合、カメラ座標系の点群変換を行います。 ・ 初期状態はカメラ座標系の点群変換の状態になります。

3-3-10-4-4. EV_LENS_PCD_ORG_POS

Table 99. EV_LENS_PCD_ORG_POS 概要

情報種別	世界座標変換の原点位置、回転情報通知
引数型	PosOrgRotation
説明	・ 世界座標系の点群変換時の原点位置、回転情報を通知します。 ・ 初期状態は原点位置が x = 0, y = 0, z = 0、回転情報が pitch = 0, roll = 0, yaw = 0 の状態になります。

3-3-10-4-5. EV_LENS_PCD_COLOR

Table 100. EV_LENS_PCD_COLOR 概要

情報種別	点群変換後の色情報に対する設定
引数型	PcdColorKind
説明	・ 点群変換後の色情報に対する設定を行います。 ・ PCD_COLOR_NONE を指定した場合、色情報には設定を行いません。 ・ PCD_COLOR_IR を指定した場合、無効点以外の点群データの色情報に、IR を入れた点群データに置き換えます（PcdData.kind= PCD_IRXYZ）。 ・ 初期状態は PCD_COLOR_NONE になります。

3-3-10-5. ステータス通知

LensConvThread での処理中にステータス通知が発生した場合、PlFw::getEvent() もしくは PlFw::addPlProc()のコールバック関数に以下の値が通知されます。

Table 101. LensConvThread クラスステータス通知

通知される値	説明
ERR_INVALID_PTR	通知パラメータの不正なポインタによる変換処理失敗

ERR_BAD_ARG	画像フォーマットが不正による変換処理失敗
ERR_BAD_STATE	状態遷移異常による変換処理失敗
ERR_SYSTEM	システム異常による変換処理失敗

3-3-10-6. LensConvThread クラス用定義

LensConvThreadEvent.h に記載している LensConvThread クラスに対するイベント通知として使用する定義を以下に示します。

3-3-10-6-1. 列挙体定義

3-3-10-6-1-1.列挙体定義一覧

Table 102. 列挙体定義一覧

名称	説明
LensConvEvent	LensConvThread クラスに対するイベント通知のイベント ID
PcdColorKind	点群変換後の色情報設定種別

3-3-10-6-1-2.列挙体定義詳細

3-3-10-6-1-2-1.LensConvEvent

Table 103. LensConvEvent 定義

定義	enum LensConvEvent: uint8_t {
	EV_LENS_PSBL_DIST,
	EV_LENS_DIST,
	EV_LENS_PCD_KIND,
	EV_LENS_PCD_ORG_POS,
	EV_LENS_PCD_COLOR
	};		

説明	・ LensConvThread クラスに対するイベント通知のイベント ID を示します。		
値	名前	値	説明
	EV_LENS_PSBL_DIST	0	歪曲補正が可能かどうかの情報通知
	EV_LENS_DIST	1	歪曲補正の ON/OFF
	EV_LENS_PCD_KIND	2	点群変換の座標系切り替え
	EV_LENS_PCD_ORG_POS	3	世界座標変換の原点位置、回転情報通知
	EV_LENS_PCD_COLOR	4	点群変換後の色情報に対する設定
参照	3-3-6-3-13. PlFw::notifyEvent()		

3-3-10-6-1-2-2.PcdColorKind

Table 104. PcdColorKind 定義

定義	enum PcdColorKind : uint8_t {
	PCD_COLOR_NONE,
	PCD_COLOR_IR
	};		

説明	・ 点群変換後の色情報設定種別を示します。		
値	名前	値	説明
	PCD_COLOR_NONE	0	色情報を設定しない

	PCD_COLOR_IR	1	IR を色情報に入れる
参照	3-3-6-3-13. PlFw::notifyEvent(), 3-3-10-4-5. EV_LENS_PCD_COLOR		

3-4. Camera クラス

3-4-1. 概要

Table 105. Camera クラス概要

提供ヘッダファイル	Camera.h	Camera クラス定義
	CameraType.h	Camera クラス共通型定義
	PlayBackType.h	PlayBack 機能専用型定義
所属名前空間	krm	
説明	カメラデバイスの制御、 または保存されたファイルの再生を行う C++クラス	
スレッドセーフ	Camera クラスはスレッドセーフとなっています。 利用するユーザプログラム側で排他処理は必要ありません。	

3-4-2. 提供機能

Camera クラスが提供する機能概要を以下に示します。

3-4-2-1. デバイス制御機能

カメラデバイスに対する制御として提供する機能概要を以下に示します。

Table 106. Camera クラス デバイス制御機能

提供機能	機能概要
デバイスパラメータ取得機能	カメラデバイスが保持している以下のパラメータ情報を取得します。 ・ カメラデバイス情報（名称、シリアル番号、調整番号） ・ デバイス内のファームウェアバージョン情報 ・ カメラデバイスの FOV 情報 ・ 外部トリガ種別、信号オフセット情報 ・ カメラデバイスの PostFilter 情報（PostFilter 処理用） ・ カメラデバイスの Lens 情報（Lens 系変換処理用） ・ 使用可能動作モード情報 ・ 動作モード毎の測距範囲、FPS、受信可能画像種別情報 ・ 出力画像のフォーマット情報 ・ 発光回数情報 ・ AE 機能の状態、制御間隔情報 ・ RAW 飽和閾値、IR 無効閾値情報 ・ 干渉防止機能情報 ・ カメラデバイス内のレジスタ情報 ・ 操作モード情報

	・ カメラキャリブレーション操作モード画像情報 ・ USB ペリフェラルコントローラアクセスキー
デバイスパラメータ設定機能	カメラデバイスの以下のパラメータ情報を設定します。 ・ 外部トリガ種別、信号オフセット情報 ＊ ・ 動作モード切り替え ＊ ・ 出力画像の切り替え ＊ ・ 発光回数設定 ・ AE 機能の状態、制御間隔情報 ＊ ・ RAW 飽和閾値、IR 無効閾値情報 ・ 干渉防止機能情報 ・ カメラデバイス内のレジスタ設定 ＊ ・ レジスタ書き込み状態解除設定 ＊ ・ 操作モード情報 ＊ ・ カメラキャリブレーション操作モード画像情報 ＊ ・ USB ペリフェラルコントローラアクセスキー ・ USB ペリフェラルコントローラアップデートモード設定 ＊これらのパラメータ設定はカメラデバイスから画像出力中は設定できません。
画像受信機能	カメラデバイスから出力される以下の画像を受信します。 ・ Depth 画像 ・ IR 画像 ・ センサ RAW 画像 受信する画像種別は動作モード切り替えおよび出力画像の切り替えによって決まります。 Camera クラスではカメラモジュールから出力される Depth 画像と IR 画像は有効画素のみ、センサ RAW 画像は全ての画素を出力します。また、画像毎にフレーム付加情報を合わせて出力します。

3-4-2-2. ファイル再生機能

ファイル再生用の拡張機能として提供する機能概要を以下に示します。

Table 107. Camera クラス ファイル再生機能

提供機能	機能概要
ファイル再生機能	・ Record クラスによって保存されたファイルを入力として使用し、保存されたファイル内の Depth 画像、IR 画像、センサ RAW 画像およびフレーム付加情報を出力する機能です。 ・ 出力される画像種別は保存されたファイルに依存します。
一時停止機能	・ ファイル再生している状態で一時停止・一時停止解除する機能です。 ・ 一時停止解除時は前の状態に関わらず、再生状態に戻ります。 ・ 一時停止中はフレームの出力は行われません。
倍速・スロー再生機能	・ ファイル再生している状態で 2 倍速・3 倍速・4 倍速の倍速再生および、½倍速・⅓倍速・¼倍速のスロー再生を行う機能です。 ・ 倍速再生は上限 120fps、スロー再生は下限 10fps までになります。
先送り・巻き戻し機能	・ ファイル再生している状態で指定フレーム分、先送り・巻き戻りを行う機能です。 ・ 先送りを行った結果、終端フレームに到達した場合はファイル再生を停止します。 ・ 巻き戻しを行った結果、先頭フレーム以前に戻る場合は先頭から再生されます。

3-4-2-2-1. ファイル再生対応フォーマットバージョン

ファイル再生機能では Record クラスが保存する RecInfo.json フォーマットバージョンが Ver.3.0.1 の場合のみ再生することができます。

このバージョン以前で保存されたファイルは再生することができません。

3-4-3. メソッド一覧

Table 108. Camera クラスメソッド

メソッド名前	説明
Camera	コンストラクタ
~Camera	デストラクタ
getDeviceList	接続しているカメラデバイスリストの取得
openDevice	デバイス Open 処理
closeDevice	デバイス Close 処理
getProperty	デバイスパラメータ取得
setProperty	デバイスパラメータ設定
startCapture	画像出力開始
stopCapture	画像出力停止
capture	出力画像の取得
cancel	出力画像の受信待ち状態の解除

3-4-3-1. 状態遷移

Camera クラスの状態遷移は以下となります。

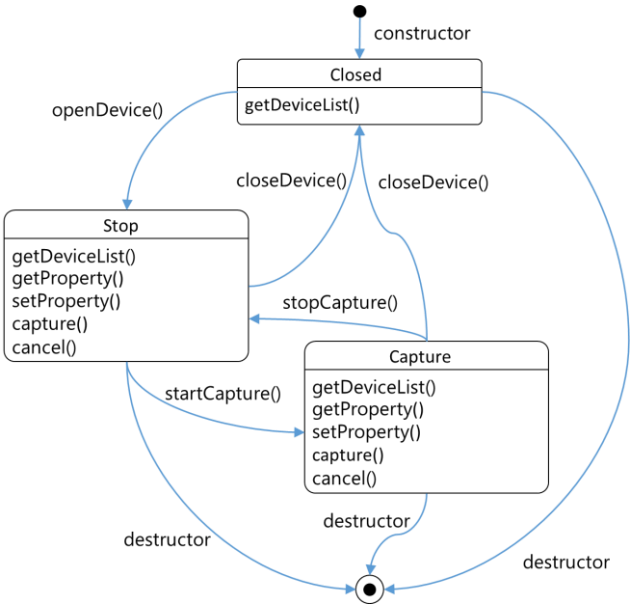


Figure 15. Camera クラス状態遷移

3-4-3-2. 状態遷移（ファイル再生）

Camera クラスのファイル再生において、ファイル再生用のプロパティコマンドを使用した場合、Capture 状態以降で以下の状態遷移となります。Pause・Fast・Slow の各状態から、Stop・Closed・終了状態への遷移は Capture 状態と同様です。

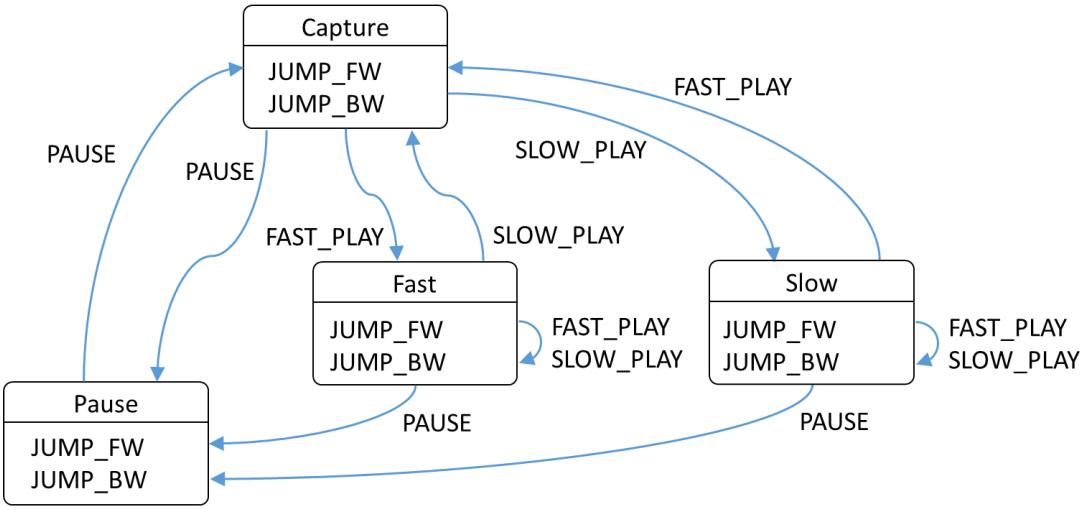


Figure 16. Camera クラス（ファイル再生）状態遷移

PlayBack::CMD_FAST_PLAY、PlayBack::CMD_SLOW_PLAY コマンドにおける、Capture・Fast・Slow の状態遷移（再生速度遷移）は下記のようになります。

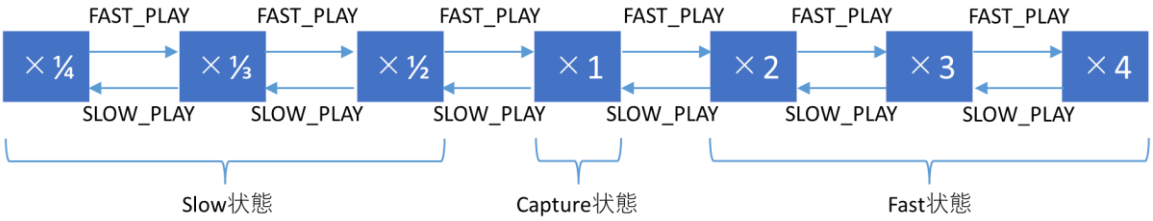


Figure 17. ファイル再生時の再生速度遷移

3-4-4. 制御シーケンス

Camera クラスを用いた基本的なカメラ制御シーケンスを記載します。なお、以下のシーケンスに関しては関数の戻り値判定など異常系の判断は省略しております。

3-4-4-1. 初期化シーケンス

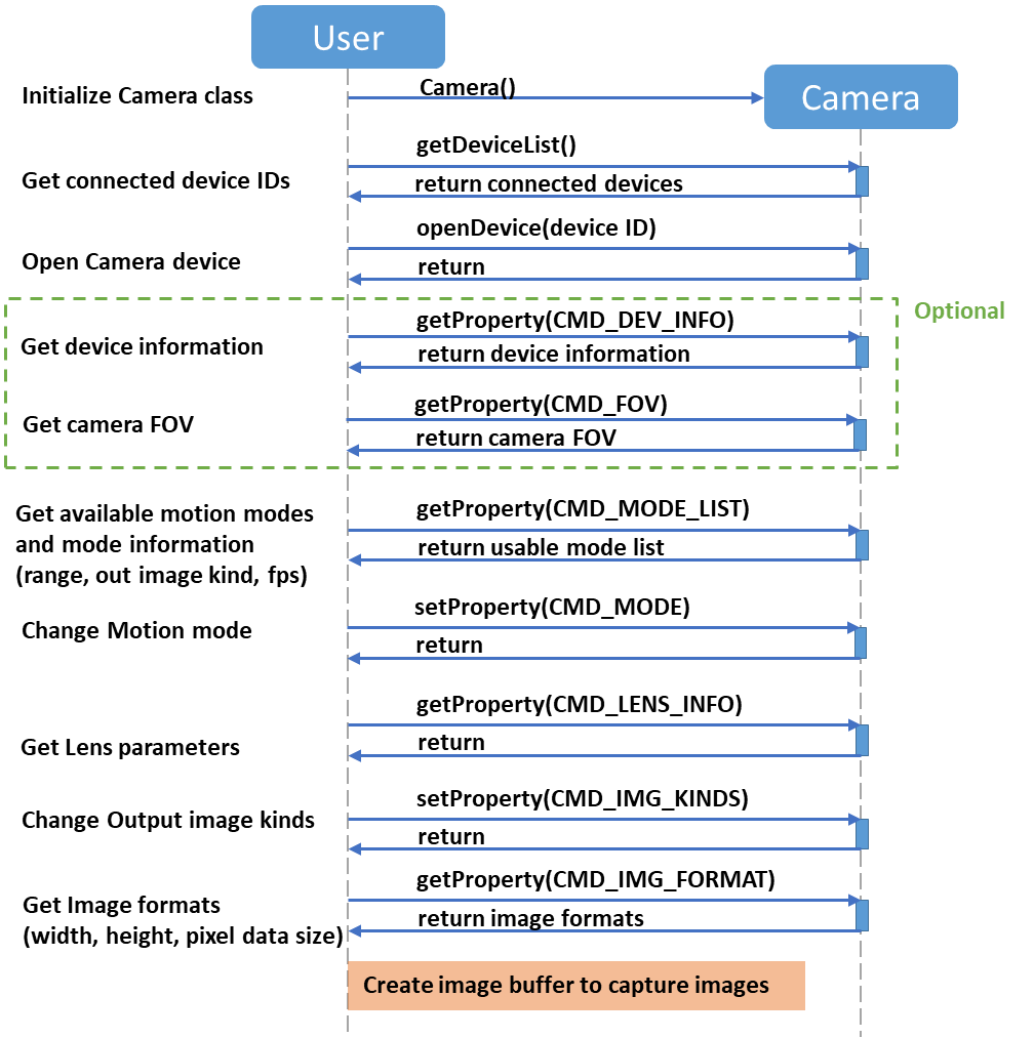


Figure 18. 初期化シーケンス図

3-4-4-2. 画像受信シーケンス

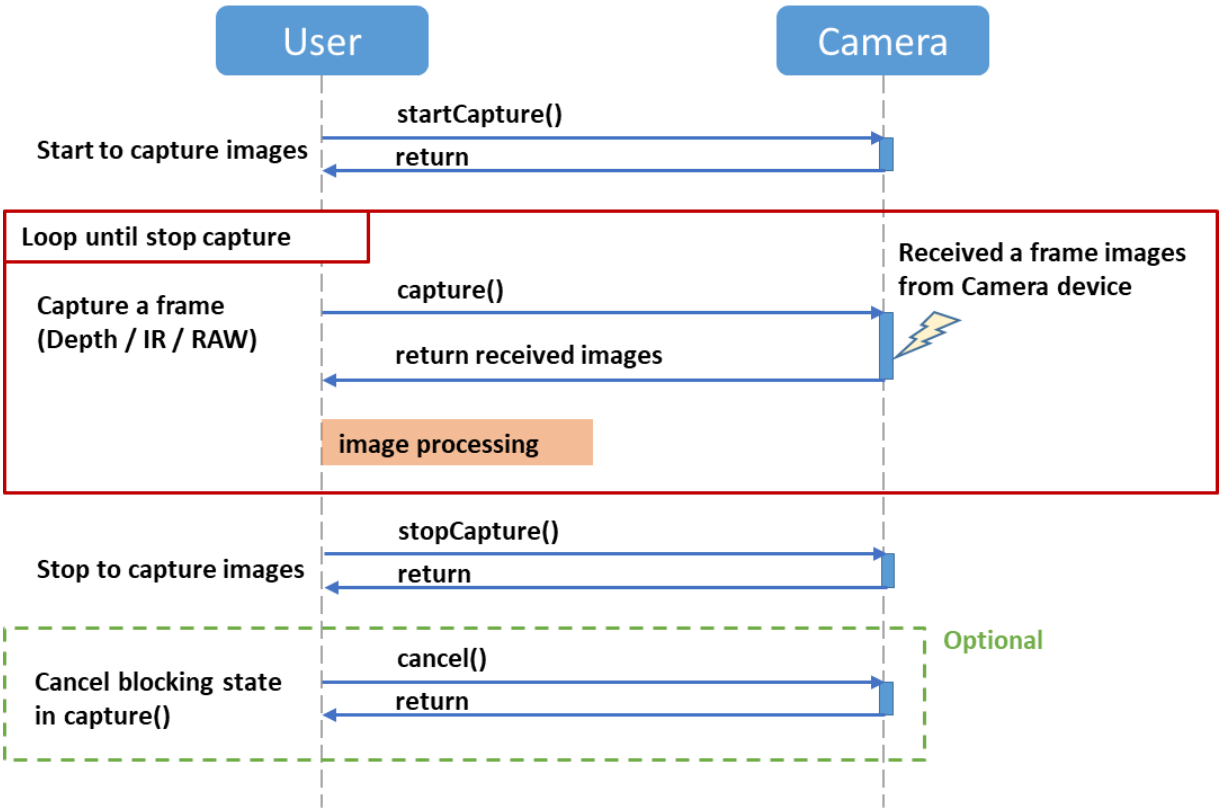


Figure 19. 画像受信シーケンス図

3-4-4-3. 終了シーケンス

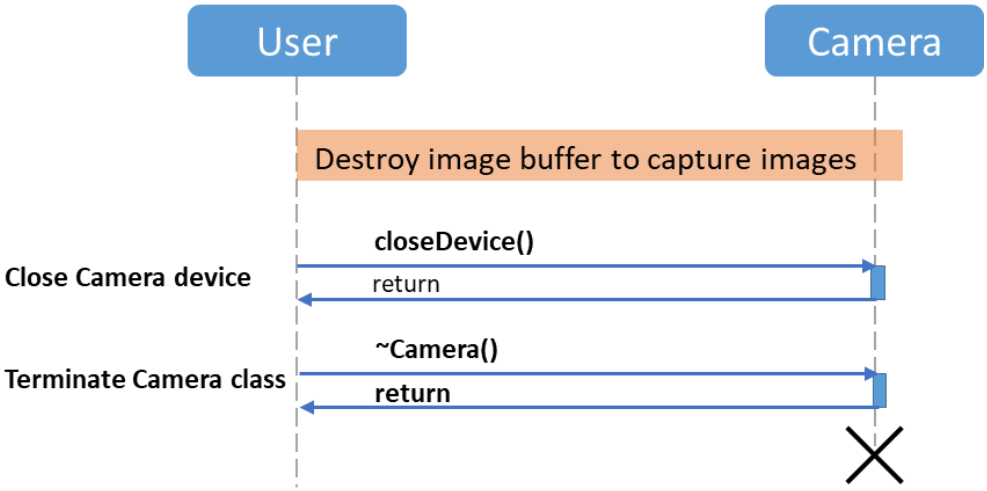


Figure 20. 終了シーケンス図

3-4-5. メソッド詳細

3-4-5-1. Camera

Table 109. Camera::Camera メソッド

機能	コンストラクタ			
書式	Camera (CameraType type, const void* param, Result& res)			
説明	・ Camera 制御の初期化を行います。 ・ 引数 type が規定値外の場合、引数 res に RET_OVER_RANGE が返ります。 ・ 3D ToF Camera カメラ（C11U）を使用する際には引数 type に C11U_USB、引数 param に nullptr または OpInfo＊型を指定してください。nullptr を指定した場合は、通常操作モード（OP_NORMAL）となります。 ・ ファイル再生機能を使用する際には引数 type に PLAYBACK、引数 param に PlayBack::ConfigParam*型を指定してください。再生対象のディレクトリパスは setProperty(PlayBack::CMD_PLAY_TARGET)でも再設定が可能です。			
引数	型	名前	in/out	説明
	CameraType	type	in	使用するカメラ種別
	const void*	param	in	デバイスパラメータ
	Result&	res	out	戻り値
戻り値	SUCCESS	0	成功	
	ERR_INVALID_PTR	3	引数のポインタが不正（PLAYBACK 指定時）	
	ERR_OVER_RANGE	4	カメラ種別が不正	
	ERR_BAD_STATE	6	状態遷移異常	
	ERR_SYSTEM	12	メモリ確保失敗	
同期/非同期	同期型			

3-4-5-2. ~Camera

Table 110. Camera::~Camera メソッド

機能	デストラクタ			
書式	~Camera (void)			
説明	・ Camera 制御の終了処理を行います。 ・ 本処理にて内部で確保したリソース開放を行うため、終了する際は必ずデストラクタが実行されるように設計してください。 ・ openDevice()実行後、closeDevice()が実行されていない場合は本メソッド内で closeDevice()が実行されます。 ・ startCapture()実行後、stopCapture()が実行されていない場合は本メソッド内で stopCapture()が実行されます。			
引数	型	名前	in/out	説明
	なし			
戻り値	なし			

同期/非同期	同期型

3-4-5-3. getDeviceList

Table 111. Camera::getDeviceList メソッド

機能	接続しているカメラデバイスリストの取得			
書式	Result getDeviceList (std::vector<ConnDevice>& dev_list)			
説明	・ 接続しているカメラデバイスのリストを取得します。			
引数	型	名前	in/out	説明
	std::vector<ConnDevice>&	dev_list	out	接続している カメラデバイスリスト
戻り値	SUCCESS	0	成功	
	ERR_NOT_EXIST	7	対象デバイス未接続	
	ERR_SYSTEM	12	システム異常（メモリ確保失敗など）	
同期/非同期	同期型			

3-4-5-4. openDevice

Table 112. Camera::openDevice メソッド

機能	デバイス Open 処理			
書式	Result openDevice (uint16_t dev_id)			
説明	・ 指定されたカメラデバイスの Open 処理をします。 ・ 引数 dev_id には getDeviceList()で取得したデバイス ID を指定してください。 ・ 既にカメラデバイスが Open されている状態で本メソッドが呼ばれた場合は ERR_BAD_STATE が返ります。 ・ 現在のカメラデバイスの状態で設定できない操作モードをコンストラクタで指定した状態で本メソッドが呼ばれた場合は ERR_BAD_STATE が返ります。			
引数	型	名前	in/out	説明
	uint16_t	dev_id	in	デバイス ID
戻り値	SUCCESS	0	成功	
	ERR_BAD_STATE	6	状態遷移異常	
	ERR_NOT_EXIST	7	対象デバイス未接続	
	ERR_NOT_SUPPORT	11	動作対象カメラの Firmware バージョンが動作対象外	
	ERR_SYSTEM	12	システム異常（メモリ確保失敗など）	
同期/非同期	同期型			

3-4-5-5. closeDevice

Table 113. Camera::closeDevice メソッド

機能	デバイス Close 処理
書式	Result closeDevice (void)

説明	・ openDevice()で Open したカメラデバイスの Close 処理を行います。 ・ openDevice()が実行されていない状態で本メソッドが呼ばれた場合は何も処理せずに SUCCESS が返ります。 ・ startCapture()実行後、stopCapture()が実行されていない場合は本メソッド内で stopCapture()が実行されます。			
引数	型	名前	in/out	説明
	なし			
戻り値	SUCCESS	0	成功	
	ERR_SYSTEM	12	システム異常	
同期/非同期	同期型			

3-4-5-6. getProperty

Table 114. Camera::getProperty メソッド

機能	デバイスパラメータ取得			
書式	Result getProperty (uint16_t prop_cmd, void* param)			
説明	・ openDevice()で Open したカメラデバイスの情報を取得します。 ・ 引数 param の内容は引数 prop_cmd によって異なります。 ・ 引数 prop_cmd と param の引数型・内容についてはプロパティコマンド（カメラデバイス用）およびプロパティコマンド（ファイル再生専用）を参照してください。 ・ openDevice()でカメラデバイスが Open されていない状態で本メソッドが呼ばれた場合は ERR_BAD_STATE が返ります。			
引数	型	名前	in/out	説明
	uint16_t	prop_cmd	in	プロパティコマンド（カメラデバイス用） プロパティコマンド（ファイル再生専用）
	void*	param	in,out	取得パラメータ
戻り値	SUCCESS	0	成功	
	ERR_INVALID_PTR	3	引数のポインタが不正	
	ERR_OVER_RANGE	4	設定値が設定可能範囲外（prop_cmd）	
	ERR_BAD_STATE	6	状態遷移異常	
	ERR_TIMEOUT	8	カメラデバイス間の通信でタイムアウト発生	
	ERR_NOT_SUPPORT	11	未サポート機能	
	ERR_SYSTEM	12	システム異常	
同期/非同期	同期型			

3-4-5-7. setProperty

Table 115. Camera::setProperty メソッド

機能	デバイスパラメータ設定
書式	Result setProperty (uint16_t prop_cmd,

	const void* param = nullptr)
説明	・ openDevice()で Open したカメラデバイスの設定を行います。 ・ 引数 param の内容は引数 prop_cmd によって異なります。 ・ 引数 prop_cmd と param の引数型・内容についてはプロパティコマンド（カメラデバイス用）およびプロパティコマンド（ファイル再生専用）を参照してください。 ・ openDevice()でカメラデバイスが Open されていない状態で本メソッドが呼ばれた場合は ERR_BAD_STATE が返ります。 ・ startCapture()でカメラデバイスから画像出力が行われている状態で本メソッドが呼ばれた場合については各プロパティコマンドの内容を確認してください。

引数	型	名前	in/out	説明
	uint16_t	prop_cmd	in	プロパティコマンド（カメラデバイス用） プロパティコマンド（ファイル再生専用）
	const void*	param	in	設定パラメータ

戻り値			
	SUCCESS	0	成功
	ERR_INVALID_PTR	3	引数のポインタが不正
	ERR_OVER_RANGE	4	設定値が設定可能範囲外
	ERR_BAD_ARG	5	その他、不正引数
	ERR_BAD_STATE	6	状態遷移異常
	ERR_NOT_EXIST	7	対象ディレクトリ・ファイルが存在しない
	ERR_TIMEOUT	8	カメラデバイス間の通信でタイムアウト発生
	ERR_NOT_SUPPORT	11	未サポート機能
	ERR_SYSTEM	12	システム異常

同期/非同期	同期型

3-4-5-8. startCapture

Table 116. Camera::startCapture メソッド

機能	画像出力開始
書式	Result startCapture (void)
説明	・ openDevice()で Open したカメラデバイスの画像出力を開始します。 ・ 画像出力開始後は、capture()にて画像が受信されます。 ・ openDevice()が実行されていない状態で本メソッドが呼ばれた場合は ERR_BAD_STATE が返ります。 ・ 既にカメラデバイスからの画像出力がされている状態で本メソッドが呼ばれた場合は何も処理せず SUCCESS が返ります。

引数	型	名前	in/out	説明
	なし			

戻り値			
	SUCCESS	0	成功
	ERR_BAD_STATE	6	状態遷移異常
	ERR_NOT_EXIST	7	対象デバイス切断（再生対象ファイルがない）
	ERR_SYSTEM	12	システム異常

同期/非同期	同期型

3-4-5-9. stopCapture

Table 117. Camera::stopCapture メソッド

機能	画像出力停止			
書式	Result stopCapture (void)			
説明	・ openDevice()で Open したカメラデバイスの画像出力を停止します。 ・ openDevice()が実行されていない状態で本メソッドが呼ばれた場合は ERR_BAD_STATE が返ります。 ・ startCapture()が実行されていない状態で本メソッドが呼ばれた場合は何も処理せず SUCCESS が返ります。			
引数	型	名前	in/out	説明
	なし			
戻り値	SUCCESS	0	成功	
	ERR_BAD_STATE	6	状態遷移異常	
	ERR_SYSTEM	12	システム異常	
同期/非同期	同期型			

3-4-5-10. capture

Table 118. Camera::capture メソッド

機能	出力画像の取得			
書式	Result capture (Frame& frame, bool block = true)			
説明	・ openDevice()で Open したカメラデバイスの出力画像を取得します。 ・ openDevice()でカメラデバイスが Open されていない状態で本メソッドが呼ばれた場合は ERR_BAD_STATE が返ります。 ・ 引数 frame の各画像種別の受信データバッファ（ImageData.data）は呼び出し元で getProperty(CMD_IMG_FORMAT)で取得できるサイズの領域を確保してください。受信バッファが確保されていない場合は ERR_EMPTY が返ります。 ・ startCapture()で画像出力を開始した後、本メソッドが呼ばれていない場合は、古いフレームから破棄されます。 ・ 引数 block の指定によって、下記のように受信待ち動作が変わります。 【引数 block が true の場合】 ・ 出力画像が受信されるまで本メソッド内で受信待ち状態となります。 ・ 受信待ち状態で cancel()が呼ばれた場合は、受信待ち状態を解除し、CANCELED を返します。 ・ 受信待ち状態で、1 秒以内に画像が受信できない場合は受信待ちがタイムアウトし、ERR_TIMEOUT が返ります。 ・ startCapture()で画像出力が開始されていない状態でも受信待ち状態となります。この時は受信待ちのタイムアウトは発生しません。 【引数 block が false の場合】 ・ 出力画像がない状態では本メソッド内で受信待ち状態にはならず、ERR_NOT_EXIST が返ります。			
引数	型	名前	in/out	説明
	Frame&	frame	in,out	受信フレーム

	bool	block	in	受信待ちフラグ true ：受信待ちする false ：受信待ちしない
戻り値	SUCCESS	0	成功	
	CANCELED	1	待ち状態が解除された	
	REACH_EOF	2	ファイル終端に到達（PLAYBACK のみ）	
	ERR_BAD_STATE	6	状態遷移異常	
	ERR_NOT_EXIST	7	出力画像が存在しない	
	ERR_TIMEOUT	8	受信待ち中にタイムアウト発生	
	ERR_EMPTY	9	バッファなどが空	
	ERR_SYSTEM	12	システム異常	
同期/非同期	同期型			

3-4-5-11. cancel

Table 119. Camera::cancel メソッド

機能	出力画像の受信待ち状態の解除			
書式	Result cancel (void)			
説明	・ capture()の受信待ち状態を解除します。 ・ 待ち状態解除時は capture()で CANCELED が返ります。 ・ openDevice()でカメラデバイスが Open されていない状態で本メソッドが呼ばれた場合は ERR_BAD_STATE が返ります。			
引数	型	名前	in/out	説明
	なし			
戻り値	SUCCESS	0	成功	
	ERR_BAD_STATE	6	状態遷移異常	
	ERR_SYSTEM	12	システム異常	
同期/非同期	同期型			

3-4-6. プロパティコマンド（カメラデバイス用）

3-4-6-1. プロパティコマンド一覧

Camera クラスの Camera::getProperty()、Camera::setProperty()で使用するプロパティコマンドおよび各操作モードにおける Camera::getProperty()、Camera::setProperty()での使用可否について以下に示します。

Table 120. プロパティコマンド一覧（カメラデバイス制御用）

コマンド名	説明	通常操作モード		カメラキャリブレーション操作モード	
		get	set	get	set
CMD_DEV_INFO	デバイス情報	✓	N/A	N/A	N/A

CMD_FOV	FOV 情報	✓	N/A	N/A	N/A
CMD_EXT_TRG_TYPE	外部トリガ種別情報	✓	✓	N/A	N/A
CMD_EXT_TRG_OFFSET	外部トリガ信号オフセット情報	✓	✓	N/A	N/A
CMD_MODE_LIST	動作モードリスト情報	✓	N/A	N/A	N/A
CMD_MODE	動作モード情報	✓	✓	✓	N/A
CMD_IMG_KINDS	出力画像種別情報	✓	✓	N/A	N/A
CMD_IMG_FORMAT	画像フォーマット情報	✓	N/A	✓	N/A
CMD_POSTFILT_INFO	PostFilter 処理情報	✓	N/A	N/A	N/A
CMD_LENS_INFO	Lens 系変換処理用パラメータ	✓	N/A	N/A	N/A
CMD_LIGHT_TIMES	発光回数情報	✓	✓	N/A	N/A
CMD_AE_STATE	AE 機能の状態情報	✓	✓	N/A	N/A
CMD_AE_INTERVAL	AE 機能の制御間隔情報	✓	✓	N/A	N/A
CMD_RAW_SAT_TH	RAW 飽和閾値情報	✓	✓	N/A	N/A
CMD_IR_DARK_TH	IR 無効閾値情報	✓	✓	N/A	N/A
CMD_INT_SUPP_INFO	干渉防止機能情報	✓	✓	N/A	N/A

3-4-6-1-1. CMD_DEV_INFO

Table 121. CMD_DEV_INFO 概要

情報種別	デバイス機器情報
set / get	取得のみ
引数型	DeviceInfo
説明	・ openDevice()で Open しているカメラデバイスの機器情報を取得します。

3-4-6-1-2. CMD_FOV

Table 122. CMD_FOV 概要

情報種別	FOV 情報
set / get	取得のみ
引数型	CamFov
説明	・ openDevice()で Open しているカメラデバイスの FOV 情報を取得します。

3-4-6-1-3. CMD_EXT_TRG_TYPE

Table 123. CMD_EXT_TRG_TYPE 概要

情報種別	外部トリガ種別情報
set / get	取得・設定
引数型	ExtTriggerType
説明	・ openDevice()で Open しているカメラデバイスの外部トリガ種別情報を取得・設定します。 ・ startCapture()でカメラデバイスから画像出力が行われている状態で設定された場合、ERR_BAD_STATE が返ります。 ・ 現在の外部トリガ種別が Slave(EXT_TRG_SLAVE)の時に設定された場合、ERR_BAD_STATE が返ります。

3-4-6-1-4. CMD_EXT_TRG_OFFSET

Table 124. CMD_EXT_TRG_OFFSET 概要

情報種別	外部トリガ信号オフセット情報
set / get	取得・設定
引数型	uint8_t
説明	・ openDevice()で Open しているカメラデバイスの外部トリガ信号オフセット情報を取得・設定します。 ・ startCapture()でカメラデバイスから画像出力が行われている状態で設定された場合、ERR_BAD_STATE が返ります。

3-4-6-1-5. CMD_MODE_LIST

Table 125. CMD_MODE_LIST 概要

情報種別	動作モードリスト情報
set / get	取得のみ
引数型	ModeList
説明	・ openDevice()で Open しているカメラデバイスが使用できる動作モードのリスト情報を取得します。

3-4-6-1-6. CMD_MODE

Table 126. CMD_MODE 概要

情報種別	動作モード情報
set / get	取得・設定
引数型	uint8_t
説明	・ openDevice()で Open しているカメラデバイスに対して、現在の動作モード ID の取得・設定を行います。 ・ 設定時は CMD_MODE_LIST で取得した動作モード ID を指定してください。 ・ openDevice()実行後の初期動作モードは CMD_MODE_LIST で取得できる先頭の動作モード ID となります。 ・ startCapture()でカメラデバイスから画像出力が行われている状態で設定された場合、ERR_BAD_STATE が返ります。

3-4-6-1-7. CMD_IMG_KINDS

Table 127. CMD_IMG_KINDS 概要

情報種別	出力画像種別情報
set / get	取得・設定
引数型	ImgOutKind
説明	・ openDevice()で Open しているカメラデバイスに対して、現在の取得画像種別の取得・設定を行います。 ・ 設定時は CMD_MODE_LIST で取得できる画像種別情報の内、受信する画像種別を指定してください。 ・ openDevice()実行後の初期動作モードは CMD_MODE_LIST で取得できる全画像種別となります。 ・ openDevice()および setProperty(CMD_MODE)実行後の初期出力画像種別はカメラデバイス側で決定されます。 ・ startCapture()でカメラデバイスから画像出力が行われている状態で設定された場合、ERR_BAD_STATE が返ります。

3-4-6-1-8. CMD_IMG_FORMAT

Table 128. CMD_IMG_FORMAT 概要

情報種別	画像フォーマット情報
set / get	取得のみ
引数型	ImageFormats
説明	・ openDevice()で Open しているカメラデバイスに対して、現在の取得画像種別の画像フォーマット情報を取得します。 ・ getProperty(CMD_IMG_KIND)で取得できる画像種別の画像フォーマット情報が取得できます。

3-4-6-1-9. CMD_POSTFILT_INFO

Table 129. CMD_POSTFILT_INFO 概要

情報種別	PostFilter 処理情報
set / get	取得のみ
引数型	PostFiltInfo
説明	・ openDevice()で Open しているカメラデバイスの PostFilter 処理情報を取得します。

3-4-6-1-10. CMD_LENS_INFO

Table 130. CMD_LENS_INFO 概要

情報種別	Lens 系変換処理用パラメータ
set / get	取得のみ
引数型	LensInfo
説明	・ openDevice()で Open しているカメラデバイスの Lens 系変換処理用パラメータを取得します。

3-4-6-1-11. CMD_LIGHT_TIMES

Table 131. CMD_LIGHT_TIMES 概要

情報種別	発光回数情報
set / get	取得・設定
引数型	LightTimesInfo
説明	・ openDevice()で Open しているカメラデバイスの発光回数を取得・設定します。 ・ 取得時は設定可能な最小発光回数(min)、最大発光回数(max)と現在の発光回数(count)が取得できます。 ・ 設定時には最小発光回数(min)、最大発光回数(max)の値は参照しません。発光回数(count)の値のみを設定します。 ・ setProperty(CMD_MODE)で動作モードを切り替えた場合、発光回数は各動作モードの初期値に再設定されます。

3-4-6-1-12. CMD_AE_STATE

Table 132. CMD_AE_STATE 概要

情報種別	AE 機能の状態情報
set / get	取得・設定
引数型	bool
説明	・ openDevice()で Open しているカメラデバイス内の AE 機能の有効/無効情報を取得・設定します。

	・ 発光回数情報(LightTimesInfo)の最小発光回数(min)と最大発光回数(max)の値が同一の場合 AE 機能は常に無効となります。 ・ startCapture()でカメラデバイスから画像出力が行われている状態で設定された場合、ERR_BAD_STATE が返ります。

3-4-6-1-13. CMD_AE_INTERVAL

Table 133. CMD_AE_INTERVAL 概要

情報種別	AE 機能の制御間隔情報
set / get	取得・設定
引数型	AEIntervalInfo
説明	・ openDevice()で Open しているカメラデバイスの AE 機能有効時の制御間隔情報を取得・設定します。 ・ 取得時は設定可能な最小 AE 制御間隔(min)、最大 AE 制御間隔(max)と現在の AE 制御間隔(interval)が取得できます。 ・ 設定時には最小 AE 制御間隔(min)、最大 AE 制御間隔(max)の値は参照しません。AE 制御間隔(interval)の値のみを設定します。 ・ setProperty(CMD_MODE)で動作モードを切り替えた場合、動作モード切替前の AE 制御間隔を引継ぎます。 ・ startCapture()でカメラデバイスから画像出力が行われている状態で設定された場合、ERR_BAD_STATE が返ります。

3-4-6-1-14. CMD_RAW_SAT_TH

Table 134. CMD_RAW_SAT_TH 概要

情報種別	RAW 飽和閾値情報
set / get	取得・設定
引数型	SignalThresholdInfo
説明	・ openDevice()で Open しているカメラデバイスの外部トリガ種別情報を取得・設定します。 ・ 取得時は設定可能な最小飽和閾値(min)、最大飽和閾値(max)と現在の飽和閾値(threshold)が取得できます。 ・ 設定時には最小飽和閾値(min)、最大飽和閾値(max)の値は参照しません。飽和閾値(threshold)の値のみを設定します。 ・ setProperty(CMD_MODE)で動作モードを切り替えた場合、飽和閾値は各動作モードの初期値に再設定されます。

3-4-6-1-15. CMD_IR_DARK_TH

Table 135. CMD_IR_DARK_TH 概要

情報種別	IR 無効閾値情報
set / get	取得・設定
引数型	SignalThresholdInfo
説明	・ openDevice()で Open しているカメラデバイスの外部トリガ種別情報を取得・設定します。 ・ 取得時は設定可能な最小無効閾値(min)、最大無効閾値(max)と現在の無効閾値(threshold)が取得できます。 ・ 設定時には最小無効閾値(min)、最大無効閾値(max)の値は参照しません。無効閾値(threshold)の値のみを設定します。 ・ setProperty(CMD_MODE)で動作モードを切り替えた場合、無効閾値は各動作モードの初期値に再設定されます。

3-4-6-1-16. CMD_INT_SUPP_INFO

Table 136. CMD_INT_SUPP_INFO 概要

情報種別	干渉防止機能情報
set / get	取得・設定
引数型	IntSuppInfo
説明	・ openDevice()で Open しているカメラデバイスの干渉防止機能情報 (動作モード、マニュアルモードパラメータ及びオートパラメータ 1～3) を取得・設定します。 ・ 取得時は数値パラメータの設定可能な最小値(min)、最大値(max)及び現在のパラメータ値(value)が取得できます。 ・ 設定時には最小値(min)、最大値(max)の値は参照しません。パラメータ値(value)の値のみを設定します。

3-4-7. プロパティコマンド（ファイル再生専用）

3-4-7-1. プロパティコマンド一覧

ファイル再生機能で使用できるプロパティコマンドの一覧と Camera::getProperty()、Camera::setProperty()での使用可否を以下に示します。

Table 137. プロパティコマンド（ファイル再生専用）

コマンド名	説明	get	set
CMD_DEV_INFO	デバイス情報	✓	N/A
CMD_FOV	FOV 情報	✓	N/A
CMD_EXT_TRG_TYPE	外部トリガ種別情報	N/A	N/A
CMD_EXT_TRG_OFFSET	外部トリガ信号オフセット情報	N/A	N/A
CMD_MODE_LIST	動作モードリスト情報	✓	N/A
CMD_MODE	動作モード情報	✓	✓
CMD_IMG_KINDS	画像種別情報	✓	N/A
CMD_IMG_FORMAT	画像フォーマット情報	✓	N/A
CMD_POSTFILT_INFO	PostFilter 処理情報	✓	N/A
CMD_LENS_INFO	Lens 系変換処理用パラメータ	✓	N/A
CMD_LIGHT_TIMES	発光回数情報	N/A	N/A
CMD_AE_STATE	AE 機能の状態情報	N/A	N/A
CMD_AE_INTERVAL	AE 機能の制御間隔情報	N/A	N/A
CMD_RAW_SAT_TH	RAW 飽和閾値情報	N/A	N/A
CMD_IR_DARK_TH	IR 無効閾値情報	N/A	N/A
CMD_INT_SUPP_INFO	干渉防止機能情報	N/A	N/A
PlayBack::CMD_PLAY_TARGET	ファイル再生対象ディレクトリ情報	✓	✓
PlayBack::CMD_PLAY_TIME	再生時間情報	✓	✓
PlayBack::CMD_PLAY_STATUS	再生状態情報	✓	N/A
PlayBack::CMD_PAUSE	再生一時停止制御	N/A	✓
PlayBack::CMD_FAST_PLAY	倍速再生制御	N/A	✓

PlayBack::CMD_SLOW_PLAY	スロー再生制御	N/A	✓
PlayBack::CMD_JUMP_FW	先送り制御	N/A	✓
PlayBack::CMD_JUMP_BW	巻き戻し制御	N/A	✓

以降、ファイル再生機能でのみ使用可能なプロパティコマンドの詳細を記載します。

3-4-7-1-1. PlayBack::CMD_PLAY_TARGET

Table 138. PlayBack::CMD_PLAY_TARGET 概要

情報種別	ファイル再生対象ディレクトリ情報
set / get	取得・設定
引数型	std::filesystem::path
説明	・ 再生対象とするディレクトリの取得・設定を行います。 ・ 設定時のディレクトリに再生可能なディレクトリ・ファイルが存在しない場合、ERR_NOT_EXIST が返ります。 ・ ファイル再生対応フォーマットバージョンがサポート対象のバージョンではない場合、ERR_NOT_SUPPORT が返ります。 ・ startCapture() でファイル再生が行われている状態で設定された場合、ERR_BAD_STATE が返ります。

3-4-7-1-2. PlayBack::CMD_PLAY_TIME

Table 139. PlayBack::CMD_PLAY_TIME 概要

情報種別	再生時間情報
set / get	取得・設定
引数型	PlayBack::PlayTime
説明	・ 現在の再生時間の取得・設定を行います。 ・ 設定時は引数内の current のみ参照し、total の値は参照しません。 ・ 一時停止状態で設定された場合、指定時間のフレームを capture()にて出力後、一時停止状態となります。 ・ 再生可能なディレクトリ・ファイルが設定されていない場合、ERR_NOT_EXIST が返ります。 ・ 再生対象の総再生時間を超える時間を設定された場合、ERR_OVER_RANGE が返ります。 ・ 停止状態で設定された場合、ERR_BAD_STATE が返ります。

3-4-7-1-3. PlayBack::CMD_PLAY_STATUS

Table 140. PlayBack::CMD_PLAY_STATUS 概要

情報種別	再生状態情報
set / get	取得のみ
引数型	PlayBack::PlayStatus
説明	・ ファイル再生の状態を取得します。 ・ 倍速再生・スロー再生中は getProperty(CMD_MODE_LIST)で取得できる受信フレームレートに対して、倍速再生時は 2 倍速、 3 倍速、 4 倍速、スロー再生時は½倍、⅓倍、¼倍のフレームレートで受信することになります。したがって、倍速再生・スロー再生中の受信フレームレートは本情報を使用してください。 ・ 再生状態が倍速再生・スロー再生中以外の状態では getProperty(CMD_MODE_LIST)で取得できる受信フレームレートと同じになります。

3-4-7-1-4. PlayBack::CMD_PAUSE

Table 141. PlayBack::CMD_PAUSE 概要

情報種別	再生一時停止制御
set / get	設定のみ
引数型	なし(nullptr)
説明	・ 再生状態、倍速再生状態、スロー再生状態で設定された場合、一時停止状態に遷移します。 ・ 一時停止状態で設定された場合、再生状態に戻ります。 ・ 停止状態で設定された場合、ERR_BAD_STATE が返ります。

3-4-7-1-5. PlayBack::CMD_FAST_PLAY

Table 142. PlayBack::CMD_FAST_PLAY 概要

情報種別	倍速再生制御
set / get	設定のみ
引数型	なし(nullptr)
説明	・ 再生状態で設定された場合、倍速再生状態（2 倍速）に遷移します。 ・ 倍速再生状態で設定された場合、2 倍速の時は 3 倍速、3 倍速の時は 4 倍速に遷移します。 ・ 4 倍速もしくは受信フレームレートが 120fps を超える場合は ERR_OVER_RANGE が返ります。 ・ スロー再生状態で設定された場合、¼倍速の時は⅓倍速、⅓倍速の時は½倍速、½倍速の時は再生状態に遷移します。 ・ 停止状態、一時停止状態で設定された場合、ERR_BAD_STATE が返ります。

3-4-7-1-6. PlayBack::CMD_SLOW_PLAY

Table 143. PlayBack::CMD_SLOW_PLAY 概要

情報種別	スロー再生制御
set / get	設定のみ
引数型	なし(nullptr)
説明	・ 再生状態で設定された場合、スロー再生状態（½倍速）に遷移します。 ・ スロー再生状態で設定された場合、½倍速の時は⅓倍速、⅓倍速の時は¼倍速に遷移します。 ・ ¼倍速もしくは受信フレームレートが 10fps を下回る場合は ERR_OVER_RANGE が返ります。 ・ 倍速再生状態で設定された場合、4 倍速の時は 3 倍速、3 倍速の時は 2 倍速、2 倍速の時は再生状態に遷移します。 ・ 停止状態、一時停止状態で設定された場合、ERR_BAD_STATE が返ります。

3-4-7-1-7. PlayBack::CMD_JUMP_FW

Table 144. PlayBack::CMD_JUMP_FW 概要

情報種別	先送り制御
set / get	設定のみ
引数型	uint32_t
説明	・ 引数で指定されたフレーム分、先送りを行います。 ・ 一時停止状態で設定された場合、先送り後のフレームを capture()にて出力後、一時停止状態となります。

	・ 先送りを行った結果、終端フレームを超える場合は停止状態に遷移し、capture()にて REACH_EOF を返します。 ・ 停止状態で設定された場合、ERR_BAD_STATE が返ります。

3-4-7-1-8. PlayBack::CMD_JUMP_BW

Table 145. PlayBack::CMD_JUMP_BW 概要

情報種別	巻き戻し制御
set / get	設定のみ
引数型	uint32_t
説明	・ 引数で指定されたフレーム分、巻き戻しを行います。 ・ 一時停止状態で設定された場合、巻き戻し後のフレームを capture()にて出力後、一時停止状態となります。 ・ 巻き戻しを行った結果、先頭フレームより前の場合は先頭フレームから再生します。 ・ 停止状態で設定された場合、ERR_BAD_STATE が返ります。

3-4-8. Cameraクラス用定義

CameraType.h に記載している Camera クラスで使用する定義を以下に示します。

3-4-8-1. 列挙体定義

3-4-8-1-1. 列挙体定義一覧

Table 146. 列挙体定義一覧

名称	説明
CameraType	カメラデバイスの種別
PropCmd	プロパティコマンド
OperationMode	カメラデバイスの操作モード
ExtTriggerType	外部トリガ種別
ImgOutKind	出力画像種別の組み合わせ情報
CamPrmKind	Camera 内パラメータの種別
RegDevType	レジスタ制御用デバイス種別
IntSuppModeType	干渉防止機能モード種別

3-4-8-1-2. 列挙体定義詳細

3-4-8-1-2-1.CameraType

Table 147. CameraType 定義

定義	enum CameraType {
	C11U_USB = 0,
	PLAYBACK,
	};		

説明	・ 使用するカメラデバイスの種別を示します。		
値	**名前**	**値**	**説明**
	C11U_USB	0	3D ToF Camera カメラ（C11U）

	PLAYBACK	1	ファイル再生用
参照	3-4-5-1 Camera		

3-4-8-1-2-2.PropCmd

Table 148. PropCmd 定義

定義	enum PropCmd : uint16_t {
	CMD_DEV_INFO = 0,
	CMD_FOV,
	CMD_EXT_TRG_TYPE,
	CMD_EXT_TRG_OFFSET,
	CMD_MODE_LIST,
	CMD_MODE,
	CMD_IMG_KINDS,
	CMD_IMG_FORMAT,
	CMD_POSTFILT_INFO,
	CMD_LENS_INFO,
	CMD_LIGHT_TIMES,
	CMD_AE_STATE,
	CMD_AE_INTERVAL,
	CMD_RAW_SAT_TH,
	CMD_IR_DARK_TH,
	CMD_INT_SUPP_INFO,
	CMD_CAM_PRM,
	CMD_REG_DEVS,
	CMD_REG,
	CMD_REG_LIST,
	CMD_REG_REFRESH,
	CMD_OP_MODE,
	CMD_CALB_OP_INFO,
	CMD_USB_PC_ACC_KEY,
	CMD_USB_PC_UPDATE,
	CMD_MAX
	};

説明	・ プロパティコマンドとして使用するコマンドを示します。

値	名前	値	説明
	CMD_DEV_INFO	0	デバイス情報
	CMD_FOV	1	FOV 情報
	CMD_EXT_TRG_TYPE	2	外部トリガ種別情報
	CMD_EXT_TRG_OFFSET	3	外部トリガオフセット情報
	CMD_MODE_LIST	4	動作モードリスト情報
	CMD_MODE	5	動作モード情報
	CMD_IMG_KINDS	6	出力画像種別情報
	CMD_IMG_FORMAT	7	画像フォーマット情報
	CMD_POSTFILT_INFO	8	PostFilter 処理情報
	CMD_LENS_INFO	9	Lens 系変換処理用パラメータ
	CMD_LIGHT_TIMES	10	発光回数情報

	CMD_AE_STATE	11	AE 状態情報
	CMD_AE_INTERVAL	12	AE 間隔情報
	CMD_RAW_SAT_TH	13	RAW 飽和閾値情報
	CMD_IR_DARK_TH	14	IR 無効閾値情報
	CMD_INT_SUPP_INFO	15	干渉防止機能情報
	CMD_CAM_PRM	16	Camera 内パラメータ情報
	CMD_REG_DEVS	17	レジスタ制御可能デバイス情報
	CMD_REG	18	レジスタ情報
	CMD_REG_LIST	19	レジスタリスト情報
	CMD_REG_REFRESH	20	レジスタ書き込み状態解除
	CMD_OP_MODE	21	操作モード情報
	CMD_CALB_OP_INFO	22	カメラキャリブレーション操作モード画像情報
	CMD_USB_PC_ACC_KEY	23	USB ペリフェラルコントローラアクセスキー
	CMD_USB_PC_UPDATE	24	USB ペリフェラルコントローラアップデートモード設定
	CMD_MAX	25	共通コマンド値上限
参照	3-4-5-6. getProperty, 3-4-5-7. setProperty, 3-4-6 プロパティコマンド（カメラデバイス用）, 3-4-7 プロパティコマンド（ファイル再生専用）		

3-4-8-1-2-3.OperationMode

Table 149. OperationMode 定義

定義	enum OperationMode {
	OP_NORMAL = 0,
	OP_CALIB,
	};		

説明	・ カメラデバイスの操作モードを示します。		
値	名前	値	説明
	OP_NORMAL	0	通常操作モード
	OP_CALIB	1	カメラキャリブレーション操作モード
参照	3-4-8-2-2-3. OpInfo		

3-4-8-1-2-4.ExtTriggerType

Table 150. ExtTriggerType 定義

定義	enum ExtTriggerType : uint8_t {
	EXT_TRG_STANDALONE = 1U,
	EXT_TRG_SLAVE,
	EXT_TRG_MASTER,
	};		

説明	・ カメラデバイスの外部トリガ種別を示します。		
値	名前	値	説明
	EXT_TRG_STANDALONE	1	Standalone
	EXT_TRG_SLAVE	2	Secondary (Slave)

	EXT_TRG_MASTER	3	Primary (Mater)
参照	3-4-6-1-3. CMD_EXT_TRG_TYPE		

3-4-8-1-2-5.ImgOutKind

Table 151. ImgOutKind 定義

定義	enum ImgOutKind {
	OUT_IMG_DEPTH = 0,
	OUT_IMG_IR,
	OUT_IMG_DEPTH_IR,
	OUT_IMG_DEPTH_IR_RAW,
	OUT_IMG_RAW
	};		

説明	・ カメラデバイスから出力される画像種別の組み合わせ情報を示します。 ・ Gate RAW データはデバッグ機能です。(出力不要)		
値	名前	値	説明
	OUT_IMG_DEPTH	0	Depth 画像出力
	OUT_IMG_IR	1	IR 画像出力
	OUT_IMG_DEPTH_IR	2	Depth 画像 + IR 画像出力
	OUT_IMG_DEPTH_IR_RAW	3	Depth 画像 + IR 画像 + センサ Gate RAW 画像出力
	OUT_IMG_RAW	4	センサ RAW 画像出力
参照	3-4-8-2-2-8. ModeInfo, 3-4-6-1-7. CMD_IMG_KINDS, 3-4-8-2-2-2. CalbOpInfo		

3-4-8-1-2-6.CamPrmKind

Table 152. CamPrmKind 定義

定義	enum CamPrmKind : uint8_t {
	CAM_PRM_ALL,
	CAM_PRM_PART,
	CAM_PRM_ARTIX_CONF,
	CAM_PRM_FIRM_2
	};		

説明	・ Camera 内パラメータの種別を示します。		
値	名前	値	説明
	CAM_PRM_ALL	0	全パラメータ
	CAM_PRM_PART	1	部分パラメータ
	CAM_PRM_ARTIX_CONF	2	フラッシュ ROM 領域 1
	CAM_PRM_FIRM_2	3	フラッシュ ROM 領域 2
参照	3-4-8-2-2-12. CamPrmRequest		

3-4-8-1-2-7.RegDevType

Table 153. RegDevType 定義

定義	enum RegDevType {
	REG_IMG_SENSOR = 0,
	REG_IMG_PROCESSOR,
	};

説明	・ レジスタ制御するデバイス種別を示します。		
値	名前	値	説明
	REG_IMG_SENSOR	0	イメージセンサ
	REG_IMG_PROCESSOR	1	画像処理デバイス
参照	3-4-8-2-2-13. RegDevInfo, 3-4-8-2-2-14. RegInfo, 3-4-8-2-2-15. RegList		

3-4-8-1-2-8.IntSuppModeType

Table 154. IntSuppModeType 定義

定義	enum IntSuppModeType : uint8_t {
	INT_SUPP_MODE_OFF = 0, */
	INT_SUPP_MODE_MANUAL,
	INT_SUPP_MODE_AUTO,
	};		

説明	・ 干渉防止機能モードの種別を示します。 ・ 自動モードは実験的機能のため、使用非推奨とします。		
値	名前	値	説明
	INT_SUPP_MODE_OFF	0	干渉防止機能 Off
	INT_SUPP_MODE_MANUAL	1	マニュアルモード
	INT_SUPP_MODE_AUTO	2	自動モード(実験機能)
参照	3-4-8-2-2-17. IntSuppInfo		

3-4-8-2. 構造体定義

3-4-8-2-1. 構造体定義一覧

Table 155. 構造体定義一覧

名称	説明
ConnDevice	接続デバイス情報
CalbOpInfo	カメラキャリブレーション操作モード時の受信設定情報
OpInfo	カメラデバイスの操作モード
DeviceInfo	カメラデバイスの機器情報
PostFiltInfo	PostFilter 処理情報
LensInfo	Lens 系変換処理で使用するパラメータ情報
CamFov	カメラデバイスの FOV 情報
ModeInfo	動作モード情報
LightTimesInfo	発光回数情報
AEIntervalInfo	AE 機能有効時の制御間隔情報
SignalThresholdInfo	RAW 飽和閾値情報、IR 無効閾値情報
CamPrmRequest	Camera 内パラメータ情報
RegDevInfo	カメラデバイスのレジスタに関する情報
RegInfo	レジスタ情報
RegList	レジスタ情報（連続アドレス）
IntSuppParamInfo	干渉防止機能モードパラメータ情報

IntSuppInfo	干渉防止機能情報
UsbPCAccKey	USB ペリフェラルコントローラアクセスキー情報

3-4-8-2-2. 構造体定義詳細

3-4-8-2-2-1.ConnDevice

Table 156. ConnDevice 定義

定義	struct ConnDevice {
	uint16_t id;
	std::string name;
	};		

説明	・ 接続中のデバイス情報を示します。		
引数	**型**	**名前**	**説明**
	uint16_t	id	デバイス ID
	std::string	name	デバイス名
参照	3-4-5-3. getDeviceList		

3-4-8-2-2-2.CalbOpInfo

Table 157. CalbOpInfo 定義

定義	struct CalbOpInfo {
	ImgOutKind kind;
	uint16_t image_w;
	uint16_t image_h;
	uint8_t num_raw;
	uint16_t fps;
	};		

説明	・ カメラキャリブレーション操作モード時の受信設定情報を示します。		
引数	**型**	**名前**	**説明**
	ImgOutKind	kind	受信画像種別
	uint16_t	image_w	センサ出力画像幅 [pixel]
	uint16_t	image_h	センサ出力画像高さ [pixel]
	uint8_t	num_raw	RAW 画像数
	uint16_t	fps	受信フレームレート [fps × 100]
参照	3-4-8-1-2-5. ImgOutKind, 3-4-8-2-2-3. OpInfo		

3-4-8-2-2-3.OpInfo

Table 158. OpInfo 定義

定義	struct OpInfo {
	OperationMode mode;
	CalbOpInfo calib_prm;
	};		

説明	・ カメラデバイスに対する操作モード情報を示します。 ・ calib_prm は mode が OP_CALB の場合のみ有効です。		
引数	**型**	**名前**	**説明**
	OperationMode	mode	操作モード

	CalbOpInfo	calib_prm	カメラキャリブレーション操作モード用パラメータ
参照	3-4-5-1. Camera, 3-4-8-2-2-2. CalbOpInfo		

3-4-8-2-2-4.DeviceInfo

Table 159. DeviceInfo 定義

定義	struct DeviceInfo {
	uint32_t hw_kind;
	uint32_t serial_no;
	Version map_ver;
	uint32_t adjust_no;
	Version firm_ver;
	uint16_t ld_wave;
	uint16_t ld_enable;
	uint16_t correct_calib;
	};		

説明	・ カメラデバイスの機器情報を示します。		
引数	型	名前	説明
	uint32_t	hw_kind	HW 品種番号 上位 16bit : センサ品種番号 下位 16bit : レンズ品種番号
	uint32_t	serial_no	機器識別番号
	Version	map_ver	カメラ内設定 MAP バージョン
	uint32_t	adjust_no	調整番号
	Version	firm_ver	カメラ Firmware バージョン
	uint16_t	ld_wave	光源波長 [nm]
	uint16_t	ld_enable	各光源の有効情報（灯数情報） 最下位 bit から順に各光源の有効情報が入ります。（0b:無効, 1b:有効） [0] LD1, [1] LD2, [2] LD3 …
	uint16_t	correct_calib	補正キャリブレーション Revison
参照	3-2-3-2-1. Version, 3-4-6-1-1. CMD_DEV_INFO		

3-4-8-2-2-5.PostFiltInfo

Table 160. LensInfo 定義

定義	struct PostFiltInfo {
	bool cam_med_filt;
	bool cam_bil_filt;
	bool cam_fly_p_filt;
	};		

説明	・ PostFilter 処理情報を示します。		
引数	型	名前	説明
	bool	cam_med_filt	カメラデバイス内でメディアンフィルタ実施
	bool	cam_bil_filt	カメラデバイス内でバイラテラルフィルタ実施
	bool	cam_fly_p_filt	カメラデバイス内でフライングピクセルフィルタ実施
参照	3-4-6-1-9. CMD_POSTFILT_INFO		

3-4-8-2-2-6.LensInfo

Table 161. LensInfo 定義

定義	struct LensInfo {
	uint16_t sens_w;
	uint16_t sens_h;
	uint32_t focal_len;
	uint8_t thin_w;
	uint8_t thin_h;
	Point2d crop;
	bool cam_dist;
	uint64_t dist[9];
	uint16_t lens_calib;
	};		

説明	・ Lens 系変換処理で使用するパラメータ情報を示します。 ・ thin_w, thin_h, crop は ModeInfo 内の同情報と同じ内容になります。 ・ cam_dist が true の場合、dist の内容は無効になります。		
引数	型	名前	説明
	uint16_t	sens_w	センサ横幅 [pixel]
	uint16_t	sens_h	センサ高さ [pixel]
	uint32_t	focal_len	焦点距離（固定小数点: 整数部 12bit,小数部 20bit）
	uint8_t	thin_w	水平間引き数（1 / thin_w）
	uint8_t	thin_h	垂直間引き数（1 / thin_h）
	Point2d	crop	センサ画素からの画像切り出し位置
	uint64_t	dist[9]	歪曲補正用パラメータ [fx, fy, cx, cy, k1, k2, p1, p2, k3] （固定小数点：符号部 1bit,整数部 16bit,小数部 47bit）
	bool	cam_dist	カメラデバイス内で歪曲補正実施
	uint16_t	lens_calib	レンズキャリブレーション Revision
参照	3-2-3-2-2. Point2d, 3-4-6-1-10. CMD_LENS_INFO		

3-4-8-2-2-7.CamFov

Table 162. CamFov 定義

定義	struct CamFov {
	uint16_t horz;
	uint16_t vert;
	};		

説明	・ カメラデバイスの FOV 情報を示します。		
引数	型	名前	説明
	uint16_t	horz	水平視野角 [degree × 100]
	uint16_t	vert	垂直視野角 [degree × 100]
参照	3-4-6-1-2. CMD_FOV		

3-4-8-2-2-8.ModeInfo

Table 163. ModeInfo 定義

<table>
<tr><td>定義</td><td colspan="3">struct ModeInfo {
uint8_t id;
std::string description;
std::vector<ImgOutKind> img_out;
Range dist_range;
uint16_t fps;
uint8_t thin_w;
uint8_t thin_h;
Point2d crop;
bool light_times;
uint16_t range_calib;
};</td></tr>
<tr><td>説明</td><td colspan="3">・ 動作モード情報を示します。</td></tr>
<tr><td rowspan="10">引数</td><td>型</td><td>名前</td><td>説明</td></tr>
<tr><td>uint8_t</td><td>id</td><td>動作モード ID</td></tr>
<tr><td>std::string</td><td>description</td><td>動作モード説明</td></tr>
<tr><td>std::vector<ImgOutKind></td><td>img_out</td><td>受信可能画像種別</td></tr>
<tr><td>Range</td><td>dist_range</td><td>測距範囲 [mm]</td></tr>
<tr><td>uint16_t</td><td>fps</td><td>受信フレームレート [fps × 100]</td></tr>
<tr><td>uint8_t</td><td>thin_w</td><td>水平間引き数（1 / thin_w）</td></tr>
<tr><td>uint8_t</td><td>thin_h</td><td>垂直間引き数（1 / thin_h）</td></tr>
<tr><td>Point2d</td><td>crop</td><td>センサ画素からの画像切り出し位置</td></tr>
<tr><td>bool</td><td>light_times</td><td>発光回数の変更可否
（true：変更可能、false：変更不可）</td></tr>
<tr><td>uint16_t</td><td>range_calib</td><td>測距キャリブレーション Revision</td></tr>
<tr><td>参照</td><td colspan="3">3-4-8-1-2-5. ImgOutKind, 3-2-3-2-5. Range, 3-4-8-3-2-1. ModeList</td></tr>
</table>

3-4-8-2-2-9.LightTimesInfo

Table 164. LightTimesInfo 定義

<table>
<tr><td>定義</td><td colspan="3">struct LightTimesInfo {
uint32_t min;
uint32_t max;
uint32_t count;
};</td></tr>
<tr><td>説明</td><td colspan="3">・ 発光回数の情報を示します。</td></tr>
<tr><td rowspan="4">引数</td><td>型</td><td>名前</td><td>説明</td></tr>
<tr><td>uint32_t</td><td>min</td><td>発光回数下限値</td></tr>
<tr><td>uint32_t</td><td>max</td><td>発光回数上限値</td></tr>
<tr><td>uint32_t</td><td>count</td><td>発光回数現在値</td></tr>
<tr><td>参照</td><td colspan="3">3-4-6-1-11. CMD_LIGHT_TIMES</td></tr>
</table>

3-4-8-2-2-10.AEIntervalInfo

Table 165. AEIntervalInfo 定義

定義	struct AEIntervalInfo {
	uint8_t min;
	uint8_t max;
	uint8_t interval;
	};		

説明	・ AE 機能有効時の制御間隔情報を示します。		
引数	型	名前	説明
	uint8_t	min	AE 間隔下限値 [frame]
	uint8_t	max	AE 間隔上限値 [frame]
	uint8_t	interval	AE 間隔現在値 [frame]
参照	3-4-6-1-13. CMD_AE_INTERVAL		

3-4-8-2-2-11.SignalThresholdInfo

Table 166. SignalThresholdInfo 定義

定義	struct SignalThresholdInfo {
	uint16_t min;
	uint16_t max;
	uint16_t threshold;
	};		

説明	・ RAW 飽和閾値または IR 無効閾値の情報を示します。		
引数	型	名前	説明
	uint16_t	min	RAW 振幅データの閾値下限値
	uint16_t	max	IR 振幅データの閾値上限値
	uint16_t	threshold	現在の閾値値
参照	3-4-6-1-14. CMD_RAW_SAT_TH, 3-4-6-1-15. CMD_IR_DARK_TH		

3-4-8-2-2-12.CamPrmRequest

Table 167. CamPrmRequest 定義

定義	struct CamPrmRequest {
	CamPrmKind kind;
	uint16_t id;
	std::vector<uint8_t> data;
	};		

説明	・ Camera 内パラメータ情報を示します。 ・ パラメータ ID はパラメータ種別が CAM_PRM_PART の場合のみ有効になります。		
引数	型	名前	説明
	CamPrmKind	kind	Camera 内パラメータ種別
	uint16_t	id	パラメータ ID
	std::vector<uint8_t>	data	Camera 内パラメータデータ
参照	3-4-8-1-2-6. CamPrmKind		

3-4-8-2-2-13.RegDevInfo

Table 168. RegDevInfo 定義

定義	struct RegDevInfo {
	RegDevType target;
	uint8_t addr_len;
	uint8_t val_len;
	uint8_t list_len;
	};		

説明	・ カメラデバイスのレジスタに関する情報を示します。		
引数	型	名前	説明
	RegDevType	target	制御対象デバイス
	uint8_t	addr_len	レジスタアドレスのサイズ [byte]
	uint8_t	val_len	レジスタ値のサイズ [byte]
	uint8_t	list_len	連続 R/W 時の最大レジスタ数
参照	3-4-8-1-2-7. RegDevType, 3-4-8-3-2-2. RegDevs		

3-4-8-2-2-14.RegInfo

Table 169. RegInfo 定義

定義	struct RegInfo {
	RegDevType target;
	uint16_t addr;
	uint16_t value;
	};		

説明	・ カメラデバイスのレジスタ情報を示します。		
引数	型	名前	説明
	RegDevType	target	制御対象レジスタ
	uint16_t	addr	レジスタアドレス
	uint16_t	value	レジスタ値
参照	3-4-8-1-2-7. RegDevType		

3-4-8-2-2-15.RegList

Table 170. RegList 定義

定義	struct RegList {
	RegDevType target;
	uint16_t addr;
	std::vector<uint16_t> values;
	};		

説明	・ カメラデバイスの連続アドレスとなるレジスタのリスト情報を示します。		
引数	型	名前	説明
	RegDevType	target	制御対象レジスタ
	uint16_t	addr	先頭レジスタアドレス
	std::vector<uint16_t>	values	レジスタ値リスト
参照	3-4-8-1-2-7. RegDevType		

3-4-8-2-2-16.IntSuppParamInfo

Table 171. IntSuppParamInfo 定義

定義	struct IntSuppParamInfo {
	uint8_t min;
	uint8_t max;
	uint8_t value;
	};		

説明	・ 干渉防止機能のパラメータを示します。		
引数	型	名前	説明
	uint8_t	min	パラメータの設定可能な最小値
	uint8_t	max	パラメータの設定可能な最大値
	uint8_t	values	パラメータ値
参照	3-4-8-2-2-17. IntSuppInfo		

3-4-8-2-2-17.IntSuppInfo

Table 172. IntSuppInfo 定義

定義	struct IntSuppInfo {
	IntSuppModeType mode;
	IntSuppParamInfoprm_m;
	IntSuppParamInfoprm_a1;
	IntSuppParamInfoprm_a2;
	IntSuppParamInfoprm_a3;
	};		

説明	・ 干渉防止機能情報を示します。		
引数	型	名前	説明
	IntSuppModeType	mode	干渉防止機能モード
	IntSuppParamInfo	prm_m	マニュアルモードパラメータ
	IntSuppParamInfo	prm_a1	自動モードパラメータ 1 (実験用) 値:31
	IntSuppParamInfo	prm_a2	自動モードパラメータ 2 (実験用) 値:65
	IntSuppParamInfo	prm_a3	自動モードパラメータ 3 (実験用) 値:4
参照	3-4-8-1-2-8. IntSuppModeType, 3-4-8-2-2-16. IntSuppParamInfo, 3-4-6-1-16. CMD_INT_SUPP_INFO		

3-4-8-2-2-18.UsbPCAccKey

Table 173. UsbPCAccKey 定義

定義	struct UsbPCAccKey {
	uint16_t key;
	};		

説明	・ USB ペリフェラルコントローラへのアクセスキーを示します。		
引数	型	名前	説明
	uint16_t	key	16 ビットのアクセスキー
参照	-		

3-4-8-3. 型定義

3-4-8-3-1. 型定義一覧

Table 174. 型定義一覧

名称	説明
ModeList	動作モードのリスト
RegDevs	レジスタ制御可能なデバイス情報リスト

3-4-8-3-2. 型定義詳細

3-4-8-3-2-1.ModeList

Table 175. ModeList 定義

定義	using ModeList = std::vector<ModeInfo>;
説明	・　動作モードのリスト情報を示します。
参照	3-4-8-2-2-8. ModeInfo, 3-4-6-1-5. CMD_MODE_LIST

3-4-8-3-2-2.RegDevs

Table 176. RegDevs 定義

定義	using RegDevs = std::vector<RegDevInfo>;
説明	・　レジスタ制御可能なデバイス情報を示します。
参照	3-4-8-2-2-13. RegDevInfo

3-4-9. ファイル再生機能用定義

PlayBackType.h に記載している Camera クラスのファイル再生機能のみで使用する定義を以下に示します。

3-4-9-1. 列挙体定義

3-4-9-1-1. 列挙体定義一覧

Table 177. 列挙体定義一覧

名称	説明
PlayBack::PlayBackCmd	ファイル再生機能のみで使用可能なプロパティコマンド
PlayBack::PlayState	ファイル再生の状態

3-4-9-1-2. 列挙体定義詳細

3-4-9-1-2-1.PlayBack::PlayBackCmd

Table 178. PlayBack::PlayBackCmd 定義

定義	enum PlayBackCmd : uint16_t { CMD_PLAY_TARGET = CMD_MAX, CMD_PLAY_TIME, CMD_PLAY_STATUS, CMD_PAUSE, CMD_FAST_PLAY, CMD_SLOW_PLAY,

	CMD_JUMP_FW, CMD_JUMP_BW, };
説明	・ ファイル再生機能のみで使用可能なプロパティコマンドを示します。

値	名前	値	説明
	CMD_PLAY_TARGET	12	ファイル再生対象ディレクトリ情報
	CMD_PLAY_TIME	13	再生時間情報
	CMD_PLAY_STATUS	14	再生状態情報
	CMD_PAUSE	15	再生一時停止制御
	CMD_FAST_PLAY	16	倍速再生制御
	CMD_SLOW_PLAY	17	スロー再生制御
	CMD_JUMP_FW	18	先送り制御
	CMD_JUMP_BW	19	巻き戻し制御

参照	3-4-5-6. getProperty, 3-4-5-7. setProperty, 3-4-7. プロパティコマンド（ファイル再生専用）

3-4-9-1-2-2.PlayBack::PlayState

Table 179. PlayBack:: PlayState 定義

定義	enum PlayState {
	STOPPED,
	PLAYING,
	PAUSE,
	FAST,
	SLOW
	};

説明	・ ファイル再生の状態を示します。

値	名前	値	説明
	STOPPED	0	停止状態
	PLAYING	1	再生状態（等速再生）
	PAUSE	2	一時停止状態
	FAST	3	倍速再生状態
	SLOW	4	スロー再生状態

参照	3-4-9-2-2-3. PlayBack::PlayStatus

3-4-9-2. 構造体定義

3-4-9-2-1. 構造体定義一覧

Table 180. 構造体定義一覧

名称	説明
PlayBack::ConfigParam	ファイル再生時の初期パラメータ
PlayBack::PlayTime	再生時間情報
PlayBack::PlayStatus	ファイル再生機能の状態

3-4-9-2-2. 構造体定義詳細

3-4-9-2-2-1.PlayBack::ConfigParam

Table 181. PlayBack::ConfigParam 定義

定義	struct ConfigParam {
	std::filesystem::path path;
	};		

説明	・ ファイル再生時の初期パラメータを示します。		
引数	型	名前	説明
	std::filesystem::path	path	再生対象ディレクトリ
参照	3-4-5-1. Camera		

3-4-9-2-2-2.PlayBack::PlayTime

Table 182. PlayBack::PlayTime 定義

定義	struct PlayTime {
	uint32_t total;
	uint32_t current;
	};		

説明	・ 再生時間情報を示します。 ・ フレーム番号を再生時間に変換する場合は、getProperty(CMD_MODE_LIST)で取得できる受信フレームレート情報(fps)を使用してください。 再生時間[sec] = フレーム番号 ÷ fps		
引数	型	名前	説明
	uint32_t	total	全体のフレーム数
	uint32_t	current	現在のフレーム番号（0 始まり）
参照	3-4-7-1-2. PlayBack::CMD_PLAY_TIME		

3-4-9-2-2-3.PlayBack::PlayStatus

Table 183. PlayBack::PlayStatus 定義

定義	struct PlayStatus {
	PlayState state;
	uint16_t playing_fps;
	};		

説明	・ ファイル再生機能の状態情報を示します。 ・ playing_fps は再生中のフレームレートを示し、state が再生状態（等速再生）の場合、実動作もオリジナルのフレームレートとなるため、getProperty(CMD_MODE_LIST)で取得できる受信フレームレートと同じになります。倍速再生状態、スロー再生状態ではオリジナルのフレームレートに対して倍速・スローとなった後のフレームレートになります。（例：オリジナルが 30fps 時、２倍速再生の場合は 60fps、½倍速スロー再生の場合は 15fps）		
引数	型	名前	説明
	PlayBack::PlayState	state	ファイル再生の状態
	uint16_t	playing_fps	再生中のフレームレート [fps × 100]
参照	3-4-9-1-2-2. PlayBack::PlayState, 3-4-7-1-3. PlayBack::CMD_PLAY_STATUS		

3-5. Record クラス

3-5-1. 概要

Table 184. Record クラス概要

提供ヘッダファイル	Record.h	Record クラス定義
	RecordType.h	Record クラス型定義
所属名前空間	krm::Record	
説明	カメラデバイスの出力画像をファイルに保存する C++クラス	
スレッドセーフ	Record クラスはスレッドセーフとなっています。 利用するユーザプログラム側で排他処理は必要ありません。	

3-5-2. 提供機能

3-5-2-1. 機能概要

Record クラスの提供する機能概要を以下に示します。

Table 185. Record クラス機能概要

提供機能	機能概要
ファイル保存機能	画像データをファイルとして保存する機能です。 入力として受け取った Depth 画像、IR 画像、センサ RAW 画像およびフレームの付加情報をファイルとして出力します。

3-5-2-1-1. 保存先データ構成

Record クラスによるファイル保存後のディレクトリ構成を以下に示します。

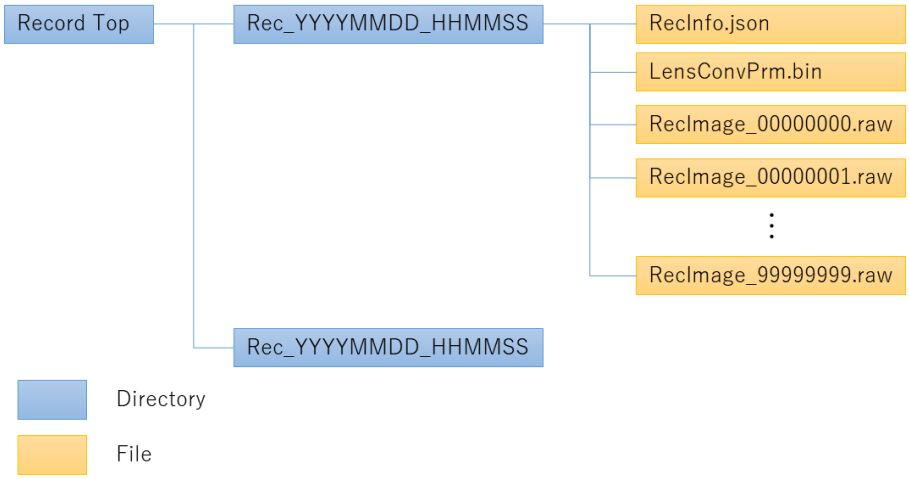


Figure 21. 保存先ディレクトリ構成

3-5-2-1-1-1.Rec_YYYYMMDD_HHMMSS ディレクトリ

Record::openRec()による保存先ディレクトリ初期化処理時に保存開始時点の日時でディレクトリ（YYYYMMDD_HHMMSS ＝ YYYY: 年、MM: 月、DD: 日、HH: 時、MM: 分、SS: 秒）が生成されます。

3-5-2-1-1-2.RecInfo.json

Record::openRec()による保存先ディレクトリ初期化処理時に設定された情報を元に生成されます。本ファイル内の構成は以下のようになります。

RecInfo.json 内に記録される現在のフォーマットバージョンは Ver3.0.1 となっています。

Table 186. RecInfo.json 内容

項目			サイズ（byte）	内容
Version			–	フォーマットバージョン情報
	major		1	Major Version
	minor		1	Minor Version
	revision		2	Revision
Record			–	保存情報
	rec_frames		4	保存フレーム数
	packing_frames		2	１ファイル内のフレーム数
Device			–	デバイス情報
	hw_kind		4	HW 品種番号 上位 16bit　：センサ品種番号 下位 16bit　：レンズ品種番号
	serial_no		4	機器識別番号
	map_ver		–	カメラ内設定 MAP バージョン
		major	1	Major Version
		minor	1	Minor Version
		revision	2	Revision
	adjust_no		4	調整番号
	firm_ver		–	カメラ Firmware バージョン
		major	1	Major Version
		minor	1	Minor Version
		revision	2	Revision
	ld_wave		2	光源波長 [nm]
	ld_enable		2	各光源の有効情報（灯数情報）
	correct_calib		2	補正キャリブレーション Revision
PostFilt			–	PostFilter 情報
	cam_med_filt		1	保存機能の入力時点でのメディアンフィルタ適用状況
	cam_bil_filt		1	保存機能の入力時点でのバイラテラルフィルタ適用状況
	cam_fly_p_filt		1	保存機能の入力時点でのフライングピクセルフィルタ適用状況
Lens			–	レンズ系変換用パラメータ情報
	sens_w		2	センサ横幅 [pixel]
	sens_h		2	センサ高さ [pixel]
	focal_len		4	焦点距離（固定小数点：整数部 12bit,小数部 20bit）
	thin_w		1	水平間引き数 (1 / thin_w)
	thin_h		1	垂直間引き数 (1 / thin_h)

	crop_x	2	センサ画素からの画像切り出し X 座標位置 [pixel]
	crop_y	2	センサ画素からの画像切り出し Y 座標位置 [pixel]
	dist[]	8×9	歪曲補正用パラメータ [fx, fy, cx, cy, k1, k2, p1, p2, k3] （固定小数点：符号部 1bit,整数部 16bit,小数部 47bit）
	cam_dist	1	保存機能の入力時点での歪曲補正状況
	lens_calib	2	レンズキャリブレーション Revision
Fov		–	FOV 情報
	horz	2	水平視野角 [degree × 100]
	vert	2	垂直視野角 [degree × 100]
Mode		–	動作モード情報
	id	1	動作モード ID
	description	32	動作モード説明
	Images []	–	画像種別情報 （RecImage.raw に含まれる画像種別の配列データ）
	kind	2~6	画像種別（文字列） “Depth”, “IR”, “RAW G1”, “RAW G2”, “RAW G3”, “RAW G4”
	width	2	画像横幅 [pixel]
	height	2	画像高さ [pixel]
	active_start []	2×2	有効画素開始位置 [X 座標位置, Y 座標位置]
	active_w	2	有効画素横幅 [pixel]
	active_h	2	有効画素高さ [pixel]
	bpp	1	１画素のサイズ [byte]
	range	–	測距レンジ情報
	min	2	最至近距離 [mm]
	max	2	最遠端距離 [mm]
	fps	2	受信フレームレート [fps × 100]
	range_calib	2	測距キャリブレーション Revision

3-5-2-1-1-3.RecImage.raw

RecImage_XXXXXXXX.raw には保存された画像データがバイナリ形式（リトルエンディアン）で保存されます。ファイル名の XXXXXXXX は 0 始まりの 10 進数値（00000000～99999999）となり、RecInfo.json の 1 ファイル内のフレーム数(packing_frames)にあたるフレーム数が入る毎に別のファイルに分割され、ファイル名の XXXXXXXX がインクリメントされます。

RecImage_XXXXXXXX.raw の内容は以下のように Record::openRec()による保存先ディレクトリ初期化処理時に設定された情報を元に 1 ファイル内に複数フレーム数(1～packing_frames)分の画像データが含まれる形になります。1 フレーム内には画像種別毎に RecInfo.json 内の Mode – Images の順序および内容に従った画像データが含まれます。

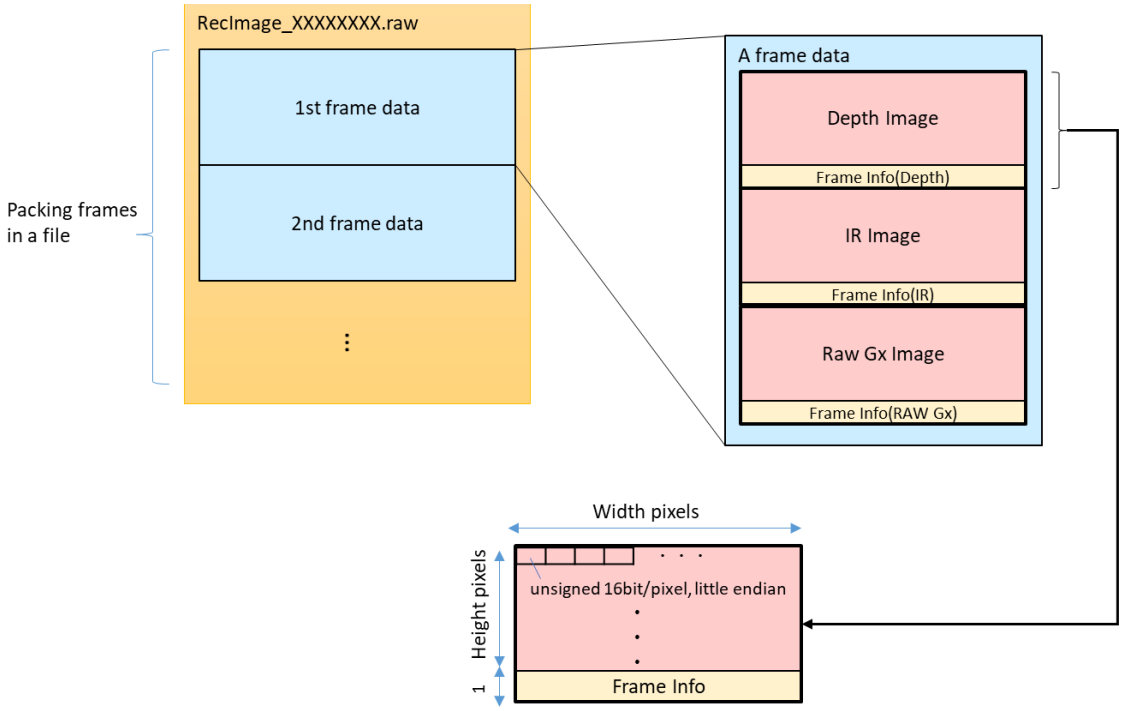


Figure 22. RecImage.raw 内容

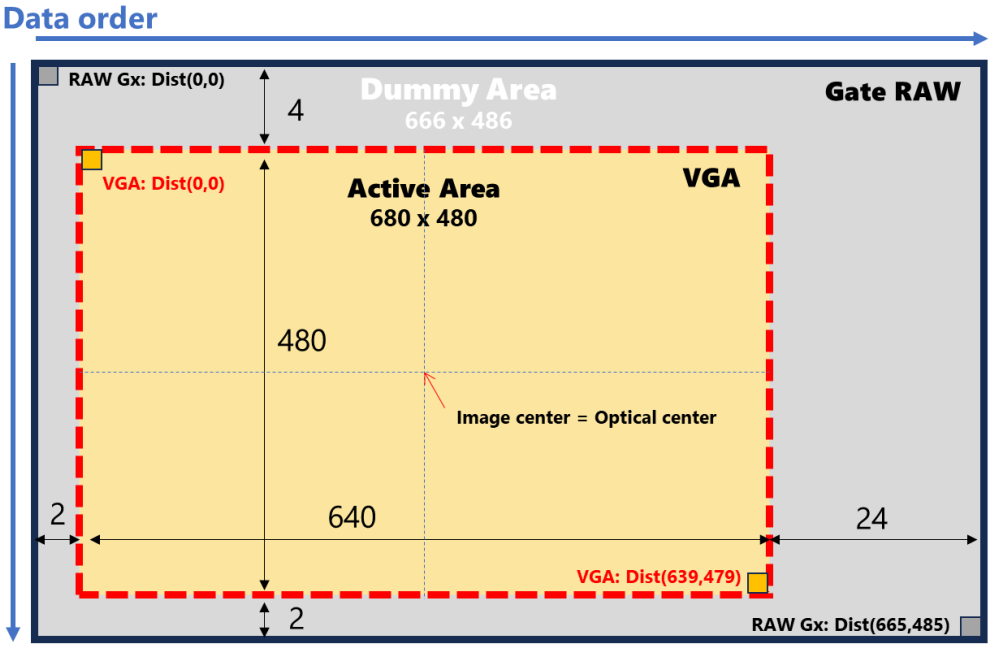


Figure 23. VGA 出力データアレイ情報

Note: IR/Depth データとゲート RAW データ(RAW G1, G2, G3, G4)の出力データサイズが異なります。
Note: IR/Depth の出力データは、Active Area のみ出力されます。
Note: ゲート RAW(RAW G1, G2, G3, G4)データ出力は、デバッグ用の機能です。

3-5-2-1-1-3-1.RecImage.raw – Frame Info

RecImage.raw 内の Frame Info の位置には各画像の 1Line 分の領域に先頭から以下に示す情報が格納されます。

Table 187. FrameInfo 内容

内容		サイズ	
Frame Number		4byte	
Time Stamp	second	8byte	
	nanosecond	8byte	
Frame Error	dropped	1bit	2byte
	crc error	1bit	
	reserved	14bit	
Temperature (Integer: 10bit, Decimal: 6bit)		2byte	
Light times		4byte	
Conversion State	distortion corrected	1bit	1byte
	median filter applied	1bit	
	bilateral filter applied	1bit	
	flying pixel filter applied	1bit	
	reserved	4bit	
reserved		1Line 分の残り（2byte×width – 29byte）	

3-5-3. メソッド一覧

Table 188. Record クラスメソッド

メソッド名前	説明
Record	コンストラクタ
~Record	デストラクタ
openRec	保存先ディレクトリ初期化処理
openRec	
closeRec	保存先ディレクトリ終了処理
recFrame	フレームデータ保存処理

3-5-3-1. 状態遷移

Record クラスの状態遷移は以下となります。

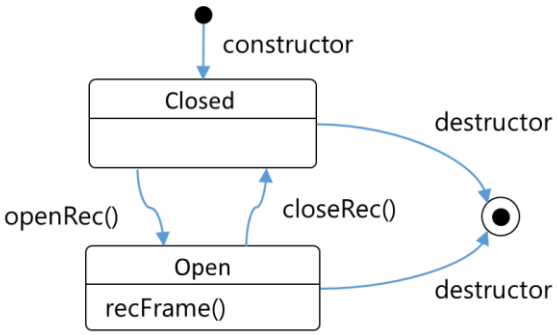


Figure 24. Record クラス状態遷移

3-5-4. 制御シーケンス

Record クラスを用いた保存制御シーケンスを記載します。なお、以下のシーケンスに関しては関数の戻り値判定など異常系の判断は省略しております。

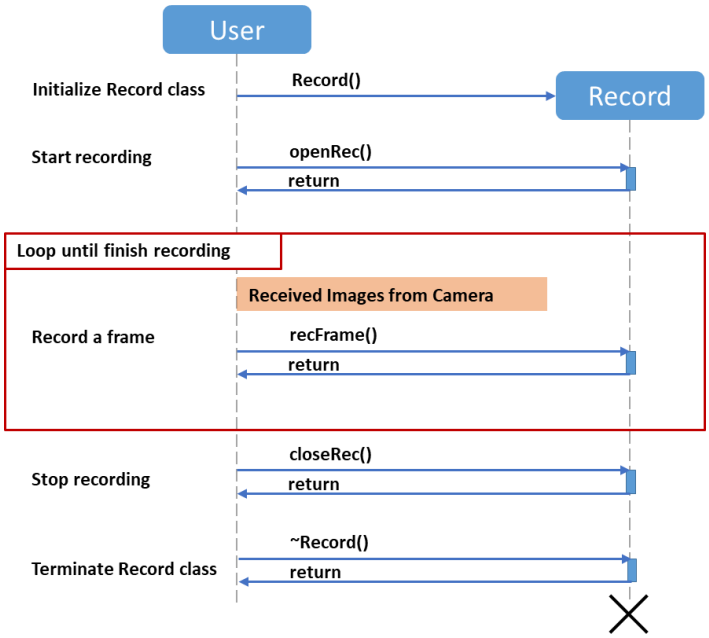


Figure 25. 保存制御シーケンス図

3-5-5. メソッド詳細

3-5-5-1. Record

Table 189. Record::Record メソッド

機能	コンストラクタ			
書式	Record (void)			
説明	・ ファイル保存機能の初期化を行います。			
引数	型	名前	in/out	説明
	なし			
戻り値	なし			
同期/非同期	同期型			

3-5-5-2. ~Record

Table 190. Record::~Record メソッド

機能	デストラクタ			
書式	~Record (void)			
説明	・ ファイル保存機能の終了を行います。 ・ openRec()実行後、closeRec()が実行されていない場合は本メソッド内で closeRec()が実行されます。			
引数	型	名前	in/out	説明
	なし			

戻り値	なし
同期/非同期	同期型

3-5-5-3. openRec (1)

Table 191. Record::openRec メソッド

機能	保存先ディレクトリ初期化処理				
書式	Result openRec (const ConfigParam& config)				
説明	・ 保存先ディレクトリの初期化処理を行います。 ・ 既に保存先ディレクトリが Open されている状態で本メソッドが呼ばれた場合は ERR_BAD_STATE が返ります。 ・ Camera から出力される画像種別が全て保存対象となります。保存対象となる画像種別は recFrame()の時に画像データを入力してください。 ・ 保存ファイルの同期はファイル終端に到達したときに行われますが、引数 config.packing_frames に 10 以上設定した場合は 10 フレーム周期でもファイルの同期が行われます。 ・ 引数 config.save_frames、引数 config.packing_frames は 1 以上の値を設定してください。どちらかが 0 の場合は ERR_BAD_ARG が返ります。 ・ 保存フレーム数（config.save_frames）を保存した後の残容量が 1 GB を下回ることが予測される場合は、ERR_FULL が返ります。				
引数	型		名前	in/out	説明
	const ConfigParam&		config	in	保存設定
戻り値	SUCCESS	0	成功		
	ERR_BAD_ARG	5	引数 config.save_frames、または引数 config.packing_frames が 0		
	ERR_BAD_STATE	6	状態遷移異常		
	ERR_NOT_EXIST	7	指定ディレクトリが存在しない		
	ERR_FULL	10	残容量が不足		
	ERR_SYSTEM	12	システム異常（メモリ確保失敗など）		
同期/非同期	同期型				

3-5-5-4. openRec (2)

Table 192. Record::openRec メソッド

機能	保存先ディレクトリ初期化処理
書式	Result openRec (const RecInfoParam& param)
説明	・ 保存先ディレクトリの初期化処理を行います。 ・ 既に保存先ディレクトリが Open されている状態で本メソッドが呼ばれた場合は ERR_BAD_STATE が返ります。 ・ Camera から出力される画像種別が全て保存対象となります。保存対象となる画像種別は recFrame()の時に画像データを入力してください。

	・ 保存ファイルの同期はファイル終端に到達したときに行われますが、引数 param.packing_frames に 10 以上設定した場合は 10 フレーム周期でもファイルの同期が行われます。 ・ 引数 param.save_frames、引数 param.packing_frames は 1 以上の値を設定してください。どちらかが 0 の場合は ERR_BAD_ARG が返ります。 ・ 保存フレーム数（param.save_frames）を保存した後の残容量が 1 GB を下回ることが予測される場合は、ERR_FULL が返ります。				
引数	型		名前	in/out	説明
	const Record::RecInfoParam&		param	in	保存設定
戻り値	SUCCESS	0	成功		
	ERR_BAD_ARG	5	引数 param.save_frames、または引数 param.packing_frames が 0		
	ERR_BAD_STATE	6	状態遷移異常		
	ERR_NOT_EXIST	7	指定ディレクトリが存在しない		
	ERR_FULL	10	残容量が不足		
	ERR_SYSTEM	12	システム異常（メモリ確保失敗など）		
同期/非同期	同期型				

3-5-5-5. closeRec

Table 193. Record::closeRec メソッド

機能	保存先ディレクトリ終了処理			
書式	Result closeRec (void)			
説明	・ 保存先ディレクトリの終了処理を行います。 ・ openRec()が実行されていない状態で本メソッドが呼ばれた場合は何も処理せずに SUCCESS が返ります。			
引数	型	名前	in/out	説明
	なし			
戻り値	SUCCESS	0	成功	
	ERR_SYSTEM	12	システム異常（メモリ確保失敗など）	
同期/非同期	同期型			

3-5-5-6. recFrame

Table 194. Record::recFrame メソッド

機能	フレームデータ保存処理
書式	Result recFrame (const Frame& frame)
説明	・ １フレームのデータ保存処理を行います。 ・ openRec()で指定した保存フレーム数に到達した場合は、REACH_EOF が返ります。その後は、closeRec()および openRec()が再度実行されるまで保存処理を行うことはできず、ERR_BAD_STATE が返ります。 ・ openRec()の時点で Camera から出力される画像種別の画像データが存在しないもしくはサイズが異なる場合は ERR_EMPTY が返ります。

	・ openRec()が実行されていない状態で本メソッドが呼ばれた場合は ERR_BAD_STATE が返ります。			
引数	型	名前	in/out	説明
	const Frame&	frame	in	保存対象フレームデータ
戻り値	SUCCESS	0	成功	
	REACH_EOF	2	指定フレーム数分の保存完了	
	ERR_BAD_STATE	6	状態遷移異常	
	ERR_EMPTY	9	バッファなどが空	
	ERR_FULL	10	容量などが足りない	
	ERR_SYSTEM	12	システム異常（メモリ確保失敗など）	
同期/非同期	同期型			

3-5-6. Recordクラス用定義

RecordType.h に記載している Record クラスで使用する定義を以下に示します。

3-5-6-1. 構造体定義

3-5-6-1-1. 構造体定義一覧

Table 195. 構造体定義一覧

名称	説明
Record::ConfigParam	保存対象の設定情報
Record::RecInfoParam	保存対象の設定情報（Camera オブジェクトなし）

3-5-6-1-2. 構造体定義詳細

3-5-6-1-2-1.Record::ConfigParam

Table 196. Record::ConfigParam 定義

定義	struct ConfigParam {		
	std::filesystem::path	path;	
	uint32_t	save_frames;	
	uint16_t	packing_frames;	
	Camera*	cam_obj;	
	bool	is_crct_dist;	
	bool	is_filt_med;	
	bool	is_filt_bil;	
	bool	is_filt_fly_p;	
	};		
説明	・ 保存対象の設定情報を示します。		
引数	型	名前	説明
	std::filesystem::path	path	保存先ディレクトリ
	uint32_t	save_frames	保存フレーム数
	uint16_t	packing_frames	１ファイル内に含めるフレーム数
	Camera*	cam_obj	Camera オブジェクト

	bool	is_crct_dist	歪曲補正済みかどうか
	bool	is_filt_med	メディアンフィルタを適用済みか
	bool	is_bil_med	バイラテラルフィルタを適用済みか
	bool	is_filt_fly_p	フライングピクセルフィルタを適用済みか
参照	3-4-5-1. Camera		

3-5-6-1-2-2.Record::RecInfoParam

Table 197. Record::RecInfoParam 定義

定義	struct RecInfoParam {		
		std::filesystem::path	path;
		uint32_t	save_frames;
		uint16_t	packing_frames;
		DeviceInfo	device_info;
		LensInfo	lens_info;
		CamFov	fov;
		ModeList	mode_list;
		ImageFormats	image_formats;
		uint8_t	mode;
		bool	is_crct_dist;
		bool	is_filt_med;
		bool	is_filt_bil;
		bool	is_filt_fly_p;
	};		
説明	・ 保存対象の設定情報を示します。		
引数	型	名前	説明
	std::filesystem::path	path	保存先ディレクトリ
	uint32_t	save_frames	保存フレーム数
	uint16_t	packing_frames	１ファイル内に含めるフレーム数
	DeviceInfo	device_info	カメラデバイスの機器情報
	LensInfo	lens_info	Lens 系変換処理情報
	CamFov	fov	カメラデバイスの FOV 情報
	ModeList	mode_list	動作モードのリスト情報
	ImageFormats	image_formats	全画像種別のフォーマット情報
	uint8_t	mode	動作モード ID
	bool	is_crct_dist	歪曲補正済みかどうか
	bool	is_filt_med	メディアンフィルタを適用済みか
	bool	is_filt_bil	バイラテラルフィルタを適用済みか
	bool	is_filt_fly_p	フライングピクセルフィルタを適用済みか
参照	3-4-8-2-2-4. DeviceInfo, 3-4-8-2-2-6. LensInfo, 3-4-8-2-2-7. CamFov, 3-4-8-3-2-1. ModeList, 3-2-4-2-1. ImageFormats		

3-6. LensConv クラス

3-6-1. 概要

Table 198. LensConv クラス概要

提供ヘッダファイル	LensConv.h	LensConv クラス定義
	LensConvType.h	LensConv クラス型定義
所属名前空間	krm	
説明	カメラデバイスの出力画像に対してレンズに関わる変換処理を行う C++クラス	
スレッドセーフ	LensConv クラスはスレッドセーフではないため、利用するユーザプログラム側で必要に応じて排他処理を行ってください。	

3-6-2. 提供機能

3-6-2-1. 機能概要

LensConv クラスの提供する機能概要を以下に示します。

Table 199. LensConv クラス機能概要

提供機能	機能概要
歪曲補正機能	レンズ曲線によって歪みが生じている画像を補正する機能です。 入力として受け取った Depth 画像または IR 画像に対して歪曲補正を行い、歪曲補正後の画像を出力します。 歪曲補正は Camera クラスが持つ Lens 系変換処理用パラメータを使って変換処理を行います。
点群変換機能	歪曲補正後の Depth 画像を入力として受け取り、３次元空間における点群データを出力する機能です。 点群変換は Camera クラスが持つ Lens 系変換処理用パラメータを使って変換処理を行います。

3-6-2-1-1. 歪曲補正機能

Camera クラスからの出力される画像はレンズの歪曲収差による歪みがある状態で出力されます。歪曲補正機能では以下のように歪曲収差による歪みを補正します。

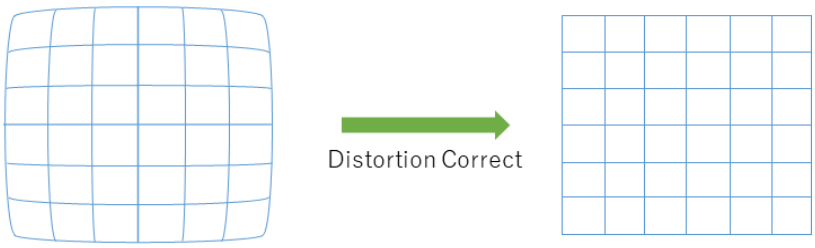


Figure 26. 歪曲補正

3-6-2-1-2. 点群変換機能

歪曲補正後の Depth 画像の各画素に対して、カメラ座標系もしくは世界座標系の点群に変換する機能です。

3-6-2-1-2-1.カメラ座標

カメラ座標の点群変換では Depth 画像の値 D をレンズ中心の 3 次元座標の点 P(Px, Py, Pz)に変換します。

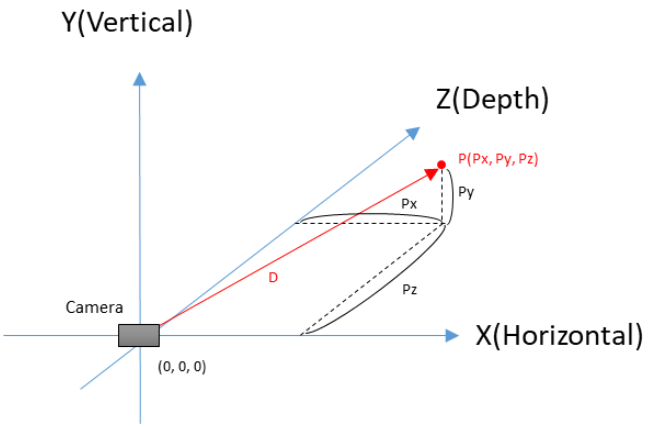


Figure 27. カメラ座標点群変換

3-6-2-1-2-1-1.世界座標

世界座標の点群変換では Depth 画像の値 D をカメラ座標の中心 O(Ox, Oy, Oz)とは異なる位置を原点とした 3 次元座標の点 P'(Px', Py', Pz')に変換します。

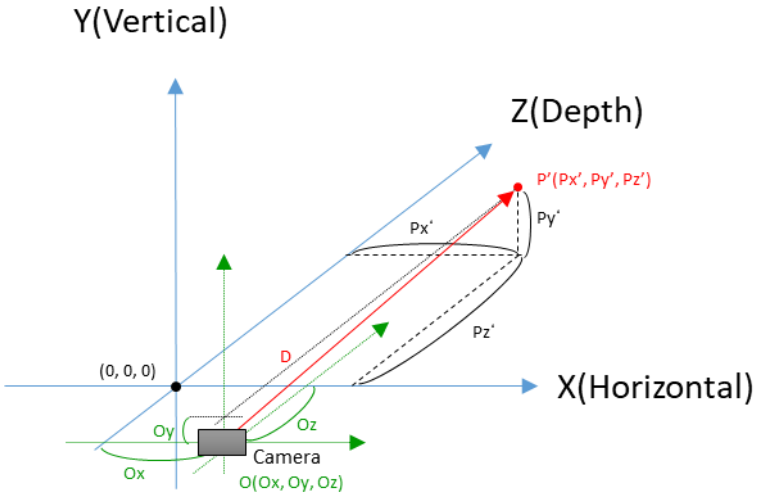


Figure 28. 世界座標点群変換

また、世界座標点群の変換時はカメラデバイスの設置条件などに沿って、点群の回転をすることができます。

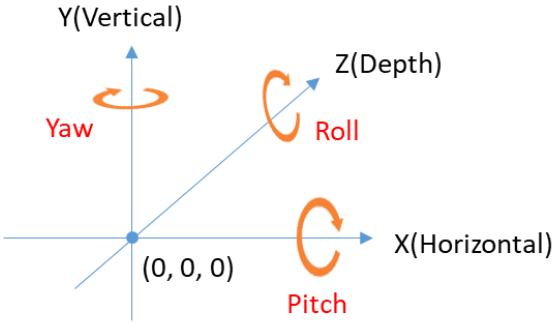


Figure 29. 点群回転方向

3-6-3. メソッド一覧

Table 200. LensConv クラスメソッド

メソッド名前	説明
LensConv	コンストラクタ
~LensConv	デストラクタ
setLensPrm	Lens 系変換処理用パラメータ設定
setFormat	画像フォーマット情報設定
setPosOrgRotation	世界座標原点、点群回転情報設定
correctDist	歪曲補正実行
convPcdCamera	カメラ座標点群変換実行
convPcdWorld	世界座標点群変換実行

3-6-3-1. 状態遷移

LensConv クラスの状態遷移は以下となります。

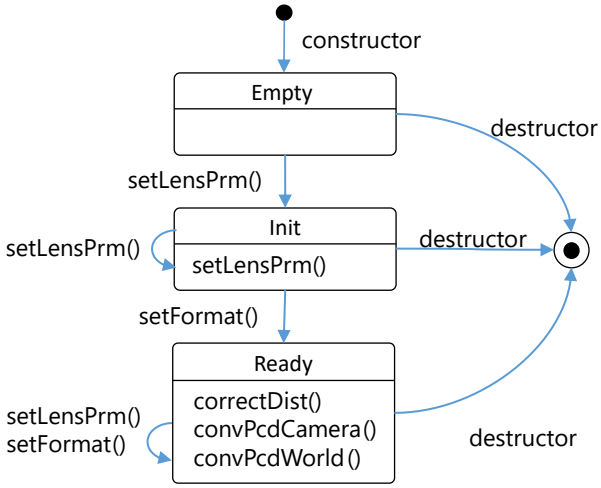


Figure 30. LensConv クラス状態遷移

3-6-4. 制御シーケンス

LensConv クラスを用いた画像変換シーケンスを記載します。なお、以下のシーケンスに関しては関数の戻り値判定など異常系の判断は省略しております。

Lens 系変換処理用パラメータの設定（setLensPrm(), setFormat()）は画像受信までに行う必要があります。

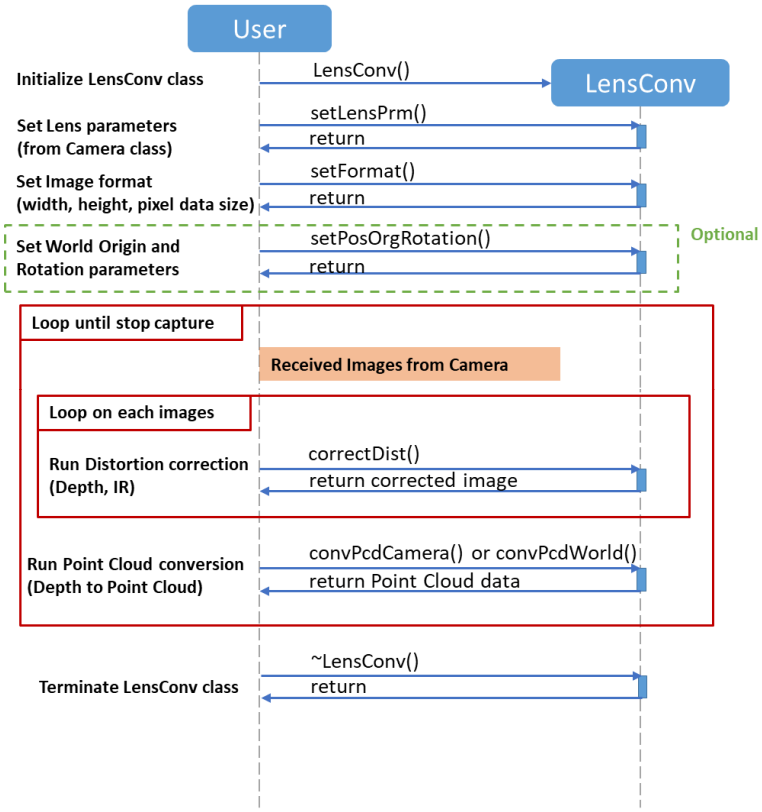


Figure 31. 画像変換シーケンス図

3-6-5. メソッド詳細

3-6-5-1. LensConv

Table 201. LensConv::LensConv メソッド

機能	コンストラクタ			
書式	LensConv (void)			
説明	・ Lens 系変換処理の初期化を行います。			
引数	型	名前	in/out	説明
	なし			
戻り値	なし			
同期/非同期	同期型			

3-6-5-2. ~LensConv

Table 202. LensConv::~LensConv メソッド

機能	デストラクタ			
書式	~LensConv (void)			
説明	・ Lens 系変換処理の終了を行います。			
引数	型	名前	in/out	説明
	なし			

戻り値	なし
同期/非同期	同期型

3-6-5-3. setLensPrm

Table 203. LensConv::setLensPrm メソッド

機能	Lens 系変換処理用パラメータ設定			
書式	Result setLensPrm (const LensInfo& lens_info)			
説明	・ 歪曲補正機能、点群変換機能で使用するパラメータ情報を設定します。			
引数	型	名前	in/out	説明
	const LensInfo&	lens_info	in	Lens 系変換処理用パラメータ
戻り値	SUCCESS	0	成功	
	ERR_BAD_ARG	5	その他、不正引数	
同期/非同期	同期型			

3-6-5-4. setFormat

Table 204. LensConv::setFormat メソッド

機能	画像フォーマット情報設定			
書式	Result setFormat (const ImageFormat& format)			
説明	・ LensConv クラスで扱う画像のフォーマット情報を設定します。 ・ 点群変換機能を使用する場合は Depth 画像の画像フォーマットを設定します。 ・ 歪曲補正機能を複数種類の画像に対して行う場合は同じ画像フォーマット（画像幅、画像高さ）である必要があります。 ・ setLensPrm()で Lens 系変換処理用パラメータが設定されていない状態で本メソッドが呼ばれた場合は ERR_BAD_STATE が返ります。			
引数	型	名前	in/out	説明
	const ImageFormat&	format	in	画像フォーマット
戻り値	SUCCESS	0	成功	
	ERR_BAD_ARG	5	画像フォーマット情報が空	
	ERR_BAD_STATE	6	状態遷移異常	
同期/非同期	同期型			

3-6-5-5. setPosOrgRotation

Table 205. LensConv:: setPosOrgRotation メソッド

機能	世界座標原点、点群回転情報設定
書式	void setPosOrgRotation (const PosOrgRotation& pos_rot)
説明	・ 世界座標系の点群変換機能で変換する原点位置、点群回転情報を設定します。

	・ 本メソッドが呼ばれていない場合、原点位置はカメラ座標系と同じ座標、点群の回転角度は全て 0 となります。			
引数	型	名前	in/out	説明
	const PosOrgRotation&	pos_rot	in	原点位置、点群回転情報
戻り値	なし			
同期/非同期	同期型			

3-6-5-6. correctDist

Table 206. LensConv::correctDist メソッド

機能	歪曲補正実行			
書式	Result correctDist (const ImageData& org_img, bool is_depth, ImageData& aft_img)			
説明	・ 入力画像に対する歪曲補正処理を行います。 ・ 引数 aft_img には引数 org_img と同じ画素数の領域を確保してください。 ・ 引数 is_depth には引数 org_img が Depth 画像の場合は true、IR 画像の場合は false を設定してください。 ・ setLensPrm(), setFormat()が実行されていない状態で本メソッドが呼ばれた場合は ERR_BAD_STATE が返ります。			
引数	型	名前	in/out	説明
	const ImageData&	org_img	in	入力画像
	bool	is_depth	in	入力画像が Depth 画像か？ true : Depth 画像 false : IR 画像
	ImageData&	aft_img	out	歪曲補正後画像
戻り値	SUCCESS	0	成功	
	ERR_BAD_ARG	5	引数 org_img、引数 aft_img に setFormat()と異なる画像フォーマットが設定された	
	ERR_BAD_STATE	6	状態遷移異常	
同期/非同期	同期型			

3-6-5-7. convPcdCamera

Table 207. LensConv:: convPcdCamera メソッド

機能	カメラ座標点群変換実行
書式	Result convPcdCamera (const ImageData& depth, PcdData& pcd)
説明	・ カメラ座標系への点群変換を行います。 ・ 引数 pcd には引数 depth と同じ画素数の領域を用意してください。 ・ setLensPrm(), setFormat()が実行されていない状態で本メソッドが呼ばれた場合は ERR_BAD_STATE が返ります。

	型	名前	in/out	説明
引数	const ImageData&	depth	in	Depth 画像
	PcdData&	pcd	out	点群データ

戻り値	SUCCESS	0	成功
	ERR_BAD_ARG	5	引数 depth、引数 pcd に setFormat()と異なる画像フォーマットが設定された
	ERR_BAD_STATE	6	状態遷移異常
同期/非同期	同期型		

3-6-5-8. convPcdWorld

Table 208. LensConv:: convPcdWorld メソッド

機能	世界座標点群変換実行			
書式	Result convPcdWorld (const ImageData& depth, PcdData& pcd)			
説明	・ 世界座標系への点群変換を行います。 ・ 引数 pcd には引数 depth と同じ画素数の領域を用意してください。 ・ setLensPrm(), setFormat()が実行されていない状態で本メソッドが呼ばれた場合は ERR_BAD_STATE が返ります。			
引数	型	名前	in/out	説明
	const ImageData&	depth	in	Depth 画像
	PcdData&	pcd	out	点群データ
戻り値	SUCCESS	0	成功	
	ERR_BAD_ARG	5	引数 depth、引数 pcd に setFormat()と異なる画像フォーマットが設定された	
	ERR_BAD_STATE	6	状態遷移異常	
同期/非同期	同期型			

3-6-6. LensConvクラス用定義

LensConvType.h に記載している LensConv クラスで使用する定義を以下に示します。

3-6-6-1. 構造体定義

3-6-6-1-1. 構造体定義一覧

Table 209. 構造体定義一覧

名称	説明
PosOrgRotation	原点位置、回転情報

3-6-6-1-2. 構造体定義詳細

3-6-6-1-2-1.PosOrgRotation

Table 210. PosOrgRotation 定義

<table>
<tr><td>定義</td><td colspan="3">struct PosOrgRotation {
struct {
int16_t x;
int16_t y;
int16_t z;
} offset;
struct {
float pitch;
float yaw;
float roll;
} rotation;
};</td></tr>
<tr><td>説明</td><td colspan="3">・ 世界座標系への点群変換における原点位置、回転情報を示します。</td></tr>
<tr><td rowspan="7">引数</td><td>型</td><td>名前</td><td>説明</td></tr>
<tr><td>int16_t</td><td>x</td><td>原点位置：X 軸方向オフセット [mm]</td></tr>
<tr><td>int16_t</td><td>y</td><td>原点位置：Y 軸方向オフセット [mm]</td></tr>
<tr><td>int16_t</td><td>z</td><td>原点位置：Z 軸方向オフセット [mm]</td></tr>
<tr><td>float</td><td>pitch</td><td>Pitch 回転角度 [degree]</td></tr>
<tr><td>float</td><td>yaw</td><td>Yaw 回転角度 [degree]</td></tr>
<tr><td>float</td><td>roll</td><td>Roll 回転角度 [degree]</td></tr>
<tr><td>参照</td><td colspan="3">3-6-5-5. setPosOrgRotation</td></tr>
</table>

4. 使用上の制約事項

4-1. データ出力に関する制約事項

C11U カメラは、撮像する間隔が空いた場合にカメラから暗電流 *Note1* によるノイズ成分を多く含んだフレームデータが出力されることがあります。暗電流の影響があるフレームデータを除去するため、フレーム出力開始直後の 2 フレームのデータは Camera クラス内部で破棄します。ただし、外部トリガ種別が Secondary(Slave, EXT_TRG_SLAVE)かつ外部トリガの間隔が空いた場合は上記に該当しないため、暗電流の影響があるフレームが出現する可能性があります。

ホストコンピュータの OS や動作環境によっては、フレームの取得が安定しない可能性があるため、フレームデータ出力開始後に取得される最初のフレームが 3 フレーム目ではない場合があります。

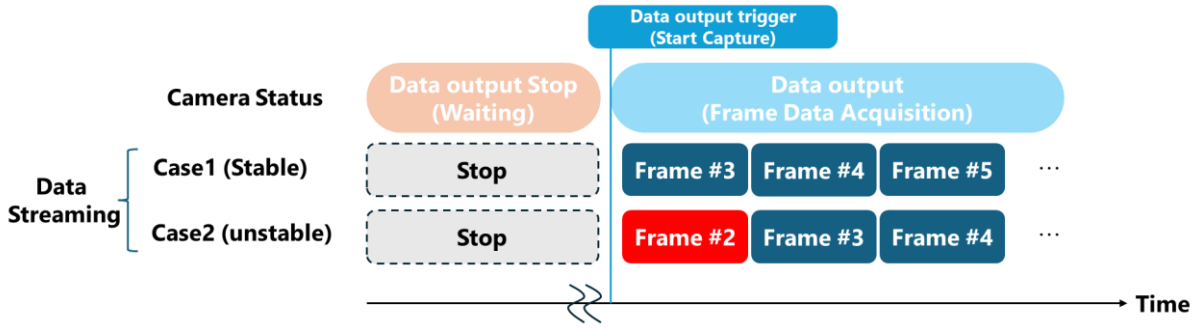


Figure 32. データ出力に関する制約事項

Note[1]:

暗電流は、受光素子（画素）が光を受けていない時でも発生するノイズの一種です。この暗電流ノイズは、本来の距離情報を示す光信号に混入し、距離測定の精度を低下させます。特に、画素の駆動が停止している時間が長い場合、再開後のフレームデータに暗電流ノイズの影響が顕著に現れることがあります。

4-2. 外部トリガ種別 Secondary(Slave, EXT_TRG_SLAVE)時の制約事項

外部トリガ種別が Secondary(Slave, EXT_TRG_SLAVE)の時、画像受信がないまま長時間受信待ち状態となると、ホスト PC に負荷がかかる恐れがあります。長時間の高負荷動作は動作不良や故障の原因となるため、長時間受信待ちになる場合は受信待ちを解除することを推奨します。

5. 使用条件・免責事項

TOPPAN ホールディングス株式会社及び TOPPAN 株式会社(以下、当社)製品の使用条件につきましては、「C11U 取扱説明書」や「TOPPAN ToF *senSPure*® SDK ライブラリ API リファレンスマニュアル」、その他関連するドキュメントをご確認ください。

- 本書の一部あるいは全部を無断で複写・複製・転載することは、固くお断りします。
- 本書の内容は、予告無く変更する場合があります。
- 当社は、正確な情報を提供するためにあらゆる措置を取っていますが、誤りや不作為について責任を負うものではありません。また、本書に記載されている情報の使用に起因するいかなる損害に対しても責任を負うものではありません。
- 当社は、本製品の使用に関連するデータ損失、機会損失、利益損失、その他付随的、間接的、あるいは二次的損害をはじめとするあらゆる損害については一切責任を負いません。
- 本書および関連文書内に記載されている商品名および会社名などの固有名詞は、それぞれの会社または個人に帰属します。本書では TM(™)、R(®)マークは省略している場合があります。これらの名称は、本書内での識別および説明の目的のみで使用しています。当社はこれらのいかなる権利を侵害する意図はありません。

改定履歴

Date	Revision	Comment
2024/07/18	1.00	初版
2024/09/20	1.01	・見出し、目次レイアウト修正 ・軽微な修正 (誤字脱字等)
2025/03/18	1.10	・C11U ES Version リリース
2025/06/23	1.11	・1.5 動作環境 更新 (Windows 11 対応追記。JetPack バージョン記載) ・3.3.6.3.11 PlFw::getEvent 更新 (slave 動作時の待機時間追加) ・4. 使用上の制約事項　追記 4-1. データ出力に関する制約事項 4-2. 外部トリガ種別 Secondary (Slave, EXT_TRG_SLAVE)時の制約事項 ・軽微な修正
2025/09/09	1.12	・1.5 動作環境 更新 (Ubuntu 22.04LTS, JetPack6.0 記載) ・軽微な修正

TOPPAN

TOPPAN ホールディングス株式会社 TOF 事業推進センター
TOF Business Development Center, TOPPAN Holdings Inc.

TOPPAN 株式会社 エレクトロニクス事業本部
Electronics Division, TOPPAN Inc.

Location
(日本語) 〒108-8539 東京都港区芝浦 3-19-26 トッパン芝浦ビル
(English) 3-19-26, Shibaura, Minato-ku, Tokyo, 108-8539
E-mail electronics@toppan.co.jp
Website https://www.toppan.com/ja/electronics/device/tof/ (TOPPAN Inc.)

ToF カメラ製品サポート窓口
E-mail btop_support@toppan.co.jp