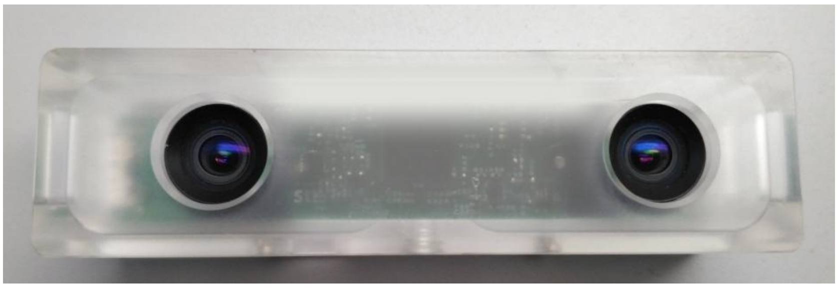
TaraXL

Accelerated SDK API Document



e-con Systems

Your Product Development Partner

Version 3.8

e-con Systems

11/8/2019

Disclaimer

e-con Systems reserves the right to edit/modify this document without any prior intimation of whatsoever.

Contents

Introduction

TaraXL is the software development kit that can be used with See3CAM_StereoA and STEEReoCAM to perceive depth, point cloud, disparity images, IMU data, and so on.

TaraXL can be used by customers who want to develop their Stereo camera algorithms and to integrate Stereo camera in their product design. It is provided with accelerated SDK for NVIDIA® Jetson™ TX2/Xavier™/Nano™ and Ubuntu 16.04 (x64) which is ideal for depth application product designs.

See3CAM_StereoA is a UVC compliant 3D stereo camera based on MT9V024 stereo sensor from ON Semiconductor® which supports WVGA ((2*752) x 480) at 60 fps over USB 3.0 in uncompressed format. This Stereo camera provides two synchronized sensor frame data interleaved side by side to the host machine over USB 3.0 interface.

STEEReoCAM is a 3D Stereo camera based on OV2311 sensor from OmniVision which supports ((2*1600) x 1300) at 30 fps with a baseline of 10 cm. This Stereo camera provides two synchronized sensor frame data side by side to the host machine over MIPI interface.

e-con Systems 3D Stereo camera is ideal for applications such as Depth Sensing, Disparity Map, Point Cloud, Machine Vision, Drones, 3D Video Recording, Surgical Robotics, and so on.

This document describes the accelerated SDK APIs of TaraXL SDK.

Prerequisites

For NVIDIA® Jetson™ TX2/Xavier™/ Nano™ device, the prerequisites are L4T - 32.2 and CUDA 10.0.

Overview

TaraXL accelerated APIs are majorly grouped into five C++ classes as follows:

- **TaraXL class** - enumerates connected TaraXL devices.
- **TaraXLCam class** - used to grab rectified/unrectified frames, get and set various camera parameters such as exposure, brightness, resolution, and so on.
- **TaraXLDepth class** - used to obtain disparity and depth frames.
- **TaraXLPointCloud** - used to obtain pointclouds in different qualities.
- **TaraXLPoseTracking** - used to obtain IMU related data.
- **TARAXL_STATUS_CODE** - has a list of status codes that are returned by this SDK to the application.

TaraXL Class

The TaraXL class belongs to the TaraXL SDK namespace and is the first class that needs to be constructed to get access to TaraXL devices using TaraXLCam objects. This section details the methods of TaraXL class.

The supported TaraXL class methods are as follows:

- TaraXL()
- TARAXL_STATUS_CODE enumerateDevices(TaraXLCamList &taraXLCamList)

TaraXL()

This is a parameterless constructor and is the first constructor that needs to be called when invoking this SDK.

Parameters	Description	Return Values
None	-	-

TARAXL_STATUS_CODE enumerateDevices(TaraXLCamList &taraXLCamList)

This method enumerates the connected devices and returns them in a container class TaraXLCamList.

Parameters	Description
`TaraXLCamList`	A list of TaraXLCam device objects will be returned in this parameter. If no devices are connected to the host, this list will be empty.

TaraXLCam Class

The TaraXLCam class represents a TaraXL device that is connected to the host. This class must not be invoked by an application directly and must be obtained using TaraXL class enumerateDevices method. Using this class, an application can get and set various device settings. The application can also grab rectified frames from the device using this class. The details of various methods of this class are described below.

The supported TaraXLCam class methods are as follows:

- TARAXL_STATUS_CODE connect()
- TARAXL_STATUS_CODE disconnect()
- TARAXL_STATUS_CODE grabFrame(cv::Mat &leftImage, cv::Mat &rightImage)
- TARAXL_STATUS_CODE getResolutionList(ResolutionList &resolutionList)
- TARAXL_STATUS_CODE getResolution(Resolution &resolution)
- TARAXL_STATUS_CODE setResolution(Resolution &resolution)
- TARAXL_STATUS_CODE setExposure(int exposureVal)
- TARAXL_STATUS_CODE getExposure(int &exposureVal)
- TARAXL_STATUS_CODE enableAutoExposure()
- TARAXL_STATUS_CODE setBrightness(int brightnessVal)
- TARAXL_STATUS_CODE getBrightness(int &brightnessVal)
- TARAXL_STATUS_CODE getQMatrix(cv::Mat &qMat)
- TARAXL_STATUS_CODE getFriendlyName(std::string &name)
- TARAXL_STATUS_CODE setGain(int gainVal)
- TARAXL_STATUS_CODE getGain(int &gainVal)
- TARAXL_STATUS_CODE getCameraUniqueId(std::string &name)
- TARAXL_STATUS_CODE getSDKVersion(std::string &version)
- TARAXL_STATUS_CODE grabUnrectifiedFrame(cv::Mat &unrectifiedLeftImage, cv::Mat &unrectifiedRightImage)
- TARAXL_STATUS_CODE getCalibrationParameters(cv::Mat &rotationMatrix, cv::Mat &translationMatrix, CalibrationParams &left, CalibrationParams &right, CalibrationParams &leftRectified, CalibrationParams &rightRectified)

TARAXL_STATUS_CODE connect()

This function is used to open a connection to a TaraXL device to grab frames and change settings.

Parameters	Description
None	-

TARAXL_STATUS_CODE disconnect()

This function is used to close a connection to a TaraXL device. After calling this method, the behaviour of calling any other methods is undefined.

Parameters	Description
None	-

TARAXL_STATUS_CODE grabFrame(cv::Mat &leftImage, cv::Mat &rightImage)

This function returns left and right rectified images of the TaraXL device as OpenCV mats.

Parameters	Description
`leftImage`	Allocate and pass a reference to OpenCV mat object to get left rectified frame data as return value.
`rightImage`	Allocate and pass a reference to OpenCV mat object to get right rectified frame data as return value.

TARAXL_STATUS_CODE getResolutionList(ResolutionList &resolutionList)

This function returns a list of supported resolutions for the TaraXL device.

Parameters	Description
`ResolutionList`	A reference to an empty ResolutionList object needs to be passed, and the function fills the list of supported resolutions. Each resolution object has width and height integer members.

TARAXL_STATUS_CODE getResolution(Resolution &resolution)

This function gets the current resolution of the device.

Parameters	Description
`Resolution`	Resolution to be obtained from the camera.

TARAXL_STATUS_CODE setResolution(Resolution &resolution)

This function changes the current resolution of the device if the passed in parameter is a valid supported resolution.

Parameters	Description
`Resolution`	Resolution to be set in the camera. This object can be any one in the list of objects returned by the getResolutionList method.

TARAXL_STATUS_CODE setExposure(int exposureVal)

This function sets the manual exposure for the selected TaraXL device.

Parameters	Description
`exposureVal`	Exposure value to be set in the camera. The value of the exposure ranges from 10 to 1000000 for See3CAM_StereoA and 1 to 7500 for STEEReoCAM.

TARAXL_STATUS_CODE getExposure(int &exposureVal)

This function gets the manual exposure for the selected TaraXL device. The value of the manual exposure ranges from 10 to 1000000 for See3CAM_StereoA and 1 to 7500 for STEEReoCAM. When the camera is in auto exposure, the return value will be 1.

Parameters	Description
`exposureVal`	A reference to an integer is passed and the subsequent exposure value is returned.

TARAXL_STATUS_CODE enableAutoExposure()

This function will enable auto exposure algorithm for the selected TaraXL device.

Parameters	Description
None	-

TARAXL_STATUS_CODE setBrightness(int brightnessVal)

This function sets the brightness for selected TaraXL device. For STEEReoCAM, this setting will work only when auto exposure enabled.

Parameters	Description
`brightnessVal`	Brightness value to be set in the camera. The value ranges from 1 to 7 (for See3Cam_StereoA) and 1 to 10 (for STEEReoCAM).

TARAXL_STATUS_CODE getBrightness(int &brightnessVal)

This function returns the brightness value for the TaraXL device. For STEEReoCAM, this setting will work only when auto exposure enabled.

Parameters	Description
`brightnessVal`	A reference to an integer is passed in which the brightness value is returned.

TARAXL_STATUS_CODE getQMatrix(cv::Mat &qMat)

This function returns the Q matrix for the TaraXL device. It will be an OpenCV mat.

Parameters	Description
`qMat`	4 x 4 perspective transformation matrix.

TARAXL_STATUS_CODE getFriendlyName(std::string &name)

This function returns the friendly name for the TaraXL device.

Parameters	Description
`name`	Name of the selected camera.

TARAXL_STATUS_CODE setGain(int gainVal)

This function sets the gain value for the TaraXL device. This setting is currently available only for STEEReoCAM.

Parameters	Description
`gainVal`	Gain value that is to be set to the camera. The value ranges from 1 to 240.

TARAXL_STATUS_CODE getGain(int &gainVal)

This function returns the gain value for the TaraXL device. This setting is currently available only for STEEReoCAM.

Parameters	Description
`gainVal`	A reference to the integer pointer is passed in which the gain value is returned.

TARAXL_STATUS_CODE getCameraUniqueId(std::string &uniqueId)

This function returns the unique ID for the TaraXL device.

Parameters	Description
`uniqueId`	A reference to a string is passed in which the unique ID of the camera is returned. It returns the serial number for See3CAM_StereoA and the connected port name for STEEReoCAM.

TARAXL_STATUS_CODE getSDKVersion(std::string &version)

This function returns the version of the TaraXL SDK.

Parameters	Description
`version`	Get the version of current SDK being in usage.

TARAXL_STATUS_CODE grabUnrectifiedFrame(cv::Mat &unrectifiedLeftImage, cv::Mat &unrectifiedRightImage)

This function returns left and right unrectified images of the TaraXL device as OpenCV mats.

Parameters	Description
`unrectifiedLeftImage`	Allocate and pass a reference to OpenCV mat object to get left unrectified frame data as return value.
`unrectifiedRightImage`	Allocate and pass a reference to OpenCV mat object to get right unrectified frame data as return value.

TARAXL_STATUS_CODE getCalibrationParameters(cv::Mat &rotationMatrix, cv::Mat &translationMatrix, CalibrationParams &left, CalibrationParams &right, CalibrationParams &leftRectified, CalibrationParams &rightRectified)

This function returns calibration parameters of the connected camera.

Parameters	Description
`&rotationMatrix`	Allocate and pass a reference to OpenCV mat object to get rotation matrix as return value.
`&translationMatrix`	Allocate and pass a reference to OpenCV mat object to get translation matrix as return value.
`&left`	Returns the intrinsic calibration parameters of the left camera.
`&right`	Returns the intrinsic calibration parameters of the right camera.
`&leftRectified`	Calibration parameters of rectified left frame.
`&rightRectified`	Calibration parameters of rectified right frame.

TaraXLDepth Class

The details regarding the supported TaraXLDepth class are explained below.

The supported TaraXLDepth class method are as follows:

- TaraXLDepth(TaraXLCam &camera)
- TARAXL_STATUS_CODE setAccuracy(ACCURACY_MODE mode)
- TARAXL_STATUS_CODE getMinDisparity(int &minDisparity)
- TARAXL_STATUS_CODE getMaxDisparity(int &maxDisparity)
- TARAXL_STATUS_CODE setDepthRange(TARAXL_DEPTH_RANGE depthRange)
- TARAXL_STATUS_CODE getMap(cv::Mat &leftImage, cv::Mat &rightImage, cv::Mat &disparityMap, bool fillDisparityMap, cv::Mat &depthMap, bool fillDepthMap)

TaraXLDepth(TaraXLCam &camera)

This function initialize this constructor with a valid TaraXLCam device object obtained from TaraXL::enumerateDevices method. An application must first call TaraXLCam::connect() method before invoking this constructor.

Parameters	Description
`camera`	The context of the camera device is returned in TaraXL.enumerateDevices() method.

TARAXL_STATUS_CODE setAccuracy(ACCURACY_MODE mode)

This function sets the accuracy modes of depth maps to high or low or ultra.

Parameters	Description
`mode`	The supported accuracy modes are: HIGH - In this mode, the disparity and depth map accuracy are high with reduced fps. Minimum range starts from 95cm and, maximum range up to 8m. LOW - In this mode, the disparity and depth map fps are high with reduced accuracy. Minimum range starts from 145cm and maximum range up to 8m. ULTRA - In this mode, the disparity and depth map accuracy is ultra with reduced fps. Minimum range starts from 115cm and maximum range up to 8m.

TARAXL_STATUS_CODE getMinDisparity(int &minDisparity)

This function gets the minimum disparity value of the various depth modes.

Parameters	Description
`minDisparity`	The value of minimum disparity value.

TARAXL_STATUS_CODE getMaxDisparity(int &maxDisparity)

This function gets the maximum disparity value of the various depth modes.

Parameters	Description
`maxDisparity`	The value of maximum disparity value.

TARAXL_STATUS_CODE setDepthRange(TARAXL_DEPTH_RANGE depthRange)

This function sets the depth range of the camera. This option is available only for ULTRA mode currently.

Parameters	Description
`depthRange`	The supported depth ranges are: TARAXL_DEFAULT_RANGE - In this mode, the depth range starts from 115-800cms for STEEReoCAM and 35-300cms for See3CAM_StereoA. TARAXL_NEAR_RANGE - In this mode, the depth range starts from 80-240cms for STEEReoCAM and 30-150cms for See3CAM_StereoA. TARAXL_VERY_NEAR_RANGE - In this mode, the depth range starts from 60-115cms for STEEReoCAM and 20-45cms for See3CAM_StereoA.

TARAXL_STATUS_CODE getMap(cv::Mat &leftImage, cv::Mat &rightImage, cv::Mat &disparityMap, bool fillDisparityMap, cv::Mat &depthMap, bool fillDepthMap, TARAXL_FILTER_TYPE filter=TARAXL_NO_FILTER)

This function fills the depth and disparity maps based on flags (fillDisparityMap, fillDepthMap) set while invoking.

Parameters	Description
`leftImage`	Allocate and pass a reference to OpenCV mat object to get left rectified frame data as return value.
`rightImage`	Allocate and pass a reference to OpenCV mat object to get right rectified frame data as return value.

`DisparityMap`	Allocate and pass a reference to OpenCV mat object to get disparity frame data as return value. fillDisparityMap parameter must be set to True.
`fillDisparityMap`	True - disparityMap OpenCV mat will be filled. False - disparityMap OpenCV mat will not be filled. The default value is False.
`DepthMap`	Allocate and pass a reference to OpenCV mat object to get depth frame data as return value. fillDepthMap parameter must be set to True.
`FillDepthMap`	True - depthMap OpenCV mat will be filled. False - depthMap OpenCV mat will not be filled. The default value is False.
`TARAXL_FILTER_TYPE`	Applies filter to the disparity map. This is an optional parameter.

TaraXLPointCloud Class

The details regarding the supported TaraXLPointCloud class are explained below.

The supported TaraXLPointCloud class method are as follows:

- TaraXLPointcloud(TaraXLCam &camera)
- TARAXL_STATUS_CODE setPointcloudQuality(TARAXL_POINTCLOUD_QUALITY pointcloudQuality)
- TARAXL_STATUS_CODE savePoints(TARAXL_POINTCLOUD_FORMAT pointcloudFormat, std::string filename)
- TARAXL_STATUS_CODE getPoints(Points::Ptr currentCloud)

TaraXLPointcloud(TaraXLCam &camera)

This function initialize this constructor with a valid TaraXLCam device object obtained from TaraXL::enumerateDevices method. An application must first call TaraXLCam::connect() method before invoking this constructor.

Parameters	Description
`camera`	The context of the camera device is returned in TaraXL::enumerateDevices method.

TARAXL_STATUS_CODE setPointcloudQuality(TARAXL_POINTCLOUD_QUALITY pointcloudQuality)

This function sets the quality mode of the Point Cloud.

Parameters	Description
`pointcloudQuality`	The supported quality modes are: HIGHEST - In this mode, the quality is best. MEDIUM - In this mode, the quality is optimal. STANDARD - In this mode, the performance is high.

TARAXL_STATUS_CODE savePoints(TARAXL_POINTCLOUD_FORMAT pointcloudFormat, std::string filename)

This function saves the current Point Cloud in the specified format (PCD, PLY, VTK).

Parameters	Description
`pointcloudFormat`	This parameter saves the Point Cloud in specified format. The supported formats are: TARAXL_PLY_CLOUD - Saves the Point Cloud in .ply format.

	TARAXL_PCD_CLOUD - Saves the Point Cloud in .pcd format. TARAXL_VTK_CLOUD - Saves the Point Cloud in .vtk format.
`filename`	The filename to save the Point Cloud.

TARAXL_STATUS_CODE getPoints(Points::Ptr currentCloud)

This function obtains the Point Cloud of the current frame and fills the data in current Cloud.

Parameters	Description
`currentCloud`	Pointer of type points in which the Point Cloud data is filled.

TaraXLIMUData Structure

The details regarding the supported TaraXLIMUData structure are explained below.

The supported TaraXLIMUData structure members are as follows:

- Vector3 angularVelocity
- Vector3 linearAcceleration
- Vector3 inclination
- Vector3 getInclination()

Vector3 angularVelocity

The vector size of 3 is used to store the x, y and z values of angular velocity. The angular velocity values are obtained in degrees per second (dps).

Vector3 linearAcceleration

The vector size of 3 is used to store the x, y and z values of linear acceleration. The linear acceleration values are obtained in milli g (mg).

1 milli g = 0.001 g = 0.001 * 9.8 ms^{-2}

Vector3 inclination

The vector size of 3 is used to store the x, y and z values of inclination. The inclination values are obtained in degrees.

Vector3 getInclination()

The vector size of 3 is used to store the x, y and z values of inclination. The inclination values are obtained in degrees.

CalibrationParams Structure

The details regarding the supported CalibrationParams structure are explained below.

The supported CalibrationParams structure members are as follows:

- cv::Mat cameraMatrix
- cv::Mat distortionMatrix
- double apertureWidth
- double apertureHeight
- double fovX
- double fovY
- double focalLength

cv::Mat cameraMatrix

This Mat stores the camera matrix of the corresponding camera.

cv::Mat distortionMatrix

This Mat stores the distortion matrix of the corresponding camera.

double apertureWidth

Physical width of the sensor in mm.

double apertureHeight

Physical height of the sensor in mm.

double fovX

Horizontal FOV in degrees.

double fovY

Vertical FOV in degrees.

double focalLength

Focal length of the lens in mm.

TaraXLPoseTracking Class

The details regarding the supported TaraXLPoseTracking class are explained below.

The supported TaraXLPoseTracking class method are as follows:

- TaraXLPoseTracking(TaraXLCam &camera)
- TARAXL_STATUS_CODE setIMUOutputFrequency(TARAXL_IMU_OUTPUT_FREQUENCY frequency)
- TARAXL_STATUS_CODE getIMUOutputFrequency(TARAXL_IMU_OUTPUT_FREQUENCY &frequency)
- TARAXL_STATUS_CODE getIMUData(TaraXLIMUData &data)

TaraXLPoseTracking(TaraXLCam &camera)

This function initialize the constructor with a valid TaraXLCam device object obtained from TaraXL::enumerateDevices method. An application must first call TaraXLCam::connect() method before invoking this constructor.

Parameters	Description
`camera`	The context of the camera device is returned in TaraXL::enumerateDevices method.

TARAXL_STATUS_CODE setIMUOutputFrequency(TARAXL_IMU_OUTPUT_FREQUENCY frequency)

This function sets the IMU output frequency.

Parameters	Description
`frequency`	The IMU output frequency is set using this value. To know the list of supported frequencies, please refer to the *TARAXL_IMU_OUTPUT_FREQUENCY* section.

TARAXL_STATUS_CODE getIMUOutputFrequency(TARAXL_IMU_OUTPUT_FREQUENCY &frequency)

This function obtains the current IMU output frequency.

Parameters	Description
`frequency`	The frequency of operation is obtained and stored in this parameter that is passed as reference. To know the list of supported frequencies, please refer to the *TARAXL_IMU_OUTPUT_FREQUENCY* section.

TARAXL_STATUS_CODE getIMUData(TARAXLIMUData &data)

This function obtains the linear acceleration, angular velocity and inclination values.

Parameters	Description
`data`	A reference of TaraXLIMUData object. This function fills the values of angularVelocity, linearAcceleration and getInclincation.

TARAXL_IMU_OUTPUT_FREQUENCY

The supported IMU output frequency configuration is shown below.

```
Revision A :
IMU_119_HZ   - 119Hz
IMU_238_HZ   - 238Hz
IMU_476_HZ   - 476Hz
IMU_952_HZ   - 952Hz

Revision B :
IMU_12_5_HZ  - 12.5Hz
IMU_26_HZ    - 26Hz
IMU_52_HZ    - 52Hz
IMU_104_HZ   - 104Hz
IMU_208_HZ   - 208Hz
IMU_416_HZ   - 416Hz
IMU_833_HZ   - 833Hz
IMU_1666_HZ  - 1666Hz
```

TARAXL_FILTER_TYPE

The supported TaraXL filter type configuration is shown below.

```
TARAXL_FILTER_TYPE:
TARAXL_DEFAULT_FILTER = 0
TARAXL_MEDIAN_FILTER = 1
```

TARAXL_STATUS_CODE

The methods of TaraXL, TaraXLCam and TaraXLDepth classes return the status codes as shown below.

```
TARAXL_FAILURE = 0
TARAXL_SUCCESS = 1
NO_DEVICES_CONNECTED = 100
FIRMWARE_NOT_COMPATIBLE = 102
INTERNAL_PIPELINE_ERROR = 104
CAMERA_OPEN_FAILED = 201
CAMERA_NOT_AVAILABLE = 202
NOT_INITIALISED = 204
UNSUPPORTED_RESOLUTION = 205
EXTENSION_UNIT_FAILED = 206
RECTIFICATION_FAILED = 207
UNSUPPORTED_FREQUENCY_FOR_THIS_REVISION = 208
IMU_EXTENSION_UNIT_FAILED = 209
EXPOSURE_SETTING_FAILED = 301
FAILED_TO_LOAD_INTRINSIC_AND_EXTRINSIC_FILES = 302
INVALID_INTRINSIC_AND_EXTRINSIC_FILE_LENGTH = 303
INVALID_INTRINSIC_AND_EXTRINSIC_FILE_DATA = 304
INPUT_MATRIX_EMPTY = 305
EXPOSURE_OUT_OF_BOUNDS = 306
EXPOSURE_GETTING_FAILED = 307
AUTO_EXPOSURE_SETTING_FAILED = 308
BRIGHTNESS_OUT_OF_BOUNDS = 309
GAIN_OUT_OF_BOUNDS = 310
GET_UNIQUE_ID_FAILURE = 311
SET_IMU_FREQUENCY_FAILURE = 312
BRIGHTNESS_SETTING_FAILED = 313
GAIN_SETTING_FAILED = 314
BRIGHTNESS_GETTING_FAILED = 315
CALIBRATION_DATA_NOT_FOUND = 316
CALIBRATION_DATA_CORRUPTED = 317
GAIN_GETTING_FAILED = 318
DEPTH_RANGE_NOT_AVAILABLE_FOR_THE_SELECTED_MODE = 400
```

Support

Contact Us

If you need any support on TaraXL product, please contact us using the Live Chat option available on our website - https://www.e-consystems.com/

Creating a Ticket

If you need to create a ticket for any type of issue, please visit the ticketing page on our website - https://www.e-consystems.com/create-ticket.asp

RMA

To know about our Return Material Authorization (RMA) policy, please visit the RMA Policy page on our website - https://www.e-consystems.com/RMA-Policy.asp

General Product Warranty Terms

To know about our General Product Warranty Terms, please visit the General Warranty Terms page on our website - https://www.e-consystems.com/warranty.asp

Revision History

Rev	Date	Description	Author
1.0	23-August-2018	Initial Draft	Vision Team
1.1	13-September-2018	Added contents for Ubuntu 16.04 PC	Vision Team
2.0	12-November-2018	Added contents for Point Cloud and IMU APIs	Vision Team
3.0	28-February-2019	Added common APIs for See3CAM_StereoA and STEEReoCAM. Updated the exposure range for STEEReoCAM	Vision Team
3.1	14-March-2019	Added Filter enums	Vision Team
3.2	30-March-2019	Removed WLS filter enum	Vision Team
3.3	24-May-2019	Added 3 new APIs	Vision team
3.4	03-June-2019	Added descriptions to some APIs	Vision Team
3.5	30-July-2019	Updated L4T versions	Vision Team
3.6	13-August-2019	Added Contents for Nano	Vision Team
3.7	21-October-2019	Added APIs in TaraXL depth class and enums	Vision Team
3.8	08-November-2019	Updated getCalibrationParameters API and CalibrationParams structure	Vision Team