



e-con Systems India Pvt Ltd

RISHABH INFO PARK,
RR Tower – IV, 7th Floor,
Super A-16 * A-17, Thiru-Vi-Ka Industrial Estate,
Guindy, Chennai - 600 032.
www.e-consystems.com

Tara - Stereo Vision USB 3.0 Camera

Tara – Linux API Manual

Revision 1.3

15-Dec-2016



Contents

1 Revision History.....	2
2 Introduction	3
3 Scope.....	3
4 Description.....	4
5 Block Diagram	4
6 Supported APIs.....	5
6.1 TaraCamParameters APIs.....	5
6.1.1 BOOL Init()	6
6.1.2 BOOL RemapStereoImage(Mat *RImage, Mat *LImage)	6
6.1.3 BOOL LoadCameraMatrix()	6
6.1.4 BOOL GetMatforCV(Mat SrcImage, Mat *DestImage)	6
6.1.5 BOOL ComputeRectifyPrms().....	6
6.2 Disparity APIs.....	7
6.2.1 BOOL InitCamera(bool GenerateDisparity, bool FilteredDisparityMap)	7
6.2.2 BOOL GrabFrame(Mat *LeftImage, Mat *RightImage)	7
6.2.3 BOOL GetDisparity(Mat LImage, Mat RImage, Mat *DisparityMap, Mat *disp_filtered)..	7
6.2.4 BOOL EstimateDepth(Point Pt, float *DepthValue)	8
6.2.5 BOOL SetExposure(int ExposureVal)	8
6.2.6 BOOL GetExposure(int *ExposureVal)	8
6.2.7 BOOL SetAutoExposure()	9
6.2.8 BOOL SetAlgorithmParam().....	9
6.2.9 BOOL Init(bool GenerateDisparity)	9
6.2.10 BOOL SetBrightness(double BrightnessVal)	9
6.2.11 BOOL SetStreamMode(UINT32 StreamMode).....	10
6.2.12 BOOL GetStreamMode(UINT32 *StreamMode).....	10
6.3 CameraEnumeration APIs	10
6.3.1 int GetListofDevicesCon().....	10
6.3.2 BOOL GetDeviceIDeCon(int *DeviceID, Size *ResolutionSelected)	10
6.3.3 BOOL IsStereoDeviceAvail(char *pid)	11
6.3.4 CameraEnumeration(int *DeviceID, Size *ResolutionSelected)	11
6.3.5 void query_resolution(int DeviceID)	11
6.3.6 void freeDevices(void).....	11
7 Conclusion	12



Tara – Linux API Manual

1 Revision History

Rev No	Date	Major Changes	Author
1.1	21-May-2016	Initial Draft	Karthikeyan A
1.2	14-Sep-2016	Added Auto Exposure and Trigger Support	Karthikeyan A
1.3	15-Dec-2016	Added Auto Exposure Support	Karthikeyan A



2 Introduction

The commonly used functions are put into a single namespace for modularity, in order to reuse them in the examples. Tara is the namespace that contains the following three classes:

- i. TaraCamParameters
- ii. Disparity
- iii. CameraEnumeration

3 Scope

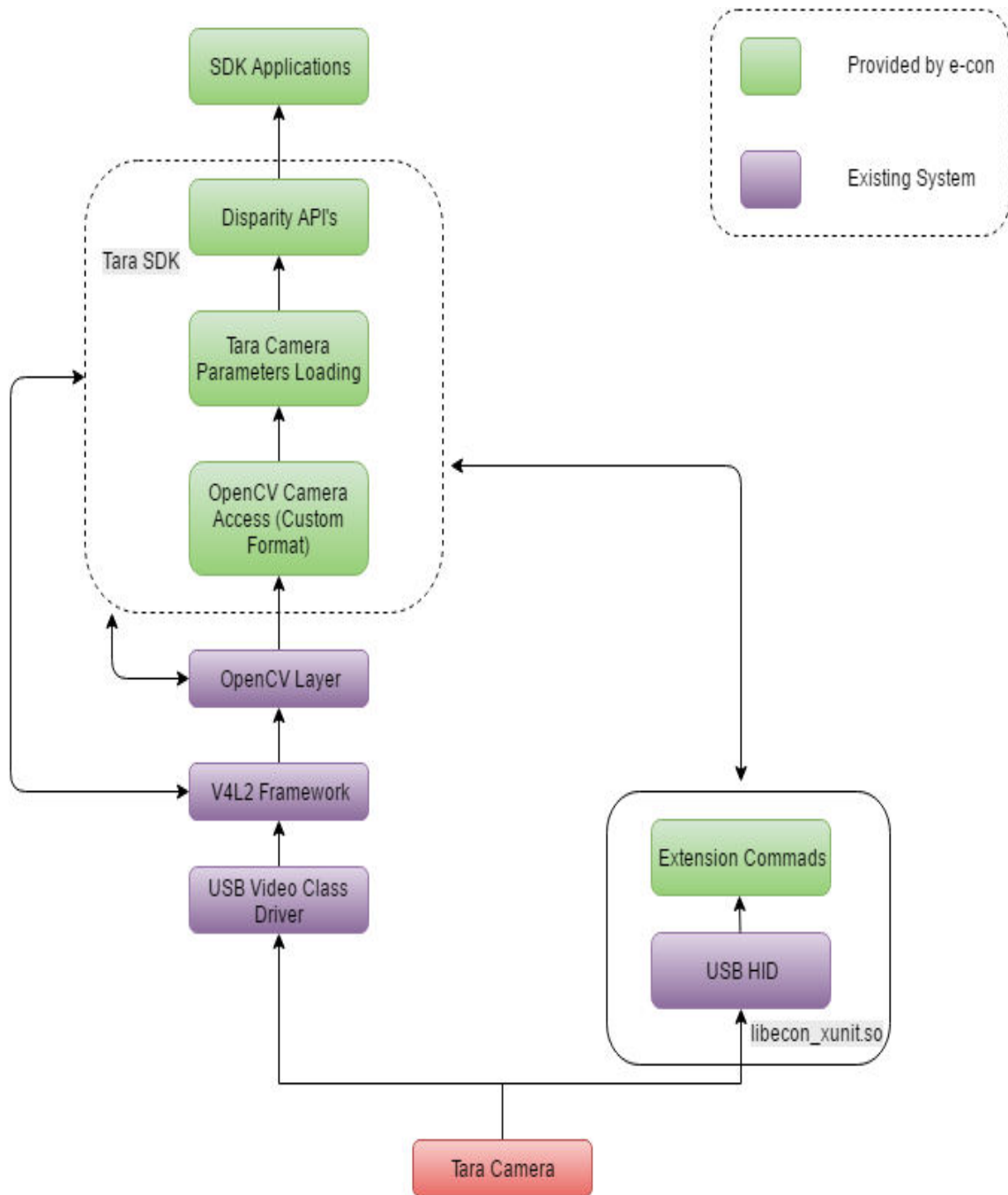
The scope of this document is to highlight the the common Tara API's that are currently used in the sample applications in Linux SDK package.

4 Description

Tara namespace contains many commonly used functions. Loading the calibrated files from the camera flash, computing disparity map, rectification of frames, camera enumeration and initialization, etc. are some of them. These functions can be readily used in your application.



5 Block Diagram



6 Supported APIs

6.1 TaraCamParameters APIs

This class contains methods to load the camera matrix from the flash, compute the Q Mat and to rectify the images.

6.1.1 BOOL Init()

Invokes the method LoadCameraMatrix.

6.1.2 BOOL RemapStereoImage(Mat *RImage, Mat *LImage)

Parameters: RImage : Pointer to store the rectified Right Image of the camera
LImage : Pointer to store the rectified Left Image of the camera

Return values: TRUE on Success
FALSE on failure

Description: Performs rectification of right and left image with the computed matrix values, such that the images differ only in the x axis.

6.1.3 BOOL LoadCameraMatrix()

Return values: TRUE on Success
FALSE on failure

Description: Loads the camera matrix from the flash of the camera using the HID commands. Loads all the data from the intrinsic and extrinsic file to the Mat objects created.

6.1.4 BOOL GetMatforCV(Mat SrcImage, Mat *DestImage)

Parameters: SrcImage : Input Image of the camera
DestImage : Pointer to store a Mat mapped for the correct version of OpenCV

Return values: TRUE on Success
FALSE on failure

Description: Method is to check the OpenCV version used to avoid the errors in using the extrinsic and intrinsic files.



6.1.5 BOOL ComputeRectifyParams()

Return values: TRUE on Success
FALSE on failure

Description: Computes the Q Mat used for the transformation of 2D points to 3D points. The Mapping matrices to rectify the image is also created.

6.2 Disparity APIs

This class contains modules to initialize the camera, disparity parameters, estimates the depth of the point specified, etc.

6.2.1 BOOL InitCamera(bool GenerateDisparity, bool FilteredDisparityMap)

Parameters: GenerateDisparity: True - Generates Disparity Map
FilteredDisparityMap: True - Generates Filtered Disparity Map

Return values: TRUE on Success
FALSE on failure

Description: Creates a object for the CameraEnumeration class where the Device ID and Resolution selected by the user are returned. Opens the camera device with the following inputs. Invokes the InitExtensionUnit method of the HID. The Init function is called based on the input the parameters of the Disparity map are set.

6.2.2 BOOL GrabFrame(Mat *LeftImage, Mat *RightImage)

Parameters: RightImage : Pointer to store the rectified Right Image of the camera
LeftImage : Pointer to store the rectified Left Image of the camera

Return values: TRUE on Success
FALSE on failure

Description: Grabs the frame from the camera device initialized. Splits the interleaved Y16 data into right image and left image respectively. Rectifies the frame using the RemapStereoImage method and pass the image as reference.

Note: For the camera to stream the Custom format, OpenCV has to be rebuild. Refer to the Tara_SDK_Linux_User_Manual in the documents folder of the SDK.



6.2.3 BOOL GetDisparity(Mat LImage, Mat RImage, Mat *DisparityMap, Mat *disp_filtered)

Parameters: RightImage : Rectified Right Image of the camera
LeftImage : Rectified Left Image of the camera
DisparityMap : Pointer to store the Disparity Map computed
disp_filtered : Pointer to store the Disparity Map for Visualization

Return values: TRUE on Success
FALSE on failure

Description: Computes the disparity map for the left and right images passed with the parameters set in the SetAlgorithmParam method.

6.2.4 BOOL EstimateDepth(Point Pt, float *DepthValue)

Parameters: Point: Point at which the depth is to be found
DepthValue: Pointer to store the Depth value of the point passed

Return values: TRUE on Success
FALSE on failure

Description: Estimates the depth of the point passed from the disparity map computed and returns the depth of the point.

6.2.5 BOOL SetExposure(int ExposureVal)

Parameters: ExposureVal: Exposure Value to be set in the camera.

Return values: TRUE on Success
FALSE on failure

Description: Invokes the HID function SetManualExposureStereo to set the exposure of the device initialized. The range of the exposure is 10 to 1000000 micro seconds.

6.2.6 BOOL GetExposure(int *ExposureVal)

Parameters: ExposureVal: Pointer to store the current Exposure value of the camera.



Return values: TRUE on Success
FALSE on failure

Description: Invokes the HID function GetManualExposureStereo to read the exposure of the device. The range of the manual exposure is 10 to 1000000 micro seconds. When the Camera is in Auto Exposure, ExposureVal will be read as 1.

6.2.7 BOOL SetAutoExposure()

Return values: TRUE on Success
FALSE on failure

Description: Invokes the HID function SetAutoExposureStereo to enable the Auto Exposure in the Camera. Exposure value has to be set to 1, to enable Auto Exposure.

6.2.8 BOOL SetAlgorithmParam()

Return values: TRUE on Success
FALSE on failure

Description: Sets up the disparity algorithm(Stereo BM / Stereo 3 Way) to run based on the macro DISPARITY_OPTION.

DISPARITY_OPTION

- I. When set to 1 - Best Quality Depth Map and Lower Frame Rate - Stereo_SGBM 3 Way generic Left and Right.
- II. When set to 0 - Low Quality Depth Map and High Frame Rate - Stereo_BM generic Left and Right

6.2.9 BOOL Init(bool GenerateDisparity)

Parameter: GenerateDisparity: True - Generates the Disparity Map

Return values: TRUE on Success
FALSE on failure

Description: Invokes the Init method of TaraCamParameters. Initialises all the disparity variables and invokes the SetAlgorithmParam.



6.2.10 BOOL SetBrightness(double BrightnessVal)

Parameters: BrightnessVal: Brightness Value to be set in the camera.

Return values: TRUE on Success
FALSE on failure

Description: Uses the set property of VideoCapture in OpenCV to set the brightness. Value ranges from 1 to 7. The value has to be normalized (by the max val 7), so the range to be given to the API is (1/7 to 7/7).

6.2.11 BOOL SetStreamMode(UINT32 StreamMode)

Parameters: StreamMode: Sets the Stream mode passed to the camera

Return values: TRUE on Success
FALSE on failure

Description: Sets the camera to the stream mode passed. Tara supports two modes i) Master mode ii) Trigger Mode. 1 is passed to set Master Mode and 0 is passed to set Trigger Mode.

6.2.12 BOOL GetStreamMode(UINT32 *StreamMode)

Parameters: StreamMode: Pointer to the store the Stream Mode is passed.

Return values: TRUE on Success.
FALSE on failure.

Description: Returns the stream mode in which the camera is streamed. Tara supports two modes i) Master mode ii) Trigger Mode. StreamMode is set to 1 when the camera is in Master Mode and 0 when in Trigger Mode.

6.3 CameraEnumeration APIs

6.3.1 int GetListofDevicesCon()

Return values: Returns the number of connected devices to system.

Description: Enumerates the Devices connected to the computer using the V4L2 framework.



6.3.2 BOOL GetDeviceIDeCon(int *DeviceID, Size *ResolutionSelected)

Parameters: DeviceID: Pointer to store the Device ID of the camera to be streamed.
ResolutionSelected: Pointer to store the Resolution to be set to the Camera Device

Return values: TRUE on Success
FALSE on failure

Description: Gets the user input for the device to be streamed Device ID and Resolution to be set for the camera to stream.

6.3.3 BOOL IsStereoDeviceAvail(char *pid)

Parameters: pid: Pointer holding the Product ID of the device selected.

Return values: TRUE on Success
FALSE on failure

Description: Checks the whether the enumerated device is e-con's Stereo Camera.

6.3.4 CameraEnumeration(int *DeviceID, Size *ResolutionSelected)

Parameters: DeviceID: Pointer to store the Device ID of the camera to be streamed.
ResolutionSelected: Pointer to store the Resolution to be set to the Camera Device

Description: Constructor of the class which invokes the GetDeviceIDeCon method.

6.3.5 void query_resolution(int DeviceID)

Parameters: DeviceID: Device ID of the camera to be streamed.

Description: Query the resolutions supported by the selected camera in Y16 format and saves it in a vector declared in the CameraEnumeration class.

6.3.6 void freeDevices(void)

Description: Destroys the memory allocated while creating the device list of enumerated cameras.



7 Conclusion

This document provides the Tara API Design framework to be used in the Linux application.

