1. Abstract

This guide explains how to get started with the MicroVision Windows Software Development Kit (SDK) for its consumer LiDAR Scanning Engines. The LiDAR Windows SDK allows developers to quickly and easily integrate Scanning Engine control into a Windows application. The Windows Application can communicate with the unit over a USB cable. There is a package reference in this doc which includes the C Application Programming Interface (API), SDK libraries, documentation, and a sample visualization project that demonstrate use of some basic API functions. The sample application has been developed and built using Microsoft Visual Studio 2015.

2. Table of Contents

3. Acquiring and Installing the Windows SDK

The latest Windows SDK can be cloned or downloaded from:
https://github.com/MicroVision-Inc/Consumer-LiDAR

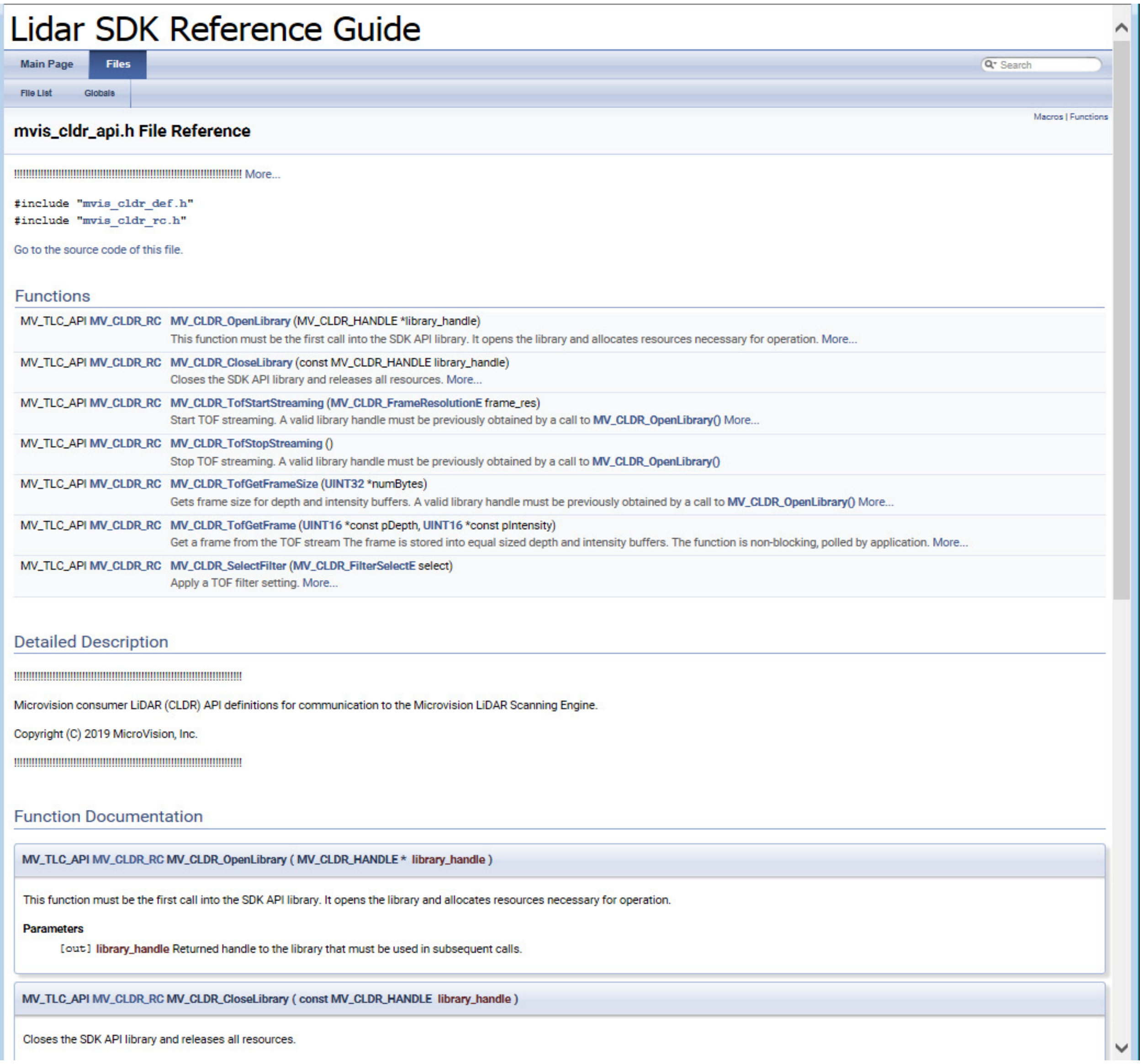
The downloaded destination folder will be under mv_cldr_sdk subfolder:

The files/folders included in the SDK distribution package are described in more detail below:

LiDAR_Windows_SDK				*SDK Root folder*
	lib			
		Mvis_cldr_api.dll		*Windows DLL*
		Mvis_cldr_api.lib		*Windows library*
		Lib_cldr_api.so		*Linux so file*
	inc			
		Mvis_cldr_api.h		*Api header file*
		Mvis_cldr_def.h		*Def header file*
		Mvis_cldr_rc.h		*Return code header file*
	doc			
		user_guide		*Detailed Programmer's User Guide*
			html	
			index.html	*Programmer's Reference Guide entry point*
			*.html	*Programmer's Reference Guide documentation*
		Consumer_LiDAR_Windows_SDK _Getting_Started_Guide.pdf		*This Getting Started guide for LiDAR Windows SDK.*

4. Accessing SDK Documentation

For description of the C-language API, please refer to the Programmer's Reference Guide at *Doc\User_Guide\html\index.html*. The Reference Guide is a set of hyperlinked HTML files containing detailed description of all Function interfaces and definitions provided by the API.



Lidar SDK Reference Guide

mvis_cldr_api.h File Reference

!!! More...

```
#include "mvis_cldr_def.h"
#include "mvis_cldr_rc.h"
```

Go to the source code of this file.

Functions

MV_TLC_API MV_CLDR_RC	MV_CLDR_OpenLibrary (MV_CLDR_HANDLE *library_handle) This function must be the first call into the SDK API library. It opens the library and allocates resources necessary for operation. More...
MV_TLC_API MV_CLDR_RC	MV_CLDR_CloseLibrary (const MV_CLDR_HANDLE library_handle) Closes the SDK API library and releases all resources. More...
MV_TLC_API MV_CLDR_RC	MV_CLDR_TofStartStreaming (MV_CLDR_FrameResolutionE frame_res) Start TOF streaming. A valid library handle must be previously obtained by a call to MV_CLDR_OpenLibrary() More...
MV_TLC_API MV_CLDR_RC	MV_CLDR_TofStopStreaming () Stop TOF streaming. A valid library handle must be previously obtained by a call to MV_CLDR_OpenLibrary()
MV_TLC_API MV_CLDR_RC	MV_CLDR_TofGetFrameSize (UINT32 *numBytes) Gets frame size for depth and intensity buffers. A valid library handle must be previously obtained by a call to MV_CLDR_OpenLibrary() More...
MV_TLC_API MV_CLDR_RC	MV_CLDR_TofGetFrame (UINT16 *const pDepth, UINT16 *const pIntensity) Get a frame from the TOF stream The frame is stored into equal sized depth and intensity buffers. The function is non-blocking, polled by application. More...
MV_TLC_API MV_CLDR_RC	MV_CLDR_SelectFilter (MV_CLDR_FilterSelectE select) Apply a TOF filter setting. More...

Detailed Description

!!!

Microvision consumer LiDAR (CLDR) API definitions for communication to the Microvision LiDAR Scanning Engine.

Copyright (C) 2019 MicroVision, Inc.

!!!

Function Documentation

MV_TLC_API MV_CLDR_RC MV_CLDR_OpenLibrary (MV_CLDR_HANDLE * library_handle)

This function must be the first call into the SDK API library. It opens the library and allocates resources necessary for operation.

Parameters

`[out]` **library_handle** Returned handle to the library that must be used in subsequent calls.

MV_TLC_API MV_CLDR_RC MV_CLDR_CloseLibrary (const MV_CLDR_HANDLE library_handle)

Closes the SDK API library and releases all resources.

5. Using the LiDAR Application Programming Interface (API)

The LiDAR SDK for Windows allows you to easily interface your application software with the LiDAR Scanning Engine. This section explains the steps of the integration process.

5.1. Step 1: Link SDK Library to the application

To include the LiDAR Windows SDK to an application, include the **`mvis_cldr_api.h`** header file into the application source code and link the **`mvis_cldr_api.lib`** library to the application executable:

```
// Include this into source code …
#include "mvis_cldr_api.h"

<!-- Make sure Visual studio project file includes the proper -->
<!-- dependencies to the picop_api_sentinel.lib library and include paths -->
<!--See chapter 7.2 on how to configure Visual Studio Project properties -->
<Link>
   <AdditionalDependencies>mvis_cldr_api.lib;%(AdditionalDependencies)
   </AdditionalDependencies>
   <AdditionalLibraryDirectories>../../Lib</AdditionalLibraryDirectories>
</Link>
<ClCompile>
   <AdditionalIncludeDirectories>../../Interface</AdditionalIncludeDirectories>
</ClCompile>
```

5.2. Step 2: Initialize the SDK Library

The first step in connecting to a LiDAR is to initialize the LiDAR library by calling **`MV_CLDR_OpenLibrary().`** The **`MV_CLDR_OpenLibrary()`** function returns a handle to the library that can be used to open a connections to the LiDAR device.

5.3. Step 3: Connect to LiDAR

After successful initialization of the library, the next step is to enable streaming of point cloud data from device and capture point cloud frame data for processing.

5.4. Step 4: Configure and Control LiDAR

The LiDAR API is split into the following functional categories:

- **Connection & Library Management**: Connect to LiDAR over USB interface.
- **3D Sensing:** Configure and control LiDAR 3D Sensing interface.

5.4.1. Connection Management

The Connection Management Functions are used to connect to the LiDAR Scan Engine through USB.

Function	Description
MV_CLDR_OpenLibrary( )	Opens the library and allocates resources necessary for operation. It returns a handle to the library that must be used in subsequent calls This MUST be the first call into the XXX library
MV_CLDR_CloseLibrary( )	Closes the library and releases all resources. It also closes all the open connections. This MUST be the last call into the XXX library

5.4.1. 3D Sensing Functions

The 3D Sensing functions are used to configure the 3D sensor as well as to retrieve the 3D depth data.

Function	Description
MV_CLDR_TofStartStreaming ( )	Turns the 3D Sensing function on. When the 3D Depth Sensing function is on, the IR laser is pulsing and 3D depth sensing data is sent over either the USB or MIPI-CSI interfaces depending upon the specific MVIS Sensor.
MV_CLDR_TofStopStreaming()	Turns the 3D Sensing function off. When the 3D Depth Sensing function is off, the IR laser is not pulsing and no depth 3D sensing data is sent over the USB or MIPI-CSI.
MV_CLDR_TofGetFrameSize()	Get frame size in bytes for depth and intensity buffers. Both buffers are of the same size.
MV_CLDR_TofGetFrame()	Get a frame of 3D depth sensing data
MV_CLDR_SelectFilter()	Apply a depth filter setting. Currently not implemented

5.5. Step 5: Exit Application

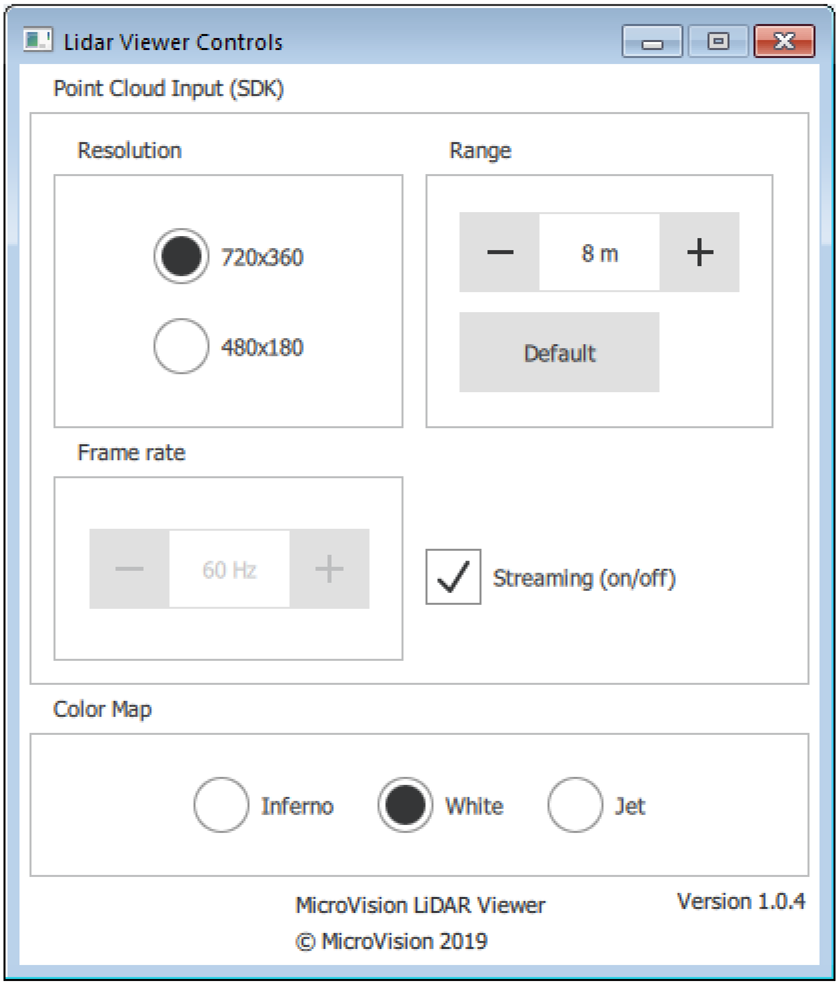
To gracefully exit the host application, call the and **`MV_CLDR_CloseLibrary()`** functions to shut down the connection to and to release all resources used by the library.

6. LiDAR Viewer Sample Project

The **`LiDAR Viewer`** sample project is a viewing application developed in C++. It demonstrates the use of SDK interfaces to communicate with the device and to view and evaluate point cloud data captured by the Sentinel sensor. The application can control the following operations:

- resolution
- turning streaming on and off

The application typical user interface is shown below and actual user interface might be slightly different



Point Cloud Input (SDK)

Resolution: ● 720x360 ○ 480x180

Range: 8 m

Frame rate: 60 Hz

☑ Streaming (on/off)

Color Map: ○ Inferno ● White ○ Jet

MicroVision LiDAR Viewer
© MicroVision 2019

Version 1.0.4

6.1. Prerequisites

The System Requirements for running and compiling the sample viewer project are:

- 64-bit Windows operating system: Windows 7 or later
- 2.2 GHz or faster processor, minimum 4GB of RAM
- Microsoft Visual Studio 2015 or later.

6.2. Acquiring Sample Application

The sample application can be cloned or downloaded from:
https://github.com/MicroVision-Inc/MicroVision-LiDAR-Vision

6.3. Accessing Sample Application Documentation

Refer to readme.md of GitHub repo https://github.com/MicroVision-Inc/MicroVision-LiDAR-Vision for prerequisites and how to build the sample viewer application.