



Command Line App

SDK Documentation

Version 2.7
November 8, 2018

Table of contents

1. About This Guide.....	3
2. Introduction.....	3
3. Getting started with the code.....	3
4. Run the application	3
5. Commands	4

1. About This Guide

This document will help software developers to integrate the INOGENI HID interface to their software. Using the HID interface software developer will be able to update the INOGENI USB controller (FX3), video processor (FPGA) and EDID in the field. You will also be able to change the device configuration if this one applies to the current device.

The application relies on the cross-platform libusb¹ API.

2. Introduction

The INOGENI device is a composite USB device made of a UVC device, a UAC device and a HID device. The UVC function is used to send video from the device to the host. The UAC device is used to send audio to the host. The HID interface is used to send / receive commands between the device and the host. Both functions do not need a custom driver to work. The UVC, UAC and HID functions are using the built-in driver of popular operating systems like Windows, macOS and Linux. This document describes all the details about the HID function only.

To update the device software, we need to update the binary code for the FPGA, FX3 and EDID inside the device EEPROM. Once the binary in the EEPROM have been updated, a reset of the device is necessary so the device use the newly updated binary.

3. Getting started with the code

In order to compile the sample code, you need first to compile the libusb library.

On Windows using Visual Studio 2015 or later, you must open the solution at the following path: libusb/1.0.21/msvc/libusb_2015.sln. You can then build the library on x86 and x64 platform, using Debug and Release. **You can then open the command line solution. The solution should have proper libraries linked. Verify project settings if the solution does not build.**

Using Linux, you must make sure you have libusb installed on your system. Using Ubuntu, do the following:

```
>> sudo apt-get install libusb-1.0-0-dev
```

If you want to build the command line app under Linux, you can use gcc using the following command:

```
>> gcc main.c HIDCommand.c INOGENIHID.c -o inogeni -lusb-1.0
```

4. Run the application

You can run the application using the following syntax

```
>> INOGENICmdLineApp <command> <parameter>
```

¹ <https://libusb.info/>

5. Commands

Here is a list of the commands.

Command	Description
version	Get version command.
video	Get video input info.
uedid <filepath>	Update EDID. where <filepath> points to the file to update.
vedid <filepath>	Verify EDID. where <filepath> points to the file to verify.
ufpga <filepath>	Update FPGA. where <filepath> points to the file to update.
vfpga <filepath>	Verify FPGA. where <filepath> points to the file to verify.
ufx3 <filepath>	Update FX3. where <filepath> points to the file to update.
vfx3 <filepath>	Verify FX3. where <filepath> points to the file to verify.
reset	Reset device.
setaudio <parameter>	Set audio channel. Applies only on SHARE2. where <parameter> is: 0 => AUTO 1 => HDMI 2 => DVI 3 => ANALOG
sdescr <string>	Set descriptor. Applies to 4K2USB3. where <string> is the string to program on board.
gdescr	Get descriptor. Applies to 4K2USB3.
sview <parameter>	Set the view mode. Applies only on SHARE2 and SHARE2U. where <parameter> is: 0 => VIEW_MODE_SINGLE1 1 => VIEW_MODE_SINGLE2 2 => VIEW_MODE_SIDE_BY_SIDE 3 => VIEW_MODE_BIG_AND_SMALL 4 => VIEW_MODE_PIP_TL 5 => VIEW_MODE_PIP_TR 6 => VIEW_MODE_PIP_BL 7 => VIEW_MODE_PIP_BR *** Priority on input 1 *** 8 => VIEW_MODE_SIDE_BY_SIDE_1 9 => VIEW_MODE_BIG_AND_SMALL_1 10 => VIEW_MODE_PIP_TL_1 11 => VIEW_MODE_PIP_TR_1 12 => VIEW_MODE_PIP_BL_1 13 => VIEW_MODE_PIP_BR_1 *** Priority on input 2 *** 14 => VIEW_MODE_SIDE_BY_SIDE_2 15 => VIEW_MODE_BIG_AND_SMALL_2 16 => VIEW_MODE_PIP_TL_2 17 => VIEW_MODE_PIP_TR_2 18 => VIEW_MODE_PIP_BL_2 19 => VIEW_MODE_PIP_BR_2
swap	Set the swap mode. Applies only on SHARE2 and SHARE2U.
help	Get info for all commands.