APPS047-Codensity T408^{TM} Decklink Interworking Configuration and Usage Application Note

V4.0

2022-09-06

Table of Contents

1 Table of Abbreviations

Abbreviation:	Full Form:
T408	Codensity T408 Video Transcoder
Decklink	Video Capture and Playback solution from Blackmagic like Decklink 8K Pro PCIe cards.
AI	Artificial Intelligence
API	Application Programming Interface
AUD	Access Unit Delimiters
AV	Audio Video
AV1	Alliance Open Media Video 1 Codec
CABAC	Context Adaptive Binary Arithmetic Coding
CBR	Constant Bit Rate
CPU	Central Processing Unit
CRF	Constant Rate Factor
DSP	Digital Signal Processing
EOF	End Of File
FIXQB	Fixed Quantization Parameter
FF	Form Factor
FPS	Frames Per Second
GOP	Group of pictures
HDR	High Dynamic Range
HHHL	Half Height, Half Length
HHFL	Half Height, Full Length
HLG	Hybrid Log Gamma
HRD	Hypothetical Reference Decoder
IE	Inference Engine
Mbps	Mega Bit per second
MBps	Mega Byte per second
NLP	Natural Language Processing
NPU	Neural Processing Unit
NVMe	Non-Volatile Memory express
PCIe	Peripheral Component Interconnect express
PPS	Picture Parameter Set
QP	Quoted Printable
RGB	Red Green Blue
RGBA	Red Green Blue Alpha
RDO	Optimization

ROI	Region of interest
SDK	Software Development Kit
SEI	Supplemental Enhancement Information
SRIOV	Single Root I/O Virtualization
SPS	Sequence Parameter Set
VBR	Variable Bit Rat
VCL	Video Coding Layer
VPS	Video Parameter Set
VUI	Video Usability Information

2 Background

2.1 Scope

This document details the interworking of Blackmagic Decklink card with T408 Transcoder. Decklink cards allow capture of camera live feeds directly into video processing software like ffmpeg. In such use case, the raw video feed can be fed directly to T408 for Encoding to file or live streaming.

The Section 3 covers the Decklink environment install on a Linux Host. Section 4 demonstrates the ffmpeg compile with Decklink support. Section 5 lists the ffmpeg commands demonstrating Decklink features and an encoding operation using Decklink input and T408 Encoder. Section 6 concludes with instructions on Decklink environment for a Windows host.

2.2 Intended Audience

This document is intended for engineers/developers/technicians operating NETINT transcoders.

2.3 Compatibility

T408 / T432 Product	**Version**
Minimum Hardware Revision	2.5.0
Minimum Firmware Revision	2.5.0
Minimum FFmpeg Revision	4.2.1

3 Decklink Environment - Linux

3.1 Scope

This section details the installation of Decklink drivers on a Linux Host.

3.2 Installation and Check

Download the Blackmagic Decklink software update package from Blackmagic support website. Refer Section 3.2.2 for link. File will be titled like Blackmagic_Desktop_Video_Linux_12.1 or similar.

Untar the file.

```
tar -xvf Blackmagic_Desktop_Video_Linux_12.1.tar
cd Blackmagic_Desktop_Video_Linux_12.1.tar/deb/x86_64
sudo dpkg -i mediaexpress_3.7a7_amd64.deb
sudo dpkg -i desktopvideo_12.1a9_amd64.deb
sudo dpkg -i desktopvideo-gui_12.1a9_amd64.deb
```

Reboot the host. Check installation using the following commands:

```
sudo dmesg
sudo modinfo blackmagic
sudo modinfo blackmagic-io
lsmod |grep magic
```

Sample 'sudo dmesg' command log confirming Decklink detection.

```
[17080.501524] BlackmagicIO: Driver version 12.1a9 loading
[17080.501548] blackmagic-io 0000:02:00.0: enabling device (0000 -> 0002)
[17080.502031] BlackmagicIO: WARNING: "DeckLink 8K Pro" has a firmware version mismatch (valid: 1 device: 0x0028 driver: 0x0030)
[17080.502040] BlackmagicIO: Enabled device "DeckLink 8K Pro (1)" x4/5 Gbps (5953fb0,580200) FW Date: 02-26 15:56
```

[17080.520443] BlackmagicIO: DeckLink 8K Pro (1) as blackmagic!io0 [0000:02:00.0]

[17080.520445] BlackmagicIO: Initialised serial driver for blackmagic!io0

[17080.520886] BlackmagicIO: WARNING: "DeckLink 8K Pro" has a firmware version mismatch (valid: 1 device: 0x0028 driver: 0x0030)

[17080.520913] BlackmagicIO: Enabled device "DeckLink 8K Pro (3)" x4/5 Gbps (5953fb2,580201) FW Date: 02-26 15:56

[17080.528019] BlackmagicIO: DeckLink 8K Pro (3) as blackmagic!io1 [0000:02:00.0]

[17080.528020] BlackmagicIO: Initialised serial driver for blackmagic!io1

[17080.528304] BlackmagicIO: WARNING: "DeckLink 8K Pro" has a firmware version mismatch (valid: 1 device: 0x0028 driver: 0x0030)

[17080.528312] BlackmagicIO: Enabled device "DeckLink 8K Pro (2)" x4/5 Gbps (5953fb1,580202) FW Date: 02-26 15:56

[17080.535336] BlackmagicIO: DeckLink 8K Pro (2) as blackmagic!io2 [0000:02:00.0]

[17080.535337] BlackmagicIO: Initialised serial driver for blackmagic!io2

[17080.535628] BlackmagicIO: WARNING: "DeckLink 8K Pro" has a firmware version mismatch (valid: 1 device: 0x0028 driver: 0x0030)

[17080.535637] BlackmagicIO: Enabled device "DeckLink 8K Pro (4)" x4/5 Gbps (5953fb3,580203) FW Date: 02-26 15:56

[17080.542665] BlackmagicIO: DeckLink 8K Pro (4) as blackmagic!io3 [0000:02:00.0]

[17080.542667] BlackmagicIO: Initialised serial driver for blackmagic!io3

Sample log from 'lsmod' command.

```
$ lsmod |grep magic
blackmagic_io         1908736   8
blackmagic             561152   0
```

3.2.1 Decklink Software Update

Run the following command to check firmware status on Decklink card.

```
$ DesktopVideoUpdateTool --all –list
Available Devices:
```

```
DeckLink 8K Pro (G5953fb0)
Firmware is out of date

$ DesktopVideoUpdateTool -update
You must specify which devices to update using --all or -device
Options:
-l [ --list ]                 List devices
-u [ --update ]               Update firmware on the specified
device
-d [ --device ] ID            The identifier of the device to
update
-a [ --all ]                  Update all connected devices
-q [ --quiet ] [=LEVEL(=2)]  Reduce the verbosity of the output
-h [ --help ]                 Display this help message

$ DesktopVideoUpdateTool -d G5953fb0 -u
Updating DeckLink 8K Pro... 100.0%

DeckLink 8K Pro: Please shut down your computer to complete the
software update.

All updates complete
```

Note – If after firmware update the card is not detected, then re-run the dpkg commands to completion and reboot the host. The card will then be detected. Check using 'dmesg'.

3.2.2 Useful Links and Reference

1] Decklink drivers and SDK - https://www.blackmagicdesign.com/support/family/capture-and-playback
2] https://gist.github.com/afriza/879fed4ede539a5a6501e0f046f71463
3] https://trac.ffmpeg.org/wiki/Capture/Blackmagic

4 Compiling ffmpeg with Decklink support

Pre-requisite is a Linux Host installed with T408 and FFmpeg environment and confirmed for transcoding operation.

Proceed with downloading the Decklink SDK, for example, 'Desktop Video 10.11.4 SDK', from Blackmagic support website. Refer Section 3.2.2 for link. The file name will be, for example, Blackmagic_DeckLink_SDK_10.11.4.zip or similar.

Note – SDK 10.11.4 is used as an example in this section. User can download a newer available version.

Unzip the file.

The SDK Linux folder will have 2 sub-folders – include and Samples as follows.

```
$ ls -al ~/'Blackmagic DeckLink SDK 10.11.4'/Linux

total 16

drwxrwxr-x  4 nvme nvme 4096 Sep 30 12:42 .

drwxrwxr-x  6 nvme nvme 4096 Sep 30 12:42 ..

drwxrwxr-x  2 nvme nvme 4096 Sep 30 12:42 include

drwxrwxr-x 15 nvme nvme 4096 Sep 30 12:42 Samples
```

Copy these folders to 'FFmpeg' folder as follows.

First, go to your FFmpeg folder. Then run the following commands.

```
cd ..

mkdir -p FFmpeg/BMD_SDKL10114

cd FFmpeg/BMD_SDKL10114

cp -r ~/'Blackmagic DeckLink SDK 10.11.4'/Linux/include .

cp -r ~/'Blackmagic DeckLink SDK 10.11.4'/Linux/Samples .

ls -al FFmpeg/BMD_SDKL10114

total 16

drwxrwxr-x  4 nvme nvme 4096 Sep 30 12:44 .
```

```
drwxrwxr-x 21 nvme nvme 4096 Sep 30 12:46 ..
drwxrwxr-x  2 nvme nvme 4096 Sep 30 12:43 include
drwxrwxr-x 15 nvme nvme 4096 Sep 30 12:44 Samples
```

Now compile the FFmpeg with Decklink support. The steps can be run on any Linux OS. The environment and commands were verified on an Ubuntu 20.04.2 LTS Linux host.

In the FFmpeg folder, run the following commands to compile and install the ffmpeg. **Note the extra flags added to configure command to enable decklink.**

Note – User can get the configure command dump by running 'bash build_ffmpeg.sh --dry' on command line in the FFmpeg folder.

```
$ ./configure --pkg-config-flags=--static --enable-gpl --enable-nonfree --extra-ldflags=-lm --extra-ldflags=-ldl --enable-libxcoder --enable-pthreads --extra-libs=-lpthread --enable-encoders --enable-decoders --enable-avfilter --enable-muxers --enable-demuxers --enable-parsers --enable-x86asm --disable-debug --disable-ffplay --disable-ffprobe --disable-libx264 --disable-libx265 --disable-cuda-nvcc --disable-cuda --disable-cuvid --disable-nvdec --disable-nvenc --disable-libvmaf --enable-static --disable-shared --extra-cflags=-UNIENC_MULTI_THREAD   --enable-cuvid --extra-cflags=-IBMD_SDKL10114/include --extra-ldflags=-LBMD_SDKL10114/include --enable-decklink

$ make -j

$ sudo make install
```

Check ffmpeg.

```
$  ffmpeg

ffmpeg version 4.2.1 Copyright (c) 2000-2019 the FFmpeg developers

  built with gcc 9 (Ubuntu 9.3.0-17ubuntu1~20.04)

  configuration: --pkg-config-flags=--static --enable-gpl --enable-nonfree --extra-ldflags=-lm --extra-ldflags=-ldl --enable-libxcoder --enable-pthreads --extra-libs=-lpthread --enable-encoders --enable-decoders --enable-avfilter --enable-muxers --enable-demuxers --enable-parsers --enable-x86asm --disable-debug
```

```
--disable-ffplay --disable-ffprobe --disable-libx264 --disable-
libx265 --disable-cuda-nvcc --disable-cuda --disable-cuvid --
disable-nvdec --disable-nvenc --disable-libvmaf --enable-static -
-disable-shared --extra-cflags=-UNIENC_MULTI_THREAD --enable-
cuvid --extra-cflags=-IBMD_SDKL10114/include --extra-ldflags=-
LBMD_SDKL10114/include --enable-decklink

  libavutil      56. 31.100 / 56. 31.100

  libavcodec     58. 54.100 / 58. 54.100

  libavformat    58. 29.100 / 58. 29.100

  libavdevice    58.  8.100 / 58.  8.100

  libavfilter     7. 57.100 /  7. 57.100

  libswscale      5.  5.100 /  5.  5.100

  libswresample   3.  5.100 /  3.  5.100

  libpostproc    55.  5.100 / 55.  5.100

Hyper fast Audio and Video encoder

usage: ffmpeg [options] [[infile options] -i infile]... {[outfile
options] outfile}...


Use -h to get full help or, even better, run 'man ffmpeg'
```

5 ffmpeg commands using decklink

```
$ ffmpeg -hide_banner -sources decklink
```

```
Auto-detected sources for decklink:

88:05953fb0:00000000 [DeckLink 8K Pro (1)]

88:05953fb1:00000000 [DeckLink 8K Pro (2)]

88:05953fb2:00000000 [DeckLink 8K Pro (3)]

88:05953fb3:00000000 [DeckLink 8K Pro (4)]
```

```
$ ffmpeg -hide_banner -f decklink -list_formats 1 -i
88:05953fb3:00000000
```

```
[decklink @ 0x55dadb863280] Supported formats for
'88:05953fb3:00000000':

format_code     description

ntsc            720x486 at 30000/1001 fps (interlaced, lower
field first)

pal             720x576 at 25000/1000 fps (interlaced, upper
field first)

23ps            1920x1080 at 24000/1001 fps

24ps            1920x1080 at 24000/1000 fps

Hp25            1920x1080 at 25000/1000 fps

Hp29            1920x1080 at 30000/1001 fps

Hp30            1920x1080 at 30000/1000 fps

Hp47            1920x1080 at 48000/1001 fps

Hp48            1920x1080 at 48000/1000 fps

Hp50            1920x1080 at 50000/1000 fps

Hp59            1920x1080 at 60000/1001 fps

Hp60            1920x1080 at 60000/1000 fps

Hp95            1920x1080 at 96000/1001 fps
```

```
Hp96            1920x1080 at 96000/1000 fps
Hp10            1920x1080 at 100000/1000 fps
Hp11            1920x1080 at 120000/1001 fps
Hp12            1920x1080 at 120000/1000 fps
Hi50            1920x1080 at 25000/1000 fps (interlaced, upper
field first)
Hi59            1920x1080 at 30000/1001 fps (interlaced, upper
field first)
Hi60            1920x1080 at 30000/1000 fps (interlaced, upper
field first)
hp50            1280x720 at 50000/1000 fps
hp59            1280x720 at 60000/1001 fps
hp60            1280x720 at 60000/1000 fps
2d23            2048x1080 at 24000/1001 fps
2d24            2048x1080 at 24000/1000 fps
2d25            2048x1080 at 25000/1000 fps
2d29            2048x1080 at 30000/1001 fps
2d30            2048x1080 at 30000/1000 fps
2d47            2048x1080 at 48000/1001 fps
2d48            2048x1080 at 48000/1000 fps
2d50            2048x1080 at 50000/1000 fps
2d59            2048x1080 at 60000/1001 fps
2d60            2048x1080 at 60000/1000 fps
2d95            2048x1080 at 96000/1001 fps
2d96            2048x1080 at 96000/1000 fps
2d10            2048x1080 at 100000/1000 fps
2d11            2048x1080 at 120000/1001 fps
2d12            2048x1080 at 120000/1000 fps
```

```
4k23            3840x2160 at 24000/1001 fps
4k24            3840x2160 at 24000/1000 fps
4k25            3840x2160 at 25000/1000 fps
4k29            3840x2160 at 30000/1001 fps
4k30            3840x2160 at 30000/1000 fps
4k47            3840x2160 at 48000/1001 fps
4k48            3840x2160 at 48000/1000 fps
4k50            3840x2160 at 50000/1000 fps
4k59            3840x2160 at 60000/1001 fps
4k60            3840x2160 at 60000/1000 fps
4d23            4096x2160 at 24000/1001 fps
4d24            4096x2160 at 24000/1000 fps
4d25            4096x2160 at 25000/1000 fps
4d29            4096x2160 at 30000/1001 fps
4d30            4096x2160 at 30000/1000 fps
4d47            4096x2160 at 48000/1001 fps
4d48            4096x2160 at 48000/1000 fps
4d50            4096x2160 at 50000/1000 fps
4d59            4096x2160 at 60000/1001 fps
4d60            4096x2160 at 60000/1000 fps
88:05953fb3:00000000: Immediate exit requested
```

5.1 Encoding using T408 and Decklink

Sample command for ffmpeg capturing the decklink input stream and encoding it using T408 HEVC Encoder.

```
$ sudo ffmpeg -hide_banner -f decklink -i 88:05953fb0:00000000 -
format_code Hp60 -pix_fmt yuv420p -r 60 -c:v h265_ni_enc -enc 0 -
xcoder-params RcEnable=1:bitrate=6000000 -f null -
```

```
[decklink @ 0x5595425f4580] Autodetected the input mode
[decklink @ 0x5595425f4580] Found Decklink mode 1920 x 1080 with
rate 30.00
Guessed Channel Layout for Input Stream #0.0 : stereo
Input #0, decklink, from '88:05953fb0:00000000':
  Duration: N/A, start: 0.000000, bitrate: 996864 kb/s
    Stream #0:0: Audio: pcm_s16le, 48000 Hz, stereo, s16, 1536
kb/s
    Stream #0:1: Video: rawvideo (UYVY / 0x59565955),
uyvy422(progressive), 1920x1080, 995328 kb/s, 30 tbr, 1000k tbn,
1000k tbc
Stream mapping:
  Stream #0:1 -> #0:0 (rawvideo (native) -> hevc (h265_ni_enc))
  Stream #0:0 -> #0:1 (pcm_s16le (native) -> pcm_s16le (native))
Press [q] to stop, [?] for help
[h265_ni_enc @ 0x55954263a980] Session state: 0 allocate frame
fifo.
[h265_ni_enc @ 0x55954263a980] dts offset: 2, gop_offset_count: 0
Output #0, null, to 'pipe:':
  Metadata:
    encoder         : Lavf58.29.100
    Stream #0:0: Video: hevc (h265_ni_enc), yuv420p(progressive),
1920x1080, q=2-31, 6000 kb/s, 60 fps, 60 tbn, 60 tbc
    Metadata:
      encoder         : Lavc58.54.100 h265_ni_enc
    Stream #0:1: Audio: pcm_s16le, 48000 Hz, stereo, s16, 1536
kb/s
    Metadata:
      encoder         : Lavc58.54.100 pcm_s16le
frame= 3210 fps= 30 q=-0.0 size=N/A time=00:01:47.00 bitrate=N/A
speed=   1x
```

6 Decklink Environment - Windows

6.1 Scope

This section details the installation of Decklink drivers and SDK on a Windows Host.

6.2 Installation and Check

6.2.1 Decklink Application

Download the Blackmagic Decklink software update Windows package from Blackmagic support website. Refer Section 3.2.2 for link. File will be titled like Blackmagic_Desktop_Video_Windows_12.1.zip or similar.

Run the ‘Desktop Video Installer v12.1.msi’ installer to completion. Reboot the Windows host.

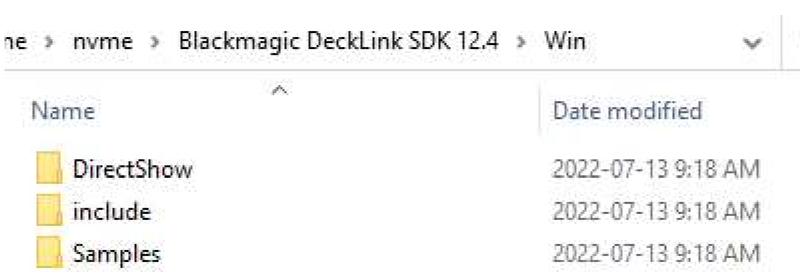
Confirm in Windows Control Panel → Device Manager that the Decklink card is detected and installed fine.

6.2.2 Decklink SDK

Pre-requisite is a Windows Host installed with T408 and FFmpeg environment, and confirmed for transcoding operation. Refer app note APPS024 – Codensity T408 Windows msys2 Setup Application Note, for details.

Run the following steps to get ffmpeg with Decklink support.

Download a Decklink SDK version from link provided in Section 3.2.2. Untar the package. It will have a sub-folder ‘Win’. Copy this folder to the FFmpeg folder on your Windows host.



Open a mingw64 terminal window with administrator privileges.

```
cd ~
cp -rpf ~/"Blackmagic DeckLink SDK 12.4"/Win  ~/FFmpeg/BMD_SDK124
```

Note – we are copying the SDK folder to FFmpeg folder as 'BMD_SDK124'. User can name it anything but will need to make it corresponding to the configure command example below.

Run the following mandatory step,

```
cd ~/FFmpeg/BMD_SDK124/include
widl -h -u DeckLinkAPI.idl
```

Confirm the following two files are generated,

```
C:\msys64\home\nvme\FFmpeg44\BMD_SDK1240\include>ls -al DeckLinkAPI.h DeckLinkAPI_i.c
-rw-r--r-- 1 nvme None 901100 Aug  8 13:05 DeckLinkAPI.h
-rw-r--r-- 1 nvme None  17160 Aug  8 13:05 DeckLinkAPI_i.c
```

Next steps will help compile the FFmpeg.

```
cd ~/FFmpeg
bash build_ffmpeg.sh -w -dry
```

This will provide a dump like,

```
C:\msys64\home\nvme\FFmpeg44>bash build_ffmpeg.sh -w --dry
./configure --enable-cross-compile --arch=x86_64 --target-os=mingw32 --pkg-config-flags=--static --enable-gpl --enable-nonfre
e --extra-ldflags=-lm --enable-libxcoder --enable-ni --enable-w32threads --extra-libs=-lwinpthread -lws2_32 --enable-encoders
 --enable-decoders --enable-avfilter --enable-muxers --enable-demuxers --enable-parsers --enable-x86asm --disable-debug --dis
able-ffplay --disable-ffprobe --disable-libx264 --disable-libx265 --disable-cuda-nvcc --disable-cuda --disable-cuvid --disabl
e-nvdec --disable-nvenc --disable-libvmaf --enable-static --disable-shared --extra-cflags=-UNIENC_MULTI_THREAD --extra-cflags
=-UNI_DEC_GSTREAMER_SUPPORT
```

Append Decklink specific arguments (in bold below) to the configure command and run as follows,

bash configure --enable-cross-compile --arch=x86_64 --target-os=mingw32 --pkg-config-flags=--static --enable-gpl --enable-nonfree --extra-ldflags=-lm --enable-libxcoder --enable-ni --enable-w32threads --extra-libs='-lwinpthread -lws2_32' --enable-encoders --enable-decoders --enable-avfilter --enable-muxers --enable-demuxers --enable-parsers --enable-x86asm --disable-debug --disable-ffplay --disable-ffprobe --disable-libx264 --disable-libx265 --disable-cuda-nvcc --disable-cuda --disable-cuvid --disable-nvdec --disable-nvenc --disable-libvmaf --enable-static --disable-shared --extra-cflags=-UNIENC_MULTI_THREAD --extra-cflags=-UNI_DEC_GSTREAMER_SUPPORT THREAD **--extra-cflags=-IBMD_SDK124/include --extra-ldflags=-LBMD_SDK124/include --enable-decklink**

Note – ensure to check the quotations in above command for `--extra-libs='-lwinpthread -lws2_32'`

Next, run the following two commands:

```
make -j

make install
```

The user can confirm that ffmpeg is compiled with decklink support by running 'ffmpeg' command as follows.

```
C:\msys64\home\nvme\FFmpeg44>ffmpeg
ffmpeg version 4.4 Copyright (c) 2000-2021 the FFmpeg developers
  built with gcc 12.1.0 (Rev3, Built by MSYS2 project)
  configuration: --enable-cross-compile --arch=x86_64 --target-
os=mingw32 --pkg-config-flags=--static --enable-gpl --enable-
nonfree --extra-ldflags=-lm --enable-libxcoder --enable-ni --
enable-w32threads --extra-libs='-lwinpthread -lws2_32' --extra-
cflags=-IBMD_SDK1240/include --extra-ldflags=-
LBMD_SDK1240/include --enable-decklink --enable-encoders --
enable-decoders --enable-avfilter --enable-muxers --enable-
demuxers --enable-parsers --enable-x86asm --disable-debug --
disable-ffplay --disable-ffprobe --disable-libx264 --disable-
libx265 --disable-cuda-nvcc --disable-cuda --disable-cuvid --
disable-nvdec --disable-nvenc --disable-libvmaf --enable-static -
-disable-shared --extra-cflags=-UNIENC_MULTI_THREAD --extra-
cflags=-UNI_DEC_GSTREAMER_SUPPORT
  libavutil      56. 70.100 / 56. 70.100
  libavcodec     58.134.100 / 58.134.100
  libavformat    58. 76.100 / 58. 76.100
  libavdevice    58. 13.100 / 58. 13.100
  libavfilter     7.110.100 /  7.110.100
  libswscale      5.  9.100 /  5.  9.100
  libswresample   3.  9.100 /  3.  9.100
  libpostproc    55.  9.100 / 55.  9.100
Hyper fast Audio and Video encoder
usage: ffmpeg [options] [[infile options] -i infile]... {[outfile
options] outfile}...
```

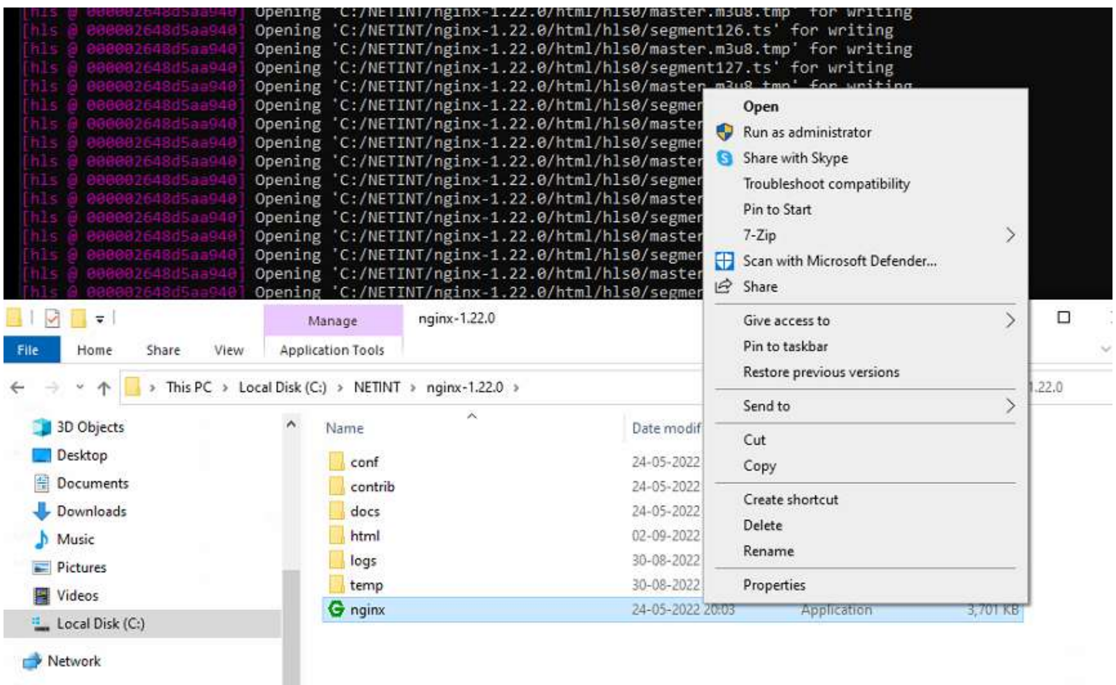
User can now run the commands in Section 5 to trial ffmpeg with decklink support.

7 Appendix A – ffmpeg streaming example with decklink input

Connect a suitable SDI input to the Decklink card and then run the ffmpeg command example below to stream the encoded output to, say, HLS. A user can then use a player like VLC media player to view the network stream.

Note – configuration of Blackmagic Decklink hardware is not in the scope of this document.

Pre-requisite:
1] Download and run a Nginx server on the host. Link - http://nginx.org/en/download.html .
Unzip the Nginx download and then run the 'nginx' application as administrator.



2] Create a folder 'hls0' in the nginx->html folder, for example, nginx-1.22.0/html/hls0 folder in above snapshot.

Run the following ffmpeg command to begin streaming:

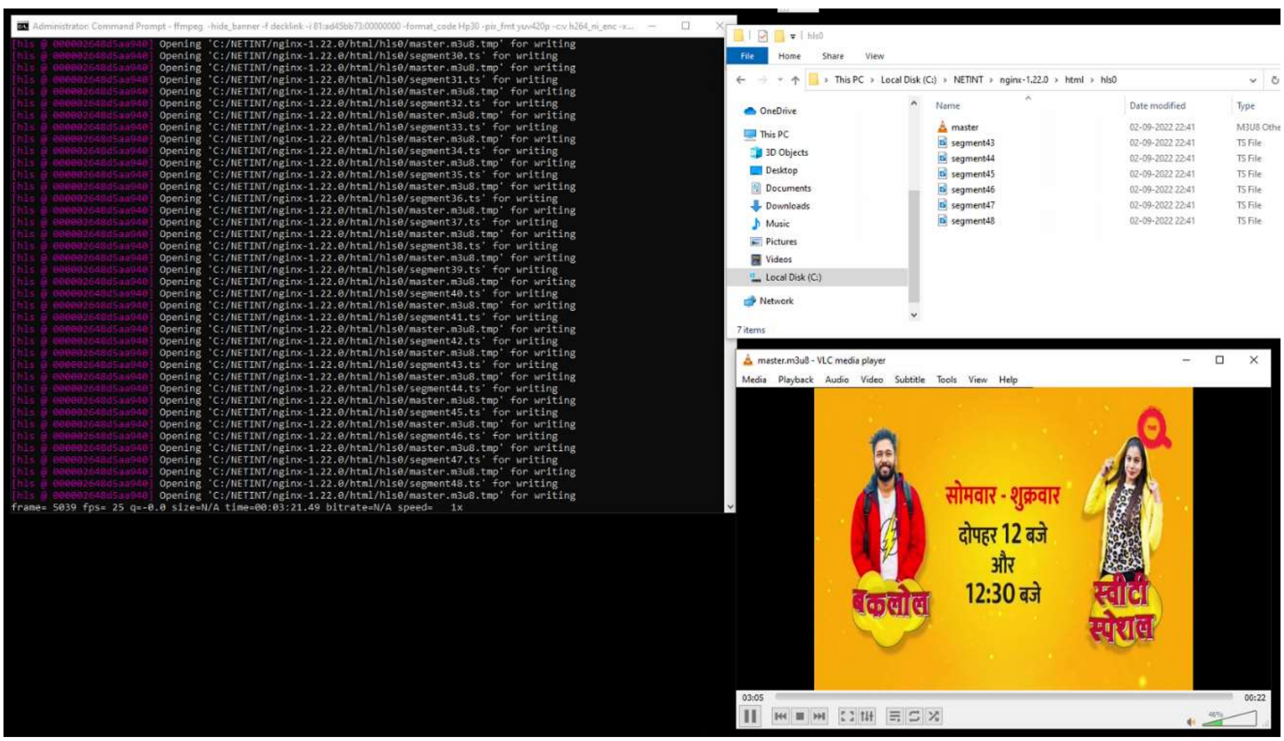
```
ffmpeg -hide_banner -f decklink -i 81:ad45bb73:00000000 -
format_code Hp30 -pix_fmt yuv420p -c:v h264_ni_enc -xcoder-params
bitrate=6000000:RcEnable=1 -f hls -hls_segment_filename
C:/NETINT/nginx-1.22.0/html/hls0/segment%d.ts -hls_time 4 -
master_pl_name master.m3u8 -hls_flags delete_segments
C:/NETINT/nginx-1.22.0/html/hls0/master.m3u8
```

where,
-i 81:ad45bb73:00000000 is the decklink card port where an active SDI signal is connected. Modify this value to the correct value. To get decklink port IDs refer commands in Section 5.

-format_code Hp30 this value should match the resolution of the SDI input signal

Note – update the hls0 paths in above example to match your host nginx installation

Next, open a VLC player to view the network stream. General format is
http://<ip address>/hls0/master.m3u8



8 Revision History

Version	Change	Create Date	Author
0.1	Document Created	September 30, 2021	Balwinder Sidhu
1.0	Issued Version	September 30, 2021	Balwinder Sidhu
2.0	Update as per new app note template	December 10, 2021	Balwinder Sidhu
3.0	Update Section 6 for Windows host	August 8, 2022	Balwinder Sidhu
4.0	Add Appendix A on ffmpeg streaming example using decklink input	September 6, 2022	Balwinder Sidhu

9 Legal Notice

Information in this document is provided in connection with NETINT products. No license, express or implied, by estoppel or otherwise, to any intellectual property rights is granted by this document. Except as provided in NETINT's terms and conditions of sale for such products, NETINT assumes no liability whatsoever and NETINT disclaims any express or implied warranty, relating to sale and/or use of NETINT products including liability or warranties relating to fitness for a particular purpose, merchantability, or infringement of any patent, copyright or other intellectual property right.

A "Mission Critical Application" is any application in which failure of the NETINT Product could result, directly or indirectly, in personal injury or death. Should you purchase or use NETINT's products for any such mission critical application, you shall indemnify and hold NETINT and its subsidiaries, subcontractors and affiliates, and the directors, officers, and employees of each, harmless against all claims costs, damages, and expenses and reasonable attorney's fees arising out of, directly or indirectly, any claim of product liability, personal injury, or death arising in any way out of such mission critical application, whether or not NETINT or its subcontractor was negligent in the design, manufacture, or warning of the NETINT product or any of its parts.

NETINT may make changes to specifications, technical documentation, and product descriptions at any time, without notice. The information here is subject to change without notice. Do not finalize a design with this information. The products described in this document may contain design defects or errors known as errata which may cause the product to deviate from published specifications.

NETINT, Codensity, and NETINT Logo are trademarks of NETINT Technologies Inc. All other trademarks or registered trademarks are the property of their respective owners.