

Shanghai Naxing Electronics Co., Ltd.

Panel-Link

USB Media Stream Transport Protocol Specification

Rev. 1.0



Contents

1. Overview.....	- 4 -
2. Panel-Link Protocol	- 5 -
Panel-Link Media Stream	- 5 -
Stream Tag	- 5 -
Typical Stream Tag for Formatted Media Stream	- 6 -
Stream Tag for Continuous PNG Media Stream.....	- 7 -
Stream Tag for Continuous JPEG Media Stream	- 7 -
Typical Stream Tag for Raw Media Stream	- 8 -
Stream Payload	- 9 -
3. Raw Media Types	- 10 -
Metadata	- 11 -
Format Description	- 11 -
4. Formatted Media Types	- 50 -
5. Endianness Declaration	- 52 -
6. 16Bits Checksum Algorithm.....	- 53 -
7. Contact Information.....	- 54 -
8. Revision History	- 54 -

List of Tables

2-1 Stream Tag Layout	- 5 -
2-2 Stream Tag Field Description	- 6 -
2-3 Formatted Media Tag	- 6 -
2-4 PNG Media Stream Tag	- 7 -
2-5 JPEG Media Stream Tag	- 7 -
2-6 Raw Media Tag	- 9 -
4-1 Formatted Media Types	- 51 -

1. Overview

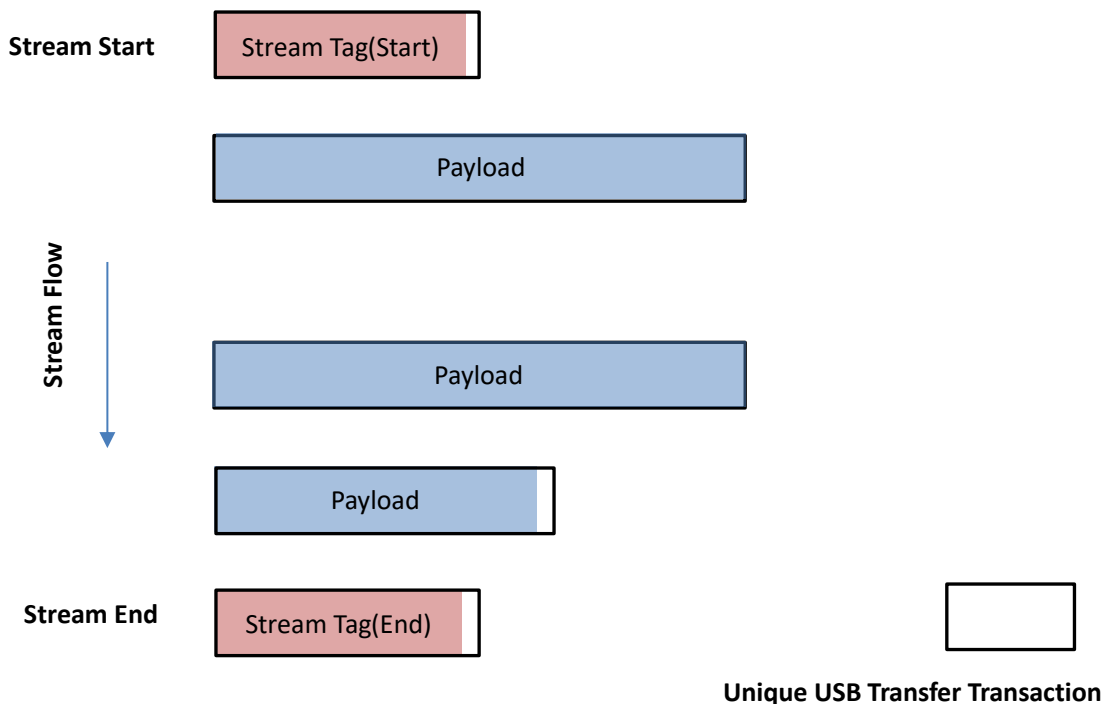
Panel-Link is a streaming media transport protocol to transfer various streaming medias between USB host and device. Panel-Link does not restrict the format of streaming media (pictures, videos, sounds). Through its up layer application (host side or device side), user can access the streaming media directly without knowing the details of the underlying USB transmission.

Copyright and interpretation of this Agreement shall be vested in Shanghai Naxing Electronics Co., Ltd.

2. Panel-Link Protocol

Panel-Link Media Stream

Panel-Link media stream is transmitted in a uni-direction USB pipe. It can be transferred from USB Host to USB device or vice versa.



Stream Tag

Stream Tag will occupy a unique USB transfer transaction when it is being transmitted. In Panel-Link version 1, the length of a stream tag package is 270 bytes. Stream Tag layout is described as below:

Byte Sequence	High Byte	Low Byte
0-9	Protocol String	
10-11	Tag Type	Version
12-267	Format String	
268-269	16 Bits Checksum	

2-1 Stream Tag Layout

Detailed descriptions of each field are shown in the following table:

Field	Description
Protocol String	Constant ASCII String "PANEL-LINK"
Tag Type	1 – Legacy Command 1 2 – Legacy Command 2 3 – Reset of Display 4 – Screen Clear 5 – Start of a Media Stream 6 – End of a Media Stream
Version	1 – Current version
Format String	0 – default, if payload in Formatted Media type ¹ ASCII string - Tag for payload of unstructured and uncompressed Raw Media ²
16 Bits Checksum	16Bits Checksum ³

2-2 Stream Tag Field Description

Note 1: Refer to Section 4 for details of Formatted Media type

Note 2: Refer to Section 3 for details of Raw Media type

Note 3: Refer to Section 6 for detail of 16Bits Checksum Algorithm

Typical Stream Tag for Formatted Media Stream

Byte Sequence	Stream Start Tag		Stream End Tag	
	High Byte	Low Byte	High Byte	Low Byte
0-1	0x41	0x50	0x41	0x50
2-3	0x45	0x4e	0x45	0x4e
4-5	0x2d	0x4c	0x2d	0x4c
6-7	0x49	0x4c	0x49	0x4c
8-9	0x4b	0x4e	0x4b	0x4e
10-11	0x5	0x1	0x6	0x1
12-267	0x0	0x0	0x0	0x0
268-269	0xb2	0x79	0xb1	0x79

2-3 Formatted Media Tag

Since up layer media player has it's own typefinder so no format string(byte 12 to byte 267) needed when Panel-Link transmitting a well-known media stream¹. But host still need to provide a format string if media stream is in a so called 'un-well-known' format. We will discuss these special scenarios (PNG Media Stream and JPEG Media Stream) in next sections.

Note 1: Refer to Section 4 for details of Formatted Media type

Stream Tag for Continuous PNG Media Stream

Format string of this media stream is “image/png”

Byte Sequence	Stream Start Tag		Stream End Tag	
	High Byte	Low Byte	High Byte	Low Byte
0-1	0x41	0x50	0x41	0x50
2-3	0x45	0x4e	0x45	0x4e
4-5	0x2d	0x4c	0x2d	0x4c
6-7	0x49	0x4c	0x49	0x4c
8-9	0x4b	0x4e	0x4b	0x4e
10-11	0x5	0x1	0x6	0x1
12-13	0x6d	0x69	0x00	0x00
14-15	0x67	0x61	0x00	0x00
16-17	0x2f	0x65	0x00	0x00
18-19	0x6e	0x70	0x00	0x00
20-21	0x00	0x67	0x00	0x00
22-267	0x00	0x00	0x00	0x00
268-269	0x3f	0x72	0xb1	0x79

2-4 PNG Media Stream Tag

With this format string, consequent PNG pictures can be transmitted in one Panel-Link session.

Stream Tag for Continuous JPEG Media Stream

Format string of this media stream is “image/jpeg”

Byte Sequence	Stream Start Tag		Stream End Tag	
	High Byte	Low Byte	High Byte	Low Byte
0-1	0x41	0x50	0x41	0x50
2-3	0x45	0x4e	0x45	0x4e
4-5	0x2d	0x4c	0x2d	0x4c
6-7	0x49	0x4c	0x49	0x4c
8-9	0x4b	0x4e	0x4b	0x4e
10-11	0x5	0x1	0x6	0x1
12-13	0x6d	0x69	0x00	0x00
14-15	0x67	0x61	0x00	0x00
16-17	0x2f	0x65	0x00	0x00
18-19	0x70	0x6a	0x00	0x00
20-21	0x67	0x65	0x00	0x00
22-267	0x00	0x00	0x00	0x00
268-269	0xd6	0x79	0xb1	0x79

2-5 JPEG Media Stream Tag

With this format string, consequent JPEG pictures can be transmitted in one Panel-Link session.

Typical Stream Tag for Raw Media Stream

Format string of this raw media is “video/x-raw, format=RGB16, height=480, width=800, framerate=0/1”

Byte Sequence	Stream Start Tag		Stream End Tag	
	High Byte	Low Byte	High Byte	Low Byte
0-1	0x41	0x50	0x41	0x50
2-3	0x45	0x4e	0x45	0x4e
4-5	0x2d	0x4c	0x2d	0x4c
6-7	0x49	0x4c	0x49	0x4c
8-9	0x4b	0x4e	0x4b	0x4e
10-11	0x5	0x1	0x6	0x1
12-13	0x69	0x76	0x00	0x00
14-15	0x65	0x64	0x00	0x00
16-17	0x2f	0x6f	0x00	0x00
18-19	0x2d	0x78	0x00	0x00
20-21	0x61	0x72	0x00	0x00
22-23	0x2c	0x77	0x00	0x00
24-25	0x66	0x20	0x00	0x00
26-27	0x72	0x6f	0x00	0x00
28-29	0x61	0x6d	0x00	0x00
30-31	0x3d	0x74	0x00	0x00
32-33	0x47	0x52	0x00	0x00
34-35	0x31	0x42	0x00	0x00
36-37	0x2c	0x36	0x00	0x00
38-39	0x68	0x20	0x00	0x00
40-41	0x69	0x65	0x00	0x00
42-43	0x68	0x67	0x00	0x00
44-45	0x3d	0x74	0x00	0x00
46-47	0x38	0x34	0x00	0x00
48-49	0x2c	0x30	0x00	0x00
50-51	0x77	0x20	0x00	0x00
52-53	0x64	0x69	0x00	0x00
54-55	0x68	0x74	0x00	0x00
56-57	0x38	0x3d	0x00	0x00
58-59	0x30	0x30	0x00	0x00
60-61	0x20	0x2c	0x00	0x00
62-63	0x72	0x66	0x00	0x00
64-65	0x6d	0x61	0x00	0x00
66-67	0x72	0x65	0x00	0x00
68-69	0x74	0x61	0x00	0x00
70-71	0x3d	0x65	0x00	0x00

72-73	0x2f	0x30	0x00	0x00
74-75	0x00	0x31	0x00	0x00
76-267	0x00	0x00	0x00	0x00
268-269	0x00	0xdf	0xb1	0x79

2-6 Raw Media Tag

Consequent raw video data can be transmitted in one Panel-Link session(between Stream Start/Stream End).

Stream Payload

Payload of Panel-Link stream can be contained in one or more USB transfer transactions.

3. Raw Media Types

Supported formats of unstructured and uncompressed raw video data:

video/x-raw

- **width**, `G_TYPE_INT`: Mandatory. The width of the image in pixels.
- **height**, `G_TYPE_INT`: Mandatory. The height of the image in pixels
- **framerate**, `GST_TYPE_FRACTION`: Default 0/1. The framerate of the video, 0/1 for variable framerate
- **max-framerate**, `GST_TYPE_FRACTION`: Default as framerate. For variable framerates this would be the maximum expected framerate. This value is only valid when the framerate is set to "variable" (0/1)
- **views**, `G_TYPE_INT`: Default 1. The number of views for multiview video. Each buffer contains multiple `GstVideoMeta` buffers that describe each view. use the frame id to get access to the different views.
- **interlace-mode**, `G_TYPE_STRING`: Default progressive. The interlace mode (also see the [interlaced video design docs](#)). The following values are possible:
 - *"progressive"*: all frames are progressive
 - *"interleaved"*: 2 fields are interleaved in one video frame. Extra buffer flags describe the field order.
 - *"mixed"*: progressive and interleaved frames, extra buffer flags describe the frame and fields.
 - *"fields"*: 2 fields are stored in one buffer. Use the frame ID to get access to the required field. For multiview (the 'views' property > 1) the fields of view N can be found at frame ID (N * 2) and (N * 2) + 1. Each view has only half the amount of lines as noted in the height property, pads specifying the "fields" property must be prepared for this. This mode requires multiple `GstVideoMeta` metadata to describe the fields.
 - *"alternate"*: one field per buffer, with buffer flags indicating whether the field is the top field (`GST_VIDEO_BUFFER_FLAG_TOP_FIELD`) or bottom field (`GST_VIDEO_BUFFER_FLAG_BOTTOM_FIELD`). Top and bottom fields are expected to alternate in this mode. Caps for this interlace mode must also carry a `format:Interlaced` caps feature (`GST_CAPS_FEATURE_FORMAT_INTERLACED`) to ensure backwards compatibility for the new mode (Since: 1.16)
- **chroma-site**, `G_TYPE_STRING`: Default `GST_VIDEO_CHROMA_SITE_UNKNOWN`. The chroma siting of the video frames.
 - *"jpeg"*: `GST_VIDEO_CHROMA_SITE_JPEG`
 - *"mpeg2"*: `GST_VIDEO_CHROMA_SITE_MPEG2`
 - *"dv"*: `GST_VIDEO_CHROMA_SITE_DV`

- ...
- `"none"`: `GST_VIDEO_CHROMA_SITE_NONE`
- **colorimetry**, `G_TYPE_STRING`: Default `GST_VIDEO_COLORIMETRY_UNKNOWN`. The colorimetry of the video frames. Predefined colorimetry is given with the following values:
 - `"bt601"`
 - `"bt709"`
 - `"smpte240m"`
- **pixel-aspect-ratio**, `GST_TYPE_FRACTION`: Default 1/1. The pixel aspect ration of the video
- **format**, `G_TYPE_STRING`: Mandatory. The format of the video. See the Formats section for a list of valid format strings.

Metadata

- `GstVideoMeta` contains the description of one video field or frame. It has stride support and support for having multiple memory regions per frame. Multiple `GstVideoMeta` can be added to a buffer and can be identified with a unique id. This id can be used to select fields in interlaced formats or views in multiview formats.
- `GstVideoCropMeta` contains the cropping region of the video.

Format Description

- **"I420"** planar 4:2:0 YUV

Component 0: Y

```
depth:          8
pstride:        1
default offset: 0
default rstride: RU4 (width)
default size:   rstride (component0) * RU2 (height)
```

Component 1: U

```
depth:          8
pstride:        1
default offset: size (component0)
default rstride: RU4 (RU2 (width) / 2)
```

```
default size:  rstride (component1) * RU2 (height) / 2
```

Component 2: V

```
depth          8
```

```
pstride:       1
```

```
default offset: offset (component1) + size (component1)
```

```
default rstride: RU4 (RU2 (width) / 2)
```

```
default size:  rstride (component2) * RU2 (height) / 2
```

Image

```
default size: size (component0) +
```

```
              size (component1) +
```

```
              size (component2)
```

- **"YV12"** planar 4:2:0 YUV

- Same as I420 but with U and V planes swapped

Component 0: Y

```
depth:         8
```

```
pstride:       1
```

```
default offset: 0
```

```
default rstride: RU4 (width)
```

```
default size:  rstride (component0) * RU2 (height)
```

Component 1: U

```
depth          8
```

```
pstride:       1
```

```
default offset: offset (component2) + size (component2)
```

```
default rstride: RU4 (RU2 (width) / 2)
```

```
default size:  rstride (component1) * RU2 (height) / 2
```

Component 2: V

depth: 8

pstride: 1

default offset: size (component0)

default rstride: RU4 (RU2 (width) / 2)

default size: rstride (component2) * RU2 (height) / 2

Image

default size: size (component0) +

size (component1) +

size (component2)

- "YUY2" packed 4:2:2 YUV

```
+---+---+---+---+ +---+---+---+---+
```

```
|Y0|U0|Y1|V0| |Y2|U2|Y3|V2| ...
```

```
+---+---+---+---+ +---+---+---+---+
```

Component 0: Y

depth: 8

pstride: 2

offset: 0

Component 1: U

depth: 8

offset: 1

pstride: 4

Component 2: V

depth: 8

offset: 3

```
pstride:      4
```

Image

```
default rstride: RU4 (width * 2)
```

```
default size:  rstride (image) * height
```

- **"YVYU"** packed 4:2:2 YUV

- Same as "YUY2" but with U and V planes swapped

```
+---+---+---+---+ +---+---+---+---+
```

```
|Y0|V0|Y1|U0| |Y2|V2|Y3|U2| ...
```

```
+---+---+---+---+ +---+---+---+---+
```

Component 0: Y

```
depth:      8
```

```
pstride:    2
```

```
offset:     0
```

Component 1: U

```
depth:      8
```

```
pstride:    4
```

```
offset:     3
```

Component 2: V

```
depth      8
```

```
pstride:   4
```

```
offset:    1
```

Image

```
default rstride: RU4 (width * 2)
```

```
default size:  rstride (image) * height
```

- **"UYVY"** packed 4:2:2 YUV

```
+-----+-----+ +-----+-----+
```

```
|U0|Y0|V0|Y1| |U2|Y2|V2|Y3| ...
```

```
+-----+-----+ +-----+-----+
```

Component 0: Y

depth: 8

pstride: 2

offset: 1

Component 1: U

depth: 8

pstride: 4

offset: 0

Component 2: V

depth: 8

pstride: 4

offset: 2

Image

default rstride: RU4 (width * 2)

default size: rstride (image) * height

- **"AYUV"** packed 4:4:4 YUV with alpha channel

```
+-----+-----+ +-----+-----+
```

```
|A0|Y0|U0|V0| |A1|Y1|U1|V1| ...
```

```
+-----+-----+ +-----+-----+
```

Component 0: Y

```
depth:      8
pstride:    4
offset:     1
```

Component 1: U

```
depth:      8
pstride:    4
offset:     2
```

Component 2: V

```
depth      8
pstride:   4
offset:    3
```

Component 3: A

```
depth      8
pstride:   4
offset:    0
```

Image

```
default rstride: width * 4
```

```
default size:  rstride (image) * height
```

- **"RGBx"** sparse rgb packed into 32 bit, space last

```
+---+---+---+---+ +---+---+---+---+
|R0|G0|B0|X | |R1|G1|B1|X | ...
+---+---+---+---+ +---+---+---+---+
```



```

Component 0: R
    depth:      8
    pstride:    4
    offset:     0

Component 1: G
    depth:      8
    pstride:    4
    offset:     1

Component 2: B
    depth      8
    pstride:   4
    offset:    2

Image
    default rstride: width * 4
    default size:   rstride (image) * height
    
```

- **"BGRx"** sparse reverse rgb packed into 32 bit, space last

```

+-----+-----+
|B0|G0|R0|X | |B1|G1|R1|X | ...
+-----+-----+

Component 0: R
    depth:      8
    pstride:    4
    offset:     2

Component 1: G
    
```

```

depth:      8

pstride:    4

offset:     1

Component 2: B

depth       8

pstride:    4

offset:     0

Image

default rstride: width * 4

default size:  rstride (image) * height

```

- **"xRGB"** sparse rgb packed into 32 bit, space first

```

+-----+-----+
|X |R0|G0|B0| |X |R1|G1|B1| ...
+-----+-----+

```

```

Component 0: R

depth:      8

pstride:    4

offset:     1

```

```

Component 1: G

depth:      8

pstride:    4

offset:     2

```

```

Component 2: B

depth       8

```

```
pstride:      4
```

```
offset:      3
```

Image

```
default rstride: width * 4
```

```
default size:  rstride (image) * height
```

- **"xBGR"** sparse reverse rgb packed into 32 bit, space first

```
+---+---+---+---+ +---+---+---+---+
|X |B0|G0|R0| |X |B1|G1|R1| ...
+---+---+---+---+ +---+---+---+---+
```

Component 0: R

```
depth:      8
```

```
pstride:    4
```

```
offset:    3
```

Component 1: G

```
depth:      8
```

```
pstride:    4
```

```
offset:    2
```

Component 2: B

```
depth      8
```

```
pstride:   4
```

```
offset:   1
```

Image

```
default rstride: width * 4
```

```
default size:  rstride (image) * height
```

- **"RGBA"** rgb with alpha channel last

```
+-----+-----+ +-----+-----+
```

```
|R0|G0|B0|A0| |R1|G1|B1|A1| ...
```

```
+-----+-----+ +-----+-----+
```

Component 0: R

depth: 8

pstride: 4

offset: 0

Component 1: G

depth: 8

pstride: 4

offset: 1

Component 2: B

depth 8

pstride: 4

offset: 2

Component 3: A

depth 8

pstride: 4

offset: 3

Image

default rstride: width * 4

default size: rstride (image) * height

- **"BGRA"** reverse rgb with alpha channel last

```
+-----+-----+

```

```
|B0|G0|R0|A0| |B1|G1|R1|A1| ...

```

```
+-----+-----+

```

Component 0: R

depth: 8

pstride: 4

offset: 2

Component 1: G

depth: 8

pstride: 4

offset: 1

Component 2: B

depth 8

pstride: 4

offset: 0

Component 3: A

depth 8

pstride: 4

offset: 3

Image

default rstride: width * 4

default size: rstride (image) * height

- **"ARGB"** rgb with alpha channel first

```
+-----+-----+

```

```
|A0|R0|G0|B0| |A1|R1|G1|B1| ...
```

```
+---+---+---+---+ +---+---+---+---+
```

Component 0: R

depth: 8

pstride: 4

offset: 1

Component 1: G

depth: 8

pstride: 4

offset: 2

Component 2: B

depth 8

pstride: 4

offset: 3

Component 3: A

depth 8

pstride: 4

offset: 0

Image

default rstride: width * 4

default size: rstride (image) * height

- **"ABGR"** reverse rgb with alpha channel first

```
+---+---+---+---+ +---+---+---+---+
```

```
|A0|R0|G0|B0| |A1|R1|G1|B1| ...
```

```
+-----+ +-----+
```

Component 0: R

```
depth:      8
pstride:    4
offset:     1
```

Component 1: G

```
depth:      8
pstride:    4
offset:     2
```

Component 2: B

```
depth      8
pstride:   4
offset:    3
```

Component 3: A

```
depth      8
pstride:   4
offset:    0
```

Image

```
default rstride: width * 4
```

```
default size:  rstride (image) * height
```

- **"RGB"** rgb

```
+-----+ +-----+
```

```
|R0|G0|B0| |R1|G1|B1| ...
```

```
+-----+ +-----+
```

Component 0: R

depth: 8
pstride: 3
offset: 0

Component 1: G

depth: 8
pstride: 3
offset: 1

Component 2: B

depth: 8
pstride: 3
offset: 2

Image

default rstride: $RU4 \text{ (width * 3)}$
default size: $\text{rstride (image) * height}$

- **"BGR"** reverse rgb

```
+---+---+---+ +---+---+---+
|B0|G0|R0| |B1|G1|R1| ...
+---+---+---+ +---+---+---+
```

Component 0: R

depth: 8
pstride: 3
offset: 2

Component 1: G

depth: 8
pstride: 3
offset: 1

Component 2: B

depth 8
pstride: 3
offset: 0

Image

default rstride: RU4 (width * 3)
default size: rstride (image) * height

- **"Y41B"** planar 4:1:1 YUV

Component 0: Y

depth: 8
pstride: 1
default offset: 0
default rstride: RU4 (width)
default size: rstride (component0) * height

Component 1: U

depth 8
pstride: 1
default offset: size (component0)
default rstride: RU16 (width) / 4
default size: rstride (component1) * height

Component 2: V

```
depth:          8

pstride:        1

default offset: offset (component1) + size (component1)

default rstride: RU16 (width) / 4

default size:   rstride (component2) * height
```

Image

```
default size: size (component0) +
              size (component1) +
              size (component2)
```

- **"Y42B"** planar 4:2:2 YUV

Component 0: Y

```
depth:          8

pstride:        1

default offset: 0

default rstride: RU4 (width)

default size:   rstride (component0) * height
```

Component 1: U

```
depth          8

pstride:        1

default offset: size (component0)

default rstride: RU8 (width) / 2

default size:   rstride (component1) * height
```

Component 2: V

```
depth:          8

pstride:        1

default offset: offset (component1) + size (component1)
```

```
default rstride: RU8 (width) / 2  
default size:   rstride (component2) * height
```

Image

```
default size: size (component0) +  
              size (component1) +  
              size (component2)
```

- **"Y444"** planar 4:4:4 YUV

Component 0: Y

```
depth:         8  
pstride:       1  
default offset: 0  
default rstride: RU4 (width)  
default size:   rstride (component0) * height
```

Component 1: U

```
depth          8  
pstride:       1  
default offset: size (component0)  
default rstride: RU4 (width)  
default size:   rstride (component1) * height
```

Component 2: V

```
depth:         8  
pstride:       1  
default offset: offset (component1) + size (component1)  
default rstride: RU4 (width)  
default size:   rstride (component2) * height
```

Image

default size: size (component0) +
 size (component1) +
 size (component2)

- **"v210"** packed 4:2:2 10-bit YUV, complex format

Component 0: Y

depth: 10

Component 1: U

depth 10

Component 2: V

depth: 10

Image

default rstride: RU48 (width) * 128

default size: rstride (image) * height

- **"v216"** packed 4:2:2 16-bit YUV, Y0-U0-Y1-V1 order

+---+---+---+---+ +---+---+---+---+

|U0|Y0|V0|Y1| |U1|Y2|V1|Y3| ...

+---+---+---+---+ +---+---+---+---+

Component 0: Y

depth: 16 LE

pstride: 4

offset: 2

Component 1: U

depth 16 LE
pstride: 8
offset: 0

Component 2: V

depth: 16 LE
pstride: 8
offset: 4

Image

default rstride: RU8 (width * 2)
default size: rstride (image) * height

- **"NV12"** planar 4:2:0 YUV with interleaved UV plane

Component 0: Y

depth: 8
pstride: 1
default offset: 0
default rstride: RU4 (width)
default size: rstride (component0) * RU2 (height)

Component 1: U

depth 8
pstride: 2
default offset: size (component0)
default rstride: RU4 (width)

Component 2: V

depth: 8
pstride: 2
default offset: offset (component1) + 1
default rstride: RU4 (width)

Image

default size: RU4 (width) * RU2 (height) * 3 / 2

- **"NV21"** planar 4:2:0 YUV with interleaved VU plane

Component 0: Y

depth: 8
pstride: 1
default offset: 0
default rstride: RU4 (width)
default size: rstride (component0) * RU2 (height)

Component 1: U

depth 8
pstride: 2
default offset: offset (component1) + 1
default rstride: RU4 (width)

Component 2: V

depth: 8
pstride: 2
default offset: size (component0)
default rstride: RU4 (width)

Image

default size: RU4 (width) * RU2 (height) * 3 / 2

- **"GRAY8"** 8-bit grayscale "Y800" same as "GRAY8"

Component 0: Y

depth: 8

offset: 0

pstride: 1

default rstride: RU4 (width)

default size: rstride (component0) * height

Image

default size: size (component0)

- **"GRAY16_BE"** 16-bit grayscale, most significant byte first

Component 0: Y

depth: 16

offset: 0

pstride: 2

default rstride: RU4 (width * 2)

default size: rstride (component0) * height

Image

default size: size (component0)

- **"GRAY16_LE"** 16-bit grayscale, least significant byte first

- **"Y16"** same as "GRAY16_LE"

Component 0: Y

depth: 16 LE
 offset: 0
 pstride: 2
 default rstride: RU4 (width * 2)
 default size: rstride (component0) * height

Image

default size: size (component0)

- **"v308"** packed 4:4:4 YUV

```
+---+---+---+ +---+---+---+
|Y0|U0|V0| |Y1|U1|V1| ...
+---+---+---+ +---+---+---+
```

Component 0: Y

depth: 8
 pstride: 3
 offset: 0

Component 1: U

depth: 8
 pstride: 3
 offset: 1

Component 2: V

depth: 8
 pstride: 3
 offset: 2

Image

default rstride: RU4 (width * 3)

default size: rstride (image) * height

- **"IYU2"** packed 4:4:4 YUV, U-Y-V order

+---+---+---+ +---+---+---+

|U0|Y0|V0| |U1|Y1|V1| ...

+---+---+---+ +---+---+---+

Component 0: Y

depth: 8

pstride: 3

offset: 1

Component 1: U

depth 8

pstride: 3

offset: 0

Component 2: V

depth: 8

pstride: 3

offset: 2

Image

default rstride: RU4 (width * 3)

default size: rstride (image) * height

- **"RGB16"** rgb 5-6-5 bits per component

+---+---+---+ +---+---+---+

|R0|G0|B0| |R1|G1|B1| ...

+---+---+---+ +---+---+---+

Component 0: R

depth: 5

pstride: 2

Component 1: G

depth 6

pstride: 2

Component 2: B

depth: 5

pstride: 2

Image

default rstride: RU4 (width * 2)

default size: rstride (image) * height

- **"BGR16"** reverse rgb 5-6-5 bits per component

+---+---+---+ +---+---+---+

|B0|G0|R0| |B1|G1|R1| ...

+---+---+---+ +---+---+---+

Component 0: R

depth: 5

```
pstride:      2
```

```
Component 1: G
```

```
depth        6
```

```
pstride:      2
```

```
Component 2: B
```

```
depth:       5
```

```
pstride:      2
```

```
Image
```

```
default rstride: RU4 (width * 2)
```

```
default size:  rstride (image) * height
```

```
- ***"RGB15"*** rgb 5-5-5 bits per component
```

```
+---+---+---+ +---+---+---+
```

```
|R0|G0|B0| |R1|G1|B1| ...
```

```
+---+---+---+ +---+---+---+
```

```
Component 0: R
```

```
depth:       5
```

```
pstride:      2
```

```
Component 1: G
```

```
depth        5
```

```
pstride:      2
```

```
Component 2: B
```

```
depth:       5
```

```
pstride:      2
```

Image

```
default rstride: RU4 (width * 2)
```

```
default size:  rstride (image) * height
```

- **"BGR15"** reverse rgb 5-5-5 bits per component

```
+---+---+---+ +---+---+---+
```

```
|B0|G0|R0| |B1|G1|R1| ...
```

```
+---+---+---+ +---+---+---+
```

Component 0: R

```
depth:      5
```

```
pstride:    2
```

Component 1: G

```
depth      5
```

```
pstride:   2
```

Component 2: B

```
depth:     5
```

```
pstride:   2
```

Image

```
default rstride: RU4 (width * 2)
```

```
default size:  rstride (image) * height
```

- **"UYVP"** packed 10-bit 4:2:2 YUV (U0-Y0-V0-Y1 U2-Y2-V2-Y3 U4 ...)

Component 0: Y

depth: 10

Component 1: U

depth 10

Component 2: V

depth: 10

Image

default rstride: RU4 (width * 2 * 5)

default size: rstride (image) * height

- **"A420"** planar 4:4:2:0 AYUV

Component 0: Y

depth: 8

pstride: 1

default offset: 0

default rstride: RU4 (width)

default size: rstride (component0) * RU2 (height)

Component 1: U

depth 8

pstride: 1

default offset: size (component0)

default rstride: RU4 (RU2 (width) / 2)

default size: rstride (component1) * (RU2 (height) / 2)

Component 2: V

```
depth:          8
pstride:        1
default offset: size (component0) + size (component1)
default rstride: RU4 (RU2 (width) / 2)
default size:   rstride (component2) * (RU2 (height) / 2)
```

Component 3: A

```
depth:          8
pstride:        1
default offset: size (component0) + size (component1) +
                size (component2)
default rstride: RU4 (width)
default size:   rstride (component3) * RU2 (height)
```

Image

```
default size:   size (component0) +
                size (component1) +
                size (component2) +
                size (component3)
```

- **"RGB8P"** 8-bit paletted RGB

Component 0: INDEX

```
depth:          8
pstride:        1
default offset: 0
default rstride: RU4 (width)
default size:   rstride (component0) * height
```

Component 1: PALETTE

```
depth          32
pstride:       4
default offset: size (component0)
rstride:       4
size:          256 * 4
```

Image

```
default size:  size (component0) + size (component1)
```

- **"YUV9"** planar 4:1:0 YUV

Component 0: Y

```
depth:         8
pstride:        1
default offset: 0
default rstride: RU4 (width)
default size:   rstride (component0) * height
```

Component 1: U

```
depth          8
pstride:        1
default offset: size (component0)
default rstride: RU4 (RU4 (width) / 4)
default size:   rstride (component1) * (RU4 (height) / 4)
```

Component 2: V

```
depth:         8
pstride:        1
default offset: offset (component1) + size (component1)
default rstride: RU4 (RU4 (width) / 4)
```

```
default size: rstride (component2) * (RU4 (height) / 4)
```

Image

```
default size: size (component0) +  
             size (component1) +  
             size (component2)
```

- ****"YVU9"*** planar 4:1:0 YUV (like YUV9 but UV planes swapped)

Component 0: Y

```
depth:      8  
pstride:    1  
default offset: 0  
default rstride: RU4 (width)  
default size: rstride (component0) * height
```

Component 1: U

```
depth      8  
pstride:    1  
default offset: offset (component2) + size (component2)  
default rstride: RU4 (RU4 (width) / 4)  
default size: rstride (component1) * (RU4 (height) / 4)
```

Component 2: V

```
depth:      8  
pstride:    1  
default offset: size (component0)  
default rstride: RU4 (RU4 (width) / 4)  
default size: rstride (component2) * (RU4 (height) / 4)
```


Image

default size: size (component0) +
 size (component1) +
 size (component2)

- **"IYU1"** packed 4:1:1 YUV (Cb-Y0-Y1-Cr-Y2-Y3 ...)

+---+---+---+ +---+---+---+

|B0|G0|R0| |B1|G1|R1| ...

+---+---+---+ +---+---+---+

Component 0: Y

depth: 8
 offset: 1
 pstride: 2

Component 1: U

depth 5
 offset: 0
 pstride: 2

Component 2: V

depth: 5
 offset: 4
 pstride: 2

Image

default rstride: RU4 (RU4 (width) + RU4 (width) / 2)

default size: rstride (image) * height

- **ARGB64** rgb with alpha channel first, 16 bits per channel

+-----+ +-----+

|A0|R0|G0|B0| |A1|R1|G1|B1| ...

+-----+ +-----+

Component 0: R

depth: 16 LE

pstride: 8

offset: 2

Component 1: G

depth 16 LE

pstride: 8

offset: 4

Component 2: B

depth: 16 LE

pstride: 8

offset: 6

Component 3: A

depth: 16 LE

pstride: 8

offset: 0

Image

default rstride: width * 8

default size: rstride (image) * height

- **"AYUV64"** packed 4:4:4 YUV with alpha channel, 16 bits per channel (A0-Y0-U0-V0 ...)

+---+---+---+---+ +---+---+---+---+

|A0|Y0|U0|V0| |A1|Y1|U1|V1| ...

+---+---+---+---+ +---+---+---+---+

Component 0: Y

depth: 16 LE

pstride: 8

offset: 2

Component 1: U

depth 16 LE

pstride: 8

offset: 4

Component 2: V

depth: 16 LE

pstride: 8

offset: 6

Component 3: A

depth: 16 LE

pstride: 8

offset: 0

Image

default rstride: width * 8

default size: rstride (image) * height

- **"r210"** packed 4:4:4 RGB, 10 bits per channel

+---+---+---+ +---+---+---+

|R0|G0|B0| |R1|G1|B1| ...

+---+---+---+ +---+---+---+

Component 0: R

depth: 10

pstride: 4

Component 1: G

depth 10

pstride: 4

Component 2: B

depth: 10

pstride: 4

Image

default rstride: width * 4

default size: rstride (image) * height

- **"I420_10LE"** planar 4:2:0 YUV, 10 bits per channel LE

Component 0: Y

depth: 10 LE

pstride: 2

default offset: 0

default rstride: RU4 (width * 2)

default size: rstride (component0) * RU2 (height)

Component 1: U

depth: 10 LE
pstride: 2
default offset: size (component0)
default rstride: RU4 (width)
default size: rstride (component1) * RU2 (height) / 2

Component 2: V

depth 10 LE
pstride: 2
default offset: offset (component1) + size (component1)
default rstride: RU4 (width)
default size: rstride (component2) * RU2 (height) / 2

Image

default size: size (component0) +
size (component1) +
size (component2)

- **"I420_10BE"** planar 4:2:0 YUV, 10 bits per channel BE

Component 0: Y

depth: 10 BE
pstride: 2
default offset: 0
default rstride: RU4 (width * 2)
default size: rstride (component0) * RU2 (height)

Component 1: U

```
depth:          10 BE
pstride:        2
default offset: size (component0)
default rstride: RU4 (width)
default size:   rstride (component1) * RU2 (height) / 2
```

Component 2: V

```
depth          10 BE
pstride:       2
default offset: offset (component1) + size (component1)
default rstride: RU4 (width)
default size:   rstride (component2) * RU2 (height) / 2
```

Image

```
default size: size (component0) +
              size (component1) +
              size (component2)
```

- **"I422_10LE"** planar 4:2:2 YUV, 10 bits per channel LE

Component 0: Y

```
depth:          10 LE
pstride:        2
default offset: 0
default rstride: RU4 (width * 2)
default size:   rstride (component0) * RU2 (height)
```

Component 1: U

```
depth:          10 LE
pstride:        2
```

```
default offset: size (component0)
default rstride: RU4 (width)
default size: rstride (component1) * RU2 (height)
```

Component 2: V

```
depth          10 LE
pstride:       2
default offset: offset (component1) + size (component1)
default rstride: RU4 (width)
default size: rstride (component2) * RU2 (height)
```

Image

```
default size: size (component0) +
              size (component1) +
              size (component2)
```

- **"I422_10BE"** planar 4:2:2 YUV, 10 bits per channel BE

Component 0: Y

```
depth:        10 BE
pstride:       2
default offset: 0
default rstride: RU4 (width * 2)
default size: rstride (component0) * RU2 (height)
```

Component 1: U

```
depth:        10 BE
pstride:       2
default offset: size (component0)
default rstride: RU4 (width)
```

default size: rstride (component1) * RU2 (height)

Component 2: V

depth 10 BE

pstride: 2

default offset: offset (component1) + size (component1)

default rstride: RU4 (width)

default size: rstride (component2) * RU2 (height)

Image

default size: size (component0) +
size (component1) +
size (component2)

- **"Y444_10BE"** planar 4:4:4 YUV, 10 bits per channel
- **"Y444_10LE"** planar 4:4:4 YUV, 10 bits per channel

- **"GBR"** planar 4:4:4 RGB, 8 bits per channel
- **"GBR_10BE"** planar 4:4:4 RGB, 10 bits per channel
- **"GBR_10LE"** planar 4:4:4 RGB, 10 bits per channel

- **"NV16"** planar 4:2:2 YUV with interleaved UV plane
- **"NV61"** planar 4:2:2 YUV with interleaved VU plane
- **"NV24"** planar 4:4:4 YUV with interleaved UV plane

- **"NV12_64Z32"** planar 4:2:0 YUV with interleaved UV plane in 64x32 tiles zigzag

Component 0: Y

depth: 8

pstride: 1


```
default offset: 0
default rstride: RU128 (width)
default size:  rstride (component0) * RU32 (height)
```

Component 1: U

```
depth          8
pstride:       2
default offset: size (component0)
default rstride: (y_tiles << 16) | x_tiles
default x_tiles: RU128 (width) >> tile_width
default y_tiles: RU32 (height) >> tile_height
```

Component 2: V

```
depth:         8
pstride:       2
default offset: offset (component1) + 1
default rstride: (y_tiles << 16) | x_tiles
default x_tiles: RU128 (width) >> tile_width
default y_tiles: RU64 (height) >> (tile_height + 1)
```

Image

```
default size: RU128 (width) * (RU32 (height) + RU64 (height) / 2)
tile mode:    ZFLIPZ_2X2
tile width:   6
tile height:  5
```

4. Formatted Media Types

Those formatted media types are supported in Panel-Link version 1:

Type	Format	Extensions	Remark
image	x-degas	no extensions	
image	x-icon	no extensions	
image	vnd.wap.wbmp	no extensions	
image	vnd.adobe.photoshop	psd	
image	svg+xml	svg	
image	x-sun-raster	ras	
image	x-xpixmap	xpm	
image	x-jng	jng	
image	x-xcf	xcf	
image	x-portable-pixmap	pnm, ppm, pgm, pbm	
image	x-exr	exr	
image	webp	webp	
image	tiff	tif, tiff	
image	bmp	bmp	
image	png	png	
image	gif	gif	
image	jpeg	jpg, jpe, jpeg	
image	jp2	jp2	
image	x-quicktime	qif, qtif, qti	
video	x-pva	pva	
video	x-ivf	ivf	
video	vivo	viv	
video	x-dirac	no extensions	
video	x-theora	no extensions	
video	x-mng	mng	
video	x-dv	dv, dif	
video	x-mve	mve	
video	x-matroska	mkv, mka, mk3d, webm	
video	x-flv	flv	
video	mj2	mj2	
video	quicktime	mov, mp4	
video	x-nuv	nuv	
video	x-h265	h265, x265, 265	
video	x-h264	h264, x264, 264	
video	x-h263	h263, 263	
video	mpeg4	m4v	

video	mpeg-elementary	mpv, mpeg, mpg	
video	mpegts	ts, mts	
video	mpeg-sys	mpe, mpeg, mpg	
video	x-fli	flc, fli	
video	x-vcd	dat	
video	x-cdxa	dat	
video	x-msvideo	avi	
video	x-ms-asf	asf, wm, wma, wmv	

4-1 Formatted Media Types

5. Endianness Declaration

All the data structures listed in this document should be transferred and/or stored in little-endian format, or USB peer endpoint may receive corrupted data.

6. 16Bits Checksum Algorithm

```
unsigned short checksum16(unsigned short *buf, int nword)
{
    unsigned long sum;

    for (sum = 0; nword > 0; nword--)
        sum += *buf++;
    sum = (sum >> 16) + (sum & 0xffff);
    sum += (sum >> 16);

    return ~sum;
}
```

7. Contact Information

email to: weidong.zhou@nxelec.com

Website www.nxelec.com

8. Revision History

Rev.	Date	Description	Author
0.1	14-Jan-2019	Initial release	wdzhou
0.2	14-Mar-2019	Table 2-2 Add Tag Type 3 -- Reset of Display	wdzhou
0.3	15-Mar-2019	Table 2-2 Add Tag Type 4 – Clear Screen	wdzhou
1.0	26-July-2023	Table 2-2 Add Tag Type 5 and Tag Type 6 Add tag descriptions for continuous PNG and JPEG streams	wdzhou