



## Contents

- 1. Overview ..... - 4 -
  - 1.1 Features ..... - 5 -
  - 1.2 Block Diagram ..... - 6 -
- 2. Function Descriptions ..... - 6 -
  - 2.1 USB Descriptors ..... - 6 -
  - 2.2 Supported Media Type..... - 8 -
    - 2.2.1 Raw Media Types ..... - 8 -
    - 2.2.2 Format Media Types ..... - 48 -
  - 2.3 LCD Display..... - 49 -
  - 2.4 Touch Panel ..... - 49 -
- 3. Quick Start Guide ..... - 50 -
- 4. Electrical Specifications ..... - 50 -
  - 4.1 Absolute Maximum Ratings ..... - 50 -
  - 4.2 Operating Conditions ..... - 50 -
  - 4.3 Power Consumption ..... - 50 -
- 5. Environmental Specifications..... - 50 -
- 6. Mechanical Dimensions..... - 51 -
- 7. Ordering Information ..... - 51 -
- 8. Warranty Terms ..... - 52 -
- 9. Contact Information..... - 52 -
- 10. Revision History ..... - 52 -

## LIST OF TABLES

TABLE 2-1 DEVICE DESCRIPTOR.....	- 7 -
TABLE 2-2 CONFIGURATION DESCRIPTOR .....	- 7 -
TABLE 2-3 INTERFACE DESCRIPTOR.....	- 7 -
TABLE 2-4 ENDPOINT DESCRIPTOR 0x01 .....	- 7 -
TABLE 2-5 ENDPOINT DESCRIPTOR 0x82 .....	- 7 -
TABLE 2-6 SUPPORTED FORMAT MEDIA TYPES.....	- 49 -
TABLE 4-1 ABSOLUTE MAXIMUM RATINGS .....	- 50 -
TABLE 4-2 OPERATING CONDITIONS.....	- 50 -
TABLE 4-3 POWER CONSUMPTION.....	- 50 -
TABLE 5-1 ENVIRONMENTAL SPECIFICATIONS .....	- 50 -
TABLE 7-1 ORDERING INFORMATION.....	- 51 -

# 1. Overview

What is BeadaPanel?

BeadaPanel is a versatile media display. It supports playback of images and videos in various common format.

BeadaPanel comes with a 7" LCD display with a resolution of 800x480 and a 4-wire resistive touch panel.

BeadaPanel supports USB 2.0 High Speed standard and WiFi communication.

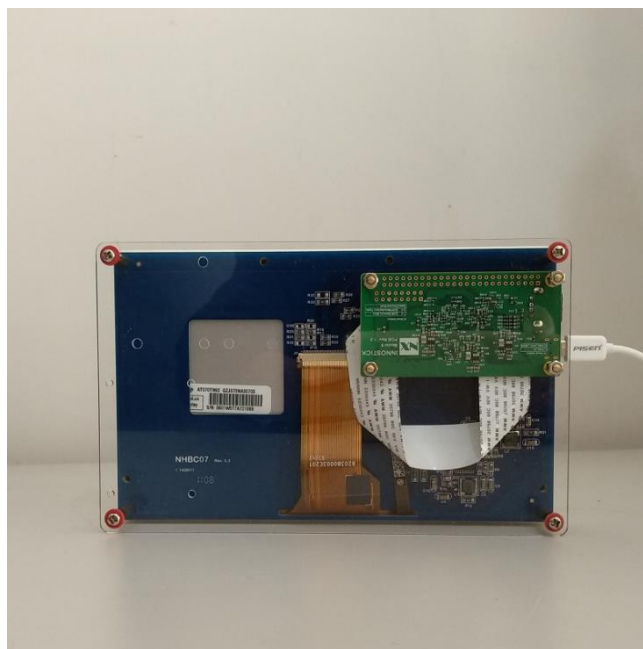
BeadaPanel has up to 16GB/32GB eMMC memory for media storage and playback.

BeadaPanel consumes very low power and can be powered by a single Micro-USB cable.

BeadaPanel has an ARM Cortex-A7 core, running Debian Stretch operating system, which is convenient for software upgrade.

BeadaPanel comes with its own ABS bezel, which is easy to integrate into the end products of customers.

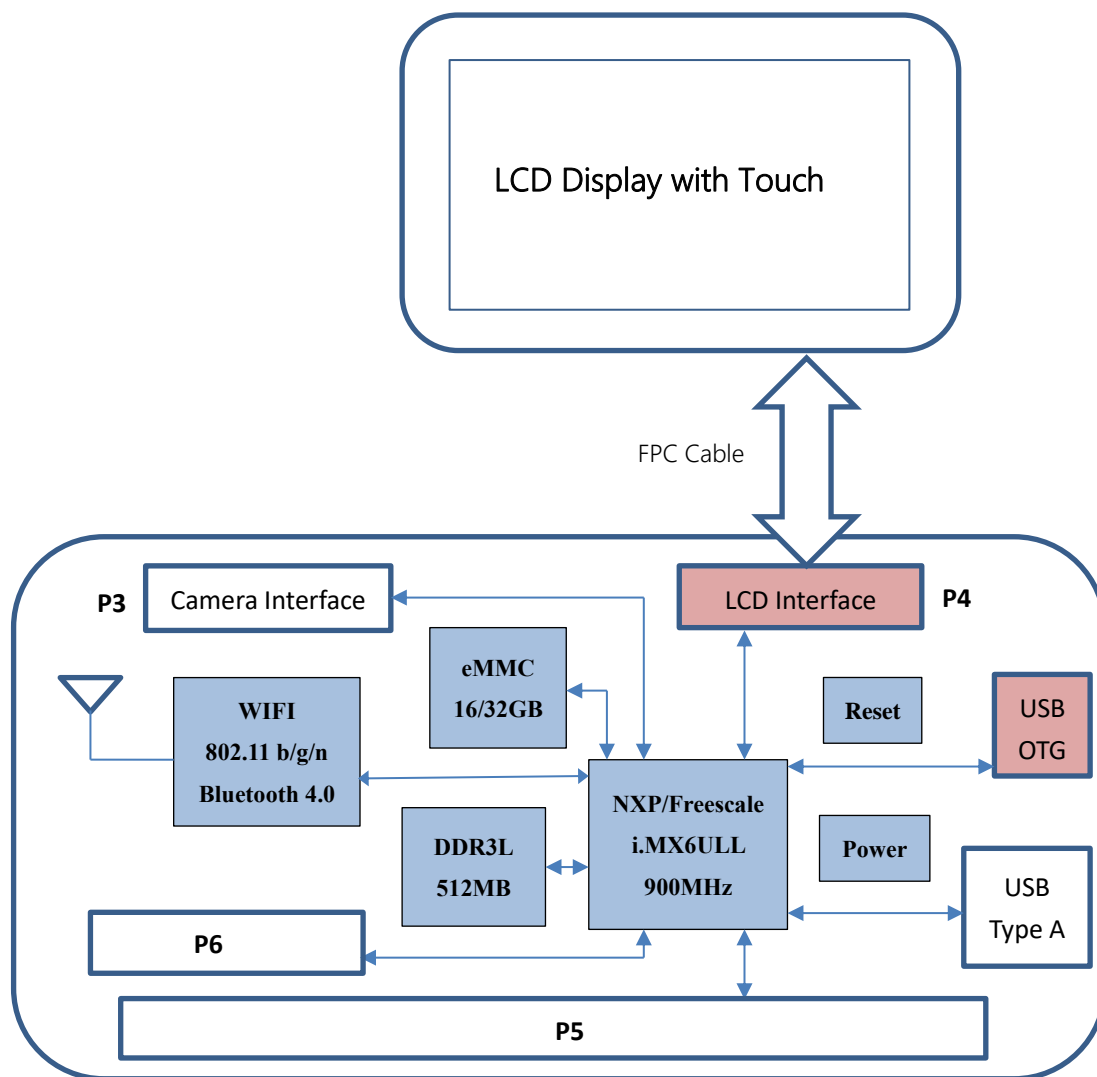




## 1.1 Features

- Processor — NXP i.MX6 ULL, Cortex-A7@900MHz
- Memory/Storage — 512MB DDR3L; 16GB or 32GB eMMC
- Wireless — 802.11b/g/n
- Display:
  - ✓ 7" 800x480 TFT LCD
  - ✓ 4-Wire Resistive Touch Screen
- Other I/O:
  - ✓ USB 2.0 Type A Host port
  - ✓ Micro-USB 2.0 OTG port
- OS — Debian Stretch
- Communication Protocol — Panel-Link V1.0
- Supported Media Types:
  - ✓ JPG, BMP, PNG, GIF, TIFF
  - ✓ MPG4, ASF, OGG, FLV, WebM, RMVB, MKV, avi, quicktime, H.263/264/265, MPEG-1, MPEG-2
  - ✓ RGB in various pack
  - ✓ YUV in various pack
- Dimensions — 188 x 118 x 35mm

## 1.2 Block Diagram



Functional Block Diagram

## 2. Function Descriptions

### 2.1 USB Descriptors

USB host identifies device interfaces and endpoints through USB descriptors. Key fields in BeadaPanel Device Descriptor are as below:

Field Type	Predefine Value
bcdUSB	0x0200
bDeviceClass	0
bDeviceSubClass	0
idVendor	0x4e58
idProduct	0x1001
iManufacturer	"NXElec"
iProduct	"BeadaPanel 800x480"

Table 2-1 Device Descriptor

Key fields in BeadaPanel Configuration Descriptor are as below:

Field Type	Predefine Value
bmAttributes	0x80
MaxPower	175

Table 2-2 Configuration Descriptor

The host communicates with BeadaPanel on NXElec Panel-Link<sup>1</sup> protocol, which usually occupies a unique interface. Key fields in BeadaPanel Interface Descriptor are as below:

Field Type	Predefine Value
bNumEndpoints	2
bInterfaceClass	0xff
bInterfaceSubClass	0
bInterfaceProtocol	0

Table 2-3 Interface Descriptor

Panel-Link protocol uses 2 bulk Endpoints for communication. Endpoint 0x01 is for traffics from host and Endpoint 0x81 is for traffics to host. Key fields in BeadaPanel Endpoint Descriptor are as below:

Field Type	Predefine Value
bEndpointAddress	0x01
bmAttributes	bulk
wMaxPacketSize	512

Table 2-4 Endpoint Descriptor 0x01

Field Type	Predefine Value
bEndpointAddress	0x81
bmAttributes	bulk
wMaxPacketSize	512

Table 2-5 Endpoint Descriptor 0x81

**Note 1:** Refer to "Panel-Link USB Media Stream Transport Protocol Specification" for details of protocol

## 2.2 Supported Media Type

### 2.2.1 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.

## Formats

- `"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 (width * 3)
default size:   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
```

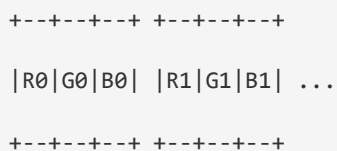
```
offset:    2
```

#### Image

```
default rstride: RU4 (width * 3)
```

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

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



Component 0: R

```

depth:      5
pstride:    2

```

Component 1: G

```

depth       6
pstride:    2

```

Component 2: B

```

depth:      5
pstride:    2

```

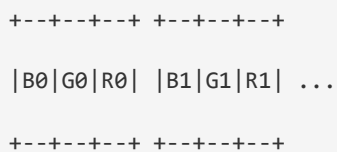
Image

```

default rstride: RU4 (width * 2)
default size:   rstride (image) * height

```

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



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
```

## 2.2.2 Format Media Types

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-pximap	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	

Table 2-6 Supported Format Media Types

## 2.3 LCD Display

BeadaPanel comes with a 7 "TFT LCD, 800x480 resolution.

## 2.4 Touch Panel

BeadaPanel comes with a 4-Wire Resistive touch panel.

## 3. Quick Start Guide

Use of BeadaPanel is straightforward. It can be connected to the host through a Micro-USB cable, which plays a dual role of power supply and communication. The software on BeadaPanel ARM Cortex-A7 core will automatically start after cable connecting. When there is animation playing on LCD screen, then BeadaPanel is ready to receive media stream either from USB or from WiFi.

## 4. Electrical Specifications

### 4.1 Absolute Maximum Ratings

Symbol	Description	Min	Max	Unit
VDD_5V	Main Power Supply	-0.5	5.5	V
USB_OTG_VBUS	USB VBUS	-	5.5	V

Table 4-1 Absolute Maximum Ratings

### 4.2 Operating Conditions

Symbol	Description	Min	Max	Unit
VDD_5V	Main Power Supply	4.4	5.5	V
USB_OTG_VBUS	USB VBUS	4.4	5.5	V

Table 4-2 Operating Conditions

### 4.3 Power Consumption

Mode	Supply	Current	Power	Note
Idle	VDD_5V	70mA	0.35W	
Max CPU load	VDD_5V	350mA	1.75W	

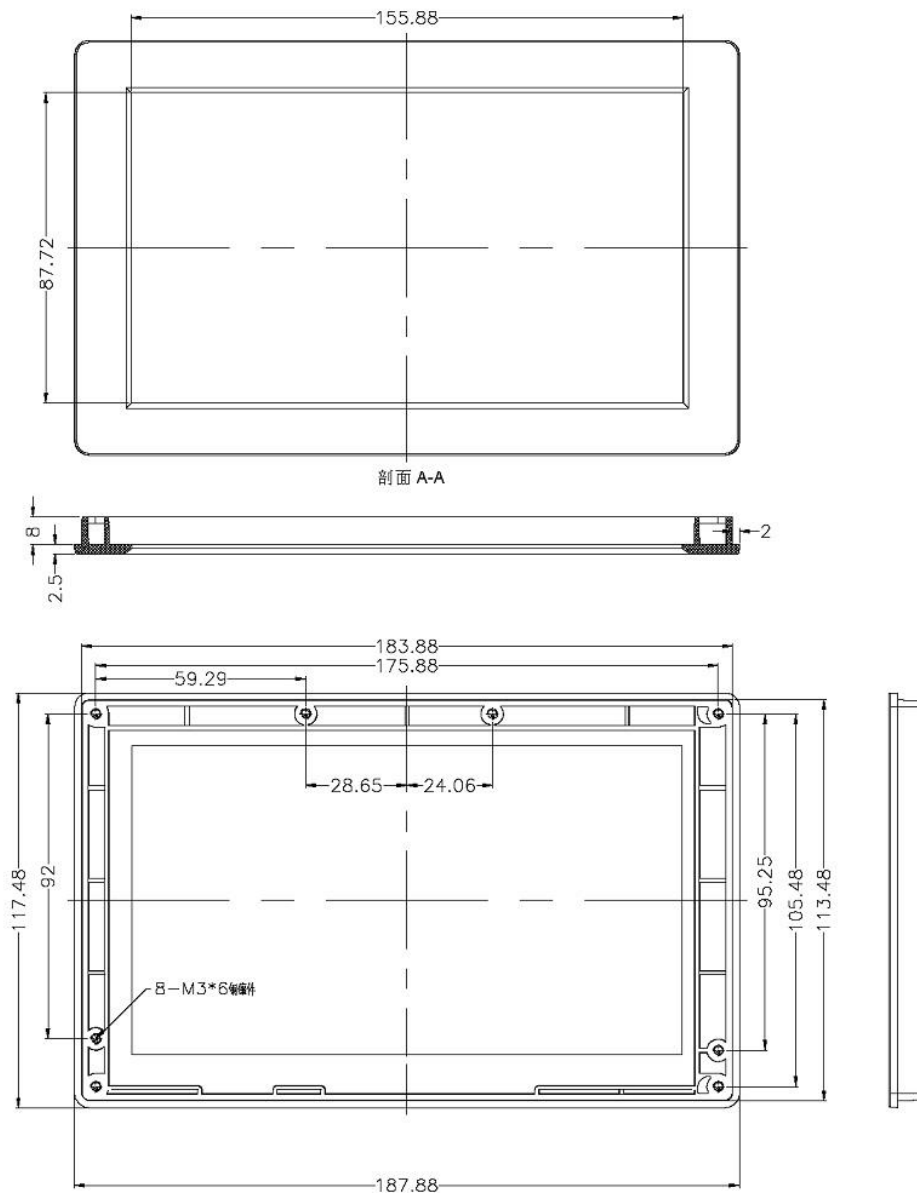
Table 4-3 Power Consumption

## 5. Environmental Specifications

Parameter	Min	Max	Unit
Commercial Operating Temperature Range	0	70	°C
Storage Temperature Range	-40	85	°C
Humidity	10	95 Non-Condensing	%

Table 5-1 Environmental Specifications

## 6. Mechanical Dimensions



Mechanical Dimensions

## 7. Ordering Information

Part No.	Part Name	Screen Size	eMMC Size	Touch
NDM07002	BedaPanel 7" Media Display 8GB	7" 800x480	8GB	Resistive

Table 7-1 Ordering Information

## 8. Warranty Terms

Naxing Electronics guarantees hardware products against defects in workmanship and material for a period of one (1) year from the date of shipment. Your sole remedy and Naxing Electronics' sole liability shall be for Naxing Electronics, at its sole discretion, to either repair or replace the defective hardware product at no charge or to refund the purchase price. Shipment costs in both directions are the responsibility of the customer. This warranty is void if the hardware product has been altered or damaged by accident, misuse or abuse.

### Disclaimer of Warranty

THIS WARRANTY IS MADE IN LIEU OF ANY OTHER WARRANTY, WHETHER EXPRESSED, OR IMPLIED, OF MERCHANTABILITY, FITNESS FOR A SPECIFIC PURPOSE, NON-INFRINGEMENT OR THEIR EQUIVALENTS UNDER THE LAWS OF ANY JURISDICTION, EXCEPT THE WARRANTY EXPRESSLY STATED HEREIN. THE REMEDIES SET FORTH HEREIN SHALL BE THE SOLE AND EXCLUSIVE REMEDIES OF ANY PURCHASER WITH RESPECT TO ANY DEFECTIVE PRODUCT.

### Limitation on Liability

UNDER NO CIRCUMSTANCES SHALL VARISCITE BE LIABLE FOR ANY LOSS, DAMAGE OR EXPENSE SUFFERED OR INCURRED WITH RESPECT TO ANY DEFECTIVE PRODUCT. IN NO EVENT SHALL VARISCITE BE LIABLE FOR ANY INCIDENTAL OR CONSEQUENTIAL DAMAGES THAT YOU MAY SUFFER DIRECTLY OR INDIRECTLY FROM USE OF ANY PRODUCT.

## 9. Contact Information

Inquiries about product information or purchasing, please email [sales@nxelec.com](mailto:sales@nxelec.com)

Technical support, email to [service@nxelec.com](mailto:service@nxelec.com) or technical forum

[www.esky-sh.com/bbs](http://www.esky-sh.com/bbs)(recommended)

Business and marketing, contact [market@nxelec.com](mailto:market@nxelec.com)

Website [www.nxelec.com](http://www.nxelec.com)

## 10. Revision History

Rev.	Date	Description	Author
0.1	13-Feb-2019	Initial release	wdzhou
0.2	25-Feb-2023	Correct endpoint description in section 2.1	wdzhou