APPROVAL SHEET

2M LVDS video encoder

Username: ADMIN
Password: 1234

Designed	Checked	Approved

REVISION HISTORY

Date	Revision	Description
2024.01.	0.1	Init
2025.02	1.0	Add mode setting of LVDS (dual or single mode)

1. SoC board
   1.1. Specification

Model Name	ME-330
SoC	32bit Arm Cortex A9 NT98528
Memory(DDR / Nand)	DDR3 4Gbit, Nand flash 2Gbit
Video	
Video Compression	H.264 / H.265 / MJPEG
Streaming capacity	Triple Streaming + SVC Stream (3 x H.264/H.265 + 1 x SVC + 1 x JPEG)
Stream	Main Stream: 1920x1080, 1280x1024, 1024x768, 1280x720, 704x576, 704x480, 640x480,640x360, 352x288,352x240 Second/Third Stream : 640x480, 640x360,352x288,352x240
Max Video Frame rate	1920 x 1080 30fps
Video Bit rate	Main Stream : 512 ~ 10000 kbps Second Stream : 512 ~ 8000 kbps Third Stream : 512 ~ 3000 kbps
Video Bit rate control	CBR/VBR
Motion detection	-
Region of Interest	-
Flip Horizontal / Flip Vertical	-
Privacy masking	-
Audio	
Audio compression	G.711 ulaw
Network	
Network	10/100Mbps
Protocol	IPv4, IPv6, IGMP, ICMP, ARP, TCP, UDP, DHCP, RTP, RTSP, RTCP, DNS, NTP, UPnP, HTTP, HTTPS, SSL/TLS
Stream method	Unicast / Multicast
SDK	HTTP based API, Onvif Profile S
Compatibility	ONVIF(Profile S), CGI
Web viewer	Internet Explorer, Chrome, Firefox, Safari
Interface	
Alarm In/Out	1/1

Audio In/Out	1/1
RS-485	-
USB	-
Storage	-
Power	
Power supply	12V
Power consumption	TBD
General	
Demension	42 x 42
Operating Temperature/Huminity	0℃ ~ 50℃/10%~90%

Table 1.1 Specification

1.2. Layout

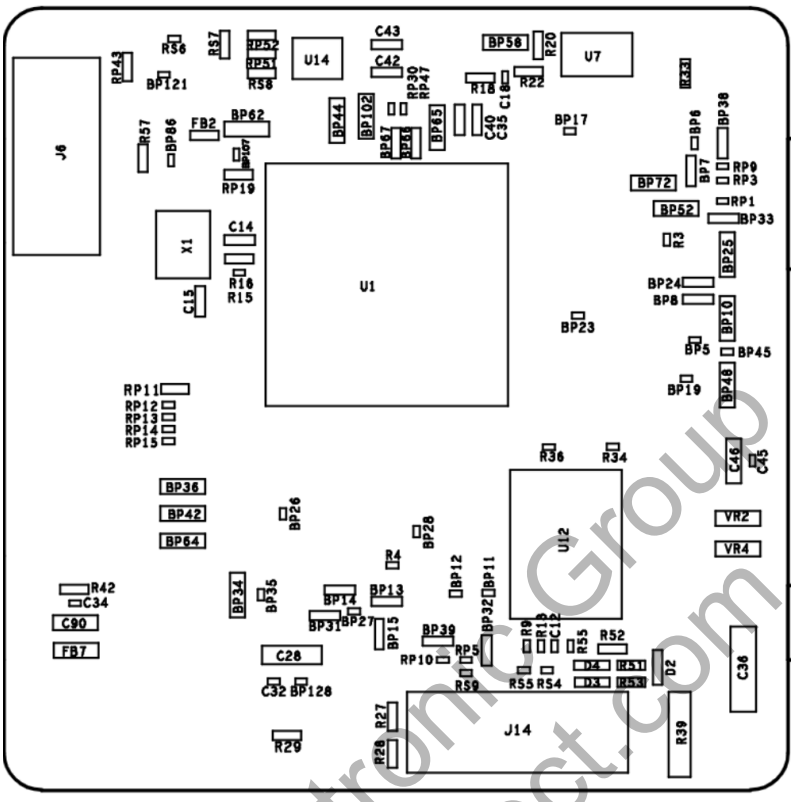


Figure 1.1 Top view of SoC board

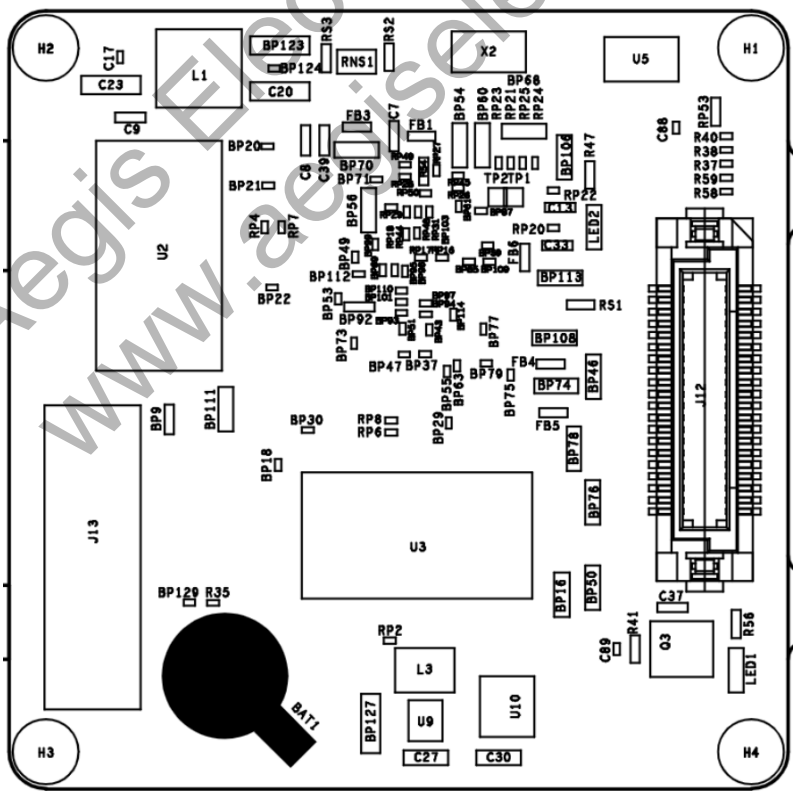


Figure 1.2 bottom view of SoC board

2. LVDS receiver board

2.1. Specification

Video Interface	
Resolution	Upto 2M : 1920 x 1080 30fps
Input interface	input : LVDS 4/8 lanes(Sony LVDS) via 30 pin micro coaxial cable LVDS mode : selectable single / dual
Output interface	output : BT1120/BT601 16bit
Interface	
VISCA	support TTL level interface
Alarm in/out	Input : 1 Ouput : 1(MOSFET relay)
Audio in/out	Input : 1 Ouput : 1
AD input	support upto 1.8V analog signal
Power	
Power supply	9~15V / 1A
General	
Demension	42 x 42
Operating Temperature/Huminity	0℃ ~ 50℃/10%~90%

2.2. Layout

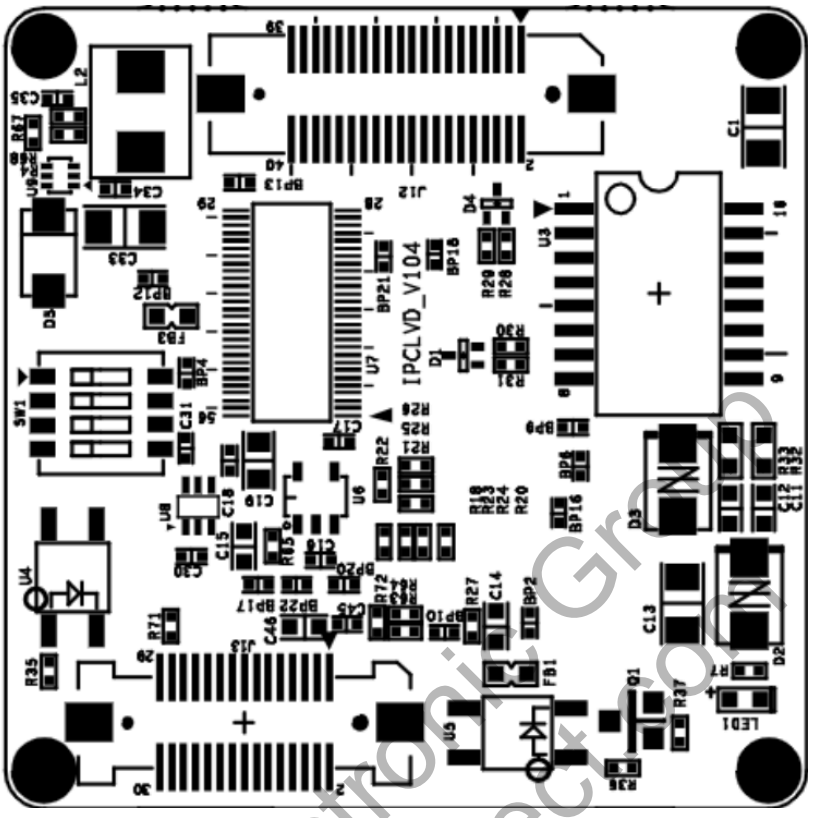


Figure 2.1 Top view of LVDS receiver board

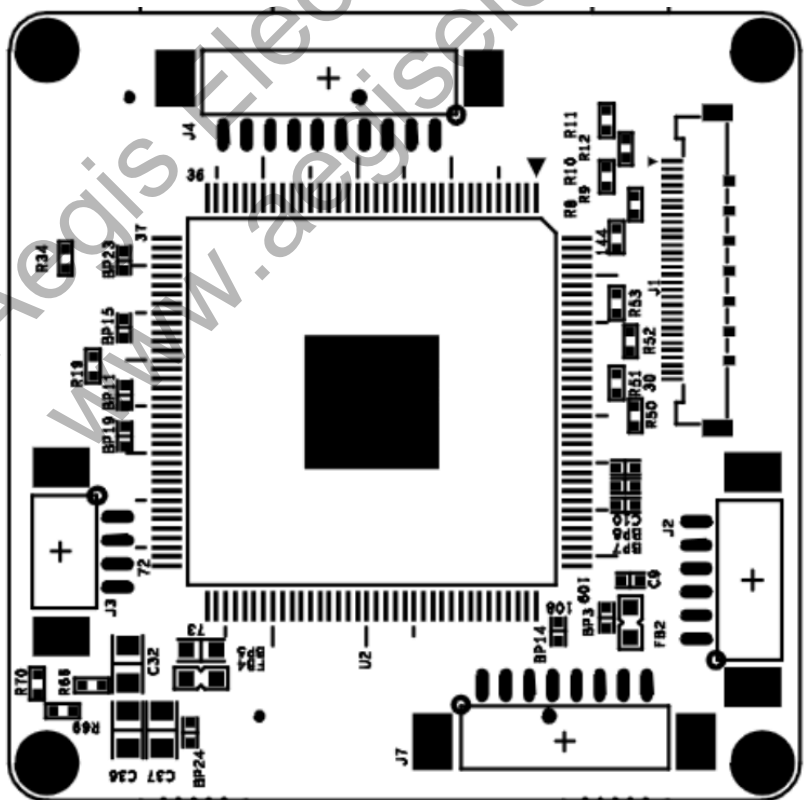


Figure 2.2 bottom view of LVDS receiver board

2.3. Pin map

2.3.1. J1

Pin	signal	in/out	description
1	TXOUT3+	INPUT	Image LVDS data
2	TXOUT3-	INPUT	Image LVDS data
3	TXCLKOUT+	INPUT	Image clock
4	TXCLKOUT-	INPUT	Image clock
5	TXOUT2+	INPUT	Image LVDS data
6	TXOUT2-	INPUT	Image LVDS data
7	TXOUT1+	INPUT	Image LVDS data
8	TXOUT1-	INPUT	Image LVDS data
9	TXOUT0+	INPUT	Image LVDS data
10	TXOUT0-	INPUT	Image LVDS data
11	GND1		
12	TXD	OUTPUT	3.3V CMOS
13	RXD	INPUT	3.3V CMOS
14	DC IN1	POWER OUT	supply power to Zoom module(12V)
15	DC IN2	POWER OUT	supply power to Zoom module(12V)
16	DC IN3	POWER OUT	supply power to Zoom module(12V)
17	DC IN4	POWER OUT	supply power to Zoom module(12V)
18	DC IN5	POWER OUT	supply power to Zoom module(12V)
19	GND2		
20	GND3		
21	TXOUT7+	INPUT	Image LVDS data
22	TXOUT7-	INPUT	Image LVDS data
23	TXOUT6+	INPUT	Image LVDS data
24	TXOUT6-	INPUT	Image LVDS data
25	NC1		
26	Reset	Output	Zoom module reset(1.8V)
27	TXOUT5+	INPUT	Image LVDS data
28	TXOUT5-	INPUT	Image LVDS data
29	TXOUT4+	INPUT	Image LVDS data
30	TXOUT4-	INPUT	Image LVDS data

J1 Connector : USL00-30L
J1 Plug : USL30P

2.3.2. J2

Pin	signal	in/out	description
1	NET_TXP	OUTPUT	ethernet
2	NC		
3	NET_TXN	OUTPUT	ethernet
4	NET_RXP	INPUT	ethernet
5	NC		
6	NET_RXN	INPUT	ethernet

J2 connector : 12505WR-6

2.3.3. J3

Pin	signal	in/out	description
1	12V	POWER INPUT	main power input
2	12V	POWER INPUT	main power input
3	GND		
4	GND		

J3 connector : 12505WR-4

2.3.4. J4

Pin	signal	in/out	description
1	EXT_RESETn	OUTPUT	RESET output
2	UART RX	INPUT	UART RX 3.3V CMOS
3	UART TX	OUTPUT	UART TX 3.3V CMOS
4	UART RTSN	OUTPUT	UART RTSN 3.3V CMOS
5	ALARM_COM	OUTPUT	Alarm common(Photo coupler)
6	ALARM_NO	OUTPUT	Alarm normal open(Photo coupler)
7	ALARM_IN	INPUT	Alarm input
8	GND		
9	VISCA RX	INPUT	3.3V CMOS, alarm input
10	VISCA TX	OUTPUT	3.3V CMOS, alarm output

J4 connector : 12505WR-10

2.3.5. J7

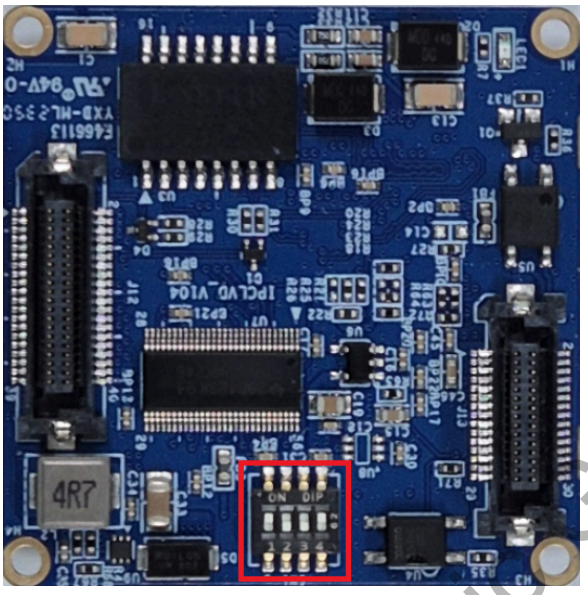
Pin	signal	in/out	description
1	RJ45_LED0	OUTPUT	LED signal of ethernet
2	RJ45_LED1	OUTPUT	LED signal of ethernet
3	GND		
4	FACTORY_RST	INPUT	3.3V CMOS To reset to factory reset, press and hold for more than 10 seconds
5	AD_KEY	INPUT	analog input(0 ~ 2.7V)
6	GND		
7	AUDIO_OUT	OUTPUT	audio input 0.8Vrms
8	AUDIO_IN	INPUT	audio input 1Vrms

J7 connector : 12505WR-8

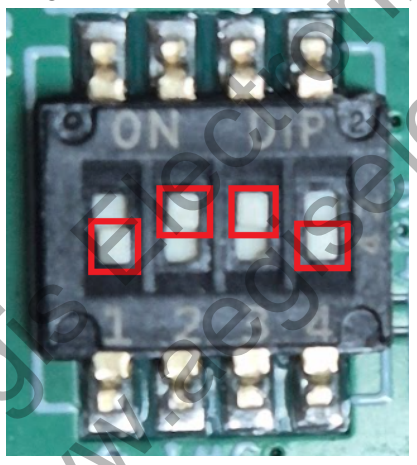
3. Install cable / mode setting

3.1. Mode setting

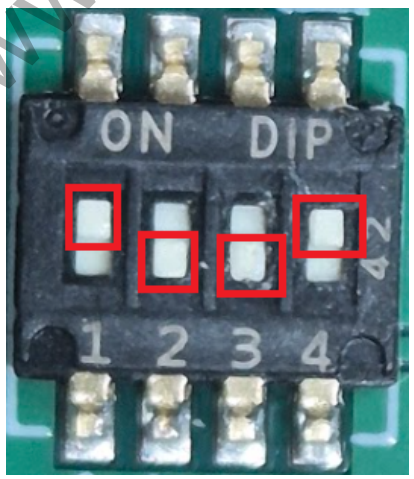
ME-330 can set LVDS mode of zoom module by SW1.



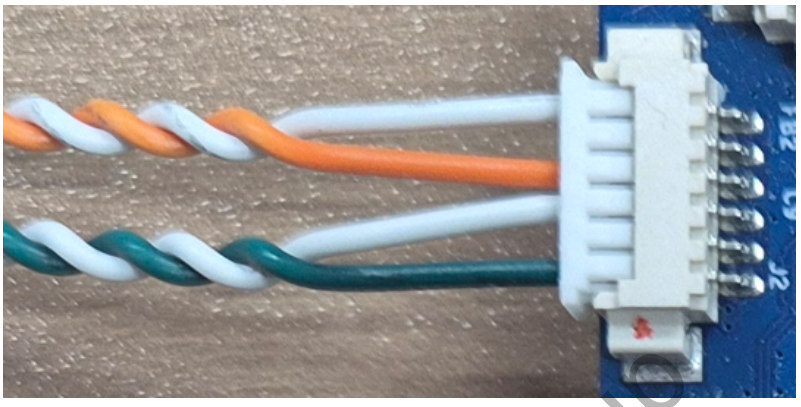
3.1.1. Single mode



3.1.2. Dual mode

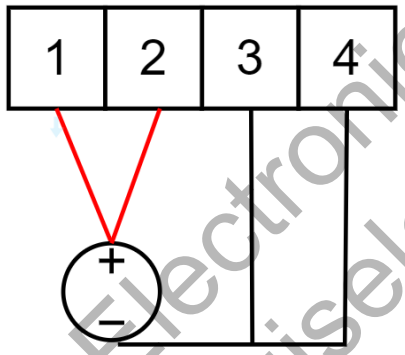


3.2. J2(Network)



A twist pair cable must be used as shown in the figure above.

3.3. J3(Power input)



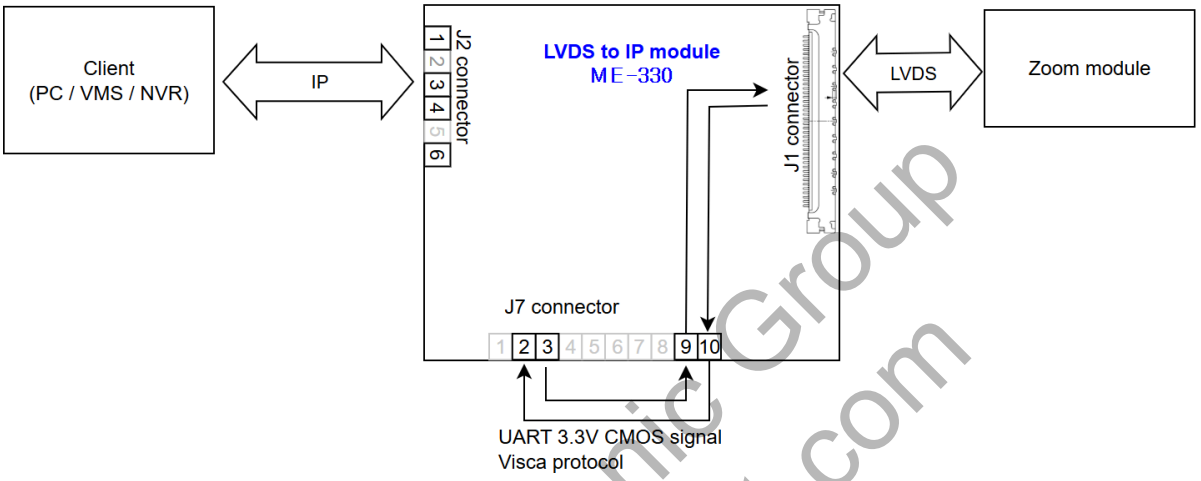
Range of power 9~15V.

3.4. J4(Function 1)

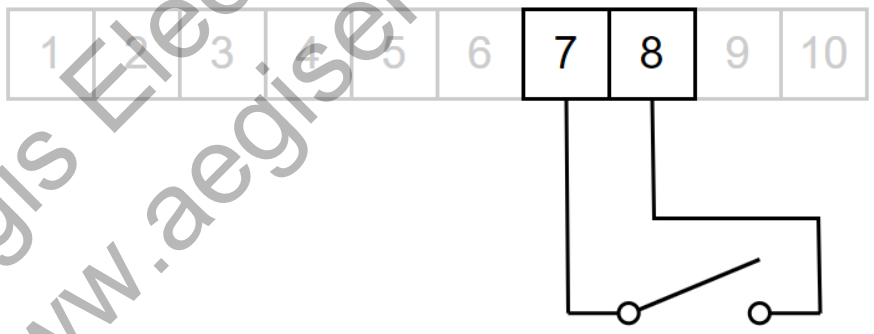
3.4.1. Visca cable connection

3.4.1.1. example connection with zoom module

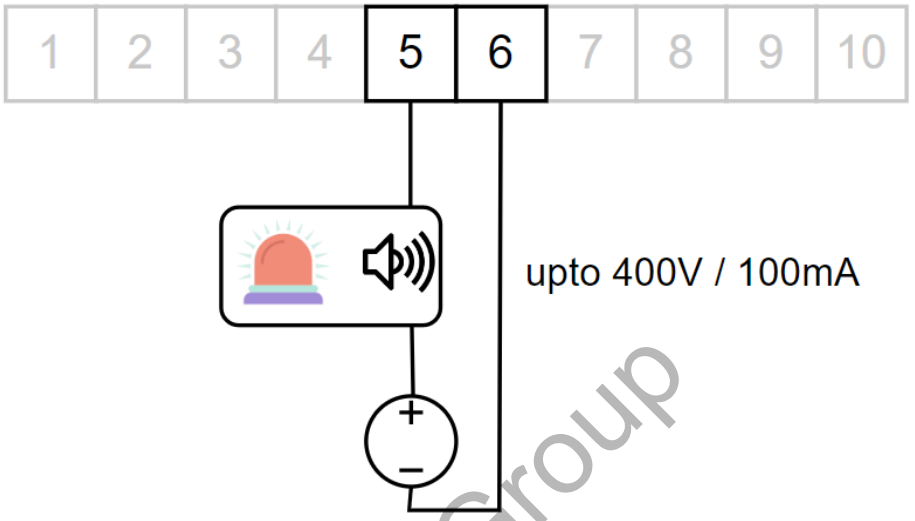
Case only connecting the zoom module, connect the loop cable as shown below and use it.



3.4.2. Alarm In(Normal open)

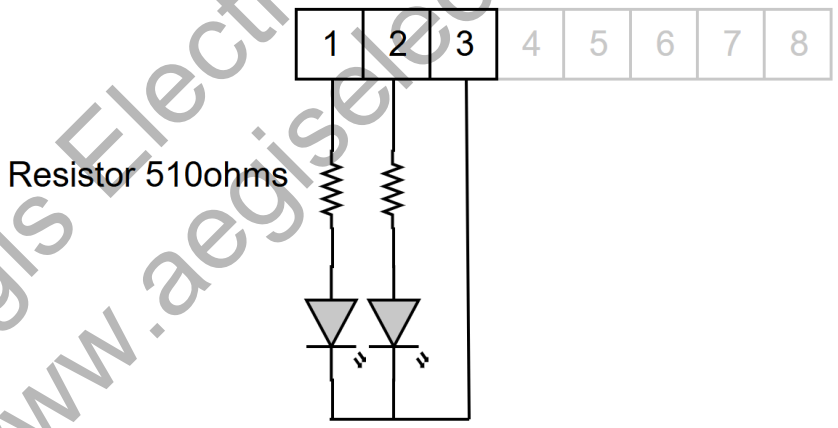


3.4.3. Alarm Out
Mosfet relay type

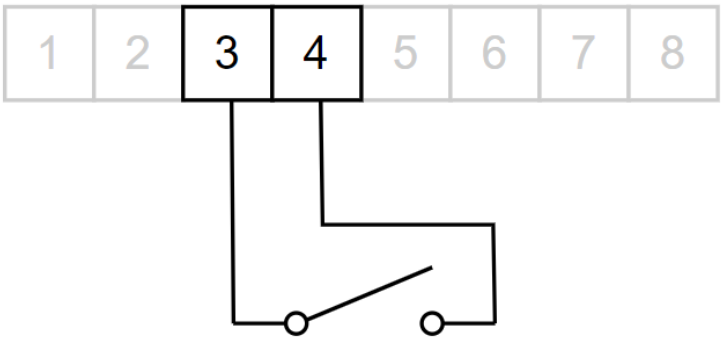


3.5. J7(Function 2)

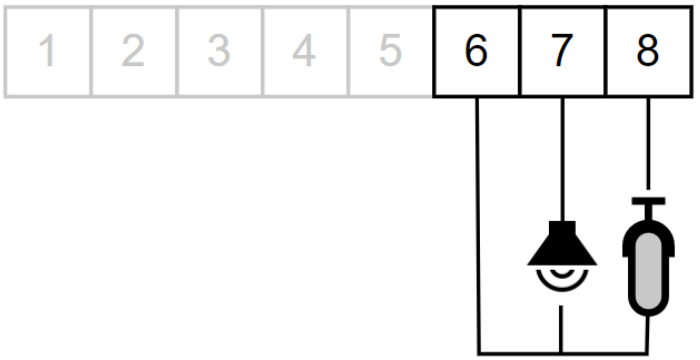
3.5.1. Indicator LED of RJ45



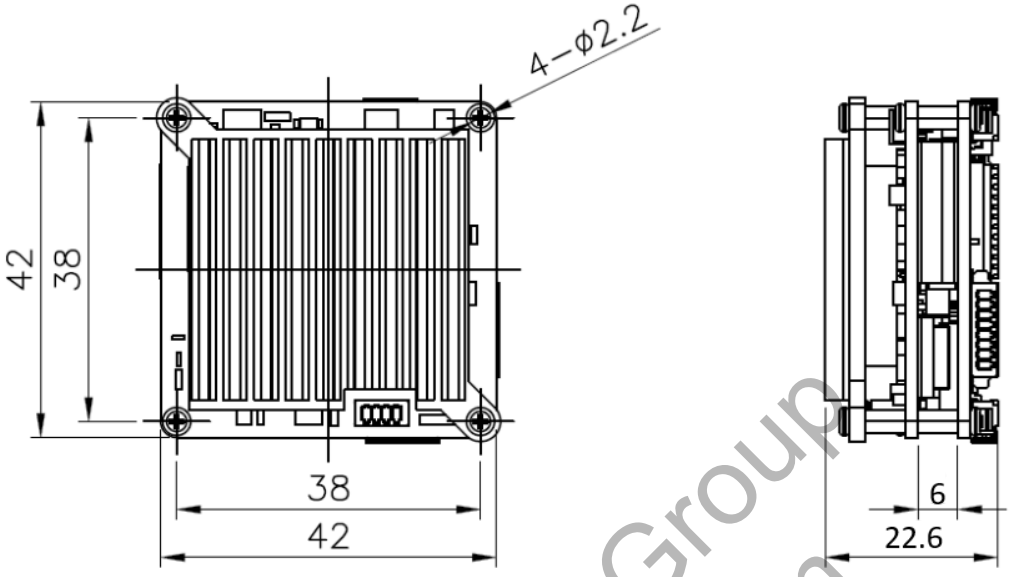
3.5.2. Reset of Factory default



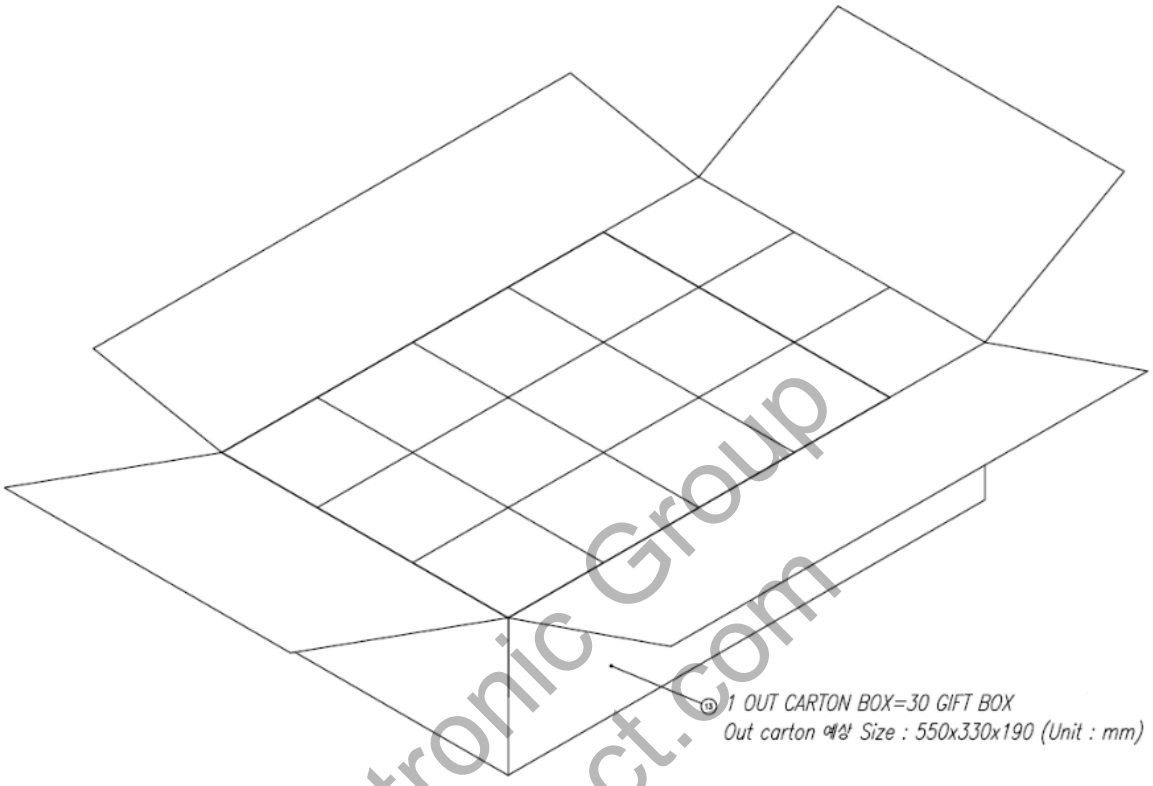
3.5.3. Audio Input / Output



4. Dimension(with heatsink)



5. Packaging



HTTP API

Version. 1.1.1

Version History

Version	Description
1.1.1	• Add system.module field to the API response. <GET> /app/v1/deviceInformation
1.1.0	• Add system.health.cpu field to the API response. <GET> /app/v1/deviceInformation • Masking specific pattern strings in sample data • Renaming the category("API List" -> "General API") • Add API related to Codec settings <GET,POST> /app/v1/media/video/encoder/resolution <GET,POST> /app/v1/media/video/encoder/bitrate
1.0.0	• Explain Authentication • Explain API List : Login,Logout,Check Sesison,JPEG,SDP,Reboot,Factory Reset • Explain Firmware Upgrade API List: Firmware Upload, Device Firmware Upgrade, Module Firmware Upgrade

Index

Copyright

Authentication Information

> For API calls that require authentication, authentication information is needed. Below is a description of the necessary information for authentication and how to handle the authentication token.

JWT

Session information required for API requests that need authentication is managed using JWT (https://en.wikipedia.org/wiki/JSON_Web_Token). A JWT consists of `header`, `payload`, and `signature`, which distinguish session information.

After entering the users `ID` and `Password` and successfully logging in, you obtain the token information as shown below.

```
{
    "token": "<token_value>"
}
```

For more details, refer to the `2. /app/v1/login` API description.

Assign authentication information to the HTTP Authorization header

For API calls that require authentication, the JWT token must be assigned to the HTTP Authorization header. It should be assigned in the format `Authorization: Bearer <JWT Token>`.

Below is an example of an HTTP request header with the authentication information assigned.

```
GET /app/v1/deviceInformation HTTP/1.1
...
Authorization: Bearer <token_value>
...
```

Access Permission Type

APIs have the following access permissions.

Permission	Details
noAuth	No authentication information is required.
Authentication	Requests must be made with a valid authentication token.
setup	Requests must be made with a valid authentication token that has setup permissions.
reboot	Requests must be made with a valid authentication token that has reboot permissions.

Regardless of the permission information, when authenticated with an `administrator` account, all APIs can be accessed.

API Response Status

Status Codes

Status Code	Description
200 OK	Successful state when the request is successful and there is response data.
204 OK	Successful state when the request is successful and there is no response data.
400 Bad Request	Invalid request data.
401 Unauthorized	Authentication required.
403 Forbidden	Login locked state.
500 Internal Server Error	Server encountered an error processing the request.

General API

1. /app/v1/login

POST /app/v1/login

Attribute	Details
Description	Authenticate user information and return the authentication token. If login attempts fail more than 5 times, login attempts for that ID are blocked for a certain period.
Permissions	Requires noAuth permission.
Request Content-Type	application/json
Response Content-Type	application/json

Request Fields

Field	Type	Description	Valid Values/Range	Optional
userId-ADMIN	string	User input ID	0 - 16	false
userPassword-1234	string	User input Password encrypted with SHA256 one-way encryption string (The request example below is an example of `asdfqwer` encrypted with SHA256 one-way encryption)	0 - 256	false

Request Example

```
{
    "userId": "<user_id>",
    "userPassword": "<password_encryption>"
}
```

Response Status

- 200 OK
- 400 Bad Request
- 401 Unauthorized
- 403 Forbidden

Response Fields

Field	Type	Description	Optional
token	string	Token for maintaining the authentication session	false

Response Example

```
{
    "token": "<token_value>"
}
```

2. /app/v1/logout

POST /app/v1/logout

Attribute	Details
Description	Discard the authentication token.
Permissions	Requires authentication.
Request Content-Type	-
Response Content-Type	-

Request Example

```
{}
```

Response Status

- 204 OK

3. /app/v1/session/check

PUT /app/v1/session/check

Attribute	Details
Description	Verify if the authentication token is valid.
Permissions	Requires authentication.
Request Content-Type	-
Response Content-Type	-

Response Status

- 204 OK : Authentication information is valid
- 401 Unauthorized : Authentication information is not valid

4. /app/v1/jpeg

GET /app/v1/jpeg

Attribute	Details
Description	Returns the Main stream Jpeg snapshot data at the requested time.
Permissions	Requires authentication.
Request Content-Type	-
Response Content-Type	image/jpeg

Response Status

- 200 OK
- 401 Unauthorized

Response Example

```
<JPEG BINARY DATA>
```

5. /app/v1/stream/sdp

GET /app/v1/stream/sdp

Attribute	Details
Description	Returns Main Stream RTSP SDP file information.
Permissions	Requires noAuth permission.
Request Content-Type	-
Response Content-Type	text/plain

Response Status

- 200 OK

Response Example


```
v=0
s=STREAMING_SERVER_Q
i=Session description protocol
t=0 0
a=type:broadcast
o=- 1 1 IN IP4 <IP>
c=IN IP4 0.0.0.0
m=video 0 RTP/AVP 98
a=rtpmap:98 H265/90000
a=control:rtsp://<IP>/<VIDEO_URI>
m=application 0 RTP/AVP 100
a=rtpmap:100 VND.ONVIF.METADATA/8000
a=control:rtsp://<IP>/<META_URI>
```

6. /app/v1/deviceInformation

GET /app/v1/deviceInformation

Attribute	Details
Description	Retrieves overall information about the device, including network, stream, and system.
Permissions	Requires authentication.
Request Content-Type	application/json
Response Content-Type	application/json

Response Status

- 200 OK
- 400 Bad Request
- 401 Unauthorized

Response Fields

Field	Type	Description
network.httpActivation	boolean	HTTP activation status
network.httpPort	int	HTTP port number
network.httpsActivation	boolean	HTTPS activation status
network.httpsPort	int	HTTPS port number
network.rtspPort	int	RTSP port number
network.ipv4.dhcpActivation	boolean	DHCP activation status for IPv4
network.ipv4.dnsDhcpActivation	boolean	DNS DHCP activation status for IPv4

Field	Type	Description
network.ipv4.ip	string	IPv4 address
network.ipv4.linklocal	string	Link-local IPv4 address
network.ipv4.gateway	string	IPv4 gateway address
network.ipv4.subnetmask	string	IPv4 subnet mask
network.ipv4.dns.primary	string	Primary DNS server for IPv4
network.ipv4.dns.secondary	string	Secondary DNS server for IPv4
network.ipv6.useType	int	IPv6 use type
network.ipv6.dnsDhcpActivation	boolean	DNS DHCP activation status for IPv6
network.ipv6.dns.primary	string	Primary DNS server for IPv6
network.ipv6.dns.secondary	string	Secondary DNS server for IPv6
network.ipv6.gateway	string	IPv6 gateway address
network.ipv6.ip	string	IPv6 address
network.ipv6.linklocal	string	Link-local IPv6 address
network.ipv6.prefixLength	int	IPv6 prefix length
system.version.firmware	string	Firmware version
system.modelName	string	Model name
system.mac	string	MAC address
system.time.year	int	Year
system.time.month	int	Month
system.time.day	int	Day
system.time.hour	int	Hour
system.time.min	int	Minute
system.time.sec	int	Second
system.time.isDst	boolean	Daylight saving time status
system.time.osdString	string	On-screen display time string
system.health.cpu.temperatureC	int	CPU Temperature (Celsius)
system.health.cpu.temperatureF	int	CPU Temperature (Fahrenheit)
stream[].activation	boolean	Stream activation status
stream[].rtspUrl	string	RTSP URL
stream[].resolution.width	int	Resolution width

Field	Type	Description
stream[].resolution.height	int	Resolution height
stream[].fps	int	Frames per second (stream 1)
system.module.connected	int	0 : Not connected, 1 : Connected
system.module.version	string	Zoom module version.
system.module.modelName	string	Zoom module model name.

Response Example


```
{
    "network": {
        "httpActivation": true,
        "httpPort": 80,
        "httpsActivation": false,
        "httpsPort": 443,
        "rtspPort": 554,
        "ipv4": {
            "dhcpActivation": false,
            "dnsDhcpActivation": true,
            "ip": "<IPv4>",
            "linklocal": "<IPv4>",
            "gateway": "<IPv4>",
            "subnetmask": "<IPv4>",
            "dns": {
                "primary": "<IPv4>",
                "secondary": "<IPv4>"
            }
        },
        "ipv6": {
            "useType": 0,
            "dnsDhcpActivation": true,
            "dns": {
                "primary": "",
                "secondary": ""
            },
            "gateway": "",
            "ip": "",
            "linklocal": "",
            "prefixLength": 64
        }
    },
    "system": {
        "version": {
            "firmware": "firmware_version"
        },
        "modelName": "modelname",
        "mac": "FFFFFFFFFFFF",
        "time": {
            "year": 2025,
            "month": 7,
```

```
            "day": 24,
            "hour": 9,
            "min": 10,
            "sec": 35,
            "isDst": true,
            "osdString": "24-07-2037 09:10:35"
        },
        "module": {
          "connected": 1,
          "version": "<zoom_module_version>",
          "modelName": "<zoom_module_model_name>",
        },
        "health": {
          "cpu": {
            "temperatureC": 0,
            "temperatureF": 0
          }
        }
    },
    "stream": [
        {
            "activation": true,
            "rtspUrl": "stream1",
            "resolution": {
                "width": 1920,
                "height": 1080
            },
            "fps": 30
        },
        {
            "activation": false,
            "rtspUrl": "stream2",
            "resolution": {
                "width": 1280,
                "height": 720
            },
            "fps": 30
        },
        {
            "activation": false,
            "rtspUrl": "stream3",
            "resolution": {
                "width": 640,
                "height": 360
            },
            "fps": 30
        }
    ]
}
```

7. /app/v1/system/reboot

PUT /app/v1/system/reboot

Attribute	Details
Description	Restart the device.
Permissions	Requires reboot permission.
Request Content-Type	application/json
Response Content-Type	application/json

Response Status

- 200 OK
- 401 Unauthorized

Response Fields

Field	Type	Description	Optional
result	int	Whether the authentication request was successful 1: Success	false

Response Example

```
{
    "result":1
}
```


8. /app/v1/system/factoryDefault

POST /app/v1/system/factoryDefault

Attribute	Details
Description	Reset the device to factory default state. If network information is updated, the HTTP request may be interrupted.
Permissions	Requires setup permission.
Request Content-Type	application/json
Response Content-Type	application/json

Response Status

- 204 OK
- 401 Unauthorized

9. /app/v1/media/video/encoder/resolution

GET /app/v1/media/video/encoder/resolution

Attribute	Details
Description	Retrieves the currently set resolution.
Permissions	Requires authentication.
Request Content-Type	-
Response Content-Type	application/json

Response Status

- 200 OK
- 401 Unauthorized

Response Fields

Field	Type	Description
stream	Array	1st : Main stream 2nd : Second stream 3rd : Third Stream
active	int	0 means disabled, 1 means enabled. The main stream is always active.
Width	int	Resolution Horizontal Pixels
Height	int	Resolution Vertical Pixels

Response Example

```
{
    "stream": [
        {
            "resolution": {
                "width": 1920,
                "height": 1080
            }
        },
        {
            "active": 1,
            "resolution": {
                "width": 640,
                "height": 360
            }
        },
        {
            "active": 0,
            "resolution": {
```

```
                "width": 640,
                "height": 360
            }
        }
    ]
}
```

POST /app/v1/media/video/encoder/resolution

Attribute	Details
Description	Sets the resolution of a specific stream.
Permissions	Requires setup permission.
Request Content-Type	application/json
Response Content-Type	-

Request Fields

Field	Type	Description	Optional
stream	string	Stream index 0 : Main stream 1: Second stream 2: Third Stream	N
width	int	Resolution Horizontal Pixels	N
height	int	Resolution Vertical Pixels	N

Request Example

```
{
    "stream": 0,
    "resolution" : {
        "width" : 1280,
        "height" : 720
    }
}
```

Response Status

- 204 OK
- 400 Bad Request
- 401 Unauthorized

10. /app/v1/media/video/encoder/bitrate

GET /app/v1/media/video/encoder/bitrate

Attribute	Details
Description	Retrieves the currently set bitrate.
Permissions	Requires authentication.
Request Content-Type	-
Response Content-Type	application/json

Response Status

- 200 OK
- 401 Unauthorized

Response Fields

Field	Type	Description
stream	Array	1st : Main stream 2nd : Second stream 3rd : Third Stream
active	int	0 means disabled, 1 means enabled. The main stream is always active.
bitrate	int	Bitrate of the stream

Response Example

```
{
    "stream": [
        {
            "bitrate": 8000
        },
        {
            "active": 1,
            "bitrate": 3000
        },
        {
            "active": 0,
            "bitrate": 3000
        }
    ]
}
```


POST /app/v1/media/video/encoder/bitrate

Attribute	Details
Description	Sets the bitrate of a specific stream.
Permissions	Requires setup permission.
Request Content-Type	-
Response Content-Type	application/json

Request Fields

Field	Type	Description	Optional
stream	int	Stream index 0 : Main stream 1: Second stream 2: Third Stream	N
bitrate	int	Bitrate of the stream	N

Request Example

```
{
    "stream" : 0,
    "bitrate" : 3000
}
```

Response Status

- 204 OK
- 400 Bad Request
- 401 Unauthorized

Firmware Upgrade API

> The firmware upgrade process follows a certain sequence. Below is a description of the general process of Firmware upgrade and the related API specifications.

Description Of Firmware Upgrade Process

Firmware upgrade mainly proceeds through the steps of `Firmware file upload` -> `Firmware upgrade request` -> `Firmware upgrade status check.`

1. Firmware file upload: Call the `1. /app/v1/upload/firmware` API to upload the firmware file.
2. Firmware upgrade request: If the correct firmware file has been uploaded, call each firmware upgrade request API to start the firmware upgrade.

3. Firmware upgrade status check: After the upgrade request, periodically check the firmware upgrade status to confirm it.

For detailed explanations of the firmware upgrade process, refer to the descriptions of each API.

1. /app/v1/upload/firmware

POST /app/v1/upload/firmware

Attribute	Details
Description	Upload the firmware file or zoom module firmware file. If the firmware file size is too large or an upload error occurs, a 500 error or failure message is returned.
Permissions	Requires noAuth permission.
Request Content-Type	application/octet-stream
Response Content-Type	text/plain
Response Status	**200 OK , 500 Internal Server Error**

Request Example

```
<FW FILE BINARY DATA>
```

Response Fields

Message	Description
Success to save the file. REQUEST_URI:/app/v1/upload/firmware	Success
Failed to save the file. REQUEST_URI:/app/v1/upload/firmware	Fail

Response Example

```
Success to save the file.
REQUEST_URI:/app/v1/upload/firmware
```

##

2. /app/v1/system/firmware

GET /app/v1/system/firmware

Attribute	Details
Description	Retrieves the firmware information of the device.
Permissions	Requires authentication.
Request Content-Type	application/json
Response Content-Type	application/json

Response Status

- 200 OK
- 401 Unauthorized

Response Fields

Field	Type	Description
version	string	Current firmware version

Response Example

```
{
    "version": "firmware_version"
}
```

3. /app/v1/system/firmware/upgrade

POST /app/v1/system/firmware/upgrade

Attribute	Details
Description	If the firmware file has been uploaded, start upgrading the device. Upon successful upgrade request, the device will restart and proceed with the upgrade. After restarting, the firmware upgrade results can be checked.
Permissions	Requires setup permission.
Request Content-Type	application/json
Response Content-Type	application/json

Request Fields

Field	Type	Description	Optional
withFactoryReset	boolean	If true, the device will be reset to factory default state upon successful firmware upgrade.	Y

Request Example

```
{
  "withFactoryReset": true
}
```

Response Status

- 200 OK : Request successful
- 400 Bad Request : File is missing or incorrect
- 401 Unauthorized : No permission

4. /app/v1/system/firmware/upgradeState

GET /app/v1/system/firmware/upgradeState

Attribute	Details
Description	After the device upgrade is completed, check the upgrade results.
Permissions	Requires noAuth permission.
Request Content-Type	application/json
Response Content-Type	application/json

Response Status

- 200 OK

Response Fields

Field	Type	Description
result	int	Whether the upgrade was successful -1 : Upgrade failed 1 : Upgrade Succeed
resultCaption	string	String describing whether the upgrade was successful

Response Example

```
{
    "result": 1,
    "resultCaption": "Upgrade succeeded."
}
```

5. /app/v1/system/zoom/firmware/version

GET /app/v1/system/zoom/firmware/version

Attribute	Details
Description	Check the connection status and firmware version of the zoom module.
Permissions	Requires authentication.
Request Content-Type	application/json
Response Content-Type	application/json

Response Status

- 200 OK
- 401 Unauthorized

Response Fields

Field	Type	Description
connected	int	Zoom module connection status 1 : Connected 0 : Disconnected
version	string	Zoom module firmware version

Response Example

```
{
    "connected": 1,
    "version": "Ver.02.14"
}
```

6. /app/v1/system/zoom/firmware/upgrade

PUT /app/v1/system/zoom/firmware/upgrade

Attribute	Details
Description	Request zoom module firmware upgrade. If the zoom module is not connected or the uploaded file is missing, the request is ignored.

Attribute	Details
Permissions	Requires setup permission.
Request Content-Type	application/json
Response Content-Type	application/json

Request Example

```
{}
```

Response Status

- 200 OK
- 401 Unauthorized

Response Fields

Field	Type	Description
result	int	Zoom module upgrade request result 1: Success

Response Example

```
{
    "result" : 1
}
```

7. /app/v1/system/zoom/firmware/upgradeState

GET /app/v1/system/zoom/firmware/upgradeState

Attribute	Details
Description	Check the firmware upgrade status of the zoom module. Remains in Ready state until the Upgrade start request After starting the Upgrade, check the progress in the Upgrading state. When checking the upgrade results (Success or Fail), the device will restart.
Permissions	Requires authentication.
Request Content-Type	application/json
Response Content-Type	application/json

Response Status

- 200 OK
- 401 Unauthorized

Response Fields

Field	Type	Description
status	int	Upgrade progress status 0: Ready 1: Upgrading 2: Upgrade Success 3: Upgrade Fail
cur	int	Current progress
tot	int	Maximum progress
bad	int	Not used

Response Example

```
{
    "status": 1,
    "cur": 7,
    "tot": 100,
    "bad": 0
}
```