Software Development

for Fluke RSE30/60, Pi33/36 Series Thermal Camera

User Guide

November 2022, Rev 1.00

Contents

Title **Page**

1 General

Fluke RSE/Pi is thermal camera series for industrial applications. All device control interfaces are based on RESTful Web Service. This document mainly describes the specific interface definitions and specifications required to control the device.

For video stream acquisition, please refer to the Demo program in SDK for secondary development.

2 Terms and Definitions

2.1 REST

REST (Representational State Transfer) is a set of architectural constraints and principles. Applications or designs that meet these constraints and principles are RESTful. It should be noted that REST is a design style, not a standard. REST is usually based on the use of HTTP, URI, XML (a subset of the Standard Generalized Markup Language), and HTML (an application of the Standard Generalized Markup Language), which are currently widely popular protocols and standards.

REST defines a set of architectural principles, according to which you can design system resource-centric Web services, including clients written in different languages processing and transferring resource status through HTTP. Given the number of Web services using it, REST has become the most important Web service design pattern in recent years. REST has a great impact on the Web. Because of its convenience, it has generally replaced the interface design based on SOAP and WSDL.

2.2 JSON

JSON (JavaScript Object Notation) is a lightweight data exchange format. It is based on a subset of ECMAScript (W3C JS Specification), which stores and represents data in a text format that is completely independent of the programming language. The concise and clear hierarchical structure makes JSON an ideal data exchange format. It is easy for people to read and write, and it is also easy for machines to parse and generate, and effectively improves the network transmission efficiency.

2.3 Blackbody Radiation

Any object has the property of continuously radiating, absorbing, and reflecting electromagnetic waves. The radiated electromagnetic wave is different in each wave band, that is, it has a specific spectral distribution. This spectral distribution is related to the characteristics of the object itself and its temperature, so it is called thermal radiation. To study the law of thermal radiation that does not depend on the specific physical properties of matter, physicists have defined Blackbody, an ideal object as the standard object for thermal radiation research.

2.4 Emissivity

Emissivity, also known as specific emissivity or emission coefficient, refers to the ratio of the radiation flux emitted by an object to the radiation flux of a blackbody at the same temperature. The emissivity of an object is related to its properties and surface conditions (such as roughness, color, etc.), and is a function of temperature and wavelength.

3 Basic Functions

3.1 Configuration Set

For devices, many parameters need to be configured by users, and users often need to use different parameters for different scenarios. It is a useful method for users to save and switch parameters in groups through configuration sets.

All parameters included in the configuration set are configuration set parameters, which can be saved and switched by users in groups. In addition, the configuration set can be also switched automatically by external events.

The current configuration of the device can be saved as a copy, which is named by the user. When the user needs the copy, it is called by the copy name (see /admin/cfgsets). The number of configurations sets you can save varies according to the device model.

3.2 Device Control Interface

The device interface uses the RESTful HTTP Web services, which can facilitate secondary development and debug, and all functions are implemented through a unified interface.

3.2.1 HTTP Service

Users can access the HTTP service of the device through port 10080, which acts as a proxy to communicate with independent internal function modules. Port 10081 is used to access stream data (RFC2616). The HTTP service will authenticate each access through HTTP Digest Access Authentication (RFC2617) protocol, see 3.7.

- Example HTTP Request

  `http://DEVADDR:10080/path-to-resource?arg1=1&arg2=2`

  `http://DEVADDR:10081/path-to-stream`

```
GET /path HTTP/1.1
Host: server
Accept: */*
Connection: keep-alive
Authorization:Digest username="guest",
realm="REST-API.xxx.cn",
nonce="dcd98b7102dd2f0e8b11d0f600bfb0c093",
uri="/admin/info",
qop=auth,
nc=00000001,
cnonce="0a4f113b",
response="6629fae49393a05397450978507c4ef1",
opaque="5ccc069c403ebaf9f0171e9517f40e41
```

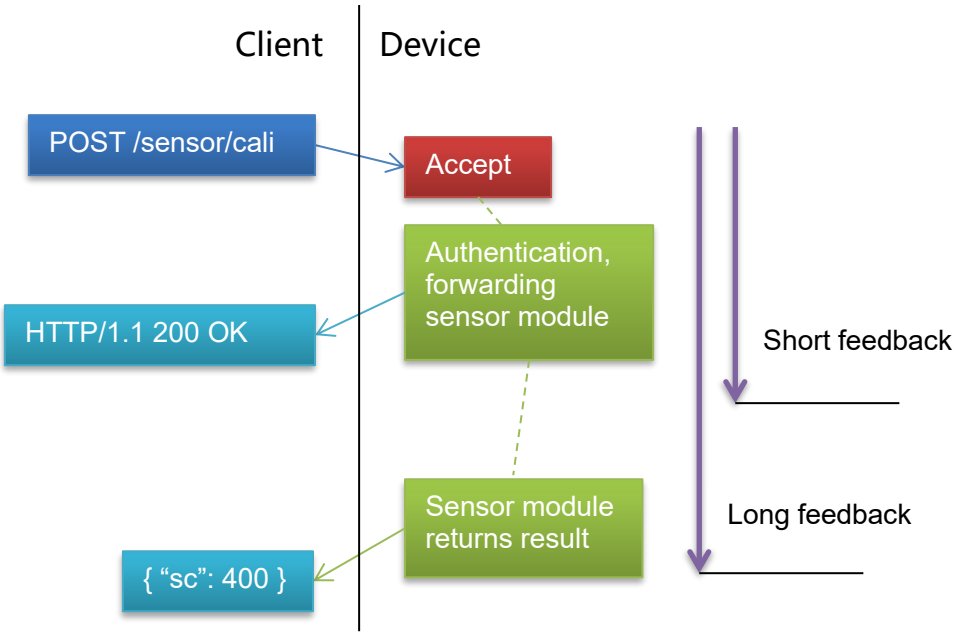
- The response carries the error code information

```
HTTP/1.1 200 OK
Content-Type: text/xml
Content-Length: 156
Connection: close
```

3.2.2 Response Extension

The device's response to the client extends the standard HTTP protocol, and the user will receive two return codes.

The HTTP service will immediately return the response message according to the definition of the HTTP protocol, but the internal module has not responded. To use the HTTP method to control the device, the HTTP connection will remain until the internal module returns or the connection times out. Therefore, the short feedback and long feedback process are implemented by returning the HTTP message header first, maintaining the connection, and returning the message body.



When the HTTP service successfully forwards the message, the client will receive a normal HTTP response with a status code of 200. When the internal return is correct, the returned content is the body defined by the interface. When the interface returns an error, it returns in JSON format:

```
{“sc”: 400}
```

The value of sc is the same as the HTTP response status code. See the Appendix.

3.2.3 Cache Redirect

Most interface commands transfer information in JSON format. When a large amount of data needs to be transferred, the access is redirected in the form of links according to the REST state transition style.

For example, to take a snapshot of the raw thermal image

GET /isp/snapshot

```
{ "path" : "/file/cache/9aubeEZTjKYBj23k" }
```

The device will save the image in the cache, which the client can obtain through command GET /file/cache/9aubeEZTjKYBj23k.

The cached data should be read as soon as possible because the size of the device cache is limited. When the cache usage reaches the limit, older data will be deleted, and the cached data will also be

lost after the device is powered off and restarted. Writing the cache too fast may cause data to be deleted before it is used.

3.2.4 Null Response

When there is no content returned from the interface, the device will return an HTTP response with an empty body. This is slightly different from the standard JSON, and the user will not receive null body.

The empty array "[]" and the empty object "{}" are treated as the same as null, when used as the return value. So some interfaces may return a zero-length body.

3.2.5 JSON Format Extension Definition

The string type in JSON used by the access device interface uses UTF-8 encoding instead of UNICODE-wide characters defined in the JSON standard.

3.2.6 JSON Configuration Tree

The configuration information inside the device will be accessed and modified by users through different URIs. In this document, the internal configuration information of the device is defined by JSON. Because JSON has a tree hierarchy, it is called a configuration tree. All the information that an interface can modify is called the **Full Configuration Tree** of the interface. The configuration tree obtained after removing some subtrees and nodes from the complete configuration tree is the **Partial Configuration Tree** of the interface. The empty tree is the partial configuration tree of any configuration tree. If a node in the configuration tree cannot be found in the full configuration tree with the same search path, it is called an **Unrelated Node**; otherwise, it is called a **Related Node**.

When the user reads the device information, the user will get the full configuration tree and additional read-only status information, and all the corresponding internal available information of the interface will be converted into JSON data.

When the user writes the device information, the user can write the full configuration tree or partial configuration tree. The device first verifies the syntax correctness of JSON, then verifies the correctness of the relevant node data on the JSON configuration tree, and finally loads the data into the internal configuration and takes effect. Configuration information not included in the configuration tree will remain in the state before writing, while irrelevant nodes written by users will be ignored, but will not cause the device to return an error.

The full configuration tree of `admin/ifaces/eth0` is defined as follows:

```
{
    "dhcp" : false,
    "dns" :
    [
        "192.168.1.1"
    ],
    "gateway" :
    [
        "192.168.1.1"
    ],
    "ip" : "192.168.1.115",
    "netmask" : "255.255.255.0"
}
```

The user can write its partial configuration tree:

```
{
    "dhcp" : true
}
```

At this time, the DHCP function of the device will be turned on, and other information will remain unchanged, but the static IP will no longer work.

When the user writes the following data:

```
{
    "ip": "192.168.1.12",
    "foo": "bar"
}
```

The device IP will be modified and the foo field will be ignored.

3.2.7 Common Status Code

The following are the common error status codes of device interfaces, which apply to all interface paths

Status Code	Description
401	Insufficient user permissions to access this interface
500	Device internal exception, usually I/O exception (such as insufficient memory), or function module exception
400	The parameters of the input are incorrect

3.2.8 Device Resource Naming Convention

Users often need to create some resources on the device and read and write this resource information through the device interface, such as creating user names, creating new temperature measurement areas, creating new alarm rules, etc. These are identified by corresponding names.

For example, to create a new user:

```
PUT /user/user1
```

The new username is user1, and the name of the username should meet the requirements of the URL (RFC3986).

The resource names created through the device interface should conform to the following rules:

- ASCII encoded printable characters
- Cannot use reserved characters including : / ?# [ ] @ !$ & ” ' * ( ) , = + ;
- The length of the resource name should be less than 1024 characters unless otherwise specified.

3.3 Basic Concepts of Thermal Camera

All objects radiate electromagnetic waves, which are called thermal radiation. The thermal camera receives these thermal electromagnets and converts them into digital images. Different from common color photos, thermal radiation images have a wide dynamic sampling range. Generally, 14-16 bit integers are used to represent a radiation measurement value, which is called an AD value (because it is a digital value obtained through Analog-to-Digital conversion).

According to the principle of blackbody radiation, the temperature of an object can be inferred from the amount of thermal radiation, which is the principle of temperature measurement of thermal cameras. The relationship between each AD value and the blackbody temperature will be carefully calibrated during the production of thermal cameras and stored in each device in the form of a temperature mapping table (LUT).Each device has multiple temperature mapping tables, which need to be correctly selected before temperature measurement.

3.3.1 Temperature Range

If a higher temperature is to be measured, the range of the thermal camera needs to be changed. Selecting a suitable range of the thermal cameral can ensure that the image sampling will not be saturated due to overexposure.

The range list can be obtained through the following interface:

```
GET /sensor/t-range
```

The list in devices of different models may be different, please consult the supplier

The image quality will be different in different ranges because higher temperature response capability will make it less clear when using low-temperature objects.

3.3.2 Lens

Another factor that affects the temperature measurement is the len used by the thermal camera. The lens of each thermal imager will be calibrated before delivery, so please do not change the matching of the lens and device at will, which may affect the temperature measurement accuracy.

If your model supports multiple lenses, you should configure the device when changing lenses.

The list of supported lenses can be obtained as follows:

```
GET /sensor/lens
```

The list in devices of different models may be different, please consult the supplier

3.3.3 Switching Temperature Mapping Table

Only when the measuring range and lens are correctly selected can the temperature be measured correctly. The selection of measuring range and lens is made through the list number, which starts from 1. If the second measuring range and the first lens are selected for temperature measurement:

PUT /sensor

```
PUT /sensor/jconfig

{
   "selected-lens" : 1,
```

```
    "selected-t-range" : 2
}
```

The user can get the relationship between these ranges and lens combinations, and the temperature mapping table:

```
GET /sensor/luts
```

Not all combinations have a corresponding temperature mapping table, which needs to be confirmed with the supplier before purchasing.

Each temperature mapping table can be downloaded separately:

```
GET /sensor/luts/[idx]?list
```

The temperature mapping table currently selected according to the range and lens will affect the temperature output by the temperature measuring tool. If you need to know the temperature mapping table currently used, you can use the interface:

```
GET /sensor/lut
```

The range and lens combination selected by the user through PUT /sensor/jconfig may not include the corresponding factory temperature mapping table, so an error will be returned when obtaining the current temperature mapping table.

3.3.4 Emissivity

The accurate temperature measurement of the thermal imager depends not only on the acquired thermal radiation AD value but also on the properties of the object to be measured. The most important property is emissivity. The blackbody is a theoretical model with the emissivity of 1. The emissivity of all objects is less than 1, so it needs to be compensated through parameter settings.See 3.4 for specific setting methods.

3.3.5 Use of Temperature Mapping Table

The temperature mapping table obtained from the device is a lookup table for converting AD value to temperature.

```
[
    {
        "r" : 7000,
        "t" : 10
    },
    {
        "r" : 7500,
        "t" : 20
    },
    {
        "r" : 8000,
        "t" : 30
    }
```

```
]
```

Where r represents the AD value and t represents the corresponding Celsius temperature.

The order of the temperature mapping table is always from the lowest AD value to the highest. If the AD value to be converted is not included in the temperature mapping table, the adjacent AD values can be used for interpolation calculation.

Assume that the adjacent terms of AD value r_a in the temperature mapping table are r_0和r_1, and their corresponding temperatures aret_0和t_1

Then the temperature can be obtained by one interpolationt_a

$$t_a = t_0 + \frac{t_1 - t_0}{r_1 - r_0}(r_a - r_0)$$

All calculated temperatures represent blackbody temperatures with an emissivity of 1.

3.4 Built-in Temperature Measuring Tool

Each pixel of the thermal radiation image can be used as the temperature sampling point, but the amount of data is large. The device provides some temperature tools to facilitate the user to select the location of interest for data collection.

The device supports three temperature measurement objects: point, region, and polyline. The device has a set of global temperature measurement compensation parameters. Each temperature measurement object can inherit the global parameters or modify each parameter separately.

3.4.1 Temperature Measurement Compensation Parameters

The temperature measurement compensation parameters supported inside the device include:

- emissivity, Emissivity

  It is related to the property of the object to be measured and has the greatest impact on the temperature measurement results. The emissivity of different materials can be found in the emissivity table of common materials.

- ambient-t, Ambient Temperature

  In addition to its radiation, the rest of the measured object is mostly the reflection of radiation from other objects in the environment. For unpolished general objects, this radiation can be considered as the diffuse reflection of the ambient temperature.

- reflect-t, Reflection Temperature

  When the measured object is surrounded by a high-temperature object and this effect cannot be avoided, additional correction is required for this strong interference, that is, compensation is made by the reflect-t parameter.

- distance, Distance

The distance factor will become obvious only when the measured object is far away (thousands of meters away). At this time, the absorption of thermal radiation by the atmosphere needs to be compensated. The unit of distance is meter.

- offset Offset

  Due to systematicness or other factors not involved, the deviation of temperature measurement value can be adjusted directly. To modify this parameter, you need to refer to the standard blackbody.

- lens-transmissivity, Lens Transmissivity

  When there are additional optical components in front of the lens of the device (such as the device being placed in a waterproof bin), the absorption of thermal radiation by the optical components will affect the temperature measurement results and needs to be compensated. To change this parameter, the user should refer to the standard blackbody or contact the supplier for calibration during production.

These temperature measurement parameters will be applied to each temperature measurement object, and the device will retain a set of global parameters as default values, which can be modified through the following interface:

```
PUT /isp/instrument/jconfig

{
    "ambient-t" : 20,
    "distance" : 1,
    "emissivity" : 0.92
}
```

To obtain complete global parameters, you can use this interface:

```
GET /isp/instrument/jconfig
```

3.4.2 Temperature Measurement Point

The user can set temperature measuring points at any position inside the thermal imaging surface

```
PUT /isp/instrument/objects/points/p1

{
    "emissivity" : 0.97,
    "pos" :
    {
        "x" : 150,
        "y" : 50
    },
    "label":"My point 1"
}
```

The above command can set a temperature measurement point at the 150th column and the 50th row of the image, and change the emissivity to 0.97, which will update the global emissivity value.

The label attribute is used to display a more readable object name on OSD.

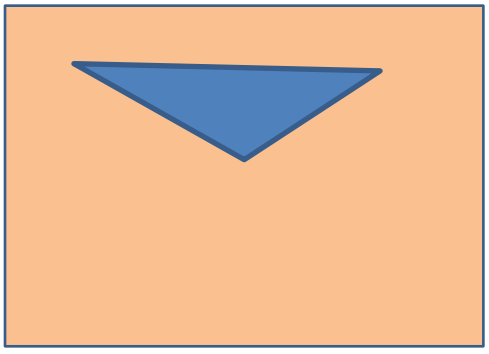
3.4.3 Temperature Measurement Region

The temperature measurement region can be defined by polygons or region masks,

The number of vertices of the polygon shall be no less than 3 and no more than 50. The polygon region is added as follows:

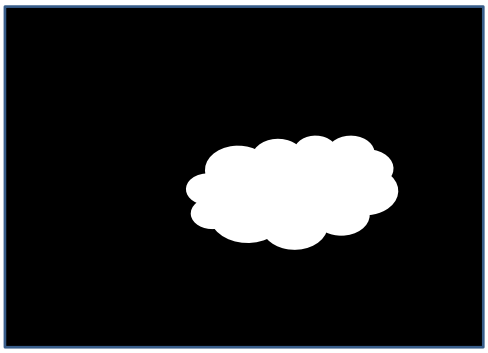
```
PUT /isp/instrument/objects/regions/r1

{
    "polygon" :
    [
        {
            "x" : 30,
            "y" : 50
        },
        {
            "x" : 300,
            "y" : 60
        },
        {
            "x" : 150,
            "y" : 120
        }
    ],
    "label" : "My Region1"
}
```

The above command will add a triangle region:



Note that the device only supports adding convex polygons.

The mask region will provide users with a more flexible way to define the region. Users can define this region by uploading a mask image of a single connected region. The size of the mask image should be the same as the resolution of the thermal imager. Each pixel is represented by an 8-bit byte. The non-zero value represents within the region, and zero represents outside the region. Also, the mask region must contain only one 8-connected region.



For a 384x288 device, the user needs to create a 384x288 byte cache, set the pixels in the corresponding region to non-zero values, and upload them to the device cache:

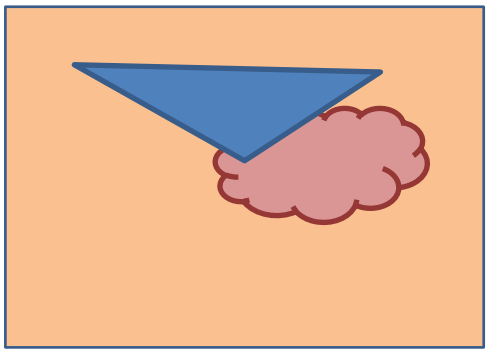
```
POST /file/cache
```

After the upload is successful, the device will return a cache token:

```
{
    "path" : "8kn4VBYmAw0XKUvI"
}
```

Now you can create a mask region:

```
PUT /isp/instrument/objects/regions/m1

{
    "mask":"8kn4VBYmAw0XKUvI"
    "label" : "My Mask Region1"
    "layer" : -1
}
```



In the above command, we used the layer attribute, which is represented by an integer. Each region object has a layer attribute. The default value is 0. When there is an overlap between regions, the region with the same layer attribute determines the level of the region according to the order of addition. The part of the region obscured by other regions will not participate in the temperature measurement of the region object. The higher the layer value, the closer the region is to the top. For example, the layers of regions a, b, c, and d we added are 4, 0, -1, and 0 respectively, so their levels are a, b, d, and c from top to bottom

3.4.4 Temperature Measurement Polyline

In some special applications, you only need to sample temperature along a linear region, and then you can use polyline objects.

The polyline object defines a polyline region connected end to end through a series of vertices, and the device collects the temperature of the pixel grid (Bresenham algorithm) along the polyline.

The number of vertices of the polyline shall not be less than 2 and shall not exceed 50. Adding a polyline is similar to adding a polygon:

```
PUT /isp/instrument/objects/lines/l1

{
    "polyline" :
    [
        {
            "x" : 30,
            "y" : 50
        },
        {
            "x" : 300,
            "y" : 60
        },
        {
            "x" : 150,
            "y" : 120
        }
    ],
    "label" : "My Line1"
}
```

3.4.5 Global Region

The simplest measurement is to get the highest and lowest temperature points on the whole image

```
GET /isp/instrument/objects/global?value
```

The global temperature value uses the global temperature measurement compensation parameter for temperature correction.

3.4.6 OSD Display

Each temperature measurement region will be displayed in the video stream by default. If you want to turn off the OSD display of an object, you can do so in the following way:

```
PUT /isp/instrument/objects/points/p1

{
    "osd-visible":false
}
```

Note that if the p1 object does not exist on the device, the above command will fail because the above command uses the partial configuration tree. If the object does not exist, the full configuration tree does not exist.

3.4.7 Obtain Temperature Measurement Data

Each object can obtain temperature measurement data separately:

```
GET /isp/instrument/objects/lines/l1?value
```

In this way, the user can get the highest and lowest temperature and AD values on this polyline l1

The user can connect the tag stream to obtain the real-time measured value of the temperature measurement object through the 10081 port:

```
/tag/values
```

Refer to 3.6

3.4.8 Sampling Rate

The temperature measuring tool can change the rate of sampling temperature,

```
PUT /isp/instrument/sample-rate

2
```

The sampling rate refers to the number of samples taken in one second. It is invalid to set the rate higher than the frame rate of the thermal imager.

3.4.9 Read Temperature through Serial Port

After setting the required region through API, the user can bind the temperature measurement object to the MODBUS data address to read the temperature through the serial port. Refer to 6.3

```
PUT /isp/instrument/objects/regions/r1

{
    "mb-slot":1
}
```

The above command binds the region object r1 to the MODBUS data block 1, accesses the data block 1 through the external serial port of the device, and reads the highest temperature and lowest temperature in the region r1.

3.4.10 Temperature Alarm

The device has an automatic alarm, which can output the alarm level signal through the optocoupler. The user can modify the alarm threshold by modifying the alarm-t attribute of the temperature measurement object. When the temperature measurement object is initially created, the alarm is turned off by default.

```
PUT /isp/instrument/objects/regions/m1
```

```
{
    "alarm-t":120
}
```

Alarm signal output will be triggered when the maximum temperature of the m1 region exceeds 120 ℃.

You can also customize other alarm methods. Please contact the supplier for specific methods.

3.5 Capture through External Trigger

The device can control the capturing of thermal video through external trigger signals. It is worth noting that thermal image capture needs to maintain a stable sampling frequency to obtain a stable image. The image obtained by triggering is the first image obtained in the continuous sampling after the trigger time.Therefore, external triggering can only ensure limited sampling synchronization.

Switch to triggering capture mode:

```
PUT /capture/mode

"trigger"
```

In addition to external signals, the capture can be triggered through software:

```
POST /capture/trigger
```

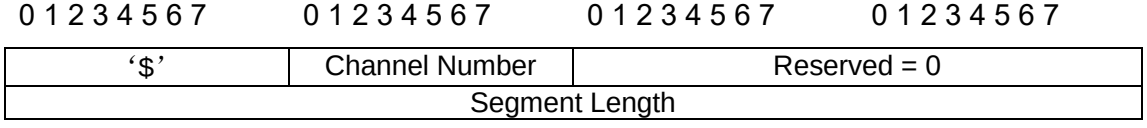
Refer to the Installation Manual for the connection mode of external signals. Trigger events are generated by edge change detection.

3.6 HTTP Based Real-time Data Stream

In addition to redirecting file data, the device also supports real-time data streams.

According to the specification of the HTTP protocol, when there is no header information defining the length of the body such as Content-Length defined in the HTTP header, the end of the body is marked by the disconnection (the Connection header information must be close).Therefore, the data stream can be continuously sent to the client in this way until the client stops receiving or the device is reset.

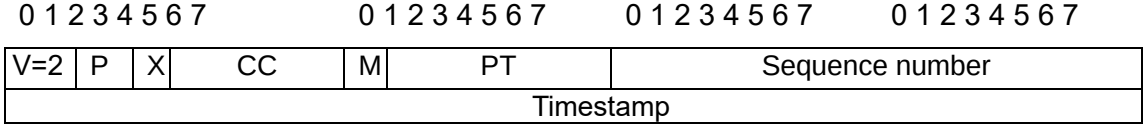
The data stream is composed of continuous data frames. To delimit the data frames, the following 8-byte segmented header is defined referring to RTSP/TCP protocols:



- The channel number is used to distinguish different data frames when the data stream is multiplexed.
- The segment length refers to the data frame length excluding the segment header, and theoretically, the maximum supported data frame size is 4G bytes.
- The client distinguishes data frames through segmented headers and needs to determine its receive buffer size through the data stream description information in the device interface.

3.6.1 Real Time Data Frame

Most of the data streams provided by the device are real-time data streams. Each data frame contains timing information. The real-time data frame will use the SRTP header:



- Consistent with RTP definition (RFC3550)
- The unit of timestamp is determined by the specific data stream attribute

3.6.2 Video Streaming Service

The device uses an separate service port to provide video streaming service to the client. The port number is 10081. Each video stream will have a corresponding video stream attribute included in the STREAM control module. For example, the stream attribute of the /video/raw video stream can be obtained through the GET /stream/video/raw command.

3.7 Device Access Control

Access control on device functions can be divided into network layer restrictions (TBD) and application layer restrictions. The network layer restricts access by filtering the IP address and MAC address of the client (TBD).

Application layer restrictions include username control and production authorization.

3.7.1 Username Access Control

The user uses the device through the username for permission management. When the device is in the factory state, the default user list is empty, and the device can be accessed anonymously. Anonymous users instead of admin/123456 can be used to facilitate demonstration and development, and also to clearly distinguish device access control modes.

When there is no username list (GET /admin/users) in the device, the device is in **anonymous access mode** or **no access control mode**, and all interfaces can be accessed. When the user needs security, adding a username list will switch the device status to the **username access control mode**, and then the user cannot access the device anonymously.

You can add usernames and change passwords through the PUT /user/[username] interface.

3.7.2 Permission Group

In the username access control mode, different usernames need to be given permission groups, and different permission groups correspond to different functional permissions. When the user list is created, the first user is given the root group by default, and the root user can be deleted only after all other users are deleted. When the user list returns to empty, the device enters the no access control mode.

Users in higher permission groups have the permissions of all lower permission groups, so the permission item in the interface definition below is the group name of the lowest accessible permission group. The permissions in the following table are arranged from high to low order:

Permission	Description

Group Name	
root	This group can contain only one user. It can add and change other users and reset passwords
manager	All actions except user management
operator	Operator, see interface definition for permissions
viewer	Viewer, see interface definition for permissions

3.7.3 Password Protection

HTTP Digest Authentication (RFC 2617) is used to verify the access of each interface of the device, which can effectively protect the password.

When creating a user and password, the software can encrypt the plaintext password once and then write it to the device. The subsequent login access should use this encryption method to form an equivalent password.

4 Application Programming Interface (API) Definition

All parameter configuration class interfaces in the definition use a full configuration tree, while users can use a partial configuration tree for practice use.

When using interfaces to get data, the user will get some read-only status nodes in the configuration tree. These nodes are not included in the full configuration tree and cannot be modified.

4.1 Command Definition

The device is accessed through the HTTP protocol, and each command consists of a resource path (URL) and an HTTP method. HTTP methods include: GET, PUT, DELETE, POST

When describing a command, we only reserve the path part of the URL:

```
GET /admin/info
```

When the command contains parameters, they are indicated in italics:

```
PUT /user/user_name
```

In the actual command, replace the parentheses with real parameters. The parameter naming rules should conform to 3.2.8

```
PUT /user/somebody
GET /sensor/luts/2?list
```

The body formats of the request and response are listed separately in the table, and each available field is defined with a description. For commands corresponding to GET / PUT, the field definition is the same for both operations.

4.2 Device Management ADMIN

This function module includes device information, device network interface, user management, access control, firmware upgrade, and other device management functions. This module is not included in the configuration set.

4.2.1 Get Device Information

GET /admin/info	**Permission:ALL**
Request and Response	**Description**
	None
{ "create-date" : "", "device-fw" : { "build" : "20161202", "name" : "iir-fw", "tag" : "rc7", "version" : [3, 0, 0, 101] }, "device-id" : "gl9ARQ1Fc/dEzDuOwQvDsA==", "device-model" : "xxxxxx", "device-sn0" : "00000", "device-sn1" : "",	device-id Device ID device-fw Firmware information version Firmware version tag Version tag build Generation date name Firmware name device-model Device Model device-sn0 Device serial number device-tag0 Device tag (whether the device is a prototype) device-tag1 device-tag2 sensor Sensor type sensor-sn Sensor serial number

"device-tag0" : "sample", "device-version" : "1.0", "hardware" : "a", "hardware-version" : "2321", "modify-date" : "", "seal-date" : "", "sensor" : "", "sensor-sn" : "2131123", "update-counter" : 4 }	hardware Hardware type hardware-version Hardware version create-date Device creation date modify-date Last modification date of device information seal-date Device production date update-counter Device information update count
Status Code	
200	

4.2.2 Get Device Time

GET /admin/clock	**Permission:ALL**
Request and Response	**Description**
	None
"20161202071928"	The time is represented by a compressed numerical string
Status Code	
200	

4.2.3 Set Device Time

PUT /admin/clock	**Permission:MANAGER**
Request and Response	**Description**
"20161201071900"	The time must be represented by 14 digits, and the time less than seconds cannot be set
Status Code	
200	
400	Incorrect time format

4.2.4 Get NTP Server List

GET /admin/ntp	**Permission:MANAGER**
Request and Response	**Description**
	None
["pool.ntp.org", "time.nist.gov"]	
Status Code	
200	

4.2.5 Set NTP Server List

PUT /admin/ntp	Permission:MANAGER
Request and Response	**Description**
["pool.ntp.org", "time.nist.gov"]	Empty list will turn off NTP function
Status Code	
200	

4.2.6 Get Device Network Information

GET /admin/ifaces	Permission:ALL
Request and Response	**Description**
	None

{
"eth0" :
{
"dhcp" : false,
"dns" :
[
"192.168.1.1"
],
"gateway" :
[
"192.168.1.1"
],
"ip" : "192.168.1.115",
"mac" : "06:00:01:FE:4F:29",
"netmask" : "255.255.255.0",
"prefix" : 24
}
}
	・ip Device IPv4 Address
	・dhcp Whether DHCP protocol is used to automatically obtain network parameters
	・dns Domain Name Server Address List
	・gateway Gateway List
	・netmask Netmask
	・mac Network card address

GET /admin/ifaces/eth0	**Permission:ALL**
	None
{ "dhcp" : false, "dns" : ["192.168.1.1"], "gateway" : ["192.168.1.1"], "ip" : "192.168.1.115", "ipv6" : "fe80::400:1ff:fefe:4f29/64", "mac" : "06:00:01:FE:4F:29", "netmask" : "255.255.255.0", "prefix" : 24	Ditto

}	
Status Code	
200	
404	Network device not found

4.2.7 Set Device Network Information

PUT /admin/ifaces/*eth0*	**Permission:MANAGER**
Request and Response	**Description**

{
"dhcp" : false,
"dns" :
[
"192.168.1.1"
],
"gateway" :
[
"192.168.1.1"
],
"ip" : "192.168.1.115",
"netmask" : "255.255.255.0"
}
	The listed network parameters can be modified

	None
Status Code	
200	
400	Network parameter error

4.2.8 Get User List

GET /admin/users	**Permission: MANAGER**
Request and Response	**Description**
	None

{
"admin" : "root",
"user1" : "viewer"
}	

Status Code	
200	

4.2.9 Add and Update Users

PUT /user/*user_name* ・ user_name New user name	**Permission: ROOT**
Request and Response	**Description**

{
"pw": "EzDuOw",
“group”:”operator”
}
	・ pw Password
	・ group root, manager, operator, viewer

	None

Status Code	
200	Success
403	The user already exists or the password is too short

4.2.10 Remove Users

DELETE /user/*user_name*	Permission: ROOT
Request and Response	**Description**
	None
	None
Status Code	
200	
403	

4.2.11 Get Boot ID

GET /admin/boot-id	Permission: ALL
Request and Response	**Description**
	None
"hVHruyn2TY6j7zyAfZ2hSg=="	BASE64 encoding of UUID
Status Code	
200	

4.2.12 Update Firmware

POST /admin/update	Permission: MANAGER
Request and Response	**Description**

{
"path" : "/file/cache/9aubeEZTjKYBj23k"
}
	・ path Firmware path uploaded to the device cache
	・ Executing this command will restart the device

	None (The device starts to restart after the command returns response)
Status Code	
200	
404	Firmware not found
POST /admin/update	**Permission: MANAGER**
Request and Response	**Description**

{
"link":"47.88.34.128",
"port":80
"path":"/upgrate/firmware/iir-fw-1.0.0.0.bin"
}
	・ link Remote server address
	・ port Server port
	・ path Server path

	None (this operation will start the asynchronous firmware download)
Status Code	
202	Start downloading firmware
503	Downloading firmware

4.2.13 Reboot Device

POST /admin/reboot	**Permission: ALL**
Request and Response	**Description**
	None
	None (The device starts to restart after the command returns response)
Status Code	
200	

4.2.14 Restore Factory Configuration

POST /admin/reset?factory **POST /admin/reset?restore** ・ factory Do not change network parameters ・ restore Restore factory network parameters at the same time	**Permission: ALL**
Request and Response	**Description**
	None
	None
Status Code	
200	

4.2.15 Save Current Configuration Set

PUT /admin/cfgsets/*cfg_name* ・ cfg_name Configuration set name	**Permission: OPERATOR**
Request and Response	**Description**
	None
	None
Status Code	
200	
403	Exceeded the maximum number of saved configuration sets

4.2.16 Get Configuration Set List

GET /admin/cfgsets	**Permission: OPERATOR**
Request and Response	**Description**
	None
["cfg1", "cfg2", "cfg3"]	None

Status Code	
200	
403	Exceeded the maximum number of saved configuration sets

4.2.17 Load Configuration Set

POST /admin/cfgsets/*cfg_name* · cfg_name Configuration set name		Permission: OPERATOR
Request and Response	**Description**	
	None	
	None	
Status Code		
200		
404	Configuration set does not exist	

4.3 SENSOR

The sensor module controls the work of the thermal imaging core, including obtaining core information, switching the temperature measurement range of the core, calibration method, mirror, etc.The core uses an integer index to reference the list item. The index value should be greater than 0, and the 0 item in the list is null.

4.3.1 Get a list of calibration modes

GET /sensor/cali-modes		Permission: VIEWER Configuration Set: No
Request and Response	**Description**	
	None	
["auto", "manual"]	· auto Auto calibration · manual Manual calibration	
Status Code		
200		
400		

4.3.2 Get Current Calibration Mode

GET /sensor/cali-mode		Permission: VIEWER Configuration Set: Yes
Request and Response	**Description**	
	None	
“auto”		
Status Code		
200		
400		

4.3.3 Set Current Calibration Mode

PUT /sensor/cali-mode	Permission: OPERATOR Configuration Set: Yes
Request and Response	**Description**
“manual”	Only values in the calibration mode list are allowed
Status Code	
200	
400	Unknown calibration mode

4.3.4 Perform a Calibration

POST /sensor/cali	Permission: VIEWER Configuration Set: No
Request and Response	**Description**
	None
Status Code	
200	
400	

4.3.5 Get Sensor Mirror Mode

GET /sensor/mirror	Permission: VIEWER Configuration Set: Yes
Request and Response	**Description**
	None

{
"h" : 0,
"v" : 0
}
	· h Whether to mirror horizontally
	· v Whether to mirror vertically

Status Code	
200	

4.3.6 Set Sensor Mirror Mode

PUT /sensor/mirror	Permission: OPERATOR Configuration Set: Yes
Request and Response	**Description**

{
"h" : 0,
"v" : 0
}
	· h Whether to mirror horizontally
	· v Whether to mirror vertically

	·
Status Code	
200	

4.3.7 Get Sensor Image Dimension

GET /sensor/dimension	Permission: VIEWER Configuration Set: No
Request and Response	**Description**

{
"h" : 80,
"w" : 80
}
	Dimension is in pixels

Status Code	
200	

4.3.8 Get Temperature Range List

GET /sensor/t-range	Permission: VIEWER Configuration Set: No
Request and Response	**Description**

[
null,
{
"high" : 150,
"low" : -20
},
{
"high" : 300,
"low" : 0
},
{
"high" : 650,
"low" : 300
}
]
	Item 0 must be null

Status Code	
200	

4.3.9 Get Lens List

GET /sensor/lens	Permission: VIEWER Configuration Set: No
Request and Response	**Description**

[
null,
{
"model" : "default"
}
]
	Item 0 must be null

Status Code	

200	

4.3.10 Get Factory Temperature Lookup Table

GET /sensor/luts	**Permission: VIEWER** **Configuration Set: No**
Request and Response	**Description**
	None

[
null,
{
"lens" : 1,
"t-range" : 3
},
{
"lens" : 1,
"t-range" : 2
},
{
"lens" : 1,
"t-range" : 3
}
]	

Status Code	
200	
400	

GET /sensor/luts/*idx* · idx List Index	**Permission: VIEWER** **Configuration Set: No**
Request and Response	**Description**
	None

{
"lens" : 1,
"t-range" : 3
}	

Status Code	
200	
404	

4.3.11 Select Factory Temperature Lookup Table

PUT /sensor/jconfig	**Permission: OPERATOR** **Configuration Set: Yes**
Request and Response	**Description**

{
"selected-lens" : 1,
"selected-t-range" : 2
}
	The index item should be in the supported list, and the index starts from 1

Status Code	
200	

400	Index out of bound
404	Temperature list does not exist

4.3.12 Get Index of Current Factory Temperature Lookup Table

GET /sensor/lut	**Permission: VIEWER** **Configuration Set: No**
Request and Response	**Description**
	None
1	
Status Code	
200	
404	Failed to switch the temperature lookup table last time

4.3.13 Get Factory Temperature Lookup Table

GET /sensor/luts/*idx*?list ・ idx Index of the temperature lookup table	**Permission: VIEWER** **Configuration Set: No**
Request and Response	**Description**
	None
[{ "r" : 7455, "t" : -24.700001 }, { "r" : 7504, "t" : -19.800001 }, { "r" : 7617, "t" : -9.8000002 }, …	・ r AD value ・ t Temperature in Celsius
Status Code	
200	
404	

4.4 Image Capture CAPTURE

4.4.1 Get Current Image Capture Mode

GET /capture/mode	**Permission: VIEWER** **Configuration Set: Yes**
Request and Response	**Description**
	None

"continuous"	
Status Code	
200	

4.4.2 Set Image Capture Mode

PUT /capture/mode	**Permission: VIEWER** **Configuration Set: Yes**
Request and Response	**Description**
"trigger"	・ trigger Capture through external signal trigger ・ continuous Continuous capture
Status Code	
200	

4.4.3 Trigger a Capture

POST /capture/trigger	**Permission: VIEWER** **Configuration Set: Yes**
Request and Response	**Description**
	None
	None
Status Code	
200	

4.5 Image Processing ISP

The image processing module includes several sub-modules, mainly for the visualization of the raw thermal image data, analysis of temperature data, intelligent analysis, etc.

4.5.1 Get Image Processing Parameters

GET /isp/jconfig	**Permission: VIEWER** **Configuration Set: Yes**
Request and Response	**Description**
	None

{
"black-level" : 0,
"contrast" : 1,
"method" : "tpe",
"roi" : "full",
}	

Status Code	
200	

4.5.2 Set Image Processing Parameters

PUT /isp/jconfig	Permission: OPERATOR Configuration Set: Yes
Request and Response	**Description**

{
"black-level" : 0,
"contrast" : 1,
"method" : "tpe",
"roi" : "full"
}
	・ black-level Black level [-30 ~ 30]
	・ contrast Contrast (0, 3.0)
	・ method Automatic gain method: linear, hp, tpe
	・ roi Automatic gain calculation region: full, middle

Status Code	
200	

4.5.3 Get Automatic Gain Mode

GET /isp/agc-mode	Permission: VIEWER Configuration Set: No
Request and Response	**Description**
	None
"continuous"	Mode: ・ once Perform automatic gain once ・ continuous Continuous automatic gain
Status Code	
200	

4.5.4 Set Auto Gainmatic Mode

PUT /isp/agc-mode	Permission: OPERATOR Configuration Set: No
Request and Response	**Description**
"continuous"	(This item will not be saved after power off)
	None
Status Code	
200	
400	

4.5.5 Get Raw Image Snapshot

GET /isp/snapshot	Permission: VIEWER Configuration Set: No
Request and Response	**Description**
	None
{ "path" : "/file/cache/8kn4VBYmAw0XKUvl"	Snapshot is in P7 format, see Appendix

}	
Status Code	
200	

4.5.6 Get Temperature Image Snapshot

GET /isp/t-snapshot	**Permission: VIEWER** **Configuration Set: No**
Request and Response	**Description**
	None

{
"path" :
"/file/cache/8kn4VBYmAw0XKUvI"
}
	Snapshot is in P7 format, see Appendix.
	The temperature data is represented by a 16-bit signed integer, of which the lowest 3 bits are fixed-point decimal digits.For example, 0x007A indicates 15.25 ℃ temperature

Status Code	
200	

4.5.7 Get Point Temperature on Image

GET /isp/t **GET /isp/t?x=*xpos*&y=*ypos*** ・ x, y Image coordinate ・ The default is the center point when there is no parameter	**Permission: VIEWER** **Configuration Set: No**
Request and Response	**Description**
	None

{
"r":7812,
"t":23.234
}
	Apply global temperature compensation parameters

Status Code	
200	
404	Object does not exist

4.6 Temperature Measurement ISP / INSTRUMENT

The temperature measurement module adds temperature measurement objects of different shapes to the image according to the user settings, calculates the image temperature, and generates temperature measurement events according to the trigger rules.

The temperature measurement tool supports adding temperature measurement points, temperature measurement regions, and temperature measurement polylines to the image. It can be used to get the temperature value of the temperature measuring point, the highest and lowest temperature in the temperature measuring region, and the temperature measuring polyline.

- The temperature measurement region can be defined in two ways: polygon and region mask.
- The number of polygon and polyline vertices cannot exceed 50.
- The name of the temperature measurement object cannot exceed 40 characters.

4.6.1 Get Global Temperature Measurement Parameters

Global temperature measurement parameters are the basic parameters of device temperature measurement.The temperature measurement object inherits all global temperature measurement parameters by default, and can also set its own local temperature measurement parameters.

GET /isp/instrument/jconfig	**Permission: VIEWER** **Configuration Set: Yes**
Request and Response	**Description**
	None

{
"ambient-t" : 20,
"distance" : 1,
"emissivity" : 0.97,
"lens-t" : 20,
"lens-transmissivity" : 1,
"offset" : 0,
"reflect-t" : 20,
"rh" : 0.5
}
	・ambient-t Ambient temperature (greater than -273.15)
	・distance Distance of the measured object (greater than 0)
	・emissivity Emissivity of the measured object (0, 1]
	・lens-t Lens Temperature (greater than -273.15)
	・lens-transmissivity Lens transmissivity (0, 1]
	・offset Offset (system error correction)
	・reflect-t Reflection temperature (greater than -273.15)

Status Code	
200	

4.6.2 Set Global Temperature Measurement Parameters

PUT /isp/instrument/jconfig	**Permission: OPERATOR** **Configuration Set: Yes**
Request and Response	**Description**

{
"ambient-t" : 20,
"distance" : 1,
"emissivity" : 0.97,
"lens-t" : 20,
"lens-transmissivity" : 1,
"offset" : 0,
"reflect-t" : 20,
"rh" : 0.5
}	

	None
Status Code	
200	

4.6.3 Get Temperature Measurement Object List

GET /isp/instrument/objects/points **GET /isp/instrument/objects/regions** **GET /isp/instrument/objects/lines**	**Permission: VIEWER** **Configuration Set: Yes**
Request and Response	**Description**
	None

["p1", "p2"]	・
Status Code	
200	

4.6.4 Get Temperature Measurement Object Parameters

GET /isp/instrument/objects/points/*obj_name* GET /isp/instrument/objects/regions/*obj_name* GET /isp/instrument/objects/lines/*obj_name* GET /isp/instrument/objects/global ・ obj_name Object name	**Permission: VIEWER** **Configuration Set: Yes**

Request and Response	**Description**
	None

{
"emissivity" : 0.97,
"pos" :
{
"x" : 55,
"y" : 50
},
"distance" : 1000,
"polygon" :
[
{
"x" : 30,
"y" : 50
},
{
"x" : 35,
"y" : 10
},
{
"x" : 60,
"y" : 60
},
{
"x" : 5,
"y" : 76
}
],
"label" : "My Region1"
}
	Temperature measurement parameters that can be modified separately for each object:
	・ emissivity
	・ reflect-t
	・ distance
	・ offset
	
	・ osd-visible Display on the OSD
	・ api-visible Label stream transmission
	・ mb-slot Allocated modbus address space
	・ label Label content used in display
	・ alarm-t Temperature alarm threshold
	
	regions Regions
	・ layer Layer
	・ polygon Polygon vertex list
	・ mask Mask image cache token
	
	lines Lines
	・ polyline Polyline vertex list

Status Code	
200	
404	Object does not exist

4.6.5 Create and Update Temperature Measurement Objects

PUT /isp/instrument/objects/points/*obj_name* **PUT /isp/instrument/objects/regions/*obj_name*** **PUT /isp/instrument/objects/lines/*obj_name*** **PUT /isp/instrument/objects/global** ・ obj_name Object name	**Permission: OPERATOR** **Configuration Set: Yes**
Request and Response	**Description**

{
"emissivity" : 0.97,
"pos" :
{
"x" : 55,
"y" : 50
},
"mb-slot" : 1
}	

	None
Status Code	
200	
201	New object

4.6.6 Delete Temperature Measurement Objects of Same Type

DELETE /isp/instrument/objects/points **DELETE /isp/instrument/objects/regions** **DELETE /isp/instrument/objects/lines**	**Permission: OPERATOR** **Configuration Set: Yes**
Request and Response	**Description**
	None
	None
Status Code	
200	
404	Object does not exist

4.6.7 Delete Temperature Measurement Object

DELETE /isp/instrument/objects/points/*obj_name* **DELETE /isp/instrument/objects/regions/*obj_name*** **DELETE /isp/instrument/objects/lines/*obj_name*** ・ obj_name Object name	**Permission: OPERATOR** **Configuration Set: Yes**
Request and Response	**Description**
	None
	None
Status Code	
200	
404	Object does not exist

4.6.8 Get Temperature Measurement Object Data

GET /isp/instrument/objects/points/*obj_name*?value **GET /isp/instrument/objects/regions/*obj_name*?value** **GET /isp/instrument/objects/lines/*obj_name*?value** ・ obj_name Object name	**Permission: VIEWER** **Configuration Set: No**
Request and Response	**Description**
	None

{
"r":7812,
"t":23.234
}
	・ r Sampling value
	・ t Temperature in Celsius

Status Code	
200	
404	Object does not exist

4.6.9 Get Maximum Number of Temperature Measurement Objects

GET /isp/instrument/max-point-num **GET /isp/instrument/max-region-num** **GET /isp/instrument/max-line-num**	**Permission: VIEWER** **Configuration Set: No**
Request and Response	**Description**
	None
10	
Status Code	
200	

4.6.10 Get Global Maximum and Minimum Temperature

GET /isp/instrument/objects/global?value	**Permission: VIEWER** **Configuration Set: No**
Request and Response	**Description**

{
"max" :
{
"r" : 7848,
"t" : 0.6256697,
"x" : 72,
"y" : 18
},
"min" :
{
"r" : 8125,
"t" : 25.35555,
"x" : 72,
"y" : 18
}	

}	
	None
Status Code	
200	

4.6.11 Get Sampling Rate

GET /isp/instrument/sample-rate	**Permission: VIEWER** **Configuration Set: Yes**
Request and Response	**Description**
2	
Status Code	
200	

4.6.12 Set Sampling Rate

PUT /isp/instrument/sample-rate	**Permission: OPERATOR** **Configuration Set: Yes**
Request and Response	**Description**
2	
Status Code	
200	

4.7 Palette ISP / T-RAY

4.7.1 Get Device Predefined Palette List

GET /isp/t-ray/presets	**Permission: OPERATOR** **Configuration Set: No**
Request and Response	**Description**
	None
["grey", "iron", "rain", "rainbow", "greyred", "glowbow", "yellow", "midgrey", "midgreen"]	
Status Code	
200	

4.7.2 Set Current Palette

PUT /isp/t-ray/plt	Permission: OPERATOR Configuration Set: Yes
Request and Response	**Description**

{
"inverse" : false,
"name" : "grey"
}	

Status Code	
200	
404	

4.7.3 Get Current Palette

GET /osd/t-ray/plt	Permission: VIEWER Configuration Set: Yes
Request and Response	**Description**
	None

{
"inverse" : false,
"name" : "grey"
}
	・inverse Whether to invert the palette
	・name Palette name

Status Code	
200	
404	

4.7.4 Set Custom Palette

Consult the supplier for details

4.8 Auto Focus ISP / AF

4.8.1 Perform Auto Focus

The Focus window is in the screen center, please place the featured scene in the center of the camera

POST /isp/af	Permission: VIEWER Configuration Set: No
Request and Response	**Description**

{
"x" : 10,
"y" : 10
}
	Focus center position
	When the request is empty, it defaults to the screen center

	None
Status Code	
200	

4.8.2 Get Auto Focus Result

The Focus window is in the screen center, please place the featured scene in the center of the camera

GET /isp/af/result	**Permission: VIEWER** **Configuration Set: No**
Request and Response	**Description**
	None
"pending"	Result: ・ pending Being focusing ・ good Good ・ nofeat No focusing feature found ・ timeout Focus timeout ・ unknown Unknown error
Status Code	
200	

4.9 On Screen Display OSD

4.9.1 Get OSD List

GET /osd/layout	**Permission: VIEWER** **Configuration Set: Yes**
Request and Response	**Description**
	None
[“time”, “title”]	
Status Code	
200	

4.9.2 Get OSD Object Parameters

GET /osd/layout/*obj_name* ・ obj_name Object name	**Permission: VIEWER** **Configuration Set: Yes**
Request and Response	**Description**
	None

{
"show" : true,
"pos" :{
"x": 5,
"y": 2
},
"align" :"topright"
}	

Status Code	
200	
404	

4.9.3 Set Time Display

PUT /osd/layout/time	Permission: VIEWER Configuration Set: Yes
Request and Response	**Description**

{
"show" : true,
"pos" :{
"x": 5,
"y": 2
},
"align" :"topright"
}	

Status Code	
200	
404	

4.9.4 Get Current Display Time Zone

GET /osd/tz	Permission: VIEWER Configuration Set: Yes
Request and Response	**Description**
"CST-8"	
Status Code	
200	

1.1. Set Current Display Time Zone

PUT /osd/tz	Permission: OPERATOR Configuration Set: Yes
Request and Response	**Description**
"CST-8"	
Status Code	
200	

4.9.5 Set Title Display

PUT /osd/layout/title	Permission: VIEWER Configuration Set: Yes
Request and Response	**Description**

{
"show" : true,
"pos" :{
"x": 5,
"y": 2	

}, "align" :"topright" }	
Status Code	
200	
404	

4.9.6 Take a Snapshot

GET /osd/snapshot	**Permission: VIEWER** **Configuration Set: No**
Request and Response	**Description**
	None

{
"path" :
"/file/cache/8kn4VBYmAw0XKUvl"
}
	Snapshot is in P7 format, see Appendix

Status Code	
200	

4.10 Data stream STREAM

The data stream service provided by the data stream module uses port 10081, and each data stream corresponds to the data stream attribute in the data stream module.

4.10.1 Get Primary Stream Properties

GET /stream/video/pri	**Permission: VIEWER** **Configuration Set: Yes**
Request and Response	**Description**
	None

{
"bitrate" : 400000,
"height" : 256,
"max-bitrate" : 400000,
"max-packet-size" : 65536,
"pixel-format" : "yuv420",
"width" : 256,
"max-connection-num" : 0,
"connections": 0
}
	・ max-bitrate Max bitrate
	・ bitrate Bitrate of the current video stream
	・ connections Current numbers of connections

Status Code	
200	

4.10.2 Get Sub Stream Properties

GET /stream/video/sub	**Permission: VIEWER** **Configuration Set: Yes**

Request and Response	Description
	None

{
"bitrate" : 80000,
"height" : 128,
"max-bitrate" : 80000,
"max-packet-size" : 16384,
"pixel-format" : "yuv420",
"width" : 128,
"max-connection-num" : 0,
"connections": 0
}	

Status Code	
200	

4.10.3 Get Raw Stream Properties

GET /stream/video/raw	Permission: VIEWER
	Configuration Set: Yes
Request and Response	**Description**
	None

{
"fps" : 16.129032,
"height" : 80,
"max-packet-size" : 12800,
"width" : 80
"max-fps" : 0
"max-connection-num" : 0
"pixel-format" : "grey16le",
"connections": 0
}	

Status Code	
200	

4.10.4 Set Raw Stream Properties

PUT /stream/video/raw	Permission: VIEWER
	Configuration Set: Yes
Request and Response	**Description**

{
"fps" : 5,
}	

	None
Status Code	
200	

4.10.5 Get Event Stream Properties

GET /stream/tag/events	Permission: VIEWER
	Configuration Set: Yes

Request and Response	Description
	None

{
"heartbeat" : 5,
"max-packet-size" : 8192
}	

Status Code	
200	

4.10.6 Get Value Stream Properties

GET /stream/tag/values	**Permission: VIEWER** **Configuration Set: Yes**
Request and Response	**Description**
	None

{
"max-packet-size" : 8192
}	

Status Code	
200	

4.11 Peripheral Control PERI

4.11.1 Perform Manual Focus

POST /peri/focus?op=*op*&step=*step* ・ op near / far ・ step Step 10-1000	**Permission: OPERATOR** **Configuration Set: No**
Request and Response	**Description**
	One step is one thousandth of the full travel of the motor So 1000 means one trip from the nearest to the farthest
	None
Status Code	
200	

4.11.2 Get Serial Port Parameters

GET /peri/serial-port	**Permission: MANAGER** **Configuration Set: No**
Request and Response	**Description**
	None
{ "baudrate" : 9600, "command-gap" : 0, "csize" : 8, "modbus-address" : 1,	・ baudrate Serial port baud rate ・ csize Word length ・ parity Parity bit ・ stopbit Stop bit ・ modbus-address Modbus address

"modbus-slave" : false, "parity" : "none", "stopbit" : "one" }	• modbus-slave Whether modbus slave is enabled • command-gap Minimum waiting time between consecutive commands
Status Code	
200	

4.11.3 Get Supported Serial Port Baud Rate List

GET /peri/serial-port/baudrates	**Permission: MANAGER** **Configuration Set: No**
Request and Response	**Description**
	None
[150, 200, 300, 600, 1200, 1800, 2400, 4800, 9600, 19200, 38400, 57600, 115200]	
Status Code	
200	

4.11.4 Get Supported Stop Bit List

GET /peri/serial-port/stopbits	**Permission: MANAGER** **Configuration Set:**
Request and Response	**Description**
	None
["one", "onepointfive", "two"]	
Status Code	
200	

4.11.5 Get Supported Parity Bit List

GET /peri/serial-port/parities	**Permission: MANAGER** **Configuration Set:**
Request and Response	**Description**

	None
["none", "odd", "even"]	
Status Code	
200	

4.11.6 Set Serial Port Parameters

PUT /peri/serial-port	**Permission:** **Configuration Set:**
Request and Response	**Description**

{
"baudrate" : 9600,
"command-gap" : 0,
"csize" : 8,
"modbus-address" : 1,
"modbus-slave" : false,
"parity" : "none",
"stopbit" : "one"
}	

	None
Status Code	
200	

4.11.7 Control PTZ Movement

POST /peri/ptz/move?id=*ptz_id*&vx=*vx*& vy=*vy* ・ ptz_id Serial port address of PTZ (Pelco-d) ・ vx Horizontal velocity ・ vy Vertical velocity	**Permission: OPERATOR** **Configuration Set: No**
Request and Response	**Description**
	None
	None
Status Code	
200	

4.11.8 Set PTZ Preset Position

POST /peri/ptz/set?id=*ptz_id*&data=*val* ・ ptz_id Serial port address of PTZ (Pelco-d) ・ val Preset bit number (0~255)	**Permission: OPERATOR** **Configuration Set: No**
Request and Response	**Description**
	None
	None
Status Code	
200	

4.11.9 Go to PTZ Preset Position

POST /peri/ptz/goto?id=*ptz_id*&data=*val* ・ ptz_id Serial port address of PTZ (Pelco-d) ・ val Preset bit number (0~255)	**Permission: OPERATOR** **Configuration Set: No**
Request and Response	**Description**
	None
	None
Status Code	
200	

4.11.10 Clear PTZ Preset Position

POST /peri/ptz/clear?id=*ptz_id*&data=*val* ・ ptz_id Serial port address of PTZ (Pelco-d) ・ val Preset bit number (0~255)	**Permission: OPERATOR** **Configuration Set: No**
Request and Response	**Description**
	None
	None
Status Code	
200	

4.11.11 Stop PTZ

POST /peri/ptz/stop?id=*ptz_id* ・ ptz_id Serial port address of PTZ (Pelco-d)	**Permission: OPERATOR** **Configuration Set: No**
Request and Response	**Description**
	None
	None
Status Code	
200	

4.11.12 Set MODBUS Parameters

PUT /peri/modbus	**Permission: MANAGER** **Configuration Set: No**
Request and Response	**Description**

{
"command-gap" : 0,
"enabled" : true,
"modbus-address" : 1
}	

	None
Status Code	
200	

4.11.13 Get MODBUS Parameters

GET /peri/modbus	Permission: MANAGER Configuration Set: No
Request and Response	**Description**
	None

{
"command-gap" : 0,
"enabled" : true,
"modbus-address" : 1
}	

Status Code	
200	

4.12 General File Cache Service FILE

4.12.1 Download File

GET /file/cache/*file_name* ・ file_name File name	Permission: VIEWER Configuration Set: No
Request and Response	**Description**
	None
(binary)	(File content)
Status Code	
200	
404	

4.12.2 Upload File

After uploading the file, the user will get a file ID, which can be used as a link in other interfaces that support the path field.The size of the uploaded file cannot be greater than 20MB.

POST /file/cache	Permission: MANAGER Configuration Set: No
Request and Response	**Description**
	None

{
"path" : "8kn4VBYmAw0XKUvI"
}
	File relative path

Status Code	
200	
400	

4.12.3 Get Log File

GET /file/log/*log_name* ・ log_name Log file name	Permission: MANAGER Configuration Set: No

Request and Response	Description
	None
(binary)	File content Logs include: messages daemon.log auth.log …
Status Code	
200	
404	

4.13 Others

4.13.1 Escape REST Command

POST /_rest?put=*rest_path* **POST /_rest?delete=*rest_path*** ・ rest_path Effective command path (URL encoding)	**Permission: ALL** **Configuration Set: No**
Request and Response	**Description**
	Same as the original command
	Same as the original command
Status Code	
200	

5 Device Panel

5.1 Reset Button RST

The reset key is used to restore the device parameters to the factory state.

After powering on, the user ensures that the device has been initialized or waits for at least 30 seconds, and then presses the reset button for at least 10 seconds. The device performs the reset operation and restarts. The original IP address of the device is restored to the factory value.

The user list of the device will not be removed. If you forget the user or password, please contact the local supplier.

6 MODBUS Interface Support

MODBUS protocol is a universal protocol applied to electronic controllers. The controllers communicate with each other through physical networks (such as serial networks and Ethernet), which is a widely used industry standard. On a MODBUS network, each device is assigned a communication address. The protocol defines how the controller accesses other devices at different addresses and performs various operations. (Refer to http://modbus.org/docs/Modbus_Application_Protocol_V1_1b3.pdf)

This protocol supports traditional RS-232, RS-422, RS-485, and Ethernet devices. Many industrial devices, including PLC, DCS, and intelligent instruments, are using Modbus protocol as their communication standard.

In addition to the HTTP interfaces, in order to facilitate communication with other industrial devices, the fixed thermal cameras provide Modbus interfaces using RS-485 interfaces for data collection and control.

MODBUS protocol based on serial port generally has two modes: RTU and ASCII. This device only supports RTU mode with smaller data size. Serial port parameters can be modified through /peri/serial-port (4.11.6).

6.1 Serial Port RTU Protocol

1.1. Frame Format

Start	Address	Function	Data	CRC	End
3.5 characters*	8 bits	8 bits	N x 8bits	16 bits	3.5 characters*

* Serial communication word length

6.1.1 CRC Check

See Appendix

6.1.2 Read Holding Registers

Function Code	0x03
Field	**Example Value (hex) Read 5 registers with addresses from 0x102C to 0x1030**
Request Frame	
Address	01
Function	03
Starting Address (high)	10
Starting Address (low)	2C
Quantity of Registers (high)	00
Quantity of Registers (low)	04
CRC (low)	40
CRC (high)	C0
Response Frame	
Address	01
Function	03
Byte Count	0A
Register 0x102C (high)	00
Register 0x102C (low)	02
Register 0x102D (high)	00
…	…
Register 0x1030 (low)	1A
CRC (low)	-
CRC (high)	-

6.1.3 Exception Response

When an exception returns, the highest bit of the function field in the response frame is 1, which is equivalent to the request function plus 0x80, and the data field returns the corresponding exception code

Function	0x03

Code	
Field	**Example Value (hex)** **Read 5 registers with addresses from 0x102C to 0x1030**
Request Frame	
Address	01
Function	03
Starting Address (high)	10
Starting Address (low)	2C
Quantity of Registers (high)	00
Quantity of Registers (low)	04
CRC (low)	40
CRC (high)	C0
Response Frame	
Address	01
Function	83
Exception Code	0A
CRC (low)	-
CRC (high)	-
Field	**Example Value (hex)**
Address	01
Function	83

6.2 Address Space Definition

The device address is defined in a continuous single space . The unit is a 16-bit word, starting from 0. All integers are placed from the big end

Address Space (hex)	Length (word)	Content
System Information Area	**512**	
0000	1	Fixed value 0x4952
0001	1	Protocol version (==1)
0002 ~ 0003	2	Firmware version (0x03010011 indicates 3.1.0.17)
0004	1	Number of supported measurement data blocks
0008 ~ 0029	40	Model
0030 ~ 0058	40	Serial number
Measurement Data Area	**512**	
0200 ~ 0207	8	Data block 1
0208 ~ 0210	8	Data block 2
…		
0200 + 8 x n ~ 0200 + 8 x (n+1) - 1	8	数据块n

Control Area (TBD)		

6.3 Measurement Data Block

The temperature measurement data is mapped with the data block through the mb-slot field (0). When the measured object is associated with the data block, the measured value can be read out through the corresponding data block. The size of each data block is 8 words. A 32-bit signed integer (big endian) composed of two words is a Q15.16 fixed-point integer. The first word is a 16 digit integer and the last word is a 16 digit fixed-point decimal

See https://en.wikipedia.org/wiki/Q_(number_format)

Base address of data block n: 0x200 + n * 8:

Address Range (hex)	Length (word)	Content
0000~0001	2	Temperature, maximum temperature
0002~0003	2	Minimum temperature

7 Onvif Interface and RTSP Video Stream

Some device models support Onvif 2.0 interface for accessing mainstream security monitoring systems. RTSP video streams that are part of the Onvif protocol can also be used separately. The specific interfaces supported by the device are listed below.

7.1 Basic Functions (Core Spec. Ver 2.4.2)

Function Name	Description	Notes
Device Discovery	Device discovery	Online Hello is supported, offline Byte is not supported
GetCapabilities	Get device capabilities	
SystemReboot	Reboot device	

7.2 Event Handling (Core Spec.Ver 2.4.2)

Function Name	Description	Notes
GetEventProperties	Get event properties	
Subscribe	Event subscription	
Renew	Event subscription update	
Unsubscribe	Unsubscribe	
Notify	Message notification	Sent by device to subscribers

7.3 Media Service (Media Service Ver 2.4.2)

Function Name	Description	Notes
GetVideoSources	Get video source list	
GetProfile	Get video configuration	
GetProfiles	Get video configuration list	
GetVideoSourceConfigurations	Get video source configurations	
GetVideoEncoderConfiguration	Get video encoder configuration	
GetVideoEncoderConfigurations	Get video encoder configuration list	
GetVideoEncoderConfigurationOptions	Get video encoder configuration option list	
GetStreamUri	Get video stream URI	
GetOSDOptions	Get OSD options	
GetOSD	Get OSD	OSD header string
SetOSD	Set OSD	
CreateOSD	Create OSD	
DeleteOSD	Delete OSD	

7.3.1 RTSP Video Stream

If the device supports the Onvif protocol, it will also support video output through RTSP.The device supports RTSP / RTP / TCP mode and uses standard port 554.The access path refers to the HTTP video stream 4.10, specifically:

Primary stream:

```
RTSP://xxx.xxx.xxx.xxx/video/pri
```

Sub stream:

```
RTSP://xxx.xxx.xxx.xxx/video/sub
```

7.4 PTZ Control (PTZ spec.Ver 2.4.2)

Function Name	Description	Notes
GetConfigurations	Get all PTZ configuration information	
GetConfiguration	Get PTZ	
ContinuousMove	Move PTZ and focus	Currently, only focusing is supported
Stop	Stop PTZ and focus	
GetPresets	Get preset list	256 presets
SetPreset	Set preset	
GotoPreset	Go to preset	
RemovePreset	Remove preset	

8 Other Reference Information

8.1 Data Type Definition

Type	Length (byte)	Notes
UUID	16	
MD5	16	
String	80	utf-8
Integer	4	
Unsigned Integer	4	
Floating Point	4	32-bit IEEE 754 floating point number
Enumeration	4	Use Integer (unsigned)
Date	16	

8.2 Common HTTP Response Status Code

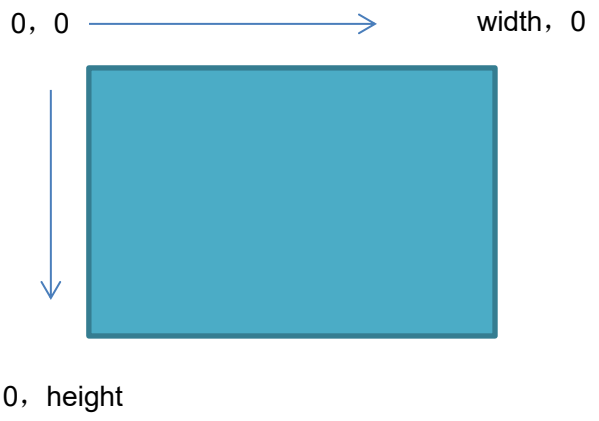
Status Code	Type	Description
200	Success	OK
201	Success	Created
301	Redirect	Moved Permanently
302	Redirect	Found
400	Client Error	Bad Request
401	Client Error	Unauthorized
403	Client Error	Forbidden
404	Client Error	Not Found
500	Server Error	Internal Server Error
501	Server Error	Not Implemented
502	Server Error	Bad Gateway

8.3 Snapshot Image Format

Image header (16 bytes):

0 1 2 3 4 5 6 7	0 1 2 3 4 5 6 7	0 1 2 3 4 5 6 7			0 1 2 3 4 5 6 7
‘P’	‘7’	Image width width (16bit)			
Image height (16bit)		Bit depth depth	C	E	Image type type
Line size linesize (bytes)					
Reserved					

Image data is stored in row-major order, and each row is defined by linesize



The number of bytes in each row of the image, and the calculation method of the offset in each row of the image:

$$\text{offset of} i_{th} \text{ line} = \text{ptr} + linesize * i$$

- Bit depth: number of bytes per pixel (not bits)
- Image type: 0 represents the raw image, an image plane; 1 means yuv420p mode
- C: Image block identification
- E: Extension header

Extension head (64 bytes)

0 1 2 3 4 5 6 7	0 1 2 3 4 5 6 7	0 1 2 3 4 5 6 7	0 1 2 3 4 5 6 7
E	Type=1	Reserved	
Image data offset			
Fixed chunk list size			
Reserved (52 bytes)			

8.4 MODBUS Exception Code List

Exception Code	Name	Description
01	Illegal command	The device cannot recognize the function code in the request frame
02	Illegal address	The device cannot access the requested address
03	Data error	Wrong data value in request frame
04	Internal error	An error occurred while processing the function inside the device
05	Handshake	
06	Device busy	

8.5 MODBUS CRC Generation Code

From Modicon Modbus Protocol Reference Guide (PI – MBUS – 300)

```
/*
   MODBUS CRC Generation fucntion
Input Paramers:
   puchMsg: message to calculate CRC upon
   usDataLen: quantity of bytes in message
*/
unsigned short CRC16(unsigned char *puchMsg, unsigned short usDataLen)
{
   unsigned char uchCRCHi = 0xFF ; /* high byte of CRC initialized */
   unsigned char uchCRCLo = 0xFF ; /* low byte of CRC initialized */
   unsigned uIndex ; /* will index into CRC lookup table */
   while (usDataLen--) /* pass through message buffer */
   {
      uIndex = uchCRCHi ^ *puchMsgg++ ; /* calculate the CRC */
      uchCRCHi = uchCRCLo ^ auchCRCHi[uIndex} ;
      uchCRCLo = auchCRCLo[uIndex] ;
   }
   return (uchCRCHi << 8 | uchCRCLo) ;
}
```

```
/* Table of CRC values for high-order byte */
static unsigned char auchCRCHi[] = {
0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0, 0x80, 0x41, 0x01, 0xC0, 0x80, 0x41, 0x00,
0xC1, 0x81,
0x40, 0x01, 0xC0, 0x80, 0x41, 0x00, 0xC1, 0x81, 0x40, 0x00, 0xC1, 0x81, 0x40,
0x01, 0xC0,
0x80, 0x41, 0x01, 0xC0, 0x80, 0x41, 0x00, 0xC1, 0x81, 0x40, 0x00, 0xC1, 0x81,
0x40, 0x01,
0xC0, 0x80, 0x41, 0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0, 0x80, 0x41, 0x01, 0xC0,
0x80, 0x41,
0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0, 0x80, 0x41, 0x00, 0xC1, 0x81, 0x40, 0x00,
0xC1, 0x81,
0x40, 0x01, 0xC0, 0x80, 0x41, 0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0, 0x80, 0x41,
0x01, 0xC0,
0x80, 0x41, 0x00, 0xC1, 0x81, 0x40, 0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0, 0x80,
0x41, 0x01,
0xC0, 0x80, 0x41, 0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0, 0x80, 0x41, 0x00, 0xC1,
0x81, 0x40,
0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0, 0x80, 0x41, 0x01, 0xC0, 0x80, 0x41, 0x00,
0xC1, 0x81,
0x40, 0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0, 0x80, 0x41, 0x00, 0xC1, 0x81, 0x40,
0x01, 0xC0,
0x80, 0x41, 0x01, 0xC0, 0x80, 0x41, 0x00, 0xC1, 0x81, 0x40, 0x00, 0xC1, 0x81,
0x40, 0x01,
0xC0, 0x80, 0x41, 0x01, 0xC0, 0x80, 0x41, 0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0,
0x80, 0x41,
0x00, 0xC1, 0x81, 0x40, 0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0, 0x80, 0x41, 0x00,
0xC1, 0x81,
0x40, 0x01, 0xC0, 0x80, 0x41, 0x01, 0xC0, 0x80, 0x41, 0x00, 0xC1, 0x81, 0x40,
0x01, 0xC0,
```

```
0x80, 0x41, 0x00, 0xC1, 0x81, 0x40, 0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0, 0x80,
0x41, 0x01,
0xC0, 0x80, 0x41, 0x00, 0xC1, 0x81, 0x40, 0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0,
0x80, 0x41,
0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0, 0x80, 0x41, 0x01, 0xC0, 0x80, 0x41, 0x00,
0xC1, 0x81,
0x40
} ;

/* Table of CRC values for low-order byte */
static char auchCRCLo[] = {
0x00, 0xC0, 0xC1, 0x01, 0xC3, 0x03, 0x02, 0xC2, 0xC6, 0x06, 0x07, 0xC7, 0x05,
0xC5, 0xC4,
0x04, 0xCC, 0x0C, 0x0D, 0xCD, 0x0F, 0xCF, 0xCE, 0x0E, 0x0A, 0xCA, 0xCB, 0x0B,
0xC9, 0x09,
0x08, 0xC8, 0xD8, 0x18, 0x19, 0xD9, 0x1B, 0xDB, 0xDA, 0x1A, 0x1E, 0xDE, 0xDF,
0x1F, 0xDD,
0x1D, 0x1C, 0xDC, 0x14, 0xD4, 0xD5, 0x15, 0xD7, 0x17, 0x16, 0xD6, 0xD2, 0x12,
0x13, 0xD3,
0x11, 0xD1, 0xD0, 0x10, 0xF0, 0x30, 0x31, 0xF1, 0x33, 0xF3, 0xF2, 0x32, 0x36,
0xF6, 0xF7,
0x37, 0xF5, 0x35, 0x34, 0xF4, 0x3C, 0xFC, 0xFD, 0x3D, 0xFF, 0x3F, 0x3E, 0xFE,
0xFA, 0x3A,
0x3B, 0xFB, 0x39, 0xF9, 0xF8, 0x38, 0x28, 0xE8, 0xE9, 0x29, 0xEB, 0x2B, 0x2A,
0xEA, 0xEE,
0x2E, 0x2F, 0xEF, 0x2D, 0xED, 0xEC, 0x2C, 0xE4, 0x24, 0x25, 0xE5, 0x27, 0xE7,
0xE6, 0x26,
0x22, 0xE2, 0xE3, 0x23, 0xE1, 0x21, 0x20, 0xE0, 0xA0, 0x60, 0x61, 0xA1, 0x63,
0xA3, 0xA2,
0x62, 0x66, 0xA6, 0xA7, 0x67, 0xA5, 0x65, 0x64, 0xA4, 0x6C, 0xAC, 0xAD, 0x6D,
0xAF, 0x6F,
0x6E, 0xAE, 0xAA, 0x6A, 0x6B, 0xAB, 0x69, 0xA9, 0xA8, 0x68, 0x78, 0xB8, 0xB9,
0x79, 0xBB,
0x7B, 0x7A, 0xBA, 0xBE, 0x7E, 0x7F, 0xBF, 0x7D, 0xBD, 0xBC, 0x7C, 0xB4, 0x74,
0x75, 0xB5,
0x77, 0xB7, 0xB6, 0x76, 0x72, 0xB2, 0xB3, 0x73, 0xB1, 0x71, 0x70, 0xB0, 0x50,
0x90, 0x91,
0x51, 0x93, 0x53, 0x52, 0x92, 0x96, 0x56, 0x57, 0x97, 0x55, 0x95, 0x94, 0x54,
0x9C, 0x5C,
0x5D, 0x9D, 0x5F, 0x9F, 0x9E, 0x5E, 0x5A, 0x9A, 0x9B, 0x5B, 0x99, 0x59, 0x58,
0x98, 0x88,
0x48, 0x49, 0x89, 0x4B, 0x8B, 0x8A, 0x4A, 0x4E, 0x8E, 0x8F, 0x4F, 0x8D, 0x4D,
0x4C, 0x8C,
0x44, 0x84, 0x85, 0x45, 0x87, 0x47, 0x46, 0x86, 0x82, 0x42, 0x43, 0x83, 0x41,
0x81, 0x80,
0x40
} ;
```

9 Appendix

9.1 Product Installation Guide

9.1.1 Important Notice before Installation

9.1.1.1 Disclaimers

This manual may contain technical inaccuracies, areas inconsistent with product functions and operations, or printing errors. Our company will update the content of this manual according to the enhancement or change of product functions, and will regularly improve and update the software and hardware products described in this manual. The updated contents will be added to the new version of this manual without notice.

The contents in this manual are only for reference and guidance and do not guarantee that they are completely consistent with the real product. Please refer to the real product.

To the maximum extent permitted by law, the products described in this manual (including hardware, software, firmware, etc.) are provided "as is". The company does not provide any explicit or implied warranty for possible defects, errors, or failures, including but not limited to the warranties of merchantability, quality satisfaction, suitability for specific purposes, and non-infringement of the rights of third parties; Neither will it compensate for any special, incidental, accidental or indirect damages caused by the use of the User Manual or the use of the Company's products, including but not limited to loss of business profits, loss of data or documents. To the maximum extent permitted by law, the Company's liability for compensation shall not exceed the factory value of this product.

If the product is interrupted or the service is terminated due to the following reasons, our company will not assume any responsibility for your or others' personal damage or property loss: it is not installed or used correctly in strict accordance with the requirements; to safeguard national or public interests; force majeure; your own or third-party reasons (including but not limited to the use of third-party products, software or components).

If you connect the product to the Internet, you may face risks including but not limited to network attacks, hacker attacks, virus infections, etc. Our company will not be responsible for the abnormal work of the product, information leakage and other problems caused thereby but will provide you with technical support in a timely manner.

The correct installation and use of this product can sense illegal intrusions and fire alarm events in specific areas, but can not avoid the occurrence of accidents or the resulting personal damage or property loss. In daily life, you should be vigilant and strengthen your awareness of safety precautions.

The pre-installed software of this product has legal rights or has obtained legal authorization. We do not recommend you to install unauthorized software. In case of incompatibility, inability to use, infringement, personal damage, property loss, third-party compensation, or punishment caused by your own installation of software, our company will not assume any responsibility.

Please strictly follow the applicable laws when using this product. You agree that this product is only for civil use, and shall not be used for applications that infringe the rights of third parties, medical / safety equipment, or other products whose failures may lead to life danger or personal injury, as well as weapons of mass destruction, biological and chemical weapons, nuclear explosions or any unsafe nuclear energy use or dangerous or anti-humanism purposes. Any loss or liability arising from the above purposes will be borne by you.

If the above contents conflict with applicable laws, the legal provisions shall prevail.

9.1.1.2 Safety Information

The purpose of the safety information is to ensure that users use this product correctly to avoid danger or property loss. Before using this product, please read this instruction manual carefully and keep it properly for future reference.

As shown below, preventive measures are divided into two categories: "Warning" and "Caution":

Warning: Ignoring warning information may cause death or serious injury

Caution: Ignoring caution information may cause injury or property loss

	
Warning matters remind users to prevent potential death or serious injury	**Caution** matters remind users to prevent potential injury or property loss

 **Warning:**

- Please use the power supply that meets the SELV (Safety Extra Low Voltage) requirements, and the rated voltage of the limited power source is 12/24V according to IEC60950-1.
- If the equipment does not work properly, please contact the dealer or our company, and do not disassemble or modify the equipment in any way (the problem caused by unauthorized modification or maintenance shall be borne by yourself).
- To reduce the risk of fire or electric shock, do not expose indoor products to rain or moisture.
- The installation shall be carried out or guided by professional service personnel and comply with local regulations.
- Easily powered-off equipment should be incorporated into the building installation wiring.

 **Caution:**

- Before running the thermal camera, check whether the power supply is correctly connected.
- Do not drop this product to the ground or be strongly knocked.
- Do not directly touch the sensor or lens. If it is necessary to clean up, please slightly wet the clean cloth with alcohol and gently wipe away the dust. When the thermal imager is not in use, please place it in a dust-free environment and pay attention to dust prevention.
- Avoid focusing on the sun or objects with super high temperatures or observing for a long time, otherwise, the sensor life will be reduced or the sensor will be damaged or temporary black spots will appear (minor cases can be recovered after NUC calibration, serious cases will lead to permanent and irrecoverable damage of the detector).
- Avoid placing the product in damp, dusty, extremely hot, extremely cold, strong electromagnetic radiation, and other places.
- Please ensure that the installation position is kept at a sufficient distance from the surrounding electromagnetic-sensitive equipment to avoid possible electromagnetic interference
- Avoid heat accumulation and maintain smooth ventilation around the thermal imager.
- Avoid water or any liquid flowing into the thermal imager during use.

- When transporting the thermal imager, it is strongly recommended to use the factory packaging for protection.

9.1.2 Appearance

Caution: The specific appearance is subject to the actual product.

9.1.3 Interface Definition

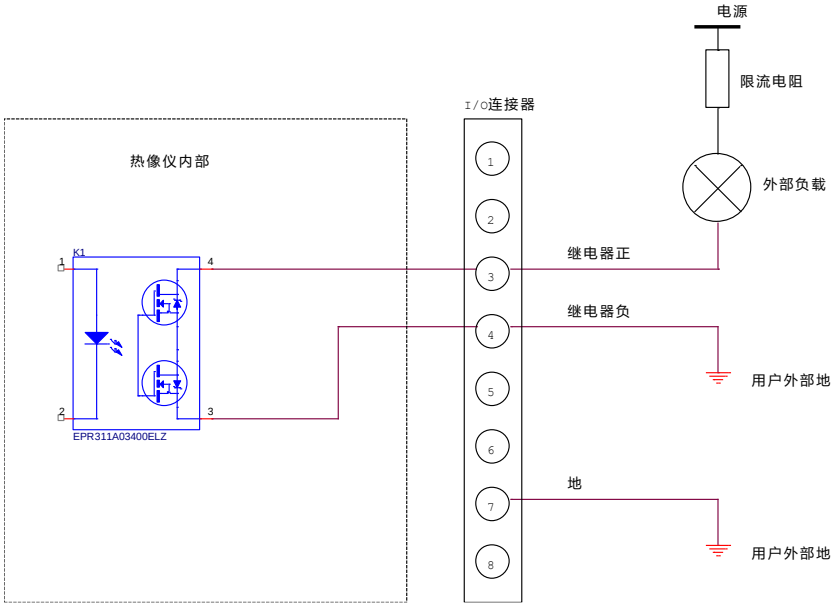
<table>
<tr><td rowspan="3">Power Supply</td><td>+</td><td colspan="2" rowspan="3">DC12/24V</td></tr>
<tr><td>–</td></tr>
<tr><td>⏚</td></tr>
<tr><td rowspan="2">RS485</td><td>A</td><td rowspan="2"></td><td rowspan="2"><table><tr><td>Protocol</td><td>Pelco-D</td></tr><tr><td>Default baud rate</td><td>9600</td></tr><tr><td>Default address code</td><td>1</td></tr><tr><td>Default parity bit</td><td>None</td></tr></table></td></tr>
<tr><td>B</td></tr>
<tr><td rowspan="2">Relay</td><td>+</td><td colspan="2" rowspan="2">Maximum conduction capacity
DC24V/1A, normally open</td></tr>
<tr><td>–</td></tr>
<tr><td rowspan="3">Optocoupler Isolation</td><td>Out</td><td colspan="2">Output</td></tr>
<tr><td>In</td><td colspan="2">Input</td></tr>
<tr><td>⏚</td><td colspan="2">GND</td></tr>
<tr><td>Analog Video</td><td>Video</td><td colspan="2">No function</td></tr>
<tr><td>RST</td><td>Reset</td><td colspan="2">Restore factory settings</td></tr>
<tr><td>LAN</td><td>RJ45 network port</td><td colspan="2">Ethernet</td></tr>
</table>

9.1.4 I/O Port Wiring Diagram

Pin	Pin Name	Description
1	RS485_A	485A
2	RS485_B	485B
3	RELAY_P	Relay positive
4	RELAY_N	Relay negative
5	OPTO_OUT	Optocoupler output
6	OPTO_IN	Optocoupler input
7	GND	Ground

8	CVBS	No function

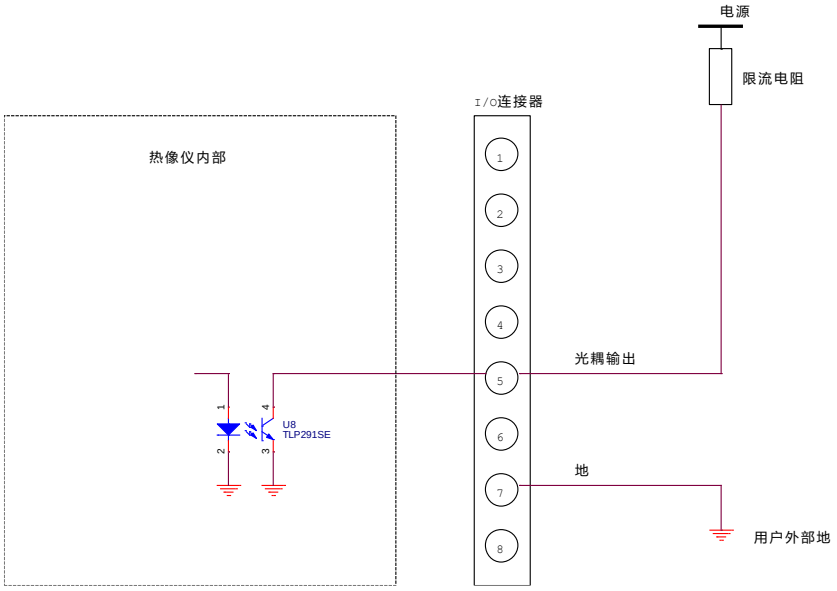
Relay Wiring Diagram



Usage and Attentions

1. For the power supply provided by the user, the voltage shall not be greater than 24V.
2. The relay can carry 1A current at most. Please calculate the load carefully.
3. If it is necessary to work in the circuit beyond the capacity, it is recommended to use it together with large capacity switch devices such as intermediate relay and contactor.
4. Current limiting resistance, **which is not inside the thermal imager**, is provided by the user outside the thermal camera according to the actual load.
5. The load is connected to the Pin 3 (relay positive) of the I/O connector
6. Pin 4 (relay negative) of the I/O connector is connected to the ground of the user side.
7. Pin 7 (ground) of the I/O connector is connected to the ground of the user side.

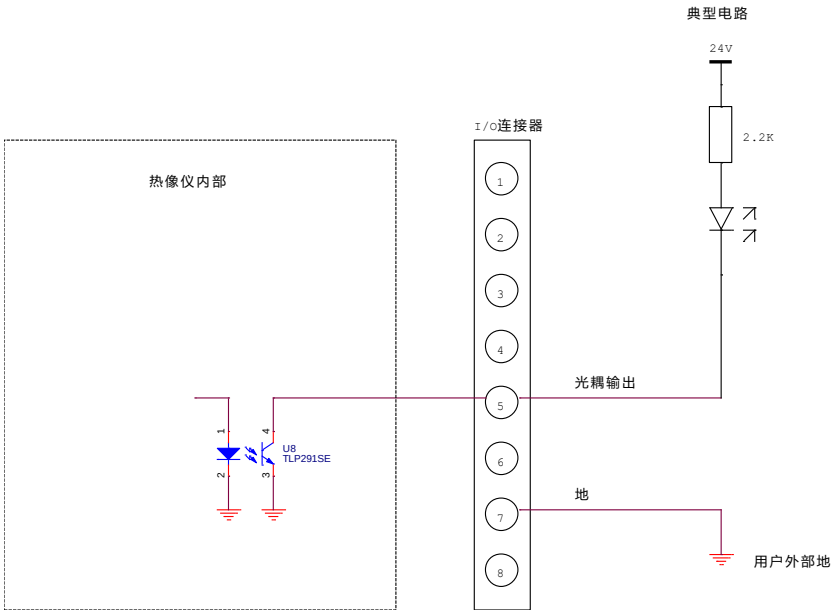
Optocoupler Isolated Output Wiring Diagram



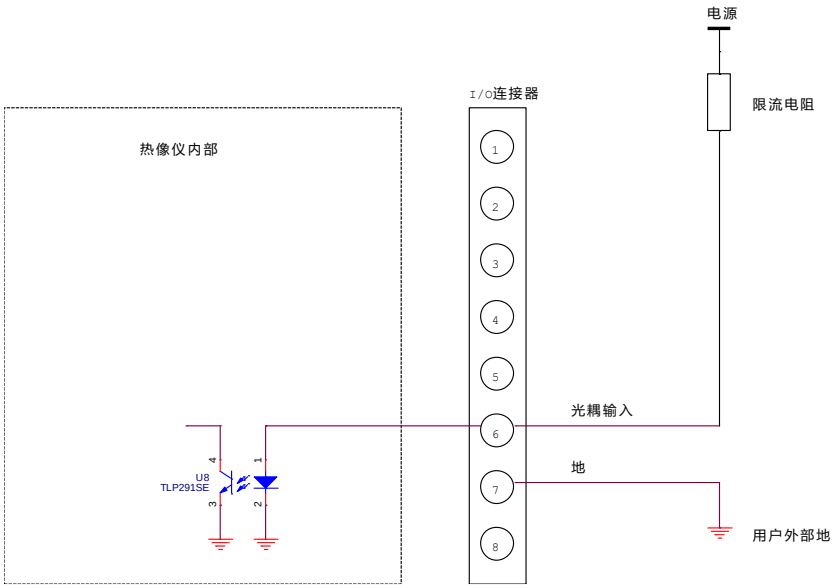
Usage and Attentions

1. The optocoupler isolated output cannot drive a large load, and the maximum output current is 50mA
2. The user needs to use the current limiting resistance according to the external load. The current limiting resistance is not inside the thermal camera and should be provided externally by the user.
3. For the power supply on the user side, the voltage shall not be greater than 24V.
4. Pin 5 (optocoupler isolated output) of the I/O connector is connected to the load
5. Pin 7 (ground) of the I/O connector is connected to the ground on the user side.

Typical Application Circuit of Optocoupler Isolated Output



Optocoupler Isolated Input Wiring Diagram



Usage and Attentions

1. Optocoupler isolated input is not used for heavy load input, and the input current is 5mA to 15mA
2. The user needs to use a current limiting resistance according to the external load. **The current limiting resistance is not inside the thermal imager** and should be provided externally.
3. For the power supply on the user side, the voltage shall not be greater than 24V.
4. Input voltage and logic value

Voltage	Description
+0 to +24 VDC	Safe operating voltage

+0 to +1.4 VDC	Logical zero
+1.4 to +2.8 VDC	Uncertain
+2.8 to +24 VDC	Logical one

5. User input is connected to Pin 6 (optocoupler isolated input) of the I/O connector
6. Pin 7 (ground) of the I/O connector is connected to the ground on the user side.

9.1.5 Device Configuration

Search Device and Update IP Address

After the device is powered on, connect the device to the computer using the network cable, and update the local IP address to 192.168.1.x, which cannot conflict with other devices (the factory IP address is set to 192.168.1.x, which can be found in the device label).

Open **IPconfig**. Double-click **IPConfig.exe**, and click **Refresh**, as shown in the figure below：

As shown in the figure above, the IP address and related information of the device can be read out.

If you need to update the IP address, select the device, enter a new IP address and subnet mask under the network configuration item on the right, and click the Quick Update button.

Please refer to the use of the RST button in the interface and function definition for restoring the factory IP address of the device. After resetting the device with RST button, the IP address of the device will return to the default: 192.168.1.100

9.1.6 FAQ and Troubleshooting

1. The device or IP is not found after the device is powered on. What is the reason for the abnormal connection?

   After the device is powered on, the power indicator is green, and the device starts normally 1-2 minutes after powering on. Before the device completes booting up, the IP address cannot be found. After the device starts normally, connect the computer to ensure that the network card connected to the computer is on the same network as the device and those IP addresses do not conflict.

2. What is the reason for inaccurate temperature measurement after device connection?

   Please confirm whether the thermal camera works normally and whether the parameter setting is reasonable according to the following aspects

   - The device has replaced the lens, and the parameter has been set correctly.
   - The emissivity parameter is accurate.
   - Focus accurately.
   - The thermal camera is stable after starting (15-30min).
   - The appropriate temperature range is selected.
   - Whether the automatic calibration mode is selected for the thermal camera (manual calibration is not recommended if the thermal camera is used normally in most cases).
   - Other temperature measurement parameters should be reasonably set (transmittance, ambient temperature, distance, etc.).

3. How to handle it when there is no image after connecting the device?

   - Check whether the computer firewall is turned off. If not, turn off the Windows firewall first.
   - Check whether other antivirus software (360, Computer Housekeeper, etc.) is running. If so, try to shut down or uninstall it and restart the computer before connecting the device.

9.2 Modbus Operation Instruction

This document explains how to use Modbus serial port output for the secondary development of the fixed thermal cameras.

The fixed thermal cameras only supports RTU mode for Modbus protocol based on serial port communication.

The fixed thermal cameras have been set to enable the Modbus slave mode in the factory state.

The data acquisition process of Modbus is as follows:

1. Set serial port parameters

   `put /peri/serial-port` (section 9.6 in Software Development Manual)

   The setting parameters are as follows: `baudrate`: baud rate; `csize`: data bit; `parity`: parity checking; `stopbit`: stop bit.

```
{
"baudrate": 9600,
 "csize": 8,
 "parity": "none",
 "stopbit": "one"
}
```

2. Create and update temperature measurement objects (refer to the class OWBCamera and the methods PutPoint and PutRegion in TADemo)

   `put /isp/instrument/objects/points/{0}`

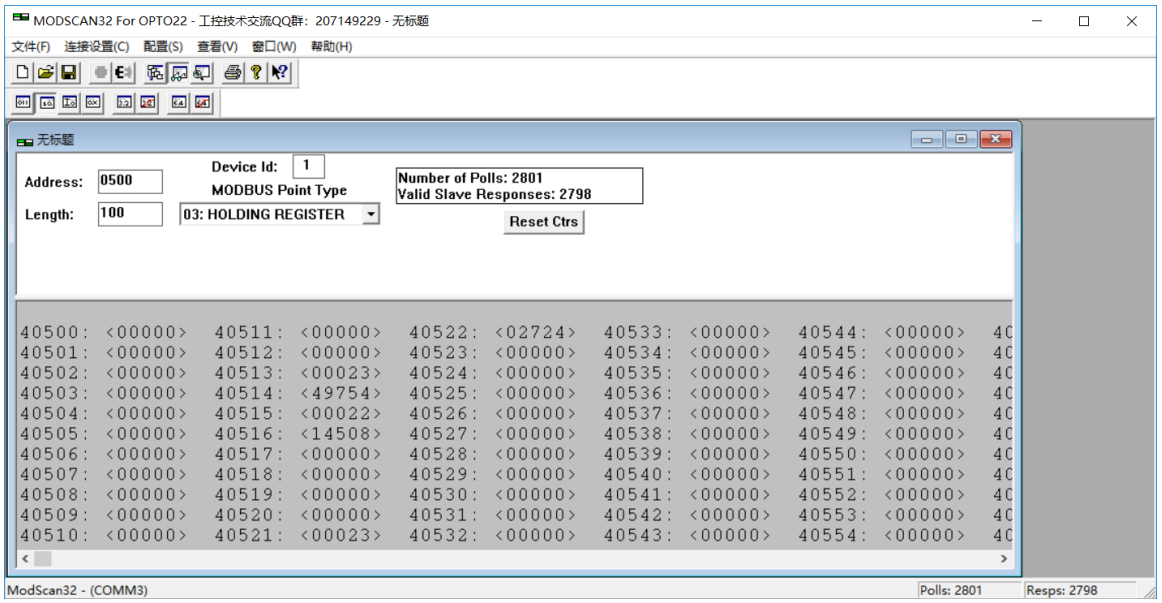
   put/isp/instrument/objects/regions/{0} (section 4.5 in Software Development Manual 4.5)

   The setting parameters are as follows: `pos`: point location; `emissivity`: emissivity; `mb-slot`: Modbus data block.

```
{
"pos":
 {
  "x": 55,
  "y": 50
 },
 "emissivity": 0.97,
 "mb-slot": 1
}
```

3. Read Modbus data (can be viewed through modscan32)

   Using the command code 0x03, the starting address of the measurement data area is 512. According to the description in the Software Development Manual, each data block occupies 8 bytes. Taking the mb-slot=0 in Step 2 as an example, the register address of the temperature data is 512 to 520.The first byte is for a 16-bit integer, and the last byte is for a 16-bit fixed-point decimal.


MODSCAN32 For OPTO22 - 工控技术交流QQ群：207149229 - 无标题
文件(F) 连接设置(C) 配置(S) 查看(V) 窗口(W) 帮助(H)
无标题
Address: 0500
Length: 100
Device Id: 1
MODBUS Point Type
03: HOLDING REGISTER
Number of Polls: 2801
Valid Slave Responses: 2798
Reset Ctrs
40500: <00000> 40511: <00000> 40522: <02724> 40533: <00000> 40544: <00000>
40501: <00000> 40512: <00000> 40523: <00000> 40534: <00000> 40545: <00000>
40502: <00000> 40513: <00023> 40524: <00000> 40535: <00000> 40546: <00000>
40503: <00000> 40514: <49754> 40525: <00000> 40536: <00000> 40547: <00000>
40504: <00000> 40515: <00022> 40526: <00000> 40537: <00000> 40548: <00000>
40505: <00000> 40516: <14508> 40527: <00000> 40538: <00000> 40549: <00000>
40506: <00000> 40517: <00000> 40528: <00000> 40539: <00000> 40550: <00000>
40507: <00000> 40518: <00000> 40529: <00000> 40540: <00000> 40551: <00000>
40508: <00000> 40519: <00000> 40530: <00000> 40541: <00000> 40552: <00000>
40509: <00000> 40520: <00000> 40531: <00000> 40542: <00000> 40553: <00000>
40510: <00000> 40521: <00023> 40532: <00000> 40543: <00000> 40554: <00000>
ModScan32 - (COMM3)
Polls: 2801
Resps: 2798

9.3 Quick Development Guide

To facilitate your secondary development of an online thermal imager, please read the content.

The online thermal imager is mainly communicated and controlled via Ethernet. This document is used to introduce DEMO examples and provide development assistance.

First, here is the overview of the SDK used in the DEMO example:

1. StreamSDK includes a general function interface for accessing streaming data of online thermal imagers. Please refer to the StreamSDK Manual for details;
2. Restc encapsulates the general access request interface of HTTP, including GET, PUT, POST, DELETE, and other methods;

C++ Development

- If you use C / C++ for development, include `StreamSDK.h` and `restc.h` header files (located in cpp/DLMFCDemo/include in the installation directory), link StreamSDK.lib and restc.lib library files (located in cpp/DLMFCDemo/lib in the installation directory), and then you can directly call the interface functions of StreamSDK and restc in the project.
- We provide a C++ Demo project (located in cpp/DLMFCDemo in the installation directory), which is included in the DEMO (OWB.MVIRs.sln) solution.
- The project includes the following features:

  Use StreamSDK to connect the H.264 video stream of the device, and you can switch between the mainstream and the substream

  Use StreamSDK to connect the original stream of the device and convert the original stream data into a visual image

  Switch between remote palette and local application palette

  Take snapshots (raw, video)

  Read and display the temperature at the mouse position

C# Development

- If you use C# for development, you can use similar methods in C# DEMO to load the `StreamSDK` and `restc`.
- We provide two C# DEMO projects (located in the csharp folder of the installation directory), which are included in the DEMO (OWB.MVIRs.sln) solution.
- `DLDemo` includes the following features:

  Use StreamSDK to connect the H.264 video stream of the device, and you can switch between the mainstream and the substream

  Use StreamSDK to connect the original stream of the device and convert the original stream data into a visual image

  Switch between remote palette and local application palette

  Display control of palette in H.264 video stream picture, and display control of the highest temperature and lowest temperature

  Read and display the temperature at the mouse position

Take snapshots (raw, video, temperature map), where the temperature map is saved in a user-defined format (.p7). The temperature map can be loaded offline, and the temperature value of each point in the temperature map can be obtained

Add / Update / Remove Users

- `TADemo` includes the following features:

  Display the mainstream

  Read and display the temperature at the mouse position

  Set temperature measuring point, measuring polygon, measuring line, and measuring mask

  Read the real-time temperature values of the measuring points and polygons in the marked stream through StreamSDK

  Modify emissivity and other temperature measurement parameters

9.4 Real Time Temperature Data Acquisition Process

This document explains how to obtain temperature data for the secondary development of an fixed thermal camera.

The process of obtaining temperature data in real-time is as follows:

1. Obtain the temperature mapping table

   `sensor/luts/{0}?list` (Software Development Manual 2.13)

   `Input: lens_index`, the current lens index, can be obtained through "`/sensor/lut`" (Software Development Manual 2.12).

   Output: a collection of the following data structure.

```
 {
"r": AD value,
 "t": Temperature in Celsius
 }
```

2. The temperature mapping table obtained in Step 1 is corrected according to the current emissivity, humidity, reflection temperature, ambient temperature, and distance of the thermal camera.(This correction can be made only when the above parameters change. For the above two steps, refer to the `UpdateFactoryLUT` function in `DLDemo`)
3. Get the fully-radiometric flow in the SDK through the callback function. For this step, refer to the `GrabberRaw` function in `DLDemo`.
4. The fully-radiometric is converted into real-time temperature data through the temperature mapping table. For this step, refer to the `GetTemperatureFrame` function in DLDemo.

9.5 Example Program Guide

9.5.1 General

The fixed thermal cameras is mainly communicated and controlled via Ethernet. This document introduces demo sample programs and provides development assistance.

9.5.2 Introduction to SDK Libraries

The SDK used in the example includes:

1. StreamSDK

   StreamSDK includes a general function interface for accessing streaming data of thermal cameras. Please refer to the StreamSDK Development Guide for details.

2. Restc

   Restc encapsulates the general access request interface of HTTP, including GET, PUT, POST, DELETE, and other methods.

9.5.3 C# Demo

If you use C# for development, you can use similar methods in the C# demo to load the StreamSDK and restc.

We provide 6 C# DEMO projects (located in csharp folder in the installation directory), which are included in the demo (OWB.MVIRs.sln) solution.

9.5.3.1 DLDemo

- Use StreamSDK to connect the H.264 video stream of the device, and you can switch between the mainstream and the substream
- Use StreamSDK to connect the raw stream of the device and convert the original raw data into a visual image
- Switch between remote palette and local applied palette
- Display control of palette in H.264 video stream picture, and display control of the highest temperature and lowest temperature
- Read and display the temperature at the mouse position
- Take snapshots (raw, video, thermal image), where the thermal image is saved in a user-defined format (refer to P7 in the Software Development Manual). The thermal image can be loaded offline, and the temperature value of each point in the thermal can be obtained
- Switch temperature range and lens
- Auto focus / manual focus (only for electric focusing device)
- Take snapshots of thermal image at custom intervals

9.5.3.2 TADemo

- Display the mainstream

- Read and display the temperature at the mouse position
- Set temperature measuring point, polygon, line, and measuring mask
- Read the real-time temperature values of the measuring points and polygons in the marked stream through StreamSDK
- Change emissivity and other temperature measurement parameters

9.5.3.3 MCDemo

- Up to 4 devices can be connected at the same time
- Select calibration mode
- Video gray level adjustment and contrast adjustment

9.5.3.4 RestDemo

- Generate a template example according to the interface definition in the Software Development Manual

9.5.3.5 TAGDemo

- An example of obtaining the numerical stream described in the Software Development Manual

9.5.3.6 VIDEODemo

- An example of obtaining the raw stream described in the Software Development Manual
- Generate real-time temperature values from raw streams and LUT tables in the device.

9.5.4 C++ Demo

If you use C / C++ for development, include StreamSDK.h and restc.h header files (located in sdk in the installation directory), link StreamSDK.lib and restc.lib library files (located in sdk in the installation directory), and then you can directly call the interface functions of StreamSDK and restc in the project.

We provide 4 C++ demo projects (located in cpp folder in the installation directory), which are included in the demo (OWB.MVIRs.sln) solution.

9.5.4.1 DLMFCDemo

- Use StreamSDK to connect the H.264 video stream of the device, and you can switch between the mainstream and the substream
- Use StreamSDK to connect the raw stream of the device and convert the raw stream data into a visual image
- Switch between remote palette and local applied palette
- Take snapshots (raw and video)
- Read and display the temperature at the mouse position

9.5.4.2 RESTMFCDemo

- Generate a template example according to the interface definition in the Software Development Manual

9.5.4.3 TAGMFCDemo

- An example of obtaining the numerical stream described in the Software Development Manual

9.5.4.4 VIDEOMFCDemo

- An example of obtaining the raw stream described in the Software Development Manual
- Generate real-time temperature values from raw streams and LUT tables in the device.