Online Acoustic Camera
HZ-FA-371T

Released March 31, 2026

Document version	V1.0
software version	V1.0

Content

Revision information

releases	dates	author	Change Description
V1.0	2026.03.31	Ethan	initial version

1 System Introduction

1.1 Overview

The HZ-FA-371T Acoustic & Thermal Imager is an advanced visual diagnostic device integrating multiple sensing technologies: sound, visible light, and infrared thermal imaging. By synchronously capturing and fusing acoustic information (inaudible to the human ear), temperature distribution (invisible to the naked eye), and high-definition visible light images, it innovatively generates a new "acoustic-thermal-optical" composite image.

Acoustic imaging is based on microphone array measurement technology. It determines the position of sound sources by measuring phase differences of sound waves arriving at each microphone (according to the phased array principle), measures the amplitude of the sources, and displays the spatial distribution of sound sources as an image — a sound field distribution map (sound image) where color and brightness represent intensity. The device integrates sound acquisition, image acquisition, and sound source localization visualization, enabling visual diagnosis of equipment faults. It is compact and lightweight, can be mounted on various inspection robots or PTZ platforms, further empowering inspection methods.

Our product adopts internationally leading ESR (Environmental Sound Recognition) technology and sound source localization based on beamforming theory. It can passively measure the position and radiation status of sound sources, covering the full frequency range from audible to ultrasonic. The acquired sound is visualized as a color contour map. The sound field distribution is perfectly superimposed with visible light video, infrared, and thermal images, providing multi-dimensional visualized information, greatly improving the efficiency and accuracy of fault diagnosis and equipment status assessment in complex industrial environments.

The HZ-FA-371T is widely used in industrial inspection (e.g., partial discharge detection in high-voltage equipment, gas leak detection, mechanical fault prediction), R&D, building assessment, security monitoring, etc. It provides engineers and professionals with an unprecedented powerful tool for predictive maintenance and intelligent management.

Figure 1-1 Gas Leakage Monitoring Scenario

1.2 Product Features

- **Integrates sound, visible light, and infrared monitoring.**
- **Wide detection frequency band covering both audible and ultrasonic ranges.**
- **Visualized automatic sound field detection – intuitive, fast, and precise localization.**
- **Supports steady continuous detection mode and transient detection mode.**
- **Supports imaging for high-pressure pipeline gas leakage.**
- **Provides HTTP interface and protocol for secondary development.**

2 Product Specification

2.1 Exterior Dimensions

Figure 2-1 shows the actual appearance of the product.

Figure 2-1 Product Appearance

Figure 2-2 shows the product's external dimensions.

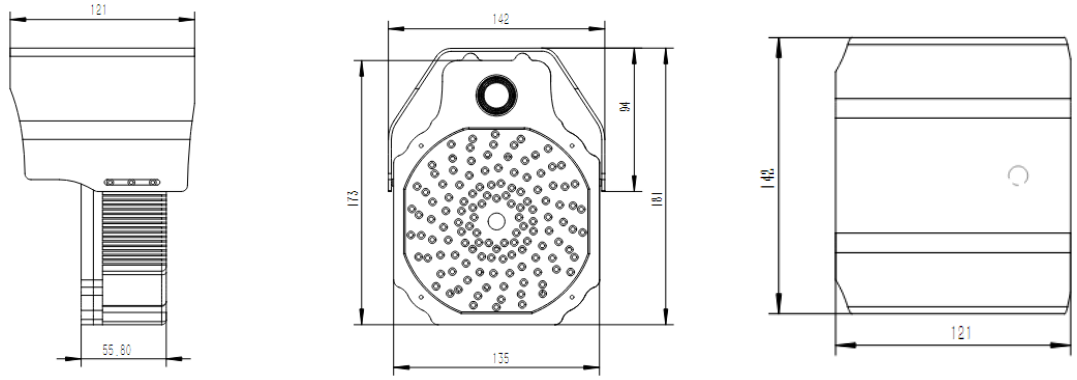


Figure 2-2 Exterior Dimension Drawing

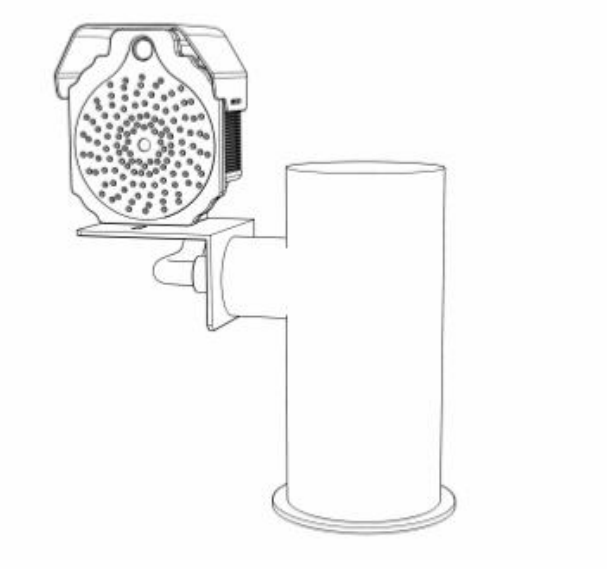
2.2 Performance indicators

Parameter type	Specification
Model	HZ-FA-371T (overseas version)
Acoustic imaging parameters	
Number of acoustic sensors	128
Acoustic sensor type	MEMS digital microphone
Sensitivity	-26±1 dBFS (1kHz sensitivity @1Pa)
Measurement SPL range	-30 dB ~ 120 dB
Frequency band range	2 kHz ~ 65 kHz
Minimum imaging sensitivity (at 1m)	5dB SPL @18kHz, 7dB SPL @30kHz, 11dB SPL @42kHz, 14dB SPL @60kHz
Multi-target imaging	Supported
Acoustic imaging frame rate	⩾30 FPS
Detection distance	0.3 m ~ 130 m
Typical application scenario	Compressed gas leakage detection
Infrared camera	
Sensor type	Uncooled VOx
Detector resolution	640×512
Pixel pitch	12 μm
Spectral band	8 ~ 14 μm
NETD	≤40mK (@25°C, F#=1.0)
Temperature measurement range	-20°C ~ 150°C and 100°C ~ 550°C
Temperature measurement accuracy	±2°C or ±2% of reading (whichever greater)
Spatial resolution	0.8 mrad
Focal length	15 mm
Field of view	29.3° (H) × 23.4° (V), 37.5° (D)
Frame rate	30 Hz

Visible light camera	
Resolution	13 MP
FOV	72°±3°
Video resolution	1920×1080@30fps
Power supply	
Rated input	DC 12V 3A
Others	
Supported PTZ protocol	PELCO-D
Operating temperature	-20°C ~ +55°C
Storage temperature	-30°C ~ +70°C
Humidity	10% ~ 95% non-condensing
Weight	<1.6 kg
Data interface	Ethernet

3 System deployment

3.1 Equipment Installation



The device is compact and lightweight. It can be installed on a PTZ, robot, or other suitable location. Ensure the device is not obstructed by other objects, and minimize vibration from the mounting carrier to avoid affecting detection performance. The orientation can be adjusted according to the carrier's requirements; structural parts for installation must be prepared by the user.

3.2 network topology

Figure 3-1 shows the typical network connection topology. The device and computer are generally in the same LAN. Live video can be viewed and parameters configured via HTTP protocol.

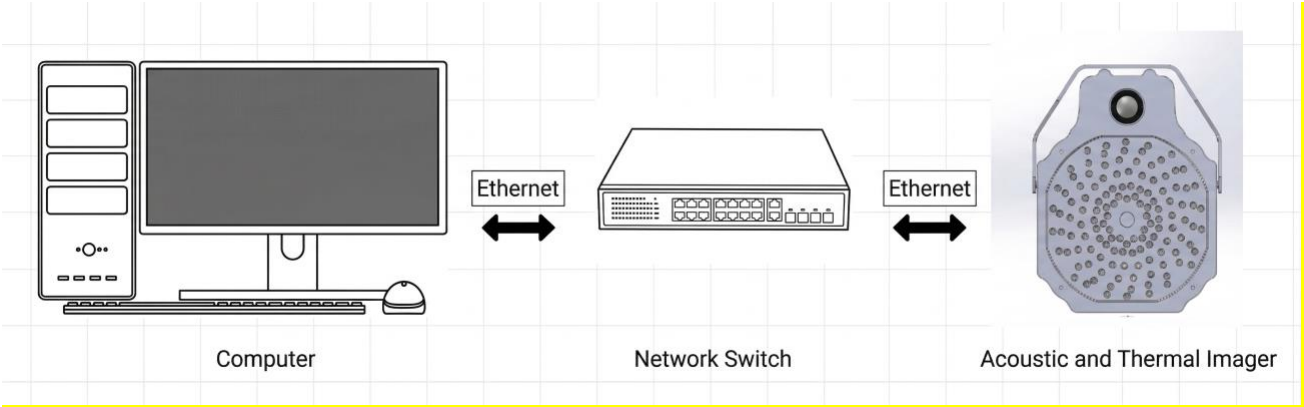


Figure 3-1 Network Connection Topology Diagram

Note: The computer IP must be in the same subnet as the device IP.

4 Function Introduction

4.1 System Login

The device has a built-in web page. After connecting to the network, enter the device IP in a browser to access the login page (Figure 4-1). Default credentials:

Default User Name: admin

Default Password: 123456

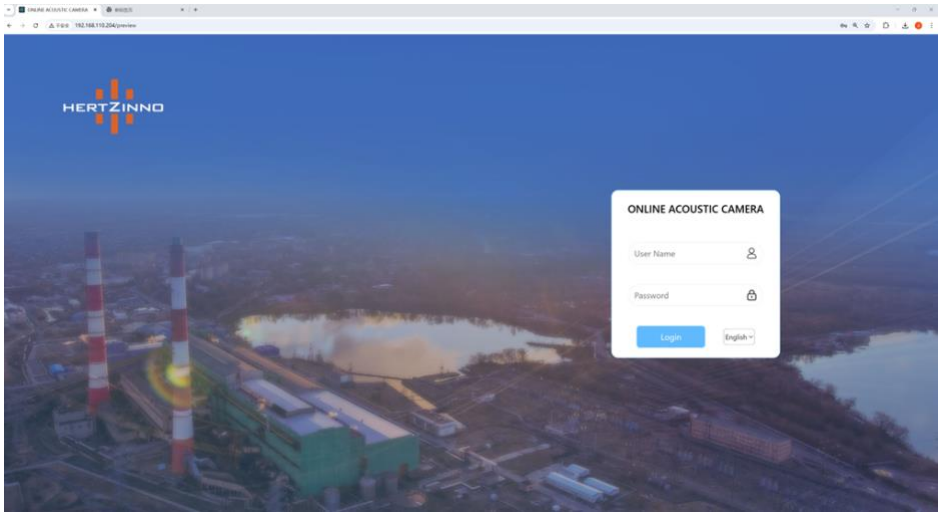


Figure 4-1 Login Page

4.2 Screen Preview

After logging in, you will enter the screen preview interface, the left side of the page displays the video screen in real time, and the right side of the page is the parameter console. The "Leak Detection" in the upper right corner is the current working mode of the device, and the power button ⏻ can exit the system login.

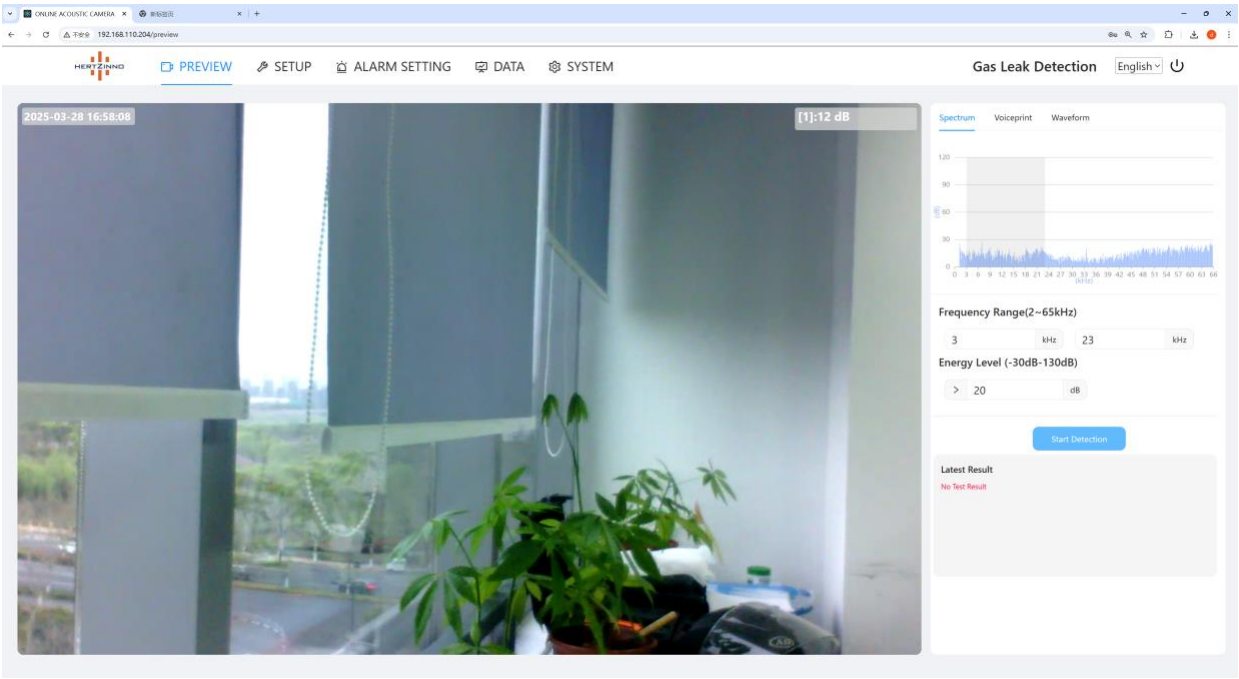


Figure 4-2 Screen Preview Interface

4.2.1 Preview

The system time is displayed in the upper left corner of the video, and the real-time environmental sound energy of the current monitoring frequency band is shown in the upper right. When the environmental energy exceeds the set threshold, the energy value turns red, and an acoustic imaging point appears on the real-time video.

The video area supports visual-audio fusion, thermal imaging, and picture-in-picture (PiP) modes.

The bottom left corner displays the currently estimated leakage rate and corresponding financial loss.

The control bar below the video allows you to switch display modes, take photos, record videos, control the PTZ rotation angle, view real-time PTZ position data, and use online audio monitoring

4.2.2 Setup of video

The functions of each part of the console are described below:

- Spectrum: Analyze and draw the spectrum of the current environmental sound signals in real time, through the spectrogram can be visualized in the environment of the distribution of the sound of each frequency band, to provide a reference for the imaging frequency band adjustment.
- Voiceprint: Real-time analysis mapping Characterization of the current ambient sound signal in both time and frequency dimensions.
- Waveform: real-time analysis and drawing of the current environmental sound signal

changes over time, visualizing the signal amplitude, phase and other changes over time.

- Imaging Band: Used to adjust the frequency band range of the sound currently being monitored. The adjustable range is 2~ 65KHz and the numeric entry must be an integer.
- The difference between the upper and lower limits cannot be greater than 20 or less than 2 and the upper limit must be greater than the lower limit. After adjusting the frequency band, the shaded area in the spectrum graph is the currently selected frequency band range.
- Energy Level: Used to adjust the threshold value for detection. When the environmental energy value of the scene is higher than the threshold value, the status will change to abnormal, and it is capable of real-time acoustic field imaging, and the acoustic field imaging marker screen can be observed in the video.

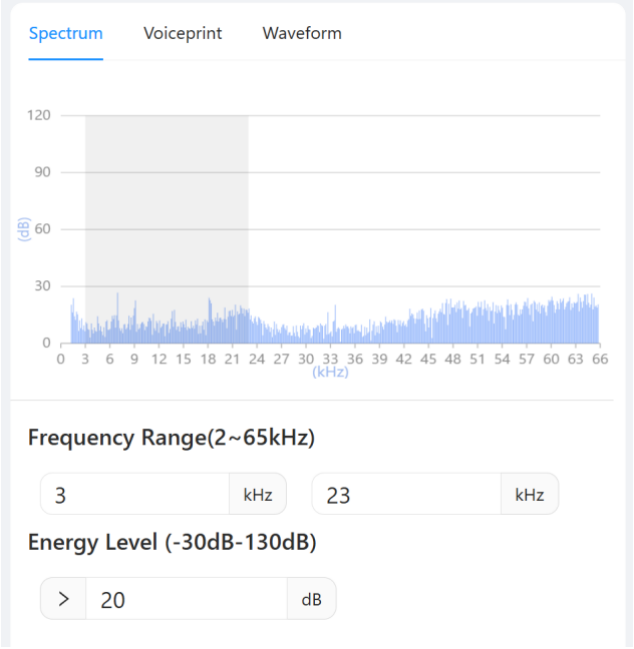


Figure 4-3-1 Console Interface

- Start Detection: Used for on-site detection and recording the detection results, after starting detection, it will record the abnormal point information in real time and generate corresponding data which can be viewed in the data view page. (Enable single detection mode)

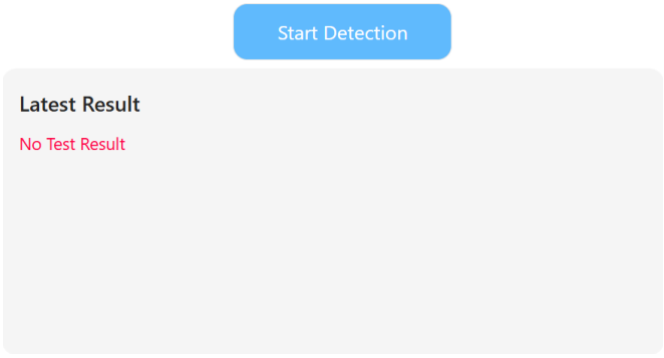


Figure 4-3-2 Console Interface

- Continuous Detection (Smart Patrol Mode): Loops the detection process automatically. Anomalies will trigger alarms and real-time records, viewable in the data page.

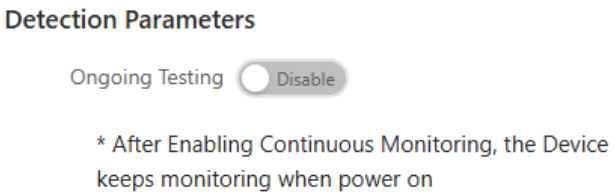


Figure 4-3-2 Console Interface

4.3 Setup

Entering the parameter configuration interface allows you to change the alarm settings, video settings and other parameters.

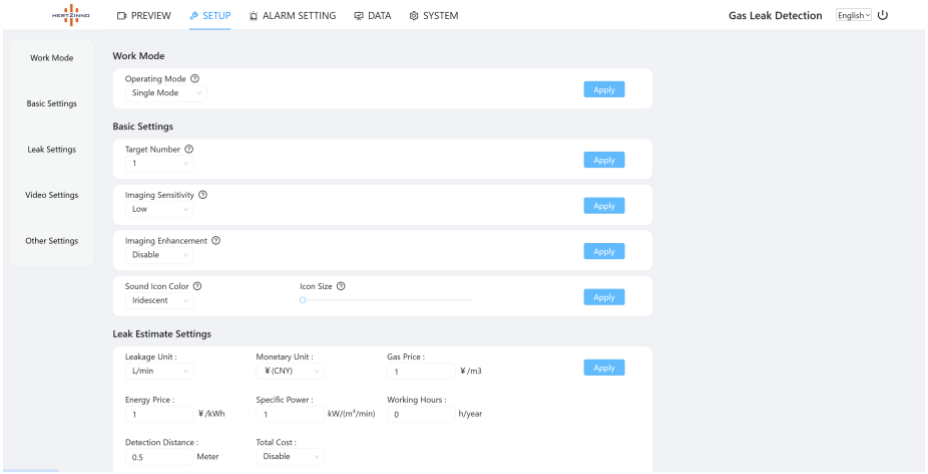


Figure 4-4 Parameter Configuration Screen

4.3.1 Work mode

The device operation mode can be set to single detection or intelligent cruise.

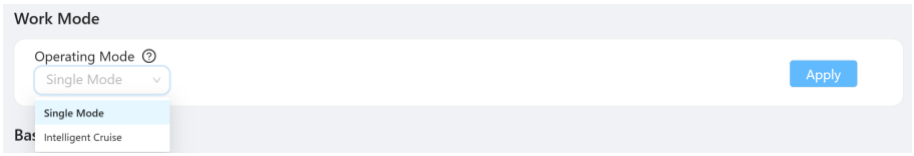


Figure 4-5 Mode Setting

- Single time mode: It means that the detection is triggered according to the demand of the site, and stops after one detection is completed, waiting for the next trigger.
- Intelligent Cruise Mode: When Intelligent Cruise Mode is turned on, the alarm policies that are set to be turned on will be detected automatically and cyclically, so there is no need to repeat the trigger manually.

4.3.2 Basic settings

The base contains settings for detection target, imaging sensitivity, imaging enhancement, and cloud color size.

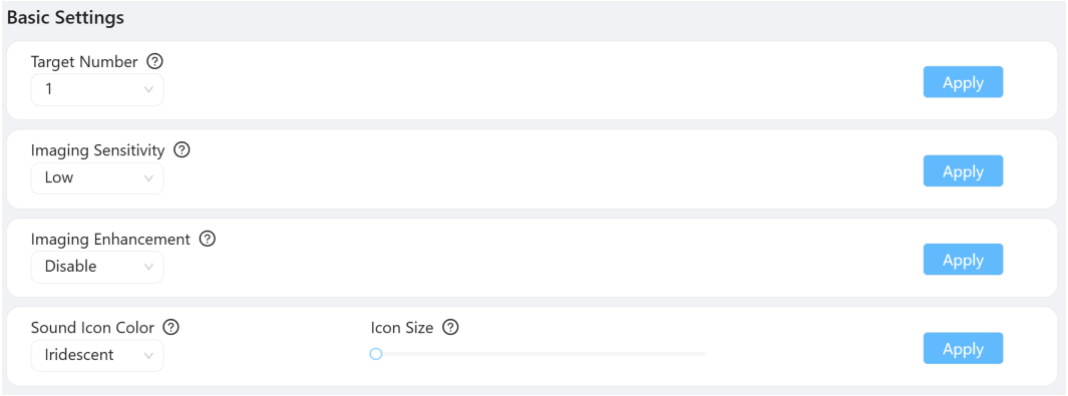


Figure 4-6 Basic Settings

- Targe Number: It is the maximum number of target points within the sound field that the system can mark and output position information in real time.
- Imaging sensitivity: is the degree of sensitivity of the sound field detection system to the sound signal. The higher the sensitivity the better the real-time performance.
- Imaging Enhancement: Improves the stability of the imaging point.
- Sound Icon Settings: You can customize the color and size of the cloud map of the imaged points.

4.3.3 Video Settings

Configurable video resolution, video bit rate, video flip and other content, video encoding default H.264, video frame rate default 30fps (can not be modified); video resolution support 1920 * 1080, 1280 * 720; video stream support 4096, 2048, 1024, 512; video flip function support horizontal mirror, vertical mirror, center mirror and so on.

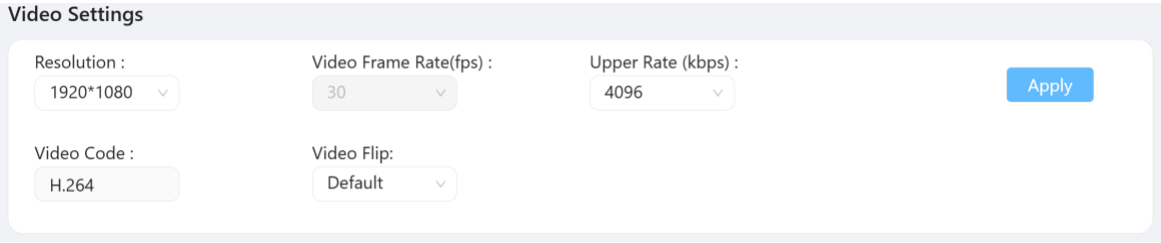


Figure 4-7 Video Settings

4.3.4 Thermal Imaging Settings

Emissivity, distance, ambient temp, humidity, temperature unit (°C/°F/K), range, palette (white

hot, lava, iron red, etc.), cursor enable.

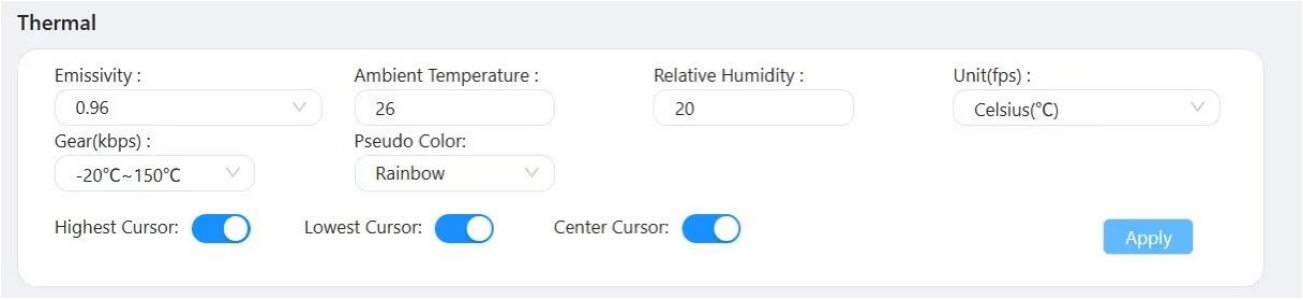


Figure 4-8 Thermal Imaging Settings

4.3.5 Leak Estimate Settings

Leakage cost estimation can be made according to the leakage location, and the setup parameters include leakage volume, leakage unit, currency unit, gas value, energy cost, specific power, working hours, and detection distance; after the leakage cost is turned on and displayed, the real-time information can be viewed on the screen preview page;

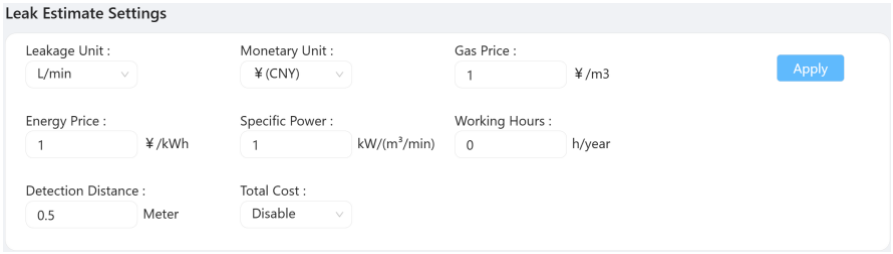


Figure 4-9 Leakage Settings

4.3.6 Alarm Push Settings

Supports custom push addresses. When alarm push is enabled, alarm content will be sent to the specified push address when the system generates an alarm.

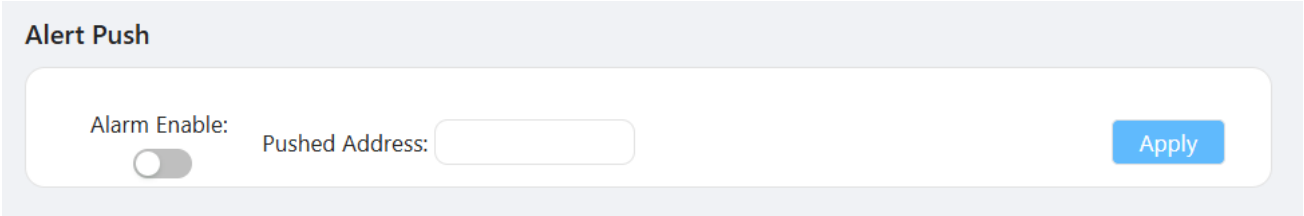


Figure 4-10 Alarm Push Settings

4.3.7 Other settings

Other settings include Focus Mode and Partition Mode. Turning on Focus Mode, the system will only detect foreign sound points within the range of the focus box; the same turning on Partition Mode, the system will only detect foreign sound points within the range of the customized box selection.

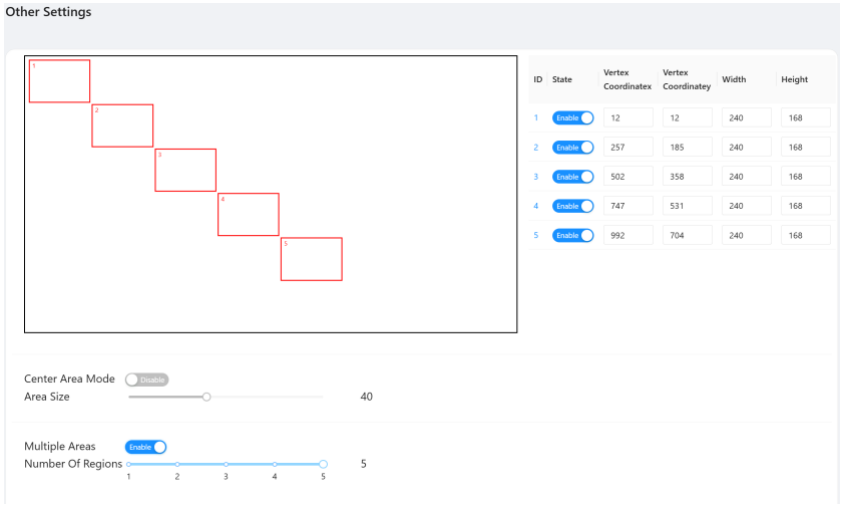


Figure 4-11 Other Settings

4.4 Alarm Setting

On this page, users can customize the alert policy according to their needs, and the custom alert policy contains the following:

- Strategy Name: Strategy Name
- Alarm conditions: including decibel alarm, main frequency alarm, spectrum comparison, users can flexibly select and set these alarm conditions according to the actual needs to ensure that the alarm can be accurately triggered in specific situations.
- Detection cycle: the length of time for a single detection can be set from 1 to 30 seconds. Users can adjust the length of the detection cycle to balance the detection accuracy according to the actual demand.
- Sensitivity: This parameter measures the rate at which the detection threshold is exceeded in a single detection. The higher the sensitivity, the more sensitive the system's response to abnormal signals, but it may also lead to an increase in the false alarm rate. Therefore, users need to set the sensitivity parameter reasonably according to the actual situation.
- Alert Level: This field is used to mark the urgency or importance level of the policy. By setting different alert levels, users can prioritize different types of alerts to ensure that urgent and important alerts are prioritized when resources are limited.

- Diagnostic Description: This field provides a way for the user to record a description of the policy definition. Users can enter a detailed description of the background, purpose, applicable scenarios, and other information about the strategy's development for subsequent review and understanding. By completing the diagnostic description, the user can better communicate the intent of the strategy's development and make it easier for other users to understand and apply the strategy.

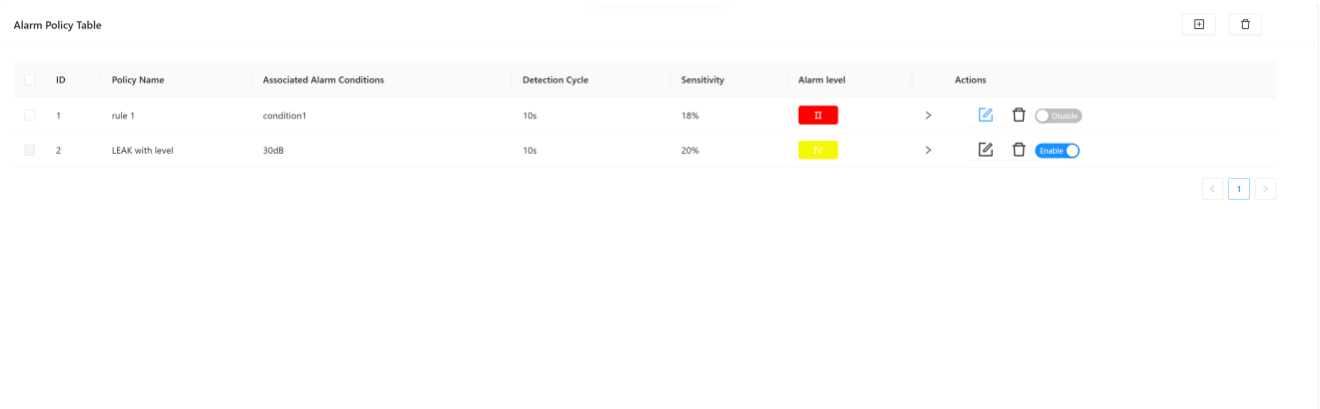


Figure 4-4-1 Alarm Policy Interface

4.5 Data Viewing

This page mainly records the file information of the test records generated during the operation of the device. Since the device has limited storage space, it will only keep the file information of the last 100 detections, and if it exceeds the limit, it will delete the earliest saved records and record new data to realize cyclic storage. In this page, you can search and query the detection records according to the time of detection, and you can view the characteristic values of the environment at the time of detection, etc. The detection records support the viewing and downloading of pictures and videos generated at the time of detection.

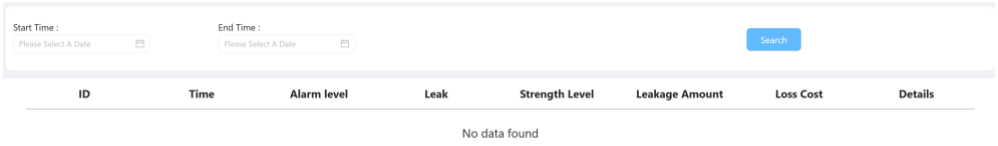


Figure 4-10 Data View Interface

4.6 Statistical mapping

In order to comprehensively understand and analyze the acoustic characteristics of each sound field point during a historical day, we need to provide detailed statistics and visualization of the

sound pressure information as well as the dominant frequency information at a specific moment in time.

- **Sound pressure information statistics:**

For each sound field point during a historical day, its sound pressure level is accurately measured and recorded at specified moments. This data will help us to understand the distribution of sound intensity at each site over time and assess whether it meets the relevant noise standards or has potential acoustic problems.

- **Master frequency information statistics:**

In addition to the sound pressure information, we will also count the main frequency information of each sound field point at the specified moment. The dominant frequency is the frequency component with the highest energy in the sound signal, which is important for determining the source, nature and possible acoustic impact of the sound signal. Through the statistics of the main frequency information, we can further analyze the sound characteristics of each sound field point, such as whether there is a specific noise source or vibration source.

Figure 4-6-1 Statistical Mapping Interface

4.7 System Configuration

Entering the system configuration interface allows you to make changes to the device time, network, user management, RTSP management, and system information.

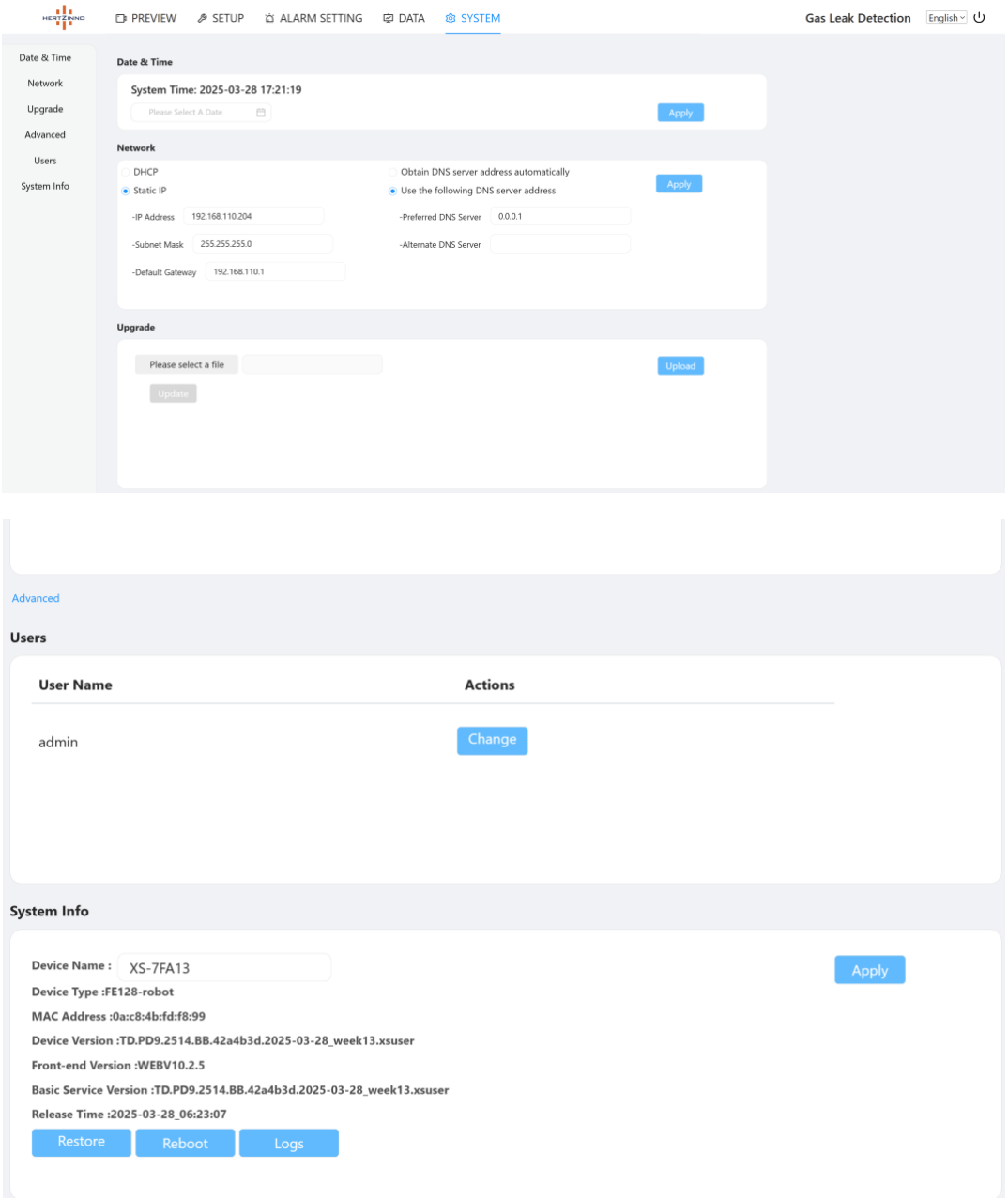


Figure 4-11 System Configuration Interface

4.7.1 Date & Time

The system time is the current time of the device, click the calendar icon , users can set the device time by themselves.

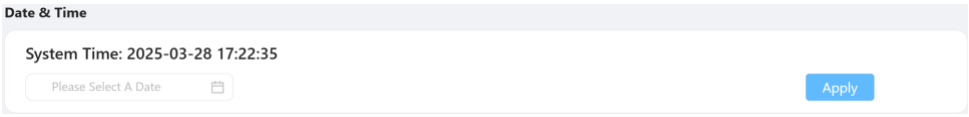
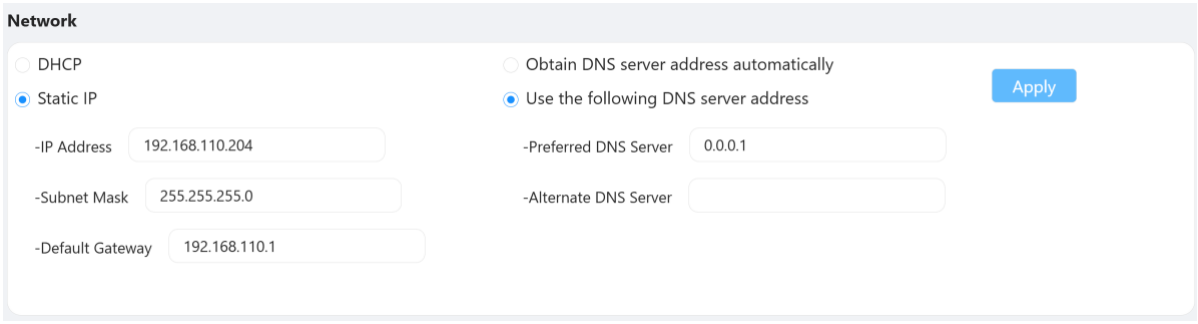


Figure 4-12 Time Settings

4.7.2 Network Configuration

The factory default IP address of the device is 192.168.2.168, if you need to change it, you can



easily adjust it through its network configuration interface. Simply go to the appropriate setting option, enter the new IP address you wish to set, and click the "Set" button to complete the modification. Please note that once the settings are complete, the system will automatically disconnect the current login session to ensure that the configuration takes effect immediately. At this point, you will need to re-enter the new IP address you just set in the address bar of your browser in order to log in and access the system again.

Figure 4-13 Network Configuration

4.7.3 Upgrade

The device supports firmware upgrade, when there is a version update of the software, the obtained update package can be loaded in the device as a remote upgrade.

Method of operation:

1. **Select the update package:** First, please click the "Select File" button, from your local storage to select the update package file you want to use for upgrading.
2. **Upload the update package:** After selecting the file, please make sure your selection is correct and click the " Upload" button again. At this point, the system will begin to upload the update package file to the device.
3. **Execute the upgrade operation:** After the file is uploaded successfully, you will see a "Update" button becomes operable. Please click this button, the device will immediately start to execute the upgrade process of the firmware.

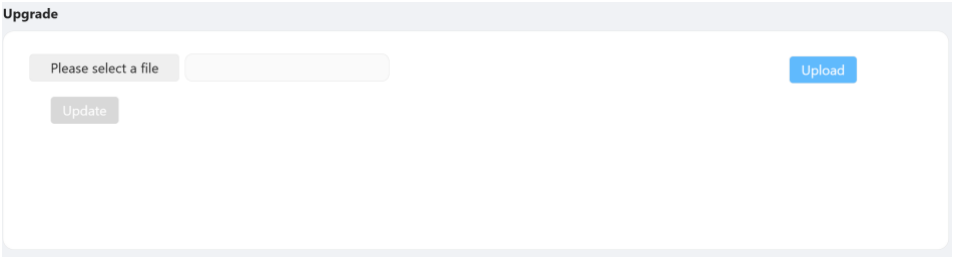


Figure 4-14 System Upgrade

4.7.4 User management

Click Change Password, users can change the login password and confirm it after change.

Default account: admin (not modifiable) Password: 123456

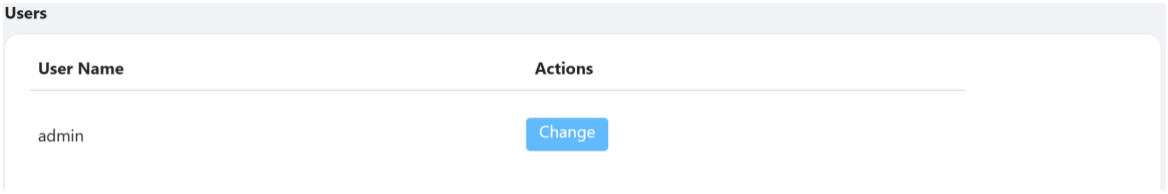


Figure 4-15 User Management

4.7.5 Advanced Settings

The advanced settings include RTSP settings, timed reboot settings.

The RTSP Settings are where users are allowed to configure RTSP in detail. The user has the following privileges:

1. Enable/Disable RTSP: Users can freely choose to enable or disable RTSP service according to the actual needs.
2. set the maximum number of connections: different application scenarios, users can customize the setting of the maximum number of concurrent connections allowed by RTSP.
3. custom link address: In order to provide more flexible access, the user has the right to customize the link address of the RTSP.
4. view the current number of connections: In order to facilitate user monitoring and management, the system displays the current number of RTSP connections in real time.

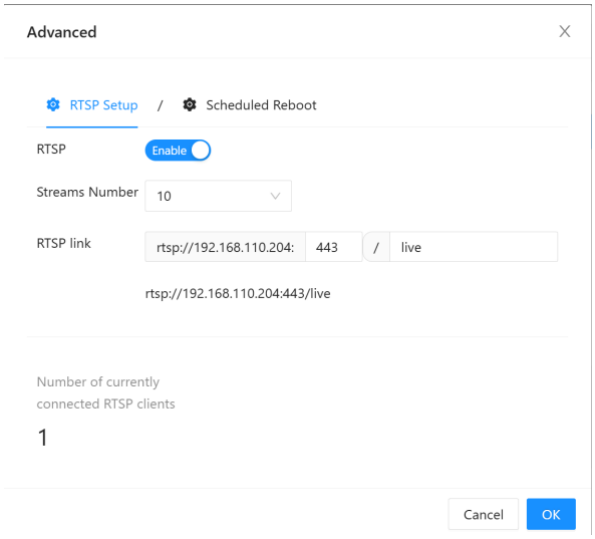


Figure 4-16-1 RTSP Settings

The Timed Reboot setting allows the user to reboot at regular intervals or at intervals.

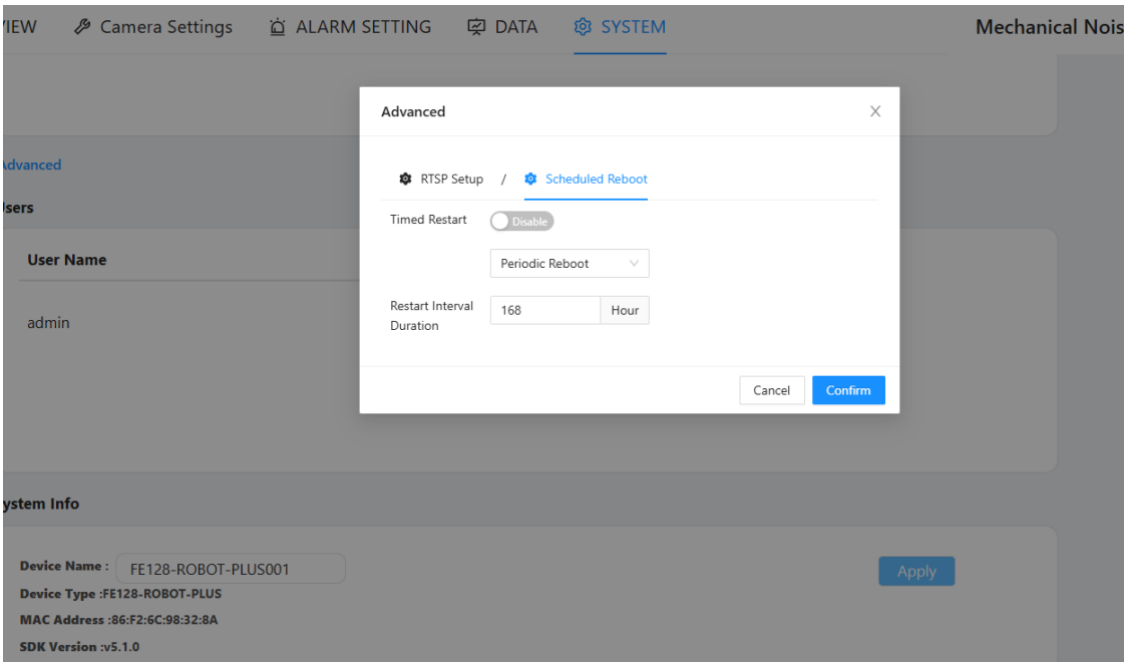


Figure 4-16-2 Timer Reboot Device Interface

4.7.6 System Information

System Information displays the specific information of the current device and supports custom naming of the device.

- Restore Initialization: One-click to restore default settings. Information such as IP, username, and password will be reset to default values, and all historical data will be deleted.
- Restart Device: Click the restart device button to remotely reboot the device. After rebooting, the system will automatically exit; you will need to re-enter the IP address in the browser to access the system once the device boots up automatically.
- Device Logs: Click to set logs, and the system will automatically download the log file locally, named system_logs.log.
- Device Self-Test: Automatically detects whether the hardware microphones are working properly.

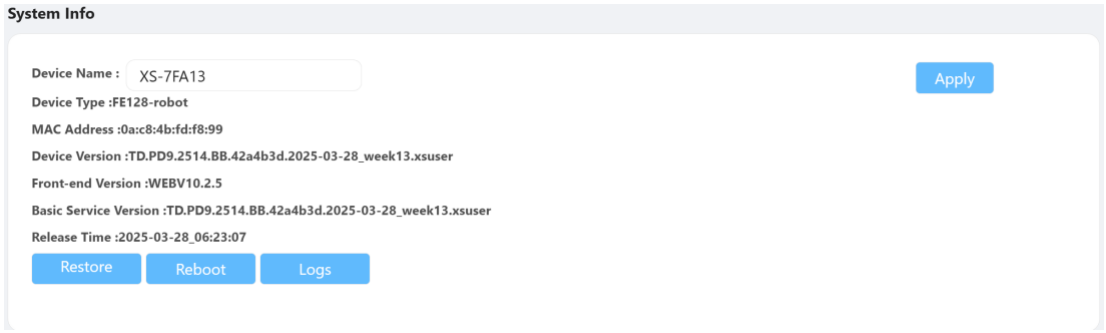


Figure 4-17 System Information

5 Open Interface Description

5.1 HTTP interface

5.1.1 Get token authentication

Function description: authentication token distribution interface, through the authentication of user name and password, return encrypted token value.

All subsequent interfaces require a token in the HTTP header to be attached to the Authorization for authentication. If authentication fails or the token is invalid, a 401 error is returned.

Request format: http://{IP}/login

Request method and URL: POST /login

Interface request body: Yes

Example of a ciphertext request:

```
{
        "username": "admin",
        "password": "********************"
}
```

Description of the interface request parameters:

Parameter	Type	Description
username	string	user ID
password	string	Password + lfxs MD5 encrypted ciphertext, 32 bit lowercase

The interface returns JSON data as follows:

```
{
        "token":  "***************************"
}
```

5.1.2 Get Sound Field Configuration

Function Description: Used to get the values of parameters such as MIC matrix sampling rate (unit kHz), analyzing frequency band upper limit (unit kHz), analyzing frequency band lower limit (unit kHz).

Request format: http://{IP}/system/acoustic

Request method and URL: GET /system/acoustic

Interface request body: none

Returns JSON data as follows:

```
{
  "freq_up_limit": 25,
  "freq_down_limit": 20,
  "samplerate": 13200,
  "ret": 200,
  "msg": "SUCCESS",
}
```


Description of interface return parameters

Parameter	Type	Required	Description
freq_up_limit	integer	Yes	Upper limit of the analyzed frequency band (in KHz)
freq_down_limit	integer	Yes	Lower limit of analyzed frequency band (in KHz)
samplerate	integer	Yes	MIC Matrix Sampling Rate in KHz
ret	integer	Yes	return code
msg	String	Yes	Back to information

5.1.3 Setting up the sound field configuration

Function Description: Used to set the parameter values of MIC matrix sampling rate (unit KHz), upper limit of analyzing frequency band (unit KHz), lower limit of analyzing frequency band (unit KHz), and so on.

The Analyze Band Upper and Analyze Band Lower limits are used to adjust the frequency band range of the sound currently being monitored. The adjustable range is 2~ 65KHz, and the value input must be an integer. The difference between the upper and lower limits cannot be greater than 20 or less than 2 and the upper limit must be greater than the lower limit. 96KHz and 132KHz are supported as the sampling rate of the MIC matrix, and it is not recommended to change it during use.

Request format: http://{IP}/system/acoustic

Request method and URL: PUT /system/acoustic

Interface request body: Yes

Headers:

Parameter	Value	Required
Content-Type	application/JSON	Yes

Request Example:

```
{
  "freq_up_limit": 25,
  "freq_down_limit": 20,
  "samplerate": 13200,
}
```

Description of interface request parameters

Parameter	Type	Required	Description
freq_up_limit	integer	No	Upper limit of analyzed frequency band (in KHz) (4~65 integers)
freq_down_limit	integer	No	Lower limit of analyzed frequency band (in KHz) (2~63 integers)
samplerate	integer	No	MIC matrix sampling rate in KHz (96000 or 132000)

5.1.4 Get Imaging Configuration

Function Description: Used to obtain the values of parameters such as energy threshold, number of detected targets, etc.

Request format: http://{IP}/system/image

Request method and URL: GET /system/image

Interface request body: none

Returns JSON data as follows:

```
{
  "energy_threshold": 25,
  "target_number": 1,
  "imaging_sensitivity":0,
  "imaging_enhance":0,
  "ret": 200,
  "msg": "SUCCESS",
}
```

Description of interface return parameters

Parameter	Type	Required	Description
energy_threshold	integer	Yes	imaging threshold
target_numbe	integer	Yes	Target number
imaging_sensitivity	integer	Yes	Imaging filter sensitivity (filter level) in the range [0,2].

imaging_enhance	integer	Yes	Imaging Enhancement, value range [0,2], where 0 means off
ret	integer	Yes	return code
msg	String	Yes	Back to information

5.1.5 Setting up the imaging configuration

Function Description: Used to set the values of parameters such as imaging threshold, number of detected targets, etc.

The imaging threshold is used to adjust the threshold value for detection. When the on-site environmental energy value is higher than the threshold value, the status will change to abnormal and the sound field imaging can be performed in real time, and the sound field imaging marker screen can be observed in the video. The number of detection targets currently supports up to 3 detection targets.

Request format: http://{IP}/system/image

Request method and URL: PUT /system/image

Interface request body: Yes

Headers:

Parameter	Value	Required
Content-Type	application/JSON	Yes

Request Example:

```
{
  "energy_threshold": 25,
  "target_number": 1,
  "imaging_sensitivity":0,
  "imaging_enhance":0,

}
```

Description of interface request parameters

Parameter	Type	Required	Description
energy_threshold	integer	Yes	imaging threshold
target_numbe	integer	Yes	Target number
imaging_sensitivity	integer	Yes	Imaging filter sensitivity (filter level) in the range [0,2].

imaging_enhance	integer	Yes	Imaging Enhancement, in the range [0,2], where 0 means off

5.1.6 Obtain video configuration (video parameters)

Function Description: Used to get the values of resolution, video frame rate, video bit rate, video flip and other parameters.

Request format: http://{IP}/system/video

Request method and URL: GET /system/video

Interface request body: none

Returns JSON data as follows:

```
{
    "rtsp_client_limit": 10,
    "rtsp_connected_num": 1,
    "rtsp_enabled": 1,
    "video_bit_rate": 4096,
    "video_flip_model": 0,
    "video_fps": 30,
    "video_height": 1080,
    "video_port": 443,
    "video_url": "live",
    "video_width": 1920，
    "msg": "SUCCESS",
    "ret": 200,

}
```

Description of interface return parameters

Parameter	Type	Required	Description
video_bit_rate	integer	Yes	video bitrate
video_fps	integer	Yes	video frame rate
video_height	integer	Yes	high resolution
video_width	integer	Yes	resolution width
video_flip_model	integer	Yes	Video Flip Mode
video_port	integer	Yes	rtsp service port

video_url	integer	Yes	rtsp service url
rtsp_connected_num	integer	Yes	Number of rtsp connections made
rtsp_client_limit	integer	Yes	rtsp client cap
rtsp_enabled	integer	Yes	rtsp service enable,0-off,1-on
ret	integer	Yes	
msg	String	Yes	

Note: The resolution takes a fixed value: 1920*1080、1280*720、640*480

5.1.7 Setting up video & RTSP configuration (video parameters)

Function Set the value of resolution, video frame rate, video bit rate, video flip and other parameters.

Request format: http://{IP}/system/video

Request method and URL: PUT /system/video

Interface request body: Yes

Headers:

Parameter	Value	Required
Content-Type	application/JSON	Yes

Example: When setting the resolution, you need to transmit both the resolution height and resolution width fields according to fixed values.

```
{
    "rtsp_client_limit": 6,
    "rtsp_connected_num": 1,
    "rtsp_enabled": 1,
    "video_bit_rate": 4096,
    "video_flip_model": 0,
    "video_fps": 30,
    "video_height": 1080,
    "video_port": 443,
    "video_url": "live",
    "video_width": 1920，

}
```

Description of interface request parameters

Parameter	Type	Required	Description

video_height	integer	No	high resolution
video_width	integer	No	resolution width
video_fps	integer	No	Video frame rate, supports 30
video_bit_rate	integer	No	Video bit rate, take a fixed value: 4096, 2048, 1024, 512.
video_flip_model	integer	No	Video flip mode, 0: default display, 1: vertical flip, 2: horizontal flip, 3: 180° flip
video_port	integer	No	rtsp service port
video_url	integer	No	rtsp service url
rtsp_client_limit	integer	No	rtsp client upper limit, integer range [1,6]
rtsp_enabled	integer	No	rtsp service enable,0-off,1-on

Note: The resolution takes a fixed value: 1920*1080、1280*720、640*480

5.1.8 Get Infrared Configuration (Infrared Video Parameters)

Function Description: Used to obtain the values of infrared module parameters, including emissivity, measurement distance, ambient temperature, relative humidity, temperature level, temperature unit, pseudo-color, highest temperature cursor, lowest temperature cursor, and center temperature cursor.

Request Format: http://{IP}/system/infrared

Request Method & URL: GET /system/infrared

Interface Request Body: None

Returned JSON Data:


```
{
    "ir_colorize": 5,
    "ir_cursor_high": 1,
    "ir_distance": 1.0,
    "ir_emiss": 0.95,
    "ir_envir": 4.0,
    "ir_humidity": 99.0,
    "ir_level": 0,
    "ir_low": 1,
    "ir_school": 1,
    "ir_unit": 0,
    "msg": "SUCCESS",
    "ret": 200
}
```

Description of interface request parameters

Parameter	Type	Required	Description
ir_emiss	number	Yes	Infrared module emissivity: 0.01-1
ir_distance	number	Yes	Measurement distance
ir_envir	number	Yes	Ambient temperature
ir_humidity	number	Yes	Relative humidity
ir_unit	integer	Yes	Temperature unit: 0 Celsius (℃), 1 Fahrenheit (°F), 2 Kelvin (K)
ir_level	integer	Yes	Temperature level
ir_colorize	integer	Yes	Pseudo-color
ir_cursor_high	integer	Yes	Highest temperature cursor: 0 Disable, 1 Enable
ir_school	integer	Yes	Center temperature cursor: 0 Disable, 1 Enable
ir_low	integer	Yes	Lowest temperature cursor: 0 Disable, 1 Enable
ret	integer	Yes	Return code
msg	String	Yes	Return message

5.1.9 Set Infrared Configuration (Infrared Video Parameters)

Description: Used to configure parameters of the infrared module, such as emissivity, distance, ambient temperature, relative humidity, temperature range (level), temperature unit, pseudo-color, and cursor settings (High/Low/Center temperature cursors).

Request Format: http://{IP}/system/infrared

Request Method & URL: PUT /system/infrared

Request Body: Required

Headers:

Parameter	Value	Required
Content-Type	application/JSON	Yes

Example: When setting specific parameters, certain fields must be transmitted together (e.g., when setting resolution, both width and height fields are required).

```
{
    "ir_colorize": 5,
    "ir_cursor_high": 1,
```

```
    "ir_distance": 1.0,
    "ir_emiss": 0.95,
    "ir_envir": 4.0,
    "ir_humidity": 99.0,
    "ir_level": 0,
    "ir_low": 1,
    "ir_school": 1,
    "ir_unit": 0,
    "msg": "SUCCESS",
    "ret": 200
}
```


Request Parameter Definitions

Parameter	Type	Required	Description
ir_emiss	number	Yes	Emissivity of the infrared module (Range: 0.01 - 1)
ir_distance	number	Yes	Temperature measurement distance
ir_envir	number	Yes	Ambient temperature
ir_humidity	number	Yes	Relative humidity
ir_unit	integer	Yes	Temperature Unit: 0: Celsius (°C), 1: Fahrenheit (°F), 2: Kelvin (K)
ir_level	integer	Yes	Temperature Level / Range
ir_colorize	integer	Yes	Pseudo-color (Palette selection)
ir_cursor_high	integer	Yes	Max Temperature Cursor: 0: Disable, 1: Enable
ir_school	integer	Yes	Center Temperature Cursor: 0: Disable, 1: Enable
ir_low	integer	Yes	Min Temperature Cursor: 0: Disable, 1: Enable
ret	integer	Yes	
msg	String	Yes	

5.1.10 Getting the mode of operation

Function Description: Used to get the current running mode

Request format: http://{IP}/system/detect_status

Request method and URL: GET /run_mode

Interface request body: none

Returns JSON data as follows:

```
{
  "run_mode": 1,
  "ret": 200,
  "msg": "SUCCESS",
}
```

Description of interface return parameters

Parameter	Type	Required	Description
run_mode	integer	Yes	Detection mode, 1: Single detection mode. -1: Intelligent cruise mode
ret	integer	Yes	return code
msg	String	Yes	Back to information

5.1.11 Setting the operating mode

Function Set the operation mode.

Request format: http://{IP}/system/detect_status

Request method and URL: PUT /run_mode

Interface request body: Yes

Headers:

Parameter	Value	Required
Content-Type	application/JSON	Yes

Request Example:

```
{
  "run_mode": 1,
}
```

Description of interface request parameters

Parameter	Type	Required	Description
run_mode	Integer	Yes	Detection mode, 1: Single detection mode. -1: Intelligent cruise mode

5.1.12 Getting configuration related to leakage estimation (leakage mode)

Function: Used to get the values of the parameters used in leakage estimation.

Request format: http://{IP}/system/detect_status

Request method and URL: GET /system/leakage

Interface request body: none

Returns JSON data as follows:


```
{
      "cost_display": 0,
      "leakage_distance": 0.5,
      "leakage_unit": 0,
      "leakage_unit_str": "l/min",
      "monetary_unit": 0,
      "monetary_unit_str": "CNY",
      "energy_cost": 0,
      "specific_power": 0,
      "value_of_gas":1，
      "working_hours": 1
     "ret": 200,
     "msg": "SUCCESS",

}
```


Description of interface return parameters

Parameter	Type	Required	Description
cost_display	integer	Yes	Whether or not to display the results of the leakage cost estimation on the web site, 0 - not to be displayed, 1 - to be displayed
leakage_distance	integer	Yes	Leakage estimation distance, range [1~50], unit meter
leakage_unit	integer	Yes	Leakage estimation and display unit, 0--l/min, 1--ml/s
leakage_unit_str	string	Yes	Leakage estimation and display unit, l/min or ml/s
monetary_unit	integer	Yes	Leakage and loss estimates are in 0 - ¥ (CNY) [Chinese Yuan], 1 - £ (GBP) [Pound Sterling], 2 - $ (USD) [United States Dollar], 3 - € (EUR) [Euro], 4 - ₽ (RUB) [Ruble].

monetary_unit_str	string	Yes	Leakage and loss estimates use strings of currency units, including the five mentioned above - CNY, GBP, USD, EUR, RUB
value_of_gas	float	Yes	Gas value in monetary units/m^3
energy_cost	float	Yes	Energy costs in monetary units/kWh
specific_power	float	Yes	Specific power in kW/(m^3/min)
working_hours	integer	Yes	Average annual working hours in h/year
ret	integer	Yes	return code
msg	String	Yes	Back to information

5.1.13 Setting leakage estimation related configuration (leakage mode)

Function Set the value of the parameter used in leakage estimation.

The parameter name is the same as above, and the corresponding field is sent separately if a parameter needs to be configured at the time of the request.

Request format: http://{IP}/system/detect_status

Request method and URL: PUT /system/leakage

Interface request body: Yes

Headers:

parameter name	parameter value	compulsory
Content-Type	application/JSON	Yes

Example:

```
{
"cost_display":0,
"energy_cost":1,
"leakage_unit":0,
"monetary_unit":0,
"specific_power":1,
"value_of_gas":1,
"working_hours":0,
"leakage_distance":0.5
}
```

Description of interface request parameters

Parameter	Type	Required	Description
cost_display	integer	No	Whether or not to display the results of the leakage cost estimation on the web site, 0 - not to be displayed, 1 - to be displayed
leakage_distance	integer	No	Leakage estimation distance, range [1~50], unit meter
leakage_unit	integer	No	Leakage estimation and display unit, 0--l/min, 1--ml/s
monetary_unit	integer	No	Leakage and loss estimates are in 0 - ¥ (CNY) [Chinese Yuan], 1 - £ (GBP) [Pound Sterling], 2 - $ (USD) [United States Dollar], 3 - € (EUR) [Euro], 4 - ₽ (RUB) [Ruble].
value_of_gas	float	No	Gas value in monetary units/m^3
energy_cost	float	No	Energy costs in monetary units/kWh
specific_power	float	No	Specific power in kW/(m^3/min)
working_hours	integer	No	Average annual working hours in h/year

5.1.14 Get detection area

Function Description: Used to get the parameter value of the detection area.

Request format: http://{IP}/system/detect_status

Request method and URL: GET /system/scan_area

Interface request body: none

The interface returns JSON data as follows:

```
{
    "areas": [
        {
            "area_height": 168,
            "area_width": 240,
            "detect_area_status": 1,
            "left_top_x": 12,
            "left_top_y": 12
        },
        {
```

```
                "area_height": 168,
                "area_width": 240,
                "detect_area_status": 1,
                "left_top_x": 257,
                "left_top_y": 185
            },
            {
                "area_height": 168,
                "area_width": 240,
                "detect_area_status": 1,
                "left_top_x": 502,
                "left_top_y": 358
            },
            {
                "area_height": 168,
                "area_width": 240,
                "detect_area_status": 1,
                "left_top_x": 747,
                "left_top_y": 531
            },
            {
                "area_height": 168,
                "area_width": 240,
                "detect_area_status": 1,
                "left_top_x": 992,
                "left_top_y": 704
            }
        ],
        "center_focus": 2,
        "msg": "SUCCESS",
        "ret": 200
    }
```


The interface returns a description of the parameters:

Parameter	Type	Required	Description
area_height	integer	Yes	rectangular height
area_width	integer	Yes	rectangular width
detect_area_status	integer	Yes	Detect area drawing status, 0 hide 1 show
left_top_x	integer	Yes	upper-left horizontal coordinate

left_top_y	integer	Yes	Upper left vertical coordinate
center_focus	integer	Yes	Drawing frame modes: 1: center focus mode, 2: frame selection mode

5.1.15Setting the detection area

Used to set the value of the detection area parameter, the parameter name is the same as above, all fields need to be sent when requesting.

Request format: http://{IP}/system/detect_status

Request method and URL: PUT /system/scan_area

Interface request body: Yes

Headers:

Parameter	Value	Required
Content-Type	application/JSON	Yes

Example of interface request parameters:

```
{
 "areas":[
{"area_height":432,
"area_width":768,
"detect_area_status":1,
"left_top_x":576,
"left_top_y":324}],
"center_focus":1,
}
```

Parameter	Type	Required	Description
area_height	integer	Yes	rectangular height
area_width	integer	Yes	rectangular width
detect_area_status	integer	Yes	Detect area drawing status, 0 hide 1 show
left_top_x	integer	Yes	upper-left horizontal coordinate
left_top_y	integer	Yes	Upper left vertical coordinate
center_focus	integer	Yes	Drawing frame modes: 1: center focus mode, 2: frame selection mode

5.1.16 Request to turn on detection

Function Description: Used to start the detection task

In single detection mode, after starting the detection only once, and return this detection results data id, after the end of the detection and, can be viewed in the data view page can be viewed; in the intelligent cruise mode, after starting the detection will be implemented after the opening of the alarm policy cycle detection, detection of anomalies before the detection results will be recorded.

Request format: http://{IP}/system/detect_status

Request method and URL: POST /turn_on_detect

Interface request body: none

Returns JSON data as follows:

```
{
  "detect_id": 1,
  "ret": 200,
  "msg": "SUCCESS",
}
```

Description of interface return parameters

Parameter	Type	Required	Description
detect_id	integer	No	Detection id (returned when single detection mode is on)
ret	integer	Yes	Return code -1: Indicates that no policy is currently available
msg	String	Yes	Back to information

5.1.17 End of request detection

Function Description: Used to end the detection task (currently all detection results are transmitted asynchronously via websocket, so the end of the detection request does not return content, see 5.2.3 for details)

Request format: http://{IP}/system/detect_status

Request method and URL: POST /turn_off_detect

Interface request body: none

5.1.18 Get the current detection status (whether or not it is being detected)

Function Description: Used to get the detection status

Request format: http://{IP}/system/detect_status

Request method and URL: GET /system/detect_status

Returns JSON data as follows:

```
{
    "detect_status": 0
}
```

The interface returns a description of the parameters:

Parameter	Type	Required	Description
detect_status	boolean	Yes	Whether detection is in progress 0: No detection is in progress 1: Detection is in progress

5.1.19 Getting Inspection Logs

Function Description: Used to get the results of the inspection task

Request format: http://{IP}/detect_logs/{detect_id}

Request method and URL: GET /detect_logs

Interface Request Body: The request parameters need to be put into the Query Param of the HTTP request, the relevant parameters are as follows.

Description of the interface request parameters:

Parameter	Required	default value	clarification
detect_id	No	1	Detection log id

The interface returns JSON data as follows:


```
{
 "bin_mic_data_name": "mic_data_2019_06_20_20_30_39_940.bin",
 "data_1": 0.0,
 "data_2": -1.0,
 "data_3": 0.0,
 "data_4": 0.0,
 "detect_mode": 2,
 "end_time": 1561033842,
 "files_status": 1,
 "id": 7,
 "power_spect_photo_name": "power_spectrum_2019_06_20_20_30_39_940.jpg",
 "power_spect_video_name": "power_spectrum_2019_06_20_20_30_39_940.mp4",
 "saving_path": "2019_06_20_20_30_39_940/",
 "soundmap_intensity": 20.4,
 "soundmap_photo_name": "sound_2019_06_20_20_30_39_940.jpg",
 "soundmap_video_name": "sound_2019_06_20_20_30_39_940.mp4",
 "start_time": 1561033839,
```

```
    "status": 1,
    "voiceprint_photo_name": "voiceprint_2019_06_20_20_30_39_940.jpg",
    "voiceprint_video_name": "voiceprint_2019_06_20_20_30_39_940.mp4",
    "wav_mic_data_name": "mic_data_2019_06_20_20_30_39_940.wav"
},
"msg": "SUCCESS",
"ret": 200
}
```


The interface returns a description of the parameters:

Parameter	Type	Description
detect_logs	array	Returns the log group, structured as in the example above.
id	int	Log Number
detect_mode	int	detection mode
start_time	string	Detection start time
end_time	string	Detection end time
status	int	detection state
soundmap_intensity	int	sound field strength
file_status	int	File Save Status: 0-Deleted 1-Saved
saving_path	string	Start Time Named Folder Records File Save Path
power_spect_video_name	string	Labeled Spectrogram Video File Name
power_spect_photo_name	string	Logo Spectrogram Image File Name
soundmap_video_name	string	Identifies the sound field imaging video file name
soundmap_photo_name	string	Identifies the sound field imaging image file name
voiceprint_video_name	string	Identifies the name of the voiceprint video file
voiceprint_photo_nam	string	Logo sound image file name

e		
bin_mic_data_name	string	Identifies the sound field audio file name, bin format
wav_mic_data_name	string	Identifies the sound field audio file name, wav format
data_1	number	Leak volume
data_2	number	leakage cost
data_3	number	Unit of leakage
data_4	number	Leakage cost unit
alarm_strategy_id	int	Strategy ID
alarm_level	int	Alarm level (same as policy level)
nc_file	string	Sample file, file name
last_alarm_db	number	dB alarm value
last_alarm_df	number	Master Frequency Alarm Value
last_alarm_nc	number	Sample file variance
is_alarm	number	Alarm or not 0: No 1: Yes
nc_file_path	string	Sample File Path
nc_file_fft_photo_name	string	Sample file fft mapping

5.1.20 Access to test result files

Function Description: Documentation used to obtain test results

Request format: http://{IP}/detect_file/{log.saving_path}/{logs.file_name}

Request method and URL: GET /detect_file/{log.saving_path}/{logs.file_name}

Interface request body: none

Example of interface request parameters (using the log results returned in Example 5.1.15 as an example):

http://192.168.2.204/detect_file/2023_04_07_14_31_29_133/sound_2023_04_07_14_31_29_133.jpg

Returns the corresponding video or image file.

5.1.21 Delete detection logs

Function Description: Used to delete the results of the detection task

Request format: http://{IP}/detect_logs

Request method and URL: DELETE/detect_logs

Interface request body: Yes

Headers:

Parameter	Value	Required
Content-Type	application/JSON	Yes

Example of interface request parameters:

```
{
    "id": [ 2 , 3 ]
}
```

Description of the interface request parameters:

Parameter	Type	Description	Required
Id	string	Array of ids of logs to be deleted	No

5.1.22 Get system time

Function Description: Used to get the current system time

Request format: http://{IP}/system_time

Request method and URL: GET /system_time

Interface request body: none

Returns JSON data as follows:

```
{
    "sys_time": 17220332
}
```

The interface returns a description of the parameters:

Parameter	Type	Required	Description
sys_time	number	Yes	system timestamp

5.1.23 Setting the system time

Function Description: Used to set the system time

Request format: http://{IP}/time_set_manually

Request method and URL: PUT /time_set_manually

Interface request body: Yes

Headers:

Parameter	Value	Required
Content-Type	application/JSON	Yes

Example of interface request parameters:

```
{
    "time": 12345678,
}
```

Request parameter description:

Parameter	Type	Required	Description
time	number	Yes	system timestamp

5.1.24 Get Infrared Camera Configuration

Description: Used to retrieve the configuration parameters of the infrared camera.

Request Format: http://{IP}/system/infrared

Request Method & URL: GET /system/infrared

Request Body: None

Response JSON Example:

```
{
    "ir_colorize": 4,
    "ir_cursor_high": 1,
    "ir_distance": 1.03,
    "ir_emiss": 0.96,
    "ir_envir": 26,
    "ir_humidity": 20,
    "ir_level": 0,
    "ir_low": 1,
    "ir_school": 1,
    "ir_unit": 0,
    "msg": "SUCCESS",
    "ret": 200
}
```

The interface returns a description of the parameters:

Parameter	Type	Required	Description
ir_emiss	number	Yes	Emissivity of the infrared module (Range: 0.01 - 1)
ir_distance	number	Yes	Temperature measurement distance (Range: 0.3 - 130)
ir_envir	number	Yes	Ambient temperature
ir_humidity	number	Yes	Relative humidity (Range: 0 - 100)

ir_unit	integer	Yes	Temperature Unit: 0: Celsius (°C) 1: Fahrenheit (°F) 2: Kelvin (K)
ir_level	integer	Yes	Temperature Level / Range: 0: Low Range, 1: High Range (Celsius: [-20°C ~ 150°C], [100°C ~ 550°C] Fahrenheit: [-4°F ~ 302°F], [212°F ~ 1022°F] Kelvin: [253K ~ 423K], [373K ~ 823K])
ir_colorize	integer	Yes	Pseudo-color / Palette: 0: White Hot, 1: Black Hot, 2: Fusion 1, 3: Rainbow, 4: Fusion 2, 5: Iron Red 1, 6: Iron Red 2, 7: Sepia, 8: Color 1, 9: Color 2, 10: Ice and Fire, 11: Rain, 12: Red Hot, 13: Green Hot, 14: Dark Blue
ir_cursor_high	integer	Yes	Max Temperature Cursor: 0: Disable, 1: Enable
ir_school	integer	Yes	Center Temperature Cursor: 0: Disable, 1: Enable
ir_low	integer	Yes	Min Temperature Cursor: 0: Disable, 1: Enable

5.1.25 Set Infrared Camera Configuration Parameters

Description: Used to set the configuration parameters of the infrared camera.

Request format: http://{IP}/system/infrared

Request method and URL: PUT /system/infrared

Request Body: Required

Headers：

Parameter	Value	Required
Content-Type	application/JSON	Yes

Request Payload Example:

```
{
    "ir_colorize": 4,
    "ir_cursor_high": 1,
    "ir_distance": 130,
```

```
        "ir_emiss": 0.96,
        "ir_envir": 26,
        "ir_humidity": 20,
        "ir_level": 0,
        "ir_low": 1,
        "ir_school": 1,
        "ir_unit": 0,
    }
```


Request Parameter Definitions:

Parameter	Type	Required	Description
ir_emiss	number	Yes	Emissivity of the infrared module (Range: 0.01 - 1)
ir_distance	number	Yes	Temperature measurement distance (Range: 0.3 - 130)
ir_envir	number	Yes	Ambient temperature
ir_humidity	number	Yes	Relative humidity (Range: 0 - 100)
ir_unit	integer	Yes	Temperature Unit 0: Celsius (°C) 1: Fahrenheit (°F) 2: Kelvin (K)
ir_level	integer	Yes	Temperature Level / Range: 0: Low Level, 1: High Level • Celsius: [-20°C ~ 150°C], [100°C ~ 550°C] • Fahrenheit: [-4°F ~ 302°F], [212°F ~ 1022°F] (Negative signs from original text corrected) • Kelvin: [253K ~ 423K], [373K ~ 823K]
ir_colorize	integer	Yes	Pseudo-color / Palette: (0: White Hot, 1: Black Hot, 2: Fusion 1, 3: Rainbow, 4: Fusion 2, 5: Iron Red 1, 6: Iron Red 2, 7: Sepia, 8: Color 1, 9: Color 2, 10: Ice and Fire, 11: Rain, 12: Red Hot, 13:

			Green Hot, 14: Dark Blue)
ir_cursor_high	integer	Yes	Max Temperature Cursor: 0: Disable, 1: Enable
ir_school	integer	Yes	Center Temperature Cursor: 0: Disable, 1: Enable
ir_low	integer	Yes	Min Temperature Cursor: 0: Disable, 1: Enable
Parameter	Type	Required	Description
time	number	Yes	System Timestamp

5.2 PTZ (Pan-Tilt-Zoom) Control Interfaces

5.2.1 Universal Command Settings

Description: Universal PTZ control commands used to set the PTZ home (zero) position, pan speed, and tilt speed.

Request Format: http://{IP}/ptz_universal_cmd

Request Method & URL: PUT /ptz_universal_cmd

Request Body: Required

Headers：

Parameter	Value	Required
Content-Type	application/JSON	Yes

Request Payload Example:

```
{
    "cmd_fun": 1,
   "dev_speed": 1,

}
```

Request Parameter Definitions:

Parameter	Type	Required	Description
cmd_fun	integer	Yes	Function Code: (0: Return to Home position, 1: Pan Speed)
dev_speed	integer	Optional	Optional Parameter (Range: 0 - 63). Note: Required when cmd_fun = 1

Response Parameter Definitions:：

Parameter	Type	Required	Description
ret	number	Yes	Return Code
msg	string	Yes	Return Description

5.2.2 Get PTZ Angle Information

Description: Used to retrieve the PTZ angle information.

Request Format: http://{IP}/ptz_read_angle

Request Method & URL: GET /ptz_read_angle

Request Body: Required

Headers：

Parameter	Type	Required
Content-Type	application/JSON	Yes

Response Parameter Definitions:

Parameter	Type	Required	Description
status_code	integer	Yes	Device Status Code 0: idle, 1: busy, 2: err
err_code_1	integer	Yes	Error Code bit0: Pan Motor Stall bit1: Tilt Motor Stall bit2: Configuration Busy
version	string	Yes	Version Information
azimuth_angle	integer	Yes	Azimuth Angle (-180 ~ 180)
pitch_angle	integer	Yes	Pitch Angle (-30 ~ 30) (Reserved field)
h_speed	integer	Yes	Horizontal (Pan) Rotation Speed
v_speed	integer	Yes	Vertical (Tilt) Rotation Speed (Reserved field)
mode	integer	Yes	Operation Mode 0x00: Low Power Mode 0x01: Set Target Mode

			0x02: Horizontal Left Scan Mode 0x03: Horizontal Right Scan Mode 0x04: Vertical Up Scan Mode 0x05: Vertical Down Scan Mode
ret	number	Yes	Return CodeReturn Code
msg	string	Yes	Return Description

5.2.3 Set PTZ Angle

Description: Used to retrieve the PTZ angle information.

Request Format: http://{ip}/ptz_set_angle

Request Method & URL: PUT /ptz_set_angle

Request Body: Required

Headers：

Parameter	Type	Required
Content-Type	application/JSON	Yes

Request Payload Example:

```
{
   "azimuth_angle": 120,
  "pitch_angle": 1,
}
```

Request Parameter Definitions:

Parameter	Type	Required	Description
azimuth_angle	integer	Yes	Azimuth Angle (-180 ~ 180)
pitch_angle	integer	Yes	Pitch Angle (-30 ~ 30) (Reserved field)

Response Parameter Definitions:

Parameter	Type	Required	Description
ret	number	Yes	Return Code
msg	string	Yes	Return Description

5.2.4 Continuous Rotation Interface

Description: Used to control the continuous rotation of the PTZ camera.

Request Format: http://{IP}/ptz_continuous_rotation

Request Method & URL: PUT /ptz_continuous_rotation

Request Body: Required

Headers：

Parameter	Type	Required
Content-Type	application/JSON	Yes

Request Payload Example:

```
{
  "direction_rotation": 0,
}
```

Request Parameter Definitions:

Parameter	Type	Required	Description
direction_rotation	integer	Yes	Rotation Direction (0: Left, 1: Right)

Response Parameter Definitions:

Parameter	Type	Required	Description
ret	number	Yes	Return Code
msg	string	Yes	Return Description

5.3 Websocket Interfaces

To ensure real-time data transmission, the WebSocket protocol is used to retrieve device parameters via port 80. Once the connection is established, device parameters can be acquired according to the custom communication protocol.

Request Format: ws://{IP}:port

Heartbeat Mechanism: To guarantee a correct and secure WebSocket connection, a heartbeat mode is implemented. After connecting to the device's WebSocket server, all normally operating clients must send an arbitrary message (content unrestricted) to the server within every minute to ensure the connection is not interrupted. Otherwise, the server will determine that the client is not in use and forcibly terminate the connection.

5.3.1 Get FFT Data

Request URL: ws://{IP}:port/fftData

The device returns FFT data approximately five times per second (the first half of the FFT, length

512). The return format is a 16-bit int binary stream. The returned data should be parsed according to the byte order.

5.3.2 Get Voiceprint Image

Request URL: ws://{IP}:port/voiceprintData

The device returns one frame of voiceprint image data approximately per second. The return format is an int binary stream. The returned data should be parsed according to the byte order.

5.3.3 Get Detection Results

Request URL: ws://{IP}:port/detectFeedback

After each detection, the device returns a package of detection result data. The return format is a JSON string, which should be parsed according to the byte order.

Response JSON Example:


```
  {
"detect_log": {
   "bin_mic_data_name": "mic_data_2019_06_20_20_30_39_940.bin",
   "data_1": 0.0,
   "data_2": -1.0,
   "data_3": 0.0,
   "data_4": 0.0,
   "detect_mode": 2,
   "end_time": 1561033842,
   "files_status": 1,
   "id": 7,
   "power_spect_photo_name": "power_spectrum_2019_06_20_20_30_39_940.jpg",
   "power_spect_video_name": "power_spectrum_2019_06_20_20_30_39_940.mp4",
   "saving_path": "2019_06_20_20_30_39_940/",
   "soundmap_intensity": 20.4,
   "soundmap_photo_name": "sound_2019_06_20_20_30_39_940.jpg",
   "soundmap_video_name": "sound_2019_06_20_20_30_39_940.mp4",
   "start_time": 1561033839,
   "status": 1,
   "voiceprint_photo_name": "voiceprint_2019_06_20_20_30_39_940.jpg",
   "voiceprint_video_name": "voiceprint_2019_06_20_20_30_39_940.mp4",
   "wav_mic_data_name": "mic_data_2019_06_20_20_30_39_940.wav"
},
"msg": "SUCCESS",
```

```
"ret": 200
}
```


Response Parameter Definitions:

Parameter	Type	Description
detect_log	array	Returned log group; structure matches the example above.
id	int	Log ID
detect_mode	int	Detection Mode
start_time	string	Detection Start Time
end_time	string	Detection End Time
status	int	Detection Status
soundmap_intensity	int	Sound Field Intensity
file_status	int	File Saving Status:0: Deleted, 1: Saving
saving_path	string	Folder named by start time, recording the file save path
power_spect_video_name	string	Identifies the power spectrum video name
power_spect_photo_name	string	Identifies the power spectrum photo name
soundmap_video_name	string	Identifies the sound field imaging video name
soundmap_photo_name	string	Identifies the sound field imaging photo name
voiceprint_video_name	string	Identifies the voiceprint video name
voiceprint_photo_name	string	Identifies the voiceprint photo name
bin_mic_data_name	string	Identifies the sound field audio file (.bin format)
wav_mic_data_name	string	Identifies the sound field audio file (.wav format)
data_1	number	Leakage Amount
data_2	number	Leakage Cost
data_3	number	Leakage Amount Unit
data_4	number	Leakage Cost Unit

5.3.4 Get Anomalous Sound Point Information

Request URL: ws://{IP}:port/energyData

The device periodically returns energy value data of anomalous sound points. The return format is a 32-bit int binary stream. The returned data should be parsed according to the byte order.

Multiple sets of target point information are transmitted based on the currently set number of targets. The target point information is arranged in sequence. For example, if the number of targets is 3, the information is arranged as follows: [[Target 1 Info], [Target 2 Info], [Target 3 Info]]. The data arrangement within each set is: [Energy, Pixel X-coordinate, Pixel Y-coordinate, Grazing Angle, Azimuth Angle].

5.3.5 Get PCM Data

Request URL: ws://{IP}:port/pcmData

The device periodically returns PCM data with a sampling rate of 48000 Hz. The return format is a 32-bit float binary stream. The returned data should be parsed according to the byte order.

5.3.6 Get Temperature Data

Request URL: ws://{IP}:port/inrared_data

The device returns real-time temperature data. The return format is a 32-bit float binary stream. The returned data should be parsed according to the byte order. Data structure: { emissivity: Emissivity, fAvrTmp: Center Temp, fMaxTmp: Max Temp, fMinTmp: Min Temp }.

Note: Temperature information requires enabling the corresponding Max Temp, Min Temp, and Center Temp cursors in "5.2.25 Set Infrared Camera".

5.3.7 Get Imaging Point Temperature Information

Request URL: ws://{IP}:port/pointtempData

The device returns real-time temperature information for imaging points on the fusion screen. The return format is a 32-bit float binary stream. The returned data should be parsed according to the byte order. Data structure: { point_x: X Coordinate, point_y: Y Coordinate, point_temp: Temperature Value }.

5.4 RTSP Interfaces

5.4.1 Get Live Video Stream

Visible Light Video Stream: Request URL: rtsp://{IP}:443/live

By accessing rtsp://<Device IP Address>:443/live via the RTSP protocol, you can directly obtain the device's real-time visible light video stream.

Thermal Imaging Video Stream: Request URL: rtsp://{IP}:443/livei

By accessing rtsp://<Device IP Address>:443/livei via the RTSP protocol, you can directly obtain the device's real-time thermal imaging video stream.

Fusion Video Stream: Request URL: rtsp://{IP}:443/lfxs/fusion

By accessing rtsp://<Device IP Address>:443/lfxs/fusion via the RTSP protocol, you can directly obtain the device's real-time fusion video stream.