
H2 红外热成像仪 操作手册

目录

1、	产品外观	3
2、	产品规格	5
3、	快速入门指南	7
4、	附件清单	7
5、	热像仪部件	8
6、	机械安装	9
7、	接口插针配置	11
7.1、	M12 X 型连接器插针定义	11
7.2、	M12 A 型连接器插针定义	12
8、	热像仪 客户端 界面	13
8.1	支持的系统	13
8.2	PC端IP修改	13
8.2.1	登录界面	13
8.2.2	登录凭证	14
8.3	热成像仪功能	14
9、	用户二次开发接口	46

1、 产品外观





2、产品规格

红外参数	H2
探测器	256*192
热灵敏度/NETD	<60mk
帧频	25fps
视场角/FOV	42.0° × 32.1°
镜头	6.8 mm
调焦	固定
调色板	至少 6 种调色板 (黑热/白热/铁红/等)
可见光参数	
分辨率	1920x1080
视场角/FOV(D)	69°
镜头	5.75 mm
调焦	固定
补光	内置 LED 灯
多光谱融合	
可见光融合	双波段图像融合增强
测温	
测温范围	-20°C to + 120 °C (支持宽动态-20 °C to +400°C)
测温精度	±2°C 或量程的 2% (宽动态状态下量程的 2%)
点测温	6 个可移动点测温, 可与其他测温点或测温区域的最高温度、最低温度、平均温度做比较
区域测温	6 个可移动区域测温, 区域内有最高温度、最低温度和平均温度, 可以与其他测温点或测温区域的最高温度、最低温度、平均温度做比较
线测温	1 条线测温, 线上有最高温度、最低温度和平均温度
测量预设值	有
温差	个温度测量值与参考温度之间的温度差
参考湿度	手动设置
大气传递校正	自动, 基于距离、大气温度及相对湿度的输入值
发射率校正	0.01 到 1.0 不等
反射表观温度校正	自动, 基于反射温度输入值
测温校正	总体及单个目标参数
报警	
报警功能	所有测温点、所有测温区域和测温线内的最高温度、最低温度和平均温度均可配置单独的报警输出
报警输出	I/O 输出、日志、保存图像、文件发送 (FTP.) 电子邮件 (SMTP.) 通知
协议	
ONVIF	支持
GB/28181	支持

图像流	
图像流格式	MJPEG/H.264
图像流分辨率	可见光 1920x1080 热成像 480x360(upscale)/240x180 融合图像 480x360(upscale)/240x180
自动图像调节	支持
手动图像调节	支持
图像存储	
存储介质	内置存储器 (Flash) 可选配内置 TF 卡, 最大支持 64GB
图像存储模式	红外/可见光图像单独存储 红外和可见光图像同时存储 可见光和红外融合图像存储
文件格式	JPEG、MP4
设置	
B/S 配置界面	有
C/S 配置工具	有
SDK	对外输出原始温度数据, 用户可进行二次开发
电源	
外部电源	DC12V
POE	802.3 at
外部接口	
M12 X 型	包含 10M / 100M 自适应以太网口和 POE 供电
M12 A 型	包含 DC12、报警 IO 和模拟视频
环境参数	
防护等级	IP67
工作温度	-20°C to +60°C
存储温度	-40°C to +70°C
湿度	≤95% 非冷凝
电磁兼容性	EN 61000-6-2:2001(抗干扰) EN 61000-6-3:2001(抗辐射) FCC 47 CFR 15 部分 B 级(抗辐射)
抗冲击	25G, IEC68-2-29
抗振性	2G, IEC68-2-6
物理参数	
尺寸	55.5 x30.4 x 88.5 mm (不带连接器) 55.5 x30.4 x102.4 mm (带连接器)
配件	
附件 (非标配)	接口线缆 支架 散热板

3、 快速入门指南

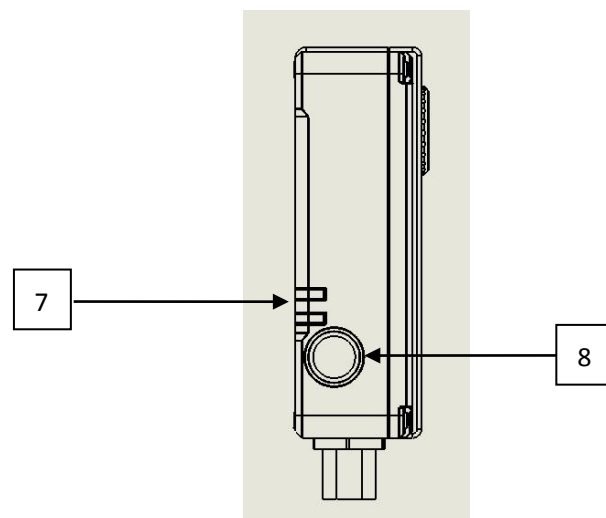
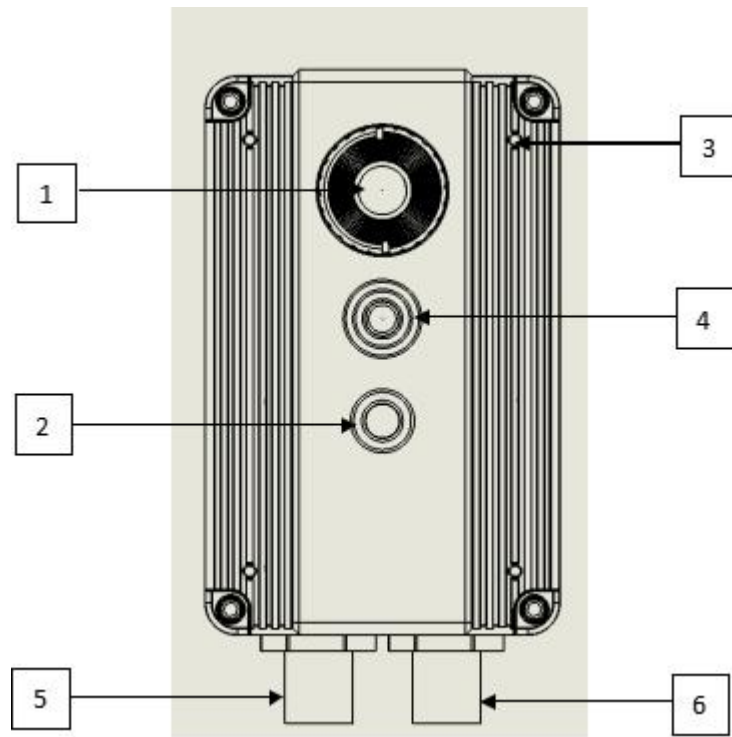
请遵循以下的安装步骤：

- 安装热像仪。
- 连接热像仪所需线缆。
- 热像仪出厂默认 IP 地址为 192.168.100.2，请通过 PC 机配置同网段 IP 地址后使用浏览器访问地址 <http://192.168.100.2:8080>，默认用户名 admin，默认密码 admin。
- 设置热像仪的 IP 地址。
- 接入网络中。
- 通过 InfiView 检索设备或直接通过软件设置和控制热像仪。

4、 附件清单

- H2 红外热像仪
- 连接线
- 网线
- 一次性运输箱
- 电子版说明书

5、热像仪部件

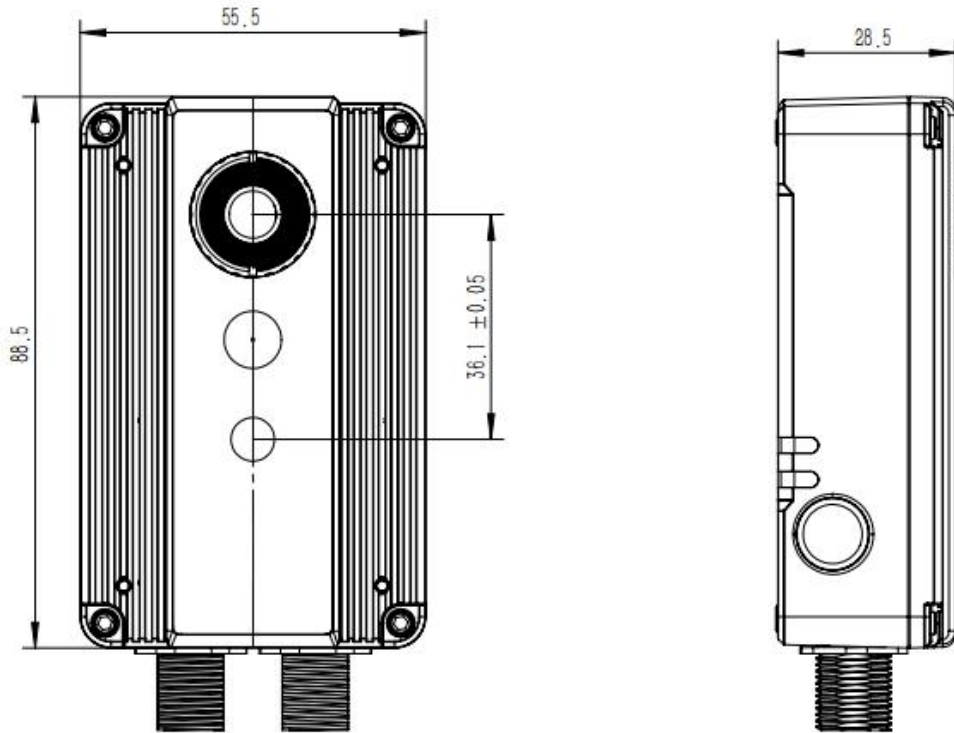


1	热成像镜头
2	可见光镜头
3	前置安装孔
4	LED 补光灯

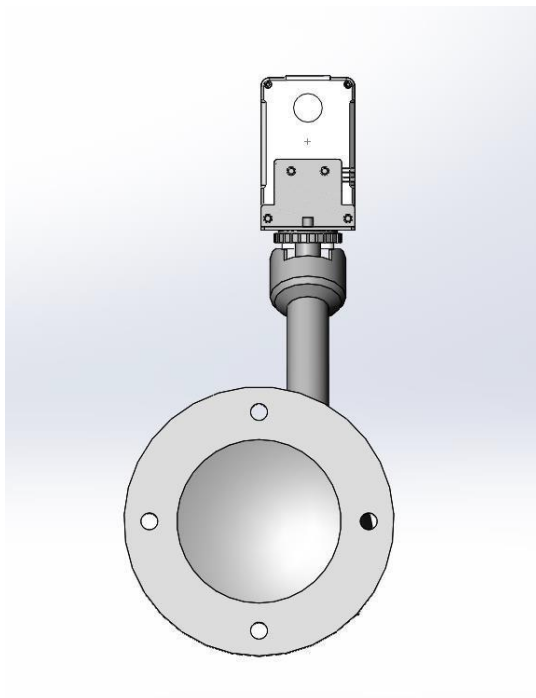
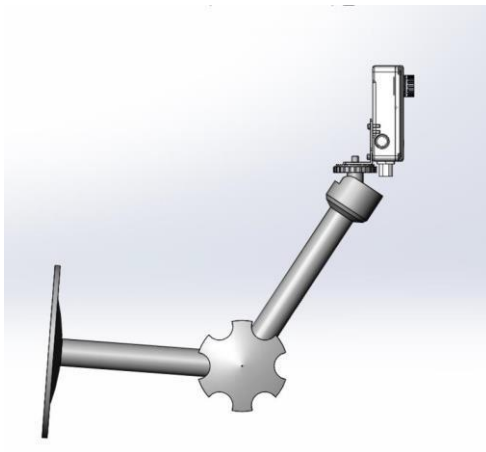
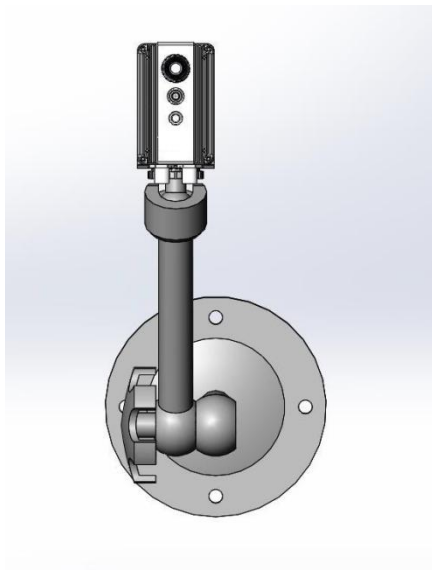
5	M12 X 型连接器
6	M12 A 型连接器
7	指示灯
8	复位按键

6、机械安装

本热像仪装置适合以任意姿态安装。热像仪的前后均设有安装接口。提供钻孔模板。



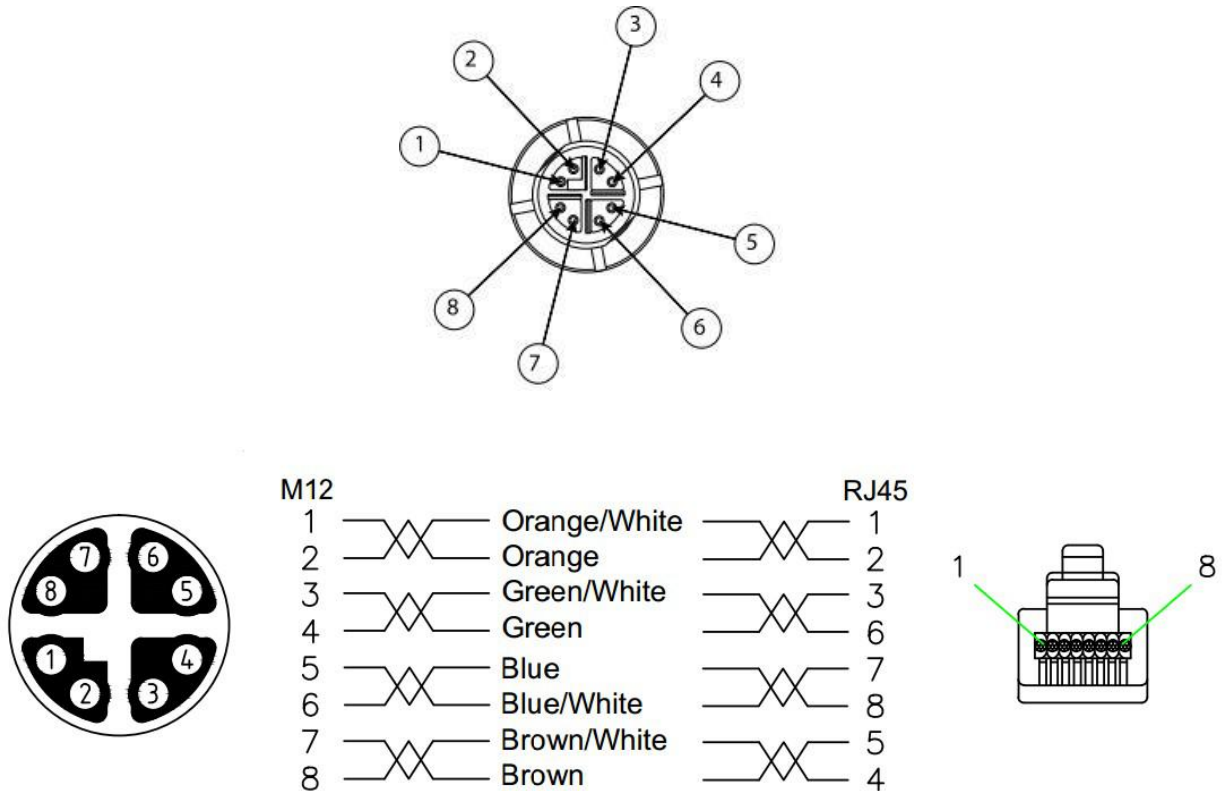
或者通过支架进行安装（支架和连接板可以自行采购，下图中支架仅为安装方式演示）



7、接口插针配置

产品对外提供两个接口，分别是 **M12 X** 型接口和 **M12 A** 型接口。

7.1、 M12 X 型连接器插针定义

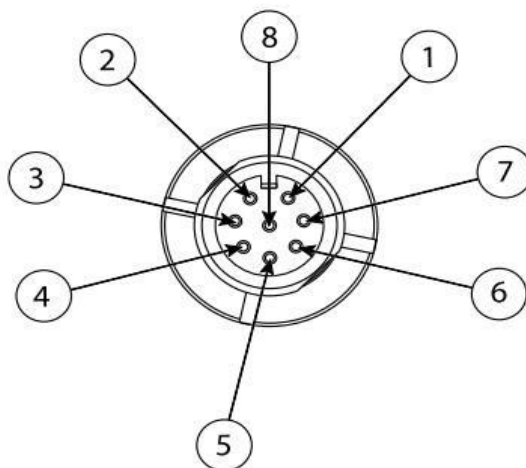


M12 X	网络名称
1	MAC_TX_D1+ (以太网网发送正)
2	MAC_TX_D1- (以太网网发送负)
3	MAC_RX_D2+ (以太网网接收正)
6	PR45

7	PR45
8	MAC_RX_D2- (以太网网接收正)
5	PR78
4	PR78

注意：1,2；3,6；4,5；7,8 需要双绞后连接

7.2、 M12 A 型连接器插针定义



M12 A	网络名称	释义
1	Adapter_Power(外部电源)	12~24V，可扩展至更高电压
8	Adapter_GND(外部电源地)	电源地
6	CVBS_OUT(模拟视频)	模拟视频
7	CVBS_GND(模拟视频地)	模拟视频地
5	Warning_In(警告输入)	已光耦隔离

2	Warning_Out(警告输出)	已光耦隔离
3	空	
4	空	

8、热像仪 客户端 界面

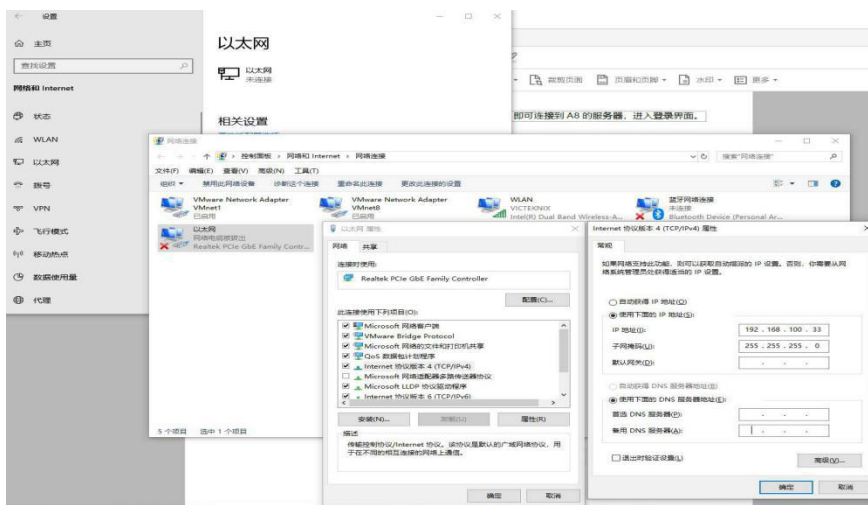
8.1 支持的操作系统

热像仪 InfiView客户端 界面可以支持

Windows7、Windows8.1、Windows10等操作系统

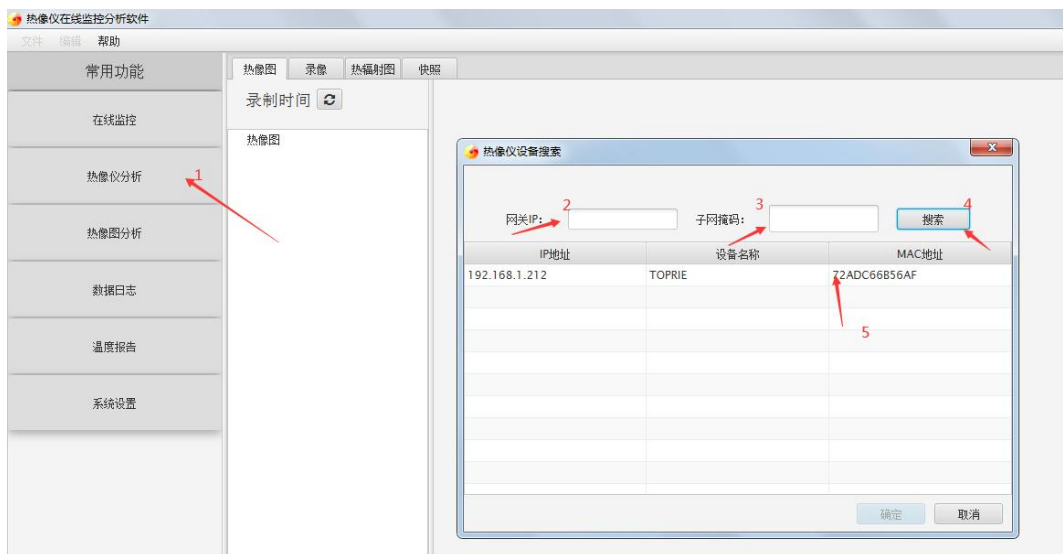
8.2 PC端IP修改

以win10操作系统为例，如图所示：



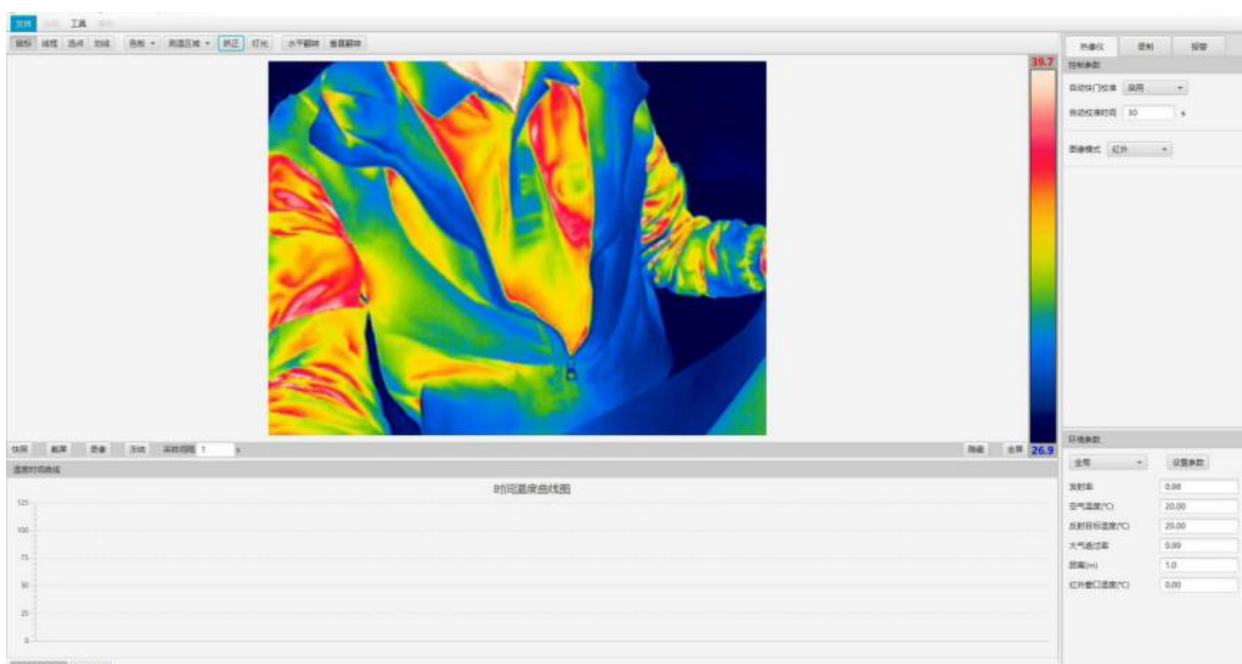
8.2.1 登录界面

将PC端的IP地址修改成和热像仪同网段的地址后，双击InfiView桌面图标会进入以下登录界面。



8.3 热成像仪功能



热成像仪具有实时监控视频流、执行温度测量、温度报警、拍摄快照、校准热像仪、管理图像与视屏等功能。初次登录时，热成像仪的参数均为出厂默认值，用户可以通过手动修改相关参数，实现对热成像仪某功能的控制使用，如下图所示。



8.3.1.1 概述

热成像仪可以同时捕捉红外和可见光图像。通过选择图像模式，你可以选择在网页上显示哪种类型的图像。

热成像仪支持以下图像模式。

图像模式	图像
红外： 显示整个红外图像	 An infrared thermal image of a ceiling grid. The image uses a color scale where warmer areas are shown in yellow and orange, and cooler areas are in purple and blue. The grid lines are clearly visible, and there are some bright spots indicating heat sources.
可见光： 显示数码摄像头拍摄的可见光图像	 A standard visible light photograph of the same ceiling grid. The image is in grayscale, showing the grid pattern and a door handle in the bottom right corner. The lighting is even, and the details of the grid are clearly visible.

8.3.2 补偿灯

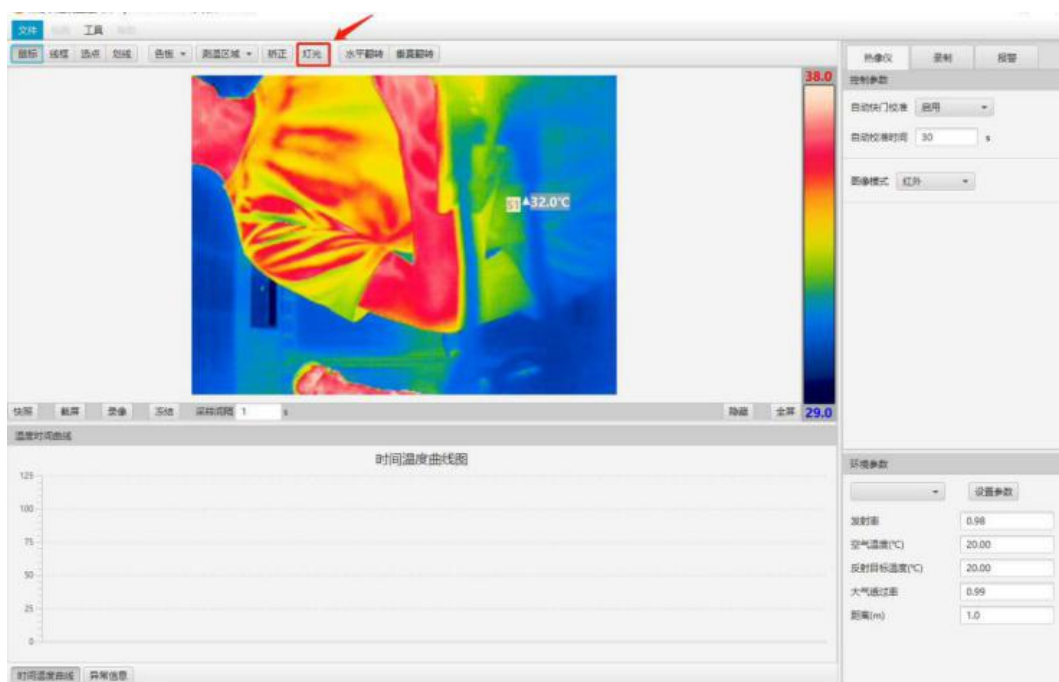
8.3.2.1 概述

热像仪配置了一个白炽灯，可用作灯光补偿。

8.3.2.2 使用

打开和关闭补偿灯，可按以下步骤操作。

在上方的工具栏中，单击图标“灯光”，热像仪就会打开灯光补偿，再单击一次就会熄灭灯光，如下图所示。

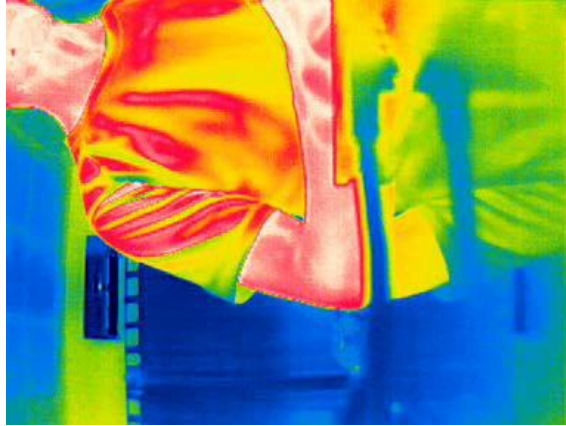


8.3.3 图像翻转

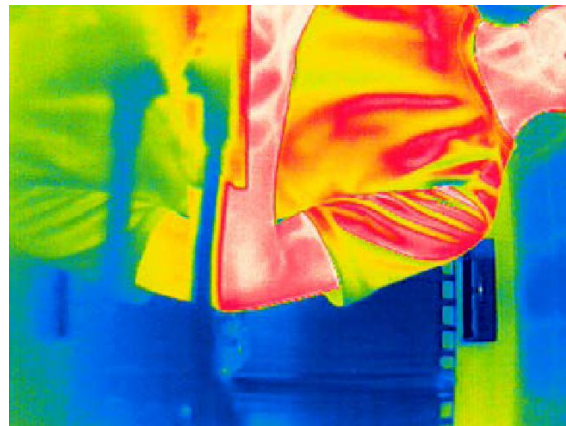
8.3.1.1 概述

图像可以水平或垂直翻转，如下图所示。

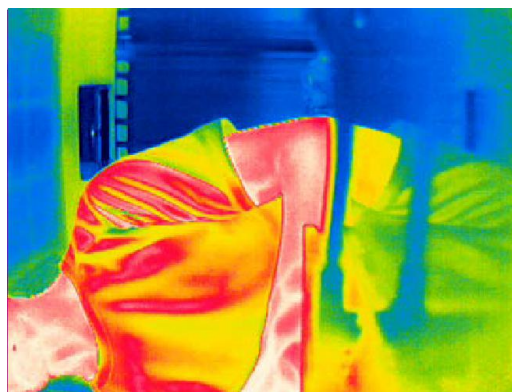
1. 普通图像



2. 水平翻转

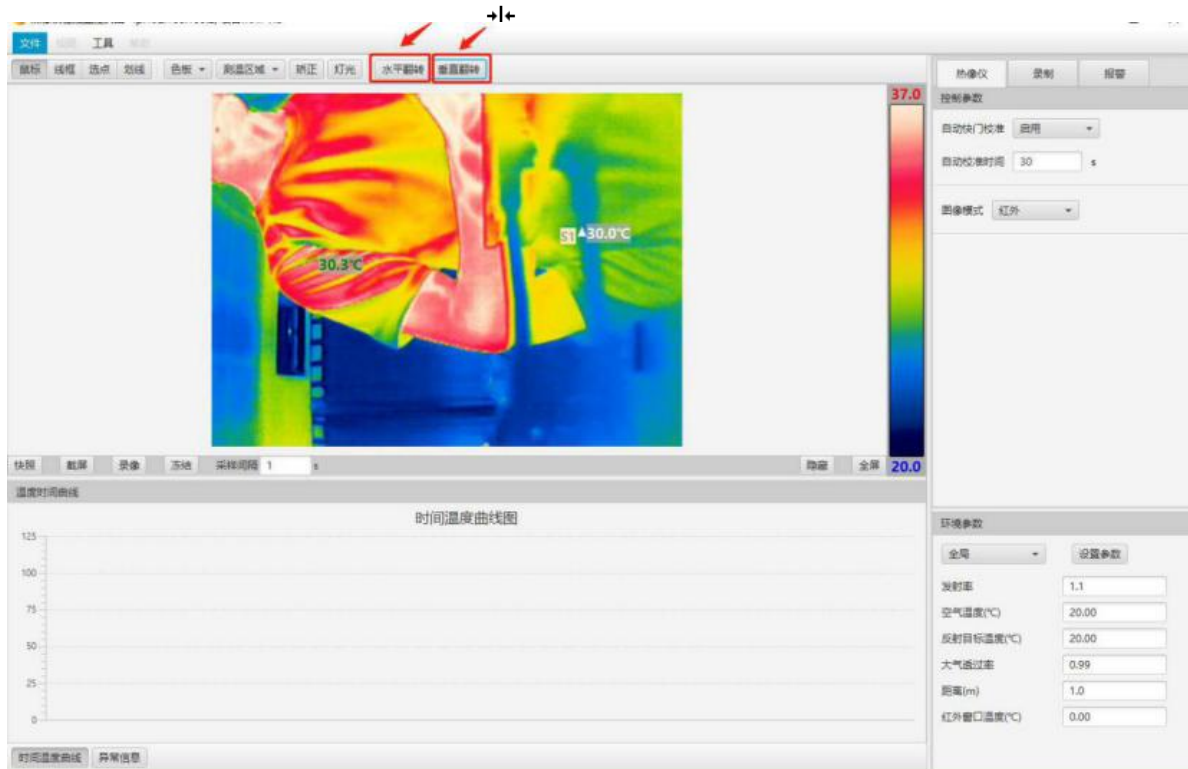


3. 垂直翻转



8.3.3.2. 使

在上方的工具栏中，单击“水平翻转”、“垂直翻转”如下图所示：



选择其中一个复选框：

- 水平方向：选中后水平方向翻转
- 垂直方向：选中后垂直方向翻转

注意：不支持水平方向和垂直方向同时翻转。

8.3.4 图像着色

8.3.4.1. 概述

热成像仪可以用不同方法为图像着色。您可以选择不同的调色板。

8.3.4.2. 调色板

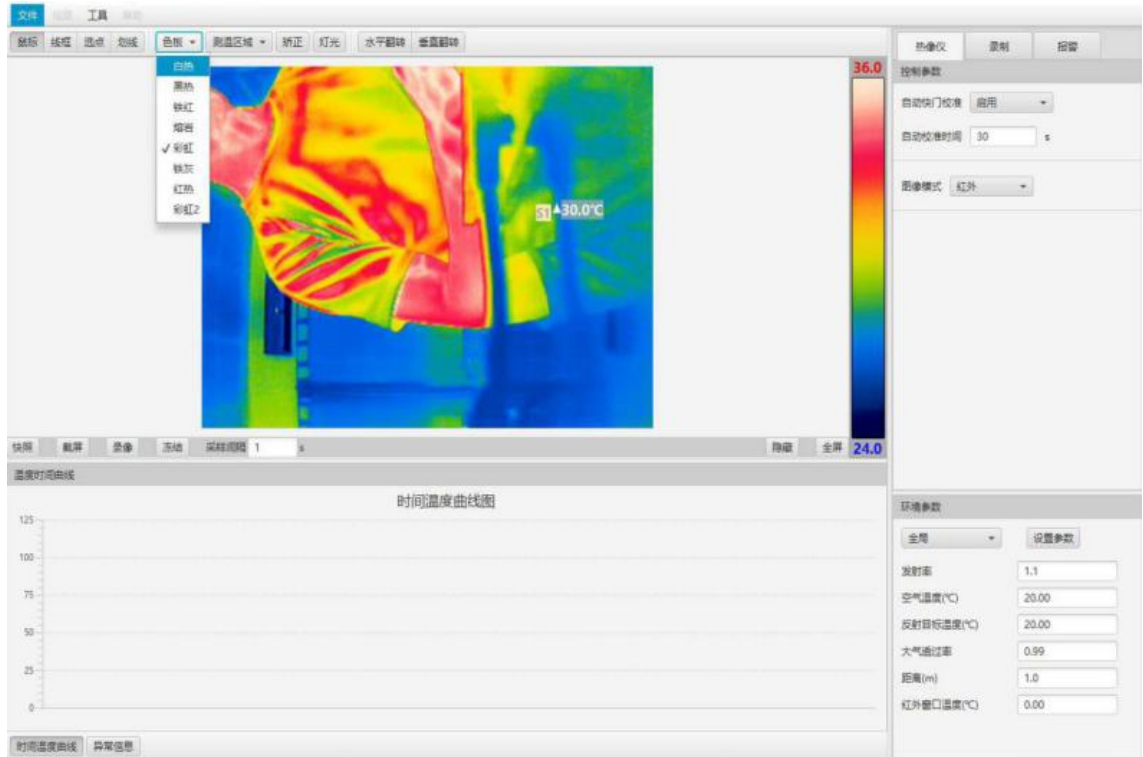
通过选择不同的调色板，可以让图像在不同环境下更容易被分析。而特定环境下最适合的调色板，取决于很多因素，比如目标温度、环境温度、热像仪与对象之间的距离等。

热成像仪支持以下 8 种调色板：

- 白热
- 黑热
- 铁红
- 紫红黄
- 蓝绿红
- 黑红
- 彩虹 1
- 彩虹 2

如要更改调色板，可按以下步骤进行：

在上方的工具栏中，单击图标 “色板”，会弹出一个带有下拉框的对话框，如下图所示。



选中下拉框中的选项即可更换调色板。

8.3.5 校准

8.3.5.1. 概述

热成像仪的校准采用非均匀性校正 (NUC) 非均匀性校正通过热成像仪软件执行的图像校正，可针对探测元件的不同灵敏度以及其他光学和几何干扰进行补偿。

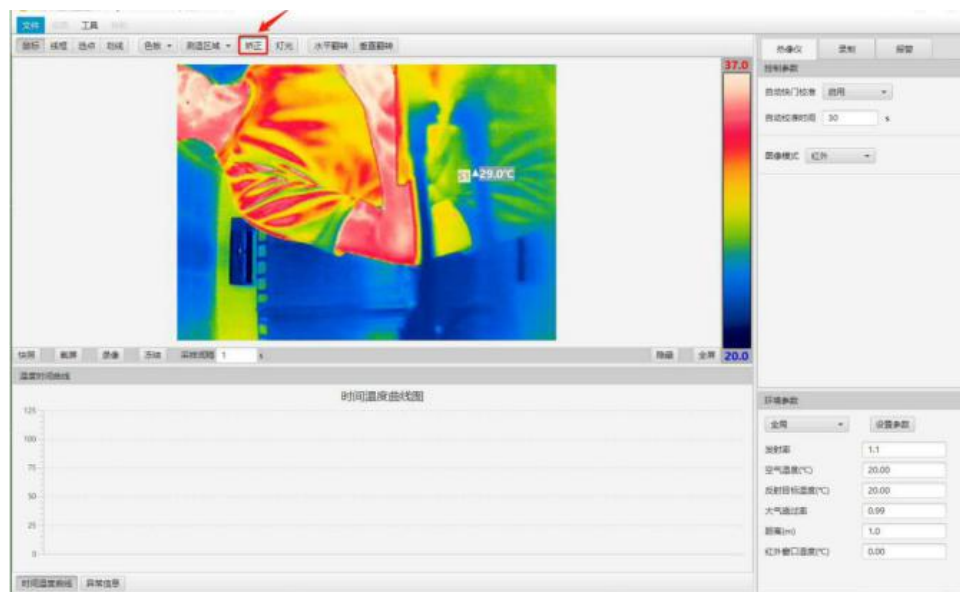
每当输出的图像出现空间干扰时，就应该进行校准。例如温度变化时可能造成这种干扰。

可以选择手动校准，也可以选择自动校准。

8.3.5.2. 手动校准

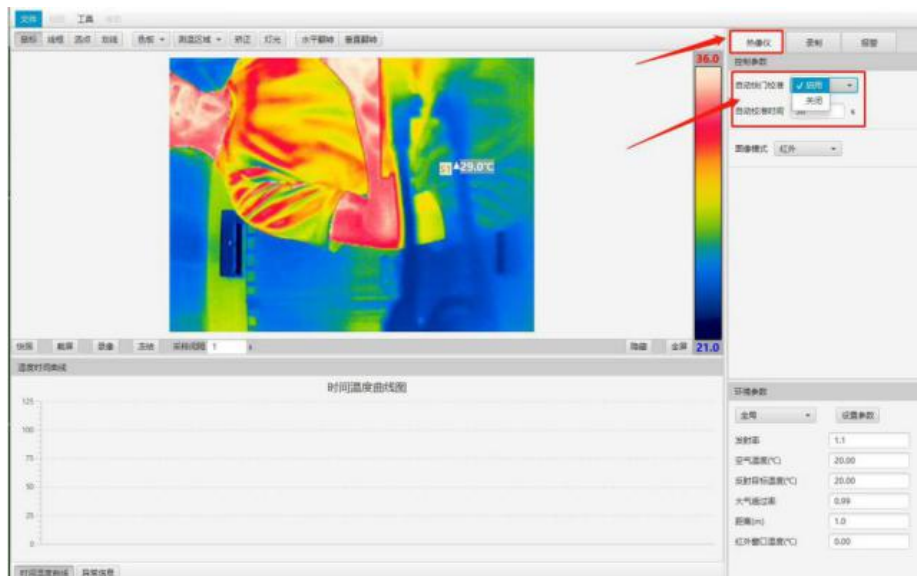
当图像出现干扰，影响分析时，可进行手段校准，执行步骤如下：在

上方的工具栏中，单击“矫正”，即可进行校准，如下图所示。



8.3.5.3. 自动校准

在热像图右侧工具栏中，单击“热像仪”“自动快门校准”，会弹出带有“启用”、“关闭”的选项，如下图所示。



- 自动校准允许：选中自动校准允许，则使能自动校准功能
- 自动校准时间：输入热成像仪自动校准的间隔时间，单位秒(s)

注意：自动校准时间最小值为 30s。

8.3.6 全局环境参数

8.3.6.1. 概述

要实现准确测量，需为热成像仪提供测量环境的一些参数：

- 反射率：是指一个物体发出的辐射量，是与理论参考物体（黑体）在相同温度下的辐射量相比较得出。与辐射率相对的是反射率。辐射率决定了从物体发出的辐射量，而不是反射的辐射量。
- 发射目标温度：用于补偿由物体反射至热成像仪的环境辐射。物体的这种属性称为反射率。
- 大气透过率：电磁波在大气中传播时，经大气衰减后的电磁辐射通量与入射时电磁辐射通量的比值。

- 空气温度：热成像仪与目标对象之间的空气温度。
- 距离：热成像仪对目标对象之间的距离。
- 红外窗口：热成像仪与目标对象之间设置了任何窗口等选项时使用该功能。
- 红外窗口温度：外部红外窗口的温度。

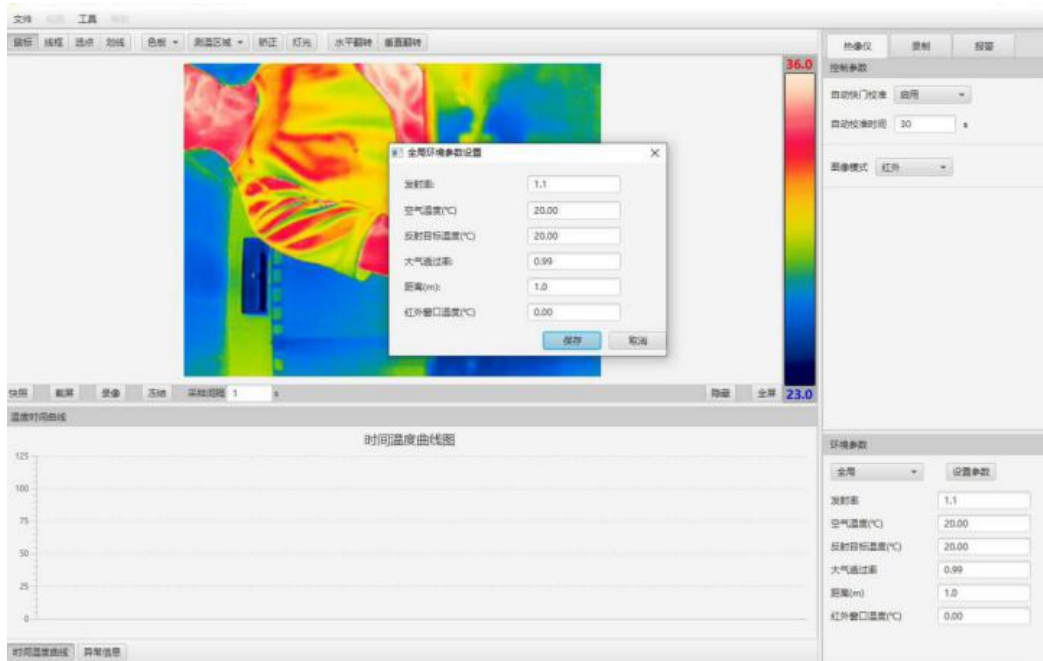
全局环境参数是为热成像仪的成像窗口的环境参数，局部环境参数可在相应测量范围内设置，具体请参阅。

注意：这些环境参数之中，辐射率和反射温度是热成像仪需要的两个最重要参数。

8.3.6.2. 设置全局环境参数

若修改全局环境参数，可按以下步骤操作：

在右下方的工具栏中，单击下拉按钮，选择全局或者其他点线框，然后单击“设置参数”，会弹出一个带有相关参数输入框的对话框，如下图所示。



在各个参数的输入框输入相应数值，点击保存按钮保存即可。

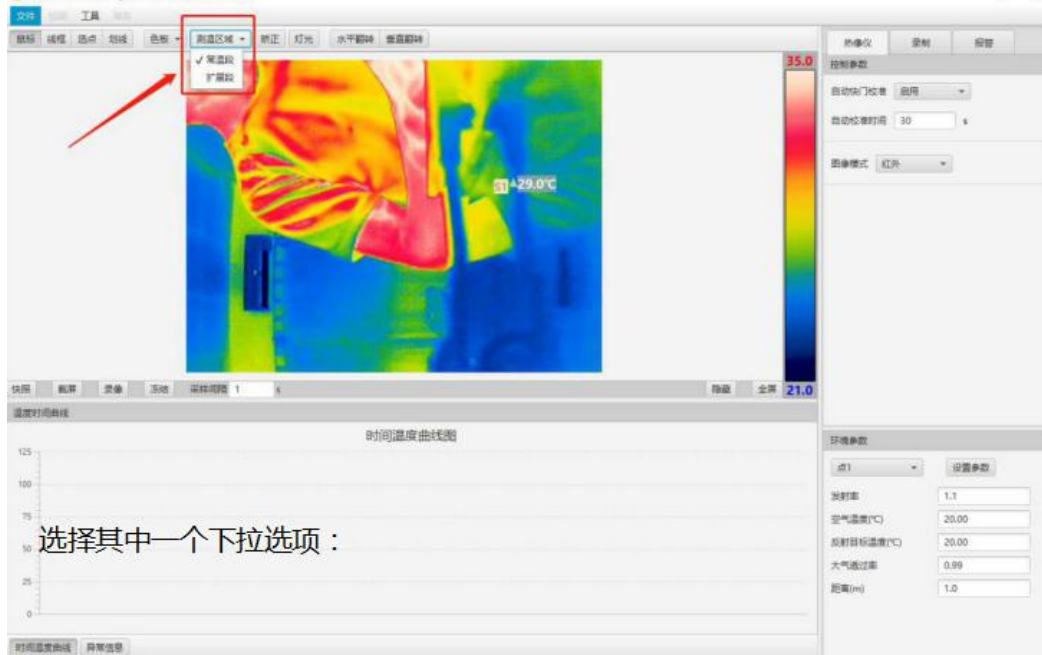
8.3.7 测温区段

8.3.7.1 概述

设置热成像仪的测量范围。

8.3.7.2 选择测温段

在上图的工具栏中，单击图标 “测温区域”，会弹出一个带有下拉框的对话框，如下图所示。



选择其中一个下拉选项：

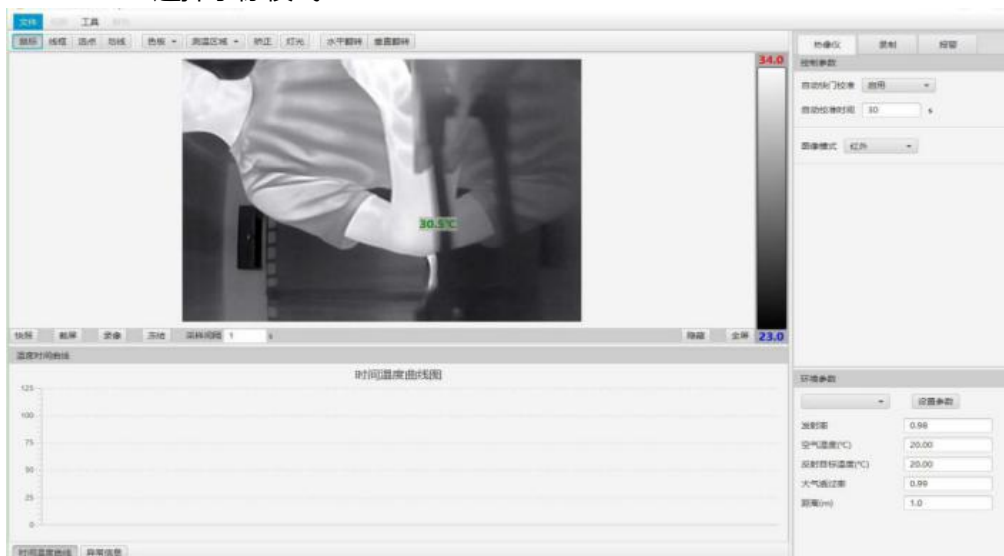
选择其中一个下拉选项：

- 常温段：测量范围在-20°C~120°C，精度±3°C。
- 扩展段：测量范围在-20°C~400°C，精度±2%。

8.3.7.3 温度跟踪

在温度跟踪下拉框中选择允许，则当鼠标滑动到视频上时，会实时显示当前位置的温度，如下图所示。

8.3.9.2 选择录像模式

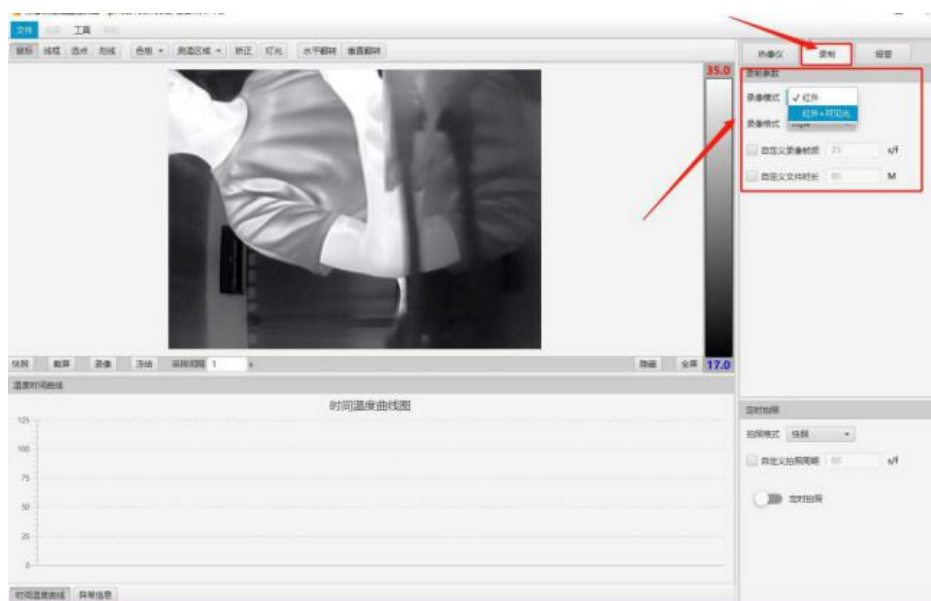


8.3.9 视频录像

8.3.9.1 概述

提供视频录像功能。

在热像图右侧的工具栏中，单击“录像”，会弹出一个带有下拉框的对话框，如下图所示。



选中其中一个下拉选项：

- 红外：只录像红外视频。
- 红外+可见光：同时录制红外和可见光视频。

注意：视频开始录制后，如果时长超过 5 分钟，会以 5 分钟为一段，分成数个小视频。

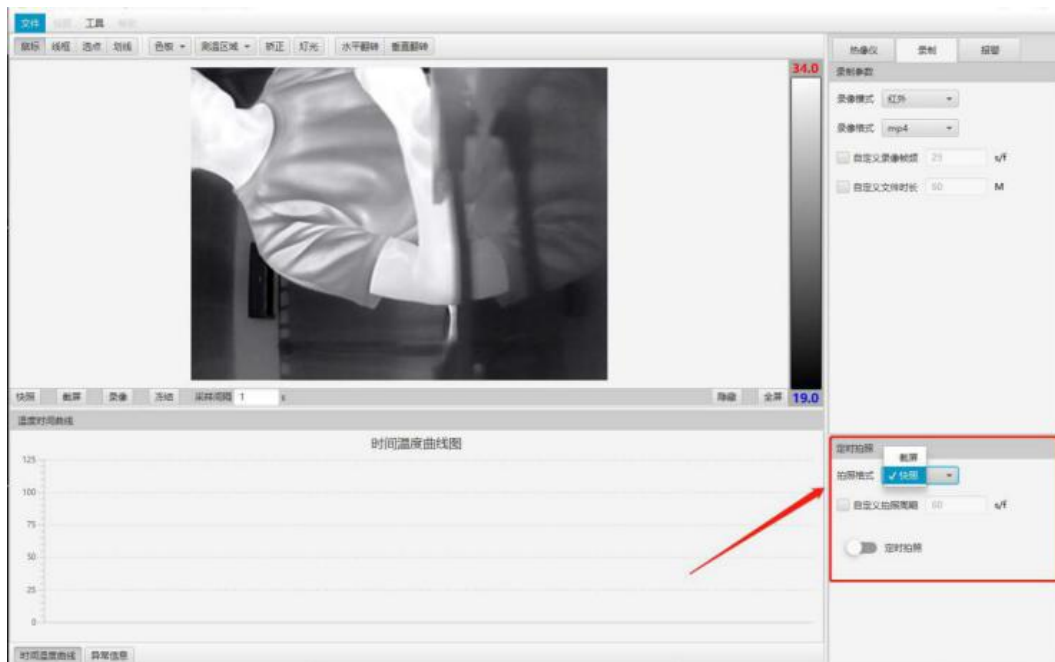
8.3.10 定时拍照

8.3.10.1 概述

可以通过该功能设定相隔时间段进行拍照。

8.3.10.2 更改定时周期

在右下方的工具栏中，单击“拍照格式”，会弹出一个带有单选框的对话框，如下图所示。



可以选择“截图”、“或”快照”两种方式，自定义拍照的周期。

8.3.11 测量工具

8.3.11.1 概述

本产品提供多种测量工具用来测量温度，比如测温框、测温点、测温线。

测温工具都有单独标号。

8.3.11.2 框测量工具

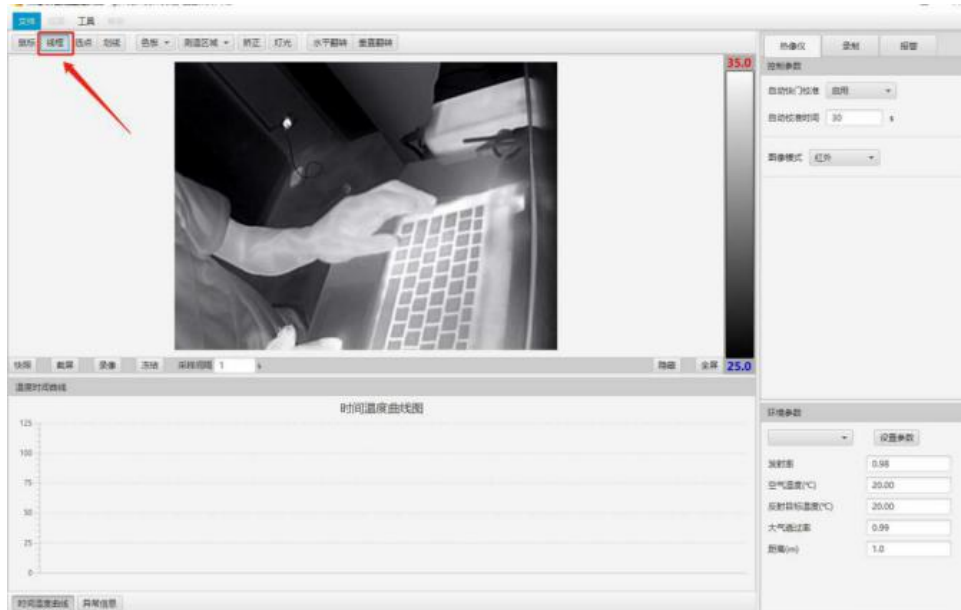
框测量工具可以显示测量框区域内的最高温度、最低温度、平均温度。共有 6 个测量框可以

使用。

要使用框测量工具，可按以下步骤操作：

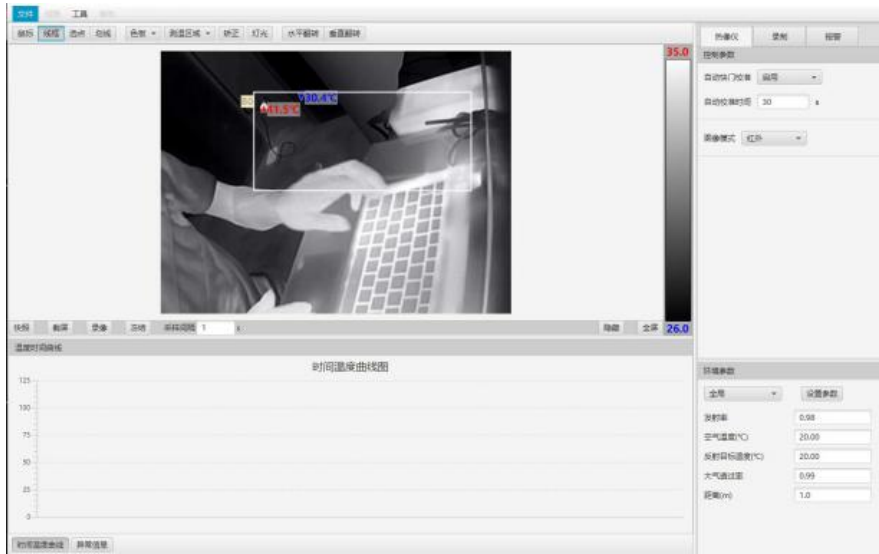
1. 选择框测量工具

在上方的工具栏中，点击"线框"选项，如下图所示：



2. 绘制测量框

选择好框测温工具后，将鼠标点在左侧图像上要测量的位置，然后按住鼠标左键不放，拖拽鼠标，可改变测量框的大小。测量框大小确定后，松开鼠标左键，测量框即绘制完成，如下图所示。



3. 拖动测量框

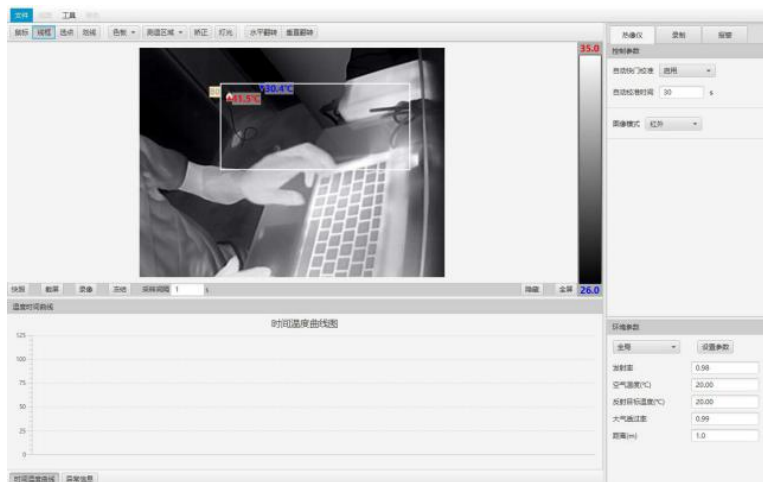
如果测量框的位置与待测量区域不一致，可以点击上方工具栏的“鼠标”选项，将鼠标移动到测量框上，按下鼠标左键不放，拖动鼠标，使测量框移动到合适的位置。

4. 测量框内容

测量框中，有红色、蓝色的两个点，红色点代表测量框内温度最高点位置，蓝色点代表测量框内温度最低点位置。

5. 温度显示

绘制测量框后，测量框内自动捕捉最高温，最低温，如下图所示。



6. 报警设置

当某标号测温框被绘制后，该框测量工具的报警功能即可被使用，具体步骤参考 8.3.12 章节内容。

7. 环境参数设置

若想对测量框内的温度精确测量，可设置该测量框的环境参数，具体步骤参考 8.3.13 章节内容。

8. 删除测量框

要删除测量框，鼠标放在相应的框里点击右键就可以弹出“删除”选项。

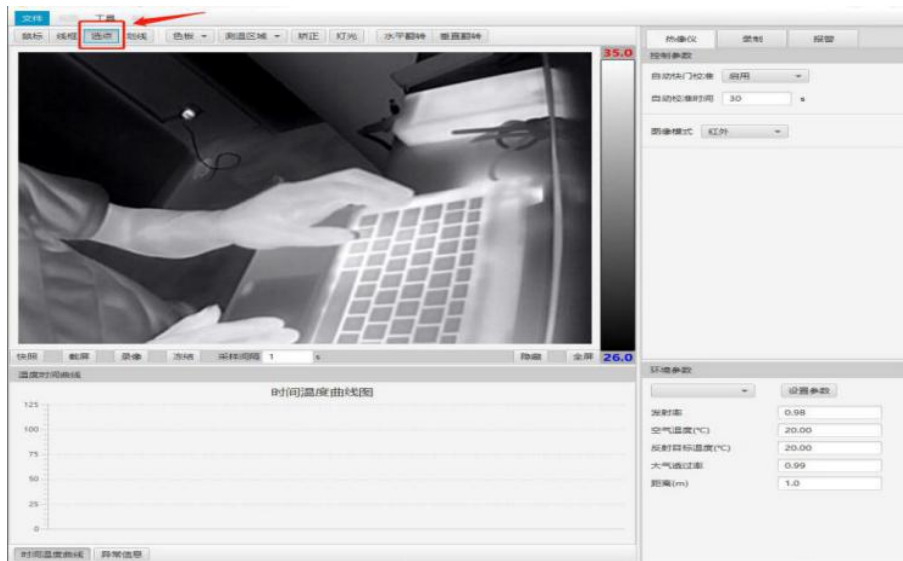
8.3.11.3 点测量工具

若只想测量图像上某一位置点的温度，可使用点测量工具测量。

共有 6 个点测量工具可以使用，可按以下步骤操作：

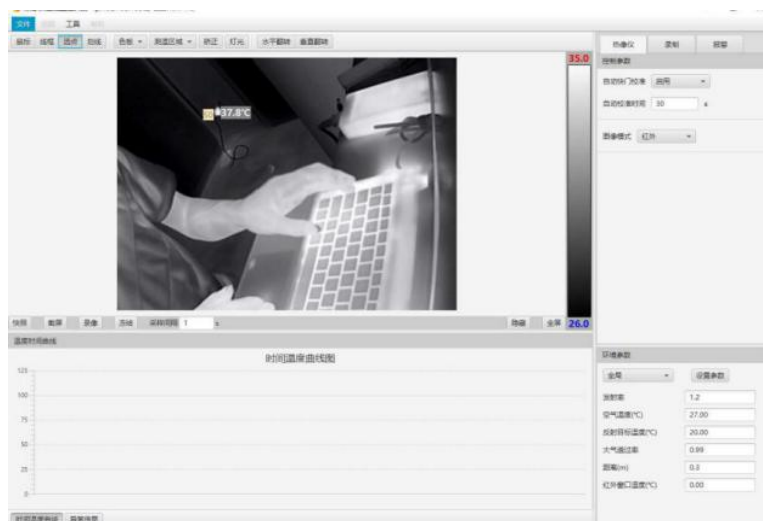
1. 选择点测量工具

在上方工具栏中，选择“选点”选项，如下图所示：



2. 绘制测量点

选择好测温工具后，将鼠标点在左侧图像上要测量的位置，然后按住鼠标左键点击一下即可，如下图所示。



3. 拖动测量点

如果测量点的位置与待测量位置不一致，选择上方工具栏的“鼠标”选项，将鼠标移动到测量点上，按下鼠标左键不放，拖动鼠标，使测量点移动到合适的位置。

注意：测量框的优先级高于测量点，无论绘制，还是拖动，若测量点是在测量框范围之内动作，则需先移动测量框，然后对测量点动作。

4. 温度显示

绘制测量点后，测量点位置上的温度值会在热像图上对应的点中显示。

5. 报警设置

当某标号测温点被绘制后，该点测量工具的报警功能即可被使用，具体步骤参考 8.3.12 章节内容。

6. 环境参数设置

若想对测量点位置上的温度精确测量，可设置该测量点的环境参数，具体步骤参考 8.3.13 章节内容。

7. 删除测量点

要删除点，鼠标选择相对应的“点”点击右键就可以弹出“删除”选项。

8.3.11.4 线测量工具

若想测量某条直线上的温度信息，可用线测量工具测量。

共有 1 个线测量工具可以使用，可按以下步骤操作：

1. 选择线测量工具并且划线

在上方的工具栏中，选择“划线”选项，会出现如下图所示：

选择好线测温工具后，将鼠标点在左侧图像上要测量的位置，然后按住鼠标左键不放，拖拽鼠标，可改变测量线的长短和方向。测量线的长短和方向确定后，松开鼠标左键，测量线即绘制完成，如下图所示。



3. 拖动测量线

如果测量线的位置与待测量区域不一致,可以将鼠标移动到测量线上,按下鼠标左键不放,拖动鼠标,使测量线移动到合适的位置。

内动作,则需先移动测量框,然后对测量线动作。

4. 测量线内容

测量线上,有红色、蓝色的两个点,红色点代表测量线上温度最高点位置,蓝色点代表测量线上温度最低点位置。

5. 温度显示

绘制测量线后,测量线上会显示出的最高温,最低温的具体数值。

6. 报警设置

当测温线被绘制后，该线测量工具的报警功能即可被使用，具体步骤参考 8.3.12 章节内容。

7. 环境参数设置

若想对测量线上的温度精确测量，可设置该测量线的环境参数，具体步骤参考 8.3.13 节内容。

8. 删除测量线

要删除线，鼠标选择相对应的“线”点击右键就可以弹出“删除”选项。

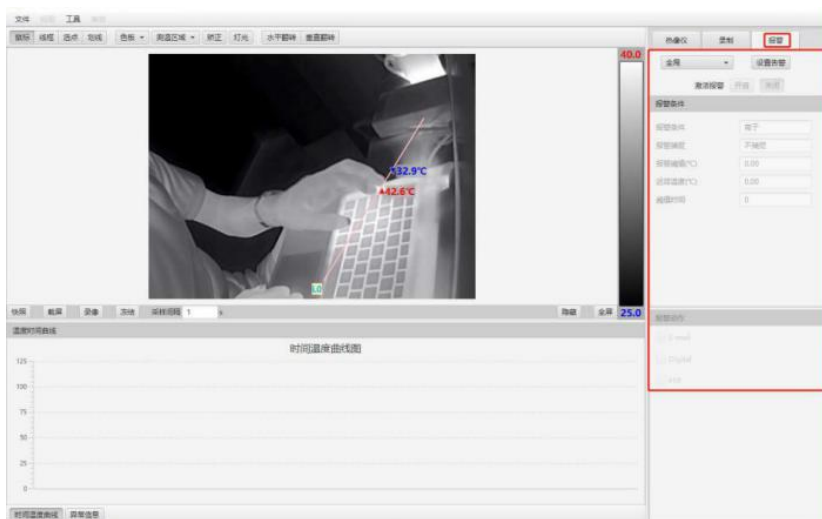
8.3.12 报警设置

8.3.12.1 概述

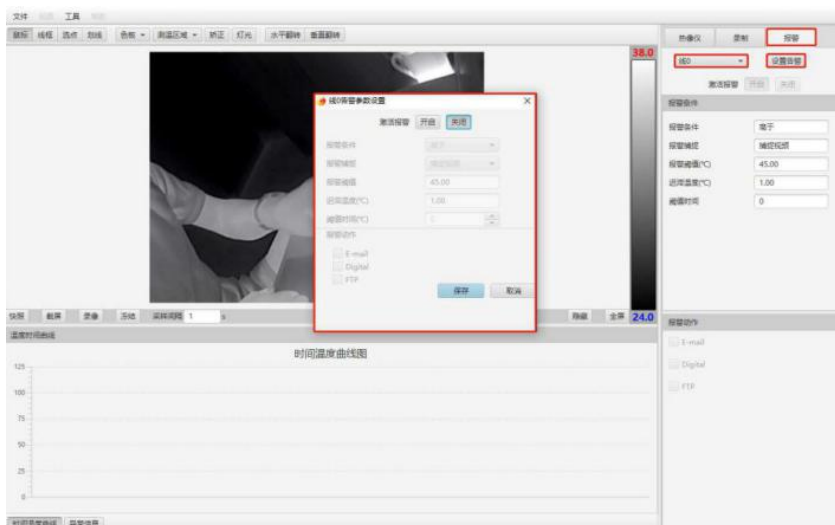
当测温框、点、线被绘制后，即可使用相应的报警功能。

当有报警被触发时，热成像仪会执行一项或多项动作，比如将图像或视频保存至内存、将图像或视频通过电子邮件发送给设定好的收件人、将图像发送到 FTP 站点

当报警为不可使用状态时，右侧的参数为灰色，如下图所示。



点击报警设置会弹出如下图所示的对话框：



8.3.12.2 报警设置

设置报警各参数值，可按以下步骤进行：

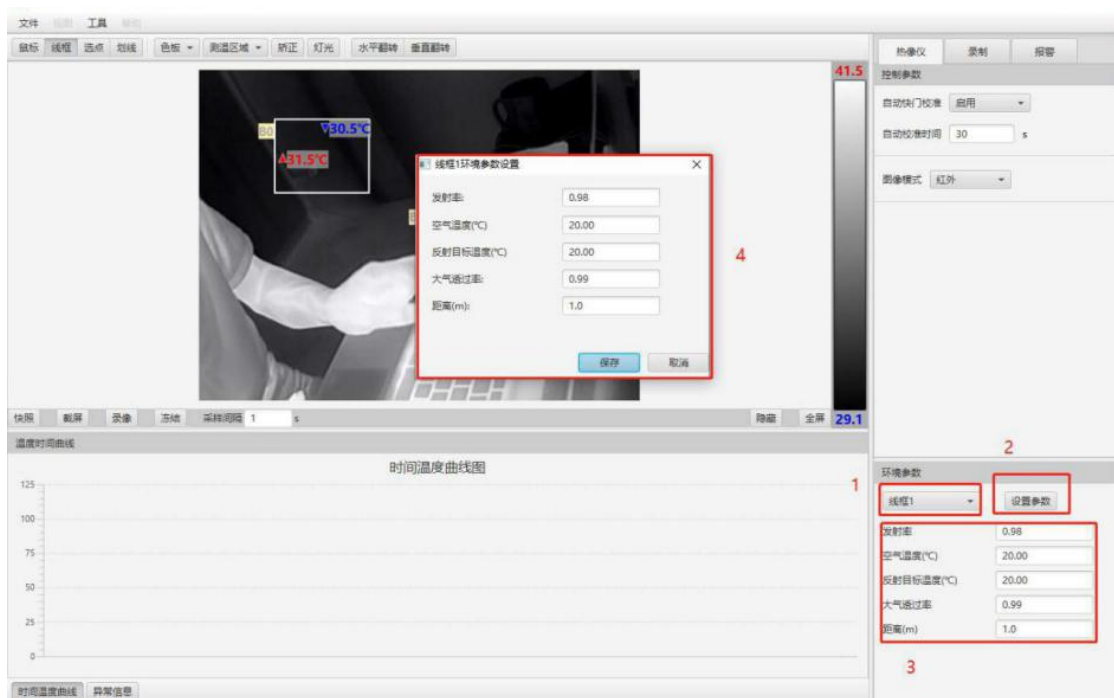
1. 激活报警：选中“关闭”，即不激活报警，选中“开启”，即激活报警；
2. 报警条件：选中“低于”，即测量区域有温度低于阈值时报警，选中“高于”，即测量区域有温度高于阈值时报警；
3. 报警捕捉：选中“不捕捉”，即不进行报警捕捉，选中“捕捉图片”，即拍下报警时的图像，选中“捕捉视频”，即录像报警时的视频，可在存储页面观看；
4. 报警阈值：设定触发报警的温度值，与报警条件一起，组成触发报警条件。
5. 迟滞温度：为避免误触发报警解除，而设定的参数。当测量区域温度不在报警阈值反报警条件方向迟滞温度范围内，才会解除报警。例如，设置报警条件为“高于”，报警阈值为 30°C，迟滞温度为 2°C。当测量范围内有温度高于 30°C 时，触发报警，而只有测量范围内温度不高于 (30-2) °C 时，才会解除报警。
6. 阈值时间：当测量范围内有温度超过报警阈值，并且持续时间超过阈值时间时，才会触发报警。
7. 报警动作：通过对复选框的勾选，确定报警时热成像仪将要执行的动作：
 - Disable Calib：捕捉图像和视频时，暂时禁止定期校准；
 - E-mail：自动将捕捉的图像或者视频，发送到设置-->平台-->邮箱设置-->收件人地址中定义的接收人；
 - Digital：输出数字脉冲；
 - FTP：自动将捕捉的图像或者视频，发送到设置-->平台-->TFTP 设置-->地址中定义的地址。

完成上述操作后，点击下方的保存按钮保存即可。

8.3.13 环境参数设置

8.3.13.1 概述

除了全局环境参数外，每当使用一个测量工具时，都可以对该测量工具所测量的区域进行局部环境参数设置，如下图所示。



8.3.13.2 设置环境参数局部环境参数的设置，请参考8.3.6 局部环境参数设置。

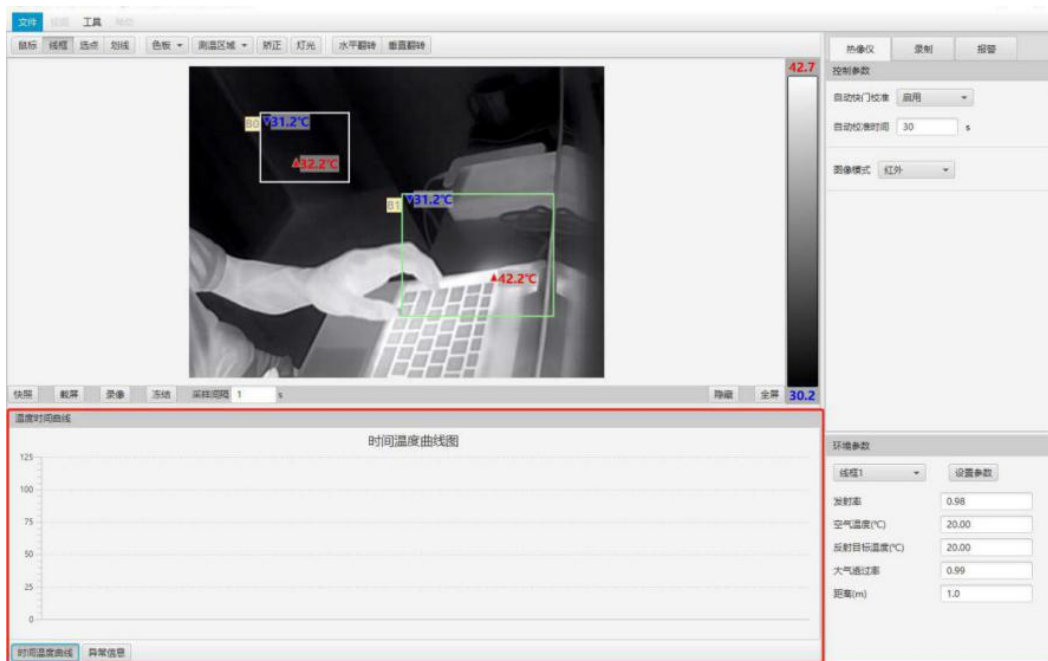
8.3.14 温度实时曲线

8.3.14.1 概述

为了是用户更直观的观测测量区域的温度变化，本产品提供了温度实时曲线测量功能。

8.3.14.2 打开工

在屏幕左下方的工具栏中,单击图标“波形图”，会显示或隐藏温度实时曲线图，如下图所示：

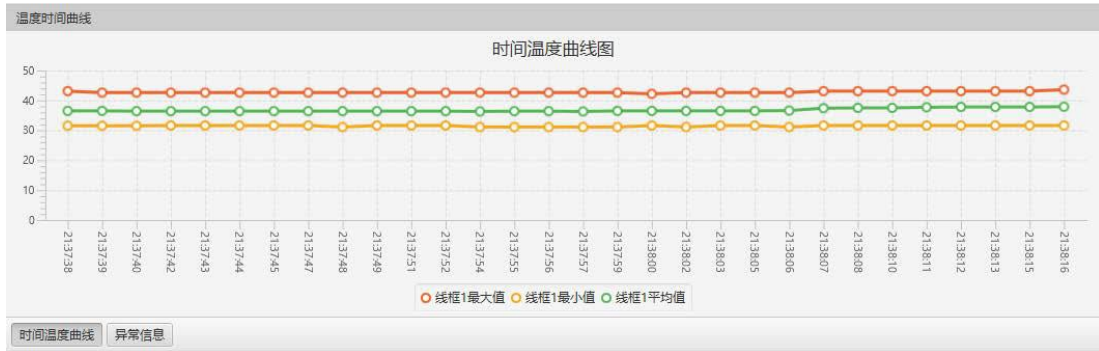


本产品共提供了最大值、最小值、平均值三条测温曲线，每个选项区域里的最大值或最小值或平均值都能在趋势图中直观监控到。

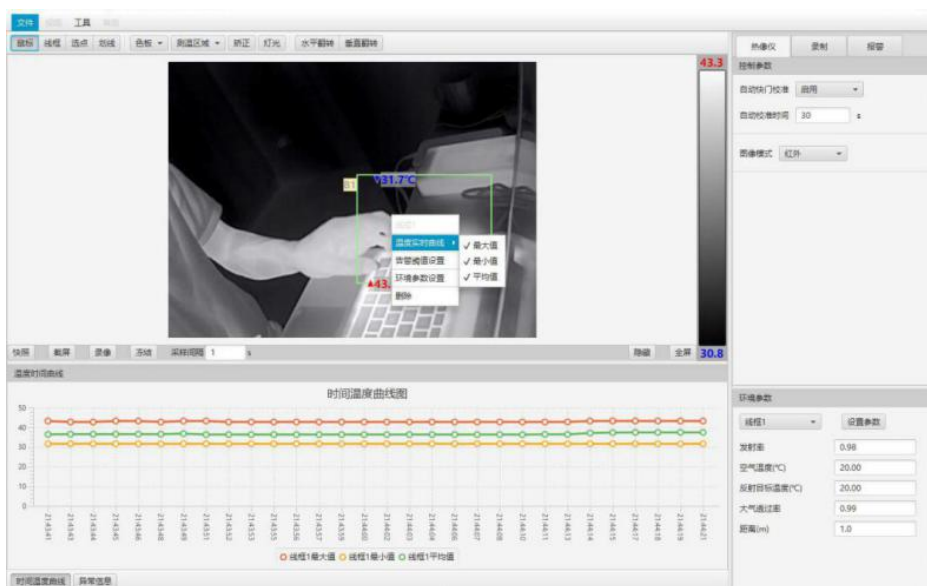
8.3.14.3 操作步骤

若要对某一个测温区域的温度进行实时曲线跟踪时，可按以下步骤进行操作。

1. 选择测量区域：右键弹出菜单，选中要测量的最值和平均值，如下图所示：



2. 选择测量温度值：在温度实时曲线下拉框中，选中要测量的温度值。如下图所示。



注意：只有选中全局测温、框测温、线测温时，才会有最大温度、最小温度、平均温度的选项可以选择。当选中点测温时，测量温度值会直接设定为平均温度，且不可更换。

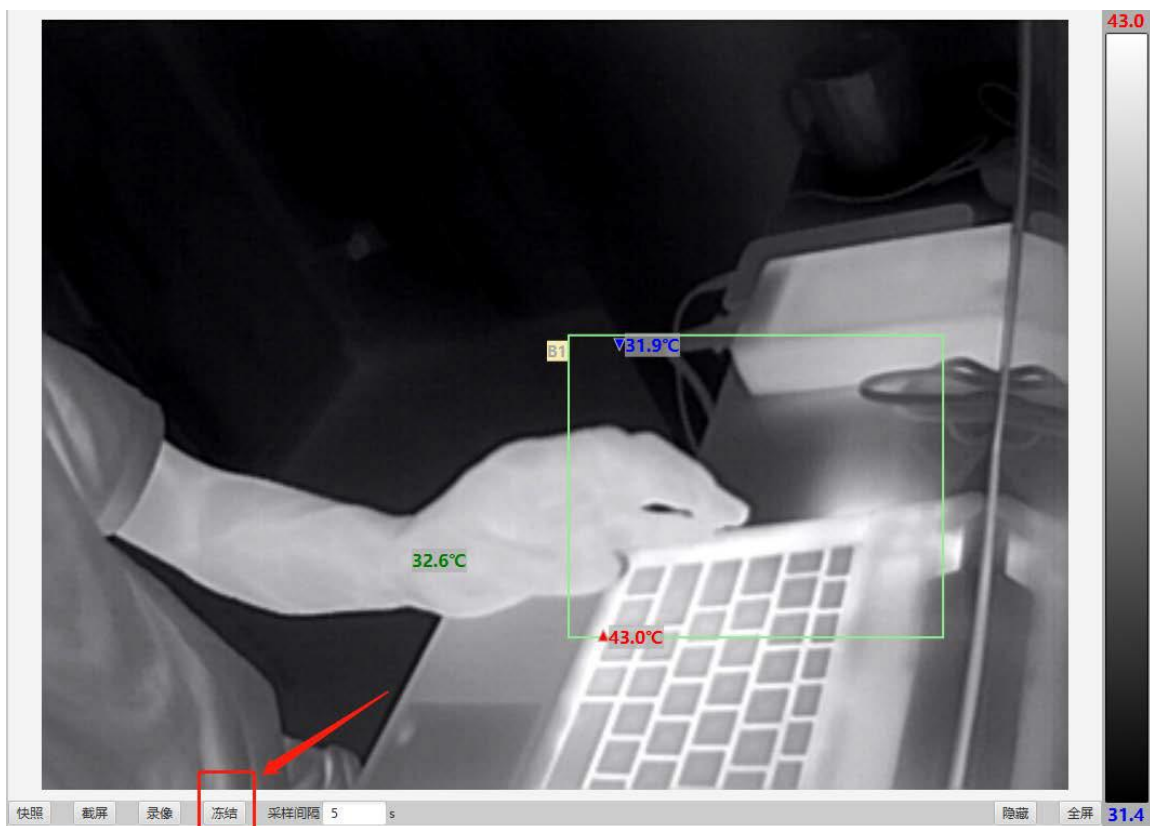
8.3.15 监控视频

8.3.15.1 概述

本产品提供实时视频流播放功能。

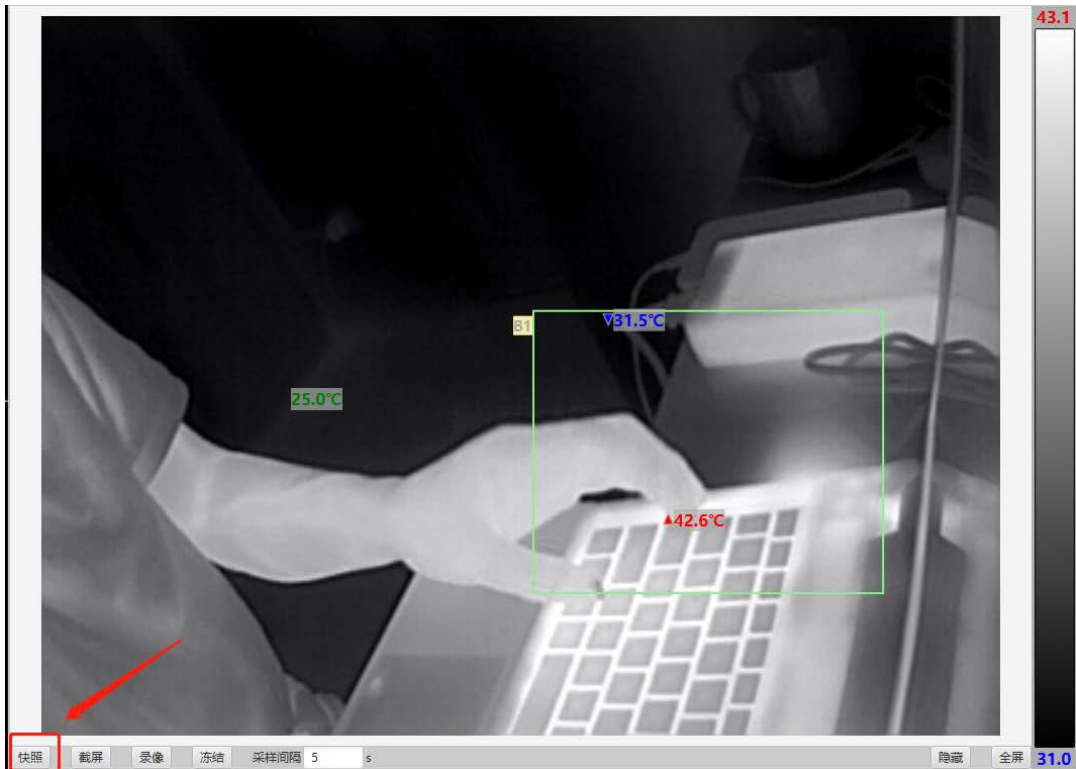
8.3.15.2 暂停

要暂停视频，可点击视频下方“冻结”按钮，视频会暂停
要播放视频，可再一次点击“冻结”按钮，视频恢复，如下图所示。



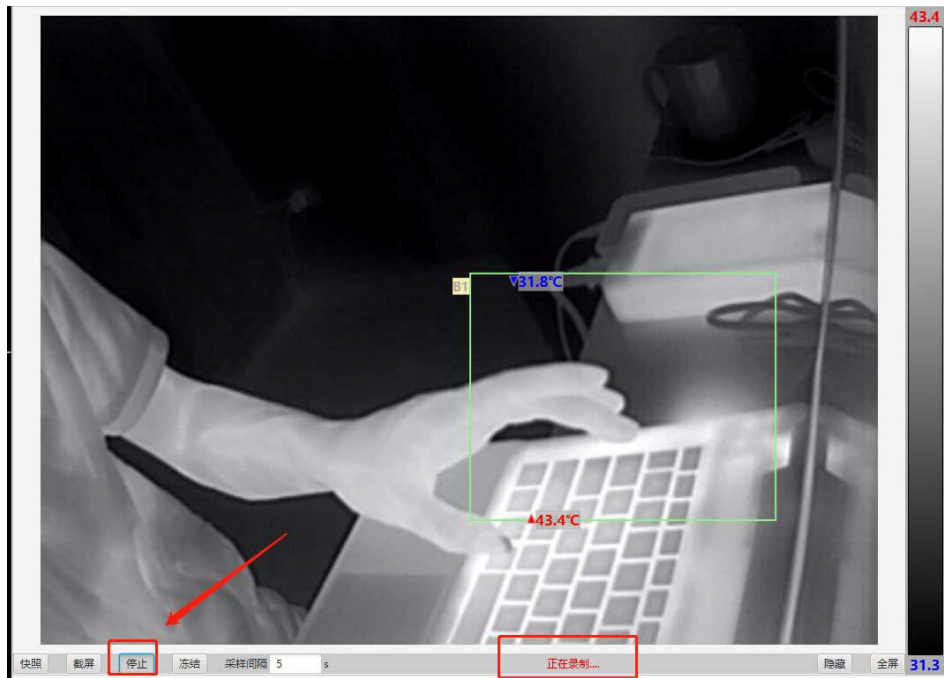
8.3.15.4 快照

若想将当前图像保存为图片，可使用快照功能，单击“快照”按钮即可：



8.3.15.5 录像

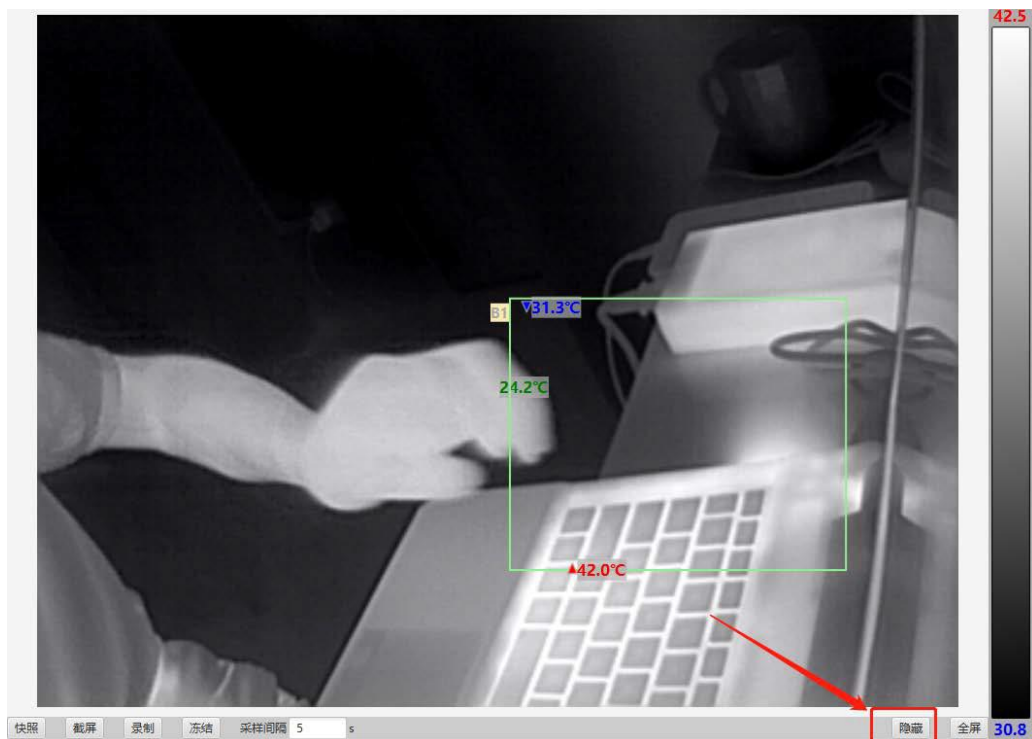
若想录制视频，单击“录制”按钮即可，如下图所示：



若要停止录像，再次单击“停止”按钮即可。

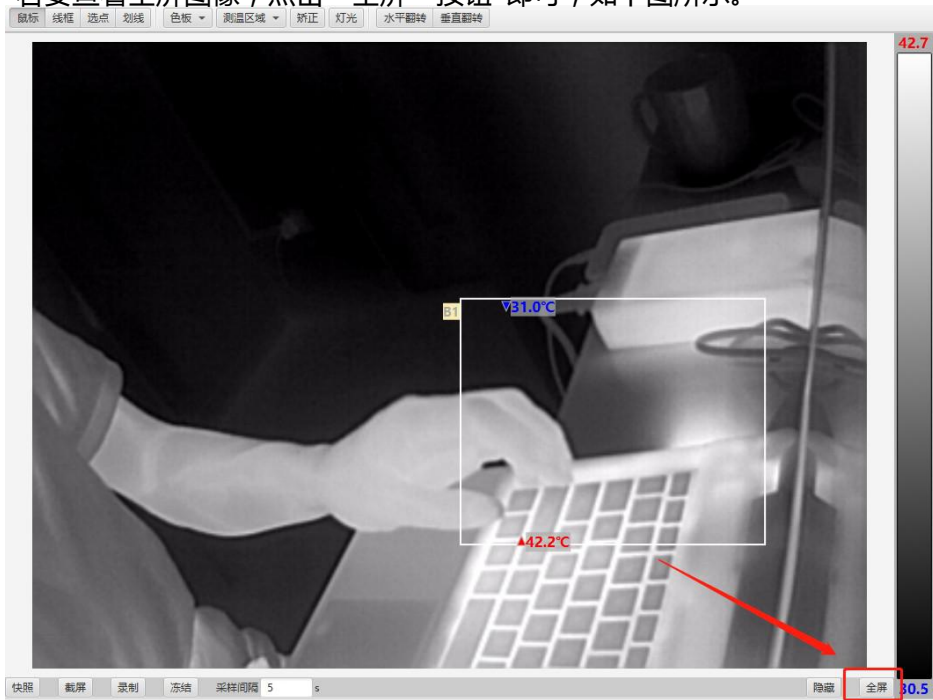
8.3.15.6 隐藏叠加图像

叠加图形提供了关于图像的信息，如测量工具等，若想将他们隐藏，单击“隐藏”按钮，如下图所示：



8.3.15.7 全屏

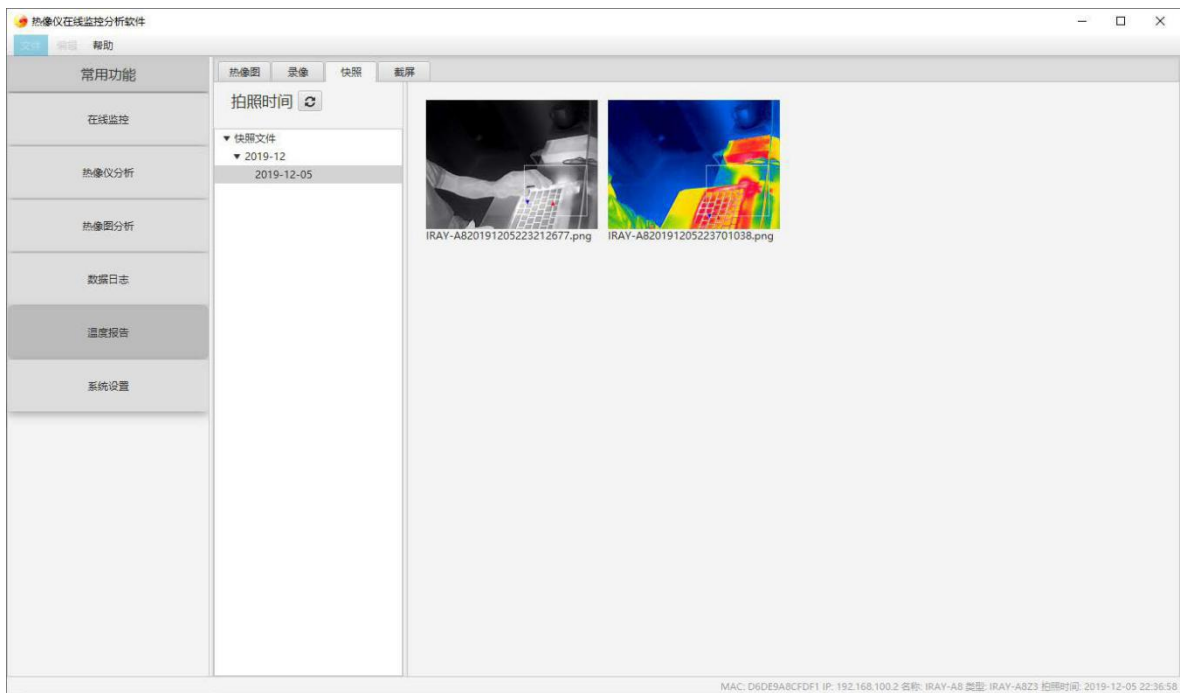
若要查看全屏图像，点击“全屏”按钮即可，如下图所示。



8.4 存储

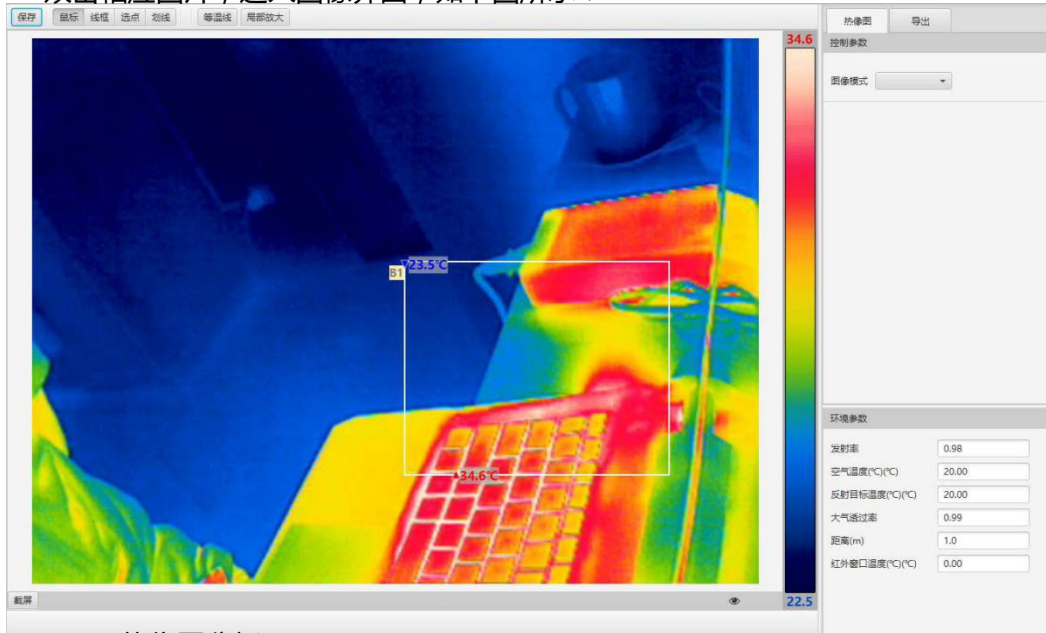
8.4.1 概述

在初始登录界面中，可以看到右侧有“热像图”、“快照”、“录像”、“截图”等图片和录像视频等文件可以预览，如下图所示：



8.4.2 图像二次分析

双击相应图片，进入图像界面，如下图所示：



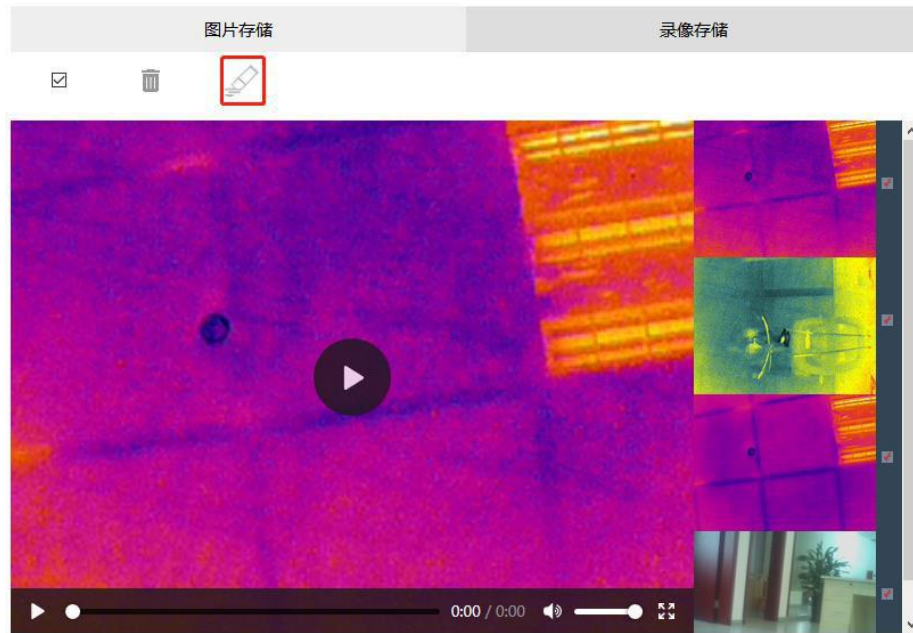
8.4.2.1 热像图分析

在热像图分析界面中，可以使用测量工具进行测量，局部放大等等操作，也可以保存或者截图。

8.4.2.2 文件存储路径

在客户端上获取的视频，照片会存储在客户端的安装路径中的文件夹，右击鼠标可以快速打开文件路径，获取视频或照片。

因为视频都是存储在 sdcard 中,若想对 sdcard 格式化,点击格式化图标即可,如下图所示。



9、 用户二次开发接口

9.1 指令集概述

[sdk_search_device](#) : 搜索设备IP地址。

[sdk_shutter_correction](#) : 进行快门校正。

[sdk_set_shutter_auto_correction](#) : 设置自动快门间隔时间。

[sdk_get_shutter_auto_correction](#) : 获得自动快门间隔时间。

[sdk_set_color_plate](#) : 设置色板模式。

[sdk_get_color_plate](#) : 获取色板模式。

[sdk_set_video_mirror](#) : 设置镜像状态。

[sdk_get_video_mirror](#) : 获取镜像状态。

[sdk_set_video_mode](#) : 设置视频模式。

[sdk_get_video_mode](#) : 获取视频模式。

[sdk_set_area_pos](#) : 设置单个温度测量区域位置。

[sdk_get_area_pos](#) : 获取单个温度测量区域位置。

[sdk_set_spot_pos](#) : 设置单个温度测量点位置

[sdk_get_spot_pos](#) : 获取单个温度测量点位置

[sdk_set_line_pos](#) : 设置温度测量线位置。

[sdk_get_line_pos](#) : 获取温度测量线位置。

[sdk_set_temp_range](#) : 设置扩展温度段开闭。

[sdk_set_video_isp_x_offset](#) : 设置融合图像x轴方向偏移距离。

[sdk_get_video_isp_x_offset](#) : 获取融合图像x轴方向偏移距离。

[sdk_set_video_isp_y_offset](#) : 设置融合图像y轴方向偏移距离。

[sdk_get_video_isp_y_offset](#) : 获取融合图像y轴方向偏移距离。

[sdk_set_video_isp_x_scale](#) : 设置融合图像x轴方向缩放比例。

[sdk_get_video_isp_x_scale](#) : 获取融合图像x轴方向缩放比例。

[sdk_set_video_isp_y_scale](#) : 设置融合图像y轴方向缩放比例。

[sdk_get_video_isp_y_scale](#) : 获取融合图像y轴方向缩放比例。

[sdk_set_led](#) : 设置led灯的开闭状态。

[sdk_get_led](#) : 获取led灯的开闭状态。

[sdk_set_email_server](#) : 设置邮件服务相关信息。

[sdk_get_email_server](#) : 获取邮件服务相关信息。

[sdk_set_tftp_server](#) : 设置tftp传输服务相关信息。

[sdk_get_tftp_server](#) : 获取tftp传输服务相关信息。

[sdk_set_network_eth](#) : 设置网络连接相关信息。

[sdk_get_network_eth](#) : 获取网络连接相关信息。

[sdk_set_fusion_distance](#) : 设置融合距离。

[sdk_get_fusion_distance](#) : 获取融合距离。

[sdk_set_envir_param](#) : 设置单个对象环境参数。

[sdk_get_area_envir_param](#) : 获取单个测温区域环境参数。

[sdk_get_spot_envir_param](#) : 获取单个测温点环境参数。

[sdk_get_line_envir_param](#) : 获取测温线环境参数。

[sdk_get_globo_envir_param](#) : 获取全局环境参数。

[sdk_set_alarm_param](#) : 设置单个对象温度报警开闭、阈值等相关参数。

[sdk_get_area_alarm_param](#) : 获取单个测温区域温度报警开闭、阈值等相关参数。

[sdk_get_spot_alarm_param](#) : 获取单个测温点温度报警开闭、阈值等相关参数。

[sdk_get_line_alarm_param](#) : 获取测温线温度报警开闭、阈值等相关参数。

[sdk_get_area_temp_data](#) : 获取单个测温区域温度信息。

[sdk_get_spot_temp_data](#) : 获取单个测温点温度信息。

[sdk_get_line_temp_data](#) : 获取测温线温度信息。

[sdk_get_globo_temp_data](#) : 获取全局温度信息。

[sdk_get_all_temp_data](#) : 获取所有温度信息。

[sdk_power_reboot](#) : 设备重启指令。

[sdk_param_recover](#) : 设备参数初始化指令。

[sdk_update](#) : 设备固件升级指令。

[sdk_heartbeat](#) : 设备心跳检测。

9.2 具体指令集描述

9.2.1 sdk_search_device

【描述】

搜索设备 IP 地址。

【函数】

int sdk_search_device(char* device_list, int list_len);

【参数】

参数名称	描述	输入/输出
device_list	字符串, 存放获取的 IP 地址。	输出
list_len	字符串大小	输入

【返回值】

返回值	描述
0	指令发送成功。
1	输入参数不符合规范。
2	连接失败。

【需求】

·头文件：H2sdk.h

·库文件：H2sdk.dll、H2sdk.lib

【注意】 无

【相关函数】

[sdk_set_network_eth](#)

9.2.2 sdk_shutter_correction

【描述】

进行快门校正。

【函数】

int sdk_shutter_correction(const char* ip);

【参数】

参数名称	描述	输入/输出
ip	字符串，填入设备 IP 地址。	输入

【返回值】

返回值	描述
0	指令发送成功。
1	输入参数不符合规范。
2	连接失败。

【需求】

·头文件：H2sdk.h

·库文件：H2sdk.dll、H2sdk.lib

【注意】 无

【相关函数】

无

9.2.3 sdk_set_shutter_auto_correction

【描述】

设置自动快门间隔时间

【函数】

`int sdk_set_shutter_auto_correction(const char* ip, int time_sec);`

【参数】

参数名称	描述	输入/输出
ip	字符串，填入设备 IP 地址。	输入
time_sec	自动快门间隔时间	输入

【返回值】

返回值	描述
0	指令发送成功。
1	输入参数不符合规范。
2	连接失败。

【需求】

·头文件：H2sdk.h

·库文件：H2sdk.dll、H2sdk.lib

【注意】

自动快门间隔时间范围在 30~1800s 之内，否则视为关闭。

【相关函数】

[sdk_get_shutter_auto_correction](#)

9.2.4 sdk_get_shutter_auto_correction

【描述】

获取自动快门间隔时间

【函数】

```
int sdk_get_shutter_auto_correction(const char* ip, int* time_sec);
```

【参数】

参数名称	描述	输入/输出
ip	字符串，填入设备IP地址。	输入
time_sec	自动快门间隔时间	输出

【返回值】

返回值	描述
0	指令发送成功。
1	输入参数不符合规范。
2	连接失败。

【需求】

·头文件：H2sdk.h

·库文件：H2sdk.dll、H2sdk.lib

【注意】

无

【相关函数】

[sdk_set_shutter_auto_correction](#)

9.2.5 sdk_set_color_plate

【描述】

设置色板模式

【函数】

```
int sdk_set_color_plate(const char* ip, int color_plate);
```

【参数】

参数名称	描述	输入/输出
ip	字符串，填入设备IP地址。	输入
color_plate	色板模式	输入

【返回值】

返回值	描述
0	指令发送成功。
1	输入参数不符合规范。
2	连接失败。

【需求】

·头文件：H2sdk.h

·库文件：H2sdk.dll、H2sdk.lib

【注意】

色板模式，现共有八种：白热、黑热、铁热、紫红黄、蓝绿红、彩虹 1、黑红、彩虹 2，分别对应参数 0~7.

【相关函数】

[sdk_get_color_plate](#)

9.2.6 sdk_get_color_plate

【描述】

获取色板模式

【函数】

```
int sdk_get_color_plate(const char* ip, int* color_plate);
```

【参数】

参数名称	描述	输入/输出
ip	字符串，填入设备IP地址。	输入
color_plate	色板模式	输出

【返回值】

返回值	描述
0	指令发送成功。
1	输入参数不符合规范。
2	连接失败。

【需求】

·头文件：H2sdk.h

·库文件：H2sdk.dll、H2sdk.lib

【注意】

色板模式共有八种：白热、黑热、铁热、紫红黄、蓝绿红、彩虹 1、黑红、彩虹 2，分别对

应参数 0~7.

【相关函数】

[sdk_set_color_plate](#)

9.2.7 sdk_set_video_mirror

【描述】

设置视频镜像状态

【函数】

```
int sdk_set_video_mirror(const char* ip, int mirror_mode);
```

【参数】

参数名称	描述	输入/输出
ip	字符串，填入设备IP地址。	输入
mirror_mode	镜像模式	输入

【返回值】

返回值	描述
0	指令发送成功。
1	输入参数不符合规范。
2	连接失败。

【需求】

·头文件：H2sdk.h

·库文件：H2sdk.dll、H2sdk.lib

【注意】

镜像模式有常态、水平翻转、垂直翻转三种状态，分别对应参数 0~2。

【相关函数】

[sdk_get_video_mirror](#)

9.2.8 sdk_get_video_mirror

【描述】

获取视频镜像状态

【函数】

```
int sdk_get_video_mirror(const char* ip, int* mirror_mode);
```

【参数】

参数名称	描述	输入/输出
ip	字符串，填入设备IP地址。	输入
mirror_mode	镜像模式	输出

【返回值】

返回值	描述
0	指令发送成功。
1	输入参数不符合规范。
2	连接失败。

【需求】

头文件：H2sdk.h

库文件：H2sdk.dll、H2sdk.lib

【注意】

镜像模式有常态、水平翻转、垂直翻转三种状态，分别对应参数 0~2。

【相关函数】

[sdk_set_video_mirror](#)

9.2.9 sdk_set_video_mode

【描述】

设置视频模式

【函数】

```
int sdk_set_video_mode(const char* ip, int video_mode);
```

【参数】

参数名称	描述	输入/输出
ip	字符串，填入设备IP地址。	输入
video_mode	视频模式	输入

【返回值】

返回值	描述
0	指令发送成功。
1	输入参数不符合规范。
2	连接失败。

【需求】

·头文件：H2sdk.h

·库文件：H2sdk.dll、H2sdk.lib

【注意】

视频模式有红外、可见光、融合 1~18 共 20 种模式，分别对应参数 0~19。

【相关函数】

[sdk_get_video_mode](#)

9.2.10 sdk_get_video_mode

【描述】

获取视频模式

【函数】

int sdk_get_video_mode(const char* ip, int* video_mode);

【参数】

参数名称	描述	输入/输出
ip	字符串，填入设备IP地址。	输入
video_mode	视频模式	输出

【返回值】

返回值	描述
0	指令发送成功。
1	输入参数不符合规范。
2	连接失败。

【需求】

·头文件：H2sdk.h

·库文件：H2sdk.dll、H2sdk.lib

【注意】

视频模式有红外、可见光、融合 1~18 共 20 种模式，分别对应参数 0~19。

【相关函数】

[sdk_set_video_mode](#)

9.2.11 sdk_set_area_pos

【描述】

设置单个测温区域位置

【函数】

```
int sdk_set_area_pos(const char* ip,int index, area_pos* area_data);
```

【参数】

参数名称	描述	输入/输出
ip	字符串，填入设备IP地址。	输入
index	测温区域编号	输入
area_data	结构体，包括区域的位置信息	输入

【返回值】

返回值	描述
0	指令发送成功。

1	输入参数不符合规范。
2	连接失败。

【需求】

·头文件：H2sdk.h

·库文件：H2sdk.dll、H2sdk.lib

【注意】

测温区域的编号范围为 0~5 共 6 个区域。结

构体具体元素

typedef struct

```
{  
  
    int enable; int  
  
    x;  
  
    int y;  
  
    int width; int  
  
    height;  
  
} area_pos;
```

enable 为 1 使能，为 0 禁用；x、y 为区域起始点坐标；width、height 为区域宽高；坐标与宽高输入为 240*180 比例下的值。

【相关函数】

[sdk_get_area_pos](#)

[sdk_get_area_temp_data](#)

9.2.12 sdk_get_area_pos

【描述】

获取单个测温区域位置

【函数】

```
int sdk_get_area_pos(const char* ip,int index, area_pos* area_data);
```

【参数】

参数名称	描述	输入/输出
ip	字符串，填入设备IP地址。	输入
index	获取测温区域编号	输入
area_data	结构体，包括区域的位置信息	输出

【返回值】

返回值	描述
0	指令发送成功。
1	输入参数不符合规范。
2	连接失败。

【需求】

·头文件：H2sdk.h

·库文件：H2sdk.dll、H2sdk.lib

【注意】

测温区域的编号范围为 0~5 共 6 个区

域。结构体具体元素

typedef struct

```
{  
    int enable;  
  
    int x;  
  
    int y;  
  
    int width;  
  
    int height;  
}  
} area_pos;
```

enable 为 1 使能, 为 0 禁用; x、y 为区域起始点坐标; width、height 为区域宽高; 坐标与宽高输入为 240*180 比例下的值。

【相关函数】

[sdk_set_area_pos](#)

9.2.13 sdk_set_spot_pos

【描述】

设置单个测温点位置

【函数】

```
int sdk_set_spot_pos(const char* ip,int index, spot_pos* spot_data);
```

【参数】

参数名称	描述	输入/输出
ip	字符串, 填入设备IP地址。	输入

index	设置测温点编号	输入
spot_data	结构体，包括点的位置信息	输入

【返回值】

返回值	描述
0	指令发送成功。
1	输入参数不符合规范。
2	连接失败。

【需求】

·头文件：H2sdk.h

·库文件：H2sdk.dll、H2sdk.lib

【注意】

设置测温点编号范围为 0~5，共 6 个区域。

具体结构体元素

typedef struct

```
{
    int enable;

    int x;

    int y;
} spot_pos;
```

enable 为 1 使能，为 0 禁用；x、y 为 240*180 比例下测温点坐标。

【相关函数】

[sdk_get_spot_pos](#)

[sdk_get_spot_temp_data](#)

9.2.14 sdk_get_spot_pos

【描述】

获取单个测温点坐标

【函数】

```
int sdk_get_spot_pos(const char* ip,int index, spot_pos*  
spot_data);
```

描述	输入/输出
字符串，填入设备IP地址。	输入
获取测温点编号	输入
结构体，包括点的位置信息	输出

【返回值】

返回值	描述
0	指令发送成功。
1	输入参数不符合规范。
2	连接失败。

【需求】

·头文件：H2sdk.h

·库文件：H2sdk.dll、H2sdk.lib

【注意】

设置测温点编号范围为 0~5，共 6 个点。

具体结构体元素

typedef struct

{

int enable;

int x;

int y;

} spot_pos;

enable 为 1 使能，为 0 禁用；x、y 为 240*180 比例下测温点坐标。

【相关函数】

[sdk_set_spot_pos](#)

9.2.15 sdk_set_line_pos

【描述】

设置测温线位置

【函数】

```
int sdk_set_line_pos(const char* ip, line_pos* line_data);
```

【参数】

参数名称	描述	输入/输出
ip	字符串，填入设备IP地址。	输入

line_data	结构体，包括线的位置信息	输入
-----------	--------------	----

【返回值】

返回值	描述
0	指令发送成功。
1	输入参数不符合规范。
2	连接失败。

【需求】

·头文件：H2sdk.h

·库文件：H2sdk.dll、H2sdk.lib

【注意】

具体结构体元素

typedef struct

```
{
    int enable; int
    sta_x;
    int sta_y;
    int end_x;
    int end_y;
} line_pos;
```

enable 为 1 使能，为 0 禁用；sta_x、sta_y、end_x、end_y 分别为 240*180 比例下线起始点与终点的坐标参数。

【相关函数】

[sdk_get_line_pos](#)

[sdk_get_line_temp_data](#)

9.2.16 sdk_get_line_pos

【描述】

获取测温线位置

【函数】

```
int sdk_get_line_pos(const char* ip, line_pos* line_data);
```

【参数】

参数名称	描述	输入/输出
ip	字符串，填入设备IP地址。	输入
line_data	结构体，包括线的位置信息	输出

【返回值】

返回值	描述
0	指令发送成功。
1	输入参数不符合规范。
2	连接失败。

【需求】

·头文件：H2sdk.h

·库文件：H2sdk.dll、H2sdk.lib

【注意】

具体结构体元素

typedef struct

```
{  
    int enable;  
    int sta_x;  
    int sta_y;  
    int end_x;  
    int end_y;
```

```
} line_pos;
```

enable 为 1 使能, 为 0 禁用; sta_x、sta_y、end_x、end_y 分别为 240*180 比例下线起始点与终点的坐标参数。

【相关函数】

[sdk_set_line_pos](#)

9.2.17 sdk_set_temp_range

【描述】

设置扩展温度段开闭

【函数】

```
int sdk_set_temp_range(const char* ip, int temp_mode);
```

【参数】

参数名称	描述	输入/输出
ip	字符串，填入设备IP地址。	输入
temp_mode	扩展温度段模式	输入

【返回值】

返回值	描述
0	指令发送成功。
1	输入参数不符合规范。
2	连接失败。

【需求】

·头文件：H2sdk.h

·库文件：H2sdk.dll、H2sdk.lib

【注意】

扩展温度段模式 0 为常温段，1 为扩展温度段。

【相关函数】

无

9.2.18 sdk_set_video_isp_x_offset

【描述】

设置融合图像 x 轴方向偏移值

【函数】

```
int sdk_set_video_isp_x_offset(const char* ip, int lr_offset);
```

【参数】

参数名称	描述	输入/输出
ip	字符串，填入设备IP地址。	输入
lr_offset	x轴方向偏移值	输入

【返回值】

返回值	描述
0	指令发送成功。
1	输入参数不符合规范。
2	连接失败。

【需求】

·头文件：H2sdk.h

·库文件：H2sdk.dll、H2sdk.lib

【注意】

x 轴方向偏移值，正值右偏负值左偏。

【相关函数】

[sdk_get_video_isp_x_offset](#)

9.2.19 sdk_get_video_isp_x_offset

【描述】

获取融合图像 x 轴方向偏移值

【函数】

```
int sdk_get_video_isp_x_offset(const char* ip, int* lr_offset);
```

【参数】

参数名称	描述	输入/输出
ip	字符串，填入设备IP地址。	输入
lr_offset	x轴方向偏移值	输出

【返回值】

返回值	描述
0	指令发送成功。
1	输入参数不符合规范。
2	连接失败。

【需求】

·头文件：H2sdk.h

·库文件：H2sdk.dll、H2sdk.lib

【注意】

x 轴方向偏移值，正值右偏负值左偏。

【相关函数】

[sdk_set_video_isp_x_offset](#)

9.2.20 sdk_set_video_isp_y_offset

【描述】

设置融合图像 y 轴方向偏移值

【函数】

`int sdk_set_video_isp_y_offset(const char* ip, int tb_offset);`

【参数】

参数名称	描述	输入/输出
ip	字符串，填入设备IP地址。	输入
tb_offset	y轴方向偏移值	输入

【返回值】

返回值	描述
0	指令发送成功。
1	输入参数不符合规范。
2	连接失败。

【需求】

·头文件：H2sdk.h

·库文件：H2sdk.dll、H2sdk.lib

【注意】

y 轴方向偏移值正值上偏，负值下偏。

【相关函数】

[sdk_get_video_isp_y_offset](#)

9.2.21 sdk_get_video_isp_y_offset

【描述】

设置融合图像 y 轴方向偏移值

【函数】

int sdk_get_video_isp_y_offset (const char* ip, int tb_offset);

【参数】

参数名称	描述	输入/输出
ip	字符串，填入设备IP地址。	输入
tb_offset	y轴方向偏移值	输出

【返回值】

返回值	描述
0	指令发送成功。
1	输入参数不符合规范。
2	连接失败。

【需求】

·头文件：H2sdk.h

·库文件：H2sdk.dll、H2sdk.lib

【注意】

y 轴方向偏移值正值上偏，负值下偏。

【相关函数】

[sdk_set_video_isp_y_offset](#)

9.2.22 sdk_set_video_isp_x_scale

【描述】

设置融合图像 x 轴方向缩放比例

【函数】

```
int sdk_set_video_isp_x_scale(const char* ip, int scale_x);
```

【参数】

参数名称	描述	输入/输出
ip	字符串，填入设备IP地址。	输入
scale_x	x轴方向缩放比例	输入

【返回值】

返回值	描述
0	指令发送成功。
1	输入参数不符合规范。
2	连接失败。

【需求】

·头文件：H2sdk.h

·库文件：H2sdk.dll、H2sdk.lib

【注意】

x 轴方向缩放比例正值缩小，负值放大。

【相关函数】

[sdk_get_video_isp_x_scale](#)

9.2.23 sdk_get_video_isp_x_scale

【描述】

获取融合图像 x 轴方向缩放比

【函数】

int sdk_get_video_isp_x_scale (const char* ip, int scale_x);

【参数】

参数名称	描述	输入/输出
ip	字符串，填入设备IP地址。	输入
scale_x	x轴方向缩放比例	输出

【返回值】

返回值	描述
0	指令发送成功。
1	输入参数不符合规范。
2	连接失败。

【需求】

·头文件：H2sdk.h

·库文件：H2sdk.dll、H2sdk.lib

【注意】

x 轴方向缩放比例正值缩小，负值放大。

【相关函数】

[sdk_set_video_isp_x_scale](#)

9.2.24 sdk_set_video_isp_y_scale

【描述】

设置融合图像 y 轴方向缩放比

【函数】

int sdk_set_video_isp_y_scale (const char* ip, int scale_y);

【参数】

参数名称	描述	输入/输出
ip	字符串，填入设备IP地址。	输入
scale_y	y轴方向缩放比例	输入

【返回值】

返回值	描述
0	指令发送成功。
1	输入参数不符合规范。
2	连接失败。

【需求】

·头文件：H2sdk.h

·库文件：H2sdk.dll、H2sdk.lib

【注意】

y 轴方向缩放比例，正值缩小，负值放大。

【相关函数】

[sdk_get_video_isp_y_scale](#)

9.2.25 sdk_get_video_isp_y_scale

【描述】

获取融合图像 x 轴方向缩放比

【函数】

```
int sdk_get_video_esp_y_scale(const char* ip, int* scale_y);
```

【参数】

参数名称	描述	输入/输出
ip	字符串，填入设备IP地址。	输入
scale_y	y轴方向缩放比例	输出

【返回值】

返回值	描述
0	指令发送成功。
1	输入参数不符合规范。
2	连接失败。

【需求】

·头文件：H2sdk.h

·库文件：H2sdk.dll、H2sdk.lib

【注意】

y 轴方向缩放比例，正值缩小，负值放大。

【相关函数】

[sdk_set_video_esp_y_scale](#)

9.2.26 sdk_set_led

【描述】

设置 led 灯开

【函数】

```
int sdk_set_led(const char* ip, int light_param);
```

【参数】

参数名称	描述	输入/输出
ip	字符串，填入设备IP地址。	输入
light_param	led开闭参数	输入

【返回值】

返回值	描述
0	指令发送成功。
1	输入参数不符合规范。
2	连接失败。

【需求】

·头文件：H2sdk.h

·库文件：H2sdk.dll、H2sdk.lib

【注意】

led 开闭参数 1 为开，0 为闭

【相关函数】

[sdk_get_led](#)

9.2.27 sdk_get_led

【描述】

获取 led 灯开闭状

【函数】

```
int sdk_get_led (const char* ip, int light_param);
```

【参数】

参数名称	描述	输入/输出
ip	字符串，填入设备IP地址。	输入
light_param	led开闭参数	输出

【返回值】

返回值	描述
0	指令发送成功。
1	输入参数不符合规范。
2	连接失败。

【需求】

·头文件：H2sdk.h

·库文件：H2sdk.dll、H2sdk.lib

【注意】

led 开闭参数 1 为开，0 为闭

【相关函数】

[sdk_set_led](#)

9.2.28 sdk_set_email_server

【描述】

设置邮件服务相关信息

【函数】

```
int sdk_set_email_server(const char* ip, email_server* server_param);
```

【参数】

参数名称	描述	输入/输出
ip	字符串，填入设备IP地址。	输入
server_param	结构体，包含邮件服务相关信息	输入

【返回值】

返回值	描述
0	指令发送成功。
1	输入参数不符合规范。
2	连接失败。

【需求】

·头文件：H2sdk.h

·库文件：H2sdk.dll、H2sdk.lib

【注意】

此功能需要开启 dhcp 联网后方能有效使用。

结构体具体元素

```
typedef struct
```

```
{
```



```

int enable;

char recv_addr[20];

char send_addr[20];

char send_pwd[20];

} email_server;

```

enable 为 1 使能，为 0 禁用；recv_addr 为字符串，填入收件人邮箱地址；send_addr、send_pwd 字符串分别填入发件人邮箱地址以及密码。

【相关函数】

[sdk_get_email_server](#)

9.2.29 sdk_get_email_server

【描述】

获取邮件服务相关信息

【函数】

```
int sdk_get_email_server(const char* ip, email_server* server_param);
```

【参数】

参数名称	描述	输入/输出
ip	字符串，填入设备IP地址。	输入
server_param	结构体，包含邮件服务相关信息	输出

【返回值】

返回值	描述

0	指令发送成功。
1	输入参数不符合规范。
2	连接失败。

【需求】

·头文件：H2sdk.h

·库文件：H2sdk.dll、H2sdk.lib

【注意】

此功能不需开启 dhcp 便可使用。

【相关函数】

无

9.2.30 sdk_set_tftp_server

【描述】

设置 tftp 服务相关信息

【函数】

```
int sdk_set_tftp_server(const char* ip, tftp_server* server_param);
```

【参数】

参数名称	描述	输入/输出
ip	字符串，填入设备IP地址。	输入
server_param	结构体，包含tftp服务相关信息	输入

【返回值】

返回值	描述
0	指令发送成功。
1	输入参数不符合规范。
2	连接失败。

【需求】

·头文件：H2sdk.h

·库文件：H2sdk.dll、H2sdk.lib

【注意】

结构体具体元素

typedef struct

```
{  
  
    int enable;  
  
    char tftp_addr[20];  
  
} tftp_server;
```

enable 为 1 使能，为 0 禁用；tftp_addr 为字符串，填入 tftp 传输 IP 地址。

【相关函数】

[sdk_get_tftp_server](#)

9.2.31 sdk_get_tftp_server

【描述】

获取 tftp 服务信息

【函数】

`int sdk_get_tftp_server(const char* ip, tftp_server* server_param);`

【参数】

参数名称	描述	输入/输出
ip	字符串，填入设备IP地址。	输入
server_param	结构体，包含tftp服务相关信息	输出

【返回值】

返回值	描述
0	指令发送成功。
1	输入参数不符合规范。
2	连接失败。

【需求】

·头文件：H2sdk.h

·库文件：H2sdk.dll、H2sdk.lib

【注意】

结构体具体元素

`typedef struct`

`{`

`int enable;`

`char tftp_addr[20];`

`} tftp_server;`

enable 为 1 使能，为 0 禁用；tftp_addr 为 tftp 传输 IP 地址。

【相关函数】

[sdk_set_tftp_server](#)

9.2.32 sdk_set_network_eth

【描述】

设置网络连接相关信息

【函数】

```
int sdk_set_network_eth(const char* ip, network_eth* network_param);
```

【参数】

参数名称	描述	输入/输出
ip	字符串，填入设备IP地址。	输入
network_param	结构体，包含静态IP,dhcp使能等参数信息	输入

【返回值】

返回值	描述
0	指令发送成功。
1	输入参数不符合规范。
2	连接失败。

【需求】

头文件：H2sdk.h

·库文件：H2sdk.dll、H2sdk.lib

【注意】

具体结构体元素

typedef struct

```
{  
  
    int enable;  
  
    char static_ip[20];  
  
    char netmask[20];  
  
    char gateway[20];  
  
    char dns1[20];  
  
    char dns2[20];  
  
} network_eth;
```

enable 为 1 使能 dhcp，~~剩余参数可为空~~，设备 IP 联网后自动分配；enable 为 0 禁用 dhcp，需配置 static_ip 静态 IP、netmask 子网掩码、gateway 网关、dns1、dns2 域名信息。

【相关函数】

[sdk_search_device](#)

[sdk_get_network_eth](#)

9.2.33 sdk_get_network_eth

【描述】

获取网络连接相关信息

【函数】

int sdk_get_network_eth(const char* ip, network_eth* network_param);

【参数】

参数名称	描述	输入/输出
ip	字符串，填入设备IP地址。	输入
network_param	结构体，包含静态IP,dhcp使能等参数信息	输出

【返回值】

返回值	描述
0	指令发送成功。
1	输入参数不符合规范。
2	连接失败。

【需求】

·头文件：H2sdk.h

·库文件：H2sdk.dll、H2sdk.lib

【注意】

具体结构体元素

typedef struct

{

int enable;

char static_ip[20];

char netmask[20];

```

char gateway[20];

char dns1[20];

char dns2[20];

} network_eth;

```

enable 为 dhcp 使能位, static_ip 为静态IP netmask 为子网掩码, gateway 为网关 dns1、dns2 为域名信息。

【相关函数】

[sdk_set_network_eth](#)

9.2.34 sdk_set_fusion_distance

【描述】

设置融合距离

【函数】

```
int sdk_set_fusion_distance(const char* ip, int distance);
```

【参数】

参数名称	描述	输入/输出
ip	字符串，填入设备IP地址。	输入
distance	融合距离参数	输入

【返回值】

返回值	描述
0	指令发送成功。

1	输入参数不符合规范。
2	连接失败。

【需求】

·头文件：H2sdk.h

·库文件：H2sdk.dll、H2sdk.lib

【注意】

distance 单位为 cm，范围为 0~120。

【相关函数】

[sdk_get_fusion_distance](#)

9.2.35 sdk_get_fusion_distance

【描述】

获取融合距离

【函数】

```
int sdk_get_fusion_distance(const char* ip, int* distance);
```

【参数】

参数名称	描述	输入/输出
ip	字符串，填入设备IP地址。	输入
distance	融合距离参数	输出

【返回值】

返回值	描述

0	指令发送成功。
1	输入参数不符合规范。
2	连接失败。

【需求】

·头文件：H2sdk.h

·库文件：H2sdk.dll、H2sdk.lib

【注意】

无

【相关函数】

[sdk_set_fusion_distance](#)

9.2.36 sdk_set_envir_param

【描述】

设置单个对象环境参数

【函数】

```
int sdk_set_envir_param(const char* ip, envir_param* envir_data);
```

【参数】

参数名称	描述	输入/输出
ip	字符串，填入设备IP地址。	输入
envir_data	结构体，包含辐射率、发射率等参数	输入

【返回值】

返回值	描述
0	指令发送成功。
1	输入参数不符合规范。
2	连接失败。

【需求】

·头文件：H2sdk.h

·库文件：H2sdk.dll、H2sdk.lib

【注意】

结构体具体元素

typedef struct

{

int method; int

num;

float emissivity;

float airTemp;

float targetTemp;

float atmosTrans;

float distance;

float infraredTemp;

float infraredRadia;

```
} envir_param;
```

emissivity 为发射率参数,airTemp 为空气温度,targetTemp 为反射物体温度,atmosTrans为大气透过率,distance 为距离,infraredTemp 为红外窗口温度,infraredRadia 为热辐射量。

结构体中 float 数据精度为三位,如 emissivity = 0.784。

结构体中 float 数据精度为三位,如 emissivity = 0.784。

【相关函数】

[sdk_get_area_envir_param](#)

[sdk_get_spot_envir_param](#)

[sdk_get_line_envir_param](#)

[sdk_get_globa_envir_param](#)

9.2.37 sdk_get_area_envir_param

【描述】

获取单个测温区域环境参数

【函数】

```
int (const char* ip, int index, envir_param* envir_data);
```

【参数】

参数名称	描述	输入/输出
ip	字符串,填入设备IP地址。	输入
index	测温区域编号,范围0~5	输入

envir_data	结构体，包含辐射率、发射率等参数	输出
------------	------------------	----

【返回值】

返回值	描述
0	指令发送成功。
1	输入参数不符合规范。
2	连接失败。

【需求】

·头文件：H2sdk.h

·库文件：H2sdk.dll、H2sdk.lib

【注意】

结构体具体元素

typedef struct

```
{
    int method;
    int num;
    float emissivity;
    float airTemp;
    float targetTemp;
    float atmosTrans;
    float distance;
```

```

float infraredTemp;

float infraredRadia;

} envir_param;

```

emissivity 为发射率参数,airTemp 为空气温度,targetTemp 为反射物体温度,atmosTrans为大气透过率,distance 为距离,infraredTemp 为红外窗口温度,infraredRadia 为热辐射量。

结构体中 float 数据精度为三位,如 emissivity = 0.784。

【相关函数】

[sdk_set_envir_param](#)

9.2.38 sdk_get_spot_envir_param

【描述】

获取单个测温点环境参数

【函数】

```
int sdk_get_spot_envir_param(const char* ip, int index, envir_param* envir_data);
```

【参数】

参数名称	描述	输入/输出
ip	字符串,填入设备IP地址。	输入
index	测温点编号,范围0~5	输入
envir_data	结构体,包含辐射率、发射率等参数	输出

【返回值】

返回值	描述
0	指令发送成功。
1	输入参数不符合规范。
2	连接失败。

【需求】

·头文件：H2sdk.h

·库文件：H2sdk.dll、H2sdk.lib

【注意】

结构体具体元素

typedef struct

{

int method; int

num;

float emissivity;

float airTemp;

float targetTemp;

float atmosTrans;

float distance;

float infraredTemp;

float infraredRadia;

```
} envir_param;
```

emissivity 为发射率参数,airTemp 为空气温度,targetTemp 为反射物体温度,atmosTrans为大气透过率,distance 为距离,infraredTemp 为红外窗口温度,infraredRadia 为热辐射量。
结构体中 float 数据精度为三位,如 emissivity = 0.784。

【相关函数】

[sdk_set_envir_param](#)

9.2.39 sdk_get_line_envir_param

【描述】

获取测温线环境参数

【函数】

```
int sdk_get_line_envir_param(const char* ip, envir_param* envir_data);
```

【参数】

参数名称	描述	输入/输出
ip	字符串,填入设备IP地址。	输入
envir_data	结构体,包含辐射率、发射率等参数	输出

【返回值】

返回值	描述
0	指令发送成功。

1	输入参数不符合规范。
2	连接失败。

【需求】

·头文件：H2sdk.h

·库文件：H2sdk.dll、H2sdk.lib

【注意】
结构体具体元素

typedef struct

{

int method; int

num;

float emissivity; float

airTemp; float

targetTemp; float

atmosTrans;

float distance; float

infraredTemp; float

infraredRadia;

} envir_param;

emissivity 为发射率参数,airTemp 为空气温度,targetTemp 为反射物体温度,atmosTrans为大气透过率,distance 为距离,infraredTemp 为红外窗口温度,infraredRadia 为热辐

射量。

结构体中 float 数据精度为三位，如 emissivity = 0.784。

【相关函数】

[sdk_set_envir_param](#)

9.2.40 sdk_get_globa_envir_param

【描述】

获取全局环境参数

【函数】

```
int sdk_get_globa_envir_param(const char* ip, envir_param* envir_data);
```

【参数】

参数名称	描述	输入/输出
ip	字符串，填入设备IP地址。	输入
envir_data	结构体，包含辐射率、发射率等参数	输出

【返回值】

返回值	描述
0	指令发送成功。
1	输入参数不符合规范。
2	连接失败。

【需求】

·头文件：H2sdk.h

·库文件：H2sdk.dll、H2sdk.lib

【注意】

结构体具体元素

typedef struct

```
{  
  
    int method;  
  
    int num;  
  
    float emissivity;  
  
    float airTemp;  
  
    float targetTemp;  
  
    float atmosTrans;  
  
    float distance;  
  
    float infraredTemp;  
  
    float infraredRadia;  
  
} envir_param;
```

emissivity 为发射率参数,airTemp 为空气温度,targetTemp 为反射物体温度,atmosTrans为大气透过率,distance 为距离,infraredTemp 为红外窗口温度,infraredRadia 为热辐射量。

结构体中 float 数据精度为三位,如 emissivity = 0.784。

【相关函数】

[sdk_set_envir_param](#)

9.2.41 sdk_set_alarm_param

【描述】

设置单个对象报警参数

【函数】

```
int sdk_set_alarm_param(const char* ip, alarm_param* alarm_data);
```

【参数】

参数名称	描述	输入/输出
ip	字符串，填入设备 IP 地址。	输入
alarm_data	结构体，包含报警阈值、迟滞时间、报警方式使能等参数信息。	输入

【返回值】

返回值	描述
0	指令发送成功。
1	输入参数不符合规范。
2	连接失败。

【需求】

·头文件：H2sdk.h

·库文件：H2sdk.dll、H2sdk.lib

【注意】

具体结构体元素

typedef struct

```
{  
    int method;  
  
    int num;  
  
    int active;  
  
    int condition; int  
    captrue;  
  
    int disableCalib; int  
    email;  
  
    int digital; int  
    ftp;  
  
    float threshold; float  
    hysteresis;  
  
    int thresholeTime;  
  
} alarm_param;
```

method 代表设置对象的类型，根据区域、点、线分别为 1、2、3；num 为区域、点、线编号，区域与点编号为 0~5，线设置 0 即可；active 为 1 使能报警，为 0 禁用报警；condition 为报警条件，0 为低温报警，1 为高温报警；captrue 为报警捕捉方式，0 为不捕捉，1 为捕捉照片，2 为捕捉视频；disableCalib、email、digital、ftp 为对应功能使能位；threshold

为报警阈值；hysteresis 为迟滞温度，即低于（高于）报警温度此值时，取消报警；

thresholdTime 为迟滞时间，当满足报警条件并经过该时间后，触发报警。

结构体中 float 数据精度为一位，如 threshold = 20.6。

【相关函数】

[sdk_get_area_alarm_param](#)

[sdk_get_spot_alarm_param](#)

[sdk_get_line_alarm_param](#)

9.2.42 sdk_get_area_alarm_param

【描述】

获取单个测温区域报警信息

【函数】

```
int sdk_get_area_alarm_param(const char* ip, int index, alarm_param* alarm_data);
```

【参数】

参数名称	描述	输入/输出
ip	字符串，填入设备 IP 地址。	输入
index	测温区域编号，范围 0~5。	输入
alarm_data	结构体，包含报警阈值、迟滞时间、报警方式使能等参数信息。	输出

【返回值】

返回值	描述
0	指令发送成功。
1	输入参数不符合规范。
2	连接失败。

【需求】

·头文件：H2sdk.h
·库文件：H2sdk.dll、H2sdk.lib

【注意】

具体结构体元素

typedef struct

{

int

method;

int num;

int active;

.....

int captrue;

int disableCalib;

int email;

int digital;

int ftp;

float threshold;

```
float hysteresis;
```

```
int thresholeTime;
```

```
} alarm_param;
```

active 为 1 使能报警，为 0 禁用报警；condition 为报警条件，0 为低温报警，1 为高温报警；captrue 为报警捕捉方式，0 为不捕捉，1 为捕捉照片，2 为捕捉视频；disableCalib、email、digital、ftp 为对应功能使能位；threshold 为报警阈值；hysteresis 为迟滞温度，即低于（高于）报警温度此值时，取消报警；thresholeTime 为迟滞时间，当满足报警条件并经过该时间后，触发报警。

结构体中 float 数据精度为一位，如 threshold = 20.6。

【相关函数】

[sdk_set_alarm_param](#)

9.2.43 sdk_get_spot_alarm_param

【描述】

获取单个测温点报警信息

【函数】

```
int sdk_get_spot_alarm_param(const char* ip, int index, alarm_param* alarm_data);
```

【参数】

参数名称	描述	输入/输出
------	----	-------

ip	字符串，填入设备IP地址。	输入
index	测温点编号，范围0~5.	输入
alarm_data	结构体，包含报警阈值、迟滞时间、报警方式使能等参数信息。	输出

【返回值】

返回值	描述
0	指令发送成功。
1	输入参数不符合规范。
2	连接失败。

【需求】

·头文件：H2sdk.h

·库文件：H2sdk.dll、H2sdk.lib

【注意】

具体结构体元素

typedef struct

```
{
    int method;
    int num;
    int active;
    int condition;
    int captrue;
```

```
int disableCalib;  
  
int email;  
  
int digital;  
  
int ftp;  
  
float threshold;  
  
float hysteresis;  
  
int thresholeTime;  
  
} alarm_param;
```

active 为 1 使能报警，为 0 禁用报警；condition 为报警条件，0 为低温报警，1 为高温报警；captrue 为报警捕捉方式，0 为不捕捉，1 为捕捉照片，2 为捕捉视频；disableCalib、email、digital、ftp 为对应功能使能位；threshold 为报警阈值；hysteresis 为迟滞温度，即低于（高于）报警温度此值时，取消报警；thresholeTime 为迟滞时间，当满足报警条件并经过该时间后，触发报警。

结构体中 float 数据精度为一位，如 threshold = 20.6。

【相关函数】

[sdk_set_alarm_param](#)

9.2.44 sdk_get_line_alarm_param

【描述】

获取测温线报警信息

【函数】

```
int sdk_get_line_alarm_param(const char* ip, alarm_param* alarm_data);
```

【参数】

参数名称	描述	输入/输出
ip	字符串，填入设备IP地址。	输入
alarm_data	结构体，包含报警阈值、迟滞时间、报警方式使能等参数信息。	输出

【返回值】

返回值	描述
0	指令发送成功。
1	输入参数不符合规范。
2	连接失败。

【需求】

·头文件：H2sdk.h

·库文件：H2sdk.dll、H2sdk.lib

【注意】

具体结构体元素

typedef struct

```
{  
  
    int method;  
  
    int num;  
  
    int active;  
  
    int condition;
```

```
int captrue;

int disableCalib;

int email;

int digital;

int ftp;

float threshold;

float hysteresis;

int thresholeTime;

} alarm_param;
```

active 为 1 使能报警，为 0 禁用报警；condition 为报警条件，0 为低温报警，1 为高温报警；captrue 为报警捕捉方式，0 为不捕捉，1 为捕捉照片，2 为捕捉视频；disableCalib、email、digital、ftp 为对应功能使能位；threshold 为报警阈值；hysteresis 为迟滞温度，即低于（高于）报警温度此值时，取消报警；thresholeTime 为迟滞时间，当满足报警条件并经过该时间后，触发报警。

结构体中 float 数据精度为一位，如 threshold = 20.6。

【相关函数】

[sdk_set_alarm_param](#)

9.2.45 sdk_get_area_temp_data

【描述】

获取单个测温区域温度信息

【函数】

int sdk_get_area_temp_data(const char* ip,int index, area_temp* area_data);

【参数】

参数名称	描述	输入/输出
ip	字符串，填入设备IP地址。	输入
index	测温区域编号，范围0~5	输入
area_data	结构体，包含最值、平均温度信息以及对应位置等信息	输出

【返回值】

返回值	描述
0	指令发送成功。
1	输入参数不符合规范。
2	连接失败。

【需求】

·头文件：H2sdk.h

·库文件：H2sdk.dll、H2sdk.lib

【注意】

具体结构体元素

typedef struct

{

int enable;

int max_temp;

```
    int max_temp_x;  
  
    int max_temp_y;  
  
    int min_temp;  
  
    int min_temp_x;  
  
    int min_temp_y;  
  
    int ave_temp;  
  
} area_temp;
```

enable 为 1 表示该测温区域存在；max_temp 为该区域最高温度，max_temp_x、max_temp_y 分别对应该温度点在240*180 比例下的坐标；min_temp 为该区域最低温度，min_temp_x、min_temp_y 分别对应该温度点在 240*180 比例下的坐标；ave_temp 为该区域平均温度。

获取的温度数据为精确一位放大十倍后的结果，比如 24.3 数据中为 243。

【相关函数】

[sdk_set_area_pos](#)

9.2.46 sdk_get_spot_temp_data

【描述】

获取单个测温点的温度信息

【函数】

```
int sdk_get_spot_temp_data (const char* ip, int index, spot_temp* spot_data);
```

【参数】

参数名称	描述	输入/输出
ip	字符串，填入设备IP地址。	输入
index	测温点编号，范围0~5	输入
spot_data	结构体，包含该点温度信息。	输出

【返回值】

返回值	描述
0	指令发送成功。
1	输入参数不符合规范。
2	连接失败。

【需求】

·头文件：H2sdk.h

·库文件：H2sdk.dll、H2sdk.lib

【注意】

具体结构体元素

typedef struct

```
{
    int enable;
    int temp;
} spot_temp;
```

enable 为 1 表示该测温点存在，temp 为该点温度信息。

获取的温度数据为精确一位放大十倍后的结果，比如 24.3 数据中为 243。

【相关函数】

[sdk_set_spot_pos](#)

9.2.47 sdk_get_line_temp_data

【描述】

获取测温线温度信息

【函数】

```
int sdk_get_line_temp_data(const char* ip, line_temp* line_data);
```

【参数】

参数名称	描述	输入/输出
ip	字符串，填入设备IP地址。	输入
line_data	结构体，包含最值、平均温度信息以及对应位置等信息	输出

【返回值】

返回值	描述
0	指令发送成功。
1	输入参数不符合规范。
2	连接失败。

【需求】

·头文件：H2sdk.h

·库文件：H2sdk.dll、H2sdk.lib

【注意】

具体结构体元素

typedef struct

```
{  
  
    int enable;  
  
    int max_temp;  
  
    int max_temp_x;  
  
    int max_temp_y;  
  
    int min_temp;  
  
    int min_temp_x;  
  
    int min_temp_y;  
  
    int ave_temp;  
  
} line_temp;
```

enable 为 1 表示该测温线存在 ;max_temp 为线最高温度 ,max_temp_x、max_temp_y 分别对应该温度点在 240*180 比例下的坐标 ;min_temp 为线最低温度 ,min_temp_x、min_temp_y 分别对应该温度点在 240*180 比例下的坐标 ;ave_temp 为线平均温度。
获取的温度数据为精确一位放大十倍后的结果 ,比如 24.3 数据中为 243。

【相关函数】

[sdk_set_line_pos](#)

9.2.48 sdk_get_globa_temp_data

【描述】

获取全局温度信息

【函数】

```
int sdk_get_globa_temp_data(const char* ip, globa_temp* globa_data);
```

【参数】

参数名称	描述	输入/输出
ip	字符串，填入设备IP地址。	输入
globa_data	结构体，包含全图最值温度信息。	输出

返回值	描述
0	指令发送成功。
1	输入参数不符合规范。
2	连接失败。

【注意】

具体结构体元素

```
typedef struct
```

```
{  
  
    int max_temp;  
  
    int min_temp;
```

```
} globa_temp;
```

max_temp 为全图最高温度、min_temp 为全图最低温度。

获取的温度数据为精确一位放大十倍后的结果，比如 24.3 数据中为 243。

【相关函数】

无

9.2.49 sdk_get_all_temp_data

【描述】

获取所有温度信息

【函数】

```
int sdk_get_all_temp_data(const char* ip, image_temp* all_data);
```

【参数】

参数名称	描述	输入/输出
ip	字符串，填入设备IP地址。	输入
all_data	结构体，包含所有区域、点、线、 全局温度信息。	输出

【返回值】

返回值	描述
0	指令发送成功。
1	输入参数不符合规范。
2	连接失败。

【需求】

·头文件：H2sdk.h

·库文件：H2sdk.dll、H2sdk.lib

【注意】

具体结构体元素

typedef struct

```
{  
  
    area_temp area[6];  
  
    spot_temp spot[6];  
  
    line_temp line;  
  
    globa_temp globa;  
  
} image_temp;
```

该函数一次性获取所有测温结果。

获取的温度数据为精确一位放大十倍后的结果，比如 24.3 数据中为 243。

【相关函数】

[sdk_get_area_temp_data](#)

[sdk_get_spot_temp_data](#)

[sdk_get_line_temp_data](#)

[sdk_get_globa_temp_data](#)

9.2.50 sdk_power_reboot

【描述】

设备重启指令

【函数】

int sdk_power_reboot(const char* ip);

【参数】

参数名称	描述	输入/输出
ip	字符串，填入设备IP地址。	输入

【返回值】

返回值	描述
0	指令发送成功。
1	输入参数不符合规范。
2	连接失败。

【需求】

·头文件：H2sdk.h

·库文件：H2sdk.dll、H2sdk.lib

【注意】

无

【相关函数】

无

9.2.51 sdk_param_recover

【描述】

参数初始化指令

【函数】

int sdk_param_recover (const char* ip);

【参数】

参数名称	描述	输入/输出
ip	字符串，填入设备IP地址。	输入

【返回值】

返回值	描述
0	指令发送成功。
1	输入参数不符合规范。
2	连接失败。

【需求】

·头文件：H2sdk.h

·库文件：H2sdk.dll、H2sdk.lib

【注意】

无

【相关函数】

无

9.2.52 sdk_update

【描述】

固件升级指令

【函数】

int sdk_update(const char* ip);

【参数】

参数名称	描述	输入/输出
ip	字符串，填入设备IP地址。	输入

【返回值】

返回值	描述
0	指令发送成功。
1	输入参数不符合规范。
2	连接失败。

【需求】

·头文件：H2sdk.h

·库文件：H2sdk.dll、H2sdk.lib

【注意】

该指令使用前需要 sdk 中已存在升级包。

【相关函数】

无

9.2.53 sdk_heartbeat

【描述】

固件升级指令

【函数】

int sdk_heartbeat(const char* ip);

【参数】

参数名称	描述	输入/输出
ip	字符串，填入设备IP地址。	输入

【返回值】

返回值	描述
0	设备仍然在线。
-1	设备运行状态不正常或已掉线。

【需求】

·头文件：H2sdk.h

·库文件：H2sdk.dll、H2sdk.lib

【注意】

无

【相关函数】

无

9.3 获取图像数据

图像可通过不断获取并刷新图片实现，H2 设备提供基于 UDP 协议的图像数据发送服务。

UDP 端口：18891，请求数据时，向 H2 发送“GET_IMAGE”，然后不断的接受数据，并

将接收到的完整数据作为 jpg 处理即可，图像数据格式为：

固定头部(6字节)	图片数据长度(4字节)	图片数据
+IMAGE	Data Length	Data

备注：H2 只提供一路图像数据服务，如果有新的请求，则会停止向原来的请求者发送数据。

9.4 获取原始温度数据

H2 设备提供基于 TCP 协议的原始温度数据发送服务。

TCP 端口：8081，温度数据的格式为：

固定头部(5字节)	温度数据长度(4字节)	温度数据
+TEMP	Data Length	Data

温度数据的更新频率约每秒一条，在红外模组分辨率确定的情况下，每条温度数据的长度（表格中的 Data Length）固定为“宽*高*2”，比如分辨率为“240*180”，则每条温度数据的长度为“240*180*2”。

温度数据的逻辑分布如下所示（简单起见，以假设的分辨率“4*3”为例说明，其他分辨率类似）


L1 L2 L3

L4 L5 L6 L7

L8 L9 L10 L11 L12

H1 H2 H3 H4

H5 H6 H7 H8

将接收到的完整数据作为处理即可。图像数据格式为：
而实际的温度数据分布（即表格中的 Data 分布）为：

L1 L2 L3 ... L10 L11 L12 H1 H2 H3 ... H10 H11 H12

其中 L 表示温度值的低 8 位，H 表示高 8 位。第 1 个像素点的开氏温度 * 10 =

$H1 * 256 + L1$ ，由此得出第 1 个像素点的摄氏温度 = $(H1 * 256 + L1) / 10$

- 273，其他像素点温度计算，以此类推。

10、 简单故障排除

- 上电后指示灯不亮

请检查设备端 M12A 行接口线缆是否固定牢固，请检查电源适配器规格和工作状态，如果是自制线缆，请确认线缆线序是否符合要求。

- 上电后指示灯红灯亮

请检查设备端 M12X 型接口线缆是否固定牢固，请检查线缆 RJ45 水晶头端是否已与连接设备连接牢固，如果是自制线缆，请确认线缆线序是否符合要求。

- 上电后指示灯绿灯亮，但无法连接设备

请检查输入的地址是否正确，如果是已经接入网络中，请确认网络中不存在 IP 冲突