



# **Thermal camera addition HTTP POST protocol**

Version 1.0

Last update: 12.03.2024

## Table of contents

1. HTTP POST for FIRE Detection Event Data.....	3
2. HTTP POST Temperature Measure Data.....	6
3. GetDotTemperature.....	9
Support.....	11

PROX

# 1. HTTP POST for FIRE Detection Event Data

Description	When fire detection event triggered, send fire detection info to HTTP Server
URL	POST /SendAlarmData
<pre>&lt;?xml version="1.0" encoding="UTF-8" ?&gt; &lt;config version="1.7" xmlns="http://www.ipc.com/ver10"&gt; &lt;types&gt; &lt;smartType&gt; &lt;enum&gt;MOTION&lt;/enum&gt; &lt;enum&gt;SENSOR&lt;/enum&gt; &lt;enum&gt;PERIMETER&lt;/enum&gt; &lt;enum&gt;TRIPWIRE&lt;/enum&gt; &lt;enum&gt;PEA&lt;/enum&gt; &lt;enum&gt;AVD&lt;/enum&gt; &lt;enum&gt;OSC&lt;/enum&gt; &lt;enum&gt;CPC&lt;/enum&gt; &lt;enum&gt;CDD&lt;/enum&gt; &lt;enum&gt;IPD&lt;/enum&gt; &lt;enum&gt;VFD&lt;/enum&gt; &lt;enum&gt;VEHICE&lt;/enum&gt; &lt;enum&gt;AOIENTRY&lt;/enum&gt; &lt;enum&gt;AOILEAVE&lt;/enum&gt; &lt;enum&gt;PASSLINECOUNT&lt;/enum&gt; &lt;enum&gt;TRAFFIC&lt;/enum&gt; &lt;enum&gt;FIRE&lt;/enum&gt; &lt;enum&gt;TEMPRATURE&lt;/enum&gt; &lt;/smartType&gt; &lt;subscribeOption&gt; &lt;enum&gt;ALARM&lt;/enum&gt; &lt;enum&gt;FEATURE_RESULT&lt;/enum&gt; &lt;enum&gt;FEATURE_RULE&lt;/enum&gt; &lt;/subscribeOption&gt; &lt;/types&gt; &lt;!--Type of Event--Fire Event--&gt; &lt;smartType type="openAramObj"&gt;FIRE&lt;/smartType&gt;</pre>	

```

<subscribeRelation type="subscribeOption">FEATURE_RULE</subscribeRelation>
<currentTime type="tint64">1668482701705110</currentTime>
<macAddress type="string"><![CDATA[00:18:ae:00:33:37]]></macAddress>
<iveFire>
<fireInfo type="list" count="1">
<item>
<thermalInfo>
<targetId type="uint32">0</targetId>
<status type="perStatus">SMART_NONE</status>
<rect>
<!-- Upper Left Coordinates of Fire Event Target Box-->
<x1 type="uint32">4335</x1>
<y1 type="uint32">4114</y1>
<!-- Lower Right Coordinates of Fire Event Target Box-->
<x2 type="uint32">5233</x2>
<y2 type="uint32">5936</y2>
<!-- Method to get Fire target box central temperature:
(1) Calculation of central coordinates  $[(x1+x2)/2, (y1+y2)/2]$ 
(2) Call API GetDotTemperature to get central temperature -->
</rect>
</thermalInfo>
</item>
</fireInfo>
</iveFire>
<opticalImageInfo>
<sourceDataInfo>
<dataType type="uint32">0</dataType>
<!-- Optical channel image--Width-->
<width type="uint32">1280</width>
<!-- Optical channel image--Height-->
<height type="uint32">720</height>
<!-- Optical channel imag--image data size-->
<sourceBase64Length type="uint32">202388</sourceBase64Length>
<!-- Optical channel imag--image data-->
<sourceBase64Data type="string"><![CDATA[optical full image data ...]]></sourceBase64Data>
</sourceDataInfo>
<listInfo type="list" count="0">
</listInfo >
</opticalImageInfo>
<thermalImageInfo>
<sourceDataInfo>
<dataType type="uint32">0</dataType>

```

```
<!--Thermal channel image-- width-->
<width type="uint32">1280</width>
<!--Thermal channel image-- height-->
<height type="uint32">720</height>
<!-- Thermal channel image—image data size-->
<sourceBase64Length type="uint32">85786</sourceBase64Length>
<!--Thermal channel image—image data-->
<sourceBase64Data type="string"><![CDATA[thermal full image data ...]]></sourceBase64Data>
</sourceDataInfo>
<listInfo type="list" count="1">
<item>
<targetId type="uint32">0</targetId>
<rect>
<x1 type="uint32">0</x1>
<y1 type="uint32">0</y1>
<x2 type="uint32">0</x2>
<y2 type="uint32">0</y2>
</rect>
<targetImageData>
<dataType type="uint32">0</dataType>
<targetType type="uint32">0</targetType>
<width type="uint32">720</width>
<height type="uint32">720</height>
<targetBase64Length type="uint32">54164</targetBase64Length>
<targetBase64Data type="string"><![CDATA[thermal target image data
...]]></targetBase64Data>
</targetImageData>
</item>
</listInfo>
</thermallImageInfo>
</config>
```

## 2. HTTP POST Temperature Measure Data

Description	Send Temperature Measurement Data to HTTP Server
URL	POST /SendAlarmData
<pre>&lt;?xml version="1.0" encoding="UTF-8" ?&gt; &lt;config version="1.7" xmlns="http://www.ipc.com/ver10"&gt; &lt;types&gt; &lt;smartType&gt; &lt;enum&gt;MOTION&lt;/enum&gt; &lt;enum&gt;SENSOR&lt;/enum&gt; &lt;enum&gt;PERIMETER&lt;/enum&gt; &lt;enum&gt;TRIPWIRE&lt;/enum&gt; &lt;enum&gt;PEA&lt;/enum&gt; &lt;enum&gt;AVD&lt;/enum&gt; &lt;enum&gt;OSC&lt;/enum&gt; &lt;enum&gt;CPC&lt;/enum&gt; &lt;enum&gt;CDD&lt;/enum&gt; &lt;enum&gt;IPD&lt;/enum&gt; &lt;enum&gt;VFD&lt;/enum&gt; &lt;enum&gt;VEHICE&lt;/enum&gt; &lt;enum&gt;AOIENTRY&lt;/enum&gt; &lt;enum&gt;AOILEAVE&lt;/enum&gt; &lt;enum&gt;PASSLINECOUNT&lt;/enum&gt; &lt;enum&gt;TRAFFIC&lt;/enum&gt; &lt;enum&gt;FIRE&lt;/enum&gt; &lt;enum&gt;TEMPRATURE&lt;/enum&gt; &lt;/smartType&gt; &lt;subscribeOption&gt; &lt;enum&gt;ALARM&lt;/enum&gt; &lt;enum&gt;FEATURE_RESULT&lt;/enum&gt; &lt;enum&gt;FEATURE_RULE&lt;/enum&gt; &lt;/subscribeOption&gt; &lt;/types&gt; &lt;!--Type of Event--Temperature Measurement&gt; &lt;smartType type="openAlramObj"&gt;TEMPRATURE&lt;/smartType&gt;</pre>	

```
<subscribeRelation type="subscribeOption">FEATURE_RULE</subscribeRelation>
<currentTime type="tint64">1668483173303777</currentTime>
<macAddress type="string"><![CDATA[00:18:ae:00:33:37]]></macAddress>
<iveTemperature>
<temperInfo type="list" count="1">
<item>
<ruleId type="uint32">0</ruleId>
<maxTemper>
<!--Highest Temperature -->
<temerValue type="uint32">2605</temerValue>
<!--X coordinate of highest temperature point-->
<thermalX type="uint32">5546</thermalX>
<!--Y coordinate of highest temperature point -->
<thermalY type="uint32">8229</thermalY>
</maxTemper>
<minTemper>
<!--Lowest Temperature-->
<temerValue type="uint32">2305</temerValue>
<!--X Coordinate of lowest temperature point-->
<thermalX type="uint32">2382</thermalX>
<!--Y Coordinate of lowest temperature point-->
<thermalY type="uint32">5625</thermalY>
</minTemper>
<!--Average Temperature-->
<avTemper type="uint32">2415</avTemper>
<ruleInfo>
<enable type="uint32">1</enable>
<ruleId type="uint32">0</ruleId>
<ruleName type="string"><![CDATA[area1]]></ruleName>
<ruleType type="uint32">2</ruleType>
<thermalBoundary type="list" count="6">
<item><point><x type="uint32">2225</x><y type="uint32">5733</y></point></item>
<item><point><x type="uint32">4200</x><y type="uint32">4833</y></point></item>
<item><point><x type="uint32">6100</x><y type="uint32">5100</y></point></item>
<item><point><x type="uint32">7200</x><y type="uint32">7800</y></point></item>
<item><point><x type="uint32">5075</x><y type="uint32">8400</y></point></item>
<item><point><x type="uint32">3525</x><y type="uint32">8000</y></point></item>
</thermalBoundary>
</ruleInfo>
</item>
</temperInfo>
</iveTemperature>
```

```
<opticalImageInfo>
<sourceDataInfo>
<dataType type="uint32">0</dataType>
<!--Optical channel image width-->
<width type="uint32">1280</width>
<!--Optical channel image height -->
<height type="uint32">720</height>
<!--Optical channel image data size-->
<sourceBase64Length type="uint32">202388</sourceBase64Length>
<!--Optical channel image data -->
<sourceBase64Data type="string"><![CDATA[optical full image data ...]]></sourceBase64Data>
</sourceDataInfo>
<listInfo type="list" count="0">
</listInfo >
</opticalImageInfo>
<thermallImageInfo>
<sourceDataInfo>
<dataType type="uint32">0</dataType>
<!--Thermal channel image width-->
<width type="uint32">1280</width>
<!--Thermal channel image height-->
<height type="uint32">720</height>
<!--Thermal channel image data size-->
<sourceBase64Length type="uint32">85786</sourceBase64Length>
<!--Thermal channel image data-->
<sourceBase64Data type="string"><![CDATA[thermal full image data ...]]></sourceBase64Data>
</sourceDataInfo>
<listInfo type="list" count="0">
</listInfo>
</thermallImageInfo>
</config>
```



### 3. GetDotTemperature

GetDotTemperature	
Description	Get the temperature value data of the specified coordinate point
Typical URL	POST or GET http://<host>[:port]/GetDotTemperature
Channel ID	None
Action name	None
Entity Data	<p>Enter the X,Y value of the specified point.For example:</p> <pre> &lt;!-- Send data format of dot--&gt; &lt;?xml version="1.0" encoding="UTF-8"?&gt; &lt;config&gt; &lt;dotTemperature&gt; &lt;!-- X coordinate of dot--&gt; &lt;hotX type="uint32" min="0" max="10000"&gt;4285&lt;/hotX&gt; &lt;!-- Y coordinate of dot --&gt; &lt;hotY type="uint32" min="0" max="10000"&gt;4433&lt;/hotY&gt; &lt;/dotTemperature&gt; &lt;/config&gt; </pre>
Successful Response	<p>The device detail will be included in the entity of the successful response. For example:</p> <pre> &lt;?xml version="1.0" encoding="UTF-8"?&gt; &lt;config version="1.0" xmlns="http://www.ipc.com/ver10"&gt; &lt;types&gt; &lt;!--Enumeration of unit type of temperature --&gt; &lt;tempUnitsType&gt; &lt;!--Centigrade--&gt; &lt;enum&gt;centigrade&lt;/enum&gt; &lt;!--Fahrenheit--&gt; &lt;enum&gt;Fahrenheit&lt;/enum&gt; &lt;/tempUnitsType&gt; &lt;/types&gt; &lt;switch type="boolean"&gt;&gt;false&lt;/switch&gt; &lt;!--Return dotTemperature information--&gt; &lt;dotTemperature&gt; &lt;!-- X coordinate of dot--&gt; </pre>

```
<hotX>4285</hotX>
```

```
<!-- Y coordinate of dot-->
```

```
<hotY>4433</hotY>
```

```
<!-- The temperature of this dot-->
```

```
<temperature>2885</temperature>
```

```
<!-- return unit type of current temperature-->
```

```
<tempUnits type="tempUnitsType">centigrade</tempUnits>
```

```
</dotTemperature>
```

```
</config>
```

# Support

In case of any questions and problems please feel free to contact with us.

- [wsparcie@ipox.pl](mailto:wsparcie@ipox.pl)  
or just make a call:
- 774-404-404
- 510-510-571

IPPOX