

Partner integration guide

Axis D6310 Air Quality Sensor

July 2025

Table of contents

<u>1</u>	<u>THE SENSOR COMPONENT</u>	<u>3</u>
<u>2</u>	<u>INTEGRATION INTO AXIS CAMERA STATION, ACS</u>	<u>4</u>
<u>3</u>	<u>APIS</u>	<u>4</u>
<u>4</u>	<u>MQTT</u>	<u>8</u>

Introduction

The D6310 is the second product in the environmental sensor category for system products. This is the first stand-alone sensor device and offers the same sensor capabilities as the D6210 while also adding a lot more features. Since it's not a portcast device it does not require a host device, which is one difference between this and the Axis D6210.

Some of the main features of this device are:

- Air quality sensing
 - The device uses the same sensor component as the previous D6210 and will measure the same parameters with the same coverage range and installation requirements.
- Speaker and microphone
 - Will be integrated into AXIS Audio Manager Pro
 - Comes with AXIS Audio Analytics
 - Has a physical switch to enable and disable the microphone
- 4 Multi-colored LED's
- Has its own IP-address
- Offers PoE forwarding meaning it can power a second device with PoE Class 3. This second device will get its own IP address and act just as if it is connected straight to the switch.
- Fully integrated into the AXIS Event System, just like the previous AXIS D6210 Air Quality Sensor
 - Can trigger on smoking/vaping alarms or a user set upper and lower threshold for any of the measured air quality parameters
- The device has a virtual stream of the sensor dashboard that can be integrated as any camera stream
- Data available from a virtual stream, MQTT stream, RTSP stream etc.
 - Also offers a sensor dashboard in the web user interface and gives users a recommendation on numbers to help give an understanding of your indoor air quality
- Recorded data can be downloaded through the host device as a csv file in a zip package

The Axis D6310 Air Quality Sensor is an indoor product aimed at giving the user an understanding of the general air quality of a room by offering the user a variety of different measurements that can contribute to poor air quality when they are too high. The user interface gives a recommendation on what could be considered good levels and when action should be taken to improve them. Should the user not be interested in each value individually there is also a number for AQI, Air Quality Index, which offers a number for the general air quality over time. This number utilizes the particulate matter levels over time and applies a universally accepted algorithm for determining the value.

On top of air quality, the device also works in detecting and alerting for vaping and smoking. This information is given in a detected/undetected value and is based on an algorithm using the measurement values.

More information on air quality as a concept can be found in the whitepaper published [here](#).

1 The sensor component

The measurements given by the sensor are:

- Vaping or smoking
- Air quality index (AQI)
- Particulate Matter (PM 1.0, PM 2.5, PM 4.0, PM 10.0)

-
- Temperature (°C/F)
 - Relative Humidity (%RH)
 - Volatile Organic Compounds (VOC)
 - Nitrogen Oxides (NOx)
 - Carbon dioxide (CO₂)

2 Integration into Axis Camera Station, ACS

In ACS the air quality sensor can be integrated into the Axis Data Insights Dashboard, DID. With this you can visualize data from one or multiple sensors in one dashboard. The dashboard can be viewed as a split view together with the camera stream or as a separate dashboard. There is not a limit to how many sensors can be added, although with an increased number of sensors the data retention time will be decreased as the total data size is limited.

3 APIs

With AxisOS 12.6 these are the available API's related to air quality sensing for the Axis D6310:

1. Get real-time sensor data (GET)

axis-cgi/airquality/metadata.cgi

Description: Get all the real-time sensor data through this CGI.

Example 1:

```
curl -u root:pass --digest http://192.168.77.220/axis-cgi/airquality/metadata.cgi  
response: PM1.0 = 11.60, PM2.5 = 12.20, PM4.0 = 12.20, PM10.0 = 12.20, Temperature = 29.16,  
Humidity = 40.67, VOC = 462.00, NOx = 1.00, CO2 = 20, AQI = 0, Vaping = 0
```

Example 2:

```
curl -u root:pass --digest http://192.168.77.220/axis-cgi/airquality/metadata.cgi  
response: PM1.0 = 11.60, PM2.5 = 12.20, PM4.0 = 12.20, PM10.0 = 12.20, Temperature = 29.16,  
Humidity = 40.67, VOC = 462.00, NOx = 1.00, CO2 = 20, AQI = 25, Vaping = 1
```

-
2. Set/Get configuration (POST)

axis-cgi/airquality/config.cgi

Description: Set/get the limit value for the sensor configuration.

Payload:

```
{"apiVersion":"1.1","method":"paramList"}
{"apiVersion":"1.1","method":"paramUpdate","params":{"key1":"string_value1","key2":int_value2,...}}
```

Method list: paramList | paramUpdate

"paramUpdate" params field support list:

- TemperatureScale = celsius | fahrenheit
- MinPM1p0 = [0, 1000]
- MaxPM1p0 = [0, 1000]
- MinPM2p5 = [0, 1000]
- MaxPM2p5 = [0, 1000]
- MinPM4p0 = [0, 1000]
- MaxPM4p0 = [0, 1000]
- MinPM10p0 = [0, 1000]
- MaxPM10p0 = [0, 1000]
- MinTemperature = [-10, 45]
- MaxTemperature = [-10, 45]
- MinHumidity = [0, 100]
- MaxHumidity = [0, 100]
- MinVOC = [0, 500]
- MaxVOC = [0, 500]
- MinNOx = [0, 500]
- MaxNOx = [0, 500]
- MinCO2 = [0, 40000]
- MaxCO2 = [0, 40000]
- MinAQI = [0, 500]
- MaxAQI = [0, 500]
- DashboardTitle = string
- VapeSensitivity = [1, 5] (1: low, 5: high)

Example 1:

```
curl -u root:pass --digest --header "Content-Type: application/json" --request POST --data
'{"apiVersion":"1.1","method":"paramList"}' http://192.168.77.220/axis-cgi/airquality/config.cgi
```

Response:

```
{"apiVersion":"1.1","method":"paramList","data":{"TemperatureScale":"fahrenheit","MinPM1p0":0,"Max
PM1p0":1000,"MinPM2p5":0,"MaxPM2p5":35,"MinPM4p0":0,"MaxPM4p0":1000,"MinPM10p0":0,"Max
PM10p0":154,"MinTemperature":-
10,"MaxTemperature":45,"MinHumidity":0,"MaxHumidity":90,"MinVOC":0,"MaxVOC":300,"MinNOx":0,"
MaxNOx":150,"MinCO2":0,"MaxCO2":1000,"MinAQI":0,"MaxAQI":100,"DashboardTitle":"Dashboard","
VapeSensitivity":3}}
```

Example 2:

```
curl -u root:pass --digest --header "Content-Type: application/json" --request POST --data
'{"apiVersion":"1.1","method":"paramUpdate","params":{"TemperatureScale":"fahrenheit","MaxAQI":25
0}}' http://192.168.77.220/axis-cgi/airquality/config.cgi
```

Response:

```
{"apiVersion":"1.1","method":"paramUpdate","data":{}}
```

3. Get statistics (POST)

axis-cgi/airquality/statistics.cgi

Description: Get the statistics according to the input "category" and "startTime/endTime".

Payload:

```
{"apiVersion":"1.1","method":"getHistoryData","params":{"category":"ALL","startTime":1718168987,"endTime":1718169460}}
{"apiVersion":"1.1","method":"getVapingRecord","params":{"startTime":0,"endTime":1719306211}}
```

Method list: getHistoryData | getVapingRecord

"getHistoryData" params field support list:

category = ALL | PM1_0 | PM2_5 | PM4_0 | PM10_0 | TEMPERATURE | HUMIDITY | VOC | NOX | CO2 | AQI

startTime = UTC timestamp in second

endTime = UTC timestamp in second

"getVapingRecord" params field support list:

startTime = UTC timestamp in second

endTime = UTC timestamp in second

Example 1:

```
curl -u root:pass --digest --header "Content-Type: application/json" --request POST --data
'{"apiVersion":"1.1","method":"getHistoryData","params":{"category":"ALL","startTime":1730186422,"en
dTime":1730186542}}' http://192.168.77.220/axis-cgi/airquality/statistics.cgi
```

Response:

```
{"apiVersion":"1.1","method":"getHistoryData","data":{"TIMESTAMP":[1730186440,1730186471,17301
86502,1730186533],"PM1_0":[2.9,2.7,2.7,2.6],"PM2_5":[3.4,3.3,3.3,3.2],"PM4_0":[3.8,3.7,3.7,3.5],"PM
10_0":[4.0,3.8,3.8,3.7],"TEMPERATURE":[27.93,27.92,27.92,27.92],"HUMIDITY":[33.93,33.94,33.96,3
3.98],"VOC":[210.0,209.0,208.0,207.0],"NOX":[1.0,1.0,1.0,1.0],"CO2":[661,660,660,657],"AQI":[17,17,
17,17]}}
```

Example 2:

```
curl -u root:pass --digest --header "Content-Type: application/json" --request POST --data
'{"apiVersion":"1.1","method":"getHistoryData","params":{"category":"AQI","startTime":1732525613,"en
dTime":1732526113}}' http://192.168.77.213/axis-cgi/airquality/statistics.cgi
```

Response:

```
{"apiVersion":"1.1","method":"getHistoryData","data":{"TIMESTAMP":[1732525635,1732525666,17325
25697,1732525728,1732525759,1732525790,1732525821,1732525852,1732525883,1732525914,17
32525945,1732525976,1732526007,1732526038,1732526069,1732526100],"AQI":[0,0,0,0,0,0,0,0,0,
0,0,0,0,0,0]}}
```

Example 3:

```
curl -u root:pass --digest --header "Content-Type: application/json" --request POST --data '{"apiVersion":"1.1","method":"getVapingRecord","params":{"startTime":0,"endTime":1719306211}}' http://192.168.77.220/axis-cgi/airquality/statistics.cgi
```

Response:

```
{"apiVersion":"1.1","method":"getVapingRecord","data":{"TIMESTAMP":[1732525635,1732525666,1732525697,1732525728,1732525759]}}
```

4. Get environmental sensor status (POST)

axis-cgi/airquality/status.cgi

Description: Get status of the environmental sensor. Use this CGI to check whether the sensor is connected or not.

Method list:**getSensorStatus:****Purpose:**

Get environment sensor type and connection status of portcast device.

Payload:

```
{"apiVersion":"1.1","method":"getSensorstatus"}  
response support list:  
sensorType = (string) d6210 | d6310 | "" if no sensor is connected  
connected = (bool) true | false (Always be false if the sensor type is invalid)
```

Example:*request:*

```
curl -u root:pass --digest --header "Content-Type: application/json" --request POST --data '{"apiVersion":"1.1","method":"getSensorStatus"}' http://192.168.77.243/axis-cgi/airquality/status.cgi
```

response:

```
{"apiVersion":"1.1","method":"getSensorStatus","data":{"sensorType":"d6210","connected":true}}
```

5. Download ZIP (POST)

axis-cgi/airquality/download.cgi

Description: Download the statistics data within a ZIP format file according to the input "category" and "startTime/endTime".

Payload:

```
{"apiVersion":"1.1","method":"downloadHistory","params":{"category":"ALL","startTime":1718168987,"endTime":1718169460}}
{"apiVersion":"1.1","method":"downloadVaping","params":{"startTime":0,"endTime":1719306211}}
{"apiVersion":"1.1","method":"downloadAll","params":{"startTime":0,"endTime":1719306211}}
```

Method list: downloadHistory, downloadVaping, downloadAll

"downloadHistory" params field support list:

category = ALL | PM1_0 | PM2_5 | PM4_0 | PM10_0 | TEMPERATURE | HUMIDITY | VOC | NOX | CO2 | AQI

startTime = UTC timestamp in second

endTime = UTC timestamp in second

"downloadVaping" params field support list:

startTime = UTC timestamp in second

endTime = UTC timestamp in second

"downloadAll" params field support list:

startTime = UTC timestamp in second

endTime = UTC timestamp in second

Example 1: curl -u root:pass --digest --request POST --data '{"apiVersion":"1.1","method":"downloadHistory","params":{"startTime":0,"endTime":1719306211,"category":"ALL"}}' -o - <http://192.168.77.245/axis-cgi/airquality/download.cgi>

Example 2: curl -u root:pass --digest --request POST --data '{"apiVersion":"1.1","method":"downloadHistory","params":{"startTime":0,"endTime":1719306211,"category":"PM1_0"}}' -o - <http://192.168.77.245/axis-cgi/airquality/download.cgi>

Example 3: curl -u root:pass --digest --request POST --data '{"apiVersion":"1.1","method":"downloadVaping","params":{"startTime":0,"endTime":1719306211}}' -o - <http://192.168.77.245/axis-cgi/airquality/download.cgi>

Example 4: curl -u root:pass --digest --request POST --data '{"apiVersion":"1.1","method":"downloadAll","params":{"startTime":0,"endTime":1719306211}}' -o - <http://192.168.77.245/axis-cgi/airquality/download.cgi>

4 MQTT

With an increasing number of Axis devices supporting MQTT the Axis D6210 will be no exception. You can use the Web GUI to set the MQTT setting and let the sensor publish air quality data and let other devices subscribe to that information.

There is also the option of having the air quality data displayed in the camera stream as an MQTT-overlay. Either it can be showed a live number or as a graphical overlay showing the data in a graph.

Additional information here:

<https://help.axis.com/axis-os#knowledge-base>

<https://www.axis.com/vapix-library/subjects/t10175981/section/t10173845/display?section=t10173845-t10173910>

<https://www.axis.com/dam/public/78/37/2c/device-integration-with-mqtt-en-US-359893.pdf>

Publish MQTT event:

<https://youtu.be/7-5x1uKPBiU>

Subscribe to MQTT event:

<https://youtu.be/QSsGnHkUfF8>

Who do you contact if you have questions?

Place a support ticket through Axis partner pages.