



Delta Electronics Inc.

UNOweb HTTP API Manual

Version 0.4.3

Aug. 9, 2024

Table of Contents

1.	Introduction	3
2.	API Manual	5
	A. Requirement	5
	B. API	5

1. Introduction

Each model of UNO sensor is a multi-function sensor, e.g. UNOnext, UNOLite and so on. It provides Temperature ($^{\circ}\text{C}/^{\circ}\text{F}$), Humidity (rH%), Ambient Light (lux), CO_2 (ppm), $\text{PM}_{2.5}$ ($\mu\text{g}/\text{m}^3$), PM_{10} ($\mu\text{g}/\text{m}^3$). The advance model optionally provides PM_1 ($\mu\text{g}/\text{m}^3$), TVOC (ppb or $\mu\text{g}/\text{m}^3$), HCHO (ppb or $\mu\text{g}/\text{m}^3$), CO (ppm), and O_3 (ppb). This document introduces using UNOweb HTTP API to retrieve the sensor data of the UNO sensor based on JSON format. Additionally, UNOweb HTTP API also provides the moving average data of sensor based on Taiwan regulation. The data density is 1 record per 6 minutes when the UNO sensor is online.

Note. UNOweb HTTP API only supports the UNO sensor which is already set WiFi and connected to UNOweb.

Table 1 Sensor Table

Sensor Type	Key	Data Unit
Temperature	TEMP	°C
NTC Temperature (opt.)	NTC	°C
Temperature °F	TEMP_F	°F
NTC Temperature °F (opt.)	NTC_F	°F
Humidity	HUMI	rH%
Ambient Light	LUX	lux
CO ₂	CO2	ppm
PM2.5	PM2p5	µg/m ³
PM10	PM10	µg/m ³
PM1 (opt.)	PM1	µg/m ³
TVOC (opt.)	TVOC	ppb
TVOC (opt.)	TVOC_UGM3	µg/m ³
HCHO (opt.)	HCHO	ppb
HCHO (opt.)	HCHO_UGM3	µg/m ³
CO (opt.)	CO	ppm
O ₃ (opt.)	O3	ppb

Table 2 Sensor Moving Average Data

Sensor Type	Key	Data Unit	Rule Description
CO ₂	CO2_ma	ppm	8 hours
PM2.5	PM2p5_ma	µg/m ³	24 hours
PM10	PM10_ma	µg/m ³	24 hours
PM1 (opt.)	PM1_ma	µg/m ³	24 hours
TVOC (opt.)	TVOC_ma	ppb	1 hours
TVOC (opt.)	TVOC_UGM3_ma	µg/m ³	1 hours
HCHO (opt.)	HCHO_ma	ppb	1 hours
HCHO (opt.)	HCHO_UGM3_ma	µg/m ³	1 hours
CO (opt.)	CO_ma	ppm	8 hours
O ₃ (opt.)	O3_ma	ppb	8 hours

PS. If value of sensor is "null" presented unmounted or data unavailable.

2. API Manual

A. Requirement

- UNOweb account with token API permission.
- The SN of an on-line UNO sensor.
- Prepare a HTTP API client. E.g. Postman (<https://www.postman.com/>)

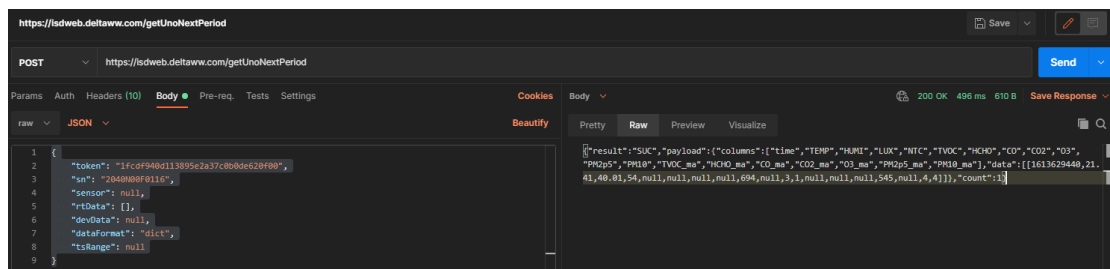


Figure 1 Postman screenshot

B. API

Current UNOweb provides following HTTP API for UNO sensor.

<https://isdweb.deltaww.com/api/getUnoNextPeriod>

Table 3 getUnoNextPeriod Usage

API	Protocol	Description
getUnoNextPeriod	POST	Get UNO sensor data based on moving average data.
Authorization: Bearer Token (In HTTP Request Header)		
User token: Each user has a unique token. Length is 32.		
<i>Format</i>		
Bearer <User token>		
<i>E.g.</i>		
Bearer xxx		
Post Body (JSON Format)		
E.g.		
<pre>{ "sn": "2040N00F0116", "sensor": null, "rtData": [], "dataFormat": "dict", "tsRange": null, "dataInterval": "6m", }</pre>		
JSON Key Description		
Key	Description	
sn	SN of belonged UNOsensor.	
sensor	Moving avg. data of sensor string array. <i>null</i> means all sensor. Empty array [] means no interested moving avg. data.	
rtData	Real-time data of sensor string array. <i>null</i> means no interested real-time data. Empty array [] means all sensor data.	
dataFormat	Accept "dict" ; " csv" ; " json" . Use "dict" for most case.	
dataInterval	Accept "3m" ; " 6m" . Default is "6m" . * 3m interval data only be supported by newest firmware of UNO devices. Please check firmware version and release note. If the device is not support 3m data, the response will contain empty	

	data in the array.
tsRange	Epoch time stamp array. [start, end] → [1613633000, 1613633201] <i>null</i> means the last data in 1 hour. Epoch Example: https://www.epochconverter.com/

Response (application/json)

E.g. Take "dict" format as an example

```
{
  "result": "SUC",
  "payload": {
    "columns": [
      "time",
      "TEMP",
      "HUMI",
      "LUX",
      "NTC",
      "TVOC",
      "HCHO",
      "CO",
      "CO2",
      "O3",
      "PM2p5",
      "PM10",
      "PM1",
      "TVOC_ma",
      "HCHO_ma",
      "CO_ma",
      "CO2_ma",
      "O3_ma",
      "PM2p5_ma",
      "PM10_ma",
      "PM1_ma",
      "TEMP_F",
      "NTC_F",
```

```
"TVOC_UGM3",
"TVOC_UGM3_ma",
"HCHO_UGM3",
"HCHO_UGM3_ma"
],
"data": [
  [
    1716785280,
    27.68,
    55.23,
    0,
    null,
    871,
    null,
    null,
    439,
    null,
    5,
    5,
    null,
    847,
    null,
    null,
    678,
    null,
    2,
    2,
    null,
    81.82,
    null,
    3919,
    3811,
    null,
    null
  ]
]
},
"rowCount": 1,
```


<pre>"count": 1 }</pre>	
<h3>JSON Key Description</h3>	
Key	Description
result	<p>"SUC" is SUCCESSFUL.</p> <p>"FAIL" and "ERR" are return with error message.</p>
payload.columns	<p>Responded array presented sensor column. "time" is epoch stamp. Others can be found in Table 1 and Table 2</p>
payload.data	<p>Responded nested data array, each sub-array means all sensor reading at this time, and each item with i-index in the sub-array is the reading of the corresponded sensor columns in. <i>null</i> means no data at this time stamp, unmounted or sensor abnormal.</p>
count	<p>If the "result" is "SUC" , the "count" presents the length of valid data (not all <i>null</i> data) array.</p>
rawCount	<p>If the "result" is "SUC" , the "count" presents the length of data (contain all <i>null</i> data) array.</p>