

AirSensa Data API

AirSensa: Data Specification

20 April 2016
Bruno Beloff
bruno.beloff@deliverchange.org

Background

The AirSensa device is an air quality monitoring unit designed by Deliver Change. The goal behind the project - broadly speaking - has been to create a piece of hardware that had 80% of the accuracy of the Automatic Urban and Rural Network (AURN) units used by government, at 20% of the cost. This has essentially been achieved, as demonstrated by continuing field trials and early deployments. The current device measures particulate density (PM1, PM2.5 and PM10), plus gas concentrations (NO2, NO, O3 and CO). It also measures temperature, humidity and acoustic peak, and GPS location. Future devices will have options for SO2 and volatile organic compounds (VOCs), the latter being relevant to internal air quality. Current devices are almost always used to measure external air quality. Alternative future lightweight devices will sample only NO2, O3, temperature, humidity and acoustic peak. This saves on unit cost, size, data volume and power. The saving on power is significant, because it enables solar power operations. AirSensa data is uploaded via GSM. Future devices will have options for LoRa, Thread, Thread over sub-gigahertz radio, and (possibly) WiFi.

Data sampling and interpretation

Except for gas concentrations, data interpretation is performed on the device before the data are uploaded. For gas concentrations, "raw" electrode voltages are uploaded. Translation into gas concentrations is performed on the server, based on electrode voltages, temperature, humidity, sensor calibration data, sensor cross-sensitivities, and baselining (zero offset) effects. Both raw and interpreted data are stored on the data capture server. Sampling rates for each device are set by the server, and can change at any time. Normal rates are:

- particulates - every 10 seconds
- gas concentrations - every 10 seconds (based on a linear regression of samples taken every second)
- acoustic peak - every 10 seconds
- temperature, humidity - every 5 minutes

In addition, the device senses "housekeeping" values:

- GPS location - every hour
- input voltage, battery level - every 5 minutes

Sensing operations are co-ordinated, so all sensed phenomena are grouped with the single timestamp of each operation. All AirSensa devices use only UTC. An asset management system keeps track of the current timezone for the device.

Note that, because GSM communications can interfere with the gas sensors, gas sampling is suspended during communications events. Thus, although timeline drift is eliminated by a real-time clock, there will be gaps in the sample timeline.

Communications

Data is uploaded to the Deliver Change server systems via HTTP post (for future versions, HTTPS), over GSM, with a JSON payload. Typically, data is uploaded every five minutes. Data is staged - and the upload object constructed - in flash memory on the device.

Retrieving environmental data

Currently, only a very simple API is available to access the stored, interpreted data. The API was designed for creating timeline charts of single phenomena, for public data. A request looks like this:

http://airsensa.w34u.com/air_ajax/get_box_sensor_data/LOCATION_ID/START_TIME/PERIOD/SENSOR_TYPE/SAMPLE_RATE/

- LOCATION_ID - a numeric unique ID for the location where the device is deployed (we use location ID rather than device ID, because a device may have to be replaced)
- START_TIME - a UNIX millisecond timestamp, or 0 to indicate "now"
- PERIOD - the number of seconds back in time from the START_TIME
- SENSOR_TYPE - this indicates the sensed phenomenon. See **Appendix: Environmental phenomena codes**
- SAMPLE_RATE - the data point frequency. See **Appendix: Sample rates**

An example request:

{+}http://airsensa.w34u.com/air_ajax/get_box_sensor_data/300015/0/3600/opc.pm_1/avg10min/+

The response is:

```
{ "cols": [{"type": "datetime"}, {"type": "number"}, {"type": "number"}, {"type": "number"}], "rows": [{"c": [{"v": "Date(2016,3,19,14,0,6)", "z": "1461070806", "y": "60", {"v": "2.41"}, {"v": "1.67"}, {"v": "3.45"}]}, {"c": [{"v": "Date(2016,3,19,13,50,6)", {"v": "2.14"}, {"v": "1.58"}, {"v": "2.74"}]}, {"c": [{"v": "Date(2016,3,19,13,40,7)", {"v": "2.2"}, {"v": "1.54"}, {"v": "2.99"}]}, {"c": [{"v": "Date(2016,3,19,13,30,8)", {"v": "2.12"}, {"v": "1.47"}, {"v": "2.73"}]}, {"c": [{"v": "Date(2016,3,19,13,20,9)", {"v": "2.19"}, {"v": "1.57"}, {"v": "2.77"}]}]}
```

The three "v" values are average, minimum and maximum, in that order. Sample rates without min / max result in single "v" values.

The date / time for each sample is given as the current local time for the device.

For an example request:

{+}http://airsensa.w34u.com/air_ajax/get_box_sensor_data/300015/0/3600/opc.pm_1/basicone/+

The response is:

```
{ "cols": [{"type": "datetime"}, {"type": "number"}], "rows": [{"c": [{"v": "Date(2016,3,19,14,15,0)", "z": "1461071700", "y": "60", {"v": "2.02"}]}, {"c": [{"v": "Date(2016,3,19,14,14,0)", {"v": "1.98"}]}, {"c": [{"v": "Date(2016,3,19,14,13,0)", {"v": "2.09"}]}, {"c": [{"v": "Date(2016,3,19,14,12,0)", {"v": "2"}]}, {"c": [{"v": "Date(2016,3,19,14,10,10)", {"v": "2.25"}]}, {"c": [{"v": "Date(2016,3,19,14,9,10)", {"v": "2.31"}]}, {"c": [{"v": "Date(2016,3,19,14,8,10)", {"v": "1.99"}]}, {"c": [{"v": "Date(2016,3,19,14,7,6)", {"v": "2.34"}]}, {"c": [{"v": "Date(2016,3,19,14,5,1)", {"v": "3.01"}]}, {"c": [{"v": "Date(2016,3,19,14,4,1)", {"v": "2.51"}]}, {"c": [{"v": "Date(2016,3,19,14,3,1)", {"v": "2.54"}]}, {"c": [{"v": "Date(2016,3,19,14,2,1)", {"v": "2.22"}]}, {"c": [{"v": "Date(2016,3,19,14,0,1)", {"v": "2.22"}]}, {"c": [{"v": "Date(2016,3,19,13,59,1)", {"v": "2.53"}]}, {"c": [{"v": "Date(2016,3,19,13,58,1)", {"v": "2.63"}]}, {"c": [{"v": "Date(2016,3,19,13,57,1)", {"v": "2.5"}]}, {"c": [{"v": "Date(2016,3,19,13,55,11)", {"v": "2.44"}]}, {"c": [{"v": "Date(2016,3,19,13,54,11)", {"v": "2.16"}]}, {"c": [{"v": "Date(2016,3,19,13,53,11)", {"v": "2.28"}]}, {"c": [{"v": "Date(2016,3,19,13,52,7)", {"v": "2.21"}]}, {"c": [{"v": "Date(2016,3,19,13,50,2)", {"v": "2.04"}]}, {"c": [{"v": "Date(2016,3,19,13,49,2)", {"v": "2.09"}]}, {"c": [{"v": "Date(2016,3,19,13,48,2)", {"v": "2.01"}]}, {"c": [{"v": "Date(2016,3,19,13,47,2)", {"v": "2.1"}]}, {"c": [{"v": "Date(2016,3,19,13,45,12)", {"v": "2.18"}]}, {"c": [{"v": "Date(2016,3,19,13,44,12)", {"v": "2.14"}]}, {"c": [{"v": "Date(2016,3,19,13,43,12)", {"v": "2.2"}]}, {"c": [{"v": "Date(2016,3,19,13,42,8)", {"v": "2.33"}]}, {"c": [{"v": "Date(2016,3,19,13,40,3)", {"v": "2.18"}]}, {"c": [{"v": "Date(2016,3,19,13,39,3)", {"v": "2.25"}]}, {"c": [{"v": "Date(2016,3,19,13,38,3)", {"v": "2.3"}]}, {"c": [{"v": "Date(2016,3,19,13,37,3)", {"v": "2.06"}]}, {"c": [{"v": "Date(2016,3,19,13,35,3)", {"v": "2.1"}]}, {"c": [{"v": "Date(2016,3,19,13,34,3)", {"v": "2.02"}]}, {"c": [{"v": "Date(2016,3,19,13,33,3)", {"v": "2.12"}]}, {"c": [{"v": "Date(2016,3,19,13,32,3)", {"v": "2.21"}]}, {"c": [{"v": "Date(2016,3,19,13,30,13)", {"v": "2.11"}]}, {"c": [{"v": "Date(2016,3,19,13,29,13)", {"v": "2.13"}]}, {"c": [{"v": "Date(2016,3,19,13,28,13)", {"v": "2.13"}]}, {"c": [{"v": "Date(2016,3,19,13,27,9)", {"v": "2.11"}]}, {"c": [{"v": "Date(2016,3,19,13,25,4)", {"v": "2.24"}]}, {"c": [{"v": "Date(2016,3,19,13,24,4)", {"v": "2.45"}]}]}
```

Note that the data format was designed to match the requirements of the Google Chart APIs.

Restrictions exist on the number of data points that can be pulled in a single request, and the total volume of data pulled over periods of time.

Future versions of the API will allow multiple sensed phenomena to be retrieved in a single request.

Retrieving housekeeping data

The API for housekeeping data is almost identical to that for sense data: http://airsensa.w34u.com/air_ajax/get_box_status/LOCATION_ID/START_TIME/PERIOD/STATUS_TYPE/basic/

- LOCATION_ID - a numeric unique ID for the location where the device is deployed
- START_TIME - a UNIX millisecond timestamp, or 0 to indicate "now"
- PERIOD - the number of seconds back in time from the START_TIME
- STATUS_TYPE - this indicates the status. See **Appendix: Housekeeping codes**

An example request:

{+}http://airsensa.w34u.com/air_ajax/get_box_status/300015/0/3600/power.vdcin/basic/+

The response is:

```
{ "cols": [{"type": "datetime"}, {"type": "number"}], "rows": [{"c": [{"v": "Date(2016,3,19,15,16,18)", "z": "1461075378", "y": "60", {"v": "12.1"}]}, {"c": [{"v": "Date(2016,3,19,15,11,18)", {"v": "12.2"}]}, {"c": [{"v": "Date(2016,3,19,15,6,18)", {"v": "12.2"}]}, {"c": [{"v": "Date(2016,3,19,14,56,19)", {"v": "12.2"}]}, {"c": [{"v": "Date(2016,3,19,14,51,19)", {"v": "12.2"}]}, {"c": [{"v": "Date(2016,3,19,14,46,19)", {"v": "12.2"}]}, {"c": [{"v": "Date(2016,3,19,14,41,19)", {"v": "12.2"}]}, {"c": [{"v": "Date(2016,3,19,14,36,19)", {"v": "12.1"}]}, {"c": [{"v": "Date(2016,3,19,14,31,19)", {"v": "12.2"}]}, {"c": [{"v": "Date(2016,3,19,14,26,19)", {"v": "12.2"}]}, {"c": [{"v": "Date(2016,3,19,14,21,20)", {"v": "12.1"}]}]}
```

A special call is available for GPS information. This is because GPS signal quality can vary, so the data capture server stores the "best, most recent" GPS fix. The response to the call includes other housekeeping information.

An example request:

{+}http://airsensa.w34u.com/air_ajax/get_box_last_gps/300015/+

The response is:

```
{ "_id": "$id", "571630eb7d21bdebfc1c1a4", "data_format": "pv1", "system_id": "200015", "location_id": "300015", "location_zone": "london_camden", "data_segment": "sensor", "data_rate": "basic", "bogus": 0, "time": {"utc": {"sec": "1461072080", "usec": 0}, "loc": {"sec": "1461075680", "usec": 0}, "zone": "Europe/London", "offset": "3600", "dst": "1", "day": "2"}, "sensor_data": {"power": {"source": "2", "charging_state": "4", "vdcin": "12.1", "fuel_temp": "30.2", "fuel_voltage": "4170", "fuel_full_capacity": "2474", "fuel_level": "100", "charge_rate": "0"}, "attitude": {"acceleration_x": "-0.033", "acceleration_y": "0.029", "acceleration_z": "0.974"}, "gps": {"pos_lat": "51.526123", "pos_lng": "-0.133258", "fix_level": "3", "quality_siv": "13", "quality_hdop": "0.8", "quality_max_snr": "44", "quality_avg_snr": "25.2"}, "phone": {"mcc": "234", "mnc": "33", "lac": "6", "ci": "7579", "bsic": "44"}, "outlier": false, "formatted_local_date": "19th Apr 2016 14:21:20" }
```

The "phone" information identifies the mast ("base station") with which the device is communicating. The GPS location of the mast may be queried on an open databases.

Future versions of the API will include the nearest three masts, to enable triangulation.

Note that, for the purposes of environmental data analysis, the location of the device should be known to within approximately one metre. This is obviously beyond the capabilities of a commercial GPS system (added to which, some locations provide too restricted a view of the sky for any GPS fix to be obtained). Because of this, locations effectively carry two separate geolocation co-ordinates: the "reported location" (the one described here, derived from the GPS receiver on the device, and accessed via the data capture API), and the "registered location" (entered manually when a device is installed, and available via the asset management system). The asset management system APIs are documented elsewhere.

Use case

The following is a web application that makes use of the above APIs:

<http://aed.bwk.co/wellcome/>

(Clicking on a pin or the name of a location shows all phenomena at one location; clicking on a timeline chart or the name of a phenomenon shows a comparison of one phenomenon at all the locations in the group.)

Also:

<http://aed.bwk.co/wellcome/display/>

(This is a rolling display, based on a gauge component. It was designed for continuous presentation at the Wellcome Trust atrium on Euston Road. Note that the layout was designed for use on a 1920x 1080 screed. For best effect, the browser window should be set to roughly this size before the page is loaded. Note that it can take up to one minute for the display to initialise.)

In the above cases, the relevant Javascript code can be examined. The Javascript code can be made available more formally, if requested.

Location IDs for the devices in this group are:

- 300012 - Wellcome Trust (western), Gower Street
- 300013 - Wellcome Trust (central), Euston Road (S)
- 300014 - Wellcome Trust (northern), Stephenson Way
- 300015 - Wellcome Trust (eastern), Gordon Street
- 300016 - Wellcome Trust (southern), Gower Place
- 300017 - Wellcome Trust (central), Euston Road (N)

In accordance with standard Wellcome Trust arrangements, data from these devices is open.

Appendix: Environmental phenomena codes

Atmospheric:

- `atmos.temp` - temperature in degrees centigrade
- `atmos.humid` - relative humidity as a percentage

Acoustic:

- `acoustic.db` - relative sound pressure on a VU scale, i.e. loudest sound is 0 dB, quietest is -48 dB (future devices will be calibrated to dBA)

Gasses:

- `gas.gas.no2.1` - concentration of NO2 as ppb
- `gas.gas.o3.2` - concentration of O3 as ppb
- `gas.gas.no.3` - concentration of NO as ppb
- `gas.gas.co.4` - concentration of CO as ppb

Particulates:

- `opc.pm_1` - density of PM1 as $\mu\text{g}/\text{m}^3$
- `opc.pm_2p5` - density of PM2.5 as $\mu\text{g}/\text{m}^3$
- `opc.pm_10` - density of PM10 as $\mu\text{g}/\text{m}^3$
- `opc.bin_0` - bins work together to provide a histogram of particle sizes from 0.38 to 17 μm
- `opc.bin_1` -
- `opc.bin_2` -
- `opc.bin_3` -
- `opc.bin_4` -
- `opc.bin_5` -
- `opc.bin_6` -
- `opc.bin_7` -
- `opc.bin_8` -
- `opc.bin_9` -
- `opc.bin_10` -
- `opc.bin_11` -
- `opc.bin_12` -
- `opc.bin_13` -
- `opc.bin_14` -
- `opc.bin_15` -

Appendix: Housekeeping codes

Power:

- `power.vdcin` - input voltage
- `power.fuel_level` - battery charge level as a percentage
- `power.charge_rate` - current in milliamps (negative is battery discharge rate)

Location (GPS):

- gps.pos_lat - decimal geolocation
- gps.pos_lng - decimal geolocation
- gps.fix_level - degree of assurance of GPS fix. values are 0, 2 or 3
- gps.quality_siv - GPS satellites in view
- gps.quality_hdop - horizontal dilution of precision
- gps.quality_max_snr - maximum signal / noise ratio of satellite communications
- gps.quality_avg_snr - average signal / noise ratio of satellite communications

Location (mobile network):

- phone.mcc - mobile country code
- phone.mnc - mobile network code
- phone.lac - location area code
- phone.ci - cell ID
- phone.bsic - base station ID

Appendix: Sample rates

- basic - the sample rate of the device itself. This rate may vary.
- basicone - one minute average. This will be the actual value is the phenomenon is being sampled every minute. There will be missing values if the sampling frequency is less than every minute.
- avg10min - 10 minute average / minimum / maximum. Currently, this is not available for all devices.
- avg1hr - 60 minute average / minimum / maximum. Currently, this is not available for all devices.

Appendix: Environmental data schema

```

{
  "data_format" : "pv1",
  "location_id" : "300003",
  "location_zone" : "london_croydon",
  "data_rate" : "basic",
  "time" : {
    "utc" : ISODate("2016-04-19T11:03:17.000+0000"),
    "loc" : ISODate("2016-04-19T12:03:17.000+0000"),
    "zone" : "Europe/London",
    "offset" : NumberLong(3600),
    "dst" : "1",
    "day" : "2"
  },
  "sensor_data" : {
    "atmos" : {
      "temp" : 18.1,
      "humid" : 36.6
    },
    "acoustic" : {
      "db" : -32.7
    },
    "opc" : {
      "pm_1" : 6.62,
      "pm_2p5" : 9.09,
      "pm_10" : 13.28,
      "bin_0" : NumberLong(1326),
      "bin_1" : NumberLong(75),
      "bin_2" : NumberLong(47),
      "bin_3" : NumberLong(21),
      "bin_4" : NumberLong(3),
      "bin_5" : NumberLong(4),
      "bin_6" : NumberLong(2),
      "bin_7" : NumberLong(2),
      "bin_8" : NumberLong(0),
      "bin_9" : NumberLong(0),
      "bin_10" : NumberLong(0),
      "bin_11" : NumberLong(0),
      "bin_12" : NumberLong(0),
      "bin_13" : NumberLong(0),
      "bin_14" : NumberLong(0),
      "bin_15" : NumberLong(0)
    },
    "gas" : {
      "co" : {
        "i0" : {
          "ch" : NumberLong(4),
          "val" : 51.0
        }
      },
      "no" : {
        "i0" : {
          "ch" : NumberLong(3),
          "val" : 52.0
        }
      },
      "no2" : {
        "i0" : {
          "ch" : NumberLong(1),
          "val" : 146.0
        }
      },
      "o3" : {
        "i0" : {
          "ch" : NumberLong(2),
          "val" : -39.0
        }
      }
    }
  }
}

```

Appendix: Housekeeping schema

```

{
  "data_format" : "pv1",
  "location_id" : "300003",
  "location_zone" : "london_croydon",
  "time" : {
    "utc" : ISODate("2016-04-19T09:43:21.000+0000"),
    "loc" : ISODate("2016-04-19T10:43:21.000+0000"),
    "zone" : "Europe/London",
    "offset" : NumberLong(3600),
    "dst" : "1",
    "day" : "2"
  },
  "sensor_data" : {
    "power" : {
      "vdcin" : 12.1,
      "fuel_level" : NumberLong(100),
      "charge_rate" : NumberLong(0)
    },
    "gps" : {
      "pos_lat" : 51.349197,
      "pos_lng" : -0.012953,
      "fix_level" : NumberLong(3),
      "quality_siv" : NumberLong(15),
      "quality_hdop" : 0.8,
      "quality_max_snr" : NumberLong(41),
      "quality_avg_snr" : 26.3
    },
    "phone" : {
      "mcc" : NumberLong(234),
      "mnc" : NumberLong(30),
      "lac" : NumberLong(2178),
      "ci" : NumberLong(26238),
      "bsic" : NumberLong(58)
    }
  }
}

```