A short study in fine particles measurement at meteoLCD by inexpensive LLS sensors

Francis Massen francis.massen@education.lu
Claude Baumann claude.baumann@education.lu
Raoul Tholl raoul.tholl@education.lu
Mike Zimmer mike.zimmer@education.lu
Nico Harpes mailto:nico.harpes@ms.etat.lu

Date: 19 November 2018 (version 1.01)

Summary:

The aim of this study is to investigate the capacities of inexpensive fine particle sensors to measure PM10 and PM2.5 mass concentration. Three different types of LLS (laser light scattering) sensors were used: AIRMASTER AM7, IQAir Airvisual Pro and two Nova SDS011 sensors. The three sensors were exposed from 25 September to 22 October 2018 at the meteoLCD location and one supplementary SDS011 at the home of one of the authors. The measurements of the sensors were compared to those of the official Beidweiler station of the Luxembourg Department of Environment which uses a Horiba sensor.
The sensor readings are well correlated to the reference station, agree in the varying concentration patterns but show a more or less important bias.

Index:

1. The equipment

2. Correlations between 3 official PM stations of Luxembourg

3. The results of the test period (26 Sep to 21 Oct 2018)

4. Conclusion and discussion

References
Addendum 1 and 2
History

1. The equipment

1.1. The AIRMASTER AM7 sensor.

This all-in-one device has different sensors to measure PM2.5, PM10, CO_2, T and relative humidity RH. The CO_2 readings will be discarded in this paper. The fine particle sensor is a PMS5003 module from the Chinese company Plantower **[ref.1**]. The device, when powered on, will stream its measurement data at an approximate interval of 4.65s (the interval is not quite constant, and may jitter between 4.50 and 4.70s the long-time average interval between readings being 4.65s); the stream format are normal lines of ASCII text.
A Raspberry Pi mod. 3B nano-computer with an added real-time clock runs a Python script and is used as a data-logger, storing the data-stream as a time-stamped csv file on its micro-SD card. This sensor has been installed on the windows sill of the meteoLCD bureau, with the window constantly open.

The software was written so that the Pi computer automatically creates a new time-stamped file at boot time; after a power loss a new data file will be automatically created without the original data file being corrupted.

The next two sensors are installed in a Stevenson hut mounted on the meteoLCD terrace: (see fig.1 and 2)

1.2. The SDS011A test station

This equipment is based on the Chinese NOVA SDS011 sensor **[ref.2]**; the sensor's output is binary and needs special code for reading. The sensor also accepts binary commands, and depending on the firmware version must be put by a special command into active mode to stream its binary data through a serial port. This sensor too is driven by a Raspberry Pi 3 mod. B, fitted with a Papirus e-ink display which contains a real-time clock. A second SDS011 station (SDS011B) was installed at Bettendorf in the home of one of the authors (F.M.). Both stations were programmed for a sampling interval of 60 seconds.

Numerous examples of Python code found on the Internet do not work as they lack the command to put the sensor into streaming mode. We modified an available script **[ref. 3**] to add time-stamps to the file-name and to every line of a .csv data file.

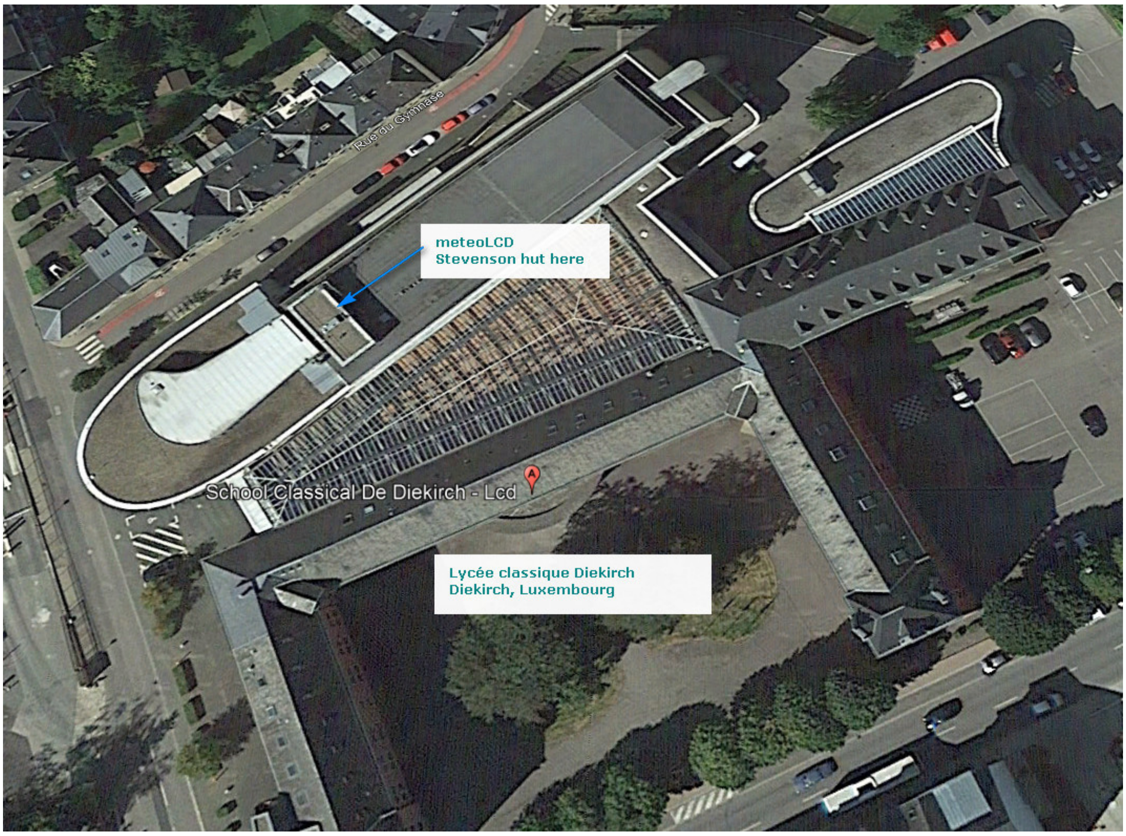

Fig.1: View from above on the meteoLCD terrace with the Stevenson hut.

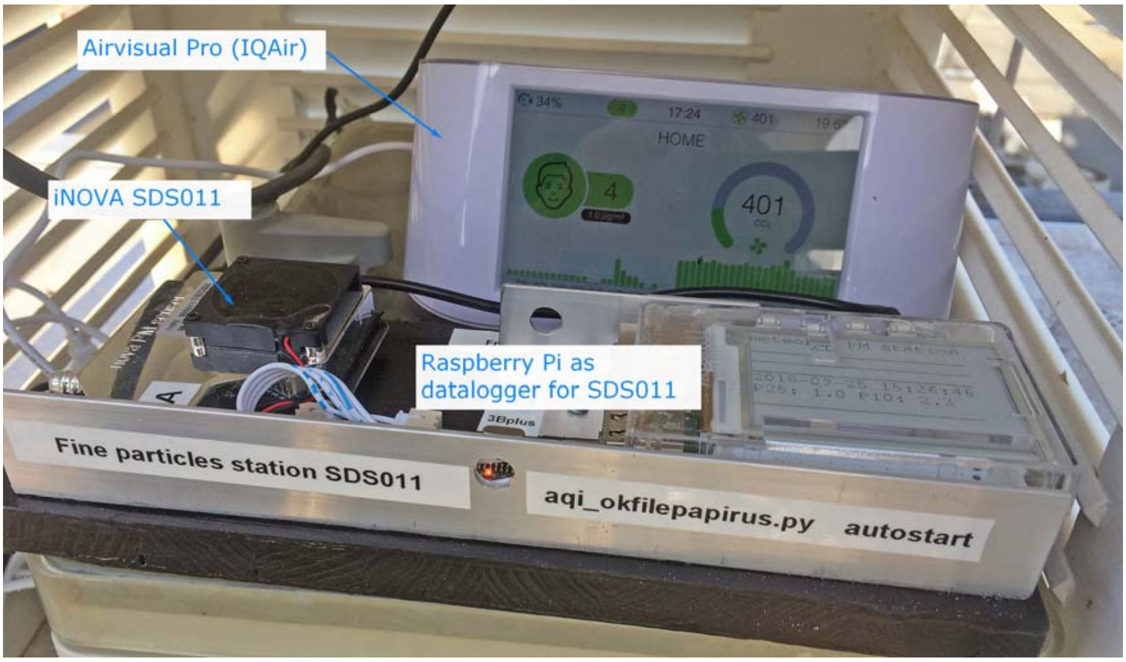

Fig.2: The sensors in the Stevenson hut.

1.3. The Airvisual Pro.

This stylish sensor by the Swiss company IQAir is a standalone device measuring fine particles PM2.5 and PM10, CO_2, temperature and relative humidity RH. The internal PM sensor AVPM25b is of the LLS type and was co-developed by IQAir with an unknown company. The Airvisual Pro uses the A33 quad-core Cortex microprocessor, has its own internal datalogging function (4GB flash storage) and communicates through Wifi [**ref.5**]. The data can be downloaded as a .csv file using the SMB protocol or published live in a Airvisual cloud. The sampling interval is 10 seconds and can not be modified.
A link to a laboratory evaluation by the AQ-SPEC (Air Quality Performance Evaluation Center) is given at [**ref.4]**.

2. Correlation between 3 official PM stations of Luxembourg

As we wanted to compare our PM measurements with those of the official stations of the Environmental Administration (Département de l'Environnement), a first question to be answered is "is such a comparison meaningful?". The closest measuring station to Diekirch is that of Beidweiler, located in a rural environment (Diekirch is assumed semi-rural) at a distance of 18.5 km. The altitude of the meteoLCD terrace at Diekirch is 218m asl, that of Beidweiler 277m asl. The live-readings of the Luxembourg official stations can be found at https://environnement.public.lu/fr/loft/air/mesures/mesures-actuelles.html

The next figure 3 shows the daily PM10 averages from the 18th July to 13th August 2018 at Beidweiler, Bonnevoie and Pl. Churchill. The last two stations are located inside the city of Luxembourg, at about 18 km from Beidweiler. Bonnevoie and Pl. Churchill are classified as urban stations. Despite these different environments one can see that the overall variations seem to be well correlated between Beidweiler and Bonnevoie, and somewhat less between Beidweiler and Pl. Churchill.

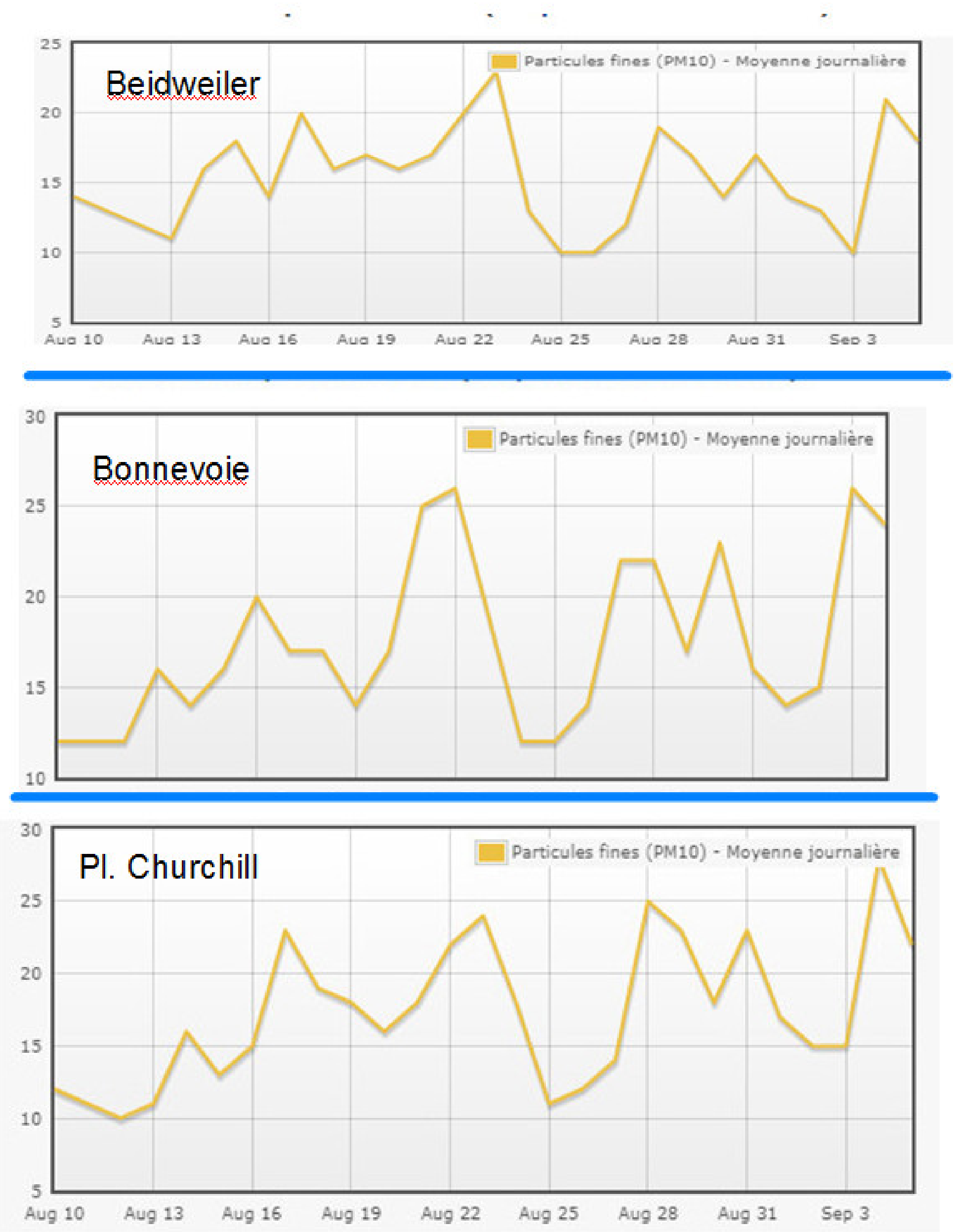

Fig.3: The Beidweiler and Luxembourg-City data series (daily means)

The Pearson correlation coefficients (calculated from the values picked from the plots) between Beidweiler and the two city stations are 0.86 and 0.82, all significant at the 5% level. This relative high correlation is a tell-tale sign that similar regional meteorological conditions induce similar variations of daily mean PM10 concentrations at urban and rural locations. Such a high correlation due to extended pollution plumes is for instance mentioned in **[ref.6]**.
In this paper the Beidweiler station will be taken as the reference to our fine particle measurements done at Diekirch (and at Bettendorf, a rural village situated 4 km from Diekirch). The instrument used at Beidweiler is the Horiba

APDA-371 ambient dust monitor which works by beta radiation attenuation (BAM).

3. The results of the test period (26 Sep to 21 Oct 2018)

The sensors were installed the 25th September and removed the 22th October 2018. We will use the 26 days period from 26 September to 21 October. This period was rather dry with RH varying between 40 and 80% (mean = 55%).
The sampling intervals of all sensors are not the same and are not absolutely constant: about 4.65s for the Airmaster AM7, about 10s for the Airvisual Pro and about 60s for the SDS011 stations. The large data files were handled using the DADISP software package to ravel the data into hourly and daily means. As the information given at the website "environnement.public.lu" do not contain PM2.5 measurements, these were taken from the website of Airvisual https://www.airvisual.com/world-air-quality, the page for Beidweiler being https://www.airvisual.com/luxembourg/grevenmacher/bourglinster/beidweiler. It should be noted that the curves of hourly concentrations at the European AQI site (https://www.eea.europa.eu/themes/air/air-quality-index) correspond to a 24h running mean and not to simple hourly measurements.(personal communication by Pierre Dornseiffer, head of the Administration de l'Environnement, Air et Bruit).

3.1. Time series of daily PM10 means.

Fig.5 (next page) shows that PM10 peaked noticeably the 18th October, and that all sensors, even the SDS011B at Bettendorf home, registered this peak. The SDSA (SDS011A) sensor had three missing data episodes, so the missing data were replaced by interpolation (dashed segments in the plot).

Fig.4 gives the correlations between the different sensors: all are rather high and significant at the 5% level:

	Correlations (dailys_26Septo21Oct18) Marked correlations are significant at p < .05000 N=21 (Casewise deletion of missing data)				
Variable	BEIDWLR_PM10	AV_PM10	AM_PM10	SDSA_PM10interpol	SDSB_PM10
BEIDWLR_PM10	1.00	0.88	0.92	0.70	0.93
AV_PM10	0.88	1.00	0.93	0.82	0.94
AM_PM10	0.92	0.93	1.00	0.76	0.93
SDSA_PM10interpol	0.70	0.82	0.76	1.00	0.66
SDSB_PM10	0.93	0.94	0.93	0.66	1.00

Fig.4: Correlation between different sensors.

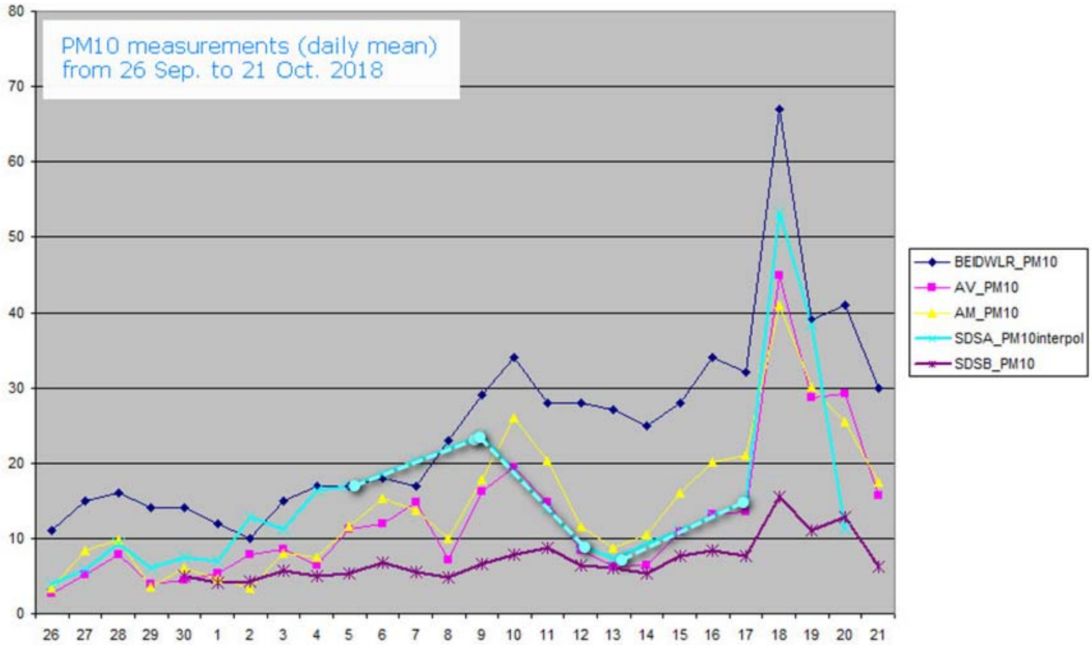

Fig.5: Daily mean PM10 measurements by 4 instruments and Beidweiler (BEIDWLR) reference station. The interpolated segments of the SDSA sensor are dashed.

The correlation between the Airmaster (AM) and the Airvisual (AV) is very high, and both track the changes found at Beidweiler very well. It should be mentioned that the terrace of meteoLCD is about 20m above ground-level (see fig.1), which would suggest somewhat lower readings than in Beidweiler**.** The next figure (Fig.6) gives the calibration lines (linear regression lines) of the three sensors (AM, AV and SDS011A with non interpolated readings) versus Beidweiler:

Obviously the slopes of AV and AM are very close, and all lines nearly parallel. The goodness of the fits ($\mathbf{R^2}$) are (in the same order as the equations given in Fig. 6): 0.80, 0.87 and 0.70.

Conclusion for PM10: the 3 sensors exposed at meteoLCD are well correlated to the readings of Beidweiler. All show the short-time peak during the 18th October. The bias (offsets) are different but rather small. Notice the brown SDSB_PM10 curve in figure 4: even a sensor running inside a house (with only small ventilation slits open to the outside) shows correctly the moment of peak PM10 concentration.

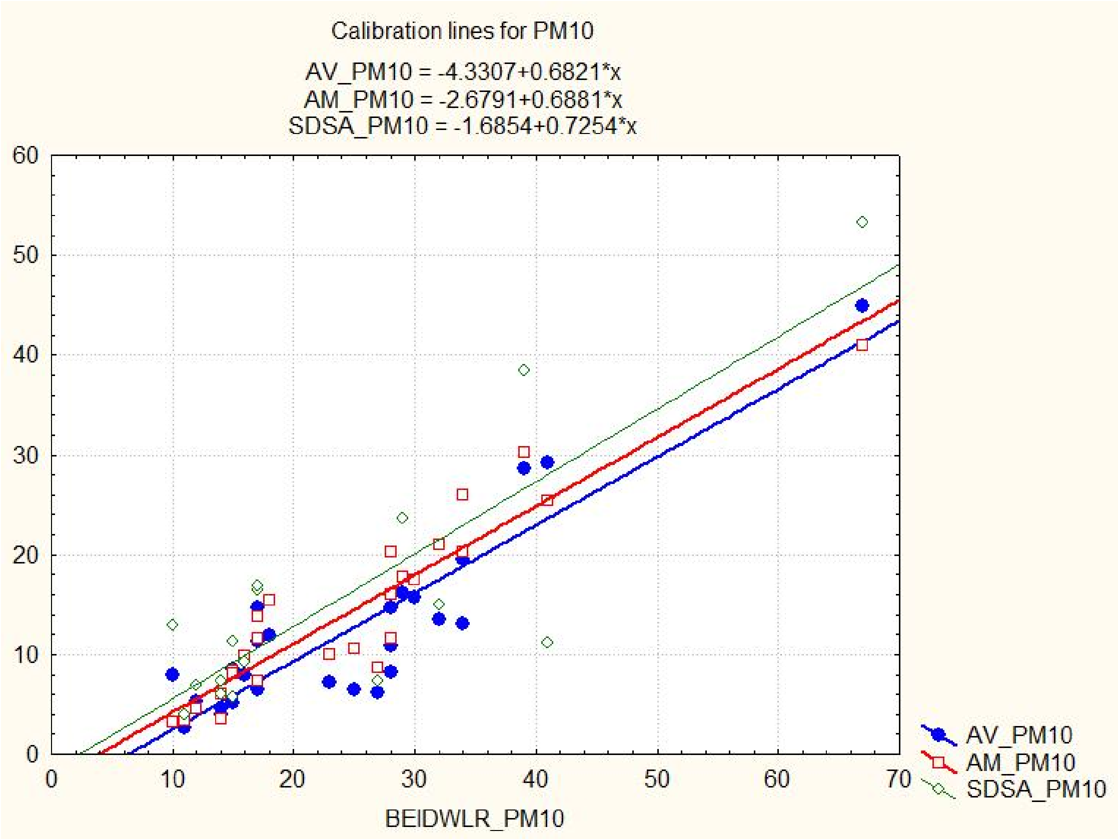

Fig.6: Calibration lines for PM10 (R^2(AV) = 0.80, R^2(AM) = 0.87, R^2(SDSA, only raw data, no interpolation) = 0.70.

3.2. Time series of hourly PM10 means during the peak period.

The next figure shows the hourly mean measurements for the 3 days centered on the peak event (17, 18 and 19 October). The hourly Beidweiler PM10 data are not available neither on the website Emwelt.lu, the Airvisual website nor the EAQI page. All these sites only give the 24h moving averages, but not the raw hourly readings The EEA discomap site (link) where all hourly data are uploaded by the different European AQ stations, has at the time of writing this report no data after the 30th September. So Fig. 7 only shows the readings of the sensors at meteoLCD and that at home in Bettendorf (SDSB):

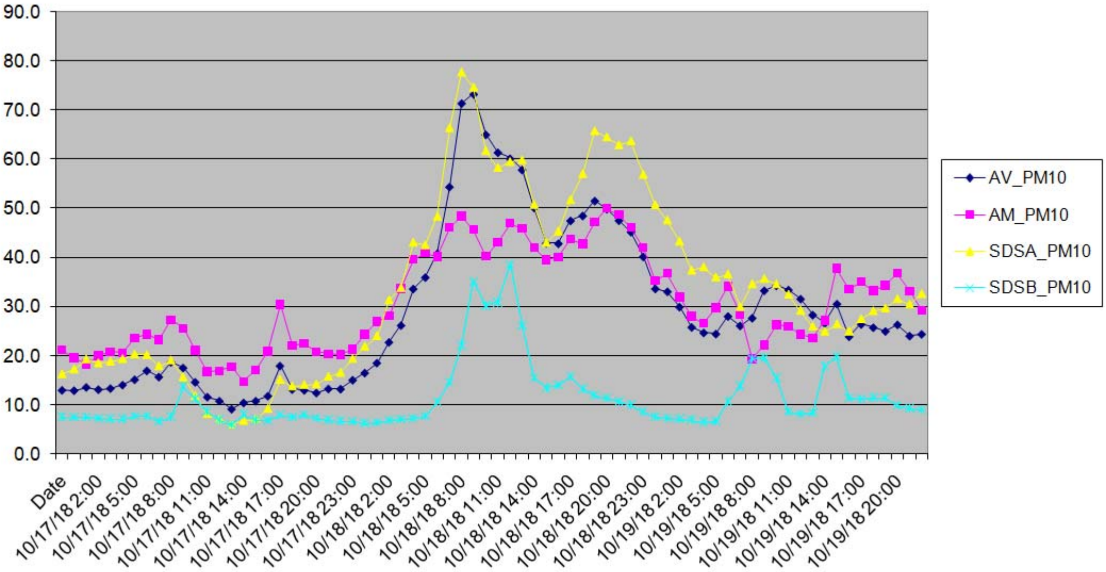

Fig.7: Hourly PM10 readings during the 3 days centered on the peak event.

It seems normal that the readings of the Airmaster located on the windows sill are lower than those of the 2 sensors exposed in the well ventilated Stevenson hut. But all 3 sensors show the same pattern, and the correlation between them is excellent:

Correlations (hourlys_17to19Oct18)
Marked correlations are significant at p < .05000
N=72 (Casewise deletion of missing data)

Variable	AV_PM10	AM_PM10	SDSA_PM10
AV_PM10	1.00	0.89	0.95
AM_PM10	0.89	1.00	0.91
SDSA_PM10	0.95	0.91	1.00

Fig.8: Correlation coefficient between the 3 sensors during the peak event.

The regression line between the Airvisual and SDS011A sensors is given in the next figure:

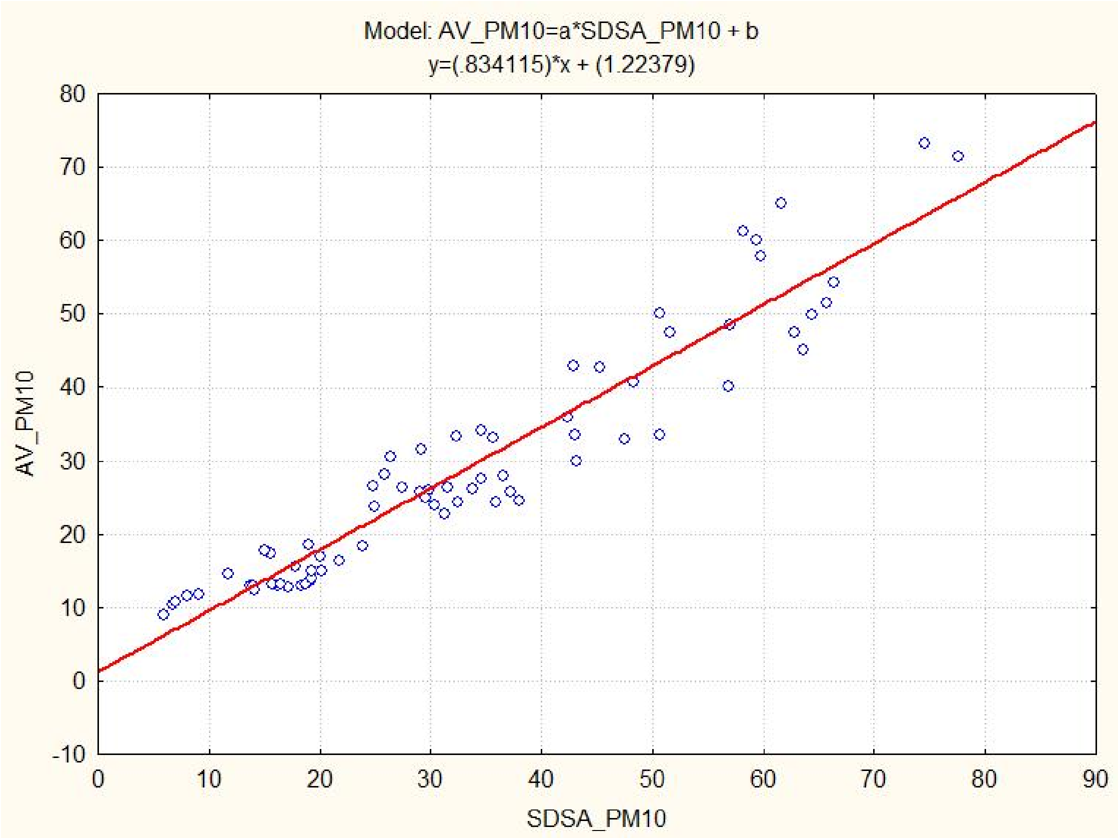

Fig.9: Linear regression between Airvisual and SDS011 measurements during peak event (72 data points).

The Airvisual and SDS011 sensors are the prime candidates to be installed at meteoLCD: the differences between both do not exceed 19 ug/m^3 (about 20%) during the peak event and usually are lower than 10 ug/m^3, the average of the differences being 4.3 ug/m^3.

3.3. Time series of daily PM2.5 means.

The Emwelt.lu website does only show the PM10 measurements, not the PM2.5 concentrations. So these PM2.5 data for Beidweiler must be fetched either from the EAQI, the AIRVISUAL or the discomap sites. The Airvisual website shows the last 30 days, and has been used as the data source for PM2.5.
Fig.10 shows the time series of all measurements (the SDS011A missing data have not been replaced here by interpolation):

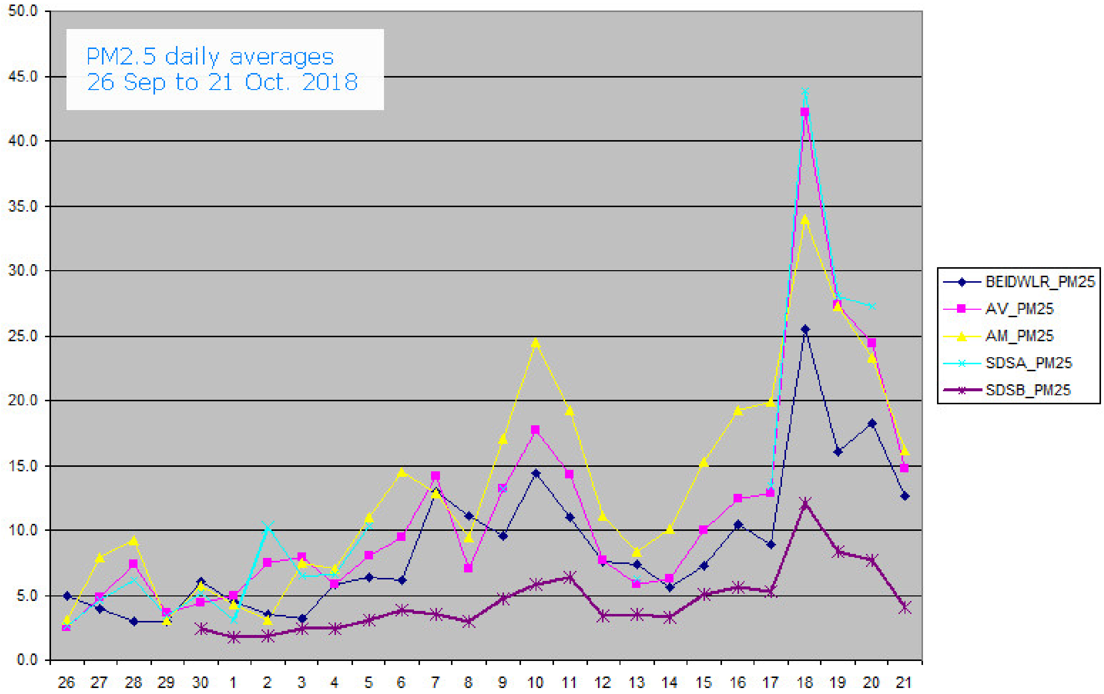

Fig.10: Daily averages of PM2.5 in ug/m3.

As with the PM10 series, all sensors vary more or less synchronously; even the SDS011B sensor running at home in Bettendorf shows the two peaks during the 10th and 18th October, but its general profile seems to be smoothed out in comparison to the sensors exposed in the open.

The correlations between the different sensors are even better than those found for the PM10 data, as shown on the next figure 11:

	Correlations (dailys_26Septo21Oct18.sta) Marked correlations are significant at p < .05000 N=12 (Casewise deletion of missing data)				
Variable	BEIDWLR_PM25	AV_PM25	AM_PM25	SDSA_PM25	SDSB_PM25
BEIDWLR_PM25	1.00	0.97	0.94	0.97	0.98
AV_PM25	0.97	1.00	0.95	0.99	0.98
AM_PM25	0.94	0.95	1.00	0.94	0.98
SDSA_PM25	0.97	0.99	0.94	1.00	0.98
SDSB_PM25	0.98	0.98	0.98	0.98	1.00

Fig.11: Correlation coefficients between the 5 sensors (daily means).

Taking the Beidweiler readings as x-values, the 3 linear regressions with the sensors at meteoLCD are shown in the next figure 12.

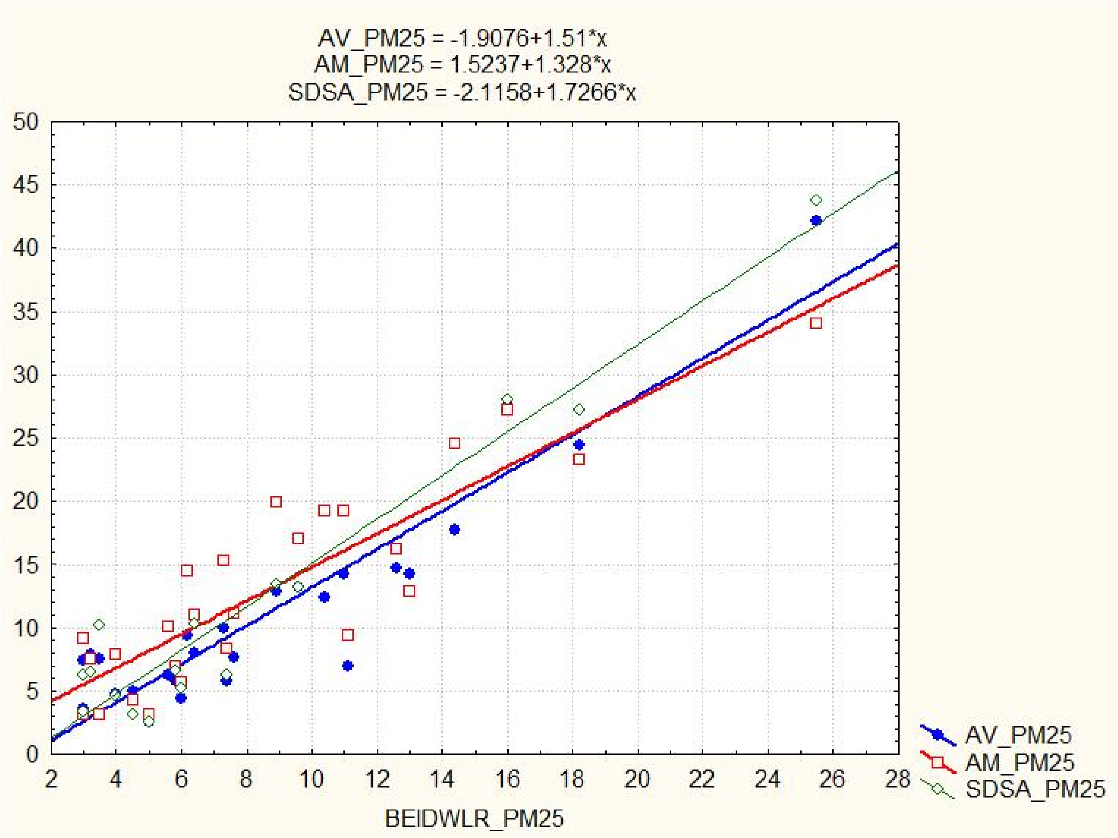

Fig.12: Calibration lines for PM2.5 (R^2(AV) = 0.87, R^2(AM) = 0.79, R^2(SDSA, only raw data, no interpolation) = 0.97

Whereas the PM10 readings of Beidweiler were most of the times higher than those of the other sensors, the situation here is opposite: the AV, AM and SDSA readings are practically always higher than those of Beidweiler.

4. Conclusion and discussion

The aim of the test was to check if inexpensive LLS fine particle sensors are useable for air pollution measurements. The approximate prices of these sensors are the following:

Airvisual Pro:	259 Euro
Airmaster AM7	159 Euro
SDS011 based sensors:	130 Euro (sensor: 40, RPi 90 with case/display)
Horiba ADPA-371	~**20000** Euro [**ref.7**]

So we compare here equipments that are two order of magnitude less expensive than the professional Horiba BAM device!

The EEA accepts a large uncertainty of +/-25% for fine particle measurements. That number can not be guaranteed by the LLS sensors, but the main result to retain is that these sensors are able to reproduce the time pattern of dust contamination, and that their measurements are usually well correlated to that of the reference station in this test.

Several factors are limiting the accuracy and repeatability of the LLS sensors:

- the air flow is produced by a small fan, so its magnitude is not constant and varies with atmospheric air pressure and wind conditions

- the air is not dried, which means that the LLS sensors tend to have too high readings due to the accumulation of condensation on the air particles when relative humidity is high [**ref. 8, 11**]. As the SDS011 and Plantover sensors were found in the first of these papers to be effected when relative humidity levels exceed 75%, **our test series should be considered as "dry" and unaffected by RH.**
  Fig. 13 shows the PM10 and RH measurement of the Airmaster (Plantover sensor): no systematic correlation between RH and PM10 can be seen. (see also the addendum 2)

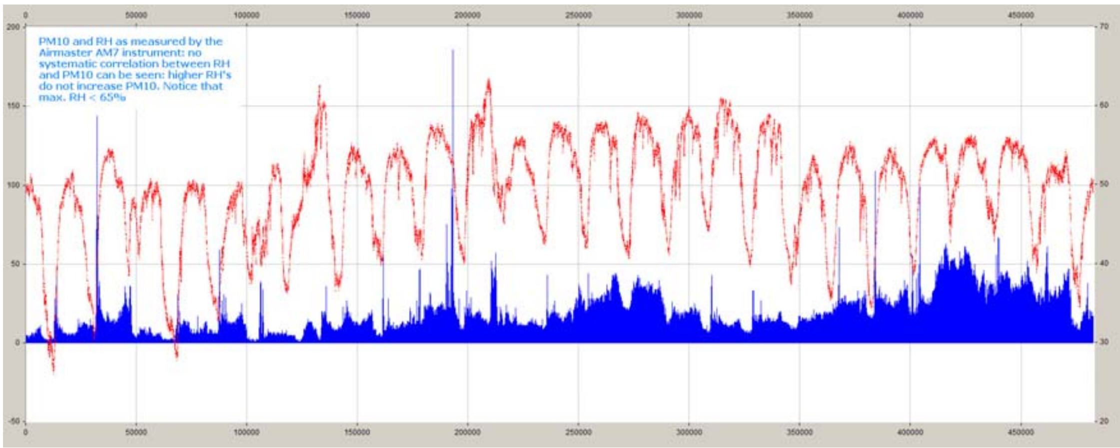

Fig.13: PM10 (blue) and RH (red) data from the Airmaster AM7: no visible correlation!

- the laser scattering principle is a counting procedure, and the mass concentration must be deduced by assuming a certain density and reflectance of the particles. These calculations are made by the embedded firmware of the LLS sensors, and are normally not documented to protect their intellectual propriety. The BAM and microbalance methods directly deliver a mass per volume.

The following figure shows the **absolute** differences between the PM10 measurements at the Beidweiler station and the three sensors exposed at meteoLCD. As the meteoLCD sensors were located about 20m above ground level and the Beidweiler probably close to ground level, a normal negative bias w.r. to Beidweiler seems plausible (but conflicts with the opposite sign bias for PM2.5). This would reduce the maximum difference during the peak event of 18th October to an acceptable magnitude.

The LLS sensors could deliver valid results during peak events, and the small absolute differences during low concentration situations might be tolerable.

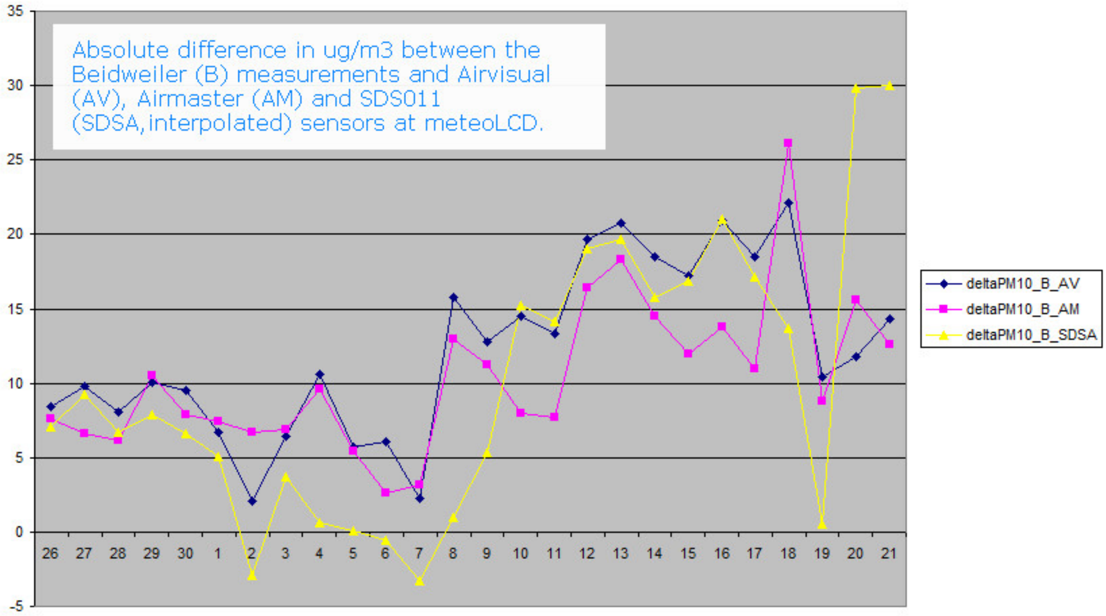

Fig.14: Absolute differences between Beidweiler and the 3 sensors at meteoLCD

These very inexpensive sensors are more than a gimmick or a gadget. They can deliver useful results but their intrinsic limitations should always be kept in mind, and they should not be used to verify if AQ standards are met. [**ref. 8, 9**].

As this test series was made during rather a low RH situation, a second study is planned for a period of very high RH or foggy meteorological conditions. High RH is reported in many papers as the most important factor influencing LLS measurement error; the question if a correction for high RH is possible or not still has no definitive answer **[ref.8, 12]**

There is a large ongoing popular movement for "citizen science measurements", aiming to install a great number of inexpensive sensors to monitor air quality, see for instance the German "Luftdaten" initiative **[ref.10]**. Making comparative tests between these grass-root stations and the official ones is a prerequisite to their acceptance**.**

References:

1. The Plantower PMS5003 and PMS7003 Air Quality Sensor experiment: http://aqicn.org/sensor/pms5003-7003/de/

2. Nova SDS011 sensor. http://aqicn.org/sensor/sds011/

3. AQI.py code used: see addendum 1.

4. AQ-SPEC: Laboratory evaluation of the Airvisual Pro. http://www.aqmd.gov/aq-spec/evaluations/laboratory

5. Airvisual inside: https://wiki.liutyi.info/display/CO2/AirVisual+inside:

6. Chunshui Lin et al.: Extreme air pollution from residential solid fuel burning. Sep. 2018, DOI: 10.1038/s41893-018-0125-x

7. Etude 3/2-2011 Mesure des particules en suspension dans l'air. LCSQA 2011 (link)

8. Jayaratne R. et al.: The influence of humidity on the performance of low-cost air particle sensors, 2018. Atmos. Meas. Tech., 11, 4883–4890, 2018 (link)

9. Ben Kinh Tan: Laboratory Evaluation of Low to Medium Cost Particle Sensors. Master Thesis, University of Waterloo, Ontario, Canada. 2017 (link)

10. luftdaten.info

11. Dunea D. et al.: Relationship between airborne particulate matter and weather conditions in Targoviste urban area during cold months. *Rev. Roum. Chim.*,2015, *60*(5-6), 595-

12. Laquai B. Kompensation des Feuchte-Effekts bei Low-Cost Feinstaub-sensoren nach dem Streulichtverfahren www.opengeiger.de/Feinstaub/FeuchteKompensation.pdf

Addendum 1

Python script to read SDS011 sensor and store the time-stamped data:

```
#!/usr/bin/python
# coding=utf-8
# "DATASHEET": http://cl.ly/ekot
# https://gist.github.com/kadamski/92653913a53baf9dd1a8
#
# adaptation by F.Massen  02-July-2018
#   1. disable all json stuff
#   2. add time  to screen output
#   3. add  logging data to file where filename holds datetime
#   03 July 2018
#   filename is aqi_okfile.py
#--------------------------------------------------------------

from __future__ import print_function
import serial, struct, sys, time, json, datetime, os

DEBUG = 0
CMD_MODE = 2
CMD_QUERY_DATA = 4
CMD_DEVICE_ID = 5
CMD_SLEEP = 6
CMD_FIRMWARE = 7
CMD_WORKING_PERIOD = 8
MODE_ACTIVE = 0
MODE_QUERY = 1

ser = serial.Serial()
ser.port = "/dev/ttyUSB0"
ser.baudrate = 9600

ser.open()
ser.flushInput()

byte, data = 0, ""

def dump(d, prefix=''):
    print(prefix + ' '.join(x.encode('hex') for x in d))

def construct_command(cmd, data=[]):
    assert len(data) <= 12
    data += [0,]*(12-len(data))
    checksum = (sum(data)+cmd-2)%256
    ret = "\xaa\xb4" + chr(cmd)
    ret += ''.join(chr(x) for x in data)
    ret += "\xff\xff" + chr(checksum) + "\xab"

    if DEBUG:
        dump(ret, '> ')
    return ret

def process_data(d):
    r = struct.unpack('<HHxxBB', d[2:])
```

```
    pm25 = r[0]/10.0
    pm10 = r[1]/10.0
    checksum = sum(ord(v) for v in d[2:8])%256
    return [pm25, pm10]
    #print("PM 2.5: {} Î¼g/m^3  PM 10: {} Î¼g/m^3 CRC={}".format(pm25,
pm10, "OK" if (checksum==r[2] and r[3]==0xab) else "NOK"))

def process_version(d):
    r = struct.unpack('<BBBHBB', d[3:])
    checksum = sum(ord(v) for v in d[2:8])%256
    print("Y: {}, M: {}, D: {}, ID: {}, CRC={}".format(r[0], r[1],
r[2], hex(r[3]), "OK" if (checksum==r[4] and r[5]==0xab) else "NOK"))

def read_response():
    byte = 0
    while byte != "\xaa":
        byte = ser.read(size=1)

    d = ser.read(size=9)

    if DEBUG:
        dump(d, '< ')
    return byte + d

def cmd_set_mode(mode=MODE_QUERY):
    ser.write(construct_command(CMD_MODE, [0x1, mode]))
    read_response()

def cmd_query_data():
    ser.write(construct_command(CMD_QUERY_DATA))
    d = read_response()
    values = []
    if d[1] == "\xc0":
        values = process_data(d)
    return values

def cmd_set_sleep(sleep=1):
    mode = 0 if sleep else 1
    ser.write(construct_command(CMD_SLEEP, [0x1, mode]))
    read_response()

def cmd_set_working_period(period):
    ser.write(construct_command(CMD_WORKING_PERIOD, [0x1, period]))
    read_response()

def cmd_firmware_ver():
    ser.write(construct_command(CMD_FIRMWARE))
    d = read_response()
    process_version(d)

def cmd_set_id(id):
    id_h = (id>>8) % 256
    id_l = id % 256
    ser.write(construct_command(CMD_DEVICE_ID, [0]*10+[id_l, id_h]))
    read_response()

if __name__ == "__main__":
```

```
#    outputFilePath = os.path.join(os.path.dirname(__file__),
datetime.datetime.now().strftime("%Y-%m-%d_%H.%M.%S") + ".csv")
    filename = "SDS011_" +
str(datetime.datetime.now().strftime("%Y_%m_%d_%H_%M_%S")) + ".csv"
    while True:
        cmd_set_sleep(0)
        cmd_set_mode(1);
        for t in range(15):
            values = cmd_query_data();
            if values is not None:
                now = datetime.datetime.now()
                print(now.strftime("%Y-%m-%d %H:%M:%S"), " ... ",
"PM2.5: ", values[0], ", PM10: ", values[1])
                #f = open("SDS011.csv", 'a+')
                f = open(filename, 'a+')
            dt0 = now.strftime("%Y-%m-%d")
                dt1 = now.strftime("%H:%M:%S")
                v0 =  "%.3f" % values[0]
                v1 =  "%.3f" % values[1]
                f.write(dt0 + ", " + dt1 + ", " + v0 + ", " + v1 +
"\n")
                f.close
            time.sleep(60)

#        # open stored data
#        with open('/var/www/html/aqi.json') as json_data:
#            data = json.load(json_data)

#        # check if length is more than 100 and delete first element
#        if len(data) > 100:
#            data.pop(0)
#
#        # append new values
#        data.append({'pm25': values[0], 'pm10': values[1], 'time':
time.strftime("%d.%m.%Y %H:%M:%S")})
#
#        # save it
#        with open('/var/www/html/aqi.json', 'w') as outfile:
#            json.dump(data, outfile)
#
#        print("Going to sleep for 5min...")
#        cmd_set_mode(0);
#        cmd_set_sleep()
#        time.sleep(300)
```

Addendum 2

A simple experiment to detect the influence of humidity on SDS011 sensor

A standalone SDS011 sensor with an OLED display module is used in a dry box (RH = 45%, measured by the small SNDWAY sensor). Note the PM2.5 and PM10 readings of 3.8 and 3.9 ug/m^3.

Fig.15: Dry box: RH = 45%, PM2.5 = 3.8, PM10 = 3.9 ug/m3

Now a wet tissue is put into the box . After about 5 minutes RH rises to 74%, but the PM2.5 and PM10 remain practically the same.
This is an opposite behavior to papers discussing a positive correlation between RH and PM. Actually many authors like Jayaratne **[ref.8]** find that this positive influence only kicks in at a certain level (about 75%) and at visible fog.

This small experiment shows that the Nova SDS011 sensor is not influenced by **moderate** humidity levels. BTW it also shows that the readings of the small SNDWAY PM2.5 sensor (using a Plantover LLS device) are visibly different at these low levels.

Fig.16. A damped green tissue increases the relative humidity in the box up to 74%, but the PM levels remain more or less the same.

History:

version 1.0	08 Nov 2018:	original version
version 1.01	19 Nov 2018:	missing pictures in PDF version corrected minor typos corrected file name now has date of revision