COMPUTING SUBJECT:	**Socket programming**
TYPE:	Assignment
IDENTIFICATION:	UDPSensor
COPYRIGHT:	*Michael Claudius*
LEVEL:	Intermediate
TIME CONSUMPTION:	1 hours
EXTENT:	20 lines
OBJECTIVE:	UDP-sockets receiving data from a sensor
PRECONDITIONS:	Computer Networks Ch. 2.7
COMMANDS:	

IDENTIFICATION: UDPSensorPollution

The Mission

We are shall explore receiving data broadcasted on the net from a pollution sensor placed in the teachers room. The sensor is continuously sending out data on the local LAN.

Domain description

At the school at Roskilde train station is set up a sensor, which can measure the important pollution factors: CO(measured in mg), NOx(measured in ug) and small particles concentration(Alarm, High, Normal, Low). These data are sent out on the nearby school-LAN by UDP-broadcasting on port 9000. A data set has the following format:

```
Pollution sensor v.1.0. \r\n
Location: Ro's Torv
Time: 04-Oct-17 12:34:42 PM
CO: 0.38 \r\n
NOx: 140.30 \r\n
Particle level: Medium \r\n \r\n
```

Right now you have a solution for UDP number receiving: UDPNumberReceiver and UDPEchoClient, which are good starting points for this assignment.

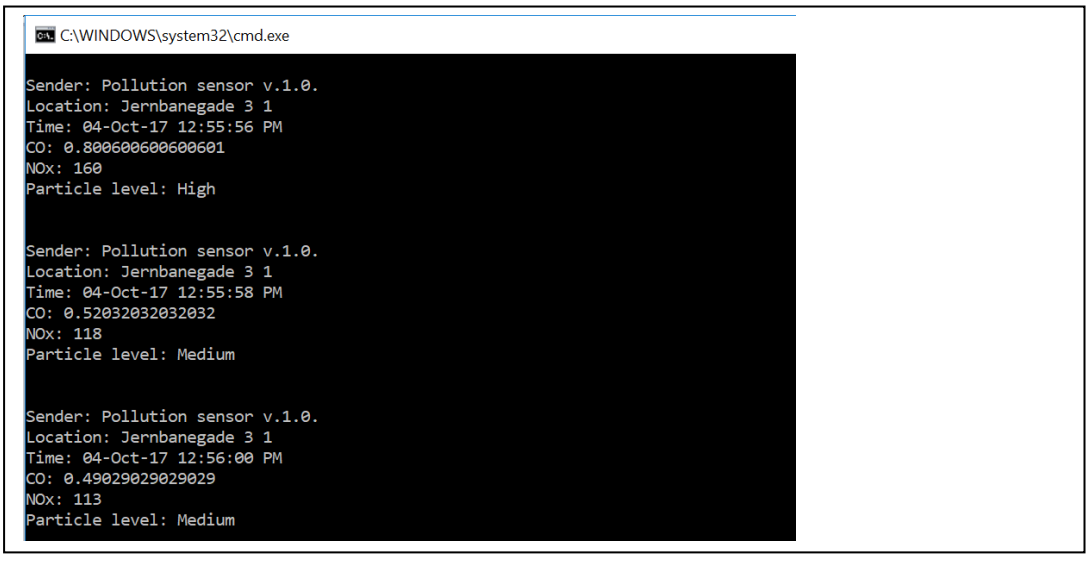
Assignment 1: Application class: UDPSensorRecceiver

a. Create a new Console Application project, UDPSensorReceiver, with the usual main method, which apply UdpClient for listening on port 9000.

b. Extend the program to:
   - Capture sensor data and print them to the console.
   - Split the data into the individual text-lines values and print them to the console.
   - Save the numerical values of CO and NOx in appropriate variables.

Print out these variables as well.

When running you should receive some data like:



Depending of course what you decide to print out.

Assignment 2: Analyzing data
Extend the program so it is calculating the sum of the measured numerical values inside the while-loop.

Assignment 3: Stop the receiver
Create a new project UDPSensorStop, which sends a "Stop" command to the port on which the receiver is listening. When stop-command is received, the receiver's while-loop is stopped and it print out the sum and average of the measured values.