

Pollution Gathering: Real time Pollution Map

Zuzana Kosová¹

¹ Czech Technical University in Prague, Czech Republic

Abstract: *This article describes how to create a widespread system of special sensors using smart phones or smart watches to generate a real-time air pollution map based on data gathered by these sensors. We developed an application for mobile devices, which communicates with sets of sensors and sends preprocessed data to a server via data transfer. Static units containing different sensors are distributed in a few areas and send the gathered data to the server. In addition, mobile units with sensors send measured values to mobile devices. Pollution Gathering application runs on these mobile devices, collects and forwards data to the server, adding the location of the device. All these data are processed at a server and offered to client applications creating online maps with recorded values. The application is capable of sending the user subjective observations in the form of filled-in questionnaires with photos included. This system of sensors is expected not only to spread out in the Czech Republic but in the whole Europe or other continents. The concept of this project is to use the collected data in different ways – navigations taking into account of pollution, predictions, creating statistics etc.*

Keywords: *sensors, realtime map, mobile, air pollution, BLE, bluetooth low energy*

1. Introduction

In recent years, a huge expansion of smart phones and their complementary devices has arisen. These wearable devices are connected to phones or tablets via a smart Bluetooth protocol. They collect data and send them to servers through phones or tablets processing inputs into human readable outputs.

There are many companies or internet servers that can evaluate data from these sensors and create a map of air pollution in the area. In the Czech Republic, the most accurate data comes from the Czech Hydrometeorological Institute or Geoportal Prague. However, all of these maps are static and do not change when data from sensors differs from the preceding samples. The information can be several hours old and as a consequence is not exact. In certain cases it is crucial to receive the pollution information constantly.

Our development team constructed units containing sensors which measure certain physical quantities as content of SO₂ or CO₂ in the air. Afterwards, these sensors send measured values via special packets along with information about the units' state to paired mobile devices, which can be phones or watches. Devices participate in the project by running the application called PollutionGathering. Immediately after the application starts, it tries to connect to all authorized units. If no authorized units are accessible, users have the option to use manual BLE (Bluetooth Low Energy) devices to scan and add any of units within Bluetooth range to the list of authorized units. Afterwards, mobile devices try to connect to these units, read their data and establish special flags informing PollutionGathering about any change in the measured values. If the mobile device receives impulses that something has changed, it displays the updated values.

Measured values are sent to servers in periodical intervals that can be changed according to the application's settings. Data is saved in the database and evaluated for all interfaces displaying real-time maps. Users can complete and send a predefined questionnaire and add photos or notes to each question.

This application can be crucial for creating online maps. It can participate in data gathering for navigation systems for cars or bikes. It depends on units for distribution, but can cover the whole of the Czech Republic and in future it is envisioned to cover Europe. Bikers and drivers will be allowed to choose their route in accordance with the displayed pollution values and thus they will be able to avoid areas with increased air contamination.

This article describes how to create a widespread system of the special sensors that use smart phones or smart watches to generate a real-time air pollution map based on data gathered by these sensors. Consequently, this map can be used for real-time navigation which takes the air pollution into account (e.g. for cyclists or pedestrians).

2. Pollution Gathering

2.1. Definition

The goal of the project is to build a system collecting data from sensors in the wide area. All measured values are transferred to the central node (server). Furthermore, that server cleans the data from noise and fills the database of with all the measured values. This database is accessible to the subsequent processing through its application programming interface (API).

2.2. System Topology

Figure 1 shows the system’s topology. There are two types of units – static and mobile. Static units are placed in predefined locations and communicate with the gateway via a wireless network. The server is informed when their states change and receives the measured data. Mobile units are expected to be attached to the user’s clothes or to cars, bikes etc. These units communicate with mobile devices such as smart mobile phones and these mobile devices provide all the communication with the central server.

In this paper, the mobile device part will be described in detail.

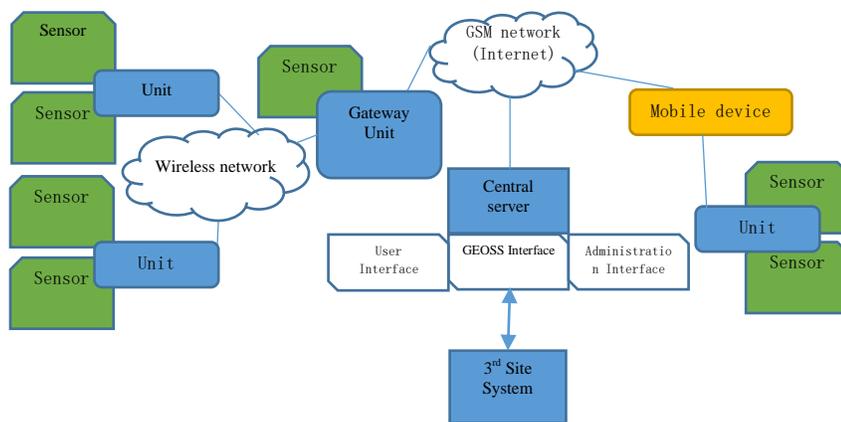


Fig. 1. The system’s topology

The android application is called Pollution gathering. It is intended to run in smartphones, some smartwatches or tablets which fulfill requirements described in Table 1.

Software requirements	Android 4.3. (API Level 18)
Hardware requirements	Bluetooth Low Energy
	Data transfer (GPRS, 3D or LTE)
	GSM
Units description	Camera
	Containing sensors measuring physical quantities as SO ₂ or CO ₂ in the air
	Bluetooth Low Energy

The installation requires some special permissions as access to GPS, Internet, Bluetooth or phone state reading. After users approve these permissions, the application PollutionGathering is installed into the device.

2.3. Description of the Application

After the application is started, it checks whether data service, GPS position and Bluetooth Low Energy module are enabled. If not, the user is allowed to enable it. Afterwards, it tries to connect automatically to all authorized units via BLE protocol. If there are no authorized units or none of them are within the device's range, the screen for BLE scanning is displayed. Providing any units are found, user authorizes them by touching the record and the application tries to connect to this unit.

A unit contains up to three sensors. To get data from three sensors, the application has to get addresses of individual sensors from Bluetooth GATT bundle. GATT is an acronym for the Generic Attribute Profile, and it defines the way that two Bluetooth Low Energy devices transfer data back and forth using concepts called Services and Characteristics. It makes use of a generic data protocol called the Attribute Protocol (ATT), which stores services, characteristics and related data in a simple lookup table using 16-bit IDs for each entry in the table [1].

Bluetooth Low Energy GATT transactions are based on objects called profiles, services and characteristics. Profiles are sets of services (e.g. heart rate profile) and services, distinguished by unique identifier UUID, containing characteristics. The characteristic encapsulates a data like heart rate meter, accelerometer etc. The Pollution Gathering Application has to read characteristics with UUID 0x29 (see Table 2) and its value handles the list of all registered sensors. This list is created by first 4 bytes from 6 bytes, each 2 bytes characterise the single sensor address. As a result, units are limited to have a maximum of 2 sensors.

TABLE II: HANDLES AND UUIDS OF SENSORS IN THE UNIT

Handle	UUID	UUID Description	Value
0x0023	0x2800	GATT Primary Service Declaration	F0:FF
0x0024	0x2803	GATT Characteristic Declaration	0A:25:00:F1:FF
0x0025	0xFFF1	Simple Profile Char 1	05:05:01:00
0x0026	0x2901	Characteristic User Description	Period of the measuring
0x0027	0x2803	GATT Characteristic Declaration	10:28:00:F2:FF
0x0028	0xFFF2	Simple Profile Char 2	DB:7A:94:3F
0x0029	0x2902	Client Characteristic Configuration	00:00
0x002A	0x2901	Characteristic User Description	Sensor 1
0x002B	0x2803	GATT Characteristic Declaration	10:2C:00:F3:FF
0x002C	0xFFF3	Simple Profile Char 3	F0:FF:7E:43
0x002D	0x2902	Client Characteristic Configuration	00:00
0x002E	0x2901	Characteristic User Description	Sensor 2
0x002F	0x2803	GATT Characteristic Declaration	10:30:00:F4:FF
0x0030	0xFFF4	Simple Profile Char 4	EA:00:26:42
0x0031	0x2902	Client Characteristic Configuration	00:00
0x0032	0x2901	Characteristic User Description	Battery level
0x0033	0x2803	GATT Characteristic Declaration	02:34:00:F5:FF
0x0034	0xFFF5	Simple Profile Char 5	00:FF:AA:00
0x0035	0x2901	Characteristic User Description	List of sensors

After the application reads the value representing the list of active sensors, it parses its bytes and gets 1 or 2 sensor IDs. For each non zero sensor ID, it tries to get the GATT characteristic with a predefined address (0xFFF2 for sensor 1, and 0xFFF3 for sensor 2, see Table II, UUID for Simple Profile Char). For each sensor, we load the name and unit for the measured quantity and display it to the user together with the battery state. The list of names and units for quantities is downloaded from the server after the application is started. Afterwards, the application gets the GATT descriptor for notifications and enables it. Whenever the value for this characteristic changes, the unit sends a notification to the application which is able to react in an appropriate way – display and store the current measured value. In predefined interval, changeable in the application's settings, last measured values from each sensor are sent to the server. The format of this packet is described in the following section – 2.4 Server packet format.

2.4. Server packet format

The format of the data packet containing measured values is described in the Table 3. The header contains 11 bytes – Note, Unit ID, Current time and number of sensors.

TABLE III: PACKET FORMAT

Quantity	1x	1x	1x	1x	(1-n) x	
Size	2B	2B	6B	1B	2B + 4B	
Variable	Note	Unit ID	Current time	Nr of sensors	Sensor ID	Value
Range	DM	0x0000-0xFFFF	yymmddhhmmss	0-255	0-255	float (data type)

In the last part of the packet, there is an array containing doubles with sensor IDs and its values for each sensor. Except from measured values, there are some mandatory fields which must be filled: e.g. latitude, longitude, the unit's battery state and the mobile device's battery state with predefined IDs.

2.5. Questionnaire

The special feature of the application is that the user can fill in the predefined questionnaire with questions requiring subjective answers.

The example is: "Is there a noticeable smog around?" The answer can be Yes/No and the user is allowed to add a camera image to the question in order to support their reply.

If there are any unexpected situations not covered in the questionnaire, users can describe them in the Note field. Afterwards, the questionnaire with images is sent to the server by pressing the Send button. According to the server's answer, the user gets a response of the upload.

The questionnaire can be modified by loading the definition file in XML format with new questions. Except for the text, each question contains a unique ID. On that account, the server can process answers correctly even when the questionnaire is changed. It is also possible to change the questionnaire's language simply by reloading of the right questionnaire.

Here follows the description of XML file defining the questionnaire:

```

<questionForm>
  <groups>
    <group>
      <name>Question group's name</name>
      <questions>
        <question>
          <text>Question's text</text>
          <questionCode>Question's ID</questionCode>
        </question>
        <question>
          <text>Another question's text</text>
          <questionCode>Another question's ID</questionCode>
        </question>
        ...
      </questions>
    </group>
    <group>
      <name>Question group's name</name>
      <questions>
        <question>
          <text>Question's text</text>
          <questionCode>Question's ID</questionCode>
        </question>
        ...
      </questions>
    </group>
  </groups>
</questionForm>

```

Questions related to the same area are grouped together. In the application, each group of questions can be collapsed to the single line or expanded to the list of questions, thus users can affect the questionnaire's appearance.

2.6. Offline mode

Sometimes it is not possible to send real time data to the server. In this case users can switch on the offline mode. If it is not feasible to use data transfer, users can decide whether to use the offline mode and store the measured data into the file or merely display them and not to store them.

After the application has started again, it checks whether there are any new data to be sent to the server. In online mode it sends them to the server in a batch before it starts to search for BLE devices. Packets contain also the measuring time value, thus data will be processed additionally in the server. Therefore pollution maps can get more exact retrospectively.

3. Conclusion

The system of a great number of sensors located in the terrain is the method of the future and it will definitely substitute all systems with smaller amount of moving sensors which gather the data within the range of their route. The acquisition price of sensors (and subsequently of units composed of sensors) is already very low and the long-term trend shows further price decreasing, that is why it will not be that much money-consuming to build the network of sensors. The accuracy of this system depends on the sensors density and can be increased by adding these sensors to the existing sensor network.

Difference between the PollutionGathering and other existing applications as [2] or [3] is that the application developed at our University uses static as well as mobile sensors. Therefore it is possible to manage and organize them to get data from required positions.

PollutionGathering application can be run in any android device with given prerequisites fulfilled (see Table 1) – in mobile phones or also in smart watches. Users are also expected to wear an accessories containing units with sensors. Our prototypes of units are placed in boxes 7cm x 10cm. However, our goal is to develop units as accessories like a band or a little box which fits in the pocket. Therefore it will be really mobile and users will not be expected to change their habits. All they have to do is to wear the bands on their wrists or put the boxes into their pockets, install the application into their smart device and run it.

4. References

- [1] K. Townsend. (2014). GATT – Introduction to Bluetooth Low Energy, <https://learn.adafruit.com/introduction-to-bluetooth-low-energy/gatt>
- [2] Air Pollution in Asia: Real-time Air Quality Index Visual Map, <http://aqicn.org/map/>
- [3] European Environment Agency. Unvalidated real-time air quality map, <http://www.eea.europa.eu/themes/air/air-quality/map/real-time-map>