MindPollution

1 Introduction

2 Implementation

Figure 1 shows an overview of the architecture of our project, showing the main components.

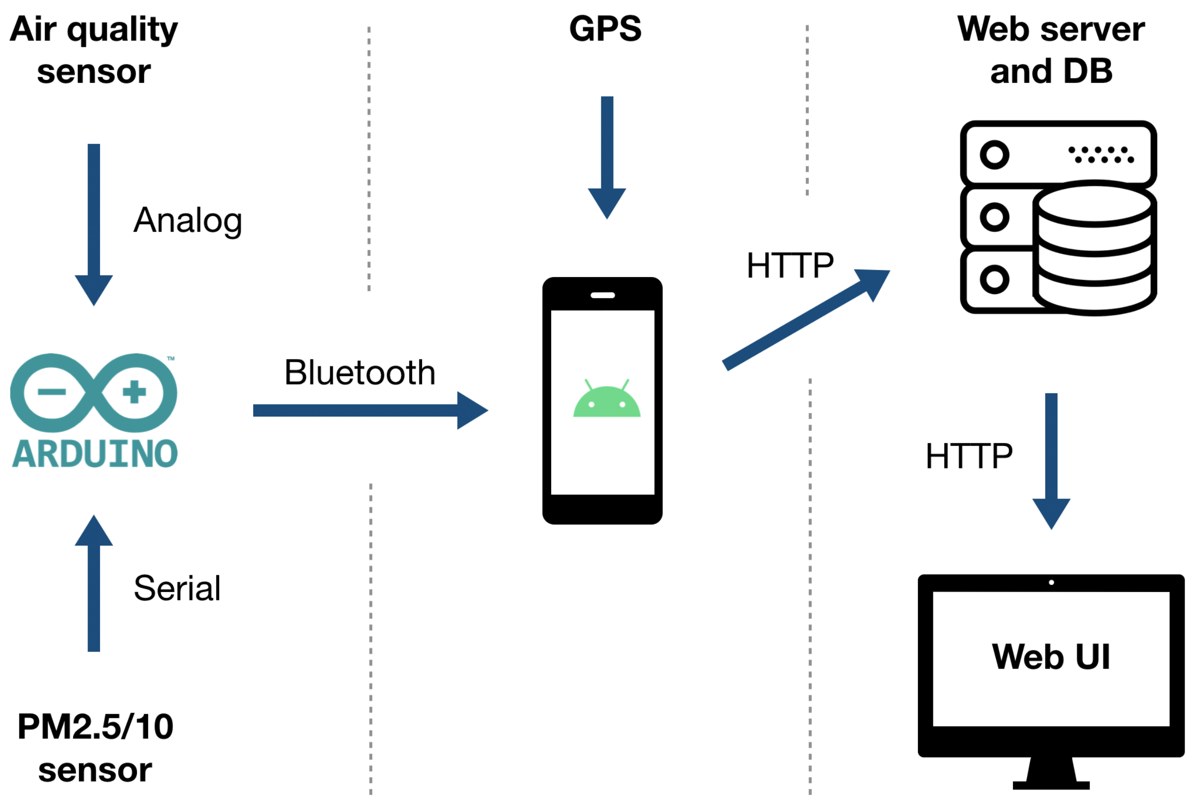

Figure 1: An overview of the main components in our system.

2.1 Arduino code

2.2 Sensors

To measure the air quality and pollutions levels we use two sensors that we connect directly to the Arduino board.

2.2.1 Air quality

We measure the air quality by using a module that includes a MQ-135 gas sensor (see datasheet[1]), which is sensitive to several pollutants including NH_3, NO_x, alcohol, benzene,

[1]https://www.olimex.com/Products/Components/Sensors/Gas/SNS-MQ135/resources/SNS-MQ135.pdf

smoke and CO_2. Its output is an aggregated level of *air quality*, either as an analog voltage or as a digital output based on a threshold. An higher value implies that the density of the detected gases is higher, and therefore the air quality is worse.

This module requires a supply voltage of 5 *volts*, and we connect its analog output to an analog input pin (i.e. $A0$) of our Arduino Nano board.

2.2.2 Particulate matter (PM2.5/PM10)

Another sensor that we use to measure the pollution is the Nova Fitness SDS011 (see datasheet[2]), which measures the density of particulate matter (i.e. PM2.5 and PM10) in the air. Its range of measured values is between 0 and $999\mu/gm^3$.

To output those values, the module has a serial interface based on the UART communication protocol, described in Figure 2, and the rate of transmission of the measurements is $1Hz$. We connected the output pin of this serial interface to the serial input pin of our Arduino Nano board. Furthermore, similarly to the air quality sensor, the module requires a supply voltage of 5 *volts*.

#	Name	Content
0	Header	AA
1	Commander No.	C0
2	DATA 1	PM2.5 Low Byte
3	DATA 2	PM2.5 High Byte
4	DATA 3	PM10 Low Byte
5	DATA 4	PM10 High Byte
6	DATA 5	ID Byte 1
7	DATA 6	ID Byte 2
8	Checksum	Checksum
9	Tail	AB

Figure 2: The bytes sent over the serial interface

2.3 Android application

The goal of the Android application is to act as a *gateway* between the Arduino board that collects pollution data from sensors and the web application that stores and visualizes the aggregated data. While forwarding pollution measurements, it also adds the current location coordinates obtained from the phone's GPS sensor.

The application runs on a mobile phone (in our demo a Huawei P10 lite running Android 8.0) that is connected via Bluetooth to the Arduino Nano 33 BLE board, as well as with an active GSM connection. Its source code is based on an example application, `BluetoothLeGatt`, which is bundled with the official Android SDK, that allows to scan for Bluetooth Low Energy devices and connect to them.

[2]https://cdn-reichelt.de/documents/datenblatt/X200/SDS011-DATASHEET.pdf

What we added to that code was to, on every message received from the Arduino board, retrieve the location coordinates (using the Android FusedLocation API) and send that data together with the pollution information to a web server as a JSON object by performing a POST request. An example of such JSON object is shown in Figure 3. On the other hand, the message sent from the board is structured as follows: `123,12,5;`. It contains the measurements from the air quality sensor, as well as the two measured values from the PM2.5/10 sensor.

```
{
  "createdAt": "2019-12-13T09:22:49.824Z",
  "lat": 46.0037,
  "long": 8.9511 ,
  "bikeId" : "00002add-0000-1000-8000-00805f9b34fb",
  "pm10": 5,
  "pm25": 12,
  "airQuality": 123
}
```

Figure 3: An example JSON object containing the pollution measurements and location

2.4 Web application

3 Conclusions