

# An IoT Environmental and Motion Assistance System for the Smart Mobility Context

Dragoş Sebastian CRISTEA<sup>1</sup>, Silviu Florin PETRACHE<sup>2</sup>, Felix Jesus VILLANUEVA<sup>3</sup>,  
Carmen Cătălina RUSU<sup>4\*</sup>, Renato Luigi MISTODIE<sup>4</sup>, Xiaoshuan ZHANG<sup>5</sup>,  
Daniela ŞARPE<sup>1</sup>, Magdalena TUREK<sup>6</sup>

<sup>1</sup> University Dunărea de Jos, Faculty of Economics Science, Galaţi, Romania  
dragoscristea@yahoo.com, daniela.ancuta.sarpe@gmail.com

<sup>2</sup> University Politehnica of Bucharest, Faculty of Mechanical Engineering and Mechatronics, Romania  
silviupsf@yahoo.com

<sup>3</sup> University of Castilla-La Mancha, Ciudad Real, Spain  
felix.villanueva@uclm.es

<sup>4</sup> University Dunărea de Jos, Faculty of Engineering, Galaţi, Romania  
carmen.rusu@ugal.ro (\*Corresponding author), luigi.mistodie@ugal.ro

<sup>5</sup> China Agricultural University, College of Economics and Management, Beijing, China  
zhxshuan@cau.edu.cn

<sup>6</sup> University Dunărea de Jos, Faculty of Engineering and Agronomy, Brăila, Romania  
magdalena.turek@ugal.ro

**Abstract:** Currently, even in the actual socio-technological context, characterized by connectivity and easy data acquisition, there is a lack of holistic platforms built for managing heterogeneous vehicle data integration into intelligent transportation systems. This paper presents the development of an innovative smart assistant platform for the smart mobility context inside smart cities, based on a prototype of a portable device able to acquire and communicate environmental and pollution parameters together with mobility/drive information. The smart assistant platform was developed to assist public authorities, business and casual users, in various transportation/mobility scenarios, enhancing their level of comfort, safety and environmental awareness. The presented platform is able to integrate data provided by third-party systems through a custom developed software library. It offers visualization and management tools, allowing the development of worldwide distributed sensor networks based on an open data access API for externalizing public sensor data.

**Keywords:** Smart city, Smart mobility, IoT, Environment, Sustainable transportation, Environmental awareness.

## 1. Introduction

This paper presents the development of an innovative technological system for the smart mobility context, an important dimension of the smart societies and smart cities, respectively. Thus, the main objective of the current research was the development of an innovative smart assistant platform (*E.M.A* – Environmental and Motion Assistant) for the smart mobility context of a smart city, based on a prototyped device able to acquire integrated data packages by simultaneously monitoring environmental conditions and motion characteristics. The main scenario involved measuring the environmental conditions inside a vehicle, together with its motion characteristics, being further expanded with an extensive array of functionalities. For this, a portable hardware prototype was developed, the hardware being able to determine, store and analyse, over multiple driving periods and in different environmental contexts, information related to position, environment and motion.

Secondly, this research aimed to develop the smart mobility assistant platform not only for business or public scenarios, but also for the casual users, with the purpose to create an accessible solution that would promote IoT usage among casual users. A ‘casual’ user could use the developed hardware and the platform in almost any scenario that involves location-environmental-motion awareness. Thus, the current research aimed to offer an affordable technological solution to a variety of users, providing the comfort of knowing, both in real-time and historically, the position-environmental-motion characteristics of any context informationally enhanced through the deployed devices. As an example, by associating portable devices with interest vehicles, users will be able to see, through specific visuals, integrated vehicle data for any moment when the devices were active. The third objective of the research was to develop a platform that easily integrates (stores, uses and visualizes) and shares through

public API environmental data provided by third-party systems, easing the creation of a worldwide distributed sensors network. It was considered that third parties' data integration capabilities should be mandatory for any smart city solution development as, globally, there are already many deployed environmental data sources (sensors). In accordance with the aforementioned objective, the E.M.A. platform allows an easy integration of the data coming from external sources. This is accomplished by using a custom developed communication library that allows, through a publish-subscribe mechanism, an easy and intuitive sensor data transfer to the internal data structures. Due to the current context, in which few "official" public air quality metering stations are providing a coarse-grained idea about the environmental air quality, a fourth objective of the research was to create a crowd-sensing driven and fine-grained platform for air quality metering, that any citizen could use, being able to share the collected air quality information. This kind of information could be successfully used for creating smart applications able to: a) monitor the air quality of any street/area at any time, b) provide green routes for leisure/sports activities, c) help in defining fine-grained pollution models for city layout plans, d) establish air quality impact in business (like housing prices), e) provide user, on long term, with a vision about air quality level of his work and personal environment.

The remainder of this paper is structured as follows. The background of smart mobility literature and existing studies are presented in Section 2. Section 3 describes the E.M.A. development from various perspectives: necessity, technical architecture, hardware prototyping, data storage and functionality and section 4 ends the article with the conclusion and the directions for future research.

## 2. Literature Review on Smart Mobility

The Internet of Things is envisaged to interconnect billions of devices, from sensors and actuators to smart objects, computers and vehicles (Năstase, Sandu & Popescu, 2017). Smart mobility represents an important aspect embedded in the larger multidimensional smart city concept, characterized by personalized environments capable of providing continuous remote monitoring via sensor networks

(Coardoş & Marinescu, 2020). Nazarko et al. (2015) emphasizes that the smart mobility solutions development involves many actors like researchers, municipalities, environment agencies, transport companies and, why not, software development houses. In (Ismagilova et al., 2019) it is mentioned that the main smart city aspects are related to smart mobility, smart economy, smart architecture, smart living, smart government, smart environment, smart people, smart technologies.

As stated in (Benevolo, Dameri & D'Auria, 2016) and in (Pinna, Masala & Garau, 2017), smart mobility represents a set of coordinated actions developed to increase the effectiveness, efficiency and environmental cities sustainability, being part of a larger concept, which is the sustainable transportation.

Smart mobility is characterized by connectivity and huge amounts of data that are transmitted in real time by mobile users, data that can be used by public institutions, business and individual users for a simultaneous conduct dynamic management. Examples of the smart mobility concept implementation are presented in (Dispenza et al., 2017), where the authors propose a smart city platform that introduces and integrates new technologies in urban context with the aim to improve mobility services for citizens, in (Anastasiadou & Vougiaris, 2019), where a model for a proper implementation of Intelligent Transport Systems (ITS) based on human-centred approach in smart cities design is proposed, in (Battarra et al., 2018), where the Italy's metropolitan cities models that promote sustainable and efficient mobility are presented and analysed and in (Gross-Fengels & Fromhold-Eisebith, 2018), where the authors describe mixed methods combining qualitative and quantitative data acquisition and analysis for the adoption of Smart Mobility in rural areas. Several other scientific studies, like (Faouzi, Leung & Kurian, 2011), describing transportation data fusion, (Mangiaracina et al., 2017), presenting the Intelligent Transportation System, (Papa, Gargiulo & Russo, 2017), emphasizing on smart mobility behaviours and (Battarra et al., 2018), showing a concrete ITS examples, confirm the necessity of ICT systems for the urban smart mobility enhancement. Also, in (Chandra, Shiva Harun & Reshma, 2017), the ITS (intelligent transportation

systems) are presented as a technological infrastructure based on multiple software components, implementing functionalities related to transportation safety, traffic congestion management or air pollution reduction. Thus, an ITS should put together telecommunications, electronics and information technologies for helping transport engineering to design, plan, operate, manage and maintain transportation systems, as described in (Lewald, 2017).

### 3. E.M.A.: An Environmental and Motion Assistance Platform

The *Environmental Motion Assistant* (E.M.A.) represents a technological platform based on prototyped portable devices that allows any user to obtain a clear understanding over the environmental and motion characteristics of the contexts where the devices are deployed. While being specifically designed for closed mobile contexts, for example the interior of a vehicle (cars, buses, trams, trains, etc.), E.M.A platform, through its associated portable devices, is not limited to inside environments. The user has the possibility to use it also outdoors, in various situations, for example when riding a bike or taking a walk. E.M.A. platform continuously measures and stores environmental and motion parameters (CO<sub>2</sub>, Alcohol Concentration, Temperature, Humidity, Air Quality, Dust, UV, Speed, Accelerations, Rotations, etc.) for any user context, providing insightful visualizations through a web application. If the portable device is deployed in public or business vehicles, authorities or companies are able to track and analyse current and historical conditions of a vehicle, from both environmental and motion perspectives, being able to obtain useful insights over its condition. For example, E.M.A. devices can be used by a fleet of buses allowing people to see the buses in real time as they are moving between designated stations on different city routes or when the vehicles are getting closer to their location. The users will see not only the motion details of vehicles, but also the inside conditions of each vehicle, knowing in advance the comfort level that is provided. E.M.A. usage inside a bus is just one of the many examples that could be provided, as the purpose of this paper was to build a solution that could be easily integrated in any vehicle scenario. Thus, not only buses can be monitored by using E.M.A., but also trains or any other vehicle types. For example,

it could prove useful to also visualize, in real time, the train movement, especially in countries/regions where delays are to be expected. E.M.A platform's architecture allows also the integration of vehicle internal data (collected by its devices) with outside environmental data coming from third-party sensor clusters or from its own devices adjusted to be used in outdoor conditions. Thus, it permits its users to gain a clear understanding of what's happening inside and outside of the targeted vehicles, from the perspective of what a clean and healthy air environment should be. The most important reasons why a person should be aware about the surrounding context from a multidimensional 'location-environmental-motion' perspective are *safety and comfort*.

#### 3.1 The Platform Architecture

E.M.A. platform's development was performed based on a distributed system architecture designed in accordance with different social needs of the modern society and the smart mobility framework. The endeavour of this research was to design a system architecture suitable to be embedded in smart mobility strategies and from that to build a technological platform that would seamlessly integrate location, motion and environmental data sent from various scenarios by proprietary portable devices and data sent by third parties' sensors, that is data found within different mobility contexts involving public institutions, business and casual users. Thus, Figure 1 describes the E.M.A.'s technological architecture.

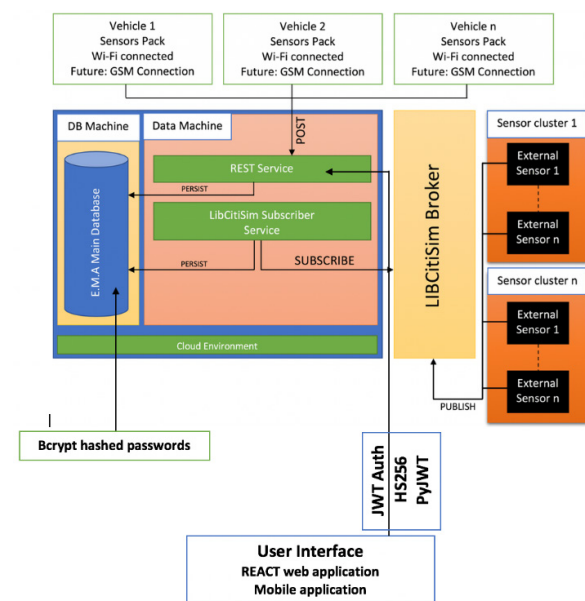


Figure 1. E.M.A. platform's architecture

As it can be observed, E.M.A. platform integrates two scenarios: 1) a scenario involving the portable device being used in a mobility context (defined by vehicles in the current case, but with the possibility of being extended to different scenarios like biking or walking); 2) a scenario that regards the integration of external third parties' sensors, grouped as sensor clusters.

For implementing these scenarios, the platform contains several components, presented in the following sections of the papers: A PostgreSQL database, a REST API, a subscriber service to collect external sensors data, a data broker that publishes external sensors data and a web application developed with React framework.

### 3.2 The Hardware Prototype

E.M.A. development involved building a portable hardware device for sustaining platform data acquisition needs, as currently the market doesn't provide any similar device that would be able to fulfil the functional requirements. The main characteristic of the device was its ability to combine data related to location, environment and motion into one unitary package. Secondly, the device was built to be portable and easily deployable in any context involving: mobility, environmental awareness and localization. Its functionality was designed to include human safety features like alerts and notifications. Also, it was targeted to help smart city communities to easily collect data in various scenarios involving mobility and environment. Last, but not least, it was designed to be cheap and easily affordable by anyone and also to allow further enhancements, especially in the artificial intelligence domain and computer vision, respectively for identifying driver drowsiness. Figure 2 presents E.M.A. prototyped device architecture.

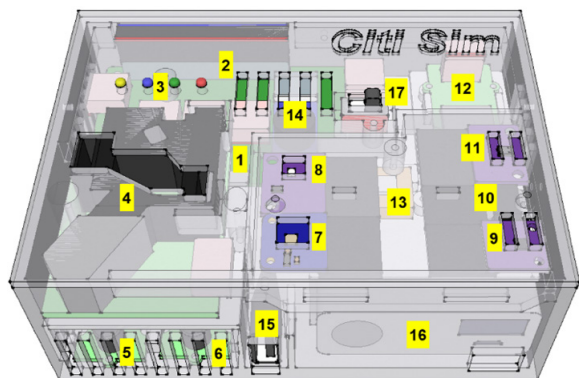


Figure 2. E.M.A prototyped device

The portable multisensory device prototype presented in Figure 2 includes in a 7x10 cm box the following elements: Raspberry Pi 0 W (1) module; shield GROVEPI ZEROGROVEPI0 (2) having attached: LEDs to indicate particular situations (3), Dust Sensor (4); Alcohol sensor (5); Air Quality Sensor (6); BME-680 Temperature-Humidity-Pressure sensor (7); Si1147 UV sensor (8); 9DOF sensor accelerometer and air pressure (9); real time clock (RTC) (10); CO2 (CCS811) sensor (11); Grove GPS Sensor (12); GPS antenna (13); buzzer (14); USB 2.0 type A female connector (15); 10000mA powerpack (16); on-off power switch used to turn on/off the device (17).

Thus, through the system of attached sensors, the device has the following characteristics:

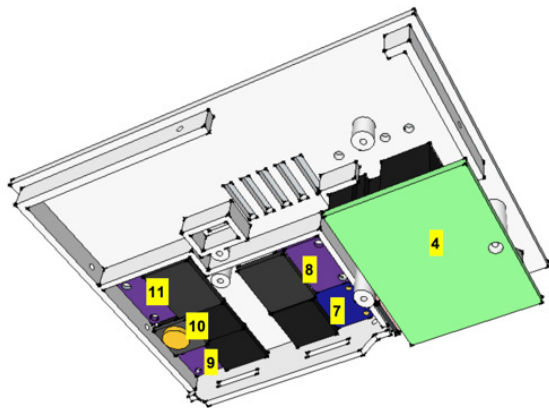
- Grove - Dust Sensor (4): Detectable range of concentration 0~28,000 pcs/liter [or 0 ~ 8000 pcs/0.01cf]. Detecting the particle diameter >1 µm. This Dust Sensor gives a good indication of the air quality in an environment by measuring the dust concentration. The Particulate Matter level (PM level) in the air is measured by counting the Low Pulse Occupancy time (LPO time) in a given time unit. This sensor can provide reliable data for air purifier systems; it is responsive to PM of diameter 1µm. This sensor uses a counting method to measure dust concentration, not a weighting method, and the unit is pcs/L or pcs/0.01cf;
- Grove - Alcohol Sensor (5): It is a complete alcohol sensor module. It is built with MQ303A semiconductor alcohol sensor. It has good sensitivity and fast response to alcohol;
- Grove - Air Quality Sensor (6): This sensor is designed for comprehensive monitor over indoor air condition. It's responsive to a wide scope of harmful gases, as carbon monoxide, alcohol, acetone, thinner, formaldehyde and so on;
- Temperature-pressure-humidity sensor (7): This module is based on the BME680 sensor. It measures humidity with an accuracy of ± 3%, pressure with an accuracy of ± 1 hPa and temperature with an accuracy of ± 1.0 ° C. Due to the fact that the pressure varies with altitude, it can also determine altitude with an accuracy of ± 1 m;
- Si1147 UV Light Sensor (8): a new SiLabs sensor with a calibrated light detection algorithm that can calculate the UV index.



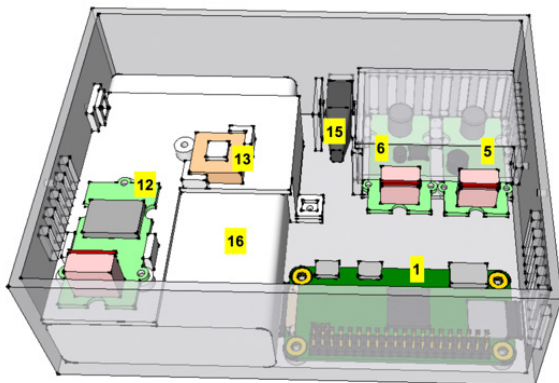
It does not contain an element of UV detection, but it approximates the UV index based on visible light and IR (infrared) emitted by the sun;

- 9DOF sensor accelerometer and air pressure (9): This module is based on MPU-9250 where MPU-9250 is a 9-axis Motion Tracking device that combines a 3-axis gyroscope, 3-axis accelerometer, 3-axis magnetometer;
- Real Time Clock (RTC) (10): It allows to keep track of time if the power is lost;
- CCS811 (11) is a gas sensor for air quality monitoring. It detects carbon monoxide (CO) and a wide range of volatile organic components (VOCs) and it has an intelligent algorithm for calculating eCO<sub>2</sub> values;
- Grove – GPS (12 & 13): This module features 22 tracking / 66 acquisition channels GPS receiver;
- Active buzzer (14).

The sensors were grouped taking into account their behaviour, their needs to access external environment and the required connections (see Figures 3 and 4).



**Figure 3.** Device lid and components



**Figure 4.** Device box and components

The GPS sensor and the antenna were placed inside the box because they are not conditioned to exterior access. These elements are attached to a support which also has the role of fastening the position of the powerpack. The alcohol and air quality sensor were separated into a mini enclosure by a removable wall as they are heated and it was desired that this heat should not be added to the one released in the box, especially by the development board. The enclosure has the outside slots (3mm thick) on both walls to allow air access to the sensors.

The sensors communicating on the I2C bus were grouped in the same area. Since some require external access in order to measure exactly certain parameters (e.g. UV sensor), they have been placed on the lid. In order to be less influenced by the heat inside the box, which normally accumulates at the top, they were locked in an enclosure that has slots on the lid next to each sensor (which requires this). In addition to sensors, the RTC module was also introduced due to optimization of the (physical) architecture of the I2C bus.

Also, a slit has been made on the lid according to the profile of the dust sensor in order not to obstruct in any form the particles that can reach it.

In order to access the Internet, the device needs a hotspot, the easiest way to provide it being via personal smartphones. The active connection to a Wi-Fi network is optically indicated by a green LED. If the device cannot connect to any hotspot, the Bluetooth connection of the device is activated. Its activation is indicated by a blue LED. The Bluetooth connection is used for being able to send custom hotspot details from any device, so the box could achieve internet connectivity.

Thus, the user could send a file via Bluetooth connection that will include the new network SSID and the access password. Afterwards, the moment the device connects to a Wi-Fi network, the Bluetooth connection is disabled. By using JSON POST requests every several seconds (configurable) to the E.M.A. Rest API, the following data is transmitted to the platform: time, coordinates, device ID, temperature, humidity, pressure, alcohol, dust, air quality, UV level, accelerations and rotations, speed. A POST request is sent only when GPS coordinates are determined. In case there is no Internet connection, the data will be stored locally in a file, which will

be handled based on FIFO strategy: when the Internet connection is restored, the data stored locally will be sent to the server / database on first in – first out approach. The buzzer will be used for acoustic signalling of particular situations through different frequency codes and different timings. For high autonomy, a power pack of 10050mA was used. Following the tests, it turned out that the device consumes, on average, after the boot process, about 500mA. The system is powered by a switch and the presence of voltage is indicated by a red LED. Also, it is worth mentioning that the device has a USB connector type A that allows the attachment of a webcam in order to acquire images and video streams needed by the future development of a driver fatigue monitoring application that would detect if the monitored person shows signs of fatigue.

### 3.3 The Data Storage

E.M.A platform uses a PostgreSQL database receiving data from the hardware device at a configurable five second transmission rate. It should be emphasized that future developments consider also moving the EMA platform to the Amazon Cloud and make use of non-relational databases, like DynamoDB, optimized for big-data scenarios.

For sustaining platform functionalities, E.M.A database uses 20 tables, as follows: ‘*User*’ – for storing all necessary information about authenticated users, ‘*UserCategory*’ – for defining different user types, like casual, business, or public authorities, ‘*WayPoint*’ – for defining waypoints in E.M.A platform. Semantically, a waypoint is a general point of interest. It can be a bus station, a tram or railroad station, a special place part of a larger itinerary, anything that a public authority would consider as being of interest, ‘*WayPointCategory*’ – needed to distinguish among different waypoint categories; ‘*Device*’ – for storing information describing the hardware device: the device’s MAC number, production date and its discoverable state, ‘*DeviceRegister*’ – a device owned by a user should be registered on the platform, ‘*Tag*’ – for defining device taxonomies (e.g. bike, car, bus, etc.), ‘*UserCategoryType*’ – used to implement casual or administrative users, ‘*Record*’ – the main data table of the platform, storing all incoming data from the devices,

‘*Route*’ – represents a table used to store the user’s defined routes (collection of waypoints logically linked) inside a city, ‘*DeviceRoute*’ – for implementing the connection between routes and devices, ‘*RouteWaypoint*’ – used to make the connection between a route and its associated waypoints, ‘*ExtSensorCluster*’ – a collection of one or many sensors deployed at some coordinates sending data to E.M.A, ‘*ExtSensor*’ – for storing each deployed sensor, ‘*ExtSensorRecords*’ – for storing all incoming data from external sensors, ‘*City*’ – a table that contains cities that have either external sensor clusters associated or public devices that can be followed, ‘*Parameters*’ – containing E.M.A measured parameters, ‘*RuleType*’ – table containing what type of rules can be used for the notification, ‘*ParameterRule*’ – a table that contains all the rules defined for each device and parameter, ‘*ExternalSensorRules*’ – table needed to implement notification messages, based on rules defined for the external sensors (e.g., notify when the temperatures exceed specific values, when it is raining, when it is too humid, when the air quality is poor, etc.).

*The E.M.A REST API service* handles the required database (DB) operations offering the following functionality: it is responsible for getting information about a specified user, returns a collection of points of interest in GeoJSON format, returns all cities that are containing waypoints or external sensor clusters, returns information on the external sensor clusters like latitude, longitude and the last recorded values for the measured parameters, returns all the routes defined in a specific city, returns information on external sensor clusters when a city is specified, returns the last recorded parameter values for all the devices of a specified user, returns the last recorded parameter values for all the devices associated to a specific route, returns information on every waypoint associated to a specific route, returns the parameters values for all the records sent by a specific device in a specified time interval, gets extended information on the waypoints associated to a specified route, returns parameter values for a specified device for the last past n hours – used for charts and analytic purposes, returns the parameters measured in each sensor cluster, returns the last n records for a specific cluster parameter, *a set of open data*

*access methods* for getting distinct sensor types on different geographical criteria and further on parameter cluster values based on city, country or continent filters. From the security perspective, E.M.A platform uses Bcrypt hashed passwords and JSON Web Authentication tokens in order to provide a proper security level for its users. JSON Web Tokens (or JWTs) offers the possibility of transmitting information from the client to the server in a stateless, secure way.

### 3.4 E.M.A. Functionality

Users' interactions with the platform can be performed through a web-based application. The E.M.A. web application was developed using React.js and additional Java Script libraries (e.g. Leaflet, Moment, react-calendar, D3, gauge-chart, recharts, etc.) providing the following functionalities: a) real time location tracking of public deployed devices; b) real-time visualization of the environmental conditions inside public transportation vehicles and their motion characteristics; c) routes definition - based on waypoints collections, d) defining routes–devices relationships, e) real-time visualizations of the environmental conditions around a specific waypoint, f) real-time visualization of the surrounding environmental conditions and motion characteristics for each user device, g) analytical charting of the surrounding environmental conditions and motion characteristics of each user device, h) visualization of the historical environmental and motion characteristics, i) real time location tracking for any of the user devices, j) environmental-motion context sharing between users, k) rule-based notifications for every measured parameter on each device, l) device alerts when environmental conditions are dropping or are higher than specific limits, m) visualizing sensor clusters at different geographical coordinates, n) real time visualization of any external sensor cluster data; o) historical charting of any parameter found in an external cluster; p) visualizations focused on any environmental-motion parameter; q) externalization of public sensor data through an open REST API.

The E.M.A front end interface makes possible for the user to have access to the previous mentioned functionalities. For example, the user can switch to a specific city where he can see a specific route and its associated waypoints – see Figure 5.



Figure 5. Public devices and route waypoints

By default, the web application runs in *public mode* which allows any user (authenticated or guest) to visualize public deployed devices. When choosing to visualize a route, the user will see not only the waypoints of that route, but also all the real time movement together with the environmental parameters of every device currently associated to that route.

By clicking any device icon, as presented in Figure 6, the user will be able to see the actual environmental conditions around that device (in the example below, the user can see what the real conditions are inside a bus).

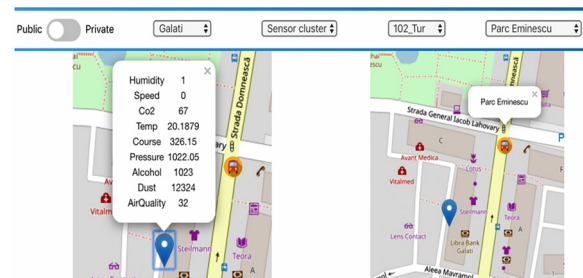


Figure 6. Public device sensor data

When choosing *private mode*, by default, all user devices will be displayed on the map, together with real time positioning, motion and environmental parameter data. The user can click on any device and see its real time status, as in Figure 7, see the devices overview, but also, he can choose any of his devices from menu drop down list and follow it in real time.



Figure 7. Personal device status overview



After choosing a specific device, the visuals will focus on that particular device, the gauges will display the last measured values of the parameters and the charts on the right will display the last hours' variation for each of the most important parameters (the number of hours is a custom value, 5 hours, in the present case).

The user can register any number of rules for a specific device parameter, but only the newest one will be taken into consideration when analyzing incoming data.

As examples of such notification rules, there can be mentioned the basic 'greater than', 'less than' or 'between' rules, that will be applied on every new data record – see Figure 8.

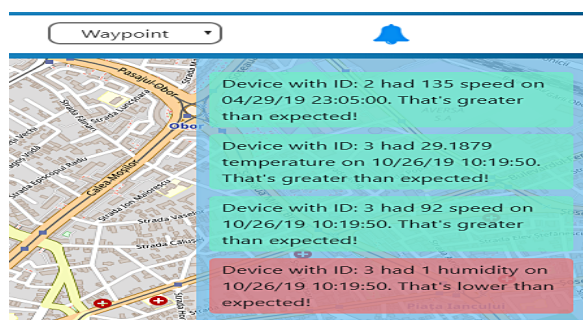


Figure 8. Rule based device alerts

The case presented in the figure above is of a user which owns 2 devices and had four custom defined rules: three for the device with id 2 and one for the device with id 1.

The rules applied for device with id=3 refer to parameters like temperature, speed and humidity, while for the device with id = 2, a speed rule is active.

In guest mode (Figure 9), the user will be able to access all the public functionalities but also a sharing context (available for all the users) that could be found while switching to private mode. Sharing the context, means that if the user knows a specific device id and a special keyword, defined by the owner of the device, he will be able to follow that device and visualize its every detail, as if he was the owner.

This functionality could prove valuable, in terms of safety, when, for example, a user must be followed while travelling in a sensitive context.

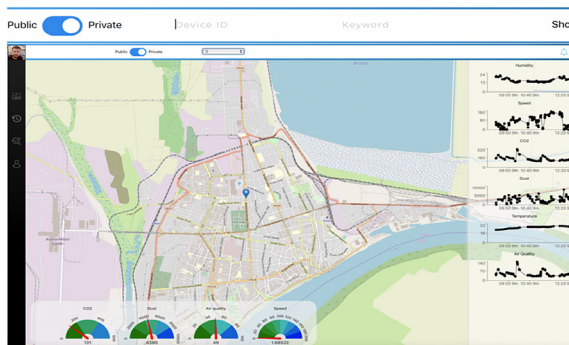


Figure 9. Guest mode

The platform provides also a specific functionality ('Analytics') which allows users to visualize any historical data obtained from their deployed devices – see Figure 10.

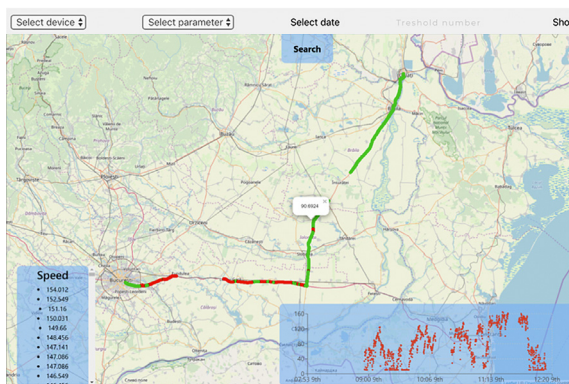


Figure 10. E.M.A Parameter analytics screen

By choosing the 'Analytics' mode, the user has the possibility to visually analyze any measured parameter from his devices. As such, after choosing from the top menu one of his devices, the wanted parameter, a time interval and a threshold number (for a different visualization of the points below/above the threshold), the user will be able to visualize the history of that parameter for the specified period plotted on the map and a time series chart presenting the parameter evolution over time. If the user clicks on any value in the list (Figure 10 left side), the focus will switch on that precise point where the value occurred, see Figure 11.

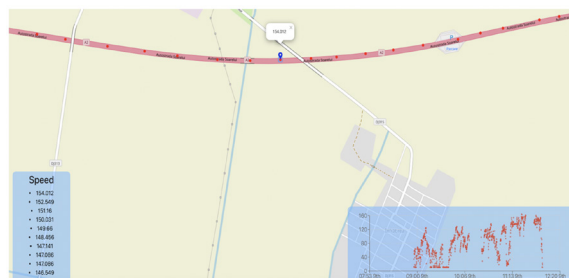


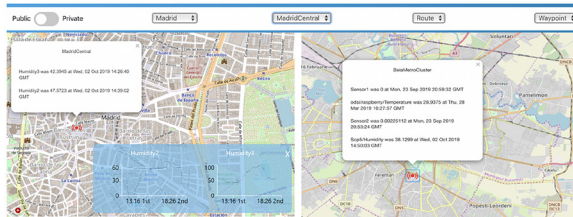
Figure 11. Focus on a specific value point



So far, emphasis was placed on the E.M.A. platform's capabilities of using its own hardware device in order to collect integrated location, motion and environmental data.

Still, besides that, the platform allows also data collection from any sensor arrays that can provide an API data access. The data collected from the external sensors is meant to be public available, therefore every user should be able to choose a city, for example, and visualize all the associated clusters and their sensor data.

The following figure (Figure 12) presents two sensor clusters that are publishing the data collected by E.M.A. platform, one in Madrid and the other one in Bucharest, allowing users to see real time last values coming from that sensors, but also the historical charts for each sensor in the chosen cluster.



**Figure 12.** E.M.A external sensors integration

In order to easily integrate external sensors, a Python library has been developed in order to provide access to a specific set of core services like: event subscription/publish, property access, simulation of events.

By using this library, a developer can subscribe to receive incoming data from various deployed sensors. Thus, city councils can provide access to relevant data (e.g. traffic, pollution, transportation etc.) for promoting the development of new smart services for citizens.

The library provides access to a set of logical topics/channels of information which group information regarding its type provided by the sensor (e.g. pollution, traffic, bin level states, bus positions, etc.). The programming model implemented by the library is a call-back publish/subscribe model where a developer defines a function that it is going to be called each time a sensor sends an event with new information to a topic.

The set of topics currently defined is presented in Table 1 (A = Analog).

**Table 1.** Libciticism library topics

Topic name	Type	Topic name	Type
ActivePower	A	Energy	A
Anemometer	A	Ozone	A
BinStatus	A	PluviometerCurrent	A
CO <sub>2</sub>	A	Power	A
CO	A	SolarRadiation	A
Vibration	A	EvacuationRoute	String
Humidity	A	Alarm	Event
SoilMoisture	A	OccupancyChanges	Event
SoilTemp	A	CarsCount	A
AccessState	Bool	CitizenReport	Report
Current	A	Temperature	A

Besides integrating worldwide distributed external sensors, E.M.A development targeted the creation of an open data repository, easily accessible by anyone (casual, public or business user) interested in collecting data for other various location scenarios.

Therefore, as previously described, E.M.A REST API contains a set of publicly available methods which provide easy programmatic access to the data collected from external sensors. The available methods allow the user to query for available sensor types/clusters based on the name of the continent, country or city.

When useful sensor clusters are identified, the user can query to get the actual data. Figure 13 displays a request made from a Python code which queries the last 10 000 points of data from all the sensors clusters in Madrid, between two dates.

```

8 import requests
9 import pandas as pd
10 import seaborn as sns; sns.set()
11 import numpy as np
12
13 resp = requests.get('http://86.127.100.48:5000/getdatacityperiod/Madrid/2019.1.1/2019.12.31/10000')
14 data_response = resp.json()
15 df = pd.DataFrame(data_response)
16 df.columns = ['ClusterName', 'Lat', 'Long', 'SensorName', 'SensorType', 'Value',
17              'RecordedTime', 'City', 'Country']
18 sensors_df = df[['RecordedTime', 'Value', 'SensorName']].copy()
19 sensors_df = sensors_df.assign(rec_id=np.arange(len(sensors_df))).reset_index(drop=True)
20 sensors_df['RecordedTime'] = pd.to_datetime(sensors_df['RecordedTime'])
21 sns.set(rc={'figure.figsize':(10,5)})
22 ax = sns.lineplot(x='RecordedTime', y='Value', hue='SensorName', palette='dark', data = sensors_df)

```

**Figure 13.** Python request for public sensor data in Madrid

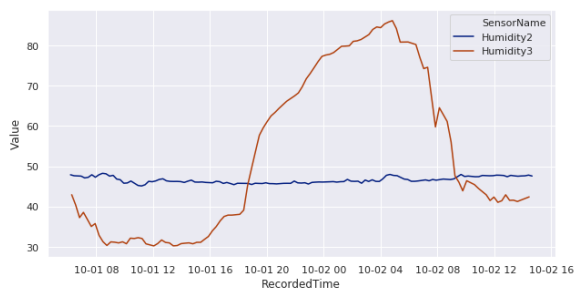
After performing the request, the data will be available in the memory space of the machine which queried for it.

Figure 14 shows the Python generated data frame based on the received data.

Index	ClusterName	Lat	Long	SensorName	SensorType	Value	RecordedTime	City	Country
0	MadridCentral	40.4154	-3.7074	Humidity2	Humidity	47.8889	Tue, 01 Oct 2019 06:11:...	Madrid	Spain
1	MadridCentral	40.4154	-3.7074	Humidity3	Humidity	42.9387	Tue, 01 Oct 2019 06:18:...	Madrid	Spain
2	MadridCentral	40.4154	-3.7074	Humidity2	Humidity	47.8387	Tue, 01 Oct 2019 06:26:...	Madrid	Spain
3	MadridCentral	40.4154	-3.7074	Humidity3	Humidity	48.4289	Tue, 01 Oct 2019 06:33:...	Madrid	Spain
4	MadridCentral	40.4154	-3.7074	Humidity3	Humidity	37.251	Tue, 01 Oct 2019 06:49:...	Madrid	Spain
5	MadridCentral	40.4154	-3.7074	Humidity2	Humidity	47.5489	Tue, 01 Oct 2019 06:56:...	Madrid	Spain
6	MadridCentral	40.4154	-3.7074	Humidity3	Humidity	38.5615	Tue, 01 Oct 2019 07:08:...	Madrid	Spain
7	MadridCentral	40.4154	-3.7074	Humidity2	Humidity	47.8938	Tue, 01 Oct 2019 07:11:...	Madrid	Spain
8	MadridCentral	40.4154	-3.7074	Humidity3	Humidity	36.8252	Tue, 01 Oct 2019 07:21:...	Madrid	Spain
9	MadridCentral	40.4154	-3.7074	Humidity2	Humidity	47.251	Tue, 01 Oct 2019 07:26:...	Madrid	Spain
10	MadridCentral	40.4154	-3.7074	Humidity3	Humidity	35.8527	Tue, 01 Oct 2019 07:36:...	Madrid	Spain
11	MadridCentral	40.4154	-3.7074	Humidity2	Humidity	47.8818	Tue, 01 Oct 2019 07:41:...	Madrid	Spain
12	MadridCentral	40.4154	-3.7074	Humidity3	Humidity	35.7813	Tue, 01 Oct 2019 07:55:...	Madrid	Spain

**Figure 14.** Python data frame containing received public data

As it can be noticed from Figure 15, the received data is acquired from 2 humidity sensors described by the sensor coordinates, the recorded time and the actual value.



**Figure 15.** Seaborn chart for data coming from an external sensor cluster

The E.M.A. public data could be further used in time series analysis for various contexts. As an example, Figure 15 briefly displays a time-based chart with data coming from the two sensors (humidity2 which is an internal sensor and humidity3 – an external sensor). A simple visual shows how the external sensor data varied in time, while the interior sensor data remained constant.

## 4. Conclusion

This research emphasizes the need and the opportunity of developing modern technological assistant platforms for the smart mobility context inside contemporary smart cities. The current research led to the development of an innovative portable solution, easily deployable and configurable that allows unitary aggregation of motion, location and environmental data. The prototyped device can be used to determine, store and provide useful notifications (speed alerts, air quality warnings or dust and humidity alerts) in various contexts that involves personal

or public vehicles. By using the E.M.A. platform it is possible to continuously measure, store and visualize locations, internal/external air condition metrics and motion parameters (latitude, longitude, CO<sub>2</sub>, alcohol concentration, temperature, humidity, air quality, ultra-violet index, speed, dust, accelerations, rotations etc.) anywhere the user is deploying the hardware (a car, a bus, a train, a truck, a bike, even on the back of a backpack). More than that, the current research provides a solution for a holistic platform able to collect and integrate data coming from various sensors distributed all over the world through a custom developed communication library.

Also, this research aims to fill a gap in the automotive industry that currently has not offered yet clear solutions for effectively determine the environmental conditions inside vehicles.

E.M.A. platform enriches the already existing mobility software stack as it targets functionalities mostly omitted by other platforms. Most of the available tools deal with functionalities related to emissions estimations, travel routing, car parking management, car sharing services, traffic congestions and control, autonomous vehicles, while E.M.A. implements the concept of combining environmental data, motion data, third parties' external sensors, offering a wider array of functionalities and a unitary perspective over the mobility data.

In terms of future development, E.M.A. platform aims to implement an artificial intelligence software module for detecting the driver drowsiness, module that will be deployed on the portable hardware device. This module will use deep-learning convolutional neural networks models for analysing the video streams provided by the camera attached to the E.M.A.'s device.

## Acknowledgements

The research reported in this paper was supported by a grant of the Romanian National Authority for Scientific Research - UEFISCDI and ITEA 3 Eureka Cluster Program, project no. PN-III-P3-3.5-EUK-2016-0043, contract 58/2017, entitled "Smart City 3D Simulation and Monitoring Platform" - CITISIM

---

## REFERENCES

- Anastasiadou, K. & Vougiaris, S. (2019). “Smart” or “sustainably smart” urban road networks? The most important commercial street in Thessaloniki as a case study, *Transport Policy*, 82(1), 18-25.
- Battarra, R., Gargiulo, C., Zucaro, F. & Tremitterra, M. R. (2018). Smart Mobility in Italian Metropolitan Cities: A comparative analysis through indicators and actions, *Sustainable Cities and Society*, 41(1), 556-567.
- Benevolo, C., Dameri, R. P. & D’Auria, B. (2016). Smart mobility in smart city action taxonomy, ICT intensity and public benefits, *Lecture Notes in Information Systems and Organisation*, 11(1), 13-28.
- Chandra, Y. R. V. S., Shiva Harun, M. & Reshma, T. (2017). Intelligent transport system, *International Journal of Civil Engineering and Technology*, 54(1), 57-64.
- Coardos, D. & Marinescu, I. A. (2020). Monitorizarea stării de sănătate a persoanelor în vârstă la domiciliu. Casele inteligente – provocări și tendințe, *Revista Română de Informatică și Automatică*, 30(1), 9-26.
- Dispenza, G., Antonucci, V., Sergi, F., Napoli, G. & Andaloro, L. (2017). Development of a multi-purpose infrastructure for sustainable mobility. A case study in a smart cities application, *Energy Procedia*, 143(1), 39-46.
- El Faouzi, N. E., Leung, H. & Kurian, A. (2011). Data fusion in intelligent transportation systems: Progress and challenges - A survey, *Information Fusion*, 12(1), 4-10.
- Gross-Fengels, S. & Fromhold-Eisebith, M. (2018). Adapting transport related innovations to rural needs: Smart Mobility and the example of the Heinsberg region, Germany, *Advances in Transport Policy and Planning*, 2(1), 125-162. DOI: 10.1016/bs.atpp.2018.09.007
- Ismagilova, E., Hughes, L. & Dwivedi, Y. & Raman, K. (2019). Smart cities: Advances in research - An information systems perspective, *International Journal of Information Management*, 47(1), 88-100.
- Lewald, A. (2017). *Unlocking Mobility’s Potential How to Make Mobility Smarter and Cleaner*. Available at: <<<http://climatechange-theneweconomy.com/sustainable-transport-kapsch/>>>, last accessed: 12 June 2020.
- Mangiaracina, R., Perego, A., Salvadori, G. & Tumino, A. (2017). A comprehensive view of intelligent transport systems for urban smart mobility, *International Journal of Logistics Research and Applications*, 20(1), 39-52.
- Nazarko, J., Ejdys, J., Halicka, K. & Nazarko, L. (2015). *Clean Mobility and Intelligent Transport Systems*. The Institution of Engineering and Technology.
- Năstase, L., Sandu, I. & Popescu, N. (2017). An experimental evaluation of application layer protocols for the internet of things, *Studies in Informatics and Control*, 26(4), 403-412. DOI: 10.24846/v26i4y201704
- Papa, R., Gargiulo, C. & Russo, L. (2017). The evolution of smart mobility strategies and behaviors to build the smart city. In *Proceedings of 5th IEEE International Conference on Models and Technologies for Intelligent Transportation Systems, MT-ITS 2017* (409-414). DOI: 10.1109/MTITS.2017.8005707
- Pinna, F., Masala, F & Garau, C. (2017). Urban policies and mobility trends in Italian smart cities, *Sustainability*, 9(4), 494.