# Smart Environmental Monitoring Beacon

Pop Alexandru[1], Manea Andrei[1], Sabău Cristina-Mădălina[1], Ovidiu Stan[1]
Department of Automation, Technical University of Cluj Napoca, Cluj, Romania
*alex.pop1819@gmail.com*

***Abstract*—The purpose of this paper is to present a beacon created for monitoring the environmental conditions, like weather parameters, air pollution, sound levels and UV radiation index. The data is stored in the cloud through an internet connection using a GSM module. The statistics regarding the weather evolution conditions and life quality in the areas where the beacon is installed are available on natively designed iOS and Android applications. The beacon is intended for the use of the local authorities, who can obtain and manage the statistics or for personal use. The users can access the history of the statistics, they can write reviews or they can notify the local authorities if they seize a problem in their neighborhoods.**

***Keywords— beacon, weather, raspberry pi, pollution, air quality, smart cities***

## I. INTRODUCTION

Nowadays, the continuous technological evolution opens new benefits, greatly increasing our life quality. There are also side effects: environmental pollution and fast-paced lifestyle stress. There comes one rather ambiguous question: how is our endpoint (infrastructure development, scientific breakthroughs) affecting us and the environment? Monitoring weather conditions, pollution levels and facilities is easily accessible to the public [1]. The main problem is raising awareness regarding these facts. According to studies conducted by Yale University in 2015 [2], around 40% of adults worldwide have never heard of global warming. This number depends on education level, local culture and economy of the issued country.

An important aspect that major cities consider and are concerned about is the air quality in the coverage area because it has been shown to have a significant impact on human health [3]. One more problem we have identified was the lack of comparison terms between lifestyle and life quality in distinct big cities. Having a dynamic lifestyle implies travelling or even relocation. How much do we know about the places we are travelling to? How could local authorities gather information from their citizens in order to caringly improve their life quality.

The convergence of three main aspects is used in order to define the urban areas' quality of air [4]. Those aspects are: regional factors, the industrial "hot spot" areas, human activities (heating and lighting of houses), public and personal transportation vehicles, construction sites, etc. At the same time, urban pollution is different spatially in the same city due to the arrangement of the natural and artificial physical features of an area and local meteorology. Urban air quality does not only affect the health of residents, it also produces a number of effects on materials, vegetation and visibility [5].

In rural areas, the situation is quite different, if we consider that there are no industrial "hot spots". Here, the quality of air is determined by weather conditions and airborne pollutants from other areas and, generally, the concentration levels of pollutants are lower than in the urban areas [4].

Air pollution monitoring with low-cost sensors has gained a great deal of interest in recent years due to the development of technology and the widespread deployment of the Internet of Things (IoT). IoT is used for creating an interconnection between different objects using the exchange of data over networks without human interaction. Nowadays, there is a lot of implementation of IoT in different domains, such as medical, agriculture, automotive, transportation, etc. [6][7][8].

The rest of the paper is structured as follows: firstly, the system architecture is outlined, in Section II, along with the important core concepts. Section III describes the implementation of our solution, having in mind the hardware and the software implementation, while Section IV discusses the software development and the evaluation of the system's performance, with an emphasis on crowbar protection design. Finally, Section V gives the concluding remarks.

## II. CORE CONCEPTS

### *A. Overall System Architecture*

The designed and implemented device is a monitoring beacon which gathers data from the environment and stores it into the cloud (Figure 1). Using analytics and statistics we can obtain useful information based on gathered data. This information has several use cases, depending on their nature.

For environmental pollution air quality data provides information regarding CO2 concentration (percentage). The sound level data allows computation of sound pollution's (dB) average and its evolution in metropolitan areas.

For weather monitoring the device provides information regarding: temperature, humidity, atmospheric pressure, UV radiations level. This information can easily prove global warming phenomena, on long term increasing public awareness, if the coverage is large enough.

The event handling section safeguards the building the device is mounted on, notifying the users and the authorities in

978-1-5386-2205-6/18/$31.00 ©2018 IEEE

case of fire in the surroundings (using the air quality sensor) and in case of an earthquake. These events' accuracy is affected due to low quality sensors used for manufacturing.

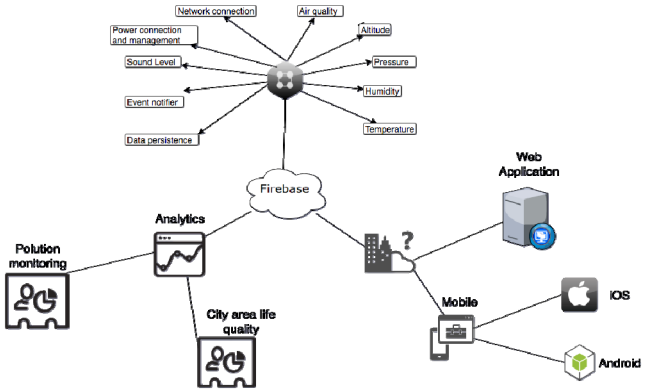


Fig. 1. Workflow of the designed beacon

For the open public native mobile applications are available (on iOS and Android). The main features of the applications include: life sensors feed, history, events notifiers, locating the device and its validation area, get and post reviews regarding the concerned area (from locals, visitors). The administrator feature allows the administrator to access the device system itself. This feature will be presented in the Software implementation section.

A web application is available for the local authorities giving them full access to all the devices history and statistics. For the authorities the reviews section is also available, in order for them to easily identify the lower rated areas and their reasons, for actions to be taken.

### B. *Firebase*

Firebase is a real-time storing database and analytics cloud provided by Google and it uses the JSON language [9]. The Firebase database is a NoSQL database and its services are based on Node.js, so its response will be fast, compared to other web services. We decided to use Firebase in order to enable our middleware and for interfacing with third-party applications. Despite the claimed 16 milliseconds response, the average real-time response of Firebase is 30 milliseconds, which is fast enough for the current application. The free database storage limit is up to 1 GB, allowing up to 100 simultaneous connections. Averaging JSON size at 300 bytes, the free storage limit will be exceeded after approximately 35 days, considering an update frequency of 1Hz.

The primary reason why we chose Firebase is due to the fact that it uses the JSON format and allows developers to create listeners for specific sections of the JSON document. These will be triggered when the data is modified, added, deleted or moved. This leads to the creation of a simple mechanism for interfacing with mobile or web applications. Furthermore, because Firebase allows users to add authentication to their own firebase, users can be sure their data is protected against malware attackers.

### C. *Raspberry Pi 3B*

Raspberry Pi 3B is the third generation of Raspberry Pi. This is a low cost single board computer and replaced the model Raspberry Pi 2 Model B in February 2016. This model includes 802.11n Wi-Fi, Bluetooth 4.0, quad core 64-bit Ram, ARM Cortex A53 (instruction set ARMv8) running at quad-core 1.2 GHz 50% faster than the previous version. It also has Broadcom video core IV @ 400 MHz, 4 USB ports and exchangeable and changeable memory, which depends on the SD cards used: 8 GB, 16 GB, 32 GB [10].

### D. *MCP3008 analog to digital converter*

Raspberry Pi 3B board does not have a way to read analog inputs. Therefore, we use the MCP3008 to acts as a bridge between digital and analog. The MCP3008 is an 8-channel 10-bit analog-to-digital converter [11] and Raspberry Pi can interrogate using 4 digital pins. MCP3008 connects to Raspberry Pi using a serial SPI connection. You can either use the SPI hardware bus or any four GPIO pins and the SPI software to talk to MCP3008. The SPI software is a bit more flexible because it can work with any pin on Pi, while the SPI hardware is slightly faster but less flexible because it only works with the specific pins. This makes it perfect for Pi to integrate simple sensors such as photocells, potentiometers, thermistors, etc.

### E. *SIM800L GSM/GPRS*

SIM800 is a complete Quad-band GSM/GPRS solution that support Quad-band 850/900/1800/1900MHz. This module can be used to transmit SMS, Voice or data information, it has a compact size and a low power consumption. The operating Voltage is recommended to be between 3.4 and 4.4 Volts and the recommended current is between 1 and 2 Amp.

## III. Design of Hardware and Software Components

### A. *Hardware implementation*

The electronic hardware is composed of sensor modules, power supply and main logic board, designed as a hat for Raspberry Pi 3B as shown in Fig. 2. Sensor include digital and analogic modules as they are described bellow.

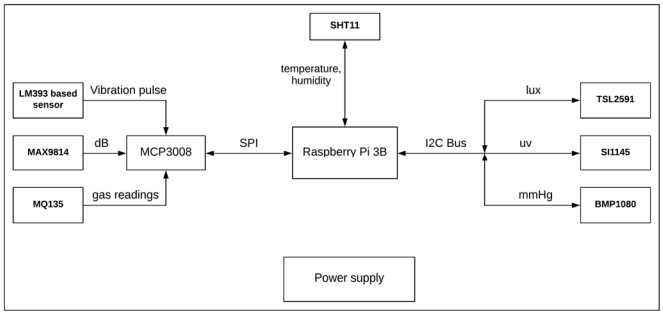


Fig. 2. Hardware design

As digital modules (I²C, SPI and other) we use a the light sensor TSL2591, the UV level sensor SI11145, the barometric pressure sensor BMP1080, the SHT11 temperature sensor and the MCP3008 analog to digital converter. The light sensor

TSL2591 determines the level of light in lumens, for accurate statistics based on the day/night cycle. The SI1145 sensor is used to determinate the UV index for approximating the area's exposure to UV light. BMP1080 reads atmospheric pressure, in order to determine correlation with other weather states. SHT11 is a high precision temperature sensor and it is using its own communication protocol. The last digital module used is the MCP3008 analog to digital convertor used with a SPI interface.

As analog modules we have a vibration sensor, a microphone and a gas sensor. For the vibration sensor we used the LM393 which provides a voltage pulse in case of device movement/vibration. This is used to simulate the earthquake reading. The adaptive/controllable gain microphone MAX9814 provides dB level statistics for the area near the device and is used to monitor the sound pollution. As gas sensor we pick the MQ135. It sends the levels of the CO2 as a percentage and it is also used to detect other hazardous gases such as CH4 - methane, CO - carbon monoxide and benzene based composites.

*1) Crowbar protection*

As for the power supply circuit, a crowbar protection design was implemented to prevent overvoltage's. Two Schottky diodes and a N-FET Transistor allow the circuit to be supplied by a secondary power source (2S 1.8Ah LiPo battery) other than a 12V 3.6A pluggable transformer (Fig. 3). High current power supplies are needed because of SIM800l GSM/GPRS module which can take up to 2A when operational. This circuit also supplies a 12V overclockable fan, isolated under the Raspberry Pi board, feedback controlled by the board in order to maintain a constant temperature of the processor and not to affect the other readings.

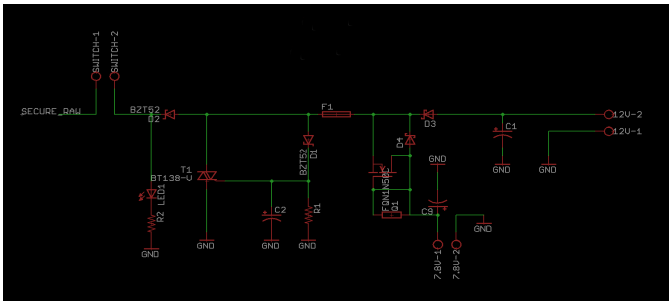


Fig. 3. Power supply circuit schematic

The logic circuit connects all previous modules in order to assure proper operation state. The external LiPo power supply can keep the device running for up to 2 hours (without GSM module attached) and half an hour with the GSM module attached. Analog sensors are connected to the MCP3008 module, provided proper filtering (100uF/16v capacitors). The MCP3008 module is connected to the Raspberry Pi board using hardware Serial-Peripheral Interface. The I²C sensors are directly linked to the Inter-Integrated Circuit interface of the Raspberry Pi.

*2) Device housing*

Device housing was specifically modelled for it to fit in all the electronic modules, while also providing an ergonomic form (Figure 4), for easier assembly and setup. Special orifices providing greater and not influenced opening for the sensors was designed. Sensors' placement was chosen so that they do not interact with each other, for obtaining results as accurate as possible.

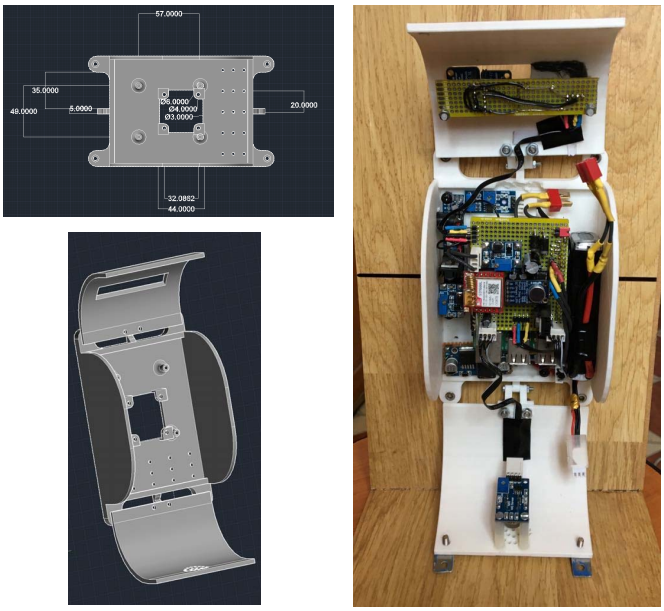


Fig. 4. Device housing designed and the result

*B. Software implementation*

IoT devices are part of a scenario in which every device talks to every other related device in an environment of an automates home and an industry and communicates more and more usable data to users, businesses and other interested parties.

The designed and created beacon integrates environmental sensors reading, localization, data backup and dispatch to cloud in a portable and stable way (Figure 4).

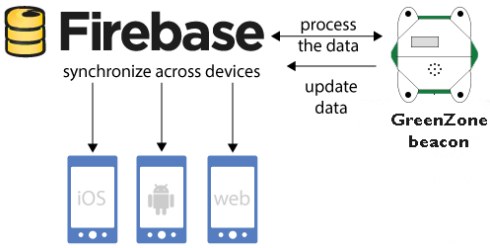


Fig. 5. Data distribution and processing workflow

Starting from low level processing, reading analog and digital sensors, the data is parsed using Python and C++ languages. Using a custom Python library, all the readings are backed-up on the device under an encoded JSON file. After building the JavaScript objects, the library tries to access the Firebase cloud database in order to synchronize last readings. If the synchronization returns a proper OK response (200), the JSON local database is updated with Firebase new snapshot keys, in order to maintain persistence. If the synchronization attempt fails, returning a code from range 400 (Client Error) to 500 (Server Error), the script backs-up data in the local

database and attempts to synchronize again when a new network connection is established or after an established timeout period passes.

Based on the transmitted data object type, the information's distribution and processing takes place (Figure 4).

The beacon creates two types of data objects: updates and events. Updates basically contain sensors readings. They are stored and can be accessed for every new beacon's entity. Based on these updates the beacon can be localized, provides evolution charts and live sensor data streaming. The mobile application workflow is shown in Figure 5.

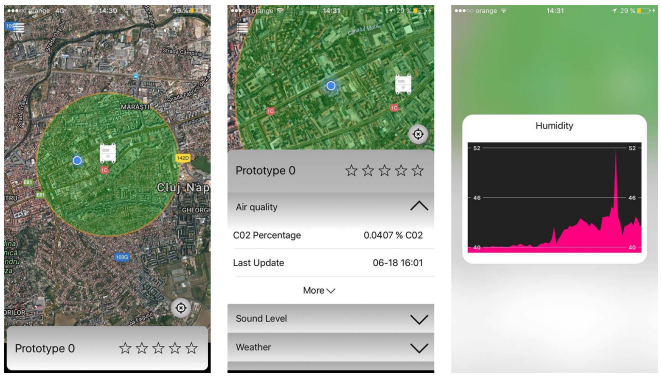


Fig. 6. Mobile application screenshoots

The update type of object is transmitted using HTTP requests from the Firebase Cloud storage. The displaying beacon is selected based on the current location of the user. The green area around the beacon's transmitting location represents the reliable range (1 km radius - this is estimative data). Charts are generated based on the previous 150 readings of the selected beacon's sensors. The full data is available using a web application. This application is designed to generate long term reports for local authorities.

The event type of object notifies the users inside the reliable range if an event happens. The event type can be an earthquake or an arson. These events are detected using the vibration sensor and gases level sensor. The accuracy of the events is questionable due to the low-quality components used.

Network connection can be established in two different ways: using local WiFi connection or using GPRS connection. The SIM800L module for GSM/GPRS connection is used when the WiFi connection is unavailable, thus generating the local network's status report (uptime/downtime). The module is also used for getting the coordinates of the device.

For maintenance purposes a local socket server is running on each beacon. Debugging, error report, data synchronization start/stop and local database read/reset can be accessed in the local network, when having provided valid authentication token. These are available inside a module implemented in the mobile application.

## IV. Conclusions

The development of this device could raise public awareness regarding global warming and help local authorities monitor pollution levels in metropolitan areas and help increase life quality of the citizens.

Due to the implementation of our system on Firebase, there are some drawbacks. Firstly, because the system uses its own programming system to enable listeners, it is possible that when a large number of sensors and applications are connected to the system, the delay in pulling the listeners could lead to data that is no longer current. Another disadvantage is that Firebase allows a free account to use only 500 MB of data [9]. Although in many situations this storage space is more than enough, if users want to store the history for their devices, they will surely end up exceeding the free space available. However, as cloud storage for IoT goes, Firebase seems to be the best available today, which is why we chose to use it.

As for further improvements we have to mention the fact that the SIM800L power consumption is too high. Therefore, we must adapt other module with better hardware implementation for GSM/GPRS connection. Other important improvements are: change the LiPo battery with a safer and stable power supply, use a more ergonomic core technology (Raspberry Pi 0 W / STM32 + SPWF04), implement web platform on a dedicated platform, reate CRON and notification server for mobile platforms, conduct studies on use case validation, reliability of the statistics (radius of beacon data validation, update time research).

## References


[1] Edward W. Maibach, Jennifer M. Kreslake, Connie Roser-Renouf, Seth Rosenthal, Geoff Feinberg, Anthony A. Leiserowitz, "Do Americans Understand That Global Warming Is Harmful to Human Health? Evidence From a National Survey", Annals of Global Health, VOL. 81, NO. 3, 2015 ISSN 2214-9996

[2] Tien Ming Lee, Ezra M. Markowitz, Peter D. Howe, Chia-Ying Ko, Anthony A. Leiserowitz, "Predictors of public climate change awareness and risk perception around the world", Nature Climate Change 5, 1014–1020 (2015)

[3] Vallejo M, Jáuregui-Renaud K, Hermosillo AG, Márquez MF, Cárdenas M., "Effects of air pollution on human health and their importance in Mexico City", Gac Med Mex. 2003 Jan-Feb;139(1):57-63.

[4] Jes Fenger, "Urban air quality", Atmospheric Environment 33 (1999) 4877-4900

[5] Tidblad, J., Kucera, V., "Materials damage - Urban Air Pollution, European Aspects", Kluwer Academic Publishers, Dordrecht, pp. 343-361

[6] J. Gutierrez, J. F. Villa-Medina, A. Nieto-Garibay, M. Porta-Gandara, "Automated irrigation system using a wireless sensor network and GPRS module" IEEE Trans. Instrum. Meas., vol. 63, no. 1, pp. 166–176, Jan. 2014

[7] Amandeep Kaur, Ashish Jasuja, "Health monitoring based on IoT using Raspberry PI", International Conference on Computing, Communication and Automation (ICCCA), 5-6 May 2017

[8] Niket Patii , Brijesh Iyer, "Health monitoring and tracking system for soldiers using Internet of Things(IoT)", International Conference on Computing, Communication and Automation (ICCCA), 5-6 May 2017

[9] "Firebase documents," https://www.firebase.com/docs/ access: 12.12.2017

[10] Raspberry Pi Foundation, Raspberry Pi Model B, https://www.raspberrypi.org/products/raspberry-pi-3-model-b/, access: 28.12.2017

[11] Microchip Technology Inc, MCP3004/3008 datasheet, 2008, https://cdn-shop.adafruit.com/datasheets/MCP3008.pdf, access: 12.12.2017