IJCRT.ORG ISSN : 2320-2882

INTERNATIONAL JOURNAL OF CREATIVE RESEARCH THOUGHTS (IJCRT)

An International Open Access, Peer-reviewed, Refereed Journal

INTELLIGENT AIR POLLUTION MONITORING SYSTEM USING IOT

R. SHANKAR GANESH[1]
1 ASSOCIATE PROFESSOR
DEPARTMENT OF POWER SYSTEM ENGINEERING
VINAYAKA MISSION'S KIRUPANANDA VARIYAR ENGINEERING COLLEGE, VINAYAKA MISSION'S RESEARCH FOUNDATION (DEEMED TO BE UNIVERSITY), SALEM 636 308, TAMILNADU, INDIA.

T. JEEVA[2]
2 PG SCHOLAR
DEPARTMENT OF POWER SYSTEM ENGINEERING
VINAYAKA MISSION'S KIRUPANANDA VARIYAR ENGINEERING COLLEGE, VINAYAKA MISSION'S RESEARCH FOUNDATION (DEEMED TO BE UNIVERSITY), SALEM 636 308, TAMILNADU, INDIA.

ABSTRACT

The level of pollution was increased due to several factors such as industrialization, urbanization, vehicles, electronic wastes etc. While burning those fuels the ejection of several toxic gases are released into the atmosphere which causes severe harmful effects to the people. The monitoring of weather is really helpful in various applications like in critical scientific systems or for simulation purposes. Weather sensing is one of the major functions in aerospace applications to check suited weather environments of other planets too. With the ongoing deterioration of environmental parameters. A chamber has been designed to monitor Temperature, CO, and Humidity and air quality level changes within the specific area. For this reason, we creating air pollution monitoring system using IOT based technology. Our objective is to monitor air quality in Real time using IOT.The air quality is monitored by different sensors used in our project.

Keywords: Air pollution, IOT technology & Gaseous pollutants

1. INTRODUCTION

Air pollution is the biggest problem of every nation, whether it is developed or developing. Health problems have been growing at faster rate especially in urban areas of developing countries where industrialization and growing number of vehicles leads to release of lot of gaseous pollutants. Mostly in industrial smoke gases causing a more air pollution in an atmosphere and the easily affected by these problems. For this reason, we are creating real time air quality monitoring system using different gas sensors.

2. EXISTING SYSTEM

Existing system focusing on utilizing pollution monitoring for distinguishing the different gases. This framework Pollution check in vehicles and cautioning framework utilizes GSM Technologies. At the point when the contamination level shoots past the effectively set edge level, there will be a notice show in the vehicle to demonstrate that the breaking point has been ruptured and this data will be send to the enrolled versatile number of RTO office and vehicle proprietor utilizing GSM.

3. SYSTEM SPECIFICATIONS

3.1 HARDWARE REQUIREMENT

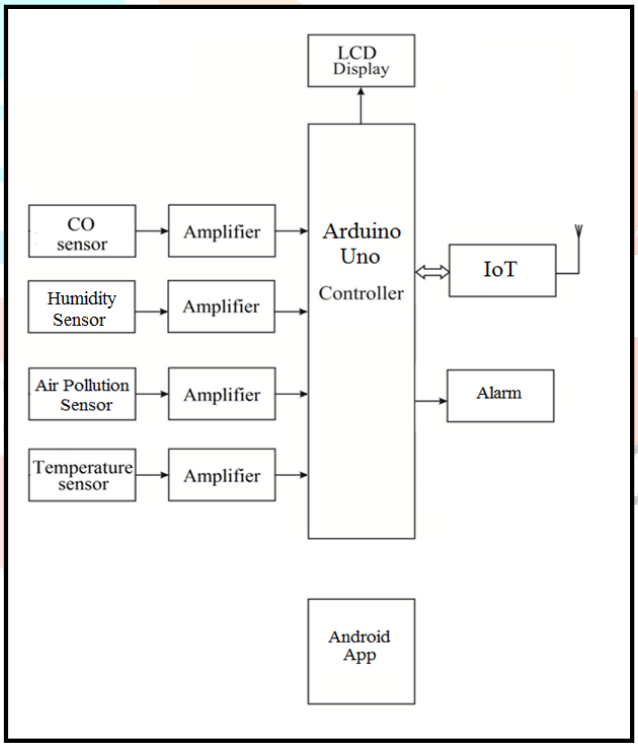
ARDUINO UNO
LCD DISPLAY
AIR POLLUTION SENSOR
HUMIDITY SENSOR
CO SENSOR
TEMPERATURE SENSOR
ALARM

3.2 SOFTWARE REQUIREMENT

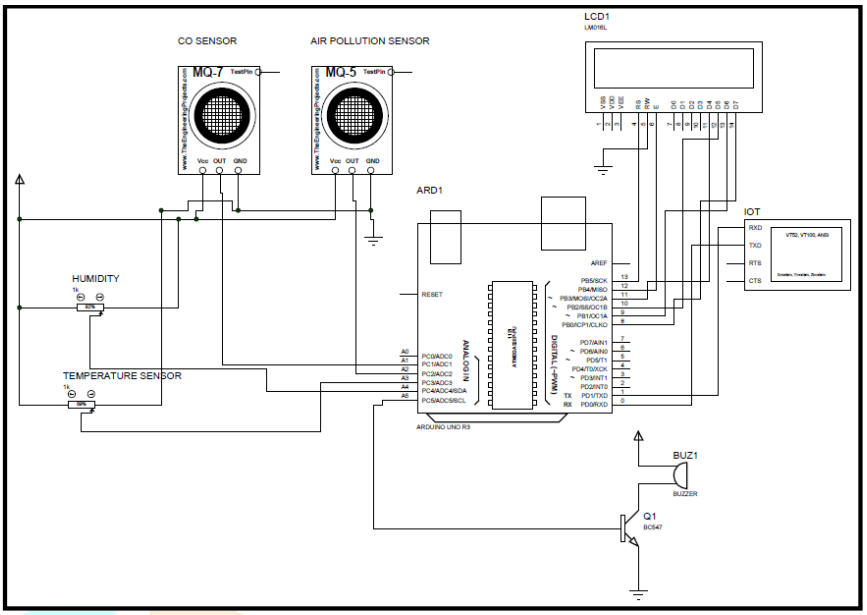
ARDUINO IDE
ANDROID STUDIO
PROTEUS

4. HARDWARE DESCRIPTION

4.1 BLOCK DIAGRAM



4.2 CIRCUIT DIAGRAM



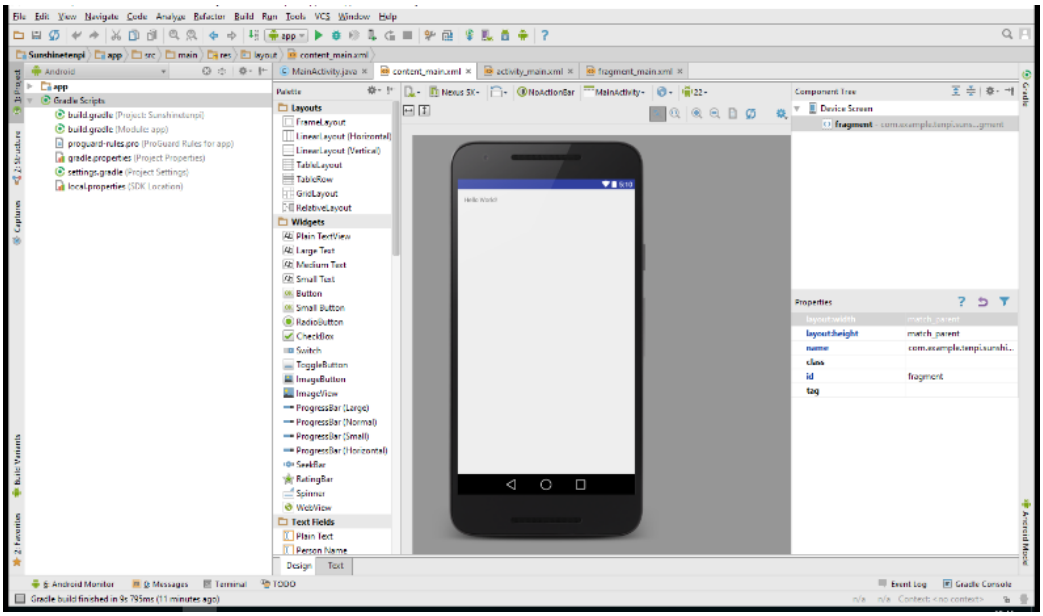
5. SOFTWARE DESCRIPTION

5.1 EMBEDDED C

Embedded C is a set of language extensions for the C Programming language by the C Standards committee to address commonality issues that exist between C extensions for different embedded systems. Historically, embedded C programming requires nonstandard extensions to the C language in order to support exotic features such as fixed-point arithmetic, multiple distinct memory banks, and basic I/O operations. In 2008, the C Standards Committee extended the C language to address these issues by providing a common standard for all implementations to adhere to. It includes a number of features not available in normal C, such as, fixed-point arithmetic, named address spaces, and basic I/O hardware addressing. Embedded C use most of the syntax and semantics of standard C, e.g., main () function, variable definition, datatype declaration, conditional statements (if, switch. case), loops (while, for), functions, arrays and strings, structures and union, bit operations, macros, unions, etc.

5.2 ANDROID STUDIO

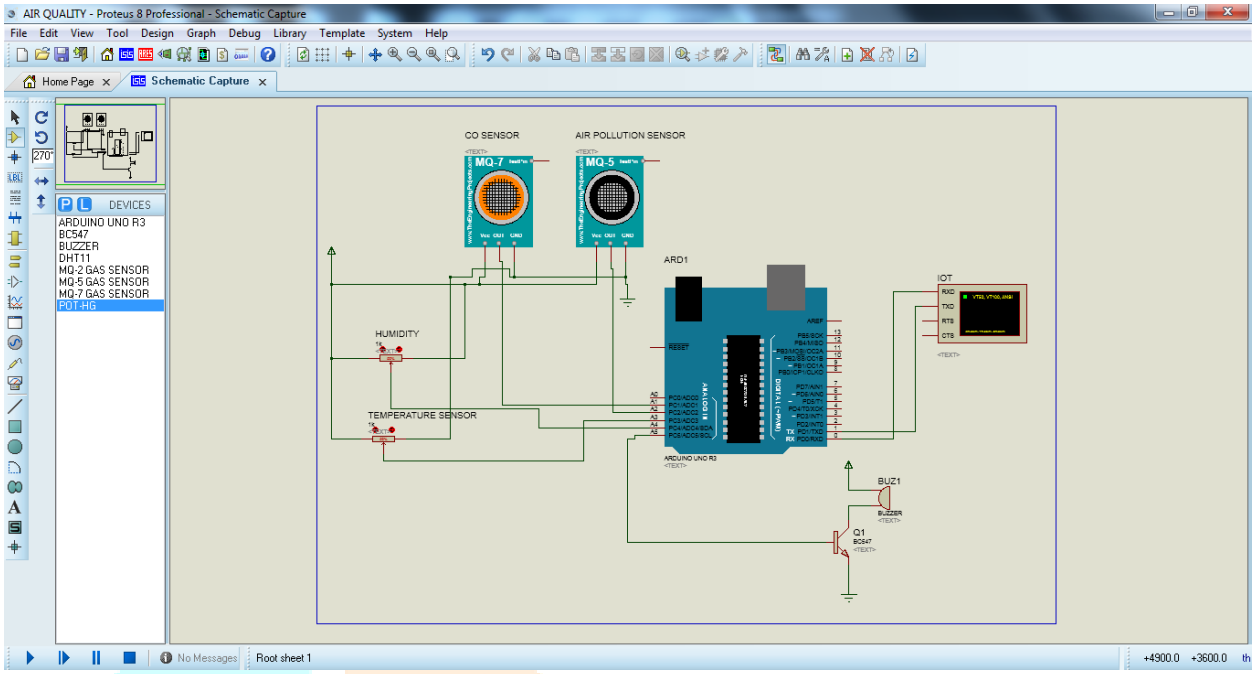
Android Studio is a windowed environment. To make the best use of limited screen real-estate, and to keep you from being overwhelmed, Android Studio displays only a small fraction of the available windows at any given time. Some of these windows are context-sensitive and appear only when the context is appropriate, while others remain hidden until you decide to show them, or conversely remain visible until you decide to hide them. To take full advantage of Android Studio, you need to understand the functions of these windows, as well as how and when to display them. In this chapter, we're going to show you how to manage the windows within Android Studio. One of the essential functions of any integrated development environment (IDE) is navigation. Android projects are typically composed of many packages, directories, and files, and an Android project of even modest complexity can contain hundreds of such assets. Your productivity with Android Studio will depend in large measure on how comfortable you are navigating within these assets and across them. In this chapter, we're also going to show you how to navigate in Android Studio. Finally, we'll show you how to use the help system within Android Studio. To take full advantage of this chapter, open the HelloWorld project we created in Chapter 1. If this project is already open in Android Studio, you're ready to go.

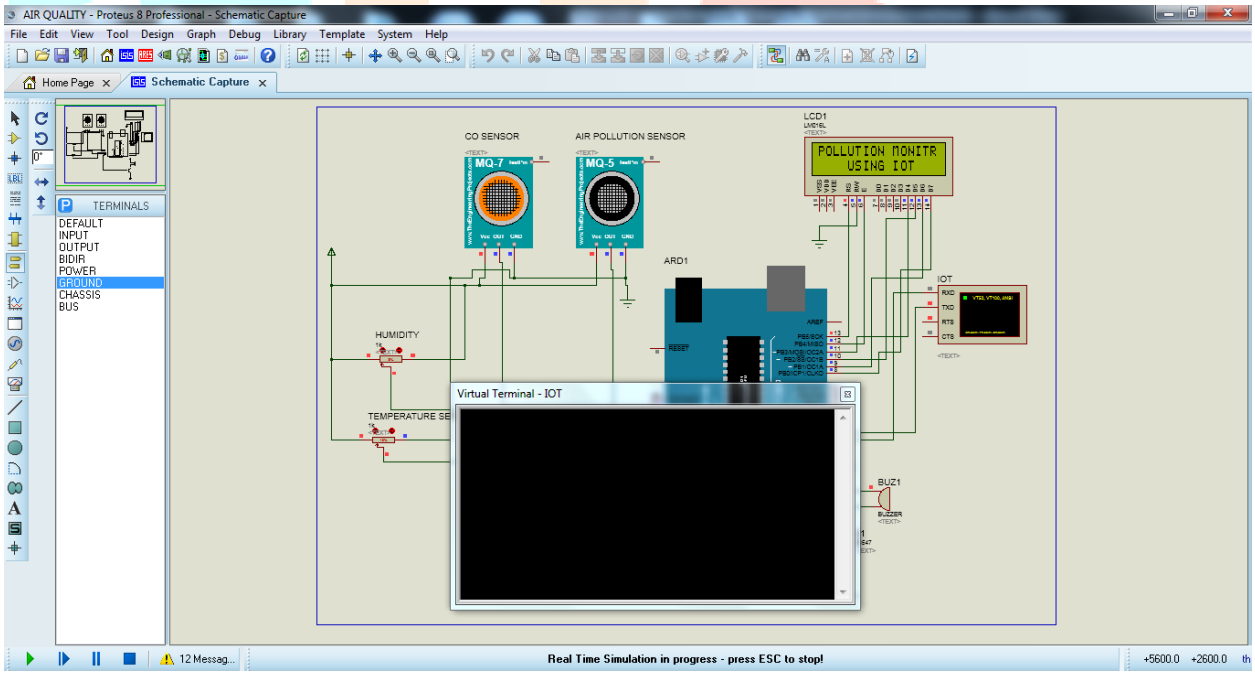


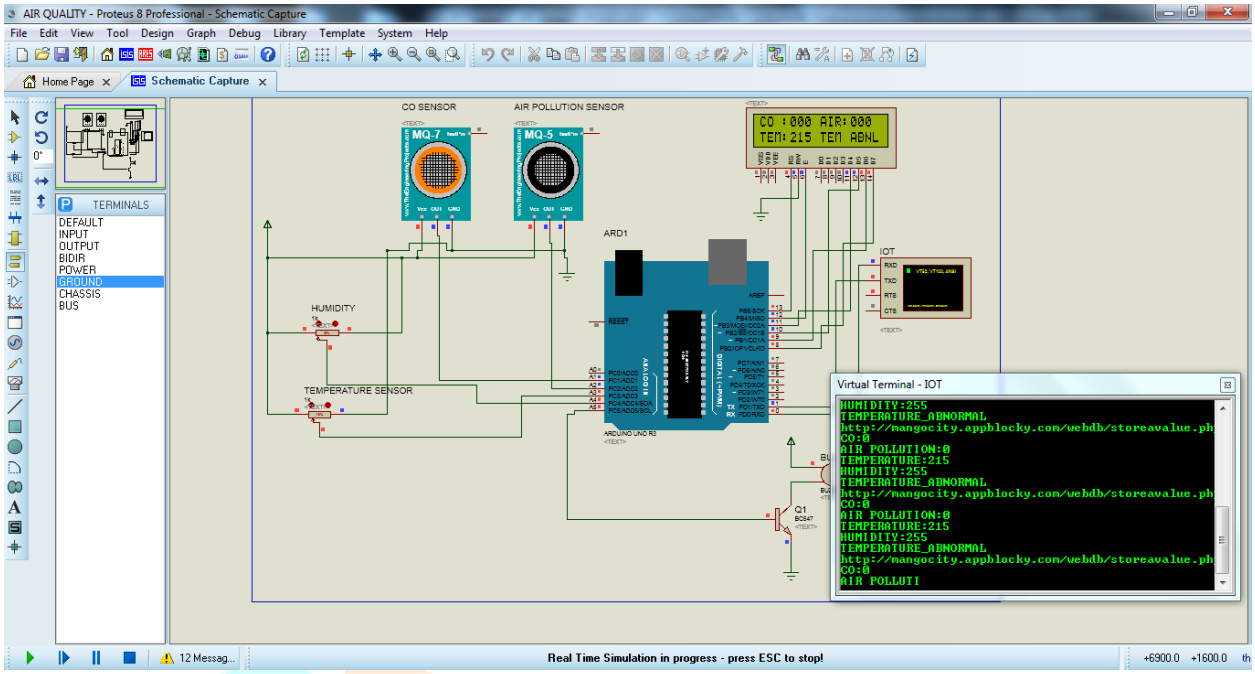
The Editor

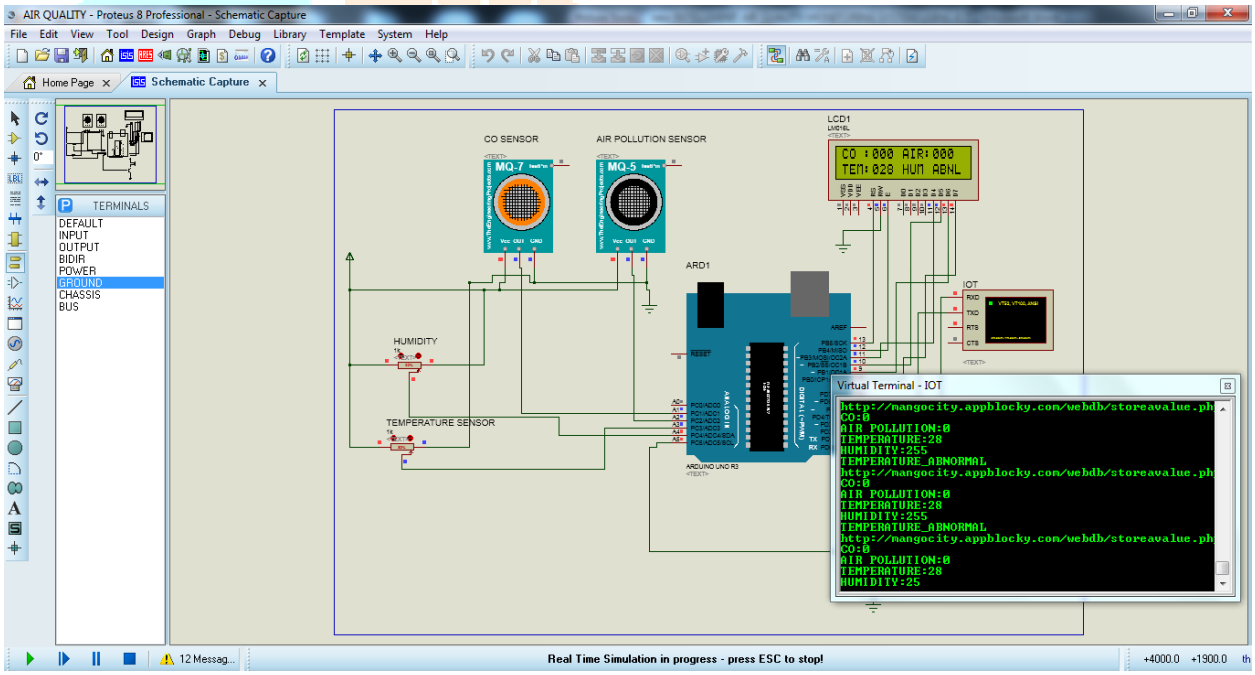
The primary purpose of any IDE is to edit files. As one would expect, the window that allows users to edit files in Android Studio is located in the center pane of the IDE. The Editor window is unique among windows in that it is always visible and always located in the center pane. In fact, the Editor window is such a pervasive feature of Android Studio that from here on out, we refer to it simply as the Editor. All the other windows in Android Studio are called tool windows and cluster in side panes (left, bottom, and right) around the Editor. The Editor is a tabbed window, and in this respect it resembles a contemporary web browser. When you open a file from one of the tool windows, from a keyboard shortcut, or from a context menu, the file displays as a tab of the Editor. As you already discovered when you built your first project, HelloWorld, the MainActivity.java and the activity_main.xml files were automatically loaded in the Editor as tabs. Android Studio tries to anticipate which files you're likely to start editing, and then opens them automatically as tabs in the Editor upon completion of the New Project Wizard. Virtually any file may be opened in the Editor, though raw image and sound files cannot (yet) be edited from within Android Studio. You may also drag and drop a file from a tool window onto the Editor; doing this opens the file as a tab in the Editor. Along the top of the Editor are the Editor tabs. Along the left margin of the Editor is the gutter, and along the right margin of the Editor is the marker bar. Let's examine each in turn.

6. SIMULATION RESULT









7. CONCLUSION

In this effective way we are monitoring an air quality in industry to avoid harmful effects for human. We are covering an all aspects of monitoring an air quality. The main feature of this project is IOT used to monitor the air quality in real time in industry. The emergency alarm is activated in industry to control their industrial air pollution. A low-cost, high-fidelity air quality monitoring device was designed, built and tested. The device can collect data at every second and transmit data via Wi-Fi and notify the personnel depending on the threshold level. And in places where the air-quality is very bad and can be a health hazard by alerting the people to dangerous levels of these sensed pollutants.

8. REFERENCES

[1] Lo Re, G., Peri, D., Vassallo, S.D.: A Mobile Application for Assessment of Air Pollution Exposure. In: Mobile and Information Technologies in Medicine and Health 2013
[2] R. Shepherd, "IOT based Monitoring chemical plumes in an environmental sensing chamber with a wireless chemical sensor network", Sens. and Act. B, vol. 121, pp. 142-149, (2007)
[3] M. Hefeeda and M. Bagheri, “Forest fire modeling and early detection using internet of thinks,” Ad Hoc Sensor Wireless Netw., vol. 7, nos. 3–4, pp. 169–224, 2009.
[4] W. Ye, J. Heidemann, and D. Estrin, “An energy-efficient MAC protocol for wireless sensor networks,” in Proc. IEEE INFOCOM, vol. 3, Jun. 2002, pp. 1567–1576.
[5] T. He, B. M. Blum, J. A. Stankovic, and T. Abdelzaher, “AIDA: Adaptive application independent data aggregation in wireless sensor networks,” ACM Trans. Embedded Comput. Syst., vol. 3, no. 2, pp. 426–457, May 2004.
[6] Baronti, P.; Pillai, P.; Chook, V.W.C.; Chessa, S.; Gotta, A.; Hu, Y.F. WSN: Wireless sensor networks: A survey on the state of the art and the 802.15.4 and ZigBee standards. Comput. Commun. 2007, 30, 1655-1695.
[7] H.-Y. Kung, J.-S. Hua, and C.-T. Chen, “Drought forecast model andframework using wireless sensor networks,” J. Inf. Sci. Eng., vol.22, no. 4, pp. 751–769, Jul. 2006.
[8] A. Pardo, L. Caara, J. Care, “Gas Measurement System Based on IEEE451.2 standards”, Sensors and Acctutaors B, Vol.16, No.1, 2009, pp.11-16
[9] Zigbeespecification,www.zigbee.org/Products/Download/Zigbee Technical Document aspx, December2010 [10] Atzori, L., Iera, A., Morabito, G.: The Internet of Things: A Survey. Computer networks 54(15), 2787– 2805 (2010)
[10] Lo Re, G., Peri, D., Vassallo, S.D.: A Mobile Application for Assessment of Air Pollution Exposure. In: Mobile and Information Technologies in Medicine and Health 2013