Evaluation of an Arduino-based IoT Person Counter

Bruno F. Carvalho, Caio C. M. Silva, Alessandra M. Silva, Fábio Buiati and Rafael Timóteo
Laboratório LATITUDE – Faculdade de Tecnologia, Universidade de Brasília, Campus Darcy Ribeiro, Brasília, Brazil

Keywords: IoT Devices, Arduino, Motion Sensor, Ultrasonic Sensor.

Abstract: The IoT devices can provide a wide range of information, which can be used to infer the behavior patterns with a large semantic bias. In this sense, an IoT network has the ability to use trafficked information to perform its own management. One type of information that can be used by an IoT network is the amount of people in a certain place. This information, combined with others, can help IoT-based systems discover characteristics about the environment in which it is deployed. Thus, the integration of the data captured provides the achievement of manifold applications, such as air conditioning regulation, security access and people management in a working environment. In this research, it is proposed the implementation of an IoT person counter. Two different technologies were used, aiming to verify the best option to design a counter device with low-cost microcontrollers and sensors. Experimental results shows that in controlled environments the IoT person counter has a satisfactory accuracy. Some limitations were also identified in order to clarify the scenarios where those devices can be used.

1 INTRODUCTION

The Internet of Things enables opportunities for more direct integration between the physical world and computer-based systems. Devices inserted in an IoT environment must be able to communicate and propagate information about the digital and physical environment, in which they are present. The information gathered by those devices can be used to provide services with fully awareness of current execution environment, and also, can be used to minimize the power consumption, processing time and the amount of packages trafficked on the network.

Information of several natures can be propagated in an IoT network, from the device hardware specification to the data inferred based on the information captured from the physical environment. For example, an IoT device can send a message with a given information about the temperature, humidity, luminosity, or motion detection, among others. Such information can be used to characterize the execution environment.

However, the information provided by IoT devices have an innate characteristic of imperfection and its quality is highly influenced by the way it is gathered. In fact, this information may even be incorrect. Most sensors feature an inherent inaccuracy (e.g., a few meters for GPS positions). Furthermore, each technology has its own specifications and limitations, creating some technical restrictions during the construction of an IoT device (Henricksen, 2004).

In this way, this paper presents the implementation of an IoT device that has the ability to inform how many persons there are in a room. This information can be useful in a variety of scenarios: to control the room temperature; to alert if the number of persons exceeds the supported by the site; to send information about trespassing; to count how many persons are using public places, in addition to several others.

Since imperfect or incomplete data may hinder the proper functioning of an IoT-based system, the proposal was implemented using two different approaches: motion and ultrasonic technology. Those technologies have been combined with microcontrollers to create a device capable of informing, with accuracy, the difference of individuals entering and exiting a particular room. In order to find the best alternative for this purpose, all implementations were evaluated in the aspect of validity of the information provided.

The remainder of this paper is organized as follows. Section II presents a brief discussion about IoT devices and used technologies. The IoT person

Carvalho, B., Silva, C., Silva, A., Buiati, F. and Timóteo, R.
Evaluation of an Arduino-based IoT Person Counter.
DOI: 10.5220/0005954601290136
In *Proceedings of the International Conference on Internet of Things and Big Data (IoTBD 2016)*, pages 129-136
ISBN: 978-989-758-183-0

counter implementations are described in section III. In section IV it is presented a set of experimental results. Finally, conclusions and discussions of future work are shown in section V.

2 RELATED CONCEPTS

In this section, we raise some concepts related to IoT objects, in order to characterize the scenario, which fits our proposal. First, on section 2.1 is presented some concepts related to IoT devices. Next section, describes concepts on microcontrollers and Arduinos. Sections 2.3, 2.4 and 2.5 clarify the concepts related to motion and ultrasonic technologies, respectively.

2.1 IoT Devices

The increasing number of smart devices, ranging from computers to simple domestic appliances, together with the ease of access to the internet, brought up the notion of Internet of Things (IoT). There are several ways to define IoT, but the essence of the concept is the device interconnection. IoT can be defined as a global network infrastructure where physical devices, with a unique identity and a virtual representation, have the ability to communicate with other devices and distributed architectures, like clusters, grids and clouds (Manna, 2014). Besides the communication with each other, IoT devices are able to access information on the internet, to retrieve and manipulate data and to interact with users. Thus, it is possible to increasingly observe the fusion of the physical and digital world.

In a typical IoT structure there are different types of sensors, such as temperature, motion, humidity, RFID, used to obtain specific data about the environment. These sensors are connected with embedded devices that collect and process the data provided by the sensors (Toma, 2014).

2.2 Microcontroller

Primarily, it was called a microcomputer due to the fact that includes built-in RAM, ROM, and I/O (Gridling, 2006).

Differently from microprocessors, the objective in developing microcontrollers was to create a complete computer in a single chip with memory, peripherals components and a processor that can be used as an embedded system (Sickle, 2001). Nowadays, there are embedded systems that require a minimal of memory and processing, while others are extremely sophisticated.

2.2.1 Arduino

Arduino is an open-source prototyping platform composed by a microcontroller or physical programmable circuit board and a software or IDE that runs in a computer. The Arduino boards interacts with the environment through peripheral components, like electronic sensors, lights, and motors. Then, using the Arduino programming language and the Arduino Software (IDE), it is possible to manipulate data captured (W. Arduino, 2016).

Nowadays, there are several different types of controllers on Arduino platform, which differ mostly on IO pins and version of flash memory (Raspaile, 2013). The most important ones are Arduino Uno, Arduino Mega2560, Arduino Nano, Arduino Mini, Arduino Due, Arduino Leonardo and Arduino ADK.

2.3 Motion Sensor

A motion sensor is a device used to detect movements in a certain area and it can use multiple types of technologies to perform this task, which defines aspects such as range, precision and sensitivity (M.Sensor, 2016). There are two main types of motion sensors:

- Passive sensor: it detects variations in energy in the surrounding area, but it does not emity energy.
- Active sensor: it transmits infrared light, microwave radiation, or sound waves and waits for its response. Due to continuous activity, this type of sensor consumes a significantly larger amount of energy.

There are several applications for motion sensors. The most common are: security systems, to open and close automatic doors and to turn on lights when a person enters a room.

2.4 Ultrasonic Sensor

The operation of an ultrasonic sensor is based on the transmission of ultrasonic pulses and the time response of the pulses. In this way, the principle underlying this technology is that speed of sound in air is approximately constant (Wijk, 1998). Thus, estimating the time for pulse reflection allows knowing the distance to the object due to the proportionality relation. Therefore, ultrasonic sensors are used frequently to detect objects and for distance measurement applications.

These sensors are capable of detecting any type of object that have sufficient acoustic reflectivity. On

Table 1: Person counter REST request.

Method	Request	Description
PUT	http://server-ip/service?serv_name=A&dev_id=B&info=C	Generic request
PUT	http://172.2.2.10/service?serv_name=count&dev_id=5&pc=5	Example of a request where the device id is 5, the number of people in the room is 5, and service name is count
PUT	http://172.2.2.10/service?serv_name=count&dev_id=1&pc=9	Example of a request where the device id is 1, the number of people in the room is 9, and service name is count

the other hand, sound absorbing material, such as cloth and foam, are not easily identified.

3 IoT PERSON COUNTER

In this section are presented the proposed IoT person counters. Such devices have the ability to inform how many people are present in the environment in which they are installed. The IoT person counter takes advantage of well-established technologies, for example ultrasonic, to perform the counting of individuals transiting in a certain place.

It is noteworthy that all these devices communicate with a central server through a REST interface (Silva, 2016), informing their unique identification (dev_id), the service name (serv_name) and the people count information (pc). The data is sent via Wi-Fi technology using an Arduino Wi-Fi shield (A.WiFi, 2016). Table 1 presents the generic structure for REST, as well, examples of the requests performed by the IoT devices.

The proposal presentation is divided into two subsections. First, it is presented the IoT person counter based on motion sensor technology. Next subsection, presents an implementation using ultrasonic sensors. All subsections presents a circuit schematic, an algorithm that performs the counting process and the specification of technologies used to build the devices. It also elucidates the limitations of each technology and their impact on the capture of environmental data.

3.1 Motion Approach

The motion approach for an IoT person counter was designed with a microcontroller and two motion sensors. The microcontroller verifies which sensor was activated and waits for the activation of the other sensor. If both were activated in a given time interval, it is identified that a person has passed in front of the device. The prototype of this device was built with an Arduino Uno (A.Uno, 2016) and two motion sensors model DYP-ME003 (Motion, 2011).

3.1.1 Circuit Schema

The circuit schema of the motion person counter is presented on Figure 1. This circuit is composed by two PIR sensors (motion sensor) connected with one Arduino Uno. The power of these sensors (VCC) is supplied by the 5V Arduino pin. The ground (GND) of each sensor is attached in the GND Arduino pin, the signal output of the first sensor is connected at the digital pin 7 and the signal of the second one at digital pin 8.

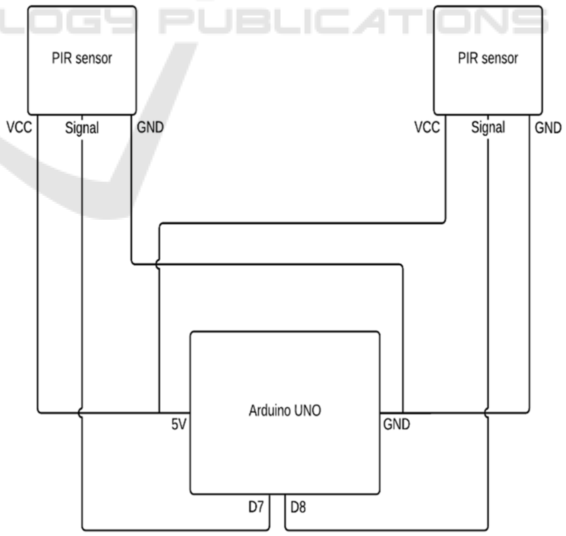

Figure 1: Motion person counter circuit schema.

In its default state, the motion sensor sends through the signal pin a LOW (0) output, meaning that there is no movement. When the PIR sensor detects motion, the signal output is HIGH (1) on the correspondent Arduino digital pin. The combine of two of these sensors allows four different states: two sensors on 0, one sensor on 1 and another on 0 or two sensors on 1. When the two sensors are HIGH, can be detected that a person passed in front of the sensors.

3.1.2 Algorithm

The motion approach algorithm is presented in Algorithm 1. The implementation defines a procedure named *read_opposite_sensor(byte, bool, int)*, and other named main. The read_apposite_sensor receives as parameter: a byte, which indicates the sensor_id, a boolean value that gives the direction in which the person is passing and an int value that represents the time interval when the sensor has to be read.

This procedure performs the reading of the opposite sensor during the time interval *max_time*, if it recognizes any motion, then it detected passage of an individual. Depending on the direction, the procedure can add or subtract the value in the persons counter.

The main procedure is executed while the device is on and checks if any motion was detected in any of the sensors. In the case of motion detection, *read_opposite_sensor* function is called to see if anyone passed in front of the other sensor. It is important to notice that this implementation does not use external libraries, just Arduino pre-defined functions.

```
Algorithm 1: Motion person counter
procedure read_opposite_sensor(byte
sensor, bool direction, int max_time) :
void
       int time = 0;
       while(millis() - time <
max_time)
           if(is_high(sensor) &&
direction)
                number_of_people++;
           else if(is_high(sensor) &&
!direction)
                number_of_people--;
      end while
      time - millis();
end procedure

procedure main() : void
      int number_of_people = 0;
       while(device_on)
           if(sensor_is_high(first))

      read_oposite_sensor(second, 1);
             else
if(sensor_is_high(second))
                      read_oposite_sensor
              (first,0);
      end while
end procedure
```

The motion sensors used on this implementation have some technical restrictions that should be taken into account as they directly impact in the device implementation: *The sensor stands in HIGH for 2.5 seconds*: When the DYP-ME003 sensor detects motion, it stands in high for approximately 2.5 seconds. In this time interval, the sensor is incapacitated to perform data capture. This limitation undermines the counting process, if people pass in a time interval shorter than 2.5 seconds. Besides that, only one person can pass in front of the device, if a group of people pass, the device will recognize as just one person.

3.2 Ultrasonic Approach

The ultrasonic IoT person counter was built with one microcontroller and two ultrasonic sensors. This device performs its functionality similarly to the motion device. The difference consists on the information provided by the sensor. The ultrasonic device informs the distance of an object that is ahead of it.

Therefore, the device checks at what distance an object is from the sensor and waits until the other sensor recognizes the object in the same distance interval. In this case, it is recognized that someone has passed in front of the device. The prototype of this device was built with an Arduino Uno (A.Uno, 2016) and two ultrasonic sensors model HC-SR04 (Ultrasonic, 2011).

3.2.1 Circuit Schema

The ultrasonic circuit schema is shown in figure 2. An Arduino UNO is responsible for collecting the information captured by the ultrasonic sensors. This circuit has two HC-SR04 sensors (ultrasonic sensor) connected with one Arduino Uno. The VCC is supplied by the 5V Arduino pin and the GND of each sensor is attached in the GND Arduino pin. Each sensor has one Trigger and Echo pin. The first sensor has Arduino digital pin 7 and 6 connected, respectively, to Trigger and Echo. Besides that, the second one is attached with digital pin 9 on Trigger and digital pin 8 on Echo.

The ultrasonic sensor constantly measures the distance of objects ahead, sending a signal through Trigger pin e receiving through Echo. When it is first turned on, the device measures the distance of all stationary objects in the environment. In this paper, this distance will be called environment distance. Besides that, when a person passes in front of the device, the distance measured will be named person distance.

The combining of these two sensors heads to four different states: two sensors reading the environment distance, one reading environment and the other a person distance, and the two sensors detecting a person distance. When two devices read a person distance, that means someone passed in front of it.

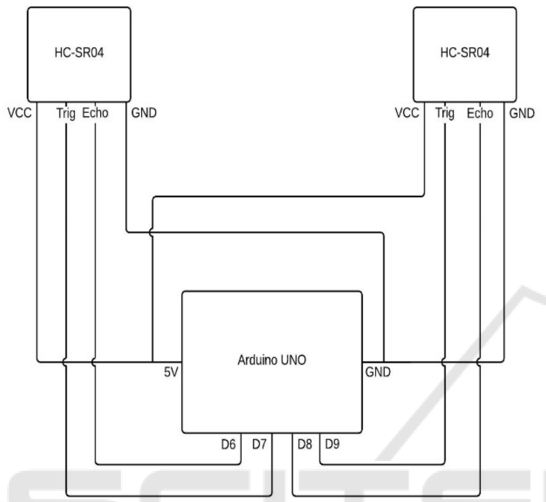

Figure 2: Ultrasonic person counter circuit schema.

3.2.2 Algorithm

To perform people counting process the algorithm presented in Algorithm 2 was deployed in the circuit. It use two procedures and one library to compute the information. The library is called "ultrasonic.h" (), and it is used to perform two types of processing:

- *timing()*: Gives the time passed between the emission and reception of an ultrasonic wave.
- *convert(bool, int)*: receives the sensor id (in this case a boolean value), and the precision wanted (in this case centimeters).

The library was adapted to work with boolean values as a sensor id. Besides that, two procedures were implemented; *read_ultrasonic_distance(bool)* and *count_persons(float,float,bool)*. The first one uses the library functions to return the distance that an object is from the sensor. The count_persons procedure counts the number of persons who passed in front of one sensor.

The main procedure uses the following pre-configured information: *delay*, *min_dist* and *max_dist*. *Delay* indicates the time when the microcontroller has to switch the direction of reading. For example, if the delay time is set to 200 milliseconds, the device will verify how many persons are entering the room for 200ms, then it will verify how many persons are exiting the room for 200ms. The *min_dist* and *max_dist* represents the distance range, this interval indicates the size (width) of the corridor where people can pass.

Therefore, the algorithm is executed while the device is on, and checks if anybody has passed in any direction. If so, this amount is incremented or decremented from the people global counter. In each iteration the direction is changed so the device can read in both directions. To finish the procedure the device sends to the server the captured information, through the rest interface, in order to preserve a historical ballast.

```
Algorithm 2: Ultrasonic Persons Counter

procedure read_ultrasonic_distance(bool
direction) : float distance
      microsec = ultrasonic.timing();
      distance =
ultrasonic.convert(direction,
Ultrasonic::CM);
      return distance;
end procedure

procedure count_persons(float
min_distance, float max_distance, bool
direction): int amount_people

      float distance_dir = 0;
      float distance_oposite_dir = 0;
      number_of_people = 0;
      do
            distance_dir =
read_ultrasonic_distance(direction);
               distance_oposite_dir =
        read_ultrasonic_distance(!direct
        ion);
              if(same_distance_range(dis
        tance_dir,
        distance_oposite_dir))
                    number_of_people++;

        while(distance > min_distance &&
distance < max_distance)
      return number_of_people;
end procedure


procedure main() : void

      float delay = 200;
      int max_dist = 100; //centimeters
        int min_dist = 40; //centimeters
        int direction = 1;
        amount_of_people = 0;
```

```
        while(device_on)
                while(time < delay)
                 if(direction)
                       amount_of_people +=
                count_persons(min_dist,
                max_dist,direction);
                 else
                   amount_of_people -=
                count_persons(min_dist,
                max_dist,direction);
                end while
      end while
      direction = !direction;
      send_to_server(amount_of_people);
end procedure
```

The ultrasonic counter has limitations related to the distance that the object is from the device. In other words, if someone passes in front of the device at a distance greater than one meter it usually loses precision.

In addition to that, if there is a group of people passing in front of the device, in the same distance range, it is possible that the proposed device also loses accuracy. Even with those limitations, for counting a large numbers of people the ultrasonic approach was more consistent than motion technology based approach.

4 EXPERIMENTAL RESULTS

4.1 Experimental Environment

In order to compare the proposed solutions in a real scenario, all IoT person counters were installed in the same laboratory room. The devices were deployed next to the door, objectifying only to capture data related to the entry and exit of people. This room is used by fifteen distinct persons at different times of day.

The proposed implementations were analyzed in relation of the accuracy of the information provided by the IoT devices. Experiments were conducted in the following approach:

- Controlled tests: Devices were tested in specific scenarios with minimum external interference. These scenarios can be seen in Figures 3,4, 5,6,7,8 and 9.

The scenarios proposed for the execution of controlled tests were designed in order to analyse the effectiveness of implementations in different situations. Thus, the scenarios differ in number of people, time interval between the passage of persons and direction of passage, in other words, entering or leaving.

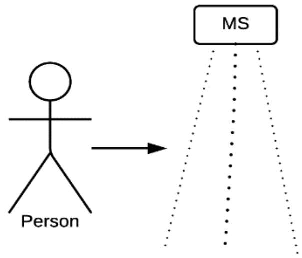

Figure 3: Scenario 1 - One person entering or leaving the room.

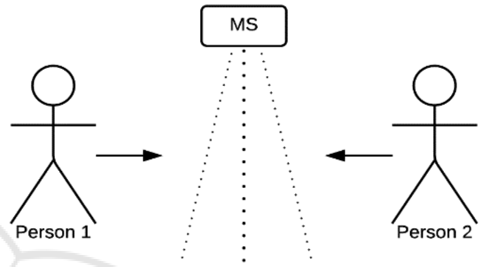

Figure 4: Scenario 2 - One person entering and other person leaving the room at the same time.

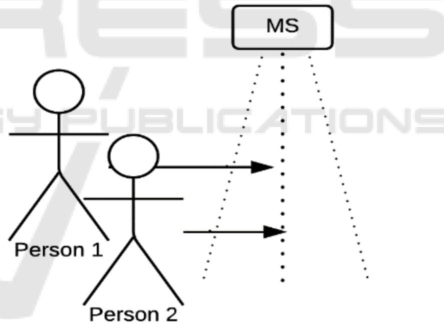

Figure 5: Scenario 3 - Two persons entering or leaving the room at the same time.

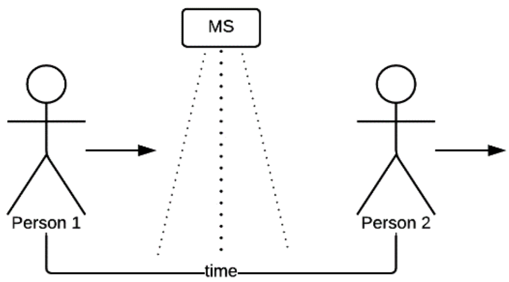

Figure 6: Scenario 4 - Two persons entering or leaving the room in a short time interval.

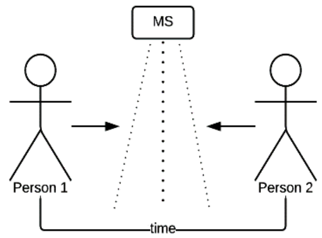

Figure 7: Scenario 5 - One person entering and other person leaving the room in a short time interval.

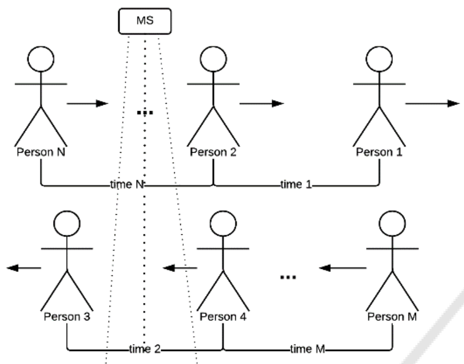

Figure 8: Scenario 6 - A group of N people entering and another group of M people leaving the room in a short time interval.

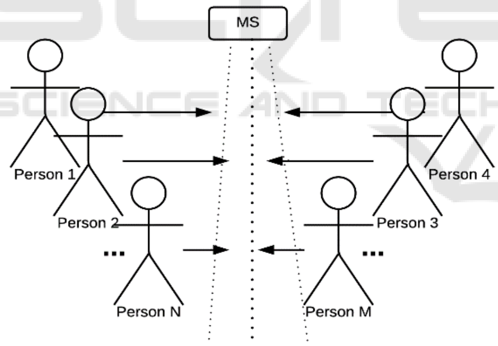

Figure 9: Scenario 7 - A group of N people entering and another group of M people leaving the room, all at the same time.

Table 2 presents the result of accuracy evaluation for all controlled tests. Each row represent a specific scenario evaluated by motion and ultrassonic approaches.

For scenarios where only one individual passes in front of the device, in a controlled time frequency, the devices obtained results next to those expected, indicating that they can be used in real environments. Both motion and ultrasonic technologies have a inactivation time range limitation, respectively, of 2,5 seconds and 500 milliseconds.

The experimental results also indicate that in scenarios with heavy human traffic both implementations have an undermining accuracy. Besides that, in scenarios where there are several persons passing at the same time the two approaches recognize as one individual.

Table 2: Accuracy of proposed implementations.

Scenario	Accuracy Result: Ultrasonic	Accuracy Result: Motion
1	90%	81%
2	0%	0%
3	0%	0%
4	87%	50.3%
5	63.2%	49.8%
6	58.7%	12.4%
7	0%	0%

5 CONCLUSIONS AND FUTURE WORK

The research work presented in this paper proposes a IoT person counter implemented with two different technologies, motion and ultrasonic. The information provided by person counter device can be used in several scenarios, such as to ensure security of access to a room, or to regulate the temperature of a room. Algorithms have been suggested to carry out the management of the information provided by the sensors. As result of the proposal, experiments indicate that in controlled environments, where there are not many people transiting, both approaches have a considerable accuracy. In contrast, scenarios in which the flux of people is significant, both applications have an impoverishment of its accuracy.

Given this scenario, where devices are susceptible to imprecision of the information captured from the environment, it is understood that it is not possible to send information, continuously and accurately, over a given situation occurring in the environment, with only one IoT device. In this sense, for future work it is proposed an IoT network person counter (INPC). The INPC utilizes information provided by different devices on the network to infer how many people are in a certain place. Such implementation will take advantage of information provided by an IoT authenticator (based on RFID technology), an IoT person counter (based on motion technology), and an IoT face recognizer (based on face recognition algorithms). Those three devices will send information to the network and all the information combined tends to indicate with a better accuracy and precision the amount of the number of people in a certain place.

ACKNOWLEDGEMENTS

The authors wish to thank the Brazilian research, development and innovation Agencies CAPES (Grant FORTE 23038.007604/2014-69),FINEP (Grant RENASIC/PROTO 01.12.0555.00) and PNPD/CAPES - Programa Nacional de Pos-Doutorado/CAPES for their support to this work.

REFERENCES

A.Uno, Arduino Uno datasheet. On-line reference acessed in 19/02/2016. https://www.arduino.cc/en/Main/ArduinoBoardUno

A.WiFi, Arduino WiFi shield datasheet. On-line reference acessed in 17/02/2016. https://www.arduino.cc/en/Main/ArduinoWiFiShield

Gridling G., Weiss B. (2006). Introduction to Microcontrollers. Vienna University of Technology.

Henricksen, K., Indulska, J. (2004). "Modelling and using imperfect context information," in Pervasive Computing and Communications Workshops. Proceedings of the Second IEEE Annual Conference on. IEEE, 2004, pp. 33–37.

Manna, S., Bhunia, S.S., Mukherjee, N. (2014). Vehicular Pollution Monitoring Using IoT. IEEE International Conference on Recent Advances and Innovations in Enineering.

Motion sensor: spcification of DYP-ME003. On-line reference acessed in 19/02/2016. http://elecfreaks.com/ store/download/ datasheet/sensor/DYP-ME003/ Specification.pdf

M. Sensor, The Beginner's Guide to Motion Sensors. On-line reference acessed in 23/02/2016. http://www. safewise.com/resources/motion-sensor-guide

Raspaile, P., Keswani, V. (2013). An Approach Towards Low Cost Wireless Sensor Network for the Applications of IoT. *International Journal of Engineering Science and Advanced Technology.*

Sickle, T. V. (2001). Programming Microcontrollers in C. LLH Technology Publishing, Eagle Rock.

Silva, C. C .M., Ferreira, H. G. C., Sousa Júnior, R. T., Buiati, F. and Villalba, L.J.G, 2016. Design and Evaluation of a Services Interface for the Internet of Things. *Wireless Personal Communications.* p. 1-38. Springer.

Toma, C., Popa, M. (2014). Internet of Thing Architecture for Context Aware Sensors Data Processing in Waste Management Solution. *Journal od Mobile, Embedded and Distributed Systems*, vol. VI.

Ultrasonic sensor: specification of HC-SR04. On-line reference acessed in 19/02/2016. http://www.elecfreaks. com/store/download/product/ Sensor/HC-SR04/HC-SR04_Ultrasonic_Module_ User_Guide.pdf

Wijk, O., Jensfelt, P., Christensen, H.I., (1998). Triangulation based fusion of ultrasonic sensor data. IEEE International Conference on Robotic and Automation.

W.Arduino, What is Arduino. On-line reference acessed in 19/02/2016. https://www.arduino.cc/en/Guide/ Introduction