Electronic rain meter for mobile sensor node using law of Torricelli

J Gomez-Rojas¹, L Camargo¹, E Martinez¹, and M Gasca²
¹ Universidad del Magdalena, Santa Marta, Colombia
² Universidad Antonio Nariño, Santa Marta, Colombia

E-mail: jgomez@unimagdalena.edu.co

Abstract. Rain in a city can cause material damage and risk for the population, hence the importance of implementing prevention and mitigation measures. These measures must be taken based on the analysis of the data collected by networks of environmental sensors. The rainfall-meter is one of the instruments used to measure rain, these are designed to operate at a fixed point. Coverage of the entire area of a city requires the installation of several of these elements. This paper shows the development of an electronic rain gauge that can operate in motion applying the principles of fluid dynamics. Two stages are proposed for its elaboration. The first step is the design, construction and testing of the sensor and transducer for the rain gauge. In the second step, the rain gauge communication is implemented. For this, the internet of things technology is incorporated, and the network is designed to provide mobility. The main result is a prototype mobile electronic rain gauge with a measurement error of 8.5%. Besides, mathematical model for the sensor, algorithm for the transducer, and communications architecture are obtained. It can be concluded that, rainfall can be monitoring in a city with few sensitive units in motion.

1. Introduction

A solution to monitor environmental variables in cities is the implementation of wireless sensors network (WSN). These networks allow the collection of environmental data that can be converted into information that can be used to make decisions that positively impact the city and its inhabitants [1]. The WSN are composed of sensitive nodes that incorporate a wide range of sensors and sink nodes that collect the data sent by the sensitive nodes and transmit it to the monitoring center. Sensitive nodes can be integrated into urban infrastructure as fixed nodes or nodes in motion [2].

The introduction of mobility in WSNs (MWSN) makes it easier to expand network coverage and improve the scope of applications, due to the versatility in topologies and their implementation [3]. In this scenario, the sensitive nodes could be installed in the public transport buses, they travel the city, collecting, storing, processing, and sending the data to the sink nodes that would be in certain strategic vehicular intersections. But at the same time, the mobility of the sensor node brings challenges in the deployment of the network [4] and difficulties in the implementation of sensors that are designed to operate in a base station.

The internet of things (IoT) allows sensors and actuators to be continuously connected to the internet; IoT makes it easy to send and receive data over wireless networks with little human intervention [5]. The application presented in this paper integrates technologies and concepts of WSN and IoT. For the physical level and the control of access to the medium, devices are incorporated that use the standard
IEEE 802.15.4 [6], commonly used protocols in WSN. To give globality to the data the message queuing telemetry transport (MQTT) protocol is used at higher levels [7]. MQTT is a lightweight, publish-subscribe network protocol, this is one of the main pillars of the IoT [8].

In cities, rain falls irregularly both in quantity and in time and space, and it can cause material damage and generate danger for citizens. Therefore, the importance of implementing risk prevention and mitigation measures [9]. These measures must be taken based on the analysis of the data collected by networks of environmental sensors. The rain gauge is the element in the sensitive node that measures the amount of rain that falls in a specific place, this instrument is commonly used in meteorological base stations [10]. The problem with this type of device is that they cannot be incorporated into a mobile platform, due to their design and construction. Because this is based on a double tipping bucket that collects small amounts of water through a funnel, and where the balance varies depending on the amount of water in the buckets, so every time the rain falls and One bucket is filled, the scale oscillates, emptying the full bucket, while the other begins to fill [11]. The measurement of the frequency of the oscillation determines the millimeters of water that fall in the place where it was installed. The oscillation frequency is altered when the device is set in motion, considering the irregularity of the streets and changes in vehicle speed.

From the above, the research questions studied arise: How to measure rainfall in a city with few sensors? How to measure rainfall using fluid dynamics? How to measure rainfall with a mobile sensor? This paper presents the design, implementation and testing of a prototype of an electronic rain gauge to be used in a mobile sensor node. With this, the previous questions are answered.

2. Method and materials
For the development of the prototype, applied research is carried out that seeks to generate knowledge, by solving a specific problem; this research is based primarily on technological findings and the results of previous basic research and is intended to move from theory to a product [12]. This project has two stages: the first is the construction of the sensor and transducer of the rain gauge and the second is the implementation of its connection to IoT.

2.1. Sensor and transducer design and construction
For the construction of the proposed rain gauge, a water collection container was used with a hole in its lower part, a pair of copper bars deployed at the top of the container, a low power voltage comparator, an analog to digital converter with an output to connect to the I2C bus [13], a GPS [14], and a Raspberry pi [15] that processes and conditions the data.

Figure 1 shows the elements mentioned and the interaction between them; when the copper bars come into contact with the collected water, they close the circuit and generate a potential difference that is detected by an operational amplifier that at its output generates an analog voltage signal that varies between 3.3 volts to 0 volts depending on the height of water in the container or the depth to which the rods are immersed in the water. The voltage signal is read by an analog to digital converter that generates data that is sent to the Raspberry pi [15] on the I2C bus [13].

![Figure 1. Schematic of the prototype.](image)
To relate the voltage signal of the operational amplifier with the height of the water in the container, a series of measurements are taken and using a polynomial regression a mathematical model of the height (h) in cm, as a function of the voltage (V) is obtained.

Based on the Torricelli’s law, see Equation (1), and on the measurement of the height of the water in the collecting container, the real average velocity of the liquid at the outlet of the lower hole can be estimated. In Equation (1), \( v \) is the velocity of the liquid at the exit of the orifice, \( g \) is the acceleration of gravity, and \( h \) is the height of the stored water [16]. Initial velocity is neglected.

\[
v = \sqrt{2gh}.
\]  

The flow or volume of the fluid that dislodges the container is calculated with the velocity and the area of the lower orifice \( (a_1) \), see Equation (2). The outlet flow rate \( (q_1) \) is considered equal to the inlet flow rate \( (q_2) \), see Equation (3); it is assuming the container is initially empty [12].

\[
q_1 = v \cdot a_1,
\]

\[
q = q_1 = q_2.
\]

The millimeters of water that fall in the sampling area is obtained from the measurements of the inlet area of the collection container \( (a_2) \), the flow rate \( (q) \), the time it takes to take the sample \( (t) \), the geographical coordinates \( (GC) \) from where the sample is taken, and a computational algorithm. Subsequently, the values obtained from millimeters of water are stored in the Raspberry pi [15] (sensor node) in JavaScript Object Notation (JSON) [17]. This is possible with firmware made for the node.

2.2. Connecting the rain gauge to the internet of things

To connect the electronic rain gauge to the IoT, the sink node is implemented; in addition to, a broker for IoT and the database are implemented in the cloud; the sink node his composed of a Raspberry pi [15] and the communication devices to connect this with the sensitive nodes and the internet. The sensitive node sends the data to the sink nodes through Digi Xbee RF long-range radios [18]. The Digi Xbee supports the following protocols: IEEE 802.15.4 [6], DigiMesh [19], ZigBee [20], ZigBee smart energy [21]. These protocols are normally used in WNS for connection. The sink node is also connected to the Internet and is the gateway for the sensitive nodes. This publishes the data through the MQTT protocol to a broker (Mosquitto) [22] with the topic “Precipitation”, and additionally this data is stored in a NoSQL database (MongoDB) [23]. This is possible with firmware made in Python [24] and run on the Raspberry pi [15] of sink node.

Users (citizens) can access the database to view the collected historical data or receive notifications about the weather, by subscribing to the topic “Precipitation”. For notifications, WebSocket [25] is used to provide a bi-directional and full-duplex communication channel over a single transmission control protocol (TCP) socket [26]. This allows MQTT messages to be received directly from the web browser. This application was developed in JavaScript [27], HTML [28], and CSS [29].

3. Results and discussions

The main result is the construction of a precipitation sensor with specific characteristics of use, through the application of physical principles and the combination of hardware and software. The results obtained when developing the proposed methodology, and that give an answer to the questions asked, are presented below.

3.1. Mathematical model of height as a function of voltage

Figure 2 shows a polynomial regression; this is performed to model a non-linear relationship between the potential difference at the output of the comparator LM393 (independent variable) and height of the water in the container (dependent variable).
Figure 2. Model the height of the water in a container.

The Equation (4) is obtained; this model is adjusted considering the coefficient of determination, $R^2 = 0.99$.

$$h = 34.578 - 62.7 \cdot V + 45.499 \cdot h^2 - 14.598 \cdot V^3 + 1.7042 \cdot V^4.$$

(4)

3.2 Algorithm for calculating millimeters of water in a mobile sensor node

Figure 3 shows the algorithm elaborated to firmware of the sensitive node; this is developed in Python [24] and implemented in the Raspberry pi [15] of the sensor node. The objective is to calculate the millimeter of water (mm or litter/m²).

![Diagram](attachment:algorithm.png)

**Figure 3.** Algorithm to estimate the millimeters of water.
3.3. Communications architecture proposed

Figure 4 shows the communication scheme that integrates various technologies with the purpose of connecting the sensors to IoT. In this it is observed how the sensitive node connects the sink node, and the sink node connects to the Mosquitto IoT broker [22] to make publications about the sensed variables.

![Communications architecture diagram](image)

**Figure 4.** Communications architecture.

3.4. Satisfactory evaluation of the sensor of the prototype

Three laboratory tests are performed to evaluate the prototype; in a controlled environment, one liter of water was applied directly to the surface (0.0182 m$^2$) of the collection container over a period. This is 54.94 mm or liter/m$^2$. In Table 1 are recorded the values of millimeters of water calculated by electronic rain gauge in each test.

| Tests | Measured $X_i$ | Absolute error $|\bar{X} - X_i|$ |
|-------|---------------|----------------------------------|
| 1     | 58.7          | 2.8                              |
| 2     | 48.7          | 7.2                              |
| 3     | 60.2          | 4.3                              |

Table 1. A slightly more complex table with a narrow caption.

Absolute error ($E_a$) is calculated with Equation (5), and the relative error ($E_r$) with Equation (6). Where $n$ is number of tests, $X_i$ the value measured in each test, and $\bar{X}$ is the mean value.

$$E_a = \frac{\Sigma |X - X_i|}{n} = 4.77,$$  \hspace{1cm} (5)

$$E_r = \frac{E_a}{\bar{X}} \times 100 = 8.5\%.$$ \hspace{1cm} (6)

3.5. Satisfactory evaluation of the proposed communication system

Figure 5 shows the correct functioning of each of the stages of the communication process, such as: Figure 5(a) the publication of the data through the Mosquitto IoT broker [22], Figure 5(b) the storage of the data in MongoDB [23], and Figure 5(c) the display of the information on the user's computer.
4. Conclusions

It is possible to build a mobile electronic rain gauge applying the principles of fluid mechanics; this rain gauge can provide rainfall monitoring coverage to a city with few sensitive moving units. The electronic rain gauge developed has an error of 8.5%. This is optimal bearing in mind that the sensor mobile node of the macro project only must determine if light, moderate or heavy rains are occurring in different parts of the city.

The communications architecture of the proposed system facilitates emptying the data collected by the sensitive node by delivering the data to the sink nodes in different parts of the city while it is in motion, improving the response times of the system. The connection of the electronic rain gauge to the internet of things using the MQTT protocol allows sending and receiving messages between devices with light network traffic, using the concept of publication and subscription. The data of the millimeters of water entered in the database can be used to obtain information on the volume and frequency of rain in the different parts of the city and with this take risk prevention measures.

The development of the electronic rain gauge to be used on the move is one of the challenges encountered in the macro research project that aims to develop a mobile sensor node and a fixed sink node for a wireless sensor network in the city of Santa Marta, Colombia, projecting a smart city. In future works, the incorporation of artificial intelligence in the sensor node will be studied to improve mobility, by estimating the characteristics of the communication as a function of the speed of the vehicle. In addition, to incorporate other mobile sensors to the nodes.

References

[1] Camargo L, Gómez J, Gasca M 2020 La Ciudad Inteligente y la Gestión de las TIC. Caso de Estudio: Ciudad de Santa Marta (Santa Marta: Editorial Unimagdalena)
[2] Karl H, Willig A 2007 Protocols and Architectures for Wireless Sensor Networks (Chichester: Wiley & Sons)
[3] Kamble A, Patil B 2021 Systematic analysis, and review of path optimization techniques in WSN with mobile sink Computer Science Review 41 100412
[4] Gomez-Rojas J, Camargo L, Montero R 2018 Mobile wireless sensor networks in a smart city International Journal on Smart Sensing and Intelligent Systems 11(1) 1
[5] Rueda J, Portocarrero J 2017 Similitudes y diferencias entre redes de sensores inalámbricas e internet de las cosas: hacia una postura clarificadora Revista Colombiana de Computación 18(2) 58
[6] Institute of Electrical and Electronics Engineers (IEEE) 2020 *IEEE Standard for Low-Rate Wireless Networks, IEEE 802.15.4-2020* (United States of America: Institute of Electrical and Electronics Engineers)

[7] Organization for the Advancement of Structured Information Standards (OASIS) 2019 *MQTT Version 5.0* (United States of America: Organization for the Advancement of Structured Information Standards)

[8] Kishorebabu V, Sravanthi R 2020 Real time monitoring of environmental parameters using IOT *Wireless Personal Communications* 112(2) 785

[9] Abraham M, Satyam N, Rosi A, Pradhan B, Segoni S 2020 The selection of rain gauges and rainfall parameters in estimating intensity-duration thresholds for landslide occurrence: case study from Wayanad (India) *Water* 12(4) 1000

[10] Sympa P 2019 Dynamic real-time volumetric correction for tipping-bucket rain gauges *Agricultural and Forest Meteorology* 271 158

[11] Liao M, Liao A, Liu J, Cai Z, Liu H, Ma T 2021 A novel method and system for the fast calibration of tipping bucket rain gauges *Journal of Hydrology* 597 125782

[12] Lozada J 2014 Investigación aplicada: definición, propiedad intelectual e industria *CienciAmérica: Revista de Divulgación científica de la Universidad Tecnológica Indoamérica* 3(1) 47

[13] Texas Instruments (TI) 2009 *ADS111x Ultra-Small, Low-Power, FC-Compatible, 860-SPS, 16-Bit ADCs With Internal Reference, Oscillator, and Programmable Comparator* (United States of America: Texas Instruments, Inc.)

[14] U-blox 2015 *NEO-M8, U-blox M8 Concurrent GNSS Modules* (Switzerland: U-blox)

[15] Raspberry Pi Trading 2019 *Raspberry Pi 4 Model B Datasheet* (The United Kingdom: Raspberry Pi Trading Ltd.)

[16] Mott R 2006 *Mecánica de Fluidos Aplicada* (Mexico: Pearson)

[17] European Computer Manufacturers Association (ECMA International) 2017 *The JSON Data Interchange Syntax, ECMA-404* (Switzerland: European Computer Manufacturers Association)

[18] Digi International (DIGI) 2018 *Digi Xbee-pro® 900HP Datasheet* (United States of America: Digi International, Inc.)

[19] Digi International (DIGI) 2020 *Digi XBee 3 DigiMesh 2.4 Datasheet* (United States of America: Digi International, Inc.)

[20] ZigBee Alliance (ZigBee) 2016 *Base Device Behavior Specification, Version 1.0* (United States of America: ZigBee Alliance, Inc.)

[21] ZigBee Alliance (ZigBee) 2014 *ZigBee Smart Energy Standard* (United States of America: ZigBee Alliance, Inc.)

[22] Eclipse Foundation 2021 *Eclipse Mosquitto 2.0.12* (Canada: Eclipse Foundation)

[23] MongoDB 2021 *MongoDB 5.0.3* (United States of America: MongoDB, Inc.)

[24] Python Software Foundation 2019 *Python 3.8.12* (Beaverton: Python Software Foundation)

[25] Internet Engineering Task Force (IETF) 2011 *The WebSocket Protocol, RFC 6455* (United States of America: Internet Engineering Task Force)

[26] Suseno H 2020 *Comparison of Wireless Sensor Network for Energy Monitoring System Between MQTT and Websocket Protocol* (Yogyakarta: Universitas Gadjah Mada)

[27] European Computer Manufacturers Association (ECMA International) 2021 *ECMAScript 2021 Language Specification, ECMA-262* (Switzerland: European Computer Manufacturers Association)

[28] Web Hypertext Application Technology Working Group (WHATWG) 2021 *HTML, Living Standard* (United States of America: Web Hypertext Application Technology Working Group)

[29] World Wide Web Consortium (W3C) 2016 *Cascading Style Sheets Level 2 Revision 2 (CSS 2.2) Specification* (United States of America: World Wide Web Consortium)