IoT-based Microseismic Monitoring System for the Evaluation of Structural Health in Smart Cities

Santiago González[1], Juan Carlos Jiménez[2,3], Remigio Guevara[3], Iván Palacios[3]

[1] Department of Electric, Electronic and Telecommunication Engineering
[2] Department of Civil Engineering, [3] Red Sísmica del Austro
Universidad de Cuenca, Av.12 de Abril, CP. 010203, Cuenca, Ecuador
santiago.gonzalezm, juan.jimenez, remigio.guevara,
ivan.palacios@ucuenca.edu.ec

Abstract. This paper presents the design and implementation of a prototype station, which aim to monitor microseismic events. The data collected by the sensors are used for detecting events on a specific zone, as well as for applications of structural health. The prototype was implemented using IoT devices (Internet of Things), such as MEMS accelerometers (Microelectromechanical Systems) and a single board computer with embedded Linux.

The station can be accessed remotely by means of a web application based on Node.js, which provides users with real time data. Moreover, the prototype has the capacity of getting information of geolocation as well as platform status (e.g. CPU load, temperature and energy consumption).

The main advantage of this proposal consists in the design of a solution based on open hardware architecture, open source, portability and low cost. Regarding the prototype evaluation, results show an appropriate operation, which represents the contribution of this work to the design of monitoring applications in the context of smart cities.

Keywords: Microseismic monitoring, Internet of Things, Structural health.

1 Introduction

Monitoring systems based on IoT technology (Internet of Thing) represent a current solution for the design of applications focused on capturing and processing physical variables. In particular, IoT-based systems allow the interconnection and cooperation between users, sensors and actuators.

Traffic monitoring systems, urban mobility management, environmental monitoring, emergency response systems, precision agriculture, among others, are some examples of applications [1]. These sorts of solutions contribute significantly to an urban development based on sustainability, security, and efficient resources management.

In this context, one of the most important applications is the monitoring and detection of natural events (e.g. seismic events), which can put both population and infrastructures at risk. In fact, it is worth mentioning that earthquake prediction is an active research area.

Although it is not possible to predict exactly the location, magnitude and time for a next seismic event to occur in a region, the seismology research working along with computer sciences and information technologies has significantly contributed to improving the understanding of the seismic processes, especially with regard to the temporal characteristics, as is described in [2]. In this case, the prediction is based on the analysis of seismic indicators also called seismic precursors. Such indicators are useful in order to generate emergency alerts. A well-known example about the success achieved by means of an early warning system is the case of the earthquake occurred in Haicheng (China, 1977). Seismic activity changes were monitored during months along with changes in groundwater levels, then when unusual behaviors were detected, an alert was generated to the population, an action that saved the lives of hundreds of thousands of people. According to the estimations detailed in [3], the number of victims was reduced to 1%.

In regard to the seismic precursors, there are studies in the literature which describe a relationship between seismic events and indicators: changes in seismic waves behavior [4], electrical signals presents in the lithosphere [5], electromagnetic changes in the ionosphere [6], temperature variations detected in the lithosphere [7], changes in the concentration of carbon dioxide in volcanic regions [8], radon emissions from soil [9], human activities (e.g. mining) [10], [11] and even unusual behaviors in animals [12].

Therefore, the level of complexity for deploying a monitoring system depend on the type of seismic precursor used. In particular, the analysis of seismic wave patterns, i.e. the detection and characterization of seismic events, presents a real potential for research and development of future solutions. To this end, the proper operation of transducer devices is highly important. In this context, sensors such as geophones and accelerometers show great versatility in order to detect ground motions.

Moreover, when a set of monitoring stations, including these kinds of sensors, is deployed along a specific region, it is possible to obtain microseismic information which can be useful as an indicator of a great magnitude earthquake.

In this paper, we present the design and implementation of a prototype station for microseismic monitoring. This work has been carried out as a first step towards the deployment of a monitoring network. Also, recent technology has been considered in order to reduce significantly the cost per station.

The prototype was implemented using IoT devices (Internet of Things), such as MEMS accelerometers (Microelectromechanical Systems) and a single board computer (SBC) with embedded Linux. The station can be accessed remotely by means of a web application based on Node.js, which provides users with real time data. Moreover, the prototype has the capacity of getting information about geolocation as well as platform status (e.g. CPU load, temperature and energy consumption).

The main advantage of this proposal consists in the design of a solution based on open hardware architecture, open source, portability and low cost. These characteristics are useful for both monitoring structural health and the microseismic analysis along specific zones. Regarding the prototype evaluation, results show an appropriate operation.

The paper is organized as follows: Section 2 presents related works. The design and implementation of the prototype, as well as the monitoring architecture, are detailed in Section 3. Section 4 describes the experimental evaluation and results. Finally, Section 5 presents conclusions and future lines of work.

2 Related Works

Monitoring systems focused on smart cities involve recent technological developments such as Internet of Thing [13] and communication architectures based on Wireless Sensor Networks (WSNs) [14]. In this context, risk management in order to mitigate disaster caused by seismic activity is one of the main tasks toward achieving sustainable urban development. The following are some of the most relevant and representative studies in the literature.

In [15], an early warning architecture is presented. This architecture takes advantage from sensors available in user devices (e.g. accelerometers in smartphones) in order to obtain acceleration data as well as to deploy a monitoring system. In [16], the authors analyze the importance regarding early warning systems implemented in Mexico after the devastating earthquake of 1985. On the other hand, these monitoring systems can be used to prevent disasters resulting from earthquakes, such as landslides or tsunamis. For example, [17] presents a set of projects in order to evaluate tsunami risks along the Mediterranean coast.

Regarding transducer devices, the accelerometers based on MEMS technology show high reliability and reduced size. MEMS technology comes from research on the design and manufacture of devices for navigation systems and space exploration [18]. In this case, the basic structure of a seismometer (i.e. the mass-spring system) is built within the multi-layer structure of an integrated circuit using a special process called micromachining. Currently, MEMS sensors are used in several research areas such as seismology, biomedicine, chemical analysis, wireless communications and robotics [19], [20].

In seismology, due to low cost and versatility, MEMS accelerometers represent an adequate alternative for implementing monitoring networks and early warning systems. Furthermore, some proposals that use this kind of accelerometers for seismic monitoring are detailed in [21], [22], [23] and [24]. These systems can be used in applications of structural health control focused on heritage buildings or even in communications infrastructures, as described in [25] and [26], respectively.

Additionally, [27] describes a system for detecting landslides caused by rock falls, a proposal that would be useful in order to improve road safety. Finally, there are some studies focused on monitoring the impact of seismic events on infrastructures required to provide basic services such as gas and water distribution networks, [28], [29].

On the other hand, regarding data acquisition and processing, accelerometers can be configured using a development platform type SBC, for example, Raspberry Pi

[30], Arduino [31], Beagle Bone [32], among others. Moreover, the interaction between a set of monitoring stations can be achieved by means of a communication architecture called Wireless Sensor Networks (WSNs) [33]. These sorts of networks have the capacity to self-configuring without the need of a central station or a preexisting infrastructure. Therefore, WSNs can be useful for both rural and urban environments [34]. Next, in Section 3, we present the system implementation.

3 System Architecture

This section describes the implementation of the prototype station as well as the main characteristics enabled for the microseismic monitoring system.

3.1 Prototype Implementation

The microseismic station was implemented using a development platform with embedded Linux (Raspberry Pi 3) [30] and accelerometers based on MEMS technology, Figure 1 shows the functional diagram of the station.

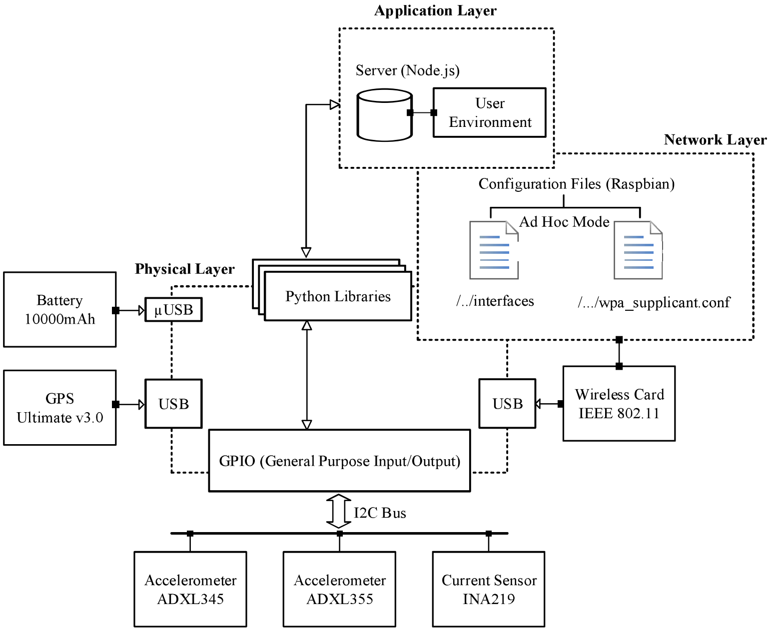


Fig. 1. Functional diagram of the microseismic station

As can be seen, the physical layer has five peripheral devices. In particular, the microseismic monitoring is performed by means of the Adxl345 [35] and Adxl355 accelerometers [36]. Table 1 describes the main technical characteristics for the accelerometers. The measurement range is specified in relation to the standard acceleration due to gravity (g, equivalent to 9.81m/s^2). Additionally, the current sensor INA219

[37] has been attached to the Raspberry Pi platform in order to assess the level of energy consumption demanded by the station. The sensors are handled by a set of open source Python libraries that are available at [38], [39], [40]. The connection to the platform is carried out via the I2C bus (Inter-Integrated Circuit) in the GPIO pins (General Purpose Input / Output). Moreover, with the intention of implementing a set of monitoring stations, a GPS sensor [41] was included in the prototype, particularly for synchronization tasks.

Table 1. Technical specifications for the accelerometers used in the prototype station

Parameter	Adxl345	Adxl355	Unit
Measurement range	±2, ±4, ±8, ±16	±2, ±4, ±8, ±16	g
Output resolution	10	20	bits
Sensitivity at ±2g	256	256000	LSB/g
Bandwidth	3200	1000	Hz
Current demanded	140	200	µA
Operating temperature	-40 a +85	-40 a +125	ºC

In regard to the network layer, we used an external wireless card (IEEE 802.11n), which was configured in ad hoc mode in order to allow the future integration of our station in a wireless sensor network. On the other hand, the wireless interface available on the platform board operates as a gateway for enabling Internet connection.

Moreover, the prototype is powered by a high current density lithium battery (10000 mAh) and for the remote management, we set up an HTTP server based on Node.js [42].

Finally, an electronic board was implemented in order to facilitate the interconnection between the sensors with the platform. Figure 2 shows the physical prototype implemented and the components.

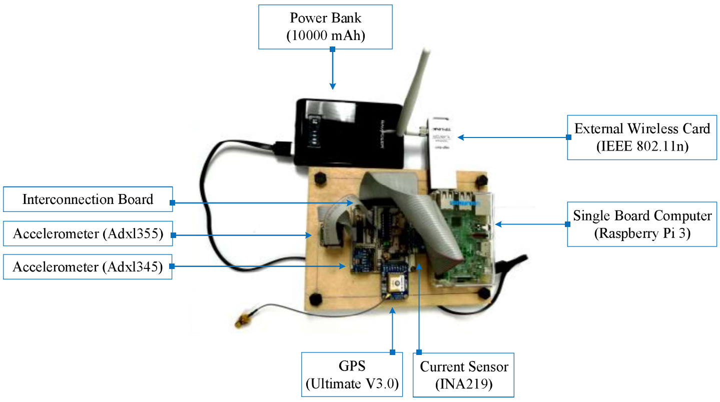


Fig. 2. Description of the microseismic station components

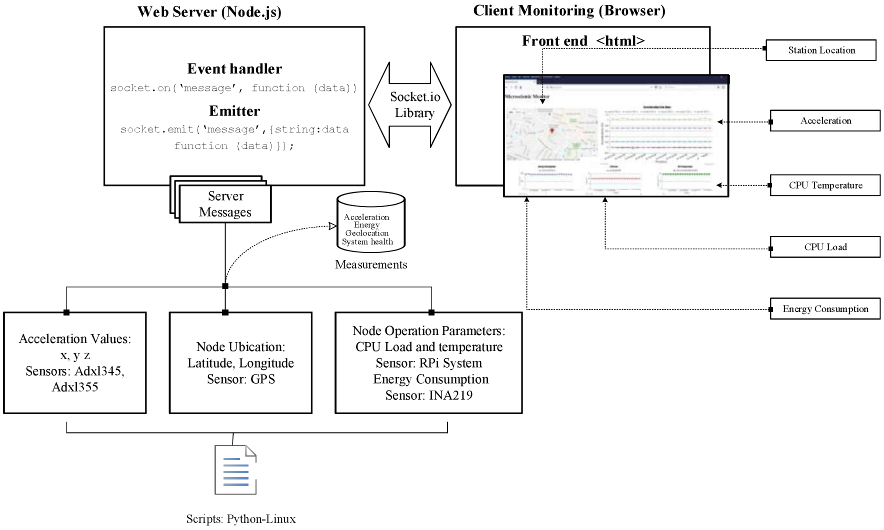
3.2 Microseismic Monitoring System

Figure 3 shows the functional diagram of the monitoring architecture. As can be seen, we developed a web application using Node.js, which allows the remote access to the station using a browser. Specifically, to monitor events as well as visualize the accelerometers information in real time, we have used WebSockets through the socket.io library [43].

In particular, a set of Python scripts were implemented in order to control the sensors operation. The data captured from the sensors are used to generate messages on the server side (*socket.emit* function). Then, the server sends to the remote client the information with the acceleration values detected (x, y and z). Additionally, the server sends messages about geolocation and energy consumption obtained from the GPS and from the current sensor, respectively. The send intervals for the messages can be configured or modified on the Python scripts. Similarly, to evaluate the system status, the server sends messages with information about temperature and CPU load, obtained in this case from the Raspberry Pi platform.

On the other hand, the web interface provides users with dynamic charts in order to detect instantaneous changes in the information from the sensors.

Finally, the sensor and system information is stored in the memory card of the station to keep a history of measures. The evaluation of the prototype station is detailed below.



```
socket.on('message', function (data))
```

```
socket.emit('message',{string:data
function (data)});
```

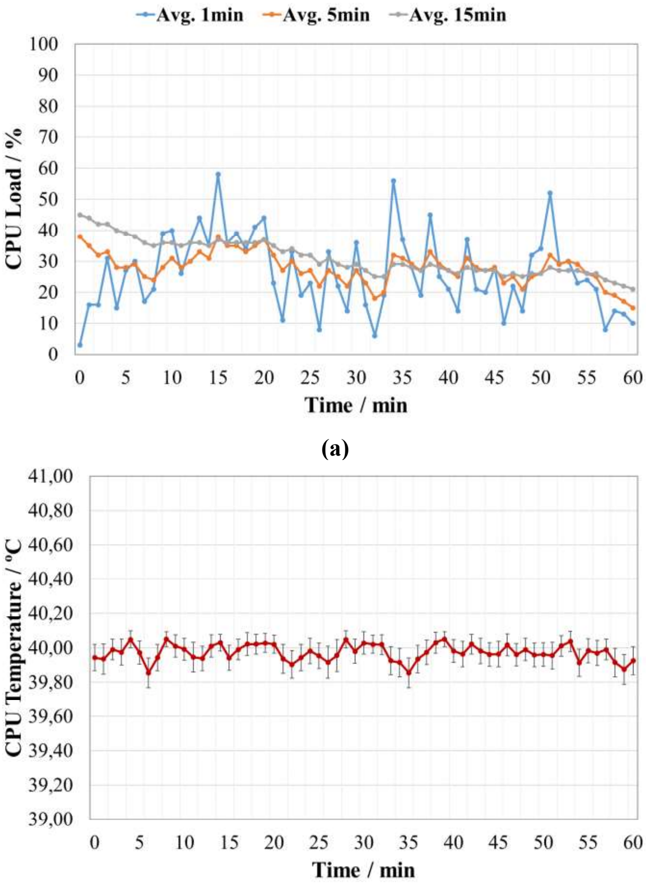
Fig. 3. Functional diagram: Architecture of the microseismic monitoring system

4 Experimental Evaluation and Results

This section presents a set of measurements that describes the station performance under operational conditions, i.e. during acquisition, processing and transmission of the data captured by the sensors.

4.1 Prototype Evaluation

In this experiment we carried out an evaluation about the load as well as the temperature measured on the CPU of the Raspberry Pi platform. To this end, we activated all processes that enable data acquisition from sensors (i.e. accelerometers, GPS and current sensor). The samples were captured at intervals of 1 second. Moreover, during the experiment, we also enable the web application for displaying information on the user interface. Figure 4 shows the results throughout one hour of operation.



(a)

(b)

Fig. 4. Prototype evaluation: (a) CPU load monitoring. (b) CPU temperature monitoring

Figure 4 (a) describes the behavior of the CPU load. In particular, the samples were captured by means of the *uptime* command that provides information about the system load averages for the past 1, 5, and 15 minutes. As can be seen, results show that the processes demanded by the sensors along with the web application does not involve overload on the CPU. Specifically, the maximum load level detected was approximately 60%, while the average load for the analyzed interval is close to 30%.

On the other hand, Figure 4 (b) presents the temperature monitored on the CPU, the confidence interval showed were stated at the 95% confidence level, it can be seen, there are minimal changes on the CPU temperature. In particular, the average value is closed to 40ºC, that is distant from the maximum value recommended in the technical specifications (70ºC). Also, it is worth indicating that the average ambient temperature along the experiment was 20ºC.

4.2 Energy Consumption

In the previous experiment, we also captured current samples in order to determine the energy consumption demanded by the prototype. Similarly, the samples were captured at intervals of 1 second. Figure 5 shows the results throughout one hour of operation.

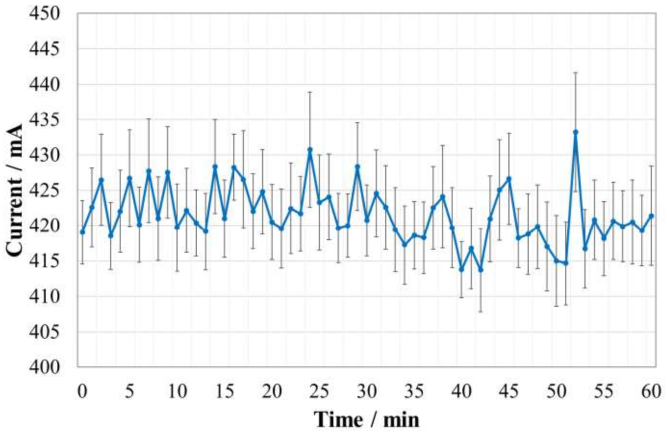


Fig. 5. Current samples captured during the prototype operation

In particular, it is worth noting the narrow confidence intervals (95% confidence level) meaning minimal variation among samples, specifically within a range of between 414mA and 433mA. In regard to the average current for the entire interval (computed from the samples), the value is 421mAh. Furthermore, we carried out additional experiments with the aim of analyzing the energy consumption by each device on the prototype. Figure 6 shows the results depending on the connected peripherals.

As can be seen, the energy demanded by the platform along with the accelerometers (i.e. without the GPS sensor), presents a consumption of approximately 385mAh.

On the other hand, when the prototype station uses one accelerometer (Adxl355 sensor), it reduces the energy expenditure to 379mAh. In regard to the platform and the external wireless card (i.e. without accelerometers), results show a consumption of 370mAh. Finally, it can be seen that the platform presents a base consumption of approximately 216mAh.

Accordingly, taking into account the battery capacity used to power the prototype (10000mAh), then the operation time ranges from 23 to 26 hours. The estimated autonomy is useful for setting up outdoor experiments.

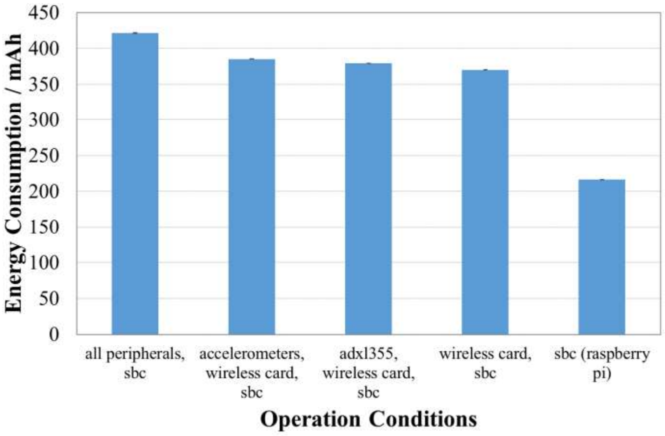


Fig. 6. Average energy demanded by the prototype under different operational conditions

4.3 Microseismic Monitoring

The characterization of the accelerometers, was carried out by means of multiple tests. Also, the prototype was enabled to operate in a continuous data acquisition mode. This methodology was defined to evaluate the sensitivity level for detecting seismic events.

Figure 7 shows the results obtained using the Adxl355 accelerometer throughout one hour of operation. Also, it is worth clarifying that for the results presented, we performed a baseline correction. This procedure consists in removing the direct component (DC value) from the acquired signal (i.e. the data offset with regard to the zero level of the acceleration).

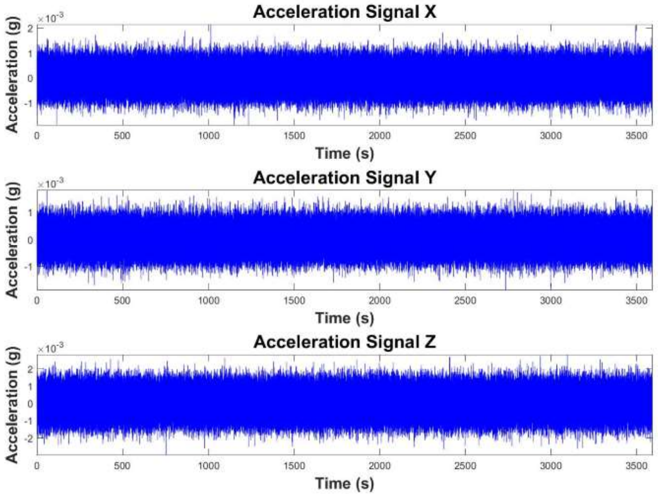


Fig. 7. Acceleration records (x, y, z) vs. time, obtained by the Adxl355 sensor

As shown, the graphics describe the acceleration values detected by the sensor on the x, y and z axes. It is important to point out that along the tests, there were no reported seismic events close to the facilities where the station was located (central campus, University of Cuenca). Consequently, the acceleration samples captured correspond to the background noise or urban seismic noise. Additionally, the signal behavior can be seen in more detail in Figure 8, which shows an interval corresponding to 1 minute of the acquired data. In particular, the maximum variations have been reported to range between approximately ±0.002g. This threshold represents the prototype sensitivity taking into account the location selected to carry out the experiments.

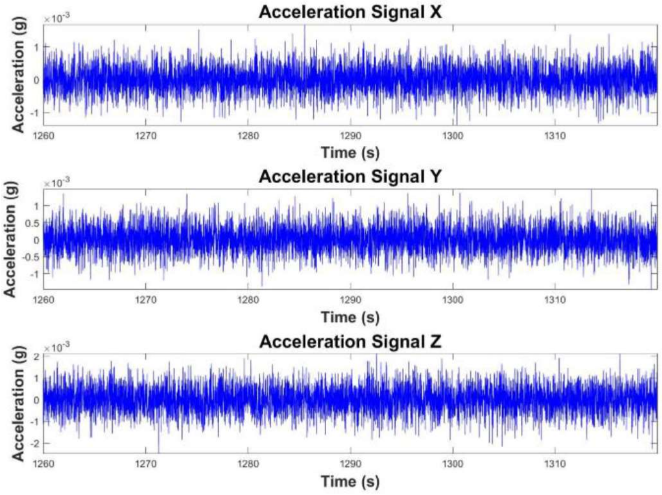


Fig. 8. Acceleration records corresponding to an interval of 1 minute

5 Conclusions

In this paper, we presented the design of a prototype station to evaluate microseismic events. Specifically, the prototype was designed and implemented using a single board computer as well as accelerometers based on MEMS technology, which significantly reduce the overall cost per station. Also, a web application was implemented in order to monitor the information captured by sensors in real time.

Regarding the prototype evaluation, experiments carried out show an appropriate operation. In particular, a maximum CPU load of 60% was detected due to simultaneous operation of peripherals. Also, the average value of temperature was approximately of 40ºC, which is distant from the threshold value recommended in the technical specifications (70ºC).

Moreover, a current sensor was included in order to characterize the energy expenditure as well as to estimate the autonomy of operation. Results indicate a lifetime between 23 to 26 hours, depending on the number of peripherals connected.

In regard to the experiments to evaluate the accelerometers sensitivity, the results show that our prototype allows to detect seismic events that cause accelerations higher than 0.002g. Although, this sensitivity is influenced by the site where the prototype is located, we consider that the value is suitable for applications of structural health.

Finally, as future work we plan to implement a set of stations with the aim to deploy a microseismic monitoring network.

Acknowledgements

The authors gratefully acknowledge the support of the Research Management Department of the University of Cuenca- Ecuador (DIUC) and the research group member of the Thematic Network Ciudades Inteligentes Totalmente Integrales, Eficientes y Sostenibles (CITIES-CYTED).

References

[1] B. Rashid and M. H. Rehmani, “Applications of wireless sensor networks for urban areas: A survey”, *J. Netw. Comput. Appl.*, vol. 60, pp. 192–219, 2016.

[2] K. F. Tiampo and R. Shcherbakov, “Seismicity-based earthquake forecasting techniques: Ten years of progress”, *Tectonophysics*, vol. 522–523, pp. 89–121, 2012.

[3] M. Wyss and Z. Wu, “How Many Lives were Saved by the Evacuation before the M 7.3 Haicheng Earthquake of 1975?”, *Seismol. Res. Lett.*, vol. 85, no. 1, pp. 126–129, 2014.

[4] E. Florido, F. Martínez-Álvarez, A. Morales-Esteban, J. Reyes, and J. L. Aznarte-Mellado, “Detecting precursory patterns to enhance earthquake prediction in Chile”, *Comput. Geosci.*, vol. 76, pp. 112–120, 2015.

[5] M. S. Lazaridou-Varotsos, *Earthquake prediction by seismic electric signals: The success of the VAN method over thirty years*. 2013.

[6] M. Akhoondzadeh, “Firefly Algorithm in detection of TEC seismo-ionospheric anomalies”, *Adv. Sp. Res.*, vol. 56, no. 1, pp. 10–18, 2015.

[7] A. Bhardwaj *et al.*, “A review on remotely sensed land surface temperature anomaly as an earthquake precursor”, *Int. J. Appl. Earth Obs. Geoinf.*, vol. 63, no. September 2016, pp. 158–166, 2017.

[8] L. Pierotti, F. Gherardi, G. Facca, L. Piccardi, and G. Moratti, “Detecting CO2 anomalies in a spring on Mt. Amiata volcano (Italy)”, *Phys. Chem. Earth*, vol. 98, pp. 161–172, 2017.

[9] P. Bormann, “From earthquake prediction research to time-variable seismic hazard assessment applications”, *Pure Appl. Geophys.*, vol. 168, no. 1–2, pp. 329–366, 2011.

[10] A. Carpinteri and O. Borla, “Fracto-emissions as seismic precursors”, *Eng. Fract. Mech.*, vol. 177, pp. 239–250, 2017.

[11] A. Janusz, M. Grzegorowski, M. Michalak, Ł. Wróbel, M. Sikora, and D. Ślęzak, “Predicting seismic events in coal mines based on underground sensor measurements”, *Eng. Appl. Artif. Intell.*, vol. 64, no. April, pp. 83–94, 2017.

[12] R. A. Grant and T. Halliday, “Predicting the unpredictable; evidence of pre-seismic anticipatory behaviour in the common toad”, *J. Zool.*, vol. 281, no. 4, pp. 263–271, 2010.

[13] B. N. Silva, M. Khan, and K. Han, “Towards sustainable smart cities: A review of trends, architectures, components, and open challenges in smart cities”, *Sustain. Cities Soc.*, vol. 38, no. January, pp. 697–713, 2018.

[14] A. Belghith and M. S. Obaidat, *Wireless Sensor Networks Applications to Smart Homes and Cities*. Elsevier Inc., 2016.

[15] A. M. Zambrano, I. Perez, C. Palau, and M. Esteve, “Technologies of Internet of Things applied to an Earthquake Early Warning System”, *Futur. Gener. Comput. Syst.*, vol. 75, pp. 206–215, 2017.

[16] R. R. Castro *et al.*, “A review on advances in seismology in Mexico after 30 years from the 1985 earthquake”, *J. South Am. Earth Sci.*, vol. 70, no. June 1911, pp. 49–54, 2016.

[17] Papadopoulos, *Tsunami Early Warning Systems and Risk Mitigation*. Elsevier Inc., 2016.

[18] NASA, “‘Microdevice Laboratory,’ MEMS Inertial Technology.” [Online]. Available: https://microdevices.jpl.nasa.gov/capabilities/nano-and-micro-systems/mems-gyroscope.php. [Accessed: 27-Jun-2018].

[19] E. Y. Chow, S. P. Sanghani, and V. Ramesh, *Emerging research in wireless and MEMS for medical applications*. Elsevier Ltd., 2016.

[20] R. Osiander and A. G. Darrin, *Wireless MEMS for space applications*. Elsevier Ltd., 2016.

[21] D. Chen *et al.*, “A micro electrochemical seismic sensor based on MEMS technologies”, *Sensors Actuators, A Phys.*, vol. 202, pp. 85–89, 2013.

[22] Z. Shen, C. Y. Tan, K. Yao, L. Zhang, and Y. F. Chen, “A miniaturized wireless accelerometer with micromachined piezoelectric sensing element”, *Sensors Actuators, A Phys.*, vol. 241, pp. 113–119, 2016.

[23] G. Li *et al.*, “A MEMS based seismic sensor using the electrochemical approach”, *Procedia Eng.*, vol. 47, pp. 362–365, 2012.

[24] S.Sindhuja and J. Siluvai, “MEMS-based wireless sensors network system for post-seismic tremor harm evaluation and building monitoring”, pp. 1–4, 2015.

[25] E. Lenticchia, R. Ceravolo, and C. Chiorino, “Damage scenario-driven strategies for the seismic monitoring of XX century spatial structures with application to Pier Luigi Nervi’s Turin Exhibition Centre”, *Eng. Struct.*, vol. 137, pp. 256–267, 2017.

[26] A. Ismail, “Seismic assessment of guyed towers: A case study combining field measurements and pushover analysis”, *HBRC J.*, vol. 12, no. 1, pp. 1–7, 2014.

[27] V. L. Zimmer and N. Sitar, “Detection and location of rock falls using seismic and infrasound sensors”, *Eng. Geol.*, vol. 193, pp. 49–60, 2015.

[28] C. Wang, W. Liu, and J. Li, “Artificial earthquake test of buried water distribution network”, *Soil Dyn. Earthq. Eng.*, vol. 79, pp. 171–185, 2015.

[29] H. Miao, W. Liu, C. Wang, and J. Li, “Artificial earthquake test of gas supply networks”, *Soil Dyn. Earthq. Eng.*, vol. 90, no. February, pp. 510–520, 2016.

[30] A. Lady, “Introducing the Raspberry Pi 2 - Model B”, 2015.

[31] M. Banzi, *Getting Starting with Arduino*, Second Edi. 2011.

[32] D. Molloy, *Exploring BeagleBone:Tools and Techniques for Building with Embedded Linux*. 2015.

[33] M. C. and S. Giordano, “Mobile ad hoc networking: milestones, challenges, and new research directions”, *IEEE Commun. Mag.*, vol. 52, no. 1, pp. 85–96, 2014.

[34] Aqeel-Ur-Rehman, A. Z. Abbasi, N. Islam, and Z. A. Shaikh, “A review of wireless sensors and networks’ applications in agriculture”, *Comput. Stand. Interfaces*, vol. 36, no. 2, pp. 263–270, 2014.

[35] A. Devices, “Data Sheet ADXL345”. [Online]. Available: www.analog.com/media/en/technical.../data.../ADXL345.pdf. [Accessed: 01-May-2018].

[36] Analog Devices, “Low Noise, Low Drift, Low Power, 3-Axis MEMS Accelerometers ADXL354/ADXL355”, 2016.

[37] A. Lady, “Adafruit INA219 Current Sensor”, pp. 1–19, 2017.

[38] GitHub, “Python library for the ADXL345”, [Online]. Available: https://github.com/pimoroni/adxl345-python. [Accessed: 27-Jun-2018].

[39] GitHub, “Python 3 library to connect ADXL355 PMDZ accelerometer on Raspberry Pi”. [Online]. Available: https://github.com/gpvidal/adxl355-python.

[40] GitHub, “Python library INA219 current sensor”. [Online]. Available: https://github.com/scottjw/subfact_pi_ina219. [Accessed: 27-Jul-2018].

[41] A. Lady, “Adafruit Ultimate GPS”, 2017.

[42] G. Ornbo, *Programación Node.js*. Anaya Multimedia, 2013.

[43] GitHub, “Realtime application framework (Node.JS server)”. [Online]. Available: https://github.com/socketio/socket.io. [Accessed: 27-Jun-2018].