International Journal for Innovation Education and Research

ONLINE ISSN: 2411-2933 PRINT - ISSN: 2411-3123

Low cost Device for Online Monitoring of Noise in Libraries using Internet of Things

Antonio Leandro Martins Candido;Sandro Cesar Silveira Jucá;Renata Imaculada Soares Pereira;Paulo Marvin de Brito Lima;Solonildo Almeida da Silva

Abstract

The present paper describes the development and practical application of an embedded system that performs the online monitoring of sound noises in a library of an educational institution using Internet of Things. The main objective of the proposed project is to record sound levels in closed areas and to alert users when allowed limits were exceeded. Thus, it was used a set of cost-effective tools, like free software and low-cost technologies. Raspberry Pi, PIC microcontroller and a sound sensor module were applied in the first phase. In the second phase, only ESP8266 microcontroller with sound sensor were used. Results of these two implemented phases are discussed. It was observed that both developed embedded systems, which use Internet of Things concept, contributed satisfactorily offering a better space for concentration.

Keyword: Embedded System; Internet of Things; Sound Noises.

Published Date: 8/31/2018 **Page.133-144** **Vol 6 No 08 2018**

DOI: https://doi.org/10.31686/ijier.Vol6.Iss8.1127

Low cost Device for Online Monitoring of Noise in Libraries using Internet of Things

Antonio Leandro Martins Candido[1], Sandro Cesar Silveira Jucá[1], Renata Imaculada Soares Pereira[2*], Paulo Marvin de Brito Lima[1], Solonildo Almeida da Silva[3]

[1] Computing Department, Federal Institute of Ceará (IFCE) - Maracanaú Campus, Brazil

[2] Electrical Engineering Department, Federal University of Ceará (UFC) - Pici Campus, Brazil

[3] Pedagogical Department, Federal Institute of Ceará (IFCE) - Maracanaú Campus, Brazil

*Corresponding Author: renata@dee.ufc.br; Tel. +5585 3366.9585

Abstract

The present paper describes the development and practical application of an embedded system that performs the online monitoring of sound noises in a library of an educational institution using Internet of Things. The main objective of the proposed project is to record sound levels in closed areas and to alert users when allowed limits were exceeded. Thus, it was used a set of cost-effective tools, like free software and low-cost technologies. Raspberry Pi, PIC microcontroller and a sound sensor module were applied in the first phase. In the second phase, only ESP8266 microcontroller with sound sensor were used. Results of these two implemented phases are discussed. It was observed that both developed embedded systems, which use Internet of Things concept, contributed satisfactorily offering a better space for concentration.

Keywords: Embedded System; Internet of Things; Sound Noises.

1. Introduction

The NBR 10152 regulation of the Brazilian Association of Technical Standards (ABNT) - "Noise Levels for Acoustic Comfort", establishes values between 35 and 45 decibels (dB) for places as a library and guides dB values above this level do not necessarily entail risks of harm to health, but are considered acoustic discomfort [1].

Although 45 dB is, in practical terms, relatively low and difficult to maintain in a space where there is usually a lot of movement, it is recognized that levels far above this value generate discomfort for those who intend to establish a reading activity. Silence is a goal pursued in the reading environments, since a quiet and pleasant environment favors the study [2]. However, absolute silence in an environment in which there is human activity is impractical. For this reason, efforts should be to educate users that there are tolerable noise levels for environments such as a library.

The development of the embedded system for online monitoring of sound noises originated after the knowledge of the needs of the users of libraries in general and the main one of them is the predominance of the silence in the environment [3].

Thus, to implement techniques that help in the control of these levels of noise is necessary, since such noises affect the concentration in these environments and, in a certain way, exclude the real purpose of these places, which is to provide a suitable place for the studies.

The embedded systems presented in this project has as main objective to assist in the control of the noise levels in such environments, thus being able to supply the users needs. From a low cost and always prioritizing to use free technologies and tools, this system can be used by other institutions or researchers and can be modified as needed.

2. Material and methods

According to [4], users of a library have diverse needs. They use not only the collection disposed, but also the space and equipment. Regarding spaces, at least two different needs can be identified: individual or group study/reading, being essential the noise level control in reading and individual study environments, where the aid of electrical and electronic devices presents to libraries an alternative to combating noise pollution.

Environmental comfort in libraries is little studied in Brazil, especially when referring to noise levels and user satisfaction [5]. These concerns are often neglected in the country even though noises cause a reduction of up to 60% in productivity, as it makes it difficult to concentrate, causing errors, waste due to distraction. In Brazil, libraries in general are adaptations of existing buildings that have these functions, although these results are not the adequate, some professionals architect a space that is basically summarized in two environments: the deposit of books and the place of reading.

The embedded systems proposed use principles of Internet of Things (IoT). IoT is an extension of the Internet, which gives everyday objects (“things”), but with computational and communication skills, connection to the Internet [6]. The Internet connection enables to remotely control the objects and to allow them to be accessed as service providers.

The Internet currently connects people and enterprises through computers and computational devices of any shape and capacity [7] and can collect large amounts of data [8].

3. Methodology

The methodology used to develop this project was firstly to carry out an experimental research in the library of an education institution. During the development of this embedded system, some equipment, devices, technologies and software were tested to find the most cost-effective and with less difficulty so that the system could be replicated.

The two proposed embedded systems for noise monitoring are described in the next subsections. The second phase is developed to reduce costs, applying ESP8266 IoT module (US$ 3.00) to replace both Raspberry Pi (US$ 35.00) and PIC microcontroller (US$ 4.00), quotation for 2018.

3.1 First version of the proposed system

In this section, an initial possibility of implementing the embedded system for monitoring noise is proposed. Firstly, it was implemented with Raspberry Pi (RPi), considered the smallest computer in the world, with the size of a credit card, USB connections to connect keyboard and mouse used in desktop computers. HDMI output is available, as can be seen in Figure 1 along with the description of the other connections. The embedded software Debian is free, based on Linux [9].

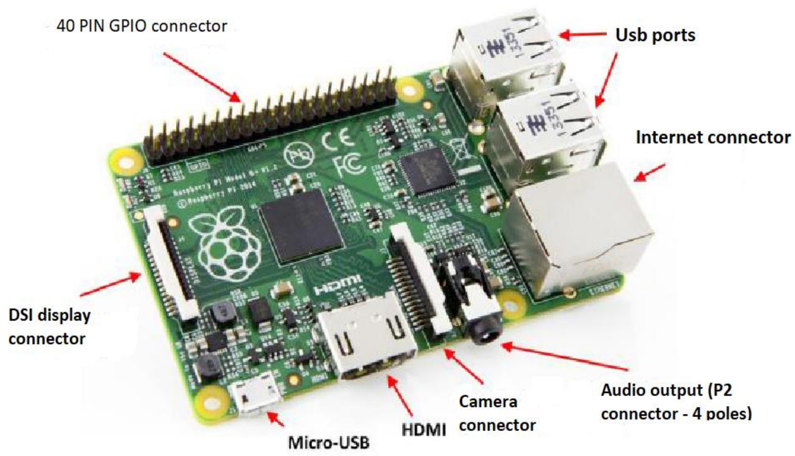

Figure 1. Raspberry Pi B+ [9].

The RPi used is the B+ Model has the following features:

- More GPIO. The GPIO header has grown to 40 pins while retaining the same pinout for the first 26 pins as model A and B;
- More USB ports. Four USB 2.0 ports, compared to 2 of Model B;
- Micro SD. The old friction-fit SD card socket has been replaced by a push-pull micro SD plug;
- Lower energy consumption. By replacing the linear regulators, power consumption is reduced with power between 0.5 W and 1W.
- Better audio quality. The audio circuitry incorporates a dedicated low-noise power supply;

The Model B+ is suited for use in schools and academic projects, offers greater flexibility than the simpler A or A+ model, is more useful for integrated projects that require very low power, and require more USB ports than model B [10].

Considering that RPi has no analog-digital converter (ADC), a PIC18F2550 microcontroller for this first proposed version was selected (pin diagram shown in Figure 2). It is a complete computer system, which includes internally a CPU (Central Processor Unit), RAM (data), flash (program memory) and EEPROM, I/O pins (Input/Output), plus other internal peripherals such as oscillators, USB channel, USART asynchronous serial interface, timing modules and A/D converters, among others, integrated in the same component (chip)

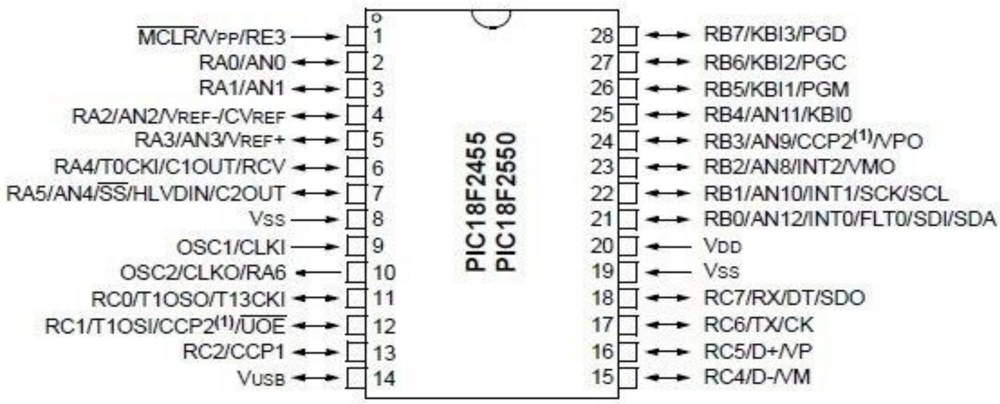

Figure 2. PIC18F2550.

In this version of the system the features offered by the Raspberry Pi GPIO pin set must be used, where values are received on the Rx serial pin (pin 18) and pin 7 is used to trigger the buzzer. These pins can be switched as either input or output. In other cases, each pin can be flexibly configured to accept different logic voltage sources. Figure 3 shows the table of the RPi GPIO pins.

The Raspberry Pi was used to communicate via serial interface with the PIC microcontroller, and the microcontroller received the data from an LM393 sound sensor module, which did not have a good reading of the noise changes in the environment, especially at longer distances. This module was later replaced. LM393 communicates with the microcontroller through an ADC input. The purpose of this sensor is to measure the sound intensity in the environment by varying the state of its digital and analog output if a beep is detected. It has an electric condenser microphone and can be used in alarm systems for example.

pi@raspberrypi gpio readall

B Plus

BCM	wPi	Name	Mode	V	Physical	Physical	V	Mode	Name	wPi	BCM
		3.3v			1	2			5v		
2	8	SDA.1	IN	1	3	4			5V		
3	9	SCL.1	IN	1	5	6			0v		
4	7	GPIO. 7	IN	0	7	8	1	ALT0	TxD	15	14
		0v			9	10	1	ALT0	RxD	16	15
17	0	GPIO. 0	IN	0	11	12	0	IN	GPIO. 1	1	18
27	2	GPIO. 2	IN	0	13	14			0v		
22	3	GPIO. 3	IN	0	15	16	0	IN	GPIO. 4	4	23
		3.3v			17	18	0	IN	GPIO. 5	5	24
10	12	MOSI	IN	0	19	20			0v		
9	13	MISO	IN	0	21	22	0	IN	GPIO. 6	6	25
11	14	SCLK	IN	0	23	24	0	IN	CE0	10	8
		0v			25	26	0	IN	CE1	11	7
0	30	SDA.0	IN	0	27	28	0	IN	SCL.0	31	1
5	21	GPIO.21	IN	0	29	30			0v		
6	22	GPIO.22	IN	0	31	32	0	IN	GPIO.26	26	12
13	23	GPIO.23	IN	0	33	34			0v		
19	24	GPIO.24	IN	0	35	36	0	IN	GPIO.27	27	16
26	25	GPIO.25	IN	0	37	38	0	IN	GPIO.28	28	20
		0v			39	40	0	IN	GPIO.29	29	21
BCM	wPi	Name	Mode	V	Physical	Physical	V	Mode	Name	wPi	BCM

B Plus

Figure 3. Raspberry Pi GPIO Pinboard Table.

The detection limit can be set via a potentiometer in the module that regulates the digital output D0. However, for a better resolution it is possible to use the analog output A0 and connect to the ADC of the PIC18F2550 microcontroller. LM393 module specifications can be seen in Figure 4.

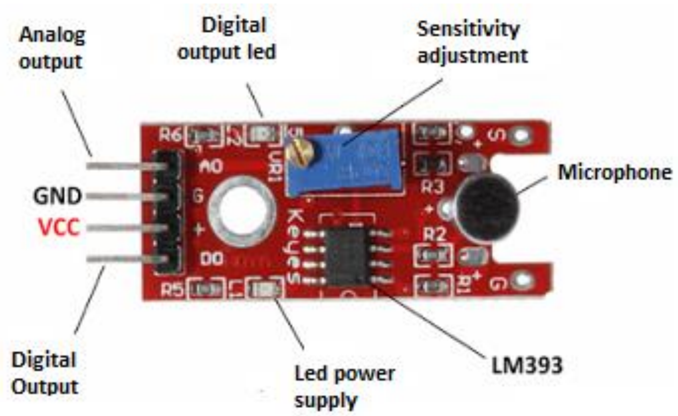

Figure 4. LM393 Sound Sensor Module, Model KY-038.

The microcontroller sends the data to the RPi through the TX serial pin, after the data acquisition, the code sends this data to a MySQL database hosted in the Cloud. Thus, this project is based on the Internet of Things (IoT) concept. A C library developed by Gordon was used to access the GPIO ports of RPi called "WiringPi" [11]. To avoid having to use an internal loop inside the Raspberry Pi program, where the values would be constantly sent to the database, the technique offered by Crontab, a Linux application capable of scheduling the execution of tasks in the operating system, was used. In this schedule it was set for the Linux operating system to run the RPi program every five seconds.

In addition to the program running on RPi, a program (firmware) was also developed to run on the PIC18F2550 microcontroller. Figure 5 highlights the steps required to burning this program in the microcontroller, where the free SanUSB educational platform to manage this recording was used.

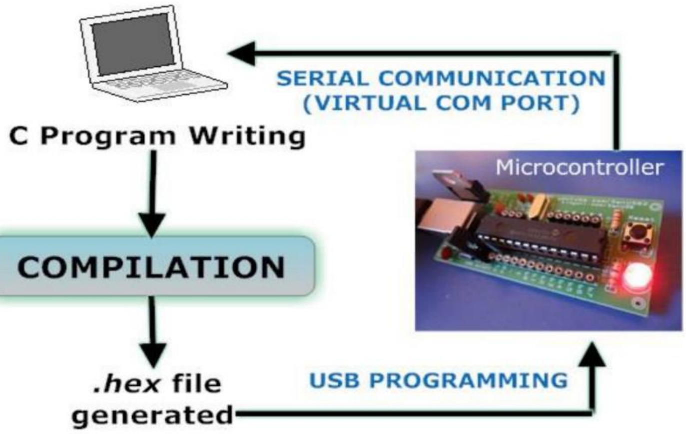

Figure 5. Recording PIC microcontroller through SanUSB [12].

The SanUSB tool, also applied to intelligent embedded systems, such as the online microgeneration monitoring via WiFi modem proposed in [13], is characterized as an educational application software with technological purpose that emphasizes concepts related to the practical application, which contributes to the assimilation of the content addressed [14]. This educational software was used by students during the 4th Robotics Competition of the IFCE (Federal Institute of Education, Science and Technology of Ceará-Brazil) and in the Engineering Category of the Brazilian Science and Engineering Fair (FEBRACE 09) held at University of São Paulo (USP). The students, without the need to consult the teachers, obtained the first places in both competitions. This fact presented satisfactory indications related to the didactic efficiency of software interaction. In addition, the placements achieved also provided satisfactory indications as to the technological contribution of the software in the general performance of the students [12].

In addition to the program for the microcontroller that realized the sensorial reading; the RPi program that received this sensor readings and posted in a database besides triggering a buzzer (sound device) when the allowed limit was exceeded, a web page hosted in the Cloud was also developed for the real time presentation of the obtained values from this embedded system.

The goal was achieved using the tools described above, but a new challenge arose, making a better cost-effective embedded system. This fact is due to the RPI cost in average of US$ 35.00 in 2018, which is still low cost for a computational embedded system, but it offers more resources than the necessary for the implementation of the noise detection proposed application. Faced with such a situation, a research was done on new devices that could replace those already used, maintaining the purpose of the embedded system and lowering its cost of acquisition.

3.2 Second version of the proposed system

It is possible to acquire a small board today where 11 input and output pins are available, with voltage regulation circuits, USB connectivity for programming, which can be done in Lua programming or by Arduino IDE, WiFi connectivity through the module ESP8266 12- E, standard 802.11 wireless, GPIO with PWM functions, I^2C, SPI, operating voltage: 4.5 to 9V, transfer rate of 110-460800bps, Supports 5 TCP/IP connections and costs around US$ 3.00 up to a maximum of US$ 4.00. This is the NodeMCU Esp8266 Wifi Module, however, it is characterized a self-sufficient development board for IoT projects, currently considered one of the most used. The NodeMCU module is one of the most complete and simple boards to use. This means that the additional use of other components, such as PIC microcontroller and RPi, used previously in the first version of the proposed project is no longer necessary.

In this IoT project, all sensor data processing is performed on the NodeMCU ESP8266 and transmitted through WiFi to the platforms in the Wireless Monitor web page developed or to the ThingSpeak Cloud server.

The Wireless Monitor web platform aims to enable embedded system developers to send the data obtained by IoT equipment to the Cloud and view it in the internet browser. Being an open source application licensed by the GPLv3 (GNU Public License), it can be used by both teachers and students of advanced courses and technicians to study microcontrollers, embedded systems as well as by companies or people who want to interact with their personal equipment [15]. The platform was developed using the Laravel framework plugin features [16], and new plugins can be created according to the users needs.

ThingSpeak is an IoT platform that lets you collect and store sensor data in the Cloud and develop IoT applications. The ThingSpeak platform provides applications that allow analyzing and visualize data in MATLAB and then act on the data. The sensor data can be sent to the ThingSpeak from the NodeMCU, Arduino, RPi, BeagleBone Black and other embedded systems with embedded WiFi module [17].

These IoT platforms are responsible for displaying the data acquired on the web. With the launch of NodeMCU, new research and applications have been started, since these same applications have become more robust and inexpensive, for example, applications such as for smart cities, smart garbage collection, as shown in [18], applications that use Radio Frequency Identification (RFID) [19], some applications in artificial intelligence as shown in [20]. Even in the smart health area, such as biofeedback control in HealthCare [21], in agriculture, such as hydroponics projects and in fish farming, where aquaponics projects also use the NodeMCU [22].

To measure the volume of the sound and to be able to compare different measures, it was necessary to use a module where the gain is configurable, without changing automatically. Four sound modules were evaluated, and their comparisons are presented in Table 1. The low efficiency HXJ17 was replaced by the TK0862, and then replaced by KY038. In the final tests, MAX4466 had the best results.

Table 1: Comparison between sound sensor modules.

Model	Adjustable gain	Perception distance	Price
MAX4466	Yes	+ 10 m	≈ US$7.50
KY038	Yes	≈ 1 m	≈ US$2.70
TK0862	Yes	≈ 30 cm	≈ US$3.10
HXJ17	Yes	≈ 5 cm	≈ US$3.30

The MAX4466 sound sensor module showed a greater perception capacity of alterations in noise levels, making it the best choice among the four evaluated modules, mainly for detecting changes of levels noise at greater distances (more than 10 m). This module has an adjustable gain that can be controlled with a small potentiometer of one turn. There is a Vcc pin, a grounding pin and an analog output pin. The analog pin emits a waveform where "0" is Vcc/2 and the amplitude depends on the gain and volume of the sound. It has optimum performance because the MAX4466 is an operating amplifier specifically optimized for use as a microphone amplifier [23].

The audio signal from the MAX4466 output is a variable voltage. To measure the sound level, several steps to find the minimum and maximum extensions or "peak-to-peak amplitude" of the signal are needed. In the program developed in C language, using the Arduino IDE, an example window of 50 milliseconds was chosen, this was enough to measure sound levels of frequencies of at least 20 Hz (the lower limit of human hearing). After finding the minimum and maximum samples, the difference was calculated and converted into Volts, which could range from 0 to 3.3V [24]. After performing this calculation, the resulting values were compared with values obtained through the Sound Meter decibel meter application, developed for the Android operating system by the company Abc Apps. This application has great precision in the calculation of the decibels in the environment and has already been used by more than ten million users. To make this comparison the PRO version of the application was acquired [25].

It is worth mentioning that the ideal is to use 3.3V as the reference voltage. The reason for this is that the 3.3 V is usually more stable than the 5 V which can vary up and down [26], especially when the NodeMCU is getting its power from the USB connection. The NodeMCU uses a 3.3V liner regulator, this calibrates the MAX4466's ADC output to map the 0-3.3V range to the analog input of the board.

The architecture of the embedded system after board mounting can be seen below. In Figure 6, on the left side, the sound module MAX4466, in the center the Wifi Esp8266 NodeMCU module, on the right a buzzer that acts as a siren to warn the users in the environment when the permitted noise limit is exceeded and finally just below, a USB connection for powering the embedded system, it is emphasized the use of a conventional mobile phone USB charger for power supply.

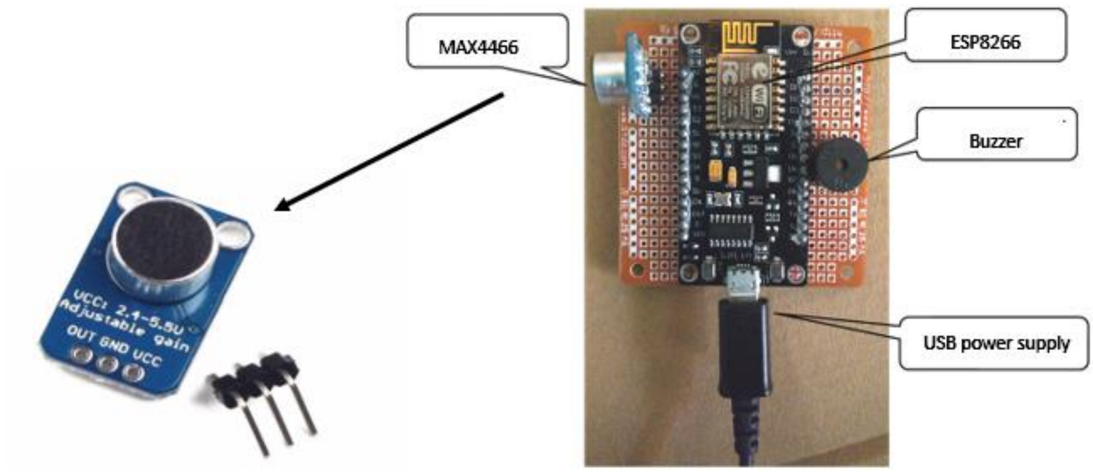

Figure 6. Embedded system architecture.

4. Results

For a presentation of the embedded system results, two versions for sending data to the Cloud platform from the Esp8266 NodeMCU WiFi were developed: using ThingSpeak platform and using the Wireless

Monitor platform.

To communicate the IoT module with the ThingSpeak platform, it is necessary to register and create a specific communication channel for the IoT module. The values received on the platform are presented in a graph that is publicly available on the web, as shown in Figure 7.

There are some limitations when using the ThingSpeak platform. Although it is easy to configure, it displays only the last values obtained, it has a minimum sending limit of fifteen seconds and it is not allowed to modify the platform and access all the values so that own graphics could be develop. All this can be done using the Wireless Monitor platform.

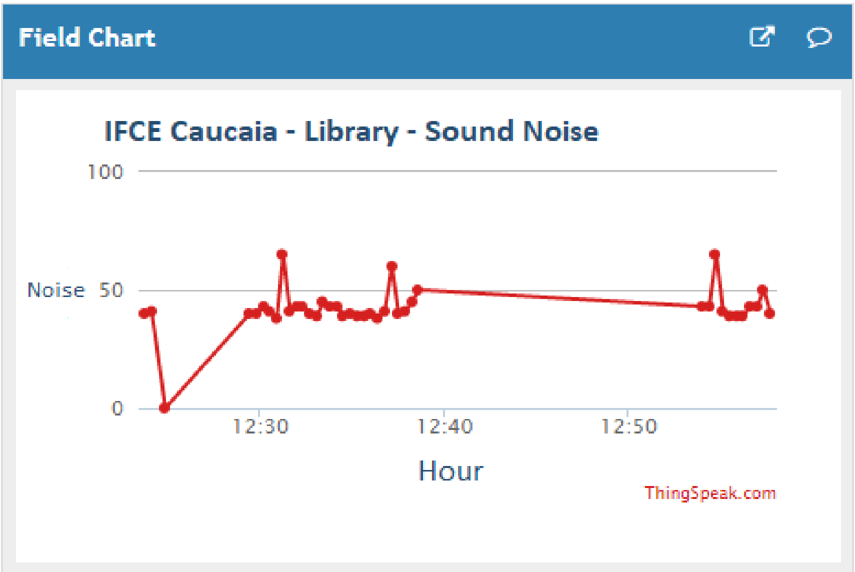

Figure 7. Results displayed on the ThingSpeak platform.

To achieve the communication between the embedded system and the Wireless Monitor platform, a plugin called Sound Monitor was created for the connection to the IoT NodeMCU device, as shown in Figure 8.

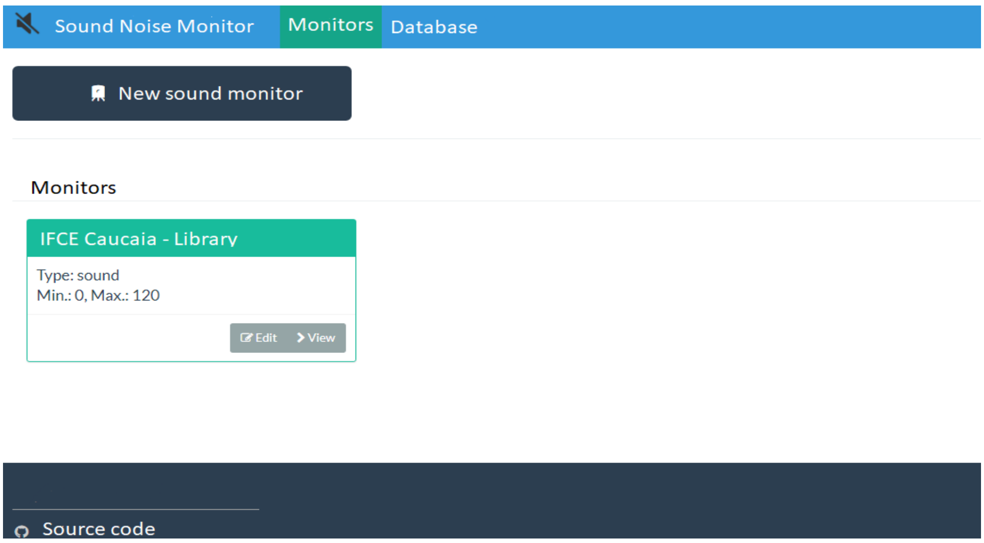

Figure 8. Screen for creating new Sound Monitor.

A developer who chooses to use the Wireless Monitor platform can either use it in the default framework or modify it by creating new screens, new graphical display guides as needed, and can also perform direct queries to the database used by the platform [25]. the total control of all the data collected by the IoT modules and it is worth mentioning that it can be used by three different database management systems, MySQL, PostgreSql or Sqlite.

One of the needs that was observed in the development of this embedded system was to create the possibility of changing the noise limit allowed for the environment in which the embedded system would be applied, considering the different profiles of users of the environments. The Wireless Monitor platform enabled the creation of this new functionality, a new screen was developed with the option to change the noise level allowed for the environment as shown in Figure 9, with the objective of adapting the embedded system to the reality of any environment that will use it, or even so that in any eventual changes of the custom in the environment, such as a building reform, the siren would not play due to the noise level being greater than the normal limit.

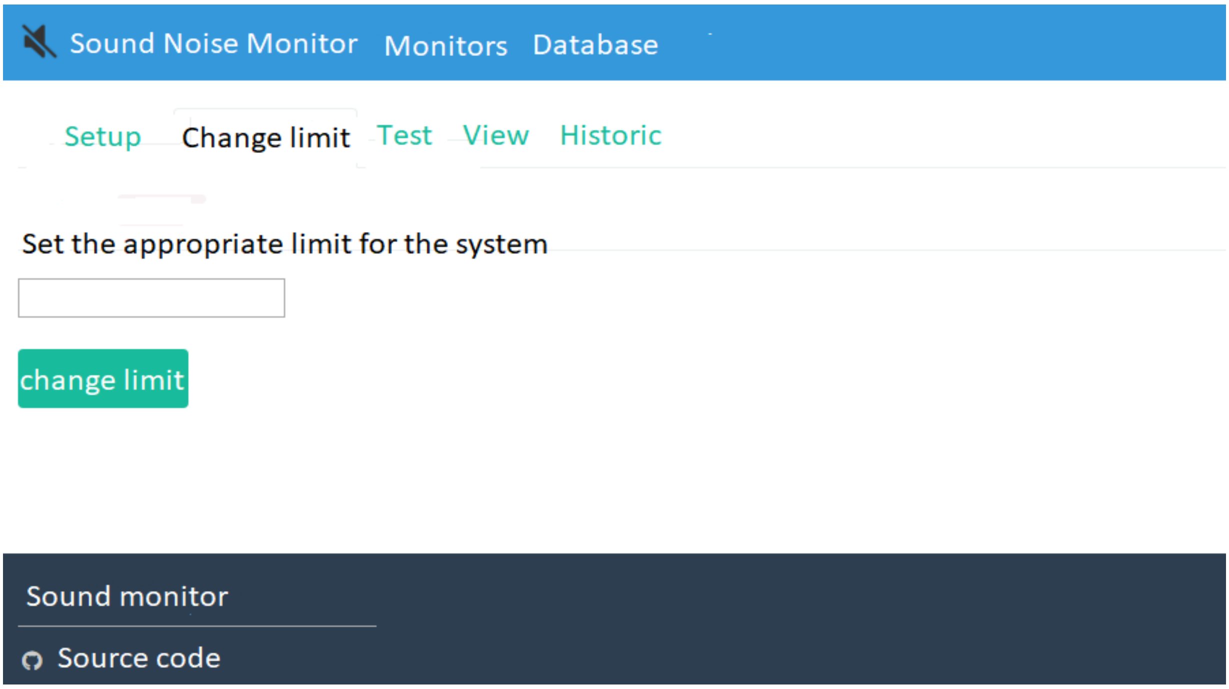

Figure 9: Change screen of the noise limit allowed in the environment.

Like the ThingSpeak platform, it was also possible to display graphs with the latest values obtained from the IoT device in the Wireless Monitor. On the left side of Figure 10, it is possible to see the last value received from the equipment and on the right side, it is possible to see the last 30 values received.

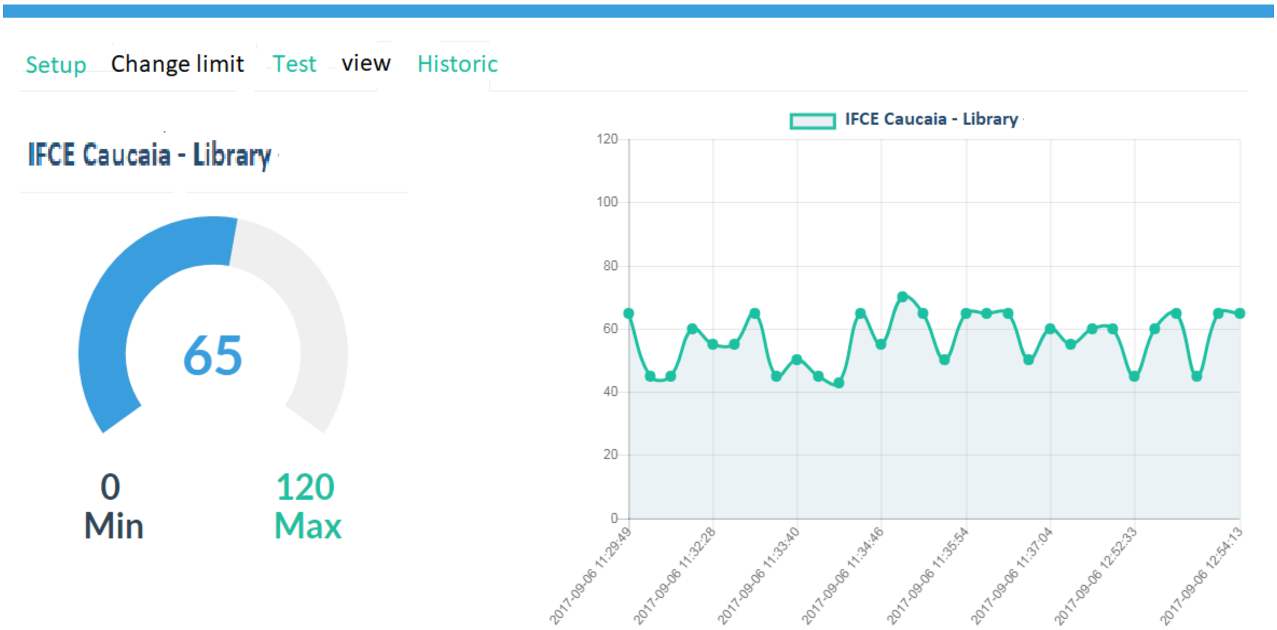

Figure 10. Screen showing values obtained.

5. Conclusion

IoT refers to the way electronic components communicate with others and with the outside world in an environment. With the application of this project in an educational institution library, it fits this concept, making communication with the outside world through the web using the IoT module ESP 8266 NodeMCU with embedded WiFi module described in this article and meeting the objectives, which was to use low cost equipment, free software and technologies.

In addition to the communication, the main objective of this project was achieved, because with the operation of the noise detection embedded system, monitoring in real time the levels of noise and alerting the people present in the area when the allowed limit was exceeded was done and a few times levels of noise beyond the limit established as suitable for the environment were recorded. There was also a re-education of users, who adapted themselves to the new reality established, providing a pleasant environment and conducive to concentration and reading during the studies.

This embedded system can be replicated for use in other libraries, or even in other areas, such as hospitals, clinics, schools, day care centers, that is, in places where noise levels must be controlled.

It is intended in future works, to develop new graphic models that can help the analysis of the data obtained through the embedded system, also to maintain the continuity of the dissemination of the project, to encourage other developers to replicate the project and also to suggest improvements.

6. References

[1] ABNT Brazilian Association of Technical Standards (ABNT). 10152: noise levels for acoustic comfort [Internet]. Rio de Janeiro; 1987 - [cited 2018 July 17]. Available from: http://www.joaopessoa.pb.gov.br/portal/wp-content/uploads/2015/02/NBR_10152-1987-Conforto-Ac_stico.pdf.

[2] L.R. Silveira and K.R.F. Silva. Analysis of operant behavior and conditioning of library users [Internet]. São Luís - MA; 2010 - [cited 2018 July 23]. Available from: https://repositorio.ufma.br/jspui/handle/1/428.
[3] J.M. Juran, The quality since the project: the new steps for the planning of the quality in products and services.Pioneira, São Paulo, 1992. pp.558.
[4] K.R.F. Silva and C.M.A. Veras Júnior. Development of a microcontrolled device for noise control in the library of the Federal Institute of Maranhão. In: Picos PI. Proceedings of the III Regional School of Informatics of Piauí - Articles and Minicourses, 2017 June, pp.12-14, Picos, PI.
[5] J.Y.P. Leite, R.L. Lopes, E.S. Amaral, J.C. Santos and M.R. Fernandes. Noise level: a measure of quality in libraries. In: São Paulo SP, Brazilian Association of Sanitary and Environmental Engineering, editor. Proceedings of the XIX Brazilian Congress of Sanitary and Environmental Engineering, São Paulo, SP.
[6] B.P. Santos, L.A.M. Silva, C.S.F.S. Celes, N.J.B. Borges, B.S.M. Peres, A.M. Vieira, L.F.M. Vieira, O.N. Goussevskaia and A.A.F. Loureiro. Internet of Things: From Theory to Practice [Internet]. Minas Gerais; 2016. Available from: http://homepages.dcc.ufmg.br/~mmvieira/cc/papers/internet-das-coisas.pdf.
[7] A. Zanella, N. Bui, A. Catellani, L. Vangelista and M. Zorzi. “Internet of things for smart cities.” IEEE Internet of things journal. 2014 Feb; vol.1, no.1, pp.6-15, doi : 10.1109/JIOT.2014.2306328.
[8] A. Zanella, N. Bui, A. Catellani, L. Vangelista and M. Zorzi. “Internet of things for smart cities.” IEEE Internet of things journal. 2014 Feb, vol.1no.1, pp.17-19, doi : 10.1109/JIOT.2014.2306328.
[9] Jucá SCS. Practical applications of Raspberry Pi with PIC microcontrollers [Internet]. Maracanaú-CE; 2016 - [cited 2018 July 23]. Available from: https://github.com/SanUSB-grupo/Aplicacoes-Praticas-de-Raspberry-Pi-com-microcontroladores-PIC.
[10] RaspberryPi ORG. Raspberry Pi model B+ [Internet]. 2017 - [cited 2018 July 23]. Available from: https://www.raspberrypi.org/products/raspberry-pi-3-model-b-plus.
[11] Wiring Pi. GPIO Interface library for the rasberry pi [Internet]. 2018 - [cited 2018 July 17]. Available from: http://wiringpi.com.
[12] S.C.S. Jucá, P.C.M. Carvalho and F.T. Brito. SanUSB: Educational software for the teaching of microcontroller technology. Science & Cognition. 2009 Nov, vol.14, no.3, pp.134-144.
[13] R.I.S. Pereira, S.C.S. Jucá and P.C.M. Carvalho. “Online Monitoring System for Electrical Microgeneration via Embedded WiFi Modem”. IEEE Latin America Transactions. 2016 July, vol.14, no.7, pp. 3124-3129, doi, 10.1109/TLA.2016.7587611
[14] R.S. Gonçalves, “Application of LEGO Mindstorms Kits for Teaching Mechatronics Engineering” IJIER, v. 5, no. 10, 2017, pp.99-113.
[15] A.C. Alves and S.C.S. Jucá. Wireless Monitor – Free web application to receive and display data from IoT equipment. III Regional School of Informatics of Piauí. Book Anais - Articles and Minicourses. 2017 jun, vol.1, no.1, pp.117-122.
[16] P. Korop. How to create a Laravel 5 package in 10 easy steps. Laraveldaily [Internet]. 2017 - [cited 2018 July 17]. Available from: https://laraveldaily.com/how-to-create-a-laravel-5-package-in-10-easy-steps/.
[17] MATLAB. ThingSpeak [Internet]. 2017 - [cited 2018 July 17]. Available from: https://www.mathworks.com/hardware-support/thingspeak.html.
[18] K.N. Sathish, B. Vuayalakshmi, P.R. Jenifer and A. Shankar. Iot based smart garbage alert system using arduino uno. IEEE Region 10 Conference (TENCON), 2016, pp. 1028–1034. doi: 10.1109/TENCON.2016.7848162.

[19] X. Cheng, J. Lu and W. Cheng. A survey on RFID applications in vehicle networks. In Beijing CH. Proceedings of the International Conference on Identification, Information, and Knowledge in the Internet of Things; 2016 Oct, 22-23, pp. 146-151.

[20] A. Javed, H. Larijani, A. Ahmadinia, R. Emmanuel, M. Mannion and D. Gibson. "Design and implementation of cloud enabled random neural network based decentralized smart controller with intelligent sensor nodes for HVAC". IEEE Internet of Things Journal. 2016 nov, vol.4 no.2, pp.1-11, doi: 10.1109/JIOT.2016.2627403.

[21] A. Vadnerkar, S. Figueiredo, N. Mayo and R. Kearney. "Design and validation of a biofeedback device to improve heel-to-toe gait in seniors". IEEE Journal of Biomedical and Health Informatics. 2017 Feb, vol.22 no.1, pp.1-7, doi: 10.1109/JBHI.2017.2665519.

[22] M. Sha, D. Gunatilaka, C. Wu and C. Lu. "Empirical study and enhancements of industrial wireless sensor-actuator network protocols". IEEE Internet of Things Journal. 2017 june; vol.4 no.3, pp.1-8, doi: 10.1109/JIOT.2017.2653362.

[23] A. Yavilevich. Arduino sound level meter and spectrum analyser [Internet]. 2017 - [cited 2018 July 17]. Available from: https://blog.yavilevich.com/2016/08/arduino-sound-level-meter-and-spectrum-analyzer.

[24] R. I. S. Pereira, P. C. M. Carvalho, and S. C. S Jucá. "WiFi Data Acquisition System and online monitoring applied to thermoelectric microgeneration modules". International Conference on Renewable Energies and Power Quality (ICREPQ'15). v. 13, 2015 pp.370.

[25] Android Boys Lab. Sound Meter PRO [Internet]. 2017 - [cited 2018 July 23]. Available from: https://play.google.com/store/apps/details?id=kr.aboy.sound.

[26] R. I. S. Pereira, S. C. S. Jucá, and P. C. M. Carvalho. "Online Monitoring System for Electrical Microgeneration via Embedded WiFi Modem" IEEE Latin America Transactions, vol. 14, no. 7, 2016 pp. 3124-3129.

[27] R. I. S. Pereira, I. M. Dupont, P. C. M. Carvalho, P. C. M. Jucá. "IoT embedded linux system based on Raspberry Pi applied to real-time cloud monitoring of a decentralized photovoltaic plant". Measurement Journal. v. 114, 2016 pp.3124-3129.