Modular Automation Design for Equipment Management

Fabiano Santos, Bruno Prado, Daniel Dantas and Kalil Bispo
Department of Computing, Federal University of Sergipe, Marechal Rondon Avenue, São Cristóvão, Brazil

Keywords: Residential Automation, Internet of Things, Wireless Sensor Network, Modular Design.

Abstract: This paper presents a modular and low cost residential automation solution, using the Internet of Things (IoT) concept, for control and monitoring electronic equipment. The development of this work is divided in two parts: the configuration of electronic devices and use of Home Assistant platform. The Home Assistant platform allows the control and monitoring of electronic equipment over the Internet. It uses the MQTT communication protocol for integration with the devices. An auxiliary configuration application has been developed to configure dynamically the pins of the sensors and actuators. An electronic device was developed which is responsible for reading and controlling a set of sensors and actuators. The results of the proposed project were positive. It supports a larger number of sensors and has the lowest cost, of US$1.78.

1 INTRODUCTION

Through personal computers, the Internet, mobile telephony and other technologies that have emerged in the world of personal computing, the acceptance of residential technologies has come to have a strong appeal (Muratori and B, 2017). With the creation of the Internet and in conjunction with residential automation, it was possible to connect the automated devices used daily to the Internet. This set of Internet enabled devices is what we call the Internet of Things (IoT) (Braga, 2017a).

The Internet of Things is considered a technological revolution and more and more new products, of the most varied types such as computers and smartphones, that can be connected to the Internet and other devices, are to be found (Zambarda, 2014). However, its application in people's daily lives, or even access to it, seems to belong more to the future than to the present (Freitas, 2016). There are some factors that are responsible for this, such as: incompatibility of technologies for the operation of a single system, high cost of investment, mobile Internet infra-structure below expected and complexity of use and installation of systems.

There are several home automation solutions, but many are not standardized. When a solution provides some sort of standard, it has a high cost and is not compatible with other market solutions. Another important factor is the complexity of using and maintaining the systems. It is necessary to create standards that facilitate the connection between devices from different manufacturers. Users generally want to control their devices from a single platform and not have one for each device, as it currently occurs (Takahashi, 2016).

This article aims to develop a modular and low-cost residential automation system using the Internet of Things (Kelly et al., 2013) concept for the control of electronic equipment, such as light bulbs and household appliances, through the Internet.

2 RELATED WORKS

The work developed by (Mikhaylov and Huttunen, 2014) (**Work 1**) presents a system of modular electronic devices, using the concept of Wireless Sensor and Actor Network (WSAN), with connectivity between plug & play modules. Because it is scalable, it enables the customization of the modules for the desired application, and together with the plug & play connection, makes it easy to use. The proposal of this approach allows the system to be scalable through an IMP-bus bus, which made the hardware relatively large and with custom components that make design expensive. The purpose of the work of (Mikhaylov and Paatelma, 2015) (**Work 2**) was to develop the software architecture for the modular hardware platform introduced in the (Mikhaylov and Huttunen, 2014) project. With the proposed architecture, it was possible to support the dynamism of the

Santos, F., Prado, B., Dantas, D. and Bispo, K.
Modular Automation Design for Equipment Management.
DOI: 10.5220/0006701506010606
In *Proceedings of the 20th International Conference on Enterprise Information Systems (ICEIS 2018)*, pages 601-606
ISBN: 978-989-758-298-1

hardware platform. For each available peripheral in the module, a driver is required, which increases the computational load for the devices. With the use of a FreeRTOS operating system, the need for a microcontroller with greater processing power and memory has increased, since only the RTOS kernel requires about 5 to 10 KB of space in ROM (FreeRTOS, 2017) memory, cost of the project.

In the work of (Chi et al., 2014) (**Work 3**) a device was designed with a reconfigurable interface of intelligent sensors for industrial Wireless Sensor Networks (WSN) in the IoT environment. The system uses the IEEE 1451 (Song and Lee, 2008) interface standard, which allows the system to collect sensor data intelligently. The interface is plug & play, compatible with various types of sensors. As this system is geared towards the industrial environment, it requires more robust components, making the system relatively large, and the use of the ZigBee standard, makes the project more expensive.

The work of (Suh and Ko, 2008) (**Work 4**) addresses an intelligent home control system, based on WSN. The system has hardware modularity and through several additional modules built, allows customization of the platform for the desired application. Through a single module, we could allow the connection of a set of predefined sensors and there would be no need for additional modules, which would simplify the maintenance of the hardware if any sensors are defective or need to be updated.

The (Kruger et al., 2013) project (**Work 5**) proposes a new platform related to WSN. The platform uses a modular design approach, supporting two operating systems: TinyOS and ContikiOS. In addition, it is composed of two modular units, one of processing and communication, another of storage and energy harvesting. The inclusion of the solar energy harvesting module facilitated the use of the platform in harsh environments where there is no electricity. The sensors were not mentioned and their modules were not designed, which limits the use of the platform.

The project of (Kelly et al., 2013) (**Work 6**) presents a residential automation system for the monitoring of environmental conditions, energy management and control of household devices, such as lamps and water heater, using the IoT concept. The hardware for data acquisition was built in a modular way, where each type of detection unit has a specific purpose. In addition, the system controls the appliances automatically, based on some parameters such as: sensors, use of the inhabitants and the value of the energy tariff. Using the ZigBee standard, it was possible to create a wireless sensor network and make the system scalable, but made the project more expensive. Supported modules and sensors are limited, allowing no further customization of the sensors.

The purpose of (Higuera et al., 2015) work (**Work 7**) is to build an intelligent interoperable lighting solution, combining different technologies. Through human-focused lighting studies, the system can change the intensity of light to increase visual comfort. In addition, this system follows the guidelines defined by ISO/IEC/IEEE 21451 (formerly IEEE 1451) and ZigBee Light Link (Wang, 2013) standards. The use of the IEEE 21451 standard, which required additional components, and the ZigBee, made the project more expensive.

In the work of (Magno et al., 2015) (**Work 8**) a new system was proposed to control LED lighting with a wireless sensor network. The use of light and motion sensors in combination with user preferences allowed distributed intelligence to save energy by reducing the intensity of light. Using the ZigBee standard, the system has become flexible and scalable, but has become more expensive. In addition, the system only supports two types of sensors, which limits its use.

3 SYSTEM DESCRIPTION

The development of this work is divided into two parts: the Home Assistant platform configuration and usage and the proposed electronic device development.

Figure 1 shows the project architecture, which consists of the server (Mosquitto broker + Home Assistant) and the electronic device (Arduino Nano + ESP8266). The data communication between server and electronic device is done over WiFi, using the MQTT protocol. The server can control actuators and monitor sensors.

3.1 Home Assistant Platform

Home Assistant is an open source home automation platform developed in Python 3 (Assistant, 2017b). This platform provides control, monitoring and automation of devices, in addition to being able to run on the most used operating systems, such as Windows, Linux and MacOS, it supports more than 750 components, including MQTT, ZigBee and Z-Wave. Home Assistant has, as frontend, a web application with a single interface for mobile devices where you can control all the equipment. When you start the application for the first time, a default configuration file named **configuration.yaml** is created. This is the main application file, which enables the web interface

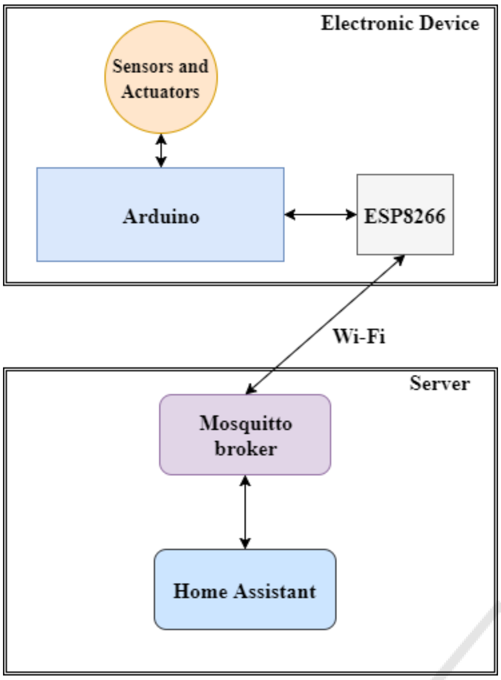

Figure 1: Project architecture.

and device discovery. All components (sensors, actuators, etc.) and their settings, basic settings and communication are contained in this file.

3.1.1 Communication using the MQTT Protocol

For communication between the devices and the Home Assistant, the MQTT protocol version 3.1 with the broker Mosquitto (Eclipse, 2017) was used. MQTT uses the publish/subscriber paradigm, where an element of the network **subscribes** when it wants to receive data and **publishes** when wants to send data(Barros, 2015). Message identification is done through topics, with levels separated by bars. For this project, topics were created for each sensor and actuator, where data sent by the devices is stored. The default name convention for creating these topics is [IP]/assistant.residential/[sensor or actuator]. The IP corresponds to that of the device where the sensor or actuator is connected. This makes it easy to control the components that will be associated with the device. The actuator is controlled by the ON and OFF commands, sent to the [IP]/assistant.residential/control topic. For the sensors, four types of topics were created:

- **[IP]/assistant.residential/luminosity:** stores the brightness value provided by the LDR sensor;
- **[IP]/assistant.residential/temperature:** stores the temperature provided by the DHT11 sensor;
- **[IP]/assistant.residential/humidity:** stores the humidity of the air supplied by the DHT11 sensor;
- **[IP]/assistant.residential/movement:** stores a flag that shows if there is movement, according to the PIR sensor DYP-ME003.

The Home Assistant accesses these information through its configuration file, where all the topics are, each related to its component, as discussed in the 3.1.2.

3.1.2 Device Configuration

Home Assistant has a platform configuration file, **configuration.yaml**, where all components are configured. For each new component that will be used, you need to edit this file with your settings for the platform to recognize it. In order to facilitate user configuration and use, a web application was developed in Python with CGI (Common Gateway Interface) (Foundation, 2017) that will automatically generate this file. In addition, this application will allow the customization of the devices and their components, configuring the pins of the sensors and actuators. Several devices can be added, which will be displayed in a table in the main screen of that application, with the option to edit them.

3.2 Proposed Electronic Device

The developed electronic device is responsible for reading and controlling a set of certain sensors and actuators. For this device were chosen components that met the objectives of the project: low cost and easy availability, such as the Arduino Nano and the module ESP8266. These components, plus the set of sensors and actuators supported, form the electronic device of this project.

3.2.1 Hardware Development

The Arduino Nano is a free hardware electronic prototyping platform that uses an ATmega328 microcontroller (Braga, 2017b). It was chosen for having a cost of US$1.78 (AliExpress, 2017f) and providing support for a wide range of sensors, being responsible for reading these sensors and driving the devices through the actuators. The ESP8266 module was used because it provides the Arduino Nano wireless communication (Thomsen, 2015). This module communicates with the Arduino via serial interface. With it, the Arduino can connect in 802.11 b/g/n wireless networks, operating in AP (Access Point) or STA (Station) modes. There are several models, but the one chosen was the ESP-01 because it is compact, 25 mm high and 15 mm wide (Thomsen, 2017), and has a cost of US$2.21 (AliExpress, 2017d). Three types of sensors and one actuator were used, all of which are easily available and supported by Arduino Nano. The three types of sensors used are: brightness sensor LDR (FilipeFlop, 2017c), temperature and humidity sensor DHT11 (e Cia, 2013) and the PIR motion sensor (e Cia, 2014). The relay module of a channel was used as actuator (Thomsen, 2013).

3.2.2 Firmware Development

The firmware of the electronic device, that is, the Arduino Nano and the ESP8266, was developed. The Arduino Nano has the function of reading the sensor data and actuating the actuators, as well as communicating with the ESP8266 module. This communication was made through the serial interface. In the diagram of Figure 2 you can observe the execution flow of your firmware. The Arduino sends to the ESP8266 sensor data and actuator status, previously configured when adding the device to the application. In this communication, we observed that the exchange of several messages in a short time caused losses of characters, thus, we created a message pattern. This pattern consists of several concatenated messages, that is, the values of each sensor and actuator are concatenated and sent at one time. A function has been implemented that associates the sensor/actuator to the pin configured in the developed application and stores the pin value in the EEPROM memory of the Arduino Nano. This pin is stored in the memory address of the sensor type. The ESP8266 receives from the configuration application the pins of the sensors and the actuator used in the device and sends it to the Arduino Nano. After the pins are configured, Arduino starts reading the connected components and sends its data to the ESP8266.

The ESP8266 module is responsible for providing wireless communication to the Arduino Nano. In the diagram of Figure 3 is shown the execution flow of its firmware. It connects to the Wi-Fi network and also to the Mosquitto broker, establishing communication between the device and the Home Assistant. The WiFiManager library, which implements all the configuration features of the (Tzapu, 2017) network, was used. These settings are made through a web application, included in the library itself. In addition to the Wi-Fi network configuration, the library provides configuration of the connection to the Mosquitto broker.

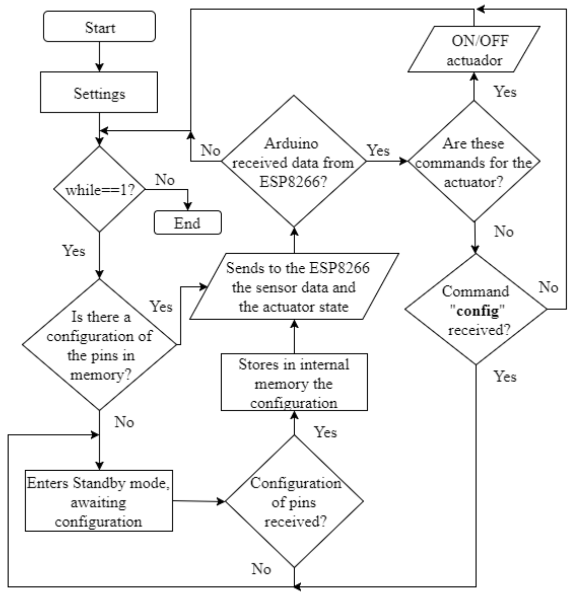

Figure 2: Arduino Nano firmware execution flow.

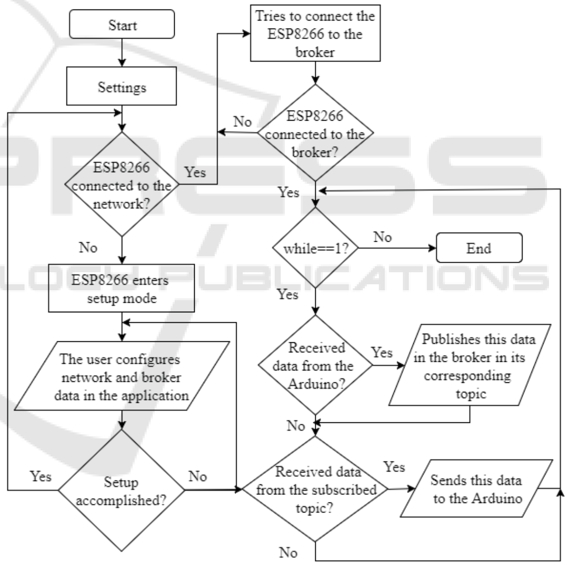

Figure 3: ESP8266 firmware execution flow.

4 RESULTS AND DISCUSSIONS

All the analyzed works are scalable, allowing the expansion to control more equipment. In the Table 1, one can observe a comparative analysis of all the works and their quantitative parameters. For the platform was estimated the cost of the component used as the main processor of the device. Cost estimates were made through prices researched at virtual stores, giving preference to the website AliExpress (AliExpress, 2017c). Some components were not found on this website and had their prices searched on the page of Mouser (Electronics, 2017) and Amazon (Amazon, 2017). The types of sensors supported refer to the number of sensor types that can be used in the device, that is, how many sensors with different purposes are supported, allowing a wider range of applications for the device.

Table 1: Comparative analysis of related works.

Platform	Cost	# Supported Sensors
Work 1/2	US$3.85	4
Work 3	US$10.26	7
Work 4	-	7
Work 5	US$6.21	-
Work 6	US$49.55	4
Work 7	US$2.50	3
Work 8	US$2.50	2
Proposed	**US$1.78**	**8**

The project developed in this work has as one of its objectives to be low cost. The proposed platform has a cost of US$1.78 (AliExpress, 2017f) and in addition to the ESP8266 Wi-Fi module, at a cost of US$2.21 (AliExpress, 2017d), the total cost of the device is US$3.99, without considering the sensors and actuators that could be used. The sensors used in the experiments were: luminosity (LDR), which costs US$0.40 (AliExpress, 2017g); temperature and humidity (DHT11) at a cost of US$0.78 (AliExpress, 2017b); and movement (PIR), which has a cost of US$0.78 (AliExpress, 2017e). For the actuator, a one-channel relay module was used that costs US$0.53 (AliExpress, 2017a). In addition to these three sensors used, the device can provide support for five different sensors, which are: infrared (E18-D80NK) (UsinaInfo, 2017), sound intensity (KY-038) (FilipeFlop, 2017d), rainfall detection (YL-83) (FilipeFlop, 2017a), soil moisture (YL-69) (Makers, 2017) and gas detection (MQ-2) (FilipeFlop, 2017b). Considering sensors/actuators that use 1 data pin, such as those supported by the device, up to 20 components could be used in a single device (Arduino, 2017).

5 CONCLUSION

Most of the existing residential automation solutions are complex and they have high cost (Freitas, 2016). The purpose of this work is the development of a low-cost, modular residential automation solution that can be easily adopted.

The use of Home Assistant application provides the control and monitoring of the equipment through the Internet. This application, together with the use of WiFi communication, has increased the scalability of the project. For communication between devices with the Home Assistant, the MQTT protocol was used, which reduced the hardware requirements and facilitated its implementation due to its low complexity. When compared to the related works, the proposed project has the lowest cost platform and supports a greater variety of sensors. Its installation, configuration and maintenance is straightforward, which facilitates its use. In addition, this platform allows the expansion to control more equipment.

Future work includes the implementation of new sensors and actuators for the device, since only three types of sensors have been tested. Automation can also be made in the Home Assistant (Assistant, 2017a) application, creating automatic rules for controlling electronic equipment, such as turning on lights when it is dark and detecting presence. Another future work would be to use the device data for big data applications. This data can be stored and used by some software with artificial intelligence, such as pattern recognition, making the devices predict actions that users would perform.

REFERENCES

AliExpress (2017a). 1 pcs 5v low level trigger one 1 channel relay module interface board shield for pic avr dsp arm mcu arduino. Available at: https://goo.gl/PTzNiQ. [Accessed 21 Sep. 2017].

AliExpress (2017b). 1x dht11 dht-11 digital temperature and humidity temperature sensor for arduino diy kit. Available at: https://goo.gl/zcN4i9. [Accessed 21 Sep. 2017].

AliExpress (2017c). Aliexpress. Available at: https://pt.aliexpress.com. [Accessed 13 Oct. 2017].

AliExpress (2017d). Diy esp8266 esp-01s wireless wifi sensor for arduino esp-01 advanced version wireless transceiver. Available at: https://goo.gl/TJQKxS. [Accessed 21 Sep. 2017].

AliExpress (2017e). Hc-sr501 adjust infrared ir pyroelectric infrared pir module motion sensor detector module we are the manufacturer. Available at: https://goo.gl/N7u7aZ. [Accessed 21 Sep. 2017].

AliExpress (2017f). Nano 3.0 controller compatible for arduino nano ch340 usb driver no cable. Available at: https://goo.gl/pnVfcn. [Accessed 13 Oct. 2017].

AliExpress (2017g). Photosensitive sensor module light detection module for arduino. Available at: https://goo.gl/gDTjB9. [Accessed 13 Oct. 2017].

Amazon (2017). Amazon. Available at: https://www.amazon.com. [Accessed 13 Oct. 2017].

Arduino (2017). Arduino nano. Available at: https://goo.gl/TQB9En. [Accessed 07 Oct. 2017].

Assistant, H. (2017a). Automating home assistant. Available at: https://goo.gl/2RY6FW. [Accessed 17 Oct. 2017].

Assistant, H. (2017b). Awaken your home. Available at: https://home-assistant.io/. [Accessed 27 Apr. 2017].

Barros, M. (2015). Mqtt - protocolos para iot. Available at: https://goo.gl/zVdBSD. [Accessed 24 Apr. 2017].

Braga, I. N. C. (2017a). A internet das coisas (iot) (art2054). Available at: https://goo.gl/YhoqLA. [Accessed 14 Apr. 2017].

Braga, I. N. C. (2017b). A onda do arduno (col001). Available at: https://goo.gl/FqfQLS. [Accessed 19 Sep. 2017].

Chi, Q., Yan, H., Zhang, C., Pang, Z., and Xu, L. D. (2014). A reconfigurable smart sensor interface for industrial wsn in iot environment. *IEEE Transactions on Industrial Informatics*, 10(2):1417–1425.

e Cia, A. (2013). Sensor de umidade e temperatura dht11. Available at: https://goo.gl/XZto8d. [Accessed 21 Sep. 2017].

e Cia, A. (2014). Sensor de presena com mdulo pir dyp-me003. Available at: https://goo.gl/ut8hTc. [Accessed 07 Oct. 2017].

Eclipse (2017). Mosquitto. Available at: https://mosquitto.org/. [Accessed 23 Sep. 2017].

Electronics, M. (2017). Mouser electronics. Available at: http://br.mouser.com. [Accessed 13 Oct. 2017].

FilipeFlop (2017a). Sensor de chuva. Available at: https://goo.gl/cGJgL7. [Accessed 16 Oct. 2017].

FilipeFlop (2017b). Sensor de gas mq-2 inflamvel e fumaa. Available at: https://goo.gl/H2XycT. [Accessed 16 Oct. 2017].

FilipeFlop (2017c). Sensor de luminosidade ldr (light dependent resistor) 5mm. Available at: https://goo.gl/q2uoa7. [Accessed 21 Sep. 2017].

FilipeFlop (2017d). Sensor de som ky-038 microfone. Available at: https://goo.gl/g37Hyf. [Accessed 16 Oct. 2017].

Foundation, P. S. (2017). Common gateway interface support. Available at: https://docs.python.org/2/library/cgi.html. [Accessed 24 Sep. 2017].

FreeRTOS (2017). Freertos faq - memory usage, boot times & context switch times. Available at: https://goo.gl/43E8LA. [Accessed 25 Feb. 2017].

Freitas, L. (2016). O que ainda falta para a internet das coisas decolar. Available at: https://goo.gl/HK01y1. [Accessed 25 Apr. 2017].

Higuera, J., Hertog, W., Perlvarez, M., Polo, J., and Carreras, J. (2015). Smart lighting system iso/iec/ieee 21451 compatible. *IEEE Sensors Journal*, 15(5):2595–2602.

Kelly, S. D. T., Suryadevara, N. K., and Mukhopadhyay, S. C. (2013). Towards the implementation of iot for environmental condition monitoring in homes. *IEEE Sensors Journal*, 13(10):3846–3853.

Kruger, C. P., Abu-Mahfouz, A. M., and Isaac, S. J. (2013). Modulo: A modular sensor network node optimised for research and product development. In *2013 IST-Africa Conference Exhibition*, pages 1–9.

Magno, M., Polonelli, T., Benini, L., and Popovici, E. (2015). A low cost, highly scalable wireless sensor network solution to achieve smart led light control for green buildings. *IEEE Sensors Journal*, 15(5):2963–2973.

Makers, M. (2017). Sensor de umidade de solo yl-69. Available at: https://goo.gl/wJKFn1. [Accessed 16 Oct. 2017].

Mikhaylov, K. and Huttunen, M. (2014). Modular wireless sensor and actuator network nodes with plug-and-play module connection. In *IEEE SENSORS 2014 Proceedings*, pages 470–473.

Mikhaylov, K. and Paatelma, A. (2015). Enabling modular plug amp;play wireless sensor and actuator network nodes: Software architecture. In *2015 IEEE SENSORS*, pages 1–4.

Muratori, J. R. and B, P. H. D. (2017). Captulo i. Available at: https://goo.gl/5h3JHZ. [Accessed 25 Apr. 2017].

Song, E. Y. and Lee, K. (2008). Understanding ieee 1451-networked smart transducer interface standard - what is a smart transducer? *IEEE Instrumentation Measurement Magazine*, 11(2):11–17.

Suh, C. and Ko, Y. B. (2008). Design and implementation of intelligent home control systems based on active sensor networks. *IEEE Transactions on Consumer Electronics*, 54(3):1177–1184.

Takahashi, P. (2016). Internet das coisas: o que falta para decolar? Available at: https://goo.gl/4KTQaH. [Accessed 25 Apr. 2017].

Thomsen, A. (2013). Controlando lmpadas com mdulo rel arduino. Available at: https://goo.gl/brNb6p. [Accessed 07 Oct. 2017].

Thomsen, A. (2015). Tutorial mdulo wireless esp8266 com arduino. Available at: https://goo.gl/o0Todn. [Accessed 21 Apr. 2017].

Thomsen, A. (2017). Qual mdulo esp8266 comprar? Available at: https://goo.gl/a6P8ZY. [Accessed 07 Oct. 2017].

Tzapu (2017). Wifimanager. Available at: https://github.com/tzapu/WiFiManager. [Accessed 09 Oct. 2017].

UsinaInfo (2017). Sensor de proximidade infravermelho - deteco 3 a 80cm - e18-d80nk. Available at: https://goo.gl/bKHwsn. [Accessed 16 Oct. 2017].

Wang, J. (2013). Zigbee light link and its applicationss. *IEEE Wireless Communications*, 20(4):6–7.

Zambarda, P. (2014). Captulo i. Available at: https://goo.gl/4S3M6G. [Accessed 25 Apr. 2017].