International Journal of Applied Engineering Research ISSN 0973-4562 Volume 13, Number 11 (2018) pp.
© Research India Publications. http://www.ripublication.com

Technological Proposal for Controlling a Residential Lighting System

Lenin Coca[1], Cesar Hernández[1*]
Universidad Distrital Francisco José de Caldas, Technological Faculty, Bogotá, Colombia.
Email: lfcocas@correo.udistrital.edu.co, cahernandezs@udistrital.edu.co
[*]Corresponding author, [1]Orcid: 0000-0001-9409-8341*

Summary

The technological breakthrough in process automation and electronic components has allowed households to have a higher level of comfort and security. This article describes a domotics application for the control and supervision of a lighting system of a 54 m2 apartment's living room, bedroom and kitchen. This was achieved through the integration of compatible sensors into the Arduino IDE: the motion sensor PIR HC-SR501 and the light intensity sensor BH-1750 which are controlled through the NodeMcu board. The latter communicates via Wi-Fi to a mobile phone through an app developed in the APP inventor platform. It allows the user to execute ON/OFF control, intensity control (dimer), motion control of people, lighting-based control or automatic mode and programmed control.

Keywords: APP, Arduino, Domotics, IoT, NodeMcu.

INTRODUCTION

The advances in technology have made real time communication to be considered as a valuable tool for an endless number of applications. The Internet of Things (IoT) has emerged as one the most powerful communication solutions of the 21st century. An example of this is the automation of a residential household (Domotics) (Xie, 2016). The internet of things is a paradigm that connects the devices with the real world, turning residential units into Smart-Buildings (Pozza, 2015; Hernández, et al., 2015).

In addition, Arduino is a platform based on open source hardware and software which is easy-to-use and is low cost. Up to this day, a large community has been created where many developers have launched a variety of sensors and boards on the market with different communication and programming modes. Arduino's IDE offers the opportunity to articulate them in a same platform whose programming language is C (Arduino, s.f.). Hence, it is only necessary to download and update the corresponding libraries to the modules as well as updating the Arduino's IDE. There are currently eight types of boards which are capable of reading a variety of sensors as well as sending and receiving data and signals, both analog and digital. All the Arduino boards are open source, which allows users to build them independently and adapt them to a specific need.

This research has the purpose to increase the comfort and motivate the rational use of energy by controlling the lighting system of an apartment (living room, bedroom and kitchen) located in the city of Bogotá. The lighting system is controlled in a versatile manner through an Android app. With the app, the user can use the ON/OFF control, the control of four levels of luminous intensity (natural or artificial), motion-based ON/OFF control, ON/OFF control based on the level of natural or artificial lighting present in the room and finally, the user can program the lighting's activation through an alarm. This is all directed at increasing the added value to the apartment. (Gómez, Peña, Hernández, 2010; Romero, Giral, Hernández, 2010).

Wouldn't it be comfortable to turn off the bedroom lights without the need of getting up to do so? Each point of the lighting system will include a module with dimensions 7 cm x 4 cm x 3 cm which contains a NodeMcu board, a PIR HC-SR501 motion sensor, a BH-1750 lighting sensor, an isolation module for the actuators and supply module. Hence, its implementation will not require additional wiring or modifications on the existing lighting system, pointing out that it must be incandescent or dimmable LED in case the user wishes to vary its intensity. Finally, the cost of the materials for each lighting point is 92.500 COP.

RELATED WORK

In (Montesdeoca Contreras J.C, 2015), an IoT application related to the control of cameras, lighting and opening doors is presented. This is achieved through voice and button-based commands in an application developed in App Inventor. This is all communicated via a Wi-Fi network, but it still has SMS back-up. The main controller is an Arduino Mega 2560 and a Shield 5100 Ethernet card provides the corresponding Wi-Fi communication. The 900 SIM card sends SMS or sends calls to a mobile phone. The communication scheme, the operation of the GSM module, the operation modes of the voice recognition system and the app are also shown. This system focuses on home security since it involves security cameras, magnetic sensors and back-up systems.

In (Quintero, 2005), domotics are defined as "the integration of several areas of knowledge such as telecommunications, electronics and electricity to improve the quality of life of human beings. The two types of architecture are also shown which are crucial for the location of devices inside the house: distributed and centralized architecture. Additionally, the main communication protocols for households are discussed. The article focuses on theoretical and research aspects.

In (Navjot Kaur Walia, 2016), a project that controls lighting (remotely or locally) is described. The ThingSpeak is also shown which sends the signals from two sensors (humidity and temperature). The ESP 8266Wi-Fi module is fed through a 5V-3.3V converter. The system also articulates a tablet or smartphone for lighting control and uses the ThingSpeak server to control lighting and display of the signals for temperature and humidity. The project focuses on the display of sensors and

ON/OFF lighting control, which all depend on the ThingSpeak platform.

In (Navjot Kaur Walia, 2016), the control of security cameras and lighting through a Wi-Fi network and an app is described. The app was created in App Inventor and its inputs can be either voice commands or issued directly by buttons. The project involves the display through the ThingSpeak server. Voice recognition is also implemented: a PWM signal is incorporated for controlling them and a LUA-based programming example of the ESP 8266 module's IDE Explorer is shown.

In (Lukas Muller, 2015), the use of Arduino is explored for power electronics applications with the use of some libraries, PWM and ACD signals. The Arduino is adapted to illustrate the use of digital controllers since it gives the basic parameters for the phase control based on PWM signals. It also has the capacity to operate in frequencies of up to 100 kHz.

METHODOLOGY

In Figure 1, the domotics system of the apartment is described. Each NodeMcu is located next to a lighting point that will be controlled. Initially, the user chooses the space to control when entering the application (Smart Home). Afterwards, the control operations are enabled allowing the mobile device to send the custom parameters via Wi-Fi to the corresponding NodeMcu.

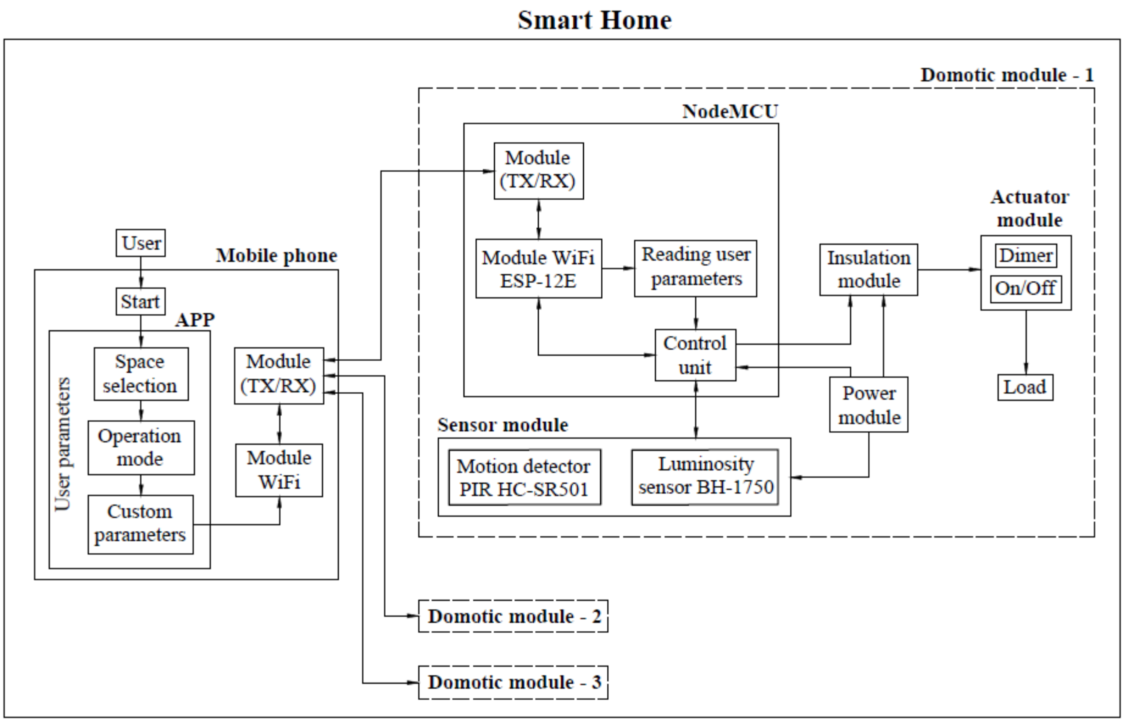


Figure 1. Domotics system

Each NodeMcu is set to be automatically connected to the local Wi-Fi, with a specific IP and communication port (Pedraza, Hernández, López, 2013). Once the custom parameters have been received and processed, a logical state or PWM signal will be defined to one of its outputs. Hence, and through the isolation-actuator module, the target lighting is controlled safely and efficiently.

The sensor module supervises the people's movement and the intensity of natural or artificial lighting. This module contributes to the efficient use of energy, harnesses natural sunlight to the limit and allows automatic control of lighting through the people's motion. Furthermore, it enables the user to visualize the state of the lighting points in real time.

APP

With a size of 17.39 MB, the app does not store data inside the phone and was developed in the App Inventor platform given its versatility, simplicity and free use (Hernández, Márquez, Martínez, 2015). The programming logic is Scratch. Figure 2 shows the initial interface where the user chooses the lighting point according to the place. Then, the control options are enabled such as: ON/OFF control and intensity-based control. While the first one is executed through a single button and changes the state of the energized light source, the second one is achieved through a pre-disposed slider in the lower part of the screen. This slider can change the intensity of the lighting point (25, 50, 75 and 100%). The motion control is managed by choosing the image shown in the upper left section. In this control mode, the turnoff time of the lighting point is directly adjusted in the PIR HC-SR501 sensor. The lighting control is selected through a button (automatic). In this mode, the lighting point is energized if the light falls under 10 lux. Finally, a programmed activation control was set through the clock image shown. It enables a wake-up alarm in 24-hour format allowing the user to choose the exact time for the lighting to be automatically energized.

Figure 2. Smart Home application

In figure 3, the three main functions are shown. Initially, a variable with fixed IP is created for each IoT device. For the case of this image, the allocated IP ends with 128 and is allocated to the living room's NodeMcu. Then, a procedure is created which contains the request format of type Get to the previously selected IP without any specific command. Finally, when the user chooses the ON button and the lighting has been previously chosen, the execution of the procedure is initiated by adding the ON variable. It will be processed by the NodeMcu board and offer the ON or OFF control of the lighting point. For the other control cases, it is the same case except for the intensity control where a sequence of characters is sent.

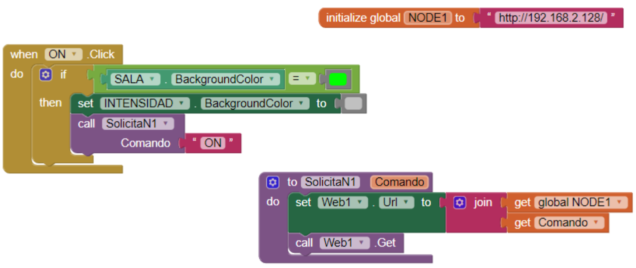


Figure 3. Connection to the web

It is important to point out that the application is visually updated each time that the user chooses some type of control which includes the current lighting control and the type of control.

NodeMcu V 1.0

It is a board still in development which integrates the main characteristics of the Arduino UNO and a an ESP-12E Wi-Fi module. The NodeMcu board has similarities with the Arduino Nano in terms of the pin disposition, port management, analog and digital signals, PWM, I2C, among others. It integrates the ESP-12E Wi-Fi module into a 4 MB Flash drive and was initially developed by AdThinker. To program with the Arduino IDE, it is necessary to use the ESP8266WiFih library which contains the setting of the connection modes known as client, server and access point. Furthermore, the board can work with the *SoftwareSerial.h* libraries, which can control and supervise the state of the inputs and outputs. The disposition of the pins for this project is shown in Table 1. (Santander, Prada, Hernández, 2017; Márquez, Salgado, Hernández, 2017; Hernández, Salcedo, Diaz, 2009).

Table 1. Disposition of the pins

Pin	Assignment
D2	PIR control output (5V)
D3	Crossing by zero entry
D4	BH-1750 SCL entry
D5	BH-1750 SDA entry
D6	PWM control output
D7	On/Off control output

To integrate the components of the Arduino IDE, the firmware from each board was updated and they were programmed through the libraries created by the developers of the Arduino's Open Source community and posted in the *www.github.com* repository. Figure 4 shows the programming logic used.

PIR HC-SR501 sensor

Figure 5 shows the PIR (Passive Infrared Red Movement Sensor) sensor. The thermal energy radiated by the body is caught by the concave lens also called the Fresnel lens which focuses the radiation of an opening angle of 110° into a specific point directed towards the optical filter. Then, the infrared frequencies are filtered which directly impact the piro-electric sensor that transforms radiation into a small voltage signal that then enters the integrated circuit BISS0001, is amplified and compared leading to an output signal with 3.3 V and 5 mA.

The sensor can be adjusted in terms of: the alarm time in high state from 3 seconds to 5 minutes, the detection distance from 3 to 7 meters and, finally, a jumper allows setting the alarm output into a mono-shot (it remains active during the scheduled time) or repeated shot (every time it detects, the counter is reset).

BH1750 sensor

Figure 6 displays the BH-1750 light sensor which delivers lux values and handles a range from 1 to 65535 lux. Its supply voltage is 3.3V to 4.5V with an adjustable resolution between 0.5 to 4 lux. Additionally, these parameters can be defined to perform constant measurements or a single measurement.

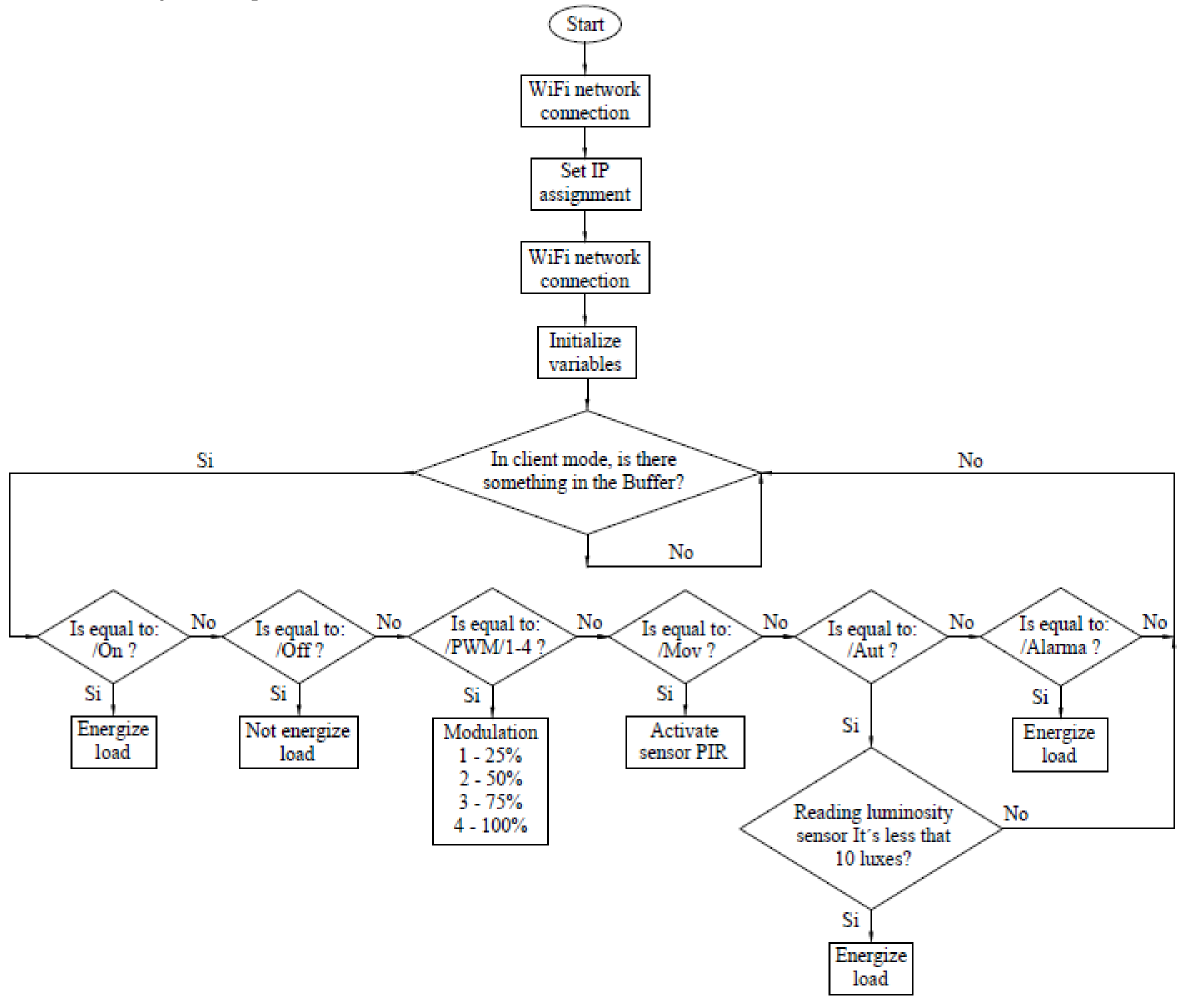


Figure 4. Programming logic

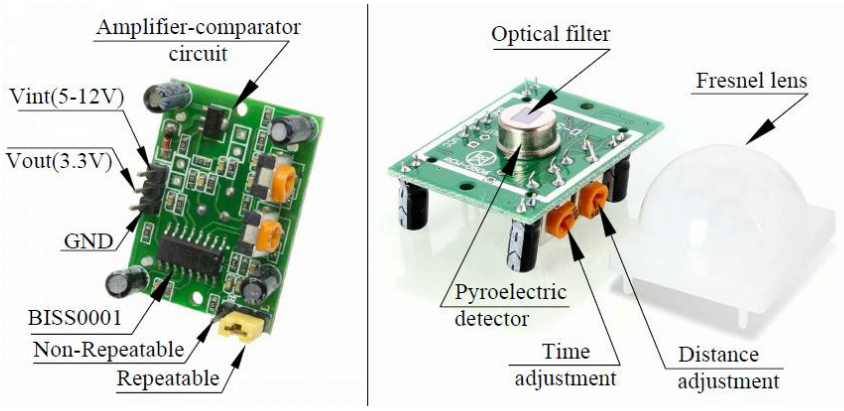


Figure 5. PIR HC-SR501 sensor

The light diode receives the luminous radiation which produces a small current, whose magnitude is proportional to the intensity of the luminous radiation. Afterwards, the operational integrators convert the current into voltage and then the analog signal is converted into a 16-bit digital signal with the ADC. To transmit the information, the sensor integrates the I2C serial bus system whose voltage level is 1.6 V and a double thread protocol: SCL and SDA. The SDA transmits the data and the SCL indicates when to read data. The registered data allow to calculate the light in the environment.

International Journal of Applied Engineering Research ISSN 0973-4562 Volume 13, Number 11 (2018) pp.
© Research India Publications. http://www.ripublication.com

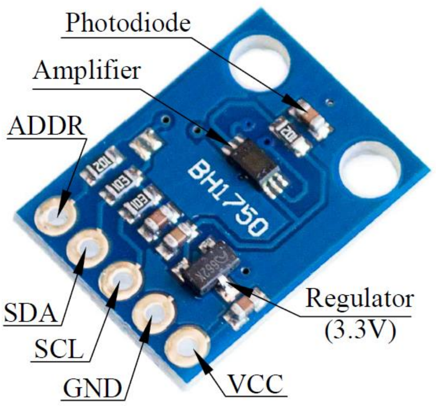


Figure 6. BH-1750 sensor

Isolation-actuator module

The module in Figure 7 is in charge of isolating the control applied by the NodeMcu board and deploy the different types of control. This is achieved through an opto-couplers array, a BT137 Triac that along with a snubber which efficiently controls lighting as shown in Figure 7. Two opto-couplers are destined for ON/OFF control and intensity control, i.e., a PWM signal. In addition, the H11AA1 opto-transistor delivers the zero crossing of the sinusoidal input signal at 120 VAC to perform the ON/OFF or PWM control. The opto-couple is also in charge of energizing the PIR HC-SR501 sensor which requires a second opto-coupler to directly control the Triac.

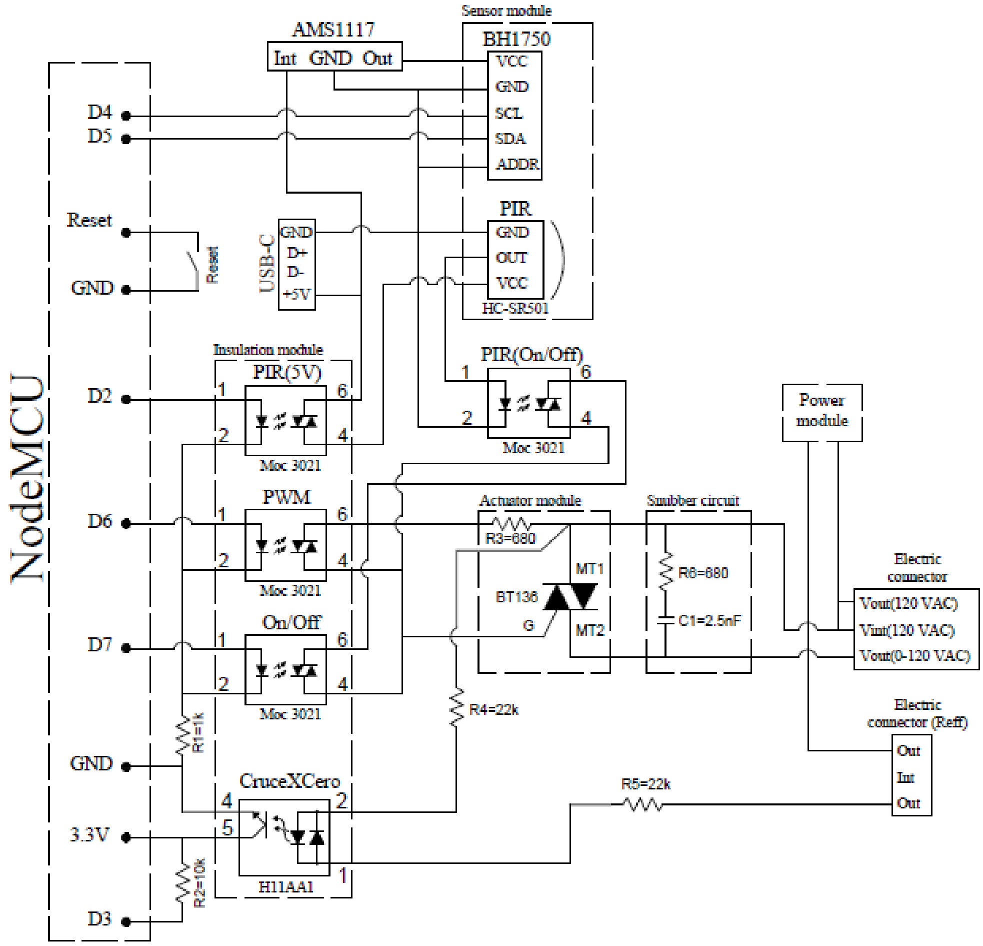


Figure 7. Isolation-actuator module

RESULTS

The programming logic identified that, if the NodeMcu outputs in the D1, D2 nomenclature are needed, they must be previously defined through the command #define. Otherwise, the GP105, GP104 nomenclature should be used.

Given that the BH-1750 sensor uses I2C communication, the *wire.h* library must be used which carries the parameters of the communication protocol. Additionally, any pin of the NodeMcu must be initialized as ports (SCL and SDA) through the *wire.begin()* command.

The *ESP8266WiFi.h* library allows the configuration of the ESP-12E module contained within the NodeMcu board as a client or server and can assign a fixed IP to establish communication with the App.

For the ON/OFF and PWM (dimmer) control, a zero-crossing signal of the input sinusoidal signal is required which is present at all moment in the D3 digital pin assigned to the variable (crossing point).

The inputs and outputs of the NodeMcu can be used analogically or digitally. It is not possible to use a pin as analog and digital at the same time. On the other hand, according to the working capacity of the NodeMcu board which is 10-bit long. The analog output for the dimmer (PWM) control is defined to work in the 1024 decimal range. However, it is possible to modify such range with the *analogWriteRange (1023)* function. In the same mode the PWM frequency can be defined through the *analogWriteFreq(60)* function. For the dimmer control, the *analogWrite(pin range)* function where the range is indirectly the percentage of duty cycle that will be applied to such pin. For the discussed project, the selection was divided into four intervals or values of duty cycle: 255, 510, 760 and 1024, which correspond to 25, 50, 75 and 100%.

The return function within the loop void allows that not all programming is executed. Originally, this function was applied with the purpose of making the board execute some action only when the client sends the information. Therefore, the programming strategy forced the board to always be on the lookout for any client. However, when automatic lighting control was included, the programming had to be modified for the case when the user chose the automatic mode so that the board would receive the information from the lighting sensor and then process it. Afterwards, it would check if any client would try to connect and if it did happen then the control could be changed from ON/OFF to PWM to energizing the PIR sensor.

Through the adjustment of the PIR sensor and according to the location of the prototype, the turnoff time of the lighting can be adjusted. When the user chooses the motion control, the board is not only in charge of energizing the sensor. The lighting's turnoff time is specific to the PIR sensor and the criteria were the following:

- ✓ The sensor has the capacity to adjust the turnoff time (3 seconds to 5 min), which depends on the device's location.
- ✓ The NodeMcu board has conflicts when the void loop logic involves times over 2 s.
- ✓ The current required by the PIR sensor is 65 mA.
- ✓ It is unpractical to have the motion sensor activated if the user does not need it.

Energy consumption

Given that the current required by the module is 120 uA, the power consumption is relatively low (close to 0.39 mW) so the lighting module is constantly energized. This allows the visualization of the current level of illumination each time that the user configures a parameter in the app.

To determine the energy consumption of the prototype, the Voltimeter - Amperimeter was connected to the supply input of the module for 1 hour. During this timeslot, the different types of control were made giving more importance to the ON/OFF and PWM control. The energy consumption is monitored every 2 minutes. The gathered data are shown in Figure 8.

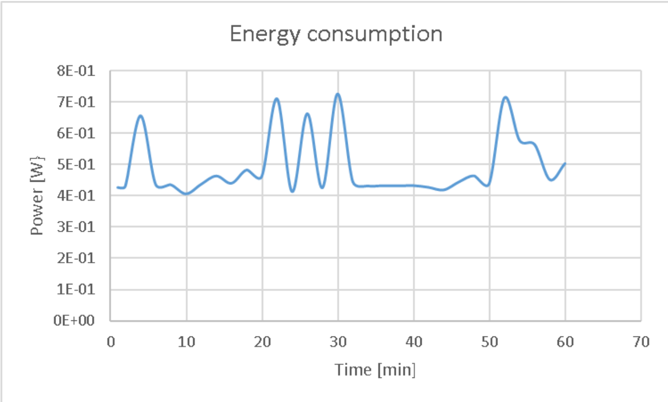
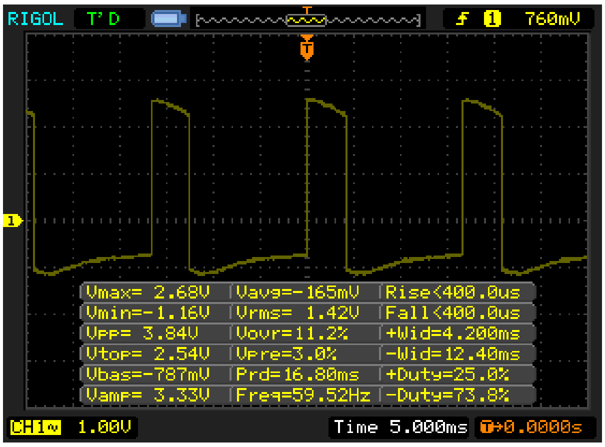


Figure 8. Energy consumption

In Figure 8, the current required by the device depends on how fast the user changes the control options, i.e., as more requests are made, the current required by the prototype tends to grow. Finally, it was determined that the average energy consumption was 0.49 W/h.

PWM signal

Figure 9 shows two measurements by the Rigoll oscilloscope of the PWM signal produced by the NodeMcu board according to the dimmer control set at 25% and 75%.



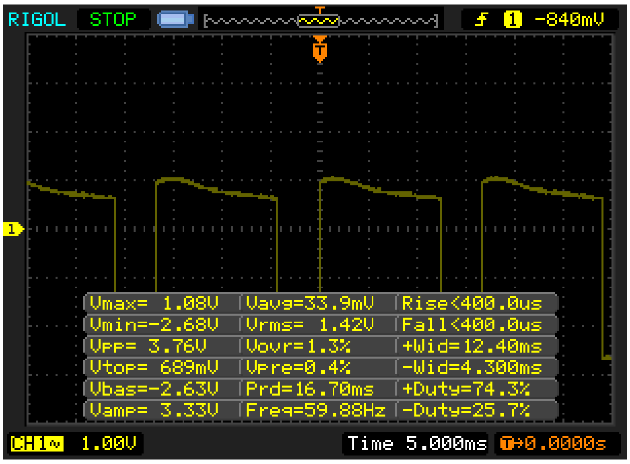


Figure 9. PWM signal of the NodeMcu board

Developed prototype

In figure 10, the developed prototype is shown. Given that the programming does not include waiting times, except for the zero-crossing, the prototype has worked quite well with the Wi-Fi network of the testing apartment with an immediate response time between the app configuration and the prototype's response. Nonetheless, depending on the installation of the site, possible sources of electromagnetic interference must be considered since it leads to an improper operation of the prototype.

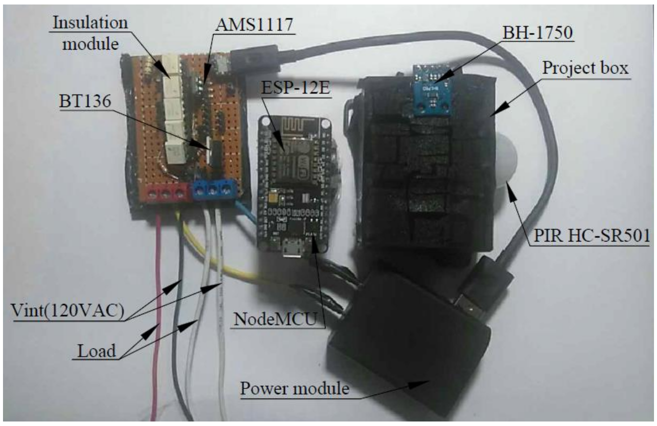


Figure 10. Developed prototype

Technical specifications

In table 2, the technical specifications of each domotics module are shown.

Table 2. Technical specifications

Technical specifications	
Connectivity:	WiFi 2.4GHz
Module WiFi:	ESP-12E
Supply voltage:	5VDC
Maximum current:	140mA
Rated current:	84mA
Approximate energy consumption:	0.49 W/h
Commercial Cost (Und):	$ 481.446
Cost Direct Materials (Und):	$ 92.500
Weight (g):	120
Dimensions (Length * Width * Height) cm:	7*4*3
Maximum load current:	4 A
Maximum load voltage:	220 V

Marginal cost

The marginal cost was estimated as a basis for the market analysis of automation and domotics companies, located in the city of Bogotá. The values include IVA (national tax rate). See table 3.

Table 3. Marginal cost

Id	Description	Quantity	COL$/Und	Total price
1	NodeMCU 1.0	1	$ 25.000	$ 25.000
2	MOC3010M	4	$ 2.000	$ 8.000
3	Resistances	8	$ 100	$ 800
4	Luminosity sensor BH-1750	1	$ 13.000	$ 13.000
5	Motion detector PIR HC-SR501	1	$ 5.000	$ 5.000
6	Optoisolators H11AA1	1	$ 1.500	$ 1.500
7	Connector USB-C	1	$ 2.000	$ 2.000
8	Regulator AMS1117-3.3	1	$ 2.500	$ 2.500
9	Triac BT136	1	$ 2.000	$ 2.000
10	Heat dissipator	1	$ 1.200	$ 1.200
11	Terminal X 3	2	$ 1.500	$ 3.000
12	Push-button	1	$ 1.000	$ 1.000
13	Condenser	1	$ 300	$ 300
14	Baquela (4X7)cm	1	$ 1.200	$ 1.200
15	Project box	1	$ 6.000	$ 6.000
16	Power module	1	$ 20.000	$ 20.000
	Total (Tax included)			**$ 92.500**

User manual

The app's management consists on two main panels. When the app starts, the main panel is deployed where the lighting to control can be chosen. Once chosen, the assigned site will be highlighted in green and the control options will be enabled which are: ON/OFF control, intensity control, motion control, automatic control and alarm-based control

ON/OFF control: A button is used to control the ignition or turnoff of the lighting. It is important to point out that the icon would change status only when the NodeMcu has executed the assigned command. It it does not work, this means that the board is not connected to the Wi-Fi network.

Intensity control: Through a slider, the user can vary the intensity of the lighting between four ranges (25, 50, 75 and 100%). It is noteworthy to mention that this type of control can only be applied to incandescent lighting points or dimmable LEDs.

Motion control: By using a switch, the PIR HC-SR501 motion control is activated or deactivated. The icon changes status only when the NodeMcu board has executed the assigned command. If the background is green, it means that the sensor is activated. It is also noteworthy to point out that the distance and turnoff time after the detection are two parameters that are manually adjusted in the sensor.

Automatic control: By using a button, the lighting level control is activated or deactivated. The NodeMcu board will constantly process the level of lighting (natural or artificial), energizing the lighting when the level of light is under 10 lux. In the opposite case, the lighting will be de-energized. Each time the user manipulates the app, the current lighting level can be visualized in the chosen location of the lighting point.

Alarm-based control: In this type of control, the user can set a specific time in which the lighting should be energized. To access this type of control, the user should select the clock icon. In Figure 11, the alarm-based control panel is shown.

Figure 11. Alarm-based control programming

CONCLUSIONS

Internet of Things is a new field in the academic and industrial sectors. It is constantly producing new components and integration forms that mark a real time overview of the Smart Home concept within a community of Open Source developers and innovators.

The IoT concept is base on sustainable bases such as the rational use of energy, comfort, security and even more the added value that the apartment acquires with these types of technologies.

The versatility of Arduino and the platforms for app development make possible to offer a domotics server for a low cost.

References

[1] Arduino. (s.f.). arduino.cc. Recuperado el 2017 de 06 de 21, de https://www.arduino.cc/

[2] Benjamin Xie, H. A. (2016). Skill Progression in MIT App Inventor. Department of Electrical Engineering and Computer Science.

[3] Gómez Víctor, Peña Robin, Hernández Cesar. (2010). Diseño y construcción de un prototipo de medición de los indicadores de calidad del servicio de energía eléctrica (des y fes) para usuario residencial. Revista Lasallista de Investigación, vol. 7, no. 2, pp. 101-112.

[4] Hernández Cesar, Márquez Hans, Martínez Fernando. (2015). Propuesta tecnológica para el mejoramiento de la educación y la inclusión social en los niños sordos. Formación Universitaria, vol. 8, no. 6, pp. 107-120.

[5] Hernández Cesar, Salcedo Octavio, Diaz Andrés. (2009). An ARIMA Model for forecasting Wi-Fi data network traffic values. Revista Ingeniería e Investigación, vol. 29, no. 2, pp. 65-69.

[6] Hernández Cesar, Salgado Camila, Rodríguez-Colina Enrique, López Hans. (2015). Multivariable algorithm for dynamic channel selection in cognitive radio networks. EURASIP Journal on Wireless Communications and Networking, vol. 2015, no. 216, pp. 1-17

[7] Márquez Hans, Salgado Camila, Hernández Cesar. (2017). Multichannel Assignment with Load Balance in Cognitive Radio Networks. Indian Journal of Science and Technology, vol. 10, no. 33, pp. 1-17.

[8] Montesdeoca Contreras J.C, A. C. (2015). Mobile Applications Using TCP/IP-GSM Protocols Applied to Domotic.

[9] Navjot Kaur Walia, P. K. (2016). An IOT by Information Retrieval aprroach: Smart Light controlled using WiFi.

[10] Pedraza Luis, Hernández Cesar, López Danilo. (2013). Sistema de comunicación TCP/IP para el control de una intersección de tráfico vehicular. Revista Ingeniería Investigación y Tecnología, vol. 14, no. 4, pp. 583-594.

[11] Quintero, L. F. (2005). Viviendas inteligentes (Domótica). REVISTA DE INGENIERÍA E INVESTIGACIÓN, 1.

[12] Riccardo Pozza, M. N. (2015). Neighbor Discovery for Opportunistic Networking in internet of Things Scenarios: A Survey. IEEE Access.

[13] Romero Ricardo, Giral Diego, Cesar Hernández. (2010). Analysis and review of the contribution of neural networks to saving electricity in residential lighting by a design in Matlab. Revista DYNA, vol. 77, no. 163, pp. 248-259.

[14] Santander Hubert, Prada Michael, Hernández Cesar. (2017). Measurement and Evaluation of the Spectral Duty Cycle in a Wi-Fi Network. International Journal of Engineering and Technology, vol. 9, no. 5, pp. 3783-3789.