Use of an embedded system with wifi technology for domotic control of conventional environments

F O Ovalles¹, A E Bolivar² and A J Rodriguez²

¹ Grupo de Investigación en Competitividad y Sostenibilidad para el Desarrollo, Universidad Libre de Colombia, San José de Cúcuta, Colombia
² Grupo de Investigación GISOFT, Universidad de Santander, San José de Cúcuta, Colombia

E-mail: freddyo.ovallesp@unilibre.edu.co

Abstract. The rise of emerging technologies and the emergence of open and free hardware and software trends have allowed the emergence of new embedded system options such as the Nodemcu card based on the esp8266 microcontroller, among many others, which at very low cost allows implement an On/Off control or domotic control for an environment or location that improves its comfort and energy saving conditions as main objectives. Understanding home automation as the system capable of automating a home or building of any kind, which is capable of providing energy management, security, welfare, maintenance and communication services, and which can be integrated by means of wireless networks or Wi-Fi technology, to give a control that provides certain ubiquity, from inside and outside the place through the use of an application or app, provides a huge opportunity to take advantage of current technology adapted to any type of environment or space. This use could be defined as the integration of technology in the intelligent design of a closed area, including conventional training environments or classrooms, which unlike the rest of other locations or modern properties, do not enjoy the same advances with which they enjoy new establishments of offices, banks and hotels that make our daily activities more pleasant. This research will use the quasi-experimental method since with the collaboration of experts through the network and the state of the art consulted, small experiments or tests have been developed using low-cost technology modules, thus checking the functioning of different connections and configurations, which generates the sending and receiving of data, which allows us to give or not advance to the analysis, programming and revision of the behavior of the occupation of an environment, making the focus towards a short-term future of the research quantitative.

1. Introduction

Global warming is generating an increase in environmental temperatures, generating that people adapt their spaces for habitat, work and training, in addition to this, the use of electrical appliances in the home or office, increases in search of comfort, even for scenarios of learning, making the cost of energy increase due to its use and also to its waste due to the lack of automated systems or domotic systems [1] that allow the management and optimization of the use of an environment or coexistence space for the development of daily tasks.

Home automation [2] is responsible for automating these tasks and processes in places such as homes, buildings or classrooms and gives us the option to improve power consumption with the implementation of some high-cost commercial systems that also allow remote administration, using
wireless or Wi-Fi connection, unlike this type of solutions, this project seeks to implement a low-cost solution, which is achievable for a common person with basic knowledge, taking into account the benefits of advances in technologies emerging markets and the opportunity to acquire modules and devices through the Internet.

A prototype was designed with the Nodemcu card based on the ESP8266 microcontroller, using Arduino's integrated IDE development environment [3], one of the most common open-hardware and software embedded system boards, available in its programming for the creation of hardware prototypes with its own development card or other cards such as Nodemcu, which in turn is an open access platform for the creation of electronic prototypes.

The programming of the control will be done through functions that send AT commands and codes in HTML language that transmit the data of the occupancy schedules of the environment from a simple database [4] hosted on a free server such as https://infinityfree.net/ and tests with appliances such as a lamp and an outlet that are established in a conventional environment; tests were carried out with the prototype controller already programmed with a free app [5].

2. Methodology

2.1. Justification and problem statement
The community in general, and especially the academic community, seeks to be more aware of the protection of the planet and especially the use and consumption of energy in conventional environments in which waste of energy, which are present, for example, in the management of lights, electrical outlets and the air conditioners of the classrooms is very evident [6].

However, this project seeks to reduce the waste of energy applied to the connection or disconnection of different electrical appliances from an online programming and thus demonstrate once again, [7] that ICT can be available to everyone even in the most unimaginable.

This allows us to raise the following problem question, to which the present investigation responds: How to improve the management of a conventional environment by using an embedded system with Wi-Fi technology?

2.2. Methodological design
The methodological design is based on the ADDIE model whose phases are: analysis, design, implementation and evaluation; the general objective of the project corresponds to: Use the implementation of an embedded system with Wi-Fi technology for the domotic control of a conventional environment [3]. The achievement of the general objective is related to the specific objectives that correspond to the phases of the methodological design: Analyze the problem situation of the waste of electrical energy caused by the habitat in conventional environments, Design the prototype of the device and the base of data that complies with the control of the system, Implement the embedded system and information on the web that works as intended and Evaluate the device prototype implemented by running tests of the application.

This research is an ongoing project that uses the quasi-experimental method since with the collaboration of experts in the network and the documentary review we have developed small experiments using technology and so we will be checking the behavior of different connections and configurations, which generates or receives data, which allow us to give continuity to the test, making the focus of research to a short-term future is quantitative by the rise of an emerging technology such as the Internet of Things [7], this can be seen in works consulted and mentioned.

2.3. Theoretical framework and state of the art
Rasgado and Reyes, in 2013, [8] in their degree work presented a prototype controlled via the internet with an internal network; that controls the equipment among themselves; home automation advances every day in the automation of homes, so that in the future people have the possibility to control eg. air conditioning from anywhere through the use of an intranet network through a time schedule.
An example of the use of the Arduino embedded system and low-cost components to control and improve the daily coexistence tasks such as the supply of water for the home by villagers in Indonesia, can be found in the article: "Automatic Water Tank Filling System Controlled Using Arduino Based Sensor for Home Application", [9] here the system allows to improve the waste of water and the waste of electric power for the drive of the tank filling pump from the well.

In the OpenFridge project, [10] an Internet of Things semantic platform is designed and implemented, which was evaluated with real-life users distributed globally. Real-life user and refrigerator data measurements have been compiled and published in open source repositories. A high potential has been demonstrated in the facilitation of data economy in evaluations. This work is stated in the article: "Contribute to the energy efficiency of appliances with the Internet of Things, smart data and user participation."

2.4. The embedded system wi-fi nodemcu card
The ESP8266 is a low-cost Wi-Fi chip with full TCP/IP protocol stack and MCU (Micro Controller Unit) capability produced by the Chinese manufacturer Espressif Systems, based in Shanghai. The first chip came to the attention of Western manufacturers in August 2014 with the ESP-01 module. This small module allows microcontrollers to connect to a Wi-Fi network and make simple TCP / IP connections using Hayes commands.

The Nodemcu v2 is a card with a microcontroller based on the ESP8266, it has 6 more pins, 4 are dedicated to SPI communication, (CS0, MISO, MOSI, and SCLK). It can be programmed with the Arduino IDE with the ESP8266 plugin. This microcontroller can do the same functions as an Arduino UNO, it has the firmware that is the low level software that allows to control the electrical circuits. By default, the version with which ESP8266 can be communicated by AT commands through the serial port is already installed; This type of communication allows bridge between the Arduino and the ESP8266 and in this way the Arduino UNO connects to a Wi-Fi network.

To perform this research, several materials was provided such as a Nodemcu card based in microcontroller ESP8266 with Wi-fi connectivity, a relay, a LCD 16x2 display with I2C module and electric plastic box to cover the equipment as shown in Figure 1.

![Figure 1](image1.png)

2.5. The prototype of the domotic control system based nodemcu card
Device prototype based on the Nodemcu card and experimental tests of sending and receiving data from the web to the device that does the On/Off control, as can be shown in Figure 2.

2.6. Arduino™ code segment for prototype of the domotic control system.
Arduino is an open source electronics platform based on easy to use hardware and software. This microcontroller board uses the Atmega328P. The script must firstly be uploaded to the board so that the device can be controlled based on our desire. Table 1 shows the Arduino™ code of the domotic control system.
Figure 2. Prototype of the domotic control system.

Table 1. Arduino™ codes for prototype of the domotic control system.

| The void setup Arduino™ codes                                                                 | The void loop Arduino™ codes                                                                 |
|-----------------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------|
| `#include <ESP8266WiFi.h>`                                                                  | `lcd.setCursor(0, 1);`                                                                       |
| `#include <WiFiClient.h>`                                                                    | `lcd.println("Conectando... ");`                                                             |
| `#include <ESP8266WebServer.h>`                                                              | `while (WiFi.status() != WL_CONNECTED) {`                                                      |
| `#include <Adafruit_GFX.h>`                                                                   | `delay(1000);`                                                                               |
| `#include <ESP_Adafruit_SSD1306.h>`                                                           | `Serial.println(".");`                                                                       |
| `#include <LiquidCrystal_I2C.h>`                                                              | `}                                                                                           |
| `#define OLED_RESET 4`                                                                          | `lcd.setCursor(0, 1);`                                                                       |
| Adafruit_SSD1306 display(OLED_RESET);                                                        | `Serial.println("Connected to ");`                                                           |
| LiquidCrystal_I2C lcd(0x3F, 16, 2);                                                            | `lcd.setCursor(0, 0);`                                                                       |
| #if (SSD1306_LCDHEIGHT != 64)                                                                  | `lcd.print("Conectando a: ");`                                                               |
| #error("Height incorrect, please fix Adafruit_SSD1306.h!");                                 | `Serial.println(ssid);`                                                                      |
| const char* ssid = "UnilibreCucuta";                                                          | `lcd.print(ssid);`                                                                           |
| const char* password = "";                                                                    | `delay(1500);`                                                                               |
| ESP8266WebServer server(80);                                                                  | `lcd.clear();`                                                                               |
| const int output1 = 14;                                                                       | `Serial.print("Direccion IP:   ");`                                                           |
| const int output2 = 16;                                                                       | `lcd.clear();`                                                                               |
| const int output3 = 02;                                                                       | `Serial.println(WiFi.localIP());`                                                            |
| const int output4 = 15;                                                                       | `lcd.print(WiFi.localIP());`                                                                 |
| boolean device1 = false;                                                                      | `if (MDNS.begin("esp8266")) {`                                                                |
| boolean device2 = false;                                                                      | `Serial.println("MDNS responser started");`                                                 |
| boolean device3 = false;                                                                      | `}`                                                                                          |
| boolean device4 = false;                                                                      | `server.on("/status1=1", [](){`                                                              |
| void handleRoot() {                                                                             | `server.send(200, "text/plain", "device1 = ON");`                                          |
| String cmd;                                                                                   | `digitalWrite(output1, HIGH);`                                                                |
| cmd += "<!DOCTYPE HTML>\n\n";                                                                  | `device1 = true;`                                                                             |
| cmd += "<html>\n\n";                                                                            | `});                                                                                          |
| Webserver<title><h1>ESP8266 Web Server Control</h1><header>";                                | `server.send(200, "text/plain", "device1 = OFF");`                                           |
| cmd += "<head>";                                                                               | `digitalWrite(output1, LOW);`                                                                |
| cmd += "<meta http-equiv='refresh' server.on("/status2=1", [](){`                            | `device1 = false;`                                                                            |
|                                                                                               | `});                                                                                          |
void handleNotFound()
{
    String message = "File Not Found\n\nURI: "+ server.uri();
    message += "\nMethod: " + (server.method() == HTTP_GET) ? "GET" : "POST";
    message += "\nArguments: ";
    for (uint8_t i=0; i<server.args(); i++){
        message += " " + server.argName(i) + " : " + server.arg(i) + ";" ;
    }
    server.send(404, "text/plain", message);
}

void setup(void){
    pinMode(output1, OUTPUT);
    pinMode(output2, OUTPUT);
    pinMode(output3, OUTPUT);
    pinMode(output4, OUTPUT);
    digitalWrite(output1, LOW);
    digitalWrite(output2, LOW);
    digitalWrite(output3, LOW);
    digitalWrite(output4, LOW);
    Serial.begin(115200);
    server.send(200, "text/plain", "device2 = ON");
    digitalWrite(output2, HIGH);
    lcd.clear();
    lcd.setCursor(0, 0);
    lcd.print("Salida #2: ");
    lcd.setCursor(0, 1);
    lcd.print("<<Encendida>>");
    device2 = true;
    server.on("/status2=0", [](){
        server.send(200, "text/plain", "device2 = OFF");
        digitalWrite(output2, LOW);
        lcd.clear();
        lcd.setCursor(0, 0);
        lcd.print("Salida #2: ");
        lcd.setCursor(0, 1);
        lcd.print("<<Apagada->>>");
        device2 = false;
    });
    server.on("/status3=1", [](){
        server.send(200, "text/plain", "device3 = ON");
        digitalWrite(output3, HIGH);
        lcd.clear();
        lcd.setCursor(0, 0);
        lcd.print("Salida #3: ");
        lcd.setCursor(0, 1);
        lcd.print("<<Encendida>>");
        device3 = true;
    });
    server.on("/status3=0", [](){
        server.send(200, "text/plain", "device3 = OFF");
        digitalWrite(output3, LOW);
        lcd.clear();
        lcd.setCursor(0, 0);
        lcd.print("Salida #3: ");
        lcd.setCursor(0, 1);
        lcd.print("<<Apagada->>>");
        device3 = false;
    });
    server.on("/status4=1", [](){
        server.send(200, "text/plain", "device4 = ON");
        digitalWrite(output4, HIGH);
        device4 = true;
    });
    server.on("/status4=0", [](){
        server.send(200, "text/plain", "device4 = OFF");
        digitalWrite(output4, LOW);
        device4 = false;
    });
    server.onNotFound(handleNotFound);
    server.begin();
}

void loop(void){
    server.handleClient();
}
WiFi.begin(ssid, password);
Serial.println(
""");
lcd.begin(16,2);
lcd.init();
lcd.backlight();
lcd.setCursor(1, 0);
lcd.print("CONTROL ESP8266");
lcd.setCursor(1, 1);
lcd.print("Proy AirConditii");
delay(2000);

2.7. Test for prototype of the domotic control system with free App

Other tests were carried out with the free app as an alternative to managing the home automation system from the smartphone. Figure 3 shows the ESP8266 Wifi Control [5] app running on a Samsung Galaxy S tablet.

![Figure 3. Test for Prototype of the domotic control system with free App.](image)

As an additional result, [4] seeks to minimize energy consumption and optimize the use of domestic appliances and air conditioners, with the use of emerging technologies applied to the conservation of the environment, such is the case of applicability that students intend to perform Emily Bolivar and Argel Rodriguez at the Universidad de Santander with the work of improving the electric power consumption of air conditioners at the Universidad de Santander Campus Cúcuta through the use of an embedded system with Wi-Fi technology.

The development of this project has been based on tests of a prototype and the review of several researches [12,13] that has allowed us to advance the group of researchers within the research center formed by the research teacher and young students engineers in training within the strategy pedagogical.

3. Conclusion

This work is an ongoing investigation, that is open to continue advancing in the improvements related to the collection, evaluation and analysis of collected data, which allow establishing methods and times, such as average use times between appliances and users, as well as the user experience in front of them, an efficient management of energy, through the monitoring, diagnosis, control and maintenance of electrical appliances and the implementation of IoT protocols such as the MQTT and
its security aspects, all this is highly important thanks to the use of Wi-fi connectivity, WSN networks and technological trends such as big data, IoT and emerging technologies.

Acknowledgments
This project was developed within the Research Seedbed "ICT in the Processes" and with the support of the Sectional Research Center of the Universidad Libre Seccional Cucuta, in collaboration with the Universidad de Santander within the framework of a technology transfer.

References
[1] Laclaustra I M, Alonso J M, Del Barrio A A y Botella G 2016 Sistema Domótico Distribuido para Controlar el Riego y el Aire Acondicionado en el Hogar Enseñanza y Aprendizaje de Ingeniería de Computadores 6 87-101
[2] Li M, Gu W, Chen W, He Y and Wu Y 2018 Smart Home: Architecture, Technologies and Systems Procedia Computer Science 131 393-400
[3] Sánchez E L 2012 Diseño de un sistema de control domótico basado en plataforma arduino (Valencia: Universidad Politécnica de Valencia) p 44
[4] Putra L, Yudishitra M and Kanigoro B 2015 Design and Implementation of Web Based Home Electrical Appliance Monitoring, Diagnosing, and Controlling System Procedia Computer Science 59 34-44
[5] Wifi Control Google Play ESP8266 Consulted on: https://play.google.com/store/apps/details?id=com.lacour.vincent.wificaresp8266&hl=en_US
[6] Park J, Han S H, Park J, Park J, Lee M and Kwalika J 2018 Development of a web-based user experience evaluation system for home appliances International Journal of Industrial Ergonomics 67 216-228
[7] Cravioto J, Yasunaga R and Yamasue E 2017 Comparative Analysis of Average Time of use of Home Appliances Procedia CIRP 61 657-652
[8] Kashyap M, Vidushi S and Neeti G 2018 Taking MQTT and NodeMcu to IOT: Communication in Internet of Things Procedia Computer Science 132 1611-1618
[9] Rasgado F R y Reyes C E 2013 Control de aire acondicionado mediante Internet (Mérida, México: Instituto Técnico de Mérida) p 44
[10] Prima E C, Munifaha S S, Salam R, Aziz M H and Suryanic A T 2017 Automatic Water Tank Filling System Controlled Using Arduino™ Based Sensor for Home Application Procedia Engineering 170 373-377
[11] Fensel A, Tomic D K and Koller A 2017 Contributing to appliances’ energy efficiency with Internet of Things, smart data and user engagement Future Generation Computer Systems 76 329-338
[12] Abdurahman T A, Isiwekpeni O H, Surajudeen-bakinde N T and Otuoze A O 2016 Design, Specification and Implementation of a Distributed Home Automation System Procedia Computer Science 94 473-478
[13] Korkmaz I, Metin S K, Gurek A, Gur C and Gurakin C 2015 A cloud based and Android supported scalable home automation system Computers & Electrical Engineering 43 112-128