IIETA International Information and Engineering Technology Association
Advancing the World of Information and Engineering

Review of Computer Engineering Studies
Vol. 7, No. 3, September, 2020, pp. 49-54
Journal homepage: http://iieta.org/journals/rces

Information Gathering and Controlling over the Internet by Internet of Things (IoT)

Abderrahmane Herbadji[1*], Djamel Herbadji[2], Abdelhadi Labiad[3]

[1] Laboratoire d'Analyse des Signaux et Systémes (LASS), Department of Electronics, University of M'sila, BP 166, Route Ichebilia, M'sila 28000, Algeria
[2] Laboratoire de Recherche en Electronique de Skikda, Université 20 Août Skikda, BP 26 Route d'El-hadaeik Skikda 21000, Algeria
[3] Telecommunication, Signals &amp; Systems Laboratory, University of Laghouat, Laghouat 30000, Algeria

Corresponding Author Email: abderrahmane.herbadji@univ-msila.dz

https://doi.org/10.18280/rces.070301

Received: 11 July 2020
Accepted: 20 August 2020

Keywords:
Internet of Things, embedded systems, WiFi, Arduino

ABSTRACT

The main goal of this work is to create an application that has the ability to communicate, connect, and control physical entities via the Internet. This purpose can be achieved by the Internet of Things or Internet of everything (The more complete form) which is a new revolution of the Internet to allow objects to be connected to the internet. Connected objects can communicate with each other or with us via an HTML interface that allows to create a communication between the user and the server ESP8266 through its IP address. Objects have been connected wirelessly (WiFi communication). A sensor is used to collect data then transfer it to the user via the Internet in order to make correct decisions in anywhere at any time.

1. INTRODUCTION

Technology has become an integral part of our lives, especially the internet. The internet is now widely used by most people for a wide range of services: file sharing, information retrieval, online shopping, banking, social media, etc. [1-3]. But as the Internet continues to evolve, it not only connects people with each other or with a service, but also allows objects to connect with each other to get and share information or to take action [4, 5]. The Internet of Things is currently translating into an increase in the number of connected objects. These objects carry a growing number of sensors and actuators allowing them to measure the environment and act on it, thus making the link between the physical world and the virtual world [6, 7]. Specifically, the Internet of Things poses several problems, in particular because of its very large scale, its dynamic nature and the heterogeneity of the data and the systems that compose it (powerful / not very powerful, fixed / mobile devices, powered by batteries / DC power supplies, etc.). These characteristics require tools and methods for the creation of applications capable of extracting useful information from the many available data sources and of interacting with the environment, by means of actuators, as well as with users, through dedicated interfaces. Indeed, the Internet of Things must be designed for easy use masking the underlying technological complexity, and quiet handling preventing threats and potential risks [8]. In the IoT, any object is potentially connected to the Internet and capable of communicating with other objects. This creates new security threats [9, 10]. In the past, the Internet linked traditional objects such as PCs, tablets, etc. But currently the internet has to connect non-traditional objects [11].

In this work anew will try to connect previously unconnected objects such as sensors and actuators (example: LEDs, fans, temperature sensors) (see Figure 1), where information technology will become integrated into the objects of our daily life.

An application based on HTTP protocol has been created. this application allow user to control the objects via The ESP8266 module, which offers a WiFi communication. This module is an electronic “component” that can easily be integrated into electronic assemblies and will allow us to create a client-server application. The application is subdivided into two operating modes, automatic mode and manual mode, in the latter mode the user will interact with the objects connected through the IP address of the server. The fan and LED control are managed by PWM pulse width modulation. We are going to create an HTML interface in order to allow the user to send and receive requests and responses respectively using a web browser that uses the HTTP protocol.

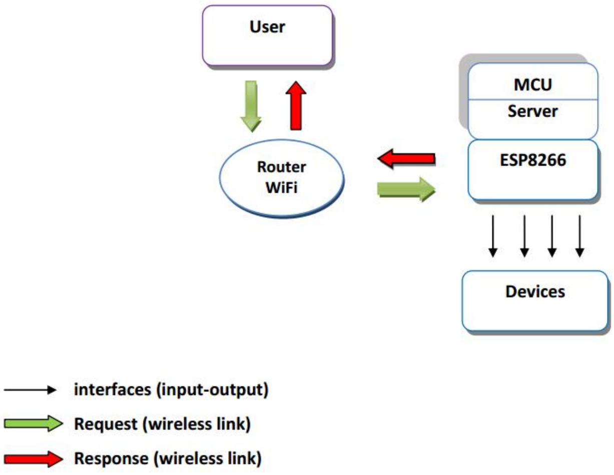


Figure 1. Principle of operation

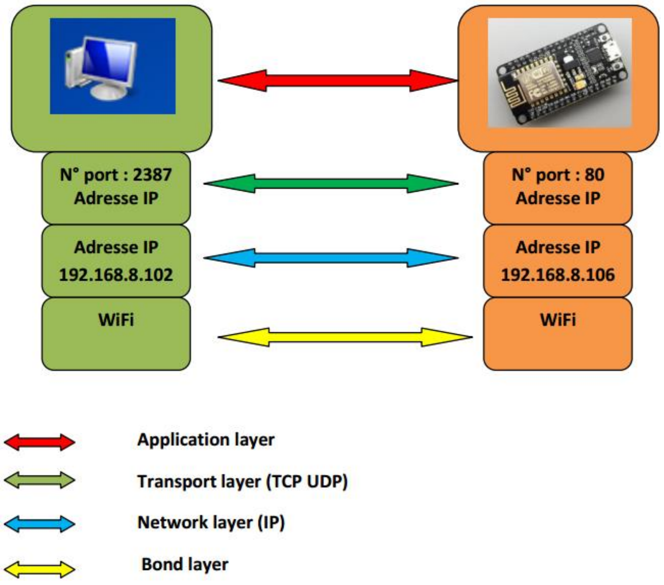
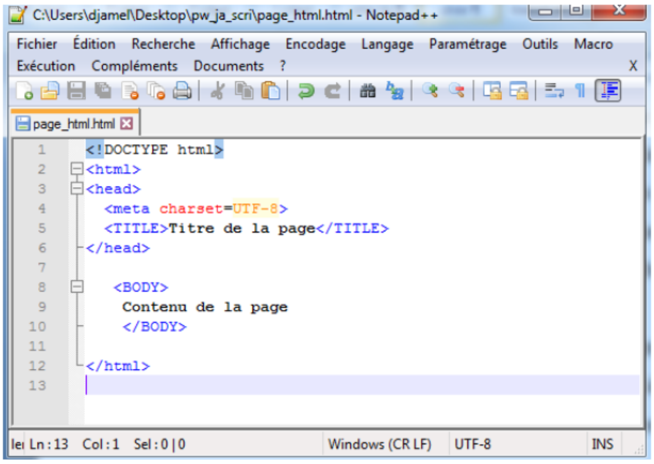


Figure 2. Communication protocol sn


C:\Users\djamel\Desktop\pw_ja_scri\page_html.html - Notepad++
Fichier Édition Recherche Affichage Encodage Langage Paramétrage Outils Macro
Exécution Compléments Documents ?
page_html.html
lei Ln : 13 Col : 1 Sel : 0 | 0
Windows (CR LF)
UTF-8
INS

```
<!DOCTYPE html>
<html>
<head>
  <meta charset=UTF-8>
  <TITLE>Titre de la page</TITLE>
</head>

   <BODY>
    Contenu de la page
    </BODY>

</html>
```

Figure 3. Minimalist HTML5 page

As a result, various physical objects have been controlled. where the elements are controlled remotely and automatically. Each physical object has an IP address in the communication network. Therefore, we could independently send/receive the data to/from the various physical objects.

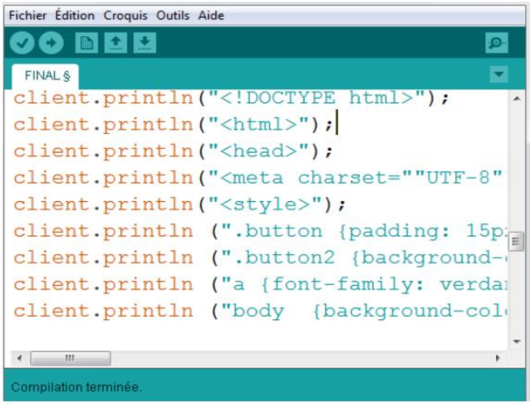
Objects are capable of communicating, either with each other or with a command interface. We will use a development kit based on the ESP8266 module which integrates GPIO, PWM.

I2C, ADC and WiFi communication (with an integrated mini antenna). As additional components we will use a temperature sensor (LM35), LEDs and a fan (DC motor).

2. PRINCIPLE OF OPERATION

The ESP8266 behaves like a web server and like a microcontroller. The user issues a request to the server using his IP address and his port number, which designates a particular service of the server, the server receives the request and responds using the address of the client machine and its port. The router ensures the orientation of the data towards such or such port according to the source and destination IP addresses of the message. The microcontroller is passive until the arrival of a request. When a request arrives, it performs the appropriate action.

Our application resembles the client-server model. The application layer provides an interface to the user, the communication between the processes is managed by the transport layer and its protocols (TCP, UDP) [12, 13]. Network layer protocols (IP address) organize communication data from the source host to the destination host. Finally, the data link layer prepares the network layer packets in order to transmit them. (see Figure 2).



```
client.println("<!DOCTYPE html>");
client.println("<html>");
client.println("<head>");
client.println("<meta charset=""UTF-8"
client.println("<style>");
client.println (".button {padding: 15p
client.println (".button2 {background-
client.println ("a {font-family: verda
client.println ("body  {background-col
```

Figure 4. Arduino software interface

In the figure the port number 2387 represents the local port of the client process - randomly determined by the operating system, while port 80 is the port of the web server.

3. DEVELOPMENT TOOLS

3.1 Hardware tools

ESP8266 node MCU: The ESP-12E WiFi module is developed by Ai-thinker Team, it offers a complete and autonomous Wi-Fi network solution [14].

3.2 Outils software

Notepad ++: It is a website creation tool that supports several languages. By optimizing many functions while maintaining ease of use and a certain user-friendliness. Each user can define the language they want to use, we used Notepad++ with HTML (Hyper Text Markup Language) and CSS (Cascading Style Sheets) to create our web interface.

HTML5 language: Html is the standard markup language for creating web pages, it defines the content of web pages.HTML elements are represented by tags. Browsers do not display HTML tags, but use them (see Figure 3).

CSS3 language: CSS3 is the latest CSS standard. CSS isa language that describes the style of an HTML document. It describes how HTML elements should be displayed. In a CSS code, there are three different elements: tag names, CSS values and properties.

Arduino software: The Arduino modules programming software is a Java application, free, multi-platform, source ope, and easy to use, serving as code editor and compiler. It also makes it possible to transfer the firmware and the program to our module through the USB serial link [15]. The programming language used in the Arduino software is the C language. The source code of our web interface was written with the editor Notepad ++ using the HTML5 language and CSS3(see Figure 4). With the Arduino software, we translate the code into C language and transfer it to the ESP8266.

4. IMPLEMENTATION

Users can communicate with the connected objects (fan control, control of the LED brightness and the temperature measured by the LM35) through their electronic devices such as PCs, tablets and smart phones using the Internet source. The fan is controlled using an BC337 npn transistor (see Figure 5).

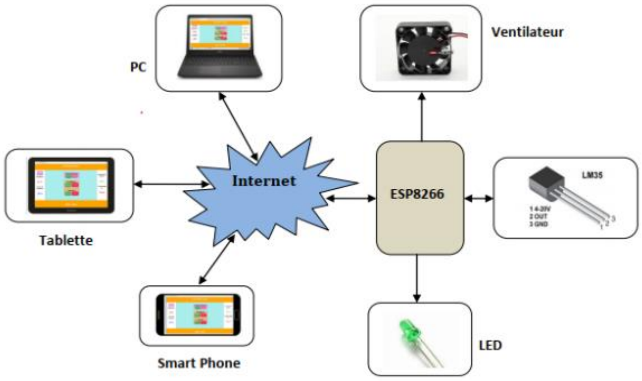


Figure 5. The proposed client-server application

4.1 Communication between ESP8266 and WiFi router

The first step we do is to have the ESP8266 communicate with the router wirelessly because the ESP8266 development board offers WiFi communication. This wireless communication solves distance and obstacle problems using radio waves. WiFi communication can be presented according to the Figure 7.

4.2 Client (user) - server communication (esp8266)

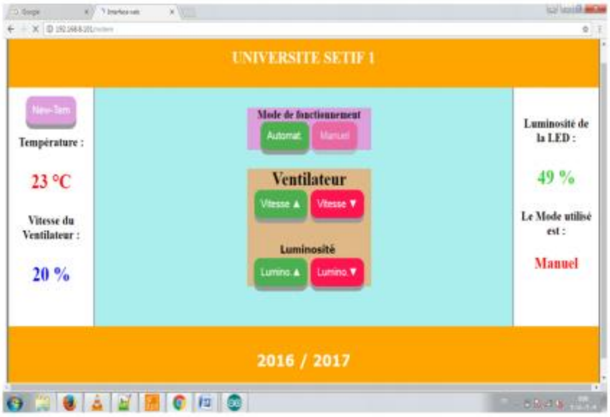
The server must still be active and wait for a request. If a client connects to the server, the server waits for a request. The request is sent to the server by the client (user) asking it to perform an operation. The server can respond to or serve multiple users or clients simultaneously. After processing the request sent by the user, the server sends a response to the client [5]. The flowchart below explains this communication, see Figure 8.

4.3 The fan control

After reading the temperature displayed on our web interface (see Figure 6), we can increase or reduce the fan speed by 20% (the step chosen by us). The fan control is based on the PWM pulse width modulation, the operating principle of this modulation as shown in Figure 10 is to generate an analog signal using a digital source (the Esp8266 offers 9 pins which can be configured as PWM output). A PWM signal consists of two elements (Figure 9):

The duty cycle: describes the duration during which the signal is high, it is expressed in%.

Frequency: determines the speed at which the PWM cycles. The average voltage then depends on the duty cycle. The fan speed varies depending on this average voltage.


UNIVERSITE SETIF 1

Température : 23 °C

Vitesse du Ventilateur : 20 %

Luminosité de la LED : 49 %

Le Mode utilisé est : Manuel

2016 / 2017

Figure 6. The configuration of the operating mode

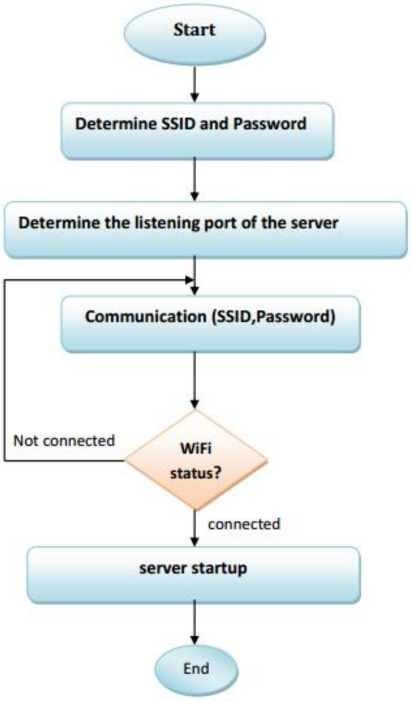

Start
Determine SSID and Password
Determine the listening port of the server
Communication (SSID,Password)
Not connected
WiFi status?
connected
server startup
End

Figure 7. ESP8266 WiFi communication flowchart

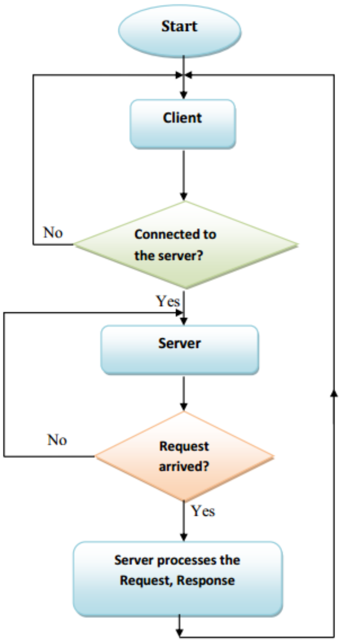


Figure 8. User communication program flowchart-Esp8266

The following flowchart (Figure 11) explains how the fan speed is varied based on a PWM signal. the flowchart is divided into two parts: one part is executed at the server level and the other part is performed by the user.

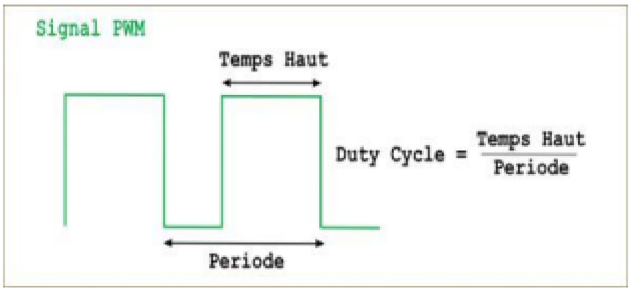


Figure 9. A PWM signal

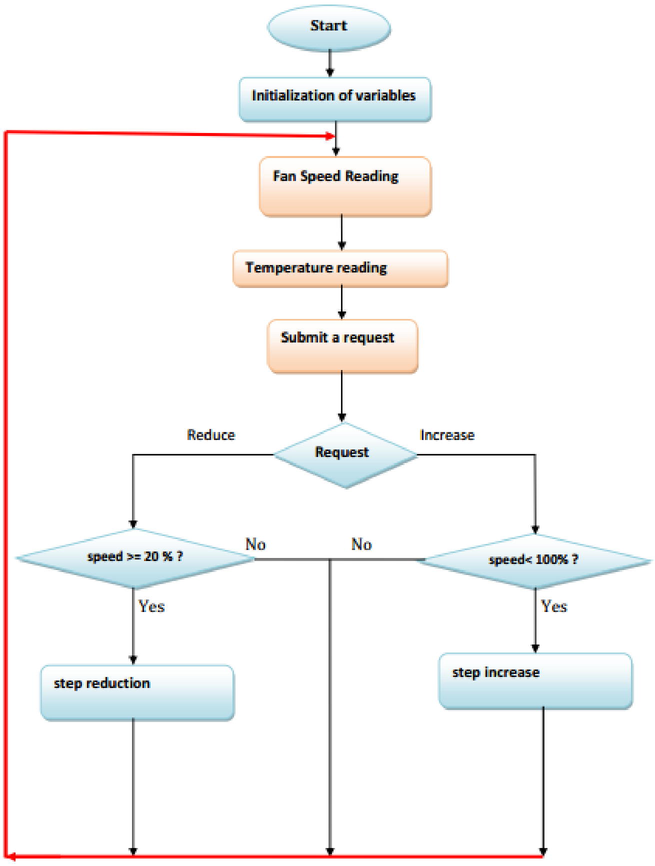


Figure 10. The fan control flowchart

4.4 LED brightness

Control The brightness of the LED is managed by a PWM signal, the step of variation of the brightness is ±25% (see Figure 12). After the brightness change (by the user), the server displays the new brightness value in the web interface. In our web interface, we have created buttons in order to manage the speed of rotation of the fan or the brightness of the LED, by pressing a button, the web browser sends a request to the server. The request / response exchange between the user and the server is carried out by the hypertext transfer protocol (HTTP) via the IP address and the port (80) of the server. HTTP was designed specifically for the web to allow access to information stored on servers. It works like a request - response protocol between the client and the server. The following figures show the interaction between the user and the server.

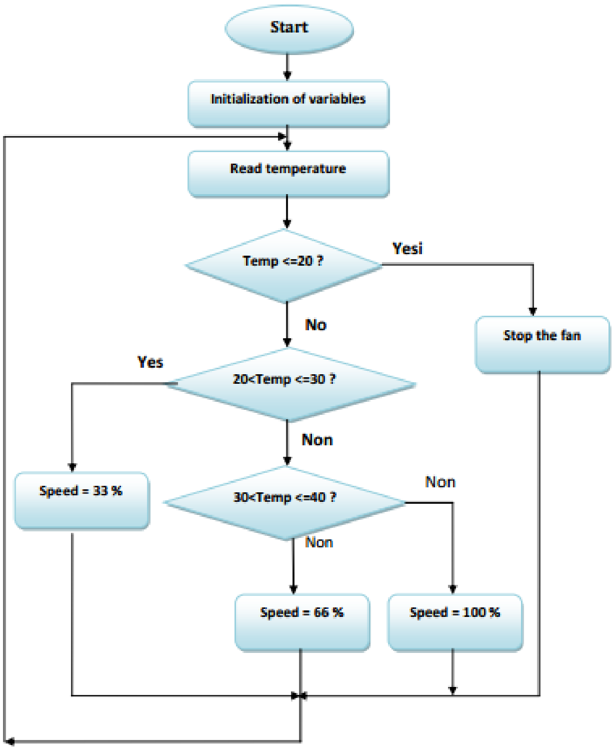


Figure 11. Automatic fan control flowchart

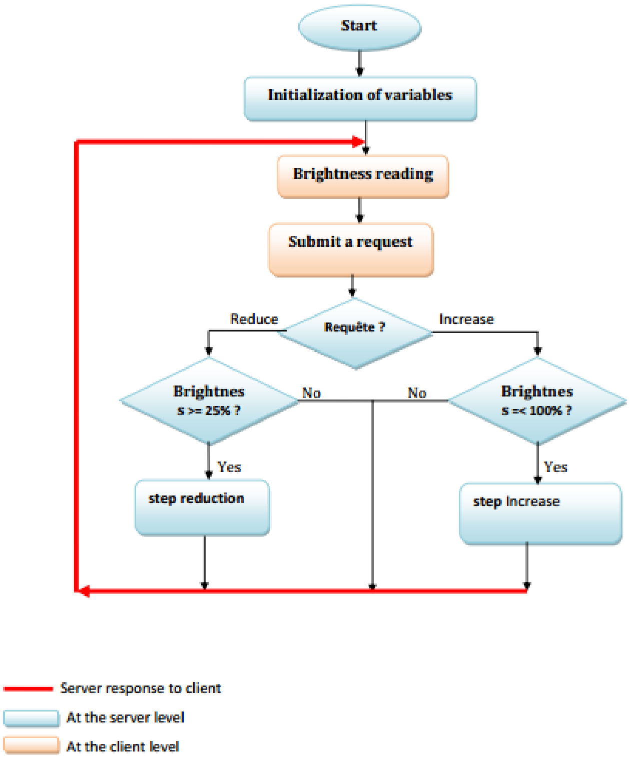


Figure 12. LED brightness control flowchart

4.5 Connected objects

In order to strengthen our application, we use two operating modes, a manual mode which allows us to monitor and control the objects connected remotely. This is the mode that we explained in the previous titles. The second mode is the automatic mode where objects communicate with each other. In this mode the ESP8266 module behaves like a classic microcontroller, and decides for itself the actions to take. The operating mode is configured (chosen) remotely by the user. The server deactivates the command buttons when a request for automatic mode arrives in order to avoid any conflict (see Figure 11). The electrical diagram of the connected objects is displayed in Figure 14.

5. MODELIZATION

The internet of things consists of four elements: process, data, people and things, our client-server system (application) can be translated into an IoE model as shown in Figure 13.

Things: the "things" are the physical devices and objects connected to the internet in our project the "things" are: the fan and the LED.

Data: The Data collected by the LM35 temperature sensor is processed at the level of the microcontroller and then disseminated on the Internet so that the best decisions are made.

Process: we can define a process as a transformation of information, it is the process carried out by the microcontroller. The microcontroller plays an important role because it receives information from the outside (temperature sensor), processes the information and then sends the commands to the outside (fan).

People: users are able to interact with connected objects thanks to the IP address of the ESP8266 module. Users (customers) can control objects from anywhere and anytime.

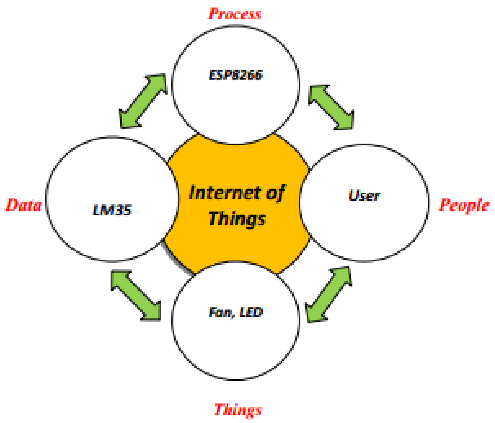


Figure 13. Connected objects

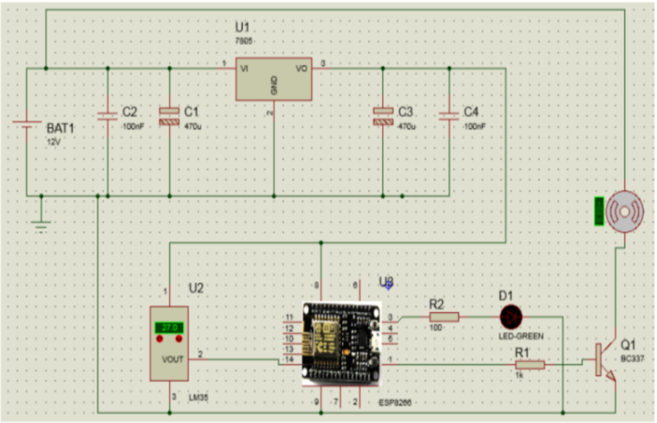


Figure 14. Electrical diagram of our application carried out

6. CONCLUSION

In this project we studied the new paradigm "Internet of things" (IoT), and as we have seen, IoT is a very promising concept where technologies are naturally integrated into everyday objects, because it affects almost all areas of our life. IoT relies heavily on the infrastructure used by the Internet to communicate between objects. The transition from the Internet of Things to the Internet of Everything (IoE) paves the way for connecting a very large number of objects, IoE tries to connect everything that can be connected to the Internet. The key elements of IoE are people, data, processes and objects (things). IoE makes it possible to create new connections between its different elements: Machine-to-machine (M2M), Machine-to person (M2P) and Person-to-person (P2P). The appearance of IPv6 which offers an almost unlimited address field facilitates communication between a large number of objects between them and also with us (users or customers). The large number of heterogeneous objects connected to the internet and the information exchanged between them in different fields with advanced technologies amplify the risks, vulnerabilities and security threats, IoT requires data confidentiality, integrity and authentication.

The objective of our application is to connect physical entities (examples: fan, temperature sensor and LED) to the Internet. The information collected by the temperature sensor is processed at the level of the ESP8266 microcontroller, then sent back to customers or other physical entities using the Internet telecommunications infrastructure. Our ESP8266 module also behaves like a web server to allow the user to interact with objects connected via the IP address of the ESP8266 server and the HTTP protocol. The user can interact with these objects through a web interface designed using the HTML5 and CSS3 languages. An "Internet of things" application allows us to control and monitor connected objects from anywhere at any time when needed. In the future work automatic measuring the level to alert the peoples via secured Short Message Service (SMS) mobile messages.

REFERENCES

[1] Herbadji, D., Belmeguenai, A., Derouiche, N., Liu, H. (2020). Colour image encryption scheme based on enhanced quadratic chaotic map. IET Image Process, 14(1): 40. https://doi.org/10.1049/iet-ipr.2019.0123

[2] Batalla, J.M., Mastorakis, G., Mavromoustakis, C.X., Pallis, E. (2017). Beyond the internet of things. Springer, Switzerland. https://doi.org/10.1007/978-3-319-50758-3

[3] Rayes, A., Salam, S. (2017). Internet of things from hype to reality. Springer, Switzerland. https://doi.org/10.1007/978-3-319-44860-2

[4] AL-mawee, W. (2012). Privacy and security issues in IoT healthcare applications for the disabled users a survey. Master's Theses. Department of Computer Science, Western Michigan University, USA.

[5] Patil, P., Desai, B.L. (2013). Intelligent irrigation control system by employing wireless sensor networks. International Journal of Computer Applications, 79(11): 33-40. https://doi.org/10.5120/13788-1882

[6] Fortino, G., Trunfio, P. (2014). Internet of things based on smart objects: Technology, middleware and applications. Springer, Switzerland. https://doi.org/10.1007/978-3-319-00491-4

[7] Singh, K.J., Kapoor, D.S. (2017). A survey of IoT platforms create your own internet of things. IEEE Consumer Electronics Magazine, 6(2): 57-68. https://doi.org/10.1109/MCE.2016.2640718

[8] Rizvi, S., Pipetti, R., McIntyre, N., Todd, J. (2020). Threat model for securing internet of things (IoT) network at device-level. Internet of Things, 11: 10024. https://doi.org/10.1016/j.iot.2020.100240

[9] Abomhara, M., Køien, G.M. (2015). Cyber security and the internet of things: Vulnerabilities, threats, intruders and attacks. Journal of Cyber Security and Mobility, 4(1): 65-88. https://doi.org/10.13052/jcsm2245-1439.414

[10] Herbadji, D., Derouiche, N., Belmeguenai, A., Herbadji, A., Boumerdassi, S. (2019). A tweakable image encryption algorithm using an improved logistic chaotic map. Traitment du Signal, 36(5): 407-417. https://doi.org/10.18280/ts.360505

[11] Agyemang, J.O., Kponyo, J.J., Klogo, G.S., Boateng, J.O. (2020). Lightweight rogue access point detection algorithm for WiFi-enabled internet of things (IoT) devices. Internet of Things, 11: 100200. https://doi.org/10.1016/j.iot.2020.100200

[12] Karar, M.E., Al-Rasheed, M.F., Al-Rasheed, A.F., Reyad, O. (2020). IoT and neural network-based water pumping control system for smart irrigation. Information Sciences Letters, 9(2): 107-112. http://dx.doi.org/10.18576/isl/090207

[13] Karar, M.E., Reyad, O., Shehata, H.I., Abdelhamid, A.A., Alotaibi, S. (2020). A new intelligent system for performance enhancement of dual-sensor cardiac pacemakers using iterative learning control. 6th Conference on Data Science and Machine Learning Applications (CDMA), Riyadh, Saudi Arabia, pp. 158-163. http://dx.doi.org/10.1109/CDMA47397.2020.00033

[14] Raju, M.P., Laxmi, A.J. (2020). IoT based online load forecasting using machine learning algorithms. Procedia Computer Science, 171: 551-560. https://doi.org/10.1016/j.procs.2020.04.059

[15] Cézar, K.L., Caldas, A.G.A., Caldas, A.M.A., Cordeiro, M.C.L., Dos Santos, C.A.C., Ochoa, A.A.V., Michima, P.S.A. (2020). Development of a novel flow control system with Arduino microcontroller embedded in double effect absorption chillers using the LiBr/H_2O pair. International Journal of Refrigeration, 111: 124-135. https://doi.org/10.1016/j.ijrefrig.2019.11.014