

Internet of things applied to agriculture using the ESP32 module in connection with the Ubidots platform

Internet de las cosas aplicado a la agricultura usando el módulo ESP32 en conexión con la plataforma Ubidots

MANZANERO-VAZQUEZ, Daniel Jesus†*, MANRIQUE-EK, Josué Abraham, CARDOZO-AGUILAR, Guadalupe and DECENA-CHAN, Carlos Alberto

Instituto Tecnológico Superior De Calkiní en el Estado de Campeche, Avenida AH-Canul, Sin Número, San Felipe, 24900 Calkiní, Campeche

ID 1st Author: *Daniel Jesus, Manzanero-Vazquez* / ORC ID: 0000-0002-8938-683X, Researcher ID Thomson: AAV-8327-2021, CVU CONACYT ID: 1148626

ID 1st Co-author: *Josué Abraham, Manrique-Ek* / ORC ID: 0000-0002-1369-35269727, Researcher ID Thomson: I-5873-2018, arXiv Author ID: EFYDCI-9MW4UV

ID 2nd Co-author: *Guadalupe, Cardozo-Aguilar* / ORC ID: 0000-0001-80332280, Researcher ID Thomson: I-5874-2018, arXiv Author ID: TLTSYW-VLXBVS

ID 3rd Co-autor: *Carlos Alberto, Decena-Chan* / ORC ID: 0000-0002-0223-8106, Researcher ID Thomson: AAX-9426-2021, CVU CONACYT ID: 1155496

DOI: 10.35429/JTP.2021.20.7.12.20

Received August 30, 2021; Accepted October 20, 2021

Abstract

This article presents the use of the ESP32 module as a linking device for the Internet of Things (IoT). A prototype was implemented in the area of agriculture, specifically, in greenhouses. This prototype allows us to read and control some variables present in the microclimate of a greenhouse. These variables are: percentage of relative humidity of the soil, the amount of illumination that the crops receive, the measurement of the pH of the soil and the environmental temperature. In the second instance, a pump that supplies water for irrigation can be controlled, as well as the on/off of the ultraviolet light that falls on the plants. That is why the MQTT protocol was used to communicate with the Ubidots platform, creating a client from the ESP32 module, which was programmed using the Arduino IDE only as an editor, compiler and code store.

ESP32 module, Protocol, Communication, Ubidots

Resumen

El presente presenta muestra el uso del módulo ESP32 como dispositivo de enlace para internet de las cosas (IoT). Se implementó a un prototipo del área de la agricultura, en concreto, en los invernaderos. Este prototipo nos permite leer y controlar algunas variables presentes en el microclima de un invernadero. Esas variables son: porcentaje de humedad relativa del suelo, la cantidad de iluminación que reciben los cultivos, la medición del pH que tiene el suelo y la temperatura ambiental. En segunda instancia podrá controlarse una bomba que suministre agua para el riego, así como el encendido/apagado de la luz ultravioleta que incida sobre las plantas. Es por ello que para la comunicación con la plataforma Ubidots se empleó el protocolo MQTT, creando un cliente desde el módulo ESP32, el cual se programó empleando el IDE de Arduino únicamente como editor, compilador y almacenador de código.

Módulo ESP32, Protocolo, Comunicación, Ubidots

Citation: MANZANERO-VAZQUEZ, Daniel Jesus, MANRIQUE-EK, Josué Abraham, CARDOZO-AGUILAR, Guadalupe and DECENA-CHAN, Carlos Alberto. Internet of things applied to agriculture using the ESP32 module in connection with the Ubidots platform. Journal of Technological Prototypes. 2021. 7-20: 12-20

* Correspondence to Author (email: 5695@itescam.edu.mx)

† Researcher contributing first author

Introduction

For around 40 years, technology has developed systems that support industries to monitor, manage their applications and processes with the purpose of reducing costs by improving the efficiency of a process. However, there are technological advances that increase the range of monitoring systems and methods available. As the world connects through smartphones and cloud technologies on the Internet, these technologies will continue to grow in what is known as the Internet of Things (IoT - Internet of Things) which is directly related to Industry 4.0. IoT is a new technology on the rise, as it develops rapidly over the years. Our control system is composed of software and hardware that allow us to control, monitor and acquire greenhouse data locally, collecting and processing data in real time. But by deploying the IoT to this system, it will support direct interaction with smart devices and a web server that records the data.

In this age, IoT is revolutionizing the industry by offering more standardization and openness. IoT also provides scalability, interoperability, and enhanced security. Basically, IoT is used to increase overall productivity by integrating smart maintenance. In addition, thanks to them it is possible to optimize the user's reaction time to the control of their monitored system. Our system focuses on monitoring and control, while IoT focuses on analyzing the data of the system (prototype) to improve its productivity and impact the process. Industry 4.0 determines an era in which the automation of emerging trends and the exchange of data in manufacturing technologies are enabling a change in systems, traditionally implemented to one with IoT. With IoT, cloud computing and Industry 4.0 determine an era that will change the dynamics of the entire automation industry.

Theoretical framework

Internet of Things

Today, innovations in the area of electronics, computing, information and communication technologies, have made possible the exponential growth of the processing capacity of computer systems accompanied by the miniaturization of microprocessors, which are used for information processing.

The expansion of these advances throughout the world has led to a significant reduction in the manufacturing and marketing costs of such hardware over the years, facilitating access to them.

These advances have allowed a continuous expansion of the Internet and the creation of new technologies, services and platforms, which has allowed the development of the phenomenon known as IoT, which is nothing more than the evolution of the Internet from a network of interconnected computers to a network of interconnected objects. According to Andrés (2018) "IoT has been called by qualified voices to be one of the agents of the fourth industrial revolution". This so-called fourth industrial revolution is also known as Industry 4.0, which is driven by the internet of things accompanied by artificial intelligence, robotics, 3D printing, nanotechnology, biotechnology and materials science.

It is a fact that IoT for a few years has been a reality present in the daily lives of many people, which has managed to automate many of the household appliances at home, such as televisions, refrigerators, automatic lights and curtains, and voice assistants that we can use to control with simple voice commands all these devices interconnected to the same network. Rose (2015).

Greenhouses with internet of things

Currently we can find a variety of resources from the Internet of Things with applications in the area of agriculture such as greenhouses. Normal greenhouses work by creating controlled weather environments to facilitate the growth of certain crops. In general, in these greenhouses, the regulation of the microclimate conditions is done manually, so supervision by the farmer is needed. Through the IoT, it is possible to use sensors that read the climatic conditions in which the greenhouse is located and regulate it by means of actuators capable of performing a mechanical task that eliminates the surplus of some climatic variable or, on the contrary, increases this variable depending on the optimal conditions required for cultivation. Thanks to this, it will not be necessary to carry out a manual inspection in the greenhouse, leading to an increase in production and profitability.

As a consequence, IoT allows us to have total knowledge about the conditions of the crop and we will be able to make a much more efficient use of resources, especially water. IoT can facilitate a significant increase in profitability and save more of the natural resources of our planet, and discover the most suitable way to obtain a production with higher quality and in larger proportions.

Communication protocol used for the project

For this project, the MQTT (MQ Telemetry Transport) protocol was used, which is a machine-to-machine (M2M) communication protocol, which is based on the TCP / IP stack as the basis for communication.

MQTT is a push messaging protocol, it is used by sending messages from a remote server to a device in a publisher / subscriber (pub-sub) way. The author Llamas (2019) explains that in this type of infrastructure, clients connect to a central server called Broker.

This protocol uses a filter for the messages that are sent to each client, for this they are hierarchically organized into topics, which allows any client to publish a message on a specific topic and another client can subscribe to this topic, the broker will get the subscribed messages to the new customer.

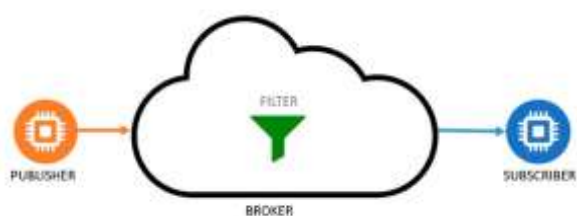


Figure 1 Process to filter the messages sent to each client from the MQTT server
Source: Llamas (2019)

This protocol has many advantages when used as an M2M communication system. According to the Author Camarillo (2020), the advantages of the pub / sub method are, among others, asynchronism, scalability and decoupling between clients. In addition, this protocol provides a series of characteristics that have made it stand out from others. The main one is its simplicity and lightness of processing. This makes it ideal for applications where low processing power devices are used.

As less processing is required, the processor is less demanding, which translates into lower power consumption. This is essential for devices that work all the time and especially in battery-powered devices. Another advantage of the lightness of the protocol is that it requires a very short bandwidth, which is very useful in wireless networks, or low speed connections.

Microcontroller and IoT module used

The fundamental hardware applied to this project is the ESP32 module, which is ideal since it incorporates in a single device a WI-FI module and a microcontroller with digital and analog inputs and digital outputs. So, it is possible to send and receive information from the Internet, read analog sensors and activate the actuators used in the greenhouse.



Figure 2 ESP32 module used for the project
Source: www.espressif.com (2021)

The ESP32 module is manufactured by Espressif Systems, and specifically, it is a series of low-cost and low-power microcontrollers, with Wi-Fi connectivity and dual-mode Bluetooth integrated all on a single board.

Characteristics	of esp32
Microprocessor	Xtensa Dual-Core 32-bit LX6, 600 DMIPS
Wi-Fi	(802.11 b / g / n) HT40
Bluetooth	Bluetooth 4.2 and BLE
Frequency of operation (typical value)	160 MHz
SRAM	448 KB
Flash	520 KB
GPIO	34
PWM (software)	16 channels
SPI	4
I2C	2
UART	2
ADC	12-bits-resolution

Table 1 Technical characteristics of the ESP32 module

MANZANERO-VAZQUEZ, Daniel Jesus, MANRIQUE-EK, Josué Abraham, CARDOZO-AGUILAR, Guadalupe and DECENA-CHAN, Carlos Alberto. Internet of things applied to agriculture using the ESP32 module in connection with the Ubidots platform. Journal of Technological Prototypes. 2021

Beningo (2020) indicates that the ESP32 module is an all-in-one Wi-Fi / Bluetooth solution, providing not only wireless connectivity, but also a built-in processor with interfaces to connect with various peripherals with input and output pins. The microcontroller actually has two processing cores whose operating frequencies can be independently controlled between 80 MHz and 240 MHz, making it superior to other low-cost microcontrollers.

Sensors used

The sensors used for the greenhouse were obtained from a 3-in-one meter module to measure soil moisture and pH, also to measure the amount of light. This module provides economical and ideal sensors for the greenhouse.



Figure 3 Three types of sensors for humidity, pH and plant luminosity

Source: *appandweb.es* (2019)

The module consists of two electrodes and a photovoltaic cell for measuring the variables, one electrode for humidity and another for pH, while the photovoltaic cell measures the amount of ambient light. The module includes the selector to measure the selected variable. The reading is displayed on a scaled galvanometer. The same galvanometer is used for all three variables, but with different scales. To apply these sensors in the greenhouse and to be able to take readings, the signal had to be digitized. This was possible thanks to the fact that the ESP32 module has ADCs with a resolution of 12 bits, with a voltage range of 0 to 3.3v that have enough range to read the analog signal from the sensors.

The temperature sensor used is the LM35 that provides an analog output, which means that its signal pin provides a voltage directly proportional to the temperature.

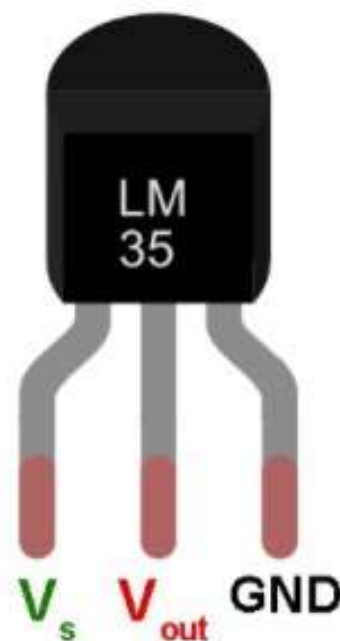


Figure 4 LM35 temperature sensor connections
Source: *Texas Instrument* (2021)

The LM35 is a temperature sensor with very good characteristics at a lower cost than other sensors. Its working range goes from -55°C to 150°C . Its output is analog and linear with a resolution of $10\text{mV} / ^{\circ}\text{C}$. E-Marmolejo (2018).

The sensors generate a voltage directly proportional to the scale of the variable they measure, knowing that the value of the variable can be quantified in the ESP32 with the analog reading of the sensor, then the operating values of the sensors are shown.

Sensor	Mesaurement Scale	Sensor voltage range
Light	0 a 2000 lúmens	0 – 850 mV
Humidity	0 a 100%	0 – 450 mV
pH	3.5 a 8 pH	0 – 600 mV
Temperature	-55°C a 150°C	-550 mV to 1500 mV

Table 2 Analog sensors operating range

The ADCs of the ESP32 module have a resolution of 12 bits, which means that when reading the pin the ESP32 will give us a reading with a range from 0 to 4095. We know that its reading range is from 0 to 3.3V, so it was possible to adjust the value of the ESP32 reading to the scale of the 3 in 1 meter.

Objetives

General:

Design a device for agricultural use that communicates the ESP32 module with the Ubidots platform through the Internet.

Specific:

- Create the MQTT client from the ESP32 module.
- Connect the MQTT client to the Ubidots server.
- Read the sensors connected to the ESP32 module from the Ubidots platform.
- Control the digital outputs of the ESP32 module from the Ubidots platform.

Methodology

Actuator control

The actuators used for the greenhouse were three, a water pump for the irrigation of the crops, fans to dissipate the excess temperature, and UV lamps for the crops.

Actuator	Operating Voltage
Fan	12V
Water Pump	12V
UV lamps	12V

Table 3 actuator operating voltaje

These actuators work at 12V, so it was necessary to design and simulate a relay circuit for the activation of these actuators. The simulation was done in Proteus.

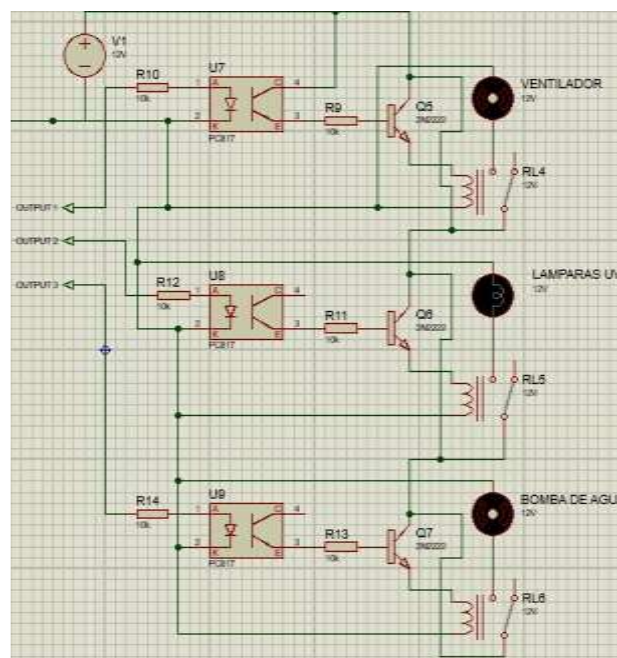


Figure 5 Circuit for activating the relays that control the actuators

Arming the circuit

The circuit for the control of the actuators was assembled following the simulation software diagram, for the part of the sensors they were only connected to ground and the signal was connected directly to the analog inputs of the ESP32 since it does not need any other component to read the signal from the sensors.

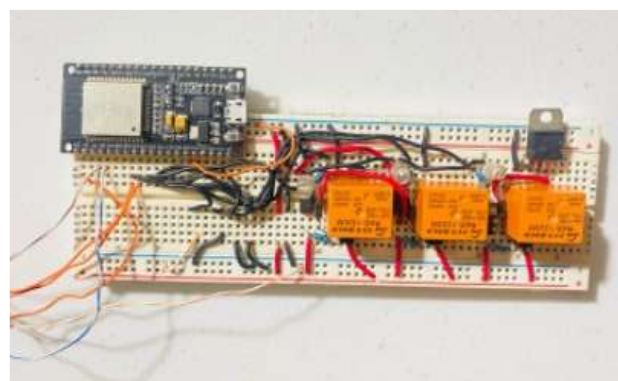


Figure 6 Physical connections of the ESP32 with the control circuit

Programming the ESP32 module

The Arduino IDE was used to program this module since these boards can be programmed in this interface. The library used to apply the MQTT protocol in ESP32 is "PubSubClient.h", which is an open source library that provides a client to perform simple publish / subscribe messages with a server that supports MQTT.

The code for this microcontroller was made in four files, the first contains the credentials, it is called "credentials.h".




```

1
2 #define WIFI_SSID "TF-LINK_873C"
3 #define WIFI_PASS "35458228"
4
5 #define UBI_HOST "industrial.api.ubidots.com"
6 #define UBI_PORT 1883
7 #define UBI_TOKEN "88FF-9LzLtnkbW0htYj2cCjV9dWeQ9gyfQj"
8 #define UBI_CLIENT_ID "40f3h1fc52f6308e000a2daf54"
9
10 #define UBI_CLIENT_LABEL "esp32"
11 #define UBI_RELAY1 "la"
12 #define UBI_RELAY2 "ra"
13 #define UBI_RELAY3 "va"
14
15 #define UBI_HUM "va"
16 #define UBI_LUM "vb"
17 #define UBI_PH "vc"
18 #define UBI_TEMP "vd"

```

Figure 7 File containing the credentials for the MQTT client connection

This file only declares the constants that will have as value the credentials for the MQTT client to work, which are the SSID and password of the Wi-Fi network to which the ESP32 will connect, the server, the port that by default MQTT uses the 1883, the Token, the client ID, and the variables that are necessary for connection and communication with the server. The next file is called "pinout.h", which contains the constants with the value of the analog inputs and the digital outputs with which the ESP32 and the greenhouse will work.



```

1 #define ESP_32
2
3 #define PIN_HUM 32
4 #define PIN_LUZ 33
5 #define PIN_PH 39
6 #define PIN_TEMP 36
7
8 #define PIN_RELAY_1 14
9 #define PIN_RELAY_2 12
10 #define PIN_RELAY_3 13

```

Figure 8 File containing the connection pins of the ESP32 sensors and actuators

The next file is called "mqttclient", this file contains the code that makes the MQTT client work correctly.



```

1
2 #include "credentials.h"
3 #include <WiFi.h>
4 #include <PubSubClient.h>
5 /*
6 Instancia de objeto para el wifi en modo
7 cliente y declaracion e instancia de objeto
8 client de la libreria pubsubclient
9 */
10 WiFiClient ubidots;
11 PubSubClient client(ubidots);

```

Figure 9 File that the MQTT client creates

The file "credentials.h" must be included in this file so that all credential constants can be accessed and thus provide the information to the MQTT server. This code mainly uses the functions of the <PubSubClient.h> library in conjunction with the <WiFi.h> library, using the data from the "credentials.h" file.

Finally, there is the file "INVERNADERO_IOT", this is the main code, the one that the ESP32 will execute in an infinite loop, in this are all the instructions that the ESP32 must perform to read the analog inputs, write the digital outputs and times for communication with the MQTT server.



```

1 #include "credentials.h"
2 #include "pinout.h"
3
4 int conmutador=1,bloquear=0;
5 int adcl_val;
6 unsigned long long1,tiempoMs;
7
8
9 void Relay1Callback(char * pVal) {
10     if (MQTTCLIENTE_GetBool(pVal) == HIGH) {digit
11
12 void Relay2Callback(char * pVal) {
13     if (MQTTCLIENTE_GetBool(pVal) == HIGH) {digit
14
15 void Relay3Callback(char * pVal) {
16     if (MQTTCLIENTE_GetBool(pVal) == HIGH) {digit
17

```

Figure 10 Main file that controls the ESP32

This file must include the documents "credentials.h" and "pinout.h" to be able to access the credentials and the pins that will serve as analog inputs and digital outputs for the greenhouse.

Interface using the MQTT server

The interface was made in Ubidots. This platform offers resources compatible with the MQTT protocol and M2M (machine-to-machine) communication.

MANZANERO-VAZQUEZ, Daniel Jesus, MANRIQUE-EK, Josué Abraham, CARDOZO-AGUILAR, Guadalupe and DECENA-CHAN, Carlos Alberto. Internet of things applied to agriculture using the ESP32 module in connection with the Ubidots platform. Journal of Technological Prototypes. 2021

The first thing that was done on this platform is the creation of a blank device called "ESP32". This device contains all the variables used to control the greenhouse.

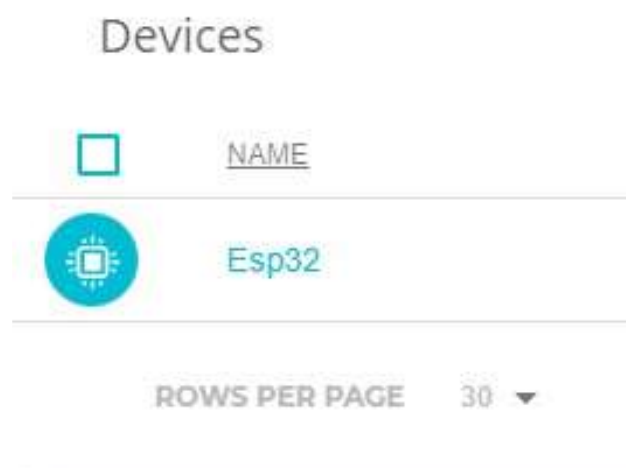


Figure 11 Creation of the device on the Ubidots platform for the interface.

Then the variables were created one by one specifying that they will get their value from the MQTT client.



Figure 12 Creation of variables for sensors and actuators

The variables used are four for the sensors: percentage of soil moisture, ambient light, soil PH, and ambient temperature. Three variables were also used for the actuators, which are fans, UV lamp, and irrigation pump.

The next thing that was done was to elaborate the visual interface in which the information from the sensors and the status of the actuators were represented.



Figure 13 Creation of the visual interface of the sensors and actuators

The elements used were four semicircular indicators to visualize the value of the sensors and three on / off switches for the status of the actuators.

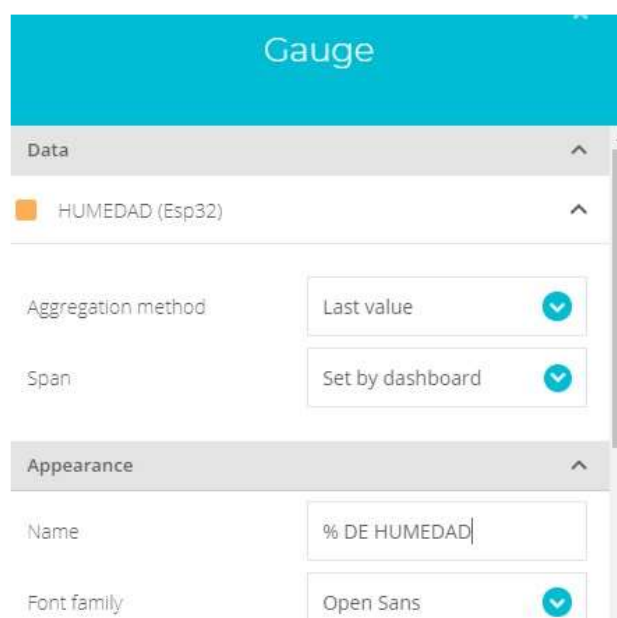


Figure 14 Configuration of the variables in the interface

For each element the corresponding variable was added and in the case of the sensors it was configured so that the color of the indicator changes depending on the value of the variable.

Results

Communication tests between the client and the server

To carry out the tests, only the sensors were connected to the ESP32 module and some indicator LEDs to visually show the operation of the relays that activate the actuators.



Figure 15 Successful communication between the MQTT server and client

It was obtained as a result that the communication between the ESP32 and the server was successful, in which the value of the sensors and indicators were observed working correctly, in addition it was possible to correctly activate each of the actuators from the interface.

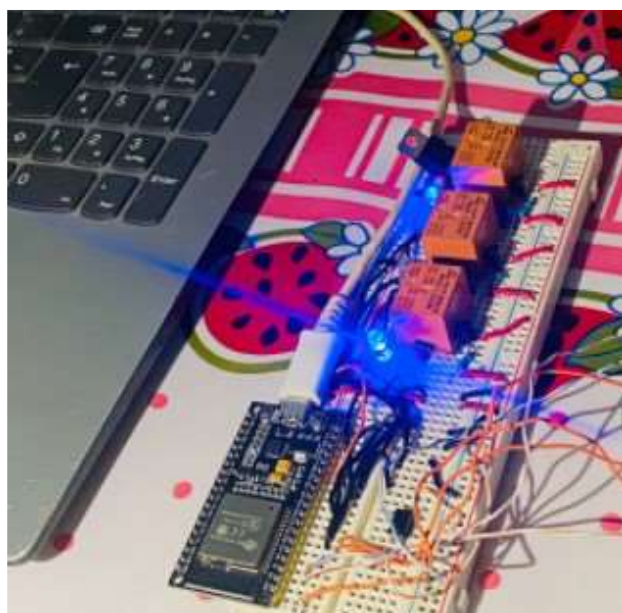


Figure 16 Activation of the relays that control the actuators

In order to observe the operation of the relays, an indicator led was placed on them, which turns on every time the corresponding actuator is activated from the server.



Figure 17 Sunlight and humidity sensor readings

These were the sensor readings of the percentage of soil moisture and the percentage of ambient light.



Figure 18 PH and temperature sensor readings

These were the readings obtained from the PH and temperature sensor, we can observe the correct operation of the ESP32 module and the server.



Figure 19 Record of sensor readings on the Ubidots platform

In addition, from the server it is possible to access the record of the readings for the desired period of time.

Conclusions

With this prototype, it was possible to communicate the ESP32 module with the Ubidots platform correctly using a client with the MQTT protocol to view the greenhouse sensor readings in real time and activate the actuators to control certain environmental variables. Thanks to this, it is concluded that today it is possible to properly apply the internet of things for the control of many productive sectors, this thanks to the great advance of technology in recent years, since there are various platforms for engineering solutions in the area of computing, the internet of things and artificial intelligence with great accessibility and ease of use for the creation of virtual interfaces in which it is possible to communicate with objects from any device with internet access, this in conjunction with The great advance in the development of the new generation microcontrollers, more compact, more powerful, more efficient, and more versatile, it is possible to carry out IoT projects more effectively.

The interface created to control this device is functional, efficient and works in real time, it has a striking design with indicators that change color according to the value of the variable, however, this interface can be improved with a future application creation. For mobile devices that operate directly as a server using the MQTT protocol, this would be more practical, since currently with the interface created in Ubidots it is impractical to access from mobile devices, since the interface is optimized for computers.

References

Am, M. (2015). *Sensores de humedad para plantas*. Jardineria On. <https://www.jardineriaon.com/sensores-de-humedad-para-plantas.html>

Andrés, M. B. (2018). *Internet de las Cosas* (1.^a ed.). Reus Editorial. <https://books.google.nl/books?id=jF-LDwAAQBAJ>

appandweb.es. (2019, 27 diciembre). *Aplicaciones de IoT en el sector de la agricultura*. App&Web. <https://www.appandweb.es/blog/iot-agricultura/>

Beningo, J. (2020, 21 enero). *Cómo seleccionar y usar el módulo ESP32 con Wi-Fi/Bluetooth adecuado para una aplicación de IoT industrial*. Digi-Key ELECTRONICS. <https://www.digikey.com.mx/es/articles/how-to-select-and-use-the-right-ESP32-wi-fi-bluetooth-module>

Camarillo, A. (2021, 29 enero). *¿Qué es MQTT?* 330ohms. <https://blog.330ohms.com/2020/07/12/que-es-mqtt/>

ESPRESSIF. (2021). *ESP32 Wi-Fi & Bluetooth MCU I Espressif Systems*. espressif.com. <https://www.espressif.com/en/products/socs/esp32>

Llamas, L. (2019, 17 abril). *¿Qué es MQTT? Su importancia como protocolo IoT*. Luis Llamas. <https://www.luisllamas.es/que-es-mqtt-su-importancia-como-protocolo-iot/>

Marmolejo, R.-E. (2018, 18 enero). *LM35 - El sensor de temperatura más popular*. HETPRO. <https://hetpro-store.com/TUTORIALES/lm35/>

Rose, K. (2015, octubre). *LA INTERNET DE LAS COSAS— UNA BREVE RESEÑA*. Internet Society. <https://www.internetsociety.org/wp-content/uploads/2017/09/report-InternetOfThings-20160817-es-1.pdf>

TEXAS INSTRUMENT. (2017, diciembre). *LM35 Precision Centigrade Temperature Sensors*. <https://www.ti.com/lit/ds/symlink/lm35.pdf>

ubidots. (2021). *IoT platform | Internet of Things | Ubidots*. ubidots.com. <https://ubidots.com/>