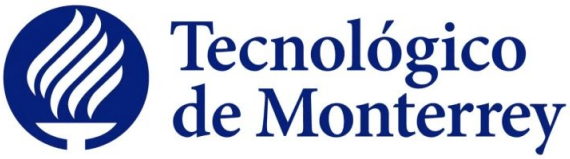


Implementation of the Internet of Things (Group 06)

Challenge Documentation

Teacher: Roberto Treviño and Rafael Dávalos

Jesus A. Trujillo	A00827538
Edith P. Benvenuto	A00828840
Diego A. Torres	A01283874
Luis Gutierrez	A01283825
Gerardo L. Arrambide	A01283504

Monterrey, Nuevo León, December 4, 2020.

IoT Background

The Internet of Things or better known as “Internet of Things” is something that, today, is part of our lives at all times and I honestly cannot imagine a world without this concept. The concept of

connecting any object to the Internet and thereby creating an infinity of new applications is something that is surprising but at the same time not surprising. It seems that it is a concept that has been with us for a long time given its omnipresence but in reality, this is a fairly recent concept.

The term Internet of Things was first given publicly by MIT professor Kevin Ashton at a conference in 2009, and since then this concept has grown exponentially.

But what is the Internet of things?

It refers to the digital interconnection of any object with the Internet, this transforms them into intelligent objects capable of exercising autonomy for solving problems. This can not only have an economic impact on the industry but also a positive impact on the environment.

Here is how we implement everything we have learned about this concept in our project.

Introduction

Throughout the semester, we were learning about digital systems and database analysis and design. The purpose of this challenge will be to demonstrate the knowledge learned in an interconnected computer system between the houses of the different team members to have as a final result, a home automation application that receives information from the multiple sensors that exist in the team.

The first step to achieve this was to model a database that has the necessary attributes to receive the information that was planned to be sent with the sensors of each member. To make the challenge more challenging, not all members had the same sensors connected to the NodeMcu, which makes the database can have more varied information and left a greater challenge for the team in terms of modeling and implementation. .

The second step was to connect each of the necessary components of each team member to be able to send data from their respective NodeMcu through WiFi networks. Likewise, the code that would serve to send the data received from the hardware to the database hosted in the cloud was developed. Developing the code was one of the most challenging steps of the job due to the different connections that were made to successfully send the sensor data.

The purpose of this document is to further explain each of the aforementioned steps, from the perspective of each student, in order to check the work that was done during the class.

Description of the Challenge Elements

- NodeMCU Microcontroller

  The NodeMCU microcontroller is based on Arduino and supports its programming language. This has an ESP8266 module that makes the connection to the Internet via WiFi. This Internet connection allows us to send the data received by the sensors to the database.

  We plan to use this Microcontroller to receive the data that is sent by the sensors and send it to the database. All team members must have this Microcontroller since the purpose of the challenge is for everyone to send the information received by their sensors and this is integrated with the information of others.

- Breadboard

  A Breadboard is a board which allows the passage of current to create different circuits. This has several holes in which the different electronic components and cables are inserted to make the connection between them. It is essential to know how it works and how the current passes through it for the creation of any circuit. We must use a breadboard to be able to make the circuits that contain the sensors and the NodeMCU. As all the members of the team decided to work with the physical circuit, we must all know how this board works. In it we will connect the NodeMCU, the different sensors that each person implements and different LEDs that indicate certain conditions. We will also use cables and resistors to pass current from the nodeMCU to the sensors and to the positive and negative current.

- Humidity and Temperature Sensor

  The temperature and humidity sensor that was implemented was the DHT11. It integrates a capacitive humidity sensor and a thermistor to measure the air. Displays the data received by a digital signal on the data pin.

- Ultrasonic Sensor

  The sensor model used is the HC-SR04. This is a sensor that measures distance using ultrasound in a range from 2 to 450cm. It has two transducers, an emitter and a receiver. The emitter emits 8 ultrasound pulses (40 KHz) when it receives the command from the TRIG pin, the sound waves travel through the air until they bounce when meeting an object. This shock is detected by the receiver and the ECHO pin sends the signal back.

- Sound Sensor

  The sound sensor model used is the KY-037. This model allows detecting any type of sound above the range set by the included potentiometer. It is composed of two outputs, an analog output and another digital signal that is activated when the sound intensity has reached a previously configured limit.

- Magnetic Sensor

  This sensor is a magnetic switch that sends a signal, when the separation between the magnets is 2 cm or more, it is mainly used in both door and window entrances. The model used is . ALA-005.

- MQTT broker

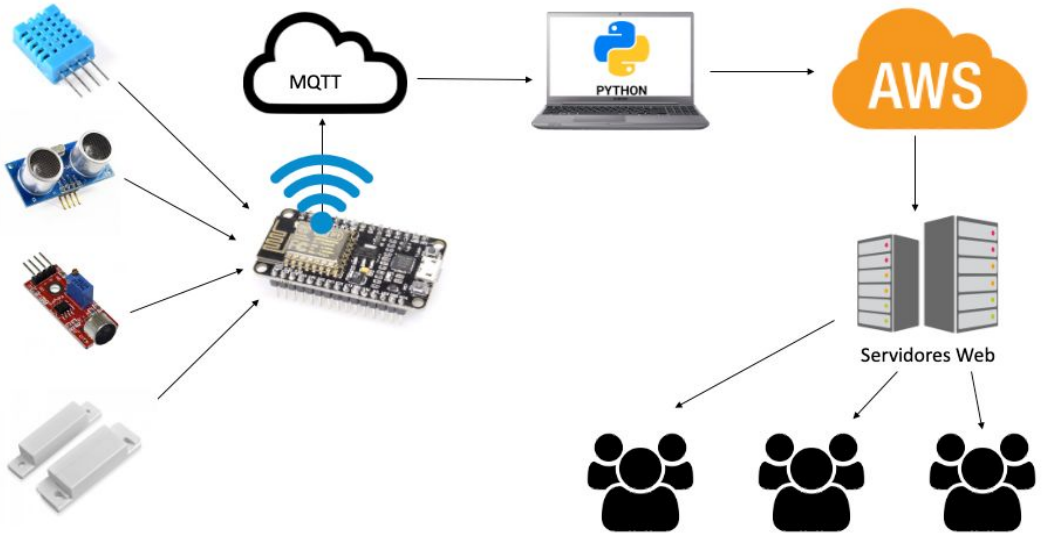
  MQTT is a network protocol used for machine-to-machine communication, generally running over TCP / IP as a basis. It is a push messaging service with a pub / sub pattern. The MQTT architecture follows the star topology where it has a central node, the server also known as the MQTT broker is said central server where clients connect, it is in charge of managing the network and sending messages keeping the channel active. It is important to mention that the broker used by the team

  will be the HiveMQ, with host in the address http://www.hivemq.com/demos/websocket-client/

- Database

  It is an organized collection of data and structured information, which are stored for later use, databases are usually modeled in tables (with rows and columns) to have a data query with greater efficiency. It has been decided to incorporate the use of a hosted relational database on AWS into the design.

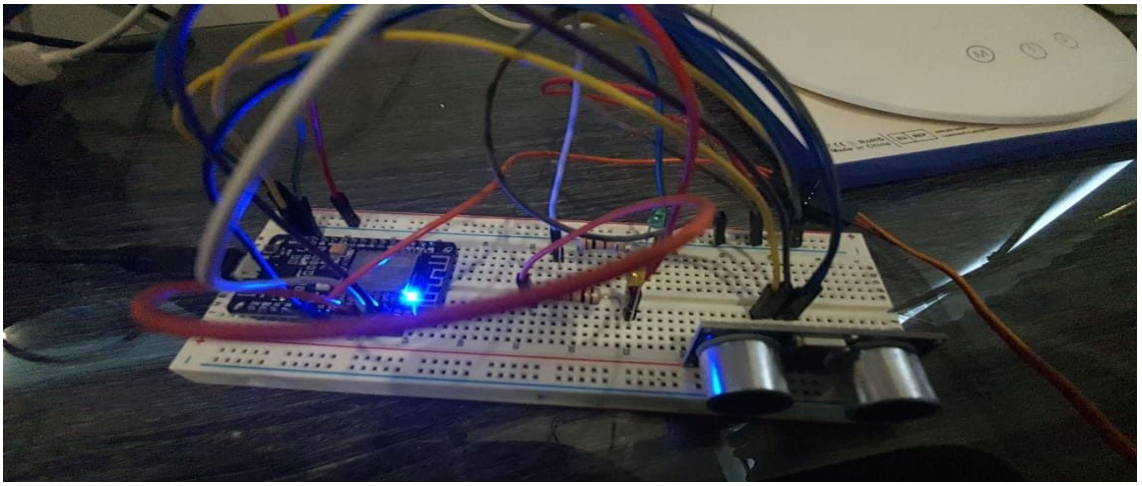
Design of Interconnected Elements



Development

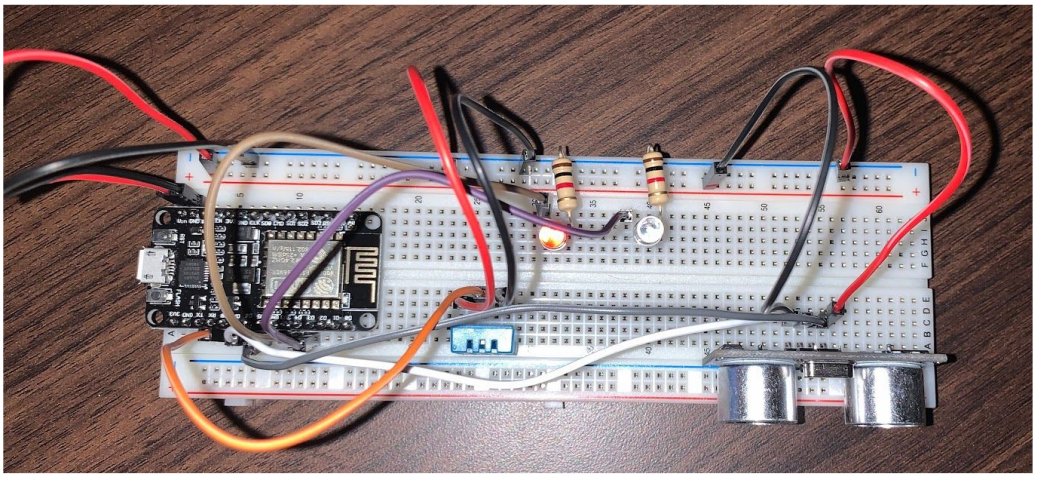
Since no one on the team was familiar with the Proteus platform, we all opted to buy the components and build the circuit physically. So the first step was to get the components and put the circuit together. This was not so easy for some as there were several problems connecting these circuits to the computer and starting to manipulate them with code.

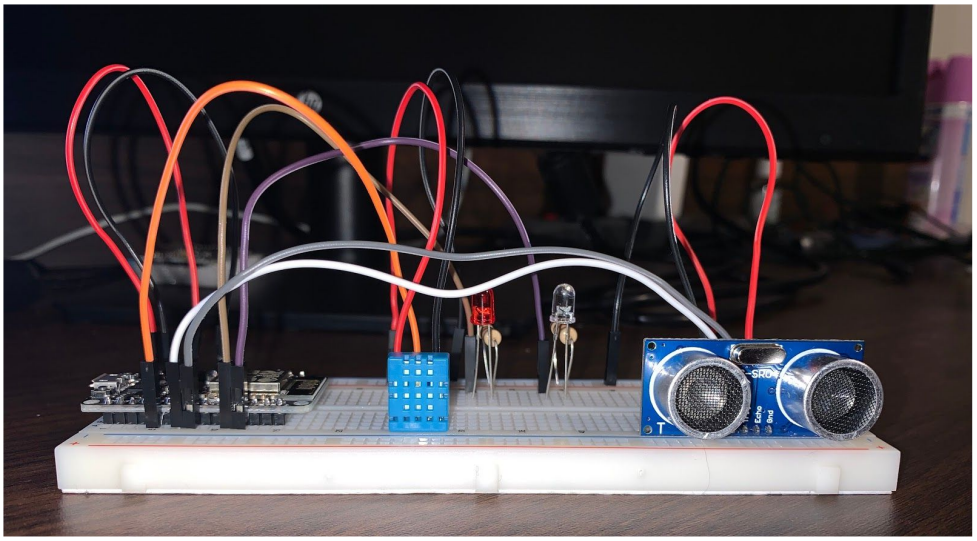
The first circuit to be successful and to connect correctly was that of Jesús Trujillo. There was no serious problem with this circuit, although in order to expand the functionality of the actuators, a servomotor and a 5-volt relay were added, this in order to turn on a desk lamp and a fan, respectively.



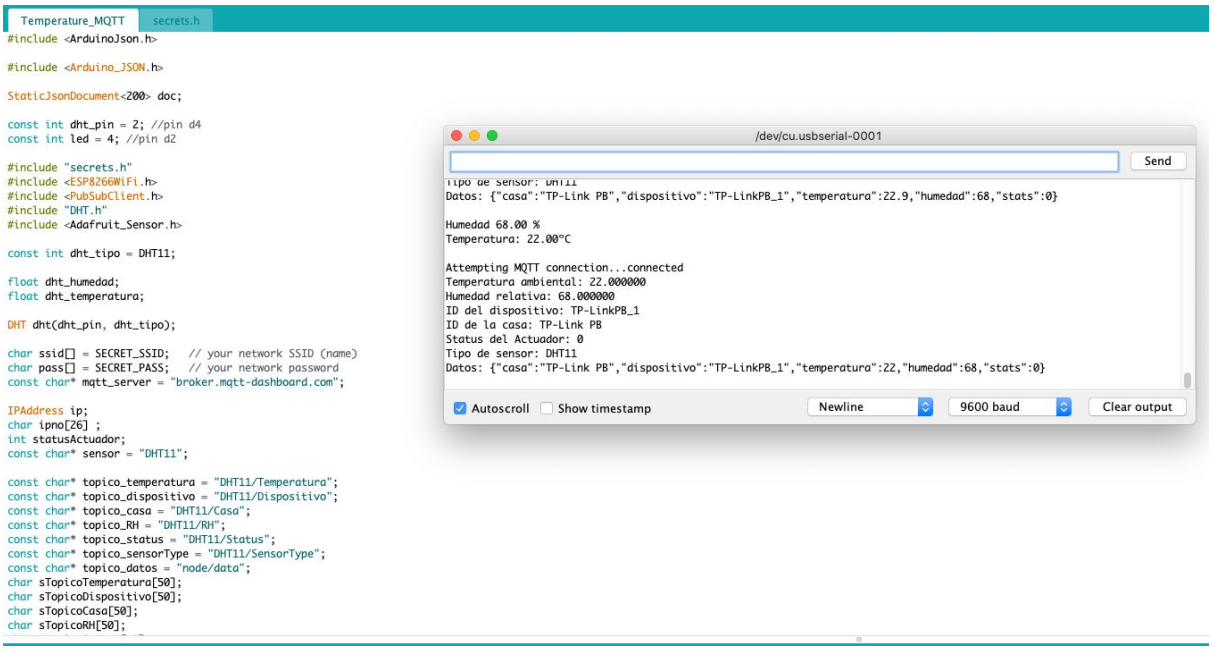
There were several technical problems with Gerardo Arrambide's circuit. In the first part, it is believed that one of the sensors used in the first circuit he made was defective, which caused a short. In addition to this, I had no practice with this type of electrical circuit since this was the first time I had worked with one. As a result, the temperature sensor burned out and the NodeMCU stopped working.

After getting new components to replace the faulty ones, the circuit was reassembled from zero and finally it was functional.





and with this new circuit it was possible to send the temperature and humidity data to the database through MQTT.



All team members used the NodeMCU ESP8266 development board, with the exception of Edith Benvenuto, since the NodeMCU's WiFi module was the ESP32, the most notable difference between them is that the ESP 32 is a dual-core processor and the ESP8266 is one. However, the ESP32 usually has a very common failure, that when uploading a program to the board, it takes too long and shows the error “Failed to connect to ESP32: Time out waiting for packet header”, this happens when the time of waiting is exceeded, so a radial electrolytic capacitor was soldered, from 10 micro Farads to 63 Volts, this served to increase the electrical power and to modulate the signal in the power supply. After soldering the electrolytic capacitor (see picture 1), the error disappeared and the NodeMCU ESP32 had no further problems. However, in case the ESP32 has another problem, the NodeMCU development board was obtained with ESP8266 (image 2).

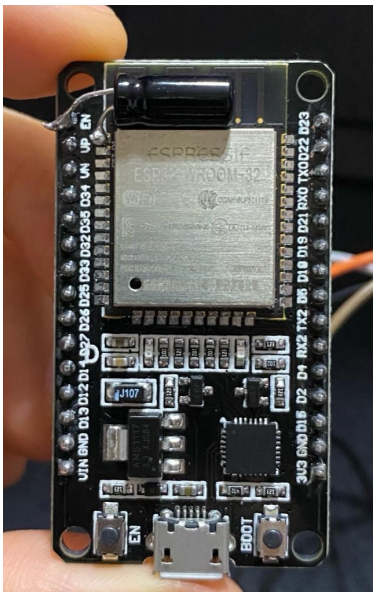


Image 1 NodeMCU ESP32 with electrolytic capacitor

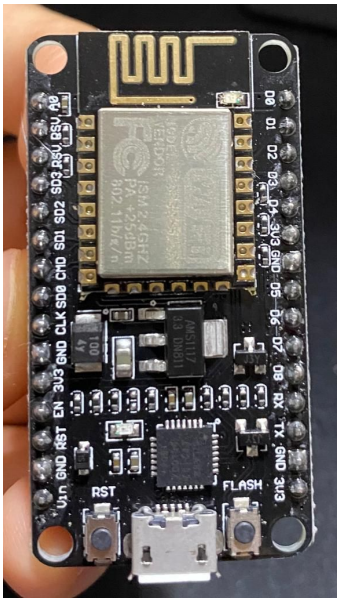
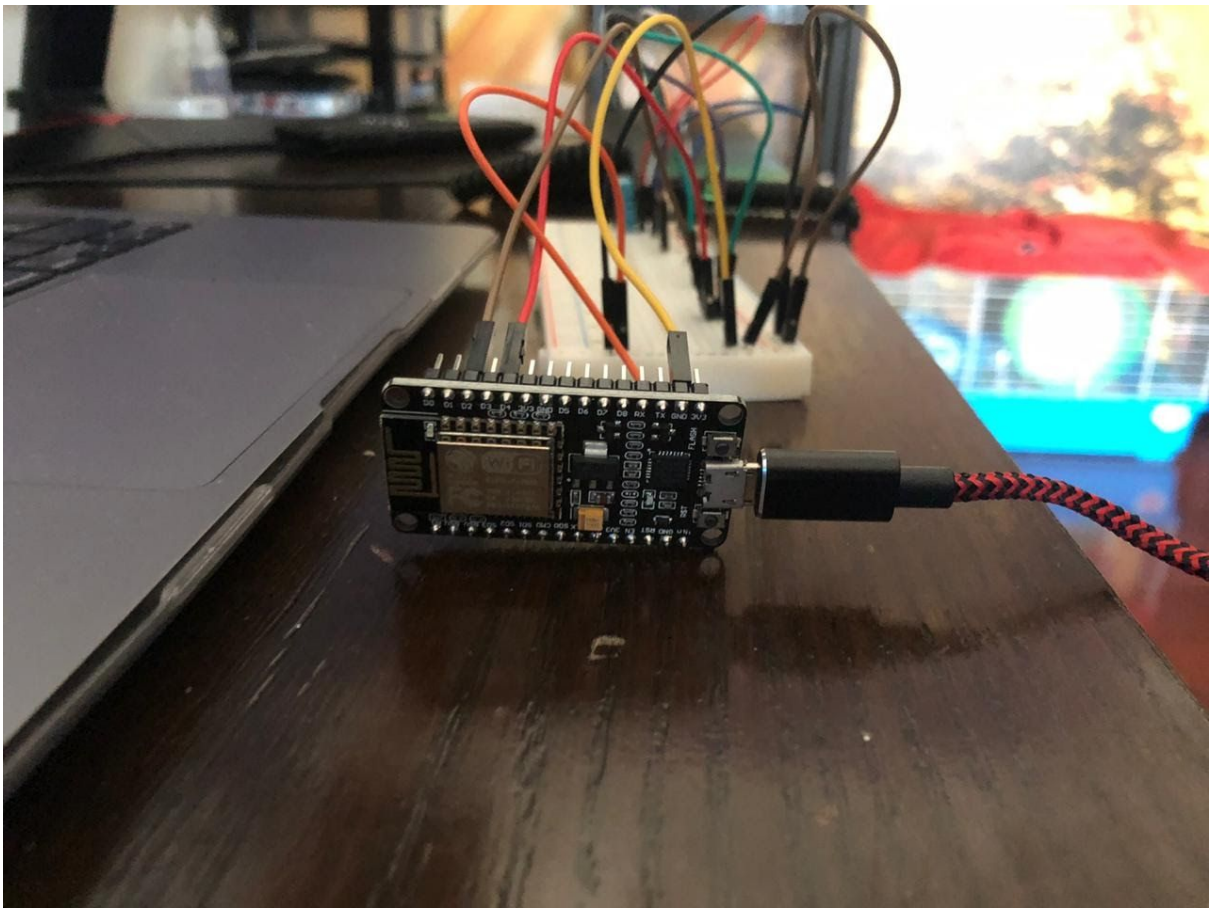
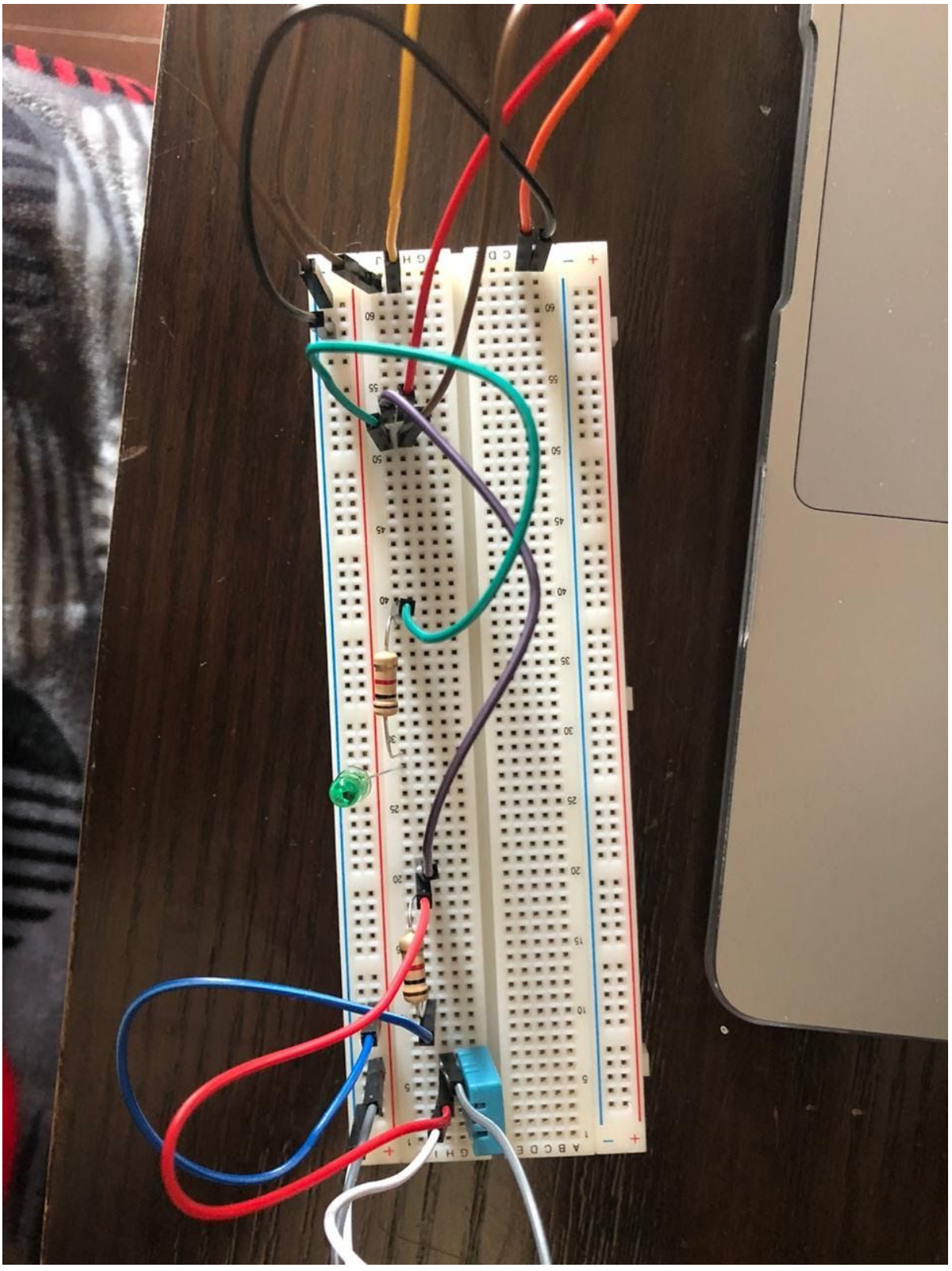


Image 2 NodeMCU ESP8266

The next circuit to be tested was that of Alex Torres, at first there were problems since it was believed that the project would not work on Alex's computer because of the way he was taking the signal through the USB. This turned out to be a problem with how the breadboard was wired rather than a problem with the computer, which brought optimism to the team.



After this, Alex ran into a problem similar to Edith's, where the packet header is not being received. This problem could be solved as Alex realized that he was connecting to the wrong port. Being able to do this, a few tests were done to make sure that the sensors and other components were working correctly, there was no problem on this part.



The only thing missing was the final step, connecting to Wi-Fi to be able to send the information to the database. Multiple tests were carried out to try and make it work but for some reason, the NodeMcu was not connecting to Alex's home network, or to his cell phone network. We could not understand why this happened then

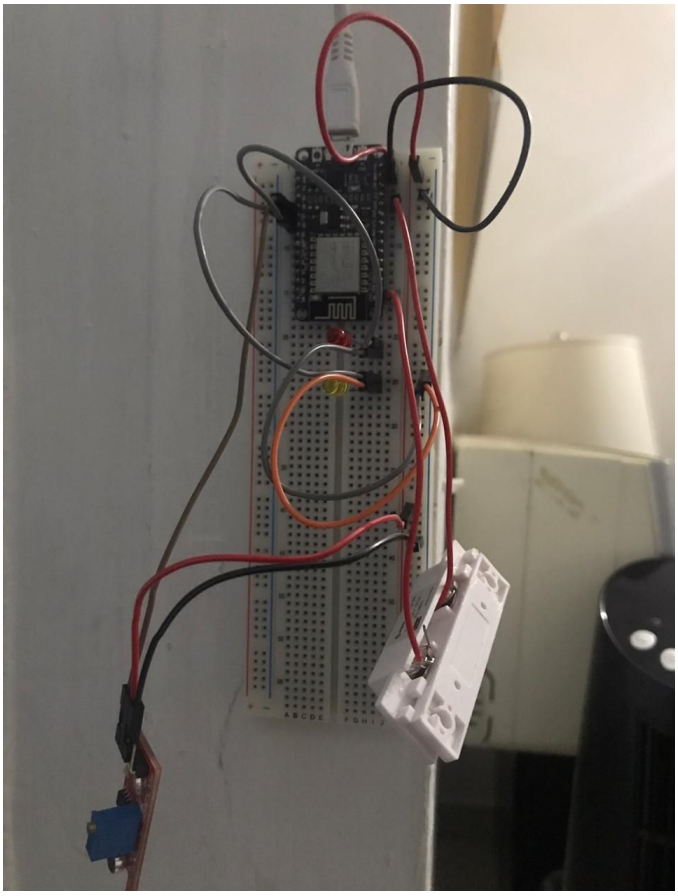
We decided to continue with the project, however a lot was learned during this stage and we do not consider it a failure since some team members did not know about hardware before this block and they still managed to connect the board and receive information.

On the other hand, Luis was in charge of connecting two extra sensors to the project, one acoustic and the other magnetic for doors. The process carried out is described below.

For the acoustic sensor, its digital information output was used, which is activated when the data detected by the sensor exceeds the voltage of the potentiometer included in it. Following this, only the power supply part was connected to the positive line connected to the 5V pin of the board and the ground part to the negative line connected to earth.

On the other hand, the magnetic door sensor implements an analog output which receives the current transmission data through the circuit. One end connects to the NodeMCU's analog read pin and the other to the board's 3.3V power supply. Now when the part of the magnet comes into contact with this connected part, it closes the circuit and lets the current flow to the analog pin. Thus, when the signal received in the analog reading is greater than 0, it means that both parts of the sensor are joined, indicating that the door is closed.

Finally, two light-emitting diodes were added, which function as actuators and are activated when the signals from the sensors are received.



Arduino script development

In order to develop an embedded system like the one requested by the challenge requirements, it would be necessary to write an Arduino code that manages to control the generic NodeMCU ESP8266 and ESP32 board. In this case, the NodeMCU must be able to connect with an MQTT messaging protocol broker and publish data.

The team was given the task of investigating the format used for writing topics in MQTT, resulting in that the best thing would be to create a topic by data, or to create a topic of topics, which would be something like this: "node / #" . After conducting joint tests with the Python code, it was decided to change the format of the topics to a simpler one ("node / data"), where the Python code would then be in charge of differentiating into which table the data will be inserted in AWS.

Usually, an arduino program uses two functions as a resource, the setup function that only executes once, and the loop function that executes as long as the board is powered. In this case the program would be much more complicated since it incorporates the use of MQTT and the connection to the network to publish the topic.

Various functions will be included in the arduino program, one of them will be in charge of connecting the NodeMCU to the network, so the ESP8266WiFi library should also be downloaded; It is important to mention that when the NodeMCU manages to connect to the network, it will show a message on the serial monitor with the network and the IP address assigned by the router. Subsequently, the function would be called to connect the device to the MQTT broker, where this cycle will run until the device manages to connect to the broker.

Having already gone through these two functions, the arduino would begin to infinitely execute the loop function, which inside it calls other functions. The first function to be executed would be to measure, there the numerical data to be sent will be established, and also the string in JSON format that the node / data topic will send to MQTT would be created and formed. In the end, only the data is loaded with snprintf to the topic buffers.

Subsequently, the connect mqtt function will be called, which the first thing that is verified is that the device is connected to the broker, in case this is false, the connect function would be called. Having checked this, it is verified that there is no data splice with a time counting algorithm, then the values that the JSON document will include are printed and the publish method of the MQTT library for arduino is called. Since the topic is published in MQTT, the program will have a condition in which it will wait 10 minutes to execute the loop function again.

The following is an address to the repository on GitHub of the programs developed for the challenge:

- https://github.com/ElTrujo01/IoT_TC1004B

IoT Connection and Application

Among the main requirements requested by the teaching staff for the development of this project would be to connect each board used by the team to an MQTT protocol broker service, which in turn must be connected to a server on AWS, although if necessary It should be mentioned that this data must first be processed through the implementation of an algorithm in python, and the proper use of a couple of libraries related to the publication and subscription of the data.

With the support of the temporary files of type JSON generated by the NodeMCU and the designated programming in Python, the data found inside this temporary document could be processed. It is important to mention that this can also be done due to the structure with which the topic used is defined, since previously, when multiple topics were used, the Python program and the insertion of the data failed.

The Python program itself is straightforward, and it is quite convenient that it is divided and handled with function calls. It is worth mentioning that part of the code was found inside a GitHub repository, however, this did not have any type of documentation, so the team undertook the task of investigating the operation of each of the methods used .

In order for Python to be able to connect in some way to the MQTT broker, it will be necessary to use an external library, in this case paho-mqtt was used, which had to be added to the library by executing the following command at the Windows Command Prompt : pip install paho-mqtt. In order to use this library and its methods, it would be necessary to create an object of the type paho-mqtt, which was designated to be called "mqtt". Later, in the

"Main" of the program would be called a method called client, to which variables declared globally to connect to the HiveMQ broker. Then the on_connect function would be called, which would connect Python with the database, having achieved this, it is waited for a topic to arrive at MQTT.

In order to perform a search or insert data into the database in AWS, the pymysql library had to be incorporated into the design, which was again installed when running the command pip install PyMySQL at the Windows Command Prompt. As before, an object of type pymysql was created, without definition, which will mainly be used to insert the processed data into the database.

Every time the on_connect function detects the arrival of a series of data from the broker, it must be detected on which of the 4 tables the data set will be inserted, this is done with the support of some specific variables used precisely to differentiate each set. A great example of this is that within the data set sent by the NodeMCU to the broker is the variable "temperature", while for the ultrasound sensor the specific variable would be "distance". When it is possible to differentiate which data set is sent, by means of a series of ifs, the specific function that will insert the data in the correct table is called, with the support of the execute method of the pymysql library. When the signal is received that the insertion is successful,

It is extremely important to mention that this program must be running 24 hours a day on a computer with an internet connection, that is why an additional computing device was installed for the staff, which is accessed every time it is necessary to make changes to the program. Python by using Windows Remote Desktop functions.

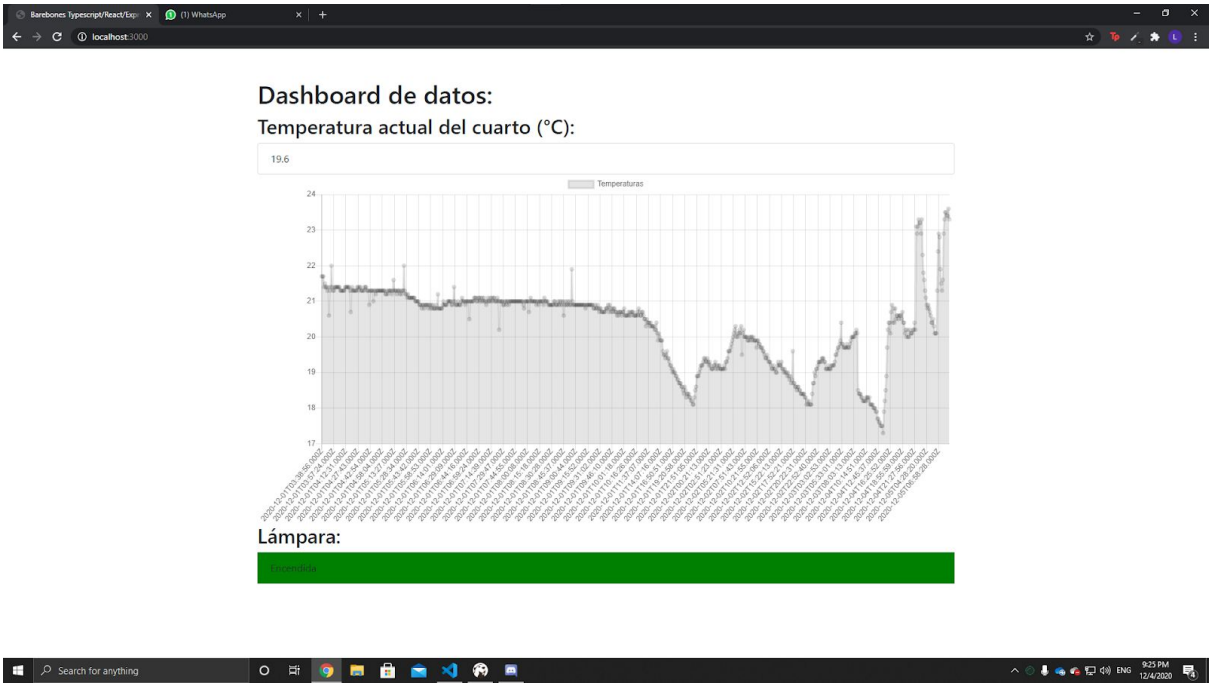
```
PS C:\Users\jtruj\Documents\Arduino> python -u "c:\Users\jtruj\Documents\Arduino\MQTT_MySQL_Publisher.py"
MQTT Client Connected
MySQL Client Connected

Transmission received from DHT11  Alondra2_2  at 2020-12-04 20:08:31.324009
data logged

Transmission received from ultrasound  Alondra2_1  at 2020-12-04 20:09:08.117932
data logged

Transmission received from DHT11  TP-LinkPB_1  at 2020-12-04 20:10:01.667500
data logged

Transmission received from DHT11  Alondra2_2  at 2020-12-04 20:18:31.822941
data logged

Transmission received from ultrasound  Alondra2_1  at 2020-12-04 20:19:08.460277
data logged
```

At the following address you can find the developed Python program, which is saved in the challenge's GitHub repository:

- https://github.com/ElTrujo01/IoT_TC1004B/blob/main/MQTT_MySQL_P ublisher.py

Finally, a data visualization system was developed through a Web application developed with the React Native framework. For this, a connection was made to the database hosted in Amazon Web Services to obtain the necessary data through queries executed from the application.

After this, using TypeScript components, a visualization component was created where the data of the current temperature, the status of the lamp and a graph of the last 25 temperatures recorded were displayed. The prototype of this application is shown below, attaching the code examples of the queries made.




```
export const last = async() =>{
    return new Promise((resolve,reject)=>{
        Connection.query('SELECT * FROM TempData ORDER BY DateTempStatus DESC LIMIT 1;', (err,results)=>{
            if(err){
                return reject(err);
            }
            resolve(results);
        });
    });
}

export const all = async() =>{
    return new Promise((resolve,reject)=>{
        Connection.query('SELECT * FROM TempData WHERE ID_HOUSE = "Alondra2" LIMIT 25', (err,results)=>{
            if(err){
                return reject(err);
            }
            resolve(results);
        });
    });
}
```

```
export const last = async()=>{
    return new Promise((resolve,reject)=>{
        Connection.query('SELECT * FROM UltrasoundData ORDER BY DateDistStatus DESC LIMIT 1;', (err,results)=>{
            if(err){
                return reject(err);
            }
            resolve(results);
        });
    });
}
```

General conclusion

Without a doubt, the realization of this project was not entirely easy. Unfortunately not all team members were familiar with Arduinos and had never worked with them. Fortunately, we learned to recognize the strengths of each member to find the functionality that each one could have. As a result, we worked efficiently as a team and were able to carry the project forward.

This IoT topic was really very interesting to explore and we are sure that this will not be the last time we apply concepts on this topic. In addition, learning to send information from different devices to a database is essential for our career. We consider that this project worked as a good introduction to this topic that will be very important in our career.

As a team, we would like this kind of projects to continue to be applied in the following semesters since they help a lot to understand how the integration of all topics occurs, something that is more difficult to see when we only work with theory.

Individual Conclusions

Gerardo Arrambide

In general, I liked this challenge since I managed to implement everything I learned in the different modules of the block. Without a doubt, the most complicated part of the challenge was the circuit part and its connection with the database. I had never worked with Arduino or anything like it so this was a completely new experience for me. Fortunately, several members of my team were already familiar with this issue and helped me with all the difficulties that arose. At the beginning I struggled a lot with the connection of the circuit for this same reason and even with the help of my team, I ended up burning the whole circuit. For this reason I consider that this project

it would have been much more effective and practical if it had been done in person in the classroom.

The databases part did not seem very complicated to me since it is a topic that I had already worked with before and understood quite well. On the other hand, the connection between NodeMCU and the database was a challenge. I had to support a lot from my colleagues and external investigations to carry this part forward and fortunately in the end everything went well.

Finally, I would like to thank the two professors of this block since without their help this project would not have been able to succeed.

Edith benvenuto

Personally, this final project really challenged me to do my best, since I could see the application of most of the concepts seen in class in our daily lives and how they could really facilitate many really simple tasks and reduce expenses in certain areas such as electricity, in the case of our project.

The most difficult thing for me was working as a team remotely, since when any of us had a problem, whether it was with the digital connections or with the connections to the database, it was really difficult to find such a problem among all of us.

As I mentioned in the development section, I had certain problems in the connections with the NodeMCU and it was a bit complicated for me to know the differences between the two NodeMCUs with different wifi module that I used, ESP32 and ESP8266, find how to connect it and translate the input of the pins to the Arduino IDE. Also understanding how MQTT works and how it connected to the database was difficult for me.

The challenge was really interesting and applicable to our day to day, despite the fact that I had some complications and difficulties, there was always a lot of support, both from the teachers of both modules, as well as from my teammates, they were all always willing to resolve any questions that may arise.

Diego Torres

Personally, this project seemed very fun and I learned a lot, I would have liked to do it in person but I still had a good experience.

The proposed project challenged me to understand and apply new knowledge that I was learning as I was working. I had never handled any kind of hardware or Arduino before but thanks to the help of my classmates and teachers I was able to understand him.

I had some difficulties at the time of carrying out this project, at the beginning they were failures that could be solved in a very simple way but as I did not have knowledge about the subject, it took me a while to find out what was happening. In fact, I was not able to complete the entire project as my NodeMcu was never able to connect to the internet, however the experience I gained will help me a lot in future projects and I think it was a good start.

In general, I found this challenge extremely interesting and I would like that from now on they were all similar, where different concepts are taught and we can apply them in a project since personally I think that it is the best way to learn.

Jesus A. Trujillo

Personally, I must say that it was already something related to the world of the internet of things, since my father integrates more and more devices.

of this kind at home. It is very common to see how different devices are added, some range from Wi-Fi cameras to outlets controlled by a mobile application, I have even seen how the air conditioners in the social areas turn on themselves according to the temperature of the room.

When I saw this subject in my curriculum, I sincerely believed that it would be a little more different, in fact I believed that it would be a little more focused on the software part than hardware and embedded systems since I personally want to know the kubernetes part. The truth is that at first I was a bit discouraged when we started to see digital modules, although little time passed by when the class began to form my interest.

Personally, I can say that I had no major complications when creating the Arduino circuit on the breadboard, the worst thing that happened was that I connected the ultrasound sensor with inverted polarity, therefore it did not send the data correctly. Actually what I consider a bit more complicated, or rather what took more time was to get the arduino mqtt topic to be sent in JSON format, this in order to use parse either in python or with node-red .

Thanks to the development of the challenge, I was able to apply numerous knowledge and ideas formulated in class, many of which were quite abstract, so I really finished understanding the concept by executing some action related to the challenge itself. I greatly appreciate the way in which the challenge was developed, since the team in general was able to make an effort in this way and try to get the best out of each member.

Luis Eduardo Gutierrez

I found this subject to be very interesting, I think it has been one of the subjects in which I have learned the most throughout my career and I think I was able to apply a lot of knowledge. I think the solution we came up with as a team was very complete and met the standards.

Personally, I was mainly in charge of the connection of the acoustic sensors and the door, as well as the development of the web application to visualize the data. I believe that the joint contribution of each of the team members was what made us able to fulfill the requirements of this job.

On the other hand, I feel that this is a first step towards finding new projects within the IoT area, which personally draws my attention and I feel that through the different concepts learned in class we can apply new solutions to this area in a future.

In general, this challenge seemed very interesting to me and I think I was able to learn a lot, and one of the aspects that added the most value was the fact of being able to use the acquired knowledge in a practical way to implement it in a real solution.

References

Oracle. (2018). What is a database? Recovered in December 01, 2020, of . https://www.oracle.com/mx/database/what-is-database/

Naylamp Mechatronics. (2020). Temperature and relative humidity sensor DHT11. Recovered in December 01, 2020, of https://www.naylampmechatronics.com/sensors-temperatura-y-humida d / 57-sensor-de-temperature-y-humidity-relative-dht11.html

IoT code. (2019). WHAT IS THE NODEMCU MICROCONTROLLER? Recovered in December 01, 2020, of https://www.codigoiot.com/que-es-el-microcontrolador-nodemcu/#:~:text= A% 20microcontroller% 20NodeMCU% 20es% 20un, connect% 20a% 20Internet% 20via% 20WiFi . .

Naylamp Mechatronics. (2020). Ultrasound Sensor HC-SR04. December Recovered in 01, 2020, of https://www.naylampmechatronics.com/proximity-sensors/10-sensor-u ltrasound-hc-sr04.html? search_query = sensor + ultrasonic

Naylamp Mechatronics. (2020, December 01). KY-037 Sound Sensor Module. Recovered in December 01, 2020, of https://cdmxelectronica.com/producto/modulo-ky-037-sensor-de-sonido/

Telliottosceola. (2020, January 28). Mosquitto Subscriber MySQL Publisher. GitHub. Retrieved on November 15, 2020 from https://github.com/ncd-io/Mosquitto_Subscriber_MySQL_Publisher