Universitat de Lleida

PREMIS A TREBALLS DE RECERCA DE LA UdL
per a l'estudiantat de batxillerat i cicles formatius de grau superior

Development of a gas monitoring system based on the internet of things

Judith Shan Masip Capdevila

Centre: Col·legi Episcopal – Mare de Déu de l'Acadèmia (Lleida)

Tutor/a: Angel Murillo Esteban

Data: juny, 2021

TREBALL DE RECERCA

DEVELOPMENT OF A GAS MONITORING SYSTEM BASED ON THE INTERNET OF THINGS

2n de Batxillerat Tecnològic

“Everything that can be automated will be automated”

Robert Cannon

INDEX

Abstract…… 5
INTRODUCTION…… 6
1. IOT…… 8
1.1. What is exactly the Internet of Things, IoT? …… 8
1.2. History of the IoT …… 9
1.3. What are the roots of the tree we know as IoT? …… 11
2. COMPONENTS OF THE IOT …… 14
3. TYPES OF IOT …… 16
3.1. Based on the abilities …… 16
3.2. Based on the level of intelligence …… 16
3.3. Based on the system complexity …… 17
4. AFVANTAGES AND DISADVANTAGES OF IOT …… 21
5. THE ENVIRONMENT AND IOT…… 23
5.1. Main pollutant gases …… 24
5.1.1. Final gasses study choice …… 26
5.2. Gas sensors …… 27
5.2.1. Types of sensors …… 27
5.2.2. Final sensors choice …… 32
5.3. Development board …… 33
5.3.1. Types of development boards …… 34
5.3.2. Final development board choice …… 39
5.4. Platforms …… 40
5.4.1. Types of platforms …… 40
5.4.2. Final platform choice …… 43
6. PRACTICAL PART…… 44
6.1. Requirements…… 45
6.2. Setting up the sensors and power supply…… 46
6.3. Connection of ESP32 to Wi-Fi and ThingSpeak…… 56
6.4. Final assembly…… 61
6.5. Setting up an alarm to Twitter through ThingSpeak…… 68
6.6. Experimentation…… 70
6.7. Video…… 72
7. CONCLUSION…… 73
Webography …… 74
Table of figures…… 78

ABSTRACT

The world is constantly evolving and so is technology. One of the newest technologies is being implemented more and more into everyday life is the Internet of Things (IoT), which connects all the desired objects to the network.

In this research project, an IoT based real-time monitoring system is created from scratch. The system detects gases that in their excess, can cause health problems. In addition, an alert system is established though a public message to the well-known social platform Twitter.

Therefore, during the process of research, assembly and programming, knowledge about the basic elements of electronics, embedded C Language and the internal functioning of a system based on the Internet of Things has been acquired.

INTRODUCTION

The Internet is used daily and almost constantly to a great part of the world. It allows the connection between people and just like any other technology, it is in constant evolution.

One of the branches in which the Internet has been developed is the "Internet of Things" (IoT), named by Kevin Ashton, the director of Auto-ID in 1999.

This is the Internet that connects together frequently used objects. In general terms, all these objects are what they are now called "Smart", intelligent devices capable of connecting to a network and performing a variety of actions that are previously commanded by the user.

The number of devices implementing the new technology grows rapidly. Business Insider forecasts that there will be 41 billion OR devices by 2027. As a matter of fact, 127 devices are connected to the Internet every second at the moment, according to Cisco founder David Evans.

On the other hand, air pollution is one of the most discussed topics worldwide. It is especially in large cities where, through toxic fog, air pollution is most common and evident.

Carbon dioxide is a greenhouse gas. It is considered pollutant when it comes from elements that require the use of fossil fuels. Since there are so many activities that generate this gas, it has become the pollutant that is causing the most damage on global warming.

Another gas that causes greenhouse effect on the planet is methane, which is originated from the gases emitted by bestiary and the currently prohibited chlorofluorocarbons, used in refrigerants and aerosols.

Therefore, a system for monitoring gases that are harmful to health such as carbon monoxide, carbon dioxide, methane and oxide, based on the integration of "cloud computing[1]" and IoT, is presented in this research.

[1] *Cloud computing is the delivery of different services through the Internet. These include data storage, servers, databases, networking, and software. Rather than keeping files on a proprietary hard drive or local storage device, cloud-based storage makes it possible to save them to a remote database. As long as an electronic device has access to the web, it has access to the data and the software programs to run it.*

Also, this device analyses data in real time and sends it via the internet to the platform "ThingSpeak". In sum, the platform has been programmed to issue an alert through a publication to the personal Twitter account when the configured digit is exceeded.

Real-time monitoring and a rapid alert system are an efficient system for gas detection.

The main objectives of the study are as following:

i. To propose the construction of the gas detection system as a final example of a current model of IoT monitoring.
ii. To propose the use of IoT for efficient monitoring of days in real time.

Once the objectives have been mentioned, clarify that the data collection digits are not completely real since each sensor requires an extremely elaborate calibration process and the main objective of the work is the first number above (i). Therefore, the calibration has been established through experimentation during the process.

1. IOT

1.1. What exactly is the Internet of Things, IoT?

The Internet of Things could be defined as the consolidation through the network of a “network” that houses a large multitude of objects or devices that are able to be connected with everything in the world.

In simpler words, it is a concept or technology which aims to connect all the devices to the internet and help them communicate with each other using the internet as a medium.

Not only these devices could be big and complex things as vehicles, household appliances, mechanical devices but also simple objects such as furniture, footwear, suitcases, measuring devices, biosensors, or any object that exists at this moment. All in all, anything that can be connected to the internet can be considered as a device for the Internet of Things.

- **Difference between the Internet and the IoT:**

The internet also is known as “the network of the networks”, which its practical schematics would not arrive until the early 1960s, is the product of people. This innovative technology, as Benson Hougland expressed, was created by people, for people, and about people. In other words, the internet connects people.

A new internet has been invented and it is called the Internet of Things. This one does not connect people but it connects things. These things can “share their experiences with other things”, and are given the ability to sense and communicate.

- **What the IoT offers to us:**

Like the vast majority of technological advances, the objective of this technology is to make our lives more comfortable as well as to provide greater security in various areas. Let's take an example: Imagine a fridge that would be able to warn us when the temperature drops or indicate that some food has expired or has passed or a desk that will record where everything has been left, or that we could know where each object that belongs to us is located at all times, or control our entire home from a smartphone or PC (from the entrance door to the toilet flush), or that we are alerted to allergens and their air concentration in a bracelet for people with allergies, etc.

These are some examples of everything that the Internet of Things could allow us to play, the total control of things.

1.2. History

The Internet of things has not been out there for that long. When I was younger, I had never heard nothing about it. That was due to its late popularization in 2010 and its mass market reach in 2014.

Despite the first examples of IoT from the early 1980s, where a Coca Cola machine located in the Carnegie Mellon University that local programmers would connect by the internet to the machine to see if there was drink available and if it was cold, before making the trip, the official term was not invented until 1999. Kevin Ashton named this technology when he was working at Procter & Gamble in order to promote Radio Frequency Identification (RFID), which is used, for instance as anti-theft alarms in products, or in key cards.

Figure 2: Kevin Ashton

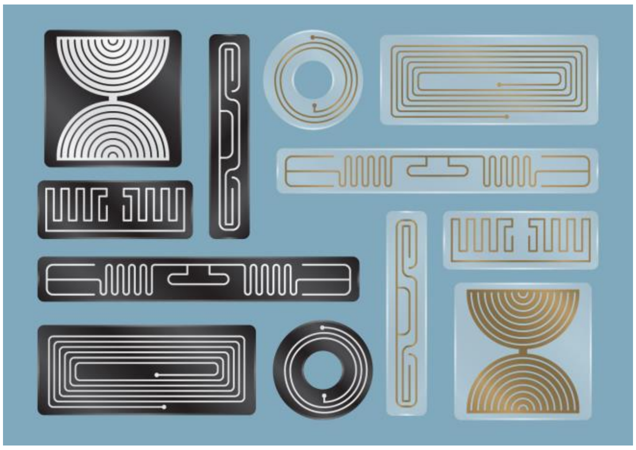


Figure 1: RFID tags

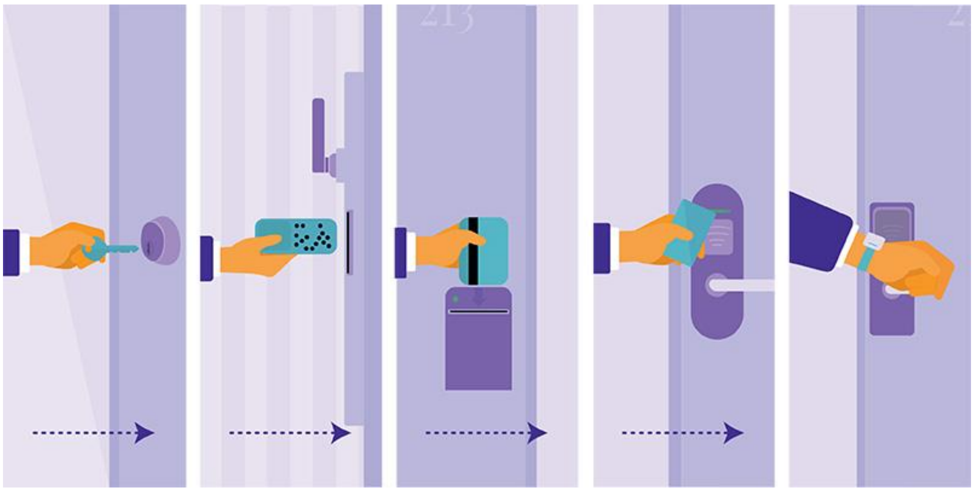


Figure 3: Key evolution

- **The take-off of IoT**

→ 2010: Google Maps Street View information leaked that Google was storing tons of people's Wi-fi network data.

The same year, Chinese Government used IoT as a strategic priority of the Five-year-plan (an economic program that set ambitious goals for industries and areas of production).

→ 2012: Europe's biggest Internet conference LeWeb theme was "Internet of Things". After that, famous magazines as Forbes, Fast Company or Wired used the same term to describe this new technology.

Figure 4: LeWeb Paris 2012

→ October 2013: IDC (International Data Corporation) stated that the IoT would be a $8.9 trillion market in 2020

→ 2014: Awareness was marked when Google announced that bought Nest for $3.2 billion.

"Worldwide spending on the Internet of Things (IoT) is forecast to reached $745 billion in 2019, an increase of 15.4% over the $646 billion spent in 2018", according to a new update to the International Data Corporation (IDC). It was clearly less than IDC suspected, nevertheless still a huge fast-growing market.

Segment	**2018**	**2019**	**2020**
Utilities	0.98	1.17	1.37
Government	0.40	0.53	0.70
Building Automation	0.23	0.31	0.44
Physical Security	0.83	0.95	1.09
Manufacturing & Natural Resources	0.33	0.40	0.49
Automotive	0.27	0.36	0.47
Healthcare Providers	0.21	0.28	0.36
Retail & Wholesale Trade	0.29	0.36	0.44
Information	0.37	0.37	0.37
Transportation	0.06	0.07	0.08
Total	**3.96**	**4.81**	**5.81**

Figure 5: IoT Endpoint Market by Segment, 2018-2020, Worldwide (Installed Base, Billions of Units)

1.3. What are the roots of the tree we know as IoT?

→ 1832: An electromagnetic telegraph was created by Baron Schilling in Russia.

→ 1833 Carl Friedrich Gauss and Wilhelm Weber invented their own code to communicate over a distance of 1200 m within Göttingen, Germany.

→ 1926: Nikola Tesla in an interview with Colliers magazine: *"When wireless is perfectly applied the whole earth will be converted into a huge brain, which in fact it is, all things being particles of a real and rhythmic whole.........and the instruments*

Figure 7: Electromagnetic telegraph by Baron Schilling, 1832

Figure 6: Nikola Tesla, 1926

through which we shall be able to do this will be amazingly simple compared with our present telephone. A man will be able to carry one in his vest pocket."

→ 1950: Alan Turing in his article Computing Machinery and Intelligence in the Oxford Mind Journal: *"...It can also be maintained that it is best to provide the machine with the best sense organs that money can buy, and then teach it to understand and speak English. This process could follow the normal teaching of a child."*

These days it would be identified as voice assistant.

→ 1964: In Understanding Media Marshall McLuhan stated: *"....by means of electronic media, we set up a dynamic by which all previous technologies -- including cities -- will be translated into information systems"*

→ 1966: Karl Steinbuch a German computer science pioneer said *"In a few decades time, computers will be interwoven into almost every industrial product"*

→ 1989: Tim Berners-Lee proposes the World Wide Web

Figure 9: Marshall McLuhan, 1964

Figure 8: World Wide Web

→ 1990: Considered the first IoT device, John Romkey created a toaster that could be turned on and off over the Internet for the October '89 INTEROP conference.

→ 1991: The first web page was created by Tim Berners-Lee

→ 1991: Mark Weiser's Scientific American article on ubiquitous computing called 'The Computer for the 21st Century' is written. *"The most profound technologies are those that disappear. They weave themselves into the fabric of everyday life until they are indistinguishable from it".*

→ 1993: Trojan Room Coffee Pot was created by Quentin Stafford-Fraser and Paul Jardetzky. It was located in the 'Trojan Room' within the Computer Laboratory of the University of Cambridge and was used to monitor the pot levels with an image being updated about 3x a minute and sent to the building's server. It was later put online for viewing once browsers could display images.

→ 1994: Steve Mann creates WearCam, a similar idea with the Coffee Pot, however it was faster, having better real time images.

→ 1995: The Internet goes commercial with Amazon and Echobay (Ebay).

→ 1997: Paul Saffo's prescient article "Sensors: The Next Wave of Infotech Innovation".

→ 1998: Google is founded.

→ 1998: Mark Weiser states: *"Ubiquitous computing is roughly the opposite of virtual reality. Where virtual reality puts people inside a computer-generated world, ubiquitous computing forces the computer to live out here in the world with people".*

→ 1999 - A big year for the IoT and MIT, when The Internet of Things term is officially named and established by Kevin Ashton, the executive director of the Auto-ID Center: *"I could be wrong, but I'm fairly sure the phrase "Internet of Things".*

Figure 11: John Romkey, 1990

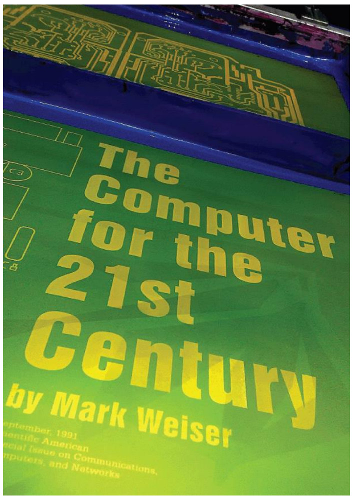


Figure 12: Mark Weisler's article, 1991

Figure 10: Trojan Room Coffee Pot, 1993

2. MAIN COMPONENTS OF IOT

Although IoT extends to a broad variety of use cases and applications, there are always 4 distinct components:

Those are sensors/devices, connectivity, data processing and user interface.

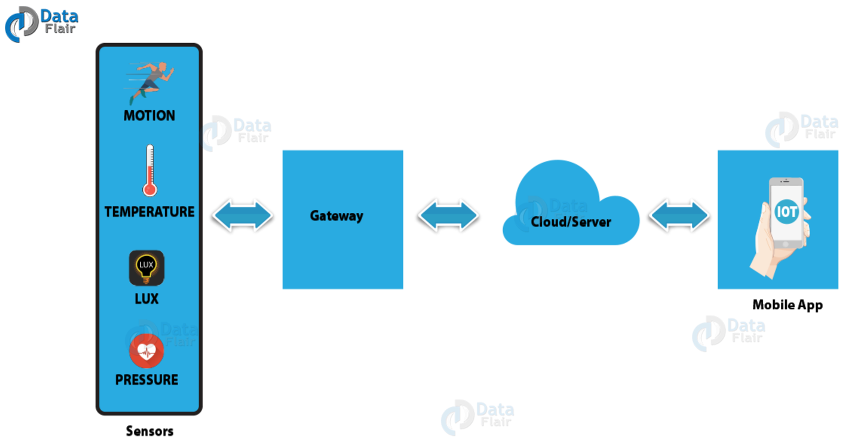


Figure 13: IoT process

- **Sensors or devices**

Sensors or devices collect data from their environment. Multiple sensors can be bundled together or sensors can be part of a device that does more than just sense things.

Figure 14: Sensors

- **Connectivity**

Is the medium **of** transport that the sensor's data use to transfer to the cloud. It can be through cellular, satellite, Wi-Fi, Bluetooth, low-power wide-area networks (LPWAN), connecting via a gateway/router or connecting directly to the internet via ethernet.

- **Data processing and analytics**

Once the data gets to the cloud, software performs some kind of processing on it. From very simple data, such as checking that the temperature reading is within an acceptable range. Or very complex, such as using computer vision on video to identify objects.

Just after the data is processed some data analytics needs to be applied to it, in order get more specific information

Figure 15: Data analytics

- **User Interface**

All the information collected by the system, has to be sent to the end-user, which could be via notification, text, email, etc. For example, an alert that the temperature has extremely risen. The user might have an interface that allows them to proactively check in on

the system as a mobile app or a web browser.

However, depending on the IoT device or application, the user may also be able to perform an action and affect the system. At the same time, it is possible that the system performs this actions via predefined rules or commands set by the user.

Figure 16: User interface

3. TYPES OF IOT

3.1. Based on the abilities

At the moment there are few capacities the IoT devices have. Some of them are:

- Communication and cooperation: Those objects will have the ability to connect with internet servers and/or with themselves. Being able to exchange and update data and establish communications with the servers.

- Directioning: This group of devices can be configured and located from any space of the net.

- Identification: The objects can be identified by means of technologies like RFID (Radio Frequency Identification), NFC (Near Field Communication), optical reading barcodes, QR codes, or hundreds of ways to identify a device in the net.

- Location: They will constantly have the knowledge of their physical location and capable of knowing the location at any time.

- Action: Those objects will have the capacity of environment manipulation.

3.2. Based on the level of intelligence

According to the experts at least 5 levels of intelligence:

- **Level 1: Unplugged**

This is the lowest level of intelligence. At this level, the device or sensor is 'unplugged' from the network, which means the machine is offline to the engineering organization. The state of the machine cannot be seen at any point in time.

Example: a vast variety of industrial pumps still are completely mechanical devices with no sensors to instrument them.

It will have the ability to communicate its state and characteristics.

- **Level 2: Edge intelligence**

At this level of intelligence, the device is connected to the network.

Examples: Cars that can alert the drivers to basic conditions that need intervention, a boiler that has edge intelligence to switch on/switch off valves based on steam pressure.

- **Level 3: Remote monitoring**

At this stage, the device can be remotely monitored and monitored from a central command centre network. That means it will be able to communicate its state as well as its characteristics in real-time.
Example: Remote Patient Monitoring (RPM), a technology used to electronically transmit information between patients and physicians, and it is just one delivery system within the broader telehealth industry.

- **Level 4: Predictive**

At this stage, there is enough insight available to the object that it will be capable of perceiving the environment and predict failures, defects and the need for replacements when they are prone to breaking. It can help since some repairs can be expensive and unaffordable.
Example: Predictive maintenance by industries like process manufacturing, oil and gas, electric power industry, railways or construction.

- **Level 5: Omnipresence intelligence**

This is the most evolved state of engineering intelligence where all assets the organization has deployed are connected in real-time seamlessly to field force, head office engineers and command centre observers in real-time. The objects communicate, identify, locate, analyse their environment, decides and executes in real-time. Very few global engineering firms are at this level of maturity.

3.3. Based on the system complexity

- **Level 1:**

The system has a single node/device that performs sensing and/or actuation, stores data, performs analysis and hosts the application. They are suitable for modelling low-cost and low-complexity solutions where the data involved is not big and the analysis requirements are not computationally intensive.
Example: Home Automation System.

- **Level 2:**

The system has a single node that performs sensing and/or actuation and local analysis. Data is stored in the cloud and the application is usually cloud-based. They are suitable for solutions where the data involved is big; however, the primary analysis requirement is not computationally intensive and can be done locally.
Example: Smart Irrigation

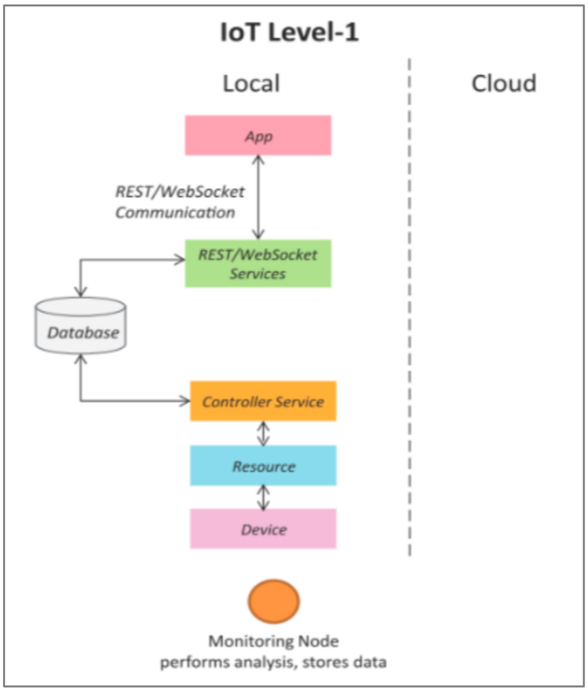


Figure 17: IoT system level 1

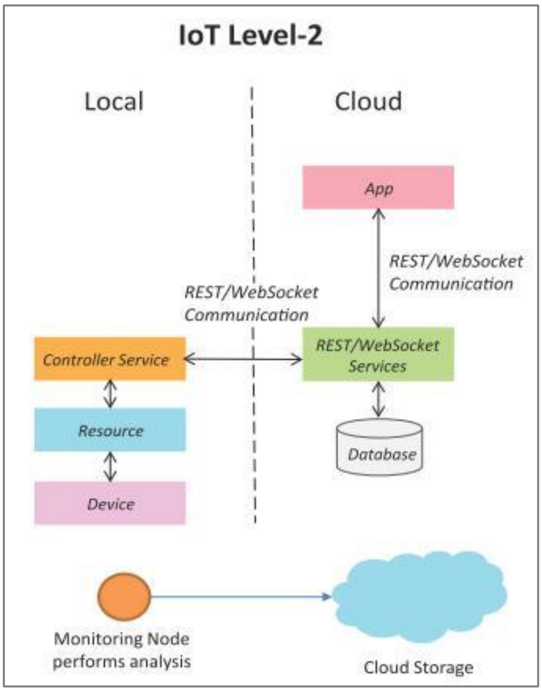


Figure 18: IoT system level 2

- **Level 3:**

The system has a single node. Data is stored and analysed in the cloud and the application is cloud-based. They are suitable for solutions where the data involved is big and the analysis requirements are computationally intensive.
Example: Tracking Package Handling Sensors.

- **Level 4:**

The system has multiple nodes that perform local analysis. Data is stored in the cloud and the application is cloud-based. Observer devices can receive information collected in the cloud from IoT devices. They are suitable for solutions where multiple nodes are required, the data involved is big and the analysis requirements are computationally intensive.

Example: Noise Monitoring Sound Sensors.

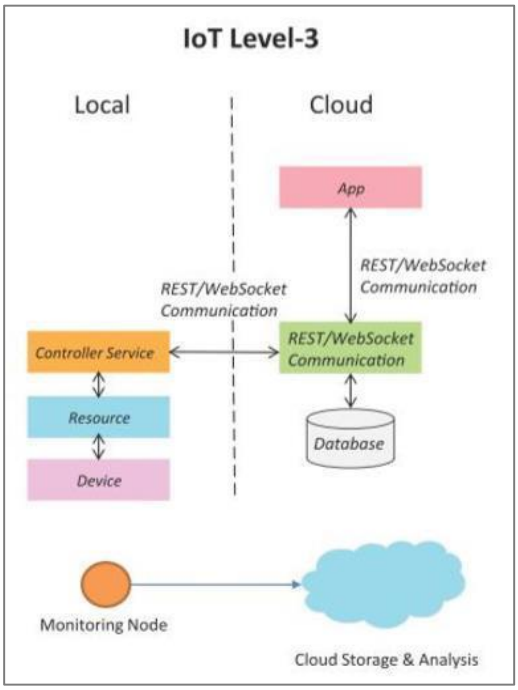


Figure 19: IoT system level 3

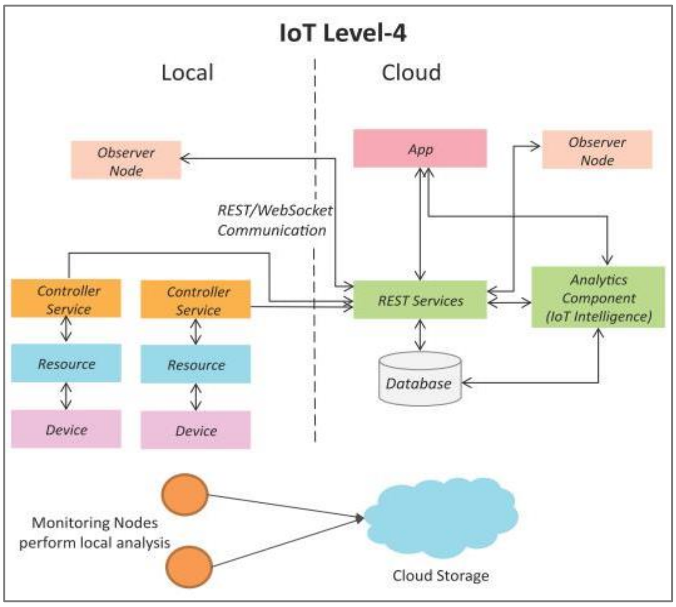


Figure 20: IoT system level 4

- **Level 5:**

The system has multiple end devices and one coordinator device. The end nodes perform sensing and/or actuation. The coordinator node collects data from the end nodes and sends it to the cloud. Data is stored and analysed in the cloud and the application is cloud-based.

They are suitable for solutions based on wireless sensor networks, in which the data involved is big and the analysis requirements are computationally intensive.

Example: Forest Fire Detection: sensors measure the temperature, smoke, weather, slope of the earth, wind speed, speed of fire spread, flame length.

- **Level 6**

The system has multiple independent end devices that perform sensing and/or actuation and send data to the cloud. Data is stored in the cloud and the application is cloud-based. The analytics component analyses the data and stores the results in the cloud database. The results are visualized with the cloud-based application. The centralized controller is aware of the status of all the end devices and sends control commands to the devices.

Example: Weather Monitoring System. Sensors used: wind speed and direction, solar radiation, temperature (air, water, soil), humidity, precipitation, snow depth, barometric pressure, soil moisture, etc.

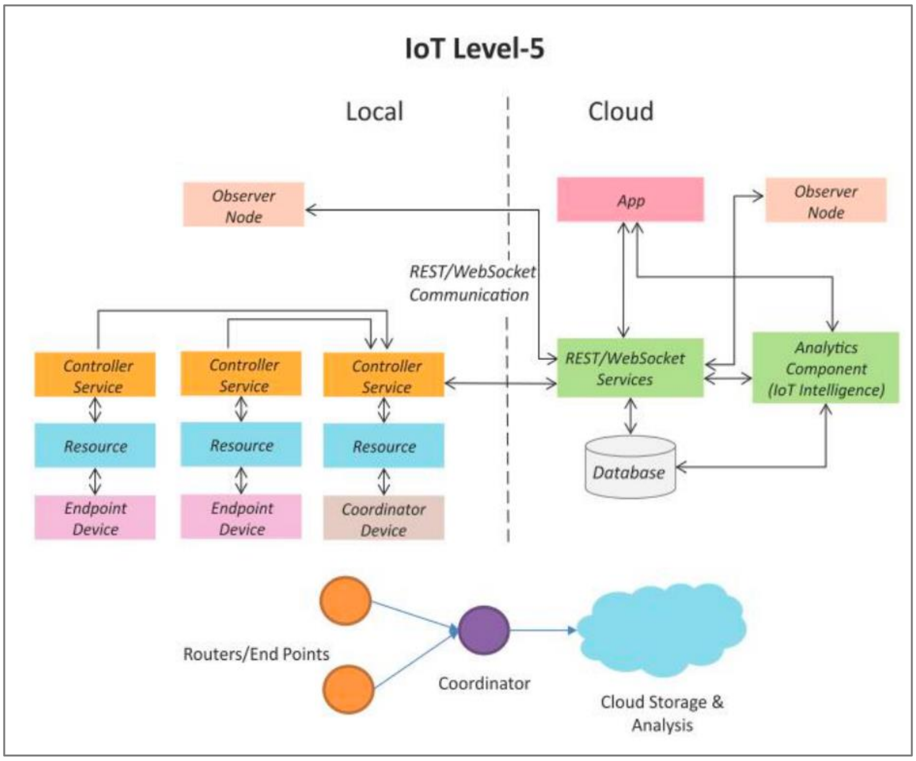


Figure 21: IoT system level 5

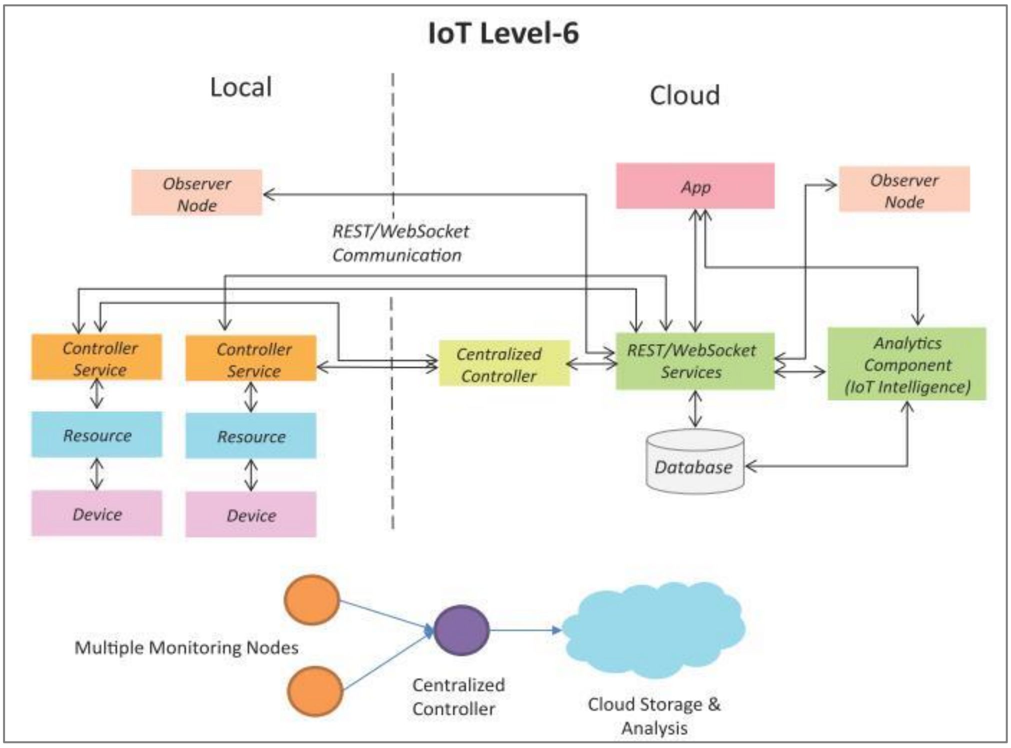


Figure 22: IoT system level 6

4. ADVANTAGES AND DISADVANTAGES OF IOT

- **Major advantages**

Monitor and information

It is obvious that having more information helps making better decisions. Knowing the exact quantity of supplies or the air quality in your home, can further provide more information that could not have previously been collected easily. As a result, it could save you from “urgent” situations.

Money saving

Although IoT devices are not cheap, if the price of the equipment is less than the amount of money saved, then it becomes a great advantage. Moreover, we can be alerted in case of possible breakdowns or damages to the system.

Automation of daily tasks leads to better monitoring of devices

The IoT allows to automate and control the tasks that are done on a daily basis, avoiding human intervention. It helps to maintain transparency in the processes, maintain the quality of service and can also take necessary action in case of emergencies.

Time saving and efficient

The amount of time saved because of IoT could be quite large.

The machine-to-machine interaction provides better efficiency; therefore, accurate results can be obtained fast. This results in saving valuable time.

Better Quality of Life

In the end the main purpose of IoT is to improve our life quality. All the applications of this technology culminate in increased comfort, convenience, and better management, thereby improving the quality of life.

- **Major disadvantages**

Over dependence on technology

As the IoT continues developing and growing, there's a strong possibility that someday we will be totally relying on IoT devices. This could create trouble in the human health.

Losing security on privacy

As different technologies and devices get involved in a single system, there is monitoring by more than one company. That means that personal data is stored by more companies that we thought.

Lesser Employment Prospects

With IoT, daily activities getting automated and naturally there will be fewer requirements of human resources, which may create employment issues in the society.

Complexity

With all complex systems, there is the possibility of failure. Failures could be skyrocket in the IoT and the big disappointment of the user.

Incompatibility

Although this is a new technology, there are already numerous brands in the market and not all of them can work with each other due to their different operating system. Therefore, after carefully deciding the brand we will be using, when purchasing another gadget, our new options become very reduced.

5. THE ENVIRONMENT AND IOT

The applications of IoT in environmental monitoring are broad − environmental protection, extreme weather monitoring, water safety, endangered species protection, commercial farming, and more. In these applications, sensors detect and measure every type of environmental change.

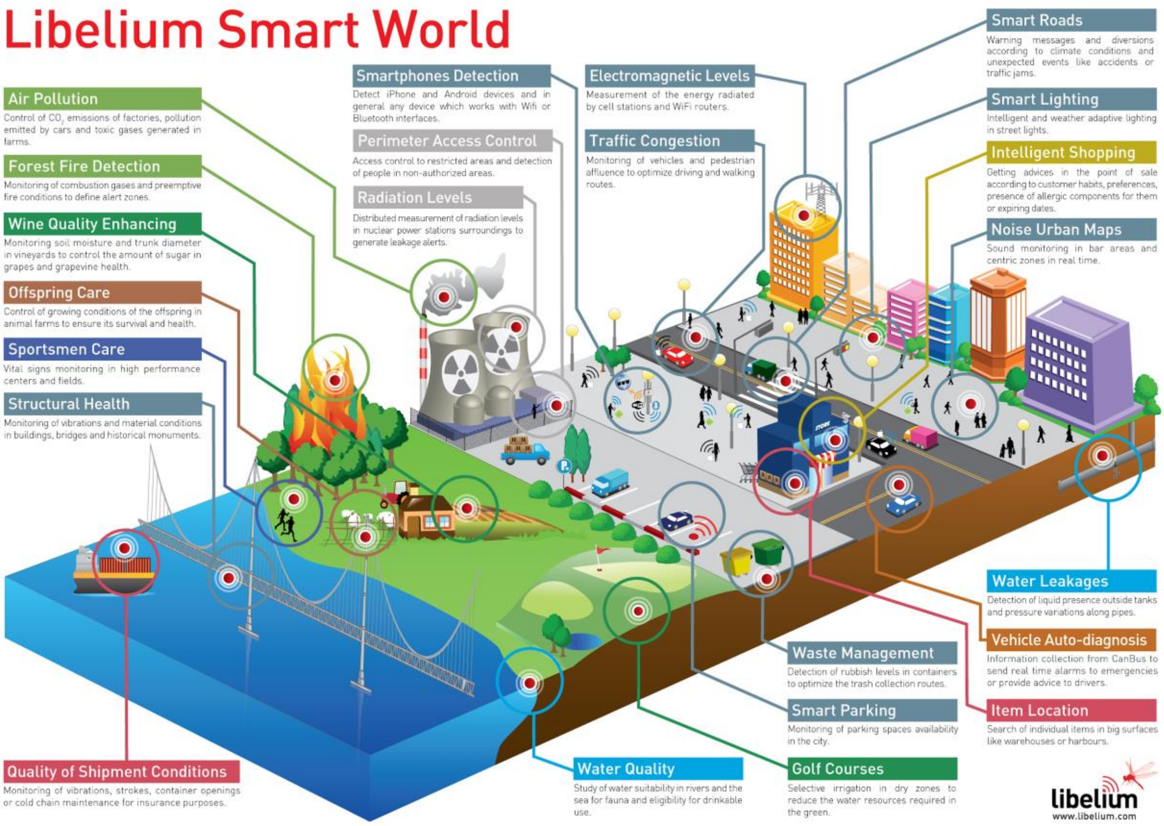


Figure 23: Libelium Smart World

As you already know for the title, this project is focused on air pollution applications using IoT. IoT improves on this technology by reducing the need for labour, allowing frequent sampling and increasing its range, sophisticated testing on-site, and binding response efforts to detection systems.

A system that detects specific pollutant and harmful gases through different sensors will be created. It will be connected to a platform which is going to constantly collect all the data. When pollution goes beyond some level, it will trigger some alert so that it can be noticed at the exact moment.

In order to make this project possible, first of all, it will be needed to study and have some basic knowledge of some of the main pollutant gases so that I can decide which gas sensors I will need to use. Secondly, a processor system and a platform will also have to be chosen.

All the materials final decision will be based on the overall combination of these three specific topics:

1. Necessity
2. Usefulness
3. Economy

5.1. Main pollutant gases

- **Ozone**

Ozone or trioxygen (O3), is an inorganic molecule. Ground level ozone is different from the ozone that protects people from the sun UV rays. It is actually created on the ground when volatile organic compounds (motor vehicle exhaust, emissions from electric utilities and industrial facilities, chemical solvents, and gas vapours) chemically react with oxides of nitrogen in the presence of sunlight. During heat waves in urban areas, ground level ozone pollution can be 20% higher than usual. This gas, not only can have a harmful effect on the ecosystem, causing changes to the quality of habitats, nutrient cycles, and water, but also it can have a detrimental effect on human's health, especially if he has been diagnosed with asthma, COPD, or bronchitis.

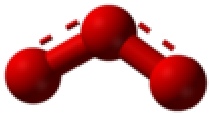


Figure 24: Ozone

- **Carbon Monoxide**

Carbon Monoxide (CO) is a colourless, odourless, and tasteless but highly toxic gas. It has a role in the formation of ground-level ozone. It binds to the site in hemoglobin that normally carries oxygen, leaving it ineffective for delivering oxygen to bodily tissues. Concentrations as low as 667 ppm may cause up to 50% of the body's hemoglobin to convert to carboxyhemoglobin. A level of 50% carboxyhemoglobin may result in seizure, coma, and fatality.

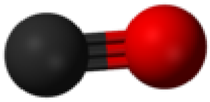


Figure 25: Carbon Monoxide

- **Carbon Dioxide**

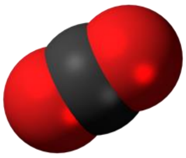


Figure 26: Carbon Dioxide

Carbon Dioxide (CO_2) is a colourless gas that occurs naturally in Earth's atmosphere as a trace gas. Atmospheric carbon dioxide is the primary carbon source for life on Earth and its concentration has been regulated by photosynthetic organisms.

Although CO2 is not classified as toxic or harmful in accordance with Globally Harmonized System of Classification and Labelling of Chemicals standards of United Nations Economic Commission for Europe, it is heavier than air and in concentrations up to 1% (10,000 ppm), it will make some people give the lungs a stuffy sensation. Concentrations of 7% to 10% (70,000 to 100,000 ppm) may cause suffocation, even in the presence of sufficient oxygen, manifesting as dizziness, headache, visual and hearing dysfunction, and unconsciousness within a few minutes to an hour.

- **Nitrogen Oxides**

Nitrogen Oxides (NO_x) contribute to the formation of smog and acid rain, as well as affecting tropospheric ozone. It is produced by the emissions formed from power plants, on-road vehicles, and off-road vehicles. When reacting with other components, some small particles can penetrate deeply into sensitive lung tissue and damage it, causing or worsening respiratory diseases.

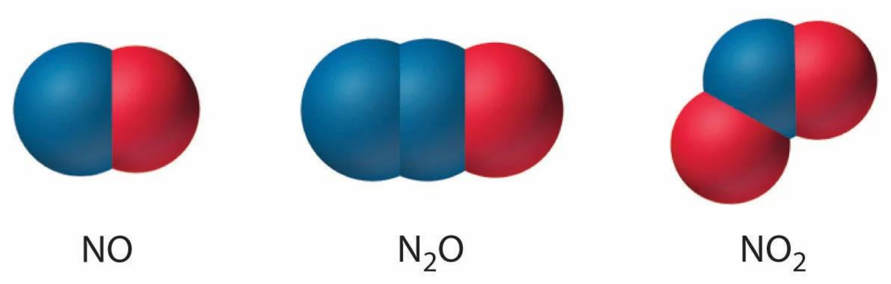


Figure 27: Nitrogen Oxides

- **Methane**

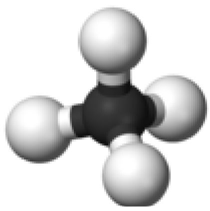


Figure 28: Methane

Methane (CH_4), commonly known as marsh gas, comes from the gas emitted by livestock, and swamps. If methane leaks into the air before being used it absorbs the sun's heat, warming the atmosphere. For this reason, it's considered a greenhouse gas, like carbon dioxide.

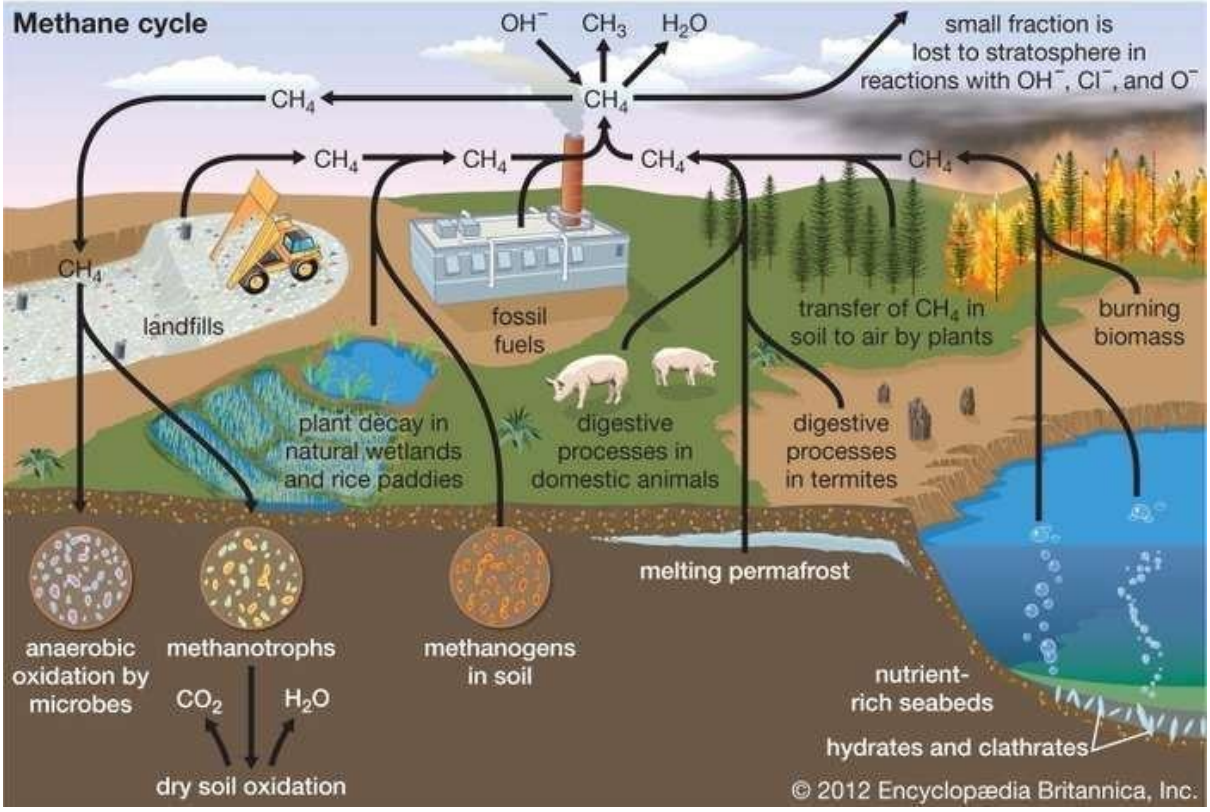


Figure 29: Methane cycle

5.1.1. Final gas study choice

After knowing some of the main pollutant gases, carbon monoxide and methane are the ones chosen to be studied. The decision is based on the fact that ozone and carbon monoxide are extremely harmful to human's health, since they could end with someone's life. Furthermore, methane must be studied as a huge part of our province's economy is provided by farms, crop fields and swamps, which are basically the main causative of methane.

5.2. Gas Sensors

A gas sensor is a device which detects the presence or concentration of gases in the atmosphere. Based on the concentration of the gas the sensor produces a corresponding potential difference by changing the resistance of the material inside the sensor, which can be measured as output voltage. Based on this voltage value the type and concentration of the gas can be estimated.

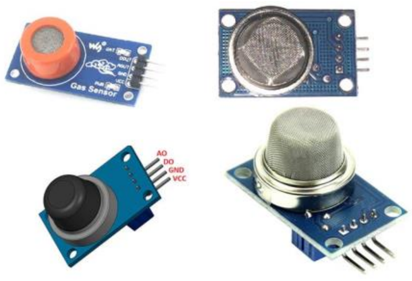


Figure 30: Gas sensors

5.2.1. Types of sensors

They typically classified into various types based on the type of the sensing element it is built with:

- **Metal Oxide based gas sensor**

Figure 31: Metal Oxide based gas sensors

The ability to detect gases depends on the chemiresister to conduct current. The most commonly used chemiresistor is Tin Dioxide (SnO_2) which has free electrons. The oxygen particles attract the free electrons present in SnO_2 which pushes them to the surface of the SnO_2. As there are no free electrons available output current will be zero.

The below images shown the oxygen molecules (blue colour) attracting the free electrons (black colour) inside the SnO_2 and preventing it from having free electrons to conduct current.

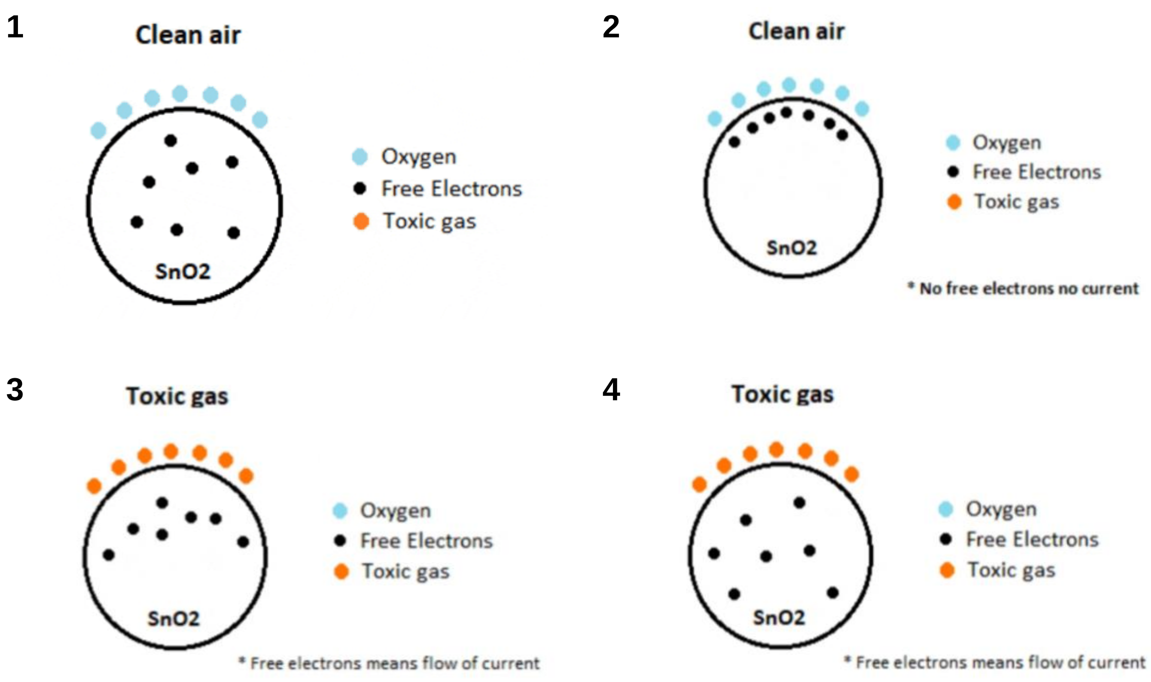


Figure 32: Metal Oxide gas sensor cycle

When the sensor is placed in the toxic or combustible gases environment, this reducing gas (orange colour) reacts with the adsorbed oxygen particles and breaks the chemical bond between oxygen and free electrons thus releasing the free electrons. As the free electrons are back to its initial position, they can now conduct current, this conduction will be proportional the number of free electrons available in SnO2, if the gas is highly toxic more free electrons will be available.

Due to their low cost and flexibility in production; simplicity of their use; large number of detectable gases/possible application fields this type of gas sensor is the most used ever. MQ gas sensors are the most popular these days due to their excellent compatibility with Arduino.

MQ sensors are special sensors designed to have sensitivity for detecting some gas like LPG, CO, Propane etc. In Chinese, 'sensitive' means 'Mǐngǎn' and 'Gas to' means 'Qǐ lai'. So, more or less, MQ stands for the sensors having sensitivity towards (or to) gas.

As it can be seen in the table below, they are qualified depending on the gases they can detect:

Sensor	Detects	Heater Voltage
MQ-2	Methane, Butane, LPG, smoke	5V
MQ-3	Alcohol, Ethanol, smoke	5V
MQ-4	Methane, CNG Gas	5V
MQ-5	Natural gas, LPG	5V
MQ-6	LPG, butane gas	5V
MQ-7	Carbon Monoxide	Alternating 5V and 1.4V
MQ-8	Hydrogen Gas	5V
MQ-9	Carbon Monoxide, flammable gasses.	Alternating 5V and 1.4V
MQ131	Ozone	6V
MQ135	Air Quality (Benzene, Alcohol, smoke)	5V
MQ136	Hydrogen Sulfide gas	5V
MQ137	Ammonia	5V
MQ138	Benzene, Toluene, Alcohol, Acetone, Propane, Formaldehyde gas, Hydrogen	5V
MQ214	Methane, Natural gas	6V
MQ216	Natural gas, Coal gas	5V
MQ303A	Alcohol, Ethanol, smoke	0.9V
MQ306A	LPG, butane gas	0.9V
MQ307A	Carbon Monoxide	Alternating 0.2V and 0.9V.
MQ309A	Carbon Monoxide, flammable gasses	Alternating 0.2V and 0.9V
MG811	Carbon Dioxide (CO2)	6V
AQ-104	Air quality	-

Figure 33: List of MQ gas sensors

- **Optical gas sensor**

A bit more complex than the Metal Oxide based gas Sensor, absorption, fluorescence, scattering, and reflection of light, and changes in refractive index and optical path length are successfully used for detection of gas components.

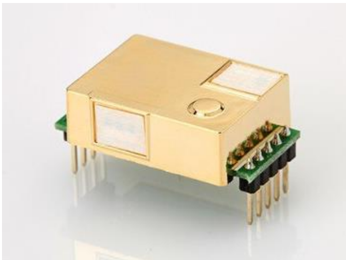


Figure 34: Optical gas sensor

- **Electrochemical gas sensor**

This sensor type works by means of gas diffusion. Gas finds its way into the outlet of the membrane on top of the sensor housing. Once the gas reaches the working electrode, an electrochemical reaction occurs.

This reaction is either an oxidation or a reduction, depending on the type of gas.

- Oxidation causes the flow of electrons to move from the working electrode to the counter electrode through the external circuit.
- Reduction on the contrary, causes the flow of electrons to move from the counter electrode to the working electrode. Either direction of electron flow creates an electric current proportional to the concentration of gas.

The electrons within the external circuit detect and amplify this current. It then scales the output accordingly with the calibration to give a reading in a percentage of the volume of gas.

Nevertheless, electrochemical sensors depend on chemical processes with proportional temperature rates. Since temperatures vary, some form of temperature compensation is encouraged for the most accurate readings.

Figure 35: Electrochemical gas sensor

- **Capacitance-based gas sensor**

It can have excellent characteristics such as high sensitivity, high selectivity, high stability, and low power consumption since it can detect gases through the touch of a porous material. Thanks to the highly sensitive material, the gas level can be detected and later, studied.

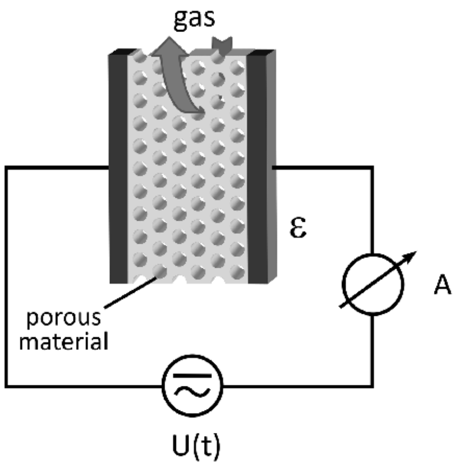


Figure 37: Capacitance-based gas sensor functioning

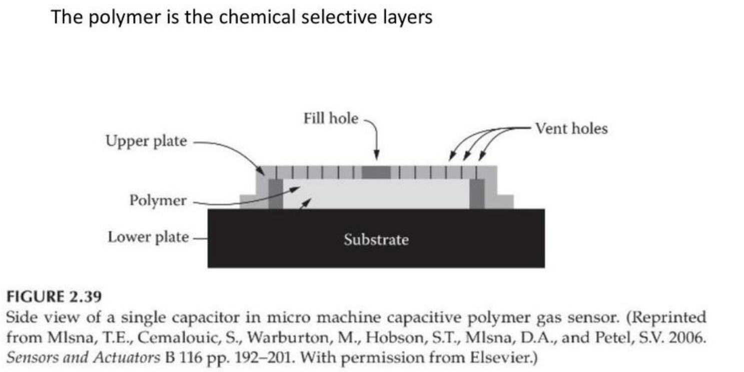


Figure 36:Side view of a single capacitor in micro machine capacitive polymer gas sensor

- **Calorimetric gas sensor**

It is a chemical sensor that uses micromachining techniques. The principle of operation is to detect the heat evolved from combustible gases on the sensor surface.

Figure 38: Calorimetric gas sensor

Conventional calorimeters for fuel gas, natural gas, or coal gasification products are highly complex, sophisticated apparatus for monitoring the heats of combustion of fuels in the burners or turbines of power plants. Typical calorimeters are large, self-standing units that are used to monitor large-scale systems, and there are no smaller calorimeters appropriate for the various burners or turbines of smaller-scale systems.

- **Acoustic wave based gas sensor**

They are sensitive to temperature, pressure, force, electric field, viscosity, vapour, gas and ions concentrations, meaning they are highly complex. These physical and chemical quantities act on the device either directly or indirectly by means of an interface that transforms the measured quantity into quasistatic stress, thus modifying the elastic properties of the substrate. This deformation is caused by the particle displacement due to vibration, causing a variation, for example, in the phase velocity or in the insertion loss of the acoustic wave.

This technology is used in monitoring of: vehicle gas species, gas constituents in industrial exhaust stacks to determine compliance and industrial combustion processes for control and optimization.

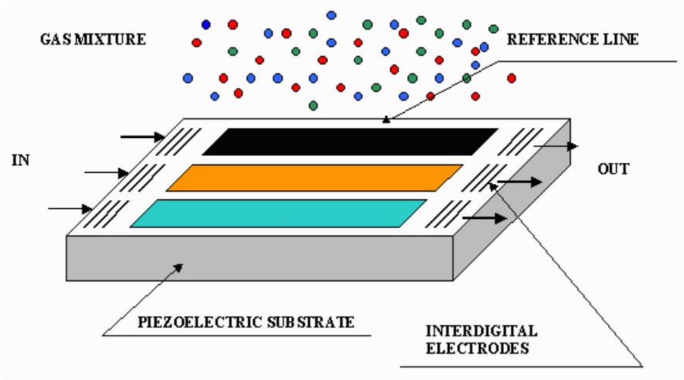


Figure 39: Acoustic wave based gas sensor

5.2.2. Final sensors choice

In short, since the project is not at any industry level, gas sensors like Capacitance-based, Calorimetric and acoustic-based, can be completely discarded. Besides, the Electrochemical gas sensor depends on chemical processes with proportional temperature rates, as it has been said before. As the temperature can vary, it can get quite unstable. The Optical gas sensor only works with light, which means no data would be collected at night.

As a result, the Metal Oxide based gas Sensor is the only option left. MQ is the most popular among all the Metal Oxide based gas sensors owing to their low price and great effectivity. For this reason, MQ gas sensors are the ones that will be used.

What is more, since all of the MQ sensors cost the same (around 5 or 6€ in Amazon), there won't be any economical comparison.

For the detecting the first gas, ozone MQ-131 will be used due to the fact that is the only MQ sensor that can detect this gas.

Sensor	Detects	Heater Voltage
MQ131	Ozone	6V

As for the second gas, carbon monoxide, there are four different sensors that can work, which are:

Sensor	Detects	Heater Voltage
MQ-9	Carbon Monoxide, flammable gasses.	Alternating 5V and 1.4V
MQ307A	Carbon Monoxide	Alternating 0.2V and 0.9V.
MQ309A	Carbon Monoxide, flammable gasses	Alternating 0.2V and 0.9V

The heater voltage should be as close as the ozone sensor. Therefore, MQ307A and MQ309A are no longer a choice. MQ-9 is also discarded because there are not any intentions to study flammable gasses. As a consequence, MQ-7 will be used to detect Carbon Monoxide.

The last gas is methane, and like ozone, there is more than one choice:

Sensor	Detects	Heater Voltage
MQ-2	Methane, Butane, LPG, smoke	5V
MQ-4	Methane, CNG Gas	5V
MQ214	Methane, Natural gas	6V

All of them can detect two or more gases, although MQ-2 can detect four different gases, which is not my interest. As for MQ214, the tutor told me it was hard to find owing to the fact it is not very common. Following the tutor's recommendation, it has been decided to use MQ-4 because of its simplicity and popularity.

5.3. Development Board / Processor system

A development board is a printed circuit board with circuitry and hardware designed to facilitate experimentation with a certain microcontroller.

It also a method to prototype applications in products. It will have some provision to accept and run a user supply program such as downloading a program through a serial port to flash memory or some form of programmable memory in a socket in earlier systems.

Typical components of a development board include:

- Power circuit: typically set up to run off of a 9V power supply
- Programming interface: an easy way program the microcontroller from a computer
- Basic input: usually buttons
- Basic output: usually LEDs
- I/O pins: to be used for everything else, motors, sensors, LCD screens, etc.

For instance, in Arduino, the most popular development board, the components are distributed this way:

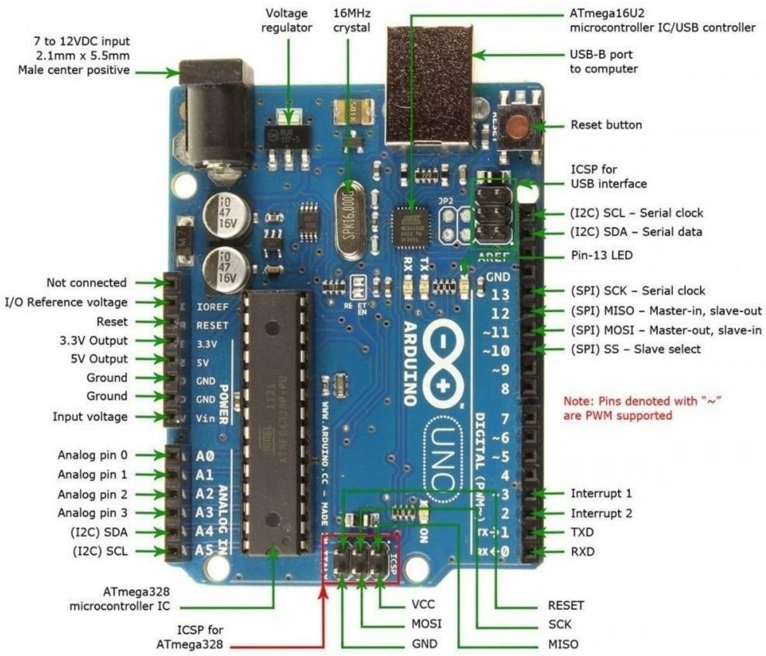


Figure 40: Arduino component distribution

5.3.1. Types of development boards

Although there are many different types and brands of development boards designed for IoT projects, the top 5 are going to be compared in order to decide the one that will be used in my project.

a) **Arduino Uno**

Price: 22.57 €

Arduino Uno R3 is simplest yet the most powerful prototyping environment. It is based on the ATmega328P which has 14 digital input/output pins and six analog inputs. Though it comes with just 32 KB of Flash memory, it can accommodate code that deals with complex logic and operations.

From sensors to actuators to libraries, it has a thriving ecosystem. Almost every prototyping environment tries to be compatible with the Arduino pin breakout. The open source IDE to develop sketches is another reason for its popularity. With a simple syntax based on ‘C’ language, the code is easy to learn.

However, Arduino doesn’t have built-in Wi-Fi, which means an external Wi-Fi microchip must be added.

Figure 41: Arduino UNO

b) **Raspberry Pi**

Price 35 €

Raspberry Pi 4 includes built-in Wi-Fi and Bluetooth. It is a powerful platform since it is based on a Broadcom BCM2711 SoC with a 1.5 GHz 64-bit quad-core ARM Cortex-A72 processor and 2GB, 4GB or 8GB RAM (depending on the model). It has two micro-HDMI ports, four USB ports and 40 GPIO pins, which means you can connect many peripherals and accessories to the Pi.

There are third party breakout boards to connect various Arduino shields to the Pi.

On the contrary to Arduino, Raspberry Pi’s source its private.

Raspberry Pi 3 is the most affordable and powerful at the same time development board.

Figure 42: Raspberry Pi 4

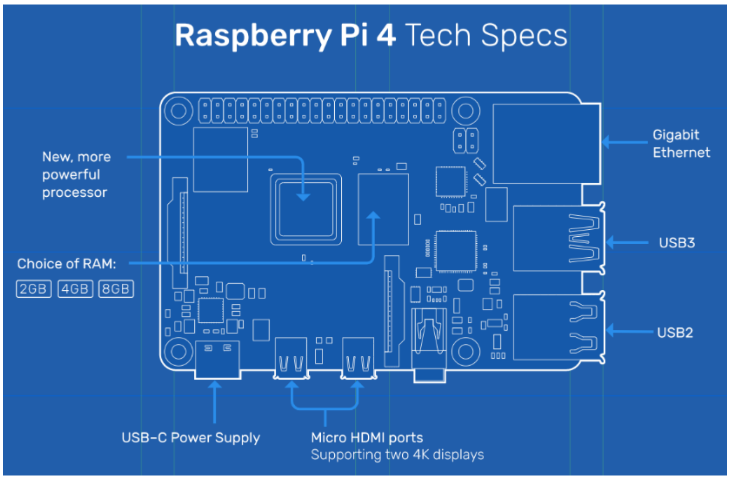


Figure 43: Raspberry Pi 4 component distribution

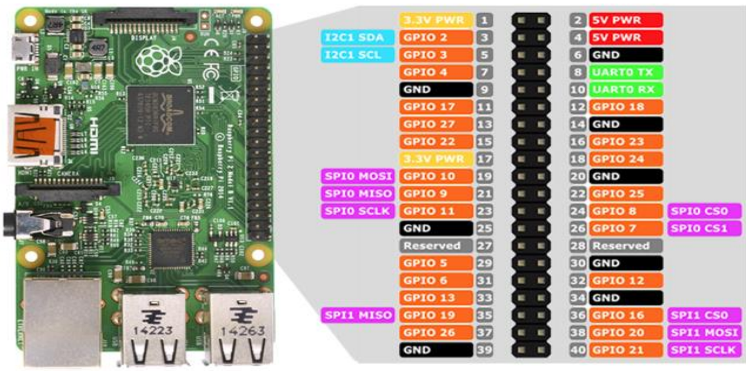


Figure 44: Raspberry Pi GPIO pin distribution

c) Intel Edison

Price: 70 €

Intel Edison is a high-performance, dual-core CPU with a single core microcontroller that can support complex data collection. It has an integrated Wi-Fi, Bluetooth® 4.0 support, 1GB DDR and 4GB flash memory.

It comes with two breakout boards; one that's compatible with Arduino and the other board designed to be a smaller in size for easy prototyping.

The Arduino breakout board has 20 digital input/output pins, including four pins as PWM outputs, Six analog inputs, one UART (Rx/Tx), and one I2C pin.

Edison runs on a distribution of embedded Linux called Yocto. It got certified by Microsoft, AWS, and IBM for cloud connectivity.

Figure 45: Intel Edison

	Linux GPIO pin	GPIO pin mux			SoC pin modes		Output enable (high = output)	Pullup enable
		Linux pin	0 (low)	1 (high)	0	1	Linux	Linux
IO0	130				GPIO	UART	248	216
IO0	130				GPIO	UART	248	216
IO1	131				GPIO	UART	249	217
IO2	128				GPIO	UART	250	218
IO3	12				GPIO	PWM	251	219
IO4	129				GPIO	UART	252	220
IO5	13				GPIO	PWM	253	221
IO6	182				GPIO	PWM	254	222
IO7	48				GPIO		255	223
IO8	49				GPIO		256	224
IO9	183				GPIO	PWM	257	225
IO10	41	263	PWM	see 240	GPIO	I2S or SPI	258	226
		240	GPIO or I2S	GPIO or SPI_FS				
IO11	43	262	PWM	see 241	GPIO	I2S or SPI	259	227
		241	GPIO or I2S	GPIO or SPI TXD				

Figure 46: Intel Edison component distribution

d) ESP8266

Price: 8,50 €

It is a low-cost Wi-Fi SOC (System on a Chip[2]) that helped Arduino to make a big step. This module, allows to connect the projects to the net and work on them through the internet. It has a full TCP/IP battery and a microcontroller. It has 3.3V of intensity and an 80 Mhz Tensilica Xtensa LX106 processor. A 64 KB RAM memory for commands and 96 KB for data, 16 GPIO pins, UART pins, and SPI and 12C interface.

In terms of Wi-Fi connectivity, is compatible with the standard IEEE 802.11 b/g/n and supports WEP, WPA and WPA2 security.

A negative aspect is that it only has one analog pin (ADC)

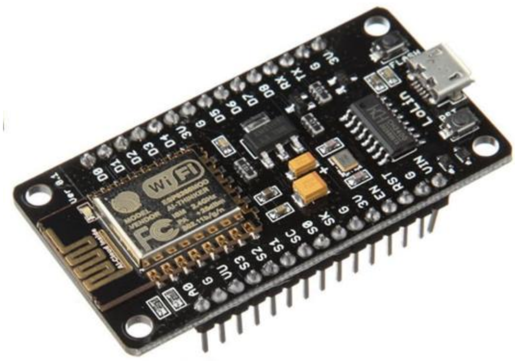


Figure 47: ESP8266

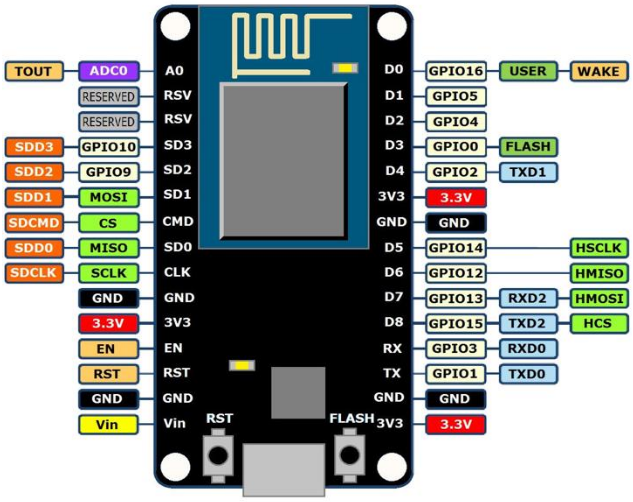


Figure 48: ESP8266 pin distribution

e) ESP32

Price: 10 €

Is the successor of ESP8266. It provides a significant increase in power since it has a dual core with frequencies of up to 240 megahertz, that is 50 percent more. In addition the ESP32 adds an ultra-low consumption processor, which allows maximum energy savings when the main processor is not needed. It has 32 GPIO pins, a 12-bit, 18 channel analog to digital converter (ADC), and more features like MAC ethernet interface or CAN controller.

[2] *Wi-Fi SOC (System On a Chip) is a complement that allows the development board to connect to Wi-Fi in order to be able to share the project.*

Figure 49: ESP32

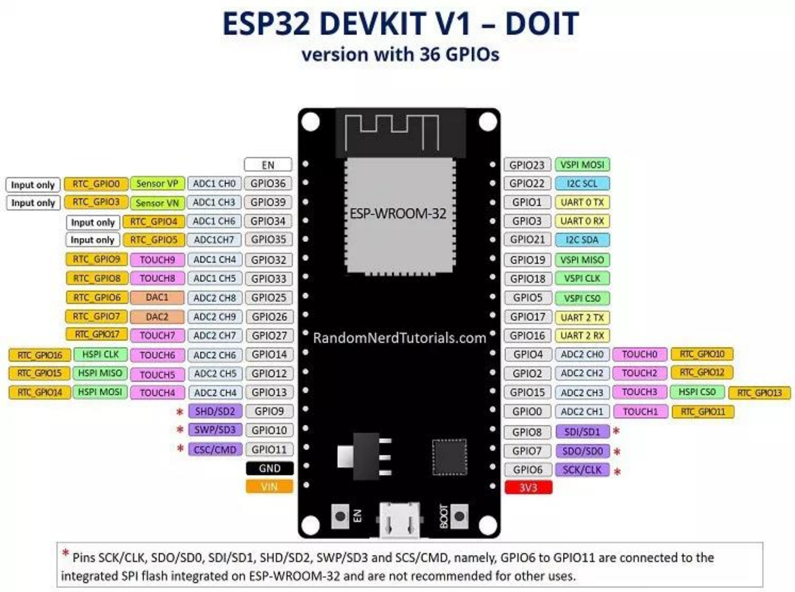


Figure 50: ESP 32 general pin distribution

5.3.2. Final development board choice

On the one hand, although its affordable price, Raspberry Pi has been completely discarded due to the fact that USB ports won't be used at all. In addition, it would have to be used third party breakout boards to connect the MQ gas sensors to Pi.

On the other hand, Intel Edison was not either the best choice. First of all, the price is quite higher than the other boards. Secondly, having 4GB of flash memory is not what is looked for. In other words, there is no need of such a powerful board as Intel Edison is.

Yet Arduino UNO is discarded due to the simple fact that doesn't have inbuilt Wi-Fi. It is possible to connect a Wi-Fi chip, but my tutor recommended to directly use a board that already has it. Despite this, it is going to be used in the process in order to ensure all the components work.

It can be clearly seen that ESP32 doubles his predecessor in terms of speed and exceeds in the number of GPIO pins and analog to digital converters (ADC). In the project more analog signals than the ESP8266 board can handle will be needed, since it has only one ADC pin while ESP32 has 18. In terms of language, both use similar to the language C from Arduino, meaning that it won't be a problem to handle it because it has been learned during the past year.

All the same, it is believed choosing ESP32 will be the best choice.

5.4. Platform

An IoT platform is a multi-layer technology that enables management, connectivity and network management, data management, processing and analysis, application development, security, access control, monitoring, event processing and interfacing/integration of connected devices within the Internet of Things universe. It basically connects the hardware, however diverse, to the cloud by using flexible connectivity options, enterprise-grade security mechanisms, and broad data processing powers.
IoT platforms are a vital component of the IoT ecosystem and a fast-growing market, expected to exceed $22 billion by 2023.

5.4.1. Types of platforms

There is a very high number of different IoT platforms through the internet. Some of the most popular platforms with free versions will be compared.

- **Samsung Artik Cloud**

It comes from the old platform called SAMlio and later SmartThings. It is not only a cloud platform, we can also buy hardware, the Artik 1020 device. This does not mean that we must have an Artik module to communicate with the cloud, we can do it with any compatible device.
One of the biggest advantages is that it supports IFTTT[3], allowing us to communicate with anything you can imagine.
The downside is that it does not have a library to manage communications from other development boards like ESP32 or even Arduino which means it has to be done manually.

Maximum number of messages per device and day	Historic time (months) Quick access and statistics	Number of devices	Monthly price
150 (1 msg every 10 min)	3 months	Unlimited	Free

- **aREST Framework**

aRest Framework is focused on supporting boards like Arduino, ESP8266, ESP32 and Raspberry Pi. It is a complete solution to create applications with RESTful services. Supports WiFi, Ethernet, Bluetooth and Serial communications. It includes libraries for Arduino and also on the server side, to control communications.
This framework is open source. Within the official Arduino environment itself, we can download the library from its repository and start working.
One of the least attractive features is that the documentation is quite scarce and only supports the REST protocol, therefore we are restricted to use the REST API that it offers us.

Maximum number of	Number of active	Monthly

[3] *The initials of IFTTT mean "IF This, Then That". It refers to the fact that it is a page with which you can create and schedule actions between different applications.*

messages per device and month	devices per day	price
1000 per month	5 per day	Free

- **Thinger.io**

It is an open source platform; it has a very simple programming and a library in the official Arduino repository. It is compatible with ESP8266, ESP32, Raspberry Pi or Intel Edison.

It has a fairly powerful administration console where we can manage and geolocate our connected objects. Furthermore, the documentation is very extensive and free.

Maximum number of web services connections	Historic time (months) Quick access and statistics	Number of devices	Monthly price
10	30 days	3	Free

- **ThingSpeak**

ThingSpeak is MathWorks, the creators of MathLabs, for the Internet of Things. It is exclusively focused on building IoT applications. It allows to store data, visualize it and expose it to other APIs. This application and API is open source.

The base of this platform are the channels. It stores the data and it is made up of 3 elements:

- ➔ 8 fields to store data of any type.
- ➔ 3 fields to store the location, latitude, longitude and elevation. Of course, I would need a component that would give me this information.
- ➔ 1 field to store the state.

- ➔ Each of these fields can be updated every 15 seconds.

Within the code there are libraries for many programming languages. It has a very extensive documentation with examples and is completely free.

5.4.2. Final platform choice

After comparing them, it has been decided to choose ThingSpeak as the platform that will be used.

The main reason is due to the tutor's recommendation and the ease it will be since he is already familiar with it. Moreover, it is completely free to use compared to the others, that they work with a free plan. Besides, the features ThingSpeak has are more than sufficient for the project.

Having all the main components needed decided, the assembly of the project can go ahead.

6. PRACTICAL PART

6.1. Requirements

Hardware requirements:

- MQ7 (CO)
- MQ4 (Half)
- MQ131 (Ozone)
- MQ135 (CO2)
- ESP32 DevkitV1
- Arduino
- Bread board insert template
- Resistances
- Female-male and male-male cables
- 5V power supply
- Ammeter

Software requirements:

- Arduino IDE
- Embedded C Language

Incidentally, the power supply adapter and ammeter are lent by the school.

6.2. Setting up the sensors and power supply

Before programming with the ESP32, the tutor recommended to use the Arduino board, as it would be easier to check if all the gas sensors were working properly. Therefore, before the final "assembly", it will be tested trough the Arduino.

The first thing it has been done is to set up the power supply adapter. This serves to reduce the number of volts coming out of the electric current of the enclosure to 5 volts, which is the maximum that the Arduino can withstand.

Three male-male cables to “COM” have been connected, which is the negative current, and three more wires to “+ V”, which is the positive current.
Three of them have been placed because they are too thin and one alone could not carry all the load well.

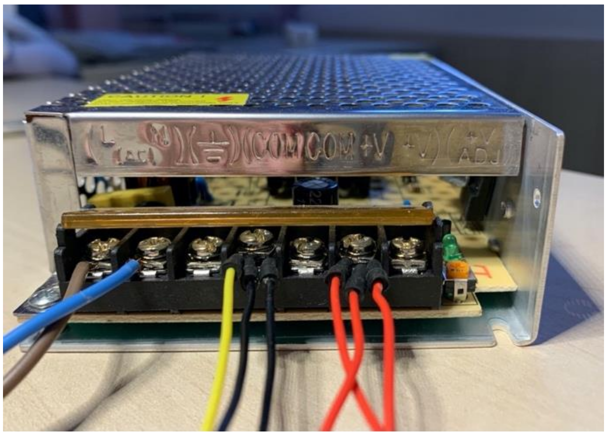


Figure 51: Power supply with three positive and negative cables

The other side of the wires have been connected to the protoboard as follows:

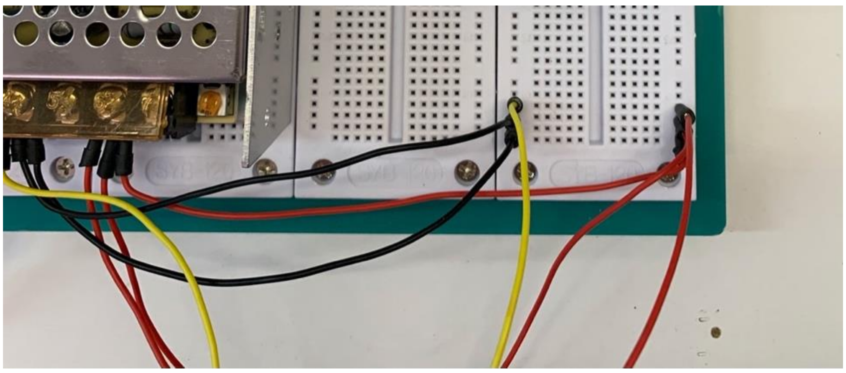


Figure 52: Connection of the power supply with the bread board

Then connected the sensors to the protoboard.

A white cable for analog input (AO), a red one (or orange on the MQ-135 sensor) for positive current (VVC) and a black one for negative (GND) have been used.

The following image shows the MQ-4 sensor connection:

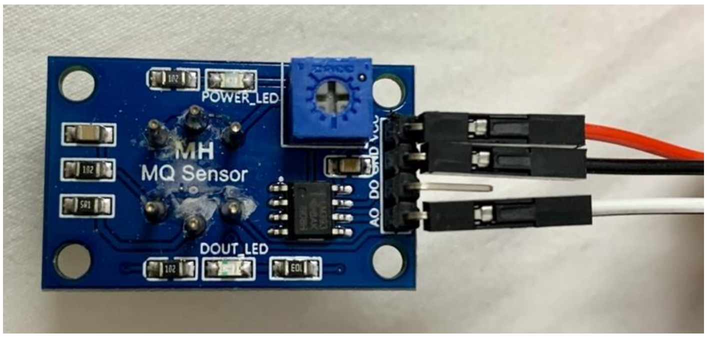


Figure 53: MQ-4 sensor connection

The positive and negative male wires have been connected to the prootboard and the analog input to the Arduino's A0, A1, A2 and A3 pins.

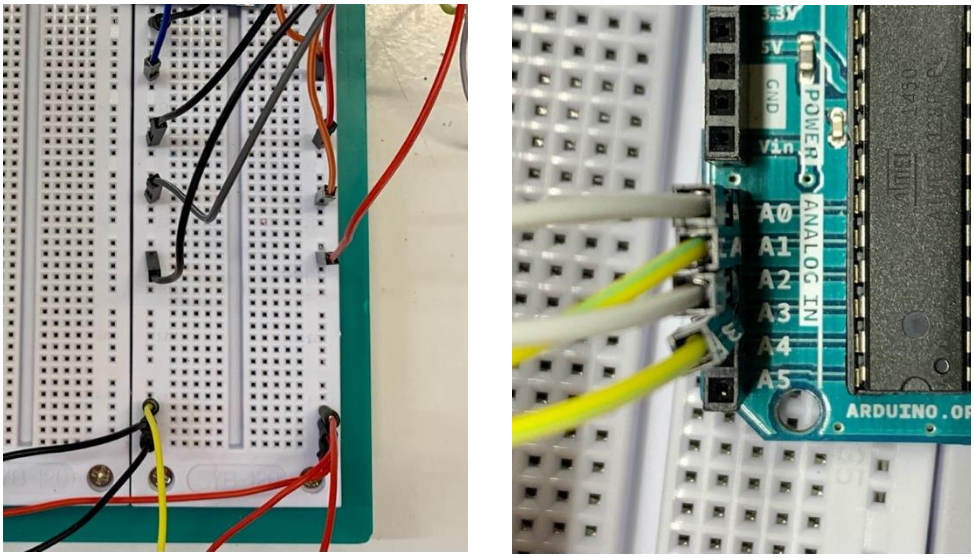


Figure 54: Connection of the four MQ sensors with the protoboard

At the same time, a case for each sensor has been chosen and printed with the 3D printer from the school, this being the result:

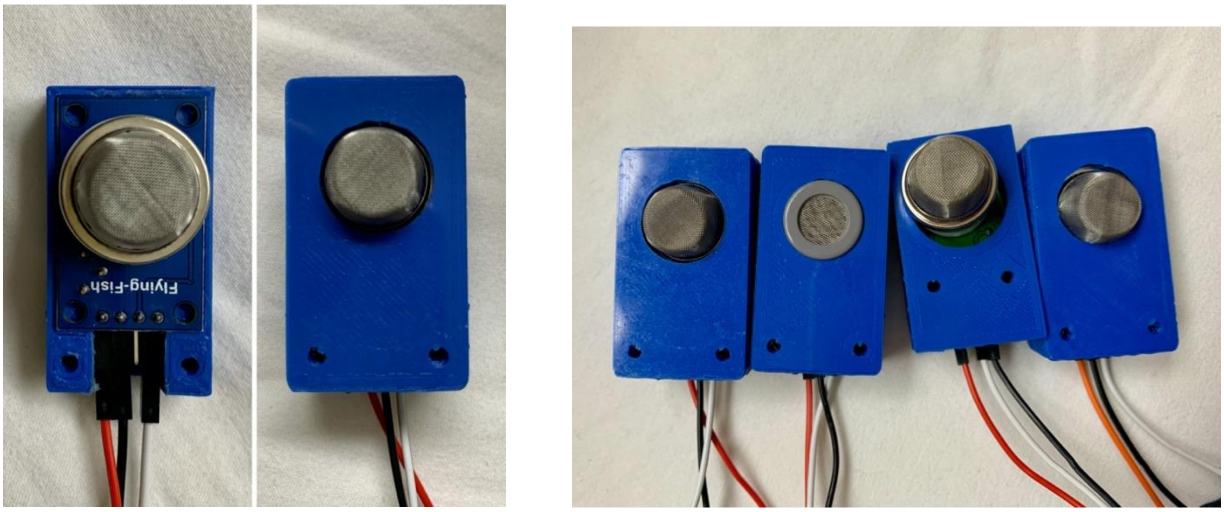


Figure 55: MQ sensors with the 3D printed case

Once assembled, all that's left is to run the Arduino program created, relying on a program already created by a guide to help connect a gas sensor to the Arduino board.

To complete the program and be able to calculate the concentration of the gases, apart from "fixing" it to match sensors, the sensor resistance (Rs), a necessary equation had to be found.
This is not in the technical documentation because the function that relates the gas concentrations to the voltage obtained at the outlet is not linear.

To find it, it has been looked for the technical file of each sensor, in the part where there is a graph of its sensitivity at a temperature of 20 · C, humidity of 65%, O2 concentration of 21% and the RL of 20k in all but the MQ-7.

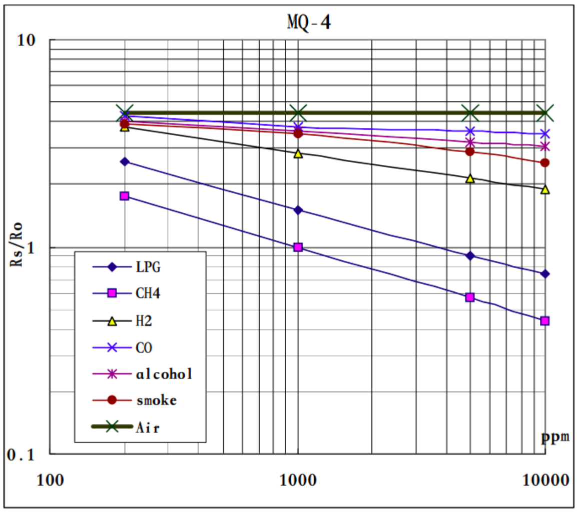


Figure 57: Sensitivity characteristics of the MQ-4

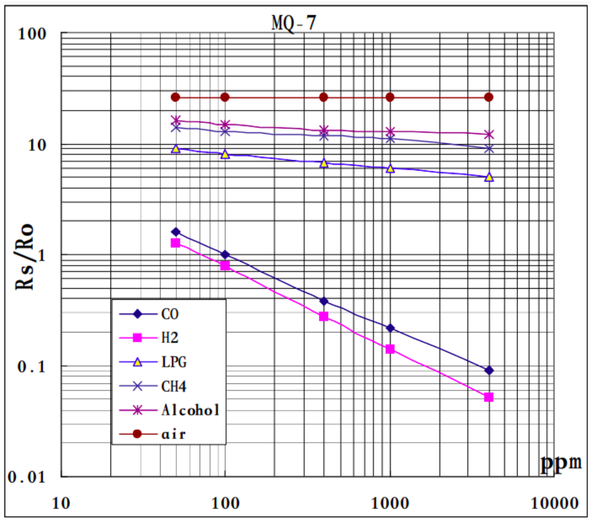


Figure 56: Sensitivity characteristics of the MQ-7

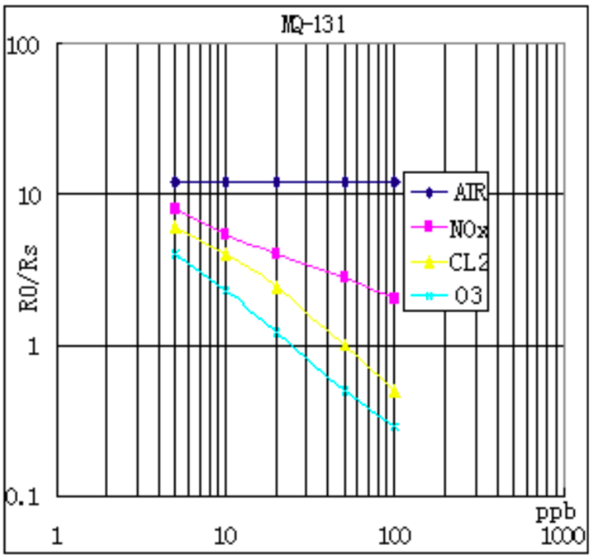


Figure 59: Sensitivity characteristics of the MQ-131

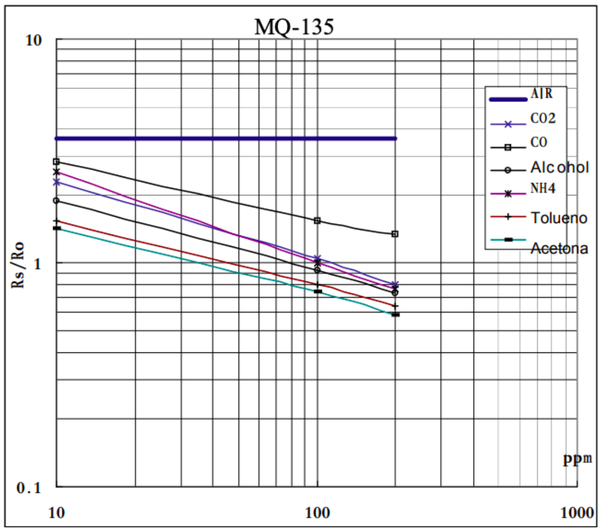
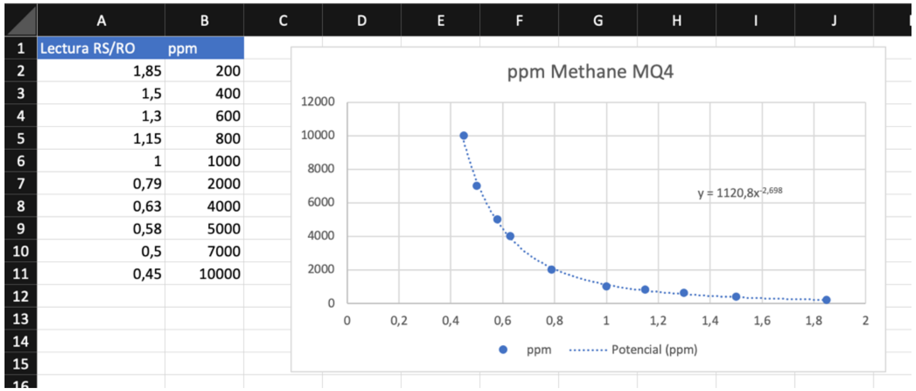


Figure 58: Sensitivity characteristics of the MQ-135

In an excel it has been noted the data in the graph. On the one hand the RS / RO Reading and on the other hand the parts per million (ppm).

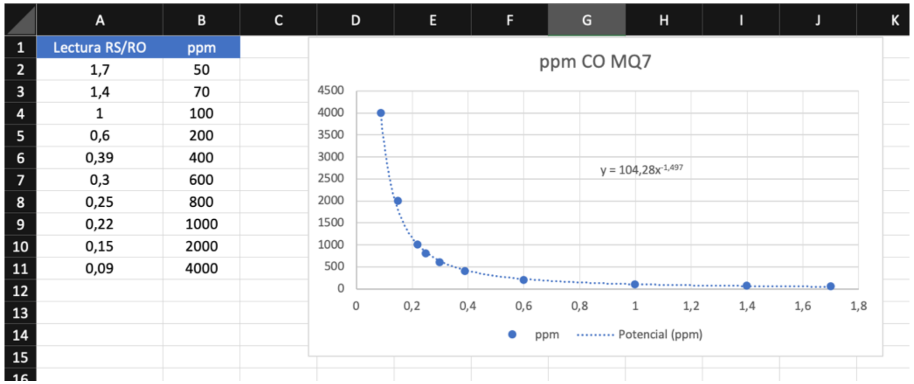
The graph has been inserted and added the potential type trend line and the graph equation.

That's how Rs has been found.



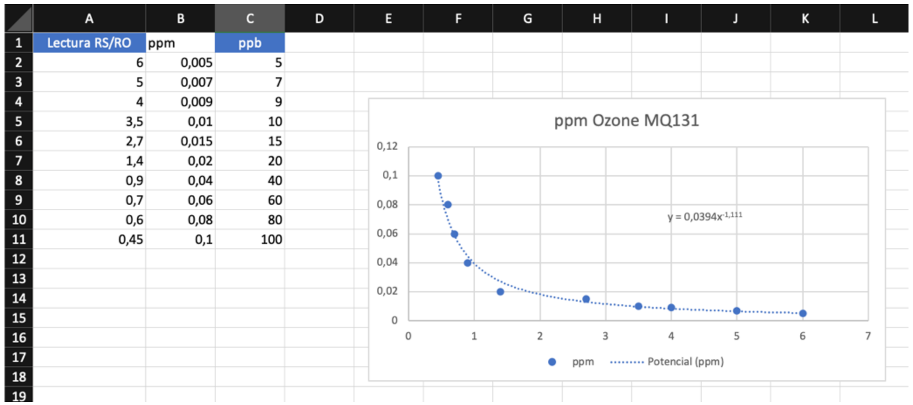
Lectura RS/RO	ppm
1,85	200
1,5	400
1,3	600
1,15	800
1	1000
0,79	2000
0,63	4000
0,58	5000
0,5	7000
0,45	10000

Figure 62: Methane sensitivity characteristics of MQ-4



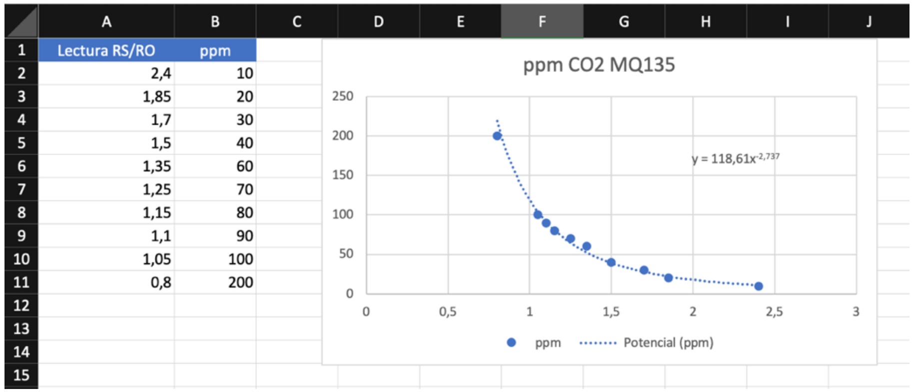
Lectura RS/RO	ppm
1,7	50
1,4	70
1	100
0,6	200
0,39	400
0,3	600
0,25	800
0,22	1000
0,15	2000
0,09	4000

Figure 61: CO sensitivity characteristics of MQ-7



Lectura RS/RO	ppm	ppb
6	0,005	5
5	0,007	7
4	0,009	9
3,5	0,01	10
2,7	0,015	15
1,4	0,02	20
0,9	0,04	40
0,7	0,06	60
0,6	0,08	80
0,45	0,1	100

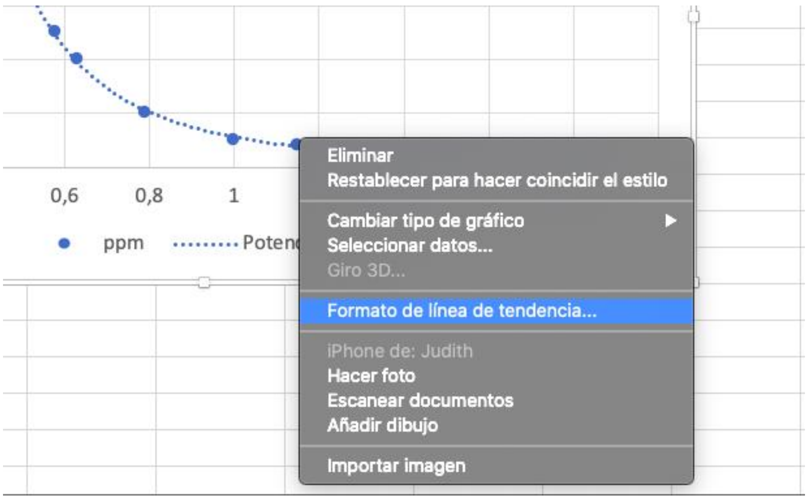
Figure 63: Ozone sensitivity characteristics of MQ-131



Lectura RS/RO	ppm
2,4	10
1,85	20
1,7	30
1,5	40
1,35	60
1,25	70
1,15	80
1,1	90
1,05	100
0,8	200

Figure 60: CO2 sensitivity characteristics of MQ-135

Here's the process of how the trend line and equation have been addeed:



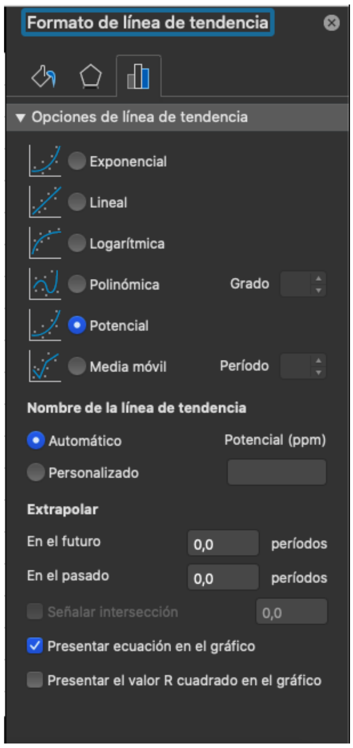


Figure 64: Process of making a trend line on excel

Finally, the program has been as follows:

```
//mq4, mq7, mq131, mq135
void setup() {
  Serial.begin(9600);
}

void loop() {

  int adc_MQ4 = analogRead(A0); // We read the analog
output of the MQ4
  float voltage1 = adc_MQ4 * (5.0 / 1023.0); // We convert
the reading into a voltage value
  float Rs1=1000*((5-voltage1)/voltage1);  // We calculate
Rs with an RL of 1k
```

```
  double methane=1120.8*pow(Rs1/5463, -2.698); // we
calculate the alcohol concentration with the equation
obtained.

// ------- We send the values via the serial port ---------
---
/*Serial.print("adc:");
  Serial.print(adc_MQ4);
  Serial.print("      voltage:");
  Serial.print(voltaje1);
  Serial.print("      Rs:");
  Serial.print(Rs1);
  */Serial.print("     Methane:");
  Serial.print(methane);
  Serial.println("ppm");
  delay(1000);

//We do the same process with the remaining sensors

   int adc_MQ7 = analogRead(A1);
  float voltage2 = adc_MQ7 * (5.0 / 1023.0
  float Rs2=1000*((5-voltage2)/voltage2);
  double CO=104.28*pow(Rs2/5000, -1.497);

 /*Serial.print("adc:");
  Serial.print(adc_MQ7);
  Serial.print("     voltage:");
  Serial.print(voltage2);
  Serial.print("      Rs:");
  Serial.print(Rs2);
  */
  Serial.print("     Carbon monoxide:");
  Serial.print(CO);
  Serial.println("ppm");
```

```
  delay(1000);



  int adc_MQ131 = analogRead(A2);
  float voltage3 = adc_MQ131 * (5.0 / 1023.0);
  float Rs3=1000*((5-voltage3)/voltage3);
  double ozone=0.0394*pow(Rs3/5000, -1.111);

 /*Serial.print("adc:");
  Serial.print(adc_MQ131);
  Serial.print("     voltage:");
  Serial.print(voltaje3);
  Serial.print("      Rs:");
  Serial.print(Rs3);
  */
  Serial.print("     Ozone:");
  Serial.print(ozono);
  Serial.println("ppm");
  delay(1000);



  int adc_MQ135 = analogRead(A3);
  float voltage4 = adc_MQ135 * (5.0 / 1023.0);
  float Rs4=1000*((5-voltage4)/voltage4);
  double CO2=118.61*pow(Rs4/5000, -2.737);

  /*Serial.print("adc:");
  Serial.print(adc_MQ135);
  Serial.print("     voltage:");
  Serial.print(voltage4);
  Serial.print("      Rs:");
  Serial.print(Rs4);
 */ Serial.print("     Carbon Dioxide:");
  Serial.print(CO2);
```

```
  Serial.println("ppm");
  Serial.println("");
  delay(1000);
}
```

Figure 65: First draft program made to ensure the sensors are working

When the program was updated to Arduino, it had been found that it only receives data from one sensor, the MQ-4 exactly.

The tutor, realises that it is because the protoboard does not work as it was thought, where each row of the protoboard is connected, but on this case they are not. Faced with this small problem, he has put a few bridges between each positive and negative cable so that the 5V current reaches all the sensors. After adding them, all the sensors worked perfectly.

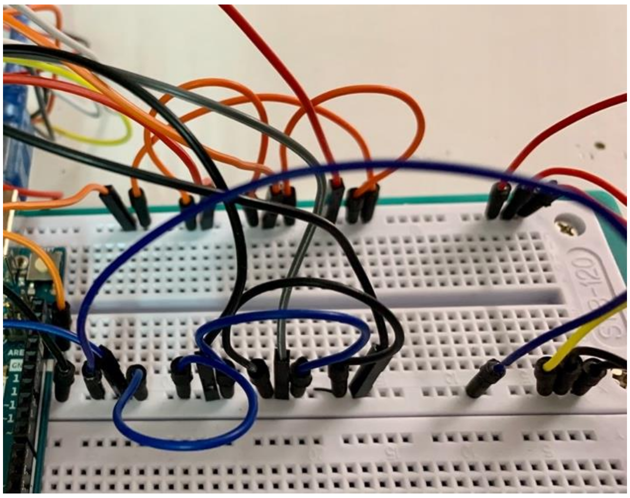


Figure 66: Bridges made to connect the separated sections of the bread board

Arduino's job of making sure everything works is done. Now it's time to work with the final board: ESP32 WROOM-32 (Devkit V1), the one that allows data to be sent over Wi-fi.

6.3. Connection of ESP32 to Wi-Fi and ThingSpeak

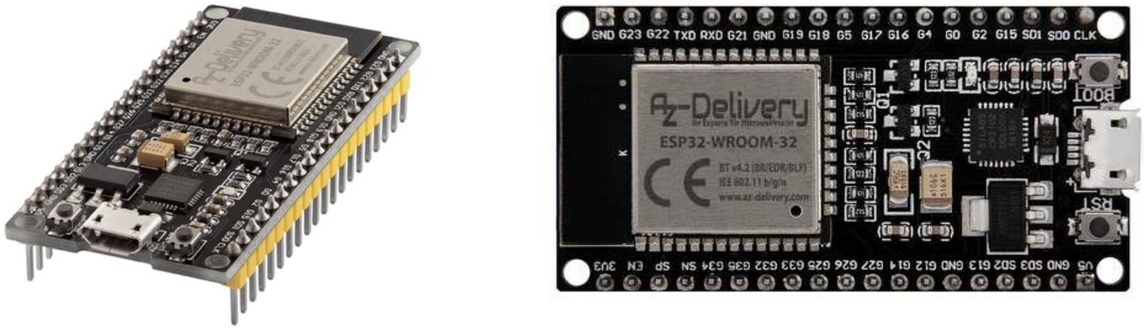


Figure 67: AZ-Delivery model sample of ESP-32-WROOM-32

Before placing it on the protoboard along with the gas sensors, it is necessary to check that it works properly. Which means that it has to connect properly to the wifi and is able to send random data (for now) to the ThingSpeak platform.
It has been done by connecting ESP32 to the computer via a micro-usb cable and from a program.

In order to use ThingSpeak it had to be downloaded to Arduino IDE, the platform's communication library from the internet.

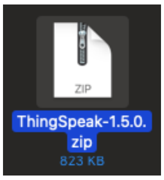


Figure 68: ThingSpeak library for Arduino IDE

In addition, ESP32 board had to be installed so that the computer would recognize it when connected.
This has been done by going to Arduino> Preferences
To “URL addicionals de gestor de plaques” added:
https://dl.espressif.com/dl/package_esp32_index.json

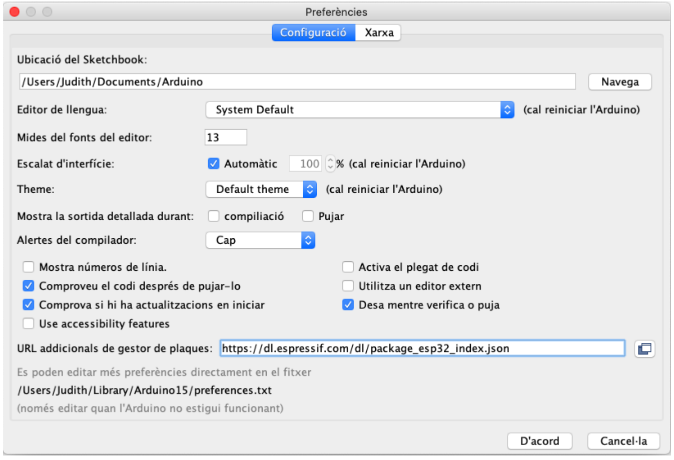


Figure 69: "Preferences" in Arduino IDE settings

Then opened Tools> Plate> Plate Manager ...

Searched for "esp32" and installed it.

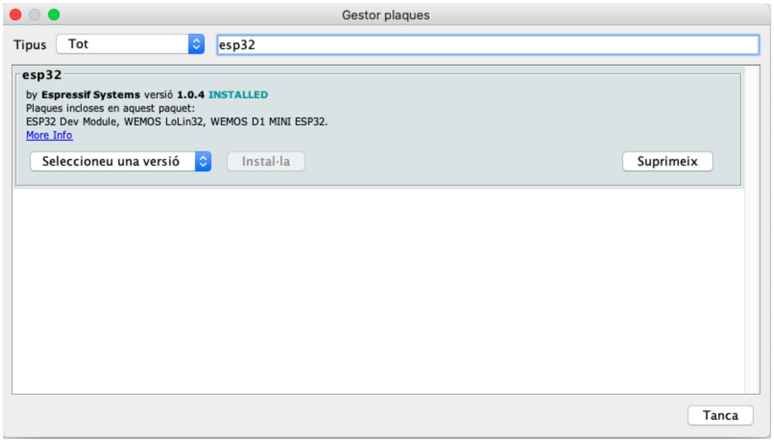


Figure 70: Installation of ESP32 library

And so, DOIT ESP32 DEVKIT V1, the board used, has been selected.

Figure 71: Selection of the board used

Finally, a required driver for it to work, CP210x USB to UART Bridge VCP Drivers, has been downloaded.

The following programming includes the ThingSpeak platform with the Channel number and API Key previously created on the same platform. Obviously, the Wi-fi name and password (SSID) of the site where the project is being developed is also included. In this case, the same program had to be duplicated. One for working at home and one for school. In order not to paste the same program to the document twice, it is clarified that what it will be seen below is what the home Wi-fi contains.

At the moment checking that the data is sent correctly to ThingSpeak without connecting the sensors to the ESP32 is the objective, four fields with the name of the gases that want to be studied have been created.
If it works, both in the serial monitor and in ThingSpeak, random numbers from 0 to 100 will appear in each field, except for the "Methane" field which will start with the number 0 in order. Which means the graph seen in Methane's ThingSpeak will be the only linear one.

This is how the program that tests the connection between ESP32 and ThingSpeak looks like:

```
#include <ThingSpeak.h>
#include "secrets.h"
#include <WiFi.h>


char ssid[] = "vodafone83AA";   // name of the SSID net
char pass[] = "******";   // password of the net
int keyIndex = 0;            // your network key Index
number (needed only for WEP)
WiFiClient  client;

unsigned long myChannelNumber = 1205761;
const char * myWriteAPIKey = "0VSM6WBB3KYC1ZR9";

// Initialize our values
```

```
int Methane = 0;
int CO = random(0,100);
int Ozone = random(0,100);
int CO2 = random(0,100);
String myStatus = "";

void setup() {
  Serial.begin(115200);  //Initialize serial port

  WiFi.mode(WIFI_STA);
  ThingSpeak.begin(client);  // Initialize ThingSpeak
}

void loop() {

  // Connect or reconnect to WiFi
  if(WiFi.status() != WL_CONNECTED){
    Serial.print("Intentando conectar a SSID: ");
    Serial.println(SECRET_SSID);
    while(WiFi.status() != WL_CONNECTED){
      WiFi.begin(ssid, pass);  // conection to net
WPA/WPA2.
      Serial.print(".");
      delay(5000);
    }
    Serial.println("\nConnected.");
  }

  // set the fields with the values
  ThingSpeak.setField(1, Methane);
  ThingSpeak.setField(2, CO);
  ThingSpeak.setField(3, Ozone);
  ThingSpeak.setField(4, CO2);


  // set the status
  ThingSpeak.setStatus(myStatus);

  // write to the ThingSpeak channel
  int x = ThingSpeak.writeFields(myChannelNumber,
myWriteAPIKey);
  if(x == 200){
    Serial.println("Channel actualizado correctmente.");
    Serial.println(CO);
```

```
    Serial.println(Ozone);
    Serial.println(CO2);
    Serial.println(Methane);

  }
  else{
    Serial.println("Problemas al actualizar el channel. 
HTTP error code " + String(x));
  }

  // change the values
  Methane++;
  if(Methane > 99){
    Methane = 0;
  }
  CO = random(0,100);
  Ozone = random(0,100);
  CO2 = random(0,100);

  delay(20000); // Wait 20 seconds to update the channel 
again
}
```
Figure 72: Second draft program made to ensure ESP32 connects to Wi-Fi and consecutively to ThingSpeak

Once the board has been connected to the internet, it has been possible to visualize the data on the Arduino IDE serial monitor and on ThingSpeak.

Figure 73: Connection process shown in Ardiono IDE the serial monitor

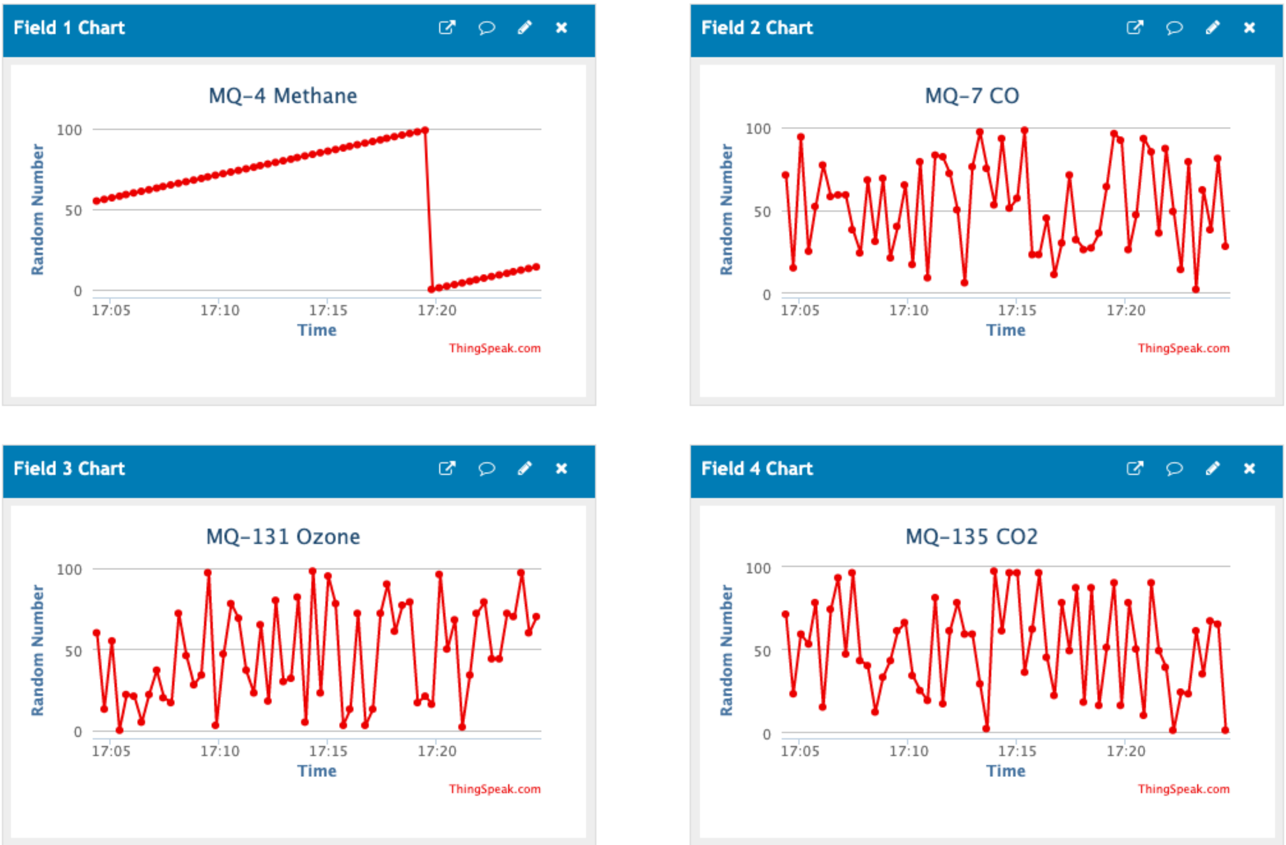


Figure 74: Graphics on ThingSpeak that indicate aleatory numbers for each sensor

Now all the components work have been checked, the final assembly can be done.

Final assembly

For the final programming, both programs created above have been merged, as well as some new lines have been added, such as the connection of the four sensors to the pins of the ESP32. Voltage reading of the sensors in order to have more information for being able to identify errors easily.

Owing to the fact that ESP32 will use Wi-Fi to send the data collected by the sensors, there are some specific pins that cannot be used as input when using Wi-Fi. These pins are the ADC2. On the other hand, ADC1 pins can be used when Wi-Fi is in use.

So, after taking a look at the data sheet of the company that made the board, it has been decided to connect the sensors to pins G32, G33, G34 and G35, as shown in the following program.

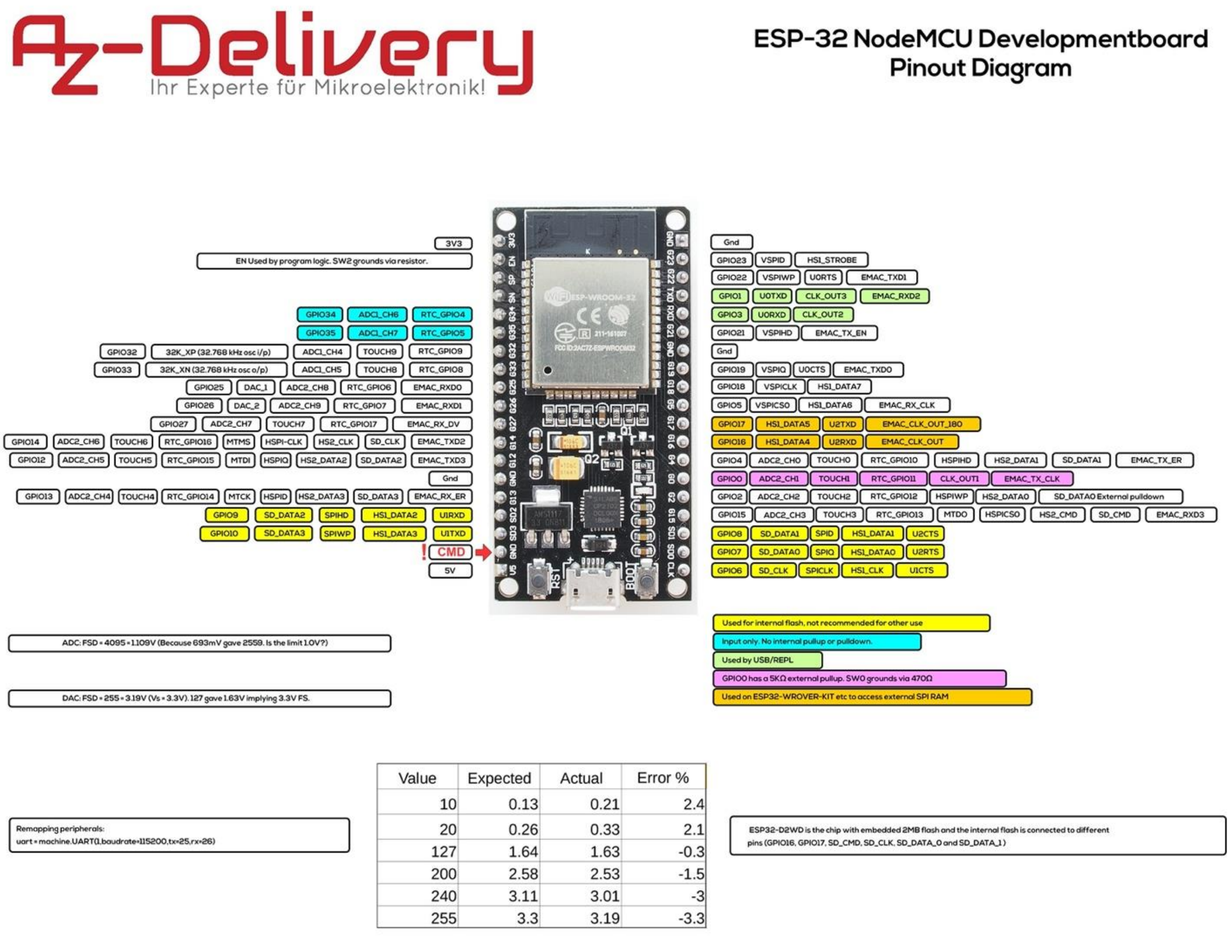


Figure 75: AZ-Delivery ESP32-WROOM32 Datasheet

Final assembly program:

```
#include <ThingSpeak.h>
#include "secrets.h"
#include <WiFi.h>

//mq4, mq7, mq131, mq135
int sensor_metano=34;
int sensor_CO=35;
int sensor_ozono=32;
int sensor_CO2=33;

char ssid[] = "vodafone83AA";   // name of the SSID net
char pass[] = "******";   // password of the net
int keyIndex = 0;
WiFiClient  client;

unsigned long myChannelNumber = 1205761;
const char * myWriteAPIKey = "0VSM6WBB3KYC1ZR9";



void setup() {
  Serial.begin(115200);  //Initialize serial port

  WiFi.mode(WIFI_STA);
  ThingSpeak.begin(client);  // Initialize ThingSpeak
}

void loop() {

  // Connect or reconnect to WiFi
  if(WiFi.status() != WL_CONNECTED){
    Serial.print("Trying to connect to SSID: ");
    Serial.println(SECRET_SSID);
    while(WiFi.status() != WL_CONNECTED){
      WiFi.begin(ssid, pass);  // conection to net
WPA/WPA2.      Serial.print(".");
      delay(5000);
    }
    Serial.println("\nConnected.");
```

```
  }


  int adc_MQ4 = analogRead(sensor_metano); //Read MQ4
analagog output
  float voltaje1 = adc_MQ4 * (3.3 / 4095); //Convert
lecture to a voltatge
  float Rs1=1000*((3.3-voltaje1)/voltaje1);  //Calculate Rs
with 1k RL
  float Methane=1120.8*pow(Rs1/5463, -2.698); //Calculate
methane concentration with the equation obtained

  int adc_MQ7 = analogRead(sensor_CO); // Read MQ7 analagog
output
 float voltaje2 = adc_MQ7 * (3.3 / 4095);
  float Rs2=1000*((3.3-voltaje2)/voltaje2);
  float CO=104.28*pow(Rs2/5000, -1.497);  //suposedly
resistance R0 5000

  int adc_MQ131 = analogRead(sensor_ozono); // Read MQ131
analagog output
  float voltaje3 = adc_MQ131 * (3.3 / 4095);  float
Rs3=1000*((3.3-voltaje3)/voltaje3);
  float Ozone=0.0394*pow(Rs3/5000, -1.111);

  int adc_MQ135 = analogRead(sensor_CO2); // Read MQ135
analagog output
  float voltaje4 = adc_MQ135 * (3.3 / 4095);
  float Rs4=1000*((3.3-voltaje4)/voltaje4);
  float CO2=118.61*pow(Rs4/5000, -2.737);


  // set the fields with the values
  ThingSpeak.setField(1, Methane);
  ThingSpeak.setField(2, CO);
  ThingSpeak.setField(3, Ozone);
  ThingSpeak.setField(4, CO2);


  // set the status
  //ThingSpeak.setStatus(myStatus);

  // write to the ThingSpeak channel
```

```
  int x = ThingSpeak.writeFields(myChannelNumber,
myWriteAPIKey);
  if(x == 200){
    Serial.println("Channel actualizado correctmente.");
    Serial.println(Methane);
    Serial.println(CO);
    Serial.println(Ozone);
    Serial.println(CO2);

    Serial.println("");


    Serial.println(voltaje1);
    Serial.println(voltaje2);
    Serial.println(voltaje3);
    Serial.println(voltaje4);

  }
  else{
    Serial.println("Problemas al actualizar el channel.
HTTP error code " + String(x));
  }



  delay(20000); // Wait 20 seconds to update the channel
again
}
```

Figure 76: Final program that will detect gases and upload the data to ThingSpeak in real time

Please note that ESP32 is 3.3V, which means that a voltage split must be made as the power supply that is used is 5V.

To find out what kind of resistance is needed to make this division, it has been relied on this formula.

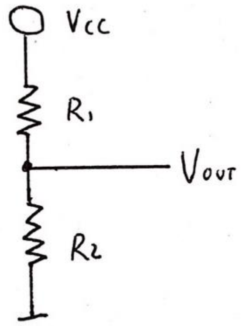


Figure 77: Scheme of a general resistance division

$$\boxed{\frac{V_{OUT}}{V_{CC}} = \frac{R_2}{R_1 + R_2}}$$

$$V_{OUT} = \frac{R_2}{R_1 + R_2} \cdot V_{CC}$$

$$3{,}3 = \frac{R_2}{R_1 + R_2} \cdot 5V$$

(The resistance in a voltage divisor is linear, that is, $V_{OUT} \to R_2;\ V_{CC}R_1 + R_2$)

For R_1 it will be set to 10K since this is a highly common value. For that $R_1 = 10k\ \Omega$

$$3{,}3 \cdot 10k + 3{,}3R_2 = 5R_2 \to$$
$$33k + 3{,}3R_2 = 5R_2 \to$$
$$33k = (5 - 3{,}3)R_2 \to$$
$$33k = 1{,}7R_2 \to$$
$$R_2 = \frac{33000}{1{,}7} \to$$
$$\boxed{R_2 = 19411{,}76 \sim 20.000\Omega}$$

Therefore, using 10,000 Ω resistors would be fine. However, in the package of resistors owned strangely don't come with this type, so it had to be used the resistance closest to this value instead, 8kΩ. These will also work well.

In this case, the resistance voltage will be the same as the 10k one, but the 8k ones will have a higher consumption.

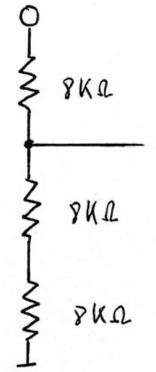


Figure 78: Scheme of the resistance division used

Then the sensors were connected the sensors with their respective resistors.

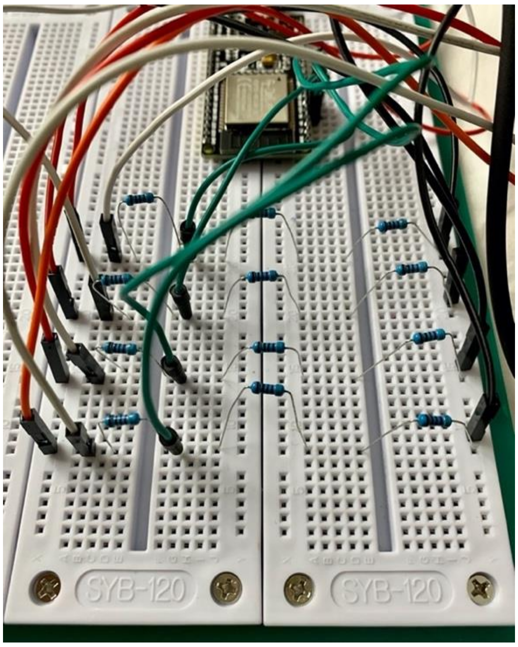


Figure 80: Sensors connection and resistance colocation

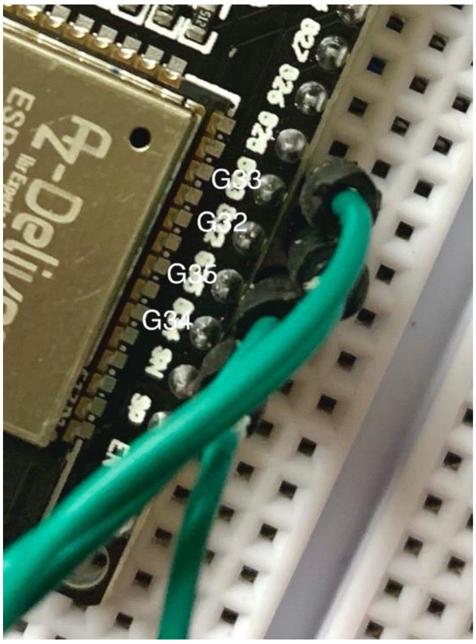


Figure 79: GPIO pins used to connect each gas sensor

And finally, after the protoboard has been connected to the power supply and some bridges have been added in order to ensure the continuity of the circuit, the negative row (black cables) has been connected from the ESP32 board to the ground (GND), resulting in this assembly:

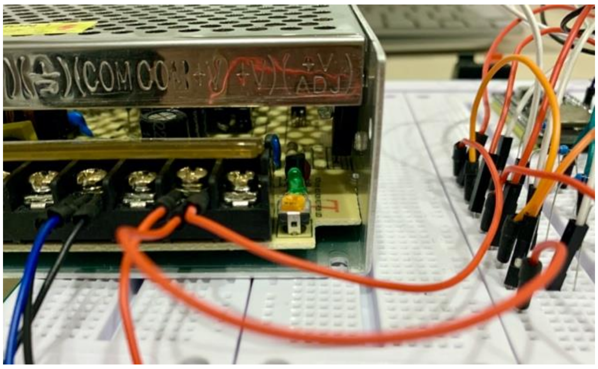


Figure 82: Positive column connection

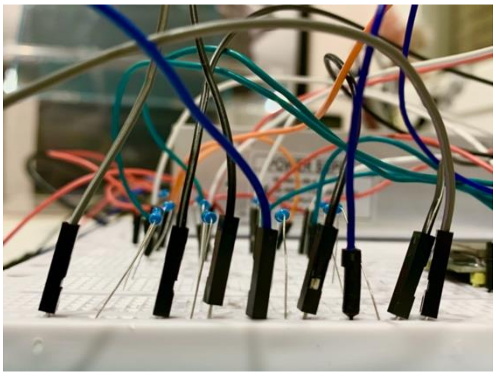


Figure 81: Negative column connection

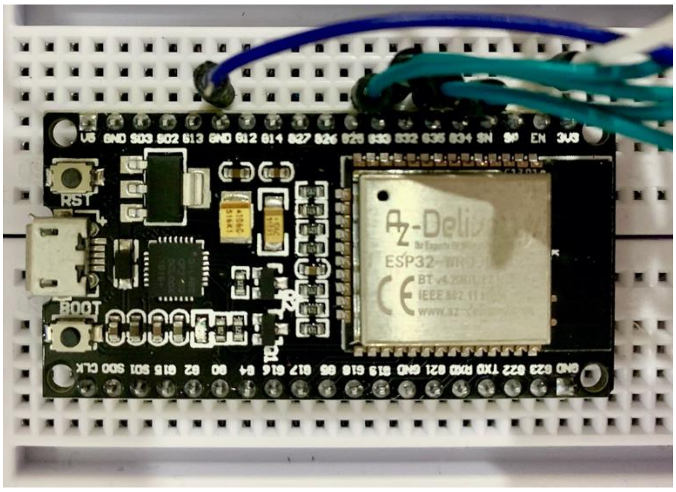


Figure 83: ESP32 connection with the sensors and the power supply

Needless to say, during the assembly some problems and occurrences have been encountered:

i. The last step mentioned was not taken; the connection of the negative row with GND.
ii. There was a bad connection between the cables and the breadboard.
iii. Since there MQ7 is also highly sensitive to LPG and methane is a liquefied gas, when methane is sprayed out, MQ7 sends a high number.

After noticing the first (i) and second problem (ii) and quite few attempts, modifications and reviewing the assembly, the gas detection started working properly.

Setting up an alarm to Twitter through ThingSpeak

In order to receive an alert when a specific number is equalled or exceeded, ThingTweet has been used to do this work. ThingTweet is a ThingSpeak option which allows users to link their account to their selected Twitter account. It creates an API key that can be used later when programming.

Despite that, programming is not strictly necessary due to the fact that ThingSpeak has another option called “React”. It allows to create different reactions towards specific events detected.

Therefore, four different reactions have been created in purpose of receiving a Twitter alert when the events occur.

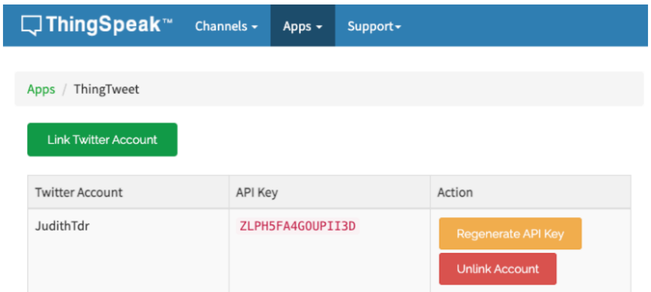


Figure 85: Linking Twitter account to Thingspeak

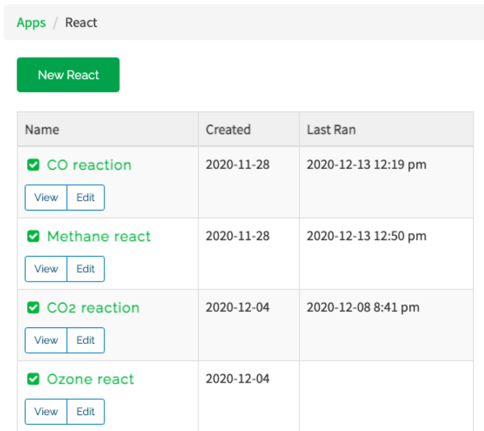


Figure 84: Reactions for each senosrs

This is how the settings of the CO react look like. All the reacts are generally the same except for the triggering number, the message, and obviously the field number/name.

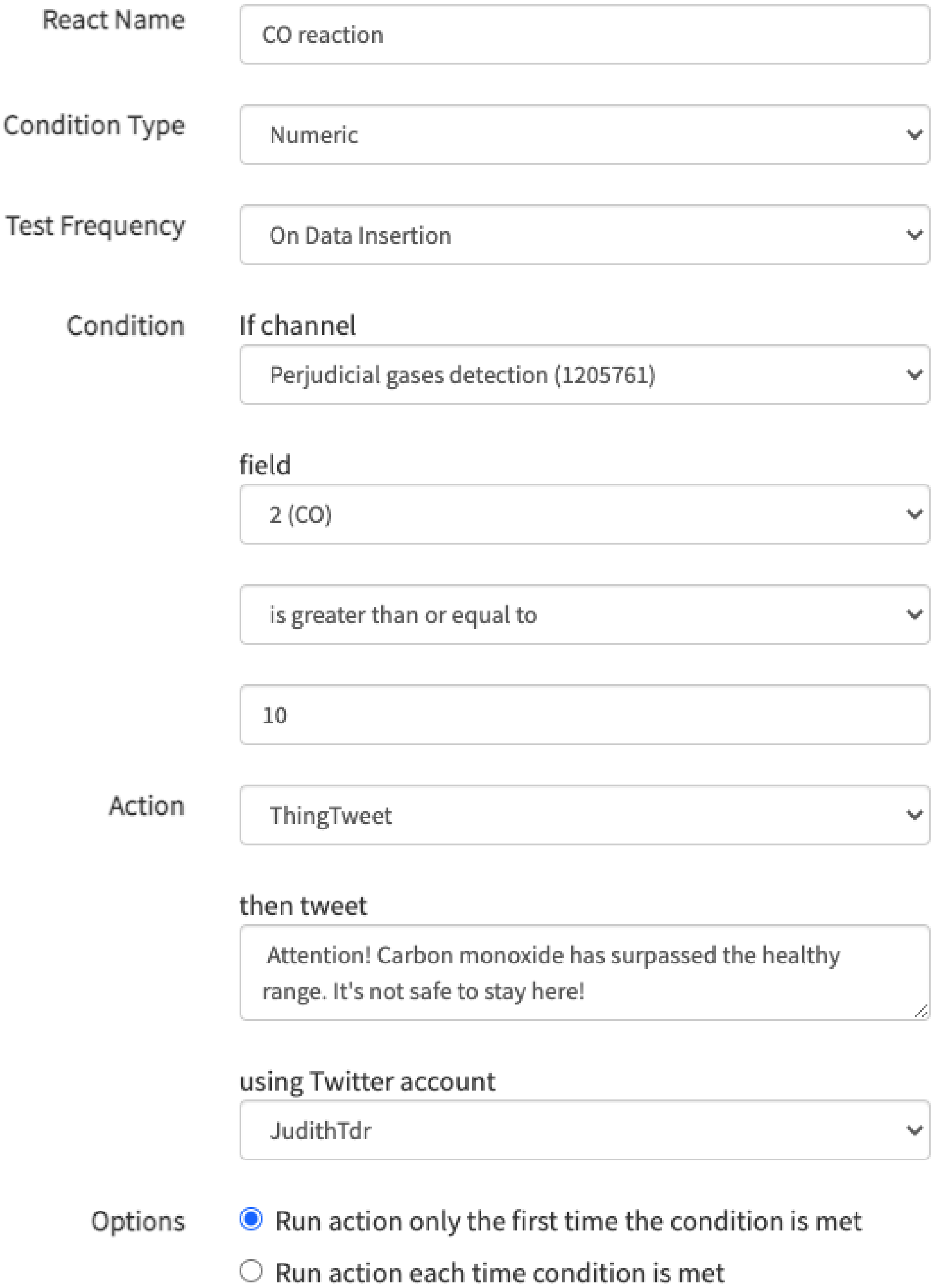


Figure 86: React of MQ-7 (CO) sensor

Experimentation

Several tests have been carried out to ensure the system remains connected to the Internet and triggers the alerts set out earlier.

The system has been left working during several hours and all the data has been sent to the internet, therefore ThingSpeak successfully.

A screenshot of the graphics provided by ThingSpeak of the MQ-4 and MQ-7 sensors is what is seen below.

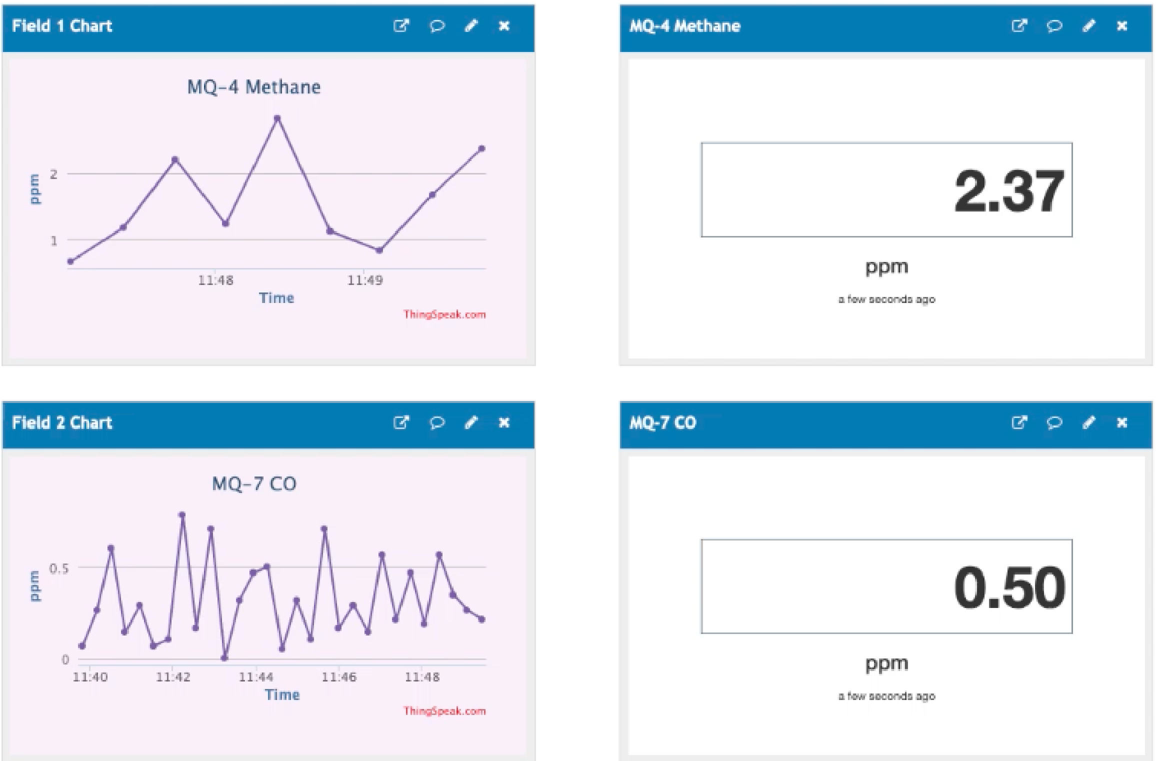


Figure 87: Graphics of the gas's ppm detected by the MQ-4 and MQ-7 sensors

In order to trigger the alert, gases have been generated from the combustion of a phosphorus and a methane spray.

The gases are thus measured by sensors and the alert is triggered by a certain concentration of those.

In the figure below, the alerts that were generated during the different days of experimentation can be observed.

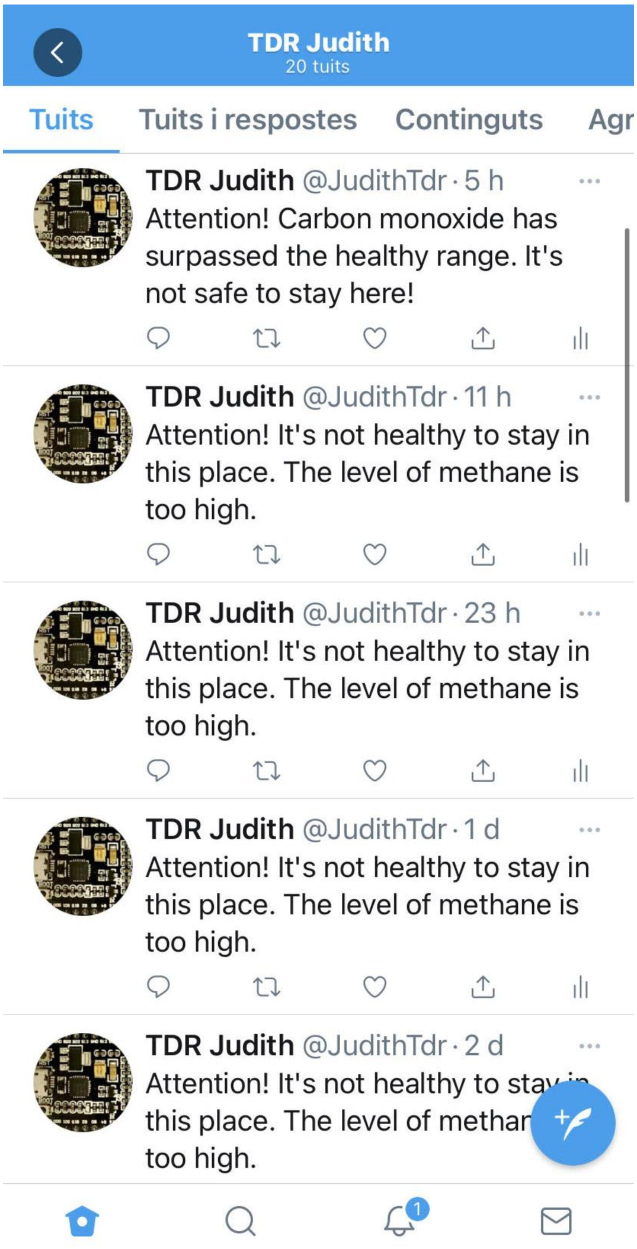


Figure 88: Alerts triggered during different experimentations

Video

This QR code links to a video created in aims of showing the functioning of the gas monitoring system based on the IoT.
Please scan the QR code in order to be able to watch it.

CONCLUSION

In this research, the development of a gas detector monitoring platform based on the IOT is presented. Experiments have been carried out to verify the connection and the transmission of data.

All the objectives initially proposed have been achieved, which are as follows:

1. It is demonstrated that the project is an example of a system using IoT since the data collected by the sensors are successfully transmitted to the ThingSpeak platform cloud.
2. The transmission of gas concentration data in real time allows the activation of the alarm that warns the user in case of overexposure.

In order to improve the project, the calibration of sensors would be studied so as to obtain more accurate gas concentration data. Therefore, it can be considered that the data obtained by the system are truly accurate. After that, a statistical work would be carried out to make an evaluation of the air quality in a particular area. The system for visualising gas sensor measurements would also be developed. A platform or application that allows a more direct and dynamic visualization of the data for the user would be created. In addition, an automation system would be worked on by means of an alert, such as turning on a large or several smaller LEDs at the same time as symbolising the alert or the danger of remaining in that place.

At present, Lleida is fortunate not to have a high level of contamination. Though on the other hand, in our neighbouring city of Barcelona, pollution is one of the main issues that is already starting to have harmful effects on their citizens.

Even so, the nearby cities should not take it easy since, given the global circumstances, it appears that we will not have to wait long for the unwanted. Therefore, measures must be taken in advance to prevent pollution while protecting ourselves from it.

This paper is therefore a small introduction to how new technology can contribute to this significant and complex issue that everyone is involved in.

Webography

ALL ABOUT ELECTRONICS. (2017). *Difference between Microprocessor and Microcontroller*
https://www.youtube.com/watch?v=dcNk0urQsQM
ANALYTICAL TECHNOLOGY. (2019) *How Do Electrochemical Sensors Work?*
https://www.analyticaltechnology.com/analyticaltechnology/gas-water-monitors/blog.aspx?ID=1327&Title=How%20Do%20Electrochemical%20Sensors%20Work
ANDREWS, Evan. (Original: 2013. Updated: 2019). *Who Invented the Internet?*
https://www.history.com/news/who-invented-the-internet
ARDUINO PLAYGROUND. *MQ Gas sensors*
https://playground.arduino.cc/Main/MQGasSensors/
AVSYSTEM. (Original: 2019; Updated: 2020): *What is an Internet of Things Platform?*
https://www.avsystem.com/blog/what-is-internet-of-things-platform/
AZZOLA, Francesco. (2018). *12 IoT Platforms for Building IoT Projects*
https://dzone.com/articles/12-iot-platforms-for-building-iot-projects
BAHGA & MADISETTI. (2015). *Introduction to IoT*
http://speed.cis.nctu.edu.tw/~ydlin/course/cn/iot/slides/Chapter-1.pdf
BHAGAT, Varun. (2019). *What are Pros and Cons of Internet of Things?*
https://www.pixelcrayons.com/blog/what-are-pros-and-cons-of-internet-of-things/
BROMLEY, Doug. (Original: 2018. Updated:2019). *MQ Gas Sensors List*
https://www.strawdogs.co/2018/06/MQ-Gas-Sensors-List/
CALIENDO, C.; VERONA E.; D'AMICO, A.. (1992). *Surface Acoustic Wave (SAW) Gas Sensors*
https://link.springer.com/chapter/10.1007/978-94-011-2737-0_8
CAVICCHI, Richard E.. (2011). *Calorimetric Sensors*
https://www.nist.gov/publications/calorimetric-sensors
CHOUDHARY, Abhishek. (2017). *What does MQ stand for in MQ gas sensors?*
https://www.quora.com/What-does-MQ-stand-for-in-MQ-gas-sensors
COMPONENTS 101. (2019). *Introduction to Gas Sensors: Construction Types and Working*
https://components101.com/articles/introduction-to-gas-sensors-types-working-and-applications
COMSOL Multiphysics. (2019). *Surface Acoustic Wave Gas Sensor*
https://www.comsol.com/model/surface-acoustic-wave-gas-sensor-2129
D. FOOTE, Keith. (2016). *A Brief History of the Internet of Things*
https://www.dataversity.net/brief-history-internet-things/#
DATAFLAIR TEAM. (2018). *How IoT Works – 4 Main Components of IoT System*
https://data-flair.training/blogs/how-iot-works/
DOLAN, Shelagh. (2019). *The technology, devices, and benefits of remote patient monitoring in the healthcare industry*
https://www.businessinsider.com/remote-patient-monitoring-industry-explained?IR=T
DOMODESK. *A fondo: ¿Qué es IoT (el internet de las cosas)?*
https://www.domodesk.com/221-a-fondo-que-es-iot-el-internet-de-las-cosas.html
EGHAM. (2019). *Gartner Says 5.8 Billion Enterprise and Automotive IoT Endpoints Will Be in Use in 2020*
https://www.gartner.com/en/newsroom/press-releases/2019-08-29-gartner-says-5-8-billion-enterprise-and-automotive-io

ELIZALDE, Daniel. (2016). *What is an IoT Platform? (And How to Choose One)*
https://danielelizalde.com/iot-platform/
ESP. ESP32 Features and Specifications
http://esp32.net/
EXPRESSIF. (Updated: 2020). *ESP32 Series Datasheet*
https://www.espressif.com/sites/default/files/documentation/esp32_datasheet_en.pdf
FERNANDEZ, Ariz B. *Gas sensor and transmitter*
https://www.slideserve.com/toviel/gas-sensor-and-transmitter
GODDARD, William. (2019). *History of IoT: What It Is, How It Works, Where It's Come From, and Where It's Going*
https://itchronicles.com/iot/history-of-iot-what-it-is-how-it-works-where-its-come-from-and-where-its-going/
GRIZHNEVICH, Alex. (2018) *A comprehensive guide to IoT-based predictive maintenance*
https://www.scnsoft.com/blog/iot-predictive-maintenance-guide
GYARMATHY, Kaylie. (2020). *Comprehensive Guide to IoT Statistics You Need to Know in 2020*
https://www.vxchnge.com/blog/iot-statistics?hs_amp=true
HAMBURG, Steve. *Methane: The other important greenhouse gas*
https://www.edf.org/climate/methane-other-important-greenhouse-gas
HARSHVARDHAN, Mishra. (2020). *MQ Gas Sensors | Gas Sensors ?*
https://iotbyhvm.ooo/mq-gas-sensors-gas-sensors/
HARWOOD, Trevor. (2019). *Internet of Things (IoT) History*
https://www.postscapes.com/iot-history/
HEK. (Original: 2017. Updated: 2020). *Gas Sensor*
https://www.mysensors.org/build/gas
HERNÁNDEZ, Luis del Valle. (2016). *Proyectos IoT con Arduino, las plataformas más importantes*
https://programarfacil.com/podcast/proyectos-iot-con-arduino/
HILL, Lauren. (2012). *The 9 Most Common Air Pollutants*
https://www.business2community.com/health-wellness/the-9-most-common-air-pollutants-0342288
HONNUR, Prakash. (2019). *IoT Levels and Deployment Templates*
https://www.slideshare.net/pakku009/iot-levels-and-deployment-templates
IOT FOR ALL. (2020). *An introduction to IoT*
https://www.iotforall.com/ebooks/an-introduction-to-iot
I-SCOOP. (2018). *What is an IoT Platform? (And How to Choose One)*
https://www.i-scoop.eu/internet-of-things-guide/iot-platform-market-2017-2025/
JOSE, Derik. (2014). *5 Levels of Intelligence in Industrial IOT*
https://www.linkedin.com/pulse/20140828073038-3175536-5-levels-of-intelligence-in-industrial-iot
JOSE, Derik. (2015). *5 Levels of maturity in IOT intelligence*
https://www.slideshare.net/judederick/5-levels-of-iot-intelligence-49545717
KAA PROJECT. (2020). *What is an IoT platform?*
https://www.kaaproject.org/what-is-iot-platform
KHVOYNITSKAYA, Sandra. (2019). *The history and future of the internet of things*
https://www.itransition.com/blog/iot-history
KING'S COLLEGE LONDON. (2018). *What are the components of an IoT product?*
https://www.futurelearn.com/courses/internet-of-things/0/steps/8437

KOYANAGI, Fernando. (2017). Introduction to ESP32
https://www.instructables.com/id/Introduction-to-ESP32/
LASSE LUETH, Knud. (2014). *Why the Internet of Things is called Internet of Things: Definition, history, disambiguation*
https://iot-analytics.com/internet-of-things-definition/
LOOKING FOR SOLUTIONS, (2018*). Send a Tweet Alert based on your sensor data condition*
https://lookingforsolutions.com/blogs/news/send-a-tweet-alert-based-on-your-sensor-data
MATHWORKS. *React App Documentation*
https://es.mathworks.com/help/thingspeak/react-app.html
McCLELLAN, Calum. (2020). *What is an IoT Platform?*
https://www.iotforall.com/what-is-an-iot-platform/
MSV, Janakiram. (2016). *10 DIY Development Boards for IoT Prototyping*
https://thenewstack.io/10-diy-development-boards-iot-prototyping/
MUÑOZ FERRARA, Javier. (2018). *Basic components to design a successful IoT project*
https://iotsens.com/basic-components-to-deisgn-successful-iot-project/
PARK, Nam-hee; AKAMATSU, Takafumi; ITOH, Toshio; IZU, Noriya; SHIN, Woosuck. (2014). *Calorimetric Thermoelectric Gas Sensor for the Detection of Hydrogen, Methane and Mixed Gases*
https://www.ncbi.nlm.nih.gov/pmc/articles/PMC4063010/
PERENIO. (2019). *The History of the Internet of Things*
https://perenio.com/blog/the-history-of-the-internet-of-things
PROGRAMO ERGO SUM. *¿Qué es GPIO?*
https://www.programoergosum.com/cursos-online/raspberry-pi/238-control-de-gpio-con-python-en-raspberry-pi/que-es-gpio
QIAO, Kurt. (2015) *Intel Edison Board Bluetooth control GPIO*
http://kurtqiao.github.io/iot/2015/06/29/Intel-Edison-Board-Bluetooth-control-GPIO.html
RAFFERTY, John P. (Updated: 2020). *Methane*
https://www.britannica.com/science/methane
RASPBERRY PI. (Updated: 2020). *Raspberry Pi 4-Model B Specifications*
https://www.raspberrypi.org/products/raspberry-pi-4-model-b/specifications/
REDDY SANNAPUREDDY, Bhaskara. (2015). *Pros & Cons of Internet Of Things (IOT)*
https://www.linkedin.com/pulse/pros-cons-internet-things-iot-bhaskara-reddy-sannapureddy
ROGOVA, Galina; SHAHBAZIAN, Elisa. (2015). *Meeting Security ChallengesTthough Data Analytics and Decision Support*
https://books.google.es/books?id=_BKhDQAAQBAJ&pg=PA273&lpg=PA273&dq=intel+edison+wifi+certification+countrie&source=bl&ots=njDTDQ8uvy&sig=ACfU3U2isfvRWjV0A6g0b3kK_S5Tqg-20Q&hl=ca&sa=X&ved=2ahUKEwjp17W2z_fpAhVmxoUKHUZjAQoQ6AEwAHoECAoQAQ#v=onepage&q=intel%20edison%20wifi%20certification%20countrie&f=false
RUNBERG, Derek. (2017). *A Guide to Development Boards for Educators: Microprocessors vs. Microcontrollers*
http://blog.sparkfuneducation.com/a-guide-to-development-boards-for-educators-microprocessors-vs-microcontrollers
SANDIA NATIONAL LABORATORIES. (2009). *High Temperature Acoustic Wave Gas Sensors*

https://www.sandia.gov/mesa/_assets/documents/Fact_Sheets/sensors/2High_Temp_Acoustic.pdf
TECH TERMS. (2012). *Processor*
https://techterms.com/definition/processor
THE AUDIOPEDIA. (2016). *What is a microcontroller?*
https://www.youtube.com/watch?v=CIRCvq53a-g
TUTORIALS POINT. (2015). *IoT - Environmental Monitoring*
https://www.tutorialspoint.com/internet_of_things/internet_of_things_environmental_monitoring.htm
UTSOURCE. (2018). *What is a development board and what is it used for?*
https://www.youtube.com/watch?v=JrO2LziLXtM&t=56s
WANG, Chengxian; YIN, Longwei; ZHANG, Luyuan; XIANG, Dong; GAO, Rui. (2010). *Metal Oxide Gas Sensors: Sensitivity and Influencing Factors*
https://www.ncbi.nlm.nih.gov/pmc/articles/PMC3264469/
WEIL, Nick. (2012). *Q&A: What is a development board?*
http://www.nickweil.com/2012/01/q-what-is-development-board.html
WIKIPEDIA. (Updated: 2020). *Carbon monoxide*
https://en.wikipedia.org/wiki/Carbon_monoxide
WIKIPEDIA (Updated: 2020). *Microprocessor development board*
https://en.wikipedia.org/wiki/Microprocessor_development_board
WIKIPEDIA. (Updated: 2020). *Ozone* https://en.wikipedia.org/wiki/Ozone
WIKIPEDIA. (Updated: 2020). *Sulfur dioxide: As an air pollutant*
https://en.wikipedia.org/wiki/Sulfur_dioxide#As_an_air_pollutant

Table of figures

Figure 1: RFID tags 9
https://twitter.com/kevin_ashton
Figure 2: Kevin Ashton 9
https://www.vecteezy.com/vector-art/130720-rfid-vector
Figure 3: Key evolution..10
https://www.idcband.com/blog/rfid-hotel-wristband-keycards-from-wallet-to-wrist/

Figure 4: LeWeb Paris 2012 10
https://leweb.co/
Figure 5: IoT Endpoint Market by Segment, 2018-2020, Worldwide (Installed Base, Billions of Units)..11
https://www.gartner.com/en/newsroom/press-releases/2019-08-29-gartner-says-5-8-billion-enterprise-and-automotive-io
Figure 6: Nikola Tesla, 1926...11
Figure 7: Electromagnetic telegraph by Baron Schilling, 1832 11
Figure 8: World Wide Web 12
Figure 9: Marshall McLuhan, 1964 12
Figure 10: Trojan Room Coffee Pot, 1993...13
Figure 11: John Romkey, 1990 13
Figure 12: Mark Weisler's article, 1991 13
https://www.postscapes.com/iot-history/ (from 6 to 12)
Figure 13: IoT process 14
https://data-flair.training/blogs/iot-tutorials-home/

Figure 14: Sensors 14
https://iotsens.com/basic-components-to-deisgn-successful-iot-project/
Figure 15: Data analytics..15
https://depositphotos.com/36190487/stock-illustration-web-and-seo-analytics-concept.html
Figure 16: User interface 15
https://iotsens.com/basic-components-to-deisgn-successful-iot-project/
Figure 17: IoT system level 1 18
Figure 18: IoT system level 2 18
Figure 19: IoT system level 3 19
Figure 20: IoT system level 4 19
Figure 21: IoT system level 5 20
Figure 22: IoT system level 6 20
https://profile.iiita.ac.in/bibhas.ghoshal/IoT_2019/Lecture_Slides/Chapter-1_Introduction.pdf (from 17 to 22)
Figure 23: Libelium Smart World 23
http://www.libelium.com/libelium-smart-world-infographic-smart-cities-internet-of-things/
Figure 24: Ozone...24
https://en.wikipedia.org/wiki/Ozone
Figure 25: Carbon Monoxide 24
https://en.wikipedia.org/wiki/Carbon_monoxide
Figure 26: Carbon Dioxide..25
https://en.wikipedia.org/wiki/Carbon_dioxide#Toxicity

Figure 27: Nitrogen Oxides...25
http://nbrienvis.nic.in/Database/1_2039.aspx
Figure 28: Methane 25
https://en.wikipedia.org/wiki/Methane

Figure 29: Methane cycle 26
https://www.britannica.com/science/methane

Figure 30: Gas sensors 27
https://components101.com/articles/introduction-to-gas-sensors-types-working-and-applications
Figure 31: Metal Oxide based gas sensors 27
http://www.alphasense.com/index.php/products/metal-oxide-sensors-2/

Figure 32: Metal Oxide gas sensor cycle 28
https://components101.com/articles/introduction-to-gas-sensors-types-working-and-applications
Figure 33: List of MQ gas sensors..29
https://www.mysensors.org/build/gas
Figure 34: Optical gas sensor...29
https://es.aliexpress.com/item/32672336586
Figure 35: Electrochemical gas sensor 30
https://www.indiamart.com/proddetail/electrochemical-gas-sensor-19863884888.html
Figure 36:Side view of a single capacitor in micro machine capacitive polymer gas sensor 31
https://www.researchgate.net/figure/Schematic-setup-of-a-capacitive-sensor-measuring-changes-in-the-permittivity-e-of-a_fig2_233901079
Figure 37: Capacitance-based gas sensor functioning..31
https://slideplayer.com/slide/14867274/ (slide 39)
Figure 38: Calorimetric gas sensor..31
Figure 39: Acoustic wave based gas sensor 32
https://slideplayer.com/slide/5719410/ (38 and 39)
Figure 40: Arduino component distribution..34
https://www.jameco.com/jameco/workshop/circuitnotes/cn-arduino-uno.html
Figure 41: Arduino UNO 35
https://store.arduino.cc/arduino-uno-rev3
Figure 42: Raspberry Pi 4..36
https://www.raspberrypi.org/products/
Figure 43: Raspberry Pi 4 component distribution..36
https://www.raspberrypi.org/products/raspberry-pi-4-model-b/specifications/

Figure 44: Raspberry Pi GPIO pin distribution..36
https://www.programoergosum.com/cursos-online/raspberry-pi/238-control-de-gpio-con-python-en-raspberry-pi/que-es-gpio
Figure 45: Intel Edison 37
https://software.intel.com/content/www/us/en/develop/articles/get-started-edison-windows.html
Figure 46: Intel Edison component distribution...37
http://kurtqiao.github.io/iot/2015/06/29/Intel-Edison-Board-Bluetooth-control-GPIO.html

Figure 47: ESP8266 38
https://www.amazon.com/-/es/JacobsParts-ESP8266-Microcontrolador-desarrollo-MicroPython/dp/B0787LMTZD
Figure 48: ESP8266 pin distribution 38
https://support.infocenter.gr/product/#

Figure 49: ESP32 39
https://tienda.bricogeek.com/arduino-compatibles/1274-esp32-wroom-wifi-bluetooth.html?gclid=CjwKCAjwyo36BRAXEiwA24CwGWwXkKUFdtLgDldos3ZjmRqWxsi_y3CK8XEc9iat7U94hYE2wnl3GRoCkGkQAvD_BwE

Figure 50: ESP 32 general pin distribution 39
https://randomnerdtutorials.com/esp32-pinout-reference-gpios/
Figure 51: Power supply with three positive and negative cables 46
Figure 52: Connection of the power supply with the bread board..........................46
Figure 53: MQ-4 sensor connection 47
Figure 54: Connection of the four MQ sensors with the protoboard 47
Figure 55: MQ sensors with the 3D printed case...48
Figure 57: Sensitivity characteristics of the MQ-4 49
Figure 56: Sensitivity characteristics of the MQ-7 49
Figure 58: Sensitivity characteristics of the MQ-135 49
Figure 59: Sensitivity characteristics of the MQ-131 49
Figure 60: CO2 sensitivity characteristics of MQ-135...50
Figure 61: CO sensitivity characteristics of MQ-7...50
Figure 63: Methane sensitivity characteristics of MQ-4 50
Figure 62: Ozone sensitivity characteristics of MQ-131 50
Figure 64: Process of making a trend line on excel 51
Figure 65: First draft program made to ensure the sensors are working 54
Figure 66: Bridges made to connect the separated sections of the bread board...54
Figure 67: AZ-Delivery model sample of ESP-32-WROOM-32 55
https://www.az-delivery.de/en/products/esp32-developmentboard
Figure 68: ThingSpeak library for Arduino IDE 55
Figure 69: "Preferences" in Arduino IDE settings 56
Figure 70: Installation of ESP32 library 56
Figure 71: Selection of the board used..56
Figure 72: Second draft program made to ensure ESP32 connects to Wi-Fi and consecutively to ThingSpeak...59
Figure 73: Connection process shown in Ardiono IDE the serial monitor..............59
Figure 74: Graphics on ThingSpeak that indicate aleatory numbers for each sensor 60
Figure 75: AZ-Delivery ESP32-WROOM32 Datasheet 61
https://cdn.shopify.com/s/files/1/1509/1638/files/ESP_-_32_NodeMCU_Developmentboard_Datenblatt_AZ-Delivery_Vertriebs_GmbH_10f68f6c-a9bb-49c6-a825-07979441739f.pdf?v=1598356497
Figure 76: Final program that will detect gases and upload the data to ThingSpeak in real time...64
Figure 77: Scheme of a general resistance division 65
Figure 78: Scheme of the resistance division used 66
Figure 79: GPIO pins used to connect each gas sensor 66
Figure 80: Sensors connection and resistance colocation 66
Figure 81: Negative column connection 66
Figure 82: Positive column connection..66
Figure 83: ESP32 connection with the sensors and the power supply 67