الجمهـــوريــــــة الجزائـــريــــــة الديمقـــراطيــــة الشـعبيــــة
République Algérienne Démocratique et Populaire
وزارة التـــــعلـــيـــــــم العــــــــالــــــــــي والبــحـــــــــث العــلـمـــــــي
Ministère de l'Enseignement Supérieure et de la Recherche Scientifique

جامعة وهران 2 محمد بن أ حمد
Université d'Oran 2 Mohamed Ben Ahmed

معهد الصيانة و الأمن الصناعي
Institut de Maintenance et de Sécurité Industrielle

Département de Maintenance en Instrumentation

MÉMOIRE

Pour l'obtention du diplôme de Master

Filière : Génie Industriel
Spécialité : Ingénierie de la Maintenance en Instrumentation

Thème

Réalisation d'un système d'arrosage automatique supervisé par l'IdO(Internet des Objets)

Présenté et soutenu publiquement par :

Nom : Belhaouari Prénom : Mohamed Bilel

Devant le jury composé de :

Nom et Prénom	Grade	Etablissement	Qualité
TITAH Mawloud	MCB	IMSI-Univ. D'Oran2	**Président**
MEKKI Ibrahim El Khalil	MCA	IMSI-Univ. D'Oran2	**Encadreur**
BELKHODJA Leila	MCB	IMSI-Univ. D'Oran2	**Examinateur**

Année Universitaire : 2021-2022

الجمهـــوريــــة الجزائــــريــــة الديمقــراطيــــة الشــعبيــــة
People's Democratic Republic of Algeria
وزارة التــــعلـــيــــــم العــــــالـــــــي والبــــحـــــــث العـــلمـــــي
Ministry of Higher Education and Scientific Research
جامعة وهران 2 محمد بن أ حمد
University of Oran 2 Mohamed Ben Hmed

معهد الصيانة و الأمن الصناعي
Institute of Maintenance and Industrial Safety

Instrumentation Maintenance Department

Thesis

For obtaining a master's degree

Faculty : Industrial Engineering
Speciality : Instrumentation Maintenance Engineering

Theme

Making of an automatic irrigation system supervised by IoT(Internet of Things)

Presented and defended publicly by :

First Name : Belhaouari Last Name : Mohamed Bilel

In front of the jury Composed of :

First and Last name	Grade	Establishment	Quality
TITAH Mawloud	MCB	IMSI-Univ. Of Oran2	**President**
MEKKI Ibrahim El Khalil	MCA	IMSI-Univ. Of Oran2	**Supervisor**
BELKHODJA Leila	MCB	IMSI-Univ. Of Oran2	**Examiner**

University Year: 2021-2022

Acknowledgment

After bismiallah, i thank Allah the almighty. I thank my dear family for the continuous support over every year since birth, without them this work would not have been achieved, my friends whom without I wouldn't have the courage to move forward, I name Ayoub, Oussama, Abdou, Sedik, Amir, Rania, Hanaa.

To my professors of the institute of maintenance and industrial safety who taught me over the course of 5 years and helped me gain the knowledge to achieve this project.

I particularly thank my supervisor Mr.Mekki Ibrahim El Khalil for accepting to supervise my thesis and help me progress through it and encourage me, I thank him for being humble and available whenever needed.

Special thanks to the jury team for agreeing to examine my work and evaluate it , I hope they find this work satisfying.

Summary

Figures list

Tables list

General Introduction

General introduction:

Over the decades, satisfaction of the population over many aspects was one of the challenges facing any country, one of those aspects is nutrition and the solution to it was and still is the development of the agricultural sector.

Humankind practiced the agricultural activity ever since they inhabited earth; this activity never ceases to develop following the development of humanity and technology. In modern agriculture, we use high-tech (high technology) for many reasons such as high efficiency and available resources control

Without a question, an irrigation system is Key in an aspect such as agriculture for the sole reason that life grows from water, over time, irrigation systems varied and developed into multiple types, meant for different places, environments, purposes, and results.

In this work, we will focus on implementing a smart surface irrigation system, which can be accessed, monitored, and controlled from distance at any given time and place. The implementation is on this institute's greenhouse prototype scale.

This smart irrigation system requires the monitoring and control of the following settings:

- Temperature and air humidity: One of the most important variables to monitor, they give an insight into the environment's status, letting the user know if it is the right time to irrigate or not.
- Soil humidity: It is crucial to know when your soil needs irrigation and when it does not, this variable is important in order to monitor periodically the state of soil (dry or wet).
- Water level: Without the presence of water, an irrigation cannot be executed, this variable helps keep in track the level of water in a tank without dropping below a threshold, in order to avoid any cease of irrigation process, which would result in damaging the plants or trees.

New technologies such as electronic boards like Arduino and NodeMcu, which are used for different controls over various systems, offered the capability to monitor and control the settings we mentioned before.

One of the main goals of this work is to help the Algerian agricultural aspect prosper, help it implement new technologies that have been missing, with the recent orientations towards the Saharan Agriculture, proposing new solutions and technologies that haven’t been put to use yet will definitely help fasten the process of production and expedition.

As part of our work, we will be implementing a smart irrigation system piloted by NodeMCU V3 and monitored using Internet Of Things (Arduino IoT Cloud), we can check the temperature and humidity of both air and soil and water level using a web site or a mobile application anytime from anywhere.

This final thesis is made of four chapters:

- The first chapter carries a general study of irrigation systems, types of irrigation, irrigation system choosing factors, water requirements factors.
- The second chapter carries a general study of the NodeMCU Board, the Arduino IoT Cloud web site and mobile application, the components used.
- The third chapter carries the making of our system and its implementation in the IoT(Internet of Things) as well as the programing side.
- The fourth chapter carries the electronic aspect (wiring) and the finale test and results of the entirety of our system.

Chapter 1

Irrigation System

1.1. Introduction

Water, it is the first resource scientists look for when exploring space. Agriculture cannot and will not exist without the presence of water, but the presence itself is not sufficient, because each type of soil adapts to a certain quantity of water, this led humans throughout the time to invent and develop the idea of irrigation.

In this chapter, we will present a general idea on the irrigation system, types of irrigation, the necessary factors on which we choose the correct irrigation method and the required amount of water for the task, as well as the advantages and disadvantages of said methods.

1.2. Irrigation

Irrigation is the agricultural process of applying controlled amounts of water to land to assist in the production of crops, as well as to grow landscape plants and lawns, where it may be known as watering. Agriculture that does not use irrigation but instead relies only on direct rainfall is referred to as rain-fed. Irrigation has been a central feature of agriculture for over 5,000 years and has been developed independently by many cultures across the globe.

Irrigation helps to grow agricultural crops, maintain landscapes, and revegetate disturbed soils in dry areas and during periods of less than average rainfall [1].

1.3. Types of irrigation

After briefly talking about irrigation, it is necessary to mention and explain its types, we will summarize these types in the diagram below:

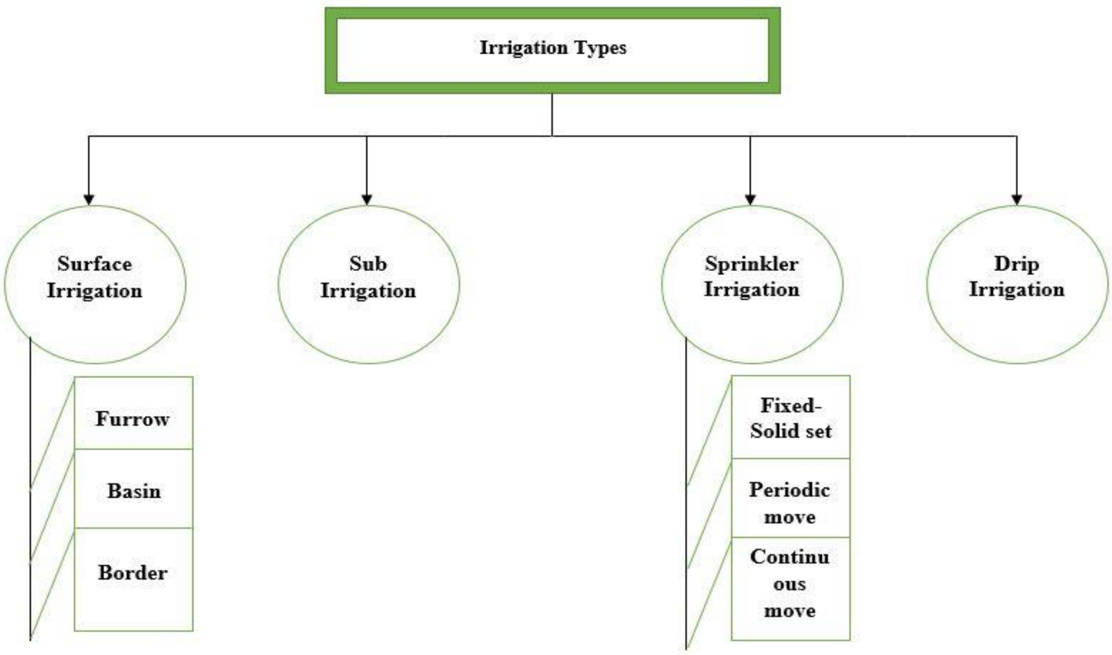


Figure 1-1 Irrigation types.

1.3.1. Surface Irrigation

"Surface irrigation" refers to a gravity-fed application of water to crops through a system of canals, dams, and furrows or basins that can be opened or blocked off as needed. Surface irrigation with basins is the most common irrigation method for fruit trees, and surface irrigation with furrows is the most common method for row crops. Currently, surface irrigation is the most commonly used type of irrigation system for agriculture, accounting for 85% of the world's irrigated land [2].

a) Surface Irrigation types and methods

➢ **Furrow Irrigation**

- Here water is irrigated to the field through the furrows spaced at 0.1 to 0.3 meters.
- It can be operated with less technical knowledge, whereas drip and sprinkler methods require technical knowledge.
- The furrow method also requires less capital investment. This regulation of water flow is difficult. It is suitable for crops like maize, sorghum which are sown in rows.
- In the fields with higher slope low infiltration rates and in fields with the lower slope, higher inflow rates are required to irrigate the field uniformly [2].
- Generally, siphon tubes are used to supply the water to the furrows from the ditches

Figure 1-2 Furrow Irrigation

- **Basin Irrigation**

It is one of the types of surface irrigation methods. In this, a basin, which is proportionate to the size of the tree is prepared. It is mostly used in orchards. After making basins all these are connected with each other and with irrigation channels. Thus, water moves from one basin to another basin by irrigating the previous basin. In this method of irrigation, there might be the tendency of contamination of pests and diseases from one tree to another through irrigation water [3].

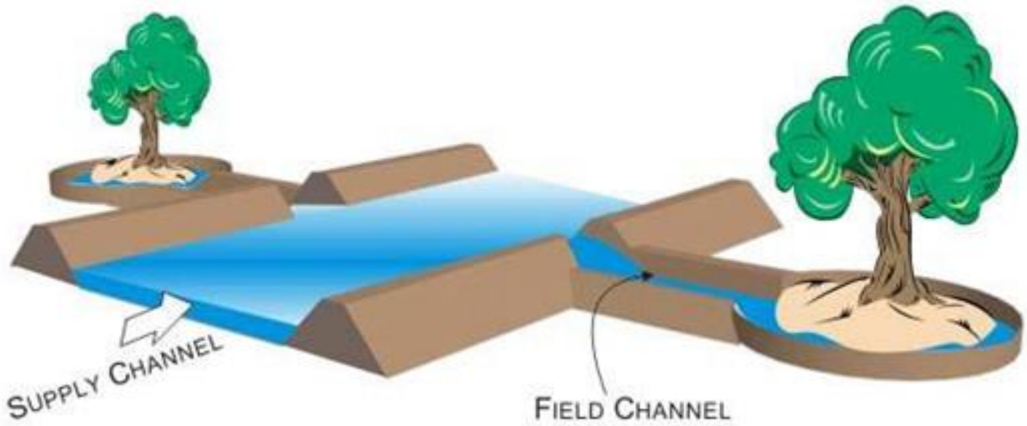


Figure 1-3 Basin Irrigation

- **Border Irrigation**

- It is an old method of irrigation. In this method, the field is divided into blocks, and borders are made with the help of the earth. In this method, water flows between dikes that divide the sloping field into rectangular strips with free drainage at an end. The purpose of the dikes is mainly to contain water as it flows across the field. This method is most suitable for the fields with slopes compared to the basin method [3].

Figure 1-4 Border Irrigation

b) Surface Irrigation advantages:

Followings are the surface irrigation advantages.

- Management is quite easy, you do not need any modern technology. If you have local traditional knowledge, you can do it.
- You do not need high financial support. You can be beneficial with small lands too.
- If you have short time water supplies, then this is the best process for you.
- If your drainage system is far, then you just need longer tubes.
- This is a nature-friendly system, you can utilize rainwater.
- It also works effectively in a low filtration rate.
- Low capital and no energy cost needed.
- You can use this irrigation process in sloping lands and long fields [4].

c) Surface Irrigation Disadvantages:

Followings are the surface irrigation disadvantages.

- Level lands require high accuracy, you cannot use it there.
- This is a big no-no for big fields.
- Not applicable on soil with a high filtration rate.
- Plants are always covered with water even when they do not need it.
- Sometimes limited space gets more water than required.
- No drainage outlet [4].

1.3.2. Sprinkler Irrigation

Sprinkler Irrigation is a method of applying irrigation water which provides a rainfall-like effect. Water is distributed through a system of pipes usually by pumping. It is then sprayed into the air and the entire soil surface is irrigated through spray heads so that it breaks up into small water droplets that fall on the ground.

Sprinklers provide efficient coverage for small to large areas and are suitable for use on all types of farmlands. It is also adaptable to nearly all types of irrigable soils as sprinklers are available in a wide range of discharge capacities [5].

Figure 1-5 Sprinkler Irrigation

a) Sprinkler Irrigation Types and methods

- **Solid set (Fixed Grid) sprinkler irrigation system**

Solid set (or Fixed Grid) irrigation involves installing above ground sprinklers on posts in a grid pattern. This type of system is usually installed in hard to irrigate areas like pivot corners, irregular shaped blocks or areas that have specific site characteristics that make others forms of irrigation unworkable (i.e. protected trees) [6].

Figure 1-6 Solid set sprinkler irrigation

- **Periodic move sprinkler irrigation system**

A periodic move sprinkler system is set in a fixed location for a specified length of time to apply a required depth of water. This is known as the irrigation set time. After an irrigation set, the lateral or sprinkler is moved to the next set position. Applications range from 50% - 75% [7].

Figure 1-7 Periodic move sprinkler system (Hand Roll)

- **Continuous (self) move sprinkler irrigation system:**

Continuous Move/Self Move Systems include center pivots, linear move laterals, and traveling gun sprinklers. Pressure for sprinkler systems is generally provided by pumping powered mainly by diesel or electric and some gasoline engines [7].

Figure 1-8 Continuous move sprinkler system (Continuous pivot irrigation)

b) Sprinkler irrigation advantages:

- Sprinkler irrigation provides high efficiency as the water distribution is uniform
- One of the main advantages of Sprinkler irrigation system is, It can be adopted for all types of soil.
- In a sprinkler irrigation system, the water distribution can be controlled, so it helps in saving water and supplying water in the required quantity for plants.
- There is no loss of cultivable area due to the construction of sprinkler irrigation channels.
- Expansive land levelling or terracing is not required;
- Lower labour requirements as compared to traditional surface irrigation approach [8].

c) Sprinkler irrigation disadvantages:

- The initial investment is high as the cost of equipment (sprinklers and pipes) are expensive and high operating costs due to energy requirements for pumping and labor costs.
- Chances of evaporation loss are high under high wind conditions and high temperatures.
- Unavoidable wetting of foliage in field crops results in increased sensitivity to disease;
- When a sprinkler irrigation system is used to supply Highly saline water causes leaf burning, when the temperature is higher than 35 degrees (Celsius).
- The main disadvantage of sprinkler irrigation system is it requires a continuous power supply to operate [8].

1.3.3. Drip Irrigation

Drip irrigation, also referred to as micro irrigation, trickle irrigation or localized irrigation, involves dripping water onto the soil at very low rates (2 - 20 liters/hour) from a system of small diameter plastic pipes fitted with outlets called emitters or drippers. Water is applied close to plants so that only part of the soil in which the roots grow is wetted, unlike surface and sprinkler irrigation, which involves wetting the entire soil profile. With drip irrigation water, applications are more frequent (usually every 1 - 3 days) than with other methods, thereby providing a favorable high moisture level in the soil for the plant. As long as the application rate is below the soil's infiltration capacity, the soil remains unsaturated and no free water stands or runs over the surface [9].

Figure 1-9 Drip Irrigation

a) Drip Irrigation types and methods

- **Sub-Surface Drip Irrigation:**

Sub-surface irrigation (SDI) is a more sophisticated and hence expensive and rare method, which employs narrow plastic tubes of about 2 cm diameter. These are buried in the soil at a depth between 20 and 50 cm, deep enough so as not to interfere with normal tillage or traffic. The tubes are either porous throughout, or are fitted with regularly spaced emitters or perforations. If porous, the tubes exude water along their entire length. If fitted with emitters, they release water only at specific points [9]. The released water then spreads or diffuses in the soil. The pattern of wetting depends on the properties of the surrounding soil, as well as on the length of the interval between adjacent emitters and their discharge rates .

- **Surface Drip Irrigation:**

Surface drip irrigation is much more common and uses a very large range of drip emitter devices. Lateral lines, supplied from a field main, are laid on the surface. They are commonly 10 to 25 mm in diameter and are either perforated or fitted with special emitters. The latter are designed to drip water on to the soil at a controlled rate, ranging from 1 to 10 liters per hour per emitter. The operating water pressure is usually in the range of 0.5 to 2.5 atmospheres. This pressure is dissipated by friction in flow through the narrow passages or orifices of the emitters, so the water emerges at atmospheric pressure in the form of drops rather than a jet or spray [9].

b) Drip Irrigation advantages:

- Extensive land leveling and bunding is not required, drip irrigation can be employed in all landscapes;
- Irrigation water can be used at a maximum efficiency level and water losses can be reduced to a minimum;
- Soil conditions can be taken into account to a maximum extent and soil erosion risk due to irrigation water impact can be reduced to a minimum;
- Fertilizer and nutrients can be used with high efficiency; as water is applied locally and leaching is reduced, fertilizer/nutrient loss is minimized (reduced risk of groundwater contamination);
- Positive impact on seed germination and yield development [9].

c) Drip Irrigation disadvantages:

- High initial investment requirements;
- Regular capital requirement for replacement of drip irrigation equipment on the surface (damage due to movement of equipment, UV-radiation);
- Drip irrigation emitters are vulnerable to clogging and dysfunction (water filters required, regular flushing of pipe system);
- High skill requirements for irrigation water management in order to achieve optimal water distribution;
- Soil salinity hazard [9].

1.3.4. Sub-Irrigation

Sub-irrigation is a type of irrigation method that provides water to a plant from beneath the soil surface. This type of irrigation is also called "seepage irrigation," and it is often used to grow field crops. Tomatoes, peppers, and sugar cane are often watered in this way. In addition, house plants can be maintained using this type of irrigation.

Many plant experts agree that taking care of house plants is easier to do with the help of sub-irrigation. Rather than watering plants from the top, this method allows plant owners to use a self-watering plant system. Creating a small reservoir of water in the bottom of a plant container allows a plant to soak up water as needed. Unlike the above-watering process, watering a plant from the bottom of a container is easier on the plant [10].

Figure 1-10 Sub-Irrigation

a) Sub-Irrigation advantages:

- It provides moisture to crops by upward capillary action.
- Applying water through SDI maintains dry crop foliage.
- It can be used in low water capacity and in high infilteration rate.
- It reduces incidence of foliar disease.
- It reduces loss of applied pesticides [11].

b) Sub-Irrigation disadvantages:

- Here soil is replaced with water and nutrients suspended in solution.
- Water appliances may be largely unseen, it is more difficult to evaluate system operation and application uniformity.
- It reduces upward water movement.
- It is very costly.
- It is a less developed technology than some alternative Irrigation system [11].

1.3.5. Irrigation system choosing factors

➢ **Soil type:** The type of soil in an area can affect not only the type irrigation method used but also the irrigation run times. Sandy soils typically require frequent applications of water at a high rate to keep moisture in the root zone. Tighter clay soils can hold moisture longer that sandy soils, but may require frequent applications at a lower rate to prevent runoff.

➢ **Land topography:** In particular, hilly or sloping land can be a challenge. Drip irrigation works well if the laterals can be run along topographic lines. System run times may need to be adjusted to prevent runoff. Travelers and center pivot systems are usually out of the question on hilly and severely sloping land.

➢ **Local weather patterns:** For example, sprinklers are less desirable in areas where high winds are common and in arid areas with a low humidity since water losses due to evaporation can be extremely high. Drip irrigation works well in both of these situations.

➢ **Type of crops grown:** Sprinkler and drip systems can require high levels of investment. For this reason, it's better to reserve their use for high-value crops like vegetables, small fruits and orchard crops rather than applying them to commodity crops like wheat and soybeans.

➢ **Water quality:** All drip irrigation systems require some type of filtration. Overhead systems such as sprinklers seldom require filtration. Irrigation water should be tested for water borne pathogens. Depending on the crop grown and irrigation method used chlorine injection may be required. Other water quality issues that may be of a concern include levels of soluble iron and other dissolved minerals [12].

1.4. Factors Affecting Determination of Water Requirement

- Climate
- Type of Crop
- Water table
- Ground Slope
- Intensity of Irrigation
- Conveyance Losses
- Type of soil
- Subsoil water
- Age of canal
- Wetted perimeter
- Method of Application of water [13].

1.4.1. Influence of climate

In hot climate the evaporation loss is more and hence the water requirement will be more and vice versa. A certain crop grown in a sunny and hot climate needs more water per day than the same crop grown in a cloudy and cooler climate. There are, however, apart from sunshine and temperature, other climatic factors which influence the crop water need. These factors are humidity and wind speed. When it is dry, the crop water needs are higher than when it is humid. In windy climates, the crops will use more water than in calm climates.

The highest crop water needs are thus found in areas which are hot, dry, windy and sunny. The lowest values are found when it is cool, humid and cloudy with little or no wind. From the above, it is clear that the crop grown in different climatic zones will have different water needs [13].

Table 1 Effect of major Climatic Factors on Crop Water Needs.

Climatic Factor	High crop water need	Low crop water need
Sunshine	Sunny (no clouds)	Cloudy (no sun)
Temperature	Hot	Cool
Humidity	Low (dry)	High (humid)
Wind speed	Windy	Little wind

1.4.2. Influence of crop type

As different crops require different amount of water for maturity, duties are also required. The duty would vary inversely as the water requirement of crop. The influence of the crop type on the crop water need is important in two ways:

- The crop type has an influence on the daily water needs of a fully grown crop; i.e. the peak daily water needs of a fully developed maize crop will need more water per day than a fully developed crop of onions.
- The crop type has an influence on the duration of the total growing season of the crop. There are short duration crops, e.g. peas, with a duration of the total growing season of 90-100 days and longer duration crops, e.g. melons, with a duration of the total growing season of 120-160 days. There are, of course, also perennial crops that are in the field for many years, such as fruit trees [13].

1.4.3. Water Table

If the water table is nearer to the ground surface, the water requirement will be less & vice versa [13].

1.4.4. Intensity of irrigation

It is directly related to water requirement, the more the intensity greater will be the water required for a particular crop [13].

1.5. Conclusion

In this chapter we have thoroughly explained some of the most important irrigation methods, as well as how to choose the correct method.

Facing the reality that the agricultural sector in this country lacks the implementation of new technologies and advanced methods, we have made the choice to do this work on the institute's greenhouse prototype to present a small-scaled smart irrigation system that can be developed onto a bigger scale in the future, in order to offer new-generation solutions to said sector.

Chapter 2

IoT Based smart irrigation system

2.1. Introduction

A smart irrigation system will auto-execute according to the environment's status, by diagnosing the environmental settings, it will not need any human intervention, especially with the implementation of Internet of Things which allows the user to monitor, control, and diagnose from distance without having to physically be at the agricultural sites.

In this chapter, we will talk about the NodeMCU V3 development board which we use to pilot our system, alongside all the components that allow us to execute irrigation tasks upon changes in the environment, as well as Internet of Things which allows us to monitor and control the entirety of our system.

2.2. Smart irrigation system description

Soil moisture is the main factor of our system, our goal is to monitor it while using a set of thresholds for when to start irrigating and when to stop, the specifications needed to realize this system are the following:

- ✓ The system must monitor Soil moisture and keep the user updated about it.
- ✓ The system must monitor Temperature and Air humidity if used in a greenhouse.
- ✓ Following the type of plant, calibration of the start/stop irrigation threshold for soil moisture is necessary prior to installing the system.
- ✓ Wi-Fi is required on the site for communication between the system and the IoT cloud (GSM can be an alternative).
- ✓ An additional water source is required prior to the water tank on which a plant is irrigated from, this source keeps the main water tank level at a certain minimal/maximal level to provide water at any needed time and avoid cease of process.

2.2.1. Soil moisture

Monitoring your soil moisture levels is crucial for all gardens, lawns, and landscapes because different plants and soil types react differently to certain soil moisture contents.

Following a brief research, we will set our soil moisture thresholds depending on the 4 categories of plants below:

- Flowers: The majority of flowers require moisture levels between 21% - 40%.
- Trees and shrubs: The majority of trees and shrubs require moisture levels between 21% - 40%.
- Fruits: Fruits require moisture levels between 41% - 60%.
- Vegetables**:** All vegetables require moisture levels between 41% - 80% [14].

Ps: Note that the thresholds can be changed anytime if a certain plant does not fit into the categories above.

Our work is tested on a greenhouse prototype, which is commonly used for fruits growing:

If the Soil humidity drops below the threshold of 40%:

- The water pump will start irrigating the surface.
- User will be notified via the Arduino IoT cloud interface and mobile application.

If the Soil humidity exceeds 60%:

- The water pump will stop irrigating the surface.
- User will be notified via the Arduino IoT cloud interface and mobile application.

2.2.2. Temperature and humidity

Additional settings that we need to keep the user informed about are temperature and humidity, if we install our system on a certain greenhouse, these two parameters play a major role and must be kept at a certain level to ensure a proper environment for the plants.

Our system will constantly monitor these two parameters and update them on the Arduino IoT Cloud.

We mention two high level alerts:

If the humidity level exceeds 90%:

- A warning notification will be sent to the Arduino IoT Cloud interface and the mobile application as: “Hygrometry rate too high!!, possible appearance of water droplets!”.

If the temperature is too low (0 Celsius or lower):

- A warning notification will be sent to the Arduino IoT Cloud interface and the mobile application as: “Temperature too low!!,freeze warning!”.

2.3. Components and devices used

In this project, the equipment we used are the following:

Table 2 Hardware Equipment and accessories.

Components number	Components	Number of pieces
1	Arduino Mega 2560 Board	1
2	NodeMCU V3 Board	1
3	Humidity/Temperature sensor DHT11	1

4	Soil moisture sensor	1
5	5V Relay	1
6	5.5/12 V Submersible Water Pump	1
7	Connection Cable 30cm	15
8	Water Pipe	2m
9	Ultrasonic sensor	1

2.4. NodeMCU ESP8266 Development Board

ESP8266 NodeMcu is a popular and widely used development board based on the ESP-12E WiFi Module that combines elements of easy programming with Arduino IDE (C\C++) and WiFi capability. Through the build-in programmer and CH340G USB-to-Serial chip, flashing the ESP8266 and serial output on a PC, development and prototyping projects are done with ease. Just like Arduino boards, the ESP8266 NodeMcu has GPIO pins, voltage regulator, ADC, Micro-USB port (for flashing and serial output) – all on one board. On top of that the ESP8266 NodeMcu has a full WiFi that takes care of the WiFi communication to a server or client [15].

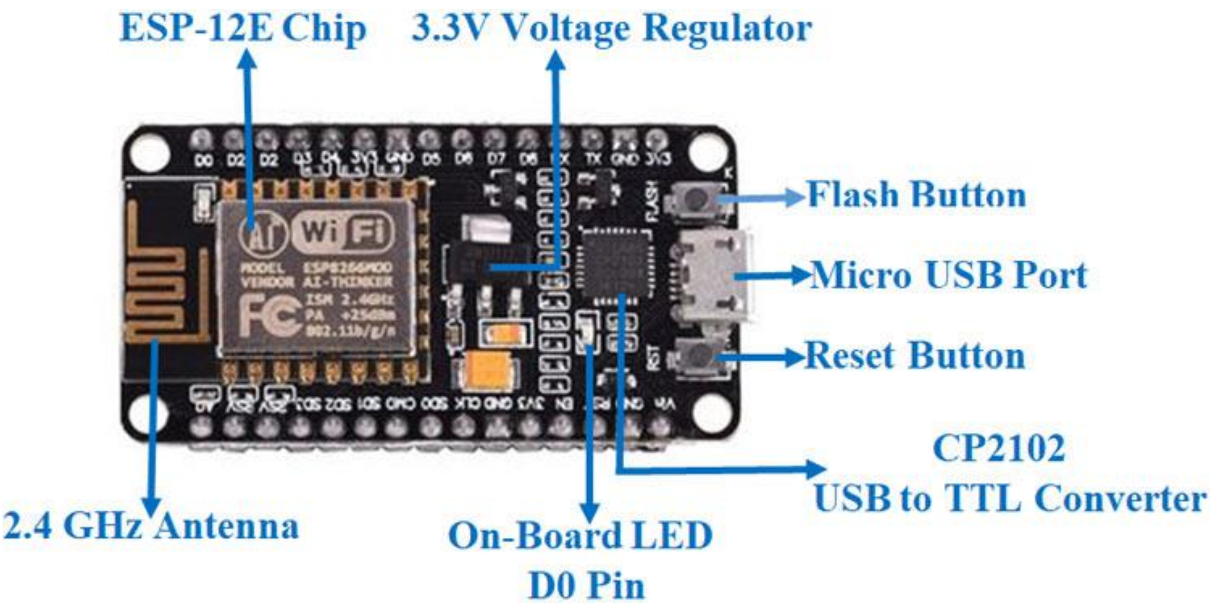


Figure 2-1 NodeMCU V3.

2.4.1. ESP-12E Wi-Fi chip

What makes the NodeMCU board IoT friendly and easy to use, is its integrated Wi-Fi module, the esp-12e. Our work depends on sending and receiving data through the cloud, which makes Wi-Fi a necessary asset to have, so instead of using an Arduino uno board and externally plug a Wi-Fi module to it, we have decided to use a more simplified fast and developed board.

ESP-12E is a miniature Wi-Fi module present in the market and is used for establishing a wireless network connection for microcontroller or processor. The core of ESP-12E is ESP8266EX, which is a high integration wireless SoC (System on Chip).

It features the ability to embed Wi-Fi capabilities to systems or to function as a standalone application. It is a low cost solution for developing IoT applications (Check annex for more details about this chip) [16].

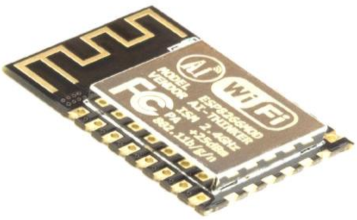


Figure 2-2 ESP-12E Wi-Fi Chip.

2.4.2. Flash and Reset buttons

Our board has 2 integrated buttons as shown in **Figure 2.1**:

- **Flash button**

What is the flash button on NodeMCU? It’s tied to gpio0 and can be used as an input, it is also used to flash the module by holding flash and reset. If you look up the schematics for the NodeMCU, there is a part of the circuit connected to the USB to serial converter that allows the serial connection to do this for you.

- **Reset button:**

The reset button does pretty much the same as unplugging the board and plugging it back in. It restarts your program from the beginning. The same thing happens when you program the board the USB interface presses the reset button for you.

2.4.3. NodeMCU Advantages and disadvantages

We mention some advantages and disadvantages of the board we are using:

➢ **Advantages of NodeMCU platform relative to the Arduino**

- Low cost
- Integrated support for WIFI network
- Reduced size of the board
- Low energy consumption

➢ **Disadvantages**

- Need to learn a new language and IDE
- Reduced pinout
- Scarce documentation

2.5. Sensors

A sensor is a device that detects and responds to some type of input from the physical environment. The specific input could be light, heat, motion, moisture, pressure, or any one of a great number of other environmental phenomena. The output is generally a signal that is converted to human-readable display at the sensor location or transmitted electronically over a network for reading or further processing [17].

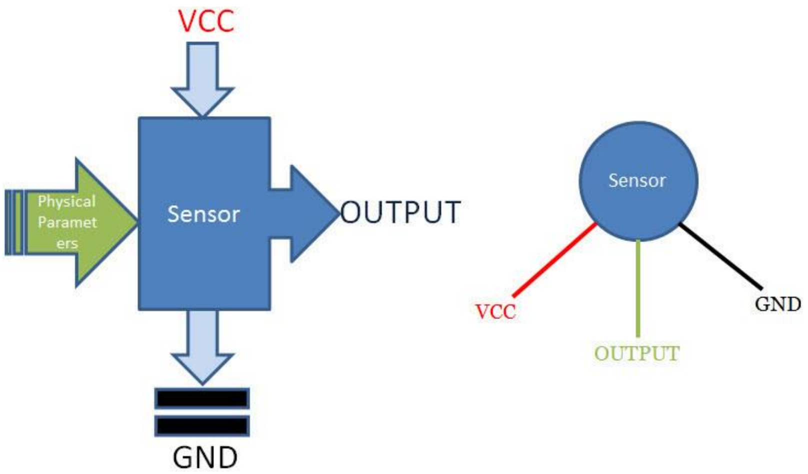


Figure 2-3 Sensor.

2.5.1. Classification of sensors

We can classify sensors into three categories according to:

- Type of input required to get desired output from a sensor.
- The sensing mechanism.
- Type of output.

- **Active/passive sensors**
  - **Active sensors** are a type of sensors that require continuous input to get the required output from the sensing element. RADAR, Chip-based humidity and Temperature Sensors, Gas Sensor, GPS, are examples of Active Sensors. These Sensors require continuous electrical power to operate.
  - **Passive sensors** are the type of sensor that does not require any external input such as power signal to get the required output. Examples of passive sensors are metal detectors, magnetic field detectors, photodiode, thermistors, and strain gauges. They require electric power only to respond or give output results.
- **Electrical, chemical or mechanical Sensors**
  - **Electrical Sensors** are the sensors in which a sensing element detects a physical parameter and converts it into an electrical signal. This electrical signal is used to determine the value of that physical parameter. Most of the sensors used in automation are electrical sensors. NTC, humidity, and capacitive sensors are examples of electrical sensors.
  - **Chemical sensors** are the type of sensors that respond to the chemical reaction. For example, the water PH chemical sensor shows different colors according to the PH level
  - **Mechanical Sensors** utilize mechanisms to detect a physical parameter. For example, a strain gauge is a mechanical sensor that detects mechanical deformation and converts this deformation into an electrical signal. Examples of mechanical sensors are touch sensors, biopic, and stress gauges.
- **Analog /Digital sensors**
  - **A digital sensor** produces a discrete output or binary (0 or 1). These days digital sensors are more popular compared to analog sensors
  - **AnAnalog sensor** produce a continuous output that is proportional to the measured parameter. Examples of analog sensors are accelerometers, pressure sensors, light sensors, and temperature sensors [18].

2.5.2. Used sensors in our project

Our system uses two different sensors:

a) Soil moisture sensor

Soil moisture sensors measure the volumetric water content in soil. Since the direct gravimetric measurement of free soil moisture requires removing, drying, and weighing of a sample, soil moisture sensors measure the volumetric water content indirectly by using some other property of the soil, such as electrical resistance, dielectric constant, or interaction with neutrons, as a proxy for the moisture content.

The relation between the measured property and soil moisture must be calibrated and may vary depending on environmental factors such as soil type, temperature, or electric conductivity [19].

We will be using the FC-28 soil moisture sensor in our system, it operates on a range of 3.3V to 5V DC.

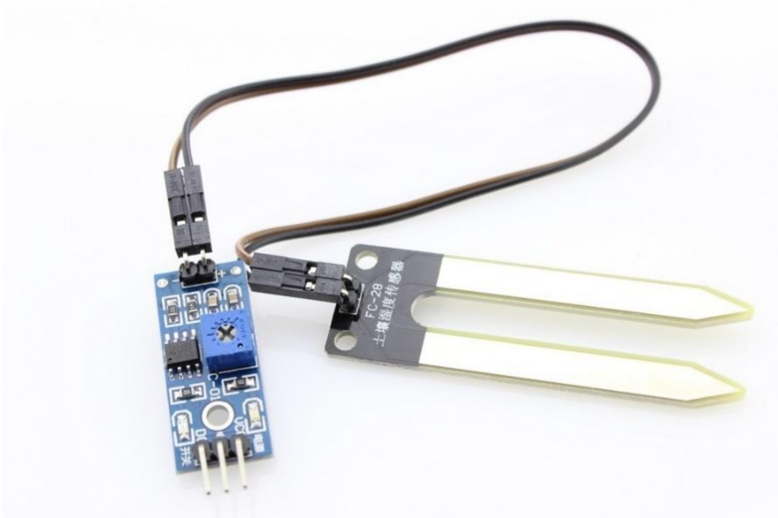


Figure 2-4 Soil moisture sensor

b) Temperature and Air humidity sensor DHT11:

This is a calibrated digital temperature and humidity module with onboard sensor DHT11. It can be used for detecting ambient temperature and humidity, through the standard single-wire interface [20].

Some specifications about this sensor are mentioned below:

- Temperature

Resolution: 1°C.

Accuracy: ±2℃.

Measuring range: 0°C ~ 50°C

- Humidity

Resolution: 1%RH

Accuracy: ±5%RH (0~50°C)

Measuring range: 20%RH ~ 90%RH (25°C)

- Operating voltage: 3.3V ~ 5.5 V

Recommended storage condition

Temperature: 10°C ~40°C

Humidity: 60%RH or below [20].

Figure 2-5 Temperature and air humidity sensor DHT11

c) Ultrasonic sensor

The HC-SR04 ultrasonic sensor uses sonar to determine the distance to an object. This sensor reads from 2cm to 400cm (0.8inch to 157inch) with an accuracy of 0.3cm (0.1inches), which is good for most hobbyist projects. In addition, this particular module comes with ultrasonic transmitter and receiver modules.

The ultrasonic sensor uses sonar to determine the distance to an object. Here’s what happens:

- The ultrasound transmitter (trig pin) emits a high-frequency sound (40 kHz).
- The sound travels through the air. If it finds an object, it bounces back to the module.
- The ultrasound receiver (echo pin) receives the reflected sound (echo) [21].

Figure 2-6 Ultrasonic sensor

2.6. Water Pump

Our system is based on a greenhouse prototype scale, which requires a non-powerful water pump, one that is sufficient to extract water and irrigate a small plant, so we will be using a 5V-12V submersible water pump (DC-1020). (Check Annex for more details about this water pump)

This Water pump will be powered and controlled using a 5V relay and a set of thresholds for the soil moisture.

Figure 2-7 12V Submersible water pump DC-1020

2.7. Power Supply

Our system will be powered directly with 12V using a USB cable that connects the NodeMCU to the computer, the rest of the components will be powered at 3.3V by the board.

However, the water pump requires 12V which we will power up using an external power supply and a 5V relay module.

2.8. 5V Relay

Relays are electromechanical components, acting as a switch, they open/close electrical circuits from incoming logical signals.

Our NodeMCU board outputs 3.3V power, which is not enough for the relay to function, because of this we will power the relay using a USB cable directly from the computer (12V), and at the same time command it using the NodeMCU digital pins, it will then act upon incoming logical signals and transform the 12V into an operating 5V for the water pump.

Figure 2-8 5V Relay

2.9. Internet of Things (IoT)

Internet of Things (IoT) describes the network of physical objects— "things"—that are embedded with sensors, software, and other technologies for the purpose of connecting and exchanging data with other devices and systems over the internet. These devices range from ordinary household objects to sophisticated industrial tools. Devices do not need to be connected to the public internet, they only need to be connected to a network and be individually addressable [22].

While the idea of IoT has been in existence for a long time, a collection of recent advances in a number of different technologies has made it practical.

- **Access to low-cost, low-power sensor technology**. Affordable and reliable sensors are making IoT technology possible for more manufacturers.
- **Connectivity**. A host of network protocols for the internet has made it easy to connect sensors to the cloud and to other "things" for efficient data transfer.
- **Cloud computing platforms**. The increase in the availability of cloud platforms enables both businesses and consumers to access the infrastructure they need to scale up without actually having to manage it all.
- **Machine learning and analytics**. With advances in machine learning and analytics, along with access to varied and vast amounts of data stored in the cloud, businesses can gather insights faster and more easily. The emergence of these allied technologies continues to push the boundaries of IoT and the data produced by IoT also feeds these technologies.
- **Conversational artificial intelligence (AI)**. Advances in neural networks have brought natural-language processing (NLP) to IoT devices (such as digital

personal assistants Alexa, Cortana, and Siri) and made them appealing, affordable, and viable for home use [22].

IoT plays a major role in our project, it is the necessary ingredient for a perfect autonomous system that does not need any human intervention, we integrated IoT so that our system sends and receives data automatically and acts upon that information, of course, after injecting a certain program to the NodeMCU. We will detail the program used in the next chapter.

2.10. Conclusion

In this chapter, we have theoretically talked about the different equipment we need to create our project, as well as the Internet of Things addition which allows us to monitor and control from distance our system.

Chapter 03

Smart irrigation system making

3.1. Introduction

Our system will be installed on the greenhouse prototype that we mentioned before, the idea is to give it multiple additions, so on top of the smart irrigation, we want to introduce IoT to the different settings that the greenhouse already had (doors, lights, ventilation, hot plates).

In this Chapter, we will be talking about the working principle of the smart irrigation system, as well as the programming side of it including IoT , we will also include a diagram explaining the process. In addition to that, we will explain how to establish communication between the NodeMCU and the Arduino in order to add manual mode (command from the cloud) and automatic mode supervision (Supervision from the cloud) to the greenhouse.

3.2. Smart irrigation system working principle

Irrigation is done through a small water tank (containing enough water to water a plant for a certain time) by using a submergible 12V DC water pump.

This pump acting upon incoming signals from a 5V relay (commanded by the NodeMCU board through the S pin and can be used either as normally open or normally closed) will extract water from inside the tank at a maximum volume flow rate of 200L/H.

Extracted water will then flow through a roughly 2,5 meters long water tube connected to the pump whereas the end of the tube is put on the surface of the plant soil (Surface irrigation type).

Irrigation will start or stop following a set of thresholds we mentioned in chapter 2, the relay will work as a closed circuit if the soil moisture sensor detects lack of moisture meaning irrigation starts, and vice versa.

Arduino IoT Cloud will offer us a dashboard allowing us to monitor irrigation rates and greenhouse status and to also control certain actions from distance (such as manually start/stop irrigation or open/close the greenhouse doors, powers up/shuts off the ventilation, turns on/off the lights, ect…)

The whole system needs little energy, so we will be powering it up using the NodeMCU board, except for the relay.

The advantage of this system is offering a portable device that allows for home smart irrigation, our system is indeed dedicated to the greenhouse, but by making it a device model you can carry it around and irrigate your plants anytime any place, but that is not the only advantage, expanding this project to a bigger scale will allow agriculture in Saharan areas to prosper with the help of energy sources such as solar energy and wind energy.

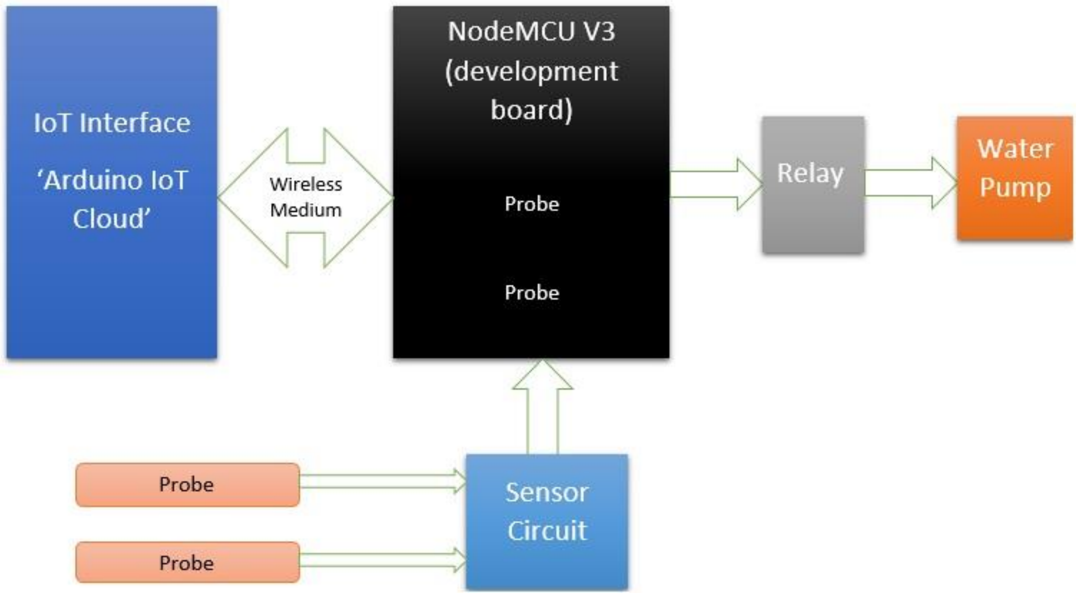


Figure 3-1 NodeMCU V3 Based smart irrigation system diagram

The diagram above summarises the concept of our system, with the NodeMCU V3 being the control system of it.

3.3. Programming and Arduino IoT Cloud

The presence of equipment alone is insufficient for a NodeMCU project to work; we need to program the equipment at hand, both aspects are mandatory to create a finished product.

To command and monitor our irrigation system we are going to use the platform we mentioned in chapter 2 (Arduino IoT Cloud) to program our development board, we need to plug our board to the PC or laptop, register in the platform, and connect the board with the platform via Wi-Fi

3.3.1. Arduino IoT Cloud

The Arduino IoT Cloud is a online platform that makes it easy for you to create, deploy and monitor IoT projects, with a user friendly interface, and an all in one solution for configuration, writing code, uploading and visualization.

You can build visual dashboards to monitor and control your Arduino and similar boards projects, integrate with other services and much more.

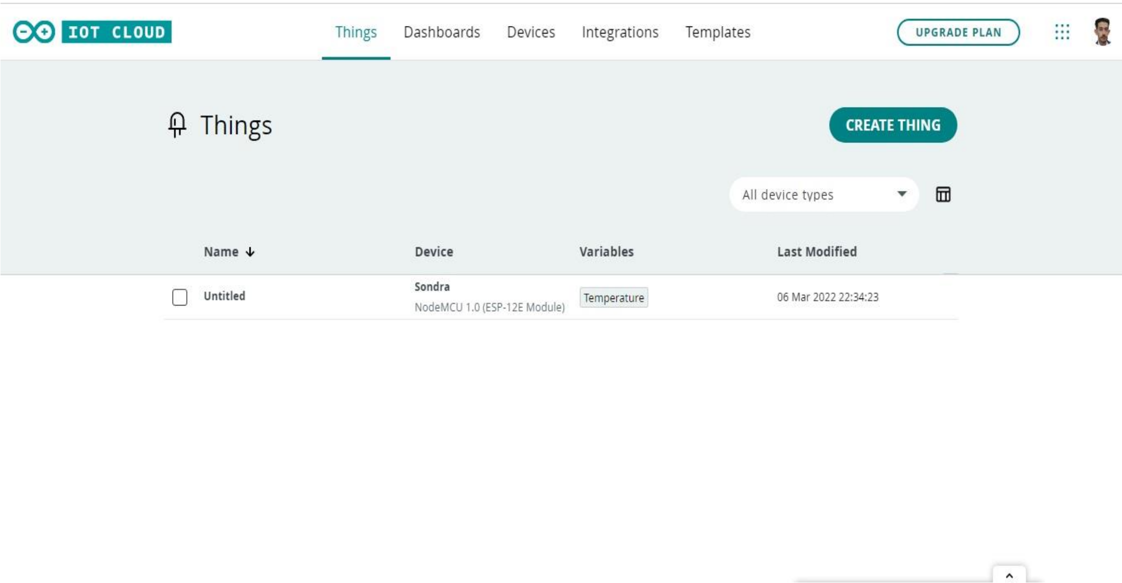


Figure 3-2 Arduino IoT Cloud online platform

After singing up, the site will let you open up your Arduino IoT Cloud account, showing the interface and its different tabs above, we will explain thoroughly each tab and its function and how we use them in this project.

a) Things

As shown in **Figure3.3** the first tab we have is Things, in order to use our devices in IoT Cloud, we need to associate them to a Thing (A project for example). A Thing is an abstract concept that includes the configuration of the variables and other settings, as well as the history of the data collected for the variables.

On the Things tab click "**CREATE THING**", From there we are Shown 2 tabs:

b) Setup

This is the part where you add your device by pressing "**Select Device"**, after selecting it from a list of compatible boards (Arduino and other boards such as NodeMcu) you need to connect to a Wi-Fi network, to do that press "**Configure"** on the Network area, after going online, you can finally add your variables, choose from a wide list of different variable types.

In our project we will be adding a Temperature variable, Humidity variable for both Air humidity and Soil Moisture, and a Boolean variable for the relay.

Please note that each variable we create will auto-declare its variable name, which we will be using later on in the program section.

c) Sketch

This part is where you will include your program for a certain project, you can either use the normal editor or the full editor by pressing **"Open full editor"**, the full editor contains other functions like a terminal to monitor the results of your program, a sketch library where you can upload sketches from the internet that you want to use, and many more functions.

Note that the Arduino IoT Cloud uses the same language as the Arduino IDE, so there should be no problem copying your IDE programs or editing them in the Cloud.

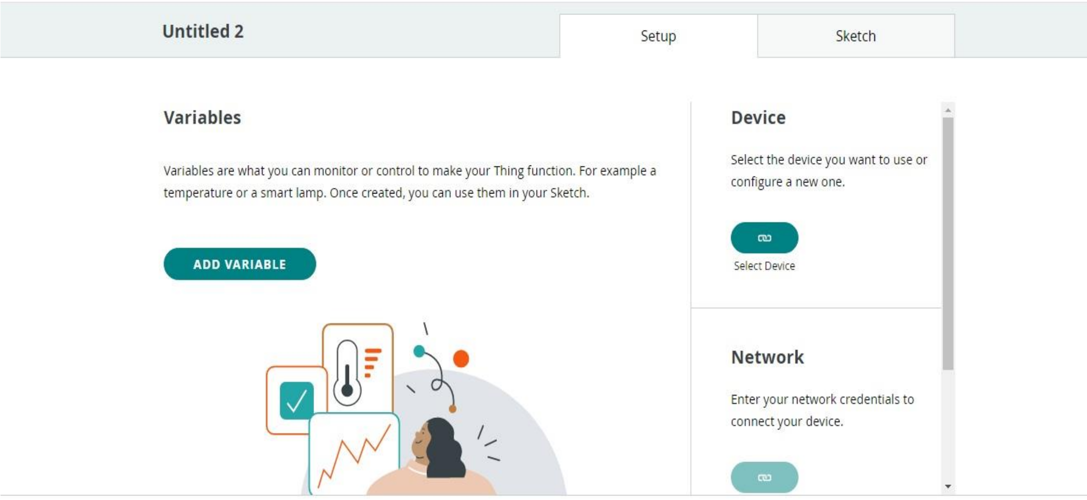


Figure 3-3 Things Tab from the Arduino IoT Cloud

d) Dashboards

In this tab you can create a dashboard of various widgets that will allow you to monitor the different variables you've created in **"Things"**, from gauges to switches to percentages and many more, these widgets will receive real-time data from sensors and show them depending on the type of variable and desired widget type.

To start, from the Dashboards tab, press **"Build Dashboard"**, then from the page shown in **Figure 3.4** click **"Add"** and then choose the desired widget type.

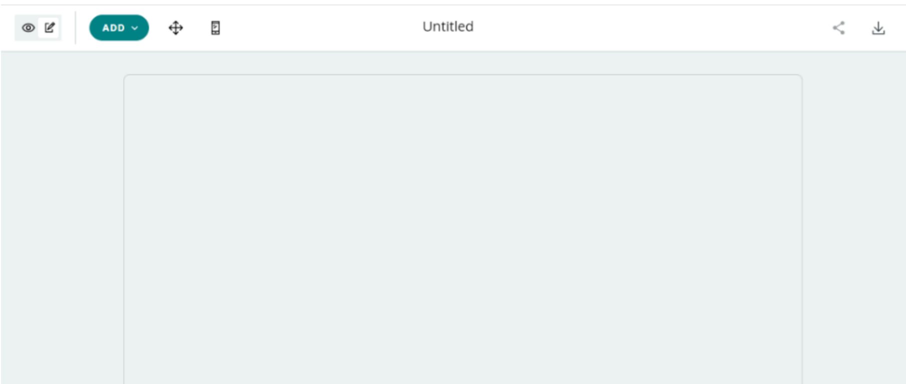


Figure 3-4 Arduino IoT cloud Dashboards

After choosing the desired widget, you will now be redirected to **“Widget Settings”**, where you can name your widget and link it to a certain variable by pressing **“Link Variable”** as shown in **Figure 3.5**, after finishing the steps below, press **“Done”**.

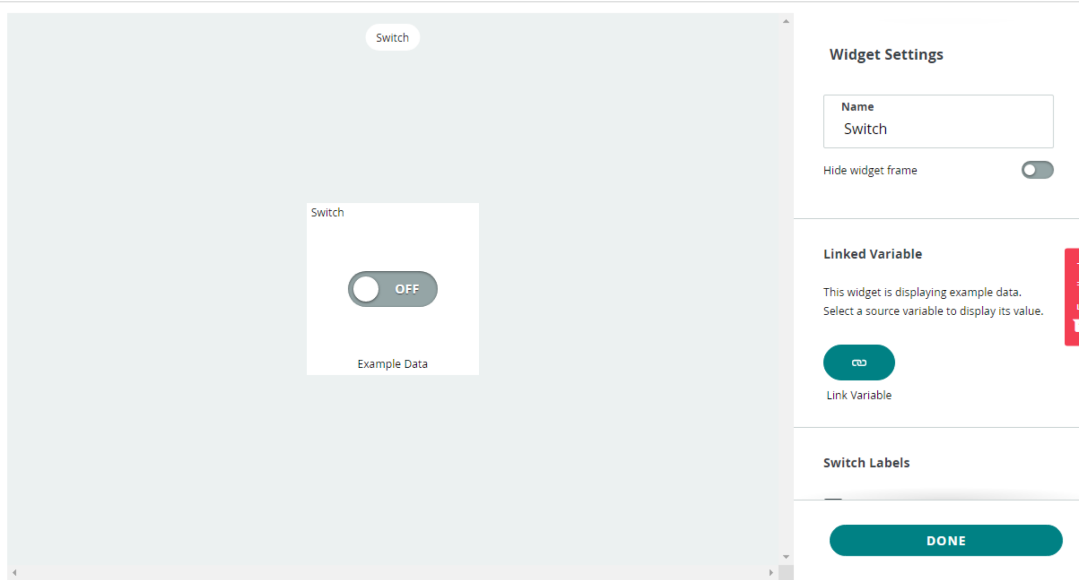


Figure 3-5 Widget Settings

In our project, we will be linking Temperature variable to gauge widget and both humidity and soil moisture to percentage widget, and the relay to switch widget.

We will also add a messaging widget that we can use to receive alerts from the web interface and the mobile application.

3.3.2. Programming on Arduino IoT Cloud

After setting up our board and variables and dashboards, the most important part comes to play, the sketch.

To fully benefit from the sketch part, we need to open the full editor by pressing this button:

The window given should look like the figure below (**Figure 3.6)**.

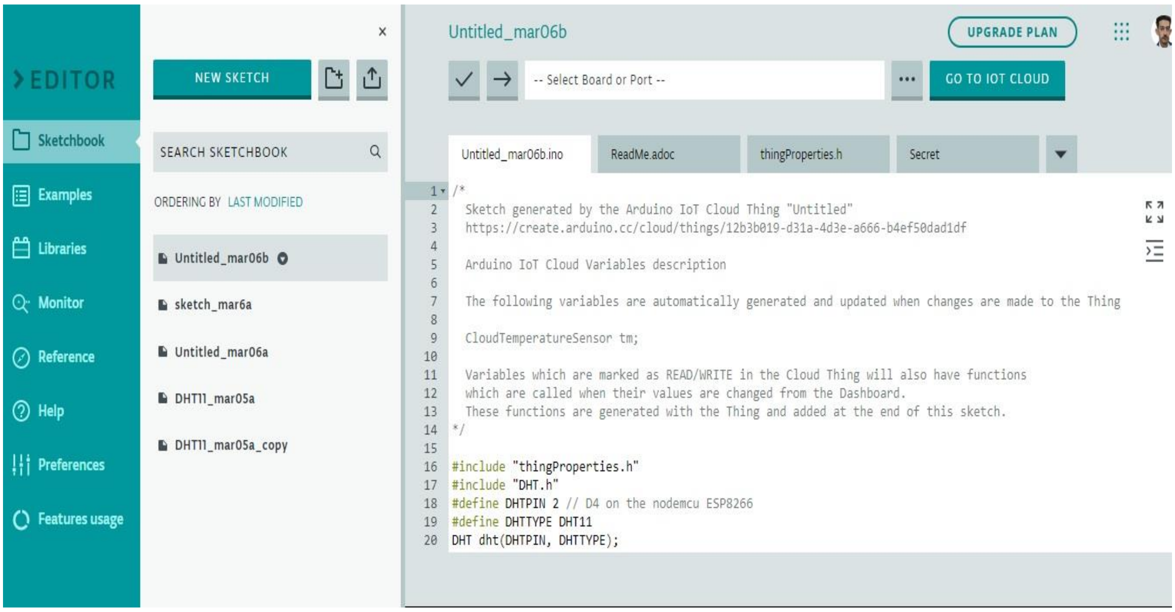


Figure 3-6 Full sketch editor

- As the figure shows, the full editor can be confusing for new users, so let us explain the most important blocks that will help you set up your program.
- ‘Select Board or Port’ is the most important part, we need to select the development board we are using to control our system in order to upload the sketch into it.
- ‘Sketchbook’ this is the part where we add/modify/import sketches (source code), this part uses the same language as Arduino IDE and it is where we will be writing our full program for the project.
- ‘Examples’ is the part where we can upload samples of programs for sensors, actuators, motors, etc…, we can find already built-in examples because the Arduino IoT Cloud library is vast and contains pretty much all of the libraries unlike the Arduino IDE, or we can import certain examples that do not exist already.
- ‘Libraries’ is usually a crucial part, but as mentioned above, the library of this platform is huge, so most of the libraries for different sensors and motors already exist, but in-case we are missing a certain library, this block allows you to search or import libraries that you need to complete your program (certain programs will not function if their proper library is not installed).

- 'Monitor' or terminal, is the part where you can monitor your program's execution through a certain serial port, it can help you detect errors or miscalculations on your program before you have to export them into the final project, it is the practical execution of your theoretical program.
- 'Upload and save', this will upload the program to your development board, save it, and run it for you to monitor.
- 'Verify and save', this will only verify the integrity of your source code and check for errors, as well as save it, it will not run it or upload it to your board.

3.3.3. Arduino IoT Cloud program structure

Writing a program on this platform requires a certain structure that is similar to that of Arduino IDE. We distinguish three essential parts:

- Variables declaration and libraries call: the first step to making a complete program is to declare variables that we are going to use throughout the whole coding process, as well as calling forth the different libraries we need for certain purposes.

```
#include "thingProperties.h"
#include "DHT.h"
#define DHTPIN 2 // D4 on the nodemcu ESP8266
#define DHTTYPE DHT11
int f=1
DHT dht(DHTPIN, DHTTYPE);
```

- Void setup(): This is the initialization loop, this part will allow the code written in it to run only once per use, meaning after every RESET.

```
void setup() {
  // Initialize serial and wait for port to open:
  Serial.begin(9600);
  // This delay gives the chance to wait for a Serial Monitor without blocking if none is found
  delay(1500);
```

- Void loop(): This part will run the code in an endless loop as long as the development board keeps running, this part is where most of the coding will be written.

```
void loop() {
  ArduinoCloud.update();
  // Your code here
```

3.3.4. Language Syntax

In order to avoid any mistakes in the source code or error messages, we need to make sure our syntax is respected properly:

- Colours: we distinguish 3 different colours to help us define the status of a certain syntax
- Orange/Green: Key-words for Functions in a source code are highlighted in orange or green.
- Grey: Commentaries are highlighted in grey. They will not run with the program.
- Blue: Highlights key-words for constants.
- Punctuations: Punctuations are mandatory for a program to run properly, it is the main source of many error messages, we not 3 essential punctuations:
- Semicolons "; ": most of lines in a source code need to end with a semicolon.
- Parentheses "( ) ": each Function has its settings, we write those settings between parentheses.
- Braces "{ } ": Functions, loops and blocks are put between braces.
- Commentaries: commentaries are useful to write certain notes about each part of the source code, we use " // " at the start of a line.

3.3.5. Arduino Iot Cloud program example

We will run an example code through the platform with the DHT11.

Coding the DHT 11 is a simple task, we just need to include its library, and call for the **Function** to read from the sensor:

```
#include "thingProperties.h" // Auto generated by the cloud.
#include "DHT.h" // This includes the DHT11 Library.
#define DHTPIN 2 // Our DHT11 is plugged in to D4, So this will allow the code to read from it.
#define DHTTYPE DHT11 // Defining the DHT Type.
DHT dht(DHTPIN, DHTTYPE);
void setup() {
  // Initialize serial and wait for port to open:
  Serial.begin(9600);
  // This delay gives the chance to wait for a Serial Monitor without blocking if none is found
  delay(1500);
  // Defined in thingProperties.h
  initProperties();
  // Connect to Arduino IoT Cloud
  ArduinoCloud.begin(ArduinoIoTPreferredConnection);
```

Figure 3-7 DHT11 example code part 1

```
void loop() {
  ArduinoCloud.update();
  // Your code here
    hm=dht.readHumidity();
   Serial.print("Humidity :");
   Serial.print(hm);Serial.println(" %");

   tm=dht.readTemperature();
    Serial.print("Temperature :");
   Serial.print(tm);Serial.println(" C°");


   delay(1000);

}
```

Figure 3-8 DHT11 example code part 2

The following variables are automatically generated and updated when changes are made to the Thing: (CloudTemperatureSensor tm;),(CloudRelativeHumidity hm;).

The serial monitor will then show the following results:

```
Connected to Arduino IoT Cloud
Thing ID: 4d3f02a5-33ae-4525-af55-6f2030bfd77e
Humidity :56.00 %
Temperature :31.80 C°
Humidity :56.00 %
Temperature :31.80 C°
Humidity :56.00 %
```

Figure 3-9 DHT11 reading results

The dashboard widgets will then receive real time data and show the following results:

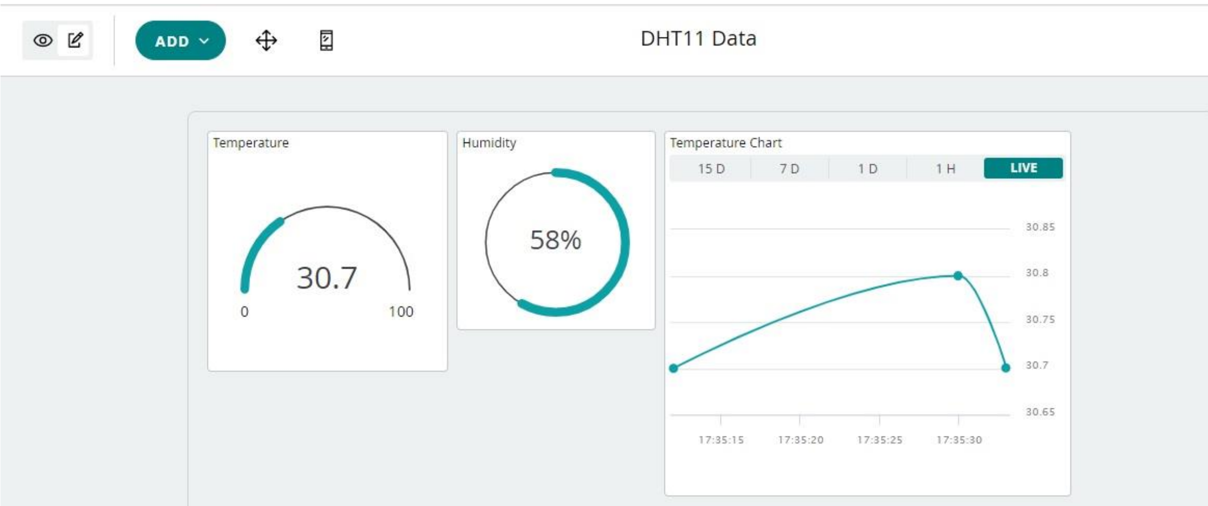


Figure 3-10 Arduino IoT Cloud widgets real-time readings

3.4. Inclusion of the smart greenhouse into the cloud

We want to add cloud supervision (auto mode) and manual override (manual mode) to the greenhouse, the Arduino Mega 2560 does not have Wi-Fi connection therefore it cannot directly connect to the cloud.

In order to establish that connection a specific serial communication is needed between the Arduino Mega (greenhouse control centre) and the NodeMCU (wireless medium), this will allow two-way communication between both boards where one will be the master (NodeMCU) and the other the slave (Arduino Mega 2560)

After doing a thorough research about serial communications, we have come to conclusion that I²C is best suited for our purpose.

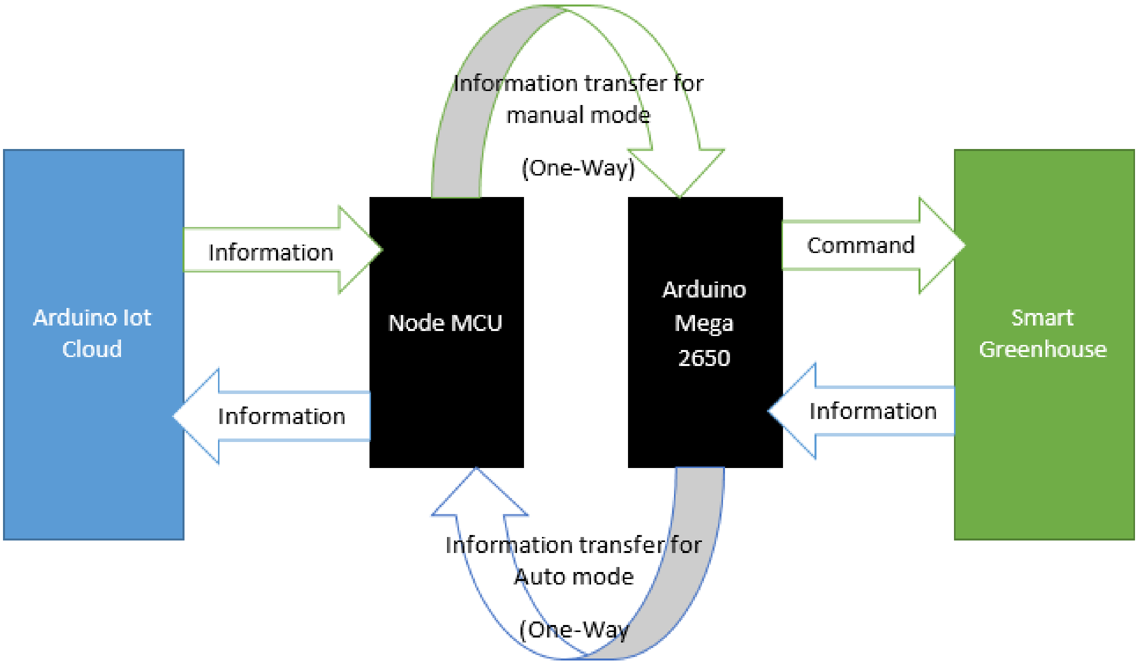


Figure 3-11 Communication for Auto/Manual mode diagram

3.5. I²C Communication between the NodeMCU V3 and Arduino:

I2C stands for Inter-Integrated Circuit. It is a bus interface connection protocol incorporated into devices for serial communication. It is a widely used protocol for short-distance communication. It is also known as Two Wired Interface(TWI). It uses only 2 bi-directional open-drain lines for data communication called SDA and SCL. Both these lines are pulled high.

- Serial Data (SDA) – Transfer of data takes place through this pin.
- Serial Clock (SCL) – It carries the clock signal.

I2C operates in two modes:

- Master mode
- Slave mode

Each data bit transferred on SDA line is synchronized by a high to the low pulse of each clock on the SCL line.

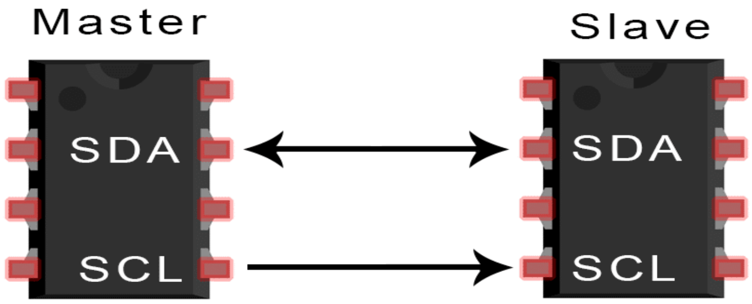


Figure 3-12 I²C Serial communication protocol

3.6. Conclusion

In this chapter, we have presented the two phases of the making of our system.

The first one: the functioning principle of the smart irrigation system where we briefed about the system's specifications.

The second one: setting up the programming part including the platform we used for that where we explained thoroughly how to use the Arduino IoT Cloud for both programming and monitoring, we also talked about the inclusion of the greenhouse parameters into the cloud and how we established communication between both boards (NodeMCU and Arduino Mega 2560).

The next chapter will include the electronic part and wiring as well as real-time test of the system and validation of the results.

Chapter 04

Test and results

4.1. Introduction

This chapter will cover the electronic aspect and the wiring done to create our project and to set up the cloud, our work is then concluded with a final test and validation of the whole system (including the integration of the greenhouse settings into the cloud).

The final project will be an IoT based smart greenhouse prototype with irrigation.

4.2. Smart irrigation system description

Every irrigation system must be controlled via certain thresholds of soil moisture and sometimes air humidity and temperature, for the reason that it cannot be done at any time, but only when it is needed to.

Following some specifications, we can control the times we need to start or stop irrigating (we are using fruits crop as an example).

- The system must monitor the soil moisture percentage.
- If soil moisture percentage drops below 40%, irrigation starts, user will be notified in the cloud.
- If soil moisture percentage rises above 60%, irrigation stops, user will be notified in the cloud.
- If humidity exceeds 90%, user will receive an alert in the cloud.
- If temperature reaches 0 C°, user will receive an alert in the cloud.

Each electronic component we mentioned in chapter 2 needs to be wired differently, we will describe these components and show how they are wired following the specifications of our system.

In addition to that, we will talk about the serial communication between the NodeMCU V3 we are using and the Arduino Mega 2560 that the greenhouse uses in order to integrate its settings into the Arduino IoT Cloud.

4.3. NodeMCU V3 Board

As mentioned in chapter 2, the NodeMCU development board has many digital pins, and only one analog pin.

In order to program the board, we need a computer and arduino language knowledge, the source code needs to be small and practical.

4.3.1. Wiring the DHT11 with the NodeMCU V3 Board

The DHT11 sensor will help us keep an eye on the plants and crops and ensure a safe environment for them, because at certain circumstances like overheats or frost they can be damaged for good.

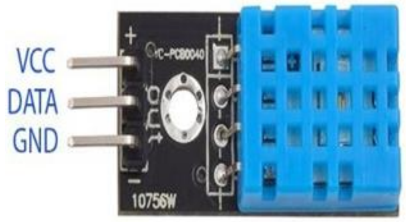


Figure 4-1 DHT11 sensor pinout.

The wiring of this sensor is done like this:

- VCC needs to be connected to the 3.3V pin of the board.
- GND needs to be connected to the GND pin of the board.
- DATA needs to be connected to the D4 pin of the board.

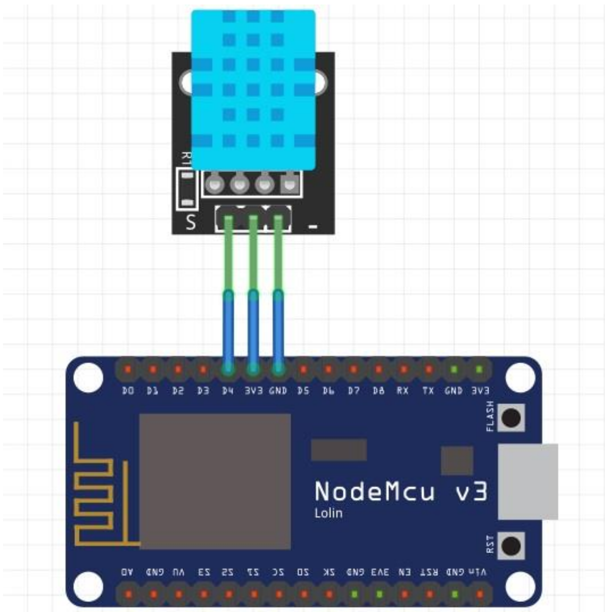


Figure 4-2 DHT11 sensor wiring with NodeMCU V3 Board.

4.3.2. Wiring the Soil moisture sensor with the NodeMCU V3 board:

The soil moisture sensor is the main equipment for a smart irrigation system, it will keep track of the soil status and sends signals of readings to the Node MCU which then acts upon certain thresholds

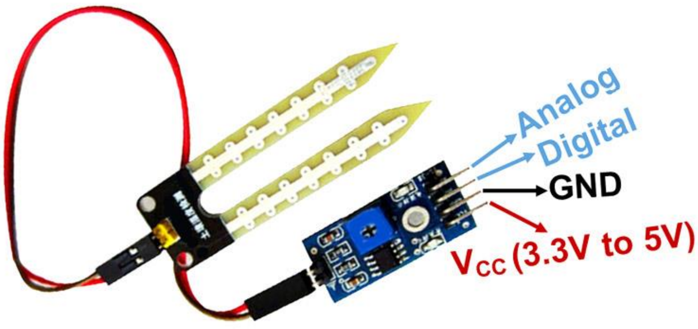


Figure 4-3 Soil moisture sensor pinout

- The wiring of this sensor is done like this:
- VCC needs to be connected to the 3.3V pin of the board.
- GND needs to be connected to the GND pin of the board.
- A0 needs to be connected to the A0 pin of the board.

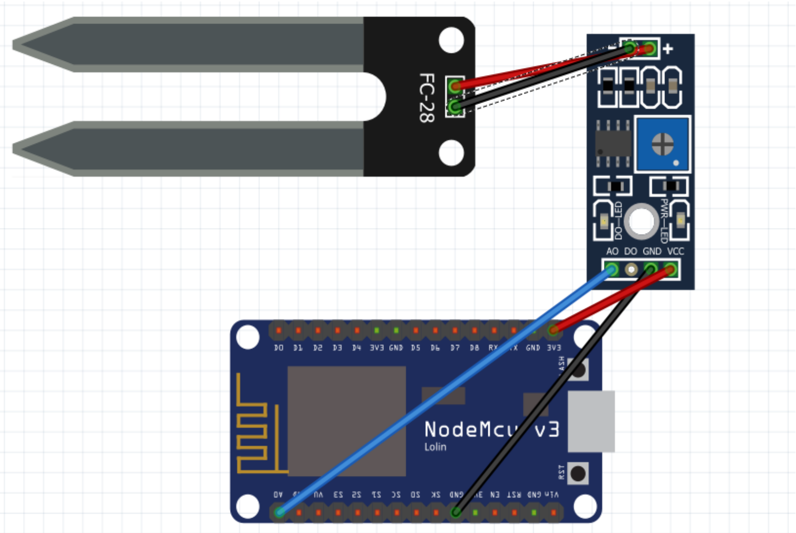


Figure 4-4 Soil moisture sensor wiring with NodeMCU V3 Board

4.3.3. Wiring the water pump and relay with the NodeMCU V3 board

The water pump contains only positive and negative wires, no command wire, therefor in order to command it we will need a 5V relay powered and controlled by the NodeMCU, but we will use an external 12V power source to power-up the water pump using NO jack from the relay.

Figure 4-5 5v relay module pinout

The wiring of the relay and water pump is done like this:

- VCC needs to be connected to the 3.3V pin of the board.
- GND needs to be connected to the GND pin of the board.
- S needs to be connected to the D2 pin of the board.
- NO needs to be connected to + (positive) of external power source (12V).
- COM needs to be connected to + (positive) of water pump.
- **- (**negative) of water pump needs to be connected to – (negative) of power source.

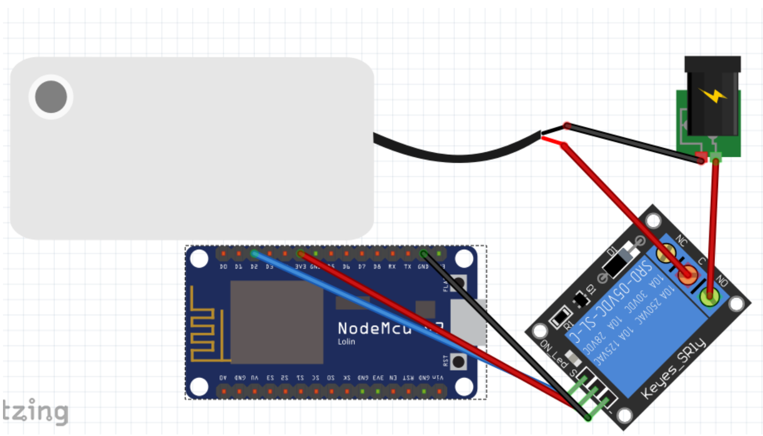


Figure 4-6 Relay and water pump wiring with NodeMCU V3 Board

4.3.4. Wiring the ultrasonic sensor with the NodeMCU V3 Board:

This addition is to monitor the level of water in the water tank, because irrigation cannot be done without the presence of water, so this sensor will help keep track of its level by measuring the distance between the water surface and the length of the tank (following a threshold full distance reading means empty and minimal distance reading means full).

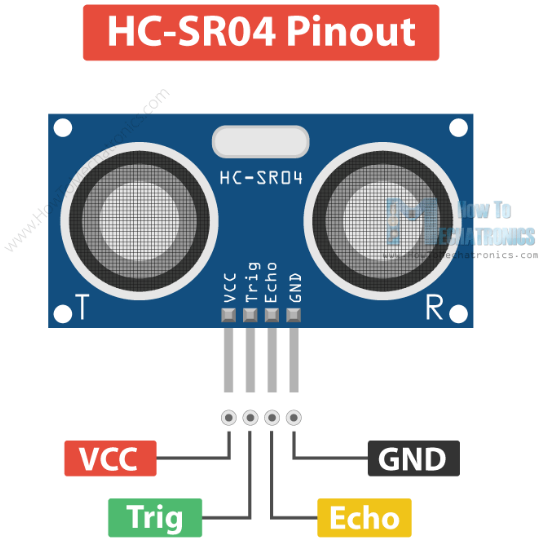


Figure 4-7 Ultrasonic sensor pinout

The wiring of this sensor is done like this:

- VCC needs to be connected to the 5V pin of the Arduino Mega board.
- GND needs to be connected to the GND pin of the Arduino Mega board.
- Trig needs to be connected to the D8 pin of the NodeMCU board.
- Echo needs to be connected to the D7 pin of fthe NodeMCU board.

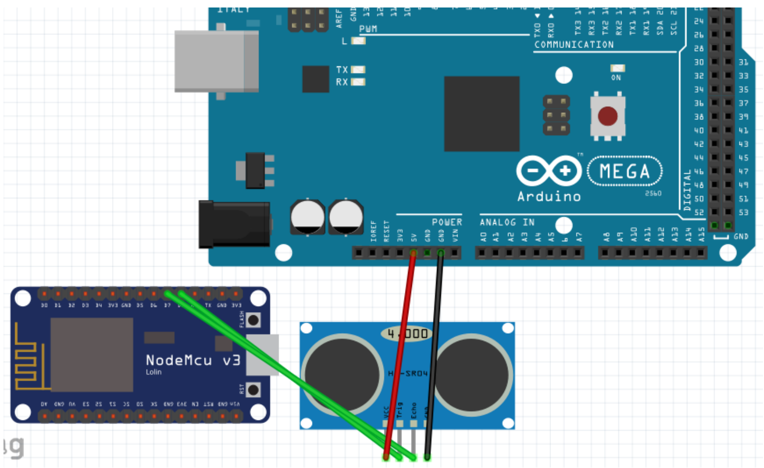


Figure 4-8 Ultrasonic sensor wiring with the NodeMCU Board

4.3.5. Serial communication wiring between the NodeMCU and Arduino Mega:

In order to implement the smart greenhouse’s parameters into the cloud (IoT), we need to establish communication between both boards so that the slave device (Arduino Mega 2650) sends and receives data from the master device (NodeMCU V3).

The wiring between the two boards is done like this:

- D1 (SCL) of NodeMCU needs to be connected to D20 (SDA) pin of the Arduino Mega.
- D2 (SDA) needs to be connected to the D21 (SCL) pin of the Arduino Mega.
- GND should be a common pin between both boards.

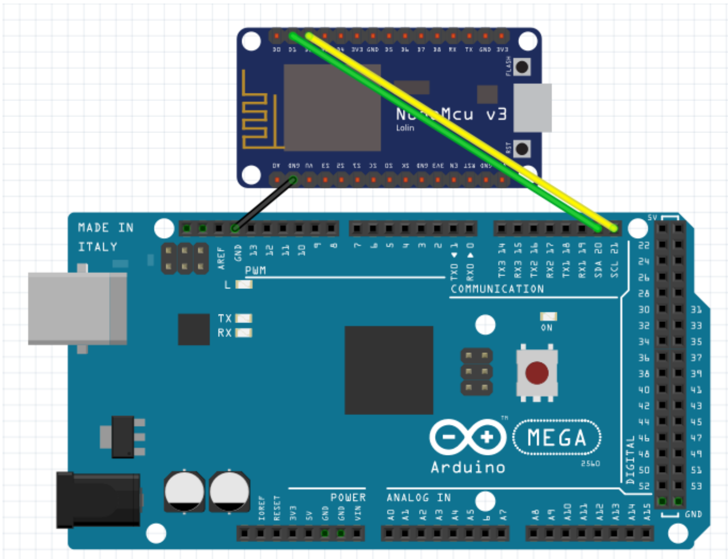


Figure 4-9 Serial communication between NodeMCU and Arduino Mega 2650

4.4. Finale test and results of the system:

In this part, we will study our finale system’s performance by testing the two sections of our system and showing the results.

4.4.1. Smart irrigation system test and results:

We will start testing each component alone:

a) Irrigation start:

Irrigation should start once the soil moisture drops below 60% threshold, we can monitor the percentage and the state of the water pump from the cloud.

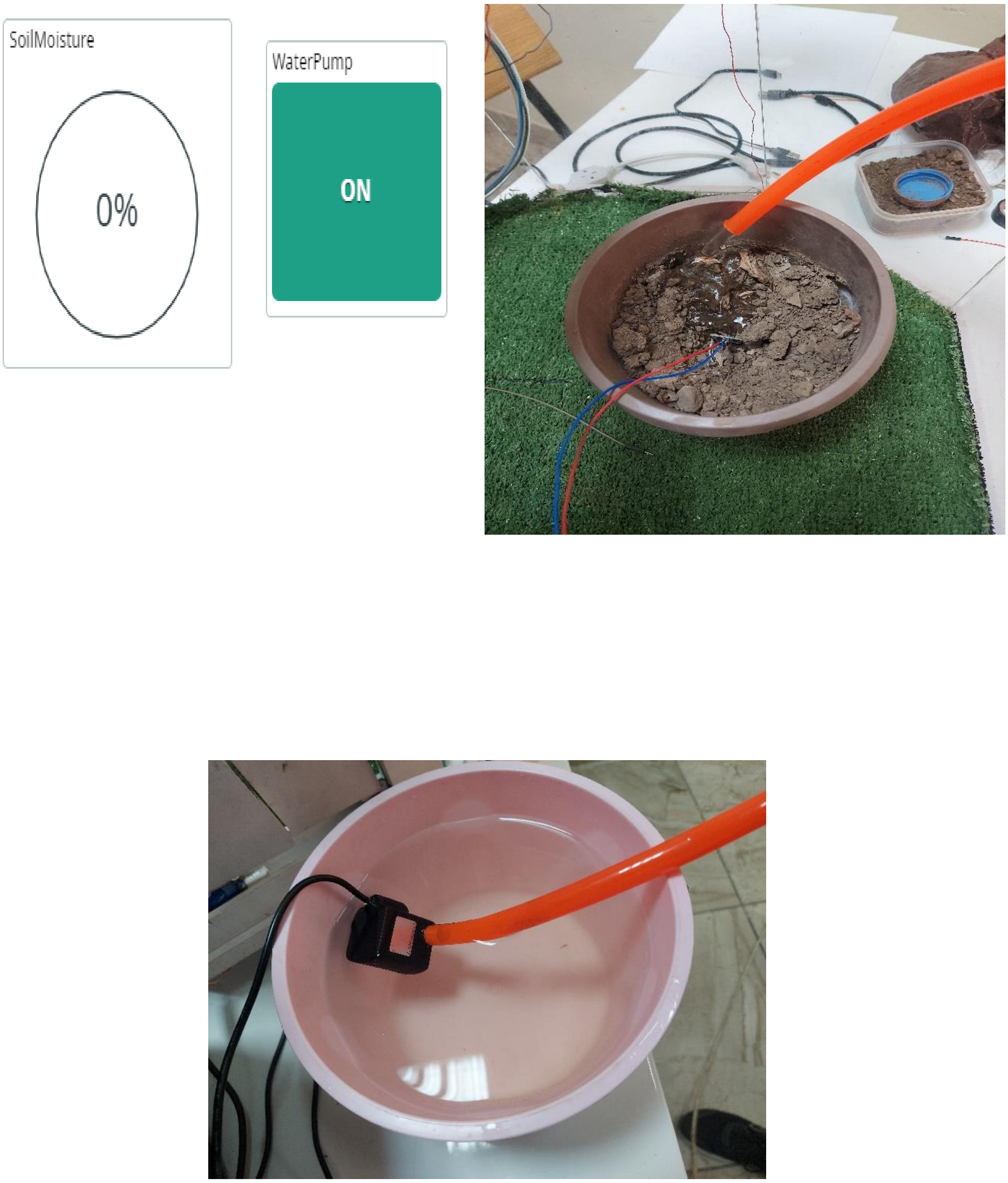


Figure 4-10 Irrigation start and monitoring

b) Irrigation stop:

Irrigation should stop once the soil moisture exceeds 60% threshold, we can monitor the parameters following that state from the cloud.

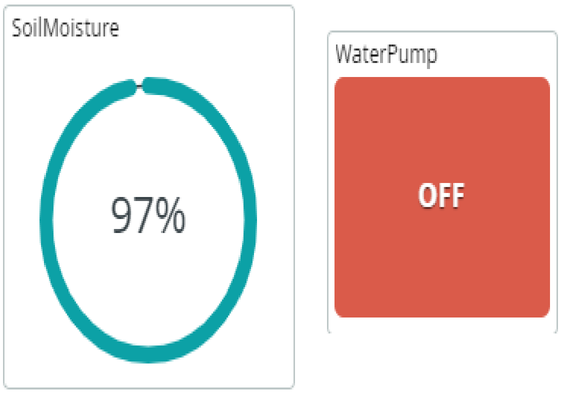


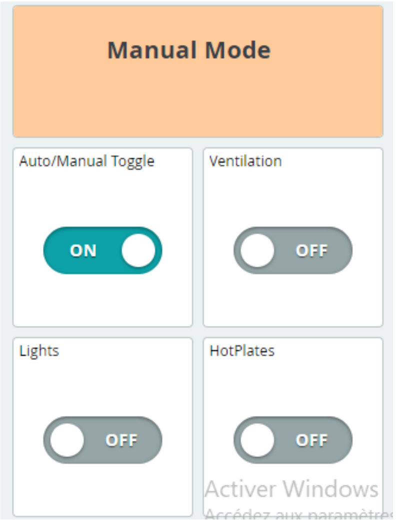
Figure 4-11 Irrigation stop and monitoring

4.4.2. Implementation of the smart greenhouse into the cloud test and results:

This part will show the test and results of the communication between the NodeMCU and Arduino Mega, which adds the greenhouse parameters to the cloud.

a) Greenhouse manual mode test and results:

Manual mode is controlled by the cloud, to test it we will press the auto/manual mode to switch from automatic to manual; an LED light connected to the Arduino Mega should light up indicating the manual override.



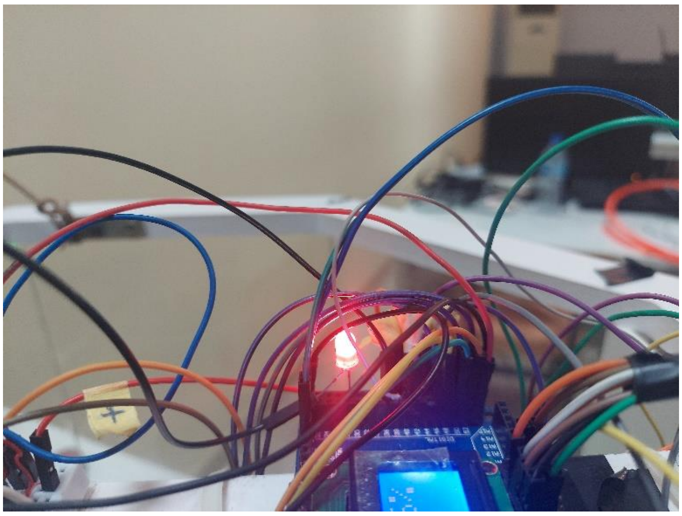


Figure 4-12 Manual mode activation

Following these results, we will then test turning on/off the lights, hotplates, and ventilation/doors, by switching their state from the cloud.

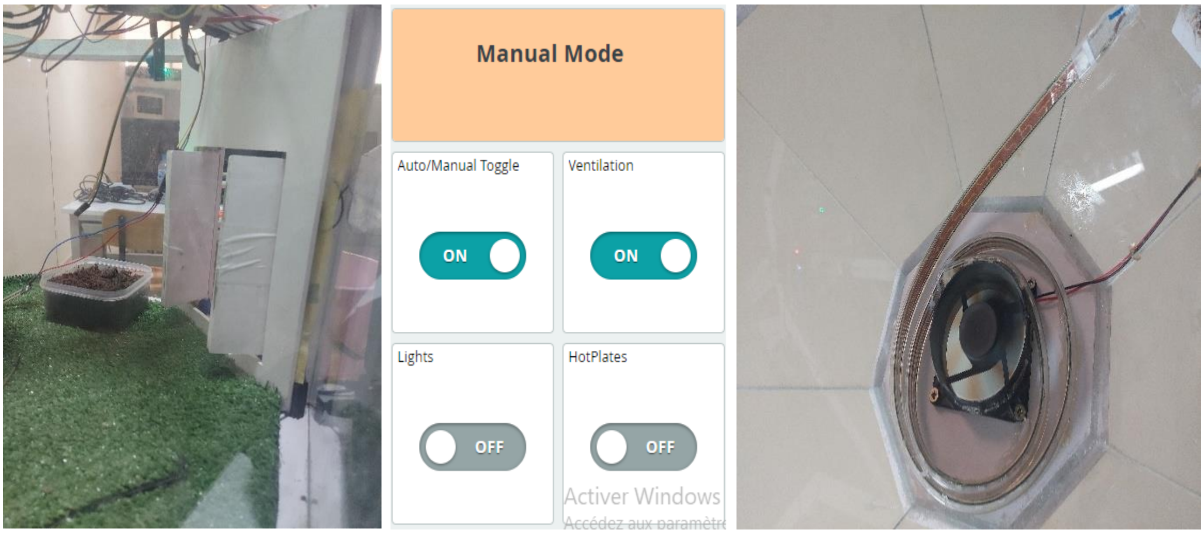


Figure 4-13 Ventilation manual mode on example

We conclude that the manual mode test shows the results we wanted.

b) Greenhouse automatic mode test and results

The cloud will monitor the state of the greenhouse while on automatic mode and its different parameters (temperature, humidity, lights, hotplates, ventilation/doors), to test it we will start the greenhouse on automatic mode and monitor the changes from the cloud, we provide them by changing temperature and lights thresholds.

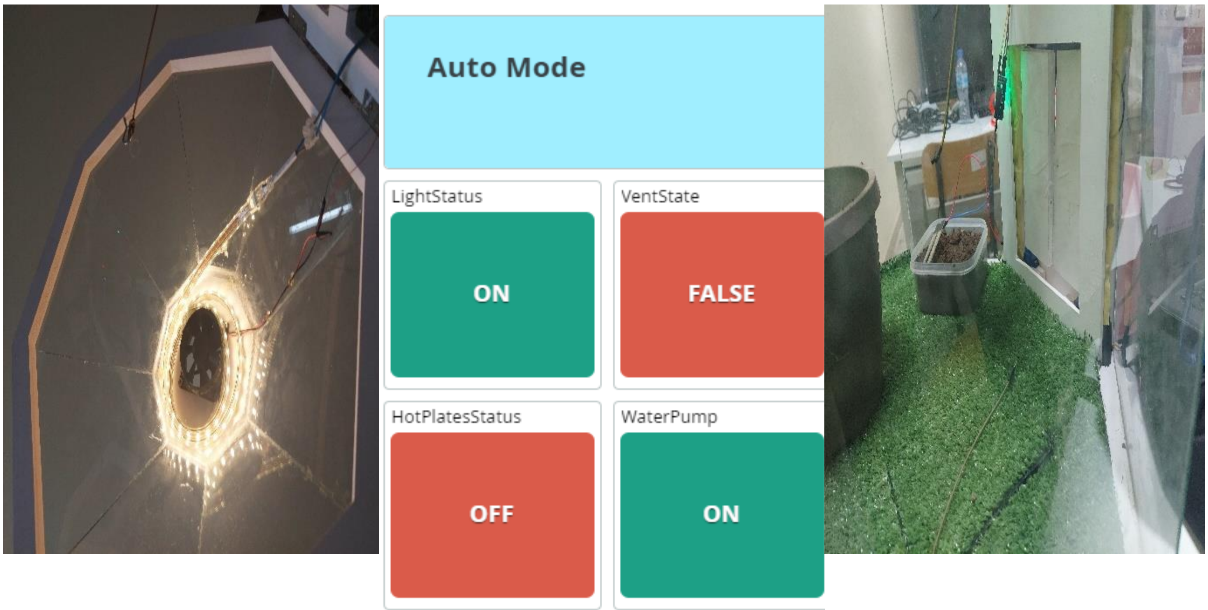


Figure 4-14 Ventilation and lights automatic mode example

We conclude that the automatic mode is successfully monitored from the cloud.

4.5. Conclusion:

In conclusion, after having satisfying test results, this chapter showed that we have established a smart environment where irrigation alongside the greenhouse can be monitored and controlled at any given time from distance.

Conclusion and perspective

Conclusion and perspective:

Our project initially aimed to develop an electronic embedded system. During the development period, we have tried our best to reach the envisioned objective: monitor and control a safe environment for agricultural practices. This project helped us develop our theoretical knowledge and gain a pleasant experience on the application aspect, we had the occasion to study and use different hardware and software and compensate between our knowledge and the information given.

At first we cited the different types of irrigation and specifications needed for a smart irrigation system then as we were including it in the Internet of Things, we have made it an additional objective to develop the university's already existing greenhouse and implement it to the cloud as well, which at the end given the results offered a successful smart agricultural environment which contains all the necessary parameters for a crop to grow into a desired plant.

As a perspective, the entirety of this system can be developed. Electronics is a vast field that offers intelligence, so this system can benefit in the future from the following additions:

Use of solar energy to power up the entire system.

Voice command addition, allowing the user to vocally command different parameters from distance.

Implementation of a specific sensor that senses the crop and soil's health, and alerts if needed to treat the soil with chemicals.

Creating this system with an automate.

Bibliography

Bibliography references:

[1] Wiki, 2022. [En ligne]. Available: https://en.wikipedia.org/wiki/Irrigation.

[2] Ranga, 2019. [En ligne]. Available: https://www.farmpractices.com/types-of-irrigation.

[3] I. Blog, 2021. [En ligne]. Available: https://fruitgrowers.com/the-3-main-types-of-irrigation-systems-for-agriculture/.

[4] Bidisha, 2020. [En ligne]. Available: https://civiltoday.com/water-resource-engineering/irrigation/287-advantages-and-disadvantages-surface-irrigation.

[5] Jain, 2021. [En ligne]. Available: https://www.jains.com/irrigation/sprinkler%20Irrigation%20systems.htm.

[6] TWC, 2022. [En ligne]. Available: https://thinkwatercanterbury.co.nz/services-and-solutions/irrigation/solid-set-fixed-grid-irrigation.

[7] Amend, 2005. [En ligne]. Available: https://www.nrcs.usda.gov/Internet/FSE_DOCUMENTS/nrcs141p2_017641.pdf.

[8] aac, 2021. [En ligne]. Available: https://allaboutcivil.org/advantages-and-disadvantages-of-sprinkler-irrigation/.

[9] E. Pedia, 2020. [En ligne]. Available: https://energypedia.info/wiki/Drip_Irrigation.

[10] H. Halepis, 2022. [En ligne]. Available: https://www.homequestionsanswered.com/what-is-sub-irrigation.htm.

[11] RitulShekar, 2018. [En ligne]. Available: https://brainly.in/question/3925872.

[12] wplawinc, 2022. [En ligne]. Available: https://wplawinc.com/5-factors-to-consider-when-choosing-an-irrigation-system.

[13] H. Jamal, 2017. [En ligne]. Available: https://www.aboutcivil.org/factors-affecting-crop-water-requirement.html.

[14] A. Team, 2018. [En ligne]. Available: https://www.acurite.com/blog/soil-moisture-guide-for-plants-and-vegetables.html.

[15] ESP, 2022. [En ligne]. Available: https://esp8266-shop.com/product/nodemcu-esp8266-esp-12e/.

[16] Components101, 2018. [En ligne]. Available: https://components101.com/wireless/esp12e-pinout-datasheet#:~:text=ESP%2D12E%20is%20a%20miniature,SoC%20(System%20on%20Chip)..

[17] R. Sheldon, 2022. [En ligne]. Available: https://www.techtarget.com/whatis/definition/sensor.

[18] smlease, 2022. [En ligne]. Available: https://www.smlease.com/entries/automation/what-are-different-types-of-sensors-and-their-applications/.

[19] Wiki, 2022. [En ligne]. Available: https://en.wikipedia.org/wiki/Soil_moisture_sensor.

[20] waveshare, 2021. [En ligne]. Available: https://www.waveshare.com/wiki/DHT11_Temperature-Humidity_Sensor.

[21] R. Santos, 2013. [En ligne]. Available: https://randomnerdtutorials.com/complete-guide-for-ultrasonic-sensor-hc-sr04/.

[22] oracle, 2021. [En ligne]. Available: https://www.oracle.com/internet-of-things/what-is-iot/.

[23] dfrobot, 2019. [En ligne]. Available: https://image.dfrobot.com/image/data/KIT0003/DHT11%20datasheet.pdf.

[24] electrobist, 2020. [En ligne]. Available: https://electrobist.com/product/hc-sr04-hc-sr04-ultrasonic-sensor/.

[25] components101, 2020. [En ligne]. Available: https://components101.com/sensors/dht11-temperature-sensor.

[26] curtocircuito, 2021. [En ligne]. Available: https://curtocircuito.com.br/datasheet/modulo/rele-1canal.pdf.

Annexes

NodeMCU ESP8266 Specifications & Features:

Development Board	NodeMCU ESP8266
Microcontroller	Tensilica 32-bit RISC CPU Xtensa LX106
Operating Voltage	3.3V
Input Voltage	7-12V
Digital I/O Pins (DIO)	16
Analog Input Pins (ADC)	1
UARTs	1
SPIs	1
I2Cs	1
Flash Memory	16 MB Max
SRAM	64 KB
Clock Speed	80~160MHz
802.11b mode	+19.5dBm output power
Weight	Small Sized module to fit smartly inside your IoT projects

Micro USB Port:

This part is used for communications between the board and the computer using a USB cable, this will allow us to program our board using a software or in our case a web site with integrated programming language.

Pins:

The NodeMCU Board has 2 categories of pins:

- **Power Pins**

The NodeMCU can be powered using a Micro USB jack and VIN pin (External Supply Pin)

The board operates at an input voltage of 7 to12 Volts, if it is powered up with less than 7 volts, the 3.3V Pin may provide less than 3.3V and the board might become unstable, if we use more than 12 V, the board will risk overheating and burning.

- **Input/output Pins**

NodeMCU has 16 digital pins that can be used as input-output using **pinMode (),digitalWrite ()** and **digitalRead ()**, some pins have certain functionalities. NodeMCU has 1 Analog input pin, to read its signal, Analog to Digital conversion is required. A NodeMCU has 10 bit ADC which means it scales an analog signal in a range of 0-1023, this can be measured using **analogRead()**, with the board including an internal voltage divider, the voltage range of the analog pin is 0V to 3.3V.

DHT 11 Specifications:

Table 3 DHT11 specifications

Model	DHT11
Power supply	3-5.5V DC
Output signal	digital signal via single-bus
Sensing element	Polymer resistor
Measuring range	humidity 20-90%RH; temperature 0-50 Celsius
Accuracy	humidity +-4%RH (Max +-5%RH); temperature +-2.0Celsius
Resolution or sensitivity	humidity 1%RH; temperature 0.1Celsius
Repeatability	humidity +-1%RH; temperature +-1Celsius
Humidity hysteresis	+-1%RH
Long-term Stability	+-0.5%RH/year
Sensing period	Average: 2s
Interchangeability	fully interchangeable
Dimensions	size 12*15.5*5.5mm

[23]

Where to use DHT11 Sensors:

The DHT11 is a commonly used Temperature and humidity sensor. The sensor comes with a dedicated NTC to measure temperature and an 8-bit microcontroller to output the values of temperature and humidity as serial data. The sensor is also factory calibrated and hence easy to interface with other microcontrollers.

The sensor can measure temperature from 0°C to 50°C and humidity from 20% to 90% with an accuracy of ±1°C and ±1%. So if you are looking to measure in this range then this sensor might be the right choice for you [24].

.Ultrasonic sensor specifications:

Table 4 Ultrasonic sensor specifications

Electrical Parameters	HC-SR04 Ultrasonic Module
Operating Voltage	DC-5V
Operating Current	15mA
Operating Frequency	40KHZ
Farthest Range	4m
Nearest Range	2cm
Measuring Angle	15 Degree
Input Trigger Signal	10us TTL pulse
Output Echo Signal	Output TTL level signal, proportional with range
Dimensions	45*20*15mm

[25]

5V Relay specifications:

Relay modules 1-channel features

• Contact current 10A and 250V AC or 30V DC.

• Each channel has indication LED.

• Coil voltage 12V per channel.

• Kit operating voltage 5-12 V

• Input signal 3-5 V for each channel.

• Three pins for normally open and closed for each channel.

How to connect relay module with Arduino:

As shown in relay working idea it depends on magnetic field generated from the coil so there is power

isolation between the coil and the switching pins so coils can be easily powered from Arduino by

connecting VCC and GND bins from Arduino kit to the relay module kit after that we choose Arduino

output pins depending on the number of relays needed in project designed and set these pins to output [26].

and make it out high (5V) to control the coil that allow controlling of switching process