An-Najah National University

Faculty Of Engineering And Information Technology

Computer Engineering Department

Gardenizer - Smart garden system

Students:
Amr Abdul Salam
Moayad Abu Zaroor

Supervisor:
Dr. Ashraf Armoush

June 18, 2023

Acknowledgements

“

First, We would like to thank everyone who helped in the realization of this project, represented by Dr. Ashraf Armoush, who did not hesitate to help us and receive us when we needed him and did not hesitate to give us useful advice to complete the project in the best way . In addition to thank the Department of Computer Engineering, whose doors were open to students.
Finally, we do not forget our parents who helped us get here to complete our educational journey.

”

-Amr , Moayad

Disclaimer

Contents

List of Figures

Chapter 1

Abstract

We have seen that many people who own agricultural lands far from their homes have difficulty taking care of these lands and do not insure strangers to enter their properties in addition to people who travel and own crops that need to be taken care of.
Therefore, we will create a system that takes care of the crops and enables the user to control many things remotely without having to reach the place of the plants and also to fill the water tanks without the need of the owner to be present by controlling it with iot system that enables the user to control the system remotely.
This project makes it easier for the owner to take care of the crops, in addition to saving the cost of labor. It also aims for continuously monitoring all the farms remotely and finally keeps the farms or houses more secured .
First we'll design a farm with many types of plants each plant will have a humidity sensor , water hose , water pump and a water ratiometer sensor in the water tanks to measure the ratio of water and when it's under 20 % send a text message for the water truck owner to provide water immediately for the tanks.
Also we are looking to add a way for fertilizing the plants remotely using the IOT modules and finally we are thinking of adding a light source for some plants to do the photosynthesis.
locally there is no similar project but globally there are similar projects but they don't have a system to call the water truck for water supply, in addition to the process of adding fertilizer automatically.

Chapter 2

Introduction

2.1 Background

Taking care of home gardens, whether they are near or far, or for those who own more than one garden, is a matter that requires time and effort and needs continuous follow-up. Therefore, the new technology has made it easier for us to control the garden remotely , automatically and schedule watering the plants at any time through the Internet IOT.

2.2 Problem Statement

The problem is that plants, by their nature, need constant care and require constant attention, such as watering,fertilizing etc. The owner may not always have time to do all this, which may lead to the plant withering.

2.3 Purpose

This project makes it easier for the owner to take care of the crops, in addition to saving the cost of labor. It also aims for continuously monitoring all the farms remotely and finally keeps the farms or houses more secured.

2.4 Objectives & Scope

Since this is a model project, you can use the web or mobile application to have a look to all the dashboard content that was read from the communication between the hardware components and cloud .
To meet the needs of the user,The system can do following :

1. Read the values of soil moisture , temperature , humidity , water level percentage , light intensity and notify the user when it starts raining .
2. Control water pump , manuring , outdoor light of the building , garage door and a light for the photosynthesis manually .
3. All the hardware components can be controlled automatically , also the user can schedule to irrigate the plants in a specific days and times of the week .
4. Send a text message to the owner if the water tank is almost empty .

2.5 Report Organization

- In the next chapter **(Constraints & Earlier Coursework)** , we will discuss what difficulties we faced in our project and how we were able to overcome them, in addition to the courses that helped us in this project.
- In the fourth chapter **(Literature Review)** , we will review some of the works and projects similar to ours and how our project differs from these projects.
- In the fifth chapter **(Methodology)** , we will discuss in detail the system process and what hardware components we used , how we connect these components and how it works, in addition to how we make communication between components and the application that control the system.
- The sixth chapter **(Results & Discussion)** is talk about the project results and what is the expected from project.
- In the final chapter **(Conclusion)** , we will briefly discuss the most important things that went through the project and what we learned from it, in addition to some recommendations that may be developed to be implemented in the future.

Chapter 3

Constraints & Earlier Coursework

3.1 Constraints & Limitations

On our way to accomplish this project, we faced many difficulties and challenges that stood in our face and required us to stand and confront them and get the best possible result.
The most important of these constraints are the following :

- Poor performance of some sensors, as they did not give the correct readings, which had to be changed in some cases or use other sensors.
- The connection between the ESPS , where it took some time until we learned how to deal with it and understand the basics of it, this is the first time we deal with ESPS.
- Choosing what is needed for the project and making a scenario to be applied by the project, as each time something new came to our mind and we added it or modified something that was present, which wasted some time for us until we reached the last scenario that we applied.
- The late arrival of some parts to the supplier, as days passed and the shops did not have what we needed, so we were waiting for their arrival.
- The high prices of some equipment, as we were often forced to buy equipment at high prices instead of waiting and buying them at low prices.
- The difficulty of dealing with the GSM module, as we did not find many who used it before, in addition to adding it to our project, it showed many unexpected problems. When we were using GSM , the process of communicating and sending data stopped.Until we finally managed to reach the solution so that everything works regularly And without problems.
- The weight of the project, as the design was heavy and its movement was difficult, so we could not easily find a place to work on it.
- Lack of time, as this project was completed in a summer semester, which is rather short and requires work almost daily.

3.2 Standards

We used some common ready-made techniques in our project to make the work easier :

- Arduino IDE which is the compiler that provide us to upload the code into boards after connecting them .
- We use ESP-NOW protocol for communication between the ESPs which enables multiple devices to communicate with one another without using Wi-Fi .
- We use MQTT protocol for communication between ESP and the application . MQTT allows for messaging between device to cloud and cloud to device .
- We use Arduino IoT Cloud as a web and mobile application to control the system .

3.3 Earlier Coursework

There are many courses taught in the Department of Computer Engineering that we benefited from in our project, in addition to external courses and online videos.
The following are some examples :

- Microcontroller projects , Digital lab and CPU lab which helped us understand how to connect the components.
- Wireless and its project which helped us how the communication does between component and Internet.
- IEEE Arduino workshops that learn how to deal with boards like Arduino.
- Online videos that explain how to deal with some components such as ESP and GSM modules.

Chapter 4

Literature Review

Gardenizer is a system that aims to keep the owner in constant contact with his garden and perform routine tasks without the need to attend in person. Gardenizer is not the first project aimed at controlling home gardens remotely, as there are many projects similar to it that have been developed previously :

1. **Smart Indoor Farming System** :
   This project is implemented as graduation project at An-Najah National University , this project include some sensors like temperature , soil moisture to control water pump, fan and light intensity sensor to turn on/off lights .

2. **Smart Garden Monitoring and Control System with Sensor Technology**
   (Conference: 2021 3rd International Conference on Signal Processing and Communication (ICPSC)) this projects also have a Temperature and humidity sensors , soil mositure sensor , driver , DC motor and a fan .

In our project we explored the obstacles and problems that were in the previous projects and tried to improve them. For example, a machine was added to add fertilizer to the plant, in addition to an alert system for the owner so that he would send a text message (using GSM module) to him if the water tank was empty.

We have also scheduling feature that scheduled garden control as set by the owner where he can set the process on different days and at different hours and can repeat it according to what he wants.

In addition, in our project, we can control the processes in the system from pumping water, controlling the gate, and so on, even if the micro-controller that contain sensors are broken or stopped. In our project, we separated the micro-controllers from each other and each one works separately that performs operations so that our system does not work as master - slave .

Chapter 5

Methodology

In this chapter we will discuss in detail the system process and what hardware components we used, how we connect these components and how it works, in addition to how we make communication between components and the application that control the system.

5.1 Hardware Components

In this section we are going to show how the hardware components are connected together and how they work .

5.1.1 ESP32 Microcontroller

ESP32 is a series of low-cost , low-power system on a chip microcontroller with integrated Wi-Fi and dual-mode Bluetooth it also a built-in antenna switches , RF balun , power amplifier , low-noise receive amplifier and power-management modules.

We choose ESP32 in our system because it has a built-in Wi-Fi module so it can be controlled from any place in the world and send the data to the cloud,also it has 38 pins to connect all sensors needed for system.

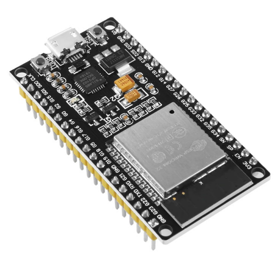


Figure 5.1: ESP32 Microcontroller

5.1.2 GSM Module

Global System for Mobile Communication is (GSM) is a standard used in cellular communication networks. It offer connectivity along with wireless data communication transfer it is lightweight device with low power consumption .

We used the GSM module in our project so when the water tank is almost empty (when the water level sensor is less than 10%) it sends a text message to the owner and informed him that the water tank must be fill in immediately .

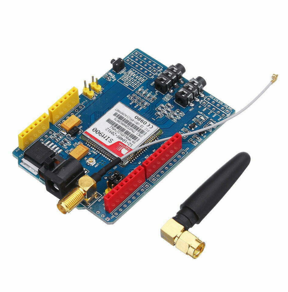


Figure 5.2: GSM Module

5.1.3 L298N Motor Driver Module

L298N Motor Driver Module is a high power motor driver module . This module consists of an L298 motor driver IC and a 78M05 5V regulator. L298N Module can control up to two motors by using the enable signals.

In our system the L298N driver is used to control the water pump , led strip and outdoor lights.When the enable lines receives a signal from the web/mobile application it either connect the voltage to the object else cut the voltage from it .

Figure 5.3: L298N Motor Driver

5.1.4 DHT Sensor

DHT sensor is basically used for measuring the temperature & humidity.Our system uses DHT11 type we chose it rather the DHT22 sensor because it's sampling rate is 1Hz (reading values every 1 second) while the DHT22 is 0.5Hz (reading values every two seconds) .

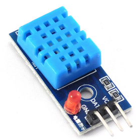


Figure 5.4: DHT11 Sensor

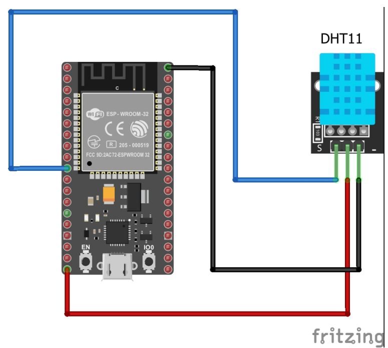


Figure 5.5: DHT11 Sensor Connection

5.1.5 Raindrop Sensor

Raindrop sensor is basically a board on which nickel is coated in the form of lines. It works on the principal of resistance. We use it to inform the owner that it started raining.

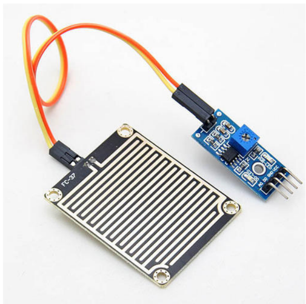


Figure 5.6: Raindrop Sensor

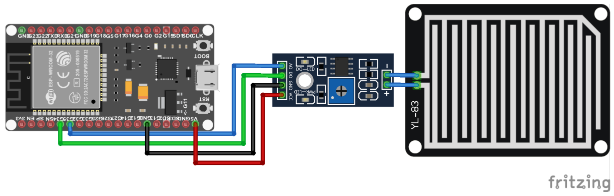


Figure 5.7: Raindrop Sensor Connection

5.1.6 Soil Moisture Sensor

Soil Moisture sensor measures the volumetric content of water inside the soil and gives us the moisture level as output. This soil moisture sensor measures soil moisture levels by capacitive sensing rather than resistive sensing like other sensors on the market.

Figure 5.8: Soil Moisture Sensor

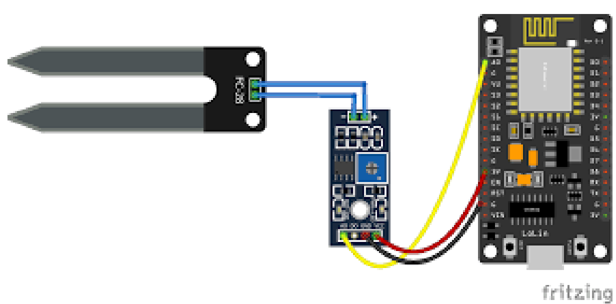


Figure 5.9: Soil Moisture Connection

5.1.7 LDR Sensor

LDR sensor is used to measure the light on the planets so it can deiced if the planet needs light for the photosynthesis or not .

Figure 5.10: LDR Sensor

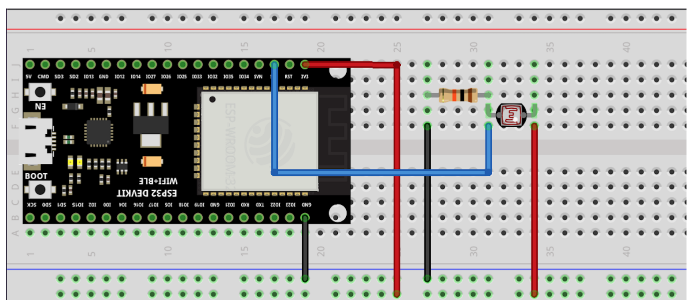


Figure 5.11: LDR Sensor Connection

5.1.8 Led Strip

Led strip used to increase or decrease the intensity of the light around the plant by turning on or off depending on what light intensity each plant needs and what the current light intensity around the plant, turning on/off the led strip is done by connecting it with a motor driver . If the light intensity is less than the light intensity that the plant need, the led strip is turn on, else, it turns off.

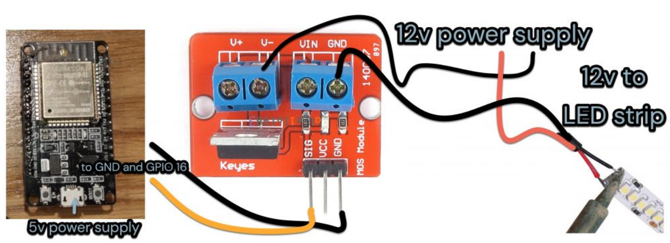


Figure 5.12: Led Strip Connection

5.1.9 Water Pump

Our system contains a water tank and inside it there is a water pump also with a water level sensor. Water pump simply pulls the water from the water tank and water the plants using the plastic tube. Water pump can be controlled by many ways using auto-mode , manually by using a switch button on the website and can also be controlled by the schedule feature (water the plants in specific days and times).

Figure 5.13: Mini Water Pump

And this is the water pump connection with ESP32 wired to the motor driver.It start pulling water from the tank when the driver received a signal from website/mobile application it connect the water pump to the voltage source and send water from the tank to the plants using the tube.

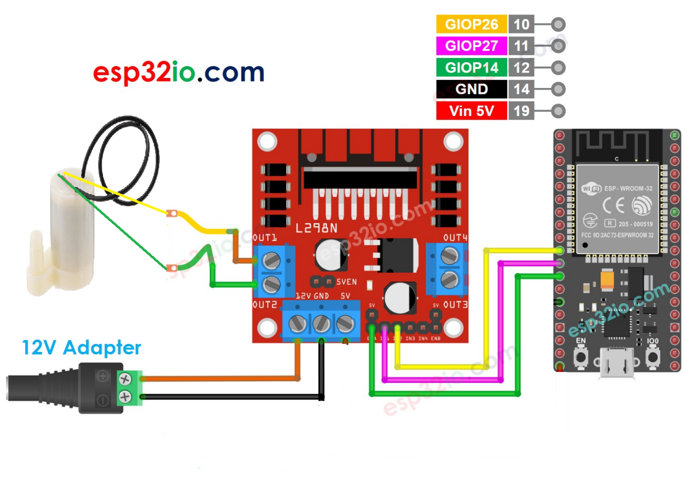


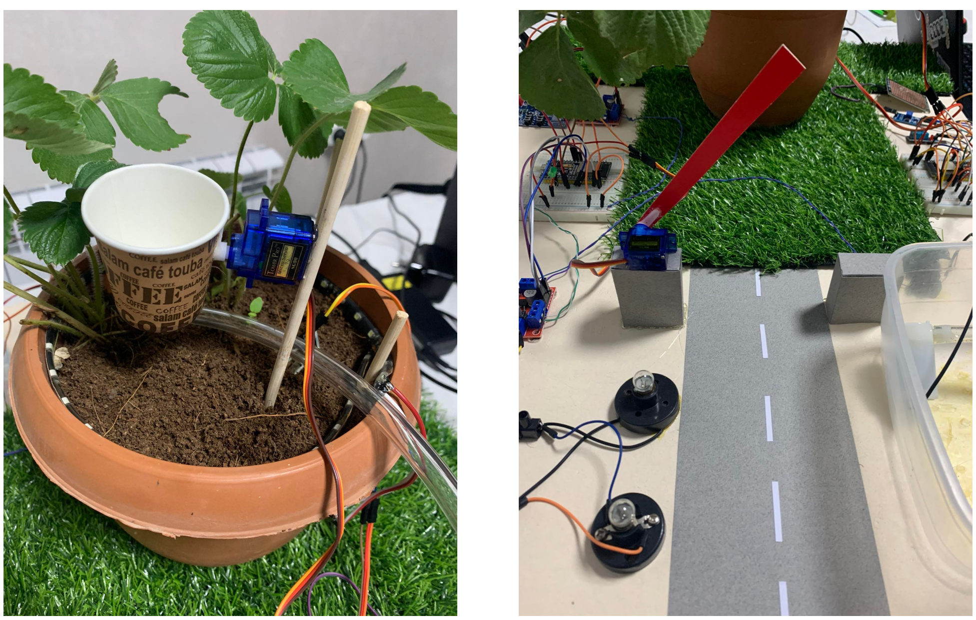
Figure 5.14: Water Pump Connection

5.1.10 Servo Motor

A servo motor is a type of motor that can rotate with great precision. Normally this type of motor consists of a control circuit that provides feedback on the current position of the motor shaft.

Our system uses two servo motors one is used for opening and closing the garage door and the other is engraved in the soil and responsible for fertilizing the plant .

Figure 5.15: Servo Motor



(a) label 1 (b) label 2

Figure 5.16: Garage & fertilizing

5.1.11 Water Level Sensor

The water level sensor is a device that measures the liquid level in a fixed container that is too high or too low. We use it in our project to measure the level ratio of water in water tank and display it on application.

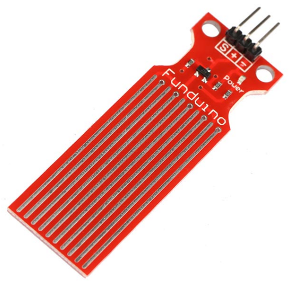


Figure 5.17: Water level sensor

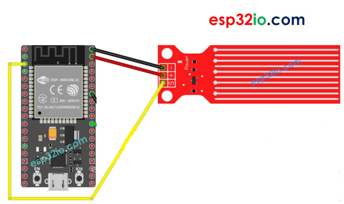


Figure 5.18: Water level sensor connection

5.1.12 Light Bulb

We connected two light bulbs in parallel and connected them to a driver and ESP to control them via application.

Figure 5.19: Light bulb

5.2 System Design

Basically Gardenizer (Garden Organizer) system have an entrance for the garden and a garage that can be opened using the switch button which can be controlled from the web/mobile application also the user can light the bulbs next to the garage door using the switch button also.

The owner have access to garden from any where in the world since our project is based on IOT systems. The owner also can be informed if the water level in the tank is low (under 10%).Using the GSM module the owner will receive a text message informing him that the tank need to re-fill in.

The plants is surrounded with a led strip which is needed for the photosynthesise. It can be light up in 2 ways either using the switch button else from the auto-mode. Auto-mode simply keeps reading the light values from the LDR Sensor. When it read values below 10 (night mode) it will automatically turn on the led strip.

Auto-mode can also be used to water the plants. Simply it reads values from the soil moisture sensor and when its value is above the 60% it means the plant needs water.

Rain detection is also found in the system.When it start raining the raindrop sensor indicates that rain started and inform the user with a check sign under its status.

Our system also contains fertilizing process. Its a servo motor engraved in the soil holding a cup with fertilizer.

Our system contains two ESP32 connected to the cloud and transferring data between them.

5.2.1 Controller - 1

- Water Level Sensor.
- Soil Moisture Sensor.
- Raindrop Sensor.
- LDR Sensor.
- DHT Sensor
- Servo Motor (used for fertilizing).

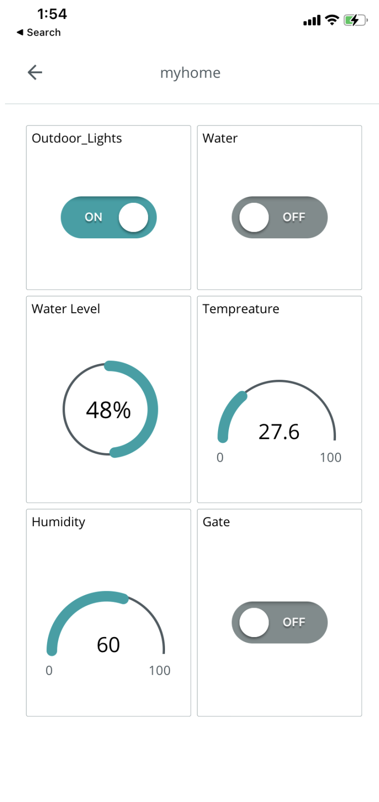
5.2.2 Controller - 2

- Water Pump.
- Servo Motor (used for garage).
- Light Bulbs.
- Led Strip.
- GSM Module (for sending messages).

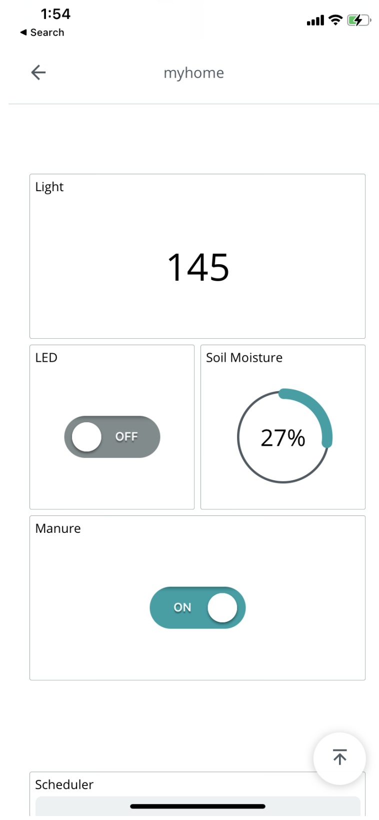
5.3 Software

Our system can easily be controlled from a web and mobile application.The application is user friendly and easy to use.
It contains a dashboard showing all the sensors percentage with a control buttons for turning on water pump , led strip , gate , fertilizing and auto-mode.

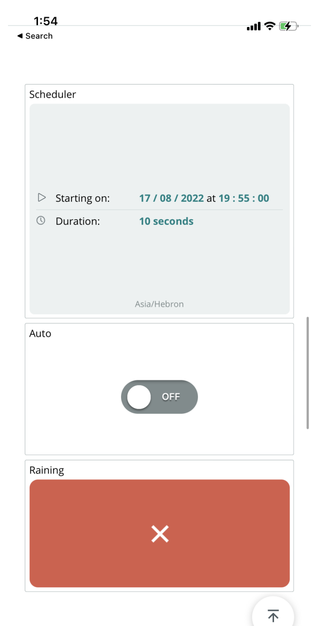
Figure 5.20: Web View



(a) label 1



(b) label 2



(c) label 3

Figure 5.21: Mobile View

5.3.1 Auto-Mode

Our system also have an auto-mode option that read values of sensors and control some components automatically. So, if the auto-mode is activated the system check the rain status and soil moisture sensor value, if it does not raining and soil moisture is more than 60% then water pump will turn on, and check the LDR sensor value if less than 10 then the LED strip will turn on.

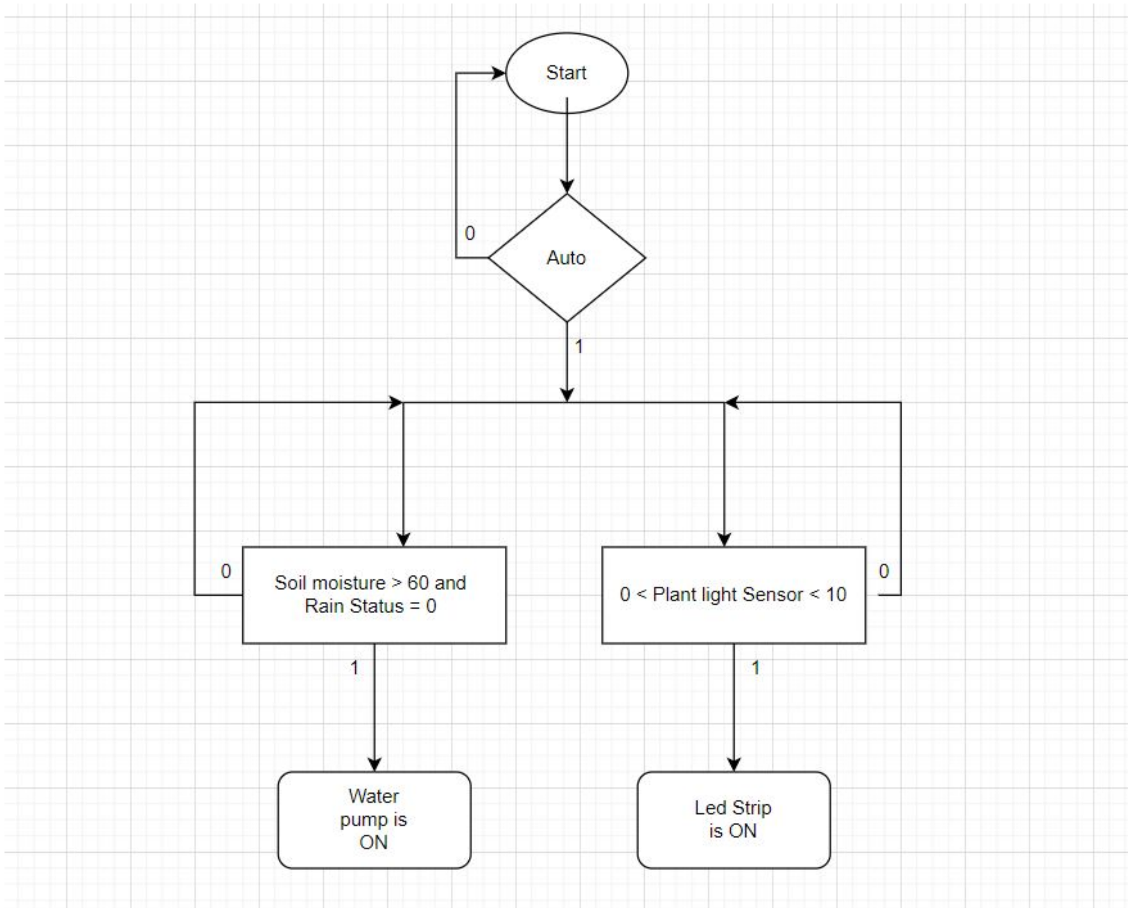


Figure 5.22: ASM chart - Auto Mode

Auto Mode Code :

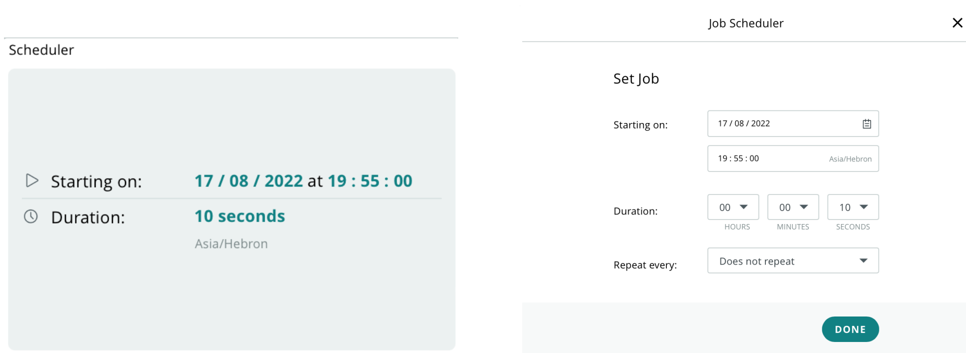
```
if(autoo){
if(myData.soil_mon>60 && myData.rain==0){
  digitalWrite(water_pump, HIGH);
  digitalWrite(water_pump2, LOW);
  Serial.println("pump is ON");
}
else{
  digitalWrite(water_pump, LOW);
  digitalWrite(water_pump2, LOW);
  Serial.println("pump is OFF");
}

if(myData.plant_light>=0 && myData.plant_light<10){
  digitalWrite(led1, HIGH);
  digitalWrite(led2, LOW);
  Serial.println("light is ON");
}
else{
  digitalWrite(led1, LOW);
  digitalWrite(led2, LOW);
  Serial.println("light is OFF");
}
}
```

Figure 5.23: Auto Mode Code

5.3.2 Schedule Process

In our system, user can schedule some processes like watering plants from water pump. The user can specify the days and times he wants to run the water pump without the need for manual control or activating the automatic mode.



(a) label 1 (b) label 2

Figure 5.24: Schedule option

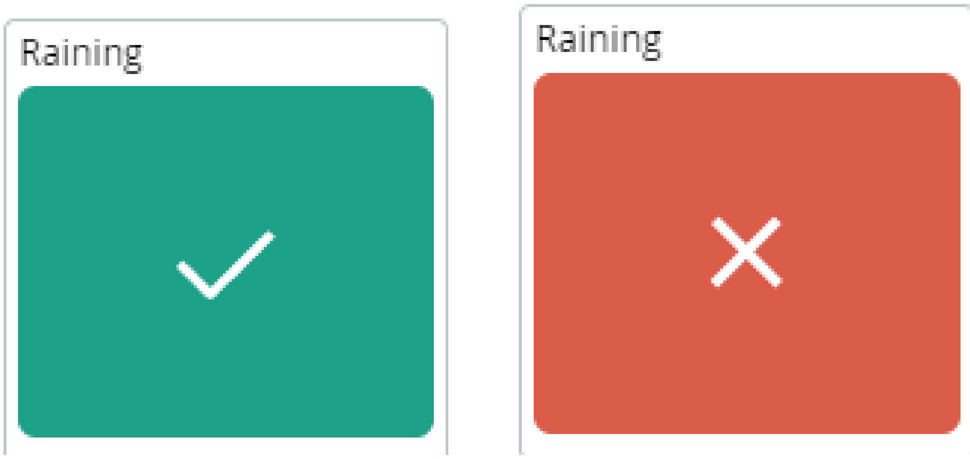
Schedule Code :

```
  if (systemschedule.isActive()) {
   Serial.print("inside scheduale ");
   myData.waterPumpOn = true;
   delay(1000);
   flag_sch = 0;
    }
    else if(flag_sch == 0) {
      Serial.print("false :(");
      myData.waterPumpOn = false;
      delay(1000);
        flag_sch = 1;
    }
```

Figure 5.25: Schedule Code

5.3.3 Raining Process

We add a rain sensor that reading and notify if there is rain or not, if it raining then the irrigation process in auto mode will not activate. Notification is appear in the application like this :



(a) Raining (b) Not Raining

Figure 5.26: Rain Status in Application

```
//for rain drop sesnor
rainstatus = analogRead(RAINSTA);
//rainbool = rainstatus;
if(rainstatus > 2800){
  rainbool = false;
  myData.rain = false;
 }
 else{
  rainbool = true;
  myData.rain = true;
  }
Serial.print("Rain Status = ");
Serial.println(rainbool);
```

Figure 5.27: Raining Code

5.4 Communication between ESPS

Since our system is based on IOT the ESPS in our system sends the need data between them.
To make the communication process we used the ESP-NOW protocol to make it happen .We used this type of protocol because it enables multiple devices to communicate with one another without using Wi-Fi.
Pairing between devices is needed prior to their communication. After the pairing is done, the connection is secure and peer-to-peer, with no handshake being required.
This means that after pairing a device with each other, the connection is persistent. In other words, if suddenly one of your boards loses power or resets, when it restarts, it will automatically connect to its peer to continue the communication.

ESP-NOW supports the following features :

- Encrypted and unencrypted unicast communication.
- Mixed encrypted and unencrypted peer devices.
- Up to 250-byte payload can be carried.
- Sending callback function that can be set to inform the application layer of transmission success or failure.
- It can deliver data up to 250m (distance between ESPS).

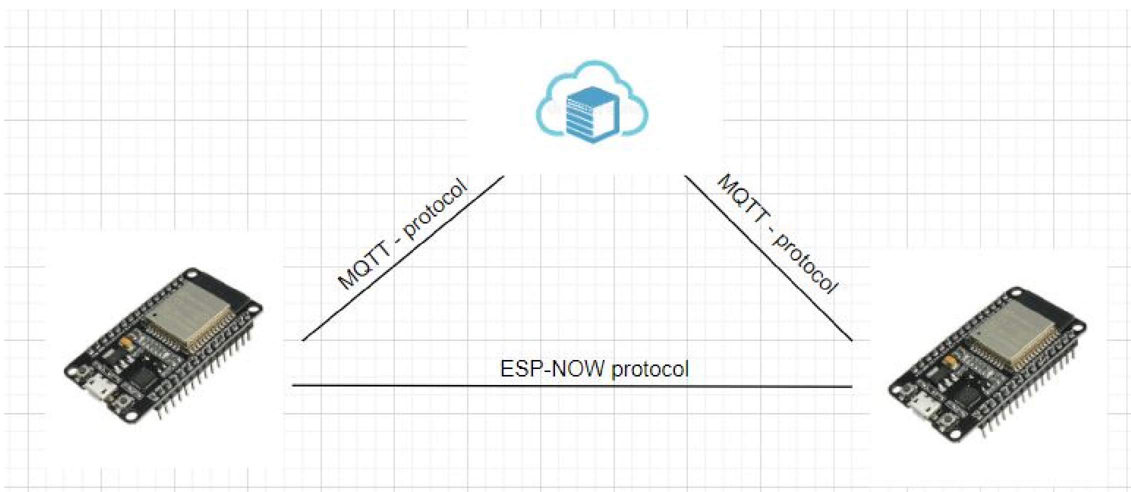


Figure 5.28: Communication diagram

To make the communication first we need to know the MAC address for each ESP.We do this by using a simple code uploaded to each ESP.

MAC adress for ESP:

- **ESP-1 :** 78-E3-6D-12-29-24 .
- **ESP-2 :** 78-E3-6D-09-36-E0 .

The following picture contains the type of data needed to be send(Sensor values and motors) with the MAC address to send the data for it.

```
uint8_t broadcastAddress[] = {0x78, 0xE3, 0x6D, 0x09, 0x36, 0xE0};
typedef struct struct_message {
  char esp_mac[32];
  float humidity;
  bool outdoor_light ;
  bool gate ;
  int water_level;
  int soil_mon;
  int plant_light;
  float temperature;
  bool rain;
  bool waterPumpOn;
} struct_message;
```

Figure 5.29: Communication structure

We need to make an object from the structure above so we can send data to the other ESP called ”myData”.

```
water =analogRead(poto);
watersensor =  map(water , 0 , 1500 ,0 , 100);
strcpy(myData.esp_mac, "Sent From ESP32-1 ");
myData.humidity = dht.readHumidity();;
myData.water_level = watersensor;
myData.temperature = dht.readTemperature();
//read soil mo .. sensor ;
output_value= analogRead(LIGHT);
//output_value = map(output_value,0,1000,0,100);

myData.plant_light = light;
```

Figure 5.30: Sent Data

In the loop() function , we'll send a message via ESP-NOW every 2 seconds (2000 milliseconds). This what shows when the data is sent successfully from the sender side :

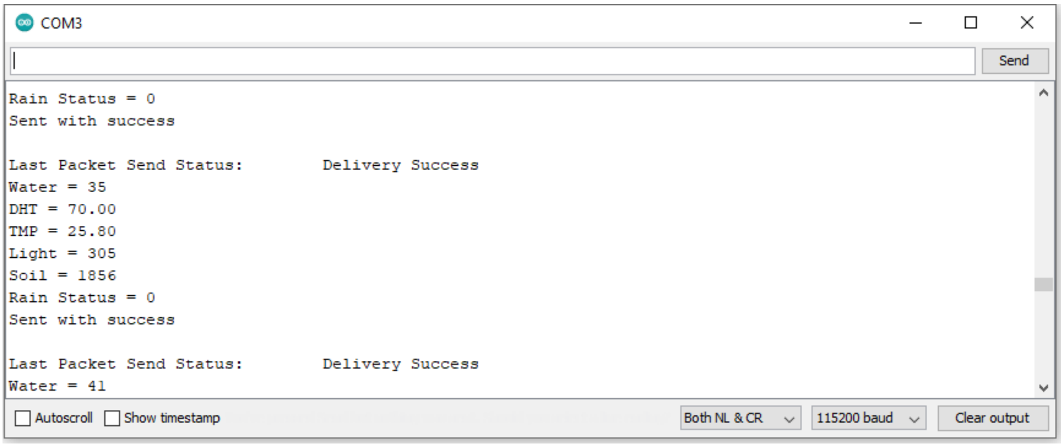


Figure 5.31: Sending Data

And this is the received data :

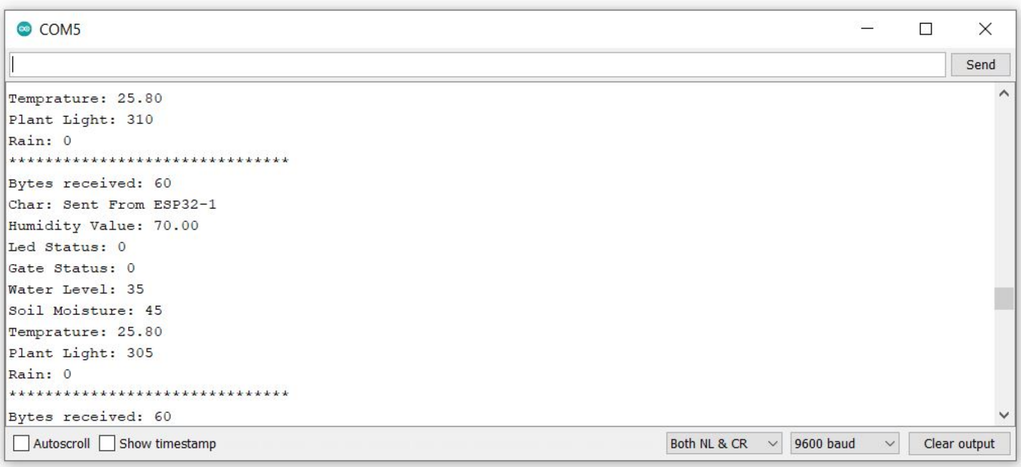


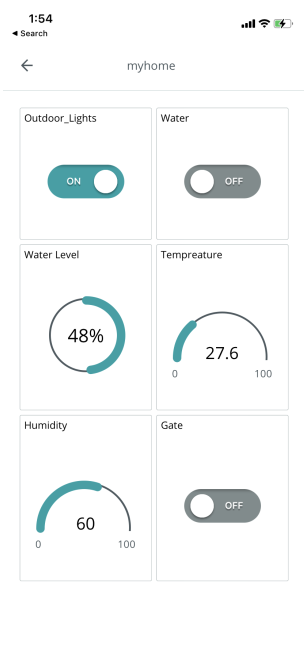
Figure 5.32: Received Data

Chapter 6

Results & Discussion

In this chapter we will talk about the project results and what is the expected from project.The final produced project is a smart garden system with IoT that control a simple garden that provides user to control it without need to be attended.

As a result, we made a web/mobile application that show all sensors values like temperature, humidity, water level, rain status, soil moisture and light intensity in addition to control components like water pump, lights, led strip, garden gate and fertilizing process .



(a) Mobile View

(b) Web View

Figure 6.1: Application View

We made a simple garden design so that we can put all the sensors together and the pieces that we talked about previously, in addition to the main equipment such as the ESP, the GSM and many other tools and we run them to see a complete system with all its contents and see the result in reality.

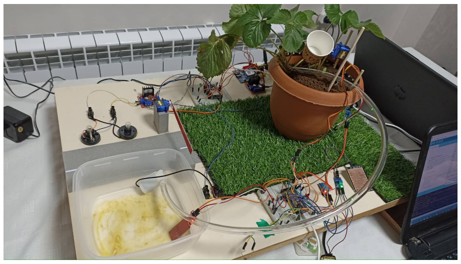


Figure 6.2: Side 1 view

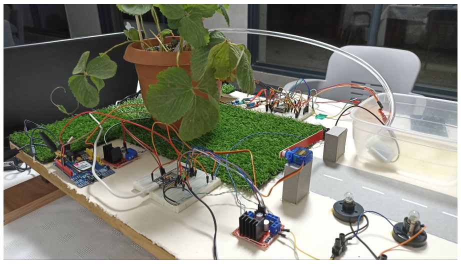


Figure 6.3: Side 2 view

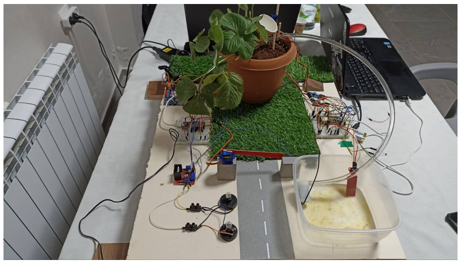


Figure 6.4: Top view

Chapter 7

Conclusion

7.1 Summary

In the end, we make a smart garden system that make the user able to control it remotely if he was out and to save time & effort and more comfortable. The system help user to take care of plants manually with web/mobile application.
Our system can be controlled manually as explained before or automatically according to readings of the sensors. Process also can be scheduled according to what user want.

7.2 What did we learn ?

While building this project, we learn how to deal with ESP and make serial communications between them with ESP-NOW protocol and how to make communication with cloud using MQTT protocol.In addition to learn how We dealt for the first time how to send SMS messages from ESP to mobile using GSM module. We learn how to connect many sensors and components to one system and control it from the internet (IoT).

7.3 Recommendation

To improve our system maybe we can add a raspberry pi and more ESPs, each ESP have a similar one and if the first one was stopped, then the raspberry pi can detect it and change communication to the other ESP.
We suggest also to add a camera and using AI algorithm, machine-learning and image processing techniques to detect if the plant need water or if the plant has a disease.

7.4 Future Work

We hope we can add more functionalities to this project like send messages to system from user using the GSM to send commands not just controlling from application. We can also add more units to control more plants and modify it to control large trees.

Chapter 8

References

[1] "Send Receive SMS Call with SIM900 GSM Shield Arduino" , https://lastminuteengineers.com/sim900-gsm-shield-arduino-tutorial/

[2] "How to use GSM module with Arduino" , https://www.rkichu.com/how-to-use-gsm-module-with-arduino/

[3] "Understanding Natural Light for Houseplants" , https://www.thespruce.com/lighting-for-houseplants-1902691

[4] "Get ESP32/ESP8266 MAC Address and Change It" , https://randomnerdtutorials.com/get-change-esp32-esp8266-mac-address-arduino/

[5] "How Soil Moisture Can Affect Your Plant's Growth" , https://www.greenwaybiotech.com/blogs/gardening-articles/how-soil-moisture-affects-your-plants-growth

[6] "Getting Started with ESP-NOW" , https://randomnerdtutorials.com/esp-now-esp32-arduino-ide/

[7] "INTERFACING OF DHT11 SENSOR WITH ESP8266 NODEMCU", https://techatronic.com/interfacing-of-dht11-sensor-with-esp8266-nodemcu/

[8] "Stack Over Flow to solve our problems" , https://stackoverflow.com/

[9] "Connecting the Servo Motor to the ESP32" , https://randomnerdtutorials.com/esp32-servo-motor-web-server-arduino-ide/

[10] "How to use L298n Driver " , https://create.arduino.cc/projecthub/ryanchan/how-to-use-the-l298n-motor-driver-b124c5