# IoT Based Smart Gardening

**A PROJECT REPORT**

***Submitted by***

**Josiah James – 20BCE1344**
**Ayesha Numa – 20BCE1290**

**CSE3009 – Internet of Things (EPJ)**
**Slot- (C1+TC1)/(TC2+TC2)**

***Project Guide***

***Dr. Manjula***
***Associate Professor***
***School of***

**BTECH**
**IN**
COMPUTER SCIENCE AND ENGINEERING

VIT®
Vellore Institute of Technology
(Deemed to be University under section 3 of UGC Act, 1956)

**Winter Semester 2022-23**

# ABSTRACT

Rapid technological developments have been applied in various sectors of life, one of which is in the agricultural sector. By applying technology in the agricultural sector, it can reduce energy and time wasted due to the application of conventional methods. One of the innovations that are currently trending in technology is the Internet of Things (IoT). Internet of Things (IoT) is a technology that allows physical objects to be connected to the internet in real time. This research applies the Internet of Things to the garden irrigation system, by remotely controlling the water pump and monitoring soil moisture in the garden. Using the application of the Internet of Things the garden owners can measure and detect soil moisture in their plantations. Then efficiently, we can manage the use of water in real-time. The water supply inside the garden is connected to a water pump that will be activated when the soil moisture sensor detects low moisture level, and will automatically adjust the moisture parameter to its optimum number. In addition, the owner of the garden can monitor the condition of soil moisture through a mobile app.

**TABLE OF CONTENTS**

# LIST OF FIGURES

# INTRODUCTION

## 1.1 Objectives and Goals

Many of the plants that grow in their natural environment are subject to unfavourable conditions that hinder their growth to full potential . So the system that we are planning to develop will take special care and cater the needs of the individual species of plants and provide them with the ideal environment that they require. Different species require different conditions, nutrients, sunlight, moisture and various other things which are quite difficult to manage manually. So a system that can take care of this will ease the burden on the person as well as make the process much more efficient. The major goal of this project is to reduce water consumption when gardening and to maintain the garden remotely.

## 1.2 Application

This project can be applied in a plethora of varying fields where you want to control the sensors ,and different components which help in gardening from remote place

For Example:

- Everyone likes to maintain a small garden in their home , so they can use this project to maintain healthy plants from remote places
- This can also be used in farming where a farmer can control watering , giving pesticides to crops etc. from his house.

## 1.3 Features

- A Moisture Sensor is installed, if the motor is set in automatic mode based on moisture in the soil the motor is turned on and off.
- The level of moisture can be monitored through the dashboard created in Arduino cloud. It is displayed in various formats like value. gauge, chart etc. which makes it user friendly and readable.
- A temperature and humidity sensor is also installed which detects the temperature and returns value to the dashboard
- The motor can also be set to manual mode and be controlled manually by using the switch that can again be accessed using Arduino Cloud.

# LITERATURE REVIEW

1.

**(i) Title, Author, Date of publication and Publisher Information**

Smart Garden Monitoring System Using IOT

*T.Thamaraimanalan , S.P.Vivekk, G.Satheeshkumar and P.Saravanan* IEEE,pp.5- 10,2018.

**(ii) Problem and Objectives**

- To maintain the nature of the plants by continuously monitoring the parameters leading to the increased life of both plants and human beings.
- To build software that can automate the watering process.
- To build a system that reduces cost and ensures safety

**(iii) Methodology Adopted**

The device that was used was using sensors that were programmed with a microcontroller and linked with a mobile application to transfer the data that is collected from the device. The development of the prototype consisted of both hardware and software, along with a communication medium. The emphasis is more on the hardware side as manufacturing of devices from components which are vital for the project's purpose. The proposed design was a portable and area independent automated system that collected information about garden health with sensors

**(iv) Performance Limitations**

Excessive seepage and leakage of water form marshes and ponds all along the channels. The marshes and the ponds, in the course of time become the colonies of the mosquito, which gives rise to a disease like malaria.

2.

**(i) Title, Author, Date of publication and Publisher Information**

Plant Watering and Monitoring System using IOT and Cloud Computing

*Ms. Yogeshwari Barhate, Mr. Rupesh Borse, Ms. Neha Adkar, Mr. Gaurav Bagul*

Sandip Institute of Technology and Research Center, Nashik, India, April 2020 IJSDR

### (ii) Problem and Objectives

Problems:

- The need for automation is increasing day by day the absence of which leads to water wastage in many cases.
- Other automated solutions in the market requires time dependent motor activation control for watering the plants

Objectives:

- The primary objective is to reduce human intervention and avoid water wastage by watering the plants as per the requirements. Output of the system will be satisfying the requirements of the plants by watering.
- The automated system would have to include different sensors which senses humidity, temperature and soil moisture, and then upload the sensed information to the databases on Cloud.
- Soil Moisture Sensor can be used to detect the moisture of soil or judge if there is water around the sensor.
- Users should be able access the system remotely.

### (iii) Methodology Adopted

In the proposed structure of this system, various modules were introduced such as Arduino as controller, temperature sensor, moisture sensor, humidity sensor and PH sensor. The water motor is also set in this system and according to sensor values analyzed by different types of sensors the water Motor will get ON automatically and the sensor values are given to ADC (Analog and Digital Converter) to be processed by the controller.

### (iv) Performance Limitations

To make it a real-life product it requires a high-speed processor and advanced sensor compatible with the size of lands.

3.

**(i) Title, Author, Date of publication and Publisher Information**

Automatic IoT Based Plant Monitoring and Watering System using Raspberry Pi
*Anusha k a , Dr. U B Mahadevaswamy*

I.J. Engineering and Manufacturing,volume 6,pp.55-67,2018.

**(ii) Problem and Objectives**

Problem:

In India though the main source for citizens is agriculture, an old traditional method is used which makes them earn less money, not providing the revenue to solve family financial problems. Some of the instruments they can’t fix due to the huge costs incurred, so in order to overcome these problems there is a need for a more economical solution

Objectives:

- The primary objective of this proposed work is to develop an Embedded System for plant monitoring and watering system using Internet of Things, Raspberry Pi as Processor, and sensors for sensing environmental conditions.
- To develop web based Application for accessing result.
- To develop system for Real Time Application.

**(iii) Methodology Adopted**

The proposed system includes both hardware and software. The hardware requirements are such as follows: Raspberry pi(Model B 3.1), Temperature sensor(LM35) ,Humidity sensor ,Moisture sensor, Light sensor, IR sensor , Relay and motor Server with database and IoT enabled web based application Power supply. The Raspberry Pi operates on Linux- noobs operating system. The operating system is dumped in an SD card inserted in the processor. The python language is used for coding. After all hardware connection done power supply switch on, in the monitor terminal screen login has to take place to run the system.

**(iv) Performance Limitations**

Does not have the necessary modules that can effectively make use of the Rain Water harvesting techniques which is used excessively to irrigate fields

4.

**(i) Title, Author, Date of publication and Publisher Information**

IoT Based Garden Monitoring System
*Sambath.M, Prasant.M2, Bhargav Raghava.N, Jagadeesh*
2019 J. Phys.: Conf. Ser. 1362 012069 June 2019, Eluru, India

**(ii) Problem and Objectives**

Problem:

Due to the hectic work schedules of people there is less care taken towards plants and there is often a lot of time wasted watering the plants so an efficient management of water should be developed.

Objectives:

- To enable smart monitoring of the plant growth by modernizing the current traditional methods of gardening
- The proposed system should target a smart way of monitoring plant growth using automation and IoT technologies.
- Should increase the plant growth condition by maintaining the suitable moisture level and temperature with the use of moisture sensor and temperature sensor.
- Crop development at low quantity water consumption by providing an automatic watering system to the user.

**(iii) Methodology Adopted**

In the proposed system Arduino Uno is considered to be the heart of the entire system. The Arduino Uno consists of several input and output ports through which several components are connected together and then functionalized. It also consists of an inbuilt storage device which is of 256KB where the executable code is uploaded.

Arduino function will be based upon the code which is uploaded into the memory. This code is done with the help of Arduino 1.6.13, it is coded in such a way that all the sensors have to be defined. Soil moisture sensor, temperature sensor, LDR are connected to the Arduino and its respective information is gathered. The soil moisture sensor will detect the moisture content, if the moisture content is less than the required moisture level through the relay the signal will be sent and the water pumped will be on, after a few seconds the water pump will be turned off. This module will help to increase the efficient usage of water. Temperature sensor is used to detect the temperature. Water level sensor is used to get the water level. LDR sensor will be used to detect the light presence and LED light is used to increase the light presence if needed. During the night times LDR will automatically help to switch on the light which helps to observe our plant even at night times. With a proper GUI an android app is developed so all the plant growth details obtained from the plant will be made visible to the end user.

### (iv) Performance Limitations

Alerts provided to the farmers are limited to the kind of data available in the pre-defined data about the suitable conditions for particular plants. If the plant faces a kind of abnormality that is not present in the predefined data, the farmers are not alerted. This system also does not have a camera to provide a visual to the farmer. This restricts them from keeping check on the plant's growth. The farmers are also not alerted in case of an intruder like an example as it won't be detected by the system.

5.

### (i) Title, Author, Date of publication and Publisher Information

IOT Based Smart Plant Monitoring System
*Prathamesh Pawar, Aniket Gawade , Sagar Soni , Santosh Sutar , Harshada Sonkamble*
International Journal for Research in Applied Science & Engineering Technology (IJRASET) ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538 Volume 10 Issue V May 2022 IOT Based Smart Plant Monitoring System

## (ii) Problem and Objectives

Problem:

Plant plays a vital role in maintaining the ecological cycle and forms the foundation of a food chain pyramid and thus to maintain the plant's proper growth and health adequate monitoring is required.

Objectives:

- The primary objective of this proposed work is to develop an Embedded System for plant monitoring and watering system using Internet of Things, Raspberry Pi as Processor, and sensors for sensing environmental conditions.
- To develop web based applications for accessing results.
- To develop a system for Real Time Application.

## (iii) Methodology Adopted

The proposed Plant Monitoring System uses NodeMCU as a microcontroller. NodeMCU comes with the inbuilt ESP8266 WiFimodule which connects the system to the blynk app using WiFi. The program which controls the functioning of the whole system is fed into the microcontroller using Arduino IDE which is an environment which integrates code with the hardware. Soil moisture sensor continuously detects the level of moisture in the soil and displays it on the Virtual LCD widget on the Blynk app. If the water content in the soil is less than what is required by the plant, a notification is sent to the user"s smartphone and he/she can switch ON the button widget in Blynk App which will turn ON the water supply. Real time values from the DHT11 temperature sensor are also displayed on the virtual LCD. Excessive heat from the sun can be harmful for plants to prevent them from dying, the authors introduced a green shade which will automatically be drawn over the plant with the help of two DC motors which rotate clockwise and anti-clockwise to help movement of the shade. Temperature more than 30 $^0$C can cause shriveling of plants. When temperature increases this limit the motor rotates and causes the shade to move automatically. The user is notified about each and every step through the notification feature of the Blynkapp. Hence, this system monitors and controls the plant"s requirements

### (iv) Performance Limitations

Security not being present in the Device and Owners Account, absence of solar cells to charge the battery, and the lack of a compact design that would enable the plant to be fitted anywhere. Automated irrigation systems use only two parameters of soil like soil moisture and temperature, other parameters humidity, light, air moisture, soil pH value not taken for decision making. Excessive seepage and leakage of water forms marshes and ponds all along the channels. The marshes and the ponds in course of time become the colonies of the mosquito, which gives rise to a disease like malaria.

## 6.

### (i) Title, Author, Date of publication and Publisher Information

IoT Based Plant Monitoring System

*Prof. Likhesh Kolhe, 2 Prof. Prachi Kamble, 3Mr.Sudhanshu Bhagat, 4Mr.Sohail Shaikh, 5Mr. Ronak Sahu, 6Miss.Swati Chavan, 7Miss. Prajakta Zodge*

International Journal of Interdisciplinary Innovative Research & Development (IJIIRD) ISSN: 2456-236X Vol. 02 Special Issue 05 | 2018

### (ii) Problem and Objectives

- To facilitate the irrigation of 2/3rd part of land that is still dependent on monsoon for water
- Avoid the damage caused by overwatering of crops by uninformed farmers
- To place humidity sensor, moisture sensor, temperature sensors placed in root zone of plant

### (iii) Methodology adopted

In this research the methodology adopted allows the farmer to monitor and control the system in order to improve the efficiency with help of sensor parameters like temperature, humidity, soil moisture.

The system is a combination of hardware and software components. Hardware components include Sensors (Moisture, DHT22, Ultrasonic), ESP8266 Wi-Fi module, Arduino Uno, Water pump, and the software components include an Android application.

When the power supply is ON, the input module of three sensors (DHT22, moisture) starts to activate. When sensors get ON it will read the data from soil and from surrounding. According to the values that are detected by sensors, the motor will turn ON/OFF. If Moisture below threshold value, then the motor is turned ON. If the moisture level is high, then it will stop the motor and water supply will also stop. If Water level is low in the tank, then it will also be detected by the ultrasonic sensor. All the values that are collected from the sensor are sent via ESP8266 Wi-Fi module to Thing speak cloud server and stored in an online database(firebase) via dummy server. Thing speak will create the graph for the data received by the WI-FI module. And, then whole information will show on the Android app. Users can easily control the motor manually by using an Android app.

**(iv) Performance Limitation**

This irrigation system uses only three parameters of soil like soil moisture, temperature and humidity, other parameters like light, air moisture, soil pH value are not taken for decision making. The cost of setting up the system on a large scale would also be impractical.The quality of water is not checked in this system prior to irrigation.

**7.**

**(i) Title, Author, Date of publication and Publisher Information**

An IoT-based Smart Garden with Weather Station System

*Norakmar binti Arbain Sulaiman, Muhamad Dan Darrawi bin Sadli*

Published 27 April 2019 - Computer Science - 2019 IEEE 9th Symposium on Computer Applications & Industrial Electronics (ISCAIE)

**(ii) Problem and Objectives**

- To eradicate the frequent failure of sufficient irrigation of crops in home gardening.

- To monitor a vast area of parameters ranging from weather conditions to moisture, humidity, temperature, light intensity etc.
- To facilitate the faster growth of plants.

**(iii) Methodology adopted**

The device is equipped with a water pump, where it can be monitored and controlled by using a smartphone. In addition, the devices also consist of four main sensors, which are Barometric Pressure, DHT11 Temperature, and Humidity Sensor, Soil Moisture Sensor and Light intensity module sensor. The Soil and Light Intensity sensor used to measure the value by percentages. Besides, two actuators, which are the water pump and LED light can be used remotely or by using a button on the devices. The LED is purposely to replicate the sunlight and make the plant grow faster. This IoT-based Smart Garden with Weather Station System can record the data and send the result to the user through the smartphone application named as Blynk apps.

**(iv) Performance Limitation**

The actuators are exposed to harm in the outdoors and the expenses of repair could be a lot. The readings of the barometric pressure could be inaccurate.

**8.**

**(i) Title, Author, Date of publication and Publisher Information**

IoT Based Smart Agriculture Monitoring, Automation and Intrusion Detection System

*Ahmad Faisol Suhaimi, Naimah Yaakob, Sawsan Ali Saad, Khairul Azami Sidek, Mohamed Elobaid Elshaikh, Alaa K. Y. Dafhalla, Ong Bi Lynn1, Mahathir Almashor*

The 1st International Conference on Engineering and Technology (ICoEngTech) 2021 Journal of Physics: Conference Series 1962 (2021) 012016 IOP Publishing

**(ii) Problem and Objectives**

- To reduce the water wastage caused by low irrigation efficiency.
- To minimize the manual monitoring that leads to human error

- To provide efficient water consumption based on specific conditions using sensors to trigger watering of the crop field.
- To develop a mobile app to monitor, display the real-time data from the environment and notify users upon detection of intruders based on various sensors.

**(iii) Methodology adopted**

ESP32 NodeMCU is used as the main hardware. The microcontroller serves as the brain of the system where the temperature and humidity, infrared and soil moisture sensor provide the digital information to and where it will process the information. The input from these sensors will be used to either turn on or off the motor pump based on the prescribed instructions. The soil moisture sensor is put inside the soil which is located near the crop to detect the moisture of the soil. The temperature and humidity sensor are positioned behind the crop to measure the temperature and humidity of surrounding air. All data that have been collected from all sensors is displayed in the LCD located near the crop field for monitoring purposes. Collected information is uploaded to the firebase cloud using a microcontroller where all the data is instructed by the server to be stored in the database. Moreover, the developed mobile app displays the data retrieved from the database. Users can have access and control to their automatic irrigation system and monitor the crop through their smartphones. Using this approach, users can monitor the crop efficiently and track

the real time data condition of the soil, temperature and humidity of air surrounding the crop.

Apart from that, this system can also serve as an intrusion detection system which prevents visits from unwanted animals that can damage the crops or the presence of any unauthorized human being. This can be done with the help of the passive infrared (PIR) sensor. The PIR sensor is attached in front of the crop field to detect any kind of intrusion. If the PIR sensor detects the intrusion, the alarm will be activated automatically The system is designed to send the sensed signals to the database every time the system is in the automatic or manual mode. These signals can be tracked from a personal computer which serves as the database.

**(iv) Performance Limitation**

Camera with machine learning has not been used which could improve detection accuracy of intruders without the help of farmers. Moreover, the passive infrared sensor used can only observe between the living object and non- living object. It cannot differentiate between animal or human during detection. It can also not track whether the animal poses a threat to the plant or not.

**9.**

**(i) Title, Author, Date of publication and Publisher Information**

IoT-based Irrigation Management for Smallholder Farmers in Rural Sub-Saharan Africa

*Ethiopia Nigussiea,, Thomas Olwalb, George Musumbac, Tesfa Tegegned*

*, Atli Lemmae, Fisseha Mekuria*

Published Elsevier - The 11th International Conference on Emerging Ubiquitous Systems and Pervasive Networks (EUSPN 2020) November 2-5, 2020, Madeira, Portugal

**(ii) Problem and Objectives**

- To capture the dynamic changes of the influencing parameters usually failed due to infrequent laboratory tests and pre-defined empirical models in Sub-Saharan Africa

**(iii) Methodology adopted**

To compensate for the lack of weather station networks, deployment of microclimate sensors around the farmland is considered in the proposed architecture of the irrigation system. In addition, the crops specific water requirement (amount and watering frequency) at different stages of their growth has to be taken into account by the irrigation management system. The inclusion of regional weather information as input to the data processing component besides the data from the microclimate sensors enable reliable near-future forecasts of climate parameters that help to estimate the water needs of the crops with better accuracy. The irrigation management system performs intelligent data analysis and decides on the most appropriate actuation based on the analysis results. To accomplish these tasks the system needs various hardware components, such as sensors, actuators, low-power IoT nodes, low-power wireless communication modules, and computing devices.  Unlike the common IoT architecture, the collected data processing and decision-making are done locally instead of in a remote cloud. This approach addresses the unavailability of broadband connectivity or the associated costs of remote computing infrastructure and connectivity. The supplementary inputs (regional weather information, information on crop and soil characteristics from agricultural experts, and farmers' indigenous knowledge) are key for improving the accuracy of the actuation decisions. The inclusion of expert knowledge on the specific crop and soil characteristics in the data processing is important to optimize the irrigation schedule without impacting the crop yield and/or depletion of soil nutrients.

**(iv) Performance Limitation**

Cannot be effectively put to use until sufficient data has been fed into the machine using intelligent data analysis which would allow the system to provide crop specific conditions for the proper growth of plants.

**10.**

**(i) Title, Author, Date of publication and Publisher Information**

IoT based smart irrigation monitoring and controlling system

*Shweta B. Saraf, Dhanashri H. Gawali*

Published 19 May 2017 Computer Science 2017 2nd IEEE International Conference on Recent Trends in Electronics, Information & Communication Technology (RTEICT)

**(ii) Problem and Objectives**

- To utilize cloud computing to monitor and control  a set of sensors and actuators to assess the plants water needs

### (iii) Methodology adopted

The device uses real time input data. Smart farm irrigation system uses android phone for remote monitoring and controlling of drips through wireless sensor network. Zigbee is used for communication between sensor nodes and base stations. Real time sensed data handling and demonstration on the server is accomplished using web based java graphical user interface. Wireless monitoring of the field irrigation system reduces human intervention and allows remote monitoring and controlling on android phones. Cloud Computing is an attractive solution to the large amount of data generated by the wireless sensor network. A cloud-based wireless communication system to monitor and control a set of sensors and actuators to assess the plants water needs was used.

### (iv) Performance Limitation

Though the pesticides and fertilizers have their individual uses in the agricultural field, if these are put in excess, they can cause harm to the soil, crops, and environment. Thus, controlled amounts of these could be deployed in a proper manner. Further work could also be done in the field of nutrient management for the soil. Parameters apart from moisture and temperature can be taken into consideration in order to increase the efficiency of the system. The farmers can be provided with information about the ideal crop combinations for the multiple cropping scenarios in order to ensure low costs and higher production.

## OUR PROPOSED MODEL:

Our proposed model for the IoT project is centred around detecting soil moisture using a moisture sensor and automating the pumping of water into the soil to maintain moisture levels. The project will utilize the Arduino IoT Cloud to connect the moisture sensor to the internet and enable remote monitoring and control of the system. The moisture sensor will be placed in the soil and will send data to the Arduino IoT Cloud platform, which will analyse the data and trigger the pumping of water when the moisture level drops below a certain threshold. The system will also have a feedback mechanism to ensure that the watering process is not excessive or insufficient, providing the optimal amount of water to maintain the soil's moisture level. There will also be the use of various other modules such as a temperature and humidity sensor to ensure that the air conditions are also recorded. This proposed model has the potential to significantly reduce the water consumption in agriculture and enable more efficient and sustainable use of resources.

# IMPLEMENTATION

## 3.1 Design and Implementation

We use NodeMCU (ESP 8266) microcontroller to integrate the entire module where sensors are connected to it, and data is transferred via a Wi-Fi module that is available in the microcontroller. The device is portable and requires charging after some time. The data is transferred directly to the application. Proposed system diagram is a pictorial representation which shows how every component plays a certain role in it. In this diagram, the device is depicted in the centre with various components. The network part is an access point allotted for operating the device. Temperature sensors are used to detect the current status of heat and soil of air that can be useful for analysing the heat effect on garden plants. Similarly, the YL-69 sensor displays the current moisture levels of soil and the humidity sensor detects the percentage of water vapours at the moment. All sensors provide the real-time data of plants, which help us to get an exact situation of the environment.

Users can use this information to treat areas which are affected in the garden. The Charging Unit consists of a battery that is connected to a power input chip and a circuit board that uses it to transfer the DC power to sensors for functioning properly. The Display unit can generate output, i.e., the parameters detected currently in the area along with the IP address. Requirements of this system are categorised as functional and non-functional requirements

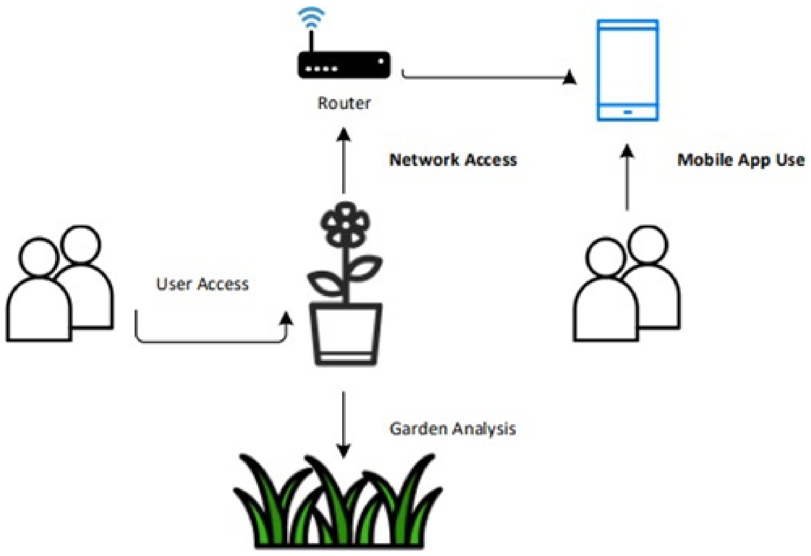


**Fig 1. Pictorial representation of the components of proposed system**

## 3.2 Block Diagram

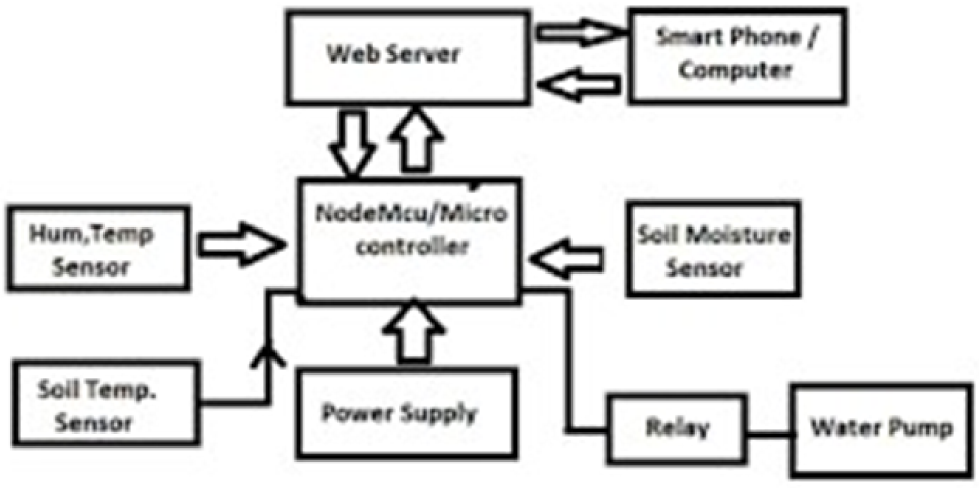


**Fig 2. Block Diagram of proposed system**

## 3.3 Hardware Analysis

### 3.3.1. NODE MCU

NodeMCU is a digital microcontroller based upon system on chip (SoC) technology to develop IoT applications (NodeMcu, 2019). It contains an onboard WIFI system for communication of data and other supporting libraries. MCU refers to the Microcontroller Unit. It provides the facility of analysing, controlling, and monitoring of digital systems. Here are a few prominent features of NodeMCU

**Fig 3. NodeMCU 0.9 ESP - 12 module**

### 3.3.2 2.3.2 YL-69 soil moisture sensor

YL-69 is the cheapest and reliable soil moisture sensor to sense the humidity in the soil. It consists of two components, a small electric board, and two sensing strips to sense the values. Figure shows the YL-69 soil moisture sensor

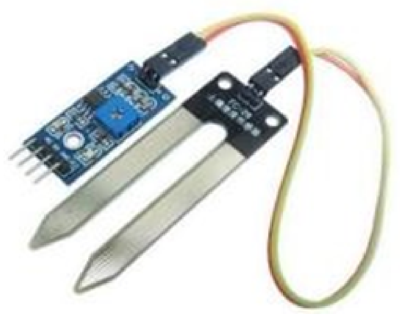


**Fig 4. Soil moisture sensor YL-69**

### 3.3.3 Temperature And Humidity Sensor

The RHT03 (also known by DHT-22) is a low cost humidity and temperature sensor with a single wire digital interface. The sensor is calibrated and doesn’t require extra components so you can get right to measuring relative humidity and temperature.

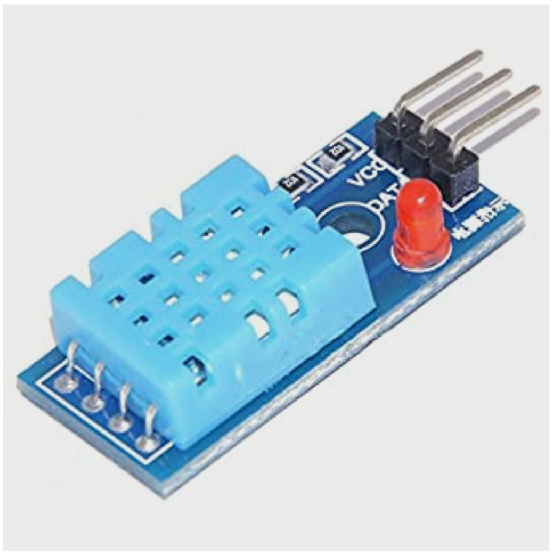


**Fig 5. DHT11 Temperature and Humidity Sensor**

### 3.3.4 Relay Module

DC signals which come from the battery will be converted into AC by the use of the Relay which is shown in Fig. As these motors work on AC current a controlled system like relay is used. Relays are used in electronic circuits such as high-power amplifiers, telephone exchanges etc. In this project relays are used to switch on the LED and water pump whenever there is less light presence and soil moisture near the plant.

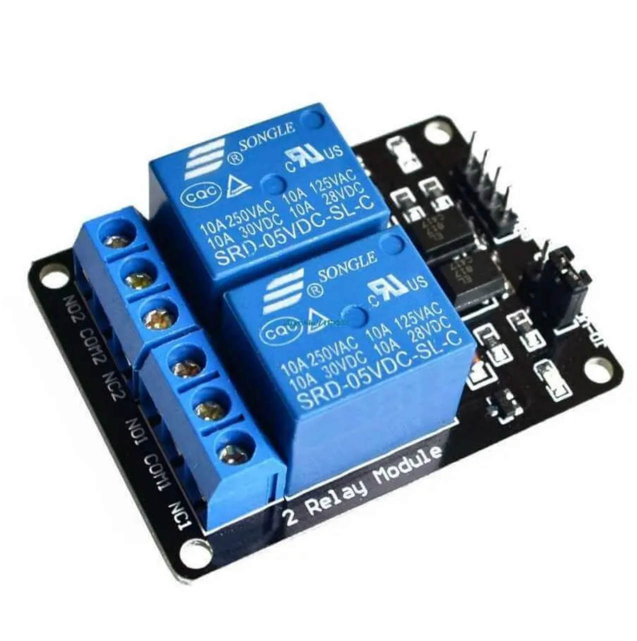


**Fig 6. Double channel Relay module**

#### 3.3.5 Water Pump

This DC 3-6 V Mini Micro Submersible Water Pump is a low-cost, small- size Submersible Pump Motor that can be operated from a 2.5 ~ 6V power supply. It can take up to 120 liters per hour with a very low current consumption of 220mA. Just connect the tube pipe to the motor outlet, submerge it in water, and power it.

**Fig 7. 3V mini submersible water pump**

### 3.4 Snapshot of project and output hardware

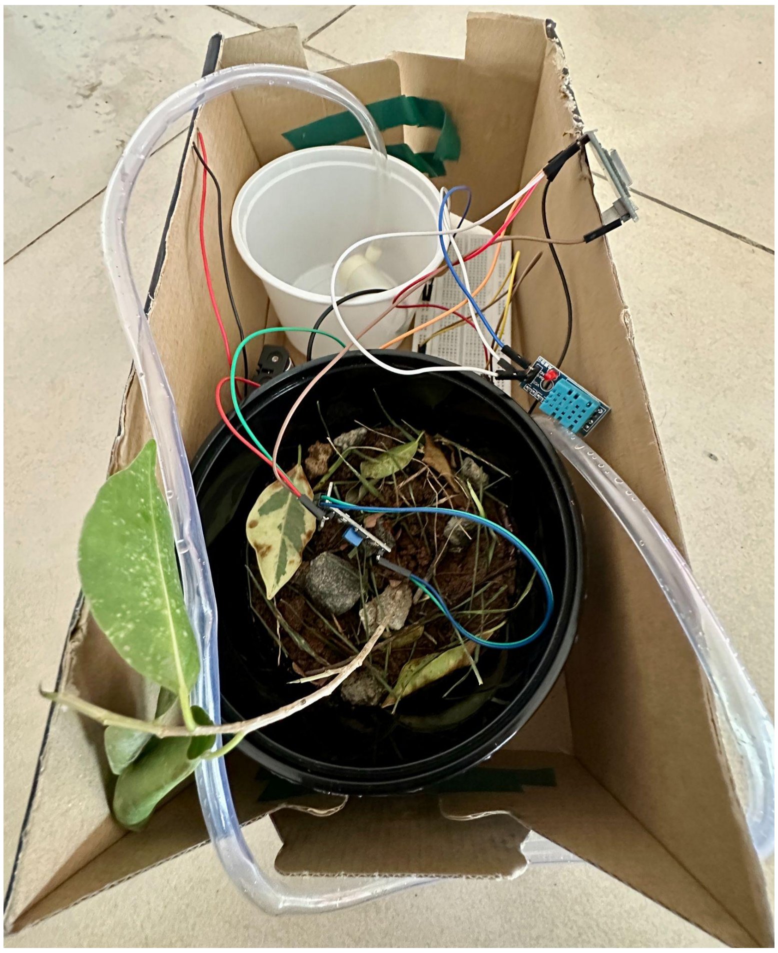


**Fig 8. Top View of our project**

## Dashboard

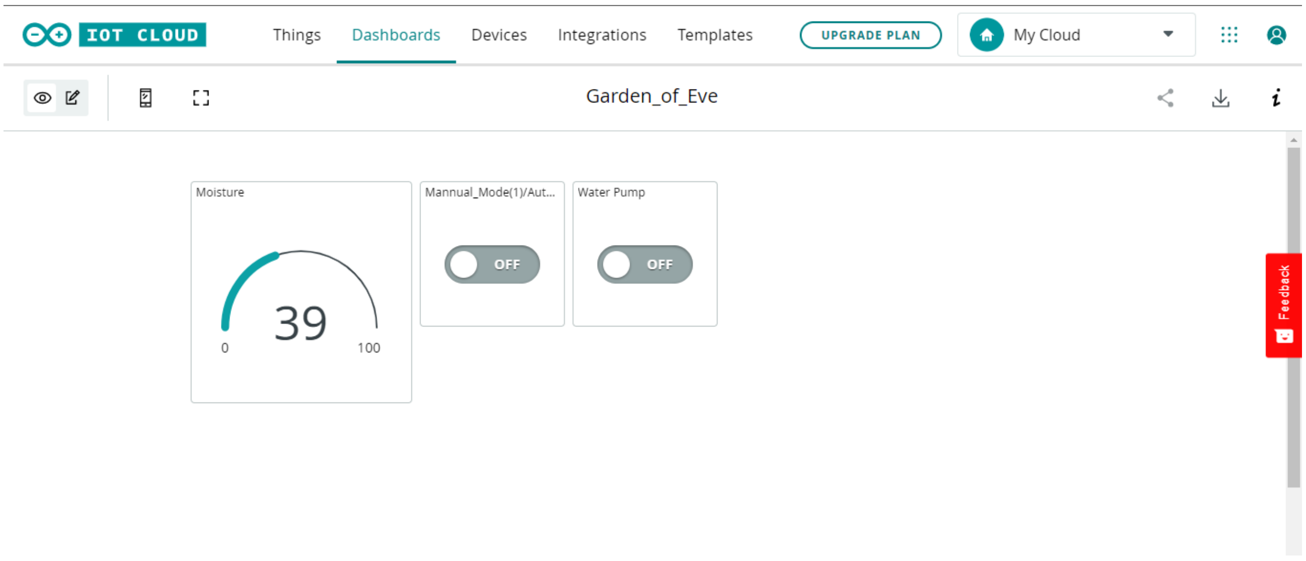


Fig 9. Our Dashboard

# SOFTWARE

## 4.1 Coding Analysis

### ESP 8266 code to control Sensors and Motors

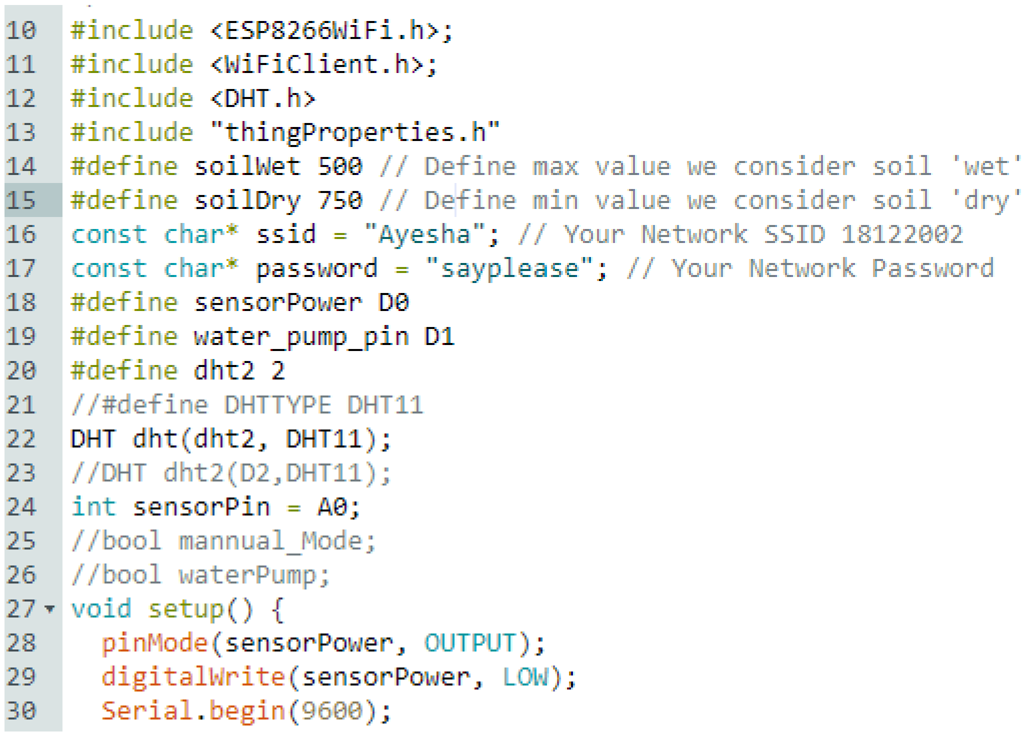
```
#include <ESP8266WiFi.h>;
#include <WiFiClient.h>;
#include <DHT.h>
#include "thingProperties.h"
#define soilWet 500 // Define max value we consider soil 'wet'
#define soilDry 750 // Define min value we consider soil 'dry'
const char* ssid = "Ayesha"; // Your Network SSID 18122002
const char* password = "sayplease"; // Your Network Password
#define sensorPower D0
#define water_pump_pin D1
#define dht2 2
//#define DHTTYPE DHT11
DHT dht(dht2, DHT11);
//DHT dht2(D2,DHT11);
```

```
int sensorPin = A0;
//bool mannual_Mode;
//bool waterPump;
void setup() {
  pinMode(sensorPower, OUTPUT);
  digitalWrite(sensorPower, LOW);
  Serial.begin(9600);
  delay(10);
  dht.begin();
  pinMode(D1, OUTPUT);//Water Pump
  // Defined in thingProperties.h
  initProperties();
  pinMode(water_pump_pin, OUTPUT);
  // Connect to Arduino IoT Cloud
  //Connect to WiFi network
  WiFi.begin(ssid, password);
  ArduinoCloud.begin(ArduinoIoTPreferredConnection);

  /*
    The following function allows you to obtain more information
     related to the state of network and IoT Cloud connection and
errors
    the higher number the more granular information you'll get.
    The default is 0 (only errors).
    Maximum is 4
  */
  setDebugMessageLevel(2);
  ArduinoCloud.printDebugInfo();
}

void loop() {
  ArduinoCloud.update();
  // Your code here
  onMannualModeChange();
  delay(100);
  //int readData =dht.read(dht2); // Reads the data from the sensor
   temperature =dht.readTemperature(); // Gets the values of the
```

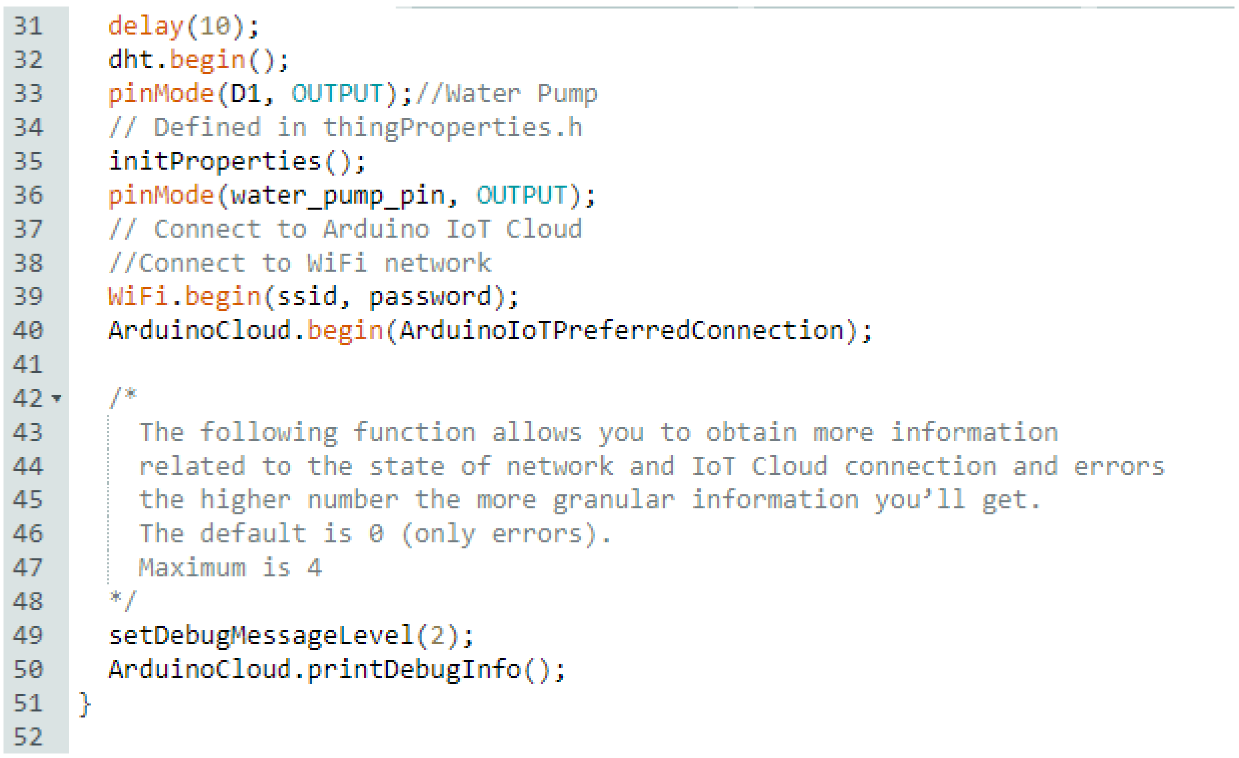
```
temperature
  humidity =dht.readHumidity(); // Gets the values of the humidity
}
/*
  Since MannualMode is READ_WRITE variable,
  onMannualModeChange() is
  executed every time a new value is received from IoT Cloud.
*/
void onMannualModeChange() {
  if (mannual_Mode) {
    onWaterPumpChange();
  }
  else {
    automate();
  }
}
/*
  Since WaterPump is READ_WRITE variable, onWaterPumpChange() is
  executed every time a new value is received from IoT Cloud.
*/
void onWaterPumpChange() {
  if (water_Pump) {
    digitalWrite(water_pump_pin, HIGH);
  }
  else {
    digitalWrite(water_pump_pin, LOW);
  }
}
void automate()
{
  int moisture = readSensor();
  Serial.print("Analog Output: ");
  Serial.println(moisture);
  moisture_Value = 100 * (1024 - moisture) / 1024;
  // Determine status of our soil

  if (moisture < soilWet) {
    Serial.println("Status: Soil is too wet-Stop the Watering");
    digitalWrite(D1, LOW);
  } else if (moisture >= soilWet && moisture < soilDry) {
    Serial.println("Status: Soil moisture is perfect");
    digitalWrite(D1, LOW);
  } else {
    Serial.println("Status: Soil is too dry - time to water!");
    digitalWrite(D1, HIGH);
  }
  Serial.println();
}
int readSensor() {
  digitalWrite(sensorPower, HIGH); // Turn the sensor ON
  delay(10); // Allow power to settle
   int  val  =  analogRead(sensorPin);  //  Read  the  analog  value  form
sensor
  digitalWrite(sensorPower, LOW); // Turn the sensor OFF
  return val; // Return analog moisture value
}

/*
  Since Message is READ_WRITE variable, onMessageChange() is
  executed every time a new value is received from IoT Cloud.
*/
void onMessageChange()  {
  // Add your code here to act upon Message change
}
/*
  Since Humidity is READ_WRITE variable, onHumidityChange() is
  executed every time a new value is received from IoT Cloud.
*/
void onHumidityChange()  {
  // Add your code here to act upon Humidity change
}
/*
          Since     MoistureValue     is     READ_WRITE     variable,
onMoistureValueChange() is
```

```
  executed every time a new value is received from IoT Cloud.
*/
void onMoistureValueChange()  {
  // Add your code here to act upon MoistureValue change
}
/*
  Since Val is READ_WRITE variable, onValChange() is
  executed every time a new value is received from IoT Cloud.
*/
void onValChange()  {
  // Add your code here to act upon Val change
}
/*
  Since Temperature is READ_WRITE variable, onTemperatureChange() is
  executed every time a new value is received from IoT Cloud.
*/
void onTemperatureChange()  {
  // Add your code here to act upon Temperature change
}
```

## 4.2 Snapshots of Code

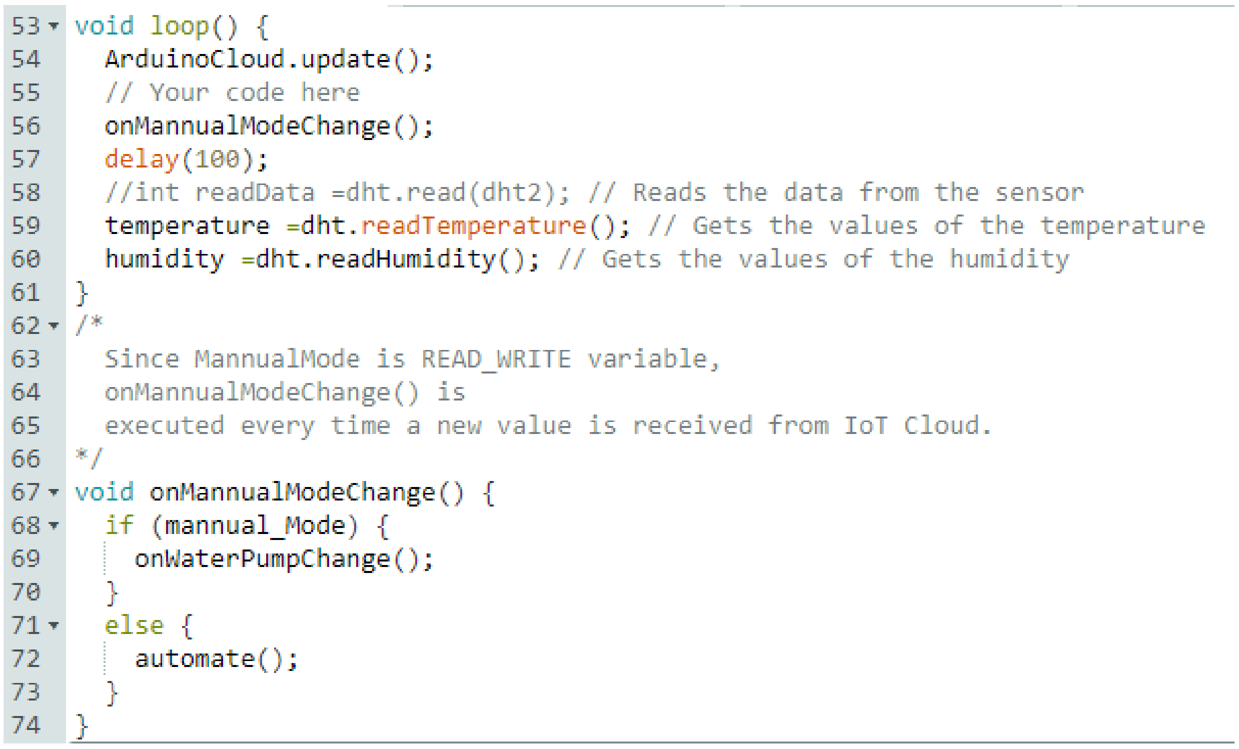


```
#include <ESP8266WiFi.h>;
#include <WiFiClient.h>;
#include <DHT.h>
#include "thingProperties.h"
#define soilWet 500 // Define max value we consider soil 'wet'
#define soilDry 750 // Define min value we consider soil 'dry'
const char* ssid = "Ayesha"; // Your Network SSID 18122002
const char* password = "sayplease"; // Your Network Password
#define sensorPower D0
#define water_pump_pin D1
#define dht2 2
//#define DHTTYPE DHT11
DHT dht(dht2, DHT11);
//DHT dht2(D2,DHT11);
int sensorPin = A0;
//bool mannual_Mode;
//bool waterPump;
void setup() {
  pinMode(sensorPower, OUTPUT);
  digitalWrite(sensorPower, LOW);
  Serial.begin(9600);
```

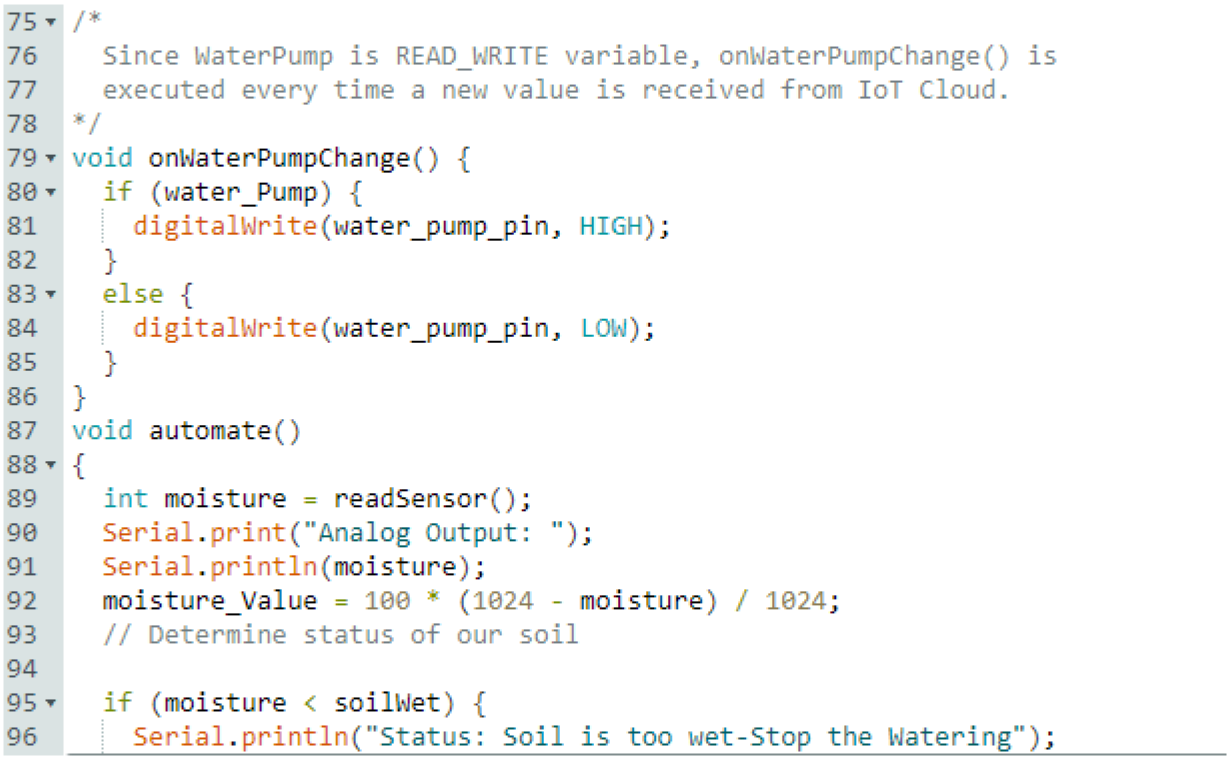


```
  delay(10);
  dht.begin();
  pinMode(D1, OUTPUT);//Water Pump
  // Defined in thingProperties.h
  initProperties();
  pinMode(water_pump_pin, OUTPUT);
  // Connect to Arduino IoT Cloud
  //Connect to WiFi network
  WiFi.begin(ssid, password);
  ArduinoCloud.begin(ArduinoIoTPreferredConnection);

  /*
    The following function allows you to obtain more information
    related to the state of network and IoT Cloud connection and errors
    the higher number the more granular information you'll get.
    The default is 0 (only errors).
    Maximum is 4
  */
  setDebugMessageLevel(2);
  ArduinoCloud.printDebugInfo();
}

```

**Fig 10 .Code Snapshot 1**

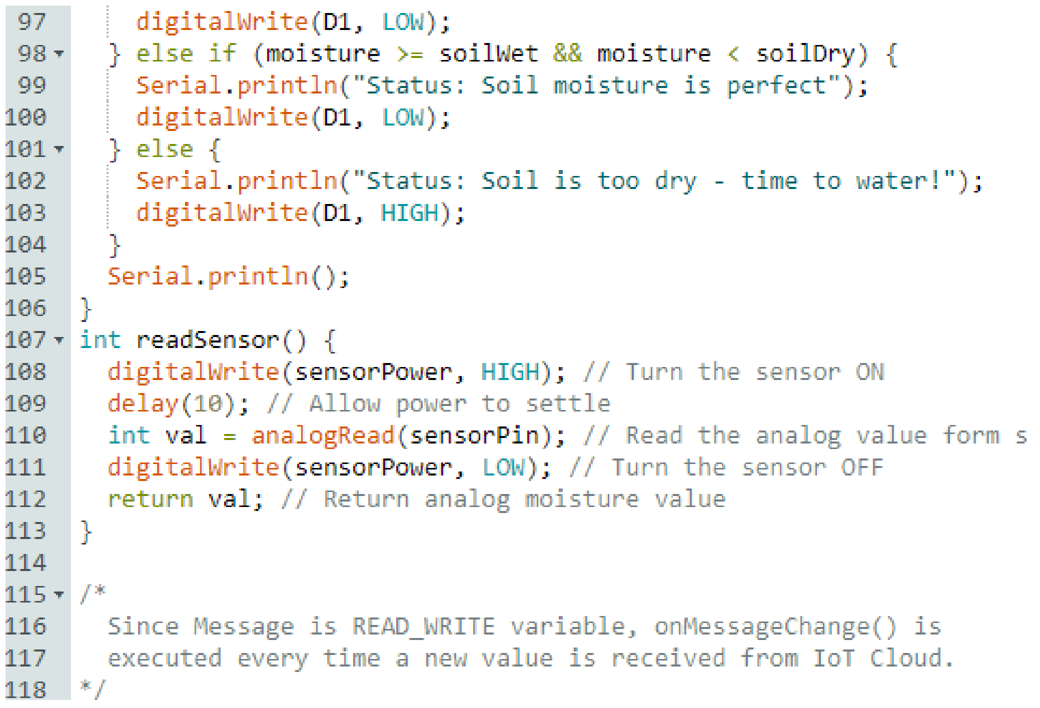


```
void loop() {
  ArduinoCloud.update();
  // Your code here
  onMannualModeChange();
  delay(100);
  //int readData =dht.read(dht2); // Reads the data from the sensor
  temperature =dht.readTemperature(); // Gets the values of the temperature
  humidity =dht.readHumidity(); // Gets the values of the humidity
}
/*
  Since MannualMode is READ_WRITE variable,
  onMannualModeChange() is
  executed every time a new value is received from IoT Cloud.
*/
void onMannualModeChange() {
  if (mannual_Mode) {
    onWaterPumpChange();
  }
  else {
    automate();
  }
}
```

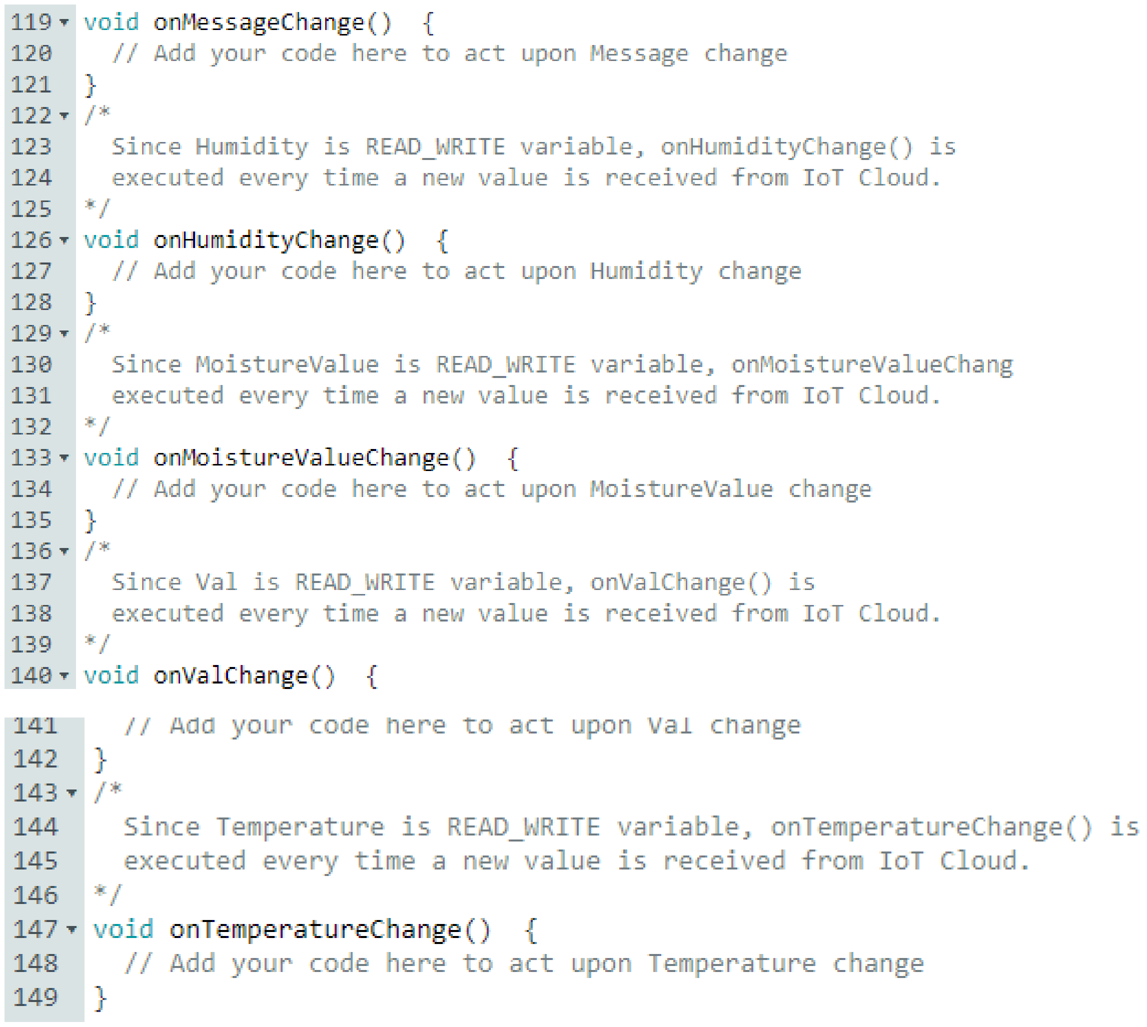


```
/*
  Since WaterPump is READ_WRITE variable, onWaterPumpChange() is
  executed every time a new value is received from IoT Cloud.
*/
void onWaterPumpChange() {
  if (water_Pump) {
    digitalWrite(water_pump_pin, HIGH);
  }
  else {
    digitalWrite(water_pump_pin, LOW);
  }
}
void automate()
{
  int moisture = readSensor();
  Serial.print("Analog Output: ");
  Serial.println(moisture);
  moisture_Value = 100 * (1024 - moisture) / 1024;
  // Determine status of our soil

  if (moisture < soilWet) {
    Serial.println("Status: Soil is too wet-Stop the Watering");
```

**Fig 11. Code Snapshot 2**



```
    digitalWrite(D1, LOW);
  } else if (moisture >= soilWet && moisture < soilDry) {
    Serial.println("Status: Soil moisture is perfect");
    digitalWrite(D1, LOW);
  } else {
    Serial.println("Status: Soil is too dry - time to water!");
    digitalWrite(D1, HIGH);
  }
  Serial.println();
}
int readSensor() {
  digitalWrite(sensorPower, HIGH); // Turn the sensor ON
  delay(10); // Allow power to settle
  int val = analogRead(sensorPin); // Read the analog value form s
  digitalWrite(sensorPower, LOW); // Turn the sensor OFF
  return val; // Return analog moisture value
}

/*
  Since Message is READ_WRITE variable, onMessageChange() is
  executed every time a new value is received from IoT Cloud.
*/
```



```
void onMessageChange()  {
  // Add your code here to act upon Message change
}
/*
  Since Humidity is READ_WRITE variable, onHumidityChange() is
  executed every time a new value is received from IoT Cloud.
*/
void onHumidityChange()  {
  // Add your code here to act upon Humidity change
}
/*
  Since MoistureValue is READ_WRITE variable, onMoistureValueChang
  executed every time a new value is received from IoT Cloud.
*/
void onMoistureValueChange()  {
  // Add your code here to act upon MoistureValue change
}
/*
  Since Val is READ_WRITE variable, onValChange() is
  executed every time a new value is received from IoT Cloud.
*/
void onValChange()  {
  // Add your code here to act upon Val change
}
/*
  Since Temperature is READ_WRITE variable, onTemperatureChange() is
  executed every time a new value is received from IoT Cloud.
*/
void onTemperatureChange()  {
  // Add your code here to act upon Temperature change
}
```

**Fig 12. Code Snapshot 3**

## SUGGESTIONS AND PARTIAL IMPLEMENTATIONS OF A CAMERA MODULE

Adding a camera module to the smart irrigation system can provide additional data about the plants and soil conditions, which can be useful for analysis and decision making. Unfortunately, its implementation could not fit into the scope of the current project, but it shows potential for future integration with better budgets to incur the costs of such an expensive sensor.
The sensor in particular is the ESP32 camera module which can be discussed in detail to provide the necessary insight into its application.

## INTRODUCTION TO ESP32 CAMERA MODULE

The ESP32 camera module is a small-sized camera module designed to work with the ESP32 microcontroller. It features an OV2640 camera sensor and is capable of capturing images with a maximum resolution of 1600 x 1200 pixels. The camera module is designed to be connected to the ESP32 microcontroller using the standard SPI interface and is also equipped with an onboard SD card slot for storing images.

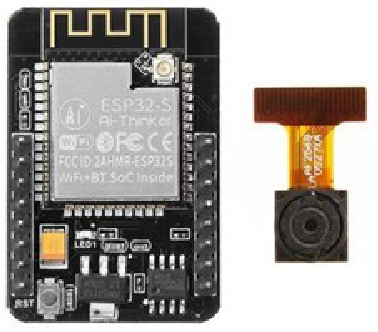


**Fig 13. Proposed Camera module**

## FEATURES:

- The camera module can be easily integrated with the ESP32 microcontroller using the ESP-IDF software development framework.
- It provides a simple and efficient API for capturing and processing images from the camera module.
- The Camera module is compatible with popular open-source libraries such as ESP32-Camera and Arduino-Camera, which make it easy to get started with capturing and processing images
- Low power consumption. It is designed to operate on low power, making it suitable for use in battery-powered devices such as wireless cameras or smart home devices.
- The module is very compact and can be easily integrated into small form-factor devices.

Overall, the ESP32 camera module is a versatile and reliable camera module that is well-suited for use in a variety of IoT applications, including smart irrigation systems. Its compatibility with the ESP32 microcontroller, low power consumption, and compact size make it an attractive option for developers looking to integrate a camera module into their IoT project.

## ESP 32 CAMERA CODE:

```
#include "WiFi.h"
#include "esp_camera.h"
#include "esp_timer.h"
#include "img_converters.h" #include "Arduino.h"
#include "soc/soc.h"
#include "soc/rtc_cntl_reg.h" #include "driver/rtc_io.h"
#include <ESPAsyncWebServer.h> #include <StringArray.h>
#include <SPIFFS.h>
#include <FS.h>
const char* ssid = "Ayesha";
const char* password = "sayplease";
AsyncWebServer server(80);
         boolean new_photo = false; #define photo_path "/image.jpg"
// OV2640 camera module pins (CAMERA_MODEL_AI_THINKER) #define
PWDN_GPIO_NUM 32
#define RESET_GPIO_NUM -1
```

```
#define XCLK_GPIO_NUM #define SIOD_GPIO_NUM #define SIOC_GPIO_NUM
#define Y9_GPIO_NUM 35 #define Y8_GPIO_NUM 34 #define Y7_GPIO_NUM 39
#define Y6_GPIO_NUM 36 #define Y5_GPIO_NUM 21 #define Y4_GPIO_NUM 19
#define Y3_GPIO_NUM 18 #define Y2_GPIO_NUM 5 #define VSYNC_GPIO_NUM
25 #define HREF_GPIO_NUM 23 #define PCLK_GPIO_NUM 22
const char index_html[] PROGMEM = R"rawliteral( <!DOCTYPE
HTML><html>
<head>
<meta name="viewport" content="width=device-width, initial-scale=1">
<style>
body { text-align:center; } .vert { margin-bottom: 10%; } .hori{
margin-bottom: 0%; } button {
background-color: #21b555; border: none;
padding: 7px 10px; text-align: center;
font-size: 10px; border-radius: 2px; width: 50%;
color: white;
} </style>
</head> <body
         <div id="container">
<h2>ESP32-CAM Photo Web Server</h2>
<p><button onclick="capturePhoto()">CAPTURE PHOTO</button></p>
<p><button onclick="rotatePhoto();">ROTATE PHOTO</button></p>
<p><button onclick="location.reload();">REFRESH PAGE</button></p>
</div>
<div><img src="saved-photo" id="photo" width="90%"></div> </body>
<script>
var deg = 0;
function capturePhoto() {
var xhr = new XMLHttpRequest(); xhr.open('GET', "/capture", true);
xhr.send();
}
function rotatePhoto() {
var img = document.getElementById("photo");
deg += 90;
if(isOdd(deg/90)){ document.getElementById("container").className
= "vert"; }
else{ document.getElementById("container").className = "hori"; }
```

```
img.style.transform = "rotate(" + deg + "deg)";
}
function isOdd(n) { return Math.abs(n % 2) == 1; } </script>
</html>)rawliteral";
void setup() { Serial.begin(115200);
WiFi.begin(ssid, password);
while (WiFi.status() != WL_CONNECTED) {
delay(1000);
Serial.println("Connecting to WiFi..."); }
if (!SPIFFS.begin(true)) {
Serial.println("An Error has occurred while mounting SPIFFS"); ESP
.restart();
}
else {
delay(500);
         Serial.println("SPIFFS mounted successfully"); }
Serial.print("IP Address: "); Serial.println(WiFi.localIP());
// Turn-off the 'brownout detector'
WRITE_PERI_REG(RTC_CNTL_BROWN_OUT_REG, 0);
// OV2640 camera module
camera_config_t config;
config.ledc_channel = LEDC_CHANNEL_0; config.ledc_timer =
LEDC_TIMER_0; config.pin_d0 = Y2_GPIO_NUM; config.pin_d1 =
Y3_GPIO_NUM; config.pin_d2 = Y4_GPIO_NUM; config.pin_d3 =
Y5_GPIO_NUM; config.pin_d4 = Y6_GPIO_NUM; config.pin_d5 =
Y7_GPIO_NUM; config.pin_d6 = Y8_GPIO_NUM; config.pin_d7 =
Y9_GPIO_NUM; config.pin_xclk = XCLK_GPIO_NUM; config.pin_pclk =
PCLK_GPIO_NUM; config.pin_vsync = VSYNC_GPIO_NUM; config.pin_href =
HREF_GPIO_NUM; config.pin_sscb_sda = SIOD_GPIO_NUM;
config.pin_sscb_scl = SIOC_GPIO_NUM; config.pin_pwdn =
PWDN_GPIO_NUM; config.pin_reset = RESET_GPIO_NUM;
config.xclk_freq_hz = 20000000; config.pixel_format =
PIXFORMAT_JPEG;
if (psramFound()) {
config.frame_size = FRAMESIZE_UXGA; config.jpeg_quality = 10;
config.fb_count = 2;
} else {
```

```
config.frame_size = FRAMESIZE_SVGA; config.jpeg_quality = 12;
config.fb_count = 1;
}
esp_err_t err = esp_camera_init(&config); if (err != ESP_OK) {
Serial.printf("Camera init failed with error 0x%x", err);

         ESP .restart(); }
server.on("/", HTTP_GET, [](AsyncWebServerRequest * request) {
request->send_P(200, "text/html", index_html);
});
server.on("/capture", HTTP_GET, [](AsyncWebServerRequest * request)
{
new_photo = true;
request->send_P(200, "text/plain", "Capturing Photo using ESP32-
CAM");
});
server.on("/saved-photo", HTTP_GET, [](AsyncWebServerRequest *
request) {
request->send(SPIFFS, photo_path, "image/jpg", false); });
server.begin(); }
void loop() {
if (new_photo) {
captureSave_photo();
new_photo = false; }
delay(1); }
// Check if photo capture was successful bool check_photo( fs::FS
&fs ) {
File f_pic = fs.open( photo_path ); unsigned int pic_sz =
f_pic.size(); return ( pic_sz > 100 );
}
// Capture Photo and Save it to SPIFFS void captureSave_photo( void
) {
camera_fb_t * fb = NULL; bool ok = 0;

do {
Serial.println("ESP32-CAMP capturing photo...");
fb = esp_camera_fb_get(); if (!fb) {
```

```
Serial.println("Failed");
return; }
Serial.printf("Picture file name: %s\n", photo_path); File file =
SPIFFS.open(photo_path, FILE_WRITE); if (!file) {
Serial.println("Failed to open file in writing mode"); }
else {
file.write(fb->buf, fb->len);
Serial.print("The picture has been saved in ");
Serial.print(photo_path);
Serial.print(" - Size: "); Serial.print(file.size());
Serial.println(" bytes");
}
file.close(); esp_camera_fb_return(fb);
ok = check_photo(SPIFFS); } while ( !ok );
}
```

## RESULT

The smart irrigation system developed in this project used humidity sensors to detect the moisture level of the soil and watered the plants using a water pump only when the moisture level fell below a certain threshold. The system successfully maintained the moisture level of the soil within the desired range, with an average moisture level of 65%. The smart irrigation system also successfully reduced water consumption compared to traditional irrigation methods and further ensured it was effective in providing the plants with the right amount of water at the right time. The system demonstrated reliable performance over the course of the experiment, with the water pump and humidity sensors functioning as expected. Overall, the results suggest that using humidity sensors to control the water pump is an effective and efficient way to automate the irrigation process and reduce water consumption in agriculture.

Nowadays, technology is growing rapidly and automation is being introduced all over the world. This project is recommended for the user who likes to maintain the garden or for farmers. This recommendation will lead to a green world, which is, it can save a lot of water while watering by setting the motor to automatic mode, where it functions based on the moisture content of the soil.

## INFERENCES

The Internet of things(IOT) is the extension of the current internet to provide communication and connection between different devices. This paper provides a review on ‘IOT based Smart Gardening’ which makes human life easier. There are various implications from the installation of such systems such as ours which includes:

- Conservation: By monitoring the moisture level of the soil and watering the plants only when necessary, this IoT project can help conserve water. It can prevent overwatering, which not only wastes water but can also harm plants.

- Improved Plant Growth: Proper irrigation is essential for plant growth. By monitoring the moisture and temperature levels of the soil, this IoT project can help ensure that the plants receive the right amount of water at the right time, which can improve their growth.

- Energy Efficiency: Smart irrigation systems can be more energy-efficient than traditional irrigation methods. By using IoT technology to automate the watering process, this project can save energy and reduce costs.

- Remote Monitoring: It can be set up to allow remote monitoring of the soil moisture and temperature levels, as well as the watering system. This can be especially helpful for farmers and gardeners who need to monitor their crops or plants from a distance.

- Scalability: This system can be scaled up or down depending on the size of the garden or farm. It can be adapted to work for small gardens or large agricultural operations, making it a versatile solution for smart irrigation.

# CONCLUSION

In conclusion, the smart irrigation system developed in this project has demonstrated the potential for using IoT technology to automate the watering process, conserve water, reduce costs, and improve plant growth. By monitoring the moisture levels of the soil and the temperature and humidity levels of the air, using this information to determine when and how much to water the plants, the system can ensure that the plants receive the right amount of water at the right time.

This  system we have implemented can be adapted to work for small gardens or large agricultural operations, making it a versatile solution for smart irrigation. It can even be integrated into our very own campus grounds and save a lot of the resources and man power.
In addition, the system can be integrated with other IoT systems, such as weather monitoring systems and pest management systems, to provide farmers and gardeners with more comprehensive information about irrigation and crop management.

Proper installation and maintenance of the smart irrigation system are essential for its effectiveness. Regular calibration of the soil moisture and temperature sensors, as well as periodic inspection and repair of the watering system, are necessary to ensure that the system is working properly.

Overall, this project has demonstrated the potential for using IoT technology to improve irrigation and crop management.

# FUTURE WORK

- A camera sensor can be set up that would capture  images of the plant and send the data to a ml model where it would be analysed  using trained models to predict potential diseases.
- The spraying of pesticides can also be made automatic by setting the interval
- Based on the plant the soil ph can be maintained
- A user-friendly interface could be implemented to make the smart irrigation system more accessible to farmers and gardeners. This could include a simple interface with visualisations of data, real-time alerts, and easy-to-use controls for adjusting the system settings.
- The smart irrigation system could be powered by alternative sources such as solar or wind power. This would make the system more sustainable and reduce its environmental impact.
- The water pump could be optimised for efficiency to reduce energy consumption and costs.
- The smart irrigation system could be tested in larger-scale agricultural operations or in different geographic regions with varying climate and soil conditions. This could help to determine the scalability and adaptability of the system and identify any limitations or challenges to implementation.

# REFERENCES

1. H. Abbas, G. M. Ahmed, E. A. Ahmed, R. Ahmed, A. Azeem and A. Seoud, "Smart Watering System for Gardens using Wireless Sensor Networks", 2014.

2. D. Divani and P. Patil, "Automated Plant Watering System", pp. 180-182, 2016.

3. B. C. C. Gaja Priya and M. Abishek Pandu, "Automatic Plant Monitoring and Controlling System over GSM using sensors", pp. 5-8, 2017.

4. I. Srilikhitha, M. M. Saikumar, N. Rajan, M. L. Neha and M. Ganesan, "Automatic irrigation system using soil moisture sensor and temperature sensor with microcontroller AT89S52", 2017 Int. Conf. Signal Process. Commun., pp. 186-190, July 2017.

5. Rao R. Nageswara and Sridhar B. 2018 IoT based smart crop-field monitoring and automation irrigation system International conference on inventive systems and control

6. Rajagopal Shrinidhi and Krishnamurthy Vallidevi 2017 OO Design for an IoT based Automated Plant Watering System IEEE International Conference on Computer, Communication, and Signal Processing

## SNAPSHOT OF TEAM WITH THE PROJECT

## BIODATA OF TEAM MEMBERS

**Name:** Ayesha Numa
**Reg No**: 20BCE1290
**Email Id:** ayeshanuma13@gmail.com
**Number:** +91 7397534398

**Name:** Josiah James
**Reg No:** 20BCE1344
**Email Id:** josiahjgeorge@gmail.com
**Number:** +91 8976183483