Smart gardening

Mayank Goel (2019UCO1558)
Vertika Sah (2019UCO1574)
Garima (2019UCO1579)

Gaurav Singal [a]

[a] Netaji Subhas University of Technology, COE Section -1, Dwarka Sector 3, New Delhi, India, 110078 formerly (NSIT Delhi University)

A preprint

ABSTRACT

Agriculture is critical to the economics and growth of a country like India. Farmers in India have been adopting irrigation techniques in the modern age through manual control, in which they irrigate the ground at regular intervals. This procedure can use a lot of water, or it can take a long time for the water to reach the crops, causing them to dry out. Irrigators who keep track of soil moisture levels in the field have a much better chance of conserving water. The major goal of this study is to create an automated watering system using sensors that are connected to a microcontroller. The sensors used in this project are soil moisture sensor and photoresistor, which sense the moisture content in the soil and sunlight respectively. Based on these values, decisions are made accordingly regarding when to water the plant and when to cover the shed.The problem of taking care of plants has been resolved with the help of our project. We have made a low-cost system which takes less space. It has reduced the wastage of water as it will only water when the plant will require it. The system doesn't need much maintenance and it consumes less power. This setup has made the process smarter. Now the plant will be able to grow properly without external help.

Keywords: smart irrigation, smart gardening, plant health monitoring, smart agriculture, smart home, soil moisture, plant monitoring

1. INTRODUCTION

Agriculture has begun to compete with businesses and other sectors for water use. 84% of the total water available is consumed in irrigation. Because of growing population growth and food consumption, this percentage may rise. As a result, one of the world's challenges is water scarcity. Water is a necessity for the existence of life on earth, including all flora and fauna. Water waste is a major issue in most agricultural settings. Maximum water is wasted due to poor irrigation systems and water management. This can be decreased by employing various water conservation methods that have been developed. This rising demand, combined with rising water and energy prices, has necessitated the development of new technology for

effective water management.
Moreover, when people go on vacation the biggest concern is to make sure that plants at their home don't die. It is very hectic for people to properly water plants according to their needs as each plant has different water requirements and it is very tedious to remember all of them. Irrigation's primary goal is to deliver enough water to plants to avoid stress, diminishing production.The amount and frequency with which water is applied are determined by local meteorological circumstances, crop and growth stage, and soil-moisture plant characteristics.
To solve this problem we will use IoT to develop a tool that will automatically water plants according to their needs, multiple times a day. Our primary objective is to make sure that this system can function in every household. We will try to incorporate as many plants as possible so that it is flexible and can be used in most homes. We will also be providing a movable shed that will protect the plants from sunlight in case they require less sunlight or to save plants from scorching sunlight. If it is the rainy season and plants have already gotten enough water, the soil moisture sensor will sense if there is a requirement and act accordingly.

This automated irrigation system will use an Arduino microcontroller and a soil moisture sensor. With the help of this system, we don't have to worry about the irrigation timings of crops or plants.

- A soil moisture sensor is implemented in our project which measures dryness and sends the data to our python server which decides whether a plant needs to be water depending upon various parameters like water requirement for that specific plant, humidity, temperature and precipitation. This soil moisture sensor also allows us to determine if the water tank is empty or not.
- We have also added a shed in the system which will automatically adjust according to the level of direct sunlight (measured using a photosensor) on the plant. For this, we have used a photoresistor, which is a light-sensitive device and helps in measuring the amount of sunlight.
- We can also water plants and cover the shed using our voice i.e. using Google assistant and Alexa. Moreover, We can see the details of our last irrigation activity using an android application we have a built-in android studio.

Further sections in our paper include section 2 containing related work, section 3 containing methodology, section 4 containing result and analysis & section 5 containing conclusion and future work. In the last, the paper ends with acknowledgement and references.

2. RELATED WORK

We read five papers related to smart irrigation and drew the following learnings.

2.1

The goal of this study was to use a DHT11 sensor and a VH400 to create an automated watering system that an Arduino Uno could control. This system can interact over a cloud as it will also feature a SIM900A module that will provide internet access to connect to the cloud. The major purpose was to confine the water content to the most needed places and lower the farmer's water demand, which will benefit him a lot during periods of low rainfall, with the use of a soil moisture sensor.

Different methods of irrigation studied here are Drip Irrigation and Sprinkler systems. Drip irrigation is done by water drops that are discharged to the plants' roots to minimise water wastage and this makes it the most effective irrigation method. Sprinkler systems use sprinklers, sprays and guns at the end of tubes to irrigate farmlands. It activates only when temperature and humidity reach a particular threshold value.

Arduino Uno sends SMS according to the values it got from the sensors.

2.2

This is a microcontroller-based data processing control system. Depending on the signals received by the sensing mechanism, the pump is actuated to feed water through the tunings connected to the pump.

Because technology is advancing all the time, the main aim of this project is to create a new GSM (Global System for Mobile) controlled soil moisture sensor. Because the gadget is extremely sensitive, care was taken to employ a 5V microcontroller and interface it with a 240 V energy metre, which is used to monitor household power consumption.

The device is made up of an Arduino board, which is a microcontroller that controls the water pump and the Rotating Platform Sprinkler, which distributes water to the plants.

2.3

Automated Irrigation system using WSN(Wireless Sensor Network) and GPRS Module having the main goal is to optimise the use of water for agriculture crops. This system is composed of a distributed WSN with soil moisture and temperature sensor. Gateway units are used to transfer data from sensor unit to base station, send commands to actuators for irrigation control and manage data of sensor units. The algorithm used in the system for controlling water quantity as per requirement and condition of the field. The whole system is powered by photovoltaic panels. Communication is a duplex and takes place through cellular networks. Web applications manage the irrigation through continuous monitoring and irrigation scheduling programming.

The system has two nodes, one node collects all environmental and soil parameter values and the other consists of a camera to capture images and monitor crops.

Sensor data is sent to the base station, and Zigbee is used to receive it. For decision-making, data is processed at the base station. To make decisions on data from the sensor to drip, a data mining algorithm is used. Web apps are used to remotely monitor all observations. For irrigation control, this system uses the Naive Bayes method. If any attribute is not frequent, the algorithm uses a previous data set to make a choice, and the result is zero.

The sensor data and threshold value are saved in the mobile's local memory. The user can read sensor data and put the system to automated mode, which allows the system to switch motors based on sensor data and a previously specified threshold value.

2.4

Water and energy demand are increasing,

necessitating the development of innovative water management technology.

Different crop indications, such as colour change or leaf angle, can help us detect a crop's water needs, but this information may arrive too late. Plant water stress, soil moisture status, and soil water potential are some of the other techniques.

Various disadvantages of traditional irrigation methods are: large amounts of water are used, efficient use of fertilisers is not possible, requires large man work, net yield or productivity is not high, soil erosion etc. Modern Irrigation methods are Sprinkler Irrigation methods, Drip Irrigation methods, and Pot Irrigation methods.

In the proposed automated system, the sensors are embedded at a depth of 1m, a 5V battery is connected, and the moisture sensor produces an analogue output in the 3.3 to 5V range. This result is averaged and compared to a predetermined average value to determine the next course of action. A water level detector will be installed in the water body to alert the user to the need for water. The following are the various requirements:

1. If the moisture level is below threshold value
2. Whether the water level is adequate.
3. If yes then the motor will start otherwise the user will get the message of failure.

After the plant has been adequately watered, the watering will stop, ensuring maximum crop production while conserving water resources and soil nutrients.

Following various calculations, the major features of the system design that would promote efficient water use according to crop requirements were chosen.

2.5

Discussed the need for an ideal irrigation system that intelligently conserves water while making irrigation simple, effective, and crop-enhancing with low waste.

Drip irrigation is a popular irrigation technique because it can irrigate using valves to turn irrigation on and off, increasing yields while reducing water usage.

Controllers and solenoids can be used to automate farm or nursery drip irrigation, which uses about half the amount of water that drip irrigation requires.

The moisture content is measured by sending an input signal to an Arduino board, which is controlled by a microcontroller and is configured to receive input signals from the earth's changing dampness conditions via a dampness detecting device.

In agriculture, soil moisture is crucial. Plants and crops are always more reliant on the moisture available at the root level than on precipitation. Using a system based on a technique for detecting soil water content can save time and money while also increasing output. The technique can be used to build fully automated gardens and farmlands by combining them with the rainwater harvesting principle.

3. METHODOLOGY

Microcontroller, sensors, firebase, python server, google assistant, servo motor, android application and water pump is used to build an automated irrigation system. To make this system less costly, a minimum number of sensors is added. Instead of using sensors to find the value of humidity, and temperature there are various ways by which we can find these data through the internet. OpenWeatherApi provides this data in a fixed interval but to use this service, a premium offer needs to be bought. Another way by which this data can be found is by using a web crawler that searches for the data in Google or any other search engine and with the help of indexing, it reads the particular data and sends it back to the python server from where it is initially called. This project includes a web crawler to find the value of humidity, temperature, and precipitation.

3.1 Sensor Connection

Soil moisture sensor and photoresistor are the two sensors that are used. Both of these sensors are analog sensors and require an Analog to digital converter to be read by a microcontroller. The soil moisture sensor(*A0 pin*) is used to find the moisture content of the soil and the photoresistor(*A1 pin*) is used to measure the light intensity. These analogue sensors are connected with the Arduino Uno which also acts as a power supply to the water pump.

3.2 Arduino Uno and NodeMCu Connection

Serial connection between the Arduino Uno and ESP8266 NodeMCU is made to send the sensor data from Arduino to NodeMCU. The I2C protocol is used to communicate between the Arduino and the NodeMCU. I2C or Inter-Integrated Circuit is a serial bus interface communication protocol. Because it only requires two wires for communication, it's also known as TWI (two-wire interface). Those two wires are SCL (serial clock) and SDA (serial data) . The communication pins on the Arduino are A4 (SDA) and A5 (SCL). This allows us to use our nodemcu as master processor and arduino uno as a slave processor.

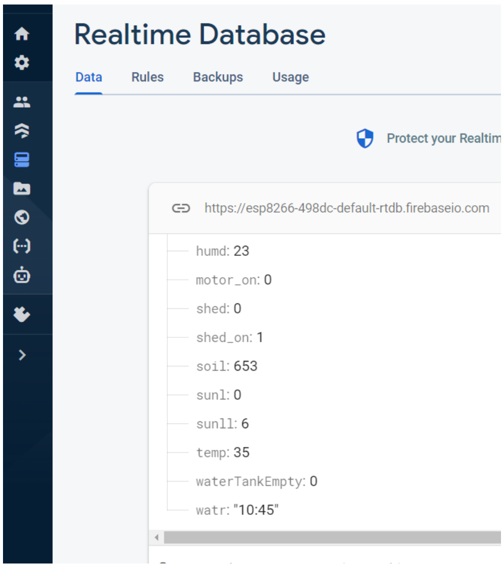
3.3 Firebase and NodeMCu Connection

Firebase Realtime Databases are used as a database to store and retrieve values easily. This is connected with the android application, python server, google assistant and the NodeMCU. NodeMCU has the firmware which runs on an ESP8266 Wifi SoC.
For the connection between NodeMCU and firebase two libraries are used which are FirebaseArduino and ESP8266WiFi.

3.4 Setting up Firebase

New project of Firebase is created for this automation wherein real-time database key-value pairs are created. Following are data objects that are created:-

1. Humd - It stores humidity value
2. motor_on - It is used to command the motor to turn on.
3. Shed -It is used to command the shed to turn on.
4. shed_on - Checks if the shed is on.
5. Soil - It stores soil moisture value.
6. Sunl - It stores value for photoresistor
7. Temp - It stores value for temperature
8. WaterTankempty - To check if the tank is empty
9. Watr last time - This shows when the plant was last watered



3.5 Python Server

Python server acts as a backend for the whole system. It consists of whole logic as to when the water will pump and when the shed is to be turned on. It is connected with Firebase with the help of an API key taken from the firebase.

3.5.1 Web Crawler

Web crawler is an Internet tool which allows our server to go to the internet like an ordinary user, search for a value and read it, without requiring a special API. It is typically used by search engines for page ranking.

BeautifulSoup is a python library used to remove html from the page and take values of ***Temperature, precipitation and humidity***.

With the help of the Request module, we request the weather of the city, in this case, Delhi weather. Then using the BeautifulSoup library we convert the data returned by the request module into a parse tree from which we can extract data in a more readable and hierarchical form.

Using an API would have made our project more expensive as most API provides only a limited number of requests and google weather provides us with more accurate data at no cost. Open Weather API also lacks some small cities but google has a big database that contains almost every city.

3.5.2 How the threshold value is calculated?

Data from the soil moisture sensor is taken every 5 minutes and updated on the firebase database. From this data, we tried to find a perfect threshold value for our plant (Holy basil). We used an app called *PLANTA* to find that our plant needed about 500-600 mL of water every day given the time of year. Assuming that our motor pumps about 10mL of water on every cycle and we are watering our plant for about 10 hours a day we need to water the plant 5 times an hour and taking the snapshot of the soil moisture sensor into account we used machine learning to get a threshold value of 480 to be best suited.

The threshold value is also affected by the humidity and temperature. For example, we map the humidity and precipitation on a scale of 0 to 1 and then calculate the threshold as

Calculated_threshold = original_threshold + 50*humidity + 100*precipitation

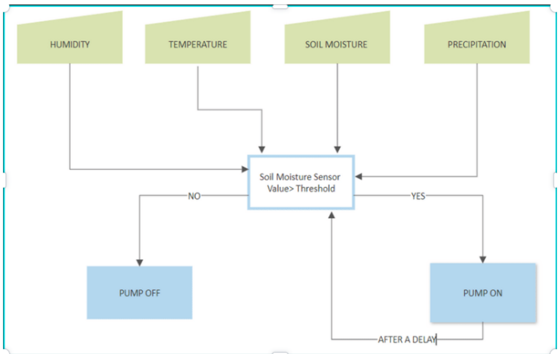
Original_threshold is taken as 480.

This is a very simple model and doesn't account for all variables. This formula has a lot of room for improvement but this works too.

When do our plants get watered?

Our plant gets watered whenever the soil moisture value is greater than our calculated threshold (480), our python server updates

the variable motor_on and nodeMCU turns on the motor for two seconds.
Our android app “SMART GARDENING” provides the user ability to bypass the threshold and water the plant manually.
A Google assistant and Alexa command “WATER THE PLANT” also waters the plant irrespective of soil moisture value.



3.5.3 Shed

Arduino Ide sends the photoresistor value to Firebase which is mapped to 0 to 10 values. This is referenced by the python server. This value is read every thirty-minute and in case it is found out that the photoresistor value is equal to or greater than the threshold value then both the key shed and shed_on are set equal to value 1. After which the Arduino turns the servo motor 180 degrees. Photoresistor value is still being taken after every thirty-minute and when the light intensity becomes lesser than the threshold value then both the key shed and shed_on are set equal to value 0. After which the Arduino turns the servo motor back to its original position.

3.5.4 Water Tank Empty

We developed a smart system which allows us to check if the water tank is empty with just a soil moisture sensor and not using any sophisticated and expensive sensors such as ultrasonic sensor/water level sensor. We have developed an algorithm:

prev_soil_val = curr_soil_val
water_plant()
water_plant()
if(prev_soil_val == cur_soil_val)
    water_tank = empty;
    water_tank_empty()

When the water tank is detected as empty we send a notification to our user using our smart garden app and the user can also be notified using google assistant on his android/iphone.
Also when we find the tank empty we will check soil moisture values in intervals of one hour and check if the tank is refilled, not wanting to damage our water motor by pumping air.

3.6 Android Application

Android has a majority share in the Indian market (>70%) and now with the introduction of Windows 11, the android apps can also run on PCs. So we decided on an android app named "SMART GARDENING" using android studio. This is a basic app that displays real-time information about plants to users:
1) Last water time
2) Ambient temperature
3) Humidity
4) Ambient Sunlight
5) Soil moisture
6) Water tank status
7) Shed status
They can monitor the plant health from anywhere and use special triggers to control actuators (motor, shed). This app can also send notifications/ toast messages whenever plants are water, the shed is triggered and if the water tank is empty. UI for the app is designed keeping usability and aesthetics in

mind. It has support for both light and dark mode and has smart animations.

Because of no use of any specialised framework, our app is very lightweight (about 10MB installed) and doesn't drain any battery. Coding for this app is done in three languages -> java, XML and kotlin. Java & Kotlin for the main code and XML to design our UI and manifest files.

Our app still lacks support for multiple plants and isn't available on the play store yet. There is a need for a new user authentication system and a big database to manage multiple users.

3.7 Voice Assistant

In 2022 most devices understand voice input and take commands from the voice assistant. There is a new trend for people to use their voices to control their smart devices. So we decided to do the same and implement it for google assistant, Alexa and Siri. We used a website called https://wayscript.com that empowered us to use voice to call python functions and connect them securely and efficiently to our python backend server. At present, it has support for commands like

i) water the plant / my plants need water / to irrigate the plants
ii) cover the plant / protect my plant from sun/ there is too much heat
iii) uncover the plant / my plants need sunlight
iv) When were my plants last watered?

These all commands do the needful and provide spoken feedback

We tested these features, Google Assistant on Android & ChromeOS, Alexa on Amazon app and Windows. Unfortunately, we didn't have access to an apple device to check its integration with Siri.

4. RESULT/ANALYSIS

4.1 Performance parameters

Hardware Performance

i) Arduino Uno:

We can’t upload a very big code in Arduino Uno because of its limited memory and RAM. It also doesn’t have built-in Wi-Fi, hence it requires ESP/NODEMCU for web connectivity.

ii) NodeMCU

It has only a single analogue pin and it can provide only 3.3V, maIoTIOT devices require 5V to operate.

iii) Photoresistor

Photoresistor is easy to break and doesn’t hold a candle to industrial-grade sensors.

iv) Servo Motor:

Servo motor causes a lot of errors. It requires high starting current due to which the voltage in other pins lowers. This continues to happen and it appears as if our servo motor is not able to start. We faced this problem a lot as it would take a time for the shed to cover and uncover our plant.

4.2 Experiment Setup

How do we use Machine learning to improve our model?

We needed to make our prototype a lot cheaper for practical use and we found that the soil moisture sensor cost about 300 rupees, which would be saved if we developed a good machine learning model to guess soil moisture value. We created a snapshot of data taken over 6 hours which included last water time, ambient sunlight, ambient temperature, ambient humidity and our target variable soil moisture value. On this data, we did a little processing and changed the last water time to a time interval, and scaled the values. We did a (0.7, 0.1, 0.2) split of training, validation and testing data. Then I made a neural network with 12960 trainable parameters and 3 hidden layers and did training for 20 epochs. We got an accuracy of about 74% on our dataset. But I found that some of the parameters didn't change much in my snapshot, like humidity, sunlight, temperature and 6 hours was a very short time. The model did a good prediction and provided a previous true value, but over time (about 1-1.5 hours) our predicted value was off by almost 120 which is a lot given that our sensor provides values in the range 0 to 1024.

This problem can be solved if taking a larger snapshot of about a week.

Fig 4.1 shows the whole workflow of our model.

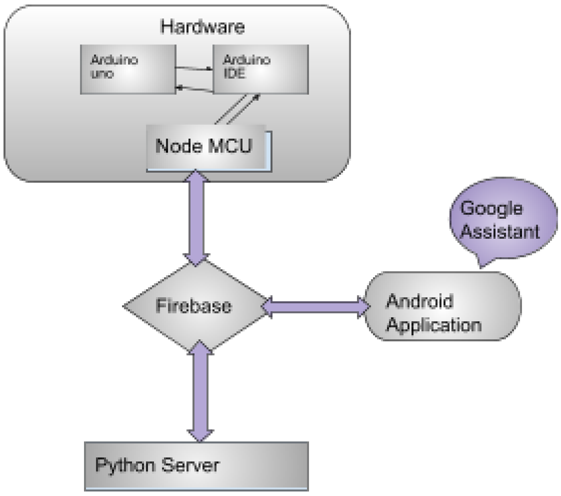


Fig 4.1

4.3 Experiment Result

We successfully created a low-cost smart irrigation system for homes/gardens and also provided easy monitoring with an android app and voice assistant integration. Our prototype cost about 1000 rupees but we can easily reduce it to 600 rupees for a final model and an extra 400 for an additional plant each.

Problems faced:

i) Firebase integration with ESP8266

We needed to connect our project with the firebase server but ESP8266 does not support using the firebase server. There are many libraries to connect it with thingspeak or a local network but no firebase library.

There are some guides online on how to use Arduino as a connector board for ESP8266 and ESP8266. We spent a lot of days trying to flash our code and update the firmware of ESP8266 for it to work with firebase or other servers/databases. We needed a real-time database for its connection with our server. Arduino has limited storage, so we also couldn't upload any big code. We tried and found that the code straight up refused to get uploaded and even if we somehow flashed it, we were facing a lot of stability issues and it crashed. Then we resigned and bought a nodeMCU module which caused its problems.

ii) Connection of Arduino UNO and NodeMCU

NodeMCU has only one analogue port and we have a total of two analogue sensors, we either required an analogue multiplexer or an analogue to digital converter. We went for a convertor, but we got a defective model. So we thought of connecting our analogue sensors with Arduino UNO with our NodeMCU and sending data between them. Here our NodeMCU acted as a master processor and UNO as a slave. We used wire library and I2C protocol for communication and we witnessed a new problem for I2C connection that it wasn't transferring the integer values, the library only supports sending characters. We also required two processors as two processors can't be interfaced at the same time on Arduino IDE. So we mapped the value on a scale of 0 to 255 and sent it to NodeMCU.

iii) Problems with the thingspeak server

ThingSpeak helps in storing, visualising and analysing our data. We can access and retrieve data from it by using the API write and read key provided by it.
But as we had other applications such as android and voice assistant we needed additional features which can not be provided by the ThingSpeak server.
Hence, instead of using ThingSpeak, we made use of Firebase. Through Firebase we were able to communicate with the python server, android application and voice assistant.

iv) Low powered motor

Water pump required only 5V power which was being supplied through Arduino Uno. Initially, a long pipe was added to water the plant even if the prototype was at an appropriate distance but as the motor was low powered it wasn't able to pump water with much force that it could reach the plant. Also the altitude at which the pipe is positioned mattered as it would only water plants if it is titled in downward direction. This created a problem as in case the altitude changes our system will not be able to water the plant. While we were making our prototype, when we started the system, water was not coming out of the pipe but as soon as we changed the altitude of the pipe, water came out and it fell on the laptop .So we included a short pipe which can be bent in any direction to avoid such further accidents.

5.CONCLUSION AND FUTURE WORK

The automated irrigation prototype is functional, low powered and cost-effective. This system incorporates drip irrigation which will water the plant at the root while using the minimum amount of water possible. This will further help in regions

where there is water scarcity. Due to drip irrigation, there will be full utilisation of water and no wastage because of evaporation.
The shed protects the plants from the sun and automates the process providing comfort to the user. App and Google Assistant commands work fine and provide easy interfacing.
The tedious task of irrigation is made easy with the help of our smart plant monitoring system at a reasonable cost.

Future Improvements:

- Using advanced leaf sensors such as leaf moisture detection, but this sensor would have made the product economically infeasible for households who are our users
- Using further data to remove dependency on soil moisture sensor
- Making a website for the product so that it can be easily accessed via web
- Collect data on more plants and make a database
- Making the system waterproof for rainy days

6. ACKNOWLEDGEMENT

The authors wish to express their gratitude towards the management of Netaji Subhash University of Technology(NSUT)(formerly NSIT(Delhi University)) and Professor Gaurav Singal for providing this wonderful opportunity and for his support. He helped bring new ideas and in debugging our code.

7. REFERENCES

1. Yang, G., Liu, Y., Zhao, L., Cui, S., Meng, Q., & Chen, H. (2010, June). Automatic irrigation system based on wireless network. In *IEEE ICCA 2010* (pp. 2120-2125). IEEE.
2. Sadasivam, S., Vadhri, V., & Ramesh, S. (2015). *Smart Plant Monitoring System* (Doctoral dissertation, Thesis/Distertation, Informatik, Technische Universitat Darmstadt, Darmastatd).
3. Taneja, K., & Bhatia, S. (2017, June). Automatic irrigation system using Arduino UNO. In *2017 International Conference on Intelligent Computing and Control Systems (ICICCS)* (pp. 132-135). IEEE.
4. Gunturi, V. N. R. (2013). Micro controller based automatic plant irrigation system. *International Journal of Advancements in Research & Technology*, *2*(4), 194-198.
5. Nallani, S., & Hency, V. B. (2015). Low power cost effective automatic irrigation system. *Indian Journal of Science and Technology*, *8*(23), 1.
6. Rajpal, A., Jain, S., Khare, N., & Shukla, A. K. (2011). Microcontroller-based automatic irrigation system with moisture sensors. In *Proceedings of the International Conference on Science and Engineering* (pp. 94-96).
7. Sudha, M. N., Valarmathi, M. L., & Babu, A. S. (2011). Energy efficient data transmission in automatic irrigation system using wireless sensor networks. *Computers and Electronics in Agriculture*, *78*(2), 215-221.
8. Rani, M. U., & Kamalesh, S. (2014, January). Web based service to monitor automatic irrigation system for the agriculture field using sensors. In *2014 International conference on advances in electrical engineering (ICAEE)* (pp. 1-5). IEEE.

9. Vagulabranan, R., Karthikeyan, M., & Sasikala, V. (2016). Automatic irrigation system on sensing soil moisture content. *International Research Journal of Engineering and Technology (IRJET)*, *3*(03), 206-208.
10. Rajendranath, U. N. V. P., & Hency, V. B. (2015). Implementation of an Automated Irrigation System. In *Smart Irrigation System. Presented at the IEEE Sponsored 2 nd International Conference on Innovations in Information, Embedded and Communication systems (ICIIECS)*.
11. Awati, J. S., & Patil, V. S. (2012). Automatic Irrigation Control by using wireless sensor networks. *Journal of Exclusive Management Science*, *1*(6), 1-7.
12. Kumar, B. D., Srivastava, P., Agrawal, R., & Tiwari, V. (2017). Microcontroller based automatic plant irrigation system. *International Research Journal of Engineering and Technology*, *4*(5), 1436-1439.
13. Cancela, J. J., Fandiño, M., Rey, B. J., & Martínez, E. M. (2015). Automatic irrigation system based on dual crop coefficient, soil and plant water status for Vitis vinifera (cv Godello and cv Mencía). *Agricultural Water Management*, *151*, 52-63.
14. Cancela, J. J., Fandiño, M., Rey, B. J., & Martínez, E. M. (2015). Automatic irrigation system based on dual crop coefficient, soil and plant water status for Vitis vinifera (cv Godello and cv Mencía). *Agricultural Water Management*, *151*, 52-63.
15. Bhaskar, L., Koli, B., Kumar, P., & Gaur, V. (2015, September). Automatic crop irrigation system. In *2015 4th International Conference on Reliability, Infocom Technologies and Optimization (ICRITO)(Trends and Future Directions)* (pp. 1-4). IEEE.