IOT IN GREEN HOUSE USING MQTT PROTOCOL

By

GANESH KUMAR DHANAPALAN

A Project Submitted in Partial Fulfillment
of the Requirements for the Degree of

Master of Science in Computer Science

SUPERVISED BY
DR. SABAH MOHAMMED

Department of Computer Science

Lakehead University

Thunder bay, Ontario, Canada.

May 2019

Abstract

Microcontroller ATmega328P is used in the field of agriculture to monitor various plant parameters. The use of Arduino UNO and sensors eases the work of a farmer. They fulfill the duties of automatic water irrigation using soil moisture sensor, determining the necessary pH content, light intensity, humidity, and temperature of the air and soil which are smartly maintained. All these parameters work well automatically, and they are based on a specific threshold value. When the sensor value goes beyond that threshold value, the actuators get initiated automatically. The sensor readings are derived and stored. An android application is developed to keep track of the data and also to activate the actuators manually by using the app whenever the user desire. The sensor data are stored in the google sheet which acts as the cloud storage and are also stored in the sd card using Arduino sd card module. The sensor data are accessed by the android application using the mqtt protocol

Table of Contents

Abstract ii
List of Tables iv
List of Figures v
Acknowledgements vi
CHAPTER 1 1
INTRODUCTION 2
CHAPTER 2 3
LITERATURE REVIEW 4
2.1 Overview 4
2.2 Mind Map of Keywords 5
2.3 Grouping of Papers 6
2.3.1 Group 1 (Architecture) 6
2.3.2 Group 2 (Effects of IoT on Agriculture) 8
2.3.3 Group 3 (Advancements) 10
2.3.4 Group 4 (Real Time Implementation Examples) 14
2.4 Research Problem Definition 16
2.4.1 Group 1 Considerations 16
2.4.2 Group 2 Considerations 16
2.4.3 Group 3 Considerations 16
2.4.4 Group 4 Considerations 17
2.5 Hypothesis 17
CHAPTER 3 18
METHODOLOGY 19
3.1 Overview 19
3.2 System Architecture 19
3.3 System Visualization 19
3.3.1 Class Diagram 19
3.3.2 Use Case Diagram 21
3.3.3 Deployment Diagram 23
3.3.4 Sequence Diagram 24
3.3.5 Component Diagram 25

CHAPTER 4 27
PROTOTYPE 28
4.1 System Design 28
4.2 Hardware and Software Requirements 39
4.3 Experimental Setup 39
4.4 Working 45
4.5 Output 53
4.6 Screenshots 54
CHAPTER 5 61
CONCLUSION AND FUTURE SCOPE 62
5.1 Conclusion 62
5.1.1 Benefits 62
5.1.2 Drawbacks 62
5.2 Future Scope 62
CHAPTER 6 63
BIBILIOGRAPHY 64

List of Tables

Table 2.1: List of papers on group 1 6
Table 2.2: List of papers on group 2 8
Table 2.3: List of papers on group 3 11
Table 2.4: List of papers on group 4 14
Table 4.1: Hardware interfacing between arduino Uno and DHT 22 41
Table 4.2: Hardware interfacing between arduino Uno and Ph sensor........ 42
Table 4.3: Hardware interfacing between arduino Uno and LDR 42
Table 4.4: Hardware interfacing between arduino Uno and Soil moisture sensor 42
Table 4.5: Hardware interfacing between arduino Uno and DS18B20 43

List of Figures

Figure 2.1: Mind map of keywords.. 5
Figure 3.1: Class diagram of the system.. 20
Figure 3.2: Use case diagram of the mobile application.. 22
Figure 3.3: Deployemnet diagram of the system 23
Figure 3.4: Sequence diagram of the system 24
Figure 3.5: Component diagram of the system... 26
Figure 4.1: Block diagram of the system... 40
Figure 4.2: List of Sensors.. 41
Figure 4.3: List of Actuators... 44
Figure 4.4: Flow Diagram sensor device intraction.. 46
Figure 4.5: Flow Diagram Actuator device interaction 49
Figure 4.6: Flow Diagram Arduino Uno-Node MCU8266 Interaction 51
Figure 4.7: Flow Diagram device cloud interaction 52
Figure 4.8: Flow Diagram cloud mobile interaction... 53
Figure 4.9: Screenshot of cloud MQIT instance 54
Figure 4.10: Screenshot of google sheet.. 55
Figure 4.11: Login page of android application.. 50
Figure 4.12: Registration page of android application.. 22
Figure 4.13: Sensor value display in android application.. 23
Figure 4.14: Manual mode operation in android application... 24
Figure 4.15: Notification in android application... 26

Acknowledgments

The success and outcome of this project required a lot of guidance and assistance from many people, and I am incredibly fortunate to have got this all along the completion of my project work. Whatever I have done is only due to such guidance and assistance and would not forget to thank them.

I am highly indebted to Dr. Sabah Mohammed, for giving me an opportunity to do the project under her guidance on IoT and providing me all support and advice which made me complete the project on time. I am incredibly grateful to her for providing such excellent support.

I want to express my special appreciation and thanks a lot to the Computer Science Graduate department at Lakehead University for giving me a fantastic opportunity to attest myself.

In end, I wish to reward myself of this opportunity, express a wisdom of appreciation and love to my friends and my beloved parents for their communal sustenance, strength, and inspiration which helped me in the accomplishment of this project.

Ganesh Kumar Dhanapalan

CHAPTER 1

INTRODUCTION

There are resources wasted during farming due to over usage without knowing. In this project, resources wasted during the agricultural processes are taken into concern using the Internet of Things. Resources such as water and light are used in abundance during the growth of a plant. Various sensors are used to minimize resources and maximize agricultural efficiency connected with Arduino UNO. This project concentrates on measuring soil moisture and providing the required water content using FC28. Measuring the light intensity surrounding the plant and providing the required electrical resources. Light intensity is measured using a photo resistor. Low light intensity is detected when the resistance of the photo resistor is high. Actuators respond to the sensor's values and are used to on/off the resources. The Arduino is serially connected with node MCU. Using the node MCU which has Wi-Fi capabilities, all the values generated in the Arduino IDE are stored in the google sheet using push bullet service and are also stored locally in SD card module. The SD card module is connected with the Arduino UNO. Using the MQTT protocol, these values are displayed within the android application for the user's purpose. Users can check the changing sensor values from time to time and can also control the actuators by themselves.

The agricultural environment also depends upon the factors like pH level, temperature, and humidity. In this case, the DHT11 sensor is used to measure temperature of the air and humidity. After measuring, to solve the problem of humidity, a humidifier is made from scratch and controlled by an actuator. The DS18B20 has been used to monitor the temperature of soil.

An android application has been developed for controlling and monitoring the sensor data. The application has features like cloud login, signup that allows a first-time user to register and to control the actuators. It takes personal information. The major uses of this android application are to manually control the actuators to on/off the resources and to monitor the sensors in real-time using cloud mqtt.

CHAPTER 2

LITERATURE REVIEW

This section will describe about the literature review of the important elements that are to be considered while developing the precision agriculture. It consist of the explanations about the related research and introduction of the important terms and their definitions.

2.1 OVERVIEW:

Smart Agriculture helps to reduce wastage, effective usage of fertilizer and thereby increase the crop yield. A framework is produced to screen crop-field utilizing sensors (soil dampness, temperature, mugginess, Light) and robotize the water system framework. The information from sensors is sent to the web server database utilizing remote transmission. In the server database, the data is encoded in JSON format. The water system is robotized if the dampness and temperature of the field fall underneath the verge. In nurseries, light power control can likewise be mechanized in expansion to the water system. The notices are sent to agriculturists' portable intermittently. The agriculturists' can be ready to screen the field conditions from anyplace. This framework will be progressively valuable in regions where water is in scarce (K. A. Patil , N. R. Kale, 2016).

In-field observing and control of agribusiness and sustenance creation can be radically enhanced by receiving the IoT idea. Such frameworks ought to guarantee the nature of items and may improve the administration and observing of ranches. Additionally, the IoT framework for agricultural applications ought to give the inside and out an examination of use's necessity and brilliant use of available assets. The idea of IoT brilliant agribusiness is additionally tending to the wellbeing of agrarian items and the trading of data among makers and customers (Gordana Gardasˇevic´, Mladen Veletic, Nebojsˇa Maletic´, Dragan Vasiljevic´, Igor Radusinovic´, Slavica Tomovic´, Milutin Radonjic´, 2016).

Programming for IoT applications must help an assortment of equipment stages. Because of memory and CPU limitations, the intricacy of activities must be low. Dissimilar to conventional OSs, the IoT OS needs to adapt to numerous constraints and weaknesses, just as with heterogeneity of gadgets. The OS for embedded gadgets ought to have a lightweight structure, in this way empowering the low code estimate, low intricacy, and low power utilization. The develop and all around tried installed OSs for WSN applications are TinyOS, Contiki, and Linux. As of late, the FreeRTOS, RIOT, and OpenWSN have been presented. Business arrangements, for example, Sensinode's NanoStack (TM) 2.0 and the Thingsquare stage, are additionally accessible on the market. In May 2015, Google declared the Brillo OS for IoT network, and the Weave, a correspondences standard for shrewd gadgets (Gordana Gardasˇevic´, Mladen Veletic, Nebojsˇa Maletic´, Dragan Vasiljevic´, Igor Radusinovic´, Slavica Tomovic´, Milutin Radonjic´, 2016).

2.2 MINDMAP OF KEYWORDS:

Fig 1.1: Mind Map of Keywords

2.3 GROUPING OF PAPERS:

In this section the surveyed or referred papers are divided into groups based upon their similarities. All of these below papers are taken for the consideration for implementing the indoor home garden. Here the papers are divided into five groups based upon their similarity in their context. The different groups and their content are listed below, they are
GROUP 1 – Architecture
GROUP 2 – Effects of IoT on agriculture
GROUP 3 – Advancements
GROUP 4 – Real time implementation examples

2.3.1 GROUP 1: (ARCHITECTURE)

TITLE	AUTHOR	YEAR
The Internet of Things: Challenges & security issues	Gurpreet Singh Matharu , Priyanka Upadhyay , Lalita Chaudhary	2014
The IoT Architectural Framework, Design Issues and Application Domains	Gordana Gardaševic, Mladen Veletic, Nebojša Maletic, Dragan VasiljevicIgor Radusinovic, Slavica Tomovic	2017

Table 2.1 List of Papers on Group 1

The general structure of the IoT consist of four layers. The first layer is called perception layer, which is also known as the sensing layer. This layer is responsible for the integration of hardware and collection of data from the sensors. It supports the sensing technologies like WSN (Wireless Sensor Network) and RFID. The second layer is known as the middleware layer which is arbitrated between the network and application layer. This is the most important layer since it is responsible for ensuring the interoperability, scalability and providing services to the customers. It handles the functionalities like data storage, analytics and visualization. The third layer is called network layer which implements the support services for securing the data transfer over the sensor networks. It uses the network technologies like 3G, Wifi, Bluetooth, etc. It handles the functionalities like data aggregation and addressing schemes. The last layer is called application layer which is responsible for the delivery of services (Gurpreet Singh Matharu, Priyanka Upadhyay, Lalita Chaudhary, 2014).

The 3-layered architectural framework in IOT provides the efficient and effective functionalities for the system. This leads to the developed framework of the system. Each layer performs some set of tasks, which are interrelated. The 3-layers are perception layer, transport layer, and application layer. The perception layer acts as the eye for the system. The transport layer is responsible for the delivery of information through various protocols and the application layer provides the intelligent tools that are necessary for processing the information (Fu Bing,2006).

At the transport layer, two standard conventions might be utilized: Transmission Control Protocol (TCP) and User Datagram Protocol (UDP). The UDP is ideally used for 6LoWPAN because of its lightweight usage. The stream control system in TCP produces the overhead that is viewed as unreasonably high for LLNs. Be that as it may, as of late created IoT equipment stages have enough memory and preparing assets to help the transmission over TCP (Gordana Gardasˇevic´, Mladen Veletic, Nebojsˇa Maletic´, Dragan Vasiljevic´, Igor Radusinovic´, Slavica Tomovic´, Milutin Radonjic´, 2016).

The instruments at the network layer are actualized by the IPv6 convention. A delegate at this layer is the Routing over Low-control and Lossy Networks (RPL) convention, institutionalized by the IETF ROLL gathering. The RPL is a connection autonomous separation vector steering convention structured explicitly for LLNs, where the associations between hubs (switches) are lossy and with low parcel conveyance rates. Such systems, as a rule, do not have a predefined topology and may send countless. Correspondingly, the job of RPL is to find ideal connections dependent on a built-up set of criteria. This convention underpins different traffic models, for example, point-to-direct, point-toward multipoint, and multipoint-to-point. The IP Security (IPSec) convention is empowered at system layer to help the protected interchanges (Gordana Gardasˇevic´, Mladen Veletic, Nebojsˇa Maletic´, Dragan Vasiljevic´, Igor Radusinovic´, Slavica Tomovic´, Milutin Radonjic´, 2016).

Programming for IoT applications must help an assortment of equipment stages. Because of memory and CPU limitations, the intricacy of activities must be low. Dissimilar to conventional OSs, the IoT OS needs to adapt to numerous constraints and weaknesses, just as with heterogeneity of gadgets. The OS for embedded gadgets ought to have a lightweight structure, in this way empowering the low code estimate, low intricacy, and low power utilization. The develop and all around tried installed OSs for WSN applications are TinyOS, Contiki, and Linux. As of late, the FreeRTOS, RIOT, and OpenWSN have been presented. Business arrangements, for example, Sensinode's NanoStack (TM) 2.0 and the Thingsquare stage, are additionally accessible on the market. In May 2015, Google declared the Brillo OS for IoT network, and the Weave, a correspondences standard for shrewd gadgets (Gordana Gardasˇevic´, Mladen Veletic, Nebojsˇa Maletic´, Dragan Vasiljevic´, Igor Radusinovic´, Slavica Tomovic´, Milutin Radonjic´, 2016).

A critical prerequisite in structuring the IoT gadgets is identified with "least energy" idea. This necessity may likewise be spread to general structure approach in making IoT equipment and programming segments. The IoT gadgets run from exceptionally lightweight sensors controlled by 8-bit microcontrollers (MCUs) to devices furnished with all the more dominant, yet vitality proficient 32-bit processors. These gadgets depend on various models, for example, x86, MSP430, ARM7, Atmel, AVR, Cortex-M0, Cortex-M3, Cortex-M4 (Gordana Gardasˇevic´, Mladen Veletic, Nebojsˇa Maletic´, Dragan Vasiljevic´, Igor Radusinovic´, Slavica Tomovic´, Milutin Radonjic´, 2016).

PROS:

- All of these papers provide the insight and theoretical knowledge that are required to develop the IoT application.
- It provides the layer wise approach for the development of application.
- It emphasis the need for the least energy consumption of the iot devices.

CONS:

- Not all the layers are required for developing the application.
- Some of the applications are developed using three layer approach and some by using four layer approach. But it doesn't provide the efficiency difference between three layer and four layer approach.

2.3.2 GROUP 2: (Effects of IoT on Agriculture)

TITLE	AUTHOR	YEAR
Agricultural crop monitoring using IOT - a study	D.K. Sreekantha ; Kavya A.M.	2017
Web-based monitoring of an automated fertigation system: An IoT application	Shah Abd Hafiz Zainal Abidin , S. Noorjannah Ibrahim	2015
Green house by using IOT and cloud computing	Sheetal Vatari ; Aarti Bakshi ; Tanvi Thakur	2016
Early detection of grapes diseases using machine learning and IoT	Suyash S. Patil , Sandeep A. Thorat	2016
Research on effect factors evaluation of internet of things (IOT) adoption in Chinese agricultural supply chain	Danping Lin , C. K. M. Lee , Kangwei Lin	2016
Wireless Sensor Network in precision agriculture application	Mohamed Rawidean Mohd Kassim , Ibrahim Mat , Ahmad Nizar Harun	2014

Table 2.2 List of Papers on Group 2

The Internet of things (IOT) is rebuilding the horticulture empowering the ranchers with the broad scope of systems, for example, exactness and economical horticulture to confront difficulties in the field. IOT innovation helps in gathering data about conditions like climate, dampness, temperature, and fruitfulness of soil. Web-based checking empowers location of weed, the dimension of water, bug recognition, creature interruption on the field, change in development, and farming. IOT use agriculturists to get associated with his ranch from anyplace also, whenever. Remote sensor systems are utilized for checking the ranch conditions and small scale controllers are utilized to control and computerize the ranch forms. The remote cameras have been used for visualizing the conditions of the farm in the form of picture and video. An advanced mobile phone engages rancher to keep refreshed with the progressing states of his farming by utilizing IOT whenever and any piece of the world. IOT innovation can lessen the expense and improve the efficiency of conventional cultivating (Dr. D.K. Sreekantha, Kavya.A.M, 2017).

Apart from the soil conditions, fertilizers play a critical role in fighting the diseases and producing healthy plant by providing the required nutrition to the plants. The system includes web-based user application, an automatic fertigation system and a communication network. The data from the web-based application is stored in the SQLite database because of its features. This type database engine could be handled by microprocessor itself. It also comprises of the emergency mode due to which the fertigation process can be stopped manually (Shah Abd Hafiz Zainal Abidin and S.Noorjannah Ibrahim, 2015).

There are numerous methods accessible for the exactness of agribusiness to screen and control, condition for the development of multiple products. Because of unequal conveyance of rainwater, it is harsh for the farmer to determine the necessity of the water for the harvests in the entire homestead. It needs some water system strategy that is appropriate for any climate condition, soil types and an assortment of yields. Green House is the best answer to control and manage this issue. It is increasingly critical to look through a technique that gives impeccable examination and control to create a legitimate condition. Huge zones secured by sensor arrange can build up greenhouse with exactness condition required for various yields. This condition is developed by utilizing two advancements: IoT and distributed computing. By using IoT(Internet on things) we control gadgets or any ecological needs whenever anyplace and the cloud which gives stockpiling and figuring assets to actualize a website page (Sheetal Vatari, Aarti Bakshi, Tanvi Thakur, 2016).

To avoid illnesses agriculturists' splash tremendous measure of pesticides, which bring about expanding the expense of creation. Additionally agriculturists cannot recognize the sicknesses physically. The illnesses are distinguished simply after the disease, yet its takes up a great deal of time and effect sly affect vineyard. The advancement of an observing framework will distinguish the odds of grape ailments in its beginning periods by utilizing Hidden Markov Model, which gives alarms by means of SMS to the rancher and the master. The framework incorporates temperature, relative stickiness, dampness, leaf wetness sensor and Zig-Bee for remote information transmission (Suyash S. Patil, Sandeep A. Thorat, 2016).

The Internet of Things in China was adopted for agricultural supply chain by the construction of Technology-Organization-Environment (TOE) framework. The information was examined utilizing Structural Equation Modeling. Through insights investigation, the impact factors were perceived and the TOE demonstrate was altered fittingly. The outcomes showed that opposition from representatives and vulnerabilities are not vital elements that impact the IoT reception. Alluding to those bolstered variables, specialized components (multifaceted nature, similarity, perceived benefit, and cost) affect the innovation selection of IoT in agriculture. Furthermore, authoritative elements (size of big business, official help, trust among the organizations in the production network, and specialized information) and natural variables (outer weight and government bolster) all have positive associations with IoT reception (Danping Lin, C. K. M. Lee, Kangwei Lin, 2016).

NPK(Nitrogen, Phosphorous, Potassium) is one of the main essential content in the soil. The presence of these elements in soil determine the growth of the plant. Nitrogen determines the leaf growth and the presence of green color in the plant. Phosphorous helps the plant to create new roots, make seeds, flowers and fruits. It also prevent and protect the plant from diseases. Potassium helps to make strong stems and determines the growth of the plant. Thus monitoring of these elements in soil becomes essential when it comes to agriculture. The NPK sensors are used in the field to monitor its composition in soil, which in turn helps to grow healthy plants (Mohamed Rawidean Mohd Kassim, Ibrahim Mat, Ahmad Nizar Harun, 2014).

PROS:

- The use of pesticides for protecting the plants from the diseases.
- The monitoring of the soil conditions for maintaining the plant growth.
- The determination of the amount of water required for the plant based upon its condition.

CONS:

- Not all the factors that are needed for the plant growth has been considered.

2.3.3 GROUP 3: (Advancements)

TITLE	AUTHOR	YEAR
Development of IoT based smart security and monitoring devices for agriculture	Tanmay Baranwal , Nitika , Pushpendra Kumar Pateriya	2016
Design and implementation of a cloud-based IoT scheme for precision agriculture	Ahmed Khattab , Ahmed Abdelgawad , Kumar Yelmarthi	2016

Monitoring system using web of things in precision agriculture	FoughaliKarim, Fathalah Karim, Ali Frihida	2017
Agricultural Production System Based on IoT	Meonghun Lee , Jeonghwan Hwang , Hyun Yoe	2013
Providing Smart Agricultural solutions to farmers for better yielding using IoT	M.K. Gayatri , J. Jayasakthi , G.S.Anandha Mala	2015
Implementation of IoT (Internet of Things) and Image processing in smart agriculture	Ayush Kapoor , Suchetha I Bhat , Sushila Shidnal , Akshay Mehra	2016
Cloud of Things in Smart Agriculture: Intelligent Irrigation Monitoring by Thermal Imaging	Mehdi Roopaei , Paul Rad , Kim-Kwang Raymond Choo	2017
A web-based IoT solution for monitoring data using MQTT protocol	Krešimir Grgic, Ivan Špeh, Ivan Hedi	2016
The IoT Architectural Framework, Design Issues and Application Domains	Gordana Gardaševic, Mladen Veletic, Nebojša Maletic, Dragan VasiljevicIgor Radusinovic, Slavica Tomovic	2017
Field Monitoring and Automation using IOT in Agriculture Domain	Mohanraj I, Kirthika Ashokumarb, Naren Jc	2016
The Application of Cloud Computing and the Internet of Things in Agriculture and Forestry	Yifan Bo , Haiyan Wang	2011

Table 2.3 List of Papers on Group 3

The remote monitoring of the agricultural farm plays a vital role in the precision agriculture. The various monitoring aspects of the agricultural farms are temperature, relative humidity, light, soil moisture, pH. These when integrated with a camera provides the solution for the security aspect of the farm. When incorporated with a three –layered architecture, it enhances the efficiency of the system (Tanmay Baranwal, Nitika, Pushpendra Kumar Pateriya, 2016).

The analysis of the data plays a vital role in making the precise decision. This can further lead to the automation of the process. The inclusion of the cloud computing in agricultural system can change it into precision agriculture. This system consist of three-layered architecture where each layer performs a set of tasks that are interrelated to other layers. The 3-layer architecture consist of front-end layer, the gateway layer and the back-end layer. The front-end layer is the physical hardware of the system whose sole purpose is to collect the sensor data and relay it to the gateway layer. The gateway layer is responsible for relaying the collected data to the cloud servers and forward the requests from the back-end to the actuators in the node. Finally, the back-end layer is responsible for facilitating the end-users ability in accessing the sensed data (Ahmed Khattab, Ahmed Abdelgawad, Kumar Yelmarthi, 2016).

To optimize water use, farmers' use of a decision-support system is unavoidable. The real-time supervision of microclimatic conditions is the only way to know the water needs of cultivation. Remote sensor systems are assuming a vital job with the appearance of the Internet of things and the speculation of the utilization of web in the network of the agriculturists. It will be sensible to make supervision conceivable by using web administrations. The IOT cloud speaks to stages that permit to make web administrations appropriate for the objects integrated on the Internet (Foughali Karim, Fathalah Karim , Ali frihida, 2017).

The statistical prediction of the outcome helps the farmers to make a decision on the agricultural field. This prediction can be about the climatic conditions, soil conditions etc. This form of analysis are done before planting the agricultural field. This provides the farmers with the insights and the forecasting knowledge that drives them to take optimized decisions in planning. It supports the farmers in the business model (Meonghun Lee, Jeonghwan Hwang, and Hyun Yoe, 2013).

The Cloud computing techniques can be used to improve the agricultural business. These techniques can create the computing ecosystem by which the data from the sensors and tools were observed and accurately fed into repositories. These technologies has the ability to store and process a wide variety of data types like sensor data(analog or digital), image etc. This creates the path for handling the large amount of real time data (M.K.Gayatri, J.Jayasakthi, Dr.G.S.Anandha Mala, 2015).

The inclusion of image processing feature in the precision agriculture helps to monitor the plant as a human. A serial JPEG Camera module is used to capture the image of the plant at regular intervals. These images are then stored for analyzing. It can be used for the plants that show moderate to drastic morphological changes within short span of time. The images taken by the camera module are tested against the test images that are taken under subtle environment and stored in a pre-defined database. The various algorithms are then used analyze the images and arrive at a conclusion. The conclusion are made with the set of histograms (Ayush Kapoor, Sushila Shidnal, Suchetha I Bhat, Akshay Mehra, 2016).

Thermal imaging is a non-contact and nonintrusive system, without the requirement for changes in the surface temperature. It is likewise fit for showing the temperature. This has been utilized in numerous modern as well as research fields when the temperature speaks to a critical variable, including meteorology, ecological investigations, therapeutic diagnostics, and design. A few surveys have shown that thermal imaging is a fitting way to deal with recognizing key parameters to plan water system. There are a few essential highlights for the water system, for example, water pressure, gas-conversion standard, evapotranspiration rate, stomatal conductance, and shutting of stomata. In water pressure condition, stomata start to close and stop to come to pass, plant warms up, and the covering temperature will rise. Along these lines, warm remote detecting can conceivably be utilized to quantify plant temperature, stomatal conductance, and evapotranspiration rate by assessing stomatal responses. Thermal imaging has the benefit of giving a temperature incentive to the majority of the pixels inside the sensor's field of view in contrast with thermometry, where the last just gives normal esteem. Along these lines, it is every so often simpler to separate between various segments, for example, sunlit versus secured plant segments and wet against dry soil surfaces (Mehdi Roopaei, Paul Rad, Kim-Kwang Raymond Choo, 2017).

An electronic Internet of things arrangement is gone for checking, following and examining information in horticulture region. The reason for this arrangement is to encourage verification of various assembling process in referenced territory utilizing IoT advancements. A piece of continuous information is accomplished by using agent-based distributing/buying in Message Queue Telemetry Transport convention. Utilization of this convention helps to avoid the resulting constrained networks in rural areas. Gathered information from sensors is appeared and put away in web some portion of the data framework. Design of the web application is portrayed as a client-server three-level configuration in which the graphical client interface (introduction layer), application capacities and rationale (application layer) and PC information stockpiling (database layer) are produced and kept up as free modules, on separate stages. This type of implementation is developed by the manufacturers who are in need of monitoring and tracking data (Krešimir Grgić, Ivan Špeh, Ivan Heđi, 2016).

The availability convention for IoT and Machine-to-Machine (M2M) applications is the Message Queuing Telemetry Transport (MQTT). This lightweight convention is intended to interface the physical gadgets and systems with applications and middleware utilized in IT and Web advancement. In contrast to the CoAP, the MQTT is running over TCP. Furthermore, this convention gives three dimensions of Quality of Service (QoS) bolster for message conveyance (Gordana Gardasˇevic´, Mladen Veletic, Nebojsˇa Maletic´, Dragan Vasiljevic´, Igor Radusinovic´, Slavica Tomovic´, Milutin Radonjic´, 2016).

e-Agriculture Application has been created dependent on the system comprising of KM-Knowledge base and Monitoring modules. To settle on product choices, agriculturists need data all through the whole cultivating cycle. The required data is dissipated in different spots which incorporates continuous data, for example, showcase costs and current creation level details alongside the accessible essential harvest learning. A learning dataflow display is developed interfacing different dissipated sources to the yield structures (Mohanraj I, Kirthika Ashokumar, Naren J, 2016).

The integration of cloud computing and the agriculture can result in precision agriculture where the sensors are used for collecting the data from the fields, and the received data are then transmitted to cloud for storage which could be used for the analysis purposes. Its application includes precision irrigation, Agriculture information transmission, and Intelligent Detection and the Safety of agriculture product (Yifan Bo, Haiyan Wang, 2011).

PROS:

- The use of cloud for storing the sensor data.
- The use of MQTT protocol for the M2M communication.
- The use of camera module for the monitoring the plant growth.
- The analysis of sensor data for the precision farming.
- The use of the decision support system for the optimizing the water need of the plant.

CONS:

- Not all the features have been implemented on a single application.
- All the papers discuss either about the manual or about the automation of the agricultural process but not both.

2.3.4 GROUP 4: (Real time implementation examples)

TITLE	AUTHOR	YEAR
Intelligent Agriculture Greenhouse Environment Monitoring System Based on IOT Technology	Liu Dan, Cao Xin, Huang Chongwei, Ji Liangliang	2015
An IoT service-oriented system for agriculture monitoring	Carlos Cambra, Sandra Sendra, Jaime Lloret, Laura Garcia	2017
A precision agriculture management system based on Internet of Things and WebGIS	Jiuyan Ye, Bin Chen, Qingfeng Liu, Yu Fang	2013
IOT Based Monitoring System in Smart Agriculture	S. R. Prathibha , Anupama Hongal , M. P. Jyothi	2017

Table 2.4 List of Papers on Group 4

The CC2530 is used as the core for the implementation of the agriculture greenhouse Environment monitoring system. It is based upon the ZigBee technology for data communication. The acquired data is then sent to the intermediate node where it aggregates all the data and then sends the data to the PC through the serial port. It helps to maintain the ambient temperature for the greenhouse (Liu Dan , Cao Xin , Huang Chongwei , Ji Liangliang, 2015).

This paper proposes the idea of complete automation for plant growth based upon the illumination, temperature, moisture, and humidity as parameters. The automation process is carried out based upon the setting up of a threshold value for the parameters. This system is built with ARM7 as the microcontroller, NODE MCU 8266 as the wi-fi module for the communication. It uses LDR for calculating the illumination, DHT 11 for temperature and humidity and a soil moisture sensor (M.Krishna Mohan, Jakkula Likhitha, Tejaswy Yamarthi, Kagitha Sravani, 2017).

The low-cost irrigation controller is developed based upon the use of the real-time data such as the variable rate irrigation and some parameters acquired from the fields. The aerial images have been used for the estimation of the index vegetation. The data about the flow level, pressure level, and wind speed has been periodically sampled. Drools Guvnor has been used for processing the data in the cloud. The bandwidth consumed during the command and data communication has been calculated (Carlos Cambra, Sandra Sendra, Jaime Lloret, Laura Garcia, 2017).

Dissecting the present improvement of accuracy horticulture in China and thinking about its points of interest and deficiencies, the biology of the ranch has been picked for instance to lead another exactness farming administration framework (PAMS). PAMS was connected to the nature ranch, which is situated at Huaihua, Hunan, China. The staff can utilize the portable customer of PAMS to post the continuous information of the day-by-day work and the vital developing details of the plants. At that point, PAMS dissects the data and recommends the staff about what to do at the following stage. PAMS has helped the homestead screen decrease a great deal of time observing the staff specialist and the plant developing data. Moreover, because of the correct planting strategies, the plants grew better than anyone might have expected. PAMS is an effective method to advance the dimension of the board on the ranch (Jiuyan Ye , Bin Chen , Qingfeng Liu , Yu Fang, 2013).

CC3200 is the principal square of this proposed framework comprises of microcontroller, organize processor and Wi-Fi unit on the same kick the bucket. It is a compact, low power for battery-worked, secure and quick association. Natural conditions varieties will influence the general yield of the harvest. Plants require legitimate unmistakable conditions for ideal development and wellbeing. Observing the state of harvest field is without question significant, so sensors are utilized. Temperature infrared thermopile sensor-TMP007 is used, it has worked in advanced control and math motor. It detects the temperature esteems progressively, and dampness sensor-HDC1010 track the overall humidity of air inside the cultivating field. The camera is interfaced with the CC3200 camera sponsor pack using PCB utilizing MT9D111 camera sensor. This is used to catch current pictures of the specific field those pictures are sent to the agriculturist through GPRS. The created framework is increasingly compelling and gainful for ranchers. It gives the data about the temperature, stickiness of the air in the rural field through MMS to the agriculturist, on the off chance that it aftermath from ideal range. The framework can be utilized in greenhouse and temperature dependent plants (S. R. Prathibha, Anupama Hongal, M. P. Jyothi, 2017).

PROS:

- The use of the temperature, humidity, ph, and light intensity for monitoring the plant growth.
- The use of NODEMCU 8266 as the wi-fi module.
- The use of threshold value for determining and maintaining the favorable conditions for the automatic plant monitoring and growth.

CONS:

- None of the papers gives emphasis about the cloud computing, data storage and analysis.
- All the models are complicated and expensive to maintain.
- It requires a great deal of knowledge for the user to understand the system.

2.4 RESEARCH PROBLEM DEFINITION:

The proposed system is a hybrid system in which the following pros and cons of each individual groups are taken into account and considered for developing and maintaining the indoor plant garden.

2.4.1 GROUP 1 CONSIDERATIONS:

PROS:

- It provides the layer wise approach for the development of application.
- It emphasis the need for the least energy consumption of the iot devices.

CONS:

- Not all the layers are required for developing the application.

2.4.2 GROUP 2 CONSIDERATIONS:

PROS:

- The monitoring of the soil conditions for maintaining the plant growth.

- The determination of the amount of water required for the plant based upon its condition.

CONS:

- Not all the factors that are needed for the plant growth has been considered.

2.4.3 GROUP 3 CONSIDERATIONS:

PROS:

- The use of cloud for storing the sensor data.
- The use of MQTT protocol for the M2M communication.
- The analysis of sensor data for the precision farming.

CONS:

- Not all the features have been implemented on a single application.
- All the papers discuss either about the manual or about the automation of the agricultural process but not both.

2.4.4 GROUP 4 CONSIDERATIONS:

PROS:

- The use of the temperature, humidity, PH, and light intensity for monitoring the plant growth.
- The use of NODEMCU 8266 as the Wi-Fi module.
- The use of threshold value for determining and maintaining the favorable conditions for the automatic plant monitoring and growth.

CONS:

- None of the papers gives emphasis about the cloud computing, data storage and analysis.
- All the models are complicated and expensive to maintain.

2.5 HYPOTHESIS:

The following hypothesis are being formulated as the necessities for the precision indoor plant growth.

HYPOTHESIS 1: The healthy plant growth can only be achieved if the plants are monitored and fertilized at proper intervals.

HYPOTHESIS 2: The remote communication and the automation of events can create a huge impact in the indoor plant growth.

HYPOTHESIS 3: The creation of plant dataset may provide valuable knowledge about the indoor plant growth.

CHAPTER 3

METHODOLOGY

This section will describe about the methodology that has been adopted to implement the smart environment home garden. This chapter will cover about the detailed explanation of the methodology that was carried out for the implementation of the project. Most of the methodology was referred from the journals and findings are done based upon the problem hypothesis.

3.1 OVERVIEW:

In this phase, all relevant information needed for the development of the project are collected. The core understanding of the technologies used in this project development and the collection of the background knowledge are done this phase.

3.2 SYSTEM ARCHITECTURE:

The 3-layered architectural framework in IOT provides the efficient and effective functionalities for the system. This leads to the developed framework of the system. Each layer performs some set of tasks, which are interrelated. The 3-layers are perception layer, transport layer, and application layer.

- PERCEPTION LAYER: The first layer called perception layer, which also known as the sensing layer. This layer is responsible for the integration of hardware and collection of data from the sensors. It supports the sensing technologies like WSN (Wireless Sensor Network) and RFID.

- TRANSPORT LAYER: The second layer called network layer, which implements the support services for securing the data transfer over the sensor networks. It uses the network technologies like 3G, Wi-Fi, Bluetooth, etc. It handles the functionalities like data aggregation and addressing schemes.

- APPLICATION LAYER: The last layer called application layer, which is responsible for the delivery of services. It also provides the intelligent tools that are necessary for processing the information.

3.3 SYSTEM VISUALIZATION:

The system visualization has been explained with the help of the following diagrams.

3.3.1 CLASS DIAGRAM:

The class diagram is a static diagram that helps to represents the static view of an application.

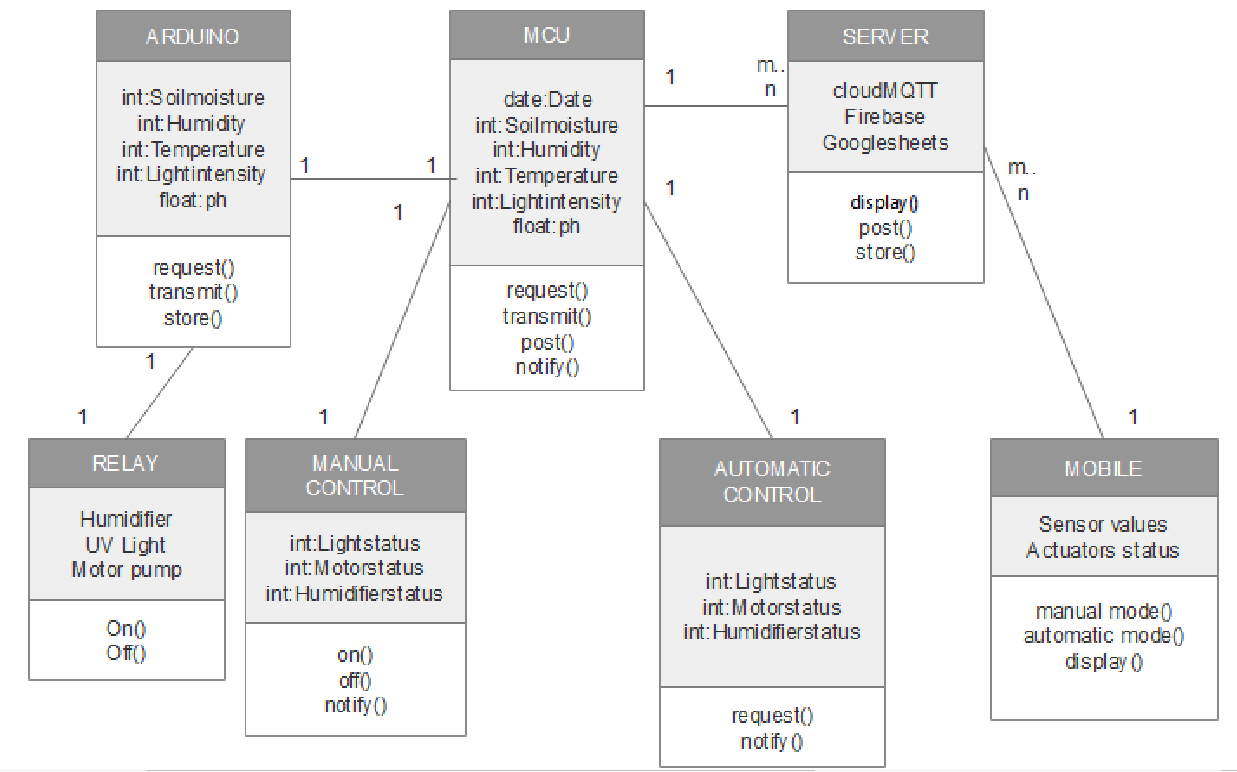


Fig 3.1 Class Diagram of the system

The above is the class diagram, which depicts the relationship between different class and their relationship between them.

Arduino

The class Arduino Uno has 5 objects which have a one to one relationship with the class MCU. It also has 3 methods which are used for requesting the data from the sensor, transmitting the data from the sensor to the Node MCU and storing the sensor data in the Arduino SD card module.

NodeMCU

The class NodeMCU has 6 objects which have one to one relationship with the class server. It also has 4 methods which 4 methods. The method "request" is to requesting the data from the Arduino Uno and the method "transmit" is used for transmitting the data to the server. The method "notify" is used for sending the notification messages through pushing box. The method "post" is used for posting the sensor data to the cloud using MQTT as the messaging protocol. The NodeMCU has one to many relationship with the class server.

Server

The class Server has 3 objects. The object "cloudMQTT" is used for the publishing and subscribing the sensor data. The object "google sheets" is used for storing the sensor data.

It has a many to one relationship with the class mobile. The method "display" is used for displaying the message in the android application. The method "post" contains the information that are needed to be posted to the android application.

Mobile

The class mobile has an one to many relationship with the class server with the sensor values and the actuator status as its objects. It consist of three methods namely "manual mode", "automatic mode" and "display".

Manual mode

The class manual mode has an one to one relationship with the class mcu. This class consist of 3 objects lightstatus, motorstatus and humidifierstatus. These objects are dependable on 3 methods namely "on", "off" and "notify".

Automatic mode

The class automatic mode has an one to one relationship with the class mcu. This class consist of 3 objects lightstatus, motorstatus and humidifierstatus. These objects are dependable on 2 methods namely "request" and "notify".

Relay

The class relay has an one to one relationship with the class arduino. This class consist of 3 objects humidifier, uvlight and motor pump. These objects are dependable on 2 methods namely "on" and "off".

3.3.2 USE CASE DIAGRAM:

Use case diagrams are used to gather the requirements of a system including internal and external influences. These requirements are mostly design requirements.

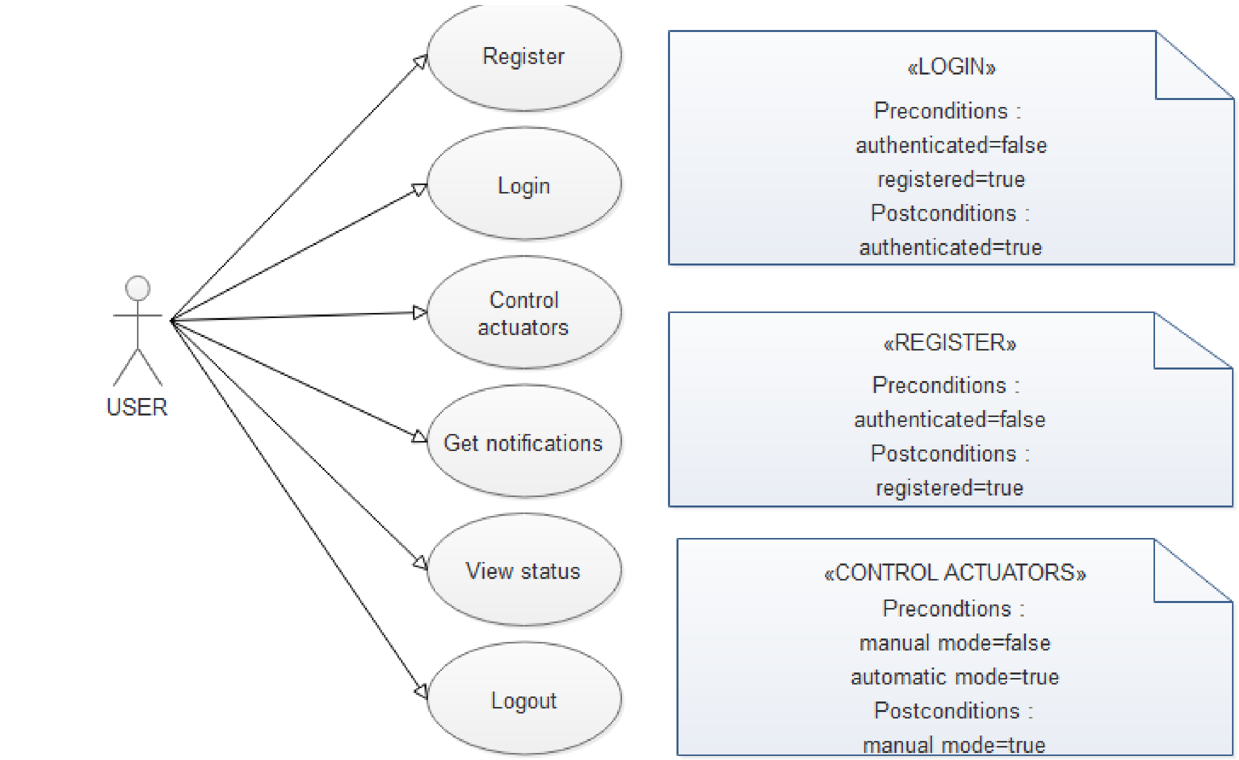


Fig 3.2 Use Case Diagram of the Mobile Application

The above diagram is a use case diagram, which represents the mobile application that is being used in the indoor plant garden. Hence, if we look into the diagram then we will find five use cases (Register, Login, Control actuators, Get notifications, View status and Logout) and one actor which is the user controlling the mobile application.

<u>Register</u>

The use case register is used for registering into the android application if the user wants to access the application for the first time. The register has a precondition in which the user should not be authenticated to access the application. The register has a post condition in which the user will be registered into the system once the registration form has been completed and submitted to the cloud.

<u>Login</u>

The use case login is used for signing in into the android application if the user wants to access the application. The login has a precondition in which the user should be registered but not to be authenticated. The login has a post condition in which the user should be registered and also authenticated. The login can be accessed by the user when the username and password matches.

<u>Control Actuators</u>

The use case control actuators is used for controlling the light, motor pump and the humidifier that are connected to the system.

Get Notifications

The use case get notifications is used for receiving the notifications from the pushbullet service. These notifications are received only when there is any change in the state of the actuator.

View Status

The use case view status has been used for monitoring the real-time sensor values from the plant. The view status displays the sensor values of the light intensity, soil moisture, air temperature, soil temperature, humidity, ph and also the state of the actuators.

Logout

The use case logout is used for signing out from the application.

3.3.3 DEPLOYMENT DIAGRAM:

Deployment diagrams are used to visualize the topology of the physical components of a system, where the software components are deployed. Deployment diagrams are used to describe the static deployment view of a system. Deployment diagrams consist of nodes and their relationships. Here, the system can is deployed based upon the automation procedures.

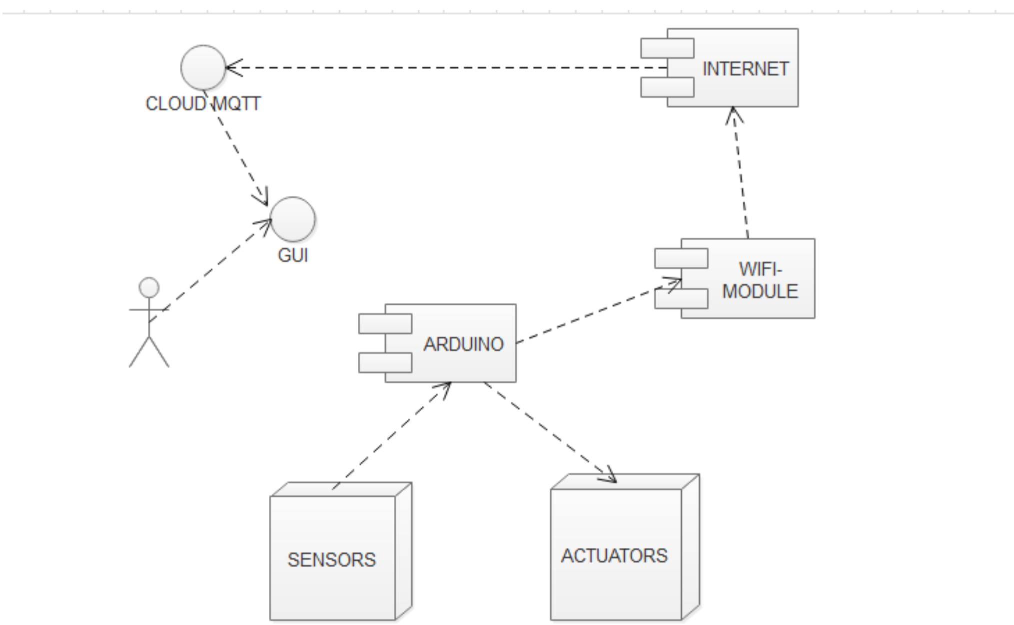


Fig 3.3 Deployment Diagram of the System

The above diagram is a deployment diagram of the system that works in automatic mode. The sensors and actuators are the nodes in the system, the Arduino, Wi-Fi module and internet acts as the components of the system and the cloudmqtt and the GUI acts as the interface of the system.

3.3.4 SEQUENCE DIAGRAM:

The diagram is used to describe some type of interactions among the different elements in the model. This interaction is a part of dynamic behavior of the system. Sequence diagram emphasizes on time sequence of messages.

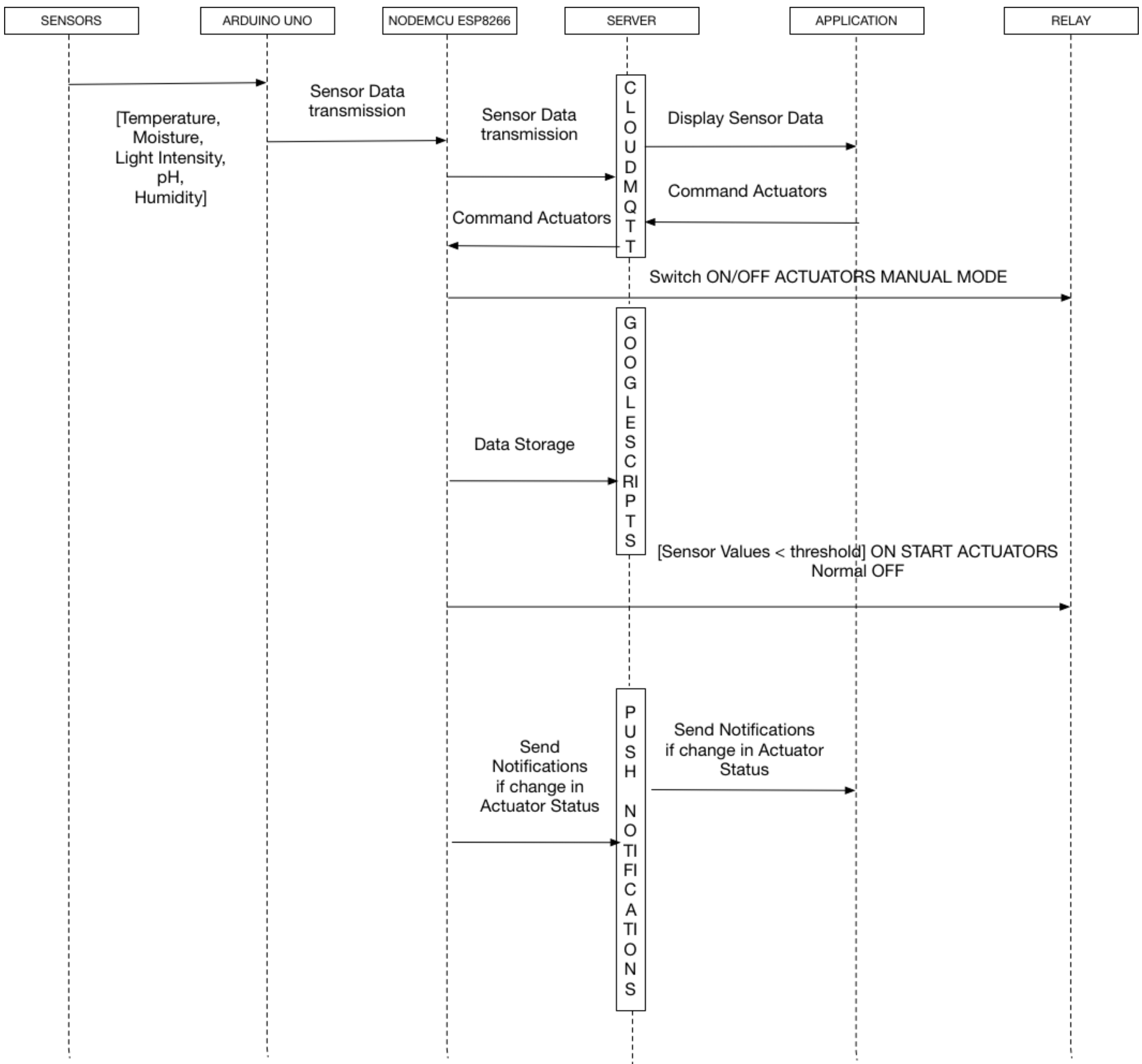


Fig 3.4 Sequence Diagram of the System

Sensors

The individual sensor senses the data and sends them to the Arduino Uno.

Arduino Uno

The Arduino Uno transmits the sensor data to the Node MCU8266 through serial communication. It also triggers the relay automatically if the sensor value is less than the defined threshold value.
It also stores the collected sensor data to the sd card module.

Node MCU8266

Two Node MCU8266 has been used in this system. The first one acts as the transmitter and the second one acts as the receiver. The Node MCU8266 which acts as the transmitter published the sensor data as an instance to the cloud. It also stores the value to the google sheets. The Node MCU8266 which acts as the receiver subscribes to the topic which are commands for activating or deactivating the actuators.

Server

This element in the sequence diagram is a combination of three services namely- cloud mqtt, google script and the pushbullet. The cloud mqtt is responsible for storing the sensor data as an instance which could be accessed by the android application. The google script is responsible for storing the sensor data and the pushbullet service is responsible for sending the notifications to the android application.

Android application

It is used for displaying the current reading of the sensor data and also to control the actuators in the manual mode.

Relay

The relay is used for activating or deactivating the actuators beased upon the commands form the Node MCU866.

3.3.5 COMPONENT DIAGRAM:

Component diagrams are different in terms of nature and behavior. Component diagrams are used to model the physical aspects of a system. Physical aspects are the elements such as executables, libraries, files, documents, etc. which reside in a node.

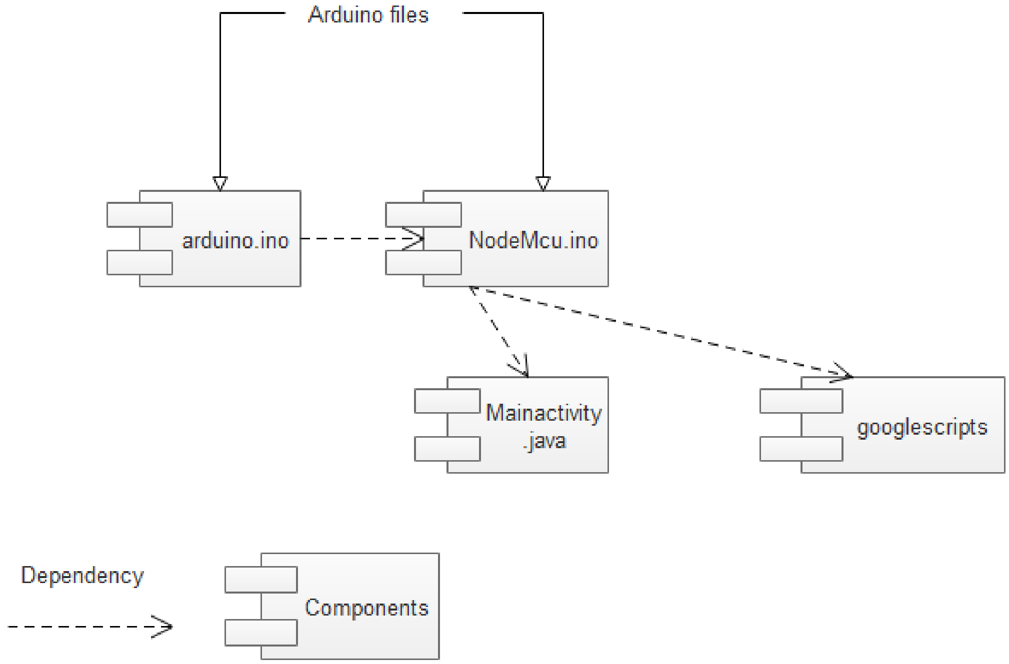


Fig 3.5 Component Diagram of the System

Arduino.ino

It contains the necessary coding required for reading the sensor data from different sensors and also to transmit the data to the Node MCU8266.

NodeMcu.ino

It contains the coding for both the transmitting the sensor data to the cloud and receiving the commands from the cloud. It depends upon the file “Arduino.ino”.

Mainactivity.java

It contains the necessary coding for the android application for displaying the current reading of the sensor data and also to activate the actuators in the manual mode. It depends upon the file “NodeMcu.ino”.

Googlescripts

It contains the necessary coding for fetching the sensor data from the cloudmqtt and store them in the google sheets. It depends upon the file “NodeMcu.ino”.

CHAPTER 4

PROTOTYPE

In this chapter the experimental setup of the project and design aspect of the independent modules are considered.

4.1 SYSTEM DESIGN:

The system design of the indoor plant garden has been divided into 3 modules. They are

- Data collection
- Data transfer
- Data storage

Data Collection:

The data collection phase is responsible for the collecting the data from the sensors. The data collected from various sensors like DS18B20, PH Sensor, Soil moisture Sensor, LDR, DHT22. These sensors are interfaced to the Arduino Uno board and the Arduino Uno board is interfaced with the NodeMCU ESP8266 board.

Data Transfer:

Embedded Systems ought to be associated with one another to exchange information between them since not all activities can be performed on a single framework. So two are more microcontrollers or microchips can be consolidated to shape a required embedded structure. The association can be built up through any of the accompanying protocols UART, I2C, SPI. This phase can be divided into three stages of information transfer. They are

- Arduino Uno to NodeMCU ESP8266: In this phase, the data are transferred from the Arduino Uno to NodeMCU ESP8266 by I2C protocol where the two wires, or lines are called Serial Clock (or SCL) and Serial Data (or SDA). The SCL line is the clock signal, which synchronize the data transfer between the devices on the I2C bus, and the master device generates it. The other line is the SDA line, which carries the data. The A4 pin of Arduino Uno acts as the SDA and the A5 pin of the Arduino Uno acts as the SCL pin. The D1 pin of the NodeMCU ESP8266 acts as the SDA and the D2 pin of the NodeMCU ESP8266 acts as the SCL. The Wire Library is used for implementing the I2C communication on the boards.

CODE SNIPPET:

Arduino Uno

```
#include <Wire.h>
// function that executes whenever data is requested from
master
void requestEvent() {
/*send string on request */
 uint8_t Buffer[6];
 Buffer[0]=table[0];
 Buffer[1]=table[1];
 Buffer[2]=table[2];
 Buffer[3]=table[3];
 Buffer[4]=table[4];
 Buffer[5]=table[5];
 Wire.write(Buffer,6);
}
```

Node MCU:

```
#include <Wire.h>
int table[]={0,0,0,0,0,0}
Wire.requestFrom(8, 6);
while(Wire.available()){
for(int i=0;i<6;i++)//organizes the data from the slave in
the table
{
int c = Wire.read(); // receive a byte as character
table[i]=c;
 Serial.print(table[i]);
}
}
//displays the data
 Serial.print('\n');
 Serial.print(table[0]);
 Serial.print('\t');
 Serial.print(table[1]);
 Serial.print('\t');
 Serial.print(table[2]);
 Serial.print('\n');
 Serial.print(table[3]);
 Serial.print('\n');
 Serial.print(table[4]);
 Serial.print('\n');
 Serial.print(table[5]);
 Serial.print('\n');
 Serial.print(table[6]);
 Serial.print('\n');
```

- NodeMCU ESP8266 to Cloud: In this phase, the data are transferred from the NodeMCU ESP8266 to Cloud by MQTT protocol. Here Cloud MQTT is used as the cloud messaging protocol for transferring the data.

CODE SNIPPET:

```
#include <PubSubClient.h>
const char *mqtt_server="m16.cloudmqtt.com";
const char *mqtt_user= "yrzmecpp";
const char *mqtt_pass="5UbSz2Rz4gVk";
client.publish("Soil Moisture",cstr);
client.publish("Light Intensity",cssttr);
client.publish("Temperature(soil)",csssttttr);
client.publish("Temperature(air)",tstr);
client.publish("Humidity",rstr);
client.publish("PH Value",bstr);
```

- Cloud to NodeMCU ESP8266: In this phase, the data are transferred from the cloud to NodeMCU ESP8266 by MQTT protocol. Here Cloud MQTT is used as the cloud messaging protocol for transferring the data.

CODE SNIPPET:

```
#include <PubSubClient.h>
WiFiClient espClient;
PubSubClient client(espClient);
char message_buff[100];

void callback(char *led_control, byte *payload, unsigned int
length)
{
Serial.print("Message arrived [");
Serial.print(led_control);
Serial.println("] ");
int i;
for (i = 0; i <length; i++)
{
message_buff[i] = payload[i];
}
message_buff[i] = '\0';
String msgString = String(message_buff);
Serial.println("THE      MESSAGE      FROM      THE      CLOUD
IS..................");
Serial.println(msgString);
if (strcmp(led_control, "command") == 0)
{
if (msgString == "son")
```

```
{
Serial.println("THE     MESSAGE     FROM     THE     CLOUD
IS..................");
Serial.println(msgString);
sendNotification("The   Pump   is   switched   ON!!!!!!!   for
irrigation");
digitalWrite(D2,LOW);
}
if (msgString == "soff")
{
Serial.println("THE     MESSAGE     FROM     THE     CLOUD
IS..................");
Serial.println(msgString);
digitalWrite(D2,HIGH);
}
}
if (strcmp(led_control, "command") == 0)
{
 if (msgString == "lon")
 {
 Serial.println("THE     MESSAGE     FROM     THE     CLOUD
IS..................");
 Serial.println(msgString);
 sendNotification("The Light is switched ON!!!!!!!");
 digitalWrite(D3,LOW);
 }
 if (msgString == "loff")
{
Serial.println("THE     MESSAGE     FROM     THE     CLOUD
IS..................");      Serial.println(msgString);
digitalWrite(D3,HIGH);
}
 }
if (strcmp(led_control, "command") == 0)
{
if (msgString == "hon")
{
Serial.println("THE     MESSAGE     FROM     THE     CLOUD
IS..................");
Serial.println(msgString);
sendNotification("The Humidifier is switched ON!!!!!!!");
digitalWrite(D4,LOW);
}
if (msgString == "hoff")
{
```

```
Serial.println("THE MESSAGE FROM THE CLOUD IS..................");
Serial.println(msgString);
digitalWrite(D4,HIGH);
  }
  }
```

- Cloud to Application services: In this phase, the data are transferred from the Cloud to various applications. Cloud MQTT is used the cloud service for transferring the data to google sheets and the android application. It is also used as the cloud service for controlling the actuators through the android application.

CODE SNIPPET:

```
importorg.eclipse.paho.android.service.MqttAndroidClient;
importorg.eclipse.paho.client.mqttv3.IMqttActionListener;
importorg.eclipse.paho.client.mqttv3.IMqttDeliveryToken;
importorg.eclipse.paho.client.mqttv3.IMqttToken;
importorg.eclipse.paho.client.mqttv3.MqttCallback;
importorg.eclipse.paho.client.mqttv3.MqttClient;
importorg.eclipse.paho.client.mqttv3.MqttConnectOptions;
importorg.eclipse.paho.client.mqttv3.MqttException;
import org.eclipse.paho.client.mqttv3.MqttMessage;

public void connect(){

    String clientId = MqttClient.generateClientId();
    final MqttAndroidClient client =
            new
MqttAndroidClient(this.getApplicationContext(),
"tcp://m16.cloudmqtt.com:13330",
                    clientId);

    MqttConnectOptions options = new MqttConnectOptions();

options.setMqttVersion(MqttConnectOptions.MQTT_VERSION_3_1)
;
    options.setCleanSession(false);
    options.setUserName("yrzmecpp");
    options.setPassword("5UbSz2Rz4gVk".toCharArray());
    try {
        IMqttToken token = client.connect(options);
        //IMqttToken token = client.connect();
        token.setActionCallback(new IMqttActionListener() {
            @Override
            public void onSuccess(IMqttToken
```

```
asyncActionToken) {
                // We are connected
                Log.d("file", "onSuccess");
                //publish(client,"payloadd");
                subscribe(client,"Soil Moisture");
                subscribe(client,"Light Intensity");
                subscribe(client,"Temperature(soil)");
                subscribe(client,"Light State");
                subscribe(client,"Pump State");
                subscribe(client,"Temperature(air)");
                subscribe(client,"Humidity");
                subscribe(client,"PH Value");
                client.setCallback(new MqttCallback() {
                    EditText tt = (EditText)
findViewById(R.id.editText10);
                    EditText th = (EditText)
findViewById(R.id.editText14);
                    EditText ti = (EditText)
findViewById(R.id.editText12);
                    EditText
a=(EditText)findViewById(R.id.editText15);
                    EditText
b=(EditText)findViewById(R.id.editText16);
                    EditText
c=(EditText)findViewById(R.id.editText11);
                    EditText
d=(EditText)findViewById(R.id.editText13);
                    EditText
e=(EditText)findViewById(R.id.editText9);
                    @Override
                    public void connectionLost(Throwable
cause) {

                    }
                    @Override
                    public void messageArrived(String
topic, MqttMessage message) throws Exception {
                        Log.d("file", message.toString());

                        if (topic.equals("Soil Moisture")){
                            tt.setText(message.toString());
                        }

                        if (topic.equals("Light
Intensity")){
                            ti.setText(message.toString());
```

```
                        }

                        if
(topic.equals("Temperature(soil)")){
                            th.setText(message.toString());
                        }

                        if (topic.equals("Light State")){
                            b.setText(message.toString());
                        }


                        if (topic.equals("Pump State")){
                            a.setText(message.toString());
                        }

                        if
(topic.equals("Temperature(air)")){
                            c.setText(message.toString());
                        }
                        if (topic.equals("PH Value")){
                            d.setText(message.toString());
                        }
                        if (topic.equals("Humidity")){
                            e.setText(message.toString());
                        }


                    }
                    @Override
                    public void
deliveryComplete(IMqttDeliveryToken token) {

                    }
                });
            }

            @Override
            public void onFailure(IMqttToken
asyncActionToken, Throwable exception) {
                // Something went wrong e.g. connection
timeout or firewall problems
                Log.d("file", "onFailure");

            }
        });
```

```
    } catch (MqttException e) {
        e.printStackTrace();
    }
}


public void publish(MqttAndroidClient client, String
payload)
{
    String topic = "foo/bar";
    byte[] encodedPayload = new byte[0];
    try {
        encodedPayload = payload.getBytes("UTF-8");
        MqttMessage message = new
MqttMessage(encodedPayload);
        client.publish(topic, message);
    } catch (UnsupportedEncodingException | MqttException
e) {
        e.printStackTrace();
    }
}


public void subscribe(MqttAndroidClient client , String
topic)
{
    int qos = 1;
    try {
        IMqttToken subToken = client.subscribe(topic, qos);
        subToken.setActionCallback(new
IMqttActionListener() {

            @Override
            public void onSuccess(IMqttToken
asyncActionToken) {
                // The message was published
            }

            @Override
            public void onFailure(IMqttToken
asyncActionToken,
                                  Throwable exception) {
                // The subscription could not be performed,
maybe the user was not
                // authorized to subscribe on the specified
topic e.g. using wildcards
```

```
            }
        });
    } catch (MqttException e) {
        e.printStackTrace();
    }
}
```

Data Storage:

In this phase, the data collected from the sensors are stored by using two approaches. They are

- Google Sheets: In the first approach, the data from the Arduino Uno is sent to the NodeMCU ESP8266. Then the data from the NodeMCU ESP8266 are sent to the google sheets and are stored in it. The google sheet is programmed by using the google script language.

CODE SNIPPET: (NODE MCU – TRANSMITTER)

```
const char WEBSITE[] = "api.pushingbox.com"; //pushingbox API server

const String devid = "vE37D6294FF6DD46"; //device ID from Pushingbox

WiFiClient client;  //Instantiate WiFi object
//Start or API service using our WiFi Client through PushingBox
    if (client.connect(WEBSITE, 80))
{
 client.print("GET /pushingbox?devid=" + devid + "&HumidityData=" + (String) HumidityData + "&TemperatureData=" + (String) TemperatureData + "&LightintensityData=" + (String) LightintensityData + "&SoilmoistureData=" + (String) SoilmoistureData + "&phData=" + (String) phData );
client.println(" HTTP/1.1");
client.print("Host: ");
        client.println(WEBSITE);
        client.println("User-Agent: ESP8266/1.0");
        client.println("Connection: close");
        client.println();
       }
       }
```

GOOGLE SCRIPT

```
function doGet(e) {
  Logger.log( JSON.stringify(e) );  // view parameters

  var result = 'Ok'; // assume success

  if (e.parameter == undefined) {
    result = 'No Parameters';
  }
  else {
    var id = '1Zr-BBLWWTU8U3KEru41RBdDROkhRSbVZH914CAtW-
I4';//docs.google.com/spreadsheetURL/d
    var sheet =
SpreadsheetApp.openById(id).getActiveSheet();
    var newRow = sheet.getLastRow() + 1;
    var rowData = [];
    //var waktu = new Date();
    rowData[0] = new Date(); // Timestamp in column A

    for (var param in e.parameter) {
      Logger.log('In for loop, param='+param);
      var value = stripQuotes(e.parameter[param]);
      //Logger.log(param + ':' + e.parameter[param]);
      switch (param) {
        case 'HumidityData': //Parameter
          rowData[1] = value; //Value in column B
          break;
        case 'TemperatureData':
          rowData[2] = value;
          break;
        case 'LightintensityData':
          rowData[3] = value;
          break;
        case 'SoilmoistureData':
          rowData[4] = value;
          break;
        case 'phData':
          rowData[5] = value;
          break;
        default:
          result = "unsupported parameter";
      }
    }
    Logger.log(JSON.stringify(rowData));

    // Write new row below
```

```
    var newRange = sheet.getRange(newRow, 1, 1,
rowData.length);
    newRange.setValues([rowData]);
  }

  // Return result of operation
  return ContentService.createTextOutput(result);
}
function stripQuotes( value ) {
  return value.replace(/^["']|['"]$/g, "");
}
```

- SD card: In the second approach, the data from the Arduino Uno are directly stored in the SD card module which can be further used for various analysis.

CODE SNIPPET:

```
// Create/Open file
  myFile = SD.open("log.txt", FILE_WRITE);
  // if the file opened okay, write to it
if (myFile) {
Serial.println("Writing to file...");
    // Write to file
   myFile.println('<');
   myFile.print(aa);
   myFile.print(',');
   myFile.print(bb);
   myFile.print(',');
   myFile.print(Celcius);
   myFile.print(',');
   myFile.print(dhthum);
   myFile.print(',');
   myFile.print(dhttemp);
   myFile.print(',');
   myFile.print(phValue);
   myFile.print('>');
   myFile.close(); // close the file
   Serial.println("Done.");
  }
  // if the file didn't open, print an error:
  else {
    Serial.println("error opening log.txt");
  }
  // Reading the file
  myFile = SD.open("log.txt");
  if (myFile) {
```

```
    Serial.println("Read:");
    // Reading the whole file
    while (myFile.available()) {
      Serial.write(myFile.read());
   }
    myFile.close();
  }
  else {
    Serial.println("error opening log.txt");
  }
```

4.2 HARDWARE AND SOFTWARE REQUIREMENT:

4.2.1 Hardware Requirements:

- Arduino Uno
- Node MCU 8266
- Arduino sdcard module
- DC 5V Relay
- Pump 5V
- DS18B20
- SOIL MOISTURE
- DHT 22
- Resistor 4.7 K ohm
- PH Sensor
- Full Spectrum A19 LED Grow
- LDR
- Jumper Cable
- Ultrasonic mist generator

4.2.2 Software Requirements:

- Arduino
- Android Studio

4.3 EXPERIMENTAL SETUP:

The tomato plant has been chosen for implementing the smart indoor plant garden. The threshold values for the sensor and actuators are described based upon the conditions that are favourabel for the tomato plant. Block diagram is shown in figure below

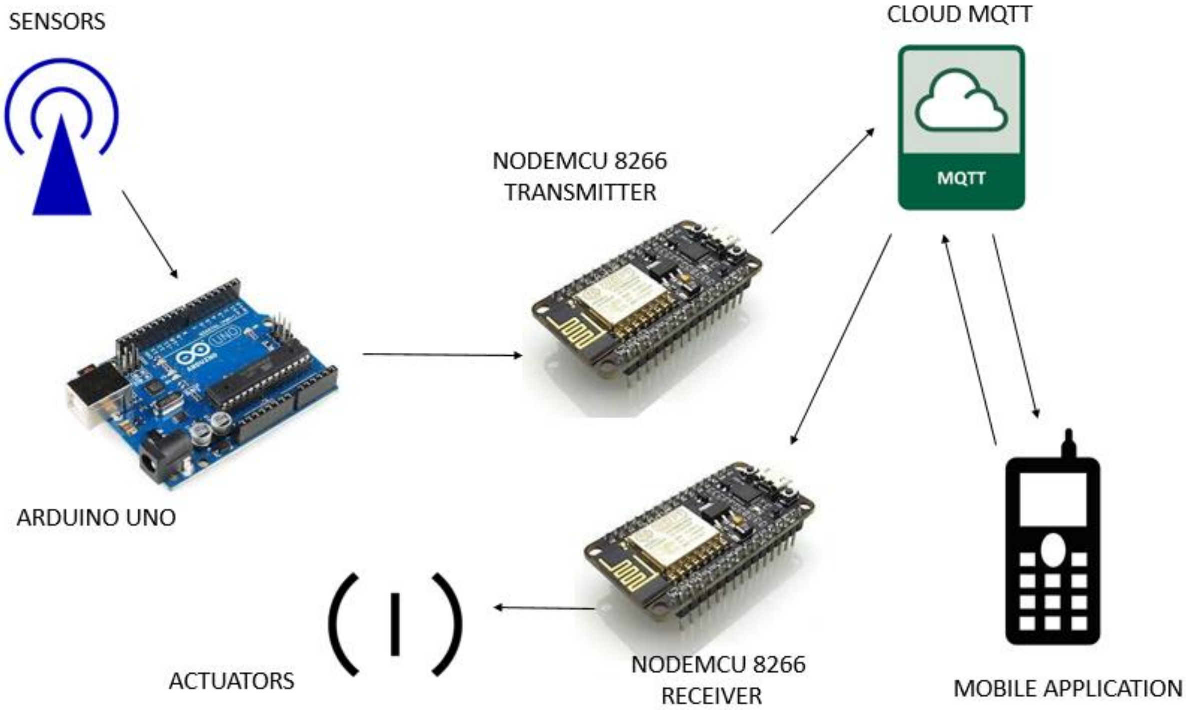


Fig 4.1 Block Diagram of the System

The main blocks of this project are:

- NodeMCU 8266
- Arduino Uno
- Sensors
- Actuators
- Cloud Services
- Mobile application

NodeMCU 8266:

It is used for the transmission of the sensor data from the Arduino to cloud via wi-fi. It acts as the master in the I2C communication with the Arduino uno.

Arduino Uno:

It is used for the interfacing of various sensors through its analog and digital pins. It acts as the slave in the I2C communication with the Node MCU 8266.

Sensors:

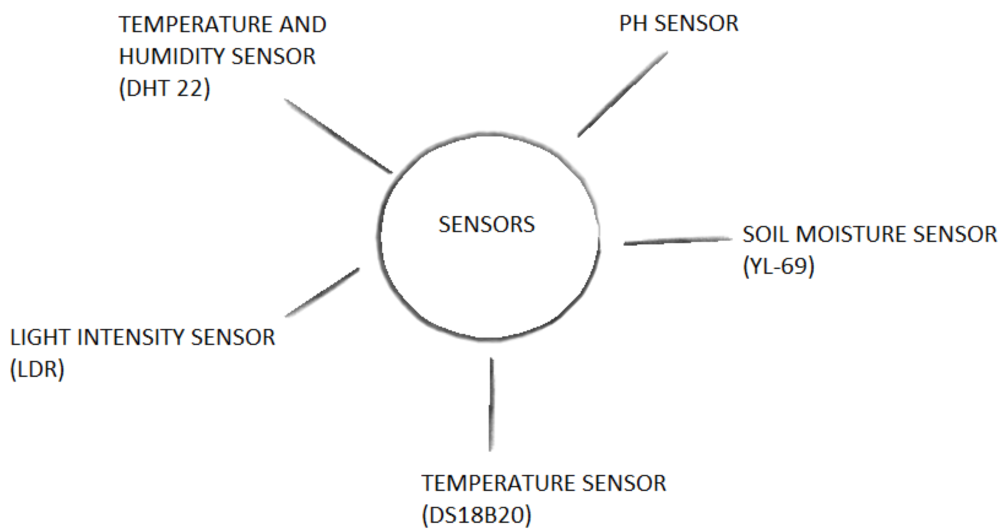


Fig 4.2 List of Sensors

- **Temperature and Humidity sensor:**

This sensor is more precise, more accurate and works in a bigger range of temperature/humidity.

HARDWARE CONNECTIONS

ARDUINO UNO	DHT22
GND	GND
5V	VCC
Analog 3 of Arduino	DATA

Table 4.1 Hardware interfacing between Arduino Uno and DHT22

RELEVANCE TO PROJECT

Humidity is important to make photosynthesis possible. Humidity levels also play a significant role in determining successful tomato growth.

- **PH sensor:**

It is used for measuring the pH of the soil, which in turn helps in the process of fertilizing the plant. It is an important parameter while growing the plants in the indoor plant garden.

HARDWARE CONNECTIONS

ARDUINO UNO	PHSENSOR
GND	GND
5V	VCC
A2	DATA

Table 4.2 Hardware interfacing between Arduino Uno and PH Sensor

RELEVANCE TO THE PROJECT

A pH assurance (soil test) will tell whether our soil will create great plant development or whether it should be provided with the acidic or basic solution to receive alter the pH level. For tomato plants, the ideal pH for growth is from 5.5 to 7.0. However, a few plants will develop in progressively corrosive soil or may require an increasingly antacid dimension.

- **Light Intensity sensor:**

An LDR is a component that has a (variable) resistance that changes with the light intensity that falls upon it. This allows them to be used in light sensing circuits.

HARDWARE CONNECTIONS

ARDUINO UNO	LDR
GND	GND
5V	VCC
A2	DATA

Table 4.3 Hardware interfacing between Arduino Uno and LDR

RELEVANCE TO THE PROJECT

It is used for the monitoring of the light intensity around the tomato plant and it helps to activate the actuators when the light intensity goes below the threshold value needed for the plant.

- **Soil moisture sensor:**

The Moisture sensor is utilized to gauge the water content (moisture) of soil. When the soil has a water shortage, the module yield is at an abnormal state; else, the yield is at a low level. This sensor reminds the client to water their plants and furthermore screens the dampness substance of soil.

HARDWARE CONNECTIONS

ARDUINO UNO	SOIL MOISTURE SENSOR
GND	GND
5V	VCC

Analog 0 of Arduino	A0

Table 4.4 Hardware interfacing between Arduino Uno and Soil Moisture Sensor

RELEVANCE TO THE PROJECT

Since soil, moisture is important for the healthy and the nutritious plant it is important to monitor them regularly. The tomato plant requires minimum of 70% soil moisture. Thus, soil moisture is used to monitor the plant regularly and activate the water pump as actuator when it becomes less than the threshold value.

- **Temperature sensor:**

The DS18B20 Digital Thermometer gives 9 to 12-bit (configurable) temperature readings, which show the temperature of the gadget. It conveys over a 1-Wire transport that by definition requires just a single information line (and ground) for correspondence with a central microprocessor.

HARDWARE CONNECTIONS

ARDUINO UNO	DS18B20
GND	GND
Arduino 5V, plus a 4.7K resistor going from VCC to Data	VCC
7 Arduino pin	Data

Table 4.5 Hardware interfacing between Arduino Uno and DS18B20

RELEVANCE TO THE PROJECT

It is used for measuring the temperature of the soil, which in turn is used for monitoring the optimum temperature for photosynthesis. In our project, the tomatoes get into trouble when the temperature exceeds about 96 degrees F (36 degrees C).

Actuators:

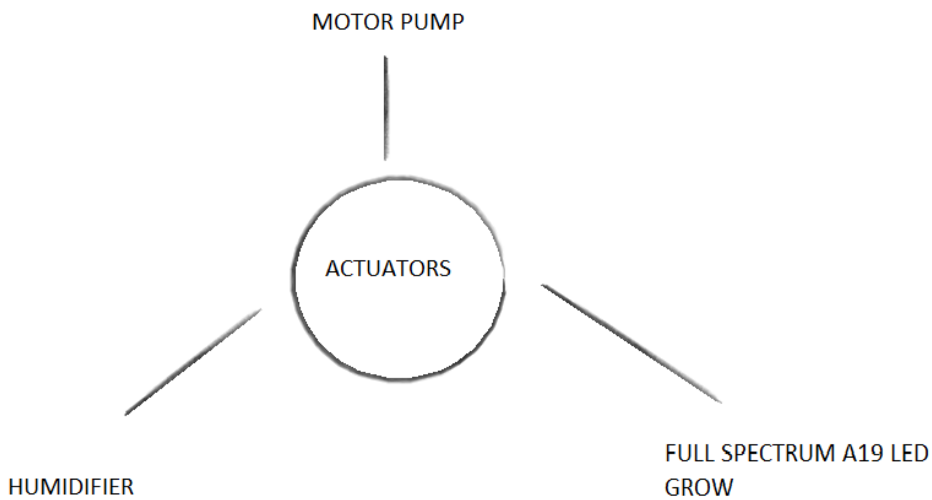


Fig 4.3 List of Actuators

- **Motor Pump:**

It is connected with the external power supply and uses 5V the relay for the controlling of the pump by Arduino Uno.

RELEVANCE TO THE PROJECT

It is used for the irrigation process of the indoor plant garden.

- **Ultrasonic mist generator:**

It is used for the creation of mist with the help of the water. It produces the water mist under that condition that the atomized pieces vibrate in high frequency.

RELEVANCE TO THE PROJECT

It is used as an actuator to produce water vapor, which in turn increases the humidity in the air surrounding the indoor plant.

- **Full Spectrum A19 LED Grow:**

Feit Electric LED Grow Light Bulb and fixtures. Our LED Grow Lights are ideal for greenhouse, horticulture, hydroponic or aquaponics planting and indoor gardening.

IMPORTANCE

It emits more light in the useful 450nm blue and 655 nm red spectrum than regular white LED bulb. Blue light promotes tall leafy plants and the red light encourages budding, flowering and fruiting. They operate cooler than HPS lamps to guard against leaf burn.

RELEVANCE TO THE PROJECT

Plants in photosynthesis use red and blue ends of the visible part of the electromagnetic spectrum. Chlorophyll-a & b are the primary pigments for photosynthesis in plants with the absorption peaks at approximately 448nm and 630nm. Healthy tomato plants need an average of 8 hours of sunlight. This can be substituted with the help of a full spectrum light.

CLOUD SERVICES:

CloudMQTT are managed Mosquitto servers in the cloud. Mosquitto implements the MQ Telemetry Transport protocol, MQTT, which provides lightweight methods of carrying out messaging using a publish/subscribe message queueing model. MQTT is the machine-to-machine convention of things to come. It is perfect for the "Internet of Things" universe of associated gadgets. Its negligible plan makes it ideal for inherent frameworks, cell phones and other memory and data transfer capacity sensitive applications. Message queues give an asynchronous communications convention, the sender and beneficiary of the message do not have to communicate with the message line in the meantime. Messages set onto the line are put away until the beneficiary recovers them or until the messages times out. MQTT and Mosquitto are for good use by data transmission sensitive applications.

Mobile Application:

The android OS has been chosen for the mobile application that controls the actuators during the manual mode of operation and displays the sensor values in real time.

4.4 WORKING:

The working of the indoor plant garden consist of four phases. Each phase performs a set of tasks which are interrelated to other phases. The output of one phase is used as the input of other phase.

PHASE1: SENSOR/ACTUATOR–DEVICE INTERACTION

In this phase, the data from the different sensors are sent to the Arduino Uno and are stored in the microsd card with the help of Arduino sdcard module. These data can be used in the future for the analytics process.

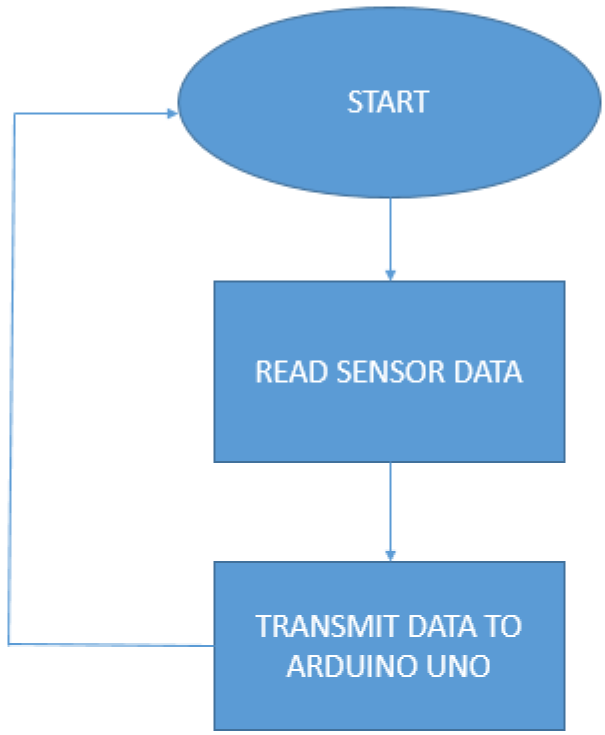


Fig 4.4 Flow Diagram Sensor-Device interaction

STEP 1: The sensor data is read from various sensors (DHT 22, Soil moisture, DS18B20, PH, and LDR)

CODE SNIPPET:

// LIBRARIES USED

```
#include <OneWire.h>
#include <DallasTemperature.h>
#include <SimpleDHT.h>
```

// FOR CALCULATING AIR TEMPERATURE AND HUMIDITY

```
SimpleDHT22 dht22;
float humidity=0;;  //Stores humidity value
float temperature=0; //Stores temperature value
int err=SimpleDHTErrSuccess;

if((err=dht22.read2(pin,&temperature,&humidity,NULL))!=SimpleDHTErrSuccess){
     Serial.println("Read DHT22 failed,err=");
     Serial.println(err);
     delay(2000);
     return;
```

```
}
 Serial.println("Sample OK: ");
 Serial.println((int)temperature);
  dhttemp=((int)temperature);//temp from dht22
  Serial.println(dhttemp);
  Serial.println((int)humidity);
  dhthum=((int)humidity);//humidity from dht22
  Serial.println(dhthum);
  Serial.println("RH%");
 delay(2500); //Delay 2 sec
```

// FOR CALCULATING THE SOIL TEMPERATURE

```
#define ONE_WIRE_BUS 7
long sensorValue =0;// reads the value of ldr
OneWire oneWire(ONE_WIRE_BUS);
DallasTemperature sensors(&oneWire);
int Celcius=0;
float Fahrenheit=0;
sensors.requestTemperatures();
Celcius=sensors.getTempCByIndex(0);
Fahrenheit=sensors.toFahrenheit(Celcius);
Serial.print(" C  ");
Serial.print(Celcius);
Serial.print(" F  ");
Serial.println(Fahrenheit);
delay(1000);
```

//FOR CALCULATING THE LIGHT INTENSITY

```
sensorValue = analogRead(A1); // read the value from the
sensor for ldr
Serial.println("LDR:");
Serial.println(sensorValue);
```

//FOR CALCULATING THE SOIL MOISTURE

```
sensorValuee = analogRead(A0);
Serial.println("soil moisture sensor :");
Serial.println(sensorValuee);
```

//FOR CALCULATING THE PH

```
const int analogInPin = A3;
unsigned long int avgValue;
int buf[10],temp;
double phValue=0.00;
```

```
for(int i=0;i<10;i++)
 {
  buf[i]=analogRead(analogInPin);
  delay(10);
 }
 for(int i=0;i<9;i++)
 {
  for(int j=i+1;j<10;j++)
  {
   if(buf[i]>buf[j])
   {
    temp=buf[i];
    buf[i]=buf[j];
    buf[j]=temp;
   }
  }
 }
 avgValue=0;
 for(int i=2;i<8;i++)
 avgValue+=buf[i];
 float pHVol=(float)avgValue*5.0/1024/6;
 phValue = -5.70 * pHVol + 21.34;
 Serial.print("sensor = ");
 Serial.println(phValue);
```

STEP 2: The sensor values of the soil moisture and light intensity are mapped to 0-100 since their values range from 0-1023.

CODE SNIPPET:

// MAPPING FOR SOIL MOISTURE

```
bb=map(sensorValuee,0,1023,0,100);
```

// MAPPING FOR LIGHT INTENSITY

```
aa=map(sensorValue,0,1023,0,100);
```

STEP 3: These data are then stored in the Arduino Uno for further processing.

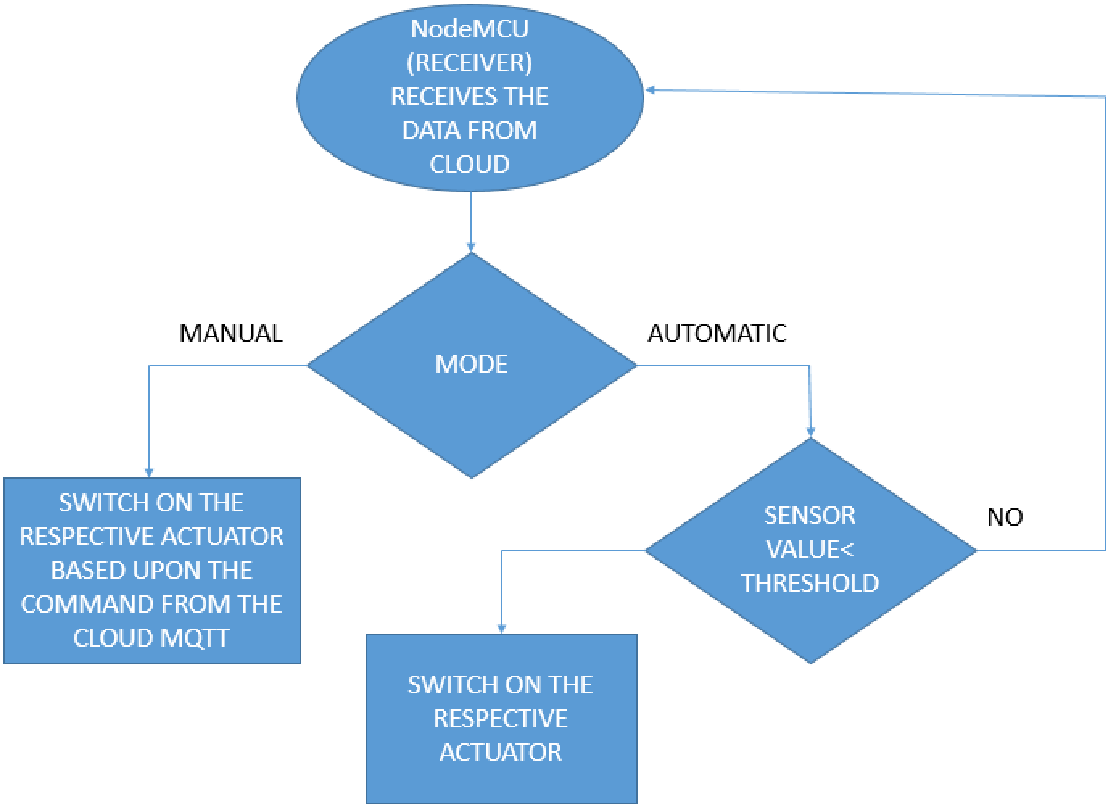


Fig 4.5 Flow Diagram Actuator-Device interaction

The NodeMCU ESP8266 (Transmitter) will check whether the sensor data < threshold value needed for the plant and transmits the command by publishing the topic to the cloudMQTT to activate or terminate the respective actuator.

PSEUDOCODE:

if (humidity < 30)
then send command "hon"
else if (humidity > 30)
then send command "hoff"
if (light intensity < 40)
then send command "lon"
if (light intensity > 40)
then send command "loff"
if (soil moisture < 50)
then send command "son"
if (soil moisture > 50)
then send command "soff"

The NodeMCU ESP8266 (Receiver) receives the data from the cloud using MQTT protocol by subscribing to a topic called “command” and the message strings.

PSEUDOCODE:

if (message string ==hon)
then humidifier is switched on
else
if (message string ==hoff)
then humidifier is switched off
else
if (message string ==lon)
then light is switched on
else
if (message string ==loff)
then light is switched off
else
if (message string ==son)
then pump is switched on
else
if (message string ==soff)
then pump is switched off

In automatic mode, the NodeMCU ESP8266 which acts as the transmitter will publish the topic to the cloudMQTT and the NodeMCU ESP8266 which acts as the receiver will subscribe to the topic named “command” and activates the respective actuator.

In manual mode, this is an overriding mode in which the user transmits the command from android application by publishing to the cloudMQTT and the NodeMCU ESP8266 which acts as the receiver will subscribe to the topic named “command” and activates the respective actuator.

PHASE2: DEVICE–DEVICE INTERACTION

In this phase, the data from the Arduino Uno are sent to the NodeMCU with the help of the serial communication. Then the data from the NodeMCU is transmitted to the cloud mqtt as an instance. This NodeMCU is used for the transmission of data to the cloud. Another NodeMCU is used for controlling the actuators based upon the threshold value for the plant growth in automatic mode and also by subscribing the commands from android application in manual mode.

STEP 1: The data obtained from the sensors are stored in the 16GB Sd Card with the help of Arduino Sd Card Module.

STEP 2: The data obtained from the sensors are stored in the Arduino in the form of a table and are transferred via serial communication.

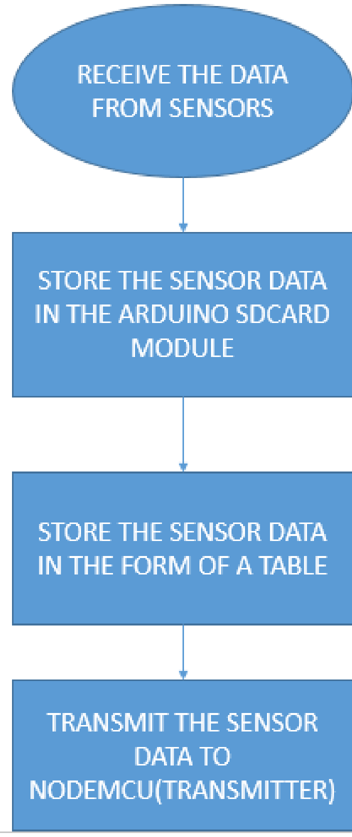


Fig 4.6 Flow Diagram Arduino Uno-Node MCU8266 interaction

PHASE3: DEVICE–CLOUD INTERACTION

In this phase, the data from the NodeMCU is transmitted to the cloud mqtt as an instance. The data are also sent to the google sheets which acts the cloud storage for the sensor data. This NodeMCU is used for the transmission of data to the cloud. The NodeMCU is also used as a receiver for subscribing the commands from the cloud mqtt.

STEP 1: The NodeMCU 8266 which acts as the transmitter receives the data from the Arduino Uno with the help of a serial communication and the data received are stored in the form of a table.

STEP 2: These data are then published in the cloud using MQTT messaging protocol as an instance.

STEP 3: These data are then stored in the google sheets with the help of the pushing box service.

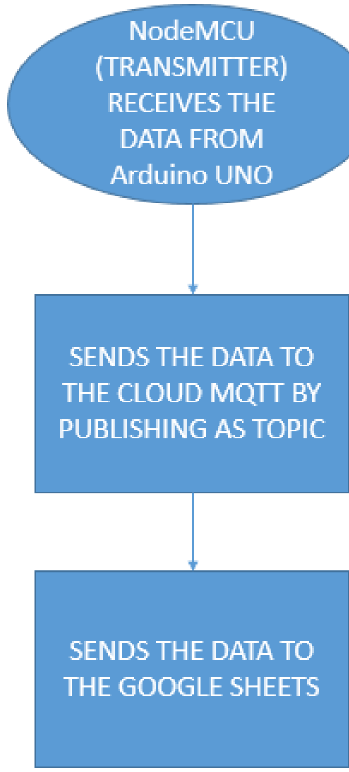


Fig 4.7 Flow Diagram Device-Cloud interaction

PHASE4: CLOUD–MOBILE INTERACTION

In this phase, the data from the cloud mqtt are displayed in real-time in the android mobile application. The android application subscribes the topic from the cloud instances which are then displayed in the application. The android application is also used for controlling the actuators by publishing the commands to the cloud as an instance.

STEP 1: The data published in the cloud using MQTT as a messaging protocol are displayed as an instance.

STEP 2: The android mobile application will then subscribe these instances and display the data of the sensors in real-time.

STEP 3: The android mobile application will also be used to control the actuators in manual mode by publishing these commands as an instance in the cloud using MQTT as a messaging protocol.

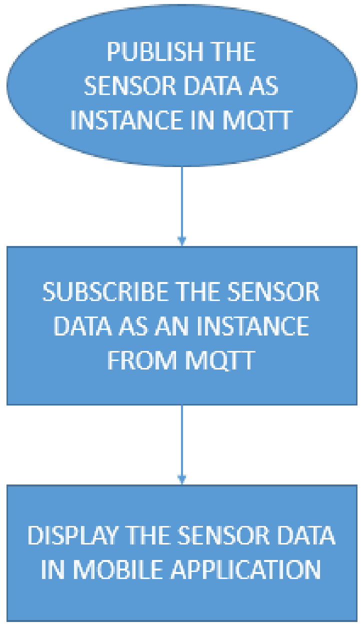


Fig 4.8 Flow Diagram Cloud-Mobile interaction

4.5 OUTPUT:

Sensor Values (Break points):

<u>Sensor value = 255:</u>

Soil Moisture = 255 {Reason: COMMUNICATION ISSUE BETWEEN NODEMCU AND MQTT}

Temperature (Air) = 255 {Reason: COMMUNICATION ISSUE BETWEEN NODEMCU AND MQTT}

Temperature (Soil) = 255 {Reason: COMMUNICATION ISSUE BETWEEN NODEMCU AND MQTT}

Humidity = 255 {Reason: COMMUNICATION ISSUE BETWEEN NODEMCU AND MQTT}

PH Value = 255 {Reason: COMMUNICATION ISSUE BETWEEN NODEMCU AND MQTT}

Light Intensity = 255 {Reason: COMMUNICATION ISSUE BETWEEN NODEMCU AND MQTT}

<u>Sensor value > Threshold:</u>

Soil Moisture = 70 {Action: Pump is switched OFF}

Humidity = 40 {Action: Humidifier is switched OFF}

Light Intensity = 55 {Action: Light is switched OFF}

Sensor value < Threshold:
Soil Moisture = 49 {Action: Pump is switched ON}
Humidity = 29 {Action: Humidifier is switched ON}
Light Intensity = 39 {Action: Light is switched ON}

Sensor value = Threshold:
Soil Moisture = 50 {Action: Pump is switched OFF}
Humidity = 30 {Action: Humidifier is switched OFF}
Light Intensity = 40 {Action: Light is switched OFF}

4.6 SCREENSHOTS:

CLOUD MQTT

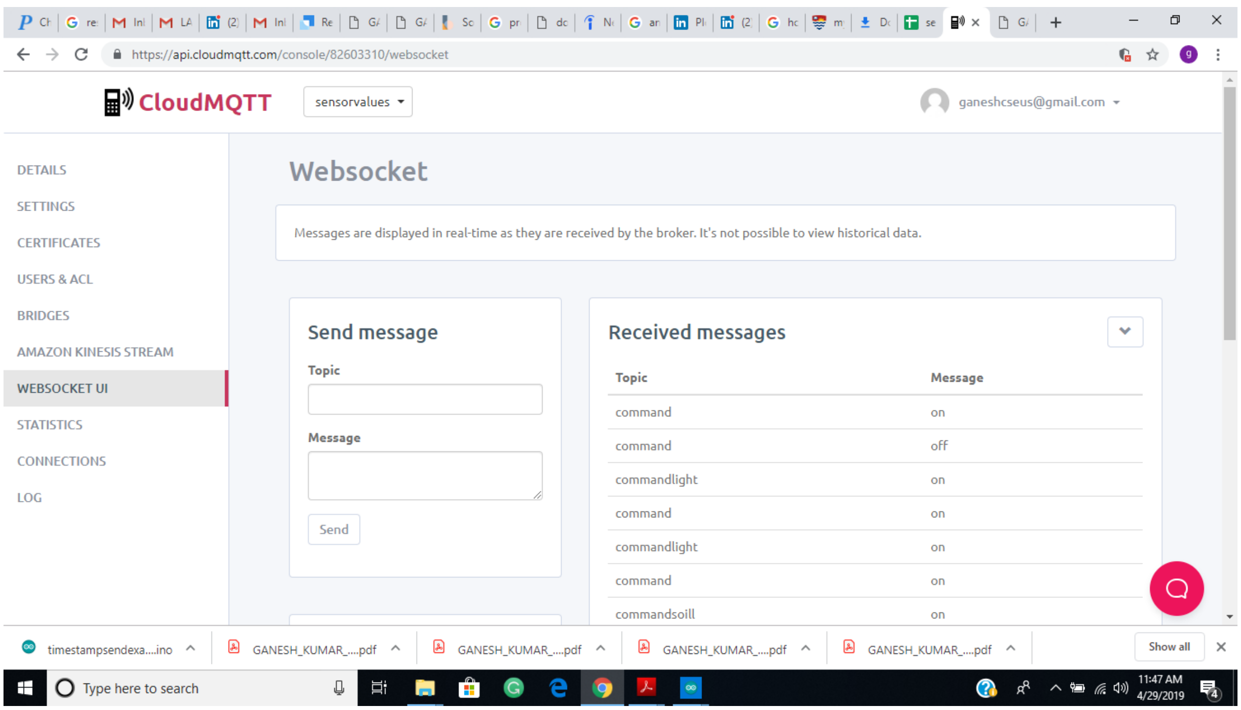
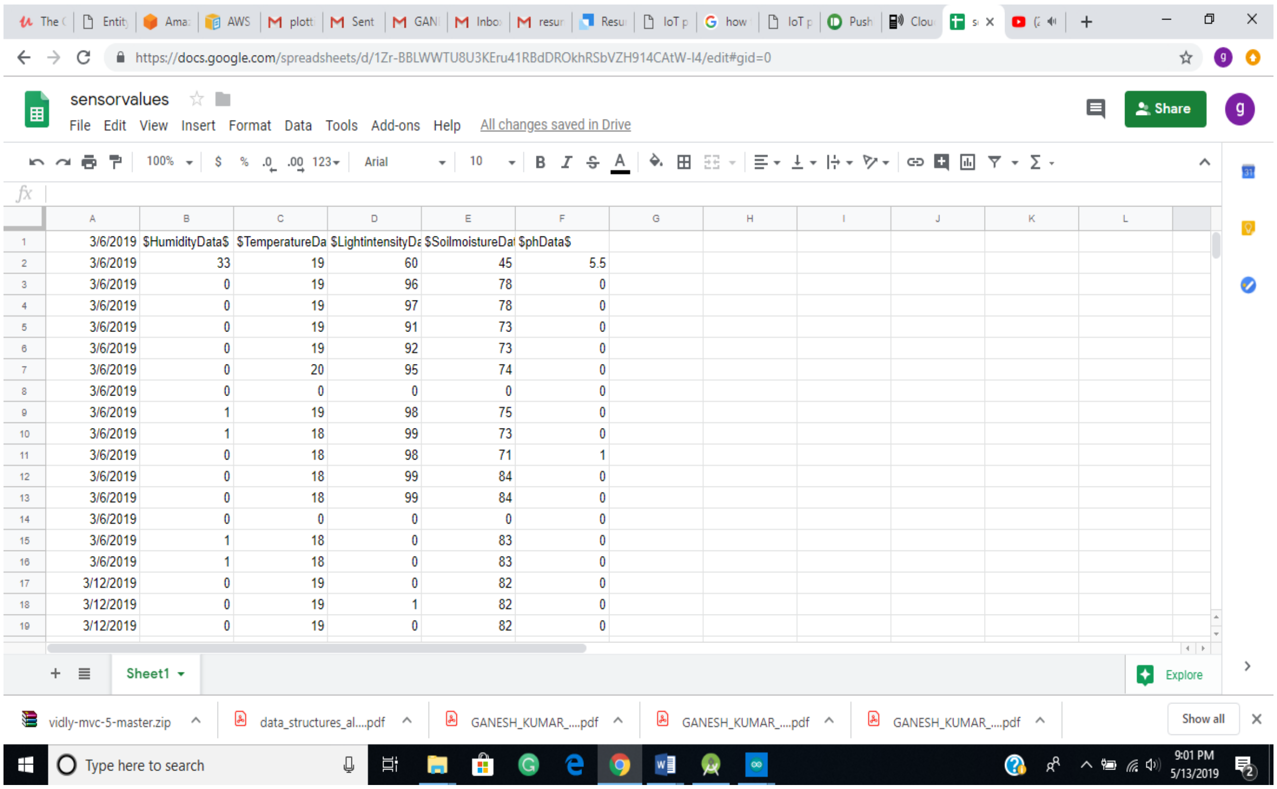


Fig 4.9 Screenshot of Cloud MQTT instances

GOOGLE SHEET



A	B	C	D	E	F
3/6/2019	$HumidityData$	$TemperatureDa	$LightintensityDa	$SoilmoistureDat	$phData$
3/6/2019	33	19	60	45	5.5
3/6/2019	0	19	96	78	0
3/6/2019	0	19	97	78	0
3/6/2019	0	19	91	73	0
3/6/2019	0	19	92	73	0
3/6/2019	0	20	95	74	0
3/6/2019	0	0	0	0	0
3/6/2019	1	19	98	75	0
3/6/2019	1	18	99	73	0
3/6/2019	0	18	98	71	1
3/6/2019	0	18	99	84	0
3/6/2019	0	18	99	84	0
3/6/2019	0	0	0	0	0
3/6/2019	1	18	0	83	0
3/6/2019	1	18	0	83	0
3/12/2019	0	19	0	82	0
3/12/2019	0	19	1	82	0
3/12/2019	0	19	0	82	0

Fig 4.10 Screenshot of Google Sheet

<u>ANDROID APPLICATION</u>

Login:

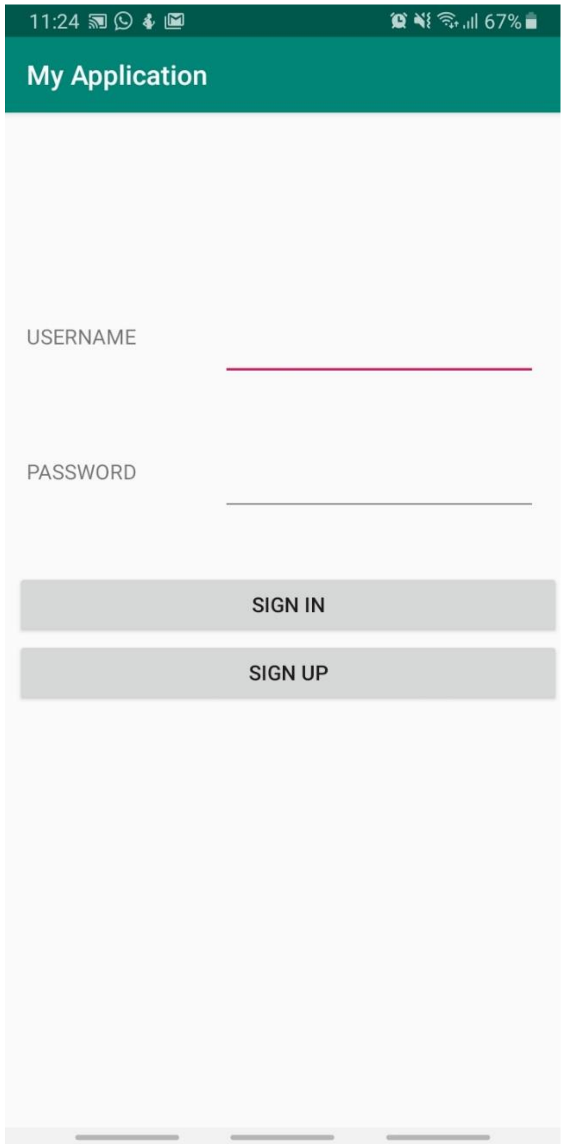


Fig 4.11 Login page of Android application

Register:

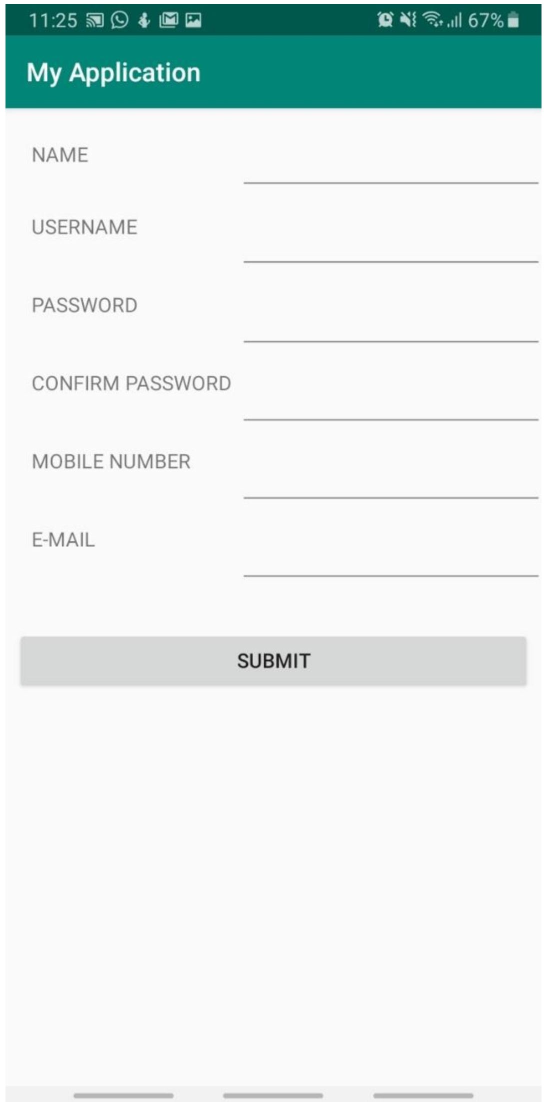


Fig 4.12 Registration page of Android application

Sensor Values:

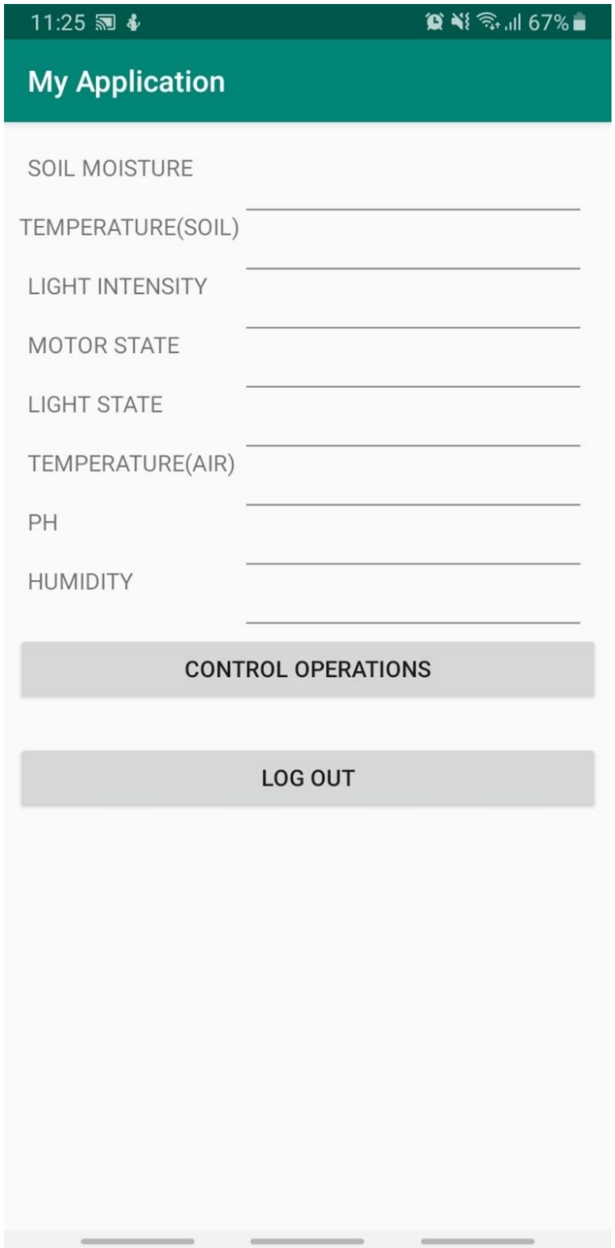


Fig 4.13 Sensor value display in Android application

Manual Control:

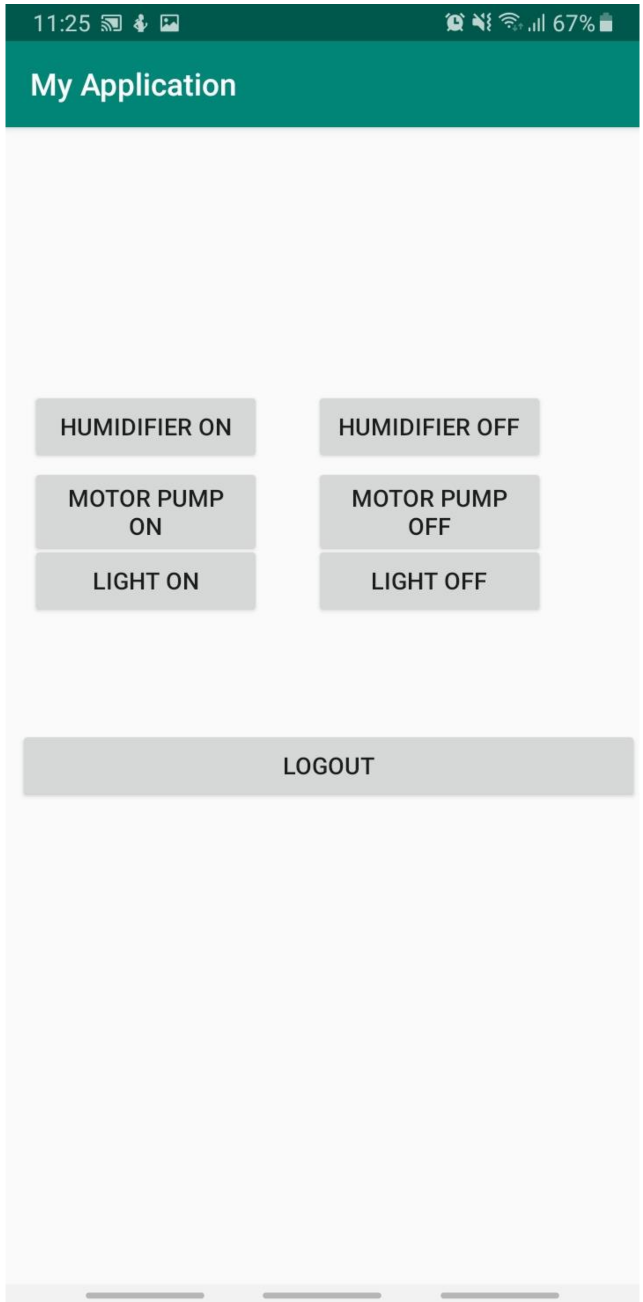


Fig 4.14 Manual mode operation in Android application

Notification:

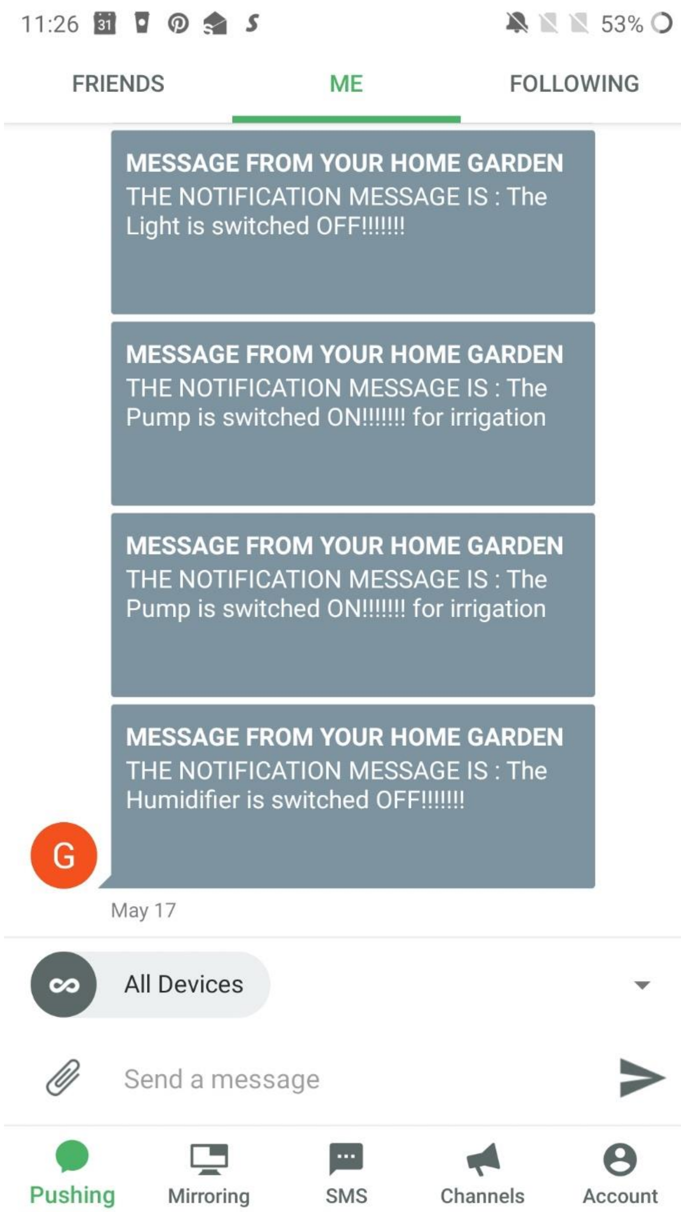


Fig 4.15 Notification in Android application

CHAPTER 5

CONCLUSION AND FUTURE SCOPE

5.1 Conclusion

In this report, we have discussed about the implementation of automation in green house. The development of this project is inexpensive and provide very superior performance for the diverse task. Microcontroller in the system promises about increase in system life by reducing the power consumption resulting in the lower power consumption of the actuators. The system is very useful since it is difficult to find the farm hand for assisting in lot of countries. It is very effective and efficient model for monitoring the life of a plant from remote location without any difficulty.

Following are the benefit and drawback of this model:

5.1.1 Benefit

- Inexpensive and easy to install.
- The use of google sheet for storing the sensor data.
- The use of MQTT protocol for faster M2M communication.
- All the necessary sensors needed for monitoring the plant growth has been used.
- Simple user interface for the android application.
- Secured model.

5.1.2 Drawbacks

- Difficult to maintain the model.
- Difficult to identify the error.
- Sensors are not reliable since they are inexpensive.
- Frequent communication error can occur due to poor or no internet connection.
- The entire model is completely dependent on the internet connection.

5.2 Future scope

For the future scope, purposes system can expand as follow:

- The use of camera module can provide the real-time surveillance of the plant.
- The use of image analysis for predicting the disease.
- The data can be analyzed for predicting the timely behavior of the actuator
- The incorporation of machine learning enhances the efficient way to monitor the growth of the plant.

CHAPTER 6

BIBILIOGRAPHY

[1] Patil, K. A., and N. R. Kale. "A model for smart agriculture using IoT." In *2016 International Conference on Global Trends in Signal Processing, Information Computing and Communication (ICGTSPICC)*, pp. 543-545. IEEE, 2016.
[2] Gardašević, Gordana, Mladen Veletić, Nebojša Maletić, Dragan Vasiljević, Igor Radusinović, Slavica Tomović, and Milutin Radonjić. "The IoT architectural framework, design issues and application domains." *Wireless personal communications* 92, no. 1 (2017): 127-148.
[3] Matharu, Gurpreet Singh, Priyanka Upadhyay, and Lalita Chaudhary. "The internet of things: Challenges & security issues." In *2014 International Conference on Emerging Technologies (ICET)*, pp. 54-59. IEEE, 2014.
[4] Sreekantha, D. K., and A. M. Kavya. "Agricultural crop monitoring using IOT-a study." In *2017 11th International Conference on Intelligent Systems and Control (ISCO)*, pp. 134-139. IEEE, 2017.
[5] Abidin, Shah Abd Hafiz Zainal, and S. Noorjannah Ibrahim. "Web-based monitoring of an automated fertigation system: An IoT application." In *2015 IEEE 12th Malaysia International Conference on Communications (MICC)*, pp. 1-5. IEEE, 2015.
[6] Vatari, Sheetal, Aarti Bakshi, and Tanvi Thakur. "Green house by using IOT and cloud computing." In *2016 IEEE International Conference on Recent Trends in Electronics, Information & Communication Technology (RTEICT)*, pp. 246-250. IEEE, 2016.
[7] Patil, Suyash S., and Sandeep A. Thorat. "Early detection of grapes diseases using machine learning and IoT." In *2016 Second International Conference on Cognitive Computing and Information Processing (CCIP)*, pp. 1-5. IEEE, 2016.
[8] Lin, Danping, C. K. M. Lee, and Kangwei Lin. "Research on effect factors evaluation of internet of things (IOT) adoption in Chinese agricultural supply chain." In *2016 IEEE International Conference on Industrial Engineering and Engineering Management (IEEM)*, pp. 612-615. IEEE, 2016.
[9] Kassim, Mohamed Rawidean Mohd, Ibrahim Mat, and Ahmad Nizar Harun. "Wireless Sensor Network in precision agriculture application." In *2014 International Conference on Computer, Information and Telecommunication Systems (CITS)*, pp. 1-5. IEEE, 2014.
[10] Baranwal, Tanmay, and Pushpendra Kumar Pateriya. "Development of IoT based smart security and monitoring devices for agriculture." In *2016 6th International Conference-Cloud System and Big Data Engineering (Confluence)*, pp. 597-602. IEEE, 2016.
[11] Khattab, Ahmed, Ahmed Abdelgawad, and Kumar Yelmarthi. "Design and implementation of a cloud-based IoT scheme for precision agriculture." In *2016 28th International Conference on Microelectronics (ICM)*, pp. 201-204. IEEE, 2016.
[12] Karim, Foughali, and Fathalah Karim. "Monitoring system using web of things in precision agriculture." *Procedia Computer Science* 110 (2017): 402-409.
[13] Lee, Meonghun, Jeonghwan Hwang, and Hyun Yoe. "Agricultural production system based on IoT." In *2013 IEEE 16th International Conference on Computational Science and Engineering*, pp. 833-837. IEEE, 2013.

[14] Gayatri, M. K., J. Jayasakthi, and GS Anandha Mala. "Providing Smart Agricultural solutions to farmers for better yielding using IoT." In *2015 IEEE Technological Innovation in ICT for Agriculture and Rural Development (TIAR)*, pp. 40-43. IEEE, 2015.
[15]Kapoor, Ayush, Suchetha I. Bhat, Sushila Shidnal, and Akshay Mehra. "Implementation of IoT (Internet of Things) and Image processing in smart agriculture." In *2016 International Conference on Computation System and Information Technology for Sustainable Solutions (CSITSS)*, pp. 21-26. IEEE, 2016.
[16] Roopaei, Mehdi, Paul Rad, and Kim-Kwang Raymond Choo. "Cloud of things in smart agriculture: Intelligent irrigation monitoring by thermal imaging." *IEEE Cloud computing* 4, no. 1 (2017): 10-15.
[17] Grgić, Krešimir, Ivan Špeh, and Ivan Heđi. "A web-based IoT solution for monitoring data using MQTT protocol." In *2016 International Conference on Smart Systems and Technologies (SST)*, pp. 249-253. IEEE, 2016.
[18] Mohanraj, I., Kirthika Ashokumar, and J. Naren. "Field monitoring and automation using IOT in agriculture domain." *Procedia Computer Science* 93 (2016): 931-939.
[19] Bo, Yifan, and Haiyan Wang. "The application of cloud computing and the internet of things in agriculture and forestry." In *2011 International Joint Conference on Service Sciences*, pp. 168-172. IEEE, 2011.
[20] Dan, L. I. U., Cao Xin, Huang Chongwei, and Ji Liangliang. "Intelligent agriculture greenhouse environment monitoring system based on IOT technology." In *2015 International Conference on Intelligent Transportation, Big Data and Smart City*, pp. 487-490. IEEE, 2015. [21]
[22]Cambra, Carlos, Sandra Sendra, Jaime Lloret, and Laura Garcia. "An IoT service-oriented system for agriculture monitoring." In *2017 IEEE International Conference on Communications (ICC)*, pp. 1-6. IEEE, 2017.
[23] Ye, Jiuyan, Bin Chen, Qingfeng Liu, and Yu Fang. "A precision agriculture management system based on Internet of Things and WebGIS." In *2013 21st International Conference on Geoinformatics*, pp. 1-5. IEEE, 2013.
[24] Prathibha, S. R., Anupama Hongal, and M. P. Jyothi. "IOT Based monitoring system in smart agriculture." In *2017 International Conference on Recent Advances in Electronics and Communication Technology (ICRAECT)*, pp. 81-84. IEEE, 2017.