A Project Report on

Grow Organic Using IOT

In partial fulfilment for the award of the degree

Of

BACHELOR OF ENGINEERING

In

COMPUTER ENGINEERING

Submitted By
Group ID: GIT_CE_18_53
Helly Patel (150120107093)
Manil Patel (150120107099)
Mihir Patel(150120107101)
Nidhi Patel(150120107103)

Guided By

Prof. Mihir Shah
Assistant Professor

Computer Engineering Department
GANDHINAGAR INSTITUTE OF TECHNOLOGY
Moti-Bhoyan, Gandhinagar

2018-2019

Acknowledgement

We have taken efforts in this project; however, it would not have been possible without the kind support and help of many individuals and organizations. We would like to extend our sincere thanks to all of them.

We are highly indebted to **Prof. Mihir Shah** for their guidance and constant supervision as well as for providing necessary information regarding the project. We take this opportunity to thank all my friends and colleagues who started us out on the topic and provided extremely useful review and feedback and for their all-time support and help in each and every aspect of the course of our project preparation. We are grateful to our institute Gandhinagar Institute of Technology, Gandhinagar for providing us all required resources and good working environment.

We would like to express my gratitude towards Head of Department, **Prof. Archana Singh** and Principal, **Dr. H N Shah** for their kind co-operation and encouragement which help us in this project.

Helly Patel (150120107093)
Manil Patel (150120107099)
Mihir Patel (150120107101)
Nidhi Patel (150120107103)

Abstract

Nowadays, the challenge for improving plant growth with lessening its input costs is justifying a concept of “Grow Organic System” application for gardening and in agriculture, particularly in the resident backyard. This will minimize the wastage of water consumption by its overhead monitoring and is a time saving devices. These feedback-based approaches facilitate more economical handling of resources than that of open-loop systems. A user-defined mobile application helps the user to access for real-time individual plant growth at regular intervals. This system comprises of several circuits and it has a relatively low-cost of installation & construction. Its different segments modules have continuously been operated and tested, and their effectiveness in reducing water consumption as well as liquid fertilizers and besides, reduction in human interference has been there. However, more testing on the system as a whole must be conducted in the future to measure the advance water consumption technology.The design is resource-efficient technology which has also low power consumption.

PLATINUM FOUNDATION MANAGED
GANDHINAGAR INSTITUTE OF TECHNOLOGY
(Affiliated to Gujarat Technological University)
Khatraj-Kalol Road, At: Moti Bhoyan, Tal: Kalol, Dist: Gandhinagar
Website: www.git.org.in

CERTIFICATE

This is to certify that the work of Industrial/User Defined Project entitled **"Grow Organic Using IOT"** has been carried out by **Helly Patel (150120107093) , Manil Patel (150120107099) , Mihir Patel (150120107101) , Nidhi Patel (150120107103)** under my guidance in partial fulfilment for the degree of Bachelor of Engineering in **Computer Engineering** 8th Semester at the Computer Engineering Department, **Gandhinagar Institute of Technology**, Moti-Bhoyan, Gandhinagar, Gujarat, during the academic year 2018-2019 and his/her work is satisfactory. This student has successfully completed all the activity under my guidance related to User Defined Project for 8th semester.

Internal Guide,
Prof. Mihir Shah,
Assistant Professor,
Computer Engineering Department

Prof. Archana Singh,
Head of Department,
Computer Engineering,
Gandhinagar Institute of Technology

GUJARAT TECHNOLOGICAL UNIVERSITY

[UNDERTAKING ABOUT ORIGINALITY OF WORK]

We hereby certify that we are the sole authors of this UDP project report and that neither any part of this UDP project report nor the whole of the UDP Project report has been submitted for a degree by other students to any other University or Institution.

We certify that, to the best of our knowledge, the current UDP Project report does not infringe upon anyone's copyright nor violate any proprietary rights and that any ideas, techniques, quotations or any other material from the work of other people included in our UDP Project report, published or otherwise, are fully acknowledged in accordance with the standard referencing practices. Furthermore, to the extent that we have included copyrighted material that surpasses the boundary of fair dealing within the meaning of the Indian Copyright (Amendment) Act 2012, we certify that we have obtained a written permission from the copyright owners to include such materials in the current UDP Project report and have included copies of such copyright clearances to our appendix.

We have checked the write up of the present UDP Project report using anti-plagiarism database and it is in the allowable limit. In case of any complaints pertaining to plagiarism, we certify that we shall be solely responsible for the same and we understand that as per norms, University can even revoke BE degree conferred upon the students submitting this UDP Project report, in case it is found to be plagiarised.

Team Id: GIT_CE_18_53

Enrolment number	Name	Signature
150120107093	Helly Patel	
150120107099	Manil Patel	
150120107101	Mihir Patel	
150120107103	Nidhi Patel	

Place: Gandhinagar **Date:** ________________

Name of Guide: Prof. Mihir Shah

Signature of Guide ___________________

Contents

	Acknowledgement		I
	Abstract		II
	Certificate (From College)		III
	Certificate (PMMS Completion Certificate)		IV
	Undertaking Form		VIII
	Contents		IX
	List of Figures		X
	List of Tables		XI
Chapter 1	**Introduction**		
	1.1	Project Detail	1
		1.1.1 Project Profile	1
		1.1.2 Project Definition	1
	1.2	Purpose	1
	1.3	Scope	2
	1.4	Objective	2
	1.5	Tools and Technology	3
	1.6	Literature Review	9
Chapter 2	**About The System**		
	2.1	System Requirement Specification(SRS)	12
	2.2	Project Planning	18
Chapter 3	**Analysis of the System**		
	3.1	E R Diagram	20
	3.2	Data Flow Diagram	20
	3.3	Use Case Diagram	21
	3.4	Sequence Diagram	22
	3.5	Activity Diagram	23
	3.6	Class Diagram	25
Chapter 4	**Design**		
	4.1	System Flow Diagram	27
	4.2	User Interface	28
	4.3	Application Navigation	28
Chapter 5	**Implementation**		
	5.1	Implementation Environment	30
	5.2	Security Feature	33
	5.3	Coding Standard	33
	5.4	Business Logic Layer	34

Chapter 6	**Testing**	
	6.1 Testing Plan	35
	6.2 Testing Strategy	36
	6.3 Testing Methods	36
	6.4 Test Cases	38
Chapter 7	**Application Screenshot**	
	7.1 Client Side with Description	39
	7.2 Server Side with Description	41
Chapter 8	**Conclusion & Future Work**	
	8.1 Conclusion	43
	8.2 Future Work	43
Appendix A	BMC Canvas	44
	References	46

List of Figures

Figure Number	Description	Page Number
Fig 1.5.4.1	Connection Diagram of DHT11 sensor	4
Fig 1.5.3	Pin Diagram of Arduino Uno	6
Fig 1.5.4.2	Arduino Uno device snapshot	7
Fig1.5.5	Soil Moisture Sensor	7
Fig 1.5.6	Relay Module	8
Fig 1.5.7	NodeMCU	9
Fig 1.5.8	Dc pump	9
Fig 3.1	E-R Diagram for grow Organic Using IOT	20
Fig 3.1.1	1 DFD level 0	21
Fig 3.3	Use Case Diagram for System	22
Fig 3.4	Sequence Diagram for System	23
Fig 3.5	Activity Diagram for System	24
Fig 3.6	Class Diagram	26
Fig 4.1	System Flow Diagram	27
Fig 4.3	Application Screenshot	29
Fig 5.1.1	Arduino IDE Screenshot	31
Fig 7.1.1	Circuit Diagram of Soil Moisture Connection	39
Fig 7.1.2	Circuit Diagram of Relay connection to Arduino	39
Fig 7.1.3	Diagram of Relay connection to pump	40
Fig 7.1.4	Diagram of DHT11 connection to Arduino	40
Fig 7.1.5	Diagram of NodeMCU connection to Arduino	41
Fig 7.1.6	System circuit diagram	41
Fig 7.1.7	NodeMCU code 1	42
Fig 7.1.8	NodeMCU code 2	42
Fig 7.1.9	NodeMCU code 3	43
Fig 7.1.10	Arduino code 1	43
Fig 7.1.11	Arduino code 2	44
Fig 7.1.12	Implementation of control parts	44
Fig 7.1.13	Complete Circuit Implementation	45
Fig 7.2.1	Application Screenshot	46
Fig 7.2.2	Balcony Layout	46
Fig 7.2.3	Backyard Layout	47
Fig 7.2.4	Terrace Layout	47

List of Table

Table Number	Description	Page Number
Table 1.5.3	Pin Identification and Configuration of DHT11	3
Table 1.5.4.1	Pin Description of Arduino	5
Table 1.5.4.2	Arduino Uno Technical Specifications	6
Table 1.5.5	Specification Of soil Moisture Sensor	8
Table 5.1	List of Tool/Technology	30
Table 6.4	Pump supply Test Case	38

Chapter 1
Introduction

1.1 Project Detail

1.1.1 Project Profile

Water is a resource that all living species need. It is therefore very precious and has to be used with moderation to be preserved for the generations to come. Agriculture is an industry that uses a lot of water. Most of the time, this resource is not used efficiently and substantial amounts of water are wasted. In the near future, these wastes will represent a large sum of money. The ones who manage this resource efficiently will be winning time and money.

In this project report, a Grow Organic System is suggested to minimize the water input and human intervention, while satisfying the plant needs. Firstly, the details of the problem are summarized. The objective and the scope of the project are described. Some general approaches to the design are reviewed. The results and conclusions of an experiment to determine the required amounts of water are discussed. Then, the suggested design is explained in detail with the purpose, requirements and constraints, simulation and test results for each of its parts. A brief cost analysis is performed to estimate the viability of such a project on the market. Finally, the design is criticized, and suggestions are made for future improvements.

1.1.2 Project Definition

- **Project Name**: Grow Organic using IOT
- **Definition:** It is based on IOT precisely on Arduino hardware and Application in which a person can check about the plant moisture level. Moreover, through this device there is automation for watering plants based on moisture sensor as a result, user can grow the plant very easily and efficiently.

1.2 Purpose

Irrigation of plants is usually a very time-consuming activity; to be done in a reasonable amount of time, it requires a large amount of human resources. Traditionally, all the steps were executed by humans. Nowadays, some systems use technology to reduce the number of workers or the time required to water the plants. With such systems, the control is very limited, and many resources are still wasted.

In this system, soil moisture sensor senses the moisture level of the soil. If soil will get dry then sensor senses low moisture level and automatically switches on the water pump to supply water to the plant. As plant get sufficient water and soil get wet then sensor senses enough moisture in soil. After which the water pump will automatically get stopped.

Also in case, when power supply gets cut-off and motor gets switched off. It will restart again automatically when there will be availability power supply, user will have not to worry about restarting the motor pump manually.

- Sensor Network : Moisture, Weather condition, Humidity, Sunlight
- Scheme : Based on real time sensor Data
- Connectivity : Wireless
- Real time Data logging : Temperature/humidity/soil conditions

1.3 Scope

A critical consideration is the installation costs, since costs generally determine the feasibility and viability of a project. The installation must be simple enough for a domestic user. The water savings was also an important aspect, since there is a demand to minimize water loss and to maximize the efficiency of water used. Since the objective is to minimize the cost of labor, minimal supervision and calibration must be needed. The system must operate with optimized consistency. The power consumption must also be monitored. For maintenance, the replacement parts must be readily available and easy to install in the case of failure. Finally, the possibility for implementing the system at a larger scale e.g. in greenhouses should be investigated.

1.4 Objective

- The objective of this project was to design a small-scale Grow Organic system for indoors that would use water in a more efficient way, in order to prevent water loss. Also provides Automation in watering and provide remote access
- Water Saving
- Conveniences
- Remote Access
- Time Saving
- Effective

1.5 Tools and Technology

1.5.1 Internet of Things

The internet of things, or IoT, is a system of interrelated computing devices, mechanical and digital machines, objects, animals or people that are provided with unique identifiers (UIDs) and the ability to transfer data over a network without requiring human-to-human or human-to-computer interaction. The internet of things, or IoT, is a system of interrelated computing devices, mechanical and digital machines, objects, animals or people that are provided with unique identifiers (UIDs) and the ability to transfer data over a network without requiring human-to-human or human-to-computer interaction.

1.5.2 Android:

The android is a powerful operating system and it supports large number of applications in Smartphone. These applications are more comfortable and advanced for the users. The hardware that supports android software is based on ARM architecture platform. The android is an open source operating system means that it's free and any one can use it. The android has got millions of apps available that can help you managing your life one or other way and it is available low cost in market at that reasons android is very popular.

1.5.3 DHT11–Temperature and Humidity Sensor :

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

No:	Pin Name	Description
1	Vcc	Power supply 3.5V to 5.5V
2	Data	Outputs both Temperature and Humidity through serial Data
3	Ground	Connected to the ground of the circuit

Table 2.5.3 Pin Identification and Configuration of *DHT11*

DHT11 Specifications:

- Operating Voltage: 3.5V to 5.5V
- Operating current: 0.3mA (measuring) 60uA (standby)
- Output: Serial data
- Temperature Range: 0°C to 50°C
- Humidity Range: 20% to 90%
- Resolution: Temperature and Humidity both are 16-bit
- Accuracy: ±1°C and ±1%

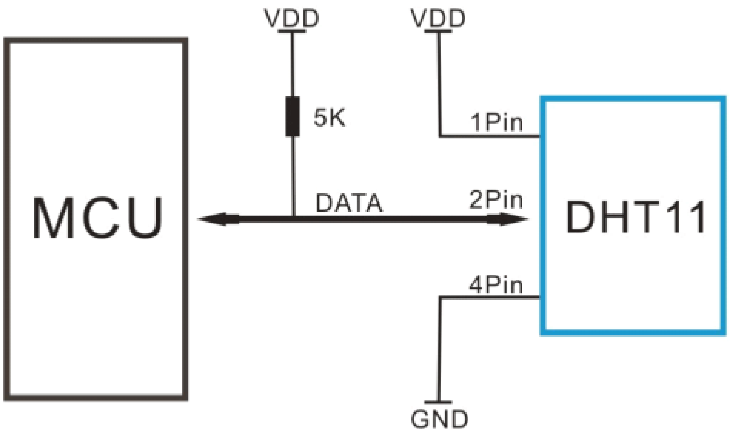


Fig 1.5.3 Connection Diagram of DHT11 sensor

As you can see the data pin is connected to an I/O pin of the MCU and a 5K pull-up resistor is used. This data pin outputs the value of both temperature and humidity as serial data. If you are trying to interface DHT11 with Arduino then there are ready-made libraries for it which will give you a quick start.

If you are trying to interface it with some other MCU then the datasheet given below will come in handy. The output given out by the data pin will be in the order of 8bit humidity integer data + 8bit the Humidity decimal data +8 bit temperature integer data + 8bit fractional temperature data +8 bit parity bit. To request the DHT11 module to send these data the I/O pin has to be momentarily made low and then held high as shown in the timing diagram below

1.5.4 Arduino

The Arduino Uno is a microcontroller board based on the ATmega328. It has 14 digital input/output pins (of which 6 can be used as PWM outputs), 6 analog inputs, a 16 MHz ceramic resonator, a USB connection, a power jack, an ICSP header, and a reset button. It contains everything needed to support the microcontroller; simply connect it to a computer with a USB cable or power it with a AC-to-DC adapter or battery to get started. The Uno differs from all preceding boards in that it does not use the FTDI USB-to-serial driver chip. Instead, it features the Atmega16U2 (Atmega8U2 up to version R2) programmed as a USB-to-serial converter.

When ATmega328 chip is used in place of Arduino Uno, or vice versa, the image below shows the pin mapping between the two.

Pin Category	Pin Name	Details
Power	Vin, 3.3V, 5V, GND	Vin: Input voltage to Arduino when using an external power source. 5V: Regulated power supply used to power microcontroller and other components on the board. 3.3V: 3.3V supply generated by on-board voltage regulator. Maximum current draw is 50mA. GND: ground pins.
Reset	Reset	Resets the microcontroller.
Analog Pins	A0 – A5	Used to provide analog input in the range of 0-5V
Input/Output Pins	Digital Pins 0 - 13	Can be used as input or output pins.
Serial	0(Rx), 1(Tx)	Used to receive and transmit TTL serial data.
External Interrupts	2, 3	To trigger an interrupt.
PWM	3, 5, 6, 9, 11	Provides 8-bit PWM output.
SPI	10 (SS), 11 (MOSI), 12 (MISO) and 13 (SCK)	Used for SPI communication.
Inbuilt LED	13	To turn on the inbuilt LED.
TWI	A4 (SDA), A5 (SCA)	Used for TWI communication.
AREF	AREF	To provide reference voltage for input voltage.

Table 1.5.4.1 Pin Description of Arduino

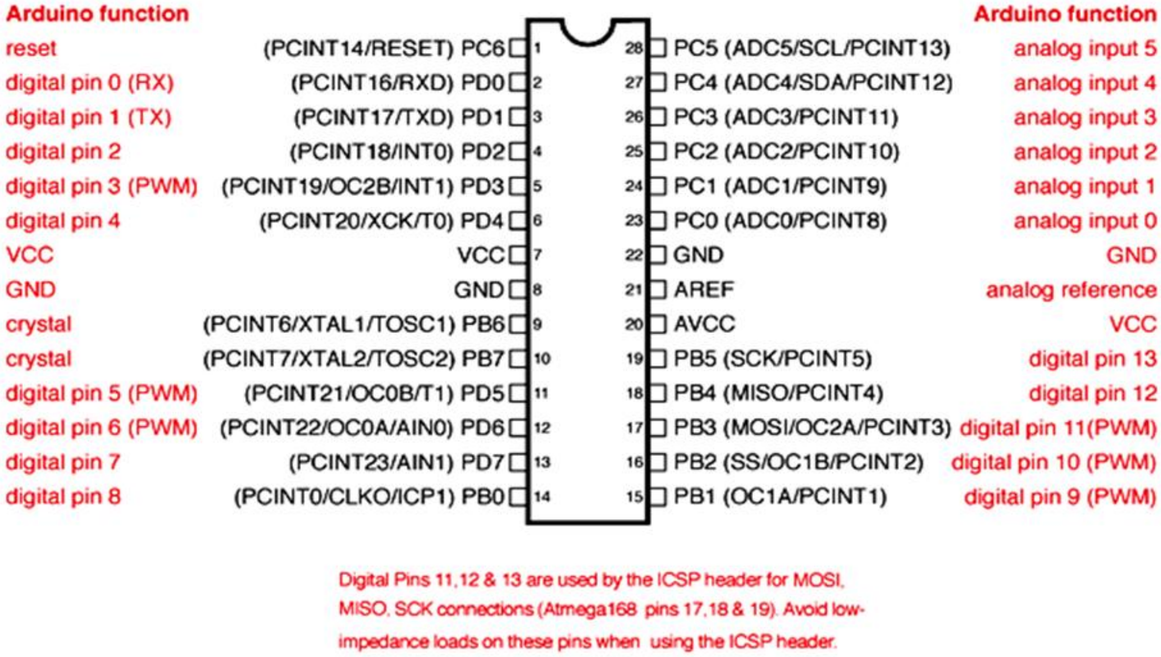


Fig 1.5.4.1 Pin Diagram of Arduino Uno

Microcontroller	ATmega328P – 8 bit AVR family microcontroller
Operating Voltage	5V
Recommended Input Voltage	7-12V
Input Voltage Limits	6-20V
Analog Input Pins	6 (A0 – A5)
Digital I/O Pins	14 (Out of which 6 provide PWM output)
DC Current on I/O Pins	40 mA
DC Current on 3.3V Pin	50 mA
Flash Memory	32 KB (0.5 KB is used for Bootloader)
SRAM	2 KB
EEPROM	1 KB
Frequency (Clock Speed)	16 MHz

Table 1.5.4.2 Arduino Uno Technical Specifications

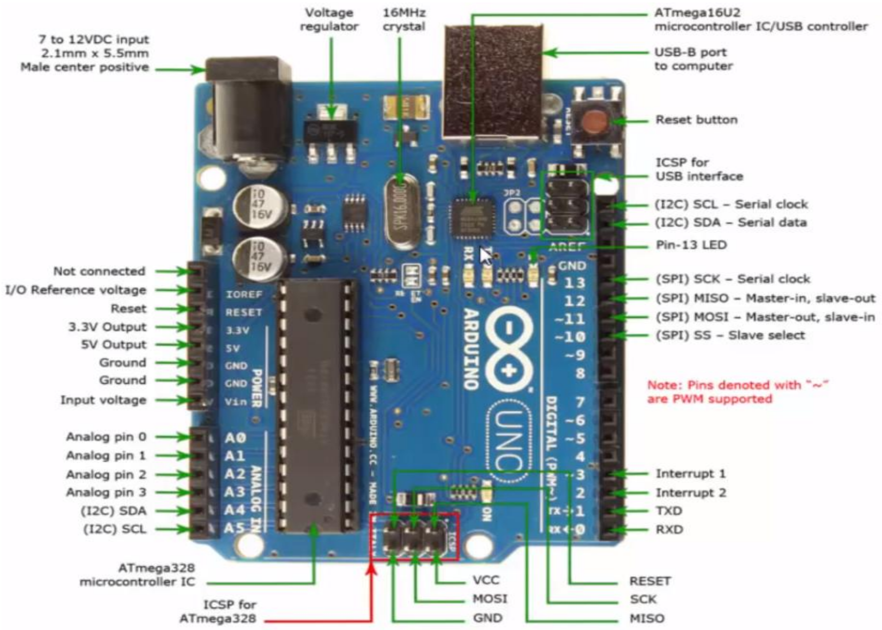


Fig 1.5.4.2 Arduino Uno device snapshot

1.5.5 Soil Moisture Sensor

Soil moisture sensors typically refer to sensors that estimate volumetric water content. Another class of sensors measure another property of moisture in soils called water potential; these sensors are usually referred to as soil water potential sensors and include densitometers and gypsum blocks.

Fig1.5.5 Soil Moisture Sensor

Brand Name	REES52
EAN	0781119450855
Item Weight	9.0 grams
Manufacturer Series Number	SOILSENSOR
Number of Items	1
Part Number	SOILSENSOR
UPC	781119450855

Table 1.3.5 Specification Of soil Moisture Sensor

Interface Description (4-wire)
1. VCC: .3 V-5V
2. GND: GND
3. DO: digital output interface (0 and 1)
4. AO: Analog Output Interface

1.5.6 Relay

Relay is an electrically operated switch. Several relays use a magnet to automatically operate a switch, however alternative in operation principles are used, like solid state relays. Relays are used wherever it's necessary to regulate a circuit by a separate low-power signal, or wherever many circuits should be controlled by one signal. The essential relays were handling in long distance communicate circuits as amplifiers, they unbroken the signal coming back in from one circuit and re-transmitted it on another circuit.

Fig 1.5.6 Relay Module

1.5.7 NodeMCU

NodeMCU is an open source IoT platform. It includes firmware which runs on the ESP8266 Wi-Fi SoC from Espressif Systems, and hardware which is based on the ESP-12 module. The term "NodeMCU" by default refers to the firmware rather than the development kits. The firmware uses the Lua scripting language.

Fig 1.5.7 NodeMCU

- Memory: 128kBytes
- Developer: ESP8266 Open source Community
- Operating system: XTOS
- Storage: 4MBytes
- CPU: ESP8266(LX106)
- Power: USB

1.5.8 Motor 12V

The water pump is used to artificially supply water for a particular task. It can be electronically controlled by interfacing it to a microcontroller. It can be triggered ON/OFF by sending signals as required. The process of artificially supplying water is known as pumping.

There are many varieties of water pumps used. This project employs the use of a small water pump

The pumping of water is a basic and practical technique, far more practical than scooping it up with one's hands or lifting it in a hand-held bucket. This is true whether the water is drawn from a fresh source, moved to a needed location, purified, or used for irrigation, washing, or sewage treatment, or for evacuating water from an undesirable location. Regardless of the outcome, the energy required to pump water is an extremely demanding component of water consumption. All other processes depend or benefit either from water descending from a higher elevation or some pressurized plumbing system.

Fig 1.5.8 Dc pump

1.6 Literature Review

It is a simple project more useful in watering plants automatically without any human interference. We know that people do not pour the water on to the plants in their gardens when they go to vacation or often forget to water plants. As a result, there is a chance to get the plants damaged. This project is an excellent solution for such kind of problems. Many irrigation systems exits such as

Archana and Priya (2016) projected a paper during which the wetness and soil wetness sensors are placed within the root zone of the plant. Supported the perceived values the microcontroller is employed to manage the availability of water to the sphere. This system doesn’t intimate the farmer concerning the sphere standing.

Sonali D.Gainwar and Dinesh V. Rojatkar (2015) planned a paper during which soil parameters like pH scale, humidity, wetness and temperature are measured for obtaining high yield from soil. This method is absolutely machine-controlled that turns the motor pump ON/OFF as per the amount of wetness within the soil. The present field standing isn't intimated to the farmer.

V. R. Balaji and M. Sudha (2016) projected a paper in that the system derives power from daylight tho' photo-voltaic cells .This system doesn’t rely on electricity. The soil wetness sensing element has been used and supported the perceived values PIC microcontroller is employed to ON/OFF the motor pump. Meteorology isn't enclosed during this system.

R.Subalakshmi (2016) projected a paper to create irrigation system less complicated, the complexities concerned in irrigation is tackled with automation 3rd National Conference on Intelligent info and Computing Technologies, IICT ‘17 a pair of system mistreatment microcontroller and GSM. Supported the perceived values from soil wetness, temperature and wetness sensors, and the GSM sends message to the farmer once these parameters exceed the edge worth set within the program. The nutrient content within the soil isn't determined by this method.

Karan kansara (2015) projected an automatic irrigation system wherever the wetness and temperature sensors ar used to sense the soil conditions and supported that microcontroller can management the water flow. Farmer can be intimated through GSM. This system doesn’t monitor the nutrient content within the soil.

Prof C.H.Chavan and P.V.Karnade (2014) projected a sensible wireless sensing element network for observance environmental parameters mistreatment Zigbee. These nodes send information wirelessly to a central server that collects information, stores it and permits it to be analyzed then displayed pro re nata and conjointly be sent to the consumer mobile. Meteorology and nutrient content isn't determined during this system.

G.Parameswaran and K.Sivaprasath (2016) planned a wise drip irrigation system mistreatment IOT during which wetness, temperature and pH scale sensors ar used. Irrigation standing is updated to the server or native host mistreatment pc. The farmer can’t access concerning the sphere condition while not web.

S.Reshma and B.A.Sarath (2016) projected AN IOT based mostly automatic irrigation system mistreatment wireless sensing element networks in that numerous sensors live} wont to measure the soil parameters. This method provides an online interface to the user to monitor and management the system remotely. Weather observance is not done during this system.

Joaquin Gutierrez (2013) proposed an entryway unit that handles sensing element info, triggers actuators, and transmits information to net application. it's hopped-up by electrical phenomenon panels and has duplex communication link based mostly on cellular web interface that permits for information examination and irrigation programming to be programmed through online page.

Chapter 2
About The System

2.1 System Requirement Specification

2.1.1 Introduction

- **Purpose**
  We will develop a model along with water storage which occupy less space and grow more plants. All watering activities will be governed by a smart controller which will be able to collect information about current Moisture and allow the user to know watering amounts and times based on those readings. The system automatically decides to water plants based on moisture readings.
- **Scope**
  Nowadays people are not able to maintain regular growth of their plants. With the help of our system user can smoothly and flexibly provide water supply to a plant.

2.1.2 Description

- **Product Perspective**
  Application-based plant watering system is increasingly used to do consistent growth of plants. In Urban area, people have not enough time to regularly supply water to plants. So, it is the need of the individual to maintain their regular growth of plants. After that they can maintain the flow of water with the help of the application.

2.1.3 Design and implementation constraints

- **Software language used**
  The language that should be used for coding this application is Objective C, Android studio etc. For working on coding phase, the Web API server needs to be installed.
- **Development tool**
  We will make use of the available Arduino for working. Also, we will make use of the online references available for developing programs in Arduino resources.
- **Class Libraries**
  We will make use of the existing Arduino IOT libraries.

2.1.4 System Feature

2.1.4.1 Functional Requirements

- If the moister sensor is dry in the line (its locations), the system will checked by the rain sensor if there is rain the system will not work because no need to irrigate at the same time of rain, otherwise the system will check the temperature sensor with light sensor if the temperature is high and the percentage of light Is high as well then the system will not work because it is not the right time for irrigation process because the water will easily evaporate.

- If the temperature is low and the light is low and there is no rain but the moisture sensor is dry then it send signal to controller to open the valve and pump.
- If the level of water in the tank is low then the system will send Notification
- The user can get information about the sensors remotely via application

2.1.4.2 Requirements Process

- **Easy to implement**
  The materials required for this project must be easy to install to implement a successful project. In addition, materials should be easy to connect with each other to build this project and become more effective. Also, the materials of this project must be easy to replace it in case of any damage.
- **Open source**
  The controller used for this project is open-source, so the used hardware is reasonably priced and has a free development software.
- **Strength**
  The tools needed for this project must be strong to operate for a long period of time to achieve the desired success. In addition to achieve one of the important goals required to save money.
- **Quality**
  The tools required to build this project must be of excellent quality to operate for a long time. Selecting quality materials is very important to avoid wasting money and to avoid the technical problems of these devices and disturbance of the process. Excellent quality is required for this project to achieve the desired success of this project.
- **Modifiability**
  The material should be chosen based on its ease of modifiability, as its common to come across designs and connections. Also, to be easy to replace or modification required in the future.

2.1.4.3 Non-functional Requirements

- **Communication:**
  Bluetooth connected to system and real time clock to take all data from prototype to mobile application.
- **Accuracy:**
  The reading of all data should be in details because it will be saved in SD card for analysis and research.
- **Performance:**
  The system must work at real time
- **Operational:**
  This system is work automatically and it is connected to mobile application so user must download the application on their mobile to control their system.

- **Cost:**
  The cost of this system must be not too expensive because we aim to decrease the worker which mean decrease the amount of money in irrigation process and solve the main problem (reduce the water consumption) at the same time.

2.1.5 Performance Requirement

This section specifies the requirements necessary for the optimal performance of system. These requirements are in place to ensure that watering schedules are resourceful and efficient to optimize water conservation. Performance requirements are also in place to encompass overall responsiveness and reliability of the system.

2.1.5.1 Sensor Accuracy

- Description: Proper sensor readings of soil moisture levels must be captured to ensure accurate and efficient watering schedules. This requires solid construction of the sensor modules and proper software analysis of the data provided by the sensors.
- Constraints: Soil moisture sensors must be properly installed according to the provided instructions.
- Standards: None
- Priority: High Priority

2.1.5.2 Rain Detection

- Description: In the event of rainy weather, the rain sensor must quickly transmit an alert to the control unit to interrupt or delay any active watering.
- Constraints: The rain sensor must be properly placed and installed in order to ensure this functionality.
- Standards: None
- Priority: Moderate Priority

2.1.5.3 Communication between Application and NodeMCU

- Description: The user should be connected to internet to access the data coming from nodemcu
- Constraints: None
- Standards: None
- Priority: Highest Priority

2.1.5.4 Communication between Soil Moisture and Arduino

- Description: The soil moisture sensors will communicate with the Arduino to relay information about soil moisture levels. This must be accomplished periodically in an accurate and timely manner in order to avoid errors and to ensure consistency throughout the system.
- Constraints: None
- Standards: None
- Priority: Highest Priority

2.1.5.5 Device Power Malfunction

- Description: If the server fails to receive communication from the central control unit for a specified period of time, it will notify the user that the system is offline.
- Constraints: Control unit must be without power for a specified period of time.
- Standards: None
- Priority: Highest Priority

2.1.4.6 Communication between Relay and pump

- Description: Relay should be connected to pump as it controls the flow of water from the pump and also switches ON/OFF
- Constraints: None
- Standards: None
- Priority: Highest Priority

2.1.6 Feasibility Study

A feasibility study is a management-oriented activity. It is an analysis used in measuring the ability and likelihood to complete a project successfully include all relevant factors. It must account for factors that affect it such as economic, technology, legal and scheduling factors. It includes cost estimation, Project impact, Financial analysis, Engineering.

Factors of the feasibility studies are

- **Conduct a market survey:** - we determine total volume in the market area and estimate expected market share.
- **Prepare a projected income statement:** -this statement provides fees for doctor's services.
- **Risk analysis**:-A risk such condition where exist possibility of bad deviation from the wanted or expected disease result. (Azaab , 2002) (landing_page, n.d.)

This section evaluates the expectations of SYSTEM and all its components. A careful analysis of each component will determine feasibility assessments for the outcome of the project. In order to provide a more detailed report each individual analysis has been examined below.

2.1.6 Scope Analysis

The overall scope of SYSTEM has been examined and discussed by all team members and has been determined to be reasonable given the project deadline. After examining the requirements with the highest
priorities we determined the majority of the work will be with the web application. Our team has two members that have had years of real world experience developing large scale web applications for companies so the outlook for the web application work schedule is very feasible.

Research on the hardware components of SYSTEM has also provided our team with a positive outlook for the scope. This assessment is based on our research of each individual components and our team's technical background with each of them. While we have a few team members who have great experience in microcontrollers our concern is with the interaction between the sensors and microcontroller. This assessment is based on the lack of knowledge and experience with soil and rain sensors. This initial assessment is based purely off the research we have done regarding the sensors and the processes of gathering readings as data. In conclusion after analyzing the requirements and prioritizing them based off importance we have evaluated that the project is within scope and the probability of completion before the deadline is high.

2.1.7 Research

Our primary focus on research has been the components we are unfamiliar with. Our team has been researching different soil sensors to determine which will be the best fit for SYSTEM. This research has been evaluated and compared to the other hardware components that will interface with the sensors. We have reviewed tutorials and other projects online to study how the interactions between the sensors and controller take place. This research has aided us in deciding which microcontroller is optimal for the project and our budget.

The information we lack in home irrigation has been another primary focus of our research. Fortunately our sponsor is very knowledgeable and has even provided us with a resource that runs water conservation for the City of Arlington. He has been extremely helpful in gathering information for our team to research and has provided us with multiple resources to help minimize the learning curve.

2.1.8 Technical Analysis

A thorough technical analysis has been a priority task for our team. We determined before purchasing or finalizing any components we first needed to examine what our technical strengths and weaknesses were.

While no one on the team has had any experience with home irrigation systems we all have had experience with the majority of the individual components that it consists of. The primary hardware

Components are the central control unit, soil sensors, rain sensor, and water valve system. We have carefully diagrammed and examined each piece of hardware to determine how they will interact and function together. We have determined that the microcontroller is the component that our team is most familiar with. The central control unit will consist of a microcontroller that will be connected to each water valve and will also be responsible for powering the soil and rain sensors. All three of these components will be controlled by the web application.

2.2 Project Planning

2.2.1 Project Development Approach

The incremental model is suitable for our web-based project. The incremental build model is a method of software development where the product is designed, implemented and tested incrementally (a little more is added each time) until the product is finished. It involves both development and maintenance. The product is defined as finished when it satisfies all of its requirements. This model combines the elements of the waterfall model with the iterative philosophy of prototyping. With the help of this model, the process of creating a functioning system is accelerated. This is facilitated by the applied principle of layout from standard blocks, through which it provides control over the process of developing changing requirements.

2.2.2 Detailed schedule and milestones

- First of all we have discussed the main idea of the project, we had several meeting and each member discussed her/his idea.
- Second we made list of materials needed for this project.
- After agreeing on the idea and method of action we divided the team into two group searching. One group responsible for searching about related works and other group responsible for materials searching and where can we find it etc.
- Several Meeting have arranged to draw project sketch and make assumptions of measurements and connections.
- Then we decided to make small version of the project until we make order for all materials needed.
- Later we start working on hardware part and make all wires connection for each device.
- Later on we started on programming part where we programmed and test all sensors individuals.
- Then we combined all programming code together for the large scale project.
- We make test for the entire project to make sure everything working as we want using different condition for each sensors.
- Then we divided the team into two groups, one for writing report and other one for working and searching for software part which is mobile application and programming.

2.2.3 Project planning process

- Develop Project Management Plan
- Collect Requirements
- Define Scope
- Define Activities
- Sequence Activities
- Estimate Activity Durations
- Estimate Activity resources
- Plan Communications
- Plan Risk Management
- Start building a project

Chapter 3
Analysis of the System

3.1 E-R Diagram

An entity relationship diagram (ERD), conjointly called associate degree entity relationship model, may be a graphical illustration of associate degree data system that depicts the relationships among folks, objects, places, ideas or events inside that system. Entity relationship diagrams offer a visible start line for info style which will even be accustomed facilitate confirm data system necessities throughout the Project. After an electronic database is unrolled, associate degree ERD will still function a referral purpose, ought to any debugging or business method re-engineering be required later. The figure below shows the E-R diagram for the system

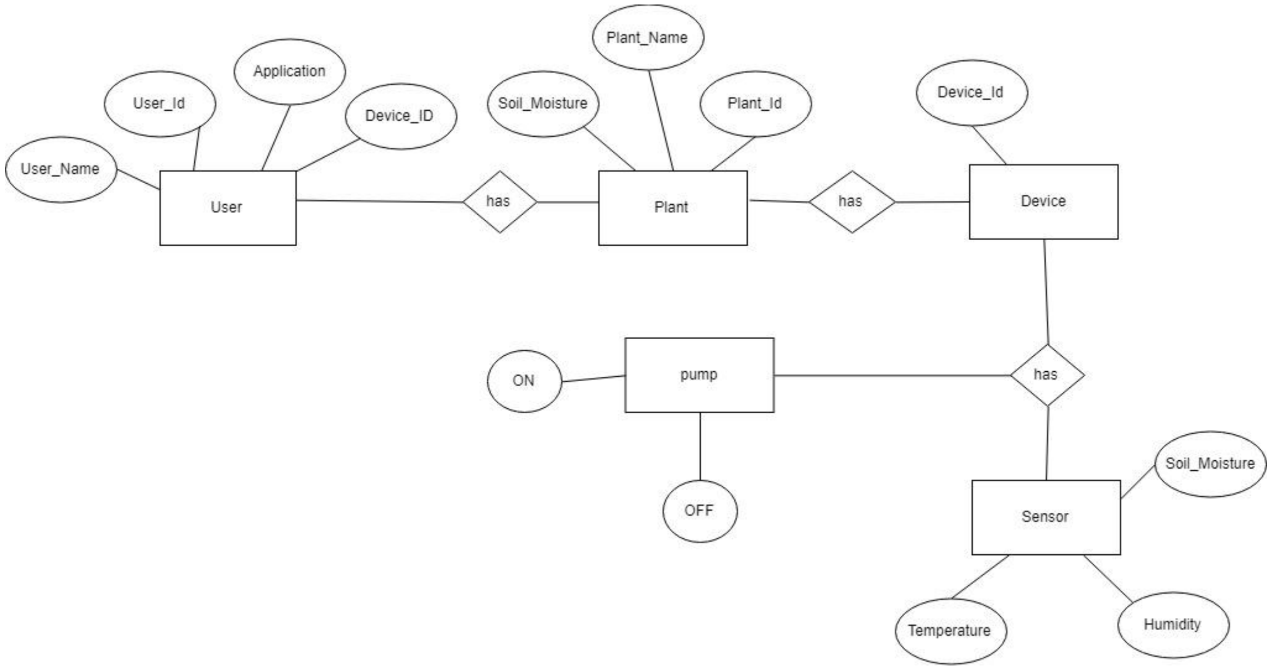


Fig 3.2 E-R Diagram for grow Organic Using IOT

3.2 Data Flow Diagram

Data Flow Diagram (DFD) provides a visual representation of the flow of information (i.e. data) within a system. By drawing a Data Flow Diagram, you can tell the information provided by and delivered to someone who takes part in system processes, the information needed in order to complete the processes and the information needed to be stored and accessed.

3.2.1 Context DFD

A context diagram could be a information multidimensional language that solely shows the highest level, otherwise called Level zero. At this level, there's just one visible method node that represents the functions of a whole system with reference to however it interacts with external entities.

The figure below shows a context information multidimensional language that's drawn for a Grow Organic using IOT System. During this System, User, Admin and Arduino are the entities can move with the system. In between the method and also the external entities, there are information flow (connectors) that indicate the existence of knowledge exchange between the entities and also the system.

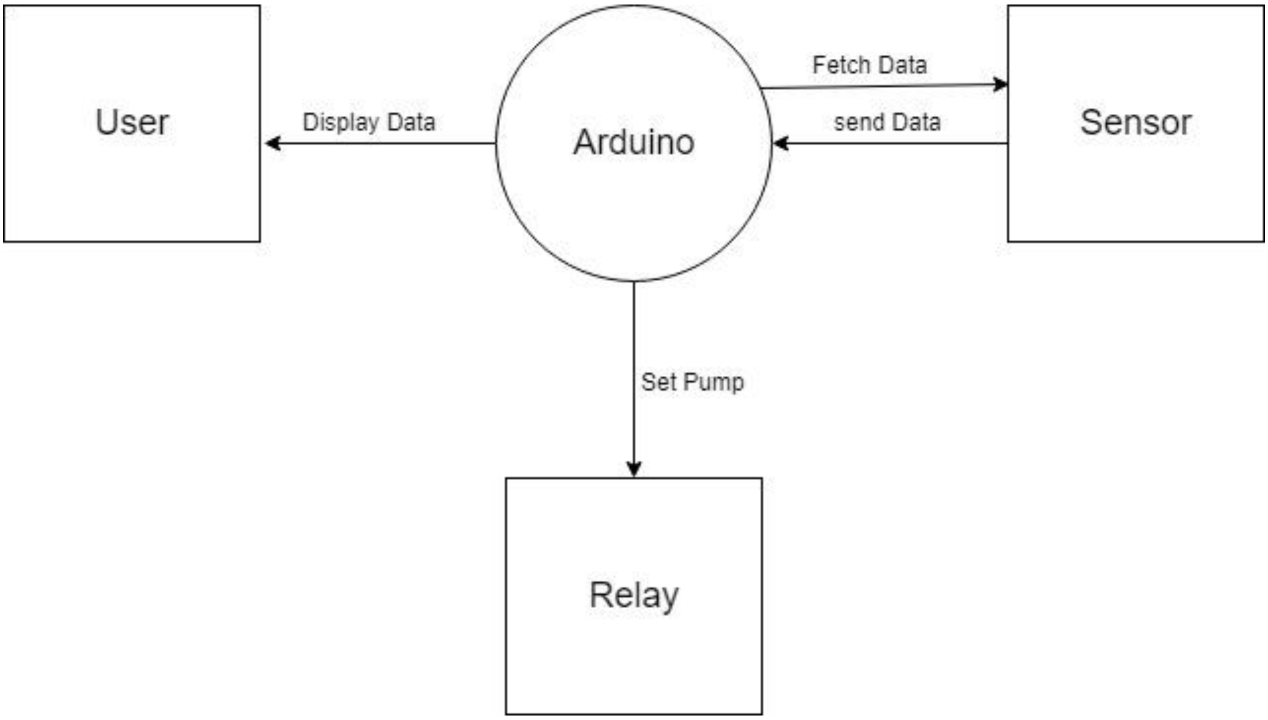


Fig 3.3.1 DFD level 0

3.3 Use Case Diagram

Use case diagrams are usually referred to as behavior diagrams used to describe a collection of actions (use cases) that some system or systems (subject) ought to or will perform unitedly with one or additional external users of the system (actors). Every use case ought to give some noticeable and valuable result to the actors or different stakeholders of the system.

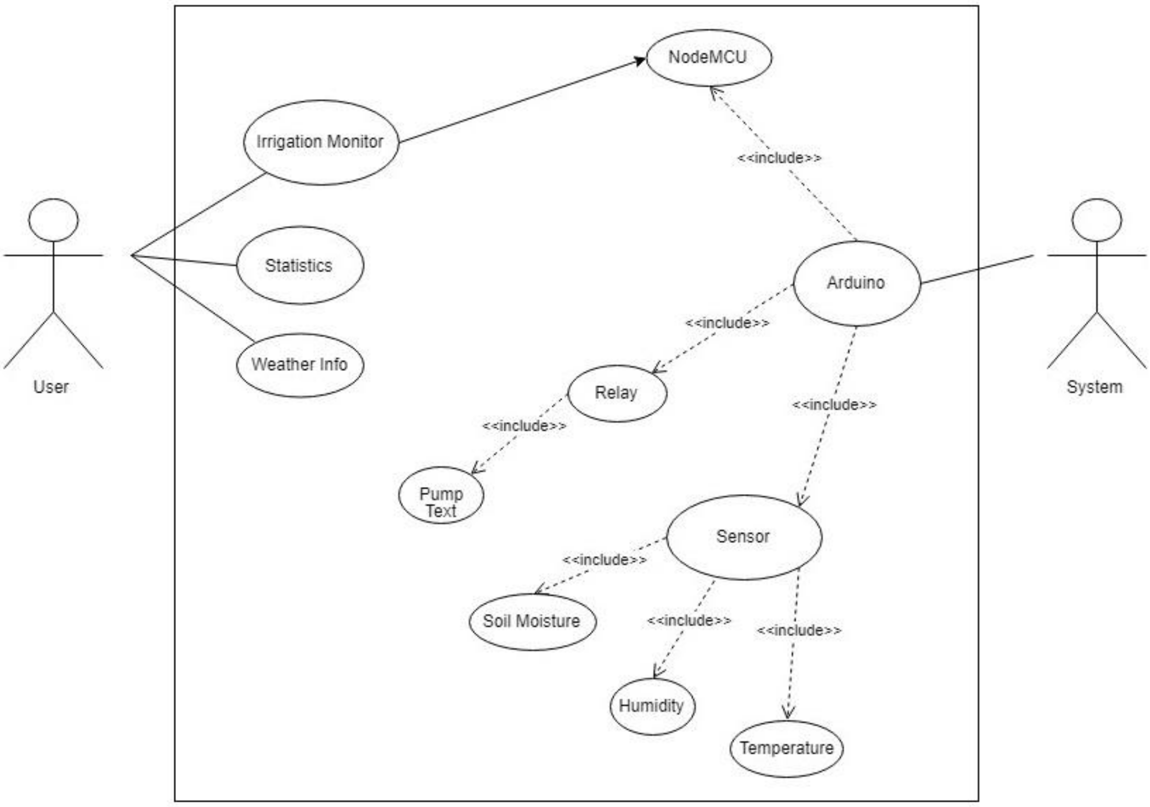


Fig 3. 4 Use Case Diagram for System

3.4 Sequence Diagram

A sequence diagram shows object interactions organized in time sequence. It depicts the objects and categories concerned within the situation and therefore the sequence of messages changed between the objects required to hold out the practicality of the situation. Sequence diagrams are usually related to use case realizations within the Logical view of the system under development.

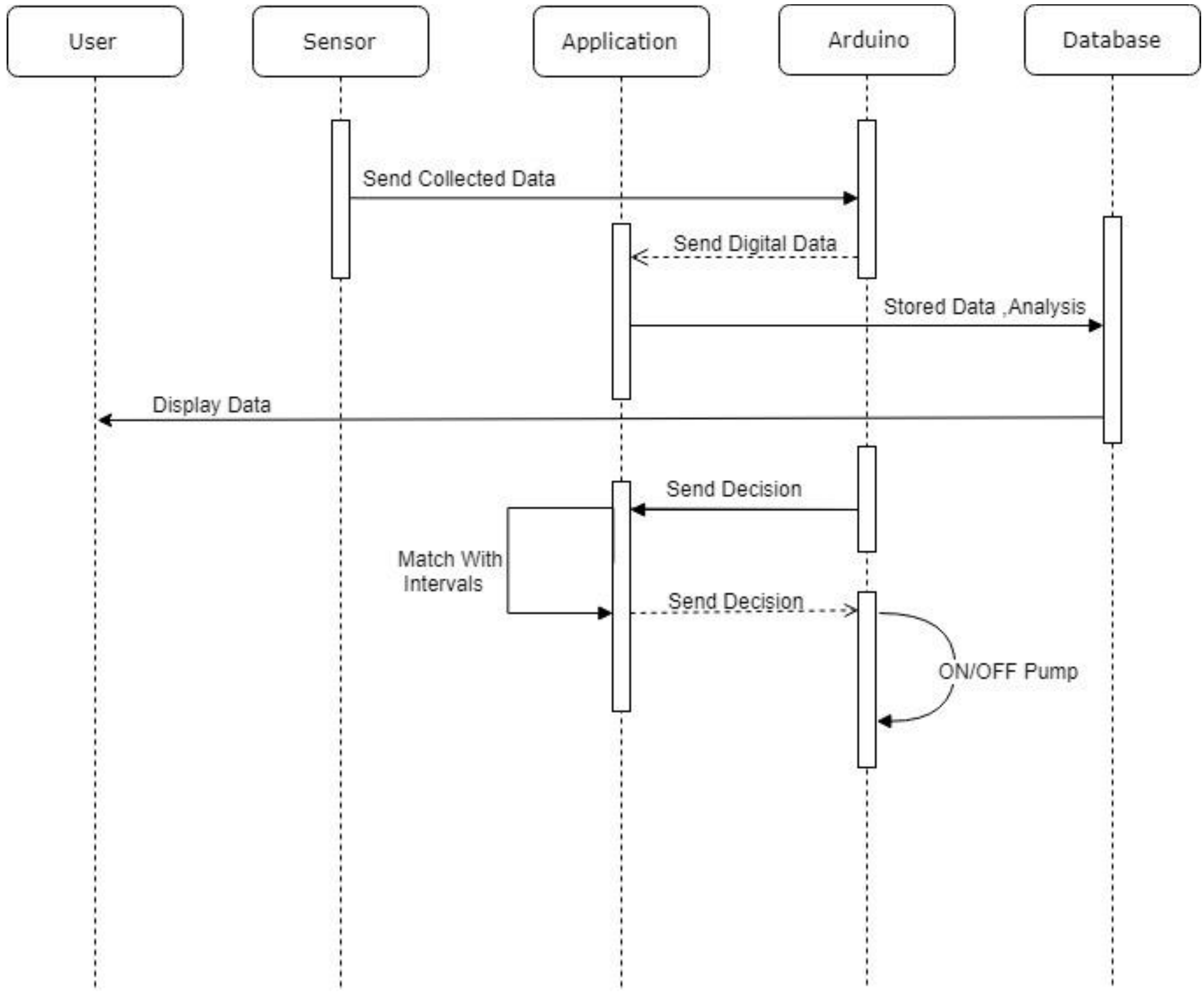


Fig 3.4 Sequence Diagram for System

3.5 Activity Diagram

The basic functions of activity diagrams are similar to alternative four diagrams. It captures the dynamic behavior of the system. Alternative four diagrams are used to show the message result one object to a different however activity diagram is used to point out message result one activity to another.

Activity may be a specific operation of the system. Activity diagrams aren't solely used for visualizing the dynamic nature of a system; however they're conjointly wont to construct the possible system by victimization forward and reverse engineering techniques.

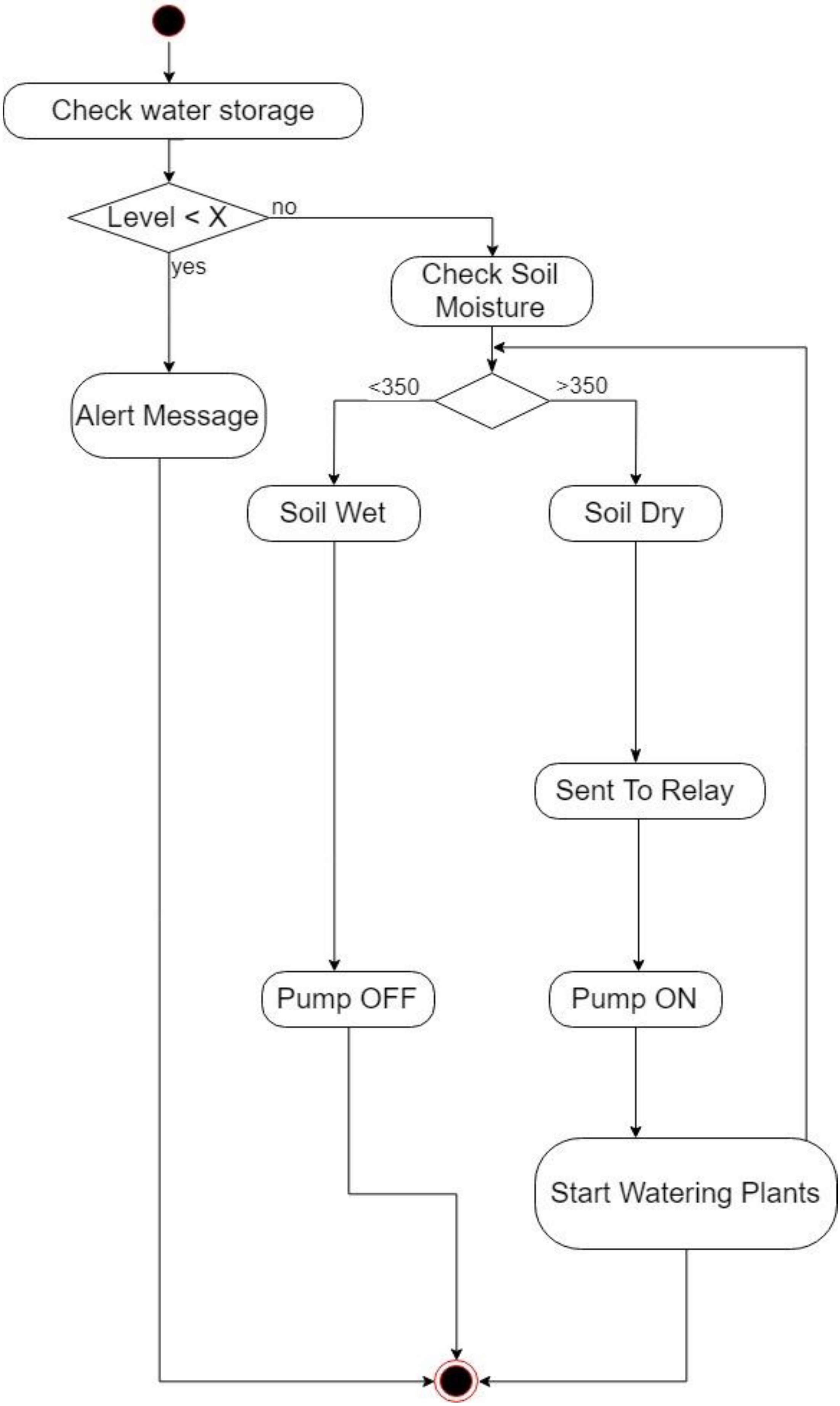


Fig 3.5 Activity Diagram for System

3.6 Class Diagram

In software engineering, a class diagram within the Unified Modeling Language (UML) may be a form of static structure diagram that describes the structure of a system by showing the system's classes, their attributes, operations (or methods), and also the relationships among objects.

The class diagram is that the main building block of object-oriented modeling. it's used for general abstract modeling of the systematic of the appliance, and for elaborated modeling translating the models into programming code. Category diagrams may also be used for information modeling.[1] The categories during a category diagram represent each the most parts, interactions within the application, and also the categories to be programmed. In the diagram, classes are represented with boxes that contain 3 compartments:

- The top compartment contains the name of the class. It's written in daring and targeted, and also the initial letter is capitalized.
- The middle compartment contains the attributes of the class. They're left-aligned and also the initial letter is lowercase.
- The bottom compartment contains the operations the class will execute. They're conjointly left-aligned and also the initial letter is lowercase.

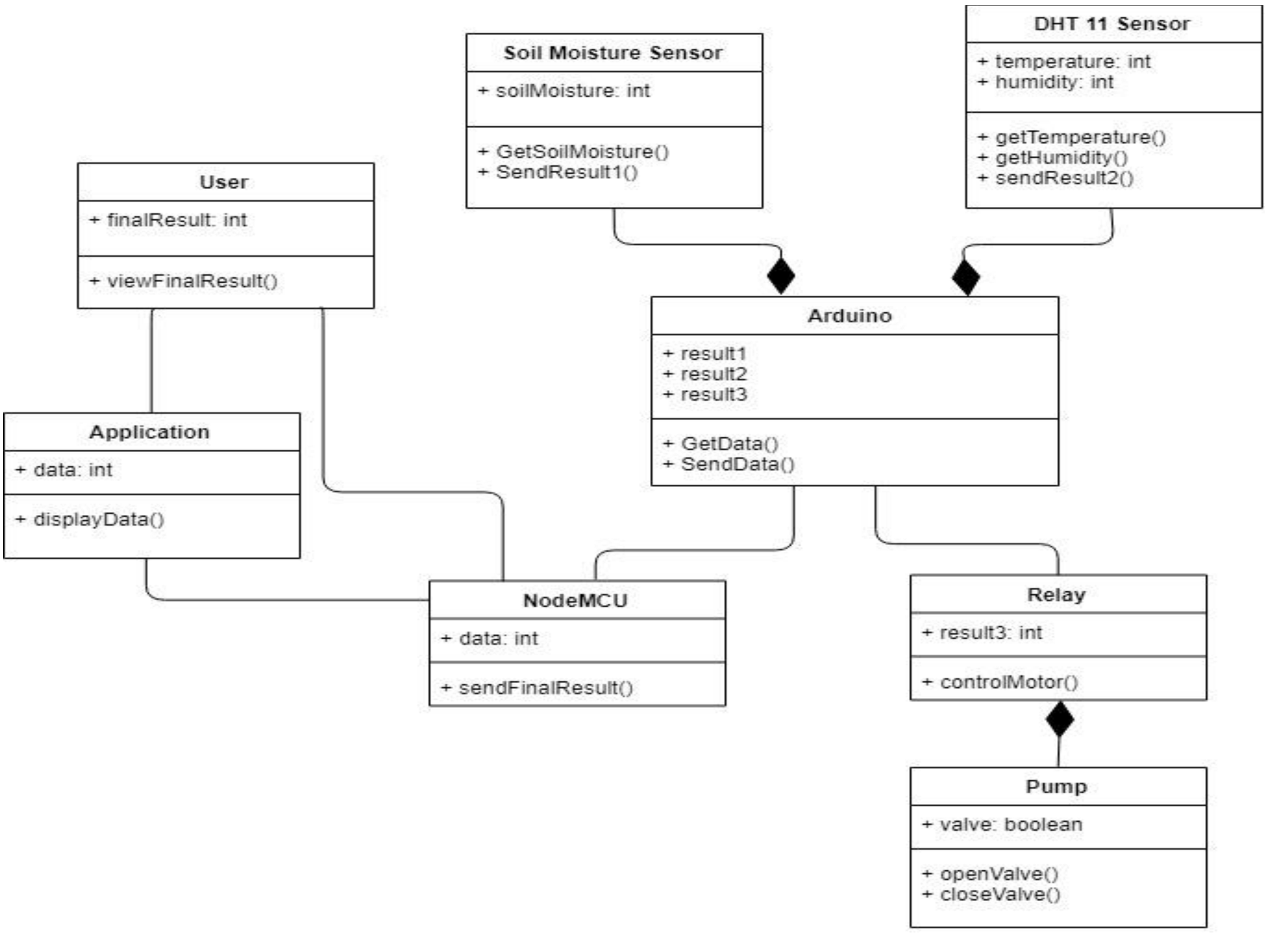


Fig 3.5 Class Diagram

Chapter 4
Design

4.1 System Flow Diagram

Our product will consist of a central control unit, sensors, and an application that work and communicate effectively with each other. See Figure for a high-level concept.

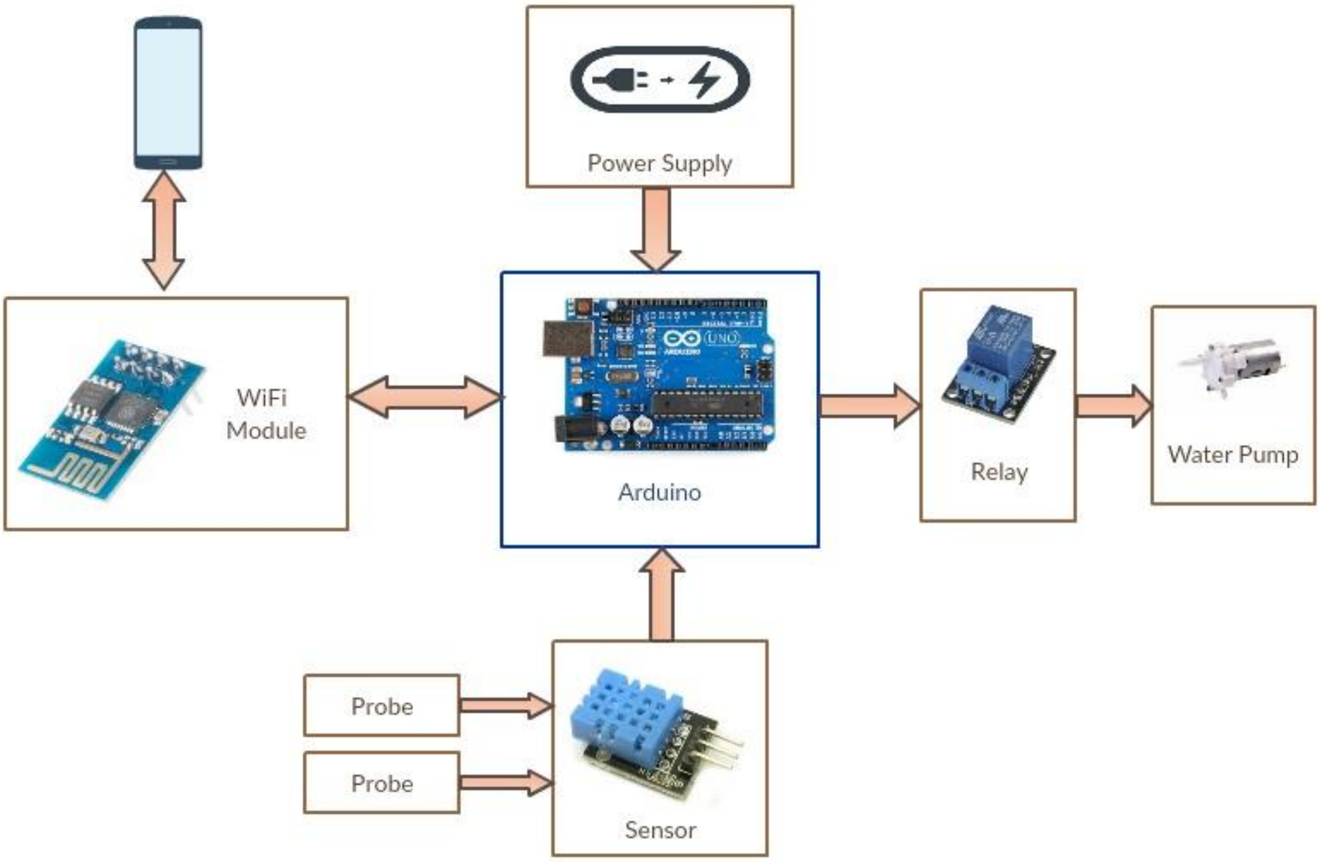


Fig 4.1 System Flow Diagram

- In this system, soil moisture sensor senses the moisture level of the soil. If soil will get dry then sensor senses low moisture level and automatically switches on the water pump to supply water to the plant. As plant get sufficient water and soil get wet then sensor senses enough moisture in soil. After which the water pump will automatically get stopped.
- The soil moisture sensor consists of two leads that are used to measure volume of water content in soil. These leads allow the current to pass through the soil and in return calculates the resistance value to measure the moisture level. If there is more water in soil then soil will conduct more electricity, means less resistance value along with high level of moisture. In the same manner if there is less water in soil then soil will conduct less electricity, means high resistance value along
- With low level of moisture. The motor pump will automatically get switched OFF once the soil moisture sensor has reached the required threshold value.

- If weather condition is such that it started raining, then the micro-controller will shut down the motor pump till raining. And after that it checks whether the soil moisture sensor has reached the threshold value or not. If it crosses the threshold value then motor pump will remain shut down otherwise it will start again automatically. This helps in saving water resource and electricity.
- Also in case, when power supply gets cut-off and motor gets switched off. It will restart again automatically when there will be availability power supply, user will have not to worry about restarting the motor pump manually.
- Also data of various sensor like- moisture sensor, temperature sensor, humidity sensor will be displayed on BOLT cloud in graphical form but due to limitation of BOLT i have only displayed one sensor data(moisture sensor data).

1.2 User interface

- The Device is very user friendly as it is an automated device that regulates the flow and control of the water without any user interface and hence becomes convenient.
- As a part of working device collects the data from the sensor and according to the requirements it allows the flow of water hence plants never get dry.
- User can also access data remotely for checking plant condition.
- User do not need physical presences and time to fulfil the water requirements of plant.

1.3 . Application Navigation

- More timely irrigation: Irrigators with automation are more inclined to irrigate when the plants need water, not when it suits the irrigator.
- Assists in the management of higher flow rates: Many irrigators are looking to increase the irrigation flow rates they receive through installing bigger channels and bay outlets. Such flow rates generally require an increase in labour as the time taken to irrigate a bay is reduced thus requiring more frequent change over. Automation allows for these higher flows to be managed without an increase in the amount of labour.
- More accurate cut-off: Automation of the irrigation system allows cut-off of water at the appropriate point in the bay. This is usually more accurate than manual checking because mistakes can occur if the operator is too late or too early in making a change of water flow.
- Real Time Feedback System: With this application irrigation is based on actual dynamic demands of the plant itself; the plant root zone is effectively reflecting all environmental factors acting on the plant. Operating within controlled parameters, the plant itself determines the degree of irrigation required. Various sensors, relative humidity sensors, rain sensors, temperature sensors etc. control the irrigation scheduling. These sensors provide feedback to the controller to control its operation
- Applicability: Almost every irrigation system can be automated. It makes sense in every region of the world as it saves time and water. Furthermore, high-tech designs allow for very efficient irrigation i.e. metering the water volumes more precisely. Once the system is optimised, labourers do not have to worry about the irrigation process and can concentrate on more important tasks.

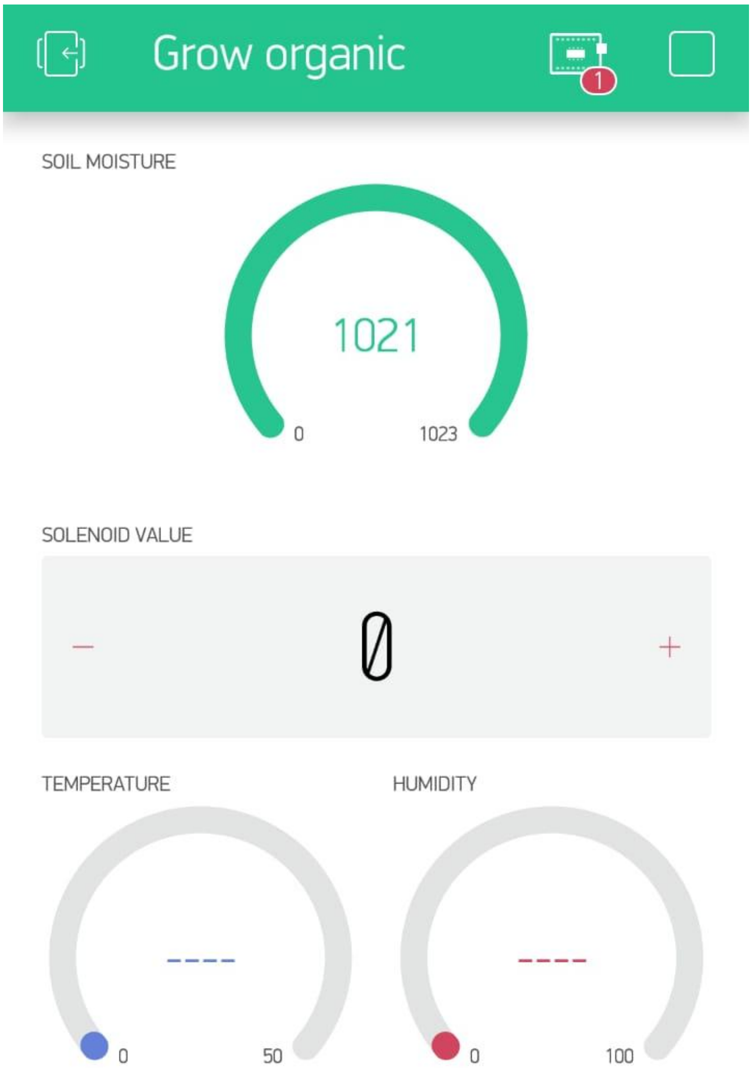


Fig 4.3 Application Screenshot

Chapter 5
Implementation Environment

5.1 Implementation environment

Sr. no.	Tools / Technology	Used for
1.	Arduino	IDE (Integrated Development Environment) for makes it easy to write code and upload it to the board
2.	IOT(Internet Of things)	It provide the management capabilities required to deliver data from IoT devices to applications
3.	Android	For Making application
4.	sensor	For fetching real time data
5.	Fritzing	For making circuit diagram

Table 5.1 List of Tool/Technology Used

5.1.1 Arduino IDE

The Arduino integrated development environment (IDE) is a cross-platform application (for Windows, macOS, Linux) that is written in the programming language Java. It is used to write and upload programs to Arduino compatible boards, but also, with the help of 3rd party cores, other vendor development boards.

The source code for the IDE is released under the GNU General Public License, version 2.The Arduino IDE supports the languages C and C++ using special rules of code structuring. The Arduino IDE supplies a software library from the Wiring project, which provides many common input and output procedures. User-written code only requires two basic functions, for starting the sketch and the main program loop, that are compiled and linked with a program stub main() into an executable cyclic executive program with the GNU toolchain, also included with the IDE distribution. The Arduino IDE employs the program avrdude to convert the executable code into a text file in hexadecimal encoding that is loaded into the Arduino board by a loader program in the board's firmware.

- Operating system: Windows, macOS, Linux
- Stable release: 1.8.8 / 7 December 2018; 2 months ago
- Developer(s): Arduino Software
- License: LGPL or GPL license
- Written in: Java, C, C++
- Platforms: x86-32 (32 bit Intel x86), x86-64, ARM architecture

Fig 5.1.1 Arduino IDE Screenshot

5.1.2 IOT (Internet Of things)

The Internet of things (IoT) is the extension of Internet connectivity into physical devices and everyday objects. Embedded with electronics, Internet connectivity, and other forms of hardware (such as sensors), these devices can communicate and interact with others over the Internet, and they can be remotely monitored and controlled.

The definition of the Internet of things has evolved due to convergence of multiple technologies, real-time analytics, machine learning, commodity sensors, and embedded systems. Traditional fields of embedded systems, wireless sensor networks, control systems, automation (including home and building automation), and others all contribute to enabling the Internet of things. In the consumer market, IoT technology is most synonymous with products pertaining to the concept of the "smart home", covering devices and appliances (such as lighting fixtures, thermostats, home security systems and cameras, and other home appliances) that support one or more common ecosystems, and can be controlled via devices associated with that ecosystem, such as smartphones and smart speakers.

The IoT concept has faced prominent criticism, especially in regards to privacy and security concerns related to these devices and their intention of pervasive presence.

5.1.3 Sensor

In the broadest definition, a sensor is a device, module, or subsystem whose purpose is to detect events or changes in its environment and send the information to other electronics, frequently a computer processor. A sensor is always used with other electronics.

Sensors are used in everyday objects such as touch-sensitive elevator buttons (tactile sensor) and lamps which dim or brighten by touching the base, besides innumerable applications of which most people are never aware. With advances in micromachinery and easy-to-use microcontroller platforms, the uses of sensors have expanded beyond the traditional fields of temperature, pressure or flow measurement,[1] for example into MARG sensors. Moreover, analog sensors such as potentiometers and force-sensing resistors are still widely used. Applications include manufacturing and machinery, airplanes and aerospace, cars, medicine, robotics and many other aspects of our day-to-day life.

5.1.4 Android Studio

Android Studio is the official integrated development environment (IDE) for Google's Android operating system, built on JetBrains' IntelliJ IDEA software and designed specifically for Android development. It is available for download on Windows, macOS and Linux based operating systems. It is a replacement for the Eclipse Android Development Tools (ADT) as the primary IDE for native Android application development.

Android Studio was announced on May 16, 2013 at the Google I/O conference. It was in early access preview stage starting from version 0.1 in May 2013, then entered beta stage starting from version 0.8 which was released in June 2014. The first stable build was released in December 2014, starting from version 1.0. The current stable version is 3.3, which was released in January 2019.

5.1.4 Fritzing

Fritzing is an open-source initiative to develop amateur or hobby CAD software for the design of electronics hardware, to support designers and artists ready to move from experimenting with a prototype to building a more permanent circuit. It was developed at the University Of Applied Sciences Of Potsdam.

The software is created in the spirit of the Processing programming language and the Arduino microcontroller and allows a designer, artist, researcher, or hobbyist to document their Arduino-based prototype and create a PCB layout for manufacturing. The associated website helps users share and discuss drafts and experiences as well as to reduce manufacturing costs.

- License: GNU GPL v3 (software); CC BY-SA (component images)
- Operating system: Mac OS X, Unix, Windows
- Stable release: 0.9.3b / June 3, 2016; 2 years ago
- Developer(s): Interaction Design Lab Potsdam
- Programming language: C++

5.2 Security Features

Automated information systems security includes consideration of all computer hardware and software functions, characteristics and features; operational procedures; accountability procedures; and access controls at the central computer facility, remote computer, and terminal facilities; management constraints; physical structures and devices, such as computers, transmission lines, and power sources; and personnel and communications controls needed to provide an acceptable level of risk for the automated information system and for the data and information contained in the system. Automated information systems security also includes the

totality of security safeguards needed to provide an acceptable protection level for an automated information system and for the data handled by an automated information system.
As it is home based irrigation system security is not a concern.

5.3 Coding Standards

- Coding conventions are a set of guidelines for a specific programming language that recommend programming style, practices, and methods for each aspect of a program written in that language. These conventions usually cover file
- Organization, indentation, comments, declarations, statements, white space, naming conventions, programming practices, programming principles, programming rules of thumb, architectural best practices, etc. These are guidelines for software structural quality. Software programmers are highly recommended to follow these guidelines to help improve
- The readability of their source code and make software maintenance easier. Coding conventions are only applicable to the human maintainers and peer reviewers of a software project. Conventions may be formalized in a documented set of rules that an entire team or company follows, or may be as informal as the habitual coding practices of an individual. Coding conventions are not enforced by compilers.
- Code conventions are important to programmers for a number of reasons
- 40%–80% of the lifetime cost of a piece of software goes to maintenance.
- Hardly any software is maintained for its whole life by the original author.
- Code conventions improve the readability of the software, allowing engineers to understand new code more quickly and thoroughly.
- If you ship your source code as a product, you need to make sure it is as well packaged and clean as any other product you create
- There are a large number of coding conventions; Common coding conventions may cover the following areas:
    - Comment Conventions
    - Indent Style Conventions
    - Line Length Conventions
    - Naming Conventions
    - Programming Principles
    - Programming Style Conventions

5.4 Business Logic Layer

- Business Logic: The overall set of rules that determine how the data will be stored or manipulated in a business or application domain.
- The Model-View-Controller (MVC): An architectural pattern that separates an application into three main logical components: the model, the view, and the controller. The data associated with the underlying business logic is represented by Model. The UI

logic of the application is represented by View layer. The Service layer or Controllers act as an interface between Model and View components to process all the business logic and incoming requests, manipulate data using the Model component and interact with the Views to render the final output.

- The service layer or controller in fact represents the core business logic for the data manipulation represented by CRUD operations.
- For larger web applications the best approach is to keep minimal amount of code in each layer and a separate layer is added centered around the business logic. This layer is termed as a business logic layer. For smaller applications the database objects itself could contain the business logic.

Chapter 6
Testing

6.1 Testing Plan

System testing is a critical element of quality assurance and represents the ultimate review of analysis, design and coding. Test case design focuses on a set of techniques for the creation of test because that meet overall testing objective. When a system is developed it is hoped that it performs properly. The main purpose of testing an information system is to find the errors and correct them. The scope of system testing should include both manual and computerized operations. System testing is comprehensive evaluation of the programs, manual procedures, computer operations and controls.

System testing is the process of checking whether the developed system is working according to the objective and requirement. All testing is to be conducted in accordance to the test conditions specified earlier. This will ensure that the test coverage meets the requirements and that testing is done in a systematic manner. The process of analyzing the software item to detect the differences between existing or required condition and evaluate the features of the software items. The thorough testing of the system before release of the software needs to be done vide the various test cases and modes so that the software becomes devoid of bugs and uses minimum space requirements as well as minimum time to perform. The test cases were selected beforehand with expected results defined and actual results recorded for comparison. The selection of test cases is done vide ―White Box Testing‖ technique to check the internal programming logic and efficiency and vide ‖Black Box Testing‖ technique to check software requirement fulfillment with intension of finding maximum number of errors with minimum effort and time. Although test cases are a design by considering the cyclamate complexity, conditional test, still the software code is not in its optional form, as all other possible alternative parts in the software are not considered. At the integration level, the software will be passing to the third party tests which would further enhance the software optimality and efficiency.

6.2 Testing Strategy

- **Unit Testing**
  The unit testing is performed to test the validity of the individual units. This is done in the coding phase with the interactive testing. Thus, it itself constitutes a majority of functionality test for each logical unit.
- **Integrity Testing:**
  When all the development of all the units or modules is completed and integrated the integrity test phase is started. In this phase the interface between the modules are tested. This phase basically verifies whether inter module exchange of information and events are as per required system behavior.
- **Validation testing:**
  Tests were performed to find conformity with their requirements. Plans and procedures were

designed to ensure that all functional requirements are satisfied. The software was alpha-tested. There are two goals in preparing test plans. Firstly, a properly detailed test plan demonstrates that the program specifications are understood completely. Secondly, the test plan is used during program testing to prove the correctness of the program

6.3 Testing Methods

6.3.1 Black box testing:

The method of Black Box Testing is used by the software engineer to derive the required results of the test cases:

1. Black Box Testing alludes to test that are conducted at the software interface.
2. A Black Box Test examines some fundamental aspect of a system with little regard for the internal logic structure of the software.
3. A limited number of important logical paths can be selected and exercised.
4. Important data structure can be probed for validity.

Black box testing was performed to find errors in the following categories: -

- Incorrect or missing functions
- Graphics error.
- Errors in data in binary format.
- Error in data in integer format. File error.
- Pointer error.
- Memory access error.
- Variable error.
- Performance error

6.3.2 WHITE BOX TESTING:

White Box Testing is sometimes called Glass Box Testing. Using White Box Testing methods, the software engineer can derive the following test cases:

1. Guarantee that all independent paths within a module have been exercised at least once
2. Exercise all logical decisions on their true and false sides.
3. Execute all loops at their boundaries and within their operational bounds.
4. Exercise internal data structures to ensure the validity.

In White Box Testing efforts were made to handle the following: -

- Number of input parameters equal to number of arguments.
- Parameters and arguments attributes match

- Number of attributes and order of arguments to build in functions correct.
- Any references to parameters not associated to build in functions correct.
- Input only arguments altered.
- Global variable definition consistent across module.
- Files attributes correct.
- Format specifications matches I/O specification.
- Files opened before use.

6.4 Test Cases

Sr No	Soil Condition	Moisture Content	Relay Status	Water Pump status	Test Case status
1	Dry	<1000 >600	ON	ON	True
2	Damp	<600 >350	ON	ON	True
3	Wet	<350	OFF	OFF	True

Table 6.4 Pump supply Test Case

Chapter 7

Application Screenshot

7.1 Server side with description:

7.1.1 Components Used

- Arduino Uno
- Soil Moisture Sensor
- Temperature and Humidity Sensor
- NodeMCU
- Breadboard
- Relay
- Pump
- Power supply

Step 1: Connect the Arduino Uno with Soil moisture Sensor. The soil moisture detects the level of moisture in the soil. On the basis of varying value, water is supplied.

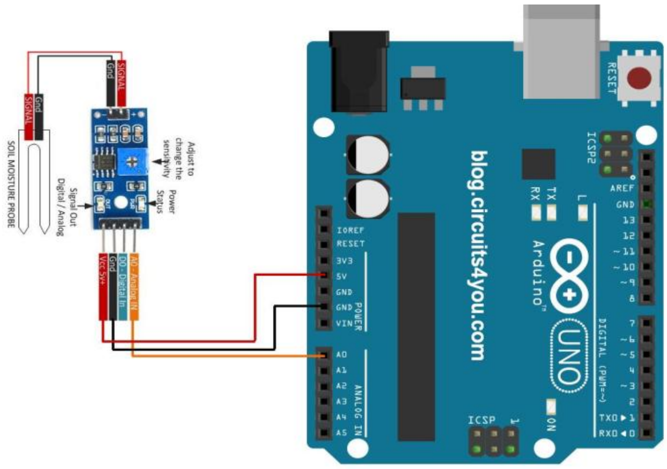


Fig 7.1.1 Circuit Diagram of Soil Moisture Connection

Step 2: connect the Relay with Arduino. The connection of the relay module controls the triggering of ON/OFF events of the water pump.

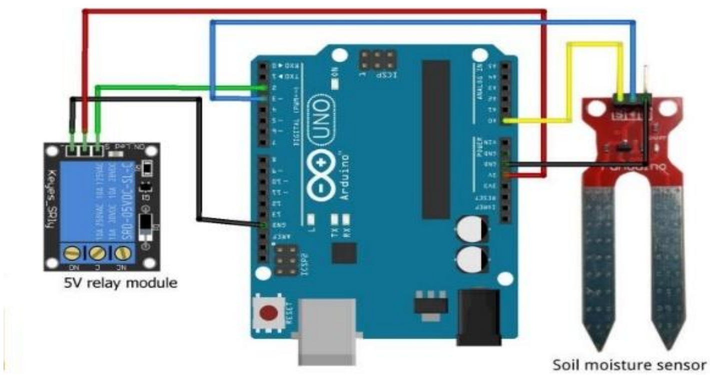


Fig 7.1.2 Circuit Diagram of Relay connection to Arduino

Step 3: connect the pump with relay. The water pump does fetching and provide a successful water supply using inlet and outlet.

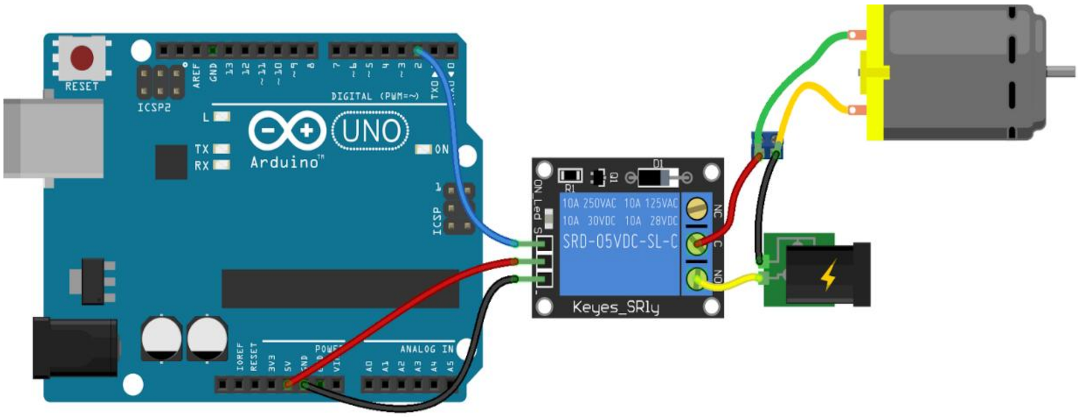


Fig 7.1.3Diagram of Relay connection to pump

Step 4: connect the temperature Humidity Sensor with Arduino. The collection of humidity and temperature value using DTH11 sensor. Using those value, its prediction of upcoming water supply.

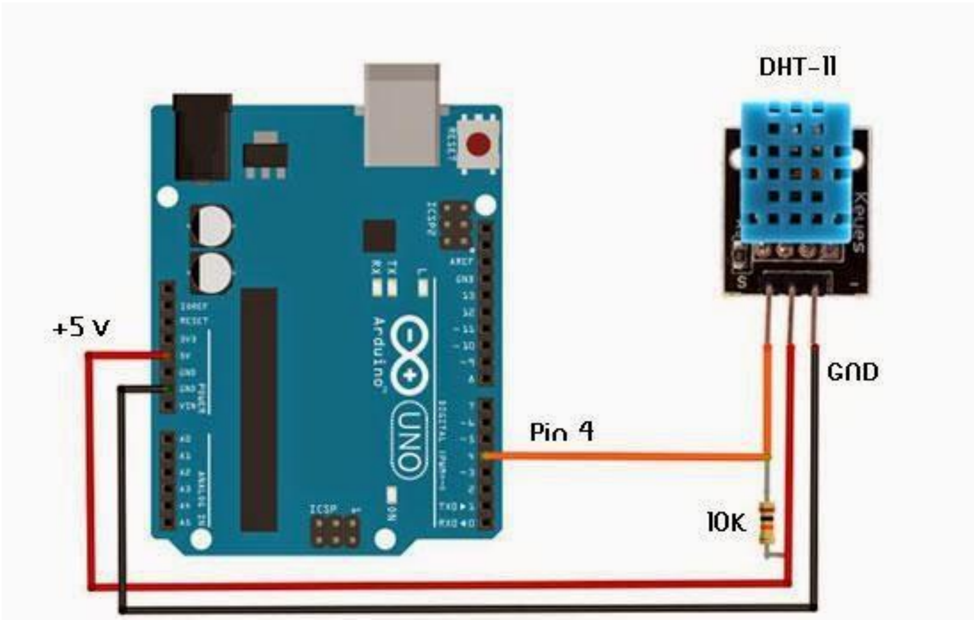


Fig 7.1.4 Diagram of DHT11 connection to Arduino

Step 5: connect the NodeMCU with Arduino. This module helps to provide an interface between Arduino and NodeMCU. Also, the wifi connection is established through this module.

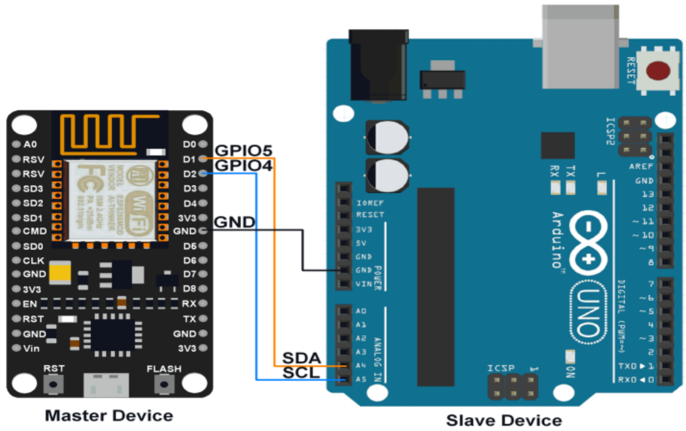


Fig7.1.5 Diagram of NodeMCU connection to Arduino

Step 6: integrate all the plants with Arduino using breadboard. The complete integrated circuit of all the provided components.

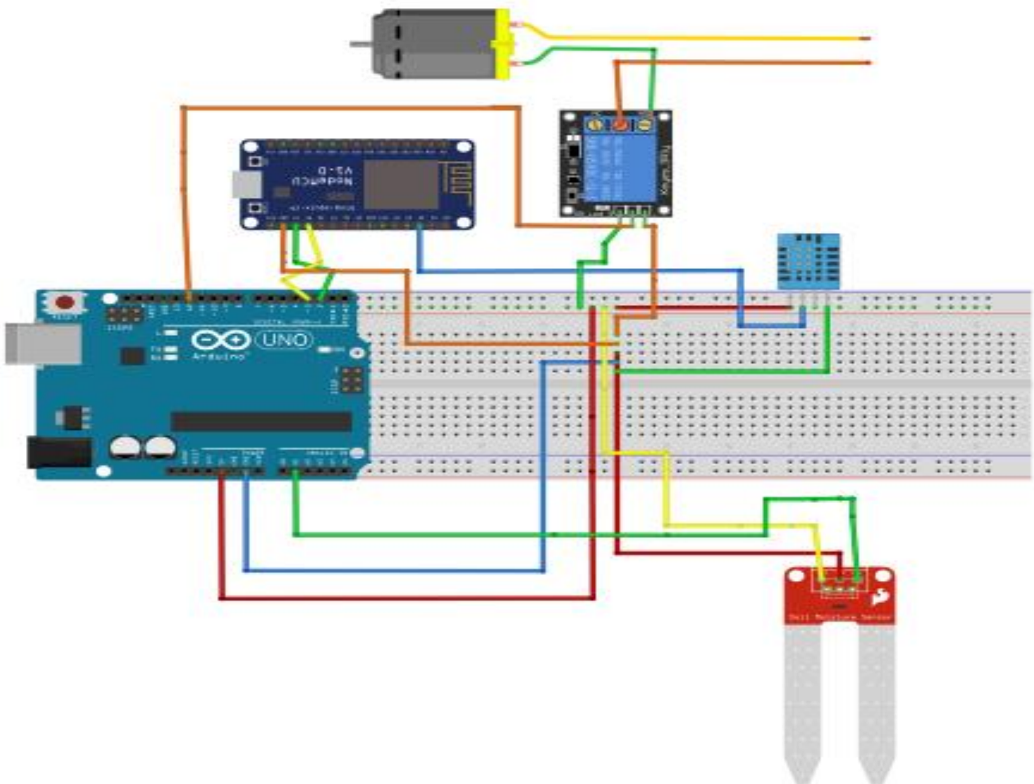


Fig 7.1.6 System circuit diagram

Step 7: Upload code to Arduino

1) The collection of soil moisture sensor data is fetched and that data is displayed in application in gauge

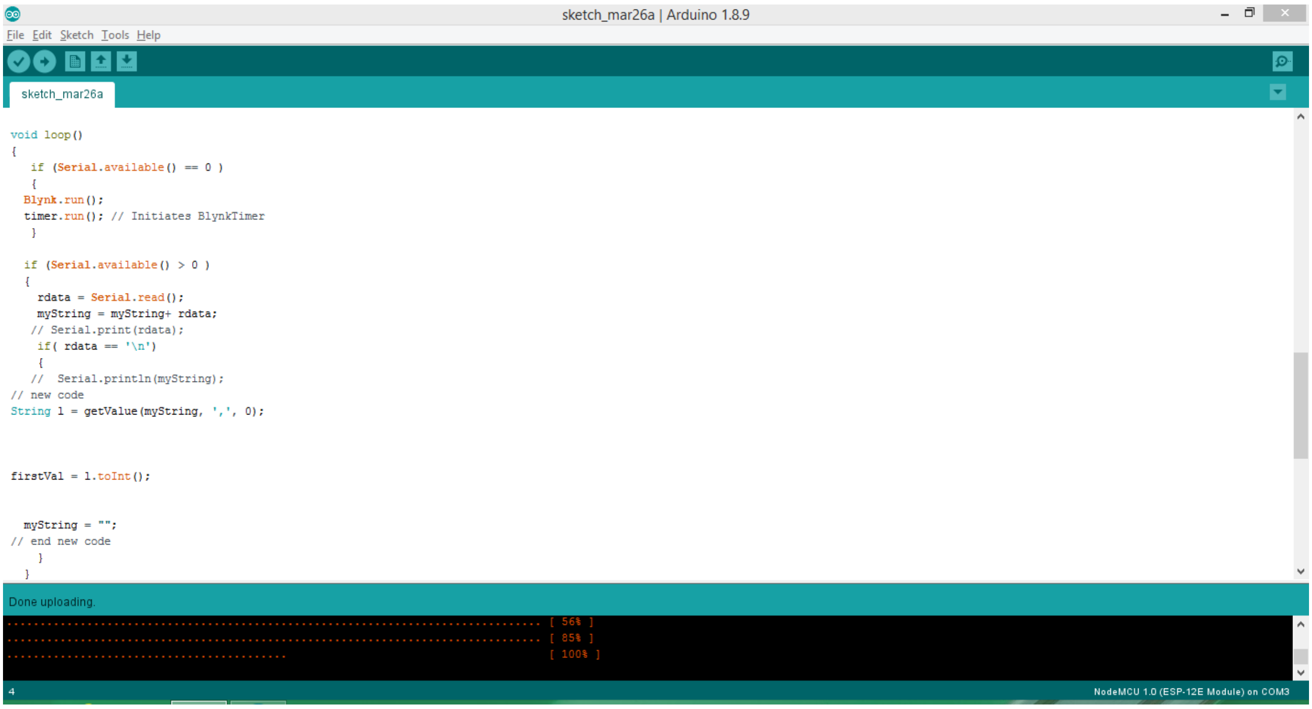


Fig 7.1.7 NodeMCU code 1

2) Adding Of DTH11 library file and code for displaying to application

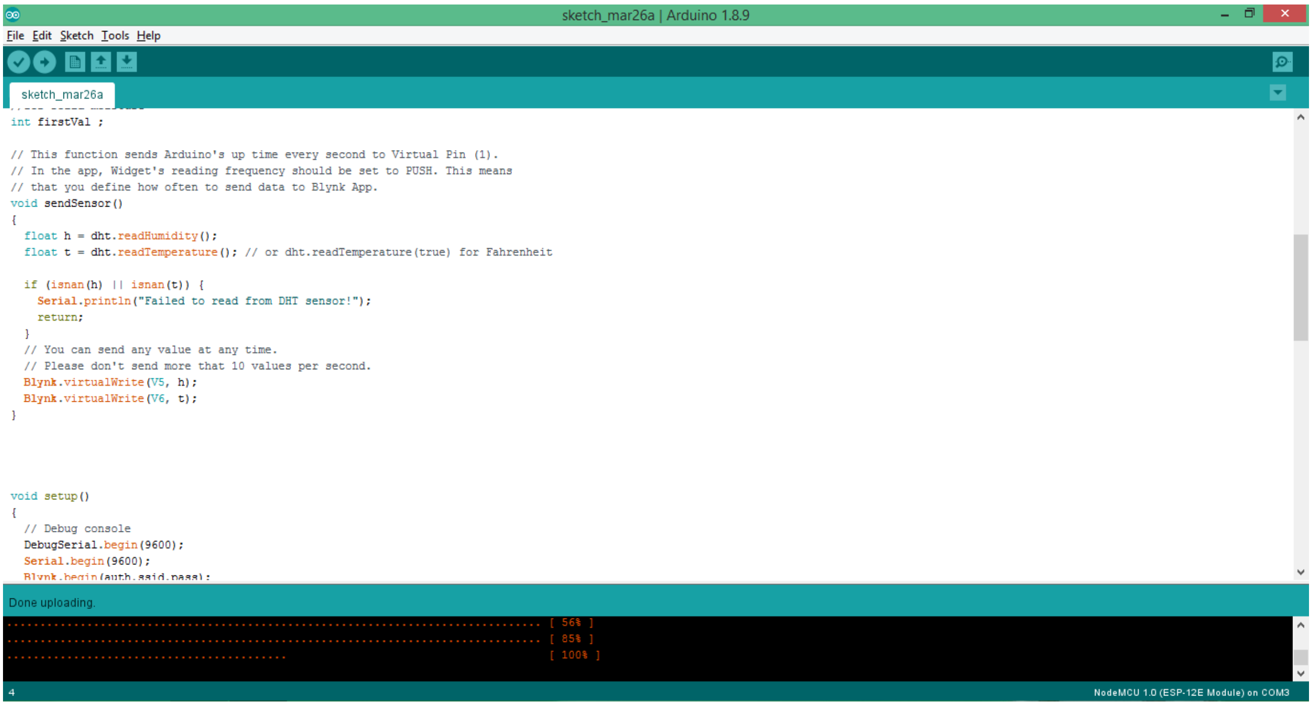
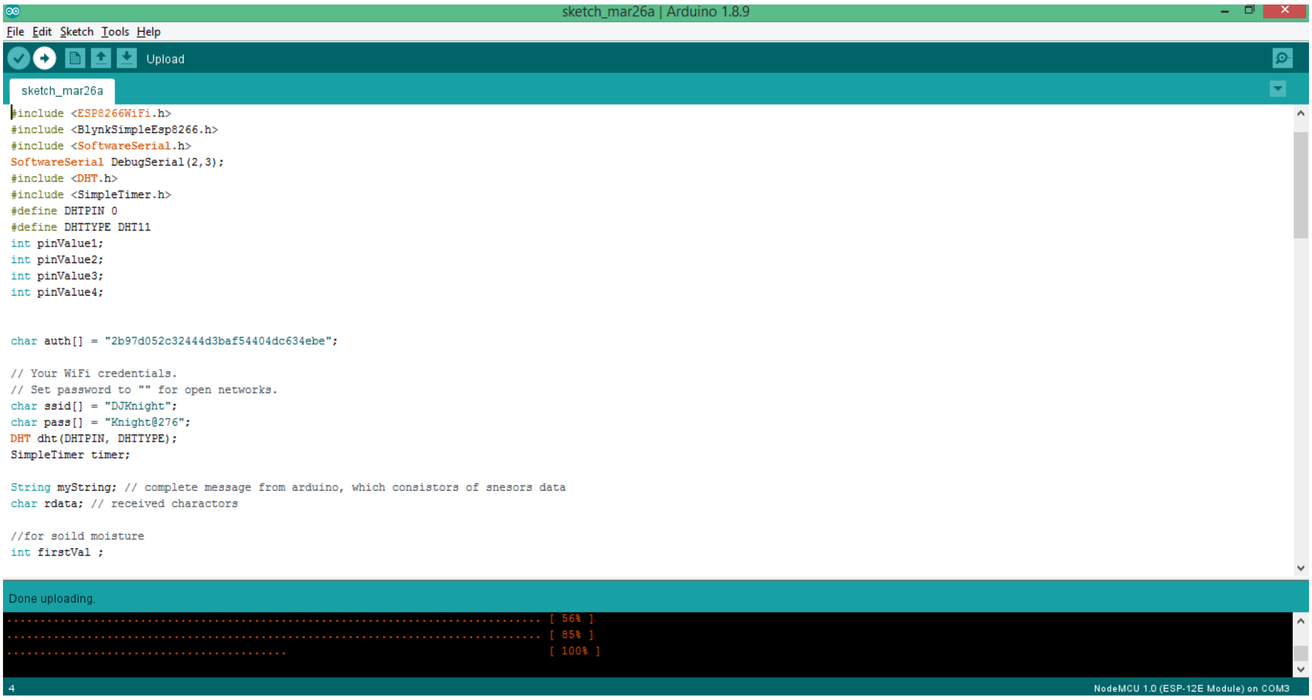


Fig 7.1.8 NodeMCU code 2

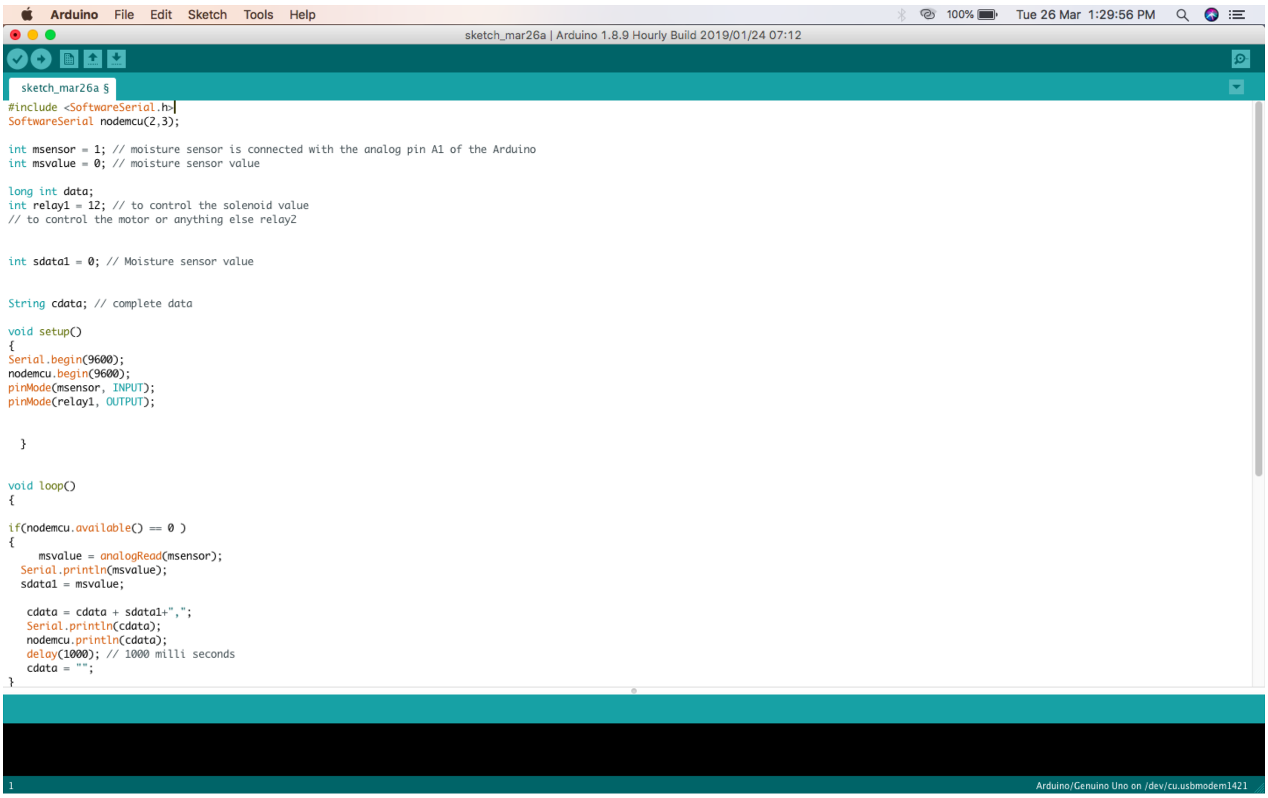
3) Include libraries and define pin in nodeMCU



```
#include <ESP8266WiFi.h>
#include <BlynkSimpleEsp8266.h>
#include <SoftwareSerial.h>
SoftwareSerial DebugSerial(2,3);
#include <DHT.h>
#include <SimpleTimer.h>
#define DHTPIN 0
#define DHTTYPE DHT11
int pinValue1;
int pinValue2;
int pinValue3;
int pinValue4;


char auth[] = "2b97d052c32444d3baf54404dc634ebe";

// Your WiFi credentials.
// Set password to "" for open networks.
char ssid[] = "DJKnight";
char pass[] = "Knight@276";
DHT dht(DHTPIN, DHTTYPE);
SimpleTimer timer;

String myString; // complete message from arduino, which consistors of snesors data
char rdata; // received charactors

//for soild moisture
int firstVal ;
```

Fig 7.1.9 NodeMCU code 3

4) Code for connecting NodeMCU with Arduino and initialize methods to start the connection between them



```
#include <SoftwareSerial.h>
SoftwareSerial nodemcu(2,3);

int msensor = 1; // moisture sensor is connected with the analog pin A1 of the Arduino
int msvalue = 0; // moisture sensor value

long int data;
int relay1 = 12; // to control the solenoid value
// to control the motor or anything else relay2


int sdata1 = 0; // Moisture sensor value


String cdata; // complete data

void setup()
{
Serial.begin(9600);
nodemcu.begin(9600);
pinMode(msensor, INPUT);
pinMode(relay1, OUTPUT);


  }


void loop()
{

if(nodemcu.available() == 0 )
{
     msvalue = analogRead(msensor);
  Serial.println(msvalue);
  sdata1 = msvalue;

   cdata = cdata + sdata1+",";
   Serial.println(cdata);
   nodemcu.println(cdata);
   delay(1000); // 1000 milli seconds
   cdata = "";
}
```

Fig 7.1.10 Arduino code 1

5) Implementing Core Algorithm

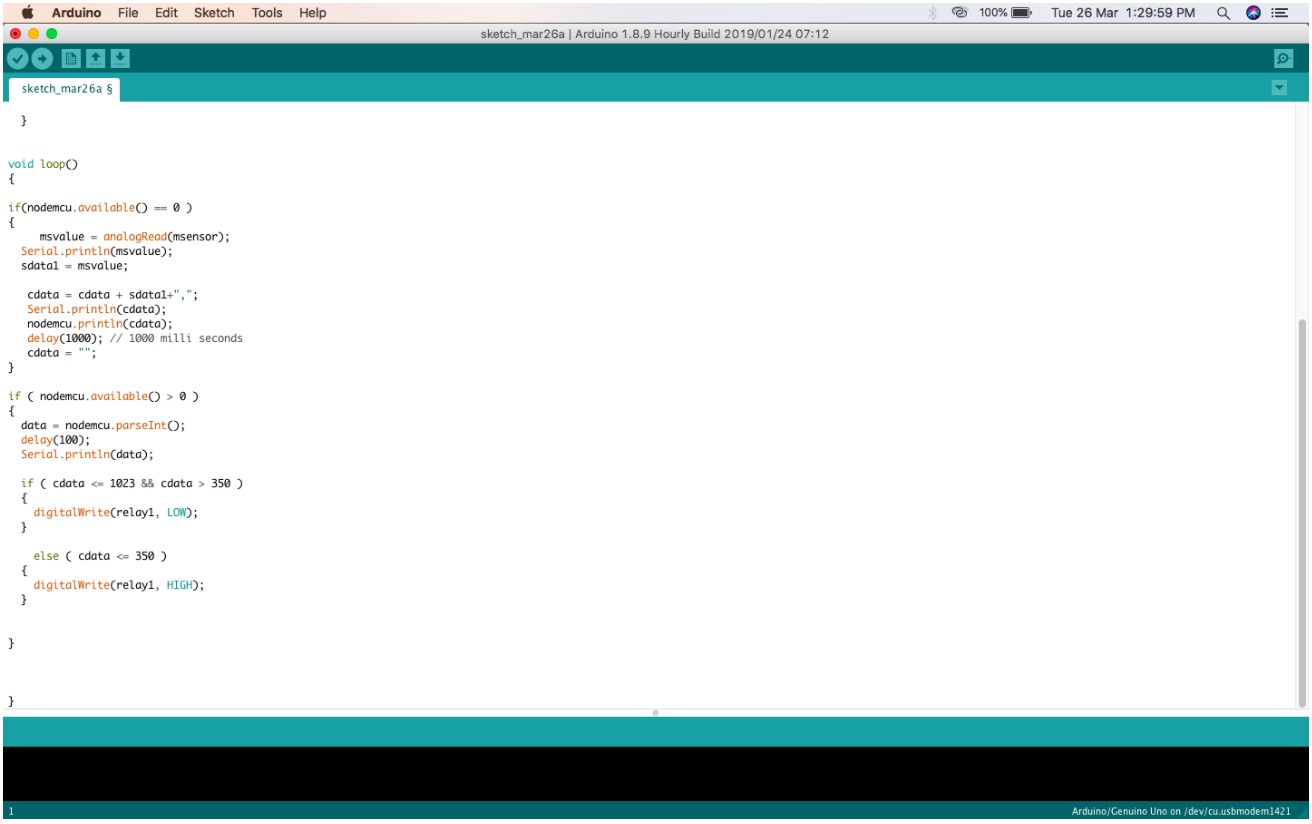


Fig 7.1.11 Arduino code 2

Step 8: Testing and Implementation

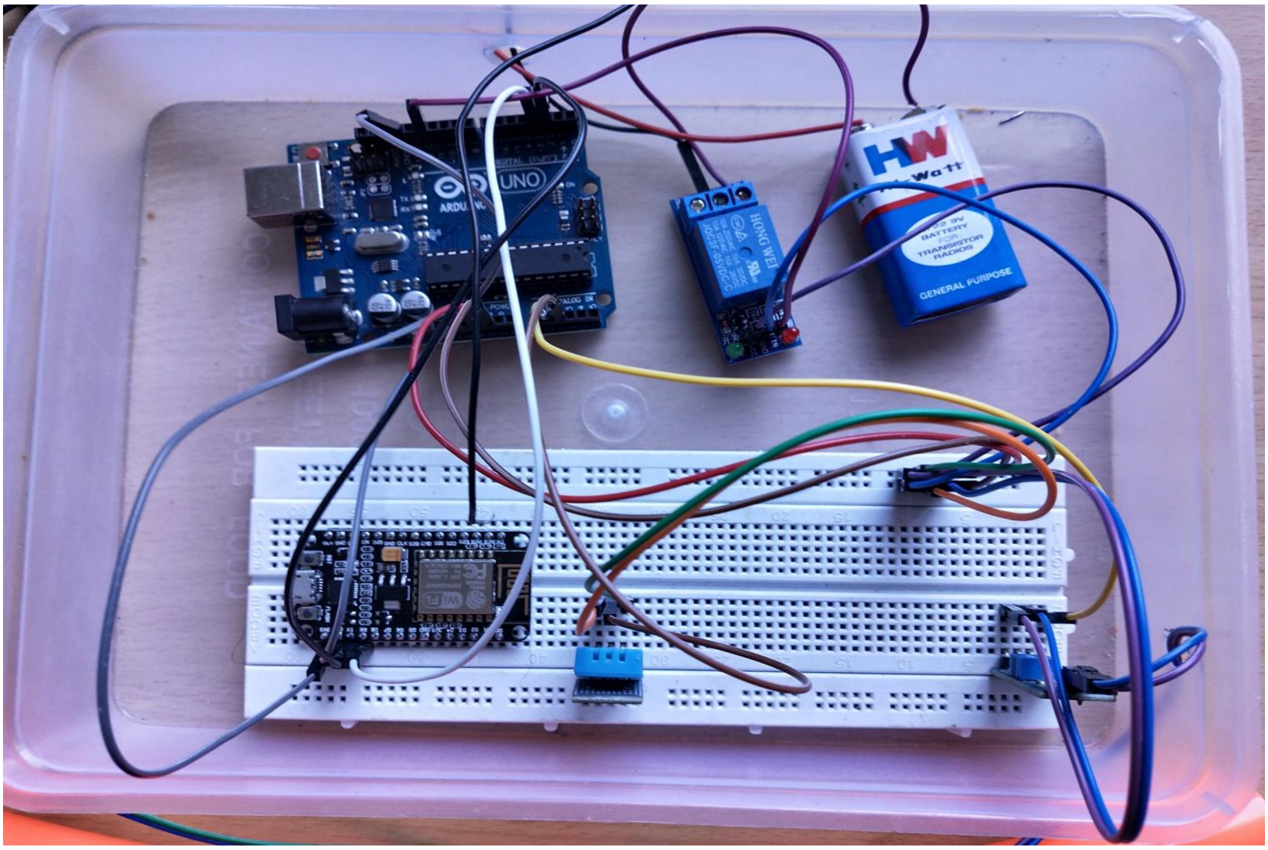


Fig 7.1.12 Implementation of control parts

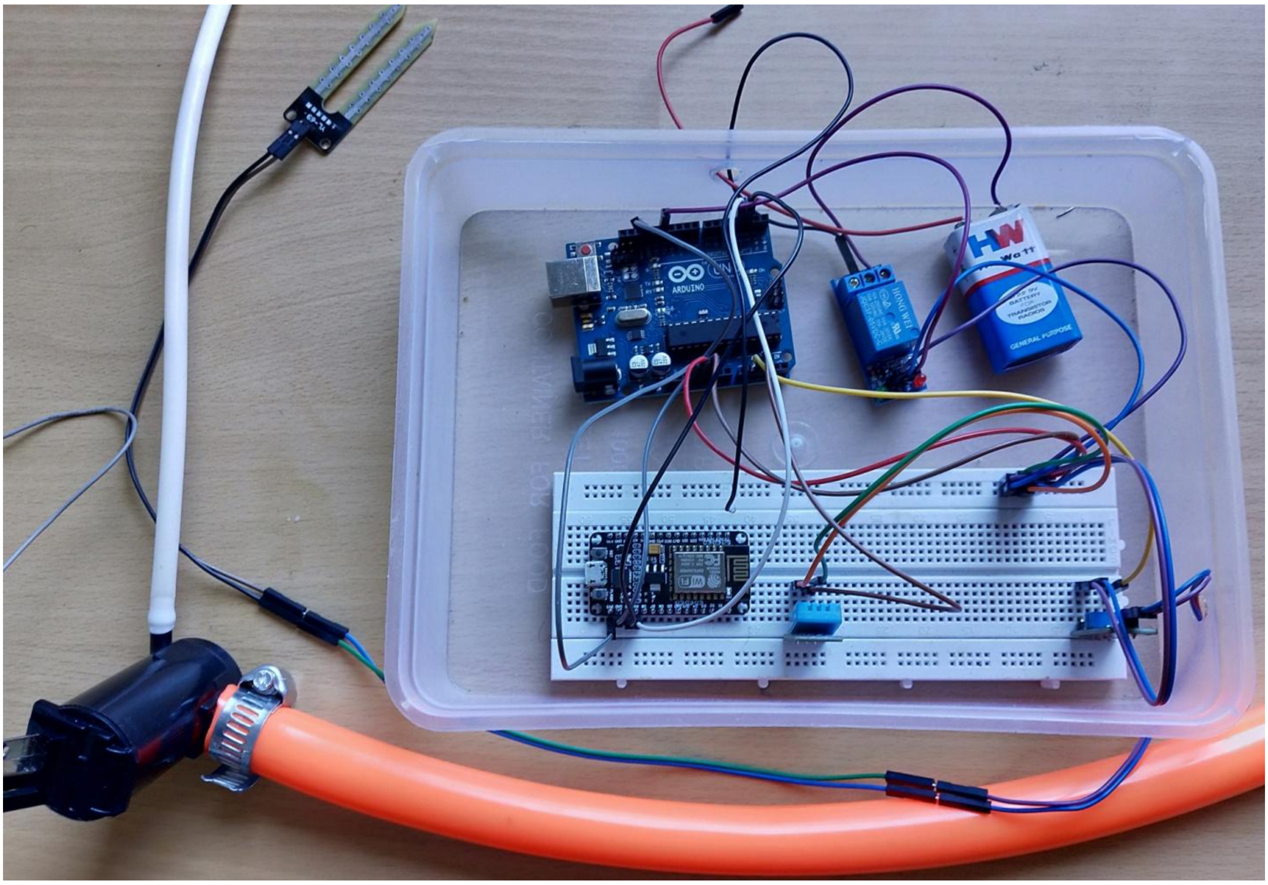


Fig 7.1.13 Complete Circuit Implementation

7.2 Client side with description:

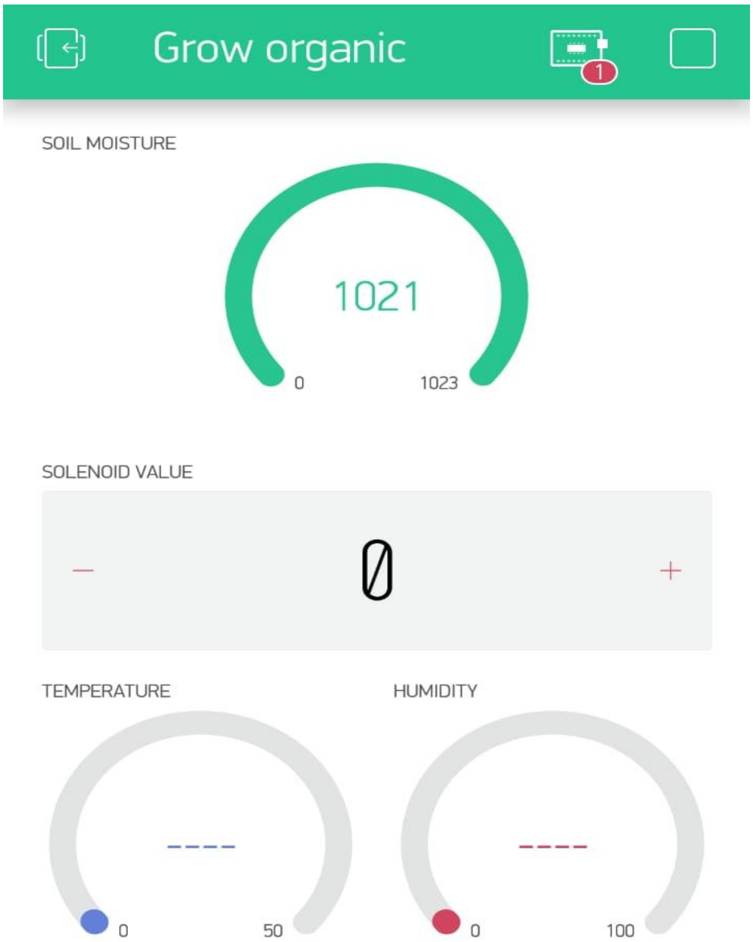


Fig 7.2.1 Application Screenshot

Below Figures are the layouts of the arrangement of Plants in different ways and position of House

1) This is the Balcony layout by which user can plant their plants and attach the Device for automation in watering

Fig 7.2.2 Balcony Layout

2) This is the Backyard layout by which user can plant their plants and attach the Device for automation in watering

Fig 7.2.3 Backyard Layout

3) This is the Terrace layout by which user can plant their plants and attach the Device for automation in watering

Fig 7.2.4 Terrace layout

Chapter 8
Conclusion & Future Work

6.1 Conclusion

The proposed model using Arduino which is fully based on Wi-Fi Module. This project developed is Time saving and Grow Organic Plant At Home. This can also useful for increasing the economy and demand of food necessity.

In daily operation associated with watering the plants need the most effortful task. Irrespective of whichever weather it’s terribly crucial to manage the quantity of water reaches to the plants. So, it'll be effective to use automatic plant watering system. The proposed model using Arduino which is fully based on Wi-Fi Module. As a result Organic Food can Grow At Home.

6.2 Future Work

This paper can be enhanced and implemented for advanced operations with the help of advanced motors, control the flow of the water. Effective use of sensor will be carried out in future work and the more parameters will be included in the future work.

Add more controls in application such

- Feedback,
- forum
- Plant selection
- Plant details

Appendix A
BMC Canvas

Customer Segments

It is the practice of dividing a customer base into groups of individuals who are similar in specific ways relevant to marketing such as interests etc.

- **General public**
  Anybody and everybody can use this device if they have a smartphone and plants garden and our device.
- **Mass market**
  Business models focused on mass markets don't distinguish between different Customer Segments.

Value Propositions

In this, we can directly come in contact with customer requirement. How it is usable and beneficial for them and which kind of services we will provide to user.

- **Water the plants using mobile devices and networks**
  This Device is used by a user through smartphone and internet for watering plants.
- **Convenience**
  User can easily water their plants at anytime from anywhere using application.
- **Efficient usage of space and time**
  User can easily grow organic plants with automated watering device in small space in their garden, terrace, and balcony.

Channels

In this, we get to know about how we will approach user. Which media should we select for the marketing of our product?

- **Advertisement**
  It is a best platform to showcase features and information about device to the users via various channels like TV, radio, poster, internet etc.
- **Connected portable devices**
  Users can connect through smart phones.
- **Internet**
  Internet is most accessed media nowadays. Developers can advertise their application on internet.

Customer Relationship

It is all about maintaining relationship with our customers by providing them facilities and services.

- **Self Service**
  We are providing predictive and preventive actions by alerting our user in advance.
- **Automated Service**

User benefit from efficiency and time savings. Efficiency increase as the user is able to water plants more quickly and in a more target-oriented manner.

Revenue Streams

In this section we get to know about revenue which will be generated from this application.

- **Installation fees**
  It is cost that user has to pay if they want to buy this device for having automation in planting.

Key Activities

In this we get to aware about how will we represent the product to User.

- **Product development Implementation**
  Device which is capable of automating the irrigation process by analysing the moisture of soil and the climate condition (like raining) and also utilize small space and water wastage is reduced.
- **Payment/billing**
  User has to pay fees for device.
- **Application development**
  Application will be created user-friendly so that users can easily use it to water and maintain their plant.

Key Resources

In this section we get idea about the kind of benefits users will get from device

- **Physical**
  Physical assets such as manufacturing facilities and automated system.
- **Development operation**
  We will collect data from sensors and utilize it for predicting water supply and acknowledging user.
- **Watering services**
  We will provide interface to user via application to water the plants.
- **Device Hardware and software**
  We provide hardware and software for auto plant watering.

Key Partnerships

In this, we can do the partnership with data provider for website, website developers, investors etc.

- **Seed and Plant suppliers**
  It provides plants, pot, seeds and additional information about Planting
- **Technology provider**
  It will provide a parts that are required for device, for example microcontroller, pump, relay, moisture sensor etc.
- **App developers and integrators**
  They develop a user-friendly App and maintain it.

Cost Structure

We can attract users towards the website by providing them facility and services in terms of cost. In which department, we should have to invest most.

- **Marketing & sales**
  It is cost used to marketing and advertisement of device
- **Device cost**
  It is cost to build the automated device.
- **Administrative cost**
  It is a cost to pay all the App developers.

Business Model Canvas

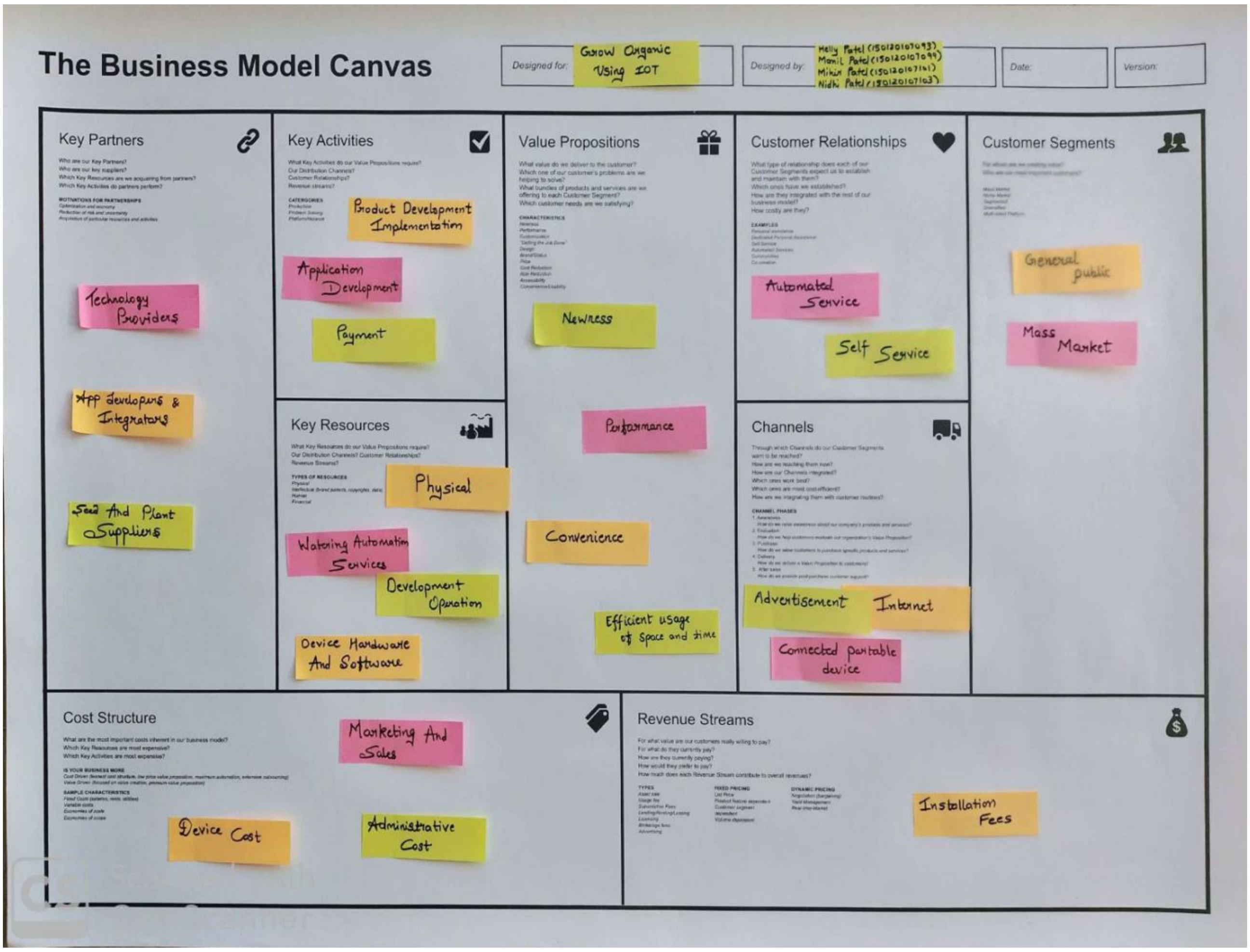


Figure B Scanned copy of BMC Canvas

References

[1] Parmod Kumar1 , Ankur Sharma2 , Rashid Hussain3"WSN Application: Intelligent Drip Irrigation System Through Moisture and Temperature Sensors" Volume 3,2014, International Journal of Scientific Research Engineering and Technology(IJSRET) ISSN2278-0882

[2] Manish Giri "Automated Intelligent Wireless Drip Irrigation using Linear Programming" Volume 2, 2013, International Journal of Advanced research in computer engineering and technology (IJARCET) ISSN: 2278-1323

[3] Dhivya, B., Justin Samuel, S., karthiga, M (Dec'2014), "ESSDM: An efficient mechanism for handling Sybil attacks in social networks", Vol.9, No.12, pp.2009 to 2013.

[4] M.Zribi1 , A.Chahbi2,"Soil Surface moisture estimation over a semi-arid region using ENVISAT ASAR radar data for soil evaporation evaluation" 2011, Hydrology and earth system sciences, 345-358.

[5] Aniket H.Hade, Dr.M.K.Sengupta"Automatic control of Drip Irrigation System and Monitoring of Soil by Wireless" Volume 7, 2014, IOSR Journal of Agriculture and veterinary science(IOSR-JAVS) ISSN:2319-2372.

[6] Emilio chuvieoo1 , David Cocero2 , David Riario3 "Combining NDVI and surface temperature for the estimation of live fuel moisture content in forest fire danger rating" 2004, Remote sensing and science 322-331.

[7] Anket H.Hade, M.K. Sengupta "Advance Research on Monitoring or Soil & remote Sensing of vegetation by various techniques for Proper Drip irrigation" Volume 7, 2014, IOSR Journal of agriculture and veterinary science(IOSRJAVS) ISSN:2319-2372 An Enhanced Automation of Garden Watering Based On IOT 2191

[8] Anket H.Hade, M.K.Sengupta"Microcontroller-based automatic irrigation system with moisture sensor" International Conference on Science and Engineering (ICSE 2011)

[9] Lakshmiprasad CN, Aashish R, Syed Muzaffar "Smart Irrigation in Agriculture" IOSR Journal of Electrical and Electronics Engineering (IOSRJEEE) e-ISSN: 2278-1676,p-ISSN: 2320-3331Volume 9,2014

[10] R.Jeberson Retna Raj, T.Sasi Praba, Surya"QWS-SEARCH: A Novel QOS driven Web Service Discovery Framework" Volume, 2014.

[11] 11.Veeramuthu,A.,Meenakshi,S.,Samuel,S.J.,Esther,S.R (2014),"Content based image classification based on texture and shape analysis using wavelet feature and clustering model", International Journal of Applied Engineering Research,Vol9,No12,pp,17