LAB REPORT

Submitted by

Bheema Shyam Kumar [RA2011003011174]

Under the Guidance of

Dr. M. Aruna

Computing Technologies
SRMIST

In partial satisfaction of the requirements for the degree of

BACHELOR OF TECHNOLOGY

in

COMPUTER SCIENCE ENGINEERING

with specialization in Core

SCHOOL OF COMPUTING

COLLEGE OF ENGINEERING AND TECHNOLOGY

SRM INSTITUTE OF SCIENCE AND TECHNOLOGY

KATTANKULATHUR - 603203

JUNE 2022

SRM INSTITUTION OF SCIENCE AND TECHNOLOGY
KATTANKULATHUR-603203

BONAFIDE CERTIFICATE

Certified that this lab report titled **Automatic Irrigation System Using Arduino Uno** is the bonafide work done by **Bheema Shyam Kumar [RA2011003011174]** carried out the lab exercises under my supervision. Certified further, that to the best of my knowledge the work reported herein does not form part of any other work.

SIGNATURE

Dr. M. Aruna
SEPM – Course Faculty
Department of Computing Technologies

ABSTRACT

Current global technology plays an important role in the field of agriculture. Automation is the technology with which a procedure or process is executed without human assistance. The main objective of this work is to determine how a person can use the automatic irrigation system of his own moderately economical facilities in a few hours to connect some electronic components and other materials. An automatic irrigation system based on sensor-based systems has been designed and implemented as one of the most widely used and advantageous automatic systems. This will help people in their daily activities, thus saving them time and hard work. This system uses sensor technology with the microcontroller, relay, DC motor and battery. Behave as an intelligent switching system that detects the soil moisture level and irrigates the plant if necessary. The ON / OFF motor will automatically be based on the dryness level of the soil. Sensor readings are transmitted to a computer to generate graphs for analysis. This type of irrigation system is easily controlled and controlled using a computer. In general, this system applies automatically for small and large gardens, nurseries, greenhouses and green roofs. This will also save time and energy, as well as minimize water loss. It will also help the farmer to benefit from the plantation without solving irrigation planning problems.

LIST OF FIGURES

LIST OF TABLES

TABLE OF CONTENTS

CHAPTER 1
PROBLEM STATEMENT

Project Description:

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.

This automated system will help farmers or a people with green fingers, to take utmost care for their farms, gardens, or backyards in a simplified way without any human intervention.

Arduino UNO is a hardware device which can be used as a base for this project.

CHAPTER 2
STAKEHOLDERS AND PROCESS MODEL

Selection of Methodology

The Agile methodology is a way to manage a project by breaking it up into several phases. It involves constant collaboration with stakeholders and continuous improvement at every stage. Once the work begins, teams cycle through a process of planning, executing, and evaluating. Continuous collaboration is vital, both with team members and project stakeholders. It's a process for managing a project that involves constant collaboration and working in iterations.

Stakeholder Name	Activity/ Area /Phase	Interest	Influence	Priority (High/ Medium/ Low)
Resource Manager – Bheema Shyam Kumar	Material Management	High	Low	3
Technical Department – Sai Harshith Yaddala	IDE and Application Development	High	High	4
Product Developer – Ganesh Chappi	Frontend and Backend Coordinator	High	High	2
Customer	Final Product	High	Low	1

. Interest and Influence matrix

Interest	Influence
High	High
Low	Low
Low	High
High	Low

CHAPTER 3
IDENTIFYING REQUIREMENTS

System Requirements

The basic requirement is to have a hands-free irrigation system so that the customer can work accordingly with a peace of mind and an ease.

Functional Requirements

This project requires a hardware component named Arduino UNO, which acts as a base to the project. Bio-sensors to detect moisture level in the soil, pump, connecting wires, and the most important one, the software component, the code to be embedded in the Arduino UNO which acts as a basic functional unit. An application shall be designed which will act as an interface between the system and the user. Customer shall be able to control the system using this application.

Non-Functional Requirements

Keeping the customer’s requirement in mind, working, durability and the quality of the product matters a lot. So, this system shall be available for use at any point of time with a continuous supply of electricity. Using the application, all controls shall be handled by the user with very much ease.

CHAPTER 4

PROJECT PLAN AND EFFORT

Stake Holder 1: - Resource Manager

The main duty of resource manager is to provide the required material and supply whatever needed for the successful deployment of the project.

Stake Holder 2: -Technical Manager

He Works for the technical development of the project providing whatever the software requirement is necessary which includes codes require for working of the system or the functioning of the application.

Stake Holder 3: - Development Manager

His work is to coordinate between the resource and technical manager for the development of the product and also acts as connection between the frontend and backend development of the project.

Stake Holder 4: - Customer

The client or the Customer is the main stakeholder who needs to be satisfied after the deployment of the project.

Kick-off Meetings: -

Regular Meetings should be held involving all the stakeholders and working staff to discuss the step-by-step completion of the project. Since the project can be executed only by a collective effort of all members, holding Regular meetings having discussions would help to complete the project on-time.

Scope Statement: - The project can start with a smaller, basic sensing and control unit that might be usable in each of these target contexts. However, the project operates with and around water, so there is also a need to protect the electronics in a watertight enclosure because the unit needs to be able to continue working even if it is misted or covered in water during its operational life. The project needs sensors for detecting humidity or moisture in the soil. This could be just a few humidity sensors when targeting a box garden, with the number of sensors increasing along with the size of the garden.

The humidity sensors might need to be placed in both shallow and deep locations under the surface of the soil to ensure that water is delivered optimally for the target plants. Temperature and light sensors can help the system better time the delivery of water based on environmental conditions. A flow meter can provide detection and verification of how much water is being delivered in use cases where a water reservoir supply might run low on water inventory, as well as detect obstruction and leaks in the water delivery system. Sufficient fault detection can prevent the irrigation system from wasting water or allowing plants to die that destroys the value of using the system.

Cost Management: - The project requires very few components and the connection is also very simple. The components are listed below:

- Arduino it costs approximately -1700
- moisture sensor -530
- 5v relay module -210
- 6v Mini water pump with small pipe -100nju7v
- Connecting wires-100
- 5v battery – 50.

Quality Management: -

Quality registrars and registration services certify that a company's quality management system meets specific industrial standards. Management system registrars perform pre-assessments and gap assessments, define corrective action plans, and perform final assessments and registration.

Resource management:

planning software, it’s a type of project management tool that enables you to plan, allocate, and then track who’s working on what project, when, and for how long.

1. **People :**

   Designer, Website designer, Coder, Tester.

2. **Activity description**

i. Design the user screen and the main page - 3 hrs

ii. Identify the data source for displaying all the main content in the front page – 9 hrs

iii. Gathering the main information from the other members of the team – 4 hrs

iv. Distribution of the designing work for the members of the team- 3 hrs

STAKEHOLDERS:

Resource manager – Bheema Shyam kumar

Technical department - Sai Harsith Yaddala

Product developer – Frontend and backend developer

Customer – final product

4. Estimation

4.1 Effort and Cost Estimation

Activity Description	Sub-Task	Sub-Task Description	Effort (in hours)	Cost in INR
Design the user screen	E1R1A1T1 (Effort-RequirementActivityTask)	Confirm the user requirements (Acceptance criteria)	3	1500
	E1R1A1T2	Communicating with the customer	2	
	E1R1A1T3	de		
Identify Data Source for displaying units of Energy Consumption		Go through Interface contract (Application Data Exchange) documents	5	2500
		Document	1	500

Effort (hr)	Cost (INR)
1	500

4.2 Infrastructure/Resource Cost [CapEx]

Category	Details	Qty	Cost per qty per annum	Cost per item
People	Network, System, Middleware and DB admin Developer , Support	3	2,000,000	6,000,000

	Consultant			
License	Operating System Database Middleware IDE	10	10000	100,000
Infrastructures	Server, Storage and Network	20	20000	400,000

4.3 Project Team Formation

4.3.1 Identification Team members

Name	Role	Responsibilities
B.Shyam	Key Business User (Product Owner)	Provide clear business and user requirements
C.Ganesh	Project Manager	Manage the project
C.Ganesh	Business Analyst	Discuss and Document Requirements
Sai Harshith	Technical Lead	Design the end-to-end architecture
B.Shyam	UX Designer	Design the user experience
Sai Harshith	Frontend Developer	Develop user interface
C.Ganesh	Backend Developer	Design, Develop and Unit Test Services/API/DB
B.Shyam	Cloud Architect	Design the cost effective, highly available and scalable architecture
C.Ganesh	Cloud Operations	Provision required Services
Sai Harsith	Tester	Define Test Cases and Perform Testing

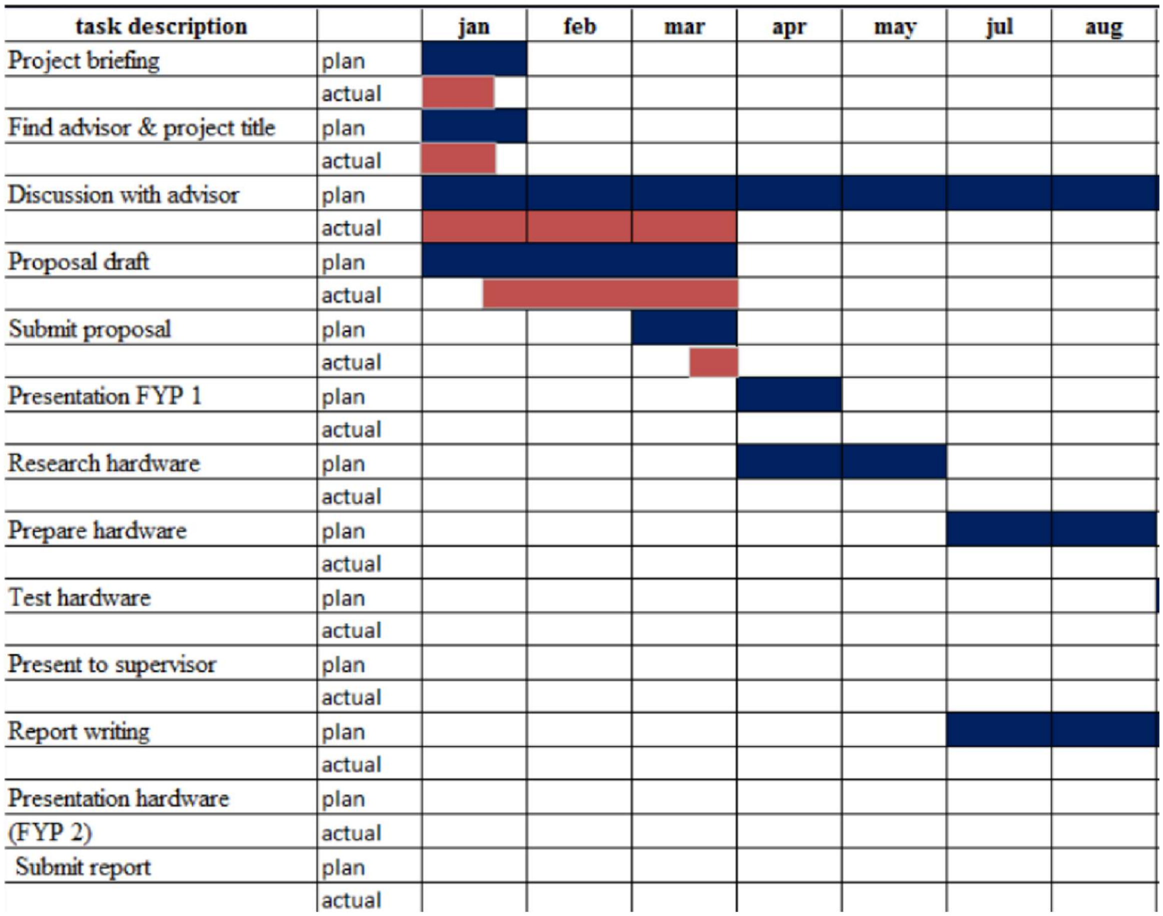
4.3.2 Responsibility Assignment Matrix

RACI Matrix	**Team Members**			
Activity	**Name (BA)**	**Name (Developer)**	**Name (Project Manager)**	**Key Business User (RESPONSIBLE)**
User Requirement Documentation	A	C/I	I	R
User required documentation	GANESH	GANESH	SHYAM	SAI HARSHITH

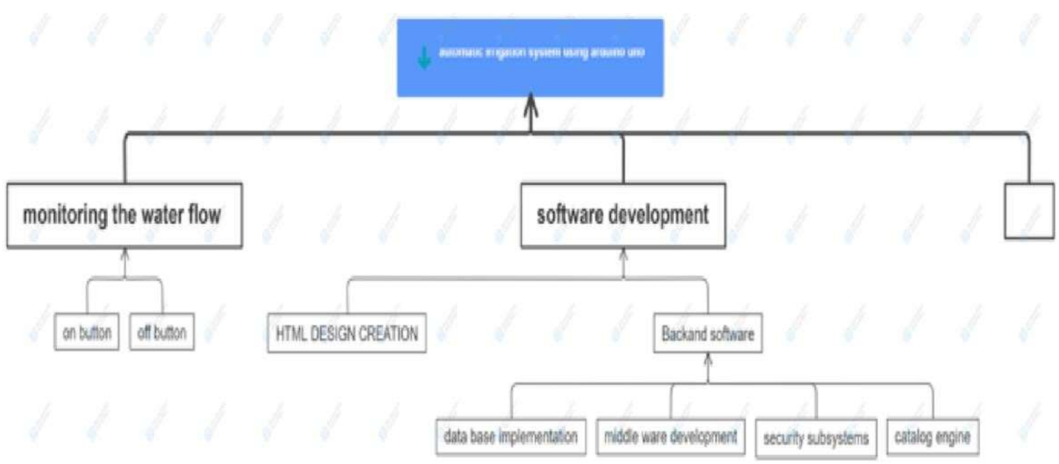
A	Accountable
R	Responsible
C	Consult
I	Inform

CHAPTER 5

GANTT CHART FOR AUTOMATIC IRRIGATION SYSTEM USING AURDINO UNO:



WORK BREAK DOWN STRUCTURE

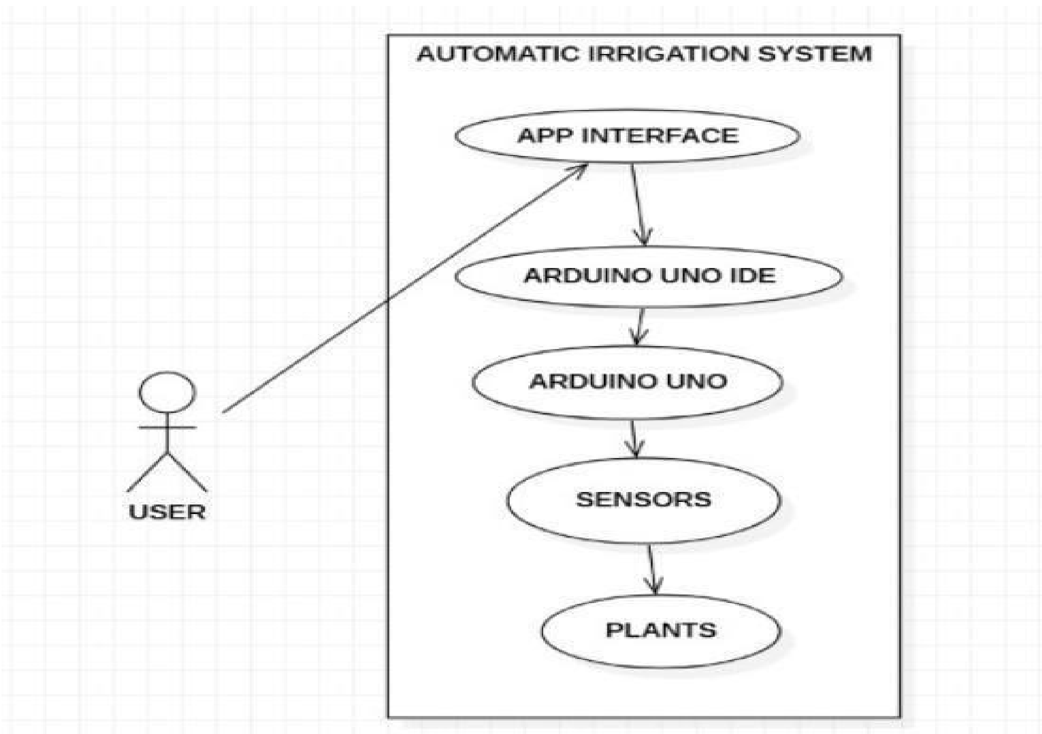


RISK MANAGEMENT:

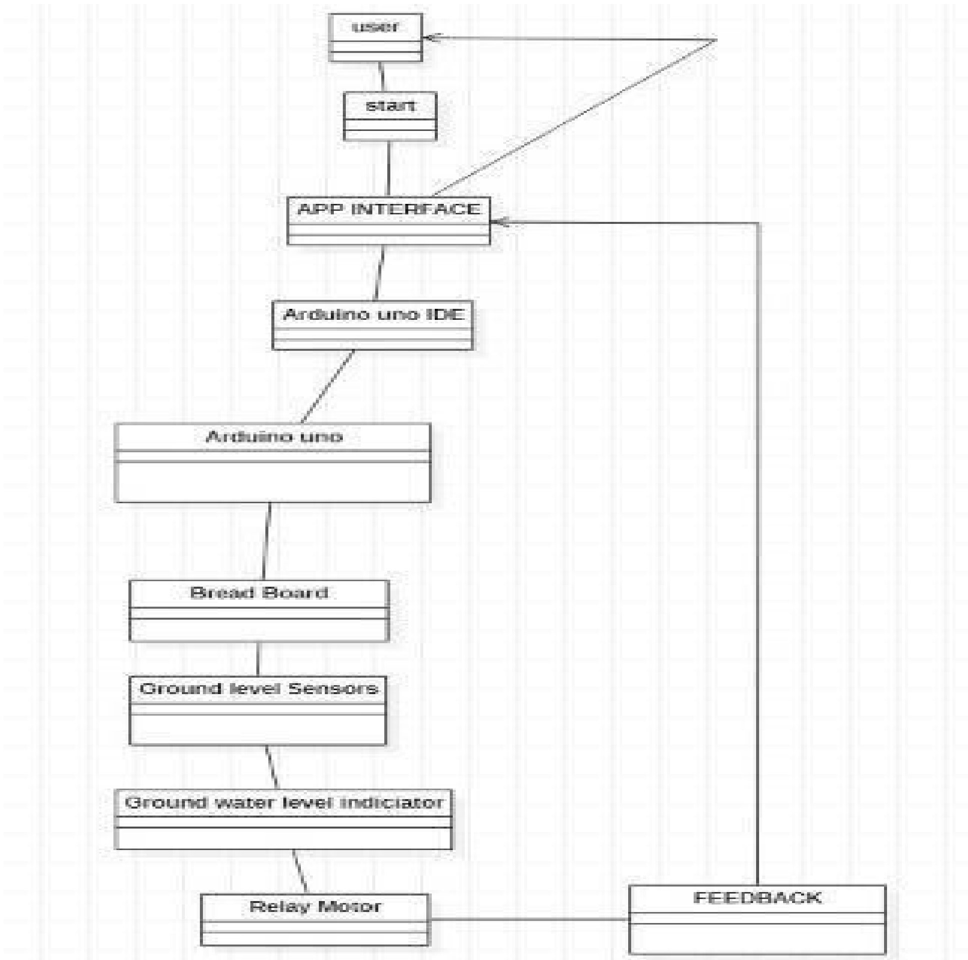
Response	Strategy	Examples
Avoid	Risk avoidance is a strategy where the project team takes action to remove the threat of the risk or protect from the impact	Extending schedules, Reducing/removing scope, Changing execution strategy
Transfer	Risk transference involves shifting or transferring the risk threat and impact to a third party. Rather transfer the responsibility and ownership	Purchasing insurance, Performance bonds, Warranties, Contract issuance (lump sum)
Mitigate	Risk mitigation is a strategy whereby the project team takes an action to reduce the probability of the risk occurring. This does not risk or potential impact, but rather reduces the likelihood of it becoming real.	Increasing testing, Ensuring stability of suppliers, Reducing complex procedures
Accept	Risk acceptance means the team acknowledges the risk and its potential impact, but decides not to take any pre-emptive action to prevent it, it is dealt with only if it occurs.	Contingency reserve budgets, Management schedule float, Event contingency

CHAPTER 6
SYSTEM ARCHITECTURE, USE CASE & CLASS DIAGRAM

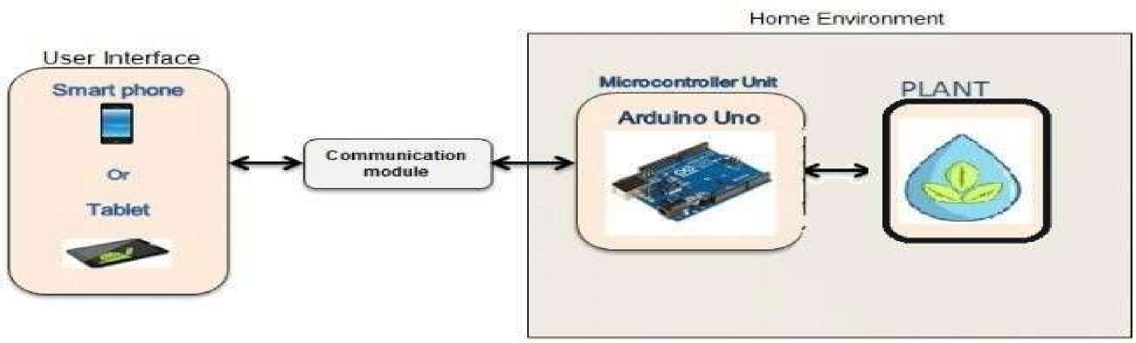
USE CASE DIAGRAM



CLASS DIAGRAM

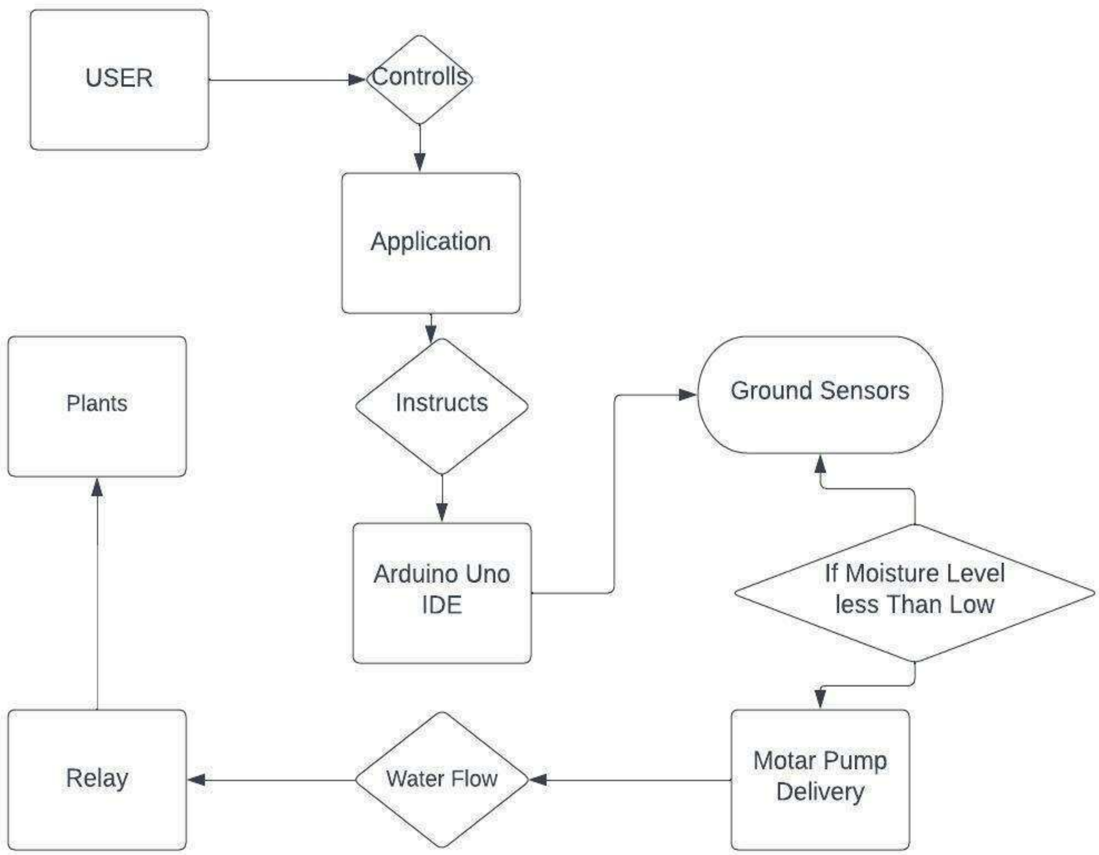


ARCHITECTURE DIAGRAM



CHAPTER 7

ENTITY REALTIONSHIP DIAGRAM



CHAPTER 8

DATA FLOW DIAGRAM

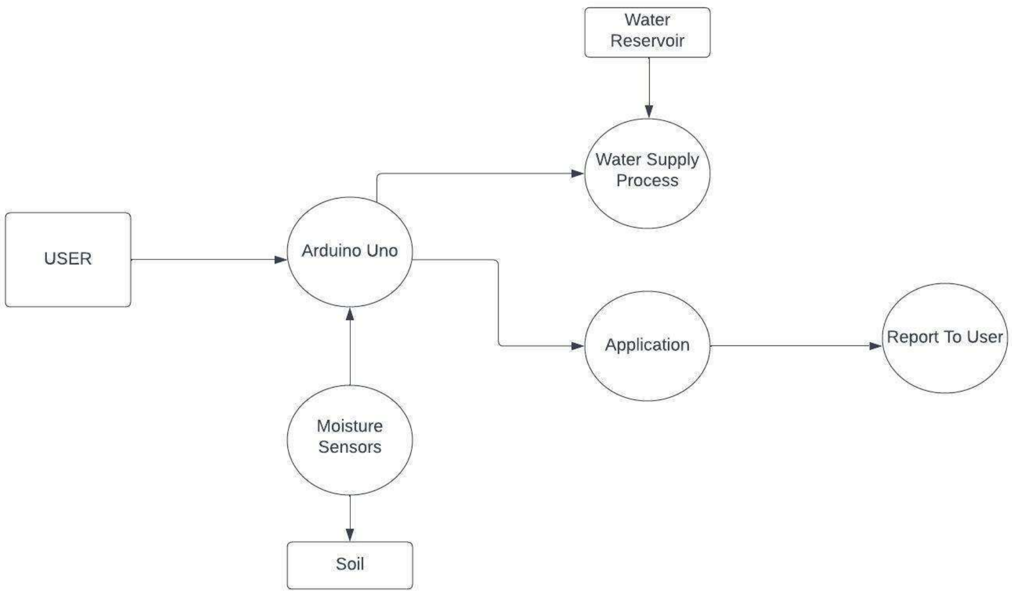
Level 0 : -

It is also known as a context diagram. It’s designed to be an abstraction view, showing the system as a single process with its relationship to external entities. It represents the entire system as a single bubble with input and output data indicated by incoming/outgoing arrows.

Level 0 DFD

1-level DFD:

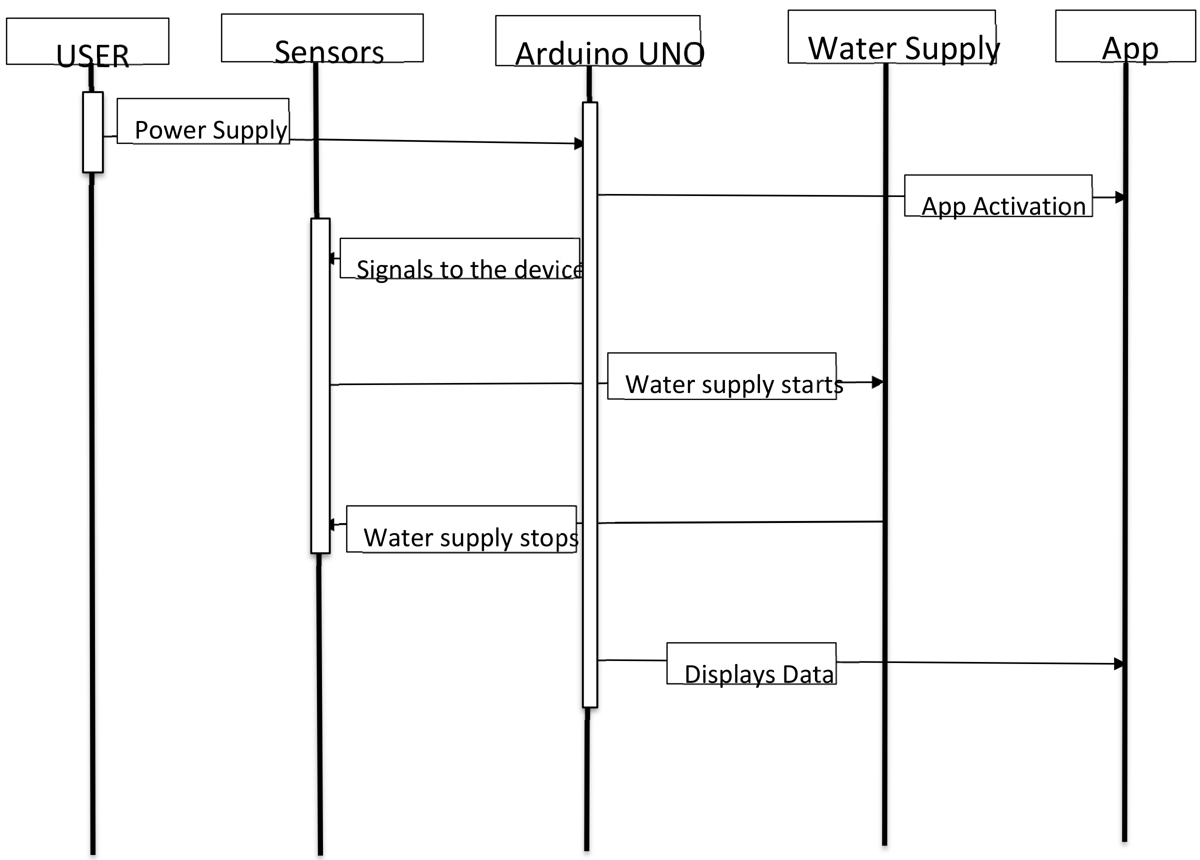
In 1-level DFD, the context diagram is decomposed into multiple bubbles/processes. In this level, we highlight the main functions of the system and breakdown the high-level process of 0-level DFD into subprocesses.



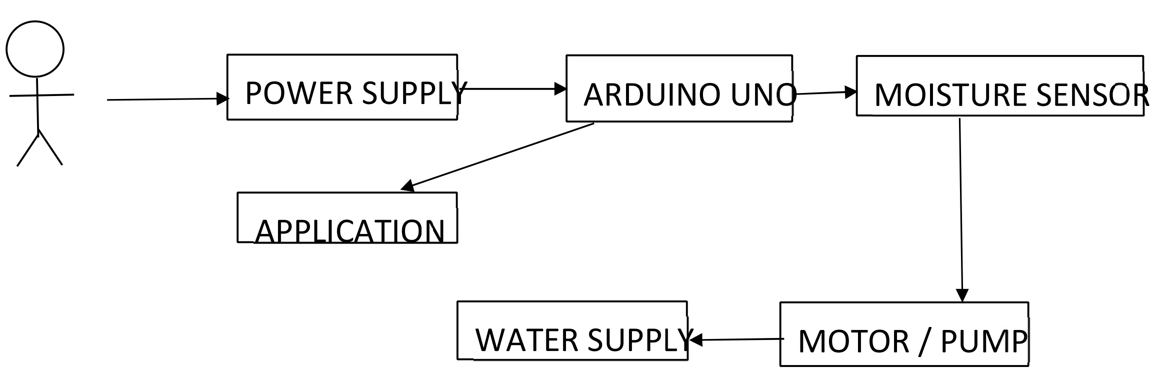
Level 1 DFD

CHAPTER 9

Sequence Diagram



Collaboration Diagram



CHAPTER 10

DEVELOPMENT OF TESTING FRAMEWORK/USER INTERFACE

User interface of the project:

Our frontend of the project will consist of a Application which uses Python Language in creation of the App, which uses code for controlling and automation of Arduino uno

Testing:

The website will be tested thoroughly on various devices so that our website don't get crashed on a set of devices. The testing will be done by sharing a link to a piece of people onvarious devices so that we can check whether the website is working properly or not. There are various types of testing phases which will be discussed below:

- **Testing real numbers**: Arduino uses a 32-bit representation for real numbers which has some problems with precision. A Unit provides a special assert to test them (assert Near) which accepts an error parameter.
- **Regression testing**: testing your code is not only about checking that it works while you are writing the code itself. It's also (and maybe most important) about ensuring it keeps on working after you have changed the code to add new features or refactor it. Unit testing helps you avoid "regressions" in your code.
- **Fixtures**: fixtures are data you feed to your test so you don't have to call test Fibonacci several times with different inputs.
- **Mock-ups**: sometimes you will need to test code that depends on another code o library. If you write your code properly you can mock that dependency so you control what it delivers.
- **Automating testing locally**: no matter what, you will forget about running your tests if you are doing it manually. You can automate your testing by using this kind of Testing.
- **Continuous Integration**: Finally you can integrate your testing With continuous integration tool testing.

And Last Step of this Testing is to Calibrating The Moisture sensor:

In the map function**,** we need to assign the dry value and wet value. To do that we need to monitor that values. You can read that values using the following code:

```
void setup() {
  Serial.begin(9600);
}
void loop() {
 int sensorValue = analogRead(A0);
  Serial.println(sensorValue);
  delay(1);
```

Upload the above code to your Arduino and open the serial monitor. Then place your soil moisture sensor in dry soil or just hold the sensor in the air and read the value. now put that value in place of 490(second term of map function).

The next step is to place the sensor in wet soil or submerge it in water and read the value and assign that value in place of 1023 (third term of map function). These values will calibrate your sensors correctly to get better results.

After converting the values we can control the pump according to the soil moisture percentage. With the help of 'If condition', I write the first condition, if the moisture percentage goes below 10, then the Arduino will turn pin 3 to LOW and the pump will turn on (our relay module uses the active low signal to trigger) and the Arduino will print pump on message in the serial monitor.

```
If (percentage < 10)
{
  Serial.println(" pump on");
```

```
  digitalWrite(3,LOW);

}
```

When the moisture percentage goes above 80 percent (indicating soil is filled with water) the Arduino will turn off the pump and print the 'pump off' on the serial monitor.

```
if(percentage >80)

{

  Serial.println("pump off");

  digitalWrite(3,HIGH);

}

}
```

By This Calibration and Testing of The Moisture Sensor is Done

Scope, Objective, and Approach to test the website:

Scope:

The Main Moto of testing logic of this system to find Maximum Defects In this system, and calibrating, the moisture sensor, The sensor senses the moisture level of the soil and when the sensor senses a low moisture level it automatically switches the water pump with the help of a microcontroller and irrigates the plant. After supplying sufficient water, the soil gets retains the moisture hence automatically stopping the pump, By this Defects should be identified as early in the Test cycle as Possible.

Objective:

The main motto of testing is to find maximum defects in a software product while validating whether the program is working as per the user requirements or not. Defects should be identified asearly in the test cycle as possible.

Approach:

There are 6 best approaches which we use while testing the website:

- Fixtures
- Regression testing
- Testing real numbers
- Mock-ups
- Automating testing locally
- Continuous Integration

Test Plan

Scope of Testing

Testing or application testing is the testing that is done before hosting or making your IDE and application to live for general use to users. Moisture sensor testing is done to find out the actual calibration and errors in sensor that can lead to Application failure in the future.

Types of Testing, Methodology, Tools

Category	Methodology	Tools Required (Software and Hardware)
Functional	Manual	ARDUINO UNO Code, Application
Compatibility	Manual	Soil, Soil Moisture
Interface	Manual	Application
Security	AUTOMATIC	Moisture Level Indicator Sensors
Performance/Load	Manual	Water Level, Moisture
Usability	Manual	Sensor Irrigation systems

CHAPTER 11

TEST CASE & REPORTING

Functional Test Cases

Test ID (#)	Test Scenario	Test Case	Execution Steps	Expected Outcome	Actual Outcome	Status	Remarks
TS_AG_REGISTRATION_001	New user registration	Accept valid user information	1. User clicks on the new user registration button. 2. Enters user details like phone number and email. 3. Verifies email and phone number through OTP. 4. User registered	User should be taken to the next page to enter more personal information and information about the farm.	-	-	In progress
TS_AG_LOGIN_002	Verify Login functionality of application login page	Enter valid user id and password.	Successful login	User should be taken to the next page for entering more user details	-	-	In progress
TS_AG_MONITORING_003	Verify the functioning of water flow using on and off.	Monitors using on and off switch	Checks the flow of water Monitors the flow of water	Monitors the flow of water using on and off switch .	-	-	In progress

Non-Functional Test Cases

Test ID (#)	Test Scenario	Test Case	Execution Steps	Expected Outcome	Actual Outcome	Status	Remarks
TS_AG_NF_001	Security	Check application security	Bugs, errors and virus checks are conducted on the application.	Provides secured transactions that include OTP as authentication verification.	-	-	In progress
TS_AG_NF_002	Low Latency of data	Check speed of data updation.	Changes are made in the back end and the updation time for each code is recorded. Code having least data latency chosen.	Data is updated on the interface without delay	-	-	In progress
TS_AG_NF_003	Performance	Check response time.	Changes are made in the back end and the responsetime for each code is recorded. Code having least response time is chosen.	The site response time should be minimum.	-	-	In progress
TS_AG_NF_004	Availability	Check site and application availability.	-	The site should be available all the time.	-	-	In progress
TS_AG_NF_005	E-mail connectivity	Check e-mail sending speed.	Application is run by beta testers.	The site should respond fast and send required e-mail responses.	-	-	In progress

CHAPTER 12

MANUAL TEST CASE REPORT

Category	Progress Against Plan	Status
Functional Testing	IDLE code	Completed
Functional Testing	Application	In-Progress
Non-Functional Testing	Network access	Moderate
Non-Functional Testing	Electricity	Good
Non-Functional Testing	Sensors	Good
Non-Functional Testing	Arduino UNO	Good

Functional	Test Case Coverage (%)	Status
Module ID	30%	Not-Started / In-Progress / Completed

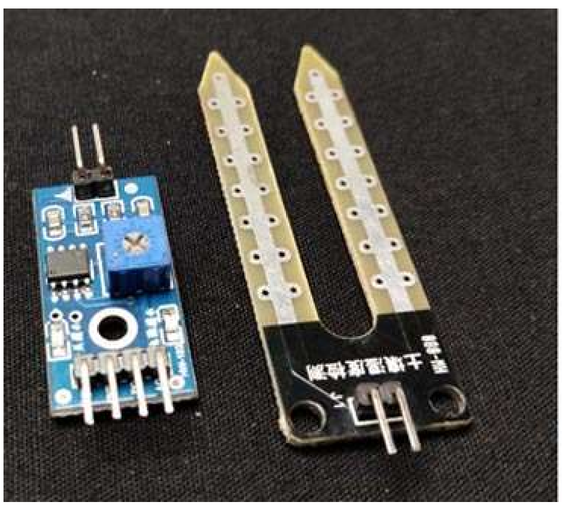
Chapter 13

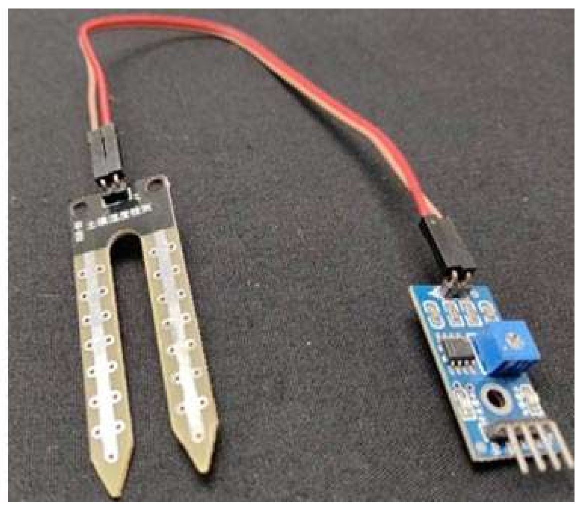
DETAILS OF ARCHITECTURE DESIGN/FRAMEWORK/IMPLEMENTATION

The Working of the Automatic Irrigation System

The logic of this system is very simple. In this system, the moisture sensor senses the moisture level of the soil and when the sensor senses a low moisture level it automatically switches the water pump with the help of a microcontroller and irrigates the plant. After supplying sufficient water, the soil gets retains the moisture hence automatically stopping the pump.

Soil Moisture Sensor





The working of the soil moisture sensor is very easy to understand. It has 2 probes with exposed contacts that act like a **variable resistor** whose **resistance varies** according to the water content in the soil. This resistance is inversely proportional to the soil moisture which means that higher water in the soil means better conductivity and hence a lower resistance. While the lower water in the soil means poor conductivity and will result in higher resistance. The sensor produces an analog voltage output according to the resistance.

The sensor comes with an electronic module that connects the probe to the Arduino. The module has an **LM393 High Precision Comparator** which converts the analog signal to a Digital Output which is fed to the microcontroller. We have covered an in-depth Arduino soil moisture sensor tutorial which covers the working of soil moisture sensor module and how to use it with the Arduino. You can check the tutorial if you want to learn more about the soil moisture sensor.

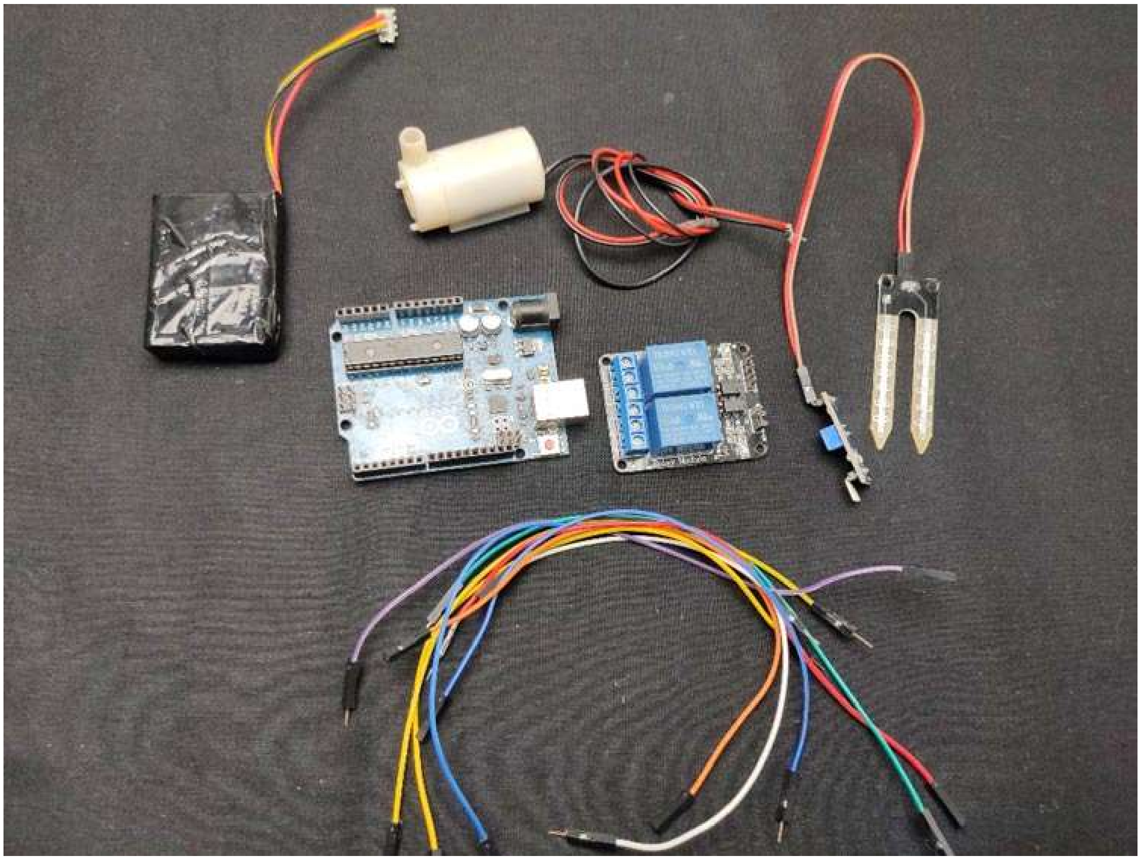
Pump

We need a small pump to irrigate the plant, but in the case of a garden, we need to drive a larger pump that can provide a higher volume of water depending on the size of your garden which can't be directly powered by an Arduino. So in case you need to operate a larger pump, a driver is necessary to provide enough current for the pump, to show that I am using a 5v relay. You can also use an AC-powered pump and use a suitable relay. The working will remain the same as shown in this project, you just have to replace the DC power input connected to the relay with an AC power input and have to power your Arduino with a separate DC power source.

Components Required for the Automatic Irrigation System

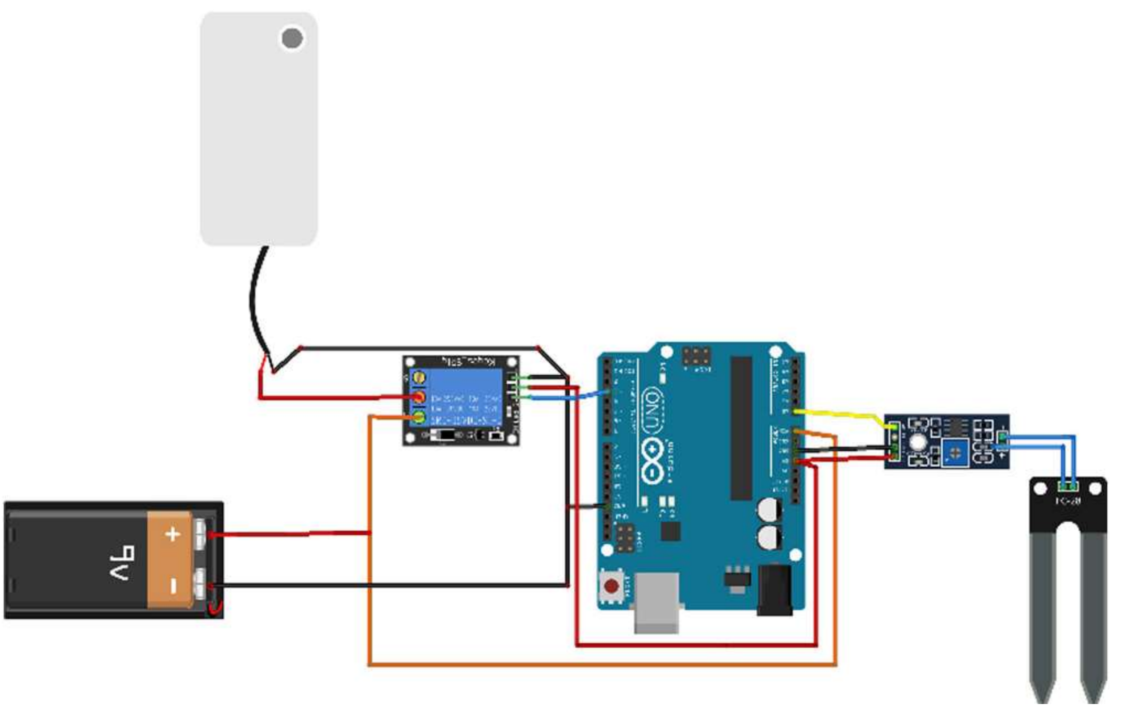
The project requires very few components and the connection is also very simple. The components are listed below:

- Arduino * 1
- moisture sensor * 1
- 5v relay module * 1
- 6v Mini water pump with small pipe * 1
- Connecting wires
- 5v battery * 1



Circuit Diagram of the Arduino Automatic irrigation system

The complete circuit diagram for the Arduino Automatic irrigation system is shown below:

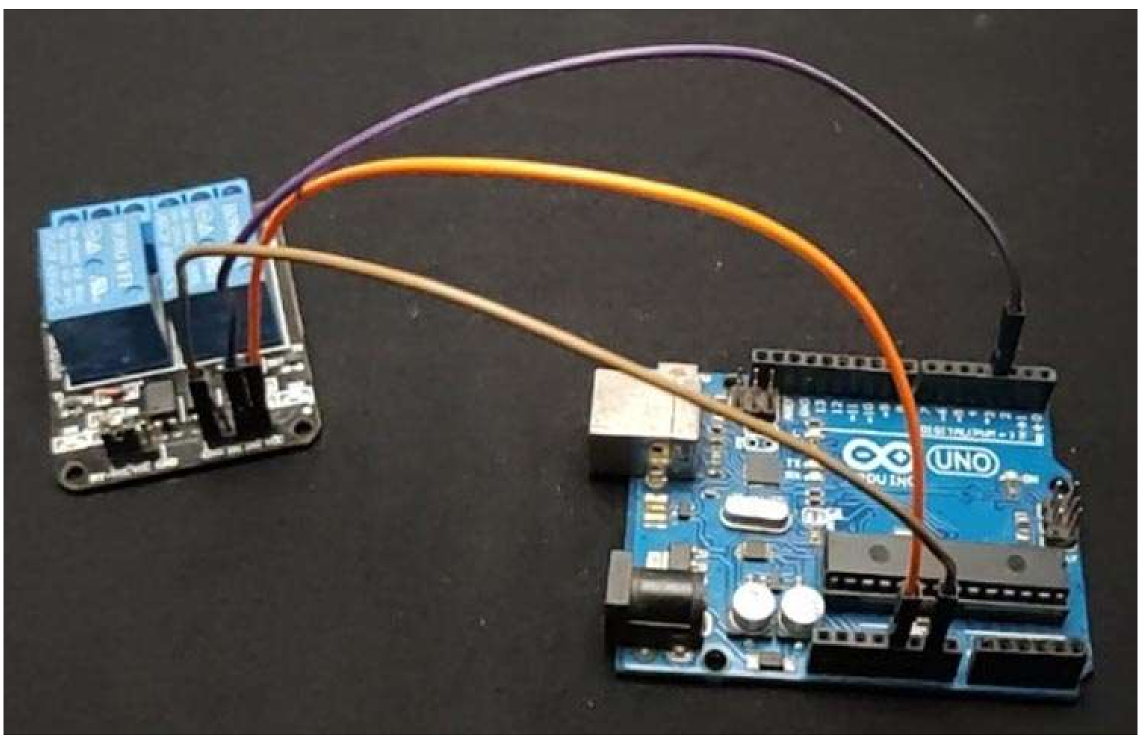


In this section, I will explain all the details with the help of the schematic diagram. The **Arduino UNO** is the brain of this whole project. It controls the motor pump according to the moisture in the soil which is given by the moisture sensor.

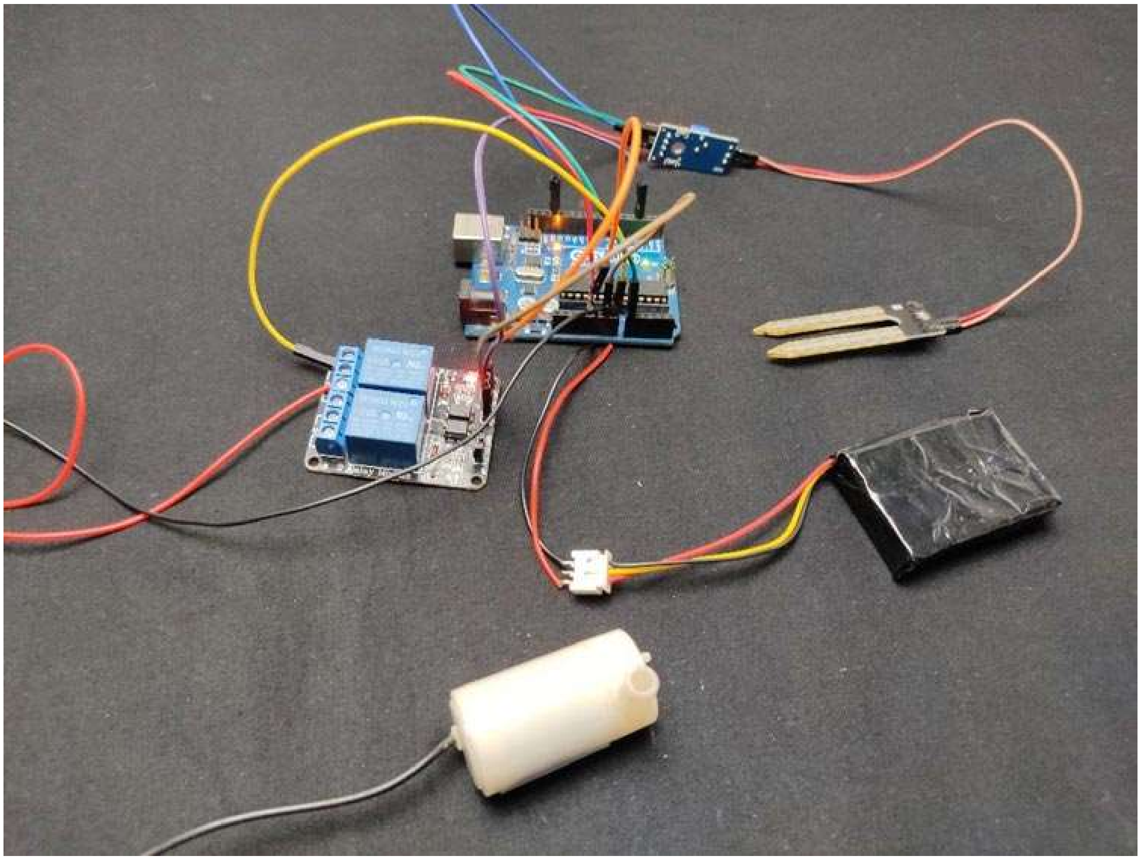
To power the circuit, I am using an external Battery. You can use any 9v or 12-volt battery. The battery is connected to the Vin and ground pins of Arduino and we can also connect the motor to this battery via a relay. Moisture sensor output is connected to the analog pin of Arduino. Do remember to use the Arduino’s 5volt pin to power the sensor and relay module.

Assembling the Automatic Irrigation System

Let's start with connecting the relay to the Arduino board. Connect the VCC of the relay module to the 5v pin of the Arduino and connect the ground of the relay to the ground of Arduino. Now connect the relay signal pin to any digital pin of Arduino except pin 13. Here I have connected it to pin 3 as shown in the image below.



The next step is to connect the soil moisture sensor with the Arduino. Connect the VCC and gnd of the sensor to the 5volt and ground pin of the Arduino. The analogue output of the sensor connects to any analogue pin of the Arduino, here I’ve connected it to pin A0 (according to our program).



Finally, connect the pump to the relay module. A relay module has 3 connection points which are common, normally closed, and normally open. We have to connect the pump positive to common and connect the normally open pin to the positive of the battery. You have to select the battery as per your pump. The next step is to connect the ground of the pump to the ground of the Arduino and finally, connect the small hose to the water pump.

Now connect the battery to the circuit and if the pump starts working then your circuit is okay. Now let's upload code to Arduino.

Explanation of the code for The Automatic Irrigation System

For this project, we are not using any library we are just using the basic functions for programming. The code is very simple and easy to use. The explanation of the code is as follows.

We start by defining all the required integers here I used two integers for storing the soil moisture and the converted moisture percentage.

```
int soilMoistureValue = 0;
int percentage=0;
```

Now, we define the pin mode, here I have used pin 3 as an output and in the next line, I have initialised Serial Monitor for debugging.

```
void setup() {
  pinMode(3,OUTPUT);
  Serial.begin(9600);
}
```

I started the *loop* section by reading the soil moisture. I used the analog Read function of Arduino to read the soil moisture and I stored that in soil MoistureValue. This value varies from 0 to 1023

```
void loop() {
soilMoistureValue = analogRead(A0);
```

In the below line, I have converted the sensor values from 0-100 percent for that we use the map function on Arduino. That means that if the soil is dry then the output moisture percentage is 0% and if the soil is extremely wet then the moisture percentage is 100%.

```
percentage = map(soilMoistureValue, 490, 1023, 0, 100);
Serial.println(percentage);
```

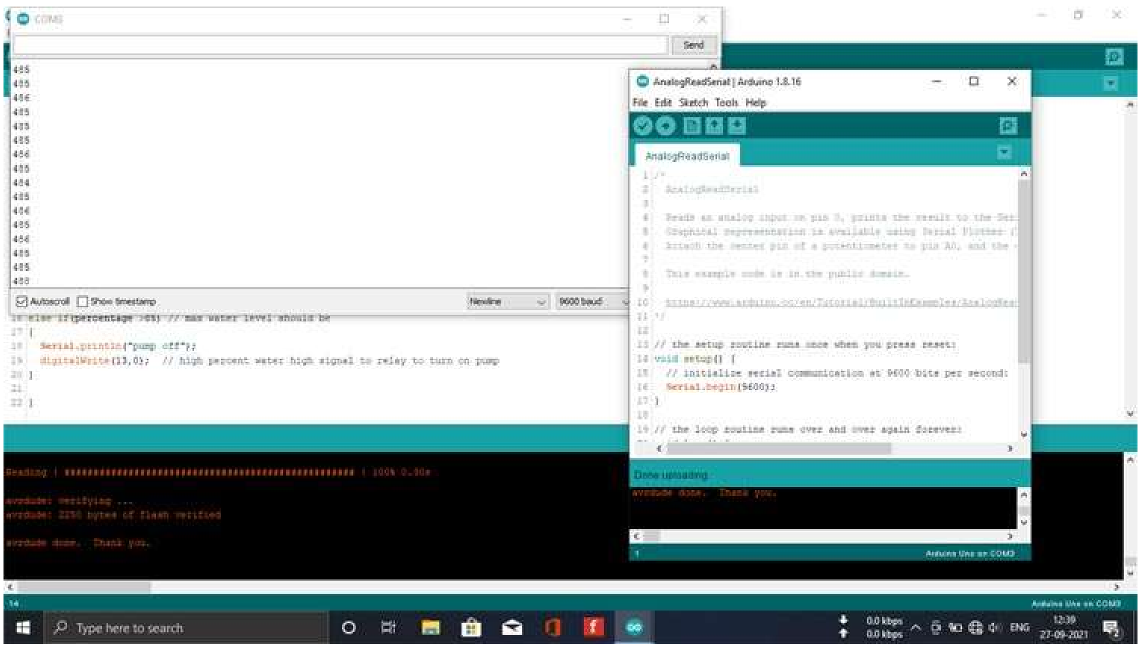
Calibrating our Moisture Sensor

In the map function, we need to assign the dry value and wet value. To do that we need to monitor that values. You can read that values using the following code:

```
void setup() {
  Serial.begin(9600);
}
void loop() {
 int sensorValue = analogRead(A0);
  Serial.println(sensorValue);
  delay(1);
```

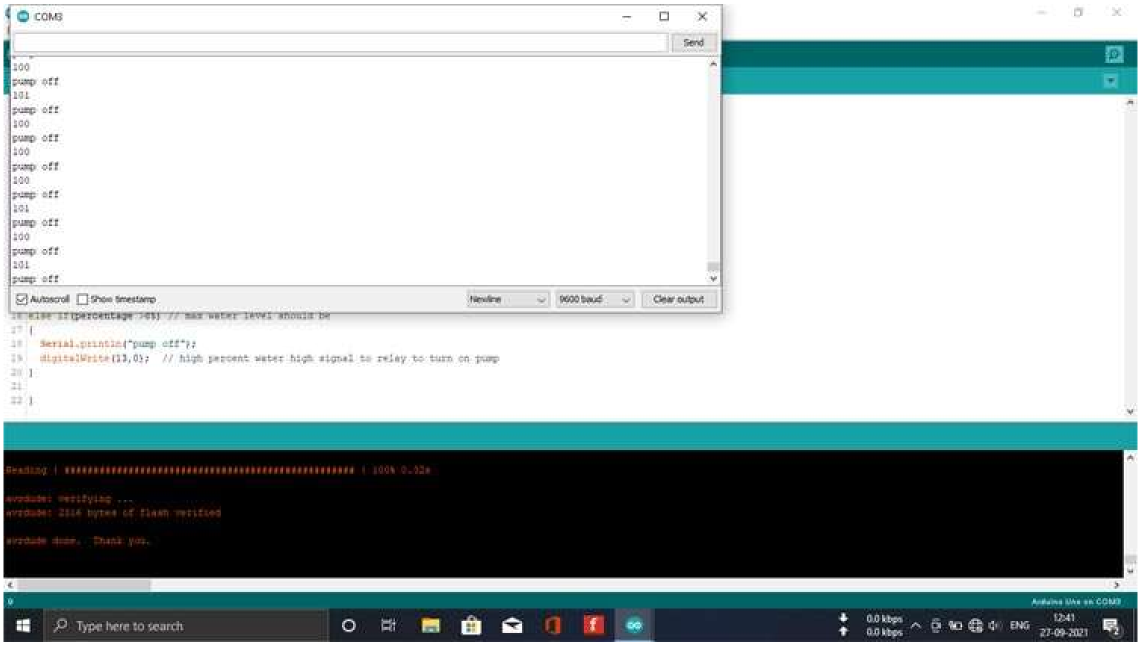
Upload the above code to your Arduino and open the serial monitor. Then place your soil moisture sensor in dry soil or just hold the sensor in the air and read the value. now put that value in place of 490(second term of map function).

The next step is to place the sensor in wet soil or submerge it in water and read the value and assign that value in place of 1023 (third term of map function). These values will calibrate your sensors correctly to get better results.



After converting the values we can control the pump according to the soil moisture percentage. With the help of ‘If condition’, I write the first condition, if the moisture percentage goes below 10, then the Arduino will turn pin 3 to LOW and the pump will turn on (our relay module uses the active low signal to trigger) and the Arduino will print pump on message in the serial monitor.

```
If (percentage < 10)
{
  Serial.println(" pump on");
  digitalWrite(3,LOW);
}
```



When the moisture percentage goes above 80 percent (indicating soil is filled with water) the Arduino will turn off the pump and print the 'pump off' on the serial monitor.

```
if(percentage >80)
{
  Serial.println("pump off");
  digitalWrite(3,HIGH);
}
}
```

Testing the Automatic Irrigation System

After uploading the code to the Arduino, I placed the whole circuit except the pump and sensor probe in a plastic box as shown in the figure below.

THE CODE:

```
int soilMoistureValue = 0;

int percentage=0;
```

```
void setup() {
  pinMode(3,OUTPUT);
  Serial.begin(9600);
}
void loop() {
soilMoistureValue = analogRead(A0);
Serial.println(percentage);
percentage = map(soilMoistureValue, 490, 1023, 100, 0);
if(percentage < 10)
{
  Serial.println(" pump on");
  digitalWrite(3,LOW);
}
if(percentage >80)
{
  Serial.println("pump off");
  digitalWrite(3,HIGH);
}
      }
```

Conclusion

Automatic irrigation system that water the plant without any human control. The automated irrigation system implemented is found to be feasible and cost effective for optimizing water resources for agricultural production. Besides the automated irrigation system, the proposed system also provides the monitoring function where users are able to check the soil moisture based on the reading on the LCD display. The proposed system has been designed and tested to function automatically. For future works, the automated irrigation system can be configured to measure the moisture level (water content) according to the moisture requirement of the different plants

REFERENCES

[1] G. Dearib, Cooperative Automatic Irrigation System using Arduino. International Journal of Science and Research 6(3) 2017, 1781-1787.

[2] S. Rakshak, Prof. R. W. Deshpande, Automated Irrigation System Based on Arduino Controller Using Sensors. International Journal of Innovative Research in Computer and Communication Engineering 5(7), 2017, 13394-13400.

[3] H.T Ingale, N. Kasat, Automated Irrigation System", Proceedings of the International Journal of Engineering Research and Development, 4(11), 2012.