CAPSTONE PROJECT REPORT

(Project Term Jan – Nov 2019)

Irrigation Improvised

Submitted by

Murikipudi Gaurav Gupta	**Registration Number :**	**11608180**
A.Naga Sai Krishna	**Registration Number :**	**11600910**
Som Rithwik	**Registration Number :**	**11603049**
Shaik Moulali	**Registration Number :**	**11608645**
Katta Karthik	**Registration Number :**	**11602353**

Project Group Number : CSERGC0450

Course Code : CSE439

Under the Guidance of

Tarun (Assistant Professor)

School of Computer Science and Engineering

LOVELY PROFESSIONAL UNIVERSITY

CAPSTONE PROJECT REPORT

(Project Term Jan – Nov 2019)

Irrigation Improvised

Submitted by

Murikipudi Gaurav Gupta	**Registration Number :**	**11608180**
A.Naga Sai Krishna	**Registration Number :**	**11600910**
Som Rithwik	**Registration Number :**	**11603049**
Shaik Moulali	**Registration Number :**	**11608645**
Katta Karthik	**Registration Number :**	**11602353**

Project Group Number : CSERGC0450

Course Code : CSE439

Under the Guidance of

Tarun (Assistant Professor)

School of Computer Science and Engineering

LOVELY PROFESSIONAL UNIVERSITY

DECLARATION

We hereby declare that the project work entitled Automated Intelligent Sensors in Irrigation is an authentic record of our own work carried out as requirements of Capstone Project for the award of B.Tech degree in Computer Science and Engineering from Lovely Professional University, Phagwara, under the guidance of Mr. Tarun, during January to November 2019. All the information furnished in this capstone project report is based on our own intensive work and is genuine.

Project Group Number**:** CSERGC0450

Name of Student 1: **Murikipudi Gaurav Gupta**
Registration Number: **11608180**

Name of Student 2: **A.Naga Sai Krishna**
Registration Number: **11600910**

Name of Student 3: **Som Rithwik**
Registration Number: **11603049**

Name of Student 4: **Shaik Moulali**
Registration Number: **11608645**

Name of Student 5: **Katta Karthik**
Registration Number: **11602353**

(Signature of Student 1)
Date:

(Signature of Student 2)
Date:

(Signature of Student 3)
Date:

(Signature of Student 4)
Date:

(Signature of Student 5)
Date:

CERTIFICATE

This is to certify that the declaration statement made by this group of students is correct to the best of my knowledge and belief. They have completed this Capstone Project under my guidance and supervision. The present work is the result of their original investigation, effort and study. No part of the work has ever been submitted for any other degree at any University. The Capstone Project is fit for the submission and partial fulfilment of the conditions for the award of B.Tech degree in Computer Science and Engineering from Lovely Professional University, Phagwara.

Signature and Name of the Mentor

Designation

School of Computer Science and Engineering,
Lovely Professional University,
Phagwara, Punjab.

Date :

ACKNOWLEDGEMENT

In performing our assignment, we had to take the help and guideline of some respected persons,

who deserve our greatest gratitude. The completion of this assignment gives us much Pleasure.

We would like to show our gratitude to Mr. Mohit Arora, Assistant Professor, Lovely Professional University for giving us a good guideline for assignment throughout numerous

consultations.

We would also like to expand our deepest gratitude to all those who have directly and indirectly guided us in writing this assignment.

Regards

Murikipudi Gaurav Gupta
A.Naga Sai Krishna
Som Rithwik
Shaik Moulali
Katta Karthik

TABLE OF CONTENTS

1. INTRODUTION

India is a vast country with more than 50% of its work force employed in agriculture and it contributes 19%-23% of India's GDP. Water is the main source for agriculture and these days ground water have become extinct in many parts of the country and many farmers suffer water scarcity because there is no adequate rainfall in many parts of the country. the rate of production of agricultural goods had decreased as a result of water scarcity. although there are some irrigational projects supplying water to agriculture, there is always a problem of over watering or underwatering of plants .

self irrigation systems are the solution to these problems as they use adequate water to grow crops by supplying water to the plants based on the moisture level in the soil. In this project we have used sensors which detect soil moisture level and humidity level and then supplies water to plants if the extreme conditions are reached. This project focuses on development of irrigation and implementing new methods of irrigating crops through intelligent systems .

Intelligent systems include various sensors and electronic and electrical components that are used to achieve the objective of improving irrigation techniques and methods to increase the productivity rate under different circumstances .

This can be achieved through integration of software and hardware together on to a compaitable platform to ensure lag free, continuous and self sustainable system . although there are lot of complexities in achieving the objective that can be possible through detailed understanding of the basic requirements and functionalities .

The main thing in the implementation of self irrigating systems is programming them to cope with the hardware components using common functions and in the most complex states this can be achieved with the help of artificial intelligence and machine learning algorithms. Through artificial intelligence and machine learning algorithms many intelligent systems have been developed by training the system regularly and updating the firmware when required.

The main aim of our project is to implement a self sustainable irrigating equipment that only uses required amount of water and improves the growth of a plant in extreme conditions.

The micro-controller based automated real time Irrigation system will supply the following: As there is no unexpected usage of water, a lot of water is saved from being wasted. The irrigation system is use only when there is not sufficient moisture in the soil and the microcontroller decides when should the pump be turned on/off, saves a lot time and water for the farmers. As there is no unanticipated usage of water, a lot of water is saved from creature wasted. This also gives much wanted rest to the farmers, as they don't have to go and revolve the pump on/off automatically. The constant increasing command of the food provisions requires a rapid improvement in food production technology. In a lot of countries like India where agriculture and the climatic conditions are isotropic, at a standstill we are not able to make full use of agricultural possessions. The main reasons is the not have of rains & insufficiency of land lake water. The continuous removal of water at normal intervals from earth is dropping the water level as a result of which the zones of un-irrigated land are frequently increasing. Also, the unexpected use of water accidentally results in wastage of water. In an Automated Irrigation System using (AVR ATMEGA-16L), the most significant advantage is that water is supplied only when the moisture in soil goes below a determined threshold value. In current times, the farmers have been using irrigation system through the

labor-intensive control in which the farmers irrigate the land at regular intervals by turning the water-pump on/off when essential. These procedures sometimes consume more water and sometimes the water supply to the land is delayed due to which the crops dry off. Water shortage deteriorate plants enlargement before visible wilting occurs. In addition to this slow development rate, lighter mass fruit follows water shortage. This problem can be absolutely rectified if we use Automated Irrigation System in which the irrigation will take place only when there will be strong requirement of water, as optional by the moisture in the soil. Irrigation is the artificial application of water to the soil usually for supporting in rising crops. In crop manufacture it is mostly used in waterless areas and in periods of rainfall shortfalls, but also to protect plants against hoarfrost.

Irrigation in India includes a network of major and minor canals from Indian rivers, groundwater well based systems, tanks, and other rainwater harvesting projects for agricultural activities. Of these groundwater system is the largest.[1] In 2013-14, only about 47.7% of total agricultural land in India was reliably irrigated.[2] The largest canal in India is Indira Gandhi Canal, which is about 650 km long. About 2/3rd cultivated land in India is dependent on monsoons. Irrigation in India helps improve food security, reduce dependence on monsoons, improve agricultural productivity and create rural job opportunities. Dams used for irrigation projects help produce electricity and transport facilities, as well as provide drinking water supplies to a growing population, control floods and prevent droughts. irrigation in India – Focus on some serious issues. Water scenario is now fast changing as a result of increasing population, rising demand for irrigation in india to raise high-yielding varieties of crops, rapid urbanization and industrialization, electricity generation, impact of global warming and erratic rainfall. Increased storage of groundwater that may be used for irrigation, municipal, household and drinking water by pumping from
wells. waterlogging and drainage problems in villages, agricultural lands, and along roads - with mostly negative consequences. The effects may be water mining, land/soil subsidence, and, along the coast, saltwater intrusion. Irrigation projects can have large benefits, but the negative side effects are often overlooked. lower the irrigation efficiency, the higher are the losses. We can say that Punjab is dying now. There is no doubt. Punjab is the food basket [of India]. Now we can say it is the disease basket. The more high-tech our lives become, the more nature we need.

2. PROFILE OF THE PROBLEM

We have seen this problem from the perspective of farmers , government and environmentalist as they are struggling to manage water scarcity and reduce the wastage of water in every possible way. As irrigation is the most important thing to government for the production of food they are building new projects and are collaborating other states to divert water for irrigation.

Due to the lack of adequate rainfall in many parts of the country water had become extinct and all the ponds and rivers are dried up resulting in the reduction of under ground water level used through borewells. government trying to assist farmers with electricity for many hours in a day but it has a less effect on agriculture due to lack of under ground water. Many trees planted on road side to maintain the ecological balance are dying due to lack of water and supervision.

To counter this we have designed a device to manage water supply to the plants that are rare and to reduce overwatering or underwatering of crops and to get acknowledged about the environmental conditions that they are in . such as temperature and moisture or humidity level in the soil.

This device helps to maintain the soil strength and improve production.

Another area of productivity lacking in trees include the distribution of the food prepared through photosynthesis at night. The smart device we developed improves this process through automation under certain lighting conditions.

3. EXISTING SYSTEM

3.1. INTRODUCTION

Till now there are many projects to improve the irrigation and to counter water scarcity and scientists are developing crops and trees that can grow in worst environmental conditions with least amount of resources supplied .

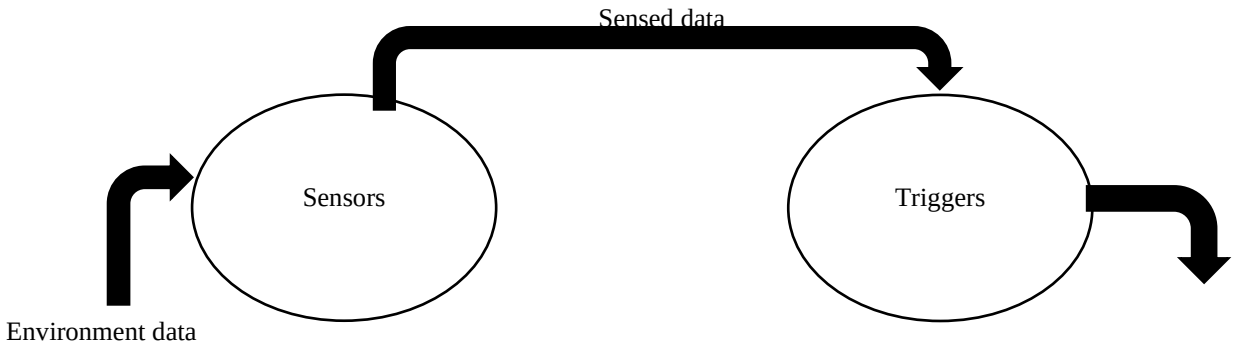
But before 10 years if you think there are less factors to counter such conditions and to improve irrigation.

3.2. EXISTING SYSTEM

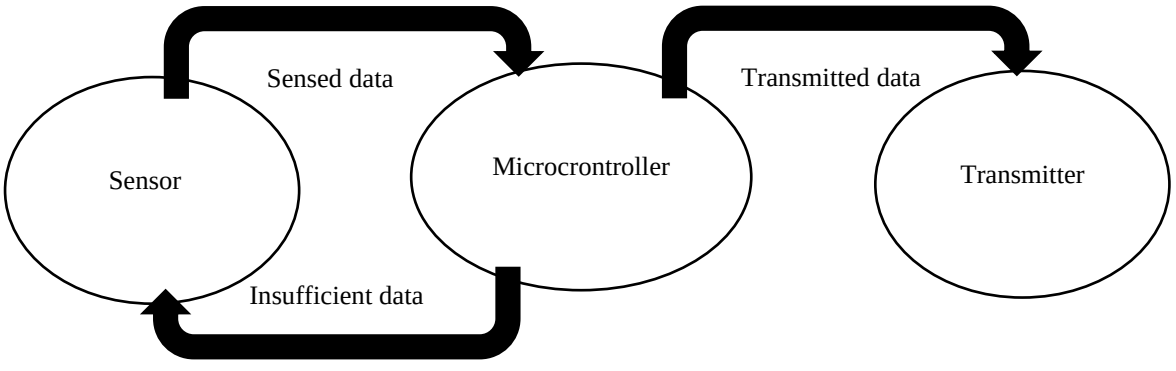
there is no such existing software for this kind of problems they are platform independent and can implemented with the help of any high level programming languages using different functionalities altogether and integrating them with different kinds of hardware components. The programmed device software must be able to be updated when required though firmware update procedure to ensure the longevity of the device.

3.3. DFD FOR EXISTING SYSTEM

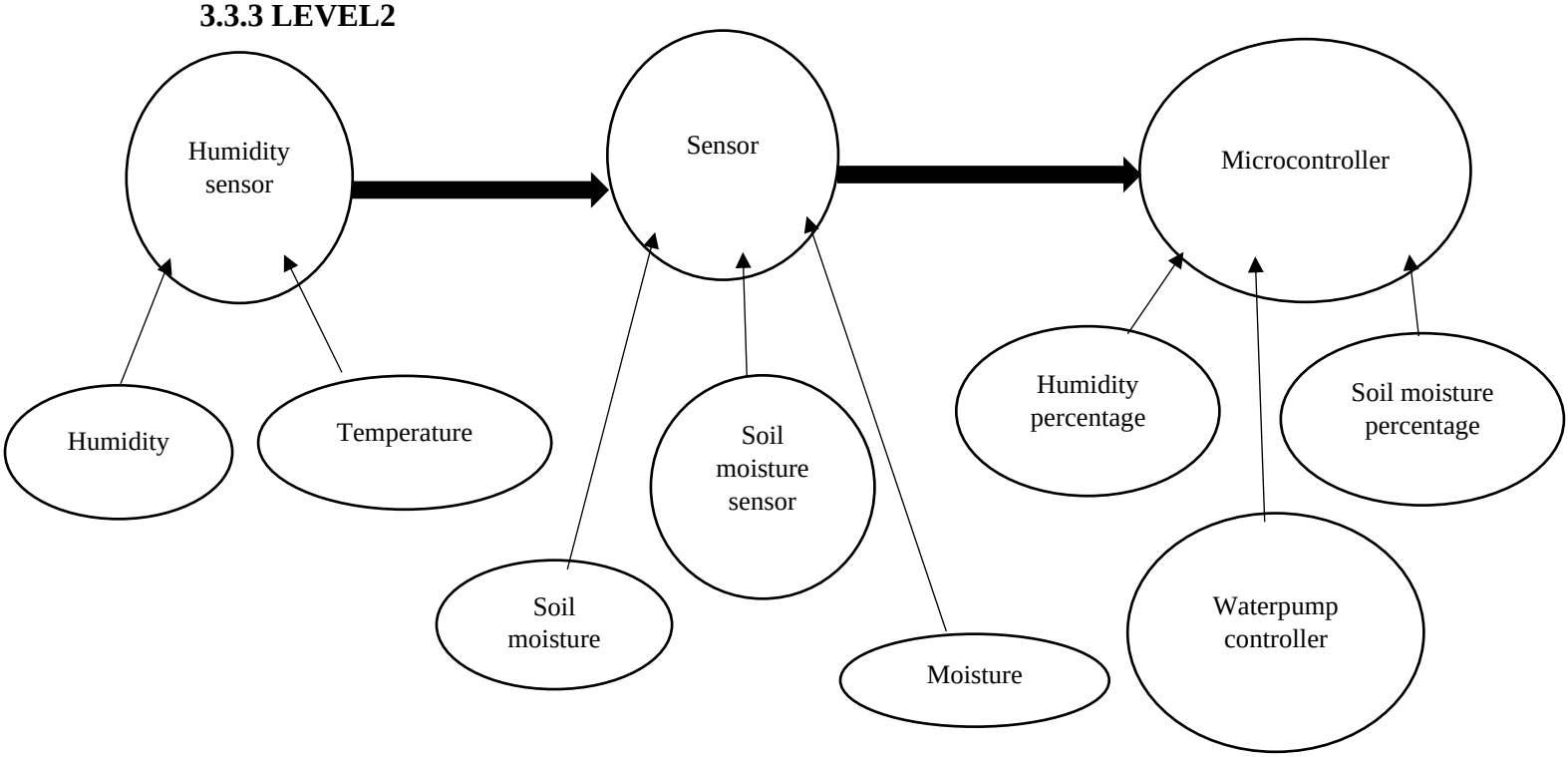
3.3.1 LEVEL 0



3.3.2 LEVEL 1



3.3.3 LEVEL2



3.4. WHATS NEW IN THE SYSTEM TO BE DEVELOPED

Our project is to design a self watering equipment that waters plants on the basis of moisture or humidity level and to make it a self sustained one.

Unlike every other project developed to improve irrigation methods we used gsm module to let user acknowledge about the environmental conditions that the plants are in . In almost 90% of such projects developers use different functionalities that are complex to implement .but we have used simple things to make every one understand the things and the functionality of the developed device.

Advantages :

reduces manpower

power efficient

accurate

cost effective

time saving

timely notices

The gsm module addition gives information about the conditions and burden on the device and with the information gathered we can analyse the working of the device and the growth rate of the plant and can calculate the water consumption level.

We have also added an other complexity by implementing the automated rgb lighting conditions that dims the lights at night time in a semi closed environment to increase the food distribution level throughout the plant efficiently and increases the growth of the plant by 30%-40% faster than the regular ones.

4. PROBLEM ANALYSIS

4.1. PRODUCT DEFINATION

Irrigation improvised is basically an automated plant watering device that uses soil moisture and humidity sensors to detect the moisture level , humidity level and temperature level and waters the plant on the basis of the conditions. It also uses gsm module to acknowledge the user through sms when the plant is watered. It also sends the fetched details through sensors to the user to analyse the conditions that the plants are in.

features of irrigation improvised

4.1.1. SELF WATRERING SYSTEM :

Our self watering system waters the plants only when the extreme conditions for the survival or the efficient growth of the plants are breached. It counters the over or inder watering of plants.

4.1 2. ACKNOWLEGEMENT SYSTEM

It sends acknowledgement to the user about the condition of the plants and it notices user when the plants are watered.

4.1.3. SELF SUSTAINABILITY

It uses solar power through the solar panel equipped to power the entire system when the battery drains and supports longevity of the device.

4.1.4. IMPROVISED FOOD DISTRIBUTION

Irrigation improvised uses certain lighting conditions at night in a semi closed environment to enable faster food distribution and faster growth of an plant compared to the regular ones.

4.2 FESIBILITY STUDY

4.2.1. TECHNICAL FESIBILITY

Technical feasibility of irrigation improvised :

First we should ensure that the software programmed is working and should make sure that the functional requirement is achieved .

The required hardware components and intelligent sensors functionality is stated in the software.

We should make sure of integration of hardware and software components in a efficient way in order to make them function according to the purpose.

First check the condition of the plants on which the device will be equipped.

4.2.2. ECONOMICAL FISIBLITY

in economical feasibility we need to make our project live and we should calculate the cost needed to design and implement the project. While making our project we should focus on software’s economic feasibility also.

Our project is ecomical and in future we will try to make it more affordable because we already used basic equipment but the addons have a slight effect on price. Excluding addons it is affordable foe growing rare and extinct plants.

4.2.3. OPERATIONAL FESIBILITY

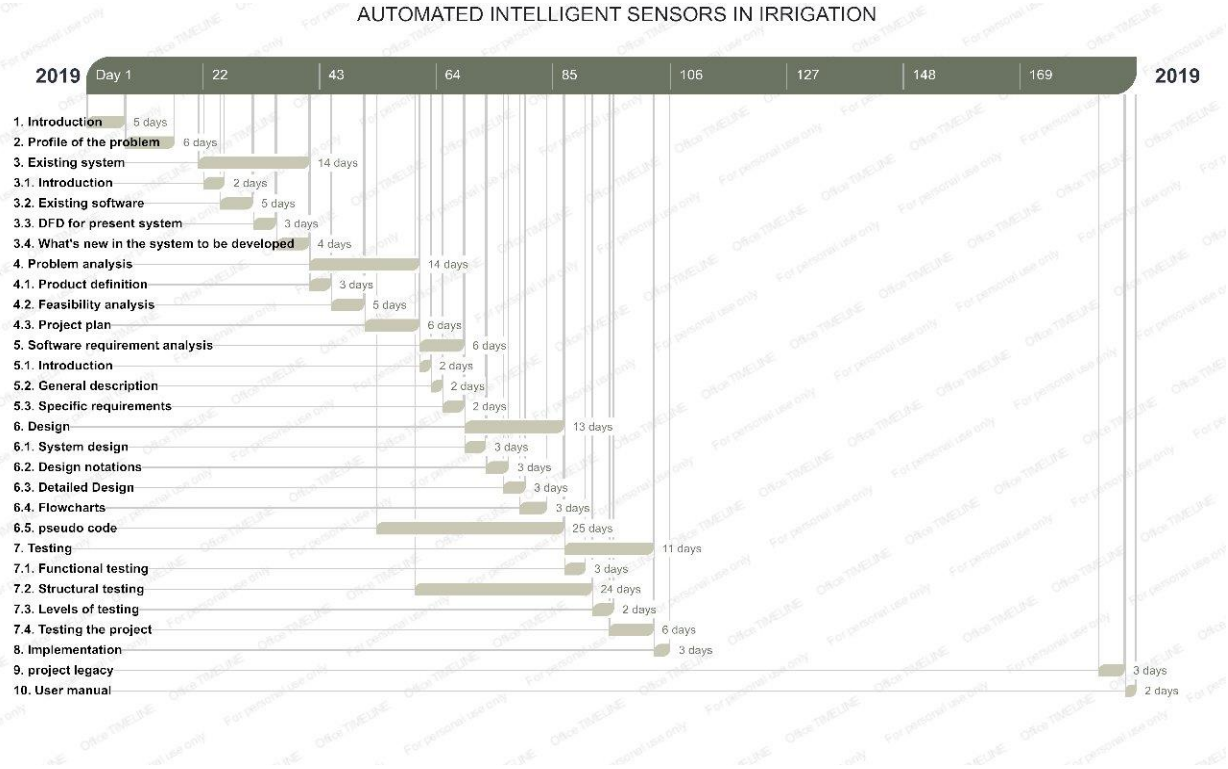
after developing the project we need to do some operations that have to be done in time to ensure the constant working of the device

monitor the water level in the tank for watering when required.

We made it self sustainable through solar panel so we need not to worry about battery on sunny days.

We need to update the firmware when required based on the operational longevity of the device.

4.3. PROJECT PLAN



IDEA OF PROJECT AND TECHNOLOGY

TASK	IDEA OF PROJECT AND TECHNOLOGY
NO OF DAYS	16

REQUIREMENT ANALYSIS

TASK	REQUIREMENT ANALYSIS
NO OF DAYS	20

IMPLEMENTATION

TASK	DESIGN AND CODING
NO OF DAYS	38

TESTING

TASK	TESTING AND DEALING WITH UUNWANTED OUTCOMES
NO OF DAYS	32

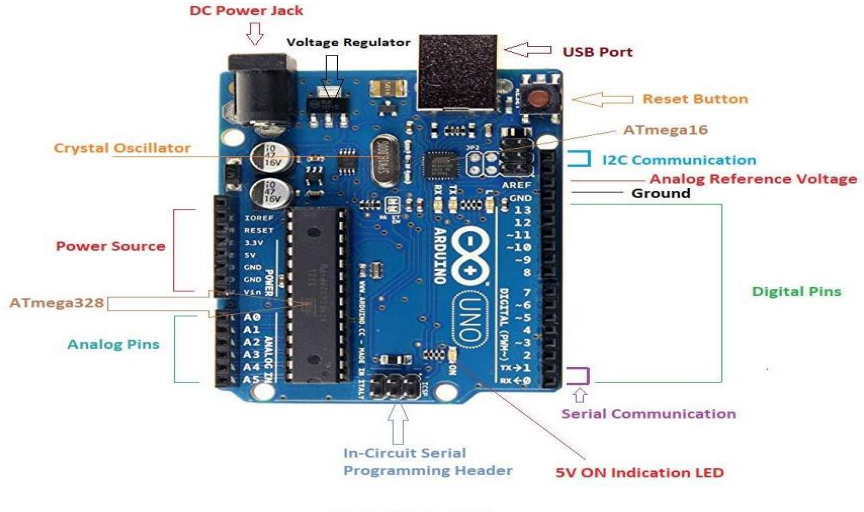
5. SOFTWARE AND HARDWARE REQUIREMENT ANALYSIS

5.1. INTRODUCTION

Here in this section the project is dependent on the software and hardware components so all the hardware components are specified here and their working temperature and voltages as the whole project is dependent on their components.

5.2. GENERAL DESCRIPTION

Arduino Uno:

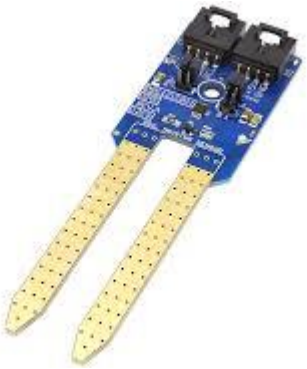


Arduino Uno is a microcontroller board developed by Arduino.cc which is an open-source electronics platform mainly based on AVR microcontroller Atmega328.

The current version of Arduino Uno comes with USB interface, 6 analog input pins, 14 I/O digital ports that are used to connect with external electronic circuits. Out of 14 I/O ports, 6 pins can be used for PWM output.

It allows the designers to control and sense the external electronic devices in the real world.

Soil moisture sensor:



The Soil Moisture Sensor uses capacitance to measure the water content of soil (by measuring the dielectric permittivity of the soil, which is a function of the water content). Simply insert this rugged sensor into the soil to be tested, and the volumetric water content of the soil is reported in percent.

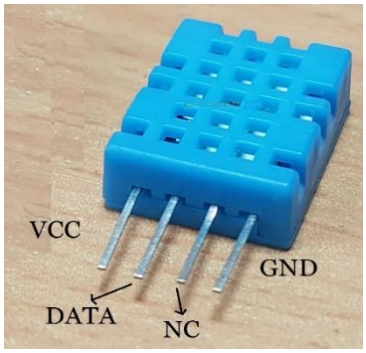
Specifications:

Range: 0 to 45% volumetric water content in soil (capable of 0 to 100% VWC with alternate calibration)
Accuracy: ±4% typical
Typical Resolution: 0.1%
Power: 3 mA @ 5VDC
Operating temperature: –40°C to +60°C
Dimensions: 8.9 cm × 1.8 cm × 0.7 cm (active sensor length 5 cm)

Bluetooth module:

Typical -80dBm sensitivity
Up to +4dBm RF transmit power
Low Power 1.8V Operation ,1.8 to 3.6V I/O
PIO control
UART interface with programmable baud rate
With integrated antenna
With edge connector

DHT11 Sensor:



The DHT11 Humidity and Temperature Sensor consists of 3 main components. A resistive type humidity sensor, an NTC (negative temperature coefficient) thermistor (to measure the temperature) and an 8-bit microcontroller, which converts the analog signals from both the sensors and sends out single digital signal.

Functioning voltage - 3 to 5 volts
Humidity range - 20-80/+- 5%
Functioning temperature - 0 to +50 Celsius

Other components used – solar panel, AC converter, wires, LDR, GSM module, sim card, mini water pump
Overall project

Functioning voltage 3-5 Volts
Humidity range - 20-80/+-5%
Functioning temperature - 0 - +50 Celsius

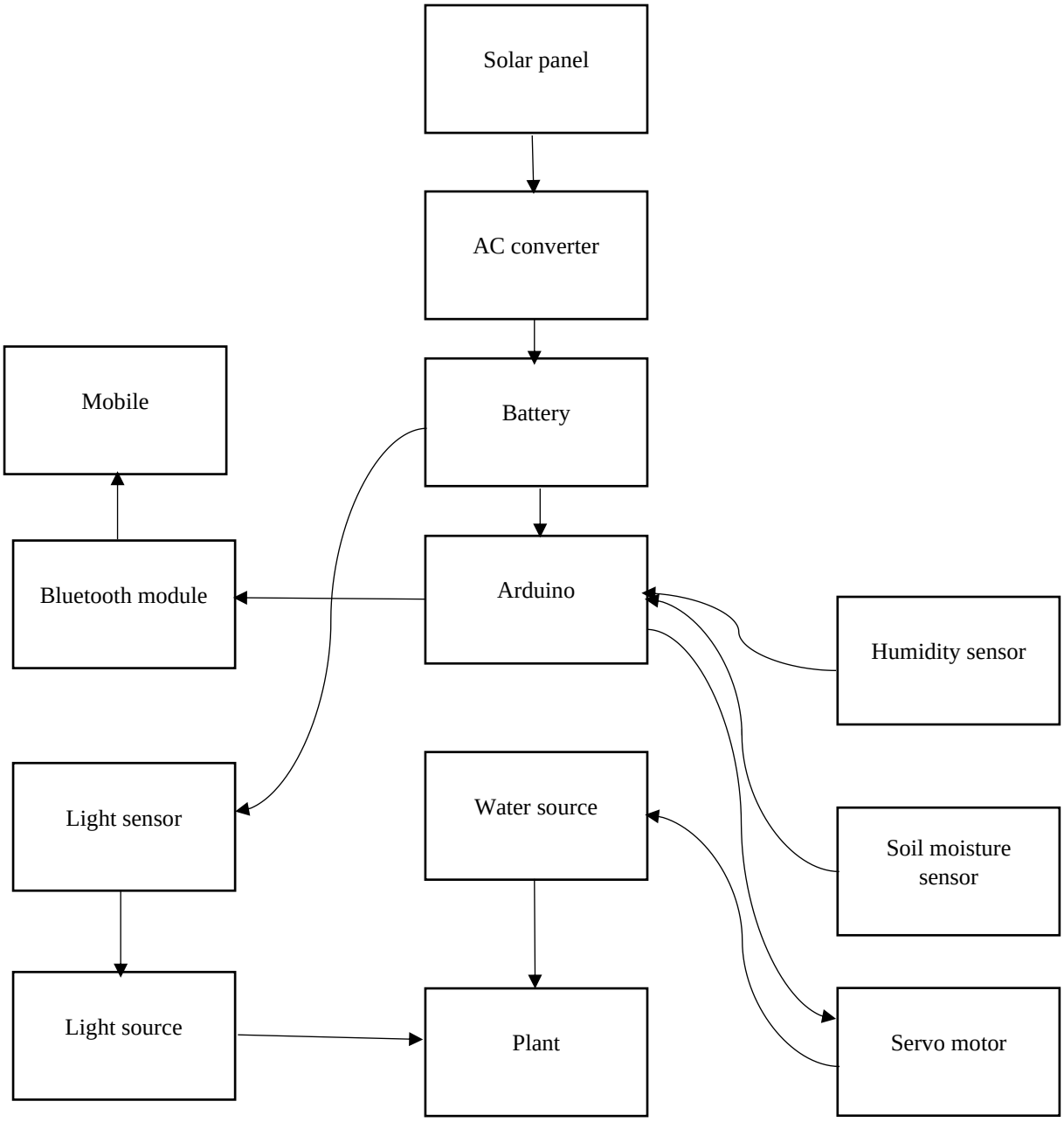
5.3. SPECFIC REQUIREMENTS

The project created is sustained on solar energy so it has to be placed under direct sunlight for efficient results
A continuous water supply so that it can pump at anytime of the day
And surrounding has to satisfy all the functioning temperature of all the sensors.

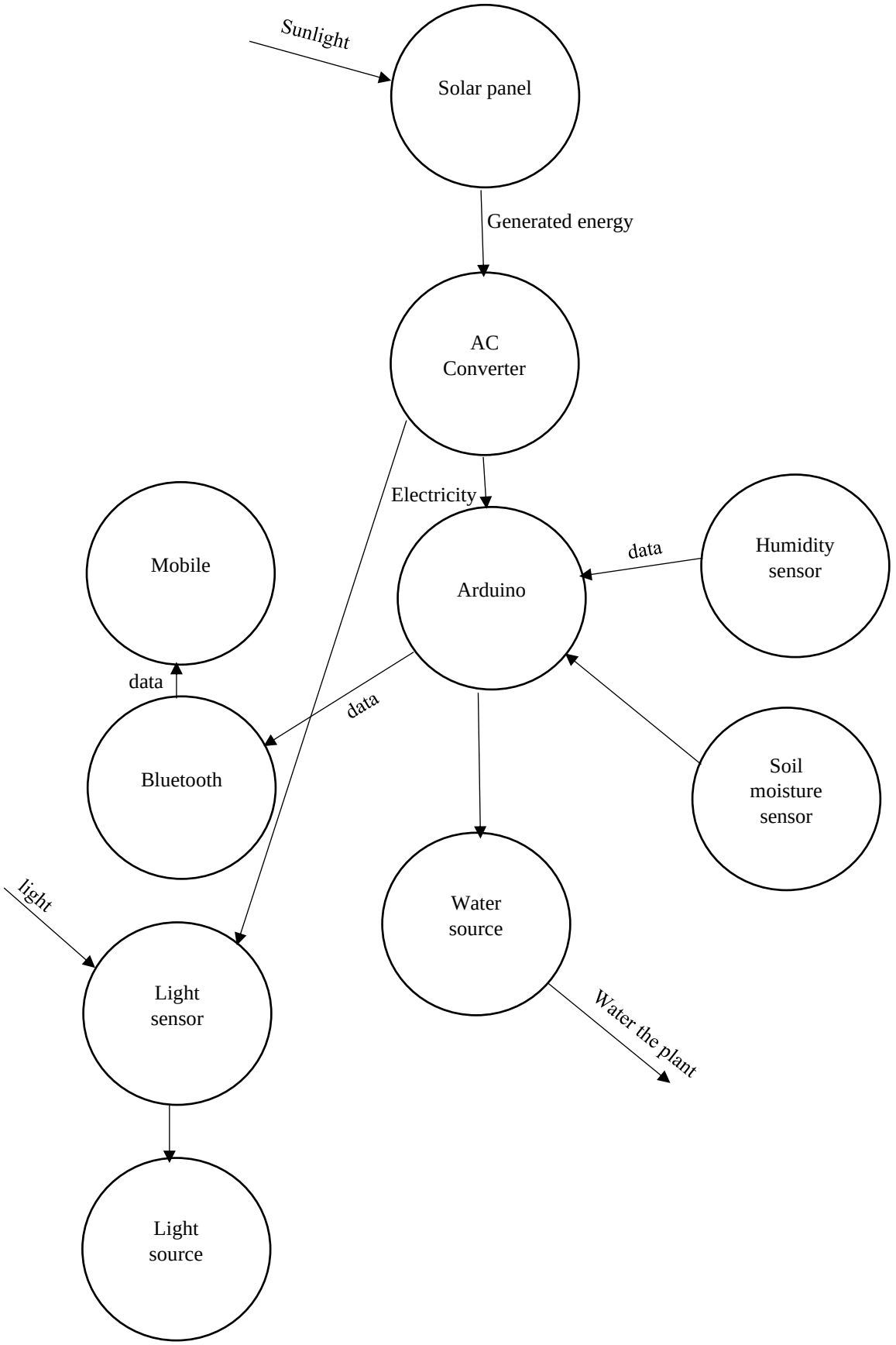
6. SYSTEM DESIGN

System design include the Data flow diagram and the entity relationship diagram defines the relation between the sensors and micro controller as well as transmitter (Bluetooth). While sensors send the data to the microcontroller and then transmitted after calculating the percentages of humidity and relative humidity in the soil as well as the temperature and relative temperature in the soil. The System is described using the micro controller Arduino as the main system. Some of the required parts are Arduino, DHT 11 Sensor, Soil Moisture Sensor, 3.5V DC submergible water pump and few servo motors if required.

6.1 SYSTEM DESIGN

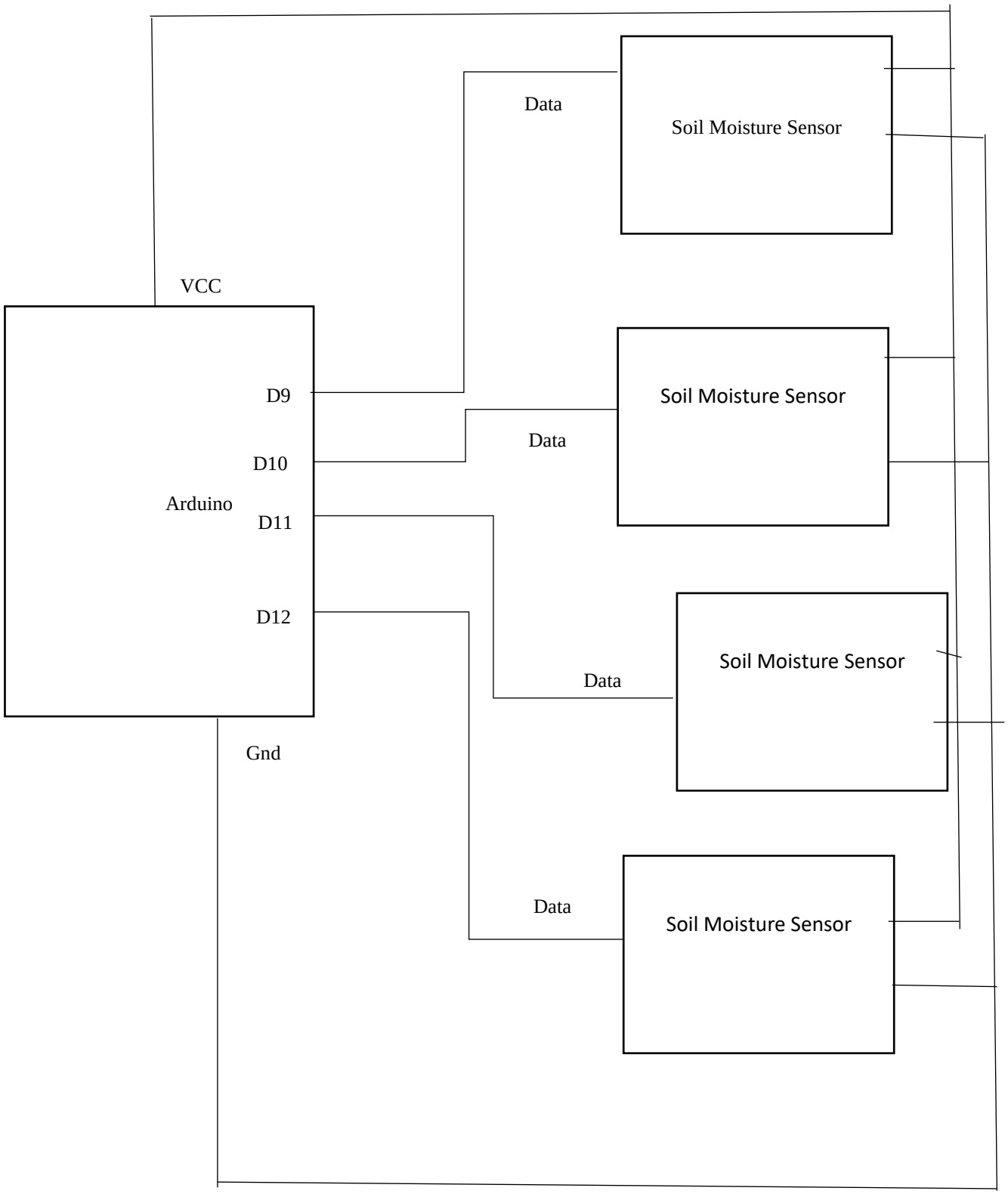


6.2 DESIGN NOTATIONS

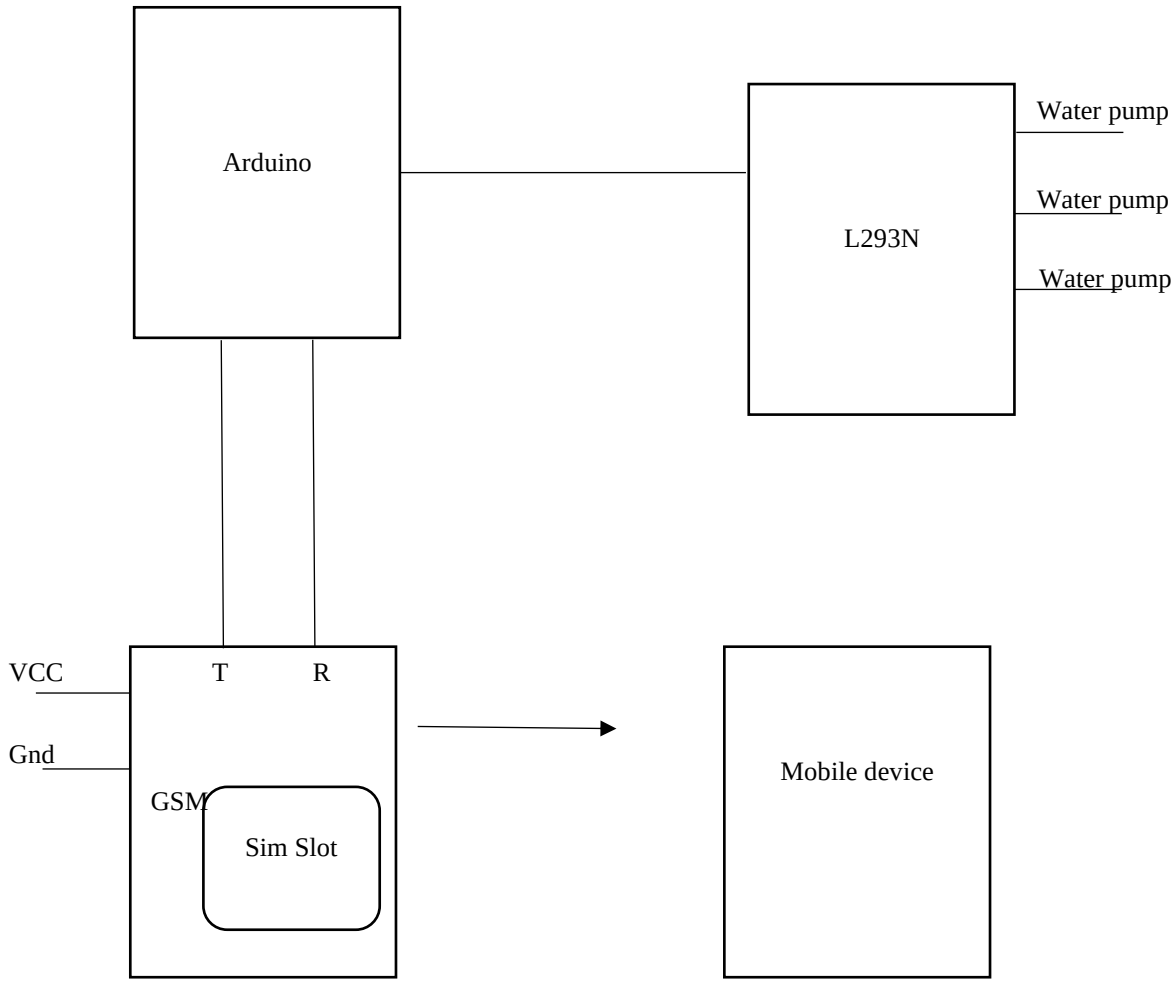


6.3 DETAILED DESIGN

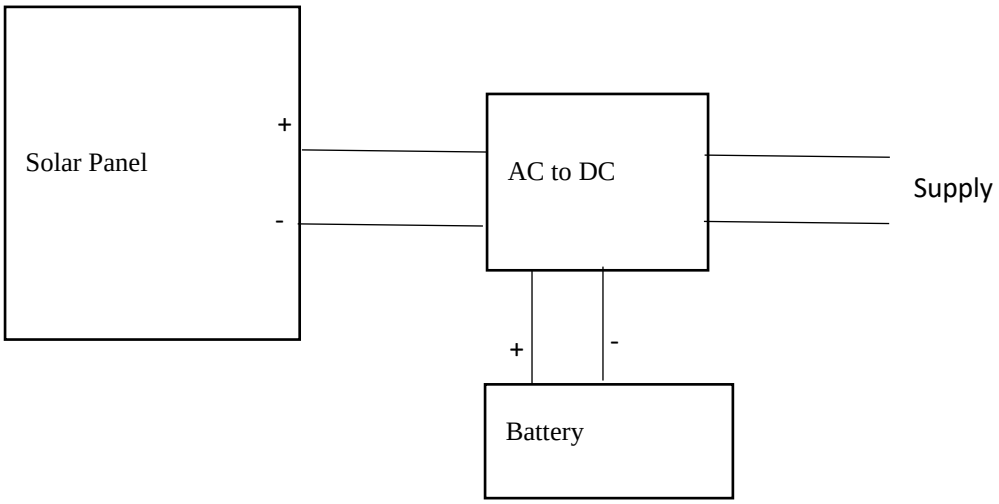
6.3.1 DATA PROCESSING UNIT



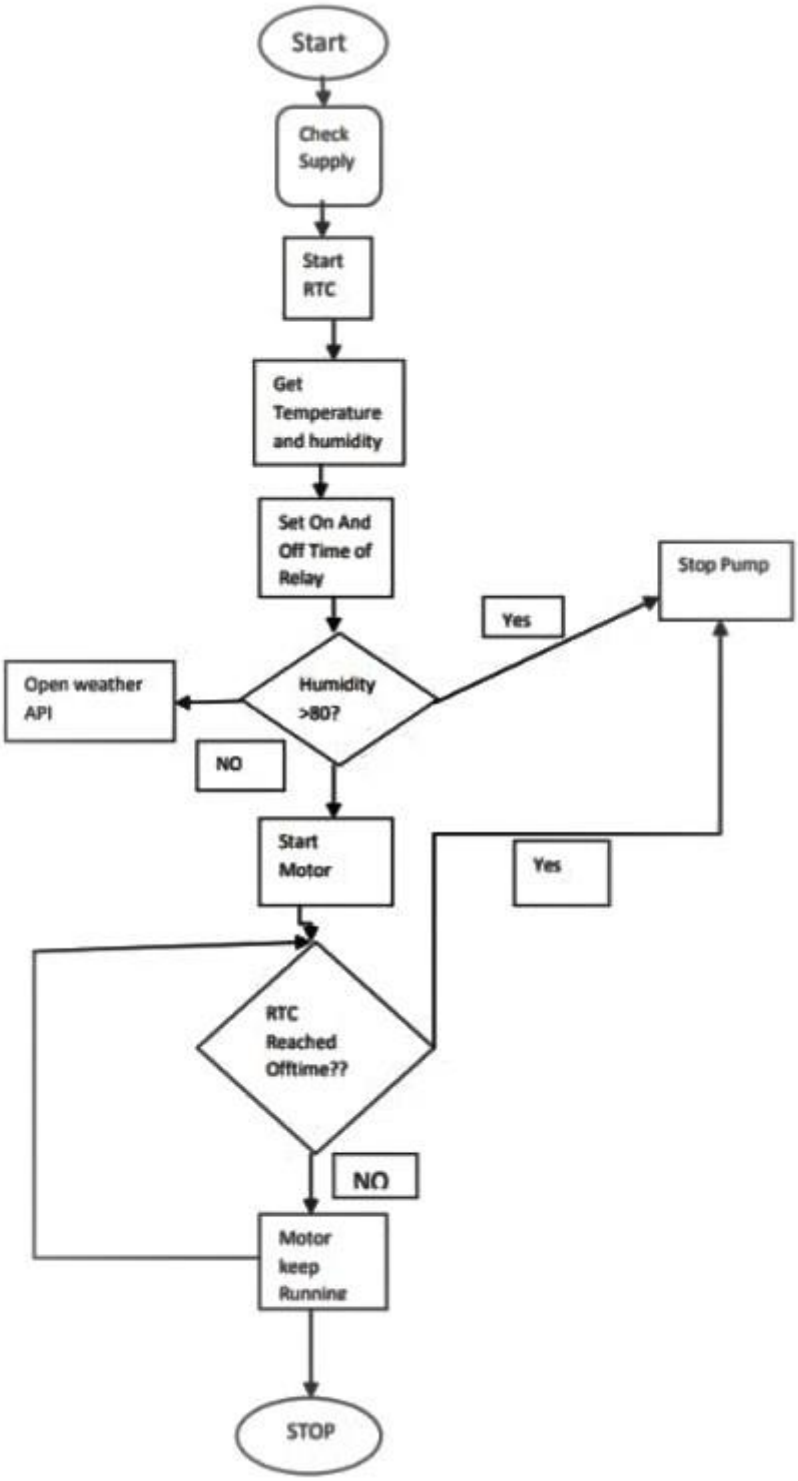
6.3.2 DATA TRANSFER UNIT



6.3.3 ENERGY GENERATION UNIT



6.4 FLOWCHARTS



6.5 PSEUDO CODE

Pseudo code(DHT11, sensor1, sensor2, sensor3, state1, state2, state3, lastState1, lastState2, lastState3, humidity, temperature)

pinDHT11<-2

SoftwareSerial GPRS(2,3)

SimpleDHT11 dht11(pinDHT11)

sensor1_pin<-8

sensor2_pin<-9

sensor3_pin<-10

setup()

Input pins

pinMode<-8

pinMode<-9

pinMode<-10

state1<-digitalRead(8)

state2<-digitalRead(9)

state3<-digitalRead(10)

lastState1<-state1

lastState2<-state2

lastState3<-state3

GPRS.begin

Serial.begin

GPRS print line (AT+CMGF=1)

Delay<-500

looping the process

loop()

byte humidity<-0

byte temperature<-0

```
DHT module
 err<-SimpleDHTErrSuccess
 if (err<-dht11.read(&temperature, &humidity, NULL)~= SimpleDHTErrSuccess)
   print(  Read DHT11 failed, err=  )
   print line (err)
   delay<-1000
 end of if
 a<-(temperature)
 b<-(humidity)

 OUT1<- digitalRead(8)
 OUT2<-digitalRead(9)
 OUT3<-digitalRead(10)

if (OUT1==HIGH)

 if (OUT2==HIGH)

   if (OUT3==HIGH)

     digitalWrite(11,HIGH)
     digitalWrite(12,HIGH)
     digitalWrite(13,HIGH)
     delay<-2000
   digitalWrite(11,LOW)
   digitalWrite(12,LOW)
   digitalWrite(13,LOW)

   else

     digitalWrite(11,HIGH)
```

```
      digitalWrite(12,HIGH)
      digitalWrite(13,LOW)
      delay<-2000
    digitalWrite(11,LOW)
    digitalWrite(12,LOW)
    digitalWrite(13,LOW)

  else
     digitalWrite(11,HIGH)
      digitalWrite(12,LOW)
      digitalWrite(13,LOW)
      delay<-2000
    digitalWrite(11,LOW)
    digitalWrite(12,LOW)
    digitalWrite(13,LOW)
else if(OUT2==HIGH)
     if (OUT3==HIGH)

      digitalWrite(11,LOW)
      digitalWrite(12,HIGH)
      digitalWrite(13,HIGH)
           delay<-2000
    digitalWrite(11,LOW)
    digitalWrite(12,LOW)
    digitalWrite(13,LOW)

     else

      digitalWrite(11,LOW)
      digitalWrite(12,HIGH)
      digitalWrite(13,LOW)
```

```
          delay<-2000
    digitalWrite(11,LOW)
    digitalWrite(12,LOW)
    digitalWrite(13,LOW)
else
  if (OUT3==HIGH)
      digitalWrite(11,LOW)
    digitalWrite(12,LOW)
    digitalWrite(13,HIGH)
         delay<-2000
    digitalWrite(11,LOW)
    digitalWrite(12,LOW)
    digitalWrite(13,LOW)

  else

    digitalWrite(11,LOW)
    digitalWrite(12,LOW)
    digitalWrite(13,LOW)
repeat
   for(GPRS.available())
    Serial.write(GPRS.read())


  lastState1<-state1
  state1<- digitalRead(8)

  lastState2 <-state2
  state2 <-digitalRead(9)

  lastState3<-state3
```

```
state3<- digitalRead(10)

if ( state1 != lastState1 && state2 != lastState2 && state3 != lastState3 )
   sendSMS1()

else if (state1 != lastState1 && state2 != lastState2 && state3 == lastState3 )
   sendSMS2()

else if (state1 != lastState1 && state2 == lastState2 && state3 == lastState3 )
   sendSMS3()

else if(state1 == lastState1 && state2 != lastState2 && state3 != lastState3 )
   sendSMS4()

else if (state1 == lastState1 && state2 == lastState2 && state3 != lastState3 )
   sendSMS5()

else if (state1 != lastState1 && state2 == lastState2 && state3 != lastState3 )
   sendSMS6()

else if (state1 == lastState1 && state2 != lastState2 && state3 == lastState3 )

   sendSMS7()

else

   sendSMS8()


  sendSMS1()
```

```
Print ( MOTOR(S) HAS BEEN TURNED ON    )
GPRS print line (  AT+CMGS=\  +917888420592\    )
  delay<-500
GPRS print(   plant A ,plant B, plant C MOTORS HAS BEEN TURNED ON  )
GPRS print(a)   GPRS print(   *C,   )
GPRS print(b)   GPRS print line (   H  )
 GPRS write( 0x1a )
  delay<-1500

  sendSMS2()

Print (  MOTOR(S) HAS BEEN TURNED ON    )
GPRS print line (  AT+CMGS=\  +917888420592\    )
  delay<-500
GPRS print(   plant A ,plant B  MOTORS HAS BEEN TURNED ON  )
GPRS print(a)   GPRS print(   *C,   )
GPRS print(b)   GPRS print line (   H  )
 GPRS write( 0x1a )
  delay<-1500

 sendSMS3()

Print (  MOTOR(S) HAS BEEN TURNED ON    )
GPRS print line (  AT+CMGS=\  +917888420592\    )
  delay<-500
GPRS print(   plant A  MOTOR HAS BEEN TURNED ON  )
GPRS print(a)   GPRS print(   *C,   )
GPRS print(b)   GPRS print line (   H  )
 GPRS write( 0x1a )
  delay<-1500
```

```
  sendSMS4()

  Print (  MOTOR(S) HAS BEEN TURNED ON    )
 GPRS print line (  AT+CMGS=\  +917888420592\    )
  delay<-500
 GPRS print(   plant B ,plant C  MOTORS HAS BEEN TURNED ON  )
GPRS print(a)   GPRS print(   *C,   )
 GPRS print(b)   GPRS print line (   H  )
 GPRS write( 0x1a )
  delay<-1500

  sendSMS5()

  Print (  MOTOR(S) HAS BEEN TURNED ON    )
 GPRS print line (  AT+CMGS=\  +917888420592\    )
  delay<-500
 GPRS print(  plant C MOTOR HAS BEEN TURNED ON  )
GPRS print(a)   GPRS print(   *C,   )
 GPRS print(b)   GPRS print line (   H  )
  GPRS write( 0x1a )
  delay<-1500

  sendSMS6()

  Print (  MOTOR(S) HAS BEEN TURNED ON    )
 GPRS print line (  AT+CMGS=\  +917888420592\    )
  delay<-500
 GPRS print(   plant A,plant C  MOTORS HAS BEEN TURNED ON  )
GPRS print(a)   GPRS print(   *C,   )
 GPRS print(b)   GPRS print line (   H  )
  GPRS write( 0x1a )
```

```
   delay<-1500

    sendSMS7()

  Print (  MOTOR(S) HAS BEEN TURNED ON    )
  GPRS print line (  AT+CMGS=\  +917888420592\    )
    delay<-500
  GPRS print(   plant B  MOTOR HAS BEEN TURNED ON  )
  GPRS print(a)   GPRS print(   *C,   )
  GPRS print(b)   GPRS print line (   H  )
   GPRS write( 0x1a )

    delay<-1500



 sendSMS8()
  Print (  MOTOR(S) HAS BEEN TURNED ON    )
  GPRS print line (  AT+CMGS=\  +917888420592\    )
    delay<-500
else
 GPRS print(   MOTORS HAS BEEN TURNED Off  )
 GPRS print(a)   GPRS print(   *C,   )
  GPRS print(b)  GPRS print line (   H  )
   GPRS write( 0x1a )
   delay<-1500
```

7.TESTING

Testing is usually defined as to find whether the expected results are matched with the results that we get after testing the project. In testing stage if any errors are found, then these errors can be re-checked and corrected. In our project we mainly focused on two types of testing. They are Functional testing and Structural testing.

7.1 FUNCTIONAL TESTING

Functional testing is one of the type of testing in which all the functions (features) are tested by providing them all types of inputs under different cases and evaluating the output received. In this functional testing, we will also check whether every function works properly and also further functions can be added for improvement based on requirement. In our project, the main functions used in the code are explained below.

1.Void Loop()-:
This is one of the functions used for watering the plants at different intervals of time. By default the humidity and temperature is zero. If there is any increase in temperature or humidity levels around the plants, then information would be sent to DHT Sensor. Based on that the water would be supplied to plants at regular intervals of time. This whole process is a loop.
Different cases are considered in this process. All the plants are considered individual in case of any single plant needs to be watered. In other case, all the plants are considered together if all the plants need to be watered together. And in final case, three plants are taken as two plants each into under different categories. For example if three plants A,B,C are present, then AB is considered together in case if they need to be watered together. Similarly BC & CA are considered together in case if they both need to be watered together.

2.Void sendSMS()-:
This is the other function that we used in our project mostly. This function is used for regular updates about the condition of the plant via text messages(SMS). Whenever the plants will be watered under various cases, all the information such as amount of water supplied, temperature at that time, humidity levels surrounding the plants will be sent to the user. If the temperature is high, then the motors would be turned on for watering the plants and later this information would also be updated to the user via a text message.

7.2. STRUCTURAL TESTING
Structural testing is also a testing which is used for testing the structure of the code. This structural testing helps in finding errors in the code before implementing the code in the project. Different techniques such as Statement coverage, Branch coverage and path coverage techniques are used in our project. All these techniques comes under structural testing. By using the statement coverage technique, we checked for the errors in the programming statements which are crucial for successful running of program. By using the branch coverage technique, we will check all the parts of the code at least once for efficient execution. In path

coverage technique, we will find out all the possible paths so that every statement of the program is executed properly.

7.3 LEVEL OF TESTING

There are mainly four major and important levels of testing. They are

7.3.1. UNIT TESTING

Unit testing generally means that testing every individual component of the project. This unit testing is carried out well in our project too. We have tested every component individually to check whether the components are working properly or not.

7.3.2 INTEGRATION TESTING

Integration generally means combination. Combining different modules together to check their behavior and working is called as Integration testing. In our project also, we have combined different modules and checked how their data is flowing from module to another module.

7.3.3. SYSTEM TESTING

System testing involves checking the overall systems response to the commands given based on the requirements. On the whole it helps in providing the interaction of different parts and modules such as its performance, load. We also have used this system testing in our project to check how the components are interacting and also its performance it gives based on the different loads.

7.3.4. ACCEPTENCE TESTING

Acceptance testing is done generally by the user. In this level of testing, we will check whether the system is accepting all the commands given by the user or not. And also its performance based on acceptance of various commands to check whether it meets all the requirements.

7.4. TESTING THE PROJECT

We have taken into consideration several steps in testing our project. We tested the project manually by putting the setup under different conditions of atmosphere such as temperature (Sun light), humidity (rain) etc. Every plant growth varies based on its genes. Some plants grow fast while some other grow slow. Everyday the plants growth rate is noted and observed. Additional nutrients are also provided and then noted its growth rate and also how those nutrients are helping in their growth rate.

Whenever there is too much sunlight, it indicates that the temperature is high and the motors are sent a command to water the plants regularly to prevent them from getting dried off easily.

We also have tested whether it sends the text message to the user or not so that the user is updated with condition of the plant so that he remains tension free wherever he is in this world.

8. IMPLEMENTATION

8.1. IMPLEMENTATION OF THE PROJECT

To implement a project, several steps have to be taken into account. The project has to be initiated, we need to plan the requirements for the project, and they have to be implemented accordingly based on the plan. Main requirements for the project are mentioned below.

DHT Sensor-:

In our project, DHT11 sensor is used for measuring temperature and humidity levels. Inside this DHT11 sensor, it is equipped with Humidity sensing component and also Thermistor for measuring the temperature. A thermistor is variable resistor which changes its resistance as temperature changes. For measuring humidity, two electrodes and moisture holding substrate is placed between them. So, as humidity changes, the conductivity of substrate changes between these electrodes. This is measured by IC. In this way, DHT11 sensor works.

Soil Moisture Sensor-:

This soil moisture sensor plays a major role in our project. The soil moisture sensor uses capacitance to measure dielectric permittivity of the medium around it. Dielectric permittivity is a function of water content in soil. This sensor creates a voltage which is directly proportional to dielectric permittivity and therefore the amount of water in the soil.

Submersible Water Pump-:

We used Submersible water pump in our project. Submersible water pump is a device that has a sealed motor which is closely coupled to the body. The whole body is submerged in the fluid to be pumped. Its main advantage is it prevents pump cavitation. Submersible water pumps push fluid to the surface.

Motor driver-:

Motor drivers are also used in our project. Motor drivers are made of different components which are integrated inside an IC. Input to this motor driver is the low current signal. Its main function is to convert low current signal to high current signal. This high current signal is then passed to motor. This is the working of motor driver.

Arduino Uno-:

Arduino Uno is an open-source microcontroller board based on microchip. Arduino board contains set of digital and analog pins to give input and output. In order to make connections and give the input to the modules, Arduino is programmed with a code which is full of instructions to different modules. Small error in the Arduino would malfunction the entire project.

8.2 CONVERSION PLAN

Our project collects lot of data such as moisture level in the soil, temperature in the surroundings and humidity level around the plants. It also sends a text message about all these details to the user through Sim module.

Every data that is generated from our project is sent to user and the user analyzes the data he receives whether the plant is healthy or not. All the data generated is in numerical form, right from level of moisture in soil to temperature surrounding the plant.

In our project, DHT sensor will take the temperature and humidity level around the plant as input. If the temperature is high, then it sends a signal to motor to supply water to plants. This is output generated. Along with this it also sends text message to the user that which motor has watered the plant with details. And also soil moisture sensor keeps on sensing the level of water in the soil constantly.

8.3 POST IMPLEMENTATION

When it comes to real world, our project is mainly useful to farmers and the gardeners and also the people who are fond of trees and plants. It also very useful for the people who maintains a nursery. People are addicted to technology these days. New technology is created daily for every small tasks in our daily life.

Our project helps in watering the plants at regular time intervals based on the temperature around these plants and water level in the soil. If the water level is low, soil moisture sensor turns on the motor so that motor supplies water to the plant. When it comes to large scale applications, heavy motors and pumps can be used to supply the water to the plants. And also water can be supplied to plants through pipeline connections by putting the sensors in the soil to detect the level of water in the soil.

8.4. SOFTWARE MAINTENANCE

In our project when it comes to software maintenance, we don’t have much software as our project completely comprised of hardware. In our project, hardware plays a major role. In this project, the values that are received by the sensors from the atmosphere around the plants are updated automatically time-time by the program of the arduino. Every value is recorded by the program itself.

9. PROJECT LEGACY

9.1. CURRENT STATUS OF THE PROJECT

The main aim of our project is to improvise the way of farming and also help in developing lawns, growing nurseries by the people by automation. This way a person can save both time and effort. With ease any person can take care of his plants from anywhere in this world as it automated.

Now-a-days there is lot of deforestation wherever we see due to rapid industrialization. In order to increase afforestation, our project is very helpful by providing them water regularly keeping track of temperature around these plants.

9.2. REMAINING AREAS OF CONCERN

Our project is mainly focused on agricultural fields, farmers, gardeners who grows small areas of trees. It is also helpful in our homes for growing our gardens and watering them. Our project is mainly aimed on growing the plants without any human efforts.

Educating the people and creating awareness among them about smart projects like these can be a huge task especially in rural areas.

9.3. TECHNICAL LESSONS LEARNT

In our project, we learnt how a technology can ease human efforts and save time. We have used various sensors such as DHT sensor, soil moisture sensor, motor drivers in our project and also their working under different conditions is observed.

All these sensors combined together to form a new technology. We also learnt about the smart usage of these sensors for different purposes in our day-to-day life.

9.4. MANAGERIAL LESSONS LEARNT

By doing this project, we also learnt about the usage of the resources we had. Although we had enough resources to use, we do not want use excess resources. We managed the work among ourselves, i.e Team work is one of the things we learnt in this project.

We also learnt to protect nature by not using any kind of chemicals which are harmful to nature in our project. Along with this we also learnt to organize the work and allocate among our team mates.

10. USER MANUAL: -

During installation the agriculturist must know in detail from the installation engineer about various parts of the system, their function and maintenance.

All the plants must be placed in a healthy atmosphere that has a good sunlight coverage and no other factors should disturb the plants from the placed position

The plant should be placed in such a direction that it should face the early sunlight

The water supply to the plants must be provided through a tank because providing water through borewells when there is a power loss is impossible.

Through the observed moisture and humidity level the water pump starts watering the plant

The light dependent resistor must always be dust free because if the LDR was covered by dust the sensor assumes that the light level has reached beyond extreme level

The soil must be removed now and then by observing the growth of the plants compared to the normal ones. If the plants grow effectively then no need of changing soil because there are abundant minerals in it.

Adjust the solar panel in a certain inclined level making the sunlight to fall onto it complete day and to make the water flow down from above the panel.

Avoid placing the plant and the device in a region where the temperature is not under control.

The sensors used in the equipment are not water resistant so the user should protect it from the rainfall or water

The user must regularly keep an eye on the equipment because they may get damaged anytime.

All the preventive and protective measure should be taken to protect it from accidental fall if placed on the roof and from fire .

The water level in the tank should be observed regularly because we may run out of water in the tank and when the sensor wants to water the plants the water pump continuously works resulting in the battery drain.

If any sensor fails that should be immediately replaced after turning the equipment off.

In this project, we will command the arduino/ microcontroller through a command to control the water pump and the rest of whole irrigation process will be automatically controlled by arduino itself.

Do's & Don't:

User can check the all connections should be perfect.

If the soil should have less threshold value, then soil sensor can detect that, and command will be sent to the arduino.

After that arduino will sent a command to water pump then it supplies water to the plant.

Then the power supply should be used through the solar panel and some batteries used.

Also, the various sensor like: - soil moisture sensor, Humidity and temperature sensor can collect the data from the plant.

Another way the user can get the data of the plant and its requirements through a sim card used in the system.

Please place the solar panel and plant on open environment.

Do not place the batteries in too cold or too hot places.

11.SOURCE CODE: -

```
#include <SimpleDHT.h>
#include <SoftwareSerial.h>
int pinDHT11 = 2;
SoftwareSerial GPRS (2,3);
SimpleDHT11 dht11(pinDHT11);
boolean state1, lastState1, state2, lastState2, state3, lastState3;
int a,b;
int sensor1_pin = 8;
int sensor2_pin = 9;
int sensor3_pin = 10;
int OUT1;
int OUT2;
int OUT3;
void setup ()
{
  pinMode(8, INPUT_PULLUP);
  pinMode(9, INPUT_PULLUP);
  pinMode(10, INPUT_PULLUP);
  state1 = digitalRead(8);
  state2 = digitalRead(9);
  state3 = digitalRead(10);
  lastState1 = state1;
  lastState2 = state2;
  lastState3 = state3;
  GPRS.begin(9600);
  Serial.begin(9600);
GPRS.println("AT+CMGF=1");
  Delay (500);
}
void loop ()
{
  byte humidity=0;
byte temperature=0;
  int err = SimpleDHTErrSuccess;
  if ((err = dht11.read(&temperature, &humidity, NULL))! = SimpleDHTErrSuccess) {
    Serial.print("Read DHT11 failed, err="); Serial.println(err); delay (1000);
    return;
  }
  a=((int)temperature);
  b=((int)humidity);
```

```
OUT1= digitalRead(8);
OUT2= digitalRead(9);
OUT3= digitalRead(10);
if (OUT1==HIGH)
{
if (OUT2==HIGH)
{
  if (OUT3==HIGH)
  {
  digitalWrite(11, HIGH);
  digitalWrite(12, HIGH);
  digitalWrite(13, HIGH);
  delay (2000);
  digitalWrite(11, LOW);
  digitalWrite(12, LOW);
  digitalWrite(13, LOW);
  }
  else
  {
    digitalWrite(11, HIGH);
    digitalWrite(12, HIGH);
    digitalWrite(13, LOW);
    delay (2000);
  digitalWrite(11, LOW);
  digitalWrite(12, LOW);
  digitalWrite(13, LOW);
  }
} else
{
  digitalWrite(11, HIGH);
  digitalWrite(12, LOW);
  digitalWrite(13, LOW);
  delay (2000);
  digitalWrite(11, LOW);
  digitalWrite(12, LOW);
  digitalWrite(13, LOW);
}
} else if (OUT2==HIGH)
{
   if (OUT3==HIGH)
   {
    digitalWrite(11, LOW);
    digitalWrite(12, HIGH);
```

```
      digitalWrite(13, HIGH);
      delay (2000);
    digitalWrite(11, LOW);
    digitalWrite(12, LOW);
    digitalWrite(13, LOW);
     }
     else
     {
      digitalWrite(11, LOW);
      digitalWrite(12, HIGH);
      digitalWrite(13, LOW);
      delay (2000);
    digitalWrite(11, LOW);
    digitalWrite(12, LOW);
    digitalWrite(13,  LOW);
        }
  }
else
{
  if (OUT3==HIGH)
  {
    digitalWrite(11, LOW);
    digitalWrite(12, LOW);
    digitalWrite(13, HIGH);
    delay (2000);
    digitalWrite(11, LOW);
    digitalWrite(12, LOW);
    digitalWrite(13, LOW);
  }
  else
  {
    digitalWrite(11, LOW);
    digitalWrite(12, LOW);
    digitalWrite(13, LOW);
  }
}

  While (GPRS.available()) {
    Serial.write(GPRS.read());
  }

  lastState1 = state1;
  state1 = digitalRead(8);
```

```
 lastState2 = state2;
 state2 = digitalRead(9);

 lastState3 = state3;
 state3 = digitalRead(10);

 if (state1! = lastState1 && state2 != lastState2 && state3 != lastState3 ) {
   void sendSMS1();
 }
 else if (state1! = lastState1 && state2 != lastState2 && state3 == lastState3 ){
  void sendSMS2();
 }
 else if (state1! = lastState1 && state2 == lastState2 && state3 == lastState3 ){
  void sendSMS3();
 }
 else if (state1 == lastState1 && state2! = lastState2 && state3 != lastState3 ){
   void sendSMS4();
 }
 else if (state1 == lastState1 && state2 == lastState2 && state3! = lastState3 ){
   void sendSMS5();
 }
 else if (state1 == lastState1 && state2! = lastState2 && state3 != lastState3 ){
  void sendSMS6();
 }
 else if (state1 == lastState1 && state2! = lastState2 && state3 == lastState3 )
 {
  void sendSMS7();
    }
 else
 {
  void sendSMS8();
 }
}

 void sendSMS1()
 {
 Serial.print("MOTOR(S) HAS BEEN TURNED ON ");
 GPRS.println("AT+CMGS=\"+917888420592\"");
 Delay (500);
 GPRS.print(" plant A, plant B, plant C MOTORS HAS BEEN TURNED ON");
 GPRS.print(a); GPRS.print(" *C, ");
 GPRS.print(b); GPRS.println(" H");
```

```
 GPRS.write(0x1a);
 Delay (1500);}

 void sendSMS2()
 {
 Serial.print("MOTOR(S) HAS BEEN TURNED ON ");
 GPRS.println("AT+CMGS=\"+917888420592\"");
 Delay (500);
 GPRS.print(" plant A, plant B MOTORS HAS BEEN TURNED ON");
 GPRS.print(a); GPRS.print(" *C, ");
 GPRS.print(b); GPRS.println(" H");
 GPRS.write(0x1a);
 Delay (1500);}
  void sendSMS3()
 {
 Serial.print("MOTOR(S) HAS BEEN TURNED ON ");
 GPRS.println("AT+CMGS=\"+917888420592\"");
 Delay (500);
 GPRS.print(" plant A MOTOR HAS BEEN TURNED ON");
GPRS.print(a); GPRS.print(" *C, ");
 GPRS.print(b); GPRS.println(" H");
 GPRS.write(0x1a);
 Delay (1500);
 }
 void sendSMS4()
 {
  Serial.print("MOTOR(S) HAS BEEN TURNED ON ");
 GPRS.println("AT+CMGS=\"+917888420592\"");
 Delay (500);
 GPRS.print(" plant B, plant C MOTORS HAS BEEN TURNED ON");
GPRS.print(a); GPRS.print(" *C, ");
 GPRS.print(b); GPRS.println(" H");
GPRS.write(0x1a);
 Delay (1500);
 }
void sendSMS5()
 {
  Serial.print("MOTOR(S) HAS BEEN TURNED ON ");
 GPRS.println("AT+CMGS=\"+917888420592\"");
 Delay (500);
 GPRS.print("plant C MOTOR HAS BEEN TURNED ON");
GPRS.print(a); GPRS.print(" *C, ");
 GPRS.print(b); GPRS.println(" H");
```

```
 GPRS.write(0x1a);
 Delay (1500);
   }
void sendSMS6()
 {
   Serial.print("MOTOR(S) HAS BEEN TURNED ON ");
 GPRS.println("AT+CMGS=\"+917888420592\"");
 Delay (500);
 GPRS.print(" plant B,plant C  MOTORS HAS BEEN TURNED ON");
GPRS.print(a); GPRS.print(" *C, ");
 GPRS.print(b); GPRS.println(" H");
 GPRS.write(0x1a);
 Delay (1500);
 }
  void sendSMS7()
 {
 Serial.print("MOTOR(S) HAS BEEN TURNED ON ");
 GPRS.println("AT+CMGS=\"+917888420592\"");
 Delay (500);
 GPRS.print(" plant B MOTOR HAS BEEN TURNED ON");
 GPRS.print(a); GPRS.print(" *C, ");
 GPRS.print(b); GPRS.println(" H");
 GPRS.write(0x1a);
 Delay (1500);
 }
 void sendSMS8()
 {
 Serial.print("MOTOR(S) HAS BEEN TURNED ON ");
 GPRS.println("AT+CMGS=\"+917888420592\"");
 Delay (500);
 GPRS.print(" plant A, plant C MOTORS HAS BEEN TURNED ON");
GPRS.print(a); GPRS.print(" *C, ");
 GPRS.print(b); GPRS.println(" H");
 GPRS.write(0x1a);
 Delay (1500);
 }
```

12. BIBLIOGRAPHY

- Ali, M. A., Islam, M. S., Sarker, M. N. I., & Bari, M. A. (2015). Study on Biology of Red Pumpkin Beetle in Sweet Gourd Plants. International Journal of Applied Research Journal, 2(1).

- Chavan, C. H., & Karande, P. V. (2014). Wireless Monitoring of Soil Moisture, Temperature & Humidity Using Zigbee in Agriculture. International Journal of Engineering Trends and Technology, 11(10),493–497.

- nurag D, Siuli Roy and Somprakash Bandyopadhyay, "Agro-Sense: Precision Agriculture using Sensor-based Wireless Mesh Networks", ITU-T "Innovation in NGN", Kaleidoscope Conference, Geneva 12-13 May 2008.

- K.N.Manjula B.Swathi and D.Sree Sandhya , Intelligent Automatic Plant Irrigation System.

- R.suresh, S.Gopinath, K.Govindaraju, T.Devika, N.Suthanthira Vanitha, GSM based Automated Irrigation Control using Raingun Irrigation System, International Journal of Advanced Research in Computer and Communication Engineering ,Vol. 3, Issue 2, February 2014.

- Subalakshmi, R., Amal, A.A. and Arthireena, S. (2016) GSM Based Automated Irrigation Using Sensors.

- International Journal of Trend in Research and Development

- D.K. and Ansari, M.H. (2014) Smart Irrigation Control System.

- International Journal of Environmental Research and Development

- R.Hussain, J.Sehgal, A.Gangwar, M.Riyag " Control of irrigation automatically by using wireless sensor network" International journal of soft computing and engineering, vol.3, issue 1, march 2013.

- Parameswaran, G. and Sivaprasath, K. (2016) Arduino Based Smart Drip Irrigation

- System Using Internet of Things.

- International Journal of Engineering Science and Computing

- Parameswaran, G. and Sivaprasath, K. (2016) Arduino Based Smart Drip Irrigation
- International Journal of Engineering