Chapter NO: 01

Introduction to Wireless Sensor Network

A WSN (Wireless Sensor Network) is spatially distributed sensor, that is used to monitor the physical conditions as well as environmental conditions such as sound, temperature, pressure to pass their information through the network to main place. The present networks are bi-directional, also permitting control of sensor activity.

1.1 What is the Wireless Sensor Networks (WSNs)?

A Wireless sensor network can be defined as a network of devices that can communicate the information gathered from a monitored field through wireless links. The data is forwarded through multiple nodes, and with a gateway, the data is connected to other networks like wireless Ethernet.

Wireless sensor network (WSN) refers to a group of spatially dispersed and dedicated sensors for monitoring and recording the physical conditions of the environment and organizing the collected data at a central location. WSNs measure environmental conditions like temperature, sound, pollution levels, humidity, wind speed and direction, pressure, etc.

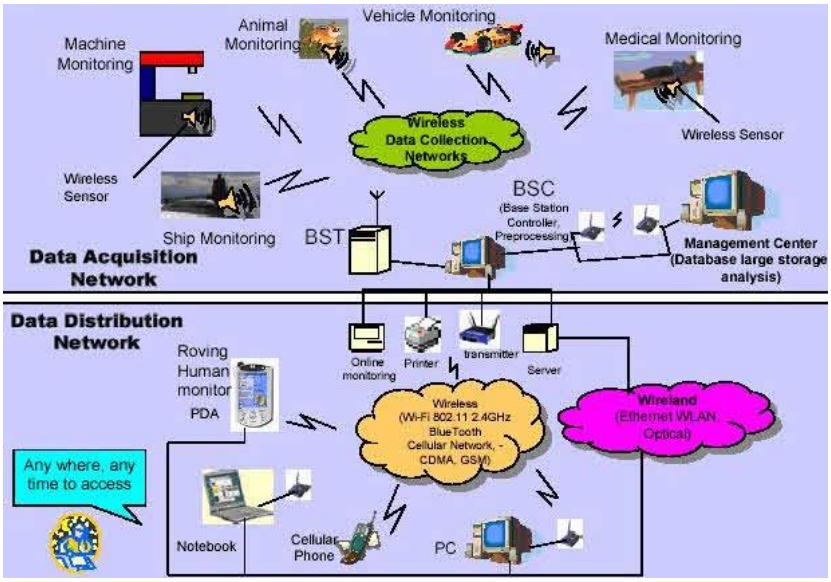

Fig 1.1 Generalised view of Wireless Sensor Network

1.2 Wireless sensor networks (wsn) topologies

There are four common sensor network topologies:

1. Point to point network
2. Star network
3. Tree network
4. Mesh network

1.Point to point network topology

In this topology, there is no central hub. A node can communicate directly to another node. This is the most common topology and it has a single data communication channel which offers a secure communication path. Each device can act as a client and a server in it.

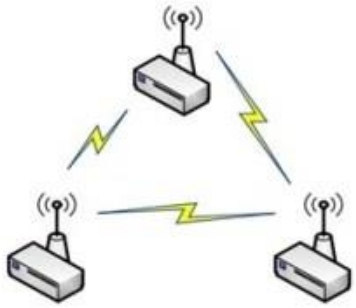

Fig 1.2 Point to Point network topology

2.Star network topology

Unlike point to point networks, a centralized communication hub is present in a star network. Each communication is done through this centralized hub and no direct communications between the nodes are possible. In this case, central hub is the server and the nodes are clients.

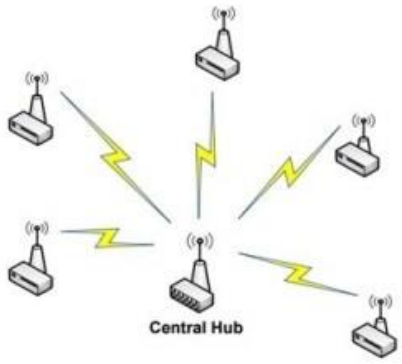

Fig 1.3 Star network topology

3.Tree network topology

This topology is said to be a hybrid of point to point and star topologies. The central hub in it is called a root node or the parent node. Data is passed on from leaf nodes to the parent node. The main advantage of this topology is less power consumption as compared to other networks.

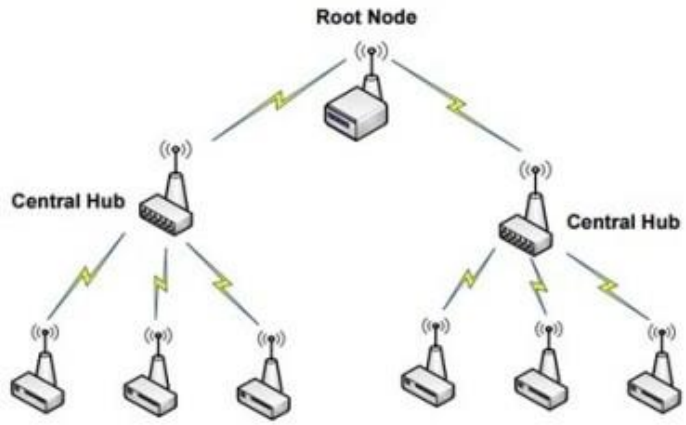

Fig 1.4 Tree network topology

4.Mesh network topology

In the mesh network, the data can 'hop' from one node to another. All the nodes can communicate with each other directly without having to depend on a central communication hub. This is the most reliable structure of network communication because there is no single point of failure in it. But this structure is very complex and requires a lot of power consumption.

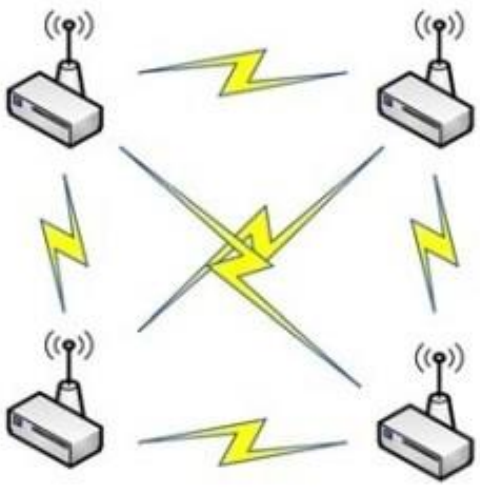

Fig 1.5 Mesh network topology

1.3 Types of WSNs (Wireless Sensor Networks)

Depending on the environment, the types of networks are decided so that those can be deployed underwater, underground, on land, and so on. Different types of WSNs include:

1. Terrestrial WSNs
2. Underground WSNs
3. Underwater WSNs
4. Multimedia WSNs
5. Mobile WSNs

1. Terrestrial WSNs

Terrestrial WSNs are capable of communicating base stations efficiently, and consist of hundreds to thousands of wireless sensor nodes deployed either in unstructured (ad hoc) or structured (Preplanned) manner. In an unstructured mode, the sensor nodes are randomly distributed within the target area that is dropped from a fixed plane. The preplanned or structured mode considers

optimal placement, grid placement, and 2D, 3D placement models. In this WSN, the battery power is limited; however, the battery is equipped with solar cells as a secondary power source. The Energy conservation of these WSNs is achieved by using low duty cycle operations, minimizing delays, and optimal routing, and so on.

2. Underground WSNs

The underground wireless sensor networks are more expensive than the terrestrial WSNs in terms of deployment, maintenance, and equipment cost considerations and careful planning. The WSNs networks consist of a number of sensor nodes that are hidden in the ground to monitor underground conditions. To relay information from the sensor nodes to the base station, additional sink nodes are located above the ground.

The underground wireless sensor networks deployed into the ground are difficult to recharge. The sensor battery nodes equipped with a limited battery power are difficult to recharge. In addition to this, the underground environment makes wireless communication a challenge due to high level of attenuation and signal loss.

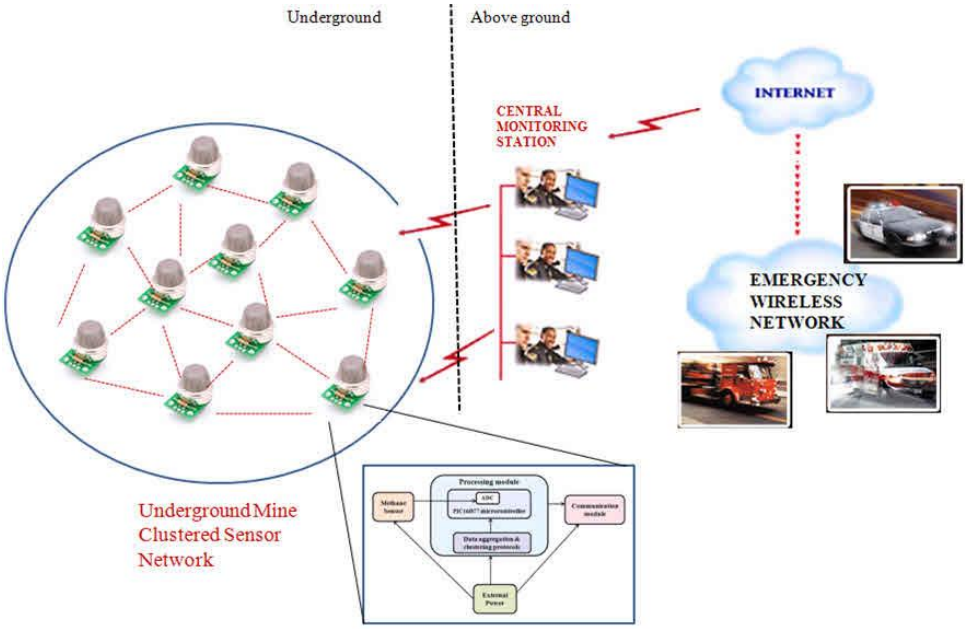

Fig 1.6 Underground WSN

3. Under Water WSNs

More than 70% of the earth is occupied with water. These networks consist of a number of sensor nodes and vehicles deployed under water. Autonomous underwater vehicles are used for gathering data from these sensor nodes. A challenge of underwater communication is a long propagation delay, and bandwidth and sensor failures.

Under water WSNs are equipped with a limited battery that cannot be recharged or replaced. The issue of energy conservation for under water WSNs involves the development of underwater communication and networking techniques.

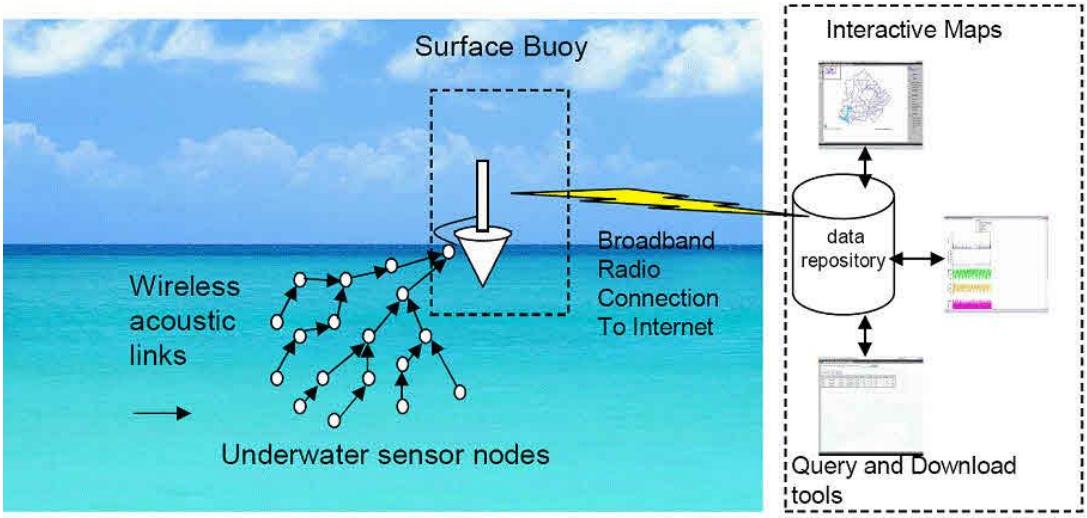

Fig 1.7 Underwater WSN

4. Multimedia WSNs

Multimedia wireless sensor networks have been proposed to enable tracking and monitoring of events in the form of multimedia, such as imaging, video, and audio. These networks consist of low-cost sensor nodes equipped with microphones and cameras. These nodes are interconnected with each other over a wireless connection for data compression, data retrieval and correlation.

The challenges with the multimedia WSN include high energy consumption, high bandwidth requirements, data processing and compressing techniques. In addition to this, multimedia contents require high bandwidth for the contents to be delivered properly and easily.

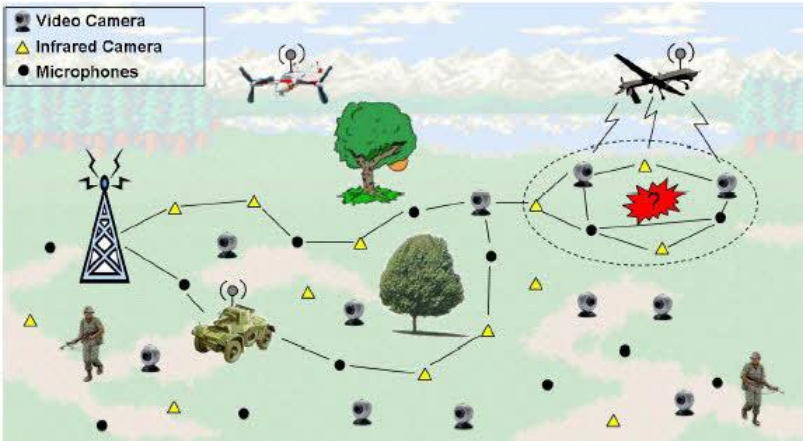

Fig 1.8 Multimedia WSN

5. <u>Mobile WSNs</u>

These networks consist of a collection of sensor nodes that can be moved on their own and can be interacted with the physical environment. The mobile nodes have the ability to compute sense and communicate.

The mobile wireless sensor networks are much more versatile than the static sensor networks. The advantages of MWSN over the static wireless sensor networks include better and improved coverage, better energy efficiency, superior channel capacity, and so on.

1.4 Application:

1. <u>Medical Applications</u>

Nowadays, the health care system is highly complex. The proposed system is designed to provide ultimate solutions to the healthcare using the wireless sensor networks. This proposed system is used to monitor the patient's health by wirelessly using RF technology. It is a very tedious method. In this proposed system transmitting module continuously reads patient's body temperature through a digital temperature sensor, displays it on the LCD screen and sends it to the microcontroller which transmits the encoded serial data over the air by RF (radio frequency) through an RF module.

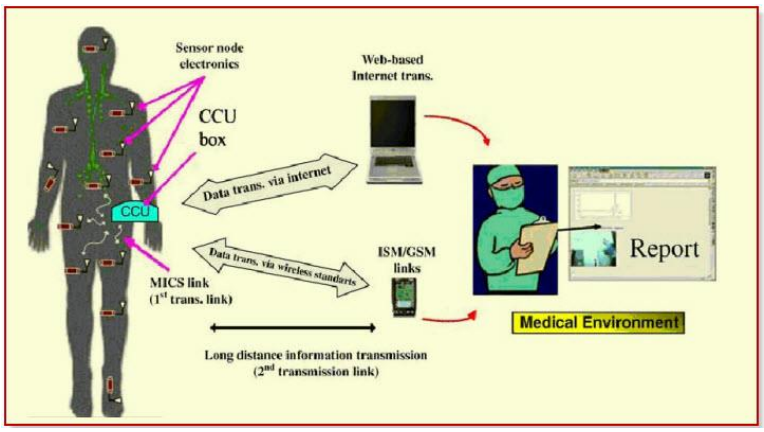

Fig 1.9 Medical application of WSN

2. Home Energy Consumption

A Wireless-sensor-network-based system is used in high energy requiring home appliances like white goods, audio/video devices, communication equipment, air conditioning systems, heating and cooling system, which make our homes one of the most critical areas for the impact of energy consumption in the natural environment. Zigbee home automation is a simple mini project for ece studentswhich can be implemented for automatic control of home appliances

Wireless Sensor Networks (WSN) emerged from advancements in the areas of micro-electro-mechanical system (MEMS) technology, wireless communication, and digital electronics. WSNs devices are small in size, low cost, and require low power to work. The basic structure of WSN sensor nodes as identified

Fig 1.10 Home energy consumption of WSN

3.Military Applications

Wireless sensor networks can be used by the military for a number of purposes such as monitoring militant activity in remote areas and Protecting the force . Being equipped with appropriate sensors these networks can enable detection of enemy movement, identification of the enemy force and analysis of their movement and progress. The focus of this article is on the military requirements for flexible wireless sensor networks.

Based on the main networking characteristics and military use-cases, insight into specific military requirements is given in order to facilitate the reader's understanding of the operation of these networks in the near to medium term (within the next three to eight years). The article structures the evolution of military sensor networking devices by identical-flying three generations of sensors along with their capabilities.

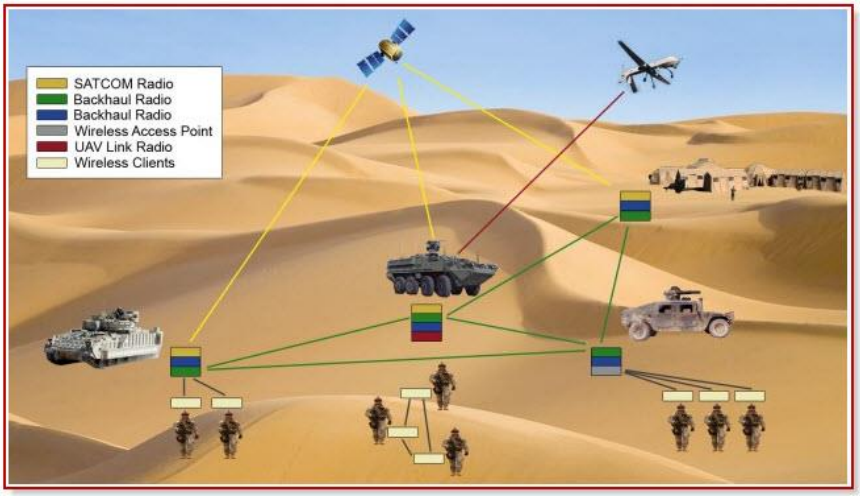

Fig 1.11 Military application of WSN

4.Automated Irrigation System

The main concept of this irrigation system is to turn ON/OFF a motor pump by sensing the soil using WSN in the field of agriculture, the status of the soil can be known by sending an SMS using GSM module.

In the agriculture field, the use of irrigation is mandatory. By using this method we can reduce the manpower. This project is designed with a preprogramed 8051 microcontroller that receives the

signal from the sensor arrangement.This procedure is attained by using an operational amplifier and it acts as interface b/n the sensing device & the 8051 microcontroller.

When the 8051 microcontroller gets this signal, it makes an output to drive a relay for operating the water pump. Using a GSM modem, it also sends an SMS to the authorized person. An LCD display is used to display the status of water pump ON/OFF condition and the soil. The sensing arrangement is designed with two stiff metallic rods and the connections of these rods are interfaced to the control unit. Furthermore, this project can be developed with Xbee or Bluetooth technology, so that when the water pump turns on or off, the data is sent to a mobile phone.

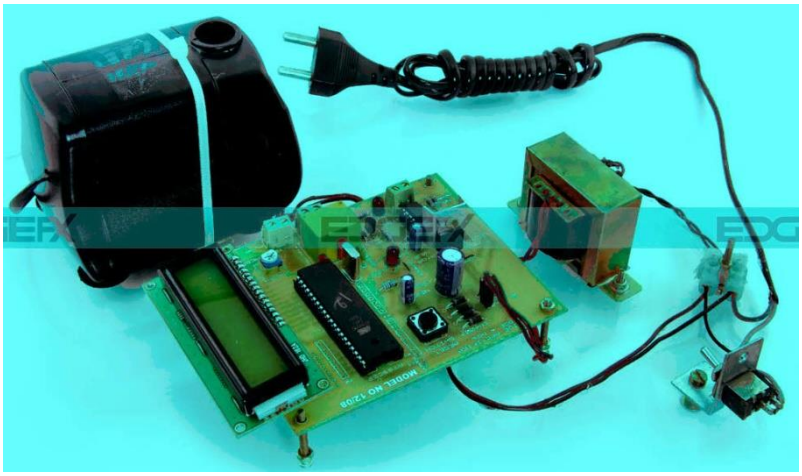

Fig 1.12 Automated Irrigation system using WSN

5. Wireless SCADA

In large scale industries, it is impossible to control the various loads. So this project is developed to control the various processes in industries by SCADA. SCADA is a one type of technology, used to monitor the remote areas without human intrusion.This project uses four temperature sensors to form a WSN, that is located in different places. If the temperature sensor increases at the particular point of GUI, then the relay activates to turn ON/OFF the load to maintain the fixed temperature. There are different sensors can be used in SCADA system to control multiple loads. Wireless Network based Wireless SCADA

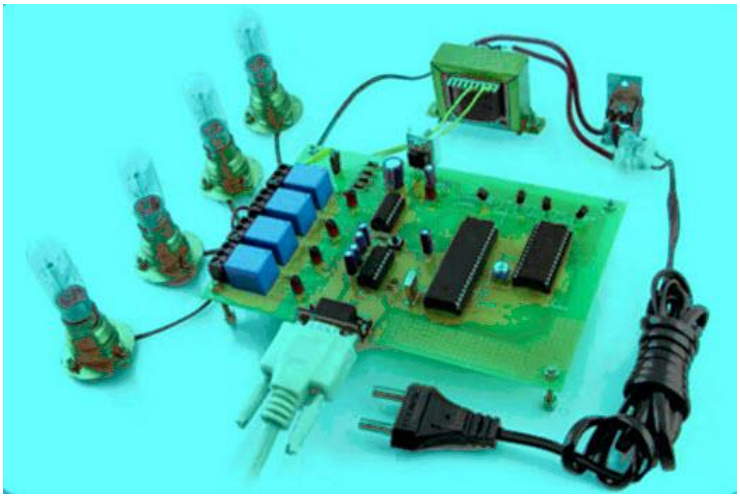

Fig 1.13Wireless SCADA

6.Remote Monitoring System Using XBEE

This XBEE based remote monitoring project is used to protect the transformer by monitoring the three parameters of a generator or transformer like temperature, voltage and current through the sensing devices such as temperature sensor, potential transformer and current transformer to monitor these parameters from a remote location using XBEE. At the transmitter end, three sensors are arranged to make a WSN with a particular range of each parameter. If these three parameters surpass at the fixed limits, then the TX transmits a signal to the receiving end using the XBEE transceiver. Here in this project, a relay is used to switch a warning load and a voice module is used to alert the user. XBEE Based Remote Monitoring System

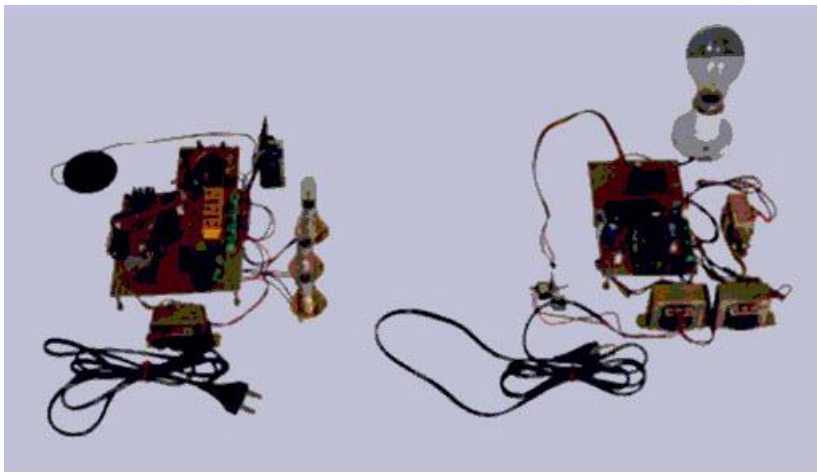

Fig 1.14 Remote Monitoring system using WSN

Chapter NO: 02

Introduction to smart gardening system

2.1 Problem definition:

Due to our busy schedules we don't have time for ourselves then how will we take time to take care of our plants and maintain them.

The problems we usually face while maintaining our plants are:

1. Watering plants on time
2. Providing sufficient amount of water to plants

2.2 Aim:

1. To create a smart musical gardening system to auto monitors our plants.
2. To auto monitor the humidity, amount of water and quality of soil of the garden.
3. To water plants with sufficient amount of water.
4. To play ragas at a fixed time and for fixed interval of time daily.

2.3 Objectives:

1. To improve the environmental sustainability
2. To trim the amount of water your sprinklers use while keeping the yard green
3. To provide comfort to user by reducing manual work
4. To improve overall performance of gardening system without user interactions.
5. To improve the growth of plants

CHAPTER NO: 03

Review of literature

Monitoring plant health is very important for their fast growth. In this busy world, people usually forget to water their plants which leads to bad growth and health of their plants.There are different smart gardening system models which do exists like a plant monitoring and watering system implemented using Evive which continuously monitors the moisture, humidity and temperature of the plant. When the moisture level in the soil goes down to a certain threshold, it automatically starts watering the plant. The notification and the moisture level is shared to user using telnet through WiFi [1].

Soil moisture, temperature, and humidity are three parameters that we can use to build our smart gardening system. One of the parameters of gardening is soil moisture. We should measure soil moisture to ensure our plant grows well. There are many options when it comes to soil moisture sensors. Soil moisture sensor used to measure moisture content in soil [5].

These tutorial explains how to connect DTH 11 model to arduino mega 2560 also tells about the packages required to run the dth11 code like dth11.h package .DTH11 not only measures temperature but also the humidity content of it .It also explains that the value required for DTH-11 [6] I

Automatic watering of plants without any human intervention using soil moisture sensor, motor and arduino board. Moisture sensor takes input from soil and compares it with value with lowest amount of moisture content. If low moisture content then notifies you and switches on the AC water pump which ultimately starts watering of plants .When the moisture content of soil will satisfies the condition then pump is switched off and message is send to user [3].

In this video tutorial the implementation Wifi module esp. 8266 -12 i.e node mcu is done .Nodemcu is use to connect to internet and send data to thingspeak website. For nodemcu we need

to install ESP8266 library and for its connection to thingspeak we need to install thingspeak library .Connection to nodemcu is easy.[15]

This tutorial and website explains how to connect ifttt and thingspeak together to send message on your phone by running an ifttt url .This process consists of two service's webhook service and sms service. The process goes as follow firstly create an applet in ifttt using webhook service then connect it to sms service and create a message youy want to send. Create channel in thingspeak and then thingshttp app .Now connect your ifttt and thingspeak by entering ifttt url obtained in documentation option of webhook service. Then create a react and link your react to your channel and thingshttp app and run at commands present in your code.[11][12]

We know that plants need sufficient amount of water and sunlight for their proper growth. Plants moves in the direction of sunlight as sunlight his absorb by the leaves and then they further prepare their food using photosynthesis process. Just like plant responds to sunlight plants also responds to the musical ragas. Mohanam one of the musical ragas and it is favourite of Balsam plants .Balsam plants moves in the direction where the ragas is being played .It is observed that musical ragas if played daily for half hour then plants grow faster , healthier ,taller and flowering of those plants begins early compare to other plants .[2]

CHAPTER NO: 04

System Description

4.1Design:

Block diagram:

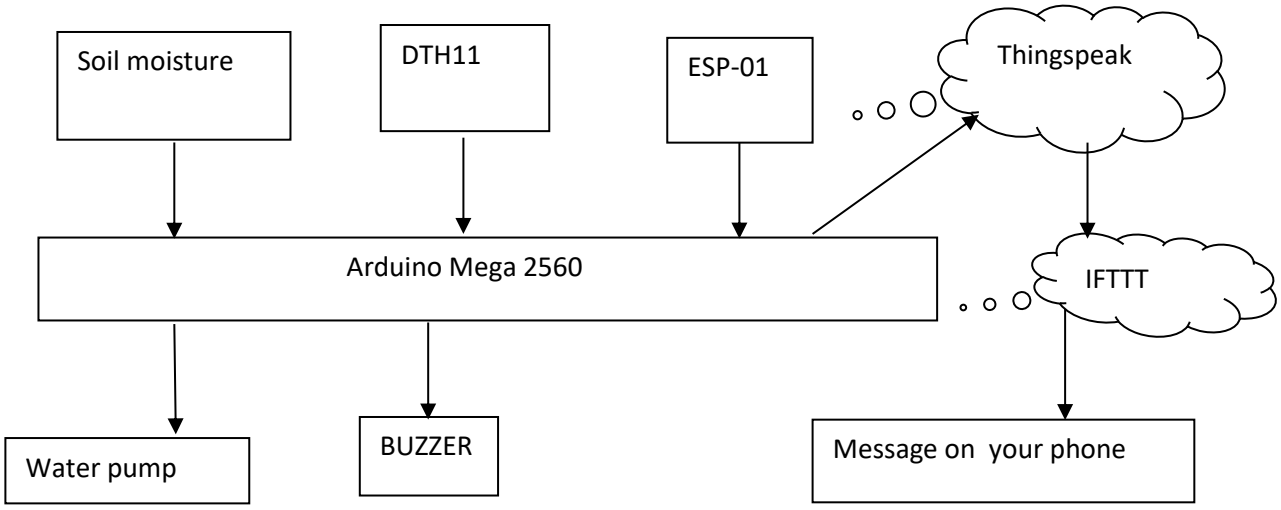

Fig 4.1 Block diagram of Smart Gardening System

Description:

The above figure represents the block diagram of Smart Garden Monitoring System. In this figure DHT11 sensor ,Soil moisture sensor and esp11 act as input and led glow ,buzzer and mobile phone act as output devices. Data taken as input is processed as send by arduino to cloud Thingspeak which is connected to ifttt and then sends sms on your mobile phone. Soil Moisture sensor (to check the water content present in the soil), DHT11 sensor (to check the humidity present in the air), blue LED (to show that plants should be watered), Buzzer (works as a speaker), Wifi Module (to send the SMS on your mobile phone to tell you about your garden condition).

Flowchart:

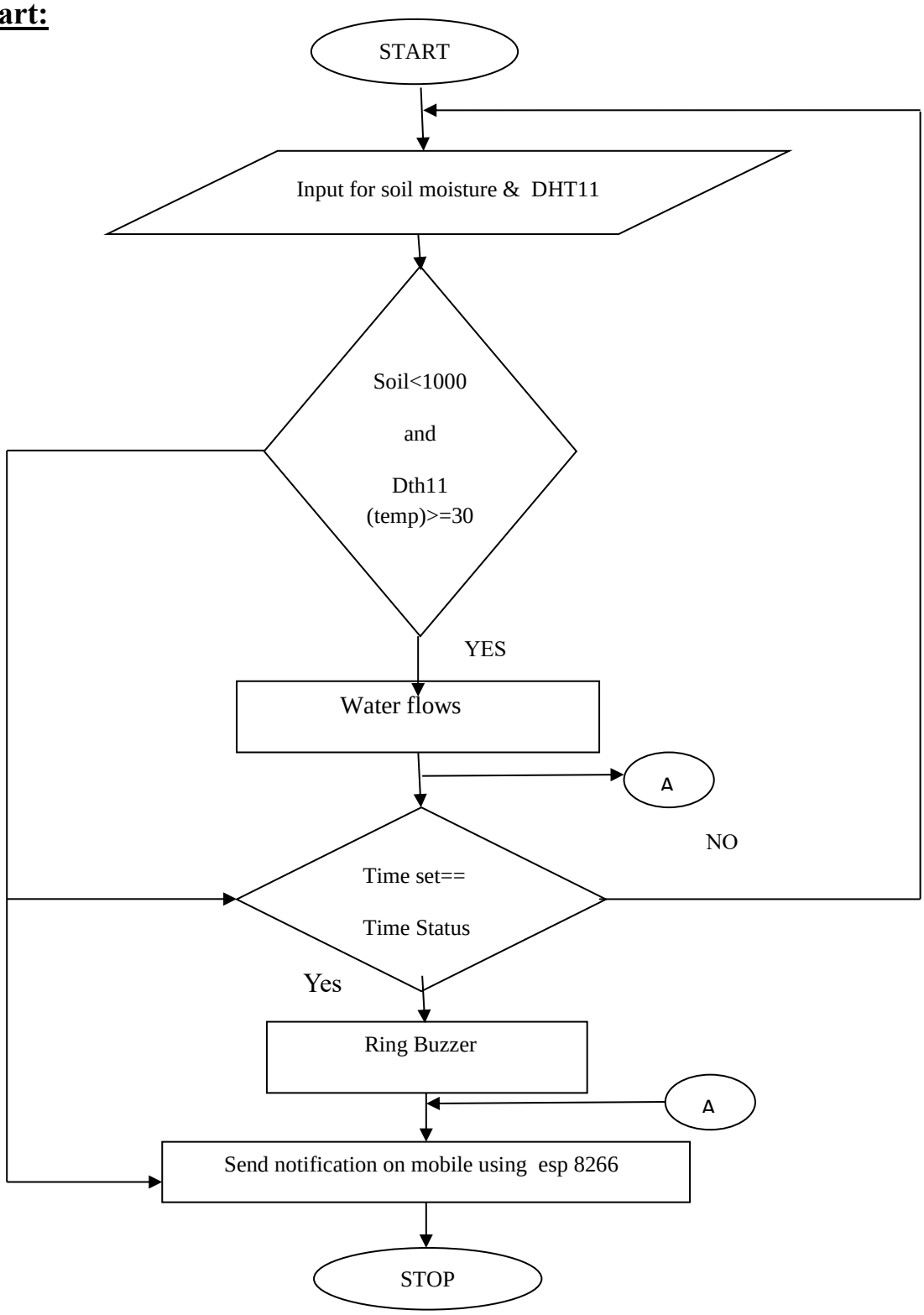

Fig 4.2 Flow-chart of Smart Gardening System

4.2Requirements:

Hardware:

1. **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.

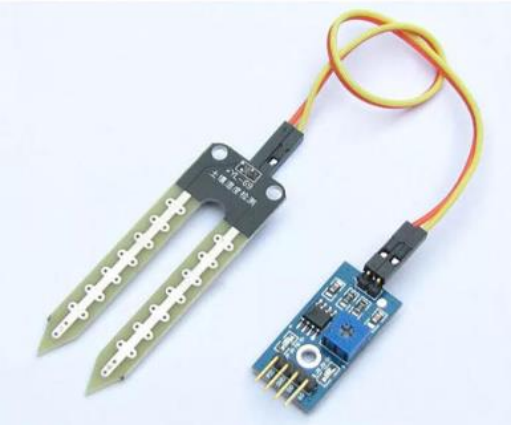

Fig 4.3: Soil Moisture Sensor

2. **Humidity sensor DHT11:**

The DHT11 is a basic, ultra low-cost digital temperature and humidity sensor. It uses a capacitive humidity sensor and a thermistor to measure the surrounding air, and spits out a digital signal on the data pin (no analog input pins needed). Its fairly simple to use, but requires careful timing to grab data. The only real downside of this sensor is you can only get new data from it once every 2 seconds, so when using our library, sensor readings can be up to 2 seconds old.

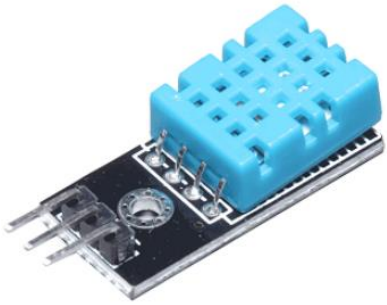

Fig 4.4: DHT 11 Sensor

3. Wi-Fi module ESP-12:

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. it has a voltage regulator to convert from 5v to 3.3v which is the required by the esp21e module.

Fig 4.5: Wifi esp8266-12

4. L293D motor driver:

A motor driver is an integrated circuit chip which is usually used to control motors in autonomous robots. Motor driver act as an interface between Arduino and the motors . The most commonly used motor driver IC's are from the L293 series such as L293D, L293NE, etc. These ICs are designed to control 2 DC motors simultaneously. L293D consist of two H-bridge. H-bridge is the simplest circuit for controlling a low current

rated motor. We will be referring the motor driver IC as L293D only. L293D has 16 pins.

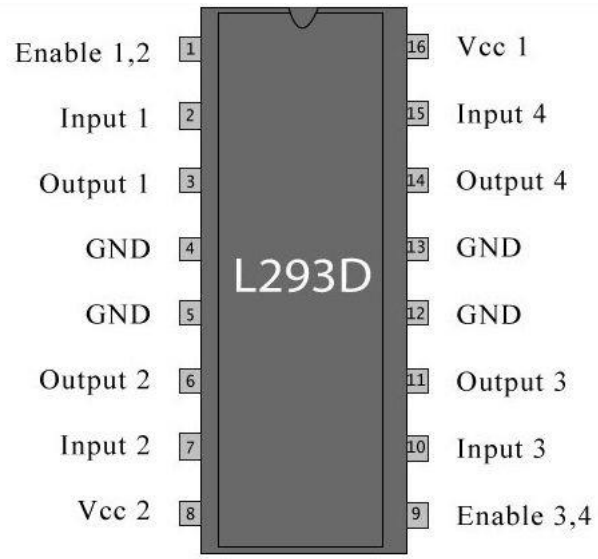

Fig 4.6: L293D motor driver

5. Water pump :

A submersible pump (or sub pump, electric submersible pump (ESP)) is a device which has a hermetically sealed motor close-coupled to the pump body. The whole assembly is submerged in the fluid to be pumped. The main advantage of this type of pump is that it prevents pump cavitation, a problem associated with a high elevation difference between pump and the fluid surface. Submersible pumps push fluid to the surface as opposed to jet pumps having to pull fluids. Submersibles are more efficient than jet pumps.

Fig 4.7: Submersible water pump

6. Blue LED :

A light-emitting diode (**LED**) is a two-lead semiconductor light source. It is a p–n junction diode that emits light when activated. When a suitable current is applied to the leads, electrons are able to recombine with electron holes within the device, releasing energy in the form of photons.

Fig 4.8: Blue LED

7. Piezo Buzzer:

The **piezo buzzer** produces sound based on reverse of the piezoelectric effect. The generation of pressure variation or strain by the application of electric potential across a piezoelectric material is the underlying principle. These buzzers can be used alert a user of an event corresponding to a switching action, counter signal or sensor input. They are also used in alarm circuits.

Fig 4.9: Piezo buzzer

8. Jumper wires:

A **jump wire** (also known as jumper, jumper wire, jumper cable, DuPont wire, or DuPont cable – named for one manufacturer of them) is an electrical wire, or group of them in a cable, with a connector or pin at each end (or sometimes without them – simply "tinned"), which is normally used to interconnect the components of a breadboard or other prototype or test circuit, internally or with other equipment or components, without soldering.[1]

Fig 4.10: male to male

Fig 4.11: female to male

Fig 4.12: female to female

9. Breadboard:

A breadboard is used to build and test circuits quickly before finalizing any circuit design. The breadboard has many holes into which circuit components like ICs and resistors can be inserted. A typical breadboard is shown below: ... To use the bread board, the legs of components are placed in the holes.

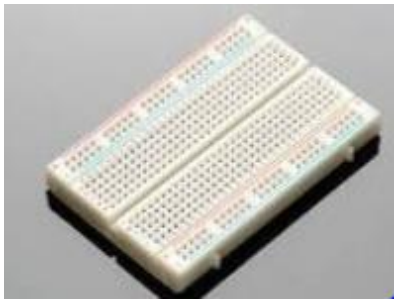

Fig 4.13: Breadboard

10. Arduino Mega 2560 board:

The **Mega 2560** is a microcontroller **board** based on the**ATmega2560**. It has 54 digital input/output pins (of which 15 can be used as PWM outputs), 16 analog inputs, 4 UARTs (hardware serial ports), a 16 MHz crystal oscillator, a USB connection, a power jack, an ICSP header, and a reset button.

Fig 4.14: Arduino mega 2560

Software:

11. Arduino 1.8.0:

Arduino 1.8.0. The open-source **Arduino** Software (IDE) makes it easy to write code and upload it to the board. It runs on Windows, Mac OS X, and Linux. The environment is written in Java and based on Processing and other open-source software.

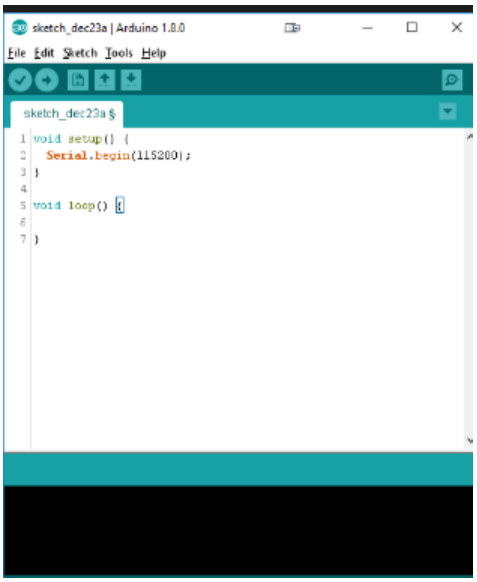

Fig 4.15: Arduino 1.8.0

Cloud Server:

10.Thingspeak

"**ThingSpeak** is an open source Internet of Things (IoT) application and API to store and retrieve data from things using the HTTP protocol over the Internet or via a Local Area Network. ThingSpeak enables the creation of sensor logging applications, location tracking applications, and a social network of things with status updates".

11. IFTTT

If This Then That, also known as **IFTTT** (/ɪft/), is a free web-based service to create chains of simple conditional statements, called applets.

An applet is triggered by changes that occur within other web services such as Gmail, Facebook, Telegram, Instagram, or Pinterest

For example, an applet may send an e-mail message if the user tweets using a hashtag, or copy a photo on Facebook to a user's archive if someone tags a user in a photo.

In addition to the web-based application, the service runs on iOS and Android

4.3Implementation of methodology

1. Deciding the sensors and components required for this project

2. Interfacing Soil moisture sensor with Arduino board as per fig 4.4a and uploading code

3. Interfacing DTH11 humidity sensor with Arduino board according to 4.4b and uploading code

4. Finalising a range of values for DHT11 and soil moisture sensor which are required for the project

5. Interfacing buzzer and time setting

Note: Buzzer resembles the playing of ragas at particular instant of time

1) Making connections according to fig 4.4.c
2) Uploading the code
3) Testing that buzzer rings at time set by user

6. Interfacing soil moisture,dth11and buzzer with Arduino as shown above , uploading the combined code and testing their outputs

7. Interfacing Servo motor and ld239 with above circuit
    1) Connecting led in above circuit
    2) Applying conditions (temp >=30 and moisture <1000) where you want to glow led
    3) Testing the uploaded code

8. Interfacing ESP-8266 with only Arduino board and connecting to internet

    1) Connecting esp 8266-11 as shown in fig 4.4.d
    2) Connecting all the sensors as shown in fig 4.4.e

9. Sending message on phone

Step 1. Create an IFTTT Applet

You can set the IFTTT webhooks service to use web requests to trigger an action. The incoming action is an HTTP request to the web server and the outgoing action is a text message.

1) Create an IFTTT account, or log into your existing account.
2) Create an Applet. Select **My Applets**, and then click the **New Applet** button.
3) Select input action. Click the word **this**.

Fig 4.16: Ifttt new applet screen

4) Select the Webhooks service. Enter Webhooks in the search field. Select the **Webhooks** card.
5) Complete the trigger fields. After you select Webhooks as the trigger, click the **Receive a web request** card to continue. Enter an event name.

   Eg :uses TooCold as the event name. Click **Create Trigger**.

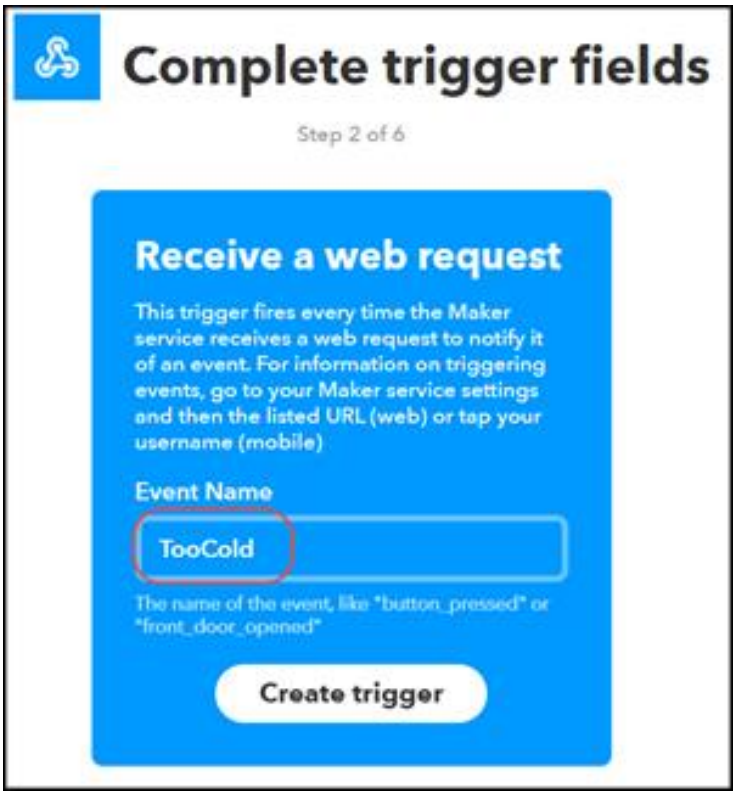

Fig 4.17: Complete trigger fire

6) Select the resulting action. Click the word **that**. The trigger word this is now the Webhooks icon. Enter SMS in the search bar, and select the SMS box.

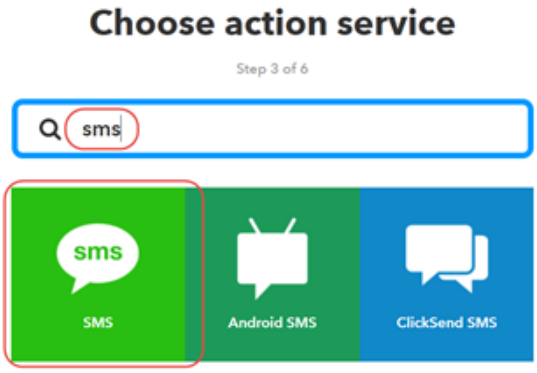

Fig 4.18: Complete trigger fire

7) Select **Send me an SMS**, and then enter the text message information. You can pass data about the event that triggered your message using ingredients. For example, including {{Event Name}} adds the event name to your text message. Click **Create action** to finish the new Applet.

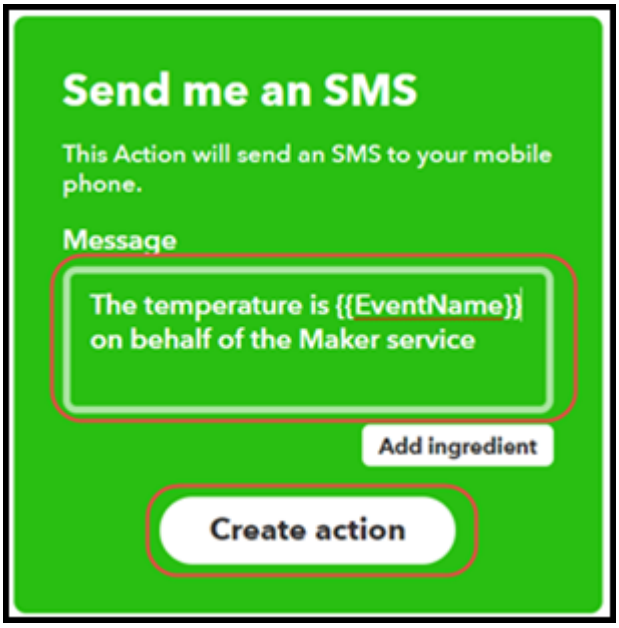

Fig 4.19: Complete trigger fire

8) Retrieve your Webhooks trigger information. Select **My Applets** > **Services**, and search for Webhooks.
9) Select **Webhooks**, and then click the **Documentation** button. You see your key and the format for sending a request.
10) Enter the event name. The event name for this example is TooCold.

    https://maker.ifttt.com/trigger/TooCold/with/key/XXXXXXXXXXXXXXXXXXXXX
    You can test the service using the **test** button or by pasting the URL into your browser. The IFTTT event trigger is not always instantaneous.

    Now create a ThingHTTP to complete the trigger request.

Step 2.Create a ThingHTTP

The ThingHTTP app lets you trigger predefined HTTP requests with an API key and a GET request from the web or from a device.

Eg:

Use ThingHTTP to trigger Webhooks at IFTTT.

1) Choose **Apps** > **ThingHTTP**, and select **New ThingHTTP**.
2) Edit your ThingHTTP settings.
    a. Choose a **Name**.
    b. Enter the **URL** from the Webhooks documentation. The URL for this example has the form https://maker.ifttt.com/triigger/toocold/with/key/xxxxxxxxxxxxxxxxxxxxxx.

c. For **Method**, enter **GET**.

3) Save the ThingHTTP. Now create a React to trigger this ThingHTTP based on your channel data.

Step 3. Create a React to Your Data

The React app can evaluate your ThingSpeak channel data and trigger other events. Create an instance of the React app that triggers when the house is too cold. Choose **Apps** > **React**, and then click **New React**.

1) Choose a **Name**.
2) Select **On Data Insertion** for **Test Frequency**.
3) Choose your temperature channel for the **Condition**.
4) Select the appropriate **field**, in this case Field 1.
5) Set the requirement to **is less than**.
6) Set the temperature level, in this case **50**.
7) Select **ThingHTTP** as the **Action**, and choose the name of the ThingHTTP you defined previously.
8) In **Options**, choose **Run action each time condition is met**.

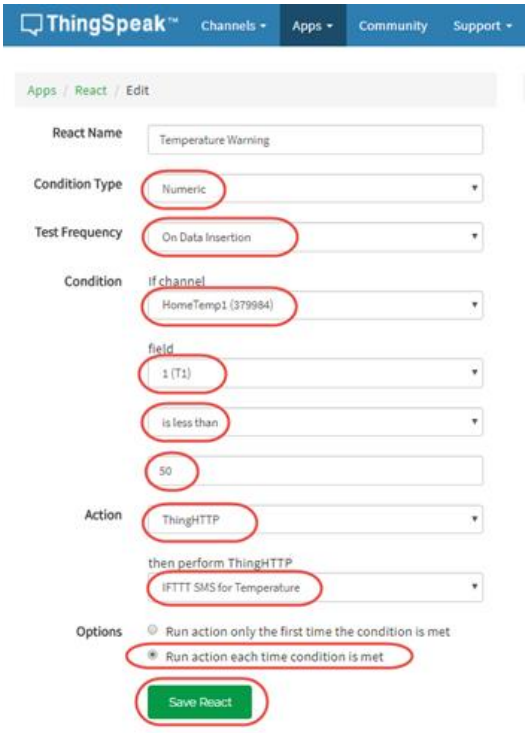

Fig 4.20: Create new app in thingshttp

Step 4. Trigger your Message

Once the temperature in the channel reaches the set point for your React, you receive a text message on your device.

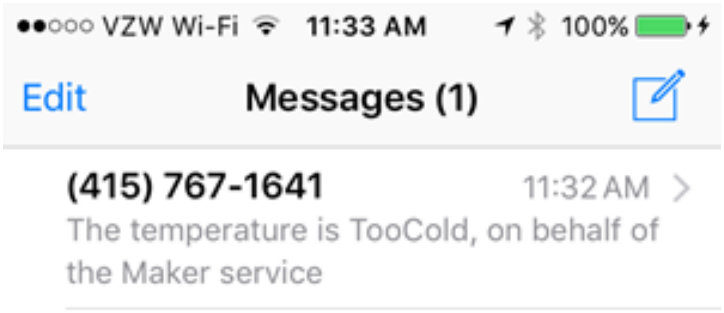

Fig 4.21: Message received on mobile

4.4 Hardware circuit diagram

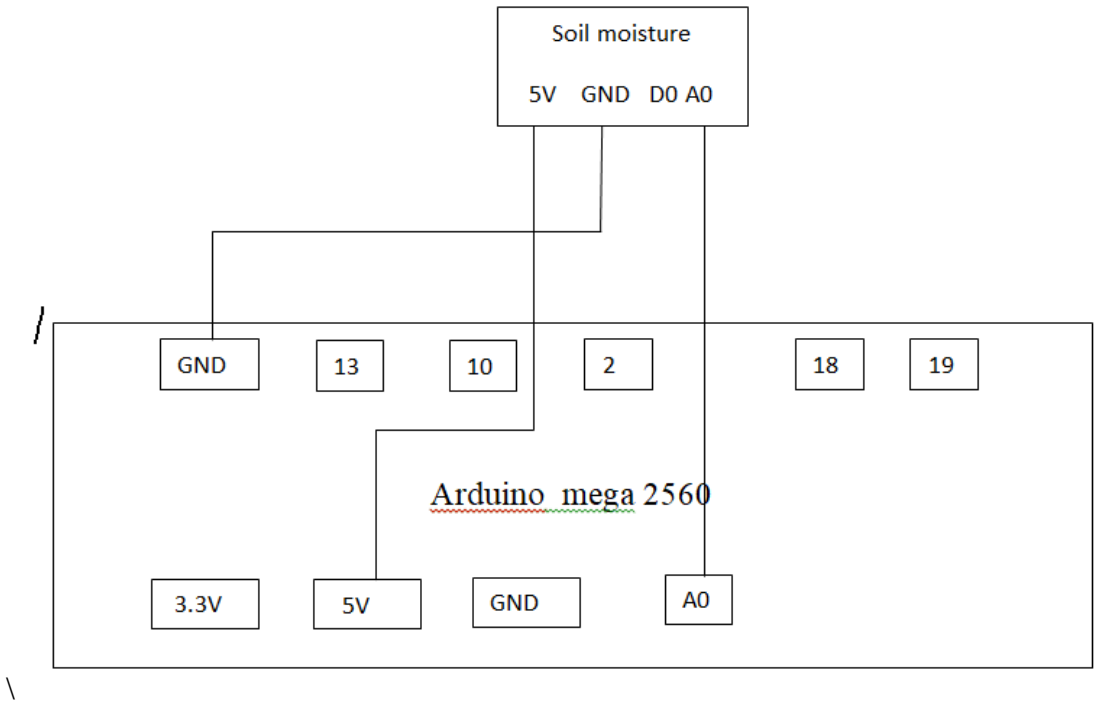

Fig 4.22: Arduino with soil moisture sensor

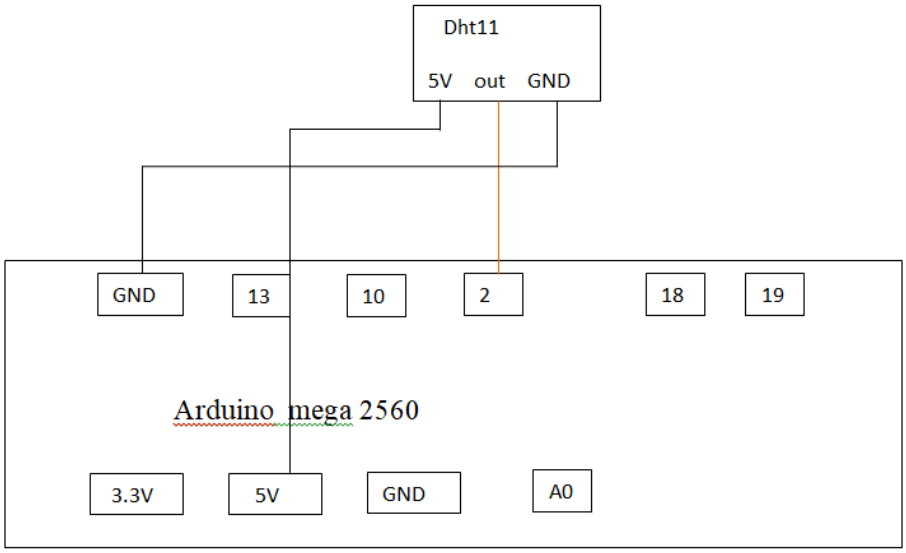

Fig 4.23: Arduino mega 2560 with Dht11

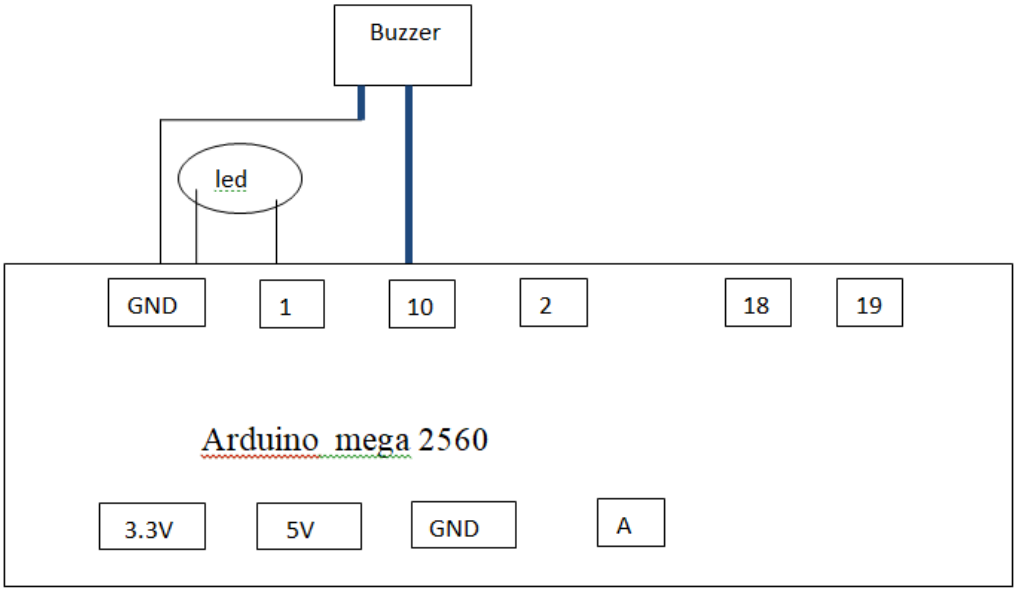

Fig 4.24: Arduino with led and piezo buzzer

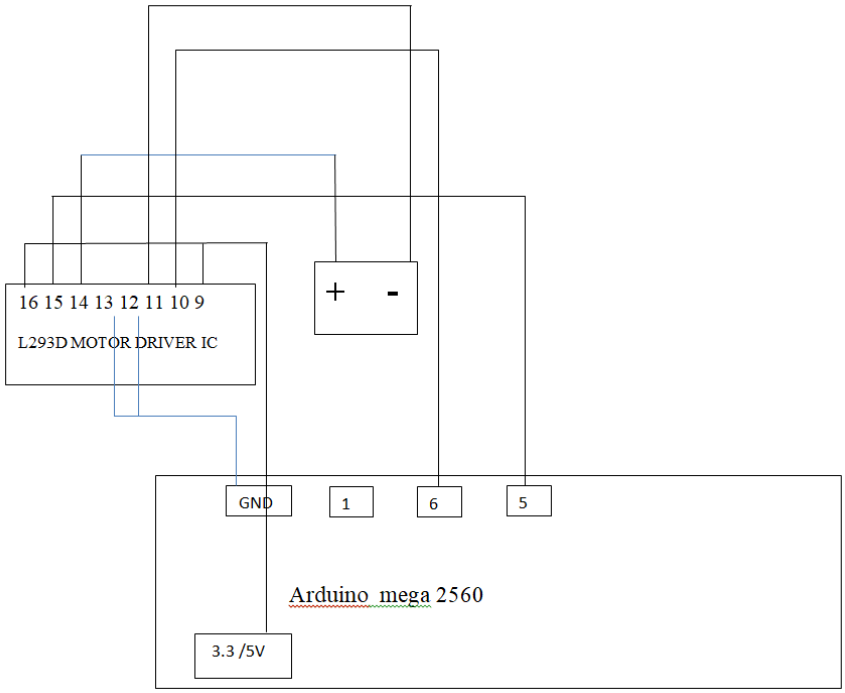

Fig 4.25: pin configuration of L293D motor driver and pump

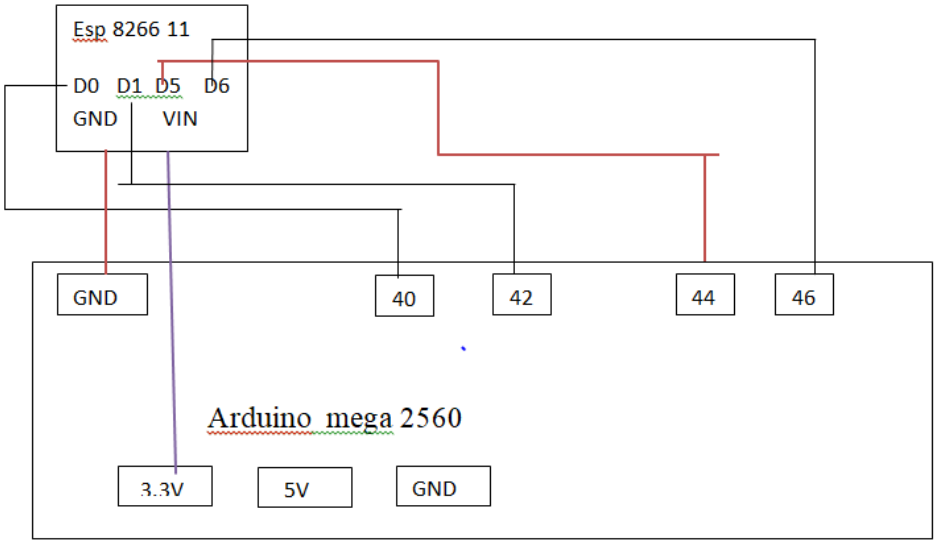

Fig 4.26: pin configuration of wifi module

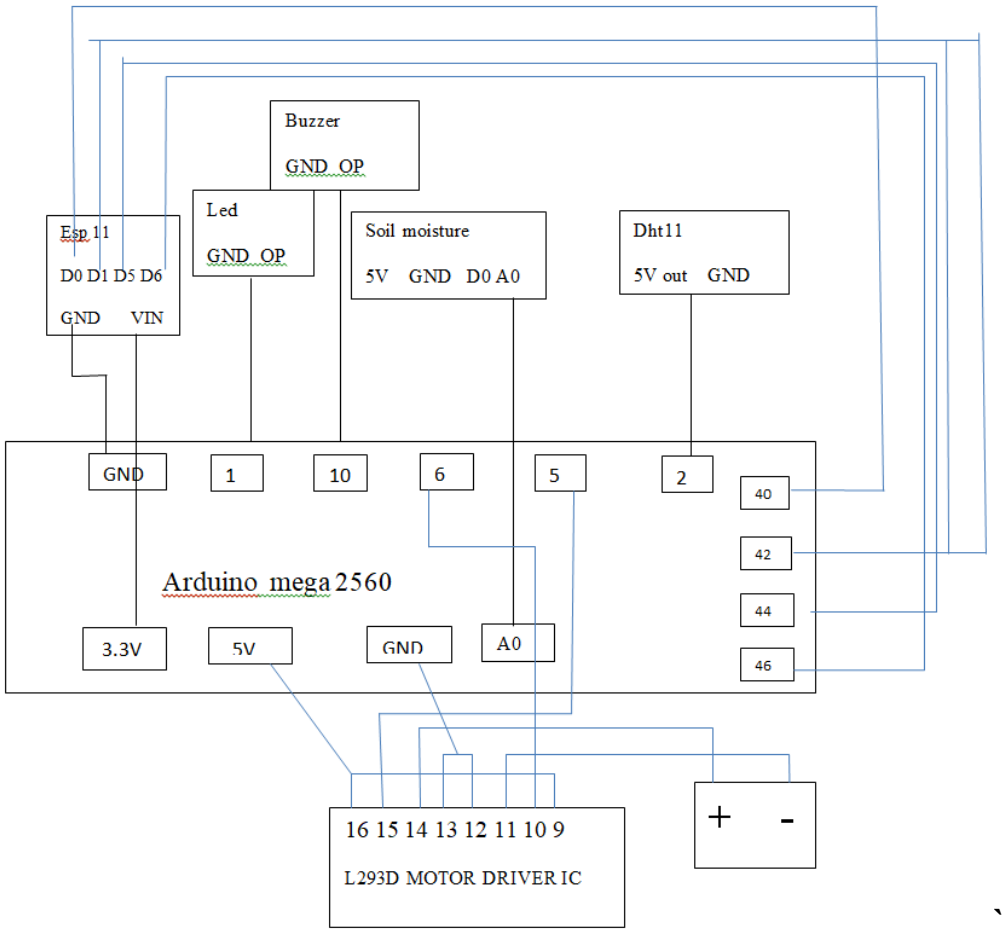

Fig 4.27: Pin diagram of Gardening system

4.5 Code

```
#include <ESP8266WiFi.h>
#include "ThingSpeak.h"
#include <TimeLib.h>
#include <TimeAlarms.h>   // ASCII bell character requests a time sync message
#include "DHT.h"
#define DHTPIN 2
#define DHTTYPE DHT11
# include <Servo.h>

const int motorPin1=5;
```

```
const int motorPin2=6;
const int motorPin3=10;
const int motorPin4=9;
DHT dht(DHTPIN, DHTTYPE);
     String apiKey ="";
     int sensor=A0;
     int pin=10;
    const char* ssid     = "kitty";
    const char* pass = "sidkittudhi";
    unsigned long myChannelNumber = "745731";
    const char * myWriteAPIKey ="UZOIN6RWF3LYH6ZO";
WiFiClient  client;
    int wifiStatus;
     AlarmId id;
String myStatus = "";
    void setup() {

      Serial.begin(115200);\
      delay(200);
      pinMode (pin, OUTPUT);
      dht.begin();
     setTime(10,11,00,17,10,18); // set time to Saturday 8:29:00am Jan 1 2011
      pinMode(pin,OUTPUT);
      pinMode(motorPin1,OUTPUT);
pinMode(motorPin2,OUTPUT);
pinMode(motorPin3,OUTPUT);
  pinMode(motorPin4,OUTPUT);
  // create the alarms, to trigger at specific times
      Alarm.alarmRepeat(10,12,00, MorningAlarm);  // 8:30am every day
      Alarm.alarmRepeat(10,12,10,EveningAlarm);  // 5:45pm every day
      Alarm.alarmRepeat(dowWednesday,8,45,30,WeeklyAlarm);  // 8:30:30 every Saturday
```

```
// create timers, to trigger relative to when they're created
Alarm.timerRepeat(15, Repeats);           // timer for every 15 seconds
id = Alarm.timerRepeat(2, Repeats2);      // timer for every 2 seconds
Alarm.timerOnce(10, OnceOnly);            // called once after 10 seconds// To connect to Wifi
    // We start by connecting to a WiFi network
   ThingSpeak.begin(client);  // Initialize ThingSpeak
    Serial.println();
    Serial.println();
    Serial.print("Your are connecting to;");
    Serial.println(ssid);
    WiFi.mode(WIFI_STA);
    WiFi.begin(ssid, pass);

    while (WiFi.status() != WL_CONNECTED)
    {
      delay(500);
      Serial.print(".");
    }
}
   void loop()
 {
  digitalClockDisplay();
Alarm.delay(1000);
 float h = dht.readHumidity();
// Read temperature as Celsius (the default)
float t = dht.readTemperature();
// Read temperature as Fahrenheit (isFahrenheit = true)
float f = dht.readTemperature(true);
int val=analogRead(sensor);
 delay(2000);
```

```
// wait one second between clock display
if (isnan(h) || isnan(t) || isnan(f))
  {
    Serial.println("Failed to read from DHT sensor!");
    return;
  }
  if (val==1023)
  {
    digitalWrite(11,HIGH);
    digitalWrite(motorPin1,HIGH);
    digitalWrite(motorPin2,LOW);
    digitalWrite(motorPin3,LOW);
    digitalWrite(motorPin4,LOW);

   delay(2000);
  }

  else
  {
      digitalWrite(11,LOW );
      digitalWrite(motorPin1,LOW);
digitalWrite(motorPin2,LOW);
digitalWrite(motorPin3,LOW);
digitalWrite(motorPin4,LOW);
  }
      wifiStatus = WiFi.status();
  if(WiFi.status() != WL_CONNECTED)
  {
    Serial.print("Attempting to connect to SSID: ");
    Serial.println(ssid);
    while(WiFi.status() != WL_CONNECTED)
```

```
  {
    WiFi.begin(ssid, pass);  // Connect to WPA/WPA2 network. Change this line if using open or
WEP network
    Serial.print(".");
    delay(5000);
  }
  Serial.println("\nConnected.");delay(1000); // check for connection every once a second
 }
 ThingSpeak.setField(1, t);
 ThingSpeak.setField(2, val);
 if(t> val){
  myStatus = String("field1 is greater than field2");
 }
 else if(t < val){
  myStatus = String("field1 is less than field2");
 }
 else{
  myStatus = String("field1 equals field2");
 }
 // set the status
 ThingSpeak.setStatus(myStatus);

 // write to the ThingSpeak channel
 int x = ThingSpeak.writeFields(myChannelNumber, myWriteAPIKey);
 if(x == 200){
  Serial.println("Channel update successful.");
 }
 else{
  Serial.println("Problem updating channel. HTTP error code " + String(x));
 }
 Serial.print("Humidity: ");
```

```
  Serial.print(h);
  Serial.print(" %\t");
  Serial.print("Temperature: ");
  Serial.print(t);
  Serial.print(" *C ");
  Serial.print("moisture:");
  Serial.print(val);
  delay(2000);

}

void Morning Alarm()
{
  digitalWrite(pin,HIGH);
  Serial.println("Music time");
   delay(1000);
}

void EveningAlarm() {

  digitalWrite(pin,LOW);
  Serial.println("music time over");
   delay(1000);
}

void WeeklyAlarm() {
  Serial.println("Alarm: - its WEDNESDAY  Morning");
}

void ExplicitAlarm()
{
```

```
  Serial.println("Alarm: - this triggers only at the given date and time");
}

void Repeats()
{
  Serial.println("15 second timer");
}

void Repeats2()
{
  Serial.println("2 second timer");
}

void OnceOnly()
{
  Serial.println("This timer only triggers once, stop the 2 second timer");
  // use Alarm.free() to disable a timer and recycle its memory.
  Alarm.free(id);
  // optional, but safest to "forget" the ID after memory recycled
  id = dtINVALID_ALARM_ID;
  // you can also use Alarm.disable() to turn the timer off, but keep
  // it in memory, to turn back on later with Alarm.enable().
}

void digitalClockDisplay()
{
  // digital clock display of the time
  Serial.print(hour());
  printDigits(minute());
  printDigits(second());
  Serial.println();
```

```
}

void printDigits(int digits)
{
  Serial.print(":");
  if (digits < 10)
    Serial.print('0');
  Serial.print(digits);
}
```

4.6 Final Prototype

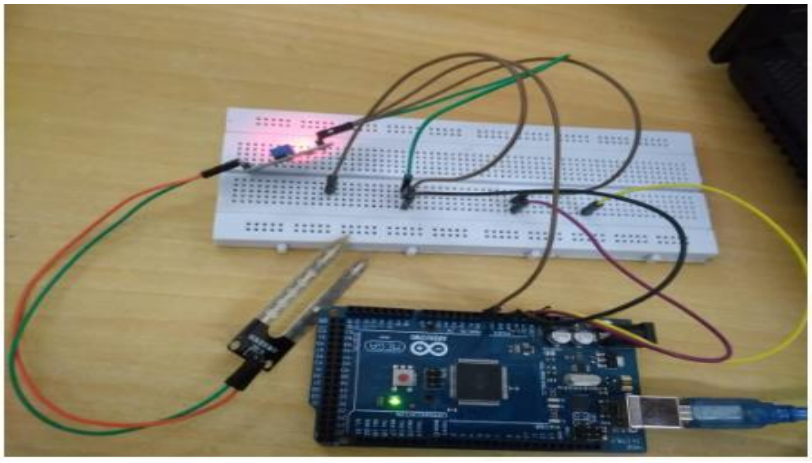

Fig 4.28: soil moisture connection

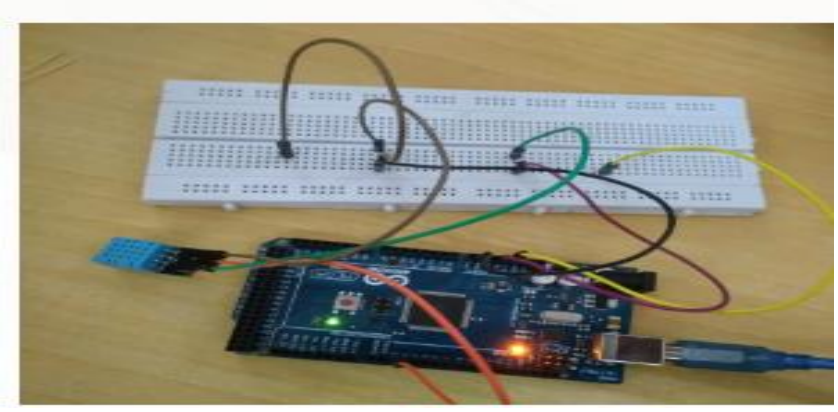

Fig 4.29: DHT11 connection

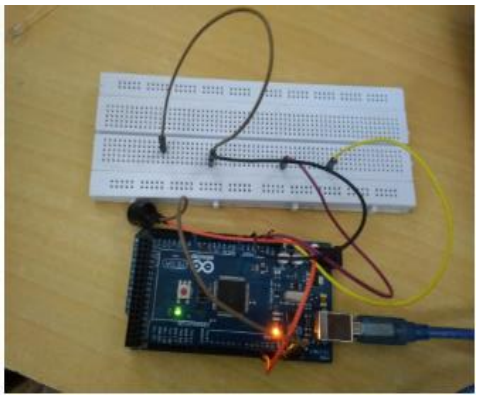

Fig 4.30 : Buzzer connection

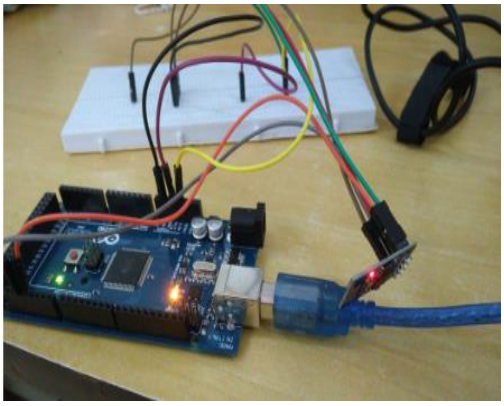

Fig 4.31: WiFi connection

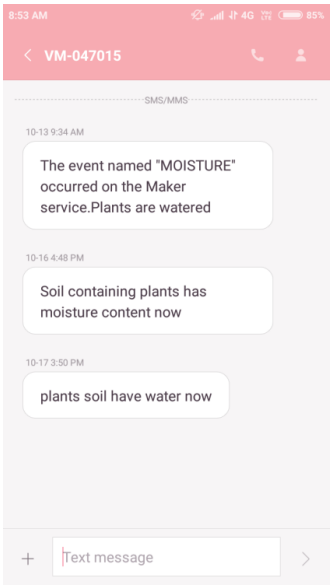

Fig 4.32: SMS

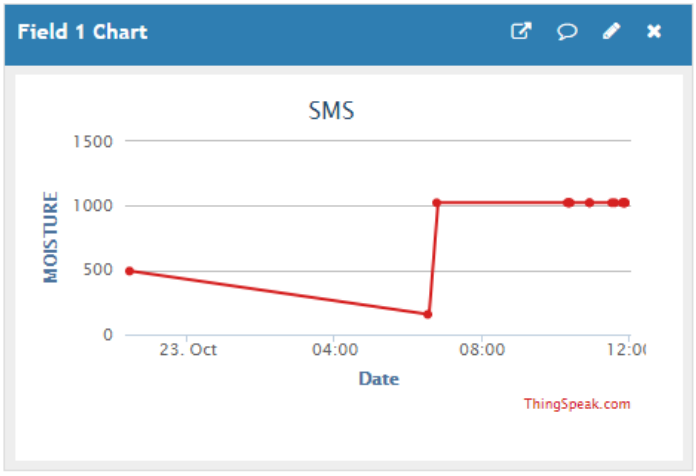

Fig 4.33: Thingspeak data uploaded

4.7 Future scope of project

1. Fertility of soil is an important factor which should be considered during farming or during gardening so if we interface PH sensor with above circuit then it will prove to be a boon
2. If we interface LDR with current smart gardening monitoring system then it will automatically switch on lights in evening and switch off lights as the day will occur this will help to save your electricity
3. Colour detection sensor can be used which will take the color of leaves as its input and will compare it with color values stored within it to detect whether the plant is healthy or not .If plant is not healthy then it will notify you about its health

4.8 Constraints for real time

1. This idea will prove as one of the reason for unemployment as it is auto monitored
2. In Future use of PH sensor for small garden won't be preferred as it is costly compared to other sensor

3. Instead of actually playing ragas using mini speakers we have used buzzer due to limited supply of resources

4. Led is used instead of water DC motor for watering plants just to keep it a simple look.

Conclusion

Smart Gardening System using IOT with the help of Arduino Mega helps to ease the most tedious job of gardening for plants lovers who are in a time of rush. This system monitors various garden parameters and informs the user about the details of the garden through their smart phones. It also helps to solve many issues occurring in the existing plants watering and gardening system. The user can control and monitor the environment of garden using the mobile phones. The controller in the system provides an economic and efficient platform to implement the plant monitoring using IOT.

Advantage:

User can monitor the garden using internet from far distances during leisure or busy time or whenever necessary. The system will provide feedback control system which will monitor and control all the activities of plants growth and irrigation system efficiently.

Use:

In agricultural sector for irrigation purpose.

References

1. https://www.hackster.io/evive-an-opensource-embedded-platoform/plant-monitoring-and-watering-system-using-evive-d48677 accessed on 01 August 20:02:48
2. music idea- Std 8th SSC BOARD English Balbharati textbook Year: Revised syllabus 2018
3. https://www.youtube.com/watch?v=nUHizmtyt74 accesed on 18 September 20:02:28
4. http://www.circuitbasics.com/how-to-set-up-the-dht11-humidity-sensor-on-an-arduino accesed on 5th August 2018 22:02:58
5. https://www.hackster.io/evive-an-opensource-embedded-platoform/plant-monitoring-and-watering-system-using-evive-d48677 accessed on 01th August 2018 23:32:18
6. http://web.csulb.edu/~bpenzens/se4s/smart.html accessed on 5[16h] August 2018 23:32:18
7. https://www.packtpub.com/mapt/book/hardware_and_creative/9781787286429/1/ch01lvl1sec10/se nsor-devices-for-a-smart-gardening-system accesed on 5th August 2018 23:32:18
8. https://www.cnet.com/news/smart-garden-buying-guide/ accesed on 19th August 2018 10:32:18
9. https://www.youtube.com/watch?v=xuk7QmY45-U accesed on 19th September 2018 23:32:18
10. 10. https://www.youtube.com/watch?v=D-N-gRaLPQM accessed on 19th October 2018 16:04:18
11. https://www.hackster.io/evive-an-opensource-embedded-platoform/plant-monitoring-and-watering-system-using-evive-d48677 accessed on 19th October 2018 16:04:18
12. https://youtu.be/nUHizmtyt74 accessed on 19th October 2018 16:01:18
13. https://in.mathworks.com/help/thingspeak/use-ifttt-to-send-text-message- notification.html accessed on 19th October 2018 16:04:18
14. https://www.esp8266.com/viewtopic.php?f=27&t=9453 accessed on 10th April 2019 15:45:15
15. https://www.youtube.com/watch?v=-6Nb5kL43GY accessed on 10th April 2019 16:15:22