## **SMART PLANTING USING WIFI**

## **ABSTRACT:**

## Smart Planting is seen as one of the most important tasks in any farming or agriculture-based environment. With the inception of Ambient technology systems, there have been a rise in the ambient intelligent based devices. Integration of such an ambient intelligent system with smart planting makes farming easier.

## The project explains about the implementation of a smart planting monitoring system which makes use of the NodeMcu which proactively handles the smart planting system. The given implementation works along with the cloud based server and a mobile based device which helps the user to control and see the status of the plant which is being monitored by the hardware devices.

## The given circuitary detects the changes in the health parameters of the plant such as temperature, humidity, light intensity and soil moisture and updates the data into the cloud there by sending the information to the owner if the changes need any supply of water and nutrients. The supply of water can be performed in two ways.one is by using Google Assistant and the other one is by using Blynk app.

## **INTRODUCTION:**

## IOT or internet of things is an upcoming technology that makes use of internet to control the all types of devices that are connected to the internet. internet of things projects have been proposed on existing system improvements and new innovative solutions to different problems. Internet of things is a technology which encounters remote monitoring and controlling of devices which being connected to it wirelessly, here the concepts of wireless sensors networks and Ethernet protocols are made of use. IoT can also makes use of Wireless sensor networks nanotechnology and Miniaturization. Internet of things will provide seamless connectivity between the things in virtual world with real world thereby ensures anytime, anywhere, anything communications. IoT will help the user to approach various applications in a smarter way which might be a smart home, smart agriculture, smart industries or enterprise. The user can check the status of any resources in the network.

## The aim of this project is to demonstrate that the Smart planting system can be used to reduce the Human force. This system consists of a distributed wireless network of soil moisture, temperature and Humidity sensor. Moisture sensor was installed near the roots of the plant and daylight sensor was installed further away to clearly detect the sunbeam. These sensors send their data to the NodeMcu to analyse. Main purpose of this project is to save the water and to increase the production of trees and plants in the garden by monitoring the growth of plants. With the help of sensors, controlled by NodeMcu, we designed a system to sprinkle accurate amount of water by detecting soil moisture, daylight intensity. Hence the soil moisture is a key variable that can be used to determine the quantity of water needed. Besides, the availability of the amount of daylight is also very crucial for a tree. Wrong timing of watering can cause more harm rather than benefit.

## **EXISTING SYSTEMS:**

## **1.AUTOMATIC IRRIGATION SYSTEM:**

## The main principle behind the system is in connecting the soil moisture sensor, which was previously embedded into the plant, to the Arduino microcontroller ,which is also connected to the other electronic components.

## Moisture measurment is done by the sensor which forwards the information and parameters regarding the soil moisture to the microcontroller, which controls the pump. If the level of soil moisture drops below a certain value, the micro controller sends the signal to the relay module which then runs a pump and certain amount of water is delivered ,the pump stops doing its work. Power supply has a task to power the complete system and the recommended voltage should respect the input supply voltage range for the microcontroller, that is from 7v to 12v.

## **2.AUTOMATIC DRIP IRRIGATION SYSTEM:**

## In this system the water will be supplied to the plants at a particular interval in a day regardless of the season. This system doesn’t need any sensors because the water will be supplied with the help of motor and relay module in a particular duration. It doesn’t considers any moisture values.

## **PROPOSED SYSTEM:**

## The proposed project explains about the smart planting using WIFI with the help of Iot and the microcontroller Arduino. All the health parameters of the plants like temperature,humidity,soil moisture are controlled with the help of DHT11 and soil moisture sensors and are also interfaced with the NodeMcu (Arduino uno+ESP8266) and the information about the garden can be directly monitored and controlled by the owner through his/her mobile phones using Iot.

## The proposed system is generally for the farmers and the people who loves gardening but are busy in their day-to-day lives. It is not possible for the people to maintain the farms regularly. This will cause improper growth of the crops. This system will helps to solve those worries. Thus by the installation of this application on the owner’s smart phone the user cannot forget about watering the plants.

## Moreover, the proposed system also tracks the humidity, soil and temperature. It then uploads this information to cloud through the database. Soil moisture sensor measures the volumetric water content in the soil and alerts the user to take required action if it is Less than the desired value. The system helps to save the water, the utility bills, also the thrust of plants is fulfilled. The application can also used by the user to control the motor manually.

## Based on the sensor reading the moisture value will be updated to the user with the help of GPRS that through the GSM module by sending the message and voice call to the user notifying that the plant doesn’t have sufficient water. The user then after receiving the notification can switch on the motor with the help of Google Assistant or by using Blynk app whenever the water is needed for the plant. After a particular duration if the sensor value meets the particular value then the moisture value will again updated to the use notifying that your plant is safe through the GSM module. The user the can switch off the motor by using Google Assistant or Blynk app.

## This system is also made to display the different charts based on the sensor reading and a graph is plotted to check the plant growth. The open source cloud database(Thing Speak) is used by this system to reduce the cost of the issues of storing large amount of data. All the data are gathered by the sensors (temp&humidity,soil moisture) will be refreshed for every two minutes. All the platforms used in this project are of open source and free to use and the primary aim of this project is to reduce the implementation cost compared to any other system.

## The front end application and the cloud are 2-way connected to retrieve the data as well as give command to hardware devices to water the plants.

## **SYSTEM ARCHITECTURE:**

## 

## **DESCRIPTION:**

## The architecture consists of Three layers.

## 1.physical layer:

## In this layer all the devices that are used for the project are connected to each other in a desired manner.It consists of soil moisture sensor to measure the moisture content,DHT11 sensor to know the temperature and humidity,a GSM module to send the notifications to the user and a relay module to control the motor.

## 2.Network layer:

## The network layer is used to transmit the data to the cloud and to store.

## It consists of a web server to store the information in the cloud , a router to provide the internet. A mobile to control the motor with the help of Google Assistant or Blynk.

## 3.Application/Presentation layer:

## This layer consists of a web browser(ThingSpeak) to store and present the updated data of soil moisture,temperature and humidity values in the graphical format.

## **REQUIREMENTS:**

## 1.NodeMcu:

## NodeMcu is the key component of the project.it consists of a microcontroller and an ESP8266 wifi module use to transmit and retrieve the data to/from the hardware devices.

## 2.DHT11:

## It is used to measure the temperature and humidity values in the surroundings of the plant.

## 3.Soil Moisture Sensor:

## This sensor is used to measure the volumetric content of the moisture and it is also

## used to pass the signal to the relay module.

## 4.GSM Module SIM900A:

## This component acts like a GPRS system to send the notifications to the user by retrieving the sensor values to the user.

## 5.LCD Display:

## It is used to display the sensor values(temperature, humidity and soil moisture) to the user externally.

## 6.Blynk App:

## Blynk app is used to control the motor. It also acts like a bridge between motor and Google Assistant.

## 7.Relay Module:

## It is used to retrieve the signals from the soil sensor and to perform the actions on motor.

## 8.USB CABLE:

## To upload code into the Key component, NodeMcu.

## 9.Bread Board:

## It is used to make the connections between the hardware components.

## 10.Jumper wires:

## These are used to make the connections between the hardware components.

## **SOFTWARE:**

## **ARDUINO IDE:**

### The Arduino ide is a cross platform java application that serves as a code editor and compiler and is also capable of transferring firmware serially to the board. The development environment is based on processing, an IDE designed introduce programming to artists unfamiliar with the software development. It is based on C language.

### The open source Arduino software makes easy to write the code and upload it to the board. It works on the Windows, Mac OS X and LINUX.

### .

### **CODING:**

## #include<SoftwareSerial.h>

## #include <DHT.h>

## #include <ESP8266WiFi.h>

## #include <WiFiClient.h>

## #include <ThingSpeak.h>

## #define BLYNK\_PRINT Serial

## #include <ESP8266WiFi.h>

## #include <BlynkSimpleEsp8266.h>

## #define DHTPIN D0

## #define DHTTYPE DHT11

## DHT dht(DHTPIN, DHTTYPE);

## char auth[] = "022dda3c213b40a99d16360ae9fc93aa";

## SoftwareSerial gsm(D2,D3);

## int sm=A0;

## const char\* ssid = "Narsimha";

## const char\* password = "Simhalp9";

## WiFiClient client;

## unsigned long myChannelNumber = 543197;

## const char \* myWriteAPIKey = "FCYOGGX6WKSZTOZN";

## uint8\_t temperature, humidity,soil,k=0, l=0,m=0;

## void setup()

## {

## Serial.begin(115200);

## dht.begin();

## delay(10);

## //code for connecting wifi

## Blynk.begin(auth, ssid, password);

## Serial.println();

## Serial.println();

## Serial.print("Connecting to ");

## Serial.println(ssid);

## WiFi.begin(ssid, password);

## while (WiFi.status() != WL\_CONNECTED)

## {

## delay(500);

## Serial.print(".");

## }

## Serial.println("");

## Serial.println("WiFi connected");

## Serial.println(WiFi.localIP());

## ThingSpeak.begin(client);

## pinMode(sm,INPUT);

## gsm.begin(9600);

## }

## void loop()

## {

## Blynk.run();//for controlling the relay module

## //code for measuring the volumetric content of soil moisture.

## int sensorvalue=analogRead(A0);

## sensorvalue=constrain(sensorvalue,485,1023);

## int soil=map(sensorvalue,485,1023,100,0);

## //code for measuring the temperature and humidity values

## temperature = dht.readTemperature();

## humidity = dht.readHumidity();

## Serial.print("Temperature Value is :");

## Serial.print(temperature);

## Serial.println("C");

## Serial.print("Humidity Value is :");

## Serial.print(humidity);

## Serial.println("%");

## //code for uploading the sensor values to the Thingspeak

## while(1)

## {

## if(temperature < 255)

## {

## k=temperature;

## ThingSpeak.writeField(myChannelNumber, 1, k, myWriteAPIKey);

## }

## if(humidity < 255)

## {

## l=humidity;

## 

## ThingSpeak.writeField(myChannelNumber, 2, l, myWriteAPIKey);

## }

## if(soil<101)

## {

## m=soil;

## ThingSpeak.writeField(myChannelNumber, 3, m, myWriteAPIKey);

## }

## }

## //code for sending the notifications to the user

## int count=1;

## if(soil<50)

## {

## while(count>=1&&count<=3)

## {

## if(count==1)

## {

## if(soil<=50)

## {

## gsm.println("ATCMGF=1");

## gsm.println("AT+CMGS=\"+918499008525\"\r"); //Mobile phone number to send message

## delay(1000);

## gsm.println ("your dearest plant moisture % is:");

## Serial.println ("your dearest plant moisture % is:");

## gsm.println(soil);

## gsm.println("%");

## delay(100);

## gsm.println("ATD8499008525;");

## delay(100);

## gsm.println("ATH");

## gsm.println((char)26);

## delay(10000);

## int sensorvalue=analogRead(A0);

## sensorvalue=constrain(sensorvalue,485,1023);

## int soil=map(sensorvalue,485,1023,100,0);

## if(soil>50)

## {

## gsm.println("ATCMGF=1");

## gsm.println("AT+CMGS=\"+918499008525\"\r"); //Mobile phone number to send message

## delay(1000);

## gsm.println (" your plat is safe");

## Serial.println ("your plant is safe:");

## delay(100);

## gsm.println("ATD8499008525;");

## delay(100);

## gsm.println("ATH");

## gsm.println((char)26);

## break;

## }

## }

## }

## else if(count==2)

## {

## if(soil<=50)

## {

## 

## gsm.println("ATCMGF=1");

## gsm.println("AT+CMGS=\"+918499008525\"\r"); //Mobile phone number to send message

## delay(1000);

## gsm.println ("your plant moisture % is:");

## Serial.println ("your plant moisture % is:");

## gsm.println(soil);

## gsm.println("%");

## delay(100);

## gsm.println("ATD8499008525;");

## delay(100);

## gsm.println("ATH");

## gsm.println((char)26);

## delay(10000);

## int sensorvalue=analogRead(A0);

## sensorvalue=constrain(sensorvalue,485,1023);

## int soil=map(sensorvalue,485,1023,100,0);

## if(soil>50)

## {

## gsm.println("ATCMGF=1");

## gsm.println("AT+CMGS=\"+918499008525\"\r"); //Mobile phone number to send message

## delay(1000);

## gsm.println (" your plant is safe");

## Serial.println ("your plant is safe”);

## delay(100);

## gsm.println("ATD8499008525;");

## delay(100);

## gsm.println("ATH");

## gsm.println((char)26);

## break;

## }

## }

## }

## else if(count==3)

## {

## if(soil<=50)

## {

## gsm.println("ATCMGF=1");

## gsm.println("AT+CMGS=\"+918499008525\"\r"); //Mobile phone number to send message

## delay(1000);

## gsm.println ("your plant moisture % is:");

## Serial.println ("your plant moisture % is:");

## gsm.println(soil);

## gsm.println("%");

## delay(100);

## gsm.println("ATD8309483023;");

## delay(100);

## gsm.println("ATH");

## gsm.println((char)26);

## delay(10000);

## int sensorvalue=analogRead(A0);

## sensorvalue=constrain(sensorvalue,485,1023);

## int soil=map(sensorvalue,485,1023,100,0);

## if(soil>50)

## {

## gsm.println("ATCMGF=1");

## gsm.println("AT+CMGS=\"+918499008525\"\r"); //Mobile phone number to send message

## delay(1000);

## gsm.println ("your plant is safe:");

## Serial.println ("your plant is safe”);

## delay(100);

## gsm.println("ATD8499008525;");

## delay(100);

## gsm.println("ATH");

## gsm.println((char)26);

## break;

## }

## }

## }

## count++;

## delay(10000);

## }

## }

## else

## {

## gsm.println("ATCMGF=1");

## gsm.println("AT+CMGS=\"+918499008525\"\r"); //Mobile phone number to send message

## delay(1000);

## gsm.println ("your plant is safe:");

## Serial.println ("your plant is safe:");

## gsm.println(soil);

## gsm.println("%");

## delay(100);

## gsm.println("ATD8499008525;");

## delay(100);

## gsm.println("ATH");

## gsm.println((char)26);

## }

## delay(10000);

## }

## 