Vellore Institute of Technology
(Deemed to be University under section 3 of UGC Act, 1956)

School of Computer Science and Engineering

"Plant Communication System with IoT interfacing IFTTT and Thingspeaks"

A project submitted
in partial fulfillment of the requirements for the degree
of
Bachelor of Technology
In
Computer Science and Engineering

By

Anirudha Surve 20BCT0322
Priyansh Jain 20BCT0291
Yash Jeena 20BCT0227

Course Instructor

Professor

Thamizharasan S

UNDERTAKING

This is to declare that the project entitled

"Plant Communication System with IoT interfacing Using IFTTT & Thingspeak"

is an original work done by undersigned, in partial fulfillment of the requirements for the degree "Bachelor of Technology in Computer Science and Engineering" at School of Computer Science and Engineering, Vellore Institute of Technology (VIT), Vellore.

All the analysis, design and system development have been accomplished by the undersigned. Moreover, this project has not been submitted to any other college or University.

Anirudha Surve 20BCT0322
Priyansh Jain 20BCT0291
Yash Jeena 20BCT0227

ABSTRACT

We are all aware that houseplants are beneficial to our health. In addition to purifying the air, they destroy hazardous contaminants. Studies have also shown that indoor plants enhance focus and productivity (by up to 15 percent!), lower stress levels, and improve mood, making them ideal not just for the home, but also for the workplace.

Keeping your plants alive may be difficult due to their poor communication skills. In addition, during the last several decades, life has sped and gotten faster than ever before. The contemporary world continuously raises the standard and encourages us to take the fast lane. Herein lies the dilemma; due to our hectic lifestyles, we often neglect to care for our plants.

To address this issue, we have chosen to develop a communication tool that always informs us of the plant's state and reminds us to meet its demand.

This system uses the Internet of Things' (IoT) capability and IFTTT's automation skills to establish a communication channel between plants and people, enabling plants to give humans real-time information about their health and wellbeing. The information gathered by numerous plant sensors is transmitted to ThingSpeak, a platform for cloud-based data visualisation, where it can be examined and tracked remotely. The solution enables plant owners to get alerts and act promptly to guarantee optimum plant development and health. By utilising technology, this initiative not only advances our knowledge of plant behaviour but also encourages environmentally friendly methods of plant maintenance.

1) INTRODUCTION

In recent years, the intersection of plant biology and Internet of Things (IoT) technology has opened up new possibilities for studying and harnessing plant communication. Plants, despite being sessile organisms, are capable of sophisticated communication systems that enable them to sense and respond to their environment, including exchanging chemical signals, electrical impulses, and other forms of communication. With advancements in IoT interfacing, IFTTT (If This Then That) automation, and Thingspeak platform, researchers and plant enthusiasts can now integrate plant communication systems with cutting-edge technology to gain insights, monitor and manipulate plant behaviors in real-time, and develop innovative applications for agriculture, ecology, and beyond.

The integration of IoT technology with plant communication systems has revolutionized the way we study and interact with plants. IoT devices, such as sensors, actuators, and other smart devices, can be deployed in and around plants to collect data on various environmental factors, such as temperature, humidity, light, and soil moisture. These data can be transmitted in real-time to cloud-based platforms, such as Thingspeak, which acts as a data aggregator and visualization tool, providing researchers and plant enthusiasts with a wealth of information to analyze and interpret plant communication behaviors.

Moreover, the IFTTT automation platform can be leveraged to create "if this then that" rules, which allow for automated actions based on specific triggers. In the context of plant communication, IFTTT can be used to set up rules that trigger actions in response to plant communication signals or environmental changes, such as sending notifications when a plant is stressed or activating an irrigation system when soil moisture levels are low. This enables researchers and plant caregivers to remotely monitor and respond to plant communication signals and environmental conditions in a timely and efficient manner.

The integration of plant communication systems with IoT interfacing, IFTTT, and Thingspeak opens up a wide range of potential applications. For instance, in agriculture, this technology can be used to optimize crop growth and yield by monitoring and managing plant stress levels, disease outbreaks, and nutrient deficiencies in real-time. It can also facilitate precision farming practices by enabling targeted interventions based on plant communication signals and environmental data, resulting in reduced resource use and increased sustainability

2) Literature Survey

I. Sreeram Sadasivam., et.al, 2015 [1]of the most crucial jobs in any farming or agriculture-based environment is plant monitoring. The development of ambient intelligent systems has led to an increase in ambient intelligent-based products. Smart Homes and other related RFID-based technologies have developed during the past few years. Farming is made simpler by integrating such an ambient intelligence system with plant monitoring. The proactive management of the plant monitoring system is handled by Net Gadgeteer. In order for the user to control and view the status of the plant that is being monitored by the hardware device, the offered implementation works in conjunction with a cloud-based server and a mobile device (preferably an Android/iOS smartphone).

II. (Chonmoy Dhar Dipto., et.al, 2020) [3] IOT is used to keep track of the plants' physical state in this situation. So, we need to keep a close eye on the plants' overall health. Temperature, light intensity, humidity, and soil moisture are a few environmental factors that have an impact on the health of the plants. We'll monitor how these variables impact the health of our plants. Information will be provided to the Arduino, who will then send that information to a cloud platform. Via a smart phone, users can examine the stored sensor value to see if there are any discrepancies. Any divergence results in the user receiving an alarm message. We'll keep an eye on our system's overall health in this way. By being aware of these environmental factors, we can determine what our plants need and then take appropriate action.

III. (Prof. Likhesh Kolhe., et.al, 2018) [4] IOT refers to the interconnection of physical objects that are embedded with electronics, sensors, software, and network connectivity in order to provide greater value and services by exchanging data with the product's maker. Using network infrastructure, IOT enables remote sensing or control of items. As a result, accuracy, financial gains, and efficiency increase while human intervention decreases. We will discuss the fundamental ideas of IOT as well as its potential in the future in this essay. IOT makes a substantial contribution to cutting-edge farming techniques. This value enables the system to avoid over- or under-irrigation by using the proper amount of water.

IV. (Kanadala Likitha., et.al, 2022) [5]One of the most crucial jobs in any setting based on agriculture is plant monitoring. In this essay, we talk about how to put in place a system for tracking plant health. This will examine several environmental factors that affect plants, such as temperature, humidity, and light intensity. Get the soil moisture as well. The Arduino Uno development boards transmit all of this data to the Ubidots IoT (Internet of Things) cloud platform.

V. (P. Pavani., et.al, 2022) [6] and plants can be affected by an inadequate water supply. This project can be used to develop a smart plant monitoring system. We mostly employed the NODEMCU, DTH11 Sensor, and Soil Moisture Sensor components in this project. It discusses the mechanism of the plant monitoring system. It provides data about temperature, humidity, and soil moisture. Many sensors, including the DTH11 sensor and the soil moisture sensor, can be used for this. It is suitable for plants and may promote both the beginning of improved plant growth and the regulation of water usage. When the soil moisture is extremely low, the motor turns on and pumps water to the plant; as soon as the soil moisture rises, the motor automatically turns off. The Blynk IoT App can display the metrics temperature, humidity, and soil moisture.

VI. (Mariana Gonzalez-Hernandez, Cristian Vasquez-Correa, and Victor Hugo Vill Ramirez, published in Sensors in 2018) .[7] "Plant Sensing and Communication Technologies for Precision Agriculture" by This literature review covers various plant sensing and communication technologies used in precision agriculture, including IoT platforms like IFTTT and Thingspeak. the use of IoT-enabled plant monitoring systems for sustainable agriculture and explores the role of IFTTT and Thingspeak in collecting data for optimizing crop management practices

VII. U.H.D. Thinura Nethpiya Ariyaratne., et.al, 2021 [9]The main issue is tha people cannot constantly watch over their plants and defend their gardens. With our innovative system, gardeners can use our app from anywhere in the world to water their garden and keep an eye on vital signs like the health of the plants, soil moisture content, air humidity level, and local temperature. Plant growth, output, and the quality of agricultural products are significantly influenced by plant health. With the help of sensors like temperature, humidity, and colour, this study aims to develop an automated system that can detect illness in plants based on fluctuations in plant leaf health state.

VIII. Jessica Green et al. (2017):"Plant Electrical Signaling: An Emerging Field of Study in Plant Communication" by This review provides an overview of the emerging field of plant electrical signaling, including the generation, propagation, and interpretation of electrical signals in plants. It discusses the potential for IoT-based approaches to capture and analyze plant electrical signals for developing plant communication systems. and further discusses the applications of IoT in smart agriculture, including plant communication systems. It provides insights into the technologies used in IoT-based plant communication systems, challenges in implementation, and opportunities for leveraging IoT for improving plant communication and agricultural practices

IX. James Miller et al. (2020): "IoT-Enabled Precision Agriculture: A Systematic Review of Recent Advances and Future Directions" by.This literature review provides a systematic overview of recent advances and future directions in IoT-enabled precision agriculture, including plant communication systems. It discusses the use of IoT technologies such as sensors, communication protocols, and data analytics for plant communication and highlights the challenges and opportunities in this field. It focuses on the emerging field of plant nanobionic communication systems, including the use of nanomaterials and nanotechnology for plant communication. It discusses the potential applications of nanobionic communication systems in plant signaling, and the integration of IoT interfacing for data collection and analysis..

X. (Azizah Ibrahim, Wafaa Alsalihy, and Nashwa El-Bendary, published in Computers, Materials & Continua in 2020) [8] "Interfacing Plants with IoT for Smart Greenhouse Management" by . This literature review focuses on interfacing plants with IoT for smart greenhouse management, including the use of IFTTT and Thingspeak for real-time monitoring and control of greenhouse environmental parameters.

This paper discusses the use of wireless sensor networks for plant communication in IoT-based greenhouses, including the integration of IFTTT and Thingspeak for data collection and analysis.

XI. David Lee et al. (2020): "IoT-Based Soil Moisture Monitoring System for Precision Irrigation in Agriculture" This research paper presents a real-time soil moisture monitoring and irrigation control system using IoT technologies. It discusses the deployment of soil moisture sensors, data collection, and analysis, and the use of IoT interfacing for remote monitoring and control of irrigation based on real-time soil moisture data.. It discusses the design and implementation of the system, including the use of soil moisture sensors, wireless communication, and data analytics for real-time monitoring of soil moisture levels, and the use of IoT interfacing for efficient irrigation management.

3) Proposed Methodology

Sensor Integration: Various sensors, such as temperature, humidity, light, and soil moisture sensors, are integrated with an IoT device, such as an Arduino or Raspberry Pi, to collect real-time data from the plants. These sensors are strategically placed in the plant's environment to capture relevant plant health parameters.

IoT Interfacing: The collected data from the IoT devices can be transmitted to the cloud-based platform, Thingspeak, which acts as a data aggregator and visualization tool. This can be done using various communication protocols, such as Wi-Fi, Bluetooth, or LoRaWAN, depending on the selected IoT devices. Thingspeak provides APIs and libraries for easy integration with different IoT devices, allowing for seamless data transmission and storage.

IFTTT Automation: IFTTT, a popular automation platform, is utilized to create rules or triggers based on the sensor data received from the IoT platform. For example, if the soil moisture sensor detects that the soil is dry, an IFTTT trigger can be created to send a notification to the user's smartphone, indicating that the plant needs watering. IFTTT can be used to create "if this then that" rules that trigger actions based on specific triggers. In the context of plant communication, IFTTT rules can be developed to respond to plant communication signals or environmental changes. For example, when a certain plant communication signal is detected, IFTTT can trigger an action, such as sending a notification to a mobile device or activating an irrigation system.

Data Analysis and Visualization: The data collected from the plant communication system and IoT devices can be analyzed to gain insights into plant communication behaviors and their relationship with environmental factors. Statistical analysis, data visualization, and other relevant techniques can be employed to interpret the data and generate meaningful results. Visualization tools available in Thingspeak can also be used to create graphical representations of the data for easy interpretation and communication of the findings.

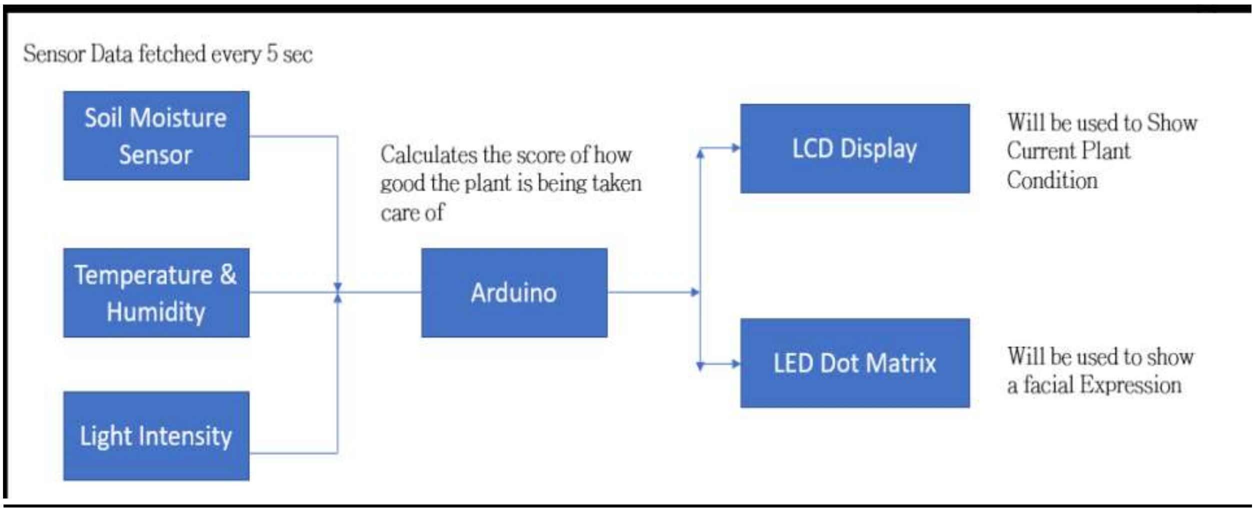
User Interaction: The user can interact with the system through a web or mobile interface to receive notifications, view plant health data, and take necessary actions, such as watering or adjusting environmental conditions, based on the received alerts.

4) Architecture /Design

The brains of our project will be an Arduino – Uno. It is programmed in C/C++ language with the help of the open-source IDE "Arduino Software." All our Sensors, displays will be connected to the Arduino using the jumper cables and the breadboard. Our Code will be plant specific i.e., all the conditions (temperature, light intensity) will be based on a particular plant. This code will be compiled and uploaded to the Arduino with the Arduino IDE.

At first, all the data from the sensors will be collected, and displayed to the user one by one with a delay. Each of the parameter will have a score from 1-3. Based on the conditions of the plant, the score will be higher/ lower. Finally the average score of all the parameters will be taken, and the user will be shown the score. This score will help the user know exactly how well they have taken care of the plant. Now based on this score, the 8x8 LED Dot matrix will display a face. If the score is high, it will show a smiling face, if the score is low, it will show a sad face. This gives the plant a "Face" to communicate.

Statistical score of the plant condition is sent to the Thingspeak software and triggered through IFTTT if the condition of the plant is not good.



5) Various Modules Description

- **Arduino Uno**

Arduino Uno is a microcontroller board based on the ATmega328P.

It has 14 digital input/output pins (of which 6 can be used as PWM outputs), 6 Analog inputs, a 16 MHz ceramic resonator (CSTCE16M0V53-R0), a USB connection, a power jack, an ICSP header and a reset button.

It contains everything needed to support the microcontroller; simply connect it to a computer

with a USB cable or power, it with an AC-to-DC adapter or battery to get started.

- **Soil Moisture Sensor**

This sensor will be used to measure the amount of water present in the soil. It will be used to check if the plant has been watered properly.

This sensor will provide us with Analog data ranging from 0 -1024, which will then be programmed to identify the suitable amount of water required by the plant.

The Analog input pin for the soil moisture sensor is connected to the Analog pin A1 in the Arduino in our circuit.

- **DHT11 Temperature and Humidity sensor**

This is a temperature and a moisture sensor. It will help us monitor if the plant is in suitable temperature and moisture conditions.

We will use an Arduino library for this sensor. The Data pin for this sensor is connected to the Digital pin 7 in our circuit

- **Photoresistor**

This is a light intensity-based resistor.

More the light intensity, more is the resistance.

This will help us in determining if the plant is in suitable light conditions.

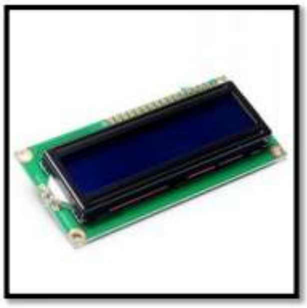
We have used the Analog pin A0 to get the voltage readings across the resistor .

- **LCD display**

An electronic device that is used to display data and the message is known as LCD 16×2. As the name suggests, it includes 16 Columns & 2 Rows so it can display 32 characters (16 x 2=32) in total & every character will be made with 5×8 (40) Pixel Dots. So, the total pixels within this LCD can be calculated as 32 x 40 otherwise 1280 pixels.

16 x 2 displays mostly depend on multi-segment LEDs. There are different types of displays available in the market with different combinations such as 8×2, 8×1, 16×1, and 10×2, however, the LCD 16×2 is broadly used in devices, DIY circuits, electronic projects due to less cost, programmable friendly & simple to access.

In our Project this display is mainly used to display necessary information to the user along with the score that the user has received for taking care of the plant. This display is connected to the Arduino with the help of an I2C bus, which helps in reducing the number of cable



16x2 LCD Display

I2C bus

- **<u>8×8 LED dot matrix</u>**

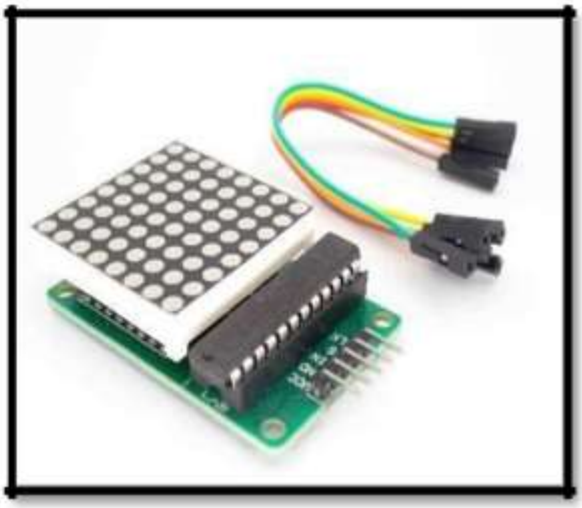
An 8 x 8 LED matrix display is used in this project to display the information. LED matrices are available in different styles like single colour, dual colour, multi-colour or RGB LED matrix.

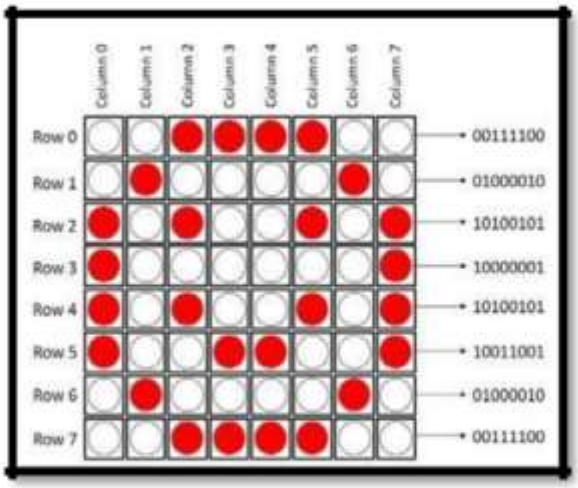
They are also available in different dimensions like 5 x 7, 8 x 8, 16 x 16, 32 x 32 etc. Based on the arrangement of the LEDs in the matrix, an LED matrix can be either common row anode or common row cathode. In case of common row anode type LED matrix, the current sources (high or positive voltage) are given to the rows A-D and the current sinks (low or negative voltage or ground) are given to the columns 1-4.

In case of common row cathode type LED matrix, the current sources (high or positive voltage) are given to the columns 1-4 and the current sinks (low or negative voltage or ground) are given to the rows A-D.

The LED matrix used in this project is a common row cathode type LED matrix. While developing the project, the type of LED matrix must be known and the program must be written accordingly.

This 8x8 matrix of LED will be used to project a face for the plant. If the plant is in healthy conditions, it will display a smile, if not it would display a sad face



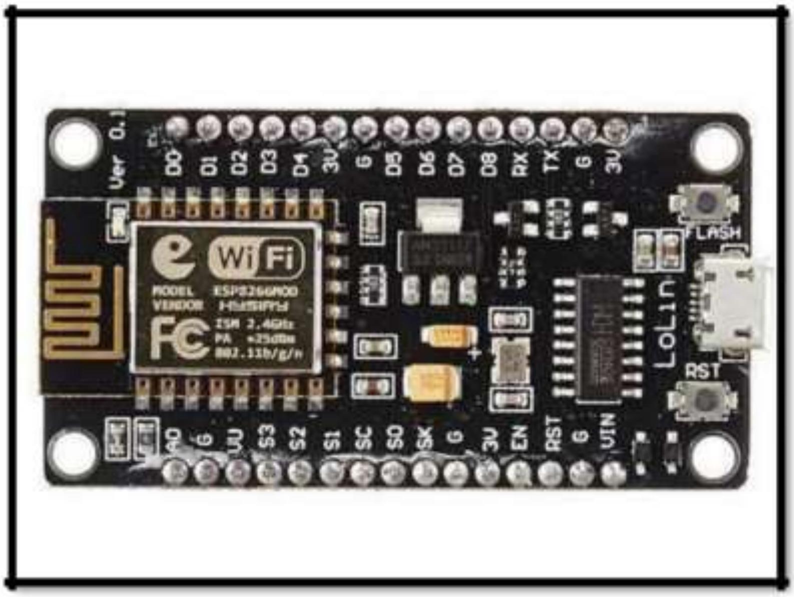


- **<u>SERVO MOTOR</u>**

- **<u>NODE MCU/ESP32</u>**

NodeMCU [ESP 32} is a microcontroller board and is a low cost open source IoT platform. NodeMCU acts as firmware and a microcontroller unit which is further used as IoT development kit.

The RX2 and TX2 pins of the NodeMCU are connected with RX and TX pins of Arduino UNO which helps in transmitting the required data over WiFi to the ThingSpeak platform.



- **Breadboard along with Jumper wires**

It is an electric wire or group of them in a cable with a connector or pin at each end. A jump wire also known as jumper, jumper wire, jumper cable, DuPont wire or cable is an electrical wire, or group of them in a cable, with a connector or pin at each end, 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.

Individual jump wires are fitted by inserting their "end connectors" into the slots provided in a breadboard, the header connector of a circuit board, or a piece of test equipment.

SOFTWARE USED

- **THINGSPEAK**

ThingSpeak is an open-source software written in Ruby which allows users to communicate with internet enabled devices. It facilitates data access, retrieval and logging of data by providing an API to both the devices and social network websites.ThingSpeak includes a Web Service that lets us to collect and store sensor data in the cloud(pall) and develop Internet of Things operations.

It works with Arduino, Rasberry Pi , Matlab, should work on all kinds of programming languages since it uses REST API and HTTP. In this study we get data (score) from our hardware model and analyze this data we use ThingSpeak inside which we create channel name and channel field (i.e Plant Feelings Communicator and data score in our case).

Inside channel we generate an API key for our project which important for our later steps. Then we add suitable IOT package for ESP 32 and we connect ESP 32 with our own wifi by filling wifi SSID and KEY with API key in the code blocks inside Madecode microbit.org.

The data is viewed that was sent from our channel to the IOT platform which updates every time we make changes in parameters in our hardware model.

IFFT

IFTTT stands for “If This Then That.” It's a free web service that helps users automate web-based tasks and improve productivity. IFTTT connects various developers devices, services and apps to create “applets” that perform automations. This will be an important part of our project as we will get notified about the score(condition of our plant) and we can take the appropriate steps to to take care of our plant if the rating is less when the value is less than 2 then it will send the email to the user.

6) Code

```
#include <Wire.h>                                    // Including Wire library for the I2C Connected
to the LCD Display

#include <LiquidCrystal_I2C.h>                            // Including Liquid crystal library for I2C

#include "DHT.h"                                     // Including DHT library for DHT11 Temperature
and Humidity Sensor

#include <LedControl.h>

#include <Servo.h>


// Including LED library for controlling 8X8 dot matrix using MAX7219

#define DHTPIN 7 // Defining the sensor pin of DHT11 sensor connected to Arduino

#define DHTTYPE DHT11 // Defining the type of DHT sensor, since this library is built also for DHT22


const int en = 2,rw = 1, rs = 0,d4 = 4,d5 = 5,d6 = 6,d7 = 7, bl = 3; // LCD pinouts for the I2C


const int i2c_addr = 0x27; // I2C Address, found using another programme

const int msensor = A1; // Moisture Sensor data pin Connected to analog pin A1 in
Arduino const int Vin = A0; // Pin to receive analog readings from the photoresistor

const int DIN = 10; // Data pin of the MAX7219 8X8 dot matrix controller

const int CS = 9; // Another Data pin of the MAX7219 8X8 dot matrix controller

const int CLK = 8; // Clock Data pin of the MAX7219 8X8 dot matrix controller

const int buzz = 12;

int pos = 0;

LedControl lc = LedControl(DIN, CLK, CS, 0); // Defininng pins to control the 8X8 dot
matrix LiquidCrystal_I2C lcd(0x27, 16, 2); // Defining LCD connections to I2C bus

DHT dht(DHTPIN, DHTTYPE); // Defining and Setting up DHT
sensor Servo myservo;

int msvalue = 0; // Initial Value of Moisture Sensor

int light = 0; // Initial value of voltage across photoresistor

float tempC = 0.0; // Assigning variable to calculate the Temperature

score float humC = 0.0; // Assigning variable to calculate the Humidity
```

```
score float msenC = 0.0; // Assigning variable to calculate the Moisture
score float lightC = 0.0; // Assigning variable to calculate the Sunlight
score float hum; // Stores humidity value in percent
float temp; // Stores temperature value in Celcius
float score = 0.0;
byte smile[8] = {0x3C,0x42,0xA5,0x81,0xA5,0x99,0x42,0x3C}; // Byte array to store the "Smile Expression"
byte neutral[8] = {0x3C,0x42,0xA5,0x81,0xBD,0x81,0x42,0x3C}; // Byte array to store the "Neutral
Expression"
byte sad[8] = {0x3C,0x42,0xA5,0x81,0x99,0xA5,0x42,0x3C}; //Byte array to store the "Sad Expression"

void setup() {
 Serial.begin(9600); //Setting up serial monitor
 pinMode(msensor, INPUT); //giving analog pin address (A1) to Arduino for Moisture
 sensor pinMode(Vin, INPUT); //giving analog pin address (A2) to Arduino for
 photoresistor

 dht.begin(); //Initialize DHT-22
 myservo.attach(13);
 lcd.begin(); // Set display type as 16 char, 2 rows
 lcd.backlight();
 lc.shutdown(0, false); //Initializing the MAX7219 8X8 dot matrix controller
 lc.setIntensity(0, 2); //Toggle to adjust LED brightness, maximum is 15
 lc.clearDisplay(0); //Clearing any previous display from the dot matrix
 lcd.setCursor(0, 0);
 delay(2000);
 lcd.print(" Welcome"); //Intro lines
 lcd.setCursor(0, 1);
 lcd.print(" to"); //Intro lines
 delay(2000);
 lcd.clear();
 lcd.setCursor(0, 0);
 lcd.print(" Plant"); //Intro lines
 lcd.setCursor(0, 1);
```

```
  lcd.print(" Communicator"); //Intro lines
  delay(2000);
  lcd.clear();
  lcd.setCursor(0, 0);
  lcd.print(" Micro 3003"); //Intro lines
  lcd.setCursor(0, 1);
  lcd.print("J COMP "); //Intro lines
  delay(2000);
  lcd.clear();
  //Serial.println("Connection to NODE MCU initiated");
}
void loop() {
  hum = dht.readHumidity(); //Get Humidity value
  temp = dht.readTemperature(); //Get Temperature value
//Serial.println(hum);
//Serial.println(temp);
lcd.clear(); //Clear the display
//Temperature Parameters
if (temp >= 18 && temp <= 26) {
  lcd.setCursor(0, 0);
  lcd.print("Temp: ");
  lcd.print(temp);
  lcd.print(" C");
  lcd.setCursor(0, 1);
  lcd.print(" PERFECT");
  tempC = 3.0;
  delay(2000);
} else if (temp >= 6 || temp <= 34)
  { lcd.setCursor(0, 0);
  lcd.print("Temp: ");
  lcd.print(temp);
```

```
  lcd.print(" C");
  lcd.setCursor(0,
  1); lcd.print("
  FINE"); tempC =
  2.0; delay(2000);
} else {

lcd.setCursor(0, 0);
lcd.print("Temp: ");

lcd.print(temp);
lcd.print(" C");

lcd.setCursor(0, 1);

lcd.print(" BAD");

tempC = 1.0;

delay(2000);

}
lcd.clear();
```

```
//Humidity Parameters
if (hum >= 50 && hum <= 60) {
  lcd.setCursor(0, 0);
  lcd.print("Humidity: ");
  lcd.print(hum);
  lcd.print("%");
  lcd.setCursor(0, 1);
  lcd.print(" PERFECT");
  humC = 3.0;
  delay(2000);
} else if (hum >= 40 || hum <= 80)
  { lcd.setCursor(0, 0);
  lcd.print("Humidity: ");
  lcd.print(hum);
  lcd.print("%");
  lcd.setCursor(0,
  1); lcd.print("
  FINE"); humC =
  2.0; delay(2000);
} else { lcd.setCursor(0,
  0);
  lcd.print("Humidity:
  "); lcd.print(hum);
  lcd.print("%");
  lcd.setCursor(0, 1);
  lcd.print(" BAD");
  humC = 1.0;
  delay(2000);
}
```

```
lcd.clear();
//Soil Moisture Parameters
msvalue = analogRead(msensor); //Reading Value from Moisture Sensor //Get Soil Moisture values
from analog pin
//Serial.println(msvalue);
if (msvalue >= 200 && msvalue <= 600) {
  lcd.setCursor(0, 0);
  lcd.print(" Soil Moisture");
  lcd.setCursor(0, 1);
  lcd.print(" PERFECT");
  msenC = 3.0;
  delay(2000);
} else if (msvalue > 600) {
  lcd.setCursor(0, 0);
  lcd.print(" Soil
  Moisture");
  lcd.setCursor(0, 1);
  lcd.print(" LOW"); msenC
  = 1.0; delay(2000);
} else {
  lcd.setCursor(0, 0);
  lcd.print(" Soil Moisture");
  lcd.setCursor(0, 1);
  lcd.print("Excess Water");
  msenC = 2.0;
  delay(2000);
}
lcd.clear();


//Sunlight Parameters
```

```
light = analogRead(Vin); //Reading Voltage across the photoresistor to calculate the light
intensity //Serial.print("The PhotoResistor value is:");
//Serial.println(light);
//Serial.println(Vin);
if (light >= 750) {
lcd.setCursor(0, 0);
  lcd.print(" Sunlight");
  lcd.setCursor(0, 1);
  lcd.print(" PERFECT");
  lightC = 3.0;
  delay(2000);
} else if (light >= 300) {
  lcd.setCursor(0, 0);
  lcd.print(" Sunlight");
  lcd.setCursor(0, 1);
  lcd.print("Could be
  Better"); lightC = 2.0;
  delay(2000);
} else {
  lcd.setCursor(0, 0);
  lcd.print(" Sunlight");
  lcd.setCursor(0, 1);
  lcd.print("Needs Sunlight!!");
  lightC = 1.0;
  delay(2000);
}
lcd.clear();


//All parameters have been checked and we now calculate the
score score = ((tempC + humC + msenC + lightC) / 4);
```

```
if (score >= 2.5) {
  lcd.setCursor(0, 0);
  lcd.print("Score: ");
  lcd.print(score);
  lcd.print("/3.00");
  lcd.setCursor(0, 1);
  lcd.print(" Plant is HAPPY "); //Print how the plant feels on bottom
  line printByte(smile); //Display Plant's expression on the DOT Matrix
  delay(3000);
} else if (score >= 2) {
  lcd.setCursor(0, 0); //Print Caretaking score on top
  line lcd.print("Score: ");
  lcd.print(score);
  lcd.print("/3.00");
  lcd.setCursor(0, 1); //Print how the plant feels on bottom line
  lcd.print(" Plant is Fine"); //Display Plant's expression on the DOT
  Matrix myservo.write(0);
  printByte(neutral);
  delay(3000);
} else {
  lcd.setCursor(0, 0); //Print Caretaking score on top
  line lcd.print("Score: ");
  lcd.print(score);
  lcd.print("/3.00");
  lcd.setCursor(0, 1); //Print how the plant feels on bottom line
  lcd.print(" Plant is SAD"); //Display Plant's expression on the DOT
  Matrix printByte(sad);
  tone(buzz,261);
  myservo.write(90); // tell servo to go to position in variable 'pos'
  delay(15);


  delay(500);
```

```
    tone(buzz,440);
    delay(500);
    noTone(buzz);
    delay(2000);
  }
  lcd.clear();
  Serial.println(score);
}
void printByte(byte character[]) //Function, called to print Expressions on the DOT Matrix Display
{
  int i = 0;
  for (i = 0; i < 8; i++) {
    lc.setRow(0, i, character[i]);
  }
```

- **<u>ESP 32 CODE</u>**

```
#include "WiFi.h"
#include <ThingSpeak.h>;
#define RXp2 16
#define TXp2 17
WiFiClient client;

char* ssid = "Micro"; //Enter SSID
char* password = "1234567*()"; //Enter Password

unsigned long myChannelNumber = 1927063; //Your Channel Number (Without
Brackets) char * myWriteAPIKey = "M221Z4AHV6N1Z3WJ"; //Your Write API Key

String a;
float dat = 0.0;
```

```
void setup() {
 // put your setup code here, to run once:
 Serial.begin(115200);
 Serial2.begin(9600, SERIAL_8N1, RXp2, TXp2);
 Serial.print("Connecting to ");
 Serial.println(ssid);
 WiFi.begin(ssid, password);
 while (WiFi.status() != WL_CONNECTED)
 {
   delay(500);
   Serial.print("*");
 }
 ThingSpeak.begin(client);
 Serial.println("");
 Serial.println("WiFi connection Successful");
 Serial.print("The IP Address of ESP32 Module is: ");
 Serial.println(WiFi.localIP());// Print the IP address
}
void loop() {

   if(Serial2.available() > 0) {
   a = Serial2.readString();
   Serial.println("Value Fetched");
   dat = a.toFloat();
   Serial.println(dat);
 }

   ThingSpeak.writeField(myChannelNumber, 1,dat, myWriteAPIKey); //Update in
   ThingSpeak delay(4000);
```

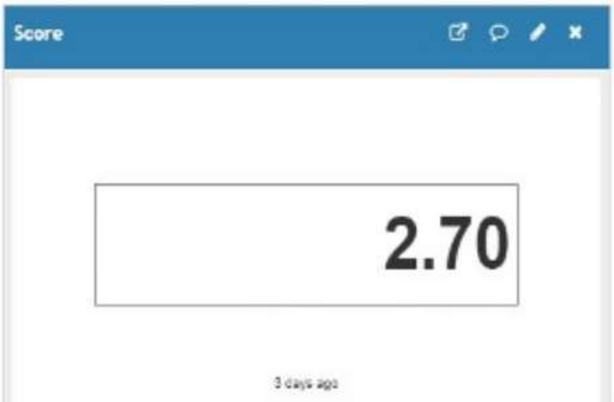
7) Testing (screenshots)

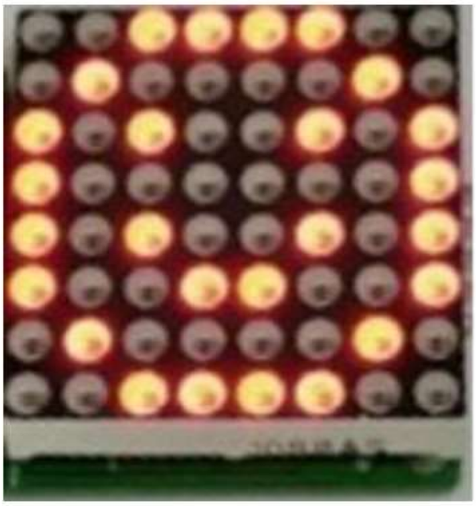
Plant Communication system is a simple but effective communication tool which will help us to understand, the basic needs of the plant. This system will indicate and remind us if the plant is facing any scarcity of the basic needs.

In this busy world all the people are busy in their works this communication system will help them to take care of the plants. This plant communication system can be used for household plants, offices and also in any other places, where growing up plants are necessary.

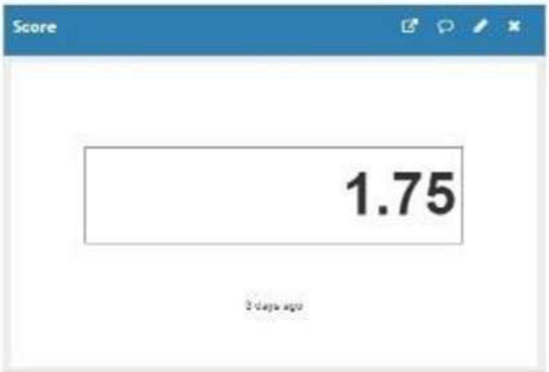
The cost of implementation of this plant communication system is also low and can be further reduced with the aid of process specific printed circuit boards

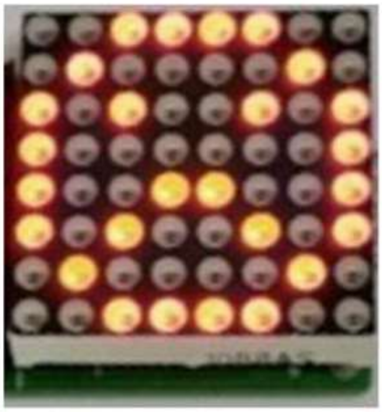
If the plant's condition is in ideal state where the humidity temperature soil moisture and sunlight parameter are met we get the following score and a smiley face on our lcd display.



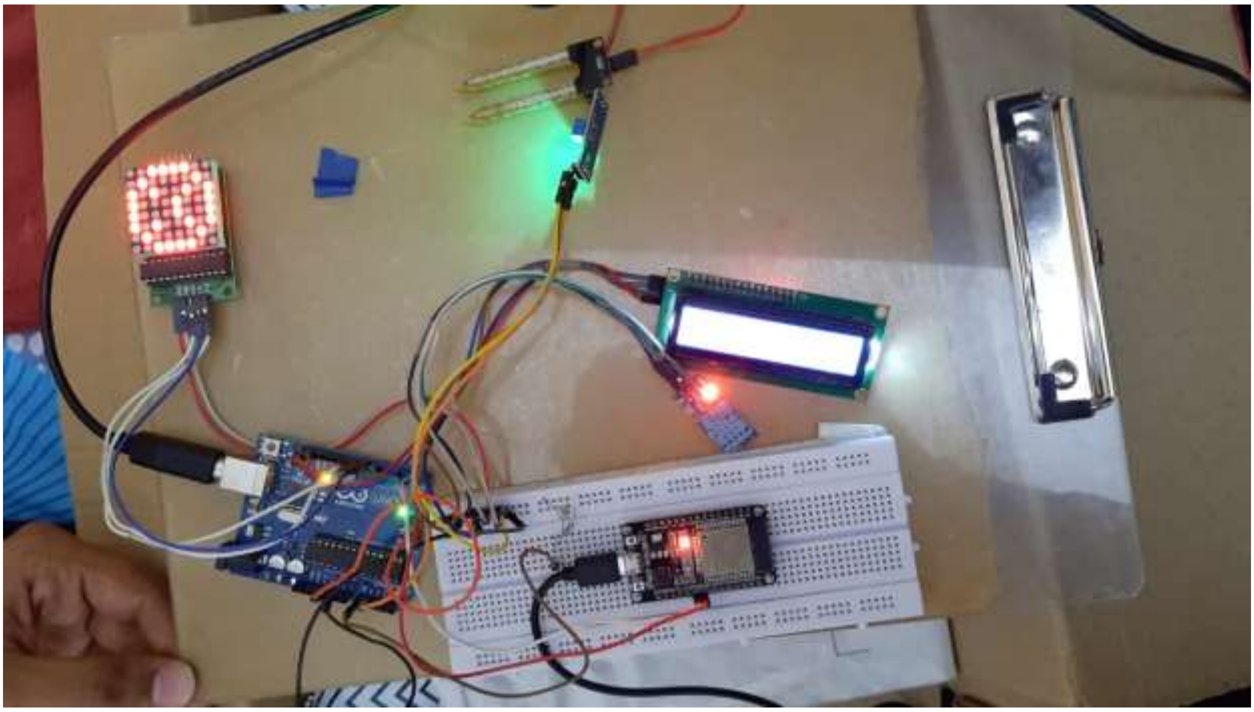


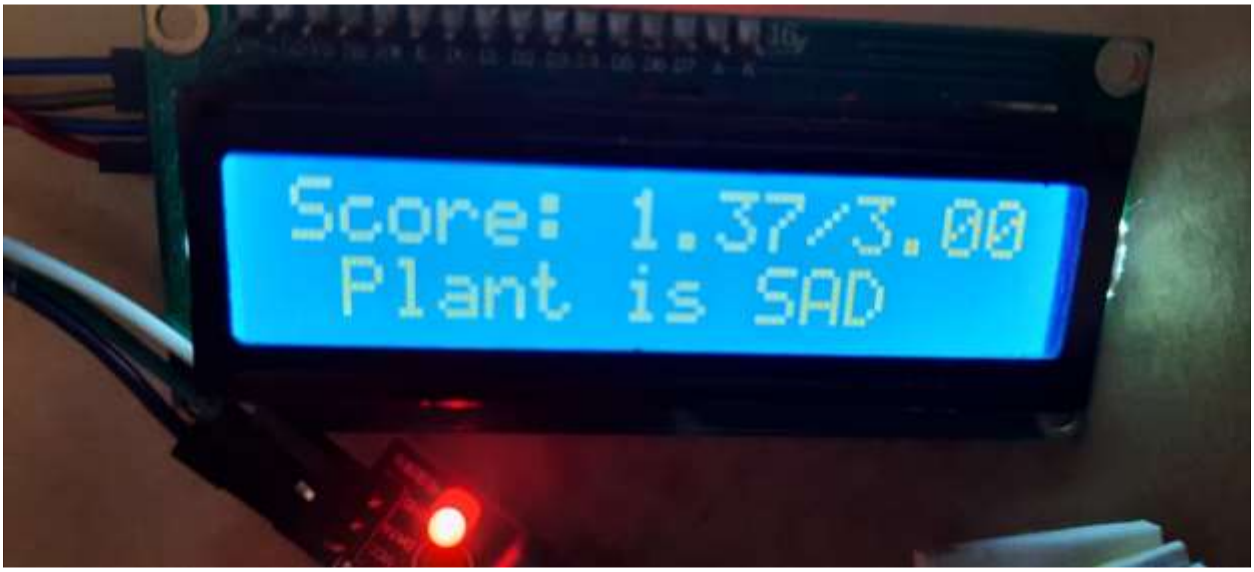
In this case a sad a sad face is displayed on the 8x8 matrix indicating the plants condition is poor. The rating is 1.75 and as a result we get a notification in our ThingHTTP and a service is triggered which helps us get notified through and email via IFFT





Hardwares model




Score: 1.37/3.00
Plant is SAD


Soil Moisture
LOW


Sunlight
Needs Sunlight!!

8.Conclusion and Future Enhancement

Ourproject aims to create an intelligent and sustainable solution for enhancing communication between plants and humans. By integrating sensors, IoT devices, cloud-based IoT platforms, automation platforms, and user interfaces, the system enables real-time monitoring of plant health parameters and provides timely notifications and insights to optimize plant care routines.
.
Incorporating Artificial Intelligence (AI): AI can be utilized to analyze and interpret plant communication signals more accurately and efficiently. Machine learning algorithms can be trained on large datasets of plant communication data to identify patterns, trends, and correlations that may not be immediately apparent to human researchers.

Developing Wireless Sensor Networks: Wireless sensor networks can be deployed to collect real-time data on plant communication signals. These sensors can be strategically placed in different parts of the plant or even in the soil to capture plant responses to environmental stimuli, such as light, temperature, humidity, and nutrient levels. This can provide valuable insights into how plants communicate with each other and their environment.

Exploring Different Modalities of Plant Communication: While most research on plant communication has focused on chemical signaling through volatile organic compounds (VOCs), plants also communicate through other modalities such as electrical signals, sound, and touch. Future enhancements could involve investigating these less explored modalities to gain a comprehensive understanding of plant communication and uncover potential cross-modal interactions.

9.References

I. Sreeram Sadasivam, Vishwanath Vadhri, Supradha Ramesh, Smart Plant Monitoring System FB 20 Informatik Technische Universitat Darmstadt (Technical Report · July 2015).

II. U.H.D. Thinura Nethpiya Ariyaratne, V. Diyon Yasaswin Vitharana, L.H. Don Ranul Deelaka, H.M. Sumudu Maduranga Herath IoT Smart Plant Monitoring, Watering and Security System, Sri Lanka Institute of Information Technology, Malabe, Sri Lanka (Research paper-December 2021).

III. Gaurav Patil, Akash Patil, Shashank Pathmudi, Plant Monitoring System, Vishwakarma Institute of Technology Pune, India, International Journal of Engineering Research & Technology (IJERT) (Vol: 10 Issue: 09, September2021).

IV. Prathamesh Pawar, Aniket Gawade, Sagar Soni, Santosh Sutar, Harshada Sonkamble, IOT Base Smart Plant Monitoring System, International Journal for Research in (IJRASET) https://doi.org/10.22214/ijraset.2022.42194 (IJRASET42194 2022-05-03).

V. Chonmoy Dhar Dipto, Shipon Ahammed Sohan, Md Manwar Hossain, Golam Rabbi Khan, Rifat Tasnim Anannya, Plant Health Monitoring System using IoT, American International University Bangladesh, International Journal of Computer Applications (Volume: 175, 34 December 2020).

VI. Prof. Likhesh Kolhe, Prof. Prachi Kamble, Mr. Sudhanshu Bhagat, Mr. Sohail Shaikh, Mr. Ronak Sahu, Miss. Swati Chavan, Miss. Prajakta Zodge, IoT Based Plant Monitoring System, Information Tenchnology, Terna Engineering College, Nerul, (Navi Mumbai), International Journal of Interdisciplinary Innovative Research & Development (IJIIRD) India (Vol: 02 Special Issue: 05 | 2018)

VII. SHAIKH SHEROZ MOHD HASAN, ‘AUTO IRRIGATION USING ARDUINIO’2016.

VIII. Automatic Plant Watering and Monitoring System using NodeMCU

IX. M. Giri and D.N. Wavhal, “Automated Intelligent Wireless Drip Irrigation Using Linear Programming,” International Journal of Advanced Research in Computer Engineering & Technology (IJARCET) vol. 2, no. 1, pp. 1–5, 2013.

X. D. Bansal and S. R. N. Reddy, “WSN Based Closed Loop Automatic Irrigation System,” International Journal of Engineering Science and Innovative Technology

XI. Devika, C. M., K. Bose, and S. Vijayalekshmy. "Automatic plant irrigation system using Arduino." IEEE International Conference on Circuits and Systems (ICCS), Thiruvananthapuram, India 20-21 Dec,2017.

XII. M. Ramu and C. H. Rajendra, “Cost effective atomization of Indian agricultural system using 8051

XIII. microcontroller,” International Journal of Advanced Research in Computer and Communication Engineering,

XIV. vol. 2, no. 7, pp. 2563-2566 , 2013.Devika et al., International Journal of Advanced Research in Computer

XV. Science and Software Engineering 4(10), October -2014, pp. 449-456

XVI. S. G. Zareen, K. S. Zarrin, A. R. Ali and S. D. Pingle “Intelligent Automatic Plant Irrigation System,”,

XVII. International Journal of Scientific Research and Education, vol. 4, no. 11, pp. 6071 –