Project Report

on

AgroPredict

by

Naman Sharma (AU19B1002)

Under the guidance of

Ravikumar Jadhav

School of Computer Engineering

Avantika University, Ujjain

2020-2021

Contents

Abstract

Nowadays Farmers are facing problems due to less proper information and better practice methods for farming and also no proper system for planning their crops to be planted according to their particular seasons.

The problem which has been faced by the farmers in the current scenario is that the crops which they plant are been not so planned as the decision of been planting a crop is been taken on the basis of last year basis or been recommended by any person's past experiences which is not a calculated way to take a big decision like this.

AgroPredict is the solution to these major problems as it helps in taking the decision by predicting the right crops after analyzing the real-life condition of the soil and the weather forecast predictions to deliver the farmer more accurate prediction which helps the farmer to take decision more wisely.

Chapter 1
Introduction

1.1 Motivation

The motivation for doing this project was primarily an interest in undertaking a challenging project in an interesting area of research. And also, the domain we had worked has still some problems which are still unsolved. So, coming up with the solution for it is somewhat self-motivating. And the opportunity to learn about a new area of IOT applications and integrate it with a real-world problem and also giving it a digital solution to it was appealing.

1.2 Problem Statement

Designing an IOT system for famers where crops will be predicted based on the real-time condition of the field through which production can be increased and profit could be maximized.

1.3 Aim

Our aim was to create an IOT application for providing the predicted results to farmers who are not able to take decision of which plant will be best suitable for their soil.

1.4 Objectives

The main objectives for designing this project are given below-

1. To collect data from the sensors for certain parameters like temperature level, water level and soil alkalinity.
2. To include weather forecast and rainfall data.
3. To collect various feeds from user.
4. To analyze the condition of field to increase the production.

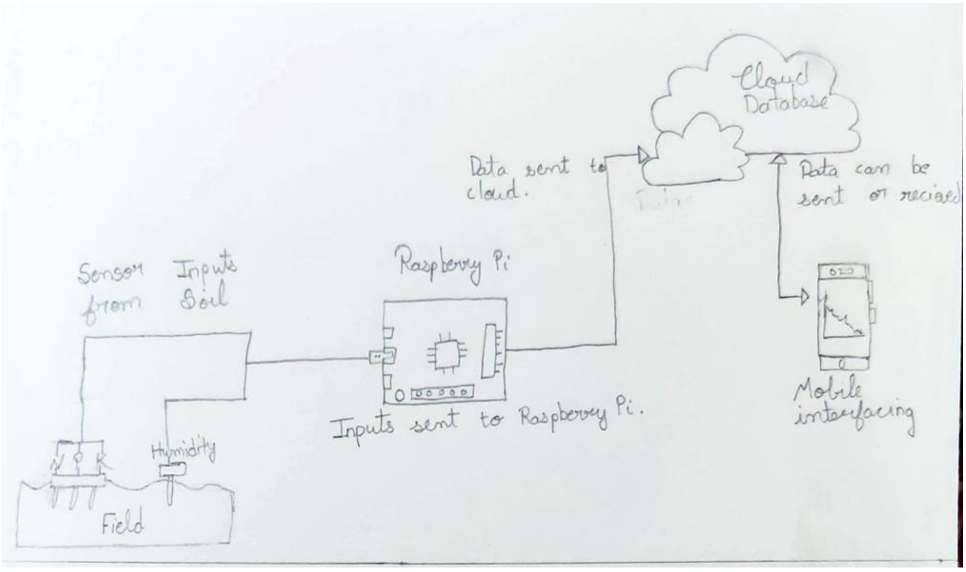
Chapter 2
Ideation

2.1 Proposed Solutions

2.1.1 Concept 1
In this concept there are certain parameters ideated for this concept generation that are briefly pointed below-

- Determining the real-time condition of the field through sensors.
- Taking field parameters from the farmer.
- Prediction of the crop based on the pre-filled inputs.

Limitations of this concept is the data used to analyze and predict is very static so that result will face challenges on more real-life scenarios.
Advantage of this concept is that it is cost effective as well faster in prediction for the crops.

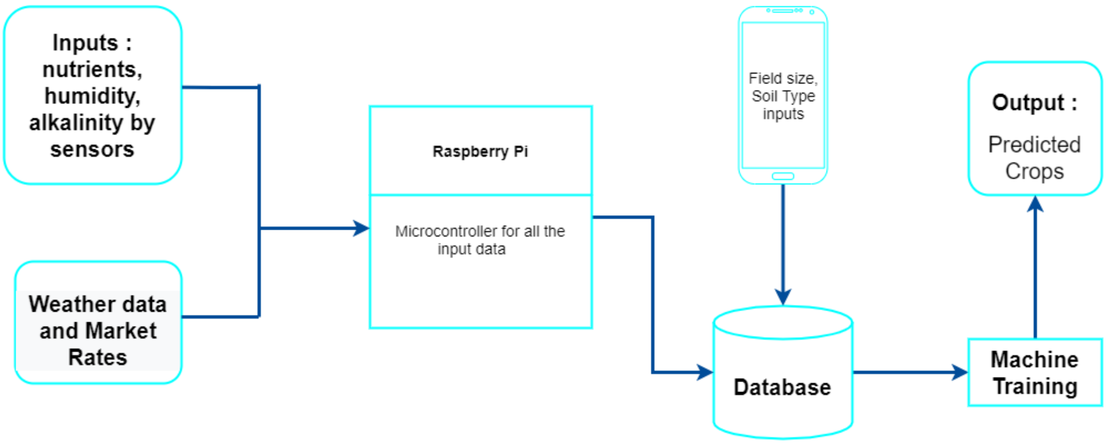


2.1.2 Concept 2

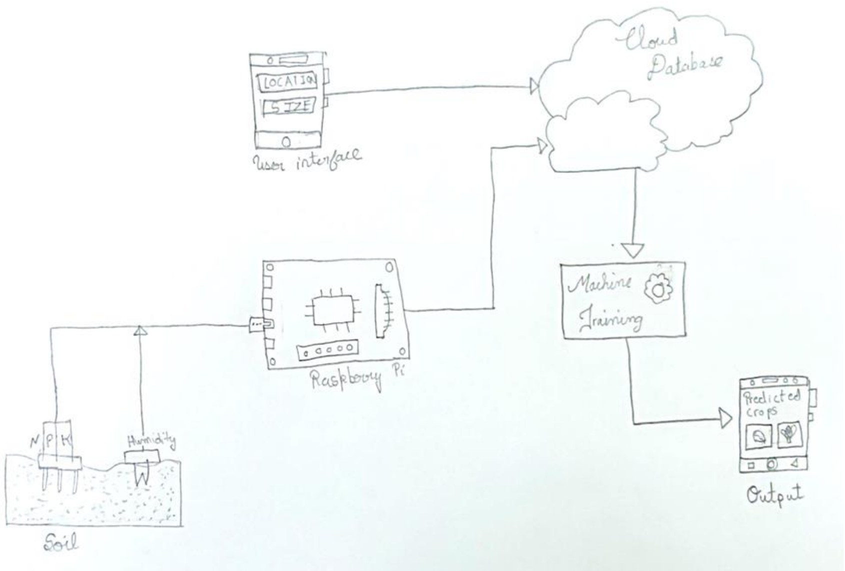
In this concept there are certain parameters ideated for this concept generation that are briefly pointed below-

- Nutrition-level, Humidity-level and soil alkalinity all the major three inputs will be analyzed taken from the sensors.
- Area of field, type of soil, and season of crop will be taken from the farmer through mobile interfacing.
- Weather forecast data and market rates will be trained to the system for more strong predictions.

- Cost of planting and caring will be calculated based on the size of field and type of crops.
- Based on several inputs from the sensors and cloud data prediction of the crop will be made of very much accuracy.



Limitation from this concept can be that natural conditions are not so fixed so if weather changes in real time scenarios so output will also face challenges.
Advantage of this system is that this will be give the most practical prediction because all of the data taken is on real-time conditions and also this system will be trained for better outputs.



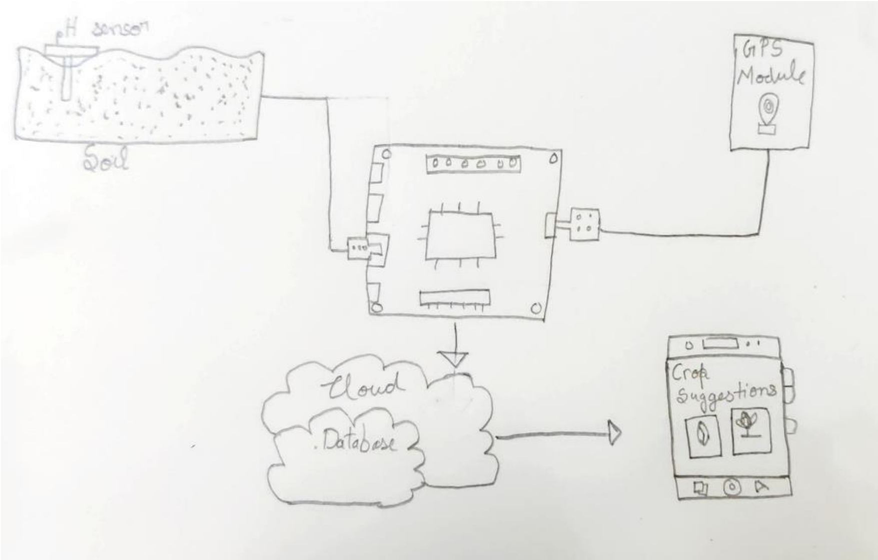
2.1.3 Concept 3

In this concept there are certain parameters ideated for this concept generation that are briefly pointed below-

- pH level of soil will be taken as an input from the sensors.

- Field soil type will be taken through GPS location of the system.
- Size of field will be taken through farmer
- Based on alkalinity and type of the soil crops will be suggested to the farmer for maximizing their profit.

Advantage of this concept is it will be cheap in cost of making but it has disadvantage of giving static output to the user.



2.2 Concept Finalization

We had finalized **Concept 2** as our final concept because of the following reasons-

Comparing the all three concepts we found that concept 2 will be more precise for crop predictions because it is using real time data as well as weather forecast and last season's market rates. Concept 3 is only working on single pH sensor which is not sufficient for better outputs and Concept 2 is working on static data so it is less implacable.

Chapter 3

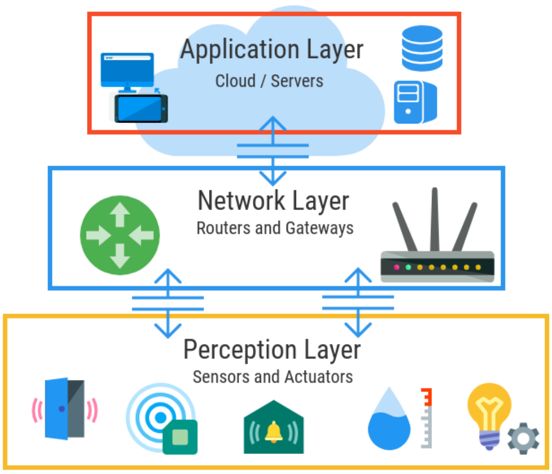
Design Methodology

1.1 IOT Architecture

IoT architecture is a framework that specifies the physical elements, network technical arrangement and setup, operating procedures, and data formats to be used. IoT architecture can differ greatly based on execution; it must be flexible enough for open protocols to handle many network applications.

According to our IOT application it is based on three-layer architecture.

1. Perception Layer
2. Network Layer
3. Application Layer



1. Perception Layer

Perception layer is all about sensors and data through it. Here are all the sensors used for this layer will be –

- Soil Moisture Sensor Module 5V
- DHT11 Temperature and Humidity Sensor
- PH Sensor Module V1.1 with PH Probe for AVR 51
- ESP32 Development Board

2. Network Layer

Network Layer is all about carries and transmits the information collected from the physical objects through sensors. Here the device used for this layer is –

- Raspberry Pi 3 Model B+

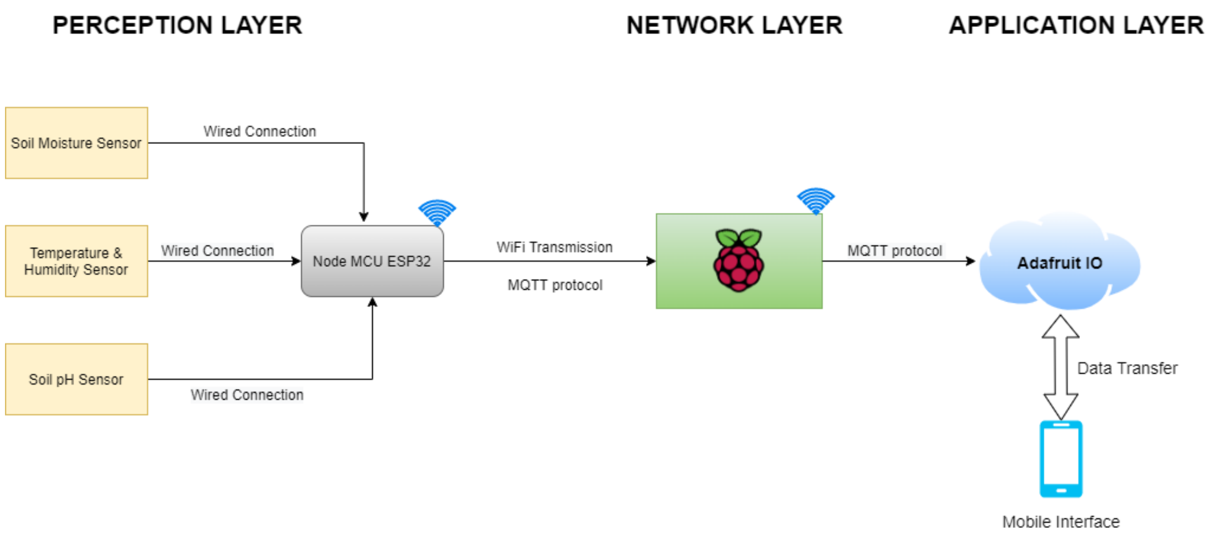
3. Application Layer

The application layer manages all application process based on information obtained from data processing. This application involves sending emails, activating alarm, turn on or off a device, smart agriculture, etc.

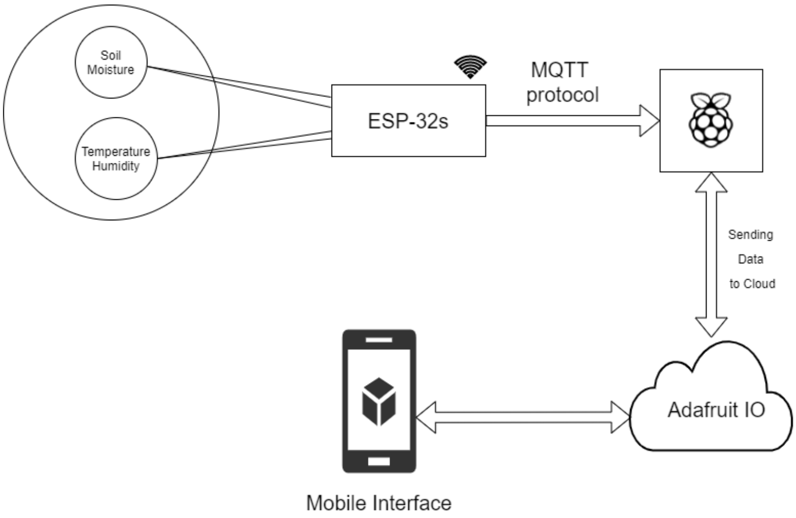
- Adafruit io cloud
- MIT app inventor
- IFTTT applets

1.2 Communication Architecture

IoT Communication architecture is a framework that specifies the communication technologies between the different layers of the architecture.



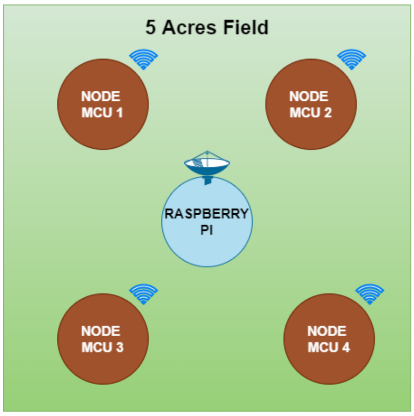
Data Flow

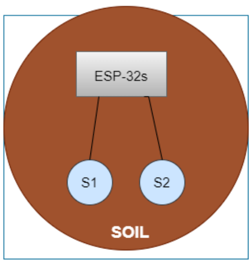


Design Concept

Here we assumed that the user has 5 acres of field and we have to setup our system so, according to this conceptual image we can understand that the node mcu's used here are wirelessly transmitting the data to the main raspberry pi which will be planted on the top pole to receive data from the other gateways.

After all the data will be collected from the different points, we will analyse the data into our datasets and will predict the crop accordingly.





Chapter 4

Implementation

Hardware Used

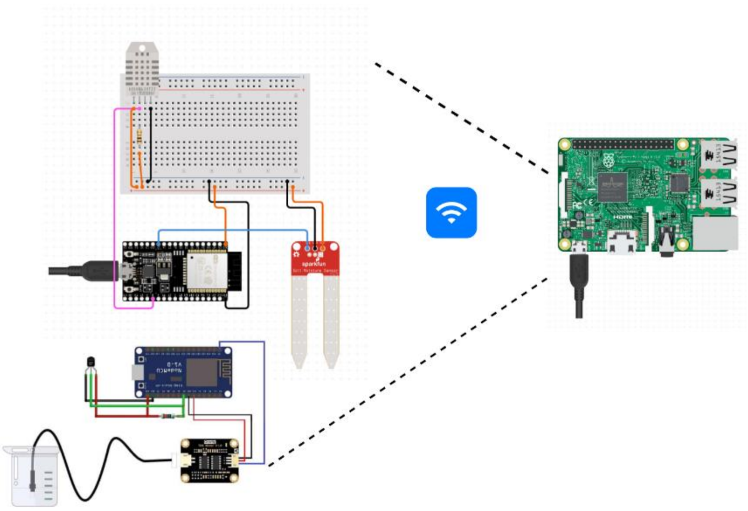
Here is the list of all the hardware used in the prototyping of the IOT application.

S.No.	Hardware Name	Quantity	Category	Est. Price
1.	Digital soil-moisture sensor	1pc	Sensor	Rs.160
2.	DHT-11 temp. & humidity sensor	1pc	Sensor	Rs.172
3.	pH sensor module	1pc	Sensor	Rs.1649
4.	ESP-32s module Node-mcu	1pc	Micro-Controller	Rs.395
5.	Raspberry pi model 3B	1pc	Micro-Processor	Rs.2805
6.	Jumper wires	8pc	Connection	Rs.80
7.	Power Cables	2pc	Connection	Rs.50
8.	TOTAL	-	-	Rs.5311

Here we got all the details about all the hardware we are using in this application. According to this list user purchased the hardware.

Circuit Design

Connection of this IOT application has been done on the basis of the following circuit design.



Here we have connected the soil moisture sensor and DHT11 temp. and humidity sensor with Esp32s node mcu and the pH sensor with Esp8266 module.

All the data collected to both the modules has been transferred to Raspberry pi through MQTT protocol via Wi-Fi signals.

After the successful connections of all the hardware devices we will going to program the code scripts for the sensors.

Coding Implementation

Before programming the code for the connections and sensors there are some pre-requisites given below —

- **Arduino IDE (in PC)**

  Certain libraries need to be installed into the IDE –

  DHT sensor library

  Esp32 wifi manager

  EspMQTT client

  PubSub client

- **MQTT Protocol Setup (in raspberry pi)**

After installing the libraries, we can use the following code to detect and send the sensor values to the pi through MQTT protocol

```
#include "DHT.h"
#include "PubSubClient.h" // Connect and publish to the MQTT

// Code for the ESP32
#include "WiFi.h" // Enables the ESP32 to connect to the local network (via WiFi)
#define DHTPIN 4  // Pin connected to the DHT sensor
#define DHTPI

#define DHTTYPE DHT11  // DHT11 or DHT22
DHT dht(DHTPIN, DHTTYPE);

// WiFi
const char* ssid ="Naman Sharma";                 // Your personal network SSID
const char* wifi_password ="1234567890"; // Your personal network password

// MQTT
const char* mqtt_server = "192.168.103.110";  // IP of the MQTT broker
const char* humidity_topic = "home/livingroom/humidity";
const char* temperature_topic = "home/livingroom/temperature";
const char* soil_topic = "home/livingroom/soil";
const char* mqtt_username = "naman"; // MQTT username
const char* mqtt_password = "1234"; // MQTT password
const char* clientID = "client_livingroom"; // MQTT client ID

// Initialise the WiFi and MQTT Client objects
WiFiClient wifiClient;
// 1883 is the listener port for the Broker
PubSubClient client(mqtt_server, 1883, wifiClient);


```

```
// PUBLISH to the MQTT Broker (topic = Humidity, defined at the beginning)
if (client.publish(humidity_topic, String(h).c_str())) {
  Serial.println("Humidity sent!");
}
// Again, client.publish will return a boolean value depending on whether it succeded or not.
// If the message failed to send, we will try again, as the connection may have broken.
else {
  //Serial.println("Humidity failed to send. Reconnecting to MQTT Broker and trying again");
  client.connect(clientID, mqtt_username, mqtt_password);
  delay(10); // This delay ensures that client.publish doesn't clash with the client.connect call
  client.publish(humidity_topic, String(h).c_str());
}

 // PUBLISH to the MQTT Broker (topic = Temperature, defined at the beginning)
if (client.publish(temperature_topic, String(t).c_str())) {
  Serial.println("Temperature sent!");
}
// Again, client.publish will return a boolean value depending on whether it succeded or not.
// If the message failed to send, we will try again, as the connection may have broken.
else {
  //Serial.println("Temperature failed to send. Reconnecting to MQTT Broker and trying again");
  client.connect(clientID, mqtt_username, mqtt_password);
  delay(10); // This delay ensures that client.publish doesn't clash with the client.connect call
  client.publish(temperature_topic, String(t).c_str());
}
```

Console Output Snippets

```
Python 3.7.3 (/usr/bin/python3)
>>> %Run MQTT_data_test.py
 MQTT to InfluxDB bridge
 Connected with result code 0
 home/livingroom/humidity 95.00
 home/livingroom/temperature 23.50
 home/livingroom/soil 744
 home/livingroom/humidity 95.00
 home/livingroom/temperature 23.50
 home/livingroom/soil 744
 home/livingroom/humidity 95.00
 home/livingroom/temperature 23.50
 home/livingroom/soil 744
 home/livingroom/humidity 95.00
```

```
Python 3.7.3 (/usr/bin/python3)
>>> %Run Soil_data.py
 The Production accuracy from this model is:  87.95698924731182
 MQTT to InfluxDB bridge
 Connected with result code 0
 The predicted crop is muskmelon
```

Output Simulations

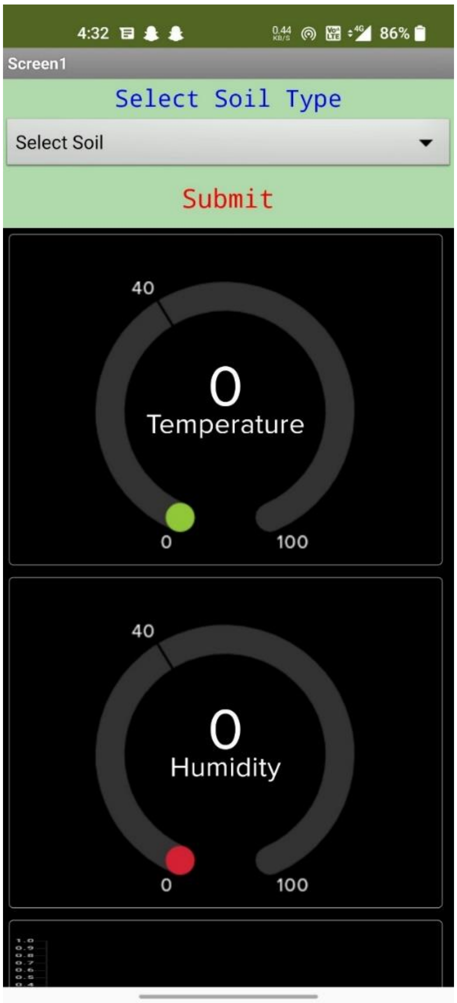
Here we had used Adafruit cloud for getting our outputs simulated and stored in the cloud services. MIT app inventor been used to represent the output stored in the cloud and interact with the user.

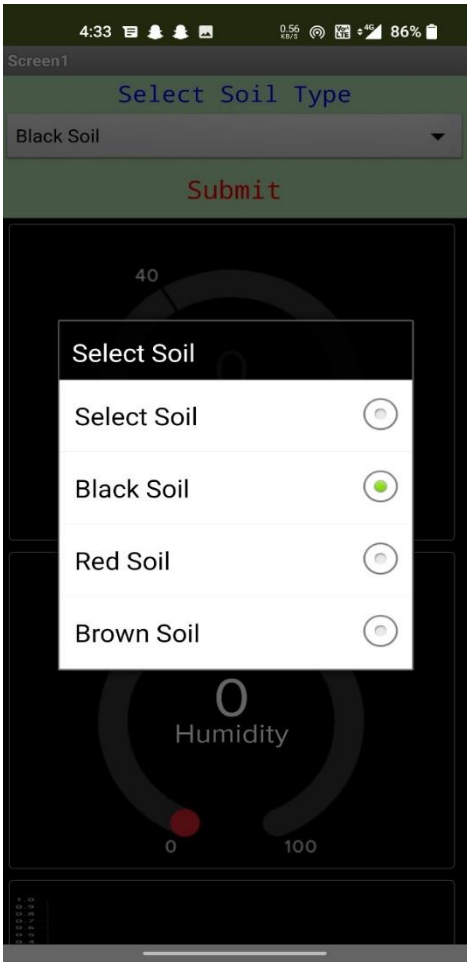
We will represent all the following outputs above –

1. Adafruit IO

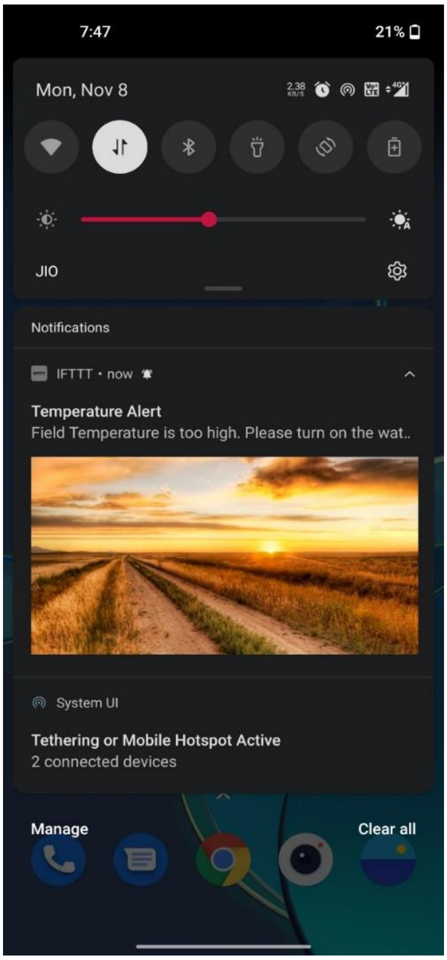
1. Temperature feed
2. Humidity feed
3. Soil Moisture feed
4. Predicted Output feed

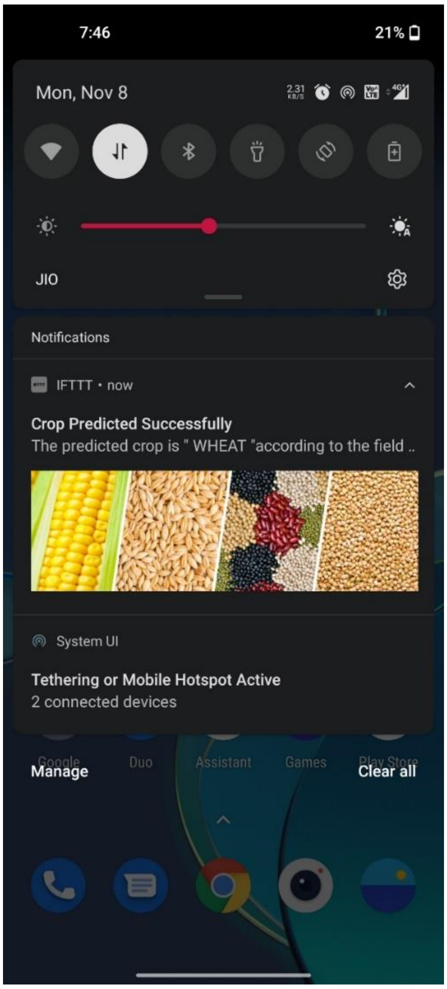
2. MIT app Inventor





3. Applets Notifications





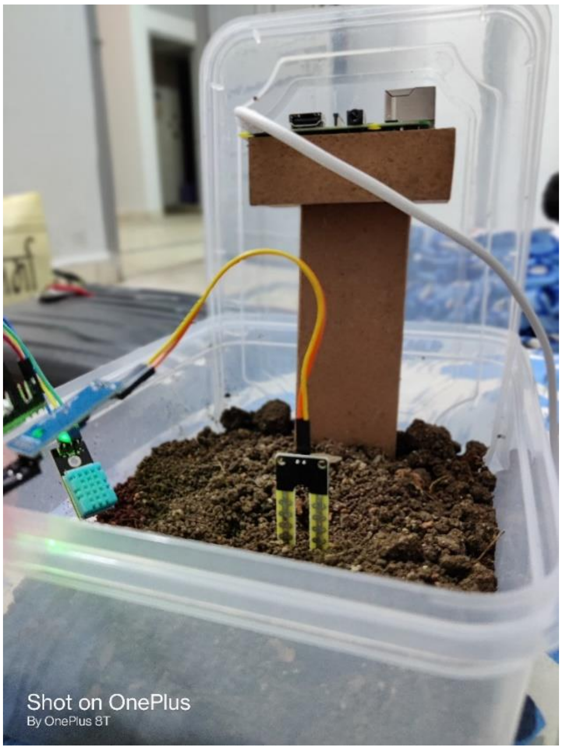
Here we have setup two type of alerts for the user that are as follows –

1. Temperature alert - When temperature of the field rises more than 40℃ than this feature will alert the user and suggest the user for turning on the water pump.
2. Crop Prediction – When the result of crop prediction will be ready the user will be notified by the output.

Physical Prototype

We had done physical prototyping of our model for one of the points with a gateway which can been observed in the following images -

In the Physical Prototyping of the system, it has been observed that the system is working fine and this model can be considered for the real-life field.



As we have completed our implementation phase for now, we can conclude that we have achieved our 70 percent of our objective that we have planned before. Still this project has to been worked on accuracy and its prototyping of the model as field has so many weather challenges too. So that conditions has to be considered.

Chapter 5

Testing and Deployment

IOT testing is a type of testing to check IOT devices. Today there is increasing need to deliver better and faster services. There is a huge demand to access, create, use and share data from any device. The thrust is to provide greater insight and control, over various interconnected IOT devices. Hence, IOT testing framework is important.

This IOT application has been tested on four different scenarios for its better improvement in the future scope –

Device Test on different soil and weather conditions

This device has been tested on different conditions. For this testing case we had different pH value and different Humidity and temperature values and our application has predicted crop accordingly.

PASSED

Device Portability

This device has been tested on its portability nature. As the current prototype of this model has to be reconsidered because this model has once given the result than it has to be used in the next crop season.

FAILED

Affects through natural conditions

This application can be affected through natural condition like in rainy season the fields will be full of water and the moisture sensor will get very high reading. So, our system will suggest crops which require high water level but the actual water level will change so here is a chance of getting less accurate results.

FAILED

Device Scalability for bigger farms

This product is designed in such a way that if the size of the field is bigger multiple test points will be created and the data will be collected on hub. And according to that it will give the results.

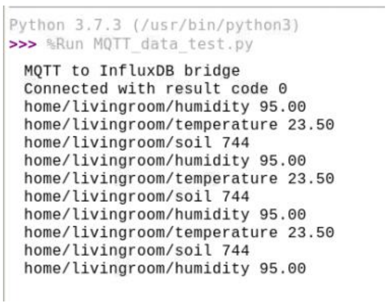
PASSED

Deployment

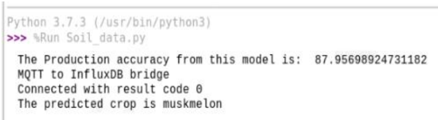
The Deployment phase for the IOT application has been done and it can be observed in the following aspects –

1. Starting from testing out the sensors in the filed condition.

2. Now this value will go to the Raspberry pi through ESP32 via MQTT protocol and this value will be processed into our code and the processed output will be displayed on console.

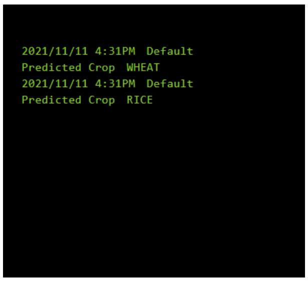


```
Python 3.7.3 (/usr/bin/python3)
>>> %Run MQTT_data_test.py
 MQTT to InfluxDB bridge
 Connected with result code 0
 home/livingroom/humidity 95.00
 home/livingroom/temperature 23.50
 home/livingroom/soil 744
 home/livingroom/humidity 95.00
 home/livingroom/temperature 23.50
 home/livingroom/soil 744
 home/livingroom/humidity 95.00
 home/livingroom/temperature 23.50
 home/livingroom/soil 744
 home/livingroom/humidity 95.00
```

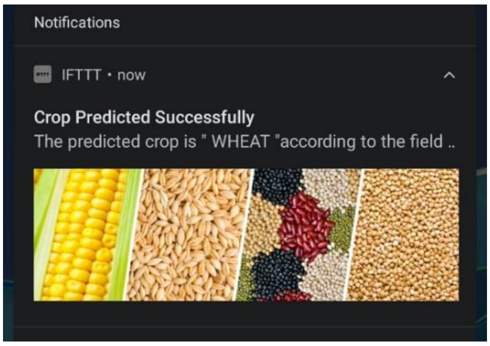


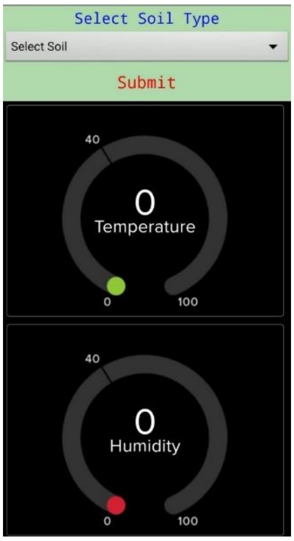
```
Python 3.7.3 (/usr/bin/python3)
>>> %Run Soil_data.py
 The Production accuracy from this model is:  87.95698924731182
 MQTT to InfluxDB bridge
 Connected with result code 0
 The predicted crop is muskmelon
```

3. The values coming from console will be redirected to the cloud database for further more processing.



4. From the Adafruit cloud database the output result will be displayed to the mobile device along with the notifications.





Chapter 6

Conclusion

Here we can conclude that our approach to finish this IOT application has been gone in a right way and we are able to achieve seventy percent of our project what we have planned before. Still this project has to been worked on accuracy and its prototyping of the model as field has so many weather challenges too. So that conditions has to be considered. Also there are future scopes in this project which can be achieved by giving some more time to this project.

Chapter 7

References

To complete this project, we had taken references from many types of documentations, datasheets, research papers and amongst them some are cited below –

https://www.raspberrypi.com/documentation/computers/

https://ieeexplore.ieee.org/document/8688794

https://components101.com/sensors/dht11-temperature-sensor

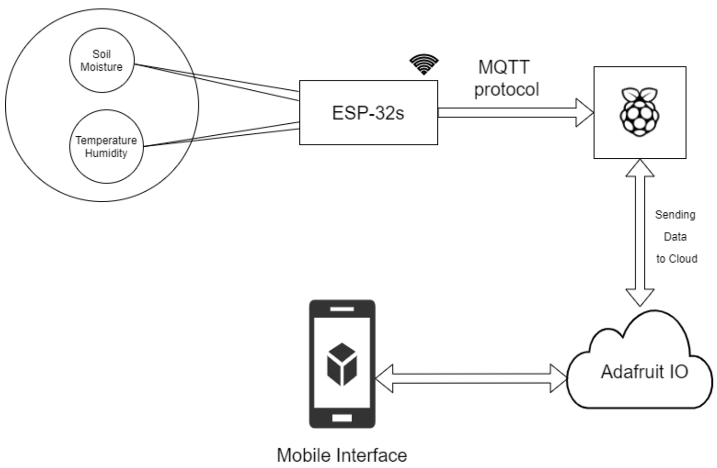
https://nodemcu.readthedocs.io/en/dev-esp32/build/

https://io.adafruit.com/api/docs/#adafruit-io-http-api

Chapter 8

Appendix

The figures and diagrams used to complete this project are been attached below for further references.



Basic representation of our IOT project technology