SmartFarmer - IoT Enabled Smart Farming Application

Team ID: PNT2022TMID19113

Team Members

Team lead	Varshni Soundarya R
Team member 1	Yugendran K M
Team member 2	Viswesvaran K E
Team member 3	Sakthi Sruthi K S

College Name

Sona College of Technology, Salem-636005

GitHub Repositories

IBM-Project-26051-1659982306

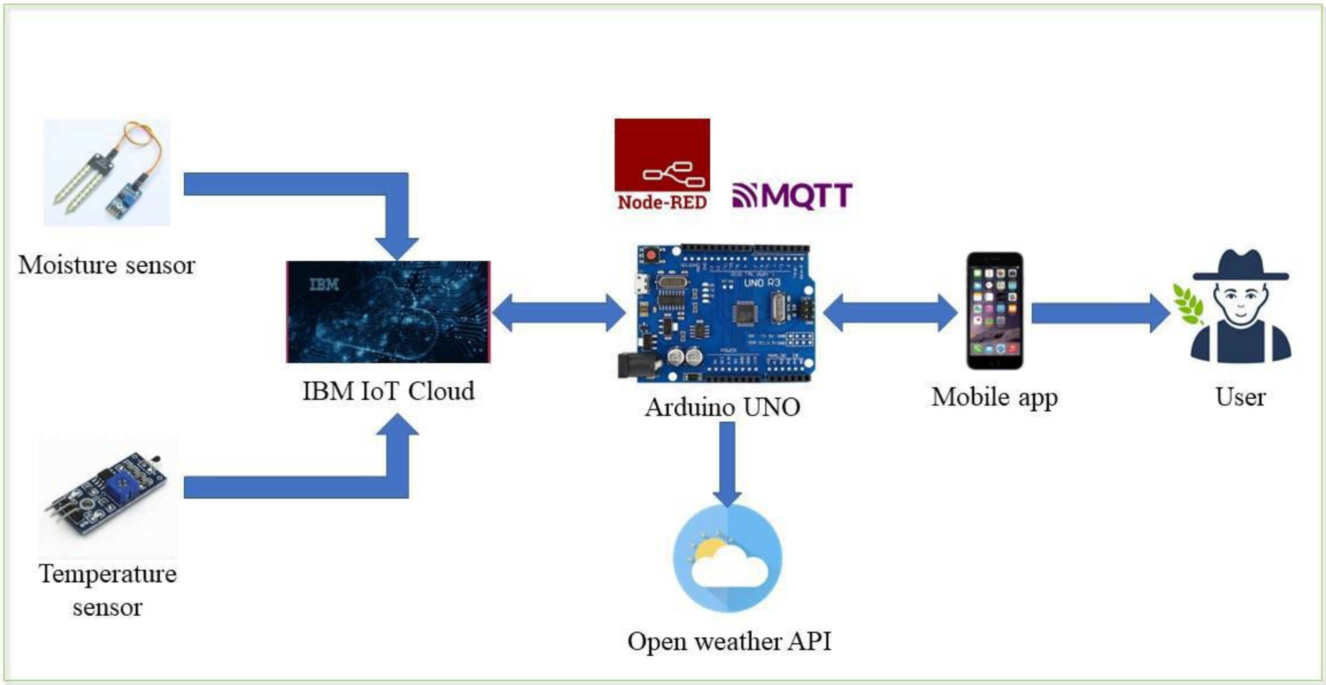
INTRODUCTION:

Our problem statement of this project is to facilitate the farmer’s work easy, by way of helping them to make the agriculture activities much smarter. Our proposed solution contains various parameters to reduce the hard work done in fields by making things smarter. IoT-based agriculture system helps the farmer in monitoring different parameters of his field like soil moisture, temperature, and humidity using some sensors.

PURPOSE:

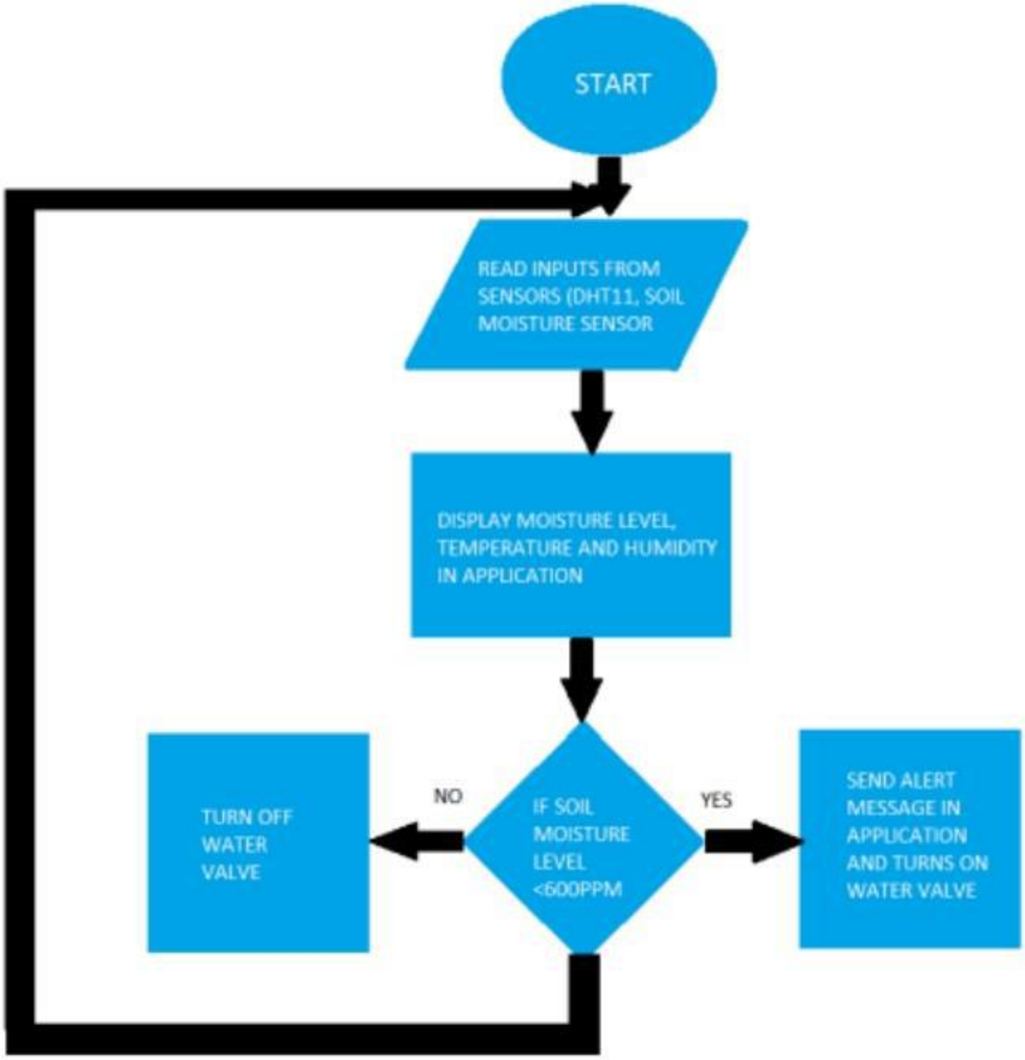
- Farmers can monitor all the sensor parameters by using a mobile application even if the farmer is not near his field. Watering the crop is one of the important tasks for the farmers.
- They can make the decision whether to water the crop or postpone it by monitoring the sensor parameters and controlling the motor pumps from the mobile application itself.
- Similarly, they can also sprinkle the liquid type insecticides by rain pipe system and weedicides by the piping system.
- Also, they will be able to know the plant’s lifetime by using the some of the parameters of his field. So that it will help him to make further decisions as per need.
- Along with it all, he can also be able to control the Animal Repeller system which is kept in the field

BLOCK DIAGRAM:



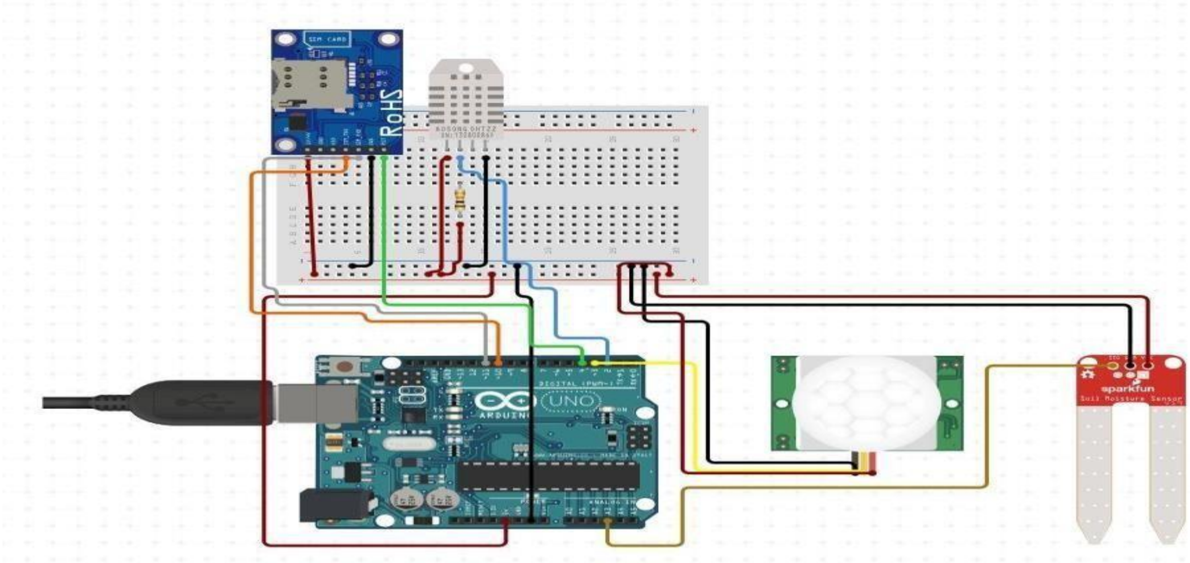
This block Diagram shows the basic outline of the project parameters and the method of process in the devices and their responses.

FLOW DIAGRAM:


START
READ INPUTS FROM SENSORS (DHT11, SOIL MOISTURE SENSOR
DISPLAY MOISTURE LEVEL, TEMPERATURE AND HUMIDITY IN APPLICATION
IF SOIL MOISTURE LEVEL <600PPM
NO
TURN OFF WATER VALVE
YES
SEND ALERT MESSAGE IN APPLICATION AND TURNS ON WATER VALVE

This is the flow of the sensors and their method of responses to the farmer when they access their smart phone to monitor their fields.

CIRCUIT DIAGRAM:



This shows the hardware circuit diagram of the project.

FINAL CODE:

```
//include libraries
#include <dht.h>
#include <SoftwareSerial.h>

//define pins
#define dht_apin A0 // Analog Pin sensor is connected
SoftwareSerial mySerial(7,8); //serial port of gsm
const int sensor_pin = A1; // Soil moisture sensor O/P pin
int pin_out = 9;

//allocate variables
dht DHT;
int c=0;

void setup()
{
pinMode(2, INPUT); //Pin 2 as INPUT
pinMode(3, OUTPUT); //PIN 3 as OUTPUT
pinMode(9, OUTPUT); //output for pump
}

void loop()
{
if (digitalRead(2) == HIGH)
{
digitalWrite(3, HIGH); // turn the LED/Buzz ON
delay(10000); // wait for 100 milli second
digitalWrite(3, LOW); // turn the LED/Buzz OFF
delay(100);
}
Serial.begin(9600);
delay(1000);
DHT.read11(dht_apin); //temprature
float h=DHT.humidity;
float t=DHT.temperature;
delay(5000);

Serial.begin(9600);
float moisture_percentage; //moisture
int sensor_analog;
sensor_analog = analogRead(sensor_pin);
moisture_percentage = ( 100 - ( (sensor_analog/1023.00) * 100 ) );
float m=moisture_percentage;
delay(1000);
if(m<40) //pump
{
while(m<40)
{
digitalWrite(pin_out,HIGH); //open pump
sensor_analog = analogRead(sensor_pin);
moisture_percentage = ( 100 - ( (sensor_analog/1023.00) * 100 ) );
m=moisture_percentage;
delay(1000);
}
digitalWrite(pin_out,LOW); //close pump
}i
f(c>=0)
```

```
{
mySerial.begin(9600);
delay(15000);

Serial.begin(9600);
delay(1000); Serial.print("\r");
delay(1000);

Serial.print("AT+CMGF=1\r");
delay(1000);

Serial.print("AT+CMGS=\"+XXXXXXXXXXX\"\r"); //replace X with 10 digit mobile number
delay(1000);

Serial.print((String)"update>"+(String)"Temprature="+t+(String)"Humidity="+h+(String)"Moisture="+m);
delay(1000);

Serial.write(0x1A);
delay(1000);

mySerial.println("AT+CMGF=1"); //Sets the GSM Module in Text Mode
delay(1000);

mySerial.println("AT+CMGS=\"+XXXXXXXXXXX\"\r"); //replace X with 10 digit mobile number
delay(1000);
mySerial.println((String)"update>"+(String)"Temprature="+t+(String)"Humidity="+h
+(String)"Moisture="+m); //message format

mySerial.println();
delay(100);

Serial.write(0x1A);
delay(1000);
c++;
}
}
```