# Smart-Irrigation-System-With-Leaf-Disease-Detection

Designed and implemented a smart irrigation system integrated with utilizing IoT sensors to optimize water usage based on the crops selected with advance leaf disease detection system with machine learning algorithms and MATLAB to ensure early identification and treatment of plant diseases, resulting in improved crop health productivity

#include "thingProperties.h"

#include <Adafruit\_GFX.h>

#include <Adafruit\_GrayOLED.h>

#include <Adafruit\_SPITFT.h>

#include <Adafruit\_SPITFT\_Macros.h>

#include <gfxfont.h>

// Adafruit SSD1306 - Version: Latest

#include <Adafruit\_SSD1306.h>

#include <splash.h>

// DHT sensor library - Version: Latest

#include <DHT.h>

#include <DHT\_U.h>

// Wire Library for I2C

#include <Wire.h>

// Set OLED size in pixels

#define SCREEN\_WIDTH 128

#define SCREEN\_HEIGHT 64

// Set OLED parameters

#define OLED\_RESET -1

#define SCREEN\_ADDRESS 0x3C

Adafruit\_SSD1306 display(SCREEN\_WIDTH, SCREEN\_HEIGHT, &Wire, OLED\_RESET);

// Define DHT22 Parameters

#define DHTPIN 8

#define DHTTYPE DHT22

DHT dht(DHTPIN, DHTTYPE);

// Define variables for Temperature

float temp;

int control=0;

int moisture\_limit;

// Sensor constants - replace with values from calibration sketch

// Constant for dry sensor

const int DryValue = 2650;

// Constant for wet sensor

const int WetValue = 1800;

// Variables for soil moisture

int soilMoistureValue;

int soilMoisturePercent;

// Analog input port

#define SENSOR\_IN 0

// Relay Port

#define RELAY\_OUT 3

// Pump Status Text

String pump\_status\_text = "OFF";

// Variable for pump trigger

\

void wheat();

void paddy();

void cotton();

void maize();

void setup() {

// Initialize serial and wait for port to open:

Serial.begin(9600);

// This delay gives the chance to wait for a Serial Monitor without blocking if none is found

delay(1500);

// Defined in thingProperties.h

initProperties();

// Connect to Arduino IoT Cloud

ArduinoCloud.begin(ArduinoIoTPreferredConnection);

display.begin(SSD1306\_SWITCHCAPVCC, SCREEN\_ADDRESS);

display.clearDisplay();

// Initialize DHT22

dht.begin();

// Set ADC to use 12 bits

analogReadResolution(12);

// Set Relay as Output

pinMode(RELAY\_OUT, OUTPUT);

// Turn off relay

digitalWrite(RELAY\_OUT, LOW);

// Set Pump Status to Off

pump\_status = false;

setDebugMessageLevel(2);

ArduinoCloud.printDebugInfo();

}

void loop() {

ArduinoCloud.update();

ArduinoCloud.update();

// Get temperature

temp = dht.readTemperature();

// Pass temperature values to cloud variables

current\_temperature = temp;

// Get soil mositure value

soilMoistureValue = analogRead(SENSOR\_IN);

// Determine soil moisture percentage value

soilMoisturePercent = map(soilMoistureValue, DryValue, WetValue, 0, 100);

// Keep values between 0 and 100

soilMoisturePercent = constrain(soilMoisturePercent, 0, 100);

// Print raw value to serial monitor for sensor calibration

Serial.println(soilMoistureValue);

// Pass soil moisture to cloud variable

current\_moisture = soilMoisturePercent;

if(my\_control==control){

switch(option)

{

case 0: wheat();

break;

case 1: paddy();

break;

case 2: cotton();

break;

case 3: maize();

break;

case 4: pumpOn();

break;

case 5: pumpOff();

break;

default: Serial.println("Error in finding the crop");

break;

}

if(option !=4 && option!=5)

{

if (soilMoisturePercent<moisture\_limit) {

// Turn pump on

pumpOn();

} else {

// Turn pump off

pumpOff();

}

}

}

else

{

if (soilMoisturePercent<my\_control) {

// Turn pump on

pumpOn();

} else {

// Turn pump off

pumpOff();

}

}

// Cycle values on OLED Display

printOLED(35, "PUMP", 40, pump\_status\_text, 2000);

printOLED(35, "TEMP", 10, String(temp) + "C", 2000);

printOLED(35, "MOIST", 30, String(soilMoisturePercent) + "%", 2000);

}

void wheat()

{

moisture\_limit=25;

}

void paddy()

{

moisture\_limit=60;

}

void cotton()

{

moisture\_limit=50;

}

void maize()

{

moisture\_limit=65;

}

void pumpOn() {

// Turn pump on

digitalWrite(RELAY\_OUT, HIGH);

pump\_status\_text = "ON";

pump\_status = true;

}

void pumpOff() {

// Turn pump off

digitalWrite(RELAY\_OUT, LOW);

pump\_status\_text = "OFF";

pump\_status = false;

}

void printOLED(int top\_cursor, String top\_text, int main\_cursor, String main\_text, int delay\_time){

// Prints to OLED and holds display for delay\_time

display.setCursor(top\_cursor, 0);

display.setTextSize(2);

display.setTextColor(WHITE);

display.println(top\_text);

display.setCursor(main\_cursor, 40);

display.setTextSize(3);

display.setTextColor(WHITE);

display.print(main\_text);

display.display();

delay(delay\_time);

display.clearDisplay();

}

void onOptionChange() {

}

void onMyControlChange() {

// Add your code here to act upon MyControl change

}

/\*

Since PumpStatus is READ\_WRITE variable, onPumpStatusChange() is

executed every time a new value is received from IoT Cloud.

\*/

void onPumpStatusChange() {

}