**TINKER CARD CODE :**

#include <Adafruit\_NeoPixel.h>

#define PIN 2 // pin Neopixel is attached to

#define SENSOR A0 //input pin for Potentiometer

#define NUMPIXELS 1 // number of neopixels in strip

/THese values adjust what is considered wet or dry/

#define dryThreshold 50 //below this value, begin alerting dry, turn red;

#define wetThreshold 200 //above this value, begin alerting wet,turn blue;

#define thresholdCenter (dryThreshold + wetThreshold)/2 //Brightest Green point

#define crossFade 20 //how much blue and red should fade in to green

char degree = 176; //ASCI value of Degre

Adafruit\_NeoPixel pixels = Adafruit\_NeoPixel(NUMPIXELS, PIN, NEO\_GRB + NEO\_KHZ800);

int delayval =1000; // timing delay in milliseconds

int piezoPin=8;

int redColor = 0;

int greenColor = 0;

int blueColor = 0;

int motor=3;

//Sensor reading and the value converted from 1024 to 255 for output

int sensorValue = 0;

int transitionValue = 0;

int const pin\_gas = A2;

int led\_green = 7;

int led\_yellow = 6;

int led\_orange = 5;

int led\_red = 4;

int sensorValue\_light= 0;

const int pingPin\_dist = 10;

const int ledPin\_dist = 11;

int gas\_piezo=12;

void setup() {

Serial.begin(9600);

pinMode(ledPin\_dist, OUTPUT);

pinMode(led\_green, OUTPUT);

pinMode(led\_yellow, OUTPUT);

pinMode(led\_orange, OUTPUT);

pinMode(led\_red, OUTPUT);

Serial.begin(9600);

pinMode(motor, OUTPUT);

pinMode(A1,INPUT);

Serial.begin(9600);

// Initialize the NeoPixel library.

pixels.begin();

pinMode(SENSOR,INPUT);

pinMode(A3, INPUT);

pinMode(9, OUTPUT);

Serial.begin(9600);

}

void loop() {

long duration, cm;

pinMode(pingPin\_dist, OUTPUT);

digitalWrite(pingPin\_dist, LOW);

delayMicroseconds(2);

digitalWrite(pingPin\_dist, HIGH);

delayMicroseconds(5);

digitalWrite(pingPin\_dist, LOW);

pinMode(pingPin\_dist, INPUT);

duration = pulseIn(pingPin\_dist, HIGH);

cm = microsecondsToCentimeters(duration);

Serial.print("Distance: ");

Serial.print(cm);

Serial.print("cm");

Serial.println();

if(cm < 100) {

tone(ledPin\_dist, 5000,600);

digitalWrite(ledPin\_dist, HIGH);

}

else {

digitalWrite(ledPin\_dist, LOW);

}

int valor = analogRead(pin\_gas);

valor = map(valor, 300, 750, 0, 100);

digitalWrite(led\_green, HIGH);

digitalWrite(led\_yellow, valor >= 30 ? HIGH : LOW);

digitalWrite(led\_orange, valor >= 50 ? HIGH : LOW);

digitalWrite(led\_red, valor >= 80 ? HIGH : LOW);

if(valor>=80)

{

tone(gas\_piezo, 3000, 500);

}

Serial.print("Soil moisture= ");

int humidity=analogRead(A0);

if(humidity>=0 && humidity<=23)

{

Serial.print(humidity);

}

else{

Serial.print((humidity-23)/10);

}

Serial.println("%");

int tmp = analogRead(A1);

float voltage = (tmp \* 5.0)/1024;//(5\*temp)/1024 is to convert the 10 bit number to a voltage reading.

float milliVolt = voltage \* 1000;//This is multiplied by 1000 to convert it to millivolt.

float tmpCel = (milliVolt-500)/10 ;//For TMP36 sensor. Range(−40°C to +125°C)

float tmpFer = (((tmpCel\*9)/5)+32);//used to convert Celsius -> Fahrenheit

if( tmpCel>=40 || tmpCel<15)

{

tone( piezoPin, 3000, 500);

}

Serial.print("Celsius: ");

Serial.print(tmpCel);

Serial.println(degree);

Serial.print("Fahrenheit: ");

Serial.println(tmpFer);

Serial.println();

sensorValue = analogRead(SENSOR);

transitionValue = map(sensorValue,0,1023,0,255);

setColor();

// read the value from the sensor

sensorValue\_light = analogRead(A3);

// print the sensor reading so you know its range

Serial.print("The photoresistor value is: ");

Serial.println(sensorValue\_light);

// map the sensor reading to a range for the LED

analogWrite(9, map(sensorValue\_light, 0, 1023, 0, 255));

// pixels.Color takes RGB values, from 0,0,0 up to 255,255,255

pixels.setPixelColor(0,redColor,greenColor,blueColor);

// This sends the updated pixel color to the hardware.

pixels.show();

// Delay for a period of time (in milliseconds).

delay(delayval);

}

long microsecondsToCentimeters(long microseconds) {

// The speed of sound is 340 m/s or 29 microseconds per centimeter.

// The ping travels out and back, so to find the distance of the

// object we take half of the distance travelled.

return microseconds / 29 / 2;

}

void setColor(){

if((transitionValue <= dryThreshold + crossFade) && (transitionValue >= 0 ))

{

digitalWrite(motor, LOW);

}

else if((transitionValue >= wetThreshold - crossFade && transitionValue >=230))

{

digitalWrite(motor, HIGH);

}

//red value greater towards higher resistance/drier

redColor = ((transitionValue <= dryThreshold + crossFade) && (transitionValue >= 0 ))? map(transitionValue,0,dryThreshold + crossFade,255,0) : 0;

//blue value greater towards lower resistance/wetter

blueColor = (transitionValue >= wetThreshold - crossFade && transitionValue <= 255)? map(transitionValue,wetThreshold - crossFade,255,0,255):0;

//green value towrds middle resistance

if(transitionValue >= dryThreshold && transitionValue <= thresholdCenter)

{

greenColor = map(transitionValue,dryThreshold,thresholdCenter,0,255);

}

else if(transitionValue > thresholdCenter && transitionValue < wetThreshold)

{

greenColor = map(transitionValue,dryThreshold,thresholdCenter,255,0);

}

else{

greenColor = 0;

}

}