

SMART CROPPING SYSTEM

TEAM ID	PNT2022TMID40841
PROJECT NAME	IoT- based smart crop protection system for argiculture

Description:

This System is used to help farmers in the irrigation process. The System provides data on the parameters which can be used to monitor the condition of the field to maintain and protect the crops. The parameters like temperature, soil moisture, etc., can be accessed through the system. The sensors in the system monitor the parameters and provide them to the farmer to take the necessary measures.

Program:

```
#include <Wire.h>
#include <Servo.h>
#include <Adafruit_LiquidCrystal.h>

Servo s;
int e = 4;
int t = 5;
int r = 12;
int g = 11;
int b = 10;
int sec = 0;
int Sensor = 0;
int data = 0;
int motorPin = 9;
Adafruit_LiquidCrystal lcd(0);

void setup()
{
    Wire.begin();
```

```

pinMode(A0,INPUT);    //Temperature Sensor
pinMode(A1,INPUT);    //Soil Moisture Sensor
pinMode(t,OUTPUT);    //Ultra sonic Trigger
pinMode(e,INPUT);    //Ultra sonic Echo
pinMode(b,OUTPUT);    //GREEN light for LED
pinMode(g,OUTPUT);    //BLUE light for LED
pinMode(r,OUTPUT);    //RED light for LED
pinMode(motorPin, OUTPUT); //DC motor
s.attach(3);          // Servo Motor
lcd.begin(16, 2);      //LCD 16x2 Display
lcd.setBacklight(0);
Serial.begin(9600);
}

float readDistanceCM(){
    digitalWrite(t, LOW);
    delayMicroseconds(2);
    digitalWrite(t, HIGH);
    delayMicroseconds(10);
    digitalWrite(t, LOW);
    int duration = pulseIn(e, HIGH);
    return duration * 0.034 / 2;
}

void loop(){
    //Soil Moisture:
    Sensor = analogRead(A1);    //Reads data from Soil Moisture sensor
    data = map(Sensor,0, 1023, 0, 100);    //Low analog value indicates HIGH moisture
level and High analog value indicates LOW moisture level
    //data = map(analogValue,fromLOW,fromHIGH,toLOW,toHIGH)
    Serial.print("Soil Moisture value:");
    Serial.println(data);
    //data = 0' indicates wet and 'data = 100' indicates dry

    //Temperature:
    double a = analogRead (A0);    //Reads data from Temperature sensor
    double t = (((a/1024)*5)-0.5)*100;

```

```
Serial.print("Temperature value:");  
Serial.println(t);
```

```
//Ultrasonic sensor:  
float distance = readDistanceCM();  
Serial.print("Measured distance: ");  
Serial.println(readDistanceCM());
```

```
    //LCD Display:  
    lcd.setBacklight(1);  
    lcd.clear();
```

```
//Conditions:  
if (t>40 & t<50){  
    digitalWrite(b,0);  
    digitalWrite(g,1);  
    digitalWrite(r,0);  
    s.write(90);  
    digitalWrite(motorPin, HIGH);  
    Serial.println("Water Partially Flows");  
}
```

```
else if (t>50){  
    digitalWrite(b,1);  
    digitalWrite(g,1);  
    digitalWrite(r,0);  
    s.write(180);  
    digitalWrite(motorPin, HIGH);  
    Serial.println("Water Fully Flows");  
}
```

```
else if (t>30 & data<30){  
    digitalWrite(b,1);  
    digitalWrite(g,1);  
    digitalWrite(r,0);  
        s.write(90);  
    digitalWrite(motorPin, HIGH);
```

```
    Serial.println("Water Partially Flows");  
}
```

```
else if (data<50){  
    digitalWrite(b,0);  
    digitalWrite(g,0);  
    digitalWrite(r,1);  
    s.write(90);  
    digitalWrite(motorPin, HIGH);  
    Serial.println("Water Partially Flows");  
}
```

```
else if (distance < 10){  
    digitalWrite(b, 0);  
    digitalWrite(g, 0);  
    digitalWrite(r, 1);  
    s.write(0);  
    digitalWrite(motorPin, LOW);  
    Serial.println("Water Does Not Flow");  
    lcd.clear();  
    lcd.println("Drain the water");  
}
```

```
else{  
    digitalWrite(b,1);  
    digitalWrite(g,0);  
    digitalWrite(r,0);  
    s.write(0);  
    digitalWrite(motorPin,      LOW);  
    Serial.println("Water Does Not Flow");  
}
```

```
    lcd.setCursor(0,0);  
    lcd.print("Temp:");  
lcd.println(t);  
    lcd.println("degree");  
    lcd.setCursor(0,1);
```

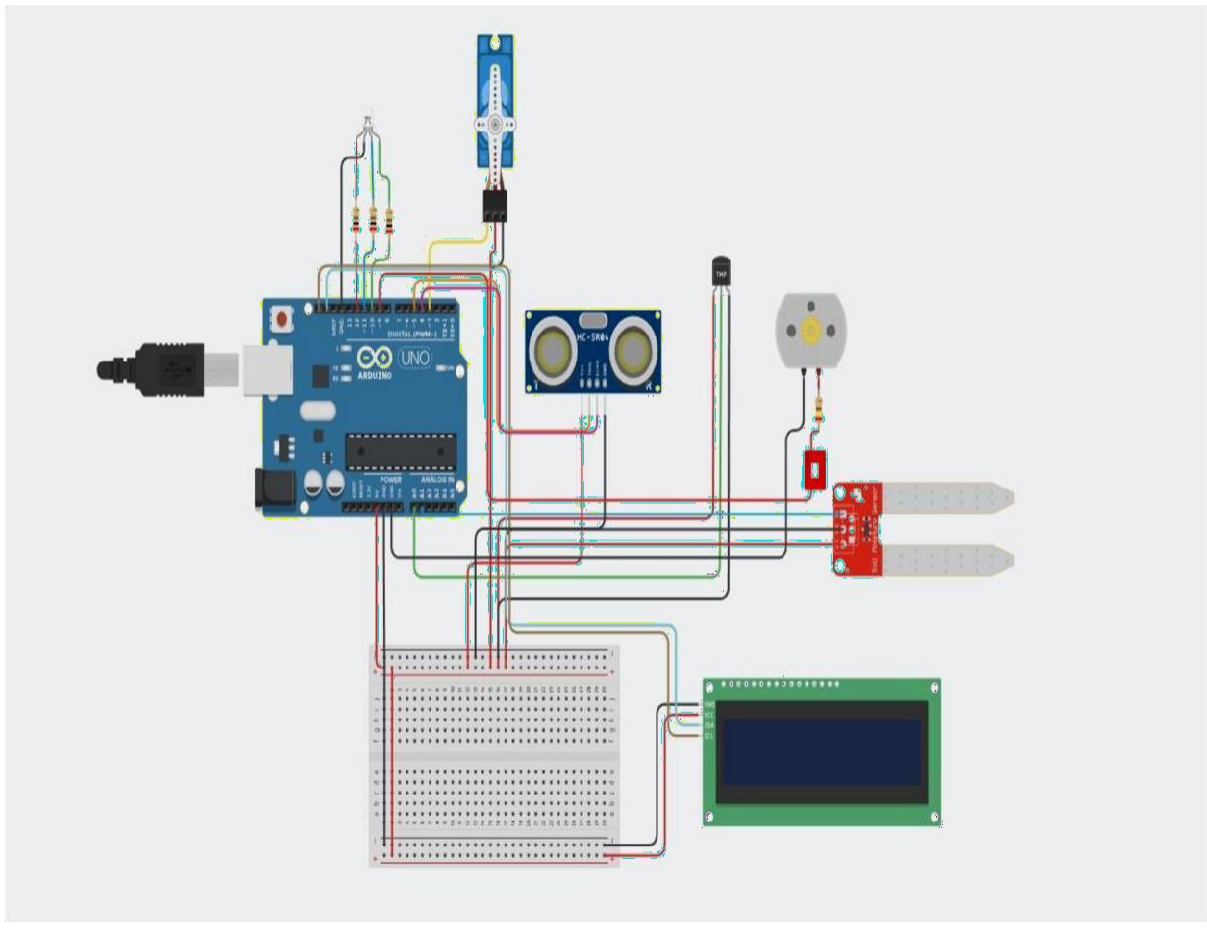
```

        lcd.print("Soil Moisture:");
    lcd.println(data);
        lcd.println("%");

    Serial.println(".....");
    delay(1000);
}

```

Circuit Diagram:



Component Used:

Name	Quantity	Component
U1A1	1	Arduino Uno R3
S1RVOMS	1	Positional Micro Servo
D1LED	1	LED RGB
R1Green LED Resistor R2Red LED Resistor R3Blue LED Resistor	3	200Ω Resistor
S1NSMS	1	Soil Moisture Sensor
M1Small 5V DC Motor	1	DC Motor
R1Motor Resistor	1	240Ω Resistor
U1TS	1	Temperature Sensor (TMP36)
D1S1Ultrasonic Distance Sensor	1	Ultrasonic Distance Sensor
U1S	1	MCP23008-based 16x2 LCD-16 x 2 (I2C)
S1WDPST Switch	1	DIP Switch DPST

OUTPUT:

Tremendous Esboo-Luulia

All changes saved

Code

Stop Simulation

Send To

Simulator time: 00:00:03.659

1

2

3

4

5

6

7

8

9

10

11

12

13

14

#include <Wire.h>

#include <Servo.h>

#include <Adafruit_LiquidCrystal.h>

Servo s;

int e = 4;

int t = 5;

int r = 12;

int g = 11;

int b = 10;

int sec = 0;

int Sensor = 0;

int data = 0;

Serial Monitor

Soil Moisture value:100

Temperature value:449.51

Measured distance: 0.00

Water Fully Flows

Soil Moisture value:100

Temperature value:449.51

Measured distance:

Send

Clear

