Smart Irrigation System using Arduino

Description:

A smart irrigation system using Arduino Uno, moisture sensor, rain sensor, water pump, and DHT11 is a system that can automatically water plants or crops based on their moisture levels, weather conditions, and temperature.

Overall, this smart irrigation system helps conserve water by only watering when necessary and ensures that plants or crops receive the appropriate amount of water based on their needs and environmental conditions.

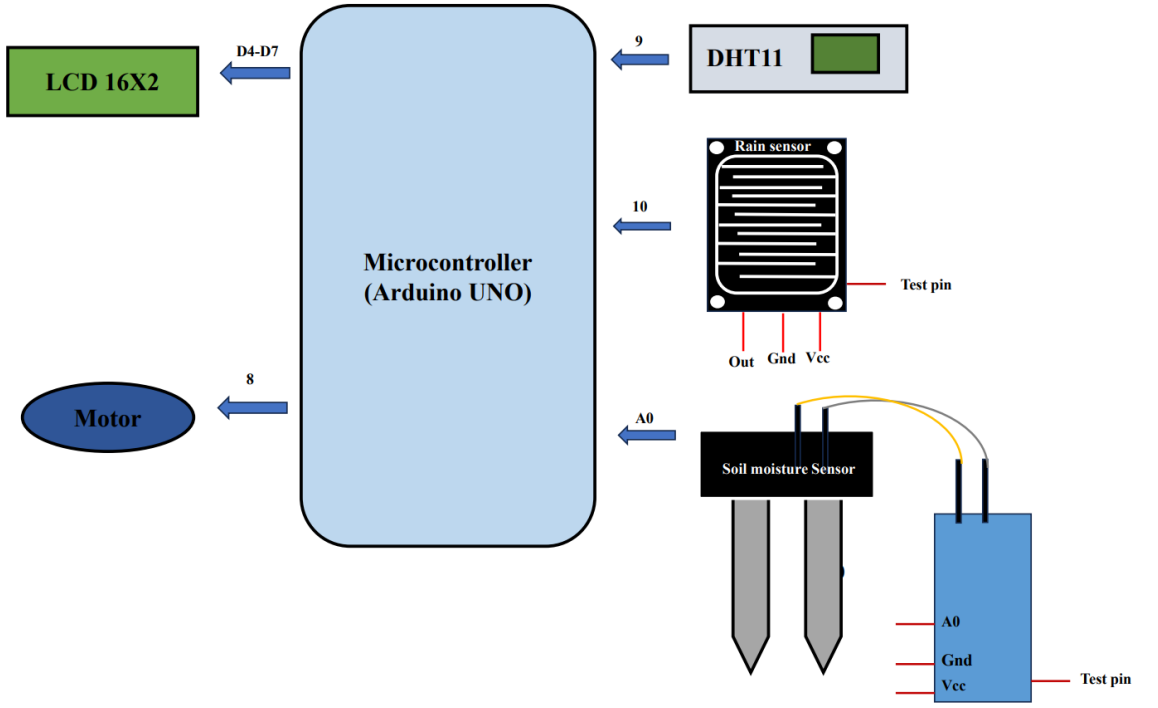
Components Description:

Moisture Sensor: The moisture sensor detects the moisture level in the soil and sends the data to the Arduino Uno. The Arduino Uno then analyses the data and decides whether to turn on the water pump or not. If the moisture level is below a certain threshold, the water pump will be turned on to water the plants or crops.

Rain sensor: The rain sensor detects rainfall and sends the data to the Arduino Uno. If there is enough rainfall, the water pump will not be turned on, as the plants or crops have already received enough water.

DHT11 sensor: The DHT11 sensor detects temperature and humidity levels and sends the data to the Arduino Uno. The Arduino Uno uses this data to determine if it is too hot or too humid for the plants or crops, and adjusts the watering schedule accordingly.

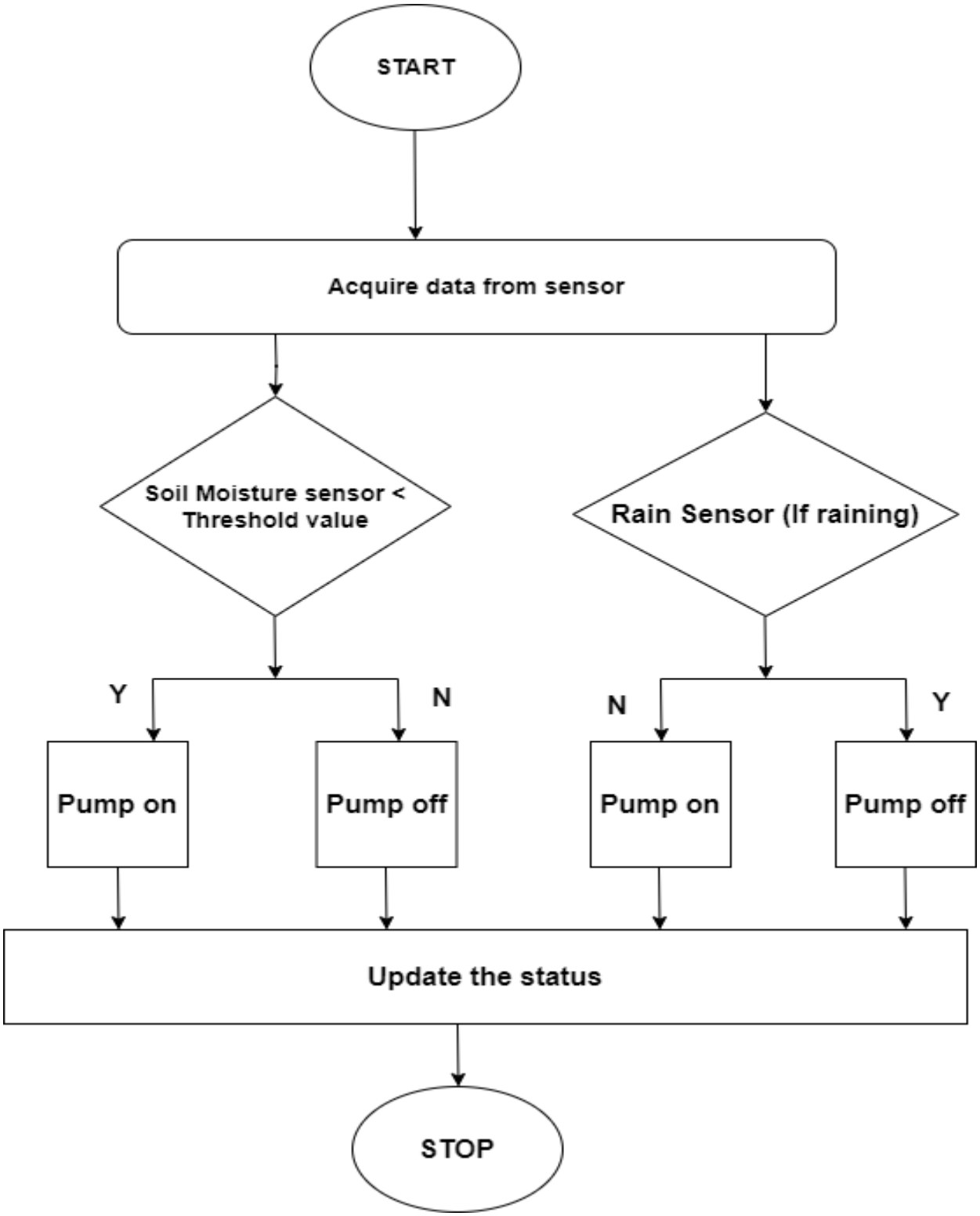
Block Diagram:



Input and Output:

S.No	Descrption	Name	Type	Data Direction	Specification	Remarks
1	DHT11 VDD	VDD	INP	DI	Digital	Active High
2	DHT11 GND	GND	INP	DI	Digital	Active High
3	DHT11 DATA	9	INP	DI	Digital	Active High
4	Rain Sensor OUT	10	INP	DI	Digital	Active High
5	Rain Sensor VCC	VCC	INP	DI	Digital	Active High
6	Rain Sesnsor GND	GND	INP	DI	Digital	Active High
7	Soil Moisture Sensor GND	GND	INP	DI	Digital	Active High
8	Soil Moisture Sensor VCC	VCC	INP	DI	Digital	Active High
9	Soil Moisture Sensor TP	RV1	INP	DI	Digital	Active High
10	Soil Moisture Sensor A0	A0	INP	DI	Digital	Active High
11	LCD DATA PIN	D4	OUT	D0	Digital	Active High
12	LCD DATA PIN	D5	OUT	D0	Digital	Active High
13	LCD DATA PIN	D6	OUT	D0	Digital	Active High
14	LCD DATA PIN	D7	OUT	D0	Digital	Active High
15	LCD RST	RS	OUT	D0	Digital	Active High
16	LCD EN	EN	OUT	D0	Digital	Active High
17	MOTOR	8	OUT	D0	Digital	Active High

Flow Chart:


START
Acquire data from sensor
Soil Moisture sensor < Threshold value
Rain Sensor (If raining)
Y
N
N
Y
Pump on
Pump off
Pump on
Pump off
Update the status
STOP

Source Code:

```
#include <LiquidCrystal.h>
#include "DHT.h"

LiquidCrystal lcd(2, 3, 4, 5, 6,7);
const int relay_Pin = 8;
const int DHT11_Sesnor = 9;
const int moisture_sensor = A0;
const int rain_Sesnor = 10;

#define DHTTYPE DHT11
int  moisture_sensor_value;
int rain_Sesnor_value;
float humudity_value,temprature_value;
DHT dht(DHT11_Sesnor, DHTTYPE);

void setup() {
  Serial.begin(9600);
  pinMode(relay_Pin, OUTPUT);
  pinMode(rain_Sesnor, INPUT);
  lcd.begin(16, 2);
  lcd.print("Smart Irrigation ");
  lcd.setCursor(0,2);
  lcd.print(" SYSTEM");
  digitalWrite(relay_Pin, LOW);
  dht.begin();
   delay(500);
}
void loop()
{

  readDTH11_Sesnor();
  moisture_level_detected();
  water_motor_start();
}


void readDTH11_Sesnor()
{

  // Reading temperature or humidity takes about 250 milliseconds!
  // Sensor readings may also be up to 2 seconds 'old' (its a very slow
sensor)
  humudity_value = dht.readHumidity();
  // Read temperature as Celsius (the default)
  temprature_value = dht.readTemperature();

  // Check if any reads failed and exit early (to try again).
```

```
  if (isnan(humudity_value) || isnan(temprature_value)) {
    Serial.println(("Failed to read from DHT sensor!"));
    return;
  }

  Serial.print((" Humidity: "));
  Serial.print(humudity_value);
  Serial.print(("%"));
  lcd.clear();
  lcd.print("Humidity %: ");
  lcd.setCursor(0,2);
  lcd.print(humudity_value);
  Serial.print("\n");
  delay(500);
  Serial.print(("Temperature: "));
  Serial.print(temprature_value);
  Serial.print(("C "));
  lcd.clear();
  lcd.print("Temperature degCel");
  lcd.setCursor(0,2);
  lcd.print(temprature_value);
  Serial.print("\n");
  delay(500);
}

void moisture_level_detected()
{

  moisture_sensor_value = analogRead(moisture_sensor);
  Serial.println("Moisture Level : ");
  Serial.println(moisture_sensor_value);
  lcd.clear();
  lcd.print("Moisture Level :");
  lcd.setCursor(0,2);
  lcd.print(moisture_sensor_value);
  delay(500);
}

void water_motor_start()
{

 rain_Sesnor_value = digitalRead(rain_Sesnor);
 delay(500);
 if(rain_Sesnor_value == false)
 {
    if(moisture_sensor_value > 700 )
    {
      digitalWrite(relay_Pin, HIGH);
```

```
      lcd.clear();
      lcd.print("Low water level");
      lcd.setCursor(0,2);
      lcd.print("Motor ON");
      delay(500);
    }
    else
    {
      digitalWrite(relay_Pin, LOW);
      lcd.clear();
      lcd.print("Water Level Ok");
      lcd.setCursor(0,2);
      lcd.print("Motor OFF");
      delay(500);
    }
 }
 else
 {
      digitalWrite(relay_Pin, LOW);
      lcd.clear();
      lcd.print("Rain Detected");
      lcd.setCursor(0,2);
      lcd.print("Motor OFF");
  delay(500);
 }
}
```

Schematic: