Project Report – Group 7 (Section 2A)
SMART IRRIGATION SYSTEM

Introduction:

We group of four members chose to make a project useful in our daily life.

In INDIA we are supposed to say that "JAI JAWAN, JAI KISAN ".

So, we are making a project which is helpful for farmers who fill our stomach every day.

This project can simplify things to the farmer and also saves water and money which play crucial role in our life.

Indian agriculture is mainly reliant on the monsoon which is not a reliable source of water. Therefore, there is a need for a smart irrigation system in the country which can provide sufficient amount of water to the farms or fields according to their soil's moisture content.

Smart irrigation system is in other words a water conservation system that monitor moisture-related and temperature conditions on your property and automatically adjust watering to optimal levels.

Importance in **INDIA**:

India is growing at a rapid pace and so is the use of technology in the growing sectors of the country. A major mass of the population is still dependent and practicing agriculture as its primary source of income. India has been in a continuous tryst with its farming infra, practices and associated communities since independence. With the sector still contributing around 15-20% to the national GDP of the country over few decades, and its diverse cum changing needs across its regions, India has been driving necessary and timely interventions at Industry, Institution, and individual farmer level for its constant manifestation. As agriculture is a sector that emotionally appeals to the masses.

Advantages of SMART IRRIGATION SYSTEM includes:

- Reduced human efforts
- More accuracy in irrigation process
- A unified view of soil characteristics, including moisture and nutrient contents
- Fast information in case of abnormalities
- Better long-term landscape health

Objectives:

- To make a Smart Irrigation System that is useful to even common people. We have tried to make the system the easiest to understand.
- We basically wanted the water pump to turn ON when the temperature goes above a certain limit and the soil moisture goes below a certain limit and to turn OFF when these conditions aren't satisfied.
- We also wanted to include an LCD screen to display the temperature in Celsius and the status of the water pump.

Experimental Materials and Methods:

Components used:

Name	Quantity	Cost in Rs.
Arduino Uno R3	1	650
Water Pump	1	85
LCD 16 x 2	1	120
330 Ω Resistor	1	1
0.2 kΩ Resistor	3	1
Red LED	1	1
Green LED	1	1
250 kΩ Potentiometer	1	15
Temperature Sensor [LM35]	1	60
5V Relay module	1	60
Soil moisture sensor	1	100
Jumper Wires	35	70
9V Battery	1	20

Arduino UNO R3:

Arduino Uno is a microcontroller board, an 8-bit microcontroller with 32KB of Flash memory and 2KB of RAM. It has 14 digital input/output pins (of which 6 can be used as PWM outputs), 6 analog inputs, a 16 MHz ceramic resonator, a USB connection, a power jack, an ICSP header and a reset button. It contains everything needed to support the microcontroller. One can simply connect it to a computer with a USB cable or power it with a AC-to-DC adapter or battery to get started.

Water Pump:

Water is drawn into the pump though the inlet side when a pressure difference is made within the pump system, the water wants to move from an area of high pressure to an area of low pressure. This is used to move the water through the pump to the outlet side and into a hose.

Temperature sensor (LM35):

LM35 sensor uses the basic principle of a diode. As the temperature increases, the voltage across a diode increases at a known rate. By precisely amplifying the voltage change, it is easy to generate an analog signal that is directly proportional to temperature.

It is calibrated directly in Celsius (Centigrade). It has a 0.5°C ensured accuracy (at 25°C). It is rated for full −55°C to 150°C range. It is suitable for remote applications.

LM35 operates from 4 V to 30 V with less than 60-μA current drain low self-heating, 0.08°C in Still Air.

Lm35 output voltage is proportional to centigrade/Celsius temperature. Lm35 Celsius/centigrade resolution is 10 mills volt. 10 mills volt represent one degree centigrade/Celsius. So, if Lm35 outputs 100 mills volts the equivalent temperature in centigrade/Celsius will be 100/10 = 10 centigrade/Celsius. Lm35 can measure from -50 degree centigrade/Celsius up to 150 degrees centigrade/Celsius.

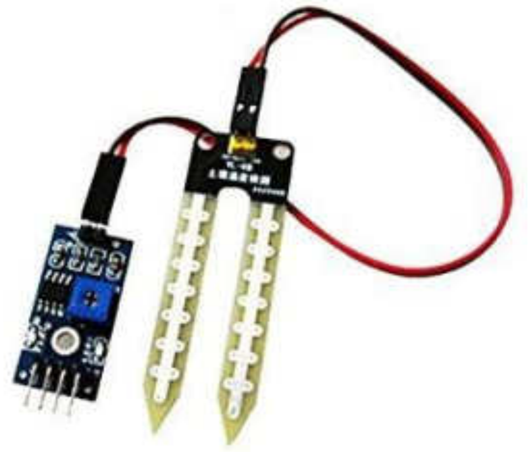
Arduino’s analog pin resolution is 1023 starting from 0. On +5 volts input it counts to 1023. Lm35 has maximum voltage output is 1500mV (At 150 degree centigrade). 1500mV is equal to 1500/1000 = 1.5 volts. So Lm35 at max outputs 1.5 voltage.

Arduino analog pin count for 1.5 volts equals to (1.5 / 5) *1023 = 307.5. At +5 volts it is 1023 and at 1.5 volts its 307.5.

New Arduino-Lm35 Resolution = 307.5 / 150 = 2.048. Now if Arduino analog pin counts 2.048 its equal to 1 degree change in centigrade/Celsius temperature of LM35.

Soil moisture sensor:

It measures the volumetric water content in soil. Since the direct gravimetric measurement of soil moisture requires removing, drying, and weighing of a sample, soil moisture sensors measure the volumetric water content indirectly by using some other property of the soil, such as electrical resistance, dielectric constant, or interaction with neutrons as a proxy for the moisture content.



The relation between the measured property and soil moisture must be calibrated and may vary depending on environmental factors such as soil type, temperature or electric conductivity. Reflected microwave radiation is affected by the soil moisture and is used for remote sensing in hydrology and agriculture. Portable probe instruments can be used by farmers or gardeners.

The sensor reads analog values ranging from 0 to 1023 where 0 corresponds to 100% moisture and 1023 corresponds to 0% moisture.

The equation for moisture percent can be given as:

Humidity % = (1 – (analog value read by sensor/1023)) * 100

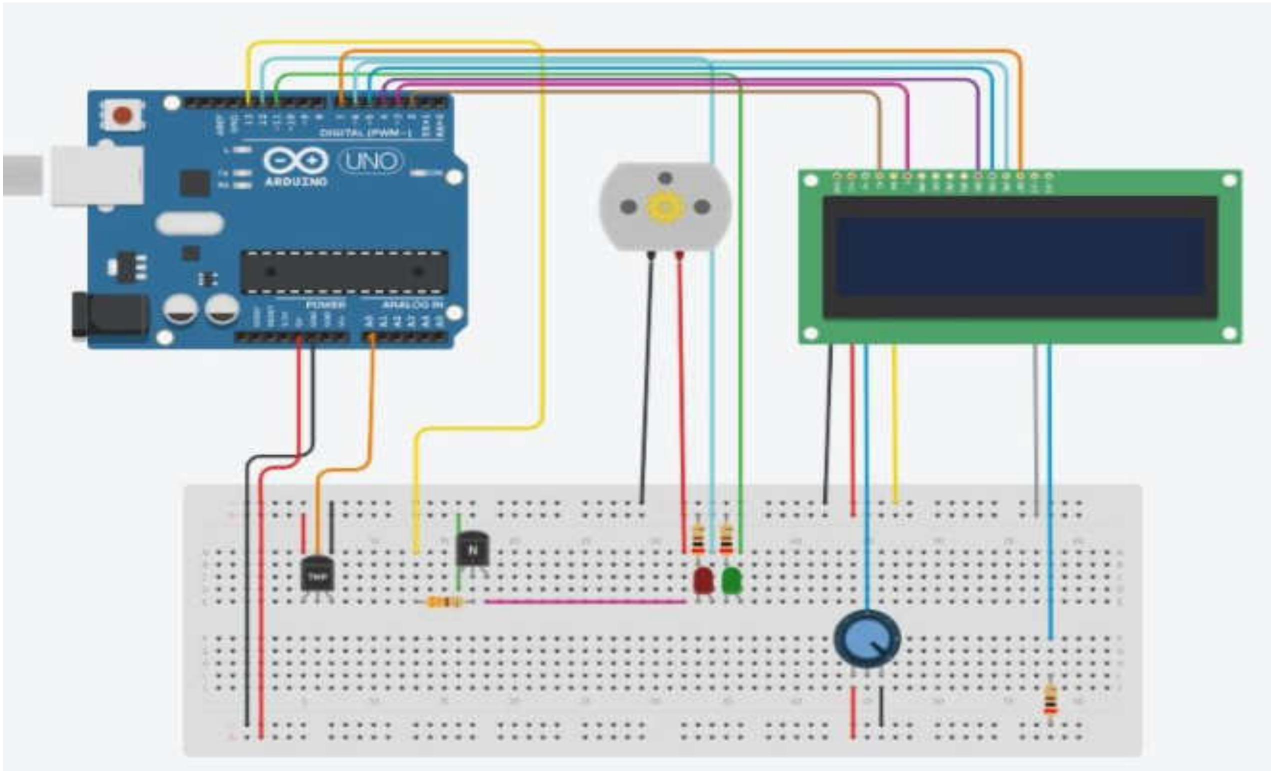
Relay module:

A power relay module is an electrical switch that is operated by an electromagnet. The electromagnet is activated by a separate low-power signal from a micro controller. When activated, the electromagnet pulls to either open or close an electrical circuit.

A simple relay consists of wire coil wrapped around a soft iron core, or solenoid, an iron yoke that delivers a low reluctance path for magnetic flux, a movable iron armature and one or more sets of contacts. The movable armature is hinged to the yoke and linked to one or more set of the moving contacts. Held in place by a spring, the armature leaves a gap in the magnetic circuit when the relay is de-energized. While in this position, one of the two sets of contacts is closed while the other set remains open.

When electrical current is passed through a coil, it generates a magnetic field that in turn activates the armature. This movement of the movable contacts makes or breaks a connection with the fixed contact. When the relay is de-energized, the sets of contacts that were closed, open and breaks the connection and vice versa if the contacts were open. When switching off the current to the coil, the armature is returned, by force, to its relaxed position. This force is usually provided by a spring, but gravity can also be used in certain applications. Most power relays are manufactured to operate in a quick manner.

Tinkercad model:



This here is the tinkercad circuit of our circuit, we weren’t able to add soil moisture sensor in this circuit as it was not available on tinkercad. In the original circuit the soil moisture sensor was connected to the Arduino via “digital pin 10”.

The relay module too wasn’t included as it was added to the circuit as a result of a problem faced in a later stage of the project. In the original circuit the transistor was replaced by relay module due to the same trouble. The relay module was connected to Arduino via “digital pin 13”.

Working Principle of the model:

- We have set the smart irrigation system such that the motor turns ON automatically when the temperature goes beyond a certain limit and the soil moisture falls below a certain limit. Here for demonstration purposes, we have set the temperature limit to 37.5°C and soil moisture limit to 50%.
- As the motor turns ON the red led blows instead of the green led and the lcd screen shows that the motor pump is ON.
- If either the soil moisture rises above the limit or the temperature fall below the limit, the motor turns OFF and the green led blows and the lcd screen shows that the water pump is OFF.

Code for Arduino:

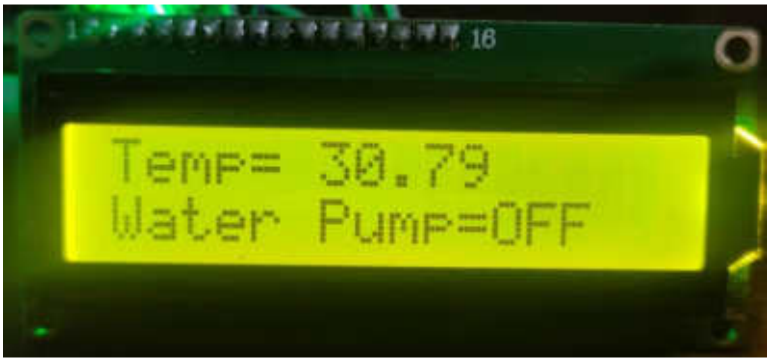
```
#include <LiquidCrystal.h>
#include <Wire.h>



 const int LM35 = A0;
 int relaypin = 13;
 const int LedRed = 12;
 const int LedGreen = 11;
 int water;

 LiquidCrystal lcd (2, 3, 4, 5, 6, 7);

 void setup() {
 Serial.begin(9600);
 lcd.begin(16, 2);
 lcd.print("WELCOME TO");
 lcd.setCursor(0,1);
 lcd.print("SMART IRRIGATION");
 pinMode (relaypin, OUTPUT);
 pinMode(LedRed, OUTPUT);
 pinMode (LedGreen, OUTPUT);
 delay (3500);
 lcd.clear();
 Wire.setClock(10000);
 lcd.print("Temp= ");
 lcd.setCursor(0,1);
 lcd.print ("Water Pump= ");
 }
```

```
void loop() {
water = digitalRead(10);
int analogReading = analogRead(A0);
float Temperature = analogReading * (5.0 / 1023.0) * 100;
lcd.setCursor(6,0);
lcd.print (Temperature);
lcd.setCursor(11,1);



if (Temperature > 37.5 && water == HIGH)
{
digitalWrite (relaypin, LOW);
digitalWrite (LedRed, LOW);
digitalWrite (LedGreen, HIGH);
lcd.print("ON ");
}
else
{
digitalWrite (relaypin, HIGH);
digitalWrite (LedRed, HIGH);
digitalWrite (LedGreen, LOW);
lcd.print("OFF");
}
delay(400);
}
```

In the codes we have done the coding such that:

- The LCD screen reads “WELCOME TO SMART IRRIGATION” displaying it for 3.5 seconds.
- The LCD screen then reads “TEMP = x” and “WATER PUMP = y” where “x” is the temperature and “y” is the status of the water pump.
- The temperature is converted from analog value to digital value.
- When the temperature goes above 37.5 and soil moisture content goes below 50% (In the codes we have given “HIGH” and “LOW” for soil water content as they correspond to soil moisture values of 0-50% and 51-100% respectively) the water pump turns ON, red LED turns ON, green LED turns OFF and the LCD screen shows “WATER PUMP = ON”.
- When the above conditions aren’t obeyed then the water pump turns OFF, the red LED turns OFF, green LED turns ON and LCD screen shows “WATER PUMP = OFF”.

Pictures of LCD Screen:



Software used:

Tinkercad: Tinkercad is a free, easy-to-use web app that equips the next generation of designers and engineers with the foundational skills for innovation: 3D design, electronics, and coding.

Arduino IDE: The open-source Arduino Software (IDE) makes it easy to write code and upload it to the board. This software can be used with any Arduino board.

YouTube Link:

https://youtu.be/njawjRvukgc

Results and Discussion:

We were able to get the expected results.

Observation Table:

Temperature	Moisture	Water Pump	Red LED	Green LED
>37.5	LOW	ON	ON	OFF
>37.5	HIGH	OFF	OFF	ON
<37.5	HIGH	OFF	OFF	ON
<37.5	LOW	OFF	OFF	ON

Problems Faced and Solutions:

We faced problems while making project and we were able to solve those problems by careful observations and proper teamwork.

Problems and solutions were:

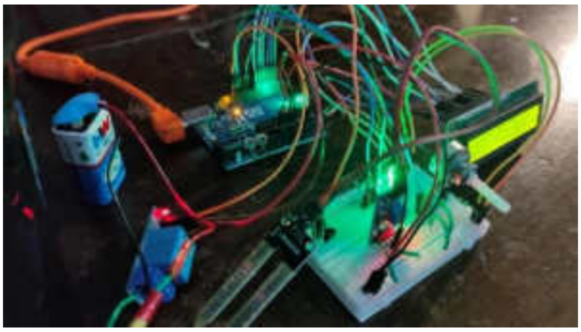
- We weren't able to show total circuit stimulation in tinkercad as all components are not available in that platform.
- Our circuit is adjusted for a specific crop, so our system will not work for multiple crops.
- At first the temperature displayed on the lcd screen was too high, then we came to learn that the lm35 temperature sensor read analog values and we have to multiply the analog value by 0.4887 to get temperature in Celsius scale.
- When we used voltage for the pump from the Arduino itself and the lcd didn't work properly. Then we had to use an external 9V source for the pump and the lcd worked fine.
- The soil moisture sensor didn't work properly at first and then we found errors in code and fixed it.

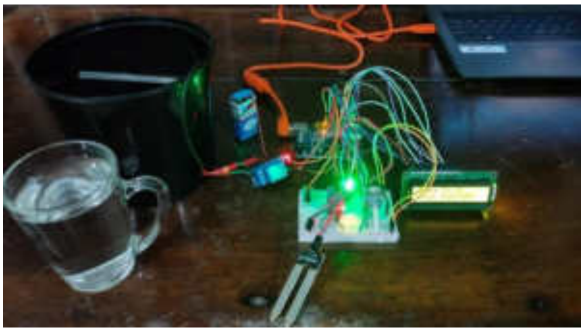
Conditions for some major crops:

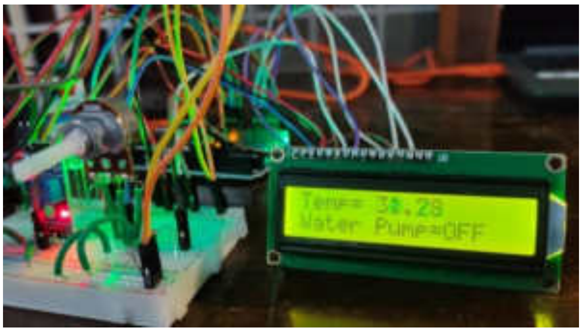
We did some research and gathered the environmental conditions required for different crops.

Sr. No.	Crop Name	Temperature	Moisture (%)	Humidity (%)
1	Rice	21-37	20-25	60-80
2	Wheat	10-15	14-20	60-70
3	Bajra	20-38	20-25	55-70

Some Pictures of the Working Model:







Conclusion:

First of all, we would like to express our sincere gratitude towards Professor Manjunatha for his guidance and support, as well as giving us the opportunity to work on this project.

The project is completed successfully. We were able to make the system as simple as possible. The one we made was a demonstration model of Smart Irrigation System.

Things we could have done better:

- We could have used soil for soil moisture sensor instead of a cup of water for dipping. This made the use of soil moisture sensor a bit unclear in the demonstration.
- We could have made the system to work for multiple crops. In the project we used a random condition of temperature and soil moisture for demonstration purposes.
- We could have added the humidity percentage in the LCD screen too.

Roles of team members:

Aswin D Menon, Allwin Johnjo Prince: collection of components and preparation of manual model

Anish Behuray: Tinkercad simulation preparation and creating codes

Addepalli Manjusri: Preparation of slides and presentation.

References:

DIY Lab IIT Kharagpur YouTube Channel

https://www.electronicsforu.com/electronics-projects/prototypes/smart-irrigation-system

https://randomnerdtutorials.com/guide-for-relay-module-with-arduino/

https://docs.arduino.cc/learn/electronics/lcd-displays

https://create.arduino.cc/projecthub/infoelectorials/project-003-arduino-lm35-temperature-sensor-project-0a43ba

https://create.arduino.cc/projecthub/MisterBotBreak/how-to-use-a-soil-moisture-sensor-ce769b