IoT-based sensing and data acquisition for Agriculture and Environmental applications

INTRODUCTION

The Internet of Things (IoT) has emerged as a megatrend for next-generation technology with far-reaching implications: advanced networking of end machines, devices, and services. Smart health care, smart cities, defense, shopping, traffic congestion, industrial control, and agriculture are only a few of the applications where IoT can help. There has been a lot of research on IoT technology in the agricultural field to improve smart farming solutions. We can gather data from sensing instruments and send it to the main servers thanks to efforts made on wireless sensor networks.

Also another aspect of farming can be the different moisture requirement, temperature and humidity of the surroundings by different crops. Some crops require less amount of soil moisture and some require more. The temperature and humidity of the surroundings also matter to different crop patterns.

In soil research, earth water study and agricultural sciences, water content contributes vitally in: roundwater renewal, agronomy, and Soil chemistry.

NECESSARY REQUIREMENTS OF THE PROPOSED SYSTEM

A. Arduino board

Arduino is an open-source electronic platform based on the ATmega microcontroller. Arduino software is compatible with Windows, Macintosh, and Linux operating systems, which are based on the C programming language and can be expanded through C++ libraries [15]. The dependent microcontroller in this paper is shown in Fig. 1. All other hardware components in this system are connected to Arduino in which receives the signal from sensors and sends/receives SMS to/from mobile through GSM Shield.

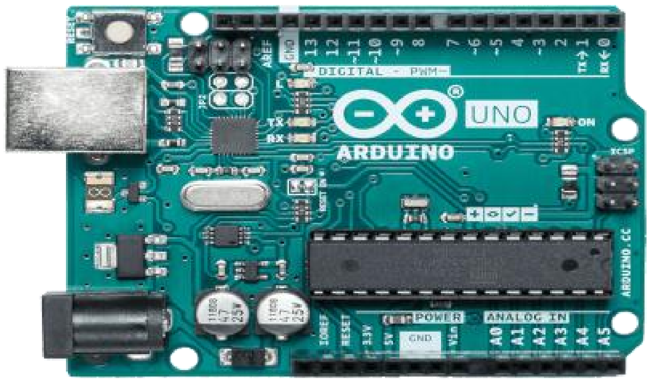
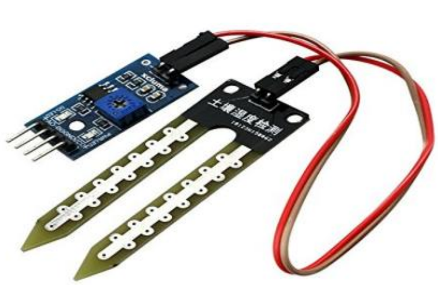


Fig. 1. Arduino UNO R3

B. Soil Sensor

Soil Moisture Sensor, as shown in Fig. 2, is one of the analog sensors. It is an easy-to-use sensor, and it works on two copper rods that increase the contact between them by increasing the water in the soil. It is suitable for monitoring soil humidity in plants and controlling the amount of water for irrigation [17]. In this system, this sensor is connected to Arduino and inserted in the soil to send a signal to Arduino when the soil is wet or dry to start/stop irrigation.



C. Jumper Wire

Jumper wires are extremely handy components to have on hand, especially when prototyping. Jumper wires are simply wires that have connector pins at each end, allowing them to be used to connect two points to each other without soldering. Jumper wires are typically used with breadboards and other prototyping tools in order to make it easy to change a circuit as needed. Fairly simple. In fact, it doesn't get much more basic than jumper wires.

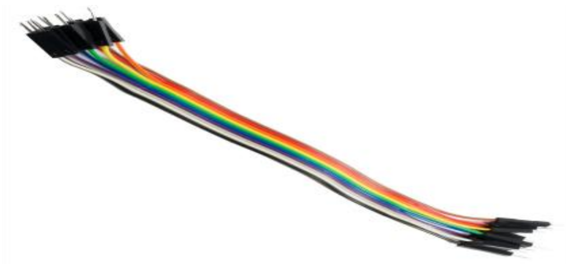


Fig.3. Jumper Wire

System Analysis And Design

The IoT data acquisition and irrigation system is divided into two parts: (i) the hardware and (ii) the software.

A. The Hardware Design

The physical design of the system includes all the hardware required for the system's successful execution. The system's circuit architecture, as well as how the components are connected to ensure the system's operation.

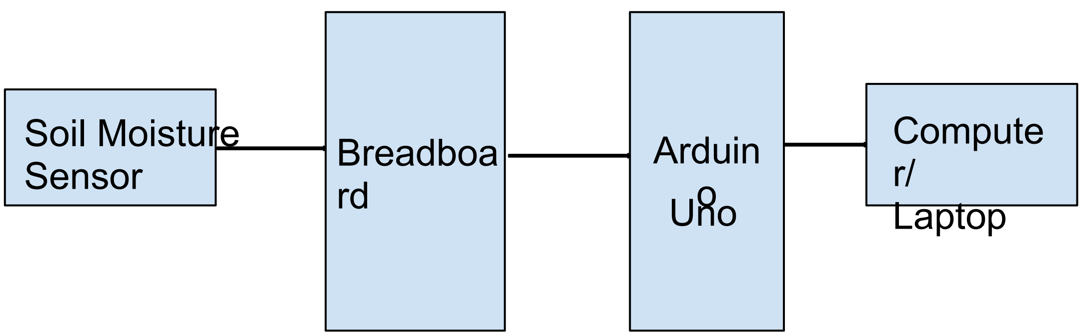


Fig. 4. Hardware Block Diagram

B. The Hardware Specifications

One major sensors are used, a soil moisture sensor, to interface with the Arduino Uno and link with the device.

S/N	HARDWARE	COMPONENTS AND SPECIFICATION
1.	Power Supply	Voltage Regulator 5V
2.	Sensing	FC-28 Soil Moisture Sensor Arduino UNO R3
3.	Control Unit	Arduino UNO R3 microcontroller

C. Circuit Design

The circuit diagram below shows the necessary connections from the Arduino to the FC-28 Soil Moisture Sensor.

The Soil Moisture sensor is connected to the Arduino board with the help of Jumper wires. The sensor contains a set of pins on two sides. The set of pins on one side is Ground(Gnd), #Voltage input(Vcc), Digital output(D0), and Analog output(A0).

These pins are connected to the Arduino chip in the following manner. The Ground pin of the Sensor is connected to the Ground pin of the #Arduino chip. The Vcc pin on the sensor is connected to the 5V supply pin of the Arduino chip.

When there is a requirement for an Analog output, the A0 pin of the sensor is connected to the A0 pin of the Arduino chip. The range of output values for the Analog output is 0 - 1023. Else, when a Digital output is required, the D0 pin of the sensor can be connected to any of the digital output pins of the Arduino chip. The digital outputs, as we know, will be either 0 or 1.

The second set of pins present on the other side of the sensor are connected to the probes that will be measuring the moisture of the soil when placed physically in it. The conducting probes are the basic sensor devices of this circuit.

As mentioned earlier, the sensor has a potentiometer and a comparator embedded on it whose functions are to hold the threshold value and compare the input value from the sensor to the threshold value fixed by the user, respectively. The detailed working will be discussed in the working of the circuit.

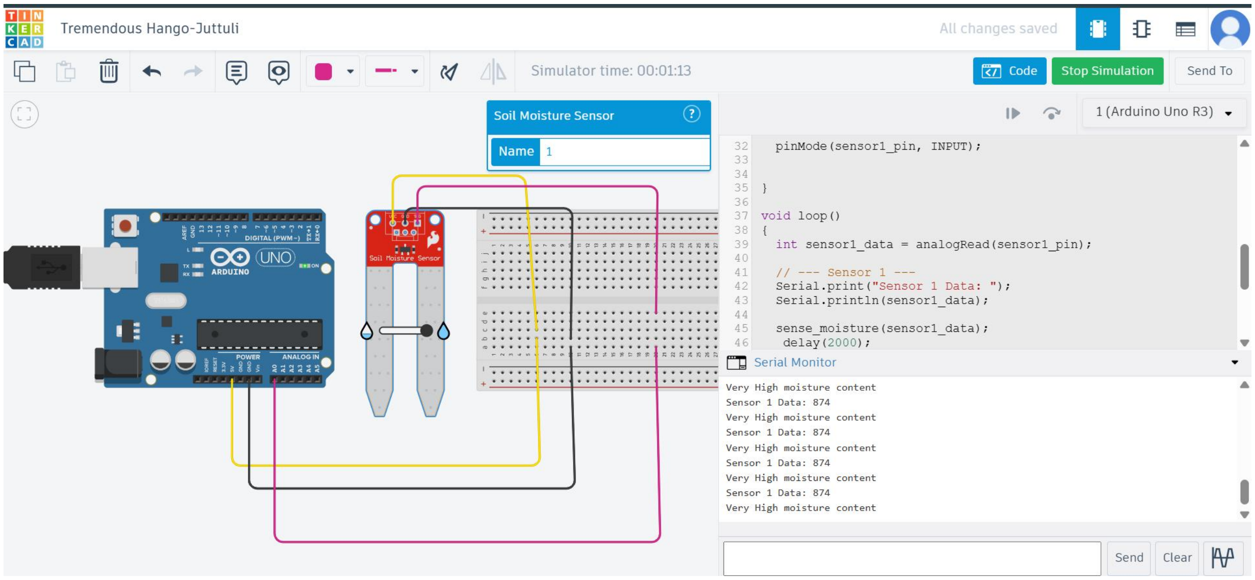


Fig.5. One sensor joins with Arduino UNO R3

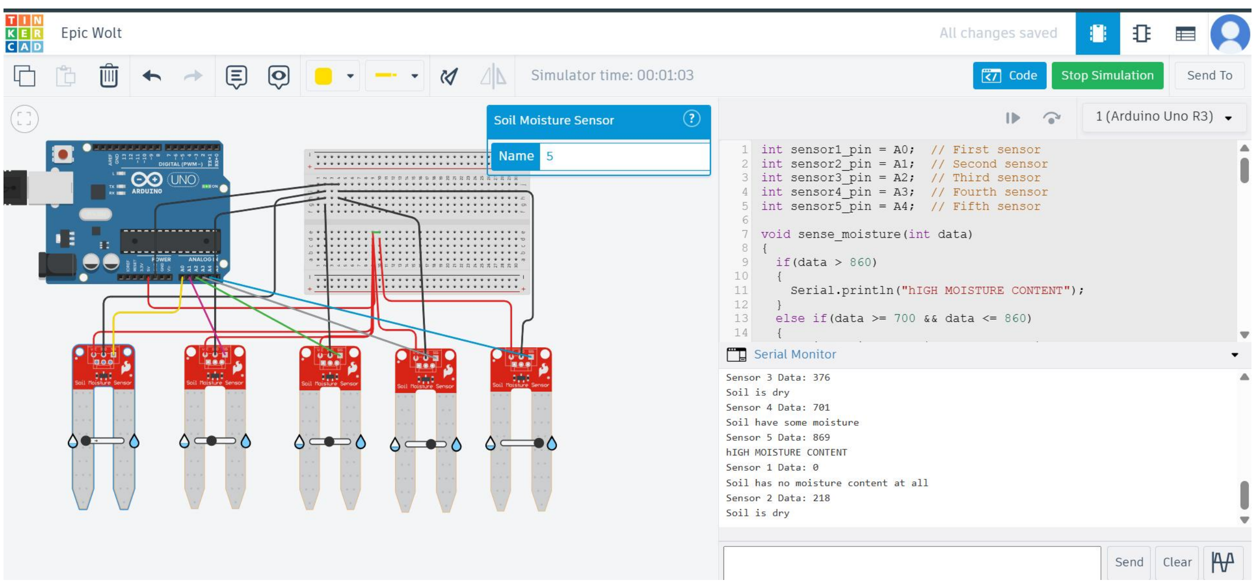


Fig.6. Multiple sensors working together.

D. Code used for one sensor

```
int sensor_pin = A0;

void setup()

{

  Serial.begin(9600);
```

```
  pinMode(sensor_pin, INPUT);
}
void loop()
{
  int sensor_data = analogRead(sensor_pin);
  Serial.print("Sensor_data:");
  Serial.print(sensor_data);
  Serial.print("\t | ");
   if(sensor_data > 950)
  {
    Serial.println("No moisture, Soil is dry");
  }
  else if(sensor_data >= 400 && sensor_data <= 950)
  {
    Serial.println("There is some moisture, Soil is medium");
  }
  else if(sensor_data < 400)
  {
    Serial.println("Soil is wet");
  }
 delay(100);
}
```

E. Code used for multiple sensors

```
int sensor1_pin = A2;  // First sensor
int sensor2_pin = A0;  // Second sensor
int sensor3_pin = A3;  //Third Sensor
```

```
int sensor4_pin = A4; //Fourth Sensor

int sensor5_pin = A5; //fifth sensor

void sense_moisture(int data)

{

  if(data > 950)

  {

    Serial.println("No moisture, Soil is dry");

  }

  else if(data >= 700 && data <= 950)

  {

    Serial.println("There is little moisture");

  }

  else if(data >= 400 && data <= 700)

  {

    Serial.println("Soil have some moisture");

  }

  else if(data >=200 && data <= 400)

  {

    Serial.println("Soil is wet");

  }

  else if(data <200)

  {

    Serial.println("Soil has very high moisture content");

  }

return 0;

}
```

```
void setup()
{
  Serial.begin(9600);
  pinMode(sensor1_pin, INPUT);
  pinMode(sensor2_pin, INPUT);
  pinMode(sensor3_pin, INPUT);
  pinMode(sensor4_pin, INPUT);
   pinMode(sensor5_pin, INPUT);
}
void loop()
{
  int sensor1_data = analogRead(sensor1_pin);
  int sensor2_data = analogRead(sensor2_pin);
  int sensor3_data = analogRead(sensor3_pin);
  int sensor4_data = analogRead(sensor4_pin);
  int sensor5_data = analogRead(sensor5_pin);
  // --- Sensor 1 —
 Serial.print("Sensor 1 Data: ");
  Serial.print(sensor1_data);
  Serial.print(" \t | ");
  sense_moisture(sensor1_data);
   delay(2000);
// --- Sensor 2 ---
  Serial.print("Sensor 2 Data: ");
  Serial.print(sensor2_data);
  Serial.print(" \t | ");
```

```
  sense_moisture(sensor2_data);
   delay(2000);
  // --- Sensor 3 ---
  Serial.print("Sensor 3 Data: ");
  Serial.print(sensor3_data);
  Serial.print(" \t | ");
  sense_moisture(sensor3_data);
 delay(2000);
  // --- Sensor 4---
  Serial.print("Sensor 4 Data: ");
  Serial.print(sensor4_data);
  Serial.print(" \t | ");
   sense_moisture(sensor4_data);
  //Serial.println("There is some moisture, Soil is medium");
  delay(2000);  // Wait 2 seconds before next reading
// --- Sensor 5---
  Serial.print("Sensor 5 Data: ");
  Serial.print(sensor5_data);
  Serial.print(" \t | ");
   sense_moisture(sensor5_data);
  //Serial.println("There is some moisture, Soil is medium");
  delay(2000);  // Wait 2 seconds before next reading}
```

SYSTEM IMPLEMENTATION AND RESULT

The implementation of each component that makes up this system, as well as the overall performance of the system and its components, are all tested. The results of the work done at various stages of the project are presented in this section.

The circuit, when switched ON and OFF, is shown below.

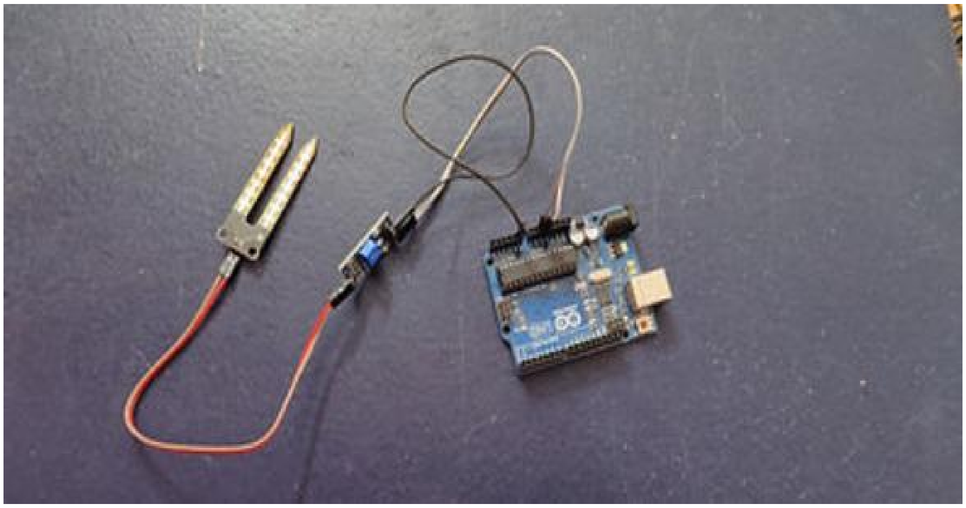


Fig. 7. When the circuit is switched off

When the connections are done, the conducting probes connected to the Soil moisture sensor must be inserted into the soil whose Moisture value is desired to be known. The circuit must simultaneously be interfaced with the Arduino IDE.

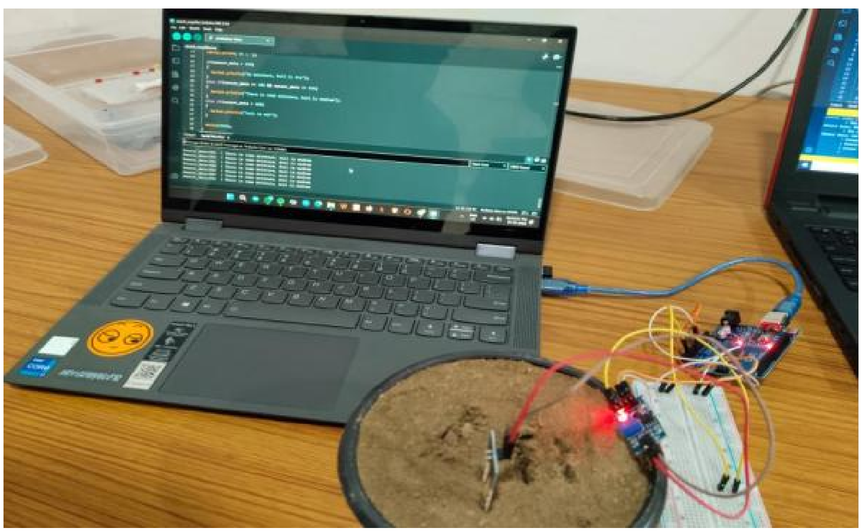


Fig. 8. When the circuit is switched on

When the circuit is switched ON, the sensor reads the values that the probes conduct. The potentiometer actually holds the threshold value fixed by the user. The LM323 comparator present on the sensor chip compares the threshold value to the sensor input value. The glowing of the LED in the sensor indicates the functioning of the sensor.

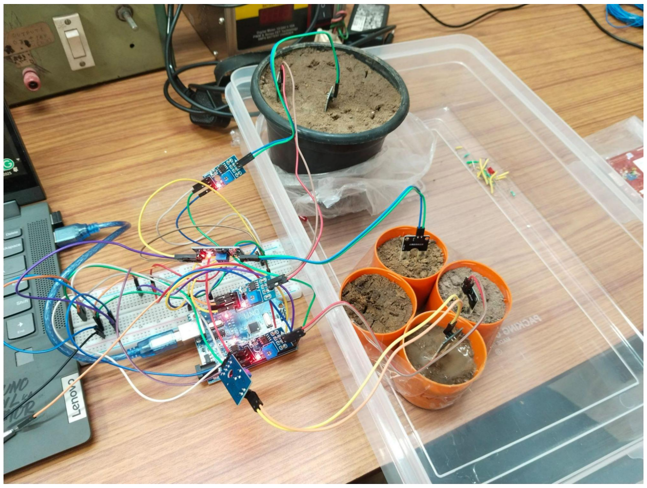


Fig. 9. when the five sensor is working together

Results

The results of this work include the building of a IoT Farm prototype that can acquire data in real time and send it to the cloud for easy access of the user and also to carry out the job of irrigation on a farmland with the use of Internet of Things technology. The major aim of the work was achieved during the construction and design of the work.

Output	Serial Monitor x		
Message (Enter to send message to ‘Arduino Uno’ on COM6)		New Line	9600 baud
Sensor_data:252 Sensor_data:256 Sensor_data:259 Sensor_data:262 Sensor_data:264 Sensor_data:266 Sensor_data:268	Soil is wet Soil is wet Soil is wet Soil is wet Soil is wet Soil is wet Soil is wet		

So this here is the data when we find out that the soil is moist/wet. When one sensor is joined with Arduino UNO.

Fig. 10. graphical representation of soil moisture sensor when single sensor is used.

Output	Serial Monitor x		
Message (Enter to send message to ‘Arduino Uno’ on COM6)		New Line	9600 baud
Sensor 2 Data: 362 Sensor 3 Data: 323 Sensor 4 Data: 0 Sensor 5 Data: 259 Sensor 1 Data: 447 Sensor 2 Data: 832 Sensor 3 Data: 879 Sensor 4 Data: 574 Sensor 5 Data: 904	Soil is wet Soil is wet Soil has very high moisture content Soil is wet Soil have some moisture There is little moisture There is little moisture Soil have some moisture There is little moisture		

The data representation of soil moisture sensor when we use 5 sensor together in different kind of soil to check the amount of water present in the soil. The different type of soil we are using is 1. Soil which is fully dry, 2. Soil having some moisture. 3. Soil having little moisture. 4. Soil has very high moisture content. 5. soil is fully wet.

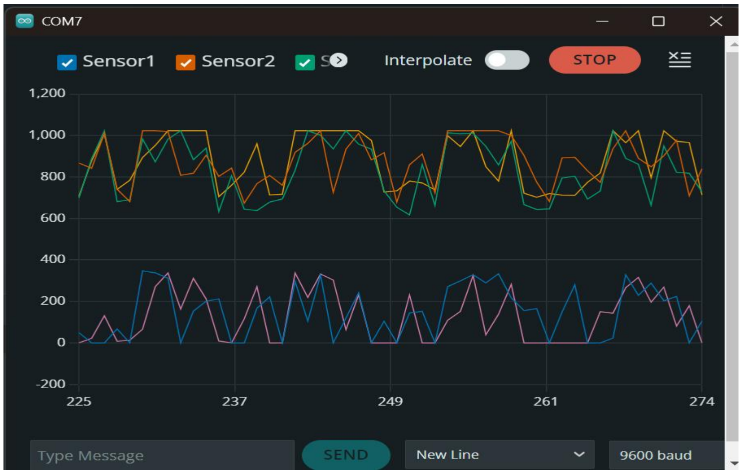


Fig. 10. Graphical representation of soil moisture sensor when 5 sensor is working together.

Fig. 11. Graphical representation of soil moisture sensor when we are changing the substance from soil to sand and the sand doesn't contain any impurities that is why the range is between 990 to 1,025.

Fig. 11. Graphical representation of soil moisture sensor when we are changing the substance from soil to sand and no of impurities content in the soil more that why the range is between 1,018.5 to 1,022.5.

CONCLUSION

The IoT data acquisition and automated irrigation system in this work was developed essentially to enable farmers remotely run their farms. The technology helps farmers with irrigation and daily data records of three agricultural parameters (humidity, temperature and soil moisture) on the internet, which the farmer can check on a web portal. The hardware system can be installed on a farm (enclosed in order to avoid environmental damage) or a greenhouse. The Sensors in the system detect agricultural parameters.