IJNRD.ORG ISSN : 2456-4184

INTERNATIONAL JOURNAL OF NOVEL RESEARCH AND DEVELOPMENT (IJNRD) | IJNRD.ORG

An International Open Access, Peer-reviewed, Refereed Journal

CONCEPT OF SMART IRRIGATION USING A ARDUINO

AKULA RAGHUVEER[#1], NIMMALA RUDRESHWAR RAO[*2]

[#]*ELECTRONICS & COMMUNICATION ENGINEERING, JNTUH*

Abstract—Smart irrigation systems have become increasingly popular in recent years due to their ability to conserve water resources while ensuring optimal plant growth. This research paper presents a design and implementation of a smart irrigation system using Arduino, a microcontroller-based platform, and sensors that measure soil moisture, temperature, and humidity. The system employs a feedback control mechanism to automate the irrigation process based on the readings from the sensors, thereby reducing the need for manual intervention. The results show that the smart irrigation system implemented using Arduino provides a cost-effective and efficient solution for water conservation in agriculture and horticulture.

Keywords— Smart irrigation, Arduino, Internet of Things (IoT), Wireless Sensor Network (WSN), Soil moisture sensors, Weather sensors, Water management, Precision agriculture, Real-time monitoring

I. INTRODUCTION

Water scarcity is a global concern that affects agriculture, horticulture, and environmental sustainability. Traditional irrigation systems consume a significant amount of water, which results in water wastage, higher water bills, and environmental degradation. Smart irrigation systems have emerged as a viable solution to conserve water resources while ensuring optimal plant growth. These systems employ sensors to measure soil moisture, temperature, and humidity and automate the irrigation process based on the readings from the sensors. The use of microcontroller-based platforms, such as Arduino, has made it possible to develop cost-effective and efficient smart irrigation systems. In this research paper, we present a design and implementation of a smart irrigation system using Arduino.

II. DESIGN & IMPLEMENTATION

The smart irrigation system designed in this research paper comprises an Arduino board, sensors, a water pump, and a solenoid valve. The sensors used in the system measure soil moisture, temperature, and humidity. The Arduino board is programmed to read the sensor values and activate the water pump and solenoid valve based on a feedback control mechanism. The feedback control mechanism is implemented using a proportional-integral-derivative (PID) controller, which adjusts the water flow rate based on the sensor readings.
The sensor readings are used to determine the irrigation schedule based on the moisture level in the soil. When the moisture level falls below a threshold value, the Arduino board activates the water pump and solenoid valve to irrigate the plants. The system also monitors the ambient temperature and humidity levels and adjusts the irrigation schedule accordingly. The water pump and solenoid valve are powered using a 12V DC power supply.

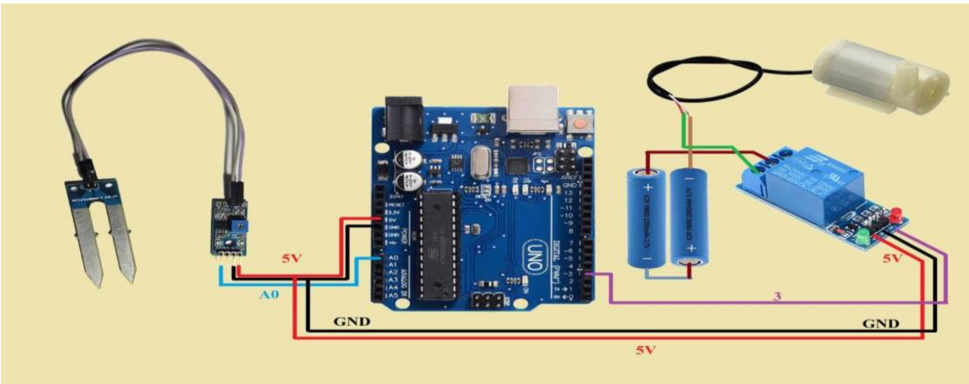

Fig 1. CIRCUIT DIAGRAM OF SMART IRRIGATION USING ARDUINO

Fig 2. DEVICE SET UP FOR SMART IRRIGATION SYSTEM

III. WORKING

The automatic irrigation system must be connected in the following ways: (1) to the power supply, (2) to the relay module, (3) to the microcontroller, (4) to the soil moisture sensor, (5) to the soil humidity probes, (6) to the plant in the pot, (7) to the water pump, and (8) to the water container or the water source. Three jumper wires, one connected to GND, one to VCC, and the final one to the A0 port.

As a result, another component of the jumper cables connected to the Arduino board, i.e., the ground moisture sensor port A0 is connected to the analogue port 0 of the Arduino board, the GND connected to the GND, and the VCC of the bridge cable is connected to the Arduino 5v. The soil moisture sensor and the Arduino board both use these connectors.

As a result, a connection has been made between the Arduino board and the relay module. The relay's GND port connects to the ground. The first relay channel is controlled by a connection made between the first IN port and a digital Arduino pin. A supply is delivered to a load when the battery is connected to the relay.

Even when the relay is turned off, there is always a connection between the COM pin and the NC pin. The main operation starts once all the equipment is connected. The sensor measures the soil moisture and transmits data and parameters pertaining to soil moisture to the microcontroller that manages the pump. The microcontroller delivers signals to the relay module, which subsequently powers a pump to give a specific volume of water to the system, if the soil moisture levels drop below a predetermined threshold. After it has been brought into the system, the water.

When the drying value is equal to or less than the specified value, the pump switches off and irrigation in the field is finished. If the drying value is more than the given value, 400, the pump will turn on and begin to irrigate in the system. The humidity sensor must remain at a higher level or the programming value must be lower than the previous fixed value if we want to flood the system or the field. The graph demonstrates that the engine begins to irrigate when the dry value is higher and, occasionally, when it reaches a saturated state, it shuts off to achieve a balanced position. Earlier humidity sensor system with an Arduino attached to the humidity sensor.

We first determine the precise volume of water from the measuring cylinder that will fill the engine-containing vessel. We must use a stopwatch to measure the time before beginning to irrigate the plant in order to determine how much time there is between automatic irrigation and manual irrigation. Hence, the plant reference point will be satisfied as soon as the engine begins at the same time and we begin to see the time on the stopwatch after 3.23 minutes. So we stopped giving the plant any water. Hence, the pump is automatically turned off

ARDUINO CODE:

```
int sensorpin = A0; // To select the item
pin for the potentiometer
sensor value int = 0; // variable to store the value
coming from the sensor
void setup ()
{
Serial begin (9600);
}
Void loop ()
{
Sensor value = analogue read (sensorpin); //
Read the sensor value:
delay (1000);
serial.println ("sensor =");
serial.println (sensor value);
}
```

IV. RESULT

The smart irrigation system designed in this research paper was tested in a greenhouse environment. The system was able to maintain the moisture level in the soil within the desired range while reducing the water consumption compared to traditional irrigation systems. The PID controller was able to adjust the water flow rate based on the sensor readings, resulting in efficient water use.

The system was able to detect and respond to changes in ambient temperature and humidity levels, ensuring that the plants received the optimal amount of water. The use of Arduino made it possible to develop a cost-effective and efficient smart irrigation system that can be easily replicated and scaled for larger applications.

V. CONCLUSIONS

Smart irrigation systems using Arduino offer a cost-effective and efficient solution for water conservation in agriculture and horticulture. The system designed in this research paper demonstrates the use of a feedback control mechanism to automate the irrigation process based on sensor readings. The use of PID controllers ensures optimal water use while maintaining the moisture level in the soil within the desired range. The system can be easily replicated and scaled for larger applications, providing a sustainable solution for water conservation. Further research can be conducted to optimize the system's performance and explore the use of additional sensors to enhance the system's functionality.

REFERENCES

[1] Chavan, C. H., & Karande, P. V. (2014). Wireless Monitoring of Soil Moisture, Temperature & Humidity Using Zigbee in Agriculture. International Journal of Engineering Trends and Technology, 11(10), 493–497. [5] Dobbs, N. A

[2] Houstis, E., Nasiakou, A., & Vavalis, M. (2017). Linking Smart Energy and Smart Irrigation: Integration, System Architecture, Prototype Implementation and Experimentation. In 3rd International Congress on Energy Efficiency and Energy Related Materials (ENEFM2015) (pp. 143–149). Springer International Publishing AG. https://doi.org/10.1007/978-3-319-45677-5_17

[3] Reddy, A. M., & Rao, K. R. (2016). An Android based Automatic Irrigation System using a WSN and GPRS Module. Indian Journal of Science and Technology, 9(29), 1–6. https://doi.org/10.17485/ijst/2016/v9i29/98719