



# **A Smart Irrigation Water Application System Based on Arduino Platform**

**Oladipo, I. O. <sup>a</sup> and Ajewole, P. O. <sup>a\*</sup>**

<sup>a</sup> *Department of Agricultural and Bio-Environmental Engineering, The Federal Polytechnic, Ado-Ekiti, Nigeria.*

## **Authors' contributions**

*This work was carried out in collaboration between both authors. Author OIO as the lead author supervises the material procurement and experimental setup. Author APO did the data collection and the compilation of manuscript. Both authors read and approved the final manuscript.*

## **Article Information**

DOI: <https://doi.org/10.9734/ajahr/2024/v11i2320>

### **Open Peer Review History:**

This journal follows the Advanced Open Peer Review policy. Identity of the Reviewers, Editor(s) and additional Reviewers, peer review comments, different versions of the manuscript, comments of the editors, etc are available here: <https://www.sdiarticle5.com/review-history/116393>

**Original Research Article**

**Received: 01/03/2024**

**Accepted: 08/05/2024**

**Published: 13/05/2024**

## **ABSTRACT**

Water wastage, erosion, and crop water oversaturation has always been a problem associated with irrigation practice in Nigeria. A smart irrigation system based on Arduino platform was developed in this research to address this problem. The objective of the system is to automate water application to crops and at the same time optimize the use of water resources. The system consists of an Arduino module, data logger, LCD display, water pumps, water storage, moisture sensor, USB connector, and battery. A C++ program was written for the Arduino board so sense the moisture content of the soil, determine the wetness status and power the pumps for water application. The results of testing the system shows that the optimal use of irrigation water by the smart irrigation system was 97% more efficient than the traditional system and it can conveniently do irrigation at the rate of 0.063m<sup>3</sup>/ha. The system is recommended for large scale use especially for dry season farming.

**Keywords:** *Irrigation; arduino; water; automation; smart system.*

\*Corresponding author: Email: [peterajewole@gmail.com](mailto:peterajewole@gmail.com);

**Cite as:** I. O., O., & P. O., A. (2024). A Smart Irrigation Water Application System Based on Arduino Platform. *Asian Journal of Agricultural and Horticultural Research*, 11(2), 132–140. <https://doi.org/10.9734/ajahr/2024/v11i2320>

## **1. INTRODUCTION**

The importance of agriculture lies in its capacity to ensure food security and fulfill the nutritional requirements of growing populations. Through the cultivation of crops and the raising of livestock, agricultural practices guarantee a consistent supply of essential sustenance for individuals and communities. Throughout history, the cultivation of staple crops such as grains, vegetables, and fruits, alongside animal husbandry for meat, dairy, and fiber, has been crucial for human survival and well-being [1]. Crop cultivation typically occurs in regions with sufficient soil water availability. However, in arid areas and during dry seasons, inadequate soil water leads to crop withering. To enable year-round farming and maximize arid land utilization, the development of irrigation systems is essential [2].

Irrigation, involving the artificial application of water to soil, has been adopted to support crop cultivation, landscape maintenance, and soil restoration in regions with arid climates or insufficient rainfall. It enhances the supply of moisture and nutrients to plant roots, lowers surface temperatures, and increases humidity levels [3]. In the northern regions, rainfall is limited to the period between June and September, ranging from 500 mm to 750 mm. The remaining months experience hot and dry conditions, marked by high annual variation in rainfall that results in floods and droughts [4]. Consequently, farmers turn to irrigation to meet the escalating demands for vegetables, fruits, and food crops in response to a rapidly growing population. The northern region of Nigeria plays a pivotal role in food supply and income generation, providing vegetables and fruits like tomatoes, okra, onions, and peppers to the southern part of the country. However, irrigation practices in Nigeria predominantly rely on manual methods such as watering cans, manually operated water channels, and backpack sprinklers, leading to considerable water wastage during [5,6]. Automated agricultural irrigation systems hold potential for accurate soil moisture management.

Conventional irrigation techniques, such as overhead sprinklers and flood-based systems, often saturate the lower portions of plants, including leaves and stems. This prolonged wetness creates a conducive environment for fungal infections. In contrast, modern drip or

trickle irrigation involves precise and slow water application to specific parts of the plant root zone, preventing soil surface saturation and minimizing moisture stress [7]. Automation equipment allows farmers to reduce runoff due to excessive watering and avoid inappropriate irrigation times. This enhances crop performance by ensuring timely provision of water and nutrients [8,9].

Given the challenges posed by climate change and population growth, the integration of electronic technologies into agriculture is essential for increased industry efficiency. Precision agriculture, enabled by electronic automation modules and sensor networks, offers a solution for resource optimization. Among its applications, irrigation systems play a crucial role due to their impact on farming efficiency. Recent advancements in control systems have led to the incorporation of informatics systems, microcontrollers, and new sensor transmission technologies in agriculture. These advancements contribute to more sophisticated and efficient farming practices [10,11].

The emergence of precision irrigation technology introduces a novel approach to water resource management, tailored to the unique needs of soil and plants. Traditional irrigation methods can lead to water overuse or inadequate supply. To ensure consistent and appropriate hydration, automated plant irrigation systems become imperative to regulate water intake based on optimal soil moisture levels specific to each plant type [12,13]. In this study, a smart irrigation system using an Arduino microcontroller was developed, constructed, and evaluated to implement precision agriculture.

## **2. MATERIALS AND METHODS**

### **2.1 The System Components**

The smart irrigation system consists of the following components:

#### **2.1.1 Arduino module**

The Arduino module is a microcontroller that reads the input to the sensor and converts it to output displayed on the LCD. The irrigation system is controlled by the Arduino by turning on or off the water. This is done by sending a set of instructions to the control unit on the Arduino board. The Arduino module is shown in Fig 1.



**Fig. 1. Arduino module**

### 2.1.2 Arduino LCD Display

The LCD (Liquid Crystal Display) is a type of display that uses the liquid crystals for its operation as shown in Fig 2. The moisture content, temperature and humidity are displayed on the LCD.

### 2.1.3 The Arduino data logger

The Arduino data logger is connected to the Arduino and it contains an SD card for data storage as shown in Fig 3. The results of the moisture content and time of turning on or off the pump are recorded and stored in the memory card. This can be extracted for further analysis.

### 2.1.4 Water pump

The water pump is shown in Fig 4 and it is used in this project to pump the water needed for irrigation from the main water tank through pipes.



**Fig. 2. LCD display**

### 2.1.5 The Soil Sensor Unit

The Soil Moisture Sensor (Fig 5) is used to measure the volumetric water content of soil. The YL-69 module was used in the design. It uses

capacitance to measure the water content of soil (by measuring the dielectric permittivity of the soil, which is a function of the water content). It is simply inserted into the soil to be tested, and the volumetric water content of the soil is displayed in percentage.

### 2.1.6 Relays

Relays are electrical switches used to establish or break an electrical connection. 12 V relays are used in the automated irrigation system to connect and control the power supply to the irrigation pipeline valves as shown in Figs 5 and 7.



**Fig. 3. Data logger module**



**Fig. 4. Water pump**

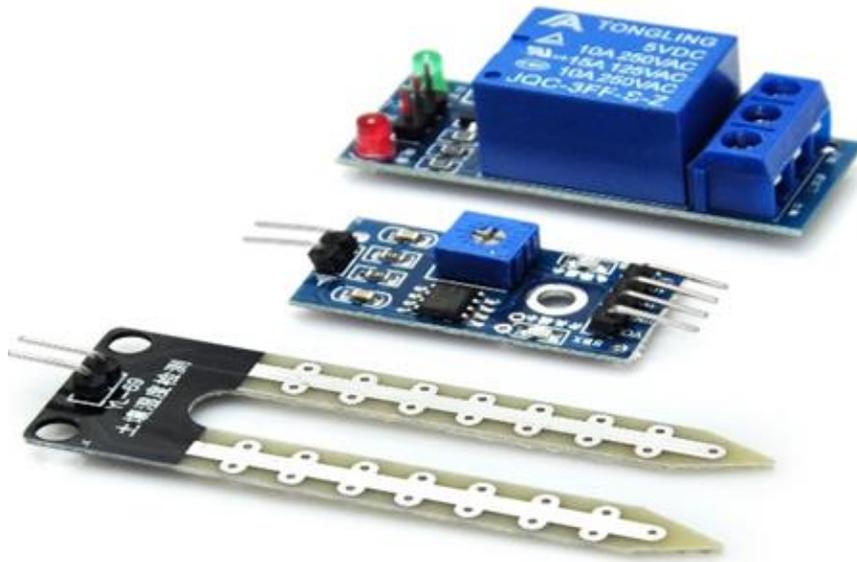


Fig. 5. Sensor and relay module



Fig. 6. Jumper wires

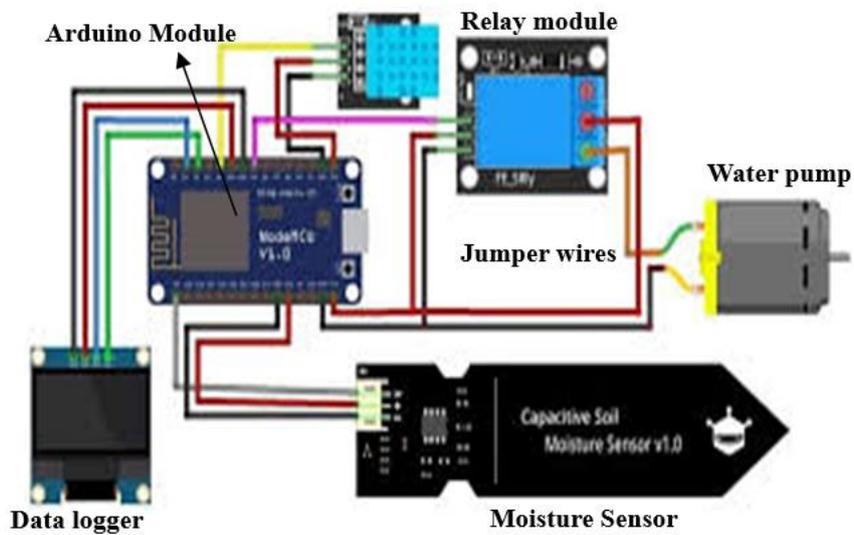


Fig. 7. Electronic circuit diagram of the system



**Fig. 8. Experimental set**



**Fig. 9. Testing of the system**

### **2.1.7 Jumper wires**

Jump wire (Fig 6) was used to interconnect the components of the circuit, internally or with other equipment or components, without soldering.

### **2.1.8 Battery**

The system is powered by a 12v lithium battery. The battery supplies power to the Arduino board, the relay and the data logger.

## **2.2 Mode of Operation**

Fig. 7 shows the electronic circuit and connection of the components of the irrigation system. A

program written in C++ was uploaded to a memory card on the Arduino module. The program listing is shown in the appendix. The moisture sensors detect the moisture content of the soil through the soil electrical conductance property and sends signal to the Arduino module in order to take the hydrometric measurement of free-soil wetness. It will also detect the temperature and humidity from the environment and show this information in the Arduino display. When the wetness is below 30%, the Arduino module sends signals to the water pumps to supply water to the plants.

**Table 1. The results obtained from the test of the irrigation system**

S/N	Pot 1			Pot 2			Pot 3			Pot 4		
	MC (%)	Time (s)	AR (cm <sup>3</sup> /s)	MC (%)	Time (s)	AR (cm <sup>3</sup> /s)	MC (%)	Time (s)	AR (cm <sup>3</sup> /s)	MC (%)	Time (s)	AR (cm <sup>3</sup> /s)
1	81	2	25.50	85	3	18.33	77	3	15.67	72	4	10.50
2	74	2	22.00	67	3	12.33	90	3	20.00	66	6	6.00
3	82	3	17.33	70	4	10.00	90	3	20.00	68	3	12.67
4	66	4	9.00	74	5	8.80	90	2	30.00	72	3	14.00
5	75	3	15.00	81	2	25.50	85	2	27.50	88	4	14.50
6	69	2	19.50	76	4	11.50	81	4	12.75	69	2	19.50
7	80	3	16.67	72	3	14.00	76	2	23.00	73	2	21.50
8	90	3	20.00	69	4	9.75	81	2	25.50	81	2	25.50
9	90	2	30.00	75	3	15.00	76	3	15.33	66	3	12.00
10	70	2	20.00	85	5	11.00	66	2	18.00	89	3	19.67
<b>Average</b>	77.70	2.60	19.50	75.40	3.60	13.62	81.20	2.60	20.78	74.40	3.20	15.58

## 2.3 Experimental Setup and Testing

The automated irrigation system is tested by supplying water to four pots with each containing cowpea seedlings as shown in Figures 8 and 9. Each pot has a sensor dipped into the pot and also has a water pump connected to it. The water pumps are connected through pipe networks to a water storage. The water requirements for each of the four pots will vary because the saturated level of each pot is different. When the measurement of the moisture sensor in pot 1 drops below the minimum moisture level of 30% wet basis, it sends a signal to pump 1. This signal causes the power supply to the water pump to be altered, switching on the water pump and allowing water from the water storage to flow pot 1 only. When the measurement of the moisture sensor in pot 1 rises above the maximum moisture level 70% wet basis, it sends a signal to pump1, and triggers it to turn off the pump as well as the display unit for the water pump. This signal causes the power supply to the water pump to be altered, switching off the water pump and preventing water from the water storage to flow pot 1 only. This process is repeated for each of the pots at different intervals depending on the moisture content of the soil in the pots.

## 3. RESULTS AND DISCUSSION

### 3.1 Results

The results obtained from the test of the irrigation system is shown in Table 1. The moisture content (MC) in the table is the moisture content registered by the sensor after water application. Results were obtained for ten trials and for each trial the water application rate (AR) was calculated

### 3.2 Discussion

It was observed that pot 3 had the highest water application rate of 20.78cm<sup>3</sup>/s while pot 2 had the lowest value of 13.62cm<sup>3</sup>/s. The average application rate was 17.37cm<sup>3</sup>/s. The difference in the application rate might be due to the difference in the sensitivity of each of the sensors. It was also observed that the pumping of water was not turned off at exactly 70% moisture content programmed into the Arduino. The moisture content detected by the sensors and which was displayed by the LCD varied between 65% and 90%. This might be due to difference in sensitivity of the sensors and the differential flow

of water through the soil in the pot. The average water application rate of the system in cubic meter per hectare (0.063m<sup>3</sup>/ha) was 97% more efficient than the traditional application rate of 2.5m<sup>3</sup>/ha for grain crops [14,15].

## 4. CONCLUSIONS

A smart irrigation system using an Arduino microcontroller was designed, set up and tested in this study. The system can automate the application of irrigation water with minimal human intervention and also optimize water resource utilization for grain crop irrigation. Experimental results indicate that the system is 97% more efficient than the traditional method of irrigation of grain crops. The system is recommended for large scale irrigation project especially in the areas of limited water availability and during dry seasons.

## COMPETING INTERESTS

Authors have declared that no competing interests exist.

## REFERENCES

1. Calicioglu O, Flammini A, Bracco S, Bellù L, Sims R. The Future Challenges of Food and Agriculture: An Integrated Analysis of Trends and Solutions. *Sustainability*. 2019; 11(1):222. Available: <https://doi.org/10.3390/su11010222>
2. Biswas B, Timsina J, Mandal K, Naorem A. Effects of different irrigation methods and mulching on yield, Growth and Water Use Efficiency of Strawberry; 2023. DOI: 10.21203/rs.3.rs-2766264/v1
3. Tyagi S, Sharma B, Singh P, Dobhal R. Water Quality Assessment in Terms of Water Quality Index. *American Journal of Water Resources*. 2020;1(3):34–38. Available: <https://doi.org/10.12691/ajwr-1-3-3>
4. Eruola A, Makinde A, Eruola G, Ayoola K. Assessment of the rainfall exceedance in Nigeria. *Nigerian Journal of Technology* 2021;40(4):751–761. Available: <https://doi.org/10.4314/njt.v40i4.22>
5. Bashir A, Kyung-Sook C. A review of the evaluation of irrigation practice in Nigeria: Past, present and future prospects. *African Journal of Agricultural Research*. 2018; 13(40):2087–2097.

- Available:<https://doi.org/10.5897/ajar2018.13403>
6. Animashaun I, Oguntunde P, Akinwumiju A, Olubanjo O. Rainfall Analysis over the Niger Central Hydrological Area, Nigeria. Variability, Trend, and Change point detection. *Scientific African*. 2020;8:e00419. Available:<https://doi.org/10.1016/j.sciaf.2020.e00419>
  7. Shen Y, Gao Y, Lin J, Ma J, Wang Z, Haapasalo M. Methods and models to study irrigation. *Endodontic Topics*. 2012;27(1), 3–34. Available:<https://doi.org/10.1111/etp.12029>
  8. Pujara, M. (2021). Solar Powered Smart Irrigation System. *International Journal of Advances in Agricultural Science and Technology*.2021;8(3):48–56. Available:<https://doi.org/10.47856/ijaast.2021.v08i3.005>
  9. Ersin A, Gürbüz R, Yakut AK. Application of an Automatic Plant Irrigation System Based Arduino Microcontroller Using Solar Energy. *Solid State Phenomena*. 2016;251: 237–241. Available:<https://doi.org/10.4028/www.scientific.net/ssp.251.237>
  10. Javaid MS. Irrigation and Drainage - Sustainable Strategies and Systems. In *Tech*; 2015. DOI: 10.5772/58498
  11. Vallejo-Gómez D, Osorio M, Hincapié CA. Smart Irrigation Systems in Agriculture: A Systematic Review. *Agronomy*. 2023;13(2):342. Available:<https://doi.org/10.3390/agronomy13020342>
  12. Talaviya T, Shah D, Patel N, Yagnik H, Shah M. Implementation of artificial intelligence in agriculture for optimisation of irrigation and application of pesticides and herbicides. *Artificial Intelligence in Agriculture*. 2020;4:58–73. Available:<https://doi.org/10.1016/j.aiia.2020.04.002>
  13. El-Naggar A, Hedley C, Horne D, Roudier P, Clothier B. Soil sensing technology improves application of irrigation water. *Agricultural Water Management*. 2020;228: 105901. Available:<https://doi.org/10.1016/j.agwat.2019.105901>
  14. Francelino FMA, Coelho FC, Manhães C MC, Oliveira DGD, Neto LR, Silva MPD. The effect of different irrigation levels in cowpea production in the mid-northern region of tocanins—Brazil. *American Journal of Climate Change*. 2022;07(04), 548–557. Available:<https://doi.org/10.4236/ajcc.2018.74033>
  15. Sousa DP, Nunes HG, Ferreira DP, Moura VB, de Aviz WLC, Santos HCA, de Novoa Pinto JV, de Oliveira Vieira IC, Fernandes GST, Silva ERR, Belém LT, da Cunha Junior JB, de Lima MJA, de Sousa AML, da Silva Farias VD, Santos JTS, de Souza PJDOP. Performance of Cowpea under Different Water Regimes in Amazonian Conditions. *Horticulturae*. 2022;8(4):335. Available:<https://doi.org/10.3390/horticulturae8040335>

## APPENDIX - C++ Program for the Arduino Module

```
// Include the necessary libraries
#include <Servo.h>

// Define the pins for the sensors and motors
const int sensorPins[] = {A0, A1, A2, A3};
const int motorPins[] = {2, 3, 4, 5};

// Define the threshold moisture level
const int moistureThreshold = 30;

// Define the servo objects for motors
Servo motors[4];

void setup() {
  // Initialize serial communication
  Serial.begin(9600);

  // Attach servo motors to corresponding pins
  for (int i = 0; i < 4; i++) {
    motors[i].attach(motorPins[i]);
  }
}

void loop() {
  // Read moisture levels from sensors
  for (int i = 0; i < 4; i++) {
    int moistureLevel = analogRead(sensorPins[i]);
    int mappedMoisture = map(moistureLevel, 0, 1023, 0, 100);

    // Check if moisture level is below threshold
    if (mappedMoisture < moistureThreshold) {
      Serial.print("Moisture level below threshold at sensor ");
      Serial.print(i);
      Serial.print(": ");
      Serial.println(mappedMoisture);

      // Trigger motor to pump water
      pumpWater(i);
    }
  }
  delay(1000); // Delay for stability
}

// Function to trigger motor to pump water
void pumpWater(int motorIndex) {
  motors[motorIndex].write(90); // Rotate motor to pump water
  delay(1000); // Wait for water to pump (adjust as needed)
  motors[motorIndex].write(0); // Stop motor
}
```

© Copyright (2024): Author(s). The licensee is the journal publisher. This is an Open Access article distributed under the terms of the Creative Commons Attribution License (<http://creativecommons.org/licenses/by/4.0>), which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Peer-review history:

The peer review history for this paper can be accessed here:

<https://www.sdiarticle5.com/review-history/116393>