CARITAS UNIVERSITY AMORJI-NIKE, EMENE, ENUGU STATE

Caritas Journal of Engineering Technology

*CJET, Volume 4, Issue 2 (2025)*

*Article History*: ***Received: 11th April 2025; Revised: 20th May, 2025; Accepted: 10th June, 2025***

# DESIGN AND IMPLEMENTATION OF SMART IOT BASED PLANT IRRIGATION SYSTEM USING NODEMCU ESP8266 MICROCONTROLLER AND BLYNK INTERFACE TECHNOLOGY

**Ifeagwu E.N.**
Department of Electrical and Electronic Engineering,
Federal University Otuoke, Bayelsa State,
*ifeagwuen@fuotuoke.edu.ng*

**Abba MaryRose Obiageli**
Department of Electrical/Electronic Engineering
Enugu State University of Science and Technology,
Agbani,Enugu State
*obiageli.abba@esut.edu.ng*

**Abstract**

*This paper presents the design and implementation of a Smart IoT-based Plant Irrigation System utilizing the NodeMCU (ESP8266) microcontroller, aimed at automating irrigation based on real-time soil moisture data. The proposed system integrates a soil moisture sensor, water pump, and Wi-Fi-enabled NodeMCU to monitor and manage irrigation processes efficiently. When the soil moisture level drops below a defined threshold, the system automatically activates the water pump, ensuring that plants receive adequate water without human intervention. The NodeMCU sends data, monitor soil moisture levels and irrigation activity through a web dashboard. The system include low-cost components, ease of deployment, and real-time remote monitoring capabilities. The use of sensor enhances the functionality by sending mobile notifications to users and enabling remote control of the pump. The system is powered via a 5V power source and is designed to be scalable and energy-efficient, making it suitable for both home gardens and small-scale agricultural applications. The methodology involves setting up the hardware with NodeMCU, DHT11 (for temperature and humidity), soil moisture sensor, relay module, and a submersible water pump. The firmware is programmed using the Arduino IDE. Moisture data is sampled at regular intervals and used to make decisions about irrigation timing and duration. The system also logs environmental parameters to support data-driven analysis of plant growth conditions. The implemented system was tested over a period of two weeks in a controlled garden environment. It successfully maintained optimal soil moisture levels by activating the water pump automatically whenever moisture fell below the 30% threshold. Data was reliably transmitted to the dashboard as expected. Notifications were received in real-time through the system. The system reduced water usage by approximately 40% compared to manual irrigation.*

**Keywords:** ***IoT, NodeMCU, Soil Moisture sensors, Real time monitoring***

## 1.0 INTRODUCTION

Irrigation is the application of controlled amounts of water to plants at needed interval (Atzori et al., 2010). Irrigation helps to grow agricultural crops, maintain landscapes, and re-vegetate disturbed soils in dry areas and during periods of inadequate rainfall (Gomathy et al., 2021). Water is a very essential natural resource that is highly needed for the proper and adequate growth of plants, and this directly affects the nutrient that man gains from the consumption of agricultural products (Kathiresan et al., 2023). There are several ways in which water is been supplied to a farm or garden for irrigation purposes, some of these include the traditional method and the modern method (Leiner 1997). In the traditional system, irrigation is done manually. Here, a farmer pulls out water from wells or canals by himself or using cattle and carries to farming fields. This method can vary in different regions, but the main advantage of this method is that it is cheap, but its efficiency is poor because of the uneven distribution of water (Negm et al., 2023). Also the chances of water loss are very high. The modern method compensates the disadvantages of the traditional methods and thus helps in the proper way of water usage (Okandeji et al., 2020). This method involves systems such as sprinkler system and Drip system (Patel and Tope 2016). These systems can be automated to operate independently from the interference of the farmer or gardener and still obtain a high productivity using limited water resources under properly controlled conditions (Rawal 2017).

This paper "Design and Implementation of Smart IOT based plant irrigation system using NODEMCU technology" uses a capacitive soil moisture sensor to detect the moisture content of the soil, collects and transmits this data to the NodeMCU (Microcontroller) which uses the real-time data to make decision based on a stored threshold level of the required soil moisture content of the given plant and activates the dc pump motor to supply water to the plant at required intervals (Srivastava et al., 2018). The system also has other components such as temperature and humidity sensor and a PIR motion sensor, which work by obtaining real-time data from the atmospheric condition of the environment and are thus used as control parameters for the adequate functioning of the system to ensure optimum and reliable operation (Tole et al., 2024) .It also has a display screen which displays the current working condition of the system by indicating the present moisture content of the soil and other operational conditions and data which helps the user to monitor the system operation.

Furthermore, the system is Web integrated (using internet of things) which enables the user to interface with the system through a designed mobile application. In addition, the system security is also put into consideration in the design and implementation. The Arduino IDE environment was used for the programming of the microcontroller which carries out the decision making of the system (Jabar et al., 2022).

## 2 BLOCK DIAGRAM OF THE SYSTEM

The block diagram of the propose system is shown in Figure 1.

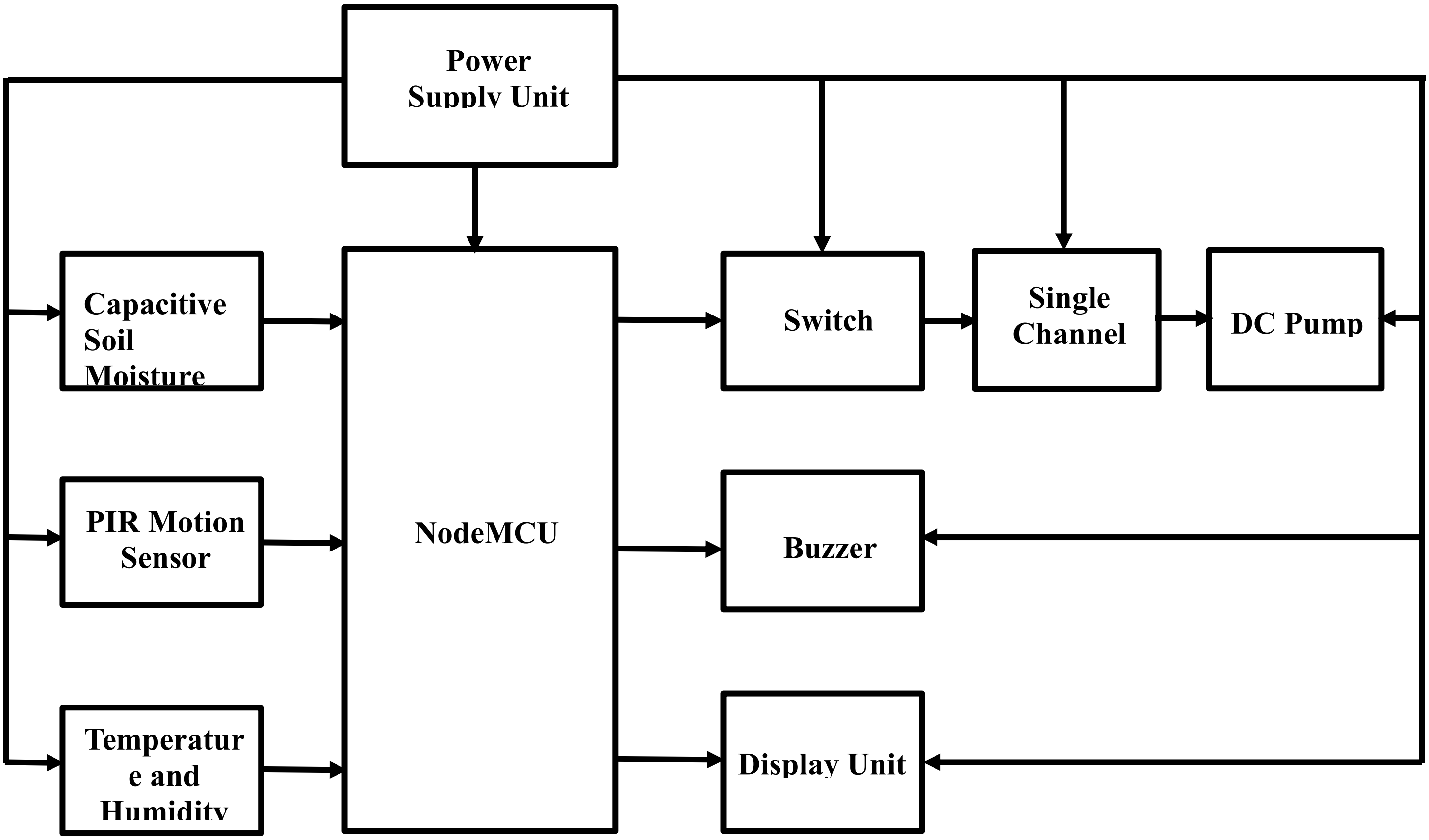



**Figure 1:** Block Diagram

**2.1 BLOCK DIAGRAM DISCRIPTION**

There are two main sensing (input) components as shown in Figure 1.. These include capacitive soil moisture sensor, and temperature and humidity sensor. It also has output components which include water pump, display unit and buzzer. The NodeMCU board is programmed using the Arduino IDE software to interpret the data received from the sensing (input) components in order to initiate the necessary command to the output components of the system (Ifeagwu et al., 2025)..

The function of the capacitive soil moisture sensor is to measure the level of moisture in the soil and sends the signal to the NodeMCU if watering is required or not. The sound of the buzzer indicates the need for watering. The temperature and humidity sensor reads the temperature and humidity of the environment surrounding the plant. The water pump supplies water to the plants. The PIR motion sensor functions by detecting thermal changes generated by moving objects or humans, and subsequently transmits an electrical signal to the controller, which then generates a notification to the user, thereby alerting the user of unauthorized intrusions. Ultimately, this arrangement fosters a robust and reliable security infrastructure of the system.

**3.0 MATERIALS AND METHOD**

Table 1 shows Capacitive Soil Moisture Sensor Specifications. Table 2 shows Temperature and Humidity Sensor Specifications. Table 3 shows PIR Motion Sensor Specifications. Table 4 shows DC pump Specifications. Table 5 shows Single Channel Relay Specifications. Table 6 shows TP4045 Module Specifications. Table 7 shows Display Unit Specifications. Table 8 shows NodeMCU Specifications. Table 9 shows Buzzer Specifications.

## 3.1 COMPONENTS AND THEIRS SPECIFICATIONS

**Table 1:** Capacitive Soil Moisture Sensor Specifications

| NAME | SPECIFICATION |
|---|---|
| Operating voltage range | 3.3 – 5.5 V |
| Operating current | 5mA |
| Interface | PH2.54-3P |
| Dimensions mm(LxWxH) | 98x23x4 |
| Analog output | Voltage(V) |
| Weight(gm) | 15 |
| Supports 3-Pin Gravity sensor interface | |

**Table 2:** Temperature and Humidity Sensor Specifications

| NAME | SPECIFICATION |
|---|---|
| Type | AM2302B |
| Accuracy | 0.1 |
| Humidity range | 0 – 100% RH |
| Temperature range | -40 – 80$^{0}$C |
| Humidity measurement precision | ±2%RH |
| Temperature measurement precision | ±0.5$^{0}$C |
| All calibration | Digital Output |
| 4-Pin package | |
| Ultra-low power | |
| Excellent long-term stability | |

**Table 3:** PIR Motion Sensor Specifications

| NAME | SPECIFICATION |
| --- | --- |
| Type | HC-SR501 |
| Operating voltage | 4V to 12V (+5V recommend) |
| Output voltage (High/Low) | 3.3V TTL/0V |
| Distance covered | About $120^0$ and 7m |
| Operating temperature | $-20^0$C to $+80^0$C |
| Low power consumption of 65mA | |
| Has 3-pin (Vcc, High/Low output, Ground) | |

**Table 4:** DC pump Specifications

| NAME | SPECIFICATION |
| --- | --- |
| Input voltage | DC 3V – 5V |
| Flow rate | 1.2 – 1.6 L/min |
| Operating current | 0..1 – 0.2A |
| Maximum suction distance | 0.8mm |
| Outside diameter of water outlet | 7.5mm |
| Inside diameter of water outlet | 5.0mm |
| Diameter of water inlet | 5.0mm |
| Wire length | 200mm |
| Operating temperature | Less than $80^0$C |
| Weight | 26g |
| Dimensions | 45x30x25mm |

**Table 5:** Single Channel Relay Specifications

| NAME | SPECIFICATION |
|---|---|
| Supply voltage | 3.75 to 6V |
| Supply current with relay de-energized | 2mA |
| Supply current with relay energized | 70 to 72mA |
| Input control signal | Active Low |
| Input control signal current | 1.5 to 1.9mA |
| Relay max contact voltage | 250 VAC or 30VDC |
| Relay max contact current | 10A |
| Dimensions (LxWxH) | 43x17.5x17mm |
| Weight | 30g |

**Table 6:** TP4045 Module Specifications

| NAME | SPECIFICATION |
|---|---|
| Method | Linear charge 1% |
| Charging current | 1A Adjustable (through RPROG) |
| Charging accuracy | 1.5% |
| Input voltage | 4.5V to 5.5V |
| Full charge voltage | 4.2V |
| LED indicator | Red is charging, and Green is full charged |
| Charging input interface | Micro USB |
| Working temperature | -10$^0$C to +85$^0$C |
| Polarity reversal | Not allowed |
| Dimensions | 25x19x10mm |

**Table 7:** Display Unit Specifications

| NAME | SPECIFICATION |
|---|---|
| Component type | 12C LCD 1602 |
| Backlight | Blue with White character colour |
| Pin definition | VCC/GND/SDL and SCA |
| Contrast adjustment | Potentiometer |
| Backlight adjustment | Jumper |
| Default address | 0x27, 0x3F |
| Working voltage | 5v |

**Table 8:** NodeMCU Specifications

| NAME | SPECIFICATION |
|---|---|
| Microcontroller | Tensilica 32-bit RISC CPU Xtensa LX106 |
| Operating voltage | 3.3V |
| Input voltage | 7 – 12V |
| Digital I/O pins (DIO) | 16 |
| Analog input pins (ADC) | 1 |
| UARTs | 1 |
| SPIs | 1 |
| 12Cs | 1 |
| Flash Memory | 4MB |
| SRAM | 64KB |
| Clock speed | 80MHz |

**Table 9:** Buzzer Specifications

| NAME | SPECIFICATION |
|---|---|
| Voltage range | 3V to 8V |
| Current (Max) | 40mA |
| Sound output | 80dB at 20cm |
| Frequency | 100Hz |
| Diameter | 26mm |
| Mounting holes | 2mm |

## 3.2 SYSTEM CONNECTION AND ANALYSIS

### 3.2.1 NodeMCU

This is the basic controlling unit of the system. It has 16 input/output pins and one analog pin. Since it is an open-source platform, it was programmed based on the desired operation using the Arduino IDE. The connections of all the components to the NodeMCU board were defined in the programming code. This is important because it helps to ensure that the program will work as intended.

**Steps for Using NodeMCU:**

**Step 1:** The Arduino IDE was installed on the computer.

**Step 2:** The NodeMCU was connected to the computer using a micro-USB cable.

**Step 3:** This involved uploading a test sketch to verify if the NodeMCU was functioning correctly. The "Blink" example sketch was opened from the "File" ->"Examples" -> "ESP8266" -> "Blink" menu. The LED_BUILTIN value in the sketch was modified to "2" to correspond with the NodeMCU's built-in LED pin. Finally, the sketch was compiled and uploaded to NodeMCU by clicking on the "Upload" button in the Arduino IDE. Successful uploading was indicated by the onboard LED blinking.

**Step 4:** This focused on connecting the components of the system. A circuit was designed on a board, connecting all the components to the NodeMCU using jumper wires. The code for the project was written in Arduino IDE. The code was then uploaded to NodeMCU using the "Upload" button in Arduino IDE. When required, the serial output was monitored for debugging purposes. The project was tested and iterated upon until it met the desired functionality.

### 3.2.2 Capacitive Soil Moisture Sensor

**Connection:** The capacity soil moisture sensor has four pins – Vcc, GND, DO and AO. The Vcc was connected to 3V3, GND to GND and AO to A0 pins of the NodeMCU respectively.

**Calibration:** The capacitive soil moisture sensor was calibrated using the NodeMCU. The readings were obtained under the following conditions: dry condition, optimum condition, and wet condition.

The reading for the dry condition was obtained when the sensor was placed in the soil under dry condition and embedded up to a fair depth of the soil. The optimum condition was obtained when water was added to the soil. The wet condition was obtained with the increase in water content beyond the optimum level. Generally,

these conditions are set based on the moisture requirement of the soil and plant. However, in this project, they were established solely for testing purposes, tailored to evaluate the efficacy of the system.
The table below shows the readings obtained at various conditions.

**Table 10:** Soil Moisture Sensor Readings

Table 10 shows Soil Moisture Sensor Readings

| Condition | Value | |
|---|---|---|
| | Analog Values | Percentage(%) |
| Dry condition | 721 | 0 |
| Optimum condition | | 30 |
| Wet condition | 543 | 100 |

The circuit diagram for the connection of the soil moisture sensor is shown in Figure 2.

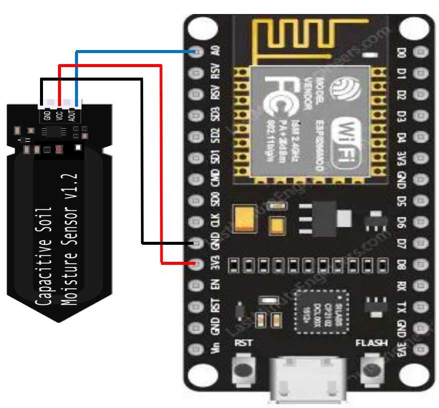



**Figure 2:** Soil Moisture Sensor Circuit

### 3.2.3 Temperature and Humidity Sensor

**Connection:** The Vcc, GND, and Data pin of the temperature and humidity sensor were connected to the 3V3, GND and D4 pins of the NodeMCU respectively. Figure 3 shows Temperature and Humidity Sensor Circuit Diagram

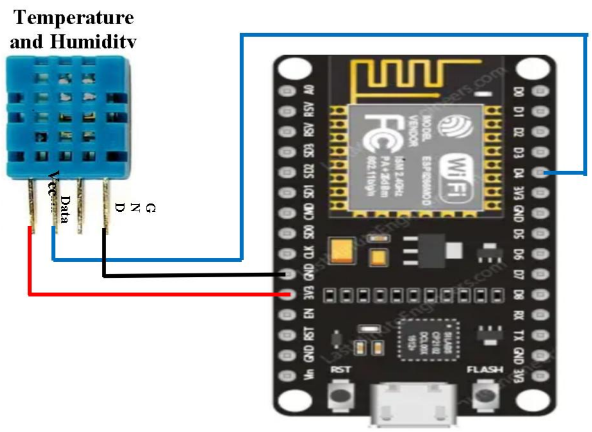



**Figure 3:** Temperature and Humidity Sensor Circuit Diagram

### 3.2.4 PIR Motion Sensor

**Connection:** Pin1 (Vcc), Pin2 (GND) and Pin3 (OUT) of the PIR motion sensor were connected to Vin, GND and D3 pins of the NodeMCU respectively. Figure 4 shows PIR Motion Sensor Circuit Diagram

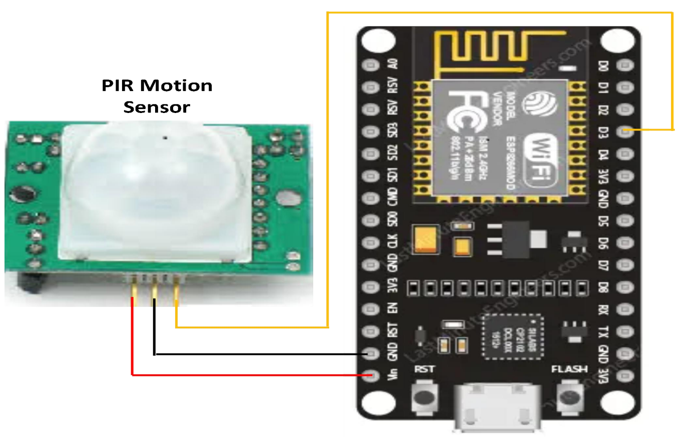



**Figure 4:** PIR Motion Sensor Circuit Diagram

### 3.2.5 Single Channel Relay

**Connection:** For the input terminal, Vcc, GND and Trigger pins were connected to Vin, GND and D0 pins of the NodeMCU respectively. For the output terminal, COM to Vin, Load black to GND and Red to normally open. Figure 5 shows Single Channel Relay Circuit Diagram

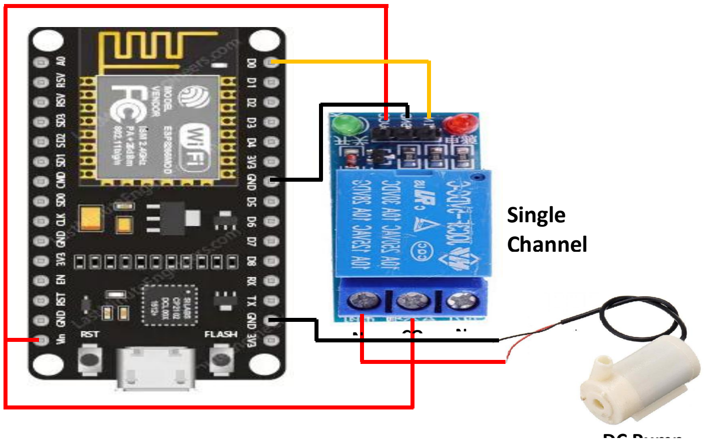



**Figure 5:** Single Channel Relay Circuit Diagram

### 3.2.6 Buzzer

**Connection:** The Red and Black wires of the buzzer were connected to D5 and GND of the NodeMCU pins respectively. Figure 6 shows Buzzer circuit Diagram.

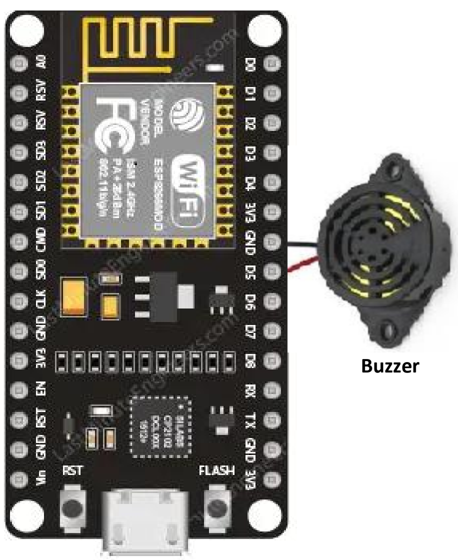



**Figure 6:** Buzzer circuit Diagram

## 3.3 SYSTEM CIRCUIT DIAGRAM

The interconnection of the various components is shown in Figure 7

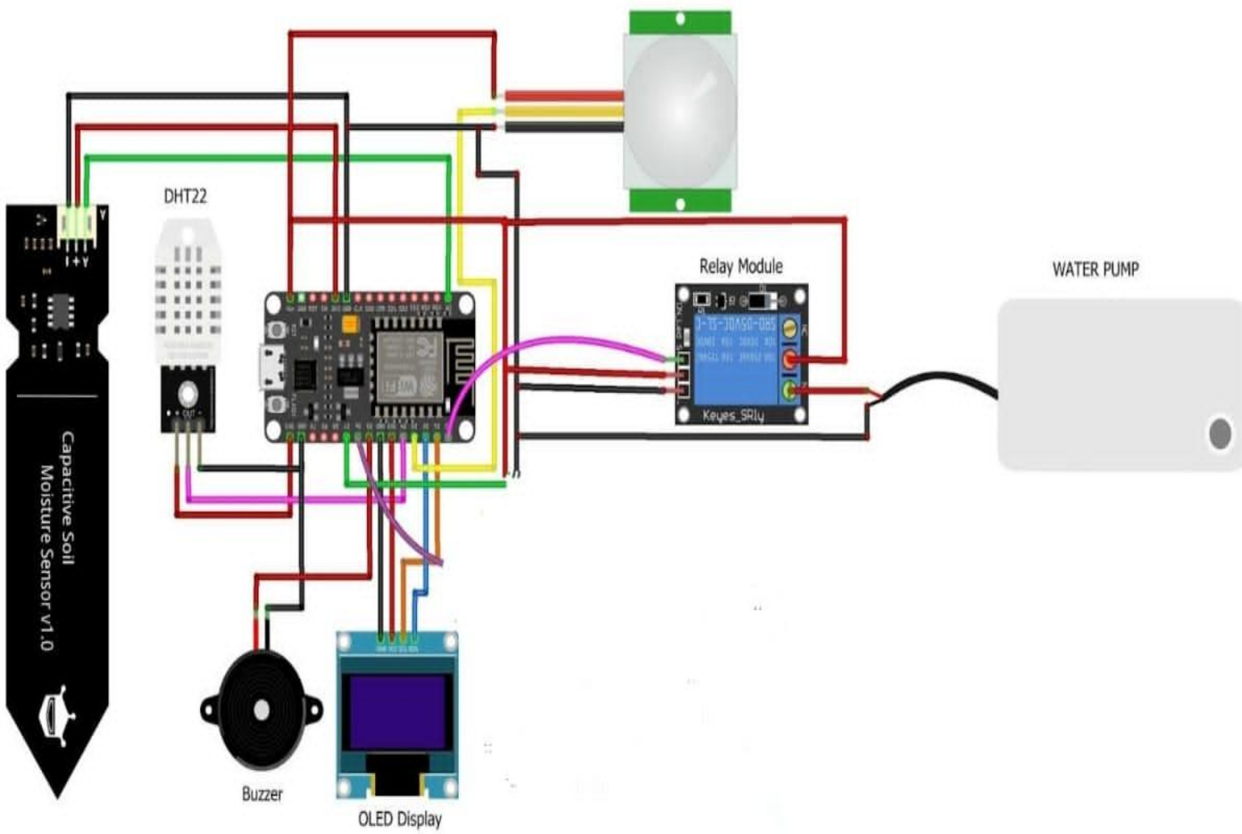



**Figure 7 :** System Circuit Diagram

## 3.4 SYSTEM FLOW CHART

Figure 8 shows System Flow Chart

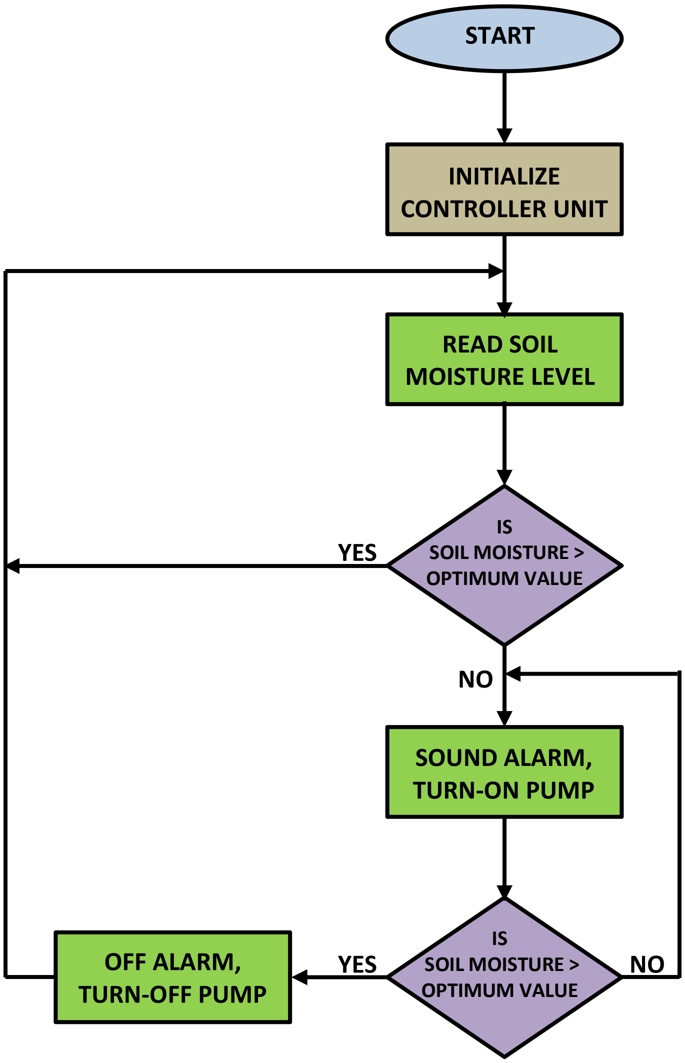



**Figure 8:** System Flow Chart

## 4.0 RESULTS AND DISCUSSION

### 4.1 WORKING PRINCIPLE OF THE OVERALL SYSTEM

Figure 9 shows the overall system. The system consists of a capacitive soil moisture sensor, PIR motion sensor, temperature and humidity sensor, buzzer, dc pump and the NodeMCU board. The controller interprets the real-time data from the soil moisture sensor and then triggers the pump when irrigation is needed. The buzzer indicates when irrigation is needed. The PIR notion sensor which provides the security infrastructure of the system senses movement and generates a notification to the use the sensor. This was tested to evaluate its functionality and effectiveness. Movements within the detection zone of the sensor were made, these movements were captured by the sensor and the notifications were sent to the user on the mobile and desktop interfaces respectively.

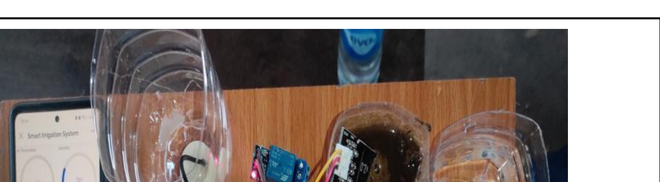


**Figure 9:** The overall system

### 3.5 PIR MOTION SENSOR

The PIR motion sensor which provides the security aspect of the system was programmed to generate a notification to the user when a movement is detected within the detection zone of the motion sensor. This is shown in Figure 10 (a) and Figure 10 (b). Figure 11 **shows** The output of PIR Motion Sensor when it detects motion.

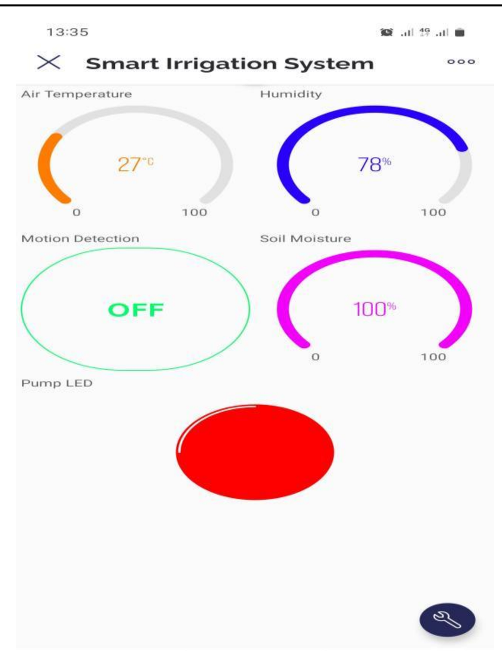



**Figure 10 (a)** PIR Motion Sensor in an OFF state

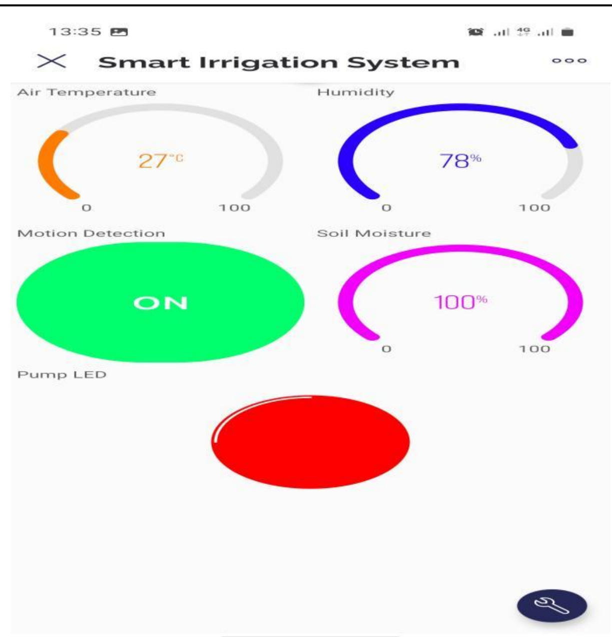
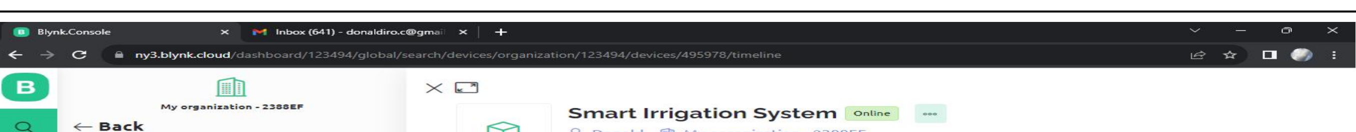



**Figure 10(b)** PIR Motion Sensor in an ON state



## 4.5 WEB INTERFACE

**Figure 11:** The output of PIR Motion Sensor when it detects motion

The Web integration of this work was actualized using the Blynk interface which is an app that allows one to control hardware devices and view sensor data using a smart phone or tablet. The app was used to configure a simple interface for viewing the sensors' data and controlling the overall system. Figure 12 shows the Blynk interface.

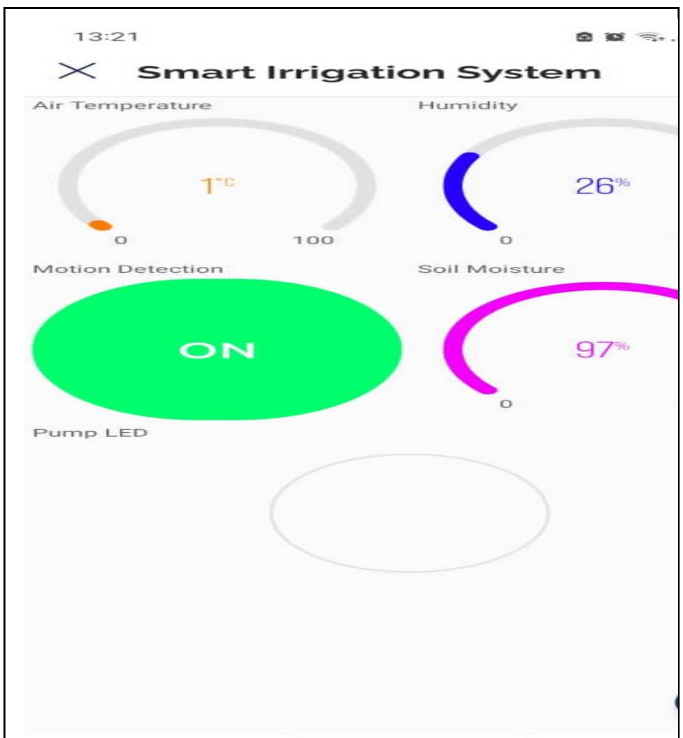



**Figure 12:** The Blynk interface

### 4.3 ANALYSIS

i. Water savings for the system as a whole have not been studied. However, the performance of the system in terms of water conservation has been demonstrated through previous experiments in real agricultural contexts.

ii. An experiment showed that it is difficult to maintain constant soil moisture using human feedback alone. In the short periods in which this system was tested, virtually no human intervention was required. The user only needs to check that the system is operational and that the water tank, if in use, is not empty. On the other hand, there is no way to notify the user about emergencies such as empty tank, component failure, etc.

iv. Further testing should be conducted in a real home or greenhouse to evaluate the reliability and durability of the system. These tests should also be extended to determine the importance of water and labor savings.

i. All components have been selected to achieve a certain energy efficiency in order
not to exceed the power supply of the microcontroller.

ii. The PIR Motion only sends a notification when it detects motion within its zone on the Blynk interface, no sound is initiated. However, the system can be modified to include a second buzzer that generates a sound in respect to the PIR motion sensor signal.

iii. No regular maintenance of the irrigation system is required other than refilling the water tank (if used), cleaning the pipes and valves, and replacing parts if they are broken or failed. Most replacement components are available at an electronics or hardware store.

## 5.0 CONCLUSION

The soil moisture sensor system is designed to place a sensor in a specific area of the garden to detect if there is enough moisture in the soil for the plants. If the soil is dry, the system will activate and provide water to the plants, but if there is sufficient moisture, it will prevent unnecessary irrigation. The Microcontroller based irrigation system effectively monitors and controls all irrigation activities by measuring soil moisture and temperature, ensuring water is supplied to the plants based on their specific needs. This not only prevents water wastage but also improves plant quality and saves time by removing the need for manual adjustments. Overall, it helps farmers maximize their profits.

## REFERECES

Atzori, L., lera, A., Morabito, G.(2010). The internet of Things: A survey. *Computer. Networks*, 54(15), 2787-2805.

Gomathy, C.K., Vamsikumar, A., Karthik, B.(2021). The Smart Irrigation System Using IoT. *International Research Journal of Engineering and Technology,* 8(10), 1-5.

Ifeagwu, E.N., Okafor, J. O. (2025). Fuzzy based Irrigation improvement system for Nigerian Agricultural Fields. *Caritas Journal of Engineering Technology, 3(2), 75-94.*

Jabar, H., Yousif, K. A., Mohamed, O. B. (2022). Experimental and Mathematical Models for Real Time Monitoring and Auto Watering Using IOT Architecture. *Computers*, 11(1), 7.

Janani, M., Naveen, K. M., Narashimabalaji, E. (2023). Solar Energy Based Smart Irrigation system Using IoT. *Journal of ISMAC*, 5(2), 83-99.

Leiner, M.(1997). The past and future history of Regulus. *Communications of the ACM*, 40(2), 102-108.

Negm, E.M. (2023). IOT-Smart irrigation with tracking system using Node MCU ESP8266. *International Journal of Scientific Research In Computer Science and Engineering*, 11(6), 43-52.

Okandeji, A.A., Onaifo, F., Kabir, M.T., Yakubu, K. (2020). Design and Implementation of Internet of Things based Irrigation System. *Arid Zone Journal of Engineering, Technology and Environment,* 16(4), 663-674.

Patel, A.S., Tope, G. (2016). Smart Irrigation System. *International Journal of Engineering and Technical Research*, 3(3), 23-26.

Rawal, S. (2017). IOT based Smart Irrigation System. *International Journal of Computer Applications*,159 (8), 7-11.

Srivastava, P., Bajaj, M., Rana, A.S.(2018). Irrigation System using IOT. *Fourth International conference on Advances in Electrical, Electronics, Information, Communication and Bio Informatics ,Piscataway, New jersey,* USA.

Tole, S., Ahmad, N., Tri, W., Watra, A. (2024). Smart Irrigation system using node microcontroller unit ESP8266 and Ubidots Cloud platform. *Computer Science and Information Technologies,* 5(2),168-175.