International Journal of Science, Strategic Management and Technology
Volume **02** Issue **02** February-2026 | ISSN: **3108-1762 (Online)** | Impact Factor: 3.8
An International, Peer-Reviewed, Open Access Scholarly Journal Indexed in recognized academic databases

Raspberry PI - Powered Automated Garden System

Dr. A. Theja[1],B. Sravani[2],B. Siddhu kumar[3],D. Sravan kumar[4].

[1]Associate Professor, Department of Electronics and communication Engineering, TKR College of Engineering and Technology.
[2,3,4] UG Scholars, Department of Electronics and communication Engineering, TKR College of Engineering and Technology, Medbowli, Meerpet.

https://doi.org/10.55041/ijsmt.v1i1.001

Cite this Article: kumar, B. S. B. S. K. ,. S. (2026). Raspberry PI - Powered Automated Garden System. International Journal of Science, Strategic Management and Technology, *Volume 10*(01). https://doi.org/10.55041/ijsmt.v2i2.038

Abstract- This paper presents an intelligent garden automation system using Raspberry Pi and Arduino for continuous environmental monitoring and smart irrigation. The system integrates sensors to measure soil moisture, temperature, humidity, and light intensity. Based on predefined thresholds, it activates a water pump using a relay module to irrigate plants automatically. The solution aims to reduce water wastage and manual labor while enhancing plant care in urban gardening or greenhouse setups. Designed with modular components and written in Python, the system provides an affordable and scalable approach to sustainable agriculture.

Indexed Terms: Raspberry Pi, Arduino, IoT, Soil Moisture Sensor, Smart Irrigation, Environmental Monitoring, Automation.

I. INTRODUCTION

Urban gardening and greenhouse farming face challenges related to environmental control and efficient irrigation. Traditional methods often lead to overwatering or neglect due to human limitations. The proposed system leverages embedded computing and IoT to automate the monitoring and irrigation process using Raspberry Pi and Arduino. It collects real-time data and responds by activating actuators, offering a sustainable and low-maintenance solution. This automation not only ensures optimal watering based on real-time conditions but also significantly conserves water resources solution. This automation not only ensures optimal watering based on real-time conditions but also significantly conserves water resources

II. LITERATURE SURVEY

Several studies have explored the Raspberry Pi or Arduino to automate irrigation based on fixed schedules or sensor inputs:

[1] **Sandhya. B,** et al. (2022), "Developed IoT Based Smart Home Garden Watering System Using Raspberry Pi 3, focused on soil moisture-based irrigation using Raspberry Pi for home gardens but lacked integration of temperature and humidity sensors.

[2] **S.N. Ishak** et al. (2019), "Developed Smart Home Garden Irrigation System Using Raspberry Pi, used Arduino with fixed-time or single-sensor irrigation, but lacked real-time adaptation to environmental changes.

[3] **Sneha Angal** (2018), "Design Raspberry Pi and Arduino Based Automated Irrigation System" focused on Combined Arduino with ZigBee for wireless control, but lacked a user- friendly interface and had limited sensor integration.

III. PROPOSED METHODOLOGY

The proposed system combines the capabilities of both Arduino Uno and Raspberry Pi to create an intelligent and automated garden monitoring solution. The Arduino Uno functions as a real-time data acquisition unit, interfacing with multiple environmental sensors such as the soil moisture sensor, DHT11 temperature and humidity sensor, and LDR. These sensors continuously monitor the garden's environmental parameters and send the data to the Raspberry Pi via serial communication.The Raspberry Pi 3 acts as the central processing and control unit.

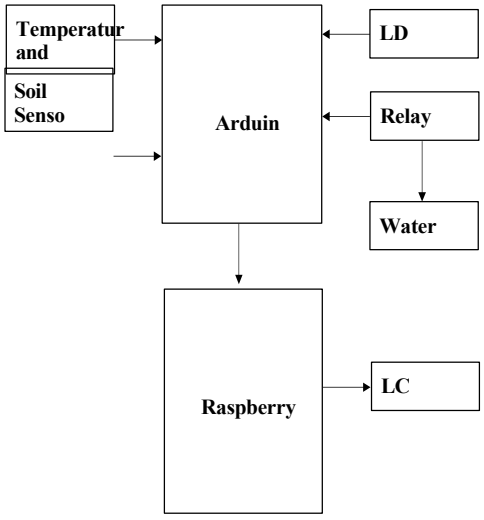

Figure 1. Block Diagram of Raspberry Pi Powered Automated Garden System

The block diagram illustrates the architecture of the system include a Raspberry Pi 3, which manages the logic, stores data, and can host a web server or dashboard. An Arduino Uno interfaces with various sensors to collect raw environmental data. The sensors used are the DHT11 for measuring temperature and humidity, a capacitive soil moisture sensor for detecting soil moisture levels, and an LDR (Light Dependent Resistor) for monitoring light intensity. A relay module is utilized to control high-power devices like the water pump, which is activated based on real-time soil moisture readings. The system is powered by either a conventional power supply or a solar- powered setup, making it suitable for off-grid applications. An LCD display is used to show live environmental parameters. On the software side, Python is employed for control logic and data handling on the Raspberry Pi, while the Arduino IDE is used for programming the Arduino to collect sensor data. Serial communication (UART) facilitates data transfer between the Raspberry Pi and Arduino. Additionally, an optional Flask framework can be used to host a lightweight local web interface for monitoring and controlling the system.

IV. FLOWCHART

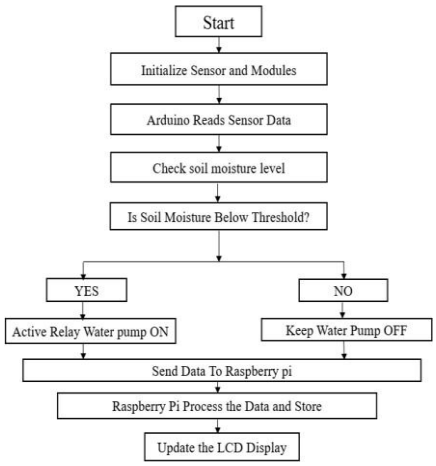

Figure 2. Flow Chart Raspberry Pi Powered Automated Garden System

1. System Initialization:

At the beginning, all hardware components Arduino, Raspberry Pi, sensors (soil moisture, temperature, humidity, and light), and the relay module are initialized. This ensures that each device is correctly powered, communication channels are established, and the system is ready to operate without errors.

2. Sensor Data Acquisition:

Once initialized, the Arduino starts collecting real-time data from all connected sensors. This includes soil moisture readings to check hydration levels, temperature and humidity data to assess atmospheric conditions, and light intensity measurements to determine sunlight exposure. These inputs are essential for making

informed decisions about plant care.

3. Soil Moisture Evaluation:

The core of the decision-making process is the evaluation of soil moisture. The Arduino compares the sensor's moisture reading with a predefined threshold value. If the value is below the threshold, it indicates that the soil is dry and requires watering to support healthy plant growth.

4. Decision-Making Process:

Based on the sensor data, the Arduino makes a decision. If watering is needed, it activates the relay module, which turns on the water pump to irrigate the plants. If the moisture level is sufficient, the system conserves water by keeping the pump off. This automation ensures timely irrigation and reduces human intervention.

5. Data Communication to Raspberry Pi: After action is taken, the Arduino sends the collected sensor data and pump status to the Raspberry Pi using serial communication. This allows centralized processing, remote access, and integration with user interfaces like web dashboards.

6. Data Processing and Storage:

The Raspberry Pi receives and processes the data, storing it in local or cloud-based databases. This enables tracking environmental trends, analyzing plant health, and refining irrigation strategies over time for better efficiency.

7. Display Update:

Finally, the system updates the LCD display to show current values such as soil moisture percentage, temperature, humidity, and whether the pump is on or off. This immediate feedback allows users to monitor the garden's status at a glance, even without a smartphone or PC.

V. SCHEMATIC EXPLANATION

The schematic diagram of the Raspberry Pi-Powered Automated Garden System illustrates the integration of sensors, microcontrollers, and actuators for intelligent irrigation. The Arduino Uno collects real-time data from sensors including a DHT11 (for temperature and humidity), a soil moisture sensor, and an LDR (light sensor). Based on the soil moisture reading, the Arduino decides whether to activate a relay module that controls a water pump, enabling automated watering when the soil is dry. This sensor data, along with system status, is transmitted to the Raspberry Pi via serial communication. The Raspberry Pi processes this information, logs it, and displays the current environmental conditions on an I2C-connected LCD. It may also host a web-based dashboard for remote access. Power is supplied through separate sources: a USB or battery for the microcontrollers and a higher-voltage source for the pump, managed safely through the relay. This setup ensures real-time environmental monitoring, efficient water usage, and minimal manual intervention, making it ideal for sustainable and smart gardening.

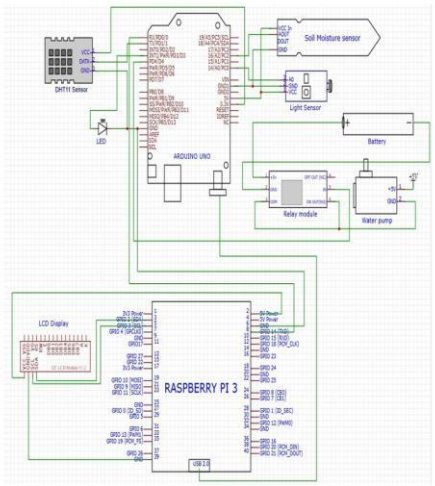

Figure 3. Schematic Diagram

VI. RESULT

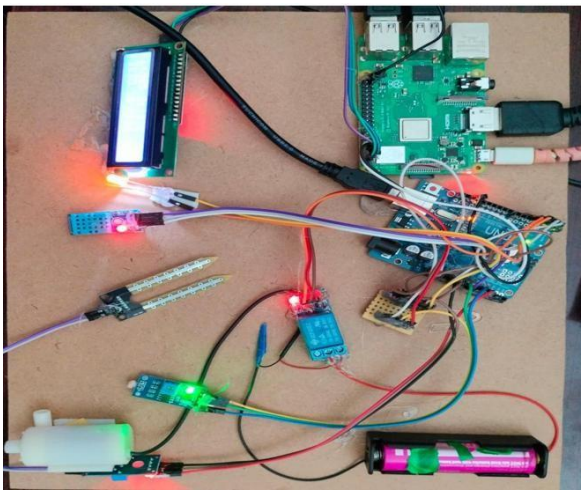

Figure 4. Hardware Setup of Raspberry Pi Powered Automated Garden System

The hardware setup of the Raspberry Pi- Powered Automated Garden System involves a combination of Arduino Uno and Raspberry Pi 3 working together to automate garden monitoring and irrigation. The Arduino

Uno is connected to various sensors such as a soil moisture sensor, a DHT11 for temperature and humidity, and an LDR for detecting ambient light. It reads sensor data and controls a relay module that switches a water pump on or off based on the soil's moisture level. The Raspberry Pi receives the sensor data from the Arduino via serial communication and processes it for display and remote access. An I2C LCD display is connected to the Raspberry Pi to show real-time values such as temperature, humidity, soil moisture, and light intensity. A reliable power supply is provided to both the Arduino and Raspberry Pi, while the water pump is powered separately through the relay. All components are interconnected with proper wiring on a breadboard or PCB, forming a complete, functional system for automated garden care.

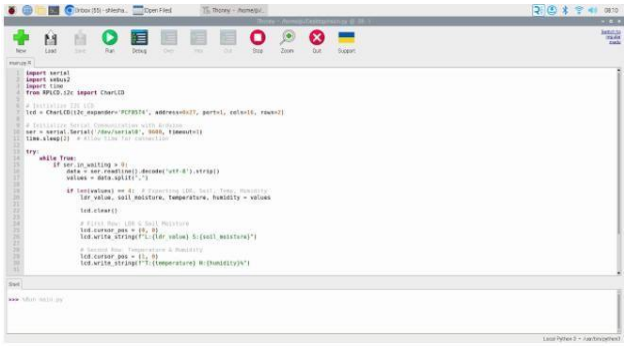

Figure 5.Execution of Code

Figure 6 Output of LCD Display

The output on the LCD display in the Raspberry Pi-Powered Automated Garden System provides real-time feedback on the garden's environmental conditions, making it easy for users to monitor the system at a glance. The LCD, typically a 16x2 I2C module, displays key sensor readings such as soil moisture percentage, temperature in degrees Celsius, humidity percentage, and ambient light intensity. This information is received and processed by the Raspberry Pi, which formats the data and sends it to the LCD for display. The LCD also indicates the status of the water pump (ON/OFF), allowing users to know when irrigation is active. By presenting this data directly on the hardware, the system ensures users have immediate access to vital garden information without needing to connect to a separate device or platform. fulfilling the IoT-based control objective of the project.

VII. APPLICATIONS

• Home gardeners can automate watering, light adjustment, and climate monitoring. This helps them care for plants easily, even when busy or traveling..
• In greenhouses, the system controls temperature, humidity, and light. It ensures plants grow well regardless of outside weather conditions.
• Urban and vertical farms use it to manage limited water and energy efficiently. Sensors adjust care across multiple layers for sustainable city farming.
• Agricultural researchers monitor soil, temperature, and light to find best crop conditions. Data helps improve farming methods and boost yields.
• Hydroponic and aquaponic setups benefit from automated water and nutrient control.

VIII. ADVANTAGES

1. Automated Garden Care
2. Real-Time Monitoring
3. Remote Control and Accessibility
4. Cost Efficiency
5. Water and Energy Conservation

IX. CONCLUSION

The Raspberry Pi Powered Automated Garden System offers an efficient and user-friendly solution for modern gardening. By automating tasks like watering and monitoring environmental conditions, it reduces manual effort and ensures plants receive optimal care. The system conserves water and energy, supports remote access, and is cost- effective, making it ideal for home gardens, greenhouses, and urban farming. Overall, it promotes smarter, more sustainable gardening practices while simplifying plant care for users of all experience levels.

X. REFERENCES

[1] Almeida, D., & Silva, R. (2017). *IoT- based smart agriculture: A survey of applications and challenges*. Int. Conf. on Computing and Communication Technologies.

[2] Goyal, A., & Agarwal, R. (2019). *Arduino and IoT-based smart garden monitoring system*. *Int. J. Eng. Res. Technol.*, 8(5), 29–34.

[3] Sharma, S., & Soni, P. (2020). *IoT- based smart irrigation and plant monitoring system*. *Int. J. Comput. Sci. Mob. Comput.*, 9(4), 1–6.

[4] Yadav, R., & Kumar, P. (2021). *AI and IoT-based smart agriculture system: Challenges and future scope*. *J. Agric. Eng. Technol.*, 5(3), 145–152.

[5] Patel, V., & Trivedi, M. (2018). *Raspberry Pi based automated plant watering system*. *Int. J. Adv. Res. Comput. Sci. Softw. Eng.*, 8(4), 45–50.

[6] Kumar, A., & Singh, A. (2020). *IoT- based smart farming: Future prospects and challenges*. Proc. 2nd ICISASC.

[7] Behera, S. K., & Ray, P. K. (2020). *Smart farming: An IoT-based approach to agriculture automation*. *Springer Adv. Comput.*, 6(2), 103–118.

[8] Shukla, S., & Soni, P. (2021). *Remote monitoring and control of garden systems using IoT*. *Int. J. Comput. Appl.*, 174(8), 14–19.

[9] Koh, P., & Leong, C. (2020). *Robotic systems for agricultural applications: The future of smart farms*. *J. Agric. Robot. Smart Technol.*, 2(1), 37–45.

[10] Liu, Z., & Zhang, Q. (2020). *Smart greenhouses: IoT-based automated monitoring and control for sustainable agriculture*. *Agric. Syst. Eng. J.*, 9(3), 221–229.