
ISSN: 2321-2152
IJMECE
International Journal of modern
electronics and communication engineering
E-Mail
editor.ijmece@gmail.com
editor@ijmece.com
www.ijmece.com

ISSN 2321-2152
www.ijmece.com
Vol 12, Issue 4, 2024

AUTOMATIC PLANT WATERING SYSTEM & SIMULATION USING TINKERCAD

Chilukeshwaram Aryan, Ms.Vijaya Sree Swarupa
UG Student, Department of Electronics and Computer Engineering, JBIET, India.
Assistant professor, Department of Electronics and Computer Engineering, JBIET, India.

ABSTRACT

Here in we introduce automatic plant watering system, which is considered as one of the most commonly used and the most beneficial automated systems nowadays, which help people in their daily activities by reducing or completely replacing their effort.

This system uses sensor technology along with microcontroller and other electronics in order to behave like smart switching system which senses soil moisture level and irrigates the plant if necessary.

Purpose of this work is to show how someone can easily make own and cheap automatic plant watering system in just few hours by connecting certain electronic components and other materials required.

As one possible agricultural solution, this system can be very helpful in keeping vegetables and other useful and specific plants watered for bigger harvest, which enables farmers from all around world to breed crops of these plants which are the most wanted and the most commonly used in diet.

INTRODUCTION

An automatic plant watering system is a convenient and efficient way to ensure that your plants receive the right amount of water without the need for manual intervention. This system can be especially beneficial for busy individuals or for maintaining plants in remote locations. By using an Arduino microcontroller, sensors, and other components, you can create a smart irrigation system that monitors soil moisture levels and waters the plants as needed.

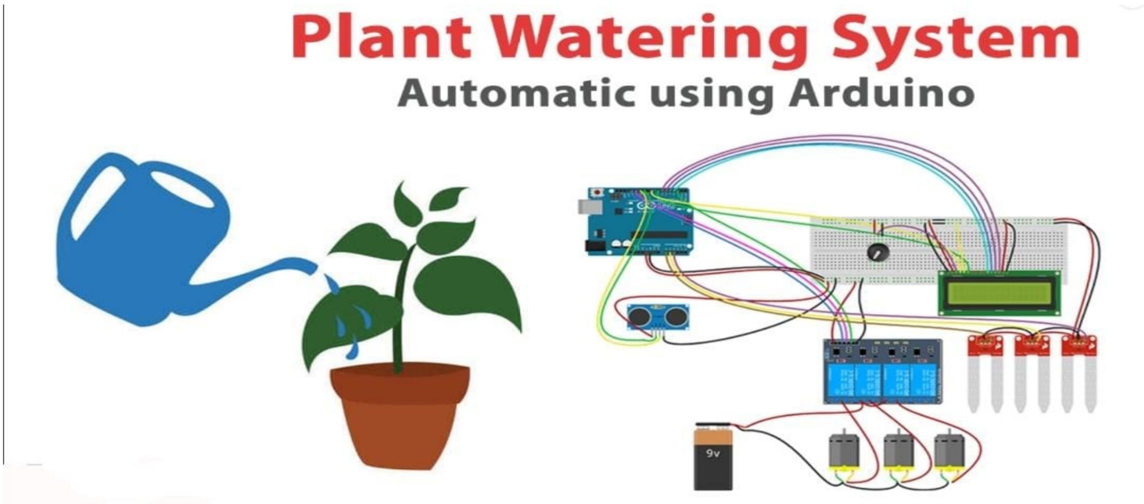

Fig:1.1General picture or Rough picture of the system.

- **How It Works**

1. **Soil Moisture Measurement**: The soil moisture sensor continuously monitors the moisture content of the soil. When it detects that the soil is dry (below a certain threshold), it sends a signal to the Arduino.
2. **Activating the Pump**: Once the Arduino receives the signal indicating low moisture, it activates the relay, which powers the water pump.
3. **Watering the Plant**: The pump draws water from a reservoir and delivers it to the plant through the tubing.
4. **Stopping the Pump**: After a predefined time or once the moisture level reaches a certain threshold, the Arduino turns off the pump.

Simulation Using Tinkercad

Tinkercad is a user-friendly online platform that allows you to create and simulate electronic circuits, including Arduino projects. Here's a brief guide to simulating your automatic plant watering system in Tinkercad:

LITERATURE SURVEY

Introduction to the system

1.Abhishek Gupta, Shailesh Kumawat, and Shubham Garg, "Automatic plant watering system", Imperial Journal of International Research (IJIR), Vol-2, Issue-4, SKIT Jaipur, 2016.

This project is taken up as India is an agriculture oriented country and the rate at which water resources are depleting is a dangerous threat hence there is a need of smart and efficient way of irrigation. In this project we have implemented sensors which detect the humidity in the soil (agricultural field) and supply water to the field which has water requirement. The project is PIC16F877A microcontroller based design which controls the water supply and the field to be irrigated. There are sensors present in each field which are not activated till water is present on the field. Once the field gets dry sensors sense the requirement of water in the field and send a signal to the microcontroller. Microcontroller then supply water to that particular field which has water requirement till the sensors is deactivated again. In case, when there are more than one signal for water requirement then the microcontroller will prioritize the first received signal and irrigate the fields accordingly.

This project uses PIC16F877A Microcontroller. It is programmed in such a way that it will sense the moisture level of the plants and supply the water if required. This type of system is often used for general plant care, as part of caring for small and large gardens. Normally, the plants need to be watered twice daily, morning and evening. So, the microcontroller has to be coded to water the plants in the greenhouse about two times per day. People enjoy plants, their benefits and the feeling related to nurturing them. However for most people it becomes challenging to keep them healthy and alive. To solve this problem we made a project for those who cannot water the plant due to their busy schedule or when they go outside for long time. The system automation is designed to be assistive to the user. We hope that through this project people will enjoy having plants without the challenges related to absent or forgetfulness.

SYSTEM ARCHITECTURE

ISSN 2321-2152
www.ijmece.com
Vol 12, Issue 4, 2024

The automatic plant watering system using Arduino is an innovative and efficient solution designed to manage irrigation for plants automatically. By leveraging the capabilities of Arduino, a popular open-source microcontroller platform, this system simplifies plant care and optimizes water usage, addressing the growing concerns of water scarcity and the challenges of maintaining healthy plants.

Below is an overview of the core components and structure of the system:

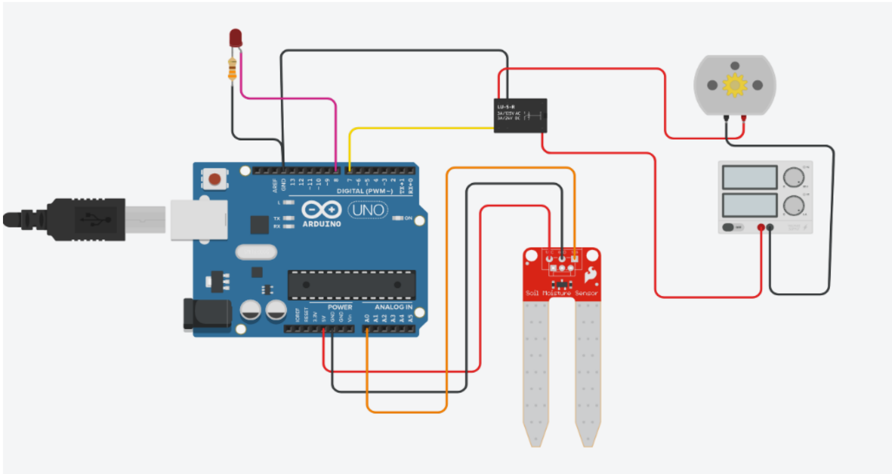

FIG: 3.1.1 External Architecture of Automatic Plant Watering System

Description and role of each component used in this system:

- **Arduino Uno**
- **Description**: The Arduino Uno is a popular microcontroller board based on the ATmega328P. It features 14 digital input/output pins, 6 analog inputs, a USB connection for programming, and a power jack.

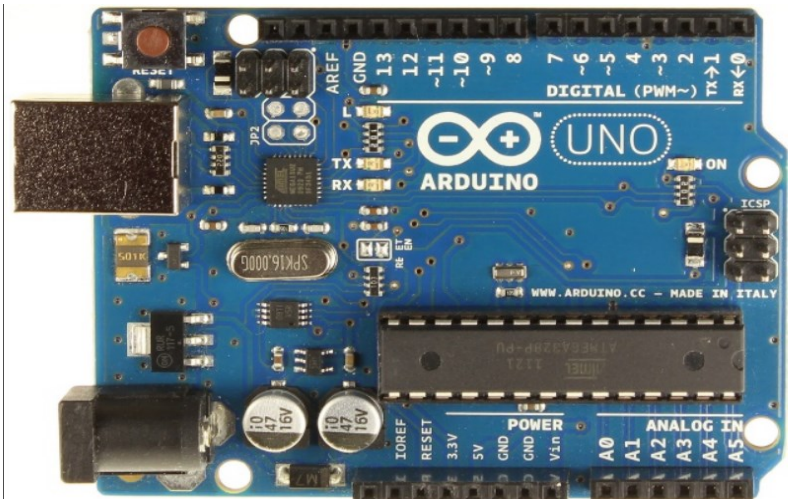

Fig:4.1.1 Arduino

- **Role in the System**: It serves as the central control unit, processing input from sensors (like the soil moisture sensor) and controlling outputs (such as the water pump via a relay). The Arduino can be programmed to read moisture levels and activate watering as needed.

Implementation and Results

ISSN 2321-2152
www.ijmece.com
Vol 12, Issue 4, 2024

Implementation of the System:

Implementing a plant watering system using Arduino in Tinkercad is straightforward and great for prototyping. Here's a brief overview of how to set it up:

Step-by-Step Implementation in Tinkercad:

1. Create a New Tinkercad Project:

- Go to Tinkercad and create a new circuit project.

2. Add Components:

- **Arduino Uno**: This will be the main controller.
- **Soil Moisture Sensor**: For measuring soil moisture levels.
- **Water Pump**: Simulate using a small motor.
- **Relay Module**: To control the pump.
- **Jumper Wires**: To connect components.
- **Power Source**: Use the Tinkercad power supply for simulation.

3. Wiring Setup:

- **Soil Moisture Sensor:**
  - Connect VCC to Arduino 5V.
  - Connect GND to Arduino GND.
  - Connect the analog output (A0) to one of the analog pins (A0) on the Arduino.
- **Relay Module:**
  - Connect VCC to Arduino 5V.
  - Connect GND to Arduino GND.
  - Connect the IN pin to a digital pin (e.g., D7).
- **Pump/Motor:**
  - Connect one terminal of the motor to the relay's NO (Normally Open) terminal.
  - Connect the other terminal to the ground (GND) of the power source.

4. Write the Code:

- Use the code provided earlier in the Arduino IDE section of Tinkercad or just take up the block coding which is easy,we need to drag and drop the blocks of the code.

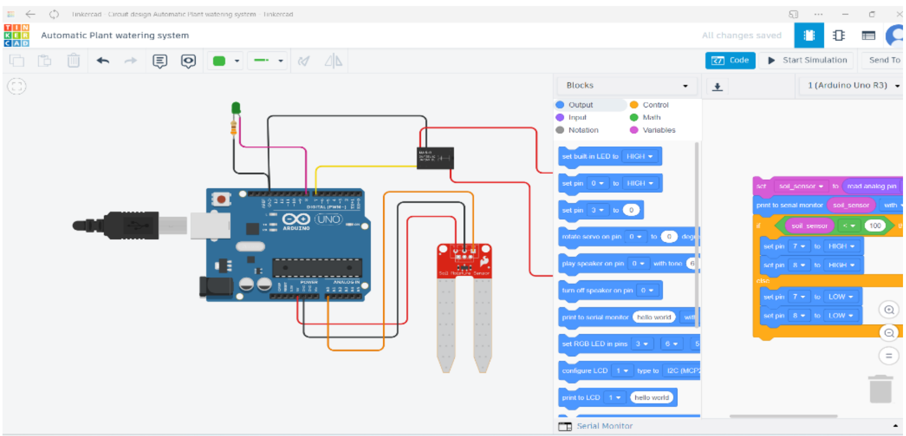

Fig:5.1.1 Code Blocks in Tinkercad

ISSN 2321-2152
www.ijmece.com
Vol 12, Issue 4, 2024

- The basic logic will read the moisture level and activate the pump when the soil is dry.

5. Simulate:

- Start the simulation in Tinkercad.
- You should see the soil moisture readings in the serial monitor. When the moisture level drops below the threshold, the pump will activate.

6. Test and Adjust:

- Modify the moisture threshold and pump duration as needed based on your simulation results.

Final Note:

Tinkercad allows for easy adjustments and visualizing how components interact. Once you're satisfied with the simulation, you can transition to a physical build using the same concepts!

5.2 Results

The results of an automatic plant watering system using Arduino in Tinkercad can be summarized in terms of functionality and learning outcomes. Here's what you can expect:

Functionality:

1. **Soil Moisture Detection**:
   - The soil moisture sensor accurately measures the moisture level in the soil.
   - When the moisture level drops below a predefined threshold, it triggers the watering system.
2. **Pump Activation**:
   - The relay module activates the water pump based on the moisture readings.
   - The pump runs for a specified duration to water the plant when needed.
3. **Data Monitoring**:
   - Users can monitor real-time soil moisture levels via the Tinkercad serial monitor.
   - This helps in understanding how the system responds to changes in soil moisture.
4. **Automatic Operation**:
   - The system operates automatically without manual intervention, providing consistent watering based on soil conditions.

Result-1

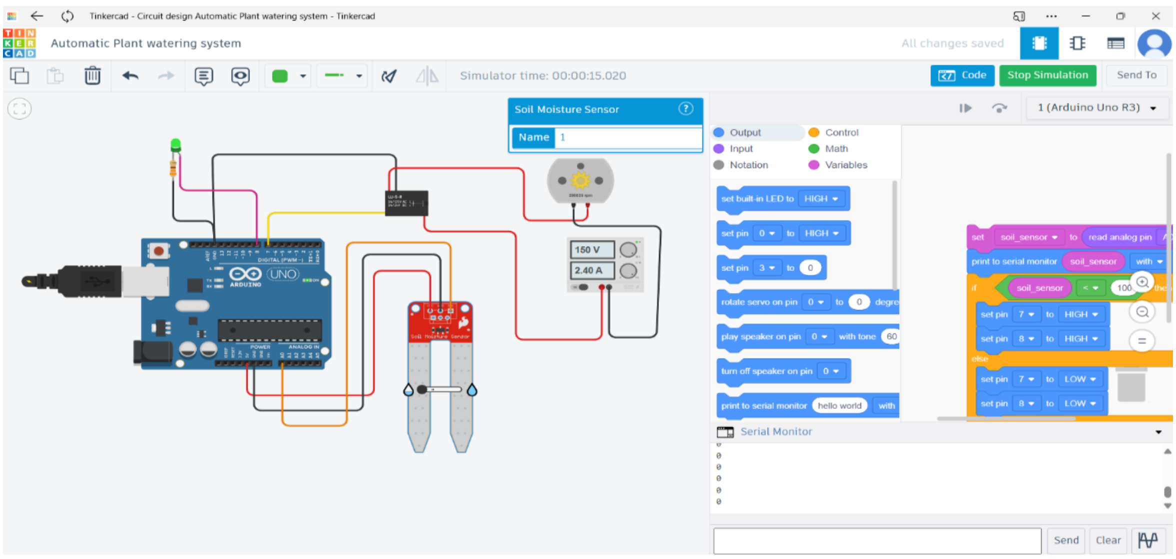

Fig:5.2.1 Results when Serial Monitor showing 0 readings

- The above image is the result when the soil requires water (i.e) moisture present in soil is zero hence it indicating that water is needed by turning on LED and the motor runs in order to water the plants/crops.
- When the serial monitor reads 0 in an automatic plant watering system using Arduino in Tinkercad, it indicates that the soil moisture sensor is detecting very low or no moisture in the soil. Here's what typically happens in this scenario:

Implications of a Reading of 0:

- **Pump Activation**:
- If your code is set to activate the pump when the moisture level falls below a certain threshold (like 400), a reading of 0 would trigger the pump to turn on. This means the system will start watering the plant.
- **Watering Duration**:
- The pump will run for the duration specified in your code (e.g., 5 seconds), allowing water to flow to the plant.
- **Soil Moisture Level**:
    - Since a reading of 0 indicates very dry soil, the system is responding correctly to the conditions and attempting to correct them by providing water.
- **Monitoring**:
    - After the watering cycle, the system will continue to monitor the soil moisture levels. The next readings will determine if the pump needs to activate again

Result-2

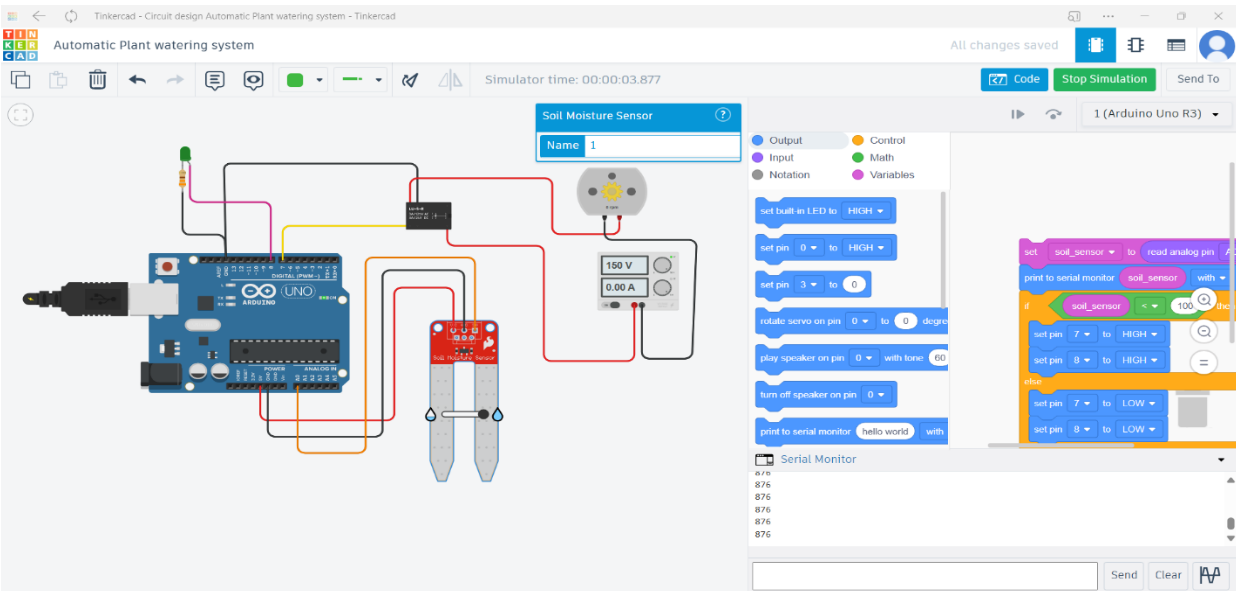

Fig:5.2.2 Results when Serial Monitor showing maximum readings

- The above image is the result when the soil requires no water (i.e) moisture present in soil is good enough hence it indicating that water is not needed. It is indicated by turning off LED and the motor turns off in order stop the water flow.

- When the serial monitor shows a maximum reading (often close to the maximum analog value of 1023 for an Arduino) in an automatic plant watering system, it indicates that the soil moisture sensor is detecting very high moisture levels. Here's what typically occurs in this scenario:

Implications of a Maximum Reading:

1. **No Pump Activation**:
    - If your code is designed to activate the pump only when the moisture level falls below a certain threshold (e.g., 400), a maximum reading would mean the pump will **not** turn on. The system recognizes that the soil is sufficiently moist.
2. **System Stability**:
    - The system remains idle, conserving water and preventing overwatering, which is crucial for plant health.
3. **Continuous Monitoring**:
    - The system will continue to monitor soil moisture levels. If the moisture level eventually drops below the defined threshold in future readings, the pump will activate as needed.

Analysis of both the cases:

Here's a result analysis of both cases in the automatic plant watering system using Arduino, based on the soil moisture sensor readings:

Case 1: Serial Monitor Reading of 0

Results:

1. **Pump Activation**:
    - The pump activates as the system responds to the dry soil condition.
    - Water is delivered to the plant, helping prevent dehydration.
2. **Soil Condition**:
    - The reading of 0 indicates that the soil is critically dry, necessitating immediate action.
3. **System Response**:
    - The system correctly identifies a need for watering, demonstrating its effectiveness in maintaining plant health.
    - After watering, the system will continue to monitor the moisture level.

Considerations:

- **Risk of Overwatering**: If the reading remains at 0 due to a faulty sensor or other issues, the pump might run longer than needed, risking overwatering.
- **Sensor Placement and Functionality**: Continuous 0 readings may indicate issues with the sensor's placement or functionality, requiring troubleshooting.

Case 2: Serial Monitor Reading at Maximum (e.g.,874)

Results:

1. **No Pump Activation**:
    - The system correctly determines that the soil is sufficiently moist and refrains from activating the pump.
    - This helps conserve water and prevent root rot or overwatering.

2. **Soil Condition**:
    - A maximum reading suggests that the soil is well-watered, indicating that recent watering was sufficient.
    - The plant is likely in a healthy moisture range.
3. **System Response**:
    - The automatic system performs as intended, maintaining optimal moisture levels without unnecessary interventions.
    - Continuous monitoring ensures that the system is ready to respond when conditions change.

Considerations:

- **Sensor Calibration**: Consistently high readings should prompt a check on sensor calibration and placement to ensure accurate moisture detection.
- **Environmental Impact**: Changes in weather (like rainfall) can affect readings; thus, the system should be adaptable to varying conditions.

Overall Analysis:

- **Effectiveness**: In both cases, the automatic plant watering system demonstrates its ability to respond to soil conditions, promoting plant health through appropriate watering.
- **System Design**: The logic implemented in the code effectively manages the watering process based on moisture readings, showcasing the importance of proper thresholds.
- **Improvements**: Future iterations could incorporate features such as:
    - Alerts for sensor malfunctions (e.g., persistent 0 or maximum readings).
    - A manual override option for user control.
    - Integration of additional sensors (e.g., temperature or light sensors) for a more holistic approach to plant care.

This analysis highlights the system's strengths and areas for potential enhancement, making it a valuable project for both learning and practical gardening applications.

Testing and Validation

Testing

Testing an Automatic Plant Watering System using Tinkercad involves several steps to ensure that the system functions as intended. Here's a general outline of the process:

1. Design the Circuit

- **Components Needed**:
    - Arduino (e.g., Arduino Uno)
    - Soil moisture sensor
    - Water pump or relay module
    - Power supply (e.g., battery or external power)
    - Jumper wires
    - Breadboard (optional for organization)

2. Build the Circuit in Tinkercad

- Create a new circuit in Tinkercad.

- Place the Arduino on the workspace.
- Connect the soil moisture sensor to the Arduino (usually to an analog pin).
- Connect the relay module or water pump to a digital pin on the Arduino.
- Ensure all components are properly wired.

3. Write the Arduino Code

- Write a simple code to read the soil moisture level and activate the pump when the soil is dry.

4.Simulate the System

- Start the simulation in Tinkercad.
- Use the moisture sensor to change its readings (you can manually adjust the analog value).
- Observe the behaviour of the pump based on the soil moisture levels.

5. Analyze Results

- Check the serial monitor for moisture readings.
- Ensure the pump activates and deactivates as expected based on the readings.
- Adjust thresholds in the code as necessary for optimal performance.

6. Make Adjustments

- Fine-tune the threshold values for moisture sensing.
- Test with different scenarios (e.g., varying moisture levels) to see how the system reacts.

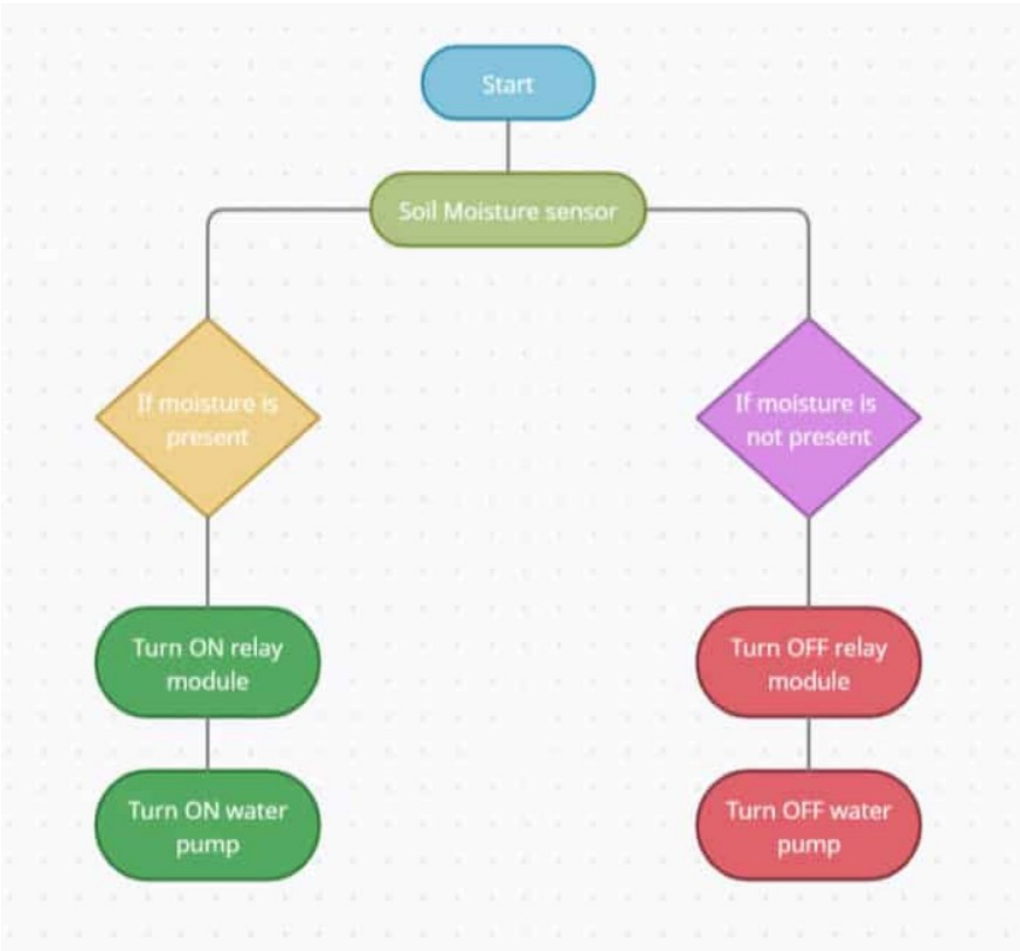

Fig:6.1.1 Block Diagram of Automatic Plant Watering System

7. Documentation

- Take notes on the testing process and results.
- Document any changes made to the circuit or code.

8. Final Thoughts

- Once satisfied with the simulation results, consider how this system could be implemented in a real-world scenario.
- Think about adding features like a display for moisture levels or a notification system for alerts.

By following these steps, you can effectively test and simulate an Automatic Plant Watering System in Tinkercad.

6.2 Validation:

Validating an Automatic Plant Watering System using Tinkercad involves checking the system's performance against defined criteria to ensure it works as intended. Here's a structured approach to the validation process:

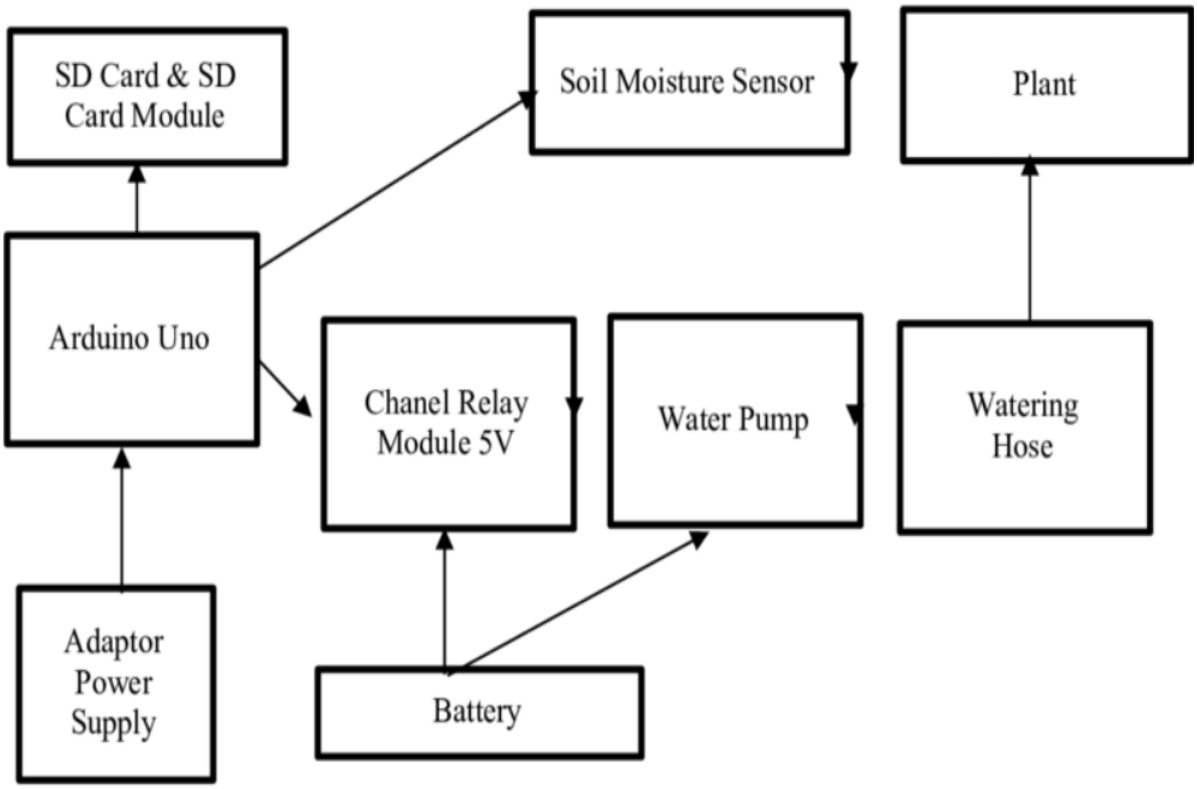

Fig:6.2.1 Validation Process

1. Define Validation Criteria

- **Functionality**: The system should accurately read soil moisture levels and control the water pump based on those readings.
- **Responsiveness**: The pump should activate promptly when the soil is dry and deactivate when moisture is adequate.
- **Threshold Accuracy**: The thresholds for activating the pump must be appropriately set based on realistic soil moisture levels.

2. Set Up Simulation Environment
   - Build the circuit in Tinkercad as previously described, ensuring all components are correctly connected.
   - Include a soil moisture sensor, pump or relay, and Arduino in the setup.
3. Conduct Initial Tests
   - **Dry Soil Test**: Simulate a low moisture level (e.g., by adjusting the sensor reading) to verify that the pump activates.
   - **Wet Soil Test**: Simulate a high moisture level to ensure the pump turns off.
   - Monitor the serial output to check the moisture readings.
4. Adjust Thresholds
   - Experiment with different threshold values in the code to find the most effective moisture levels for activation and deactivation.
   - Test each setting multiple times to confirm consistent behavior.
5. Test Different Scenarios
   - Simulate various conditions:
     - Gradually changing moisture levels to see how the system reacts.
     - Introducing scenarios where the moisture level fluctuates rapidly (to test responsiveness).
6. Monitor System Performance
   - Record the time it takes for the pump to turn on and off after moisture levels change.
   - Check for any discrepancies between the expected and actual behavior.
7. Validate System Limits
   - Identify the limits of the moisture sensor and system response. Test extreme readings to understand the boundaries of functionality.
8. Review and Document Results
   - Analyze the results against the defined criteria.
   - Document the performance, noting any issues or areas for improvement.
   - Include notes on successful tests and any anomalies observed.
9. Iterate and Refine
   - Based on the validation results, make necessary adjustments to the circuit or code.
   - Repeat the testing and validation process to ensure improvements have been effective.
10. Final Validation Check
    - After adjustments, conduct a final round of tests to confirm the system meets all validation criteria.
    - If the system performs reliably under all tested conditions, it can be considered validated.

By following this validation process, you can ensure that your Automatic Plant Watering System is not only functional but also reliable and responsive to changing soil conditions.

Conclusion

Conclusion and Future Enhancement:

Conclusion of the Project:

The conclusions of the Automatic Plant Watering System using Tinkercad highlight several key points:

1. **Automation and Efficiency**: The system successfully automates plant watering based on soil moisture levels, conserving water and reducing waste.
2. **User-Friendly Design**: It allows for easy customization to suit different plants and conditions, making it accessible for various users.
3. **Cost-Effectiveness**: Utilizes affordable components, promoting sustainable gardening practices.
4. **Simulation Advantages**: Tinkercad's simulation features enabled effective prototyping and troubleshooting, minimizing errors before physical implementation.
5. **Scalability**: The system can be expanded for larger gardens or integrated into smart home networks.

Overall, the project demonstrates the effective use of technology in enhancing plant care and sustainabilit

Future Enhancement of the system:

Future enhancements for the Automatic Plant Watering System using Tinkercad could include:

Fig:7.2.1 Advancement in Plant-watering system

1. **IoT Integration**: Connecting the system to the internet to enable remote monitoring and control through a smartphone app or web interface.
2. **Weather Forecast Integration**: Incorporating weather data to adjust watering schedules based on upcoming rain or temperature changes, optimizing water use.
3. **Multiple Plant Zones**: Expanding the system to manage multiple plants or different zones with varying watering needs using additional sensors and pumps.
4. **Data Logging**: Implementing data collection features to track soil moisture levels, temperature, and watering history, allowing for better analysis and plant care.
5. **User Notifications**: Adding alerts or notifications to inform users when the system waters plants or when maintenance is needed, such as refilling the water reservoir.
6. **Adaptive Algorithms**: Utilizing machine learning algorithms to analyze data over time and adjust watering strategies for improved efficiency based on plant responses.
7. **Solar Power Option**: Incorporating solar panels to power the system, making it more sustainable and reducing reliance on external power sources.
8. **Nutrient Dispensing**: Integrating a nutrient dispenser that can deliver fertilizers automatically alongside watering, promoting healthier plant growth.
9. **Mobile App Development**: Creating a dedicated mobile application for easier control, customization, and monitoring of the watering system.

10. **User-Friendly Interface**: Enhancing the user interface for easier setup and adjustments, possibly using touchscreen displays or voice commands.

These enhancements would not only improve the system's efficiency and user experience but also promote smarter and more sustainable gardening practices.

References

1.Abhishek Gupta, Shailesh Kumawat, and Shubham Garg, “Automatic plant watering system”, Imperial Journal of International Research (IJIR), Vol-2, Issue-4, SKIT Jaipur, 2016.

2.Yin Yin Nu, San San Lwin, Win Win Maw, “Automatic plant watering system using Arduino UNO for University Park”, International Journal of Trend in Scientific Research and Development (IJTSRD), Vol 3, 2019.

3.Manish Mayuree, Priyanka Aishwarya, A. Bagubali, “Implementation of Automatic plant watering system”, International Conference on Vision Towards Emerging Trends in Communication and Networking (ViTECoN), Vellore Institute of Technology, 2019.

4.K. Punitha, Shivaraj Sudarshan Gowda, R. Devarajnayaka, H.B. Jagadeesh Kumar, “Automated plant watering system”, International Journal of Engineering Research and Technology (IJERT), Ghousia College of Engineering, Ramanagaram, Karnataka, 2017.

5. N. S. Ishakl, Aziati H. Awang, N. N. S. Bahri, A. M. M. Zaimi, “GSM Activated Watering System Prototype”, University Teknologi MARA Shah Alam, IEEE, 2015.

6. Swapnil Bhardwaj, Saru Dhir, Madhurima Hooda. "Automatic Plant Watering System using IoT”, Second International Conference on Green Computing and Internetof Things(ICGCIoT), 2018

7.C.M. Devika, Karthika Bose, S. Vijayalekshmy, “Automatic plant Irrigation system using Arduino”, IEEE Conference on Circuits and Systems (ICCS), 29 March 2018.

8.Rajkumar Mistri, Madhupriya Kri.Singh, Eckta, “Automatic Irrigation System”, IJSART-Vol 4 Issue 5, May 2018.