Automatic Plant Watering System Using Arduino Uno

Date Varun Bhushan[1], Ghugare Sumit Dharam[2], Khedkar Sanket Shivaji[3], Prof. Sugre Dhiraj Dhanraj[4]
[1, 2, 3] Dept of Computer Engineering
[4]Guide Lecturer, Dept of Computer Engineering
[1, 2, 3, 4] Vishweshwarayya Abhiyantriki Padvika Mahavidhyalaya, Almala,Maharashtra,India.

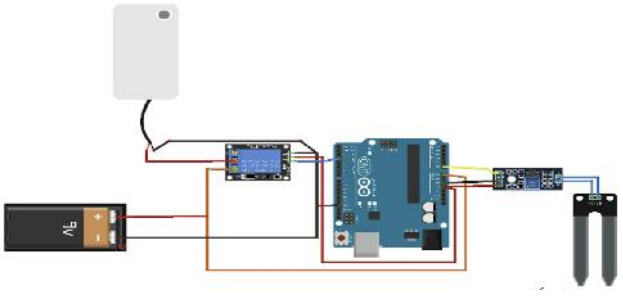
Abstract- *Plants require water to grow and thrive, and providing adequate water is essential to ensuring healthy plants. However, manually watering plants can be time-consuming, and it can be challenging to determine the right amount of water needed for each plant. The Arduino-based plant watering system is a project designed to automate the process of watering plants, making it more efficient and effective. The system utilizes an Arduino Uno microcontroller and a soil moisture sensor to sense the moisture level in the soil and water the plants whenever necessary.*

I. LITERATURE REVIEW

The Arduino-based plant watering system has gained popularity due to its ease of use, low cost, and efficient water management capabilities. Several researchers have explored the design and implementation of such systems, as well as their potential for future improvements. Overall, the studies suggest that the Arduino-based plant watering system is an effective and efficient solution for automating plant care while reducing water usage. The system's potential for future improvements includes wireless connectivity, data logging, multiple sensors, smart irrigation, and machine learning algorithms.

CONSTRUCTION

The logic of this system is very simple. In this system, the moisture sensor senses the moisture level of the soil and when the sensor senses a low moisture level it automatically switches the water pump with the help of a microcontroller and irrigates the plant. After supplying sufficient water, the soil gets retains the moisture hence automatically stopping the pump



1.1 Arduino Uno

The Arduino UNO is a standard board of Arduino. Here UNO means 'one' in Italian. It was named as UNO to label the first release of Arduino Software. It was also the first USB board released by Arduino. It is considered as the powerful board used in various projects. Arduino.cc developed the Arduino UNO board.

Arduino UNO is based on an ATmega328P microcontroller. It is easy to use compared to other boards, such as the Arduino Mega board, etc. The board consists of digital and analog Input/Output pins (I/O), shields, and other circuits.

The Arduino UNO includes 6 analog pin inputs, 14 digital pins, a USB connector, a power jack, and an ICSP (In-Circuit Serial Programming) header. It is programmed based on IDE, which stands for Integrated Development Environment. It can run on both online and offline platforms

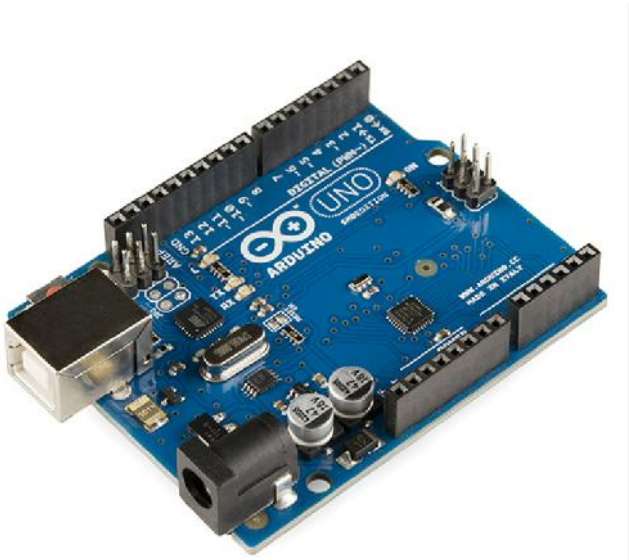


Figure 1.1: Arduino Uno

1.2 Soil Moisture Sensor

The working of the soil moisture sensor is very easy to understand. It has 2 probes with exposed contacts that act like a variable resistor whose resistance varies according to the water content in the soil. This resistance is inversely

proportional to the soil moisture which means that higher water in the soil means better conductivity and hence a lower resistance. While the lower water in the soil means poor conductivity and will result in higher resistance. The sensor produces an analogue voltage output according to the resistance. The sensor comes with an electronic module that connects the probe to the Arduino. The module has an LM393 High Precision Comparator which converts the analog signal to a Digital Output which is fed to the microcontroller. We have covered an in-depth Arduino soil moisture sensor tutorial which covers the working of soil moisture sensor module and how to use it with the Arduino. You can check the tutorial if you want to learn more about the soil moisture sensor.

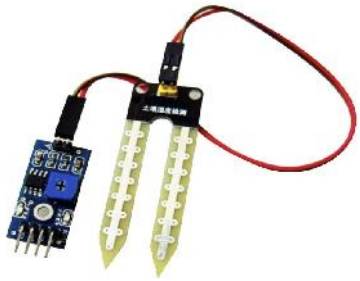


Figure 1.2: Soil Moisture Sensor

1.3 Pump

A water pump that operates at 12V is a type of pump that is designed to run on a 12-volt DC power supply. These types of pumps are commonly used in automotive, marine, and RV applications, as well as in small scale irrigation, water treatment and other similar systems.

One of the advantages of using a 12V water pump is that it can be powered by a car battery or a deep-cycle battery, which makes it portable and convenient for use in remote locations. Additionally, these pumps are relatively low power and energy efficient, which makes them ideal for use in solar-powered systems.

Figure 1.3: Pump

1.4 **Relay**

A relay is an electronic switch that can be controlled by an Arduino microcontroller to turn a water pump on and off. When the relay is energized, it closes an electrical circuit, allowing electricity to flow through the pump and turn it on. When the relay is de-energized, it opens the circuit, cutting off the electricity and turning the pump off.

To use a relay to control a water pump with an Arduino, you will need to connect the relay to the Arduino ‘s digital output pins. The relay typically has three pins, VCC, GND, and IN. VCC should be connected to 5V of the Arduino, GND to GND, and IN to a digital pin of the Arduino. Once the connections are made, you can use the Arduino ‘s digital Write function to turn the relay on and off, and thus control the water pump.

Figure 1.4: Relay

1.5 Battery 9V

A 9V battery is a type of primary (non-rechargeable) battery that provides a nominal voltage of 9 volts. It is commonly used in portable devices such as smoke detectors, remote controls, and small electronic toys. 9V batteries are also used in some low-power projects such as small robots and simple electronic circuits.

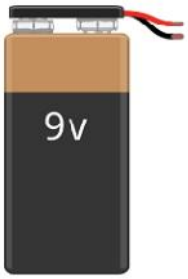


Figure 1.5: Battery 9V

II. SOURCE CODE

```
int moisture_sensor = A0;
int moisture;
int limit = 40;
int WATERPUMP = 13;
int sets;
void setup()
{
Serial.begin(9600); // opens serial port, sets data rate to 9600 bps;
pinMode(moisture_sensor,INPUT);
pinMode(WATERPUMP,OUTPUT);
}
void loop()
{
moisture = analogRead(moisture_sensor);
moisture=map(moisture,550,0,0,100);
Serial.print("Moisture=");
Serial.print(moisture);
Serial.println("%");
if (Serial.available());
{
int speed= Serial.parseInt();
{
analogWrite(WATERPUMP,speed)
;
}
if (moisture>limit)
{
digitalWrite(13,LOW);
}
else
digitalWrite(13,HIGH);
}
delay(400);
}
```

III. RESULT, CONCLUSION AND FUTURE SCOPE

1.Result

1. Improved plant growth: With a consistent and appropriate supply of water, plants are more likely to grow faster and healthier.
2. Water conservation: An automatic watering system can help to conserve water by only delivering the necessary amount of water to the plants and avoiding wastage.
3. Time-saving: The system can save time for the user by eliminating the need for manual watering and monitoring.
4. Customization: The system can be customized to meet specific plant watering requirements, such as watering frequency and volume, based on factors like plant species, soil type, and climate.
5. Increased efficiency: The system can be designed to operate at specific times of the day or when certain conditions are met, making it more efficient than manual watering.

2. Conclusion

the Arduino-based plant watering system is a useful project that can automate the process of watering plants.
The system design involves a moisture sensor, microcontroller, water supply mechanism, power supply, and user interface. The system architecture connects these components together to create a functional system, and the implementation involves programming the microcontroller, designing the user interface, and assembling the water supply mechanism.

This project provides an efficient and effective solution to ensure plants receive water only when they need it, making it an eco-friendly and cost-effective alternative to manual watering.

3. Future Scope

1. Wireless Connectivity: The system can be enhanced with wireless connectivity such as WiFi or Bluetooth to allow for remote monitoring and control. This would enable users to monitor the moisture level and water their plants from anywhere with internet connection.
2. Data Logging: The system can be improved by adding data logging capabilities to track the moisture level over time. This data can be used to analyze plant growth patterns and optimize watering schedules.
3. 4Multiple Sensors: The system can be expanded by adding multiple moisture sensors to monitor different plants or sections of a garden. This would enable users to create custom watering schedules for each plant based on its individual needs.
4. Smart Irrigation: The system can be integrated with weather forecast data and other environmental factors such as temperature and humidity to create a smart irrigation system. This would enable the system to adjust watering schedules based on the weather forecast and reduce water waste.

REFERENCES

[1] https://components101.com/ microcontrollers/ arduino-uno
[2] http://www.circuitbasics.com/setting-up-a-5v-relay on-the-arduino/
[3] https://vigyanashram.files.wordpress.com/ 2015/05/plant-watering-system.pdf
[4] https://learn.sparkfun.com/tutorials/installing arduino-ide/all.pdf
[5] [14 Dez] Devika et al., International Journal of Advanced Research in Computer Science and Software Engineering 4(10), October - 2014, pp. 449-456
[6] [16 San] Sandeep K. Shukla, IIT Kanpur, Introduction to embedded system, Aug 29, 2016
[7] [16 SHA] SHAIKH SHEROZ MOHD HASAN, 'AUTO IRRIGATION USING ARDUINIO'2016
[8] [17 N.Du] N. Đuzić and D. Đumić: Automatic Plant Watering System and its Applications, Coll.Antropol. 41 (2017)