

**KHULNA UNIVERSITY OF ENGINEERING & TECHNOLOGY**

**Department of Computer Science and Engineering**

**Course No: CSE 3104**

**Course Title: Peripherals and Interfacing Laboratory**

**Project Title**

**Automated Plant Watering System**

**Submitted by**

**Mohammad Abtahe Alam**

**Roll: 1907103**

**Md. Rakibul Hasan Adnan**

**Roll: 1907106**

**Submitted to**

|  |  |
| --- | --- |
| **Md. Badiuzzaman Shuvo** | **Md. Repon Islam** |
| **Lecturer**  **Department of Computer Science and Engineering(CSE)**  **Khulna University of Engineering & Technology (KUET)** | **Lecturer**  **Department of Computer Science and Engineering (CSE)**  **Khulna University of Engineering & Technology (KUET)** |

**Objectives**

The objectives of this project is to design and implement an automated plant watering system using Arduino. The system should be able to monitor the moisture level in the soil,the humidity of the atmosphere and provide water to the plants when necessary. The specific goals are as follows:

1. To develop a moisture sensor to measure the soil moisture accurately.
2. To design a control circuit to control the water pump based on the moisture level and the humidity level of the atmosphere.
3. Implement the system using Arduino and validate it’s functionality.
4. Evaluate the effectiveness of the system in maintaining optimal soil moisture level.

**Introduction**

Watering plants is essential for their growth and survival, but it can be a challenging task to manually monitor and water plants regularly, especially in large gardens or during vacations. An automated plant watering system can offer a convenient solution to this problem. This system utilizes an Arduino microcontroller, a moisture sensor, a humidity sensor,a water pump and a motor driver to automate the watering process.

The moisture sensor measures the moisture content in the soil and the humidity sensor detects the humidity level in the atmosphere. When the moisture level and humidity level fall below a certain threshold, the Arduino activates the water pump, providing water to the plants. Once the moisture level reaches the desired level, the pump is turned off. This ensures that the plants receive adequate water without the need for constant human intervention.

When the air is drier (humidity lower), the plant’s leaves lose more moisture (the process called transpiration) because of the simple physics of osmosis (moisture travels easily from an area of greater concentration - inside the leaf - to an area of less concentration - the open air.) Therefore, plants in low humidity will need to absorb more water through their roots to compensate for water loss through their leaves. But when the humidity is high , plants can’t evaporate extra water through there leaf stomata. At that moment watering plants is harmful for a plant’s growth.

**Apparatus Required**

|  |  |  |
| --- | --- | --- |
| Serial No. | Apparatus Required | Quantity |
| 1 | Arduino Uno | 01 |
| 2 | 5 volt Relay Motor | 01 |
| 3 | Submersible Water Pump | 01 |
| 4 | Soil Moisture Sensor | 02 |
| 5 | Breadboard | 02 |
| 6 | Motor Driver | 01 |
| 7 | Water Level Sensor | 01 |
| 8 | Humidity Sensor | 01 |
| 9 | Rechargable Li-ion Battery | 02 |
| 10 | Piezo Buzzer | 01 |
| 11 | Jumper Wires | As Required |
| 12 | LCD Display | 01 |
| 13 | Plant | 01 |
| 14 | Tube/pipe | As Required |
| 15 | Relay Switch | 01 |

**Description**

*Arduino Uno*

The Arduino uno is a microcontroller which is based on the ATmega328 datasheet. It has 14 digital inputs /output pins. It is an open source microcontroller which is used to control relay, simply connect to a computer with a USB cable or power it with a AC-to-DC adapter or battery to get started. It is large assortment of included libraries for interfacing to wide range of hardware. The Uno differs from all preceding boards in that it does not use the FTDI USB-to-serial driver chip.

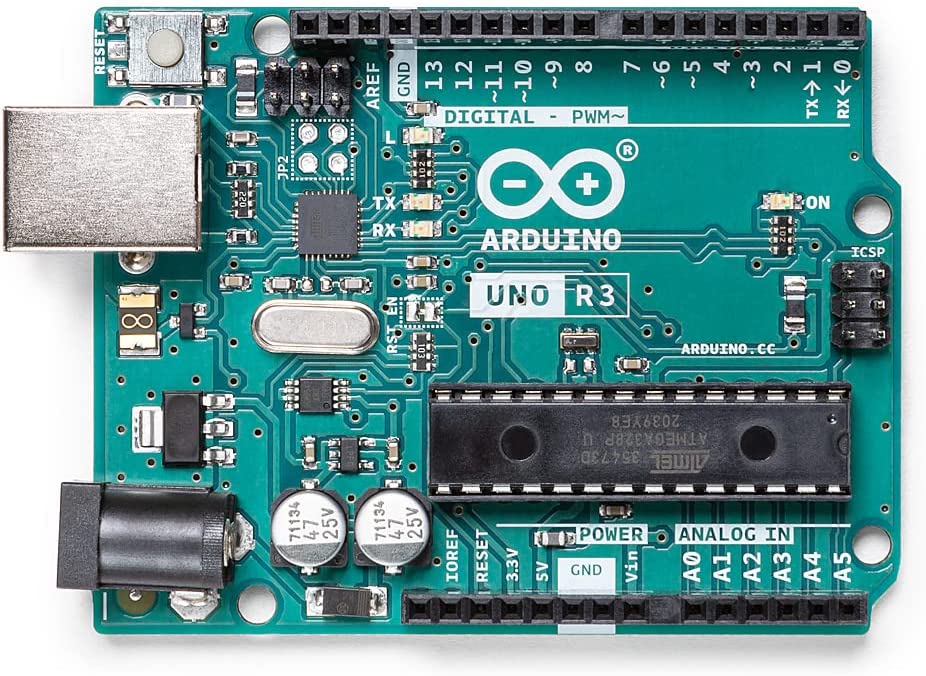


Figure 1.1 : Arduino Uno

*Relay*

One-channel relay board which operates on 5-6V is used here. The relay board consists of three pins which are normally open (NO), normally closed (NC) and common (C). The common pin is connected to NC pin when the relay is off and to the NO pin when the relay is on. The input pin receives logic high from Arduino Uno and in turn switches on the relay, thus common are connected to NO which turns the device on till the relay is on. The “VCC” and “GND” pins of the relay are connected to 5V supply and ground respectively.

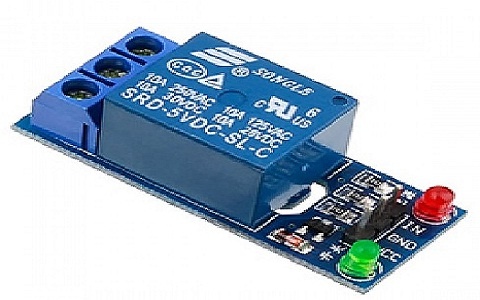


Figure 1.2 : Relay switch

*Water Pump*

The water pump is used to artificially supply water for a particular task. It can be electronically controlled by interfacing it to a microcontroller. It can be triggered ON/OFF by sending signals as required. The process of artificially supplying water is known as pumping. There are many varieties of water pumps used. This project employs the use of a submersible water pump which is connected to power supply through relay.



Figure 1.3 : Submersible Water pump

*Soil Moisture Sensor*

A soil moisture sensor measures the water content in soil by measuring the dielectric permittivity of the soil as a function of water content. The volumetric water content is measured by the soil moisture sensor indirectly by properties like electrical resistance and dielectric constant. Using this we can reduce manpower, save water to improve production and gravimetric method.

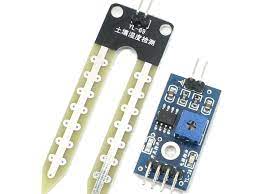


Figure 1.4 : Soil Moisture Sensor and Module

*Water sensor*

The sensor has a series of ten exposed copper traces, five of which are power traces and five are sense traces. These traces are interlaced so that there is one sense trace between every two power traces. Usually these traces are not connected but are bridged by water when submerged. There’s a Power LED on the board which will light up when the board is powered.



Figure 1.5 : Water Level Sensor

*Piezo Buzzer*

Piezo buzzer is used to generate sound, beep or even melody of a song.



Figure 1.6 : Piezo Buzzer

*Humidity Sensor*

The DHT22 is the more expensive version which obviously has better specifications. Its temperature measuring range is from -40°C to +125°C with +-0.5 degrees accuracy, while the DHT11 temperature range is from 0°C to 50°C with +-2 degrees accuracy. Also the DHT22 sensor has better humidity measuring range, from 0 to 100% with 2-5% accuracy, while the DHT11 humidity range is from 20 to 80% with 5% accuracy.

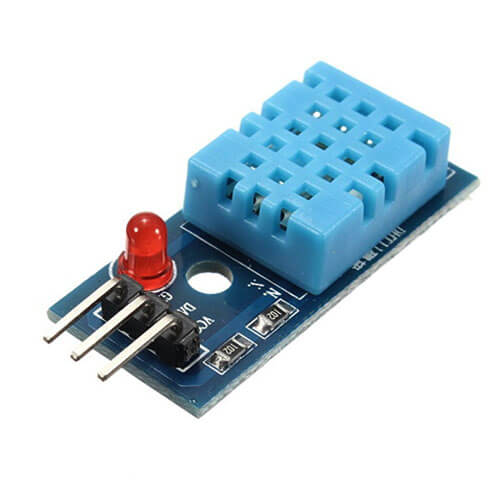


Figure 1.7 : Humidity Sensor

*Motor Driver*

This motor driver can control the speed of a motor. This driver is used to control the speed of the water pump.

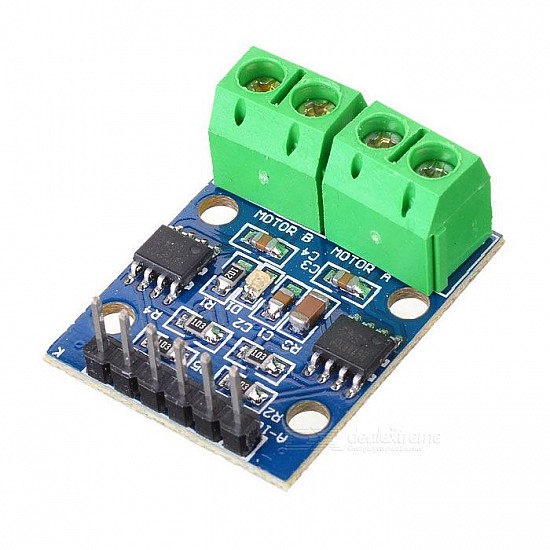


Figure 1.7 : Motor Driver

**Circuit Diagram**

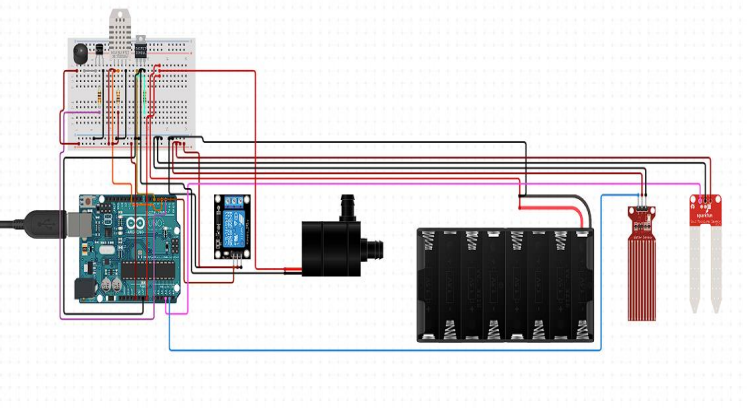


Fig 1.8 : Circuit Diagram of Automated Plant Watering System.

**Flow Chart**

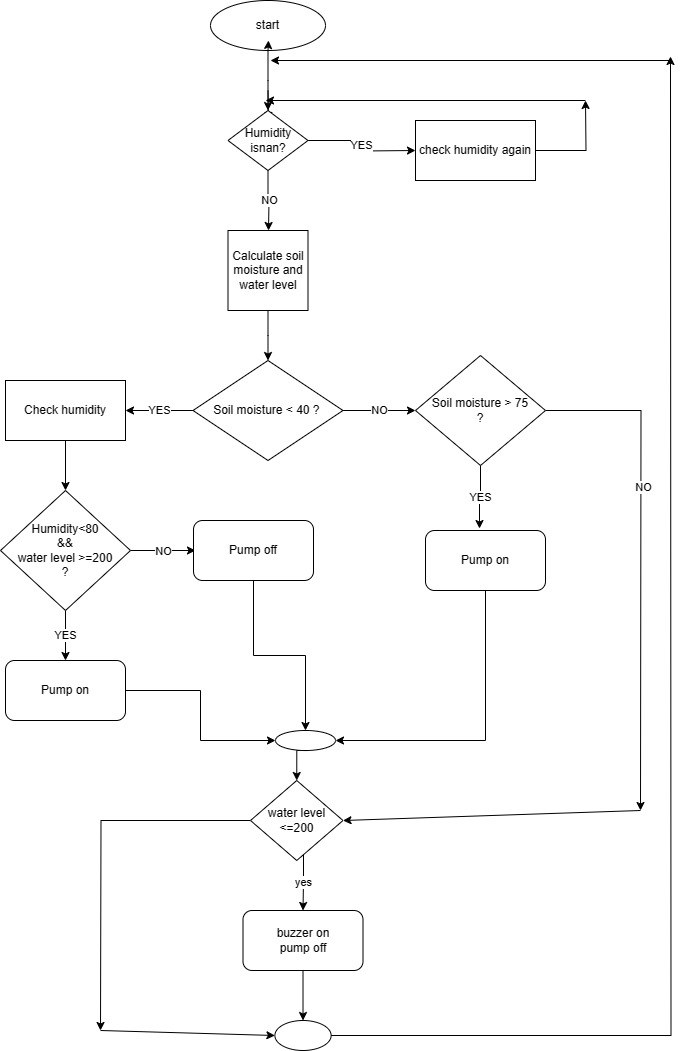
****

Fig 1.9 : Flowchart of Automated Plant Watering System.

**Pseudo Code**

INITIALIZE moisture\_sensor1\_pin\_no,moisture\_sensor2\_pin\_no, pump\_pin, waterlevel\_sensor\_pin , lcd, humidity\_sensor\_pin,buzzer\_pin;

Function setup

pinMode output in pump\_pin;

pinMode output in buzzer\_pin;

EndFunction

Function loop

humidity 🡨 Read from humidity\_sensor\_pin;

waterlevel 🡨 read from waterlevel\_sensor\_pin;

IF humidiy is not a number THEN // isnan of humidity is true

print humidity;

ELSE

moisture\_sensor1\_value 🡨 Read from moisture\_sensor1\_value;

moisture\_sensor2\_value 🡨 Read from moisture\_sensor2\_value;

IF moisture\_sensor1\_value > 800 or moisture\_sensor2\_value>800 THEN

IF humidity < 100 and waterlevel>=200 THEN

pump\_pin 🡨 HIGH;

ELSE

pump\_pin 🡨 LOW;

ENDIF

ELSE IF moisture\_sensor1\_value < 400 and moisture\_sensor2\_value <400 THEN

pump\_pin 🡨 LOW;

ENDIF

IF waterlevel <= 150 THEN

buzzer\_pin 🡨 HIGH;

ENDIF

DELAY for 1000ms;

ENDIF

EndFunction

**Methodology**

We have connected each instrument to the Arduino Uno board. The first step is to power on the humidity sensor and measure the humidity level. If there is humidity present, the Arduino will proceed to gather soil moisture information from two soil moisture sensors, namely Sensor 1 and Sensor 2.

If the soil moisture level is greater than 75%, the Arduino will then check the humidity level. If the humidity exceeds 100%, it will cut off the relay. Alternatively, if the humidity is within the acceptable range, the Arduino will proceed to check the water level.

In the case of a high water level, the Arduino will activate the pump to supply water to the soil. However, if the water level is low, a buzzer will produce an audible sound to alert the user and the pump will stop pumping water.

There is also a motor driver which is used to control the speed of the motor manually.

To facilitate this functionality, we have connected the common port and normally closed (NC) port of the relay to the battery and pump. As a result, when the relay's input is set to low, it will activate and turn on the pump.

**Conclusion and Discussion**

From this work, we can control the moisture content of the soil of cultivated land. According to soil moisture, water pumping motor turned on or off via the relay automatically. This saves water, while the water level can be obtained in a preferred aspect of the plant, thereby increasing productivity of crops. The system also allows the delivery to the plant when needed based on the type of plant, soil moisture, and observed temperature. The proposed work minimizes the efforts of major agricultural regions. Many aspects of the system can be customized and used software to fine-tune the requirements of the plant. The result is a scalable, supporting technology. Using this sensor, we can see that the soil is wet or dry. If it is dry, the motor will automatically start pumping water. Also using the water sensor, we can identify the level of the water. If it is low then we can turn on a buzzer or if it is high then we can supply the water to the field.

It was observed that the proposed methodology controls the moisture content of the soil of cultivated land. The motor automatically starts pumping water if the soil is dry and humidity is below a threshold value and need water and stops when the moisture content of the soil is maintained as required.

**References**

1. docs.arduino.cc
2. www.tutorialspoint.com