WOLDIA UNIVERSITY INSTITUTION OF TECHNOLOGY

SCHOOL OF ELECTRICAL AND COMPUTER ENGINEERING

COMPUTER STREEM

IOT BASED AUTOMATED DRIP IRREGATION BASED ON ARDUINO FOR WOLDIA UNIVERSITY

Group members

Name Id No

1. Samuel Belay...1204655
2. Idris Mohamammed..1201151
3. Bitanya Mulugeta..1204628

Advisor Name: Mr.Bimrilign

Submission Date: 26/07/2015 E C

Woldia, Ethiopia,

DECLARATION

We are student at Woldia University Institution of technology (WIT), school of Electrical and Computer Engineering. The information found in this project is our original work. Moreover, all sources of materials that we used for our project is fully acknowledged.

Name	signature
1. Samuel Belay	________________
2. Indris Mohammed	________________
3. Bitaniya Mulugeta	________________

Date of submission: This project has been submitted for examiner with our approval as a university advisor.

Advisor name	signature	date
1____________________	________________	_____________

Examiner name	signature	date
1_____________________	_________________	_______________

ACKNOWLEDGMENT

First and foremost, we would like to thank the God Almighty who blessed and supported us in every step of our journey and entitled us for this opportunity.

We would like also thank Woldia University, The staff members of School of Electrical and Computer Engineering for their cooperation and assistance during our study. And we would like to thanks our advisor Mr. Bimrlgn, This project would not be successfully completed without the supervision and critical leading of our advisor.

We would like to thank our families, friends and are second only to God, in creating good working environment by sharing our other responsibilities.

ABSTRACT

The Drip Irrigation System project at Woldia University employs a high level of abstraction to create a comprehensive and adaptable solution for the university's irrigation needs. The system uses a variety of sensors, including soil moisture, ultrasonic, rain, pH, and humidity sensors, which are abstracted as input providers that deliver real-time data about environmental conditions to the control system. The control system acts as the central decision-making unit, processing the sensor data and determining the appropriate irrigation schedule and water distribution. This abstraction allows the control system to be easily updated or replaced without affecting the rest of the system. The system includes actuators such as servo motors and pump motors to control the water flow and distribution, which are abstracted as output devices. The system also includes data tracking and analysis capabilities to monitor the irrigation performance and identify optimization opportunities. By employing these levels of abstraction, the Drip Irrigation System project achieves a highly modular and scalable design, allowing the system to adapt to changes in environmental conditions, technological advancements, or operational requirements without a complete system overhaul.

Automated drip irrigation systems have gained popularity in recent years due to their efficiency in delivering water directly to the roots of plants, resulting in water savings and improved plant growth. This abstract will explore the benefits and challenges of automated drip irrigation systems, including their ability to provide precise and consistent watering, reduce water waste, and promote healthier plants. The abstract will also discuss the various components of automated drip irrigation systems, such as timers, valves, and emitters, and how they work together to create an efficient watering system. Additionally, the abstract will highlight the potential environmental impact of automated drip irrigation systems, such as reduced water usage and improved crop yields. ADI, which stands for Automated Drip Irrigation, is a modern irrigation system that delivers water directly to the roots of plants in a controlled and efficient manner, helping to conserve water by reducing evaporation and runoff while ensuring plants receive the right amount of water at the right time, providing a sustainable and efficient solution for landscapes and agriculture.

Table Contents

LIST OF FIGURES

LIST OF TABLES

LIST OF ABRIVIATION

Abbreviation	description
AC	Alternative Current
ADC	Analog To Digital Converter
CO	Change Over
DAC	Digital To Analog Converter
DBDT	Double Pole Double Throw
DC	Direct Current
EMF	Electro Motive Force
GDP	Gross Domestic Product
GLD	Ground
GSM	Global System For Mobile Communication
IDE	Integrated Development Environment
IOT	Internet Of Things
LCD	Liquid Crystal Display
LED	Light Emitting Diode
NC	Normally Closed
NO	Normally Open
PC	Personal Computer
SMS	Short Message Service
SPST	Single Pole Single Throw

CHAPTER 1

INTRODUCTION

1.1 INTRODUCTION

Drip or trickle irrigation refers to the frequent application of small quantities of water at low flow rates and pressures. Automated drip irrigation is a cutting-edge method of delivering water directly tothe roots of plants in a controlled and automated fashion. Unlike traditional irrigation systems that may lead to water wastage and inconsistent watering, automated drip systems offer a more targeted and efficient approach to watering plants. Rather than irrigating the entire field surface, as with sprinklers, drip irrigation is capable of delivering water precisely at the plant where nearly all of the water can be used for plant growth. The uniformity of application is not affected by wind because the water is applied at or below the ground surface. Rather than traditional irrigation system, automated drip irrigation systems minimize evaporation and runoff, conserving water and promoting efficient water usage. A well designed and maintained drip irrigation system is capable of an application efficiency of 90 percent. According to the Farm and Ranch Irrigation Survey conducted by USDA, 11,239 acres of agricultural lands in Oklahoma were under drip irrigation in 2007, out of which 81 percent was under SDI.

1.2 BACK GROUND OF THE STUDY

In Ethiopia, agriculture is the primary and most important economic sector. Plants are highly dependent on irrigation systems to achieve good yields. Different plants require varying amounts of water. Woldia, like many regions in Ethiopia, has a long history of traditional irrigation practices, including techniques like furrow irrigation, basin irrigation, and the use of diversion canals. The region's agricultural practices have been shaped by its topography, climate, and water availability, with farmers developing indigenous irrigation systems to support crop cultivation. The automated drip irrigation system can help farmers make informed decisions on utilizing the optimal amount of water.

The main issue in an irrigation system is its control management, which is crucial for monitoring crop growth and preventing system damage and failure. One of the solutions to improve the control aspect is the application of an automated drip irrigation system.

The Drip Irrigation System for Woldia University will utilize key components to create a fully automated and intelligent irrigation system. Sensors, including soil moisture, ultrasonic, rain, pH, and humidity, will continuously monitor the environmental conditions and provide real-time data to the system. This data will be processed by a central control unit (microcontroller), which will then make decisions on the appropriate irrigation schedule and water distribution. The servo motor will control the water valves to ensure the correct amount of water is delivered to each irrigated area, while the pump motor will ensure the proper water flow.

In the future, the project aims to reduce the physical efforts faced by farmers by allowing them to remotely check the diesel and water levels, flow rate, and automatically turn the pump on and off, as well as receive field condition updates through a mobile phone. The goal is to develop an irrigation system that continuously monitors the availability of electricity to operate the pump, the water level in the reservoir and the water pressure and sends alert messages to the authorized person via a GSM module and via Blynk application to the user whenever a system failure occurs, such as the pump stopping due to a power outage or a lack of water in the reservoir.

The system will be designed to be easy to monitor and maintain, with user-friendly interfaces and automated alerts to notify the relevant personnel of any issues or deviations from the optimal operating parameters.

1.3 STATEMENT OF THE PROBLEM

The world's growing population has led to an exponential increase in food demand, necessitating the cultivation of more land. With changing weather patterns due to global warming, irrigation has become the only reliable method for crop production. As more land is being brought under irrigation, there is a pressing need for the optimal use of water resources.

In recent years, advancements in electronics and computational technology have been leveraged to address these challenges. Microcontrollers, combined with various sensors, can be used to measure and control physical quantities like temperature and moisture, enabling the development of automated systems. Irrigation systems in crop production have also been automated, helping to address the unreliability of climate changes and the need for water optimization.

Woldia University, located in the Amhara region of Ethiopia, is facing challenges with its current manual irrigation system. These include excessive water consumption, uneven water distribution, lack of responsiveness to environmental conditions, high labor intensity, and insufficient data tracking and analysis. To address these issues, the university has identified the need for a technologically advanced, automated drip irrigation system that can intelligently adapt to changing environmental conditions and optimize water usage.

The key significance of this project is:

- Woldia University is located in an area that may face water scarcity or irregular rainfall patterns, making this project a proactive approach to water conservation.
- The project provides an opportunity for research, development, and innovation in agricultural technologies, integrating cutting-edge sensors, data analytics, and automated control systems.
- The project can serve as a demonstration of best practices in water management and sustainable agriculture, positioning the university as a role model for the local farming community.

By implementing this Drip Irrigation System, Woldia University aims to enhance its water conservation efforts, improve the health and productivity of its greenery and agricultural areas, and contribute to the overall sustainability of the campus environment. This will increase yields, improve crop quality, reduce weed growth, minimize environmental impact, and most importantly, conserve water, leading to cost savings.

1.4 OBJECTIVES OF THE PROJECT

1.4.1 General Objective

- To “Design and develop automated drip irrigation system using Arduino microcontroller for Woldia university”

1.4.2 Specific Objectives

- To measure the moisture content of the soil
- To measure analyses the humidity and temperature of the surrounding environment
- To measure the amount of the water in the tank
- To detect the rain on the surrounding
- To design simple step down transformer, ac to dc convertor device
- To design water pump for automatically watering field
- To measure and analyze the soil PH value
- Construct schematic diagram and simulate the overall system using proteus professional

1.5 MOTIVATION

The motivation come for these project is that, for the drip irrigation that is being done now project introduced our role as an electrical student to make new automated drip irrigation is designed to be a learning experience for campus students.

This project success result in a sustainable- agricultural practice that benefits the environment and creates increased crop yields.

1.6 SCOPE OF THE PROJECT

Our comprehensive services for the automated drip irrigation system at Woldia University include:

- Measuring the moisture content in the soil to determine the optimal watering needs.
- Detecting rainfall to adjust the irrigation schedule accordingly.
- Monitoring the amount of water available in the tank to ensure sufficient supply.
- Determining the pH level of the soil to optimize nutrient availability.
- Analyzing and monitoring the temperature and humidity of the environment to fine-tune the irrigation parameters.

We specialize in sizing and selecting the appropriate water pumps and power supply for the components, ensuring optimal performance and efficiency. Our team can also assist in the design of mobile controls using IOT technology to remotely manage the drip irrigation system.

Additionally, we excel in designing and simulating the drip irrigation system using Proteus, a powerful software for comprehensive system testing and optimization. This allows us to thoroughly evaluate the system's performance and make any necessary adjustments before implementation.

With our expertise in these areas, we provide end-to-end solutions for efficient and effective drip irrigation management at Woldia University.

The system design also addresses the needs of individuals who want to access the existing irrigation infrastructure at the university, ensuring that the automated system can be integrated seamlessly with the existing infrastructure.

1.7 FEASIBILITY ANALYSIS OF THE PROJECT

1.7.1 Operational feasibility

Assessing the operational feasibility of implementing automated drip irrigation involves a comprehensive evaluation of costs, technical requirements, water management efficiency, maintenance considerations, adaptability, scalability, and environmental sustainability. By conducting a thorough analysis of these factors, organizations can determine whether automated drip irrigation is a viable and effective solution for their agricultural operations.

Talking about the operational feasibility of automated drip irrigation systems involves assessing whether implementing such a system is practical, beneficial, and sustainable within a specific context.

Key aspects to consider for Operational Feasibility of Automated Drip Irrigation:

1. **Cost Analysis:**
   - Initial Investment: Evaluate the cost of setting up an automated drip irrigation system,including equipment, installation, and system integration costs.
   - Operational Costs: Consider ongoing expenses such as maintenance, repairs, energy requirements, and monitoring tools.
2. **Technical Considerations:**
   - Compatibility: Assess if the automated system aligns with existing infrastructure, crop types,and water sources.
   - Reliability: Ensure the system components are durable, have backup measures in place, andcan operate effectively under varying environmental conditions.
3. **Water Management:**
   - Efficiency: Determine if the system effectively optimizes water usage, minimizes wastage,and provides the right amount of water to plants based on their needs.
   - Water Source: Consider the availability of water sources and the suitability of water qualityfor irrigation purposes.
4. **Maintenance and Support:**
   - Skill Requirements: Evaluate if the organization has or can acquire the necessary technicalskills to maintain, troubleshoot, and repair the automated system.
   - Technical Support: Identify if there are available support services or vendors that can provideassistance when needed.
5. **Adaptability and Scalability:**
   - Scalability: Determine if the system can be easily scaled up or down based on changing needs,expansion of cultivation areas, or crop diversification.

- Integration: Assess if the system can be integrated with other agricultural technologies,sensors, or management systems for improved efficiency.

6. **Environmental Impact:**

- Sustainability: Consider the environmental impact of the automated drip system in terms ofwater conservation, energy use, and contribution to sustainable agricultural practices.
- Compliance: Ensure that the system adheres to local regulations, environmental standards,and water usage guidelines

1.7.2 Economic feasibility

Considering the economic feasibility of implementing automated drip irrigation systems is essentialto assess the financial viability and potential returns on investment.

Economic Feasibility of Automated Drip Irrigation:

1. **Market analysis**

- Drip irrigation systems have a growing market demand due to their water efficiency,especially in regions facing water scarcity or where water conservation is critical.
- Market availability of quality components, local expertise for installation, and accessibilityto maintenance services can impact the feasibility of adopting drip irrigation.

2. **Operational Cost Savings:**

- Water Efficiency: Determine the potential reduction in water consumption and associated costs by implementing precise and targeted irrigation methods offered by automated drip systems.
- Labor Savings: Assess the labor costs saved through automation, reduced manual irrigation efforts, and streamlined maintenance procedures.

3. **Maintenance Expenses:**

- System Upkeep: Evaluate the ongoing maintenance costs, including sensor calibration, equipment servicing, software updates, and any additional maintenance requirements to ensure the continuous operation of the system.

- Cost-Benefit of Maintenance: Compare the maintenance expenses with the expected benefits in terms of water savings, crop quality improvement, and resource optimization.

4. **Risk Assessment and Contingency Planning:**

- Risk Mitigation: Identify potential risks that may impact the economic feasibility of the automated system, such as crop failure, equipment breakdowns, or market fluctuations, and develop contingency plans to mitigate these risks.
- Sensitivity Analysis: Conduct sensitivity analyses to assess how changes in key variables, such as crop prices, water costs, or energy prices, may impact the financial outcomes of implementing automated drip irrigation.

Generally, evaluating the economic feasibility of automated drip irrigation involves analyzing costs, expected savings, revenue opportunities, maintenance expenses, incentives, risks, and potential financial returns associated with implementing the system. By conducting a comprehensive economic analysis, organizations can make informed decisions regarding the adoption of automated drip irrigation to achieve cost-effective water management, enhance agricultural productivity, and drive sustainable growth.

1.7.3 Technical feasibility

When considering the technical feasibility of automated drip irrigation systems, several key factors come into play to determine whether such a system is viable and practical for implementation.

Technical Feasibility of Automated Drip Irrigation:

1. Sensor Integration

- Compatibility: Assess the compatibility of temperature, humidity, and water level sensorswith the irrigation system and control mechanisms.
- Data Accuracy: Ensure that sensor data is reliable, accurate, and consistent for makinginformed irrigation decisions.

2. Automation System

- Control Mechanism: Evaluate the efficiency of the automated system in regulating waterflow, pressure, and distribution to different zones.
- Programming: Assess the ease of programming and customization options for adjustingirrigation schedules based on sensor data.

3. Water Distribution

- Uniformity: Determine the system's capability to ensure uniform water distribution acrossthe cultivation area to meet the specific needs of different plant types.
- Pressure Regulation: Evaluate the system's ability to maintain consistent water pressure foroptimal irrigation efficiency.

4. System Reliability

- Fault Tolerance: Consider the system's resilience to failures, ability to detect malfunctions,and mechanisms for fallback measures to ensure uninterrupted irrigation.
- GSM and IOT Monitoring: Assess the system's GSM monitoring capabilities for real-timeassessment of irrigation operations and potential troubleshooting.

5. Infrastructure Requirements

- Compatibility: Ensure that the current infrastructure, such as water sources, pipelines, andpower supply, can support the automated drip irrigation system.
- Scalability: Evaluate the feasibility of integrating the automated system with existingirrigation infrastructure for scalability and expansion.

6. Power Supply and Energy Efficiency

- Power Source: Determine the energy requirements of the automated system and assess the availability of a reliable power source, such as mains electricity or alternative energy solutions.
- Energy Optimization: Consider energy-efficient components, power-saving features, and potential renewable energy options to minimize operational costs.

Assessing the technical feasibility of automated drip irrigation systems involves examining sensor integration, automation mechanisms, water distribution efficiency, system reliability, infrastructure compatibility, and energy efficiency. By evaluating these technical aspects thoroughly, organizations can determine the viability of implementing automated drip irrigation to enhance water management, optimize crop growth, and promote sustainable agricultural practices.

1.8 MAJOR ASSUMPTIONS AND CONSTRAINTS MADE FOR THE WORK/PROJECT

In the project planning and implementation of automated drip irrigation at Woldia University, several major assumptions are made to guide decision-making, resource allocation, and project execution.

Those are some key assumptions that underpin the project work:

1. **Water Availability and Quality:**
   - Assumption: A reliable water source with adequate quantity and quality is accessible for irrigation purposes on the university's agricultural land.
   - Implication: Water availability is crucial for the successful operation of the automated drip irrigation system. The assumption is that the water source meets the required standards for irrigation.
2. **Supportive Infrastructure:**
   - Assumption: Existing infrastructure, such as water pipelines, electrical connections, and land layout, can support the installation and operation of the automated drip irrigation system.
   - Implication: The assumption is that the university's infrastructure can accommodate the necessary components of the drip irrigation system without significant modifications.
3. **Budget and Funding:**
   - Assumption: Sufficient budgetary resources and funding mechanisms are available to cover the costs associated with system components, installation, automation technology, and ongoing maintenance of the automated drip irrigation system.
   - Implication: Adequate funding is crucial for the timely implementation and effectiveoperation of the system, ensuring its long-term viability and impact.
4. **The group members assume to finish the system according the given time.**

These assumptions serve as the foundation for the successful implementation of automated drip irrigation at Woldia University, guiding the project's planning, execution, and outcomes to achieve sustainable agriculture, operational efficiency, and educational advancement within the university community.

Constraint: - The major constraints of the system are internet and electric power.

1.9 RISKS AND CONTINGENCIES

Some risks and contingencies that may happen during the development of the system are includes absence of electrical power, absence of team members during working because of different problems are happene.

1.10 LIMITATIONS OF THE PROJECT

There are some limitations of our project:

- No backup power source
- Sensors value has fluctuations
- Water requirement of plant vary in growth state but our system is based only sensor values(data).

1.11 ORGANIZATIONS OF THE PROJECT

This project document consists of six chapters followed by references and appendix. The first chapter is an introduction part that contains background of the area which assures that a number of works are required in this area, contribution of the project, the scope of the project, the statement of problem, objective of project, significance of our project.

Chapter two contains a review of different literature like books, journals, websites; etc. related to automatic drip irrigation system running on solar power problem and electric power and operation.

A number of system component and their operation have been introduced in chapter three. This system component includes resister, LED diode, voltage regulator, relay, transistor, moisture sensor, Temperature sensor, level sensor, Arduino platform, liquid crystal display, and GSM module, water pump, potentio-meter, and sounder.

The fourth chapter consists of the design and analysis of different circuits. The fifth chapter explains simulation results of the project. It includes the interpretation and discussion of automatic control of the drip irrigation system.

Finally the last chapter concludes what has been done and recommends certain remaining works to be included by future workers in the related area.

CHAPTER 2

LITERATURE REVIEW

2.1 LITERATURE REVIEW

Agbetuyi, O. Evwieroghene, A. Awelewa, S. Wara Tita and O. Tobiloba [1] built an automatic irrigation system based on monitoring soil moisture.the main aim of the project is to see how human control could be removed from irrigation and also to optimize the use of water in the process. The method employed is to continuously monitor the soil moisture level to decide whether irrigation is needed, and how much water is needed in the soil. A pumping mechanism is used to deliver the needed amount of water to the soil. The advantages of automatic irrigation to the plants include saving money, water, conservation of labor and overall convenience.

R. Ghodake and A. Mulani [2] built Microcontroller Based Automatic Drip Irrigation System. The main aim of the project is to optimize water usage in agriculture by delivering the required amount of water directly to the plant roots. These projects have focused on sensor integration, water scheduling algorithms, and remote monitoring capabilities. Some projects have also investigated the use of wireless communication technologies to enable real-time data transmission and control.

M. Guerbaoui, a.ed-dahhak [3] GSM based automated drip irrigation system" we they proposed a system contribution to the development of greenhouse production in Morocco. The proposed solution involves the development of an integrated system for automate the drip fertilizing irrigation in green house. The solution adopted involves a data acquisition card controlled by PC. The irrigation is provided by a hydraulic circuit based on an electric pump. Water needs are evaluated by measuring soil water status by soil humidity sensor.

2.1.1 Strengths of the projects

- **Water conservation:** The project aims to optimize water usage by monitoring soil moisture levels and delivering the precise amount of water needed. This approach helps in conserving water resources and preventing wastage.

- **Cost savings:** By automating the irrigation process, the project can potentially save money for farmers by reducing the need for manual labor and optimizing water usage. This can lead to increased efficiency and lower operational costs.
- **Labor efficiency:** Automating the irrigation system removes the need for human intervention in monitoring and controlling the irrigation process. This frees up labor resources, allowing farmers to focus on other essential tasks and potentially increasing productivity.
- **Convenience:** The automatic irrigation system eliminates the need for constant manual monitoring and intervention, providing convenience to farmers. They can rely on the system to maintain optimal soil moisture levels without constant supervision.

2.1.2 Weaknesses of the projects

- **Initial cost:** Implementing an automatic irrigation system can involve significant upfront costs, including the purchase and installation of sensors, pumps, and other necessary equipment. This cost might be a barrier for small-scale farmers with limited financial resources.
- **Technical complexity:** Developing and maintaining an automatic irrigation system requires technical expertise. Farmers may need assistance in setting up and troubleshooting the system,which could be challenging without proper support or training.
- **Sensor accuracy and reliability:** The effectiveness of the system relies heavily on accurate and reliable soil moisture sensors. If the sensors are not calibrated correctly or if they malfunction, it can lead to incorrect irrigation decisions, resulting in over- or under-watering of plants.
- **Power dependency:** Automatic irrigation systems typically require a power source to operate the pumping mechanism and other components. In areas with unreliable or limited access to electricity, this dependency on power can be a limitation.
- **Environmental factors:** The project's success heavily depends on the assumption that soil moisture levels accurately reflect the water needs of plants. However, other environmental factors such as temperature, humidity, and plant type may also influence irrigation requirements. Failure to consider these factors might lead to suboptimal irrigation practices.

CHAPTER 3

METHDOLOGY AND SYSTEM COMPONENTS

3.1 METHODOLOGY

3.1.1 System development methodology

Implementing automated drip irrigation involves a series of methodologies to design, install, and operate the system effectively. Here are the key steps and methodologies that we used to follow in setting up an automated drip irrigation system:

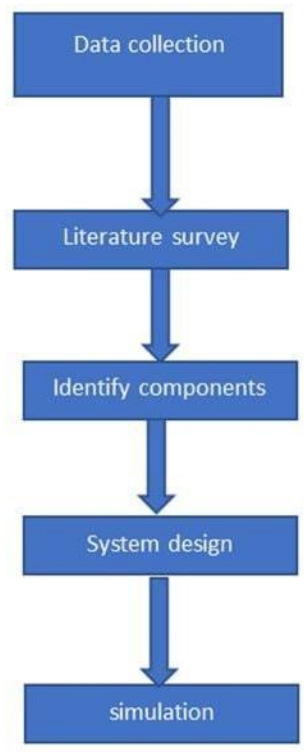

Figure 3. 1 Methodology

3.1.2 Data gathering and collection

- Data collection: - Collecting data on soil moisture levels and other requirements that helps determine when and how much water to apply to the crops.
- Collect weather data (temperature,) with irrigation data helps in adjusting watering schedules based on external conditions.

3.1.3 Data analysis

- Analyzing soil moisture data over time can reveal patterns and trends, aiding in adjusting irrigation schedules for optimal plant growth.
- Analyzing weather trends alongside irrigation performance can optimize watering strategies to account for climatic variations.

3.1.4 Literature review

- Literature will be reviewed to study the possible design options for optimum performance,and each device was selected.

3.1.5 Identify components

- We selected controllers, pumps, valves, timers, and other embedded components that support automation and can regulate water flow and distribution effectively.
- We choose the appropriate drip lines and emitters based on the soil type, crop water requirements, and system pressure.
- Determined which materials or components we utilized for the project.

3.1.6 System design

- Sensor Integration: Incorporate soil moisture sensors, temperature sensors, water level sensorwith a micro controller to enable automated adjustments based on real-time data.
- Integrate other components with appropriate place.
- Controller Programming: Program the irrigation controller with the desired wateringschedule, duration, and frequency based on crop requirements and environmental conditions.

3.1.7 Simulation

- Run or simulate the designed on proteus software or other software after interconnecting the components on the software.

3.2 COMPONENTS OF THE PROJECT

3.2.1 Hardware components

3.2.1.1 Sensors

A sensor is a device that detects and measures a physical quantity from the environment and converts it into an electronic signal. The physical quantity could be moisture, temperature, motion and light or any other physical phenomenon. Examples of sensors include: oxygen sensors, temperature sensors, infrared sensors, humidly sensors, level sensors, soil moisture sensors and motion detection sensors. The output of the sensors is usually current or voltage signal. This project requires soil moisture sensor, level sensor and humidity sensor.

1. **Soil Moisture Sensors**

A soil moisture sensor is a device that measures the volumetric water content (VWC) of soil. There are different types of moisture sensors. But for our projects we select YL-69 Moisture Sensor YL-69 Moisture Sensor; this is an Electrical resistance Sensor. The sensor is made up of two electrodes. Thissoil moisture sensor reads the moisture content around it. A current is passed across the electrodes through the soil and the resistance to the current in the soil determines the soil moisture. If the soil has more water resistance will be low and thus more current will pass through. On the other hand when the soil moisture is low the sensor module outputs a high level of resistance. This sensor has both digital and analogue outputs. Digital output is simple to use but is not as accurate as the analogue output.[6]

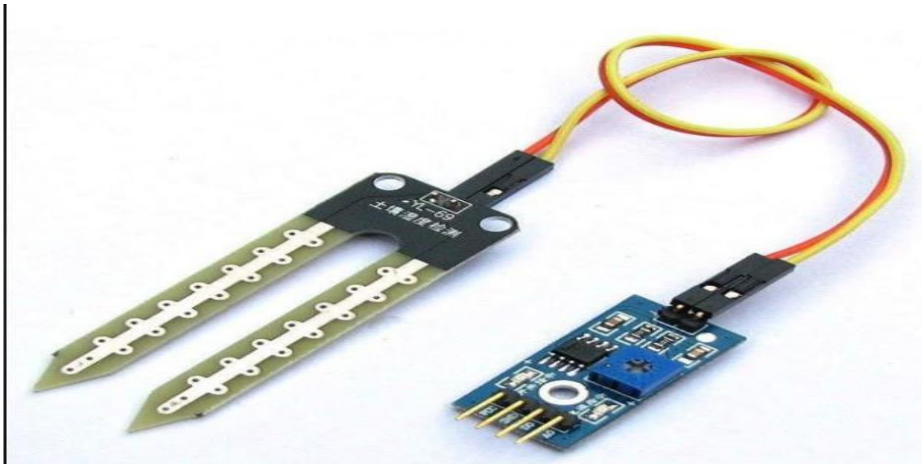

Figure 3. 2 YL-69 soil moisture sensor

Specification:

Sensor Type	Operating
Soil Moisture sensor	3.3 V - 5V
Current flow	< 20 A
Type of Interface	Analog Type
Working temperature of the sensor	10- 30 degree Celsius
PCB Dimension of comparator	3 cm x 1.5 cm
Soil Probe Dimension	6 cm x 3 cm
Cable length	20 cm
Weight	50 gms

2. **Soil PH sensor**

Soil pH is a measure of the acidity or alkalinity of the soil. A pH value is actually a measure of hydrogen ion concentration. Because hydrogen ion concentration varies over a wide range, a logarithmic scale (pH) is used Most soils have pH values between 3.5 and 10. In higher rainfall areas the natural pH of soils typically ranges from 5 to 7, while in drier areas the range is 6.5 to 9.

Soils can be classified according to their pH value:

- The PH is from 6.5 to 7.5—neutral
- Over 7.5—alkaline
- Less than 6.5—acidic, and soils with pH less than 5.5 are considered strongly acidic

Natural soil pH depends on the rock from which the soil was formed and the weathering processes that acted on it, i.e. climate, vegetation, topography & time. These processes tend to cause a lowering of pH (increase in acidity) over time.

Rain is also considered as one of the significant factors for the increase in acidity of the soil. Some fertilizers can change soil pH and increase or reduce the number of nutrients available to plants.

Soil pH affects the amount of nutrients and chemicals that are soluble in soil water. Some nutrients are more available under acid conditions while others are more available under alkaline conditions. However, most mineral nutrients are readily available to plants when soil pH is near neutral.

The development of strongly acidic soils can result in poor plant growth as a result of Aluminum & Manganese toxicity or calcium & magnesium deficiency. Alkaline soils may have problems with deficiencies of nutrients such as zinc, copper, boron & manganese.[6]

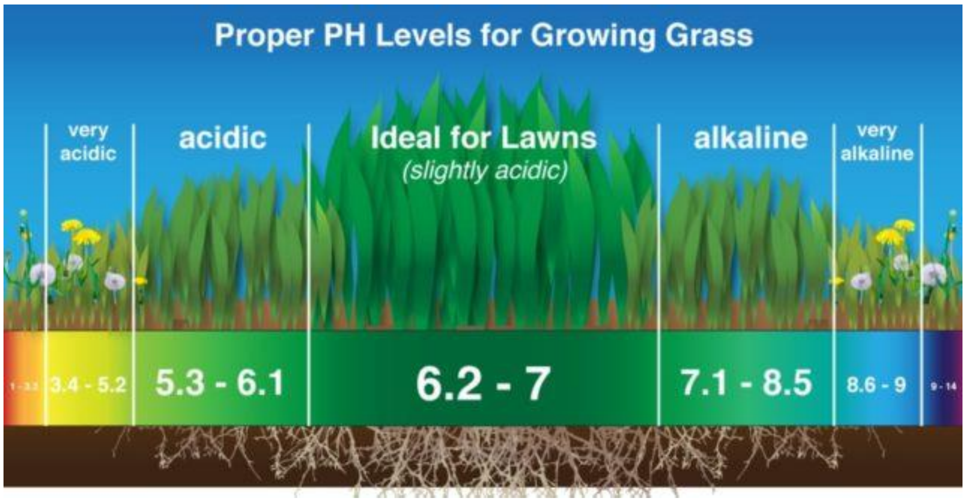

Figure 3. 3 PH level of soil

This is a waterproof and dustproof soil PH Sensor that can measure the Soil Ph value from 3 to 9 with high accuracy up to ±0.3PH. The sensor has an IP68 protective case & is sealed with High-density epoxy resin which can prevent moisture from entering the body interior part. The sensor is suitable for Suitable for agricultural cultivation, industrial production, environmental monitoring, animal husbandry, and sewage treatment.

Specifications

Probe type: Probe electrode

Measuring range: **3 ~ 9 PH**

Measurement accuracy: ±0.3PH

Resolution: 0.1 PH

Output signal: **RS485/0-5V / 0-10V / 4-20mA output**

Supply voltage: **5V~30VDC**

Working temperature range: **-30 ° C ~ 70 ° C**

Stabilization time: **5-10 Minutes** after power on

Response Speed: **≤15S**

Standard **2 meters** Cable

Long-term Stability: ≤**5%/y**

The Soil Ph Sensor has 4 pins as it need to be connected to RS485 or MAX485 Module. The four colored wires are Yellow, Blue, Black & Brown.

Figure 3. 4 Soil PH sensor

3. Rain Sensor

A rain sensor is one kind of switching device which is used to detect the rainfall. It works like a switch and the working principle of this sensor is, whenever there is rain, the switch will be normally closed.

A sensor that is used to notice the water drops or rainfall is known as a rain sensor. This kind of sensor works like a switch. This sensor includes two parts like sensing pad and a sensor module. Whenever rain falls on the surface of a sensing pad then the sensor module reads the data from the sensor pad to process and convert it into an analog or digital output. So the output generated by this sensor is analog (AO) and digital (DO).

Working Principle

The rain sensor working principle is pretty simple. The sensing pad includes a set of uncovered copper traces which mutually work like a variable resistor or a potentiometer. Here, the sensing pad resistance will be changed based on the amount of water falling on its surface. So, here the resistance is inversely related to the amount of water. When the water on the sensing pad is more, the conductivity is better & gives less resistance. Similarly, when the water on the surface pad is less, the conductivity is poor & gives high resistance. So the output of this sensor mainly depends on the resistance.

The applications of rain sensor include the following:-

This sensor is used as a water preservation device and this is connected to the irrigation system to shut down the system in the event of rainfall.

This sensor is used to guard the internal parts of an automobile against the rainfall as well as to support the regular windscreen wiper's mode.

This sensor is used in specialized satellite communications aerials for activating a rain blower over the opening of the aerial feed, to get rid of water droplets from the mylar wrap to keep pressurized as well as dry air within the waveguides [8].

Specifications

Operating voltage ranges from 3.3 to 5V

The operating current is 15 mA

The sensing pad size is 5cm x 4 cm with a nickel plate on one face.

Comparator chip is LM393

Output types are AO (Analog o/p voltage) & DO (Digital switching voltage)

The length & width of PCB module 3.2cm x 1.4cm

Sensitivity is modifiable through Trimpot

Red/Green LED lights indicators for Power & Output

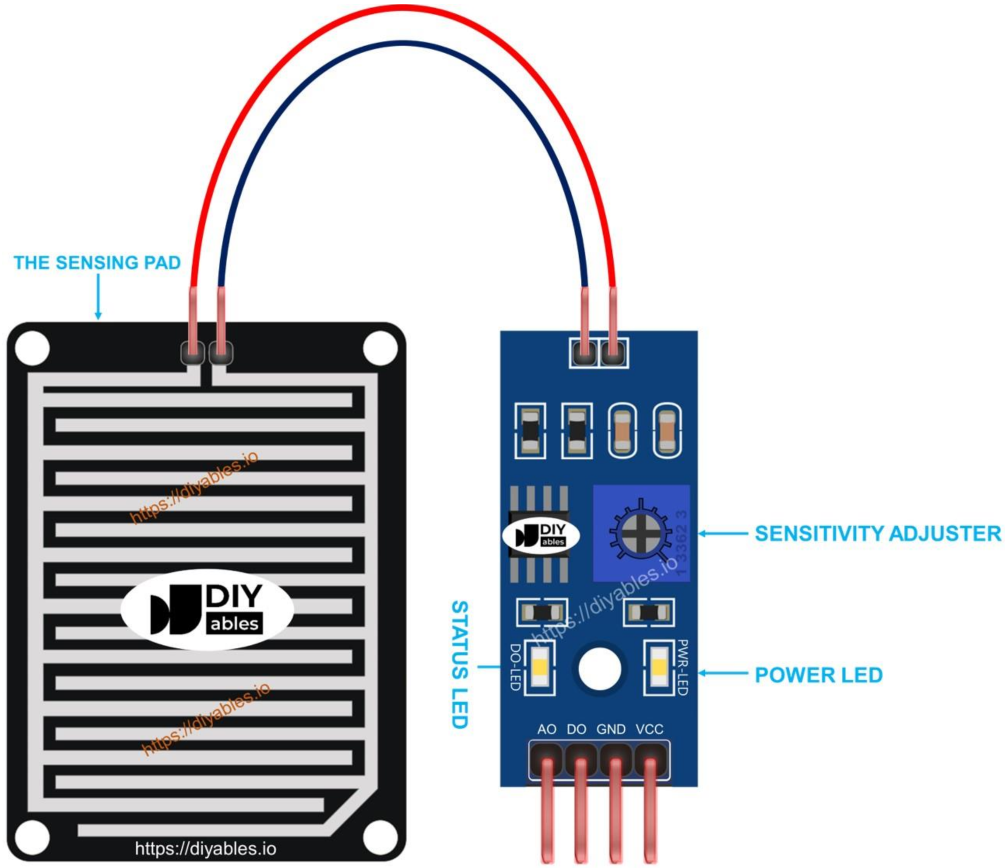

Figure 3. 5 Rain sensor

4. Humidity Sensors (DHT11)

Humidity is defined as the amount of water present in the surrounding air. This water content in the air is a key factor in the wellness of mankind. For example, we will feel comfortable even if the temperature is 00C with less humidity i.e. the air is dry. But if the temperature is 100C and the humidity is high i.e. the water content of air is high, then we will feel quite uncomfortable. Humidity is also a major factor for operating sensitive equipment like electronics, industrial equipment, electrostatic sensitive devices and high voltage devices etc. Such sensitive equipment must be operated in a humidity environment that is suitable for the device.

Temperature and humidity sensors (also known as RH temp sensors) are a device that transforms temperature and humidity into electrical impulses that are easily measurable. Temperature and relative humidity transceivers on the market compute the amount of relative humidity and temperature in the atmosphere, convert it into electrical impulses or other signal types according to some rules, and output the gadget to a device or application to meet users' environmental monitoring necessities. A humidity sensitive capacitor and a conversion circuit are the major components of the humidity and temperature sensor module. A glass substrate, a humidity sensitive material, a lower electrode, and an upper electrode make up the humidity-sensitive capacitor.

Figure 3. 6 Humidity sensor

Specifications:

Supply Voltage: 3.5 to 5.5V

Output Signal: digital signal via single-bus

Operating Range and Accuracy (Humidity): 20-80% RH; +/-5% RH

Operating Range and Accuracy (Temperature): 0 to 50 C; +/-2% C

Average Sending Period: 2 seconds

Dimensions (excluding pins): 12.6mm (0.5") length x 5.83mm (0.23") width x 16mm (0.63") height

5. **Ultrasonic Sensor**

Ultrasonic sensor is an instrument that measures the distance to an object using ultrasonic sound waves. An ultrasonic sensor uses a transducer to send and receive ultrasonic pulses that relay back information about an object’s proximity. High-frequency sound waves reflect across boundaries to produce distinct echo patterns.

Ultrasonic sensors work by sending out a sound wave at a frequency above the range of human hearing. The transducer of the sensor acts as a microphone to receive and send the ultrasonic sound. Our ultrasonic sensors, like many others, use a single transducer to send a pulse and to receive the echo. The sensor determines the distance to a target by measuring time lapses between the sending and receiving of the ultrasonic pulse.

The working principle of this module is simple. It sends an ultrasonic pulse out at 40 kHz, which travels through the air, and if there is an obstacle or object, it will bounce back to the sensor. By calculating the travel time and the speed of sound, the distance can be calculated. Ultrasonic sensors are a great solution for the detection of clear objects. For liquid level measurement, applications that use infrared sensors, for instance, struggle with this particular use case because of target translucence[6].

For presence detection, ultrasonic sensors detect objects regardless of color, surface, or material (unless the material is very soft, like wool, as it would absorb sound). To detect transparent and other items where optical technologies may fail, ultrasonic sensors are a reliable choice.

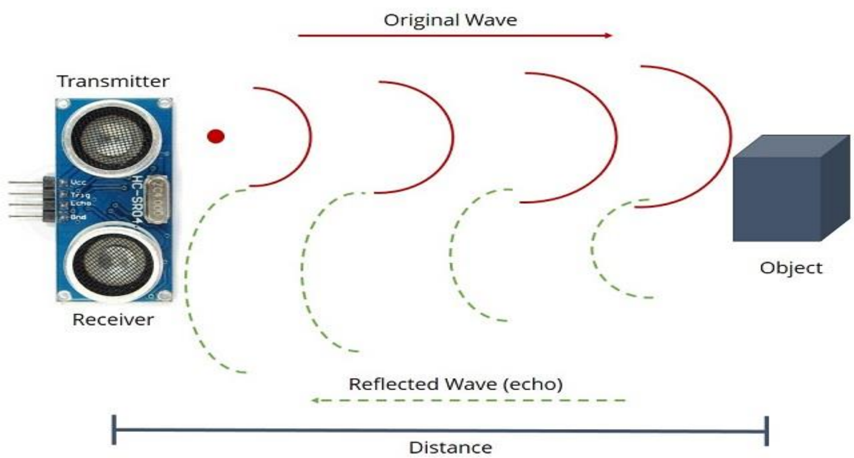

Figure 3. 7 Ultrasonic sensor

Specifications

Power Supply: 3.3V – 5V

Operating Current: 8mA

Working Frequency: 40Hz

Ranging Distance: 3cm – 350cm/3.5m

Resolution: 1 cm

Measuring Angle: 15 degree

Trigger Input Pulse width: 10uS TTL

Dimension: 50mm x 25mm x 16mm

3.2.1.2 Diode

A diode plays an important role in the system circuit project. It allows the battery voltage to flow intothe circuit only in one direction (called the diode's forward bias direction) and also block any back electromotive force that may damage the driver transistors. The diode is a two-terminal electronic component with a nonlinear current–voltage characteristic. This unidirectional behavior of diode is called rectification and it is used in this project to protect the back emf of motor.

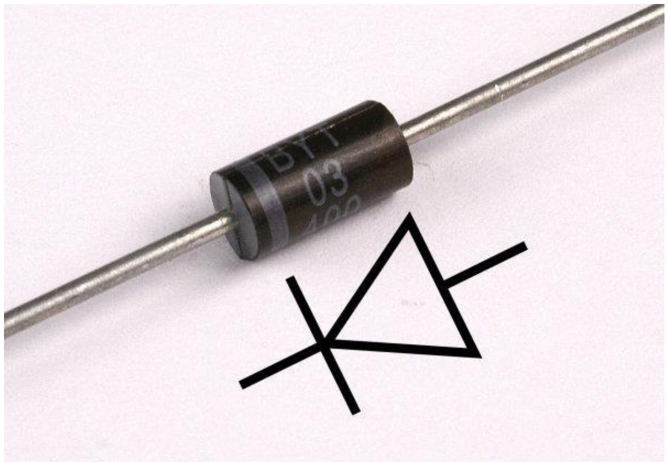

Figure 3. 8 Diode

3.2.1.3 Voltage regulator

A voltage regulator is designed to automatically maintain a constant voltage level. If the output voltage is too low (perhaps due to input voltage reducing or load current increasing), the regulation element is commanded, up to a point, to produce a higher output voltage by dropping less of the input voltage (for linear series regulators and buck switching regulators), or to draw input current for longerPeriods (boost-type switching regulators); if the output voltage is too high, the regulation elementwill normally be commanded to produce a lower voltage.

3.2.1.4 Arduino uno

The Arduino UNO is a standard board of Arduino Arduino UNO is a microcontroller board based on the ATmega328P. 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. 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.

The IDE is common to all available boards of Arduino[6].

Figure 3. 9 Arduino UNO

3.2.1.5 Relay

This is an electromagnetic switch which is activated when a current is applied to it. A relay uses small currents to switch huge currents. Most relays use principle of electromagnetism to operate but still other operating principles like solid state are also used. A contactor is a type of relay which can handle a high power required to control an electric motor or other loads directly. Solid state relays have no moving parts and they use semiconductor devices to perform switching.

Figure 3. 10 Relay

Coil Relay: A contact relay switches one or more poles each of whose contacts can be thrown by energizing the coil in three ways namely; normally open(NO), normally closed(NC) or change over(CO). Just like manual switches the relay switch part is available in various configurations.

Double pole, double throw (DPDT) configuration is most common configuration. DPDT means that the relay separately controls two switches that work together. Both switches have a normally NO and NC contacts. Other commonly used configurations are:

Single Pole Single Throw (SPST): This relay configuring has four terminals. Two of the terminals are coil terminals.

SPDT – Single Pole Double Throw. This configuring has five terminals. One of the terminals is a common terminal which connects to either of the two others.

3.2.2 Software components

3.2.2.1 Arduino IDE

The Arduino IDE is an open-source software, which is used to write and upload code to the Arduino boards. The IDE application is suitable for different operating systems such as Windows, Mac OS X, and Linux. It supports the programming languages C and C++. Here, IDE stands for Integrated Development Environment.

The program or code written in the Arduino IDE is often called as sketching. We need to connect the Genuino and Arduino board with the IDE to upload the sketch written in the Arduino IDE software. The sketch is saved with the extension '.ino.'

The Arduino IDE will appear as:

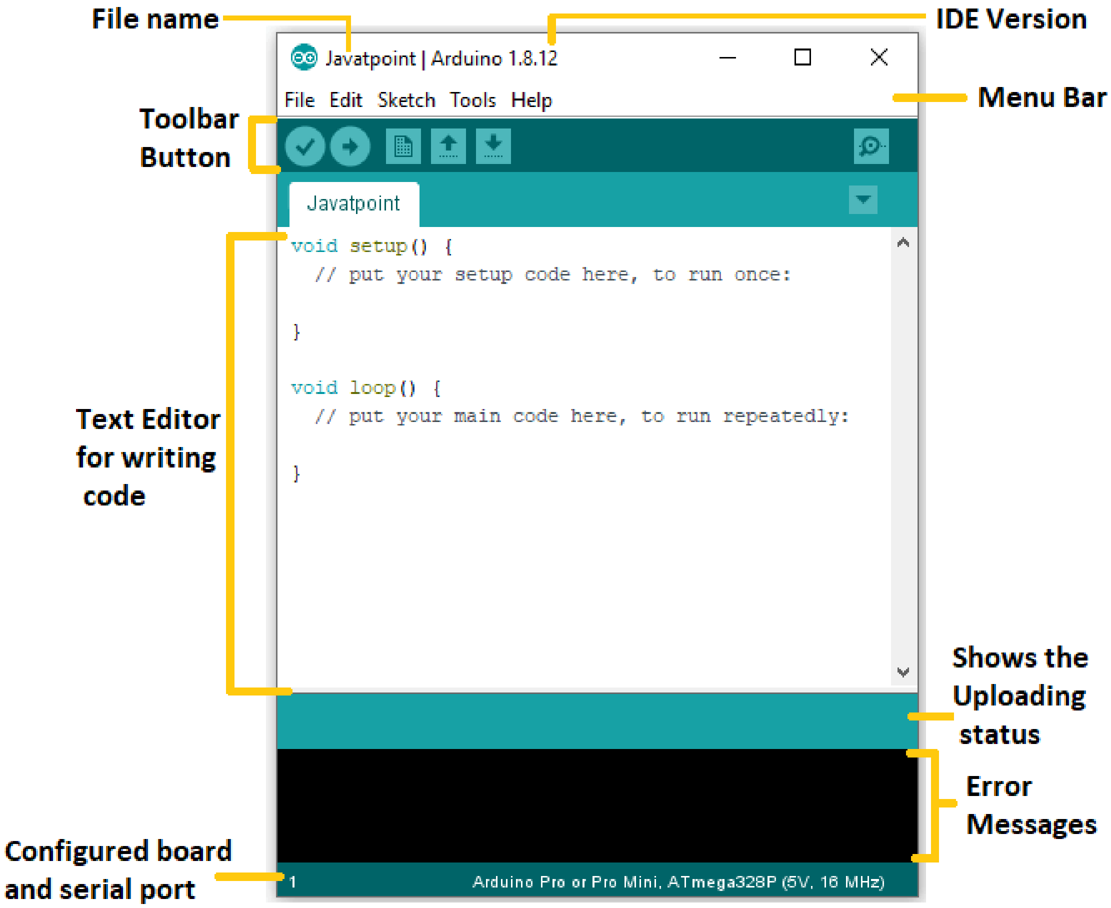

Figure 3. 11 Arduino ide program text space

Toolbar Button

The icons displayed on the toolbar are New, Open, Save, Upload, and Verify.

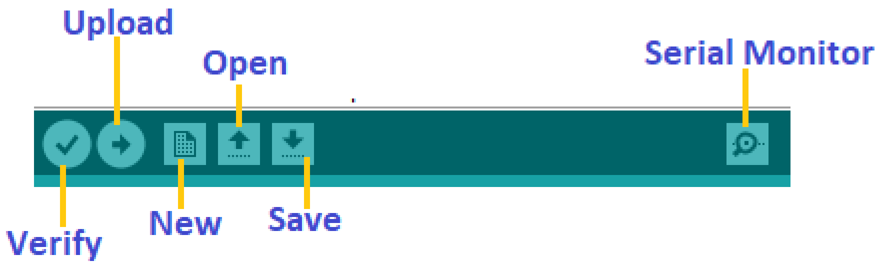

Figure 3. 12 Toolbar

- Upload

The Upload button compiles and runs our code written on the screen. It further uploads the code to the connected board. Before uploading the sketch, we need to make sure that the correct board and ports are selected.

We also need a USB connection to connect the board and the computer. Once all the above measures are done, click on the Upload button present on the toolbar.

The latest Arduino boards can be reset automatically before beginning with Upload. In the older boards, we need to press the Reset button present on it. As soon as the uploading is done successfully, we can notice the blink of the Tx and Rx LED.

If the uploading is failed, it will display the message in the error window. We do not require any additional hardware to upload our sketch using the Arduino Bootloader. A Boot loader is defined as a small program, which is loaded in the microcontroller present on the board. The LED will blink on PIN 13.

- Open

The Open button is used to open the already created file. The selected file will be opened in the current window.

- Save

The save button is used to save the current sketch or code.

- New

It is used to create a new sketch or opens a new window.

- Verify

The Verify button is used to check the compilation error of the sketch or the written code.

- Serial Monitor

The serial monitor button is present on the right corner of the toolbar. It opens the serial monitor.

- Export compiled Binary

It allows saving a .hex file and can be kept archived. Using other tools, .hex file can also be sent to the board.

3.2.2.2 Proteus

Proteus software is a computer-aided design (CAD) software suite for electronic design automation (EDA). It is used by electrical engineers and designers to create schematics, printed circuit board (PCB) layouts, and other documentation for electronic devices. Proteus software is known for its ease of use, powerful features, and wide range of supported devices.

Here are some of the key features of Proteus software:

- Schematic capture: Proteus software allows users to create schematics using a wide range of symbols and components. The software includes a library of thousands of pre-made symbols, making it easy to create designs quickly and easily.
- PCB layout: Proteus software allows users to create PCB layouts using a variety of tools and features. The software includes a powerful autorouter that can automatically route traces on a PCB.
- Simulation: Proteus software allows users to simulate electronic circuits to verify their functionality. The software includes a variety of simulation engines that can be used to simulate analog, digital, and mixed-signal circuits.
- Documentation: Proteus software allows users to create documentation for their electronic designs. The software includes a variety of tools for creating schematics, PCB layouts, and other documentation.

Proteus software is a powerful and versatile CAD software suite that is used by electrical engineers and designers around the world. The software is known for its ease of use, powerful features, and wide range of supported devices.

3.2.2.3 Blynk App

The Blynk app for irrigation systems is a mobile app that allows users to control and monitor their irrigation systems remotely. The app connects to the irrigation system via a Wi-Fi or cellular connection, and allows users to:

- Turn the irrigation system on and off
- Set watering schedules
- Adjust watering durations
- Monitor water usage
- Receive alerts when there is a problem with the irrigation system

The Blynk app is easy to use and can be customized to meet the needs of any irrigation system. The app is available for free on iOS and Android devices[9].

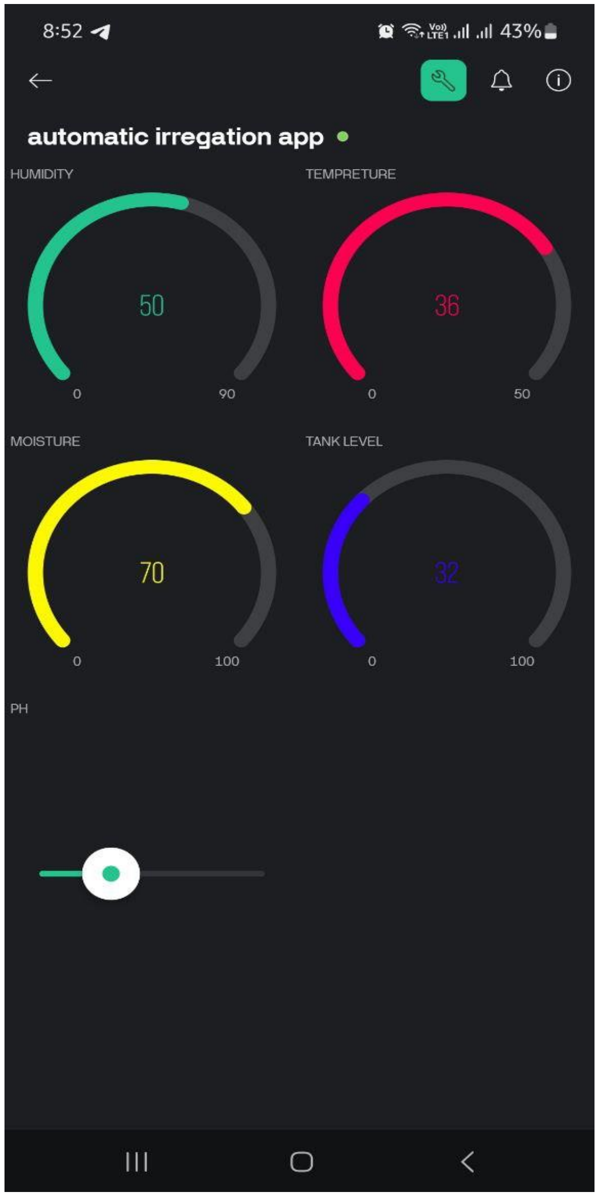

Figure 3. 13 Blynk app

Here are some of the benefits of using the Blynk app for irrigation systems:

- Remote control: The Blynk app allows users to control their irrigation systems from anywhere in the world. This is convenient for users who are away from home or who want to be able to monitor their irrigation systems remotely.
- Scheduling: The Blynk app allows users to set watering schedules for their irrigation systems. This ensures that plants are watered regularly and consistently.
- Adjustments: The Blynk app allows users to adjust watering durations on the fly. This is useful for adjusting watering schedules based on weather conditions or plant needs.

- Monitoring: The Blynk app allows users to monitor water usage and receive alerts when there is a problem with the irrigation system. This helps users to identify and fix problems quickly and easily.

Overall, the Blynk app for irrigation systems is a valuable tool for anyone who wants to be able to control and monitor their irrigation system remotely. The app is easy to use, customizable, and can help users to save water and keep their plants healthy. .

CHAPTER 4

SYSTEM DESIGN AND ANALYSIS

4.1 OVERALL SYSTEM BLOCK DIAGRAM

we start designing each parts of the system, a block diagram representing this automatic drip irrigation system based on Arduino has been made based up on the way we analyzed and figure out to do it like. The block diagram of overall system is shown in figure [7] below.

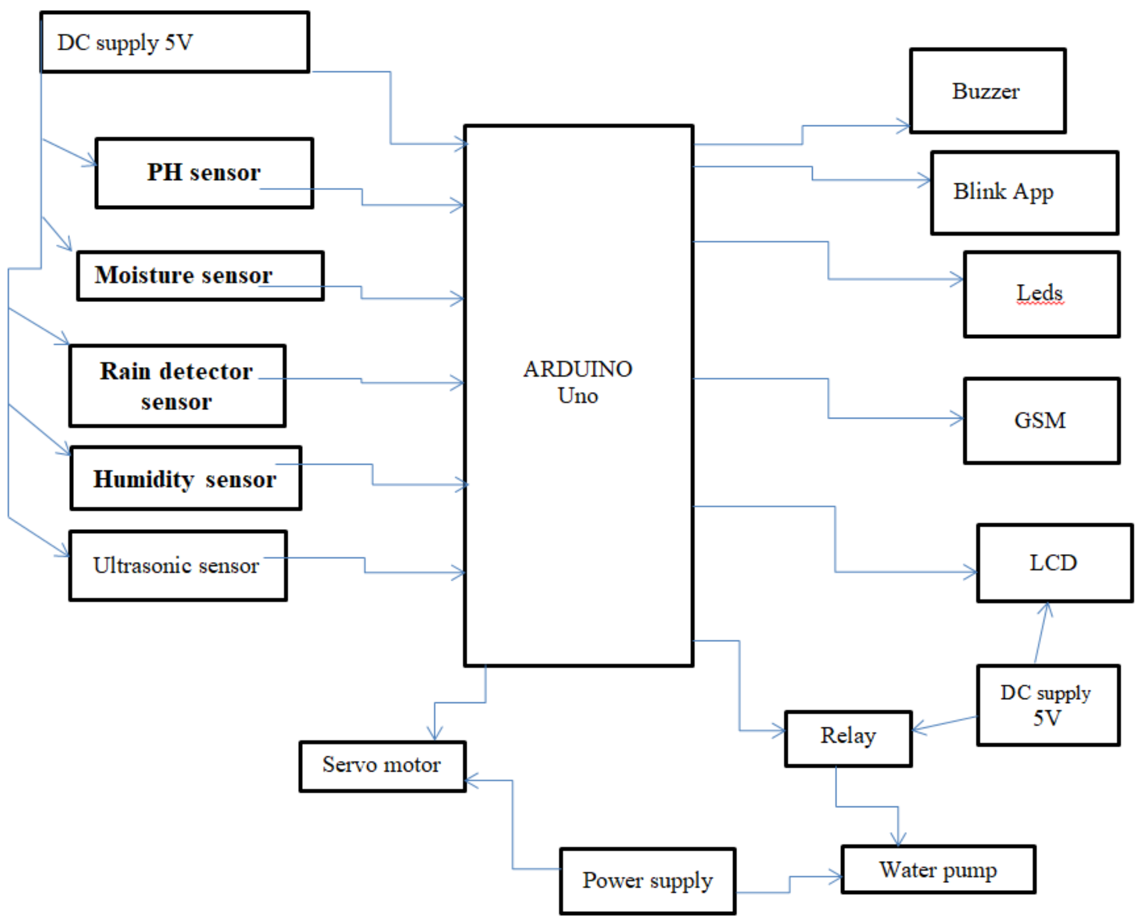

Figure 4. 1 Block diagram representation of overall system

4.1.1 Block diagram description of the system

The basic block diagram of the system includes the following blocks: Arduino uno, water level sensor soil moisture sensor, LCD, relay, watering pump motor, PH sensor, rain detector sensor, humidity sensor, servo motor, tank pump motor, and DC power supplies of 5V and 12V. The soil moisture sensor senses the moisture content of the soil and sends the data to the Arduino Uno microcontroller. The pH sensor senses the pH level of the soil.

The water level sensor detects the amount of water in the tank and is connected to the microcontroller. The humidity sensor collects data on the water content in the air surrounding the system.

The rain detector sensor detects rainfall in the area. The relay is connected to the output of the microcontroller and acts as a switch for the DC motor, based on the microcontroller's output.

The LCD display is connected to the output of the microcontroller to display the sensor readings and system status. The 12V DC power supply is used to power the DC motor, while the 5V DC power supply is used to power the Arduino Uno microcontroller.. The relay is also connected to the output of the microcontroller to act as a switch for the DC motor based on the microcontroller's output. The LCD display is connected to the output of the microcontroller to display the output. The DC power supply is used as a power supply for the DC motor (12V) and the Arduino un microcontroller (5V).

4.2 TYPES OF IRRIGATION

4.2.1 Ditch Irrigation

This is one of the earliest irrigation methods to be used. Ditches/trenches are dug out and crops are planted along the ditches in rows.

4.2.2 Terrace Irrigation

The land is shaped into steps. Crops are planted on the flat areas. Water flows down the steps watering each of the flat areas. This method is both time and labor intensive in building the terraces.

4.2.3 Sprinkler irrigation

This method uses overhead sprinklers. Each sprinkler irrigates a given area. During installation care should be taken to avoid over or under watering some areas. If poorly installed a lot of water is wasted via runoff.

4.2.4 Drip Irrigation

In this irrigation system:

- Water is applied directly to the crop i e. entire field is not wetted.
- Water is conserved

- Weeds are controlled because only the places getting water can grow weeds.
- There is a low pressure system.
- There is a slow rate of water application somewhat matching the consumptive use.
- There is reduced evaporation, only potential transpiration is considered.
- There is no need for a drainage system

Information required for the proposed areas are:

- **Soil Properties:** Texture and structure, moisture equilibrium points, water holding capacity, agricultural potential, land classification, kinds of crops that the soil can support. Water Sources: water source availability, hydrologic data of the area, water quantity, water quality and possible engineering works are necessary to obtain water.
- **Weather data:** Temperature, relative humidity, sunshine hours and rainfall. Information about crops grown in the area: Check preference by people, market potential, and adaptability to area, water demand, growth schedules and planting periods. Topography (e.g. slopes): This helps to determine the layout of the irrigation system and method of irrigation water application suited for the area.
- **History of People and Irrigation in the area:** Check past exposure of people to irrigation and land tenure and level of possible re-settlement or otherwise.

4.3 Design of Drip Irrigation System

The general principles of design, the design procedure and system layout of drip system is that the spacing of emitters is much less than that of sprinklers and that water must be filtered and treated to prevent blockage of emitters.

- Not all areas are irrigated. In design, therefore, the area irrigated is only the accounted area.
- The irrigation interval is advisable to be daily as the application rate is small.
- Emitter spacing is not a function of wind

  Emitter discharge may be described by:

$q = kh^x$

where:

q is the emitter discharge.

K is constant for each emitter.

h is pressure head at which the emitter operates and x is the exponent characterized by the flow regime.

4.3.1 Sizing of drip laterals:

- The diameter of the lateral should be selected so that the difference in discharge between emitters operating simultaneously will not exceed 10 %.
- This allowable variation is same as for sprinkler irrigation laterals already discussed.
- To stay within this 10 % variation in flow, the head difference between emitters should not exceed 10 to 15 % of the average operating head for long-path or 20 % for turbulent flow emitters.
- The maximum difference in pressure is the head loss between the control point at the inlet andthe pressure at the emitter farthest from the inlet.
- The inlet is usually at the manifold where the pressure is regulated.
- The manifold is a line to which the drip laterals are connected.
- For minimum cost, on a level area 55 % of the allowable head loss should be allocated to the lateral and 45 % to the manifold.
- The Friction Loss for Mains and Sub-mains can be computed by Darcy Weisbach equation.
- F should be used to compute head loss for laterals and manifolds with multiple outlets, by multiplying a suitable F factor.
- For minimum cost, on a level area 55 % of the allowable head loss should be allocated to the lateral and 45 % to the manifold.
- The Friction Loss for Mains and Sub-mains can be computed from Darcy Weisbach equation for smooth pipes in trickle systems when combined with the Blasius equation for friction factor.

The equation is:

$$f = K\,L\,Q^{1.75}\,D^{-4.75} \quad \text{.......................................} (4.1)$$

Where: Hf is the friction loss in m;

K is constant = 7.89×10^5 for S.I. units for water at 20 ° C;

L is the pipe length in m;

Q is the total pipe flow in l/s; and D is the internal diameter of pipe in mm.

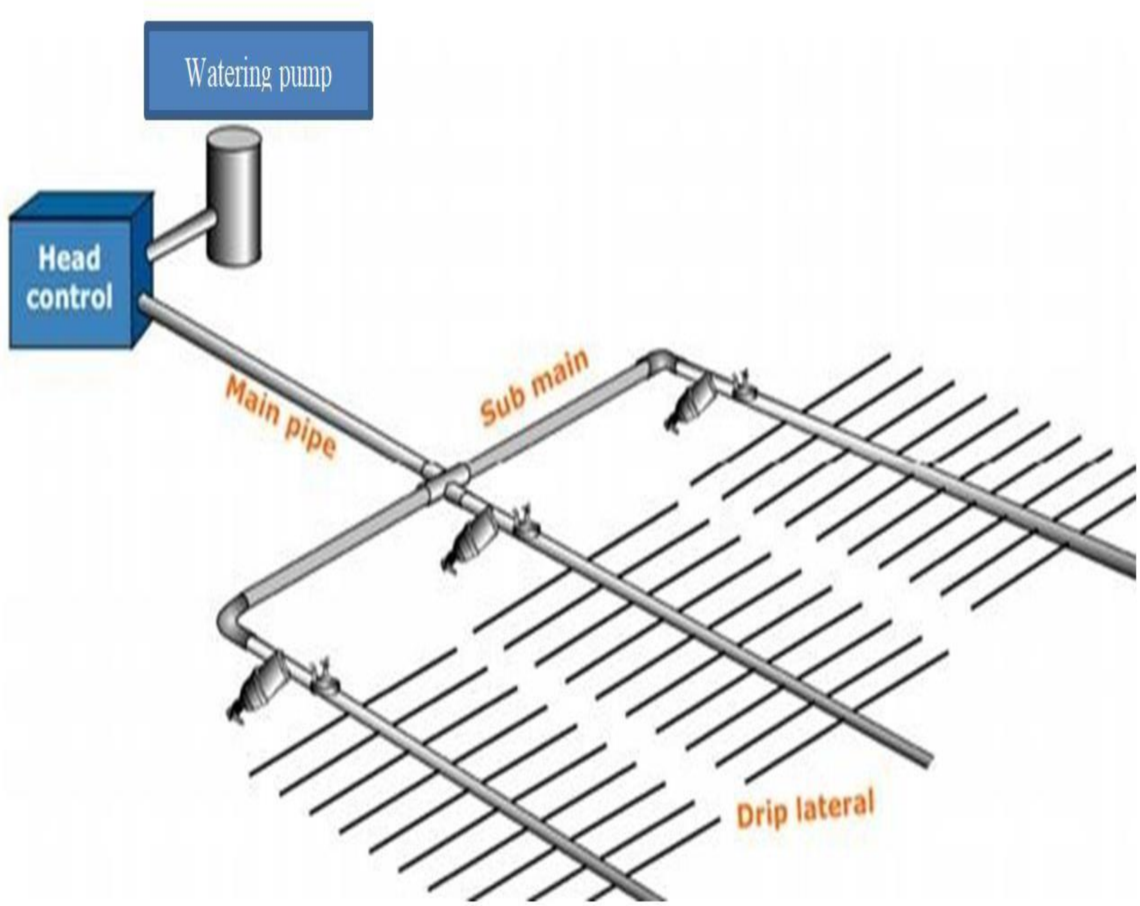

Figure 4. 2 Schematic Drip System

Design a drip Irrigation System for a fully matured orchard with the layout below. We

Assume that: The field topography level

- Maximum time for irrigation: 12 hours per day
- Allowable pressure variation in the emitters: 15%,
- Water source: Well
- Maximum suction lift at the well = 10m
- Peak ET rate: 7mm/day
- the matured orchard shades 70% of the area;
- Drip irrigation efficiency is 80%.
- Sections 1 and 2 are to be irrigated at the same time and alternated with sections3

and 4.

- Each tree is to be supplied by 4 emitters.
- Efficiency of pump=70%
- Efficiency of motor=70%

Assuming that the half lateral contains 12 tree

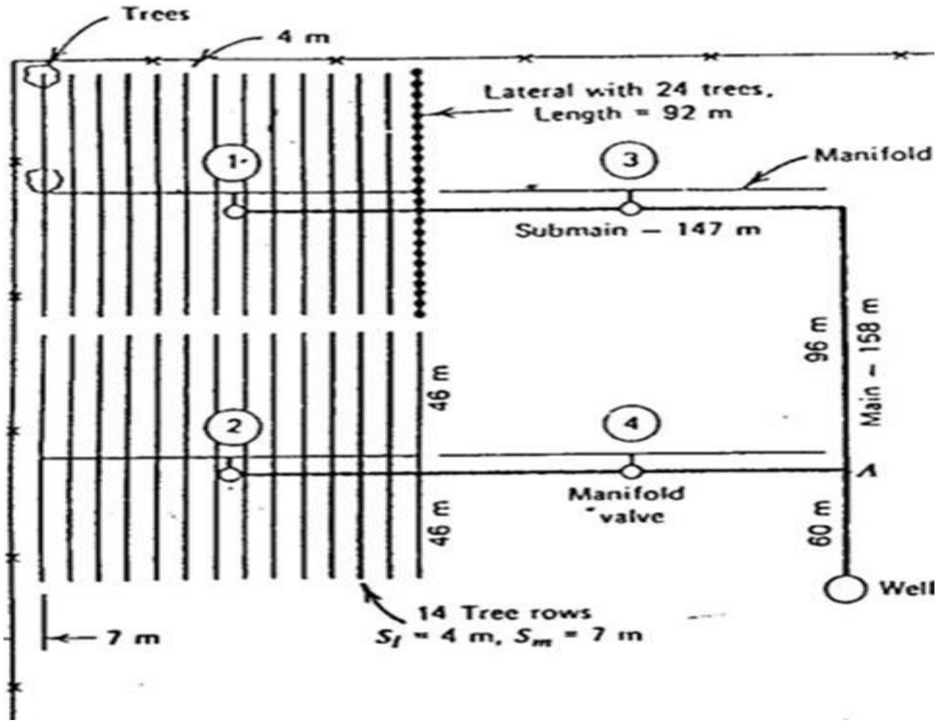

Figure 4. 3 Drip lateral

- 1st we calculate ETt

  ETt = ET x P/85 (4.2)

where: ETt is the average evaporation rate for crops

ET is the conventional evaporation rate for crops = 7 mm/day

P is the percentage of total areas shaded by the crop = 70%

Using equation (4.2) ETt = 7 mm/day x 70/85 = 5.8 mm/day.

- **2nd we determine the discharge for each tree**

  Assuming that tree spacing is 4m x 7m

  Discharge for each tree with a spacing of 4 m x 7 mqt = 4 m x 7 m x 5.8 x 10^{-3} m/day = 0.162 m^3/day qt = 0.00675 m^3/hr (24 hr. day)

For 12 hour working day the discharge is

Discharge required (qt) = 0.00675 x 24/12 = 0.0135 m3/hr = 0.00375 L/sWith an application efficiency of 80%, the required discharge per tree

is: 0.00375/0.8 = 0.0047 L/s

- 3rdThe discharge per emitter, with 4 emitters per tree is then

  = 0.0047/4 = 0.00118 L/s = 0.0012 L/s

- **4th Discharge of Each Line**

Table 4. 1 Discharge of Each Line

Line	No. of trees	No. of emitters	Required Discharge(L/S)
Half Lateral	12	48	0.0576
Half main fold	168	672	0.8060
Sub main, A to section 1	336	1344	1.6130
Main, A to pump	672	2688	3.2260

From manufacturer's catalogue, select emitter.

Say a medium long-path emitter with average discharge of 0.0012 L/s, k= 0.000073 and x= 0.63 is selected.

Substituting inequation $q=k*h^x$, the average operating pressure of the emitter can be found by first transforming the characteristic equation into logarithmic form and solving for h:

log q=logk +xlogh (4.3)

Logh= (logq-logk)/x=8.9m

Ha is then found as 8.9 m. This is the average operating head.

- **5th Total allowable pressure loss**:-of 15 % of Ha in both the Lateral and Manifold =8.9 x 0.15 =1.3 of which, 0.55 x 1.3 = 0.7 m is allowed for Lateral and 0.45 x 1.3 = 0.6 is allowed for the main fold.
- **6th Compute the friction loss** in each of the lines by Darcy-Weisbach equation by selecting a diameter to keep the loss within the allowable limits of 0.7 m and 0.6 m,already determined.

- **7th Compute the Friction Loss** in each of the Lines using equation (4).

4.3.2 Selection of Pipe Diameters

Table 4. 2 Selection of Pipe Diameters

Line	Q	Pipe Diameter OD(mm)	Pipe Diameter ID(mm)	L(m)	H_f“(m)
Half Lateral	0.0576	16	12.70	46	0.51
Half Main fold	0.8060	32	31.75	45.5	0.68
Sub-main, A to section 1	1.6130	50	44.45	243	6.59
Main, A to pump	3.2260	55	50.80	60	2.90

- 8^{th} Pressure Head at Manifold Inlet and at inlet of lateral

The pressure head at inlet of lateral:

HL = Ha + 0.75hfL ± 0.5HzL (4.4)

Where,

Ha = the operating pressure head of emitterhfL = friction head loss in lateral

HzL = elevation head difference along the lateralThe pressure head at inlet of manifold:

Hm = HL + 0.75hfm ± 0.5Hzm (4.5)

Where,

hfm = friction head loss in manifold

Hzm = elevation head difference along manifold

Therefore, using the eq (4.4) HL= 8.9 + 0.75x 0.51 = 9.28 m.And, from eq (4.5) Hm = 9.28+0.75x0.68 = 9.79 m.

- 9^{th} finally determine the Size of pump:

Total Head for Pump = Manifold Pressure = 9.79 m + Pressure loss at sub-main = 6.59 m +Pressure loss at Main = 2.90 m + suction lift = 10 m + Net Positive Suction head forpump = 3 m (assumed) + pressure head loss at control head = 3m = 35.28 m

i.e. The pump must deliver 2.03 L/s at a head of about 36 m.

Hp = (Qs x H)/(75xEpxEm) = (2.03x37.28)/(75x0.7x0.7) = 1.95 hp or we canapproximate to the highest power 2hp.

4.4 SYSTEM DESIGN

4.4.1 Water pump connection to the Arduino

To implement the final bit of the automated irrigation system an electric motor (12V DC) was selected as the water pump. The first two units of the system i.e. sensing unit and the control unit (microcontroller) are powered by 5V DC. To interface the control units with the motor a 5V DC relay was used as the isolation unit. The microcontroller was connected to the relay via an NPN transistor to protect the transistor; while turning it on, a resistor was used. The resistor limits the current flowing through the transistor. Ohms law was utilized to determine the size (in ohms) of the protection resistor to be interfaced with the transistor. From Ohms law

Rmin = (5 – 0.7) V / 40mA = 107.5Ω

To achieve current limitation a resistance value higher than the calculated Rmin was selected. A resistor of 470Ω was selected and thus the current through the transistor was limited to:

4.3V / 470Ω =9.12 mA

To protect the microcontroller from back EMF during switching a diode was connected across the relay. The connection diagram is as shown below.

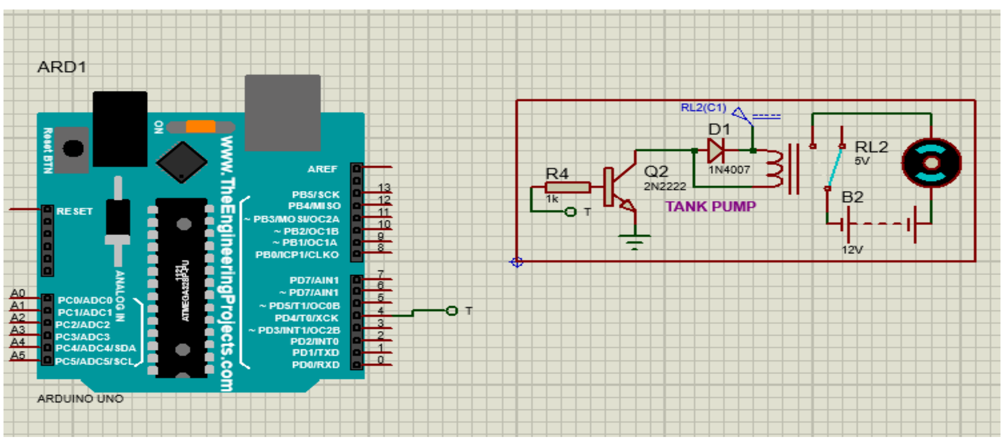

Figure 4. 4 Water pump connection to the Arduino

4.4.2 Servo motor connection to the Arduino

The Arduino can directly control the servo motor without the need for a transistor and relay.

The connection of a servo motor does not require the same level of current limiting or isolation since it operates at a lower power level.

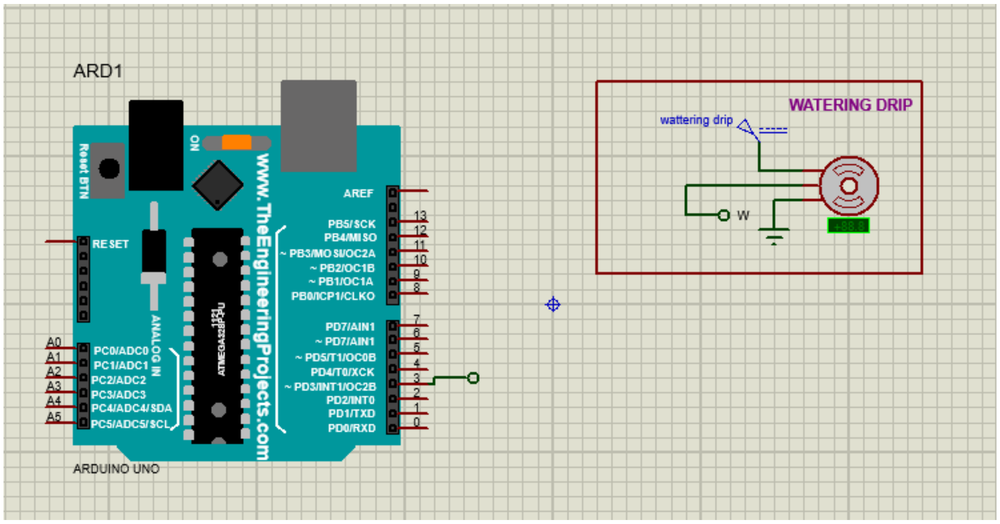

Figure 4. 5 Servo motor connection to the Arduino

4.4.3 LED and buzzer design

To indicate states of the soil LED and buzzer were used. The LED lit up depending on the soil moisture content. When the soil was dry LED connected to arduino digital pin 31 was lit. and the buzzer connected to arduino pin 32 was also ON. The LED and buzzer were connected to the microcontroller as shown below.

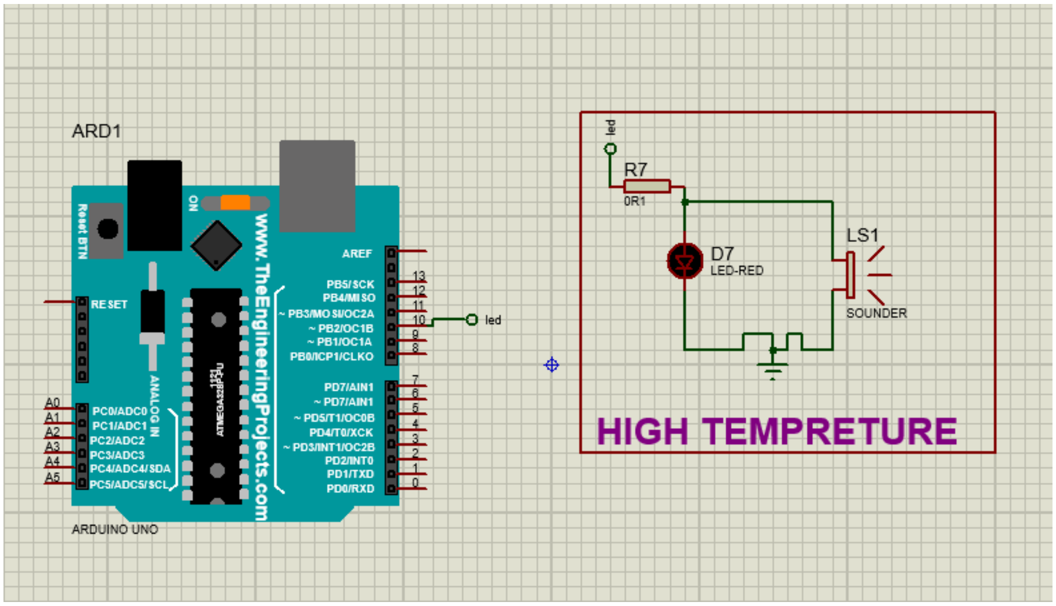

Figure 4. 6 LED and buzzer design

Ohms law was utilized to determine the size (in ohms) of the protection resistor to be interfaced with the LEDs.

From Ohms law Voltage (V) = {Current (I)}*{Resistance(R)}..............................(4.6)

Where as in our case;

V= Vcc (5V) -Voltage drop across LED (selected LEDs has 2.0V)

ILED= LED current (20mA)

VLED = LED Voltage drop (2V)

R= minimum required resistance value

Therefore, Rmin = {(5-2) V}/ {20mA} =150 Ω

To achieve current limitation a resistance value higher than the calculated Rmin was selected. A 220Ω resistor was used and thus only 13.6mA current was allowed to pass through each of the three LEDs.

4.4.4 LCD Interface with Arduino

To affect display a 20x4 Liquid Crystal Display (LCD) was chosen. LCD pins D4, D5, D6 and D7 were used as data lines in a 4 bit mode configuration. These pins were connected to arduino pins 8, 7, 6 and 5 respectively. LCD"s pin E (Enable) was connected to digital pin 11 on the arduino board. Pin RS (Register Select) on the LCD was connected to arduino digital pin 12.

R/W pin of the LCD was connected to GND (ground). The figure below shows the LCD microcontroller interface.

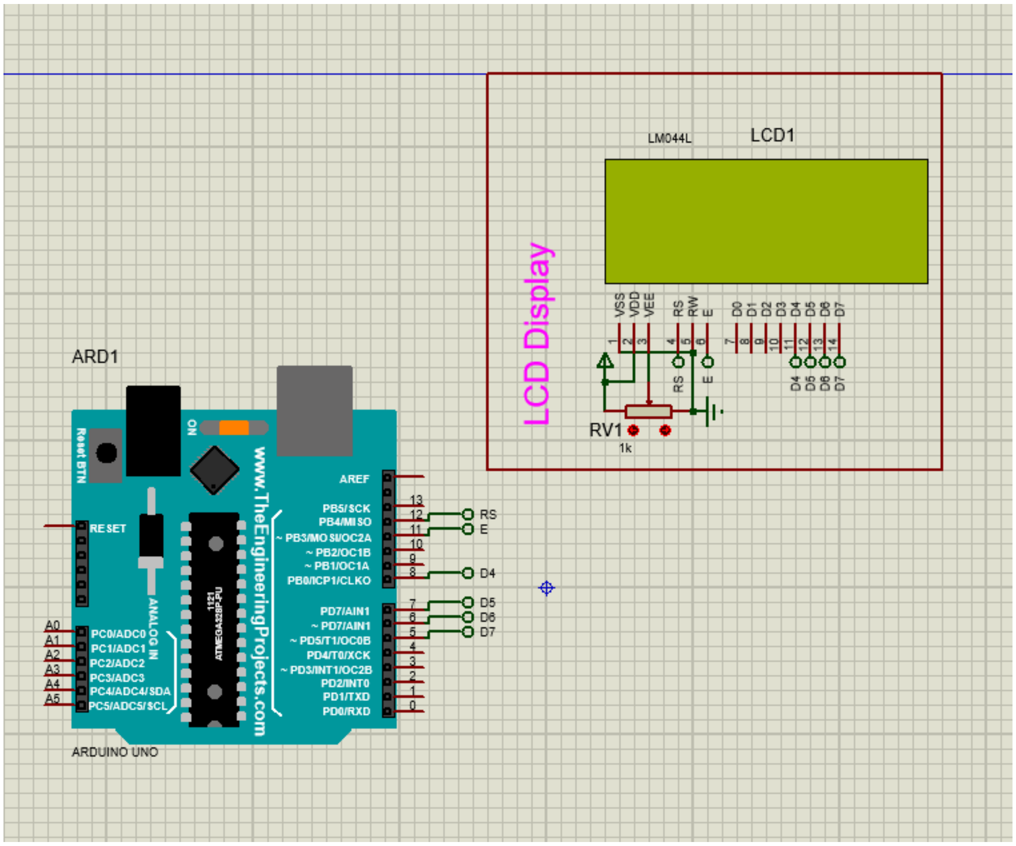

Figure 4. 7 LCD1 Interface with Arduino

4.4.5 Moisture sensor interface with arduino

To interface moisture sensor and arduion The Soil moisture sensor/the soil humidity sensor has four pins VCC, GND, Aout. These four pins can be used to get the soil moisture data from the sensor, the pinout of the Soil Moisture Sensor are as follows:

VCC is the power supply pin of the soil moisture sensor that can be connected to 3.3V or 5V of the supply. But do note that the analog output will vary depending upon the provided supply voltage.

GND is the ground pin of the board and it should be connected to the ground pin of the Arduino

Test pin

AOUT is the Analog output pin of the board that will give to arduino pin A0 an analog signal in between vcc and ground

We define the analog pin (A0) where the moisture sensor is connected. In the setup() function, we initialize the serial communication to print the moisture sensor values. In the loop() function, we call the readMoistureSensor() function to get the current moisture sensor value, and then print it to the serial monitor. The readMoistureSensor() function simply reads the analog value from the moisture sensor and returns it. The moisture sensor typically provides an analog output value that corresponds to the moisture level or water content in the soil.

The specific interpretation of the sensor value will depend on the characteristics of your particular moisture sensor. You may need to consult the sensor's datasheet or documentation to understand the relationship between the sensor value and the actual moisture level.

Set up thresholds to determine when the soil is dry, moist, or wet.

For example, you could define a threshold value (e.g., 500) and check if the sensor value is above or below that threshold to determine the moisture level.

Once the Arduino is connected to the moisture sensor and the sketch is uploaded, the system will start monitoring the moisture sensor value.

The current moisture sensor value will be printed to the serial monitor every second.

We modify the sketch to display the moisture value on an LCD, send it to a cloud platform, or use it for further processing or control purposes. This setup provides a basic interface between a moisture sensor and an Arduino Uno using analog pin A0.

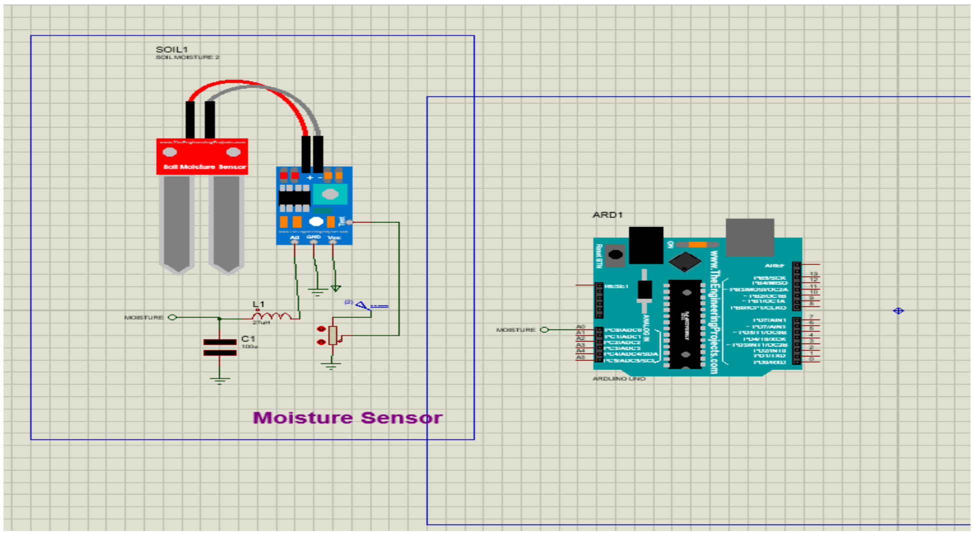

Figure 4. 8 Moisture sensor interface with arduino

4.4.6 Humidity sensor interface with arduino

To communicate humidity and arduino, connect the positive (or power) terminal of the DHT11 sensor to the 5V pin of the Arduino Uno. Connect the negative (or ground) terminal of the DHT11 sensor to the GND (ground) pin of the Arduino Uno. Connect the data terminal of the DHT11 sensor to analog pin A3 on the Arduino Uno.

We include the DHT library, which provides functions to interact with the DHT11 sensor.

We defined the analog pin (A3) where the DHT11 sensor is connected and the sensor type (DHT11). In the setup() function, we initialize the serial communication and the DHT sensor. In the loop() function, we read the humidity and temperature values from the DHT sensor, and then print them to the serial monitor. Use the dht.readHumidity() and dht.readTemperature() functions to read the sensor values. Check if the sensor readings are valid (not NaN) before printing the values.

If having issues with the sensor, make sure the connections are correct and the sensor is compatible with the Arduino Uno.

Once the Arduino is connected to the DHT11 sensor and the sketch is uploaded, the system will start monitoring the humidity and temperature values.

The current humidity and temperature values will be printed to the serial monitor every 2 seconds.

You can modify the sketch to display the sensor values on an LCD, send them to a cloud platform, or use them for further processing or control purposes.

This setup provides a basic interface between a DHT11 humidity sensor and an Arduino Uno using analog pin A3. You can further enhance the project by adding features like temperature and humidity thresholds, data logging, or integrating the sensor data with other systems or applications.

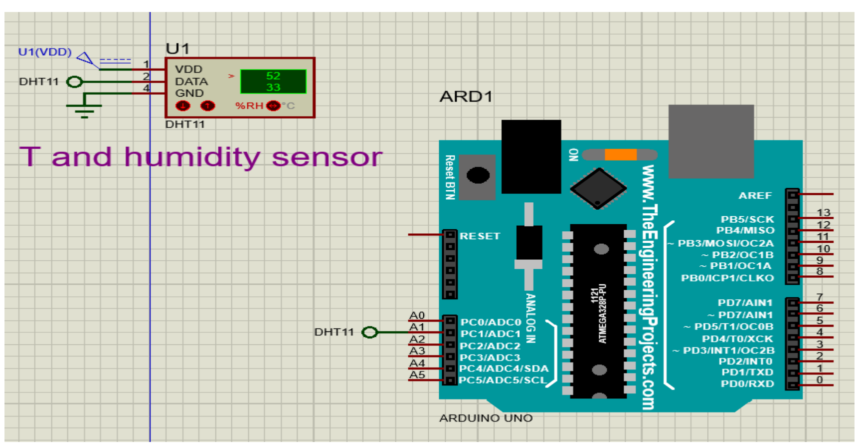

Figure 4. 9 Humidity sensor interface with arduino

4.4.7 Rain detected sensor interface with arduino

To communicate rain detector sensor and arduino, the test pin of rain sensor is connected into a logical state, vcc pin of sensor also connected into 5v pin of arduino and the GND pin of sensor connected into ground pin of arduino. The out pin of sensor connected into A1 pin of arduino and sends a data to microcontroller and decides a decision based on a given instructions. We defined the analog pin (A1) where the rain detected sensor is connected. In the setup() function, we initialize the serial communication to print the rain detected values. In the loop() function, we call the readRainSensor() function to get the current rain detected value, and then print it to the serial monitor. The readRainSensor() function simply reads the analog value from the rain detected sensor and returns it.

The rain detected sensor typically provides an analog output value that corresponds to the amount of rain or moisture detected. The specific interpretation of the sensor value will depend on the characteristics of your particular rain detected sensor.

Once the Arduino is connected to the rain detected sensor and the sketch is uploaded, the system will start monitoring the rain detected value. The current rain detected value will be printed to the serial monitor every second.

This setup provides a basic interface between a rain detection sensor and an Arduino Uno

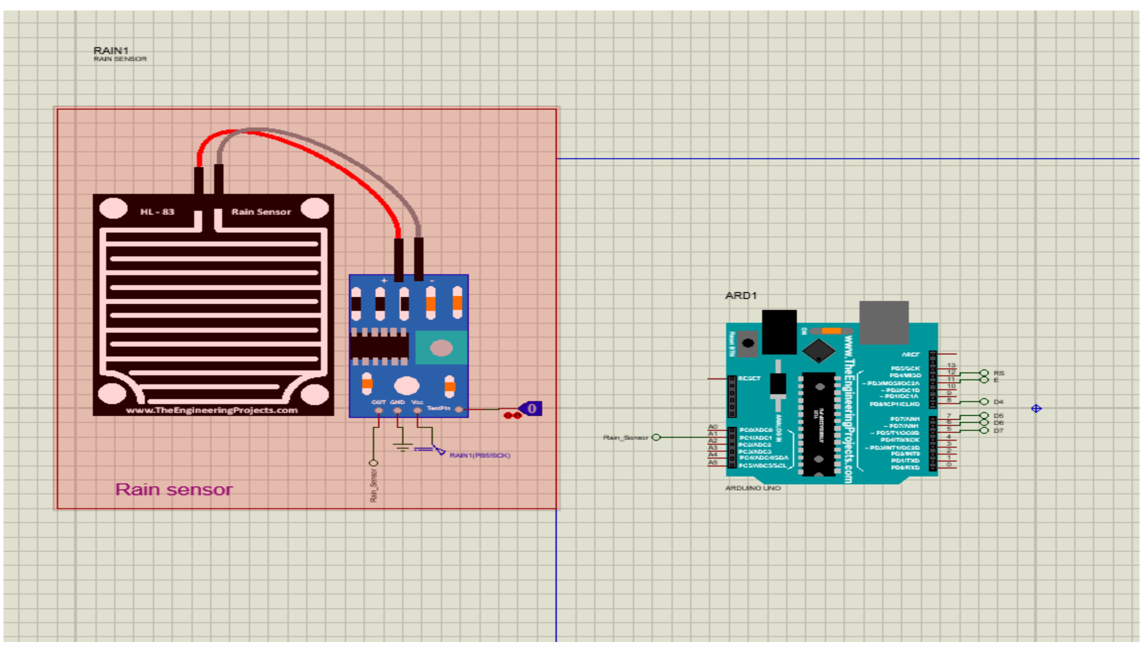

Figure 4. 10 Rain detected sensor interface with arduino

4.4.8 PH meter sensor interface with arduino

To communicate soil PH sensor with arduino, connect the positive (or signal) terminal of the pH sensor to analog pin A3 on the Arduino Uno. Connect the negative (or ground) terminal of the pH sensor to the GND (ground) pin of the Arduino Uno. Connect the power terminal of the pH sensor to the 5V pin of the Arduino Uno. We defined the analog pin (A3) where the pH sensor is connected. In the setup() function, we initialize the serial communication to print the pH values. In the loop() function, we call the readpH() function to get the current pH value, and then print it to the serial monitor. The readpH() function reads the analog value from the pH sensor, converts it to a voltage value, and then calculates the pH value using a simple formula.

PH sensors need to be calibrated to provide accurate readings. Typically, we would use standard pH buffer solutions to calibrate the sensor. The formula used in the readpH() function may need to be adjusted based on the specific characteristics of your pH sensor. You may need to consult the sensor's datasheet or documentation for the appropriate calibration formula.

Once the Arduino is connected to the pH sensor and the sketch is uploaded, the system will start monitoring the pH value. The current pH value will be printed to the serial monitor every second.

- If the pH value is between 7 and 8, it displays “Neutral.”
- If the pH value exceeds 8, it indicates “Alkaline.”
- If the pH value drops below 7, it indicates “Acidic.”

The LCD and virtual terminal displays “Alkaline” and “Very Alkaline” as the pH reading rises over certain thresholds. On the other hand, as the pH value falls below particular pH values, it signifies “Acidic” and “Very Acidic.”

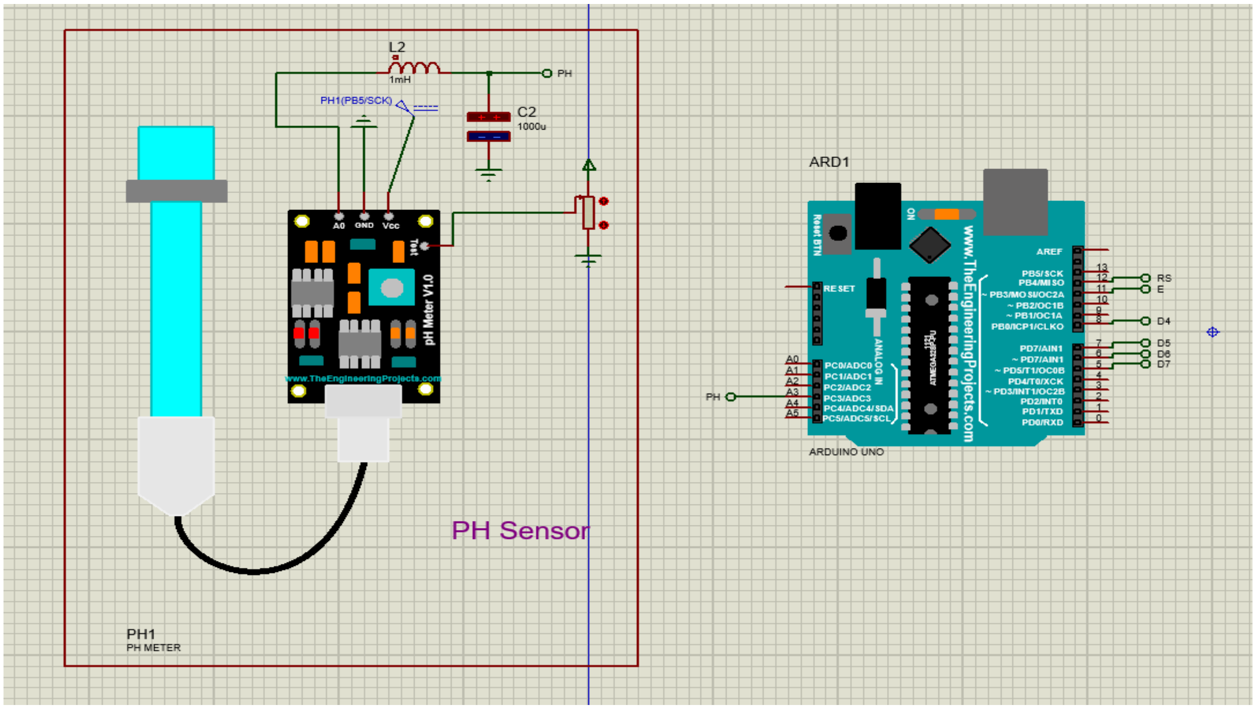

Figure 4. 11 PH meter sensor interface with arduino

4.4.9 Leds water level indicator

To communicate leds with arduino,connect the positive terminal of the first LED (representing the lowest water level) to pin 2 of the Arduino Uno. Connect the positive terminal of the second LED (representing a higher water level) to pin 9 of the Arduino Uno. Connect the positive terminal of the third LED (representing the highest water level) to pin 13 of the Arduino Uno. Connect the negative terminals of all the LEDs to the GND pin of the Arduino Uno. Connect a water sensor or a set of conductive probes to the Arduino Uno analog pins (e.g., A0, A1, A2) to detect the water level.

We defined the pin numbers for the three LEDs. In the setup() function, we set the LED pins as outputs.

In the loop() function, we read the analog values from the ultrasonic sensor.

We use a simple threshold (e.g., 50%, 30%) to determine whether the water level has reached the corresponding sensor. If the sensor value is above the threshold, we turn on the corresponding LED.

Once the Arduino is connected to the LEDs and the water sensors, and the sketch is uploaded, the system will start monitoring the water level.

As the water level changes, the appropriate LED(s) will light up to indicate the current water level.

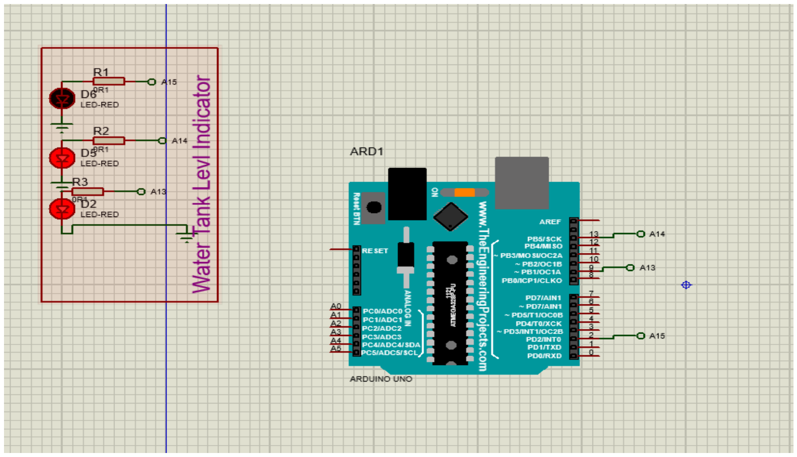

Figure 4. 12 Leds water level indicator

4.4.10 GSM module with arduino interfacing

To communicate GSM module with arduino, connect the TX pin of the GSM module to the RX pin (pin 0) of the Arduino Uno.Connect the RX pin of the GSM module to the TX pin (pin 1) of the Arduino Uno. Connect the GND pin of the GSM module to the GND pin of the Arduino Uno. Connect the VCC pin of the GSM module to the 5V pin of the Arduino Uno. SoftwareSerial.h library, which allows us to create a software serial port on any digital pins of the Arduino.We create a SoftwareSerial object named gsm and pass the RX and TX pins (1 and 0, respectively) as parameters. In the setup() function, we initialize the Serial communication at 9600 baud rate and the gsm serial at the same baud rate. In the loop() function, we check if there's data available on the gsm serial and write it to the Serial monitor. Similarly, we check if there's data available on the Serial monitor and write it to the gsm serial.

Once the Arduino is connected to the GSM module and the sketch is uploaded, start communicating with the GSM module through the Serial monitor.

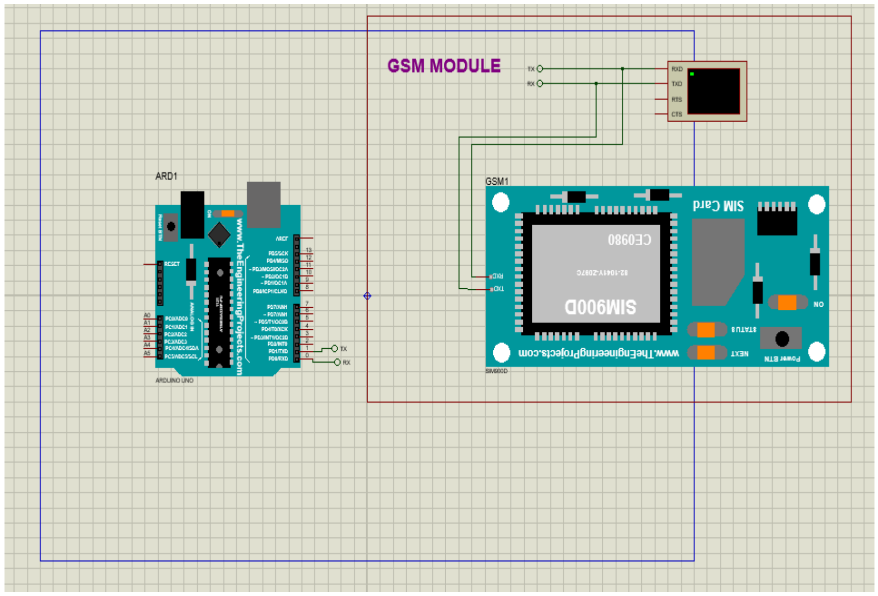

Figure 4. 13 GSM module with arduino interfacing

4.4.11 Ultrasonic with arduino interfacing

To communicate Ultrasonic with arduino, connect the VCC pin of the Sensor to the 5V pin of your Arduino. Connect the GND pin of the Sensor to any of the GND pins on the Arduino. Connect the ECHO pin of the Sensor to the pin 9 on the Arduino. Connect the TRIG pin of the Sensor to the pin 10 on the Arduino. We define two constants, trigPin and echoPin, to represent the Trig and Echo pins of the Ultrasonic sensor, respectively. In the setup() function, we set the Trig pin as an output and the Echo pin as an input, and we start the Serial communication at 9600 baud rate.

In the loop() function:

We clear the Trig pin by setting it to LOW. Set the Trig pin to HIGH for 10 microseconds to trigger the Ultrasonic sensor. Read the Echo pin using the pulseIn() function, which returns the duration of the sound wave travel time in microseconds. Calculate the distance in centimeters using the formula distance = duration * 0.034 / 2. Print the distance to the Serial Monitor.

Once the Arduino is connected to the Ultrasonic sensor and the sketch is uploaded, you can open the Serial Monitor to see the distance measurements.

The distance will be displayed in centimeters (cm) and will be updated every 100 milliseconds.

The ultrasonic sensor works by emitting high-frequency sound waves and measuring the time it takes for the waves to reflect off the water surface and return to the sensor.

Based on the time-of-flight calculation, the sensor can determine the distance to the water surface, which can then be converted to the actual water level in the reservoir.

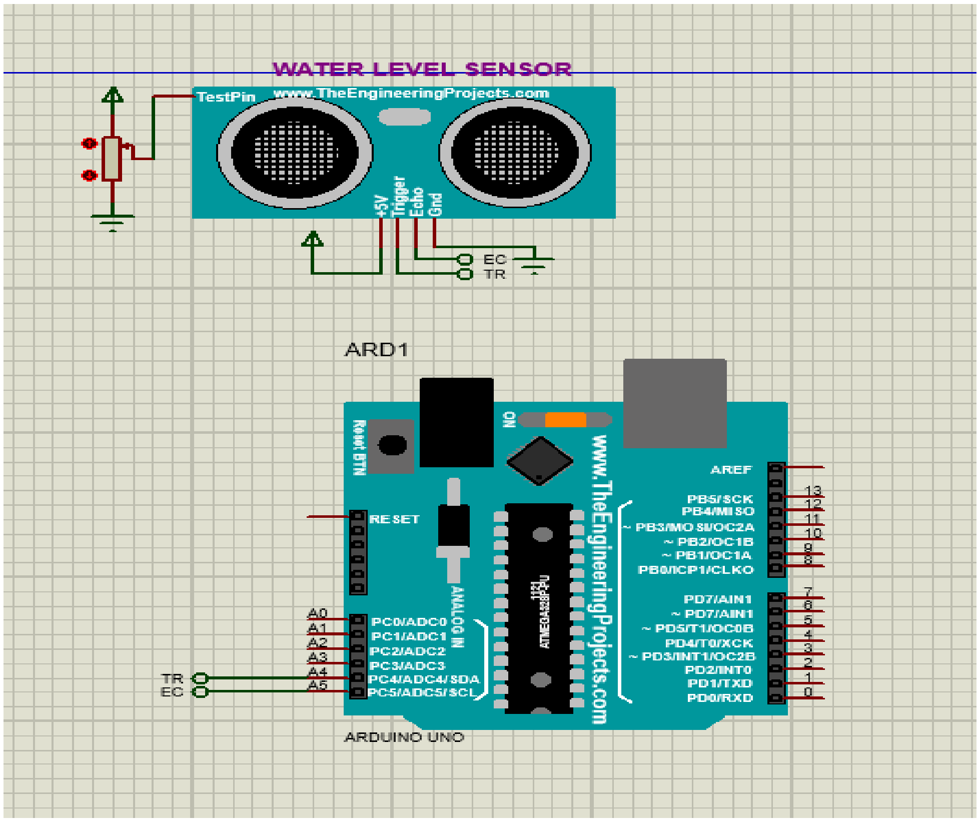

Figure 4. 14 Ultrasonic with arduino interfacing

4.4.12 Com port interface with arduino

Connecting the Arduino pins to the COM port: Connect Arduino pin 0 (RX2) to the TX2 (transmit) pin of the COM port. Connect Arduino pin 1 (TX2) to the RX2 (receive) pin of the COM port. Also, connect the ground (GND) of the Arduino to the ground of the COM port. Open the COM port settings (e.g., in Windows, you can find this in the Device Manager). Ensure that the other settings (data bits, stop bits, parity) match the default settings of the Arduino serial communication. When we upload the Arduino code and open a serial communication terminal (like the Arduino IDE's Serial Monitor), you should be able to send data from the COM port to the Arduino and vice versa.

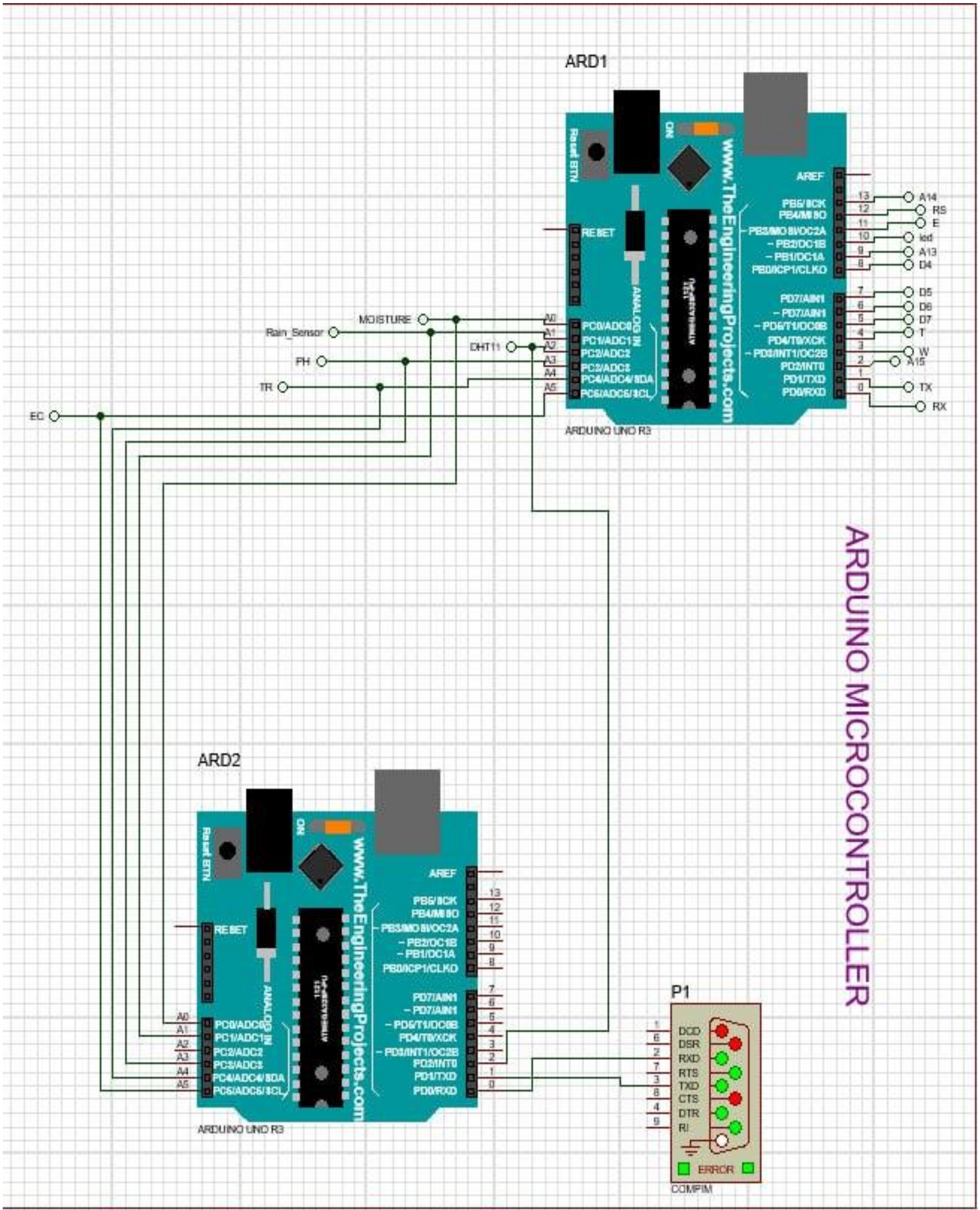

Figure 4. 15 Connecting the Arduino pins to the COM port

4.4.13 Circuit interconnection

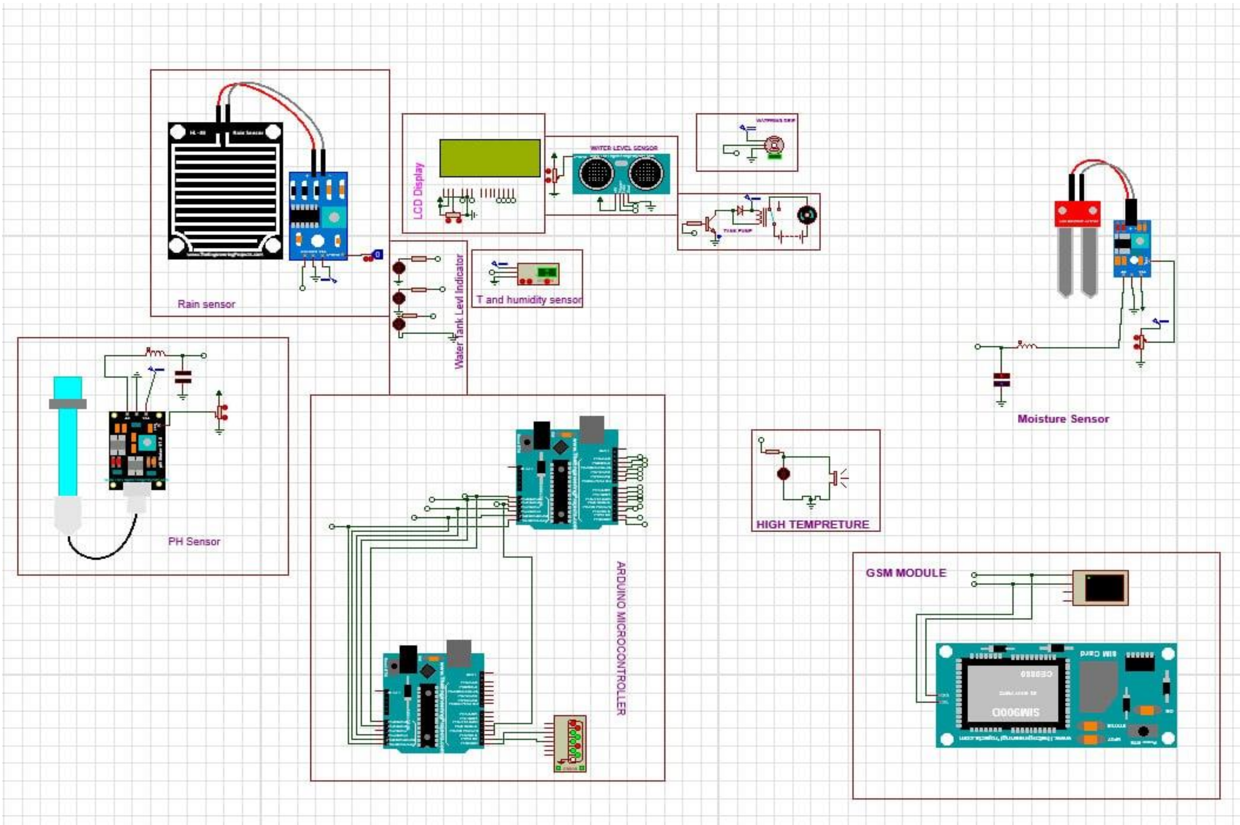

Figure 4. 16 All circuit interconnection

4.4.14 Power supply

Power supply is the circuit from which we get a desired dc voltage to run the other circuits. The voltage we get from the main line is 230V AC but the other components of our circuit require 5V DC. Hence a step-down transformer is used to get 12V AC which is later converted to 12V DC using a rectifier.

The output of rectifier still contains some ripples even though it is a DC signal due to which it is called as Pulsating DC. To remove the ripples and obtain smoothed DC power filter circuits are used. A 5V regulated supply is taken as followed.

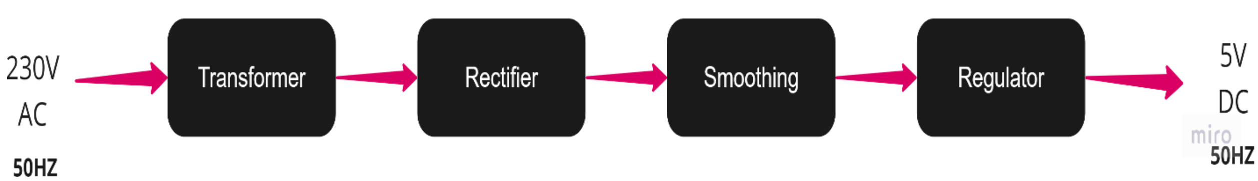

Figure 4. 17 Block diagram of regulated power supply system

- Transformer – steps down high voltage AC mains to low voltage AC.
- Rectifier – converts AC to DC, but the DC output is varying.
- Smoothing – smoothed the DC from varying greatly to a small ripple.
- Regulator – eliminates ripple by setting DC output to a fixed voltage.

Almost all electronics circuit required DC power supply. DC power supply is the circuit which converts the AC wave form of power lines to direct voltage of constant amplitude. An ideal regulated power supply is designed to provide a pre-determined Dc voltage which is independent of the current drown from the source. These circuits are special class of feedback amplifiers. All the benefits of TCs‟ are thus obtained: excellent performance small size, ease of use, low cost, and high reliability. Unregulated power supply has many disadvantages due to which it is not sufficient for manyapplications.

- Poor regulation.
- Dc output voltage varies with the AC input.
- DC output voltage variation varies with temperature because of semiconductor use to overcome the above disadvantages.

A 5V-dc power requirement will be used as input supply to the system. The choice of using a transformer is due to the low voltage requirement of the system. A transformer of 240/12V in conjunction with a regulator will be able to provide the needed input 5Vdc.

This means that the RMS value of the transformer secondary is Vrms = 12Vac.

The whole section of the project is powered from a 5V dc power source. To achieve this 5-volt output, a variable output adapter is used.

The adapter takes in 240Vac and gives out from its variable tapped output V dc, 4.5Vdc, 9Vdc, 12Vdc; the output to the required 5Vdc, the output of the adapter is passed through the regulator that makes sure that at any point in time, the output it gives is 5V. For convenience, we tap the output ofthe adapter and hence the input to the regulator at 6Vdc.

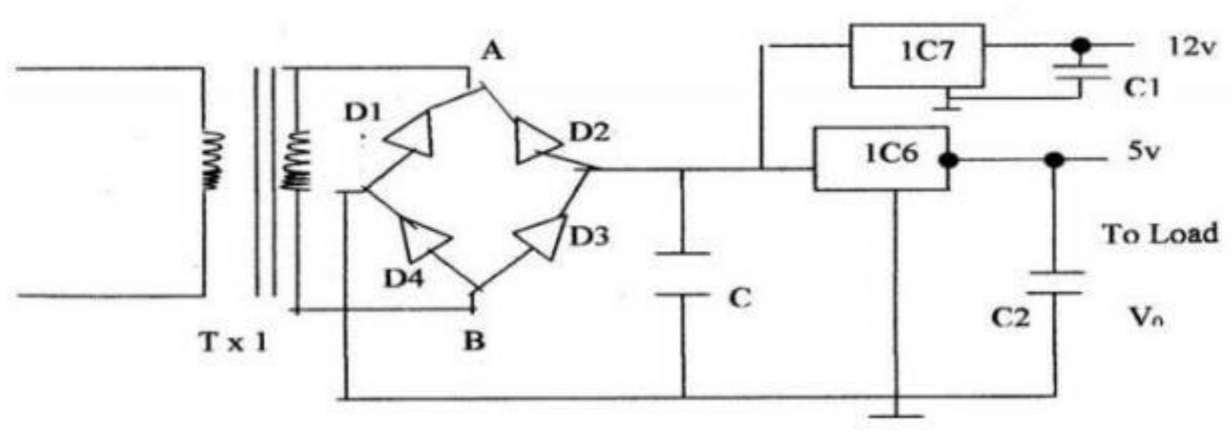

Figure 4. 18 Regulated power supply

- Step down transformer

Transformer [4] is the electrical device that converts one voltage to another with little loss of power. Most power supplies use a step-down transformer to reduce the dangerously high mains voltage to a safer low voltage. Here a step down transformer is used to get 12V AC from the supply i.e. 230V AC.

Design 50 Hz, to 1000 VA single phase transformer step down transformer from 220v to 12v. show With necessary Calculations.

Area of core

Ai=1/4.44 f.Bm.Te

For a small transformer Bm=1 to 1.2, we use 1.2Te=4 turn per volt

f=50 Hz

Where, Ai=area of core

f=operation frequencyBm=turn per volts

Te= turn per volts

φ =1.61×10^_3 = 1.45 inch^2Ai=1/50 Hz × 1.2 × 4 × 4.4

1 inch^2=6.4516×10^-4_4m^2

So, 1.45 inch^2 =0.0009359 m^2

Vp=220v

Ip=VA/VP=1000 VA/220V=4.545Amp

Copper conductor = IP/8= 4.545A/3.5A/mm^2

=1.3mm^2

E1=4.44f φ N1

N1=E1/4.44f$ = 220/4.44×50×1.6×10^_3

=0.6155×10^3

N1=616 Turns

Vs=12V

I2=1000/12V = 83.33A

E2=4.44f φ N2

N2=E2/4.44f φ =12V/4.44×50× φ

N2=33.57 turns = 34 turns

A Copper conductor =I2/8=83.33A/3.5A/mn^2

= 23.8 mm^2

Bridge rectifier

A rectifier is a circuit that converts AC signals to DC. A rectifier circuit is made using diodes. There are two types of rectifier circuits as Half-wave rectifier and Full-wave rectifier depending Upon the DC signal generated. Here Full-wave bridge rectifier is used to generate dc signal.

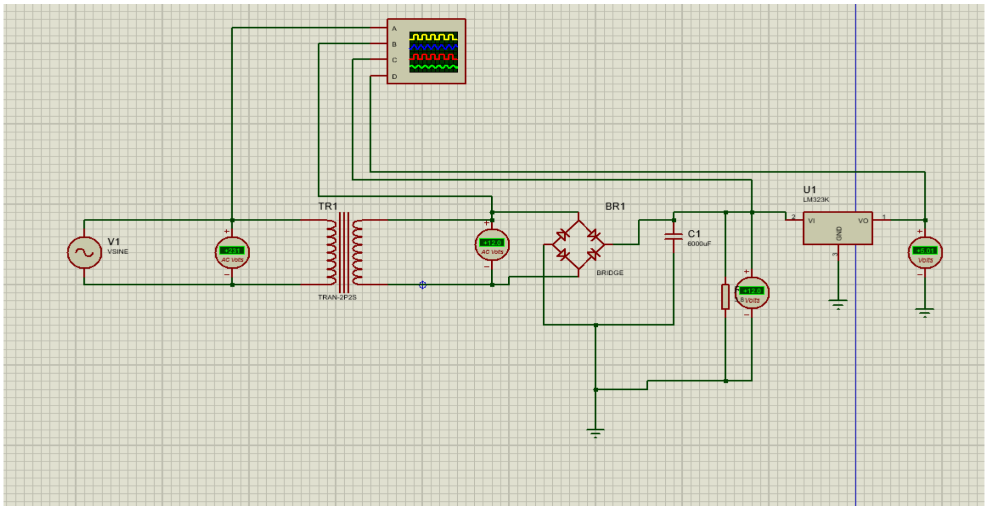

Figure 4. 19 bridge rectifier circuit

- **Smoothing**

Smoothing is performed by a large value electrolytic capacitor connected across the DC supply to act as reservoir, supplying current to the output when the varying DC voltage from the rectifier is decreasing. The diagram shows the unsmoothed varying DC and the smoothed DC. The capacitor charges quickly to the peak of the varying DC and then discharges as it supplies current to the output. Here the capacitor of 470uF is used as a smoothing circuit.

- **The voltage regulator**

Fixed voltage regulator78xx, produce fixed DC output voltage from variable DC (a small amount of AC on it). Fixed output is obtained by connecting the voltage regulator at the output of the filtered DC. It can also be used in circuits to get low DC voltage from high DC voltage.

4.5 SOFTWARE DESIGN

To be able to interpret the different states of the soil as prompted by the soil sensor the microcontroller was programmed. The arduino integrated development environment (IDE) was used. The idea is based on C++ and thus can be extended using C++ libraries. Arduino programs (sketches) are cross platform, Simple, clear and at the same time flexible for advanced programmers. The project sketch is attached at the appendices page.

4.5.1 Program pseudo code

READ sensor value

COMPARE sensor value with set threshold

IF sensor value > maximum set value

TURN-ON pump

DISPLAY soil condition on LCD

LIGHT dry soil LED

Send SMS text to user

ELSE IF sensor value < maximum set value > minimum set value

TURN-OFF pump

DISPLAY soil condition on LCD

LIGHT moist soil LED

Send SMS text to user

ELSE IF sensor value < minimum set value

TURN-OFF pump

DISPLAY soil condition on LCD

LIGHT soggy soil LED

Send SMS text to use

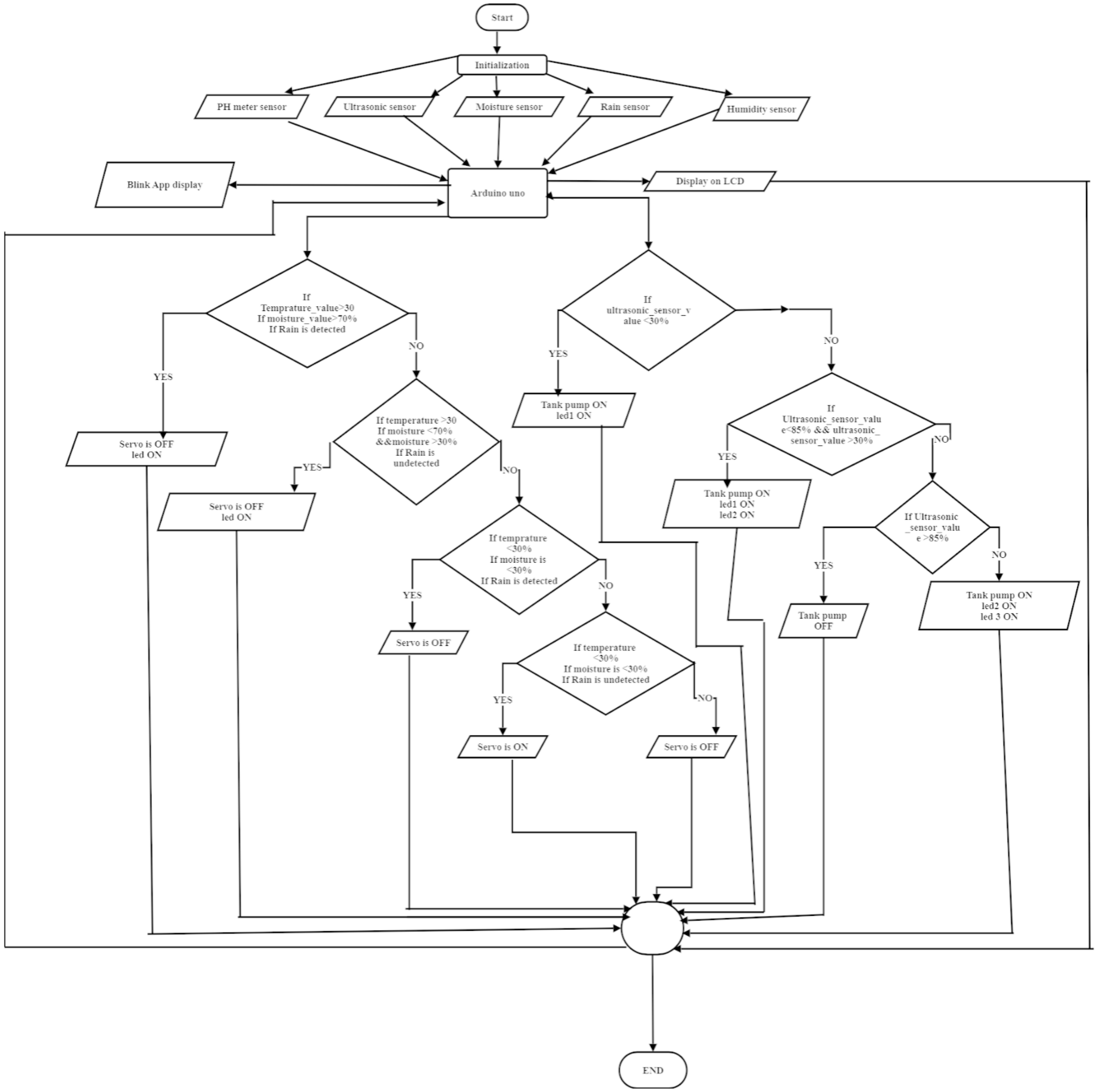

Figure 4. 20 Program flow chart

CHAPTER 5

RESULTS AND DISCUSSION

5.1 RESULTS AND DISCUSSION

After have been designed successfully, the circuit were constructed based on our design in order to simulate using selected simulation proteus software, which is based on the process of modeling a real phenomenon with a set of mathematical formulas. It is, essentially, a program that allows the user to observe an operation through simulation without actually performing that operation. Proteus 8 professional assisted by Arduino IDE have been used to simulate the whole work. Then, the simulation is compiled and the result has been observed at different soil moisture, water level, humidity, temperature, PH and rain sensor values.

- Case 1:-When the water content of the soil is high, the water tanker is full, rain is undetected, and temperature operates on normal condition. The microcontroller checks the availability of water in the soil and receives in terms of voltage from soil moisture sensor and at the same time it checks the water level of the tanker and gives low output through Arduino pin 4 that makes the water pump off. This indicates the normal water content of the soil.

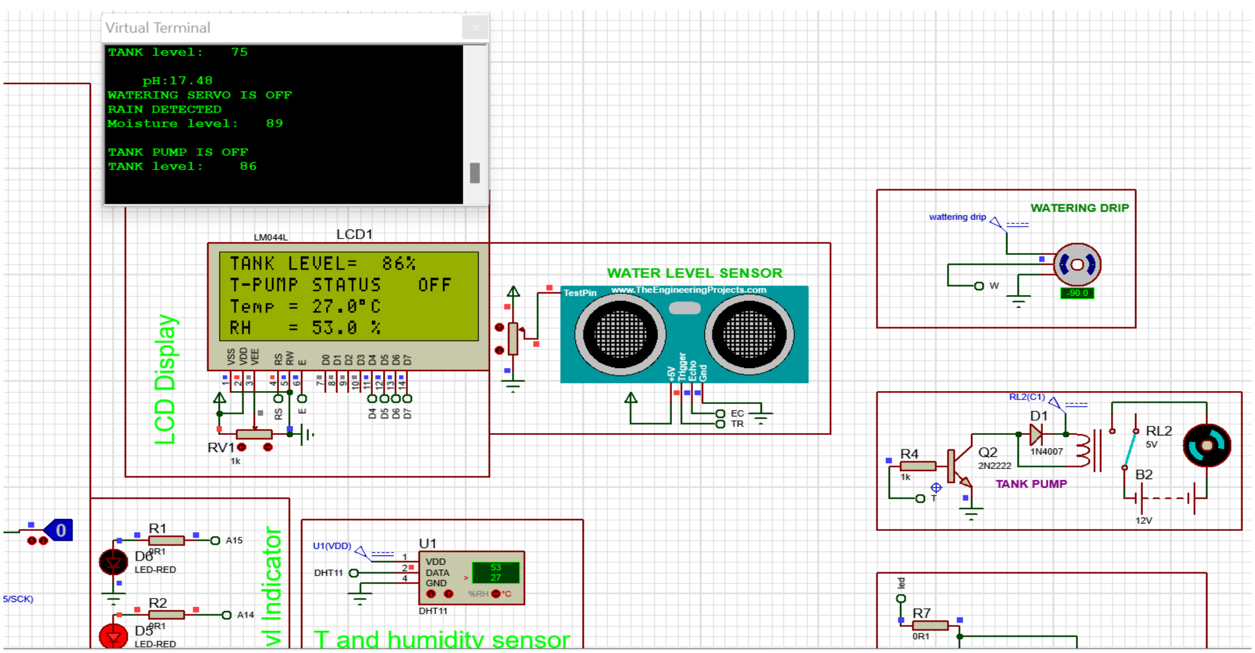

Figure 5. 1 Simulation result for at normal content condition

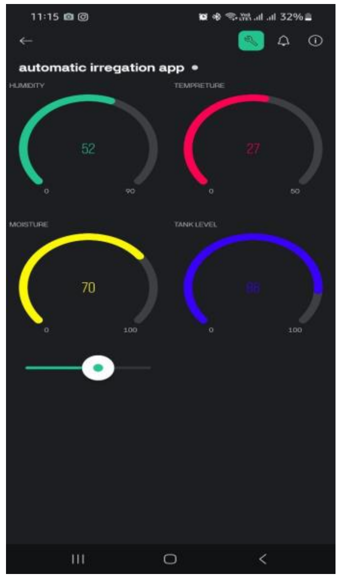

Figure 5. 2 Blink simulation for the above condition

- Case 2: when the soil is dry, water tanker is full,tempreture is on normal condition and rain is undetected. In this case the microcontroller gives high output through Arduino pin 3 that makes the servo motor becomes rotate 180^o(ON). At this condition the servomotor motor starts irrigate the farm field.

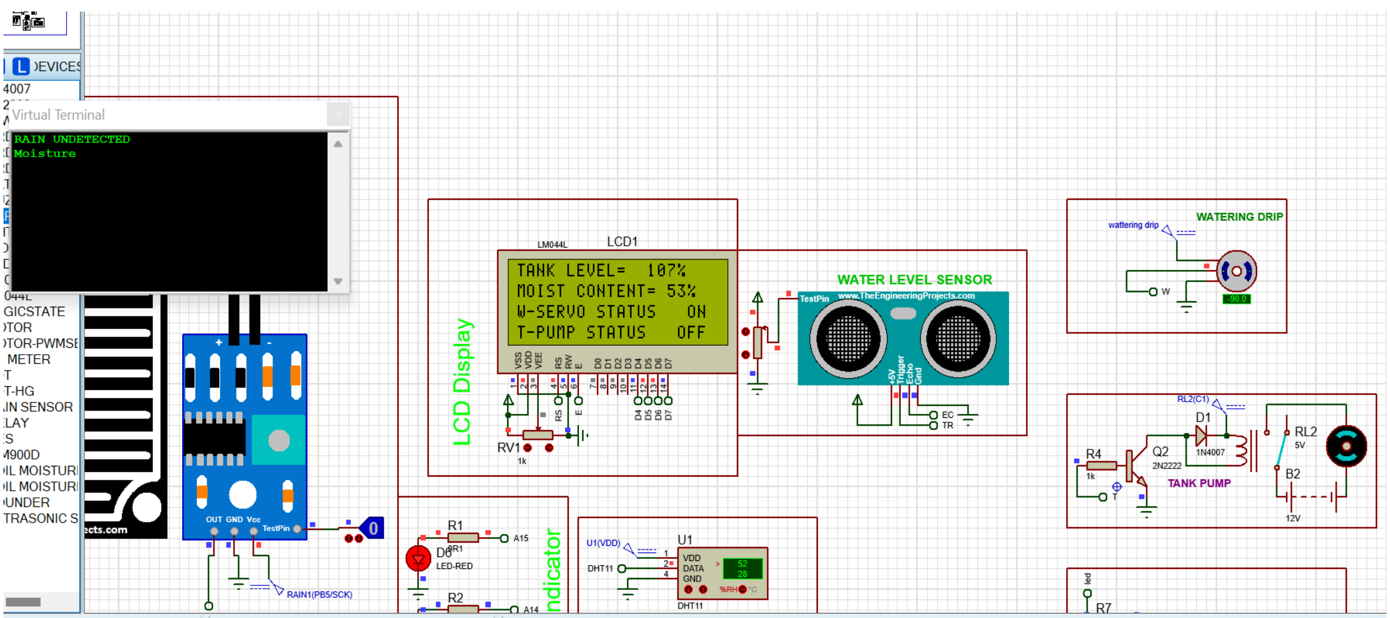

Figure 5. 3 Simulation result at soil dry and tanker full condition

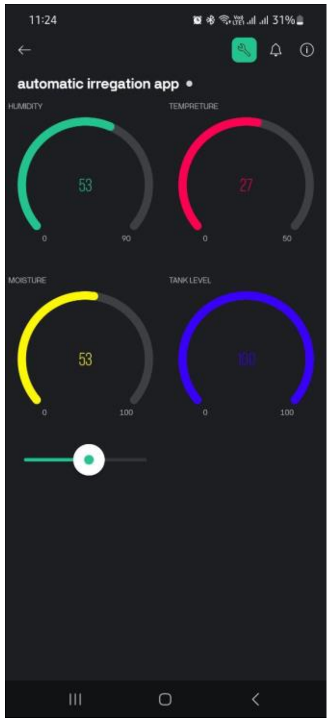

Figure 5. 4 Blink simulation result at soil dry and tanker full condition

- Case 3: when the rain is undetected, temperature is less than 30, soil is dry and water tanker is empty. In this case the microcontroller gives high output through Arduino pin 3 that makes the servo motor becomes rotate 180^o(ON) and Arduino pin 4 that makes the relays connected to a pump will be energized through the transistor and then water pump becomes ON and display on LCD and mobile app.

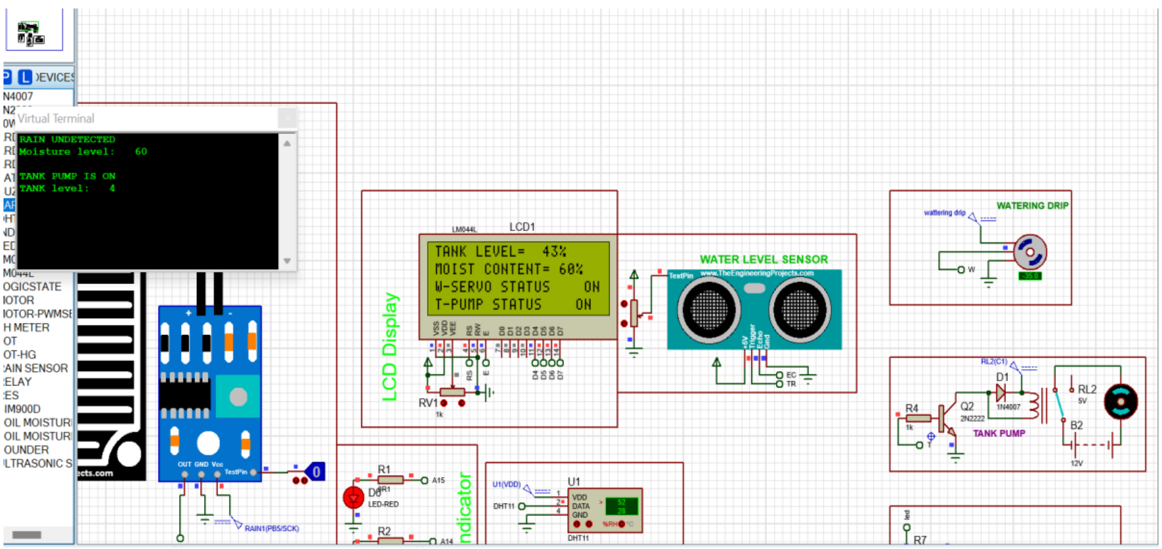

Figure 5. 5 Simulation result at soil is dry and tanker empty condition

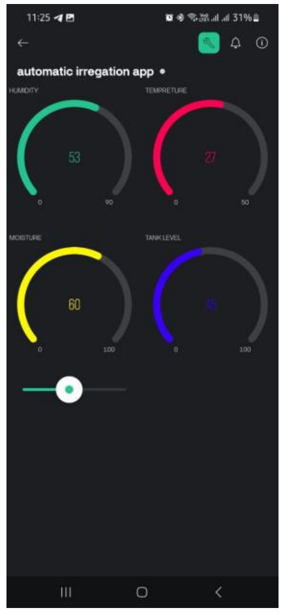

Figure 5. 6 Blink simulation at soil is dry and tanker empty condition

- Case 4 : when the temperature is high the microcontroller gives low output through Arduino pin 3 that makes the servo motor becomes rotate 0^{o}(OFF) and Arduino pin4 that makes the relays connected to a pump will be de-energized through the transistor and then water pump becomes OFF and the lcd and buzzer will be ON.

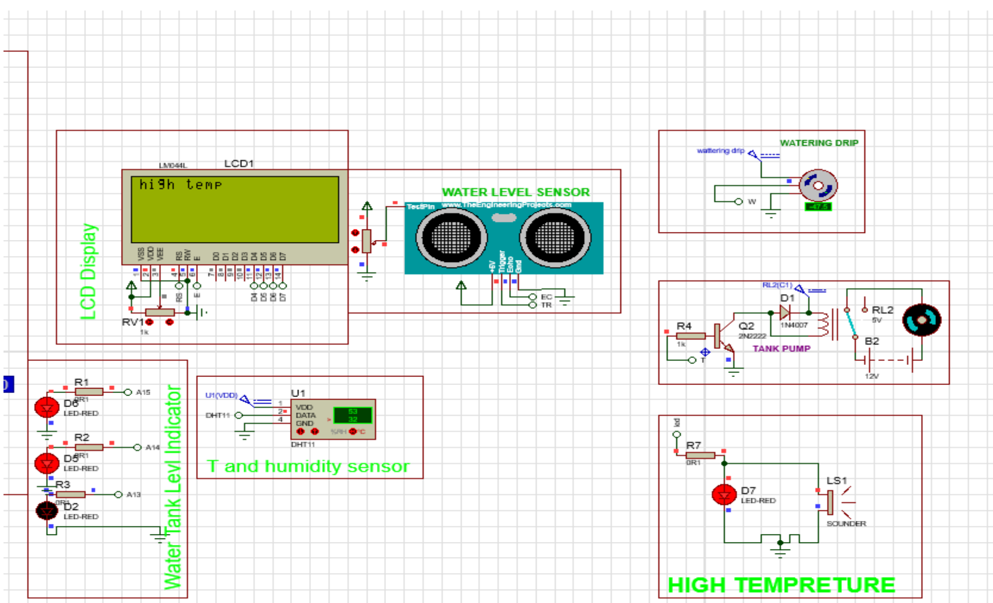

Figure 5. 7 when Temperature is high

- Case 5 when the rain is detected in these case the arduino microcontroller gives low output for tank pump and watering servo so overall opration will stop until the rain is undetected the irrigation system will watering on the rain only

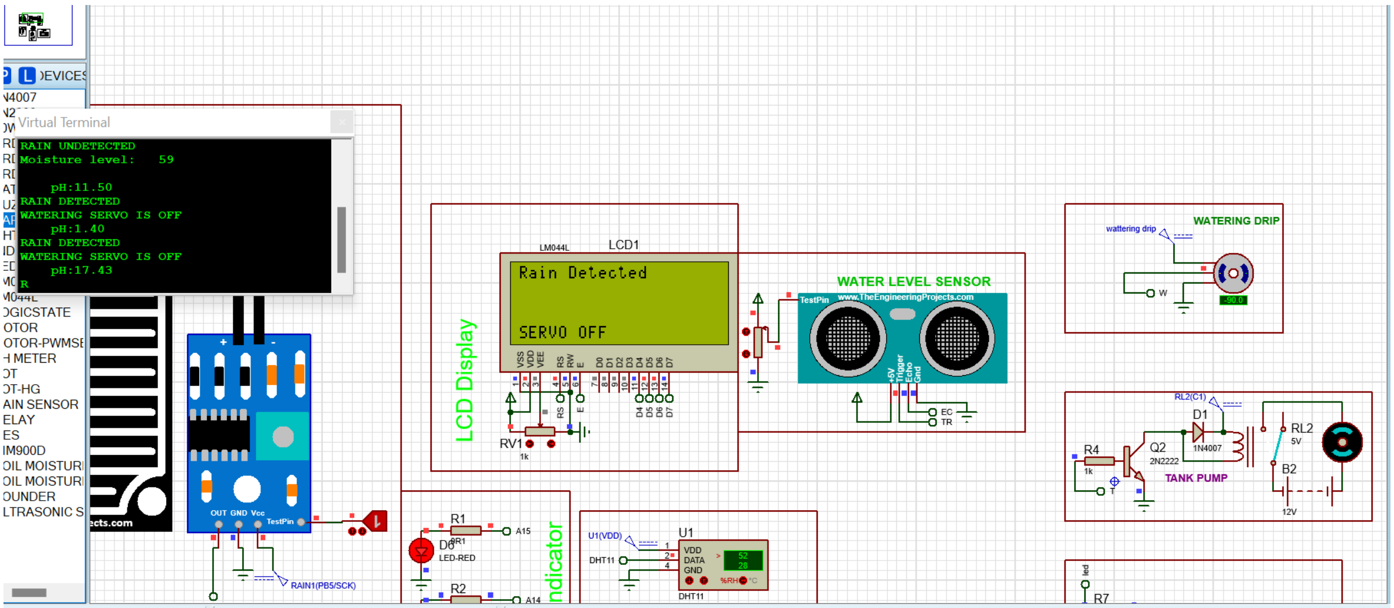

Figure 5. 8 when the rain is detected

CHAPTER 6

CONCLUSION AND RECOMMENDATION FOR FUTUREWORK

6.1 CONCLUSION

From our work we conclude that, the described work involves simulating a IOT-based automatic drip irrigation system, where in a motor regulates water flow based on field conditions. An LCD screen provides real-time updates on water level, moisture content, pH, temperature, and motor status. LEDs indicate field conditions visually, while the controller communicates with users via a GSM module, sending messages about system status. The integration of the Blink application facilitates user connection to the drip irrigation system. Through Blink, users can remotely access and monitor temperature, moisture content, pH levels, and water tank status. This technology enables users to make informed decisions regarding irrigation management. By automating irrigation processes and providing remote access, the system enhances efficiency and reduces manual labor. Additionally, it optimizes resource usage by delivering water precisely as needed, minimizing waste and maximizing crop yield. The real-time monitoring capabilities ensure timely intervention in case of any irregularities, promoting sustainable and effective agricultural practices. Overall, the system represents a significant advancement in irrigation technology, promising improved productivity and resource management for agricultural operations.

The described simulation highlights the functionality of a cutting-edge GSM-based automatic drip irrigation system. Its core operation involves a motor that adjusts water flow based on field conditions, such as moisture level, pH, and temperature. These parameters, along with motor status, are displayed in real-time on an LCD screen for user convenience. Visual indicators in the form of LEDs provide additional feedback on field conditions. The system is designed for seamless user interaction, with a controller that communicates critical updates via a GSM module.

The integration of the Blink application enhances user accessibility and control over the irrigation process. Through this platform, users can remotely monitor and manage various aspects of the system, including temperature, moisture content, pH levels, and water tank status. Such remote access empowers users to make informed decisions regarding irrigation scheduling and resource allocation.

By streamlining irrigation operations and providing accurate, real-time data, this system offers numerous benefits. It reduces manual labor and optimizes water usage, thereby minimizing costs and environmental impact. Additionally, by ensuring precise water delivery tailored to specific field conditions, it promotes crop health and yield. Overall, this simulation represents a significant advancement in agricultural technology, promising increased efficiency, productivity, and sustainability in irrigation practices.

6.2 RECOMMENDATIONS

6.2.1 Recommendation for woldia university

We recommend for Woldia University to give as hardware components and implement in the tangible way to improve our hardware skills and give experience for students.

To ensure future success in irrigation projects, it is highly recommended to prioritize the condition of fertilizer. The process of giving the fertilizer using check amount of PH value and flow in the form of liquid like as water. The next recommendation will be exploration and implementation of alternative power sources, by actively seeking out renewable energy options such as solar, wind, or hydropower irrigation systems can become more sustainable, cost-effective, and environmentally friendly. Embracing alternative power sources will not only reduce reliance on conventional energy methods but also contribute to the long-term viability and efficiency of irrigation projects.

REFERENCE

[1]Agbetuyi, A., et al. (2016). "Design and implementation of an automatic irrigation system based on monitoring soil moisture." 16: 206-215.

[2] Ghodake, R. and A. Mulani (2018). Microcontroller Based Automatic Drip Irrigation System.

[3] Guerbaoui, M., et al. (2013). "Pc-based automated drip irrigation system." International Journal of Engineering Science and Technology **5**(1): 221-225.

[4] Swagatam. (2023, November 20). *How to make step down transformers*. Homemade Circuit Projects. https://www.homemade-circuits.com/how-to-make-transformers/

[5] Abidin and Ibrahim, "used a web-based application to help users monitor the parameters such as water level, the flow of valves and pipes, and the entire operation of the automated fertigation system.," 2015.

[6] Kalwinder, "The sensors are easy to install and require very less maintenance.," 2013.

[7] Arduino tutorial - JavaTPoint. (12/08/2016.). www.javatpoint.com. https://www.javatpoint.com/arduino

[8] Snow Raindrops Humidity Rain Weather Detect Sensor module. (06/09/2016). Elecbee Factory. https://www.elecbee.com/en-24807-3pcs-Snow-Raindrops-Humidity-Rain-sensor.

[9] 'Internet of thing (IOT) based Mobile Plant Irregation Application' (2023) *International Journal of Advanced Natural Sciences and Engineering Researches* [Preprint]. doi:10.59287/as-ijanser.196.

APENDIX

```
#include<LiquidCrystal.h>
#include <SoftwareSerial.h>
#include <DHT.h>
#include <Servo.h>
#define echo A5
#define trigger A4
#define tank_pump 4
#define DHTTYPE DHT11
#define DHTPIN A2
#define SensorPin A3          // the pH meter Analog output is connected with the Arduino’s Analog
#define soilsensor A0
unsigned long int avgValue;  //Store the average value of the sensor feedback
float b;
int buf[10],temp;
Servo myservo;
long duration;
int distance;
int Temp;
int dht11__sensor;
int moisture_value;
int distance_percent;
int percentage;
SoftwareSerial SIM900(2, 3);
LiquidCrystal lcd(12,11,8,7,6,5);
//const int DHT11_Sesnor =A2 ;
const int rain_Sesnor = A1;

const int led3 = 2;
const int led4 = 13;
const int led5 = 9;
const int led6 = 10;

int rain_Sesnor_value;
float humudity_value,temprature_value;
char temperature[] = "Temp = 00.0 C  ";
char humidity[]    = "RH   = 00.0 %  ";
float maxVoltage = 5.0; // Assuming a 5V voltage reference
DHT dht(DHTPIN, DHTTYPE);
void setup () {
lcd.begin(20,4);
dht.begin();
SIM900.begin(9600);
Serial.begin(9600);
pinMode(echo,INPUT);
```

```
pinMode(soilsensor,INPUT);
pinMode(trigger,OUTPUT);
pinMode(SensorPin,INPUT);
pinMode(rain_Sesnor,INPUT);
digitalWrite(trigger,LOW);
pinMode(tank_pump,OUTPUT);
  pinMode(led6, OUTPUT);
  pinMode(led3, OUTPUT);
  pinMode(led4, OUTPUT);
  pinMode(led5, OUTPUT);
myservo.attach(3);
myservo.write(0);
digitalWrite(tank_pump,LOW);
lcd.setCursor(0,1);
lcd.print("DRIP IRRIGATION PROJECT"    );
lcd.setCursor(0,2);
lcd.print("  WOLDIA UNIVERSITY");
lcd.setCursor(0,3);
lcd.print("     GROUP 5");
delay(500);
lcd.clear();
}
void loop(){

 // LEVEL SENSOR
 digitalWrite(trigger,LOW);
 delayMicroseconds(2);
 digitalWrite(trigger,HIGH);
 delayMicroseconds(10);
 digitalWrite(trigger,LOW);
 duration=pulseIn(echo,HIGH);
 distance=duration*0.017;
 distance_percent=map( distance,0,1023,0,100);
 int moisture_value=analogRead(soilsensor);
 percentage = ( 100 - ( (moisture_value/1023.00) * 100 ) );
 condition();
 ultrasonic();
 readDTH11_Sesnor();
 ph_sensor();
}
void ultrasonic()
{
if(distance_percent <=10 ) /* Tank is empty */
     {
         LCD_1();
         digitalWrite(tank_pump, HIGH);
         digitalWrite(led3, LOW);
```

```
    delay(10);
    digitalWrite(led4, LOW);
    delay(10);
    digitalWrite(led5, LOW);
    sms3();
    delay(10);

}
else if((distance_percent >= 25) and (distance_percent <= 34) )  /* more than 20 % tank filled */
{
    digitalWrite(tank_pump, HIGH);
    LCD_1();
    digitalWrite(led3, HIGH);
    delay(10);
    digitalWrite(led4, LOW);
    delay(10);
    digitalWrite(led5, LOW);
    sms3();
    delay(10);
}
else if((distance_percent >= 35) and (distance_percent <= 49) ) /* more than 40 % tank filled */
{
    digitalWrite(tank_pump, HIGH);
   LCD_1();
    digitalWrite(led3, HIGH);
    delay(10);
    digitalWrite(led4, HIGH);
    delay(10);
    digitalWrite(led5, LOW);
    sms3();
    delay(10);
}
else if((distance_percent >= 50) and (distance_percent <= 64) ) /* more 60 % tank filled */
{
    digitalWrite(tank_pump, HIGH);
    LCD_1();
    digitalWrite(led3, HIGH);
    delay(10);
    digitalWrite(led4, HIGH);
    delay(10);
    digitalWrite(led5, LOW);
    sms3();
    delay(10);
}
else if((distance_percent >= 65) and (distance_percent <= 84) ) /* more than 80 % tank filled */
{
    digitalWrite(tank_pump, HIGH);
```

```
        LCD_1();
        digitalWrite(led3, HIGH);
        delay(10);
        digitalWrite(led4, HIGH);
        delay(10);
        digitalWrite(led5, HIGH);
        sms3();
        delay(10);
    }
    else if((distance_percent >= 85) and (distance_percent <= 100)) /* more than 80 % tank filled */
    {
        digitalWrite(tank_pump, LOW);
        LCD_3();
        digitalWrite(led3, LOW);
        delay(10);
        digitalWrite(led4, HIGH);
        delay(10);
        digitalWrite(led5, HIGH);
        sms1();
        delay(10);
    }
}
void sms(){
SIM900.print("AT+CMGF=1\r");
SIM900.println("AT + CMGS = \"+251920212832\"");// recipient's mobile number
SIM900.println("WATERING SERVO IS ON"); // message to send
SIM900.println("RAIN UNDETECTED ");
Serial.println("RAIN UNDETECTED ");
SIM900.print("Moisture level: ");
Serial.print("Moisture level:   ");
SIM900.println(percentage); //
Serial.println(percentage);
SIM900.println((char)26); // End AT command with a ^Z, ASCII code 26
Serial.println((char)26);
SIM900.println();

}

void sms1(){
SIM900.print("AT+CMGF=1\r");

 SIM900.println("AT + CMGS = \"+251920212832\"");// recipient's mobile number

SIM900.println("TANK PUMP IS OFF"); // message to send
Serial.println(F("TANK PUMP IS OFF"));
SIM900.println("TANK level: ");
Serial.print("TANK level:    ");
```

```
SIM900.println(distance_percent);
Serial.println(distance_percent);

SIM900.println((char)26); // End AT command with a ^Z, ASCII code 26
Serial.println((char)26);
delay(20);
SIM900.println();

}
void sms2(){
SIM900.print("AT+CMGF=1\r");
 SIM900.println("AT + CMGS = \"+251920212832\"");// recipient's mobile number

SIM900.println("WATERING SERVO IS OFF"); // message to send
Serial.println(F("WATERING SERVO IS OFF"));
SIM900.println("RAIN DETECTED ");
Serial.println("RAIN DETECTED ");
SIM900.print("Moisture level: ");
Serial.print("Moisture level:   ");
SIM900.println(percentage);
Serial.println(percentage);
delay(20);
SIM900.println((char)26); // End AT command with a ^Z, ASCII code 26
Serial.println((char)26);
delay(20);
SIM900.println();

}

void sms3(){
SIM900.print("AT+CMGF=1\r");
 delay(200);
 SIM900.println("AT + CMGS = \"+251920212832\"");// recipient's mobile number

SIM900.println("TANK PUMP IS ON"); // message to send
Serial.println(F("TANK PUMP IS ON"));
SIM900.println("TANK level: ");
Serial.print("TANK level:   ");
SIM900.println(distance_percent); //
Serial.println(distance_percent);
delay(20);
SIM900.println((char)26); // End AT command with a ^Z, ASCII code 26
Serial.println((char)26);
delay(20);
SIM900.println();

}
```

```
void readDTH11_Sesnor()
{
   byte RH = dht.readHumidity();
  //Read temperature in degree Celsius
  byte Temp = dht.readTemperature();

  // Check if any reads failed and exit early (to try again)
  if (isnan(RH) || isnan(Temp)) {
    lcd.clear();
    lcd.setCursor(5, 0);
    lcd.print("Error");
    return;

    }
    if (Temp > 30){
    lcd.clear();
    lcd.setCursor(0, 0);
    lcd.print("high temp");
    digitalWrite(led6,HIGH);
    delay(1000);
    digitalWrite(led6,HIGH);
    return;
  }
  else {
    digitalWrite(led6,LOW);
  }

  temperature[7]     = Temp / 10 + 48;
  temperature[8]     = Temp % 10 + 48;
  temperature[11]    = 223;
  humidity[7]        = RH / 10 + 48;
  humidity[8]        = RH % 10 + 48;
  //lcd.clear();
  lcd.setCursor(0, 2);
  lcd.print(temperature);
  lcd.setCursor(0, 3);
  lcd.print(humidity);
}
void condition(){
 rain_Sesnor_value = digitalRead(rain_Sesnor);
 delay(50);
if(rain_Sesnor_value == false){
    if ((Temp<30) && (percentage<85)){
       LCD_2();
       myservo.write(90);
       sms();
```

```
       delay(100);
      }
     else if ((Temp<30) && (percentage>85) )
      {
        LCD_4();
         myservo.write(-90);
         sms2();
        delay(100);
      }
   else if ((Temp>30) && (percentage>85))
     {

        LCD_4();
        myservo.write(-90);
      sms2();
      delay(100);
      }

     else if ((Temp>30) && (percentage<85))
       {
         LCD_4();
         myservo.write(-90);
         sms2();
       delay(100);
}
}
else
 {
      myservo.write(-90);
      lcd.clear();
      lcd.print("Rain Detected");
      lcd.setCursor(0,3);
      lcd.print("SERVO OFF");
      delay(50);
      SIM900.println(F("RAIN DETECTED")); // message to send
      Serial.println(F("RAIN DETECTED"));
      SIM900.println(F("WATERING SERVO IS OFF")); // message to send
      Serial.println(F("WATERING SERVO IS OFF"));
 }
}

  void LCD_1()
  {
  lcd.clear();
  lcd.setCursor(0,0);
  lcd.print("TANK LEVEL=  ");
  lcd.print(distance_percent);
```

```
  lcd.print("%");
  lcd.setCursor(0,1);
  lcd.print("T-PUMP STATUS  ");
  lcd.print("  ON");
  }

void LCD_2(){
  lcd.setCursor(0,0);
  lcd.print("MOIST CONTENT= ");
  lcd.print(percentage);
  lcd.print("%");
  lcd.setCursor(0,1);
  lcd.print("W-SERVO STATUS ");
  lcd.print("  ON");
  }
  void LCD_3(){
  lcd.clear();
  lcd.setCursor(0,0);
   lcd.print("TANK LEVEL=  ");
  lcd.print(distance_percent);
  lcd.print("%");
  lcd.setCursor(0,1);
  lcd.print("T-PUMP STATUS ");
  lcd.print("  OFF");
  }
  void LCD_4(){
  lcd.setCursor(0,0);
  lcd.print("MOIST CONTENT= ");
  lcd.print(percentage);
  lcd.print("%");
  lcd.setCursor(0,1);
  lcd.print("T-PUMP STATUS");
  lcd.print("  OFF");
  }
  void ph_sensor(){
  for(int i=0;i<10;i++)       //Get 10 sample value from the sensor for smooth the value
  {
    buf[i]=analogRead(SensorPin);
    delay(10);
  }
  for(int i=0;i<9;i++)        //sort the analog from small to large
  {
    for(int j=i+1;j<10;j++)
    {
      if(buf[i]>buf[j])
      {
        temp=buf[i];
```

```
        buf[i]=buf[j];
        buf[j]=temp;
      }
    }
  }
  avgValue=0;
  for(int i=2;i<8;i++)                      //take the average value of 6 center sample
    avgValue+=buf[i];
  float phValue=(float)avgValue*5.0/1024/6; //convert the analog into millivolt
  phValue=3.5*phValue; //convert the millivolt into pH value

  Serial.print(F("    pH:"));
  SIM900.println("    pH:");
  Serial.print(phValue,2);
  SIM900.println(phValue,2);
  Serial.println(F(" "));
  delay(100);
}
```