UGC & Govt. Approved
Sonargaon University (SU)
সোনারগাঁও ইউনিভার্সিটি (এসইউ)

Automatic Rain Shed with Temperature Monitoring

Submitted By

Shahadath Hossen	EEE2102023041
Md. Nazmul Hasan	EEE2102023091
Md. Rasel Mia	EEE2102023016
Md. Mostafizur Rahman	EEE2102023070
Md. Bayazid Mia	EEE2102023048

Supervised By

Md Ramjan Ali

Associate Professor

Department of Electrical and Electronic Engineering

Sonargaon University (Su)

Dhaka-1215, Bangladesh

10 January 2025

AUTOMATIC RAIN SHED WITH TEMPERATURE MONITORING

A report submitted to the Department of Electrical and Electronic Engineering, Sonargaon University, Bangladesh, in partial fulfillment of the requirements for the award of the degree of Bachelor of Science in Electrical and Electronic Engineering.

SUBMITTED BY

SHAHADATH HOSSEN	EEE2102023041
MD. NAZMUL HASAN	EEE2102023091
MD. RASEL MIA	EEE2102023016
MD. MOSTAFIZUR RAHMAN	EEE2102023070
MD. BAYAZID MIA	EEE2102023048

SUPERVISED BY

MD RAMJAN ALI

ASSOCIATE PROFESSOR

DEPARTMENT OF ELECTRICAL AND ELECTRONIC ENGINEERING

SONARGAON UNIVERSITY (SU)

DHAKA-1215, BANGLADESH

10 January 2025

LETTER OF TRANSMITTAL

January 2025

To

MD RAMJAN ALI

Associate Professor

Department of Electrical and Electronic Engineering

Sonargaon University

Dhaka-1215, Bangladesh

Subject: Submission of Project Report.

Dear Sir,

We are pleased to submit the project report on "**Automatic Rain Shed with Temperature Monitoring."** It was a great pleasure to work on such an important topic. This project was done under your supervision and according to the requirements of Sonargaon University

We expect the project to be accepted by the concerned authority. We will be happy to provide any further explanation you may feel necessary.

Thank You
Sincerely yours,

SHAHADATH HOSSEN
EEE2102023041

MD. NAZMUL HASAN
EEE2102023091

MD. RASEL MIA
EEE2102023016

MD. MOSTAFIZUR RAHMAN
EEE2102023070

MD. BAYAZID MIA
EEE2102023048

DECLARATION

We hereby solemnly declare that the work presented in this project report has been carried out solely by us and has not been previously submitted to any university or organization for the award of any degree or certificate.

We further affirm that the content of this report does not infringe upon any existing copyright laws.

Additionally, we undertake full responsibility to indemnify the university against any loss or damage arising from a breach of the above declarations



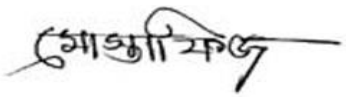
SHAHADATH HOSSEN
EEE2102023041



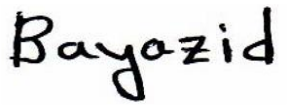
MD. NAZMUL HASAN
EEE2102023091



MD. RASEL MIA
EEE2102023016



MD. MOSTAFIZUR RAHMAN
EEE2102023070



MD. BAYAZID MIA
EEE2102023048

Under Supervision of

MD RAMJAN ALI
Associate Professor
Department of Electrical and Electronic Engineering
Sonargaon University

ACKNOWLEDGEMENT

At first, we started in the name of Almighty Allah. This thesis is accomplished under the supervision of MD RAMJAN ALI, Associate Professor, Department of Electrical and Electronic Engineering, Sonargaon University. It is a great pleasure to acknowledge our profound gratitude and respect to our supervisor for his consistent guidance, encouragement, helpful suggestions, constructive criticism, and endless patience throughout the progress of this work. The successful completion of this thesis would not be possible without his persistent motivation and continuous guidance.

The authors are also grateful to Md. Rais Uddin Mollah, Head of the Department of Electrical and Electronics Engineering, and all the teachers of the Department for their cooperation and significant help in completing the thesis work successfully. Finally, we wish to acknowledge those who directly or indirectly helped us in connecting with this work.

[Authors]

SHAHADATH HOSSEN
EEE2102023041

MD. NAZMUL HASAN
EEE2102023091

MD. RASEL MIA
EEE2102023016

MD. MOSTAFIZUR RAHMAN
EEE2102023070

MD. BAYAZID MIA
EEE2102023048

ABSTRACT

This system aims to make a smart Automatic Rain Shade for Different Ambient Conditions which can reduce human effort due to its functionality and can target the market due to its unique design, is important in modern society at risk of rainfall and wind without coverage in place. The purpose of this project is to make a rain shed that is sensitive to the incidences of rain and sunlight at the domestic and market levels. Specifically, in the summer season the floor of open areas gets so warm due to sunlight that it is difficult to work. This system covers the entire place during the rain and sunlight. This shed operates with the help of rain sensors. The opening and closing of the rain shed are being controlled through a motor automatically and manually. A motor can give a quick and instant response to the shaft so that the shaft can play a major role in power transmission for the opening and closing of the shed. The motor provides the required rpm to the shaft. Whenever the rain comes and after the rain sensor detects the rain then the system will work automatically. Arduino sends the signal further to the motor so that the motor may activate and deliver the required rpm to the coupling shaft. This project is a cost-effective way of providing automated controlled shade from high-intensity sunlight and rain. So, the problems of open and vast areas can be solved by using such types of automated rain shed.

Certification

I certify that I have reviewed this dissertation, and in my opinion, it meets the required scope and quality standards to be accepted as a dissertation for the degree of Bachelor of Science in Engineering.

MD RAMJAN ALI
Associate Professor
Department of Electrical and Electronic Engineering
Sonargaon University

TABLE OF CONTENTS

Letter of transmittal	ii
Declaration	iii
Acknowledgment	iv
Abstract	v
Certification	vi
Table of Contents	vii
List of Figures	ix
List of Tables	x

Chapter-1

INTRODUCTION

1.1 Introduction	01
1.2 Background Study	02
1.3 Problem Statement	03
1.4 Motivation	03
1.5 Objectives	04
1.6 Organization of Book	04

Chapter-2

LITERATURE REVIEW

2.1 Introduction	05
2.2 Literature Review	05
2.3 Summary	06

Chapter-3

METHODOLOGY

3.1 System Design	07
3.2 Methodology	07
3.3 Block Diagram	07
3.4 Circuit Design	08
3.5 Components List	09

3.6 Arduino Nano 09
3.7 SMPS 12
3.8 Motor Driver 17
3.9 Gear Motor 20
3.10 Motor Pulley 21
3.11 Rain Sensor 22
3.12 Bearing 25
3.13 Temperature Sensor 26
3.14 Arduino IDE 27
3.15 Easy EDA 30
3.16 Working Principle 31

Chapter-4
RESULT & DISCUSSION

4.1 Result Analysis 32
4.2 Advantage 33
4.3 Application 33
4.4 Limitation 33
4.5 Discussion 33
4.6 Cost Analysis 34

Chapter-5
CONCLUSION & FUTURE SCOPE

5.1 Conclusion 35
5.2 Future Scope of Work 35

REFERENCE 36
APENDIX 37

LIST OF FIGURES

FIGURE NO	FIGURE NAME	PAGE NO
1.1	Rain Shelter	1
3.1	Block Diagram	8
3.2	Schematic Diagram	8
3.3	Arduino Nano	9
3.4	Arduino Nano Schematic Diagram	10
3.5	How Arduino Nano looks like	10
3.6	Micro-controller IC AT Mega 328P	12
3.7	SMPS	13
3.8	SMPS Circuit	14
3.9	SMPS Circuit Connection	16
3.10	DC Power Supply	17
3.11	Motor Driver	17
3.12	L298N Circuit Diagram	19
3.13	Gear Motor	20
3.14	Pulley	21
3.15	Rain Sensor	22
3.16	Rain Sensor Pin Indication & Description	23
3.17	Bearing	25
3.18	Temperature Sensor	26
3.19	DHT11 Temperature Sensor	26
3.20	Arduino Software Interface IDE	27
3.21	Easy EDA Software Interface	31
4.1	Progress Project Picture	32

LIST OF TABLES

TABLE NO	TABLE NAME	PAGE NO
1	Motor Driver Pin out	18
2	Rain Sensor Pin description	23
3	Rain Sensor Specification	24
4	Component Name and Price	34

Chapter 1
INTRODUCTION

1.1 Introduction

An automatic rain shed is an innovative system designed to provide efficient and hassle-free protection against rain. It operates using sensors to detect rainfall and automatically deploy a protective cover, ensuring immediate shelter without manual intervention. This system is particularly useful in outdoor spaces, such as balconies, terraces, and parking areas, where unexpected rain can cause inconvenience or damage to belongings.



Figure 1.1 Rain Shelter

The rain shed typically consists of a motorized mechanism, rain sensors, and a durable, weather-resistant cover. Upon detecting rain, the system triggers the motor to extend the shed and retracts it once the rain stops, preserving its functionality during dry periods.This smart solution not only enhances convenience but also promotes sustainability by preventing water wastage and protecting valuable assets. Its automation reduces human effort, making it a practical addition to homes, businesses, and public spaces, ensuring uninterrupted usability regardless of weather conditions.

1.2 Background Study

Bangladesh’s most of its economy depends on agriculture. Most of the people in this country are farmers. Agriculture is the largest economic sector and contributes significantly to the country's overall growth. Agriculture's economic contribution to Bangladesh’s GDP has been progressively dropping in recent years, despite the country's broad-based economic development, and one of the issues is unpredictable rains that have ruined crops. Although agriculture is the backbone of Bangladesh's economy, farmers are harmed by unseasonal rain or unwanted rain. Crop production can be harmed if the weather is excessively wet. Rainfall over normal levels can leach nutrients from the soil, particularly nitrogen. During unseasonal or unwanted rain, almost two-thirds of the crops are harmed. According to the Department of Agricultural Extension DAE in 2015, heavy rains harmed crops on 1.17 lakh hectares of land in Bangladesh, putting farmers in danger of loss. As a result, people suffer greatly from this type of unwanted rain, yet no strategies exist to prevent it and as a result, our economy occasionally suffers. There arises a need for effective irrigation for the agricultural production. The control over how much water is to be supplied and when it is to be applied determines the uniformity which is key to maximizing the irrigation efforts.

Proper irrigation management takes careful consideration and vigilant observations. It has many benefits. Keen water irrigation and protection system framework is in this way accepted to be a significant arrangement. The paper along these lines presents an effective water system framework that advances the accessible water in the water supply and in this manner gives an effective and powerful mechanism for irrigation purposes. Irrigation framework would automatically begin/stop water siphons, on the agricultural site depending upon the dampness content obtained by the moisture sensor as soon as it senses the level of water in the reservoir. The deliberate sensor estimates are sent to the Arduino Uno microcontroller for arranging the controlled calculation. As every year farmers suffer a lot due to unwanted rain, this is the time to go for strategies to prevent it. This project mainly focused on a system that prevents the crops from unwanted rain through a shelter that is completely automated and most effective. With a low amount of money, any farmers can install this system on their land and as a result, it will be helpful for the country’s economy.

1.3 Problem Statement

Sudden and unpredictable rainfall poses significant challenges, particularly in outdoor environments such as residential balconies, open terraces, parking areas, and commercial spaces. Without timely protection, rain can damage furniture, equipment, and other exposed belongings, leading to unnecessary expenses and inconvenience. Additionally, manual intervention to deploy traditional rain covers is not always practical, especially when individuals are away or unaware of the weather changes.

The lack of an efficient, automated solution to address this issue results in the frequent underutilization of outdoor spaces during rainy seasons. This impacts the usability of these areas and compromises convenience, comfort, and asset safety.

The problem is further exacerbated by the reliance on traditional systems, which are often labor-intensive and fail to provide real-time protection. This project seeks to address these challenges by developing an automated rain shed system that offers immediate, hands-free shelter, ensuring uninterrupted functionality and safeguarding exposed spaces and belongings.

1.4 Motivation

The development of an automatic rain shed system is driven by the need for a practical and efficient solution to protect outdoor spaces and assets from unpredictable weather conditions. The inconvenience of manually deploying traditional rain covers, coupled with the potential damage caused by sudden rainfall, highlights the importance of a smart, automated approach. This project is motivated by the desire to enhance the usability of outdoor areas, ensuring that they remain functional and protected regardless of the weather. By leveraging technology to automate the process, the system minimizes human intervention, offering a convenient and reliable alternative to conventional methods. Furthermore, the initiative aligns with the growing emphasis on smart living solutions that combine innovation, sustainability, and user convenience. It addresses real-world challenges while promoting a lifestyle that values efficiency and preparedness, making outdoor spaces more adaptable and resilient in the face of changing weather conditions.

1.5 Objective

The objectives of this project are:

- To study the **Automatic Rain Shed with Temperature Monitoring**
- To design and construct the **Automatic Rain Shed with Temperature Monitoring**
- To take necessary notes from the project for future improvements.
- To decrease the cost of the machine by application of simple mechanism.

1.6 Organization of the Book

- **Chapter 1: Introduction-** This chapter is all about the Introduction, background study, Problem Statement, motivation, Objectives and thesis book organization.

- **Chapter 2: Literature Review-** Here briefly describe about previous book review and Summary of this chapter.

- **Chapter 3: Methodology-** This chapter discusses the block diagram, circuit diagram, and components of the list. Here we describe our whole instrument details.

- **Chapter 4: Result and Discussion-** Here briefly discuss about project discussion, result analysis, advantages, application and our system overview.

- **Chapter 5: Conclusion & Future Scope-** This chapter is all about our thesis future Scope and this project conclusion.

Chapter 2
LITERATURE REVIEW

2.1 Introduction

In this chapter, we discuss our system overview. In this paper, we are also working on an Automatic rain Shed. In the below section, we will describe some previous literature reviews. From previous literature, we gathered more knowledge to build this project and we successfully made it.

2.2 Literature review

A. Arduino and Water Sensor: In recent years, water sensors along with Arduino or Raspberry Pi have been used in many applications specifically related to smart irrigation. In [1] water flow sensors along with temperature sensors and soil moisture sensors are used with Arduino. Data collected by these sensors are sent to the interactive website by a WIFI module attached to an Arduino. This allows users to control water pumps and sprinklers from a distance which will facilitate farmers to obtain maximum quality crops. In [2] smart drip irrigation system is proposed using Arduino, Raspberry Pi, and Zigbee modules. Arduino micro-controllers are used to receive the on/off commands from the Raspberry Pi using the Zigbee protocol. The water pump can automatically start by just sending an email to the system. In [3] Arduino and water sensors are used for measuring and monitoring water quality for fish farming. It is used to measure temperature and pH value of water. Collected data is sent via Bluetooth to a mobile application for monitoring purposes.

B. Arduino and Temperature Sensor: Many researchers have used Arduino and temperature sensors along with GSM, GPRS, Bluetooth and Zigbee modules for automation of various systems. Some of the systems are as follows: “Home Weather Station” is a mobile weather station that displays weather details on an LCD screen [4]. The outdoor station measures temperature, pressure and humidity and sends it to an indoor module for display. In [5] designed a low-cost environment monitoring system using Raspberry Pi and Arduino with Zigbee. In [6] continuous heart rate and body temperature monitoring system is proposed using Arduino UNO and Android devices.

C. Arduino and Light Dependent Resistor: Light Dependent Resistors (LDR) are used in many applications where light has to be used or sensed for automation purposes. In [7] LDR is a sensor and Light Emitting Diode (LED) is a light source to design a water turbidity gauge. Turbidity of water is measured by passing a light source through water to measure the intensity of light reflected. Analog data received by Arduino from LDR is converted into digital form and the result is displayed on LCD. In [8] an intelligent street light system is proposed. Street light not only depends on darkness but also motion. Street light will only turn on if it's dark and motion is detected otherwise it will stay off, thus energy is saved.

In [9] proposed origami-based automatic solar panels. Normally, solar panels are stationary and the energy absorbed is not optimum. To make sure maximum solar energy is absorbed by solar panels rotational solar panels are proposed. LDR will obtain and measure sunlight and solar panels will automatically rotate toward maximum sunlight. Above are some of the systems that have used Arduino as a microprocessor and LDR, LM 35and water sensors individually for data collection. None of the systems have combined all these sensors and designed a system to automatically open or close a shed or a rain shed. This paper proposed a system that will automate shed depending upon temperature, light and water to facilitate user needs accordingly.

2.3 Summary

From the literature discussed above, we gained a lot of knowledge and we were inspired to do this project. We were able to do it with everyone's tireless work.

Chapter 3

METHODOLOGY

3.1 System Design

In this project, we have used an SMPS section to produce power in the system. Here we use a microcontroller. All kinds of control systems work under Arduino Nano. Here we also use a Rain Sensor, Temperature Sensor, Forward and Reverse Switch Motor Driver, and Gear Motor. All of this equipment is connected like circuit design and it fulfills our desired outcome.

3.2 Methodology

Our methodology for the project:

- Creating an idea for the design and construction of an **Automatic Rain Shed with Temperature Monitoring** with the combination of Arduino microcontroller.
- And designing a block diagram & circuit diagram to know which components we need to construct it. Collecting all the components and programming the microcontroller to control the whole system.
- Setting up all the components in a PCB board & then soldering. Lastly, assembling all the blocks in a board and running the system & for checking purposes.

3.3 Block Diagram

Here is the block diagram of the **Automatic Rain Shed with Temperature Monitoring** with all the essential components. All of the components are shown below as a block in this diagram. Here we use Arduino Nano for the main microcontroller which automatically operates the system. Other types of equipment are attached to this microcontroller and work together to perform our desired outcome.

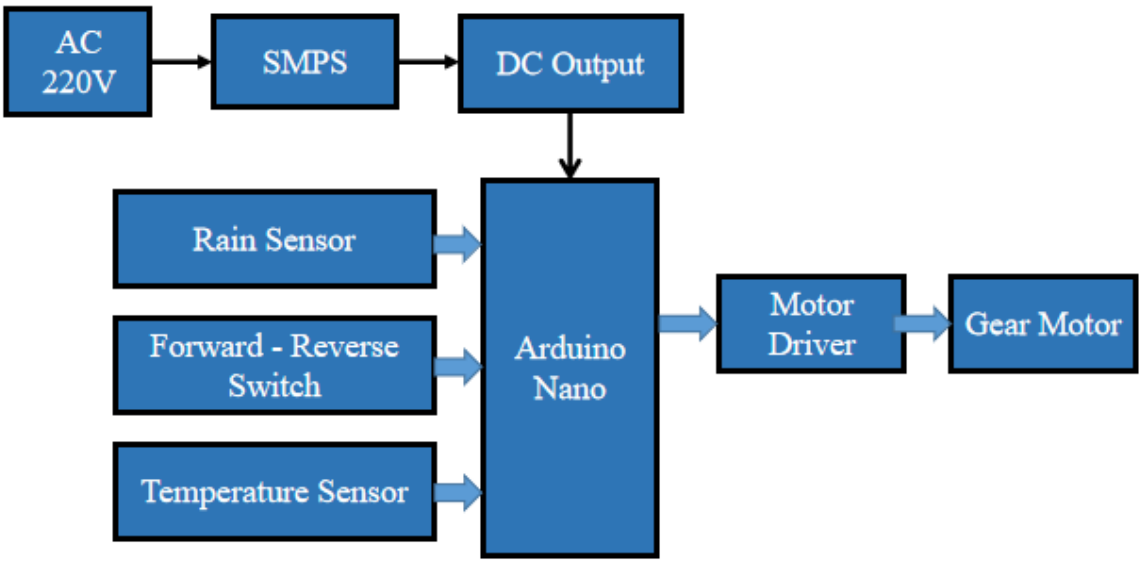


Figure 3.1: Block Diagram

3.4 Circuit Design

Arduino Pro Mini has been used as the main controller for this project. With the help of a controller, the project can be controlled by our system automatically.

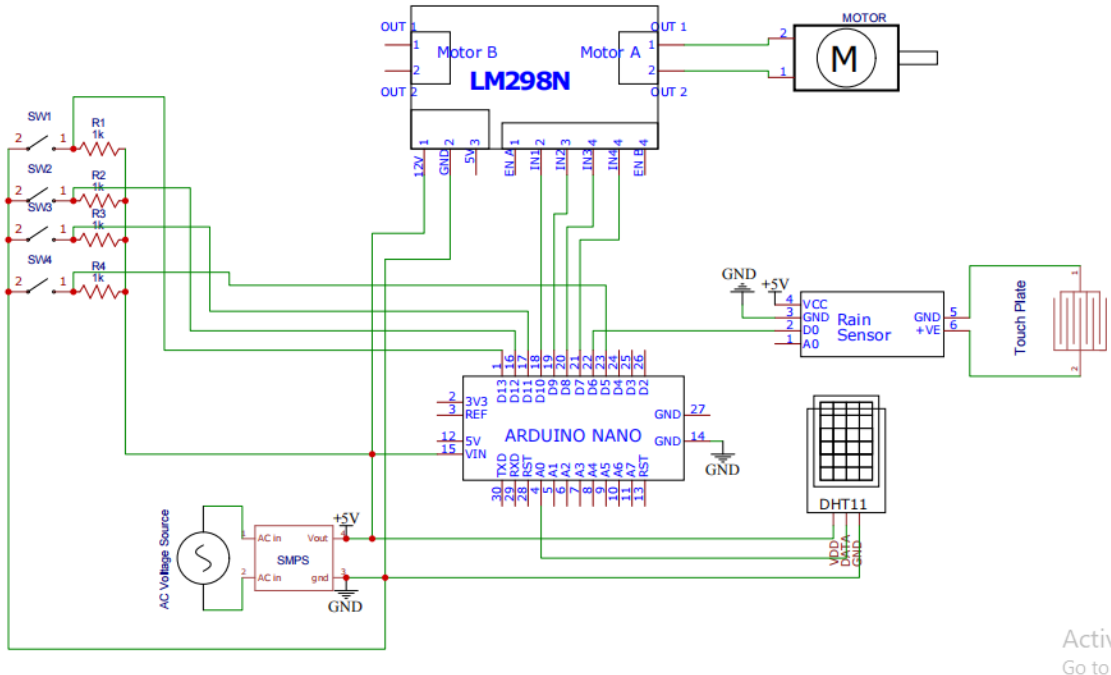


Figure 3.2: Schematic Diagram

3.5 Components List:

- Arduino Nano
- SMPS
- Gear Motor
- Motor Driver
- Bearing
- Motor Pulley
- Rain Sensor
- Temperature Sensor

Software

- Arduino IDE
- Easy EDA

3.6 Arduino Nano

Arduino is an open-source electronics prototyping platform based on flexible, easy-to-use hardware and software. It's intended for artists, designers, hobbyists, and anyone interested in creating interactive objects or environments. Arduino can sense the environment by receiving input from a variety of sensors and can affect its surroundings by controlling Lights, motors, and other actuators.



Figure 3.3: Arduino Nano

The microcontroller on the board is programmed using the Arduino programming language (based on Wiring) and the Arduino development environment (based on Processing). Arduino projects can be stand-alone or they can communicate with software running on a computer (e.g. Flash, Processing, Maxims'). Arduino Nano is a surface mount breadboard embedded version with an integrated USB. It is a small, complete, and breadboard-friendly component. It has everything that Decimal/ Duemilanove has (electrically) with more analog input pins and an onboard +5V AREF jumper. Physically, it is missing a power jack. The Nano can automatically sense and switch to the higher potential source of power.

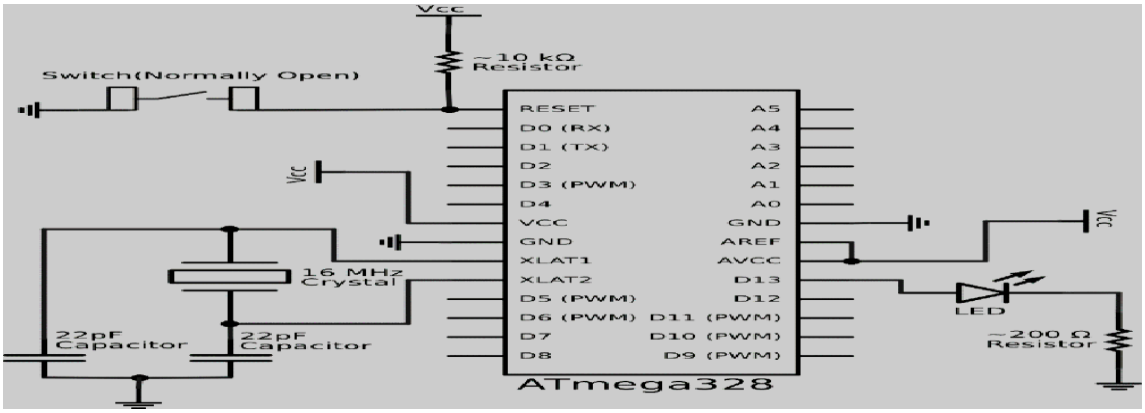


Figure 3.4: Arduino Nano Schematic Diagram

Nano's got the breadboard ability of the Boarding and the Minibus with smaller footprint than either, so users have more breadboard space. It's got a pin layout that works well with the Mini or the Basic Stamp (TX, RX, ATN, and GND on one top, power and ground on the other). This new version 3.0 comes with ATMEGA328 which offers more programming and data memory space. It has two layers. That makes it easier to hack and more affordable. One of the best features of Arduino Nano is, that it's easy to use, compact and also small.

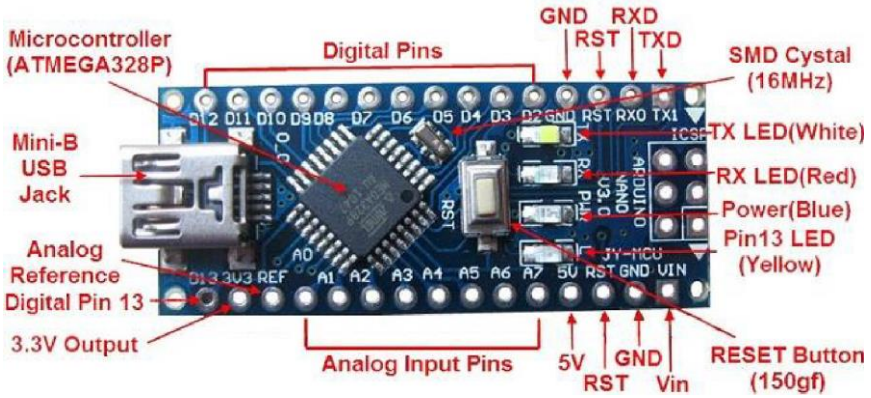


Figure 3.5: How Arduino Nano looks like

Specifications:

- Micro-controller: Atmel ATmega328
- Operating Voltage (logic level): 5 V
- Input Voltage (recommended): 7-12 V
- Input Voltage (limits): 6-20 V
- Digital I/O Pins: 14 (of which 6 provide PWM output)
- Analog Input Pins: 8
- DC Current per I/O Pin: 40 mA
- Flash Memory: 32 KB (of which 2KB is used by the boot loader)
- SRAM: 2 KB
- EEPROM: 1 KB
- Clock Speed: 16 MHz
- Dimensions: 0.70” x 1.70”

Features:

- Automatic reset during program download
- Power OK blue LED
- Green (TX), red (RX) and orange (L) LED
- Auto sensing/switching power input
- Small mini-B USB for programming and serial monitor
- ICSP header for direct program download
- Manual reset switch

Micro-controller IC ATmega328p

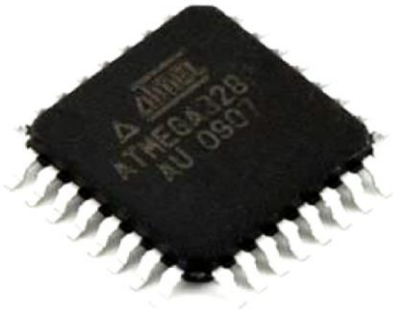


Figure 3.6: Micro-controller IC AT mega 328p

The high-performance Microchip Pico Power 8-bit AVR RISC-based micro-controller combines 32KB ISP flash memory with read-while-write capabilities, 1024B EEPROM, 2KB SRAM, 23 general purpose I/O lines, 32 general purpose working registers, three flexible timer/counters with compare modes, internal and external interrupts, serial programmable USART, a byte-oriented 2-wire serial interface, SPI serial port, a 6-channel 10-bit A/D converter (8-channels in TQFP and QFN/MLF packages), programmable watchdog timer with internal oscillator, and five software selectable power saving modes. The device operates between 1.8-5.5 volts. By executing powerful instructions in a single clock cycle, the device achieves throughput approaching 1 MIPS per MHz, balancing power consumption and processing speed.

3.7 SMPS

A switched-mode power supply (switching-mode power supply, switch-mode power supply, switched power supply, SMPS, or switcher) is an electronic power supply that incorporates a switching regulator to convert electrical power efficiently. Unlike a linear power supply, the pass transistor of a switching-mode supply continually switches between low-dissipation, full-on and full-off states, and spends very little time in the high-dissipation transitions, which minimizes wasted energy. A hypothetical ideal switched-mode power supply dissipates no power. Voltage regulation is achieved by varying the ratio of on-to-off time (also known as duty cycles)

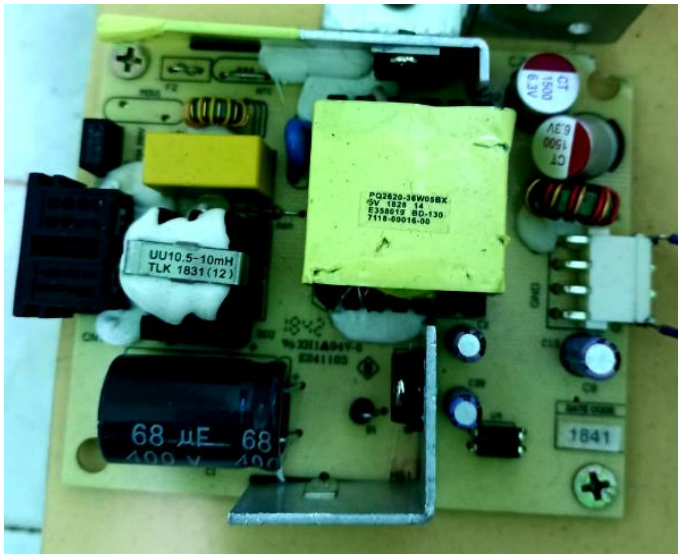


Figure 3.7: SMPS

Switching regulators are used as replacements for linear regulators when higher efficiency, smaller size, or lighter weight are required. They are, however, more complicated; their switching currents can cause electrical noise problems if not carefully suppressed, and simple designs may have a poor power factor. Switched-mode power supplies are classified according to the type of input and output voltages. The four major categories are:

- AC to DC
- DC to DC
- DC to AC
- AC to AC

A basic isolated AC to DC switched-mode power supply consists of:

- Input rectifier and filter
- Inverter consisting of switching devices such as MOSFETs
- Transformer
- Output rectifier and filter
- Feedback and control circuit

The input DC supply from a rectifier or battery is fed to the inverter where it is turned on and off at high frequencies of between 20 KHz and 200 KHz by the switching MOSFET or power transistors. The high-frequency voltage pulses from the inverter are fed to the transformer's primary winding, and the secondary AC output is rectified and smoothed to produce the required DC voltages. A feedback circuit monitors the output voltage and instructs the control circuit to adjust the duty cycle to maintain the output at the desired level.

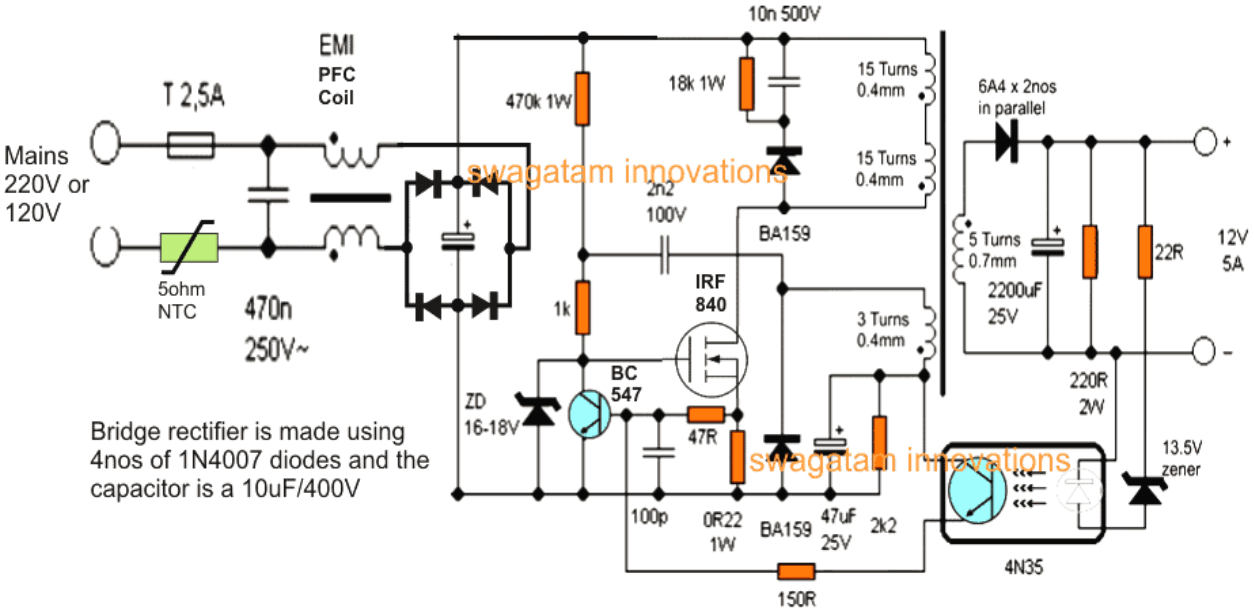


Figure 3.8: SMPS Circuit

The basic working concept of an SMPS

A switching regulator does the regulation in the SMPS. A series switching element turns the current supply to a smoothing capacitor on and off. The voltage on the capacitor controls the time the series element is turned. The continuous switching of the capacitor maintains the voltage at the required level.

Design basics

AC power first passes through fuses and a line filter. Then it is rectified by a full-wave bridge rectifier. The rectified voltage is next applied to the power factor correction (PFC) pre-regulator followed by the downstream DC-DC converter(s). Most computers and small appliances use the International Electro-Technical Commission (IEC) style input connector. As for output connectors and pin outs, except for some industries, such as PC and compact PCI, in general, they are not standardized and are left up to the manufacturer. There are different circuit configurations known as topologies, each having unique characteristics, advantages and modes of operation, which determines how the input power is transferred to the output. Most of the commonly used topologies such as fly back, push-pull, half-bridge and full-bridge, consist of a transformer to provide isolation, voltage scaling, and multiple output voltages. The non-isolated configurations do not have a transformer and the power conversion is provided by the inductive energy transfer.

Advantages of switched-mode power supplies:

- Higher efficiency of 68% to 90%
- Regulated and reliable outputs regardless of variations in input supply voltage
- Small size and lighter
- Flexible technology
- High power density

Disadvantages:

- Generates electromagnetic interference
- Complex circuit design
- Expensive compared to linear supplies

Switched-mode power supplies are used to power a wide variety of equipment such as computers, sensitive electronics, battery-operated devices and other equipment requiring high efficiency.

Switch Mode Power Supply

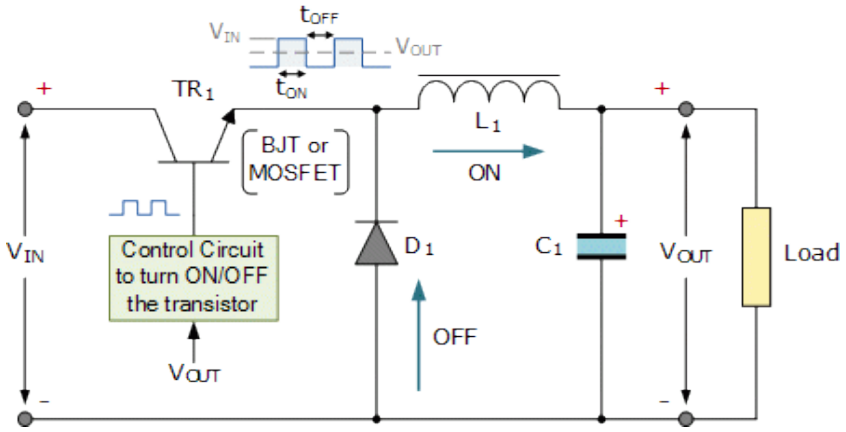


Figure 3.9: SMPS Circuit Connection

Linear voltage IC regulators have been the basis of power supply designs for many years as they are very good at supplying a continuous fixed voltage output. Linear voltage regulators are generally much more efficient and easier to use than equivalent voltage regulator circuits made from discrete components such as a Zener diode and a resistor, or transistors and even op-amps. The most popular linear and fixed output voltage regulator types are by far the positive output voltage series and the negative output voltage series. These two types of complementary voltage regulators produce a precise and stable voltage output ranging from about 5 volts up to about 24 volts for use in many electronic circuits. There is a wide range of these three-terminal fixed voltage regulators available each with its built-in voltage regulation and current limiting circuits. This allows us to create a whole host of different power supply rails and outputs, either single or dual supply, suitable for most electronic circuits and applications. There are even variable voltage linear regulators available as well providing an output voltage that is continually variable from just above zero to a few volts below its maximum voltage output.

Most D.C power supplies comprise a large and heavy step-down mains transformer, diode rectification, either full-wave or half-wave and a filter circuit to remove any ripple content from the rectified D.C producing a suitably smooth D.C voltage, and some form of voltage regulator or stabilizer circuit, either linear or switching to ensure the correct regulation of the power supply output voltage under varying load conditions. Then a typical D.C. power supply would look something like this:

Typical DC Power Supply

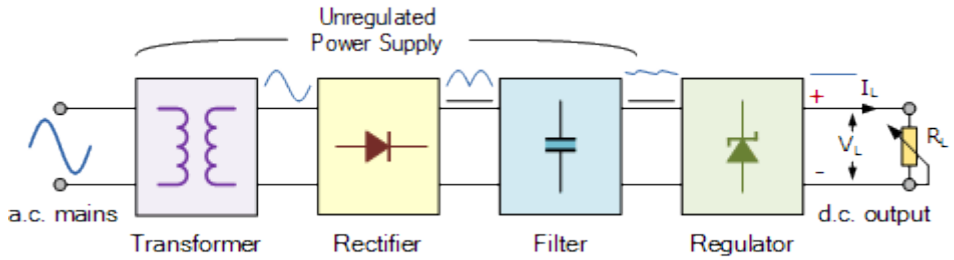


Figure 3.10: DC Power Supply

These typical power supply designs contain a large mains transformer (which also provides isolation between the input and output) and a dissipative series regulator circuit. The regulator circuit could consist of a single zener diode or a three-terminal linear series regulator to produce the required output voltage. The advantage of a linear regulator is that the power supply circuit only needs an input capacitor, output capacitor and some feedback resistors to set the output voltage.

3.8 Motor Driver

The **L298N** is a popular 16-pin **Motor Driver** IC. As the name suggests it is mainly used to drive **motors**. A single **L298N** IC is capable of running two DC **motors** at the same time; also the direction of these two **motors** can be controlled independently.

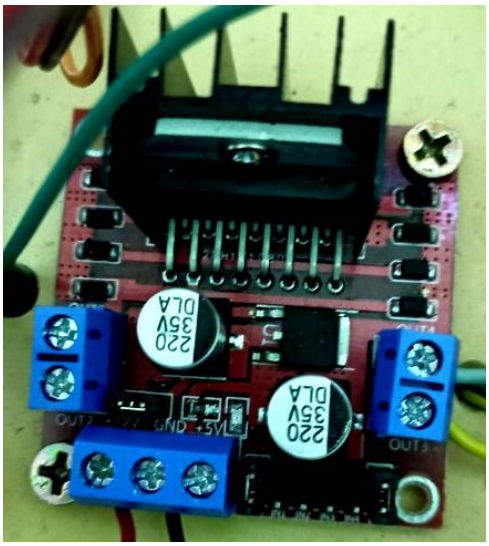


Figure 3.11: Motor driver

Working Process

L298N IC is a typical **Motor Driver** IC that allows the DC **motor** to drive in any direction. This IC consists of 16-pins which are used to control a set of two DC **motors** instantaneously in any direction. It means, that by using an **L298N** IC we can control two DC **motors**.

Features

- Can be used to run Two DC motors with the same IC.
- Speed and Direction control is possible
- Motor voltage Vcc2 (vss): 4.5V to 36V
- Maximum Peak motor current: 1.2A
- Maximum Continuous Motor Current: 600mA
- Supply Voltage to Vcc1(vss): 4.5V to 7V
- Transition time: 300ns (at 5Vand 24V)
- Automatic Thermal shutdown is available

L298N Pin Configuration

Table 1: Motor Driver Pin out

Pin Number	Pin Name	Description
1	Enable 1,2	This pin enables the input pin Input 1(2) and Input 2(7)
2	Input 1	Directly controls the Output 1 pin. Controlled by digital circuits
3	Output 1	Connected to one end of Motor 1
4	Ground	Ground pins are connected to ground of circuit (0V)
5	Ground	Ground pins are connected to ground of circuit (0V)
6	Output 2	Connected to another end of Motor 1
7	Input 2	Directly controls the Output 2 pin. Controlled by digital circuits

8	Vcc2 (Vss)	Connected to Voltage pin for running motors (4.5V to 36V)
9	Enable 3,4	This pin enables the input pin Input 3(10) and Input 4(15)
10	Input 3	Directly controls the Output 3 pin. Controlled by digital circuits
11	Output 3	Connected to one end of Motor 2
12	Ground	Ground pins are connected to ground of circuit (0V)
13	Ground	Ground pins are connected to ground of circuit (0V)
14	Output 4	Connected to another end of Motor 2
15	Input 4	Directly controls the Output 4 pin. Controlled by digital circuits
16	Vcc2 (Vss)	Connected to +5V to enable IC function

Use of a L298N Motor Driver IC:

Using this L298N motor driver IC is very simple. The IC works on the principle of **Half H-Bridge**, let us not go too deep into what H-Bridge means, but for now, just know that H-Bridge is a setup that is used to run motors both in a clockwise and anti-clockwise direction. As said earlier this IC is capable of running two motors in any direction at the same time, the circuit to achieve the same is shown below.

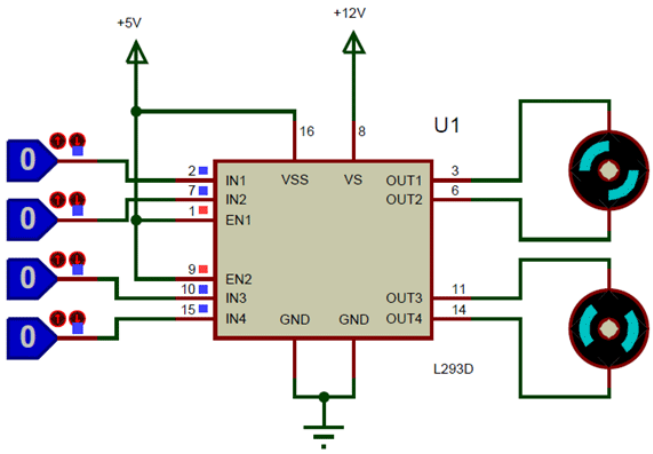


Figure 3.12: L298N Circuit Diagram

All the Ground pins should be grounded. There are two power pins for this IC, one is the Vss(Vcc1) which provides the voltage for the IC to work, and this must be connected to +5V. The other is Vss(Vcc2) which provides voltage for the motors to run, based on the specification of your motor you can connect this pin to anywhere between 4.5V to 36V, here I have connected to +12V.

The Enable pins (Enable 1, 2 and Enable 3, 4) are used to Enable Input pins for Motor 1 and Motor 2 respectively. Since in most cases we will be using both the motors both the pins are held high by default by connecting to +5V supply. Input pins 1, 2 are used to control Motor 1 and Input pins 3, 4 are used to control Motor 2. The input pins are connected to any Digital circuit or microcontroller to control the speed and direction of the motor.

Applications

- Used to drive high-current Motors using Digital Circuits
- Can be used to drive Stepper motors
- High-current LEDs can be driven
- Relay Driver module (Latching Relay is possible)

3.9 Gear Motor

A DC motor is any motor within a class of electrical machines whereby direct current electrical power is converted into mechanical power. A 12v DC motor is small and inexpensive, yet powerful enough to be used for many applications.



Figure 3.13: Gear Motor

Specification:

- Voltage: 12V DC
- Gear ratio: 1/31
- No-load speed: 200RPM
- Rated Speed: 140RPM
- Rated torque: 10kg.cm
- Rated current: 2.5Amp
- Length of Motor(including spindle): 106mm/4.17″
- Diameter: 37mm/1.45″
- Shaft length: 21mm/0.82″
- Shaft diameter: 6mm/0.24″

3.10 Motor Pulley

A pulley is a wheel on an axle or shaft that is designed to support movement and change of direction of a taut cable or belt, or transfer of power between the shaft and cable or belt. In the case of a pulley supported by a frame or shell that does not transfer power to a shaft, but is used to guide the cable or exert a force, the supporting shell is called a block, and the pulley may be called a sheave. A pulley may have a groove or grooves between flanges around its circumference to locate the cable or belt. The drive element of a pulley system can be a rope, cable, belt, or chain.

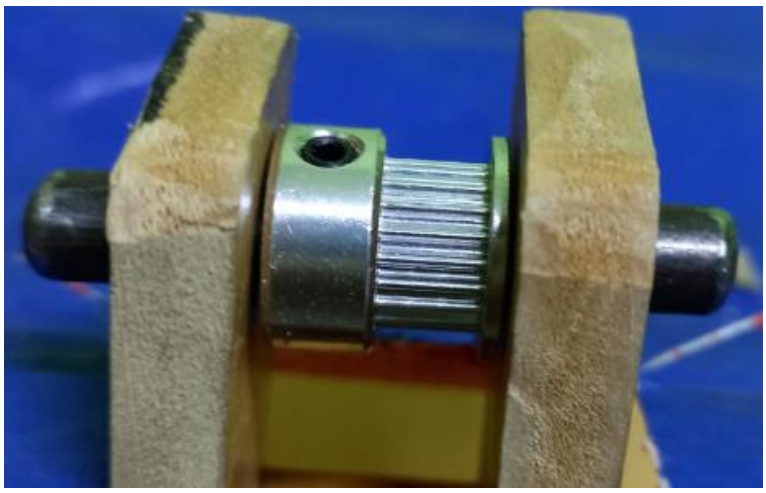


Figure 3.14: Pulley

The earliest evidence of pulleys dates back to Ancient Egypt in the Twelfth Dynasty (1991-1802 BCE) and Mesopotamia in the early 2nd millennium BCE. In Roman Egypt, Hero of Alexandria (c. 10-70 CE) identified the pulley as one of six simple machines used to lift weights. Pulleys are assembled to form a block and tackle to provide mechanical advantage to apply large forces. Pulleys are also assembled as part of belt and chain drives to transmit power from one rotating shaft to another. Plutarch's *Parallel Lives* recounts a scene where Archimedes proved the effectiveness of compound pulleys and the block-and-tackle system by using one to pull a fully laden ship towards him as if it was gliding through water.

3.11 Rain sensor

A rain sensor is one kind of low-cost electronic sensor that is used to detect rainfall or water drops. It works as a switch. Normally the switch is open. This sensor consists of mainly two parts, one is the Sensing Pad and another one is the Sensor Module. When rainfall or water drops fall on the Sensing Pad surface, then the switch will be closed.



Figure 3.15: Rain Sensor

The Sensor Module reads data from the sensor pad processes the data and converts it into a digital/analog output. So, the sensor can provide both types of output Digital output (DO) and Analog output (AO). The Rain Sensor Module's Sensing Pad consists of two nickel-coated series copper tracks. Also, it has **two Header pins**, which are internally connected to the two copper tracks of the Sensing Pad. These pins are used to connect the Sensing Pad to the rain sensor module circuit through two jumper wires. Always, one pin of the rain sensor circuit provides a +5v power supply to one track of the sensing pad, and another pin receives the return power supply from another track of the sensing pad.

Normally under dry conditions, the sensing pad provides high resistance and low conductivity. So, the 5v power supply cannot be passed from one track to another track. Its resistance varies according to the amount of water on the surface of the sensing pad. When water drops fall on the sensor pad surface its resistance will decrease and conductivity will increase. So, when water drops increase on the pad surface it can pass more power supply through one track to another track.



Figure 3.16: Rain Sensor Pin Indication & Description

Pin Description

Table 2: Rain sensor Pin Description

	Pin Name	Description
1	**VCC**	+5 v power supply
2	**GND**	Ground (-) power supply
3	**DO**	Digital Output (0 or 1)
4	**AO**	Analog Output (range 0 to 1023)

Sensor Module Hardware Overview

The Sensor module consists of some key components. These are LM393 Comparators, Variable Resistor (Trim pot), Power LED, and Output LED.

Rain Sensor Specifications

Table 3: Rain Sensor Specification

Parameter	Value
Operating Voltage	3.3V – 5V
Operating Current	15 mA
Comparator chip	LM393
Sensitivity	Adjustable via Trim pot
Output type	Analog output voltage (AO) and Digital switching voltage (DO)
LED lights indicators	Power (red/green) and Output (red/green)
Sensing pad	5cm x 4 cm nickel plate on one side.
Module PCB Size	3.2cm x 1.4cm

Application

- Rainfall detection
- It's used in an irrigation system, when rainfall is started it shuts down the watering system automatically.

3.12 Bearing

A **bearing** is a machine element that constrains relative motion to only the desired motion and reduces friction between moving parts. The design of the bearing may, for example, provide for free linear movement of the moving part or free rotation around a fixed axis; or, it may prevent a motion by controlling the vectors of normal forces that bear on the moving parts. Most bearings facilitate the desired motion by minimizing friction. Bearings are classified broadly according to the type of operation, the motions allowed, or to the directions of the loads (forces) applied to the parts. Rotary bearings hold rotating components such as shafts or axles within mechanical systems and transfer axial and radial loads from the source of the load to the structure supporting it. The simplest form of bearing, the plain bearing, consists of a shaft rotating in a hole. Lubrication is used to reduce friction. In the ball bearing and roller bearing, to reduce sliding friction, rolling elements such as rollers or balls with a circular cross-section are located between the races or journals of the bearing assembly. A wide variety of bearing designs exists to allow the demands of the application to be correctly met for maximum efficiency, reliability, durability and performance.



Figure 3.17: Bearing

3.13 Temperature Sensor

The DHT11 is a basic, ultra-low-cost digital temperature and humidity sensor. It uses a capacitive humidity sensor and a thermistor to measure the surrounding air and spits out a digital signal on the data pin (no analog input pins needed). It's fairly simple to use but requires careful timing to grab data.

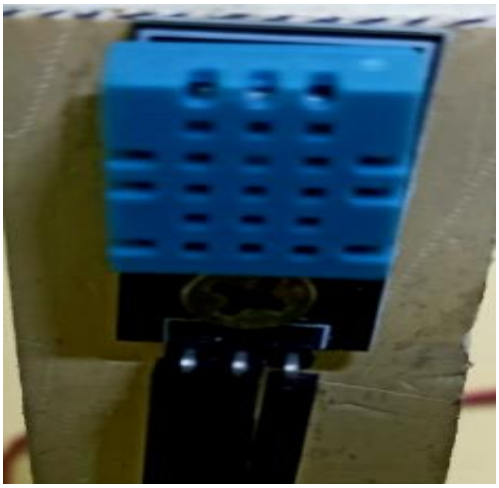


Figure 3.18: Temperature Sensor (DHT11)

DHT11 Specifications:

- Operating Voltage: 3.5V to 5.5V
- Operating current: 0.3mA (measuring) 60uA (standby)
- Output: Serial data
- Temperature Range: 0°C to 50°C
- Humidity Range: 20% to 90%
- Resolution: Temperature and Humidity both are 16-bit
- Accuracy: ±1°C and ±1%

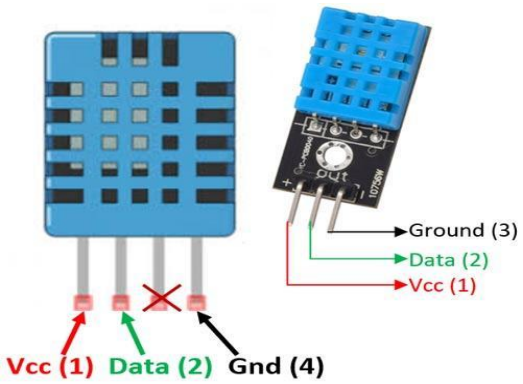


Figure 3.19: DHT11 Temperature Sensor

3.14 Arduino IDE

The digital microcontroller unit named Arduino Nano can be programmed with the Arduino software IDE. There is no requirement for installing other software other than Arduino. Firstly, Select "Arduino Nano from the Tools, Board menu (according to the microcontroller on our board). The IC used named a Tmega328 on the Arduino Nano comes pre-burnt with a boot loader that allows us to upload new code to it without the use of an external hardware programmer.

Communication is using the original STK500 protocol (reference, C header files). We can also bypass the boot loader and program the microcontroller through the ICSP (In-Circuit Serial Programming) header. The ATmega16U2 (or 8U2 in the rev1 and rev2 boards) firmware source code is available. The ATmega16U2/8U2 is loaded with a DFU boot loader, which can be activated by:

On Rev1 boards: connecting the solder jumper on the back of the board (near the map of Italy) and then resetting the 8U2. On Rev2 or later boards: there is a resistor that pulls the 8U2/16U2 HWB line to the ground, making it easier to put into DFU mode.

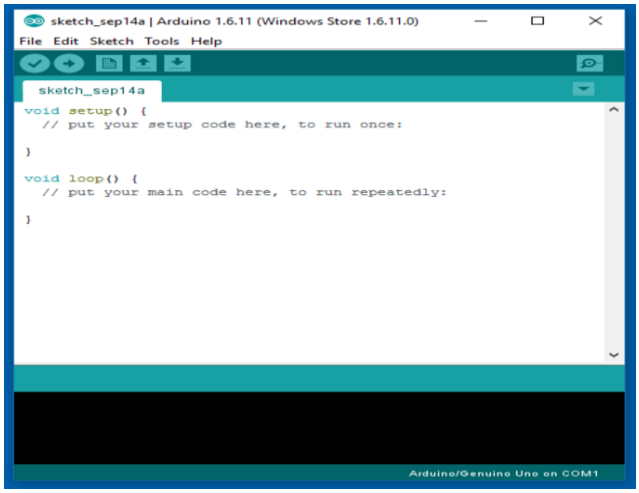


Figure 3.20: Arduino Software Interface IDE

The Arduino Nano is one of the latest digital microcontroller units and has several facilities for communicating with a computer, another Arduino, or other microcontrollers. The ATmega328 provides UART TTL at (5V) with serial communication, which is available on digital pins 0 - (RX) for receiving the data and pins no.1 (TX) for transmitting the data. An ATmega16U2 on the board channels this serial communication over USB and appears as a virtual COM port to software on the computer.

The RX and TX LEDs on the board will flash when data is being transmitted via the USB-to-serial chip and USB connection to the computer (but not for serial communication on pins 0 and 1). A Software Serial library allows for serial communication on any of the Nano's digital pins. The ATmega328 also supports I2C (TWI) and SPI communication. The Arduino software includes a Wire library to simplify the use of the I2C bus. Arduino programs are written in C or C++ and the program code written for Arduino is called Sketch. The Arduino IDE uses the GNU tool chain and AVR Lab to compile programs, and for uploading the programs it uses. As the Arduino platform uses Atmel micro-controllers, Atmel's development environment, AVR Studio, or the newer Atmel Studio, may also be used to develop software for the Arduino. The Arduino Integrated Development Environment - or Arduino Software (IDE) - contains a text editor for writing code, a message area, a text console, a toolbar with buttons for common functions and a series of menus. It connects to the Arduino and Genuino hardware to upload programs and communicate with them.

Writing Sketches

Programs written using Arduino Software (IDE) are called sketches. These sketches are written in the text editor and are saved with the file extension .ino. The editor has features for cutting/pasting and for searching/replacing text. The message area gives feedback while saving and exporting and also displays errors. The console displays text output by the Arduino Software (IDE), including complete error messages and other information. The bottom right-hand corner of the window displays the configured board and serial port. The toolbar buttons allow you to verify and upload programs, create, open, and save sketches, and open the serial monitor.

Sketchbook

The Arduino Software (IDE) uses the concept of a sketchbook: a standard place to store your programs (or sketches). The sketches in your sketchbook can be opened from the File > Sketchbook menu or the Open button on the toolbar. The first time you run the Arduino software, it will automatically create a directory for your sketchbook. You can view or change the location of the sketchbook location from with the Preferences dialog. Beginning with version 1.0, files are saved with an .ino file extension.

Tabs, Multiple Files, and Compilation

Allows you to manage sketches with more than one file (each of which appears in its tab). These can be normal Arduino code files (no visible extension), C files (.c extension), C++ files (.cpp), or header files (.h).

Uploading

Before uploading your sketch, you need to select the correct items from the Tools > Board and Tools > Port menus. The boards are described below. On the Mac, the serial port is probably something like /dev/tty.usbmodem241 (for a Uno or Mega2560 or Leonardo) or /dev/tty.USB serial-1B1 (for a Duemilanove or earlier USB board), or /dev/tty.USA19QW1b1P1.1 (for a serial board connected with a Key span USB-to-Serial adapter). On Windows, it's probably COM1 or COM2 (for a serial board) or COM4, COM5, COM7, or higher (for a USB board) - to find out, you look for a USB serial device in the port section of the Windows Device Manager. On Linux, it should be /dev/ttyACMx, /dev/ttyUSBx or similar. Once you've selected the correct serial port and board, press the upload button in the toolbar or select the Upload item from the Sketch menu. Current Arduino boards will reset automatically and begin the upload. With older boards (pre-Decimal) that lack auto-reset, you'll need to press the reset button on the board just before starting the upload. On most boards, you'll see the RX and TX LEDs blink as the sketch is uploaded. The Arduino Software (IDE) will display a message when the upload is complete, or show an error.

When you upload a sketch, you're using the Arduino Boot loader, a small program that has been loaded onto the microcontroller on your board. It allows you to upload code without using any additional hardware. The Boot loader is active for a few seconds when the board resets; then it starts whichever sketch was most recently uploaded to the microcontroller. The Boot loader will blink the on-board (pin 13) LED when it starts (i.e. when the board resets).

Libraries

Libraries provide extra functionality for use in sketches, e.g. working with hardware or manipulating data. To use a library in a sketch, select it from the Sketch > Import Library menu. This will insert one or more #include statements at the top of the sketch and compile the library with your sketch. Because libraries are uploaded to the board with your sketch,

they increase the amount of space it take up. If a sketch no longer needs a library, simply delete its #include statements from the top of your code. There is a list of libraries in the reference. Some libraries are included with the Arduino software. Others can be downloaded from a variety of sources or through the Library Manager. Starting with version 1.0.5 of the IDE, you can import a library from a zip file and use it in an open sketch. See these instructions for installing a third-party library.

Third-Party Hardware

Support for third-party hardware can be added to the hardware directory of your sketchbook directory. Platforms installed there may include board definitions (which appear in the board menu), core libraries, Boot loaders, and programmer definitions. To install, create the hardware directory, then unzip the third-party platform into its sub-directory. (Don't use "Arduino" as the sub-directory name or you'll override the built-in Arduino platform.) To uninstall, simply delete its directory. For details on creating packages for third-party hardware, see the Arduino IDE 1.5 3rd-party Hardware specification.

Serial Monitor

This displays serial sent from the Arduino or Genuino board over a USB or serial connector. To send data to the board, enter text, click the "send" button, or press enter. Choose the baud rate from the drop-down menu that matches the rate passed to Serial begin in your sketch. Note that the board will reset on Windows, Mac, or Linux (it will rerun your sketch) when you connect with the serial monitor.

3.15 Easy EDA

Easy EDA is a web-based EDA tool suite that enables hardware engineers to design, simulate, share-publicly and privately- and discuss schematics, simulations and printed circuit boards. Other features include the creation of a bill of materials, Gerber files pick and place files and documentary outputs in PDF, PNG and SVG formats. Easy EDA allows the creation and editing of schematic diagrams, SPICE simulation of mixed analog and digital circuits and the creation and editing of printed circuit board layouts and, optionally, the manufacture of printed circuit boards.

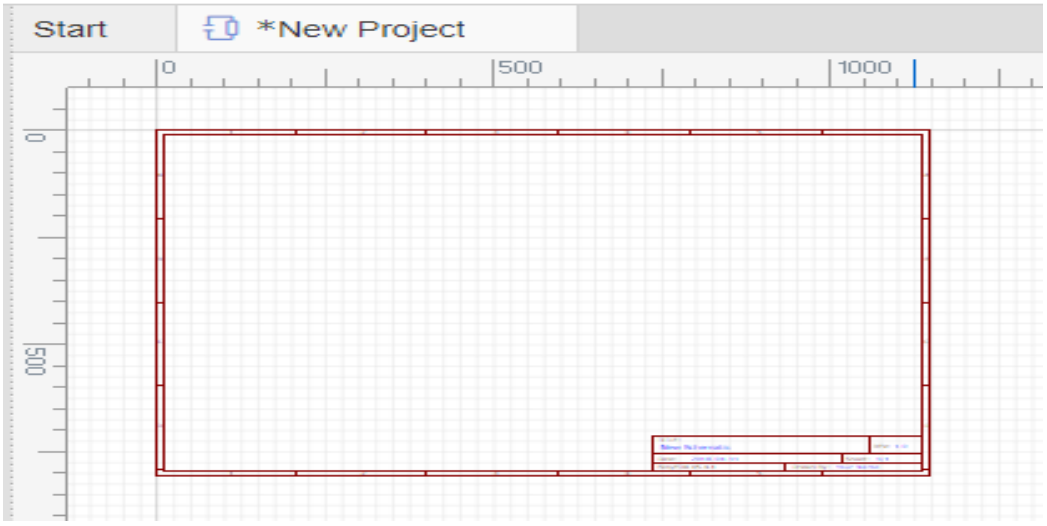


Figure 3.21: Easy EDA Software Interface

Subscription-free membership is offered for the public plus a limited number of private projects. The number of private projects can be increased by contributing high-quality public projects, schematic symbols, and PCB footprints and/or by paying a monthly subscription. Registered users can download Gerber files from the tool free of charge; but for a fee, Easy EDA offers a PCB fabrication service. This service is also able to accept Gerber file inputs from third-party tools.

3.16 Working Principle

The automatic rain shed system is designed to protect outdoor spaces from rain by automatically deploying or retracting a shed. At its core, the Arduino Nano processes inputs from a rain sensor and a temperature sensor. When rain is detected, the rain sensor sends a signal to the Arduino, which activates a gear motor through a motor driver. The motor extends the shed using forward motion. 34 A forward and reverse switch controls the motor's direction, allowing the shed to retract when rain stops. The temperature sensor monitors environmental conditions to optimize operation, ensuring the system functions reliably in varying weather. The SMPS provides stable power to the components, ensuring efficient performance. This system is fully automated, minimizing manual intervention and enhancing convenience. It is ideal for safeguarding outdoor areas such as terraces, balconies, or vehicles, offering efficient protection against sudden weather changes.

Chapter 4

RESULT & DISCUSSION

4.1 Result

Now, it's time to talk about the results. We have written our commands using the Arduino IDE and the following things can happen:

- When this project is run, it senses rain on the rain sensor, and then the shed folds or extends automatically.

- It also operates manually by a switch in manual mode.

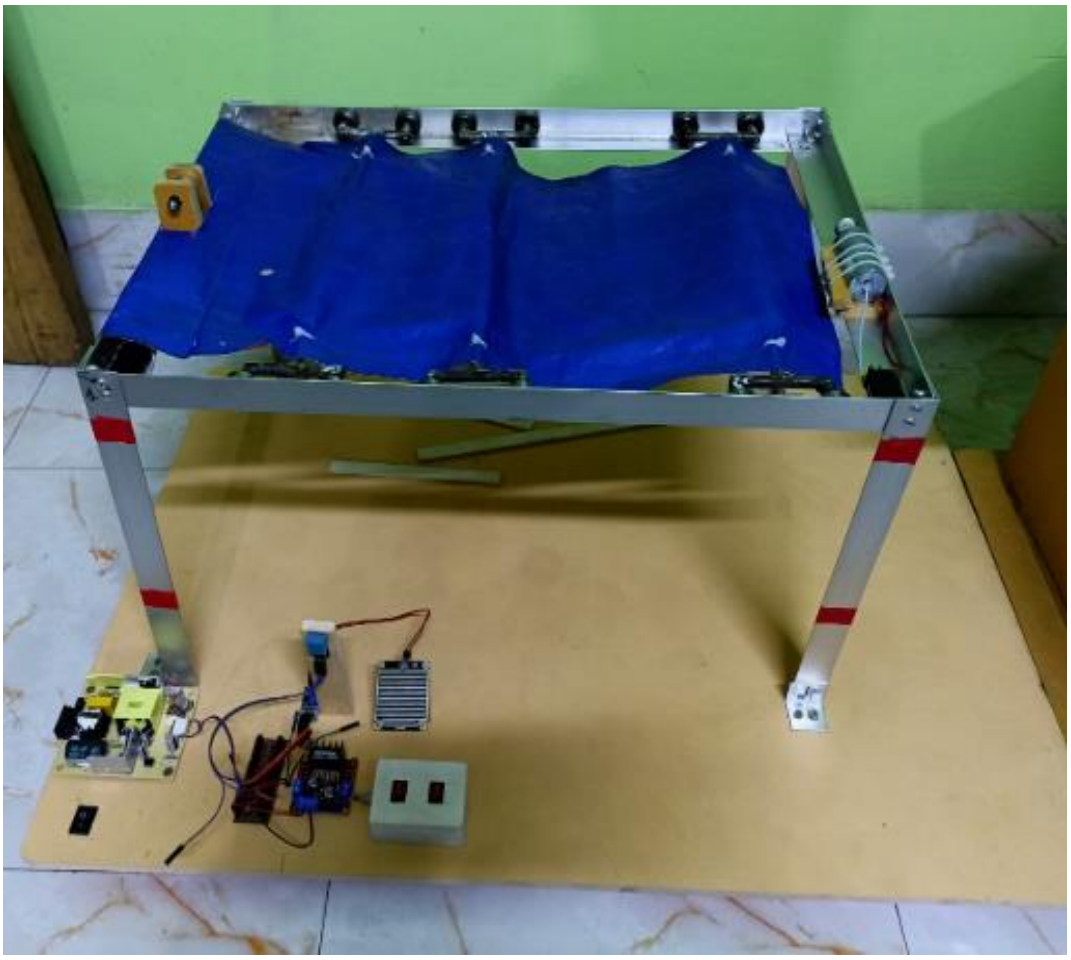


Figure 4.1: Progress Project Picture

4.2 Advantage

- Automatic Rain Protection System
- Automatic and Manual Control
- Time-saving machine for industrial work.
- No Oil consumption.
- Time-saving equipment.
- Installation is simplified very much.
- Simple construction
- Ease of operation.

4.3 Application

The project has a major application in the

- It can be used for Industrial work.
- It can be used on the football field
- It can be used on rooftop garden

4.4 Limitation

It is a demo project so we found some limitations. In the future, we will work to reduce this kind of limitation. These limitations are –

- It is a demo project so it has some errors.
- Our project may be delayed in work.

4.5 Discussion

In this project, we added all of this equipment and set it properly. Then observe our system. In the very beginning, we couldn't do it properly. After great teamwork, we solved our problem and finally were able to measure water level, water temperature and weather conditions like rain and end notifications on our phone. After using this system we really know our pond condition.

4.6 Cost Analysis

Table 4: Component Name and Price

Sl.no	Particulars	Specification	Qty.	Unit Price (Taka)	Total Price (Taka)
1	Arduino Nano		1	600	600
2	SMPS	5V 7A	1	550	550
3	Pulley		2	150	300
4	Rain Sensor	HC-SR04	1	250	250
5	Temperature Sensor	DHT 11	1	250	250
6	Forward & Reverse Switch		2	300	600
7	Bearing		12	150	1800
8	Gear Motor		1	300	300
9	Frame				1200
10	Others			1000	1000
				Total	6850/=

Chapter 5

CONCLUSION & FUTURE SCOPE

5.1 Conclusion

This is a real-time model that is used to automatically rainwater and crop-saving systems to protect crops from excess amounts of rainwater and also save water from wastage. By using Arduino operations the entire system is going to be controlled. This system saves electricity and maximizes productivity during both the rainy season and the sunny season. Controlling of the system on users virtue can also be achieved through devices like rain sensors. Hardware implementations are reliable and cheap for this project.

5.2 Future Scope

This project has great potential for enhancements and broader applications. Key future developments include:

- **Improved Accuracy**: Increasing sensor precision and refining algorithms to ensure reliable operation in diverse conditions.
- **Advanced PCB Design**: Developing custom PCBs for compact, efficient, and durable circuits.
- **IoT Integration**: Adding online connectivity for remote monitoring and control via mobile or web applications.
- **Renewable Energy**: Incorporating solar panels for sustainable and eco-friendly operation.
- **Scalability**: Expanding the design for larger areas like agricultural fields or sports grounds.
- **Smart Home Compatibility**: Integrating with smart home systems for seamless automation.
- **Self-Diagnostics**: Enabling self-check features to alert users about maintenance needs.

These upgrades will enhance functionality, making the system more versatile and practical for residential and commercial use.

Reference

[1] Carlisle, R. P. (2004). Scientific American inventions and discoveries: all the milestones in ingenuity from the discovery of fire to the invention of the microwave oven. Said.

[2] Day, S. T. (1985). U.S. Patent No. 4,534,374. Washington, DC: U.S. Patent and Trademark Office.

[3] Engst, C., Chhokar, R., Miller, A., Tate, R. B., & Yassi, A. (2005). Effectiveness of overhead lifting devices in reducing the risk of injury to care staff in extended care facilities. Ergonomics, 48(2), 187-199.

[4] Gingerich, B. S., & Ondeck, D. A. (1998). Pocket guide for the home care aide. Jones & Bartlett Learning.

[5] Hui, L., Ng, G. Y., Yeung, S. S., & Hui-Chan, C. W. (2001). Evaluation of physiological work demands and low back neuromuscular fatigue on nurses working in geriatric wards. Applied ergonomics, 32(5), 479-483.

[6] Li, W. Y. (2011). U.S. Patent No. 8,061,374. Washington, DC: U.S. Patent and Trademark Office.

[7] Martinec, A. E. C. (2011). U.S. Patent No. 7,909,048. Washington, DC: U.S. Patent and Trademark Office.

[8] Meng, S. H., Huang, A. C., Lee, C. J., Huang, T. J., & Dal, J. N. (2017, November). Design for an intelligent control system of curtains based on Arduino. In International Conference on Smart Vehicular Technology, Transportation, Communication and Applications (pp. 178-184). Springer, Cham.

[9] Nelson, A., Matz, M., Chen, F., Siddharthan, K., Lloyd, J., & Fragala, G. (2006). Development and evaluation of a multifaceted ergonomics program to prevent injuries associated with patient handling tasks. International journal of nursing studies, 43(6), 717-733.

Appendix

```
#include <Wire.h>
#include <LiquidCrystal_I2C.h>
#include <DHT.h>

// Pin definitions
int mode_switch = 2;   // SPDT for selecting mode (AUTO/MANUAL)
int control_switch = 3; // SPDT for controlling forward/reverse in manual mode
int motor1 = 4; // Motor control pin 1
int motor2 = 5; // Motor control pin 2
int rain = 6;   // Rain sensor input
int limit1 = 7; // Forward limit switch
int limit2 = 8; // Reverse limit switch
int dhtPin = 9; // DHT sensor signal pin

// DHT sensor setup
#define DHTTYPE DHT11 // Use DHT11 or DHT22
DHT dht(dhtPin, DHTTYPE);

// LCD setup
LiquidCrystal_I2C lcd(0x27, 16, 2); // 0x27 is the typical I2C address for the LCD

void setup() {
  Serial.begin(9600);

  // Configure pins
  pinMode(mode_switch, INPUT);
  pinMode(control_switch, INPUT);
  pinMode(rain, INPUT);
  pinMode(limit1, INPUT);
  pinMode(limit2, INPUT);

  pinMode(motor1, OUTPUT);
  pinMode(motor2, OUTPUT);
  digitalWrite(motor1, LOW);
  digitalWrite(motor2, LOW);

  // Initialize DHT sensor
  dht.begin();

  // Initialize LCD
  lcd.init();
  lcd.backlight();
  lcd.setCursor(0, 0);
  lcd.print("Rain Shed Ready");
```

```
  Serial.println("Rain Shed System Initialized");
}

void loop() {
  // Read the states of the switches and sensors
  int mode_state = digitalRead(mode_switch);
  int control_state = digitalRead(control_switch);
  int rainn = digitalRead(rain);
  int forward_limit = digitalRead(limit1);
  int reverse_limit = digitalRead(limit2);

  // Read temperature and humidity
  float temperature = dht.readTemperature();
  float humidity = dht.readHumidity();

  // Update LCD with temperature and humidity
  lcd.setCursor(0, 0);
  lcd.print("Temp:");
  lcd.print(temperature, 1);
  lcd.write(223); // Degree symbol
  lcd.print("C     ");
  lcd.setCursor(0, 1);
  lcd.print("Humidity:");
  lcd.print(humidity, 1);
  lcd.print("%");

  // Check for DHT errors
  if (isnan(temperature) || isnan(humidity)) {
    lcd.setCursor(0, 1);
    lcd.print("Sensor Error! ");
    Serial.println("Failed to read from DHT sensor!");
  }

  // Show mode status on LCD
  lcd.setCursor(13, 0); // Start displaying mode at column 10, row 0
  if (mode_state == HIGH) {
    lcd.print("(A)");
    Serial.println("Mode: Automatic");
    if (rainn == HIGH && forward_limit == LOW) { // Raining and forward limit not
        reached
      digitalWrite(motor1, HIGH);
      digitalWrite(motor2, LOW);
      Serial.println("Automatic: Closing Shed (Raining)");
    } else if (rainn == LOW && reverse_limit == LOW) { // No rain and reverse limit not
        reached
      digitalWrite(motor1, LOW);
      digitalWrite(motor2, HIGH);
      Serial.println("Automatic: Opening Shed (No Rain)");
    } else {
      digitalWrite(motor1, LOW);
```

```
      digitalWrite(motor2, LOW);
      Serial.println("Automatic: Motor Stopped");
    }
  } else {
    lcd.print("(M)");
    Serial.println("Mode: Manual");
    if (control_state == HIGH && forward_limit == LOW) { // Forward
      digitalWrite(motor1, HIGH);
      digitalWrite(motor2, LOW);
      Serial.println("Manual: Moving Forward");
    } else if (control_state == LOW && reverse_limit == LOW) { // Reverse
      digitalWrite(motor1, LOW);
      digitalWrite(motor2, HIGH);
      Serial.println("Manual: Moving Reverse");
    } else {
      digitalWrite(motor1, LOW);
      digitalWrite(motor2, LOW);
      Serial.println("Manual: Motor Stopped");
    }
  }

  delay(100); // Update every second
}
```