TREADMILL CONTROL SYSTEM

BY

NYARKOH AFRIYIE KWAKU UEB0909919

JENNIFER OWUSU UEB0912919

YAA BAAYAA NIMOH-ASANTE UEB0907419

A Design and Build Presented to the University of Energy and Natural Resources in Partial Fulfillment of the Requirements for the Degree of

Bachelor of Science

In

Electrical and Electronics Engineering

BUILDING A CONTROL SYSTEM FOR A TREADMILL

A Design and build Presented to the University of Energy and Natural Resources in Partial Fulfillment of the Requirements for the Degree of Bachelor of Science in Electrical and Electronic Engineering

May 2023

AUTHORS' DECLARATION

We understand that the copyright of our project is transferred to the University of Energy and Natural Resources

Nyarkoh Afriyie Kwaku _________________ ___________________

Owusu Jennifer _________________ ___________________

Yaa Baayaa Nimoh-Asante _________________ ___________________

RESEARCH SUPERVISOR

Ing. Nana Twum Duah _________________ _________________

Mr. Quarshie Ellogarh Trinity________________ _________________

ABSTRACT

This paper presents an electronic system for controlling a DC motor by using a TRIAC and the elevation motor using Arduino UNO. The system comprises an Arduino microcontroller, an H-bridge driver circuit, and, of course, the DC motor and the elevation motor. The aim of the project was to control the speed and elevation of the treadmill. The paper evaluates the performance of the speed control system by analyzing parameters such as motor speed response, speed stability, and control accuracy. The experimental results show that the implemented TRIAC control system achieves accurate speed control with good stability and response characteristics.

DEDICATION

This project is dedicated to all those who sacrifice and cooperate with us in designing and building this system. Our appreciation goes to the university for their guidance and allowing us to offer this program. We wish to express our gratitude to our Supervisor Ing. Nana Twum Duah and Mr Quarshie Ellogarh Trinity for their keen inspiration and endless encouragement till the success and completion of this work. Lastly, we dedicate this work to the Almighty God for his strength and protection and for our parents for their unwavering support.

ACKNOWLEDGEMENT

First and foremost, we would want to thank the Supreme Power of the Almighty God, who has always strengthened us in our work. This task would not have been possible without his help. Next to him is the University, which has allowed us to provide this program. We would like to take this opportunity to thank Ing. Nana Twum Duah and Mr Quarshie Ellogarh Trinity for their support and direction.

Table of Contents

LIST OF FIGURES

CHAPTER I

INTRODUCTION

1.1.Introduction

We all rely on control panels to make our treadmill workouts efficient and enjoyable. The control board controls speed, changes elevation and monitors our vital stats. However, like any electronic component, control boards can also experience problems or fail. This may be due to normal wear and tear, power surges, or technical errors. Control board repair not only saves money but also helps reduce e-waste in line with sustainable standards. This report details the method used to repair the control board, the experimental setup used, and the results. It details the troubleshooting methods used, the parts that were changed, and the testing techniques used to identify and fix problems with the controller board. The purpose of recording and sharing this repair procedure is to increase knowledge about treadmill control board repair. We believe this material will benefit treadmill manufacturers as well as technicians and fitness enthusiasts. We aim to promote a more sustainable and economical approach to fitness equipment maintenance by highlighting common problems and the best strategies for solving them. Ultimately, effective repair of the treadmill's control board will extend the life of the machine and ensure a safe and enjoyable workout for the user. We believe this project will be a useful tool and reference point for anyone seeking advice on analyzing and repairing treadmill controller boards. Together, we can advance greener and more effective ways to maintain exercise equipment.

1.2.Problem Statement

A treadmill's control board is a key component that controls many operations, such as speed control, elevation adjustment, and the tracking of vital statistics. Nevertheless, wear, overvoltage, or technical problems can cause the control board to malfunction or fail. These problems can cause the treadmill to break down or stop working, making it unusable for exercise. This is a repair request for the control board of the treadmill. The student-athletes at the University of Energy and Natural Resources who rely on treadmills for their workouts are facing major challenges due to defective control boards. Without a functioning control panel, users cannot accurately change pace or track progress. This interferes with their exercise plan and makes it difficult to reach their fitness goals efficiently.

1.3.Project Aim

- To design and develop a treadmill control system that significantly improves the performance and user experience of treadmill machines

1.4.Project Objectives

- Design and construct a control panel that effectively manages both the speed of a DC motor and the inclination angle of an elevation motor.
- Incorporate the control system into the treadmill, integrating it into the treadmill's existing framework and ensuring proper connectivity and functionality to regulate speed and elevation angle during operation.
- Test the control board to ensure it meets the desired speed and angle, and safety standards.

1.5.Scope of Work

The designing and building of a control system play a vital role in enhancing good control of the treadmill speed and elevation. Accurate regulation of the speed or angle adjustment is very crucial for comfort and exercising with the treadmill. This essay focuses on the scope of work involved in modelling and testing a digital control system using Arduino UNO, outlining the key tasks and objectives that contribute to advancing good speed regulation. The scope of the work is also limited to the use of Arduino UNO and does not include the other types of microcontrollers to control the speed and elevation of the treadmill.

CHAPTER II

LITERATURE REVIEW

Overview of Motor Control

[1] Discusses microsystem design for controlling DC motors using pulse width modulation and microblaze soft-core. This article provides an overview of the design criteria, including the use of a low-power microblaze pulse width modulation (PWM) signal to turn on and off the gate on the MOSFET that drives the high-power motor. The article also includes figures showing the proportional variation of rotation speed and voltage with duty cycle, as well as the variation of PWM with applied voltage.[2] states that DC motors are widely used in adjustable speed drives and position control applications. One common method for controlling the speed of a DC motor is through the use of chopper drives. In this method, power transistors are used to switch the motor current on and off at high frequencies, allowing for precise control over the motor speed. One type of power transistor that has gained popularity in recent years for DC motor speed control is the Insulated Gate Bipolar Transistor (IGBT). IGBTs offer several advantages over other types of power transistors, including high voltage capability, low on-state voltage drops, and fast switching speeds. Additionally, IGBTs can be easily controlled using PWM techniques.[3] discusses the use of Arduino in controlling the speed of DC motors, which are widely used in industries due to their high starting torque, response performance, and flexibility in speed control. The report highlights the advantages of using DC motors over AC motors, such as their simplicity and lower cost. [3] explain that Arduino plays a crucial role in this process by serving as a low-cost data acquisition board. They also introduce the LM393 sensor, which

measures revolutions and sends the output back to the controller. The PID controller is generated by an Arduino program and used to compare the actual speed of the DC motor with the set speed. If there is a difference, the controller will work to minimize the error and bring the motor to the set point value.[4] presents a study on the use of a fuzzy logic controller (FLC) for regulating the speed of DC motors. The authors note that traditional methods of controlling DC motor speed, such as voltage control and pulse width modulation, have limitations in terms of accuracy and efficiency. FLC is proposed as an alternative method that can handle complex and nonlinear systems. Although FLC uses linguistic variables to represent input and output variables, which makes it more intuitive and easier to, unlike traditional control methods, they only considered one type of DC motor and did not investigate the effect of load changes on the system. [5] delves into the design and development of a speed control system specifically tailored for DC motors, employing the utilization of PWM control. The author initiates the paper by emphasizing the significance of speed control in DC motors and the benefits associated with PWM techniques in achieving precise control over the motor's speed. The research presents a detailed outline of the design and assembly process for the speed control system, which comprises essential components such as an Arduino microcontroller, an H-bridge driver circuit, and, of course, the DC motor itself. PWM control is implemented through the Arduino, which generates the necessary pulse signals that effectively govern and regulate the motor's speed. The paper evaluates the performance of the speed control system by analyzing parameters such as motor speed response, speed stability, and control accuracy. The experimental results show that the implemented PWM control system achieves accurate speed control with good stability and response characteristics. [6] presents a study on the advantages of controlling the speed of DC permanent magnet motors by adjusting their permanent magnets using PWM with a relay, implemented through a PIC18F4550 microcontroller. The authors explain that this method allows for precise control of motor speed, which is important for many industrial applications. [7] explains how they used a MOSFET and two relays in an integrated driver circuit with diode clamps to drive the DC motor and how they varied the armature current using PWM of the input DC voltage. They also describe how they changed the direction of rotation of the motor by initiating an interrupt signal to the microcontroller. This paper demonstrates that their microcontroller-based system is capable of achieving precise speed control for DC motors using the PWM technique. They suggest that future work could focus on improving its performance under different load conditions or integrating it with other control systems. [8] explains the fundamental concepts behind motor speed control using PWM techniques. The authors discuss how the microcontroller generates PWM signals with varying duty cycles to control the average voltage applied to the motor, thereby regulating its speed. The paper highlights the advantages of microcontroller-based control systems, including their versatility, flexibility, and ease of implementation. The authors' focus on experimental results demonstrates the feasibility and reliability of the developed speed control system. By showcasing the performance of the system in terms of speed response, stability, and accuracy, the paper provides valuable insights for engineers and researchers interested in DC motor control.

CHAPTER III

SYSTEM DESIGN

2.1. Description of the Proposed System

In this work, we will design a purely electronic system for controlling a DC motor by using both an Arduino UNO with a triac and the elevation motor using an Arduino UNO. This microsystem controls the speed of the DC motor in two parts. The first part represents the control part containing the Arduino UNO and the Triac control system. The second assumes the role of a mediator (bridge rectifier) that converts power. An embedded system is a programmable circuit dedicated to a specific function. This is a hardware and software package that often uses a real-time approach. Depending on the application, embedded systems have to meet many of the requirements specified in the standard. Low cost, small size, low energy consumption, sufficient computing power, operational reliability, data security, and openness (interfaces).

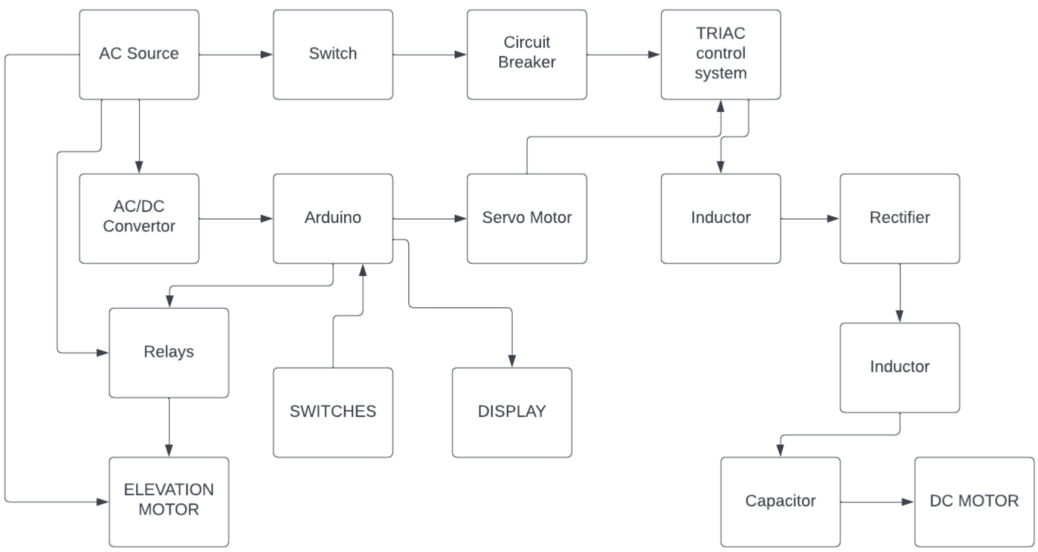

Figure 1 Block Diagram of the proposed system

2.2. Flowchart

A flowchart was made to illustrate how the system was programmed. This is the algorithm that describes the steps and processes from the start to the end. The system is initialised by turning on the power button. The reference angle and speed are then regulated by the Arduino by pressing the respective button.

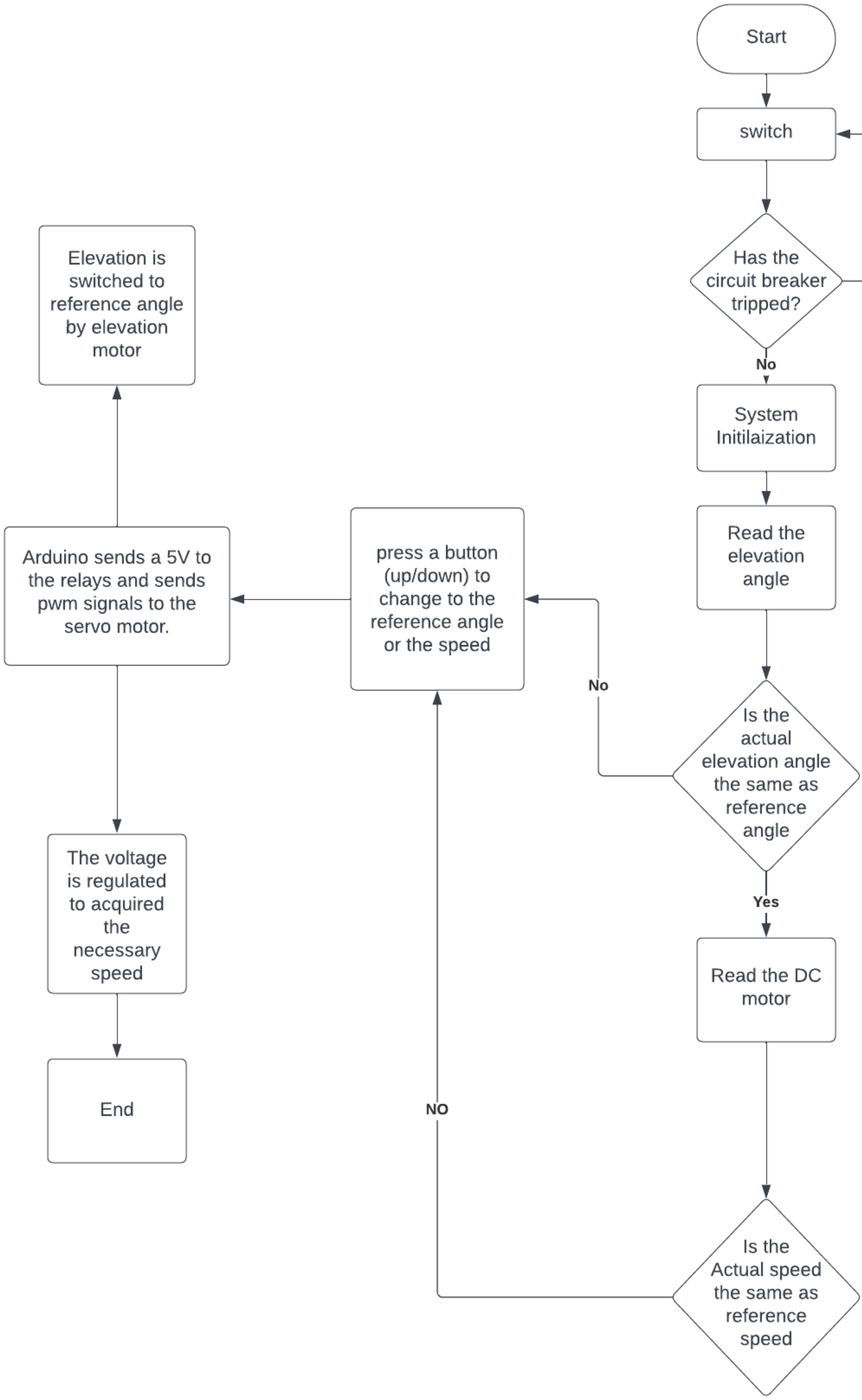

Figure 2 Flowchart

2.3. System components

The major components of the system :

TRIAC BTA41(600V,41A)

DIAC DB3 (28-36V,)

Potentiometer (150kV, 2W)

Servo motor (5V)

Arduino UNO

Relays (5V, 10A)

Elevation motor (240V)

Bridge Rectifier (15A)

Inductor (120mH)

Capacitor (400V,8900uF)

IC2 LCD Arduino Display

DC/SPEED Motor

Resistors

2.3.1. Triac Control System

A triac is a semiconductor device used to control AC power by functioning as a bidirectional switch. It has three terminals: anode, cathode, and gate, and consists of two silicon-controlled rectifiers (SCRs) connected in parallel but in opposite directions. Triacs can conduct in both directions and remain latched in the "ON" state until the current naturally falls below a threshold. To turn it off, they rely on a process called commutation, which occurs at the zero-crossing point of the AC voltage.

Control is achieved by triggering the triac with gate pulses, regulating the power delivered to the load. A snubber circuit, often a resistor-capacitor combination, is used to protect the triac from voltage transients. Triacs find applications in lamp dimming, motor speed control, and heating elements due to their ability to precisely control AC power.

2.3.2. Servo Motor

A servo motor is a specialized type of electric motor designed for precise control of position, speed, and acceleration in various applications. Its function is controlling the angular or linear position of a triac potentiometer wiper. They can rotate to a specific angle or move to a precise location with high accuracy and repeatability to allow precise control of the voltage output of the triac.

2.3.3. Arduino UNO

It serves as the central control unit responsible for generating control signals, managing feedback, and implementing control algorithms. Arduino generates control signals (PWM signals) that precisely determine the angle or position of the servo motor. The Arduino also controls the relays.

2.3.4. Relays

When a relay contact is open, it will switch power ON for a circuit when the coil is activated. Relays are used to control one electrical circuit by opening and closing contacts in another circuit. Relays can handle high-voltage and high-current loads that might exceed the capacity of small electronic components. The Arduino sends a low voltage to the relays to control the elevation motor.

2.3.5. Elevation Motor

Elevation motors are frequently used in equipment and machinery to control the height or vertical position of the system.

2.3.6. Rectifier

The primary function of a full bridge rectifier is to convert AC voltage into DC voltage. It does this by allowing current to flow in one direction (from the input terminals to the output terminals) while blocking the reverse flow. This rectification process ensures that the output voltage is always positive with respect to the input, resulting in a continuous DC waveform.

2.3.7. SPEED/DC Motor

peed DC motor is to produce rotational motion. When an electric current flows through the motor's windings, it generates a magnetic field that interacts with the field of a permanent magnet (in brushed DC motors) or another magnetic field source (in brushless DC motors). This interaction produces a torque that causes the motor's shaft to rotate.

2.3.8. Inductor

An inductor has the function of developing electromotive force in the direction that reduces fluctuation when a fluctuating current flows. Inductors can also be used as filters when combined with capacitors. The frequency of the input signal when entering the circuit is limited by the use of these filters. With increase in supply frequency, the impedance of the inductor increases.

2.3.9. IC2 LCD Arduino Display

12C LCD Arduino Display is a type of LCD display that can be interfaced with Arduino board using I2C protocol. The display has four pins: GND, VCC, SDA and SCL. The functions of the 12C LCD Arduino display are similar to those of a standard LCD display, but with the added benefit of simple wiring, requiring only two data pins to control. Display The LiquidCrystal_I2C library can be used to control displays with the same functions as the LiquidCrystal library. Functions of the 12C LCD Arduino display include displaying text and numbers, displaying special characters or symbols, adjusting the contrast of the display, and displaying custom. characters

2.4. System Design and Implementation

The development of control systems for treadmills represents a fusion of engineering, fitness, and technology. Treadmill control systems are designed to offer users a safe and versatile exercise experience, allowing them to adjust speed, elevate, and various workout parameters with ease. This project delves into the system design, processing, and implementation of a control system for a treadmill.

2.5. System Design

The design phase of a treadmill control system is crucial as it lays the foundation for the system's functionality and user experience. Several key components and considerations are involved:

1. **User Interface:** The user interface is the primary point of interaction between the user and the treadmill. It typically includes a control panel or touchscreen where users can input settings like speed and elevation, monitor their workout progress, and access pre-programmed workouts. The design of this interface should prioritize simplicity, usability, and safety.
2. **Motor Control:** Treadmills rely on one or more electric motors to drive the conveyor belt. The control system must regulate motor speed and torque to accommodate various workout intensities. It should also ensure smooth transitions between different speeds and elevation levels.
3. **Safety Features:** Safety is paramount in treadmill design. Emergency stop buttons, safety tethers, and automatic shut-off mechanisms are incorporated into the control system to prevent accidents or injuries.

4. **Data Collection:** Treadmill control systems often include sensors to collect data such as speed, distance, and elevation angle . This data can be displayed to users in real-time and stored for tracking progress over time.

2.6. Processing

Once the design is finalized, the processing phase involves the implementation of the control system's functionalities. Key aspects of processing include:

1. **Microcontroller:** The core of the control system is a microcontroller (Arduino UNO) responsible for interpreting user inputs and sensor data. It processes this information to adjust motor speed, control elevation, and monitor safety features.
2. **System Methods Development:** The control system incorporates the triac and relays to control speed and handle elevation adjustments. These methods ensure smooth and responsive operation.
3. **Safety Protocols:** The processing phase includes the implementation of safety protocols, such as detecting abnormal behaviors (e.g., sudden stops), and triggering safety mechanisms when necessary.
4. **Data Management:** Data collected from sensors and user inputs must be processed, stored, and made accessible to users. The control system manages data storage and retrieval processes efficiently.

2.7. Implementation

The final step is the physical implementation of the control system into the treadmill. This includes:

1. **Hardware Integration:** Installing the control panel, sensors, motor controller, and associated components within the treadmill's frame.

2. **User Interface Integration:** Mounting the user interface, such as a touchscreen or control buttons, in a convenient and ergonomic location for users.
3. **Wiring and Connections:** Ensuring that all electrical connections are properly established, including power supplies, sensors, and motor wiring.
4. **Testing and Quality Assurance:** Rigorous testing is conducted to verify that the control system operates as designed. This includes safety tests, usability assessments, and functional checks.
5. **User Documentation:** Providing user manuals and guides to instruct users on how to operate the treadmill safely and effectively.

CHAPTER IV

EXPERIMENTATION AND DISCUSSION

4.1. Introduction

Chapter four, experimentation and discussion present an analytical overview of the system developed. An in-debt analysis of the working of the system alongside the hardware and power testing of the system is discussed here

4.2. Testing The DC Motor

A continuity test was performed on the DC motor to check the continuity in the motor windings using a multimeter. One probe of the multimeter is connected to the positive terminal of the DC motor, and the other probe is connected to the negative terminal, after which the flywheel is then spun. The multimeter beeped to notify us that there was no open circuit in the DC motor.

4.3. Testing The Elevation Motor

A load test was performed on the elevation motor to check if it can handle certain loads by standing on the treadmill while running it. The treadmill was able to elevate to the desired angles while testing different loads. By doing this, we were able to ensure the motor could handle different weight levels.

4.4. Controlling the Elevation motor

To adjust the angle according to the user's preference, the Arduino UNO uses relays as switches to send small signals to the elevation motor. These signals serve as instructions to either open or close the appropriate circuits within the elevation mechanism. When the Arduino UNO receives input from the student, specifying the desired angle, it processes this information and determines the best course of action. Based on its calculations, the Arduino UNO sends a signal to the relay that is connected to the elevation motor. Upon receiving the signal from the Arduino UNO, the relay becomes active and completes the circuit that supplies power to the elevation motor. This allows the motor to move and adjust the angle of the mechanism accordingly. By selectively opening or closing the circuits as directed by the relay signals, the motor can rotate in the desired direction, ultimately achieving the desired angle as specified by the student.

4.5. Controlling the Speed/DC Motor

In order to operate the DC motor, the Arduino UNO sends pwm signals to the servo motor, which in turn moves the motor to a specific angle based on a set of instructions. In order to adjust the voltage of the alternating current, the potentiometer wiper is rotated by the servo motor, which is coupled to the triac. A rectifier smoothes the voltage of the alternating current before it reaches the rectifier by using an inductor. A rectifier is used to convert the voltage from alternating current to direct current and an inductor is used to reduce noise before it reaches the capacitor, resulting in a motor with better efficiency and power factor, in turn reducing energy consumption and resulting in smoother runs.

4.6. Results Analysis

The end results involved the connection of the treadmill's wiring, a meticulous process that ensured proper electrical connections and the programming of the microcontroller. Subsequently, an extensive series of tests were conducted to assess the overall performance and efficiency of the treadmill. These tests include various aspects, such as evaluating the dc motor’s speed and elevation of the elevation motor allowing the LCD to accurately display speed and angle information on it. The accuracy of these measurements was an important factor in determining the efficient functioning of the DC(Speed)motor and elevation motor in order to improve the performance of the treadmill. The results of the tests confirm that the treadmill is indeed functioning properly, meeting the expected performance standards. The LCD display reliably presents speed and angle information, demonstrating the precision of the treadmill's sensors and the effectiveness of its electronic components.

resistance(kΩ)	Firing Angle	Speed (RPM)	DC Motor (volts)
56	160	269	10
8.58	142	1340	50
2.6	120	2693	100
0.66	97	4000	150
0.18	84	4535	170

With the table, it showed the integration of the triac using an Arduino Uno to control the potentiometer(150k). This integration has been a significant enhancement to the treadmill's functionality. By using this triac control system, it can precisely control the speed of the DC motor, allowing for smoother and more gradual adjusts in speed.

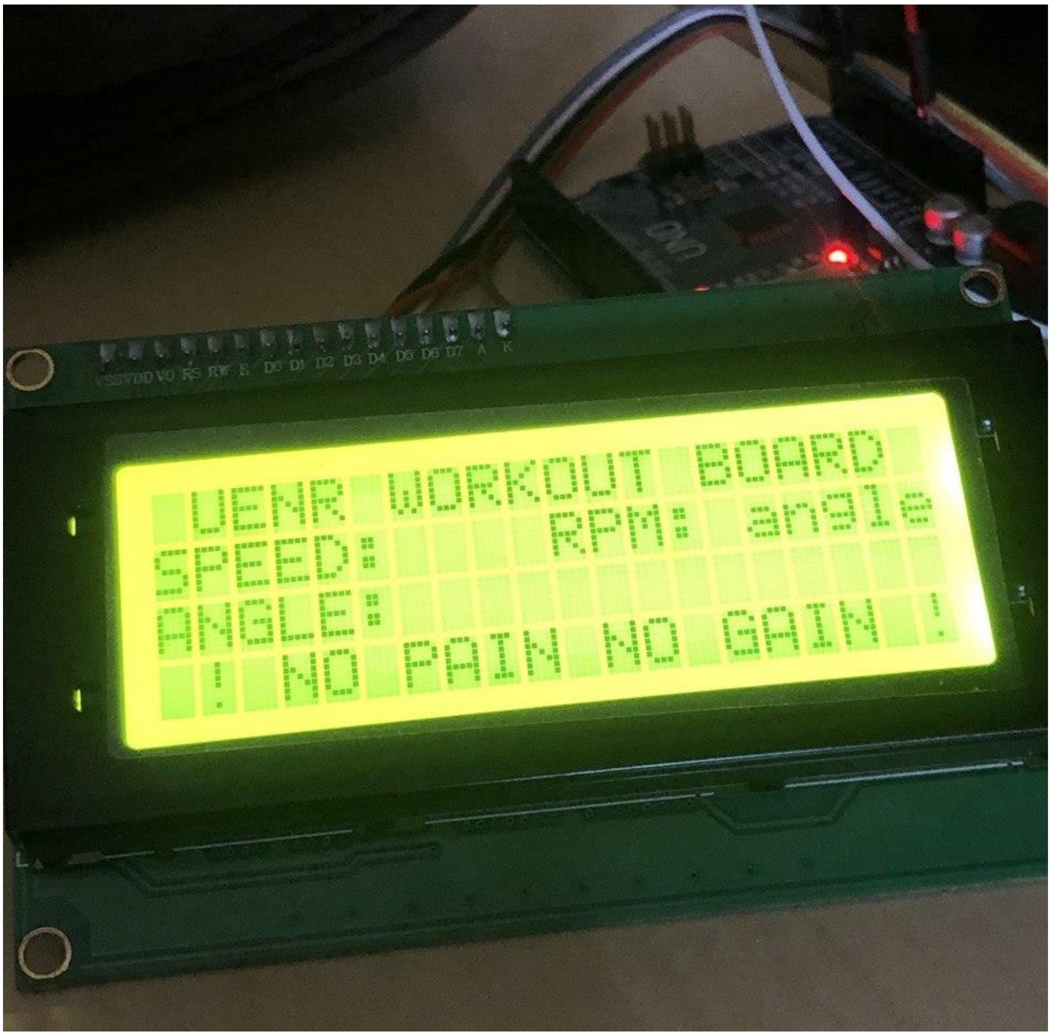

Figure 3 IC2 LCD Arduino Display

Figure 4. Treadmill

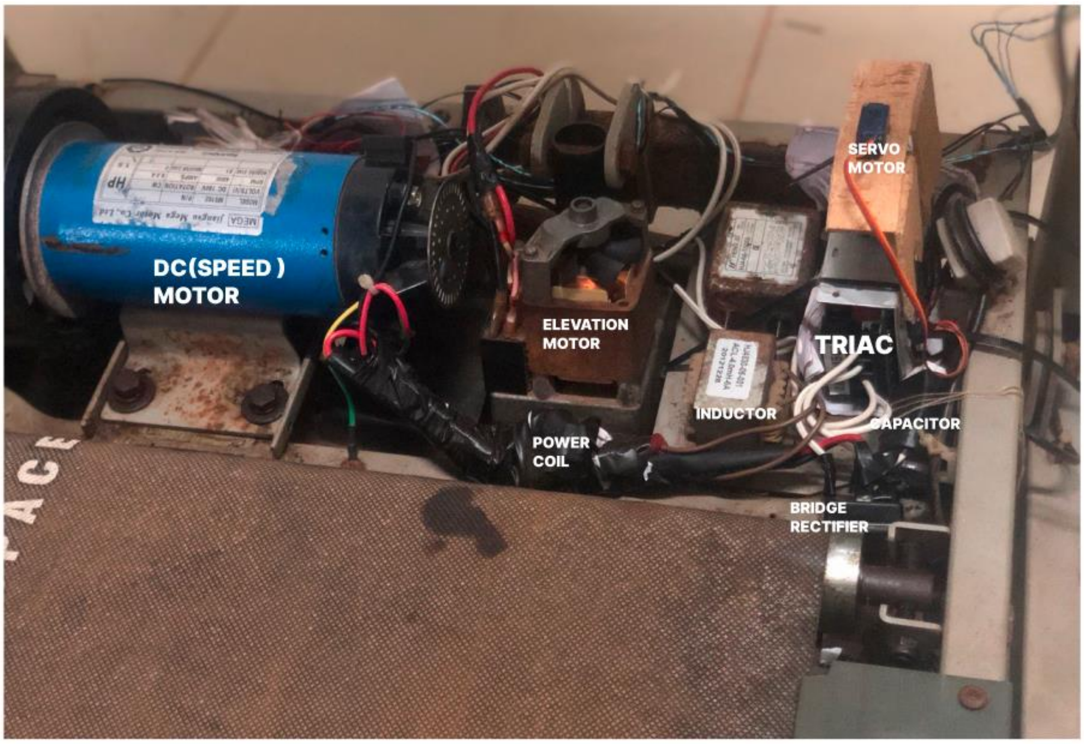

Figure 5 Control System

4.7. Challenges

Despite the significant ability of a control system, from ensuring the safety to fine-tuning the speed and elevation, there's a lot more to it than meets the eye. In this discussion, we'll explore the hurdles and complexities involved in building the control system.

4.7.1. Cost Considerations

One of the primary challenges was cost of acquiring some of the components to building the system. The initial design was to rectify the ac voltage then controlling it. DC motor control with PWM often requires additional components like transistors, optocouplers, and PWM generation circuits. On the other hand, a Triac-based AC motor control circuit might require fewer components.

4.7.2. Availability of Components

The challenge of ensuring the availability of the required components. Some specialized components, such as specialized sensors and triac have limited availability in certain regions. This led to the delay in the project timeline. In some cases, components may be available, but their quality may be questionable, particularly when purchasing from less-established or non-certified suppliers

CHAPTER V

RECOMMENDATION AND CONCLUSION

5.1. Summary

This chapter contains the summary, conclusion and recommendations of the project.The existing system was analyzed, after which information was gathered, followed by the analysis of the data. Based on the analysis made, the limitations of the existing system were identified. This was followed by the system specification, system design, system implementation and discussions. The implementation of these design units were carried out with the help of some online development tools like circuit lab to simulate and then implemented using the hardware components discussed in chapter three. The Arduino microcontroller was used to code the control of the DC and elevation motors. After the design and implementation of the various units of the system, system testing was carried out on both the various units and the general system using test data to check the expected and actual results. Finally, the performance evaluation of the system certified, since it is doing what it was designed to do.

5.2. Recommendations

5.2.1. Cost-effective solutions

Developing cost-effective DC motor control systems is a complex endeavor that requires a holistic approach. By optimizing components, embracing standardization and integration, promoting modularity, leveraging software-based solutions, and adopting open-source and community collaboration, manufacturers can significantly reduce costs. Additionally, implementing Design for Manufacturing (DFM), lean manufacturing practices, judicious materials selection, and prioritizing energy efficiency contribute to the overall cost-effectiveness of control systems. As industries continue to rely on DC motor control solutions, these strategies will play a pivotal role in making these systems affordable and accessible to a broader spectrum of applications and industries.

5.2.2. Energy Efficiency and Sustainability

In the future, there will be a greater emphasis on designing control systems that not only ensure optimal DC motor performance but also minimize energy consumption. This includes the integration of advanced algorithms, such as predictive control and machine learning, to dynamically adjust motor operation based on real-time energy demand. Additionally, building control systems will increasingly incorporate renewable energy sources, like solar panels and wind turbines, to power DC motors, making them more sustainable and environmentally friendly.

5.2.3. Artificial Intelligence and Machine Learning

Artificial intelligence (AI) and machine learning (ML) algorithms will play a pivotal role in future building control systems. These technologies will enable systems to adapt and optimize DC motor operations in real time based on changing environmental conditions, energy costs, and performance requirements. AI-driven predictive analytics will enhance fault detection and energy forecasting, contributing to greater system reliability.

5.3. CONCLUSION

Building a control system for a DC motor is a multifaceted endeavor that requires careful consideration of various factors, including motor specifications, desired performance outcomes, enviro nmental conditions, and user requirements. A well-designed control system not only enables precise manipulation of the motor's speed, direction, and torque but also enhances safety, efficiency, and reliability in diverse applications. Throughout this process, selecting appropriate control algorithms, sensors, and feedback mechanisms is important to achieve the desired level of control accuracy. Also considering factors such as fault tolerance, robustness, and energy efficiency to ensure that the control system can operate reliably in challenging environments and under adverse conditions. As we look to the future, the development of building control systems for DC motors will continue to evolve, offering innovative solutions that prioritize energy efficiency, sustainability, cybersecurity, and seamless integration with building management systems. These advancements will not only benefit industries such as manufacturing, automation, and HVAC but also contribute to the creation of smarter, more efficient, and environmentally responsible buildings and systems. Building control systems for DC motors are at the forefront of technological progress, and their continued improvement promises to revolutionize various sectors, making them more efficient, reliable, and user-friendly.

REFERENCES

[1] - Zemmouri, A., Barodi, A., Dahou, H., Alareqi, M. A., Elgouri, R., Hlou, L., & Benbrahim, M. (2023). A microsystem design for controlling a DC motor by pulse width modulation using MicroBlaze soft-core. International Journal of Electrical and Computer Engineering, 13(2), 1437-1448. DOI: 10.11591/ijece. v13i2.

[2] - Kumar, Anant, Shreerang Pradeep Munsh, Mohammed Rehan Memon, and Sitanshu Mishra. "Speed Control of DC motor using IGBT." Ph.D. diss., 2007.

[3] - Prof. P.P. Mahajan. Mahajan. Mahajan, Rounak Shetye, Pranav Vedpathak, Shubham Naik. "Speed Control of DC Motor Using Arduino." International Journal of Advanced Research in Innovative and Emerging Engineering (IJARIE), vol. 3, no. 1, 2017.

[4] - Ismail, N. L., K. A. Zakaria, N. S. Nazar, M. Syaripuddin, A. S. N. Mokhtar, and S. Thanakodi. "DC motor speed control using fuzzy logic controller." In AIP conference proceedings, vol. 1930, no. 1. AIP Publishing, 2018.

[5] - Xue, Ronghui. "Design of DC Motor Speed Control System Based on PWM Control." International Core Journal of Engineering 6, no. 8 (2020): 114-118.

[6]- Fatma, M. W., and M. I. Hamid. "PWM speed control of dc permanent magnet motor using a PIC18F4550 microcontroller." In IOP Conference Series: Materials Science and Engineering, vol. 602, no. 1, p. 012017. IOP Publishing, 2019.

[7] - Bensaoud, Othman, and Arash Tokhmechi. "Speed Control of DC Motor Using PICAXE Microcontroller." (2020).

[8] - Adejumo, O. O., and W. A. Azeez. "Development of a Microcontroller Based DC Motor Speed Control System." World Journal of Engineering and Technology 7, no. 4 (2019): 634-639.

APPENDIX I

Figures

Figure 6 Initial State

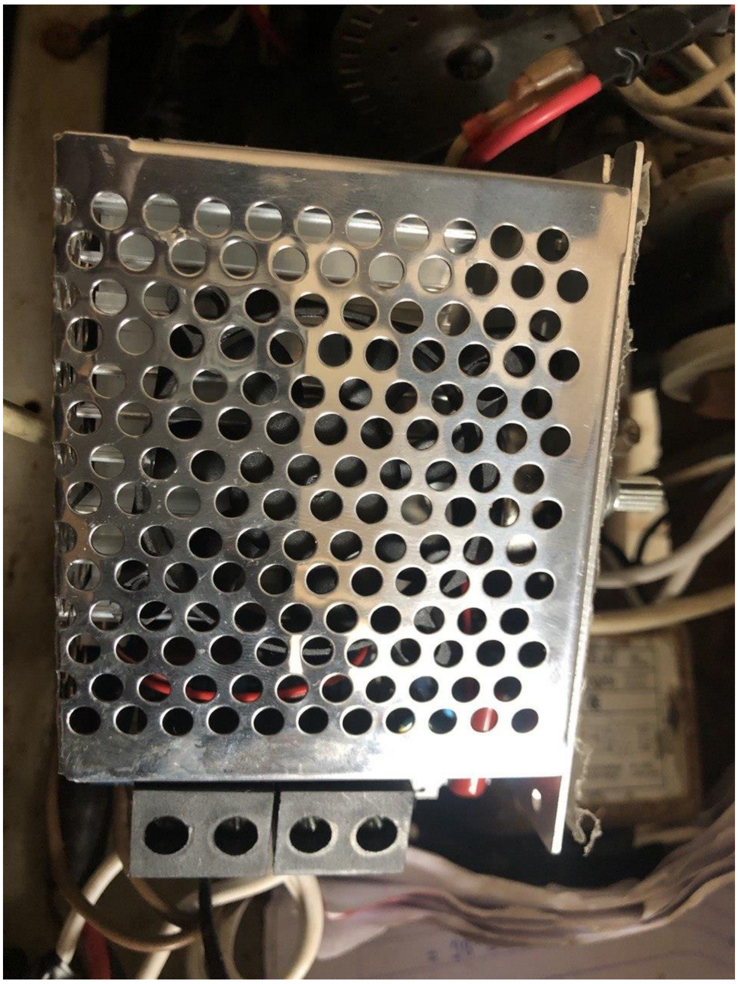

Figure 7 TRIAC

Figure 8 Servo Motor

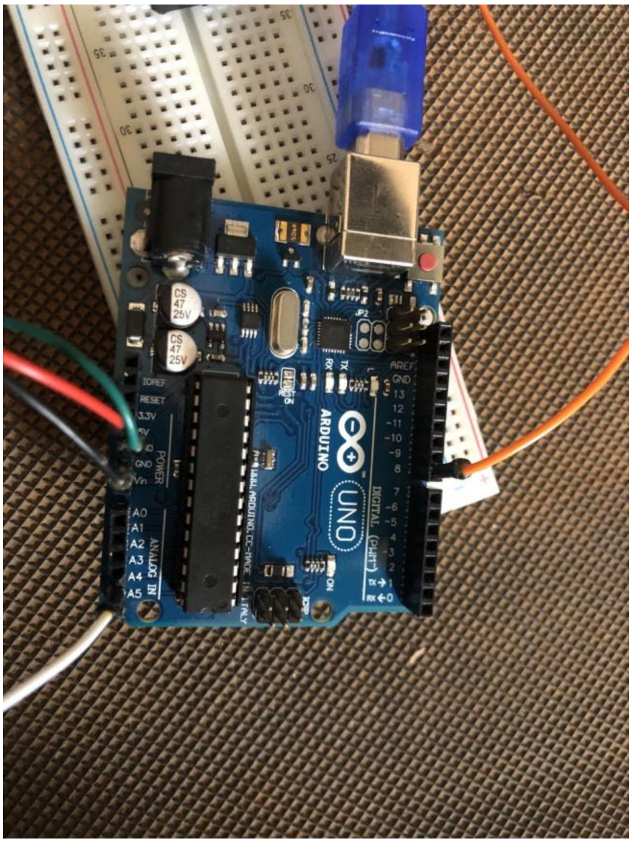

Figure 9 Arduino UNO

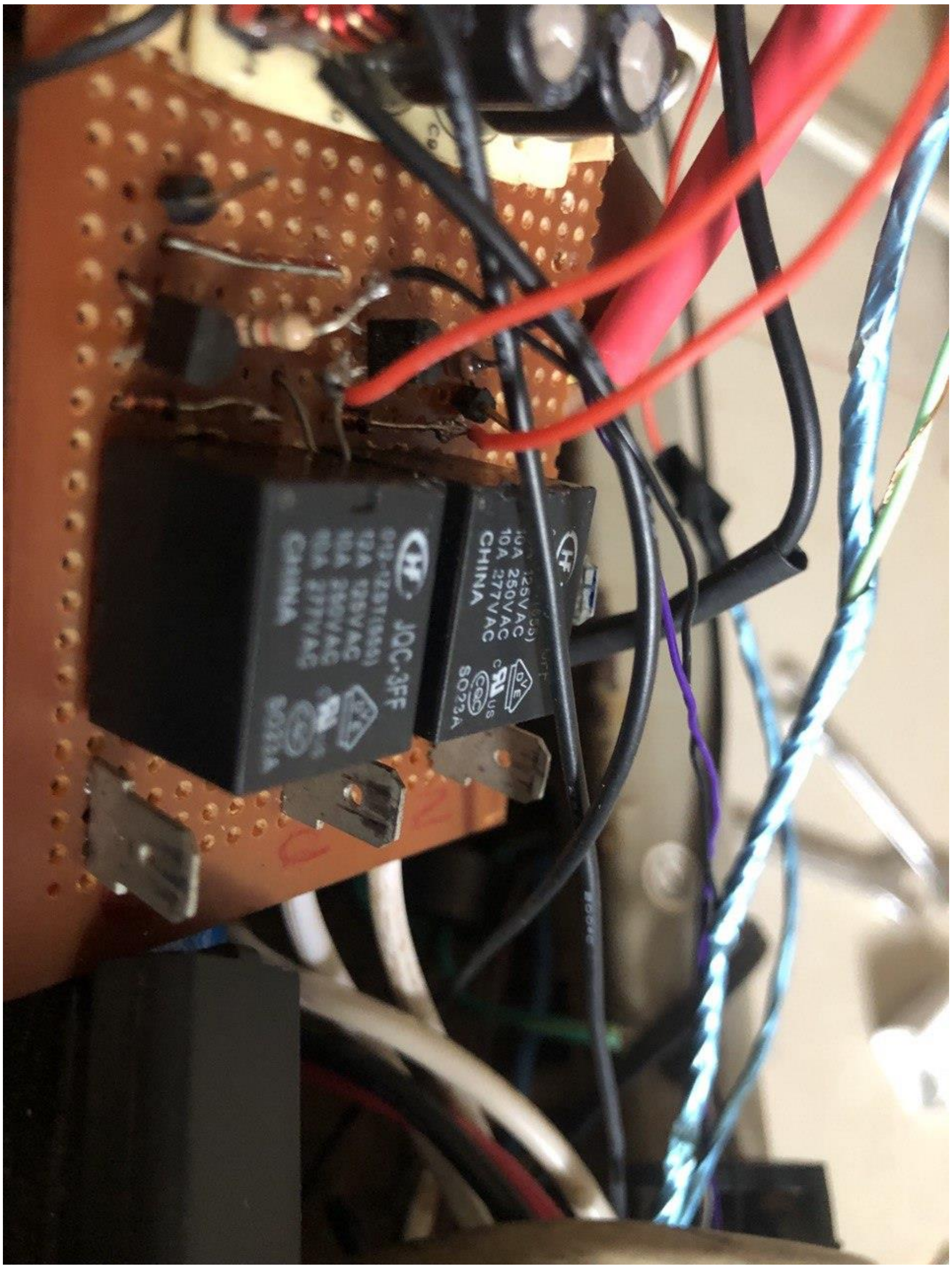

Figure 10 Relays

Figure 11 Elevation Motor

Figure 12 Rectifier

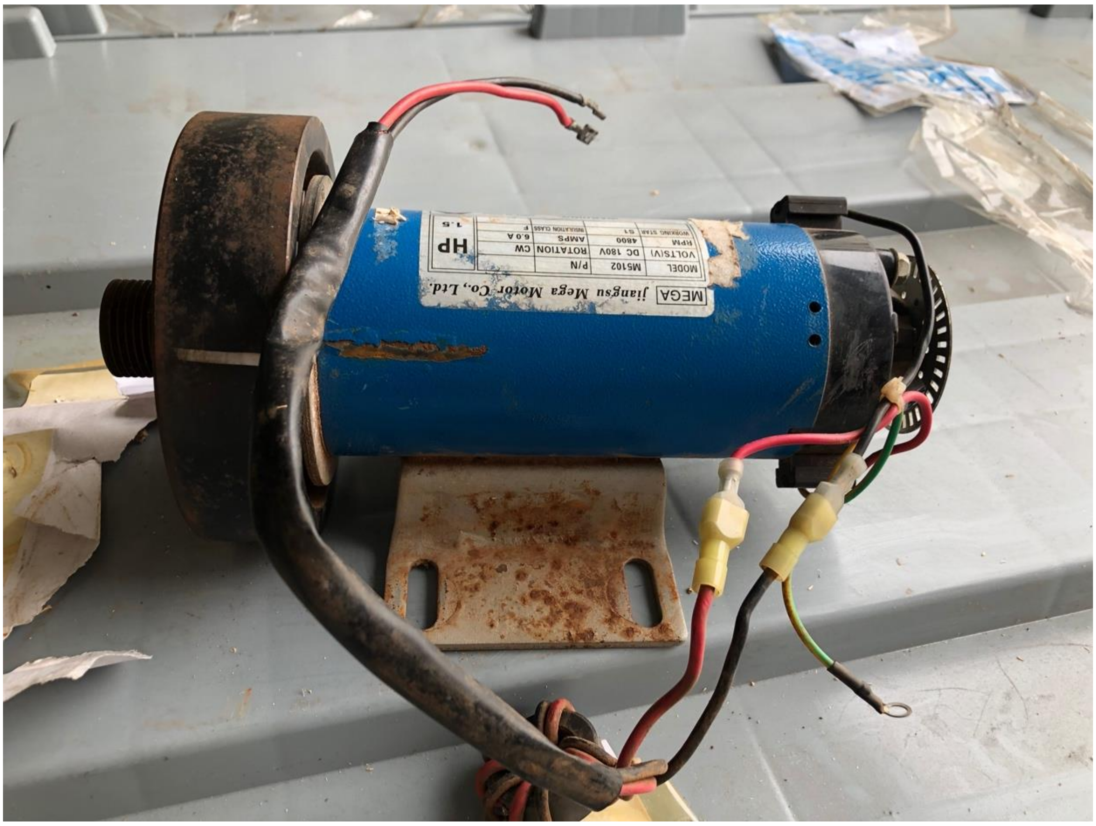

Figure 13 DC Motor

Figure 14 Inductor

APPENDIX II

CODES FOR THE CONTROL OF THE SPEED OF THE DC MOTOR AND THE ELEVATION MOTOR

```
#include <Servo.h>

#include <Wire.h>

#include <LiquidCrystal_I2C.h>


char array1[] = "THRIDMILE PROJECT BY";  // CHANGE THIS AS PER YOUR NEED

char array2[] = "     1) AFIRIYIE";      // CHANGE THIS AS PER YOUR NEED

char array3[] = "     2) JENEFA";    // CHANGE THIS AS PER YOUR NEED

char array4[] = "     3) YAABAYA";      // CHANGE THIS AS PER YOUR NEED


LiquidCrystal_I2C lcd(0x27, 20, 4); // CHANGE THE 0X27 ADDRESS TO YOUR SCREEN ADDRESS IF
NEEDED


//INCLIANTION VARIABLES

//..................................................................

const int YAABAYA1 = 4; //TO REGULATE THE INCLINATION MOTOR UP RALAY.

const int GENEFA1 = 2; //TO REGULATE THE INCLINATION MOTOR DOWN RELAY.


const int BUTTON1 = 12; //TO READ THE DIGITAL PIN 12 FROM PUSH BUTTON

const int BUTTON2 = 11; //TO READ THE DIGITAL PIN 11 FROM PUSH BUTTON


int BUTTONstate1 = 0; //VARIABLE TO STORE READ VALUE FROM PIN 12

int BUTTONstate2 = 0; //VARIABLE TO STORE READ VALUE FROM PIN 11


Servo myservo;  // create servo object to control a servo


int angle =0;    // initial angle  for servo

int angleStep =2;


#define LEFT 7   // pin 12 is connected to left button

#define RIGHT  8  // pin 2 is connected to right button
```

```
void setup() {
  // Servo button demo by Robojax.com
  Serial.begin(9600);          //  setup serial
  myservo.attach(9);  // attaches the servo on pin 9 to the servo object
  pinMode(LEFT,INPUT_PULLUP); // assign pin 12 ass input for Left button
  pinMode(RIGHT,INPUT_PULLUP);// assing pin 2 as input for right button
  myservo.write(angle);// send servo to the middle at 90 degrees
  pinMode(YAABAYA1,OUTPUT);
  pinMode(GENEFA1,OUTPUT);
  pinMode(BUTTON1, INPUT);
  pinMode(BUTTON2, INPUT);

  lcd.init();
lcd.backlight();
lcd.clear();

lcd.print(array1);              // BY DEFAULT CURSOR IS SET AT 0,0 ie, 0th ROW AND 0th COLUMN

lcd.setCursor(0,1);
lcd.print(array2);

lcd.setCursor(0,2);
lcd.print(array3);

lcd.setCursor(0,3);
lcd.print(array4);
delay(5000);
lcd.clear();

lcd.setCursor(0,0);
lcd.print(" ADJUST INCLINATION");
lcd.setCursor(0,1);
lcd.print("< OR >");
```

```
 delay(2000);
 Serial.println("RAEADY1");

BUTTONstate1 = digitalRead(BUTTON1);
BUTTONstate2 = digitalRead(BUTTON2);
Serial.println(BUTTONstate1);
Serial.println(BUTTONstate2);
if (BUTTONstate1 == LOW)
{
  digitalWrite(YAABAYA1, HIGH);
}
else{
  digitalWrite(YAABAYA1, LOW);
}
//BUTTONstate2 = digitalRead(BUTTON2);
if (BUTTONstate2 == LOW)
{
  digitalWrite(GENEFA1, HIGH);
}
else{
  digitalWrite(GENEFA1, LOW);
}
Serial.println("< OR >");

lcd.setCursor(0,2);
lcd.print(" WAIT FOR A MINUTES");

delay(7000);

}

void loop() {
 // Servo button demo by Robojax.com
 Serial.println("RAEADY2");
```

```
lcd.clear();
while(digitalRead(RIGHT) == LOW){

  if (angle > 0 && angle <= 180) {
    angle = angle - angleStep;
     if(angle < 0){
      angle = 0;
     }else{
    myservo.write(angle); // move the servo to desired angle
    Serial.print("Moved to: ");
    Serial.print(angle);   // print the angle
    Serial.println(" degree");
     }
  }

delay(300); // waits for the servo to get there
}// while

// Servo button demo by Robojax.com

while(digitalRead(LEFT) == LOW){

  // Servo button demo by Robojax.com
  if (angle >= 0 && angle <= 180) {
    angle = angle + angleStep;
    if(angle >180){
      angle =180;
     }else{
    myservo.write(angle); // move the servo to desired angle
    Serial.print("Moved to: ");
    Serial.print(angle);   // print the angle
    Serial.println(" degree");
     }
  }
```

```
delay(300); // waits for the servo to get there
}//

/*
BUTTONstate1 = digitalRead(BUTTON1);
BUTTONstate2 = digitalRead(BUTTON2);
Serial.println(BUTTONstate1);
Serial.println(BUTTONstate2);
if (BUTTONstate1 == LOW)
{
  digitalWrite(YAABAYA1, HIGH);
}
else{
  digitalWrite(YAABAYA1, LOW);
}
//BUTTONstate2 = digitalRead(BUTTON2);
if (BUTTONstate2 == LOW)
{
  digitalWrite(GENEFA1, HIGH);
}
else{
  digitalWrite(GENEFA1, LOW);
}
*/

 lcd.setCursor(0,0);
lcd.print(" UENR WORKOUT BOARD");


lcd.setCursor(0,1);
lcd.print("SPEED: ");
```

```
  lcd.setCursor(0,2);
  lcd.print("ANGLE: ");

  lcd.setCursor(10,1);
  lcd.print("RPM: angle ");



  lcd.setCursor(0,3);
  lcd.print(" ! NO PAIN NO GAIN !");
  delay(300);


}
```