Progress 5

Complex Engineering Activity

Mechatronics System Design

Project Name

MOTORIZED CAR JACK

Group Members

1 MARYAM ARSHAD 190993

2 HAMMAD AHMED 190966

3 YASIR ZAHID 191017

Department of Mechatronics Engineering

Air University

ABSTRACT

This project aims to develop a motorized car jack with maximum lifting capacity of 1 ton. The design incorporates an electric motor that is powered by a 12-volt battery. The common challenge with the currently available car jacks on the market is that they are manually operated and need a substantial amount of physical effort to lift the vehicle. Here is possible solution with concepts based on simple engineering techniques and knowledge of existing jacks. Solid works software will be used to simulate the operation and loading of the resultant design. The purpose of this study is to try and modify the design of the already existing car jacks in use.

PROGRESS REPORT 5:

SOFTWARE USED :

1. MATLAB /Simulink
2. Proteus
3. Arduino

MATLAB:

Simulink is a MATLAB-based graphical programming environment for modeling, simulating and analyzing multi domain dynamical systems. Its primary interface is a graphical block diagramming tool and a customizable set of blocks libraries. It offers tight integration with the rest of the MATLAB environment and can either drive MATLAB or be scripted from it. Simulink is widely used in automatic control and digital signal processing for multidomain simulation and model-based design.

PROTEUS:

The Proteus Design Suite is a proprietary software tool suite used primarily for electronic design automation. The software is used mainly by electronic design engineers and technicians to create schematics and electronic prints for manufacturing printed circuit boards.

AURDUINO IDE:

The Arduino Software (IDE) includes a serial monitor which allows simple textual data to be sent to and from the board. 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).

SIMULATION CODE:

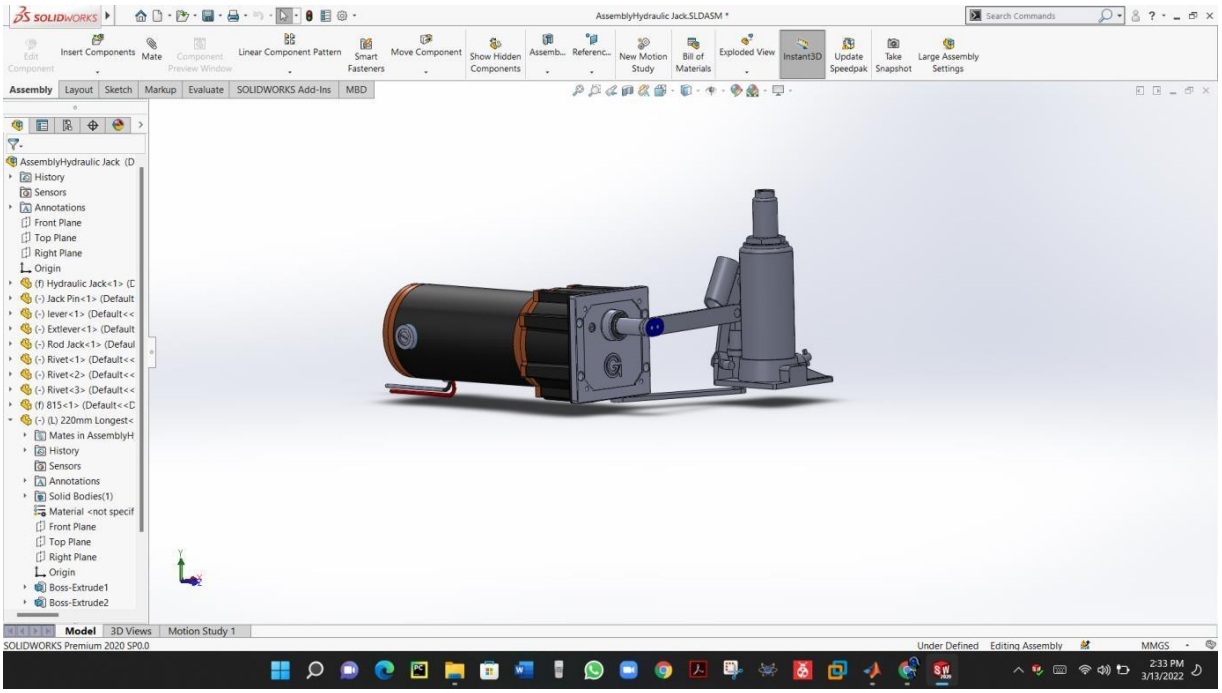
```
int y =map( duration,0,64992,0,255);

 Input = y;
```

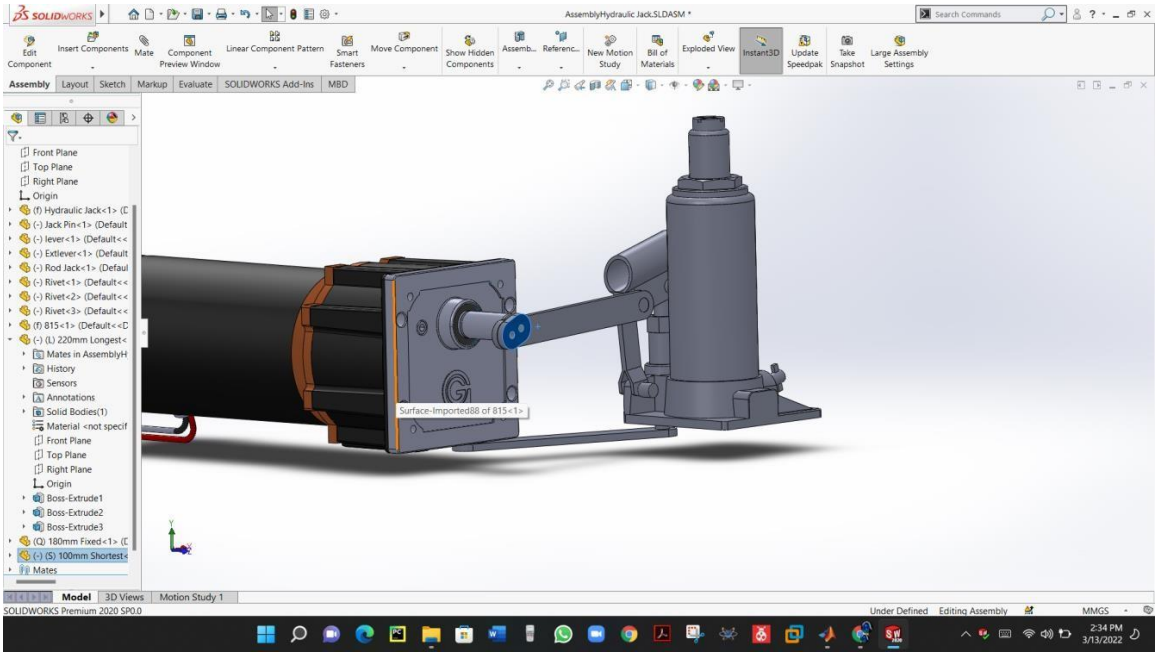
```
Serial.println(y);
myPID.Compute();
Serial.println(Output);
//duration = pulseIn(echoPin, HIGH);
//inches = microsecondsToInches(duration);
 // cm = microsecondsToCentimeters(duration);
 // Serial.print(inches);
  // Serial.print("in, ");
  // Serial.print(cm);
  // Serial.print("cm");
  // Serial.println();
  // delay(100);
//sum =z-y;
  Output=255-Output;
   analogWrite(3,Output);
  Serial.println("motor");
 //delay(1000);
}
//long microsecondsToInches(long microseconds) {
 // return microseconds / 74 / 2;
//}
//long microsecondsToCentimeters(long microseconds) {
 // return microseconds / 29 / 2;
}
```

we have not set the contant's value for pid because it will be according to final output of the system/hardware and we also need serial communication for that purpose.

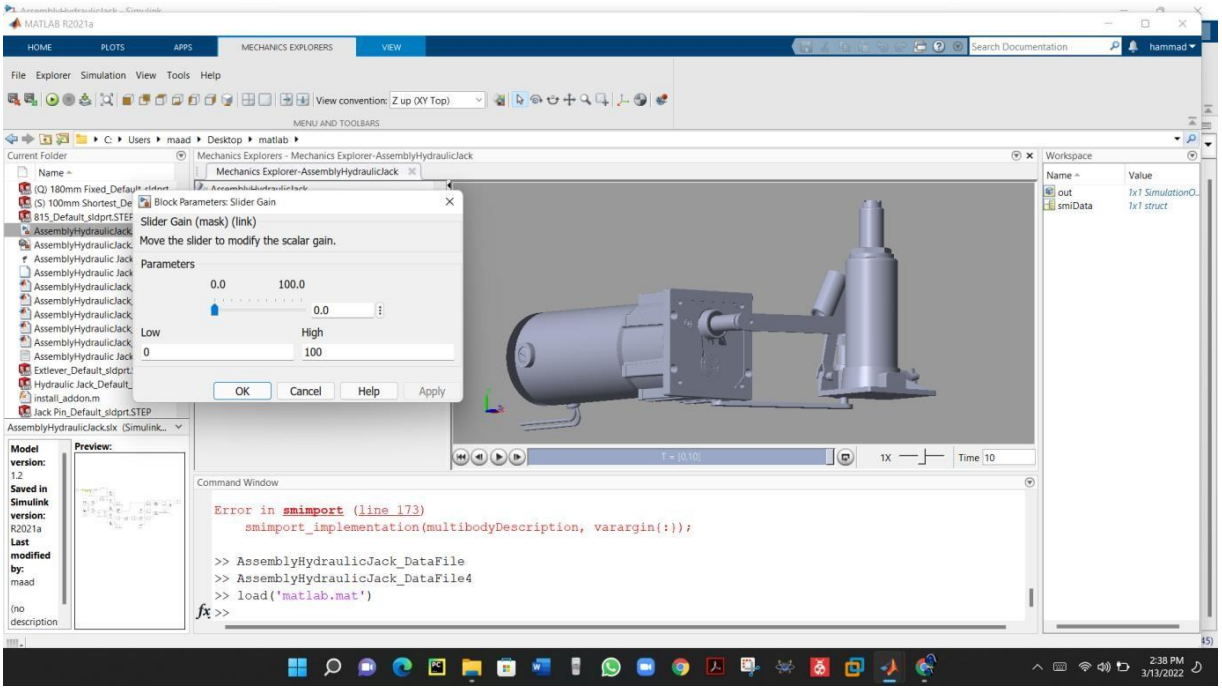
SOLIDWORKS MODEL:



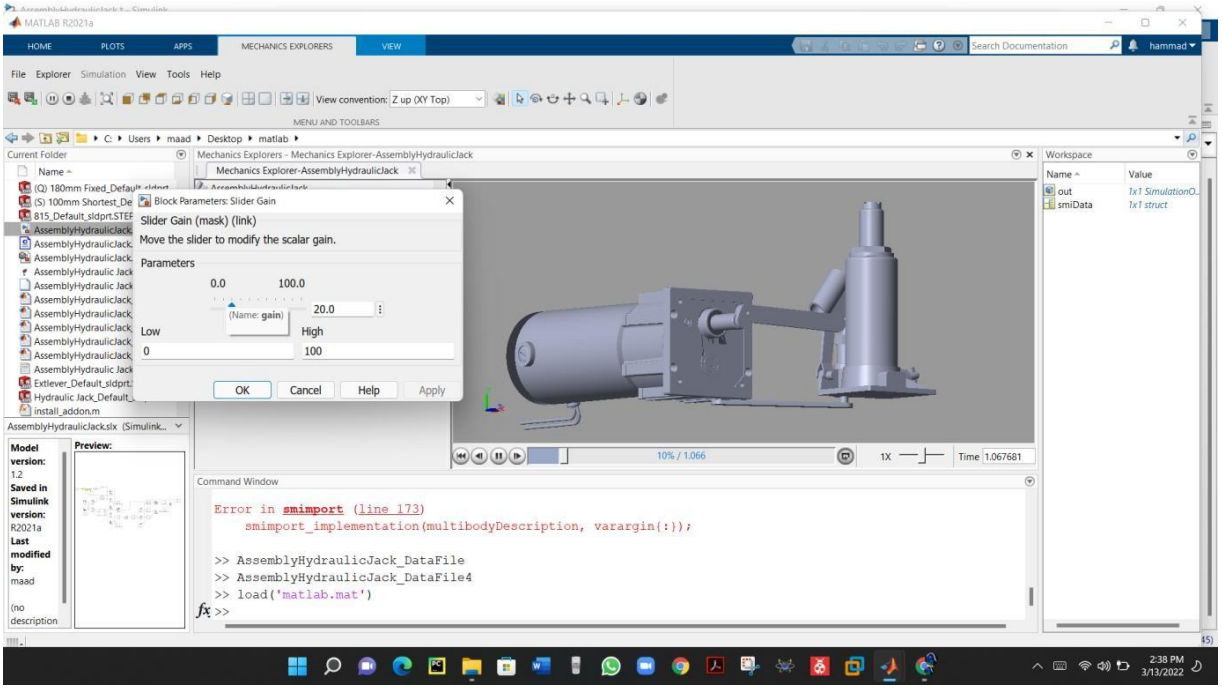
SOLIDWORKS MODEL VIEW 1



SOLIDWORKS MODEL VIEW 2



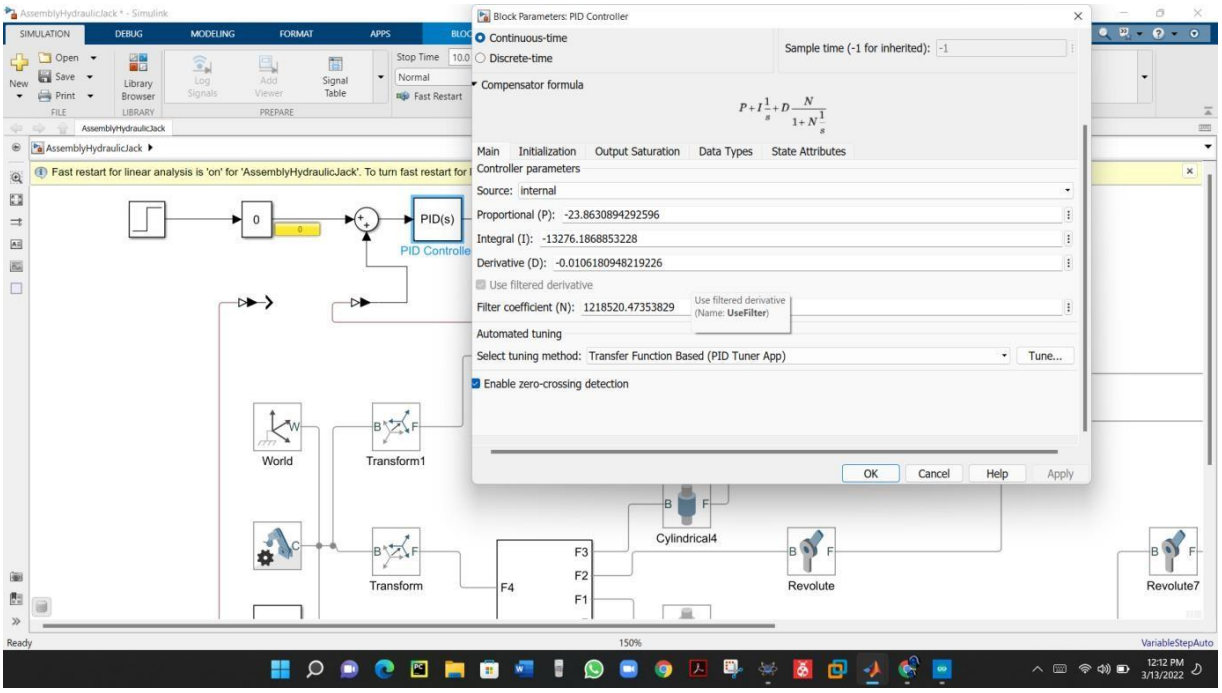
LOADING SOLIDWORKS MODEL IN MATLAB



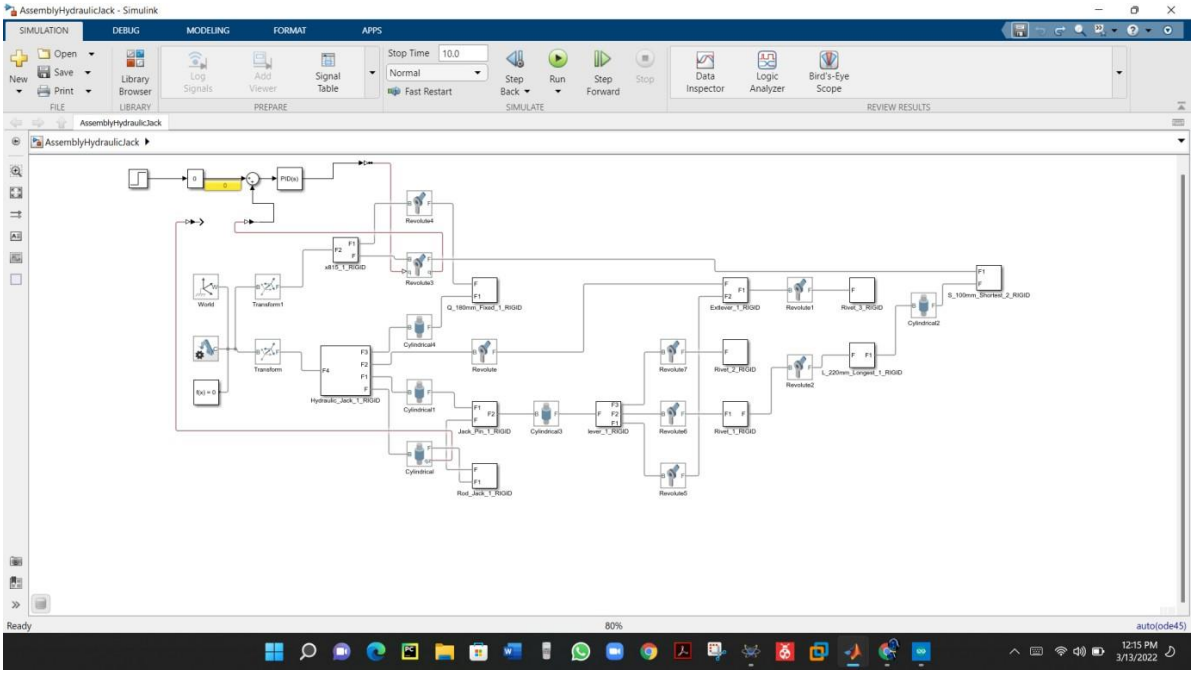
LOADING SOLIDWORKS MODEL IN MATLAB

SIMULINK MODEL:

we have used PID to control rotation angle of motor. If we want to find the output height we can switch. In that case we are controlling number of rotations or in other words we are controlling velocity.



SOLIDWORKS MODEL IN SIMULINK



SOLIDWORKS MODEL IN SIMULINK

DIAGRAM JACK :

HYDRAULIC JACK

HARDWARE COMPONENTS:

Figure :1 ULTRASONIC SENSOR

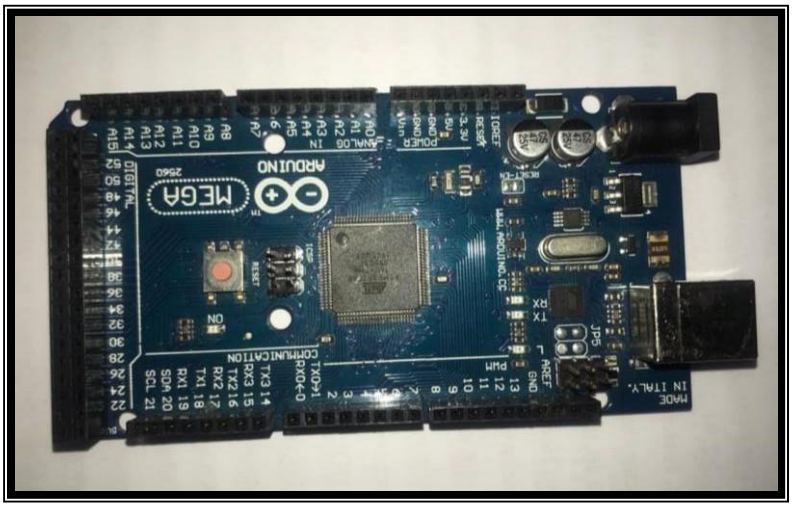


Figure ;2:MEGA 2560 CONTROLLER

Figure: 3: DC Motor

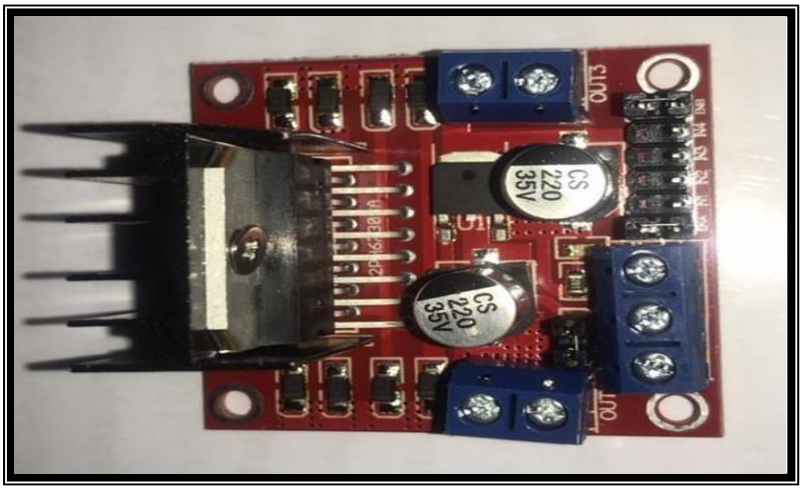


Figure: 5: Rods

Figure :6 :Jack

Figure :7 :Components

EXPLANATION:

1. Ultrasonic sensor:

It is used to get information about the height of jack uplifted. The ultrasonic sensor emits the rays and by reflecting from ground ,the distance from ground can be calculated. Initially, a reference is needed to set. From this reference the height is measured. The reference point is not ground because ultrasonic sensor is attached with jack at some height. This sensor sends feedback to controller and tells the controller what is the height of jack.

2. Arduino 2560:

Arduino Mega 2560 is used as microcontroller. It contains code for PID algorithm. It gets value from ultrasonic sensor indirectly, from comparator directly. The comparator is built in programming of Arduino. The potentiometer , ultrasonic sensor and motor driver is directly connected to Arduino. Through motor driver controller moves the DC motor. It gets information of height from ultrasonic sensor. When respective height achieves it stops the motor. But is has PID algorithm so it slows the motor as jack lifts near to desired height.

4. DC Motor:

DC high torque worm gear motor is used as actuator. The Dc motor is driven by motor driver and motor driver is controlled by controller. When controller gives the signal to motor driver, motor becomes ON and starts the Cam motion which moves the hydraulic mechanical piston up and down and moves the jack upward. As desired height is near, controller decreases the duty cycles and decreases the speed of motor. Dur to decrease in motor speed , slow lifting of jacks occurs.

5. Motor Driver L298:

It is motor driver, which gets value from controller Arduino and runs or stop the dc motor. The main purpose of motor driver is that , Arduino can't supply enough power to run the motor. We need external source to run it. So , motor driver has pins for output source by also has signal ports to control the state and speed of motors.

6. Four Bar mechanism:

Crank is attached to dc motor. When dc motor rotates, it rotates the crank and indirectly the coupler and finally rocker. The rocker is attached to piston, and moves it up and down as cam rotates. In our parts there is slot in middle, by which we can adjust the size of coupler, forming Four Bar mechanism. The longer one is for ground. The jack and motor are attached to common ground.

7. Hydraulic jack:

The piston of hydraulic jack is welded to rocker and as rocker moves the piston up and down. The fluid in jack is forced which moves the rod on other end upward by transmitting the same force. There is a screw on other side which releases the hydraulic pressure and in other words force and moves the rod downward. This jack is connected with cam rocker mechanism and a ultrasonics sensor.

SCHEMATIC/PROTEUS

(were also added in report 2)

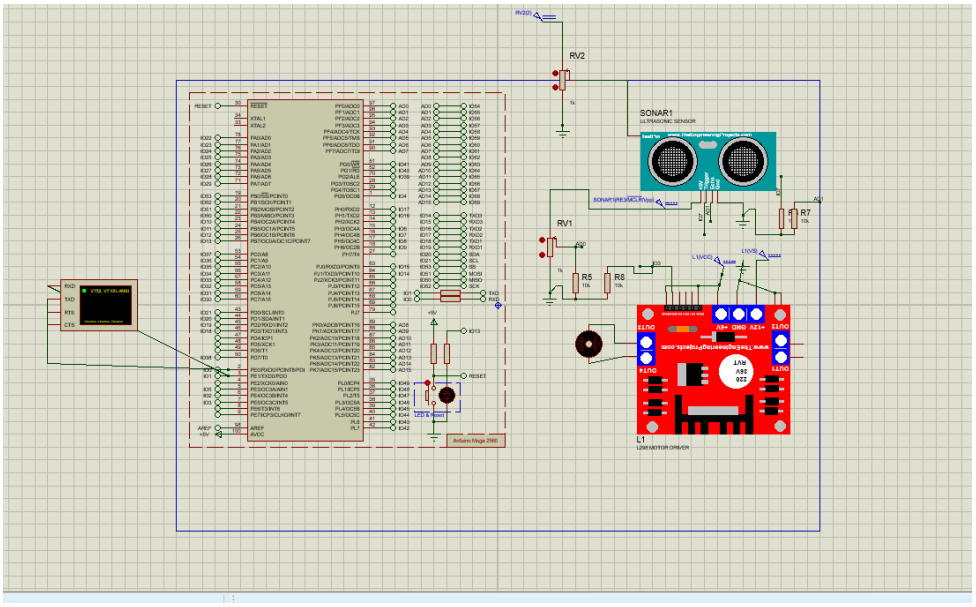


Figure: 8: proteus 1

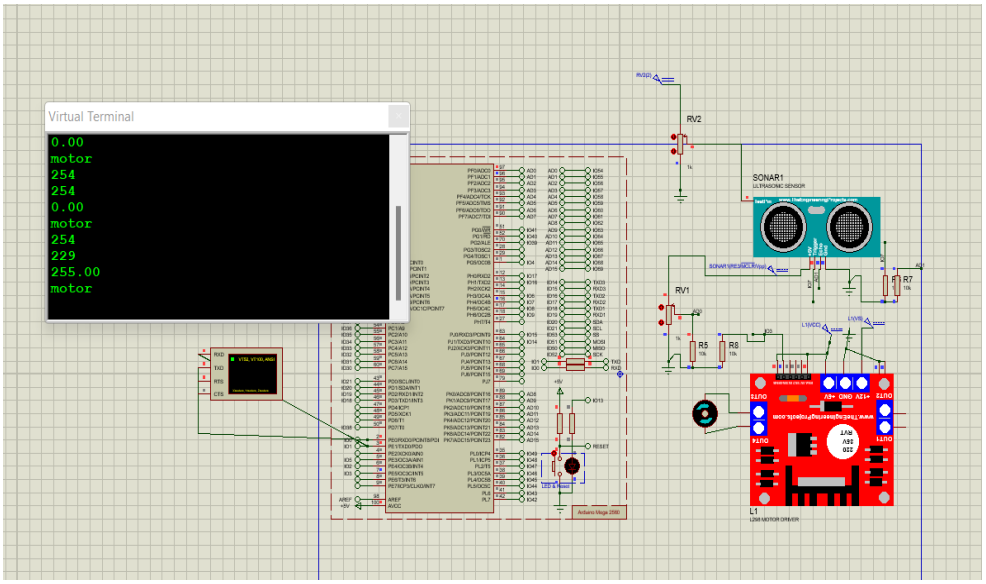


Figure :9: proteus 2

WORKING HARDWARE:

FLOW CHART:

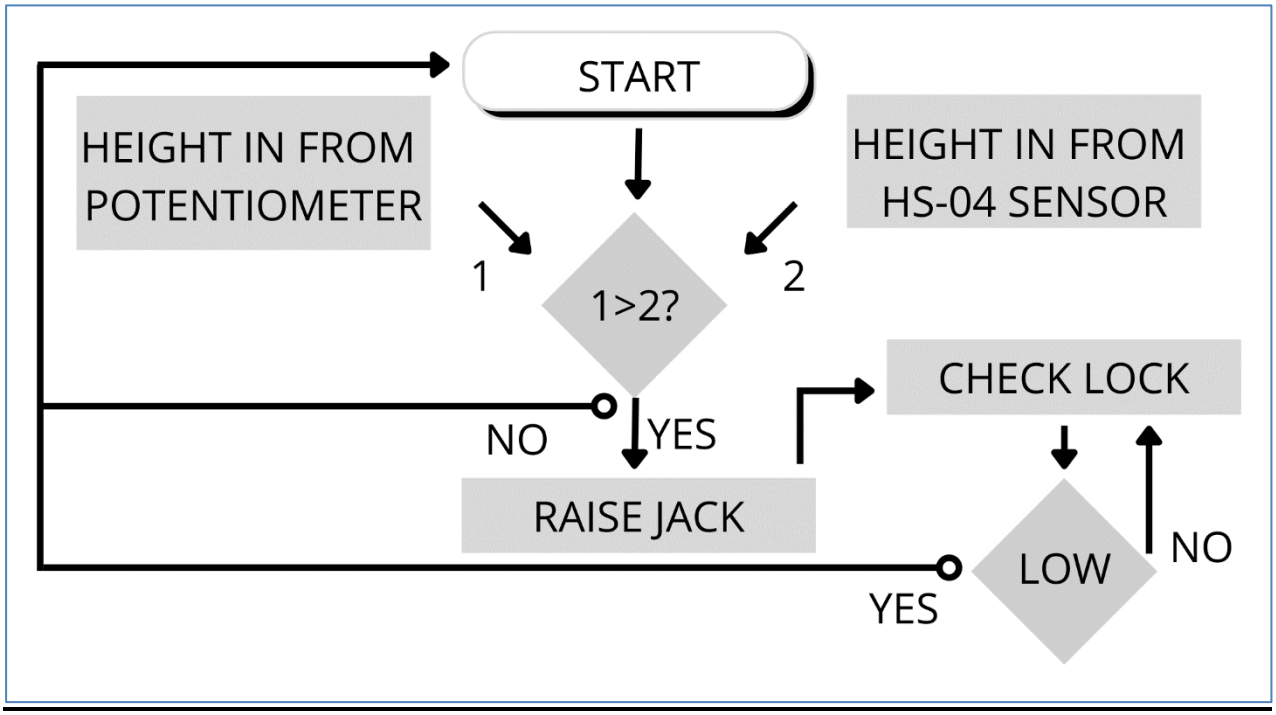
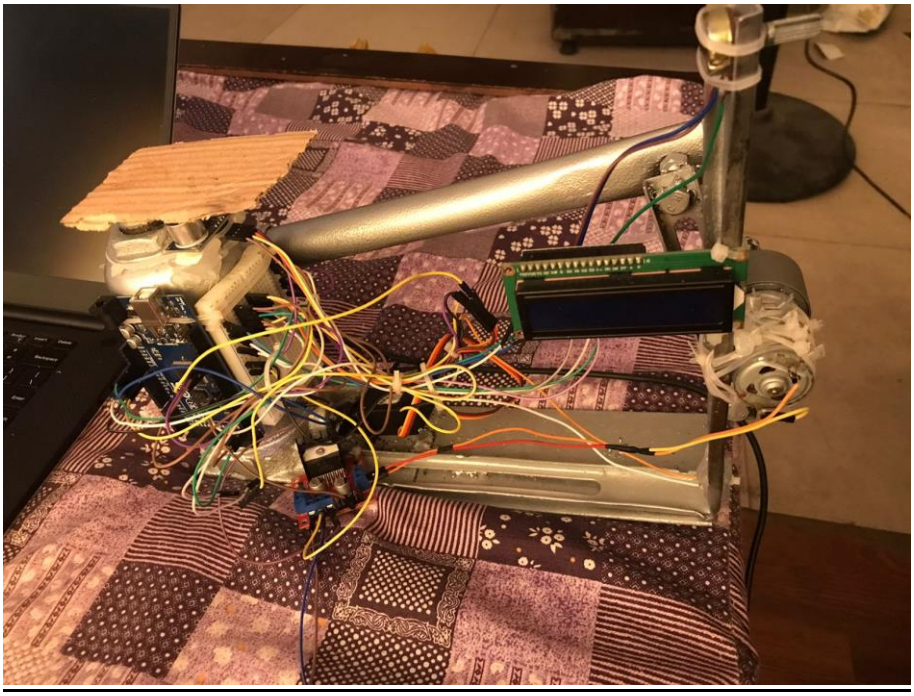
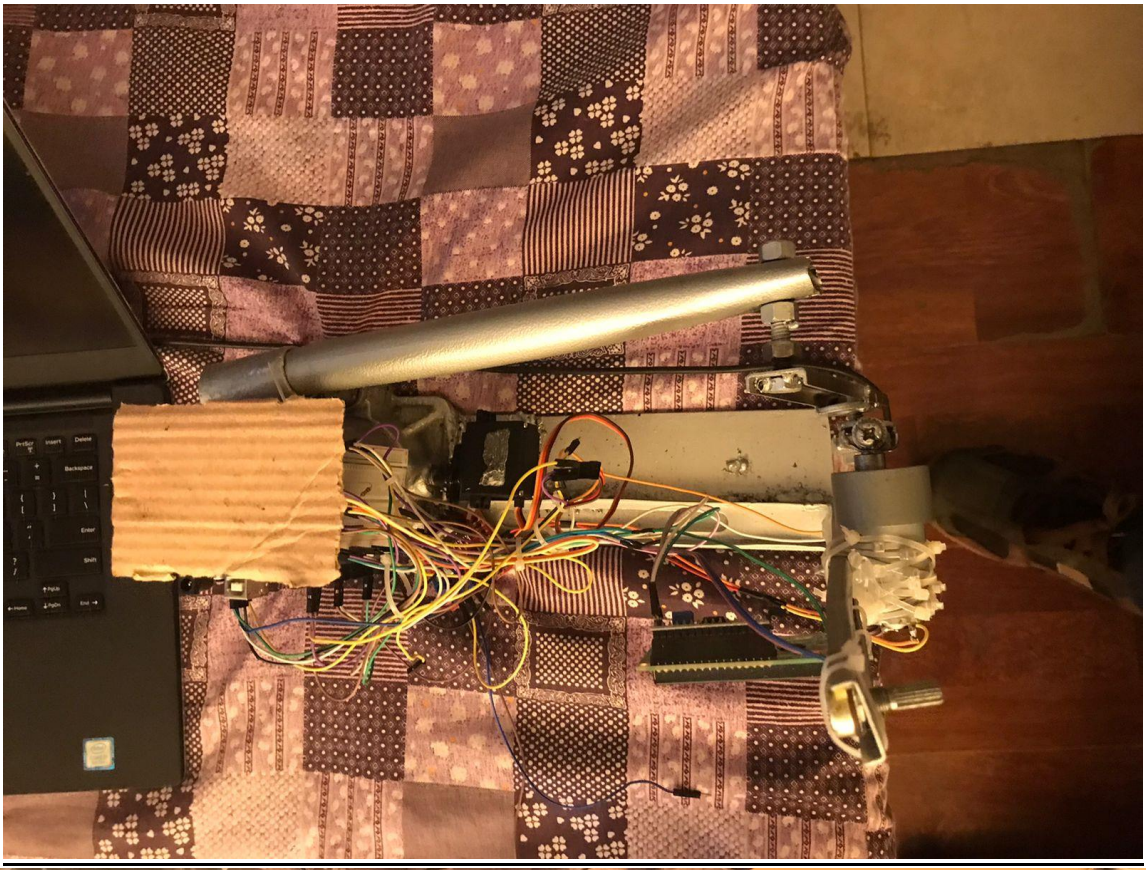
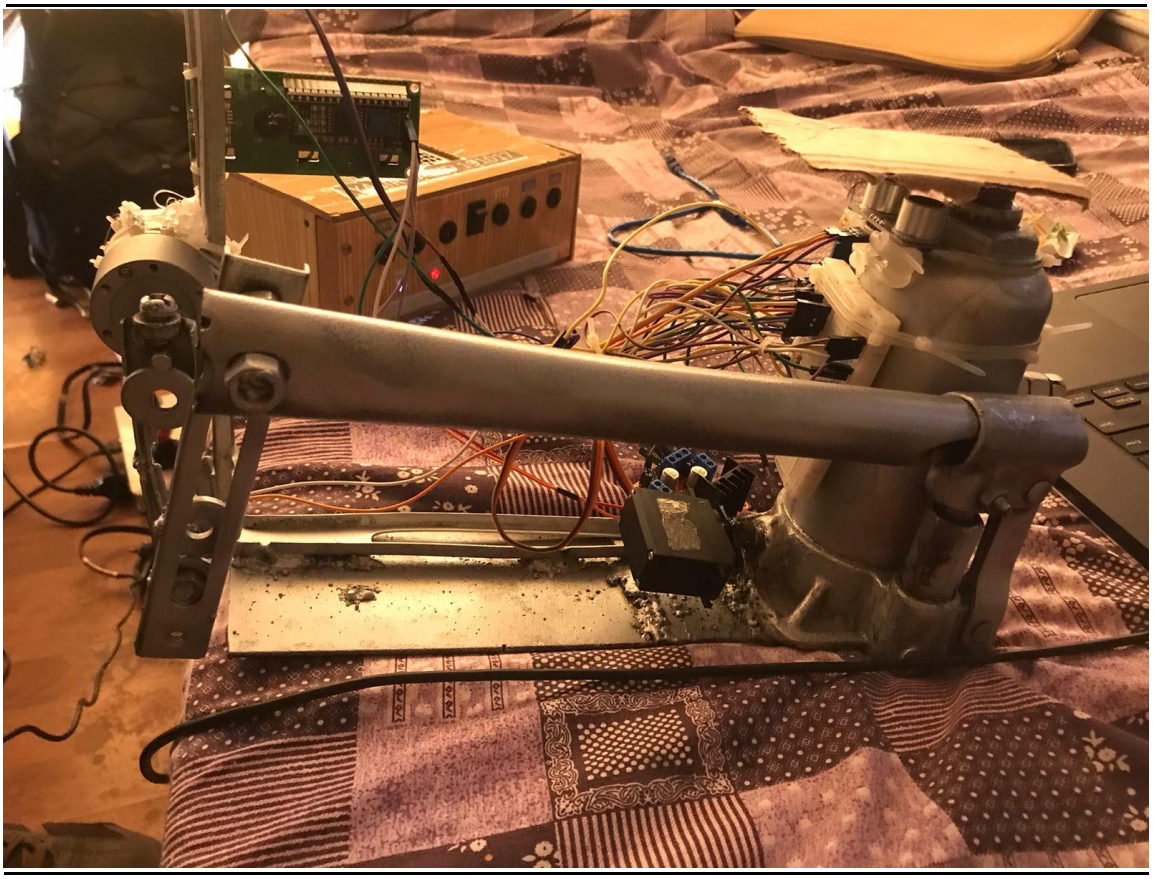


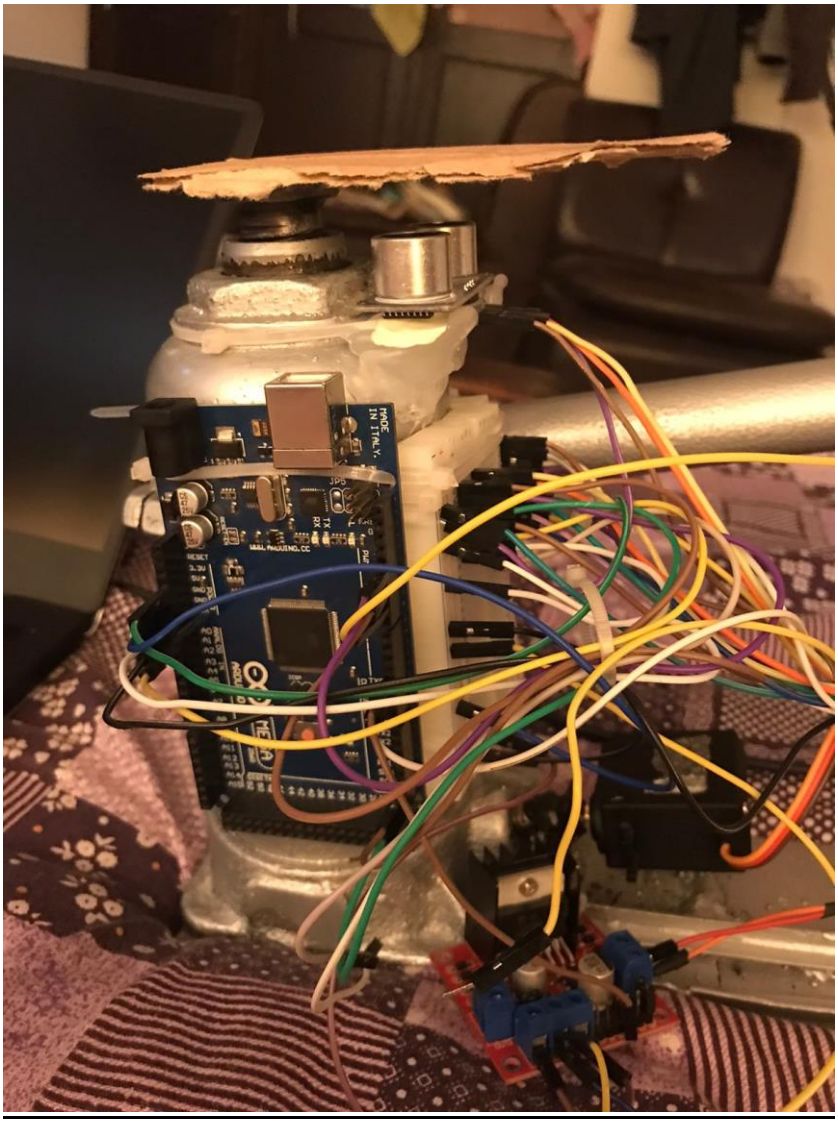
Figure 2.1 FLOW CHART FOR LOGIC

PICTURES:









VIDEO LINK:

CODE:

(ADDED AT END)

Conclusion:

Our project accomplished following objectives:

1. Successful design of a four-bar mechanism.
2. Feedback using HS-04 ultrasonic sensor.
3. Locking unlocking of lifting cylinder.
4. Display of parameters on lcd.
5. Ability to take input from user.

```
##include <Servo.h>

#include <Wire.h>
#include <LiquidCrystal_I2C.h>
#include <NewPing.h>
#define TRIGGER_PIN  11
#define ECHO_PIN     12
#define POT_PIN      A0
#define MAX_DISTANCE 20
#define D1           3
#define D2           4

int distancei,distancer,run_value;
int operate = 0;

NewPing sonar(TRIGGER_PIN, ECHO_PIN, MAX_DISTANCE);
LiquidCrystal_I2C lcd(0x27, 16, 2);
Servo myservo;

void setup() {
  myservo.attach(9);
  Serial.begin(9600);
  lcd.init();
  lcd.backlight();
  lcd.clear();
}
//
void loop() {
    if(digitalRead(2)==HIGH){                    // CODE TO LOWER THE JACK WHEN BUTTON ATTACHED AT
BUTTON 2 IS PRESSED
    myservo.write(180);
    lcd.setCursor(0,0);
    lcd.print("PISTON UNLOCK  ");
    delay(1000);
    operate=1;
    }

    if(digitalRead(2)==LOW){                     // CODE TO LOWER THE JACK WHEN BUTTON ATTACHED AT
BUTTON 2 IS PRESSED
    myservo.write(0);
    lcd.setCursor(0,0);
    lcd.print("CYLINDER LOCKED");
    delay(10000);
    operate=0;
    }

    if(operate == 1){
    distancei = analogRead(POT_PIN);
    run_value = map(distancei, 0, 1023, 0, 16);
    lcd.setCursor(0,1);
    lcd.print("SET:  ");
    lcd.setCursor(4,1);
    lcd.print(run_value);
    distancer = sonar.ping_cm();
    lcd.setCursor(7,1);
    lcd.print("PAST=   ");
    lcd.setCursor(13,1);
    lcd.print(distancer);

    if(distancer < run_value){
    while(distancer < run_value){
      lcd.setCursor(0,0);
```

```
    lcd.print("RAISING        ");
    digitalWrite(D1, HIGH);
    digitalWrite(D2, HIGH);
    distancer = sonar.ping_cm();
   }
    };
   if(distancer > run_value){
    lcd.setCursor(0,0);
    lcd.print("RAISED         ");
    digitalWrite(D1, LOW);
    digitalWrite(D2, LOW);
    delay(1000);
    };
   }else{
    lcd.setCursor(0,1);
    lcd.print("UNLOCK CYLINDER..");
    };
}
```