ELEVATOR USING AURDINO

A PROJECT REPORT SUBMITTED TO

ENGINNERING CLINICS FOR THE DEGREE

OF

INTEGRATED MTECH DEGREE

AT

VITAP-UNIVERSITY

BY

R. RASAGNA - 21MIC7025

S. AMRUTHA-21MIC7048

K.L. BHARADWAJ-21MIC7050

T. SREE-21MIC7058

A. MADHU-21MIS7022

P. NIKILESH-21MIS7087

UNDER THE GUIDANCE OF

PROF.GANESH LAKSHMANA KUMAR MOGANTI

VIT-AP UNIVERSITY

November 30, 2022

SUMMARY OF THE PROJECT:

In this project, the Arduino will be responsible for activating the elevator motor driver, controlled according to the desired floor by the buttons. Each button installed in the structure corresponds to the desired floor, which when pressed, the engine will run until the destination is reached. In this project we will understand about the structure of the Arduino elevator using the sensor reed switch module, the Arduino Mega, the step motor, as well as its drive and the Arduino UNO; Analyzing the fixation of the mechanical structure of the Arduino elevator case; Didactic through its inserts;To know the importance of each part of the Arduino elevator structure for the project as a whole; How to develop the operating logic of the Arduino elevator.

INDEX:

1.INTRODUCTION TO ELEVATOR:

An elevator or lift is a cable-assisted, hydraulic cylinder-assisted, or roller-track assisted machine that vertically transports people or freight between floors, levels, or decks of a building, vessel, or other structure. They are typically powered by electric motors that drive traction cables and counterweight systems such as a hoist, although some pump hydraulic fluid to raise a cylindrical piston like a jack.

In agriculture and manufacturing, an elevator is any type of conveyor device used to lift materials in a continuous stream into bins or silos. Several types exist, such as the chain and bucket elevator, grain auger screw conveyor using the principle of Archimedes' screw, or the chain and paddles or forks of hay elevators. Languages other than English, such as Japanese, may refer to elevators by loanwords based on either elevator or lift. Due to wheelchair access laws, elevators are often a legal requirement in new multistory buildings, especially where wheelchair ramps are not possible.

High-speed elevators are elevators that move faster than regular elevators and are common in skyscrapers and towers.

Some elevators can also travel horizontally in addition to the usual vertical motion. In this prototype, the Arduino will be responsible for activating the elevator motor driver, controlled according to the desired floor by the buttons. Each button installed in the structure corresponds to the desired floor, which when pressed, the motor will run until the destination is reached. In this review, you will understand the assembly of the didactic elevator, as well as know its mechanical structure, working.

2.BACK GROUND:

In today`s society, technology is growing at an exponential rate, and elevator is very common example of technology for consuming time. An elevator is a platform that can move up and down in a vertical direction by a mechanical mean. In the past, elevator drive mechanisms were powered by stream and water hydraulic pistons. In today`s world, there are intricate governors and switching schemes to control

elevators. In our project, the Arduino micro-controller-based elevator system is constructed to simulate as an actual elevator, and it's designed for multi-storage buildings.

The elevator design process involves description of a series of steps taken to make the logic for the 3 level elevators. Since design depends on

The particular designer‟s interests and objectives, there is no single universally accepted design procedure thus each engineer has their own twist for how the process works.

However, the most important thing is provision of a solution to a defined issue; in this case design of the elevator design process involves description of a series of steps taken to make the logic for the four-level logic for a 4-level elevator. In this design, the following assumptions will be taken as a preference and also to simplify the project and keep it to the scope.

The elevator starts and rests on the last floor of destination whenever there no more requests to be serviced. This is a one-way elevator thus it operates on a first come first serve basis ensuring that the first person to make request is attended to together with those in same direction but on the way of cab (UPWARD and DOWNWARD).

Speed used is constant since floor number is small (4 levels). The step motor has given it this ability. v. This elevator accepts multiple requests and destinations while at rest on a floor.

The input sends signals to the controller which processes the signal send to determine the necessary action. The action taken depends on the command issues at input and also the input signal received from the motor and the cab i.e., cab level/floor information which determines the necessary action by the controller.

Thus, all the logic operations are performed by the controller to enable up and down elevator motion, in order to determine the priorities in servicing the requests. The motor is connected to the cab through a belt-pulley system. This ensures movement of the cab to effect the logic operations of the elevator

3.PROBLEM STATEMENT:

The elevator model is made with the aim of acquiring knowledge of automation and regulation. It is useful for laboratory exercises because the parameters of regulator are easy to change. If the parameters are changed, then the speed of cabin will be different.

Benefits in using such system are:

ARDUINO platform is easy to program and is compatible with many operating systems,
It is easy to change elements and their parameters in code,
It is possible to measure all signals between micro-controllers and elements**.**

4. OBJECTIVE:

The main objective of the elevator control system are bringing the elevator's car to the desired floor, minimizing the time of travel, and ensuring the safe speed limit for travel.

5. PROCEDURE:

1.We started with the elevator shaft. We cut 3 pieces of fibreboard to 36" for the sides and rear (the front is open). Then using pieces of 3/4"square moulding, 36" long, we screwed the fibreboard together to make a vertical shaft. Square pieces of fibreboard on the top and bottom hold everything square. The elevator car is next.

2. The motor sits on top of the elevator. We attached a regular wood spool (firm sewing thread) to the shaft on the motor. The motor shaft has a flat edge but of course the hole in the spool is round.

3. The hole in the spool is larger than the motor shaft. To act as a bushing to take up the space, we have to use a very short piece of appropriately sized rubber hose.

4. Then we must mounted the motor to a piece of angle aluminium and screwed that to the top of the shaft. To support the other side of the spool, we put a screw through a second piece of angle aluminium.

5. We need more fibreboard to make the button panels. On the left, we used a board the height of the shaft for the call button. We used alternating red and green buttons. Finally, we have mounted 4 screw-bones. Two on the call panel, and two on the side next to the inside-elevator panel.

Buttons Recognition:

Red for down, green for up. The top floor has only a red and the bottom has only a green. Our "in elevator" panel is on the right. We must cut holes for the LED matrix and seven-segment display in addition to the 5 holes for the buttons.

6. we were concerned that something would inevitably become disconnected if we just used the standard Arduino pin connections. With the prototyping shield, we could solder wires directly and avoid that risk. In addition, it gave us a convenient place to put things like resistors, which are necessary for our 7segment display.

7.So every wire has some sort of removable end on it. We used screw blocks between the switch panels and the Arduino, and for the 7-segment and LED matrix, one end of the wire are still quick-disconnect pins.

8. For make it easy them all together with a nice custom wiring harness. The signal pins (one for the switch and one for the light on each button) are all separate, running to the screw blocks.

9.We must use solid-core wire to solder to the prototyping shield, but we used stranded wire for the crimp- on connectors to the backs of the buttons. The screw blocks gave us a place to switch from one to the other without soldering

6.RESULTS:

The prototype Elevator project:

We manly focus discuss about the problem statement facing by people solving by this prototype elevators

In this result or outcome of the project

1) We finally discussed the transportation people between one floor to other floors and saving time

The people mainly several places use elevators as for medical purpose and time saving

2)In this project we discuss about Arduino Mega, step motor and how they are used in elevator

3) in this elevator we should move from one floor to other floor in the kart with help of step motor

4) we fixed some sensors for lights and fans and motor drive for equally supply the current for Arduino

5)using this prototype Arduino project we mainly concluded the project usefulness and operation in floor.

7.CONCLUSION AND FUTURE SCOPE:

This prototype Elevator project we are devices that work with the use of several sensors in their structure.

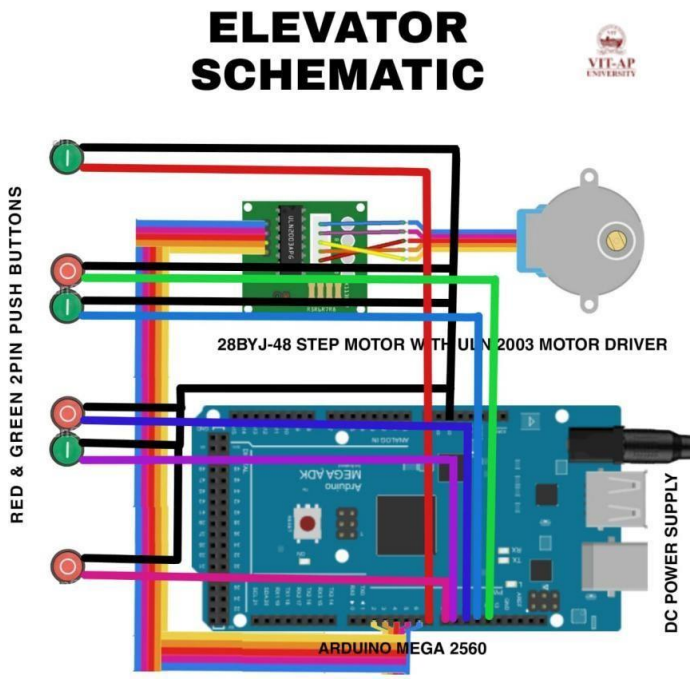
In Scope of the project

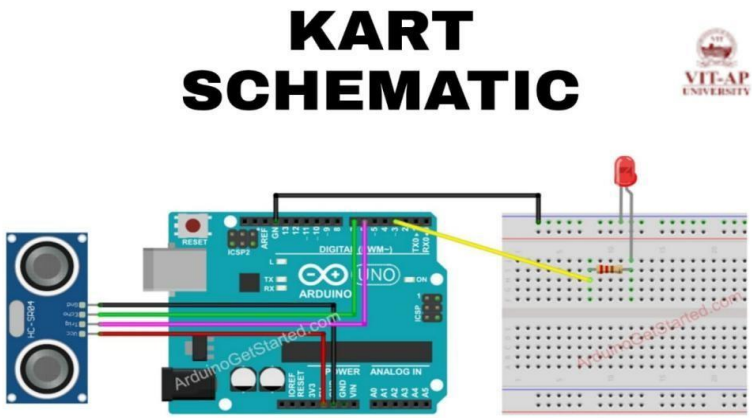
1) the being used to transport people, several places use elevators as a platform for cargo handling in construction sites, restaurants, reception of product delivery, and much more..

2) Finally, time management of humans to do their other resources.

3) using this prototype elevators project, we should save the power supply for every floor equally disruption of electricity or saving power supply.

CIRCUIT DIAGRAM:





LIFT:

8.REFERENCES:

https://www.electrosal.com/product/mini-elevator-using-arduino/

https://images.app.goo.gl/rzdaJcT17jCLD7247

9.CODE AND APPENDIX:

```
const int CW =1;
const int CCW =2;
const int STOP =3;

int Pin1 = 2;//IN1 is connected to 8 int Pin2 = 3;//IN2 is
connected to 9 int Pin3 = 4;//IN3 is connected to 10 int Pin4 =
5;//IN4 is connected to 11 int switchSTOP =2;//define input pin
for STOP push button int stopType=1;//1=normal stop, 2=hold
stop (consumes power) int speedFactor =1;//1=fastest, 2=slower
or 3 more slower

long angles[] = {2880, 2880, 1000, 2790, 2790.5, 1500};//angles of each push
button int pushButtons[] ={6, 7, 8, 9, 10, 11};//digial pin for each push button
int directions[] ={CCW, CW, CCW, CW, CCW, CW};//direction of
eacch
push button int speedFactors[] = {1, 1, 1, 1, 1, 1};//speed for each
push button


int correction_CW = 150;
int correction_CCW = 150;


int poleStep = 0; long stepVale =0; const int
SPR=64*64; long goToAngle=0; int
activeButton=0; int pole1[] ={0,0,0,0, 0,1,1,1,
0};//pole1, 8 step values int pole2[] ={0,0,0,1, 1,1,0,0,
0};//pole2, 8 step values int pole3[] ={0,1,1,1, 0,0,0,0,
0};//pole3, 8 step values int pole4[] ={1,1,0,0, 0,0,0,1,
0};//pole4, 8 step values




int count=0; int dirStatus = STOP;// stores direction status 3= stop (do
not change)

void setup()
{
  Serial.begin(9600);
  Serial.begin("elevator using arduino");
```

```
    pinMode(Pin1, OUTPUT);//define pin for ULN2003 in1 pinMode(Pin2,
    OUTPUT);//define pin for ULN2003 in2 pinMode(Pin3,
    OUTPUT);//define pin for ULN2003 in3 pinMode(Pin4,
    OUTPUT);//define pin for ULN2003 in4

    pinMode(switchSTOP,INPUT_PULLUP);
    attachInterrupt(digitalPinToInterrupt(switchSTOP), stopMotor,
FALLING );
    for (byte i = 0; i < (sizeof(pushButtons) / sizeof(pushButtons[0])); i++) {
      pinMode(i,INPUT_PULLUP);
     }



    } //setup
     void loop()
    {
       stepVale = (SPR * goToAngle)/360 ;
    for (byte i = 0; i < (sizeof(angles) / sizeof(angles[0])); i++) {
       if(digitalRead(pushButtons[i]) == LOW)
       { goToAngle
         =angles[i];
         dirStatus =directions[i];
         count =0;
         activeButton =i;
       }// if ends }//for
    loop ends


    if(dirStatus ==CCW){

      poleStep++;
      count++;
      if(count+correction_CCW <= stepVale)
      {
       driveStepper(poleStep);
      }else{
        stopMotor();
      }
     //full explannation at Arduino Course on Udemy.com see link above
    }else if(dirStatus ==CW){
     poleStep--; count++;
      if(count+correction_CW <=stepVale)
```

```
   {
    driveStepper(poleStep);
   }else{
     stopMotor();
   }
}else{
 stopMotor(); }
if(poleStep>7){
  poleStep=0; }
if(poleStep<0){
   poleStep=7;
} delay(speedFactors[activeButton]);

  //Robojax.com Stepper Push button Any Angle STPB-5

}// loop

void driveStepper(int c)
{
    //Robojax.com Stepper Push button Any Angle STPB-5
digitalWrite(Pin1, pole1[c]); digitalWrite(Pin2, pole2[c]);
digitalWrite(Pin3, pole3[c]); digitalWrite(Pin4, pole4[c]);
}//driveStepper ends here

void stopMotor()
{

for (byte i = 0; i < (sizeof(angles) / sizeof(angles[0])); i++) {
    digitalWrite(pushButtons[i], HIGH);

}//for loop ends dirStatus
= STOP;
if( stopType ==2)
{ driveStepper(8);
}
}//stopMotor()
```

"THANK YOU"