زانكۆی سەلاحەددین - هەولێر

Salahaddin University-Erbil

Elevator Control system Based on Arduino Microcontroller

Research Project

Submitted to the department of (Electrical) in partial fulfillment of the requirements for the degree of BSc. in (Electtical engineering)

Prepared by:
Barzi law
Huda jihanger

Supervised by:
Mr.Goran wnis hama ali

2022-2023

SUPERVISORS CERTIFICATION

We certify that this project, titled “Elevator Control system Based on Arduino Microcontroller” and presented by “Barzi law & Huda jihanger” was prepared under our supervision at the salahaddin university-college of engineering as a partial requirement for the BSc. degree of Department of electrical engineering.

Supervisor

Signature:

Name: Mr. Goran wnis hama ali

Date: /   /

ABSTRACT

This project proposes the development of an Elevator Control System based on the Arduino microcontroller. The aim is to design a cost-effective and efficient elevator control system that is capable of managing elevator movements and ensuring passenger safety. The system will incorporate sensors, microcontrollers, and actuators to control the elevator's motor and ensure that it stops at the desired floor.

The elevator control system will use a combination of digital and analog signals to monitor and control the elevator's position and speed. The system will be designed to respond quickly to user inputs and make necessary adjustments to ensure safe and reliable operation.

The implementation of the Elevator Control System will involve the use of an Arduino Uno board, a stepper motor driver, and various sensors such as limit switches and ultrasonic sensors. The system will be programmed in the Arduino programming language and will be designed to be easily configurable and scalable.

The Elevator Control System will be tested using a simulation environment and in a real-world setting. The system's performance will be evaluated based on its response time, accuracy, and safety. This project aims to provide a practical solution for controlling elevators that is both affordable and reliable.

ACKNOWLEDGEMENTS

Any attempt at any level can't be satisfactorily completed without the support and guidance of learned people. We would like to express my immense gratitude to our final year project supervisor Mr.Goran wnis hama ali for his constant support, motivation, patience and effective assistance that has encouraged us to come up with this project. Besides that, We gratefully acknowledge all help and valuable information from our friends and mates in the process of this project. The precious and endless opinions from them really helped us in the whole process. Last but not least, thanks are given to our family for their flawless love and always supporting us during undergo of our project.

Table of Contents

List of Figures

CHAPTER ONE

1.1 INTRODUCTION

An elevator is a platform, either open or enclosed, used for lifting people or freight to upper floors within a building. Elevators are a standard part of any tall commercial or residential building. Manually operated elevators were used for lifting freight in warehouses and manufacturing plants Elevator is a transport device that is very common to us nowadays. We use it every day to move peoples vertically in a high building such as shopping center, working office, hotel and many more. It is a very useful device that moves people to the desired floor in the shortest time. Lift or elevators are generally powered by electric motors. The elevator control system is one of the important aspects in electronics control module in automotive application. In this project the elevator control system is based in Arduino, and the elevator control system is implemented for multi- floor building. Control systems on early elevators required human operators to regulate the speed of the lift and descent, to stop the elevator at each floor, and to open and close the doors. In the 1950s automatic pushbutton control systems replaced manual controls. In the 1970s electromechanical controls were gradually replaced with solid state electronic controls. An elevator (a lift in British English) is a type of vertical transportation that moves people or good between floors, levels or decks of a buildings, vessels or other structures. In this system, elevator has four stations and the cabin speed depends on difference between current position and position where cabin needs to go. It has four switches for four floors. Magnetic Contact Sensors are used as position sensors for detecting current elevator cabin position and stop the DC motor at reaching the user desire floor.

ARDUINO UNO

Arduino/Genuine Uno is a microcontroller board based on the ATmega328P (datasheet). It has 14 digital input/output pins (of which 6 can be used as PWM outputs), 6 analog inputs, a 16 MHz quartz crystal, a USB connection, a power jack, an ICSP header and a reset button. It contains everything needed to support the microcontroller; simply connect it to a computer with a USB cable or power it with an AC-to-DC adapter or battery to get started... You can tinker with your UNO without worring too much about doing something wrong, worst-case scenario you can replace the chip for a few dollars and start over again.

"Uno" means one in Italian and was chosen to mark the release of Arduino Software (IDE) 1.0. The Uno board and version 1.0 of Arduino Software (IDE) were the reference versions of Arduino, now evolved to newer releases. The Uno board is the first in a series of USB Arduino boards, and the reference model for the Arduino platform; for an extensive list of current, past or outdated boards see the Arduino index of boards.

1.2 Background

Manually operated elevators were used for lifting freight in warehouses and manufacturing plants as early as the 1600s. The modern elevator is a direct descendant of a design first shown by Elisha G. Otis at the New York World's Fair in 1853. A notable feature of the Otis elevator, and the principal reason for its popular acceptance, was a safety device that immediately engaged and held the elevator in the event the hoisting cables broke. The first elevators were operated by steam power to turn the cable drums. In the 1800s, the first hydraulic elevators were introduced using water pressure as the source of power. At first the hydraulic rams were one-piece, which meant a hole had to be dug under the elevator shaft as deep as the elevator was to be high. Later multiple section, telescoping hydraulic rams allowed shallower holes. The first commercially successful electric elevator was installed in 1889, and electricity quickly became the accepted source of power. Electric-powered elevators offered two significant advantages. First, electric power was clearly becoming universally available, and any building likely to be equipped with an elevator would also have electric power. Second, hydraulic elevators were severely limited in the height to which they could rise, while electric elevators, using a simple cable and pulley system, had virtually no height limit. For many years, electric elevators used either direct current (DC) motors or alternating current (AC) motors. The power elevator debuted mid-19th century in the U.S. as a simple freight hoist operating between just two floors in a New York City building. By 1853, Elisha Graves Otis was at the New York Crystal Palace exposition, demonstrating an elevator with a "safety" to break the cab's fall in case of rope failure, a defining moment in elevator development. By 1857, the country's first Otis passenger elevator was in operation at a New York City department store, and, ten years later, Elisha's sons went on to achieve mass production of elevators. Various other elevator designs appeared on the landscape, including screw-driven and rope-geared, hydraulic models. Later in the 1800s, with the advent of electricity, the electric motor was integrated into elevator technology by German inventor Werner von Siemens. With the motor mounted at the bottom of the cab, this design employed a gearing scheme to climb shaft walls fitted with racks. In 1887, an electric elevator was developed in using a revolving drum to wind the hoisting rope, but these drums could not practically be made large enough to store the long hoisting ropes that would be required by skyscrapers.

Figure 1 first electric elevator was in 1876

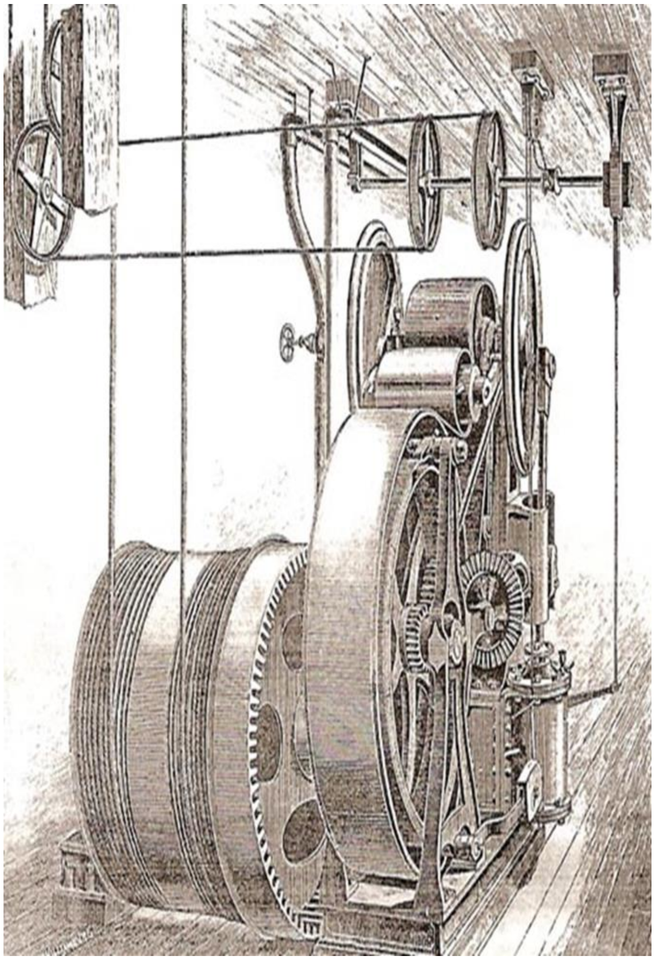
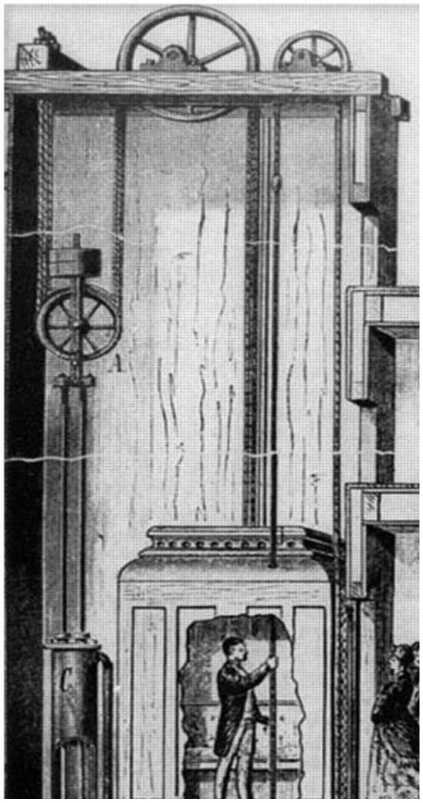
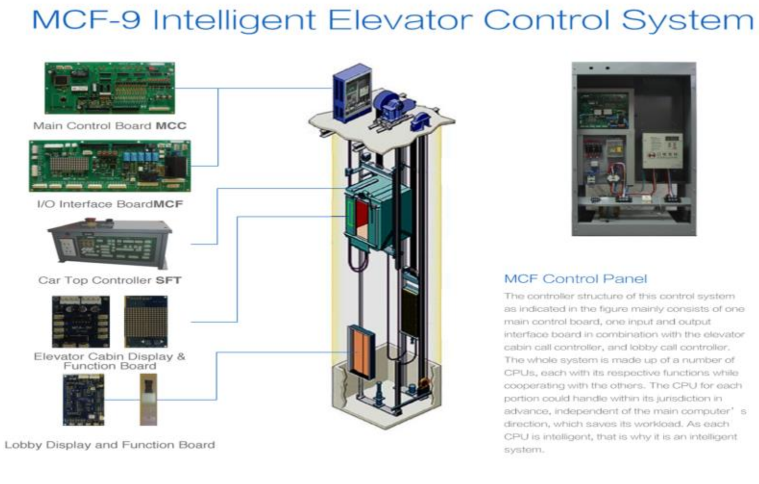


Figure 2 in 1889





MCF-9 Intelligent Elevator Control System

Main Control Board MCC
I/O Interface BoardMCF
Car Top Controller SFT
Elevator Cabin Display & Function Board
Lobby Display and Function Board

MCF Control Panel

The controller structure of this control system as indicated in the figure mainly consists of one main control board, one input and output interface board in combination with the elevator cabin call controller, and lobby call controller. The whole system is made up of a number of CPUs, each with its respective functions while cooperating with the others. The CPU for each portion could handle within its jurisdiction in advance, independent of the main computer' s direction, which saves its workload. As each CPU is intelligent, that is why it is an intelligent system.

Figure 3 in 2021

1.3 Problem Statement

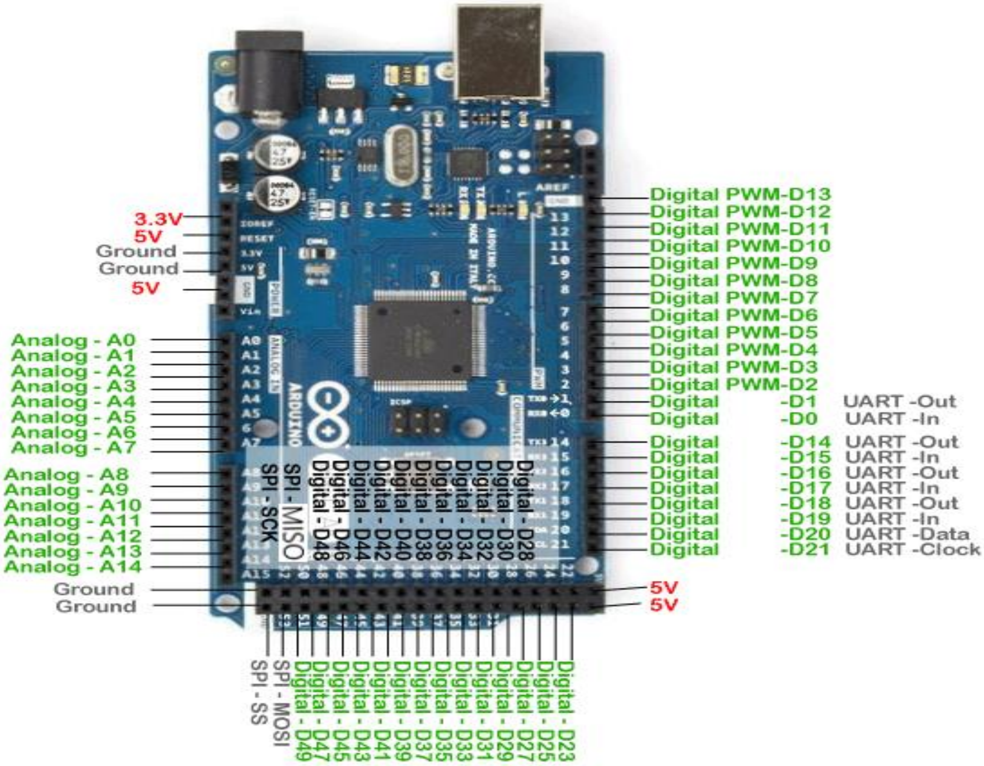
We will compare different elevator control strategies with respect to the average waiting time and total travelling time in an office building with two elevators. These strategies will be compared during two different stages of the day; the up-peak in the morning when passengers arrive from an lobby floor and the down-peak when passengers are leaving the building.

1.4 Arduino Uno

Arduino/Genuine Uno is a microcontroller board based on the ATmega328P (datasheet). It has 14 digital input/output pins (of which 6 can be used as PWM outputs), 6 analog inputs, a 16 MHz quartz crystal, a USB connection, a power jack, an ICSP header and a reset button. It contains everything needed to support the microcontroller; simply connect it to a computer with a USB cable or power it with an AC-to-DC adapter or battery to get started... You can tinker with your UNO without worring too much about doing something wrong, worst-case scenario you can replace the chip for a few dollars and start over again.

"Uno" means one in Italian and was chosen to mark the release of Arduino Software (IDE) 1.0. The Uno board and version 1.0 of Arduino Software (IDE) were the reference versions of Arduino, now evolved to newer releases. The Uno board is the first in a series of USB Arduino boards, and the reference model for the Arduino platform; for an extensive list of current, past or outdated boards see the Arduino index of boards.

Figure 4 ARDUINO



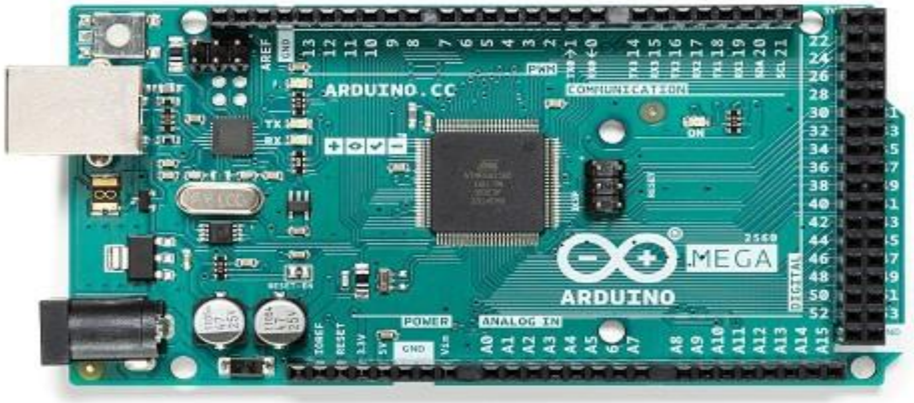


Figure 5 Arduino Mega

1.5 Aims and Objectives

The main objectives for this project are:

- To study an elevator control system.
- To build an elevator model to simulate the actual system by using Arduino Mega.
- To construct a program (software) for the overall system according to the real elevator traffic management algorithm.
- To integrate the hardware and software in order to simulate the functions of a basic elevator system.
- minimizing the cost of making an elevator.
- Using less tools to make an elevator.

1.6 project outlines

After the testing and design procedures were done, the thesis was written in well planned manner. In this thesis, the discussions were mainly about the designed model as well as the control algorithm. Chapter 1 aims to tell the objectives of this project. a brief history of the elevator's development is reviewed. Besides that, the works of completing this project are also introduced.

CHAPTER TWO

2.1 Introduction

An elevator is a platform, either open or enclosed, used for lifting people or freight to upper floors within a building. Elevators are a standard part of any tall commercial or residential building. Manually operated elevators were used for lifting freight in warehouses and manufacturing plants Elevator is a transport device that is very common to us nowadays. We use it every day to move peoples vertically in a high building such as shopping center, working office, hotel and many more. It is a very useful device that moves people to the desired floor in the shortest time. Lift or elevators are generally powered by electric motors. The elevator control system is one of the important aspects in electronics control module in automotive application. In this project the elevator control system is based in Arduino, and the elevator control system is implemented for multi- floor building. Control systems on early elevators required human operators to regulate the speed of the lift and descent.

2.2 Software Model

Modeling is used in many walks of life, going back to early civilizations such as Ancient Egypt, Rome, and Greece, where modeling was used to provide small-scale plans in art and architecture. Modeling is widely used in science and engineering to provide abstractions of a system at some level of precision and detail. The model is then analyzed in order to obtain a better understanding of the system being developed. According to the Object Modeling Group (OMG) “modeling is the designing of software applications before coding “.

2.2.1 Waterfall Model

This model is used only when the requirements are very well known, clear and fixed. Uses clear structure. When compared with other methodologies, Waterfall focuses most on a clear, defined set of steps. Once the product is developed and if any failure occurs then the cost of fixing such issues are very high, because we need to update everything from document till the logic.

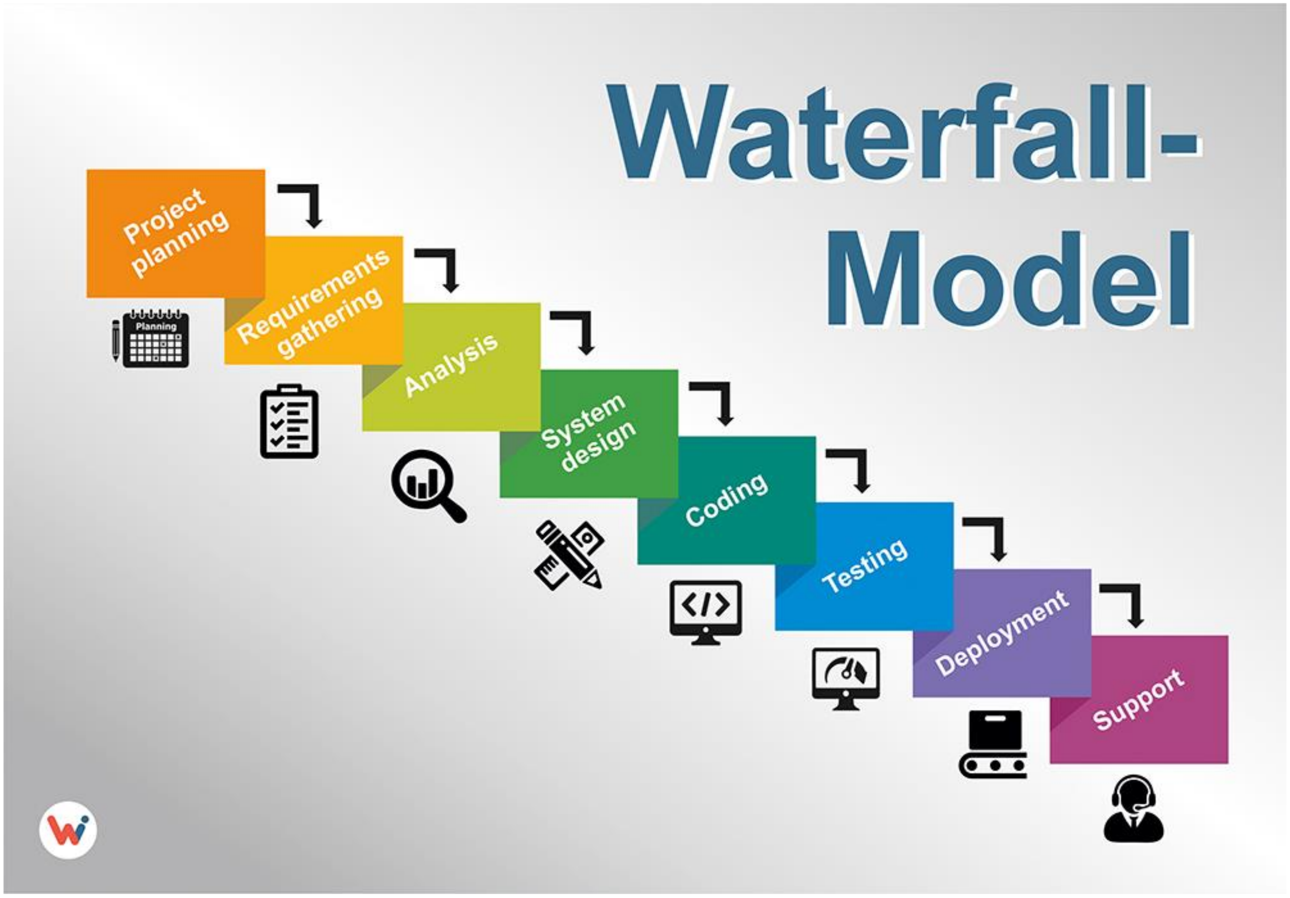


Figure 6 Waterfall Model

2.2.2 Elevator Control advantages

- Before the next phase of development, each phase must be completed
- Elaborate documentation is done at every phase of the software's development cycle.

2.2.3 Disadvantages of Elevator Control

- Error can be fixed only during the phase.
- It is not desirable for complex project where requirement changes frequently.
- Clients valuable feedback cannot be included with ongoing development phase.

2.2.4 Project planning

The elevator control system project is provided with a prebuilt elevator simulator and a PC-based DAQ system. Containing an analog, I/O card and a digital I/O card. The objective is to design a control system for the elevator such that it mimics the operation of a typical elevator. This project is appropriate for a measurements and instrumentation course since it requires both hardware and software design for interfacing the elevator to a PC-based DAQ system for a measurement and control application. Floor sensors on the elevator are used for controlling the position of the car. The response characteristics for this system are sufficiently slow such that control theory for dynamic control applications is not needed.

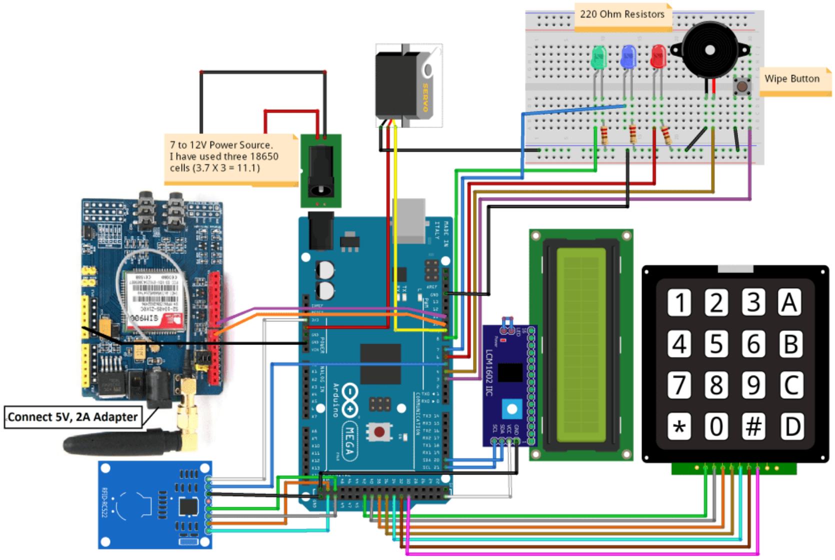


Figure 7 Project planning

2.2.5 Requirements and Analysis

The goal of our development of the elevator control system is to control an existing elevator model. This model is built with Fischer Technik components. Consequently, this model defines part of the requirements on the control system. The elevator has four floors and each floor has its request button and a control light. A sensor is located at every floor. We can use these sensors to locate the current position of the elevator car. The car itself consists of several parts: A door, which can be opened and closed by a motor. Two sensors inform the control system about the door position. A light sensor can detect objects while the door is closing. The elevator car engine moves the car up or down.

Informally, the elevator behavior is defined as follows. If you press the request button at a floor, the request light is switched on. The car moves to this floor within finite time. When the floor is reached, the door opens, and the request lamp is turned off again. The door stays open for some time (10 seconds) to allow passengers to enter or exit the car. After this time, the door closes again. If, while the door closes, the optical sensor is interrupted or the floor’s request button is pressed by a user, the door must open again immediately. After a shorter waiting time (5 seconds) the door closes again.

Before we specify the elevator control system in Autofocus, we split the large starting problem into smaller sub problems. First, we look at a real elevator system, and determine its components. These components will be mapped to the elevator control system specification. The elevator consists of two or more floors, a car and a control system. Each floor has its request button and an elevator door. Every elevator door owns a motor to open and close it. Door sensors are used to determine the door position. Every button has an assigned control light. This light displays a request from the corresponding floor. The car itself consists of several buttons, one car door and a motor to move the car up and down. The component “Elevator Control System” controls the behavior of the whole elevator system.

2.2.6 Software design

Design and test a control system for a service lift operating for four floors. The microcontroller (Arduino MEGA) is used to determine which contactor to provide power in order to activate the lift motor to move the lift up or down by checking the floor the lift is currently at. The control system should be also able to stop the power to the lift when it has reached the required floor. An integral safety factor should be implemented to always close the lift door before moving up or down. The control system is successfully able to determine which contactor needs to be powered in order to move the lift to the desired floor. The system is able to control the safety lift door motor in order to close the door when the lift needs to move and open the door automatically when it has reached the required floor.

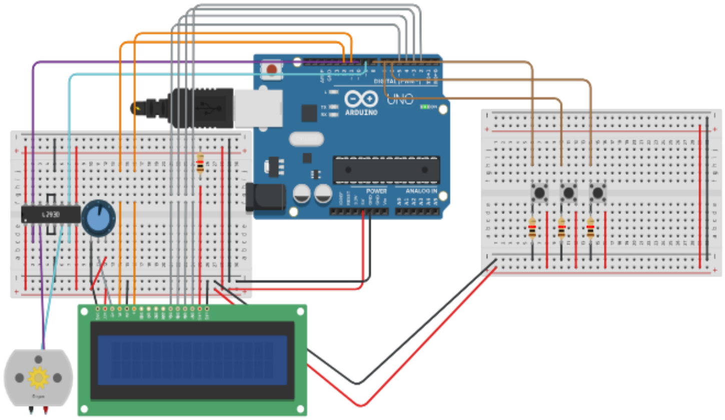


Figure 8 Software design

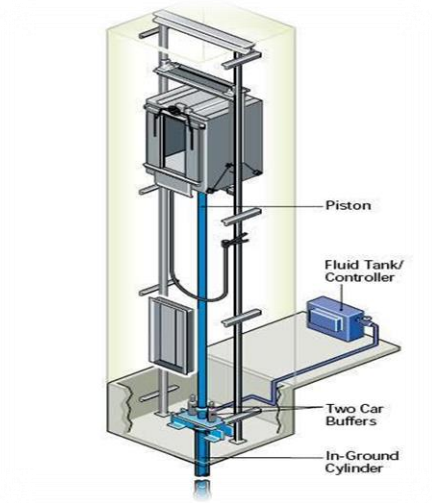
2.2.7 Coding

Coming to the code portion of elevator control system. There are two main functions Floor () and Down Floor (). void main () function is checking if any key is pressed by the user. If any then it calls the Floor () function. Floor () function moves the Lift up wards and after reaching the desired floor it calls the Down Floor () function which moves the lift back to the ground floor. The status of elevator on reaching and crossing each floor is displayed on 16×2 lcd also it can also be visualized on the led bar. Led bar moves upwards when elevator is moving upwards and Led bar moves down wards when elevator is approaching ground floor.

Description of the Functions present in the Code

Void Floor Moving the Elevator Upwards

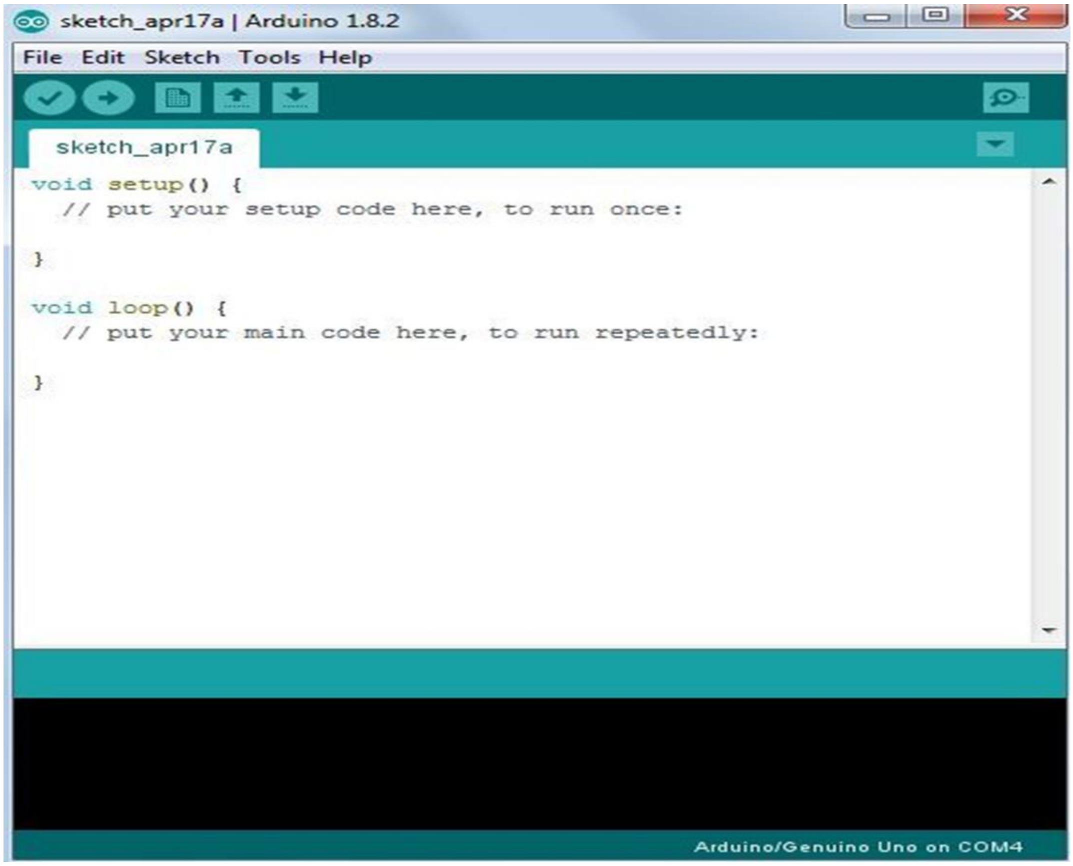
Void Down Floor Moving the Elevator Downwards



Void delay Generating a time delay of some microseconds to be used where necessary

Figure 9 software design

Void Time delay	Generating a time delay of some seconds to be used where necessary
Void lcdcmd	Sending Commands to 16×2 lcd
Void display	Sending Data to be displayed on 16×2 lcd
void clint	Initializing 16×2 lcd


sketch_apr17a | Arduino 1.8.2
File Edit Sketch Tools Help
sketch_apr17a
Arduino/Genuino Uno on COM4

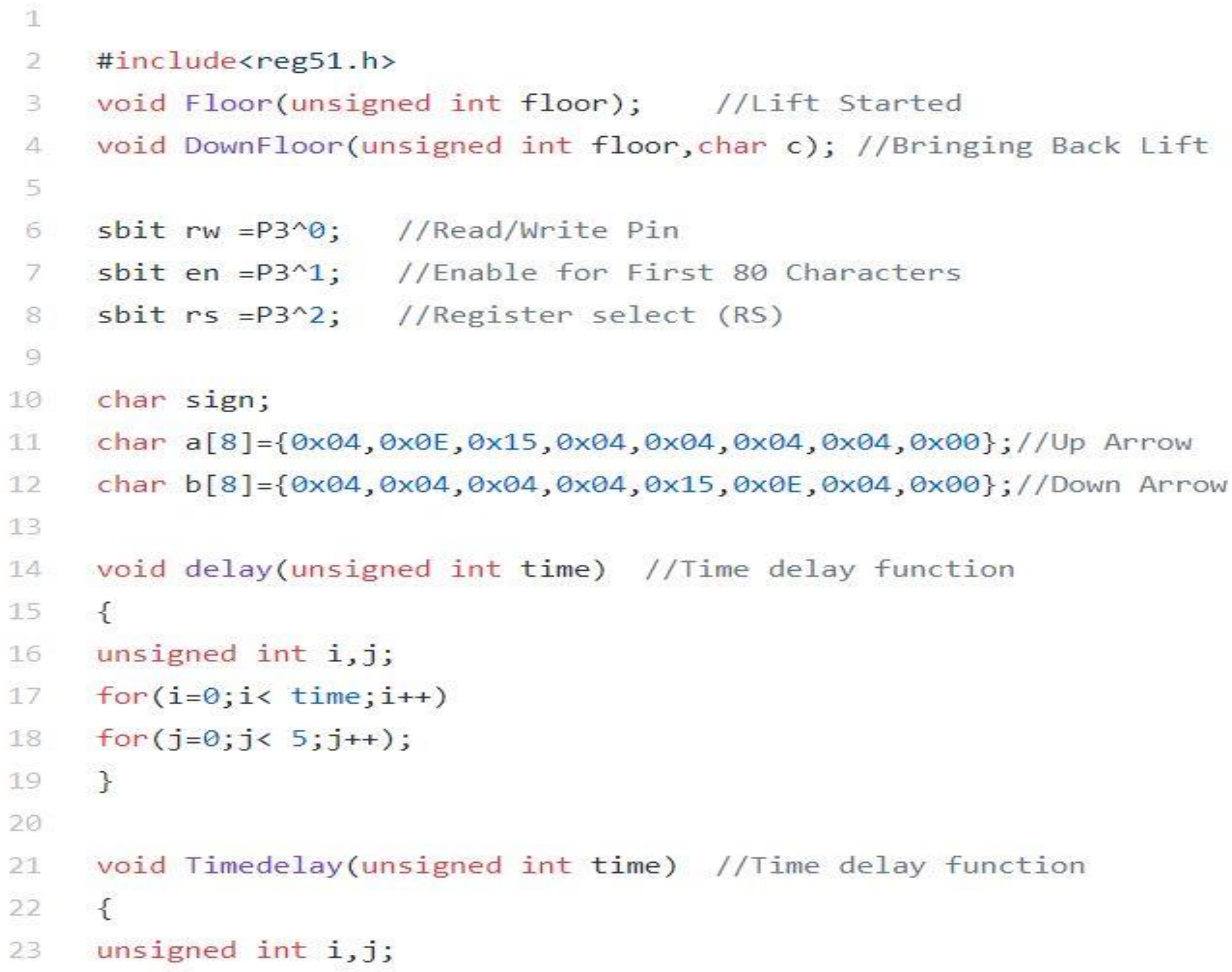
```
void setup() {
  // put your setup code here, to run once:

}

void loop() {
  // put your main code here, to run repeatedly:

}
```

Figure 10 show programs



```

#include<reg51.h>
void Floor(unsigned int floor);    //Lift Started
void DownFloor(unsigned int floor,char c); //Bringing Back Lift

sbit rw =P3^0;   //Read/Write Pin
sbit en =P3^1;   //Enable for First 80 Characters
sbit rs =P3^2;   //Register select (RS)

char sign;
char a[8]={0x04,0x0E,0x15,0x04,0x04,0x04,0x04,0x00};//Up Arrow
char b[8]={0x04,0x04,0x04,0x04,0x15,0x0E,0x04,0x00};//Down Arrow

void delay(unsigned int time)  //Time delay function
{
unsigned int i,j;
for(i=0;i< time;i++)
for(j=0;j< 5;j++);
}

void Timedelay(unsigned int time)  //Time delay function
{
unsigned int i,j;
```

Figure 11 Coding

2.2.8 Testing

For this system, the simulation result is done by using Proteus professional schematic software. After the program is compiled as hex files, the compiled code sent to a Arduino Uno microcontroller to load the microcontrollers with no errors. The program has successful complication and then the circuit is tested using the proteus software This control system is completely tested and can be used in real time. Proteus is very handy software and is used for circuit and PCB designing. It is also used for circuit and programming testing. Note that the same Arduino code used in Proteus is used in the experiment.

2.2.9 Software Deployment

software deployment is the process of delivering completed software to the client who ordered it or rolling out the software to consumers. Software deployment should only take place after thorough testing to ensure that all the flaws and bugs have been identified and fixed.

2.2.10 Operator (Support)

Finally, the application is ready for deployment to a live environment. The operations stage entails not just the deployment of the application, but also subsequent support and maintenance that may be required to keep it functional and up-to-date.