LEEFT3: Lift (Elevator) Environment Forged Tenderly by the Team of Timisoara

Final Report

Victor Tomescu
Mihai Zilahi – Stoiculescu
Mihai Martalogu
Zoltan Lorincz

Hard&Soft Contest
Suceava 2005

20.05.2005

Abstract

The projects developed by the Team of Timisoara at the 2005 Hard & Soft Contest Suceava, consists of three different approaches: a) a hardware implementation of an elevator controller, containing a complete elevator operating algorithm which processes the necessary input sensor data, schedules the elevator's movements and generates the necessary output data for the actuators, indicators etc; b) a GUI based software simulation of the elevator operating algorithm; and c) a hybrid solution that implements state-of-the-art ideas in the field.

As an additional feature, we have also built a working scaled model of a real life elevator with four floors.

This document describes in some detail and discusses the solutions achieved during our contest work.

Contents

1 Introduction

This project, which is the object of the "Hard & Soft" International Contest Suceava 2005, has 3 main goals:

- To implement the controller of an elevator using a "pure" hardware solution. The control algorithm will be implemented on the PIC microcontroller, the drive motor, the cage and floor buttons will be also implemented in hardware. However some actuators and indicators signals will be interpreted and displayed on the computer display.
- To implement a GUI driven simulation of the system described above. This will be a "pure" software solution that will use an environment created in Visual C++ 6.0 in order to simulate the movement of the elevator and the algorithm being used.
- To implement a third solution for the elevator system. This solution will implement some new characteristics like multiple lift shafts controlled by a central dispatcher. It will be a hybrid solution consisting of simulating two elevators: one will be a GUI driven simulation, the other will be implemented in hardware.

Currently, our project development achievements can be summarized as follows:

- Implementation of the algorithm for the first project in PIC microcontroller assembly language;
- Implementation of the algorithm for the second project in Visual C++;
- Implementation of the GUI component of the second project in Visual C++;
- Controller board for Project 1 – Hardware (PIC microcontroller, keypad, pushbuttons, H – Bridge for the , Motor [1, 2, 7]);
- Demo application for Project 2 – Software;
- Working scaled model of an elevator system;
- Implementation of the algorithm for the third project (the hybrid solution);
- Dispatcher application of the third project (the hybrid solution);

Chapter two, Details and Implementation Details, contains the details of implementation for all the three projects.

Explanations about the hardware project, some details about the PIC board, how is the DC motor controlled and the specific algorithm being used are shown in sub-chapter 2.1.

Sub-chapter 2.2 contains details about the software project, differences from the first project.

The details about the third project: essential differences from the first two projects, the new algorithm being used and the software that controls the whole system are presented in sub-chapter 2.3.

Sub-chapter 2.4 presents the working scaled model of an elevator: how the DC motor is controlled, how the lift cage is stabilized.

The conclusions of the projects and the "would-be" future development of the projects are discussed in Chapter Three.

Some appendices containing algorithm diagrams, circuit schematics are included at the end of this document.

2 Design and Implementation Details

2.1 Project 1 – Hardware

This project implements a "pure" hardware solution.

2.1.1 Hardware

This solution implemented only one lift shaft. The shaft has 4 floors: ground floor and three others. Each floor has:

- Two buttons used for calling the elevator for going up or down;
- Sensors used to detect the presence of the cage;

The cage of the elevator contains a 7-segment display showing the current floor number. The cage buttons are implemented using a 12 contacts keypad. A system will be implemented for the detection of the presence of the passenger inside the cage.

There is a DC motor that is controlled through the PWM capabilities of the PIC microcontroller. Also there is a H – Bridge implemented to control the DC motor [1, 2, 7].

The majority of the outputs are simulated by the PIC microcontroller by sending the desired values on a RS232 interface to PC system. The PC runs an application that has the capacity to interact with the serial port. This application listens on the serial port for any incoming sets of data, retrieve the sent data, parse it and display the results on the GUI.

2.1.2 Algorithm

The algorithm for the elevator controller is embedded in an application that runs on the PIC microcontroller.

The application's first priority is to plan the elevator stops based on the external requests (passengers press the "going up" or "going down" buttons situated on the floors) and the internal requests (passengers press the floor number on the keypad inside the elevator cage). Second, the application is responsible to generate all the necessary signals that would eventually control the elevator actuators, indicators, sensors.

A simple description of the algorithm [3, 4] of the elevator controller would be:

- The passengers press one of the floor buttons on each floor to choose the direction they will go: up or down;
- Given any direction of movement for the elevator (up or down) the elevator will serve the request that is closest to him in the direction if it's movement by stopping at the floor the request was made on;
- The elevator will stop at the next floor to satisfy the internal request (the floor number press by the passengers once inside the lift) that is closest to his current position in the direction of its movement.

The initial state of the algorithm is presented in the diagram from Annex A, Figure 2.

The movement algorithm is presented in the diagram from Annex A, Figure 1.

All the application for the PIC microcontroller has been written in C language, using a compiler from Hi-Tech C [6].

This algorithm although efficient has some minor disadvantages:

- If the passenger initiates a request for an elevator just as the lift cage has passed him (worst case scenario), he will have to wait for the lift to complete its run, as the elevator cannot change its direction as long as long as there are still pending requests in the initial direction.
- Also this algorithm is not protected against the false intention that the passengers might have. If a passenger request the lift to go down, waits for the doors to open, enters the lift but chooses a floor located above his location, that passenger will have to wait for the lift to finish its run serving the current pending requests.

2.1.3 PC Application

This application runs on a PC and listens on the serial port. The PIC microcontroller, while running its application, sends data on the RS232 serial interface. This data is mainly composed of signals for actuators, indicators.

The application listens on the serial port for incoming data. When data comes, it is parsed and interpreted by the application, and displayed on a GUI.

2.2 Project 2 – Software

This project implements a “pure” software solution.

2.2.1 Algorithm

The algorithm used for this “pure” software implementation of an elevator controller is roughly the same as the algorithm described in paragraph 2.1.2, with some additional features [3, 4].

2.2.2 Software

This solution is similar to the solution presented above with respect to the algorithms used, but it is a GUI driven simulation. This means that all the actuators, indicators, buttons, sensors, 4 floors and the cage will be simulated inside an environment developed in Visual C++ 6.0.

An additional feature of the current solution is the vocal announcement system for the passengers inside the cage.

This software solution is developed in Visual C++ 6.0 and the GUI is designed in order to comply with the way an actual elevator keypad looks like. For this reason non rectangular buttons are used inside dialog based applications that will incorporate the client application and the server application.

2.3 Project 3 – Hybrid

This project implements a hybrid solution: it contains both a hardware system connected to a working scaled model and a software system that simulates an elevator shaft and controls the whole project.

2.3.1 Differences from the other solutions

The differences from the other two solutions are listed below:

- Multiple elevator shafts: this solution implements two elevator shafts instead of one as the other two solutions.
    - First elevator shaft will be implemented in hardware, using parts of the first project;
    - Second elevator shaft will be implemented in software, using parts of the second project;

- The two elevators are moving in tandem. Although they are not actually connected if one elevator moves in a direction, the other one is moving in the opposite direction.
- The floor numbers keypad that was situated on each elevator for the other projects is now situated on every floor. This gives the opportunity for a much better planning of the elevator's movement.
- There is a dispatcher application that has "supreme" control over the elevators both in the case of an emergency and for manual override.

2.3.2 Algorithm

From an algorithmic point of view the two elevators are working in tandem, although they are not linked in any physical way. If one of the elevators moves, the other one will move in the other direction.

This method increases the power consumption of the general system, but is used for a much better planning of the way the elevators satisfy the requests of the passengers.

This method is very efficient in dense populated buildings. The algorithm used for the first two projects had a particular disadvantage: if the request of a passenger was made after the lift had passed him, that passenger was obligated to wait for the lift to end its run in the direction it was moving. Using two elevators, moving in different directions, the situation described above is avoided.

There is also a disadvantage when using the two elevators: if the request for an elevator comes from a floor that both elevators are leaving behind. This disadvantage is somehow eluded by the fact that there are two elevators, and in the worst case, it would take half the time for an elevator to show up then it would using the old algorithm.

This algorithm is very efficient when used with another state-of-the-art idea: the floor number keypads are removed from the interior of the elevator cage and placed on each floor instead of the usual elevator calling buttons (for going up or down).

Although the control keypad has been removed from the elevator, inside the lift cage there still are the safety buttons like: STOP, EMERGINCY.

The passengers arrive at the elevator door and choose the floor they want to go. The dispatcher will then plan their route by retrieving and processing information including:

- the direction specified by the passenger,
- the direction the lifts are moving,
- the closest elevator to the calling passenger.

2.3.3 Hardware

The hardware used for this project is similar to the one described in paragraphs 2.1.1 and 2.4.

2.3.4 Software

The whole system is controlled from a centralized location which is called a dispatcher. This dispatcher receives the messages from the individual controllers of both shafts and, based on a special planning algorithm, sends back the necessary commands.

The dispatcher has also “supreme” powers on deciding how the lift will move. This means that in case of an emergency (a fire, an earthquake) the dispatcher automatically directs both elevators to the emergency floor (ground floor, parking, etc.). In case of such emergencies the supervisor using the dispatcher can manually override any decisions made by the application and direct any or both lifts to the desired location.

2.4 Working scaled model

A scale model of an elevator is being built using a DC motor. The motor is controlled by the hardware board according to the commands received from the keypad and the floor buttons, based on a planning algorithm developed for the PIC microcontroller.

The DC motor is controlled by the PIC microcontroller using just 2 lines: one for speed and one for direction. In order to do this, a simple H – Bridge [1, 2, 7] has been implemented in order to isolate the microcontroller from the large currents that are necessary to the DC motor when working under payload.

The schematics of the H-Bridge controlling the DC motor that was used are presented in Annex B, Figure 1.

The DC motor has at two speeds: half-speed and full-speed. This feature was introduced to make the passengers of the lift car more comfortable and to avoid various unpleasant accelerations. This feature is implemented by using the PWM capabilities of the PIC microcontroller.

Although the scale model has only 4 floors the algorithm used to implement the controller of the elevator has been designed to work with an unbounded number of floors.

As a safety feature, sensors have been added to all the four floors to detect the presence of the elevator cage. These sensors consist of a pair of infrared photodiode and a phototransistor along with the necessary circuitry [5]. The elevator cage has to go through these sensors and block the light between the photo sensors. The schematics of the photo sensors are presented in Annex B, Figure 1. This feature was introduced in the design in order to now exactly, at every moment, the location of the elevator cage. The simpler solution of placing one photodiode on each floor and the phototransistor on the elevator cage was considered not very safe because if the phototransistor would malfunction, the whole system would be left without the capacity to detect the floors.

The location of the elevator can be read from the sensors immediately as they generate TTL levels and are connected directly to four of the PIC pins.

In order to have the elevator be more energy efficient, there is a counterbalance connected to the lift cage across the shaft of the DC motor. The weight of the counterbalance is half the weight of the maximum payload in addition to the weight of the elevator cage [3, 4]. This situation provides the DC motor with the opportunity to lift or descend only half the weight of the maximum payload, and not the weight of the filled elevator cage.

Guide shafts have been implemented on the elevator shaft to keep the elevator cage from spinning and to keep the small light obturator in line with the photo sensors.

When powered with more the 3V the DC motor rotates very quickly causing the elevator cage to go out of control. This disadvantage is avoided by using double mechanical reduction of speed. The small wheel connected to the drive shaft of the DC motor is, in turn, connected to a larger wheel by the means of a drive belt. The cord that lifts the elevator cage passes over the shaft of the large wheel. This difference in wheel sizes and rotating radiuses assures that the speed of the DC motor is decreased enough to lift the elevator cage in safe condition.

3 Discussion and Conclusions

The hybrid solution could be developed further, if more time was available. Some more evolved algorithms could be added to the project in order to make the planning of the elevator movements more efficient.

Such algorithms are Artificial Intelligence algorithms like: genetic algorithms and neural networks.

These A.I. algorithms are trained over time to make the planning more efficient taking into consideration different parameters like: traffic, time and date, emergency scenarios etc.

In the case of the third project, the dispatcher application could also be improved by adding database support. The database would be used to store information about the elevators, the traffic etc. This information, stored over a longer period of time would be used to train the A.I. algorithms. That information would also be used to generate accurate statistics that would help predict rush hours, slow hours and heavy traffic.

Using the statistics generated from the database, the system would be able to automatically change its operating mode according to the type and quality of the traffic.

One of the features that would be implemented on our system is two sets of doors: one for the elevator cage, one for the floor. The lift stops only when the two sets of doors are perfectly aligned and would start only if both the two sets of doors are closed.

A final improvement of the projects would be the implementation of some more safety features like a protection system for the passengers in case there is a power outage. This system would automatically activate the breaks on the elevator cage if it senses any power loss from the motor. A backup system sends alarms to the dispatcher location.

Annex A – Algorithm Diagrams

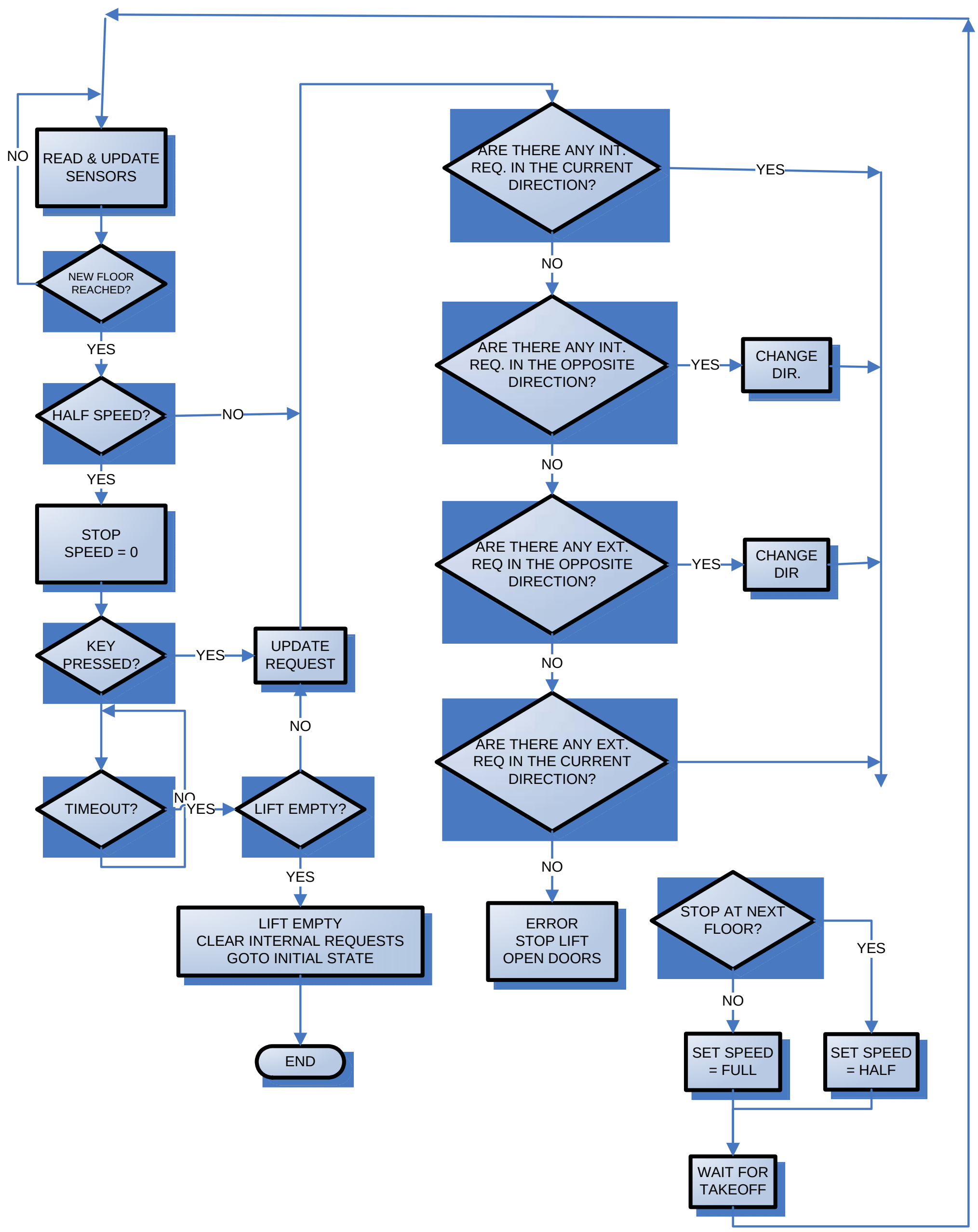

Figure 1: Algorithm for the moving elevator

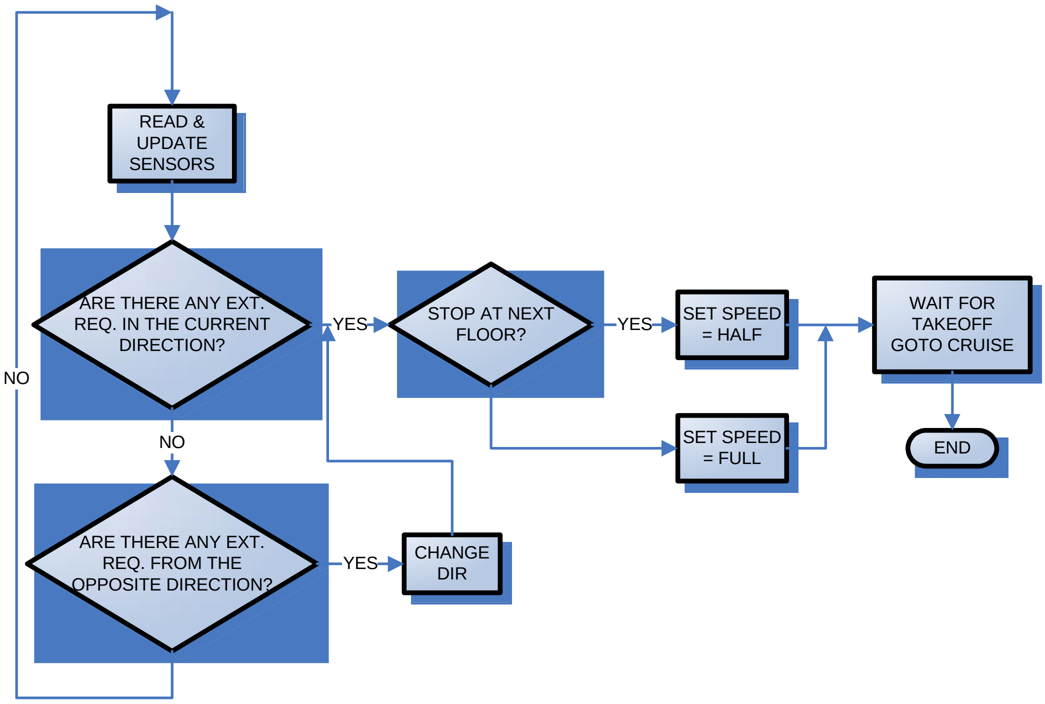

Figure 2: Initial State of the Elevator

Annex B - Circuit Schematics

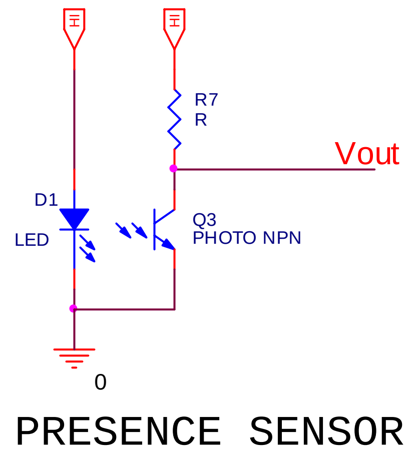

Figure 3: Presence sensor schematics

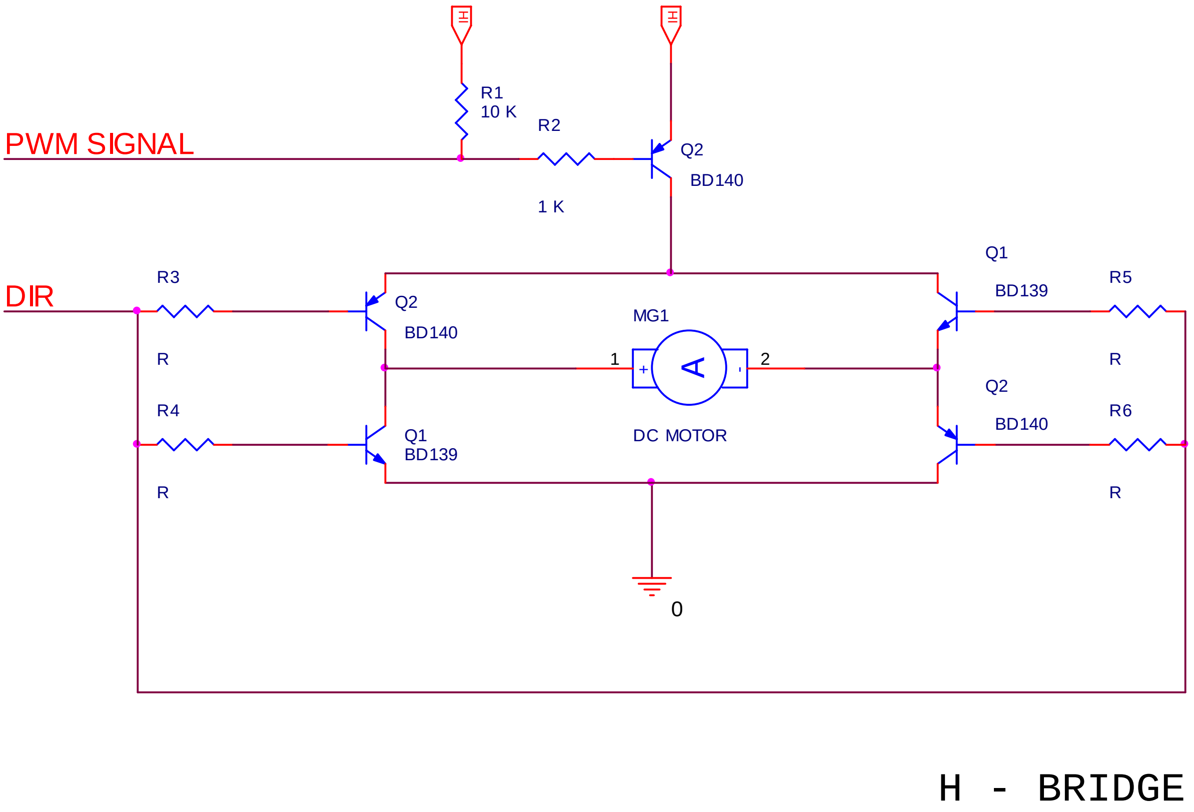

Figure 4: H-Bridge Schematics

References

[1] Jim Brown, "Brief H-Bridge Theory of Operation", April 1998, online: http://www.dprg.org/tutorials/1998-04a/

[2] Scott Edwards, "Electronic Control for DC Motors Using Discrete Bridge Circuits" , January 1997, online: http://www.parallax.com/dl/docs/cols/nv/vol1/col/nv23.pdf

[3] Dr. Albert T.P. So, S.K. Liu, "An Overall Review of Advanced Elevator Technologies", June 1996, online: http://www.elevator-world.com/magazine/archive01/9606-001.htm

[4] LockerGnome Encyclopedia, "Elevator. Everything you wanted to know about Elevator but had no clue how to find it". online http://encyclopedia.lockergnome.com/s/b/Elevator

[5] Dan Stixrud, 'Dan's "Canned Mouse"', sections: "Optical Xmtrs", "Optical detectors", "Schematic" online : http://www.amsky.com/atm/accessories/mouse/mouse.html

[6] Microchip, Inc., "C sample code for PIC micros and Hi-Tech C", online: http://www.microchipc.com/sourcecode/

[7] Circuit Exchange International, "DC Motor Control Circuit" online: http://www.mitedu.freeserve.co.uk/Circuits/Switching/dcmotor.html