

**LEGO elevator controlled by Arduino UNO**

Ambrus Bence Tibor, D1O4NV

Automatic manufacturing systems project I.

University of Óbuda,

Kandó Kálmán Faculty of Electrical Engineering

04/04/2016

Preface:

This project is a subject in my 6th semester as an electrical engineering student. The basic concept is to design, build and control some kind of electric device. It was hard to make a choice but finally I decided to make a fully functioning mock-up elevator. I find it very interesting yet challenging to create something from scratch by myself.

In this short document I am going to represent the specification of my project based on given points of view.

Purpose, introduction:

When I did my first year at university I was told by other students that by the time I finish my studies I will be able to program the control of a lift. So I think this is the main reason I chose the elevator as my project.

The device I am going to build will be made of LEGO bricks. I chose this because it is easy to build anything from LEGO and I do not have access to DIY tools or garage and I can do it on my desk as well.

As for the programming part I have ordered an Arduino UNO from the internet because it can be easily used for controlling circuits and is widely supported by the community. I have wanted to start learning Arduino for a few months now, so this is a great „apropos” to do it.

Requirements:

By the end of the semester I would like my lift to be called from all the levels and controlled with the externally connected inner panel. It should be operated with small delay and within +-2mm accuracy.

Features:

I would like to build a 3 or 4 storey tall elevator with a cabin connected to a rope from the top. The rope will be connected to a stepper motor through a pulley, this will be responsible for the vertical moving. On each level there will be a pushbutton that can call the lift. There will be a panel connected to the cabin with buttons to each levels. Later (in Project II maybe) I would like to add more functions, buttons, sensors.

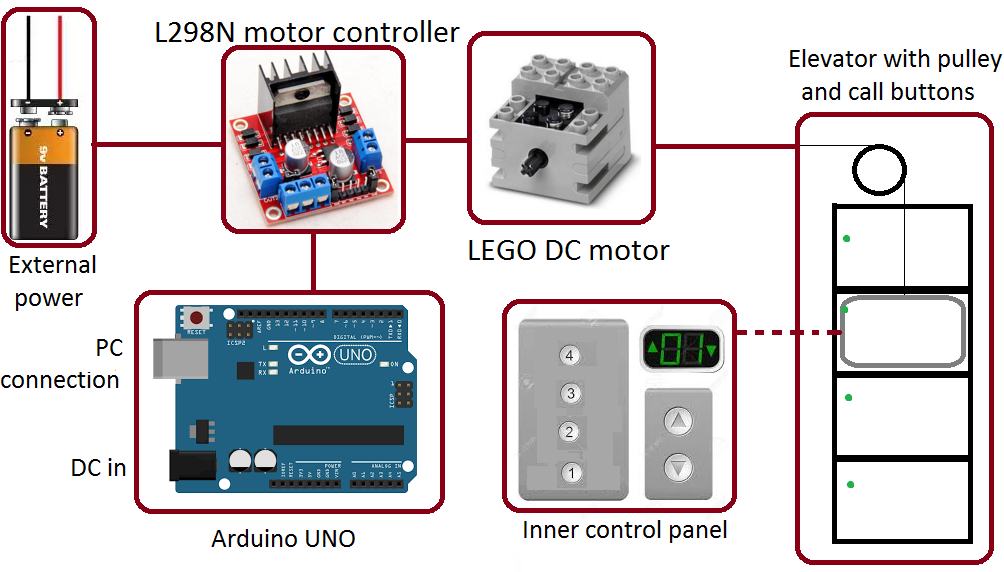
Input:

The input of the lift will be the pushbuttons on each levels and a panel responsible for the inner control.

Output:

On a 7 segment display the elevator will tell the user on which level it currently is. I will ask for response on the PC screen about the stepper motor’s actual status.

Block diagram



Implementation description:

The heart of the device is an **Arduino UNO Rev 3** microcontroller. Since it is open-source, I bought a Chinese copy for a few dollars but it has the same functionality.

Instead of a stepper motor which was the original idea I will use a LEGO DC motor. It is nice because it will fit other LEGO components thus I got it from a

The motor will be run by a circuit called **L298N** which is made for controlling DC motors.  
The motor needs about 6-12V of voltage so I’ll connect an **external power source**, e.g. a 9V battery.

The **inner control panel** will be a circuit made by me, and it will be connected to the lift cabin. Probably I’ll use a simple breadboard for testing. It will have buttons and a 7 segment display.

The **elevator** itself will be built of LEGO bricks, I have not started to design it. It will contain a cabin and a pulley on top for the rope.

Software:

Though Arduino has its own programming language which is built on C, I will stick with to standard C because it gives me more freedom and hopefully better memory usage than built-in libraries.

Gantt chart:

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | 8-Feb-2016 | 15-Feb-2016 | 22-Feb-2016 | 29-Feb-2016 | 7-Mar-2016 | 14-Mar-2016 | 21-Mar-2016 | 28-Mar-2016 | 4-Apr-2016 | 11-Apr-2016 | 18-Apr-2016 | 25-Apr-2016 | 2-May-2016 | 9-May-2016 |
| Week: | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 |
| Idea pitching | x |  |  |  |  |  |  |  |  |  |  |  |  |  |
| First specification |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| Extended specification |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| Building the lift |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| Building the circuits |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| Program to test circuits |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| Lift control program |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 1st demo |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| Adjusting SW-HW |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| Final demo |  |  |  |  |  |  |  |  |  |  |  |  |  |  |

Blue: now. Yellow/red: important dates.

Budget plan:

* I have ordered my Arduino for 3 USD. It’s coming from China, but I’m worried about shipping time... Borrowing one from colleague as rescue until it arrives.
* The L298N motor driver cost 500 HUF, bought from a friend.
* The 9V battery is about 500 HUF, as well.
* The LEGO DC motor I also got for free (free as a beer).
* The LEGO I use is partly mine, partly from friends.
* I already had the breadboard and the jumper cables.
* Buttons, LEDs, resistors, capacitors are not really worth to mention, they are very cheap.

Milestones:

1. Building the elevator from LEGO
2. Connecting everything together
3. Write some program that does the first movements.
4. Requirement #1 verified
5. 1st demo (5th April)
6. Final demo (9th May)

Development history

**04/04/16**

There has been a few major changes during the last weeks.

MOTOR:

* Instead of a stepper motor I got a LEGO Technic 9V DC motor which fits better the overall picture of a LEGO elevator.
* Unfortunately its torque is low and its speed is unnecessarily high. Luckily, it is strong enough to lift the cabin but some gear reduction will be necessary to slow down the movements.
* For going on desired levels I don’t know if the DC motor will be punctual enough and how to control it. A stepper motor might be necessary so that I would be able to count the rotations for moving one level.

MOTOR DRIVER:

* I bought an L298N driver from a friend. Somehow I couldn’t manage to exceed +/- 1,5 V of voltage. I got another one to try but had the same result. (Now I think I messed up that I didn’t connect the ground wire with the Arduino’s GND.)
* So I changed it to an L9110 driver which has smaller size but has no separate PWM input. So I used the PWM outputs of the Arduino.
* It took me a while to figure out that a common ground is necessary to stabilize the signal levels.

EXTERNAL POWER:

* Instead of a 9V battery I use a 9V 600mA power supply of a WiFi router.
* I bought some cable and a female DC connector to extend it so that I didn’t have to cut the connector to use the wire directly.

ARDUINO:

* Unfortunately the UNO I ordered arrived in more than 6 weeks. Until then I used a borrowed one.
* When it finally arrived I had to solder the jumper connectors for myself. One of them doesn’t work properly, maybe it’s my fault, maybe it’s not. Anyway, it’s just the RESET pin, for which I use the button instead of some external restarting.
* I couldn’t connect it to the computer but I only tried with my girlfriend’s Macbook, maybe my PC will recognize and be able to program it.

ELEVATOR:

* The current status is I think acceptable, a few minor changes will be made but the base is very promising.
* There was a lack of Technic components so I was struggling to build the lift. A friend of mine came over with the missing parts and gave me some useful ideas how to he would build it.
* To build the cabin I use damil and a pulley. I used thread but it tore all the time. The motor itself is on the ground floor.

CONTROL PANEL:

* Currently it has only UP and DOWN buttons.
* The elevator is operating as long as one pushes the buttons.
* Going up uses full speed of the motor.
* Going down uses reduced speed with PWM but it is still very fast. (Gearing is needed!)

I have to admit that though I did work on the lift I didn’t spend enough time on it. It is related to other things to study, work, illness, lack of necessary components to continue. But after all, there is no excuse, I was bad at time management. So I’ll have to change my attitude because I do not wish to fail the subject and I actually am interested in this project.

What is very important to implement:

* Gear reduction
* Separating levels
* Calling buttons on each levels
* Sensors to sense actual level
* Dedicated buttons for each levels
* 7 segment display to show actual level
* Lots of codes...