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 document I am going to represent the specification and final results through development history 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 have built is 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. It is also relatively portable.

As for the programming part I ordered an Arduino UNO from the internet because it can be easily used for controlling circuits and is widely supported by the community. I had wanted to start learning Arduino for a few months before, so this was a great „apropos" to do it.By the way I like the concept of open source, and Arduino is open source as well.

Requirements:

In this section you can read the requirements I defined for myself at the beginning of the semester:
„By the end of the semester I would like my lift:

- to be called from all the levels and controlled with the externally connected panel,
- it should be operated with small delay and
- within +-2mm accuracy."

Features:

Again, first I write my original ide:
„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."

The actual device I have built is 4-storey-tall, has a cabin connected to the motor through 2 pulleys with fishing line. It is controlled by a stepper motor, which is actually not the fastest but the accuracy of it is very good. I did not put pushbuttons on the levels, but there are 4 calling buttons for each levels and 2 more buttons for going 1 level up and 1 level down. I could connect these buttons on the lift itself but it would need some more cabling and the functioning of it would still be the same, so I thought it is quite useless.

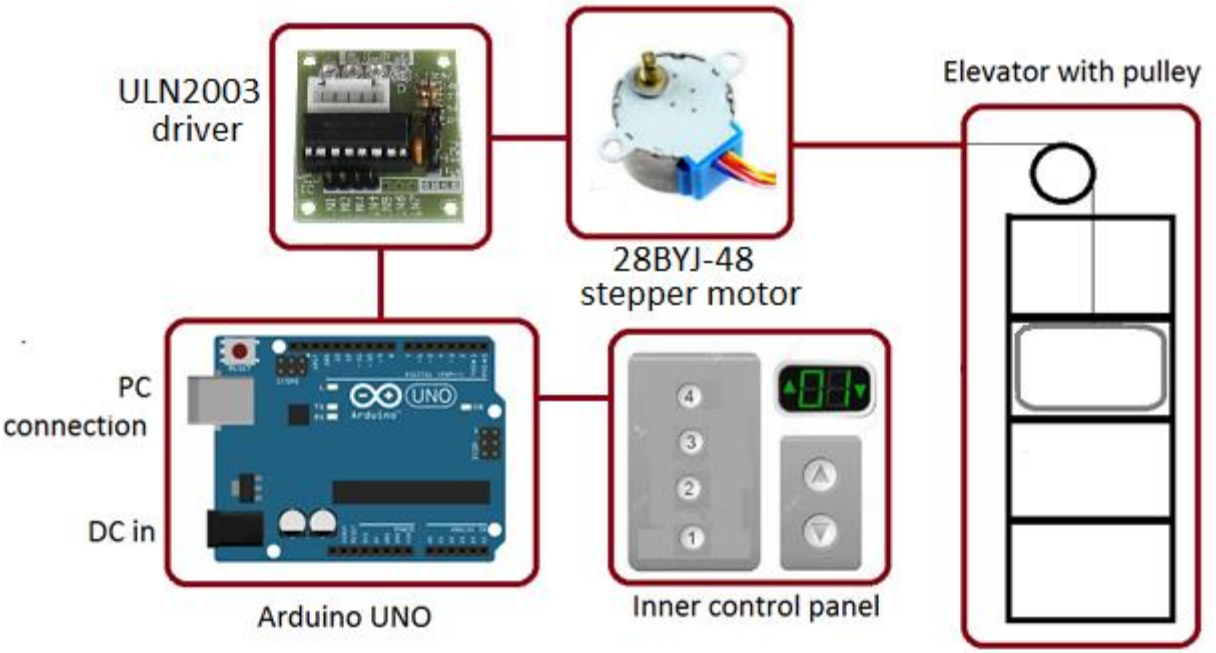
Input:

The input of the lift is a panel responsible for the control with six pushbuttons.

Output:

On a 7 segment display the elevator tells the user on which level it currently is. It can ask for response on the PC screen about the current level, as well.

Block diagram



Implementation description:

The heart of the device is an **Arduino Duemilanove** microcontroller. It is practically the same as the well-known Arduino UNO. 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 in the beginning I changed a LEGO DC motor. It was nice because it did fit other LEGO components but I wasn't able to control it precisely. The motor was

supposed to be run by a circuit called L298N which is made for controlling DC motors. But after all I changed back to using stepper motor, especially **28BYJ-48**. It is much slower but has about 10x torque compared to the LEGO motor. It is controlled with **ULN2003** driver circuit.

The **inner control panel** is be a circuit made by me, and it is connected to the lift cabin. I have used a simple breadboard doing it. It has 6 buttons, 7 segment display with a shift register, 2 LEDs and some shunt resistors.

The **elevator** itself is built of classic LEGO and LEGO Technic bricks. It contains a cabin and two pulleys on top for the rope.

Software:

Though Arduino has its own programming language which is built on C. I like it a lot because it is made for controlling hardware. It allows the user to focus on the actual controlling code instead of defining basic elements.

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:

- Arduino: 3 dollars
- L298N motor driver: 500 HUF.
- LEGO DC motor: got for free (free as a beer).
- LEGO: partly mine, partly from friends.
- 28BYJ-48 + ULN2003: 1000 HUF (from friend)
- Already had: breadboard and jumper cables.
- Shift register: ~110 HUF
- 7 segment display: ~250 HUF
- Buttons, LEDs, resistors are not really worth to mention, they are very cheap.

Milestones:

1. Building the elevator from LEGO DONE
2. Connecting everything together DONE
3. Write some program that does the first movements. DONE
4. Requirement #1 verified DONE
5. 1st demo (5th April) DONE
6. Final demo (9th May) NOT YET DONE

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. First I completely forgot about common ground, so I didn't connect the driver's ground with Arduino's GND. This way I couldn't manage to exceed +/- 1,5 V of voltage.
- Before I figured out what the problem was 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 way too long 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 had ordered arrived in more than 6 weeks. Until then I have used a borrowed one.
- When it finally arrived I had to solder the jumper connectors for myself. Reset pin doesn't work properly, maybe it's my fault, maybe it's not.
- I decided to use the borrowed Duemilanove.

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 about building.
- 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...

05/05/16

Since the last month a lot in my project has changed.

MOTOR + MOTOR DRIVER:

- I was unable to achieve an acceptable control for the LEGO DC motor.
- So I changed back to the original idea and bought a **28BYJ-48** stepper motor. It has slow speed but much higher torque than my DC motor.
- Fortunately it came with a **ULN2003** driver circuit. It is easy to connect and use.
- I found a good library for controlling it (called unistep).
- In the end I'm glad for changing motors though it took a lot of time, but it also made me look after the working principles of them. So I learned a lot from it.

EXTERNAL POWER:

- Since my stepper motor is working on 5 volts it is no nore necessary to connect external power. It makes things easier to assemble and more portable.

ARDUINO:

- I run the software on the Arduino Duemilanove. As I mentioned before it is technically the same as Arduino UNO, only minor differences are there between them but not noticable. This one is borrowed so I have to give it back but then I'll just continue with my UNO.

ELEVATOR:

- I've made quite big changes about the elevator itself:
    - First it got taller and a little bit more stable.
    - Secondly, I solved the pulleys so that the rope is exactly in the middle.

CONTROL PANEL:

- This is much more complex than it was before. Let's look what it has:
    - Call buttons for levels 0, 1, 2 and 3.
    - Go up and go down buttons.
    - A 7 segment display which shows the actual level (with a shift register).
    - 2 LEDs indicating actual direction.
- It is still on a breadboard because I didn't know if I want to put any more functions before, so I didn't see the reason to make it on a PCB.

What is left from this semester:

Handling multiple level calls at the same time. I am very sad about this but I just couldn't fit it in my time. A plus thing would have been an emergency stop.
But let's be optimistic: at least I have things to implement next semester.
I know that when I started this project I was already in late and I was worried that I weren't going to be able to make anything concrete and I would fail the subject.

Future:

I am looking forward to continue this project in next semester, and I've also chosen the elevator as my thesis topic (unless I will find some more interesting topic, which I kind of doubt).

Conclusion:

I think that the beginning of the planning was the hardest thing of all. I made some conceptions and it was hard to choose which way to follow. I sat in my room for long hours trying to build the elevator itself with no success. Moreover, the circuits did not seem to work properly either. But when I finished the lift construction and finalized what I had wanted to implement in the hardware part, things went a bit faster.

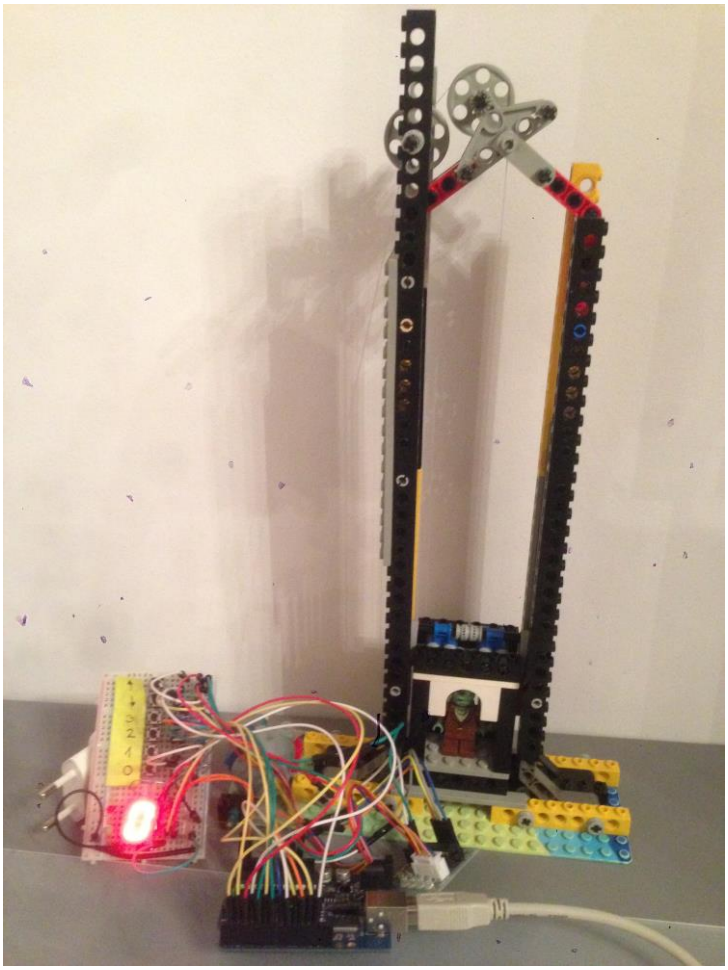
Summary:

I consider Project I. to be a good chance for me to get ready for Project II. and the thesis. I didn't get on with this task as planned but now I have something existing that I "only" have to accomplish. Maybe a good idea would have been to start it in Project 0.
Next semester I can focus on improvements.

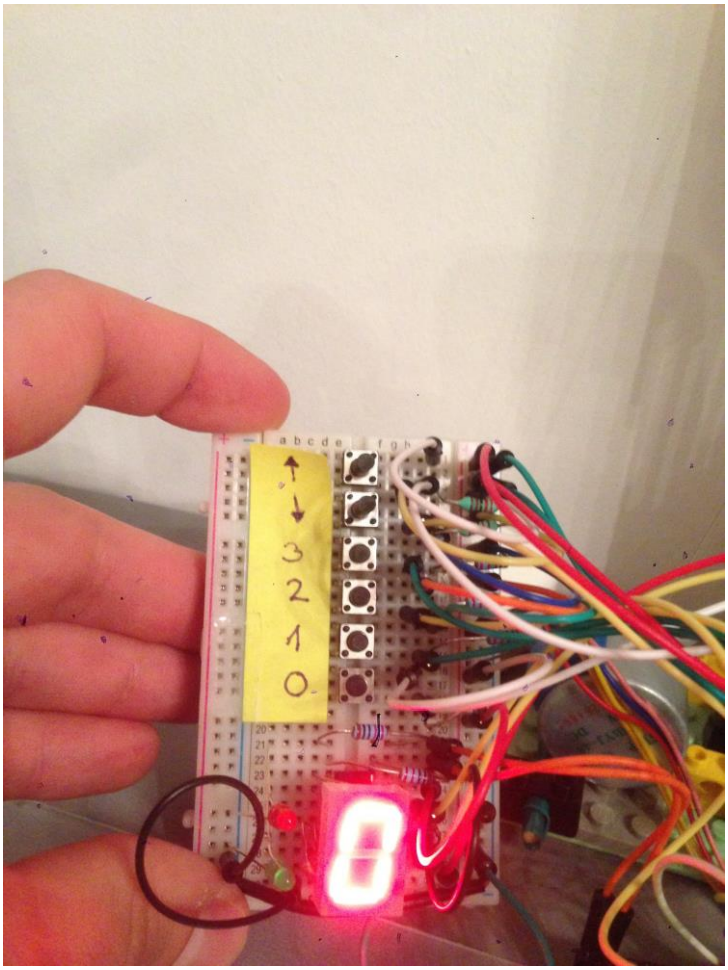
Pictures:

After the summary, last but not least, here are the pictures of the final results:

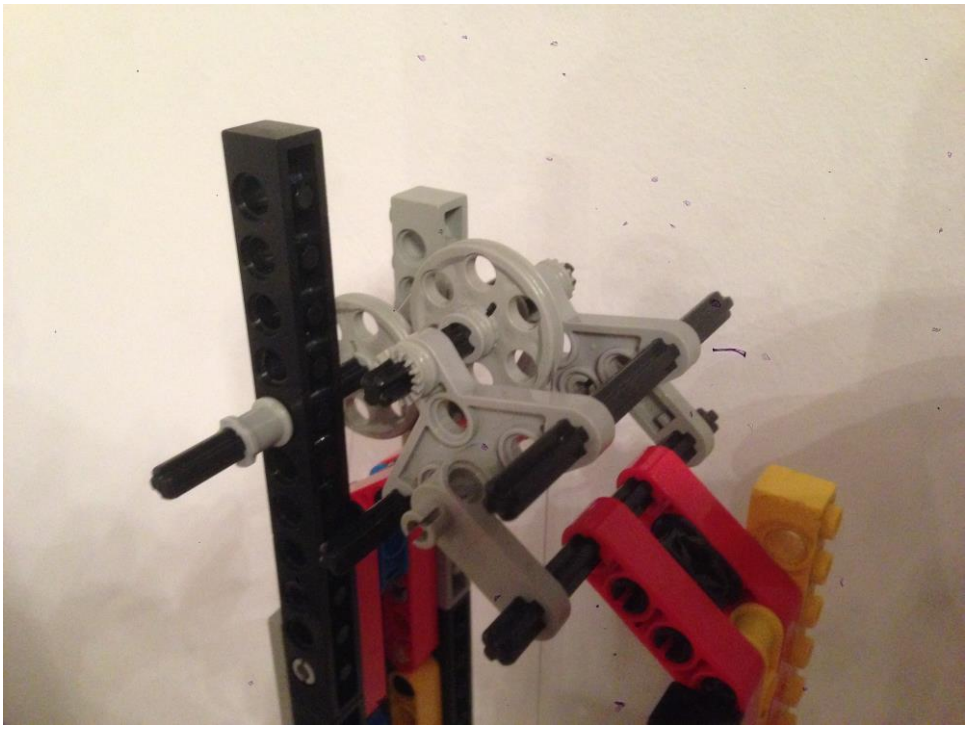
The elevator when started:



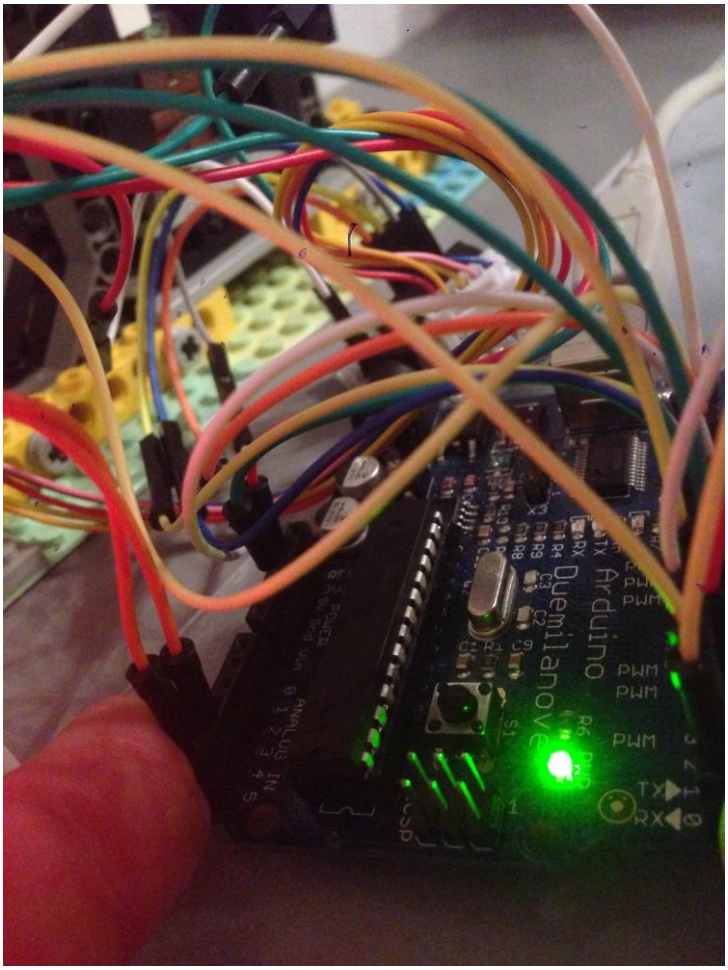
The control panel:



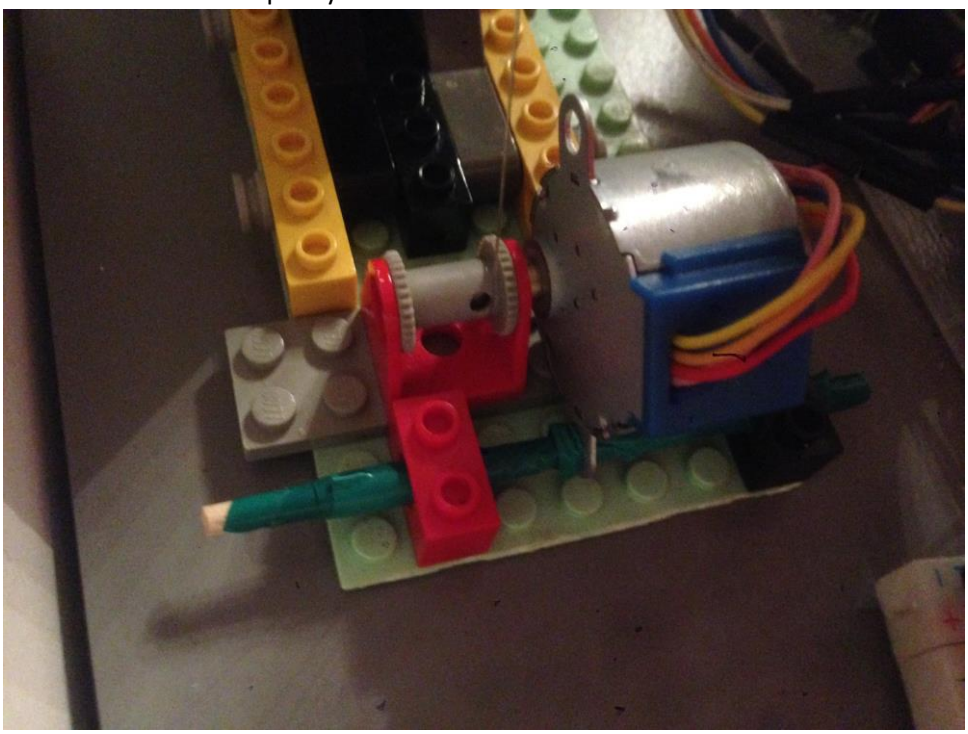
The pulleys on top:



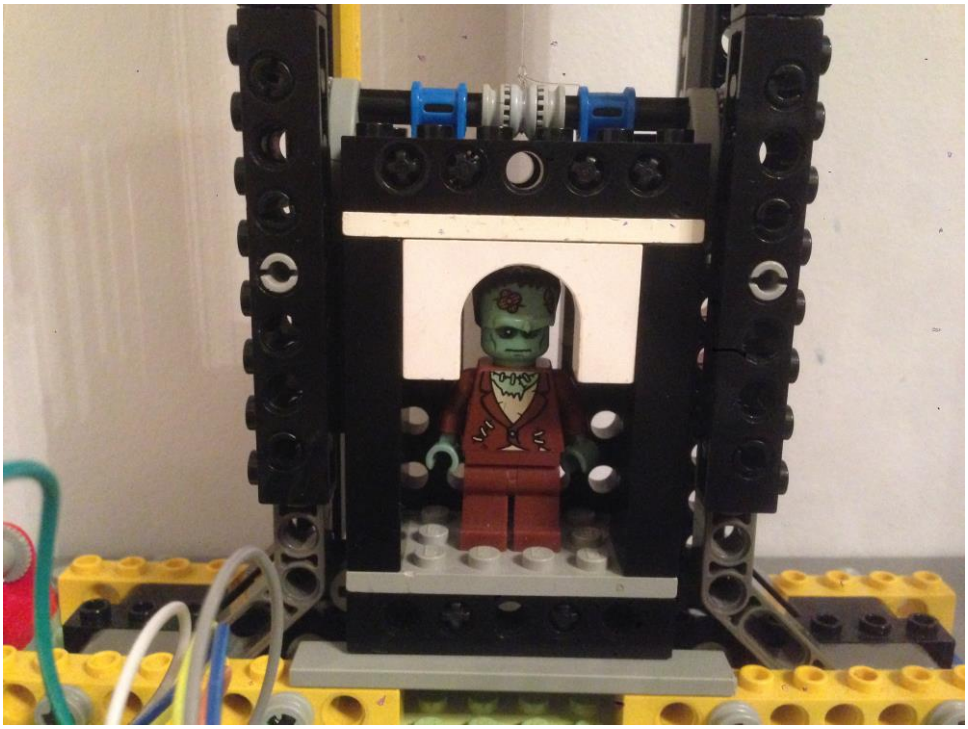
The controller with the jungle of cables:



The motor with LEGO pulley attached:

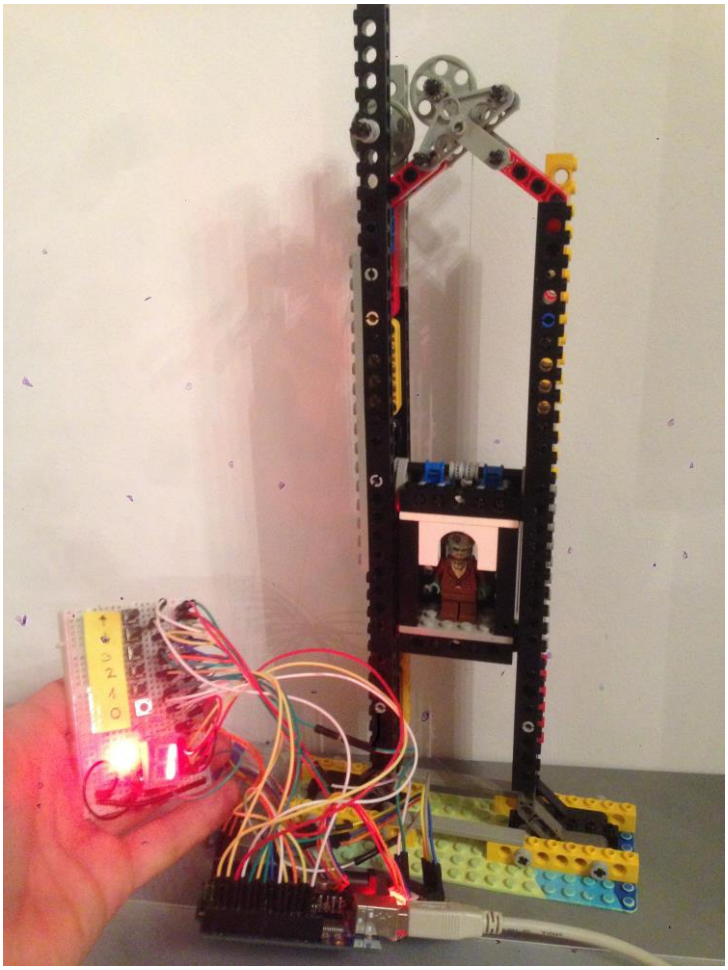


The cabin and its passenger:

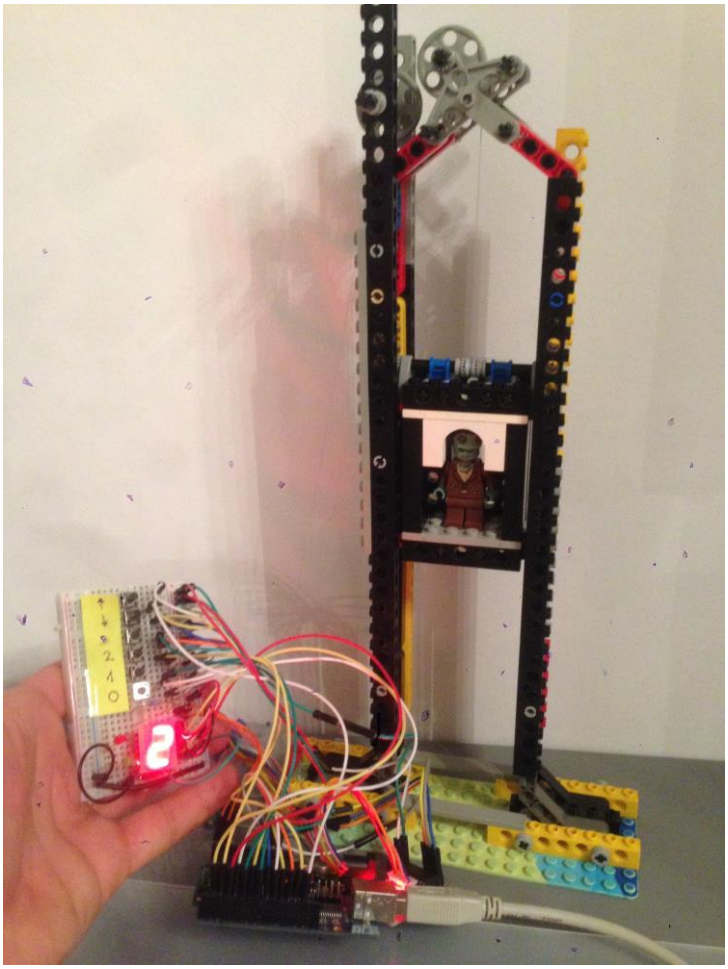


And now let's see it operating:

Moving up (red LED indicates the direction):



Arrival on level 2:



Going down from level 3 (green LED shows direction):