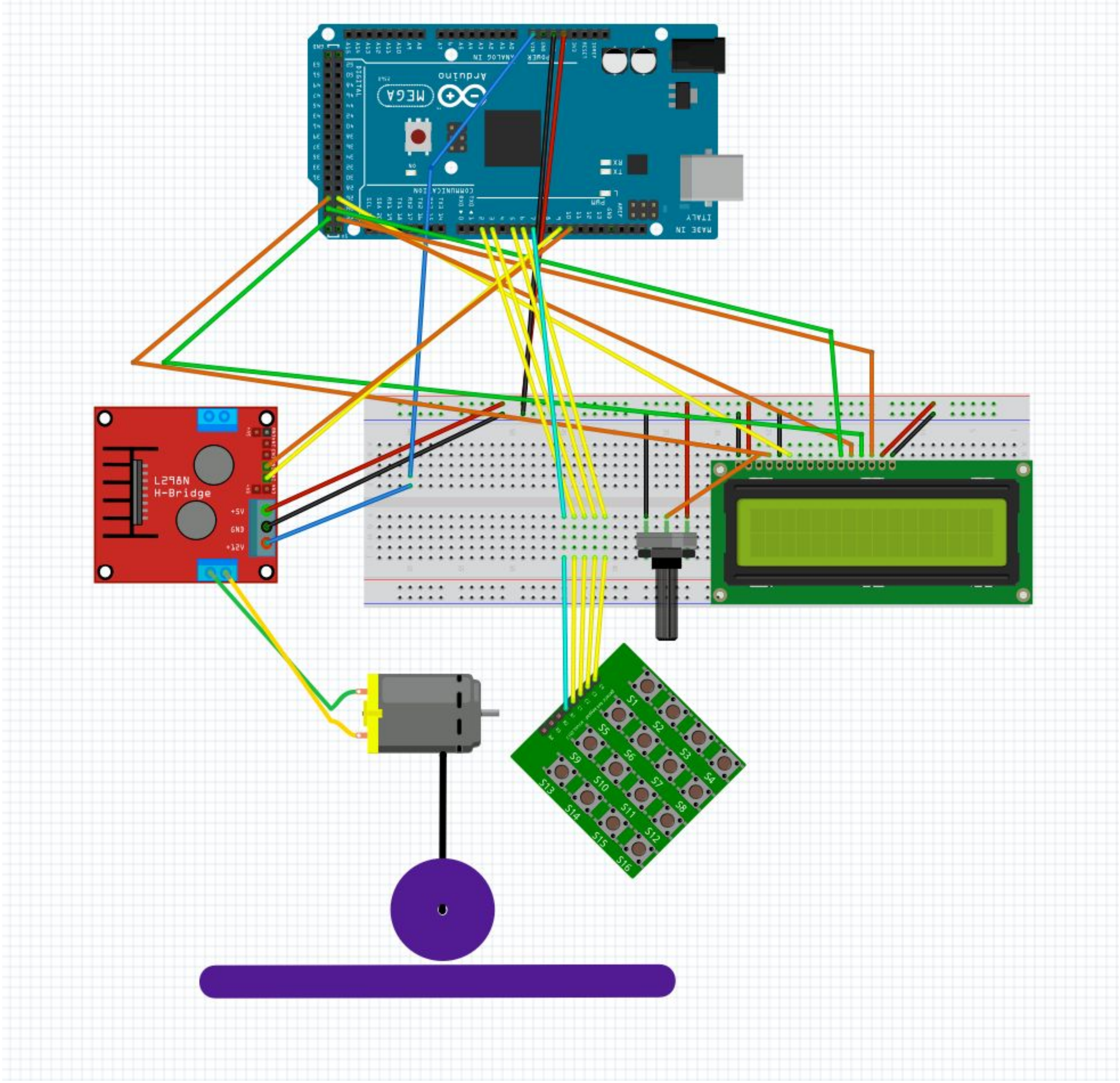
Microcontroller based system components:

1. Arduino Mega2560
2. DC Motor, model: HC02-48
3. H-Bridge, model: L298N
4. LCD 16x2
5. Potentiometer
6. 4x4 Keypad
7. Rack and pinion
8. Breadboard and jumpers

Below is the scheme of the system and here (https://drive.google.com/file/d/15ctDk0xUulHCBYjKSKEy8zE38HjRbYDC) is a short video with system behavior.



I chose Arduino Mega2560 because it has enough pins to avoid conflicts. For example, I used two external interrupts, one internal interrupt and two PWM outputs.

Due to the fact that the motors require a significant current intensity to produce movement, i've connected the output pins of microcontroller to the H-Bridge. To change the rotation rate i've used two PWM outputs. For this model of DC Motor, the output (0 - 255) is same with the value of rotation rate (RPM).

Because the requirement says that the length of the rack and pinion is by my choice, I chose a rack which has the same length as pinion circumference (~10.6cm). I think it was a good choice because it simplifies the equations.

rack_length = 10 cm
pinion_circumference = 2 * pi * pinion_radius = 10 cm
linear_velocity = pinion_circumference * rotation_rate
I chose rotation_rate = 60 rpm and that implies that
linear_velocity = 60 * pinion_circumference / 60s => linear_velocity = 10 cm/s
(all of these above are ideal values, because I didn't take care about decimals)

To control the rotation rate in real time, i would have needed an encoder and an IR module to detect DC Motor rpm. So, if I can't measure rotation rate, I can't display the acceleration. (I could use an acceleration sensor and then I had to make the DC Motor and pinion mobile.)

I used a 4x4 keypad because I needed 4 buttons to set a target value to control the position of rack, 2 buttons to move left/right, 1 button as interrupt to stop the system and select the position, and the last one to start the movement.

The user can see the linear velocity, acceleration, rack and pinion position displayed on LCD. In the first line, using timer one, I displayed speed and acceleration, and in the second one, the rack and pinion position, but in this case, using button interrupts.

Because I heard pretty late about this internship and I haven't seen a deadline for applications, I've accepted the challenge and solve it, but the code is not optimal. :D

I chose to solve it in embedded C, using Arduino IDE. I've splitted the code in 5 parts:
one file.ino with #defines, global variables, setup and loop
display.ino with display functions
interrupts.ino with external interrupts routines
keypad.ino with keypad functions
motor.ino with motor modes and start/stop functions