Firstly, I connected DC motor to Arduino and motor driver as in the *Figure 5*.

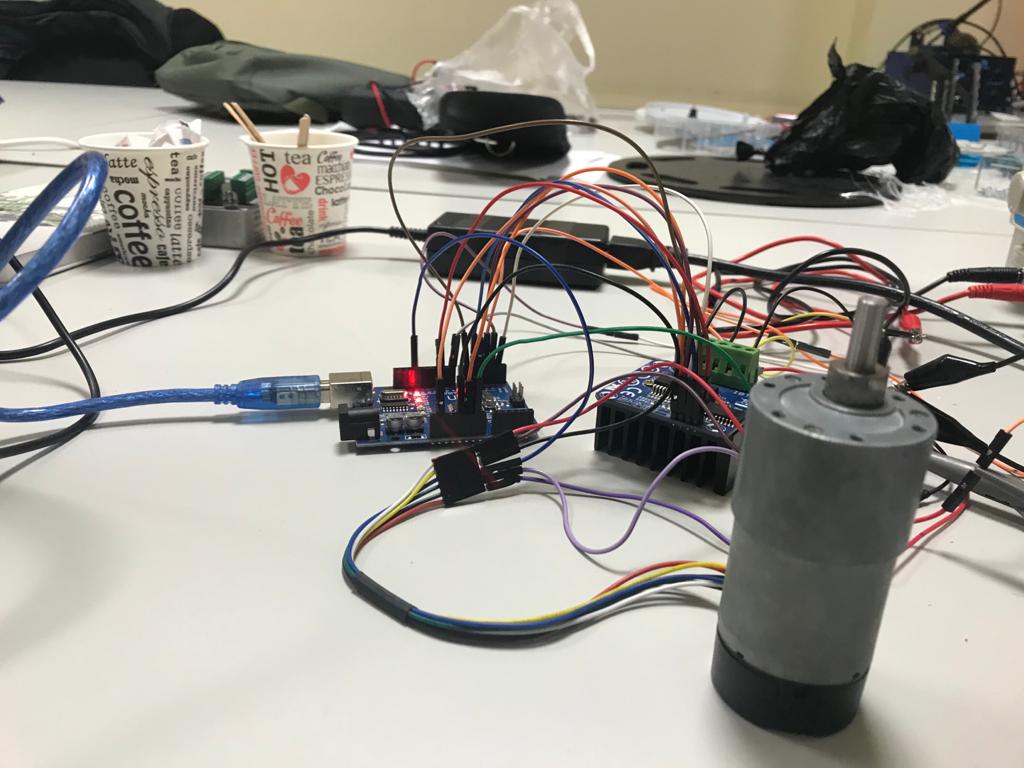


Figure 5

I examined encoder motor output-A and output-B duty cycle, rising-falling edges and clockwise-counter clockwise results on oscilloscope. Encoder gave us 2 output signals (*Figure 6* and *Figure 7*). So, I could analyze easily.

I examined encoder motor output-A and output-B rising edges on oscilloscope when output-A rising, if output-B is equal to 1 it means that motor rotating clockwise, if it is 0 it means that motor counter clockwise. So, I applied “attachInterrupt “method to count Encoder outputs’ rising edges, experiment to 2100 rising or falling edges are equal to 360 degrees. So, we can say that 5.85 rising edges are equal to 1 degree.

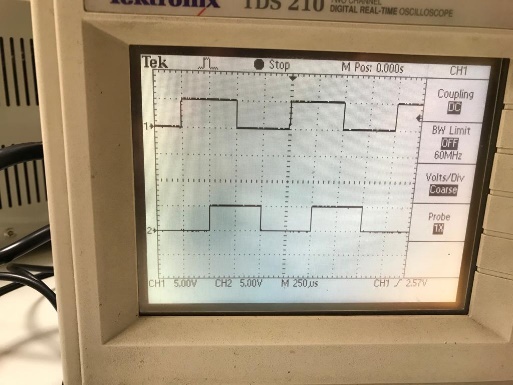


Figure 6

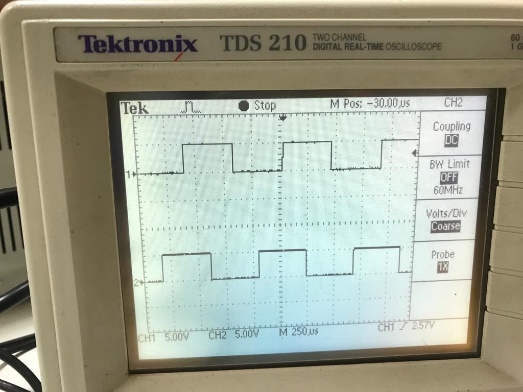


Figure 7

When I control with if() conditions, motor had resonance, it generated vibration. After that, I created PID control algorithm and applied the Ziegler-Nichols method, for PID tuning parameters that are Kp, Ki, and Kd, which is:

1. Set all gains to 0.
2. Increase Kd until the system oscillates.
3. Reduce Kd by a factor of 2-4.
4. Set Kp to about 1% of Kd.
5. Increase Kp until oscillations start.
6. Decrease Kp by a factor of 2-4.
7. Set Ki to about 1% of Kp.
8. Increase Ki until oscillations start.
9. Decrease Ki by a factor of 2-4.

Then, I determined PID parameters on Matlab. First, I analyze time versus position graph and determining what function is on excel at different setpoints as in the *Figure 8*.

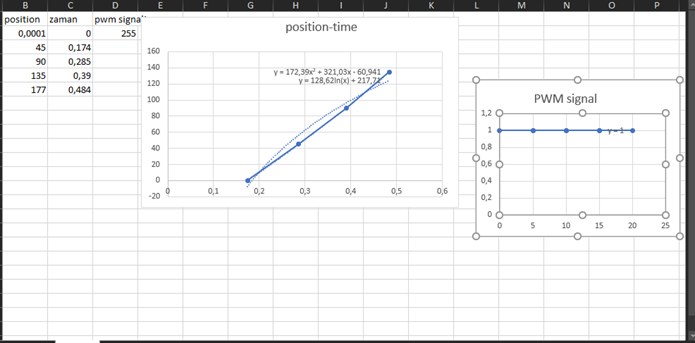


Figure 8

Then, I created our model on Matlab as in the *Figure 9*.

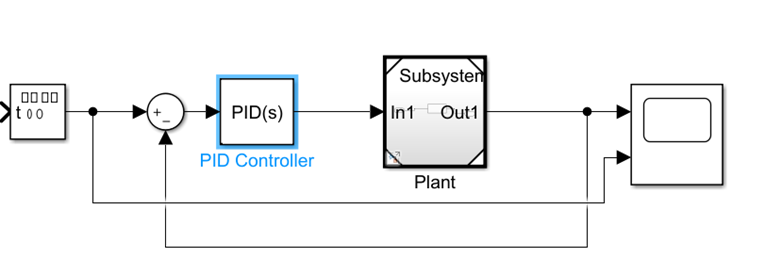


Figure 9

On PID Tuner on Matlab I determine PID tuning parameters as what we want to see like response of the system. Also, I calculated Laplace transform of transfer function and applying values on Matlab to tune PID parameters.

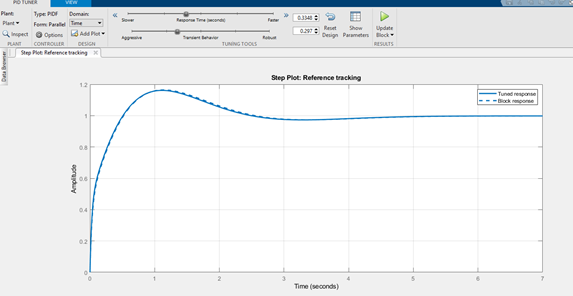


Figure 10

Finally, test results were as expected. Motor didn’t have resonance and vibration (*Figure 11*).

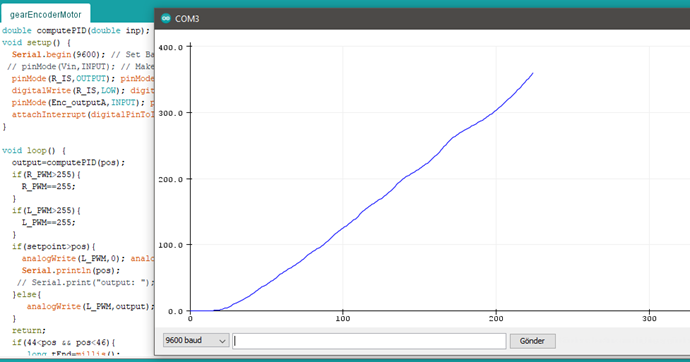


Figure 11