TOBB UNIVERSITY OF ECONOMICS AND TECHNOLOGY DEPARTMENT OF MECHANICAL ENGINEERING

MAK 451 MECHATRONIC INSTRUMENTATION

2019-2020/SUMMER

MINI PROJECT 1

10/06/2019

Hande YILDIRIM 141201047

TABLE OF CONTENTS

1. PROJECT DESCRIPTION

In this project, we will try to move our device using specific values which are defined by user and we have to get trapezoidal velocity profil during this motion. For this project we need some devices which are pc (for write our algorithmic code), Arduino (for use as a processor), bread board (for build our circuit), ETU kit (device that we are trying to move),stepper (source of movement),motor driver (create a connection between arduino and stepper), and limit switch (for bring our device to starting position). We can describe the how this system work with a few basic steps. First of all we are setting starting position of our device using limit switch signals. Then we are take some specific values from user and according to these values arduino sends signals to our stepper . According to these signals our stepper starts rotating different durations and speeds. When we reach to the desired position (our target) we are waiting a few seconds there. Then, stepper rotating opposite side and our device (ETU kit) starts moving to starting position until the limit switch signal is turned off and asks new specific values. This loop continues until power is off.

2. ANALYSES AND CALCULATIONS

These calculations are executed to solve the problem in this project. The aim of the calculations is to control delay time between two microsteps. Circumference of the stepper motor is measured as 40 mm. It is shown in Equation 2.1 and Equation 2.2.

$$\frac{40mm}{200step} = 0,2mm/step \qquad \text{Equation 2.1}$$

$$\frac{0,2mm/step}{16microstep/step} = 0,0125mm/microstep \qquad \text{Equation 2.2}$$

80 microsteps are stepped for 1 mm. It means that if delay time is $\frac{1000000}{80} = 12500 microseconds$, the speed would be 1 mm/s. Thus, for any "V" value, delay time should be 12500/V microseconds. It is shown in Equation 2.3.

$$\#of\ total\ stepnumber = \frac{position}{0.0125} \qquad \text{Equation 2.3}$$

2

$(0,5)*a*\ \ \ \ \ Va$

$$required\ \#\ of\ step\ for\ acceleration = \frac{}{0{,}0125}$$ Equation 2.4

Required # of step acceleration and deceleration are same. When these two values minus from required total step, the steady-state velocity step number is found. It is shown as Equation 2.4.

As is known, when motor accelerate, delay time cannot should be diminish for sent signals. The “Add” value is a constant how much value of delay is decreased by each step. Equation 2.5 defines this.

$$Add = \frac{(200 - 12500 / Speed)}{Required\ \#\ of\ step\ for\ acceleration}$$ Equation 2.5

Thanks to this calculation, when motor steps as “# of step for acceleration” the final delay time cause the motor to reach velocity that user enters. It means delay time can calculated as below.

$$Delay\ time = 200 - Add * (\#\ of\ step)$$ Equation 2.6

In these calculation, value of 200 is a value that based on observation. In this delay time as if stepper motor stops.

When # of step complete acceleration, delay time stay constant as its final value and continue stepping to complete steady-state velocity profile. When it is also completed, delay time should be increase for deceleration velocity profile. This time, value of “Add” is added to final delay time for each step. Thus, delay time is changed until # of step is completed by motor.

3. FRITZING DIAGRAM

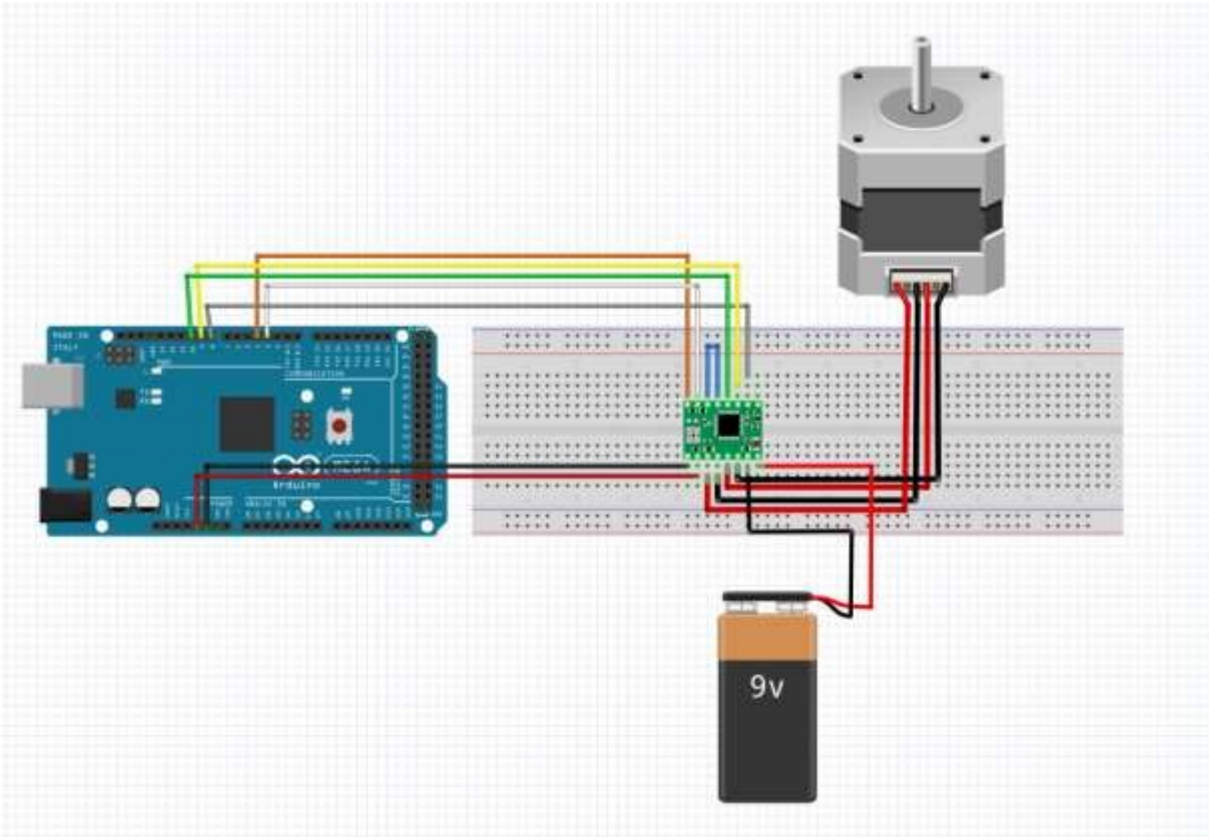

Figure 3.1. Fritzing Diagram

Note 1: In this diagram 12V adapter represented by 9 V battery block.

Note 2: Because of there is no limit switch component in fritzing application , limit switch could not be shown in this diagram.

4. BILL OF MATERIALS (ELECTRICAL)

Material	Price (₺)
10 x Jumper	2,00
1 x A4988 motor driver	25,00
1 x Arduino Mega 2560	212,00
1 x Stepper	60,00
1 x Adapter	15,00
Total Amount	**314,00**

5. SOFTWARE FLOWCHART

Flowchart of the software is given as Figure 5.1.

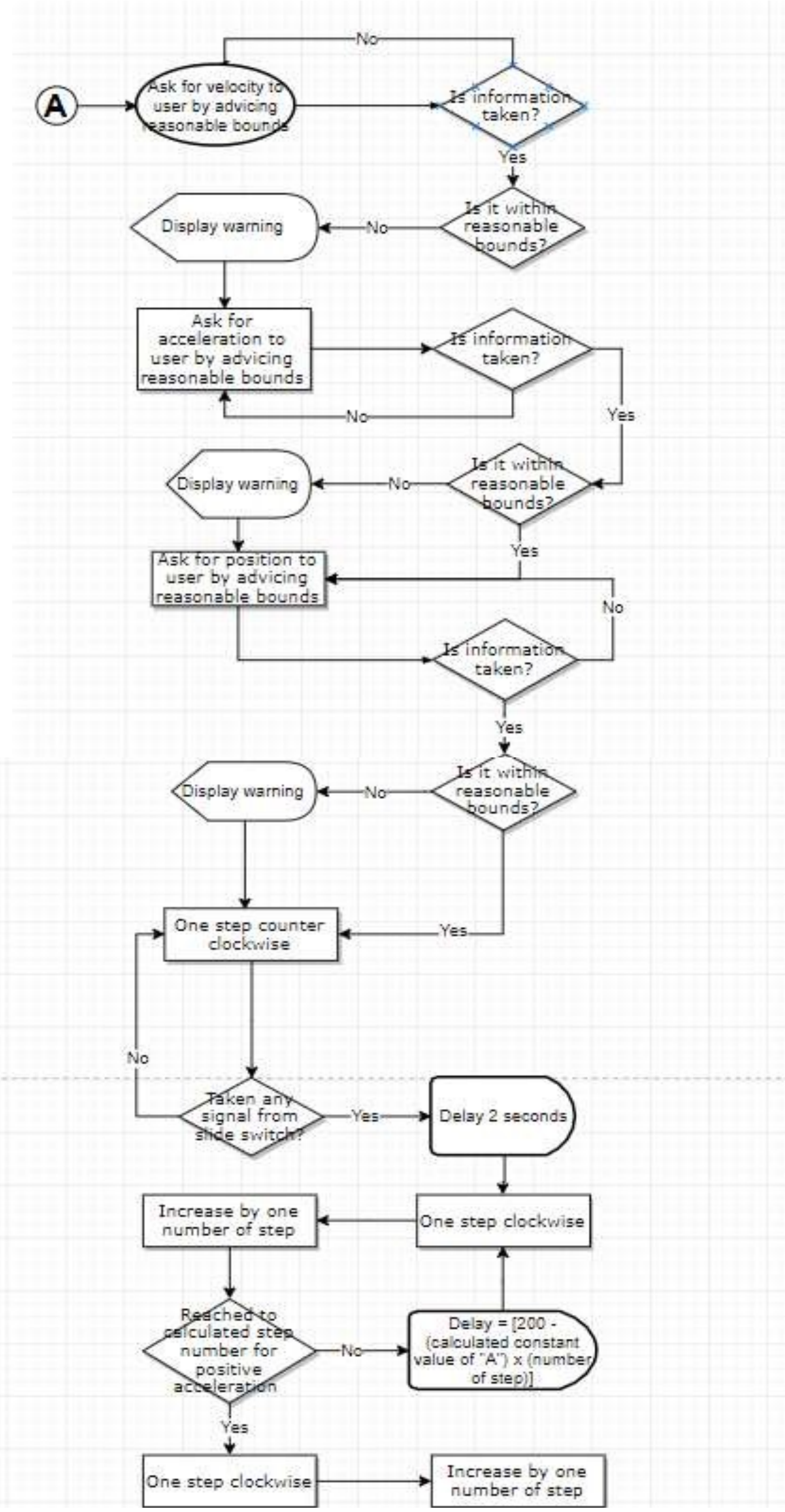

Figure 5.1.a. Flowchart of the software

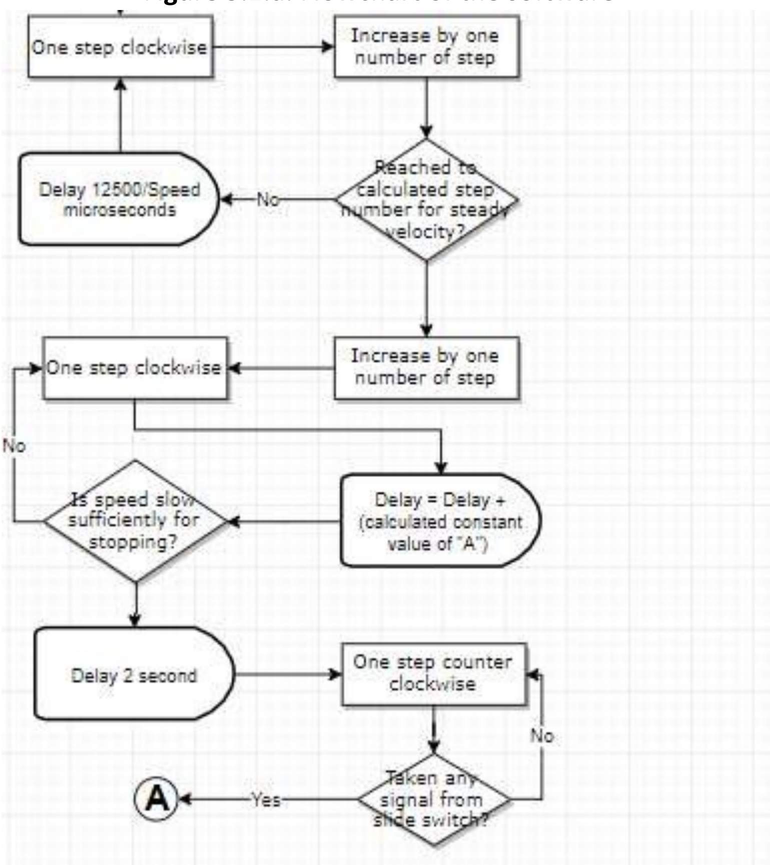

Figure 5.1.b. Flowchart of the software

6. KNOWN BUGS & LIMITATIONS

We have three type of limitations. These are :

1. Position Limitation : The length of our system is 210 mm. So we can define target position between 0-210 mm. When we enter different values from this limits, our stepper working until reach the desired position.

2.Speed limitation : We can define our maximum speed as 1000 mm/sn. Because we don't have enough length to reach bigger speed values.

3.Acceleration limitation : We can define our maximum acceleration as 5000 mm/sn^2. Because we don't have enough length to reach bigger acceleration values.

Known Bugs :

1. Acceleration bug : When we defined our acceleration value as 0 mm/sn^2 , we expect a constant speed motion. But our system don't move anywhere. Because our algorithm makes our speed value automatically zero when we defined our acceleration zero.
2. Speed bug : When we defined negative speed value, our system stops suddenly before reach the desired position.

3. Position bug : When we defined negative position value, we expect a motion to opposite direction. But our system works normally.

7. 3D CAD MODEL OF THE MECHANİCAL ASSEMBLY

The 3D CAD Model of the assembly that drawn sketchy by SOLID Works is below as Figure 7.1

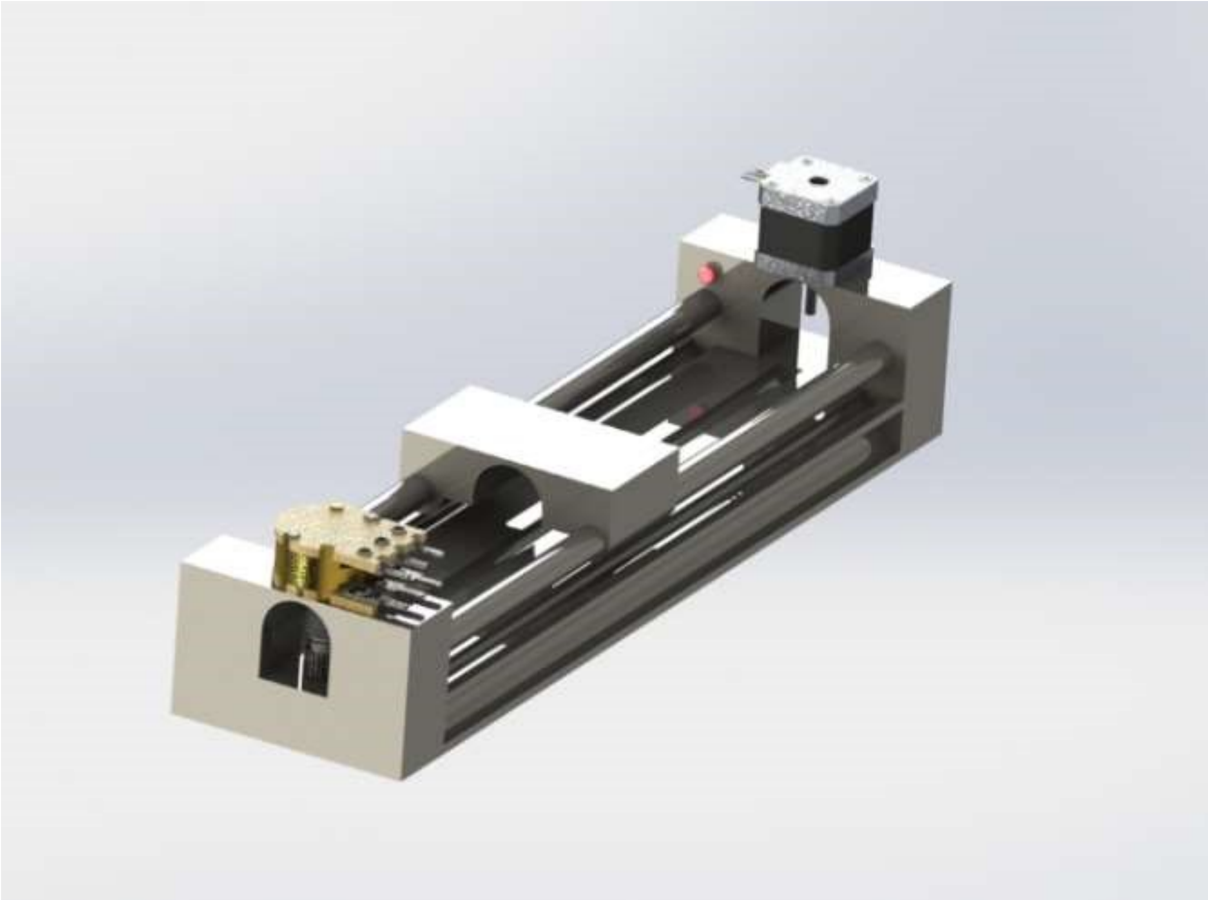

Figure 7.1 The 3D CAD Model of the assembly

8. BILL OF MATERIALS (MECHANICAL)

(List a Bill of Materials of the mechanical parts in your project (1 page max))

(Do not list the individual parts that are included in ETUKit, just write 1 x ETUKit)

- 1 x ETUKit

9. 2D/3D CAD DRAWİNGS OF CUSTOM PARTS

There is no any part that bought or design specially.

10. ARDUİNO SCETCH

Digital pin 3 is connected driver's STEP pin that used for motor stepping. Digital pin 4 is connected driver's STEP pin that determines direction of motor. Digital pin 8, 9, 10 are connected separately, MS3, MS2 and MS1 belonging to driver. These three pins always write HIGH value for switching step to microstep. 5V pin goes to driver's VDD pin to give always HIGH value. Finally, relevant to driver, a ground pin is connected with driver's ground. Another ground pin belonging to Arduino goes to directly switch's common connection. 3.3V pin goes to the board firstly, then it goes to switch's connection in the middle with help of a resistor. Lastly, digital pin 7 is connected directly 2 with switch's upper connection. With help of the last 3 connection related to switch, when button on switch is pushed, pin 7 get a signal from 3.3V pin.