

**TOBB ETU**

**Mekatronical Insturmatation Project-1**

**Video:** <https://www.youtube.com/watch?v=hPZ8onPFUYQ&feature=youtu.be>

|  |  |
| --- | --- |
|  | |
| **İsimler** | Batu Kaan Özen  Hande Yıldırım |
| **Numaralar** | 141201079  141201047 |
| **Tarih** | 27/02/2020 |

**Project Aim:**

We will ask user the location of stepper motor under serial communication window between Arduino and Personal Computer. For instance, they initialize our system one tour and then our system is supposed to move one tour and then our system ask about location.

**Equipment List:**

• PC

• Arduino (Nano/Uno/Mega)

• Breadboard x 1

• ETUKit x 1

• Stepper motor (Nema 200 steps) x 1

• LN298N motor driver

• DC162 micro switch x1

**System Description and Mathematical Calculation:**

We know from our measurement that our system has 31 cycles and one cycle contain 200 steps. Our system asks the user which location they want the stepper. It asks it as the number of stepper cycles. To be clear, if the user enters 20,23 circles, our stepper motor steps for one step 360/200 degree. One of the important things in our system is finding reference point. For example if we move our system without knowing the starting point we do not know where our system is going to be ended up. Because of this reason Micro Switch is used, when we enter the location, our system is going to move first to reference point, where our Micro Switch located and then our stepper press our Micro Switch and we understand that we are in the center point. After finding center point we count steps until we complete our cycles. You can see our system in foto-1.

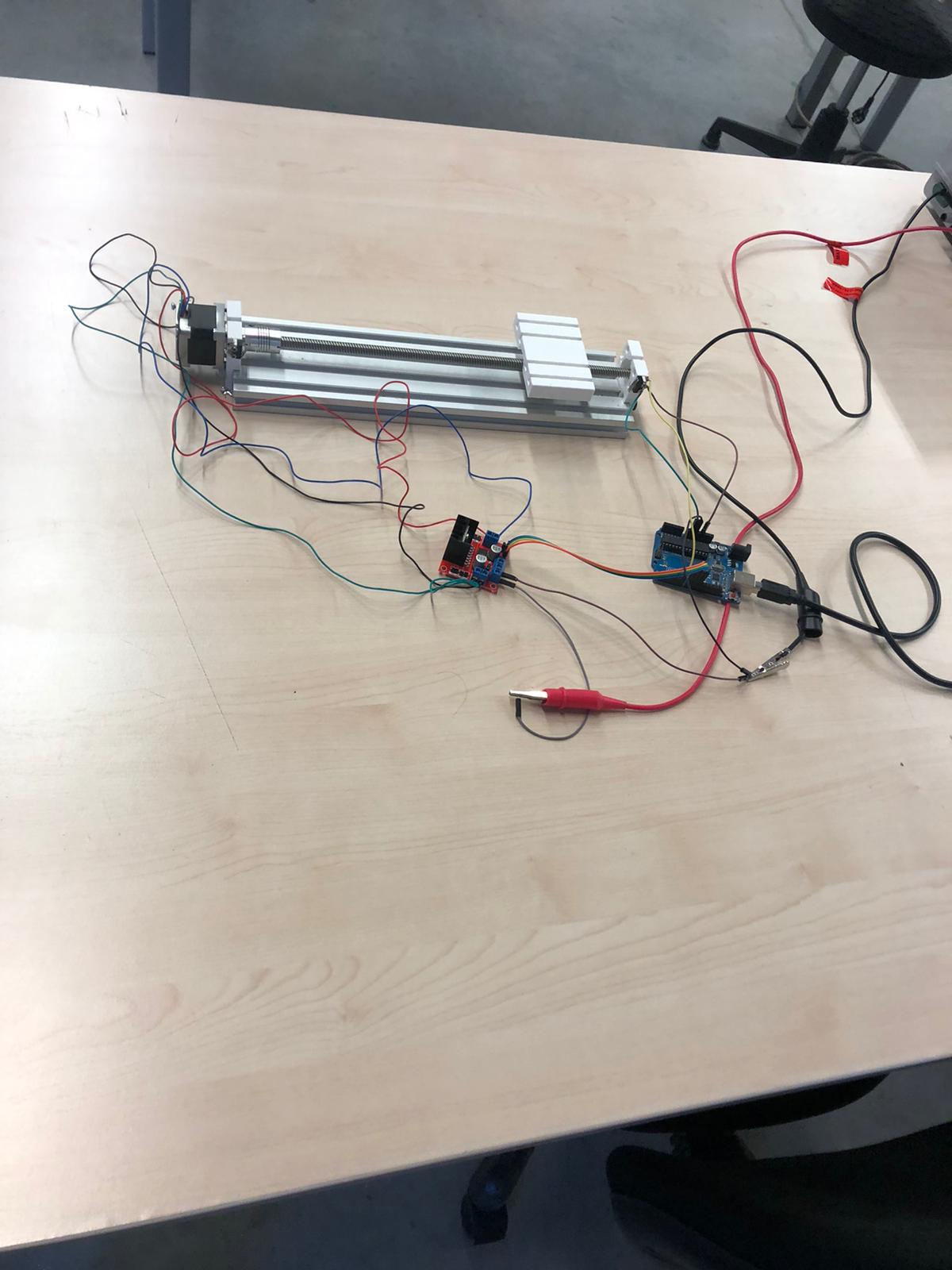


Foto-1

**Codes:**

|  |
| --- |
| /\*Batu  Kaan Özen/ Hande Yıldırım Open Loop Control code\*/ |
|  |  |
|  | #include <Stepper.h> |
|  |  |
|  | const int stepsPerRevolution = 200; // change this to fit the number of steps per revolution |
|  | // initialize the stepper library on pins 8 through 11: |
|  | int cycle; |
|  | int cyclestatue=0; |
|  | int location = 0; |
|  | int Lswitch1 = 7; |
|  |  |
|  | Stepper myStepper(stepsPerRevolution, 8, 9, 10, 11); |
|  | void setup() { |
|  | // set the speed at 60 rpm: |
|  | myStepper.setSpeed(60); |
|  | pinMode(Lswitch1, INPUT); |
|  | // initialize the serial port: |
|  | Serial.begin(9600); |
|  | } |
|  |  |
|  | void loop() { |
|  | takeCycleFromSerial(); |
|  | Serial.available(); |
|  | Serial.println(cycle); |
|  | delay(100); |
|  | Serial.println(cyclestatue); |
|  | //myStepper.step(-Location); |
|  | //Serial.println(i); |
|  | //(delay(500); |
|  | //} |
|  | // step one revolution in the other direction: |
|  | //Serial.println("counterclockwise"); |
|  | //myStepper.step(-stepsPerRevolution); |
|  | //delay(500); |
|  |  |
|  | } |
|  |  |
|  | void takeCycleFromSerial(){ |
|  | while (Serial.available() > 0) { |
|  | cycle = Serial.parseInt(); |
|  | int blank = Serial.parseInt(); |
|  | } |
|  | if ( cyclestatue !=cycle){ |
|  | cyclestatue = cycle; |
|  | findlocation(); |
|  | boolean entschalter1 = 0; |
|  | while(!entschalter1){ |
|  | myStepper.step(+stepsPerRevolution); |
|  | entschalter1 = entSchalter1(); |
|  | } |
|  | goToLocation(); |
|  | } |
|  | } |
|  |  |
|  | void findlocation(){ |
|  | location = cyclestatue\*stepsPerRevolution; |
|  | } |
|  |  |
|  | void goToLocation(){ |
|  | myStepper.step(-location); |
|  | } |
|  |  |
|  | boolean entSchalter1() { |
|  |  |
|  | if (digitalRead(Lswitch1) == 1) { |
|  | return true; |
|  | } |
|  | else { |
|  | return false; |
|  | } |
|  | } |