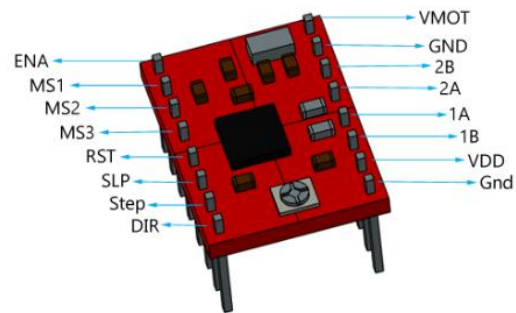


Moving the Timing Belt Linear Actuator to the Desired Distance by Entering the Length

A4988 Stepper



Motor Pins:

Vdd and GND: Should be connected to the 5v and GND parts of the Arduino.

Vmot and GND: Should be connected to 12 volt and GND to provide the 12 volt needed by the stepper motor.

1A,1B,2A,2B: Pins to which the stepper motor is connected.

Dir: Controls the direction of the motor.

Step: Controls the steps.

MS1, MS2, MS3: Microstep Selection Pins.

Sleep and Reset: When they are connected to each other, the controller becomes active.

En: When the Enable pin is active, the motor is grounded. We can limit the power usage by making this pin active and passive.

Micro step Resolution	MS1	MS2	MS3
Full Step	low	low	low
Half Step	high	low	low
Quarter Step	low	high	low
Eighth Step	high	high	low
Sixteenth Step	high	high	high

For one revolution needed steps are calculated as;

Full Step mode

$$\frac{360}{0.9^\circ} = 400$$

Half step;

$$\frac{360}{0.45^\circ} = 800$$

Quarter step;

$$\frac{360}{0.225^\circ} = 1600$$

Eighth step;

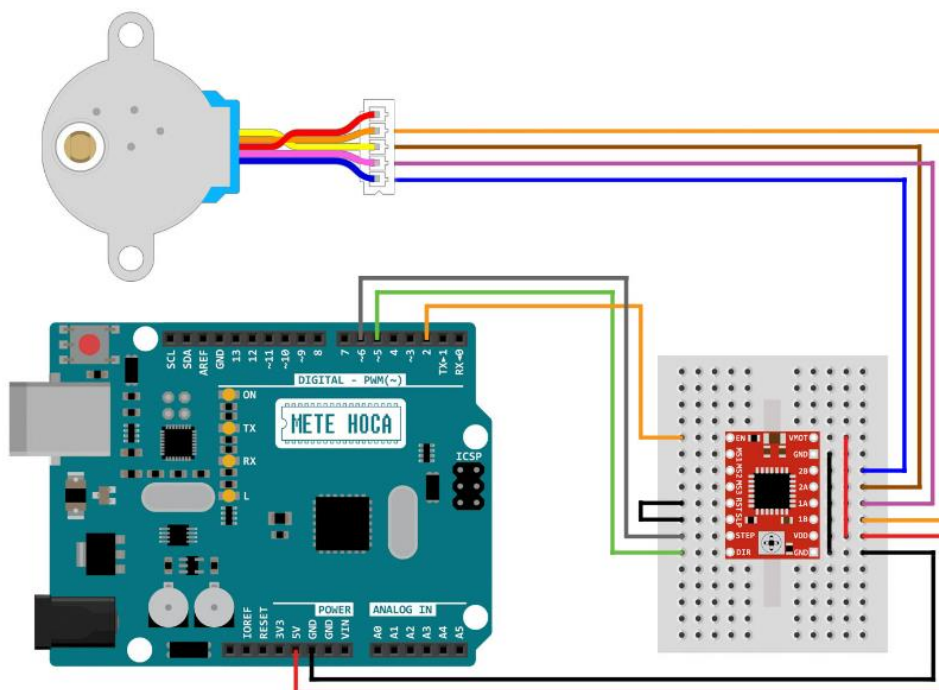
$$\frac{360}{0.1125^\circ} = 3200$$

Sixteenth step;

$$\frac{360}{0.05625^\circ} = 6400$$

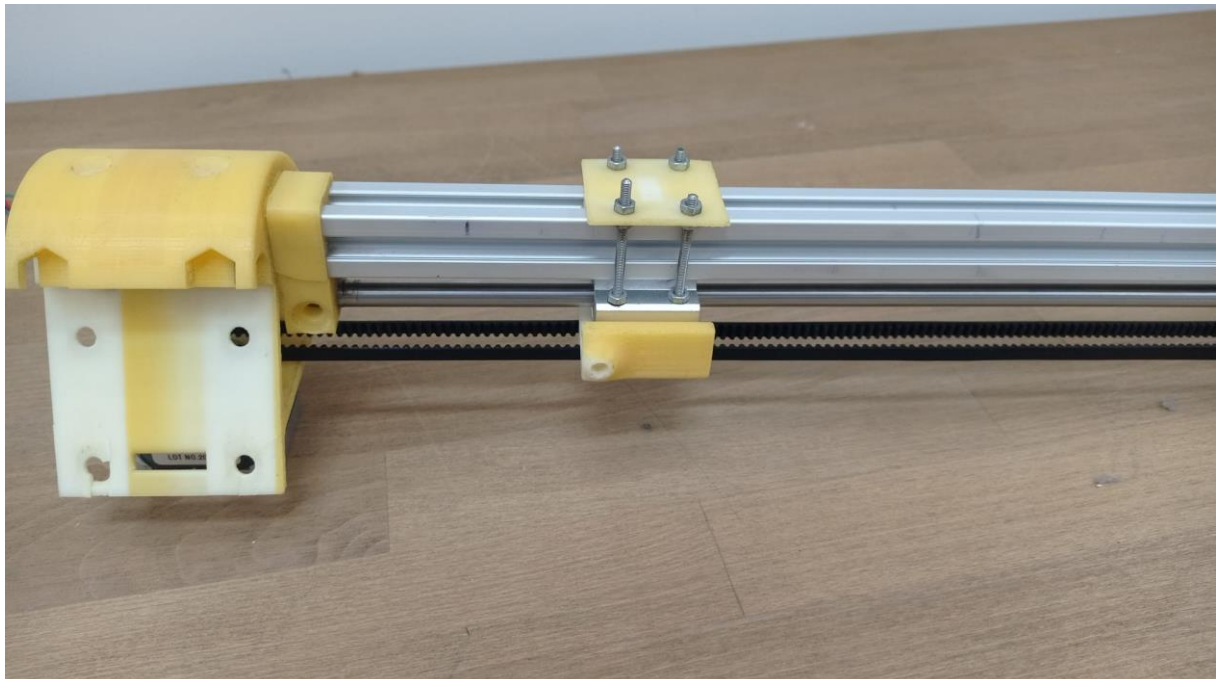
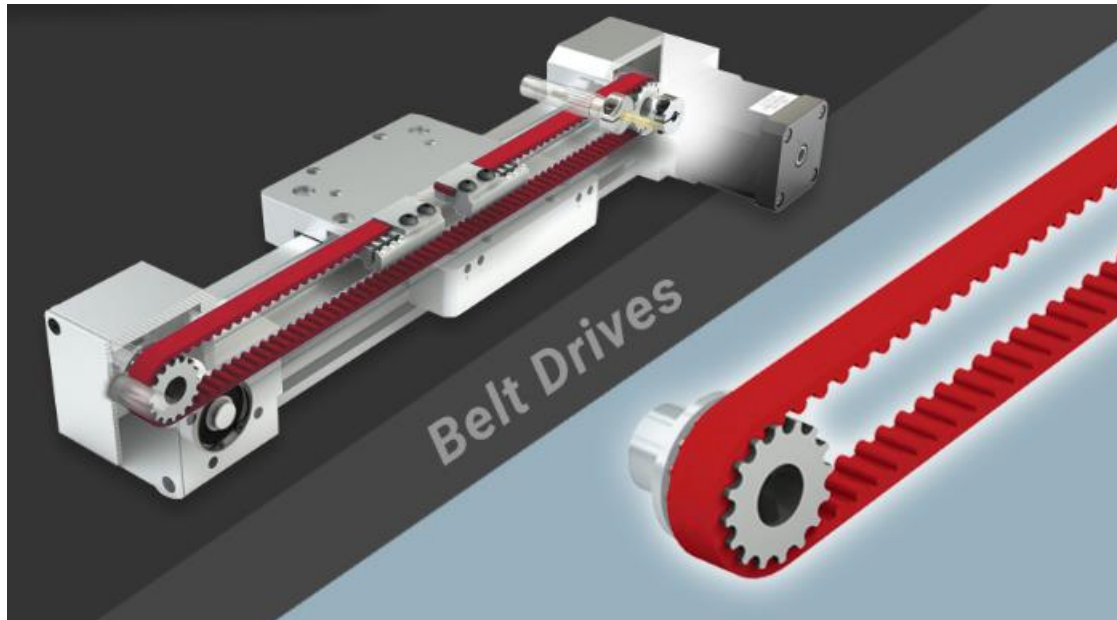
Stepper Motor Controller Connections:

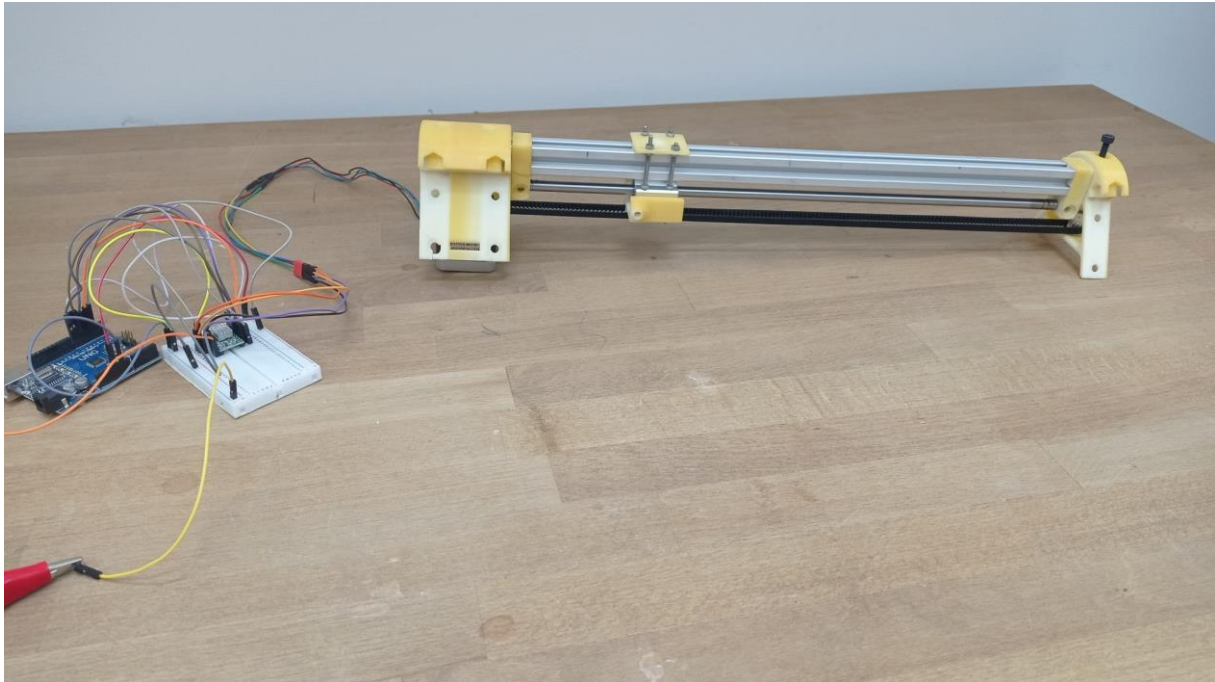
Stepper motor model is: JK42HM34-1334



Timing Belt Moving Principle:

As the gear wheel rotates, the gear of the gear wheel drive the gear of the timing belt. The timing belt starts to move in this way. As the timing belt moves, the part connected to the timing belt also starts to move.





Explanation of the movement code:

```
toplamAdim = (mmFinal/d) * stepsPerRevolution;
```

The StepsPerRevolution code is the number of times the stepper motor rotates one full turn. The amount the stepper motor moves when it rotates one full turn must be determined. When StepsPerRevolution is divided by this determined amount, the number of steps it needs to rotate to move 1 mm is determined. The mmFinal command is also set as the amount we want the actuator to move. When the determined values are substituted into the equation, the number of steps required for the actuator to move the desired distance is obtained.

The relationship between the rotation of the stepper motor and the distance is as follows.

$$d = m_g \times n_g$$

Number of gear on gear wheel connected to stepper motor: n_g

Metric of gear wheel and timing belt: m_g

The distance traveled by the timing belt in one step of the stepper motor: d

$$d = m_g \times n_g$$

$$36 = 3 \times 12$$

Moving the Timing Belt Linear Actuator to the Desired Distance by Entering the Length in Arduino UNO

```
#define dirPin 6
#define stepPin 7
#define controlPin 2
#define MS1 3
#define MS2 4
#define MS3 5
#define stepsPerRevolution 6400
#define joyX A0
#define joyY A1
double xValue;
double yValue;
double toplamAdim;
double mmFinal = 10;
double i;
double son;
double toplamAdim2;

void setup(){
  pinMode(stepPin, OUTPUT);
  pinMode(dirPin, OUTPUT);
  pinMode(MS1, OUTPUT);
  pinMode(MS2, OUTPUT);
  pinMode(MS3, OUTPUT);
  Serial.begin(9600);}

void loop() {

  // Microstep control settings:
  digitalWrite(MS1, HIGH);
  digitalWrite(MS2, HIGH);
  digitalWrite(MS3, HIGH);
  toplamAdim = (mmFinal/36) * stepsPerRevolution;

  toplamAdim2 = (mmFinal/36) * stepsPerRevolution;
  Serial.println(toplamAdim2);
  // Set the spinning direction counterclockwise:

  if(mmFinal > 0){
    digitalWrite(dirPin, HIGH);
    if(son != 1){
```

```
for( i = 0; i < floor(toplamAdim); i++){

    digitalWrite(stepPin, HIGH);
    delayMicroseconds(2000);
    digitalWrite(stepPin, LOW);
    delayMicroseconds(2000);
    son = 1;
    }
}

if(mmFinal < 0){
    // Set the spinning direction counterclockwise:
    digitalWrite(dirPin, LOW);
    if(son != 1){
        for( i = 0; i < -floor(toplamAdim); i++){

            digitalWrite(stepPin, HIGH);
            delayMicroseconds(2000);
            digitalWrite(stepPin, LOW);
            delayMicroseconds(2000);
            son = 1;
            }
        }
    }
}
```