ENME489C/ENME808M: Problem Set 2
Build and Test a 1DOF Linear Stage Robot

Prof. Axel Krieger

Fall 2017

Due Date: Monday 09/18 5 PM

September 5, 2017

In this individual lab we will make our linear stage, and test the movement accuracy of the linear stage. In the final project at the end of the semester, three students will combine their efforts and equipment to build a 3DOF cartesian robot and steer a needle into phantom brain targets using MRI and camera guidance.

It is permissable for this first lab to get help from other students in the class for machining and assembling the linear stage and to get help with connecting the Arduino controller. The matlab programming, testing of the stage, and final report have do be completed individually!

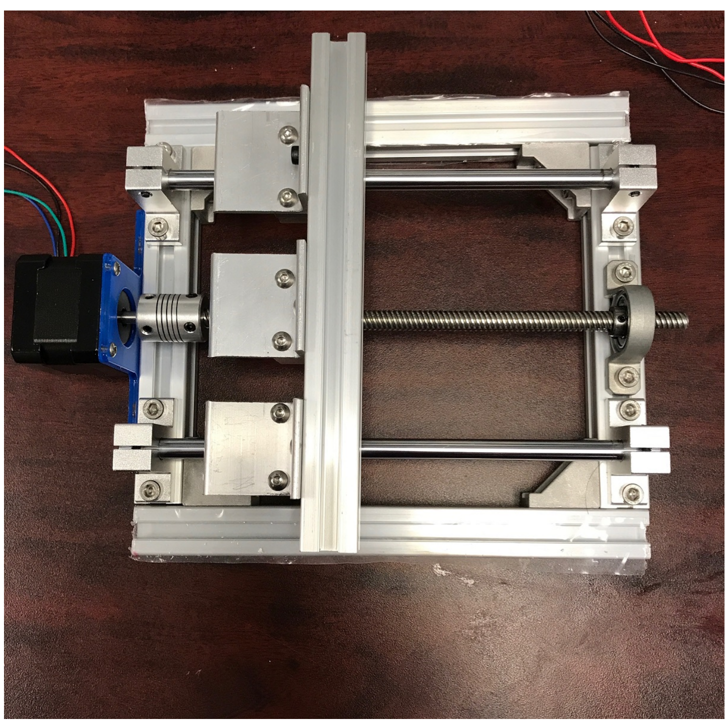


Figure 1: Final Assembly of the Linear Stage

For the linear stage, we are using a lead screw and a stepper motor as an actuator. Stepper motors convert electrical pulses into discrete mechanical movements. Stepper motors work with no positional feedback, but provide good accuracy. Rotational positioning is achieved with the help of electromagnets located inside the motors. The following video provides a good illustration of how stepper motors function and of the different excitation modes:

https://www.youtube.com/watch?v=Qc8zcst2blU&t=686s

The rotational accuracy combined with a leadscrew gives us an accurate linear positioning system. The provided leadscrew has a diameter of **8mm** and has a pitch of **8mm**. For Stepper motor specifications – as to how many steps it will need to complete a full revolution – you need to refer to the datasheet attached in the resource section. The lab is split in five sections:

1. Build the Stage (20 points)
2. Make the motor spin (20 points)
3. Attach the motor and leadscrew (20 points)
4. Move the stage and test the accuracy of the robot (40 points)
5. Repeatability analysis of the stage [MUST for grad students and OPTIONAL for undergrad students] (20 points)

Before you start attempting this lab, make sure you have the latest version of MATLAB (R2017a at present) installed on your system. The newest version of Matlab can be downloaded from terpware.umd.edu. Also, please make sure that you have all the components required for this lab. The required components are listed at the start of each section.

1. **Build the Stage:**
   For this part you will use the following items in the kit provided to you:

   – 205mm long Aluminium Extrusion bar(2 Nos)
   – 150mm long Aluminium Extrusion bar (2 Nos)
   – 187mm long Aluminium Extrusion bar (1 Nos)
   – 1 ft. Aluminium L-shaped channel (1 Nos)
   – LeadScrew Set (1Nos with following - 1 8mm x 200mm Leadscrew, 1 LeadScrew Nut Housing, 2 8mm x 200mm cylindrical Shafts, 2 SC8UU Linear Bearings, 4 SK8 bearings, 1 KP06 Bearings, 1 Shaft Coupler, 1 Mounting Plate)
   – Corner Braces (4 Nos)
   – M5 T-nuts (18 Nos)
   – M4 T-nuts(6 Nos)
   – M4 x 8 Socket-Head Screws (6 Nos)
   – M4 x 10 Socket-Head Screws (6 Nos)

- M5 x 10 Socket Head Screws (10 Nos)
- M5 x 8 Socket Head Screws (8 Nos)
- M3 x 6 Socket-Head Screws (2 Nos)
- M3 x 10 Socket-Head Screws (2 Nos)
- Hex-Key Set (1.5mm, 2mm, 2.5mm, 3mm, 4mm)

As a first step we will need to machine **three** brackets from the 1ft Aluminum L-shaped channel you are provided with. Refer to figure 2 for the dimensions of the holes and size of the bracket. Be extremely careful, when cutting the channel and bars. Electrical saws are very dangerous. Complete the safety training first and follow proper safety instructions and wear proper safety gear such as safety glasses. The bars are locked together with the help of corner braces. Each corner brace needs two M5 nuts and two M5x8 bolts. The figure 3 illustrates the way these pieces are used:

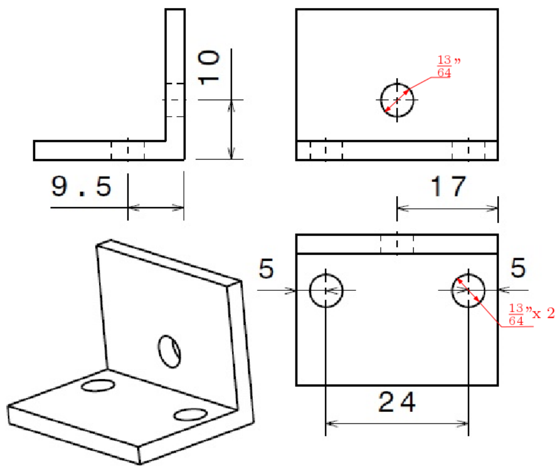


Figure 2: Machine three such brackets. Use drill bit of size $\frac{13}{64}$". All other dimensions are in mm.

You would want to assemble the base of the stage using the two 206mm, and two 150mm beams as shown in figure 4. After the base is assembled, we will mount our leadscrew arrangement onto the base. For assembling the base you should follow these steps as a guideline:

- Use 4mm Hex Key and M5x8 bolts for assembling the corners. Refer to figure 5 for the process.
- Mount the SK8 bearing using M5 bolts and M5 T-nuts at the end of the extrusion bars, such that the edge of the bearing coincides with the edge of the 150mm beam. Also, make sure that nut on the side of SK8 bearing is on the outside to facilitate the tightening process. Refer to figure 6 for a picture.

(a) The T-nut drops into the T channel of the beam with its smaller length parallel to the length of the beam. Once the bolt is tightened, it rotates to have the smaller length lock inside the beam.

(b) Corner Brace Assembled

Figure 3: Using T-nuts to assemble the corner braces

– Slide the 8mm shafts through the SK8 bearings on one end, then slide the two SC8UU bearings and finally slide through the SK8 bearings at the other end. Clamp the shafts inside the bearing tightly in that position. Use 3mm Hex Key for tightening the nut on SK8. Figure 7 illustrates this process.

We will assemble the leadscrew later, when we will mount the motor. First we will check our connections.

2. **Make the motor spin:**
For this part you will use the following components:

   – Arduino Uno (with USB cable).
   – 12V 1A Power Supply
   – BreadBoard
   – M-M Jumper Cables
   – TB6612 stepper motordriver
   – Soldering Gun and solder for soldering some header pins (not provided with the kit)
   – 12V 0.4A NEMA-17 Stepper motor
   – Motor Mounting Plate

As a first step solder header pins and the terminal block onto the stepper driver. Also, you will need the Arduino IDE setup on your computer. Make sure you have the 'stepper' library installed. If you are new to using Arduino, you can check the resources section at the bottom of this writeup for helpful links related to Arduino. We are using

Figure 4: Base of the Stage

(a)

(b)

Figure 5: Base Assembly. 205mm beams (length) used along with 150mm (width) beams.

TB6612 stepper motor driver from Adafruit. More details about this driver can be found here. You might want to assemble the motor first to the mounting plate as shown in figure 8. After you complete the assembly, we make the connections as follows:

- Vmotor to 12V (Vin pin on Arduino)
- Vcc to 5V
- GND to ground
- AIN2 to Digital 2
- AIN1 to Digital 3
- BIN1 to Digital 4
- BIN2 to Digital 5

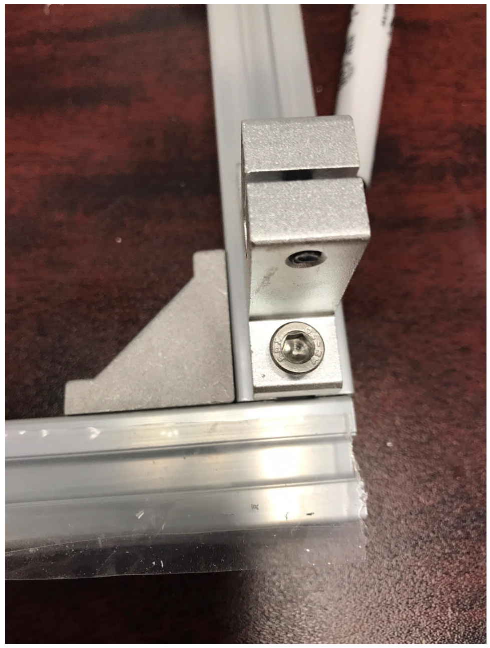


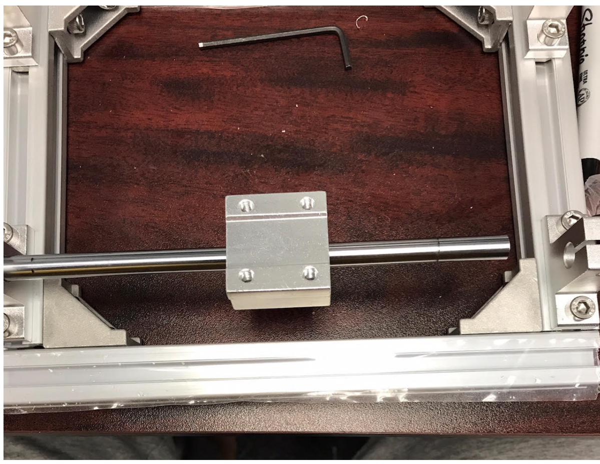
Figure 6: Sk8 assembly

- PWMA and PWMB to Vcc (orange wire)
- MOTORB to RED and BLUE motor wires
- MOTORA to BLACK and GREEN motor wires

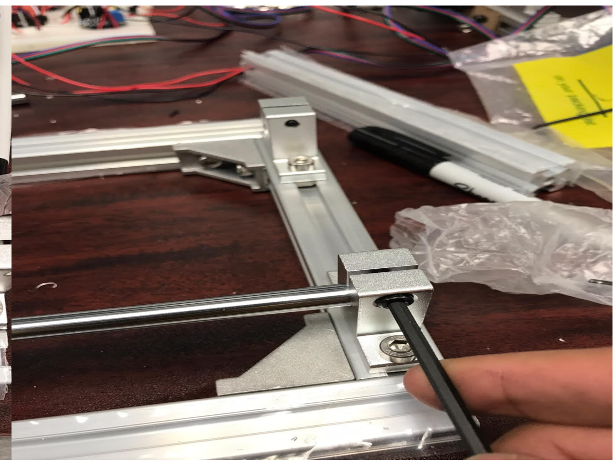
Refer figure 9 for a connections. After you have completed the wiring you can check your connections by uploading the sample script **test_stepper.ino**, which is provided to you in the .zip file. Connect the Arduino through USB and upload the sketch to the Arduino. If you get any error while uploading, verify that the IDE recognizes the correct port from Tools>Port option. **CAUTION: Power the Arduino using the external power supply (12V 1A) first, before uploading the sketch.** This way we are ensuring that we get 12V from the Vin pin from the Arduino, which is what is required for powering our stepper motor. If your connections are correct, you should be able to spin the motor smoothly forwards and backwards.

3. **Attach the motor and leadscrew:**
   Make sure you have completed previous two sections before starting this section. For this part you will need the following components:

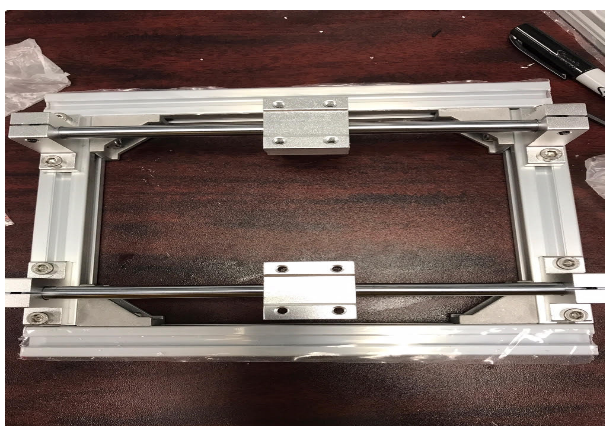
   - 3 M4 bolts and 3 M4 T-nuts
   - 2 M5x10bolts and 2 M5x10 T-nuts
   - 2 M3x10 bolts
   - KP06 bearing
   - Shaft Coupler



(a)



(b)



(c) Base Assembly

Figure 7: Assembling the Shaft and the Linear Bearings

Figure 8: Use M3x6 bolts to fasten the motor to the plate. Also, you can put some M4x8 bolts along with the M4-T nuts loosely as shown.

  - 8mm leadscrew
  - Nut Housing

  You can follow these steps as a guideline:

  - Use M3x10 bolts to fasten the leadscrew nut to the nut housing. Refer figure 10.
  - Attach the mounting plate to the extrusion bar using M4 bolts and M4 nuts such that the plate is equidistant from both the neighbouring Sk8 bearings. This is shown in figure 11
  - Attach the shaft coupler to the motor shaft. The shaft should go inside the coupler for about 50% of length. Do not tighten the setscrews on the coupler yet. Slide the leadscrew in through the nut housing. Make the leadscrew axis coincident with the shaft coupler and push it inside the shaft coupler. Tighten the set-screws on the coupler at both ends. Support the other end of the leadscrew using a KP06 bearing and mount this bearing on the extrusion bar using M5 bolts and M5 T-nuts. This step is shown in figure 12.

- Mount your machined brackets on the nut housing and the SC8UU linear bearings using the M4x10 bolts. Finally, mount your 187 mm beam on the brackets and fasten it in place. This is shown in figure 13.

- Make sure the connection of the motor shaft to the coupler is smooth and the stage translates smoothly. Now, your entire stage should be ready.

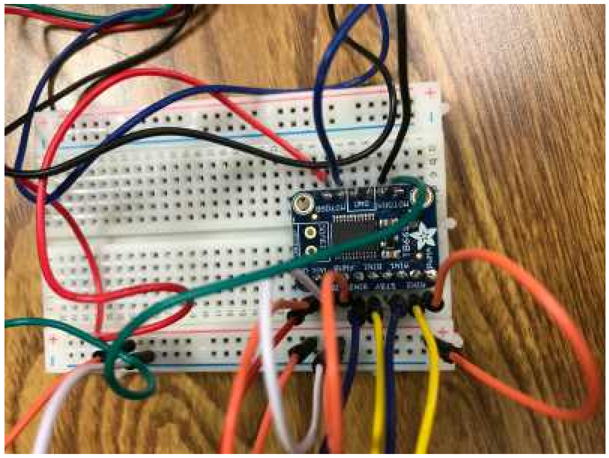


Figure 9: Connections to the driver

Figure 10: Use M3x10 bolts to assemble the LeadScrew Nut to the Nut Housing. You will need to use 2mm hex key for this

4. **Move the Stage and test the accuracy of the robot:**

   – **Task_0:** Before you proceed to the next section, make sure you manually rotate the lead screw such that stage is at the home position. We will define home as the position when the nut housing and the shaft coupler are in contact.

   We want to control our steppers using MATLAB. This is because, we will be doing kinematic control and image processing in future labs and would like to incorporate that in the final project. MATLAB environment is ideally well suited for Image processing and will also be used for the kinematic calculations. You could say that the sole purpose of Arduino is to spin the motors and we will control the direction and amount of rotation of the steppers by serially communicating with Arduino through MATLAB.

   You are provided with an Arduino Sketch called **stepper_move_matlab.ino**. Upload this sketch to Arduino (while ensuring that the Arduino is powered by the external

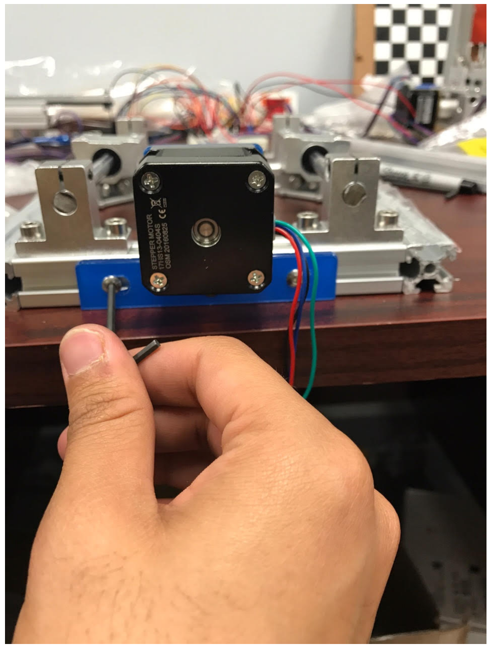
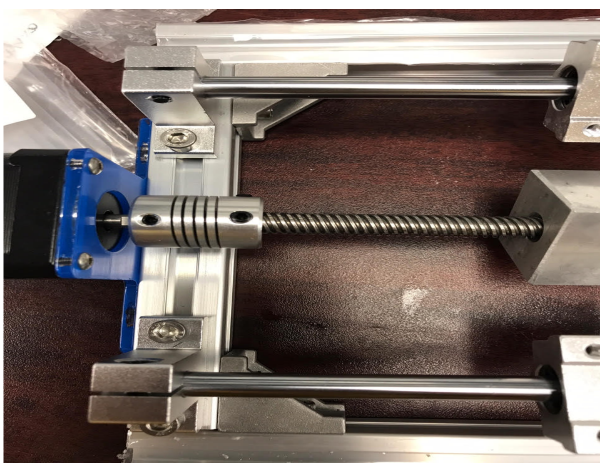


Figure 11: Use M4x8 bolts and M4 T-nuts to assemble the motor to the base. You will need to use 2.5 mm hex key for this



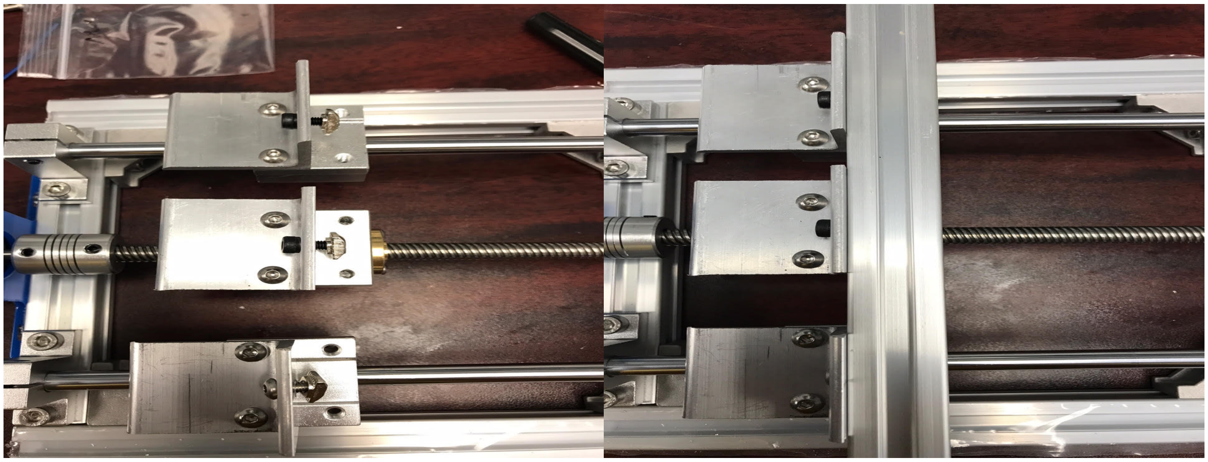
(a)

(b)

Figure 12: Attaching the Leadscrew and Motor to the stage. Use 2mm Hex Key to tighten the setscrews on the coupler.

power supply). This sketch basically listens to serial inputs from MATLAB and controls the stepper motors using the Stepper library of Arduino. Also, you are provided with four MATLAB(.m) files:

- **linearstage_lab.m**: This is our main script which we will run to complete the activity for this lab. This file will call your code you write in task.m. You need not change anything in this file except the **port_name** variable – which might differ



(a) (b)

Figure 13: Attach the 187 mm beam across the stage to restrict the rotational motion of the leadscrew nut. Use 2.5mm Hex Key to fasten the bolts on the beam.

based on your computer. You can check for your port name from Tools>port in Arduino IDE.

- **stepper_move.m**: This is an utility function which you will call in your implementation in task.m. You can read more about this function and what all arguments it needs in the first few lines of the stepper_move.m file. You don't need to modify this file.
- **setupSerialArduino.m**: This is an utility function which deals with the serial communication part of controlling the Arduino to spin the motors. Again, you don't need to modify this file. However, you might want to skim through the code to get an understanding of how we are setting up the serial connection.
- **task.m**: You have to write your code in this file. You should write a logic to control the stepper motor such that the stage moves by **50 mm** first then further **30 mm** and again go back to the zero position – all operations in one go.

-An important variable which you should be concerned about is **curr_pos_x**. As the name suggests this variable keeps track of the current position of the stage. Since, with steppers we don't have a positional feedback, we update the **curr_pos_x** variable according to the input we provide it to move. This is how we keep track of the position of the manipulator stage.

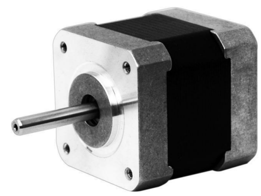
- **Submission:**
  To test if your stage achieves the desired position accurately, you will use a graph paper and mark the positions of the stage at 0mm, 50mm, and 80mm when your code is running and your linear stage is being moved. You can use the tip of the nut housing as the reference position for the stage. Then, you will report the error between your measured and expected positions. You will need to turn in the following in a zip file:
  - Your file task.m

- A picture of the graph paper with start, middle, and end positions marked clearly.
- A short lab report (<= 1 Page) describing your work and results including a picture of your assembled stage and reporting the accuracy of your linear stage.
- Grad students need to include a report with analysis on the repeatability of the linear stage as well. You will need to repeat the accuracy testing several times and report the average, median, and standard deviation of the positional repeatability of your stage.

- **TroubleShooting:**
  - **Motor Fails to spin or doesn't spin smoothly:** Make sure that correct colored wires of the stepper are connected to the stepper driver. Also, verify your other connections to the TB6612 driver again.
  - **Serial Communication Failure between MATLAB and Arduino:** Make sure the **port_name** variable in linear_stage.m matches with what is selected in Arduino IDE. The port name is selected through the Tools>Port option on the Arduino IDE. If problem persists, clear all variables in MATLAB (MATLAB command "`clear all`") and run linear_stage.m again.
- **Resources:**
  - **Stepper Motor Specification Sheet:** as seen in Figure 14
  - Useful Arduino Links:- Download IDE :
    https://www.arduino.cc/en/Main/Software
    Working with IDE:
    https://www.arduino.cc/en/Guide/Environment
    More About Arduino:
    https://www.arduino.cc/en/Guide/Introduction

1.8° 42MM High Torque Hybrid Stepping Motor

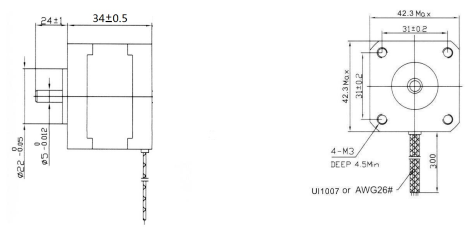
Item	Specifications
Step Angle	1.8°
Step Angle Accuracy	±5% (full step, no load)
Resistance Accuracy	±10%
Inductance Accuracy	±20%
Temperatru Rise	80°CMax. (rated current,2 phase on)
Ambient Temperatuar	−20°C~+50°C
Insulation Resistance	100M?Min.,500VDC
Dielectric Strength	500VAC/ for one minute
Shaft Radial Play	0.02Max. (450 g−load)
Shaft Axial Play	0.08Max. (450 g−load)
Max. radial force	28N (20mm foom the flange)
Max.axial force	10N



● 42MM Hybrid Stepping Motor Specificatiors

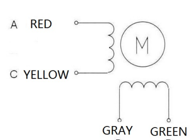
Model No	Rated Voltage	Current /Phase	Resistance /Phase	Inductance /Phase	Holding Torque	# of Leads	Rotor Inertia	Weinght	Detent Torque	Length
	V	A	Ω	mH	Kg−cm		g−cm²	kg	g−cm	mm
XY42STH34-0354A	12	0.35	34	33	1.6	4	35	0.22	120	34

● Dimension



● Wiring Diagram

4 LEADS



● Pull out Torque Curve

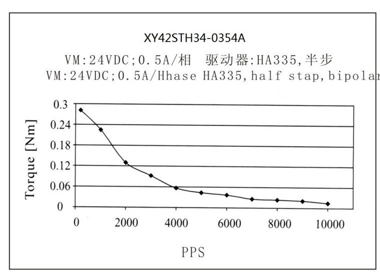


Figure 14: Stepper Motor Datasheet