Designing an Automated Linear Translation Stage

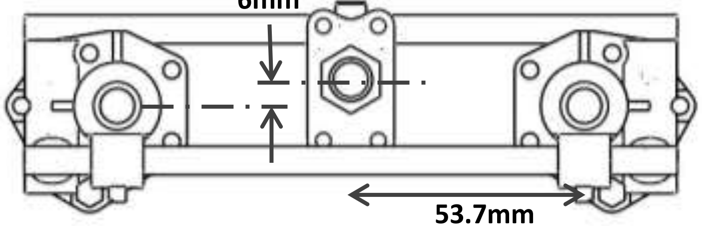
Introduction
This guide is intended to provide an overview of the components and techniques required to build a simple automated translation system. Rather than a step-by-step guide I have tried to keep the sections as general as possible so that you can build your stage out of whatever components you have to hand. However if you would like to recreate the linear translation stage used in the OpenLabTools microscope, detailed instructions of the whole construction can be found here.

The translation stage that we build for the OpenLabTools consisted of four main subsystems:
- Structure – to support all the mechanical components.
- Actuation – the mechanical drive system for the microscope – in our case we used a leadscrew mechanism for the vertical motion.
- Stage assembly – the flat stage to hold samples: and the supporting steel frame.
- Control – an Arduino microcontroller and hardware to control the steppers, prevent the stage from running out of bounds, and to allow the user to interact with the stage.

Structure
The first requirement of a linear translation stage is a rigid support structure to support the shafts and mechanical components. It is very important that the shafts remain perfectly vertical and parallel, so the structure must be rigid enough to avoid twisting or sagging.

Before building the structure there are a few things you should think about:
- **Stage size**
  A pretty obvious point, but the structure will be the main thing that restricts the size of the stage. Keeping the structure as open as possible means that you may be able to expand the stage outside the bounds of the frame.
- **Stability**
  Consider the profile of the base of the structure. A triangular base will mean that the stage will be stable on any surface. If you are using OpenBeams, as we were (see below) a square or trapezoid profile makes sense as the connections between beams are strongest at 90 degrees.

- **Stepper motor**
  How are you going to attach the stepper motor to the structure? If you are using a gearing system, how will this be integrated into the structure?
- **Vertical shafts**
  How will the shafts be connected to the structure? You should think about the layout of the shafts and drive system relative to the stage: do you want to support the stage in 4 corners or have it cantilevered from one side?



In the OpenLabTools microscope we used a modular building system called OpenBeam for our rigid frame. OpenBeam is a Mechano-like system consisting of extruded aluminium section and plastic brackets. It is excellent for very quickly assembling a solid frame. OpenBeam is an American open source project which began life on the crowd-funding website Kickstarter. The designs for all the brackets can be found online here, so if you have access to a 3D printer you can even print them yourself. For more information on OpenBeam, see www.openbeamusa.com.
OpenBeam is available as starter kits or as individual pieces from www.technobotsonline.com. Starter kits are also available from www.adafruit.com.

In the OpenLabTools microscope

4	OpenBeam Aluminium Extrusion 300mm	Technobots
12	OpenBeam Aluminium Extrusion 150mm	Technobots
2	OpenBeam Aluminium Extrusion 60mm	Technobots
12	OpenBeam L-Brackets	Technobots
16	OpenBeam T-Brackets	Technobots
2	OpenBeam Shaft Clamp Sets (pack of 2)	Technobots
1	OpenBeam Stepper Motor Mount (NEMA 17)	Technobots
156	M3 Buttonhead Screws 6mm	Technobots
156	M3 Hex Nuts	Technobots

Actuation
Stepper motors
There are three main types of motor used in small scale model applications: brushless DC, stepper and servo motors. DC motors are the most basic, used where accurate positioning is not important. Servo motors are DC motors which have built-in feedback sensing and control. Stepper motors work on a slightly different principle to DC motors: rather than continuous rotation, they move in discreet steps, which makes them ideal for CNC applications. Stepper motors were the perfect choice for our linear stage, because it meant that we could gain very precise control over the position of the stage by moving up and down individual steps.

Before choosing a stepper motor, there are a few things you will need to think about. Firstly you will need to consider the resolution that you want to achieve with your stage. Because stepper motors move in discreet steps, there will be a minimum fixed distance that your stage can move. This depends on three things:

- Steps-per-revolution of the stepper motor
- Pitch of the lead-screw
- Gear ratio

These factors are related by the formula:

$$resolution = \frac{leadscrew\ pitch\ \times gear\ ratio}{steps\ per\ rev.}$$

In our linear stage we used a 200 steps-per-revolution stepper motor, a M8 lead-screw (1.25mm pitch) and a direct drive i.e. no gears. We calculated theoretical resolution of 6.25 micrometres per step. We later verified this and found our actual resolution to be closer to 6.4 micrometres, demonstrating that even with an inexpensive setup it is possible to achieve very precise control.

You may also wish to note whether the stepper motor is unipolar or bipolar. This refers to the way in which the electromagnet coils are wound within the motor. We chose to use an

Adafruit motor shield which can handle both unipolar and bipolar steppers with ease, but if you wish to use a different motor driver beware that the two types require different drivers. Unipolars are simpler and with some coding skill may be driven directly from a microcontroller, while a bipolar will require a dedicated 'H-bridge' driver circuit. Unipolar and bipolar steppers can easily be distinguished by the number of leads: a unipolar typically has 5 leads whilst a bipolar only has 4.

The final crucial point to note is the supply voltage of your stepper. These generally range from 3V to 12V for hobby stepper motors. If using the Adafruit shield you can simply plug in and play any motor between 4.5V and 13.5V. If your motor falls outside this range you will need to connect an external power supply to the board, and set power to external by removing the jumper on the shield. Lower voltage steppers require a higher current to achieve the same torque as the larger steppers, and will quickly get hot and burn out your shield if connected to the wrong shield (as we unfortunately discovered!).

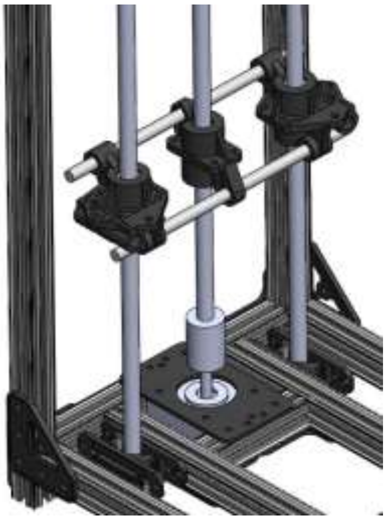
Adafruit.com has some excellent tutorials on using stepper motors with their V2 motor shield; with an Arduino and driver chip; or with a Raspberry Pi and driver chip:

Arduino and V2 motor shield	Link
Arduino and L293D motor control chip	Link
Rasperry Pi and L293D motor control chip	Link

In the OpenLabTools microscope

1	Adafruit 12V Stepper Motor 200 steps/rev.	Adafruit
1	Aluminium Flex shaft coupler	Adafruit

Lead-screws



A lead-screw is a very simple mechanism for converting rotational motion into linear motion. It consists of a threaded shaft, which rotates with the motor; and a nut, which is constrained from rotating. The relative rotation causes the nut to move up and down the shaft.

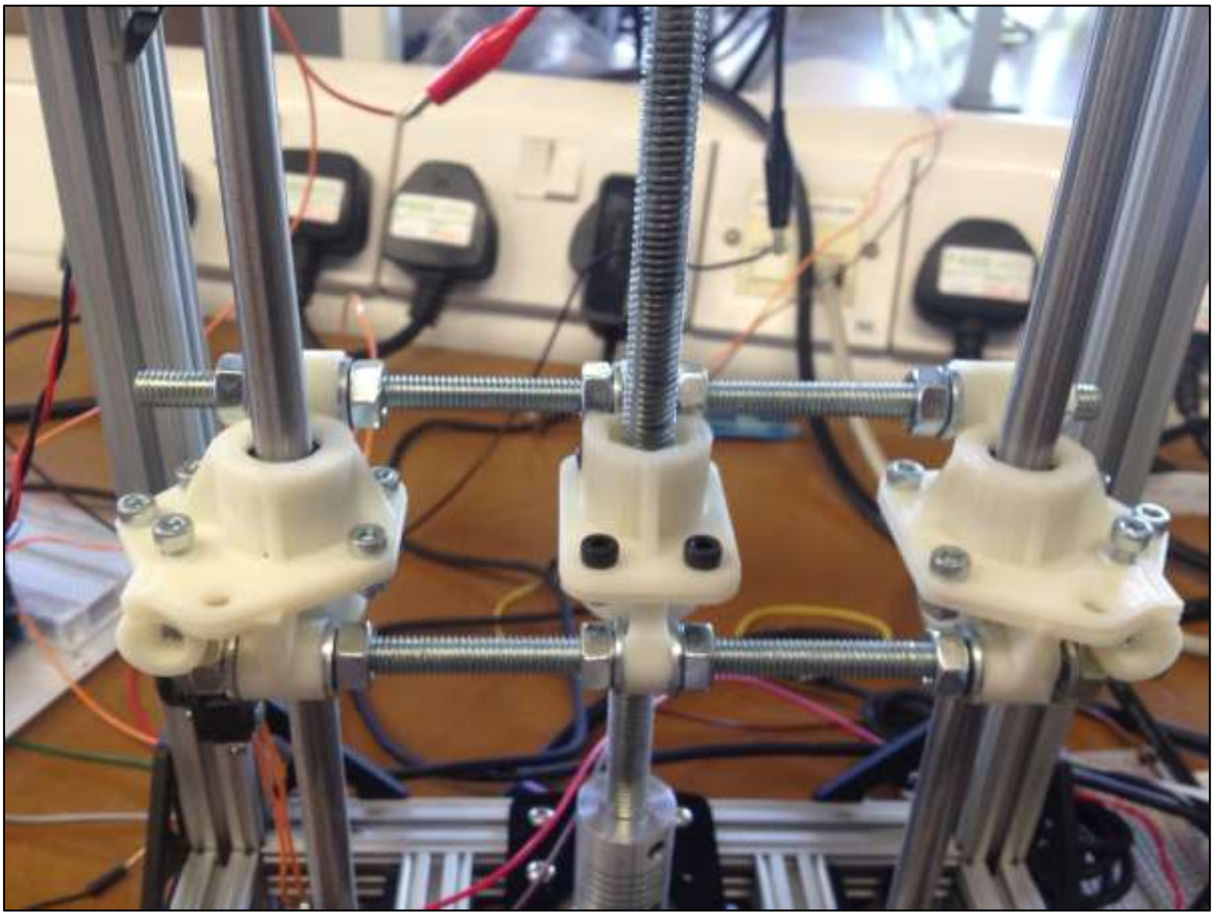
A basic lead-screw can be built using exactly the same threaded rod and nuts which you might use as fasteners in any hardware application. These will provide just almost as good precision and control as the much more expensive ball-screws or ACME screws. The main downside with a basic leadscrew is that there is significant friction between the nut and the thread, and over time the thread will wear. If your application involves large loads or continuous use, it may be worth investing in a ball-screw, which uses ball bearings to make operation almost frictionless.

Surprisingly, a more expensive screw does NOT in general lead to greater precision. Ball screws make use of bulky ball bearings and so typically have a much larger screw pitch than standard threaded rod of the same diameter. The precision can only be improved with a finer stepper motor or with a gearing system, as described above.

The table below details the precision you can expect to achieve using standard metric studding. The precision is the distance the stage will move with each step of the stepper motor.

Stepper motor	Screw	Screw pitch	Resolution
200 steps per revolution	M5	0.8mm/rev	4.00μm
	M6	1 mm/rev	5.00μm
	M8	1.25 mm/rev	6.25μm
	M10	1.5 mm/rev	7.50μm
	M12	1.75 mm/rev	8.75μm

The main consideration in designing the leadscrew mechanism is how to prevent the lead-nut from rotating relative to the shaft. We achieved this using a custom 3D printed mount which an M8 hex nut could be placed inside and the two halves bolted together. The mount itself is prevented from rotating by two crossbeams, which connect to linear bearings which slide up and down on steel rods.



In the OpenLabTools microscope the vertical leadscrew is only constrained in 2 places. At the bottom the flex shaft coupler prevents horizontal and vertical movement and rotation about the axis of the screw. However the 'flex' coupling allows rotation in the horizontal axes. This rotation is restricted by the lead nut and bearing assembly itself.

In the OpenLabTools microscope

1	M8 threaded rod (studding) 270mm	Wickes
1	M8 hex nut	Wickes

Shafts and bearings

The shafts used in the OpenLabTools microscope are 8mm chromed steel rods. You should ideally use hardened steel as over time the ball bearings will cause some wear to the rod.

The bearings are LM8UU linear bearings, encapsulated in a custom 3D printed case. The '8' in LM8UU refers to the inner diameter of the bearing: hence LM6UUs are used for 6mm shafts. The bearings each contain 4 rows of ball bearings, with a rubber seal to protect the bearings from getting clogged. Be very careful when inserting these onto a shaft as the ball bearings can sometimes pop out.

The linear bearings have the following dimensions:

	Inner diameter	Outer diameter	Length
LM8UU	8mm	15mm	24mm
LM6UU	6mm	12mm	19mm

For the bearing mounts I used the following size holes to contain the LM8UUs and LM6UUs. These gave a tight push-fit which prevents the bearings from moving relative to the mounts.

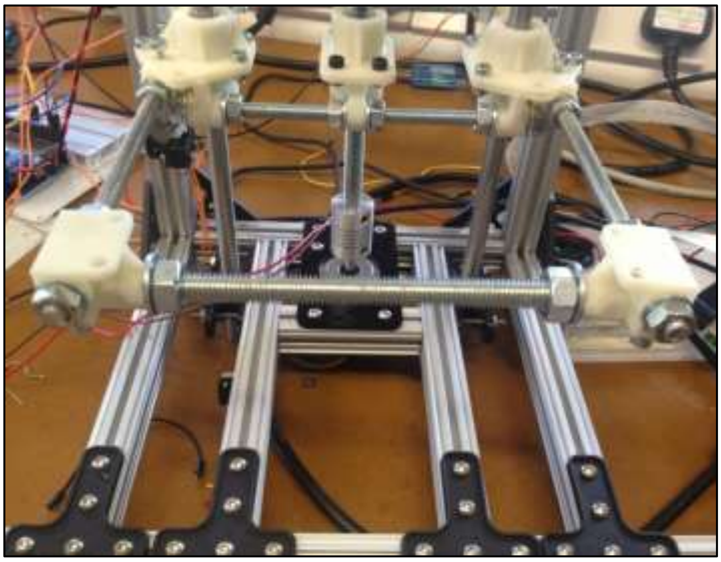
	Inner diameter	Length
LM8UU	16mm	25mm
LM6UU	13mm	20mm

In the OpenLabTools microscope

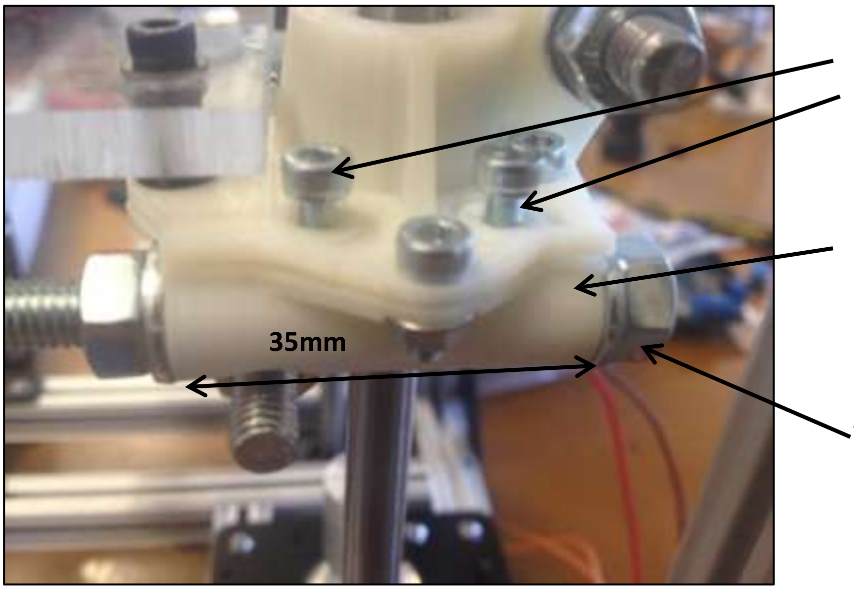
2	LM8UU linear bearings	[Adafruit](#)
2	8mm Silver steel rod 333mm	[Technobots](#)

Stage

As with the main structure, the most crucial feature of the stage is rigidity. We chose to use a cantilevered stage to give maximum access to the platform area. Because this design is only supported at the back, a steel rod frame with 3D printed connectors provides additional support.



While ABS is a very stiff plastic, there will still be some inherent flexibility in the system, so if you are creating your own 3D parts bear this in mind.



M6 threaded rod clamped down with two M3 screws to prevent rotation

Rod is supported by long tube 2mm wall thickness

M6 nuts and washers tightened onto plastic prevent movement and rotation

The stage itself is manufactured from a 5mm thick sheet of Perspex. This by itself is very stiff and could support most loading cases without the need for the metal frame. However the metal frame allows you to adjust the stage independently from the frame. This is important if you want to achieve a very flat surface. It also allows you to attach additional features, such as lighting, to the stage.

If you don't have the facilities to manufacture a Perspex stage yourself there are a number of companies which provide cut-to-size plastics who will also drill any required holes to your specifications. In the UK ThePlasticPeople.co.uk offer this service in a vast range of materials including clear acrylic (Perspex) as well as a large selection of other colours.

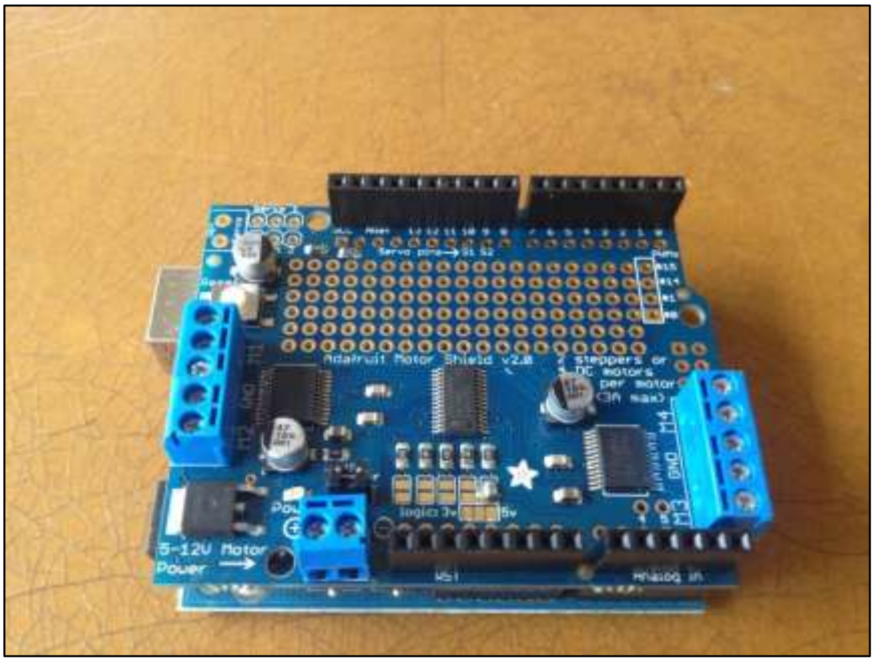
In the OpenLabTools microscope

1	Perspex stage	ThePlasticPeople
4	3D printed M3 4mm spacers	OpenLabTools
4	M6 Threaded rod 150mm	Technobots
14	M6 hex nuts	Technobots
14	M6 washers	Technobots
1	M8 threaded rod 120mm	Wickes
2	M8 hex nuts	Wickes
2	M8 washers	Wickes
4	M3 20mm caphead screws	Technobots
4	M3 hex nuts	Technobots
4	M3 Washers	Technobots

Control system

Microcontroller

We chose to use an Arduino microcontroller to control the linear translation stage in our microscope. Arduino is a fantastic open source platform for automated projects, and there is a phenomenal amount of support online. Any of the Arduino boards are capable of running the stepper motors, however if you wish to use an Adafruit stepper shield (which I would definitely recommend!) you will need an Arduino UNO, Leonardo, ADK/Mega R3, Diecimila or Duemilanove.



Alternatively you could use a Raspberry Pi to control the stage. There isn't currently a dedicated motor shield for the Pi but you can easily build your own driver circuit using an L293D driver IC. Adafruit.com has an excellent tutorial on this here. Note that if you wish to use bipolar stepper motors, as we have, you have to use the L293D chip: the alternative ULN2803 will not work.

If you do choose to use an Arduino with an Adafruit stepper motor shield, please see the main step-by-step guide for details on connecting the shield and programming the microcontroller. Additionally the OpenLabTools project has detailed source code for controlling the linear stage for our microscope project on our Github repository.

In the OpenLabTools microscope

1	Arduino Uno R3 (updated version of the Duemillanove)	Adafruit
1	Adafruit Motor Shield V2	Adafruit

Endstops

Endstops are microswitches which detect when the stage has reached the top or bottom of its range. You should factor these into your design to protect your setup and any delicate equipment. It is relatively easy to keep track of the number of steps your motor has taken, so it is possible to get away with a single microswitch which is used as an anchor point. However the steps aren't 100% precise and the stepper may be liable to skip steps, especially under high torques, and so it is recommended to use a pair. There's nothing special about the microswitches, we used 14mm lever microswitches from Technobots. However these are quite chunky, and you can get really mini ones from Maplins.

In the OpenLabTools microscope

1	3A 14mm Microswitch	Technobots