# IMPLEMENTATION OF CLOUGH42 ELS ON MYFORD S7

## Background

I have to admit to coming late to the series of YouTube videos by James Clough about his project to create an electronic leadscrew for his lathe. His videos give a complete series of step by step descriptions on the concept and his progress. The presentations are very professional and he is clearly a perfectionist in his approach. (Something that I identify with).

Despite having trained on screw cutting in my very distant past, I have never cut a screw thread with my Myford Super 7 Big Bore lathe. Why you might ask ? Because it looked too complicated to work out the gearbox settings. James' project looked like an ideal step forward in solving my procrastination. I recommend you put aside some time to watch the series (and his other videos) before trying to make sense of what follows.

A word of warning that if you buy all the items needed for this project you could be looking at a budget of £500 or thereabouts. This figure will depend on whether you choose a servo or stepper solution.

There is nothing that is overly complex about implementing the ELS. It is mainly an assembly activity but it does require some electrical knowledge. The main challenge is deciding how you couple the encoder and the servo/stepper to the lathe being modified.

## Concept

The ELS consists of four key constituent parts : -

| | | |
|---|---|---|
| A rotary encoder | - | that measures the RPM of the lathe spindle |
| A control panel | - | that allows selection of the screw thread or feed rate required |
| An electronics box | - | that interprets the panel settings to provide a stepper drive |
| A stepper motor | - | that drives the leadscrew on the lathe |

The rotary encoder has to be coupled to the spindle gearing from which it derives a quadrature signal. This information is sent to the microcontroller in the control box.

The control panel allows the selection of both Imperial and Metric screw threads. It sends this information to the microcontroller which in turn commands the stepper/servo. The control panel can also generate simple feed rates without the screw cutting option being enabled. The control panel also acts as a tachometer to provide a continuous readout of the spindle RPM value.

The electronics box contains the microcontroller and the stepper driver unit together with power supplies for these two items and all the associated wiring and connectors.

The stepper motor has to be coupled to the leadscrew in some way or other to give either a direct drive or a ratio drive. Note also that the stepper can also be a hybrid servo device which has some advantages and this is the route I chose. This is the same device that James opted to use.

There is a good pictorial image of the system and how it is inter-wired is shown on the Clough42 GitHub site.

## IMPLEMENTATION

The presentation by James' is very clear regarding the processes needed to build the system. There is also excellent supporting documentation via his website and via GitHub. I could repeat this information but there is little advantage in doing this as his video pictures speak a thousand words. What I will focus on is the things that confused me and that challenged me. These boil down to two aspects. Mounting the encoder on the Super 7 spindle gearwork and mounting the servo on the leadscrew. Before we get to those aspects here are some of my experiences with my electronics box and the control panel together with some images.

### Enclosure

I used a standard enclosure from UK supplier RS Components. This enclosure is supplied with a removeable panel on one end for ease of mounting all the interface connectors. My blogsite has a ZIP file for download that contains the Fusion 360 3D models that I created to manufacture the ELS. One of these files is the layout and CAM details for machining this end panel. Note also that as delivered the panel is held in place using large self-tapping screws. I replaced these with M5 Nutserts for ease of repeated fixing in place. The enclosure is also delivered with a demountable baseplate that allows all the component parts to be test positioned on the plate prior to drilling the fixing holes. I drilled and tapped the base place mounting holes to M3 size. No CAM for this just pencil markings onto the base plate through the mounting holes on each component. I did mount the microprocessor on a sub plate for ease for taking in and out.

My experiences as a wireman taught me that the use of Hellerman sleeves makes for an easy life when wiring large assemblies. I also used the Hellerman sleeves as insulator sleeves on the various solder connections to the plugs and sockets. If you use Hellerman sleeves you will need an associated pair of expanding pliers. The wiring is clearly shown in James' diagram mentioned above. I used small bore heatshrink sleeving to insulate the soldered connections inside the external cable mating connectors.

The use of colour coded instrument wires reduces wiring confusion and potential errors. You can buy packs of ten colour instrument wires of 1m length from EBay and these are very convenient for this project.

There is one fiddly task. Three connections to the control panel and to the microcontroller assembly require making off a 5 way 0.1" pitch header (generically known in my engineering days as Molex connectors). Unless you have the special tool for crimping these connections you must resort to hand soldering and pseudo crimping with pliers. This is a functional solution but is tedious until you master the technique. Unfortunately, you have to buy a bag of 100 pieces of the inserts from RS but at least you will have stock for the next project. Note that the headers on the microprocessor board do not have bump polarisers so watch the orientation.

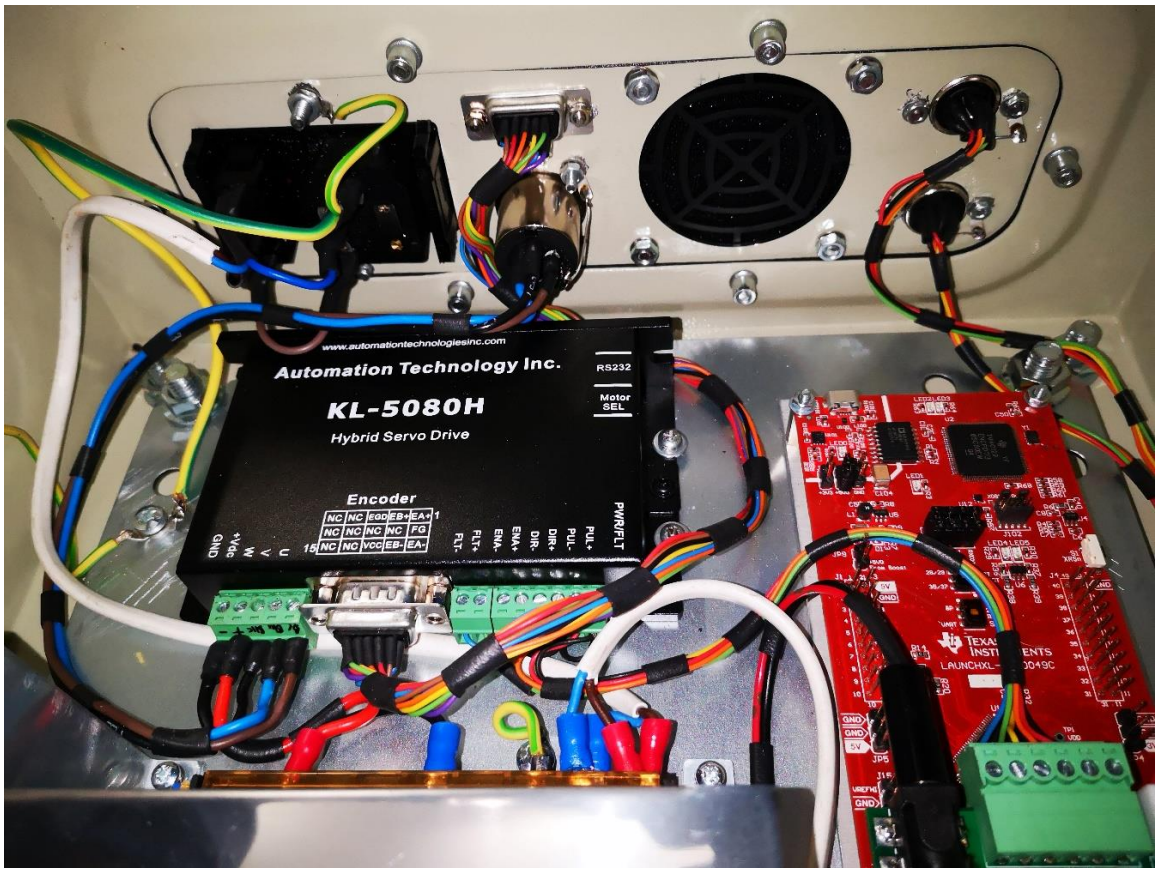
Old habits die hard and I made sure that my earth wiring bonded the shells of the connectors to the connector mounting panel, this bonded this panel to the enclosure body and the body to the component mounting panel.

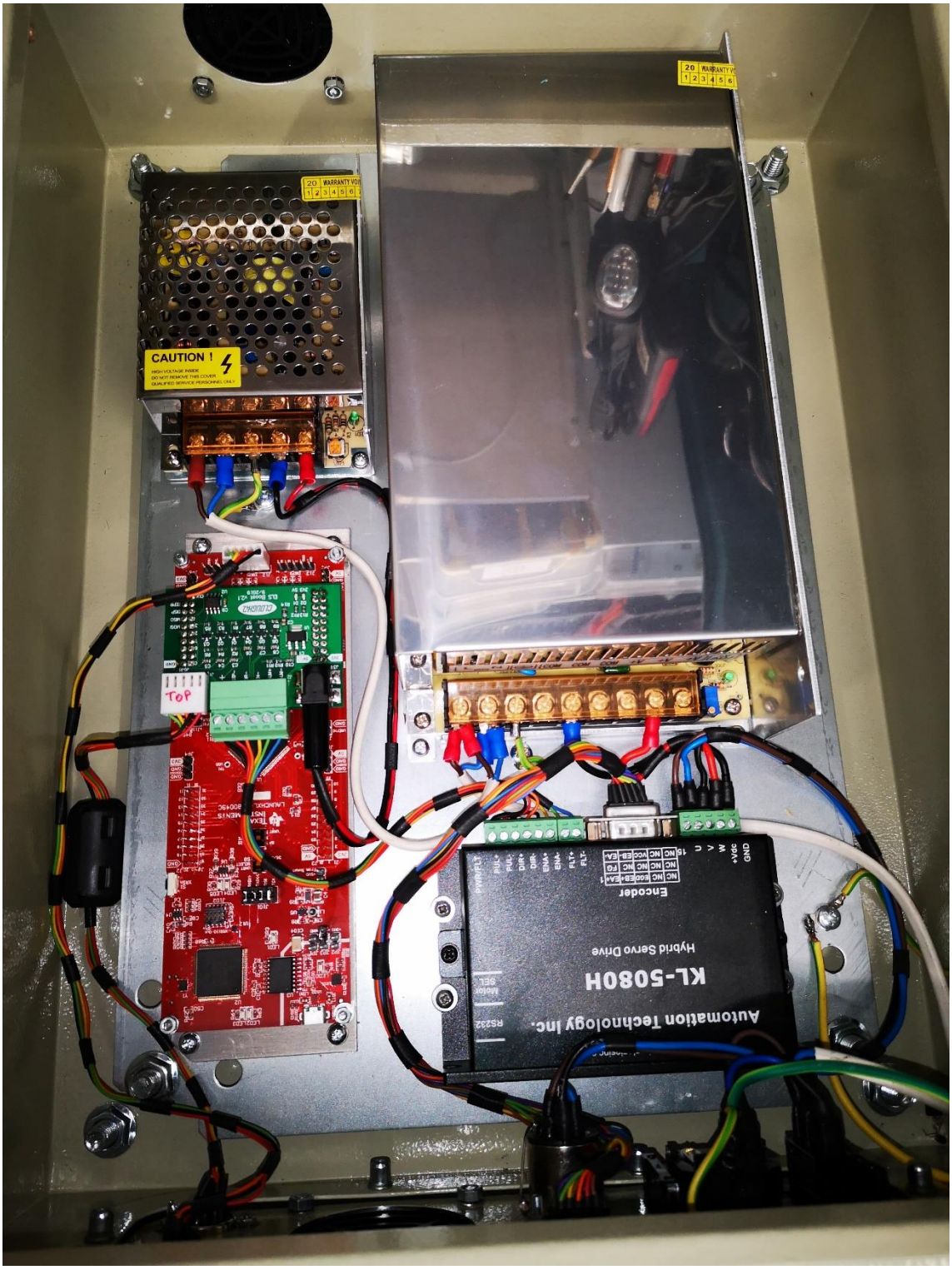
The 15way high density D Type jumper from the servo controller to the box wall is a potential wiring confusion. Use coloured wires and match the numbers on the shell at each end.

I added an air vent on the top and bottom of the box to allow air to circulate. James mounts a fan on his enclosure but my tests so far indicate that there is very little heat dissipated by the components in the box.

The following pages have images of the electronics enclosure.


ENCODER
DISPLAY
SERVO


www.automationtechnologiesinc.com
Automation Technology Inc.
KL-5080H
Hybrid Servo Drive
R5232
Motor SEL
Encoder
PWR/FLT


CAUTION !
TOP
KL-5080H
Hybrid Servo Drive
Automation Technology Inc.
Encoder

## Control Box

James offers a kit of the display board, front panel, red filter material, flexible buttons and a 3D printed spacer to correctly distance the panel to the display. This is quite low cost as a kit but the freight charge to the UK doubles the cost so be aware of this.

I did not have the specified BudBox to mount the display panel into but did have a larger 'Eddystone' style diecast box. In the ZIP file you will find the 3D model of the front panel machining to match the display module. I painted it Myford blue and mounted it on the top cover as shown below.


## Mounting the Encoder

I admit to spending a lot of time just looking at the gearwork behind the Myford spindle end cover. After a lot of such pondering I had a plan and I decided to order 48 and 24 tooth 5mm wide timing pulley and 400mm and 450mm timing belts from Bearing Boys.

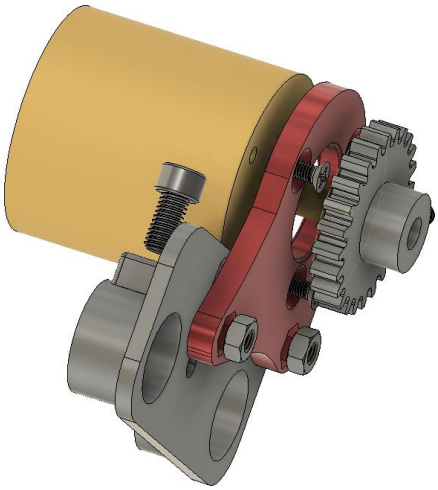
In preparation for the delivery of these I began to strip gear wheels from the spindle gearing. The Super 7 has a forward/reverse and 'disconnected' lever that has two idler wheels. These drive the next gearwheels in the train and these gears sit on stub shafts. As I removed these gears from the shafts it dawned on me that I could use these shafts to make a mounting for the encoder to mesh with the idler gears. On checking I found that the idler ran at the same speed as the spindle. This was even more encouraging. My order on Bearing Boys looked like it was wasted.

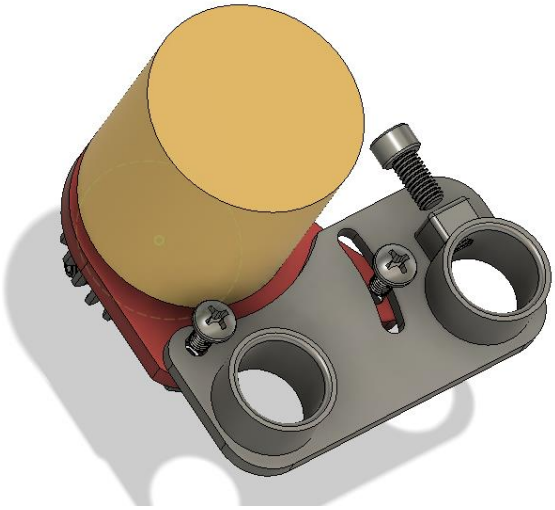
Using Fusion 360 I modelled a mounting that picked up on the two shafts and had fixings for a swing plate that would hold the encoder and allow the mesh into the spindle gear idler to be adjusted. I decided I would create a 24-tooth prototype gear wheel in Fusion and 3D print this for test purposes.

The gearwheel would have to match the idler and I had no idea what the profile or module was for the wheel I was trying to match. I had measured the outer diameter of the wheel as 33mm and I knew it had to have 24 teeth so with some back calculating I could derive the modulus.

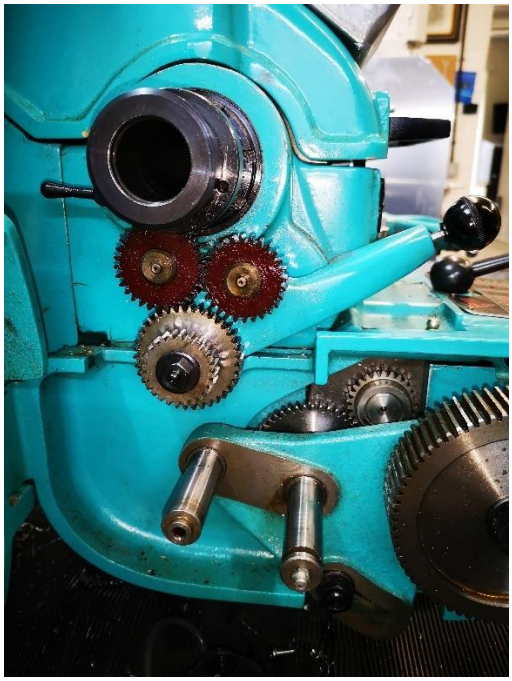
Fusion 360 has a very useful script to allow gearwheels to be drawn as 3D models. Using this scripting engine and looking at the graphical results in comparison with the gear on the lathe led me to believe I needed a 14.5-degree pressure angle, a root value of 0.75 and a backlash figure of 0.15. I cannot be sure if this is totally correct but the 3D printed wheel worked fine (albeit a bit noisy) and the mounting also gave me enough back and forth swing to mesh the two wheels together.

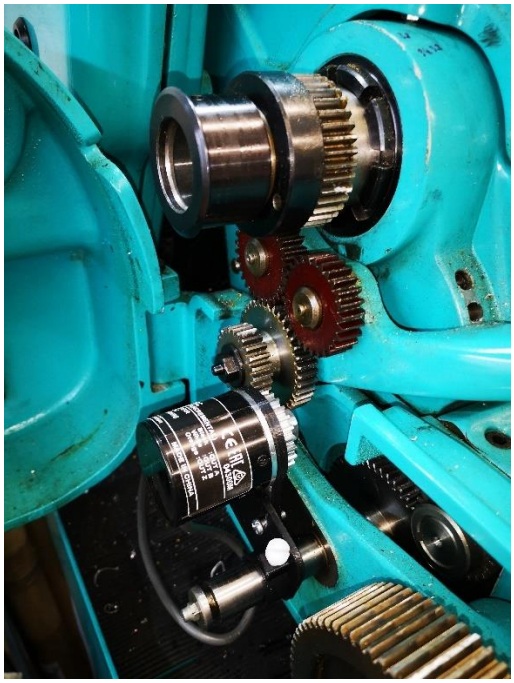
Here some Fusion 360 screenshots showing the encoder mounting bracket and swing plate. The grub screw is a nylon screw that clamps the bracket in place on one of the stub shafts. I created M4 nut profile cavities in the swing plate to grip the nuts for ease of tightening.





Here is the complete assembly mounted in place in the spindle end enclosure. The mounting is such that the end cover still closes as before with no interference from the encoder assembly.





**Driving the Leadscrew**

As with the encoder mounting there was a further session of standing and looking at the lathe and how this might be achieved. The options were to dive in and further strip back the gearbox at the spindle end or work on the tailstock area. I was reluctant to strip more from the lathe mechanics than I had already done with the encoder so focus moved to the tailstock end of the leadscrew.

James used a 2:1 ratio on the servo coupling and I decided to mimic this. To prove the concept, I modelled a 62-tooth gearwheel that was a snug fit on the leadscrew winding handle sitting over the scale diameter together with an associated 31 tooth wheel to fit on the servo. I mounted the servo on a simple bracket and meshed the 31-tooth with the 62 tooth gears. Using toolmakers clamps I gripped the servo in position using the rigidity of the side wall of the Myford coolant tray.

I ran up the system and it seemed to work successfully with one or two side issues. Firstly, the control panel readout was jittery and secondly the mathematic calculations seemed to be wrong.

On the control panel I could stop the jitter when I held the control panel cable with my hand. On this basis it was clear it was an EMC issue. I fitted a split ferrite (RS 811-9109) on the control panel wiring inside the display box and also inside the electronics enclosure and the problem disappeared.

The mathematics were not quite so straightforward and led to re-runs of James' video on programming the TI board and also re-programming the servo controller. This led me to believe I needed to change the setup on the servo driver unit. There appears to be two versions of the KL5080H on the market, one with switch set ups and one with a RS232 connection. Mine was the RS232 version and this required a connection via a RJ12 connector (6 pin RJ11) to a RS232 port on the computer. Such a dongle can be bought ready to go from Automation Technologies but it is a simple thing to make up. The connections are detailed at the end of this write up. The software for programming can be downloaded from AT's website and it purports to be Leadshine originated. I loaded the code and changed the steps value from 4000 to 1000 and re-saved. There is a good video on the AT website on programming the device. This is the only parameter I changed. The dongle wiring is detailed at the end of this article.

**Programming the TI Microcontroller**

I did find programming the TI board a bit frustrating. James' video is very good but I did get bogged down trying to mimic it. I admit to clearing out the CSS a couple of times before it all sunk in.

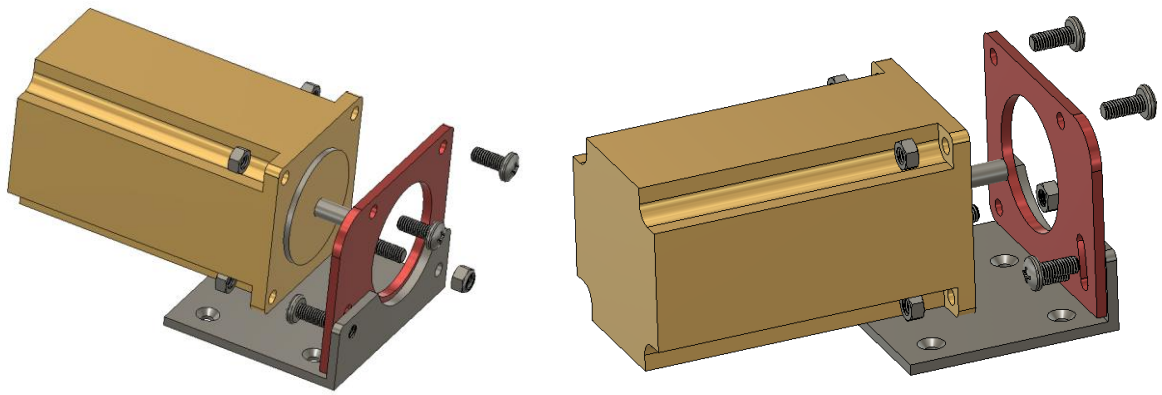
When I first downloaded the CSS it delivered version 10.x.x and this did not seem to want to accept the GitHub code provided by James. I cleared it out and reloaded version 9.x.x and this seemed to work fine. Once I had got used to the Hammer, Spider, Continue, Terminate buttons things made more sense and I got the code to load and work. You have to do the Continue button to complete the loading.

If you watch James' video on programming the TI board you will see how he changes the configuration.h file to suit the devices in use. I first changed the leadscrew TPI value to 8 turns per inch to reflect the Super 7 value. I also changed the steps size to 1000 and the microsteps to 2 to reflect the servo settings and the 2:1 ratio of the leadscrew drive. Note that this will be different if you are using a conventional stepper and controller.

Having completed these changes, I re-ran the system. I placed a piece of reflective tape on the leadscrew for my handheld tachometer to read. If you set up the control panel to cut an Imperial thread with the same TPI value as the leadscrew on your lathe then the tacho readout on the control box will match the tach reading of the leadscrew. This is a simple test that all is well.

**Engineered Servo Mounting**

After proving the system was working, I set about engineering the mounting of the servo and as luck would have it my Bearing Boys order came good. I decided to mount the servo on the rear of the Super 7 bed but needed to be careful on positioning as I have a long DRO scale using all the available standard holes on the bed rear. A new bracket was made and new holes drilled and tapped in the bed rear face. The servo was mounted on a second 'swing' plate very similar to how I had mounted the encoder with the hinge point on the bottom rear corner hole as shown below.



I fitted the 24-tooth gearwheel on the servo. The gearwheel as supplied needed a ¼" centre hole drilling and a M4 grub screw hole cross drilled. The leadscrew tension/slop as originally installed depends on the handwheel pressing against the face of a bearing in a block on the front face of the lathe bed. The handwheel has an extended shank with a 4mm cross slot that engages with a cross pin in the leadscrew.

I removed the handwheel (still with my 3D plastic gear prototype fitted). I drilled and bored the 48 - tooth gearwheel to 11.05mm. I skimmed the 'rear' boss of the gearwheel to around 25mm diameter such that it would slide onto the end of the leadscrew and be as close to the end of the bed as possible. This still allowed the belt drive to the servo sufficient clearance. I now needed a replacement bush to engage with the cross pin on the leadscrew and 'fill the gap' to butt onto the gearwheel. This was satisfied by a 20mm length of 25mm diameter brass rod. The 4mm cross slot was machined in this to 5mm depth. I chamfered the bearing / slotted end of the bush to 45 degrees.

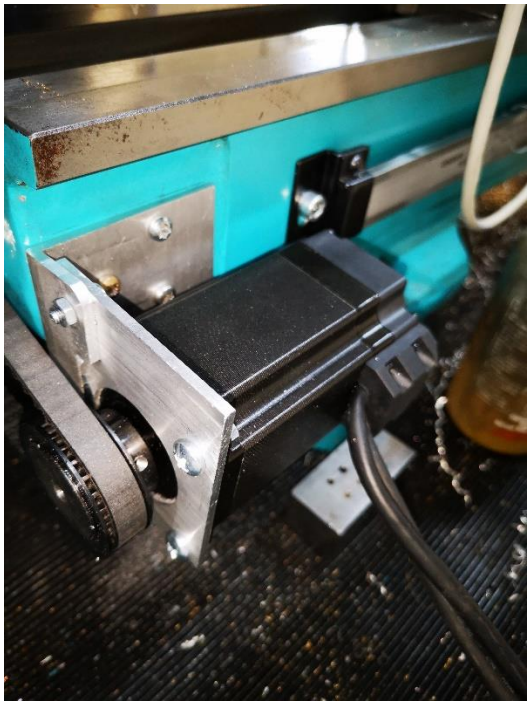
I now had the bush and gearwheel in place on the leadscrew but no drive between the two items and no room to fit the retaining nut at the end of the leadscrew.

The gearwheel outer face was machined back with a 30mm wide cavity to allow more room for the retaining nut to grip. Once this was completed, four 3.3mm holes were drilled on a 19mm PCD on the face of the newly machined nut cavity on the gearwheel. These holes were then spotted through onto the face of the brass spacer bush while still mounted on the leadscrew. The bush was then removed and these four holes drilled and tapped M4. (Note I did this in steps, a single hole to start with so I could lock the position of the gearwheel and the bush, then the diametrically opposite hole and locked this in place and then the other two holes).

The four holes in the gearwheel were enlarged to M4 clearance and countersunk to take 4 x 25mm x M4 screws which would lock the brass bush to the gearwheel.

The retaining nut was still protruding more than ideal and was not meshing with the nylon grip section so I skimmed the nut to compensate. An alternative would have been to make the nut cavity deeper on the front of the gearwheel.

This completes the mounting of the servo and the drive to the leadscrew. On the next page are some images including the first trial drive with the printed gear on the leadscrew manual handle.

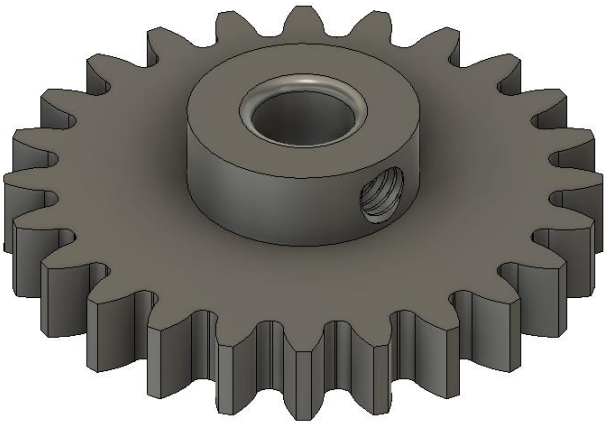


You can see the brass spacer boss and the 4mm cross drive slot.

You can just see inside the nut cavity the four locking countersunk head screws that fasten through the gearwheel and into the brass spacer block.

## Replacement Encoder Gear

With the system now working I used the 3D model of the encoder drive wheel to create a CAM program to create an aluminium replacement for the 3D printed version. The resulting gearwheel appears to run quieter than the plastic one.



## Closing Comments

There is one big advantage that I did not spot when I chose the encoder coupling method. Because it directly drives from the forward/reverse/off idler I can use this lever to impact on the ELS operation. The encoder 'knows' which direction the idlers are rotating and this acts as a forward/reverse action on the electronics which mimics what the control panel does. It also helps with the cutting of Metric screws (see James' video explanation of this). By putting the idler lever to the mid/off position the encoder stops rotating and the ELS is disabled. Any residual noise (which is very low level) from the encoder train disappears and the spindle runs quietly on its own. I can now listen to Springsteen easier.

Likewise, there is an advantage using the hybrid servo driving combination. If for any reason the ELS meets a user induced cock up that blocks the movement of the carriage, the servo drops steps and returns an error flag which stops the drive. I hadn't really thought about this having been locked off into thinking that the screws locking the drive to the brass block could act as shear bolts if there was a disaster.

All in all, this has been an interesting project that has enhanced my Super 7. Thank you to James for stimulating the activity and for giving such a vivid, clear description of 'how to do it'.

## Myford Owners Caution

Other Myford owners should be aware that the methodology I have used on my Super 7 Big Bore will in general work for other Myford models but gear trains and leadscrew characteristics might differ.

## Costing Sheet

| Electronic Leadscrew Costing and Purchasing Details | | | | | | |
|---|---|---|---|---|---|---|
| **Part** | **Part Number** | **Source** | **Qty** | **Each** | **VAT/Duty/Carriage** | **Total** |
| **Main Components** | | | | | | **336.49** |
| TI Microprocessor | LAUNCHXL-F280049C | Mouser | 1 | 51.47 | | 51.47 |
| Rotary Encoder | Omron 1024P | Ebay | 1 | 23.90 | | 23.90 |
| Hybrid Servo | KL-2N-1000 | Automation Technology | 1 | 148.78 | 28.15 | 176.93 |
| Servo Controller | KL5080H | Automation Technology | 1 | | | 0.00 |
| Control Panel Kit | | Clough42 | 1 | 43.70 | | 43.70 |
| Servo Interface PCB | | Clough42 | 1 | 19.03 | 21.46 | 40.49 |
| **Power Supplies** | | | | | | **53.98** |
| 48V @ 10A | | Ebay (Ebuyer) | 1 | 45.99 | | 45.99 |
| 5V @ 5A | | Ebay (Ebuyer) | 1 | 7.99 | | 7.99 |
| **Enclosure Hardware** | | | | | | **103.40** |
| Wall mounting box | 7755798 | RS | 1 | 37.50 | 7.50 | 45.00 |
| IEC mains socket | 2110985 | RS | 1 | 5.98 | 1.20 | 7.18 |
| 6 way DIN fixed skt | 490905 | RS | 1 | 2.05 | 0.41 | 2.46 |
| 6 way DIN free plug | 491027 | RS | 1 | 2.40 | 0.48 | 2.88 |
| 5 way DIN fixed skt (180 degree) | 7863423 | RS | 1 | 2.05 | 0.41 | 2.46 |
| 5 way DIN free plug (180 degree) | 7863441 | RS | 1 | 2.77 | 0.55 | 3.32 |
| 3 way XLR fixed socket | 6660598 | RS | 1 | 3.05 | 0.61 | 3.66 |
| 3 way XLR free plug | 6660516 | RS | 1 | 5.05 | 1.01 | 6.06 |
| 15 way Hi density D Type Socket | 6740947 | RS | 1 | 0.64 | 0.13 | 0.77 |
| 15 way Hi density D Type Plug | 6740953 | RS | 1 | 0.69 | 0.14 | 0.83 |
| 60mm fan vent | 7374083 | RS | 2 | 1.53 | 0.31 | 3.37 |
| 5 pin 0.1" Molex socket | 6795322 | RS | 10 | 0.14 | 0.03 | 1.40 |
| Molex hand crimp pins | 433084 | RS | 100 | 0.13 | 0.03 | 12.73 |
| Instrument wire | Pack 10 x 1m colours | Ebay | 1 | 2.99 | | 2.99 |
| Hellerman sleeves 1.25mm | 1707213 | RS | 1 | 2.00 | 0.40 | 2.40 |
| Hellerman sleeves 2.5mm | 1707235 | RS | 1 | 2.00 | 0.40 | 2.40 |
| Heatshrink sleeves | | Ebay | 1 | 2.00 | | 2.00 |
| 2.1mm DC free socket | | Ebay | 1 | 1.50 | | 1.50 |
| **Pulley & Belt** | | | | | | **30.22** |
| 450mm 5mm timing belt | 450-5M-09 (#6060) | Bearingboys | 1 | 3.10 | 9.48 | 12.58 |
| 24 tooth 5mm pitch pulley | 24-5M-09 (#21216) | Bearingboys | 1 | 5.93 | | 5.93 |
| 48 tooth 5mm pitch pulley | 48-5M-09 (#21224) | Bearingboys | 1 | 11.71 | | 11.71 |
| **Misc** | | | | | | **13.34** |
| Programming Lead for Servo | | Ebay | 1 | 10.00 | | 10.00 |
| Ferrite clip on filter | 8119109 | RS | 2 | 1.52 | 0.30 | 3.34 |
| | | | | | **Total Spend** | **537.43** |

Notes

1. The costing is based on using a hybrid servo and controller and not a stepper and controller so this does inflate the price.
2. The use of RS Components is a convenience of a one stop shop approach and I am sure prices could be nibbled at by judicious buying elsewhere.
3. The power supplies used are very over specified for the application and could be reduced in rating.
4. Buying James' control panel kit and interface board was a convenience but at the cost of heavy importation costs to the UK.
5. If you buy the servo from Automation Technology then don't forget to add the RS232 programming dongle to your order. (But it is easy to make one up).
6. Bearing Boys are my go-to source for bearings and gearwheels but there may well be lower cost sources.

**RS232 Programming Dongle**

# RJ11 Connector for RS232 Communication

## RJ11 Connector


**Applied To:**

**Stepper Drive:**

**DM432C, DM442, DM556, DM856, 3DM683, AM882, DM1182, DM2282**

**Servo Drive:**

**DCS810, DCS810S, ACS306, ACS606, ACS806**

| Pin | Signal | Description | I/O |
|---|---|---|---|
| 1 | NC | - | - |
| 2 | +5V | +5V power only for STU, left it unconnected when connect to a PC serial port | O |
| 3 | TxD | RS232 transmit. | O |
| 4 | GND | Ground. | GND |
| 5 | RxD | RS232 receive. | I |
| 6 | NC | - | - |

## 2. Cable connections to a PC serial port (Male DB9)