ADDING STEPPER POWER FEEDS TO A MYFORD VMB MILL

Background

Having just completed the Clough42 Electronic Leadscrew project (ELS) on my Myford Super 7 I felt somewhat more comfortable around stepper motors and their controllers. I had long hankered to fit a motor on the Z axis of my Myford VMB to eliminate the frustration of manually winding the spindle up and down. The time was now technically ripe to address this.

Concept

I decided that I would build a control box that would have the capacity to drive the X,Y and Z axis but only be able to have one axis functional at a time. There was no desire to go the full route to total CNC as this was adequately provided by my Tormach PCNC440. Full CNC would have involved lead screw upgrades to ball screws on the VMB and was really not justified. Instead I decided I would be happy to be able to select any of the three axes where an associated stepper motor would drive UP/DOWN, LEFT/RIGHT or FORWARD/BACKWARD as appropriate. The control panel would have a rotary switch to select the axis, a speed control and a centre off toggle switch to select the direction. Quite basic in format but functional for my needs.

I subscribe to DroneBot Workshop on YouTube and had seem him demonstrate some simple Arduino code to test drive a large stepper motor. I tried this experiment and had success. I decided I would write an enhanced version of this code which would form the control software mechanism. The resulting stepper drive code would feed to the selected axis stepper driver and onwards to the stepper. Note that the PULSE signal is quite basic and is not shaped in any way being simply a PWM train. It does work and the code is very simple but I freely admit to not be a software writer.

In addition to driving the steppers, the software would also accept an input that would act as a limit switch / emergency stop inhibit of whichever stepper was in motion. This would be a simple loop that if broken would stop all movement of the stepper selected.

Circuit Diagram

The circuit diagram is very simple in its concept. It is shown at the end of this article as is the Arduino code and both need to be looked at to understand the functionality.

The complete circuit is powered from a 12V modular power supply which is stabilised to 5V via an on card regulator. This powers the Arduino and most of the other electronic circuitry.

The Arduino is configured to have five inputs that are normally pulled LOW with resistors. Three of these are the axis select inputs and two are used to select the direction of travel. These inputs are selected by taking them HIGH to the 5V rail. There is some simple software logic to allow only one axis to be selected at any one time and to only allow direction control when the associated switch is not in the centre position. The centre position disables all stepper drivers and saves power.

The limit switch is also a digital input. This has a pull up resistor to 5V but in normal operation the input is held LOW by the external circuitry looping to ground. Any break in the loop would activate the input HIGH. The limit loop would have micro-switches on the axis extremes with normally closed contacts. It would also include an emergency stop button, also configured to be normally closed and in series with the limit switches. A change of state of any switch from normally closed to normally open would trigger the Arduino input HIGH and stop all steppers.

The physical connection to the control box is a 3.5mm audio jack with its contacts wired to be normally closed when there was no mating connector in place. This would allow normal (but potentially risky) operation with no connection to the limit input.

The speed control is a potentiometer across the 5V and Ground supply rail and with its wiper feeding an analogue input on the Arduino. The voltage from the potentiometer is mapped in the Ardiuno such that 0V to 5V represents 0 to 1024 digital levels.

The stepper control consists of three identical circuits. Each uses a 74LS125N Quad Tri-State Buffer of which only three gates are used in each. These devices allow their inputs to be tied together and only pass the input condition to the output when their enable pin is taken LOW. Each group of three gates in each 74LS125 has their enable pins connected together and this common line is taken LOW to allow control of the axis in question.

The three gates on each axis pass the stepper driving PULSE signal, the DIRECTION signal and the ENABLE signal.

The PULSE and DIRECTION signals are digitally created in the Arduino. The PULSE signal provides a series of pulses that are spaced apart depending on the SPEED signal derived from the potentiometer voltage. The DIRECTION signal is a HIGH or LOW signal depending on the position of the control panel Direction Switch.

The ENABLE signal drives a relay. When the relay is not energised its contacts short the ENABLE signal on the stepper driver module to GROUND. This switches the stepper OFF and puts it into a low current state. When the relay is energised the stepper becomes live and that axis is operational. There is a LED across the relay to give a diagnostic status signal to show which axis is operational.

The logic is very simple and the operation appears stable. An input to X,Y or Z from the axis selector switch is sensed by the Arduino and a matching output from the Arduino enables the matching 74LS125 set of three gates. The enable signal can only become active if the direction switch is not in the centre off position.

The stepper motor drivers have two connectors feeding each one. One connector carries ENABLE, PULSE and DIRECTION with an associated +5V reference signal. The second connector carries +48V and GROUND to provide the driving power for the stepper and the four coil drive signals. This 48V supply voltage is provided by a second modular power supply.

<u>Stepper Motors</u>

A 'finger in the air' calculation suggested that each axis of the VMB would need around a 2Nm capable stepper. These are readily available as 1.9Nm NEMA 23 2.8A items and can be purchased paired with an associated driver module (DM942Y). The pricing on these varies with the source. The shaft size is ¼". These were bought on EBay from eu-stepperonline for around GBP40 for a stepper with driver. What's not to like ?

The 12V is fed to the relays and their drivers but only because I had 12V relays in stock. These could just as easily be 5V devices fed from the 5V regulator supply. Alternately the 12V power supply module cold be replaced by a 5V version and with no further need of a 12V source.

Electronics Enclosure

I chose to use a smaller version of the RS enclosure I had used for the ELS. This looked as if it would accommodate all the hardware without a problem. It appeared to do so until I tried to close and fasten the door. The door latch protrudes into the available volume of the box and fouled my chosen 48V power supply. The power supply is overrated for the project and could have been a lower current version which would have avoided this clash. Some minor re-engineering of the fastener mechanism solved the problem but be aware of this as a potential gotcha.

This issue aside, this range of enclosures is a reasonable price and they come with demountable electronics mounting plate and also a demountable connector panel.

I modified the ELS Fusion 360 CAD and CAM files to suit the new connector configuration and also added an air vent cut out to match a customised air intake grill that I had 3D printed. I have mounted a matching 60mm extract fan in the top of the enclosure. This is a bit academic as there is little or heat generated due to the intermittent usage and the fact that the steppers are put into quiescent current mode when not selected. The hole was a chain drill and file job and unless you want the exercise doing this, I would forget about it.

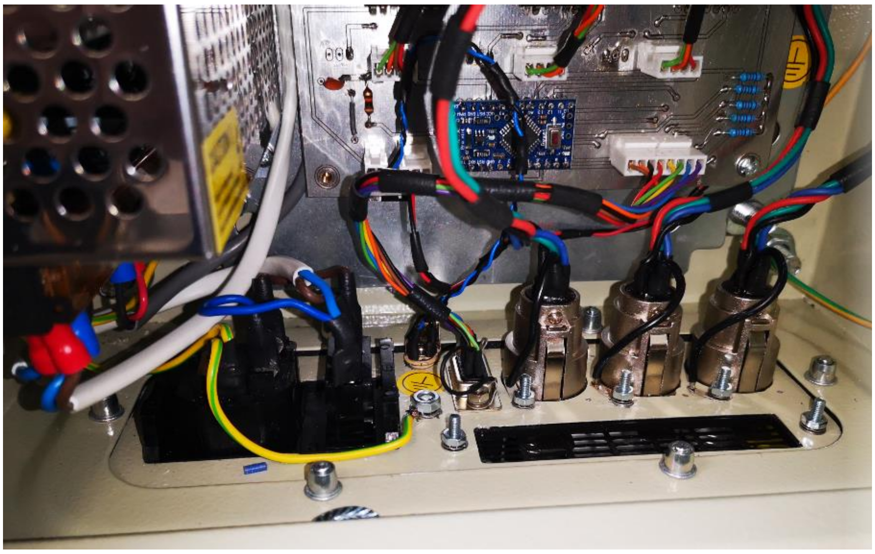
The external cable feeds to the stepper motors are all XLR-4 style connectors. These have good current ratings. The control panel connection is a 9 Way D Type and the AC input is a standard IEC modular integrated unit. As already mentioned, the Limit input is a 3.5mm jack socket.

I designed a printed circuit board (PCB) using Fusion 360 Electrical. This was my first use of the Fusion Electrical module and was a steep learning curve. The resulting PCB artwork manufacturing export package was used to mill the circuit board on the Tormach CNC mill after interpretation of the Gerber and Excellon files by FlatCAM. Details of this process is documented elsewhere on my blog.

One clarification is needed. The relay driver transistor shown on the circuit diagram is stated as being a MUN2214T1. This is equivalent to the DTC114. These are digital transistors with built in resistors. I have a good stock of DTC114 (with 10k resistors) in the TO92 leaded version. I could not find this package format in any of the supporting libraries in Fusion Electrical. There were plenty of examples of the SMD version however. To get round this I have placed three single pin headers around each SMD device to provide an alternative set of mounting holes for the TO92 version instead of the SMD version.

All low current connections were made with Molex 0.1" pitch connectors. The 48V connections are directly wired from the 48V PSU and the four stepper drive signals are also wired directly from the stepper driver module to the XLR-4 connectors on the panel.

Here are some images of the finished enclosure with my first prototype PCB in place.



Stepper Mounting

On the ELS project, I had removed the graduated slip ring dial from the manual winding wheel at the end of the lathe bed and replaced this with a 48 tooth 5mm pitch timing pulley. This pulley was belt driven by a 24 tooth pulley on the stepper. This gave a 2:1 drive reduction. I made a mount for the stepper that fastened from the back of the lathe bed. This has worked very well. Belt and pulleys came from Bearing Boys in the UK.

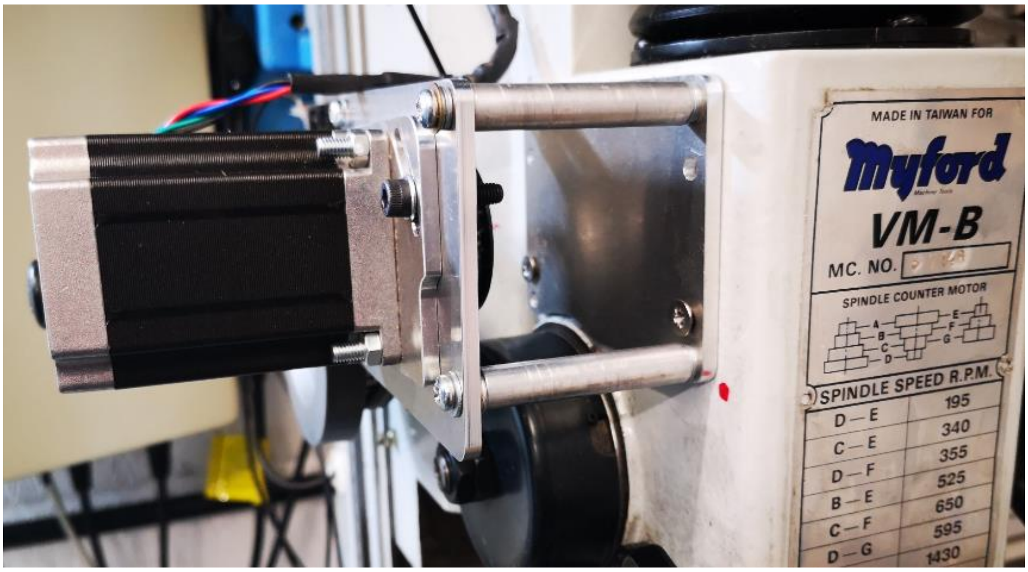
I opted to use the same general method on each VMB axis, namely remove the graduated slip ring on the manual wheels and replace them with 48 tooth pulleys and then screw lock the pulley to the handwheel such that manual movement is still possible with the handwheel. This avoids the need for a clutch or similar mechanism. This does mean that manual movements will generate a back emf into the stepper driver but at a very low level and insufficient to do any damage.

The 48 tooth pulley needs boring out to match the slip ring core diameter and the pulley boss needs turning down so it aligns with the back face of the hand wheel. The other side of the pulley also needs facing so it does not bind on the mill body. Three radially spaced M3 screws lock the hand wheel to the pulley.

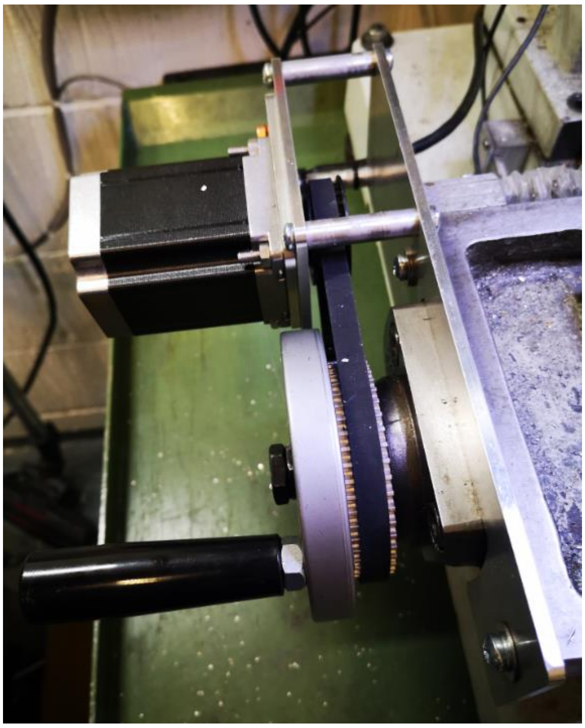
The stepper physical mounting is also similar to that adopted on the ELS. It uses the same three plate mechanism - one plate to fasten to the mill body, one as a standoff plate from the body plate and then a swivel plate holding the stepper that swings and secures against the standoff plate to allow belt tensioning. The standoff plate spacers are cut to align the handwheel pulley with the stepper pulley when the 48 tooth wheel is in place on the handwheel.

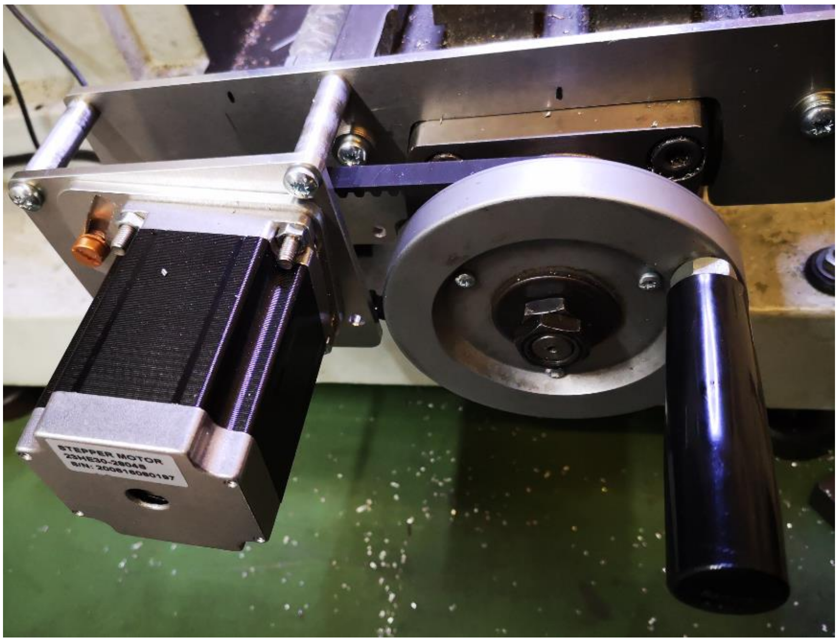
The Z axis went to plan but I had to 'suck it and see' by doing a temporary 'G' clamping in position to optimise the best placement position. The mounting holes were drilled and tapped (M5) into the VMB casting. Only three standoff pillars were used as the fourth one would have fouled the belt path. The assembly was still more than adequately rigid.

The following images show the Z mount structure. The mounting plate can be seen shaped over the original end plate fixture and the M5 mounting screws. The belt tensioning adjuster screw and associated adjustment slot can be seen on the stepper swivel plate.



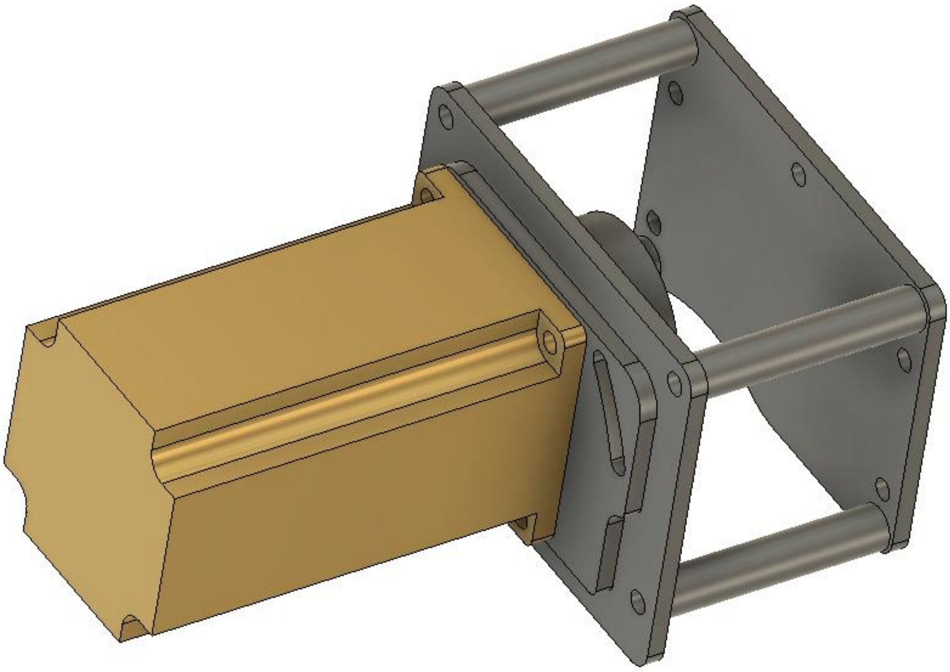
The X axis require a customised plate to fit on the end of the milling table. This is shown in the following two images. Only two M5 screws were used for this and the plate was 4mm cast aluminium. This seems to be sufficiently stable and rigid. With hindsight the stepper is a little too close to the handwheel handle and presents a potential knuckle bruising risk. I think a shaped end knob on the existing handle will solve this.

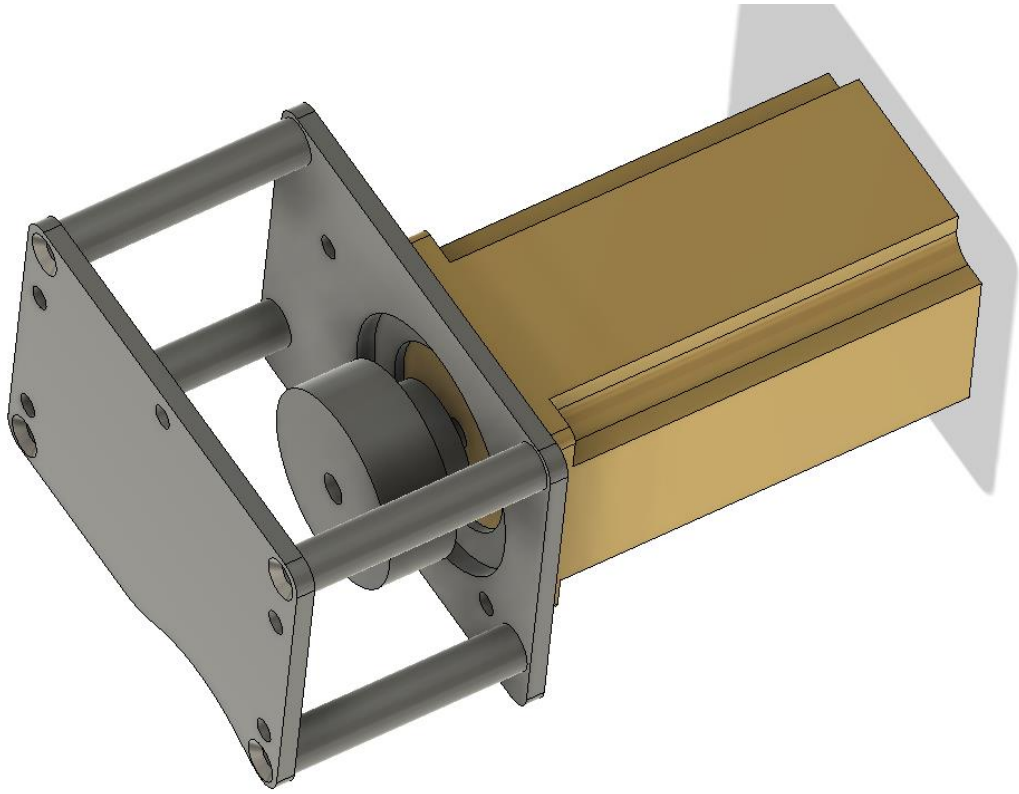




To date I have only installed the steppers on the Z and X axis and images are shown. There is no rush on my part to install the Y axis as this rarely creates a demand for motor control. The control box and control panel are fully populated ready to accommodate this when the incentive to do it arises (i.e. – a rainy day job).

Here are my original Fusion images of the mounting plate concept.





The lower view shows the two fixing points for the hinge plate onto the standoff spacer plate. The lower hole is the hinge mount and the upper hole rides in the slot to adjust the tension on the belt.

Control Panel

The control panel is a simple 3D printed enclosure that holds the axis selector switch, speed control and direction switch (centre off). This is mounted on a simple bent panel on the front of the VMB coolant tray stand. The angle plate acts as the bottom of the enclosure. I used CAT5 cable for the interconnection to the Electronics Box with a 9 pin D Type free plug at the Electronics Box end of the cable. The other end of the cable is hard wired into the Control Panel. The cable enters the enclosure from the rear via the mounting plate and has cable ties as strain relief. The switch was an EBay find.

As can be seen I attempted to 3D print the control panel legend into the box but it did not print very well. It needs back filling with a contrasting colour. This is a very tricky thing to do on a 3D printed enclosure as any flooding of the filler into the 3D striations is a nightmare to clean up.

Costing

A costing spreadsheet is attached at the end of the article.

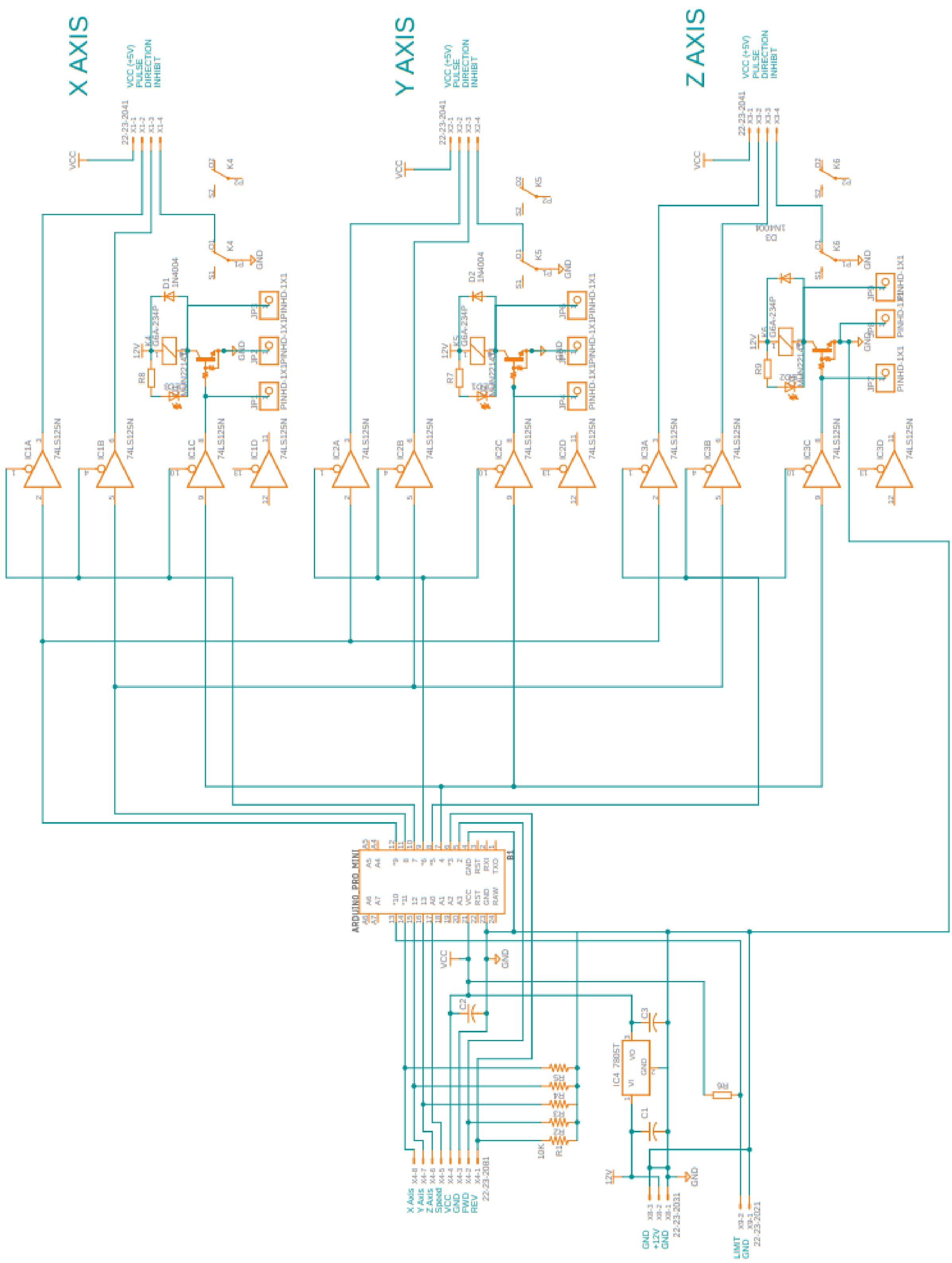
Conclusion

This modification to the VMB has long been threatened. Without having undertaken the ELS project I don't think I would have considered stepper motors for this modification. Watching DroneBot on how to use steppers tipped the balance completely. My manual winding of the Z axis to accommodate my Haimer had always been a frustration. It is now a much more pleasurable experience totally lacking in physical exertion other than flipping two switches and twiddling a knob.

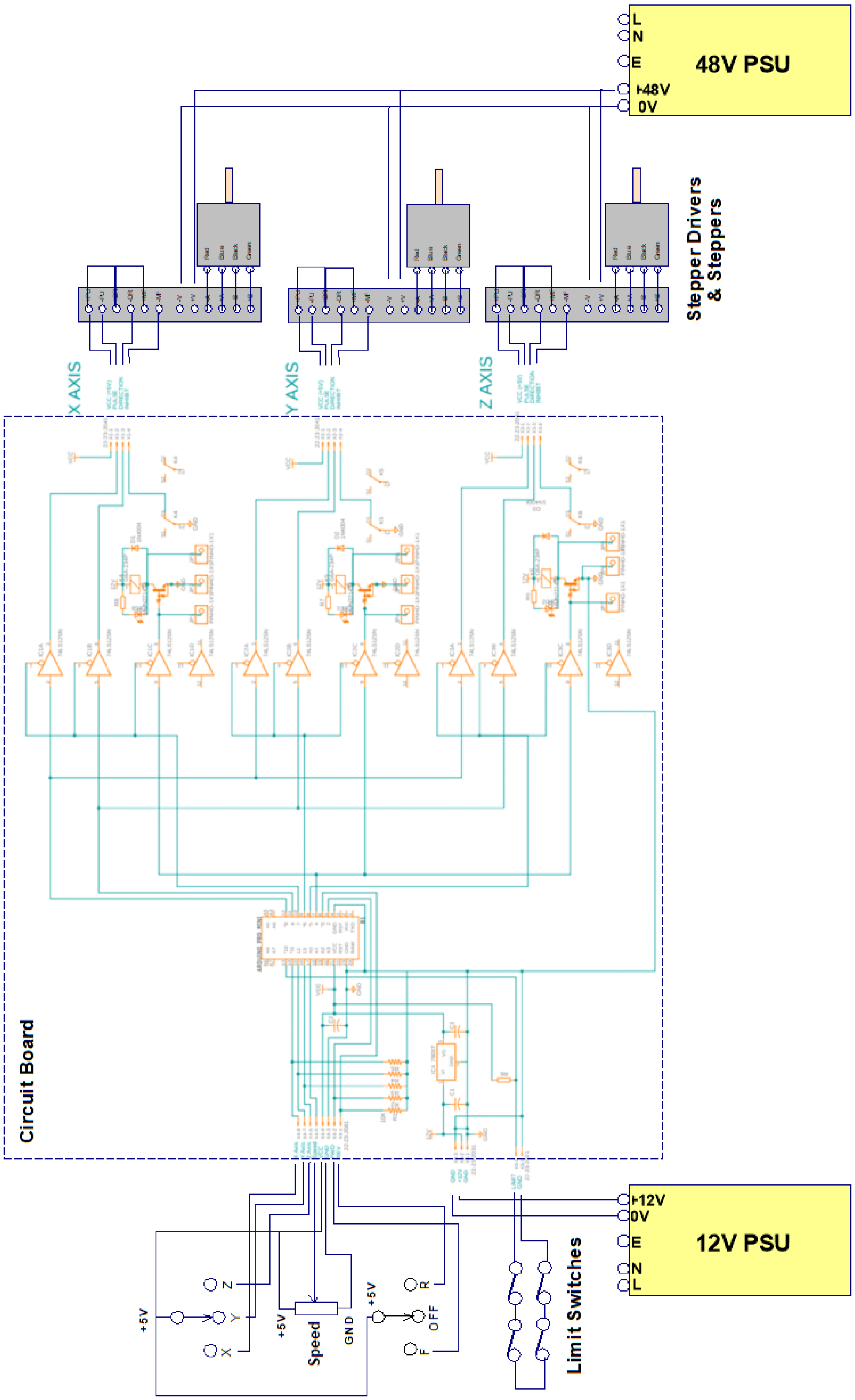
While I have implanted this on the Myford VMB, there is no reason why the installation could not be made to suit any other mill or lathe with only the stepper mounting and coupling as the challenge.

I am more than happy to provide more detailed information on any aspect of the conversion should this be needed. If there is sufficient interest, I could get a batch of printed circuit board blanks manufactured. Please email me on subs999@altrish.co.uk.

PCB Circuit Diagram



Connection Diagram



Arduino Code

Readers should note that I am not a software writer and the code below may well be rubbish but it does work.

```
/*
 XYZ Axis Stepper Motor Driver with Limit Switch Override
 Woody's Workshop October 2020
*/

// Define pins
int reverseSwitch1 = 2;   // Left/Up direction from a taken to +5V from a centre off direction switch
int reverseSwitch2 = 3;   // Right/Down direction from a taken to +5V from a centre off direction switch
int relayDriver = 4;      // relay enables operation of stepper subject to position of reverse switch and limits input, otherwise shorts -MF on stepper driver to ground
int xAxis = 11;           // X axis select input from rotary switch where wiper goes to +5V
int yAxis = 12;           // Y axis select input from rotary switch where wiper goes to +5V
int zAxis = 13;           // Z axis select input from rotary switch where wiper goes to +5V
int zAxisEn = 5;          // Z axis enable output to drive associated quad tristate buffer enable pins
int xAxisEn = 7;          // X axis enable output to drive associated quad tristate buffer enable pins
int yAxisEn = 6;          // Y axis enable output to drive associated quad tristate buffer enable pins
int driverPUL = 9;        // PULSE clock to stepper driver -PU pin (variable mark space ratio dependent on speed pot)
int driverDIR = 8;        // DIRECTION change to stepper driver -DR pin on driver which sits either at +5V or GND
int limits = 10;          // Limit switches input. Pulled HIGH with 10k resistor and goes LOW if limit switch or emergency stop is triggered open circuit.
int spd = A0;             // Wiper of speed setting 10K potentiometer across +5V and GND. Wiper voltage range is mapped to be 0 to 1024 digital value.

// Variables
int pd = 500;             // Pulse Delay starter period for stepper driver pulse waveform

void setup() {
 pinMode (driverPUL, OUTPUT);
 pinMode (driverDIR, OUTPUT);
 pinMode (relayDriver, OUTPUT);
 pinMode (xAxis,INPUT);
 pinMode (yAxis,INPUT);
 pinMode (zAxis,INPUT);
 pinMode (xAxisEn,OUTPUT);
 pinMode (yAxisEn,OUTPUT);
 pinMode (zAxisEn,OUTPUT);
 pinMode (reverseSwitch1,INPUT);
 pinMode (reverseSwitch2,INPUT);
 pinMode (limits,INPUT);

 }
```

```
void loop() {

  if(digitalRead(xAxis)==HIGH){digitalWrite(xAxisEn,LOW);}else {digitalWrite(xAxisEn,HIGH);}
  if(digitalRead(yAxis)==HIGH){digitalWrite(yAxisEn,LOW);}else {digitalWrite(yAxisEn,HIGH);}
  if(digitalRead(zAxis)==HIGH){digitalWrite(zAxisEn,LOW);}else {digitalWrite(zAxisEn,HIGH);}

if(digitalRead(reverseSwitch1)==HIGH){digitalWrite(driverDIR,HIGH);digitalWrite(relayDriver,HIGH);}

if(digitalRead(reverseSwitch2)==HIGH){digitalWrite(driverDIR,LOW);digitalWrite(relayDriver,HIGH);}
  if(digitalRead(reverseSwitch1)==LOW &&
digitalRead(reverseSwitch2)==LOW){digitalWrite(relayDriver,LOW);}
  if(digitalRead(limits)==HIGH){digitalWrite(relayDriver,LOW);}

  pd = map((analogRead(spd)),0,1023,2000,50);

  digitalWrite(driverPUL,HIGH);
  delayMicroseconds(pd);
  digitalWrite(driverPUL,LOW);
  delayMicroseconds(pd);

}
```

I don't think it could be much more simple ….. ?

<u>Project Costing</u> (based on a full purchase of all hardware)

VMB Axis Power Upgrade						
Part	**Part Number**	**Source**	**Qty**	**Each**	**VAT/Duty/Carriage**	**Total**
Main Components						**149.17**
Servo Controller and Servo		eu-stepperonline	3	42.63		127.89
Arduino Pro Mini		Ebay	1	1.98		1.98
74LS125 Quad Tri-State Buffer		Ebay	3	1.10		3.30
DTC144 Digital Transistor		Ebay	6	1.00		6.00
7805 Regulator						
Servo Interface PCB		Homemade	1	10.00		10.00
Power Supplies						**32.63**
48V @ 5A		Ebay (Ebuyer)	1	25.64		25.64
12V @ 2A		Ebay (Ebuyer)	1	6.99		6.99
Enclosure Hardware						**125.17**
Wall mounting box	7755300	RS	1	29.50	5.90	35.40
IEC mains socket	2110985	RS	1	5.98	1.20	7.18
Rotary Switch (4P x 3W)	320708	RS	1	2.46	0.49	2.95
Knob	4638508	RS	1	1.05	0.21	1.26
End cap for knob0	4638542	RS	1	0.54	0.11	0.65
Speed Pot (10k)		Ebay	1	2.40		2.40
Relay (DPCO)	1762865	RS	3	2.79	1.67	10.04
Centre Off DPDT switch		Ebay	1	0.99		0.99
3.5 mm audio jack socket		Ebay	1	1.50		1.50
3.5 mm audio jack plug		Ebay	1	1.50		1.50
4 way XLR fixed socket	5489111	RS	3	4.37	2.62	15.73
4 way XLR free plug	457908	RS	1	2.09	0.42	2.51
9 way D Type Socket	6740947	RS	1	0.64	0.13	0.77
9 way D Type Plug	6740953	RS	1	0.69	0.14	0.83
D Type Shell	484789	RS	1	0.98	0.20	1.17
60mm fan vent	7374083	RS	2	1.53	0.61	3.67
2 pin 0.1" Molex housing	6795322	RS	2	0.14	0.05	0.33
3 pin 0.1" Molex housing		RS	1	0.14	0.03	0.17
4 pin 0.1" Molex housing	6795388	RS	3	0.14	0.08	0.49
8 pin 0.1" Molex housing	6795417	RS	1	0.24	0.05	0.29
2 pin 0.1" Molex PCB header		RS	2	0.21	0.08	0.50
3 pin 0.1" Molex PCB header		RS	1	0.21	0.04	0.25
4 pin 0.1" Molex PCB header	6795596	RS	3	0.21	0.12	0.74
8 pin 0.1" Molex PCB header	6795606	RS	1	0.34	0.07	0.41
Molex hand crimp pins	433084	RS	100	0.13	2.54	15.24
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
4 Core unscreened cable	2012212	RS	10	0.70	1.40	8.40
Pulley & Belt (per axis)						**73.94**
375 mm 5mm timing belt	375-5M-09 (#6057)	Bearingboys	3	2.90	1.74	10.44
24 tooth 5mm pitch pulley	24-5M-09 (#21216)	Bearingboys	3	5.93	3.56	21.35
48 tooth 5mm pitch pulley	48-5M-09 (#21224)	Bearingboys	3	11.71	7.03	42.16
Misc						**45.00**
4mm aluminium plate			3	10.00		30.00
Spacers and fastenings			3	5.00		15.00
					Total Spend	425.92