

Encoder Threading with Mach4 Lathe and the PoKeys57CNC

by Dan Mauch

Photos and drawing by Author

I was excited to hear that threading with Mach4 Lathe and the PoKeys57CNC with an encoder was now working. I just had to try it, since I had mixed success with index only threading in Mach3 Turn. I had tested other software that was supposed to do encoder threading, but they were beta versions and had several serious issues.

From what I have heard, LinuxCNC has had encoder threading for some time, but I have had problems with LinuxCNC in the past. Additionally, I didn't want to have another PC solely dedicated to Linux.

Prerequisites

This is a "general" configuration and setup of Mach4 and the PoKeys.

My article is intended for those with some understanding of Mach and CNC setup. Individual systems may differ from the following:

The journey begins with an explanation of encoder threading. An incremental shaft encoder needs to have an index and channels A and B. The program needs to detect the encoder index signal and channels A and B to electronically gear the spindle rotational speed to the feed rate of the Z-axis, thus maintaining the perfect pitch. If the spindle slows down or speeds up, the encoder feedback to the controller adjusts the feed rate of the Z-axis.

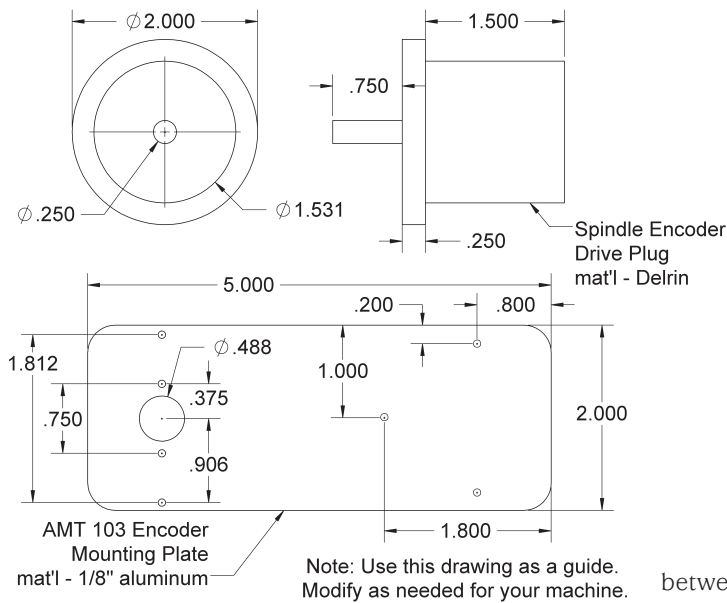
However, on most lathes I've seen it is difficult to mount an encoder. There are two types of

encoders that can be utilized. Type 1 uses an encoder with an external shaft, and type 2 has a hollow disk where a shaft is inserted.

Type 1 is usually connected by a timing belt to the spindle. This method can induce errors due to pitch errors of the timing belt, slack, out-of-round, or poorly manufactured pulleys. Additionally, if there is a pulley ratio involved this adds Pulses Per Revolution (PPR) counts to the settings in the PoKeys 57CNC configuration.

Type 2 eliminates most of the problems with type 1 installations, but adds the problem of how to direct drive an encoder by the spindle. In my case, I have an old Jet 13" x 25" lathe. I machined a plug to fit snugly into the rear of

Figure 1
Drive Plug and Mounting Plate

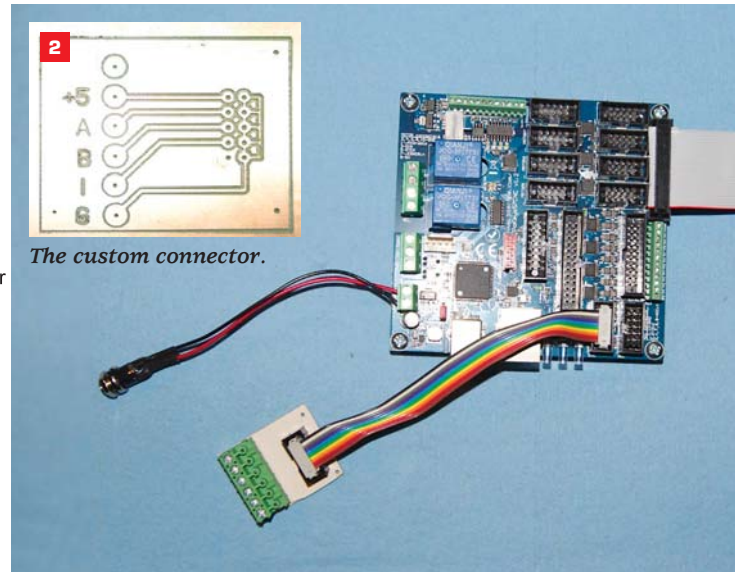


the spindle bore. The plug has a 1/4" diameter shaft that protrudes outward, past the rear gear cover (**Photo 1** and **Figure 1**).

The problem with this design is you need to remove the plug and encoder when working on a long piece of stock that passes through the spindle. In my case, I have 19" between the forward end of the plug and the front of my three-jaw chuck. While the encoder and spindle adapter plug can easily be removed if necessary, I rarely use any stock that long. If needed, long stock can always be sawn shorter, of course.

For my setup I decided to use a CUI AMT 103 encoder, available from www.CNC4PC.com. The encoder is a capacitive type and has an index, with 16 DIP switch adjustable PPR settings. At first, I chose the 48 PPR setting while trouble shooting, but reset it to 385 PPR for threading.

The Pokeys57CNC from Polabs is also available at CNC4PC and is an extremely powerful and complex controller, but one with a low cost. The Pokeys57CNC is a blend



between a general purpose PoKeys device and a motor controller. The device is targeted primarily for controlling up to eight STEP/DIR signal driven motors (stepper motors, servo drives, etc.) in various applications with the addition of powerful PoKeys device features.

The device contains dedicated connectors for motor drivers, pendants (HD44780-compatible), LCD module, etc. In addition, five analog inputs with 12-bit resolution are available. The device also features four galvanic-isolated open-collector outputs and isolated 0 to 10 V analog outputs.

At first I had difficulty in interfacing the PoKeys57CNC to my existing servo system on my lathe. While the insulation displacement connectors (IDC) make for a compact board, I had troubles matching up with my existing servo system until I figured out a few tricks on how to do it.

The easy part was the 26-pin male IDC connector that is preconfigured for most step and direction (S/D) control boxes. On page eight of the PoKeys57CNC manual (www.poscope.com/PoKeys57CNC)

this connector is identified as a DB25 but it is a 26-pin IDC male header. There is a drawing that shows the function of each pin, so it was easy enough for me to make a 26-pin female IDC connector to a male DB25 connector adapter. The pinout matched the pinout of my breakout board for my servo system so I didn't have to change any of the wiring to my C10 breakout board.

The problem was the 10-pin IDC header for the encoder. I didn't have a 10-pin IDC to terminal adapter, although there are many on eBay. I decided to machine one using my Printed Circuit Board (PCB) drill machine that I also use for 3D printing. I removed the 3D print head and installed a motor mount for my 50 year old Dremel moto tool. I laid out the pads and traces using my electronic design CAD program to generate a Gerber file.

Next, the file was imported into CamBam. From there it was exported to Bobcad21 as a dxf file. I then had the program put points where the arc centers were because I didn't want to deal with an Excellon drill file. Finally, I used the CAM feature of Bobcad to create the G-code to dimple the pad centers and machine the out-

3

The PoKeys57CNC, ready to go.

The easy way to do it – the Mach4 threading wizard.

32 or 64-bit laptop or desktop Windows XP, Windows Vista, Windows 7, Windows 8, or 10 operating system

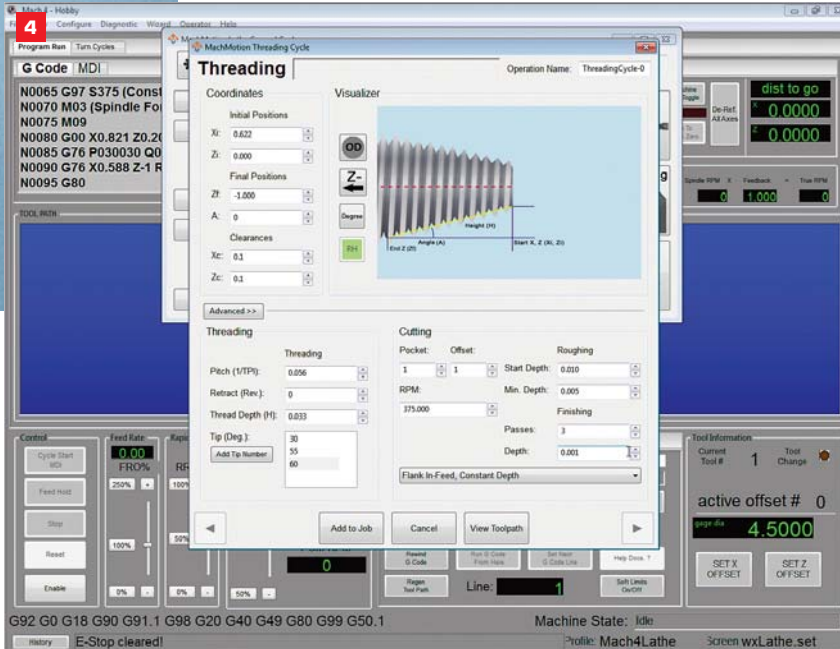
An appropriate external motion controller

My PC is a Win7, 32-bit and I'm using the very reliable Ethernet port. You could use the USB port of the Pokey57CNC, but I haven't tested that feature. Installation of both Mach4 and the PoKeys57CNC was not complicated. Making it work together presented problems, but in the end the solution was simple. I did not have the latest development version of Mach4 Hobby, nor the latest plugin and firmware. Once everything was updated it was pretty easy.

At your PC desktop, select Mach4 Lathe and run it. In Mach4 Lathe, select *Configure*. Next, select *Motors*. In my installation, Motor 0 is the X-axis. You set up the step per unit and acceleration like you did with Mach3, then click on *Apply*. My Z-axis configuration is Motor 2 and you set up the steps per unit and acceleration.

Back in the Mach4 Lathe menu select *Configure*, select *Plugins*, and then select the Pokeys plugin. A new window will open. It should show that the PoKeys57CNC was connected. Highlight the Pokeys57CNC and select *Configure*. The PoKeys configuration menu will appear.

Select *Spindle*. Click on *Enable Spindle Speed Measurement*. Click on the bullet for *Encoder*.



lines of the traces. The machined board is shown in **Photo 2**.

The PoKeys57CNC PCB and my 26-pin female IDC connector to DB25 connector, together with the machined 10-pin IDC header board and cable to the encoder, appears in **Photo 3**.

The AMT103 uses a MTA type connector. I added an encoder line driver because the distance between the encoder and my control box is about eight feet. The magic of the PoKeys57CNC is that the encoder is connected directly to the encoder 10-pin header on the board. The PoKeys then connects to Mach4 Lathe via an Ethernet connection.

Mach4 Hobby requires a dedicated license for the computer that you will be using. The computer needs to meet the following requirements:

The Mach4 plugin for the selected motion controller

2 GHz CPU

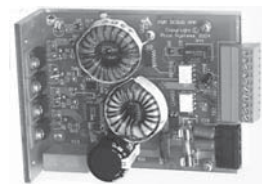
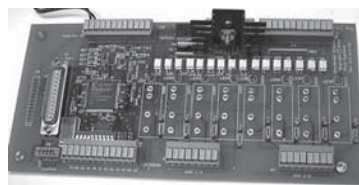
1GB RAM

Video card with 256MB RAM (large G-code files, especially 3D files, will require a video card with 512MB RAM or higher)

Pico Systems

The servo loop is closed all the way back to the PC's CPU for guaranteed accurate positioning.

- Fast Stepper Controller
- PWM Servo Controller
- PWM Servo Amplifier



We now have both Brush and Brushless Servo Amplifiers!
(314) 965-5523 <http://pico-systems.com>

NORTH AMERICAN MODEL ENGINEERING
28TH ANNUAL
EXPO

APRIL
22-23

2017

YACK ARENA
WYANDOTTE, MICHIGAN

- PRECISION MACHINED GAS, STEAM & STIRLING CYCLE ENGINES AND A WIDE VARIETY OF MINIATURE MACHINE TOOLS
- OVER 1,000 MINIATURE MODELS & MACHINE TOOLS ON DISPLAY!
- VARIOUS SEMINARS & DEMOS ALL FREE WITH ADMISSION!

30,000 SQ. FT.
ON DISPLAY!

GENERAL INFO: (810) 599-1618
or namesexpo@gmail.com



www.modelengineeringsociety.com



Enter the PPR of your encoder in the field for *Encoder Tics per Rev*. If you used any pulley ratio then that would be a multiplier of the PPR.

Click on *Enable Threading Support*. Click on *Apply* and then close out *Configure*.

You should now be able to turn on your spindle and see the RPM in the Mach4 Lathe True RPM window. If Mach4 isn't reading RPM, then stop the spindle and click on *Diagnostics* and select *PoKeys Plugin*. You will see the PoKeys status display. Turn the spindle by hand. You should see Pokeys pins 8 and 12 changing state as the spindle is rotated. This indicates that the encoder is working. Pin 13 is the index and even if it is working, Pokeys' status display wouldn't indicate it.

To verify that the encoder index function is being detected, go back

to *Configure*, select *Pokeys*, then select *Spindle*. Change the bullet from *Encoder* to *Index* and then turn on your spindle motor again. You should see the correct RPM.

If in the PoKeys status display you weren't detecting the encoder, then recheck your wiring of the encoder to the 10-pin IDC plug. Page eight of the Pokeys57CNC manual shows that pin 1 is +5DC to the encoder, pin 3 is encoder channel A, Pin 5 is encoder channel B, pin 7 is encoder index, and pins 2, 4, 6, 8, and 10 are ground.

Now for the fun part – threading:

You should first read the Mach4 Lathe manual from www.machsupport.com/wp-content/uploads/2014/05/Lathe%20GCode%20Programming.pdf. There are massive differences compared to Mach3 Turn. Pages 29-32 show the G76 syntax for threading

G-code. It follows the Fanuc OT, 16T, and 18T syntax. It's different from the Fanuc 10/11/15T, which uses a single line G76, whereas Mach4 Lathe requires two G76 lines of code. The next section walks you through an example.

In the first line of G76, P, Q, and R are defined as:

P is a 6 digit code. The first two digits are the number of finish cuts. The middle two digits are the number of leads for pullout, and the last two digits are the thread angle.

Q is the minimum cutting depth.

R is the fixed amount of material for finish allowance.

Here is the confusing part: In the second line of G76, the P, Q, and R are not the same functions as in the first line.

The G76 structure of the second G76 code uses the variables X, Z, R, P, Q, and F, where: