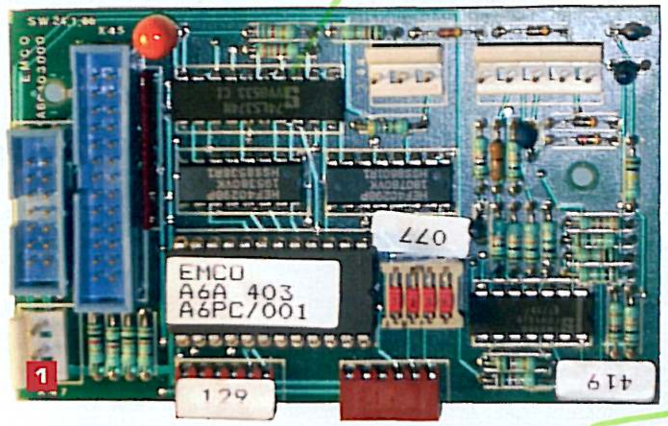
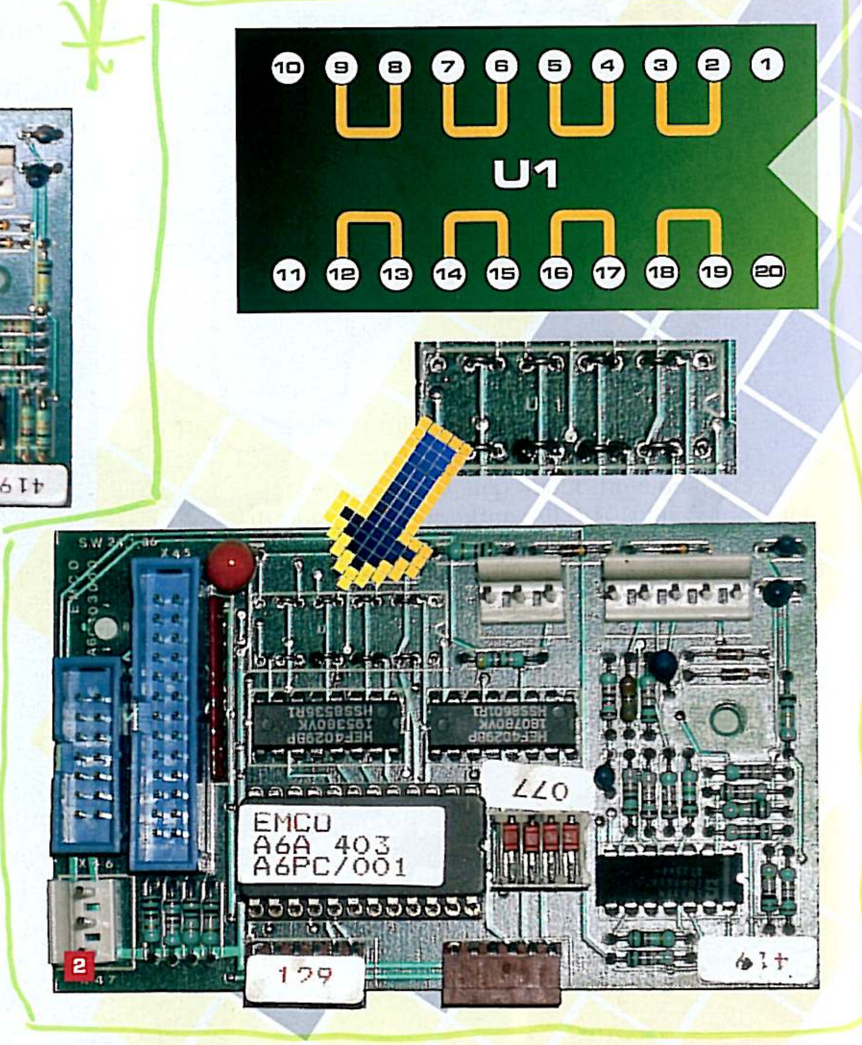
Software Upgrade for the EMCO Compact 5 PC Lathe

by Jerry Gold

Photos and drawings by Author





There are a number of old CNC machines that are in serviceable condition but have outdated control software and/or electronics. A complete retrofit of electronics and software can be expensive, labor intensive, and not always necessary. In some cases, doing a retrofit is possible, if primarily using new software.

I purchased a previously-owned CNC lathe; an EMCO Compact 5 PC that was in relatively good condition but it came with neither its associated software nor dongle. I wanted to get it operational for the least amount of money and modifications to the machine. To get it running, I needed to make a decision between obtaining the original software and dongle, or doing some sort of retrofit.

The Compact 5 PC's original software, which came with the machine, was an old DOS program that used an archaic interface (used function & arrow keys – *not* a mouse). I really didn't want to use the original software, even though it did have some advantages. For example, since it was designed specifically for the machine, there is no need to set or figure out any program parameter values. The software, like many older control programs that run under DOS, will only work on a PC that uses a Windows 98 or older operating system. I wanted modern software with a modern interface and the ability to do some customizing, so it was clear that doing a retrofit was the only option.

As far as I could tell, the stepper motors and all of the electronics on the lathe functioned properly. I did not see any significant reason to change anything on the machine that didn't need to be changed. As the saying goes, "if it ain't broke, don't fix it." Why spend money if you don't need to? With this attitude in mind, I started the retrofit of my Compact 5 PC.

The first thing I looked at was the machine's interface to a computer. In the case of the Compact 5 PC, the interface is a parallel port connection. A number of modern control software programs, like Mach3 and EMC, use this type of interface. For this retrofit, I chose Artsoft's Mach3 control program. Of course, these modern control programs do require more advanced and capable computers than the older programs. For example, Mach3 requires a PC with Windows XP or 2000, at least a 1GHz processor, and a minimum display resolution of 1024 x 768. Having a parallel port connection, my first thought was that I could just use the step and direction pin assignment information from the Compact 5 PC manual (getting a manual can be a tremendous help) to configure Mach3 and be good to go. Unfortunately, when I tried it, my lathe gave no response to any of the control signals. So, now I had to determine why: Was it because the signals were set incorrectly; would the interface not support a program like Mach3, or something else?

There was no way to obtain the schematics for the machine's printed circuit boards (PCB) so some electrical detective work was necessary to determine how the machine functions internally. However, the interconnect diagram was in the manual, which gave many valuable hints to figure out the machine's electronics.

After a lot of head scratching, I decided to trace the signals from the parallel port connector on the back of the lathe to their destinations on the U3 PCB (I used an ohmmeter). The parallel port signals from pins 1 through 9 go to an integrated circuit (IC), which is an octal latch (74LS374) on U3. The latch's signals on the eight input pins transfer to the eight output pins and are held only when a pulse is sent to the latch's output control pin. Mach3 does not have the ability to send a signal to the output control pin of the latch every time a step or direction signal changes. My first thought was to scrap the U3 PCB and get a commercially-available parallel port breakout board, but before I went out and spent the money, I thought it would be a good idea to check what I could to make sure it would work.

The first thing I noticed was there were too many signals (four for each motor) coming from PCB U3 to PCB U2 (the stepper motor driver PCB) for just step and direction signals. After more investigating, it became apparent that the four signals were logic level signals that drive each of the four coils on the (unipolar) stepper motors. This meant the U3 PCB is more than a breakout board, it converts the step and direction signal to logic level signals to drive the individual stepper motor coils. The U2 PCB, which is the stepper motor driver, essentially just amplifies the logic level coil signals to drive the motors. Bottom line - a standard commercial breakout board will not work with the existing stepper motor driver PCB.

It was now looking like the retrofit would require more than just a new breakout board because of the unusual way functions were divided between the U2 & U3 PCBs. It would also require two new unipolar stepper motor drivers. This was getting to look like a somewhat more expensive and complicated operation than I had hoped. It was time to look at the U3 PCB again to see if there was a way to get around the latch. I tried to think of why EMCO would put a latch there in the first place – my best guess was it was put there so the computer could talk to the dongle, which was also on the parallel port, without the signals affecting the lathe. If I was right, the latch IC could be removed and the eight input pins could be connected to their respective eight output pins (Figure 1). So that's what I did, and it worked! Photo 1 shows the U3 PCB before any modifications and Photo 2 shows the PCB after the 74LS374 was removed and the eight jumpers added (note: PCB U3 dip switch 1 must be in the ON position). At this point, the lathe will work electrically with Mach3.

Practical Notes:

It's not worth trying to save the 74LS374; just clip it out and then unsolder the leads individually.

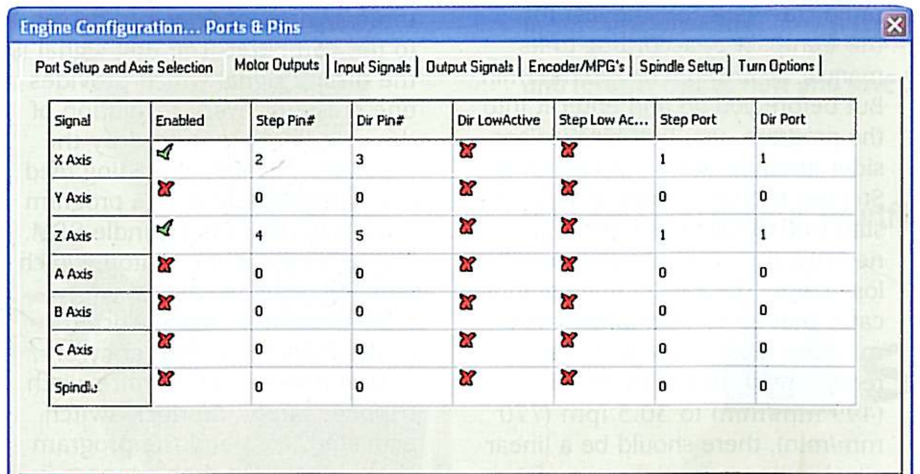
Use a soldering iron appropriate for PCBs, that is, a low wattage one with a small tip.

Think about investing in a desoldering pump (solder sucker) to clean out the holes, they can be purchased for as little as $6.00.

Use solid wire for the eight jumpers.

Once the electrical connections were fixed, it was time to configure the software. Since Mach3 or any generic control program has no idea what machine you are connecting it to, you have to provide the program with some information about the machine. The information can be divided into output from the program to the machine and input from the machine to the program.

The control program outputs step and direction signals over the parallel port for the two stepper motors. Figure 2 is a screen capture of the Mach3 Motor Outputs



Signal	Enabled	Step Pin#	Dir Pin#	Dir LowActive	Step Low Ac...	Step Port	Dir Port
X Axis	✓	2	3	✗	✗	1	1
Y Axis	✗	0	0	✗	✗	0	0
Z Axis	✓	4	5	✗	✗	1	1
A Axis	✗	0	0	✗	✗	0	0
B Axis	✗	0	0	✗	✗	0	0
C Axis	✗	0	0	✗	✗	0	0
Spindle	✗	0	0	✗	✗	0	0

FIGURE 1

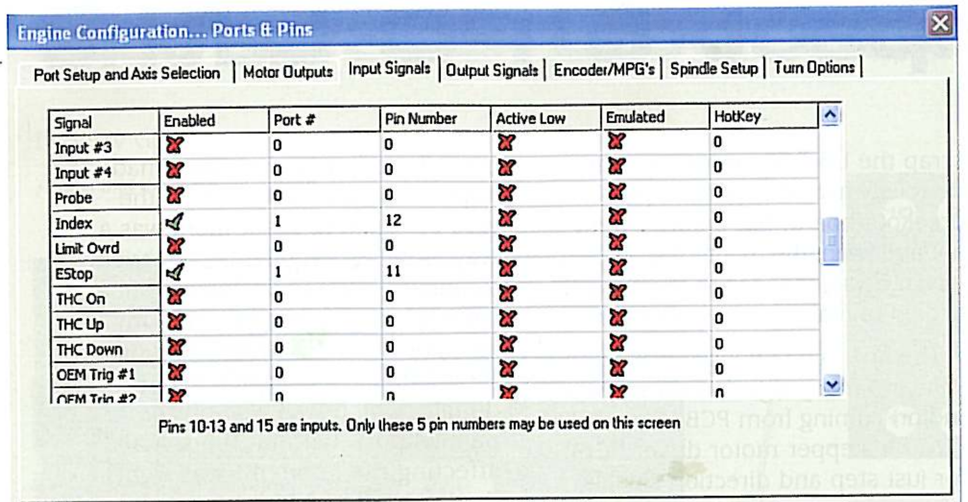


FIGURE 2

Ports and Pins window that shows the step and direction signal settings for the Compact 5 (this assumes that the U3 PCB dip switch switches 2 & 3 are in the ON position). If you have a manual, you will note this is different from what EMCO says because the step and direction pins are transposed. If the X-axis moves for a Z-axis input and vice-versa, swap the two stepper motor cables in the back of the lathe.

Mach3 also needs to know how many pulses are in each unit of length (determines the distance an axis moves per step), how many pulses it can send a second (velocity), and how fast it can change the step pulses (acceleration). To determine the number of steps per unit length, you need to find out the step angle of the motor, how many microsteps (if any) per full step, the reduction ratio (if any) between the motor and lead screw, and the threads per unit length of the lead screw. The Compact 5 stepper motors are 5° per step (1.8° per step is more typical). There is a pulley ratio of 5:2 between the stepper motor and the lead screw (16 teeth on the motor pulley and 40 teeth on the lead screw pulley). The lead screw is 10.16 tpi or 4 threads/cm (it's a metric lead screw). With 5° per step and a pulley ratio of 5:2, it takes 180 steps for one rotation of the lead screw ((360°/5°) x (5/2) = 180). So the number of steps per unit length is 1829 steps per inch (180 x 10.16 = 1829) or 72 steps per mm, assuming the full step mode. In terms of distance, it yields a distance of 0.000546" (1/1829) or 0.0138mm (1/72) per step. The preceding data is for the full step mode, when U3 dip switch 4 is in the ON position. In the half step mode, when dip switch 4 is in the OFF position, the number of steps per unit length is 3658 steps per inch or 144 steps per mm. In the half step mode, you get twice the resolution (0.000273" vs. 0.000546" or 0.00694mm vs. 0.0138mm), but this is at the cost of the new intermediate steps having only about 70% of full torque (this is due to the intermediate steps having only one motor coil (instead of two) active).

The maximum velocity for rapid traverses (G0 moves) for the Compact 5, according to its manual, is 30.3 ipm or 770 mm/min. But before you go and enter it into the program, you first have to consider another factor – acceleration. Stepper motors cannot go from a standstill to full speed instantaneously; if you try, the motor will lose steps. The EMCO manual indicates that up to 19.6 ipm or 499 mm/min there is no acceleration requirement, but from 19.6 ipm (499 mm/min) to 30.3 ipm (770 mm/min), there should be a linear acceleration over the span of 1 second (which seems like an awfully long time – it's very possible that it's a misprint). Unfortunately, Mach3 does not allow for two different acceleration rates. If the 1 second acceleration time between 499mm/min and 770mm/min is correct, then the acceleration time to 30.3 ipm would be 3.08 seconds (3.08 seconds total acceleration gives 1 second of acceleration between 499mm/min and 770mm/min). The 3.08 seconds to ramp up to full speed of 30.3 ipm was longer than I wanted so, taking the EMCO manual for its word, I decided to use 19.6 ipm for the maximum velocity with a very short acceleration time (I chose 0.05 seconds). If you would like to experiment with different velocities and accelerations, the Mach3 manual gives a number of qualitative techniques to determine what will work.

The above parameters for the stepper motors are entered into Mach3 via the Motor Tuning and Setup window (Figure 3). The number of steps per unit length is typed in, but the velocity and acceleration information can either be typed in or entered via a slider. The Step Pulse needs to be changed to 3 because the Compact 5's electronics system does not always register the narrower default step pulse width. The direction pulse can be left to the default value, 0. The data for the Steps per Unit, Velocity and Acceleration must be entered for both the X- and Z-axis.

The Compact 5 PC generates three signals that need to be sent to the computer. The first signal is the timing signal which provides one pulse for every revolution of the spindle. This is used by the Mach3 to calculate threading feed rates and also allows the program to display the actual spindle RPM. The second signal is E-stop, which tells the program that a safety related problem has occurred in the machine (emergency off button pressed, axis limit switch tripped, safety interlock switch activated, etc.) and the program should stop. On the Compact 5, when the front panel red emergency stop button is pressed or the chip guard (if the switch is installed) is opened, the E-stop signal is raised. If you do not have the optional chip guard switch installed, the X6 connector in the back of the Compact 5 must have its plug connected for E-stop to work (in the plug there are two jumper wires that connect pin 1 to pin 3 and pin 5 to pin 7, which emulate the chip guard switch contacts being closed). Figure 4 is a screen capture of the Mach3 Input Signals Ports and Pins window that shows the setting for the Speed Pulse Light Barrier (EMCO term) Timing (Mach3 term) and the Emergency Off and Chip Protection Door (EMCO term) Estop (Mach 3 term) signal. The third signal is the Synchronous Pulse Light Barrier (EMCO term) Index (Mach3 term), which is assigned to pin 12.

Before you actually start using your retrofitted Compact 5 with Mach3, I strongly suggest you go to the Artsoft web site and review the "Using Mach3Turn" manual and watch the "Turning – Intro" video.

For only the price of new software, a bit of detective work, and a little bit of soldering, a Compact 5 PC was retrofitted with modern software – a good deal. ⊕

FIGURE 3