Stepconf: Configure EMC 2.2 for Step & Direction CNC Machines

August 1, 2007

Abstract

EMC2 is capable of controlling a wide range of machinery using many different hardware interfaces. Stepconf is a program which generates EMC configuration files for a specific class of CNC machine: those connected to the PC using a standard **parallel port** and controlled with **step & direction** signals.

Contents

1 Step by Step Instructions

1.1 Basic Information

Machine Name Choose a name for your machine. Use only uppercase letters, lowercase letters, digits, "-" and "_".

Axis Configuration Choose XYZ (Mill), XYZA (4-axis mill) or XZ (Lathe).

Machine Units Choose Inch or mm. All subsequent questions (such as machine travel, leadscrew pitch, etc) will be answered in the chosen units

Driver Characteristics If you have one of the stepper drivers listed in the pulldown box, choose it. Otherwise, find the 4 timing values in your driver's datasheet and enter them. If the datasheet gives a value in nanoseconds, divide by 1000. For example, enter "500ns" as ".5".

Latency Test Result Enter the result of the Latency Test (see Appendix B) here.

Onscreen Prompt For Tool Change If this box is checked, EMC will pause and prompt you to change the tool when **M6** is encountered. Leave this box checked unless you plan to add support for an automatic toolchanger in a custom hal file

1.2 Parallel Port Setup

For each pin, choose the signal which matches your parallel port pinout. Turn on the "invert" checkbox if the signal is inverted (0V for true/active, 5V for false/inactive).

By clicking one of the **Output pinout presets**, pins 2 through 9 will be set to Step and Direction according to the Sherline standard (Direction on pins 2, 4, 6, 8) or the Xylotex standard (Direction on pins 3, 5, 7, 9).

By turning on **Include custom .hal configuration file**, you can add additional hal customizations in the file `custom.hal` after running stepconf.

1.3 Axis Configuration

Motor Steps Per Revolution The number of full steps per motor revolution. If you know how many degrees the motor is (e.g., 1.2°), then divide 360 by the degrees to find the number of steps per motor revolution.

Driver Microstepping The amount of microstepping performed by the driver. Enter "2" for half-stepping.

Pulley Ratio If your machine has pulleys between the motor and leadscrew, enter the ratio here. If not, enter "1:1".

Leadscrew Pitch Enter the pitch of the leadscrew here. If you chose "Inch" units, enter the number of threads per inch here (e.g., enter 8 for 8TPI). If you chose "mm" units, enter the number of millimeters per thread here (e.g., enter 2 for 2mm/rev). If the machine travels in the wrong direction, enter a negative number here instead of a positive number.

Maximum Velocity

Maximum Acceleration The correct values for these items can only be determined through experimentation. See “Finding Velocity and Acceleration” below.

Home Location The position the machine moves to after completing the homing procedure for this axis. For machines without home switches, this is the location the operator manually moves the machine to before pressing the Home button.

Table Travel The range of travel that gcode programs must not exceed. The home location must be inside the Tabel Travel. In particular, having Home Location exactly equal to one of the Tabel Travel values is incorrect configuration

Home Switch Location The location at which the home switch trips or releases during the homing process. This item and the two below only appear when Home Switches were chosen in the Parallel Port Pinout.

Home Search Velocity The velocity to use when moving towards the switch. If the switch is near the end of travel, this velocity must be chosen so that the axis can decelerate to a stop before hitting the end of travel. If the switch is only closed for a short range of travel (instead of being closed from its trip point to one end of travel), this velocity must be chosen so that the axis can decelerate to a stop before the switch opens again, and homing must always be started from the same side of the switch.
If the machine moves the wrong direction at the beginning of the homing procedure, negate the value of **Home Search Velocity**.

Home Latch Direction Choose “Same” to have homing back off the switch, then approach it again at a very low speed. The second time the switch closes, the home position is set.
Choose “Opposite” to have homing slowly back off the switch. When the switch opens, the home position is set.

Time to accelerate to max speed

Distance to accelerate to max speed

Pulse rate at max speed Information computed based on the values entered above. The greatest **Pulse rate at max speed** determines the **BASE_PERIOD**, and values above 20000Hz may lead to slow response time or even lockups (the fastest usable pulse rate varies from computer to computer)

1.4 Spindle Configuration

These options only appear when “Spindle PWM”, “Spindle A” or “Spindle PPR” are chosen in the **Parallel port pinout**.

1.4.1 Spindle Speed Control

If “Spindle PWM” appears on the pinout, the following information should be entered:

PWM Rate The “carrier frequency” of the PWM signal to the spindle. Enter “0” for PDM mode, which is useful for generating an analog control voltage. Refer to the documentation for your spindle controller for the appropriate value.

Speed 1 and 2, PWM 1 and 2 The generated configuration file uses a simple linear relationship to determine the PWM value for a given RPM value. If the values are not known, they can be determined. See the section “Determing Spindle Calibration” below.

1.4.2 Spindle-synchronized motion (lathe threading)

When the appropriate signals from a spindle encoder are connected to the parallel port, EMC supports lathe threading. These signals are:

Spindle PPR Also called “Index pulse”, this is a pulse that occurs once per revolution of the spindle.

Spindle A This is a pulse that occurs in multiple equally-spaced locations as the spindle turns.

Spindle B (optional) This is a second pulse that occurs, but with an offset from **Spindle A**. The advantages to using both **A** and **B** are increased noise immunity and increased resolution.

If “Spindle A” and “Spindle PPR” appear on the pinout, the following information should be entered:

Cycles per revolution The number of cycles of the **Spindle A** signal during one revolution of the spindle

Maximum threading speed The maximum spindle speed at which threading will be used. Like high motor speeds in Hz, high **SPINDLE A** speeds require a low **BASE_PERIOD** setting (TODO)

1.5 Machine Configuration Complete

Click “Apply” to write the configuration files. Later, you can re-run this program and tweak the settings you entered before.

2 Finding Velocity and Acceleration

2.1 Finding Maximum Velocity

Enter a low acceleration (e.g., 2 in/s^2 or 50mm/s^2) and velocity 20% above the speed you hope to attain. Finish the remaining steps of the configuration process, then launch emc with your configuration.

Turn the machine on and select the axis to test. Turn the “Jog Speed” slider to the speed you wish to test. Move the axis to a position that can easily be measured by eye or with a dial gauge. Press “Touch Off” and enter “0”. Now jog the axis in both “+” and “-” directions, moving far enough that the machine accelerates to the speed being tested. If it is convenient and safe to do so, push the table against the direction of motion to simulate cutting forces. If the machine stalls, reduce the jog speed and try again from the “Touch Off” step.

If the machine did not obviously stall, change to the MDI tab and enter “G1 X0 F<number>” to return the axis to the starting point. If the position is incorrect, then the axis stalled or lost steps during the test. Reduce the jog speed and try the process again from the “Touch Off” step.

If the machine doesn’t move, stalls, or loses steps no matter how low you turn Jog Speed, verify the following:

- Correct step waveform timings
- Correct pinout, including “Invert”
- Correct, well-shielded cabling
- Physical problems with the motor, motor coupling, leadscrew, etc.

Once you have found a speed at which the axis does not stall or lose steps during this testing procedure, reduce it by 10% and use that as the axis Maximum Velocity.

2.2 Finding Maximum Acceleration

With the Maximum Velocity you found in the previous step, enter the acceleration value to test. Finish the remaining steps of the configuration process, then launch emc with your configuration.

Perform the same Touch Off, Jog and Measure procedure as above, setting Jog Speed to its maximum value. To test a higher or lower value, run Stepconf again. Once you have found a value at which the axis does not stall or lose steps during this testing procedure, reduce it by 10% and use that as the axis Maximum Acceleration.

3 Determining Spindle Calibration

Enter the following values in the Spindle Configuration page:

Speed 1:	**0**	PWM 1:	**0**
Speed 2:	**1000**	PWM 1:	**1**

Finish the remaining steps of the configuration process, then launch emc with your configuration. Turn the machine on and select the MDI tab. Start the spindle turning by entering: `M3 S100`. Change the spindle speed by entering a different S-number: `S800`. Valid numbers range from 0 to 1000.

For two different S-numbers, measure the actual spindle speed in RPM. Record the S-numbers and actual spindle speeds. Run Stepconf again. For “Speed” enter the measured speed, and for “PWM” enter the S-number divided by 1000.

Because most spindle drivers are somewhat nonlinear in their response curves, it is best to:

- Make sure the two calibration speeds are not too close together in RPM
- Make sure the two calibration speeds are in the range of speeds you will typically use while milling

For instance, if your spindle will go from 0RPM to 8000RPM, but you generally use speeds from 400RPM to 4000RPM, then find the PWM values that give 1600RPM and 2800RPM.

A Axis Travel, Home Location, and Home Switch Location

For each axis, there is a limited range of travel. The physical end of travel is called the **hard stop**.

Before the **hard stop** there is a **limit switch**. If the limit switch is encountered during normal operation, emc shuts down the motor amplifier. The distance between the **hard stop** and **limit switch** must be long enough to allow an unpowered motor to coast to a stop.

Before the **limit switch** there is a **soft limit**. This is a limit enforced in software after homing. If a MDI command, or gcode program would pass the soft limit, it is not executed. If a jog would pass the soft limit, it is terminated at the soft limit.

The **home switch** can be placed anywhere within the travel (between hard stops). As long as external hardware does not deactivate the motor amplifiers with the limit switch is reached, one of the limit switches can be used as a home switch.

The **zero position** is the location on the axis that is 0 in the machine coordinate system. Usually the **zero position** will be within the **soft limits**. On lathes, constant surface speed mode requires that machine **X=0** correspond to the center of spindle rotation when no tool offset is in effect.

The **home position** is the location within travel that the axis will be moved to at the end of the homing sequence. This value must be within the **soft limits**.

A.1 Operating without Limit Switches

A machine can be operated without limit switches. In this case, only the **soft limits** stop the machine from reaching the **hard stop**. **Soft limits** only operate after the machine has been homed.

A.2 Operating without Home Switches

A machine can be operated without home switches. If the machine has limit switches, but no home switches, it is best to use a limit switch as the home switch (e.g., choose **Minimum Limit + Home X** in the pinout). If the machine has no home switches, or the limit switches cannot be used as home switches for another reason, then the machine must be homed “by eye”. Homing by eye is not as repeatable as homing to switches, but it still allows the **soft limits** to be useful.

B Latency Test

Generating step pulses in software has one very big advantage - it’s free. Just about every PC has a parallel port that is capable of outputting step pulses that are generated by the software. However, software step pulses also have some disadvantages:

- limited maximum step rate
- jitter in the generated pulses
- loads the CPU

Latency is how long it takes the PC to stop what it is doing and respond to an external request. In our case, the request is the periodic "heartbeat" that serves as a timing reference for the step pulses. The lower the latency, the faster you can run the heartbeat, and the faster and smoother the step pulses will be.

Latency is far more important than CPU speed. A lowly Pentium II that responds to interrupts within 10 microseconds each and every time can give better results than the latest and fastest P4 Hyperthreading beast.

The CPU isn’t the only factor in determining latency. Motherboards, video cards, USB ports, and a number of other things can hurt the latency. The best way to find out what you are dealing with is to run the RTAI latency test.

DO NOT TRY TO RUN EMC2 WHILE THE TEST IS RUNNING

On Ubuntu Breezy and Dapper, you can run the test by opening a shell and doing:

```
cd /usr/realtime-*/testsuite/kern/latency;
./run
```

You should see something like this:

```
ubuntu:/usr/realtime-2.6.12-magma/testsuite/kern/latency$ ./run
* * * Type ^C to stop this application. * *
## RTAI latency calibration tool ##
# period = 100000 (ns)
# avrgtime = 1 (s)
# do not use the FPU
# start the timer
# timer_mode is oneshot
RTAI Testsuite - KERNEL latency (all data in nanoseconds)
RTH| lat min| ovl min| lat avg| lat max| ovl max| overruns
RTD|   -1571|   -1571|    1622|    8446|    8446| 0
RTD|   -1558|   -1571|    1607|    7704|    8446| 0
```

```
RTD|   -1568|   -1571|    1640|    7359|    8446| 0
RTD|   -1568|   -1571|    1653|    7594|    8446| 0
RTD|   -1568|   -1571|    1640|   10636|   10636| 0
```

While the test is running, you should "abuse" the computer. Move windows around on the screen. Surf the web. Copy some large files around on the disk. Play some music. Run an OpenGL program such as glxgears. The idea is to put the PC through its paces while the latency test checks to see what the worst case numbers are.

The last number in the column labeled "ovl max" is the most important. It contains the worst latency measurement during the entire run of the test. Record this number, and enter it in Stepconf when it is requested.

In the example above, that is 10636 nano-seconds, or 10.6 micro-seconds, which is excellent. However the example only ran for a few seconds (it prints one line every second). You should run the test for at least several minutes; sometimes the worst case latency doesn't happen very often, or only happens when you do some particular action. For instance, one Intel motherboard worked pretty well most of the time, but every 64 seconds it had a very bad 300uS latency. Fortunately that was fixable (see "Fixing Dapper SMI Issues" `http://wiki.linuxcnc.org/cgi-bin/emcinfo.pl?FixingDapperSMIIssues`)

So, what do the results mean? If your "ovl max" number is less than about 15-20 microseconds (15000-20000 nanoseconds), the computer should give very nice results with software stepping. If the max latency is more like 30-50 microseconds, you can still get good results, but your maximum step rate might be a little dissapointing, especially if you use microstepping or have very fine pitch leadscrews. If the numbers are 100uS or more (100,000 nanoseconds), then the PC is not a good candidate for software stepping. Numbers over 1 millisecond (1,000,000 nanoseconds) mean the PC is not a good candidate for EMC, regardless of whether you use software stepping or not.

Note that if you get high numbers, there may be ways to improve them. Another PC had very bad latency (several milliseconds) when using the onboard video. But a $5 used Matrox video card solved the problem - EMC does not require bleeding edge hardware.

C Home and Limit Switch wiring options

The ideal wiring for external switches would be one input per switch. However, the PC parallel port only offers a total of 5 inputs, while there are as many as 9 switches on a 3-axis machine. Instead, multiple switches are wired together in various ways so that a smaller number of inputs are required.

The figures below show the general idea of wiring multiple switches to a single input pin. In each case, when one switch is actuated, the value seen on INPUT goes from logic HIGH to LOW. However, emc expects a TRUE value when a switch is closed, so the corresponding "Invert" box must be checked on the pinout configuration page.

The following combinations of switches are permitted in Stepconf:

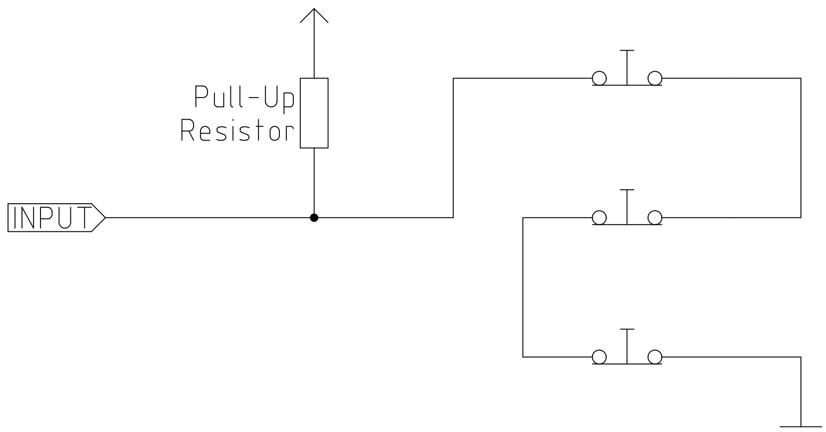

Figure 1: Wiring Normally Closed switches in series (simplified diagram)

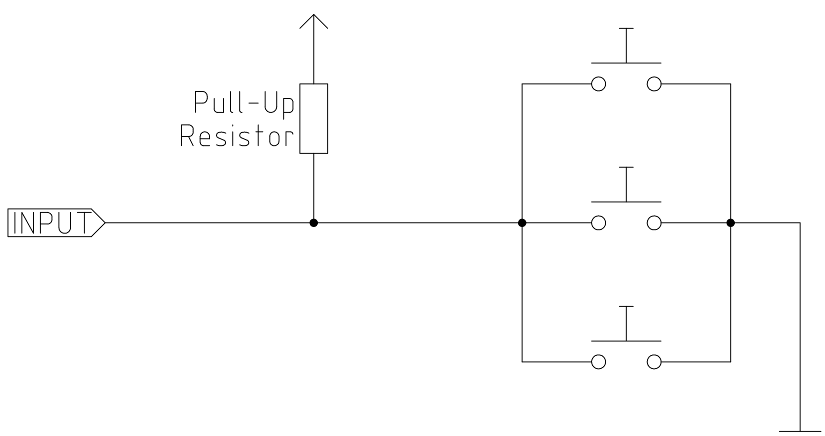

Figure 2: Wiring Normally Open switches in parallel (simplified diagram)

- Combine home switches for all axes
- Combine limit switches for all axes
- Combine both limit switches for one axis
- Combine both limit switches and the home switch for one axis
- Combine one limit switch and the home switch for one axis

The last two combinations are also appropriate when a “home to limit” is used.