

How to initialize the PMAC2

With the PMAC installed. Run Executive.
Do a full card reset: \$\$\$***

Each motor has several critical variables that needs to be set before it can even move.
The x stands for any motor from 1-8. Most likely we will set motor 1,2, and 3 to the x, y,
and z axis. That is, x will stand for 1,2, and 3 to setup all three motors.

Ix30 = 2000000 (i.e. I130 = 2000000<cr> I230 = 2000000<cr> I330 = 2000000)
This is the PID gain. It acts like a spring. Higher the value, the stiffer the spring
on the axis. The higher the value, the less following error.

Ix00 = 1
Turns on the motor

#xj/
#x means address motor x. j/ means to close the loop on that axis.

I9x0 = 3
Set the direction of encoders to 4x CW

I9x6 = 3
Output mode select. Note that default of this I variable is 0 which is PWM mode
while in other versions of the PMAC ROM it defaults to DAC output mode. We want
DAC output mode so be sure to set I9x6 = 3 for motors 1,2,3, and 4.

Now, to make the operator panel usable:

Download Adv600M.cgf file. This should be located in:
C:/program files/delta tau/NC 2.36/Mill/Adv600m.cfg.

This is a PLC file to run the operator panel.

After successful download, check *Status->PLCProgram Info*.

Programs 1,2,4,5,13 should be loaded

I5=2 to enable PLC programs to run

Enable plc 1,2,4,5,13 to enable those particular programs to run in the background

Finally issue a **save** to save all those I variables to the EEPROM.

Close out of the Executive Software and run the NC program.
After the NC Program Starts, the operator panel should function.

Other commands that may come in handy but are not required to make the motors move
are:

To view the contents of an I variable, issue the command without an assignment. I.e. **I125** Then the current value will be displayed.

Ix25 = \$2C0nn

This I variable sets the functionality of the limit switches and fault states. The **nn** part depends on where the address of the flags we want to assign to motor **x**. For example: we correlate LIM1, HMFL1.. etc normally to motor one. So a generic assignment would be: **I125 = \$C000** since **\$C000** is the address to the first set of limit switches. NOTE: When looking in the PMAC manual, the limit switch addresses are different from the PMAC2 manual. We need to use the PMAC2 manual addresses.

Ix11 = 32000

This is the maximum following error before a fault. But this value is divided by 16. So $32000/16 = 2000$ counts maximum following error

Ix69

Max voltage magnitude. Default at +/- 6V. Set to 32,767 for +/- 10V Dac output range.

Ix15

Motor **x** deceleration rate on position limit or abort. Units in cnts/ms^2

Encoder info that may come in handy

625 pulses/rev. But in quadrature mode it's 2500 cts/rev

The ball screw pitch are 0.25" per turn.

This means each encoder count stands for 0.0001"

Jog Parameters

After setting up the motors we may want to jog the motors to check their functionality. Set **Ix19 = 0.1**, **Ix20=100**, **Ix21 = 50** which are tested to give results. These set the maximum acceleration and the s-curve parameters.

Also **Ix22** sets the velocity

Tuning motors

To correctly tune the motors, we may need to zero out the offsets. We can turn POT4 of the Advanced Motion Controls amplifiers to adjust the motor outputs to zero. Further we may fine-tune it even more with **Ix29**.

Origin direction

On Dave's machine:

Z positive moves the spindle up
Y positive moves the table towards the front
X positive moves the table towards the left
Homing causes all the directions to move in the positive direction.

We are setting up the BostoMatic the same direction as Dave's machine.

Homing

Homing procedure does not work yet but this is what I got so far.

In23 = 5 set home speed to 5 ct/ms positive. This makes it go in the positive direction when it homes.
I9n1 = 0 Note i911 stays at 1 because setting it to 0 means the same thing.
I9n2 = 3 Index and flag high for capture – may need to change it to just flag so it wont run into the second limit switch trying to look for the index pulse.
I9n3 = 1 Use the positive end limit flag as the homing capture flag
Ix26 = 32000 Set home offset to $32000/16 = 2000$ cts. I.e. back off the flag 2000 cts.
In25 = \$2C0nn Where nn depends on the motor we are addressing. The number 2 disables the limit switch so that we can use the limit switches for homing.

Home1

In25 = \$C0nn Change the limit back so we are protected by the first limit switches again.

PMAC Terminal Window basic functions

#1<cr> Address motor 1
P<cr> get position of currently addressed motor
F7 update position

DPRAM NOTES USING W2K

Address that works best is \$E8000

To see who is using what addresses, look under the Control Panel->System Properties->Hardware tab -> Device manager -> View properties by type -> look under memory.

Use the Motion Exe program to load the correct memory location for PMAC2 to use. Run Motion Exe and click setup. Choose the correct address and close out of the program. Loading may take a while. If the PMAC is in bootstrap mode, it will alert you and give you an option to get out of bootstrap mode. If you say yes, simply go back to the Motion Exe program again and set the PMAC2 address once again for changes to take effect.

Note that these notes are for W2K and will probably be very different from the setup procedure used in Windows 98.

Abit Motorboard BIOS checksum error problem

Sometimes when restarting the machine, there will be a BIO checksum error giving a low tone beep. The computer will not continue until it is shut off and the power on reset jumper placed on the PMAC card. As of now, it appears that when the DPRAM is turned on, it is conflicting with either the operating system or the motherboard.

The quick work around for now is to turn the computer off and place a jumper on E3, the reinitialization jumper. This will bring the PMAC to its default and make it appear once again that it does not have dual port ram. The computer will then once again boot up. After booting, once may load the DPRAM again using the MotionExe Program.

The address that works best under W2k is \$E8000.

Following the notes above under DPRAM NOTES USING W2K to reload the DPRAM.

Why does the Amplifier fault?

This was a recurring problem that can have many causes. When this happens, make sure the pots on the amplifier are adjusted correctly.

What currently works well is to adjust

Pot1 fully CCW

Pot2 fully CW

Pot3 fully CW then 4 turns CCW to reduce the reference gain.

Pot4 is the offset so it must be adjusted until the outputs to the motor are at zero volt.

Another major problem could stem from the PMAC defaulting the motor's modes to PWM. We do not have a digital amplifier so we are going to use DAC mode. Using PWM mode while having DAC mode hardware could cause the following error to increase until the limit and cause the amplifiers to fault. The following error increases because the motors are not moving to the location the PMAC expects it to move.

Also, a elastic looking setup where the movements seem to have too much lag may need an increase in PID gain. Increasing it is like increasing the tension on a spring pulling the milling machine table to the expected location. PID gain (Ix30) should be set around 2000000 in the BostoMatic application.

Output Voltage on motors

Ix69 Motor x command (DAC) Limit

Range 0 to 32,77

Default 20,480 (~6.25V)

Since the measured supply voltage on the amplifiers is 180V.

Our motors are rated at 130 V

This means:

0 to 10 V input to the amplifiers gives 0 to 180V on the motors.

0 to 6.25V input to the amplifiers gives 0 to 112.5V on the motors.