



DELTA TAU

Power PMAC NC 2014

Motion Commander Foundation
©2014 Greene & Morehead Engineering, Inc.



Quick Start

Power PMAC NC 2014



Delta Tau Data Systems, Inc.

June, 2015

COPYRIGHT INFORMATION

Software: © 2014 Delta Tau Data Systems, Inc. All rights reserved.

Software User Manual: © 2014 Delta Tau Data Systems, Inc. All rights reserved.

Motion Commander Foundation: © 2012-2014 Greene & Morehead Engineering, Inc. All rights reserved.

This document contains proprietary information of Delta Tau Data Systems, Inc. The information contained herein is not to be used by or disclosed to third parties without the express written permission of an officer of Delta Tau Data Systems, Inc.

TRADEMARK ACKNOWLEDGMENT

Windows, Visual Studio and .NET Framework are registered trademarks of Microsoft Corporation. MTConnect is a registered trademark of the MTConnect Institute. Other brands, product names, company names, trademarks and service marks are the properties of their respective holders.

REVISION HISTORY

Version	Date	Description
1.0	6/16/2015	Initial release

Power PMAC NC 14™ - Quick Start Manual

Contents

Introduction	4
Requirements.....	5
Installation	5
Deployment.....	6
Configuration File.....	7
The Power PMAC Project.....	7
Hardware Key.....	8
Power NC Software Levels	8
Working With Power NC Software.....	9
Install.....	9
Quick Start – Get an NC Machine Running	10
Overview:	10
Target PC System Setup:	10
Power PMAC Setup	11
Power PMAC Virtual Motors Configuration File code	13
Power NC Software Setup.....	15
Running the Machine.....	17
What Next ?	19
Deploy to Actual Target Machine	20

Power PMAC NC™ - Quick Start Manual

Introduction

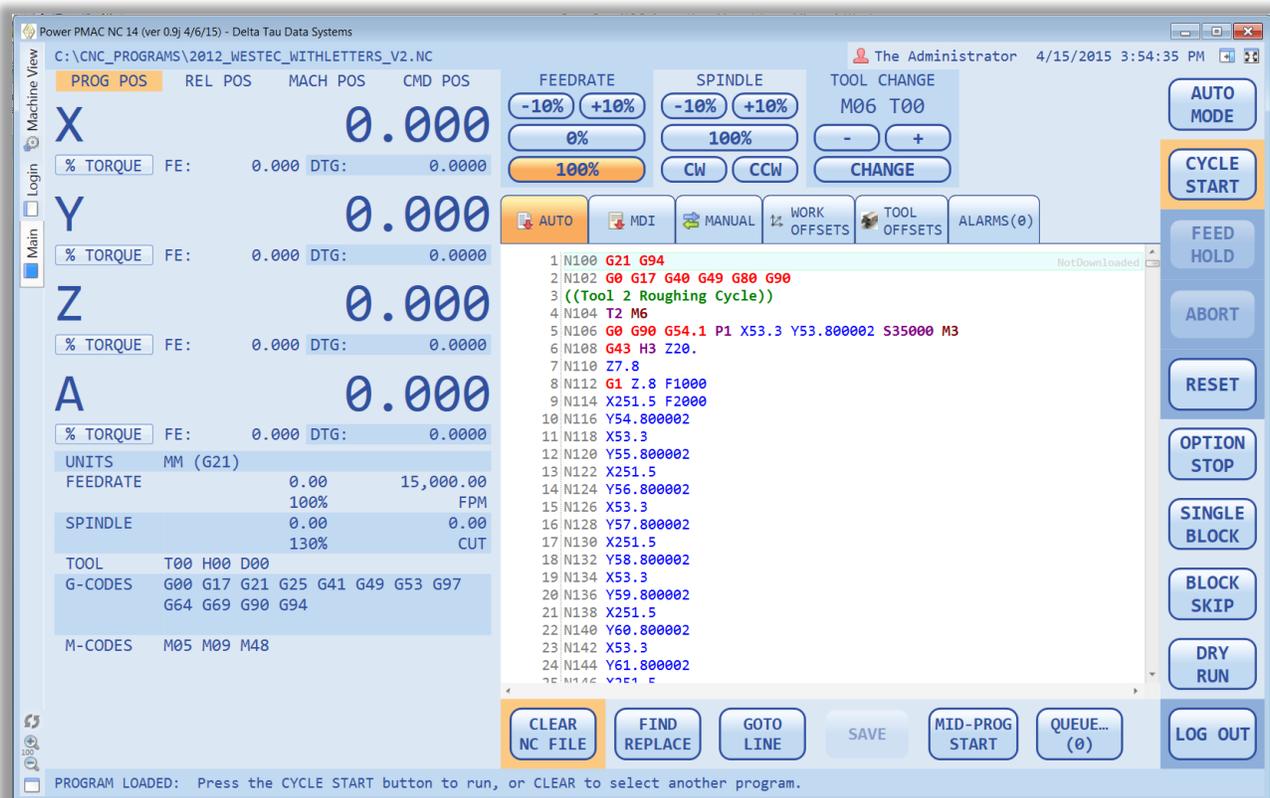
The *Power PMAC NC 14* HMI (PPNC14) is a host PC application for Delta Tau Power PMAC controlled CNC machines. This document is a Quick Start Manual for the *Power PMAC NC 14* application. It contains information about how to quickly get going with your Power PMAC NC software. This is not a user manual but designed for an Engineer level OEM / Integrator familiar with developing custom code and logic for a PMAC based NC type machine.

As a Quick Start manual there are also many assumptions made that may require the reader to consult with the full setup of manuals available for the PMAC and the NC14 application.

Note the much of the example code and detailed manuals are only supplied with the **SDK** version of the NC14 software and are not included with the basic **Runtime** version.

Other Manuals to refer to:

- MCF Developer's Guide.pdf* reference for working with the MCF coding engine that Power NC is based on
- PowerPmacNC Developer's Guide.pdf* reference for customizing the Power NC code to a specific application
- PowerPmacNC Software Users Manual.pdf* user reference for how to use the actual application to run the machine
- Using GitHub.pdf* quick guide to using the GitHub application to keep your SDK up to date



Requirements

The PPNC14 program is compatible with Windows 7 or newer (64-bit or 32-bit).

The application requires .NET 4.0 and the Visual C++ 2010 runtime libraries. The application will install these components automatically. The following links can be used for manual installation of the same libraries.

Microsoft .NET Framework 4 (Web Installer)

<http://www.microsoft.com/en-us/download/details.aspx?id=17851>

Microsoft Visual C++ 2010 Redistributable Package (x64)

<http://www.microsoft.com/en-us/download/details.aspx?id=14632>

Microsoft Visual C++ 2010 Redistributable Package (x86)

<http://www.microsoft.com/en-us/download/details.aspx?id=5555>



Some versions of the PPNC14 program also support the Turbo PMAC and are 32-bit applications, requiring the x86 version of the Visual C++ Redistributable - even on 64-bit Windows systems.

Installation

The *PPNC14 Runtime Software* is distributed in 2 ways:

1. On Delta Tau Media available by Download, CD, or USB drive as a set of Install style files.
 - insert the media and select the "Setup.exe" application
 - follow the Wizard style prompts to install the Power NC Software
 - if your system does not meet the requirements you will be prompted to have these installed
2. Via a private GitHub repository. In order to access the online repository sign up for a [free GitHub account](#) and give your account name to **Delta Tau Data Systems** when you purchase the Runtime. You will be given read-only access to the repository. Install [GitHub for Windows](#) on your development PC, log in, and "Clone" the repository. You will want to "Sync" occasionally to insure you have the latest release version.



It is highly recommended that you make working copies of both the latest release of *PPNC14* Runtime and your working copy avoid losing your changes when you Sync. If a Sync fails for any reason, simply delete the entire "GitHub\PowerPmacNc14-Runtime" folder and Clone again.

Deployment

The *PPNC14* HMI can be deployed by simply copying the "PowerPmacNc14-Runtime" folder to any location on your machine. The folder may be renamed if desired. Your distribution must include the files shown below. (Files not shown in this list may be deleted without affecting the application.)

Name	Type
Languages	File folder
Skins	File folder
Temp NC	File folder
VCP4	File folder
DeviceMembers.xml	XML Document
DynamicDataDisplay.dll	Application extens...
ICSharpCode.AvalonEdit.dll	Application extens...
Interop.PCOMMSERVERLib.dll	Application extens...
MCF.DeltaTau.dll	Application extens...
MCF.DeltaTau.PcommServer.dll	Application extens...
MCF.MTConnect.dll	Application extens...
MCFoundation.dll	Application extens...
MessageLogViewer.exe	Application
Messages.xml	XML Document
Microsoft.WindowsAPICodePack.dll	Application extens...
Microsoft.WindowsAPICodePack.Shell.dll	Application extens...
PowerPmacNC.exe	Application
PowerPmacNC.ini	Configuration sett...
PowerPmacNC_MessageLog.txt	Text Document
PowerPmacNC_Notes.rtf	Rich Text Format
PowerPmacNC_Settings.xml	XML Document
Renci.SshNet.dll	Application extens...
Routrtek.Granados.dll	Application extens...
SecureDongle_Control32.dll	Application extens...
SecureDongle_Control64.dll	Application extens...
WpfGauge.dll	Application extens...

What makes this unique for your machine or project?

- Several key files located in the PPNC14 directory are used to customize the application for your machine:
- These files will need to be also added to the above set of files for custom machine HMI to be complete.

PowerPmacNC_Settings.xml

- This file is created when you edit the settings inside the NC14 – Machine View – Settings.
- For example in this Machine View – Settings such as “Jog Speed1” are saved in this file.
- When you get machine setup this file needs to be saved and distributed.

PowerPmacNC.ini

- Rename the file “Reference PowerPmacNC.ini” to “PowerPmacNC.ini”
- Edit this file to customize the Power NC application – see more in “Quick Start” section.

Messages.xml

- Rename the file “Reference Messages.xml” to “Messages.xml”
- Edit this file so that custom messages for this machine can be displayed to the user.

Configuration File

The application reads the "PowerPmacNC.ini" configuration file in its exe directory at start-up to obtain its configuration data. A *Reference* copy of this file is included in the project for convenience. Copy "Reference PowerPmacNC.ini" to the exe directory, rename it to "PowerPmacNC.ini", and edit it to specify your machine type, axis definitions, units, velocity units, and other important parameters. The configuration file is well commented for convenience.

```
[Machine Constructor]
; TODO: Specify from one to ten axis labels separated by commas.
; Axis labels can be more than one character but must be short. Suggest two characters max.
AxisLabels=X,Y,Z,A,B

; TODO: Specify motor numbers separated by commas (for status monitoring).
; The first motor number will be used to monitor the status of the first axis, etc.
MotorNumbers=1,2,3,4,5

; TODO: Specify native length units (INCH or MM) and decimal places of precision (0-6).
NativeLengthUnits=MM
NativeLengthDecimalPlaces=3

; TODO: Select the controller (PowerPmacController, TurboPmacController or MockController)
Controller=PowerPmacController
```

The Power PMAC Project

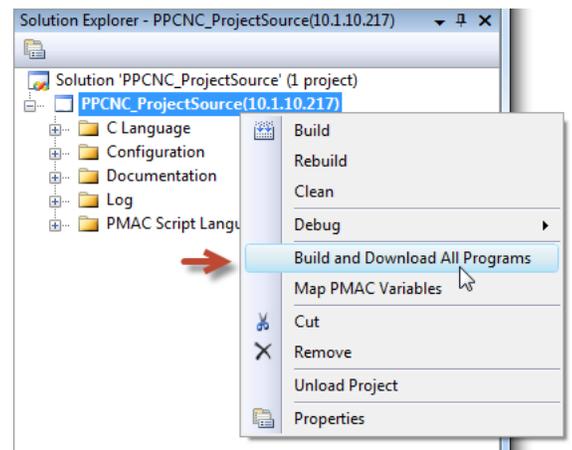
The PPNC14 application requires the Power PMAC controller be configured with its source code counterpart to enable proper functionality and handshaking. This Power PMAC project is included with your product.

The Power PMAC project will be located in the "PowerPmacNc14-Runtime\PMAC Source Code\PowerPMAC" folder. Make a working copy of this directory before you download the project to the controller.

Open the "PPCNC_ProjectSource.PowerPmacSuite_sln" solution file in the *Power PMAC IDE*, right-click and select "Build and Download" as shown. Look for the "Download Successful" message in the Output window.

```
Download Successful.
Total Project Download time = 13.057 sec
Total Project Build and Download time = 24.212 sec
```

 After downloading the project, use the Terminal window to issue a "save" command to copy the project to nonvolatile flash memory, then issue a "\$\$\$" command to reset the controller.

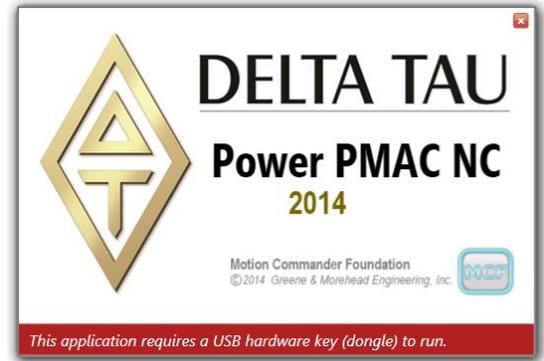


At this point the Power PMAC controller is now ready to work with the *PPNC14* program in a virtual mode. Actual machine functionality will require the appropriate integration of the motors, I/O, safety systems, etc. The default Power PMAC code is used as a starting point for all machine integrations which will utilize the PPNC14 software.

Hardware Key

This application requires a USB hardware key (dongle) to run. You will receive hardware keys when you purchase copies of the PPNC14 program from Delta Tau Data Systems.

The hardware key is compatible with all versions of Windows and does not require a driver to be pre-installed.



Power NC Software Levels

There are now 3 levels of the Power NC software distributed:

DEMO: This is a free to download version which will fully run in a simulated mode. It will not connect to a PMAC controller, but runs with a Mock Controller. It does not require a Hardware Key to be used. It does fully demonstrate the features of the NC product, able to open load and run (in simulated mode) NC files.

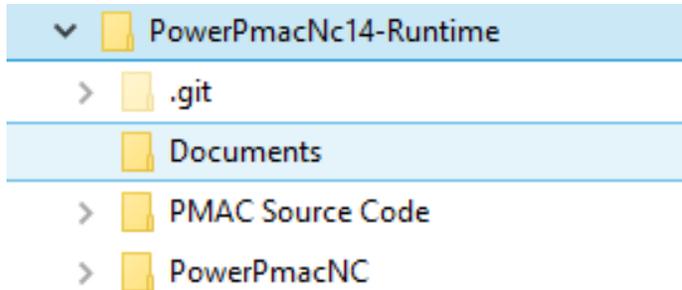
RUNTIME: This is a single license which allows a fully configured Power NC software package to run a machine that it is designed for. This does require a Hardware Key to be used. It can work with a Power PMAC, Turbo PMAC or no hardware controller in a simulated Mock Controller. This version can be fully customized to show custom panels, screens and extra controls as designed by the OEM/Integrator. However purchasing this version alone does not provide the examples, manuals or needed support to do this, see SDK version for this.

SDK: This version includes the RUNTIME license and does require a Hardware Key to be used. This version also includes extra developer manuals and example(s) of custom panels and “Add-Ins” used with the Basic Power NC. It is intended for an OEM or Integrator to purchase this license to get the extra examples, documentation and support to fully customize the basic Power NC application for his machine or project. Once this customization is done it can be freely distributed with each RUNTIME license purchased for each deployment on an actual machine. This also includes access to the GitHub SDK Repository online which contains continuous updates and example code for download and continued enhancements of projects.

Working With Power NC Software

Install

Now you have installed the Power NC Software suite and are ready to work with it. Whether you did this using media from Delta Tau download, USB drive or via the GitHub repository online you should now have a single installed directory on your development system that looks similar to below:



Look here for up to date documentation and help.

Here is starting PMAC controller code for your machine.

Here is Power NC HMI application Runtime code.

If you used GitHub to download the repository you should first make a copy of this GitHub version and put the copy in a different directory on your development system this will keep your ongoing working version of code and files separate from the version that is being synchronized with the GitHub repository online. If not the next time you update with the GitHub online repository you may lose your working changes.

So working with GitHub do this:

Use GitHub for Windows to “Clone” or copy the software to your system.

Copy this set of files to a working directory on your system.

Leave the “Clone” version alone and only update as needed to keep this in sync with online version.

If you lose or “mangle” the official “Clone” version, simply delete it’s directory and “Clone” again from Online.

Now you will have a fresh up to date copy in sync with the Delta Tau current release of the software.

If a new version is released:

- Allow GitHub to update your Clone version.
- Use a “diff” (Kdiff.exe) program to do compare with your working source files and update as needed.
- The files in PowerPmacNC can simply be copied over your current working set (backup working first).
 - you key setup files Deployment section will not be overwritten
- The files in PMAC Source Code have to be carefully compared and edited one by one since your PMAC code has likely now been customized for your machine.
 - however changes to these files are usually limited to the NC files (M,Gcodes) and some functions related to the HMI interaction
 - likely custom machine PLC’s and Programs have been created in New files that would not be affected by any changes made to the core group of PMAC files used by the NC HMI

To get back to old version on GitHub:

- In GitHub for Windows select the older version in History window you want to get.
- Select GitHub online link in upper left – this takes you to online GitHub in Browser
- Select the Browse Files Button in upper Right
- Select the “Download ZIP” button in lower Right
- This will download a zipped set of files from that version. Note name includes the version Hash value.

Quick Start – Get an NC Machine Running

Overview:

Here we will quickly go thru basic steps to get a fully working NC machine up and running with the Power NC software. This will entail setting up and configuring a Power PMAC to run 4 virtual motors and work with the Power NC HMI. We will also go thru basic setup of the NC HMI to setup a 3 axis Mill and be able to Jog motors and run NC code from the HMI. Using the Power PMAC IDE we can verify that the PMAC and HMI are working together to control the motors of the NC machine.

Target PC System Setup:

-On your Target HMI PC system:

Create a new Working directory for our Machine development:

C:\NC14

Copy directory and files from GitHub Clone directory: "PowerPmac"

-this is found in \PMAC Source Code\PowerPMAC

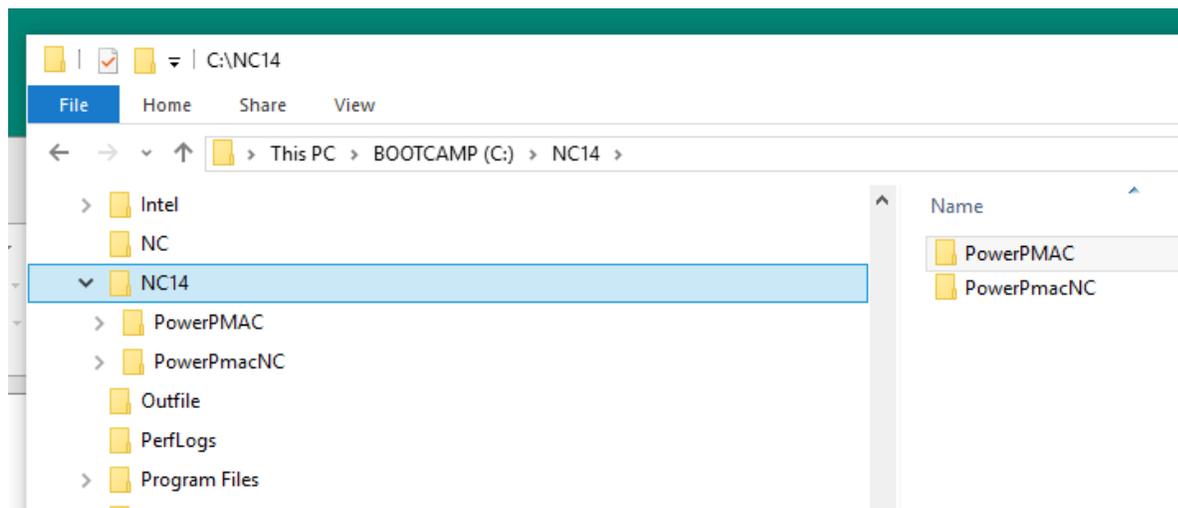
Copy directory and files from GitHub Clone directory: "PowerPmacNc14-Runtime"

-Now you have a working directory (NC14) for your project.

In here you have 2 more working directories:

PowerPMAC: this has the pmac program code to download to pmac controller

PowerPmacNC: this has the NC HMI Application code for the user interface



Power PMAC Setup

Here will use the Power PMAC IDE to configure, download and setup the Power PMAC to run with the NC HMI.

Open the Power PMAC project in the IDE using the file:

```
"C:\NC14\PowerPMAC\PPCNC_ProjectSource\ PPCNC_ProjectSource.PowerPmacSuite_sln"
```

Now you should have the Project with base NC PMAC code in the project and be communicating with a Power PMAC CPU. For this example the type of Power PMAC does not matter. This will work even with a bare CPU that has no GATE hardware connected.

For this example we want to use Virtual Motors and have them setup from a configuration file NOT from Global Includes
-if you have a file "ppnc_virtualmotors.pmh" in the "\PMAC Script Language\Global Includes" REMOVE it.

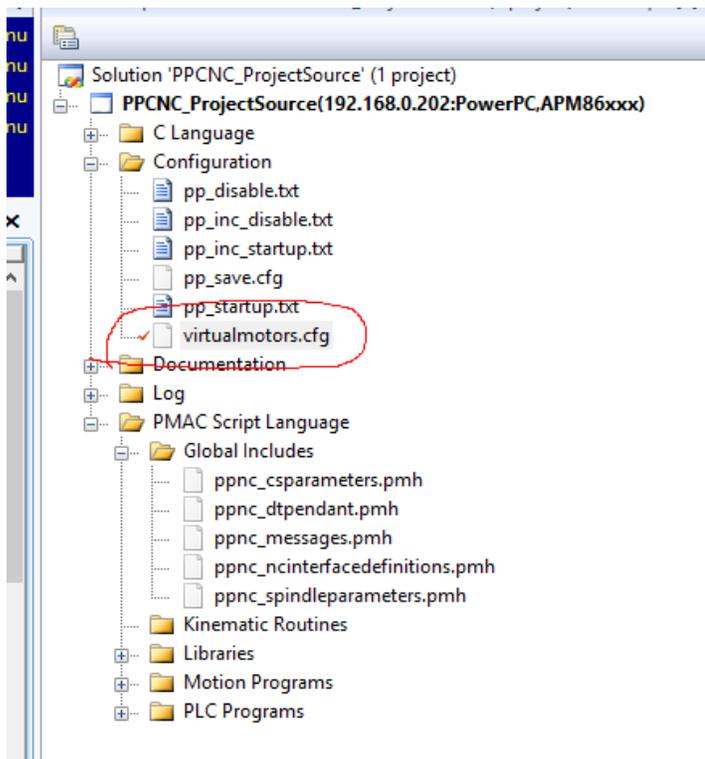
Now create a Configuration file for using Virtual motors. (this assumes the project does Not now have one)

- In "Configuration" Right-Click – Add – New Item – Custom Config
- use file name "virtualmotor.cfg" – Add
- now you should have a blank configuration file open called "virtualmotors.cfg"
- from next section of this document copy/paste the virtual motors code to this config file
- if you are using a Power PMAC CPU with NO Gate hardware edit the line below

```
Sys.CPUTimerIntr=1 //set to 1 if not GATEs in system
```
- save the project with this new file

Now select this file to be downloaded with Configuration:

- Right-Click on the file "virtualmotors.cfg" and select "Check to Download File"
- now a small RED Check should be next to the file in the Tree



-Clear the Power PMAC of any existing Project code “\$\$\$***”

-Build and Download this new Project to Power PMAC

-Right-Click on project at top of tree and select “Build and Download”

-watch the Output window at bottom of IDE it should end with No errors “Download Successful.”

-note this downloaded all the PLC’s and Program logic bit did NOT load any Configuration

-Now Right-Click on Configuration and select “Download Config Files”

-watch Output window for “Downloading Configuration Files is Successful!”

-be sure you had the “virtualmotors.cfg” file with Red check when did this, if not it was NOT loaded

-Now Save the PMAC with it’s current Logic and Configuration – “save”

-Reset the PMAC “\$\$\$”

-Now you should be able to Jog motors 1 – 4 and see Position update as if a real motor was running...

-you have your Power PMAC running with 4 virtual motors

-the NC14 software can now connect with this PMAC and run it as if a real NC machine

Send	On Demand	Command	Response
	<input type="checkbox"/>	Sys.ServoCount	72011
	<input type="checkbox"/>	Sys.CpuTemp	43
	<input type="checkbox"/>		

Position	Position
#1	160,081.00 mu #1
#2	0.00 mu #2
#3	0.00 mu #3
#4	0.00 mu #4

```
pp_custom_save.tp virtualmotors.cfg
//*****
// PowerPMAC-NC Project Source File.
// 4 Axis Setup File to allow the use of a PowerPMAC CPU Only in Virtu
// - will run on CPU Only and simulate 4 virtual motors
// Original File Date: November 22, 2013
//*****
//*****
// Comments / Revisions:
// 112113 - Initial Submit to GitHub
// 020414 - Added Axes 5-8
//
//
//*****
Sys.WpKey = $AAAAAAAA
Sys.CPUTimerIntr=0 //set to 0 if have Gate, 1 if just
Sys.WpKey = 0

Motor[1].pDac=sys.udata[1].a
EncTable[1].pEnc=sys.udata[1].a
Motor[1].Ctrl=sys.PosCtrl
Motor[1].ServoCtrl=1
Motor[1].FatalFeLimit=0
```

Power PMAC Virtual Motors Configuration File code

Copy / Paste the below program lines to your virtualmotors.cfg file

```

/*****
// PowerPMAC-NC Project Source File.
// 4 Axis Setup File to allow the use of a PowerPMAC CPU Only in Virtual Motor Mode.
// - will run on CPU Only and simulate 4 virtual motors
// Original File Date: November 22, 2013
*****/

/*****
// Comments / Revisions:
// 112113 - Initial Submit to GitHub
// 020414 - Added Axes 5-8
//
//
*****/

Sys.WpKey = $AAAAAAA
Sys.CPUTimerIntr=0 //set to 0 if have Gate, 1 if just cpu with NO Gates
Sys.WpKey = 0

Motor[1].pDac=sys.udata[1].a
EncTable[1].pEnc=sys.udata[1].a
Motor[1].Ctrl=sys.PosCtrl
Motor[1].ServoCtrl=1
Motor[1].FatalFeLimit=0
Motor[1].pAmpFault=0
Motor[1].pAmpEnable=0
EncTable[1].type=1
EncTable[1].index1=0
EncTable[1].index2=0
EncTable[1].index3=0
EncTable[1].index4=0
EncTable[1].index5=0
EncTable[1].pEnc1=Sys.udata[1].a
EncTable[1].pEnc=Sys.udata[1].a
EncTable[1].MaxDelta=0
EncTable[1].ScaleFactor=1
EncTable[1].TanHalfPhi=0
EncTable[1].CoverSerror=0

Motor[2].pDac=sys.udata[2].a
EncTable[2].pEnc=sys.udata[2].a
Motor[2].Ctrl=sys.PosCtrl
Motor[2].ServoCtrl=1
Motor[2].FatalFeLimit=0
Motor[2].pAmpFault=0
Motor[2].pAmpEnable=0
EncTable[2].type=1
EncTable[2].index1=0
EncTable[2].index2=0
EncTable[2].index3=0
EncTable[2].index4=0
EncTable[2].index5=0
EncTable[2].pEnc1=Sys.udata[2].a
EncTable[2].pEnc=Sys.udata[2].a
EncTable[2].MaxDelta=0
EncTable[2].ScaleFactor=1
EncTable[2].TanHalfPhi=0
EncTable[2].CoverSerror=0

Motor[3].pDac=sys.udata[3].a
EncTable[3].pEnc=sys.udata[3].a
Motor[3].Ctrl=sys.PosCtrl
Motor[3].ServoCtrl=1
Motor[3].FatalFeLimit=0
Motor[3].pAmpFault=0
Motor[3].pAmpEnable=0
EncTable[3].type=1

```

```

EncTable[3].index1=0
EncTable[3].index2=0
EncTable[3].index3=0
EncTable[3].index4=0
EncTable[3].index5=0
EncTable[3].pEnc1=Sys.udata[3].a
EncTable[3].pEnc=Sys.udata[3].a
EncTable[3].MaxDelta=0
EncTable[3].ScaleFactor=1
EncTable[3].TanHalfPhi=0
EncTable[3].CoverSerror=0

Motor[4].pDac=sys.udata[4].a
EncTable[4].pEnc=sys.udata[4].a
Motor[4].Ctrl=sys.PosCtrl
Motor[4].ServoCtrl=1
Motor[4].FatalFeLimit=0
Motor[4].pAmpFault=0
Motor[4].pAmpEnable=0
EncTable[4].type=1
EncTable[4].index1=0
EncTable[4].index2=0
EncTable[4].index3=0
EncTable[4].index4=0
EncTable[4].index5=0
EncTable[4].pEnc1=Sys.udata[4].a
EncTable[4].pEnc=Sys.udata[4].a
EncTable[4].MaxDelta=0
EncTable[4].ScaleFactor=1
EncTable[4].TanHalfPhi=0
EncTable[4].CoverSerror=0

// Disable overtravel limit inputs
// May be needed if there are no physical switches present
Motor[1].pLimits=0
Motor[2].pLimits=0
Motor[3].pLimits=0
Motor[4].pLimits=0

// Motor CS Def Scale Factors
Motor[1].PosSf=1
Motor[1].Pos2Sf=1
Motor[2].PosSf=1
Motor[2].Pos2Sf=1
Motor[3].PosSf=1
Motor[3].Pos2Sf=1
Motor[4].PosSf=1
Motor[4].Pos2Sf=1

// Set derivative gain term in servo loop to zero
// This is a Type 1 servo (single integration); does not need Kd
Motor[1].Servo.Kvfb=0
Motor[2].Servo.Kvfb=0
Motor[3].Servo.Kvfb=0
Motor[4].Servo.Kvfb=0

// Lower proportional gain term from default
Motor[1].Servo.Kp=40
Motor[2].Servo.Kp=40
Motor[3].Servo.Kp=40
Motor[4].Servo.Kp=40

// [MOTOR_DEF]
&1#0->0
#1->1000X
#2->1000Y

//**** END virtualmotors.cfg File ****

```

Power NC Software Setup

Here we will do initial configuration and setup of the Power NC application to connect and work with the Power PMAC.

- Make sure the USB hardware key (dongle) is installed on a USB port of the PC
- Now while working In NC14 HMI working directory “\NC14\PowerPmacNC\”
- **Rename** the file “Reference Messages.xml” to “Messages.xml”
 - this file contains the user messages that are displayed while running the HMI
 - OEM/Integrator will likely edit this with custom messages for his project
- **Rename** the file “Reference PowerPmacNC.ini” to “PowerPmacNC.ini”
 - this file has basic machine configuration information used when NC14 starts
 - we need to edit this with Notepad for our simple Test NC14 Machine
- Open this file in Notepad:**

Edit the lines below that we have changed for this project:

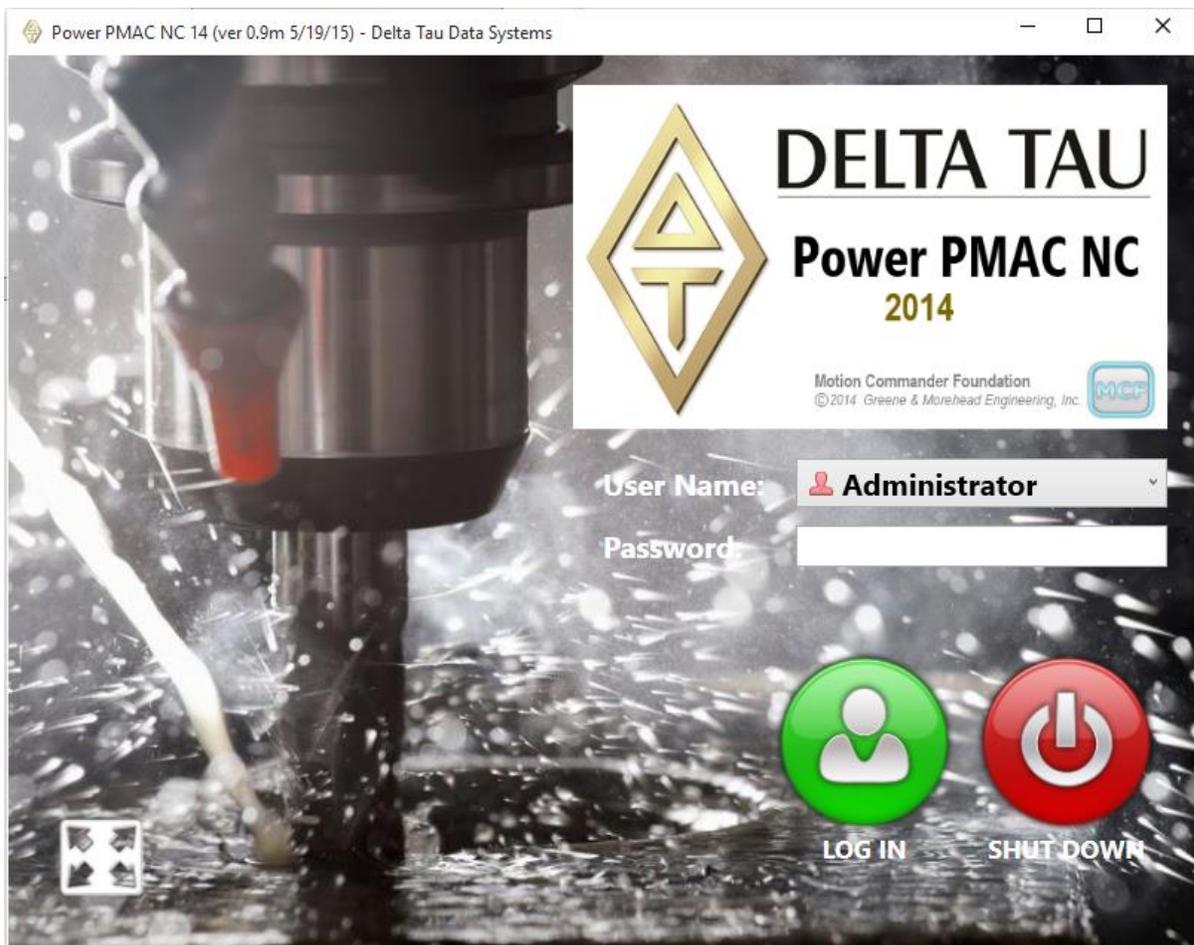
AxisLabels=X,Y,Z

MotorNumbers=1,2,3

Save this now edited INI file

-Now with Power PMAC powered and Ethernet connection to PC start the NC14 application?

- **Start** with “\NC14\PowerPmacNC\PowerPmacNC.exe”
- Note the Version number in Top Left corner.
- **Select** “LOG IN” button.



- Select the Machine View Tab in Upper Left corner
- this gets us to Machine View for further system configuration

-Here are some typical things to get setup and going:

Controller – Address – set to your Power PMAC IP Address – select Apply

Settings – Start Up – Go Online and Run at Start-Up

Settings – Jog Speed1 – 50 – Apply, Repeat for 2-5 with increasing values for faster jog speeds

Settings – Max Feedrate – 1000 – Apply (likely adjust this later for proper value)

-Lets change the Skin:

-Double Click on Settings – Skin – Click on small File Folder bottom Left of Dialog

-Select MidnightBlue.xaml – Open – close small dialog [x] upper left, now done with this.

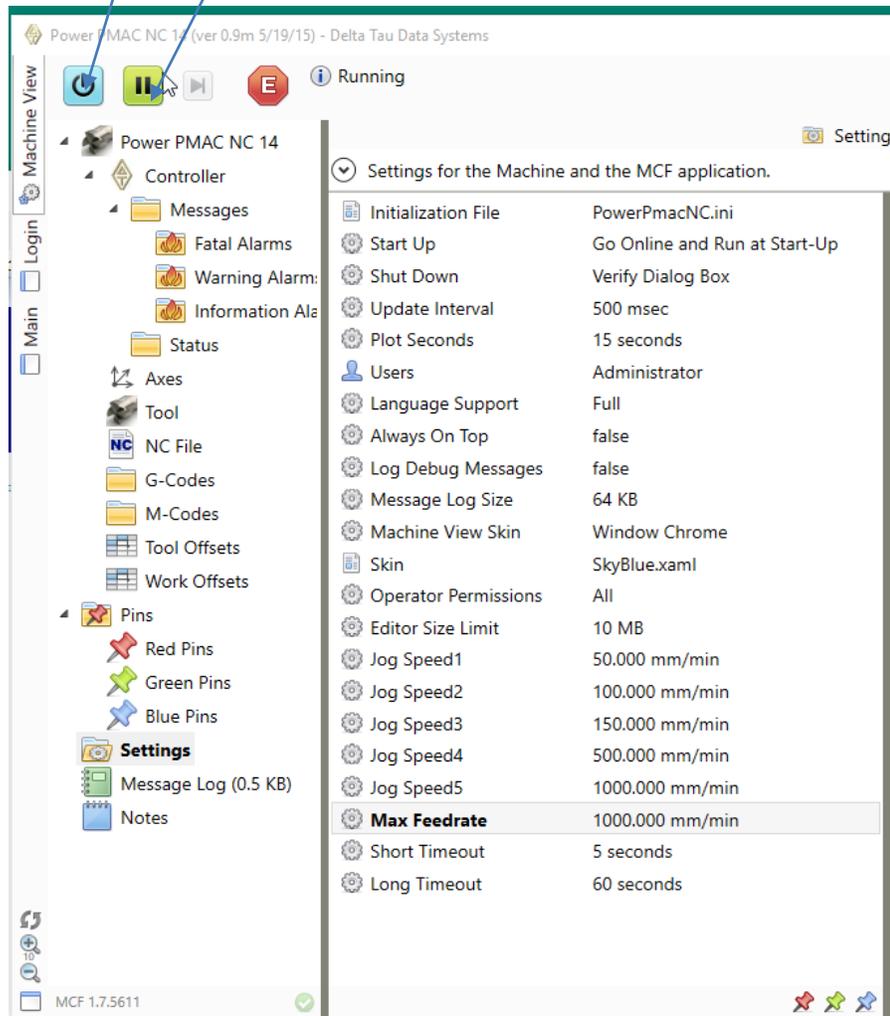
-Now get NC14 to connect with Power PMAC and start the code running:

-select the Power Button at top Left

-select the Run Button next to Power

-if all goes well the Run Button is now a Pause button and Green

Note the MCF (Motion Commander Foundation) version in lower left corner for reference.



Running the Machine

-Now **Select** the “Main” tab on Left side of Panel

-Now you are at the Main Operator HMI Panel for the NC14 application.

-Should also now be connected to the Power PMAC and live with Virtual motors.

-Select the “Reset” button on HMI.

-Select “Manual” Tab then Home Button

-If you have Alarm message, such as Jog Speeds at Zero – go to Alarms Tab and clear the message

-Jog a motor:

Go to Manual – Select “x2” speed – select “X” – “+Jog”

Now should see the position updating in both the NC14 HMI and in the Power PMAC IDE

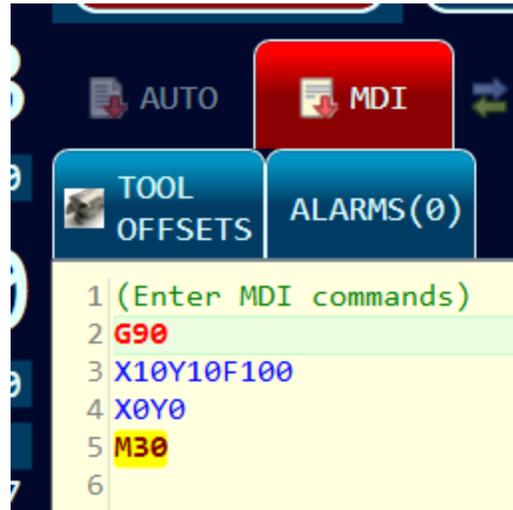
The image shows two overlapping windows. The left window is the Power PMAC IDE, displaying a configuration file named 'virtualmotors.cfg'. The right window is the HMI, showing the 'Main' tab with a 'MACH POS' display. The HMI display shows the following data:

Axis	REL POS	MACH POS	CMD POS
X	0.000	0.380	
Y	0.000	0.000	
Z	0.000	0.000	

The HMI also displays various controls including FEEDRATE (0%, 100%), SPINDLE (100%), and JOG SPEED (x1, x2, x5). The 'X' axis is selected for jogging, and the '+JOG' button is highlighted.

-Do a NC Gcode Move in MDI Mode:

- Select MDI Tab
- Enter desired moves such as this
- Press Cycle Start button
- Watch the motor position as moves made
- Note lower window that shows lines as executing.



What Next ?

Now you should have a working NC HMI with Power PMAC. It should Jog motors and run NC code as expected. Next you will want to modify this simple example to work with a real machine. Likely with more motors, inputs, outputs and special logic needed. How do I get there?

PMAC Logic:

- As much as possible do not modify the exiting ppnc PMAC files
 - so later if updating to a newer release of these file a minimum of changes would need to be made
 - it is best to create new PLC's, Motion Programs, Sub-Routine and Header files

- Create a new configuration file similar to the "virtualmotors.cfg" created earlier.
 - this will now need to have configuration for your actual machine hardware
 - you can use the IDE to help with this configuration
 - this file should have all the needed system setup such as Clocks and Motor setup such as Gains
 - when this file is downloaded to the PMAC with needed hardware you should be able to control the motors and other IO as needed for this machine

- Create custom PLC / Motion Programs as needed.
 - likely you will edit the Initialization PLC to handle machine power up settings
 - create some Reset/Power UP logic this is triggered in the "ppnc_function.plc" "UI_Reset" command
 - create custom Homing PLC or modify existing basic logic
 - create some error handling plc
 - modify Spindle plc if needed
 - add other PLC's for custom machine functions

- Edit the NC Sub-Routine files as needed for customer codes
 - these are in the \Libraries section E.G. ppnc_mcodes.pmh, ppnc_gcodes.pmh

- When done your logic should be able to react and work with the built-in HMI logic for functions like Reset and Homing.

Power NC HMI:

- Go thru the Machine View – Settings Page to be sure all settings are as should be for this machine.
 - these settings are all saved in file "PowerPmacNC_Settings.xml"
- Edit the PowerPmacNC.ini file as needed for your machine startup settings
- Edit the Messages.xml file for your custom messages to the user
- Refer to the Power NC Developer Manuals for details on how to customize features such as messages

Deploy to Actual Target Machine

How to deploy to my actual machine PC running HMI and PMAC?

-You have 2 sets of code here to work with and deploy:

1. PMAC code:

- depending on your machine you might have the PMAC source code the machine PC or not
- assuming you do then simply create directory on machine PC and put the PMAC code there
- then using the PMAC IDE (either Power or Turbo) download PMAC code to your actual controller
- here we assume you have fully developed your machine logic and configuration in the PMAC code already

2. HMI code:

- starting with the base HMI code from the GitHub download in dir \PowerPMACNC
- copy all the files and directories from \PowerPMACNC to your HMI directory on machine PC
 - directory name can be what you like E.G. C:\HMI\...

-note the files that are unique to your system and require your customization are:

PowerPmacNC_Settings.xml

- This file is created when you edit the settings inside the NC14 – Machine View – Settings
- for example in this Machine View dialog you setup “Jog Speed1” and value is saved in this file
- so when you get machine setup this file needs to be saved

PowerPmacNC.ini

- this file is not installed by default
- you must copy and rename from “Reference PowerPmacNC.ini” file included with software
 - you will rename this “PowerPmacNC.ini” and edit it for your configuration

Messages.xml

- this file is not installed by default
- you must copy and rename from “Reference Messages.xml” file included with software
 - you will rename this “Messages.xml” and edit it for your configuration
- you will edit this file so that custom messages that match your machine will be use

-So to deploy your finished NC project you will need to:

- Download PMAC program files to your PMAC
 - this source or config files do not need to be on the target HMI PC, just loaded to the PMAC
- Create the Application directory for NC14 HMI on your target system that connects to and controls the machine PMAC.
- Copy the NC14 Directory and files to this dir.
- Copy in your customized files that configure the NC14 to your machine:
 - PowerPmacNC_Settings.xml
 - PowerPmacNC.ini
 - Messages.xml
- Make sure the NC14 Dongle is installed on USB port of the target PC
- Power up PMAC and Target PC – Start the NC14 application
- If NC14 not configured to Auto connect and start go to Machine View to do this.
- Now your machine with PMAC should be running the NC14 application