

Power PMAC IDE

Typical Screen for **Integrated Development Environment (IDE)**



DELTA TAU
Data Systems, Inc.

The screenshot displays the Power PMAC IDE interface with several windows:

- User Written Code:** A C++ source code file named `usercode.c` with the following content:

```
// Project PowerPMAC Firmware
// Delta Tau Data Systems, Inc.
// Copyright 2007. All Rights Reserved.
//
// SUBSYSTEM:      Uses Devco drivers
// FILE:           usercode.c
// TEMPLATE AUTHOR:  Brady Beaulieu
//
// OVERVIEW
//
// This file is where exportable user code can be placed.
// To make a function available as a user servo, do three steps
//
// 1.) Prototype the function user_func(void *,void *)
// 2.) Export the function EXPORT_SYMBOL(user_func)
// 3.) Make sure user servo has been loaded with
//    linux-2.6.15-misc01/ppmac/usercode.c
//
// #include <linux/module.h>
// #include <linux/sched.h> // Global PL/Py Extensions and structures
//
// int heavy_user_servo(MotorData *Mptr);
// int user_plc_rtl(void *p1,void *p2,void *p3);
// int user_plc_thread(void *p1,void *p2,void *p3);
// int user_phase(void *);
// static int j = 0,k = 0;
//
// EXPORT_SYMBOL(heavy_user_servo);
// EXPORT_SYMBOL(user_plc_rtl);
// EXPORT_SYMBOL(user_phase);
// EXPORT_SYMBOL(user_plc_thread);
//
// int user_phase(void *Mptr)
// {
//     return 1;
// }
```
- Terminal:** Shows the terminal output with the following text:

```
Welcome to Power PMAC Terminal
Select device to start communication
Communication to Power PMAC successful
Motor[0] ActPos
Motor[0] ActPos=1000
Motor[0]
AbortTt
AbortTt
ActMasterDes
ActPos
ActPos2
AdvMask
AdvGain
Algo.
```
- Position:** Displays position data for four motors:

```
#1 1,000.00 Cts
#2 0.00 Cts
#3 0.00 Cts
#4 0.00 Cts
```
- F.E. (Following Error):** Displays following error data for four motors:

```
#1 0.00 Cts
#2 0.00 Cts
#3 0.00 Cts
#4 0.00 Cts
```
- Velocity:** Displays velocity data for four motors:

```
#1 0.00 Cts/msec
#2 0.00 Cts/msec
#3 0.00 Cts/msec
#4 0.00 Cts/msec
```
- Watch:** A table showing system variables and their values:

Command	Response	Comment	Macro definition
#j	1000		
Sys.ServoCount	Sys.ServoCount=15632824		
- Motor Status:** A table showing the status of four motors:

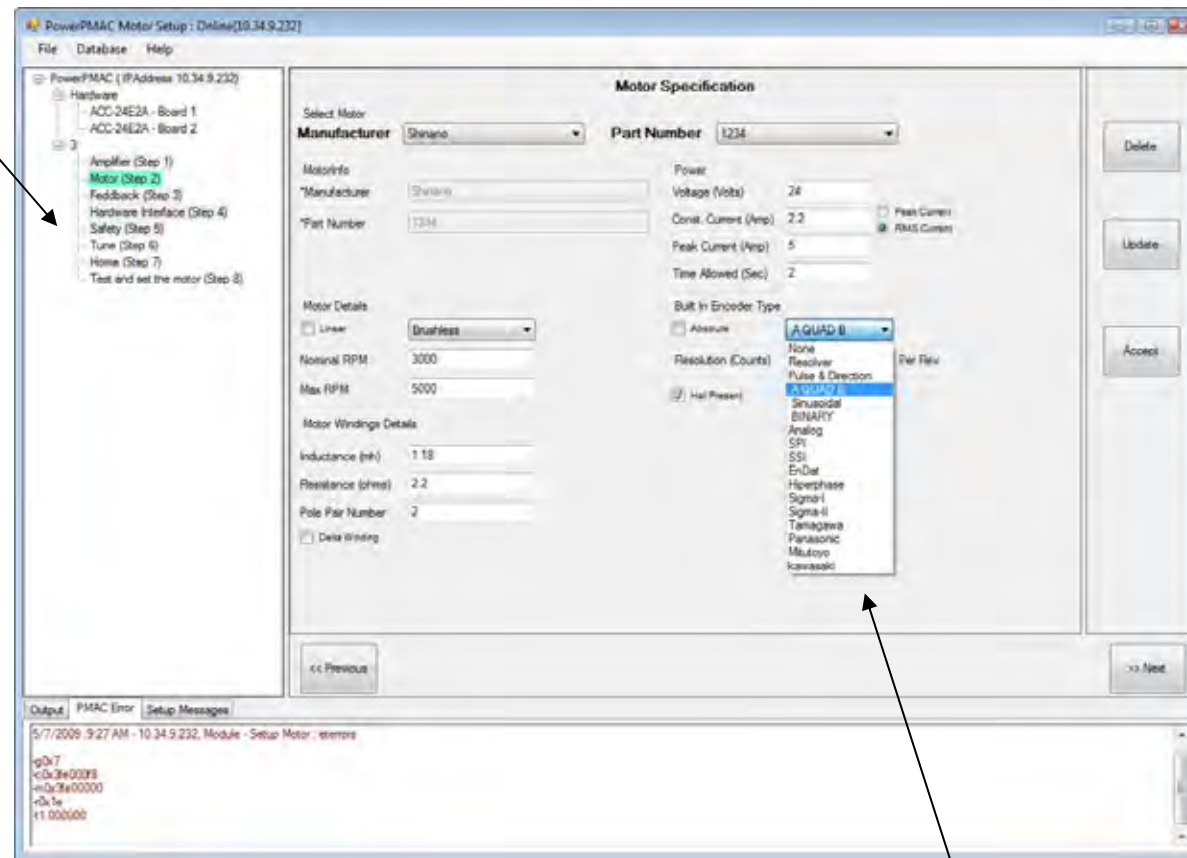
Motor #	Status
1	Trigger move
2	Negative end limit set
3	Warning following error
4	Stopped on position limit
1	HomeComplete
2	Closed loop mode
3	Impedance
4	Block request
- Message Window:** Shows the output of the `Y212` developer edition, with the message: "Developer edition, all third-party packages allowed to load."
- Project:** A Solution Explorer window showing the project structure for `PowerPMAC`, including folders for `References`, `OPCL`, `Hardware`, `Motion Program`, `PROCL.org`, `PLC`, `PLC1.plc`, `PPMACLib`, `Properties`, `RT_OPLC`, and `usercode.c`.





Power PMAC IDE Hardware Setup

Intuitive and Simple
Step-by-Step procedures



Only allows possible
configurations to be displayed

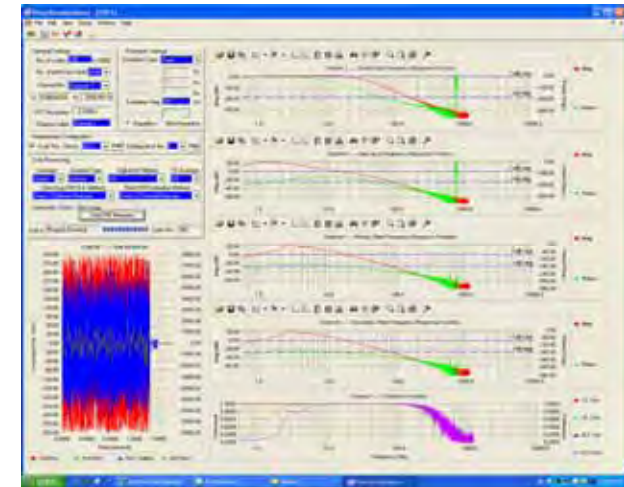
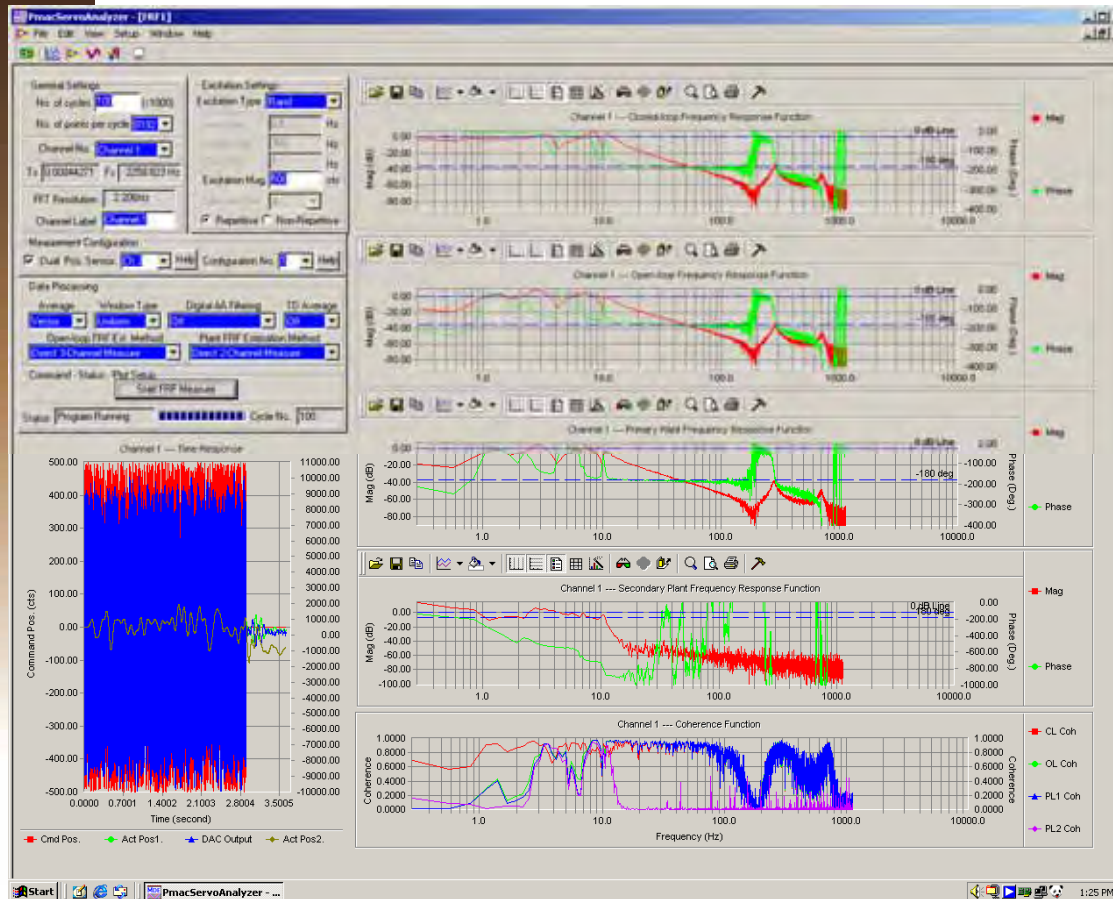
Database structure for
previously used hardware
or resource library



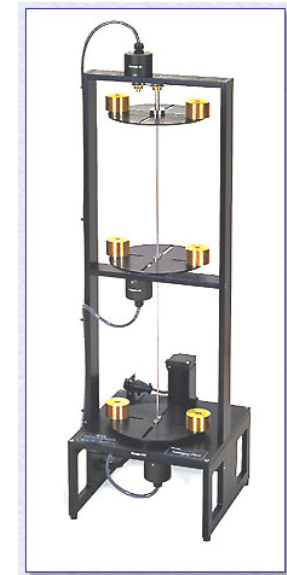


Dynamic Servo Analyzer for Turbo & Power PMAC

Bode Plot generated for system identification by PMAC Dynamic Servo Analyzer for a **Flexible** or **Rigid** body mechanical structure.



Rigid Body Mechanical Structure



Flexible Mechanical Structure

1. The Bode Plot is generated by using a selectable **white noise** or other excitation.
2. Bode plot data is used to derive the mechanical transfer function by non-linear curve fitting.
3. Final step derives an **automatically** generated PMAC servo algorithm for the mechanical structure.





Power PMAC Script Language Math and Logic

- Ability to mix all variable and data types freely
- Over 24 mathematical functions (e.g. *sin*, *sqrt*, *cbirt*, *exp*, *abs*)
- 12 math/logic operators (+, -, *, /, %, &, |, ^, >>, <<, ++, --)
- 12 assignment operators (including delayed synchronous)
- Array access to any numbered variable type
- Vector and 2D matrix operations (e.g. *inv*, *solve*) (new!)
- 8 conditional comparators (==, !=, >, <=, <, >=, ~, !~)
- Logical structures: *while*, *do..while*, *if*, *else*, *switch*
- Subroutines in motion and PLC programs: *gosub*, *callsub*, *call*
 - *Callsub* (in same program) and *call* (to separate program) permit local variables in subroutines (true argument pass and return)
 - Argument passing thru “G-code” letter/number format supported





Power PMAC Built-In Data Structures

- Main technique for user access to Power PMAC registers
 - Will largely replace I and M-variable use
 - For hardware and software registers, control and status elements
 - Key setup elements saved to flash memory (like I-variables)
 - Accessible from Script & C programs, on-line commands
- Major structures pre-defined:
 - *Sys, Motor[], Coord[], Gaten[].Chan[], CompTable[]*. etc.
 - Indexable for easy program access
 - e.g.: *Motor[Num].JogSpeed=133.3333*
 - Note that index numbers start at 0 (C convention)!
- User can substitute own name for structure element
 - e.g.: *#define M2Vj Motor[2].JogSpeed*
- User can define M-variable to any structure element





Running C/C++ Code on Power PMAC

- GNU public-domain cross-compiler built into Power PMAC IDE
- Advanced editing/debugging features in IDE
- Code can be for both real-time (e.g. servo) and non-real-time (e.g. PLC) tasks
- Can use automatically generated C code: e.g.:
 - IEC-1131 graphical programming for PLCs
 - MATLAB/Simulink Real Time Workshop™ for servos (*planned*)
- Supplied header files give access to Power PMAC data structures through shared-memory interface
 - Structure names in C are case-sensitive (unlike in Script)
- Supplied API gives access to Power PMAC function calls





IDE stands for Integrated Development Environment

The screenshot displays the PowerPmac IDE interface with several key components:

- User Written Code:** A code editor window showing C++ source code for motor control, including function definitions like `henry_user_servo` and `user_phase`.
- Terminal:** A window showing the terminal output, including the "Welcome to Power PMAC Terminal" message and the command `Motor[0].ActPos=1000`.
- Position:** A window displaying the current position of the motor in counts (Cts), showing values for #1, #2, #3, and #4.
- F.E. (Following Error):** A window showing the following error for the motor, with values for #1, #2, #3, and #4.
- Velocity:** A window displaying the current velocity of the motor in counts per millisecond (Cts/msec), showing values for #1, #2, #3, and #4.
- Watch:** A window showing a table of variables and their values, such as `Motor[0].ActPos` and `Sys.ServoCount`.
- Motor Status:** A window showing the status of the motor, including "Trigger move", "Home search in progress", and "Amplifier enabled".
- Message Window:** A window at the bottom showing system messages, such as "V811: Developer edition, all third-party packages allowed to load".
- Project:** A window on the right showing the project structure, including folders for "References", "Hardware", and "PLC".





Power PMAC PLC Program Enhancements

- Expanded math and logic capabilities of Power PMAC Script
- Subroutine and subprogram calls added
 - Local variables permit true argument passing
- Ability to command axis motion directly
 - RAPID-mode point-to-point or triggered moves
 - Can break into ongoing move at any point
 - Program execution does not pause until move over (unlike motion programs)
- Improved debugging features
 - Automatic cycle counters can be put on instructions
 - Breakpoint capability (on specified cycle count)
 - Single-step capability





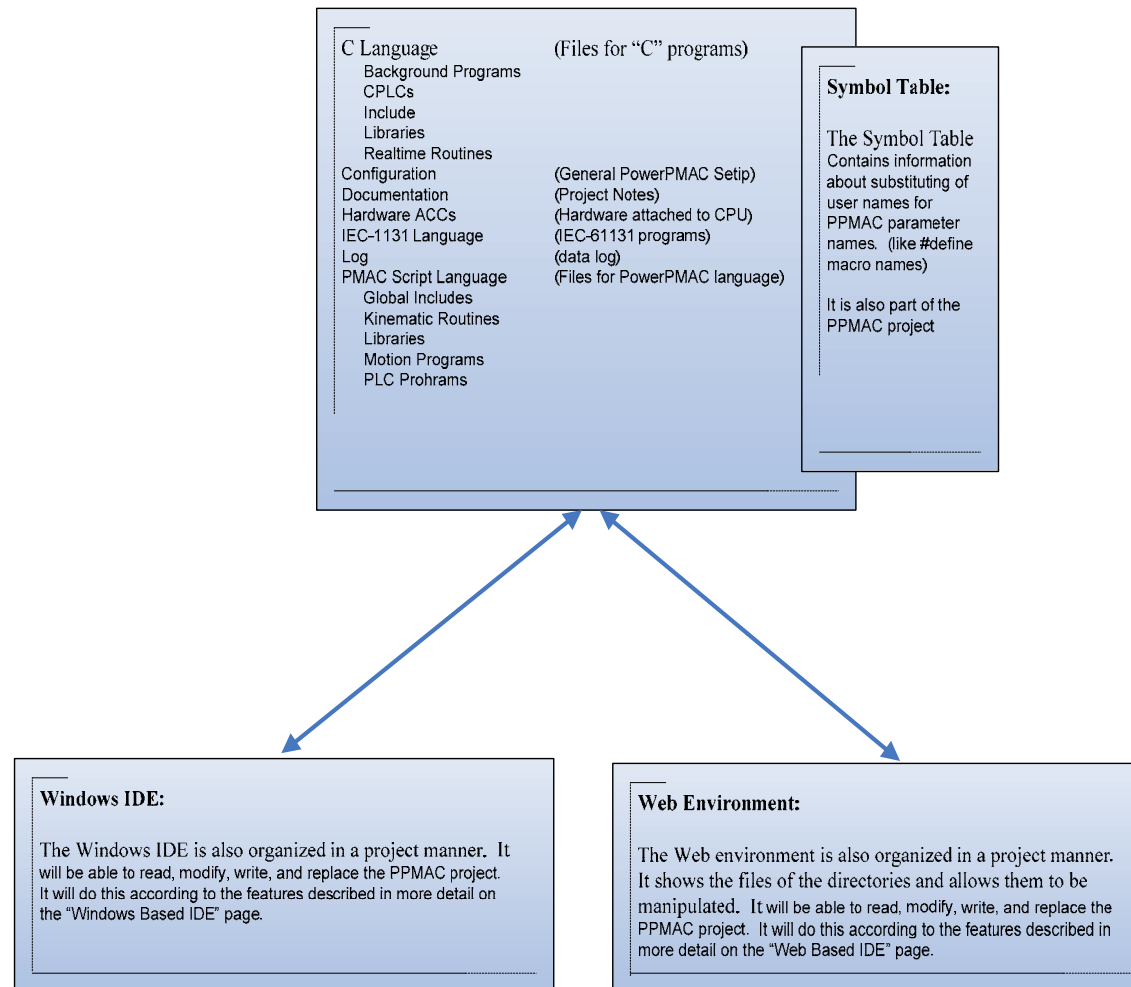
Power PMAC IDE

- Powered by VS 2008
- Full featured Windows-based system
- Limited featured web-based system
- Familiar environment for those who already program
- Easy to use for non-programmers
- Modular re-usable importable PMAC functions
- IDE components available as customer ...



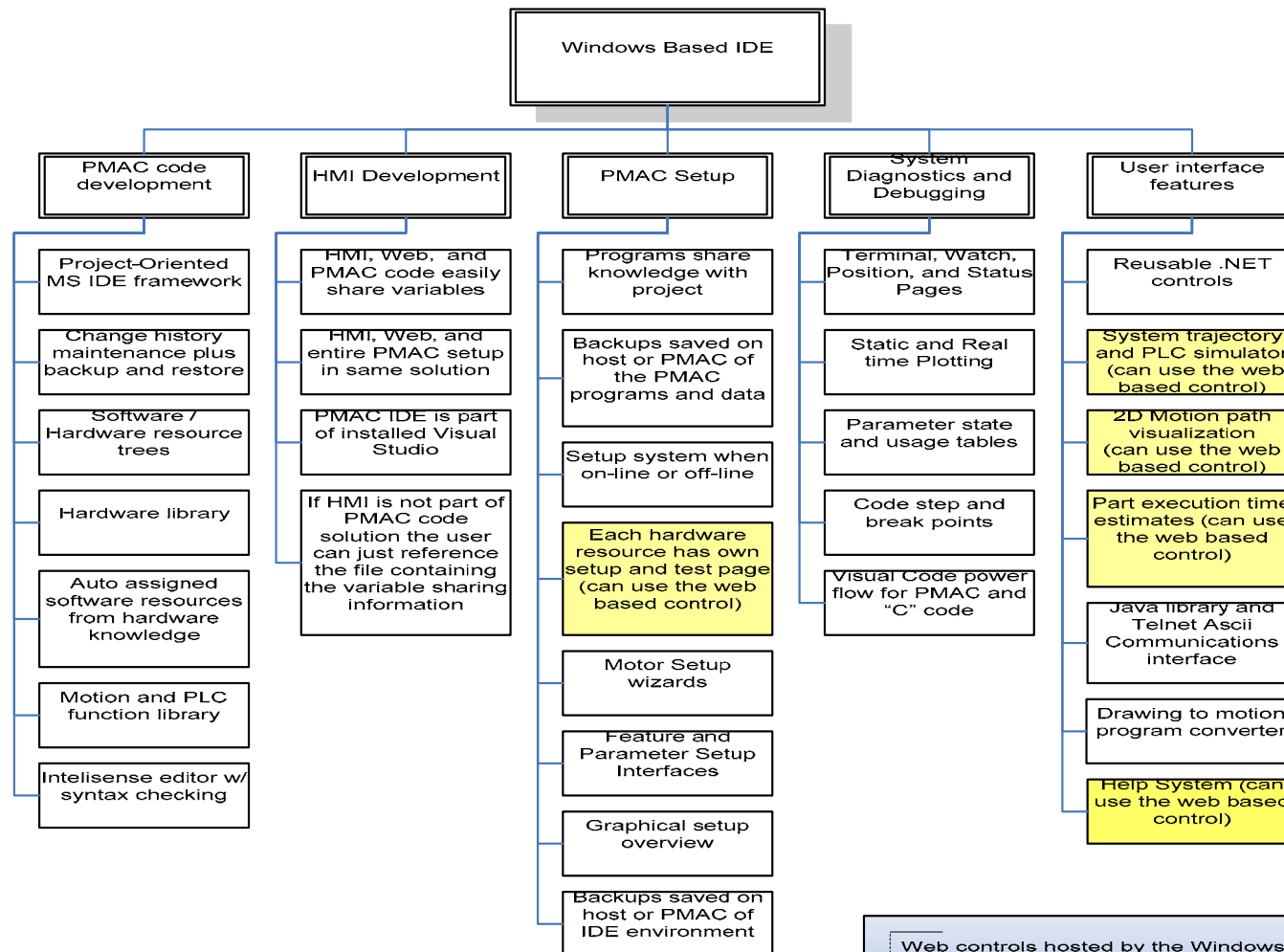


Power PMAC Project





Windows IDE Overview

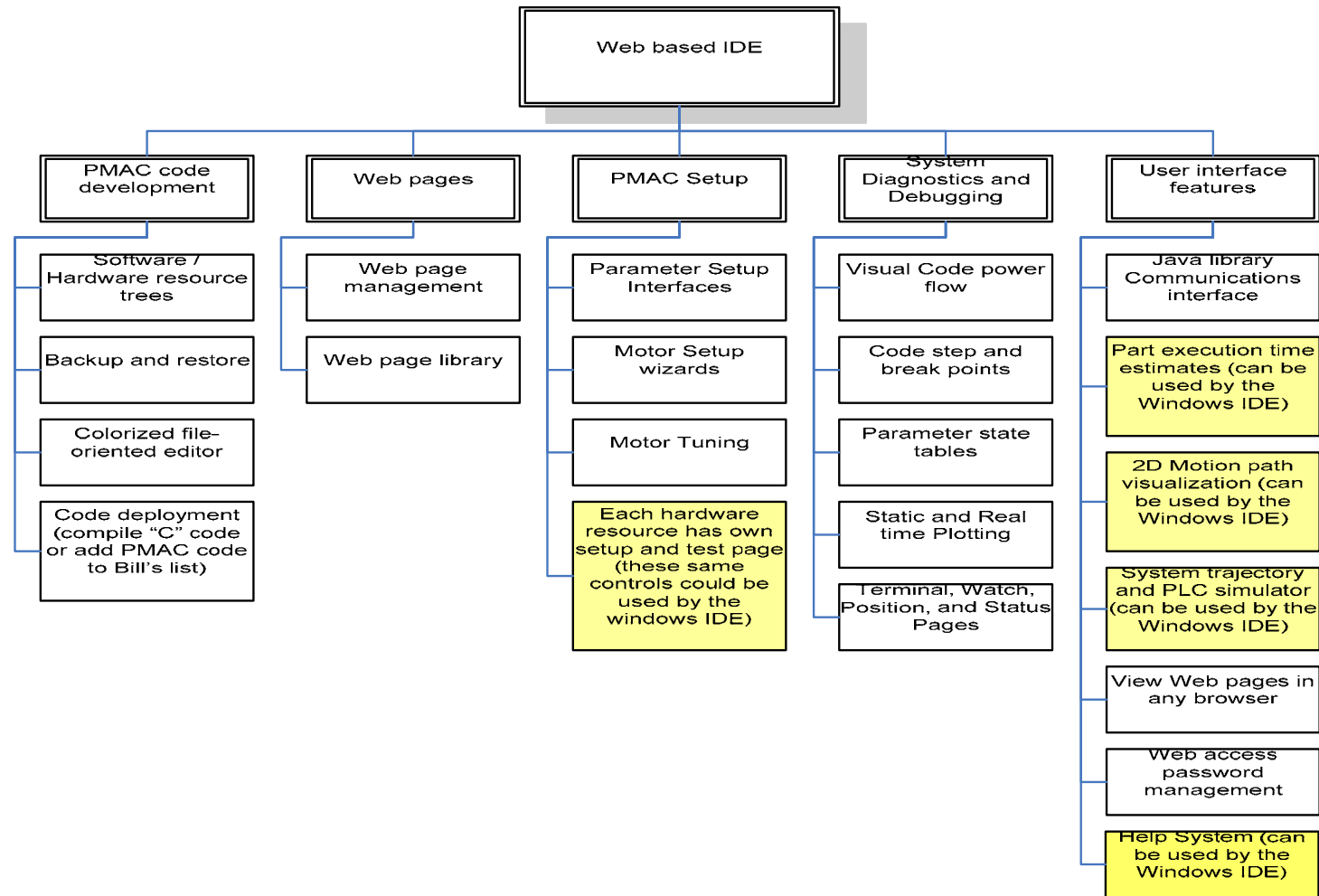


Web controls hosted by the Windows IDE pose a problem for standalone IDE operation. One solution is that a copy of the Web control and Power PMAC routines are on the local PC.





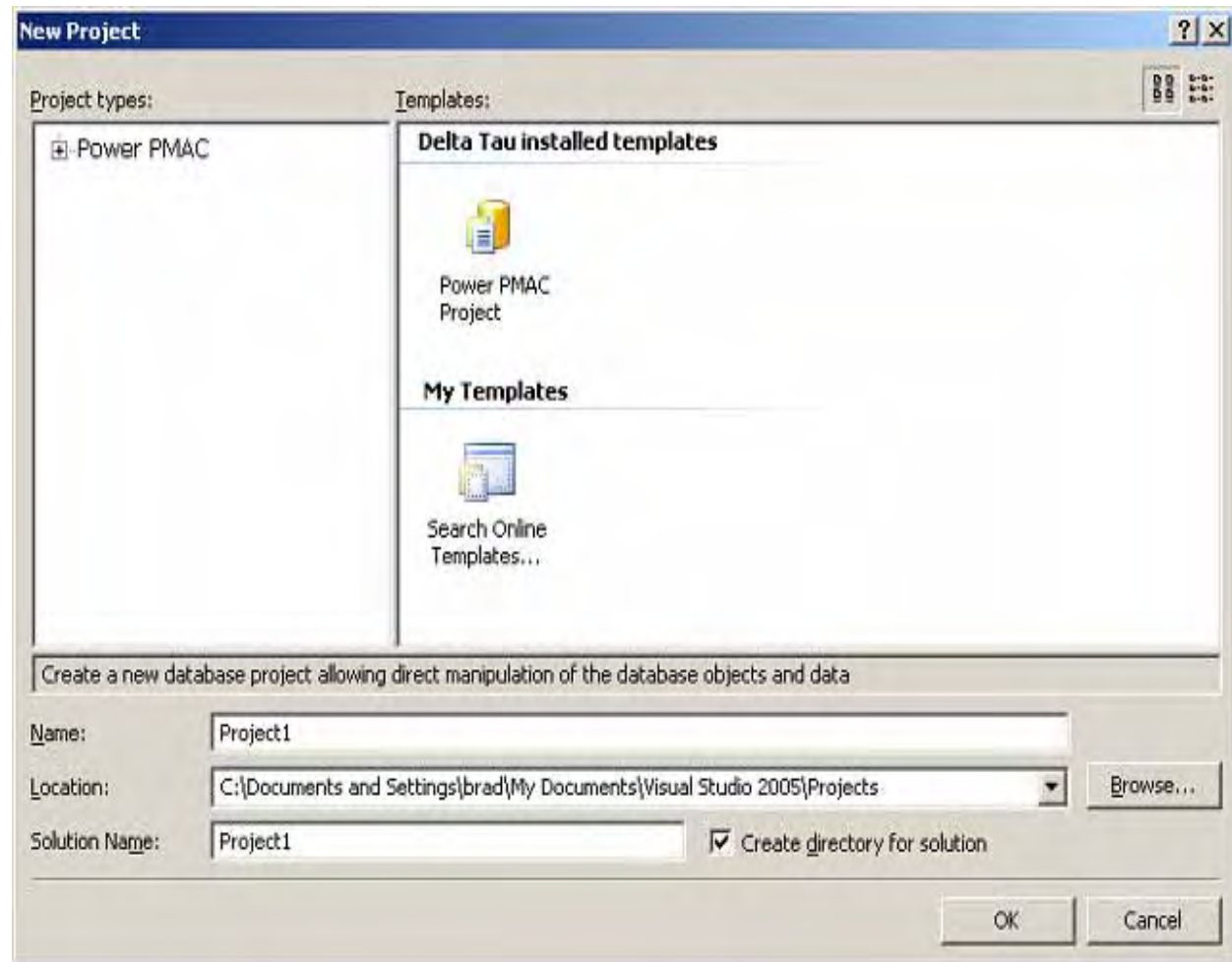
Web IDE Overview





Power PMAC Project Integration

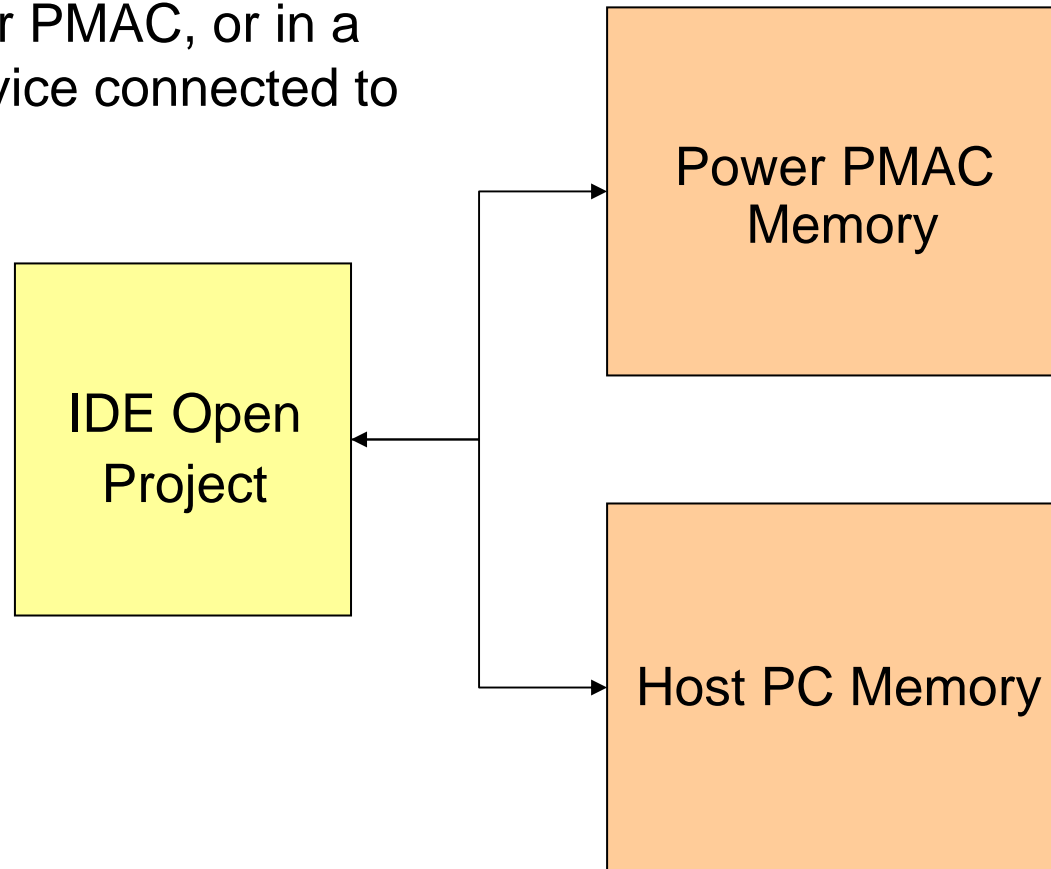
The Power PMAC project shows in the MS IDE as just another programming language.





Project Location

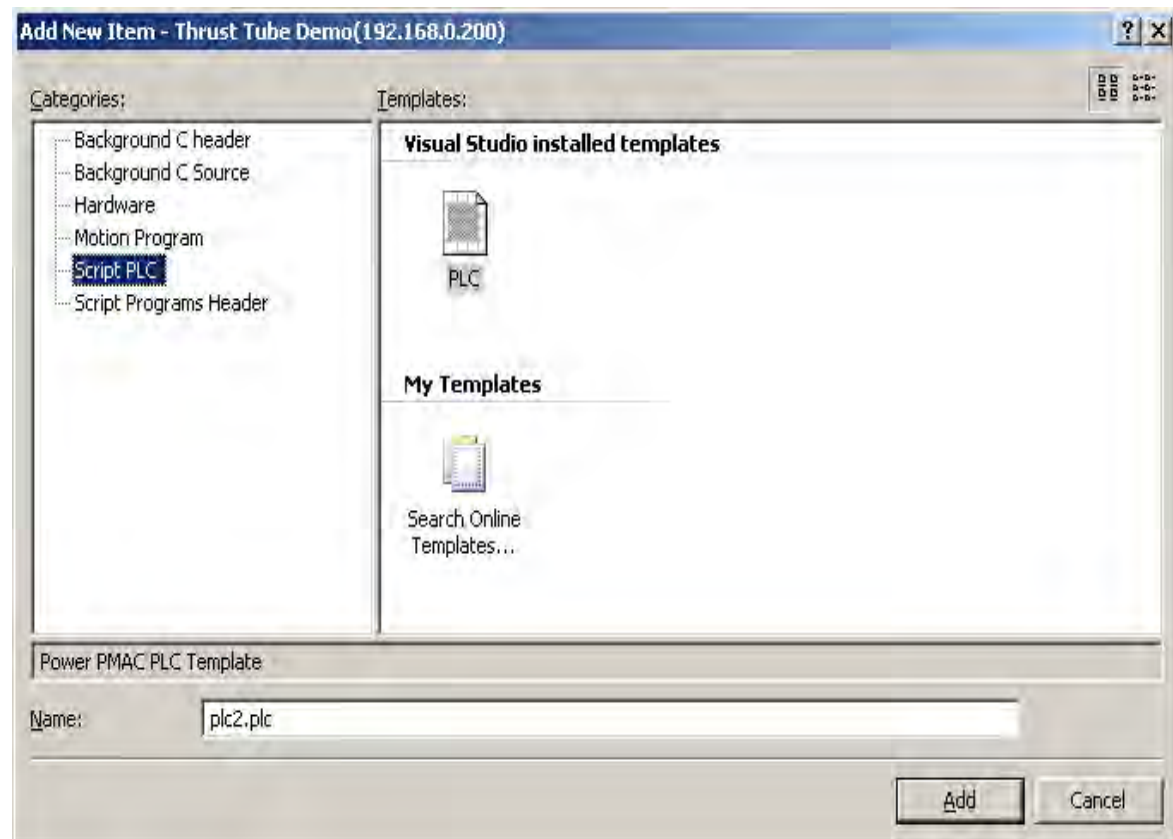
- Project data can be on the PC, in the Power PMAC, or in a memory device connected to either.





File Templates

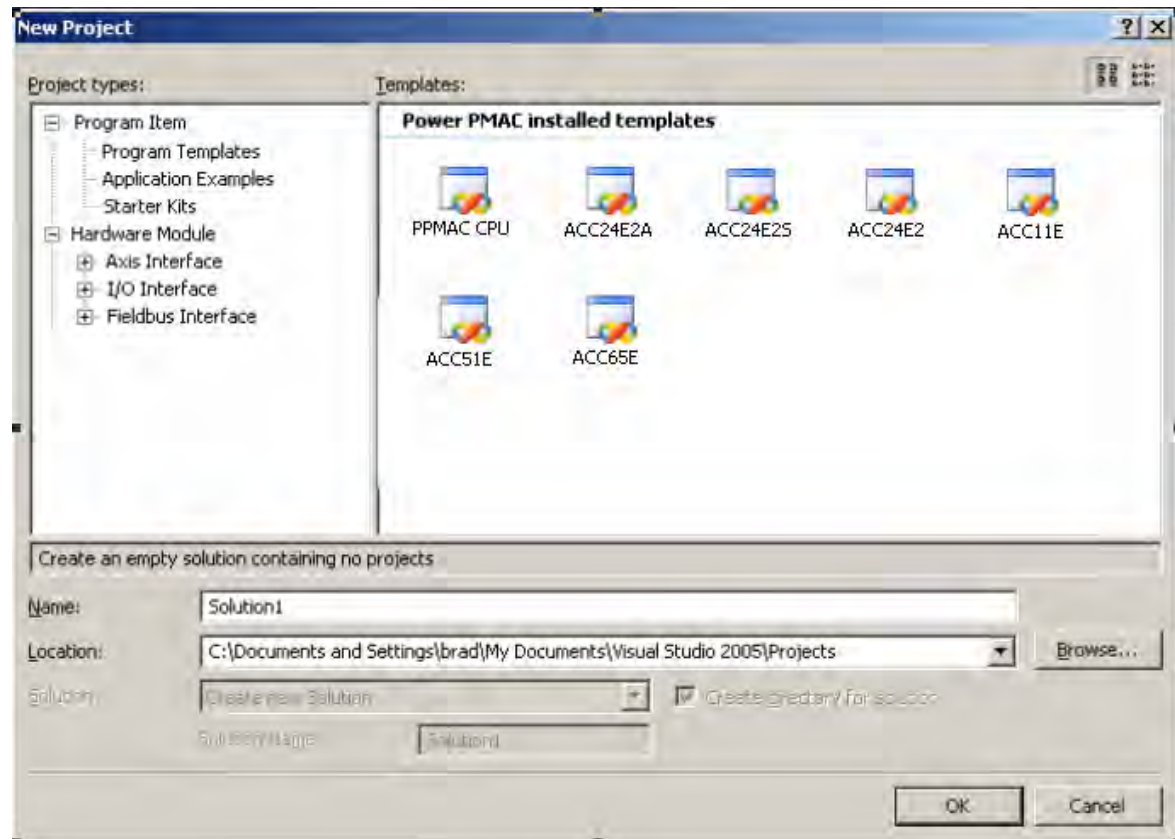
Add files from different software templates, which add structures to the active data base.



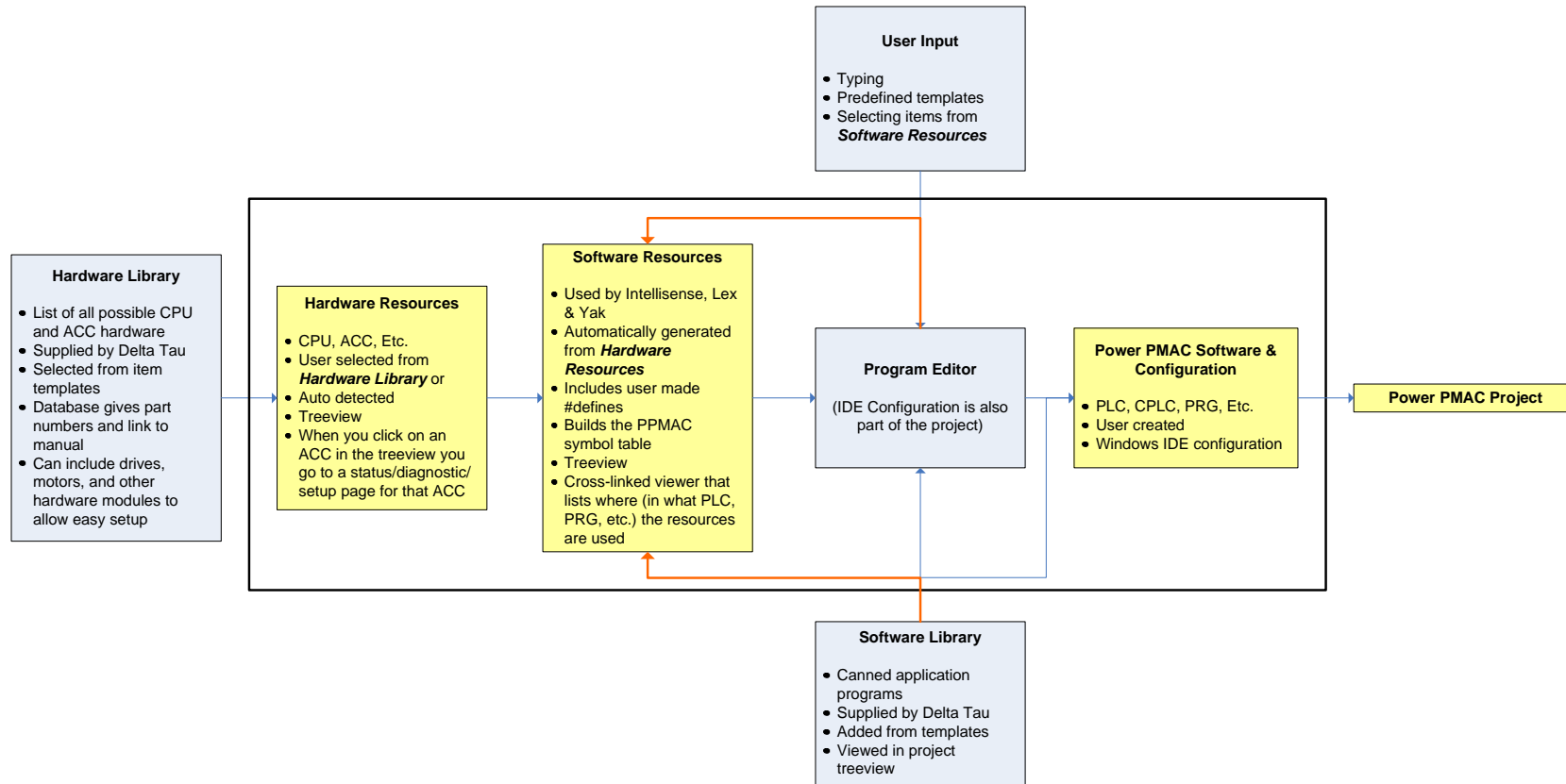


Hardware Templates

Add hardware from different hardware templates, which add structures to the active data base.



Power PMAC IDE Database

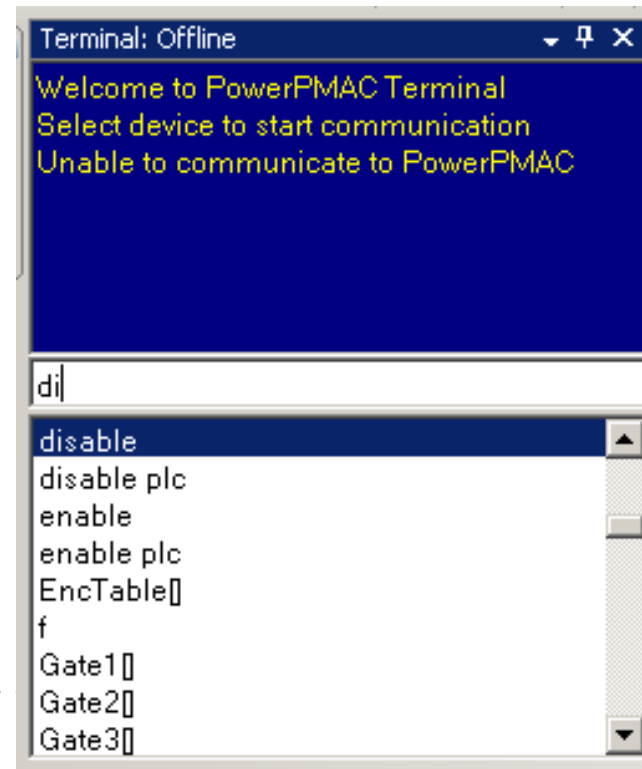




Why have a database?

A Database gives the ability of intellisense and Lex and Yac syntax parsing.

```
open prog 1
linear;
abs;
tm1000;
ta200;
ts0;
i|
if (1<2)
  if({conditional true expression})
    Conditional IF Branch
  ii
inc 11500;
int 6000y13 6000;
dwe11500;
x13 6000y4000;
dwe11500;
x4000y13 6000;
dwe11500;
)
close
```





Why have a database?

A Database gives the ability of integrated help.

Name	Value	Undo
Motor[2].JogTa	-10	*
Motor[2].JogTs	-50	*
Motor[2].JogSpeed	200	*
Motor[2].ProgJogPos	0	*
Motor[2].JogOffset	0	*
Motor[2].HomeVel	10	*
Motor[2].HomeOffset	0	*

Description	None
Range	Floating point
Units	msec if(>= 0) or msec^2 / motor unit (if < 0)
Default	-10(=0.1 motor unit / msec^2)

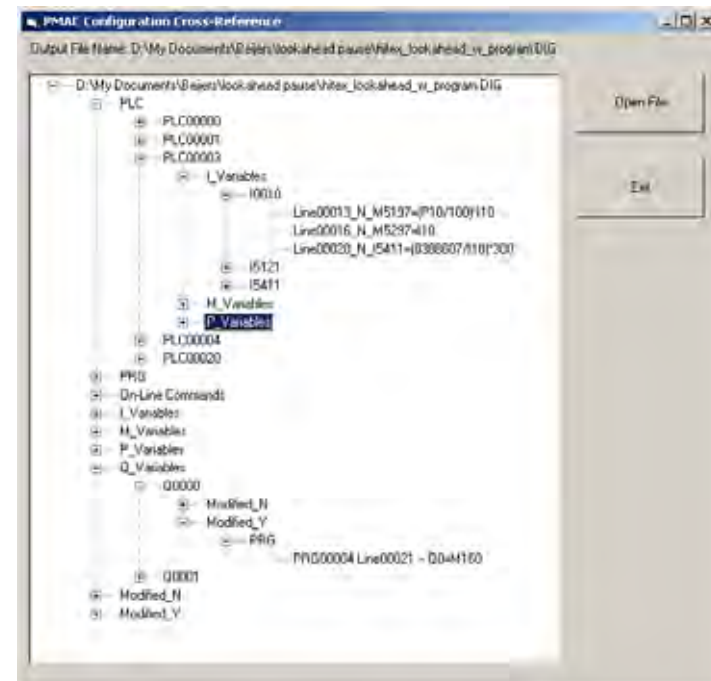




Why have a database?

A Database gives the ability of resource tracking and mapping.

Program	Variable	Location	Modified	Program Line
PLC 0	CMD	59	N	CMD*#LJ*
PLC 0	CMD	79	N	CMD*#LJ**
PLC 0	CMD	106	N	CMD*#LJ**
PLC 0	I010	19	N	I5311=(I0173*I10)/40*P80
PLC 0	I010	125	N	M5307=(I0173*I10)/40*P80
PLC 0	I010	16	N	M5307=I10
PLC 3	I010	20	N	I5411=(I036607/I10)*300
PLC 20	P0021	7	Y	P21=1
PLC 20	P1013	30	N	IF(P1013=0)
PRG 1	M0001	18	Y	M1=4
PRG 1	M0001	19	Y	M1=5
PRG 1	M0003	2419	N	N1205G1X(M2+878.59)Y(M3+1060.75)
PRG 1	M0003	2421	N	N1206G1X(M2+852.56)Y(M3+1057.59)
PRG 1	M0003	2423	N	N1207G1X(M2+823.8)Y(M3+1052.91)
PRG 1	M0003	2425	N	N1208G1X(M2+793.84)Y(M3+1060.75)
PRG 1000	I015	16	Y	I15=I17
PRG 1000	I015	34	Y	I15=I17
PRG 1000	I017	15	Y	I17=(I.2/P1062)*P201
PRG 1000	I017	15	N	I15=I17
PRG 1001	P1041	36	N	WHILE(P1041)=4)WAIT
PRG 1001	P1076	4	Y	P1076=ABS(P59.P5)
PRG 1001	P1076	5	N	IF(P1076>150)
PRG 1001	P1076	5	N	P1076=360-P1076
PRG 1001	P1076	5	Y	P1076=360-P1076
PRG 1001	P1076	5	N	IF(P1076<3)





Editor Error and Bracket Matching

```
PowerPmac1 - Microsoft Visual Studio
File Edit View Project Build Debug Tools Delta
// Power P&M&C Script PLC Program
// The following Sample PLC PROGR
// Sample PLC PROGRAM
/*****

csglobal uuu;

open plc 1
if (p1=1)
{
    p2=2;

close
```

```
PowerPmac1 - Microsoft Visual Studio
File Edit View Project Build Debug Tools Delta
// Power P&M&C Script PLC Program
// The following Sample PLC PROGR
// Sample PLC PROGRAM
/*****

csglobal uuu;

open plc 1
    p1=sin(35+3*(21))
    p3=p4*(p1+p2;
close
```





Power-flow debugging

PMAC script language allows sophisticated breakpoints for debugging program logic and Power-flow visualization for debugging machine problems.

```
OPEN PLC 3 CLEAR
If(M117=1AND P100=1)
  P102=M103
End If

If(M366<10000)
  I130=P730
  I230=P731
  I330=P732
Else
  I130=P830
  I230=P831
  I330=P832
End If

DISABLE PLC4
P500=0
P500 = 341
While(P500<1000)
  P500=P500
End While

COMMAND "&l#1kzj/&2#2kzj/&3#3kzj/"
COMMAND "&l#1r&2b2r&3b3r"
DISABLE PLC1
Close
```





Setup & Diagnostics

Combined function of setup and diagnostic screens to allow immediate testing of the software configuration.

The screenshot displays a software interface with the following components:

- Table:** A table with 5 columns: Number, Type, pEnc, pEnc1, and MaxDelt. It contains 4 rows of data.
- Setup Details for entry 3:** A panel with input fields for Raw data from (ADC51E Ch 1), A to D offset (0.0), A to D bias (0.0), and Structure Name (Gate3).
- Diagnostics:** A panel with input fields for Raw Whole Counts (12356), ADC A value (bits) (123), ADC B value (bits) (3245), and Counts between last two index pulses (512).
- Lissajou Plot:** A graph titled "Lissajou" showing two intersecting red lines forming a figure-eight shape within a red oval. The y-axis ranges from -11300.00 to 14700.00.

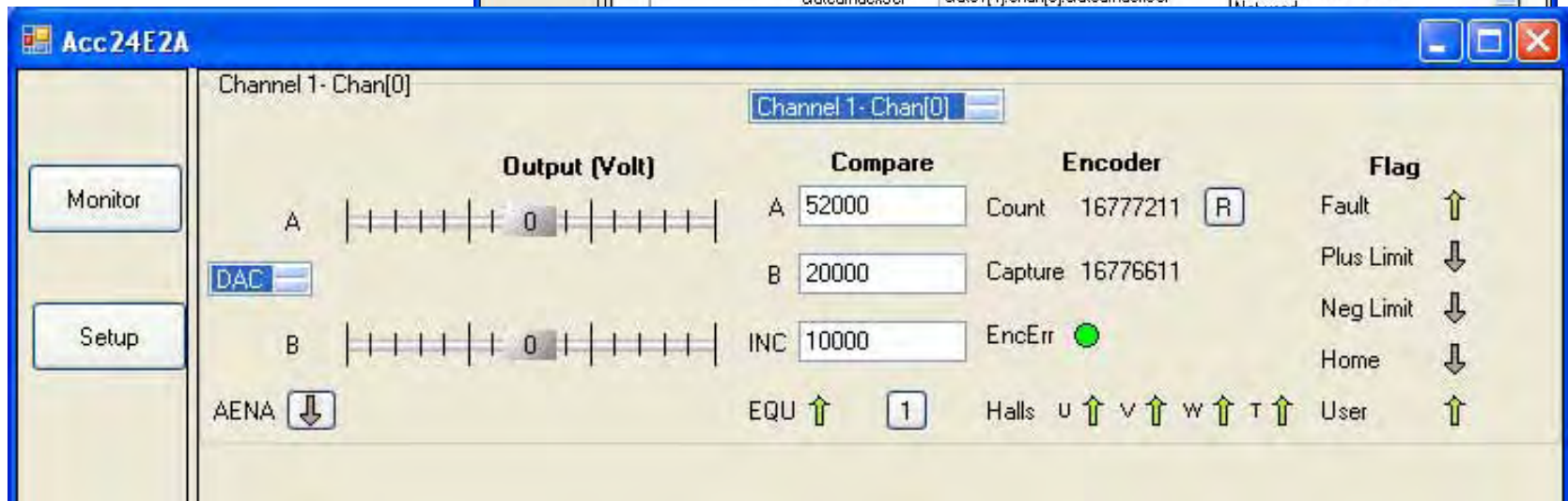
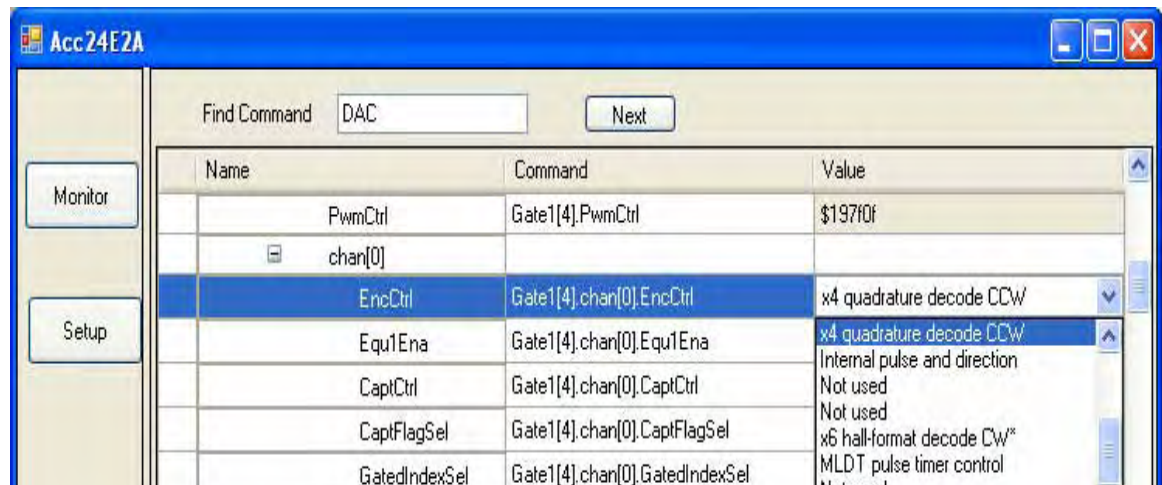
Number	Type	pEnc	pEnc1	MaxDelt
1	1	Gate1[4].Chan[0].PhaseCapt.a	Sys.pushm	0
2	1	Gate1[4].Chan[1].PhaseCapt.a	Sys.pushm	0
3	1	Gate1[4].Chan[2].PhaseCapt.a	Sys.pushm	0
4	1	Gate1[4].Chan[3].PhaseCapt.a	Sys.pushm	0





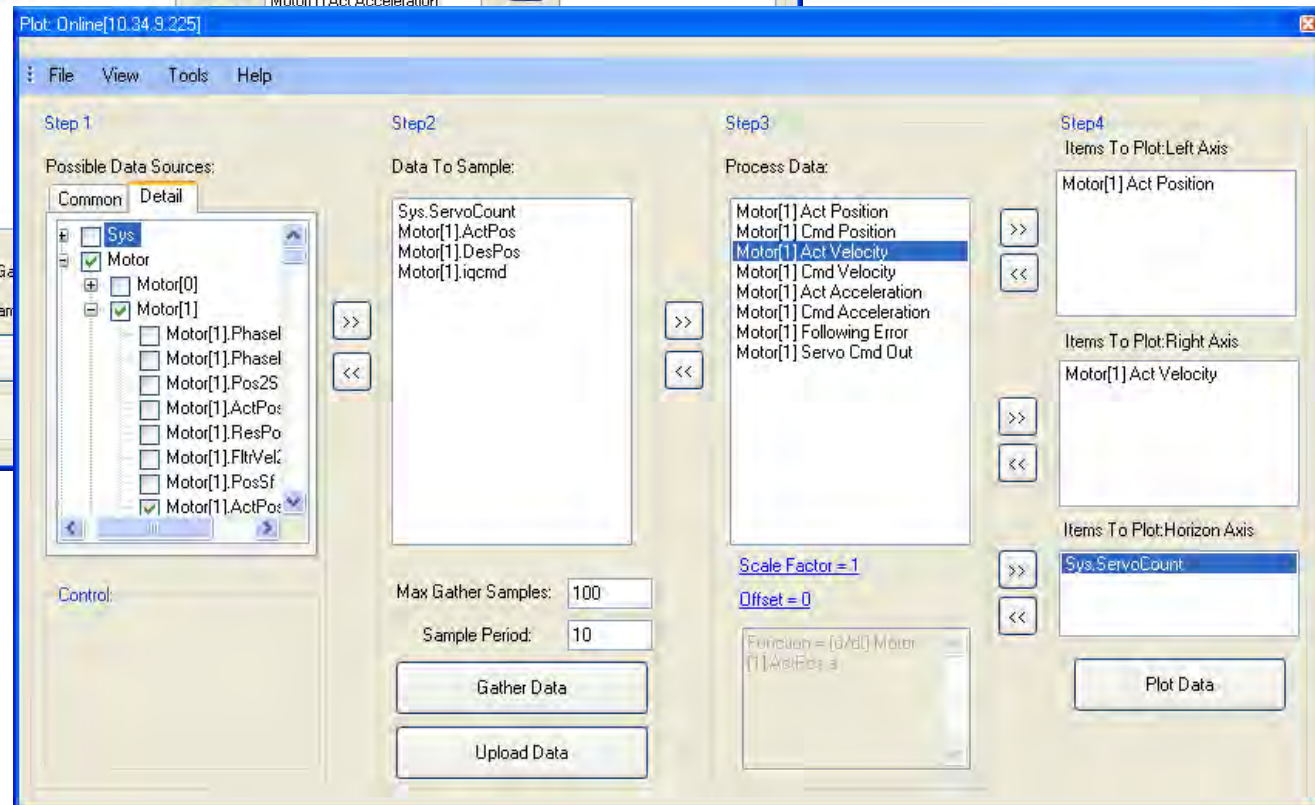
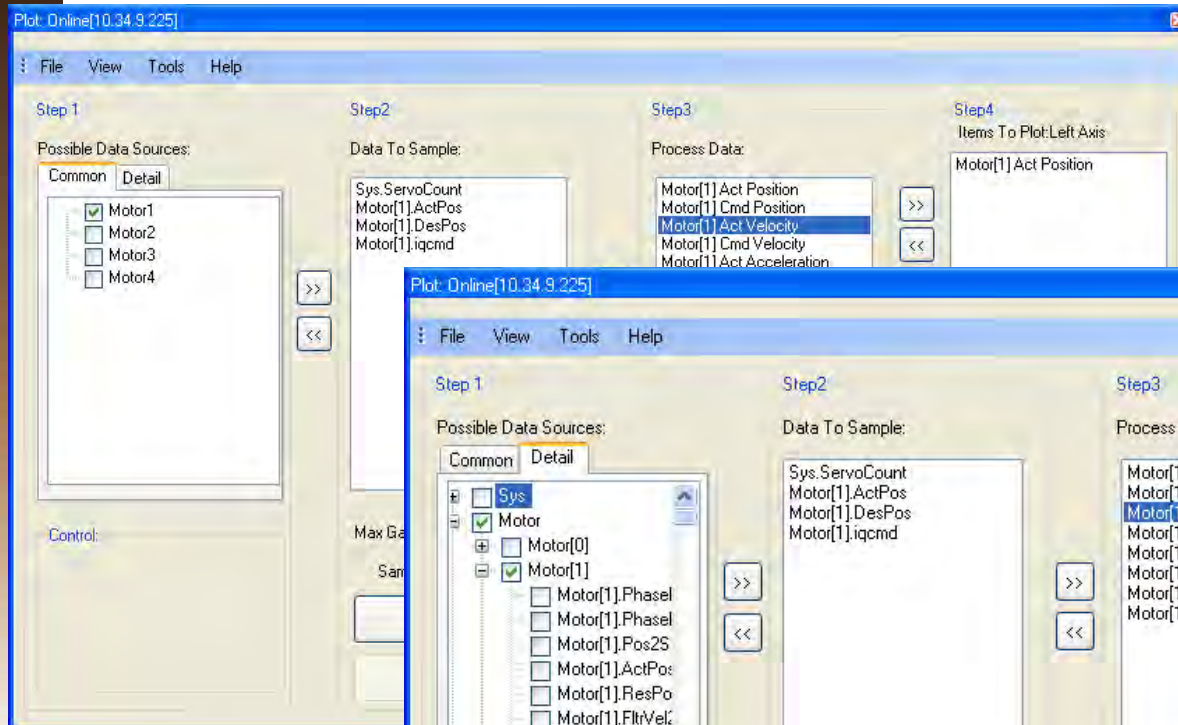
Hardware Testing

Each hardware interface has a test page which does not rely on software in Power PMAC to function.



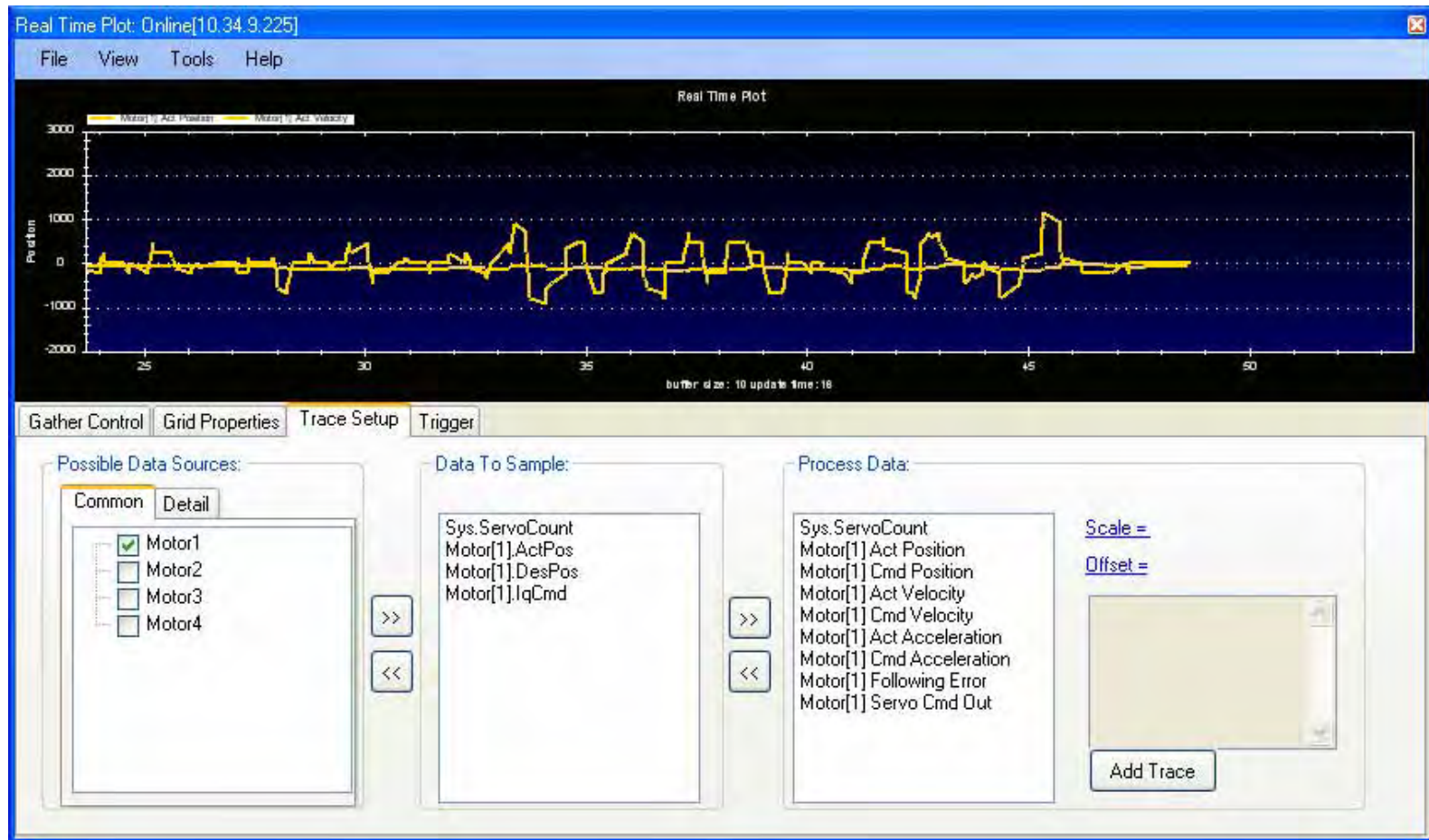


Static Plotting



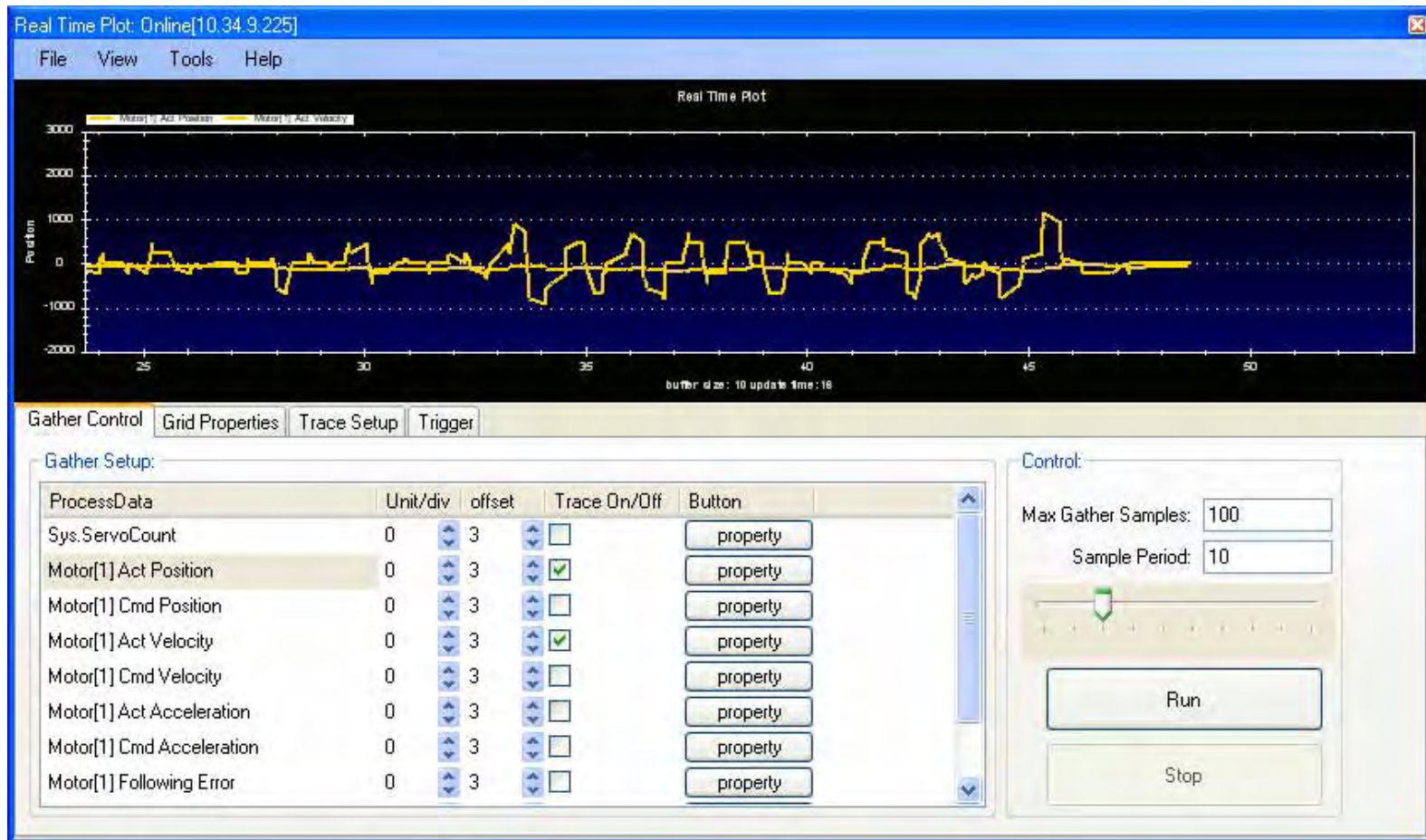


Streaming And Static Plotting



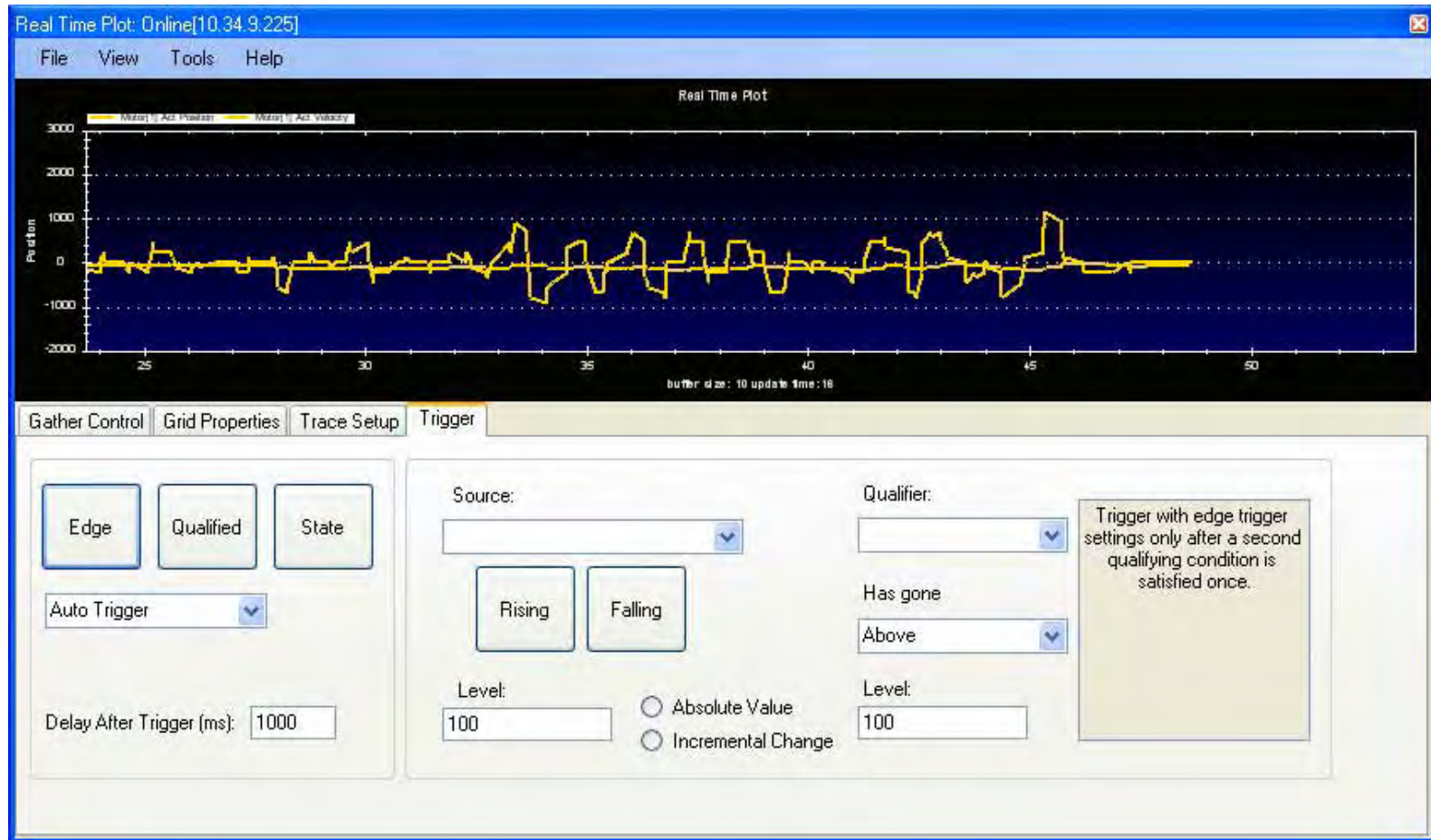


Streaming And Static Plotting



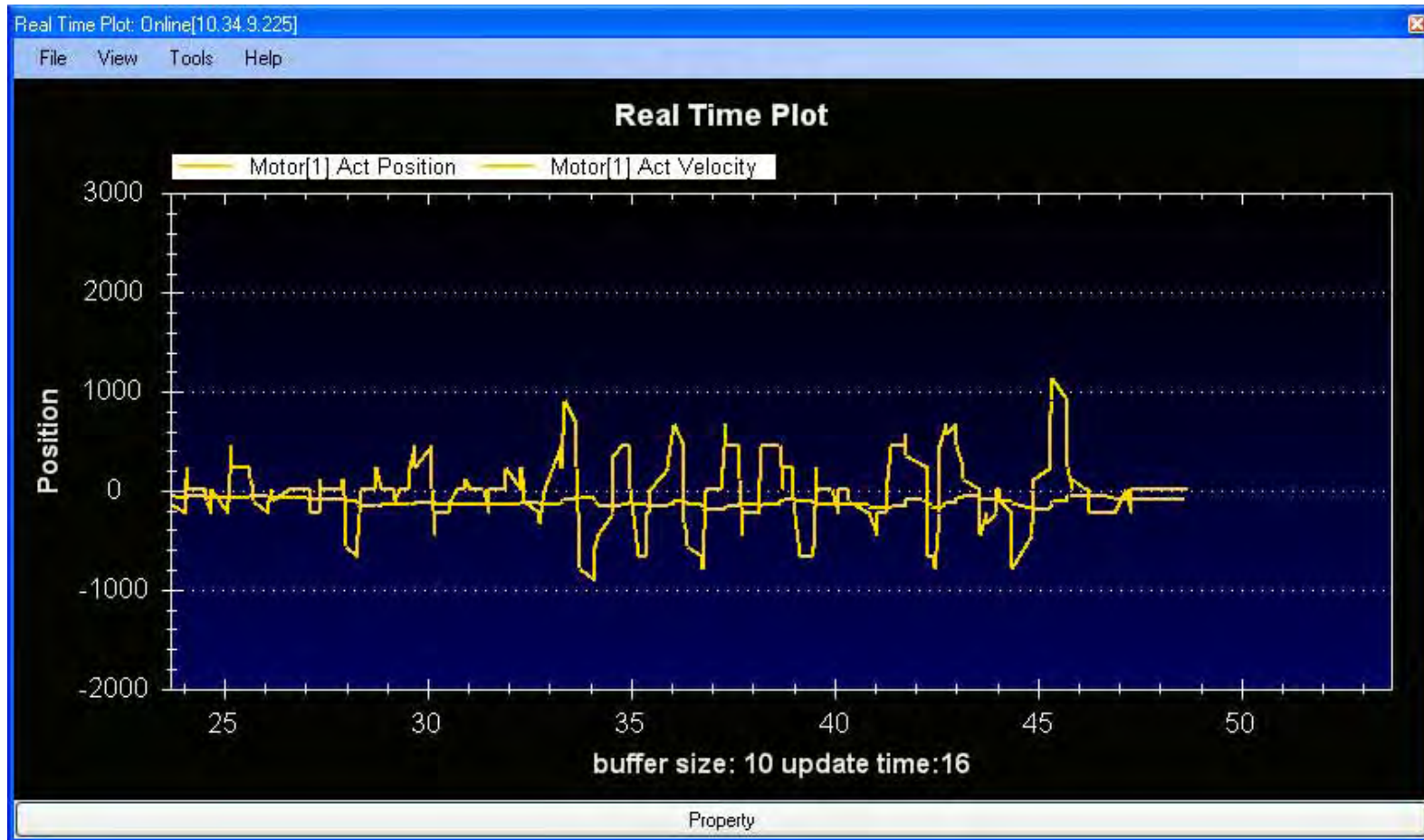


Streaming And Static Plotting





Streaming And Static Plotting





Power PMAC Task Manager

TaskManager: Online[10.34.9.235]

PMAC CPU Information | **PMAC Tasks** | PMAC PLCs | PMAC Programs | OS Resources

Foreground Tasks Overview

Foreground Tasks	Frequency	Calculation Time	Peak Load
Phase Interrupt	8.895 Khz	1.200 usec	16.744 usec
Servo Interrupt	2.256 Khz	25.052 usec	48.021 usec
Real Time Interrupt	2.256 Khz	50.558 usec	82.063 usec

Buffer Overview

Buffer	Memory Used

Details

- * No motor commutation enabled.
- * No motor digital current loop active.
- * A/D-converter demultiplexing algorithm NOT enabled.
- * Phase divider active.

TaskManager: Online[10.34.9.235]

PMAC CPU Information | **PMAC Tasks** | PMAC PLCs | PMAC Programs | OS Resources

CPU Information

Power PMAC Type	PWR PMAC UMAC	CPU Frequency	533.020 MHz
Firmware Version	0.900000	Firmware Date	Nov 20 2008
Total Memory	250 MB	Free Memory	150 MB

PMAC Memory Overview

Buffer	Total Memory	Used Memory
Program Buffer	0 MB	0 Bytes
Table Buffer	0 MB	0 MB
User Buffer	0 MB	0 MB

PMAC Processes Overview

Title	No. Of Processes
PMAC Services	2
PMAC Processes	5
OS Resources	34





Error Window

The screenshot shows a Microsoft Visual Studio environment. The main window displays the Delta Tau Data Systems, Inc. website. The website features the Delta Tau logo and the slogan "NEW IDEAS IN MOTION". Below the logo, it states "The Leader in high performance, precision and flexible machine control." and "New Ideas In Motion Control...". Navigation links for "CORPORATE", "PRODUCTS", "SALES", and "SUPPORT" are visible. A "BRAINS WITH BRAWN" logo is also present.

The bottom of the browser window shows a terminal window with the following error logs:

```
Error: Online[10.24.9.245]
Select device to start communication
Communication to PowerPMAC successful
[5/12/2008 4:17:47 PM] Motor 1: DAC limit: True
[5/12/2008 4:17:50 PM] Motor 1: DAC limit: False
[5/12/2008 4:21:55 PM] Motor 4: DAC limit: True
[5/12/2008 4:21:55 PM] Motor 4: Fatal following error: True
```

On the right side of the Visual Studio interface, there are several panels. The "Position Online" panel shows the following data:

#1	0.00 Cts
#2	0.00 Cts
#3	0.00 Cts
#4	0.00 Cts

The "Velocity Online[10.24.9.245]" panel shows the following data:

#1	0.00 Cts/msec
#2	0.00 Cts/msec
#3	0.00 Cts/msec
#4	0.00 Cts/msec

The "Following Error Online[10.24.9.245]" panel shows the following data:

#1	0.42 Cts
#2	0.00 Cts
#3	0.00 Cts
#4	0.00 Cts

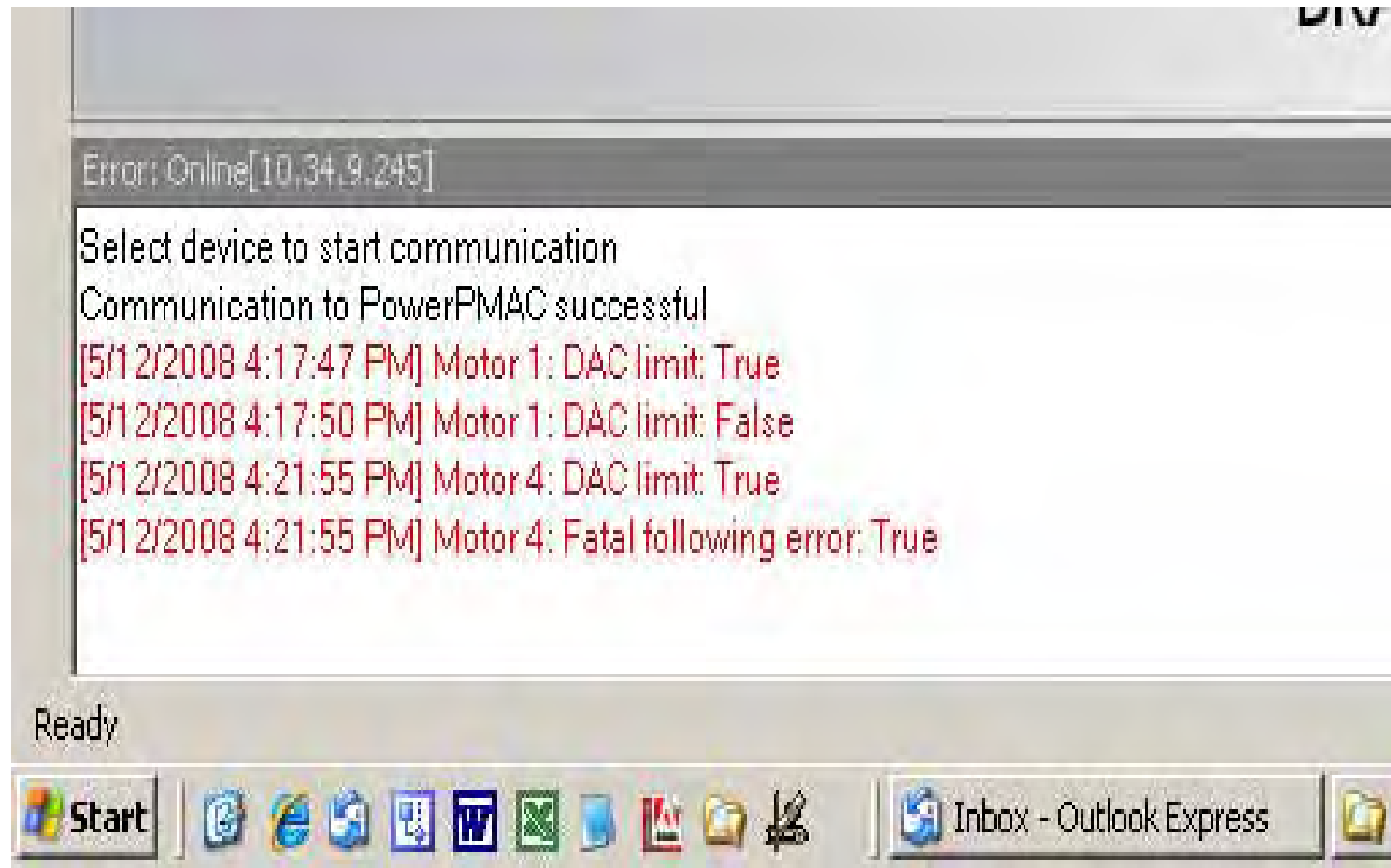
The "Terminal: Online[10.24.9.245]" panel shows the following text:

```
Welcome to PowerPMAC Terminal
Select device to start communication
Communication to PowerPMAC
successful
[5/12/2008 4:21:55 PM] Motor 4: Fatal following error: True
```





Error Window Zoom



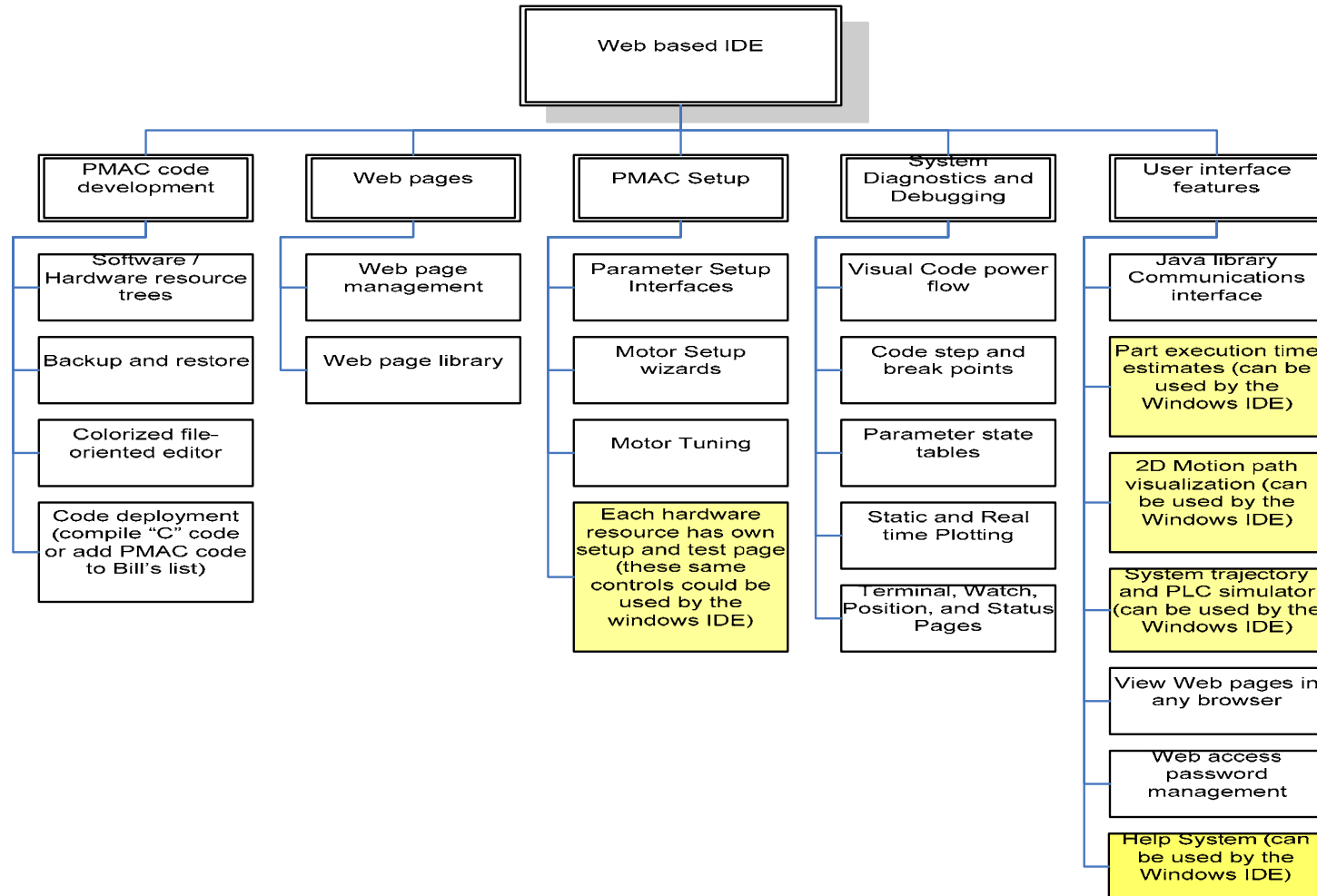


First Release IDE (May 2009)

- New Project (project templates)
- New File (Item templates)
- YAC and Lex Parser (Intelisense in editor)
- PPMAC Database (Intelisense for controls)
- Upload/Download Program
- User-Written Servo/Phase Setup
- Open/Close/Save/Save As Files/Project
- Recent Files/ Projects
- Activate Project
- Exclude File From Download
- Communication Setup and Status
- PPMAC Database Viewer
- Terminal & Watch with intellisense
- Motor, CS, Global status
- Hardware Library Explorer
- Basic Static and RT Plotting
- Basic Tuning
- Encoder Setup
- Backup¥Restore
- Program/PLC/CPLC Status and View
- CPLC compile, load, run
- CPU Resource Setup/Status
- I/O, Servo, Feedback ACC Board Setup/Status
- IDE Installation and Operation in MS Vista & XP



Web IDE Overview

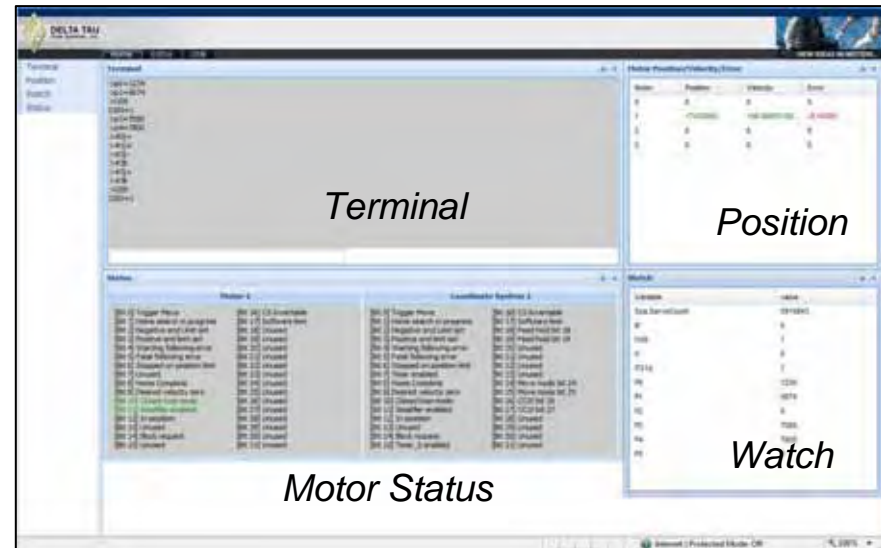


Power PMAC Embedded Web Server

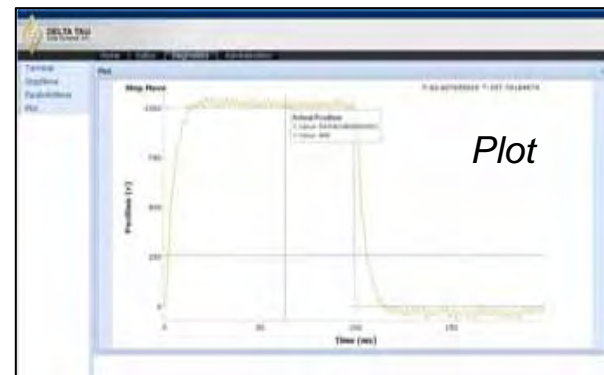


- Web Server program is written in **C#.asp.net** and is **platform independent**
- Every Power PMAC has a dedicated **IP Address**
- Any computer with a **web browser** can communicate with Power PMAC
 - Laptop with direct connection
 - Over company network*
 - **From anywhere in the world over the Internet***
- **HTTP** and FTP supported
- Many standard windows are built in:
 - Terminal, position, watch
 - Backup and restore
 - **Tuning and plotting**
 - Status reporting
- **Remote upload/download** of firmware, programs and parameters
- **User can add his own web application for easy MMI/HMI implementation**
- PMAC Library is accessible using webservice

* Must arrange to get through any firewalls



Standard Web Server Page



Web Server Plot Screen





Webserver Program Editor

The screenshot displays the Delta Tau Webserver Program Editor interface. The title bar shows 'DELTA TAU Data Systems, Inc.' and 'NEW IDEAS IN MOTION'. The menu bar includes 'Home', 'Editor', and 'URLs'. The Control Panel on the left shows a file tree with 'PPMAC Web', 'Test.txt', 'New Folder', 'New Folder - Copy', and 'Test - Copy.txt'. The main editor window shows the following C# code:

```
1 partial class JsonServer
2 {
3     protected void Page_Load(object sender, EventArgs e)
4     {
5         Console.WriteLine(DateTime.Now.ToString("dd:mm:ss:fff"));
6         if (Request[action] != null && Request[action].Trim() != "")
7         {
8             String response = null;
9             bool invalid = false;
10
11             switch (Request[action])
12             {
13                 case term:
14                     response = processTerminal(Request[cmd]);
15                     break;
16                 case pvf:
17                     String pvfCmd = Request[cmd];
18                     if (pvfCmd == null || !.Equals(pvfCmd))
19                         response = processPVF();
20                     else if (ml.Equals(pvfCmd))
21                         response = getMotorList();
22                     break;
23                 case watch:
24                     response = processWatch();
25                     break;
26                 case status:
27                     response = processStatus();
28                     break;
29                 case all:
30                     response = processAll();
31                     break;
32                 default:
33                     invalid = true;
34                     break;
35             }
36
37             StringBuilder sb = new StringBuilder();
38             if (invalid)
39             {
40                 sb.AppendLine(successFailure);
41             }
42         }
43     }
44 }
```

The status bar at the bottom shows 'Position: Ln 1, Ch 1' and 'Total: Ln 66, Ch 1663'. The system tray includes 'Internet | Protected Mode: Off' and a zoom level of '100%'.





Webserver Step Interface

The screenshot shows the Delta Tau webserver interface. At the top left is the Delta Tau logo and the text "DELTA TAU Data Systems, Inc.". A navigation bar contains "Home", "Editor", "Diagnostics", and "Administration". On the right side of the header, there is a small image and the slogan "NEW IDEAS IN MOTION".

The main content area is divided into a left sidebar and a central panel. The sidebar lists menu items: "Terminal", "StepMove", "ParabolicMove", and "Plot". The central panel displays a "Step Move Setup" dialog box with the following fields and options:

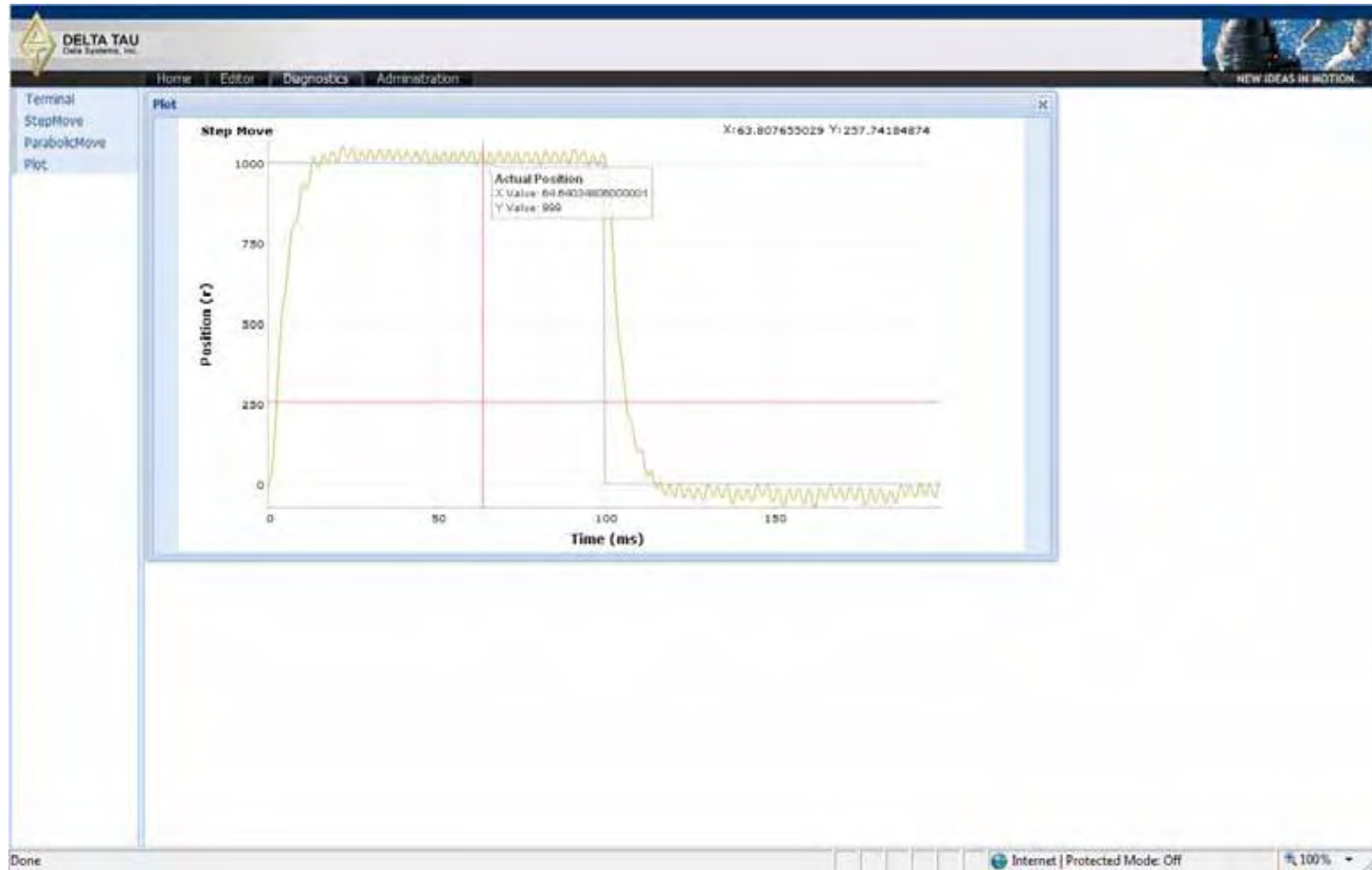
- Motor: 1
- Move Size: 1000
- Move Time: 100
- Kill:
- Single:

At the bottom of the dialog box are "Submit" and "Cancel" buttons. The browser's status bar at the bottom indicates "Internet | Protected Mode: Off" and a zoom level of "100%".





Webserver Step Response





Webserver Change IP

