DepEtchPro

The Control System

for Thin-Film Processing Systems

That Needs No Programming

Electrogrip
6945 Lynn Way
Pittsburgh PA 15208 USA
www.electrogrip.com
412-247-0992

How does DepEtchPro eliminate programming?

Configure your system and recipes ...
Run jobs ...
Form graphs of system performance ...

All without ANY software support!

- **Optimized for the needs of the thin film process community**

- **Screen 'Items'** represent tool parts such as generators, valves, gauges

- **Graphical placement of screen Items puts the system together**

- **Item Name and Color control ALL system interactions**

- All Item properties are simple natural-language text

- Comms protocols can be selected from menu lists or manually entered

- Items are user-changeable as well as being alterable by Job Files

- Job Files can also pass new parameters to other JobFiles

- *Result #1:* Future-proof both New and Legacy systems

- *Result #2:* Secure, safe, interlocked operation with password protection

- *Result #3:* Wide range of I/O devices supported

- *Result #4:* Distributed control for all-local, simple and accessible wiring

- *Result #5:* Systems can be built up piecemeal, automated from the start

- *Result #6:* Automate a simple vacuum system in less than 1/2 day

... Why Wait to Automate?
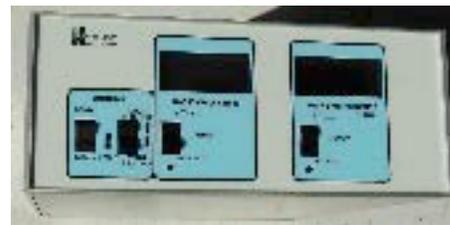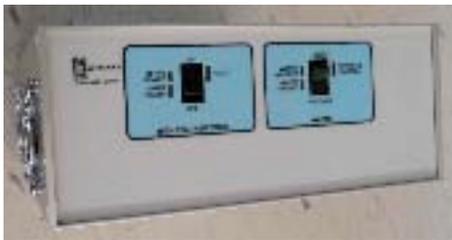
What devices does DepEtchPro support?

· PLCs, Power converter inverters, etc; Modbus RS-485 binary/ASCII devices

… PLC's are used here for input/output of digital and analog signals, not for their control capabilities. These are an expandable, economic choice. Standard Ethernet cables carry both power and signals.

- **USB-serial devices …**

- **RS-232 devices …**

- **DeviceNet devices …**
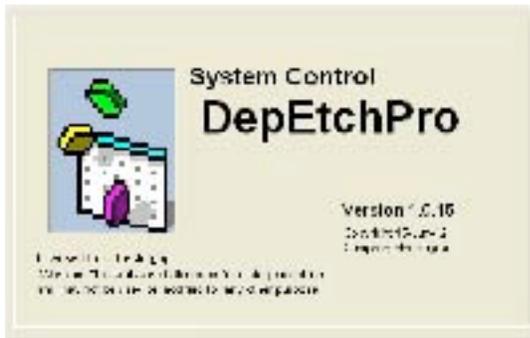

Now Test-Drive DepEtchPro …

Tour and Quick-Start Guide
Initial Startup

Install DepEtchPro's "DEP SETUP" folder on your hard drive / desktop, and put contents of the folder "WindowsUtilities\Put Into System32" into your Windows System folder "System32", if those files are not already there.

Double-click on the "DepEtchPro.exe" file to start … the following screens will show:

Your Login and later screens will show **OnScreen Help** if you log in as 'guest' or 'demo', which describes what each element of the screen does as your mouse moves over it. It is toggled on and off from the Help menu.

With a 'guest' or 'demo' login, you have full permission to act at the highest user level, as chosen in the 'Operator Level' menu. However no saving of files nor signin of other users is permitted.

After login, your screen will expand to show a blank field, and you will pull down to the 'Open Setup' menu. For this demo you will open the provided "......DemoSetup.txt" setup file:
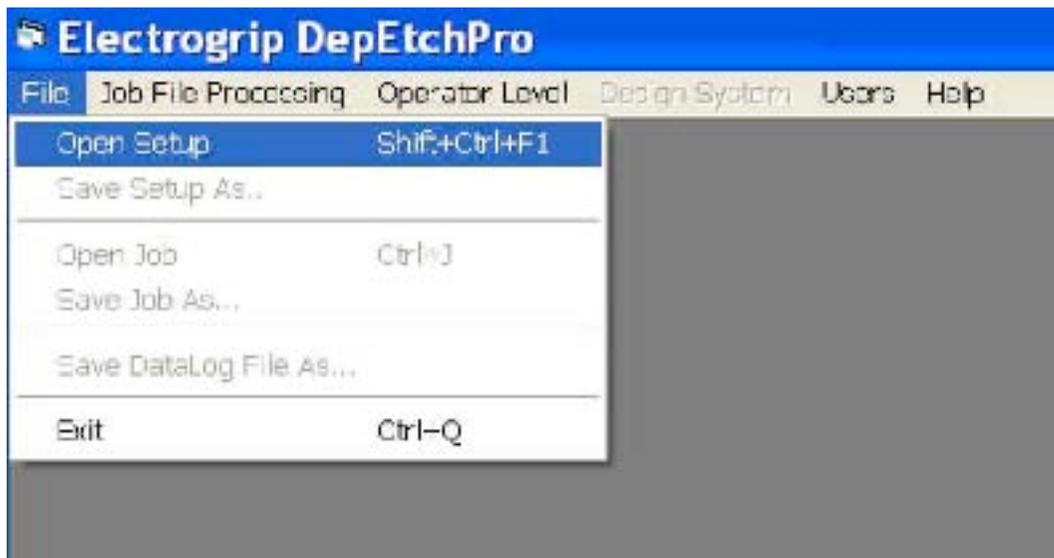

At the bottom of the screen are shown blocks with System Colors; these are normally read from a Preferences file at startup (but not in 'demo' mode) and in that way can be tailored to the needs of colorblind users. This demo set is compatible with the most common male colorblind condition.

You will be prompted to decide where to store datalog files. This should be in a dedicated folder since these would be generated in quantity, over time. With a high user privilege level, storage locations of datalogs can be changed at any time with the File menu:
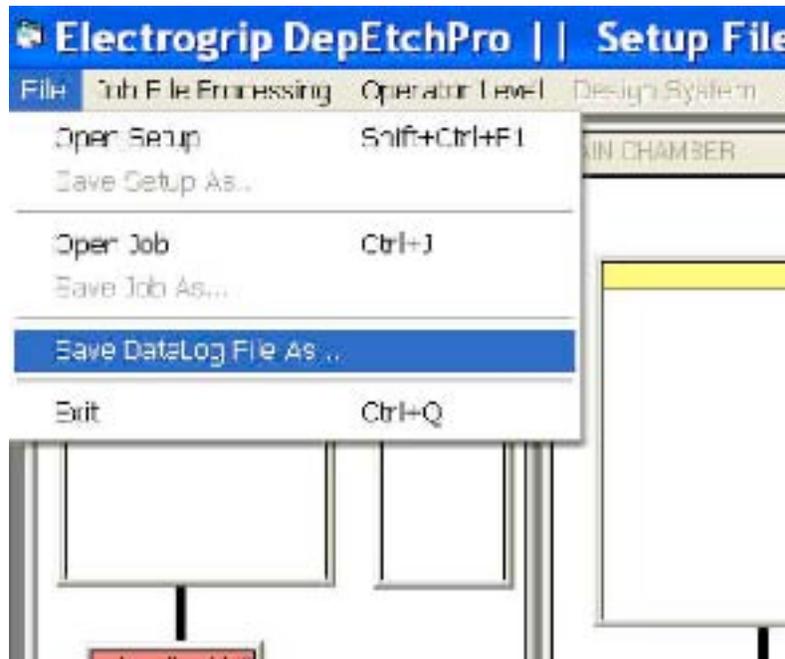
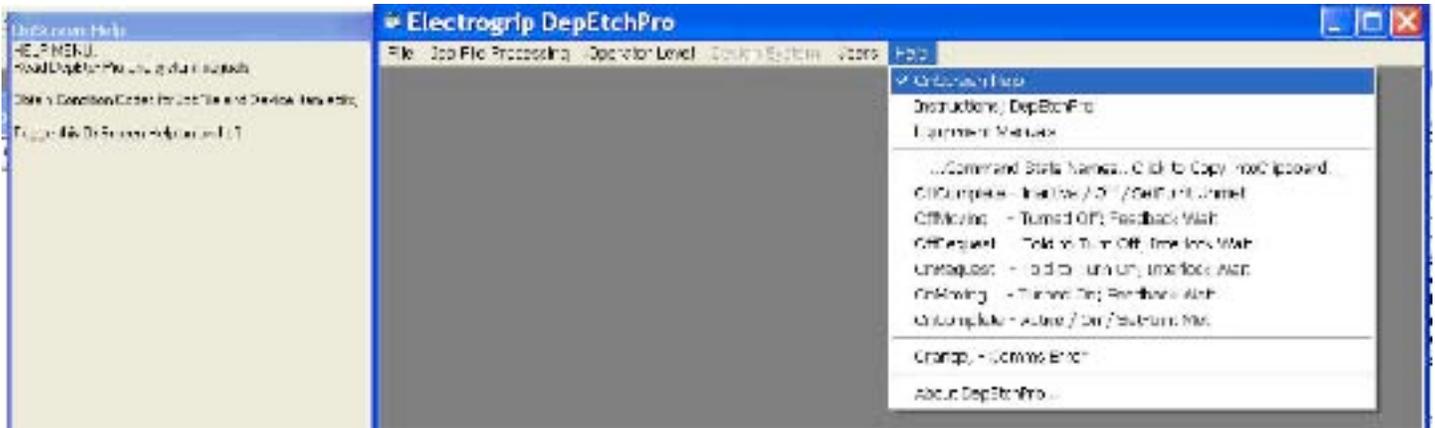
Open the "JobLight&IonGraphing.... .txt" file.
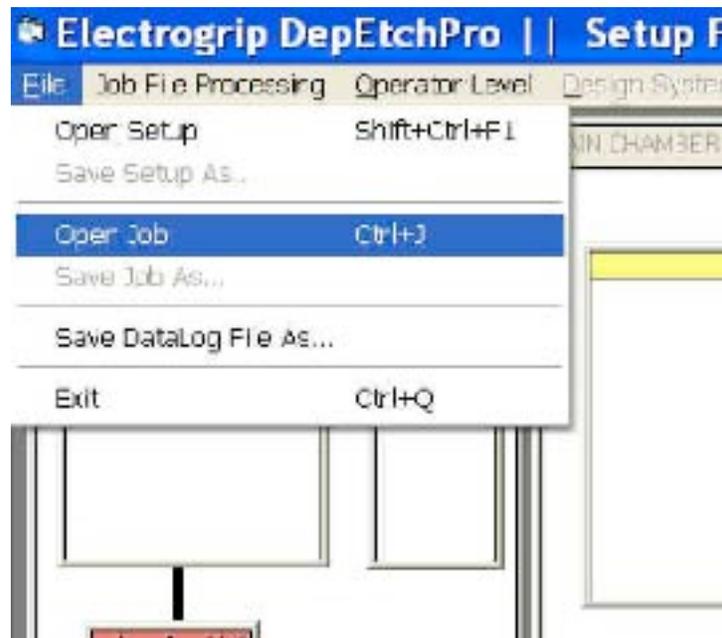
Note that all Job Files, both in the "Core Jobs" and the "Job Processing" window, have a bottom yellow region full of text. These Job Files can be stretched with a mouse to see inside and, with a sufficiently high user permission level, be edited.

The **OnScreen Help** window also lets you see 'inside' a Job File if you mouse over its top bar.

Screen Overview

The **'Parent' windows** holding smaller icons represent major system blocks. They don't take commands like their 'Child' icons, but help in system organization. They can be moved and resized, placed behind other Parents if desired, and their 'Child' forms move and scale accordingly.

The **lines between Child icons** are also a visual aid to organization. No commands are derived from, or given to, these lines. They can be drawn with various line widths, from the 'Design System:Draw Lines' menu.

The bottom right-hand **'Core Jobs'** are routines loaded with the Setup file. As with the Setup file, the highest level of user privilege is required to make changes. These routines would be basic machine operations such as venting, gas backfilling, pumpout, etc. They are called by the user-level job routines as required. Parameters can be passed between routines to attain the pressure, power, flow, and timing requirements specific to each run.

In addition there are **'monitor' and 'datalog'** Items which can yield a full range of automatic cycling, graphing, and system monitoring, together with alarm and error exit functions. All of these monitor functions may be used as-is, or reset by users as needed for each run, through commands in their user-level job files.

Understanding Item Properties

With OnScreen Help turned on;

Take a look at some Item properties by mousing over their top bars … here is the HiVacValve display;
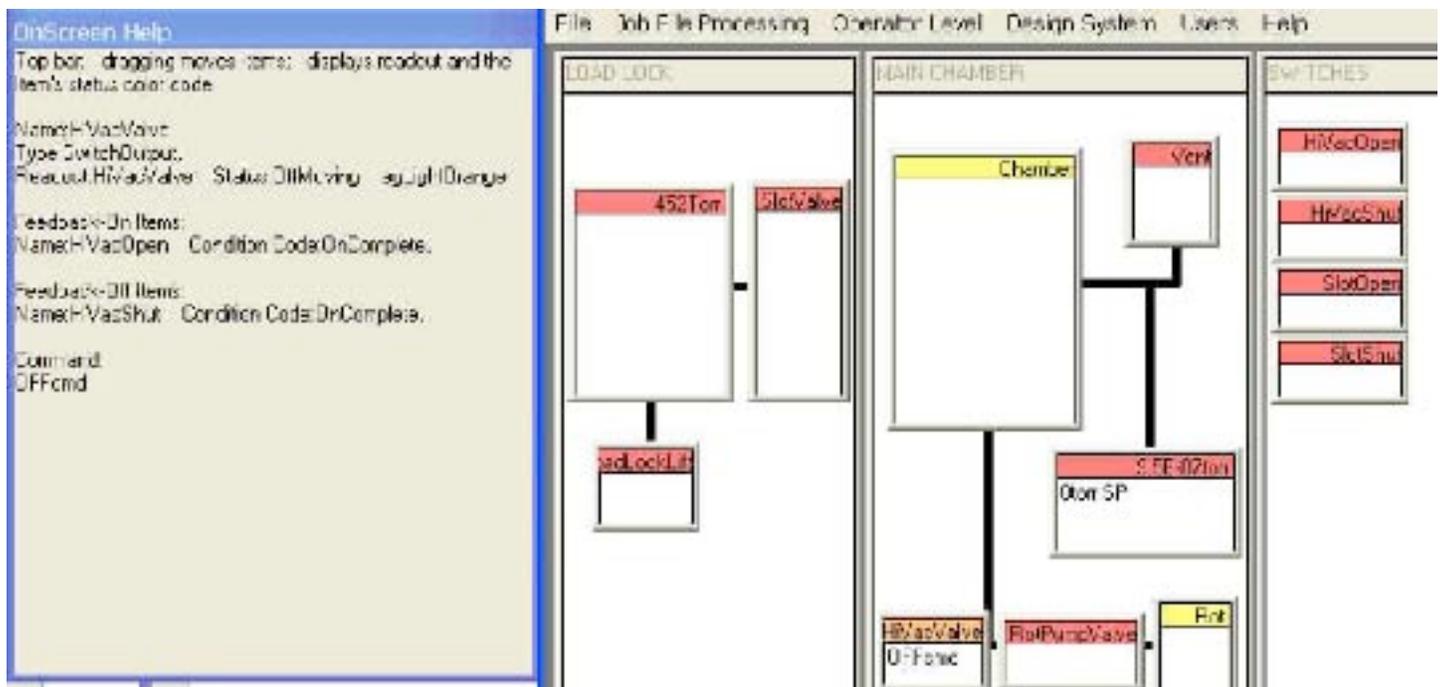
The Help screen shows the HiVacValve Item's current state, internal feedback and interlock conditions (if present), and its current command.

This HiVacValve's state is "OffMoving" because to be fully Off, its feedback criterion states that the HiVacShut switch has to be OnComplete. However HiVacShut's color is light red, signifying OffComplete. You can "Edit" (see later) to take off that Feedback condition and observe the HiVacValve then moving to the desired state.

Mouse over a JobFile and see the lines of job file code inside, listed under "Command". The JobFile commands have an Item Name, followed by some values that will be put into that Item's properties to cause an action. When that action is seen to be complete, the next line is executed.

These JobFile lines are mainly generated automatically using right-clicks on an Item that is to be commanded, selecting the desired command, then pasting that command into the JobFile line.

Running a program

The JobFile that was loaded can turn lights - and their corresponding icon colors - on and off under timer control, if you have the proper hardware attached to your computer's serial port. Without that hardware, here's how it looks as it runs.

Before running a job, you can make sure that any 'Monitored' Items are graphed.

Request to **view monitor timer charts** as follows …
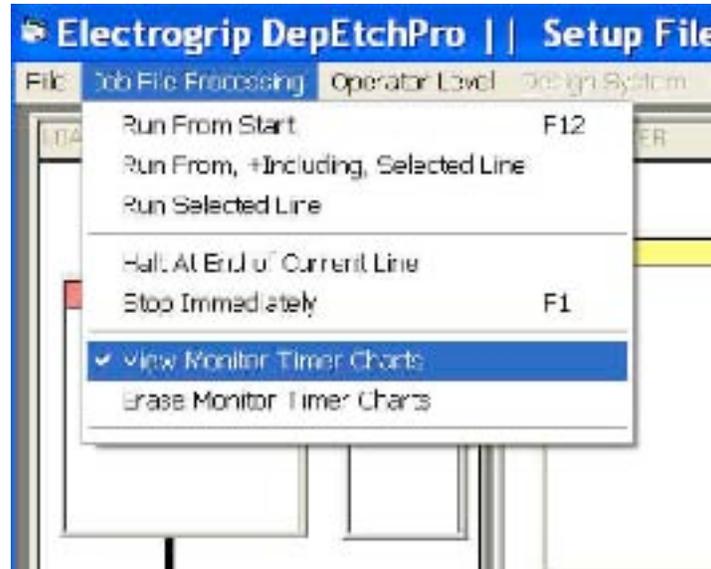
Nothing changes immediately as a result of putting a tick mark against that menu choice. The job file, when it is run, will activate the items that it needs to show graphically; only then will graphs be displayed.

Now if you were to command "Run From Start" the program would run. The file loaded is a "Main" type of job file. Only one "Main" file is permitted in any set of jobs loaded at one time, so there's only one "Start".
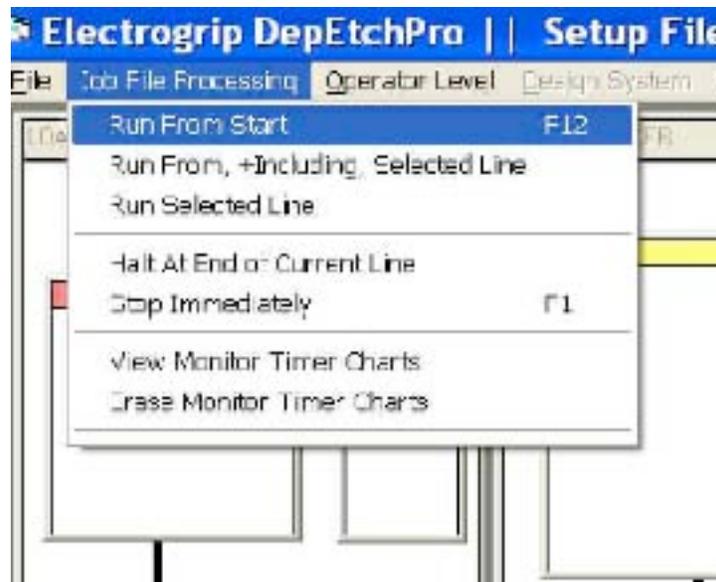
Here's the result of a test run, and you will see that the two analog items being monitored (the LoadLock pressure and the IonGauge pressure) have been driven with a wild waveform during the few seconds of program operation. One is logarithmic and the other linear - and there is no pretense that this is a real vacuum chamber. The graph axis scales are dynamic, adapting to the actual ranges of the observed data in real time. The 'BarLight…" icons graph those lights being turned on and off under program control.
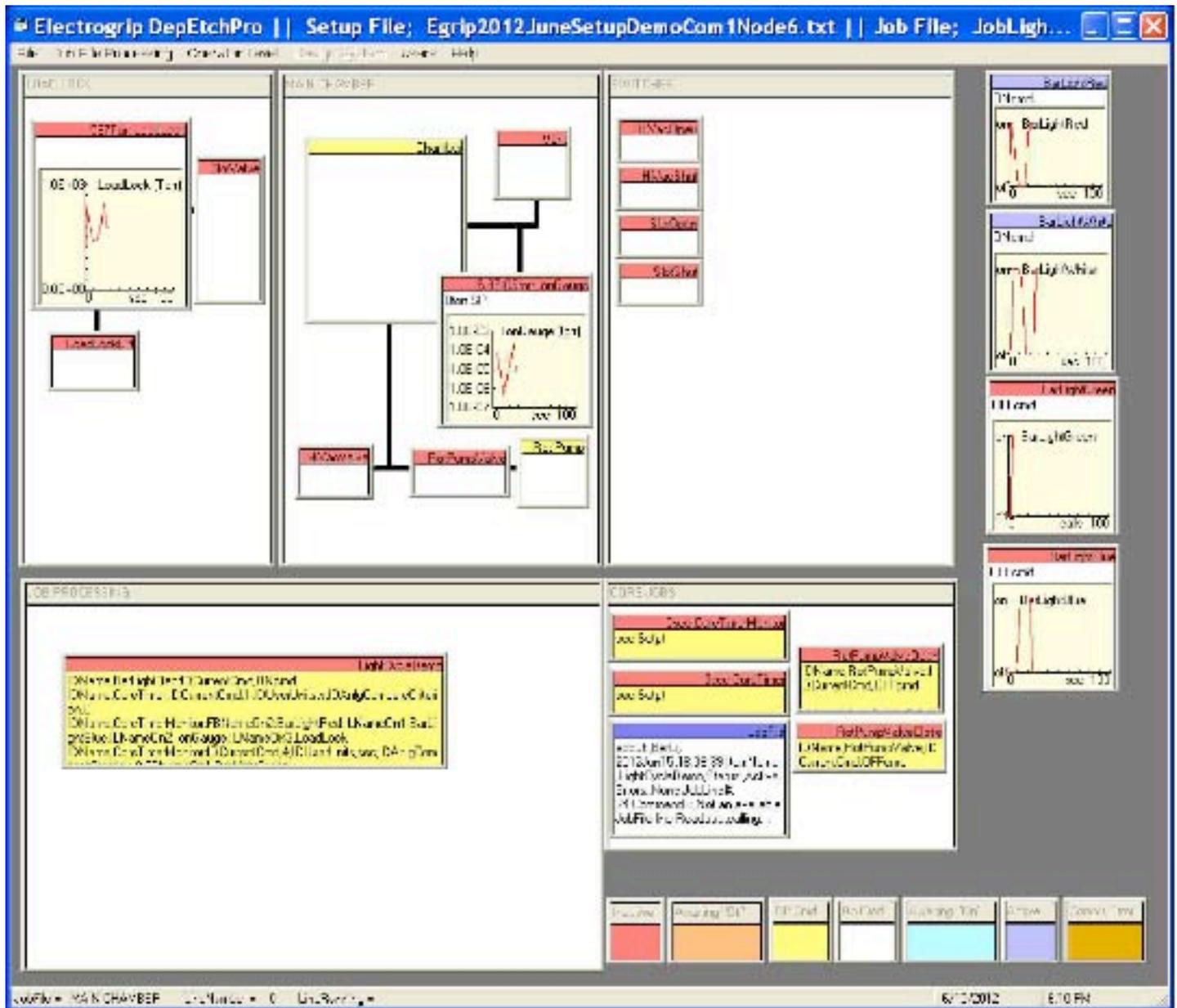
Note that the icons have shifted in size and position! During graphing, while the run is progressing and after the run, these positions and sizes are set by the user to make the screen viewing optimal using a mouse to drag and drop them around. With a high user privilege level, such locations and sizes can be saved so that the Setup file recalls where to put graphs thereafter, when they are called for.

If you now erase the graphs (using the JobFileProcessing menu command) they will remain on screen, but with their axes set to nominal scales and without the read data graph lines. Alternatively without erasure, successive runs overwrite the graphs and differences between runs can be observed in real time.

DataLogging and System Monitoring

DepEtchPro thus permits you to watch your run in real time, using its graphical features. These graphs are initiated by calling a 'Timer Monitor' icon at the start of the run. 'Timer Monitors' can also be used to make cyclic operations, and to exit if run conditions go out of specified bounds, and to generate log files that provide permanent records of the above graphical results.

DepEtchPro permits you to have any number of 'Timer Monitor' icons, and they can all generate their own log files, allowing as **fine level of datalog detail** to be recorded for any run as may be desired.

Here's the datalog of the above run and graphs, as saved by a timer monitor; the first line of the datalog file is the place where it was stored, and you will note that the filename is generated with the date, hour, and minute of the start time, and with the monitor icon's name, to ensure traceability and uniformity in datalog generation.

C:\Documents and Settings\Administrator\Desktop\DEP1015\DEPSetup\Log_2012Jun15_18_8_CoreTimerMonitor.txt

Elapsed Time (secs),BarLightGreen,BarLightWhite,BarLightRed,BarLightBlue,IonGauge (torr),LoadLock (Torr)

```
 1, 0, .1, 1, 0, 0, 797
 5, 0, 1, .7, 0, .000068, 623
 9, 1, 1, 0, 0, .0000056, 462
 13, 1, .9, 0, 1, 3.9E-07, 470
 17, .7, 0, 0, 1, .0000043, 584
 21, 0, 0, 1, 1, .000028, 814
 25, 0, 0, 1, 0, .00021, 600
 27, 0, 1, 1, 0, .0000071, 600
```

This output directly imports into spreadsheet and scientific graphics packages.

System monitoring is also enabled at a deeper level with 'Timer Monitor' icons; they can have monitored icon conditions added to them so that error routines can be called if the main routine conditions are violated, beyond the acceptable tolerance levels set in those icons.

In addition 'Timer Monitor' icons permit **cycling** up to a given number of times, before exiting to another routine for end-stage processing.

The job file that makes the above happen is made up of **plain text instructions**.

All of the above job file detail is set up with the DepEtchPro graphical interface; mouse click copy-and-paste, mouse choice of icons that need to be monitored, and mouse actions to choose setpoint conditions.

These methods are detailed next.

Setup / Job Icon Editing

Did we state: DepEtchPro requires NO PROGRAMMING? Here we see how that really pays off.

Other types of system control require specification of link paths between graphical screen elements and message-passing. Such coding is specific to each system and requires considerable setup and programming time.

DepEtchPro eliminates such coding and setup.

The screen Items in DepEtchPro communicate only with their 'IDNames' and their 'System State' top bar colors. All interactions between icons are thus reduced to recognition of a name and color, which are entered into the Item's property screens.

This enables interlocking, and feedback of process conditions, to be automated in a clear manner which is at all times under user control, and easily altered to accommodate actual system performance or equipment changes.

Interlocks: "Control" actions can be prevented if a "Target" condition is incorrect ...
If a 'control' icon has the name of a 'target' icon in its 'Interlock' text area, with a condition code color specified, then unless that color matches the target icon's color, the 'control' will not activate.

> Example; you may specify that a vent valve may not be turned ON unless a gate valve is shut, ion gauge protection valve is shut, and a rough pump valve is shut.
> You may also specify interlocks for turning OFF the vent valve, if required.

Feedback: "Control" actions are not fully completed until a "Target" condition is met ...
If a 'control' icon has the name of a 'target' icon in its 'Feedback' text area, with a color specified, then the 'control' will activate. However it will not show that its action is complete until the target shows the correct color.

> Example; you may specify that a roughing valve will only be shown to be on if its feedback switches indicate it is fully open, and that it will not be shown to be off until its feedback switches indicate full closure.

The Result?

You still need to draw the icons, give them names, provide the units of measure if it's a generator or gauge with analog input and output, ... and tell it how it is meant to communicate with your computer. [Job files are blank for most of those entries.]

In DepEtchPro those required entries are mainly done by copying and pasting, and with menu pull-down and list choices. Many icons have similar properties (eg, they are on the same communications protocol, and they may be switch inputs) so it's possible to just copy one icon into several versions of itself for even more rapid system generation.

Then you need to enter the interlock and feedback conditions, which take some care and experience with a system to learn what works best. DepEtchPro allows these to be adjusted easily, and then locked at the high 'system design' user level of access, to assure reliable, safe system operation.

Following is a step-by-step demonstration of such system setup and design.

The Edit Form

Request to EditForm/Device, the most powerful Edit mode where any type of Item can be changed and saved.
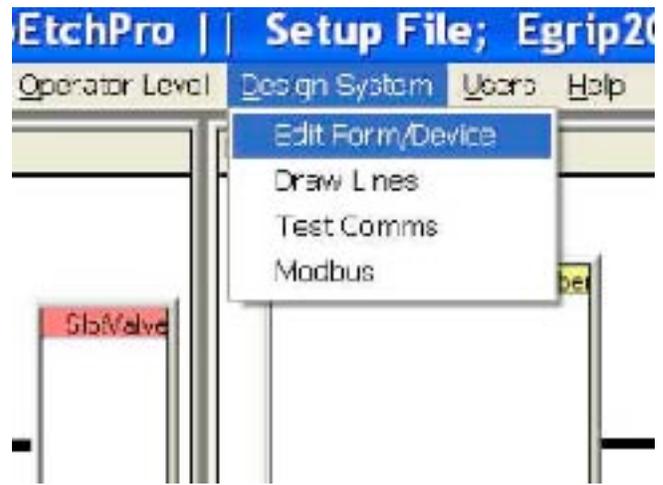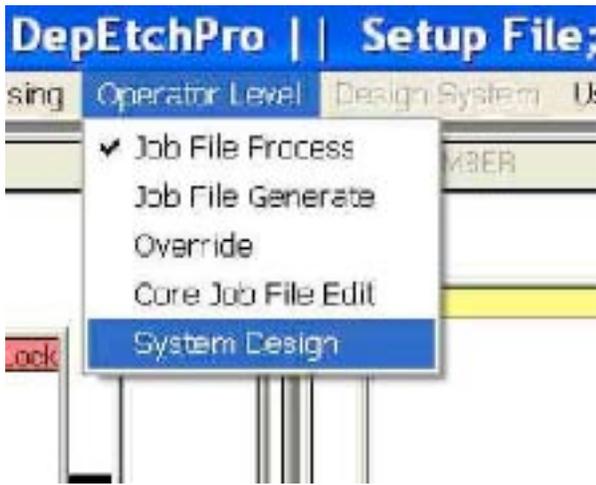

The below Edit screen will show. Right-click on an Item you wish to Edit, selecting the choice "IDName copied...", then right-click in the top text window as shown below, and choose "Paste".
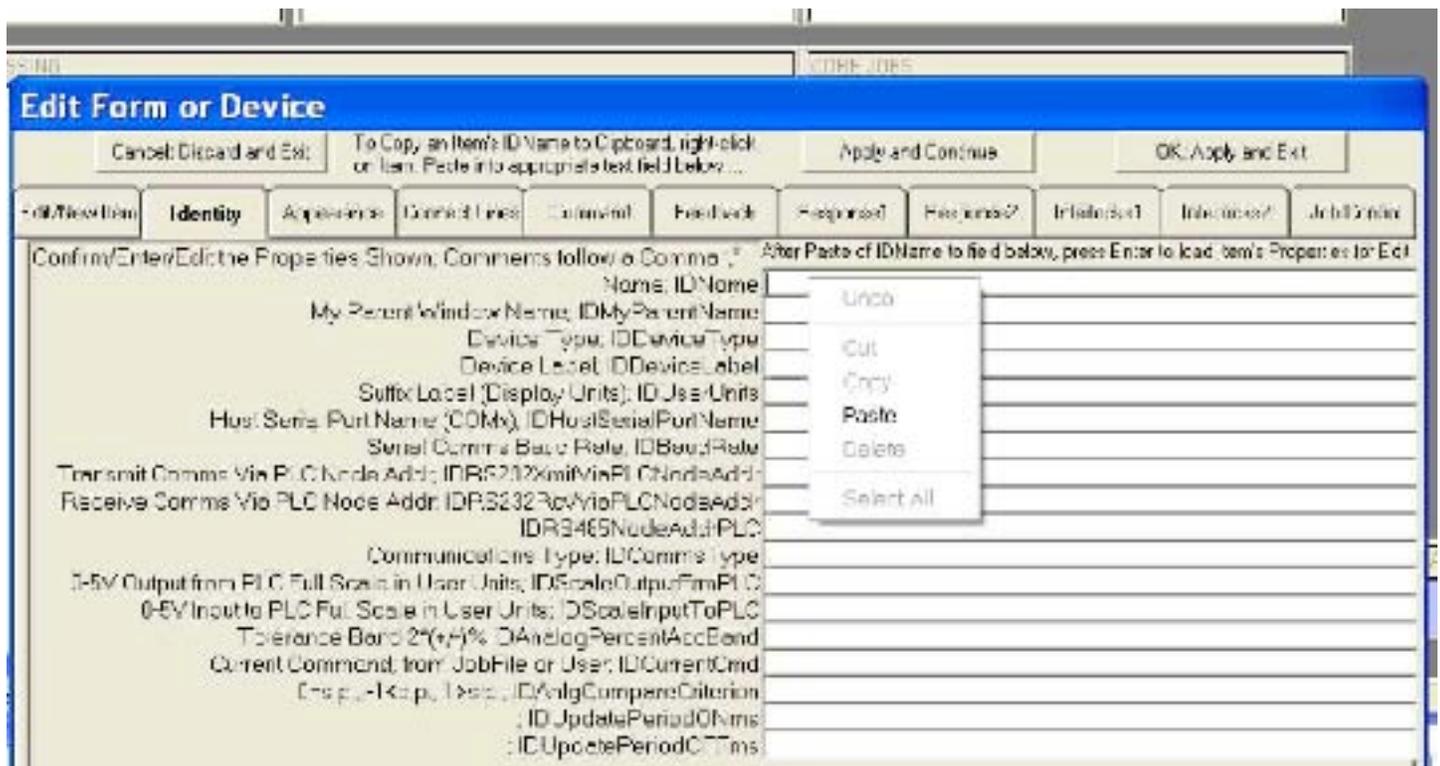
All of this Item's properties will be filled into the Edit screen's windows, including into this 'Identity' tab. Hover your mouse over each text cell and observe the OnScreen Help descriptions.
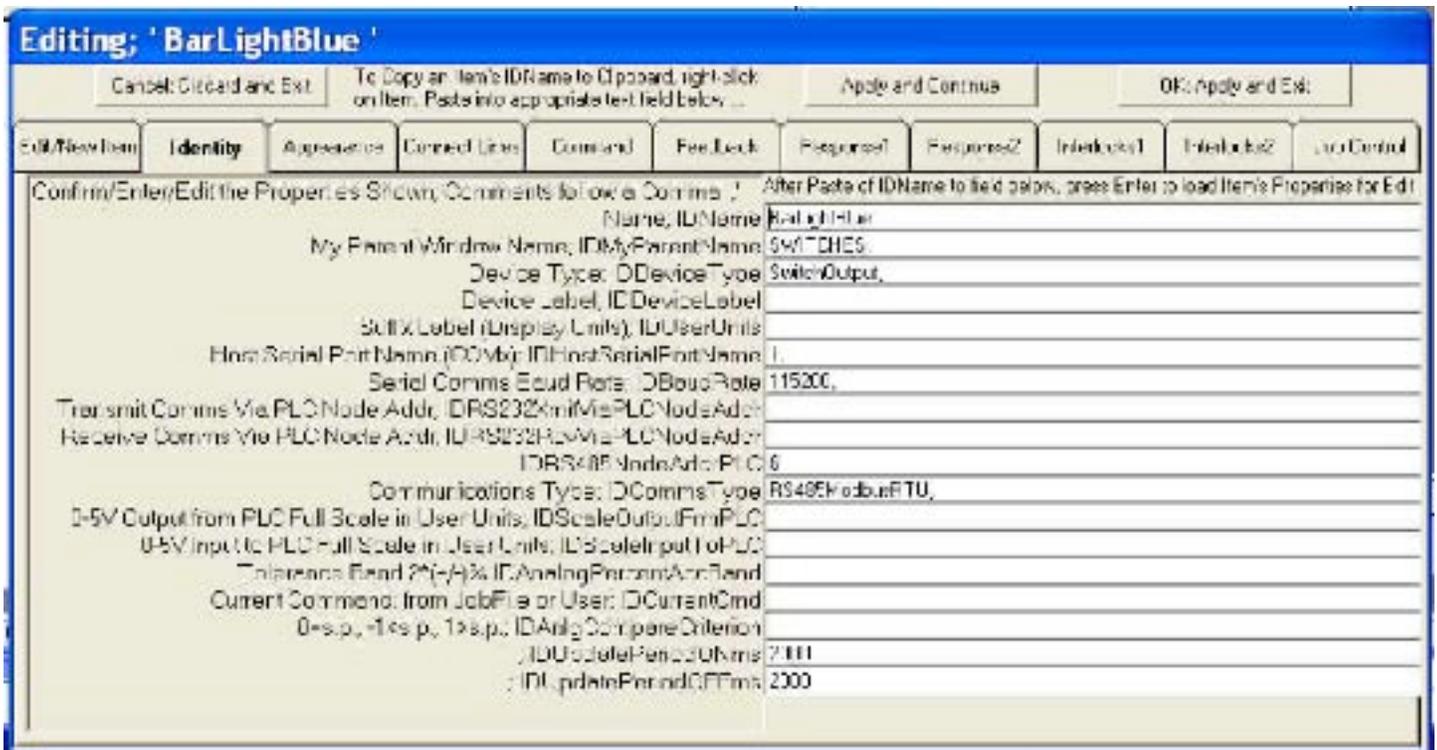
Standard settings to alter the character of this Item are on the Edit/New Item tab. Double-click on the list choices to enter the DeviceType, Parent, Communications Protocol, etc into the Identity tab data. This form also permits you to select "New Item" and enter a new name, using the current Item as a template.
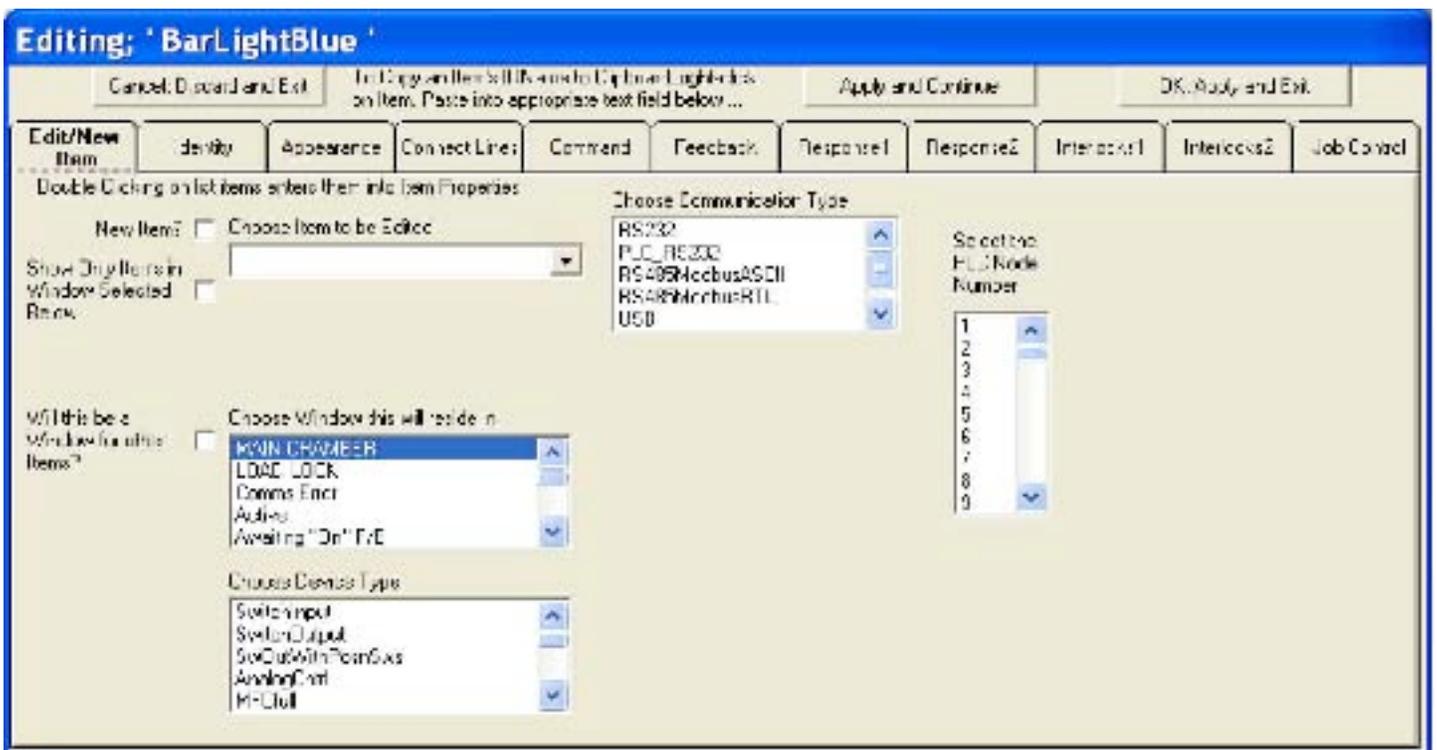
The Appearance Tab is as follows; again the OnScreen Help details each cell's properties.
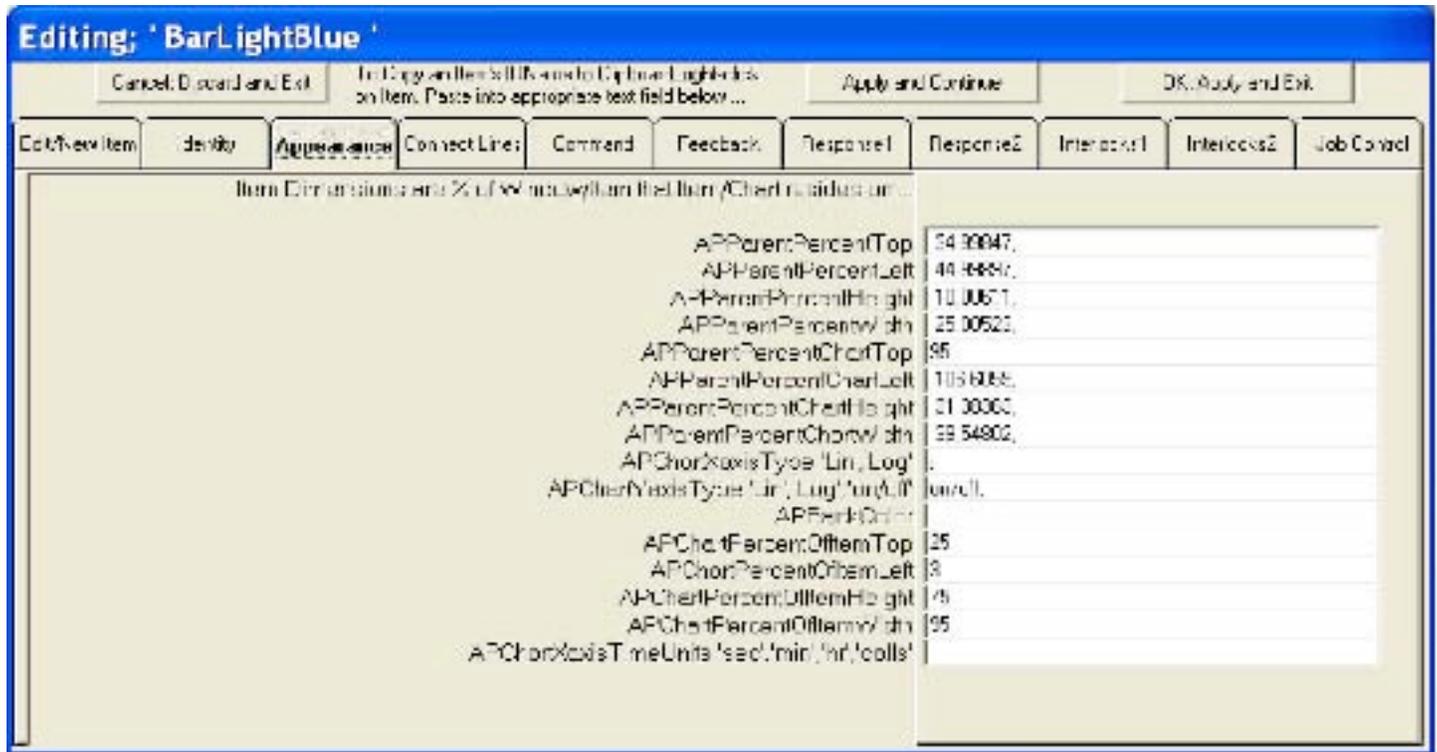
The Command and Response tabs are in sets of 4 properties: (i) Command type; digital or analog, and read or write. (ii) The address it is sent to inside the I/O device (ie, which pin number). (iii) The values being sent out currently, or the size of the data string to be received. (iv) The 'zero' command or analog zero offset.

If more than one command is put here, 'write' commands are executed in sequence to (say) turn on an interlock line, apply input power, put on a filament power switch, then command an output power level from a microwave generator. The responses from sending those commands are checked to ensure that the I/O device accepted the commands, by doing 'read' commands to the appropriate internal addresses.

Mouse over these cells to read more about their options, using "OnScreen Help".

Interlock and Feedback properties are entered in these screens: again, mouse over these cells to read more about their options, using "OnScreen Help".

Job File Generation

The next Edit tab is Job Processing; and this applies to Job Files. Select "Job File Generation" from the pulldown menu and right-click on the 'Main' job file you opened earlier, then take in its IDName for Job File editing. Paste that IDName into the Identity tab's top field and you should find its Job File lines installed in the final tab.
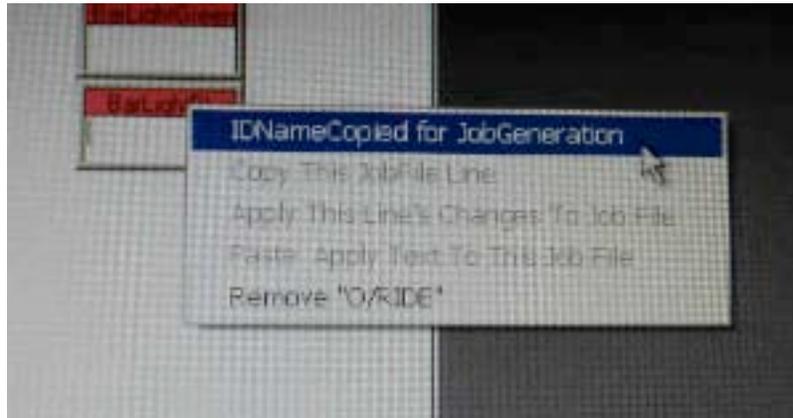
Right-click on (say) one of the right hand Items for the 'Bar Lights'. Choose an action 'command ON' or 'command OFF'. Now right-click on a blank JobFile line in the Edit screen and Paste that line in. You will see that the line has been automatically generated.
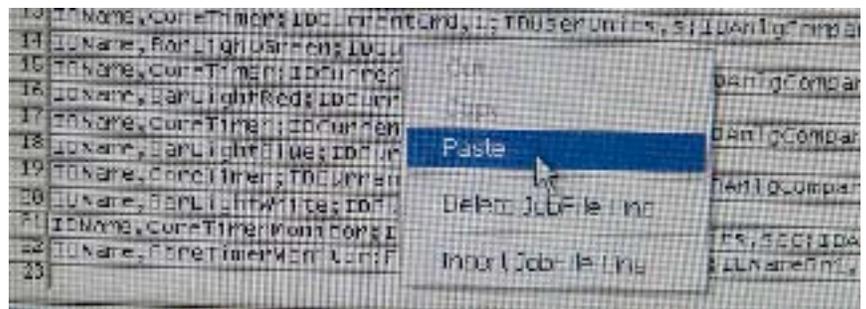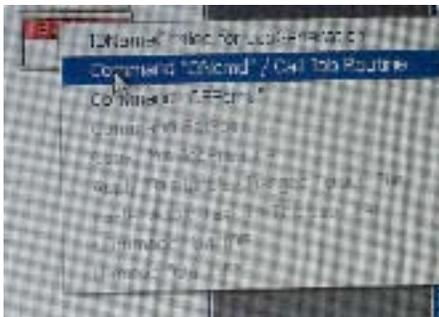
Now right-click on an analog control such as the Ion Gauge. Choose Setpoint. A window opens; note the units can be changed if it is a timer. The figures below show windows for a Timer, then a blank and filled Ion Gauge.

The bottom line shows the result of a 'Paste' of the above IonGauge setpoint into a JobFile:

Editing; ' LightCycleDemo '

Cancel, Discard and Exit | Find any entire field IDName to paste and right-click on Item. Paste into appropriate text field below ... | Apply and Continue | OK, Apply and Exit

Edit/View Item | Identity | Appearance | Connect Lines | Command | Feedback | Response | Response2 | Interlocks1 | Interlocks2 | Job Control

```
 0  IDName,BarLightRed;IDCurrentCmd,ONcmd
 1  IDName,CoreTimer;IDCurrentCmd,1;IDUserUnits,s;IDAnlgCompareCriterion,0
 2  IDName,CoreTimerMonitor;FBNameUn1,BarLightRed;ILNameun2,IonGauge;ILNameun3,LoadLock
 3  IDName,CoreTimerMonitor;IDCurrentCmd,4;IDUserUnits,sec;IDAnlgCompareCriterion,0;FBNameUn1,BarLightGreen
 4  IDName,BarLightWhite;IDCurrentCmd,ONcmd
 5  IDName,CoreTimer;IDCurrentCmd,1;IDUserUnits,s;IDAnlgCompareCriterion,0
 6  IDName,BarLightRed;IDCurrentCmd,OFFcmd
 7  IDName,CoreTimer;IDCurrentCmd,1;IDUserUnits,s;IDAnlgCompareCriterion,0
 8  IDName,BarLightGreen;IDCurrentCmd,ONcmd
 9  IDName,CoreTimer;IDCurrentCmd,1;IDUserUnits,s;IDAnlgCompareCriterion,0
10  IDName,BarLightBlue;IDCurrentCmd,ONcmd
11  IDName,CoreTimer;IDCurrentCmd,1;IDUserUnits,s;IDAnlgCompareCriterion,0
12  IDName,BarLightWhite;IDCurrentCmd,OFFcmd
13  IDName,CoreTimer;IDCurrentCmd,1;IDUserUnits,s;IDAnlgCompareCriterion,0
14  IDName,BarLightGreen;IDCurrentCmd,OFFcmd
15  IDName,CoreTimer;IDCurrentCmd,1;IDUserUnits,s;IDAnlgCompareCriterion,0
16  IDName,BarLightRed;IDCurrentCmd,ONcmd
17  IDName,CoreTimer;IDCurrentCmd,1;IDUserUnits,s;IDAnlgCompareCriterion,0
18  IDName,BarLightBlue;IDCurrentCmd,OFFcmd
19  IDName,CoreTimer;IDCurrentCmd,1;IDUserUnits,s;IDAnlgCompareCriterion,0
20  IDName,BarLightWhite;IDCurrentCmd,ONcmd
21  IDName,CoreTimerMonitor;IDCurrentCmd,0;IDUserUnits,sec;IDAnlgCompareCriterion,0
22  IDName,CoreTimerMonitor;FBNameOn1, ;FBNameOn2, ;ILNameOn1, ;ILNameOn2, ;ILNameOn3,
23  IDName,IonGauge;IDCurrentCmd,5e-4;IDUserUnits,torr;IDanlgCompareCriterion,-1
```

All of the above JobFile lines for digital commands and analog setpoints are entered using such Click-on-Item and Paste-in-JobFile methods. These lines can also be edited by hand, for example replacing "ON" with "OFF".

Other lines with more complex actions are filled by step-by-step assembly. For example, the Timer Monitor setup lines cause the FeedBack and InterLock tab fields of the Timer Monitor to be filled with IDNames and Condition Codes in order to cause the graphing and datalogging seen earlier. These Timer Monitor lines are built by clicking in the fields to be filled, as described in OnScreen Help as you mouse over the appropriate areas in the Edit Form's Identity, Feedback, and Interlock tabs. The click action 'scoops' the field descriptor, which is then pasted into the Job Control line. Then use your mouse to pick off a Condition Code from the Help Menu, or an IDName from a right-click on a screen Item, and paste that into the Job Control line.

One Job File can call another, again using a right-click on the called routine to take its data for pasting into the calling Job File line.

A caller Job File can overwrite a called Job File's line, and thus pass subroutine arguments and more, by clicking in the text field of that called Job File in its line to be changed, then right-clicking to take a copy of that line, and then pasting into the caller Job File and editing the line there. When executed, this line will be sent to the called Job File, causing the new line with its new parameter values to be executed.

DepEtchPro permits subroutines to be called to any level. DepEtchPro also permits wide latitude in the passing of arguments and causing the called subroutines to perform the needed tasks. In addition any number of Monitor Timers can be run simultaneously to datalog and graph, as well as abort to an error recovery routine in case of an error condition.

We have covered System Operation, System Design, and Job File Generation. These functions are sufficient for rapid automation of any process tool. Building upon the sample Setup file shown here, a similar simple vacuum system could be automated in less than half a day.