

Don Lancaster's

RESOURCE BIN

number seventy

Exploring PostScript PIC Flutterwumpers.

Our usual reminder here that the *Resource Bin* is now a two-way column. You can get tech help, consultant referrals and off-the-wall networking on nearly any electronic, *tinaja* questing, personal publishing, money machine, or computer topic by calling me at (520) 428-4073 weekdays 8-5 Mountain Standard Time.

I'm now in the process of setting up my new *Guru's Lair* web site you will find at (where else?) www.tinaja.com This is the place you go for instant tech answers. Among the many files in our library, you will find complete reprint sets for all of the *Resource Bin* and other columns. Plus a brand new *Research InfoPack Service*.

You will get the best results if you have both *Netscape Communicator* and *Acrobat Reader 3.0* installed.

PostScript PIC Flutterwumpers

For quite a long time now, I've been really enthused by *Adobe's* PostScript computer language. I'll routinely use PostScript well beyond its printing and graphics capabilities. What I find really exciting is that with some brand new happenings, PostScript can now expand into *robotics*.

Specifically by getting low end PIC microprocessors to speak PostScript!

PostScript and PIC's working with each other might be a killer combo for ultra low cost robotic apps. Especially when such things as reusing existing artwork, interfacing popular design programs, selecting fancy fonts, doing micro-sizing, tool path corrections, or exotic coordinate transformations.

And most of it is simple stuff that you can easily explore all by yourself. Opening up some mind-blowing new opportunities for you.

Others have been quietly extending and greatly improving on PostScript all along. PostScript now offers total web friendliness, transparent video titling apps, document scan conversion,

catalog indexing, interactive forms, internal font provisions, drivers that print anywhere, disability aides, and great heaping bunches more.

But the real biggie is that PostScript itself has recently gotten ridiculously easier to use. And lots more fun for you to play with.

Getting Started With PostScript as Language

The easiest way to start exploring PostScript is by using a *host based interpreter*. This is simply a program you run on your PC. You send the interpreter PostScript language code, and the interpreter carries out those commands for you.

Usually with nothing but a simple drag-and-drop.

There are normally three possible results from your PostScript code:

(A) You could generate a file that can be used to print or image or webify or CD ROMify a graphic picture. Such pictures are fully device independent, superbly high quality and an industry standard mix of actual images, text, line art, and fancy fonts.

(B) You can generate a simple log file that quickly gives you important answers directly on your screen.

NEXT MONTH: Don takes another look into secrets of web based research.

(C) You can write an output data file to disk. Such an output file can have virtually *any* format you want it to. In most *any* language!

These output files can then be used for anything from taking over control of a computer to running a hot tub.

PostScript can similarly get taught to read and act upon just about any input file format you care to. Details

are found in SIXCLICK.HTML.

There are two very popular host based PostScript interpreters. The first and older is that freebie *GhostScript* shareware from *Aladdin Systems*. The second is the *Acrobat Distiller* that is part of the commercial *Adobe Acrobat* package. By using a student discount, *Distiller* is available for under \$55.

Get Version 3.01 or higher.

Do not let the name mislead you. *Acrobat Distiller* is really a full blown PostScript language interpreter that is easily and quickly applied for totally general purpose computing.

When combined with a companion *Acrobat Exchange* viewer, you'll get everything important that genuine *Display PostScript* offers. Especially the ability to have source code and fast updating graphic results side by side on your screen at once.

Distiller also acts as "somewhat" of a compiler. In that it automatically reduces PostScript code into the bare minimum "just the facts ma'am" and fast running compressed .PDF format. For instance, an input PostScript proc may include very elaborate fill justify calculations or math-intensive plots. The output PDF format simply shows where to mark the pages or what final numeric data values to output.

Which PostScript interpreter is for you? My own preference is to always use the *Acrobat Distiller*. *Distiller* is faster, way easier to run, much better supported, is fully up to date, has only official bugs, and gives you more attractive final results. On the other hand, *GhostScript* is free and gives you better error reporting.

Two Examples

Let's look at two examples to get you started off using PostScript as a programming language. Assume you want to find the sine of sixty degrees. Punch this code on into your favorite editor or word processor...

%!
 % find sine of 60 degrees
 60 sin ==

...and then save it as either a .PS file or as an ordinary .TXT text file. Drag and drop this file into Distiller or GhostScript, and an output value of 0.866025 should promptly appear in the log window on screen.

We can see several things here. A PostScript file consists of an ordinary text file containing more or less plain English words and numbers. Other more compact file formats are also available for advanced users.

At a minimum, your file *must* start off with a "%!". To be formally legal, special *document structuring convention* comments are also strongly suggested for use in fancier projects.

Any line which starts with a "%" is treated as a comment. Thus, only the third line of your program actually does anything useful. This line places a numeric value of 60 onto an internal *stack*. A *sin* command then obeys the rule "take the value in degrees on the top of the stack, find its trigonometric sine, and replace the top stack value with this calculated result.

That odd "==" is shorthand for "intelligently print the top stack value to your log file".

Should the PostScript program also produce some graphic or printable output, you simply click on Exchange to view it on screen. Your round trip edit-interpret-view time is typically well under ten seconds and can be done in as little as six mouse clicks.

In PostScript, we have a powerful and quite modern general purpose language that is *interpetive* (you tell it what to do ahead of time); is *stack oriented* (everything goes on or off one or more stacks); *Polish structured* (you give it some values first, followed by your needed instructions that act on those values); is quite *object oriented* (individual procs can be manipulated with uniformly predictable results); *threaded* (any command can call any other); *weakly typed* (most any data structure can get converted to any other); *reentrant* (commands can call themselves); *dictionary intelligent* (the grouped key-value pairs play a big role); *redefinable* (you can add new commands or procs and change any existing ones); and *device independent* (the same code runs on any host and controls any PostScript device of any quality level).

Let's look at a second and fancier

app. Finding out who accesses your website when is a big deal these days. There's lots of fancy programs (such as *WebTrends*) which can give you all sorts of exotic plots and graphs. But only with horrendous use of memory and disk space.

PostScript can easily read and then interpret web log files. A no-cost 5K PostScript file can trivially extract log file data that the big programs may miss. Such as analyzing the loyalty of repeat visitors. Or maybe tracking the popularity of a brand new file as it moves up through the ranks.

The site analysis code is a tad too long to show you here. But you can grab examples of these programs as [WEBSITANI.PS](#) or [WEBLOGU2.PS](#) off my www.tinaja.com/pslib01.html

There is extensive documentation inside either file. All you have to do is rename your target and destination files for your own use.

There are dozens more PostScript as language examples where these two came from.

Recapping, PostScript is now really fast, fun, and easy to use. You first create a text file, then drag and drop it into Distiller, and then you optionally view it with Exchange. Reading any file format or creating any new file format or generating graphics.

Robotic Flutterwumpers

What does all this have to do with robots? Let's backtrack a tad. I like to call a *flutterwumper* any robotic that moves and either chomps or spits.

Printed circuit drills, sign routers, engravers, silk screen cutters, *Santa Claus* setups, or animation stands are typical examples.

The basics of flutterwumpers are shown in [FLUTWUMP.PDF](#).

Let's say you decide to build a custom 2-1/2 D flutterwumper. One that has independent X and Y motions in small steps, but only allows a gross "tool up" or "tool down" in your Z direction. Let us further assume that fonts play an important role and that you want to access large libraries of existing symbol icons or whatever. As well as use standard CAD, *Mathlab* or *Illustrator* design software.

A PIC or two is the obvious choice to handle all your low level custom flutterwumper controller. But how on earth can you get it to do the fancy stuff? While still remaining both low in cost and elegantly simple.

Your secret lies in the interactions between...

new from DON LANCASTER

ACTIVE FILTER COOKBOOK

The sixteenth (!) printing of Don's bible on analog op-amp lowpass, bandpass, and highpass active filters. De-mystified instant designs. **\$28.50**

CMOS AND TTL COOKBOOKS

Millions of copies in print worldwide. **THE** two books for digital integrated circuit fundamentals. About as hands-on as you can get. **\$28.50** each.

RESEARCH INFOPACKS

Don's instant cash-and-carry flat rate consulting service. Ask any reasonable technical question for a detailed analysis and complete report. See www.tinaja.com/info01 for specifics. **\$75.00**

INCREDIBLE SECRET MONEY MACHINE II

Updated 2nd edition of Don's classic on setting up your own technical or craft venture. **\$18.50**

LANCASTER CLASSICS LIBRARY

Don's best early stuff at a bargain price. Includes the CMOS Cookbook, The TTL Cookbook, Active Filter Cookbook, PostScript video, Case Against Patents, Incredible Secret Money Machine II, and Hardware Hacker II reprints. **\$119.50**

LOTS OF OTHER GOODIES

| | |
|-------------------------------|----------|
| Tech Musings V or VI | \$24.50 |
| Ask the Guru I or II or III | \$24.50 |
| Hardware Hacker II, III or IV | \$24.50 |
| Micro Cookbook I | \$19.50 |
| PostScript Beginner Stuff | \$29.50 |
| PostScript Show and Tell | \$29.50 |
| Intro to PostScript Video | \$29.50 |
| PostScript Reference II | \$34.50 |
| PostScript Tutorial/Cookbook | \$22.50 |
| PostScript by Example | \$32.50 |
| Understanding PS Programming | \$29.50 |
| PostScript: A Visual Approach | \$22.50 |
| PostScript Program Design | \$24.50 |
| Thinking in PostScript | \$22.50 |
| LaserWriter Reference | \$19.50 |
| Type 1 Font Format | \$16.50 |
| Acrobat Reference | \$24.50 |
| Whole works (all PostScript) | \$380.00 |
| Technical Insider Secrets | FREE |

BOOK-ON-DEMAND PUB KIT

Ongoing details on Book-on-demand publishing, a new method of producing books only when and as ordered. Reprints, sources, samples. **\$39.50**

THE CASE AGAINST PATENTS

For most individuals, patents are virtually certain to result in a net loss of sanity, energy, time, and money. This reprint set shows you Don's tested and proven real-world alternatives. **28.50**

BLATANT OPPORTUNIST I

The reprints from all Don's Midnight Engineering columns. Includes a broad range of real world, proven coverage on small scale technical startup ventures. Stuff you can use right now. **\$24.50**

RESOURCE BIN I

A complete collection of all Don's Nuts & Volts columns to date, including a new index and his master names and numbers list. **\$24.50**

FREE SAMPLES

Check Don's Guru's Lair at <http://www.tinaja.com> for interactive catalogs and online samples of Don's unique products. Searchable reprints and reference resources, too. Tech help, hot links to cool sites, consultants. email: don@tinaja.com

FREE US VOICE HELPLINE VISA/MC

SYNERGETICS
 Box 809-NV
 Thatcher, AZ 85552
 (520) 428-4073

FREE Catalog: <http://www.tinaja.com>

POSTSCRIPT PIC FLUTTERWUMPER RESOURCES

Adobe PostScript
1585 Charleston Rd
Mountain View CA 94039
(800) 833-6687

Scott Edwards
964 Cactus Wren Lane
Sierra Vista AZ 85635
(520) 459-4802

Hewlett-Packard
PO Box 10301
Palo Alto CA 94303
(415) 857-1501

Parallax
3805 Atherton Rd #102
Rocklin CA 95765
(916) 624-8333

Videonics
1370 Dell Avenue
Campbell CA 95008
(408) 866-8300

Aladdin/Ghostscript
Box 60264
Palo Alto CA 94306
(415) 322-0103

Gerber Scientific
83 Gerber Rd
S Windsor CT 06074
(203) 644-1551

Microchip Technology
2355 W Chandler Blvd
Chandler AZ 85224
(602) 786-7200

Synergetics
Box 809
Thatcher AZ 85552
(520) 428-4073

WEB Trends
621 SW Morrison #1025
Portland OR 97205
(502) 294-7025

Flutfiles and PIC's

For some strange reason, I seem to be having some difficulties burning a full PostScript version 2019 interpreter into the 12C508 baby PIC. Mostly, if I did so, it just wouldn't be very fair to Motorola or Intel.

Let's instead *partition* the problem. Let host PostScript software do the sort of things it is really good at. Let a small and low cost PIC do all your flutterwumper stepper controlling. And let's dream up some way that PostScript can talk to PIC's. And that PIC's can talk back to PostScript.

We have two obvious choices here that certainly should work. *Hewlett Packard* has long had its HPGL plotter language. And there's the *Gerber File Format* used on fancier photoplotters. Either of these will work just fine. But they might end up gross overkill for a low cost flutterwumper. And either will tie up a substantial part of your PIC's resources.

Instead, let's get elegantly simple. Let's create a new ultra simple meta language. One that's PIC optimized on one end and PostScript optimized on the other.

I'll call this one a *flutfile*.

We will limit our new language to ordinary printing ASCII characters and carriage returns in a plain old textfile. We will use the rule of *one character per action*. Now, this might seem slow and cumbersome, but you usually have more than enough time available on any mechanical motion flutterwumper. And the economics are stunning.

On a 2-1/2 D flutterwumper, there will only be eight possible horizontal actions. You can go one step positive or negative in directions of X, Y, or both at the same time.

So, let the ASCII numeral 0 be east. 1 be northeast; 2 be north. And so on around to a 7 at southeast. To round out your commands, add a U for up, a D for down, and a H for home. And

maybe a few initializing, debug, and repeat commands.

All your PIC has to do is receive a dozen or so serial ASCII characters and then act upon them. For instance, command "0" causes one full step in the positive X direction. Command "1" moves both the X and Y steppers one full step. For max smoothness, your individual partial steps between X and Y can get *interleaved*. Command "2" moves Y only, and so on.

Now, all that PostScript has to do is accept programming commands and generate a meta file. Which is a lot easier than it sounds. I've got lots of ready-to-run code for you.

Let me briefly outline the PS to flutfile process: The Distiller normally reduces all wanted motions to *moveto* positioning, *lineto* straight line paths, *curveto* curved path generation, and *closepath* operators. We then eliminate any *curveto*'s with use of PostScript's *flattenpath* command.

Since any command in PostScript can be redefined at any time for any reason, your remaining stock *moveto* and *lineto* commands are intercepted and *replaced* with your custom code. Among other tasks, this custom code does vector-to-step conversion at the exact resolution needed by your PIC and saves it in a flutfile format.

Detailed flutfile conversion tutorial examples are found in [POSTFLUT.PDF](#), [FLUTOOLS.PS](#), [FLUTDEMO.PDF](#), and [VECTSTEP.PDF](#) on my website.

Included are specific programming details on how you handle boxes, circles, and fancy font characters. You can use these as preliminary test files for your own PIC design.

How it Works

Here's a summary of how to bring PostScript to any PIC robotic: Install Acrobat Distiller and Exchange 3.01 or better on your host PC. Find suitable project material, from CAD programs that provide a PostScript output, from *Illustrator*, or by fun writing your own

raw PostScript like I always do.

Use Distiller and a custom software module to convert your pattern files into flutfiles, the simple step-by-step commands that a low end PIC can understand and easily deal with. Such magic tricks as tool path adjusting or fancy coordinate transforms could also be included in this step.

Save your flutfile as an ordinary textfile. Gather up this flutfile with a comm program or whatever and send it to your PIC. Obviously, the flutfiles can be saved for reuse, broken up into module libraries, get web distributed, or combined into projects.

Suitable handshaking takes care of machine busy times.

This two-step process thus brings PostScript to your PIC's simply and cheaply and elegantly.

For More Help

That Adobe "red" book (*PostScript Reference Manual*) and their Adobe "blue" book (*the PostScript Tutorial & Cookbook*) are your standard starting points to learn and love PostScript. These could get combined with my *PostScript Beginner Stuff* coursework or the *Whole Works* package that lets you get into PostScript all at once and really big time. Check out my nearby [Synergetics](#) ad for details.

Lots more info on exactly how to bring PostScript to PIC's appear in the previously noted files on my website. For bunches more background, also check the *PostScript*, *Acrobat*, *PIC*, and *Flutterwumper* library shelves found at www.tinaja.com

More on the PIC's themselves from manufacturer *Microchip Technology*, from *Parallax*, and from *Scott Edwards Electronics*. Some 600+ annotated hot links to useful PIC sites are found at www.tinaja.com/pic500.html and also in www.tinaja.com/picwb01.html

More on those briefly mentioned PostScript transparent video apps by way of www.videonics.com. And, of course, lots more on PostScript and

Acrobat at their www.adobe.com.

More technical help, *InfoPacks*, full consulting, custom programming, or development is available by emailing me through don@tinaja.com.

This Month's Contest

For our contest this month, just tell me what you would do with your PostScript speaking PIC. Especially one of the new minidip baby PIC's.

There should be a largish pile of my new *Incredible Secret Money Machine II*

books going to the dozen or so better entries, plus an all-expense-paid (FOB Thatcher, AZ) *tinaja quest* for two that will go to the very best of all.

Send all your *written* entries to me here at *Synergetics*, rather than to *Nuts & Volts* editorial. ♦

Microcomputer pioneer and guru Don Lancaster is the author of 33 books and countless tech articles. Don maintains his no-charge US tech helpline found at (520) 428-4073, besides offering all of his own

books, reprints, and consulting services. Don also offers a free catalog full of his unique products and resource secrets. The best calling times are 8-5 on weekdays, Mountain Standard Time.

Don is the webmaster of his Guru's Lair at <http://www.tinaja.com>

Full reprints and preprints of all Don's columns and ongoing tech support appear here. You can reach Don at Synergetics, Box 809, Thatcher, AZ 85552. Or send any messages to his US Internet address of don@tinaja.com