by Don Lancaster

Winning the Micro Game

What are the key secrets to using and profiting from microcomputers and microcontrollers? Over many years, I've gathered together a list of fundamental guidelines that seem to work for me…

Hands on is everything– The only real way to ever learn anything about micros is to jump on in with both hands and both feet and do lots of computing. Until you actually *do* and *see* what all of these micro opportunities are all about, you have accomplished absolutely nothing. You have to do things by yourself and on your own terms.

You become computer literate by using computers, not by having someone tell you about them. Or by passively reading about them.

You have to make mistakes– If you are into any sort of micro product development *half of your experiments must fail*. A canned set of exercises on some micro trainer is next to worthless when everything falls in place and works perfectly the first time. That's simply not where it is at.

In the micro world, you make mistakes to learn and to progress. You should expect mistakes. Prepare for them. Welcome them. Aggressively seek them out.

You must mix hardware and software– There are still some heads-in-the-clouds types of pure software people out there that still labor under the delusion that hardware is simply a mundane inconvenience standing between them and "pure" computing. And there are technician types who do everything with baskets full of obsolete chips.

Neither approach is any good. Sometimes, a simple and cheap hardware circuit can replace lots of lines of software. At other times and in other places, a few words of elegant software can eliminate the need for custom circuits or for inflexible and special hardware.

Winning micro products will combine *both* hardware and software to optimize the best features of each. To give you the simplest system at the lowest possible cost.

The real world is fuzzy– There are lots of textbooks and experiments that work every time with everything nice and clean. All ends are neatly tied up. Exactly what you need to do the job. No more. No less.

Unfortunately, reality just doesn't work that way. First, there will be people to deal with and that will always foul things up. Key stuff will be missing or late. The magic chip may be a figment of an ad writer's wet dream.

Or a problem may have a simple and cheap technical fix. But one that is politically or socially unacceptable. Goals might conflict. So do egos. Some problems are designed from the ground up to be purposely unsolvable.

Especially when one person's solution turns out to be another person's threat.

Expect and accept fuzziness. Recognize that there really isn't much of real world beginnings and endings. Things sort of dribble off into the great whatever instead.

Hit the fundamentals hard– For most micro uses, it makes no difference whatsoever which micro from which family you select. Even if there temporarily was a "best" micro, other factors including your own skills and attitude, the available support, the elegance of competitor's code, and so on can reduce any advantage of a "best" micro to zilch.

To beat this, hit the basics hard. All known systems have address space and address modes. All involve subroutines, clocks, ports, memory, I/O, and so on. Get the basics down solidly in the most general way that you can. But get the essentials down solid.

Reach out and put the touch on someone– The nickels are still to be made in places where end users are not yet using micros. Or are not fully utilizing distributed and embedded intelligence. Walk out into the real world and locate those places. Get involved in these areas on their own terms.

Put the latest single chip designs to work feeding cows. Or treating sewage, gambling on Wall Street, designing looms, mixing cement, baking calzones, or milking goats.

Or controlling hoists, hulling pecans, animating video, co-oping groceries, hybridizing sinsemella, improving wood stoves, altering bicycles, restoring steam calliopes, monitoring stream gauges, selling paper clips, cutting dress patterns, teaching trumpets, and the zillions of other places yet to discover the extreme enabling power of low cost and locally embedded intelligence.

Create a personal network– One made up of individuals outside of your family, work, or local community. Do this through online contacts, schools, seminars, trade shows, phone calls, user groups, clubs, web pages, whatever.

A good network expansion tool is to *ask answerable questions*. On subjects the contactee genuinely wishes to talk about. A second is the old Dale Carnegieism of *be a good listener*. And, of course, the third is to *suppress your ego*. Let them do most of the talking most of the time.

Steal the plans– Just about all of the needed and obvious background tasks involving micro aps have already been done. These are readily available if you dig deep enough.

Always find out what has gone before you. Do this by a careful study of code listings, technical articles, help lines, developer groups, tutorials, data books and system docs. Reverse engineering can be an exceptionally powerful and lots-of-fun learning tool. Always ask "Has anyone thought about this before?" Then find out exactly who they were and where they thought about it.

Get yourself online– Yeah, the Internet is a mind boggling collection of dregs and drivel that can be a monumentally addictive time waster. On the other hand, properly used net access can find you instant contacts and useful answers. While nobody has yet to turn a profit on direct sales on the net, it is only a matter of time before cybercash becomes accepted and widely used. But to tap this, you have to use the net model of: *Give something away before you attempt to sell something*.

To register your own Internet web site, you first contact www.internic.net Then follow their instructions.

The other side of online are those commercial services. Such as *America Online*, *CompuServe*, *GEnie*, or *MSN*.

Here, you often will get what you pay for. Higher quality information more up to date and vastly more concentrated. Your ultimate online source for an instant technical answer to anything still remains the *Dialog Information Service*.

Better wrong now than right later– In anything you do with micros, your first attempt *will* be wrong and *will* have to be done over. So, it pays to just kluge any old something up immediately and let it show you the proper way to go. Often, you won't even understand what the real problem is until you are on the inside and looking out.

Try something simple, quick, and dirty that at least does something in roughly the direction you want to go. Make some guesses. Take a stab at it.

Get something flying at least more or less right side up. Later on, you can go on back and improve things. Adding structure to your programming, elegance to your methods, convenience to your user and simplicity to your hardware.

Add the final spit and polish on the way out the door, not early in the game.

Aggressively subscribe to the trade journals– By far your most important resources anytime ever are all of those free industrial trade journals. These are absolutely outstanding "must have" bargains. For technical information, suppliers, samples, data books, more. Typical electronic examples are *E. E. Times*, *Electronic Design*, *EDN*, *Electronic Products*, *EE Products*, and *Electronic Component News*.

I personally subscribe to over 500 trade journals. Your best starting point is with *Ulrich's Periodicals Dictionary*. Found online or at your local libary.

Write it down– And not in chalk on the back of an old envelope either. Documentation is the password to avoid self-destructive behavior. You do not simply record final programs and schematics. Instead, you keep track of what you did and why you did it. Know where you have gone and where you are heading. Software is worthless if you can't show someone else *exactly* how to use and maintain it. Hardware has no way to operate if there is no way to connect or fix it. Be able to go back and reuse or modify what you did a week or a year ago.

Don't separate work from play– Figuring out what to do with the oily slime in some adventure game is vastly more important than designing an efficient sort algorithm for a business general ledger. Because the oily slime is what is turning people on to microcomputing.

Everything run on a computer is a game! It's just that stuffy institutions, banks, bureaucrats, and other so-called "serious" computer users have rules which say you are not supposed to smile while you are playing their games.

The most exciting, most challenging, and the deadliest adventure games of all are played on spreadsheets.

Nail down all resources– Do become a collector of data books, software, and article reprints. Especially if you live in a remote area. You should always be your own best reference library. This must be an ongoing activity done well ahead of any actual need. High momentum stays with any project where you can simply reach up and grab your own immediate answer.

Aim for a minimum of one hundred lineal feet of data books and not less than two cubic yards of raw data in your personal resource files. Initially, grab anything at all. As the resource pile builds, you can become more critical and more selective.

You will never get enough– No matter how far you go in microcomputing and no matter how much of what kind of hardware and software you have on hand, you will *always* need "more" of something. More memory. Faster clocks. *Rev 3.0.2*. There never is nor will there ever be any time when you have "enough" of anything.

You will find only one way out of the "more" syndrome: *Always go with what you have*. Make it work. Live with it as long as you can. Force it to pay its own way.

Make it do. Use it up. Wear it out.

No patents– For the overwhelming majority of individuals and smaller scale startups, patents are virtually certain to result in a stunning loss of time, energy, money, and sanity. One reason is that the mythological urban lore surrounding the patent process is nearly always dead wrong. A patent is merely a sheet of paper that sometimes gives you the right to sue someone. And nothing more.

A second reason is that very few patents are genuinely new. There is not one patent in one thousand that cannot be busted through a diligent enough search for prior art done in obscure enough places.

Third, the costs of enforcing and defending your patent place the breakeven costs of the system ($12 to $40 million in net product sales) way above what you'd first expect. To the point where: *It is absolutely insane to try and patent a million dollar idea.*

Always ask "Why are you telling me this?"– The really good and useful products and concepts are *never* heavily advertised. In fact, anything genuinely useful takes a lot of time and trouble to nail down.

When anything is widely advertised, it more than likely means that there might be something much better available elsewhere. If someone is radically trying to convert you to his microcomputer or his way of doing things, the chances are that he has gone way far off into right field and gotten snookered into a bad scene. He is looking for converts to ease the pain when he gets shot out of the saddle.

MENTIONED RESOURCES

America On-Line
8619 Westwood Cntr Dr
Vienna VA 22182
(800) 827-6364

CompuServe
5000 Arlington Center Blvd
Columbus OH 43220
(800) 848-8199

Dialog Information Svc
3460 Hillview Ave
Palo Alto CA 94304
(415) 858-2700

EDN Magazine
275 Washington St
Newton MA 02158
(617) 964-3030

EE Product News
707 Westchester Ave
White Plains NY 10604
(914) 949-8500

EE Times
600 Community Dr
Manhassat NY 11030
(516) 365-4600

Electronic Comp News
1 Chilton Way
Radnor PA 19089
(215) 964-4345

Electronic Design
611 Rt #46 W
Hasbrouck Heights NJ 07604
(201) 393-6060

Electronic Products
645 Stewart Ave
Garden City NY 11530
(516) 227-1300

GEnie
401 N Washington St
Rockville MD 20850
(800) 638-9636

InterNIC Registration Svc
505 Huntmar Park Drive
Hernon VA 22070
(703) 742-4777

MSN/MicroSoft Net
One Microsoft Way
Redmond WA 90852
(800) 877-1900

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

Ulrich's Dictionary
121 Chanlon Rd
New Providence NJ 07974
(908) 771-7714

Whenever anyone tries to tell you anything involving micros, always ask "What is the *real* reason you are telling me this?" Find their motives. Get a second opinion, check anouther route, or find a different viewpoint.

Always be a skeptic. Question everything.

Encourage open systems– It's been proven time and time again that open systems having fully documented interfaces are the ones that survive. Design all you do using "plug in" capabilities that let others expand upon your products.

By the same token, paranoid secrecy of product designs guarantees your failure every time. It is only through open communications with others that the optimum solutions and opportunities can be explored.

Practice elegant simplicity– I'd pick a peck of PIC's over a porcinely portly Pentium any time. Seek out the "do more with less" solutions. The ones which leave all the experts shaking their heads in stunned "why didn't I think of that" disbelief. Stay lean, clean, and mean.

By far the greatest invention of the twentieth century was the P-38 can opener. Seek elegant simplicity in other fields and study the real winners.

The micro is a mirror– A jackhammer is a good example of a tool that does some things very well, but forces you to play things its way. Mention the word "jackhammer", and you can immediately stereotype its user, down to the tavern he frequents, his beer (it's pronounced "airn" in his native Pittsburgh), and his favorite sport of bowling.

Micros are the exact opposite. The micro is a mirror that *reflects* the personality of its user. One sees the micro as an artistic tool. Another as a business machine. Another plays music on it, while yet another may use it to regulate the temperature ventilators on his hog farm.

Each approaches the micro in his own way. The micro in turn amplifies and reflects the personality of the user.

So, feel free to use a micro any way you like, rather than the way "they" want you to. But also recognize that you should never force your perceptions of what a micro is or what it should do upon others.

What gets reflected will always depend upon who is doing the looking.

But most important of all…

I guess those cover most of the essentials. But if I had to reduce it all down to a single most important key success guideline, I'd pick this one…

Go with your own vibes– There is no "right" or "wrong" direction with microcomputers. Most of it remains totally unknown, unexplored, and uncharted. So, if "they" insist on something, most often "they" don't know what they are talking about. If you are interested in a tech something and want to go in that direction, fine. Do it!

Your surest bet to long term winning is to roll with your own vibes. Explore what you want to. Ignore the herd that is thundering the other way. Get off the beaten path.

You are by definition, the center and the most important part of the micro universe.

So, make yourself your own best customer. Satisfy your own needs and your own curiosity. Place as much psychic energy and personal value added into the routes that you pick and you are certain to win the micro game.

For More Information

Much more on creating your own tech venture appears in my *Incredible Secret Money Machine II* book. Finding and locating useful product and info sources are covered in my *Resource Bin* collection, per my nearby *Synergetics* ad.

Tested and proven alternates to patents and patenting are found in my *Case Against Patents* package. And also in my WHEN2PAT.PDF. More on elegant simplicity appears as file ELESIMP.PDF on my *www.tinaja.com* web site. And in my *Blatant Opportunist* bound reprints. Also check out RESBN48.PDF for the fundamentals of tech startups. And RESBN08.PDF for more on industrial trade journals.

I have also just started a web site. This one is still under construction and I'm not sure just how the better offerings will get funded. Included are *Blatant Opportunist* column reprints, the *Synergetics Consultant's Network*, and scads of annotated links to other web sites.

Be sure to come visit at www.tinaja.com ✥

Microcomputer pioneer and guru Don Lancaster is the author of 33 books and countless articles. Don maintains a US technical helpline you'll find at (520) 428-4073, besides offering all his own books, reprints and various services.

Don has a free new catalog crammed full of his latest insider secrets waiting for you. Your best calling times are 8-5 weekdays, Mountain Standard Time.

US callers only, please.

Don is also the webmaster of www.tinaja.com where a special area has been set aside for Midnight Engineering readers. You can also reach Don at Synergetics, Box 809, Thatcher, AZ 85552. Or email don@tinaja.com