Hardware, Software & Robotics

By Don Lancaster

For readers who didn’t catch me last month, I’ll try to answer most any hardware (or related software) questions from you, as well as look at some neat hacker-type stuff. To join in the fun, write or phone me per box at the end of the column.

Show me a tough computer port driver circuit.

The usual microcomputer port chips, such as 6822, a 6832, or an 8212, are typically NMOS chips with very low drive abilities. While fine for just getting ones and zeros into or out of a micro circuit, they simply cannot drive higher power loads.

Figure 1 shows you a medium-power “amplifier” for computer output ports, using the Sprague ULN2813 octal peripheral driver. This circuit is best suited for things that need less than 40 volts and less than half an ampere, and where there are no safety isolation hassles noise problems.

Obvious uses include small incandescent lamps, solenoids, air valves, relays, small stepper motors, multiplexed LED arrays, hammer drivers, and the like. I’ve even used it as a “student proof” driver for plain old light-emitting diodes. Cost is under $3.

The chip holds eight darlington transistor pairs, each of which can drive up to half an ampere and block up to 40 volts. You are not allowed to run all eight outputs at full current at the same time, but that derating curve is very generous. You can easily drive the 2813 directly from almost any micro port chip. A few milliamperes at the input is all you need.

Note that a separate power supply is recommended for the loads. Note also that a common ground must be shared between the loads, the power supply, the ULN2813, and the micro that is sourcing the ones and zeros to be output. It is especially important to avoid ground loops when you have high-power loads sharing the same ground circuitry as the computer circuitry.

Since the darlington pairs invert, a high at the input produces a low at the output. In turn, this lights, powers, or turns on the load. A low at the input produces a no-current high at the output, which unlights, depowers, or turns off the load. Thus, you get what you would expect—a positive logic one turns on the load.

There are also eight freewheeling diodes in the package. These diodes must be used if you are powering an inductive load. The common connection to all the diodes is connected to the positive end of the load voltage. It is a good idea to always connect this protection circuitry for most loads.

How can I get started in assembly language?

Check into the “top 30” programs for any major personal computer, and you will find that 30 out of 30 either run directly in machine language or else make extensive use of linked machine-language routines. The marketplace has spoken—if you want to sell a decent program, it must run in machine language. BASIC and Pascal need not apply.

The usual way of getting a program to run in machine language is to use an assembler program. The assembler speaks assembly language to you and machine language to the computer, letting you automate and simplify the creation of machine-language routines.

So how do you get started? The
best way I know of involves four steps. First, and by far most important, is to hand code and hand debug several hundred lines of machine language code without use of any assembler at all. Then check out a simple mini-assembler, acting as an "automated pocket card." Then tear apart the winning programs of others as a detailed study to find out how it's done. Finally, step up to a full-blown assembler or macroassembler of your choice. Then get with the program.

The tools and resources you need along the way will, of course, change with your choice of microprocessor and personal computer. Let's assume you are a 6502 person doing Apple II development, and look at the tools I feel are important. Naturally, if you are into other things, you'll need "alike but different somehow" tools that one-on-one replace these.

Start with a 6502 pocket card. These are free or cheap from Rockwell, Synertek, MOS Technology, and Western Design Center. Larger plastic cards are available from Micro Logic and are sometimes easier to use. You will also need a programming manual for the micro you are using. The 6502 Programming Manual, again by Rockwell, Synertek, and MOS Technology is absolutely essential. A 6502 Hardware Manual is also available, although not quite as useful as the programming manual.

You will also need some books on machine-language and assembly-language programming. I'm laboring under the delusion that my Micro Cookbook I (SAMS #21828), Micro Cookbook II (SAMS #21829), and my brand new Assembly Cookbook (SAMS #22331) can help you bunches here.

You will need some debugging tools. First and foremost, you will need a way to unconditionally reset your machine so you can stop any program at any time for any reason. (An absolute reset mod for the IIE and IIc appears elsewhere in this issue.)

Next needed is a way to mini-assemble, single step, and debug a working program. There was a mini-assembler in older Apples available in the monitor at $F666 and activated by an $F666G while in the "old" Apple II ROM. This has recently been upgraded into a major and essential debugging package called the BUG-BYTER. This is available from Apple Computer as part of DOS Toolkit #A2W0011.

You will also need a disassembler or two.

A dumb disassembler is built into all Apples, activated by getting into the monitor and typing the address, followed by an L, such as "$0800L." For serious use, though, you will want to get an intelligent disassembler that lets you capture the source code of any program you care to. I think that DISASM IIE by Rak-Ware is a good choice. (It is the only one I personally use.)

The technical reference manuals for your machine are obviously needed. Normally, these do not come with the machine but have to be separately ordered. Apple offers a IIe Technical Reference Manual (#A2L2005), a IIc Technical Reference Manual (#A2L4030) and Inside MacIntosh. It is unthinkable to attempt any useful programming without these products.

You will need a good set of books for your disk operating system. On the Apple, Beneath Apple DOS and Beneath Apple ProDOS by Quality Software are essential. Naturally, you will also want the factory manuals for your particular disk system. Same goes for printers and modems, of course.

Oh yes. Somewhere along the way you will need an assembler program, although this is only a tiny part of the resources you need to do assembly work. Apple has newly overhauled their EDASM assembler so it now includes macros and lots of other new goodies. It's available on either the DOS or ProDOS toolkits. One neat thing about EDASM is that it lets you do "new way" editing, using Applewriter to make entry and editing far simpler and far more fun.

Of the two dozen or so other Apple assemblers, some people consider the "best" one to be the S-C Assembler, while the "best buy" is Big Mac by CALL A.P.P.L.E.

The final resource for assembly programming is you yourself with lots of hands-on programming experience. In theory, you can get up to where you can just barely write second-rate assembly- or machine-language programs in less than three years of solid and dedicated effort. Only it hasn't happened yet. Not even once.

But, as someone once said, the longest journey starts with a single step. And the reason for doing it, of course, is for the nickels, since there is no other alternative to writing winning and commercially successful programs.

Have at it!

Any ideas on low-cost robotics?

It ever ceases to amaze me that low-pressure pneumatics has never taken off. Yet air pressure systems in the 3-to-6-psi range have enormous advantages.

First off, low-pressure air is cheap and low-tech. It gives you lots of linear force easy and simply. It can
“amplify,” since most of the “muscle” comes from the air source. This means you do not directly have to provide high-power electronic drivers as you do with solenoids or servo motors. Air also goes around corners beautifully, particularly robotic elbows.

Most important to us, low-pressure air has fantastic hacker potential. You can literally beat one of these systems out on a brick in your back yard, and still come up with a promising and useful product.

You will need an air source. The larger aquarium pumps are ideal, when combined with a small storage vessel. The latter are easily made from plastic toilet tank floats. You can instead use a storage tank, filled with a tire pump, or even a scuba tank or a Scott airpack bottle.

A regulator is also needed. I’ve had best luck running at 3.5 psi. You can get regulators for around six bucks from most of the surplus houses mentioned below.

Now for the fun part. You need a control valve. Unlike electronics, you cannot simply connect your air supply to an actuator to move it and then disconnect it to return. Try this, and the air stays stuck in the pipe and the actuator will remain extended. So, you need three-way air valves that act like electronic spdt switches. Powered, the air supply gets connected to the actuator. When the control signal is turned off, the pressure remaining in the actuator vents to ambient through the second arm of the valve.

You simply will not believe the price of an electronic three-way air valve for low pressure robotics. How does thirty cents sound? Thirty cents!

They are called automotive EGR valves, and originally were made by Carter Carburetor. For many years now, they have been available surplus from Jerryco, C&H Sales, Herbach and Rademan, BNF Sales, Surplus Center, and all of your usual mechanical surplus outfits. Your nearest handy-dandy junkyard is also crammed full of them. Incredibly, these have gone begging.

Jerryco has far and away the lowest valve pricing, the most outrageous catalog, and the most bewildering array of other goodies. Where else under one roof can you get a real joystick for a B-17 bomber, some lumber grading crayons, scented ballpoint pens, candle molds, or a dummy howitzer shell? Check them out.

The valve is shown in Fig. 2. You power it with +12 volts dc to connect the air source to the load. Removing power vents the load to ambient through a filtered vent at the rear of the valve coil. The side pipe goes to the load. The front pipe goes to the air source. You can make a simple manifold for several valves by taking a piece of thick, large-diameter tubing and punching one pinhole in it for each valve. Then push the front arm of each valve into the pinhole. A push-on pressure seal is all you need at 3.5 psi. The driver shown in Fig. 1 works beautifully on these.

What do you use for actuators? Traditional miniature air cylinders are available from Clippard. Unfortunately, they are also available at traditional prices. Instead, let’s get nontraditional. There’s a key secret to any useful low-power, hacker-based pneumatic actuator: never have a seal that must move! All of your low-power air will either be wasted trying to move the seal, or else will leak right by it.

Figure 3 shows three different ways to build actuators that do not have moving seals. The rolling diaphragm is interesting; suitable diaphragms are available from Bellofram.

I like the bellows actuators myself. I actually had a bunch of these blow molded long ago, to use in a pneumatic attachment to use an office Selectric typewriter as a computer printer. It worked, more or less, but never became a viable product. Polypropylene is probably the best choice of material. Some scientific supply houses sell bellows pipettes that are almost useful as bellows actuators. I’ve also ripped apart those “pump the water” toys that were available a few years back. The bellows pump inside these worked fine.

Chances are that you will prefer the “bladder” or “balloon” actuator instead, since these are the cheapest and
are the simplest mechanically.

By the way, if you must have a seal anywhere at all, use an O-ring. These are far and away the best route, besides being reasonably cheap and standard.

Let us know what you come up with here. We may work up an article or an information exchange service if this thing really takes off the way it rightly deserves to.

**How can I run Applewriter IIC on an Apple IIC?**

There is a new version of Applewriter intended specifically for the IIC. It is ProDOS-based and includes new features such as a scrollable 240-character-wide-screen (spreadsheets anyone?), settable screen margins for "what you see is what you get," display of page/line position in a document, a modemlink, and a few other goodies. Price of the new Version 2.0 is $150 from your Apple dealer. An upgrade is separately available for $50 from Applewriter Upgrade; just mail them your old AWIIe first diskette and the cover off your manual. But neither of these are needed.

The only serious problem preventing your use of Applewriter IIC on an IIC is that the status line gets trashed. This happens because older Apples had two possible codings for inverse uppercase letters. One of these is reserved by the IIC as a "mouse nest" to hold icon characters. You guessed it. Older Applewriter made the wrong choice.

Anyway, the program here is a simple patch that fixes the status line so older DOS 3.3 Applewriter IIC will run just fine on a IIC. Be sure to make this patch on only your third or higher backup copy of AWIIe. Run the patch as an Applesloth program, and it will do the rest.

A minor IIC bug still remains with the cursor temporarily changing to a flashing bell or whatever when you park on an uppercase character. This is both cute and rare, so don't sweat it.

By another one of those most astounding coincidences, the Hardware Hacker phone number is also an independent Applewriter help line. I've been working with the Gila Valley Apple Growers Association to maintain this service. They have lots of free patches, as well as a mind-blowing eight-diskette-side AWIIe toolkit package.

**Show me a small and attractive MacPaint font.**

If you haven't met MacPaint or MacDraw yet, boy are you in for a surprise. There is absolutely nothing that goes on paper involving electronics or hardware hacking that cannot be done much faster, much more simply, and far more joyously on a Macintosh.

Not that there aren't any problems, though. Their smallest type fonts are the Monaco 9-point and their Geneva 9-point. To make these look good on the screen, they are letterspaced. Letterspacing means that a small lowercase i is the same width as a capital W. Worse yet, these fonts have only one pixel between the letters. The result looks OK on the screen, but for a schematic, pc layout, engineering drawing, or logic diagram, the printed result looks both awkward and cramped.

Fortunately, there are two utilities available that make custom fonts quick and easy on the MAC. One of these is called the Font Mover, and it is provided to everyone as part of MacPAINT. The second is called the Font Editor. This one is normally sold only to system developers as a minor part of a $350 package.

The mover lets you store fonts or install them in your machine. The editor lets you change the character shapes and their spacing any way you like. Early versions of the font editor have very poor error trapping, so expect the bomb icon every now and then, and be patient.

At any rate, I've designed a few fonts of my own, and "improved" a few others. You can easily do the same, or else I'll be glad to send you all of mine on disk for $19.50, sent or VISA phoned per the box.

Included are a variant of Monaco 9 that is proportionally spaced with double-pixel spacing. I call it Thatcher 9. This is just what you need for schematics and other electronic stuff. A modified version called Thatcher 12 has enough extra vertical
This AWIIe CLARIFIER Applesloth program modifies your Applewriter Ile backup diskettes to eliminate trashing of the IIC status display line.

100 REM
110 REM ***************
120 REM "CLARIFIER FOR" *
130 REM * APPLEWRITER Ile *
140 REM * 
150 REM * VERSION 1.0 *
160 REM * 
170 REM * 
180 REM * 
190 REM * 
200 REM * COPYRIGHT 1984 BY *
210 REM * DON LANCASTER AND *
220 REM * SYNERGETICS, BOX *
230 REM * 1300 THATCHER AZ. *
240 REM * 85552. 602-428-4073 *
250 REM * 
260 REM * ALL COMMERCIAL *
270 REM * RIGHTS RESERVED *
280 REM * 
290 REM ***************

300 REM This mod changes a
310 REM backup copy of AWIIe
320 REM to eliminate trashing
330 REM of the Ile status line.

340 REM This lets you use a
350 REM DOS 3.3 version of AWIIe
360 REM on either a Ile or Ile.

370 REM 

380 TEXT : HOME : CLEAR : END

Check Validity

610 PRINT
620 PRINT " [D]BLOAD OBJ.APWRT) [E,A$2300"
630 IF PEEK (14472) < > 188 THEN 880
640 IF PEEK (14709) < > 41 THEN 880
650 IF PEEK (14753) < > 57 THEN 880
660 PRINT " [D]BLOAD OBJ.APWRT) [E,A$2300"
670 IF PEEK (14815) < > 188 THEN 880
680 IF PEEK (15052) < > 41 THEN 880
690 IF PEEK (15096) < > 59 THEN 880
700 POKE 14815,60: POKE 14816,36: POKE 14817,207:
710 POKE 14818,16: POKE 14819,2: POKE 14820,169:
720 POKE 14821,62
730 POKE 15052,208: POKE 15053,42
740 POKE 15062,96
750 IF PEEK (20365) = 176 THEN POKE 20365,182:
760 REM RECONNECT HELP SCREENS
770 POKE 15096,41: POKE 15097,127: POKE 15098,201:
780 POKE 15102,201: POKE 15103,64
790 POKE 15104,144: POKE 15105,204: POKE 15106,41:
800 POKE 15107,63: POKE 15108,176: POKE 15109,200
810 POKE 15112,96: POKE 15113,176: POKE 15114,201:
820 POKE 15115,64: POKE 15116,144: POKE 15117,204:
830 POKE 15118,96: POKE 15119,176: POKE 15120,200
840 POKE 15121,64: POKE 15122,144: POKE 15123,204:
850 POKE 15124,96: POKE 15125,176: POKE 15126,200:
860 REM RECONNECT HELP SCREENS
870 TEXT : HOME : CLEAR : END

Fixes only the status line. Rare and brief changes
in the flashing cursor symbol will remain.

In the above listing, [D] stands for "control-D". All
other brackets are real.

Only the "F" version patch would normally be used by
the IIC. We have also included an "E" patch for
possible use by gonzo hackers only.

This program is available ready-to-run on the Gila
help line's AWIIe Toolkit package.
spacing that it is ideal for vertical spacing inside an integrated circuit, for the individual pin callouts.

The boldface option on Mac ruins the Wand M characters. So, a version called Thatcher 10 is available that is bold to start with. It appears bold as normal text, and there are no problems with certain characters being blocked out. I also have a miniature 3 x 5 dot font called Thatcher 14. This one is ideal for the little numbers inside the pin circles on an integrated circuit, but is too tiny and awkward for most anything else.

Rounding out the package are some larger numerals, some fonts for printed-circuit overlays, and the usual samples and demos, and a bunch of ready-to-use 7400 series integrated-circuit schematic icons.

Since these are intended as single sized fonts, I used the trailing numerals as a variation number, rather than as a font size. This is allowed, though not standard, and lets you install all of the fonts under a single name. MacPaint or MacDraw will merrily pick the font for you, instead of the size, without knowing you just put one over on it.

If you have any favorite fonts of your own design, send them along so others can share them. There's lots of room left on the disk.

---

New Television Converter Book

CIRCUITS, THEORY, WAVEFORMS,
Subscription TV Reference Manual

This information packed book details the methods used by subscription TV companies to scramble and descramble video signals. Covers the Sinewave, Gated Pulse, SSAVI system, and the methods used by most cable companies. Includes circuit schematics, theory, waveforms and trouble shooting hints. Only $12.95 plus $2.00 first class P & H. Information $2.00, refundable. Foreign orders please remit in U.S. funds only.