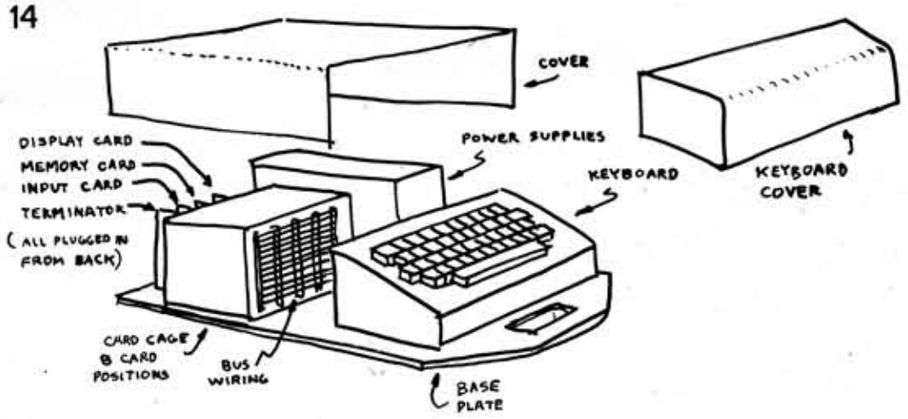


WHAT IT LOOKS LIKE

TOM SWIFT LIVES!

OUR MOTTO:

IF **WORK** IS TO BECOME PLAY THEN **TOOLS** MUST BECOME TOYS

BY LEE FELSENSTEIN, LGC Engineering

THE TOM SWIFT TERMINAL

Shortly after the Community Memory public-access information retrieval system was put into operation in Berkeley and San Francisco, it became clear that existing terminals would not be sufficient for the operation of an expanded system. Available terminals were either display or printing types, and C.M. needed a combination.

Also, existing terminals are too hard to fix. The motto of the system was "hands on," but the terminals in effect said "hands off " of the equipment. Glitches and quirks of operation were present in the cheaper display terminals which might unnerve an inexperienced user, and this system was being made for inexperienced users.

Armored terminals? Ultra-high reliability? The purpose of C.M. was not to lock people out of the system, but to give them some control over it.

The answer was to design a terminal which could be used as a toy as well as a tool; that electronic enthusiasts could learn and could hook up in various ways. A terminal like that would grow its own service organization. It could also be updated to higher levels of intelligence when Community Memory was ready to deal with smarter, editing terminals.

As the hardware arm of Community Memory, L.G.C. Engineering has developed a preliminary design for this "Tom Swift Terminal" and we are now at work on the manufacturing design. It will be part of the future Community Memory systems and will be offered for sale as well.

As a terminal, it is over-designed, but NOT over-specialized. In fact, the design effort has gone into making it as modular and adaptable as possible. In its minimum form, called the "basic system," it is a box of electronics with a keyboard. It connects to a telephone coupler and to the antenna terminals of a home TV.

You type on the keyboard and the text fills the TV screen, rolling upwards off the screen after the screen is filled. Lines are 32 characters long, and 16 lines fill a screen. After a line has rolled up off the screen, it can be retrieved by causing a "roll down," which can bring back up to 16 past lines. It operates at 30 characters per second, which is about as fast as a non-speed-reader

Now, there's nothing unusual about such operation, except perhaps the "roll down" capability. The fun begins when you want to expand the system to do more.

Open up the case. The electronics are organized on plug-in-printed-circuit boards, with room for lots more. You can plug in a card which connects to a little adding machine printer. Without changing the rest of the device, you can now command (from the keyboard or the computer) the printer to print out a line at a time or a screenful at a time. The printer takes a third of a second to print out one 32 character line.

Plug in another memory card (the system comes with one) and you will be able to retrieve up to 48 unseen lines or data. The thing is made for plug-ins. Each card plugged in has full access to all the information in the system, and can control the whole thing if properly designed.

And the terminal comes with enough information so that your local Tom Swift can start building plug-ins. If he can't quite make it, we'll be serving as a clearing house for users who can help other users. Nothing will be marked "proprietary circuitry- return to the manufacturers for service."

As the Community Memory system develops, we will be producing updates of the circuit cards which control the device's editing capability and "intelligence." But we expect to be beaten out by electronic enthusiasts who will recognize the device as a natural seed-bed for testing and developing micro-computers.

"Micros" are computers built around physically tiny but electronically powerful "microprocessor" integrated-circuit chips,similiar to those which are the guts of pocket calculators.

These chips have been industrially available for a few years now, and are not used by amateur electronics buffs because of the complex additional circuitry they require for operation. Memory, input/output, terminals,etc.

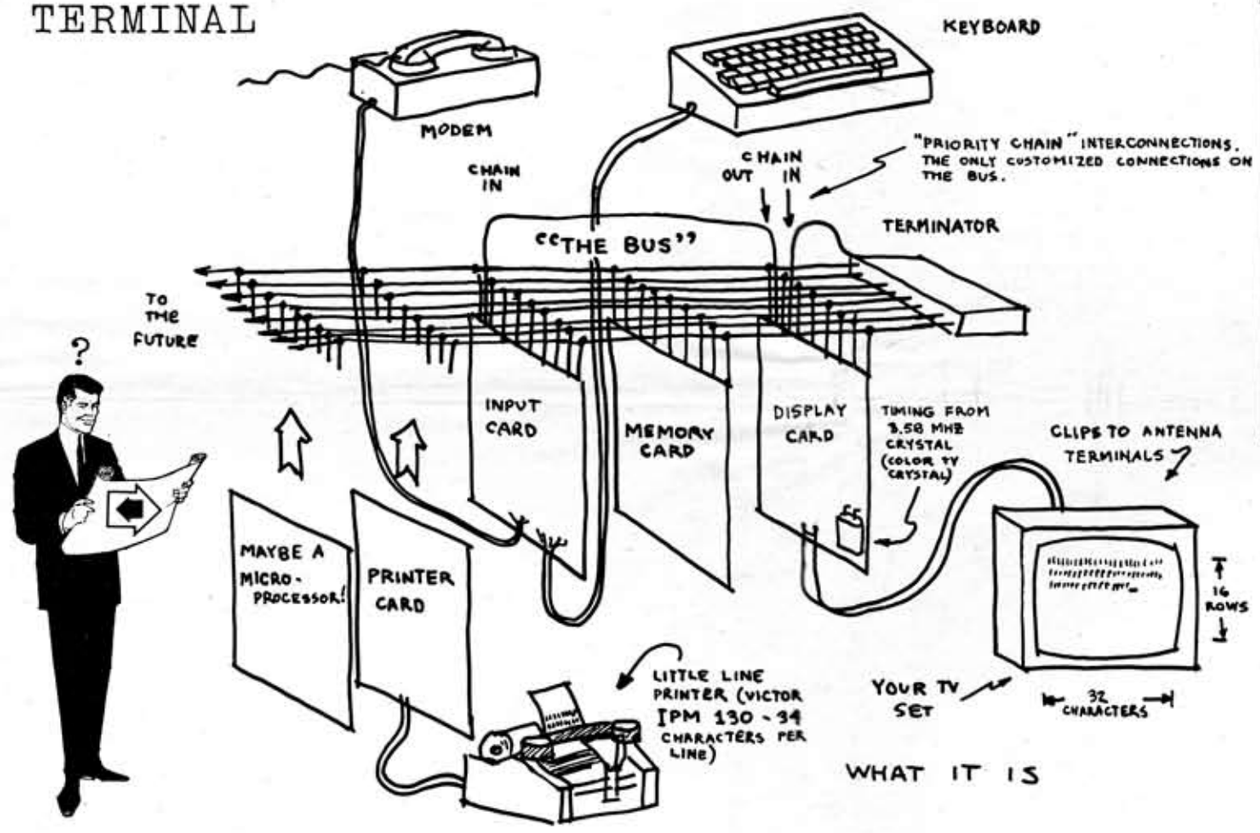
This device is specifically designed to provide such support functions for these microprocessors. Not only that, but in a home environment (TV set required) and in a relatively clean, quiet manner which makes for toys which parents like.

"Don't bother Suzie; she's playing with her computer."

"Toy" should not be seen as a put-down term. One of these devices can support a micro-computer capable of running an entire Community Memory system (of course, the disc storage units would be physically huge compared with the micro.)

"Toy" implies that the device has uses on many different levels, all of which are well under control of the user and which add to the user's general enjoyment, understanding or well-being.

For more information on this device, its design philosophy and operation, we have a 25 page discussion available for 50 cents. Write to L.G.C. Engineering, 1807 Delaware Street, Berkeley, Ca. 94703.



WHAT IT IS

convivial design

(design so that the user controls the tool, and not the reverse)

One of my engineering professors once asked what was the difference between engineers and scientists. No one answered. The prof drew a dollar sign on the blackboard. "It'll make you or break you," he said reverently.

It's becoming obvious these days that the dollar sign isn't quite where it's at. You won't be able to buy a new world when this one is used up. And design in pursuit of the dollar is busy chewing up not only our physical world, but also our ways of working with our tools and with each other.

I learned electronics as a kid by messing around with old radios. Vacuum-tube radios, which are easy to tamper with because they're designed to be fixed. I made radios into intercom amplifiers, oscillators and transmitters before I knew how to design anything. I stripped radios down to the bare metal, sorted and tested the parts, and built new things from these parts.

Then transistor radios took over the field. They were cheaper to produce, smaller, portable, and didn't need repair as often. So in the interest of the dollar sign, they were made so that you couldn't understand what was in them, couldn't do anything about it if you did understand it (the printed circuit boards would come to pieces if you melted the solder,) and couldn't use parts from one radio in other circuits.

Kids were walking around with transistor radios all the time but with no reason or opportunity to pry into them and learn what made them work. Now the first wave of these kids are grown up and the electronics industry has a shortage of technicians.

Progress? For who? Even the industry suffers. And they have the gall to moan about how people don't appreciate what they're doing for them! But they won't do anything differently , because that is the industrial way of doing things.

Well, there are different ways of doing things, even for engineers. Before there was an industrial system, people were building tools that other people could use without much training. Tools that people could use and which would not use them. People could understand how the tools worked, how to fix them when they broke, and how to alter them when the job changed.

There's no reason why even the most complex tools in use today can't be handled the same way. P.C.C. is showing how computer software can handled in this convivial fashion.

(I will use the term "convivial" to refer to this "non-industrial" type of design for tools and systems. The term is from the book," Tools For Conviviality" by Ivan Illich, Harper and Row, 1971, which first laid out this approach to the problem.)

And as a computer hardware designer, I believe that computer hardware can also be designed and handled in a convivial fashion. My own effort is the Tom Swift Terminal, described elsewhere. But aside from plugging my own products, I want to use this space to open up communication among those of us who are working on convivial design for whatever application.

Letters, project descriptions, intelligent suggestions, requests for help and offers of help can be sent to P.C.C. and I'll try to keep them straight. We'll see if we can use this page as an information exchange until the need outgrows it.

Lee Felsenstein

COMMERCIAL

LGC Engineering is offering low-speed modem circuit cards compatable with Bell 103 standards. They are intended for full or half duplex operation at 75 to 300 Baud. They use phase-locked loop frequency-shift detectors with a unique self-adjusting reference circuit which is constantly re-calibrating itself.

Minimum input level is -46 dbm, input impedance is 33kohm. Output level into 900 ohms is adjustable from +6 dbm to -20 dbm. An electronic hybrid circuit is included on the card so that the Bell CBT coupler or equivalent can be driven without additional components.

Power required is +18 volts, −18 volts unregulated, 2vpp maximum ripple. A suggested power supply circuit is provided. EIA RS−232 data input and output is included, as well as 20 ma current loop input and output. EIA carrier detect output and drive for two parallel back-to-back LED indicators is also provided.

An optional auto-answer circuit for use with CBT couplers is available. The modem is constructed on a 4.5 X 6.5 inch printed circuit card and connects to a 44-pin edge connector, Vector R644 or equivalent.

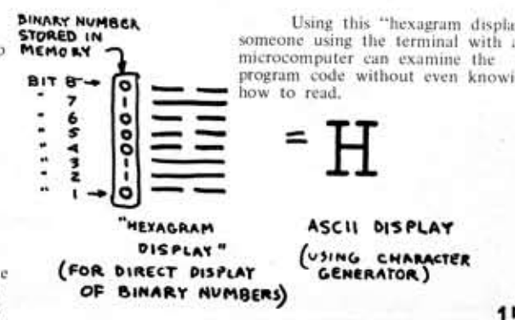
Price is $150 plain, $175 with auto-answer. Specify originate or answer mode. Further information available from LGC Engineering, 1807 Delaware St., Berkeley, CA 94703

We're also designing a microprogrammed multi-line serial-line multiplexer (for handling up to 256 teletype lines). If you need one and have some money, let us know.

also;

The Tom Swift Terminal is able to display the direct binary contents of its memory as well as the letters and numbers represented by those binary numbers.

For example, the letter "H" is represented by the binary 01000110. If a switch is thrown the terminal displays the number as a stack of solid or broken lines in the space where the "H" had been displayed. Solid lines represent "1", broken lines represent "0". The bottom line on the stack is the right-hand digit of the binary number, the least significant digit.



Using this "hexagram display", someone using the terminal with a microcomputer can examine the program code without even knowing how to read.