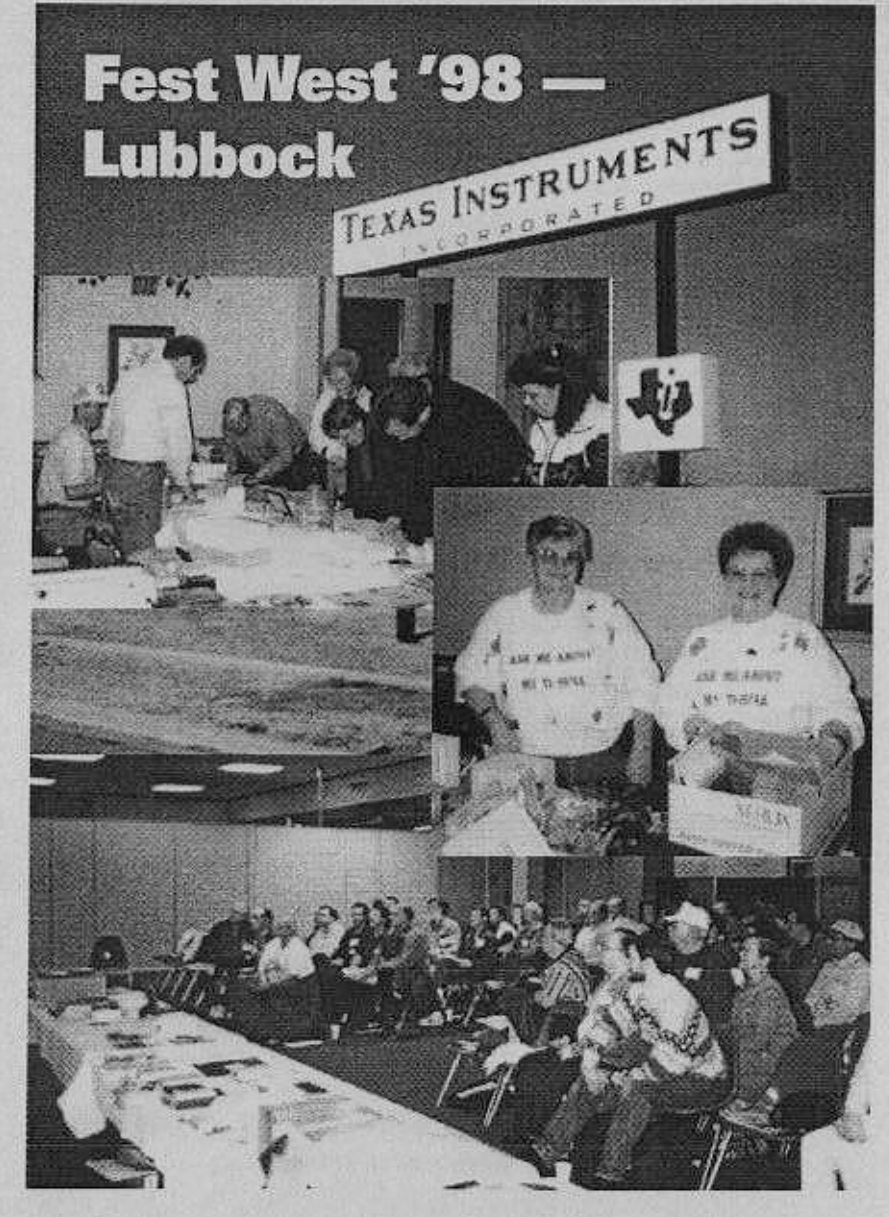
Covering the TI99/4A and Geneve home computers

MICROpendium

Volume 15 Number 2 March/April 1998 $6



CONTENTS

MICROpendium

MICROpendium (ISSN 10432299) is published bimonthly for $35 per year by Burns-Koloen Communications Inc., 502 Windsor Rd., Round Rock, TX 78664-7639. Periodical postage paid at Round Rock, Texas. POSTMASTER: Send address changes to MICROpendium, P.O. Box 1343, Round Rock, TX 78680-1343.

All correspondence should be mailed to MICROpendium at P.O. Box 1343, Round Rock, TX 78680.

Foreign subscriptions are $42.50 (Canada and Mexico); $40 surface mail to other countries; $52 airmail to other countries.

All editions of MICROpendium are mailed from the Round Rock (Texas) Post Office.

Mailing address: P.O. Box 1343, Round Rock, TX 78680.

Telephone & FAX: (512) 255-1512

Internet E-mail: jkoloen@earthlink.net

Home page: http://www.earthlink.net/~jkoloen/

John Koloen — Publisher

Laura Burns — Editor

COMMENTS

Fest West hits the spot

Laura and I had a great time at Fest West '98 in Lubbock, hometown of Buddy Holly and birthplace of the TI99 and TI99/4A. The SouthWest 99ers and other TIers who helped put this event are to be congratulated for a job very well done. It's a conference that will stand out for a long time.

Until I spoke to Tom Wills that Saturday morning I didn't know that Terrie Masters originally proposed Lubbock as a site for Fest West, nor that it took some four years to get from idea to reality. And it wasn't as if everything went smoothly. Many members of the Southwest 99ers, which sponsored Fest West, couldn't come because of illness. By mid-afternoon Saturday Wills was working on automatic pilot, having gotten little rest for days. Yet it all came together.

Listening to Lee Kitchens speak about the development and manufacturing of TI's home computer was informative and entertaining. We recorded most of what he had to say and you'll find it starting on page 10. What I found most interesting was his comments on how TI had managed to produced such a good manual about BASIC programming. It was the 8½ x 11, three-hole drilled, loose-leaf book with the green cover. It turned out a college student working at a TI retail outlet (remember when TI had outlets!) unloaded on him one day about how many errors he'd found in TI documentation. Kitchens gave the kid a prototype of a calculator with its beta-tested manual and let the student have at it. You can read how it turned out, but what's unusual to me is that here is a multi-billion dollar company that supposedly could afford to pay for the best talent and what does it do? It takes advantage of serendipity. It operates like a little mom and pop business that relies on the kindness of strangers. It definitely put a more human face on an organization that at times looks as formidable as a medieval fortress.

And then there was the chance to see a lot of people, attach faces to names that we've written about for years. Tom Wills was one. Bill Gaskill was another. During one of his presentations, Bill noted that after TI announced it was going to quit the home computer business and slashed prices, the company was selling so many computers and modules that it was making a profit. It gave some within the company second thoughts about quitting the business, but true to the company's reputation for ethical behavior and the fact that the Securities and Exchange Commission was expecting it, the company closed down operations on schedule, despite the fact that there was still money to be made. There were others. And, of course, visiting with others that we haven't seen for a year or two, such as Terrie Masters, Hal Shanafield, Gary Cox and Mike Wright.

As with many TI fairs, there were representatives from overseas, including England, Holland, and Germany. It never ceases to amaze me how widespread

Continued on page 4

COMMENTS

Continued from page 3

was the influence of the TI home computer, especially when one considers how short-lived it really was.

It's not likely that any future TI fair will be able to top Fest West '98. I'm not saying it can't happen, I'm just saying that the event in Lubbock will be hard to beat.

BACK ON SCHEDULE

I think we're back on schedule. We replaced our laser printer with a new unit that's working fine. This edition is back to 56 pages, including a good dozen devoted to Fest West '98. You'll notice some unevenness in the appearance of some pages. That's because we're printing some pages in the "econo mode." This makes the toner last longer, which helps us to keep the cost down. Most of the pages in last month's edition were printed in this mode. That's because we didn't have enough toner on hand to do the job any other way. We did learn that econo mode doesn't work well on photos or many other graphics. So this time we're reserving econo mode for text-only pages to see how that works out.

By the way, if you know of a place that sells toner for a Hewlett-Packard 6MP (same as the 5MP), please let us know. We're currently paying $42 per cartridge, including shipping.

—JK

FEEDBACK

Thanks to everyone for Fest West '98

I'd like to personally thank everyone who made Fest West '98 - Lubbock a reality. I know I've been given the credit for getting it accomplished, but there are lots of others who also helped.

Without the complete backing of my User Group, the SouthWest Ninety Niners, this whole project never would have gotten off the group. They supported me completely throughout the inception, planning, and actual Fest West '98 itself.

Terrie Masters deserves credit as it was her idea to hold such a Fest West. With the advent of the Internet and the TI99 E-Mail List Server, I was able to "take the ball and run with it." If Terrie hadn't been so persuasive in her feelings that this could be done, it might never have happened.

Then there was Zachary Douglas who at the time lived in Lubbock. He was able to find the name of the person for me to contact at Texas Instruments' Lubbock facility. Without that vital contact, Fest West '98 would never have seen the light of day.

The contact person at that time was Rodney Cates. His enthusiasm in convincing Texas Instruments this was a good idea was a crucial step in this whole process. When Rodney turned the project over to Gabriel Flores, Gab-

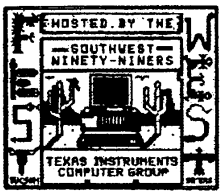
Continued on page 6

THANK YOU

To The TI Community

From the SouthWest Ninety Niners User Group and the Fest West '98 Committee!

We'd like to thank everyone involved for helping make **Fest West '98 - Lubbock** the success that it was. Without you, it wouldn't have been as great.



We'd especially like to thank the keynote speakers, ***Lee Kitchens*** and ***Bill Gaskill***, for their great talks. We'd also like to thank Texas Instruments for the use of their fine facility. Without the cooperation of Texas Instruments, **Fest West '98 - Lubbock** never would have happened. We'd also like to thank the Sheraton Four Points Hotel and the Koko Inn, the Convention and Tourism Bureau, and the entire City of Lubbock, Texas for making us feel so welcome.

But most of all, we'd like to thank everyone who came to Lubbock to participate in Fest West '98 - Lubbock. You made all of our efforts worthwhile.

TIers again proved that they are the finest group of people on the planet. Thank you to every one of you for making Fest West '98 - Lubbock a true ***Once In A Lifetime Experience!!!***

FEEDBACK

Continued from page 5

riel continued on with the same enthusiasm as Rodney. Thanks to Gabriel, we were able to secure Lee Kitchens as one of our keynote speakers.

Several SouthWest Ninety Niners User Group members were also willing to put their money when their mouths were. In August, Mike Doane and Ed McCullough accompanied me to Lubbock to meet with hotel officials and TI officials. Mike Doane, Rod Stallard, Al Armstrong and I all went to Lubbock on February 11 to finalize the preparations for Fest West '98.

All the aforementioned people were instrumental in making Fest West '98 - Lubbock a reality. They all deserve a vote of thanks for what they did over the past year. And they have all earned my deep appreciation for assisting me in this Once In A Lifetime Experience!

Also deserving of a big "Thank You" were the people who donated money to the Transportation Fund to get Bill Gaskill, our second keynote speaker, to Lubbock. Bill was short on money due to taking a retirement from his job when Grand Junction did cutbacks to its municipal services.

Last, but not least, there were those who attended Fest West '98 - Lubbock. These were the people who truly made it a success.

To everyone listed above, I wish to personally extend a personal "Thank You!" Without each and every one of you, Fest West '98 - Lubbock would never have happened.

TOM WILLS
TUSCON, ARIZONA

Storing magazines

I'm sure that most people, as I did, used a three-hole punch to prepare their copies of MICROpendium for storage into loose-leaf binders.

The new format created a problem for a few minutes. If you carefully remove the outer cover, slip it into a plastic sheet protector, line it back up on the inner pages and apply new staples next to the old ones, you can then slip it into the same binder with your old format magazine. Save a few pennies and use the economy-grade sheet protectors; they are sufficient for the task.

MATT MATTHEWS
SOUTHWEST NINETY-NINERS
TUCSON, ARIZONA

JIM PETERSON ACHIEVEMENT AWARD

1998 votes due by May 1

Voting for the 1998 Jim Peterson Achievement Award will continue until midnight May 1, according to award organizer Jim Krych. Voting may be done by mail or email at the following address: Jim Krych; Jim Peterson Award; 3969 Clague Rd.; North Olmsted, Ohio 44070-2306; email: ab453@cleveland.freenet.edu.

Here are the categories and nominees. You may vote for one entry per category:

PETERSON AWARD

Peterson Achievement Award — Hardware

Don Walden — Cecure Electronics
Michael Becker — various peripherals and the new TI clone
Turtle Enterprises — The VideoTurtle (TV RGB to S-Video)
Western Horizon Technologies — SCSI card

Peterson Achievement Award — Software

Mark Van Coppenolle and Mike Wright — PC99
Tim Tesch — software for Geneve
Rich Gilbertson — Rich XB w/AMS support
Barry Boone
Leonard Taffs
Bruce Harrison — newest AMS software including Midi-Master

Peterson Achievement Award — Community

Mike Wright — PC99
Bill Gaskill — TI History
Jerry Coffey
Jim Krych — founder of Jim Peterson Award
Charles Good — MUG fair, Lima User Group
Bruce Harrison — TI software and assembler articles
Don O'Neil — Western Horizon Technologies products
Rich Polivka — TI99/4A Web page
Bud Mills — Horizon products
Tom Wills — TI server

Peterson Achievement Award — Geneve

Tim Tesch — Geneve O/S, software
Jim Uzzell — Myarc BASIC programming
James Schroeder — various Geneve programs

FEST WEST '98 - LUBBOCK

TIers gather for once-in-a-lifetime event at birthplace of TI99/4A

BY LAURA BURNS

From across the United States and from Europe, more than 50 individuals — TI users and former users — gathered in Lubbock, Texas, on Valentine's weekend for the 1998 Fest West, hosted by the SouthWest Ninety-Niners of Tucson, Arizona.

Opening session was held at the Lubbock Texas Instruments facility, formerly the home of TI99/4A manufacturing and now a plant that makes wafers for semiconductors, facilitated by Rodney Cates and Gabriel Flores of TI. After a talk by Lee Kitchens, the engineer who headed the team that developed the 4A (see related story), and a brunch supplied by TI, attendees got the opportunity to take guided tours of the plant, which in its heyday was the workplace for 4,500 TI employees and now employs 750.

Bill Gaskill, the unofficial "historian" of the 4A began his presentation at TI's site and continued it at the Lubbock Sheraton Four Points, the Fest West headquarters. His "Trivia Quiz" contained many questions that stumped 100 percent of those listening. At numerous points in his talk, Gaskill praised the high ethical standards of Texas Instruments and the after-market support they have provided to users — "unheard of in corporate America."

Gaskill also brought for viewing

TI99/4A chronicler Bill Gaskill talks TI history at TI's Lubbock plant.

FEST WEST '98 - LUBBOCK

by the crowd numerous rare documents and photographs having to do with the history of the TI99/4A.

Other presentations were by Mike Wright and Mark van Coppenolle of CaDD Electronics and by Michael Becker and Gerd Weissmann. (See related articles and photos.)

European visitors who received T-shirts at the opening ceremonies were Becker and Weissmann, of Germany; Berry Harmsen, the Netherlands; and John Murphy, England.

The big raffle winner was David Mischler of Ogden, Utah, whose name was drawn three times. Other winners were Mac Swope of the Mid-South Users Group in Memphis, Tennessee; Mary Leard of the Dallas TI Users Group; and Harrison Hoffmann of the Cleveland TI Users Group.

Wills expressed appreciation to Terrie Masters, former president of the LA 99ers in Los Angeles, California, who had the original idea of holding a fair in Lubbock.

The SW 99ers maintained a hospitality suite throughout the gathering. A brunch Sunday morning ended the weekend's events.

The *Lubbock Avalanche-Journal* had devoted a feature to the gathering earlier in the week, which included interviews with Kitchens, Flores and Wills. In it, Wills explains that 99/4A users "think of (their computer) as a classic computer, kind of like a classic car." He remarks on the expansion of the computer after it stopped being produced and its phenomenal mathematical accuracy.

Fest visitors help injured motorist on return trip

This item was sent via the Internet by Tom Wills:

While returning home from Fest West '98—Lubbock on Sunday evening, February 15, Jack and BJ Mathis and Mike and Joan Doane witnessed an accident near the New Mexico/Arizona border. As it was late, this stretch of I-10 was almost devoid of traffic. They were the only other cars on the road at that time of night.

An eastbound car somehow went out of control and rolled over several times. The Mathises and the Doanes stopped and rendered aid to a man who appeared to have just minor injuries, but who appeared to be going into shock.

Thanks to their swift actions, the man was made to be comfortable and warm so as to prevent shock from setting in. If the Mathises and the Doanes had not stopped to render aid, the man could have possibly died as it was very cold and he was going into shock.

Jack and BJ Mathis and Mike and Joan Doane are to be commended for doing the right thing. Many people would have kept going. Kudos for helping when help was needed.

FEST WEST '98 - LUBBOCK

TIers gather for once-in-a-lifetime event at birthplace of TI99/4A

BY LAURA BURNS

From across the United States and from Europe, more than 50 individuals — TI users and former users — gathered in Lubbock, Texas, on Valentine's weekend for the 1998 Fest West, hosted by the SouthWest Ninety-Niners of Tucson, Arizona.

Opening session was held at the Lubbock Texas Instruments facility, formerly the home of TI99/4A manufacturing and now a plant that makes wafers for semiconductors, facilitated by Rodney Cates and Gabriel Flores of TI. After a talk by Lee Kitchens, the engineer who headed the team that developed the 4A (see related story), and a brunch supplied by TI, attendees got the opportunity to take guided tours of the plant, which in its heyday was the workplace for 4,500 TI employees and now employs 750.

Bill Gaskill, the unofficial "historian" of the 4A began his presentation at TI's site and continued it at the Lubbock Sheraton Four Points, the Fest West headquarters. His "Trivia Quiz" contained many questions that stumped 100 percent of those listening. At numerous points in his talk, Gaskill praised the high ethical standards of Texas Instruments and the after-market support they have provided to users — "unheard of in corporate America."

Gaskill also brought for viewing

TI99/4A chronicler Bill Gaskill talks TI history at TI's Lubbock plant.

FEST WEST '98 - LUBBOCK

by the crowd numerous rare documents and photographs having to do with the history of the TI99/4A.

Other presentations were by Mike Wright and Mark van Coppenolle of CaDD Electronics and by Michael Becker and Gerd Weissmann. (See related articles and photos.)

European visitors who received T-shirts at the opening ceremonies were Becker and Weissmann, of Germany; Berry Harmsen, the Netherlands; and John Murphy, England.

The big raffle winner was David Mischler of Ogden, Utah, whose name was drawn three times. Other winners were Mac Swope of the Mid-South Users Group in Memphis, Tennessee; Mary Leard of the Dallas TI Users Group; and Harrison Hoffmann of the Cleveland TI Users Group.

Wills expressed appreciation to Terrie Masters, former president of the LA 99ers in Los Angeles, California, who had the original idea of holding a fair in Lubbock.

The SW 99ers maintained a hospitality suite throughout the gathering. A brunch Sunday morning ended the weekend's events.

The *Lubbock Avalanche-Journal* had devoted a feature to the gathering earlier in the week, which included interviews with Kitchens, Flores and Wills. In it, Wills explains that 99/4A users "think of (their computer) as a classic computer, kind of like a classic car." He remarks on the expansion of the computer after it stopped being produced and its phenomenal mathematical accuracy.

Fest visitors help injured motorist on return trip

This item was sent via the Internet by Tom Wills:

While returning home from Fest West '98—Lubbock on Sunday evening, February 15, Jack and BJ Mathis and Mike and Joan Doane witnessed an accident near the New Mexico/Arizona border. As it was late, this stretch of I-10 was almost devoid of traffic. They were the only other cars on the road at that time of night.

An eastbound car somehow went out of control and rolled over several times. The Mathises and the Doanes stopped and rendered aid to a man who appeared to have just minor injuries, but who appeared to be going into shock.

Thanks to their swift actions, the man was made to be comfortable and warm so as to prevent shock from setting in. If the Mathises and the Doanes had not stopped to render aid, the man could have possibly died as it was very cold and he was going into shock.

Jack and BJ Mathis and Mike and Joan Doane are to be commended for doing the right thing. Many people would have kept going. Kudos for helping when help was needed.

FEST WEST '98 - LUBBOCK

Manufacturing manager Lee Kitchens tells tale of 99/4A production

Lee Kitchens was the manufacturing engineering manager from 1978 to 1983 for TI's Consumer Products Division, which manufactured the TI99/4A. He had also been part of the Texas Instruments team that manufactured the world's first transistor radio. The following are comments he made during Fest West '98 in Lubbock, Texas on Feb. 14. Photos were taken by Gary Cox.

It's fascinating to have your past brought before you again. Those were interesting days.

I came to Lubbock in 1978 to have a hand in the manufacturing of the home computer. I'd built an awful lot of calculators, had a background in microprocessors and in the past most of the manufacturing engineering managers did not have the background in microprocessors that I had and they felt that was important.

When I first came out (to Lubbock) the home computer was still in the simulation mode. What we did was to build a simulator out of discrete devices with each part of the computer simulated with TTL devices. And the fellows were struggling to be able to build something that would act like the home computer without the benefit of the microprocessor itself or all the memory or all the interconnects. But in those days we built a simulator in order to get the hardware right and get the software right. Once all of that simulating was done where we could actually operate keyboards, watch a display, and try out the various features then it went to the system design itself where a printed circuit board was laid out, the individual chips like the 9900 microprocessor, the 9918, the video driver, and all of the interface circuits necessary to create a real working device.

Parallel to that was the design of the case, the top and the bottom case, the keyboard. Each of those things was challenging in itself.

So there were five extensive teams involved. There was circuit design, algorithm design, the mechanical design and, while all of that was going on, my group was trying to figure out how to build it.

FEST WEST '98 - LUBBOCK

There are 544 parts in the 99/4A, and each had its own function. Each had its own cost, which was always an issue. Cost obviously affected selling price. The number of components affected the various steps in manufacturing.

So there were parallel efforts going on all across the board. Keeping all of these efforts communicating with each other was essentially my job and the job of my group to make sure that we had a predictable item.

Started out as video game

What many of you may not know is that in the simulation stage the design objective was to build a video game. That's what the unit started out to be. Marketing input decided that wasn't going to be sufficient. Part way through design and development the strategy was changed and the computer as we understand it today (was developed).

The 99/4 was built upside down, if you will. Everything was built into the top case and it had a limited keyboard. It only would do uppercase characters. It had a lot of limitations and all of us were learning an awful lot as we went along. In manufacturing when you build a unit upside down, this meant that the top case was straight down on the production line and subject to damage, scratching and things of that sort.

After that production got started we wound up with a new vice president who was real sharp. He convinced management that the 4A was appropriate and we should enhance the keyboard and give those of us in manufacturing engineering a voice in how the unit was put together. So there was some redesign of the case, redesign of the keyboard and an ongoing tweaking of the circuitry in order to make it more producible and more reliable.

What many of you may not know is that in the simulation stage the design objective was to build a video game. That's what the unit started out to be.

FCC rules

One of the big issues in those days was part 15 of the FCC rules and regulations that stipulated that a computer wouldn't interfere with other consumer electronics. Most computers that were being built that were available were considered a business device and not a home device and, therefore, the requirements were not as stringent. But if you're going to take one of these computers into the home, the FCC said you cannot interfere with a TV or anything else. So a number of the components in the 99/4A are there to

Continued on page 12

memory or all the interconnects. But

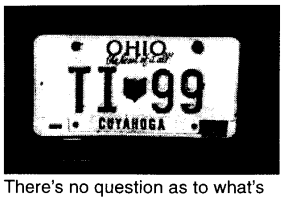
FEST WEST '98 - LUBBOCK

Continued from page 11

restrict and prohibit spurious radiation from getting into other home electronic devices.

Those of you who are familiar with the circuitry will notice a lot of ferrite beads on all of the connections to the outside world. TI had always been an extremely ethical company. That said you go buy the rules, you follow the law, you don't cheat, you don't push the edge of the envelope as far as ethics are concerned. To that extent we made very certain that our devices did not interfere with other electronic devices and that we met all of the rules and regulations.

That put us behind the 8 ball so to speak because our competitors didn't always follow the rules. We had to submit a computer to the FCC for evaluation and they would measure the radiation interference and things of that sort. Well, all of the other companies did, too, except we suspected that they may have been submitting a custom device to the FCC. We called them a "queen." Our positions were reinforced when we would go out on the open market and buy our competitors' devices and find that they didn't meet all the requirements.



There's no question as to what's important to Harrison Hoffman.
Photo courtesy Harrison Hoffman

Lee Kitchens

Another problem we had was that submission to the FCC took a long time. So we built an antenna range, a place where we could test the device ourselves. This was an octagon-shaped building built out of wood and Fiberglas. It had no metal in it. No nails. No support structure that was magnetic in any way where we could test our devices the same way the FCC did. We finally got certification of our antenna range so we could do all our own testing without having to go to the FCC.

The second area was Underwriters approval. Anything you plug into the 110 has to be approved by Underwriters Laboratories. Approval by Underwriters wasn't a law from a national standpoint, except that many states and many cities had ordinances that you couldn't sell devices, lamps, or anything that plugs into the wall without being UL approved.

There again it took a while to get these approvals, so we developed our own Underwriters certified laboratory to be able to test not only computers, but calculators and anything else so that we could certify the equipment. We also had to meet Canadian Standards Association, the CSA requirements, and for some calculators we had to meet the requirements of some of the European agencies. All of those are part of the activities necessary to develop any product.

FEST WEST '98 - LUBBOCK

NEATEST THING SINCE SLICED BREAD

I have two or three 99/4As myself. All of mine are preproduction models. Now this was the neatest thing since sliced bread. I had a problem with my engineers and my technicians spending too much time on the keyboard. We also had a problem, many of our fellows had a problem in that we working a lot of hours. A lot of long days, a lot of weekends and it created problems on the home front — wives and children wondering why daddy is never around. What's dad doing? Why does he have to spend all his time out there?

So I made it available to all of my people to have a 99/4A at home so that the families could share in this tremendous program that we had going, and so that the guys would spend their keyboard time on their own time and not company time.

When you develop a product you've got to make sure it is rugged and reliable — functionally, electronically and mechanically. So part of the scenario is that after you've built the first four or five by hand you do a pre-production run where you try out all your tooling on a production line. That's where you find all of your mistakes. That's where the rubber meets the road.

> **Cost was always an issue, and quality and reliability were always an issue and a lot of effort was spent making sure that all of the incoming material met our requirements.**

The output of that pre-production run is not the kind of thing you want to have in the hands of the general public. Those also were the initial units we used for environmental testing.

SHAKE RATTLE AND ROLL

Environmental testing, we used to call shake rattle and roll. This is where you find out what happens if you drop it off the table. Or if it's in the shipping box and UPS heaves it into the back of the truck. We wanted the thing to work when it got to the end customer. So the early test units went through all of that. Obviously you don't sell that to the customer.

The first production run of about 200 devices all went to quality assurance, and they would do their

Continued on page 14

Lee Kitchens

There's no question as to what's important to Harrison Hoffman.
Photo courtesy Harrison Hoffman

Continued from page 13

best to destroy those units. The survivors of that quality assurance test were used for marketing demonstrators and those were the units I sent home for the guys to play with at home.

The other thing that many of our people had to learn was BASIC programming, and I had to learn it too. My experience with computers dated back to an old IBM 1620 that you fed with punched cards and programmed in FORTRAN. It was not the greatest thing in the world. I had a unit at home so I could learn how to program in BASIC on my own time, not on company time. So we used those devices, called Class E devices, to educate our own people and to maximize our efficiency here.

My group was organized by product. There was a master technician or engineer in charge of each of the peripheral devices. His responsibility was seeing that his product got down the line, met all the tests, and work out the problems that you get into when you have that many activities going on. Cost was always an issue, and quality and reliability were always an issue and a lot of effort was spent making sure that all of the incoming material met our requirements. A good example is the monitor.

The monitor

Now, we did not build monitors. That was not our forte and people had been building TV sets forever and we naturally assumed we could go to a TV vendor and have them build a monitor to our specs and they would ship it to us and we would add a computer to it and ship it on out.

Well, the very first ones that came in we sent through our quality assurance testing. And five percent of all the monitors would fail. Either they were dead when they came in or when they went through our environmental tests they failed. And going back to the vendor we'd say what's going on. A five percent failure rate is totally unacceptable. We want zero failures.

"Oh well"' the vendor said, "what you do is if you get a dead one your dealer will take care of it. That's the way TV sets are made. They all get shipped to the local dealer and the dealer checks them out and if there's a problem they fix it and send us the bill."

Well, the home computer wasn't being sold through a dealer like TV dealers. They were being sold by Best Buy and Sears, and anywhere else and there was no infrastructure to fix things. So the monitor had to be redesigned. Its circuit design had to be enhanced, its mechanical design had to be enhanced so that, if the shipping companies dropped the unit off the shipping dock in the box, it had to survive.

Those monitors are super. I've got a couple of them — they run, and they run, and they run forever. I still see them every now and then in flea markets. I use those monitors for other things and grab them up whenever I can.

Peripherals developed

Well, as you go along developing products computers were coming along. I'd already built one myself, from a kit. We could see that there were other things that a home computer could do. And so the development of the peripherals were an ongoing thing. Initially, obviously, the computer and plug-ins were the thing but there was a need for the peripherals. The first printer was a 32-column printer that used the thermal print head. TI had a long history of thermal print heads. We had been building terminals since the '60s that used thermal print heads so this was something we had experience with and we could build a pretty good printer to go along with the home computer. But more people wanted a full 80 columns, so the RS-232 expansion unit with printer (was developed) which we sold under our name but it was built by other people. The printer had to go through the same environmental tests that the monitor did and we had some improvements made to that.

So it was a long and fascinating story. This building was literally full, not only with home computer production but calculator production, too. We were building calculators at the same time. There was a tremendous flow of materials and tremendous activity. It took a lot of support people. There were times when the hallway out here was full of materials coming to the production line and finished goods leaving the production line, 24 hours a day. And sometimes, just before Christmas, seven days a week.

Following his presentation, Kitchen's fielded questions from the audience.

How long has the building been here?

The Lubbock facility was started when I was overseas, in '72 or '73 I think. My wife is from here and my father-in-law lived here. If you lived in Dallas, Lubbock was considered if not the end of the world you could see it from here. We had one helluva time getting people to move here. Many people were dragged out here screaming and crying and carrying on. But once they got out here and got accustomed to our laid back environment, low cost of living and wide open spaces, when the opportunity came to move back to Dallas they raised more hell about that. There were a number of people when given the option to move back to Dallas, or Austin, or Houston just refused to go and left the company and went into business for themselves out here.

What was your feeling about termination of production of the 99/4A; and what did you think about all the third party product production that went on?

I helped start it and I helped turn out the lights. It was a tough thing to take. I was one of the last ones to leave.

Continued on page 16

FEST WEST '98 - LUBBOCK

Continued from page 15

The third party activity, I wasn't close to it. But when it did start it was kind of a vindication that we had a neat machine here and we'd done a good job in the quality of the device that had capability that we didn't know of — a lot of undocumented features that I'm not sure even the systems guys didn't know about.

But that's been true of almost any product. You never have enough time in the development of a product to do everything to it you'd like to do because the bottom line market - driven demands, financial demands, are such that you've got to get the product out and start the revenue streams coming in. That was always the pressure even though we had our quality requirements. We never did bend. There was incredible pressure to get product completed and out the back door. So yeah, we had a neat thing but it hurt to have to shut it down.

What is the relationship between the TMS9900 and the 99/4A? Was it developed for the computer?

No. The 99/4, there again you have a case of the pressure to get a product out and it came out .... The 4A was a better product. It took some internal marketing and argument. The 99/4A, that whole product, brought visibility to our activity clear to the board of directors. It was not uncommon to see board members out here, the chairman of the board and the president snooping around the hallways. But that was the nature of our manufacturing even in earlier days. I would find the chairman of the board up on the manufacturing floor in Dallas at eight or nine o'clock at night. Our whole structure was hands on, firsthand kind of management. Most of the managers are all engineers and they had a hard time keeping their hands off of things. Micromanagement was rampant. It was a strange environment. It was a tough environment for some people. I thrived on it.

When you were designing the Peripheral Expansion Box, was there any concern about the weight?

Part of that was that it had to be able to survive shipment. You could have built it differently, but there was a lot of pressure to get that thing yet it had to survive the shipping rigors. If you go to the big retailers today, such as Circuit City, and watch the material is handled — it gets mishandled. I've seen skids with rectangular holes in the box where a forklift had gone through it. The shipping environment for products is unreal. The mechanical guys made sure it would survive. Weight was not a real big issue.

Why didn't TI sell the Omni 800 printer with the 4A?

The Omni 800 was a commercial machine. It was designed for a business environment, and the cost of that device was not compatible with a home computer.

What if any relationship is there between the 4A and TI 990 business computers?

FEST WEST '98 - LUBBOCK

Few who attended Fest West missed Lee Kitchen's presentation.

The 9900 microprocessor was a good platform to do a lot of things with and the variations on that theme were used all over the country. We had those kinds of computers on the factory floors to operate some of our machinery, to keep data. In later years, when I was in Houston working on another project, I used those same kind of commercial computers in my production environment. It was a really good platform and you could do a lot of things with it. There was even one version where there was a complete computer on one printed circuit board and you used it to drive production equipment. It was a great platform and we got a lot of experience with the 9900 by virtue of what we did with the home computer.

What did you do to ensure the quality of the 99/4A?

The first 100 machines that came back in-warranty we did a detailed analysis of why it failed and there had to be a cause assigned to every machine. By going through that detailed analysis we were able to find the weak links in the design, the weak links in our production techniques, the unexpected things that can happen either due to design or customer misuse, abuse or whatever you want to call it because the customer is the one who will identify all of your weak links.

Now, we did our best to eliminate defects before they ever went out. Every computer went through a 96-hour burn-in. We had massive bookcases on wheels where we would line up the units, elevate the temperature to 40 degrees centigrade and power them on and off under elevated temperature and maximum voltage. That would kill a product quicker than anything. And we would do that with all the preproduction

Continued on page 18

Continued from page 17

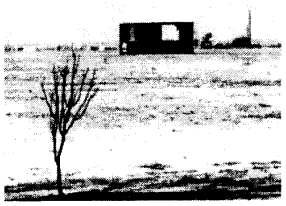
stuff to find the weak links. Still, the public will find the things you don't find, which is why we evaluated those first 100 returned units. Then after that quality assurance kept track of all field failures, both in-warranty and out of warranty so that we could make corrections to the product.

I'm sure some of you have opened your units and seen the little blue wires on some of them. I don't remember how many revisions to the main board there were. I vaguely remember revision F. But when you do a complex board layout, even though you evaluate and measure and test and everything else, and you're pressured to get products out the door, you find sometimes an etch line was left out or something was in the wrong place. In order to meet the volume requirements and the lead times we may find an error in a board and there will be 10,000 boards already in the pipeline. The blue wires are things you did to make production. The blue lines had to go through the environmental evaluation and not come loose. But the very last run I'm sure didn't have any blue wires because as soon as we found the necessity for a patch we changed the artwork and started a new revision.

Was it an idea by marketing people that caused TI to sell the 99/4A as a computer instead of a video game machine?

I think it was more a combination of marketing and engineering. Remember that TI was a very heavily engineering company. I'll give you two examples.

TI developed the pocket calculator. That was an internally driven activity. A bunch of engineers got together the first four-function calculator, called the Dataman, and they built a thousand of them and



Structure built to test interference.

gave them to marketing and marketing went out and did their focus groups or whatever it was they did and came and said "It's not a viable product, it won't sell."

The problem is the people they talked to were the business people that had adding machines on their desks and the answer was that the only people who would buy them are people who are accustomed to using adding machines and the gadgeteers. Our corporate management knew what a neat thing it was and ignored marketing input and went on their merry way.

It happened a second time around when the digital watch came along. The marketing input was that people want that analog watch because you

can look clear across the room at that clock there and get a rough idea what time it is. Nobody's going to sit there and push a button to see what time it is (they said). And the only people who will buy them are gadgeteers.

Well you still see analog watches but the internal guts are a whole lot more accurate than spring-wound watches.

There was a lot of input from engineering, especially on the scientific calculator. We all used slide rules and we all cussed them. Especially if you've gone through school and you had to interpolate and you came up with an answer different from the professor you got marked down, and with the pocket calculator you could get 13-decimal accuracy.

The engineering mind-set of TI management had a lot to do with our successes and our failures because we were so engineering oriented that we did not do a good job listening to the consumer. We thought we were all consumers. A bunch of engineers can be dangerous.

Why did TI put the Peripheral Expansion Box fuse inside the transformer so that you have to dig it out?

Dumb thing to do. If you have a fused product you need to make the fuse accessible. Even today, in a normal PC, there's a fuse inside the box. But a reason for hiding the fuse is that the consumer will put a bigger one in (if it fails). Sometimes it's a good idea to bury the fuse and sometimes it's a bad idea.

Back when the units were very expensive, one of the marketing guys that I knew quite well said that he had a friend who had bought one and his kid had spilled milk on the keyboard and what could we do about it. Could we help this guy. Pouring milk on the keyboard was probably not in keeping with the warranty agreement but we did favors for friends so I had him send it in. It sat around my office for a week or two and then we popped the case and you could see where the milk had poured in. Just for kicks I plugged it in and it came on and worked. I just put it back together and sent it back.

How did you feel personally about the marketing strategy for the TI99/4A?

You're talking to an engineer now. We were accustomed to dealing with end customers who were engineers and companies building electronic equipment, the military. We were accustomed to dealing with engineers as opposed to the general public and consumers. We were not that adept at it. We did a hell of an engineering job and a hell of a production job and not a good marketing job. That's my opinion.

How were you able to write really good tight code (in the GROMs)?

Well, there again some of our calculator experience came to bear. ROM was precious. There wasn't enough bytes available to do everything you wanted to do. So you had to get very, very creative in construct-

Continued on page 20

Structure built to test interference.

Continued from page 19

ing your code to be able to pack 10 pounds in a five-pound sack, so to speak.

We had that same problem in calculators where you try to get all the functions and everything else in there in one chip. We didn't have 64-megabit chips, or 32-meg. chips or 8-meg. chips. So you had to write really tight code to get it all in one chip.

It was learned experience, starting from the calculator days (when) the thought of adding one more component would take an act of Congress. If the design guys try to add components, I would be the first one fighting it because the cost reduction strategy was mine and I had to get cost out. And one way to get cost out was to eliminate components. It was always a struggle. You don't want to add components. Therefore, you had to get good at writing your code. I'm sure some of you have seen BASIC programmers that are real slick. A guy that really understands BASIC can write a 10-line code that does all kinds of things that somebody else might take 50 lines of code to do the same thing.

Why did every manufacturer have a different version of BASIC?

You know, I can do it better than the other guy can. That hasn't changed. Everybody thinks they're smarter than the other fellow. TI did the same thing in its PCs. It had a slightly different version of BASIC because they thought they were smarter, and in some respects they were and some respects they weren't.

Why did TI use a 16-bit CPU and 8-bit I/O?

Well, that was a capability that we had. A lot of times you're under the gun to develop a product you use what you've got, and the guys in the semiconductor arena had developed a 16-bit system. One of the things you want to try to do from a manufacturing standpoint is to not try to reinvent the wheel. So, if you can use something off the shelf, you can spend your creative efforts working in areas where nothing else existed. I decided to teach my guys don't be afraid to steal something if it works and get on with it.

Who was responsible for the BASIC manual? It was much better than manuals that came with Commodore or other computers.

That's an interesting story. We used to have a retail outlet in a mall in Dallas and there was a young fellow that worked there and I visited the store one time and he unloaded on me about errors in our owner manuals and I finally wound up using this kid. He castigated me to the extent that the next calculator product that I had I took the draft of the manual to him after it had been through all the beta tests and everything and gave it to this kid and said "Here, here's a prototype and here's a manual. Find the mistakes." And overnight he covered me up with the stuff. He was not an engineer. He came at it from a user's standpoint. I couldn't transfer him out here because he was still in school so I used him in Dallas and he was a real asset to our whole operation in helping us do our manuals properly. As soon as I got a unit I could spare I sent it to him and he's become a real computer geek since that time. It paid off.

New version of PC99 supports OPA's SOB and AMS card

Mike Wright and Mark van Coppenolle of CaDD Electronics demonstrated the latest version of PC99 (PC99 Stage 5) in Lubbock. This version of the PC-based TI emulator was still in beta testing at the time of Fest West. Among the new features are support for the "PC99 card," Myarc 512K card, AMS card, and OPAs Son Of a Board. Also included are Myarc Extended BASIC 2.12, and Mechatronics Extended BASIC II+. Both include full documentation in the form of Acrobat files.

The PC99 card appears to be the most intriguing addition to the new version of PC99. The "card" comes with a clock that can be accessed via BASIC or assembly programs, operating similar to a Triple Tech card. The clock will take its time and date from the PC's clock.

Wright hopes that third-party developers will write custom DSRs for the "card" to extend the capability of the product. One project might be to write a DSR that allows use of the

Continued on page 22

Mike Wright demonstrates the latest version of PC99. To his left is Mark Van Coppenolle.

Mike Wright demonstrates the latest version of PC99. To his left is Mark Van Coppenolle.

FEST WEST '98 - LUBBOCK

Continued from page 21

PC mouse in a TI application. He thinks it is possible to have the "card" execute other PC programs. An example could be a DSR that executes a utility that might compare two TI-Writer files on a disk and show the differences between them; or perhaps spell checks a TI-Writer

PC99 comes in two versions: PC99 Lite ($47) and the full version of PC99 ($94) which, in addition to supporting all the features of PC99 Lite, supports multiple GROM banks (up to 16 modules), p-system and p-code card, and a mini-screen debugger.

Although one of the strengths of the PC99 programming effort is its disk emulation, which is evident in the fact that it runs TI-Forth as well as the p-system in emulation, Wright said he is not planning to support the Myarc Hard and Floppy Disk Controller.

"There are a couple of problems with emulating a hard disk," Wright said. "One, nobody has the hots to do that. The second thing is that the Myarc Hard and Floppy Disk Controller does not have the world's best coding." PC99 supports up to four double-sided, double-density disks.

According to van Coppenolle, an even larger problem is the absence of documentation on which to base a hard drive emulation. "Just doing the AMS card, which is reasonably well documented, we ran into all kinds of things that were undocumented on the AMS card."

"To do that on the scale of the hard and floppy controller would be staggering, so we avoid certain products," van Coppenole said.

According to van Coppenolle, emulation of the HFDC is the second most requested item behind support for 80-column displays.

Wright can be reached at mjmw@xyvision.com.

Michael Becker, center, displays his controller and video cards at Fest West. Because of a problem with a keyboard, he was unable to demonstrate the cards during a workshop.

THE ART OF ASSEMBLY PART 69

Click on Icon!

By BRUCE HARRISON

It's a different world we live in today from the world we lived in when first we bought our TI99/4A back in 1983. Not only are there no more TIs on the market, but the PCs now available dwarf the capability available only a few years back.

A Rude Awakening

We got a wakeup call recently, when we ordered and received our new Canon BJ-200e printer. This was actually not a new machine, but a "factory refurbished" one, costing only $180. The machine is great! It makes our printed output look like laser quality.

It was the printer's manual that really opened our eyes. In the past, printer manuals always contained a section explaining all of the escape sequences that could be used with that machine, and usually also contained one or more test or sample programs in GW BASIC, which one could run from any garden variety PC. Those days are gone!

The BJ-200 was supplied with three 3 1/2-inch disks of the high density type, containing drivers and fonts. In the manual, we were told how to load these drivers and fonts using Windows 3.1 or higher. No escape codes! No sample programs! The folks at Canon not only assume that you have a PC, but that it's a modern one, capable of running Windows.

That, of course, also means the presence of one or more high density 3 1/2-inch drives. Among our eight computers, three are of the PC type, but none is capable of using either Windows or the high density disks. Thus the supplied drivers and fonts were of utterly no use in our house.

The Help Line

Fortunately, Canon has a user help line at an 800 number. We were a bit scared to call, because the manual said that for those who wanted the escape codes, Canon would fax them in response to a call. We imagined one of those "press 1 for ..." recordings coming through our phone, and that we'd then be asked to key in our fax number on the phone's touch tone keys. No, we don't have a fax either, but Canon assumes that right along with our 198 megahertz Pentium PC with 4,096 megabytes of memory and 160,000-megabyte hard drive, we've also got a fax, or a fax receiving capability built into our PC.

Luckily for us, Canon is not quite that modern. Our phone call was answered, after a slight delay, by a human operator. She heard our plea, then placed us on hold for a while, until a technician was available. After about 20 minutes of elevator music, interrupted now and then by recordings that expressed Canon's regret about our having to hold, we got to talk to a real

Continued on page 24

Michael Becker, center, displays his controller and video cards at Fest West. Because of a problem with a keyboard, he was unable to demonstrate the cards during a workshop.

THE ART OF ASSEMBLY — PART 69

Continued from page 23

person. This lady was very polite and helpful. Not only did she understand that some people still write their own drivers, she also quickly agreed to send us the complete escape code summary by mail. It took about a week to get here from California, but it did arrive safely, and contained what we wanted to know, including some information about limitations on the emulation of an Epson.

This was, however, only a brief summary, and did not explain in any detail how the escape sequences are used. For example, the ESC "k" n sequence, which selects an LQ font, was given as just that, with no indication of what values of n correspond to what LQ fonts. The same was true for the ESC q n sequence, etc.

So once again we owe thanks to Harley Ryan. Harley sent us a spare copy of his Panasonic KX-P2123 manual. That printer, as it turns out, emulates an Epson in a fashion very similar to the Epson emulation mode on our Canon BJ-200e, and its manual gives reasonably complete descriptions of the use of all its escape sequences. With very few exceptions the sequences as given in the Panasonic manual worked perfectly on the Canon. Even more exotic things, like downloading fonts, worked perfectly on the Canon using the methods shown in the Panasonic manual.

Using that information, we were able to download some of the old Jim Peterson screen fonts from our TI into the BJ-200e, and thus prove that this too could be done. We've downloaded sets of fonts from character 33 through character 126 with no problem. We were able, again with the help of Harley's manual, to adapt our own Word Processor so that we can use all of the resident fonts in the BJ-200. (There are eight very nice fonts available, seven of which are of the letter quality kind.) Even the shadow and outline modes mentioned in the Panasonic manual worked perfectly on the Canon, with exactly the same ESC "q" sequences.

The only sequences which don't work on the Canon are the ESC "a" for auto-centering and the ESC "C" for setting page length. The Canon Escape Sequences Summary mentions the non-availability of the ESC "a," but claims that the ESC "C" sequences will work. They don't!

All in all, however, we're delighted with the quality of the printing done by the Canon BJ-200e, and would recommend it to anyone who's in the market for a new printer. Besides providing fast and excellent quality printing of both text and graphics, it is quiet. There's a barely audible *whoosh* sound while printing. This is a pleasant change from the loud *brapp, brapp, brapp* sound of the impact dot-matrix printers we've used.

CoCo Revisited

Some time ago, we mentioned our initial reactions to the Radio Shack Color

THE ART OF ASSEMBLY — PART 69

Computer that we picked up at a thrift store. Since then, we also picked up the necessary parts to add disk drive capability, and also found a complete Color Computer III model.

As is our normal desire, we wanted to be able to program the CoCo in assembly language, so we shelled out some money to Radio Shack for the OS-9 operating system.

In the TI world, we keep running across things that aren't compatible with one another. Usually these involve Myarc products, such as the Hard and Floppy Disk Controller (HFDC). In the case of our CoCo II, we didn't need a third-party like Myarc to create the lack of compatibility — Radio Shack did that all by itself.

Soon after cracking the books on our new OS-9, we discovered that disks initialized through the normal BASIC built into the CoCo were completely incompatible with those initialized by OS-9, and vice versa. OS-9 organizes its disks in a UNIX-like fashion, (similar to a PC DOS disk) while CoCo BASIC uses its own peculiar mid-disk method, where the catalog information is on track 17. Thus anything done with OS-9 can't be used with BASIC, nor can any disk used with BASIC be used under OS-9.

There is another product offered by Radio Shack called BASIC09, which includes a compiler, so that one can cross the bridge in some fashion, but still one can't import an old BASIC program into BASIC09 either, since the disk formats are incompatible.

Undaunted by this, we studied some books on OS-9 assembly programming, then created some simple starter programs on our CoCo II that would run under OS-9. These worked, and quite nicely, but still we were just beginning to learn the CoCo's rather peculiar assembly language. It was like starting to learn assembly all over again. The Motorola 6809 chip's instructions are similar in some ways to the Intel 8088 that's used in our PCs, and in some ways similar to the Z80 chip that we'd had some small experience with.

It was at about this time, when we were just getting started with assembly on the CoCo II, that we found and purchased the CoCo III. This came from the same thrift store, but as a complete package deal, including a very nice RGB monitor, two printers (one dot matrix and one daisy wheel), plus a ton of software and manuals, all for only $150. Everything was in perfect working order! The RGB monitor uses a special interface cable (supplied) to plug into a connector on the bottom of the CoCo III. This can't be used with the CoCo II, however, which has only RF output.

Like the CoCo II, the CoCoIII has no parallel port, so the printers have to be compatible with the strange 4-pin RS-232 port of the CoCo. The CoCo III offers additional features in its BASIC, so that one may use 256 colors 16 at a time,

Continued on page 26

THE ART OF ASSEMBLY — PART 69

Continued from page 25

and one may use a 40-character screen mode. Of course this means that programs written in CoCo III BASIC won't always work on the CoCo II.

The Really Big Shock

One of the things we wanted to do soon after getting our CoCo III was to try out those little assembly things we'd done under OS-9. We dutifully hooked up the CoCo III to its disk drive, put in the disk with OS-9 and some of our experiments, then tried to launch OS-9. The disk drive whirred for a bit, but then nothing! The CoCo III just plain locked up.

A little consultation with our friendly Radio Shack manager indicated that indeed the OS-9 that we'd shelled out $70 for was for use only with CoCo II, and that the different version of OS-9 for the CoCo III would cost us another $100! The BASIC09 for the CoCo III was also available at a similar price. Talk about "Action Figures Sold Separately"! We'd need to lay out another $200 to get any real value out of our CoCo III, then learn another version of BASIC and assembly, and so on.

Since then, our CoCo III and CoCo II have been placed in storage, probably forever. Nice machines, with very advanced features for their time, but now merely relics that clutter up our computer room. Meanwhile, Tandy has stopped making computers altogether. The PCs offered at Radio Shack now are all made by Packard Bell, AST and even IBM, but not Tandy. (As of this spring, Radio Shack will be carrying only Compaq computers.)

Our Tandy 1400FD laptop is still in use, being used to create this article. Most of the time, however, we're pecking away at the TI console, finding new and different things that people thought were impossible to do on that "orphan." Still, we haven't thrown away the CoCos. Maybe some day Radio Shack will cut the price on the OS-9 and BASIC09 for the CoCo III, and then ....

Let This be a Lesson

One day, perhaps we'll invest in a new Pentium PC, Windows 95 and all that other stuff, but no more orphans. Of course, by the time we get around to buying a new PC, we expect it too will be made obsolete soon after we buy it. As we write this, it's impossible to buy any software that will work on our three dinosaur PCs, since none of them has windows or HD disk capability. In a few years, we predict, the present high end PCs will face a similar fate, with the new Holographic Pill replacing disk drives completely, and with millions of megabytes required to run Windows 2000. No matter, so long as we still have our old DOS 3.xx PCs running the old software, we'll still have all the computer power we need, and then some.

In the Meantime ...

We'll still use our TI to answer correspondence and to serve as an outlet for our creativity. With the new printer, our output will look just as good as

THE ART OF ASSEMBLY — PART 69

anything we could do with even the most modern PC, at a small fraction of the cost. Next month, maybe we'll start working on something new, like a mouse-oriented GUI system for the TI. (Whatever happened to Joe Ross and his C-Shell?)

TELECOMMUNICATIONS

Choices for free e-mail abound

BY JOHN KOLOEN

Although many TI'ers use Juno for e-mail, it's not the only company that offers free e-mail services. In fact, the number of Internet companies that offer free e-mail is expanding and, even if you subscribe to one already, it makes sense to compare their services. After all, you are paying a price for free e-mail, it's just not coming out of your bank account.

So what does it cost have a free e-mail account? Well, depending on the service, you're expected to pay with information about yourself. The companies want to know about you. How much money you make. Your hobbies. Your marital status. Your ZIP code. The companies use this information to sell advertising on their sites and to target particular ads based on user demographics. If you tell the e-mail provider that you're a big sports fan, don't be surprised to see web ads promoting sports-related products or web sites. But that's the price you pay for e-mail that you don't have to pay for. I wouldn't call it "free."

In addition to Juno, four sites you might want to consider are Yahoo Mail, Hotmail, MailCity, and MailExcite. These are among the major e-mail services. There are at least 20 of them across the country.

Juno	www.juno.com
Yahoo Mail	www.my.yahoo.com
Hotmail	www.hotmail.com
MailCity	www.mailcity.com
MailExcite	www.mailexcite.com

Some of them, such as Hotmail, support a variety of devices, including PCs, Macs, hand-held devices or WebTV-type units. Others, such as Juno, require a Windows PC because it uses proprietary software. Of course, users can connect to Juno using local phone numbers. This means you don't need an Internet service provider to have e-mail.

Most of the e-mail services don't care what kind of hardware or software you're using as long as you connect to the Internet using an Internet service provider and a web browser. This leaves the TI and Geneve out in the cold.

As many as one-third of all e-mail users subscribe to these free services, and

Continued on page 28

TELECOMMUNICATIONS

Continued from page 27

the number is expected to continue rising. It's hard to pass up a free e-mail account. Hotmail is currently the leader, with some 12 million subscribers.

Why would anyone who has an e-mail account through an ISP want to subscribe to a free e-mail service? To save money, of course. Most ISPs charge for additional e-mail accounts. With one of the free e-mail providers every member of your family, including the dog, can have an e-mail account.

While you sacrifice some of your privacy to obtain a free e-mail account, most of the services say they do not sell your name to advertisers or other businesses. If you notice a lot of spam coming your way, it's a simple matter to cancel your account. So, while the accounts aren't exactly free, you do get something in exchange for what you pay.

BEGINNING C99 PART 9

Wrapping Things Up

BY VERN JENSEN

This will be the last of my c99 articles. I know this may come as a surprise to some of you, but I never intended for the series to go on for a long time. As you will notice from the title, this series was intended mainly to get you started with c99. Having done that, I hope you now know enough to continue on your own.

So today we will cover a few odds and ends, and then wrap things up with a complete working program — a basic "snake eats apple" type game. But first, I have a correction to make. Last issue, I stated that there was no way to extend a line beyond the Editor's 80-character limit. This means character initializer statements, such as the ones used in last issue's first example program, were quite limited.

However, thanks to experimentation by Phil Van Nordstrand, a partial solution was found. He discovered that you actually can break up a single statement between multiple lines by using the backslash (\) character at the end of each line. So the statements from last issue's program might have been written like this:

```
char cData[] =
"000000000000000011111111111111122222222222222333333333333333\
444444444444444555555555555556666666666666677777777777777\
888888888888888899999999999999AAAAAAAAAAAAAAABBBBBBBBBBBBBB\
CCCCCCCCCCCCCCCDDDDDDDDDDDDDDEEEEEEEEEEEEEEFFFFFFFFFFFFFF";
```

Unfortunately, when I tried compiling this, the compiler gave me a "Line too long" error. Apparently, this technique can be used only when the total combined length of the line is not too long. (I'm not sure what the limit is.) So the backslash character may be helpful in some situations where you need just a bit more space, although it won't work for really long lines.

BEGINNING C99 PART 8

More on Pointers

I realize that my article on pointers a while back maybe have been a bit confusing, so I've decided to do a little review, this time with pictures to help you visualize what I'm talking about. As you may recall, pointers are variables that store the address of other variables in memory. To create a pointer, simply declare a variable as usual, adding an asterisk in front of the variable name:

```
char *charPtr;
int *intPtr;
```

The char and int specifiers only tell the compiler what type of variable these pointers will be pointing to. (i.e. charPtr will only point to char variables, while intPtr points to integers.) Both pointers are actually the same size, taking up 16-bits (the size of an Int). That is because pointers need all 16 bits to be able to specify all the various memory locations in RAM that their destination variable could be sitting in. So don't let the char and int specifiers confuse you - when applied to pointers, they only specify what type of variable the pointer will point to; it doesn't specify how much memory the pointer itself takes up.

As you may recall, assigning the address of a variable to a pointer is done with the & (address of) operator. So this would make our pointers above point to the variables below:

```
char myChar;
int myInt;

charPtr = &myChar; /* Store address of myChar in charPtr */
intPtr = &myInt;  /* Store address of myInt in intPtr */
```

If you were to take a look at the values contained in charPtr and intPtr, you would most likely find a 5-digit number (such as 31298) that specifies the physical location of the myChar and myInt variables in memory. Naturally, you don't want these numbers - you want to be able to access the variables these pointers are pointing to. To do that, you use the dereferencing operator (*):

```
*charPtr = 5;   /* myChar = 5 */
*intPtr = -500; /* myInt = -500 */
```

Whenever you use the dereferencing operator with a pointer, you gain access to whatever the pointer points to. So the statements above actually store 5 in myChar, and -500 in myInt. You can read pointers in the same way:

```
char a;
int b;
```

Continued on page 30

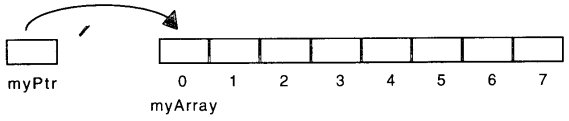
Continued from page 29

```
a = *charPtr; /* Assign contents of myChar (5) to a. */
b = *intPtr; /* Assign contents of myInt (-500) to b */
```

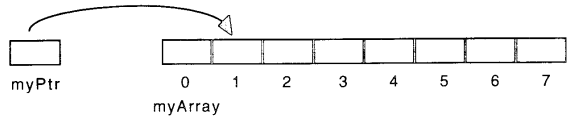
So far, so good, right? But things get interesting once you add arrays to the dimension. First of all, what is an array? It is simply a bunch of char or int variables laid out consecutively in memory. So if you have a pointer pointing to the first element of an array, and you increment the pointer, its value will now point to the next element of the array. Below is some code, as well as a visual example of what happens.

```
int myArray[8], *myPtr;

myPtr = myArray[0]; /* Point to the first element of the
array */
```



Now when you use *myPtr, you're actually reading the first element of the array. When you increment the pointer (myPtr++;), the pointer's address is changed, making it point to the next element of the array.



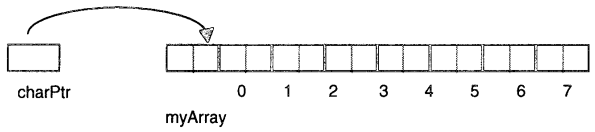
Keep in mind that incrementing the pointer itself is much different from incrementing what it points to:

```
myPtr++;      /* Increment pointer */
(*myPtr)++; /* Increment what it points to. */
```

When you increment a pointer as we do in the example above, the pointer is moved 1 byte forward in memory if it is a char pointer, or 2 bytes forward in memory if it is an int pointer. This is why the array example above works correctly. When we increment myPtr, it is moved 2 bytes forward in memory (2 bytes is the size of an int), thus making it point to the very next element in the array. If our pointer were a char pointer instead of an int pointer, incrementing

it would move it forward only 1 byte in memory, which obviously wouldn't work correctly if pointing to an int variable:

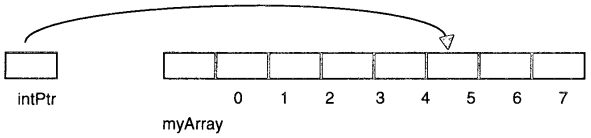
The dotted lines above show the separation between the two bytes that make up each int element. As the diagram shows, if a char pointer were used to point



to this array, incrementing it would result in a pointer that points to the second byte of the first element. So now you know what the difference is between char and int pointers, and why an int pointer should always point to an int variable (or array), and a char pointer should always point to a char variable or array.

The rule described above holds true for adding or subtracting any value from a pointer. Here's an example:

```
intPtr = myArray[0];
intPtr = intPtr + 5;
```



What's the Fuss About?

Naturally, this may seem like overkill. Who needs pointers, anyway? You get along in BASIC without them perfectly well. But in C, things are different. For one thing, you need to pass the address of a variable to a function if you want that function to be able to change the variable, as demonstrated by functions such as the Key function in GRF1:

```
int s, k;
s = Key(0, &k);
```

In addition, you can pass the address of an array to the function, and then use the function's pointer as if it were the original array, thus allowing you to access all of its elements, as well as make changes to the array if necessary.

I hope this clears up any confusion any of you may have had about pointers.

Continued on page 32

BEGINNING C99 PART 8

Continued from page 31

The concept is really quite simple once you get used to it, and enables you to do many things that would otherwise be impossible.

Our Final Project — A Game

I thought it would be fitting to conclude this series with a "real" application; one that actually does something, which in this case is a game. I actually started this game a long time ago, and only recently got around to finishing it, so not everything in the code is necessarily the best way to do things. For example, my global variables don't start with "g", making them much more likely to conflict with local variables. However, I'm sure you will find that most of the code is pretty clean.

To run this program, you will need to copy the files SOUND and SOUND;H from the c99 Libraries disk to your work disk. There are actually two sound libraries, the other one called SND;O. Since I used SND;O in Virus Attack, I decided to test out the other library for this project.

Debugging Your Code

Debugging is hands-down the hardest part of c99 development. I ran into many problems while working on Virus Attack, many of which were easy to fix, some of which were harder, and some which were nearly impossible. One or two bugs I fixed by changing my code to something similar that does the same thing, but using a slightly different method. I did this in exasperation after not being able to figure out what was causing the problem. To this day I still don't know what the problem was; I was just glad enough to have it gone when I changed my code slightly.

When you create any program that's longer that a simple test application, you are going to run into bugs. And lots of them. My suggestion is this: get a piece of paper and write down all the things you see wrong with your program when you run it. Include ideas you may have for fixing those problems. Then quit and load up the source code and work on fixing each problem. If you can't figure something out, remember to keep an open mind! The problem may be completely different from what you had suspected; something totally "unrelated" to the bug.

And finally, don't be afraid to add tests to your code. For instance, if you're not sure whether a piece of code is being executed, you may add a call to HChar within that code to place a certain character at a specific location on the screen. Then when you run the program, if you see that character, you know the code is being executed.

Finally, print your source code! If often helps to be able to go through your program, line-by-line, seeing exactly what is happening in a certain spot, while running the program on your TI.

As some bug-hunting examples, while I was working on the Snake game for this issue, I ran into two particularly nasty bugs. One was that whenever the snake moved up, the image for the snake moving left was drawn, rather than the image for the snake moving up. The code that handles this looks like this:

```
if (vDelta == -1)
  HChar(headRow, headCol, cHead, 1);
else if (hDelta == 1)
  HChar(headRow, headCol, cHead+1, 1);
else if (vDelta == 1)
  HChar(headRow, headCol, cHead+2, 1);
else
  HChar(headRow, headCol, cHead+3, 1);
```

Since I didn't know whether the first HChar statement was indeed being called and just not operating as expected, or if the first if statement wasn't working right, I modified the first HChar statement to draw an apple instead of the snake's head. Since I still got the same results after compiling and running, I knew the first if statement wasn't working right. But why?

It turns out the problem wasn't in this section of code. Is was where I declared the variables! I had stupidly declared the vDelta and hDelta variables as chars instead of ints. But chars can't have negative values! So that's why the if statement wasn't working correctly. But here's the funny part. The code below for moving the snake's position did work correctly, even though the variables were chars:

```
headRow = headRow + vDelta;
headCol = headCol + hDelta;
```

Why this worked even when the snake moved up or left is beyond me. But the point is to never assume you know what the problem is, and especially don't assume you know what the problem isn't!

The other major problem I ran into wasn't mine: it was due to the sound library I was using. It turns out the SOUND library on the c99 Libraries disk apparently has a bug in the Sound2() function that plays 2 sounds at once. For some reason it plays them much longer than it should. The solution? Either use Sound1() or Sound3() (which I haven't tested), or use the SND;O library instead. (Which I used in Virus Attack, and works beautifully.)

GetTime

One of the coolest functions in this code was actually written by Bruce Harrison. Towards the end of the program, you will see a function called GetTime. This is a simple function that returns the number of 60ths of a second that have passed since the previous call to GetTime. This code is invaluable for keeping your program running at a constant speed.

Continued on page 34

BEGINNING C99 PART 8

BEGINNING C99 PART 8

Continued from page 33

First, call GetTime to clear the counter to 0, then call it later in your code to find out how much time has passed since the first call. So pseudo-code for the main loop of your program would look something like this:

```
#define kDelay 5

RunGame()
{
  char ticks;

  GetTime(); /* Clear timer, ignoring value returned */

  PlayGame();

    /* Make sure minimum amount of time has passed */
  ticks = 0;
  do
  {
    ticks = ticks + GetTime();
  } while (ticks < kDelay);
}
```

The code above makes sure the PlayGame routine is not called any more frequently than every five 60ths of a second. If the PlayGame routine goes faster than this, the do loop at the bottom of RunGame() waits until the required amount of time passes. You can easily change the amount of delay by changing the kDelay definition.

The snake game actually modifies the amount of delay on the fly, making the game go faster and faster each time the snake eats an apple. There are many uses for a routine like this. One is for simply delaying a specified amount of time. I implemented this in the Wait() routine, also at the bottom of the program. This is called whenever the snake eats an Apple, to give the player a moment to catch their breath.

Improving the Code

There are many things that could be done to improve the game, such as adding a pause key, implementing FCTN-9 to return you to the title screen, adding a score, and more level layouts. In addition, the level and lives numbers don't display properly above 9. That's because I "cheated" - I called HChar(row,col,'0'+number,1) to display the number. This could easily be fixed by using Bruce Harrison's excellent PutNum routine.

I'm sure you could also come up with other ways to tweak the game. So feel free to make any changes you like, and most of all, have fun with it! I hope you have enjoyed this series, and have found it useful in beginning your journey into c99 programming.

Program listings associated with this article are included with the March/April MICROpendium disk. They were too lengthy to include in this issue of MICROpendium. Readers may also obtain the programs from Jensen for $2. He can be reached at: Vern L. Jensen; 910 Linda Vista Ave.; Pasadena, CA 91103—Ed.

Correction to part 8

There's an error in the Jan/Feb 98 c99 article (part 8). The first example program contains this line on page 24:

```
ChrDef(chrNum[c],&cData[n*66]);
```

The line should be changed to this:

```
ChrDef(chrNum[c],&cData[c*66]);
```


BEGINNING C99 PART 8

EXTENDED BASIC

Program tests high frequency hearing capability

BY W. LEONARD TAFFS

The following article originally appeared in the newsletter of the Southwest 99ers User Group.—Ed.

Here is a program for testing one's hearing. Unfortunately, as we get older we lose the ability to hear higher frequencies. The fact that super high-fi sets boast of magnificent frequency response ranges to 20,000 Hz. and up, hearing frequencies at the upper end of this range is somewhat subject to one's imagination.

The lower end of frequency response in this program is, of course, limited by the limits of the TI99/4A. Best results of this test are achieved if you can hook up your TI audio output to a hi-fi system. Otherwise, you have the additional limitations imposed by the frequency response of your monitor.

The ear-testing program ends at line 380. After this is a simple demonstration of a one octave scale in C (the white key scale on a keyboard).

The frequency range covered with this testing program is set in line 170. If you are one of the fortunate who can hear frequencies above 9000 Hz., change the 9000 to a higher range.

This program provides a "comparison" frequency option, which means that each time the frequency range is gone through you can hear a comparison tone to compare to the rising frequencies. This is useful when the frequency range goes out of your hearing capacity, you no longer hear two tones.

You are asked at the beginning of

Continued on page 36

EXTENDED BASIC

Continued from page 35

the program to set a STEP rate (Z). The lower the rate you input the longer will be the time it takes the program to reach higher frequencies.

The DELAY option allows you to set the time interval between sounding frequencies.

EARTESTER

```
1 REM [*EARTESTER] 10-4-97
By W. Leonard Taffs, SW99ers
 !021
100 CALL CLEAR :: DISPLAY AT
(1,10):"EAR TESTER": :"By W.
 Leonard Taffs, SW99ers" !21
5
110 DISPLAY AT(5,10):"HOW HI
GH A FREQUENCY CAN YOU": :"H
EAR? (Comparison tone opt.)"
 :: DISPLAY AT(24,1):"Press
any key to continue" !235
120 DISPLAY AT(15,1):"FREQUE
NCY RANGE SET IN L170" !121
130 CALL KEY(0,K,S):: IF S<1
 THEN 130 !200
140 INPUT "ENTER STEP AMOUNT
: ":Z :: PRINT :: INPUT "DEL
AY INTERVAL: (0-1000) ":D ::
 CALL CLEAR !222
150 CALL CLEAR :: INPUT "WAN
T COMPARISON TONE? (Y/N) ":Y
N$ :: IF (YN$="Y")+(YN$="y")
THEN YN=1 ELSE YN=0 :: IF YN
 THEN PRINT ELSE CALL CLEAR
!231
160 IF YN THEN INPUT "SET TO
 WHAT FREQUENCY? ":Y :: CALL
 CLEAR !224
170 FOR X=110 TO 9000 STEP Z
 !081
180 CALL KEY(0,K,S):: IF S<>
1 THEN 220 !226
190 IF (K=81)+(K=113)THEN 29
0 !096
200 IF (K=82)+(K=114)THEN 35
0 !158
210 CALL KEY(0,K,S):: IF S<>
1 THEN 210 !216
220 IF YN THEN 230 ELSE 250
!206
230 CALL SOUND(50,Y,5)!076
240 FOR DELY=1 TO D :: NEXT
DELY !173
250 CALL SOUND(50,X,0)!070
260 DISPLAY AT(12,6):"SOUND
FREQUENCY:";X :: DISPLAY AT(
14,6):"DELAY";D :: DISPLAY A
T(16,6):"STEP";Z !131
270 IF YN THEN DISPLAY AT(24
,1):"REFERENCE TONE:";Y !110
280 CALL KEY(0,K,S):: IF S<>
1 THEN 340 !091
290 IF (K=81)+(K=113)THEN DI
SPLAY AT(20,1):"LAST FREQUEN
CY HEARD: ";X :: CALL SOUND(
750,X,0)!206
300 IF (K=81)+(K=113)THEN IF
 YN THEN DISPLAY AT(22,1):"R
EFERENCE TONE:";Y: : :: CALL
 SOUND(750,Y,0):: GOTO 350 !
123
310 IF (K=82)+(K=114)THEN 35
0 !158
320 DISPLAY AT(24,1):"Press
Any Key to Continue." :: CAL
L KEY(0,K,S):: IF S<1 THEN 3
```


EXTENDED BASIC

```
20 !175
330 DISPLAY AT(9,1):RPT$(" "
,140):: DISPLAY AT(21,1):RPT
$(" ",84)!219
340 NEXT X !238
350 REM ** RUN AGAIN? ** !08
6
360 PRINT :: INPUT "RUN AGAI
N? (Y/N) ":RA$ :: IF (RA$="Y
")+(RA$="y")THEN X,Y,YN=0 ::
 GOTO 140 ELSE 370 !059
370 PRINT :"TONE: ";X;" SET"
;Y :: CALL SOUND(1000,X,0)::
 IF Y<>0 THEN CALL SOUND(100
0,Y,0)!164
380 END !139
390 REM ** C SCALE ** !109
400 CALL CLEAR :: ZZ=22 :: T
=1 !129
410 READ A$,B$,C !011
420 IF A$="END" THEN 460 !14
2
430 DISPLAY AT(ZZ,1):A$,T:"
";B$: : :: CALL SOUND(320,C,
0):: ZZ=ZZ-2 :: T=T+1 !142
440 FOR DELY=1 TO 50 :: NEXT
 DELY !152
450 GOTO 410 !234
460 ZZ=22 :: CT=CT+1 :: IF C
T=3 THEN PRINT :: STOP ELSE
RESTORE :: T=1 :: CALL CLEAR
 :: GOTO 410 !232
470 DATA DO 262,MIDDLE C,26
2,RE 294,D,294,MI 330,E,33
0,FA 349,F,349 !188
480 DATA SOL 392,G,392,LA 4
40,A,440,TI 494,B,494,DO 5
23,C',523,END,0,0 !031
```

TUTORIAL

Using a scanner to create pictures for Page Pro

By TOM JAKABFY

Jakabfy is a member of the OSHTI User Group. This article was originally published in the group's newsletter.—Ed.

When a friend tried to use his TI to copy a logo for another friend's small business, he used TI-Artist. He tried various fonts to emulate the words but there was no font that exactly emulated the designed letters. So he asked me if it would be possible to scan the logo into my son's PC. Thus began a series of trials and errors.

We had to become familiar with the scanner and software. Another friend gave me some instruction on how to use the software and I was off to the races.

The scanner is a DEXXA 256 gray scale scanner. The PC software is Photomagic 1.0 LE.

Continued on page 38

TUTORIAL

Continued from page 37

The scanner was set to scan line art, basically black and white with 100 DPI resolution. This is important to remember because you want only dots, not shades of gray.

I scanned the image and after cropping saved it to a 3½-inch disk in PCX format. Using PCX is important.

To minimize the size of the image, I used Photomagic's Edit-Size option. This allows you to reduce the image and determine if it is still crisp enough for reproduction. Some images could be reduced by 30 percent and still give excellent results. Reducing the size of the image also reduces the disk space required to store it in PCX format. I recommend you save the initial image at 100 percent and then play around with the size until you are satisfied.

If you do not adjust the size of the image at this point, you may end up with a rather large Pro Pro Pix.

Next I copied the file to a 5¼-inch floppy. Now we are finished with the PC. From now on we're using the TI.

To copy the PCX file to the TI, I used PC-Transfer V1.1 by Mike Dodd. PC-Transfer works with either a CorComp or a Myarc disk controller, but not a TI controller. If you don't have one of these controllers, you will have to use a program such as Magic FM to transfer the file from the PC to a TI disk.

Now we are in PC-Transfer. After identifying the PC and TI drives, you will be prompted for a "conversion" file name. Type "DSK1.DF128." This utility will convert the PC PCX file to D/F128 on the TI.

After reading the PC disk, select the file to copy and then "E" to execute. Enter a TI file name and then let the program do its thing.

Now we have the PCX file in a format that the utility program PCX2PP by Dan Gazsy can convert to Page Pro format.

You need the program GOFER V1.1 by Gazsy to do the conversion.

When you load GOFER, go to "load additional modules" and at the prompt type "DSK1.PCX2PP" to load the utility. From here it is an easy matter to select the drive where the PCX file. The program will display the names of the D/F128 files that you converted using PC-Transfer or Magic FM. Use the "S" key to select the file and then give it a name.

As the conversion proceeds, you will be informed about the version of PC Paintbrush that the PCX image relates to, as well as the number of pixels (length and width) that the image contains.

That's it. Now you have the PCX image in Page Pro format on your TI disk.

After doing several PCX image conversions I noticed that the size of the image in Page Pro image can be very large. To get the PCX picture to a smaller size, I used Page Pro FX. As noted above, you can also reduce the size in Photomagic or other PC scanning software.

HARDWARE PROJECT

Replacing a TI keyboard

This project is simple but can reap big benefits

BY HERMAN GESCHWIND

The author is a member of the VAST User Group, in whose newsletter this article first appeared. All risk for attempting this hardware project lie with the reader.—Ed.

Replacing a keyboard on a TI99/4A is a simple job that requires no special skills beyond the use of common sense and ordinary prudence. Nor are any special tools or soldering required.

By the way of tools, all that will be required is a medium-sized (one-eighth inch) Phillips head screwdriver and perhaps a flashlight.

First off, inspect your keyboard purchase and make sure that the keyboard layout is the same as your existing TI keyboard. If there are any extra keys, or if the keycaps are labeled differently, stop. Adapting a non-99/4A keyboard for use with the 99/4A requires special skills which are beyond the ken of the average user, if it can be done at all.

TI subcontracted for keyboards from various sources in Japan and Korea, thus the shape and texture of the keycaps might be different. Don't let that put you off. The main thing is that the number of keys and the layout are the same as your original.

If you keyboard passes this test, unpack it and test all the keys. Look for keys that might be binding or seem to stick. Try to get a feel of the key action. Once you are satisfied that your replacement keyboard passes this test, proceed.

Disconnect all cables from the console. Your console should be cool. If you have just used it, allow it to cool down and to dissipate any residual electrical charges. Observe normal precautions about electricity.

On a clean work surface, turn the console over. There will be seven Phillips head screws to undo — four at the narrow end and three at the other end. All are recessed. After undoing the screws, the bottom shell should come off. If it does not, check to see that you have removed all the screws.

Once the bottom shell has been removed, three components will be visible — a printed circuit board approximately four inches square which houses the

Continued on page 40

HARDWARE PROJECT

Continued from page 39

power supply, the keyboard assembly, and a larger assembly which runs parallel to the keyboard and power supply all the way across the console to the motherboard.

The power supply is held in place by two Phillips head screws along the edge closest to the keyboard assembly. Remove these screws and gently move the power supply board an inch or so to the side. Don't force anything, since these are wires attached to the power supply board.

Next, locate and undo the four Phillips head screws that secure the keyboard assembly. Gently lift the keyboard assembly an inch or so. You might have to lift the edge of the motherboard just a little to allow the keyboard to clear.

At this point you will notice that the keyboard is attached to the motherboard by a ribbon cable. Locate the connector at the motherboard side. Use the flashlight if necessary. Gently pry the connector loose by pressing the screwdriver. Do this in several steps along the length of the connector. The idea is to remove the connector without bending any of the pins on the motherboard.

Once the connector as been unsnapped, remove the keyboard assembly. Take time to take a good look at the row of gold pins that were uncovered when the connector came off. Make sure that all the pins are straight and evenly spaced. If not, gently try to straighten any pins that are bent. If this is necessary, proceed with caution and use minimum force.

Insert the new keyboard assembly without forcing it in place. Line up the connector of your new keyboard with the pins. Gently start connecting the pins with the new connector. Visually double-check that all pins are mated with the connector. If not, pry the connector loose and try again. Being hasty at this step could result in a broken pin. Be careful! Once you are sure everything is lined up properly, firmly but gently press the connector down onto the pins.

Relax now, the most ticklish part of the job is behind you. Next, observe that there is approximately one to two inches of extra ribbon cable. This extra length needs to be folded up and into the space between motherboard and console housing as you simultaneously seat the keyboard assembly into place. To seat the assembly, you also need to lift the motherboard edge just a bit for the edge keyboard assembly to slip under it, simultaneously try to get the extra length of ribbon cable to fold as described. If this sounds like a job for three hands, you are right. A helper at this point of the installation will make things easier.

Now, line up the screw holes and secure the keyboard assembly with one screw. Lift the console a little bit (remember, everything should still be upside down) and test the row of numeric keys, particularly keys numbered four through eight. If any of these keys appear to bind or feel different from the one or zero keys, then the ribbon cable is not folded properly. Undo the one screw and recheck the ribbon cable. Remember, any extra length of ribbon cable

HARDWARE PROJECT

should not touch the keyboard assembly but be tucked into the empty space above the motherboard.

Once the keys check out, replace all four screws on the keyboard.

Relocate the power board, make sure that the on/off switch connects properly. If necessary, lift the board just a little and observe the switch action. Once the switch works properly, secure the power board with its two screws.

Replace the console bottom cover. Make sure that it lines up properly with the top before replacing the screws. Seat all seven screws and then tighten.

You replacement keyboard is now installed and should be working properly. Don't throw away your old keyboard just yet. If it has only a few keys that refuse to work, take it to your friendly radio or TV repair person. Quite often the judicious application of contact cleaner and a general cleaning can restore a balky keyboard to pristine health. You may even want to do this job yourself.

NEWSBYTES

Harrison developing Play-In for MIDI-Master

Bruce Harrison is developing a new product for MIDI-Master owners, called Play-In. The program allows users to play music on the MIDI instrument and have that music recorded in the TI's memory. From there it can be saved to disk, recovered from disk, and played back through the instrument. The program also includes a "de-compile" feature so that music contained in memory can be written out to a D/V80 files in SNF format. That file can be edited with the Editor/Assembler or Funnelweb, then recompiled using the updated versions of MIDI-Master (V2.5A or V2.5Z) that was created last year.

The target date for release is in May, around the time of the Lima TI fair. Harrison says all functions work but that the program is being refined, tested, and polished for its official release.

CaDD outlines BASIC compiler

The following comes from Mike Wright of CaDD Electronics. It first appeared on the TI list server.—Ed.

The BASIC Compiler for PC99 is a project that has been spec'd. and is now under way. The basic tools (lex and yacc) for the project have also been acquired. The design goals are very simple:

• Take any BASIC or Extended BASIC program and convert it to 9900 assembly code that can be linked with supplied libraries to produce an assembly version of the BASIC program. The "any" implies that there are no restrictions on the way the BASIC program is written. For example, you will not have to

Continued on page 42

NEWSBYTES

Continued from page 41

declare variables at the front of the program. In addition, all BASIC statements, even things like DEFs are allowed.

• Generate a "library" of functions that can be called from the compiled code. Some of these will be TI ROM routines, such as FADD (add two floating point numbers), etc. For the first pass, some may also be GPL routines.

A scheme has been worked out for indexing and storing simple and array numeric variables using the TI 8-byte floating point system. Similarly for strings, but a convenient garbage collect for strings has not been written yet.

The BASIC Compiler (early name BC99) is a PC program that is designed to:

1. Take a BASIC or XB program from a PC99 disk file using dskout.
2. Compile it to produce 9900 assembly code.
3. Import the file into a PC99 disk using dskin.
4. Run the TI assembler to produce an E/A3 file, which can be converted to an E/A5.

The initial specification will allow the generated program to run on any 4A. However, it seems highly desirable for extra memory. This would first be in the Super Space area, and later using AMS memory. The current development version of PC99 now correctly emulates up to 1Mb of AMS memory. Finally, since it is trivial to exchange disks (up to DSDD) between PC99 and a 4A, the compiled programs can easily be transferred and distributed to standard 4A users.

There are many tough and interesting problems to solve in developing this product. One that needs some thought is the FCTN-4 key. Since there is no BASIC interpreter running, this key will probably be disabled. However, what happens if BREAK is used as a statement in a program, etc.

The cost is projected at $47. It is a standalone product that is not part of PC99, and does not require PC99. However, it is designed to be used with PC99. One of the prime reasons for developing PC99 was to have a BASIC compiler. However, like all the products we develop, there is no time commitment or even tentative delivery date. We will only announce the product if and when it is ready to ship.

As of March 5, Wright reports little progress has been made on the compiler as CaDD is directing its resources that finalizing version 5 of PC99.

HUG makes changes

Dan Eicher has been named the new president of the Hoosier Users Group (HUG). The group's post office box has been closed. All correspondence for the HUG, should be addressed to: Hoosier Users Group c/o Dan H. Eicher, 2720 Palo Verde Court, Indianapolis, IN 46227.

Another issue Dan has addressed is the club's S&T Software BBS, according to a TI list server posting by William Lucid. The HUG BBS will stay online as a community resource for TI and Geneve users. All callers are allowed access to download on the first call. The group only asks for the users' full name, address, and telephone number. This information is only necessary on the first call, HUG BBS will maintain user information referenced by user number. HUG BBS (317) 782-9942 parameters are 8 data bits, no parity, and 1 stop bit.

The Hugger Newsletter continues to be published. Officers for the HUG are: Dan H. Eicher (eicher@delphi.com), president; Byrant C. Pedigo (bpedigo@midlink.com), vice president and librarian; and Greg Larson, (uffda@indy.net), secretary and treasurer.

Memberships are available in the Hoosier Users Group, for an annual rate of $22. Membership includes six issues of the HUGger Newsletter, access to the Hoosier Users Group bulletin board (HUG BBS) and access to HUG disk library (no copying charge). Only charge for HUG library disks is cost of mailing for those who cannot pick up disks at monthly meeting. Dues-paying members have voting rights at HUG meetings, which are held on the third Sunday of the month. Dues should be sent to the above address for the Hoosier Users Group c/o Dan H. Eicher.

NEWSBYTES

TI-Tref set for England

The13th TI-Tref is scheduled for Britain for the first time ever Oct. 9-11, according to Richard Twyning, general secretary of the TI99/4A Users Group U.K.

The event will take place in the heart of Robin Hood country at The Beeches Hotel and Leisure Club Wilford Lane, West Bridgford, Nottingham, NG2 7RN; Phone: +44 115 981 8753; fax: +44 115 945 5838

According to Twyning Friday night will give all those attending chance to get acquainted, but the show will actually start on Saturday morning and finish on Sunday afternoon. TI User Group U.K. will charge an entrance fee of £5 which covers both days and covers the cost of the conference facilities.

Costs for rooms per night are single room, £35; double/twin room, £52; executive single, £42; executive twin/double, £62; and suite, £80. All rooms include full English breakfast, free use of the hotel's heated swimming pool, air-conditioned gymnasium, jacuzzi, saunas, and steam room.

All standard rooms have private bathrooms with shower, toilet and sink, color satellite television, direct dial telephone, tea- and coffee-making facilities, and radio. Executive rooms also have teletext TV, a bath as well as a shower, a desk/work area, a hair dryer, and a trouser press. The hotel asks for a £10 deposit per room when booking. The hotel is located within easy access of both Nottingham train station and East Midlands Airport.

Continued on page 44

NEWSBYTES

Continued from page 43

Twyning notes that for those wishing to include the show as part of a longer holiday, the area is famous as the home of the legendary Robin Hood. Visit Nottingham Castle, the city's Tales of Robin Hood Centre, Wollaton Park and Wollaton Hall. You can also visit Edwinstowe in Sherwood Forest and see the Major Oak. The area has historic buildings such as Hardwick Hall to the north, and for those wishing to travel a little further there is the Derbyshire Peak District and Chatsworth House, the home of the Duke of Devonshire.

The hotel is only a few minutes away from Nottingham Forest football ground, the famous Trent Bridge Cricket ground, and to the south, there is Holme Pierrepont, the National Watersports Centre. On the other side of East Midlands Airport there is also motor racing at Castle Donnington Park.

For further information, contact Twyning:
Phone: +44 1623 627670
Mobile Phone: +44 467 44 56 58
Phone: +44 115 913 5811 (Friday and Saturday 7 p.m.-9 p.m. G.M.T.)
Fax/Voice Msg: +44 1623 453 934 (9 a.m.-6p.m. G.M.T. Mon-Fri)
Mobile Fax: +44 467 449 009
E-mail: 007@r-twyning.demon.co.uk

Group holds elections

The Milwaukee Area TI User Group) held recently elected officers.

They are Ted Zychowicz (tedzychowicz@juno.com), president and librarian; Jonathon Johnson (johnsonn@milwaukee.tec.wi.us), vice president; Denis Dann (denisd@juno.com), treasurer; Gene Hitz (genehitz@juno.com), newsletter editor; Tim Tesch (ttesch@juno.com), Geneve librarian; and William Hoehnen (whoehnen@omnifest.uwm.edu), publicity chair.

According to Zychowicz, anyone wishing to contact the group can contact any of the officers at the above e-mail addresses or write the group c/o Hitz at 4122 Glenway St., Wauwatosa, WI 53222. Membership is $10 or family $15. If non-residents want a newsletter sent to them they should arange for it with the newsletter editor, Zychowicz says. Disks can be gotten from the disk library for postage plus $.50.

READER TO READER

Looking for UCSD Pascal

We are looking for a copy of the UCSD Version 4.0 Pascal for the IBM.

Most of our existing TI programs will run on same. Please contact Frank W. Aylstock; 4336 Eureka Ave.; Yorba Linda, CA. 92686-2343.

EXTENDED BASIC

XBCOMPRESS automatically reduces size of XBASIC programs

Compressing the size of Extended BASIC programs can be a tedious affair if done manually. Often you end up spending as much time debugging it as you did compressing it. Which makes Karl Romstedt's Extended BASIC Program Compressor a very useful tool indeed.

The program works with programs saved in MERGEd format. It automatically removes REMarks, combines lines, and shortens variable names.

XBCOMPRESS

```
100 DIM R$(99),O$(99),NV$(99
):: LA$=CHR$(255)&CHR$(255):
: CALL CLEAR :: CALL COLOR(1
3,2,14):: CALL CHAR(128,"FF0
000FF0000FF"):: CALL HCHAR(1
,1,128,352)!161
110 DISPLAY AT(3,1):" ":"
    EXTENDED BASIC": :"
PROGRAM COMPRESSOR": :" by
Karl Romstedt, 10/86": :!235
120 DISPLAY AT(15,1):"This u
tility works with     progra
ms saved in merge     format
 to reduce size and   increa
se speed. Remarks (!)" !172
130 DISPLAY AT(19,1)BEEP:"ar
e removed, lines are      co
mbined and variable names ar
e shortened. Press any keyto
 continue." !167
140 CALL KEY(0,K,S):: IF S<1
 THEN 140 !210
150 DISPLAY AT(12,1)ERASE AL
L:"INPUT FILE?  DSK1." :: GO
SUB 500 :: IN$=N$ !006
160 DISPLAY AT(12,1)ERASE AL
L:"OUTPUT FILE? DSK1." :: GO
SUB 500 :: IF IN$=N$ THEN 16
0 !228
170 ON ERROR 490 :: OPEN #1:
IN$,INPUT ,VARIABLE 163 :: O
N ERROR STOP :: CALL CLEAR :
: GOSUB 840 !150
180 GOSUB 530 :: IF L$<>LA$
THEN 180 !055
190 CLOSE #1 :: FOR I=1 TO V
 :: NV$(I)=O$(I):: NEXT I ::
 FOR I=1 TO V :: X=0 !156
200 X=X+1 :: S$=SEG$(O$(I),1
,X):: FOR J=1 TO V :: IF J=I
 THEN 220 !201
210 IF S$=NV$(J)THEN 200 !12
1
220 NEXT J :: NV$(I)=S$ :: N
EXT I :: GOSUB 840 :: DELETE
 N$ :: OPEN #1:IN$,INPUT ,VA
RIABLE 163 :: OPEN #2:N$,OUT
PUT,VARIABLE 163 :: GOSUB 53
0 !211
230 LO$=L$ !206
240 GOSUB 530 :: IF L$=LA$ T
HEN 390 !073
250 FOR I=1 TO LR :: IF R$(I
```

Continued on page 46

EXTENDED BASIC

Continued from page 45

```
)=LI$ THEN 390 !133
260 NEXT I :: X=ASC(SEG$(L$,
3,1)):: IF X=147 OR X=163 TH
EN 390 !137
270 IF X=0 THEN 240 !252
280 X=ASC(SEG$(L0$,3,1)):: I
F X=147 OR X=163 THEN 390 !0
88
290 RP=POS(L0$,CHR$(176),3):
: S=POS(L0$,CHR$(201),MAX(RP
-2,1)):: IF RP>0 AND(S=0 OR
S>RP)THEN 390 !046
300 IF LEN(L0$)+LEN(L$)>165
THEN 320 !186
310 L0$=SEG$(L0$,1,LEN(L0$)-
1)&CHR$(130)&SEG$(L$,3,LEN(L
$)-2):: GOTO 240 !154
320 RP=POS(L$,CHR$(176),3)::
 S=POS(L$,CHR$(201),MAX(RP-2
,1)):: IF RP>0 AND(S=0 OR S>
RP)THEN 390 !206
330 PX=3 !098
340 X=POS(L$,CHR$(130),PX)::
 IF X=0 THEN 390 !145
350 S=POS(L$,CHR$(201),X-2):
: IF S>0 AND S<X THEN PX=X+1
 :: GOTO 340 !105
360 S=POS(L$,CHR$(199),PX)::
 QS=S+2+ASC(SEG$(L$,S+1,1)):
: IF S>0 AND S<X AND X<QS TH
EN PX=QS :: GOTO 340 ELSE IF
 S>0 AND S<X THEN PX=QS :: G
OTO 360 !093
370 IF LEN(L0$)+X>165 THEN 3
90 !166
380 L0$=SEG$(L0$,1,LEN(L0$)-
1)&CHR$(130)&SEG$(L$,3,X-3)&
CHR$(0):: L$=SEG$(L$,1,2)&SE
G$(L$,X+1,LEN(L$)-X):: GOTO
340 !012
390 X=ASC(SEG$(L0$,3,1)):: I
F X=0 THEN L0$=SEG$(L0$,1,2)
&CHR$(154)&CHR$(0)!087
400 IF X=130 THEN L0$=SEG$(L
0$,1,2)&SEG$(L0$,4,LEN(L0$)-
3)!231
410 PRINT #2:L0$ :: IF L$<>L
A$ THEN 230 !238
420 PRINT #2:LA$ :: CLOSE #1
 :: CLOSE #2 :: CALL CLEAR !
128
430 DISPLAY AT(2,10):"SHARE-
WARE": :"Alteration or use f
or sales promotions prohibit
ed. If   you find this progr
am       useful, send $5 to"
 !249
440 DISPLAY AT(10,1)BEEP:"Ka
rl Romstedt":"2543 Cranford
Rd.":"Columbus, Ohio 43221":
 : : : : : :"PRESS ANY KEY F
OR VARIABLE  NAMES." !146
450 CALL KEY(0,K,S):: IF S<1
 THEN 450 !009
460 CALL CLEAR :: PRINT "OLD
","NEW": : :: FOR I=1 TO V :
: PRINT O$(I),NV$(I)!247
470 CALL KEY(0,K,S):: IF S=0
 THEN 470 !027
480 NEXT I :: PRINT : :"TO R
ECOVER YOUR NEW PROGRAM,TYPE
 ";CHR$(34);"NEW";CHR$(34);"
 THEN ":CHR$(34);"MERGE ";N$
;CHR$(34):: END !164
490 PRINT "YOUR INPUT FILE M
```


EXTENDED BASIC

```
UST BE A   PROGRAM SAVED IN
MERGE      FORMAT." :: END !
104
500 CALL CHAR(95,"007E"):: C
ALL HCHAR(13,16,95,15)!037
510 ACCEPT AT(12,19)VALIDATE
(UALPHA,DIGIT,"!#$%&'()*+,-/
:;<=>?@[\]^_")BEEP SIZE(-10)
:N$ :: IF POS(N$," ",1)>0 TH
EN DISPLAY AT(20,6)BEEP:"Do
not use spaces!" :: GOTO 510
 !096
520 N$="DSK1."&N$ :: RETURN
!113
530 LINPUT #1:L$ :: IF L$=LA
$ THEN RETURN !227
540 PX=3 !098
550 RP=POS(L$,CHR$(154),PX):
: IF RP=0 THEN 590 !244
560 S=POS(L$,CHR$(201),RP-2)
:: IF S>0 AND S<RP THEN PX=R
P+1 :: GOTO 550 !026
570 S=POS(L$,CHR$(199),PX)::
 QS=S+2+ASC(SEG$(L$,S+1,1)):
: IF S>0 AND S<RP AND RP<QS
THEN PX=QS :: GOTO 550 ELSE
IF S>0 AND S<RP THEN PX=QS :
: GOTO 570 !225
580 L$=SEG$(L$,1,MAX(RP-2,2)
)&CHR$(0)!144
590 PX=3 !098
600 RP=POS(L$,CHR$(131),PX):
: IF RP=0 THEN 640 !033
610 S=POS(L$,CHR$(201),RP-2)
:: IF S>0 AND S<RP THEN PX=R
P+1 :: GOTO 600 !076
620 S=POS(L$,CHR$(199),PX)::
 QS=S+2+ASC(SEG$(L$,S+1,1)):
: IF S>0 AND S<RP AND RP<QS
THEN PX=QS :: GOTO 600 ELSE
IF S>0 AND S<RP THEN PX=QS :
: GOTO 620 !069
630 L$=SEG$(L$,1,RP-1)&CHR$(
0)!149
640 LI$=SEG$(L$,1,2):: LN=AS
C(LI$)*256+ASC(SEG$(LI$,2,1)
):: DISPLAY AT(12,19):"LINE"
;LN :: IF P=2 THEN 690 !184
650 PX=3 !098
660 X=POS(L$,CHR$(201),PX)::
 IF X=0 THEN 690 !189
670 PX=X+3 :: LI$=SEG$(L$,PX
-2,2):: FOR I=1 TO LR :: IF
R$(I)=LI$ THEN 660 !005
680 NEXT I :: LR=LR+1 :: R$(
LR)=LI$ :: GOTO 660 !247
690 PX=2 !097
700 VA$="" !064
710 PX=PX+1 :: S$=SEG$(L$,PX
,1):: X=ASC(S$):: IF X>198 A
ND X<202 THEN ON X-198 GOTO
790,790,800 !191
720 IF X>47 AND X<58 AND VA$
<>"" THEN VA$=VA$&S$ :: GOTO
 710 !066
730 IF (X>63 AND X<91)OR X=9
5 THEN VA$=VA$&S$ :: GOTO 71
0 !246
740 IF VA$="" THEN 780 !076
750 IF P=2 THEN 810 !051
760 FOR I=1 TO V :: IF O$(I)
=VA$ THEN 780 !196
770 NEXT I :: V=V+1 :: O$(V)
=VA$ !183
780 IF X=0 THEN RETURN ELSE
```

Continued on page 48

EXTENDED BASIC

Continued from page 47

```
700 !211
790 PX=PX+1 :: S$=SEG$(L$,PX
,1):: X=ASC(S$):: PX=PX+X ::
 GOTO 700 !205
800 PX=PX+2 :: GOTO 700 !089
810 FOR I=1 TO V :: IF O$(I)
<>VA$ THEN 830 !183
820 L$=SEG$(L$,1,PX-1-LEN(VA
$))&NV$(I)&SEG$(L$,PX,LEN(L$
)-PX+1):: PX=PX+LEN(NV$(I))-
LEN(VA$):: GOTO 780 !158
830 NEXT I :: PRINT "VARIABL
E TABLE ERROR" :: BREAK !096
840 IF P=0 THEN DISPLAY AT(1
,1):"Forming line reference
and  variable tables." !148
850 IF P=1 THEN DISPLAY AT(1
,1):"Removing remarks, short
eningvariables, combining li
nes." !162
860 P=P+1 :: DISPLAY AT(12,1
):"PASS";P;"OF 2" :: RETURN
!159
```

MICROREVIEWS

RXB v1005B, PC99 Stage 5, and Download File Converter

By CHARLES GOOD

RXB v1005B
By Richard Gilbertson

Rich sent this to me on disk recently with a note that said, "Review this. Give it away!" I have reviewed RXB at least twice before, but the product keeps getting better. Think what we would have paid for such a product just a few years ago. I remember when XB3 was advertised (but never actually made available) by Asgard for about $70. I remember paying $100 for my first Extended BASIC module in 1983. RXB is completely compatible with TI XB and offers many additional useful features.

As you probably know, RXB is a set of GRAM files designed to replace the TI Extended BASIC and Editor/Assembler modules. To use RXB you need a GRAM device such as a Gram Kracker, or a Geneve, or PC99. I suspect that the largest market for RXB will be users of PC99, because the number of PC99 users has been increasing significantly in recent years.

An obvious change from previous versions can be seen in the RXB startup screen. When you turn on your computer, RXB takes over immediately, bypassing the TI color bar title screen. You get your choice of several options and you have to act fast or RXB will start searching all your drives for a file called LOAD. If you press a number or letter key RXB will immediately go to that drive and load LOAD. If you press the space bar RXB will default to Extended BASIC command mode. If you press <enter> RXB will search your drives for an EA5 file called UTIL1. You can optionally tell RXB which drive contains this file. If you press <period> RXB will shift to Editor/Assembler mode. This is similar to the TI Editor/Assembler module but with added features. For example, you can V(iew) a DV80 text file without loading an Editor/Assembler editor. The feature that is new to the RXB startup screen is BATCH. If you press <comma>, RXB will look for a DV80 file named BATCH and execute this file. RXB lets you create a text file and then "run" this text file just as if you were typing the text into the Extended BASIC editor in command mode.

The next obvious change from previous versions of RXB can be seen in its LOAD program. Just put the RXB disk in drive 1, start your computer, and a great menu of RXB features is displayed on screen. You can read all the various RXB docs from this menu. You can run a really fantastic BATCH file that shows off most of RXB's features. You can run a little program that gives you back your TI title screen. And, you can run the LOADERS program. This program creates a menu of all runnable files on a disk and saves this menu back to the disk as LOAD. This custom LOAD program then can be used to boot all files on the disk. The files can be long or short Extended BASIC programs or any kind of runnable assembly language programs.

With RXB you can switch back and forth between Extended BASIC and Editor/Assembler mode without restarting the computer. You can also, from RXB BASIC, go to TI BASIC without restarting the computer. Just type CALL BASIC or DELETE BASIC in command mode to do this. The only thing this TI Basic lacks is Editor/Assembler assembly language support.

A very useful feature added to RXX 1005B is a new call key routine. The format is CALL KEY("",0,K,S). If your CALL KEY statement looks like that then any keypress will continue your program. If your code looks like this — CALL KEY("yYnN",0,K,S) — then only the upper or lowercase y and n keys will be accepted as input. This is great for responses that need a yes/no answer. You can also use this format — CALL KEY(CHR$(13),0,K,S) — which will expect you to press only the <enter> key. These formats let you get rid of a lot of the code that usually follows CALL KEY statements. No more IF K=__THEN 200 ELSE IF K=__ THEN 300 ELSE IF... .

I really like using RXB on my PC as PC99's default Extended BASIC. When I start PC99 then RXB starts immediately and starts searching my PC99 "drives" for LOAD. I don't have to "Press any key to continue" and then "Press 2 for Extended BASIC." Only RXB can do this on PC99. I have Funnelweb more or less permanently installed on my last PC99 "drive" and RXB will automatically and quickly. go through all the drives until it finds LOAD on my last drive. By sector editing the RXB6 GROM file to immediately look for DSK4.LOAD I can speed this process of always automatically

Continued on page 50

MICROREVIEWS

Continued from page 49

starting Funnelweb every time I run PC99.

RXB comes on two TI DSSD disks. Send me $2 and I will mail them to you. Your money pays for the disks and postage. I can also e-mail RXB (for free) in PC99 or TI Files format. I cannot e-mail it in V9T9 format.

PC99 STAGE 5
by CaDD Electronics

It is hard to say enough good things about this 99/4A on a PC emulator. I have twice reviewed earlier versions. By the time you read this stage 5 will be available. The upgrade cost to registered owners will be about $10. New users have to pay either $47 or (I think) $95 depending on which version you purchase. If you are not an assembly programmer then the $47 version should be all you need.

The new version adds emulations of the Myarc 512k memory expansion card, a 1-meg AMS card, the Super Space II module, a clock card, and the Oasis Pensive Abacutors "son of a board" operating system. Even in today's depressed market, any one of these devices for a real 99/4A system would cost you at least $50. With PC99 you can the equivalent of a very fancy expanded 99/4A system running at better than TI speed on your desktop or laptop PC, as long as your PC is at least a 486x66.

If you have a PC, you really should have PC99 on it Then you can run all your favorite 40-column 99/4A software on your "modern" computer at home or at work.

I have my PC99 set up with the Myarc floppy disk controller controlling four DSDD "floppy drives" and the Myarc RAMdisk set up as DSK5. I have Funnelweb and a bunch of useful utilities installed on the RAMdisk and use the floppy drives as needed for other things.

Using RXB v1005B my PC99 setup goes directly to Funnelweb as soon as I start PC99. The Myarc RAMdisk can be formatted up to 1,600 sectors of RAMdisk, which is how I have mine set up. Unlike the real Myarc RAMdisk, the PC99 emulated RAMdisk keeps its memory when you shut down PC99 and turn off your computer. In fact, if you use PC99's configure program to remove the Myarc RAMdisk from your emulation (which you have to do if you want to use the AMS card) and then later add the Myarc RAMdisk back in, the software you had on the Myarc RAMdisk will still be there.

You can also configure the Myarc memory card to use Myarc Extended BASIC, if you already have it. The docs for this version of Extended BASIC are included with PC99 stage 5. Myarc XB is extra fast, but buggy, and will not run normal XB programs with embedded assembly code or with assembly CALL LOADs. You are supposed to be able to use the Myarc memory expansion card as a print spooler, but this never worked properly on a real Myarc card and thus doesn't work on PC99's emulation. Memory in the Myarc card can be bank switched under assembly software control. Using the new PC99 debugger it is possible to examine and alter the contents of each bank of Myarc card memory.

You cannot use the Myarc and AMS cards at the same time, since they both include the normal 32K memory expansion. Software you put into an emulated AMS card's memory is not retained when you shut down PC99. Few people have a real 1-meg AMS card because of the expense of purchasing the card. Soon every PC99 owner will own such a "card" and this may stimulate a market for AMS-specific software. I have earlier reviewed some of the available software.

The clock card is not a direct emulation of any real 99/4A hardware, but it is compatible with real CorComp clock cards. The emulated clock card recognizes the PC's date and maybe also the time. You can, from BASIC, alter the time and date of the emulated clock card without changing the PC's clock. When I run the appointment calendar program REMIND ME under PC99 the correct date is recognized from the emulated clock card so that REMIND ME opens with the correct day of the month indicated. REMIND ME does not recognize the emulated clock card's time of day, but this is not at all important to the operation of REMIND ME.

Super Space II is emulated and can be set up with the configuration program. Super Space is a bank switching system for the cartridge RAM. Very little software exists for Super Space II. The only such software I know of is module dump software that lets you dump a module to disk and then run the module off of disk IF you have a Super Space cartridge. This is of little use to PC99 users. However, Super Space also lets you run software that requires a "supercart," such as some of the Infocom games that were converted to 99/4A format. I recommend that you enable Super Space in your PC99 emulation and leave it permanently enabled. This does not interfere with the operation of any module or disk software and if you install the emulated E/A module then you can run supercart software. Infocom games that required supercart memory did not run on previous versions of PC99 but do run on PC99 stage 5.

The SOB (son of a board) operating system is really nice, although RXB completely bypasses all its nice features. You start with a slightly modified color bar screen and when you press the "any key" to continue you get a screen that displays the modules you have on line. Using TI's "review module library" system it is possible to have several modules available in a series of menus. The SOB screen shows all of these in one menu. The SOB screen also allows you to bring up a directory of all your drives and run any EA5 software from this menu. I can get a directory of any of my four "floppy drives" as well as my Myarc RAMdisk "drive."

A nice feature that was supposed to work in previous versions but didn't work well is the overlay file. This is the equivalent of the little hint strip you

Continued on page 52

MICROREVIEWS

Continued from page 51

place above your 99/4A number keys. Several overlay files come with PC99 and you can make your own with a text editor. You can configure PC99 to display the overlay file of your choice to the right of the "TI" screen. I made a combined TI-Writer/BASIC overlay file for myself which I find very helpful.

On a regular CRT monitor PC99's TI screen display is somewhat distorted, looking too tall and not wide enough. This distortion is not noticeable on the more rectangular screen of laptops. New to stage 5 is the ability to display a more realistically proportioned TI screen, more like a real TI on a real TI 10-inch monitor.

At the top of my wish list for additional PC99 features is 80-column TI compatibility so that I can run 80-column Funnelweb. This may happen in the future, although Mike Wright makes no promises. In the configuration program that comes with you are given the options to "change CPU type" and "change VDP type." Your options for CPUs are 9900 and 9995. For VDP you can choose 9918A, 9938 and 9958.

These "changes" have no effect in stage 5. However, in the future we may have the option of a 9995 CPU (like the Geneve) and an 80-column display using the 9938 or 9958 VDP. I don't care about the 9995 or Geneve emulation, but it would be a big advance to be able to run existing TI 80-column software using an emulated 9938 or 9958 VDP.

PC99 stage 5 is commercial software available only from CaDD Electronics.

DOWNLOAD FILE CONVERTER
by Bruce Harrison

This will take any D/F128 text file downloaded from a network, correctly word wrap the text, and put the text into a D/V80 file readable by any TI word processor. The software is in assembly language and works quickly. As the D/F128 file is read in you can see it being processed on screen. You can also go in the other direction, converting a D/V80 text file into D/F128 format.

This comes on a SSSD disk with on-disk documentation and an offer of non-toll-free telephone support. It is another of Bruce Harrison's many public domain gifts to the TI community. Send me $1 and I will mail you the disk. I can also e-mail it for free in TI Files or PC99 format.

ACCESS

Charles Good (source of RXB and Download File Converter), P.O. Box 647, Venedocia, OH 45894; Phone (419) 667-3131; e-mail good.6@osu.edu.

CaDD Electronics (source of PC99 stage 5), 45 Centerville Dr., Salem, NH 03979; Phone (603) 893-1450; e-mail mjmw@xyvision.com.

Bruce Harrison (author of Download File Converter), 5705 40th Place, Hyattsville, MD 20781; Phone (301) 277-3467

USER NOTES

Squeezing out more memory

We're not sure who the author of this item is but we saw it in the newsletter of the West Penn 99'ers.

When you're running BASIC or Extended BASIC with a disk system, the TI operating system automatically sets aside memory to serve three concurrent open files. A minimum of 534 bytes of memory are taken up by general expansion overhead, plus 518 bytes for each of the three files that are opened by default for a total of

Continued on page 54

USER NOTES

Continued from page 53

about 2 kilobytes. If you know that you will have only one file open, you can close the other two open files and gain a little more than 1 kilobyte by entering the following command:

CALL FILES(1) <enter>
NEW <enter>

This sequence can be entered as a command only and can't be used as a program statement. When you're using TI BASIC with a disk system, this procedure is essential since TI BASIC addresses only 16K of RAM.

Things to do with CALL LOAD

The following was written by Woody Wilson of the Southern California Computer Group.

I was trying out a few CALL LOADS just now and ran into this one. Type in the following program and, before you run it, try to guess how it will appear on the screen.

```
90 CALL CLEAR
100 CALL INIT
110 CALL LOAD(-31873,10)
120 PRINT "THIS IS LINE ONE"
130 PRINT
140 PRINT "THIS IS LINE TWO"
150 PRINT
160 PRINT "THIS IS LINE THREE"
170 GOTO 170
```

Now run the program (use FCTN-4 to get out). In line 120, add a semicolon at the end of the line, after the quotation mark. Run the program again and obvserve how the line spacing changed.

CALL LOAD(-31888,63,255) followed by a NEW will disable your disk drives. But put it as a statement in a program without the NEW and see what happens. Type SIZE before and after. BYE will clear or you can use CALL LOAD(-31888,55,215) and NEW to free the drives.

Notice that you have 13928 bytes of stack free with the disk drives disabled but only 11840 bytes with the drives free and the console in the default size of CALL FILES(3). You will find that your drives use 534 bytes plus 518 for each open file. So, if you should use the maximum of nine open files, the stack will drop to 8732 bytes free.

You can load large Extended BASIC cassette files into memory and save by the following procedure:

1. CALL FILES(1) <enter>
2. NEW <enter>
3. Load program into memory
4. SAVE as DSK1.FILENAME,MERGE <enter>
5. BYE
6. Go to Extended BASIC and type MERGE DSK1.FILENAME. Note, this may take some time.
7. SAVE the program to disk

The program is now on disk in Internal/Variable format and will load and run normally.

Getting graphic with envelopes

Mary Phillips of the Ozark 99ers user group likes her Panasonic KXP-1124 dot-matrix printer because it lets her put graphics on envelopes. Here's how she does it.

Our Panasonic KXP-1124 has a feature I like very much. There is an opening for loading envelopes, postcards, etc. from the lower front side. Loaded from this location, materials don't have to roll around a platen. The same procedures (outlined below) can be performed with envelopes that have tractor-feed holes on the ends (and fan-fold postcards with tractor feed strips). These are available in office supply stores or by mail order.

1. Position the envelope so it is about one-quarter inch above the printhead.
2. Run the MAXRLE program, which loads pictures in any of four formats — D/F128 RLE, D/V80 RLE, Graphix, and TI-Artist _P format.
3. Load the picture you like, press (P)rint (the printer address appears in the lower left-hand corner of the screen).
4. Press Enter and the picture is printed, just like that!
5. Load TI-Writer or other text processing program and set the left margin to 45.
6. Type the name and address of your recipient and press FCTN-9 for the command line.
7. Reposition the envelope so the printhead is on the line where you want the name and address to be printed. Press PF (Print File) and Enter.

It's done, unless you want to get really fancy and color the picture. My favorite pictures for decorating envelopes are Leland Piper's TI-Artist pictures and Sr. Pat Taylor's collages. Of course, I like to use a graphic return address label on the back flap.

USER NOTES

CLASSIFIEDS

DISKS/BACK ISSUES