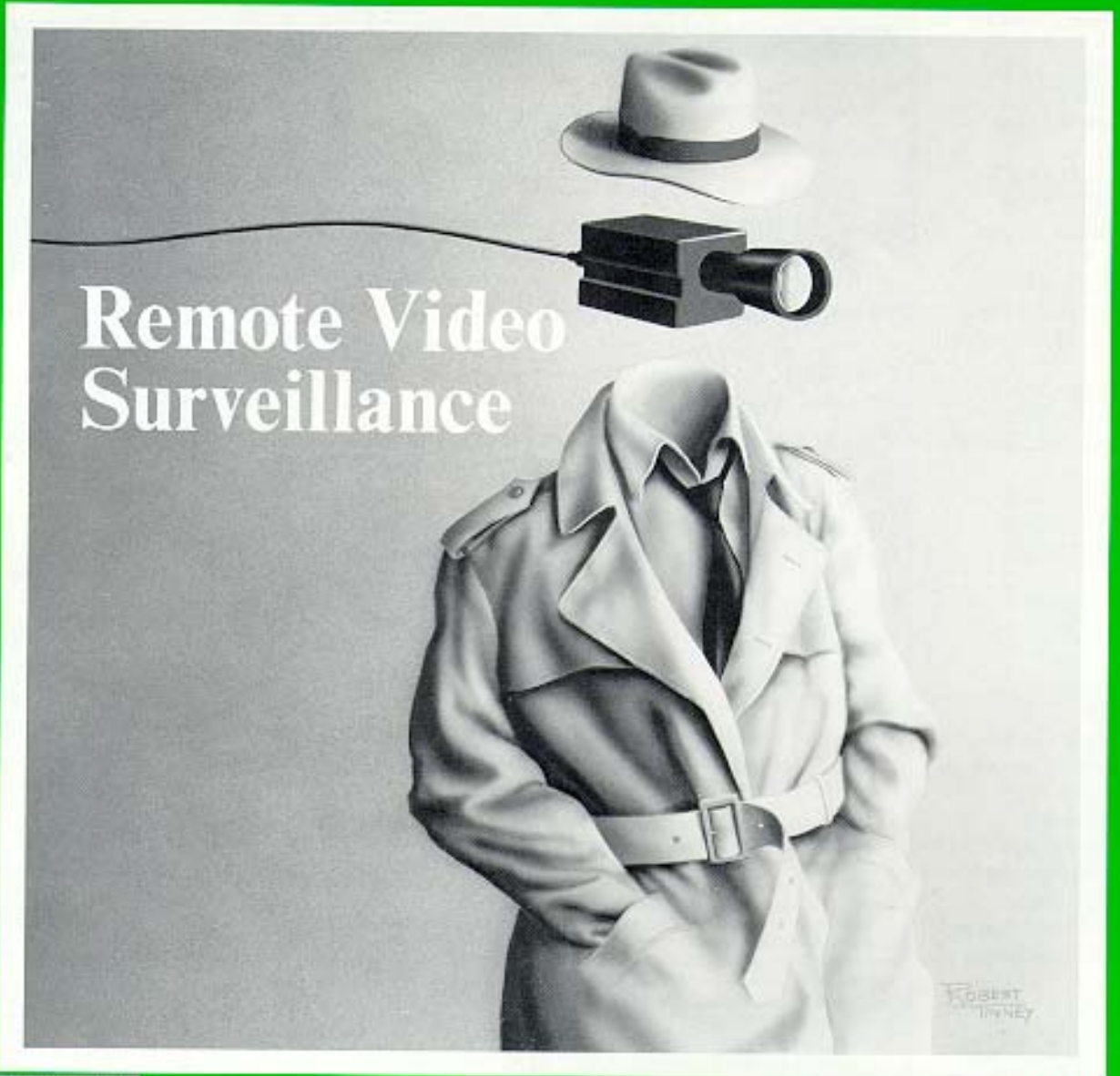
Circuit Cellar INK

Ctrl

THE COMPUTER APPLICATIONS JOURNAL



September/October 1988 - - Vol. 1, No. 5

$3.95

09

0 74470 75349 0

Ctrl

CIRCUIT CELLAR INK

THE COMPUTER APPLICATIONS JOURNAL

FEATURES

page 4

DEPARTMENTS

page 17

Circuit Cellar BBS - 24 Hrs. 300/1200/2400 bps, 8 bits, no parity, 1 stop bit, 203-871-1988

The schematics provided in Circuit Cellar INK are drawn using Schema from Omation Inc. All programs and schematics in Circuit Cellar INK have been carefully reviewed to ensure that their performance is in accordance with the specifications described, and programs are posted on the Circuit Cellar BBS for electronic transfer by subscribers.

Circuit Cellar INK makes no warranties and assumes no responsibility or liability of any kind for errors in these programs or schematics or for the consequences of any such errors. Furthermore, because of the possible variation in the quality and condition of materials and workmanship of reader-assembled projects, Circuit Cellar INK disclaims any responsibility for the safe and proper function of reader-assembled projects based upon or from plans, descriptions, or information published in Circuit Cellar INK.

CIRCUIT CELLAR INK (ISSN 0896-8985) is published bimonthly by Circuit Cellar Incorporated, 4 Park Street, Suite 20, Vernon, CT 06066 (203-875-2751). Second-class postage paid at Vernon, CT and additional offices. One year (6 issues) charter subscription rate U.S.A. and possessions $14.95, Canada $17.95, all other countries $26.95. All subscription orders payable in U.S. funds only, via international postal money order or check drawn on U.S. bank. Direct subscription orders to Circuit Cellar INK, Subscriptions, 12 Depot Sq., Peterborough, NH 03458-9909 or call (203) 875-2199
POSTMASTER: Please send address changes to Circuit Cellar INK, Circulation Dept., 12 Depot Square, Peterborough, NH 03458-9909.

Ctrl

EDITOR'S INK

Not the Same Old Song

When you design and create for a living -- whether it's process control applications, scientific applications software, commercial products, or magazines -- there is always the great temptation to fall back on tried and true formulas rather than look for truly innovative solutions. Deadline and budget pressures build up, creative frustration mounts, and before you know it you're singing "Why reinvent the wheel" instead of "I did it my way."

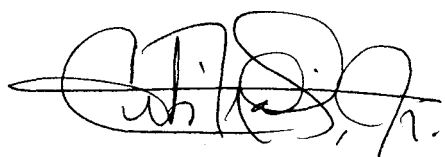
Now, don't get me wrong. There are many times when a proven solution is just the ticket for a particular problem. It's just sad to see people move from creative problem solving to rote formula recitation. We're not going to let that happen here at Circuit Cellar INK, and we're going to do our best not to let it happen to you.

This month, you'll notice that some of our articles (check out "ROVER") look a little different. We think that the new captions and layouts will help you get into the article a little more easily. In coming issues, you'll see some other differences, all of them aimed at making Circuit Cellar INK not only useful, but enjoyable.

You'll also notice the ads for our two new services. Many of you have told us that you would like to get the software mentioned in Circuit Cellar INK articles, but you can't log onto the Circuit Cellar BBS. Now, you can order all of the software for a particular issue on the Software Disk for the issue. Speaking of the BBS, if you can't log on, but you still want to get the solid information that's posted every day, now you can order the Circuit Cellar BBS On Disk, two months' worth of BBS messages on IBM PC disks. Both of these new services will get more information into your hands faster than ever before.

We're going to keep you on track by giving you articles that have solid, practical design information wrapped in a package that will let you look at problems from a fresh perspective. As an example, Steve's at his best with his latest project, beginning on page 4. Sure, other magazines could tell you how to build the hardware, but how many others can give you the itch to go out and do something creative *right now?*

It all boils down to just how good you want to be. Using the proven solution is fine now and then, but relying on the old and true too often leads to a complacent slide into mediocrity. If you want to be the best, then, it calls for striking out into new ways of thinking about problems, designs, and solutions. That's where you want to be, and you can count on us to be there with you, educating, stimulating, and entertaining.



Curtis Franklin, Jr.
Editor-in-Chief

Ctrl

READER I N K

Letters to the Editor

While looking forward to an article on the subject of autonomy and power supplies, I am dropping you word of what one user is getting by with and adding a word of thanks for your articles.

I live in an area which is sparsely populated, hilly, and subject to thunderstorms, the consequence being that long stretches of main distribution lines are exposed to every passing lightning flash. The protection system (for the electricity corporation's property) is very good, and it cuts out quickly and often. Some form of autonomous supply was necessary, especially after one motherboard received something up its parity chips, this from an unlucky lightning strike which arrived without any warning whatsoever.

Not being attracted to the idea of buying a machine to convert the main down to 12 volts, store it, and then convert it back up again, I decided to run my PC and an earlier TRS-80 directly off car batteries. The PC, of course, draws too much current for unassisted car batteries, so I ended up running off of battery chargers with large capacitances (in the form of batteries) attached.

Though I was hesitant at first, the system works well; I can leave it on all day without problems, which suggests that the power coming out is reasonably clean. Of course this system gives me no protection from either internal or external faults, but the savings are sufficient, compared to the price of a new motherboard, to get away without this protection. The actual cost of the system was little more than the PC power supply I didn't buy. With an AT or larger machine, this will no longer be true. I presume, though, that a well-protected and -regulated battery supply is a costly item. I am resigned to stopping work and disconnecting the mains when there is lightning in the offing. Having seen the state of the pair of MOVs which took the brunt of another stroke of lightning, I need some convincing that there is a really effective, sure obstacle that will bar lightning from reaching into those flimsy chips. 1 have wired in a pair of devices I found at Radio Shack labeled "field effect overvoltage protectors, DSAR 1 701 MA, part no. 270-9811, reaction time 1 microsecond" but I have no idea what they are worth.

I have wired in an automatic voltage-sensing relay which cuts off the chargers when the main battery reaches full voltage an hour or two after turning the system off.

This is not a letter that requires a reply, it just thanks you for your inexhaustible supply of good ideas and practical, viable projects. I just regret that I cannot build more of them and that I do not live within reasonable reach of the Circuit Cellar BBS.

John Negus--Bessas, France

Lightning and power outages are subjects near and dear to our hearts here at CCINK, and we're looking at some future projects that might be effective weapons in the war against blown-up equipment. We'd like to hear from any of our readers who have come up with creative solutions to power outages, line garbage, and lightning strikes.

--Editor

Sure, I'd like to subscribe to Circuit Cellar INK! I have one request to make, though: Could you occasionally expand your magazine by a page or two to discuss a uniquely neat piece of hardware you've included in a project but haven't discussed in the article?

A specific example might be the Shugart SA-300 single-sided 3.5-inch disk drive installed in the SB180 "computer in a lunchbox" article in the October 1985 issue of BYTE. I think that this would be a great drive for the experimenter to start out with. Unfortunately,

Shugart is no longer in business and, while Consumertronics' Disk Drive Tutorials give some clues for adjusting drives, they don't mention anything about the shorting pins used to configure them.

Maybe you (or your readers) can help.

Les Wenninger--Hubbards, Nova Scotia

We have a couple of ways to tell readers about "neat hardware" used in a project but not fully described in the article: Updates, such as the bottle rocket launcher in the July/August issue and, beginning in this issue, "From the Bench," conducted by Jeff Bachiochi. Jeff will be presenting practical advice on device selection and a number of interesting "small" circuits in the coming months. He'd like to hear from readers about what they want to see him cover. Technical questions should be addressed to Visible INK. We've turned your question on the Shugart drive over to the Circuit Cellar INK Research Staff.

--Editor

First, I'm writing to congratulate you on a fine magazine. The diversity and quality of articles is first-rate. I'm glad that you have included articles about trends in technology as well as projects and tutorials. I especially enjoyed the article on RISC vs CISC. It was the first article I had read that said something other than RISC is be-all and end-all of the future in computer design. I also really enjoy your editorials which have put into print feelings that I have had for a long time. Keep up the good work!

Second, I would like to reply to Mr. Dodge who wrote about displaying weather facsimile data by shortwave radio. Amateur radio operators have been doing this for some time. There are several products on the market which will receive FAX and RTTY and send the data to a computer via RS-232. These are listed below. All will receive both FAX and RTTY as well as packet radio. The MFJ interface will translate FAX data into an Epson-compatible graphics file that can be dumped to any Epson-compatible printer. Any good terminal program for the Amiga, such as Online!, VT100, or Comml.4 can be used to control the interface. I do not know of any commercial software that would convert these files to display on the Amiga screen, however the American Radio Relay League (ARRL) maintains a program exchange which may have some programs that can help. The format of Epson graphics files should not be hard to obtain, and using the capabilities of Amiga BASIC it should not be difficult to write a program to translate that file into something that can be displayed on the screen.

I hope this gives him something to start with. Once again, thanks for a great magazine.

Roy G. **Clay** III--New Orleans, LA

P.S. This letter is being prepared on an Amiga 2000. Hope to have some projects for the Amiga soon.

Thanks for the information on weather FAX. We share your hopes for some Amiga projects soon, and are also working on having Macintosh (including Mac II), Atari ST, and S-100 projects, as well as a continuing stream of projects that are built around stand-alone processors. Circuit Cellar INK is not biased for or against any computer system or processor, but we can't run articles we can't find. If anyone has put together applications for a system that they haven't yet seen in the pages of INK, they should call or write Curtis Franklin.

--Editor

Weather FAX Interfaces:

MFJ-1278
MFJ Enterprises, Inc.
P.O. Box 494
Miss. State, MS 39762
(601) 323-5869

PK-232
Advanced Electronic Applications
P.O. Box C- 1260
Lynnwood, WA 98036
(206) 775-7373

HK-232
Heath Company
P.O. Box 1288
Benton Harbor, MI 49022

ROVER

Remotely Operated Video-based Electronic Reconnaissance

by Steve Ciarcia & Ken Davidson

looked around the corner and peered through the doorway of the official-looking office. The walls were decorated haphazardly with town maps and site plans, some of which I recognized as years old. It was as if new layers of paper, like new generations of people, simply built upon the remnants of the past.

I always thought that death and taxes were the only undesirable things in life that were inevitable, but I've found a third -- local government bureaucracy. Unfortunately, it's practically impossible to live your entire life and not interact with government officials. Believe me, I tried! I use the mail for practically everything: dog licenses, permits, requests, and payments. Never before had I darkened the doorway of city hall, and never had I felt I was missing anything.

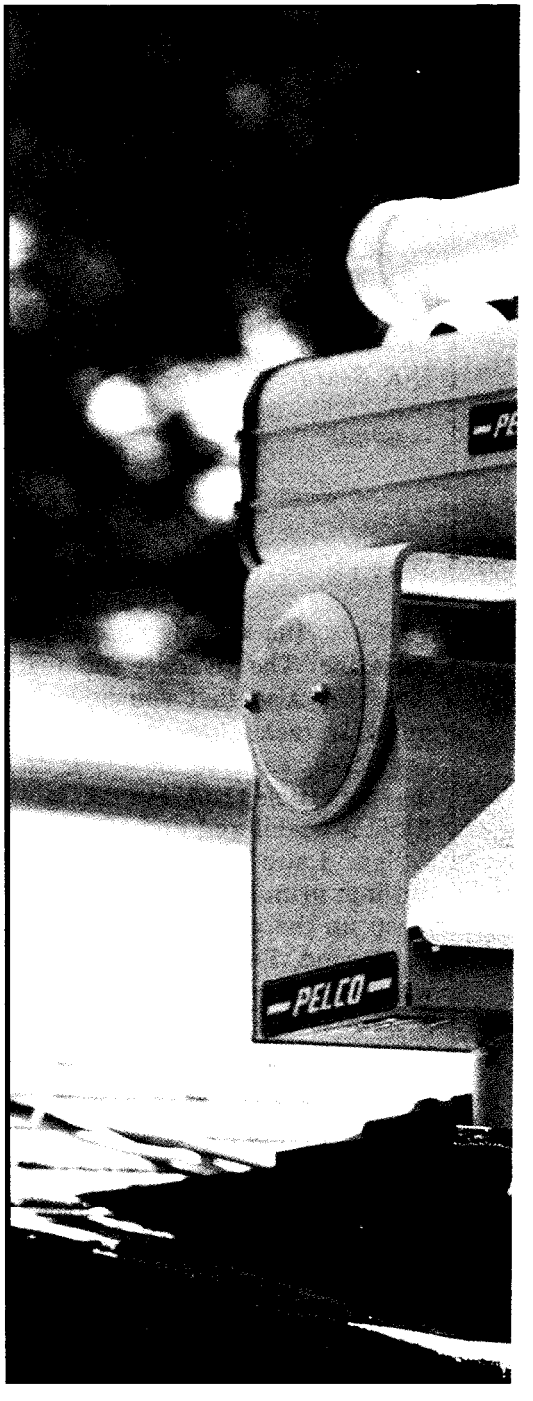
There were two desks in the room but only one was occupied. The unoccupied desk was piled high with a mountain of rolled and folded building plans. Considering the frictional variabilities, interlocking all this tubular and flat surface material into a semistable configuration was a stupendous engineering feat. The desk abutted the age-stained wall, and the pile sloped down in a straight line from a precarious height to the outer edge. It appeared that just one more site plan would bring the entire town architectural history down in a tremendous avalanche. I looked at the tightly rolled site plan in my hand and wondered, small towns being what there were, if I'd be remembered as "the one who upset the pile."

I'll never know why New England towns can't have air conditioned city halls. I'm sure that since most claimed to have been built around the time of the Mayflower, adding such amenities would deface the cultural significance of the structure. The people who worked in these buildings, as a result, were hardened pros. Like heat treating metal where you cycle the temperature often and to extreme limits, these New England city hall employees were a hard bunch. Hard to like, hard to take, and hard to satisfy.

In any case, here I was in a hallway that was about 98 degrees, standing in line waiting to get a building permit. The only consolation I had was that I was next in line and there were at least fifteen uncomfortable people behind me.

It was interesting to note a certain synchronization in mass body language. As if on cue, one group, which had been leaning against the wall on one leg, would shift to the other leg. At the same time, another group would cough, shuffle, or wipe their brows. For over an hour I observed this almost programmed pattern of shuffle, shuffle, lean right, lean left, sneeze and wipe. The only break in the pattern was when the line moved forward and two half-step shuffles were added.

It seemed like the previous person to enter the building inspector's office was taking an inordinately



long time. To alleviate the tension of waiting I swopped a few jokes with some of the others in line and nervously added, "What's the guy in front of us trying to do, build a hundred-unit condo?" A few of the more conservative town residents in the line immediately jumped to attention at hearing the word "condo" but I quickly waved my hand, "No! No! Just kidding, guys! No condos, here. Just joking." Still smarting a bit from coming so close to being lynched, I whispered to the person behind me, "What is this guy doing that takes so long? Here I am next in line and he's been in there almost an hour already."

The lady turned around, smiled like she knew she was dealing with a tenderfoot, and chuckled, "Oh nothing special. I think he just wants to put up a garage." Suddenly I had a very hollow sensation and I could feel a cold sweat coming on. She chuckled and continued, "Do you remember going to the DMV to get a car registered? Well, this place is like that. It takes about six trips just to figure out what paperwork is required. Unfortunately," she smiled, "it's a little hard to drag in a parcel of land or a building when there's some question about the way it looks!"

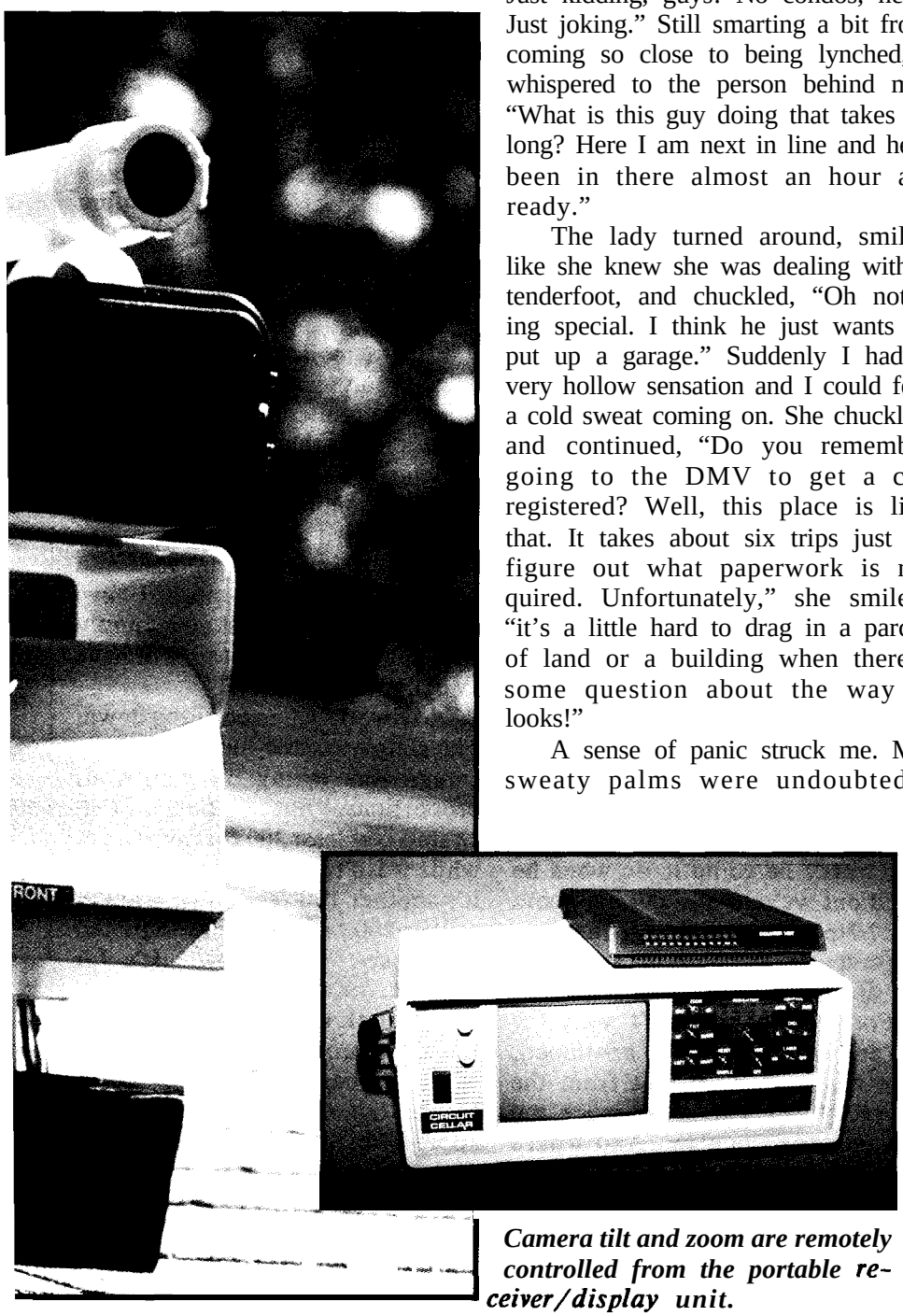
A sense of panic struck me. My sweaty palms were undoubtedly smearing all the carefully inked details of my new garage on the plot plan that I clenched tightly. I leaned back against the wall and closed my eyes. What had I gotten myself into? I was shocked back to reality with the loud call, "Next! Anyone out there? Next!" The woman behind me pushed me toward a Dutch door. As I opened the bottom half to enter the building inspector's office, the resident who had been in there for the past hour brushed past me, muttering loudly to himself, "Can't build a damned doghouse in this town without everybody's fingers in it!"

These were hardly words to inspire confidence but I entered the office, carefully avoided disturbing "Mount Paper Pile," and sat down in the chair next to the inspector's desk. I passed him the crumpled paperwork which detailed my project.

The inspector was about 45 years old with slightly graying hair and medium build; average in every way. He was dressed in a white shirt and brown trousers, both with areas shiny from extended contact with a desk chair. Scanning down from a tie that contained most of a week's menu, I noticed his heavily soiled shoes. Obviously this man did field work as well. I was beginning to hypothesize about his health and family situation when he looked up from studying the site plan and interrupted me. "Mr. Ciarcia, should I presume from this that you are building another house next to the present one? You know we can't let you do that!"

"No, no, Mr. Wright. That is the plan for a four-car garage. It just happens to be as big as a house."

It was sacrilege giving this job to anyone with a name sounding like "right." Admittedly I was better at drawing electronic sche-



Camera tilt and zoom are remotely controlled from the portable receiver/display unit.

matics than detailing the location and dimensions of a garage on a real-estate site plan. In the same breath I continued, "I've had about 40 truckloads of fill brought in, the hole for the foundation is dug, and the lumber has been delivered. All I need now, according to my carpenter, is for you to sign off on this building permit so that we can lay the forms and start building! We could start this afternoon!"

He looked up at me. I caught a glint in his eye as he ignored what I had said. Instead, in an expression of power which he obviously relished, he extended his forefinger and sharply jabbed at the drawing. "If you meet all the requirements I'll grant you a permit, but not until. The regulations are very straightforward but you'd be surprised how many people come in here and try to put one over on me. What's this?"

I answered automatically, "That's an existing two-car garage."

It was like a bell went off. He sat up quickly and questioned in an official-like manner, "Six cars?"

I'm sure he felt he had to protect the community from people who start automobile junkyards in the neighborhood but this was hardly the case. The Circuit Cellar was getting pretty full and I needed some place to put a real workshop and transfer some of the junk. As I prepared to defend my actions, I suddenly thought better of it. If this guy was upset about six cars, what's he going to do when he found out we're really talking about eight? He hadn't discovered the two-car garage in the house yet and I was sure not going to tell him ...

Fortunately he didn't wait for that answer. Instead he continued, "You have to be 25 feet from the property line. This looks much too close on your site plan. And what's down here in this area?"

I leaned forward to see where he was pointing and replied, "Oh, it's at least 25 feet. Take my word for it. Besides," I chuckled, "I own the land on the other side of that property line too." I smiled while thinking "gotcha."

"Makes no difference how much land you own, Mr. Ciarcia. Since that other piece of land is legally separate, if you are closer than 25 feet you'll have to go before the zoning board and petition for a variance. The next available slot in their agenda is in about three months, I think ... or, was it four? Before we worry any more about that, however, what's over here?" He pointed to the same area he previously designated.

I chuckled nervously, hoping that we could forget about zoning board meetings. "It's a swamp or something like that ... but, don't worry. I won't try to build a garage on it."

"Darn tootin' you won't!" he voiced authoritatively. "That would be wetlands and you can't build closer than 100 feet from wetlands without going before the wetlands commission and then the zoning board of appeals. The dimensions indicated in your site plan aren't clear and unless I have concrete evidence that you comply with zoning regulations I will have to deny your permit!*'

My world came crashing down. The simple task of building a little garage was taking on monstrous proportions. "But ... but ... that swamp is at least 200 feet away and, while it isn't indicated on the plan, it's another 100 feet down a hill as well as 200 feet away!" I implored vigorously. "Trust me! It's far away!"

"I think the solution is for you to have a professional survey done of the property that clearly marks the location of the new building. I appreciate that you have already dug for the foundation, but unless I have personal knowledge that your building location is proper or you have an official survey map which indicates same, I have to

deny your permit."

He was about to yell "Next!" when I feverishly said, "All you have to do is come out and see the garage site! You'll see that all my dimensions are in order and that any wetlands are way off." That seemed reasonable to me, and the least a public servant could do.

"Sorry, but unless you have a building permit, I am not authorized to visit the building site," he stated matter-of-factly. I was depressed. If he saw that everything was copacetic, I could get a permit. But without a permit he couldn't go see it, right?

Suddenly, I jumped up and yelled, "Wait, Mr. Wright, I've got the perfect answer! ROVER is in my car!" I bolted for the door before he could answer. I knew if I waited for his reply it would have been "Next!" and I'd be at the end of the line. I heard him yell, "What the hell has a dog got to do with ..."

I didn't hear the rest as I rushed past the line of people and dashed for my car. Fortunately, New England town halls are small and the car was parked only a few hundred feet from the building inspector's office. Before he could challenge this obvious miscarriage of protocol, I was back in his office panting from exhaustion.

In my hand was a box that looked like a portable computer. The inspector seemed surprised, but not shocked. Apparently, modern technology was within his scope of understanding and I wouldn't have to be concerned about being burned as a witch for what I was about to demonstrate. I plopped the box on top of a pile of site plans on the corner of his desk, disconnected the phone line from his telephone and plugged it into the unit. He was used to people piling paper in his office, but not messing with his desk or phone. He jumped up and remarked loudly, "I really don't have time for this right now! What is this thing anyway? Is it dangerous?"

Walking down the driveway to my house was an alien!

As I flipped on the power and pressed "CALL A," I answered, "This is ROVER! ... Remotely Operated Video-based Electronic Reconnaissance! Think of ROVER as a spy in the sky that's dedicated to my house." I had his attention but immediately knew from his expression that "spy" was not a word I should have used.

"ROVER is an electronic device that communicates with a video camera at my house. Using ROVER I can show you on a video screen exactly where the new garage will go and where the boundaries in question are. You won't have to visit my house, I'll bring it to you live and in black and white." (I wish I could have said "live and in color." I was really on a roll).

I continued describing ROVER, explaining how it used the telephone line to transmit digitized video and that the pictures we were viewing were snapshots of the real-time activity at my house. I pointed the camera at the house and oriented him to the location on the site plan. Eventually I panned the camera to a position where he could see both the foundation hole that had been dug and the stone boundary wall which marked the property line. To my amazement, he agreed that the new garage did indeed "appear" to be considerably more that 25 feet from the wall. As skeptical as Mr. Wright had initially appeared, he recognized that ROVER was indeed filling in some blanks in my permit application. I won't say that he wanted one installed in city hall, but his was obviously a case where once he got over using a laser transit he wasn't surprised to see anything.

I was patting myself on the back for this coup over the bureaucracy as town employees from the fire department, EPA, and department of public works entered the building inspector's office to see why their shared party-line phone suddenly sounded like a lot of static. Soon, even these three were mesmerized by ROVER, The concept of viewing a physical work site without having to leave their desk appealed to them.

After a while I didn't even have to operate ROVER as one of the faster learners among them took over the controls and continued panning, tilting, and zooming around my property. I smiled as I watched the four men leaning over and intently watching the video display. Suddenly one of them yelled, "Hey, what was that!? Something moved! Pan back over there."

The suited figure on my deck was ***bending forward and holding something that looked like a gun.***

I peered over the top of the huddled group to see what had attracted their attention. ROVER was still painting a high-resolution picture on the display so I offered a plausible explanation. "Gentlemen, ROVER was designed for security applications and I am trying to get a building put up. With your help of course.*' I motioned toward the building inspector. "Perhaps one of the carpenters has arrived."

ROVER had finished sending the image, and walking down my driveway toward the house was an alien! Or should I say a humanoid that looked like an alien! Well, something in a space suit anyway. They looked at me; I looked at them; and we all looked back at ROVER which was now painting another closer image. I, too, was curious and joined the crush to view the latest picture.

In full view was a person who looked like he was wearing a white space suit! I mean, it had a helmet, air hoses, and all the stuff that made it look like something from NASA. I said, almost jokingly, "A guy in a space suit at my house?"

"Turn the camera and follow that guy some more!" the EPA official ordered in unison with the fire department guy. "That's no space suit! That's an environmental hazard suit! We use those when we inspect hazardous waste dumps!"

The next picture showed the suited form on my deck bending forward and holding something that look like a gun. The EPA guy continued in an agitated voice, "Those are spraying tanks he's wearing. Whatever he's spraying that requires that kind of suit is bad news. I presume you have a permit for whatever this is, Mr. Ciarcia?"

"Really, sir. I haven't the slightest idea what is going on! I don't know what that person is doing! No, I haven't got a permit!"

"No permit to spray toxic compounds? Do you realize what kind of trouble you're in if any of that is found to be polluting the neighborhood, Mr. Ciarcia!?"

Truly I didn't know what was going on. I just sat there dumbfounded and emotionally exhausted as the next picture was painted on ROVER's screen.

This time the suited man was standing next to a truck that I instantly recognized. Suddenly everything clicked into place. I exclaimed, "That's Bob Borg! He's a friend of mine!"

"This is all a joke! I know what he's doing now! Oh boy, Bob, you really had me going for a while there." I nervously voiced the latter at ROVER's screen as if to talk directly to the suited man. The four town officials looked at me strangely.

One of them said, "I thought you didn't know who that was using toxic chemicals on your property?"

"It's Bob Borg! He's playing a joke!"

"It looks like he's spraying toxic chemicals, Mr. Ciarcia. There's nothing here to joke about that I can see," was the sarcastic reply.

"No, no. He's not spraying anything! He's just posing for my alarm system!" You couldn't believe how ridiculous such a statement sounded, but I was frantically trying to explain my way out of a predicament.

Continuing, "Bob knows I have a security system with video cameras and a VCR. The last time he was at my house killing some ants when I was away, I sent him a picture showing him that I 'caught' him creeping around my house. He laughed and said that one day he'd visit my house again and really shake things up."

"So you admit, Mr. Ciarcia, that this man is here spreading toxic chemicals with your complete knowledge and authority!? And further, you're obviously doing this in violation of the law?"

These guys weren't listening and I wasn't making any headway. Again I implored, "It's a joke. Bob is a friend who's just playing a joke on me. It was all a dare!" Without voicing it, I thought to myself that Bob was a friend all right, but I'd strangle the SOB later for his poor choice of timing.

"Get a load of this!" The EPA guy called the others to look at ROVER which was in the process of displaying the contents of Bob's truck. "Whoever this guy is he ain't carrying cord wood back there."

"I told you his name is Bob." I was interrupted abruptly by the fire official who was communicating with someone outside the office via a walkie-talkie attached to his belt.

The fireman interjected loudly. "I ran a make on the truck license plate through the state police. His name is Bob Borg. He's some kind of special exterminator that they call for the really 'bad bugs'! Sounds like we're dealing with a character who has access to all kinds of nasty stuff. However, if we're dealing with only a single truckload of chemicals that aren't too toxic, perhaps the pollution threat won't have to involve the state boys."

"I wouldn't count on that if I were you. Every time we nail one of these waste dumpers, it turns out they've just buried about 50 drums of chemicals under a pile of landfill, or they try to build a building or a garage over it," the EPA official added.

They talked among themselves as if I weren't there and continued to study the pictures from ROVER until the EPA person again said, "Oh, oh. Get a load of this. There's a big gas bottle on the back of the truck with a poison label. The only gas bottle I know of that warrants such a bold notice is Trichloro-Dimethyl-Benzimidanide or Tetrachlorosobromide. Both are so toxic that we're probably going to have to dip into the Superfund to clean up this mess."

"OK, guys. It's my opinion that we declare a class-one alert and get our people over there right now! Obviously Mr. Ciarcia and his friend Borg have been disposing of chemicals for some time now and we were fortunate enough to uncover their activities."

"But, but, I tell you it's a joke."

'Whoever this guy is, he ain't carrying cordwood bock there."

I cried out as two constables entered the room and asked that I go with them. "I just wanted to build a garage ..."

As ROVER and I were being led away past the intensely curious crowd, the building inspector stuck his head out the door and yelled, "Permit denied! Try and put one over on me will ya!"

ROVER stands for Remotely Operated Video-based Electronic Reconnaissance (we would have called it "BIG BROTHER" but we couldn't think of an appropriate expansion of the acronym). ROVER is an electronic surveillance system which allows digitized video pictures and pertinent status information to be transmitted and viewed anywhere in the world via telephone.

Similar in concept to a video telephone which has a separate video transmitter and display receiver, ROVER has a video camera which transmits a digitized picture via modem to a receiver. ROVER is more than a straight video link, however. The transmitter portion of ROVER is a complete microcomputer-based data acquisition and control system. Rather than a simple fixed-focus camera, video is provided by a low-light-level auto-iris CCD camera with motorized pan, tilt, zoom, and focus. Camera position. zoom, and so on, are controlled by discrete contact closure outputs which are controlled from the re-

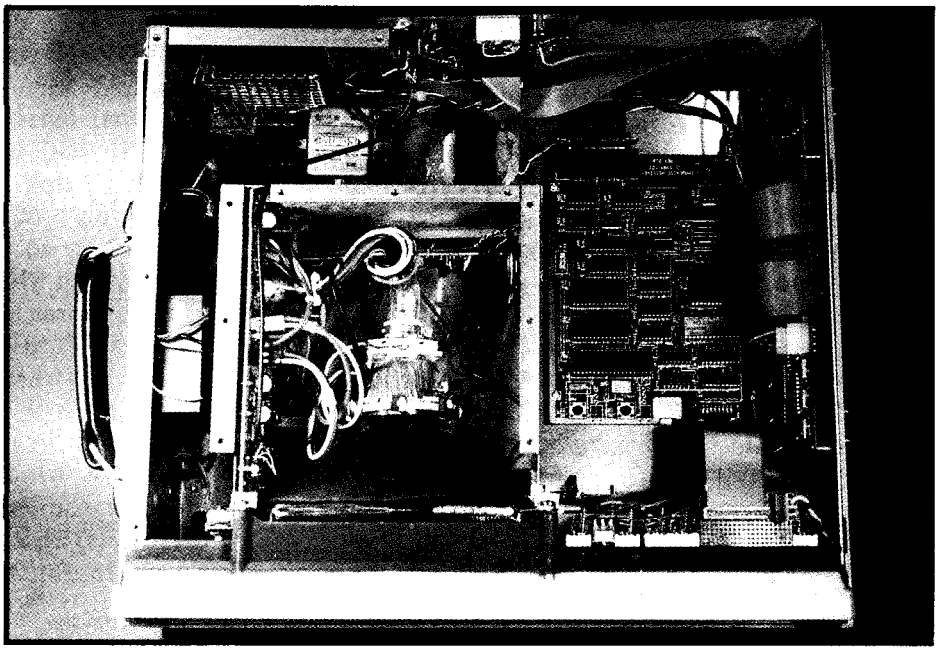
ceiving end. Similarly, ROVER passes discrete input data from various contact closures and switches to the receiver.

The ROVER receiver/display unit looks like a small suitcase computer and is designed to be portable. The receiver contains a 9600-bps modem and a digitized video display board connected to a controlling single-board computer. An array of switches initiates the telephone call to the transmitter, controls the various camera attributes, and sets the receiver's display resolution.

When I want to check the house for any reason, I merely plug ROVER into the phone line and turn it on. After initialization, I press "Call A" and the ROVER receiver (on my desk) autodials the ROVER video transmitter (on the garage roof at my house) and starts painting a picture of what it "sees" on the screen. The camera position is changed by pressing the various pan, tilt, and zoom buttons. Pictures are displayed every 5 to 50 seconds (at 9600 bps) depending upon the resolution selected.

Since my house is a local telephone call from my office at CC INK, I often leave ROVER on all day, providing a real-time view of events in my driveway. I have watched the UPS truck back over the shrubbery and salesmen looking in the windows trying to find the "lady of the house" (they really do a double-take when they notice the camera "following*' them).

Regardless of my humorous use, ROVER is real and it is off-the-shelf. The basic ingredients of ROVER are the ImageWise serial grayscale video digitizer/display system and the HD64180-based BCCi80 control computer. Both of these units were presented as Ciarcia's Circuit Cellar projects (see source box at conclusion) and are fully documented designs which can be easily built from the article descriptions and reader-support software on the Circuit Cellar BBS (they are also available as kits or assembled units).



Building a portable ROVER receiver consisted of putting a BCC180 and an ImageWise receiver/display into a surplus transportable computer case and gluing a USRobotics Courier HST modem to the outside.

I refer you to these original articles for video basics and the hardware architecture of ImageWise and the BCC180. To build ROVER, Ken and I approached it as if we simply took a pair of ImageWise DT01/DR01 boards and a couple BCC180s and did a project as end users. ROVER, at that point, just involved plugging in a couple of new ROMs, slapping on a camera, a few switches, and specialized software. Of course, getting from here (with off-the-shelf hardware) to there (configuring it into ROVER) is what this article is all about.

The ImageWise system consists of two parts: the digitizer/transmitter (DT01) and the receiver/display (DR01). The digitizer portion accepts a standard NTSC video input, digitizes a picture in 1/60th of a second at a resolution of 256x244 pixels with 64 levels of gray, and sends out the digitized picture serially over a standard RS-232 connection. The display board has an RS-232 serial input through which digitized picture information enters the board. The board then stores that picture information in its on-board memory and recreates a standard NTSC video output signal at the same resolution of 256x244 pixels and 64 levels of gray.

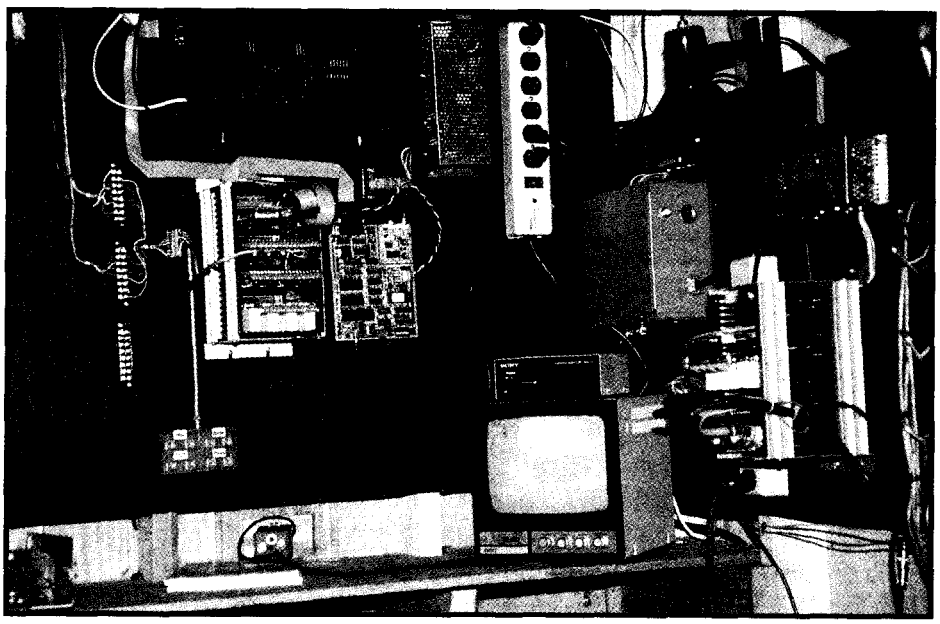
The ImageWise boards are designed to work together with no outside help. The display board sends a picture request to the digitizer and the digitizer obliges with a picture. Switch settings on the display board allow the operator to select one of three different resolutions (256x244, 128x 122, and 64x61) and the time between pictures (continuous, 4 seconds, 8 sec-

onds, or on the push of a button).

The ImageWise system was designed with remote surveillance in mind. The serial ports on both the digitizer/transmitter and receiver/display are set up in a DTE configuration so each can be plugged directly into a modem. While neither end has the necessary software support to dial a modem, neither end needs a host computer to be used in a stand-alone, free-running fashion.

The beauty of the system, however, stems from the serial interface used by both boards. Rather than connect the boards directly together, it's very easy to connect them to a host computer. The host computer can then request a picture from the digitizer and save it in main memory or a file. The picture can be manipulated, modified, and saved on the host computer. When the operator wants to see the picture at some later time, he can send it out through the host computer's serial port to the display board and get an exact duplicate of the original.

As mentioned before, it's a relatively straightforward task to connect the ImageWise boards to modems and communicate pictures between the two boards without outside help. Such help is only needed when dialing the originating modem (most Hayes-type modems will automatically answer the phone on the answer side of the connection). However, when laying out the specifications and features of ROVER, we wanted to be able to do more than just make a connection by hand and see fixed-resolution pictures from a stationary camera. We wanted to be able to move the camera around, zoom in and out, turn a targeting laser on and off (just wait for that article!), and receive status information back from my existing home control alarm system. In addition, we wanted a choice of resolutions so we could trade off transmission time with resolution depending on the situation.



The ROVER transmitter consists of a camera (see opening photograph), USRobotics Courier HST, ImageWise digitizer, and BCC180 computer with relay output and optoisolated I/O expansion boards.

The BCC180 turned out to be ideally suited to the task. (While the BCC52 is relatively inexpensive and has enough speed to handle the role of "picture traffic cop," it lacks a second full-duplex serial port and requires an expansion board to get more than 48K of memory.) The BCC180 has two full-duplex serial ports, 128K of EPROM or static RAM for program storage, plus 256K of dynamic RAM for picture storage and manipulation. In addition, it has six 8-bit parallel ports for use as switch and contact-closure inputs or indicator and relay outputs. Finally, it can be plugged into a backplane so expansion boards containing relays and optoisolators can be easily added.

As you might already know, I have had more equipment blown up by lightning than you can shake a lightning rod at. If you saw my article in issue #1 of INK ("Motion-Triggered Video Camera Multiplexer"), you know to what lengths I'll go to isolate discrete inputs and outputs so transients caused by lightning only affect isolated portions of the system. On the digitizer end of ROVER, I wanted to have various inputs to the system that indicated when the house alarm was on, the HCS was active, plus the states of various contact closures. For outputs, I needed to be able to control the power to the pan, tilt, and zoom motors and be able to turn the laser and VCR on and off.

I could very easily have connected all these inputs and outputs directly to the BCC 180's parallel ports (drive transistors for the outputs and clamping diodes for the inputs), but that wouldn't have provided sufficient protection as far as I'm concerned. Instead, we designed ROVER more along the lines of an industrial control system.

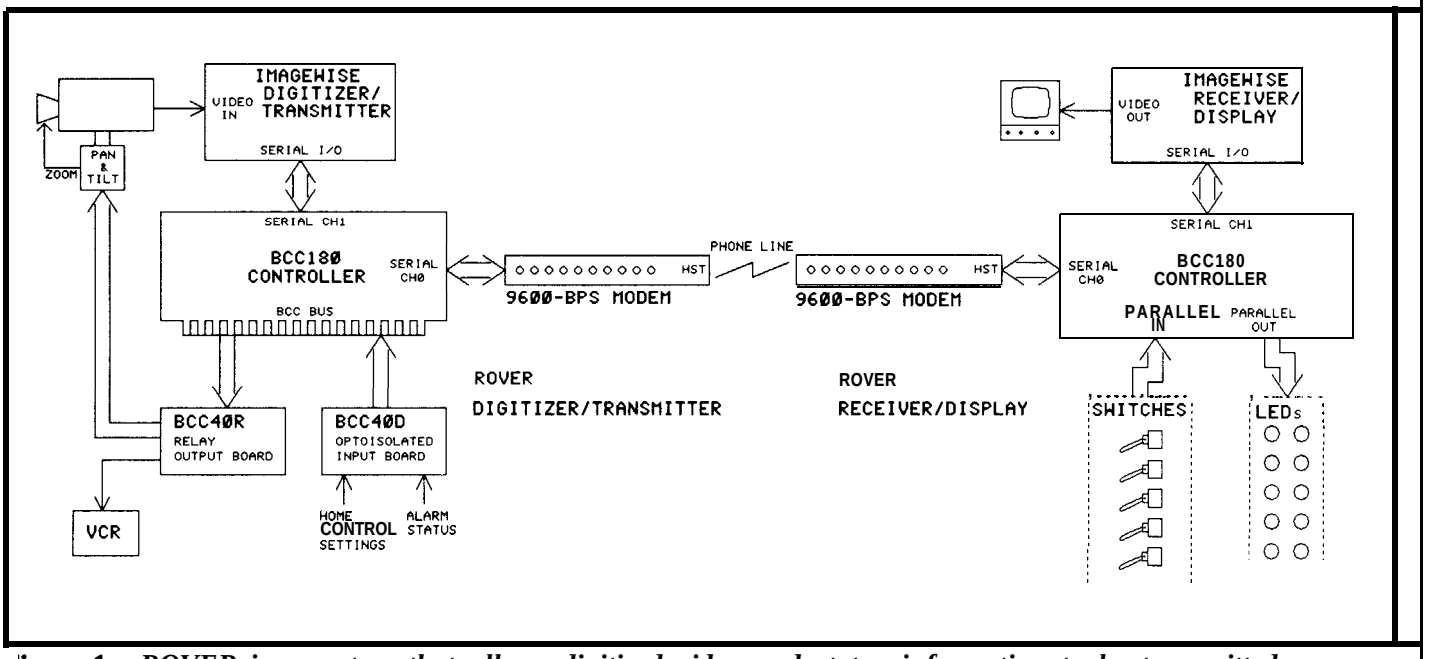


Figure 1 – ***ROVER is a system that allows digitized video and status information to be transmitted anywhere in the world via telephone.***

Figure 1 is a detailed block diagram of the ROVER system. The digitizer/transmitter portion of ROVER is mounted on the wall in the garage directly under the pan/tilt camera unit pictured in the opening photograph, and the receiver/display unit is a portable unit resembling a Compaq computer.

The ROVER transmitter consists of a camera, USRobotics Courier HST modem, ImageWise digitizer, and BCC180 control computer with relay output and optoisolated I/O expansion boards. As I previously mentioned, no computer or expansion boards are required for a ROVER transmitter configured for auto answer/fixed-camera operation). The BCC40R relay output board controls the 24-VAC power to the motorized pan and tilt and +5V/-5V to the camera focus and zoom.

The Pelco PT 175-24P all-weather pan and tilt uses separate connections for each control input: pan left, pan right, tilt up, and tilt down. To pan or tilt the camera, we momentarily close the appropriate relay (for the amount of time the switch is pressed on ROVER's receiver) which applies 24 VAC to the winding. By making these connections through DPDT relays that "break before make," we can also physically guarantee that the computer can't accidentally energize opposing directions at the same time. Compound motion such as down and right or up and left are possible, however.

The camera I used is a Sony SSC-D5 with a Cosmicar SCL 0813 AI auto iris motorized zoom lens. Even though the camera is CCD and not susceptible to image burn-in if accidently focused on bright lights, an iris is necessary to optimize the exposure for a given lighting condition. The iris fuction is automatic while the 12.5mm-to- 75mm zoom lens and focus controls are "manual." Unlike the pan/tilt, however, this unit relies on changing polarity on a single control line rather than providing pairs of separate function lines. Applying +5V to each control line zooms or focuses "out" and -5V zooms or focuses "in." In addition to the camera controls, relay outputs are provided to turn on a video tape recorder (recording everything from the camera rather than just the digitized snapshots) and a laser. While I haven't quite decided what to do with the laser, it's there. It is the thing that looks like a sighting scope mounted to the top of the camera housing.

Much of the added complexity of my ROVER configuration is due to its full-duplex data transfer. In addition to the relay board for the camera controls, ROVER utilizes a BCC40D optoisolated I/O board to monitor various 12-VDC and 115-VAC signal inputs and light the corresponding indicators on ROVER's receiver. Presently these signals consist of status bits from the home control system, AC power, and house alarm systems. Because of the line lengths and

voltages involved, optoisolated inputs were mandatory.

The ROVER receiver/display is a transportable unit with built-in video monitor. The case, without CPU board but with a nice 6" 12-VDC composite video monitor and power supply, was purchased (dirt cheap) at the Trenton Computer Festival. It was originally intended for a portable Apple clone from Franklin Computer which was never marketed in volume. Perhaps one reason is the overly expensive but ultimately beneficial construction. To meet FCC emission rules, there is a metal "Faraday cage" shell inside the attractive plastic enclosure. All the electronics built inside this shell are protected from causing as well as receiving electrical interference from other equipment. My "prototype" ROVER probably has better EM1 characteristics than many commerical computers as a result.

Building a portable ROVER receiver into this case merely consisted of jamming in a BCC180 board and an ImageWise receiver/display. To make room for all this odd-shaped hardware, I eliminated some of the existing partitions and substituted a smaller power supply. The Courier HST modem was "velcroed" to the top cover.

The only laborious task was constructing the front panel and indicator lights. All these switch inputs and LED outputs were connected to the BCC180's six on-board parallel I/O ports (the LEDs were externally buffered). To add a high-tech look to ROVER, I mounted the switches to a translucent plastic panel and back-lit the nomenclature with red LED indicators. Similarly, the lower status panel uses rear-projecting indicator lights. The bar-style LEDs are imprinted directly with the various status symbols and mounted behind the plastic panel. The symbols are only seen through the plastic when they are lit.

My part of building ROVER was easy: a couple off-the-shelf boards, a couple of trips up the roof, and some monotonous wiring. The real task of making ROVER into the video reconnaissance unit we imagined was left to Ken. Next issue we'll explain the fancy footwork behind ROVER's software.

The authors would like to thank Bob Borg for allowing us to take his photograph, use his name, and destroy his professional image.

The complete source code for ROVER is available on disk (see page 44) or can be downloaded from the Circuit Cellar BBS. For a reprint of the ImageWise Serial Video Digitizer articles presented by Steve Ciarcia in May-June '87 BYTE, order DT/DR Reprint. Send $3 postage and handling.

Add $10 for the two ImageWise assembly and user's manuals.

For a reprint of the BCC180 Multitasking Computer/Controller articles presented by Steve Ciarcia in Jan.-Mar. '88 BYTE, order BCC180 Reprint. Send $4 postage and handling.

Add $6 for the BCC180 assembly and user's manual.

Send all manual and reprint requests (mail order only, please) to:

Circuit Cellar INK
Reprints
P.O. Box 772
Vernon, CT 06066

Letters to the INK Research Staff

My name is M. Tarvirdi Zadeh. I am an Industrial Engineer and am interested in computing and electronics. I have some experience in computer interfacing. I designed and made a general-purpose interface for ZX81, ZXspectrum, and C-64. I am working with Amstrad PC1512 (XT Clone 8086 8 MHz) now. I want to do the same (design and make a general-purpose I/O board for control) for my XT clone. Because of the many regulations and limitations, I do not have any special documents to help me. I borrowed an IBM XT hardware technical reference from my friend. After much searching, I haven't been able to find any free addresses for the expansion board. I decided to use the I/O address 300-31F hex which are used for prototype cards.

I have three questions that I hope you can help me with.

1. Which I/O addresses are free to use for expansion boards?

2. Can my CPU (8086 8 MHz) or IBM XT CPU (8088 4.77 MHz) support I/O addresses higher than 64K. And which segment I should use for my addresses (300-31F hex)?

3. How can I flag my CPU that there is an active board plugged in and how is it recognized?

I hope that you can answer me personally. I am looking forward to hearing from you as soon as possible.

Thanking you in advance for your cooperation and best wishes for you.

M. Tarvirdi Zadeh
Tehran, Iran

Apparently your past experience with computer interfacing has been limited to memory-mapped I/O. The 8088/8086 processors use I/O ports to communicate with the outside world, which requires a little different approach to interfacing. The IBM PC expansion bus (called I/O channel by IBM) is also a little different than buses you may be more familiar with.

The 8088 can address up to 65536 I/O ports, but IBM implemented I/O channel addressing for only 512 ports, starting at address 200h, and ending at 3FFh. Ports used by IBM are listed in chapter 1 of your XT Hardware Technical Reference (probably on page 1-10). There are several blocks of unused addresses in the range from 200h to 3FFh, mostly reserved by IBM for future use. You can use these for your own projects, but there may sometimes be conflicts with new equipment you might install later. Using 300h to 31Fh should be safe.

Since I/O addresses are not mapped into memory address space you don't use a segment address, and no memory addresses are used up. The processor addresses memory through MOV instructions which transfer data from memory to registers, registers to memory, and so on, where addresses are always in a segment + offset ***form. I/O ports are accessed by means of INP or OUTP instructions which transfer data between registers and ports directly, and require no segment address.***

It is not necessary to tell the processor that there is an active board plugged into an expansion slot. All slots on PCs and all except the short slot on XTs are the same. No special addressing is required, and all memory and I/O ports in the range given above are accessible to boards installed in the expansion slots. You simply write your software to use the features of your board and plug it into any slot.

-- INK Research Staff

We need to obtain a mathematics library for use with the AMD 95 11 /Intel 823 1 A 8-bit coprocessor. We want it to be a linkable replacement or substitute for the FORTRAN library supplied with the old Microsoft

FORTRAN-80 compiler for the 8080/Z80 processors running under CP/M-80. Is such a library available anywhere?

We have a number of 8-bit CP/M computers in current use, each dedicated to a single laboratory instrument. Installation of the coprocessor hardware would be no problem, but our programs need to be able to access it. We need to do fast Fourier transforms in fractions of a second rather than a few seconds.

C. Sherman **Gromme**
Menlo Park, CA

In reference to your letter asking about a mathematics library for the 8080/Z80 processor under CP/M: I dug through my files and I am afraid I did not come up with much. I found an old address of a company which may or may not still be around or still have these routines available. They advertised complete 951 I driver software including reentrant conversion routines but it was from a very old advertisement.

Their address is :

Sorensen Software
Raiffeisenstr.1,
6104 Seeheim I, West Germany
Tel. 06257/83707

Another suggestion would be to contact suppliers of CP/M public domain software. The following listings I found in a recent issue of the Computer Shopper.

Public Domain Users Group
Box 1442-C2
Orangepark, FL 32067
(send SASE and computer type for info)

CP/M User Group Library -- 100 disks $150
S. Mills, 4615 Orleans Blvd.
Jefferson, LA 70121
(504) 733-9611

Another suggestion would be to place an advertisement in the Computer Shopper ***or perhaps some other publication.***

-- INK Research Staff

I am building a speech synthesizer for my Atari 800XL. I have two Radio Shack chips, CTS256A-AL2 and SP0256-AL2. I see on the enclosed panel that upon power-up or use of the hardware reset, the system speaks "OK." It doesn't. I hear two "buzzings" but they are not the same.

Please, Steve, how do I connect this to the Atari and what must I do on the keyboard to enter the data? It is not possible for me to obtain a 3.12-MHz crystal. I am using 3.2768 MHz. What about that replacement?

All my thanks in advance.

Hubert Lemort
Belgium

The Atari 800XL is an interesting computer with some severe limitations as far as interfacing to the outside world. The port provided is designed for synchronous serial I/O only. The speech chip set will accept asynchronous serial input, or parallel format with strobed handshaking. The best approach is an adapter of some sort such as an RS-232C or Centronics attachment. One source that is advertising an extensive number of Atari 8-bit accessories is: Electronic Dimensions, P.O. Box 1846, San Luis Obispo, CA 93406 USA. Using either an RS-232C serial or a Centronics parallel port, a properly wired synthesizer assembly can be addressed and treated as if it were a printer or other output-only peripheral.

Your use of the 3.2768-MHz crystal instead of the specified 3.12 MHz is unlikely to be a source of the problem you are experiencing on start-up. In this country it is not uncommon to find a 3.58-MHz color burst crystal used instead of the much rarer 3.12-MHz crystal specified.

You should double check all wiring, solder joints, and layout. Check power supply voltages at each IC and observe proper polarity on all polarized capacitors. Be sure your speaker is functioning properly: if it isn't, replace it. Trouble-shooting circuits is seldom fun, but is a necessary part of design and construction.

In order to send text to the synthesizer, it must be directed to the output port. Access instructions should be supplied by the maker of the interface converter for RS-232C or parallel. The October 1987 issue of Modern Electronics ***contains an article on the Tandy speech synthesizer set, with information that may prove useful to you. The letters columns in the January and April 1988 issues have corrections for errors in the original article.***

-- INK Research Staff

In **Visible Ink,** the Circuit Cellar **Research Staff** answers microcomputing questions from the readership. The representative questions are published each month as space permits. Send your inquiries to: INK **Research Staff, c/o Circuit Cellar INK, Box 772, Vernon, CT 06066.** All letters and photos become the property of CCINK and cannot be returned.

The Home Satellite Weather Center

Part 5

Focus on the MC68000 Peripheral Controller

by Mark Voorhees

Editor's Note: ***This is Part 5 of a 12-part series on The Home Weather series. Mark frequently refers to circuitry described in previous installments. For information on ordering back issues, see page 39.***

As I'm writing this portion of our series on weather information tools, I'm sitting in my room at the San Diego Princess Resort, overlooking an overcast Mission Bay, and wishing that I had paid more attention to the data and maps I'd looked at prior to the trip. They foretold the presence of the fronts producing the low temperatures (highs of 65-75 degrees, much better than the 108 at my home in Phoenix yesterday) and the cloudy skies (I would appreciate a little sunlight!) likely for the duration of my stay -- looks like the data was right on target. Oh well, it's a nice area to visit. Hawaii is a little too expensive to visit this year (although it's climatologically superior!), and, after all, this is a working vacation.

There is a point to all this (in case you were beginning to wonder). The conditions here could put quite a damper on a vacation trip with the wife and kids unless you had the information warning you to expect it. That's one of the ways our system will help: providing you with the tools to gather data about the weather phenomena occurring around you.

Enough of our little vacation, and back to focusing on the construction of the Home Weather Center 68000-based Peripheral Processor. In this installment, we'll talk a little more about the main processor circuitry, as well as describe the mass-memory circuit and a front-panel circuit, and I'll give recommendations on a power supply and case for the finished system.

When I discussed the main processor in the last issue, I covered the operation of the circuit in only general terms. The precise usage of the hardware will, of course, be guided by the firmware (a subject of the next installment). The circuit itself is actually a very generic 68000 application, and (experimenters take note) could probably be used for many other purposes.

Also, in the last issue I neglected to mention the purpose for connector CN4. It is the power connection for the RGB/NTSC encoder board described in the first two issues of Circuit Cellar INK. Sufficient room exists in the recommended enclosure to mount this unit, if desired.

Let's take a closer look at some of the circuit elements as they will be used in our project:

a PAL for Memory Selection

Editor's Note: ***The schematic containing U102, U105, U106, and U112-U114 was presented in the July/August*** *issue of* ***INK. Logic for the PAL is available on the Circuit Cellar BBS or the INK*** *Software* ***Disk for this issue. For downloading and ordering information, see page 44.***

This Programmable Array Logic (PAL) device has high-order address lines (A16-A19), data strobes (UDS, LDS), and the address strobe (AS) from the 68000 available to it, as well as system clock and an on-board memory select (OBMSEL) logic from U114.

The device serves the processor by handling the selection of chips within the 32K x 16 EPROM area, or the 32K x 16 on-board static RAM, which we'll use for scratchpad memory as well as stack and configuration storage areas. U112 also generates a delayed data acknowledge (DTACK) signal to allow for EPROM and SRAM setup times. The address area for EPROM is OOOOOOH to 01 FFFFH, and for on-board RAM is 020000H to 03FFFFH. We won't be using all of these areas at the moment, but the remainders are available for future use if needed.

Battery-Backed SR iii

SRAMs themselves are really rather self-explanatory, but I do want to mention something that is not on the schematic: the battery backup method used to protect these chips from memory loss.

My system design allows for battery backup for the entire project. As with any battery-type device, however, there are limitations. I wanted to provide the stack

and configuration storage with separate protection. I chose to install a special type of socket under the memories. The unit is called a "SmartSocket," and is made by Dallas Semiconductor (part no. DS1213C). The unit is basically a standard-configuration 28-pin DIP socket with a built-in CMOS controller and lithium battery source.

The SmartSocket is totally transparent until the embedded controller sees a Vcc out-of-tolerance condition. It then write-protects the SRAM and powers it with its own energy source until normal Vcc is restored.

While this addition is not required, it is recommended, especially to those who may not feel that a full battery-backup scheme is necessary for their application.

This PAL device uses A6-A8 as well as the data strobes (UDS, LDS) from the 68000, and logic from U114 indicating that the memory-mapped address area is selected (IOSEL). U113 allows for 7 active devices, each with up to 32 assigned addresses (used for register access, etc.)

The memory-mapped device area is designated to start at 300000H.

This PAL device provides for the mapped select lines, and uses A20-A23, the address strobe (AS), and the function code information (FC0-FC2) from the 68000 to create its outputs. It also provides the decoded interrupt acknowledge signal (IACK) from the 68000 when the three function code lines are high. R1SEL and R2SEL are provided as bank selects for the mass DRAM memory.

Along with providing the UART and programmable baud-rate generation for communication with the control terminal (your PC), this device provides interrupt prioritizing logic and two utility timers (which we'll make use of later on in the series).

Before we leave the main board, I want to mention something of interest to those of you who like to take a project and improve or customize it to meet your special needs: occasionally I'll provide an idea, suggestion, or circuit of a possible change in the design that could lead to enhancements to our overall system. These ideas are presented "on paper" only, and are not required for operation of the system as designed. The front panel circuitry is an example of this concept.

Figure 1 shows the schematic of the standard front-panel circuit. This display will give us basic prompts on the Peripheral Processor's "health," including status items such as "No Config," "Memory Full" (for the DRAM memory), and so on. The circuit also supports four switches for software sensing (clearing WEFAX images without downloading them is a typical application), and a "warm boot" reset switch (resetting the 68000 to "boot" state without destroying the scratchpad-held configurations).

I've also made allowance for a "processor running" indication, LED1 is driven by the processor "HALT" line, and is normally illuminated. It will extinguish if the processor halts for any reason. This is a dual-purpose indicator, in that it will only stay out if, for some reason, the "watchdog" reset circuit fails to restart the processor.

I've designated CN5 to serve as interconnect to the front panel. Figure 2 shows added connections to implement the basic design.

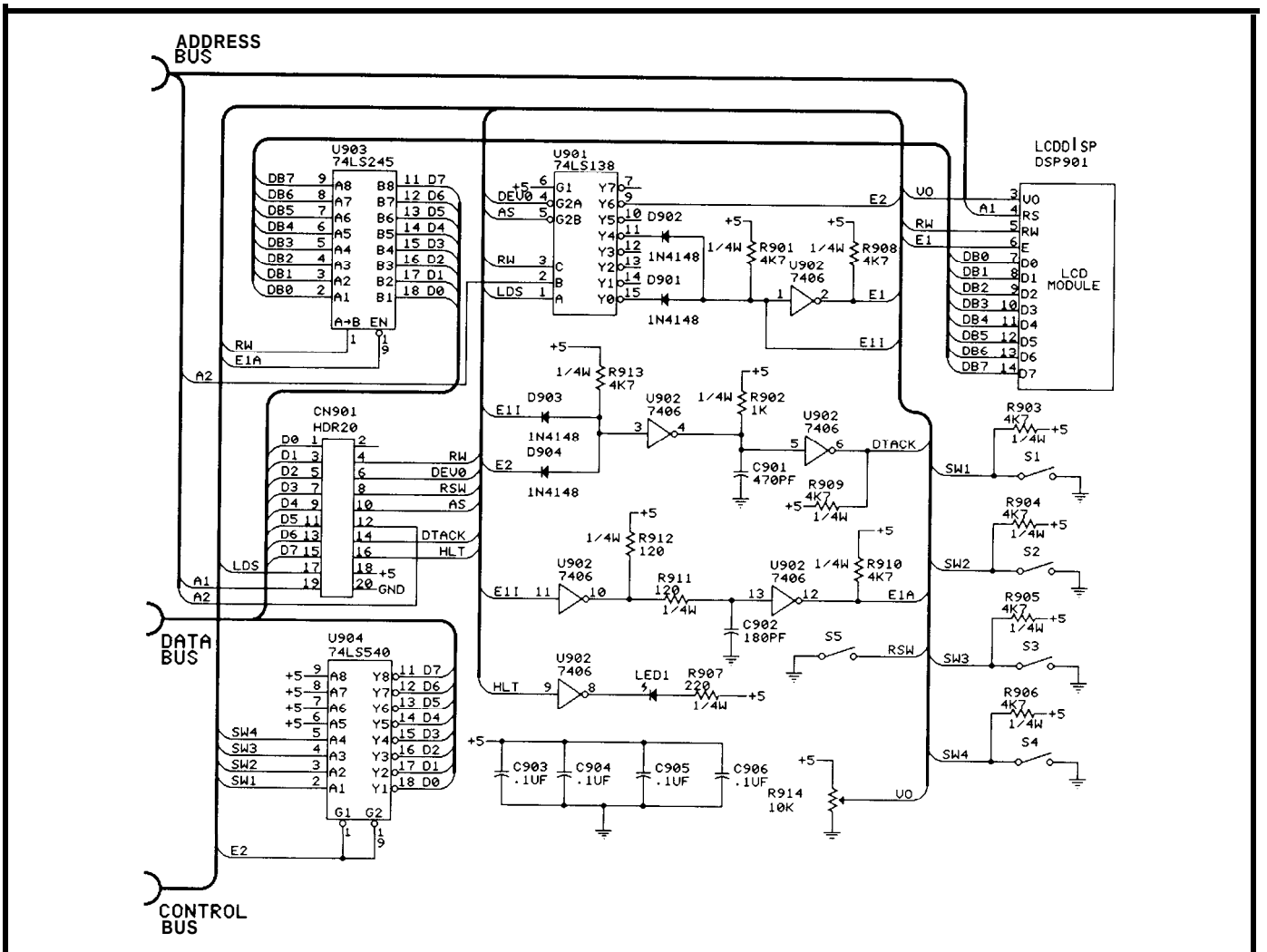


Figure 1 - ***The standard front-panel circuit, showing LCD display, switches for software sensing, and a warm boot reset switch.***

These are not critical parts omitted from the original. Rather, hese pins might be needed for ther signals if you decide to ustomize your front panel. The nain buses (CN1A and possibly CN1B) could be brought to the ront-panel circuit if you needed ophisticated access to the 68000.

As to the standard design, the ircuit of Figure 1 handles the nterfacing of the software-ensed switches via U904 (you ould add up to four more witches to this design, connecting them to U904-6, U904-7, U904-8, and U904-9), and also

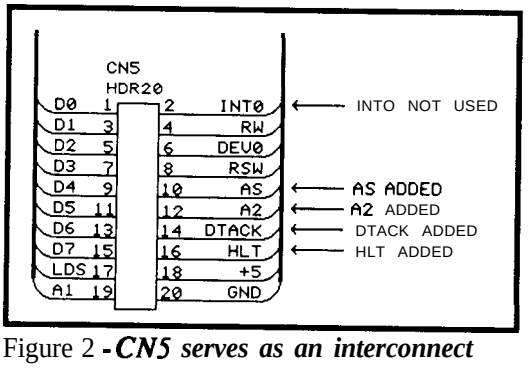


Figure 2 - ***CN5 serves as an interconnect to the front panel to implement the basic design.***

provides for the connections to the LCD module, a Philips (Amperex) LTN2 11 R- 10 or equivalent unit This module provides for display of two lines of 16 characters, with necessary cursor and display controls built in.

Final address decoding is done by U901, which provide! signals to enable the LCD module as well as U904 and U903.

The LCD module used here is described in Philips Technical Publication 238. It is quite self-sufficient, but lacks the logic to create high-Z bus conditions or tc generate a data acknowledge (DTACK) to the 68000 board. Much of the remaining circuitry

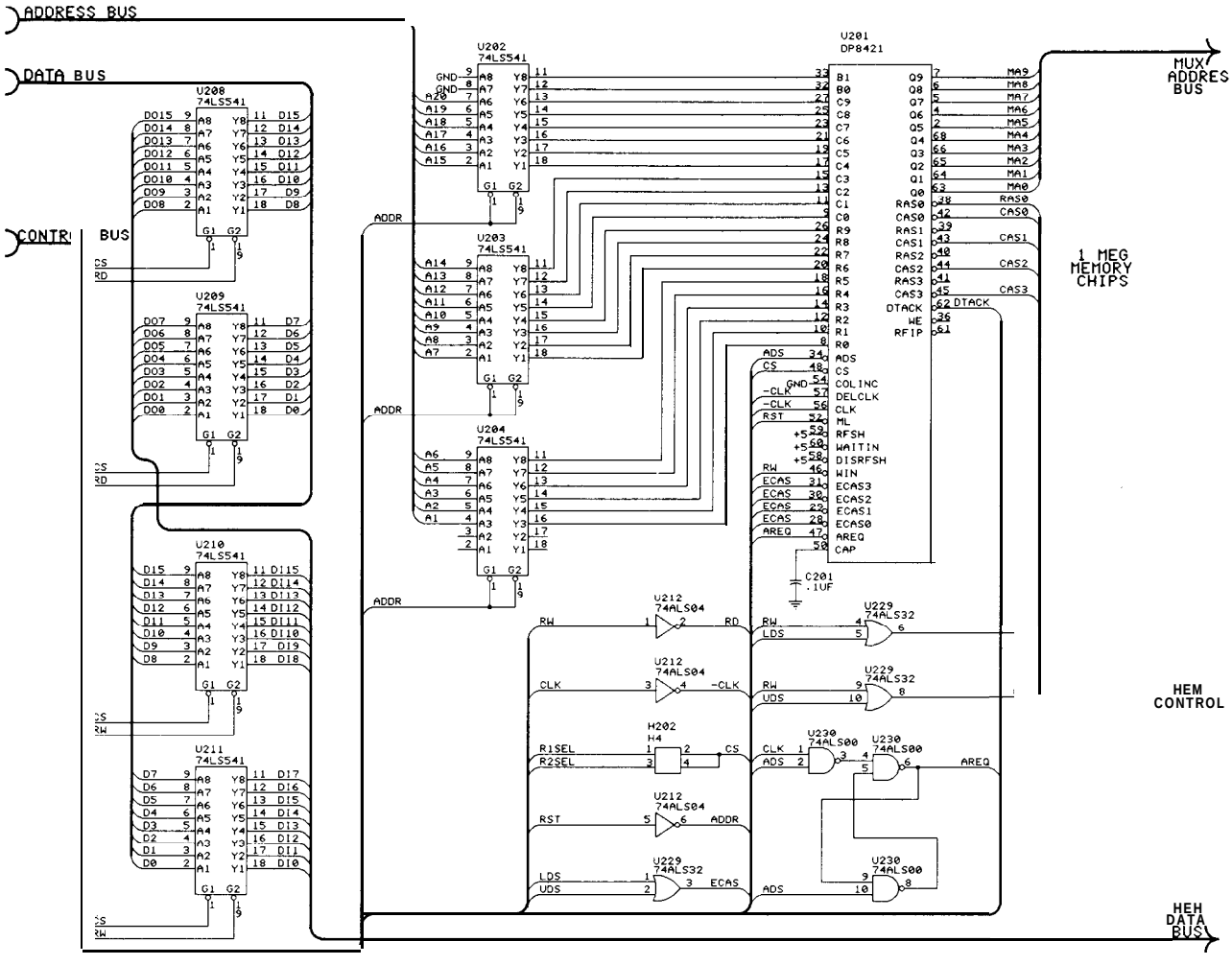


Figure 3a - ***Controller circuitry for DRAM board. The National Semiconductor DP8421 is simple to design around and requires minimal support from the hard-working MC68000.***

(U903, U902) provide, t[illegible] essary logic to accomplish these

front-

TC511000P 1 Meg x 1 page-mode [illegible] (100-ns 120-ns

(NEC's uPD411000 would be an acceptable substitute).

The DRAM controller is the Nal support requirements from the 68000 (which will have enough to do!). The controller receives configuration information at boot time and then operates virtually unat-

The design is based on the premise that the system will be using the mass memory for data storage (a "RAM disk" of sorts), and therefore

the DRAM can be expanded based on space and power availability. In the standard design, two boards is a practical limit.

Before continuing, a short explanation is necessary. As we all end in sight. I planned this circuit

current price vs performance comparisons lean to using l-Meg designs. However, hedging a bit on my "cloudy crystal ball" predictions, I'm providing a c on 256K DRAMs

&gures 3a layout of the 1M

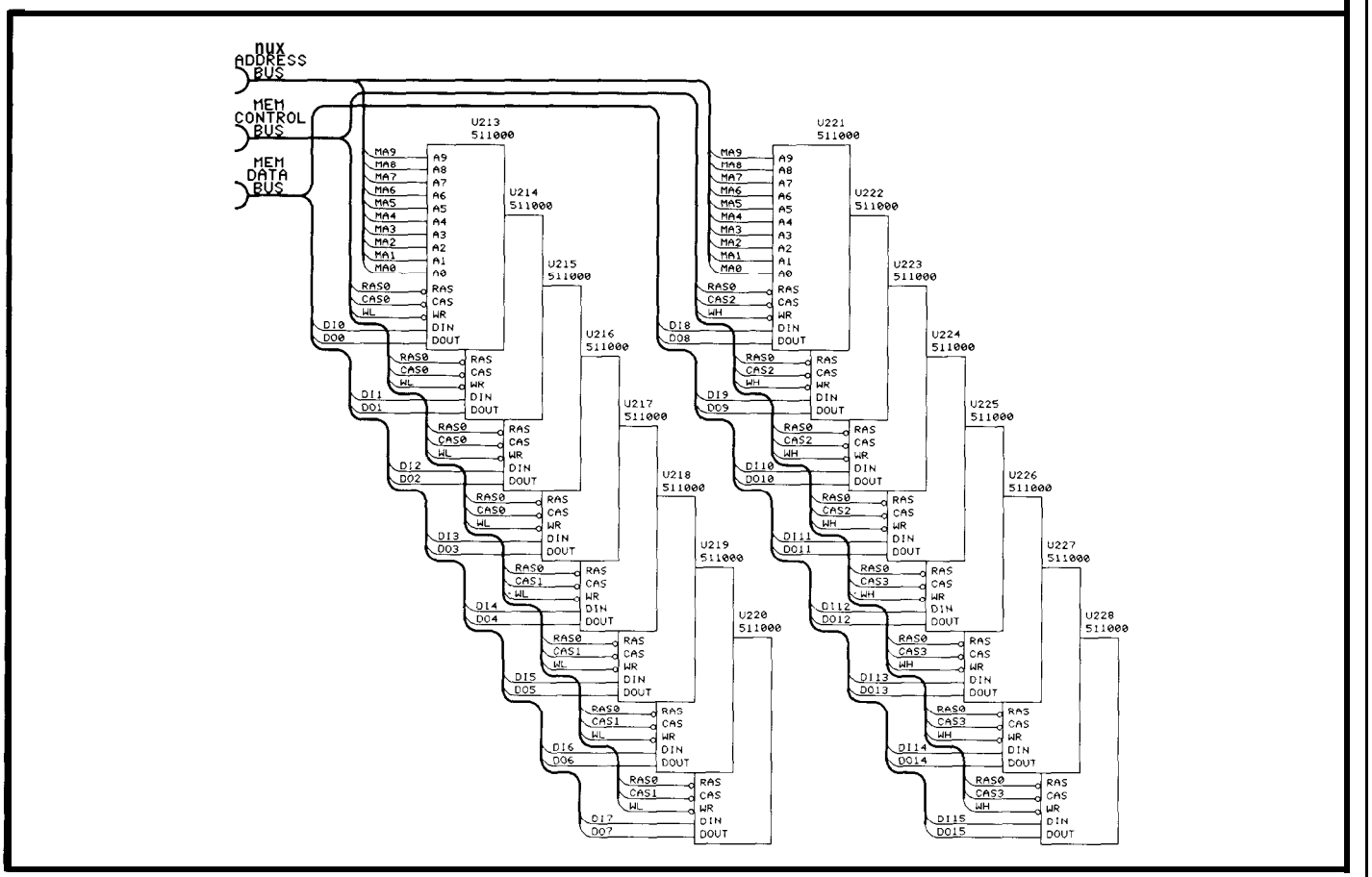


Figure 3b - ***Pin assignments for the I-Mbyte DRAMs used on the memory board.***

You *can get the schematic for the 256K DRAM version of the board by sending a SASE to: Circuit Cellar INK, P.O. Box 772, Vernon, CT 06066.*

If you have a massive supply of 256K DRAMs or find them available at a super price, then this design is for you, with the following warnings: first, you will need to increase the size of your cabinet to allow for the additional board space and ventilation due to the addition of 48 ICs, and second, you will have to increase power supply to support the extra chips. In this case, I think that I would provide a separate supply for the memory board(s), as the power supply I will mention later will likely not handle the extra load (based on entire system requirements).

I must confess that I loathe the idea of coming up with power supplies for projects. "Brute force" designs are the least costly for experimenters, and parts are readily available, but they are inherently heat-producing and tend to take up too much space in an otherwise neatly designed enclosure. They are, in many ways, inefficient.

Switching supplies, on the other hand, are efficient, but difficult to construct for an experimenter. Purchasing a ready-built supply is the alternative, but often is too costly.

I toiled over many power supply designs for this project, finding them inadequate in one way or another. We need +5 volts at a minimum of 5 amps for the system, as well as about 350 mA at +12 volts and 200 mA at - 12 volts. And all this in a minimum of space.

One of my suppliers surprised me with a product from General Instrument. The Model 35-502 switching supply slightly exceeds our power requirements, is reasonably priced, and the size and style of its enclosure lends itself to mounting on the rear of the system cabinet, thus keeping supply noise and heat as far away from the circuitry as possible.

This power supply is a widely available product, I'm told, so finding it should not be a problem. My supplier has a limited number of these units in stock as of this writing, and 1'11 gladly provide information and pricing. See the ordering sidebar for details.

The supply requires little in the way of external support, as is shown in Figure 5. The power switch on the rear will remove AC

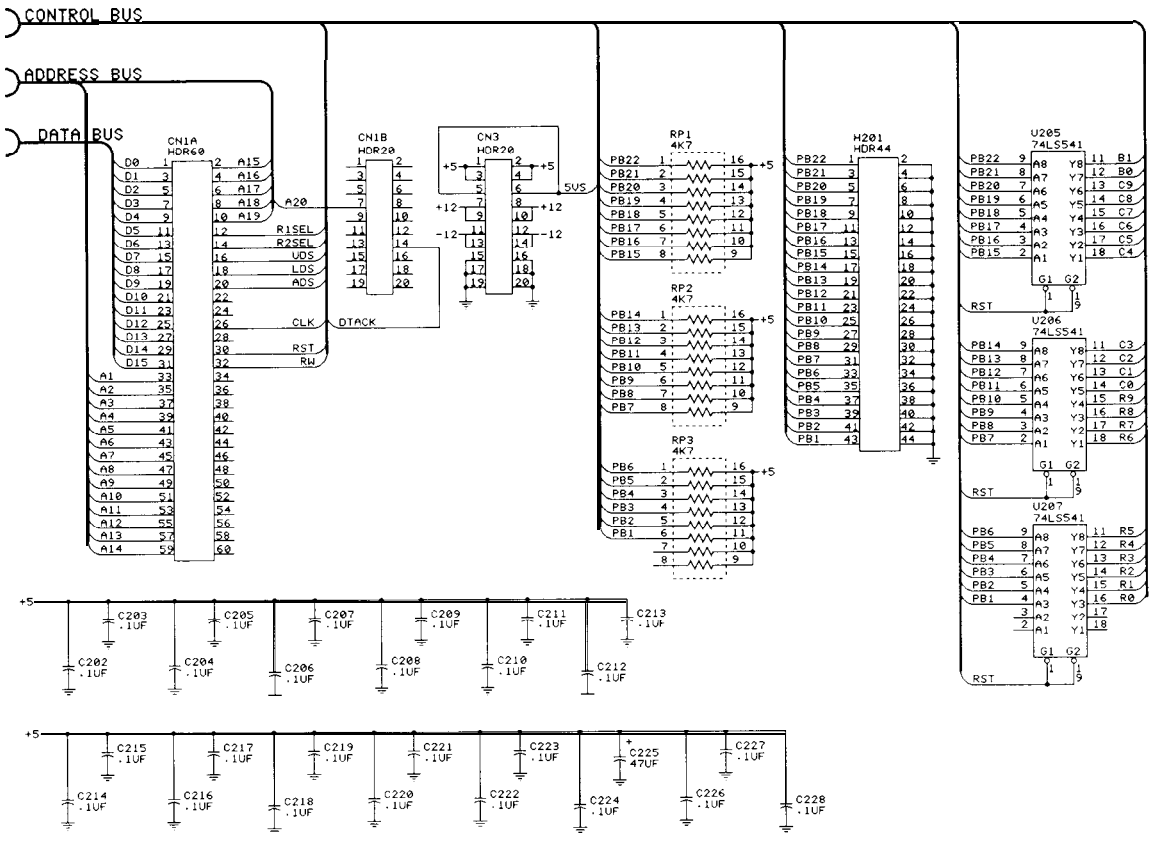


Figure 4 - ***The 68000 peripheral controller provides ample interaction with the outside world through the slots shown above.***

power from the supply, but not the battery system (to be described in a future installment, which will also cover the interconnections to the power bus; for now, the temporary connections of Figure 6 can be used for testing).

I have searched for an adequate enclosure for our project, which needs about 10" by 5" by 11" of interior space, and I've located a PacTec enclosure that meets the requirements. The model CL-525 case has front and back panels that are 5.3 12" x 12", allowing for space to install our power supply and connectors on the back, and our front-panel display with switches on the front. You can get the name of your nearest distributor by contacting PacTec directly:

PacTec
Enterprise and Executive Aves.
Philadelphia, PA 19 153

I have a limited number of these enclosures available at special pricing. See the ordering sidebar for information. I will also provide cutting and drilling templates for the standard configuration by mail (since space is limited here, and I'll be providing a full-scale drawing).

This is, by no means, the only answer to the need for a cabinet. Any case meeting the requirements above will house the standard system. If you plan to customize, or build the memory configuration which uses the 256K DRAMs, be sure to take the extra space needs into consideration. If you plan on using your system in a den or similar room (and you're familiar with fine woodworking techniques), you might want to construct your cabinet of furniture-grade hardwoods to make it a showpiece!

In our next episode, I'll be describing the firmware which runs the system, as well as describing the methods we'll use to port the data into and out of the Peripheral Processor. (You can see Figure 4 for a headstart on I/O.) We'll also venture a bit ahead to describe the WEFAX antenna and RF modules which you'll need to provide the WEFAX signal to our Peripheral Processor (these pieces of RF equipment are fairly critical, so I don't recommend building them

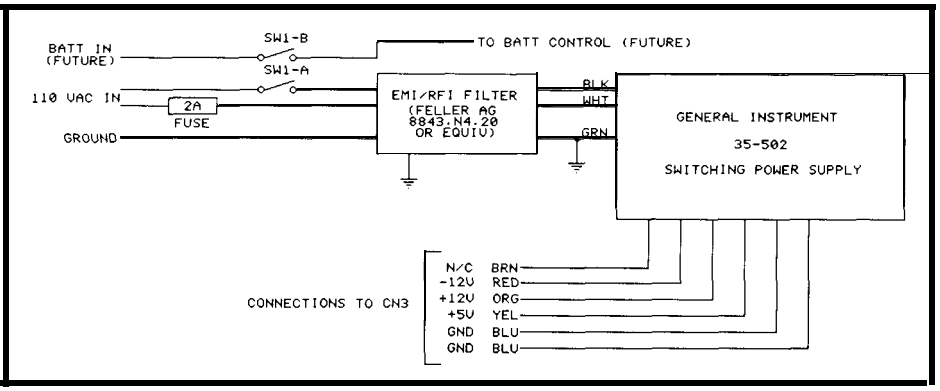


Figure 5 - *The circuit for the power supply will allow you to cut AC power while retaining the battery back-up.*

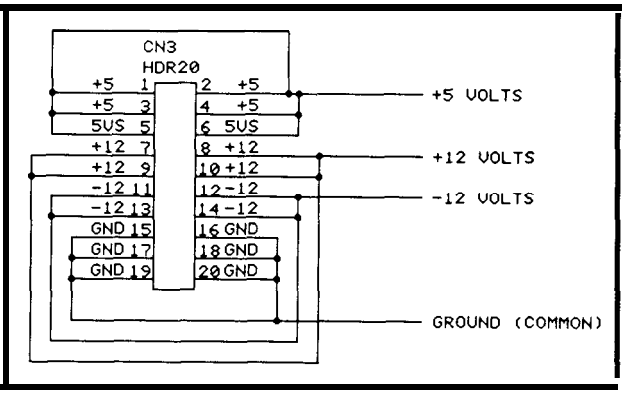


Figure 6 - *Temporary connections for testing the power supply_*

yourself; I've found units that are reasonably priced and perform well).

Before I wrap up for this issue, I want to cover some questions I've received on using the RGB/NTSC Encoder board which was presented in issues 1 and 2 of Circuit Cellar INK.

Q. Can the Encoder be used on PAL (European) TV systems?

A. *The MCI377 **IC** contains the circuitry to allow for this change. It will be necessary to make the following changes in constructing the circuit:*

I. The EPROM will have to be programmed with the proper pattern for PAL sync.

2. Pin 20 of the MC1377 will have to be pulled high (or left open).

*3. The TTL oscillator module will need to be changed to 17.72 MHz to handle the different **sub**-carrier frequency generation and provide proper timing for the EPROM.*

*Most of the setup procedures should yield results similar to the NTSC setup, but if you build a unit for PAL use, I recommend that a **PAL**-qualified serviceman or TV engineer make the adjustments for you.*

Q. Can the encoder work with a monochrome signal?

A. *It's possible, but not recommended, since the signal specs for mono TTL video are much looser than for color. You could feed the mono signal simultaneously to the R, **G**, and **B** inputs, feed the intensity signal to the I input, and feed H and V sync to their appropriate inputs, and see if the output of your graphics card will agree with the timing capabilities of your encoder.*

*The main problem stems from the fact that the NTSC color signal specifies a horizontal rate of 15.734 **kHz** and a vertical rate of 59.94 Hz. These numbers didn't come out of thin air; they relate directly to the **3.579545-MHz** color subcarrier frequency and, therefore, have a relationship to our encoder's **14.31818-MHz** clock and EPROM.*

*The mono signal, however, sets H rate at 15.75 **kHz** and V rate at 60 Hz, which could cause noise or waves in the picture due to the timing relationships.*

Q. Can analog RGB signals be fed to the encoder?

A. Yes, *within the limitations of standard horizontal and vertical sync frequencies (the same limitations as with TTL, i.e., no **double**-rate horizontal frequencies as in EGA). This is the reason that CN2 is **jumpered** at the MC1377 (I likely will make use of that header in the future also, as part of the Home Weather Center). The headers on pins 3, 4, and 5 will accept **capacitivelycoupled** (nominal 15 **uF**) R, G, and B inputs of l-volt peak-to-peak level noncomposite video (no sync). The horizontal and vertical **syncs** are usually 4 volts peak-to-peak and can be handled through the normal sync path. Remember to leave the jumpers on pins 1 and 2 of the MC1377 in place. Setup should be virtually the same as with TTL inputs.*

(As space permits, I'll be handling questions relating to our project in these articles. If you have ideas or questions to share, please forward them to the address in the ordering sidebar on page 45.)

I think it's time to end this month's installment on an optimistic note: by next summer, we should all be better equipped to handle the planning of our summer vacations, weather-wise. As for me, I vow to take advantage of a better climate next year (maybe if I start saving for Hawaii now. ..).

FROM THE BENCH

Conducted by Jeff Bachiochi

RS-232 Economic Tradeoffs:

Board Space vs. Parts Count vs. Parts Co$t

Most "system" designs include at least one serial I/O channel for communications with the outside world (modem, printer, terminal, etc.). A minimum serial I/O channel will consist of at least three connections: transmit, receive, and a common ground. Additional lines can be used for optional control functions such as request to send, clear to send, data set ready, data terminal ready, and carrier detect.

RS-232-C, the standard for serial interface between Data Terminal Equipment (DTE) and Data Communication Equipment (DCE), requires a positive level of +5 to +15 volts to indicate a "0" (SPACE or ON) and a negative level of -5 to - 15 volts to indicate a "1" (MARK or OFF). These levels are not compatible with the "system" TTL levels of "ground" (0 volts) and "+5" (volts). The TTL levels must be changed to and from RS-232-C levels. This change is accomplished through the use of two kinds of level converters: a line driver (the transmitter), which converts a O-volt TTL level to a positive 5-15-volt level and a +5-volt TTL level to a negative 5-15-volt level (depending on the supply voltage); and a line receiver, which converts a positive 5-15-volt level back to a O-volt TTL logic level and a negative 5-15-volt level back to a +5-volt TTL logic level. (See Figure 1).

For years the "standard" parts used for this level conversion were the 1488 (TTL-to-RS-232 transmitter) and the 1489 (RS-232-to-TTL receiver). These two 14-pin IC packages, shown in Figure 2, each contain four transmitters or receivers.

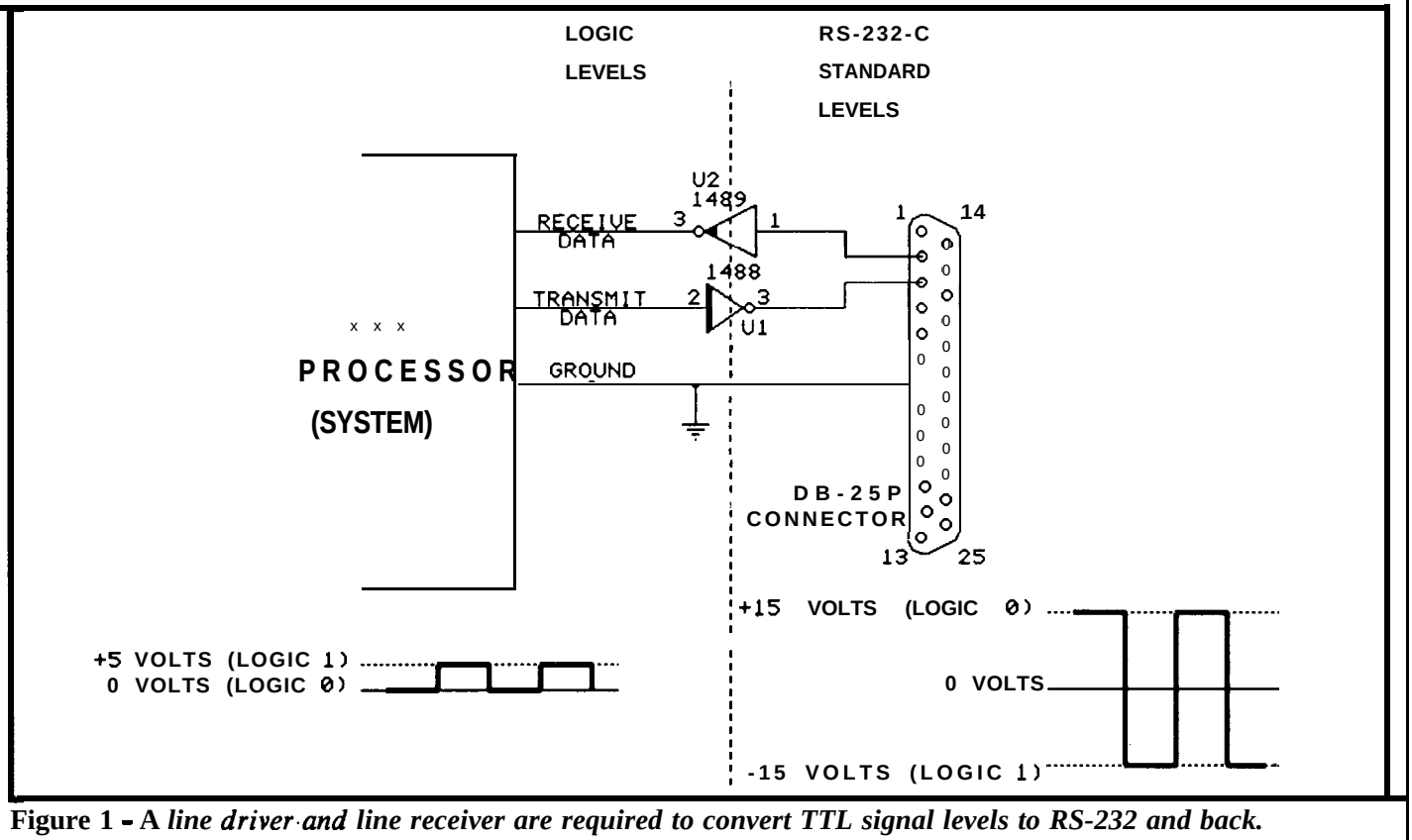


Figure 1 - A *line driver and line receiver are required to convert TTL signal levels to RS-232 and back.*

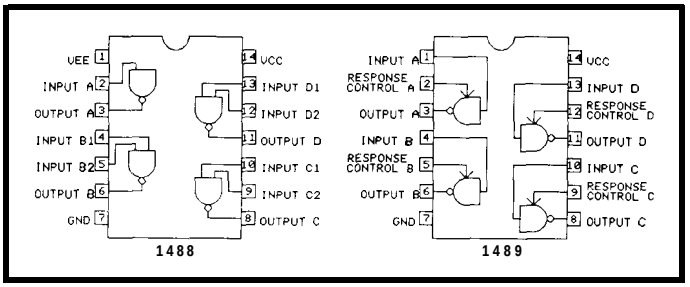


Figure 2 **-** ***The 1488 and 1489 are standard components for TTL to RS-232 conversion.***

With the onset of CMOS devices, direct replacements for the 1488 and 1489 have arrived. The 14C88 and 14C89 take advantage of CMOS technology to provide very low power consumption.

DEVICE	BOARD AREA	POWER
1488	**1/2** square inch	**±10-15** volts **@** 20 **mA** each
14C88	**1/2** square inch	**±10-15** volts **@1/2mA** each
14C89	**1/2** square inch	**+5 volts @ 1 mA**
1489	**1/2** square inch	**+5 volts @** 20 **mA**

Advantage: Reduced current requirement in a pin-compatible package.

Motorola took the next step. The MC 145406 CMOS transmitter/receiver (see Figure 3) combines three transmitters and three receivers within the same package. At sixteen pins, this IC is slightly larger than either the '88 or '89, but now only one device is necessary.

DEVICE	BOARD AREA	POWER
MC145406	**1/2** square inch	**±10-12** volts **@1/2mA** each
		+5 volts **@** 1 **mA**

Advantage: Reduced current requirement in half the board area.

Many systems can operate totally on **+5** volts. This creates a problem since RS-232 communications need both a positive and negative supply (preferably ± IO-15 volts). Solid-state voltage doublers and inverters can be used to create the +/- voltage from **+5** volts. An increase in required board space and parts count can be exchanged for the elimination of the **±10-15-volt** power supplies.

Advantage: Only **+5-volt** power is needed.
Disadvantage: Requires 4-6 times as much board space.

Monolithic voltage converters for voltage doubling and inverting have been around a few years now. They are a reasonable alternative to multiple-output power supplies whenever the current requirements are low, as with the RS-232 transmitters. Maxim, one manufacturer of converters, has taken this evolution one step further. The MAX232, shown in Figure 3, incorporates two transmitters, two receivers, and voltage converters together in one 16-pin package. Four external capacitors (used as charge pumps), in addition to the IC, combine to make a small CMOS transmitter/receiver combination that runs totally on **+5** volts.

Advantage: Only **+5-volt** supply needed.

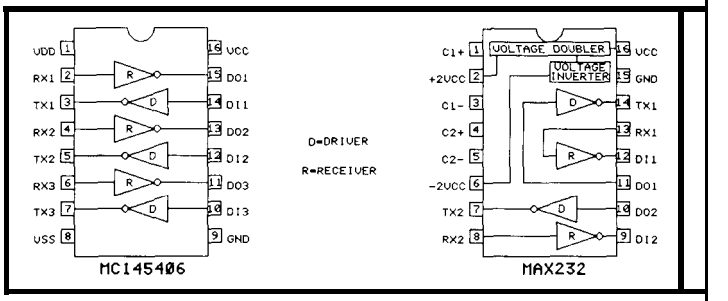


Figure 3 - MC145406 *& MAX232 have advantages over the 1488/1489 pair.*

The Maxim part is second-sourced by Dallas Semiconductor (DS232) and others. Maxim has an expanded line based on the MAX232 consisting of various combinations of transmitters and receivers.

For further information contact:

Maxim Integrated Products, Inc.
510 N. Pastoria Ave.
Sunnyvale, CA 94086
(408) 737-7600

Dallas Semiconductor Corporation
4350 **Beltwood** Pkwy. S.
Dallas, TX 75244
(214) 450-0400

Motorola Semiconductor Products, Inc.
3501 Ed Bluestein Blvd.
Austin, TX 78721
(512) 452-7673

Innovations like these help to make today's technology more cost effective, reliable, and easier to use. Please share your favorite ideas, chips, and circuits with others. We will pay $25 for any **From the Bench** accepted for publication. All submissions should be typed, double-spaced, and include neatly drawn schematics or Schema configuration, library, and page files. Include a stamped, self-addressed envelope large enough to hold everything if you wish the materials that have not been accepted to be returned. Submit to:
From the Bench, c/o Circuit Cellar Ink, Box 772, Vernon, CT 06066

10-MHz/8-b
Board for t
An Affordable D
on a Plu

We're all designers of one type or another. For those of the electronics ilk, tracking down one of those inevitable bugs frequently requires capturing an elusive glitch or some ephemeral flash across the analog scope. In both of these cases, what you really need is a way to capture a transient event: You need a digital 'scope. This article presents the single-board PC-bus digitizer needed to turn your PC into a 10-MHz digital oscilloscope.

Bytes in Bits

This digitizer converts an incoming analog or digital voltage signal into a sequence of 8-bit words. These words are stored in a fast 2K static RAM buffer until a trigger pulse with a programmable delay stops the storage. The data is then read by software running on the IBM PC, stored to disk, and plotted on a CGA display. Because the incoming signal only needs to occur once to be digitized, single-shot transient events can be captured and displayed. In this way, an IBM PC can quite inexpensively simulate the capture functions of a digital oscilloscope. Other uses for the digitizer include signal averaging, data storage for documenting circuits, and general debugging of new circuit designs. Since the board is designed to use a standard 10x oscilloscope probe, an input range of +/- 5 volts is accommodated.

The Best of Analog and Digital

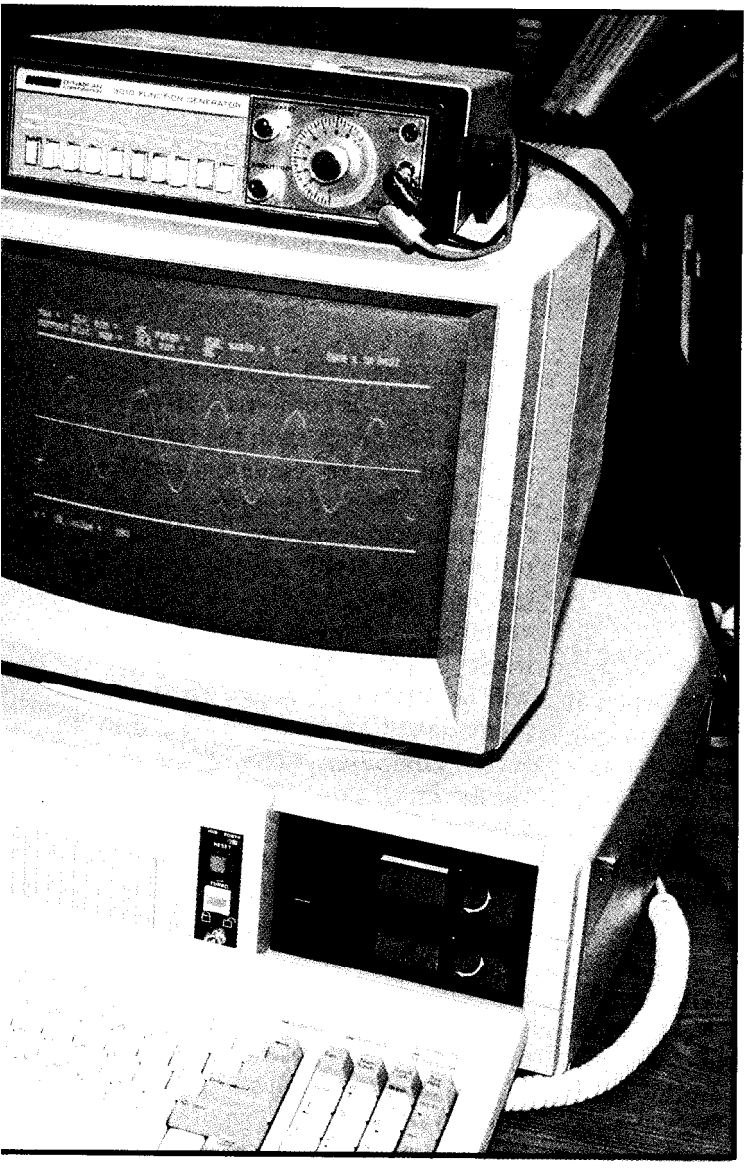
The best features of an analog oscilloscope have to be the use of standard 1 -Megohm probes, reliable triggering, and "instant" display. The best features of a digitizer are pretrigger transient capture, programmable trigger delay, and external clock and trigger control inputs. All of these features have been incorporated into this design, while maintaining low cost and good availability of parts. I would have appreciated having data storage of 32K bytes when searching for a glitch, but plotting a 32K-byte file is rather slow, even for 12-MHz ATs. And after saving a thousand 32K-byte files (which only takes a couple days of work), you're at the end of a 32-meg drive.

Two programs are available for use with this board: DSCOPE.EXE and PLOT.EXE. DSCOPE is a graphical window interface for setting up the digitizing parameters of the board (such as digitizing rate, trigger source, trigger delay), capturing and plotting signals, and storing selected files to disk. PLOT is simply a 128x640 graphical plotting program for displaying data files stored on disk onto the CGA 640x200 graphics screen. A sample screen from DSCOPE.EXE is shown in Photo 1.

it Digitizing
ne IBM PC
gital Oscilloscope
:-in Board

by Russell Lindgren

Both DSCOPE.EXE and PLOT.EXE ***are available on the Circuit Cellar BBS and the INK Software Disk for this issue. See page 44 for BBS downloading instructions and information on ordering the disk.***



For ***most frequencies within its*** range, ***the digitizing board matches the performance of a more expensive, stand-alone oscilloscope.***

Although the digitizer is primarily for analog design, even an exclusively digital design can be aided by its use. After all, doesn't everyone still use an analog scope on a microprocessor circuit? It's also a tool that really shines when debugging a mixed analog/digital/microprocessor design (such as an image digitizer board). When I'm debugging a digital circuit with this digitizer board, I use the DSCOPE program with an external TTL trigger input and a fixed trigger delay value to initiate capture on a significant "event," and to step through capturing and displaying the digital signals in question, storing the ones that seem incorrect. I then exit the DSCOPE program and use the PLOT program to show a few of the captured data files on the same screen for comparison. I also use this digitizer to capture digital signals for detecting bad connections and bus contention (among other analog aberrations of digital logic levels). Each of these problems has a certain picture that indicates what's happening, and it's the sort of thing that a logic analyzer won't show.

I began the design of the digitizer board, shown in Photo 2, by reviewing a plethora of overpriced IO-MHz FIFOs and six- to eight-bit flash A/D converters. With low-cost 100-ns 256K-bit DRAMs common today, it's difficult to accept a 40-dollar price for a 512-byte FIFO. For A/D converters with 20-MHz specs, the situation is ten times worse -- and prototyping with a 400-dollar part is unnerving. Finally I settled upon the CA3318 from GE Solid State (formerly RCA). It's a relatively new part that has been around long enough to have the 100-piece price fall to $3 1 .00. It sports a 15-MHz throughput with 8-bit resolution and an analog bandwidth of 2.5 MHz, which should be good enough for 90% of all digital and analog circuits used with wire-wrapping. It also came with applications information which solved the design problems for the analog front end: GE recommends using the CA3085 voltage regulator, with 0.1% accuracy, as the reference, and the CA3450 video buffer amp with 10-megohm input impedance and high output drive for the high-capacitance input of a flash A/D converter.

Locating the right memory for 1 5-MHz digitizer operation posed a more difficult problem. Since I was seeking a minimum parts count, I

searched for FIFO memories. Since I was also looking for a low price, I had to abandon FIFOs after a brief review because all 20-MHz static FIFOs run over $30.00 for 512 bytes of storage. After a brief bout with the NEC 41101 dynamic FIFO, which offers a 30-MHz, 910-byte storage for under four dollars, but which lacks sufficient data retention for sampling frequencies under 1 MHz, I selected the old workhorse of microcontroller- based systems: the 20 16.

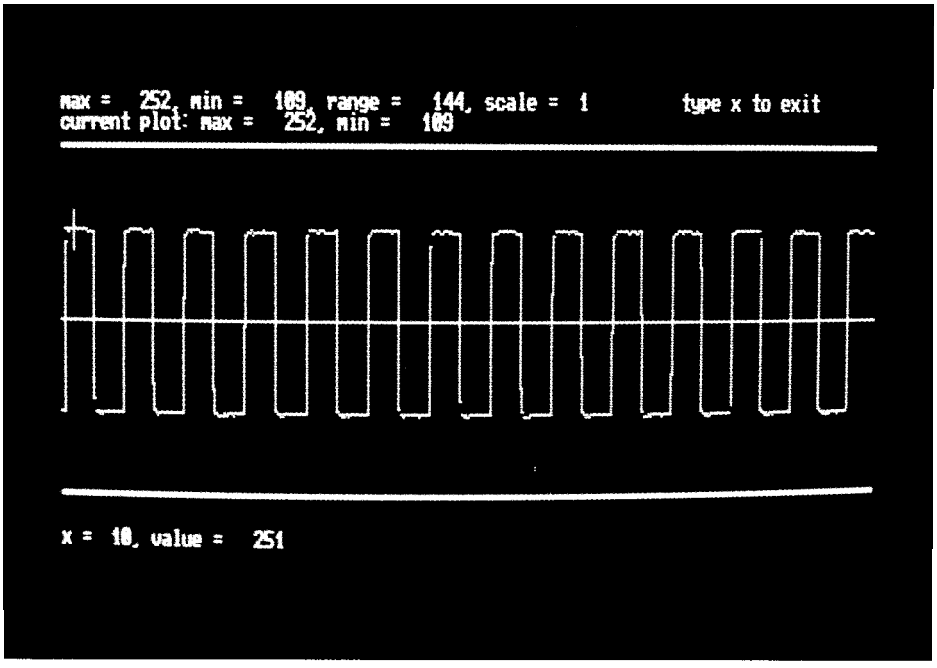


Photo 1 - ***As frequencies near the board's limits, performance begins to decrease.***

The 2016 (also known as the 4016) is a readily available 2K-byte static RAM with access times under 100 ns. If you really want to push the IO-MHz limit, special versions with access times under 50 ns are available from Cypress Semiconductor, Vitelic, and others. The only drawback of using a standard part such as the 2016 is the increase in parts count for address counters, with an offsetting decrease in memory cost. But the overall simplicity of the circuit, including the write pulse stretching circuit using the 7407, makes the circuit ideal as a "has-to-work" design for both novices and advanced engineers. I did briefly consider using a pair of 64K-byte 4-bit DRAMs available from both TI and OKI, but the increase in circuit complexity for address multiplexing and refresh was difficult to justify. If you need deeper memory circuits, keep an eye on future Circuit Cellar INK issues for a logic analyzer front end for the IBM PC. If you want more speed, use a 30-MHz hybrid crystal oscillator and a sub-70-ns RAM. Most 20X8 A-version PALs will work up to 30 MHz.

You know, I remember when it took a board full of 8- and 14-pin DIPs to get 6 bits of resolution at 1 megahertz. Today, linear signals have become much friendlier to digital designers. In this design, the linear components comprise only one quarter of the circuit board area -- four ICs. The input amp is the CA3450 video buffer from GE Solid State, which goes from a scope probe-compatible input to a high-drive 75-mA/10-MHz output in 16 pins and four dollars. The resistor-diode input protection circuit for the probe input will accept almost any over-voltage abuse when using a 10x scope probe, but be careful when using a direct DC input to the board -- don't exceed +/- 5 volts from ground if the signal source can supply over ten milliamperes. If the input is AC coupled (jumper selected on the header in Figure 3), you can be more relaxed about using the +/-500 mV input range, but for frequent use I would adjust the offset bias and change the gain of the amplifier circuit (with the famous Gain = (R-feedback)/(R-input) equation), or add a gain-selectable circuit to the front end.

The flash A/D converter block of the circuit is composed of three ICs: the LF356 op amp, the CA3085 precision voltage regulator, and the CA33 18 flash converter. The function of the voltage regulator is to provide a stable reference level (at 5.00 volts) to the flash converter, so that the measurements taken one day, on one computer, will be within 1% of the measurements taken another day on a different computer. The offset bias for the input amplifier is a negative 0.625 volts, which is produced from the reference level by a potentiometer and the inverting amplifier circuit of the LF356. The LF356 was selected because of its excellent low-drift characteristics and general availability. Any small error in the gain of the input amplifier caused by a mismatch of the input resistors can be compensated for by adjusting the reference level. For instance, if the input gain (which should be exactly 5) is 5.25, which is 5% too large, the reference level would be adjusted from 5.00 volts to 4.76 volts. To

trim-pots for P1 and P2. The offset bias for the input amplifier is also derived from the reference voltage, so if the gain has been changed, be sure to correct the offset. The function of the offset bias is to center an input signal ranging from -5 volts to +5 volts (using a 10x scope probe at the input) into the zero to +5 volts input of the flash converter. Refer to Figure 1, the schematic of the analog front end. One further note on the reference: don't try to adjust the reference level above 6 volts, because the external clamping diodes D5 & D6 will hold the maximum level to 6.2V. This is to protect the CA33 18 from damage by exceeding the maximum reference input voltage of 7.5 volts.

The clamping diodes D3 and D4 are used to protect the input of the CA3318. In order to simplify the circuit, they are Schottky diodes from HP, with an on voltage of 0.45 volts, which holds the input of the CA3318 within 0.45 volts of the digital 5-volt supply. The 10-ohm input resistor R11, is used to match the output impedance of the input amplifier to the highly capacitive input of the CA3318. For the CA3450 input amplifier to output 5 volts to the CA33 18 input, its positive supply has to be 7 volts. Since only the CA3450 requires this voltage, I used the three-terminal 78L12 regulator in Figure 2. Usually I only recommend using a three-terminal regulator referenced to ground, but in this low-current application, as long as the decoupling capacitors of the CA3450 (C5 & C13) are present, they provide a low-impedance path to ground from the -5V supply. The close placement of these two capacitors to the body of the CA3450 is crucial to stable operation of the input amp. The two gain-setting resistors R3 & R6, and the compensation capacitor Cl must also be placed as close to the CA3450 as possible.

Trigger circuits for oscilloscopes are always analog to provide glitch capture and fast, high-frequency response. That's fine, but the dark side of analog trigger circuits is their tendency for spurious oscillation. In other words, it is rather difficult to make a fast trigger stable or accurate, and a slow trigger won't cut it for debugging digital designs. Since I needed the speed for digital debugging, the only solution was a digital trigger.

The digital trigger in this design was implemented using the 74LS682 digital comparator. The output of the flash A/D is constantly updated every 100 nanoseconds. The seven most-significant bits (MSBs) of this output are compared to the 7-bit trigger value latched in U 11, a 74LS273 octal register in Figure 3 (the eighth bit is the polarity of the trigger). When the output from the A/D is greater than the trigger value, the trigger input (signal TI) goes low. This trigger signal goes to the timing logic PAL U8 in Figure 2, where it is logically combined with the external trigger (signal XTRIG) and the polarity (signal POLAR) to latch the trigger. Once the trigger has been latched the output signal GCK from U8 is enabled, clocking the trigger delay counter (U10 -- Figure 3) with each write to the data storage RAM. Since trigger is synchronous with the operation of the flash A/D, the maximum uncertainty of +/- one digitized sample is assured, and the design requirement for reliable triggering is met.

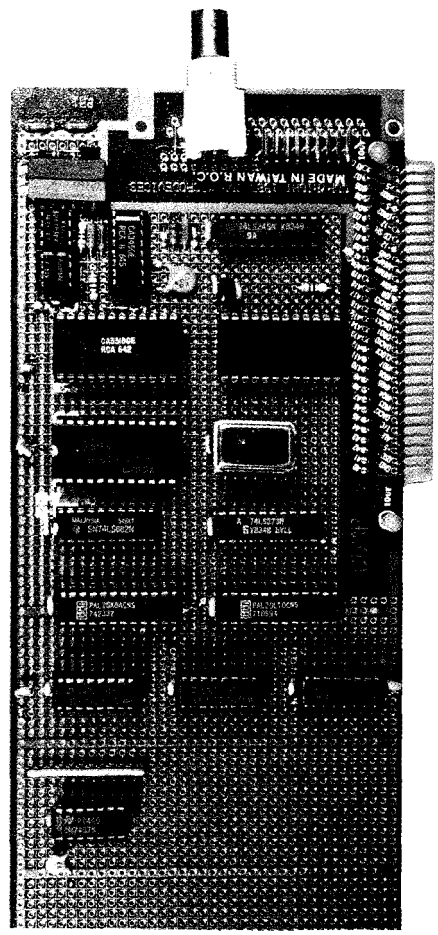


Photo 2 - *10-MHz/8-bit Digitizing Board for the IBM PC.*

The memory circuit functions as a First-In-First-Out (FIFO) buffer to interface two asynchronous timings: the fast data stream coming from the flash A/D converter, and the slow data readout to the 8088 data bus of the IBM PC. When the FIN signal is low, the data coming in is written into U5, the 100-ns RAM. When FIN is high, the data storage has ended, and the data can be read out. The software in the PC monitors the state of FIN through reading the status port, so that it knows when the data is ready. All the "glue logic" synchronizing the flash A/D, the RAM, and the data readout with the 10-MHz master clock is done inside a 20X8-A PAL IC.

Boolean equations for the PAL, used in this

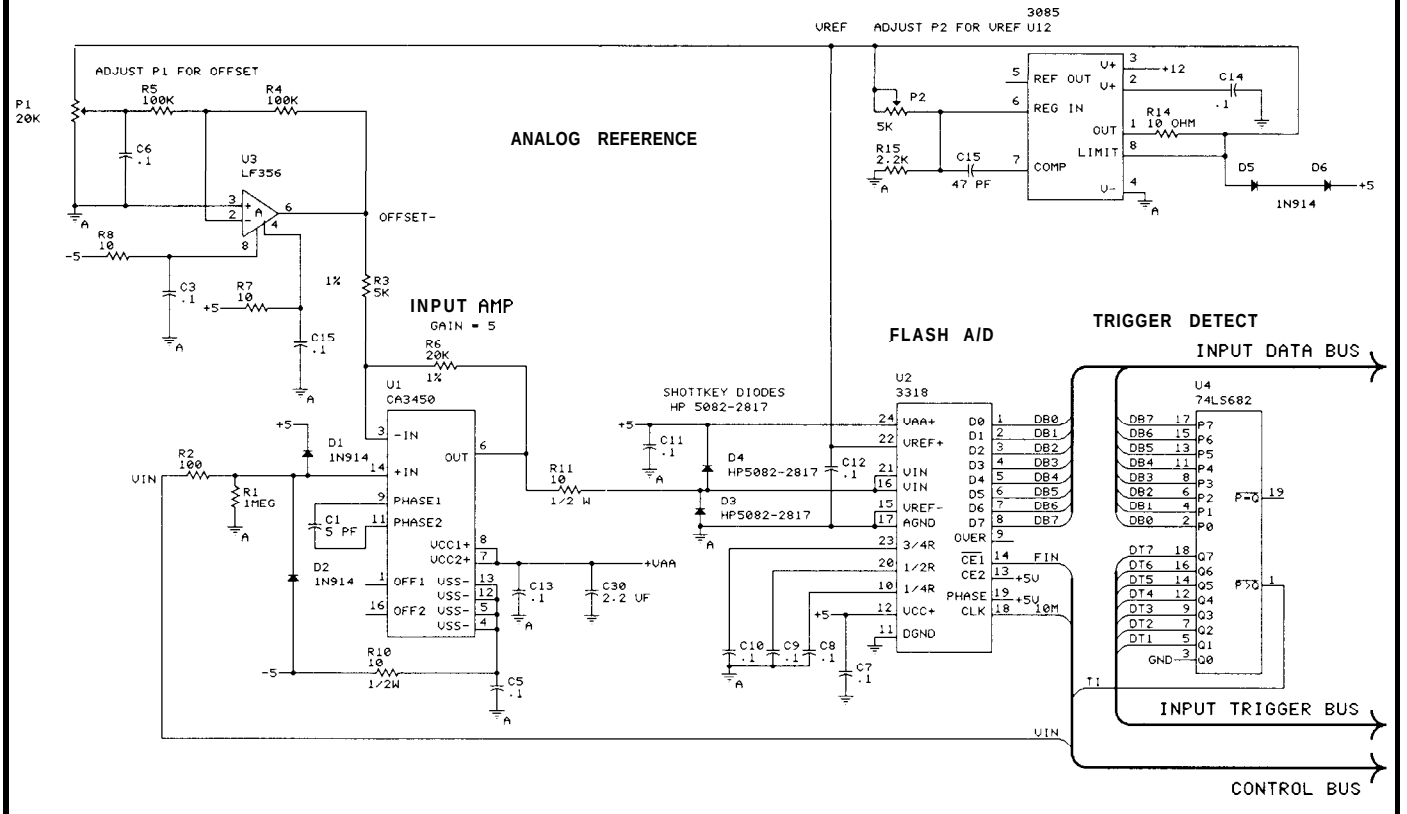


Figure 1 – *Digital scope analog front end and trigger.*

Circuit Cellar BBS or on the Software Disk for this issue. See page 44 for downloading and ordering information.

The 20X8 PAL was selected because its internal logic includes the exclusive-OR function required for the trigger and frequency doubling needed.

The memory write timing also has to accommodate the trigger. This is accomplished with the START signal. When the board is first reset by asserting the RST signal, the START signal goes low for 2048 writes, making sure that new data is present in the entire RAM before enabling a trigger pulse from either the TI signal or the XTRIG signal to latch the trigger. Once this trigger is latched inside the timing PAL U8, the trigger delay holds the FIN signal low (i.e., in the write mode) for a preprogrammed number of writes.

The flash A/D-to-memory timing works like this: the CA3318 has to always be operated at 10 MHz, due to its dynamic "charge balanced" mode of operation. New data is being clocked out onto the board's data bus (signals DBO-DB7) every 100 ns, on each rising edge of the 10M signal (IO-MHz square wave clock). The write-data clock, RCK, is synchronous with the 10-MHz clock, going low on the rising edge of 10M, and high on the falling edge of 1 OM, synchronizing the writing of data to the RAM with the data coming from the flash A/D. After each data write is completed, the address to the RAM is incremented.

This timing gives RCK a pulse width (while low) of one half the

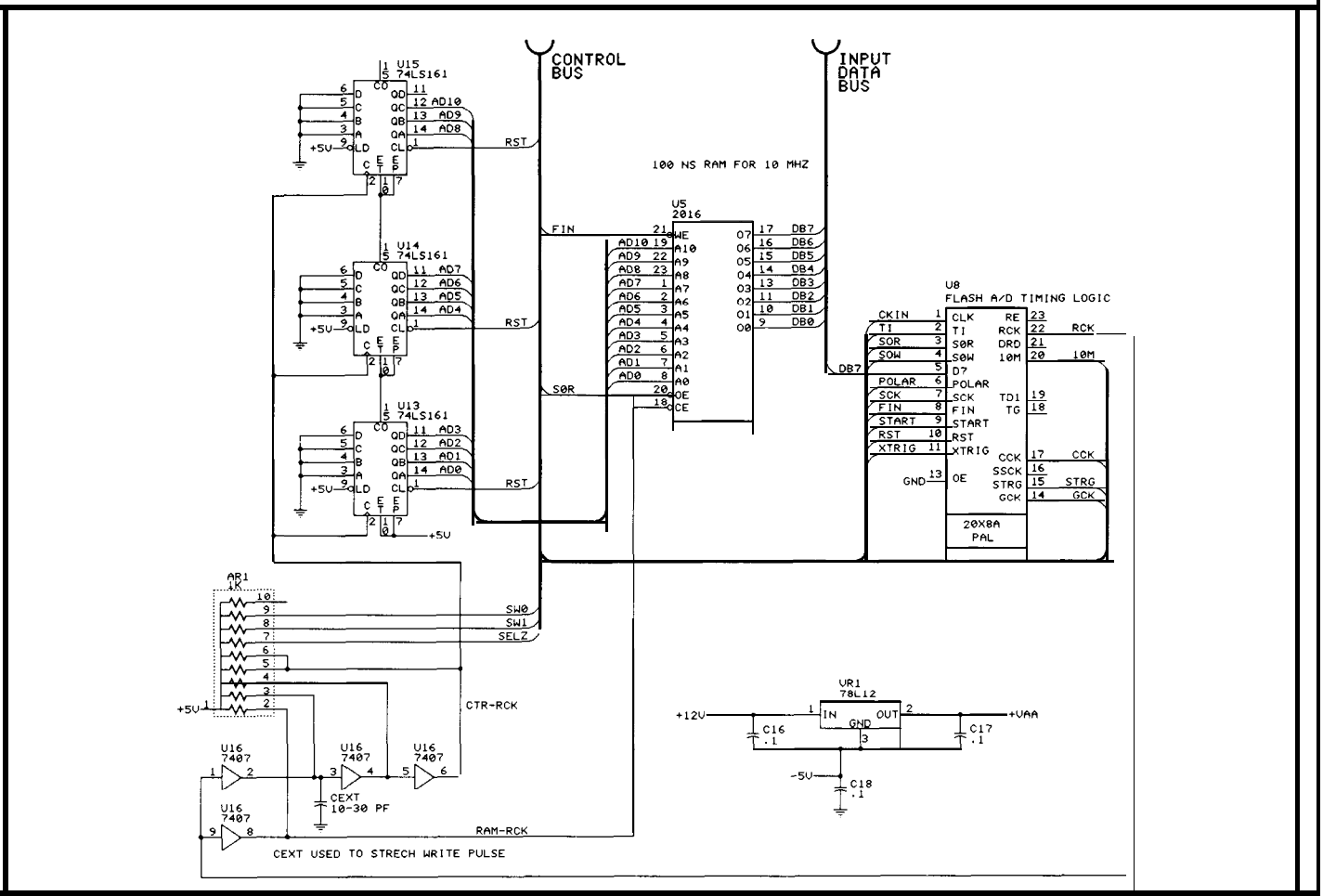


Figure 2 - ***The memory interfaces two signals: A fast data stream from the flash A/D converters, and a slow data stream from the PC bus.***

10-MHz period, or 50 ns. This time period low would be fine if we were using a 50-ns RAM, but with the standard version of the 2016 RAM used, the access time is 100 ns. How are these two timings reconciled? This is the purpose of the write pulse stretching circuit. Using U16 and an external capacitor, the time period that the RAM-RCK signal is low becomes extended by a simple RC delay, meeting the specifications for the write pulse width for a 100-ns RAM, while maintaining the 100-ns write-to-write time required for capture of the flash A/D data at 10 MHz. The 7407 also buffers the output of the timing PAL U8, so that the CTR-RCK signal will increment the address counters U13, U14, and U15 after the write pulse RAM-RCK goes high.

This design has programmable digitizing rates from a maximum rate of ten million to a minimum rate of 152 samples per second. This sixteen-bit range of frequencies, as well as both the FIN and START signals, is generated by the programmable counter IC, U10 in Figure 3. U10 is an 8254-2 IC, and has three independent counters. Each counter has a maximum input clock of 10 MHz, and a minimum divide of two. That gives a maximum output clock of 5 MHz for the SCK signal, the source clock for the digitizing rate. SCK is doubled in frequency by generating a 50-ns pulse on the RCK signal on each edge of SCK. Because the SCK frequency is doubled by the timing PAL U8, it is important that only even numbers are used for dividing down the IO-MHz clock to the programmable sampling rate desired, or a jitter of 100 ns will affect the timing accuracy of the digitizing. The trigger delay is also generated by U10. After a trigger has occurred, being latched inside the timing PAL U8, the GCK signal becomes active, mirroring the same pulses occurring on the RCK signal. The FIN counter section of U10 counts each pulse on GCK, keeping track of the number of writes to memory that have occurred since the trigger. This way the trigger delay function is ac-

complished, with a range of 1 to 65535 samples.

The interface of the digitizing board to the IBM PC uses three ICs: U6, a 74LS245 bus driver; U7, a PAL 20S10; and U11, a 74LS273 octal data latch (Figure 3). U6 isolates the board's data bus from the PC's data bus when the high-speed storage of data into the RAM occurs, but drives the PC's data bus during readout of the RAM. U11 is used to store the trigger comparison value for the digital trigger. U7, the I/O decoder PAL, determines when the board's address is on the PC's address bus, and generates the signals SOR, SOW, SEL1, and RST. The' board occupies a range of four addresses from the base address.

The base address of the board has four possible locations located in the I/O addresses allocated for a prototype board. The base address is selected by jumpers on the 16-pin header, SW0 and SW1. The default base address (as shown in Figure 3) is I/O address 768 (see Table 1 for the jumper positions). When the base address is read, the SOR signal is asserted by going low. SOR is used to enable U6 for data readout from the RAM, and to increment the RAM address at the completion of the read (if FIN is high). When the base address is written to, SOW goes low. SOW is used by the DSCOPE software to generate a software trigger, but SOW must be jumpered to the XTRIG input of the timing PAL U8 to work. When the base+1 address is either read or written, SEL1 is asserted, enabling reads or writes to U10. When the base+2 address is written to, RST is asserted, resetting the board, so that it begins digitizing the input signal and storing the data to RAM. When writing to RST it is necessary to include the trigger comparison value, for RST also stores this value into U1

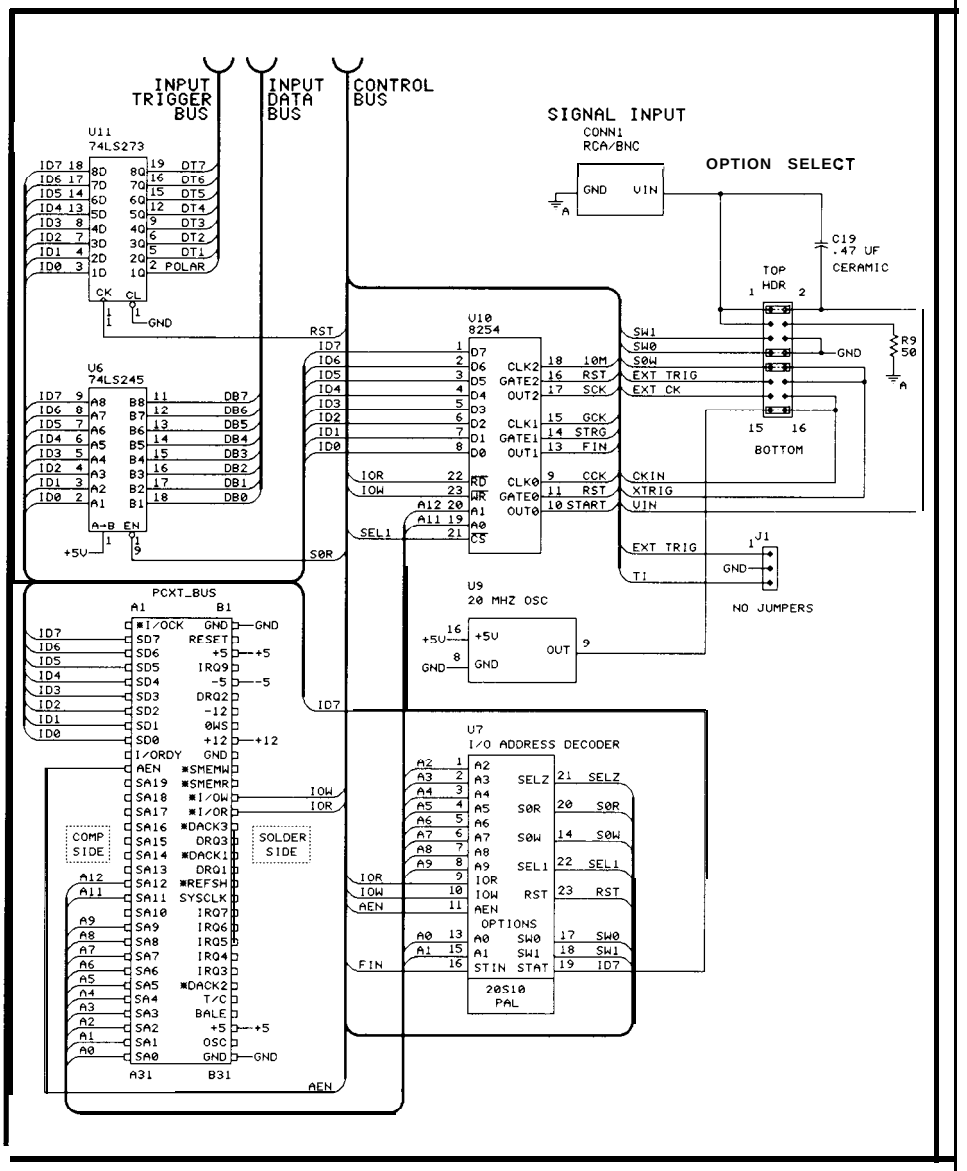


Figure 3 - & PC Interface

Table 2.

The I/O decoder PAL also has a one-bit tristate buffer that is connected to bit 7 on the PC data bus and is used to monitor the status of the FIN signal. The I/O location of the status is base+3.

In order to economize on the number of I/O addresses used by the board, the two upper address bits Al 1 and A 12, which are not decoded by the IBM PC but are still available on the address bus, are used tc

1. For access of U10 (refer to Table 3).

The addressing operation of tht RAM during readout deserve: boarc explanation. When the is capturing an input sign dresses constantly "wrap around,' so that when RAM address 2047 i: i:

readou

(remember status bit 7 is high), the firsM address points to the

Table 1 – Using jumpers, you select the base I/O address of the board

PC I/O Address	SW0	SW1	
768	---	---	--- =

Table 2 – I/O Signals, Addresses, and Functions

Signal Asserted	I/O Location	Function
SOR	BASE – Read Only	Read Data
SOW	BASE – Write Only	Software Trigger if jumpered
SEL1	BASE + 1 R/W	U10 programmable counters
RST	BASE + 2 – Write Only	Reset board : FIN goes low Write Trig. Comp. Value
(STATUS)	BASE + 3 – Read Only	Bit 7 (MSB) of PC data bus = FIN

Table 3 – Addressing and operation of 8254 Counter/Timer IC

PC I/O Address	Operation
BASE + 1	Write Counter 0 (START signal delay)
BASE + 1 + 2048	Write Counter 1 (FIN : trigger delay)
BASE + 1 + 4096	Write Counter 2 (SCK : 1/2 digitizing frequency)
BASE + 1 + 6144	Write Counter Control Values (See Note)

Note : See TDS.C test software for correct counter setup

byte of data in the buffer, that is, the earliest byte of the data stored. To read out the entire buffer, the software needs only to do 2048 reads of the base address. The 2049th read will be the same data as the first read. This is because the addresses wrap around during readout, so that the RAM operates as a FIFO buffer.

Test programs, TDS1.C and TDS2.C are available to clarify the basic software operations needed to run the digitizing board and as a debugging aid to those enthusiasts who prototype one. They were written for DeSmet C, but should work with most C compilers with little or no modification. Both programs are available on the Circuit Cellar BBS or on the INK Software Disk for this issue.

The digital oscilloscope should make your design and debugging sessions go more quickly and smoothly. In addition, the basic digitizer design presented here should be easily adaptable to a number of other uses. Have fun and, as always, be sure and let the editors of INK know if you come up with a really original application for this project.

UPDATE: *Additional information to previous articles*

The X-10 TW523 Two-Way Power Line Interface

A Step Toward Closed-Loop Power Line Control

by Ken Davidson

Just as I promised in Vol. 1, No. 3 of INK, I have an update on the status of an X-10 receiver module. Before I begin, though, let me fill in some information for those of you who may have missed the last article.

The X- 10 POWERHOUSE system is based on a concept known as carrier current communication. The AC wiring already strung throughout the house or building is used to carry control signals from command consoles to remote switch modules. These modules contain either a solenoid for switching heavy loads, or a dimmer circuit used to dim, brighten, and turn lights on and off. Each module can be set for any of 256 unique addresses, so a very elaborate control network can be set up (a typical control console can only address 8 or 16 modules, however).

In my previous article, I described a new module available from X- 10 (USA) Inc., the PL513, that allows a computer to directly access the AC power line. Using the X-10 transmission protocol, the computer then controls the timing and configuration of bits transmitted through the power line.

One of the biggest drawbacks of the X-10 system over the years has been its "open-loop" configuration and lack of two-way communication. There are quite a few devices around capable of transmitting X- 10 codes and plenty of receiver modules that provide control actions as a result. However, until now, there has been no convenient way to intelligently "listen" to the power line and "hear" all the X- 10 activity taking place.

With the introduction of the X- 10 POWERHOUSE TW523 Two-Way Power Line Interface, all that is changing. The TW523 not only contains the complete transmitter circuitry found in the PL5 13 module, but also contains all the necessary front-end circuitry to receive X- 10 codes as well.

Figure 1 shows the schematic for the receiver section of the

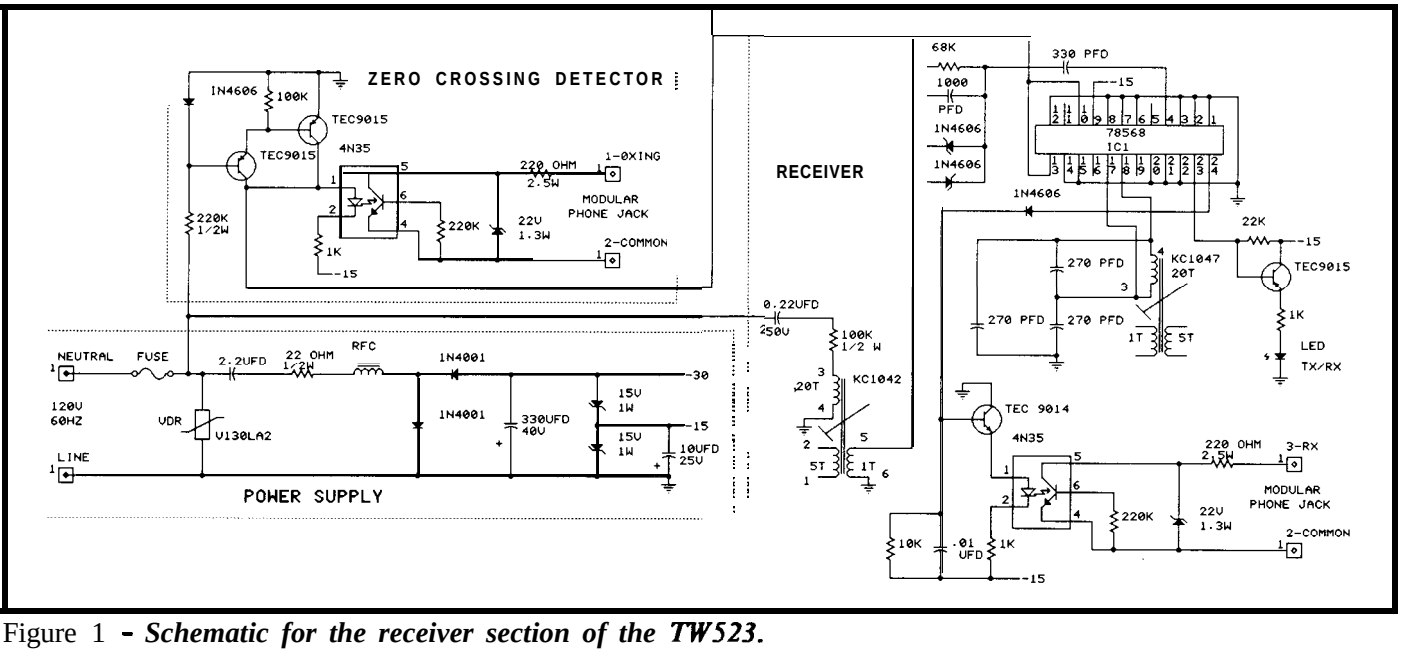


Figure 1 - ***Schematic for the receiver section of the TW523.***

TW523. Since the schematic for the transmitter section is nearly identical to that of the PL513 (which was published in the last article), I haven't reproduced it here.

As I described in detail in the last article, the X-10 protocol is made up of a start code, a house code, and a function code (which may be a module number or an actual function). One bit of the code is always sent out once immediately after the AC zero crossing and two more times to correspond with the zero crossings of two other AC phases. A "1" bit consists of a I-ms burst of a 120-kHz oscillator and a "0" bit is denoted by the lack of a burst. Except for the start code, a bit is followed on the next zero crossing by its complement. Finally, each start/house/function group is sent out twice with no delay between them. The top half of figures 2a and 2b show what a typical transmission looks like.

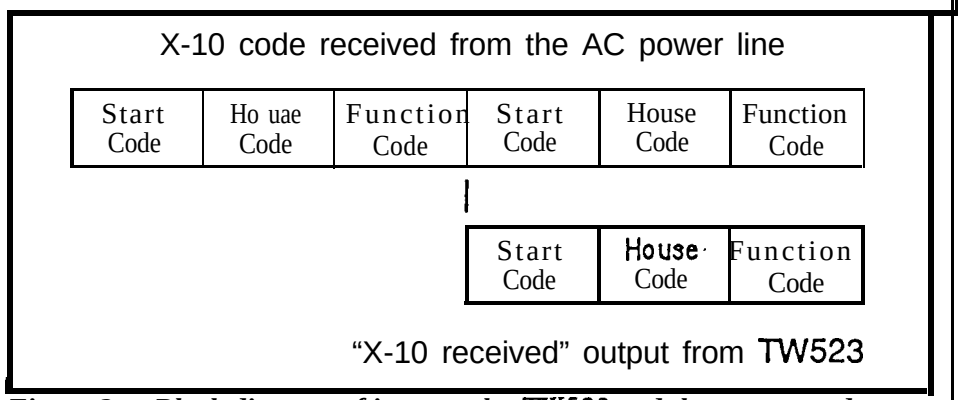


Figure 2a - Block diagram of input to the TW523 and the corresponding output.

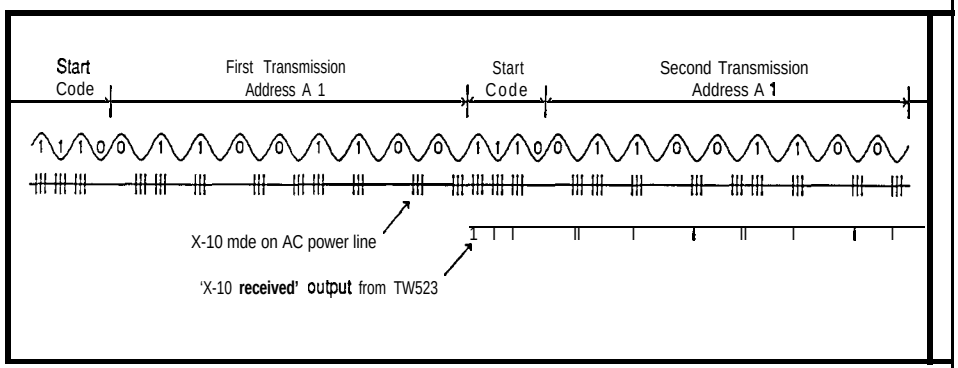


Figure 2b - ***The TW523 outputs a single I-ms pulse for every group of three 120-kHz pulses it receives.***

Imagine, if you will, a very crude receiver front end. It consists of a zero crossing detector, a filter which passes just 120 kHz, and some isolation circuitry. It is now up to the programmer to listen to the power line at the correct times and decide if a 120-kHz burst is present, if the length of the burst is proper, if the start code is good, if there is an error in the house code or function code, and so on. A great deal of time must be spent in error checking.

The TW523, however, adds some intelligence to the receiver front end. It takes care of listening for proper-length 120-kHz bursts and valid X-10 code. If it determines that what it hears isn't valid X-10 code, it won't act on it. When it does detect valid X-10 code, it presents the programmer with clean I-ms pulses coincident with the AC zero crossings. Don't expect the module to be smart enough to assemble groups of complete code transmissions, though. Something has to be left for the programmer to do.

Figures 2a & 2b show how the receiver output corresponds to the activity taking place on the power line. The TW523 listens to and validates the first occurrence of any X-10 transmission. If it determines the transmission to be error free, the receiver output will exactly match the second transmission of the code sequence. In figure 2b, each vertical "hash mark" on the transmission line represents a 1-ms, 120-kHz burst. Notice that each burst is repeated three times, as explained before. On the receiver line, each vertical hash mark represents a 1-ms pulse sent to the computer.

Since the transmitter software primitives are already written, the software to support the receiver falls into place rather quickly. We first wait for a zero crossing to occur, then wait 500 microseconds (half the duration of the l-millisecond pulse), and sample the receiver input line. That way we can be sure of sampling the center of the pulse. If it is high (no activity), we just sit back and wait for the next zero crossing. If it is low, we know a transmission is starting to come in, so we enter something of a state machine to assemble the rest of the received transmission.

Just briefly, a state machine is a hardware or software device which sits in a known state, able to move on to the next state based on one or more inputs. In the case of the receiver code, we know that we must first see a valid start code before we can move on. We can then assemble a house code, followed by a function code. If at any point, a single error is detected, we

	D1	D2	D4	D8	D16
1	0		110		0
2	1	1	1	0	0
3	0		0	10	0
4	10		10		0
5	0		0	0	10
6	10		0		10
7	0		10		10
8	1	1	0		1 0
9	0		1	1	1 0
l o		1	1	1	1 0
11	0		0		110
12	1	0		1	1 0
13	0	0	0	0	0
14	10		0	0	0
15	010		0		0

16	110	0		0
All Units Off	0	0	0	01
All Lights On	0	0		011
On	0	0	1 0	1
Off	0	0	1 1	1
Dim	0	1 0		0 1
Bright	0	10		11
All Lights Off	0	110		1
Extended Code	0	1	1 1	1
Hail Request	10	0		01
Hail Acknowledge	10	0		11
Preset Dim	1 0		1 x	1
Extended Data	110	0		1
Status=On	1	1 0	1	1
Status=Off	1	1 1 0		1
Status Request	1	1 1	1	1

Figure 3 - A *list of the existing twenty-two X-10 function codes plus ten newly defined codes. (x can be replaced with 0 or 1)*

throw in the towel and wait for the line to be quiet for at least three full line cycles before listening for valid code again. Since the front-end circuitry prechecks the incoming code, the chances of the host computer detecting an error are greatly reduced.

I'm not listing any software with this article, but I've posted on the Circuit Cellar BBS the HD64180 (280) code I wrote for the BCC180 to support the receiver section of the TW523 if anyone is interested. Support for IBM PC and BCC52 will be available when the TW523 module is actually in production.

X-10 Protocol Additions

An unexpected bonus found in the TW523 data sheet includes 10 new function codes designed to extend the usefulness of the X-10 system. I've listed these new codes along with the existing ones in Figure 3.

The more pedestrian additions are those meant to extend existing functions. "All Lights Off" (as opposed to "All *Units* Off") is one I've longed for in the past. With Preset Dim, the most-significant bit of the level replaces the "x" in the binary function code and the four least-significant bits of the level are placed where the house code is normally found. With a single code transmission, the target module can be told to what level to dim. Currently, the power line is tied up while the transmitter sends multiple dim commands.

More exciting are the completely new functions which rely, for the most part, on the existence of two-way modules. Hail Request is sent to determine whether any other transmitters are within listening range. Any such transmitters would send out a Hail Acknowledge in response.

Extended Data is sent to signal that additional 8-bit data follows. Presumably such data would be from temperature sensors or perhaps alarm contacts. Extended Code can be used to continue making additions to the X-10 command set even though all the primary command codes are used (similar to the way Zilog extended the 8080 instruction set for the 280).

All of the above additions are still only on paper, though. Computers using TW523 modules can implement most of the functions, but existing modules won't respond to, say, "All Lights Off." The last three added codes, however, have been implemented in at least one module already on the market.

When a module is selected and a Status Request is sent to it, the module is supposed to respond with either Status = On or Status = Off. The X-10 POWERHOUSE Radio Controlled System's transceiver module (RR50 1) already supports the status function. The RR501 receives codes sent via R.F. from the hand-held RT504 radio transmitter and retransmits the codes over the power lines. It, like the TW523, is a true two-way module. I've set up the TW523 to listen and have seen the RR501 respond with its status when requested, so it really works. Now if the rest of the modules would only do the same...

The Future

What does the future hold for the X-10 line of products? If the latest releases by X- 10 are any indication, we are looking at some exciting times. With the introduction of the new function codes and (hopefully) their eventual implementation, X- 10 will finally develop into a true closed-loop, full-duplex system upon which even more powerful home control systems (and control systems in general) can be based.

The TW523 is an important first step in that direction. It will now be much easier for system designers to design two-way X- 10 operation into their products, perpetuating the line (you can be sure this is X-10's main motive for developing such interface modules).

As I've said before, we plan to do a lot more with these new power line interface modules. I'll keep you informed as anything new develops. ■

Special thanks to Dave Rye for his contributions to this article.

UPDATE: *Additional information to previous articles*

The DDT-51 Lives!

Fixes, Updates, and Future Plans for the Low-Cost 8051 Development System

by INK Research Staff

e knew that the DDT-51 project in BYTE ("Ciarcia's Circuit Cellar: Why Microcontrollers?, Part 1," August, 1988; "Ciarcia's of Circuit Cellar: Why Microcontrollers, Part 2," September, 1988) was a good one, but we weren't prepared for the flood of Circuit Cellar INK subscriptions, letters, and telephone calls that have resulted. It seems like everyone wants more information about the low-cost 805 1 development system!

In this issue, we have reports on bugs that our staff and astute readers have found. There are also several suggestions for improvements or modifications to the basic DDT-5 1 design.

If you want more info on the DDT-51, you'll need to read Circuit Cellar INK in the coming months. In the November/December issue, we'll have an article on the innermost secrets of the system, additional software for increased functionality, and continuing reports on availability of components. With the DDT-51, as with other projects and systems, Circuit Cellar INK continues its tradition of bringing the best in practical design information to the most important people in the microcomputer industry ... our readers.

The following schematic fixes apply to Figures 4a and 4b on pages 306-309 of the September 1988 issue of BYTE:

Page 306, Figure 4a -- Locate the tan box containing the DB-25 connector labeled "TO IBM." The wire going to pin 18 of the DB-25 connector should instead go to pin 17. Pin 18 should be connected to ground.

Page 309, Figure 4b -- The signals on IC2 and IC3 labeled AHIO-AH17 should be labeled **AHI8-AHI15** to match similar labels on connector **S1** (i.e., change **AHI0** to AH18, **AHI1** to AH19, and so on).

Page 307, **Figure 4a** -- The signal going to pin 20 **IC1** should be "*RAM READ" and the signal going to pin 18 should be "*RAM CS/ADDR HIT" (i.e., swap the wires going to pins 18 and 20).

Page 308/309, Figure 4b -- The signal going to pin 22 of **IC7** should be *PS and not *RD.

Page 307, Figure 4a -- The signal going to pin 21 of **IC1** should be *XWR and not *WR.

IC1-21 is *WE instead of WE
IC1-20 is *OE instead of OE
IC1-18 is ***CE** instead of CE
IC1 l-4 is ***G2A** instead of G2A
IC1 l-4 is ***G2B** instead of G2B
IC7-27 is ***PGM/*WE** instead of *PGM/WE
IC7-20 is ***CE/*CE2** instead of ***CE/*CS2**

Page 308, Figure 4b -- Capacitor Cl in the EPROM programmer circuit may need to be increased in value to reduce noise in the circuit depending on the techniques used to construct the circuit.

Page 303, Photo 1 -- The photograph of the **DDT-5** 1 board shows two white switches, while the schematic found in the article shows just one switch. The extra switch on the prototype is used to turn off **+5V** to the Vpp and Vcc pins of the EPROM shown in figure 4b on page 308. By turning off **+5V** to Vpp **(IC7-1)** and Vcc **(IC7-28),** it's possible to safely remove the EPROM from the socket without turning off power to the entire

DDT-5 1 board.

Optional **change #1:**

IC7-27 (*PGM/*WE) to IC12-2 (*XWR)
IC7-22 (*OE) to IC15-8 (*RAM READ)

The prototype hardware is actually wired this way, but the signals are equivalent during EPROM programming because "*CTLS TO 803 1" is active. This should have no effect on anything. ..

Optional change **#2:**

IC15-4 to +5V instead of IC14-9
IC15-2 to +5V instead of IC14-7

This will reduce the loading on the pull-ups and improve the rise time of the signals. There should be no effect, but that's how the prototype is wired.

Diagnostic Program Change

Add one line to BUS.TST and recompile TESTER.COM:

```
> this is about line 150 of BUS.TST <
{ MUX addr = 00, disabled                          }
  SetCR(CtlTo803 1 ,ON);          ( MUX B (MSB) in }
  Load8255(I55PB,AddrHiOK);{ MUX A (LSB) in        }
  SetSR(XWR,ON);                  ( 1 - 1  & 2-2 = 0 }
  SetSR(XPSEN,OFF);               { allow XRD gating }
  SetSR(XRD,ON);                  { 1-2 & 2-1  = o   }
> Add this line <
  SetSR(XALE,OFF);                ( allow XWR gating)
```

This ensures that the Data To/From 803 1 logic tests will run correctly regardless of the preceding tests.

WRITE FOR INK!

Writing technical articles may not make you rich and famous but it might be just the incentive to finish that 100-MIPS computer you started last summer. Or, if your expertise is software, perhaps it's time you presented your talents to the world.

Unlike most narrowly specialiced publications, Circuit Cellar INK's charter is to cover a wide variety of hardware and software technology and ideas.

Send your project outline to:

Curtis Franklin, Jr.
Circuit Cellar INK
P.O. Box 772, Vernon, CT 06066
or contact him on the Circuit Cellar BBS at (203) 871-1988.

CONNECTIME *Excerpts from the Circuit Cellar BBS*

THE CIRCUIT CELLAR BBS
300/1200/2400 bps
24 hours/7 days a week
(203) 871-1988 -- 4 incoming lines
Vernon, Connecticut

Conducted by Ken Davidson

The message base of the Circuit Cellar BBS is now available on disk. See page 44 for details.

Response to the changes we made to the Circuit Cellar BBS software a few months ago has been overwhelmingly positive. TBBS is far easier than our old software for most people to use, the response time is much faster, more messages may be kept on-line at any given time, and the multiline capability has been a blessing.

One of TBBS's more useful capabilities is the ability to search the message headers for a full or partial string. The thought of wading through over 3000 active messages to find a few on a particular topic is not all that appealing. Fortunately, with the use of some creative search strings, the task is made much simpler.

Let's look at an example. At least once a month, someone pops in and asks everyone's opinion on what schematic/PCB layout software is the best. The responses are usually either identical to those everyone gave the previous month or else no one answers since they're tired of repeating themselves. A more efficient method to get good answers is to search the message base for any previous conversations before asking in the first place.

First, go to the public message areas from the main menu by pressing M. Next, press A to select all the message areas. Press R to read messages, S for selective retrieval, then S to retrieve by subject. Now is where you have to get creative. Try to think of some short string that is unique to the subject you're looking for, but not so unique that you'll miss potential matches. Partial strings often work better than full strings. To look for schematic/PCB layout programs, try searching for "CAD," "PCB," and "SCHEMATIC." When I used these three search strings, I found a total of fourteen separate message threads (each thread made up of several messages), with eleven of the threads dealing with the topic I was looking for.

As you can see, taking a few minutes to search the current messages first can save you time by possibly finding an answer to your question without having to ask it. Plus, it saves the regular callers from having to read a question which has been asked and answered several times in the past. If you don't find what you're looking for by searching, though, by all means post your message and we'll see what we can come up with.

We have often heard members of the 68000 community lament the lack of 68xxx-based Circuit Cellar projects. Rest assured, such projects are coming down the line and will eventually be published in INK (the first of which is Mark Voorhees' home weather center). Even though the processor family has been around for a while now, there is still some mystery and misinformation surrounding its features and performance. The following thread attempts to clear up some confusion and myths.

Msg#: 3837 *GENERAL*
From: MARK BALCH
To: STEVE CIARCIA
Subj: 68010 VS 68000

I have two questions about Motorola's 68010 microprocessor. First, what are the basic differences between the 68010 and the 68000. In other words what is so great about the 68010? And second, what is meant by the 68010's "virtual addressing"?
--Mark

Msg#: 3864 *GENERAL*
From: STEVE CIARCIA
To: MARK BALCH
Subj: REPLY TO MSG# 3837 (68010 VS 68000)

Any 68000 gurus out there want to tackle this one before I put my foot in my mouth and show my 68000 ignorance (notice I haven't done any 68000 projects)?
-- Steve

Msg#: 3870 *GENERAL*
From: BOB PADDOCK
To: MARK BALCH
Subj: REPLY TO MSG# 3837 (68010 VS 68000)

From Motorola's "Master Selection Guide": (1988 p 13) "MC68010: A Virtual Memory Enhancement. The internal architecture, instruction set, and bus structures are identical to the MC68000, but it offers the advantage of Virtual Memory. A high-speed loop mode operation executes tight software loops faster to enhance performance. Its instruction continuation feature has made it the choice for fault-tolerant and parallel processing systems. The MC68010 can support a governing operation system which handles the supervisory chores of any number of subordinate operation systems."

I saw a couple of messages on an Atari (1040ST) BBS where the people were proposing replacing the 1040's 68000 with a 68010 to gain

a small increase in speed, but I don't have a 68010 data sheet to look at to see if, in fact, they are even pin compatible.

As for "Virtual Memory," in the simplest, generic sense, a memory management unit would make a disk drive (or some other type of mass storage device) look like it had RAM the sise of the storage device as far as the CPU was concerned.

Msg#: 4303 *GENERAL*
From: MARK BALCH
To: BOB PADDOCK
Subj: REPLY TO MSG# 3870 (68010 VS 68000)

Thanks. I, too, looked in Motorola's MSG and saw that passage. What I did not fully understand was virtual memory, which you have helped me with just now. I guess that when I need to I will get a hold of the data sheets for the two MPUs and take a closer look.
--Mark

Msg#: 3877 *GENERAL*
From: BOB PADDOCK
To: MARK BALCH
Subj: REPLY TO MSG# 3837 (68010 VS 68000)

From "M68000 16/32-bit Microprocessor Programmer's Reference Manual (4th edition)": "In most systems using the MC68010 as the central processor, only a fraction of the 16-megabyte address space will actually contain physical memory. However, by using virtual memory techniques the system can be made to appear to the user to have 16 megabytes of physical memory available to him/her. These techniques have been used for several years in large mainframe computers and more recently in minicomputers and now, with the MC68010, can be fully supported in microprocessor-based systems.

"In a virtual memory system, a user program can be written as though it has a large amount of memory available to it when only a small amount of memory is physically present in the system. In a similar fashion, a system can be designed in such a manner as to allow user programs to access other types of devices that are not physically present in the system such as tape drives, disk drives, printers, or CRTs. With proper software emulation, a physical system can be made to appear to a user program as any other computer system and the program may be given full access to all of theresources of that emulated system. Such an emulated system is called a virtual machine.

"1.4.1 Virtual Memory: The basic mechanism for supporting virtual memory in computers is to provide only a limited amount of high-speed physical memory that can be accessed directly by the processor while maintaining an image of a much larger 'virtual' memory on secondary storage devices such as large-capacity disk drives. When the processor attempts to access alocation in the virtual memory map that is not currently residing in physical memory (referred to as a page fault), the access to that location is temporarily suspended while the necessary data is fetched from secondary storage and placed in physical memory; the suspended access is then completed. The MC68010 provides hardware support for virtual memory with the capability of suspending aninstruction's execution when a bus error is signaled and then completing the instruction after the physical memory has been updated as necessary."

What are you planning on using the 68010 for?

Msg#: 4364 *GENERAL*
From: MARK BALCH
To: BOB PADDOCK
Subj: REPLY TO MSG# 3877 (68010 VS 68000)

Whew!! That was a lot. Thank you for going to the trouble to get that! Later, I will take a closer look at this message (I downloaded it) and fully digest it.

Finding information on electronics is not the easiest thing in my position (student) and I will take just about anything I can get!

As for my application, I am still struggling with Motorola's 8-bit MPUs and have several more years to go until I hit 16 bits! I have a rather far view of things and am enthusiastic about it. Eventually I plan to design a 16-bit micro. I am probably getting in above my head now but give it another five years or so and ... Thanks again for the information.
--Mark

Msg#: 4341 *GENERAL*
From: BOB PADDOCK
To: MARK BALCH
Subj: REPLY TO MSG# 4304 (68010 VS 68000)

In five years we might be using 128-bit parts! I've found (for me at least) the best way to learn about how to use a new chip is to pick out a project for it, then do it. You learn all of the little nasty things that aren't in the data sheet that way. You might be in over your head when you start, but when you're done you'll be an expert.

Msg#: 4018 *GENERAL*
From: JONATHAN STOTT
To: BOB PADDOCK
Subj: REPLY TO MSG# 3877 (68010 VS 68000)

The 68010 is fully compatible with a 68000 (I know people who have swapped them on Amigas). There is a programming difference between the two, however. One opcode has been dropped (I think it is MOVE SR,ea or something similar). You will probably want to find out if this will cause a problem before you swap the chips.
--Jonathan

Msg#: 4100 *GENERAL'
From: BOB PADDOCK
To: JONATHAN STOTT
Subj: REPLY TO MSG# 4018 (68010 VS 68000)

I'm not sure it was dropped. It might have been made into a supervisor-state instruction, where it was a user-state instruction in the 68000. The 68010 data sheet does describe the reason; it had to do with MMU memory protection. Did the people who swapped processors think it was worth doing it after they tried it?

Msg#: 4260 *GENERAL*
From: JONATHAN STOTT
To: BOB PADDOCK
Subj: REPLY TO MSG# 4100 (68010 VS 68000)

They were split (two people). One liked the faster text scrolling it gave. The other did not feel there was enough of a speedup to be worthwhile. I should note that there are many custom chips on an Amiga that are set at 7.14 (?) MHz and it doesn't matter which microprocessor is installed, they are going to stay at the one speed. I don't know what system you have or if this would apply. **You may be right about the command becoming a privileged instruction. Either way, it is not available to the programmer. Claims of epeedup in magazines range from 3-15%. I imagine the amount of math being processed could make a big difference in these numbers. Hope this helps.**
--Jonathan

Msg#: 4699 *GENERAL*
From: CLAYTON ZEKELMAN
To: STEVE CIARCIA
Subj: REPLY TO MSG# 3864 (68010 VS 68000)

The question is, *why* would anyone *want* to do 68000 projects? Or 68xxx projects for that matter? I still haven't seen a

system with any 68xxx processor that actually runs quickly. Harris Corp. knows what to do with the 68k -- they use it as a small boot loader system for their HCX-7 RISC-based computer. It also runs the console's serial port (the slowest one on the system). I think one of the engineers over there was beingverycruel to Motorola when he did that. >>Clayton<<

Msg#: 4721 *GENERAL*
From: MARK BALCH
To: CLAYTON ZEKELMAN
Subj: REPLY TO MSG# 4699 (68010 VS 68000)

I don't claim to be a 68000 guru (to use Steve's phrase), but Motorola has been shortchanged by IBM using the 80x86 for all of its computers. Many people don't like the 80x86 family but are forced to use it because that is what Big Blue uses. If more people took a closer look at the 68000 then there would probably be more systems using it that run *fast*. I'm not saying that everyone would like it, but certainly not everyone loves the 80x86s!
.Mark

Msg#: 4715 *GENERAL*
From: JONATHAN STOTT
To: CLAYTON ZEKELMAN
Subj: REPLY TO MSG# 4699 (68010 VS 68000)

Some of us are interested in doing 68xxx projects because that is what we have for CPUs in our computers (i.e., Amiga/ST/Macintosh)!
--Jonathan

Msg#: 4722 *GENERAL*
From: MARK BALCH
To: JONATHAN STOTT
Subj: REPLY TO MSG# 4715 (68010 VS 68000)

At least someone speaks out! Not only is the 68000 used in those systems but it and its brothers (68010, 68020, 68030) are GOOD MPUs!
..Mark

Msg#: 4819 *GENERAL*
From: BOB PADDOCK
To: CLAYTON ZEKELMAN
Subj: REPLY TO MSG# 4699 (68010 VS 68000)

This is taken from EMU.TXT in CPM.ARC (CP/M emulator for the 68000-based Atari 1040ST) by Gerd Hildebrandt, Herbert Thiess:

With the growing demand for software flexibility, the limits of the 8-bit generation, especially the 64K addressing range, were encountered. The first microprocessor of the new 16-bit generation, the Intel 8086, was quickly pushed into the market. Its instruction set and register structure were designed as an extension of the 8080/8085. So it was possible to transfer existing programs easily to the 8086. The limitations of the 64K addressing range were overcome, [if they had been overcome then we wouldn't have all this junk about Small, Large, and Huge model programs] but a lack of orthogonality and addressing capability in the architecture remains. The successor 80186 was extended by some hardware features, but the poor instruction set was retained. The same applies to the 80286 in its compatibility mode. The new '286 mode is hampered by design flaws on the chip. [Probably fixed by now?]

Other microprocessor developers did not insist on upward compatibility and took the chance to introduce new concepts of CPU architecture with their 16-bit generation microprocessors. Thus the 68000, even though it came late, is now established as an alternative to the 8086. The 68000 family was extended by the more powerful 68010 and the 32-bit processor 68020 [and now the 68030 and 880001.

Today the 16-bit micro market is mainly divided between two opponents: on one side the conventional 8086 family with a large software base and on the other side the advanced 68000 family with less software as yet available. In contrast to the 8-bit generation, in the 16-bit generation no microprocessor and therefore no operating system has become established as a standard. [MS-DOS has become the de facto standard since this was written.] The ongoing confusion about future trends leads to uncertainty for customers and blocks investment in both new hardware and software. [So why is it that Sun workstations use 68020s and not 8086s or 80286s? Could it be performance?]

Msg#: 4834 *GENERAL*
From: ALAN GOLDSTEIN
To: CLAYTON ZEKELMAN
Subj: REPLY TO MSG# 4699 (68010 VS 68000)

Just to throw my two cents in, one reason that I use the 68xxx (and 6809) is the OS-9 operating system. It provides multiuser, multitasking support with pipes, filters, interprocessor communication (sounds like OS/2 -- the biggest example of vaporware to date) with modular structure and configurable support for I/O. OS-9 has been around for over 10 years and is very Unix-like, but is much smaller and less disk intensive. Since it is only available for the Motorola families, there are a lot of PC users that don't know what they are missing.

Msg#: 4976 *GENERAL*
From: DAVID HESSLER
To: ALAN GOLDSTEIN
Subj: REPLY TO MSG# 4834 (68010 VS 68000)

Alan, it really bothers me to catch you preaching the good OS-9 word to so many unbelievers. Radio Shack, no less, offered OS-9 for the 6809 color computer for $69 years ago, and had all the best minds in mainframe architecture and firmware/OS design loving it. It allowed four users on an 8-bit machine to run a board and lab control systemwhile they edited code. On the new systems like Mentor, Apollo, Sun, and Mac II it offers so much more than any other system that it is impossible to put it all in a dozen articles. And in which journals do you place those articles since it is equally important to the starter hackers and the super Unix buffs? Variants are available in C to do real-time avionics for fighters and the LHX, Blue Thunder chopper silicon copilot and incoming fire control officer while doing damage control and navigation in an integrated multiprocessor attack aircraft system. I am glad someone came forward to answer the critics of OS standards today. --Dave

HAL, the Hemispheric Activation Level detector presented by Steve in the June and July 1988 issues of BYTE, was met with a great deal of enthusiasm right from the day it appeared. There was a good deal of BBS traffic surrounding the project and the following is a sampling of that traffic.

Msg#: 3821 *PROJECTS*
From: STEVE HONODEL
To: STEVE CIARCIA
Subj: COMPUTERS ON THE BRAIN

I enjoyed reading your article "Computers On The Brain." I've been looking at the available boards for data acquisition and am most interested in one like yours because of its microvolt sensitivity and RS-232 interface. I have a Datavue Spark laptop and would like the capability of remote sensing. I work in aerospace engineering and do some strain gauge measurements that need high-gain, low-frequency

acquisition. Can your circuit be adapted to include DC measurements? Also, what is the maximum practical sample rate if only raw data collection is performed (no display or calculations)?

I have another motivation for buying your kit. The company I work for is planning to do polygraph testing on many of their employees and I am curious how this EEG setup might prepare me for such an awful experience (I may decide not to submit to it!). Do you have any ideas on how to adapt the circuit for sensing galvanic skin response, blood pressure, or heart/respiration rate? I bet that might be a popular application to write about considering all the flack about it in the news. That might also put one more nail in the polygraph coffin and bring more power to the people in defending our right to privacy, much like the personal computer has done for information.
..Steve

Msg#: 3888 *PROJECTS*
From: STEVE CIARCIA
To: STEVE HONODEL
Subj: REPLY TO MSG# 3821 (COMPUTERS ON THE BRAIN)

I'm personally opposed to widespread polygraph testing. HAL could be modified I suppose but it will take some calculation. Right now, its range is 4-20 He. To respond to EMG, galvanic, blood pressure, etc. you need to lower the frequency range to 0 Hz. This can be done by changing the filter components (I have not done these calculations) and changingsome values. I don't think it will require wiring changes, just values. Of course, HAL was designed as an EEG sensor. I don't know how good or accurate it will be as an EMG sensor. Finally, if you disconnect the four probe amplifier inputs to the A/D chip (ADC0808), you could insert other analog inputs and the software that sends it out serially wouldn't know thedifference. The A/D has its eero point at 1/2 Vcc (about 3V) and has a range of +/-1.25V above and below this point. If your strain gauges can be set to this range it will work fine.
-- Steve

P.S. Even though the A/D is 8-channel, the present EPROM code only supports four channels ('cause it would slow down the serial communication too much).

Msg#: 3884 *PROJECTS*
From: ROBERT WELKER
To: STEVE CIARCIA
Subj: HAL

Would it be hard to get HAL to turn on some sort of device (i.e., stereo, cassette, etc.) and control volume proportionally with the level of relaxation (grabbing the audio between cassette and amp, say)? I was having this odd thought of using HAL as an element in a (what would you call it?) "human process-control feedback loop." When you were researching it did you turn up anybody working in that sort of area?
-- Bob

Msg#: 3993 *PROJECTS*
From: JEFF BACHIOCHI
To: ROBERT WELKER
Subj: REPLY TO MSG# 3884 (HAL)

Robert,

The HAL project simply collects data. The demo software for the IBM PC stores and analyzes this continuous stream of information. The results are displayed on the PC's monitor. If you felt ambitious, you could write some code to take action on the output level of a particular frequency of interest. The whole purpose of this project was to put equipment in the hands of individuals who, if they had some hardware, would like to experiment in this field. This is virgin territory for most of us. Let's see what we can come up with.
..Jeff

Msg#: 3934 *PROJECTS*
From: STEVE CIARCIA
To: ROBERT WELKER
Subj: REPLY TO MSG# 3884 (HAL)

Like Jeff said, we don't know what is coming out of this thing but we know it is doing it well. :-) Play with it and you tell us whether it can be used for control.
-- Steve

Msg#: 3955 *PROJECTS*
From: ROBERT WELKER
To: JEFF BACHIOCHI
Subj: REPLY TO MSG# 3903 (HAL)

Sounds good by me! It's nice to still have the opportunity to be a "gentleman scientist" a la earlier investigators in our basic fields. Was getting worried that there were fewer and fewer things an individual could do new, exciting research on that didn't require big bucks or a used cyclotron or two. Keep up the good work. (I guess there'll be a couple people on this board to help when delving into this, huh? <grin>).
-- Bob

There are many methods that can be used to restrict access to rooms or areas of a building to only certain people. While it might be nice to be able to use HAL to characterize each person's brain patterns. the real world does prevail. The following thread deals with one person's attempt to keep room access convenient while making it more secure.

Msg#: 4079 *GENERAL*
From: JOHN MEALEY
To: MOST EVERYBODY
Subj: RANDOM DIGITAL DISPLAY KEYBOARD

I am trying to find/make/buy (without too much $$$) a 12-key keypad that would allow scrambling of entry codes for a secure door.

Background: I have a room at work where entry by about only three people is needed, yet others seem to always get the codes to get in by looking at the person keying in the security code. All I would need is some way of changing the positions of the keys (electronically, by LED display), and then have it randomly assign each key its position. Then, just because you pushed a certain sequence of keys does not mean that the same sequence of keys gets you in the next time (just the same pass code).
--Mealey

Msg#: 4114 *GENERAL*
From: KEN DAVIDSON
To: JOHN MEALEY
Subj: REPLY TO MSG# 4079 (RANDOM DIGITAL DISPLAY KEYBOARD)

Why not try something besides keyboard entry? There are plenty of card readers on the market as well as some "keys" that have electronic codes embedded in them instead of cut-outs. They are definitely more expensive than a standard keypad, but may be in line with the cost of a keypad that can randomly change its keycap legend.

Then there's the video hand scanner presented in the first issue of Circuit Cellar INK ..;-)

Msg#: 4121 *GENERAL*
From: JOHN MEALEY
To: KEN DAVIDSON
Subj: RANDOM KEYPAD

I suppose I will have to settle for a card/key type of door lock. The thought was good, though. There must be a way to accomplish the feat, but as you point out it is more cost-effective to just get used to an access card of some type. To some degree I would be really content if people would make an honest attempt to shut the door. At times I find it kept open by the carpet, clothes hangers, books, etc. I have a hard time convincing people around here that the wire room/computer room needs to have a little more security than the bathroom. I'll check into the lock catalogs. 'Later & thanks for the info.

Msg#: 4146 *GENERAL*
From: ROBERT EUGSTER
To: JOHN MEALEY
Subj: REPLY TO MSG# 4079 (RANDOM DIGITAL DISPLAY KEYBOARD)

John,

Aritech Distribution sells Digital Scrambler Keypads like you want. Each time a start button is pressed the digits are randomly positioned in one of over "3.6 million patterns." The viewing angle is limited to 4 degrees horizontally and 26 degrees vertically. This should keep anyone nearby from reading the access code. Price? A keypad runs about $275; a control panel that controls up to four access points and up to five user codes will run about $300. (It is possible that a BCC-series computer could interface to the keypad, but I don't know.) They also sell more sophisticated control panels (at a higher cost, of course). If you call (800) 432-3232 you will be automatically connected with your nearest distribution center (17 in the U.S.). The catalog is filled with all sorts of neat stuff. Aritech sells to dealers and wholesalers but you can probably still get what you want.
-- Robert

P.S. If you want more info drop me a line Email (anyone else too). Hope this helps.

Msg#: 4642 *GENERAL*
From: JOHN MEALEY
To: ROBERT EUGSTER
Subj: REPLY TO MSG# 4146 (RANDOM DIGITAL DISPLAY KEYBOARD)

Been away in SF for a week. Over good, hot Seechwan at Brandy Ho's, we talked about various ways to make cheap random digital keypads that would fulfill the requirements. One person brought up the unpleasant idea that unless many were made (economies of scale) it would be best to purchase the manufactured one that is in most security-related rags these days from $200 to $500+. Another suggested a simpler but less elegant approach: have a soft chime go off each time the door is left open for a determined length of time, coupled with a small plate over the existing simplex mechanical lock to cover "combo-droppers." We are opting for this option and will have it installed by the next weekend. Both of my friends mentioned that the most efficient way to accomplish something may not be the "funnest" or the most technical way. I had hoped to implement severe penalties for the door-left-open violation, but, alas, all I could get from management was flogging. It's been great.

--Mealey

One request we often get on the Circuit Cellar BBS is to have both the article-related software and the text of the message base available on disk. Due to the number of such requests and the time involved to fill each one, I've had to turn down everyone who's asked. The requests keep coming in just the same.

To remedy the situation, we are starting a service to provide both software and messages on disk. Starting with this issue of INK, all the software related to the articles in the issue is available on a 360K, 5.25" IBM PC-format disk. In the case of this issue, that includes the PAL logic for the home satellite weather center, the DSCOPE and PLOT programs for the IBM PC digitizing board, the sample TW523 code for the BCC180, and more. Each issue of INK will have a companion disk available that can be ordered for $12 per disk.

In addition to software on disk, we are also making available on two 360K, 5.25" IBM PC-format disks two month's worth of messages from the Circuit Cellar BBS. Messages from all three public message areas are included and span the two months previous to the cover date of the issue of INK. For example, the cover date of this issue is September/October 1988. The messages available for this issue are those posted during the months of July and August. This will allow us time to prepare the disk plus lets us include follow-up responses to questions posted near the end of the two-month period. Each two-disk set costs $15.00.

While we certainly continue to encourage you to call the Circuit Cellar BBS and exchange ideas with other INK readers, I think you'll agree that the cost of the disks is easily less than your cost in time and long-distance charges to download the same material. See the box below for details on how to order.

The Circuit Cellar BBS runs on a IO-MHz Micromint OEM-286 IBM PC/AT-compatible computer using the multiline version of The Bread Board System (TBBS 2.0M) and currently has four modems connected. We invite you to call and exchange ideas with other Circuit Cellar readers. It is available 24 hours a day and can be reached at (203) 871-1988. Set your modem for 8 data bits, I stop bit, and either 300, 1200, or 2400 bps.

SOFTWARE and BBS AVAILABLE on DISK

Software on Disk

Software for the articles in this issue of Circuit Cellar INK may be downloaded free of charge from the Circuit Cellar BBS. For those unable to download the files, they are also available on one 360K, 5.25" IBM PC-format disk for only $12.

Circuit Cellar BBS on Disk

Every month, hundreds of information-filled messages are posted on the Circuit Cellar BBS by people from all walks of life. For those who can't log on a often as they'd like, fhe text of the public message areas is available on disk in two-month installments. Each installment comes on two 360K, 5.25" IBM PC-format disks and' costs just $15. The installment for this issue of INK September/October 1988) includes all public messages posted 6uring July and August, 1988.

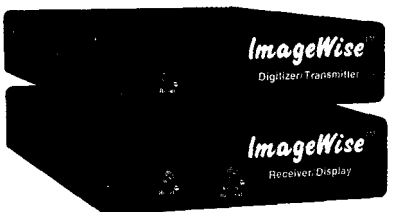
To order either Software on Disk or Circuit Cellar BBS on Disk, send check or money order to:

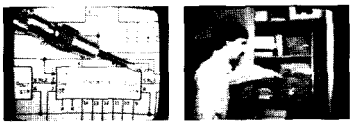
Circuit Cellar INK
Software or BBS) on Disk
P.6 Box 772
Vernon, CT 06066

or use your Mastercard or Visa and call (203) 875-2199. Be sure to specify the issue number of each disk you order.

Circuit Cellar Inc. kits are a proven vehicle for accomplishing a very special goal. With well designed circuits, pretested key components, documentation, and a knowledgeable support team you can have the thrill of making something you built yourself actually work! This is a CCI project! Call (203) 875-2751 to order your kit or for information.

- Serial Digital Imaging System





unretouched photos

The Circuit Cellar **ImageWise** Serial Digital Imaging System was designed to function intelligently as a stand-alone **digitizer** or as an integral component of a complete **tele-imaging** system.

ImageWise Transmitter Full kit.....

ImageWise Receiver Full kit ....

Both Units purchased together .....

ImageWise Transmitter Exp. kit.....

ImageWise Receiver Exp. kit, ...

Both Units purchased together , ...

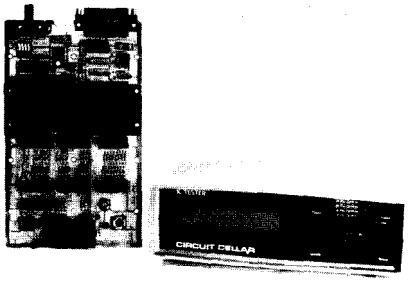
Case & power supply for either unit .....

The Serial EPROM Programmer provides a fast and efficient way of programming, verifying and copying a large variety of EPROM types. Supports 27x16 thru 27x512.

Serial EPROM Programmer complete kit.....

Serial EPROM Programmer Exp. kit.....

Power Supply.....



The IC Tester **has** the ability to identify unmarked **ICsas** well as designate specific pin failures of hundreds of **74xx00** logic chips.

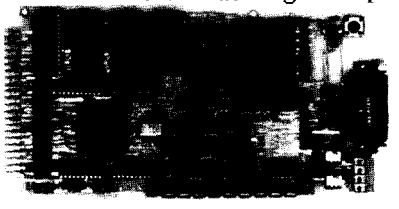
IC Tester Experimenters kit .....

IC Tester complete kit .....

Complete kit with enclosure .....

for **ICT01**.....

- Multi-Tasking Computer



The **BCC180** is a 9 **MHz** single board computer with **384K,** 6 parallel ports, and 3 serial ports **onboard.** Multi-tasking BASIC-180 runs 32 simultaneous tasks.

.....

Circuit Cellar Inc-4 Park St., Suite 12-Vernon, CT 06066 (203) 875-2751

(Sidebar for Home Weather Center - Part 5 by Mark Voorhees on page 16.)

Source code for the WDPS software is available as "shareware". The registration fee, for code and support, is $35.00. This payment will cover source for all modules in the system and all updates or fixes. Software will be released in module groups. You will receive the appropriate disk shortly after it is published.

To order source code on disk, send check or money order to:

Mark Voorhees
P.O. Box 27476
Phoenix, AZ 85061-7476

Allow 30 days for shipment.

Circuit Cellar Books

Circuit Cellar INK authors often refer to previous Circuit Cellar articles. These past articles are available in book form from Circuit Cellar Inc., 4 Park St., Suite 12, Vernon, CT 06066.

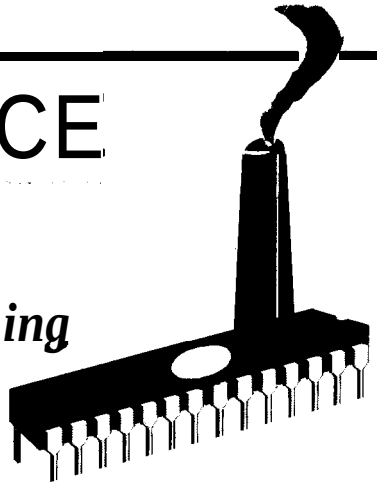
Ciarcia's Circuit Cellar Volume I covers articles in BYTE from September 1977 through November 1978. Volume II covers December 1978 through June 1980. Volume III covers July 1980 through December 1981. Volume IV covers January 1982 through June 1983. Volume V covers July 1983 through December 1984. Volume VI covers January 1985 through June 1986.

FIRMWARE FURNACE

Precision Pulses

Carrier-Current Transmission Timing

by Ed Nisley



Somehow it seems that many firmware problems come down to measuring an input time interval or producing an output pulse. In the last Furnace I demonstrated how an ordinary IBM PC can measure intervals with microsecond resolution, so it's only fair to show how to generate similar pulses.

The sample code this time comes from Ken Davidson's X- 10 PL5 13 power-line transmitter project you saw in Issue 3. Ken explained the whys and wherefores of the X-10 message format, so in this column I can concentrate on the gritty details of getting the timing exactly right. In fact, all of the code you'll see here runs within the 8.33 ms of each power line half-cycle.

Along the way I'll introduce the topic of assembly language macros, so even if you are an old hand at X- 10 code you may find something of interest.

The standard U.S. power line frequency is 60 Hz, so each cycle lasts 16.67 ms. The X-10 message format specifies that each message bit must be sent twice, in true and complement form in successive half-cycles of 8.33 ms each. Further, each of those pulses must be repeated three times in each half-cycle. Figure 1 shows how the bit stream "100" would appear in relation to the power signal. The solid blocks represent "1" pulses which are bursts of 120-kHz signal and the dashed blocks mark the corresponding "0" pulses without a signal.

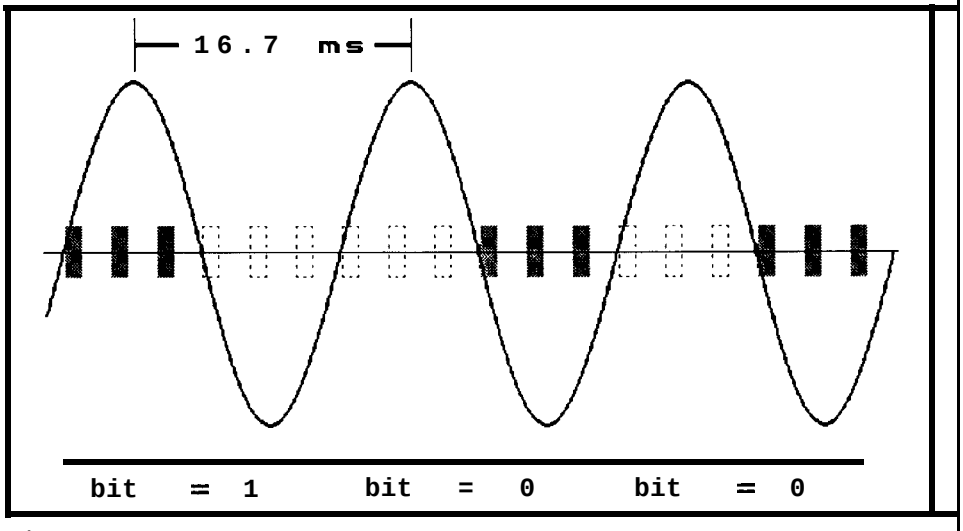


Figure 1 - *This is how the bit stream "100" would appear in relation to the power signal. The solid blocks represent "1" pulses which are bursts of 120-kHz signal and the dashed blocks mark the corresponding "0" pulses without a signal.*

The three repetitions of each bit are timed to occur just after the zero crossings of standard 3-phase

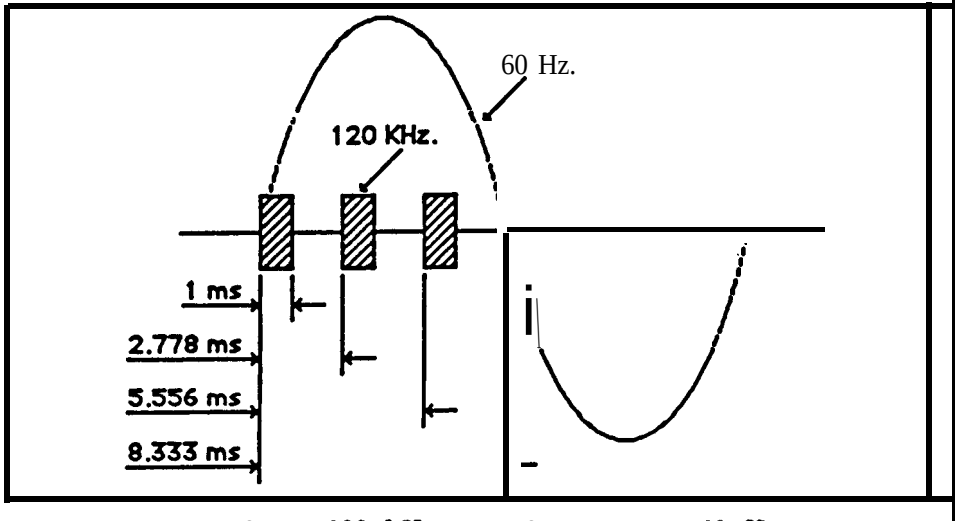


Figure 2 - *Timing of X-IO 120-kHz bursts for a 3-phase. 60-Hz power line.*

power. Industrial installations with 3-phase distribution can thus use X- 10 modules and transmitters without having to match phases. Figure 2 shows the timing requirements for the three pulses. The PL513 module provides a signal that indicates when the zero crossing on one phase occurs; the rest of the timing is entirely up to the program driving the module.

Figure 3 shows the exact timing requirements for the first pulse. As you can see, there are strict limits on the initial delay and pulse width. The prudent programmer sits up and takes notice when descriptions like "100 ms - 50 µs + 100 us" appear in specs ...

Software alone can't handle tight tolerances like that, especially when it must cope with anything from a 4.77-MHz 8088 to a 20-MHz 80386. Once again, though, Timer 2 comes to the rescue with an absolute timing reference. You'll probably want to refer back to the PC's timer schematic in the last column for details of the hardware.

The X- 10 PL513 transmitter has the simplest hardware interface imaginable: one bit in and one bit out. The output is a clipped version of the AC line voltage that provides zero-crossing information. The input signal controls the 120-kHz signal that is sent over the power lines. Figure 3 shows the exact timing relations between the power line voltage, the zero crossing signal, and the 120-kHz pulse.

Remember that the PL513 generates the 120-kHz signal and couples it to the power line. The pulse going to the PL513 simply turns that 120-kHz carrier ON and OFF. The software doesn't have to generate 120-kHz pulses, but it must ensure that the control pulse starts and ends at the right times.

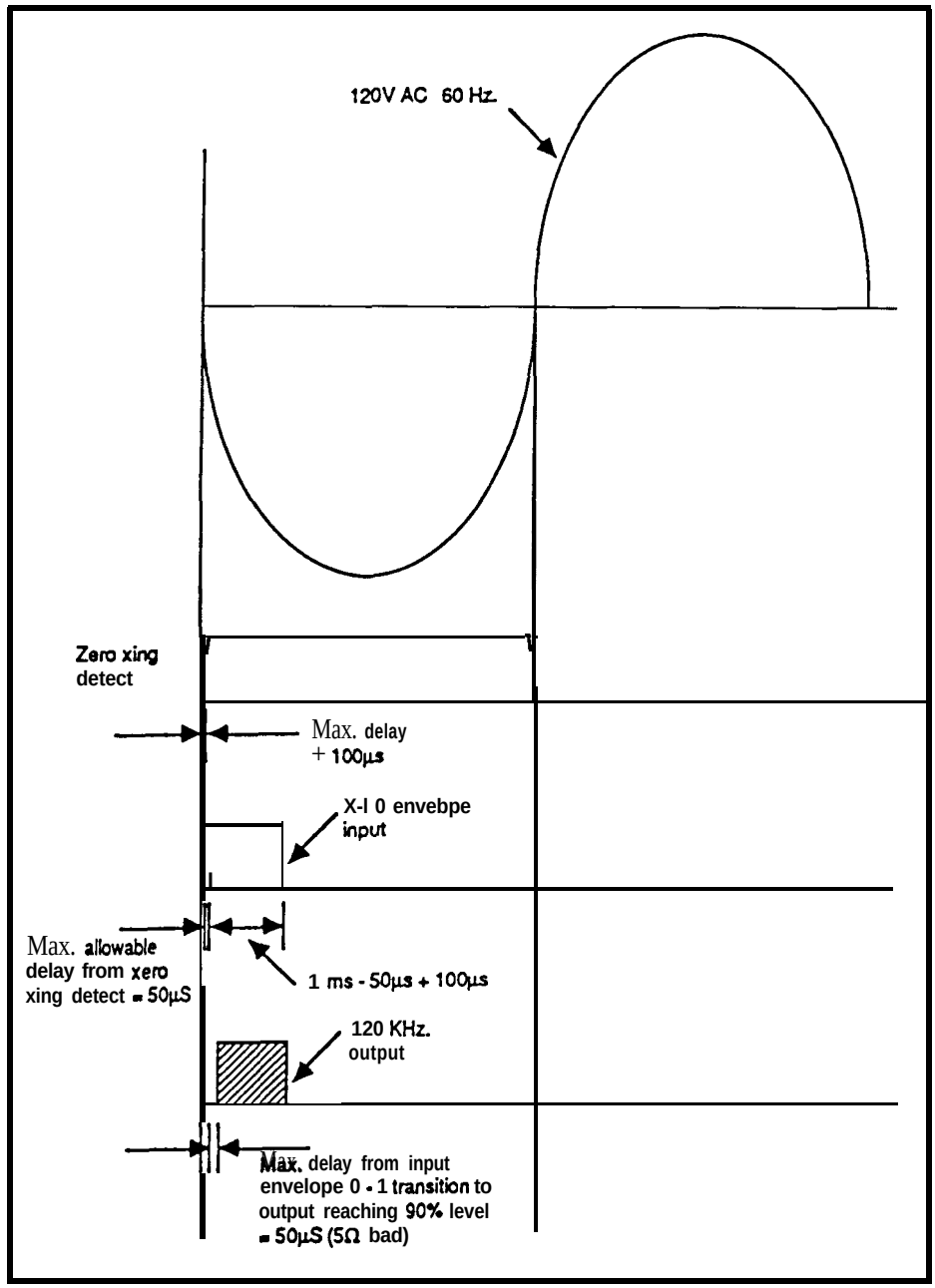


Figure 3 - ***Timing Diagrams - A square wave representing zero crossing detect is provided and is within 100 µs of the zero crossing point of the AC power line. The output signal envelope should be within 50 µs of this zero crossing detect. The signal envelope should be 1 ms - 50 µs + 100 µs.***

Ultimately the IBM PC driver program boils down to calls to the *send3ph* procedure shown in Listing 1. This routine detects the next zero crossing of the input signal and sends three copies of the current message bit. The calling routines are responsible for message formatting, bit complementing, and so forth, while *send3ph* handles the detailed timing for each half cycle.

In order for *send3ph* to do its job the higher level software must call it just before each zero crossing. The third pulse shown in Figure 2 starts 5.556 ms after the zero crossing and ends 1 .O ms later. Because the half-cycle is 8.333 ms long, there is a whole 1.77 ms available for the

Listing 1 - X- 10 bit transmission

```
;----------
; Send three copies of bit at three-phase zero crossings
; Waits for next zero crossing and restarts timer
; Gets bit value from global bitval
; Mashes BL and DX

send3ph    PROC    NEAR
;--- reset timer,  wait for zero crossing, start timer

        CALL    tinit

        MOV     BL,bitval  ; set up bit value for loop

        MOV     DX,ctlport  ; set up current sync input
        IN      AL,DX
        MOV     ctlmem,AL

reget   LABEL   NEAR
        chksync
        JZ      reget       ; loop if no edge

        tstart              ; crossing!  start timer
;--- tell them three times...

        MOV     DX,dataport
        MOV     AL,BL       ; phase A
        OUT     DX,AL       ;   ... bit active
        waitfor Time1000us
        MOV     AL,0        ;   ... and inactive
        OUT     DX,AL

        waitfor Time2778us

        MOV     AL,BL       ; phase B
        OUT     DX,AL
        waitfor Time2778us+Time1000us
        MOV     AL,0
        OUT     DX,AL

        waitfor Time5556us

        MOV     AL,BL       ; phase C
        OUT     DX,AL
        waitfor Time5556us+Time1000us
        MOV     AL,0
        OUT     DX,AL

        RET
send3ph ENDP
```

high-level code to prepare the next bit. This is no problem, even for a 4.77-MHz 8088!

However, getting the first pulse out is a little tougher. According to Figure 3, that pulse must start within 50 microseconds of the zero crossing signal. A 4.77-MHz PC has 210-ns bus cycles, so 50 us allows about 230 cycles. That sounds like a lot until you recall that "simple" 8088 instructions take four or five cycles. Something as complex as a CALL/RET requires 35 cycles!

Ken took one look at my code and decided the most polite way to describe me was as a programmer with "a penchant for macros." Guilty as charged: when you have only 50 microseconds available there simply isn't time for niceties like CALL and RET instructions!

I should mention that figuring out exact timings is quite difficult on processors with instruction prefetch queues. Sometime I'll devote a column or two to measuring instruction timing on 80x86 processors. The results are quite interesting, because they don't match up at all with what you read out of the manual.

The *send3ph* routine starts off with a call to *tinit* to initialize Timer 2. Because the zero crossing is still a few hundred microseconds in the future, the overhead of a CALL/RET pair is no problem here. Listing 2 shows *tinit*'s code. Note that Timer 2 is ready to run as soon as the speaker gate line goes active (remember that Timer 2 normally drives the PC's speaker).

Within *tinit* lies a macro called *punt* that holds a firmware sea story. The IBM manuals for the AT and the PS/2 processors contain this warning: "Back-to-back I/O commands to the same I/O ports will not permit enough recovery time for I/O chips. To ensure enough time, a JMP SHORT $+2 must be inserted between IN/OUT instructions to the same I/O chip."

What this means is that some peripheral chips may need more time between I/O accesses than 80286 and 80386 processors normally provide. Simply put, the second of two successive I/O accesses may (occasionally) not work. IBM elected to pass responsibility along to each PC programmer rather than inserting a hardware delay. Unfortunately, a good deal of old IBM PC code died on new "compatible" IBM ATs because those delays weren't built into the old code. Moral: your firmware won't work when the hardware changes, no matter what they tell you!

The reason IBM picked JMP SHORT to cause the delay is that jumps flush the instruction prefetch queue. Under normal circumstances the CPU fetches instructions beyond the one it is currently executing and places them in

a queue. By executing instructions from that queue, the CPU is isolated from memory access delays. The prefetch mechanism simply reads sequential bytes beyond the instruction pointer, so when a jump changes the IP there's no choice but to flush the queue and start over again. The delay while the CPU reads the next instruction gives the I/O chips enough time to recover, regardless of how fast the CPU runs. In any event, *punt* is the simplest useful macro you'll ever run across and a good introduction to macros in general. To comply with IBM's warning, we must insert these two lines of code between any two successive I/O operations to a single chip:

```
    JMP     SHORT 11
11 LABEL  NEAR
```

Although you could type these in by hand at all the right spots, keeping track of the labels would be a chore. The solution is to "bottle up" those lines in a macro that will assign a new label every time it's used.

The *punt* macro is included in Listing 3. Every macro must begin with a MACRO statement that gives its name and must end with an ENDM (for END MACRO) statement. Any labels defined in the LOCAL statement will not cause multiple-definition errors if the macro is invoked twice. With those three statements out of the way, the two remaining lines are exactly as shown above. The assembler will insert those lines in place of the *punt* macro wherever it appears in the program text.

The key point with a macro is that the exact statements you code between MACRO and ENDM will be duplicated every time the assembler encounters the macro name. Unlike a subroutine, in which a single copy of the executable code is called from many places, a macro produces a copy of the code at each use. A 20-line subroutine that is called 10 times will add only 30 lines (10 CALLs + 20 lines) to the program, but a 20-line macro used 10 times will add 200 lines of code (20 lines * 10 uses)

You must trade off the execution time advantage of duplicate code against the increase in program size whenever you use macros. For example, the code in *tinit* is called from several places and isn't time-critical, so it is best coded as a subroutine. The code in *tstart* is called just as often, but is coded as a macro because its function is more time-critical.

Listing 2 - Timer Initialization

```
;----------
; Sets up Timer2
; Halts timer counting, sets FFFF reload value
; Sets Mode 2
; Clears "speaker enable" bit
; Does not start timer, leaves "gate speaker" bit low

tinit   PROC    NEAR

        CL1                             ; clear control bits
        IN      AL,I8255B               ; get existing port B bits
        AND     AL,NOT MASK spkrgate ; turn off timer gate
        AND     AL,NOT MASK spkrdata ; turn off speaker
        punt
        OUT     I8255B,AL
        STI

        CL1                         ; set Timer 2 to LSB/MSB mode 2
        MOV     AL,10110100B
        OUT     I8253C,AL

        MOV     AL,OFFH             ; set reload value to FFFF
        punt
        OUT     I8253T2,AL
        punt
        OUT     I8253T2,AL
        STI

        RET

tinit   ENDP
```

Once *send3ph* has set up the timer using *tinit*, it loads the current message bit from the global variable *bitval* and waits for the next zero crossing. Because the first message pulse must start within 50 us of the zero crossing, the remaining code will hold no subroutines!

The PL5 13's "zero crossing" signal doesn't actually identify the zero crossings: it is simply a clipped version of the AC line voltage as shown in Figure 3. The signal is high for one half-cycle and low for the next, so the software must detect a zero crossing by identifying the O-to-l and l-to-0 transitions.

The exclusive-or function provides a fast way to compare two bits: the result of an XOR is 1 if the two bits differ and 0 if they are the same. The chksync macro compares the current sync input with the previous value and sets the Zero flag if they differ. The loop containing *chksync* continues until the Zero flag is set, indicating an edge has just occurred.

The next macro is *tstart*, which starts Timer 2 by enabling the SPKRGATE line. The delay between the actual zero crossing and the start of counting will affect all three repetitions of the message bit, so it is worthwhile to figure out whether it is within specification.

A zero crossing just after the IN

Listing 3 - Macros

```
;----------
; Delay required on AT after each I/O operation

punt    MACRO
        LOCAL   11
        JMP     SHORT 11
11      LABEL   NEAR
        ENDM


;----------
; Start Timer 2 by setting gate input high

tstart  MACRO
        CL1
        IN      AL,I8255B  ; get existing port B bits
        OR      AL,MASK spkrgate ; turn on timer gate
        punt
        OUT     I8255B,AL        ; set bits out again
        STI
        ENDM


;----------
; Get current Timer 2 value in AX
; Flips the values so the timer appears
; to count up from 0000 to FFFF

tget    MACRO
        MOV     AL,10000000B     ; latch Timer 2
        punt
        CL1
        OUT     I8253C,AL
        punt
        IN      AL,I8253T2       ; get LSB
        MOV     AH,AL
        punt
        IN      AL,I8253T2       ; get MSB
        STI
        XCHG    AH,AL            ; swap 'em around
        NOT     AX               ; and flip bits
                                 ; to count upward
        ENDM


;----------
; Wait for a specific value to show up in Timer2
; The comparisons are unsigned,
; so we can wait for 64K counts
; Mashes AX

waitfor MACRO   endtime
        LOCAL   reget
reget   LABEL   NEAR
        tget
        CMP     AX,endtime
        JB      reget
        ENDM


;----------
; Test sync input for an edge
; Assumes DX is set up with control port address

chksync MACRO
        IN      AL,DX            ; get current sync bit
        XCHG    ctlmem,AL        ; save for next time
        XOR     AL,ctlmem        ; different from old value?
        AND     AL,MASK syncbit  ; slightly faster than TEST
        ENDM
```

from the port won't be detected until the program goes around the entire loop and enters *chksync* again. In that case there are about 110 cycles before the timer starts, which is well within the specifications even for a 4.77-MHz 8088. A faster processor will reduce the delay even more, of course.

With the timer running, *send3ph* next sends the output bit to the PL513. There are only another dozen cycles or so involved in these instructions, so the first pulse starts within spec.

You've probably noticed that the code uses only half of the (more or less) 230 cycles available and may wonder why I'm making such a big fuss over saving a few cycles here and there. There is a good reason, but allow me to defer it until I've gone through the rest of the code.

After the first pulse is underway, the remaining functions are quite simple. All that's needed is to watch Timer 2 until the appropriate count occurs and turn the pulse output on or off. The *waitfor* macro does exactly what its name suggests: it waits for a specific count in Timer 2.

Unlike the macros you have seen so far, *waitfor* can produce different code each time it is used. The MACRO line specifies an argument called *endtime* that is the value of Timer 2 that ends the wait interval. The first line of *waitfor* is another macro, *tget*, which reads the current value of Timer 2 into AX. The comparison with *endtime* must be unsigned because the counts can range between 0 and 65536.

The *waitfor* macro is invoked five times, each time with a different argument. The values of those arguments are defined in Listing 4, which shows all of the program

constants. The times are in units of 838 ns, which is the clock rate for Timer 2. The reason for defining these constants with EQUs instead of hard-coding the hex values is that it is much easier to figure out what

waitfor Time2778us

means than

waitfor 00cf3h

when you're poring over the listings.

Listing 4 - Constant Definitions

```
;----------
; Galactic constants

I8255B      EQU     00611i    ; 8255 port B address
I8253C      EQU     0043H     ; 8253 command register
I8253T2     EQU     0042H     ; 8253 Timer 2 register

;----------
; Delay times measured in Timer 2 ticks
; These are found by computing (time/54.9ms)*64*1024

Time52ms    EQU     0f25dH    ; 52 ms count value in Timer 2
Time25ms    EQU     07485H    ; 25 ms

Time1000us EQU      004a9H    ; 1.000 ms -- bit length

Time2778us EQU      00cf3H    ; 2.778 ms -- phase B start
Time5556us EQU      019e5H    ; 5.556 ms -- phase C start
```

After the third pulse is completed, *send3ph* is done and simply returns to the caller. If the X-10 message isn't finished, *send3ph* will be called again within 1.7 ms to send a new trio of pulses.

Unfortunately, there's a complication I haven't mentioned. An interrupt occurring while the code is waiting for a zero crossing or a specific timer count can introduce an unpredictable delay. If the interrupt lasts long enough it can garble the X- 10 message enough that the receiver modules won't understand it. With timer interrupts occurring 18 times a second, some collisions are inevitable and some problems are sure to arise.

Because X- 10 messages can last for hundreds of milliseconds, it's not feasible to disable interrupts for an entire message. The IBM PC interrupt system doesn't save interrupts, so a missed interrupt is gone for good. For example, if you disable interrupts for 200 ms you will lose three timer ticks and cause the BIOS time-of-day clock to slip out of sync with reality.

Because the pulse timings are so critical, we could disable interrupts at the start of each half cycle and enable them after the end of the third pulse. Unfortunately, that would force all the interrupts into the 1.7 ms before the next zero crossing and ensure that any delays will clobber the first pulse. Because most applications use single-phase power lines, that's the only pulse that really matters!

The best solution is charming in its simplicity, and it's the same one IBM used with the AT: do nothing. In actual practice the timer interrupts are quite short, so they will affect at most one pulse. If the interrupt occurs after the pulse starts and completes before it ends, that pulse will be completely unaffected. Obviously, reducing the number of cycles required for the time-critical code increases the number of cycles available for an interrupting routine.

You can get a rough estimate of the probability that a timer or keyboard interrupt will cause a problem using simple arithmetic. A quick check of the AT BIOS listing shows that the timer interrupt has about 30 instructions in the worst case. Figuring 3 microseconds for each instruction (a 4.77-MHz PC does about 1/3 MIPS downhill on a good day) means that the timer interrupt handler requires about 100 microseconds.

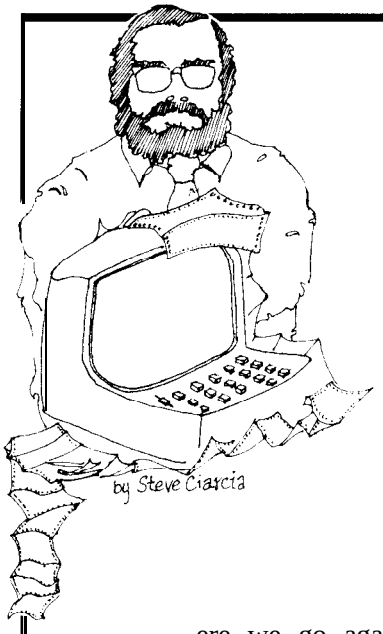
In each full power line cycle there are six pulse edges in one half-cycle that must not be delayed and another half-cycle with no pulses at all, so there are 600 us out of 16.7 ms where an interrupt may cause problems. Divide those two numbers and you'll see that a problem may occur 3.6% of the time. But a single-phase AC installation requires only the first pulse, so the probability of hitting either of those two edges is only 1.2%.

A more accurate analysis would take into account the facts that there can be only one timer interrupt every 3.3 full cycles, that pulse edges can be delayed slightly with no ill effects, and that the typical length of the interrupt routine is less than 100 us. The probability of a problem is surely well under 1%.

In fact, the X-10 message format was designed to cope with such interference when it does happen. Each message must be sent twice, so if either copy is garbled the other will probably get through. Intelligent home control systems (like Steve's HCS) send "refresh" messages every few minutes to ensure that the receivers are in the right state. The overall reliability can be quite high despite an occasional glitch.

While I don't advocate ignoring problems, part of the challenge of writing good firmware is knowing what to fix. One of my professors back at Lehigh summed it up as "Engineering is knowing what to ignore."

And on that note, it's back to the bench!



Back Here

ere we go again. Shoved to the back of the magazine. You'd think that starting your own publication could keep you from being displaced as top dog!

Only kidding. This time the choice was mine. Back here I don't have to be eloquent or discuss the politics or virtues of different technologies. I don't have to explain the direction of the magazine or the choice of articles. Curt Franklin now has that task and will definitely do a good job at it. Bringing a broader perspective of the industry than I care to have, he will be less biased and help INK become revered as a broad-based technical resource; something that other magazines are moving away from.

Up in the front, one has to maintain a persona of industry wizard and technical guru. After all, how can all these thousands of readers justify subscriptions on anything less than a religious experience in print?

Back here I can be myself. I can tell you that the real Steve Ciarcia is a guy with a sense of humor; a guy with technical expertise, but just as surprised as anyone else to find that he can write intelligently; and, above all, a guy who still enjoys the process of relating the hands-on experience of designing and building electronic things.

Up front, of course, deliveries have to be structured, and there has to be a purpose for everything: an introduction that appeals to the largest possible audience; a project with merit; and a description done with reverence and purpose. Back here I can be thought provoking, irreverent, or as humorous as the situation dictates.

Being a technical resource publication with a personal imprint need not be mutually exclusive. I recognize my lack of interest in certain areas and I've gathered the right people to provide that direction where needed. At the same time, they are being thoroughly indoctrinated in the Circuit Cellar ethic: serious technology brought to you by people who enjoy using it. Together, the spirit of INK continues.

Back here I'll enjoy it even more.