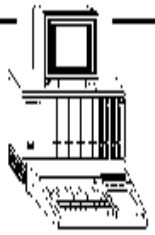
YESTERDAYS NEWS

VOLUME 1 NUMBER 3 Established 2016 JUNE 2016

IS THE A

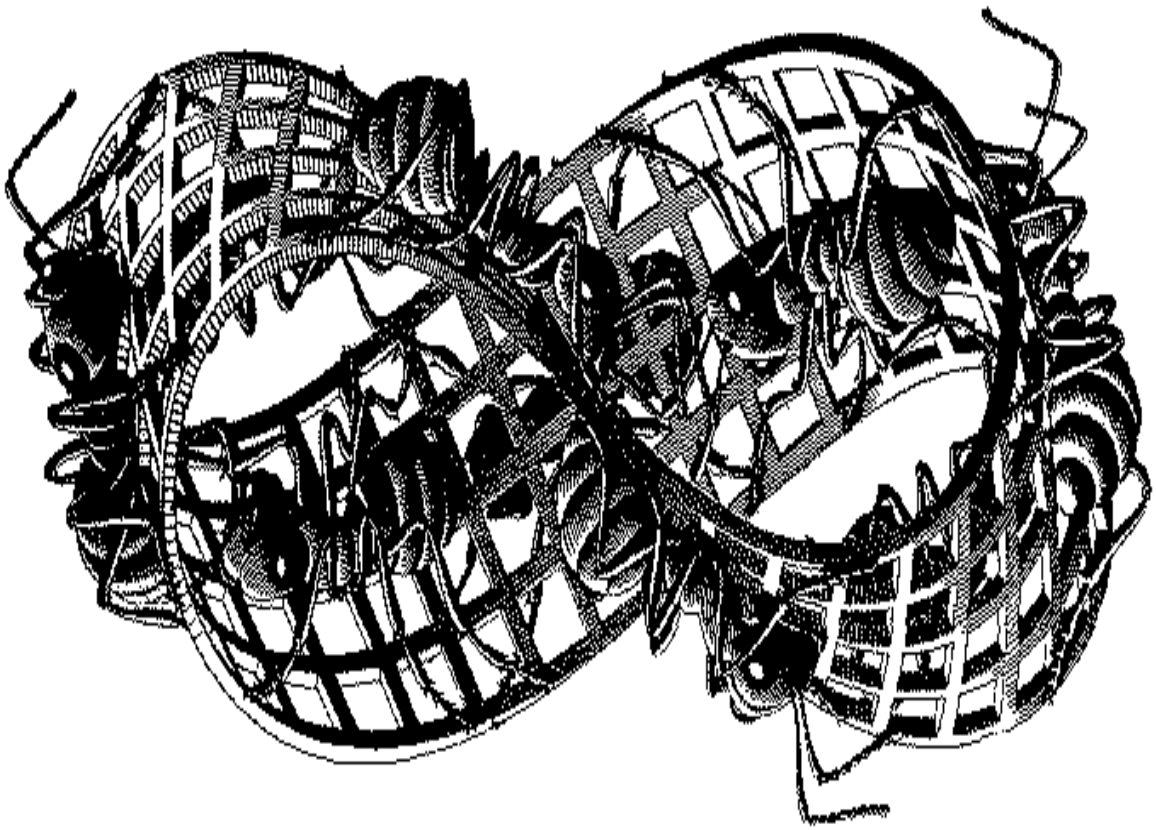


DINOSAUR?

YES, THAT'S WHY WE DIG THEM

VAST TI99ERS

INSIDE INFORMATION



keep everlasTIngly at it!

THE FACTS ABOUT TI MEMORY SYSTEMS

An editorial by Chris Bobbitt *(7/93 A9CUG NEWSLETTER)*

Over the last six months there has been a lot of noise on the computer networKs and in user group newsletters on issues related to extended memory cards for the 99/4A. Some people have blatantly asKed people to come out and choose sides on a very complex issue without understanding what they are choosing. In fact, what should be a pretty objective decision has been turned into an emotional gut-churner - a question decided by loyalties, petty rivalries, lies and innuendo. FranKly, this is why we are in the situation we are in today - and why companies liKe Myarc and Corcomp left the community. Everytime a technical debate gets turned into a personal vendetta, thinly disguised ambition is allowed to prevail over substance, and the community eats its young yet again.

When I set out almost 3 years ago with a really talented bunch of guys to put together a new Kind of memory card for the 99/4A, I had believed that the community had finally outgrown that Kind of thing. I was wrong. Seeing all of this stuff all over again has made me seriously consider throwing in the towel once and for all.

Why? Because everything we've done with these cards has either been ignored, mis-represented, or labeled as "too controversial" or "not ready for prime time".

For 8 months we've been mailing out press releases, articles and newsletters about our memory cards that apparently no one is reading, and user groups aren't re-printing or even reporting on. The only reference to what we've done that I ever see in print is usually in an article about our competitors, or in an editorial that simply says that there has been a debate and that it has gotten out of hand.

This is simply ridiculous.

This is the most important thing I've been involved with in the 10 years I've been in this community, and unless the community gives this a fair hearing, well, I guess it's finally time to cut my losses.

Here is my last attempt to get the unvarnished facts out in front of you, the reader.

1. The Asgard Memory System (AMS) is available NOW - it is NOT still in 'develop ment'. We announced the product the day it was commercially available for sale, and in stocK. In the last 8 months, we ve been refining the product, writing software and worKing on the next generation card. Our only competitor announced their product over a year ago, and have yet to release more than press notices (which all seem to be faithfully reprinted everywhere). It is pretty hard to compete with something that so far exists only on paper - especially when the unreleased product gets more press than the one that you can buy today!

2. We started AMS almost 3 years ago - long before there ever was a 'National Committee for TI Standards'. This so-called committee has never met more than once , doesn't include most of the TI hardware or software developers in the U.S., much less the rest of the world, and has produced a specification for memory systems without any real debate, which endorses our competitor's plans. Before we had a chance to object, it was the declared 'standard'. Can you say railroaded?

3. Our memory system was designed to the only standard TI ever made for extended memory on the 99/4A - the one used in the TI-99/8. In fact, the guy who DESIGNED the TI-99/8 said our design was identical to the one TI specified.

4. Because our design was built to TI's specifications, it doesn't conflict with any other card in the P-BOX - except a 32K card. You can plug it in and your Horizon RAM-DISK, Myarc HFDC, or anything else you have will still worK fine.

5. Our design uses standard, off-the-shelf components. EVERY other extended memory design uses lots of custom ICs, and even more custom programming (as in a big DSR). Custom parts not only drive up the development time, they also drive up the cost, and guarantee that the design remains proprietary. By using off-the-shelf parts, we Keep the price down, and guarantee competition. Remember how much TI used to charge for the 32K card when they were the only one maKing them?

We designed our system to the KISS method - Keep It Simple, Sam.

6. Our system is tried and tested. We use the exact same memory mapper (the chip that controls the computers use of memory) that TI used in their 99/8, their 9900 minicomputers, and that IBM used in the very first IBM PCs. This component has been available for 10 years - all bugs in it have long been removed.

7. Everything about our system is open. Anyone can write a program for it or enhance it - the hardware and software specifications are available free of charge. HecK, the 5-disK development system we've spent the last 18 months writing is even fairware - and posted on the bulletin boards.

see page 3

SLOOOW DOWN

BY MACK MCCORMICK
FROM MICROPENDIUM VOL 3, #11

DISCLAIMER: IF YOU ATTEMPT THIS PROJECT AND THE SMOKE LEAKS OUT OF YOUR COMPUTER I AM NOT RESPONSIBLE.

I can hear everyone now! Why in the world would anyone want to slow down his computer?

I couldn't agree more and therefore usually spend a substantial amount of time trying to speed things up. In fact, that was one of my main reasons for learning Assembly Language.

And so begins our first in a series of hardware articles. I assure you they will get much more involved after this one, but this is intended to ease you in to reading circuit diagrams and understanding basic electronics. The circuit described here slows down everything the computer does as long as interrupts are enabled by the running program (such as in Basic or Extended Basic).

Using this circuit, you may watch your title or program screens being built in slow motion. The beep as the computer powers up can be excruciatingly long. All this is controlled by a simple circuit in which the time delay can be varied or even frozen with the press of a button. Examine Figure 1, and I'll briefly describe what is going on in our one chip circuit.

How it works

This is your basic NE555 timer chip which is set up to periodically provide a zero logic pulse to the external interrupt pin of the TMS9900 microprocessor (CPU) chip of the TI-99/4A. This effectively suspends processing as long as it is at logic zero. These periodic interrupts serve to really slow down the executing program, in effect providing slow effect.

All of this is made possible with a mere eight parts. Capacitor C1 simply filters the incoming +5 Voltpower to reduce any unwanted signals from the rest of the computer. R1 establishes the minimum delay time and in conjunction with VR1 and 2 comprises a R C timing circuit to determine the delay prior to the 555 flip-flopping. Capacitor C3 provides a reference voltage for the control gate. Switch S1 places the reset pin low when pressed and causes the output to go low thereby generating an interrupt. Switch S2 connects the timer to the interrupt pin of the computer. The timer circuit operates whether or not it is connected.

Building the circuit

I built the entire circuit on a piece of perforated circuit board about an inch square and housed it in my speech synthesizer. It tooK less than an hour.

There are only three wires to connect to your speech synthesizer, plus three holes to drill for mounting the two switches and the potentiometer. TaKe your time when connecting the wires and remember that the odd numbers are on the underside and the even numbers are on the top.

After everything is assembled ensure you have made all the connections properly and have no solder bridges (short circuits). If you use a 7555 CMOS IC instead of the NE555 be especially cautious to avoid static damage to the chip while handling. Ensure you are well-grounded when installing the chip.The parts used in this circuit project should be available from your local Radio ShacK or ele ctronic supply store.

Testing

Load your favorite Basic program with graphics and turn on the slow motion circuit. Immediately you should see the cursor slow down. You can vary the spee d with the potentiometer and freeze it with the push button. RUN the program and you should be able to see your graphics built in slow motion. If it doesn't worK checK your wiring and, with a voltmeter, checK your supply voltages at pin 8 for +5 Volts. If all else fails, there's usually at least one member of a local user group who is into electronics who maybe able to offer help.

I've purposely not gone too much into detail on how the circuit worKs to Keep things simple for this first time out. Also, because the circuit is so simple, I have not described in great detail how to connect the wires together. Remember, only maKe a connection where there is a dot (square) when the wires cross one another.I would appreciate brief note from those of you who would liKe more hardware articles and let me Know if you had difficulty understanding this article. Until next time, don't sniff too many solder fumes.

Parts List

IC NE555 or 7555 chip

R1 110 ohm, 1/4 Watt resistor

VR1 100 Kohm potentiometer

C1 15 uf 10 Volt electrolytic capacitor

C2 3.9 uf 10 Volt electrolytic capacitor

C3 0.1 uf 10 Volt capacitor

S1 Single pole momentary contact mini-switch

S2 Single pole single throw mini-switch

SEE "CIRCUIT", PAGE 4

8. The AMS is very fast. It can switch pages over 10 times faster than any competitor, and with little program code (even in Assembly). Why is speed important? If you are sorting 512K of data, or loading 512K of pictures, you'll notice the speed - in fact , you'll notice the other system is less than half the speed.

9. Our system doesn't have its software in a DSR - and we are proud of it! Why?

A. We found that putting the operating software in a DSR makes it run much slower than if it was in RAM - and really doesn't give any benefit to the programmer or the user.

B. Any DSR increases the chance for compatibility problems - who wants to waste time finding problems with Myarc cards?

C. A DSR is fixed. If you find a bug in it, the only way to correct it is to replace it. Consider all the pain Myarc users have gone through with EPROM upgrades of the HFDC and the Geneve.

D. If programs are written to work around a DSR bug, they may not work when the DSR is fixed.

E. If the software to use the card is built into each program, than the only thing we have to do to correct a bug is issue an upgrade. Old programs written for earlier versions of our operating system software would continue to work fine, and new programs could take advantage of new features without worrying about hardware compatibility problems - since the operating system isn't in hardware.

F. Why do you think Microsoft and Apple load their operating systems from disk, and not from ROM chips?

10. We have a complete set of development tools available NOW. Even if our competitors released their card today, it would be a year before they had a system that was as easy as ours is for programmers. Because our software was designed before our hardware, we were able to design a programmer friendly system that is far easier to program than any other extended memory system. This is important - as so many people have said, who wants a memory card there are no programs for?

In the last 8 months since we released our first AMS card we've released 2 software packages that take advantage of the card (including the word processor FIRST DRAFT), and software from other people has started to appear. Around 20 AMS cards are in the hands of developers around the world.

Is any of this news? Apparently not - I've seen few of the facts above in print anywhere, even though we've put them in a half-dozen articles.

The facts, on their own merit, should be compelling enough for people to put aside their differences and really weigh the benefits of what we've done - instead of consigning it as some curiosity, or ignoring it.

We wanted to put together something that was cheap enough to build that every TI user could have one, and yet was simple enough to write programs for that every Ti programmer could do so. I think we've done that. If the TI world isn't interested at this point, doesn't care, or wants to Keep waiting for fantasies, well, I can take a hint.
Thank You.
Chris Bobbitt - July 2,1993

TYPE IN PROGRAM

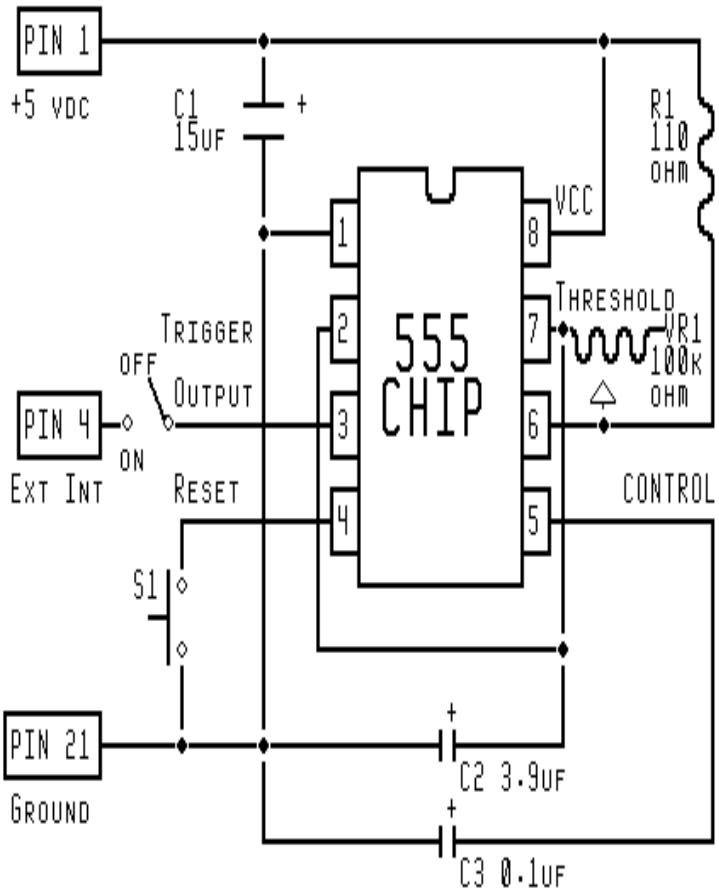
```
100 REM ** FASCINATION **
110 REM
120 REM by Jan Barnier
130 REM
140 RANDOMIZE
150 CALL SCREEN(2)
160 CALL HCHAR(1,1,31,768)
170 FOR A=1 TO 12
180 READ A$
190 CALL COLOR(A,INT(RND*14+
3),2)
200 FOR B=A TO 25-A
210 CALL CHAR(33+(A-1)*8,A$)
220 CALL HCHAR(B,4+A,33+(A-1
)*8,24-C)
230 NEXT B
240 C=C+2
250 NEXT A
260 RESTORE
270 FOR D=1 TO 12
280 FOR E=4 TO 16
290 F=INT(RND*14)+3
300 IF F=G THEN 290
310 FOR H=12 TO 1 STEP -1
320 CALL COLOR(H,INT(RND*14)
+3,F)
330 NEXT H
340 G=INT(RND*14)+3
350 IF G=F THEN 340
360 FOR I=1 TO 12
370 CALL COLOR(I,1,2)
380 NEXT I
390 CALL CHAR(E*8+1,B$)
400 NEXT E
410 READ B$
420 NEXT D
430 GOTO 260
440 DATA 1247234792387598
450 DATA 7759879577987878
460 DATA 9873982347928398
470 DATA 0340238398739879
480 DATA 2301384017359874
490 DATA 9134049723578598
500 DATA 23B1014013858898
510 DATA 3091023820375984
520 DATA 2010390138402738
530 DATA 9239183643827576
540 DATA 2013013498374587
550 DATA 2130103910579313
560 END
```

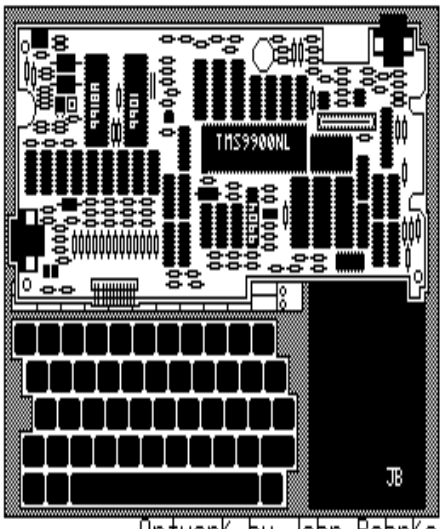
TYPE IN PROGRAM

Repairs unlistable programs. Type UNFIX in & save as a MERGE file, then load the offending program & merge UNFIX on top and type RUN.

```
1 !UNFIX-SAVE AS MERGE FILE
2 CALL INIT :: CALL PEEK(-31
952,A,B,C,D):: SL=C*256+D-65
539 :: EL=A*256+B-65536 :: F
OR X=SL TO EL STEP -4 :: CAL
L PEEK(X,E,F,G,H):: ADD=G*25
6+H-65536 :: PRINT "LINE #";
E*256+F
3 I=1 :: CALL PEEK(ADD-1,V):
: IF V THEN 7
4 CALL PEEK(ADD+I,V,W):: IF
V THEN I=I+1 :: GOTO 4
5 FOR Y=SL TO EL STEP -4 ::
CALL PEEK(Y,E,E,E,F):: IF E*
256+F-65536=ADD+I+2 OR W=0 O
R ADD+I>-3 THEN CALL LOAD(AD
D-1,I+1):: GOTO 7
6 NEXT Y :: I=I+1 :: GOTO 4
7 NEXT X :: STOP :: !@P-
```

"CIRCUIT" FROM PAGE 2





Artwork by John Behnke

THE NATIONAL FUN & GAMES

GAME TITLE	SCORE	JOYSTICK JOCKEY	TI CLUB	DATE
BACKSTEINE	155900	STEVEN JAKABFY	OSHTI UG	09/95
BIGFOOT	290500	DAVID HANDLE	OZARK 99	01/95
BLASTO	44880	MIKE CENDROWSKI	W/PENN 99	11/94
BREAKTHROUGH	1850	RAY FRANTZ	VAST	11/93
BURGER BUILDR	1000000	ELEANOR ZIC	W/PENN 99	03/94
BURGERTIME	82600	MICKEY CENDROWSKI	W/PENN 99	09/85
CAR WARS	6050	JIM WAYNE	VAST	11/93
CENTIPEDE	301930	MICKEY CENDROWSKI	W/PENN 99	01/87
COLORS	1000000	HARRY HOFFMAN	CLEVELAND	03/95
DIG DUG	262460	FRANK ZIC	W/PENN 99	03/94
ENTRAPMENT	3668	FRANK ZIC	W/PENN 99	11/93
HOPPER	4031826	TOM BEERSMAN	OZARK 99	06/94
HUSTLE	WON 52	ELEANOR ZIC	W/PENN 99	03/94
JAWBREAKER	15025	JIM WAYNE	VAST	11/93
JUMPY	131900	ELEANOR ZIC	W/PENN 99	03/94
MICRO PINBALL	1776500	NORM ROKKE	W/PENN 99	05/87
MIDNITE MASON	27100	FRANK ZIC	W/PENN 99	11/93
MINEFIELD (A)	0:00:01	NORM ROKKE	W/PENN 99	08/94
MINEFIELD (B)	0:00:05	NORM ROKKE	W/PENN 99	08/94
MINEFIELD (C)	0:00:12	NORM ROKKE	W/PENN 99	08/94
MINEFIELD (D)	0:00:31	NORM ROKKE	W/PENN 99	08/94
MINEFIELD (E)	0:00:47	NORM ROKKE	W/PENN 99	08/94
MINEFIELD (F)	0:01:27	NORM ROKKE	W/PENN 99	08/94
MINEFIELD (G)	0:02:26	NORM ROKKE	W/PENN 99	08/94
MINEFIELD (H)	0:02:36	NORM ROKKE	W/PENN 99	08/94
MINEFIELD (I)	0:03:56	NORM ROKKE	W/PENN 99	08/94
MINEFIELD (J)	0:04:27	NORM ROKKE	W/PENN 99	08/94
MOON PATROL	73150	MIKE SEALY	W/PENN 99	03/94
MUNCHMAN	202170	PAUL BROCK SR.	W/PENN 99	09/87
PACMAN	153000	GARY TAYLOR	W/PENN 99	09/87
PARSEC	47300	MICKEY CENDROWSKI	W/PENN 99	09/87
PKR SOLITAIRE	3790	JACKIE REMENSKI	VAST	11/93
POLE POSITION	57700	MICKEY CENDROWSKI	W/PENN 99	12/94
SUPER YAHTZEE	615	JACKIE REES	VAST	11/93
THE ATTACK	31800	JIM WAYNE	VAST	11/93
TI INVADERS	15930	PAUL BROCK SR.	W/PENN 99	09/87
TI TRIS	2208	FRANK ZIC	W/PENN 99	11/93
TOMBSTNE CITY	154400	DANNY MCGUIRE	OZARK 99	11/94
TRN SOLITAIRE	351	CAROL HOFFMAN	CLEVELAND	03/95
TREASURE ISLE	37800	MIKE CENDROWSKI	W/PENN 99	10/94
TRIS (ASGARD)	8393	MICKEY CENDROWSKI	W/PENN 99	12/94

IT'S WHATS INSIDE THAT COUNTS

GAME ON