

TI - D - BITS

Issue 3 Volume 14

PHILADELPHIA AREA TI-99/4A USER GROUP

May/June 1993

THE OTHER GUYS vs TI-99/4A

By Richard Lynn Gilbertson
Taken from the
Greater Akron 99er's Newsletter

Little do most of us appreciate what we have in this TI-99/4A, so it made sense to write about it. First off, what do these OTHER GUYS have, well more memory and a faster processor. They also have fantastically huge program packages to do their work. They have hundreds of companies with support lines and so many different versions of the same machines and software too, that it boggles the mind to count them. The OTHER GUYS have years of major research invested in every software package and so many different operating systems for disk and processing data that it would take several books just to list the names.

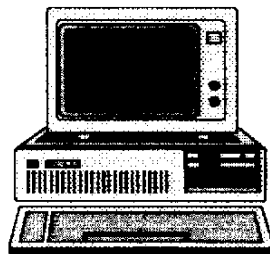
OK, what does the TI-99/4A have to compete with that? Well, hold onto your hat, the 99 does have several things and none of them are by any means minor.

First off, lets go into a little history. When one of the OTHER GUYS stores data onto a disk like say, "STORE THIS", what you will find on the disk is "STORETHIS cr lf". Now the "cr,lf" means, it's universal among the OTHER GUYS, is Carriage Return and Line Feed. So every line of data on the disk has to have this on the end: "cr,lf" to tell the computer that this is the end of the string of data. Now, the 99 has a simpler approach, "OASTORE THIS" is how the 99 does

the same thing. The "OA" stands for 10 in hexadecimal and you can see being at the front instead of at the rear of the string means that you don't have to read the whole line to see how long it is. The OTHER GUYS have to load the whole line and make the computer count how long it is. If you are searching a disk, you can see why it takes so long for the OTHER GUY.

stop The history of why that is, comes from the concept the OTHER GUYS system uses, which was conceived in 1953. The 99 uses a concept from 1975. And yes, the OTHER GUYS have not changed because, if they did, all the software written would have to be totally rewritten. Also forget them doing it any time soon. I should also mention that this system of ours is already being used on newer main frames.

Today I was asked if the 99 was compatible with the OTHER GUY and, as usual, was quite insulted. Let me show you why. Go ahead and ask an OTHER GUY when was the last time he got out his soldering iron and added something unique to his system. First off, he will look at you like you are really crazy. Then he will ask, "What do you mean, unique?!"



Say "Unique, like an interrupt switch to halt everything and do something else previously loaded. Or just stop what you are doing and do nothing. Or do a total reset and start over." He will reply, "Oh yea, I can do that last one." And he is right, he can only do that last one. His software has to do the others, the 99 is already built for those and doesn't care if

hardware or software causes it to happen. Or, ask the OTHER GUY to load and run his very best Telecommunications, Word Processor, Disk Manager, and Assembly Compiler from one disk without changing disks.

Honestly he will say "You can't get all that on one disk!", reply "Really, I can do it on mine. And I still have room for almost a third more."

[Even though the OTHER GUYS disk will hold more.]

How about operating systems. The OTHER GUY, most of them anyway, has to load one before he can do anything. First off, what are the advantages of this, one is for a particular application this will allow you to have the fastest program loaded possible, and the second is it will be as small as possible for the application. Disadvantages? History should be explained now as the advantages are distorted. Back in 1950, the concept of loading a new Operating System to speed things up was the best solution to lack of memory. Also, it had the other advantages mentioned and it made sure that the system was running at peak ability. But that is also where all the trouble started. You see, if you want to run something else, you'll load it someplace that is already being used. This means that you have to load a program that moves it out of the way, so you can load something else, then you can load this other program.

Does it require much thought to see that it is similar to digging a hole and filling it with dirt from another hole so that you can re-dig the original hole? That is without even mentioning the fact that the OTHER GUY may, on some systems, have as many as three types of Operating Systems, and they are not compatible with each other even though they are all written for the same computer.

So the 99 doesn't have the problem of crashing while loading the third program because it is too

large, it knows that the second program used up all the space. The 99 uses what is called Relocatable Code, which means that it just loads where there is a space left, and it knows how much memory is left. This is Artificial Intelligence. The 99 does not have to ruin everything in memory to see that there is a problem, The OTHER GUY has to reboot from scratch.

Expansion of the system over time on the OTHER GUY also requires a history lesson. For lack of a better term, we will say that the OTHER GUY has a 'HARD CODED' system. I got that term from a Computer Science Professor as he coined it on the air. Hard Coded means that you can't run any programs from the area dedicated to the disks or the RS232 or other peripherals. You can't load anything there even move anything there without special hardware and software. The 99 only has one Hard Coded area, the operating system. The OTHER GUY has several and can't move them to another location, remember they are Hard Coded (physically located by hardware).

Let us say that we want to add one Meg of memory to a fully expanded system the OTHER GUY has. Sorry, no can do, Hard Coded. For the 99, that is no problem as it can, without any modification at all, talk to 128 devices, so just make it a device and add it on. An example is the 192K Video Display Processor (9958) added to the 99. It can be upgraded and increased in size also. And the GRAM/GROM port for cartridges can talk to 640K of GRAM/GROM, but, turned into a device, could be expanded into an unbelievable 4095 Meg of 4 Gigabytes. This is without even mentioning the RAVE Card which is an OTHER GUY approach to expansion, but that is only 3.5 Meg maximum for the 99

Five years from now, I'll still have my TI-99/4A and will be adding devices and cards to it. The OTHER GUY will have gone through two

systems by then just to stay current with expansions. I mentioned only one thing the Professor said, he also said that Mini and Micro computers are similar to 'shoot and throw away' cameras.

The Tamira, 990/10, TI-99.4A, and NXT are the only Memory to Memory transfer, Memory Mapped Memory, and Memory to Memory Architecture computers ever built. So I don't think he knew about the TI-99/4A or it's relatives. All the rest are like the 'Hard Coded' machines he mentioned. Why do you suppose that they have to replace the mother board for the simplest upgrade? 1950's concepts in the 1990's, and does the OTHER GUY really have that much on the TI-99/4A, we shall see, we shall see.

SPEECH and RAMdisks

Taken from MICROpendium

This item, by Col Christensen, appeared in Bug Bytes, the newsletter of the TI99/4A Brisbane (Australia) User Group.

I have reported previously that opening a file called "SPEECH" in BASIC with the TEII module installed doesn't work if a RAMdisk with powerup on is also in the system. I've finally come across the answer in the documents for the Explorer program. It says that only a few modules have power-up routines in them, one being the TEII module. It seems that the power-up routine in TEII must initialize the module for, apart from other things, the text to speech utility, or maybe just the presence of the devicename "SPEECH."

What happens is, when the computer is switched on and even before the screen display is turned on, it looks at the peripheral cards for powder-up routines prior to scanning for module power-up. With a RAMdisk installed, the

RAMdisk power-up routine takes over (to hang with any other cards or modules), does whatever setting up is required and proceeds to load its own menu onto the screen. Poor old TEII doesn't get a look in. So that's the reason.

The cure is to turn the RAMdisk power-up off.

**NEW REPAIR SERVICE
NOW AVAILABLE FOR
MYRAC PRODUCTS**

Don Walden of Cecure Electronics is now the official repair center for Myarc products. Don says Cecure finalized the deal with Myarc just before the Chicago TI fair. Cecure can be reached at P.O. Box 132, Muskego, WI 53150. His Phone number is (419) 679-4343.



~~~~~ TI-101 ~~~~~

## OUR 4/A UNIVERSITY

by Jack Sughrue  
Box 459  
E. Douglas MA 01516  
#2 Holism

Happy New Century

Last time, Class, in our TI-101 classroom we introduced the historical perspective of public education in a few strong words. We stated that some of the wrongs with our schools today is the profiteering by the big book industry who would like all our children to be into some kind of large-scale, lock-stepping curriculum as devised by them. [Close to 100% of all the schools in America have curriculums established by publishers and screwed into place by administrative bureaucrats. They are not created by the teachers, the trained professionals who work directly with the children. Once in a while - such as the school in which a friend of mine teaches 3rd graders - a school is blessed with an intelligent, child-oriented principal who is not afraid to empower her teachers. But this scenario is truly rare in our country.]

Which brings me back to THE REVOLUTION in education I discussed during our last class. This is the revolution of holism in education. It is an international grass roots approach to learning. Though the spelling is H O L I S M, the meaning does not come from "holy" but from "whole." Why don't THEY spell it "wholism?" you may ask. Unfortunately, you may ask in vain.

But Whole Language is what is under discussion here. Whole Language is the most prominent movement in the revolution. It is a philosophy that asks how children learn and then seeks ways to provide those opportunities for the child. It is, in short, a research-based philosophy and an intellectual attitude and a creative style that considers the developmental learning stages of the children as a group and each child individually. But what is it, specifically?

Well, let's look at product results first, Class, although Whole Language Educators will be the first to say that process rather than product is the goal of W.L.:

In the standard achievement tests scores given world-wide the U.S. ranks 47th. On those same tests New Zealand is 1st. New Zealand has close to 100% of its teachers, K-12, using W.L. New Zealand has the highest rate of literacy of any country in the English-speaking world.

Now back to how W.L. works and what it is. In the U.S. we have had a long history of process methodology. Unfortunately, it has never been a part of mainstream education. Like jazz, as musically intricate as any form of music on the planet, has never become the mainstream of American culture. But there were many educators who understood how children think and how children learn. These people have taught and have written books and have done research. But, except for the unusual teacher or an extremely rare school staff, few people had access to these ideas and materials and methodologies. Such things as the Teacher-Writer Collaborative in New York, the Bay Area Project in California, and the Framingham Writing Project in Massachusetts spread the word through research, printed materials, workshops, teacher

training programs, sweat, blood, and tears. But these were a few of the isolated programs and projects and groups that sought to integrate the curriculum by starting at Square One and helping the students learn from their own strengths in a positive "unending" environment which tied various aspects of learning into complex, relevant activities: thinking on a large scale, understanding analogies, making connections, discovering solutions.

To explain another way, Class:

Most of us grew up learning little isolated skills. We learned to Capitalize on the 9th week of school, let's say, in the 8th Grade.

Following that week, during which we'd be forced to learn the 60-odd capitalization rules for Friday's test, we'd leap into a couple days of hyphens and dashes, before going on to colons and semi-colons, and so on.

Isolated. Irrelevant. Boring. And not a good learning environment. We learned for the immediate tests and could not apply these "learned" skills to our daily (and real world) writings.

But such isolated, "testable" skills are a publisher's dream and an administrator's idea of Heaven. Because the kids can be tested on each of these isolated pieces, numbers can be attached to their names. These numbers can then be sorted into descending order and grades issued based on this garbage.

This has nothing to do with learning, with life-long skills, with internalizing and ownership. This has to do with outside forces trying to jam 19th Century methods down the throats of the people who will be running the 21st Century.

Bad stuff.

Take almost any English book you can get your hands on, and you will not find any writing activities (or few except in the most recent books and then as a way to thwart the movement away from texts). The books tell, tell, tell, tell how YOU are supposed to know this rule and that. The books test, test, test. They introduce the English materials in the most inane ways. For the most part, traditional English text books are sappy, to say the least, and anti-education to be really honest. And, except in a splashy, surface way haven't really changed since McGuffey's Readers of a century ago.

At the time of the Industrial Revolution the sum of human knowledge doubled about every 150 years; at the turn of this century it doubled about every 75 years; after World War II every 25 years; in 1990 every 9 months!

We still need to teach our kids skills, but we need to teach them DIFFERENT skills, better skills, more relevant skills, as "coverage" is impossible. [By the time a science book is researched and written and edited and printed and sold and distributed and finally used in a classroom it is already quite a few years out of date. And this is not just for info about our Solar System, for example, since the Voyager trips; it is about dinosaurs, which we know more about today than we did last year. Information progresses at a quantum rate, and this is true in every area of our real as well as academic lives.] Coverage is impossible, Class. Remember that. It's going to be on your next test.

We need to teach our kids HOW to think. Informational regurgitation is no longer relevant as we swing into the 21st Century. We need to teach our kids HOW to think, so they can be prepared for the future. And no matter how much we may long for the good ol' simple days of yore, they just ain't a'comin' back. We are - for better or worse - in the Electronic Age. And our kids, if they are going to compete with the rest of the world or if they are just simply going to keep America great, have got to become thinkers. They've got to become thinkers who

can use the tools of the future NOW.

Einstein (Albert) was asked for his phone number by a reporter. He looked it up in the phone book, astounding the reporter. Einstein explained that it would be foolish to clutter up his brain with anything that could be looked up.

If Einstein felt he should not be cluttering up his brain with useless information, maybe we could all take heed.

Let's give our kids and everyone else's kids a headstart for the next century by supporting our overworked teachers (instead of bashing them) and joining forces with them to provide a new environment in schools and in our homes. Let's advocate FOR our kids and their teachers. On 60-MINUTES, recently, Andy Rooney said the real problem with education today is not the teachers and not the schools but that "there are too many dumb kids," and, worse, too many dumb parents who don't prize education, who don't value learning (thus, too many dumb kids). I believe, truly, that we can get rid of this dumbness (which Steve Allen calls "DUMBETH" in a wonderful book by that name about the state of American thinking) by turning off the electronic babysitters (TVs and Nintendos) and get the kids into electronic tutors (computers) and maybe even (gasp!) books!

And here we are at the point of these classes: our TIs and what they can do to reverse this terrible dumbing trend in our country.

We'll take this up in our next class by introducing you to some of our brave TI-World educational experts and what they have offered and how we can use their gifts.

Your homework is to dust off all your your educational cartridges (which includes TI-WRITER, of course, as well as TERMINAL EMULATOR and MINI-MEMORY (think about it), as well as DRAGON MIX, READING RALLY, SCHOLASTIC SPELLING, and BEGINNING GRAMMAR). You don't have to pass in any papers next session, but you must be prepared to present a 10-minute talk on at least two of your selected cartridges, being prepared to defend its educational relevance to the child of the future.

Be early for TI-101 next time and get a good seat up front. Adios.

TIPS FROM THE TIGERCUB

No. 69

Tigercub Software  
156 Collingwood Ave.  
Columbus, OH 43213  
\*\*\*\*\*

My three Nuts & Bolts disks, each containing 100 or more subprograms, have been reduced to \$5.00 each. I am out of printed documentation so it will be supplied on disk.

My TI-PD library now has almost 600 disks of fairware (by author's permission only) and public domain, all arranged by category and as full as possible, provided with loaders by full program name rather than filename. Basic programs converted to XBasic, etc. The price is just \$1.50 per disk(!), post paid if at least eight are ordered. TI-PD catalog #5 and the latest supplement is available for \$1 which is deductible from the first order.

In Tips #68 I published my solution to Dr. Ecker's challenge to alternately assign X the value of A and B without using IF...THEN or any outside help. Computer Monthly has arrived again and his solution is better than mine. Try it with any two numbers -

```
100 A=2.765 :: B=-10
110 X=A+B-X :: PRINT X :: GO TO 110
```

There has been controversy for years as to whether the TI's pseudorandom number generator is truly random. Dr. Ecker's "Computer Fun & Learning" column in Computer Monthly had a question - if you randomly generate numbers between 0 and 9, how

often will you get the same number twice in succession? Three times in succession? And etc. Since there are 10 numbers to choose from, it seems to me you would get 2 in a row 10% of the time, 3 in a row 1% of the time, 4 in a row .1%...etc. I wrote this to prove it -

```
100 RANDOMIZE
110 C=C+1 :: X=INT(RND*10)::
PRINT X:: IF X=F THEN FL=F
L+1 :: CL(FL)=CL(FL)+1 :: PR
INT "" : FL : " " : CL(FL) : "C=" : C :
"%=" : CL(FL) / C :: GOTO 110 EL
SE FL=0 :: F=X :: GOTO 110
```

After 10,000 tries, I had 2 in a row 8.75% of the time and 3 in a row .83% and 4 in a row .07%. Does that prove anything? I don't know.

(Dr. Ecker points out that those percentages could not ever quite add up to 100%!) Here is another of my XBasic programs to write assembly source code -

```
100 DISPLAY AT(2,1)ERASE ALL
:"ASSEMBLY HELP SCREEN WRITE
R": "" : " This program will wr
ite the": "source code for an
assembly": "routine which ca
n be linked"
110 DISPLAY AT(7,1): "from Ex
tended Basic to dis-": "play
any one of several help": "sc
reens at any designated": "ke
y press or input at any": "po
int in a program."
120 DISPLAY AT(12,1): "The o
riginal source code,": "autho
r unknown, was improved": "by
Karl Romstedt and further":
"modified by Bruce Harrison.
"
130 DISPLAY AT(20,1): "How ma
ny help screens?" :: ACCEPT
AT(20,24)SIZE(1)VALIDATE(DIG
IT)BEEP:N
140 FOR J=1 TO N :: H$=H$&"H
ELP"&STR$(J)&" " :: NEXT J :
: H$=" " DEF "&SEG$(H$,
1,LEN(H$)-1)
```

```
150 DATA VMBW EQU >2024,V
MWR EQU >202C,KSCAN EQU
>201C,STATUS EQU >837C
160 OPEN #1:"DSK1.HELP/S",OU
TPUT :: PRINT #1:H$ :: FOR J
=1 TO 4 :: READ M$ :: PRINT
#1:M$ :: NEXT J
170 FOR J=1 TO N :: H$="HELP
"&STR$(J):: PRINT #1:H$&" L
WPI WS": " LI R13,HEL
PS"&STR$(J)
180 IF J<N THEN PRINT #1:"
JMP SAVSCR"
190 NEXT J :: H$=RPT$(" ",7)
200 PRINT #1:"SAVSCR CLR RO
":H$&"LI R1,SAVIT":H$&"LI
R2,768":H$&"BLWP @VMBW":H$
&"LI R9,NEWSCR":H$&"MOV R
9,R1":H$&"MOV R2,R4"
210 PRINT #1:H$&"LI R3,>60
00": "ADDOFF MOVW *R13+,*R9":
H$&"AB R3,*R9+":H$&"DEC R
4":H$&"JNE ADDOFF":H$&"BLWP
@VMBW"
220 PRINT #1:"KEYLOO BLWP @K
SCAN":H$&"BLWP @KSCAN":H$&"C
B @ANYKEY,@STATUS":H$&"JNE
KEYLOO"
230 PRINT #1:"REPL LI R1
.SAVIT":H$&"BLWP @VMBW": "RET
N LWPI >83E0":H$&"B @>6
A"
240 PRINT #1:"WS BSS 32
": "SAVIT BSS 768": "NEWSCR
BSS 768": "ANYKEY BYTE >20":
H$&"EVEN"
250 DISPLAY AT(3,1)ERASE ALL
: " Enter data just as you": "
want it to appear, in 24": "l
ines. Press Enter for blank"
: "lines."
260 FOR J=1 TO N :: DISPLAY
AT(12,1): "Ready for screen #
"&STR$(J): "" : "Press any key"
270 CALL KEY(O,K,S):: IF S=0
THEN 270 ELSE CALL CLEAR
280 ACCEPT AT(1,0):M$ :: PRI
NT #1:"HELPS"&STR$(J)&" TEXT
" "&M$&RPT$(" ",30-LEN(M$))
&" "
290 FOR K=2 TO 24 :: ACCEPT
AT(K,0):M$ :: PRINT #1:H$&"T
EXT " "&M$&RPT$(" ",30-LEN(M
$))&" "
300 NEXT K :: NEXT J :: PRIN
T #1:H$&"END"
310 DISPLAY AT(3,1)ERASE ALL
```

```

:" Source code has been writ
-:"ten to DSK1 as HELP/9. T
o":"assemble, insert Editor/
":"Assembler module."
320 DISPLAY AT(7,1):"Insert
Assembler disk in drive 1
.":"Select 2 ASSEMBLER":"Loa
d Assembler? Y":"Source file
name DSK2.HELP/S"
330 DISPLAY AT(12,1):"Object
file name? DSK2.HELP/O":"Li
st file name? Press Enter":"
Options? R"
340 DISPLAY AT(15,1):"Load t
he resulting object":"file i
nto your program by":"CALL I
NIT ::":"CALL LOAD( ""DSK1.HE
LP/O"" ) or,"
350 DISPLAY AT(19,1):"much b
etter. imbed it with":"ALSAV
E or SYSTEX."
360 DISPLAY AT(21,1):"Access
the screens in your progra
m by":" CALL LINK( ""HELP1"" )
":"CALL LINK( ""HELP2"" ), etc
."
370 CALL KEY(O,K,S):: IF S=0
THEN 370 ELSE CALL CLEAR

```

For instance, at any point in a program where keyboard input is required and user may not know what to do -

```
ACCEPT AT(24,1):M$ :: IF M$=
"HELP" THEN CALL LINK("HELP1
") and the first help screen
will pop up to give instruc-
tions. Press any key and the
previous screen reappears.
```

This time I am borrowing heavily from the TI\*MES news letter of England, which has also borrowed from the REC newsletter.

This one is useless, but is a remarkable example of compact complex programming. It shows that there is an algorithm for everything. See if you can figure out how it works -

```

100 CALL CLEAR :: FOR A=1 TO
2 :: FOR B=1 TO 4 :: X=2-AB
S(SGN(B-3)):: FOR C=1 TO X :
: PRINT CHR$(84-7*A+5*B-8*X)

```

```

::: NEXT C :: NEXT B :: PRIN
T CHR$(A+31):: NEXT A

```

Another useless one that is easier to figure out -

```

100 DISPLAY AT(1,1)ERASE ALL
:"NUMBER OF MONTH(1-12)"
110 ACCEPT AT(2,12)SIZE(2)VA
LIDATE(DIGIT):A :: IF A<1 OR
A>12 THEN 110
120 DISPLAY AT(3,1):A;"x 4="
;A*4 :: A=A*4
130 DISPLAY AT(4,1):A;" +13="
;A+13 :: A=A+13
140 DISPLAY AT(5,1):A;"x 25="
;A*25 :: A=A*25
150 DISPLAY AT(6,1):A;"-200="
;A-200 :: A=A-200
160 DISPLAY AT(8,1):"Input d
ate (1-31):" :: ACCEPT AT(8,
19)SIZE(2)VALIDATE(DIGIT):B
:: IF B<1 OR B>31 THEN 160
170 DISPLAY AT(10,1):A;"+";B
;"=";A+B :: A=A+B
180 DISPLAY AT(11,1):A;"x 2="
;A*2 :: A=A*2
190 DISPLAY AT(12,1):A;"-40="
;A-40 :: A=A-40
200 DISPLAY AT(13,1):A;"x 50="
;A*50 :: A=A*50
210 DISPLAY AT(15,1):"Input
last two digits of year e
g 91:"
220 ACCEPT AT(16,16)SIZE(2)V
ALIDATE(DIGIT):B
230 DISPLAY AT(18,1):A;"+";B
;"=";A+B :: A=A+B
240 DISPLAY AT(19,1):A;"-105
00=";A-10500 :: A=A-10500
250 DISPLAY AT(24,1):"ANY KE
Y FOR ANOTHER"
260 CALL KEY(5,A,B)
270 IF B<1 THEN 260
280 RUN
290 END

```

One for the little ones - change the string to anything you want.

```

1 REM SILLY PROG BY S SHAW
MARCH 1991
2 ! did you see COMPUTER WAR
S-the film? It is said that
the star, who was required t
o type fast into a computer

```

```

3 ! could not type, so a pro
gram just like this one was
used to give a good effect!
4 ! now adjust it how you wi
sh and show your friends how
fast you can type
5 ! at end of text string pr
ogram will just stop with th
is listing but can be modifi
ed to do anything you wish!
6 !
100 A$="This is how a non-ty
pist canproduce information
on screen quickly,witho
ut "
110 A$=A$$"having to look at
what keys are being bashed!
Just bash keys and watch ho
w perfect text appears no m
atter what you press."
120 CALL CLEAR :: PRINT A$:
: : : :
130 CALL KEY(5,A,B):: IF-B<1
THEN 130
140 C=C+1 :: PRINT SEG$(A$,C
,1):: IF C=LEN(A$)THEN 160
150 GOTO 130
160 GOTO 160

```

And a very fast routine to find prime numbers -

```

100 ! FIRST 100 PRIMES
-QUICKLY-
110 ! Dr H B Phillips
from THE REC NEWSLETTER
March 1988 Vol 3 #2
120 DIM P(300),X(12)
130 A=0 :: B=1 :: D=0.5 :: E
=180
140 M=100 :: L=3 :: F=0
150 ! increase M for more- a
lso increase DIMs.
160 PRINT 2:: C=B :: IF M=B
THEN END
170 L=INT((M/C)*L+F):: N=L+L
+B
180 FOR I=B TO INT((SQR(N)-B
)*D):: PP=P(I)
190 IF PP=B THEN 230
200 IF PP=A THEN PP=I+I+B ::
PRINT PP:: P(I)=PP :: C=C+
B :: IF C=M THEN END
210 IF X(I)=A THEN X(I)=(PP*
PP-B)*D
220 FOR J=X(I)TO L STEP PP :
: P(J)=B :: NEXT J :: X(I)=J

```



```

230 NEXT I :: IF F=0 THEN S=
I
240 FOR I=S TO L
250 IF P(I)=A THEN PP=I+I+B
:: PRINT PP;:: P(I)=PP :: C=
C+B :: IF C=M THEN END
260 NEXT I :: F=(M-C)*L/E ::
S=L+B
270 GOTO 170

```

And a demonstration of how the INTERRUPT routine works independently of whatever else the computer is doing -

```

100 REM interrupt demo
110 REM
120 REM MACHINE LANGUAGE
130 REM ROUTINE LOADED AT
140 REM >2600 XB OR E/A WITH
32K
150 REM >7200 MINI MEM NO 32
K
160 REM
170 CALL INIT
180 XM=9728
190 MM=29184
200 LAD=XM
210 REM TEST XB OR MM?
220 CALL LOAD(XM,170)
230 CALL PEEK(XM,X)
240 IF X=170 THEN 270
250 REM NO 32K MUST BE MM
260 LAD=MM
270 A=LAD
280 REM LOAD M/C
290 CALL CLEAR
300 FOR D=540 TO 630 STEP 10
310 CHECK=0
320 FOR N=1 TO 10
330 READ X
340 CALL LOAD(A,X)
350 CHECK=CHECK+X
360 A=A+1

```

```

370 NEXT N
380 READ X
390 IF CHECK<>X THEN 490
400 NEXT D
410 REM POKE INTERRUPT
420 REM ROUTINE ADDRESS
430 REM INTO >83C4
440 CALL LOAD(-31804,LAD/256
)
450 REM JUST IDLE AWAY TIME
460 FOR N=1 TO 9940
470 NEXT N
480 STOP
490 PRINT "ERROR IN DATA STA
TEMENT ";D
500 STOP
510 REM EACH DATA STATEMENT
520 REM HAS 10 DATA BYTES
530 REM PLUS A CHECK SUM
540 DATA 192,236,000,092,004
,194,005,131,002,131,987
550 DATA 000,060,026,003,004
,195,006,236,000,094,624
560 DATA 203,003,000,092,060
,172,000,090,006,002,628
570 DATA 017,015,019,010,006
,002,019,004,002,000,94
580 DATA 002,039,010,083,016
,002,002,000,002,086,242
590 DATA 096,003,016,007,002
,000,000,119,010,083,336
600 DATA 016,002,002,000,000
,072,160,003,002,096,353
610 DATA 064,000,006,192,215
,192,006,192,215,192,1274
620 DATA 016,000,216,044,000
,094,140,000,004,091,605
630 DATA 000,015,000,000,138
,128,000,000,000,000,281
640 END

```

Run that, then press FCTN 4. Enter LIST. Enter NEW. To stop it, enter BYE.

This is an oldie, but well worth repeating. You can use it to turn your cassette recorder on and off, to add speech or music from tape to a running program. With the proper hardware, you could write a program to control almost anything from the cassette port. If it doesn't work, reverse the polarity of the remote. Ed Hall wrote this -

```

100 CALL INIT
110 CALL LOAD(16368,79,70,70
,32,32,32,36,252)
120 CALL LOAD(16376,79,78,32
,32,32,32,36,244)
130 CALL LOAD(8194,37,4,63,2
40)
140 CALL LOAD(9460,2,12,8,45
,29,0,4,91,2,12,0,45,30,0,4,
91,203,78)
150 PRINT "PRESS": " P Play":
"S Stop"
160 CALL KEY(3,A,B)
170 IF B<1 THEN 160
180 ON POS("PS",CHR$(A),1)+1
GOTO 160,190,200)
190 CALL LINK("ON"):: GOTO 1
60
200 CALL LINK("OFF"):: GOTO
160

```

And that is just about -

**MEMORY FULL!**

Jim Peterson

The Philadelphia Area TI-99/4A Users' Group meets twice a month. On the second and fourth Saturday of the month at the Church of the ATONEMENT, 6200 Green St. Germantown (Corner of Greene St. and Walnut Lane) at 10 AM. We invite anyone that is interested in the TI-99/4A or Geneve to visit us. Stop in and see what is available to you for your TI and how membership can benefit you!

the Philadelphia Area TI-99/4A Users' Group or any of its officers responsible for any damage, inconvenience, or loss which may result as a consequence of the use of any written material herein.

TI-D-BITS is published monthly or bimonthly by the Philadelphia Area TI-99/4A Users' Group c/o Ralph Field, 603 N. Broad St. Lansdale, Pa. 19446. All material herein may be reprinted freely by other nonprofit User Groups, (unless otherwise stated), as long as proper credit is given to both source and author. Contributions are encouraged, but no payment is made. Editorial, advertising, and classified, copy MUST be in by the LAST day of the previous month. You can either mail your copy to: TI-D-BITS, The Philadelphia Area TI-99/4A Users' Group, c/o Ralph Field, 603 N. Broad St., Lansdale, Pa. 19446 or send it via modem by contacting Ralph Field at (215)-362-2534. If your piece contains any, diagrams, charts, or code, send a paper copy AT FINAL PUBLICATION SIZE.

**Current executive board consists of :**

|                         |              |
|-------------------------|--------------|
| <b>PRESIDENT</b>        |              |
| Norm Sellers.....       | 215-353-0475 |
| <b>VICE PRESIDENT</b>   |              |
| Allan Silversteen.....  | 215-885-7910 |
| <b>SECRETARY</b>        |              |
| Tim Coyne .....         | 215-947-5881 |
| <b>TREASURE</b>         |              |
| (OPEN)                  |              |
| Committees Consists of: |              |
| <b>TI-D-BITS</b>        |              |
| Ralph Field .....       | 215-362-2534 |
| Tim Coyne.....          | 215-947-5881 |
| <b>LIBRARY</b>          |              |
| (OPEN)                  |              |
| <b>MEMBERSHIP</b>       |              |
| Ralph Field.....        | 215-362-2534 |
| <b>EDUCATION</b>        |              |
| Barry Traver            |              |
| Allan Silversteen       |              |
| <b>EQUIPMENT</b>        |              |
| Allan Silversteain      |              |
| <b>PROGRAM</b>          |              |
| (OPEN)                  |              |

The editor of TI-D-BITS or the executive board of The Philadelphia Area TI-99/4A Users' Group reserve the right to reject any material submitted for publication for any reasons.

The Philadelphia Area TI-99/4A Users' Group's program library is available to all active members at NO CHARGE for copying to your disk. A charge of \$2.00 per disk is made for club supplied disks for members. Non members may obtain copies of the library for a fee of \$5.00 per disk. A catalog of the library's contents is given to all new members upon request and updates will appear in this publication from time to time. To obtain material from the library, contact the librarian for the best procedure to obtain your requests.

REMEMBER to be considerate when calling any of the above people. Limit your calls to early evening hours. (6pm to 9pm)

The opinions expressed herein are those of the individual authors and are not necessarily those of

