BUG BYTES

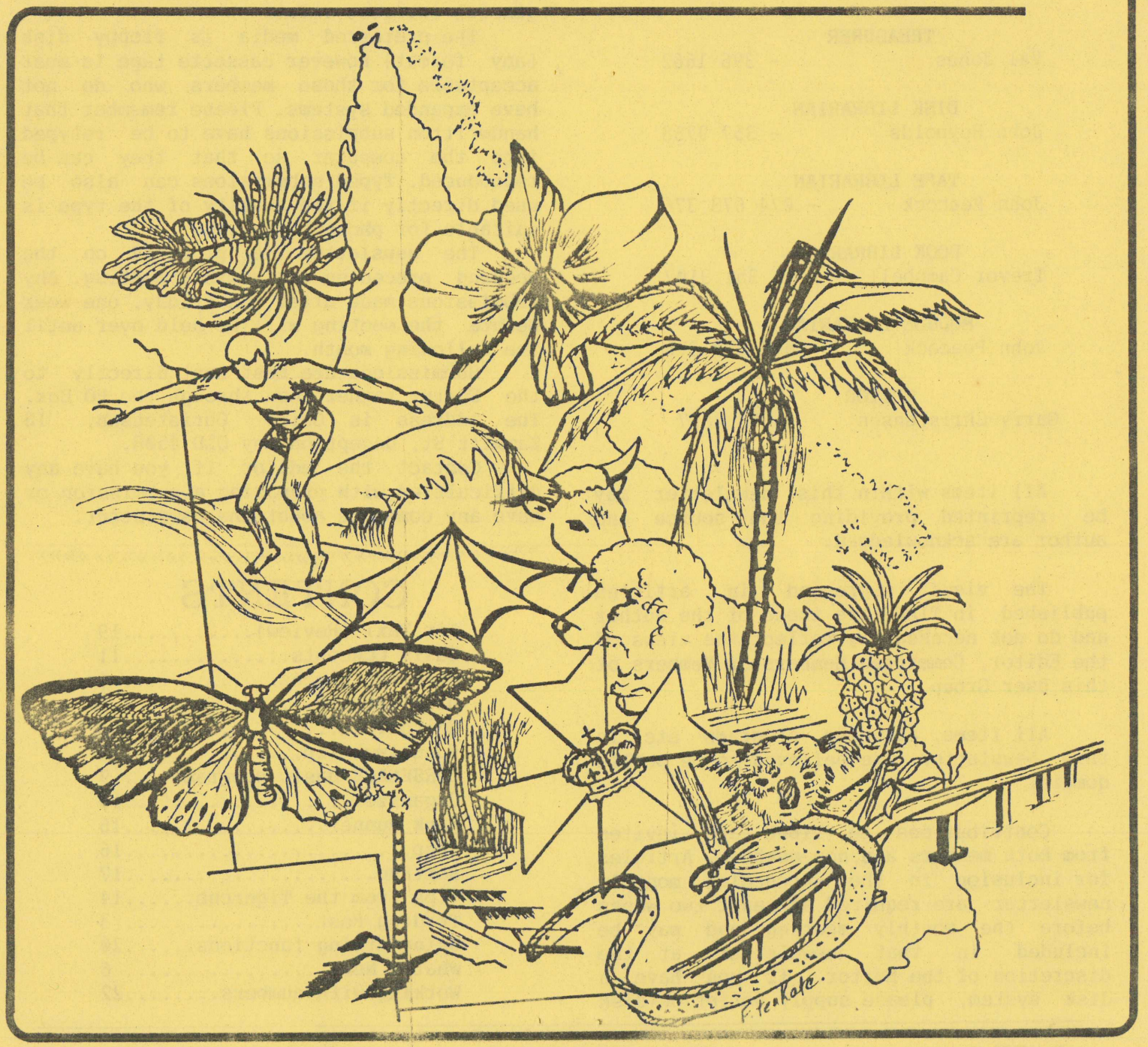
NEWSLETTER

of

TIBUG

TI - 99/4A - BRISBANE USER GROUP INC
P.O. BOX 3051
CLONTARF MDC, QLD AUST 4019

FEBRUARY 1992



COMING MEETINGS
28 FEB and 29 MARCH

7.30 p.m.
EAST BRISBANE STATE SCHOOL
CNR WELLINGTON RD. AND
STANLEY STREETS,
EAST BRISBANE.

COMMITTEE

PRESIDENT
Ralph Nielsen - 378 1237

VICE-PRESIDENT
Col Christensen - 284 7783

SECRETARY
Garry Christensen - 888 4857

TREASURER
Val Jones - 396 1662

DISK LIBRARIAN
John Reynolds - 357 9758

TAPE LIBRARIAN
John Peacock - 074 673 376

BOOK LIBRARIAN
Trevor Campbell - 351 3107

MODULE LIBRARIAN
John Peacock - 074 673 376

EDITOR
Garry Christensen - 888 4857

Contributions to TIBUG are invited from both members and non-members. Articles for inclusion in the succeeding monthly newsletter are required at least two weeks before the monthly meeting and may be included in that newsletter at the discretion of the Editor. If you have a disk system, please supply script on disk with diagrams separately on paper and as clear and black as possible to facilitate photocopying.

Most original articles by members of TIBUG in this newsletter are on disk and are available to other User Groups on request.

Submissions of articles, reviews, comments and letters from mebers is encouraged, however the editor would ask that members keep the following in mind.

Submissions should be about computers, the TI community in particular, or have general interest value.

The preferred media is floppy disk (any format) however cassette tape is most acceptable for those members who do not have expanded systems. Please remember that handwritten submissions have to be retyped into the computer so that they can be reproduced. Typed submissions can also be used directly if the quality of the type is suitable for photocopying.

The newsletter is produced on the weekend preceding the monthly meeting. Any submissions made after the Friday, one week before the meeting will be held over until the following month.

Submissions are best sent directly to the Editor rather than through the PO Box. The address is Garry Christensen, 18 Zammitt St, Deception Bay QLD 4508.

Contact the editor if you have any difficulties with preparing a submission or have any comments about the newsletter.

»»»»»»»»»»»»»»»»»»»»»»»»»»»»»»»»»»»»

CONTENTS

Air Taxi (Review)..........19
Basic Concepts.............11
Bits and Pieces............ 4
Disk Library............... 7
Editorial.................. 3
Funnelweb Tip..............13
HORSERACE Basic program..... 9
Organize...................19
Rock Runner................16
Shop.......................16
Sorting....................17
Tips from the Tigercub......14
Trading Post............... 3
Using string functions......20
What's News................ 6
Working with numbers........22

EDITORIAL

When I first joined TI-BUG, I was young (relatively), computer naive, and certainly inexperienced. With the help of some of the group members, I learned more and I soon submitted my first article to the newsletter for publication. It was a proud moment for me. It was, of course, only the first of many and I feel that I am now a part of the group's continued success.

This is not how I had envisioned the future when I wrote that first article and I still had no idea of what lay in store for the group when I was soon prevailed upon to accept the position of acting treasurer. The next AGM saw the acting dropped and I was a formal part of the committee. Time took its toll though and the leaders of the group went on to other things and it seemed that more and more of the responsibility for the operation of the group passed to me.

All this is fine but its only memories now, but the lessons are well learned. The group always needs fresh blood. Without it the group goes stale and dies. I have said before that console only users are the presidents of the future. We should be spending effort on them, not just coasting along and talking about disk drives, ramdisks and 80 column devices.

As editor I try to include something in each newsletter that will help the new and less experienced members. It may be basic programs to type in, or programming tips, or a listing of some modules or programs that are available to them. Some of the more experienced members may find it boring and a waste of space but trust me when I say the without it the group would whither like fruit in a drought.

All this is not rhetoric. It has been proved time and again. When an expanded system comes up for sale it is usually sold to a member who is looking for more from this great computer. That member then really starts learning about computers. After a while they are ready to start sharing that knowledge with others and they start having a say in the running of the group.

To conclude this editorial I would like to say that it makes me glad to see the efforts that other committee members are making. I have seen members who joined us with only a console and who are now taking responsible positions in the group. I used to worry that if I had to move away from Brisbane, what would become of the group. It seemed that I was doing a very large percentage of the work. That doesn't wory me so much any more. I look at the members of the committee and those who attend the meetings and see a new generation of enthuiasm and ideas beginning to emerge.

Keep it up team and TI-BUG will be here for some time yet.

»»

TRADING POST

FOR SALE - 2 joystick adaptors. Plugs into computer. Any commercial joystick can plug directly into the J1 or J2 socket without modification. $15. Phone Col 284 7783.

FOR SALE - As a complete unit, TIBUG's fully expanded system including console, modulator, transformer, Expansion box, RS232 card, Memory expansion card, Disk controller card, 2 slimline DSSD disk drives, and Extended Basic module and manual. $450 or $470 with free club membership. Phone Col 284 7783.

FOR SALE - Geneve 9640. Complete with manuals and lots of software. Phone Larry 07 202 1884

FOR $ALE - TI-99/4A complete with manuals, joysticks, cassette lead, modulator and transformer. $45. Phone Allan (075) 342 572.

FOR $ALE - Expanded system with 2 consoles and manuals. Disk controller card, Mem expansion, but no RS232. I DSSD half height drive in PEB and 2 bare full height SSSD drives. Ext Basic, about 8 modules and 20-30 disks of assorted programs. $450. Phone Ross 285 2173.

WANTED - FOR $ALE - FOR EXCHANGE
What do you have or need that can be listed in this column? Contact Garry or Col with details.

BITS & PIECES

by Col Christensen

Here we are near the end of February in a new year, the drought in many areas has broken and the floods have subsided. My trip to the west and north of our fair state was enjoyable (and hot). Emerald certainly lives up to its name. The town is neatly laid out and kept and the surrounding farms are covered with acres and acres of green cotton plants through which irrigation channels weave their way. At Emerald Margaret and I were able to call on David and Carolyn Kroll and their two children and spent an enjoyable time with them. Firstly a short trip to the Fairburn Dam, a lovely place with a sandy beach and parklands bordering it with the natives engaged in activities including swimming, skiing, sailing, fishing and jetboat riding. After an enjoyable tea, David and I eventually got round to discussing our TI-99/4As. This continued till about 3 a.m. next morning by which time both Margaret and Carolyn had long since retired for the night.

BASIC CONSOLE EXPANSION

I have been spending a considerable amount of time lately on a mini expansion. The aim is to provide both a word processor and a games machine at a reasonable cost to enhance the uses for a basic console. This arrangement would suit the needs of your second computer without the need for full expansion. I want to put out feelers to gauge the interest by members in such a thing before I go much further. Its projected features are:

1. Plugs into I/O port of the computer with the circuit board vertical at the side of the console and incorporating an I/O port extension.
2. Runs off the console supply voltages without the need for external transformers.
3. Speech Synthesizer able to be plugged into I/O extension of the mini expansion.
4. 32k memory expansion fitted.
5. DSR to support a PIO printer output port. This feature has already been developed.
6. DSR eprom to contain a number of CALLS that can be called from Basic or XB.
7. CALL GL (Game Loader). Calls an assembly program in the DSR eprom that will load assembly games from cassette tape. This feature is now operational in the prototype and loads any assembly program that can normally be loaded by the RUN function of the E/A module.
8. CALL ED (EDitor). This will load a character set and an Editor program stored in the Eprom and put them into memory. Great for the student doing an assignment and printing it out using the PrintFile command when the main system is otherwise engaged. Unfortunately I have not yet found an editor that will load and run correctly in the normal RUN environment. The original TI Edital/2 files rely on the TI-Writer module in some way and will not run alone. Funnelweb's ED and EE likewise require Fweb to be in memory first.

A small optional circuit board to interface to a horizon type ramdisk to the I/O port or the I/O extension.

The only real holdup is the Editor program and AN EXPRESSION OF INTEREST ON YOUR PART if you would be likely to purchase one. I would estimate the following prices allowing $30 for a commercial circuit board:

With Memory expansion only.....$55
With PIO port + mem exp and all..$100
I/O interface to ramdisk.....$15

SPELLIT and RAMDISKS

In the latest MICROpendium is a short article on the problems encountered when using Spellit on a ramdisk. See the article elsewhere in this edition. If it doesn't appear and you want info, phone Garry 888 4857.

RAINFALL CHART

Now that the skies have learned to rain again this little program could be of use. It prints a year on a page on which you can record each day's rainfall.

```
 100 OPEN #1:"PIO" :: PRINT #
1:CHR$(27);"3";CHR$(27);
110 A$=" "&RPT$(" .... ",12)
&" "
120 PRINT #1:CHR$(14);TAB(16
);"YEAR .......": : :
130 PRINT #1:"     JAN   FEB
   MAR   APR   MAY   JUN   J
UL   AUG   SEP   OCT   NOV
 DEC"
140 PRINT #1
150 FOR I=1 TO 31
160 IF I<10 THEN I$=" "&STR$
```

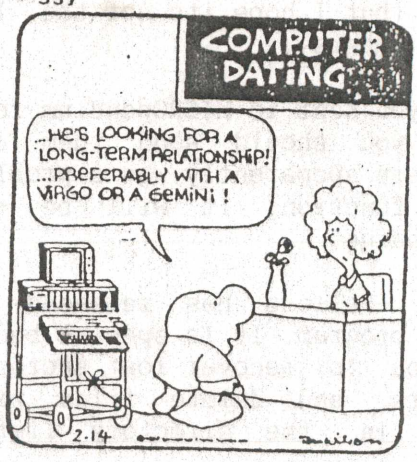
```
(I)ELSE I$=STR$(I)
170 PRINT #1:" ";I$;A$;I$
180 PRINT #1
190 NEXT I
200 PRINT #1: : :" TOT ....
210 CLOSE #1
220 GOTO 120
```

GRAPH PAPER

Print a page of small, medium or large squares with this program. The smallest have sides of .1 inch and the largest 1.0 inches. It will also print the page with just dots at the corners of each square.

```
100 REM ******************
110 REM * TO MAKE SQUARED *
120 REM *PAPER - .1" TO 1"*
130 REM ******************
140 REM Change line 190 to s
uit your printer graphics co
des for + shaped cross, horz
ontal line and vertical line
150 DATA "1   inch squares",
9,24,8,7,.9  inch squares,8,
24,7,8,.8  inch squares,7,19
,8,9,.75 inch squares,8,18,8
,10
160 DATA .7  inch squares,6,
19,7,11,.6  inch squares,5,1
6,7,13,.5  inch squares,4,18
,5,15,.4  inch squares,3,22,
3,19
170 DATA .3  inch squares,2,
16,3,26,.25 inch squares,2,1
4,3,31,.2  inch squares,1,22
,1,39,.1  inch squares,0,22,
0,79
180 CALL CLEAR
190 PLUS$=CHR$(128):: HORIZ$
=CHR$(133):: VERT$=CHR$(134)
200 DISPLAY AT(1,2):"Select
the size of square":"     yo
u wish to print"
210 RESTORE 150
220 FOR I=1 TO 12 :: READ A$
,GAPS,SPAC,LINES,RPTS :: DIS
PLAY AT(I+3,5-LEN(STR$(I))):
STR$(I);".  ";A$ :: NEXT I
230 DISPLAY AT(20,5):"Which
size?" :: ACCEPT AT(20,17)VA
LIDATE(DIGIT)BEEP SIZE(2):SZ
E
240 IF SZE>12 OR SZE<1 THEN
CALL SOUND(50,220,5):: GOTO
230
250 DISPLAY AT(24,5):"GRID o
r DOTS  G" :: ACCEPT AT(24,1
9)VALIDATE("GDgd")SIZE(-1)BE
EP:GD$
260 IF GD$="D" OR GD$="d" TH
EN PLUS$=CHR$(46):: HORIZ$,V
ERT$=" "
270 RESTORE 150
280 FOR I=1 TO SZE :: READ A
$,GAPS,SPAC,LINES,RPTS :: NE
XT I
290 CALL CLEAR :: DISPLAY AT
(12,2):"IS PRINTER SWITCHED
ON? Y" :: ACCEPT AT(12,26)VA
LIDATE("YNyn")SIZE(-1)BEEP:O
N$
300 IF ON$="N" OR ON$="n" TH
EN 290
310 DISPLAY AT(12,1):"PRINTI
NG";RPTS;A$:"    per line.":
 : : : :"  To STOP, hold any
 key": :"  until square is f
inished"
320 OPEN #1:"PIO",VARIABLE 9
6 :: PRINT #1:CHR$(27);"P";:
: PRINT #1:CHR$(27);"m";CHR$
(4);CHR$(27);"U";CHR$(1)
330 IF SZE=4 OR SZE=10 THEN
PRINT #1:CHR$(27);"M";
340 PRINT #1:CHR$(27);"3";CH
R$(SPAC)
350 IF SZE=12 THEN PRINT #1:
RPT$(PLUS$,80):: CALL KEY(0,
K,S):: IF K<>-1 THEN CLOSE #
1 :: GOTO 180 ELSE 350
360 LINE2$=VERT$&RPT$(RPT$("
 ",GAPS)&VERT$,RPTS)
370 LINE1$=PLUS$&RPT$(RPT$(H
ORIZ$,GAPS)&PLUS$,RPTS)
380 PRINT #1:LINE1$
390 CALL KEY(0,K,S):: IF K<>
-1 THEN CLOSE #1 :: GOTO 180
400 FOR I=1 TO LINES :: PRIN
T #1:LINE2$ :: NEXT I
410 GOTO 380
```

Ziggy



WHAT'S NEWS

A bit of of confusion. I have phoned OPA twice in the last month to enquire about the TIMs. The first call informed me that they would be ready 'next week'. The next call (2 weeks later) said that they started posting them at 5 per week since early January. They still haven't arrived so I will be giving them about another week and then phoning them again. I hope that I will get some sense out of them this time.

Remember WOODSTOCK, that great set of animation programs. It is said that another will be due for release soon. These are freeware so we hope to have a copy sent to us when it comes out.

Harrison Software has released a tape of MIDI music. It is aimed at those who are interested in the music but do not have the MIDI-Master 99. The tape runs for 45 minutes and all pieces bar one were produced with a TI99 attached to a Casio CT-650 keyboard. The other used a PC. Cost is $10. 5705 40th Place, Hyattsville MD 20781, USA.

Here's a tip for HFDC owners. If you want to use a high capacity drive, 8 or more heads, you will find that the controller will not work properly. The HFDC uses ST506 interface while many of these drives use ST504. To change to ST504 firstly cut the trace from pin 5 on chip U9 to pin 5 on U17, then connect a jumper from pin 12 of U9 to pin 5 of U17. The HFDC will still work with lower capacity drives. This tip comes from Barry Boone.

I have reported previously that Myarc were working on repairing the HFDCs that had been returned to them. Some have finally been repaired and returned to their owners. It seems that there is some life in Myarc yet (but I hope its not the last of the dying embers).

If you sent to MICROpendium for their Index II, you should send them a quick note. There apoparently was a problem with the query function. It will be replaced free of charge.

Norm Sellers has released a disk directory program. It is special because it allows you to recover lost sectors, mark bad sectors, and delete files with bad sectors. In the directory listing it includes the date stamp, even for the TI card (?), and also if the file is BASIC, XB, assembly embedded in BASIC or XB, or a data file. The listing can go to the screen, printer, or disk file. Sorry, I don't have an address but I hope to have one for next month.

One of our members has had some difficulty with SPELLIT when run from a ramdisk. It seems that it will not operate correctly from a ramdisk from any CRU address other than >1000. Ron Kleinschafer from what used to be the HV99ers has produced a fix to correct the problem. He has also inserted a section of code into the original that will take the input filename from Funnelweb's mailbox. When run from Funnelweb though, there will be some problems so you must exit from Funnelweb first. I hope to have a copy of the update soon.

New from Asgard is Classic Checkers. It uses the keyboard, joystick of mouse to move the pieces. You play the computer or two can play, using the computer as a playing board. The price is $14.95. Also new is Ti Pei. This is the first mahjongg game for the TI or 9640. Price $14.95 plus postage. PO Box 10306, Rockville MD 20849, USA.

FANTI is a program that is said to be nearly the same as TI-Artist and is a fairware offering from a French user group. Ver 1.2 is available in English and they are still working on the translation for Ver 1.3. Write to Jean Louis Cangy, 465 bat J cite Enrilise, 85000 La Roche Sur Yon, France.

Rave 99 are now offering their Speech Adapter Card in kit form. This card lets the Speech Synthesizer to be installed in the PE box. The kit costs $35 (US) plus postage. Also in kit form is their PE/2 expansion system. This kit includes with a backplane with space for 8 cards and guides and standoffs. I don't know if this kit could be installed into a PC case, but it raises some interesting possibilities. The cost is $150 (US) plus postage. For $240 you can get the Advanced PE/2 kit that also has the 99Flex-Card for connection to the TI99/4A. Rave 99 Company, 112 Rambling Rd, Vernon CT 06066, USA.

Comprodine are distributing a program called Artist Cardshop. It was written by

Paul Coleman and is designed specifically for producing cards (birthday, Christmas, etc). Features are direct access to TIA fonts and instances, 2 fonts and 4 instances on each side of the card (inside and out), allows printing on the back of the card, save each card to disk, single and double-density printing, and prints each card in about 3 minutes. The package includes 25 borders as well as a border making program and a 28 page instruction booklet. $20 plus postage. Comprodine, 1949 Evergreen Ave, Fullerton CA 92635, USA.

Harrison Software have released all of their assembly music disks to the public domain. They have also released a disk called Volume 2 that contains utilities for XB programmers. They include a fast menu driver, routines to quickly assign values to arrays, and a EA option 5 loader.

Heard of Airtaxi? From memory it is a program that allows you to navigate a plane over most of North America (not a flight simulator though). Son of Airtaxi has now been released. This is essentially the same except the maps now include Europe, Africa, South America, West Indies, Far East and most importantly Australia. The collection is available for $10, or $1.25 per program. Don Shorock, PO Box 501, Great Bend KS 67530-0501, USA.

««

DISK LIBRARY

ALL MUSIC. FILENAME MUSIC.

1 ABIDE WITH ME
2 BEULAH LAND
3 IN THE SWEET BYE & BYE
4 THE COWBOY'S DREAM
5 AN EVENING PRAYER
6 FAITH OF OUR FATHERS
7 WHAT A FRIEND WE HAVE..
8 IN THE GARDEN
9 BEAUTIFUL ISLE/SOMEWHERE
10 JUST AS I AM
11 CHILD OF THE KING
12 THE LAST MILE OF THE WAY
13 NEARER, MY GOD TO THEE
14 NIGHT WATCH
15 GATHER AT THE RIVER
16 I'LL BE THERE
17 THE OLD RUGGED CROSS
18 EYE ON THE SPARROW
19 THE GREAT SPECKLED BIRD
20 I LOVE TO TELL THE STORY
21 BEYOND THE SUNSET
22 WHEN I TAKE MY VACATION
23 JUST A CLOSER WALK
24 I WONDER AS I WANDER
25 WHISPERING HOPE

These Gospel Tunes are not presented in normal manner, because they are all on the one DSK.

They are controlled by a LOAD Prg. with many otions on how to play them.

They also have varying tempo's which only enhance the presentation.

Many different Fonts, Colours & Screen variations are used.

All are XB.

384 384 384

$$$$$$$$$$$$$$$$$$$$$$$$

THANK YOU VERY MUCH,

JIM PETERSON.

$$$$$$$$$$$$$$$$$$$$$$$$

RATING VG***

ABIDE WITH ME	291
ANGELS WE HAVE HEARD ON HIGH	315
BLESSED ASSURANCE	291
CAROLING CAROLING	315
CHRISTMAS MEDLEY	291
CHURCH IN THE WILDWOOD	291
CLOSER WALK WITH THEE	291
DECK THE HALLS	315
ETERNAL FATHER	291
FINALE (PLAY LAST)	291
GOOD KING WENCESLAS	315
HARK THE HERALD ANGELS SING	315
I LOVE TO TELL STORY	291
I NEED THEE EVERY HOUR	291
JESUS LOVES CHILDREN	291
JESUS SAVES	291
JINGLE BELL ROCK	315

These Carols & Hymns were presented by Bill Knecht. Excellent music and video. They should be taken as a complete disk as they run from a load program.

If you copy Disk please send $5

JUST AS I AM	291	
HOLY HOLY HOLY	291	(for each disk) to Bill Knecht, at
LET IT SNOW	315	
LITTLE DRUMMER BOY	315	the address shown on the disk.
LORD'S PRAYER	291	
LOVE LIFTED ME	291	
O CHRISTMAS TREE	315	
O COME ALL YE FAITHFUL	315	THANK YOU VERY MUCH, BILL KNECHT.
O HOLY NIGHT	315	
OLD RUGGED CROSS	291	
OLDTIME RELIGION MEDLEY	291	
PROGRAMMING COMMENTS	291	
ROCK OF AGES	291	
SANTA CLAUS IS COMING TO TOWN	315	
SLEIGH RIDE	315	VG***
SOFTLY & TENDERLY	291	
WE WISH YOU A MERRY CHRISTMAS	315	
WHERE HE LEADS ME	291	
WHY ME	291	

****

9/SYMPHONY	XB	Same by Beethoven. No Pic.	OK.		018
@SOUNDI/3	EA	EAC5.You can create load & save music.	OK.		286
AIRFORCE SONG	XB	Pic and Music. Short version.	G.	AIRFORCE	212B
ALWAYS	XB	Plays tune, No pic.	OK		218
ANGELS FROM REALMS	(EA).	Piano key mvemnts.1,2,3 select.	G.	PIANO	348
ANNAMAG 1-20	XB	Has Load. Care Disk is a flippy.	G.		362B
ARKANSAS TRAVELLER	XB.	No pic, tune on fiddle, medley 10	G.	TUNES	212B
AULD LANG SYNE	XB	Plays tune, shows words, short version.	OK.	AULDLANGSY	035B
AUSIEMUSIC	XB	Rd to Gundagai & 2 others not known.	OK.	AUSIEMUSIC	010
BACH	??	GOLDBERG 30 Variations of Aria.	OK.		373
BACH	XB	BACH 787-801. Has Load.	G.		362A
BACH/INVEN	EA	BACH 772-786.	G.		363A
BACH1+BACH2	EA	Multicoloured petterns, plays Bach.	OK.	BACH1	254
BATTLEHYMN OF REPUBLIC	XB	No vis no words.	G.	BATTLEHYMN	212A
BREEZALONG	XB	XB/BAS. You'll know this.Can vary time	G.		010
BUMBLE/BOO	BAS	Plays Bumblebee, shows 1 screen.	OK.		336
BUMBLEBOOG	XB	Jack Fina-Sam Moore. Bumble Boogie	G.		009
CAISSON (ARMY SONG)	XB.	Pic and music. Short version.	G.	ARMY	212B
CHIMES	XB	Plays several chimes, shows numbers.	OK		218
CINCINNATI HORNPIPE	XB.	No pic, tune on fiddle, medley 10	G.	TUNES	212B
COCK AND HEN	XB	No pic, tune on fiddle, medley 10	G.	TUNES	212B
CONGO/Q	EA	DLOADJS.Good video & score.JS adds too.	G.		111
DANCER	XB	Plays tune, dont know full name.	g.		218
DECK THE HALLS	XB	R. Jennings. Shows the words.	OK.	DECKHALLS	010
DEMO	XB	Variations of tune ALWAYS	OK		218
DUEL BANJO	XB	Gary C.See Next. Use LOAD/DB/O	VG.	DB/O	013
DUEL/BANJO	EA	DLOAD.Good video plays tune. Gary C.	VG.	DUEL/BANJO	112
DUET	XB	Spagnoletto. Plays tune unknown.	OK.		008
EAR TRAINING	XB	Learn chord/pitch/rythm recog. Etc	G.	EAR/TRAIN	008
ELECTRIC DREAMS	XB	Plays this tune ad. by Sid Michel.	G.	ELEC-DREAM	388
ENDTAG	XB	I dont understand it.	??		218
ENTERTAINER	XB	Shows pic, plays tune. Also on 010.	OK.	ENTERTAINE	026B
ENTERTAINER	XB	Shows pic, plays tune. Also on 010.	OK.	ENTERTAINR	010B
FANTASY	XB	Plays that tune by Thomas Morley.	G.		008
FREREJACQ	XB	Old French song. Repeats.	OK.		010
GARY OWEN	XB	No pic, tune on fiddle, medley 10	G.	TUNES	212B
GHOSTBUSTERS	XB	Small pic and music. Full version.	G.	GHOSTBUSTR	212B
GOD FATHER	XB	M. Gilbreath. Plays that tune with pic.	OK.	GOD/FATHER	008
GROOVY	XB	Shows pic and plays tune.	OK.		254
GUITARS	XB	Shows pic and plays tune.	OK.		013
HALLELUJAH	XB	Garry Christensen. Plays that tune.	G.		008
HALLELUJAH CHORUS		XB. Has pics with sprites.	OK.	HALL/CHOR	010
HAUNTED HOUSE	XB	Pic of house, plane flies over. Game??	OK.	HAUNTEDHS	212B
HOME SWEET HOME	EA	Shows pic and plays. Arrow key to REDO	OK.	XMASTREE/F	108
HOUSE/RISING SUN	XB	Shows pic, plays tune, graphics good.	G.	RISING/SUN	228
HOUSE/RISING SUN	XB	Shows pic, plays tune, graphics good.	G.	RISINGSUN2	388
IMPOSSIBLE DREAM	36BAS.	Plays this tune with static pic.	OK.	QUIXOTI	383
IN THE GARDEN	XB	Plays tune, no pic.	G.	INGARDEN	218
IRISH WASHERWOMAN		XB. No pic, tune on fiddle, medley 10	G.	TUNES	212B
JOY TO THE WORLD	EA	Shows key mvements.1,2,3 select.	G.	PIANO	348
KILL ME SOFTLY	XB	Has pic and words.	OK.	KILLSOFTLY	010
LINERNOTES	XB	Reflections on composing for TI	OK		224
LOONEY TUNE	XB	Draws Buggs Bunny and has good finish.	OK.	LOONTUNE_X	215B
MAINSCREEN	XB	Looks like TI screen. Plays tune.	OK.		009
MAME	XB	Plays tune and shows words.	G.		010
MAPLE LEAF RAG.	XB	Good music, no visual display.	OK.	MAPLLEAF_X	215B
MARINE SONG	XB	Pic and music. Short version.	G.	MARINE	212B

MASH	XB	Good visual display, has words.	VG.	MASH_X	215B
MC DONALD'S	XB	No pic, tune on fiddle, medley 10	G.	TUNES	212B
MORNING HAS BROKEN		Sam Moore. Plays tune, good pic.	G.	MORNINGBRK	009
MOZART#40	XB	40th Symph in G Minor. Phil West.	OK.		008
MUSIC MASTER	XB	Plays tune, allows change. Beginners.	G.	MUS/MASTER	008
MUSIC BOX DANCER		XB. No pic, but is it Xylophone?	G.	BOXDANCER	010
MUSIC.D.E	EA	Shows pics, plays several Xmas tunes.	OK.		223
MYST/MELDY	XB	Has 50 tunes, you guess from 2.	G.		013
MYS/MELODY	XB	Has 50 tunes, you guess from 2.	G.		336
NAVY SONG	XB	Pic and music. Short version.	G.	NAVY	212B
NUTCRACKER	XB	PETE1-8. Has load & print.	G.		363A
ODETOPUPPY	XB	Dogs racing pic & Building. Plays tune.	OK.		009
OPUS17	XB	Has load.Dsk crashed this survived.	G.		360
ORGANPLAY	XB	Others as well. You play as shown.	G.		009
OXYGENE	XB	Shows pic, plays tune.	OK.		010
PASS ME BY	XB	Father Goose. Good vis. Has words	G.	PASSMEBY_X	215B
PENSYLVANIA POLKA		XB. Static Pic Good Music,CLR to Exit	G.	PENPOLKA_X	215B
POP GOES THE WEASEL		XB. No pic, tune on fiddle, medley 10	G.	TUNES	212B
PUPPYTOWN3	XB	Dogs racing,plays tune.Sim ODETOPUPPY	G.		388
RHAPSODY IN BLUE.XB		4 Files and load to play.	OK.		252
RHAPSODY IN BLUE.EA		EAC5.Tune played. By G.Gershwin.No Pic	G.	RIB1/4	286
ROBOTBOOGY	XB	Sam Moore. Plays tune and shows pic.	OK.		009
SATURDAY	XB	Carpenters. Same moving Picture, but	OK.	SATURDAY_X	215B
SENSATION RAG	XB	Scott Joplin. Visual and tune..	OK.	SENSATON_X	215B
SHOW ME WAY TO HOME.	XB	Irving King. Good Vis. Words shown.	VG.	SHOMEWAY_X	215B
SILENT NIGHT	EA	Plays tune,shows Piano key movements.	G.	PIANO	348
SNOOPY	XB	BAS. Good pic and music.	G.		008
SNOOPYXMAS	XB	BAS. Pic and music. great for young.	G*		008
SOLDIER'S JOY	XB	No pic, tune on fiddle, medley 10	G.	TUNES	212B
STARS AND STRIPES		XB. Full version with pic.	G.	STR&STRPE	212B
STARTREK MUSIC	XB	4 diff themes, shows Pic.	G.		224
TAKE5	XB	Tune unknown to me.	G.		254
TIME-DATA		Goes with TIMEBOTTLE			009
TIMEBOTTLE	XB	Pic knight in forrest. Plays tune.	G.		009
TIZART	XB	Tune for C. by Mozart.	F.		026B
TOCCATA	EA	DLOAD.Plays this tune by Gary C.	VG.	TOCCATA	110
TOM AND JERRY REEL		XB. No pic, tune on fiddle, medley 10	G.	TUNES	212B
TONE/GUESS	XB	Guess the tones given. Educational.	G.		008
TUCKERBOX	XB	Russell Welham, SYD. Guess tune.	G.		010
TUNEUP	XB	This plays tunes for you to match.	OK.		010
TWO FORTY REEL	XB	No pic, tune on fiddle, medley 10	G.	TUNES	212B
VENETIAN-S	XB	V. Boat Song. Sam Moore. Shows pic.	OK.		009
VENUSSCAPE	XB	Shws pic, plays tune.	G.		009
WEIRDMUSIC	XB	True, different, try it.	OK.		009
WEST BOOGIE	XB	Sam Moore. Shows small Pic.	OK.		009
XMASTREE/F	EA	D.LOAD.Plays Merry Xmas,shows pic.	OK.		108
YES NO BANANAS	XB	Good Vis, with words.	G.	BANANAS	212A
YESTERDAY	XB	Plays title tune and shows words.	OK.		013
YOU LIGHT UP MY LIFE		Has 2 files and pic.	F.	LIGHT-UML	010

HORSERACE - Basic Programme

```
100 CALL CLEAR
110 PRINT TAB(11);"HORSERACE
"
120 PRINT "COPYRIGHT BY W.BA
LLSCHMIETER"
130 PRINT "5 GULLAND ST.,NTH
.IPSWICH"
140 FOR J=1 TO 2000
150 NEXT J
160 RANDOMIZE
170 MB1=100
180 MB2=100
190 GOSUB 520
200 CALL CLEAR
210 CALL SCREEN(16)
220 A$="081627957C7C4284"
230 CALL CHAR(96,A$)
240 CALL COLOR(9,2,1)
250 CALL CHAR(104,A$)
260 CALL COLOR(10,9,1)
270 CALL CHAR(112,A$)
280 CALL COLOR(11,8,1)
290 CALL CHAR(120,A$)
300 CALL COLOR(12,11,1)
310 CALL CHAR(128,A$)
320 CALL COLOR(13,5,1)
330 CALL CHAR(136,A$)
340 CALL COLOR(14,13,1)
350 CALL CHAR(144,A$)
360 CALL COLOR(15,7,1)
370 CALL HCHAR(1,13,72)
380 CALL HCHAR(1,14,79)
390 CALL HCHAR(1,15,82)
400 CALL HCHAR(1,16,83)
410 CALL HCHAR(1,17,69)
420 CALL HCHAR(1,18,82)
430 CALL HCHAR(1,19,65)
440 CALL HCHAR(1,20,67)
450 CALL HCHAR(1,21,69)
460 GOSUB 2080
470 FOR Z=2 TO 18
480 CALL VCHAR(Z,3,91)
490 CALL VCHAR(Z,30,93)
500 NEXT Z
510 GOTO 580
520 CALL CLEAR
530 CALL SCREEN(16)
540 PRINT TAB(11);"HORSERACE
"
550 PRINT
560 GOSUB 1250
```

```
570 RETURN
580 E=3
590 E1=3
600 E2=3
610 E3=3
620 E4=3
630 E5=3
640 E6=3
650 K=4
660 CALL HCHAR(K,E,96)
670 L=6
680 CALL HCHAR(L,E1,104)
690 M=8
700 CALL HCHAR(M,E2,112)
710 N=10
720 CALL HCHAR(N,E3,120)
730 P=12
740 CALL HCHAR(P,E4,128)
750 P1=14
760 CALL HCHAR(P1,E5,136)
770 Q=16
780 CALL HCHAR(Q,E6,144)
790 GOSUB 2250
800 R=INT(RND*7)+1
810 S=INT(RND*2)+1
820 ON R GOTO 830,890,950,10
10,1070,1130,1190
830 CALL HCHAR(K,E,32)
840 W=E+S
850 E=W
860 CALL HCHAR(K,E,96)
870 IF E>=30 THEN 1530
880 GOTO 800
890 CALL HCHAR(L,E1,32)
900 W1=E1+S
910 E1=W1
920 CALL HCHAR(L,E1,104)
930 IF E1>=30 THEN 1530
940 GOTO 800
950 CALL HCHAR(M,E2,32)
960 W2=E2+S
970 E2=W2
980 CALL HCHAR(M,E2,112)
990 IF E2>=30 THEN 1530
1000 GOTO 800
1010 CALL HCHAR(N,E3,32)
1020 W3=E3+S
1030 E3=W3
1040 CALL HCHAR(N,E3,120)
1050 IF E3>=30 THEN 1530
1060 GOTO 800
1070 CALL HCHAR(P,E4,32)
1080 W4=E4+S
1090 E4=W4
1100 CALL HCHAR(P,E4,128)
1110 IF E4>=30 THEN 1530
1120 GOTO 800
1130 CALL HCHAR(P1,E5,32)
1140 W5=E5+S
1150 E5=W5
1160 CALL HCHAR(P1,E5,136)
1170 IF E5>=30 THEN 1530
1180 GOTO 800
1190 CALL HCHAR(Q,E6,32)
1200 W6=E6+S
1210 E6=W6
1220 CALL HCHAR(Q,E6,144)
1230 IF E6>=30 THEN 1530
1240 GOTO 800
1250 PRINT
1260 PRINT
1270 O=INT(RND*9)+1
1280 O1=INT(RND*9)+1
1290 O2=INT(RND*9)+1
1300 O3=INT(RND*9)+1
1310 O4=INT(RND*9)+1
1320 O5=INT(RND*9)+1
1330 O6=INT(RND*9)+1
1340 PRINT ," ODDS"
1350 PRINT TAB(5);"HORSE 1",
O;"/1"
1360 PRINT TAB(5);"HORSE 2",
O1;"/1"
1370 PRINT TAB(5);"HORSE 3",
O2;"/1"
1380 PRINT TAB(5);"HORSE 4",
O3;"/1"
1390 PRINT TAB(5);"HORSE 5",
O4;"/1"
1400 PRINT TAB(5);"HORSE 6",
O5;"/1"
1410 PRINT TAB(5);"HORSE 7",
O6;"/1"
1420 PRINT
1430 PRINT
1440 PRINT "ENTER 0 TO STOP"
1450 PRINT "PLAYER 1, YOU HA
VE $";MB1
1460 INPUT "BET? $":B1
1470 IF B1=0 THEN 1820
1480 INPUT "WHICH HORSE?":H1
1490 PRINT "PLAYER 2, YOU HA
VE $";MB2
1500 INPUT "BET? $":B2
1510 INPUT "WHICH HORSE?":H2
1520 RETURN
1530 GOSUB 2290
1540 FOR J=1 TO 1500
1550 NEXT J
1560 CALL CLEAR
1570 IF E>=30 THEN 1830
1580 IF E1>=30 THEN 1860
1590 IF E2>=30 THEN 1900
1600 IF E3>=30 THEN 1930
1610 IF E4>=30 THEN 1970
1620 IF E5>=30 THEN 2010
1630 IF E6>=30 THEN 2050
1640 PRINT "THE WINNER IS";W
20
1650 IF H1=W20 THEN 1740
1660 PRINT "PLAYER 1 LOSES $
";B1
1670 MB1=MB1-B1
1680 IF H2=W20 THEN 1770
1690 PRINT "PLAYER 2 LOSES $
";B2
1700 MB2=MB2-B2
1710 FOR J=1 TO 1000
1720 NEXT J
1730 GOTO 190
1740 PRINT "PLAYER 1 WINS $"
;B1*O10
1750 MB1=MB1+B1*O10
1760 GOTO 1680
1770 PRINT "PLAYER 2 WINS $"
;B2*O10
1780 MB2=MB2+B2*O10
1790 FOR J=1 TO 1000
1800 NEXT J
1810 GOTO 190
1820 END
1830 W20=1
1840 O10=O
1850 GOTO 1640
1860 W20=2
1870 O10=O1
1880 GOTO 1640
1890 REM
1900 W20=3
1910 O10=O2
1920 GOTO 1640
1930 W20=4
1940 O10=O3
1950 GOTO 1640
1960 REM
1970 W20=5
1980 O10=O4
1990 GOTO 1640
2000 REM
2010 W20=6
2020 O10=O5
2030 GOTO 1640
2040 REM
2050 W20=7
2060 O10=O6
2070 GOTO 1640
2080 A=262
2090 B=330
2100 C=392
2110 CALL SOUND(200,A,0)
2120 CALL SOUND(100,A,0)
2130 CALL SOUND(200,A,0)
2140 CALL SOUND(100,A,0)
2150 CALL SOUND(200,A,0)
```

```
2160 CALL SOUND(100,B,0)
2170 CALL SOUND(200,C,0)
2180 CALL SOUND(100,B,0)
2190 CALL SOUND(200,C,0)
2200 CALL SOUND(100,B,0)
2210 CALL SOUND(200,C,0)
2220 CALL SOUND(100,B,0)
2230 CALL SOUND(900,A,0)
2240 RETURN
2250 CALL SOUND(1000,-1,0)
2260 FOR J=1 TO 200
2270 NEXT J
2280 RETURN
2290 CALL SOUND(500,1020,0)
2300 RETURN
```

»»»»»»»»»»»»»»»»»»»»»»»»»»»»»»

BASIC COMPUTERS

Ah, The Memories.

by Garry Christensen

Did you read last month's article? Here's a top up. There are 2 types os signal, digital and analogue. The analogue signal can use any voltage to represent a value where as digital uses discrete 1's and 0's. In a digital machine numbers are expressed in codes made up of the 1's and 0's and these can be transmitted using serial or parallel communication. Parellel is used in the computer and the bundles of lines that carry the value is called a bus. There are data, address and control busses in the TI computer. In general terms data moves from an input to the central processing unit and memory, and to an output. The CPU is the part of the computer that does all the work.

Got it? Great!

This month I will talk more about the CPU and memory.

Consider memory now as a whole lot of pidgeon holes. A value is stored by putting it into the pidgeon hole, just like used to happen in a post office before sorting machines but in this post office, the pidgeon holes ate large enough for one envelope only. If there is something elsé in there it is pushed out the back, never to be seen again This is called writing. Reading is a bit different. Unlike the post office, reading does not take a value out of a pidgeon hole. When reading from memory, a copy of the value is sent. The data remaines in tact and can be read again.

Each memory address, pidgeon hole, is made up of 8 bits. You will remember that 8 bits is called a byte and the biggest number that can be stored in a byte is 255. It also has an address. The address is not a lot different from a street address. It uniquely identifies each location but only numbers are used, not street names.

As you can imagine, there are quite a lot of addresses in the average block of memory. The TI can access 64K of memory (a 'K' is 1024 bytes) so that works out to 65536 addresses, numbered from 0 to 65535. Remember last month we considered how a computer used only 1's and 0's to represent numbers. The same applies here and it needs 16 bits to express the number of that size, but that can be confusing to write addresses as a 16 bit code. An easier way is used.

The 16 bits are broken up into 4 groups of 4 bits each. The largest number represented by 4 bits is 15. Here to is a problem. We can write values for 1 to 9 easily (use the decimal number), but 10 to 15 need 2 digits so letters of the alphabet are used to represent them. 'A' is 10, 'B'

is 11, 'C' is 12, up to 'F' for 15. When these 4 groups of bits are put together they from a 4 digit code made up from the numbers and the first 6 letters of the alphabet. Such a code could be 2A8C. In decimal that is 2, 10, 8, 12. To the computer it means 0010 1010 1000 1100. In this way all 64K addresses have a unique code.

The codes range from 0000 to FFFF. Those from 0000 to 1FFF are used to run the computer, from 2000 to 3FFF is for you to use, 4000 to 5FFF is for peripherals to use, 6000 to 7FFF is for modules, 8000 to 9FFF is reserved for the computer and A000 to FFFF is for the user.

As you can imagine, some of the codes for addresses will be only numbers, eg 1234. How do we know that we are talking about an address and not a number. When an address is written it is preceded by the 'greater than' sign, the above address becomes >1234. When we speak we say "hex one two three four" or sometimes "hex one thousand two hundred and thirty four". So now if someone says "hex two thousand" you know they are talking about an address in memory represented by the code >2000.

The word hex may seem a little strange to you. It has nothing to do with spells or curses (although programmers have been known to do a little of the latter) but comes from the word 'hexadecimal'. This is the name given to the numbering system based on 16 values instead of 10. I'll save that for some other time though.

Do you remember what an address bus is? That's right, it is a bundle of parallel wires or connections that carry the code for the address. Can you guess how many lines there are in the address bus for the TI? Did you say 16, because that is the number. Each line can carry 1 bit, 16 bits are needed to make an address so there must be 16 connections.

The data bus carries the information from or to the memory. 8 bits here so the data bus has ..... 8 lines. Actually there are some parts of the TI where 16 lines are used but I beg your indulgence to ignore that for the moment.

Let's look at how the memory and CPU communicate. To write data to memory, the address is placed on the address bus. The particular byte in memory is now aware that it is to be used. The value to be written is placed on the data bus then a signal on the control bus indicates that the byte is to be written into memory, erasing what was previously there. The read uses a slightly different order. First the address on the address bus, then the signal on the control bus to indicate that a read is to take place. The memory address activated will them place a copy of the value stored in it on the data bus so that the CPU can access it.

This is for byte operations, that is only one byte at a time. You may have heard it said that the TI is a 16 bit computer. That means that it can work with 16 bits (2 bytes) in a single read or write operation. While most of the memory has 8 lines in the data bus, the CPU has 16. There is a little circuitry that links these 2 parts together. The result is that the TI can access bytes only or 2 consecutive bytes at the same time. The pair of bytes is called a word.

I'm sure that you are starting to wonder how the CPU knows what to do. The answer is a list of instructions that you give it in the form of a program. The program looks nothing like BASIC because the only thing that a computer understands is numbers. The program that the CPU operates from is called machine code because it is in the form that the CPU can understand. All other programming languages have to be converted to machine code before they can be executed. Usually you are unaware of this process but it is a topic that we will look at in the future.

Within the CPU is a couple of bytes of memory called the Program Counter (PC). The PC keeps track of the address of the next instruction to be executed. When a program is running the CPU puts the value from the PC onto the address bus The word at that address is read and placed into a area called the Instruction Register (IR). Any area of memory, whether in the CPU or not, that has a special purpose is called a register.

The word in the IR is a code that tells the CPU which set of procedures to

follow. The CPU decodes the word and works out if it needs more information. Many instructions use more that 1 word so if that is the case the CPU adds 2 to the PC (2 bytes in a word) and reads that word also. This procedure continues till all of the instruction has been fetched. Finally the PC is updated again so that it points to the start of the next instruction. The sequence of events so far is called an 'instruction fetch'.

The CPU now enters the 'execution phase'. That is where it carries out the instruction. This may, for example, require reading 2 bytes or words from memory, adding them together and writing the result back to the memory again. The address of the bytes to add are in that extra part of the instruction. In this case the command to add was in the first word read in the instruction fetch and the addresses followed it. The term used to describe the extra words in an instruction is 'operand'.

We now see that a program is a list of instructions that are stored sequentially in memory that tell the CPU what to do. The address, data and control bus allow the CPU to communicate with the memory and perform operations that will result in some usable function.

What about when you first switch the computer on. There is no program there so how does it know what to do. Actually there is a program. In the area from addresses >0000 to >1FFF is some special memory called ROM. That stands for Read Only Memory. The contents of this memory cannot be changed and most importantly it is not lost when the power is disconnected. There are other forms of ROM, Erasable Programable Read Only Memory (EPROM), and Electronic Erasable Programable Read Only Memory (EEPROM). These act the same as ROM when in the computer but it is easy to change the data stored in them if you have the right equipment.

The memory that is used most, and is the one referred to when someone says 'memory' in called RAM. That means Random Access Memory. The random access part means that any byte can be used. The same applies to all of the read only memories but in RAM, the data can be changed by the computer. Changability means that programs can be loaded and calculations performed.

Don't let the names confuse you. Both ROM and RAM are random access. The difference is that ROM cannot be changed by the CPU. There is even some types of RAM that do not lose the data stored in them when the power is disconnected but it can still be changed by the CPU.

I'll leave you to digest that for the next month and next time I will talk about a special type of ROM called GROM and I will talk about how images are displayed on the screen.

By the way, if all this is new to you, I suggest that you re-read all the articles so far. You may find you will understand more after a second read.

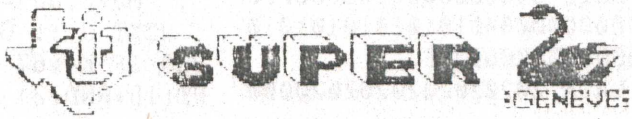
«««««««««««««««««««««««««««««««««««««««

FUNNELWEB TIP

Ever had a text file sent to you where the filename was in lower case. How do you load it into the word processor. The editor automatically converts all letters to uppercase when accepting filenames, you can't enter lower case. One way it to use Funnelweb and Diskreview to mark the file, then it's name will be there when you want to LoadFile in the editor. Here's an easier way.

Do you know about "control-." and "control-;". In the editor if you want to convert lowercase to upper, press ctrl-; (control key and the ';' key) when the cursor is sitting on the letter. It will change case. Hold the keys down and the cursor will run across the screen changing all the letters it passes. The opposite occurs with ctrl-. (control key and '.'), upper case to lower.

This also works when the editor is asking for a filename. Enter the name in upper case first, go back to the beginning (of the filename, not the devicename) and press "ctrl-.". The name changes to lower case and you will be able to load the file.



TIPS FROM THE TIGERCUB

#37

TIGERCUB SOFTWARE
156 Collingwood Ave.
Columbus, OH 43213

Here's another tune for the dulcimer player in the last Tips. Change the TO value to 94 -

```
350 DATA 9,11,13,13,13,13,13
,16,16,13,13,11,11,11,11,11,
11
360 DATA 16,18,14,21,18,18,1
6,13,9,11,9,9,9,9,9
370 DATA 21,20,18,18,16,13,1
6,16,9,11,13,11,13,14,13,13
380 DATA 21,20,18,18,16,13,1
6,16,9,13,11,9,8,6,4,4
390 DATA 9,11,13,13,13,13,13
,16,16,13,13,11,11,11,11,11
400 DATA 16,18,14,21,18,18,1
6,13,9,11,9,9,9,9,9
```

Here's one for those who like graphics, and those who make a living designing floor tiles. It borrows a bit from a Renko & Edwards program -

```
100 CALL CLEAR :: F=2 :: BC=
16 :: RANDOMIZE :: DISPLAY A
T(2,10):"ESCHER ART": :TAB(1
4);"by": :TAB(9);"Jim Peters
on"
110 DISPLAY AT(12,3):"Press
Q for new pattern":"
R to change colors":"
 C for new colors": :"
  Any key to start"
120 CALL KEY(0,K,S):: IF S=0
 THEN 120 ELSE CALL CLEAR
130 DATA 8080808080808080FF0
0000000000000010101010101010
100000000000000FF
140 DATA 2020202020202020000
0FF0000000000404040404040404
0000000000FF0000
150 DATA 1010101010101010000
000FF0000000808080808080808
00000000FF000000
160 DATA C0C0C0C0C0C0C0C0FFF
F0000000000000303030303030303
0000000000000FFFF
170 DATA F0F0F0F0F0F0F0F0FFF
FFFFF000000000F0F0F0F0F0F0F0F0
F00000000FFFFFFFF
180 DATA 8040201008040201010
2040810204080804020100804020
10102040810204080
190 DATA 101020C000000000808
0403000000000000000003040808
000000000C0201010
200 DATA FFFEFCF8F0E0C080FF7
F3F1F0F0703010103070F1F3F7FF
F80C0E0F0F8FCFEFF
210 DATA F0F0F0F0000000000F0
F0F0F0000000000000000F0F0F0
F00000000F0F0F0F0
220 DATA 80C0A090888482FFFF8
2848890A0C080FF4121110905030
101030509112141FF
230 DATA 8142241818244281814
2241818244281814224181824428
18142241818244281
240 DATA 08080808FF080808101
01010FF1010101010FF1010101
0080808FF08080808
250 DATA AA55AA55AA55AA5555A
A55AA55AA55AAAA55AA55AA55AA5
555AA55AA55AA55AA
260 DATA F0F0F0F00F0F0F0F0F0
F0F0FF0F0F0F0F0F0F0F00F0F0F0
F0F0F0F0FF0F0F0F0
270 CALL CHAR(84,RPT$("0",64
)):: FOR CH=88 TO 140 STEP 4
 :: READ CH$ :: CALL CHAR(CH
,CH$):: NEXT CH :: CALL SCRE
EN(5)
280 A=INT(6*RND+3):: H=INT(2
4/A):: HC=INT(28/A):: W=ABS(
HC/2=INT(HC/2)):: DIM M(8,8)
:: FOR P=1 TO A
290 D(P)=INT(15*RND+21)*4
300 NEXT P :: GOSUB 370
310 CALL KEY(3,K,ST):: IF K<
>81 THEN 330
320 CALL SOUND(50,500,5):: F
OR J=1 TO 4 :: FOR JJ=1 TO A
 :: M$(J,JJ)="" :: NEXT JJ :
: NEXT J :: GOTO 280
330 IF K<>67 THEN 360 :: F=I
NT(15*RND+2)
340 BC=INT(15*RND+2):: IF BC
=F THEN 340
350 FOR S=7 TO 14 :: CALL CO
LOR(S,F,BC):: NEXT S :: GOTO
 310
360 IF K<>ASC("R")THEN 310 :
: T=F :: F=BC :: BC=T :: GOT
O 350
370 ON A-2 GOSUB 380,390,400
,410,420,430 :: GOTO 520
380 RESTORE 440 :: RETURN
390 RESTORE 450 :: RETURN
400 RESTORE 460 :: RETURN
410 RESTORE 470 :: RETURN
420 RESTORE 480 :: RETURN
430 RESTORE 500 :: RETURN
440 DATA 1,2,1,2,3,2,3,1,3
450 DATA 1,2,2,1,2,3,3,2,3,4
,4,3,4,1,1,4
460 DATA 1,2,3,1,2,2,3,4,3,2
,3,4,5,4,3,4,5,1,5,4,5,1,2,1
,5
470 DATA 1,2,3,3,2,1,2,3,4,4
,3,2,3,4,5,5,4,3,4,5,6,6,5,4
,5,6,1,1,6,5,6,1,2,2,1,6
480 DATA 1,2,3,4,3,2,1,2,3,4
,5,4,3,2,3,4,5,6,5,4,3,4,5,6
,7,6,5,4
490 DATA 5,6,7,1,7,6,5,6,7,1
,2,1,7,6,7,1,2,3,2,1,7
500 DATA 1,2,3,4,4,3,2,1,2,3
,4,5,5,4,3,2,3,4,5,6,6,5,4,3
,4,5,6,7,7,6,5,4
510 DATA 5,6,7,8,8,7,6,5,6,7
,8,1,1,8,7,6,7,8,1,2,2,1,8,7
,8,1,2,3,3,2,1,8
520 FOR J=1 TO A :: FOR JJ=1
 TO A :: READ M(J,JJ):: NEXT
 JJ :: NEXT J
530 X=A+1 :: FOR J=1 TO A ::
 FOR JJ=1 TO A :: M$(1,J)=M$
(1,J)&CHR$(D(M(J,JJ)))
540 M$(2,J)=M$(2,J)&CHR$(D(M
(JJ,X-J))+1)
550 M$(3,J)=M$(3,J)&CHR$(D(M
(X-J,X-JJ))+2)
560 M$(4,J)=M$(4,J)&CHR$(D(M
(X-JJ,J))+3)
570 NEXT JJ :: NEXT J
580 CALL CLEAR :: FOR R=1 TO
 A*H STEP A :: FOR C=1 TO A*
HC STEP A
590 CALL KEY(0,K,ST):: IF K=
81 THEN 320
600 V=V+1+(V=4)*4 :: FOR T=1
 TO A :: DISPLAY AT(R-1+T,C)
:M$(V,T):: NEXT T :: NEXT C
```

```
:: V=V+W+(V=4)*4 :: NEXT R
610 RETURN
```

This routine will search a disk file for up to 10 keywords in one pass - more if you DIM K$( ) - and you may elect to find all records which contain the keyword or only those which contain it in combination with one of 1 or more secondary keywords.

```
100 CALL CLEAR
110 Y=0 :: DISPLAY AT(3,5):"
TIGERCUB KEYSEARCH" :: DISPL
AY AT(6,1):"Filename? DSK" :
: ACCEPT AT(6,14)BEEP:F$ ::
OPEN #1:"DSK"&F$,INPUT
120 DISPLAY AT(8,1):"Output
to":" (1)Screen":" (2)Printe
r":" (3)Both" :: ACCEPT AT(8
,11)VALIDATE("123")SIZE(1)BE
EP:Q
130 IF Q>1 THEN DISPLAY AT(1
3,1):"Printer name?" :: ACCE
PT AT(13,15):P$ :: OPEN #2:P
$
140 DISPLAY AT(15,1):"Search
 for":" (1)First match":" (2
)All matches" :: ACCEPT AT(1
5,13)VALIDATE("12")SIZE(1)BE
EP:S
150 DISPLAY AT(12,1)ERASE AL
L:"Press ENTER when all key-
":"words have been entered."
160 DISPLAY AT(17,1):"Press
ENTER if none -"
170 Y=Y+1 :: DISPLAY AT(15,1
):"Keyword? ";CHR$(127):: AC
CEPT AT(15,10)SIZE(-28)BEEP:
K$(Y):: IF K$(Y)=CHR$(127)TH
EN 190
180 W=W+1 :: DISPLAY AT(19,1
):"With? ";CHR$(127):: ACCEP
T AT(19,7)SIZE(-21)BEEP:W$(Y
,W):: IF W$(Y,W)=CHR$(127)TH
EN W=0 :: GOTO 170 ELSE GOTO
 180
190 Y=Y-1
200 LINPUT #1:M$
210 FOR J=1 TO Y :: IF POS(M
$,K$(J),1)=0 THEN 290
220 IF W$(J,1)=CHR$(127)THEN
 250
230 W=W+1 :: IF W$(J,W)=CHR$
(127)THEN W=0 :: GOTO 290
240 IF POS(M$,W$(J,W),1)=0 T
HEN 230
250 IF Q>1 THEN PRINT #2:M$
260 IF Q<>2 THEN PRINT M$
270 IF S=1 THEN 310
280 IF W$(J,W)<>CHR$(127)THE
N 230
290 NEXT J
300 IF EOF(1)<>1 THEN 200
310 CLOSE #1 :: DISPLAY AT(2
4,1):"FINISHED - PRESS ANY K
EY" :: CALL SOUND(200,500,5)
320 CALL KEY(0,K,ST):: IF ST
=0 THEN 320 ELSE CALL CLEAR
:: GOTO 110
```

You can set up a keyfile in TI-Writer - just remember that each 80-character line is a separate record, and keep the Alpha Lock down!

However, this is the program that I plan to use to set up a keyfile index of all the newsletters you have sent me, if I ever find the time -

```
100 DISPLAY AT(3,10)ERASE AL
L:"TIGERCUB": :"   KEYWORD I
NDEX WRITER" !by Jim Peterso
n
110 DISPLAY AT(8,1):"Filenam
e? DSK" :: ACCEPT AT(8,14):F
$ :: OPEN #1:"DSK"&F$,APPEND
 :: CALL KEY(3,K,S)
120 P$="*****" :: Y=00 :: M$
="**" :: P=00
130 DISPLAY AT(12,1):"NEWSLE
TTER? ":P$ :: ACCEPT AT(13,1
)SIZE(-28):P$ :: IF SEG$(P$,
1,3)="END" THEN CLOSE #1 ::
STOP
140 DISPLAY AT(14,1):"YEAR?"
;Y :: ACCEPT AT(14,7)VALIDAT
E(DIGIT)SIZE(-4):Y
150 DISPLAY AT(14,13):"MONTH
? "&M$ :: ACCEPT AT(14,20)SI
ZE(-9):M$
160 DISPLAY AT(16,1):"PAGE?"
;P :: ACCEPT AT(16,7)VALIDAT
E(DIGIT)SIZE(-3):P
170 DISPLAY AT(18,1):"ARTICL
E? " :: ACCEPT AT(19,1):A$
180 DISPLAY AT(20,1):"AUTHOR
?" :: ACCEPT AT(21,1):AU$
190 DISPLAY AT(22,1):"KEYWOR
DS?" :: ACCEPT AT(23,1):K$
200 PRINT #1:P$&" "&STR$(Y)&
" "&M$&" "&STR$(P)&" "&A$&"
"&AU$&" "&K$
210 GOTO 130
```

Here's one to have fun with, from an ingenious German programmer. I just couldn't resist adding a tuba to his band.

```
100 !BY TORSTEN NIEMIETZ, MA
RBACHER WEG 3,D-2800 BREMEN
1,WEST GERMANY
110 FOR J=1 TO 10 :: READ T(
J)
120 NEXT J :: E=330 :: A=440
 :: H=494 :: C=554 :: K=659
:: F=740 :: G=831
130 DISPLAY AT(3,8)ERASE ALL
:"S - O - L - O": :TAB
(10);"MIT OOMPAH": :RPT$("="
,28): : :"BY":" TORSTEN NIEM
IETZ": :"mit Oompah by Tiger
cub"
140 DISPLAY AT(18,1):"MAKE U
P YOUR SOLO WITH":"KEYS 1 TO
 9 ... COME ON !!!"
150 FOR S=1 TO 2 :: CALL SOU
ND(200,E,3,H,3):: CALL SOUND
(200,E,3,H,3)
160 CALL SOUND(200,E,3,C,3):
: CALL SOUND(200,E,3,H,3)::
NEXT S
170 M=E :: N=H :: O=C :: D=8
 :: GOSUB 210 :: M=A :: N=K
:: O=F :: D=4 :: GOSUB 210 :
: M=E :: N=H :: O=C :: GOSUB
 210 :: M=H :: N=F :: O=G ::
 D=2
180 GOSUB 210 :: M=A :: N=K
:: O=F :: GOSUB 210 :: M=E :
: N=H :: O=C :: GOSUB 210 ::
 M=H :: N=F :: O=G :: GOSUB
210
190 FOR X=10 TO 3 STEP -1 ::
 CALL SOUND(200,E,3,H,3,T(X)
,0)
200 NEXT X :: CALL SOUND(800
,E,3,H,3,K,0):: GOTO 150
210 FOR X=1 TO D :: FOR Y=1
TO 2 :: GOSUB 280
220 CALL SOUND(200,M,3,N,3,T
(R-48-(R=48))*.9375,30,-4,0)
230 NEXT Y :: GOSUB 280
240 CALL SOUND(200,M,3,O,3,T
(R-48-(R=48))*.9375,30,-4,0)
:: GOSUB 280
250 CALL SOUND(200,M,3,N,3,T
(R-48-(R=48))*.9375,30,-4,0)
260 NEXT X :: RETURN
270 DATA 587,659,784,880,988
```

```
,1175,1319,1568,1760,44733
280 CALL KEY(0,R,S):: IF S<>
0 AND R>48 AND R<58 THEN RET
URN ELSE R=57 :: RETURN

1 !ONE-LINER universal calen
dar for day of week of any d
ate since 1905 - by Dennis H
odgson in Sydney News Digest
2 !input day, month, year as
 for instance 30,4,1986
100 A=1 :: INPUT D,M,Y :: FO
R T=A TO M-A :: H=H+29+VAL(S
EG$("20212122121",T,A)):: NE
XT T :: J=H+(Y/4<>INT(Y/4)AN
D M>2)+INT((Y-A)*365.25)+D :
: PRINT SEG$("SASUMOTUWETHFR
",(J-INT(J/7)*7)*2+A,2):: RU
N
```

Yes, there are legitimate uses for GRAM copiers and track copiers and such - but there is no way to get these utilities into the hands of the few who will only use them honestly, without also getting them into the hands of the many who will use them as burglar tools. And so, a few more nails are driven into the coffin...

MEMORY FULL

Jim Peterson

SHOP

All the software listed below is in stock or will be very shortly. Prices are in Australian dollars and include postage. The recent drop in the value of the Aussie dollar has resulted in a small increase in prices.

TI Image Maker - Use an 80, hi-resolution monitor with your TI. Special price $165.

Rock Runner	$14.00
Waterworks	$14.00
Beyond Video Chess	$11.50
Rattlesnake Bend	$ 7.75
Castle Darkholm	$ 9.75
Doom Games I	$ 7.75
Doom Games II	$ 7.75
Doom Games III	$ 7.75
Page Pro	$27.00
Page Pro Pics #1	$ 8.50
Page Pro Pics #2	$ 8.50
Page Pro Pics #3	$ 8.50
Page Pro Pics #4	$ 8.50
Page Pro Pics #5	$ 8.50
Page Pro Pics #6	$ 8.50
Page Pro Pics #10	$ 8.50
Page Pro Pics #14	$ 8.50
Page Pro Pics #15	$ 8.50
Page Pro Borders #1	$ 8.50
Page Pro Borders #2	$ 8.50
Page Pro Fonts #1	$ 8.50
Page Pro Fonts #2	$ 8.50
Page Pro FX	$16.50
Page Pro Headline Maker	$11.50
Page Pro Headline Fonts #1	$ 8.50
Page Pro Headline Fonts #2	$ 8.50
Page Pro Hradline Fonts #3	$ 8.50
Page Pro Templates #5	$ 7.70
Page Pro Templates #6	$ 7.50
Page Pro Templates #7	$ 7.50
Pix Pro	$16.00
Spell It	$21.00
Quick Run	$11.00
Tournament Solitaire	$16.50

ROCK RUNNER:

-Eric LaFortune, a Belgian teenager, has created a 99/4A version of Boulder Dash, a favorite on Commodore computers, and maybe other computers too. It runs out of the Editor/Assembler module only right now and provides an excellent emulation of the Boulder Dash program. Both of my kids gave it a thumbs up, for whatever that's worth, although they complained about having to go back to the start of the program when they got zonked by the rocks. Looks like a neat piece of software, that apparently takes advantage of some here-to-fore unused area of the TMS9918A's graphics capabilities. Chris Bobbitt calls it the "half-bitmap" mode. The program is available for $12.95 plus $1.50 shipping and handling from;

ASGARD SOFTWARE
P.O. BOX 10306
ROCKVILLE, MD. 20849
703-255-3085

Courtesy of LA99ers

IN THE P.O. BOX

Newsdigest (Sydney), January/Febuary: Co-ordinator's report, Editors Comment, Secretaries Notebook, Assembly Class, PE Box Extention reviewed by Ben Takach, Shop, Graph Paper, Software Column, Techo Time by Geoff Trott, World News, GIFviewer and the Hard Disk by Geoff Trott, Printing Line Numbers by Bob Relyea , Treasurer's report, Hints, Tips and Answers, Jenny's Younger Set, TI-Bit #13, Control of CS1, Hollywood Hyjinx, Index for 1991, Printer Graphics, Sorting, Fractal Graphics with TML, Publications Index, XB Tips #14, Link-It #21, Screen Dump Routines, Beginning Forth, Newsletter Update, Number Base Converter.

KC99'er (Kansas City), Oct/Nov/Dec 1991: World News, Kinder Korner, Asgard News, Bloodbank, You Can't Do This, Formatter Tip, Spritely Sprites (by our own Col), ANSI Programs.

Wordplay (Portland), January 1992: From The President, Minutes, TI's Birthday, Editorial, Compare Health Insurance, Heartbeat Terminology, Why Should You Learn To Program.

TopIcs (LA99ers), January: XB Miscellany, review of Art Printshop, Index for 1991, Editor's column, Using Transliterates.

LA 99ers, February 1991: Ramblin' Thoughts of the President, XB Miscellany by Earl Raguse, TI-Base Tutorial by Martin Smoley, Programmer's Dilemma, How to do a Person to Person Download, What the Heck is a Boolean by Val Mehline, Archiving - a Headache by Andy Frueh.

ATTIC (Adelaide), February 1992: Words from the Co-Ordinator, Editorial, Puzzler, The Witching Hour, Programming Tips, Double Column Text Formatter, Kid's Stuff, Ontplopper, XB to TIA Instance, XB Tips, The Clock, Use a 40 Column Display with TI-W, Dips Tips by David Perkovic, TI Odities.

LETTERS:

Jim Peterson: Asking if the group is still going so he can include us in a listing, Clearinghouse BBS.

Bob Bishop (Melbourne User Group): Melbourne group still going, info about disk of adventure programs, software order. (Thanks for the letter Bob. I'll write as soon as I have finallised this newsletter - ED)

Ian Bollen: Enquiry about the group. Contact via Electronic Australia article.

0000000000000000000000000000000000000

Sorting, part 2

by Ron Brubaker, USA Courtesy of TIsHUG

The following is a modified version of the program segment given in last month's article. Two variables C and S have been added to permit measurement of the efficiency of the routine. These are initialised in lines 155 and 156. The counter in line 175 counts the number of comparisons that are made and the counter in line 185 counts the number of swaps that have been performed. (EDITOR'S NOTE - all of the programs in this series have been Listed to disk from working programs so there should be no errors.)

```
10 REM **** GENERATION OF A LIST OF RANDOM NUMBERS ****
20 REM
30 DIM A(100)
40 RANDOMIZE
50 PRINT "HOW MANY NUMBERS DO YOU WANT";
60 INPUT N
70 PRINT
80 FOR I=1 TO N
90 A(I)=INT(RND*100)+1
100 PRINT A(I);
110 NEXT I
120 PRINT
130 REM
140 REM *** SIMPLE BRUTE FORCE SORT - WITH COUNTERS ***
150 REM
155 C=0
156 S=0
160 FOR J=1 TO N-1
170 FOR I=1 TO N-1
175 C=C+1
180 IF A(I)<=A(I+1)THEN 220
185 S=S+1
190 T=A(I)
200 A(I)=A(I+1)
210 A(I+1)=T
220 NEXT I
230 NEXT J
240 REM
250 REM **** ROUTINE TO PRINT SORTED LIST OF NUMBERS ****
260 REM
270 PRINT
280 FOR I=1 TO N
290 PRINT A(I);
300 NEXT I
310 PRINT
315 PRINT
320 PRINT C;"COMPARISON'S WERE MADE"
330 PRINT
335 PRINT S;"SWAPS WERE MADE"
```

The resulting output is the following:

```
 HOW MANY NUMBERS DO YOU WANT? 15

64 48 2 79 36 5 66 71 100 24 14 67 57 1 3 34

And finally,

2 5 13 14 24 34 36 48 57 64 66 67 71 79  100

196 COMPARISONS WERE MADE

56 SWAPS WERE MADE
```

An additional routine has been added that will print the sorted list and the results accumulated by the counters.

Note that the number of comparisons is what would be predicted from the loop structure of the program (i.e. 196 is N-1 squared).

From the loops shown above it would seem unnecessary to extend the inner loop to N-1 on every pass since the largest number falls immediately to the end of the first pass and on each succeeding pass an additional number falls into position near the end of the list. Thus, the inner loop can terminate after one less step on each pass. This can be accomplished very easily by changing line 170 to terminate the inner loop at N-J instead of N-1. This change and the resulting output is shown below.

```
10 REM **** GENERATION OF A LIST OF RANDOM NUMBERS ****
20 REM
30 DIM A(100)
40 RANDOMIZE
50 PRINT "HOW MANY NUMBERS DO YOU WANT";
60 INPUT N
70 PRINT
80 FOR I=1 TO N
90 A(I)=INT(RND*100)+1
100 PRINT A(I);
110 NEXT I
120 PRINT
130 REM
140 REM ***** SIMPLE BUBBLE SORT *****
150 REM
155 C=0
156 S=0
160 FOR J=1 TO N-1
170 FOR I=1 TO N-J
175 C=C+1
180 IF A(I)<=A(I+1)THEN 220
185 S=S+1
190 T=A(I)
200 A(I)=A(I+1)
210 A(I+1)=T
220 NEXT I
230 NEXT J
240 REM
250 REM ***** ROUTINE TO PRINT LIST OF NUMBERS *****
260 REM
270 PRINT
280 FOR I=1 TO N
290 PRINT A(I);
300 NEXT I
310 PRINT
315 PRINT
320 PRINT C;"COMPARISONS WERE MADE"
330 PRINT
335 PRINT S;"SWAPS WERE MADE"
```

In this case the output is:

```
HOW MANY NUMBERS DO YOU WANT? 15

64 48 2 79 36 5 66 71 100 24 14 67 57 1 3 34

2 13 14 24 34 36 48 57 64 66 67 71 79 1 00

105 COMPARISONS WERE MADE

56 SWAPS WERE MADE
```

Note that the number of comparisons has decreased to 105 but that the number of swaps is still 56 and the output is still sorted. It is reasonable to assume that the revised program runs nearly twice as fast as the original.

It was also noted above that the sort was completed after less than N-1 passes. However, it cannot be assumed that all combinations of fifteen random numbers will be sorted in the same number of passes that the test data shown in these examples required. The safest way to determine if the sort has been completed is to check to see if any swaps have been performed on the last pass. The following version of the sort routine has been modified by replacing the outer FOR/NEXT loop with a looping structure that loops until no further swaps have been made on the last pass through the inner loop.

```
10 REM **** GENERATION OF A LIST OF RANDOM NUMBERS ****
20 REM
30 DIM A(100)
40 RANDOMIZE
50 PRINT "HOW MANY NUMBERS DO YOU WANT";
60 INPUT N
70 PRINT
80 FOR I=1 TO N
90 A(I)=INT(RND*100)+1
100 PRINT A(I);
110 NEXT I
120 PRINT
130 REM
140 REM ***** SIMPLE BUBBLE SORT *****
150 REM
155 C=0
156 S=0
160 J=1
165 Q=0
170 FOR I=1 TO N-J
175 C=C+1
180 IF A(I)<=A(I+1)THEN 220
185 S=S+1
186 Q=1
190 T=A(I)
200 A(I)=A(I+1)
210 A(I+1)=T
220 NEXT I
230 IF Q=0 THEN 250
235 J=J+1
240 REM
250 REM ***** ROUTINE TO PRINT LIST OF NUMBERS *****
260 REM
270 PRINT
280 FOR I=1 TO N
290 PRINT A(I);
300 NEXT I
310 PRINT
315 PRINT
320 PRINT C;"COMPARISONS WERE MADE"
330 PRINT
335 PRINT S;"SWAPS WERE MADE"
```

The variable Q is simply a flag that is set to zero each time the inner loop is initiated and is set to 1 if a swap is made. Thus, if the inner loop is exited with Q still having a value of zero the sort is complete.

The following results were obtained with the above version of the sorting routine.

```
HOW MANY NUMBERS DO YOU WANT? 15

64 48 2 79 36 5 66 71 100 24 14 67 57 1 3 34

2 5 13 14 24 34 36 48 57 64 66 67 71 79  100

102 COMPARISONS WERE MADE

56 SWAPS WERE MADE
```

The number of comparisons decreased slightly indicating that some savings were realised by this modification. Although this program is not likely to set any speed records it is now quite efficient and can be quite useful.

Next month we will look at the modifications necessary to use this program to sort string variables as well as look at the concept of using a pointer. o

Air Taxi by Don Shorock

reviewed by Jim Peterson, Ohio, USA

I have always wished that there were more educational programs, above the 2+2=? level, for our computer. And I have always thought that the best educational programs were those that took advantage of computer capabilities to entertain while teaching.

Also, I have always much preferred games that require me to exercise my mind, rather than depending on quick reaction or blind guessing. And, being a programmer, I admire efficient, memory-saving programming.

All that is why I was so very impressed by the new game, Air Taxi, recently released by Don Shorock. It is uniquely educational, very entertaining, and so compactly programmed that the basic version is available on cassette!

The game can be played alone, as it usually will be, or by up to 8 players. Don customises each game with the default names of whatever number of players you choose and with your home town as the starting point. Each player may select his own handicap level, ranging from A to Z for 6 to 81 cities, and his skill level ranging from 1 to 9 which determines the target size.

A black silhouette map of the entire United States and southern Canada is then displayed; the only features are the Great Lakes, Great Salt Lake, and the coast lines. You are randomly offered a destination to fly to. Since all your friends bum rides from you, and TI users are cheapskates (that is my comment, not Don's!), you are not even paid for your gas for this first trip. It may therefore pay you to refuse any offer to a distant destination - however, each refusal costs you $2.00.

When you accept an offer, you then use the S and D keys to set your initial flight direction, in 45 degree increments (i.e., north, northeast, east, etc.) and press Q. You hear the sound of the motor revving up, and a small cursor dot begins moving from your town in the direction you selected, while your gas gauge shows your fuel being used up. You can use the S and D keys to change direction. If you get close enough (depending on the skill level you selected) before your fuel runs out, the cursor will stop, the motor revs down, and you will be shown the cost of the fuel expended and your remaining bank balance. If your fuel runs out to soon, you will glide to the nearest airport and you must then set your direction from that point and try to reach your original destination. However, if you were too far from any airport when your gas tank ran dry, you will be returned to your home town and will be assessed repair costs.

Once you have reached your first destination and said goodbye to your freeloading friends, you will then be randomly offered fares, at prices depending on distance, from that point to another city. You have the option to refuse offers, at a cost of $2.00. If you can fly to that point with a minimum of maneuvering, the fare will more than cover the cost of fuel, and you will make money - plus an occasional tip.

There are too many other features to describe here. The program comes with four pages of printed documentation, and the disk version includes three additional files, which can be merged in, to add many more cities or to convert the program for use with a joystick. At the handicap and skill level K 7 which Don set for me as defaults, I found that I was able to stay ahead of the game by refusing most fares except coastal cities and then cruising along the coast until the airport radar picked me up and brought me in. Trying to find Kansas City or Cheyenne on that black silhouette map would be very difficult without consulting a regular map - and in doing so, you would learn a great deal about the relative location of cities.

This is a commercial program, not fairware, and it is customised for each purchaser. The price is $15 for the disk version, $20 for the cassette version. To get an order form, on which you can specify your own default options, write to Don Shorock, P.O. Box 501, Great Bend KS 67530. o

Organize

It has happened to all of us. You have a program or some data on a disk and something happens to corrupt it. This article will deal on good habits in your work area to avoid that situation as much as possible. We won't discuss disk recovery here. Much has been written on that subject and procedures are available to help you recover data that has been lost.

If your computer area is littered with disks everywhere, some identified and others just lying there waiting to be checked out then read on. If you have a lot of programs, data and correspondance on disks, a good idea would be to get rid of that old SSSD (single side-single density) disk drive and invest in a DSDD (double side-double density) drive. With your new drive you'll still be able to address your old diskettes.

Start out by formatting a dozen or so disks at DSDD with 40 tracks. Make sure you have a good back-up of your disk manager in case something should happen to it. Power fluctuation can cause many problems and another good investment is a surge and spike protector. Make sure your source of power is not shared by a refrigerator or other appliance which can cause drops as they turn on.

Organize your programs and data into categories and put like data on the same disks for instance (1) Extended Basic programs, (2) Basic programs, (3) Logo Files, (4) Word Processing files, and other areas you might have. Make back-up copies of everything that is valuble to you and store in an area remote from your computer room. Once you have a program or file in finished form be sure to use the protect feature of your disk manager so you won't overwrite anything. Using a protect tab on the disk itself will ensure nothing gets accidently deleted.

While you're working save your program or data from time to time. Nothing can be more frustrating that having to re-type in 40 or 50 lines if the keyboard locks up or a power glitch destroys data. Be sure to store your disk in its protective envelope and get a disk holder box to store them in.

Eating, drinking and smoking in your computer area can cause problems. Crumbs falling into the computer or printer can cause any number of problems and drinks spilled into electronic parts may require expensive repairs. Smoking in your computer area can leave deposits on delicate parts that could interfere with performance. Many companies have strict rules that forbid certain activities for their computer areas.

When not in use keep your keyboard and printer covered so loose objects will not fall in. A foreign object in the printer mechanism could result in costly repairs. Good house keeping in your computer room pays dividends. You'll know more readily where things are and your equipment will be kept in better shape.

by CHUCK BALL

PUNN USERS NINETY NINES

BASIC

Using string functions

By REGENA

Most computing done on the TI99/4A is with numbers. However, some information can be treated as *strings*, or groups of characters that are not necessarily numbers. Since we use a lot of names or words other than numbers in everyday life, we need to be able to use strings on the computer.

One way to signal to the computer that you are using a string is to enclose characters within quotation marks. PRINT 3+5 will print the number 8, but PRINT "3+5" will print exactly what is in the quotation marks, 3+5. To use a string variable, end the variable name with the dollar sign, such as A$ or NAME$.

String expressions may contain letters, numbers and symbols, and they may be up to 255 characters long. Longer strings are truncated on the right.

Strings are combined in TI BASIC by using the ampersand, such as A$&B$ or "HELLO "&NAME$. Several functions available in TI BASIC are specifically for strings. Any function that ends with a dollar sign gives a string as a result. Some functions use strings in the argument but give a numeric result. You cannot combine string and numeric expressions.

This first sample program, STRINGS1, defines the string variable A$ as "HI" and the string variable B$ as "CINDY". Line 140 prints the two variables separated by a semicolon. Notice that the semicolon indicates the next item to be printed follows the first item immediately with no spaces. Line 150 inserts a space between the two strings. Line 160 illustrates a more grammatically correct combination of the words by inserting a comma and a space between A$ and B$. Line 170 prints A$,B$. A$ is printed, then the comma puts B$ in the next print column. Line 180 prints A$ then the colon says to go to the next line before printing B$.

```
100 REM STRINGS1
110 CALL CLEAR
120 A$="HI"
130 B$="CINDY"
140 PRINT A$;B$
150 PRINT A$;" ";B$
160 PRINT A$;", ";B$
170 PRINT A$,B$
180 PRINT A$:B$
190 PRINT
200 END
```

LEN(x$) is a string function which gives the length of the string x$, or the number of characters contained in x$. In TI BASIC you may have a null string " "; the length of a null string is zero. Leading and trailing blank spaces are counted in the number of characters for the length. In the following example, Line 150 calculates the length of the string variable A$ and assigns it to the numeric variable L. Line 160 prints L.

SEG(x$,n$,n2) is the SEGment function and is comparable to LEFT$, MID$ and RIGHT$ of other versions of BASIC. SEG$(x$,n$,n2) will return the segment of string x$ starting with the character in the nl position and continuing until the segment is n2 characters long. In the following example, Line 130 prints the segment of A$ starting with the first character and containing 5 characters. Line 140 prints the segment of A$ starting with the 7th character and containing 4 characters.

POS(sl$,s2$,n) is the POSition function. sl$ and s2$ are string expressions. The numeric expression n is evaluated and rounded to an integer. POS finds the first occurrence of s2$ within sl$, starting at character n. The value returned is the character position of the first character of s2$ in sl$. If s2$ is not found, a value of zero is returned. In the following example program, Line 170 assigns P the value of the position of the space, " ", in the string A$, starting with the first character. Line 180 prints what position that is. Lines 190 and 200 then print segments determined by that position P.

```
100 REM STRINGS2
110 A$="BRETT LYNN"
120 PRINT A$
130 PRINT SEG$(A$,1,5)
140 PRINT SEG$(A$,7,4)
150 L=LEN(A$)
160 PRINT "LEN(A$) =";L
170 P=POS(A$," ",1)
180 PRINT "POS =";P
190 PRINT SEG$(A$,1,P-1)
200 PRINT SEG$(A$,P+1,L-P)
210 PRINT
220 END
```

The third example program, STRING3, illustrates the functions ASC and CHR$. ASC (x$) returns the ASCII value of the first character of the string x$. Line 130 prints the ASC(A$), which will be the ASCII value of the first character in A$. CHR$(n) prints the character corresponding to the ASCII number n. Lines 150-170 print a number J, then the CHR$(J) or the character corresponding to that ASCII number.

```
100 REM STRING3
110 A$="RICHARD"
120 PRINT A$
130 PRINT "ASC(A$) =";ASC(A$
)
140 PRINT
150 FOR J=65 TO 70
160 PRINT J;CHR$(J)
170 NEXT J
180 END
```

I have published this subroutine before, but it fits here with the discussion of strings. If you want to print a message on the screen without scrolling, or if you want to print a message at a certain position on the screen, use this subroutine. Put the message in M$, and specify the ROW and COLumn. Lines 300-330 are the subroutine that use CALL HCHAR to place the message on the screen a character at a time. First the segment SEG$ of the message M$ is taken one character at a time, and the ASCII code of that character is needed for the CALL HCHAR command. The process is repeated for the length LEN of the message. Two example messages are printed.

```
100 REM MESSAGE
110 CALL CLEAR
120 M$="PRINTING . . ."
130 ROW=10
140 COL=5
150 GOSUB 300
160 M$="EXAMPLE"
170 ROW=15
180 COL=17
190 GOSUB 300
```

REGENA ON BASIC —

(Continued from Page 9)

```
200 STOP
300 FOR C=1 TO LEN(M$)
310 CALL HCHAR(ROW,COL+C,ASC
(SEG$(M$,C,1)))
320 NEXT C
330 RETURN
340 END
```

The following MONTHS program illustrates a way to correlate the names of the months with the month numbers. One way to program using months is to have an array of 12 elements, such as M$(1)="JAN", M$(2)="FEB", etc. Another way to program is to use strings.

```
100 REM MONTHS
110 CALL CLEAR
120 M$="JANFEBMARAPRMAYJUNJU
LAUGSEPOCTNOVDEC"
130 PRINT "THE MONTHS ARE"
140 FOR M=1 TO 12
150 PRINT M,SEG$(M$,M*3-2,3)
160 NEXT M
170 PRINT
180 RANDOMIZE
190 M=INT(12*RND)+1
200 PRINT "MONTH";M;"IS ";SE
G$(M$,M*3-2,3)
210 PRINT
220 A$="MAY"
230 PRINT A$;" IS MONTH";INT
((POS(M$,A$,1)+3)/3)
240 END
```

The string M$ contains the three-letter month names all combined into one string. Lines 130-160 print the 12 months in order by using the SEG$ function to pick out three letters at a time. If you still wanted to use an array, you could use Line 150 to define M$(M)=SEG$(M$,M*3-2,3). Thus lines 140-160 would define all 12 months rather than using 12 individual statements or a DATA-READ system.

Lines 180-200 illustrate how you would determine the month name later in the program using the string method if you had a month number. Lines 220-230 illustrate how you would determine the month number if you know the month name.

I have one more sample program illustrating the use of strings. This example was sent to me by Stephen Shaw of Stockport, Cheshire, England, as a recommendation to speed up the shuffling of cards in card games such as Pyramid Solitaire (MICROpendium, April 1990). My method took 18 to 30 seconds, usually about 22 seconds, from the time you press Enter to when the first card starts drawing. Using his method, shuffling took 19 seconds (constant).

Let me just mention our main discovery. The program can be run as is in TI Extended BASIC because I don't use graphics characters in sets 15 and 16. However, the shuffling time (my method) took 40 seconds the first time I timed it and 1 minute 37 seconds the second time. Of course, that's long enough never to run the program again! And long enough for Mr. Shaw to write to me. Here's a case where TI BASIC was quicker than Extended BASIC.

I have used several different methods of card shuffling in my past programs — choosing a random number from 1 to 52 and translating to a number and suit, or choosing a random number from 1 to 13 for the number and then from 1 to 4 for the suit, and making sure the card hasn't been chosen before. This method using strings is worth trying.

```
100 REM SHUFFLE
110 CALL CLEAR
120 DIM CARD(52,2)
130 C$=""
140 FOR N=1 TO 52
150 C$=C$&CHR$(N)
160 NEXT N
170 RANDOMIZE
180 FOR N=1 TO 52
190 CD=INT(RND*LEN(C$)+1)
200 @=ASC(SEG$(C$,CD,1))
210 C$=SEG$(C$,1,CD-1)&SEG$(
C$,CD+1,52)
220 SU=INT((@-1)/13+1)
230 NU=@-(SU-1)*13
240 CARD(N,1)=NU
250 CARD(N,2)=SU
260 PRINT STR$(NU)&" "&STR$(
SU)&"; ";
270 NEXT N
280 END
```

Lines 130-160 initially define a string variable C$ of 52 different characters representing the 52 cards. Line 170 randomizes the selection. Lines 180-270 shuffle the cards. Line 190 chooses a random number CD. Line 200 selects the character in the CD position and finds out the ASCII number of that character. Line 210 then creates a new C$ string deleting that character. Line 220 determines the suit SU of the card and Line 230 determines the number NU of the card depending on the ASCII number. For purposes of illustration in this example, we put the number in CARD(N,1) and the suit in CARD(N,2) and print out the card number, then suit, in Line 260. In your own program you would "draw" the card or save CARD(N,1) and CARD(N,2) for later use.

Notice that the next time you "deal" a card, Line 190 chooses a random number CD which can be 1 to the length of C$ (which decreases by 1 each time you deal). Line 200 determines which card it is, and Line 210 "squeezes" C$ to eliminate that card (character) from being chosen again.

You can use this method of selection for random numbers other than for cards. The cards have extra calculations because of the four suits available. This method would be useful for any selection in which once an object is chosen it cannot be used again.

Just one more note this month. You may have noticed an error in the program listing for Playing Notes in the September 1991 issue. The listing is correct to Line 1360, then Jerry Stern's program and mine get mixed up. Line 1370 is at the bottom of page 13, and then Lines 1380 to 1440 are on page 14. The line right after my Line 1360 is a continuation of his program Line 810 on page 13. I might mention that I believe this is only the second time in 10 years one of my published programs has had a printing error. Really, it's all there, you've just got to find it! Best wishes for another month.

Working With Numbers Joe Nollan

There are a number of statements in BASIC that help us deal with numbers. The math functions are pretty much straight forward and covered well in the User's Guide. There are however, problems which are not algebraic, but more in the nature of house keeping. The first number problem that I encountered was after dividing $17.00 between three people and finding an answer with eight decimal places. This is a common problem when dealing with numbers. The solution is fairly simple. Consider the number 5.66666667. To reduce this to two decimal places so that it can more easily represent dollars and cents, first multiply it by 100 to yield 566.666667. The next step makes use of the INT function which will eliminate the fractional portion of the number giving us 566. This can now be divided by 100 to yield 5.66. Although this process was explained in a step by step manner it can often be done in a single statement like X=(INT(X0))/100.

Another common problem with numbers is rounding off. If you have a number like 5.98 and would like it rounded off to the nearest dollar the following method can be used. First add .5 to the value and then use the INT function. In the example of 5.98, adding .5 to it will yield 6.48 and the INT function will reduce it to 6 even. If the original value was 5.14, adding .5 would yield 5.64 and the result of the INT function would thus result in 5 even. The BASIC statement would look like this: X=INT(X+.5)

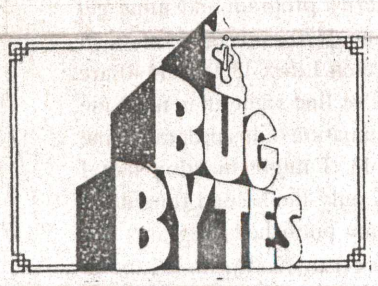
There are a couple of other statements that are invaluable where numbers are involved. The STR$(N) statement will convert the numeric variable N into a string variable. This can be used when printing the value in a statement. When printed as a string, you won't have an extra space ahead of the number The inverse of the function is the VAL(N$) statement. This statement will convert a string value in a numeric expression. You can not multiply a string value by two.

A big word with numbers is concatenation. This shows it's ugly head when you print out a column of numbers and the trailing zeros are left off making your column look like it was done by a five year old. The problem can be solved by converting the numbers to string variables and printing them as such. The trailing zero problem can also be solved by adding a small amount to the number. Working with dollars and cents for example, you may have a number like $9.50 and that would look like $9.5 when printed. Let's start with a numeric variable X and use it to represent the cost of an item (dollars-/cents). The $9.50 will be used as the value for this example. First we add a small amount to the value which will not affect the original number. In this case we are dealing with two decimal places so we will add a three decimal place number to it. X+.001 will do it. Before this figure is printed it is first converted to a string, then printed without the last character. First use X$=STR$(X) to convert it to a string variable. Now we will use the SEG$ function to drop the last character. The SEG$ function requires a string (X$), a starting value (1), and how many characters (LEN(X$-1)). The BASIC statement would be: X$=SEG$(X$,1,LEN(X$-1)). With these values the SEG$ function will take all of the X$ starting with the first character and include everything but the last character. Our new string will have our 9.50. This value can be printed in a sentence with a statement like PRINT "THE ITEM COST IS $";Z$;" WHOLESALE.".

Printing a column of numbers presents another problem. Tab settings will line up the LEFT digits which is great if the values all have the same number of digits and awful if they don't. One way to solve this problem is to use the LEN(X$) statement to determine the TAB(Z) value. Look at this example: PRINT "ITEM COST";TAB(15-LEN(X$));X$. In this example the larger the number, the smaller the TAB will be and the column will be lined up on the right.

(EDITOR NOTE: See the article "Image Statements", in the August 1989 issue of WordPlay.)

The above methods are not the only way to solve number problems in BASIC. XBASIC has routines around them as well. This article is intended to give you some ideas to help you solve your specific problem. Study up on the functions mentioned here to gain more control with your numbers.



TI BRISBANE USER GROUP
P.O. BOX 3051
CLONTARF M.D.C.
QLD AUST 4019.

43
GRAEME MORRIS
P.O. BOX 371
CLEVELAND QLD 4163