

ISSUE #11 NOVEMBER 1988

FOR THE RECORD

by Ed Bittner Recording Secretary

The October meeting of the West Penn 99'ers was as crisp as the brisk autumn air. Presidential hopeful ,VP Mickey Schmidt, opened the meeting at 7:10 in Scott Colemans absence. The immediate order of business was the solicitation of nominations from the floor for the upcoming elections. No additional names were added thus the slate remains the same as published last month. With no library report and the recording secretarys report already distributed, corresponding secretary Gene Kelly announced that a "group" rate of \$19.95/ unit for TI Base was available. TI Base is a data base similiar to IBM D-BASEII Gene also put in a plug for PUG'S BBS service (824-6779) - TRY IT 9!you'll like it !! (8 BIT, 1 STOP, NO PARITY at 300,1200, 2400 baud).

The discussion centered around the Harrisburg show with Joe Ek!, our other presidential hopeful, speaking at length,(Fortunately, our candidates are <u>not</u> slinging mud, and <u>not</u> running negative ads, <u>nor</u> shoving in our faces POLLS !!@!@!@!

Out-going president Scott Coleman (ala.-dictator-grand -po-ba) announced at a previous meeting that indeed we will have a Pizza Party at the November meeting---so-- don't miss it - What a way to get the vote out..

Demonstration of TI Base by Mike Sealy went well. MacFlix, a TI utility which can use Macintosh files to draw pictures was shown by Gary Taylor, also Gary demonstrated an animated cartoon program of dinosaurs, pretty neat ! Gary promised to bring the Geneve to the Nov, meeting. John Willforth showed with his usual karisimmaa, Perfect Push and a snake that eats itself, unwillingly, called Nibbler, great action.! Classes in Hardware (J. Willforth) and TI Tips (Tom Mainier) followed the main meeting. The raffle prize was a data base filing program.

Nominations for the Election

Pres.	:	М. J.	Schmidt Ekl	V.Pres.	:	M. s.	Sealy Coleman
Rec. Sec	:	Ε.	Bittner	Treasurer	:	J.	Trayers
Cores. Sec.	:	G. G.	Taylor Kelly	Librarian	:	R. W.	Ekl Meyers

Respectfully submitted, Scoops Bittner I can't believe the number of hardware articles appearing in newsletters from around the country, and the world. I'd like to print them all, but due to the fact that I'm hearing very little feed back from our own members, who this newsletter is really for, indicating that they are actually constructing the offerings, I don't dare print more than one in any issue. I can imagine how much interest I would have in running to the mailbox for the latest issue of the WEST PENN 99'ER, if it was just chock full of articles on structuring more efficient code for a Fortran Compiler (I don't even know if I said that right).

I would like to tell you about two new ones that we've recieved, that those of you who are interested might not come across unless you do go through newsletter from other groups. If you want a copy of any just write, or call 412 527-6656 address on front of this newsletter.

The first is the POOR MAN'S A to D (I qualify). This article is by John Martin, and appears in Sept. 1988 issue of SNUGLETter from Southern Nevada, and uses a quad opto-isolater chip (PS2501-NEC) to make the joystick port an input port for analog signals which then get digitized by an assembly program he includes which samples stereo music and represents this music on your screen. Apprx. cost 12.- 16.. The system needs E/A, 32K, Disk in order to assemble and run his program, a Mini-mem unit could also make this possible, or a program written in BASIC, XBASIC might be written, but of course would have a problem in the frequency in sampling the input.

The other project is quite interesting, but would be within the capabilities of probably 1 in 10,000 TI users, and this deals with writing your own ROM (operating system) for the TI. The article presents you with instructions on putting one wire, and one diode on your CPU board. That's the easy part, now try writing the changes to be put into EPROM. This is where I believe most of you will stop. If you do tackle this, I'll volunteer to burn the EPROM if my MECHATRONICS will handle the EPOROM you choose. In any case, this project idea comes from CARL VERLAG, of Munich, West Germany. Again if you are interested call or write for your copy of these articles.

I'll have some good hardware at very resonable * prices at the November meeting, so bring money. I am * helping to distribute an estate. E/A package, TI 32K * mem. card, TI RS232/PIO card, Magnavox amber monitor, * and many other items of interest.You can contact me * earlier if you can't wait, or don't want to lose the opportunity to get something.

FOR SALE. 1 PASCAL CARD for PEB, all docs and disks, contact John F. Willforth (412) 527-6656 or write. \$100. FROM JAN TRAYERS

*******	*************	**********				
* *10/18 *	CASH ON HAND	* \$40.00 *				
* *10/18	LIBRARY SALES	4. 00 *				
* * "	MICROPENDIUMS	* 62.50 *				
*	"WRITER'S" BOOH	< 4.00 *				
*	DISK SALES	* 36.00 *				
*	RAFFLE	* 28.00 *				
*	DUES	* 60.00 *				
*	DATA CASES	* 50.00 *				
*	TOTAL	\$284.50 *				
*	SUPPLIES &	* - 21.75 *				
*	RAFFLE PRIZE -	262.75 *				
*	POSTAGE	* - 67.50 *				
*		* 195.25 *				
* * 10/31	DEPOSIT	+ - 60.00 *				
*	CASH ON HAND	135.25 *				
* * **********************************						
* * 10718	BANK BALANCE	* \$1141,83 *				
* * 10/20	DISK CASES	* _ 92.28 *				
*		* 1049.55 *				
* 10/25 *	MICROPENDIUM	_ 30.00 *				
* * 10/31	DEPOSIT	1019.55 * + 60.00 *				
* *	BALANCE	++++++++++++++++++++++++++++++++++++++				
*		*				

November 15th Meeting at 7:00 PM FREE RUZZA PARTY

- · ELECTIONS
- DEMONSTRATIONS
- · BE THERE OR BE

GENEALOGY SOFTWARE AVAILABLE FO	R THE TI
GENEALOGY WORKSHOP	\$49.95Micro-Sphere Tenex
	14009 E. Jefferson Boulevard
	Mishawaka, IN 46545
YOUR ANCESTORS, IN BASIC	\$10\$20KEN BARBER
	9648 S.E. Ellis
	Portland, OR 97266
YOUR FAMILY TREE	\$44.95HARDWARE, INC.
	P.O. Box 241746
	Memphis, TN 38124
"FILL-IN-THE-BLANKS"	\$8.00JAN KNAPP 314 428-0752
TI-WRITER GENEALOGY HELPER	Templates to print out fill-in-the-blank
	forms with sample filled-in forms. They
JAN KNAPP	include: TREE, GROUPSHEET, LARGE GROUPSHEET,
2318 ruckert Av.	ADDRESS PAGE, SOURCE, and CENSUS and a
St Louis, MO 63114	bibliography of sources for those doing
	genealogical work.

I'M SO EMBARASSED.... another correction on the clock circuit printed in the June '88 issue of the West Penn 99'er and PUG Peripheral. Here are both of the corrections. The chip U2 pin 14 (above pin 8) should be pin 11. This is the pin that outputs to pin 4 of U3. The other correction is that the eight DATA pins that output/input on the right of the clock chip (58167) are all wrong, in that pin 22 of the clock chip (D7) is shown going to pin 34 (D0) of the I/O connector in the Speech Synthesizer, or in the console. It should go to pin 37 (D7) of the I/O connector, and pin 21 (D6) to pin 40 (D6), etc. In other words the data lines are all reversed. Symptoms will be a very odd display of time ([),(#), etc. These are the only corrections (I'm sure). This occured because of my desire to get all the hardware projects I can out to you. I hope that none of you got discouraged because it didn't work. For USERS GROUPS who either handed out this project or reprinted it, I and your members will sure appreciate it if you publish these corrections. J.F. W.

PASCAL/p-CODE PART 12 Stan Katzman

I would like to discuss some more about files in the U.C.S.D. Pascal system.

In standard Pascal you cannot have random file access, but in U.C.S.D. Pascal you can. Random access is done using the Seek statement. The Seek syntax is "Seek(Filename, Recnum)". In the Seek mode you can both read and write data to the disk. In the Seek statement the "Filename" variable is the name of the file and the "Recnum" variable is the record number. The record numbering starts at zero. The illustration of the Seek statement is given in the rather long program included.

There are several conditions that must be observed when Seek is used: 1) The file must be of type "File", 2) when you "Seek" a record the Seek statement <u>must</u> be followed by a "Get" or a "Put" statement (see program) and 3) the syntax for displaying a field on the screen or working with one field is done thusly (as shown in the program) "Readln(Order^.Name)" or "Writeln(Order^.Name)" or you can use the "With" statement such as "With Order^ Do".

Some miscellaneous thoughts about Pascal files: Reading any Pascal file from disk is slower than BASIC; if you save numbers as reals it will slow up the reads so it will take minutes in order to read a disk file. So if possible do not store data as reals. Lastly if you save a file with the ".TEXT" suffix you will be able to read the file in the Editor. You can also modify a ".TEXT" file in the Editor.

DISK DRIVES (#3) by John F. Willforth

Last month you received the basic schematic of a disk drive tester. This month, I'll describe the functions and give you a schematic for a power supply to drive the unit and the disk drive under test.

The large connector on the left (J1) is the ribbon cable that goes to the drive's logic board. The small connector to the right of center near the top (J2) is the power cable to the drive. Rotary switch (S1) is the unit select switch which will select the drive by the strapping you have set on the drive. MOTOR ON (S2) turns on the drive motor, makes it easier to test this associated circuitry in the drive, the DIRECTION of head stepping (S3), in or out, WRITE GATE control (S4), mode selector for the drive, WRITE DATA signal (S5) to the write circuitry in the drive logic, STEP in the DIRECTION selected (S6), provide write data for the WRITE DATA line (S7) when the WRITE GATE is enabled, and do all this on or to the SIDE selected (S8).

You can watch to see if you are getting INDEX pulses on D1, and if DATA read from the drive is present on D2, or see when the heads are at TRACK 00 on D3, and if the WRITE PROTECT sensor is working at D4.

You can further check the drive speed at TP2 (Scope or frequency counter), or look at the signal coming off the read head at TP2 (scope or null meter), if you are on an alignment track for disk alignment.

This unit has a lot of uses (for an unintelligent device) and enables easy benchtesting and circuit tracing will be much easier.

The power supply shown below must be built at your own risk. Dangerous high voltage exists, and only experienced electronics people should construct this part of the Disk Exerciser project. You could use a Triple Output Powersupply available from Radio Shack for sometime now, 277-1022 provided you were only testing 1/2 height disk drives, or you could also use an old TI console power supply in the same way, and hook up the appropriate pins to the three points indicated in the upper right hand corner of the schematic of the exerciser.

PARTS:

- 1 Fuse Holder w/1A fuse.. F1 1 Switch DPST 115 VAC.... S1 1 Transformer 18VAC C.T Radio Shack 273-1515... T1 4 3A diodes 1N5402 R.S. 276-1143..D1, D2, D3, D4 3 Capacitors 2200MFD 35V. R.S. 272-1020....C1,C2,C3
- 2 Capacitors 100MFD 35V. R.S. 272-1016.....C4,C5
- 1 +5 Volt Regulator
- R.S. 276-1771.....M1 1 +12 Volt Regulator R.S. 276-1770.....M2
- 1 AC Power cable

Observe polarity on any capacitor marked. "+" toward +5 and +12. It 16 VAC c.t., can be acquired, the unit will run is also IMPERATIVE to attach the

Be sure to use a heat sink compound and T firmly mount D the 2 regulator O components to a large metal mass.



The power supply above will also make a very good source of DC power for a stand-alone disk drive, as long as the unit does not draw over one amp. on the +12 volt line. This unit will get very hot due to the very high (18 VAC) on the secondary. The reg.s have to drop this to 12, and the +5 v. reg. must also work very hard because it is dropping 9 VAC to the needed +5. If a transformer that outputs about

cooler. After you have constructed this unit, and two regulators to large HEAT SINK. put it into the box with the EXERCISER, connect the

three lines +12, COMM. (GROUND), and +5 to these pts. in the EXERCISER. We'll begin next month with a disk drive.

This series could be a lot of fun, but will also be a lot of work. John F. Willforth Nov. 1988

- -

EZ-KEYS. A REVIEW

by Bill Gaskill - TopIcs - LA 99ers, CA - Apr '88

EZ-KEYS is a new product from Asgard Software, designed with the Extended Basic programmer or Extended Basic program user in mind. To use it one must have Extended Basic, 32K memory expansion, and a disk drive. The program retails for \$14.95 and is currently available directly from Asgard Software or TENEX Computer Express. I as certain it will be available from other major 99/4A retailers in the near future.

A MACRO GENERATOR

EZ-KEYS is one of three keyboard generators I know of available for the 99/4A owner. PC-KEYS II, from Techni-Graphics, and SoftKeys from Guality 99 Software, are the other two. The fact that all three programs could be generically grouped into the "macro" development classification is really about all that they have in common. PC-KEYS II effers disk cataloging, a pop-up notepad, a pop-up calendar, and sser-definable/selectable screen dump capability, along with the ability to define a limited number of "hot-keys" that perfors common functions with one keystroke. SoftKeys is basically a "hot-key" macro generator without all of the added features of PC-KEYS II or the additional cost.

EZ-KEYS takes a different approach to the concept of macro generation. It too allows you to define "hot-ksys", but the kays you define can do much, much more than mither of the other two programs. In fact, by my definition, EZ-KEYS is really the only true macro generator of the three. In my experience as a user of "other" computers, macros are short programs that "remember" knystrokes for you, so you can later call them up at the press of a single key. In other words, they are time savers that shortsn the number of steps you have to go through to perform a desired function or series of functions.

On all of the MS-DOS macro generators I have used, macro files are built in one of two ways. One method is to use a "remember" mode that tracks and then saves keystrokes as you press them and another method is to provide a macro editor that allows you to write and save small files containing the desired keystrokes. E2-KEYS is of the second type. Instead it provides a macro aditor that appears at the press of a single key.

EZ-KEYS allows up to 55 keys to be defined for macro use, with each macro capable of being 668 characters (about 7/8 of a screen) in length. Perhaps the meatest feature of EZ-KEYS is its ability to link macros together. This means one macro fils can RUM another, thus providing almost unlimited potential to the utility the EZ-KEYS environment can offer the IB programmer or user. EZ-KEYS also RUMs Extended Basic programmer or user. IB programmer.

For instance, if you wished to have a disk cataloging program available at the touch of a key, you could write it in Extended Basic, LIST it to disk so it is SAVEd in BV/80 format, then define a macro for it. In fact, such a program is provided on the EZ-KEYS disk. Programming a macro is simple if you are creating simple macros. It can become quite complex and demanding if you really want to build some mophisticated applications. Saving a macro is simple and straightforward. You simply define it in the Macro Editor, exit out of the editor, do a BYE at the READY prompt and then load the program again. You will immediately be given the option of loading or saving your macros. When you save them, all macros that exist in memory are saved to disk for use any time the EZ-KEYS movironment is loaded.

To use your macros in a RUNning program environment you simply edit a line in the EZ-KEYS program so it RUNs the first program you want to activate, then you must save the EZ-KEYS program as LOAD. When the EZ-KEYS LOAD program is rmad into memory it brings all of your macro definitions with it, them RUNs your first program. That's it! No programming expertise required herm, just a user-friendly common sense approach to interfacing with your IB program(s).

Extended Basic programs that use assembly language subroutines may also be used with EZ-KEYS. The author has included an EZLOADER that will allow you to save custom routines and your macros all together. Assembly routines are loaded first, then your macros so they do not conflict with the subroutines already in low-memory. The whole package is then saved as a emeory-image file and can be called up whomever you use the application with the custoe assembly routines. The really neat thing here is the ability to customize the EZ-KEYS environment to fit as many different uses or programs as you have.

A PROGRAMMING UTILITY

EZ-KEYS is an assembly language coded program, designed to operate in an Extended Basic environment. Aside from its ability to generate macros it also provides a set of utilities for the Extended Basic programmer. While in the command mode (at the 1 READY prompt) in XB EX-KEYS allows you to set a timer that will automatically SAVE your work in case of a power failurs or interruption. The time intervals can be from 0 to 18 sinutes apart and two files, BACKUP1 and BACKUP2 are used to save your work. All worked is saved welly to DSK1. Another option allows the setting of background and foreground colors in the programming environment, such the same as the GrameKracker Utilities or John Johnson's Horizon RAM Disk menu allow. Colors may also be set for the Macro Editor and the special characters displayed in a macro file.

Another routine will highlight numbers and arithmetic operators so they appear on screen in the reverse color of the background and foreground colors chosen. When a running program is being used you may also set all character sets to the same color by linking to a routine mamed RCDLDR.

Although the manual cautions Extended Basic might not always be able to interpret it, EZ-KEYS lets you write a

single program line that can be 23 screen lines long. How's that for expanding the capabilities of Extended Basic? Additionally, you can press Function 7 or Function 6 to move the cursor directly between program lines while in the programming (immediate) mode.

You may also write macro files in the TIW Editor, in a manner similar to writing a .BAT file for the GENEVE or any MS-BOS machine. The author included a customized CHARA1 file to use on your TI-Writer disk. This file contains the character definitions needed to display the special characters that represent specific macro functions. This is another example of the forethought put into this program. I would guess few first-time users would opt for this method of development though, since it requires the use of the Transliterate Mode in TI-Writer and is only sparsely documented in the EZ-KEYS manual. There is a chart in the documentation that shows the various equivalents that are available.

Once you have created the macro file you simply print it to disk, run the POKER program provided on the EZ-KEYS disk, and the macro file is then assigned as a macro definition.

If EZ-KEYS has a shortcoming it is in the method used to call macros from a RUNning XB program. The cursor must be displayed on screen for a sacro to be accessed. You cannot, for instance, access a sacro when the program being RUW is looping at a CALL KEY statement. Once the file instructions within a macro have been set in motion they are suspended only by a "hold" command in the macro (a Ctrl H). So you must have programmed a Control H in the macro file so it appears at the proper point in your IB program. This can be tricky and a little confusing to the novice programmer. I would have rather seen an execution routine used that could be called at any time. One curious oversight in the EZ-KEYS program involves character definitions. A custom character set is used in the macro generator and is not reset when an XB program is used. You end up with a couple of lower case letters out of line with the standard TI chracter set in your running XB program. While this is easily overcome by restoring the offending characters with CALL CHAR statements in the program you are running, it would be mice to see EZ-KEYS take care of this for you. It is one less than professional aspect of a program that is otherwise truly representative of "commercial" quality software.

EASE OF USE

While creating macros is not in the suggested domain for a new programmer, setting up the EZ-KEYS program to use macros is. More detailed documentation is meeded to help the less adventuresome wade through the rigors of complex macro development.

DOCUMENTATION

The documentation is adequate for simple macrodefinitions, but falls short of being a complete tutorial for the advanced macro programmer. It covers most of the "absolutes" of macro development in the EZ-KEYS environment, but stops short of really being a useful guide to the advanced macro developer. However, in fairness to Asgard Software, it's difficult to offer such an outstanding product and couple it with outstanding documentation for \$14.95. The documentation is well written and understandable, an important consideration in any new software purchase. If EZ-KEYS "takes off" perhaps Asgard will develop follow-up products for it such as a disk of predefined macros or a tutorial on advanced macro programming.

VALUE

Harry Wilhelm is the author of EZ-KEYS. I know mothing about Mr. Wilhelm nor do I recall ever reading his mame in any of the samy 99/4A publications. After seeing the product he has produced. I hope he continues to write programs for the 99/4A (and hopefully the 9640). If future Wilhele applications are anything like EZ-KEYS, we are all in for a treat. EZ-KEYS is a superb first-release application. It is well thought out, professionally executed and virtually error free. For the adventuresome programmer or user EZ-KEYS promises unlimited potential and utility. It is truly a professional application that needs only more complete documentation and some fine tuning to push it into the "stellar" software class. If you don't have EZ-KEYS you should buy it. You won't rearst the seager \$14.95 investment. Even if you do not use it to do ANY sacro development of your own, you will likely find another, parhaps more important, use for it. would not be surprised to see future IB type L applications developed under the EZ-KEYS environment. It is truly a powerful development tool that cries out for an imaginative programmer to come along and demonstrate some of its potential. With the right combination of good sarketing, dependable customer support and continued development of the product, EZ-KEYS could become a standard among 99/4A users. It is THAT 600D! It is only in its infancy in version 1.0. I am sure the best is yet to come.

FINAL GRADE

You will note EZ-KEYS falle down to a "B" rating in some categories. In the PERFORMANCE area I knocked it down because of the less than flexible sethed used to call up a sacro and the program's inability to suspend sacro operations more effectively. In the EASE OF USE area 1 took some points away because of the complexities required to design more than just simple macros. The DOCUMENTATION lost points because of brevity and several typos that smuck into the final product. The VALUE category makes up for all of the little shortcomings I found in this first release. It is the "missing link" that we have been looking for in making more of the 99/4A than just a single purpose machine. For \$14.95 you simply can't go wrong. If the TI Community supports EZ-KEYS like it deserves, I as confident the incentive will be present for Harry Wilhelm to continue development of the product. I am equally certain other talented programmers will develop applications to run in the EZ-KEYS environment. The end result to our support of this product is sure to be an even better product in the future.

REPORT CARD:

Performance.....B Ease Of Use.....B Documentation....B Valum......AAA FINAL GRADE.....A

Extended BASIC Speaking Program LISTer by Chris Schram

There are those out there in the Texas Orphanage who truly seem to possess "The Gift" for inventing new ways of utilizing that funny little monster. Then there are the rest of us who can from time to time, if we're lucky, assemble the little bits and pieces of others' genius into something, if not completely original, at least suitable to our own needs. This article is about a program I would not have been able to write had others not first paved the way.

I was intrigued by Steven Richardson's proofreader program from the June 1987 issue of MICROpendium, but the fact that it would only operate in console BASIC and required the TE-II module made it unsuitable for me. Most of the programs I enter are in Extended BASIC. I also found the "LIST a screenful, then RUN" nature of Mr. Richardson's program a bit awkward, to say the least.

I set out to put the pieces together. I already owned a copy of TEXT-TO-SPEECH (ENGLISH) (PHD 5076), so getting Extended BASIC to talk would be no problem. I also had a copy of Barry Traver's TOKEN/READ, a program that PEEKs into Expansion RAM and displays the words associated with the Extended BASIC tokens it finds. Most of my program was lifted from Mr. Traver's program. Mr. Traver gives credit to John Clulow and Michael Riccio for inspiration. I must do the same, for they, indirectly, assisted me, too.

1 ' SAVE DSKI.LISTINIT 12,7)ERASE ALL:X\$:: CALL LI NK("XLAT", "^>"&X\$,B\$):: CALL LINK(*SPEAK*,8\$,43,128):: 6 110 ' EX. BASIC SPEAKING! 120 1 PROSPAM LISTER 010 350 320 CALL INIT :: CALL LOAD(* 140 ! + INITIALIZER + DSK1.SPEAK*, DSK1.XLAT*, DSk 150 * ************ 1.SETUP*) 330 CALL LINK("SETUP", "DSK1. 160 ! by: Chris Schram ! 170 1 San Jose, CA DATABASE*) 340 RETURN 180 ! July 1987 190 350 ON ERROR STOP :: FOR I=3 200 : Requires: 2 TO 127 :: C\$(0,I)=CHR\$(I): 210 ! TI EXTENDED BASIC : NEXT I :: FOR I=129 TO 254 220 ! SPEECH SYNTHESIZER :: READ C\$(0,1):: NEXT I :: 230 1 EXPANSION MEMORY C\$(0,34)=CHR\$(34)&CHR\$(34) 240 ' DISK MEHORY 360 READ I.C\$(1,I):: IF I<25 250 ' TEXT-TO-SPEECH 5 THEN 360 (ENGLISH) DISKETTE 370 OPEN #1:"DSK1.LISTDATA", (PHD 5076) OUTPUT, DISPLAY , FIXED 26, REL 260 1 ATIVE 270 ' Note: 380 FOR I=0 TO 255 390 IF C\$(1,I)=** THEN X\$=C\$ 280 ! I have found that the TEXT-TO-SPEECH (0, I) ELSE X\$=C\$(1, I) 400 CALL LINK("XLAT", """%X\$. program does not work with the B\$):: PRINT \$1,REC I:CHR\$(LE FOUNDATION expansion N(C\$(0,1)))&C\$(0,1);TAB(12); 290 ' secory card. There SE6\$ (B\$&RPT\$ (CHR\$ (0), 15), 1,1 5) may be other 410 DISPLAY AT(23,1):C\$(0,1) hardware/software ;TAB(12);B\$:C\$(1,1):: CALL L conflicts yet to be INK("SPEAK",8\$,43,128) discovered. 300 DIM C\$(1.255) 420 NEXT I 310 ON ERROR 320 :: X\$="INIT 430 CLOSE #1 :: CALL CLEAR : ALIZING.... :: DISPLAY AT(: STOP.

The following two programs were originally one stand-alone program that could be MERGEd into what you wanted LISTed. The only problem was that the TEXT-TO-SPEECH machine code ate up so much memory that it was really only able to list itself. Not very useful. The way it stands now, you RUN the LISTINIT program just once to create a disk file that is used by the LIST program. That saves a head of memory at the sacrifice of some speed because of all the disk access.

Note: If you own the TEXT-TO-SPEECH program, then you already know if it works with your hardware setup. I know that it does NOT work with the FOUNDATION expansion memory cards and there may be a few surprises with other configurations. In other words, first determine if it s worth it to proceed.

Keeping everything on DSK1. for the time being, type in the two programs below. (Be careful, the LIST program uses é, [,], _, and V as variable names.) Copy SPEAk, YLAT, SETUP, and DATABASE from the TEXT-TO-SPEECH disk. Run LISTINIT to create the LISTDATA file which contains the text and speech strings used by LIST. OLD the program vou want to list. MERGE the LIST program. RUN. You will be given a chance to select what lines you want to list. You can watch the tokens as they appear one by one on the screen while the program speaks. The screen listing closely resembles a normal LISTING. Press [Fcth 4] when you've seen/heard enough. If you want to RUN the program under test, just REM out line 2.

```
510 DATA 93.) BRACKET,94,CIR
440 DATA ELSE,::,!, IF, 50, 60T
O, SOSUB, RETURN, DEF, DIM, END, F
                                   CUMFLEX,95.UNDER LINE,95,6PA
OR, LET, BREAK, UNBREAK, TRACE, U
                                   VE,123,LEFT BRACE,124,VERTIC
                                   AL LINE, 125, RIGHT BRACE, 126.
NTRACE, INPUT, DATA, RESTORE, RA
                                   TILDUH, 127, DEL
NDOMIZE.
                                   520 DATA 130,DOUBLE COLEN,13
450 DATA NEXT, READ, STOP, DELE
TE.REM.ON.PRINT.CALL.OPTION.
                                   1, TAIL REM, 134,60 2,135,60 S
                                   UB,142, BRAKE, 143, UNBRANE, 147
OPEN, CLOSE, SUB, DISPLAY, IMAGE
, ACCEPT, ERROR, WARNING, SUBEXI
                                   ,DAYTUH,162,/DISPLAY,165.AIF
T, SUBEND, RUN
                                   n₽
                                   530 DATA 167, SUB * 11,168, SU
460 DATA LINPUT..... THEN. TO
,STEP, *, *,;,:,), (, &,, OR, AND,
                                   B END.177.2.179.COMMA.180.SE
XOR.NUT.=.<.>.+.-.+./.^....
                                   HEECOLEN, 181, COLEN, 194, AMPER
EDF, ABS, ATN, COS, EXP, INT
                                   SAND. 188, X OR. 191, LESS THAN
                                   540 DATA 192, BREATER THAN. 19
470 DATA LOG, SEN, SIN, SQR, TAN
                                   3, PLUS, 194, HYNUS, 195, TIME5, 1
,LEN,CHR$,RND,SE6$,POS,VAL,S
                                   96, DIVIDED BY, 197, RAISED TO
IR$, ASC, PI, REC, MAX, MIN, RPT$,
,,,,,NUMERIC,DIGIT,UALPHA.S
                                   THE POWER, 202, E 0 F, 201, A E
                                   5,204,A T N
IZE, ALL
480 DATA USING, BEEP, ERASE, AT
                                   550 DATA 205,00 SINE,205.8 1
, BASE, , VARIABLE, RELATIVE, INT
                                   P.207, I N T.209, S & N.210, S
                                   INE,211,5 @ R,212,TAN GENT,2
ERNAL, SEQUENTIAL, OUTPUT, UPDA
TE, APPEND, FIXED, PERMANENT, TA
                                   14,C H R $,215,RAND,217,P 0
B. . VALIDATE
                                   S
490 DATA 0.END OF LINE, 32.SP
                                   560 DATA 219,5 T R $,220,A 5
ACE, 33, EX CLUHMATION, 34, QUOT
                                   C,222,REC ERD,225,R P T $,2
E QUOTE, 35, POUND SIGN, 38, AMP
                                   34.U AL FUH.239,E RACE,243.V
ERSAND, 39, APOSTRUHFEE, 43, PLU
                                   AREEUHBL, 244, REL UH TIV, 253.
                                   NURBER
S,44,COMMA
                                    570 DATA 249, )APPEND, 251, PER
500 DATA 45, DASH, 46, DOT, 58,C
                                   MANENT, 199, QUOTE, 201, LINE, 2
DLEN, 59, SEMEECOLEN, 60, LESS T
                                   55.>INITIALIZING
HAN, 62, GREATER THAN, 53, QUEST
ION MARK, 91, ( BRACKET, 92, REV
ERSE SLANT
```

1 ' SAVE DSK1.LIST, MERGE 32022 DISPLAY AT(14,): "T0" SE6\$(1\$,], ASC (SE6\$(1\$, [, [))) : 199-32044 DISPLAY X\$;:: [F Y\$=** 2 CALL LIST :: STOP :: ! eP-; MAX(F,T):: ACCEPT AT(14,16) 32082 CALL PEEK(-31952,4,5,0 32000 PEP+ VALIDATE(DIGIT)SIZE(-5)BEEP: THEN CALL LNUM (X\$)ELSE CALL ,D):: X=A+\+8-65536 :: Y=C+\ +D-65539 32002 SUB LIST SPEAK (YS) 32004 @=0 :: [=1 ::]=2 :: _ 32024 CALL FIND(T, J2):: CALL 32046 LINPUT #E,RED Y(S+E):Y 32084 IF X>=Y THEN SUBEVIT PEEK(J2,A,B):: T=A+\+B :: D \$:: #=ASC(SEG\$(Y\$,[,[)) 32085 CALL PEEKIY, A. 5 .:: 11= =12 :: \=256 :: 6010 32008 : A+1+B :: IF L1(L THEN Y=Y-4 ISPLAY AT(14,16)SIZE(5):STR# 32048 IF (LEN(X\$)>E OR K>E)A : L\$,Q\$,X\$,Y\$:: A,A1,A2,B,C ND(QF<>) AND UF<>))THEN DISP :: 6010 32084 .D.F.I.J.JI.J2.K.L.QF.S.T.UF (T)32088 !@P+ LAY * *: :: CALL KEY :: CALL PEEK :: 32026 DISPLAY AT(18,L):** PR ESS ANY KEY TO PAUSE +* 32050 QF=QF-[:: UF=UF-[32090 SUBEND DIM Y(3)32028 FOR L=J1 TO J2 STEP -4 32052 IF RF=L THEN DISPLAY C 32092 SUB SPEAK (P\$) 32006 10P-32094 6010 32096 :: CALL INT 32030 CALL PEEK(L,A,B,C,D):: HR\$(34)::: CALL SPEAK(Q\$) 3200B OPEN #L: DSK1.LISTDATA 32054 IF UF=E AND K>E THEN D T :: CALL LINK :: CALL LOAD ", INPUT , DISPLAY , FIXED 26,8 J=C+\+D-65537 :: CALL PEEK(:: '@P-ISPLAY CHR\$(32): ELATIVE J.A1) 32096 DN ERFOR 32098 :: CALL 32010 LINPUT #E,REC 255:X\$: 32032 X\$=STR\$(A+\+B):: DISPL 32056 CALL KEY(@.K.S):: IF S : DISPLAY AT(_,7)ERASE ALL:* AY ISE "::: CALL SPEAK(L\$): <E_THEN_32062</pre> LINK("SPEAK", 8\$, 43, 1281:: 5 INITIALIZING.... :: CALL SP 32058 DISPLAY BEEP:; USEXIT : CALL ENUM(X\$) 32060 DISPLAY AT([,[)SIZE(29 32098 CALL INIT :: CALL LOAD EAF (SE6\$ (1\$, ,15)) 32034 FOR I=[TO A1 :: S=0 : 1:** PRESS ANY FEY TO RESUME (*DSK1.SPEAK*,*DSk1.SETUP*) 32012 LINPUT #E,REC 199:Q\$: ; CALL PEEK(J+I,Y(0),Y(1),Y(J2100 CALL LINE (*SETUP*, *DSK : @\$=\$E6\$(@\$,_,15) +* :: DISPLAY AT(E,E)SIZE(2) (2), Y(3)1.DATABASE") 32036 IF Y(5)=199 THEN DF=Y(B): :: CALL KEY(@,K,S):: IF 32014 LINPUT #E,REC 201:L\$: : L\$=SEG\$(L\$,_,15) \$+[)+[:: I=[+] :: S=5+] :: S(E THEN 32050 32102 RETURN 32104 10F+ UF=E :: DISPLAY CHR\$(34);:: 32062 NEXT 1 32016 ON ERROR STOP :: CALL CALL SPEAK(@\$):: IF @F=E THE 32064 DISPLAY 32106 SUBEND FIND(3,F):: CALL PEEK(F,A,B) 32108 SUB LNUM:X\$> 32066 NEXT L N DISPLAY CHR\$(34);:: CALL 5 :: F=A+\+B :: CALL FIND(3200 32068 DISPLAY BEEP: : : *PRES 32110 [=1 :: =12 :: E070 32 PEAK (Q\$) 0,T):: CALL PEEK(T+4,A,B):: S ANY KEY TO CONTINUE" 112 :: Y\$:: I :: '#P-1=A+\+B 32038 IF Y(S)=200 THEN UF=Y(32070 CALL KEY(@.K.S):: IF S 32112 FOR I=E TO LEN(X\$) 32018 DISPLAY AT(_,5)ERASE A S+[)+[:: !=[+] :: S=S+] :: 32114 LINPUT #E,RED ASE SES\$ LL: "LIST FROM?"; F :: ACCEPT OF=# =@ THEN 32070 32040 IF Y(S)=201 THEN X\$=ST 32072 6010 32015 (X\$,I,[)):Y\$:: Y\$=SEG\$(Y\$,_ AT(_,16)VALIDATE(DIGIT)SIZE(32074 10P+ ,15):: CALL SPEAK(Y\$) R\$(Y(S+[) #\+Y(S+])):: Y\$="" -5) BEEP: F 32116 NEXT 1 32020 CALL FIND(F,J1):: CALL :: [=1+] :: S=S+] :: 60T0 32 32075 SUBEND 32118 !@P+ 32078 SUB FIND(L,Y) PEEk(J1,A,B):: F=A#\+B :: D 044 ELSE L[NPUT #[,REC Y(S): ISPLAY AT(_,16)SIZE(5):STR\$(32080 \=256 :: 60T0 32082 :: 32120 SUBEND YS A,B,C,D,L1,X :: CALL PEEF : F) 32042 Y\$=SE6\$ (X\$,_,15):: X\$=

GLOSARY OF COMPUTER TERMS

BIT: Describes computers, as in "OUR" computer cost quite a BIT. BOOT: What your friends do to you when you brag about your computer. BUG: What your eyes do after staring at a screen too long. CHIPS: Used to insert into DIP while working at your computer.

- COPY: What you do at school cause you were playing PACMAN so much last night.
- CUSOR: What you become when your computer breaks down.

DISK: What slips in your back after hours of sitting down.

DUMP: Where all your hobbies go'after buying a computer.

ERROR: Made when you walked into the computer store "just to lock".

EXFANSION UNIT: The room you add to your house to store your computer. FLOPPY: The condition of the user's muscles after sitting around and

eating chips.

HARDWARE: Rakes, moweres, and other things you haven't touched this summer.

MENU: What you'll never seem again, cause now you're too poor to mat out. PROGRAM: What you used to watch on the TV, until you hooked the couputer to it.

RAM: What you do to the side of your of your computer when it's broken. RETURN: What you do with the computer after RAM doesn't work. WINDOW: What you throw the computer thur when you con't RETURN IT. HELP FOR "PUSH" USERS WITH RAM DISKS.....by John F. Willforth

There is a very intricate game called PUSH, I have not yet been able to find out for sure it's reimbursement requirements, (commercial, freeware, public domain) but it is a very challenging game. The problem I have is that it takes awhile to load from a disk drive, and so using a string search utility I sought the ASCII strings that dealt with the disk designations "DSK1.", which allowed it to load it's two support programs from the same disk that the D/F program "PUSH" resided on, DSK1.. This was not as easy as it would first appear. The string "DSK1." was not found! I then took a sector editor, and slowly stepped through the sectors of the "PUSH" file, and found two very close strings that fortunately did modify to the desired drive. This is just to save you a little time in case you desire to do the same with this program.

In the ASCII mode search for "DSBK1F" and change the "1" to the drive number of the RAM DISK you want the program to run from, and then search for "DSBK1B" and change the "1" to the same drive as the first change made.

In HEX mode search for "4453424B3146" and change "31" (1) to "34" (4) drive 1 to 4, and then search for "4453424B3142" and change "31" (1) to "34" (4) drive 1 to 4. If any other drive, 2, 3, 5, 6, 7, etc. use "32", "33", "35", "36", "37", etc respectively. Thats all there is to it! Note that the search will be easier and certainly more reliable if the only file on the disk at the time is "PUSH", but I found it with many on the disk because none that were on the disk had these particular strings present.

NEW SOFTWARE IS RELEASED NUTMEG TI-99ERS...OCT., 1988

From Asgard Software, P.O. Box 10306, Rockville, MD 20850

- COLUMN ATTACK: Fast action arcade game written in Fortran 99. Defend Earth against rampaging alien spaceships. Requires 32K, disk system and Extended Basic. \$9.95 + \$.75 S/H.
- DINOSAURS: TI-Artist Graphics. 2 disks of fun for all ages. Has dinosaur font, background scenery & dinosaur animation. Create pictures, stickers, cards, reports. \$12.95 +\$.75 S/H.
- 3. QUICK-RUN: Extended basic utility makes other XB programs run instantly. Can eliminate time consuming program initialization. Takes "snap-shot" of program & saves to disk. \$9.95 + \$.75 S/H. Cont. page 10

From Genial Computerware, P.O. Box 183, Grafton, MA 01519

- MacFLIX: Allows TI users to view, print and save graphics produced by MacIntosh MacPaint. Can be saved in TI Artist _P format. \$15.00 + \$1.00 S/H.
- GRAPHICS EXPANDER: Version 2.0 reduces & expands TI-Artist fonts & instances. Also CSGD fonts & small graphics. Will also convert formats between these two programs. Many other features. \$10.00 + \$1.00 S/H. Owners of Version 1.0 may send original disk & \$3.00.

TWO JOYSTICKS IN ONE......BY Curtis Borders.....C.O.N.N.I.E.

This is how I made two joysticks out of one:

First you will need one of those surplus joystick cables. All the pins will have to be there with the exception of pin 1 and 6. (Pins 1 and 6 are not used on the TI 99/4A) You can get one at "Star Surplus" on N. High St. Columbus, OH. They sell for about \$1.99. OK, now that we have the cable, take your favorite joystick- it can be Atari, Boss, EPSY500XL or Wige but I wouldn't write the case of finance of The second starts and the second starts are starts and the second starts and the second starts are starts and the second starts and the second starts are starts and the second starts and the second starts are starts and the starts are starts and the second starts are starts and the second starts are starts and the second starts are starts and the starts are starts and the second starts are starts and the second starts are starts and the second starts are starts and the starts are starts are starts and the second starts are starts are starts are starts and the second starts are starts ar

EPSX500XJ, or Wico, but I wouldn't waste my time on TI joysticks. Take the joystick apart and unsolder the cable from the connectors, all but the ground (or common) wire. That's the wire that connects all the pads together.

Take your new cable and an ohm meter and write down all the pin numbers and what color wire it is, because all color codes may not be the same. Take the _____ color wire from pin 2 to one of the outside terminals of the switch, then take the _____ color wire from pin 7 to the other side of the switch.

Take a short piece of wire from the center of the switch and solder it to the ground (common) wire. If the switch is wired up right, when it is in one position, you will be using joystick number 1, and when it is in the other position, you will be using joystick number 2.

Take the _____ color wire from pin 3 to the joystick up position. Take the _____ color wire from pin 8 to the joystick down position. Take the _____ color wire from pin 9 to the joystick right position. Take the _____ color wire from pin 5 to the joystick left position. Take the _____ color wire from pin 4 to the joystick fire button. The other side of

the fire button will go to the ground (or common) wire. Use schematic below.

	-	CJ	0
	*	#	5
	4	4	н
	M .	U1	ы Ц
	תל	רל	
	Ö	0	З
	35	55	S .
	a í	U I	C
	N C	N C	
്തി	Do	4 00	
	E 0.	b a c d	
2.00	P B L	4 + = 3 8	
1.0	000.4	W 0 0 0 m	
لتشتقا	HUDE.	ANDR	
		ი თ – ი თ თ	



Planning a trip to Reno or Las Vegas? so you will want to try this program and If practice up for your trip. You've probably been there in the past

at one time or another. So you will recall the the whirring and the sounds of the slot machines and found yourself wondering what the fascination was.

You"ll soon find out when this program your computer into a fabulous Nevada fruit machine. All the playing inturns your compute style fruit machine. you need will appear structions oñ the screen. At certain points you will be presented with a list of options.

JAC(7) =-1

ELT JO

ET 💷

ETURN

6010 180

SENTINI

6010 300

ETURN

8F

B,115,15,8,109 90 NJ\$=CHR\$(138)

1280 J=0 :: FOR WD=1 TO 3 ::

VV=V(ND) 1290 IF VV)4 THEN FV=VV :: 6

010 1320

When you see: Insert, Hold, Play OR End enter your choice by typing the first letter of the option you want, for instance P Reeps your machine playing. The reels are numbered 1, 2, and 3. If you want to hold one or more reels, type in the appropriate number or numbers after you have entered H The reels are want to "hold for Hold.

FROM PUNN, NOVEMBER 1988

Lights will flagh and music play as the wheels whiz around. Have you won this time? Keep playing and sooner or later you are bound to hit the Jackpot.

IO REM TIGETTENDEDI-SLOT MAC 390 NEIT TI :: #21.44 400 #: DEFINE 1-4-4.TERS 410 CALL CHAR(96, "--- FFFFFF FFFFFF") HINE 20 CALL CLEAR :: RANDONIZE JO DIM WF(131,R(2,13),JAC(13)): JAC(5)=-1: JAC(61=-1: 420 CH\$="OFOFOFOFOFOFOFOF" : 420 CH4: OFOFOFOFOFOFOFOFOFOF : CALL CHARIGO,CH4) 440 CALL CHARIGO,CH4) 440 CALL COLDR(9,6,1,10,2,16 ,11,7,16,12,16,6,13,11,16,14 13.16) 13.16) 13.16) 13.16) 13.16) 14.17 14.17 15.16 15.17 15. : JAC(/)=-1 40 FDR VV=1 TD 13 :: READ WF (VV),R11,VV),R(2,VV):: MEIT VV 50 BATA 110,0,0,116,0,0,115, 0,0,109,0,0,106,B,50,105,B,4 0,114,7,30 60 DATA 113,5,25,104,3,20,13 6,3,20,137,2,10,128,2,10,112 0,0 70 DIM JPI(4),JPY(4),JF(4),J CC(4):: FOR J0=1 T0 4 :: REA D JPI(J0),JP(J0):: N FIT J0 460 CALL CHARILOB, "FFFFFFFFF FFFFFFF1 FFFFFF1 470 As*:E*5FF3D3C3CE404* 480 Call (-=#(109,A*):: Call CHAR(115,=* 490 As=*.E:S*=BC3C3C2720* 500 Call CHAR(110,A*1:: Call CHAR(116, A\$) 510 CALL CHAR(136, *020430566 BO DATA 11,8,110,12,8,116,14 A566A3C* 520 CALL CHAR(114, *00D06C7C7 C3B1000*) IOO DIM WPI(3),WPY(3),V(4),H VV(3): WPI(1)=11 :: WPI(2)= 13 :: WPI(3)=15 :: FDR [=1] D 3 :: WPY(1)=10 :: NEIT] 530 CALL CHAR1112, *0204387C7 703800*) 540 CALL CHAR (104, *020206367 8783000*) 110 DIM HPX(3) HPY(3), HFS(3) :: FOR I=1 TO 3 :: HPX(1)=WP 1(1):: HPY(1)=13 :: MEXT I 120 HI\$=CHR\$(107):: NH\$=CHR\$ 550 CALL CHAR1128, *004060607 0301E00*1 560 CALL CHAR(137, "020413C37 11300"1 570 CALL CHARPAT(36,A8):1 CA LL CHAR(106,A\$1 (108):: HHS=CHR\$ (96):: HOS=H SBO CALL CHARPAT(63,A\$):: CA LL CHARII20,A\$) 590 CALL CHAR(113, "081C3E7F3 EEICOBOO") 600 CALL CHAR(104, "OBIC2A772 A080800") 610 CALL CHAR (105, "0010387C7 170 PER BLINK SIS/S28 AND GE C103800*) 620 CALL CHAR (138, "000000000 0000000") IBO DISPLAY AT(PLY, PLI):SIS; 1: CH=0 :: FOR DE=1 TO 10 190 IF CH=O THEN CALL KEY (0, 630 RETURN GET, CH) 20D NEIT DE 1: DISPLAY ATIPL 640 RENILLAN SCREEN 650 CALL CLEAR :: CALL SCREE N(12) 660 CALL HCHAR(7,12,96,7) 670 FOR 188 TO 12: CALL HC HAR(1,11,96,91:: WEIT I 680 CALL *[=4877,20,60] 690 CALL *[=48712,30,96] 710 CALL *[=4713,12,96,7] 720 FOR 1=:4 TO 19 :: CALL H CHAR(1,13,96,5):: NEIT I 730 CALL HCHAR(20,12,96,7] 730 CALL HCHAR(20,12,96,7] 750 C PLI):S28;:: IF CH=1 THEN R 1(17) 210 FOR DE=1 TO 10 11 IF CH= 210 FOR DALL KEY(0, 5ET, CH) TTE NEIT BE 11 IF CH=1 THEN ECTERN 240 REN AND INCREMENT TO MON 260 1127 AY AT (1,7): HO +AD; 270 IF 35N W -0 THEN 290 280 CALL 500 N -50,200,21:: 290 FOR SO=500 TO 700 STEP 1. 770 DISPLAY AT(10.3):CHR#111 21; * = "; 780 FOR VV=12 TO 5 STEP -1 790 112 AT AT (23-VV,)) 1 *-"; RPT* 1-#*(#F(VV)),2); *-=";R(00 :: CALL SDUND(-50, 50, 2):: NEIT SO JOO NEIT AD (: HO=HO+IN :: R 1, VV / 1 800 DISPLAY AT (23-VV, 20) : RPT \$ (CHR\$ (NF (VV)), 31; **; R12, VV 310 REM REMOVE DOUBLE 320 DD=0 :: CALL HCHAR (2, 3, 3 2,71 330 FDR JD=1 TU 4 :: CALL HC HARIJPY(JD),JPI(JD1+2,JF(JD1 1:: JCC(JD)=0 :: WEIT JD 340 WJ=4 :: RETURN 350 CALL HCHAR(24,3,32,28):: BIO IF JAC(VV) THEN DISPLAY A 810 IF JAC(VV) HEN DISPLAT A T(23-VV,27):*•J*; 820 MEI VV :: RETURN 830 REM \$\$ ADAPT VARIABLEB 840 NT=NT+1 :: IF NI>O THEN HB=-1 :: WI=0 850 FOR HD=1 TO 3 :: HF\$(HO) =NH\$:: DISPLAY AT(HPY(HD),M PI(HO)):MN\$;:: NEXT HO RE :== 360 === JACKPOT 370 ::= 11=1 TO 4 :: CALL HC HAR(4,3,32,JC) 380 FOR C=[TO JC :: CALL SO UNG(-100,150+201C,0):: DISPL AY AF(4,C):'J';:: NEIT C

860 IF DD THEN 910 870 FDR JD=1 TO 4 :: IF JCC(JD)=0 THEN 900 880 JCC(JD)=JCC(JD)-1 :: IF DECISION THEN YOO BYO NJ=NJ+1 :: CALL HCHARIJP Y(JO), JPX(JO)+2, JFIJD) YOO NEII JD :: GOTD 920 910 DC=DC-1 :: IF DC=0 THEN 60SUB 320 920 IF JA THEN DISPLAY AT(4, 1: ' ':: JC=JC-I :: IF JC= G IAEN JA=0 930 RETURN 940 REM ## NOT EN: 5H MONEY 950 HB=-1 :: 515JB 320 960 JA=0 :: CALL MUHAR(4,3,3 UB 180 1040 IF GET(49 CR GET)51 THE N RETURN ELSE HU=6ET-48 1050 NH=NH+1 :: 1F HE\$(HU)=N HE THEN HES(HO) HIS ELSE HES M3 Inca M *IND: *Dis CLac In * (HDI === @ 1060 €::] 1020 1070 ₽2m \$\$ ND HULD 1080 63530 350 :: DISPLAY AT :24.1):*INCET.PLAY OR END"; 1076 DIs FPTs-CHR\$11201,3):: 076-DDF:(UDB/GUL 3)... 825=RPT51CHR\$(96),3):: PI 12 :: PLY=19 :: 605UB 180 1100 RETURN ´P(I≭ 1100 PETURN 1110 PET SE WHAT TO DO WITH WINN:WS? 1120 JILE JSO :: IF NOT HB THEM JISFLAY AT124,1): HOLD, 1130 BISPLAY AT124,7): GANBL E DR COLLET:: 1140 GUID 1070 1150 FEN ## SPIN GANBLE WHEE LS LS 1160 FOR T=7 TO 10 11 CALL H CHAF :,20,32):: CALL HCHAR(1 +,20,21):: WEIT 1 1170 CALL ST.N2(-4000,-7,29) 1180 F.3 1=;0 TO 7 STEP -1: 1180 P.- [1]; 0 / SIEP -1 : 2 CALL HCHAR(1,20,60):: CALL HC-AR(1+1,20,7):: NEIT 1 1199 FOR WD=1 TD 3 :: IF HFS :WC =NHS TREN C IF: AY AT(WPY WD-1,WP1(ND)): -- \$ 138); 150 HCT VD imb:.MPI(MD)::--6 13B); 1200 MEI VD :: f:- mD=1 TO 3 :: 1F HFs :mD:=mis THEN 1240 ::D: Fi=tHI(RMP100+1):: 1F FI'S THEN IF J._ F:(>O THEN v mD)=15 ELSE v:mD)=F1 :: 1 D: F:)=-1 :: 60TO 1240 17:: V(ND)=5-(F1>7)-(F(>10)-:F(-3)-(F1>23)-(F1>36)-(F1) .+F.:13-(F1)23)-(F1)36)-(F1) 49)-(F1)68)-(F1)871 1240 FOR DE=2 T0 300 :: NEIT DE :: DISPLAY AT(WPY(ND1,WP I(ND1):CHR*(138); 1250 CALL SOUND(-100,300,21) : CALL HCHAR(WPY(ND),NPI(ND) +2,WF(V(ND1)) 124 CALL SOUND(A000,-7,28); 12.05 CALL SDUND(4000,-7,29): : NEIT ND :: CALL SDUND(-1,-2,30):: RETURN 1270 REM TAKE CARE OF JOKERS

1300 JW=WD :: J=J+I :: IF JC CIVV)0 THEN 1320 1310 DISPLAY AT(JPY(VV), JPXI VV)):NJ\$;:: JCC(VV1=20 i: NJ =NJ+1 1320 WEIT WD :: DO=(NJ=0):: IF DO "--- DC=15 :: DISPLAY ATI2, :: "DOUBLE!"; 1330 RETURN 1330 RETURN 1340 REM 14 COMPUTE #1NW1K65 1253 HV=13 :: DN J+1 6073 13 67.1380,1360,1410 11:3 [F NOT(JA AND FY)10]THE N HV=FV 1370 6010 1410 1380 V(D)=V(3):: V(4)=V(1):: IF JA AND V(JW-1)()V(JW+11T HEN 1410 HEM 1410 1390 + 1 -15 :: V(4)=15 1400 :F V,JW(1)V(JW(1)THEN HV=V(JW(1)ELSE HV=V(JW(1) 1410 - F WD=1 TO 3 :: IF V(W D1<5 :HEW HVVIND)=HV ELSE HV V=D1=V=UMD) 1420 HEIT WD :: IF HVV(1)(3H V=V10R HVV(2)(3HVV(3)THEN 1 440 440 1430 IF JAC(HVV(1))IHEN JA=-1 11 JC=15 :: ECC_B 370 1440 FOR MD=1 IJ 3 :: IF HVV 1401=13 "HEN MI HUI 1455 MEIT MG :: IF JA THEN M 14-50 MC1 10 :: 17 JW INEN U 14-50 MS=-(HVV(1)=HVV(2))-(HV ::(+HVV(3)):: 17 NB)0 THEN W1=W1+R(MS,HVV(2)) 1470 IF DD THEN W1=28WI 1430 RETURN 1430 RETURN .490 REIURN .490 REM 188 GANBLE ROUTINE :5:0 DT=1 :: GV(1)=21M1 :: G V(2)=0 :: GV(3)=1NT(38M1/2): : GV(4)=1NT1W1/2) : GV(4)=INTIN/2) ISIO KEY=0 :: NG=NG+1 :: GOS UD ISO :: DISPLAY ATI24,11:* STP*: ISIO FOR LI=1 TD 3 :: CALL H CHAR(LI, 16, 96, 17):: NEIT LI 1510 RR=RR+1 :: IF RR)4 THEN 1240 RA-MARY ... RR-1 1340 SISPLAY AT(GPYIRRI, GPI(RR)):SI=4(GV(RR)); 1550 CALL SOUND(-4000, 150+50 IPP.21 1560 IF KEYCASCI'S'ITHEN CA LL KEY(GREY,CM):: GOTO .53 LL KEY(GREY,CM):: GOTO .53 1570 J1=11+R4151 SOT :: FUR DE=1 TO OT :: WEIT DE :: IF DT)ISO THEN .53 1660 DISPLAY AT(8,31: "GDT BA CK:";:: I AY AT(8,18):USI NG 1640: =: 4

1790 DN CA+(6010 1770,1800, 1820,2020 1800 WI=NI+1 :: IN=4 :: 605U 8 250 IBIO SDID 1770 IBIO 5DT0 1770 IB20 IN=2 :: 60SUB 250 IB30 50SUB .:: 1SPIN WHEELS 1840 IF D0 'FW 1860 IS30 5DSUB .:: 1SPIN WHEELS IS40 IF D0 'FW 1860 IS40 IF D0 'FW 1860 IS40 IF D0 IFW 1860 IS40 IFW 1870 1910 1900 DISPLAY AT(5,9):* YOU L UST* :: CALL SOUND(-600,200, 2):* EJED 1760 1910 212*4Y AT(5,9):*YOU NO TYLO ...: AT AICS, 7): YOU HU Y::: Display AICS, 17: W1:: FOR DU=1 TO WI :: CALL SOUN D(-50, 500, 1):: CALL SOUND(1, 500, 30):: HEI: DU 1720 IF HGL2 THEM HB=-1 1920 605UB 1120 !GET INSTRUC TION 1940 CA=-15ET=ASC1*H*)AND NO T HB1-28 SET=ASC(*G*11-3816E T=ASC *C*)) 1950 DN CA+1 6010 1930,2010, 1990, 1960 THE INFRESS BILL B 250 1980 6010 1760 1990 HB=-1 :: 6DSUB 1500 !6A THE THE WIND THEN 1910 ELSE .910 13 1 WI#0 :: 60TO 1760 1911 60SUB 1620 !END 1911 END

-11-



.

WEST PENN99ERS

% John F. Willforth R. D. # 1 BOX 73A JEANNETTE PA 1564

NOVEMBER 1988 ISSUE

NEXT MEETING TUESDAY NOV. 15th 7:00 PM at the UNITED PRESBYTERIAN CHURCH OF THE COVENANT 4th and Oak IRWIN, PA

TEXAS INSTRUMENTS HOME COMPUTER USERS GROUP PM CALL MAYS

00

Ě.

EDMONTON 99'ers P.O. BOX 11983 EDMONTON ALBERTA, CANADA, T 5J3L1