

HOOSIER USERS GROUP USERS GROUP HOOSIER USERS HOOSTER USERS GROUP USERS GROUP HOOSIER GROUP HOOSTER USERS HOOSIER USERS GROUP

THE HUGgers HOOSIER USERS GROUP People Helping People

HOOSIER USERS GROUP USERS GROUP HOOSIER **GROUP HOOSIER USERS** HOOSIER USERS GROUP USERS GROUP HOOSTER GROUP HOOSIER USERS GROUP HOOSIER USERS GROUP HOOSIER USERS GROUP HOOSIER USERS GROUP HOOSIER USERS

January/February 1993

The HUGgers Newsletter

Volume 13, Number 1

Secretary's report--

Wish to thank all whom helped get the newsletters out for 1993! I know from my roll in helping out, that it takes many hands to get this paper out each month, and month over! Thanks again to all concerni

Treasurer's Report ---

As the beginning of a new year for 1994, we as a club are doing a well in monies as could be for this time of year; however, for the BBS for DEC 93 we are paid for the cost of Tel bill, however, don't forget we still have the rest of the year to cover, either clubs expense or donations, as far as the newsletter expeses goes, we are on the clubs pay roll, unless we receive any donations as weclome always, but for now we ar OK. We will bring expense report to JAN 94 meeting soon to disscus this problem and try to find a good way to solve!

Again thanks for all for help this year!

SINCERELY,

Fred Edstrom, Jr SECT'Y-TREAS HUG 1-12-94

* ORIGIN: HUG TI99/4A BBS (317)-782-9942 / 8N1 / 24 hours

Reprint from a letter sent to our PO Box.

For sale by H. Warren are the following items:

TI994A Computer speech synthesizer joysticks p-box php1200 memory exp card php1260 disk memory dr php1250 disk memory system php1240 ti wrighter word processor personal record keeping personal report generator disk manager 2 ti ext basic parsec munch man connect four ti1050212-1 cable pcm 212-1 interface to connect to brother typewriter 99er Home Computer mag. from 1983 to 1984 manuals for all listed equipment

I may be reached at the following phone numbers: days 632-2914 nights 293-3004 or mail to me at the following address:

H. Dean Warren 3160 North Lawndale Ave Indianapolis , IN 46224

Fred Edstron, Jr Secretary-Treasurer

-- ANNOUNCING --

TI Fest West '94

A Texas Instruments 99/4A & MYARC Geneve 9640 Computer Users Convention

Location:

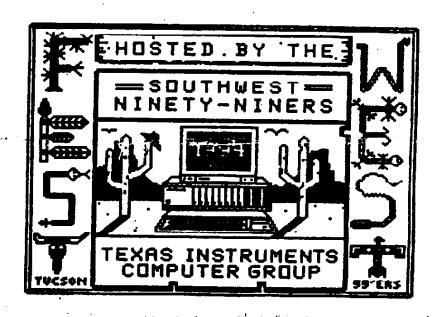
Park Inn, 88 E Broadway, Tucson, AZ 85701 Reservations: 1-800-437-PARK Welcome #TAZ143 (602)622-4000

Date/Time:

Sat Feb 19, 1994 - 9am to 5pm Sun Feb 20, 1994 - 9am to 3pm

Prizes
MYARC HFDC Controller
(w/20 meg Hard Drive)
Horizon RAM Disk 4000
(w/memory chips)
Asgard Memory System Card
Over/Under Floppy Drive
2400 BAUD Pocket Modem
Other Prizes to be Announced
Door Prizes, too!

Vendors
Software
Hardware
Accessories
Consignment Table
Speakers
Users Groups



Admission \$4 - under 15 free when accompanied by adult

TI Fest West '94 - Best in the West!

Presented By: .SouthWest Ninety-Niners PO Box 17831 - Tucson, AZ 85731

For more information about TI Fest West '94 or SouthWest Ninety-Niners User Group Call BJ Mathis - (602)747-5046 or Tom Wills (602)886-2460 Or Cactus Patch BBS (602)290-6277

Roll your own

Making that assembly program do what you want it to

By BOB CARMANY

How many times have you found an assembly language program that would almost do what you wanted it to do? Or, if it was just a few sectors smaller, it might fit on that last bit of space on your RAMdisk? That is exactly the problem that I faced several months ago.

I found that the USER dictionary for SPELLIT! worked much better if it was alphabetized so I used a rather ancient text sort program written by Dave Romer and John Clulow in the early '80s to sort the USER file each time I added a few words to it. It loaded as a program image file from Funnelweb and the sort was fairly rapid. In fact, it worked quite well and everything was fine until my USER dictionary contents approached 300 words which was the limit of the program. I was faced with the prospect of splitting the USER dictionary, sorting the two parts, and recombining the lot every time that I added some new words which was not my idea of an efficient method of operation!

A few other constraints cropped up along the way as well. The increased size of the F'WEB V.5.00 editors reduced the available space on my RAMdisk to 10 sectors (after moving the original text sort program elsewhere). So, a replacement had to accommodate more than 300 80-column records and fit into a 10-sector space. I also wanted it to operate out of the F'WEB environment and preserve the mailbox with the stored text file name. This was indeed going to be interesting!

I had just about given up on the project when I got my June 1993 copy of MICROpendium. Bruce Harrison's sort routine looked like it might fit the circumstances, but it would take a bit of "engineering" to get the job done. I thought of rewriting the source code entirely to take advantage of some of the internal F'WEB routines but there were some register contentions and the idea of going through 500-plus lines of source code didn't attract me. The best way to attack the problem, I thought, was a rewrite of selected sections of code and the addition of the suggestions that Bruce mentioned in the article. Over the space of the next couple of days, I laboriously typed in and checked the original source code from Sidebar 24.

The original program loaded into low memory and used the entire 24K of high memory to store the text string arrays. As written, the input prompts were off-center on the screen and there was no program attribution. It was a real "diamond in the rough." Bruce made the suggestion of adding appropriate "beeps" and "honks" in his article. Some other additions could be made, but that was hasically the program I started out with in the beginning.

Within these boundaries, I decided to try to customize his source code to something I could use within my system. I decided I would come up with a program image program as a final product to conserve space on my RAMdisk. Besides, program image files load much faster than object code.

(CONTINUED ON PAGE 4)

This newsletter is brought to you by the efforts of the officers and members of the Hoosier User's Group.

OPPINIONS EXPRESSED HEREIN ARE THE AUTHORS'S, AND DO NOT NECESSARILY REFLECT THOSE OF THE PUBLISHERS.

MEMBERS ARE ENCOURAGED TO SUBMIT ARTICLES FOR PUBLICATION...PLEASE!

REMEMBER
This is your user group too!

New S&T BBS

Hoosier User's Group Baud rate: 300/1200/2400 On Line 24 Hours Daily

317-782-9942 8W1

Now with 60 Megs Storage

Monthly meeting location White house next to St. Ann's School 2839 S. McClure Indianapolis, IN

Meeting starts at 2:00 pm January 16, 1993

mug officers (Area code 317)

President Warren Barnes 542-8568
Vice Pres Byrant Pedigo 255-7381
Secretary Fred Edstrom Jr 898-7300
Tresurer Fred Edstrom Jr 898-7300
Librarian Byrant Pedigo 255-7381

ROLL YOUR OWN—

Since we have gone through the preliminary constraints, let's go ahead and took at how the original program was altered. In cases where several lines of code were altered, the complete section of code will be reproduced. For the odd line or two, the tag, the line prior to the altered line and the altered line (or lines) will be reproduced. With this in mind, it is time to begin.

REF VMBW, VMBR, VSBW, VSBR REF DSRLNK, KSCAN, GPLLNK

GPLLNK was added to the REF block to accommodate the use of "beep" and "honk" sound effects as suggested in the original text by Bruce Harrison.

DEF START, SFIRST, SLAST

ACRG >2678 IRST B GSTART

SFIRST and SLAST were added to the DEF block so that the object code could be later converted to a program image file to reduce the size of the finished product. SFIRST branches to the start of the program

START LWPI Load user workspace " WS Set stack for high level autopatine LI R15.RINSIK ы RO.3 Row 1, column 4 LI R4.23*SCRWID 23 mas to clear - MINERO LINE BL. **OCLRFLD** Clear them Ы R1.TTTLNM Load title name - NEW LINE BL. **CDISSTR** Display it - NEW LINE ΑT RO, 3*SCRWID+4 Condensors and AMERICA LI . R1. INFSTR Set for input file prompt Display prompt 4DISSTR ÀΙ RO. SCRWID Down to row 2 ш R4,15 15 bytes CLR Clear R1 - NEW LINE R1.0>837C Move to GPL variage on - NW LINE MOVE BLWP. **OGPLLNK** Access GPLINK -- NEW LINE DATA >0034 Generate BREP - NEW LINE OCREIN : RL. . Use CRSIN subroutine R8.15 Has F-9 been struck PLCFN by the lift not, go on by the JNE.

The first altered line in this section of code effectively clears the screen down past any error messages that appear. Since the program restarts after an error, this was a necessity. The next line moves the prompt down and centers it. The first two new lines display "HARRISON SOFTWARE TEXT SORT" as a title and attribution. The data required for the message is at the end of the program with the other message. The next set of four new lines generates a "beep" at the "INPUT FILE" prompt. The relevant information is from page 251 in the Editor/Assembler manual.

Else quit

CEXIT.

PLCFN LI R9,TEMSTR Point at temporary string

BL @MOVSTR Movedation/TEMSTRoPABdan
LI R4,>A050 SaR4to-A050—ALTEREDLINE

Since I use SPELLIT! in conjunction with the word processor

in F'WEB, one of the self-imposed provisions was to avoid trasing the F'WEB mailbox and retain the default file name. Thus, the beginning address of the text string array storage was moved up to >A050 to avoid any conflict and keep everything intact. The slight reduction in storage space is scarcely missed.

OFNF1 LI RO, 12 *SCRWID+4 Set rev 12 column 5 - MINRO LINE
This line moves the "SORTING FILE" message down to the
middle of the screen and centers the text for easier reading

ADON LI RO.PAB1+5 Point at PAR+5 in VDP RAM

LI R10, >A050 Right at start of anay—AURHDLINE
This line points at the start address of the text array storage
which was changed earlier in the program to >A050.

CLR R1 Clear R1 — NEW LINE

NOVE R1,9>837C MOVE to GPL WORKSTAND — NEW LINE

BLWP GGPLINK Access GPLINK — NEW LINE

DATA >0036 Generate HONK — NEW LINE

MOV GENDSTR. R10 Point at the end of array

The first altered line moves the error message up a couple of lines from the original location and centers it on the screen for eater reading. The four new lines generate a "honk" sound immedately after the error message is displayed. (See page 251 in the E/A manual.)

GETOFN LI R0,3 - Set to row 1 column 3
LI R4,13*SCRWID 13 rows to clear - ALTERED LINE
AI R0,SCRWID+4 Cres down two rows - ALTERED LINE

CIR R1 Clear R1 - NEW LINE

MOVE R1,6>837C Move to GPL workspace - NEW LINE

ELMP 6GPLINK Access GPLINK - NEW LINE

DRTA >0034 Generate BEEP - NEW LINE

The first altered line clears the screen down past the "SORT FILE.." message since that chore has been finished at this point. The second line prepares for the display of the "OUTPUT FILE" prompt and centers it on the screen. As before, the four new lines generate a "beep" at the prompt prior to file name entry.

WRITP2 LI R9, > A050 Risk to the start of anay - AZERDLINE
Once again, we are avoiding trashing the F'WEB mailbox by
writing to a point above it.

EXIT BLMP 6>0 Return to XBASIC - MITERED LINE
The original program had the exit to the E/A cartridge which
worked just fine if you were operating the sort routine from there
I wanted it to exit to XBASIC where the AUTO program on m.
Quest RAMdisk would intercede and display the menu screen. It
made things a lot more convenient!

ROLL YOUR OWN---

FILER1	SLA R1,1	Double number in R1
	1 1 1	
	CLR RL	Clear R1 - NEW LINE
	MOVB R1,0837C	Move to GPL wordspace — NEV LINE
	BLWP SGPLLNK	Access GPLLNK - NEW LINE
	DATA >0036	Generate HONK - NEW LINE
•	BL SKEYLOO	Stop at keyloop
	LI RO, 20*SCRWID	Load RO for 21 rose - ACLESCO LINE
	1 1	

Once again we use the GPLLNK to generate a "honk" when an error is encountered, wait for a keypress, and then clear the error message before restarting the program. The altered line clears the first 21 rows.

```
CRIX B SSUBRET Return to SUBRET
```

This bit of text was omitted in the original but gleaned from Sidebar 5 where it appeared. This is a dummy to keep all of the JUMP instructions within range.

EDGE BYTE >2B B/ts fix init/timit etty-AZHDINE
I did not care for the use of a non-displayable character as the input/output prompt delimiter so it was changed to >2B — the "+" sign for my personal preference.

```
LEFTV BYTE 8 Left arrow
TITIAM BYTE 27 Length byte - NEW LINE
TEXT HARRISON SOFTWARE TEXT SORT'
```

The TTTLNM message displays the appropriate attribution at the top of the screen when the program starts.

SLAST END

SLAST was added to accommodate the production of a program image file.

With all of the altered lines and new lines stuffed in their appropriate places, we are ready to assemble the source code. If you haven't made any errors you should get a "<0000>" ERRORS message — and a 29-sector piece of object code. At this point, the program will run from the B/A cartridge under the name "START". If this is as far as you want to go with it, change the last line in the program to:

```
SLAST END START and the program will auto-run from E/A.
```

We were going for something a bit smaller than what we have created so far. The 29-sector object code file was too big to fit in the 10-sector space I had left on my RAMdisk but it should scrunch down nicely into a program image file.

One major obstacle to overcome was the fact that the program was AORGed into low memory. TI's SAVE utility will not handle a program that has been AORGed into low memory — but FSAVE that comes with F'WEB will!

To get the final product, enter F'WEB, load the object code file from LOADERS and load DSKx.FSAVE from the appropriate disk drive. Press PCTN-3 at the prompt and then ENter. Using the "S" and "D" keys, move the cursor to SAVE and press PCTN-6. Follow the on-screen prompts and the file will be saved in program image format. The result is a nine-sector file that will run only from F'WEB.

All of this resulted in a lively correspondence with Bruce Harrison, who suggested another way to create an E/A 5 general purpose program image file. First, remove the GPLLNK from the REF block and add the following source code lines immediately after MOVBTS:

```
* GENERAL PURPOSE GPL LINK

* BY DOUG WARREN/CRAIG MILLER
```

```
CR4
          BOU GPLMS+8
GR6
          EOU GPLWS+12
STRPNT
           EOU >8373
LOGADO
          ECU >60
XTAB27
          EQU >200B
GETSTA
          BOU >166C
GPLLINK
          DATA GLAKWS
           DATA GLINKI
RTNAD
           DATA XMLRTN
COMLAD
          DATA >176C
           DATA >50
GLNRWS
           BCU $->18
           BSS >08
           MOV *R11, @GR4
GLINK1
           MOV *R14+6GR6
           MOV GXTAB27, R12
           MOV R9, 0XIAB27
           LMPT CPLMS
           Ħ.
                *R4
           MOV GGRMLAD, 6>8302 (R4)
           INCT SSIKPNT
                GLACIADO
XHLRTN
               OGETSTK. R4
           MOV
           LMPI GLAKME
          MOV R12.0XTAR27
```

If you have a shorter general purpose GPLLNK routine (there are several around) you can substitute it for the previous source code.

Save the assembled object code as before with FSAVE, Bruce used the file name SORTEM for his program image file but you can pick another name if you wish (e.g. SORT1). Then type in the following source code as a second file. This is a tip from Merie Vogt via Bruce Harrison for stasting the E/A utilities in a second, auxiliary file.

```
DEF SFIRST, SLAST, SLOAD
SFIRST BQU >2000
SLOAD BQU >2000
SLAST BQU >23BA
```

Assemble this bit of source code into object code and use TI's SAVE utility to produce a program image file. Use the file name SORTEN for this one if you used the same name that Bruce chose

(CONTINUED ON PAGE 6)

ROLL YOUR OWN—

(CONTINUED FROM PAGE 5)

for his first file name. Otherwise, increment the last character of the first file by one for the second file name (i.e., SORT1 and SORT2, etc.)

Use an appropriate disk sector editor to change the first byte of the SORTEM file from >0000 (no more file s to load flag) to >FFFF (additional file load flag) and you have two files SORTEM and SORTEN (or SORT1 and SORT2) that will load from any E/A option 5 loader. The only cost is an additional five

sectors of disk space required for the second file. Essentially, you have loaded the required E/A utilities into low memory from >2000 to >23BA.

Incidentally, you can use this second procedure for any file that loads into low memory that you want to convert to a general purpose E/A-5 program image file.

The final word is that you can change existing source code to fit into your own system and work the way that you want it to work. It just takes a little time and a little effort to get the job done!

RDB ENTERPRISES 643 Fair Ave. Shelbyville, IN 46176

TI99/4a consoles \$15 with power	upply and m	odulator \$2
TI Peripheral expansion boxes wit	h one disk	drive, disk
controller card, memory expansion	and interfa	ce card \$85
Empty P-boxes		\$25
TI disk controllers		\$25
Memory expansion cards		
Console interface cards		\$35
1/2 height disk drives black face	 .	\$25
Full " " drives		\$15
New replacement tan keyboards		
New TI joysticks		

The WATCHAMCALLIT for those that have the Myarc HFDC with clock promblems on their TI's, this is the solution\$27.95

I can also locate almost any kind of computer cables and also I can try to locate new or used computer equipment for the TI99/4a or any other kind of computer if you let me know what you are looking for and what you are willing to pay.

I work during the day so you can call 1-800-464-8851 and leave your name and number for availability on any equipment or call 1-317-392-2312 after 6:00pm for anything else. I will get in touch with you as soon as possible.

Thanks for your support over the last year and My family and I wish you a Merry Christmas and a Happy New Year.

CS1*FINDEXAN AUTOMATIC CASSETTE TAPE PROGRAM

LOCATION SYSTEM a review by Charles Good

This one is for cassette tape users and for those interested in unusual programming techniques. Have you ever wondered if it was possible to mark with software the position of a specific program on a cassette tape full of many programs and then have the computer search the tape from the beginning until the specific desired program is found? TI did once develope such a system for its 99/8 computer. but TI's WAFERTAPE drive was never released. Coleco ADAM computers sccessfully use such a system. Not so for the TI99/4A, according to many well respected comentators. I have read again and again in our exchange newsletters expert comment to the effect that with the TI there is no way to automatically, under software control, advance a long cassette tape to the exact physical location where a program starts. Well...., way back as early as 1983 Joseph E. Bartle of Parish NY wrote a TI BASIC program that does this for the TI! I recently acquired a copy 1985 update of Joe's CS1*FINDEX program (still entirely in TI BASIC with no assembly routines) and after removing a few bugs I am quite impressed with capability of this software.

CS1*FINDEX will do its stuff even if you don't have a printed list of which programs are on a program tape, even if you are using a tape recorder that does not have a numerical tape counter, and even if you are using a tape recorder that is not automatically controlled on/off by the 99/4A. CS1*FINDEX finds semicutomatically the exact location of a program on a long tape. The manual tape recorder operations required of the user are all prompted from the screen. If you are using a TI compatible recorder, CS1*FINDEX will advance the tape to your program's location after you press fast foreward, and then automatically stop the tape. If you are using a tape recorder that the TI cannot automatically turn on and off,

CS1*FINDEX will turn the screen from green to yellow and finally to red to indicate when you should manually press cassette STOP once the location of your program has been reached. Neat!

With CS1*FINDEX you can create a catalog of up to 10 programs you want to put on one side of a C60 tape and put this catalog at the beginning of the tape. The catalog includes program name (up to 12 characters with spaces anywhere), and there is also provision for catalog to display a 12 chracter comment for each of the 10 programs. You can then put your up to 10 programs onto the tape, with CSI*FINDEX advancing the tape recorder to the correct tape location where you should SAVE CS1 each program. It is necessary to reload CS1*FINDEX for each of the programs you put on the tape. Thus, users with only a console/cassette system will appreciate the fact that CS1*FINDEX is designed to be small enough to load into the MINIMEMORY module with SAVE MINIMEM. Then each time you need to load CS1*FINDEX, all you do is type OLD MINIMEM, and CS1*FINDEX boots in a few seconds. Otherwise it takes about 90 seconds to load CS1*FINDEX from tape.

Later, when you want to use the tape you load CS1*FINDEX into the computer and then load the tape's catalog from CS1*FINDEX. From the catalog display you select the number of the desired program on the tape. You are then instructed to rewind the tape to the beginning and press FAST FOREWARD. CS1*FINDEX then advances the tape to the program's location, automatically stops the tape if you are using a TI compatible recorder. displays the name of your program on the screen, and informs you this program has been located. Then CS1*FINDEX BREAKs to command mode and allows you to load your program in the normal way by typing OLD CS1 and following all the usual screen instructions, except that you DO NOT again "rewind cassette tape". CS1*FINDEX can easily be modified in extended basic to load the located tape program into the computer from within CS1*FINDEX rather than from command

mode. Change line 1770 to read RUN "CS1".

If you already have a printed list of each program on the tape and in which order the programs occur, you can bipass the catalog loading procedure. When you RUN CS1*FINDEX your first option is "LOCATION SEARCH (Y/N)". From here you can use CS1*FINDEX to locate the first or second or third, etc, program on the tape without using time to boot the catalog.

What's the secret? How does CS1*FINDEX using only TI BASIC with no assembly routines do what all the experts say can't be done? Have you ever noticed how the tape recorder behaves when you read or write tape serial FILES (as opposed to PROGRAMS)? The recorder starts, reads in or writes what I presume to be a file header, then stops. Then the recorder starts again and reads or writes the first record and then stops. Then the recorder starts again and reads or writes the second record and then stops, etc, etc. The total number of start/stop cycles equals the number of records plus one. The computer controls the turning on and off of the tape recorder motor and IT DOESN"T MATTER TO THE COMPUTER IF THE RECORDER IS SET FORPLAYOR FORFAST FOREWARD. When searching for a program, CS1*FINDEX writes a false file to the tape, turning the tape recorder motor on and off several times as this file is written. The tape recorder is set for FAST FOREWARD rather than for RECORD as this file is written, so the tape never receives any data. The computer cannot directly sense that the tape is not getting any data, so the computer continues to turn the recorder motor on and off as it writes its fake file to the tape. When turned on, the tape advances very rapidly because the recorder is set for FAST FOREWARD. A tape file designed to write up to 10 records with a record length of 192 will go through up to 11 start/stop sequences on a C60 tape before the tape is completely wound up on the take up reel. This is how CS1*FINDX locates physical blocks of tape space in which to insert programs, and can later find

a specific program located at any one of these physical blocks of tape space. The first block (corresponding to the false file's header) is where the catalog is stored, and the next 10 blocks (each corresponding to a false file record) are where the programs are stored. Enough space is included in each of the program storage blocks to store the largest possible tape PROGRAM.

LIMITATIONS: 1--You can't use CS1*FINDEX with already existing program filled tapes. The spacing of the programs on the tape won't be right. You need to load programs onto your program storage cassette tapes using CS1*FINDEX. 2--Problems may occur if different tape recorders are used to store and later play programs. If the FAST FOREWARD speed of the two recorders differs very much CS1*FINDX will not correctly find the location of the desired program. 3--There is only room for a short program in the last (10th) program block before the tape runs out.

The author of CS1*FINDEX has written some rather wordy documentation files to explain the use of CS1*FINDEX. These files are in PROGRAM format so that they can be loaded from tape and read by console/cassette-only users. In general most users can play around with the program and figure out how to use it without these docs. A sample tape program finding catalog is printed below as is the CS1*FINDEX program listing (checksums added using EZ-KEYS PLUS) with permission of the author Joseph R. Bartle. It is released to the TI community as FAIRWARE. If you like it, send whatever you think it is worth to Joe at the address in the REM statements at the beginning of the program. Joe has other fairware offerings. Write or call him for details. User groups, not individuals, may obtain a copy of CS1*FINDX and the above mentioned doc files by sending a disk and paid return mailer to the Lima User Group, P.O. Box 647, Venedocia OH 45894



FOR THE GAMES NUTS THE SEGA MODEL 3020



CONTROL PAD

For the games nuts amongst you, our current 'GOLDEN JOYSTICK' trophy holder advised me not too long ago that the SEGA Control pads will operate with the Atari 2600 Games machine — it appears that he and his mates have been carrying out some serious experiments of late. This got me thinking that if the ATARI joysticks worked on a TI — 99 / 48 with the aid of a suitable adaptor, then maybe these 'Control Pads' would too

Well guess what....they do

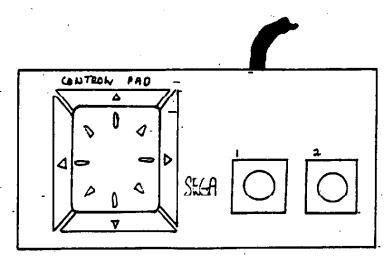
The problem is that the #2 fire button doesn't work, but who needs two fire buttons ? I certainly don't ?

I am told by the games experts that the control pad is a superior device to the joystick that gives better response and control, and who am I to argue? This kid won the last Golden Joystick competition playing left - handed, as he had broken his good wrist in a skate - boarding accident on a school camp and was in plaster up to his right elbow For the record, the device is called a SEGR MODEL 3020 CONTROL PAD, and it looks something like the illustration below

It retails at Target, BIG W, K Mart and Myer stores here in Western Australia for approx. \$20.00

Geoff WARNER

P.S. I told President Merv of my discovery. His reply.. Yup ! The grand - kids told me about it



A CONTRACTOR OF THE RESERVE OF THE STREET OF	approximate and a second region of the second secon	
	I DHONE () –	100.11 12.50
q17	YTIO	duly April 817,00 fully April 814,00 F April 814,00 F April 814,00
# 1dA	I L ADDRESS I	New membership is prorated as follows: May April 620,00
S *YAGDT 3TAG	I NAME	Annual membership is sig.00
Tine	יים בתב סער מער מער מער מער מער	359310

HOOSIER USERS GROUP P.O. Box 2222 Indianapolis, IN 46206-2222

Make check or money order payable to Hoogier Users Group. Send completed application to:

Below you will find an application for membership to the Hoosier Users Group. Active membership entitles you to the Newsletter, up and download on the HUCbbs, attendance and voting rights at regular club meetings, access to the HUCger Library of Programs, special club activities and special guest speakers for one year.

APPLICATION FOR MEMBERSHIP



HOOSIER USERS GROUP P.O. Box 2222 Indianapolis, IN 46206-2222

Forwarding and Address
Correction Requested



TIME DATED
JANUARY 18.1884
NATERIAL

May 1774

Dan H. Eicher 2720 Palo Verde Ct. Indianapolis, IN 46227

Hoosler User's Group S&T BBS 317-782-9942 300/1200/2400 8N1