

COMBINED FOUNDERS DAY MEETING by FRANK LARRICK

TODAYS MEETING WAS A SPECIAL COMBINED MEETING FOR THE PURPOSE OF CELEBRATING OUR 8th ANNIVERSARY. THE MEETING WAS CALLED TOGETHER AT APPROXIMATELY 12:15 PM BY CO-PRESIDENTS RICK KELLOGG (DAYTON) AND SAM MOON (CINCINNATI). RICK INTRODUCED ALL THE OTHER OFFICERS ATTENDING. THEY WERE: FRANK LARRICK (DAYTON SECRETARY), ERIC SAUNDERS (NEWSLETTER EDITOR) AND ERIC BISHOP (DAYTON LIBRARIAN). KEN CARPENTER (CINCINNATI TREASURER), CAROLYN CARPENTER (CINCINNATI LIBRARIAN) AND BILL STAGER (DAYTON TREASURER) WERE NOT ABLE TO ATTEND. FOUNDING MEMBERS UNABLE TO ATTEND TODAY WERE KATHY BAIER AND HAROLD PARSHALL. FOUNDING MEMBERS ATTENDING WERE ED YORK (PAST PRESIDENT), JAY HIERS (PAST LIBRARIAN), LARRY MORROW AND JIM SCHWALLER. PAST OFFICERS ATTENDING TODAY WERE HERB KLINE, DONNA KLINE, BILL POST, RICHARD WHITMER AND ERIC COSTELLO. THE INFAMOUS GROUP DISCLAIMER WAS DISPENSED WITH FOR THIS MEETING BECAUSE OF THE SPECIAL NATURE OF THIS MEETING AND OUR WISH TO MAKE IT OPEN TO THE PUBLIC. A MOTION WAS MADE TO ACCEPT THE SUMMARY OF THE MINUTES OF THE LAST MEETING CONTAINED IN THE NEWSLETTER AS AN OFFICAL READING OF THOSE MINUTES. THE MOTION WAS SECONDED AND PASSED BY VOICE VOTE. BECAUSE OF THE INFORMAL NATURE OF THIS MEETING ONLY A FEW ITEMS WERE COVERED UNDER HEADING OF OLD BUSINESS. THEY WERE: GENERAL INFORMATION CONCERNING THE UPCOMING "LIMA FAIRE" TO BE HELD ON MAY THE 20th IN LIMA, OHIO. DOORS OPEN AT 9 AM. AND ADMISSION IS "FREE". MANY OF THE SOFTWARE AUTHORS AND HARDWARE VENDORS STILL SERVICING THE "TI" WILL BE THERE. THERE WILL INFORMATIONAL MEETINGS STARTING EACH HALF HOUR AND COVERING MANY ASPECTS OF "TI" PROGRAMING AND HARDWARE. THIS FAIRE RIVALS IF NOT SURPASSES THE CHICAGO FAIRE. "IT COMES HIGHLY RECOMMENDED". RICK DELIVERED THE MONTHLY FINANCIAL REPORT LISTING INCOME, EXPENCES AND ENDING BALANCE. "MICROPENDIUM" MAGAZINES WERE MADE AVAILABLE TO THOSE WISHING TO PURCHASE THEM AT A REDUCED COST. THIS IS A PROGRAM MADE AVAILABLE TO USER GROUPS BY THE PUBLISHER. UNDER THE HEADING OF NEW BUSINESS THE FOLLOWING OCCURED: RICK FORMALLY INTRODUCED THE CHARTER MEMBERS TO THE GROUP. HE GAVE TO THESE INDIVIDUALS "CERTIFICATES OF APPRECIATION" TO ACKNOWLEDGE THEIR CONTRIBUTION TO THE FORMATION OF THE "CIN-DAY USER GROUP" EIGHT YEARS AGO. LARRY MORROW, JIM SCHWALLER, JAY HIERS AND

ED YORK TOOK POSITIONS IN FRONT OF THE GROUP WHERE THEY RECEIVED THEIR CERTIFICATES AND A ROUND A APPLAUSE FROM THE MEMBERS. CERTIFICATES WILL BE SENT TO THE TWO FOUNDING MEMBERS NOT ABLE TO ATTEND TODAYS MEETING. JIM SCHWALLER TOOK THE FLOOR AND WITH THE HELP OF THE OTHER FOUNDING MEMBERS GAVE A BRIEF HISTORY OF THE FOUNDING OF THE GROUP ON MAY 3rd, 1981. WE MEMBERS OF THE "CIN-DAY USER GROUP" AGAIN WISH TO EXPRESS OUR GRATITUDE TO OUR FOUNDING MEMBERS FOR THE FINE START THEY GAVE US AND THE DIRECTION THEY STARTED US IN. "WE THANK YOU" FOUNDING MEMBERS. AT THIS POINT WE TOOK A FIFTEEN MINUTE BREAK DURING WHICH THERE WAS GENERAL SOCIALIZING. ALSO DURING THIS TIME RAFFLE TICKETS WERE SOLD. PRIZES THIS MONTH WERE: A COPY OF "HOME PUBLISHING ON THE 99/4A", A PRINTER STAND, A DISK STORAGE BOX FOR 75 DISKETTES, COLLECTIONS OF "SUPER 99 MONTHLY", "BOSTON COMPUTER SOCIETY" NEWSLETTERS, "SMART PROGRAMER", "R & D COMPUTING" AND AUSTRALIAN (AND OTHER) NEWSLETTERS. WE WISH TO THANK DAVE ROSE AND BILL POST FOR DONATIONS TO TODAYS PRIZES. DURING THIS PERIOD CAKE, COFFEE AND SOFTDRINKS WERE AVAILABLE TO ALL AND HELPED MAKE FOR A FESTIVE ATMOSPHERE FOR ALL. AS PART OF THIS MONTHS MEETING WE HELD A SWAP MEET DURING OUR REGULAR MEETING. IT WAS OPEN TO THE PUBLIC AND SEEMED TO BE ENJOYED BY ALL. AS A SPECIAL OFFER FOR THE FOUNDERS DAY MEETING OUR LIBRARY WAS ALSO OPEN TO THE PUBLIC AND ALL DISKETTES WERE PRICED AT ONE DOLLAR. DEMONSTRATED AT THIS MONTHS MEETING WAS A SYSTEM CONSISTING OF A MODIFIED CONSOLE WITH THE "RAVE" KEYBOARD AND THE EXPANSION BOX CONTAINING A "MYARC" 512K MEMORY CARD, THE "DIJIT SYSTEMS" (AVPC) 80 COLUMN CARD WITH AN RGB COLOR MONITOR, THE "RAVE" SPEECH MODULE ADAPTOR CARD AND THE "MYARC" HFDC CARD USING ONE DSDD 5 1/4 DRIVE AND ONE 20 MEGABYTE HARD DRIVE. THE DEMO ILLUSTRATED THE SYSTEMS SPEED, VERSATILITY AND GREAT GRAPHICS CAPABILITIES ON THE RGB MONITOR. RICK HANDED OUT APPROXIMATELY 50 MISC. NEWSLETTERS COLLECTED BY OUR NEWSLETTER EDITOR. HE GAVE A SHORT REPORT ON THE GROWTH OF THE GROUP, THE STATUS OF THE NEWSLETTER, OUR VARIOUS LIBRARIES AND OUR FUTURE DIRECTION. HE ALSO ANNOUNCED THE DONATION OF MODEMS TO THE GROUP BY JOHN NEESE. THESE MODEMS WILL BE MADE AVAILABLE TO MEMBERS ON A MONTHLY RENTAL BASIS. ALSO DISCUSSED WERE THE FUTURE NEWSLETTER ARCHIVES THAT WILL BE AVAILABLE TO MEMBERS AND THEIR POSSIBLE COMPLETION DATE. THE MEETING ADJOURNED AT APPROXIMATELY 3:30 PM.

NO GRAPHICS 516 PRIOR TO 7/15 MEETING

Tech Talk (Exerpts)  
 by Mike Maksimik  
 (Chicago Times, Dec 88)

Project #1: Many of you know that I interfaced a mouse to my TI. No it wasn't that expensive Mechatronics mouse, but the inexpensive Radio Shack color mouse. This mouse is a simple potentiometer mouse with one or two buttons, depending on the model, and costing in the \$20 to \$40 range. One component of the system is a clock card. This particular card is a relatively rare one called FIRST ADE and is a clock/analog-to-digital converter/ digital I/O card. The devices you can interface to this card are endless, but I chose a mouse because TI-Artist allows a custom DSR for any device to be created for that program. First, I discovered how to interface the mouse to the analog channels and then how to quickly scan the channels for input. I added some small capacitors to compensate for the 'divider effect' which caused the potentiometers to jump from the negative side over zero to the positive side. This left a nasty gap but adding 4.7 microfarad electrolytic capacitors between the analog input and analog ground made the transition smoother. Tanalums were not working, they took too long to stabilize - this eventually caused the mouse to gain a 'too fluid' motion. Eventually I will reduce them to the limit where the response is the quickest. Perhaps the ones in the Super Sketch cartridge are the ones to use??!! I implore you, dear readers, this project is disgustingly simple and I URGE you to try it.

Now, on to other things. I recently took a systems programming class at school, and got a great idea on using the interrupt routine to control a multitasking environment on the TI. You know that when interrupts are enabled, such as in TI-BASIC and EX-Basic environments, the VDP chip (TMS 9918A video display processor) acts a a cute little 60Hz interrupt timer, and will interrupt the running program. The interrupt routine then resets that condition, then does some minor things like cassette timer update, sound chip update, peripheral interrupt check, and then it looks at [the] quit key, if enabled, and any other system interrupt. After all of that (except peripheral interrupt) the routine branches to the user interrupt routine. To get multitasking to work, this routine must reside in a "safe" area, such as a supercart, or in low memory expansion below the free address pointers.

What this routine will do is this: maintain a linked list of workspace pointers, status registers, and return addresses for the number of routines that are loaded into memory. This routine then, on each interrupt, saves the SAVED WS [workspace pointer], PC [program counter], and ST [status register] in the appropriate place in the linked list.

Then the interrupt workspace R13, R14, and R15 are loaded with new values next in the linked list. When the RTWP from interrupt occurs, the process that executes now is the next one in the linked list. 1/60th of a second later, this repeats itself and so on until the computer looks like several things going at once. There is a simple way to keep the system intact. First, the KERNEL which does the task switching must do a LIMI 0 instruction so it doesn't get interrupted. Each process which wishes to write to VDP must do it all at once (using LIMI 0) or keep track of there VDP address it wants in a WS register. Sound processing must occur from VDP so mutual exclusion must be practiced. The KERNEL handles sound, and each routine must request for sound. The same goes for disk I/O. I have four drives and 4 RS232 ports. Wouldn't it be nice to assign each user his/her drive? It can be done. The KERNEL will only read a sector at a time for each user, and RAMDISK usage makes access even faster. The next consideration is memory. Well, I'm sure there is a RAMDISK out there that can be used as a bank-switched upper 24K portion four times over. The KERNEL could bank-switch information at the same time it changes the RTWP information. The RS232 interrupt routines can be used to provide each user with virtually unlimited keyboard queue. Imagine a 5th or 6th task running at the same time. Idling until activated as print spoolers. The 'nice' terminal emulators use the same method to print incoming input to a parallel printer, so why can't this system? finally, restrictions on the user's programs, much like those found on a VAX or IBM PC must be imposed, such as I/O routines, memory usage, etc. The KERNEL in order to be fast, can not check for these so it is up to the programmer to keep his/her program in line. I hope some of you can grasp what this means. It may be a nice DEMO to show off, but imagine this:

- 4 terminals or modems on-line, connected to all 4 RS232 ports,
- 2 parallel printers,
- 4 disk drives (80 track MYARC of course),
- 2 hard drives,
- 2 RAM expansion cards at 512K each in 4 banks switchable by CPU operations,
- Supercart or Supercart II with 32K console, and expansion box.

You may be thinking DREAM ON MIKE, but this would be SOMETHING!. Hey, as long as we keep thinking about things like this, will the TI ever die.

[CIN-DAY Editor's Note: A lot of what Mike wrote more or less soared over my head, but the idea is fascinating and I wanted to share it with all the TIers out there. What do you think? Let's hear some comments.]

THE LIMA MULTI-USER GROUP CONFERENCE WAS A GREAT EVENT, BETTER THAN LAST YEAR!! THANKS, LIMA USER GROUP FOR ALL YOUR EFFORTS.

LET'S HEAR ABOUT ALL THE NEW SOFTWARE, HARDWARE, AND MISCELLANEOUS ITEMS YOU PICKED UP AT THE SHOW.

SEND IN YOUR REVIEWS, OPINIONS, COMMENTS, TRIALS, AND ERRORS TO SHARE WITH ALL THE MEMBERS OF THE CLUB.

THANKS LIMA U.G. !!

CIN-DAY News June 1989

MAY MINUTES .....	1
TECH TIPS .....	2
ALC #2: NUMBER SYSTEMS .....	3
DECIMAL-HEX-BINARY CHART .....	4
TI-BASE TUTORIAL 3 ..	5
GRAPHICS COMPATIBILITY ..	9
LET'S TALK RAMDISK ..	11



REPRINTED FROM OZARK 99'ER NEWS OCT 88

**APPENDIX B. DECIMAL-HEXADECIMAL - BINARY CONVERSION TABLE**

Dec	Hex	Binary	Dec	Hex	Binary	Dec	Hex	Binary	Dec	Hex	Binary
0	00	0000 0000	64	40	0100 0000	128	80	1000 0000	192	C0	1100 0000
1	01	0000 0001	65	41	0100 0001	129	81	1000 0001	193	C1	1100 0001
2	02	0000 0010	66	42	0100 0010	130	82	1000 0010	194	C2	1100 0010
3	03	0000 0011	67	43	0100 0011	131	83	1000 0011	195	C3	1100 0011
4	04	0000 0100	68	44	0100 0100	132	84	1000 0100	196	C4	1100 0100
5	05	0000 0101	69	45	0100 0101	133	85	1000 0101	197	C5	1100 0101
6	06	0000 0110	70	46	0100 0110	134	86	1000 0110	198	C6	1100 0110
7	07	0000 0111	71	47	0100 0111	135	87	1000 0111	199	C7	1100 0111
8	08	0000 1000	72	48	0100 1000	136	88	1000 1000	200	C8	1100 1000
9	09	0000 1001	73	49	0100 1001	137	89	1000 1001	201	C9	1100 1001
10	0A	0000 1010	74	4A	0100 1010	138	8A	1000 1010	202	CA	1100 1010
11	0B	0000 1011	75	4B	0100 1011	139	8B	1000 1011	203	CB	1100 1011
12	0C	0000 1100	76	4C	0100 1100	140	8C	1000 1100	204	CC	1100 1100
13	0D	0000 1101	77	4D	0100 1101	141	8D	1000 1101	205	CD	1100 1101
14	0E	0000 1110	78	4E	0100 1110	142	8E	1000 1110	206	CE	1100 1110
15	0F	0000 1111	79	4F	0100 1111	143	8F	1000 1111	207	CF	1100 1111
16	10	0001 0000	80	50	0101 0000	144	90	1001 0000	208	D0	1101 0000
17	11	0001 0001	81	51	0101 0001	145	91	1001 0001	209	D1	1101 0001
18	12	0001 0010	82	52	0101 0010	146	92	1001 0010	210	D2	1101 0010
19	13	0001 0011	83	53	0101 0011	147	93	1001 0011	211	D3	1101 0011
20	14	0001 0100	84	54	0101 0100	148	94	1001 0100	212	D4	1101 0100
21	15	0001 0101	85	55	0101 0101	149	95	1001 0101	213	D5	1101 0101
22	16	0001 0110	86	56	0101 0110	150	96	1001 0110	214	D6	1101 0110
23	17	0001 0111	87	57	0101 0111	151	97	1001 0111	215	D7	1101 0111
24	18	0001 1000	88	58	0101 1000	152	98	1001 1000	216	D8	1101 1000
25	19	0001 1001	89	59	0101 1001	153	99	1001 1001	217	D9	1101 1001
26	1A	0001 1010	90	5A	0101 1010	154	9A	1001 1010	218	DA	1101 1010
27	1B	0001 1011	91	5B	0101 1011	155	9B	1001 1011	219	DB	1101 1011
28	1C	0001 1100	92	5C	0101 1100	156	9C	1001 1100	220	DC	1101 1100
29	1D	0001 1101	93	5D	0101 1101	157	9D	1001 1101	221	DD	1101 1101
30	1E	0001 1110	94	5E	0101 1110	158	9E	1001 1110	222	DE	1101 1110
31	1F	0001 1111	95	5F	0101 1111	159	9F	1001 1111	223	DF	1101 1111
32	20	0010 0000	96	60	0110 0000	160	A0	1010 0000	224	E0	1110 0000
33	21	0010 0001	97	61	0110 0001	161	A1	1010 0001	225	E1	1110 0001
34	22	0010 0010	98	62	0110 0010	162	A2	1010 0010	226	E2	1110 0010
35	23	0010 0011	99	63	0110 0011	163	A3	1010 0011	227	E3	1110 0011
36	24	0010 0100	100	64	0110 0100	164	A4	1010 0100	228	E4	1110 0100
37	25	0010 0101	101	65	0110 0101	165	A5	1010 0101	229	E5	1110 0101
38	26	0010 0110	102	66	0110 0110	166	A6	1010 0110	230	E6	1110 0110
39	27	0010 0111	103	67	0110 0111	167	A7	1010 0111	231	E7	1110 0111
40	28	0010 1000	104	68	0110 1000	168	A8	1010 1000	232	E8	1110 1000
41	29	0010 1001	105	69	0110 1001	169	A9	1010 1001	233	E9	1110 1001
42	2A	0010 1010	106	6A	0110 1010	170	AA	1010 1010	234	EA	1110 1010
43	2B	0010 1011	107	6B	0110 1011	171	AB	1010 1011	235	EB	1110 1011
44	2C	0010 1100	108	6C	0110 1100	172	AC	1010 1100	236	EC	1110 1100
45	2D	0010 1101	109	6D	0110 1101	173	AD	1010 1101	237	ED	1110 1101
46	2E	0010 1110	110	6E	0110 1110	174	AE	1010 1110	238	EE	1110 1110
47	2F	0010 1111	111	6F	0110 1111	175	AF	1010 1111	239	EF	1110 1111
48	30	0011 0000	112	70	0111 0000	176	B0	1011 0000	240	F0	1111 0000
49	31	0011 0001	113	71	0111 0001	177	B1	1011 0001	241	F1	1111 0001
50	32	0011 0010	114	72	0111 0010	178	B2	1011 0010	242	F2	1111 0010
51	33	0011 0011	115	73	0111 0011	179	B3	1011 0011	243	F3	1111 0011
52	34	0011 0100	116	74	0111 0100	180	B4	1011 0100	244	F4	1111 0100
53	35	0011 0101	117	75	0111 0101	181	B5	1011 0101	245	F5	1111 0101
54	36	0011 0110	118	76	0111 0110	182	B6	1011 0110	246	F6	1111 0110
55	37	0011 0111	119	77	0111 0111	183	B7	1011 0111	247	F7	1111 0111
56	38	0011 1000	120	78	0111 1000	184	B8	1011 1000	248	F8	1111 1000
57	39	0011 1001	121	79	0111 1001	185	B9	1011 1001	249	F9	1111 1001
58	3A	0011 1010	122	7A	0111 1010	186	BA	1011 1010	250	FA	1111 1010
59	3B	0011 1011	123	7B	0111 1011	187	BB	1011 1011	251	FB	1111 1011
60	3C	0011 1100	124	7C	0111 1100	188	BC	1011 1100	252	FC	1111 1100
61	3D	0011 1101	125	7D	0111 1101	189	BD	1011 1101	253	FD	1111 1101
62	3E	0011 1110	126	7E	0111 1110	190	BE	1011 1110	254	FE	1111 1110
63	3F	0011 1111	127	7F	0111 1111	191	BF	1011 1111	255	FF	1111 1111

Figure B-1. Decimal-Hexadecimal-Binary Conversion Table

**TI-BASE - From INSCEDOT  
TUTORIAL 3 By Martin Smoley  
NorthCoast 99'ers - Oct. 1, 1988  
Copyright 1988 By Martin A. Smoley**

I am reserving the copyright on this material, but I will allow the copying of this material by anyone under the following conditions. (1) It must be copied in its entirety with no changes. (2) If it is retyped, credit must be given to myself and the NorthCoast 99ers, as above. (3) The last major condition is that there may not be any profit directly involved in the copying or transfer of this material. In other words, Clubs can use it in their newsletters and you can give a copy to your friend as long as its free.

```

SET TALK OFF
*
* Command File WHTST3
* Save as WHTST3/C
*
CLOSE ALL
LOCAL ? N 2 0
LOCAL SEL N 2 0
REPLACE ? WITH 0
WHILE .NOT. (?)
CLEAR
WRITE 2,8,"** Make A Selection **"
WRITE 4,10,"> 0 < To Quit CF"
WRITE 6,10,"> 1 < DO WHTST4"
WRITE 8,10,"> 2 < DO INITPR"
WRITE 10,10,"> 3 < SEL. THREE"
WRITE 12,10,"> 4 < SEL. FOUR"
WRITE 22,4,"Enter 0-4"
READ 22,15,SEL
WRITE 22,3,"
DOCASE
CASE SEL = 0
WRITE 18,10,"Have a nice day"
REPLACE ? WITH 1
BREAK
CASE SEL = 1
WRITE 18,15,"Number 1"
DO WHTST4
BREAK
CASE SEL = 2
WRITE 18,15,"Number 2"
DO INITPR
BREAK
CASE SEL = 3
WRITE 18,15,"Number 3"
BREAK
CASE SEL = 4
WRITE 18,15,"Number 4"
BREAK
ENDCASE
ENDWHILE
CLEAR
CLOSE ALL
SET TALK ON
RETURN
  
```

This month I'll attack the DOCASE, ENDCASE and a couple of additional tidbits. This tutorial will finish off almost all of the major points in the TIB Manual. Hopefully at that point you will have some idea what is going on with this language. Future tutorials will be less wordy and contain more intricate programming. I will also try to touch on the items we didn't cover in the manual so far.

The CF named WHTST3 is listed to the left. It is the beginning of TIB's menu capability and many other things which can be handled by combinations of WHILE, DOCASE and IF statements. Let's hit the high points. LOCAL ? N 2 0, initializes a local variable named "?". I named it ? because I couldn't come up with a good name for it, as in SEL which stands for selection. ? is a Numeric Variable with a size of 2 and 0 decimal places. A Numeric Variable can also be used as a Boolean Operator (if you're careful). A Boolean Operator is just something that transmits a True or False to TIB. To TIB and to many any programs and computers, False is represented by a Zero "0", and True is represented by a one "1". When we REPLACE ? WITH 0, ? is both a Numeric Variable which contains the value 0 and a Boolean Operator which represents False. WHILE statements need Boolean Operators to decide whether to execute the lines following the WHILE statement or skip them all and go directly to the statement after the ENDWHILE. In this case WHILE .NOT. (?) means WHILE ? is NOT true, do the statements following the WHILE. Because we placed a 0 in ? previously, it is False (or not true), so the WHILE will continue to loop until we change ? to a 1 or True, which you can do in the CASE SELECTION number 0. If you grasp this logic, you can see why I named it ? and why I said be careful. If you don't grasp the idea, just type things in as you see them. There will be more chances to sort out program logic in the future. When we enter the WHILE loop we CLEAR the screen and display a menu which can contain anything you wish TIB to do for you. At the bottom of the input screen TIB asks for your selection. Entering a number from 0 to 4 will set the variable SEL equal to that number. TIB then blanks out line 22 on your monitor and goes into the DOCASE routine. In the DOCASE, TIB goes to the first CASE and compares the value in SEL to the value on the right side of the equal sign. Therefore, if you selected 0 when asked for your choice, TIB would find a True match when it hit the first CASE comparison and would execute the lines between that CASE and the BREAK directly after it. In this case it would display the message "Have a nice day" and REPLACE ? WITH 1, which makes the variable ? True. When TIB hits the BREAK after REPLACE ? WITH 1 it goes to the ENDCASE. In this instance it would then go to the ENDWHILE which sends TIB back to the beginning of the WHILE loop. This time when we hit the WHILE .NOT. ? the ? equals 1 or True so the WHILE loop does not execute and the program goes to the next directive after the ENDWHILE. "I know that is a roundabout way to get here, but the computer can do it a lot faster than I can explain it." If you had selected 0, TIB would then finish and leave this CF which would return you to the DP. If, however, you had chosen any other number, TIB would have performed whatever tasks were present between the CASE that matched the SELECTION and the BREAK that followed it. For example, entering a 2 would DO the CF named WHTST4, or 3 would DO the CF named INITPR. I hope to eventually show you how to put a complete system together that will allow you to maintain and use a membership list for home, club, church or work, using menus and small CFs to do the work for you.

TI-BASE Tutorial 3 Page 2  
NorthCoast 99'ers (C) Martin A. Sealey

```

CLEAR
*                               9/15/88   WHILE
* Command File WHTST4           ENDWHILE
* Save as WHTST4/C
WRITE 12,15,"*****"
WRITE 13,15,"* WHTST4 *"
WRITE 14,15,"*****"
LOCAL ANS N 3 0
WRITE 22,1,"  Number of Cycles"
READ 22,22,ANS
WRITE 22,1,"
WHILE (ANS > 0)
WRITE 22,4,"  Cycles Left =",ANS
REPLACE ANS WITH ANS - 1
ENDWHILE
WRITE 22,1,"
CLEAR
RETURN

```

The CF above can be run by selecting number 1 from the menu screen of WHTST3. "Provided you type all this stuff in of course." WHTST4 doesn't really do a darn thing. When you run it, it asks you to enter a number. It will then start at that number and count down until it hits zero. You should enter a number like 4, 5 or 6 if you don't want to watch this thing counting down for a week. So you are saying to yourselves, why did this not put this junk in the tutorial. Let's go through it and I'll explain. We initialize the LOCAL ANS as a number. "No big deal here." You enter a number of your choice and then we hit the WHILE loop. In this instance it is written, WHILE (ANS > 0). Take a look at it. It's different from the last one. In this case the (ANS > 0) forms the Boolean Operator. As long as ANS holds a number which is greater than zero (0) the result is a True, and as long as the WHILE has a True stamp on it everything inside the loop is executed. Now inside the loop we find REPLACE ANS WITH ANS - 1. This is an accumulator. Each time the loop is executed you can add a quantity to your accumulator, or as in this case you can subtract a quantity from your accumulator. This is a lot like a FOR NEXT loop in Extended Basic. You enter a quantity for ANS. Each time the loop is executed 1 is subtracted from ANS. When ANS reaches 0 the loop is discontinued. I tried to show you this idea in its simplest form so you might have an easier time grasping the concept. The CF on the right side of this page is a real application of this idea. It's slightly stripped down so it wouldn't take up too much space, but it works and it's usable. It uses our old database named TNAMES. When you run it (DO WHTST5), it opens TNAMES and displays the first record in the file. It then asks you how many labels you want. If you enter a zero (0), it MOVES to the next record in the database and puts that one on the screen for you with the same question. If you enter a quantity greater than 0, like 4, it will print out 4 labels and then go to the next record. "I hope you get the idea." One thing about it that's slightly odd. The Emphasize command I placed in the first part of REPLACE TEMP1 WITH "%E" comes up as an E on the screen at the top left of the display. Just ignore it, the CF works fine.

I whipped WHTST5 up by loading WHTST4 into the FunnelWeb E/A Editor just as it looks above. Then I pulled in the old CF named LBL51 from Tutorial 1. With both CFs loaded I did a bunch of Moving and Copying. I typed over a few things and I had it. It took me about 45 minutes to seash it together and work out the bugs.

```

CLEAR
SET TALK OFF
SET RECNUM OFF
SET HEADING OFF
SET LINE=80
SET PAGE=000
* Command File WHTST5 10/07/88
* Save as WHTST5/C
* USE TNAMES and Print Multiple Labels
WRITE 11,15,"* Multiple *"
WRITE 13,15,"* Label *"
WRITE 15,15,"* Program *"
LOCAL TEMP1 C 40
LOCAL TEMP2 C 40
LOCAL TEMP3 C 40
LOCAL BLNK C 1
USE TNAMES
TOP
WHILE .NOT. (EOF)
CLEAR
REPLACE TEMP1 WITH "%E
  | " Exp. Date " | XP
WRITE 10,3,TEMP1
REPLACE TEMP2 WITH TRIM(FN) | " ";
  | MI | " | LN
WRITE 12,3,TEMP2
WRITE 14,3,SA
REPLACE TEMP3 WITH TRIM(CT) | " ";
  | ST | ". " | ZP
WRITE 16,3,TEMP3
LOCAL ANS N 3 0
WRITE 22,1,"  Number of Labels"
READ 22,22,ANS
WRITE 22,1,"
WHILE (ANS > 0)
PRINT TEMP1
PRINT BLNK
PRINT TEMP2
PRINT SA
PRINT TEMP3
PRINT BLNK
REPLACE ANS WITH ANS - 1
WRITE 22,4,"  Cycles Left =",ANS
WRITE 22,4,"
ENDWHILE
MOVE
ENDWHILE
CLEAR
CLOSE ALL
SET RECNUM ON
SET HEADING ON
SET TALK ON
RETURN

```

### TI-BASE Tutorial 3 Page 3 NorthCoast 99'ers (C) Martin A. Sooley

I'm not telling you how, or how fast I created a CF to make you feel "bad". I'm doing it to demonstrate that you should develop a logical procedure, and maintain good programming habits. When you don't understand something about a language, create a small CF or program to test your ideas. Make your new test CF as complete as possible as far as housekeeping is concerned. This will allow you to use your work as part of another larger program when the idea has been fully developed and you see it more clearly. Also, I had a lot less trouble finding bugs in WHTST4 than I would have in WHTST3 do to WHTST3's more complicated nature. I have also noticed, as I get deeper into this, that TI-BASE is slow. It is not always slow, but the more you ask it to do the slower it gets. This will not deter me from using or recommending that others use TIB, because what it can do outweighs this drawback. As I write CFs I will attempt to keep this in mind and attempt to minimize unneeded repetition. Let's take a quick look at WHTST3 for some ideas on this matter. I'd also like to add that some of this is theoretical and that any speed difference may vary greatly depending on your system. We have two WHILE loops in WHTST3. The smaller WHILE which runs from WHILE (ANS) 0) to the first ENDWHILE (which is 10 lines down) is nested inside a larger WHILE loop. The larger WHILE loop runs from WHILE .NOT. (EOF) to the ENDWHILE directly following the MOVE statement. The number of times the small loop will run, and print out labels, is determined by your answer to how many labels you want. The larger WHILE loop will run until it reaches the End Of File (or database) that is in use at the time. "Here's the time saver." Because the inner WHILE loop may run many times, depending on the number of labels you request, I have attempted to remove any non-essential code (program lines) from this loop. You'll notice that I loaded all my variables (REPLACE TEMP1 WITH " "), etc. before I got into this loop. Therefore, TIB did not have to perform that task 100 times if I said I wanted 100 labels. There is one other consideration you must make in this situation. In order to do this it was necessary to create the variables TEMP1, TEMP2 and TEMP3. You should not get carried away and use up all of TIB's variable space. You must balance the idea of speed with the lack of massive free memory space. In this case we have enough memory space to do the job and these variables will be thrown away and the space will be freed up when this CF terminates with the RETURN statement. You will probably have your first real problem with memory space when you run a CF that runs another CF which in turn runs another CF. If each CF initializes some variables of its own, by the time you get to the last one the variable space will all be used up. I'll point this out again later when it comes up in the natural scheme of things. We better move along to some items with more immediate use potential.

On this page you will find INITPR which was a selection from the menu of WHTST3. This CF is nothing new and spectacular, but I find it useful. It is the same as WHTST3, but it has been modified to send printer control codes. I think I have covered the ideas in INITPR previously. I'd like to point out (??). If this CF is run from WHTST3 as selection 2 you must not re-use the (?) as a variable. If you do use the same name here and you change the value of ?, you may cause unexpected things to happen when TIB returns to WHTST3 and carries ? with

it. Also, if you wish to use INITPR as a stand alone CF, you may want to turn the RECNUM and stuff back on at the end of the CF.

```

SET TALK OFF
SET HEADING OFF
SET RECNUM OFF
*           09/12/88
* Command File INITPR
* Save as INITPR/C
*
CLOSE ALL
LOCAL ?? N 2 0
LOCAL SEL N 2 0
LOCAL CNTRL C 2
REPLACE ?? WITH 0
WHILE .NOT. (??)
CLEAR
WRITE 1,6,"** Send Printer";
      " Controls **"
WRITE 2,9,"** Make a selection **"
WRITE 4,10,"> 0 < Leave this CF"
WRITE 6,10,"> 1 < Emphasized on"
WRITE 8,10,"> 2 < Italics on"
WRITE 10,10,"> 3 < Condensed on"
WRITE 12,10,"> 4 < Doublestrike"
WRITE 14,10,"> 5 < RESET Printer"
WRITE 22,4,"Enter 0-5"
READ 22,15,SEL
WRITE 22,3,"
DOCASE
CASE SEL = 0
CLOSE ALL
CLEAR
WRITE 18,12,"Do Not Turn Your
WRITE 20,12," Printer Off."
REPLACE ?? WITH 1
BREAK
CASE SEL = 1
REPLACE CNTRL WITH "L"
PRINT CNTRL
BREAK
CASE SEL = 2
REPLACE CNTRL WITH "L4"
PRINT CNTRL
BREAK
CASE SEL = 3
REPLACE CNTRL WITH "f"
PRINT CNTRL
BREAK
CASE SEL = 4
REPLACE CNTRL WITH "L6"
PRINT CNTRL
BREAK
CASE SEL = 5
REPLACE CNTRL WITH "L8"
PRINT CNTRL
BREAK
ENDCASE
ENDWHILE
CLEAR
RETURN

```

## TI-BASE Tutorial 3 Page 4

NorthCoast 99'ers (C) Martin A. Sooley

INITPR is merely another demonstration of what you can do with DOCASE and WHILE statements. Many of the ideas I have presented in my tutorials can be done in other ways. Some of the others ways may turn out to be more efficient or more convenient to use. I still consider myself to be a beginner at TI-Base so I anticipate changing my ideas on how to optimize program power and minimize program run time. As I stated in the last tutorial, I would appreciate letters or notes from TI-Base users with comments, tips or questions on this subject. I do not have the time to write back to you. In many cases I find myself rushing to the last minute before the newsletter deadline to finish the months tutorial.

```

100 ! ##### TIB->DV/80 #####
110 ! (C) 1988 Martin A. Sooley
120 !
130 ! Extended Basic program to read TIB I/F40 files
140 ! and write B/V80 files for TIN or FunnelWeb.
150 !
160 ! You must add one blank space to the beginning
170 ! of every line in the TIB I/F40 file.
180 ! After transfer, check all lines for any
190 ! missing characters, especially the end.
200 !
210 CALL CLEAR :: CALL SCREEN(6)
220 PRINT " Enter INPUT File ALL CAPS"
230 PRINT " Example: DSK1.OPERATOR/C"
240 INPUT "      ":INS
250 IF LEN(INS)>12 THEN OUT$=SEG$(INS,1,12)&"8DV"
260 IF LEN(INS)<13 THEN OUT$=INS&"8DV"
270 PRINT "OUT File= ":OUT$ :: PRINT
280 INPUT " Is that OK Y/N ":ANS$
290 IF ANS$="N" OR ANS$="n" THEN 210
300 OPEN #1:INS,INTERNAL,FIXED 40,INPUT :: LN=40
309 : OPEN #1:INS,DISPLAY ,VARIABLE 80,INPUT :: LN=80
310 OPEN #2:OUT$,DISPLAY ,VARIABLE 80,OUTPUT
320 IF EOF(1)THEN CLOSE #1 :: CLOSE #2 :: GOTO 480
330 !
340 INPUT #1:A$
350 PRINT A$
360 FOR I=1 TO LN
370 T$=SEG$(A$,I,1)
380 ON ERROR 440
390 IF ASC(T$)>126 THEN T$=" "
400 IF ASC(T$)<32 THEN T$=" "
410 B$=P$
420 IF I<1 THEN P$=T$ ELSE P$=B$&T$
430 NEXT I
440 PRINT #2:P$
450 P$=""
460 !
470 GOTO 320
480 CALL CLEAR :: PRINT " ### FINISHED ###" :: : : :
490 INPUT " Quit Program Y/N ":Q$
500 IF Q$="Y" OR Q$="y" THEN STOP ELSE GOTO 210
510 ! ##### TIB->DV/80 #####
520 END

```

I have also been recommending the use of FunnelWeb in the non-word wrap mode. I have had some problems with this procedure. I figured if I was having a problem, someone else must be having the same problem. The problem is hidden characters in the CF. In most cases I am in a hurry to produce code (write programs or CFs). Many times I jump into the wrong editor mode and start typing. In many instances this will not be a problem. In FunnelWeb pressing CTRL O will throw you into non-wordwrap mode, which is the same as the E/A Editor. However, if you hit the CTRL key and some other key at the same time while you are still in wordwrap mode, you can insert characters which are invisible on the screen but do crazy things when the CF is run. At one point I wasted more precious time than I could afford trying to find one of these invisible little land mines. I remembered a little Extended Basic program I had written for another task several weeks earlier. At that time I wanted to convert several Command Files (CFs) to DV/80 files so I could print them out and study them more carefully. The program I wrote was TIB->DV/80 which is listed on this page. I think some of you may get some use out of it. As is is now, it will read an I/F 40 file (like a CF), and write it to a B/V80 file for FunnelWeb. There is one thing you must do first. A control code in CFs causes the loss of the first character in every line. You can overcome this by loading the CF into TIB using MODIFY COMMAND (filename). Pressing FCTN 2 for insert mode, which stays on until you press FCTN 2 again. Then add one blank space to the beginning of every line. When you run my program the blank space will be lost instead of something you need. If you want to run the program on a B/V80 file, remove the exclamation point from line 309 and place one in front of line 300. Adding the space is only necessary with B/F40 files, not B/V80. The program will check every character in the file and will kick out all characters below 32 or above 126. That includes those invisible land mines in your CF. Unfortunately you will have to replace any printer controls.

Well, I'm running out of space and my mind is shot, so I'd like to say a couple more things and this one is finished. First I'd like to thank the people of the NorthCoast 99'ers for allowing me the space in their newsletter to write this tutorial, and a lot of miscellaneous articles in the past. The NorthCoast members are a great group of people. I'd like to throw in the fact that any TI 99/4A owner in the continental US can join the NorthCoast 99'ers for only \$15.00 per year. As you can imagine \$15.00 is little more than the cost of printing and mailing this great newsletter. If you would like to send your checks to me (Payable to the NorthCoast 99'ers U6), I will expedite your membership. Also any comments on the TIB column can be sent to Martin A. Sooley, 6149 Bryson Drive, Mentor, Ohio, 44060.

I am going to announce at this time that I will produce a TIB help disk. The disk will be a floppy and contain all of the tutorials and Command Files to date plus anything else I think may be helpful. I already have 390 sectors of tutorials. For this I would like a donation of \$3.00 to cover the Disk, Mailer, Postage, Handling and wear on my disk drives. Please make these checks payable to Martin A. Sooley at the previous address, and make two checks if you want the help disk and a membership.



### TI-BASE Tutorial 3 Page 3 NorthCoast 99'ers (C) Martin A. Swoley

I'm not telling you how, or how fast I created a CF to make you feel "bad". I'm doing it to demonstrate that you should develop a logical procedure, and maintain good programming habits. When you don't understand something about a language, create a small CF or program to test your ideas. Make your new test CF as complete as possible as far as housekeeping is concerned. This will allow you to use your work as part of another larger program when the idea has been fully developed and you see it more clearly. Also, I had a lot less trouble finding bugs in WHTST4 than I would have in WHTST3 do to WHTST3's more complicated nature. I have also noticed, as I get deeper into this, that TI-BASE is slow. It is not always slow, but the more you ask it to do the slower it gets. This will not deter me from using or recommending that others use TIB, because what it can do outweighs this drawback. As I write CFs I will attempt to keep this in mind and attempt to minimize unneeded repetition. Let's take a quick look at WHTST3 for some ideas on this matter. "I'd also like to add that some of this is theoretical and that any speed difference may vary greatly depending on your system." We have two WHILE loops in WHTST3. The smaller WHILE which runs from WHILE (ANS > 0) to the first ENDWHILE (which is 10 lines down) is nested inside a larger WHILE loop. The larger WHILE loop runs from WHILE .NOT. (EOF) to the ENDWHILE directly following the MOVE statement. The number of times the small loop will run, and print out labels, is determined by your answer to how many labels you want. The larger WHILE loop will run until it reaches the E)nd O)ff F)ile (or database) that is in use at the time. "Here's the time saver." Because the inner WHILE loop may run many times, depending on the number of labels you request, I have attempted to remove any non-essential code (program lines) from this loop. You'll notice that I loaded all my variables (REPLACE TEMP1 WITH "\*\*\*\*"), etc. before I got into this loop. Therefore, TIB did not have to perform that task 100 times if I said I wanted 100 labels. There is one other consideration you must make in this situation. In order to do this it was necessary to create the variables TEMP1, TEMP2 and TEMP3. You should not get carried away and use up all of TIB's variable space. You must balance the idea of speed with the lack of massive free memory space. In this case we have enough memory space to do the job and these variables will be thrown away and the space will be freed up when this CF terminates with the RETURN statement. You will probably have your first real problem with memory space when you run a CF then runs another CF which in turn runs another CF. If each CF initializes some variables of it's own, by the time you get to the last one the variable space will all be used up. I'll point this out again later when it comes up in the natural scheme of things. We better move along to some items with more immediate use potential.

On this page you will find INITPR which was a selection from the menu of WHTST3. This CF is nothing new and spectacular, but I find it useful. It is the same as WHTST3, but it has been modified to send printer control codes. I think I have covered the ideas in INITPR previously. I'd like to point out (??). If this CF is run from WHTST3 as selection 2 you must not re-use the (?) as a variable. If you do use the same name here and you change the value of ?, you may cause unexpected things to happen when TIB returns to WHTST3 and carries ? with

it. Also, if you wish to use INITPR as a stand alone CF, you may want to turn the RECNUM and stuff back on at the end of the CF.

```

SET TALK OFF
SET HEADING OFF
SET RECNUM OFF
*           09/12/88
* Command File INITPR
* Save as INITPR/C
*
CLOSE ALL
LOCAL ?? N 2 0
LOCAL SEL N 2 0
LOCAL CNTRL C 2
REPLACE ?? WITH 0
WHILE .NOT. (??)
CLEAR
WRITE 1,6,"** Send Printer";
      " Controls **"
WRITE 2,9,"** Make a selection **"
WRITE 4,10,"> 0 < Leave this CF"
WRITE 6,10,"> 1 < Emphasized on"
WRITE 8,10,"> 2 < Italics on"
WRITE 10,10,"> 3 < Condensed on"
WRITE 12,10,"> 4 < Doublestrike"
WRITE 14,10,"> 5 < RESET Printer"
WRITE 22,4,"Enter 0-5"
READ 22,15,SEL
WRITE 22,3,"
DOCASE
CASE SEL = 0
CLOSE ALL
CLEAR
WRITE 18,12,"Do Not Turn Your
WRITE 20,12," Printer Off."
REPLACE ?? WITH 1
BREAK
CASE SEL = 1
REPLACE CNTRL WITH "LE"
PRINT CNTRL
BREAK
CASE SEL = 2
REPLACE CNTRL WITH "L4"
PRINT CNTRL
BREAK
CASE SEL = 3
REPLACE CNTRL WITH "f"
PRINT CNTRL
BREAK
CASE SEL = 4
REPLACE CNTRL WITH "LE"
PRINT CNTRL
BREAK
CASE SEL = 5
REPLACE CNTRL WITH "LO"
PRINT CNTRL
BREAK
ENDCASE
ENDWHILE
CLEAR
RETURN

```

# TI-BASE Tutorial 3 Page 4

NorthCoast 99'ers (C) Martin A. Sooley

INITPR is merely another demonstration of what you can do with DOCASE and WHILE statements. Many of the ideas I have presented in my tutorials can be done in other ways. Some of the others ways may turn out to be more efficient or more convenient to use. I still consider myself to be a beginner at TI-Base so I anticipate changing my ideas on how to optimize program power and minimize program run time. As I stated in the last tutorial, I would appreciate letters or notes from TI-Base users with comments, tips or questions on this subject. I do not have the time to write back to you. In many cases I find myself rushing to the last minute before the newsletter deadline to finish the month's tutorial.

```

100 ! ##### TIB->DV/80 #####
110 ! (C) 1988 Martin A. Sooley
120 !
130 ! Extended Basic program to read TIB I/F40 files
140 ! and write D/V80 files for TIN or FunnelWeb.
150 !
160 ! You must add one blank space to the beginning
170 ! of every line in the TIB I/F40 file.
180 ! After transfer, check all lines for any
190 ! missing characters, especially the end.
200 !
210 CALL CLEAR :: CALL SCREEN(6)
220 PRINT " Enter INPUT File ALL CAPS"
230 PRINT " Example: DISK1.OPERATOR/C"
240 INPUT "      ":IN$
250 IF LEN(IN$)>12 THEN OUT$=SEG$(IN$,1,12)&"$DV"
260 IF LEN(IN$)<13 THEN OUT$=IN$&"$DV"
270 PRINT "OUT File= ";OUT$ :: PRINT
280 INPUT " Is that OK Y/N ":ANS$
290 IF ANS$="M" OR ANS$="n" THEN 210
300 OPEN #1:IN$,INTERNAL,FIXED 40,INPUT :: LN=40
309 ! OPEN #1:IN$,DISPLAY ,VARIABLE 80,INPUT :: LN=80
310 OPEN #2:OUT$,DISPLAY ,VARIABLE 80,OUTPUT
320 IF EOF(1)THEN CLOSE #1 :: CLOSE #2 :: GOTO 480
330 !
340 INPUT #1:A$
350 PRINT A$
360 FOR I=1 TO LN
370 T$=SEG$(A$,I,1)
380 ON ERROR 440
390 IF ASC(T$)>126 THEN T$=" "
400 IF ASC(T$)<32 THEN T$=" "
410 B$=P$
420 IF I<1 THEN P$=T$ ELSE P$=B$&T$
430 NEXT I
440 PRINT #2:P$
450 P$=""
460 !
470 GOTO 320
480 CALL CLEAR :: PRINT " ### FINISHED ###" :: : : :
490 INPUT " Quit Program Y/N ":Q$
500 IF Q$="Y" OR Q$="y" THEN STOP ELSE GOTO 210
510 ! ##### TIB->DV/80 #####
520 END

```

I have also been recommending the use of FunnelWeb in the non-word wrap mode. I have had some problems with this procedure. I figured if I was having a problem, someone else must be having the same problem. The problem is hidden characters in the CF. In most cases I am in a hurry to produce code (write programs or CFs). Many times I jump into the wrong editor mode and start typing. In many instances this will not be a problem. In FunnelWeb pressing CTRL O will throw you into non-wordwrap mode, which is the same as the E/A Editor. However, if you hit the CTRL key and some other key at the same time while you are still in wordwrap mode, you can insert characters which are invisible on the screen but do crazy things when the CF is run. At one point I wasted more precious time than I could afford trying to find one of these invisible little land mines. I remembered a little Extended Basic program I had written for another task several weeks earlier. At that time I wanted to convert several Command Files (CFs) to DV/80 files so I could print them out and study them more carefully. The program I wrote was TIB->DV/80 which is listed on this page. I think some of you may get some use out of it. As is now, it will read an I/F 40 file (like a CF), and write it to a D/V80 file for FunnelWeb. There is one thing you must do first. A control code in CFs causes the loss of the first character in every line. You can overcome this by loading the CF into TIB using MODIFY COMMAND (filename). Pressing FCTN 2 for insert mode, which stays on until you press FCTN 2 again. Then add one blank space to the beginning of every line. When you run my program the blank space will be lost instead of something you need. If you want to run the program on a DV/80 file, remove the exclamation point from line 309 and place one in front of line 300. Adding the space is only necessary with D/F40 files, not D/V80. The program will check every character in the file and will kick out all characters below 32 or above 126. That includes those invisible land mines in your CF. Unfortunately you will have to replace any printer controls.

Well, I'm running out of space and my mind is shot, so I'd like to say a couple more things and this one is finished. First I'd like to thank the people of the NorthCoast 99'ers for allowing me the space in their newsletter to write this tutorial, and a lot of miscellaneous articles in the past. The NorthCoast members are a great group of people. I'd like to throw in the fact that any TI 99/4A owner in the continental US can join the NorthCoast 99'ers for only \$15.00 per year. As you can imagine \$15.00 is little more than the cost of printing and mailing this great newsletter. If you would like to send your checks to me (Payable to the NorthCoast 99'ers US), I will expedite your membership. Also any comments on the TIB column can be sent to Martin A. Sooley, 6149 Bryson Drive, Mentor, Ohio, 44060.

I am going to announce at this time that I will produce a TIB help disk. The disk will be a floppy and contain all of the tutorials and Command Files to date plus anything else I think may be helpful. I already have 390 sectors of tutorials. For this I would like a donation of \$3.00 to cover the Disk, Mailer, Postage, Handling and wear on my disk drives. Please make these checks payable to Martin A. Sooley at the previous address, and make two checks if you want the help disk and a membership.

### GRAPHICS COMPATABILITY

COURTESY THE BYTEMONGER - LEXINGTON, KY - FEB - 1987

The multitude of Graphics programs available for the TI computer and their compatability with word processing programs has prompted a request for some description of each. This is an attempt to clarify compatability among most of the later programs. The diagram on the next page covers most of the freeware and commercial programs available.

TI-WRITER is the only prudent choice for a word processing program for those having 32K and disk drives. There are no others which come close to providing the features and versatility. All versions still use TI's WRITER program files. The best and least costly is FURNELWEB's version which in addition to freeing E/A & WRITER from their respective modules, includes C, DISKO, FORTH, FASTTERM, and our choice of others in a disk-based program which shines when installed on a RAMDISK. Almost all of the programs which we will discuss will be used either with text from TI-WRITER or through TI-WRITER.

The CSGD series of graphics programs written by Dave Rose is shown at the top of the diagram. It is compatible with both Prowriter and Epson-gemini printers, but you must purchase the correct version for your printer. The keyboard or joystick can be used in any of the Editors. The message printing program is common to CSGD-I, II, and III.

CSGD-I contains the Editor programs for creating your own (1) Character Sets, (2) 5X5 graphics, and (3) Pictures. The Editors, primarily the Character Editor, have undergone 4 revisions. Provisions are also made to jockey graphics around to convert between alternate printer types.

CSGD-II is basically a banner program which prints the message sideways and 8 times magnified. It also contains the Graphic Editor, but not the other editors. It has gone through three revisions, including the latest which allows printing lower case in the Banner mode.

CSGD-III is primarily a Label program which produces multi-width labels in 3 heights. It also contains the Message program and an "easier to operate" Letterhead program. ALL OF THE FILES CREATED IN CSGD ARE I/Y 254 and are not compatible with TI-WRITER. The Occuprinter is compatible with WRITER and will produce 1 or 2 column texts using a choice of 6 type fonts and D/Y 80 files through the Formatter. It does not handle fonts of greater height than 1 row.

The CSGD programs are supported by a multitude of graphics, pictures and over 100 fonts written by Dave Rose and contributors who use and enjoy his programs.

TI-ARTIST is an extremely versatile drawing program written by Chris Faherty. It allows creating, loading and modifying, size changes and many other features using the keyboard or a combination of the keyboard and Joystick.

NONE OF THE PROGRAMS DESCRIBED IN THIS DISCUSSION CAN BE LEARNED READING THE MANUAL. You must use them and make your mistakes: It helps you to remember.

The only files that were available with ARTIST were ART-EXTRAS; however, Dave Rose converted his I/Y 254 files to D/Y80 which can be loaded as "Instances" and were released as the COMPANIONI-III series. In addition, TI/ARTIST allows the loading of GRAPHX files so that a

rather large base of graphics is available to use and modify or create your own.

GRAPHX is a program with quite a few similarities to ARTIST; was written in Australia and introduced in the USA in 1984. It allows use of the Joystick ONLY, and is not compatible with any but the Epson printer. It is a quite versatile drawing board in the hands of an experienced user.

The support graphics available consist of a series of GRAPHX COMPANIONS and GRAPHX PICTURES. They are all well done. The availability of additional graphics is made possible through TI-ARTIST which will load and save to the Image (PGM) format.

JOYPAINT & JP.PAL are Graphics Drawing Boards well spoken of by their owners. It does have options which allow loading from and saving to ARTIST &/or GRAPHX filtypes. It is currently only compatible with the EPSW printer.

ARTCONVERT is a Program that converts TI-ARTIST files to TI-WRITER files. It is supported by 4 diskettes of files; ARTDATA-I through IV. This permits anyone with TI-WRITER to have the ability to print graphics. It will also merge and print two graphics, but there are no provisions to include text in the graphic file to complete a document.

One unique feature of ARTCONVERT allows the user to convert one row high TI-ARTIST fonts for use through TI-WRITER. This would allow conversion of all of Rose's 1 row high fonts in Companion I-III to be used though they contain 70 of 96 typewriter keys if you do not have the COMPANION series. Compatible with Prowriter and Epson-Gemini.

FONTWRITER by Peter Hoddie is really the first graphics program which is compatible with TI-ARTIST and TI-WRITER. With this program you can create your own type fonts, revise TI-ARTIST and CSGD fonts and use TI-ARTIST Instances mixed in printed documents even on the same line. The resources for this program are as broad as all of the ARTIST files plus all of those I have indicated earlier which can be converted through ARTIST.

ARTIST APPRENTICE is similar in several ways to FONTWRITER. It allows use of TI-ARTIST type files, fonts and graphics to produce files which are printed according to a "Scheduler". It is limited in being compatible only with EPSON printers.

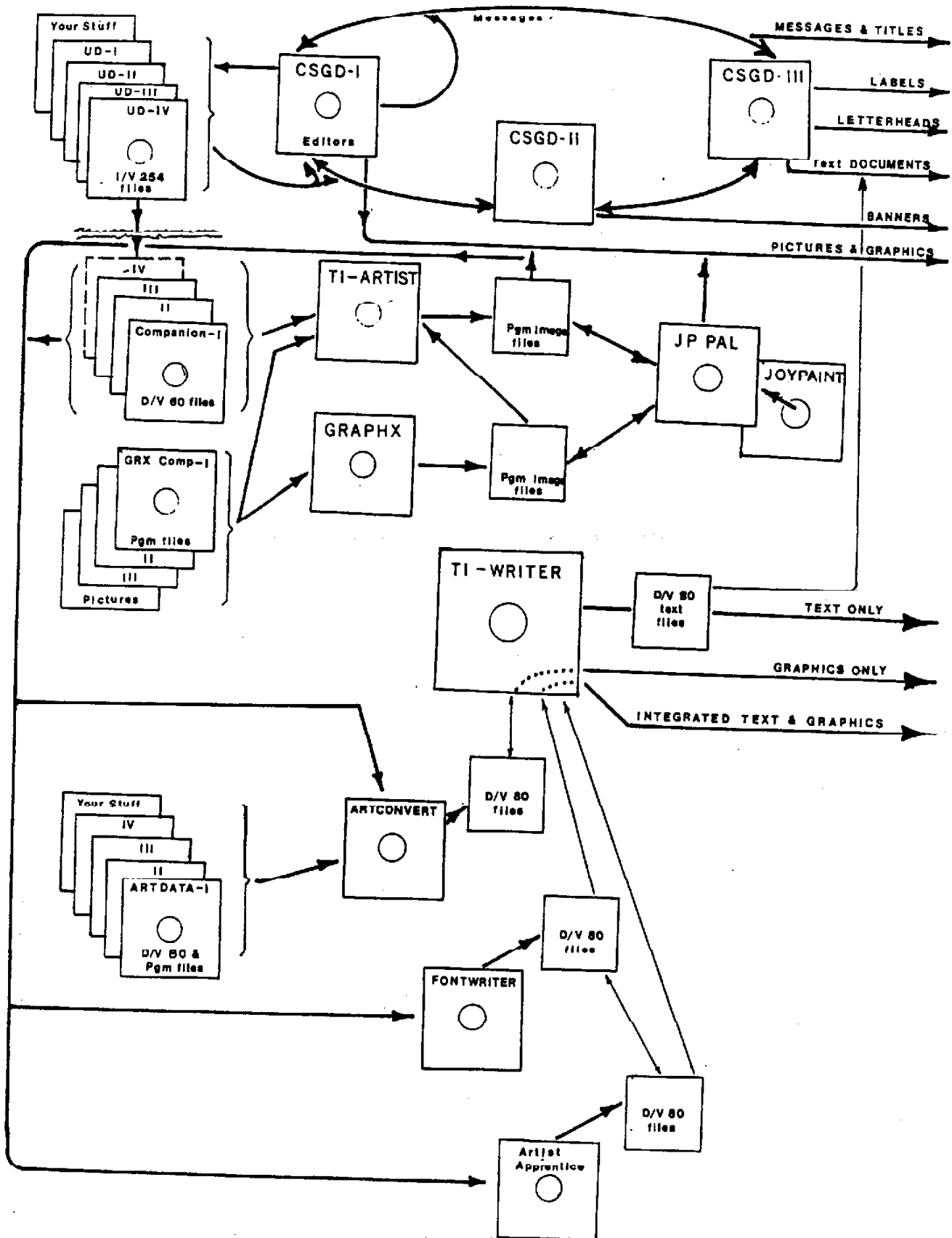
#### GENERAL OBSERVATIONS:

Most of the programs are, by necessity, somewhat complex. The MAX-RLE program with an almost infinite picture resource can also be used to convert files between D/Y 80, GRAPHX, TI-ARTIST, and IV 128.

There are many more Graphic programs available since this article was first published, ie, the series from Rodger Merritt, PRINT IT, PICTURE.IT, JIFFY FLYER, JIFFY CARD, and FORMSHOP. PRINT WIZARD, MACFLIX, CERTIFICATE 99, 1000 WORDS, etc. All of the programs are sufficiently time consuming to require the need for reproducing and also they are fairly small.

Don MacClellan - Bluegrass 99 computer Society  
(Slightly edited by DS of the NorthCoast 99ers)

## SOME POPULAR GRAPHICS PROGRAMS



## LET'S TALK RAM DISK BY JOHN F. WILLFORTH

WE HAVE A PROBLEM AS TI-994A USERS THESE DAYS THAT MOST MARKET VIEWERS AND MANY TI OWNERS WOULD NOT HAVE BELIEVED POSSIBLE JUST A FEW MONTHS AGO. THERE ARE FOUR MAJOR VENDORS OF RAM DISKS IN THE U.S.A.. THERE IS ALSO A VARIETY OF FEATURES AND SIZES IN THESE UNITS, SOME OF WHICH ARE NOT FOUND IN UNITS BEING PRODUCED FOR ATARI, COMMODORE, APPLE, OR THE P.C. LINES OF COMPUTERS. THE BIGGEST PROBLEM FACING THE USER NOW IS "WHICH TO BUY?".

THE PURPOSE OF THIS ARTICLE IS TO PROVIDE SOME THOUGHTS AND FACTS TO HELP YOU DECIDE. THE NEXT SEVERAL PARAGRAPHS ARE NOT INTENDED TO PROMOTE ANY ONE OF THE RAM DISKS MENTIONED, AND MAY CONTAIN ERRONEOUS INFORMATION. HOPEFULLY BY OMISSION RATHER THAN COMMISSION.

FIRST A RAM DISK IS BY DEFINITION, A SOFTWARE/FIRMWARE SUPPORTED RAM CIRCUIT BOARD EMULATING A DISK. I.E. A CIRCUIT CARD, THAT WHEN PLUGGED INTO YOUR PEB, WILL ALLOW YOU TO STORE AND RETRIEVE DISK TYPE FILES TO/FROM THE UNIT WITH THE SAME EASE AS YOU WOULD TO YOUR PHYSICAL DISK DRIVE (DSK1 FOR EXAMPLE). BECAUSE RAM IS A NON-MECHANICAL DEVICE, IT IS NOT SUBJECT TO THE DELAYS OF POSITIONING A READ/WRITE HEAD OVER A CYLINDER (TRACK), AND WAITING FOR THE DISKETTE TO NOW ROTATE TO THE DESIRED SECTOR, AND THEN READ/WRITE DATA FROM/TO THE SPINNING DISK IN SERIAL (LIKE CASSETTE) FORM. THESE THREE MECHANICAL LIMITATIONS ARE THE MAIN REASON THAT DISKS ARE SLOW. YES, DISKS ARE ABOUT AS MUCH FASTER THAN CASSETTES AS RAM DISKS ARE FASTER THAN DISKS! IF YOU BUY ONE, YOU WILL ENJOY THAT KIND OF IMPROVEMENT, NO MATTER WHICH BRAND YOU BUY.

THE MAJOR TWO TYPES OF RAM DISKS ARE THOSE USING DYNAMIC RAM (MYARC, CORCOMP) AND STATIC RAM (HORIZON, MIKE BALLMAN ENHANCED HORIZON ( SOLD BY BUD MILLS )).

- \* DYNAMIC RAM IS LESS EXPENSIVE, LARGER CAPACITY, BUT REQUIRES MORE SUPPORT CIRCUITRY, DRAWS MORE POWER, AND IS MORE CUMBERSOME TO SUPPORT IF THE POWER IS LOST (LIKE TURNING OFF THE PEB).
- \* STATIC RAM IS LOWER POWER AND THUS EASY TO SUPPORT DURING POWER OUTAGES. THEY ARE MORE EXPENSIVE, TAKE MORE SPACE ON A BOARD, AND THUS FOR THE AMOUNT MEMORY NEEDED, MORE EXPENSIVE THAN DYNAMIC RAM.

I WOULD LIKE TO TALK ABOUT ADDITIONAL FEATURES. THE FIRST ONE THAT I AM MOST ASKED ABOUT, IS THE SPOOLING FEATURES. ALL BUT THE HORIZON AND THE ENHANCED HORIZON, HAVE THE SPOOLING FEATURE. AT THIS PRINTING, ALL THAT HAVE SPOOLING DO IT IN A DIFFERENT MANNER, BUT JUST AS EFFECTIVELY. SOME OF YOU MAY ASK, "WHAT IS SPOOLING?" WELL TO MAKE IT SIMPLE, SPOOLING IS STORING DATA THAT IS TO GO TO A DEVICE (PRINTER MODEM, ETC.) IN MEMORY SPACE, AND RELEASING IT AS IT CAN BE USED BY THE RECEIVING DEVICE. REMEMBER THE TI SITS THERE SENDING TO THE PRINTER UNTIL ALL THE FILE IS SENT. THEN IT IS ABLE TO ACCEPT YOUR NEXT COMMAND OR CONTINUE INSTRUCTIONS. A SPOOLER ACCEPTS THIS INFORMATION AS IF IT WERE THE PRINTER, MODEM ETC., AND AT A MUCH HIGHER RATE THAN ANY OF THOSE PERIPHERALS COULD, AND IN MOST CASES WILL ACCEPT THE ENTIRE FILE TO BE PROCESSED IN A FEW SECONDS VERSES SEVERAL MINUTES. THE TI-99/4A WILL THEN ASSUME THAT ALL THAT IT HAD TO DO WAS DONE AND COME BACK TO YOU FOR FURTHER USE, WHEN IN FACT THE JOB IS STILL BEING COMPLETED BY THE SPOOLER AT A PACE THAT THE PRINTER, MODEM ETC. CAN HANDLE IT. PRETTY NEAT! HUH?

ANOTHER FEATURE IS PARTITIONING, OR MULTIPLE DISKS BEING ASSIGNED WITHIN A SINGLE RAM DISK CARD. WHAT THIS MEANS IS THAT IF YOU HAVE A SINGLE DRIVE ON YOUR SYSTEM (DSK1 FOR EXAMPLE), YOU MAY CALL A PORTION OF A RAM DISK DSK2, OR DSK3, DSK4, ETC. NOW YOU HAVE ONE PHYSICAL, AND UP TO WHO KNOWS HOW MANY OTHER DISKS WHICH ARE PART OF THE RAM DISK.

STILL ANOTHER FEATURE IS BUILT IN COMMANDS, EACH DISK MENTIONED ABOVE HAS IT'S OWN SET. FOR EXAMPLE, YOU CAN TYPE "CALL DM" IN BASIC COMMAND MODE, AND A FILE CALLED DM1 WILL BE BOOTED FROM THE DISK, FOLLOWED BY DM2. MANY COMMANDS DEALING WITH MEMORY ARE ALSO INCORPORATED.

FEATURES SUCH AS CLOCK, (TIME OF DAY), ANALOG-TO-DIGITAL, ETC. ARE NOW COMING AVAILABLE ON THE RAM DISKS.

YOU MAY NEED MORE INFORMATION TO ORDER YOUR RAM DISK THAN I'VE PROVIDED HERE. NEXT MONTH'S ARTICLE WILL GET MORE SPECIFIC ON EACH RAM DISK BUT IF YOU BELIEVE THE ADS, MAYBE YOU CAN UNDERSTAND THEM A LITTLE BETTER NOW, AND YOUR READY TO JUMP IN. GOOD LUCK!

West Penn  
88'ers

Dec 87

The "CIN-DAY NEWS" is published through the efforts of the:



CIN-DAY USER GROUP  
416 PINWOOD AVENUE  
PIQUA, OHIO 45356



NOTE: All remittances are to be made payable to the "CIN-DAY USER GROUP" and are to be mailed to the above address.

NEWSLETTER SUBSCRIPTIONS for non-members are ..... \$10.00 per year  
REMOTE MEMBERSHIPS for Newsletter & Library privileges ..... \$15.00 per year  
FULL/FAMILY MEMBERSHIPS for single or family households are .. \$20.00 per year

NOTE: Please print & submit the following information in regards to the above:  
NAME, ADDRESS, CITY, STATE, ZIP CODE, HOME and WORK TELEPHONE NUMBERS.  
For additional information, contact: Rick Kellogg - Group Contact (513) 773-5941

The CIN-DAY USER GROUP officers are:

PRESIDENT-CINCINNATI CHAPTER .....	SAM MOON .....	(513) 321-3932
PRESIDENT-DAYTON CHAPTER .....	RICK KELLOGG .....	(513) 773-5941
SECRETARY-CINCINNATI CHAPTER .....	JOHN NEESS (ACTING) .....	(513) 474-1364
SECRETARY-DAYTON CHAPTER .....	FRANK LARRICK .....	(513) 890-3957
TREASURER-CINCINNATI CHAPTER .....	KEN CARPENTER .....	(513) 729-2991
TREASURER-DAYTON CHAPTER .....	BILL STAGER .....	(513) 878-7198
LIBRARIAN-CINCINNATI CHAPTER .....	CAROLYN CARPENTER .....	(513) 729-2991
LIBRARIAN-DAYTON CHAPTER .....	ERIC BISHOP .....	(513) 847-8280
NEWSLETTER EDITOR .....	ERIC SAUNDERS .....	(513) 294-5703

NON-PROFIT MAILING PERMIT HOLDER:  
(FOR THE CIN-DAY USER GROUP)  
Frank Larrick  
8514 Elmway Drive  
Dayton, OH 45415-1609



Postmaster:

ADDRESS CORRECTION REQUESTED

THIS MONTH'S DAYTON MEETING: June 10th, 1989 at 12:15 PM  
Coin Room, 5th floor, Downtown Lazarus.  
NEXT MONTH'S DAYTON MEETING: July 15th, 1989 at 12:15 PM  
Coin Room, 5th floor, Downtown Lazarus.

THIS MONTH'S CINCINNATI MEETING: June 10th, 1989 at 12:00 Noon  
Campbell County Library, Kentucky.  
NEXT MONTH'S CINCINNATI MEETING: July 15th, 1989 at 12:00 Noon  
Campbell County Library, Kentucky.

Attn: Greg Justice, Editor  
Dallas TI Home Computer Grp.  
P.O. Box 29863  
Dallas, TX  
75229

