

TI - 99 USERS OF
PERTH

TIUP TIT BITS

VOLUME 14 No. 5
FEBRUARY 1996

TIUP Pays TRIBUTE TO GEOFF WARNER.

On the 16th. December last, at our annual Xmas Windup Function, the President of TIUP presented our past Secretary with a Certificate of Merit in appreciation of his contribution to the Texas Instrument Users of PERTH Group.



Seen in the photograph (l. to R) are: Geoff. Warner and Merve Trowbridge.

PLEASE NOTE,
OUR NEW MAILING
ADDRESS IS

SECRETARY TIUP
20 HUDSON ST.
BAYSWATER 6053
WESTERN
AUSTRALIA.

* Index Page 14*

EDITORIAL by F. Graham Secretary/Editor

Another year has passed, and by the 20th. March, 1996 the Texas Instrument (Computer) Users of Perth Group will be in its "fifteenth" year of providing continuous service to the TI-99/4A community of Western Australia.

During the intervening years our membership, which at one time was in excess of 145, has steadily grown fewer and considerably older, however, the devotion and respect for our computer has not diminished.

Our members have no doubt, what so ever, that the "orphan" was and still is the most under utilised computer ever produced, which helps to explain to some extent, the reasons why so many TI-ers all over the world, devote their time and energy into carrying out improvements which still allows the TI-99/4A to fly with the 'Eagles'.

TIUP Tit BITS DISCLAIMER

Tit BITS is the OFFICIAL newsletter of the TI-99 Users of Perth (Inc.), Western Australia (TIUP).

TIUP is a non-profit group of Texas Instruments TI-99/4A computer users not affiliated with Texas Instruments or any other computer manufacturer. The views expressed in articles published in this newsletter are those of the author, and not necessarily those of the Editor, TIUP Committee or Group Members. All articles and programmes published herin are considered to be in the public domain unless notification is received to the contrary.

All care is taken that information published is accurate, but NO RESPONSIBILITY will be accepted by Tit BITS, its publishers or contributors, the TI-99 Users of Perth (Inc.) or its members for any adverse results obtained from the application of any information provided in this publication.

Articles published in Tit BITS may be reproduced by other Groups with similar aims, provided that the source and author are acknowledged. Contributions to Tit BITS are invited from both members and non-members.

Please submit any contributions to the Editor care of:

TIUP (TI-99 Users of Perth Inc).
20 Hudson St.
Bayswater 6053.
WESTERN AUSTRALIA.

OUR COMMITTEE

President:

Merve TROWBRIDGE
90 Williamson Avenue,
Belmont, 6104. W. AUSTRALIA.
Tel. (09) 2771091

Vice President:

Bill La FRENTZ
2 Daly Place,
Padbury, 6025. W. AUSTRALIA.
Tel. (09) 4013430

Treasurer:

Greg BUCK
18 Linden Street,
Dianella, 6062. W. AUSTRALIA.
Tel. (09) 2761291

Secretary:

Frank GRAHAM
20 Hudson Street,
Bayswater, 6053. W. AUSTRALIA.
Tel. (09) 2715972

SUBSCRIPTIONS

FULL MEMBER - A\$20-00 P.A.
NEWSLETTER ONLY - A\$15-00 P.A.

ADVERTISING

Members and newsletter subscribers may advertise their PERSONAL computer - related products in Tit BITS free of charge (subject to space limitations and the Editor's discretion). Commercial rates by negotiation.

BILLS BURST # 3.

Well here I am with the 3rd article in this series, which was issued as TIUPDOM/03.

(1) ACE 27 PROGRAM (XBASIC)
Source: MICROpendium March 1987.

This is from "Turning assembly into CALL LOAD's" and was downloaded from the Tex Net database on The Source. This program prompts you all the way but before you start you must know if the program you are converting returns to XBASIC or not and it must not contain any AORG's. I have not used this program as yet to obtain any CALL LOAD's that I needed but have tested it and it works very well.

(2) BWGOLF 53 INT/VAR254 (XBASIC)
Source: MICROpendium June 1991.

This is from "Bad Weather Golf" by James B. Smith. As the name suggests it is a golf game and can be played by up to 4 players at a time with each player playing the complete hole before the next player begins. You are presented with instructions for the game and if you are a golf player you may know which club to select for each shot. You get to play 18 holes of golf with a choice of 2 levels, when entering names, use no more than 4 characters and if you have a speech synthesizer you can plug it in and hear the comments the 'computer expert' makes about your game. When checking out this program I noticed some fading in and out of some of the background trees but this did not effect the running of the program. I have checked and rechecked the program but could not find what was causing it. If you can put up with this small glitch you can play the game. (Maybe someone out there can find the problem and let me know.)

(3) DISASS 32 PROGRAM (XBASIC)
Source: MICROpendium May 1992.

This is from "What's there?" by Jacques GrosLouis. Before I start, I must apologize again for an error in the program, this time it eas not my fault as the program is exactly the same as that listed in the MICROpendium. To fix the problem list

Line 520 to screen and you will see near the end of the line,
C\$=" "&(" UT\$(X)&")"
whereas it should have read,
C\$=" "&("&UT\$(X)&")"

I am not very sure how to use this program, as the write up in the MICROpendium did not really say much about it, however, I checked it out by loading and running it and it worked perfectly, with the modifications on the the error in Line 520. After the program had run for a couple of minute , I pressed FNCT 9 to stop it and it jumped back to the start of the program. I wondered how to end the program and decided to enter the same finish address code as the starting address code and it came up with "Pres Q to Quit". I then loaded the source code listing that it produced and it proved that the program did work even though I did not know what to do with the listing as I have no idea about assembly language. If you want a printout, check Line 1410 that it suit your system or you can change it when it asks you if you want a printout and lists "PIO" as the printer.

(4) PROOFREAD 7 PROGRAM (BASIC)
Source: MICROpendium June 1987.

This program is from "Put your speech synthesizer to work as a proof-reader" by Steven L. Richardson . To run this program you will need a TE II module, MERGE this program with the program you want to proofread making sure that your program doesn't start before Line 100. Next list a section of your program that you want to read to screen and then type RUN, proofreader will then read everything that is on the screen while you check it against your listing of the program. Continue this way until you have checked all of your program. The program will freak out when confronted by XBASIC exclusive commands such as CALL SPRITE or LINPUT but you can steel clear of listing them to screen and check them in the normal way.

(5) ROCKHOP 52 INT/VAR254 (XBASIC)
Source: MICROpendium January 1987.

(6) ROCKHOP/1 52 INT/VAR254 (XBASIC)
Source: MICROpendium February 1987.

(7) ROCKHOP/2 52 INT/VAR254 (XBASIC)

(continued on page 4)

ROCKHOP is from "Rock Hopper" by David Mennenoh, ROCKHOP/1 is the same program modified by W. A. Ragsdale, who added new increments for reaching levels 5,10,15 etc., also some lines to clean up the screen between games. ROCKHOP/2 was very slightly modified by me to let you start with 10 men instead of 5 as I found this game difficult to play and could not reach the higher levels before running out of men. If you want to see what the higher levels look like you could start by playing ROCKHOP/2 and then go back to playing ROCKHOP/1 but if you want to claim the all time high score, which was 13,280 at January 1987, you will have to play ROCKHOP where you do not get extra men for each 5th level that you achieve. You will need a joystick to play this game (make sure the ALPHA LOCK key is up), and you will have to jump across the gap while dodging the falling rocks. When you reached the other side of the screen you have to ring the bell to get on to the next floor. When you ring the bell on the third round, you are presented with the second screen, where you are standing on a platform, if you back the man up to get a run up the program will crash but this is easy fixed (this is another error that I must apologize for, or 3 errors as all three ROCKHOP programs have it). Go to Line 1030 and you will see CALL LOCOTE which should be CALL LOCATE, change this and it will be alright. When you jump to the next platform it is spring loaded and launches you to the next platform. Now when you jump to the next platform it starts a conveyer belt of hooks which you have to jump up and grab, this carries you over to the next platform and you press the trigger to drop off. Next, go up the ladder which sets up the top floor where you have to jump over rolling rocks to reach another spring loaded platform to launch you up to the next platform to rescue the princess, after which you return to the first screen where the rocks move slightly faster.

(8) SUPERTRACE 24 PROGRAM (XBASIC)
Source: MICROpendium May 1988.

This is from "Supertrace" by Jim Peterson. Instructions for this program appear on the screen but before you start, the program you want to run

through Supertrace must be in MERGE format. If you have multiple statement lines which are too close together you will have to resequence it to open it up as Supertrace breaks each line down into single statement lines. You will be given a choice of dumping to screen printer or both and gives you three choices of running the new program. When you do run it, depending on how you saved it, you can step through the program, with the line number being executed listed in the lower left hand corner.

(9) TICTACTOE 39 PROGRAM (XBASIC)
Source: MICROpendium December 1986.

This is from "3-D Tic-Tac-Toe" by Rick and Jill Pelly. When you run this program you are presented with the instructions but be quick to read it it only stays there while the computer is being initialized. You are then asked if you want to play first and when it is your turn type in your move like 111 to go to the bottom left hand front corner or 444 to go to the top right hand back corner. The computer can take up to 90 seconds to move and this is counted off in a little clock just under Tic-Tac-Toe. If you want to finish before the end of the game just type in 0 for your move.

(10) UNBASHER 13 PROGRAM (XBASIC)
Source: MICROpendium May 1992.

This is from "Unbasher breaks down programs automatically" by Barry Trave (MAY92). This program breaks down XBASIC multistatement lines into single statement lines except for some complex IF/ THEN constructions. This helps if you want to modify programs that have large multistatement lines, after you make your modifications you can then run it through one of the programs to compress it again, eg; XBASHER or SMASH. After you disassemble your program you could run it through Graphicomp 1.5 to turn it into an XBASIC program with imbedded assembly language in it.

(11) WINDCALC 6 PROGRAM (XBASIC)
Source: MICROpendium July 1986.

This is from "A Window Calculator for the TI" by Bill Harms. I do not have a use for this program but I

(Continued on page 14)

PROGRAMS RELATED TO NUMBER THEORY

by Tony Falco

Three of the four programs below have few enough characters to fit on one program line. They need to be longer, however, to accommodate appropriate nesting of loops and to maintain proper syntax.

The first one is a simple one and it is kind of a warm up for the others. It prints all the divisors of any positive whole number entered by the user.

A prime number is a whole number larger than one whose only divisors are one and itself. The second program prints all the prime numbers from one to one thousand.

A perfect number equals the sum of all the divisors of the number which are less than the number itself. For example, 6 is perfect because $6 = 1 \times 2 \times 3 = 1 + 2 + 3$. There are only three perfect numbers less than 500. The third program prints all of these. It takes a very very long time to do so.

The Fundamental Theorem of Arithmetic says "Every composite (not prime) number is uniquely expressible as the product of primes." The last program (one of my all time favorites) takes any number as input and prints its prime factorization. For example, the output $12 = (2A2)(3)$ means $12 = 2 \times 2 \times 3$.

```
1 PRINT ;;;; INPUT "DIVISORS OF ";N ;;
  PRINT "THE DIVISORS ARE" ;; FOR D=1 TO
  N ;; IF N/D=INT(N/D)THEN PRINT D;
2 NEXT D ;; GOTO 1
```

```
1 CALL CLEAR ;; FOR N=2 TO 1000 ;; P=0 :
: FOR D=2 TO SQR(N):: IF INT(N/D)=N/D TH
EN P=1
2 NEXT D ;; IF P=0 THEN PRINT N;
3 NEXT N
```

```
1 CALL CLEAR ;; FOR N=2 TO 500 STEP 2 ::
S=0 ;; FOR D=1 TO N/2 ;; IF INT(N/D)=N/
D THEN S=S+D
2 NEXT D ;; IF S=N THEN PRINT N
3 NEXT N
```

```
1 PRINT ;;; ;; INPUT "N=";N ;; PRINT STR
$(N);"="; ;; FOR D=2 TO N ;; P=0 ;; IF N=
1 THEN 1
2 IF N/D<>INT(N/D)THEN 3 ELSE N=N/D ;; P
=P+1 ;; GOTO 2
3 IF P>0 THEN IF P=1 THEN PRINT "(";STR$(
D);"");ELSE IF P>1 THEN PRINT "(";STR$(
D);"^";STR$(P);"");
4 NEXT D ;; GOTO 1
```

This is taken from a series of messages on the Chicag BBS and has been edited for use on the HUG bbs. William M. Lucid, Co-sysop HUG BBS

Msg#: 11 *Announcements*
95/12/18 11:24:40 pm (Read ** times)
From: Michael Maksimik #1
To: ALL
Subj: protocols

Here is a guide for all users to connect to this bbs...

If your computer is a TI99/4A, use TERM80 (a new emulator which I am currently evaluating) or TELCO, or Mass Transfer. TELCO has some nice windowing features, and it supports 80 column cards using the V9938/V9958 video processors. Mass Transfer, in all it's inception, has some unique file transfer protocols for sending bunches of files to/from the BBS.

TERM80 supposedly allows ANSI cursor emulation, in 80 columns using a standard TI display. If you have a standard V.22 1200 baud modem, you should be able to connect to this bbs with no trouble. Bell 103 modems (300 baud) are also supported, but for your phone bill's sake, upgrade! V.22bis (2400 baud) will work with the TI, but it is really the limit that the TI can display easily unless you have some type of processor upgrade, or hardware flow control (RTS/CTS signals from modem get controlled from the terminal program to prevent loss of data) I have tested TELCO at 4800 baud, and it is marginal. Super Mass Transfer, 80 columns and ANSI monochrome, is better suited to 4800 baud/9600 baud. Some 2400 baud modems have the data compression feature and this allows a terminal rate higher than the line rate. DJ, you have such a modem (2400 v.22 carrier, but V.42/V.42bis error correction and data compression)

Such a modem (2400 line rate, with compression and error correction) needs a higher host rate so it efficiently compresses/decompresses and corrects errors with no overrun errors. On such

a modem, make your terminal rate 9600 baud, which is 4 times the line rate. It is rare that the modem will generate long sequences of 9600 baud data, but the decompression can produce short bursts of 9600 baud packets, esp. with text data which is highly compressable. Make sure you use ITU-T V.42/V.42bis (or LAP-M) protocols. Don't use MNP protocol unless you have to for maintaining a stable connection. Most phone lines now can handle higher baud rates, and since this BBS supports a maximum carrier of 14,400, you may want to invest in at least a modem of this type. Computer faires are practical giving away V.22bis/V.42bis modems (2400 baud modem with compression)

The TI and 9600 baud

It is very difficult for the TI to keep up with 9600 baud, not that it is a slow computer, but because most terminal programs use the video circular buffer for receiving characters. When a character is sent through the RS23 port, the interrupt routine will grab the character, and put it into the video RAM. This is the secret of how TE-II works.

TI is slow in this routine for a couple of reasons. First, there is only ONE IRQ line to receive interrupts from ALL peripherals. A second line was planned for the TI99/8, and an interrupt ID scheme was developed but never implemented, allowing faster detection of the interrupt. The problem occurs when an interrupt is detected, the TI does not know which peripheral generated it. It must then turn on each card and pass control to the card so that it can check itself for a possible interrupt.

Some cards implement an interrupt routine consisting only of a return instruction, other cards actually check for a data byte. The RS232 card checks for interrupts on EVERY port (serial and parallel ports) and if you have two RS232 cards, that means six ports are checked for an incoming character. If you have a DIGIT video card, it also has an interrupt routine, and generally, all cards are scanned.

(continued on page 7)

Some cards (like the Myarc 512k card) does some type of bank switching on interrupts, the bottom line is that the computer is doing a lot of unnecessary searching for one little character, which takes a lot of processor time.

The problem is compounded when we actually get the character, we need to store it somewhere. Well, the TI wasn't made with any large amount of CPU ram, so we must store it in slow VDP ram.....

VDP RAM access is byte oriented, and it is pretty slow. Having a faster video (even if it is in 40 column mode) helps...a V9938/V9958 processor is a much better choice because it can handle data faster from the CPU.

You can also do some other things to speed system up a bit:

1) Get the WHT keyboard module, which includes an updated system ROM, containing a much better interrupt scanning routine. It is more efficient than the old console routine, and the new unit allows use of new PC keyboards (101/102 key keyboards), and it adds some usable system memory.

2) You can boost processor speed during terminal I/O by having a switchable 16 Mhz crystal. The TI uses a 12 or 48 mhz crystal. The 48 Mhz crystal is used with the TIM9904 chip. You can substitute a TIM9904A, and use the 12 Mhz crystal instead, and use a switch to "turbo" the system to 16 mhz. Be aware that this will screw up baud rates, so use TELCO, which allows a custom calculated baud rate taking into account the increase in system clock speed.

3) Use some flow control. Although there are no terminal programs available which use flow control (TERMBO may use it, I haven't found out yet) this allows a high line rate, and controlled terminal rate so you don't lose data, and it is easier to start/stop the flow of data.

On the Geneve, I have been able to use Super Mass Transfer in GPL mode, with much better results. In fact, if you combine this with TEXEC, which is EXEC

modified to allow use of a single interrupt input from ONE RS232 port (serial port 1) then the only scanning done on interrupt (EXTINT) will occur to serial port 1. This devotes more processor time to useful scanning for characters, and you can actually connect at a rate of 9600 (true rate) with a line rate of 9600 if you like. I have tried higher rates, but TEXEC doesn't really keep up at higher than 12,000 bps, and many modems don't allow those terminal rates.

You have to understand that line rates (300, 1200, 2400, 4800, 7200, 9600, 12000, and 14,400) are rates negotiated on the line itself, between the two modems. Your terminal rate must be one of the following (300, 1200, 2400, 4800, 9600, 19200, 38400, 57600, 115,200)

If your modem connects at 14400 bps, you must have a terminal rate of at least 19200, to allow for software compression and error correction to work properly. Since at these rates, interrupts for programs like TELCO, MASS80, superMASS80, GEN-TRI, etc. usually fail, you must keep the rate at 9600.

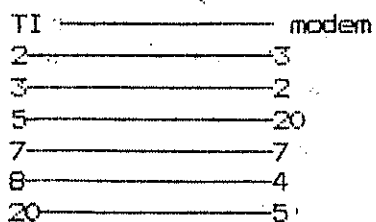
If you implement flow control on the modem, this forces the modem to buffer data while the computer is busy. On FC's, the buffer is usually built into the 16550AF uart, which has a 16, 32 or 64 byte buffer for receiving character overruns when the CPU is busy doing something else, like drawing a screen. For our geneve, we have no buffer and the CPU must read each character before the next one comes in or we lose data.

Tim has made FORT work with the interrupt routine in native MDOS mode, using high speed system memory to allow successful interrupt scanning and video (monochrome) update. Color screens at that high speed require a generally slower video mode, and a color interpreter. So you need to enable FLOW CONTROL and the modem must support this feature if it a high speed modem. Be sure your modem has HARDWARE FLOW CONTROL, which uses the RS232 pins 4 and 5 to control the flow of data.

This will allow you to use FORT's

(Continued on page 8)

enhanced color ANSI features and high speed YMODEM-G file transfer routines, and also high baud rates which most new modems use and require for successful data compression and error correction. I normally run FORT at 38,400 with full flow control using the "MIKE" cable design.



The pin 2 is data from rs232 card to modem pin 3 is data from modem to rs232 card pin 5 of rs232 is used to control DTR signal on modem, for fast hangup pin 7 is ground on both modem and rs232 card pin 8 of rs232 represents request to send flow control. Internally, the TI rs232 has one lead cut, and jumper soldered. This allows FORT an additional control line to switch this wire on and off independently of whether or not we want to send data. This line is like a faucet which shuts off the modem from sending us data, and the modem can thereby tell the remote system to stop sending data in turn. pin 20 of the TI is the inbound clear to send flow control. It is used by the modem to stop the Ti rs232 from sending data. If the modem encounters a line error, or if the distant modem stops flow, then this

The distant modem may encounter errors, or the modems may negotiate data packets or may be in the process of ranging. These situations must prevent either system from sending data so both clear to send lines on both ends are deactivated. These lines (RTS and CTS) are very important, because they allow a faster serial device to access a slower computer or host, and the flow lines prevent loss of data. They allow your computer to accept data as fast as it can receive and they allow your computer to send data as fast as the remote system can handle reception.

Jeff's cable design is similar to mine, but he uses pin 5 to activate the flow control, and pin 8 is used to handle DTR hangup, or you just disable DTR hangup in other programs which toggle that line during transmission.

In other words, if you use TELCO and FORT, with jeff's cable, you have to disable DTR hangup in TELCO and use the three plus character and ATH to hangup. My cable (mike) does not require this change and you can still use DTR hangup in FORT, Mass Transfer, and TELCO with no changes.

Jeff's cable design is ideal for those who have a non-Ti card, and who do not wish to make the mod to their rs232 card to add the extra CRU line.

Msg#: 342 *General Messages*
 95/11/08 12:32:33 am (Read 13 times)
 From: William M. Lucid #1
 To: ALL
 Subj: v9t9 Cables

The Hoosier User's Group has eight cables left from the Chicago Faire that are for use with Edward Swartz's v9t9 (TI Emulator) program. These cables are 15 foot serial cables with male/female connectors, plus an special adapter for proper handshaking. Cable and adapter are required for dumping rom/grom from TI 99/4A sytem to the IBM pc. Price of the cable is \$20.00. Shipping and handling \$5.00 or pick at a HUG meeting for no shipping and handling.

If you are not in the Indianapolis area, order from William M. Lucid, 6005 Elaine Street, Speedway, IN 46224-3032.

Make check or money order out to: Hoosier User's Group.

These adapters have been tested and known to be functional. Bye now before they are gone!!!

* ORIGIN: HUG TI 99/4A BBS (317)-782-9942 EN1 / 24 hours [0/0:0]

PROGRAMING TIPS

By Paul E. Scheidemantle

I'm certain that most of you may have mastered the CALL HCHAR statement with all it's niceities. If so STOP READING NOW... IF NOT READ ON!! lets look at this very versatile statement. With it you can put anything anywhere on the screen that you want; when coupled with other commands. The normal format of the CALL HCHAR statement is CALL HCHAR(A,B,C,D). Where A = the row, B = the column, C = the character number, and D = the number of times you wish to put it on the screen. Well when coupling it with a FOR NEXT LOOP and the ASC, LEN, and SEG\$ commands we can write a subroutine that will display at anywhere on the screen without scrolling. A full 32 columns and 24 rows!

Shown below is such a routine. We will learn what makes it work. First the routine starting at line 5000, & then the small portion at lines 1000-1060 (which is used to demonstrate the routine).

```
1000 REM TEST DISPLAY AT ROUTINE
1010 CALL CLEAR
1020 A=1
1030 B=1
1040 MSG$="Hello from: P & A
Software withthis test o
f the display at routine..
IT FINALLY WORKED*$$#!"
1050 GOSUB 5010
1060 GOTO 1060
5000 REM DISPLAY AT ROUTINE
5010 FOR LOOP=0 TO LEN(MSG$)
-1
5020 ROW=A+INT((LOOP+B-1)/32
)
5030 IF ROW>24 THEN 5070
5040 COL=B+LOOP-(32*INT((LOOP+B-1)/32))
5050 CALL HCHAR(ROW,COL,ASC(
SEG$(MSG$,LOOP+1,1)))
5060 NEXT LOOP
5070 RETURN
```

Starting at line 5000. We have a REM statement; all this does for us

is identify the routine. In case we want to find it easily later on in a large program.

Now the loop at line 5010 is set so that you can step through the message one character at a time. The string function LEN will tell the computer how many characters are in the string MSG\$. Normally you would want the loop to go from 1 to 90 (or however long the string is) but to keep all the equations working it is necessary to go from 0 to the length of the string. Thus giving us the need to subtract 1.

```
5020 ROW=A+INT((LOOP+B-1)/32)
```

Line 5020 sets up the row that the character from the string is to be printed at. In the base program we sent a value of 1 for A that we wanted to start on row 1. But that's great providing you never print anything longer than 32 characters. This line will reset the row to the next line when loop divided by 32 (the number of characters that fit on a line) is reached.

ROW [row character is to be placed at] = A [row that we want to start at] + INT [integer value/those numbers on the left side of the decimal point only] ((LOOP [for next loop to step the row and column and determine which character to print] + B [starting point/column location] -1)/32 [number of characters on a line]).

So lets put some numbers in the equation to test it. A = 1; B = 1; LOOP = 0. Now figure it out $1 + \text{INT}((0 + 1 - 1) / 32) = 1$. Thus ROW = 1. Now lets use a larger value for B (starting in the middle of the screen) A = 1; B = 16; LOOP = 16. $1 + \text{INT}((16 + 16 - 1) / 32) = 1$ OR ROW = 1. However had the loop value been 17 the ROW would have equaled 2; and so on.

Line 5030 is a trap for the row number, should it be greater than the exceptable number of 24. It works like this.... If the value of ROW is greater than 24 then it sends you to line 5070 (return to main program with no action). Otherwise the state

(continued on page 10)

ment is ignored, and control passed to the next line.

```
5040 COL=B+LOOP-(32*INT((LOOP+B-1)/32))
```

Line 5040 sets up the column that the character from the string is to be printed in. In the base program we sent a value of 1 for B (meaning that we wanted to start in col 1). This line will reset the col to the next col, and back to 1 when the loop plus the B value -1 divided by 32 (the number of characters that fit on a line) is reached.

COL [column character is to be displayed at] = B [col that we want to start at] + $LOOP$ [next character location] - (32 * INT [integer value /those numbers on the left side of the decimal point only] (($LOOP$ [for next loop to step the row and column and determine which character to print] + B [starting point column location] - 1)/32 [number of characters on a line])).

So lets put some numbers in the equation to test it. $A = 1$; $B = 1$; $LOOP = 0$. Now figure it out $1+0-(32*INT((0+1-1)/32))=1$. Thus $COL=1$. Now lets use a larger value for B (starting in the middle of the screen) $A = 1$; $B = 16$; $LOOP = 14$. $16+14-(32*INT((14+16-1)/32)) = 30$ OR $COL = 30$. However had the loop value been 17 the COL would have equaled 1; and so on.

```
5050 CALL HCHAR(ROW,COL,ASC(SEG$(MSG$,LOOP+1,1)))
```

Line 5050. We see the CALL HCHAR statement. The first two parameters of ROW and COL have been discussed as to how we get these values. Now we need to understand how to get the ASC II value required for the third parameter.

First you must understand what

ASC and SEG\$ do. When a letter is placed in quotes inside parentheses behind the ASC command; will return the ASC II value of the character. For example ASC("A") would return a value of 65. The SEG\$ command has the purpose of getting segments of strings out of strings, and works likes this. Its format is SEG\$(A\$,S,H). The A\$ would be the string that you want to use; the "S" would equal the starting location; and the "H" would be how many characters to get. This could be 1 character or all of the string (depending on the length)

In this case we trying to obtain the ASC II value of a segment of the MSG\$ string at the starting point of the LOOP+1 (since we start at 0) and 1 character long to display on the screen. The values obtained from lines 5010, 5020, and 5040 now give us enough information to use the CALL HCHAR statement. The 1st time thru the loop we would have a statement like this. CALL HCHAR(1,1,72) next like this. CALL HCHAR(1,2,69).

Line 5060 is the NEXT statement. This causes the loop to continue until the top half of the parameter is reached. Then control is returned to the program and will pass through to the next line.

Line 5070 returns us from the the subroutine to the main program.

Now the first part of the program starting with Line 1010. 1010 clears the screen. 1020 set the value of A to 1 (the starting location that we want for the row). 1030 set the value of B to 1 (the starting location that we want for the column). 1040 is the message that we want to display on the screen. 1050 branches to the subroutine and then back to 1060 to keep the message on the screen from scrolling. Well I've run out of room so I'll wish you HAPPY COMPUTING!!!!

M U L T I P L A N

CONTROL COMMANDS

HOME 1	TAB 2	NEXT CELL 3	NEXT CHAR 4	NEXT WORD 5	NEXT WINDOW 6	REL/ ABS REF 7	8	9	CANCEL 0	=
Q	W	E	R	T	Y	U	TAB I	BACK WORD O	NEXT WORD P	
MOVE CURSOR A	S	D	F	G	DEL CHAR H	J	BACK CHAR K	NEXT CHAR L		
	Z	X	C	V	B	N	ENTER M	,	.	
CTRL										FCTN

FUNCTION COMMANDS

LOWER RIGHT 1	2	3	BACK CHAR 4	BACK WORD 5	6	HELP 7	RECALC 8	BACK SPACE 9	DEL NEXT 0	=
---------------------	---	---	-------------------	-------------------	---	-----------	-------------	--------------------	------------------	---

MENU COMMANDS

QUIT Q	WINDOW W	EDIT E	R	T	Y	U	HELP I	OPTION O	PRINT P
ALPHA NUMERIC A	SORT S	DELETE D	FORMAT F	GOTO G	HELP H	J	K	LOCK L	
	Z	X	C	V	B	N	M	.	
CTRL									FCTN

Pointer Movement

Cell	Window
Funct - E - Ctrl	
Funct - S - Ctrl	
Funct - D - Ctrl	
Funct - X - Ctrl	

Error Values

#DIV/0!	Attempt to divide by 0
#Name?	Undefined label reference
#N/A	Value not available
#NULL	Disjointed areas as intersections
#NUM!	Not number/too big/small
#REF!	No relative reference
#VALUE!	Use of text

QUOTES -(WAN)

A string of embarrassing quotes have been floating about on the Internet, the sort which come back to haunt their authors:

"640K ought to be enough for anybody" - Bill Gates, 1981.

"Computers in the future may weigh no more than 1.5 tonnes" - Popular Mechanics, 1949.

"I think there is a world market for maybe five computers" - Thomas Watson, Chairman of IBM, 1943.

Bits, Bytes & Pixels

THE MEANING OF FUNNELWEB'S FILES

by Charles Good
Lima Ohio User Group

Listed below are the Funnelweb files on our two 3 disk sets of complete 40 and 80 column sets. Next to each file is Charles Good's DSKU comment detailing the function of that particular file. These are all the current files from v4.4 and v5.01 for 40 columns, and from v4.4 and v5.21 for 80 columns. Any of the original v4.4 files not listed below are now obsolete. Any user can obtain either or both of these 3 disk sets by sending 3 disks and a paid return mailer to me at P.O. Box 647, Venedocia Ohio 45894.

40 Column files, Disk 1 of 3. Main system files

Filename	TYPE	Comment
AS	PGM	Assembly code ASSEMBLER part 1
AT	PGM	Assembly code ASSEMBLER part 2
C1	PGM	Char set for use by central menus
C2	PGM	Another char set for central menus
CF	PGM	Configures all funnelweb but editor
Cb	PGM	part 2 of funnelweb config program
(NOTE: 8 bit char sets have All Chars graphics)		
CHAR01	PGM	USA 8 bit IBM graphics char set
CHAR02	PGM	British 8 bit IBM graphics charset
CHAR03	PGM	French 8 bit IBM graphics char set
CHAR04	PGM	German 8 bit IBM graphics char set
CHAR05	PGM	Italian 8 bit IBM graphics char set
CHAR06	PGM	Swedish 8 bit IBM graphics char set
CHAR07	PGM	Dutch 8 bit IBM graphics char set
CHAR08	PGM	Spanish 8 bit IBM graphic char set
(NOTE: These 7 bit char sets used in Eurowriter mode)		
CHARA1	PGM	British 7 bit national char set
CHARB1	PGM	French 7 bit national char set
CHARC1	PGM	German 7 bit national char set
CHARD1	PGM	Italian 7 bit national char set
CHARE1	PGM	Swedish 7 bit national char set
CHARF1	PGM	Dutch 7 bit national char set
CHARG1	PGM	No Spanish nationa 7 bit char set
CONF16/40	D/V	Expanded editor config data file
D1	PGM	"Disk Utilities" user list
DR	PGM	40 column Disk Review, part 1
DS	PGM	40 column Disk Review part 2
EA	PGM	Needed on disk for FW loaders work
ED	PGM	8 bit v5.01 40 column editor, part 1
EE	PGM	8 bit v5.01 40 column editor, part 2
F4TIAE	PGM	British 40 column editor command line
F4TIBE	PGM	French 40 col editor command line
F4TICE	PGM	German 40 col editor command line
F4TIDE	PGM	Italian 40 col editor command line
F4TIEE	PGM	Swedish 40 col editor command line
F4TIFE	PGM	Dutch 40 col editor command line
FO	PGM	Text formatter part 1
FP	PGM	Text formatter part 2
FW	PGM	MAIN FW PROGRAM FILE, boots as EAS

HELP4A	PGM	40 col help screen
HELP4B	PGM	40 col help screen
HELP4C	PGM	40 col help screen
HELP4D	PGM	40 col help screen
HELP4E	PGM	40 col editor help screen
HELP4F	PGM	40 col editor help screen
HELP4G	PGM	40 col editor help screen
HELP4H	PGM	40 col editor help screen
HELP4I	PGM	40 col editor help screen
HELP4J	PGM	40 col editor help screen
INSTALL/ED	PGM	40 col editor config software
LL	PGM	LOW LOADER, needed to boot LL files
LOAD	PGM	MAIN FW PROGRAM FILE, boots from XB
ML	PGM	40 column MULTI LIST user list
QD	PGM	QUICK DIRECTORY, use with formatter
QF	PGM	part 2 of QUICK DIRECTORY
SL	PGM	Needed to boot script load files
SYSCON	PGM	CF/C6 configuration data file
UL	PGM	Central menu "User List"

40 column files. Disk 2 of 3. Other system files, docs

Filename	Type	Comment
-READ-ME	D/V	General v4.4 overview doc.
424REVIEW	D/V	C. Good's review of v4.4
4PRINTFILE	D/V	Sample source for editor help file
5-01REVIEW	D/V	C Good's review of v5.01 editor
5-0REVIEW	D/V	C Good's review of v5.0 editor
AR	PGM	Archiver v3.03
C99PF1j0	D/F	Used with C99 from within funnelweb
CHARA1	PGM	Char set for use with DSKU
CHARUTIL	PGM	Converts custom chars to 8 bit set
CHRCOAL/S	D/V	Char set source code sample
CON/ED	D/V	Condensed editor config data file
CP	PGM	Boots c99 & returns to funnelweb
CTBK/O	D/F	Boots funnelweb from supercart
DU	PGM	Birdwell's DSKU v4.2 part 1
DV	PGM	DSKU v4.2 part 2
DW	PGM	DSKU v4.2 part 3
ED-BASIC	PGM	Basic 7 bit 40 col editor, part 1
EE-BASIC	PGM	Basic 7 bit v5.01 40 col editor pt2
FSAVE	D/F	SAVE utility to make EAS programs
FWDOC/EASW	D/V	PROGRAM EDITOR doc
FWDOC/LOAD	D/V	The XB LOAD menu doc
FWDOC/REPT	D/V	Bug and update report of FW & LOAD
HELPHAKE40	PGM	Turns DV80 source into 40col ed help
LDFW	D/F	Boots FW from Minimee or EA module
H6	PGM	DM1000 v3.5 part 1 of 2
MH	PGM	DM1000 v3.5 part 2
SCRIPT	D/V	Sample SL file
XB41HLD	PGM	Boots TI FORTH from XB menu

40 column files. Disk 3 of 3. Documentation files

Filename	Type	Comment
FWDOC/DR40	D/V	40 col Disk Review doc pt 1
FWDOC/DR41	D/V	40 col Disk Review doc part 2

(Continued on Page 13)

Bits, Bytes & Pixels

FWD0C/EAS0 D/V PROGRAM EDITOR doc
 FWD0C/ED40 D/V v5.01 40 col editor doc part 1
 FWD0C/ED41 D/V v5.01 40 col editor doc part 2
 FWD0C/ED42 D/V v5.01 40 col editor doc part 3
 FWD0C/PSRV D/V Assembly links to FW doc
 FWD0C/SCLL D/V SL,UL,LL, and ML40/80 doc
 FWD0C/UTIL D/V CF,CP,FSAVE,LDFW,UL,LH,CTBK/O doc

80 column system/Disk 1 of 3. Main system files
 Filename Type Comment

```
AS      PGM Assembly code ASSEMBLER part 1
AT      PGM Assembly code ASSEMBLER part 2
C1      PGM Char set for use by central aenus
C2      PGM Another char set for use by central
CF      PGM Configures all funnelweb but editor
C6      PGM part 2 of funnelweb config program
(NOTE: The 8 bit char sets have IBM graphics)
CHAR01  PGM USA 8 bit IBM graphics char set
CHAR02  PGM British 8 bit IBM graphic char set
CHAR03  PGM French 8 bit IBM graphic char set
CHAR04  PGM German 8 bit IBM graphic char set
CHAR05  PGM Italian 8 bit IBM graphic char set
CHAR06  PGM Swedish 8 bit IBM graphic char set
CHAR07  PGM Dutch 8 bit IBM graphic char set
CHAR08  PGM Spanish 8 bit IBM graphic char set
(NOTE: 7 bit char sets are for Eurowriter code)
CHARA1  PGM USA 7 bit char set
CHARB1  PGM British 7 bit national char set
CHARC1  PGM French 7 bit national char set
CHARD1  PGM German 8 bit national char set
CHARE1  PGM Swedish 7 bit national char set
CHARF1  PGM Dutch 7 bit national char set
CHARG1  PGM Spanish 7 bit national char set
DR      PGM 80 column Disk Review part 1
DB      PGM 80 column Disk Review part 2
EA      PGM Needed on disk for FW loaders work
ED      PGM v5.21 80 column editor, part 1
EE      PGM v5.21 80 column editor part 2
EF      PGM v5.21 80 column editor part 3
FBTBE   PGM French command line for v5.21editor
FBTCE   PGM German command line for v5.21editor
FBTDE   PGM Swedish command line for v5.21editor
FO      PGM Text formatter part 1
FP      PGM Text formatter part 2
FW      PGM MAIN FW PROGRAM FILE, boots as EAS
HELPBP  D/V Program editor help file for v5.21
HELPBQ  D/V Text editor help file for v5.21
LOAD    PGM MAIN FW PROGRAM FILE, boots from XB
ML      PGM 80 column MULTI LIST user list
QD      PGM QUICK DIRECTORY use with formatter
QF      PGM part 2 of QUICK DIRECTORY
SYSCON  PGM CF/C6 configuration data file
UL      PGM Central menu "User List"
```

80 column system/Disk 2 of 3. Other system files, ed dc
 Filename Type Comment

```
5-01REVIEW D/V C Good's review of v5.01 editor
AR      PGM No Archiver v3.03
C9YPF1;0 D/F Used with C99 from within funnelweb
CHARA1  PGM Char set for use with DSKU
CON/ED  D/F Short version of CONFIG/ED
CONFIG/ED D/V Configuration data/use with INSTALL80
CP      PGM Boots c99 & returns to funnelweb
CTBK/O  D/F Boots funnelweb from supercart
DU      PGM DSKU v4.2 part 1
DV      PGM DSKU v4.2 part 2
DW      PGM DSKU v4.2 part 3
FSAVE   D/F SAVE utility to make EAS programs
FWD0C/E128 D/V Doc for v5.21 80 column editor
INSTALL80 PGM Configures v5.21 80 col editor
LDFW    D/F Boots FW from Minimec or EA module
LH      PGM LINE HUNTER assembly utility
LL      PGM LOW LOADER, needed to boot LL files
MG      PGM DM1000 v3.5
MH      PGM DM1000 v3.5
SCRIPT  D/V Sample SL file
SL      PGM Needed to boot script load files
XB4THLD PGM Boots TI FORTH from XB menu
```

80 column system/Disk 3 of 3. Docs
 Filename Type Comment

```
-HEAD-RE D/V General v4.4 overview doc
434REVIEW D/V C. Good's review of v4.4
FWD0C/DK80 D/V column Disk Review doc
FWD0C/DR01 D/V column Disk Review doc
FWD0C/DR02 D/V column Disk Review doc
FWD0C/EAS0 D/V Editor/assembler doc
FWD0C/LOAD D/V The XB LOAD menu doc
FWD0C/PSRV D/V
FWD0C/REPT D/V Bug and update report of FW & LOAD
FWD0C/SCLL D/V SL,UL,LL, and ML40/80 doc
FWD0C/UTIL D/V CF,CP,FSAVE,LDFW,UL,LH,CTBK/O doc
```

DONE

HUG S&T BBS
 Hoosier User's Group
 300/1200/2400/4800/9600
 Baud, 24 Hours Daily
 317-782-9942 8N1

Now with 80 Megs Storage

=====

PRINTED MATTER

ONLY

=====



TO:

TIBUG

PO BOX 3051

CLONTARF MDC 4019, QUEENSLAND.

SENDER:

SECRETARY - TIUP(INC.)

20 HUDSON STREET,

BAYSWATER 6053.

WESTERN AUSTRALIA.

(Continued from page 4)

thought it might be usefull to someone. This program is meant to be MERGED into another program like a budgeting or checkbook program and you add an option to the menu to call up the subprogram. After you have done your calculations you exit back to the main program and whatever was on the screen where the calculator was is returned. The Window- Calculator works like any 10-key calcul- ator using the standard maths signs of the TI. You press "E" to exit and "C" to clear it for a new problem. This program will also run by itself if you want to check it out.

Well that is the end of all the programs on this disk so I am comming to an end of this burst. I hope the errors encounted in the programs that I have typed up did not cause anyone any inconvenience but I did not expect the disks to be sent out to anyone outside the TIUP Group. If any member of our Group has found an error, would they please ring me or wait till the next meeting and let me know, so that I can check it out and get back to them. Well I think that is all there is for now so I will say good-bye and happy Tling.

Bill La Frenz.

INDEX

TITLE	PAGE	TITLE	PAGE
Bill's Burst #3 (TIUP/DOM02)	3	Advertising	2
Bits Bytes and Pixels	12	Our Committee	2
Cover Storey	1	Subscriptions	2
Editorial	2	TIUP Tit BITS - Disclaimer	2
Multi Plan Layout	11		
Program Tips	9	Quotes	11
Programs related to Numbers	5		
Protocols (connecting to bbs)	6		