

INTERNATIONAL TI-LINES

Volume 4, Issue 3

August 1st., 1987

SUBJECT
INDEX

PETER
KILLICK

SUBJECT
INDEX

Formerly OXON TI USERS

A PUBLICATION
OF THE
INTERNATIONAL
TI USER GROUP

Tel. 023571 2360
**WESSEX
PRESS**

9-11 Mill Street, Wantage, Oxon OX12 9AB.

PETER G. Q. BROOKS
96 BANBURY ROAD
OXFORD OX2 6JT

OXFORD 510822

I n d e x t o a r t i c l e s

P a g e

READ THIS FIRST !.....3

A (short) editorial by another name

NATIONAL TI USERS WORKSHOP III.....5

Organised by the WEST MIDLANDS TI USER GROUP and supported
by ITUG

NATIONAL TI USERS WORKSHOP III - MAP.....7

NATIONAL TI USERS WORKSHOP III - RECEPTION.....8

TI-LINES SUBJECT INDEX.....9

PETER KILLICK's subject index to three volumes of TI-LINES

GRAPHICS REPORT.....25

On improving the density of quadruple density bit images

NOTICE BOARD.....27

If you're a Myarc owner, you could be in line for some freebies!

LETTERS.....27

JOHN STOCKS on his Word Processor and Standing Wave demonstrator

INSTRUCTIONS.....28

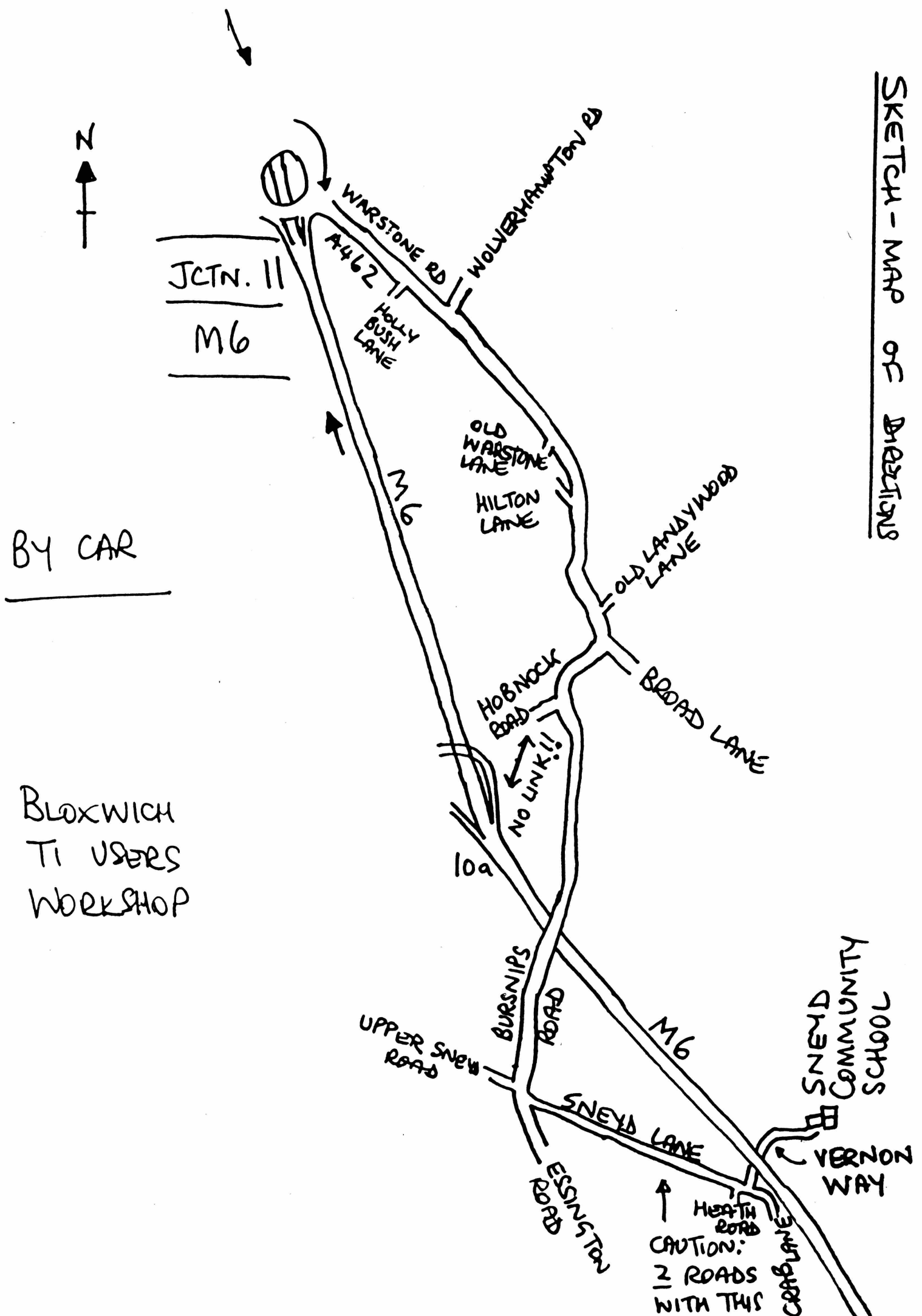
Reproduced by permission, JOHN STOCKS Word Processor. If you
find it interesting, write to him

MULTIPLAN TUTORIAL.....30

BRENDA NOELL continues with Chapters 5 and 6

SEE YOU AT BLOXWICH ON SATURDAY, SEPTEMBER 5TH! COME AND SAY HELLO!

SKETCH - MAP OF DIRECTIONS



BY CAR

BLOXWICH
TI USERS
WORKSHOP

↑ HEATH ROAD
CAUTION:
2 ROADS
WITH THIS



Reception

THE BLOXWICH WORKSHOP

SATURDAY, SEPTEMBER 5th., 1987

10.00 AM TO 6.00 PM

ENTRANCE FEE PER PERSON : £1.50 IF BOOKED AND PAID

£3.00 IF PAID AT THE DOOR

**SUBJECT
INDEX**

**SUBJECT
INDEX**

**PETER
KILLICK**

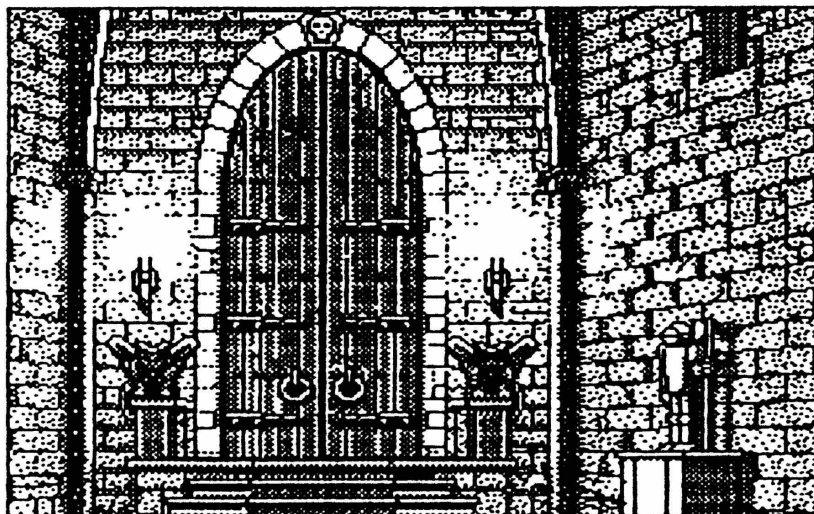
PETER KILLICK has kindly produced a subject index for Volumes 1 to 3 inclusive of INTERNATIONAL TI-LINES.

He has used a simple method of reference for each subject, which provides the Volume number, issue number, and page number in an encoded form.

In any reference, the first digit is the Volume number, the next pair of digits is the issue number, followed by a period, and then the page number.

Thus, 206.9 is: Volume 2, Issue 6, page 9; 310.26 is: Volume 3, Issue 10, page 26.

The index will be expanded as each future volume is produced.



A

Adventure Help - see Games	
Alpha lock on/off audible indicator (EA)	209.8
ASCII explained	101.7
ASSEMBLY LANGUAGE :	
Assembly language basics	206.9
Introduction to Assembly language	114.22
Changing E/A screen colours	114.14
McComic book reviewed	110.8
Programs (source code) :	
Audio Spectrum Analyser	308.30
Audio Spectrum Light Show	308.32
Sound Analyser III	308.30
Routine to drive PIO	301.21
Axiom Parallax Connector	202.20

B

BASIC (TIB) :	
Absolute Beginners Basic (read first)	
part I	305.28
part II (i)	307.9
part II (ii)	310.26
Beginner's Basic (read next; c = commands, s = statements)	
I Introduction	102.4
II Variables	103.5
C. RES, LIST, RUN ..	
S. LET, PRINT	
III C. EDIT	104.12
S. INPUT, GOTO, CALL CLEAR	
IV S. IF... THEN... ELSE	105.9
Relational Operators	
V Number Functions :	106.11
RND, INT, SQR, STR\$, CHR\$	
S. RANDOMIZE	
VI String Functions :	107.8
LEN, ASC, POS, SEG\$, VAL	
Basic Programs :	
Check program for TIB or XB running mode	311.6
Demo programs :	
Colour Squares	107.7
Shades of the TI	305.27
Games - see title listing under Games	
Routines :	
Delete disk files	210.8
- more info	211.12
Move char around screen	204.17
Multi - base number conversion	311.28
Sprites : accessing with TIB	208.11

C

'C' PROGRAMMING LANGUAGE	
C.99 review	212.27
C.99 i/o functions	304.19
C reference book list	212.23
CALL CHAR	
Character set redefined without using prog. memory	208.15
- errata	210.35
Memory saving file routine for redef. chars	112.14
- errata	113.6
Memory saving tips	106.7
CALL CHARPAT	
Hex code table for full standard char. set	107.11
CALL COINCIDENCE	
Better method using CALL PEEK	110.3
CALL KEY	
Bugged statement	101.12
Echoing key press on screen	108.19
Use to circumvent 'Alpha lock down'	108.9
- more info.	109.21
CALL LOAD locations	113.45
- errata	202.5
CALL PEEK locations	113.46
- errata	202.5
CALL SOUND	
Simple scale playing routines	105.21
CALL SPRITE	
Simple control routine	104.6
CASSETTE STORAGE	
Compatible recorders list	204.8
OLD : user def. chara to watch prog. loading	211.26
Problems with later consoles	206.4
SAVE CS1 : hints (TIB only, not XB)	102.12
Concatenation (&)	310.29
CONSOLE	
Command execution timings (TIB vs. XB)	106.23
GPL link routine for all modules (source code)	207.22
- using GPL LINK (subroutines)	207.25

GROMS description and uses	305.15
Inner workings - see Control & Function keys	
Internal RAM expansion - see D-I-Y	
Keyboard schematic drawing	210.25
Load interrupt switch - see D-I-Y	
Program storing : in TIB & XB	105.12
Variables (Symbol table)	301.14
Where your BASIC prog. is stored	113.51
Servicing : Keyboard cleaning	114.39
Module port	301.21
RAM chip repair	301.19
Video output circuit	113.29

CONTROL & FUNCTION KEYS

Part I : Understanding ASCII	101.7
Tokens explained	
Part II: Token tables	102.8
Part III: Embedded data	103.10
Part IV: Line number indicator	104.7

Conversion

Multi-base number converting prog. (TIB)	311.28
--	--------

Copyright Protection - see Security

Cursor

Redefinition routines (XB)	113.45
----------------------------	--------

D

DAI computer	107.28
--------------	--------

Debugging

Fix for Super Bugger/Navarone Bug Fixer	202.18
Hidden space error	109.5

DEF function	205.13
--------------	--------

Disk Fixer - see Forth

DISK OPERATING SYSTEM

Accessing diskettes by sector (EA)	206.14
Add a pair of disk drives	311.9
Disk Bit Map (sector '0' info)	113.14
DC power supply	113.56
- more info	114.37
Disk Controller Software Tutorial :	
I Data access; diskette format	302.23
II Diskette format concl.	303.22
III VDP memory usage	304.28
IV DSR subprograms	306.25
V DSR subprogams concl.	307.25
Disk Controller - undocumented calls	109.7
Disk Copier prog. (Forth)	113.57
- errata	114.8
Disk Directory Entry (sectors 2-21)	203.19

- addendum (Fractured File decoding)	203.23
- more info see File Descriptor Block Below	
Disk Directory Sector Review	114.24
- errata	203.6, 203.25
Disks & Drives : the basics	114.32
Disks & Drives : operation explained	207.12
- full height mechanically superior to half height	206.5
Disk Manager : reason for slow operation	301.23
Disk Manager : speed up multi-disk initialisation	208.10
Disk Memory System manual error	212.4
Diskette Repair Service	109.17
File Descriptor Block	113.26
- errata	114.8
File recovery using LINPUT (XB)	108.12
Flippies : homemade (see also Disk Basics 114.32)	205.19
Full height drives mechanically superior	206.5
Initialising diskettes (EA)	206.15
MPI drive problem solving	303.11
PEB P.S.U. for additional drives	201.13
Program storage formats	210.31

DO-IT-YOURSELF PROJECTS

Add 8K RAM to EA module	206.13
- more info	207.7
Expansion system complete	311.9
Flippies	205.19
Horizon RAMdisk - facts	212.18
- parts list	212.17
Load Interrupt Switch (SS)	114.27
- More info.	201.8
Matchbox 32K RAM expansion	209.9
wiring diagram revision	210 centresprd
more wiring info	301.22
CMOS chip suppliers	209.19, 210.7
review and construction hints	209.20
warning info	210.5, 210.6
IC chip reduction for XB	301.20
Numeric Keypad	202.16
Standalone Disk Controller conversion to DSSD	310.18
- many errata	311.3

E

EDITOR/ASSEMBLER - see also ASSEMBLY LANGUAGE

Add 8K RAM to EA module	206.13
- more info	207.7

ENHANCED BASIC - introductory paragraph

Applications :	
Handling PRK files	204.9
Loading & Saving arrays	203.18
Progs that run without a printer	206.16
(Retrieve 206.18; & Bookaccount 206.20)	
Word processing	205.10
Bugs:	
Avoiding FCTN/BACK module bug	113.12
In record altering routine	114.37
Correct way of reserving memory	114.37

Documentation (from TI) :

ACCEPT		
DISPLAY		107.23
PREP		107.26
HEADER		108.16
GETPUT		109.18
LOAD		113.36
SAVE		114.25
Tutorial I	ACCEPT AT, DISPLAY AT	114.26
Tutorial II	PREPARATION subprog.	107.12
Tutorial III	HEADER subprog.	108.14
	- errata	109.11
Tutorial IV	HEADER Contd.	110.3
Tutorial V	HEADER Concl.	110.14
Tutorial VI	GETPUT subprog.	111.13
	- errata	113.19
Tutorial VII	SAVE, LOAD	114.8
		114.29

EXTENDED BASIC - see also CALL PEEK, CALL LOAD, Programs that write programs, SECURITY, Windows

Change Edit screen colours		208.35
Character set (small caps and true lower case) prog.		208.15
- errata		210.35
Keytest (returns ASCII) prog.		208.18
Relational Expressions		301.13
Screen Digital Clock prog.		208.21
User defined Interrupt (2 demo progs.)		208.36

F

Flippies (homemade - read also Disk Basics 114.32) 205.19

Fonts (alternative character sets) see CALL CHAR, GRAPHICS, EXTENDED BASIC

FORTH

Available on German module		108.8
Bugs:		
Corrections to prog. disks (screens 58 & 59)		114.36
Disk errors (screens 58 & 72)		210.28
list for screens and manual		206.29
PAB - ADDR (also under screen 72)		110.4
PLUS bug in T.I. manual		204.12
Text mode hex values (omitted from SCREEN comms.)		204.13
Configuring to your system I		210.26
Configuring to your system II		210.28
Disk Copier prog.		113.57
- errata		114.8
Disk Fixer comparison : Forth v. Navarone		202.14
Double sided copying warning		112.17
Flower Power (demo prog.)		211.13
Generating Sound		205.15
Introduction to Forth		209.5
Learning Forth I (Basics)		211.28
Learning Forth II (Commands)		211.30
Screen & Text colour changing		112.17

FUNNELWRITER see WORD PROCESSING

FREEWARE

Concept explained	203.10
List of	202.12

G

GAMES

Adult Adventure review	308.24
------------------------	--------

Adventure Help :

Adventureland	303.26
Claymorgue Castle	309.11
Cut Throats	308.21
Enchanter	302.12, 304.23, 308.21
Golden Voyage	304.23, 307.13
Hitch-Hiker's Guide	304.23, 305.23, 306.12, 307.15
Leviathon	308.21
Mission Impossible	303.26, 305.25
Mystery Fun House	303.27, 304.21
Planetfall	302.12, 304.24, 308.22
Pyramid of Doom	303.27
Return to Pirate's Isle	302.13
Sao Vera	308.21
Savage Island I	311.20
Savage Island II	303.27, 304.23, 312.9
Sorcerer	304.24
Strange Odyssey	303.26
The Count	303.25
Voodoo Castle	303.26
Witness	308.22
Zork I	302.11, 309.9
Zork II	302.12, 309.10
Zork III	302.12, 309.10

Big Texas Spy Adventure review	308.24
--------------------------------	--------

Games programs to type in :

Awari	301.6
Checkers	308.25
Dragon's Maze	305.11
English Tennis	307.7
Forest Fire	302.9
Meteor Attack	303.14
Othello	309.18
Rescue (XB)	306.8

"Highest Scores" routine for your games progs.	105.7
--	-------

GRAPHICS

Futurist (char. definitions for font)	104.10
Graphics Tutorial part I	110.9
Graphics Tutorial part II	112.15
Graphics with T. I. Writer	210.11
Manipulating Definition Strings - routines for upside down/inverse video/mirror image - errata	301.10 303.7
Redefining character sets see CALL CHAR	
Routine for checking User Defined Graphics	114.22
T.I. Artist V2.0 : package description	212.26

H

HOME SOFTWARE COLLECTION

Improvements for Chess prog. (DE 0004)	304.18
--	--------

I

INPUT - change tone	108.8
INPUT VALIDATION (mug trapping)	206.27
INT - use for rounding negative numbers	309.5

J

JOYSTICKS

CALL JOYST routine	106.9
CALL JOYST - 6 example routines - errata	108.21 109.3
Change code from Joyst 2 to Joyst 1 (MM)	112.13
Input via Joystick port	114.18

L

Languages - see ASSEMBLY, BASIC, C.99, ENHANCED BASIC,
FORTH, PASCAL

Load Interrupt Switch

Fitted to Speech Synthesizer - see D-I-Y

LOGO II

An introductory procedure	304.9
---------------------------	-------

M

Mailing List - see PRINTER

MAXIMEM (Canadian improvement on Gram Cracker)	208.39
--	--------

N

Navarone Bug Fixer (bug in) - see DEBUGGING

Numeric Keypad - see D-I-Y

O

Ones's & Two's Complement 102.13
 - errata 103.9

OR (example of use in XB) 108.10

P

PASCAL - fault in Disk formatting utility 114.5

PEB (Peripheral Expansion Box)

Disassembly procedure 113.47
 Interface card design weakness 208.7
 Power supply boost by changing voltage regulator 113.47
 - very important warning info. 203.4, 204.5
 Reducing fan noise 206.7
 - more (important) info. 207.8
 RS232 card connections and mods see PRINTER; RS232
 Use of high - memory for 24K data storage 205.25

Poetry & Prose - computer generated 309.13

POS - use for menu selection 104.10

PRESTEL see MODEM COMMS

PRINT AT - mimicking in TIB 211.18

PRINTER

"Doodle" demo prog. 113.38
 Epson FX80 - simultaneous number/£ sign 201.9
 Interfacing with the TI99 204.23
 Mailing List - routine for PIO or AXIOM codes 203.14
 Microline 82A
 slashed zero 114.35
 unscrambling number/£ sign codes 114.35
 PIO fix for "true Centronics" output 113.10
 RS232 cable connection 103.9
 RS232 card to printer connections 308.13

Program Storage Formats 210.31

Programs that write programs (XB)

part I 310.14
 part II 402.16

PSEUDOCODE - the secret of programming	207.9
- an alternative definition	301.24
Pseudorandom Numbers - generating uniquely	312.28

Q

QUIRKS - see also BASIC (Sprites with TIB), Test Mode Unclassifiable oddities, trivia and things to mess around with see 104.11, 104.18, 105.15, 106.5 (TE 2), 106.16, 109.14, 114.8 (XB).	
---	--

R

RAMDISK (New Horizons)	
Facts	212.18
- more info	303.6
Parts list	212.17
RAM Expansion - see also EDITOR/ASSEMBLER, MYARC Matchbox 32K RAM expansion see D-I-Y projects	
Relational Expressions - see also EXTENDED BASIC Use with CALL KEY	101.12
RS232 - see also PRINTER	
PIO/Centronics (incl. Assembly routine for Busy signal)	301.20
Replacement ROM for "true centronics" interface	208.7
- obtain from ITUG	302.22
RS 232 to parallel conversion circuit	211.25
Source code booklet available	212.11

S

SAVE - see CASSETTE STORAGE

SECURITY

Embedded Code method 1 - CHR\$ (0)	103.12
Embedded Code method 2 - Program Peeker (EA/MM with mods for XB)	113.54
- errata for XB	114.7
Entrance Codes demo prog.	204.15
Password prog. (hidden screen display)	108.11
XB unprotection addresses	113.46, 114.9
- errata	201.8

SEG § - see Concatenation	
SERVICE - see CONSOLE	
SINE WAVE plotting explanation	208.27
SORTING & SEARCHING	
Tutorial I Sorting Overview	203.15
" II Bubble Sorting	204.20
" III Tags	205.20
" IV Input/Printout prog. for Bubble	206.26
" V Tag Sort	210.15
" VI Tag Sort concl.	211.32
SOUND - see also CALL SOUND	
Music progs. : Funeral March	108.10
Baa Baa Black Sheep	108.10
Speech Recognition - see MBX	
SPEECH SYNTHESIZER	
Available scientific papers	108.5
- errata	109.3
Detailed analysis part I	104.19
part II	105.23
- errata	106.4
part III	106.19
Separator characters	101.12
Sprites - see CALL SPRITE, GRAPHICS	
String Variables - valid use to compare hex numbers	112.13
Super Bugger - see DEBUGGING	

T

Test Mode	
TI Invaders (Shift 838)	205.27
Munchman (Shift 835)	205.27
Texas Instruments U.K. addresses	301.31
T.I. Writer - see WORD PROCESSING	
TK Writer - see WORD PROCESSING	
Tokens :	
explanation	101.10
tabled by ASCII number	102.(i)
tabled by reserved word/symbol	102.(iii)

WHAT PEOPLE HAVE SAID ABOUT THIS INDEX

"This index is something I can really get my teeth into!"

Billy Lugholes



"I was enjoying the index until this chopped liver fell on my head..."

Rob O'Tech



"Great index! Where is the section on turtle soup?"

Terry Pinn (Ninja Turtles Association)



"Now that DOES compute"

Dr Spock



WORD PROCESSOR - TI BASIC PLUS MINIMEM (REF. 6687)

OPERATING INSTRUCTIONS

The RUN command assumes that an RS232 interface for printer is connected and switched on. (LINE 90) If you are preparing text for filing on cassette or just practising, you should command RUN 100.

Text can be entered in blocks of up to four screen lines and each block must start and finish with speech marks ". (This is the only way that spaces at the start or finish of an entry can be retained.) Any speech marks forming part of the text should be typed in twice. If a word is broken it can be continued in the next block, as the blocks are joined together in the final text.

If you make "@" your first entry, a letter heading will be printed automatically and a prompt given for entering the date. (LINE 2690)

If you wish an entry to begin on a new line, use "~" as the first character. (This will be deleted from the final print-out.) Any entry which starts with three or more spaces, e.g. titles and main headings, will automatically be placed on a new line. To leave a blank line, just enter a blank.

If text is on cassette, enter "#" and follow on-screen instructions. The cassette will normally run until the end of the text, but the input can be terminated prematurely by pressing any key. A useful time-saver is to remove the motor control plug from the cassette player as soon as the first playback is under way. This will avoid the time-wasting breaks between entries.

When all text has been entered, initiate the formatting process by entering "*". (This will happen automatically when all the available memory is full.) The entered blocks of text will then be concatenated and divided into printer lines of up to 70 characters, divisions being made only between complete words. If the "justification" option has been chosen, each line will then be padded out to 70 characters by adding extra spaces, these being positioned in pseudo-random manner to minimise visibility.

When the caption "READY FOR CHECKING" appears, the formatted text may be inspected and edited on a line-by-line basis by means of keys 1 to 0, which have the functions listed below. (These functions may conveniently be pencilled in on the white label strip above the keys.) Note that the CTRL and FCTN keys must NOT be used during editing. To CLEAR and begin a new text entry, use FCTN4 in normal manner.

KEY 4 : CHECK. Allows the formatted text to be inspected line-by-line in forward sequence. The display takes place without breaking words, so one printer line will occupy about three screen lines. Symbol "~" on screen denotes the start of a printer line.

KEY 9 : BACK. As 4, but in reverse sequence.

KEY 7 : RETURN. Returns the inspection point to the beginning of the text. This is useful, for example, when you have inspected through to the end and wish to begin printing.

KEY 8 : REDO. Accepts as input an altered or corrected version of the line currently on display.

KEY 0 : CORRECTION. This is for rectifying misprints in the displayed text. The printer line will be redisplayed in "direct" form so that it occupies exactly two and a half screen lines of 28 characters each. This makes it possible to estimate the serial number of the misprinted character. As instructed on-screen, enter the location of the faulty character followed by the correction, e.g. 43f, whereupon the line of text will be reprinted using the corrected character. For a deletion, enter e.g. -43, which will remove the 43rd character and move the rest of the line along to fill the gap. To add a character, enter e.g. +43f, which will cause character f to be inserted at position 43. If margin justification has been used, however, the line will have been padded to its maximum permissible length, so it will be necessary to delete one of the added spaces before the extra character will be accepted. Note that if the corrected or added character is a space or comma, the "location+character" entry must be enclosed between speech marks. When a line has been fully corrected, enter a blank to return to the normal checking routine.

KEY 2 : INSERT. Accepts as input a new line of text and inserts it directly after the line currently on display.

KEY 1 : DELETE. This must be followed by a number key from 1 to 9, whereupon the line on display will be deleted together with following lines up to the total specified by the number key. Thus the maximum deletion at one time is 9 lines. The deleted text is not lost but is stored until the DELETE key is next operated, thus providing a TRANSFER facility.

KEY 5 : TRANSFER. Reinserts the last block of deleted text immediately after the line currently on display. This reinsertion can take place anywhere in the complete text, but only until the DELETE facility is next used. The TRANSFER key can be used as often as required, to give a REPEAT facility.

KEY 6 : PRINT. Feeds the formatted text out to a printer via the RS232 interface, starting with the line currently on display. Printing will normally continue to the end of text, but may be interrupted by pressing any key.

KEY 3 : FILE. Files the formatted text on cassette for future use, starting with the line currently on display. Each line appears on the screen immediately before it is filed and the filing process can be terminated by pressing any key.

EMERGENCY RESCUE FACILITY

It has not been possible to anticipate every possible misoperation during the editing process, so if the wrong key is pressed a lockout may occur. If this should happen, then switch off the computer and reload the program but DO NOT RUN, as this will destroy the entered blocks of text which are stored in Minimem. Enter EDIT 120 and delete the REM information leaving only the GOTO, then do the same with EDIT 170. The program can now be safely run, text being drawn from MINIMEM and reformatted. Any corrections made before the lockout will have been lost however, and must be remade.

--[<0>]--

