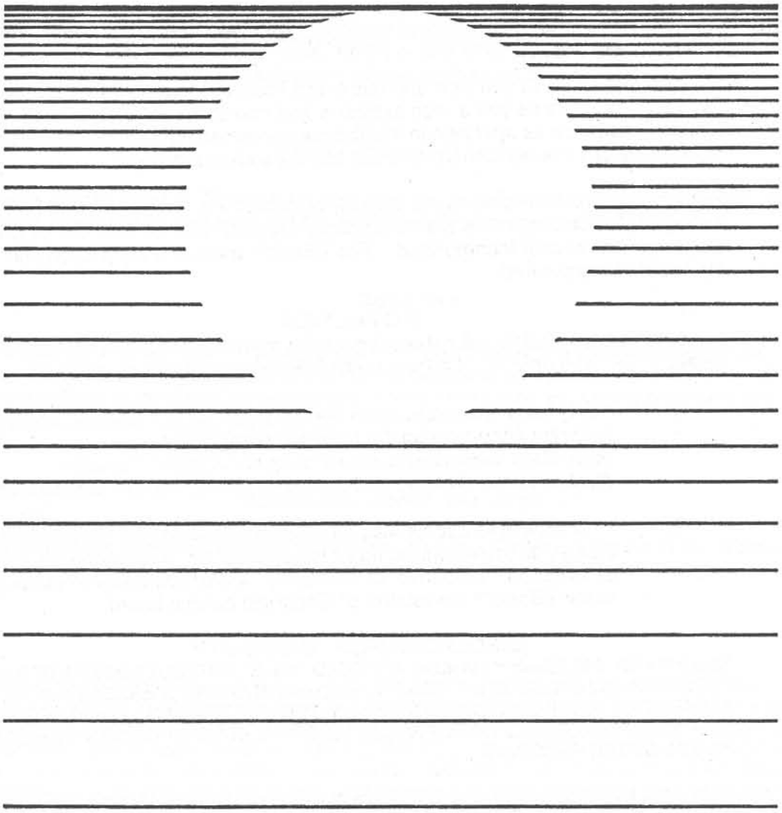
ZBasic™

Interactive BASIC Compiler



TRS-80™ Model I, III and Compatibles APPENDIX

Minimum 48K, 1 Disk system
Uses TRSDOS, LDOS, MULTIDOS, NEWDOS and DOSPLUS

Supports the Radio Shack and Micro-Labs High-Res Graphics Boards!!

* MODEL 1 FORMAT DISK USE CONVERT ON MODEL III *

USER'S COPY OF THE SIGNED LICENSE AGREEMENT

ZBasic™ License Agreement

On the condition that you sign and return this license agreement to Zedcor Incorporated, Zedcor hereby grants you a non exclusive and non transferable license to use this copy of ZBasic™ software as specified in this license agreement on one computer only. Failure to sign and return this agreement and still use the software is illegal.

I agree that ZBasic™ remains the property of Zedcor Inc. and accept that giving away or selling copies of ZBasic™ is theft of Zedcor's property and for this I will be prosecuted by the lawyers of Zedcor Incorporated. The ZBasic™ manual is my property but is copyrighted by Zedcor Incorporated.

ROYALTIES

I understand that Zedcor will not require royalty payments for programs I create with ZBasic™ *IF*, *and only IF*, I adhere to the following terms and conditions:

1. I will post prominently in the start of my software and in the front of my software documention the following paragraph :
   Parts of this software were created using the ZBasic™ Compiler.
   Portions of this Code are © Copyrighted, 1985 by Zedcor Incorporated.

2. I will submit to Zedcor Inc., an accurate description of my software along with the machines that it will operate on, so that Zedcor may, *at its option*, announce its availability and/or publish its availability in the ZBasic™ Newsletter or Electronic bulletin board.

DISCLAIMER OF WARRANTY

ZBasic™ AND THE ZBasic™ MANUAL ARE SOLD "AS-IS" WITHOUT WARRANTIES AS TO PERFORMANCE OR MERCHANTABILITY. THE SELLER'S SALESPERSONS MAY HAVE MADE STATEMENTS ABOUT THIS SOFTWARE. ANY SUCH STATEMENTS DO NOT CONSTITUTE WARRANTIES AND SHALL NOT BE RELIED ON BY THE BUYER IN DECIDING WHETHER TO PURCHASE THE PROGRAM.

ZBasic™ IS SOLD WITHOUT ANY EXPRESS WARRANTIES WHATSOEVER. BECAUSE OF THE EXTREME DIVERSITY OF CONDITIONS AND HARDWARE UNDER WHICH ZBasic™ MAY BE USED, NO WARRANTY OF FITNESS FOR A PARTICULAR PURPOSE IS OFFERED. THE USER IS ADVISED TO TEST ZBasic™ THOROUGHLY BEFORE RELYING ON IT. THE USER MUST ASSUME THE ENTIRE RISK OF USING ZBasic™. ANY LIABILITY OF SELLER OR ZEDCOR INC. WILL BE LIMITED EXCLUSIVELY TO PRODUCT REPLACEMENT OR REFUND OF THE PURCHASE PRICE.

TERMINATION OF LICENSE

Any breach of one or more of the provisions of this agreement by me shall constitute an immediate termination of this agreement. Nevertheless, I agree that in the event of such a termination, all provisions of the agreement which protect the rights of Zedcor Incorporated shall remain in force.

ZBasic™ © Copyright Notice

Single Person License

The price paid for one copy of ZBasic™ licenses you to use the product for your personal use only and only when you have signed and returned the enclosed licensing agreement.

Royalties

Programs you create with ZBasic™ (Source code or Object code) may be sold without paying royalties to Zedcor Inc. IF and ONLY IF you fill out and return the licensing agreement in the front of this manual and adhere to the notices in the licensing agreement. Persons creating or selling programs with illegal or pirated copies of ZBasic™ will be prosecuted.

Comments about the Product

If you have comments about this manual or software, fill in the Comment Report in this manual in front of the Reference Section.

Trademarks

Because ZBasic™ runs on so many different products, this manual makes reference to many manufacturers and products that are trademarks of their respective owners. We acknowledge these companies and thank those that have been so helpful in making ZBasic™ a success: ZBasic™ is a trademark of Zedcor Inc.. Apple and Macintosh™ are registered trademarks of Apple Computer. IBM PC, IBM PC jr., IBM PC-XT, PC-AT and PC-DOS are registered trademarks of International Business Machines Corporation. MBASIC™, MS™, Xenix ™ and MS-DOS™ are registered trademarks of Microsoft™ Corporation. CP/M is a registered trademark of Digital Research. True Basic is a registered trademark of Addison Wesley. Turbo Pascal is a registered trademark of Borland International. TRS-80, Radio Shack, Tandy 2000, Tandy and Tandy 1000 are registered trademarks of Tandy Corporation. Kaypro II, 4, 10, 16 and 286i are registered trademarks of Kaypro Corporation. Commodore 64 is a trademark of Commodore International. Franklin Ace 1000 is a trademark of Franklin Corporation. Osborne is a trademark of Osborne Corporation. Compaq and Deskpro are trademarks of Compaq Computers. Panasonic Senior and Executive Partners are trademarks of Panasonic Industrial Corporation. Data General One is a trademark of Data General Corporation. Quadram is a trademark of Quadram Corporation. VAX is a trademark of Digital Equipment Corporation. Unix is a trademark of AT&T Corporation. We apologize for any unintentional omissions.

Zedcor Incorporated reserves the right to make changes to the specifications of ZBasic™ and to the ZBasic™ documention without obligation to notify any persons or organization of such changes.

ZBasic™

Interactive BASIC Compiler
Version 3.0

TRS-80™ Model I and III
and Compatibles Version
ZBasic APPENDIX

Appendix
and
TRS-80™ versions
and enhancements

by

A. Gariepy

Table of Contents

ZBasic TRS-80™ Model I, III APPENDIX

Notes on this version of ZBasic 3.0

MEMORY ZBasic 3.0 for the TRS-80 models 1 & 3 is designed to run with both the TRS-80 model 1 under TRSDOS 2.3 or compatibles, and the TRS-80 model 3 TRSDOS v1.3 or Compatible DOS. The typical programming area available in a 48k machine is aproximately 21.5k with the resulting code and variable area being up to 38k. This version also has the capability of doing OVERLAYS which are explained later in this Appendix.

FILENAMES The filenames in ZBasic are the standard TRSDOS filenames specified in the TRSDOS manual. Some examples are as follows: `ZBASIC/CMD,  PROGRAM/BAS:1,  MYPROG/BAS:Ø.SECRET`

RENAME Does not function with this version.

SOUND Will use the cassette port to do sound if available. Connect a Radio Shack apmplifier to the cassette port for sound output.

MODE 8 to 15 Will use the Radio Shack **640x240 high resolution graphics board** if available (Micro-Labs also).

MOUSE Does not function with this version.

KEYS

Up arrow	--->	List previous line
Down arrow	--->	List next line
Left arrow	--->	List first line
Right arrow	--->	List last line
/	--->	List next 10 lines
<SHIFT> up arrow	--->	**ESC**ape

HELP FILE The file used in the ZBasic HELP command is called "ZBASIC/HLP". This file can be deleted to allow more room on the disk, in which case typing HELP from the editor will generate a "FILE NOT FOUND" error.

ZBasic TRS-80™ Model I, III APPEND

Executing ZBasic from TRSDOS

* MODEL I FORMAT DISK USE CONVERT ON MODEL III *

There are basically two ways of starting ZBasic from the operating system:

1. **Type:** `ZBASIC <ENTER>`

   This is the standard way to startup ZBasic. See "GETTING STARTED" in the ZBasic standard manual. Also see "TRSDOS CONFIGURATION" in this appendix.

2. **Type:** `ZWARM <ENTER>`

   This is a special way to startup ZBasic to recover a ZBasic text program after a crash or reset (To restart a program like BASIC*). A version of ZBasic must be created using the <W>arm start creator option from the ZBasic start up screen.

   To Create this WARM start version configure ZBasic for your machine and save using the <S>ave option from the ZBasic startup menu. (DO NOT use your MASTER DISK only use a BACKUP COPY of your master diskette).

   Exit ZBasic using QUIT and re-enter the just created configured ZBasic and use the <W>arm Start Creator option to create a WARM start version of ZBasic to be called **ZWARM/CMD** which can be used to recover ZBasic program text after a RESET or program lock-up.

   The ZWARM version on the disk can only recover the program if it is still intact in memory and has not been overwritten by itself or another program. This cannot recover from a program NEW as it destroys the program in memory.

   The ZBasic 3.0 patch option allows the user to PATCH specific addresses in ZBasic to Change areas in the JUMP TABLE for special hardware or software and to apply fixes to the actual program as specified by Zedcor to provide some special features. These changes may be save by using the <S>ave option from the start-up MENU.

TRSDOS CONFIGURATION

After typing "C" in the initial prompt screen, ZBasic will ask for the standard configuration parameters explained in the "GETTING STARTED" section of the manual. Following these standard parameters are the TRSDOS specific configuration parameters. The additional prompts displayed are as follows:

Default Clear nnnnn Size 03E8 01000 ?

This Configuration selects the default amount of memory to be set aside for strings in the INDEX$ area at compile time. The actual amount of memory in the running compiled program can be found by using MEM function in the program. If this area becomes less than or equal to zero due to high memory drivers a 'Not Enough Memory' error will be displayed and the program will exit back to DOS.

List First Line <Key> 0019 00025

This key when pressed as the first key on a line will cause the ZBasic editor to LIST the first program line in memory and make it the current line. (Defaults to the <SHIFT> Left arrow Key).

List Last Line <Key> 012C 00300

This key when pressed as the first key on a line will cause the ZBasic editor to LIST the last program line in memory and make it the current line. (Defaults to the Right arrow Key).

List Previous Line <Key> 012C 00300

This key when pressed as the first key on a line will cause the ZBasic editor to LIST the previous program line in memory and make it the current line. (Defaults to the Up Arrow Key).

List Next Line <Key> 012C 00300

This key when pressed as the first key on a line will cause the ZBasic editor to List the next program line in memory and make it the current line. (Defaults to the Down Arrow Key).

Configuration: (cont.)

Find Next occurrence <Key> 003B 00059

This key when pressed as the first key on a line will cause the ZBasic editor to Find the next occurrence of the string last used in FIND in memory a make it the current line. (Defaults to the ';' key (semi-colon)).

TRSDOS or Other type <T/O> 012C 00300

T/N/O

This special Configuration is needed to tell the ZBasic EDITOR which type of DOS you are using so the DIR command will be available from the editor. Type a "T" if you are using TRSDOS. Type an "O" if using any other DOS. If it is not configured correctly, a system crash may occur when **DIR** is used from the editor. This is one of those things in machine language that was never truly standardized by TRSDOS and OTHERS.

DIR does not function from the ZBasic editor with Model 1 TRSDOS.
Most other Model 1 DOSes like LDOS, MultiDOS etc., will.

N FOR NewDos 80 2.0

Overlay Offset (See Appendix): 00000 00000 ?

Allows you to set the Offset for overlay programs. See OVERLAYS.

MEMORY CONSIDERATIONS

The TRSDOS version of ZBasic has three different modes of operation concerning memory organization -- EDIT mode, RUN mode, and RUN* mode (see memory map on following page). At least 32k of memory is required for the EDIT and RUN modes (the development stage of the program). However, after a program has been compiled and saved using RUN*, it can be run on as little as 16k of memory depending on the size of the program (the RUN* mode shown on the memory map).

NOTES ON THE ZBasic™ MEMORY MAP:

1. The system top of memory is obeyed by ZBasic 3.0 in both the editor and object code files. The CLEAR area in the /CMD file created by ZBasic is the only area of the compiled program which can adjust to different sizes of high memory drivers or machine language routines. If this area is too small when an attempt is made to execute this program from DOS a "Not Enough Memory" error will occur and it will return to DOS.

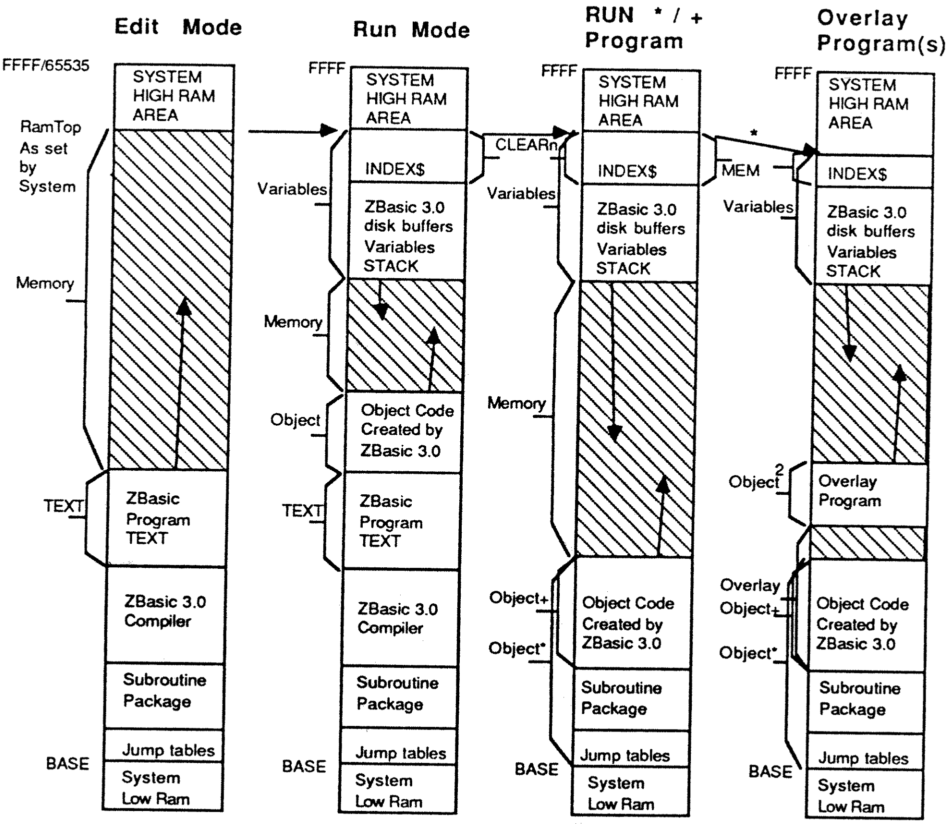
2. The ZBasic subroutines and jump tables are not saved to disk when a program is compiled as a chain file using RUN+. Thus, chain files take up 10k less on disk.

3. ZBasic is located immediately after the DOS. There may be drivers or other applications installed at the top of memory. ZBasic does, however, assume to own all of the memory from DOS to the TOP of memory as defined by the DOS.

4. The size of the INDEX$ is determined by the **CLEAR** statement (see reference section in main manual). The TRSDOS version of ZBasic defaults to **CLEAR 1000**, making the INDEX$ equal to 1k. If there is not enough memory to create a CLEAR of the size specified, then the largest size available will be allocated. The size of the INDEX$ memory can be determined using the MEM function within the running program.

5. When the **CALL *string*** statement is used to execute a DOS function (see this appendix under "CALL statement"), the DOS function must not use the area where ZBasic resides else a system crash may occur (This DOS function jump vector is located in the ZBasic jump table so it may be re-vectored for different DOS'es).

MEMORY MAP



BASE ADDRESSES

CPM80 2.2 or 3.0	0100 HEX OR 256 DECIMAL
TRS80 Model 1 or 3	5200 HEX OR 21976 DECIMAL
TRS80 Model 4 ver.6.2	3000 HEX OR 12288 DECIMAL

*TopRam is the highest RAM address the system will allow ZBasic to use. This address varies from system to system and even on the same system!. The INDEX$ CLEAR area is the only area of ZBasic that can adjust to this area if not enough room is allocated. When the object code is executed, a "Not Enough Memory" error will occur and the object code will stop and return to the operating system. MEM will return memory for INDEX$.

▨ Memory available for ZBasic and ZBasic Object Code.

ZBasic TRS-80™ Model I, III APPENDIX

RS-232 COMMUNICATION

ZBasic for the TRS-80 Model I and III supports asynchronous communication using the filenumber #-1 for the TRS-80 standard serial Interface. Baud rate, parity, stop bits, and word length are all controlled in the OPEN "C" statement (see "REFERENCE SECTION" in main manual).

The serial interface should work properly on most TRS-80 compatibles.

RS-232 PROBLEMS

If the asynchronous communication is not working, try any of the following:

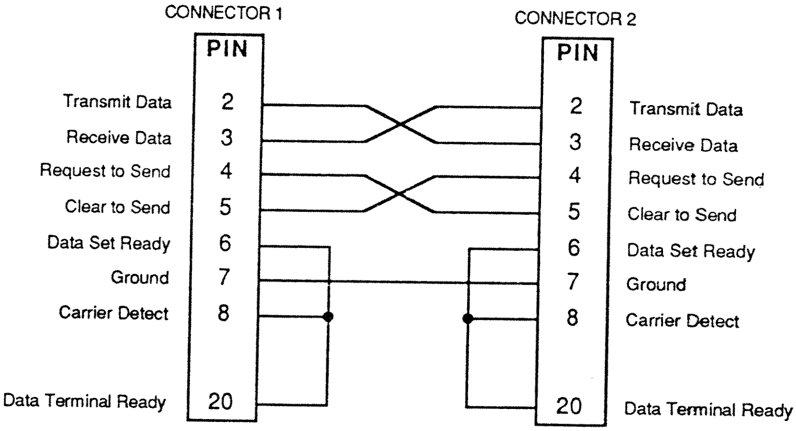
1. Check to make sure the baud rate, parity, stop bits, and word length settings are the same on both sides of the communication.

2. Check for proper cable wiring. The cable must support the standard RS-232 asynchronous interface. If the serial transfer works at a low baud rate (like 300 baud) but fails at higher baud rates, the cable is wired improperly.

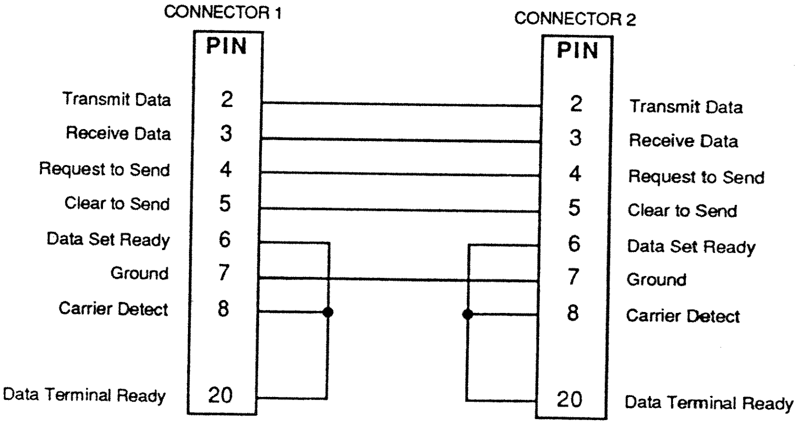
   The diagrams on the next page show the two most typical cable configurations. The top diagram is for communication between two DTE's (Data Terminal Equipment) or two DCE's (Data Communication Equipment). The diagram below it is for communication between a DTE and a DCE. These cable configurations are not the "rule", they are only the most typical for proper RS-232 interfacing.

RS-232 : (cont.)

Communication devices: DTE <----> DTE
DCE <----> DCE



Communication devices: DTE <----> DCE
DCE <----> DTE



ZBasic JUMP TABLE

The TRSDOS version of ZBasic makes available a jump table starting at address 5200 hex. These jumps can be altered to jump to some other routine to handle the same function. This can be useful for implementing special hardware or for handling non-compatible DOS or HARDWARE. Also included in this jump table are the USR function jumps, many of which are predifined. The following is a list of all the available jump locations with a short description of each:

XX=01 for CPM80 2.2 & 3.0
XX=30 for TRS-80 Model 4 TRSDOS/LDOS 6.2
XX=52 for TRS-80 Model 1 and 3 versions

Address	Vectors	Description	Special Notes (*Save registers!!)
XX00	JP SUBRTN	Cold Start entry point	NONE
XX03	JP WARM	Warm Start entry point	NONE
XX06	JP EXIT	Exit to System	NONE
XX09	JP CHRINP	Get Keyboard Character	* RETURNS KEY IN A
XX0C	JP VIDOUT	Output Character to display	* SENDS CHAR IN A
XX0F	JP LPROUT	Output character to printer	* SENDS CHAR IN A
XX12	JP SCANKY	Get Key from kybrd no waiting	* A=0 NO KEY ELSE A=KEY
XX15	JP ICOMM	INITIALIZE RS232 PORT	Set Bd,Par,Stop bits,Wrd len UART
XX18	JP BAUD	SET RS232 BAUD RATE	HL=BAUD RATE 300,1200..
XX1B	JP PARITY	SET RS232 PARITY	HL=PARITY 0,1,2...
XX1E	JP STOPBT	SET RS232 STOP BITS	HL=STOP BITS 0,1,2...
XX21	JP WORDLN	SET RS232 WORD LENGTH	HL=WORD LEN 5,6,7,8...
XX24	JP RSREAD	READ 1 CHAR FROM RS232	*On exit A=char from rs-232 set Z flag
XX27	JP RSWRT	WRITE 1 CHAR TO RS232	*A=CHAR TO SEND set Z flag on exit
XX2A	JP RSSCAN	GET RS232 CHR NO WAIT	* A=0 NO Char,Z flag set else A=CHAR
XX2D	JP TIME	GET TIME STRING	ON Exit HL >=> 9 bytes: len,H,H,:,M,M,:,S,S
XX30	JP DATE	GET DATE STRING	ON Exit HL >=> 9 bytes: len,M,M,/,D,D,/,Y,Y
XX33	JP PRINTA	PRINT @(X,Y) ROUTINE	HL=Y,DE=X SET CURSOR for next character
XX36	JP PRINTG	PRINT %(X,Y) GRAPHIC	Loc HL=Y,DE=X set cursor for next character
XX39	JP SOUND	GENERATE SOUND	HL=DURATION MS,DE=FREQ in HZ
XX3C	JP MOUSE	READ MOUSE STATUS	HL=TYPE RETURNS IN HL
XX3F	JP CONVXY	CONVERT GRAPHIC POS	HL=Y,DE=X RETURN IN HL,DE
XX42	JP MODE	SET GRAPHICS MODE	HL=MODE #
XX45	JP COLOR	SET COLOR FOR GRAPHICS	HL=COLOR
XX48	JP SETXY	SET POINT DE,HL	LOCAL COORDINATES
XX4B	JP PLOTXY	PLOT FROM LAST POINT TO	DE,HL LOCAL
XX4E	JP FILLXY	FILL FROM POINT	DE,HL FILL AREA around X,Y
XX51	JP POINT	READ COLOR AT POINT	DE,HL RETURNS HL=COLOR
XX54	JP DOSCALL	CALL DOS COMMAND	HL >==> STRING

ZBasic TRS-80™ Model I, III APPENDIX

JUMP TABLE: (cont.)

Default USR functions table

XX=01 for CPM80 2.2 & 3.0
XX=30 for TRS-80 Model 4 TRSDOS/LDOS 6.2
XX=52 for TRS-80 Model 1 and 3 versions

USR digit(*expression*)

On Entry:	expression >==> Z80 reg. HL
On Exit:	HL reg. returned in the *expression* which contained the USR function.

XX57	JP	USR0	OLD HL RETURNS	HL ON COLD START ENTRY
XX5A	JP	USR1	RETURN	
XX5D	JP	USR2	RETURN	
XX60	JP	USR3	RETURN	
XX63	JP	USR4	RETURN	
XX66	JP	USR5	RETURN	
XX69	JP	USR6	RETURNS LAST LINE # EXECUTED WITH TRON ACTIVE	
XX6C	JP	USR7	RETURNS RAW RANDOM # 0 TO 65535	
XX6F	JP	USR8	RETURNS SIN OF HL IN BRADS AS A VALUE OF +/- 256	
XX72	JP	USR9	RETURNS COS OF HL IN BRADS AS A VALUE OF +/- 256	

Special strings and constants

These Strings and constants may be changed by POKES or using the PATCH function from the MENU.

XX80	CLSSTR	4 BYTE CLEAR SCREEN STRING
XX84	PRNTAM	6 BYTE PRINT AT CONTROL STRING <ESC>,<=>,Y+32,X+32,0,0 (TRS-80 versions 1st 2 bytes are RS232 config bytes)
XX8A	CUROFF	4 BYTE CURSOR OFF STRING
XX8E	CURON	4 BYTE CURSOR ON STRING
XX92	CLRLINE	4 BYTE CLEAR TO END OF LINE STRING
XX96	CLRPAGE	4 BYTE CLEAR TO END OF PAGE STRING
XX9A	PAGE0	1 BYTE PRINTED LINES PER PAGE (0=disabled)
XX9B	PAGE1	1 BYTE TOP MARGIN (0=none)
XX9C	PAGE2	1 BYTE ACTUAL PAGE LENGTH IN LINES (0=disabled)
XX9D	PAGE3	1 BYTE PRESENT LINE (line#1=0,line#2=1....)

Other important Addresses

XXA0 TO XXBF	------> User area for PATCHES (Saved with <s> config option)
XX00 +200H	------> 256 buffer (O.K. to use whole buffer during machine language routine)

OVERLAYS

ZBasic 3.0 on Z80 based computers allows for Overlays to be used to make the most efficient use of a systems available memory.

An Overlay is a program which is loaded from disk (without destroying the program in memory) and executed. After it is executed, it will RETURN to the main program. As long as it is in memory it may be called over and over again by RUN 0(zero) until it is overwritten by another overlay or program.

The main advantage of Overlays is they are small and will normally load up quickly. After they have been loaded, they work like a GOSUB with the variables being chained that appear in the DIM statements at the start of the main program and the overlay program.

Here are the steps in creating an overlay program.

#1. Create the MAIN program and define the Overlay subroutine(s).
#2. To determine th e *OFFSET* for the overlay subroutine:

A Type **RUN+** from the MAIN program
(type **<ENTER>** when it asks for a filename)

B. Type: **MEM**

```
00000   Text
00000   Memory    <--- This is the available room for the overlay.
00000   Object    <--- This PLUS 100= OFFSET amount.
00000   Variable
```

The number after "Object" +100 is what is used in the <C>onfigure startup to create the OFFSET for the overlay program.

#3 Set up all variables which will be used by the Overlay program in identical DIM statements at the start of both the MAIN program and OVERLAY subroutines.
#4 Compile and save the **MAIN** program using the **RUN*** command.
#5 Compile and save the **OVERLAY** subroutine using **RUN+.**
#6 When the MAIN program requires the use of the overlay use: `OPEN"I",1,"overlay filename"` and use `RUN 1` to execute the overlay subroutine the 1st time.

#7 After the overlay is loaded it may be executed again *without* reloading the OVERLAY by using the RUN 0 (zero) statement.
See the OVERLAY program examples on the next page...

OVERLAY Example Program

MAIN Program

```
ZBASIC<cr>
(in  configure)
OVERLAY  OFFSET  =  0
E
ZBasic Ready

00010 CLEAR 5000 : DIM E,X,Y,Z,T$
00020 PRINT "STARTING MAIN PGM"
00030 OPEN"I",1,"PGMOVL"
00040 PRINT "GOING TO OVERLAY"
00050 E=0  :  RUN 1
00060 PRINT "BACK FROM OVERLAY"
00070 E=1  :  RUN 0
00080 E=2  :  Y=1  : Z=8  :  RUN 0
00090 E=3  :  RUN 0
00100 PRINT "T$='";T$;"'"
00110  STOP



RUN+
<cr>   don't Save Object!!!
MEM
00217 Text
----- Memory
00200 Object
06000 Variable

RUN*            (TRSDOS)     (CPM80)
Object File.. MAIN/CMD or MAIN.COM

MEM
00217 Text
----- Memory
11000 Object   >==> (size on disk)
06000 Variable

QUIT<cr>

(Compile overlay program)
```

OVERLAY Program

```
ZBASIC<cr>
(in configure All Else Same!)
OVERLAY OFFSET = 200+256 (aprox)
E               = 456
ZBasic Ready

00010 CLEAR 5000 : DIM E,X,Y,Z,T$
00020 IF E=0 THEN PRINT "OVERLAY"
00030 ON E GOTO "HELLO","TEST",120
00040 PRINT "*BAD COMMAND*"
00050 RETURN
00060 "HELLO"
00070 PRINT "HELLO" : RETURN
00080 "TEST"
00090 FOR X=Y TO Z
00100     PRINT X,
00110 NEXT X : PRINT : RETURN
00120 T$=STRING$(20,"X")
00130 RETURN

RUN+
Object Filespec.... PGMOVL<CR>

MEM
00208 Text
----- MEMORY
00150 Object >==> (size on disk)
06000 Variable

QUIT<cr>

MAIN
STARTING MAIN PGM
GOING TO OVERLAY
IN OVERLAY
*BAD COMMAND*
BACK FROM OVERLAY
HELLO
1     2     3     4     5     6     7     8
T$='XXXXXXXXXXXXXXXXXXXX'
Break in 00110
```

ZBasic TRS-80™ Model I, III APPENDIX

MODE

ZBasic allows you to utilize different screen modes. The standard mode is 64 x 16. The standard graphics are 128 x 48. If you have the Radio Shack High resolution graphics board installed in your computer you may use modes 8 and higher.

character = no graphics, uses text instead.
32 x 16 = expanded characters
128 x 48 = Regular low resolution graphics
640 x 240 = Radio Shack High Resolution Board
or compatible. (Micro-Labs Hi-Res Board.)

TRS-80 Model I and III MODE CHART

MODE	TEXT	GRAPHICS
0	32 x 16	character
1	32 ~~64~~ x 16	64 ~~128~~ x 48
2	32 x 16	character
3	32 ~~64~~ x 16	64 ~~128~~ x 48
4	64 ~~32~~ x 16	character
5	64 x 16	128 x 48
6	64 ~~32~~ x 16	character
7	64 ~~64~~ x 16	128 x 48
8	32 x 16	640 x 240
9	32 ~~64~~ x 16	640 x 240
10	32 x 16	640 x 240
11	32 ~~64~~ x 16	640 x 240
12	64 ~~32~~ x 16	640 x 240
13	64 x 16	640 x 240
14	64 ~~32~~ x 16	640 x 240
15	64 x 16	640 x 240

ADDENDUM

We inadvertantly forgot to include these predefined user routines under USR in the manual (the integer SIN and COS are especially quick). Please insert this addendum in the manual at page R-157

ZBasic Predefined USR functions

All versions

USR function — **Description**

USR6(*expr*)

Returns the last line number executed that used any of the TRON functions (*expr* is not used).

```
10 TRONX
20 I=USR6(0)
30 PRINT I
```

USR7(*expr*)

Returns ZBasic's random number seed used in the RND function (*expr* is not used).

```
10 FOR I=1 TO 10
20    PRINT USR7(0)
30 NEXT I
```

USR8(*angle*)

Returns the integer sine of *angle* in the range +/- 255 (corresponding to +/- 1). The angle must be in brads.

```
10 MODE7 : CLS
20 FOR I=0 TO 255
30    PLOT I<<2,-USR8(I)+384
40 NEXT I
```

USR9(*angle*)

Returns the integer cosine of *angle* in the range +/- 255 (corresponding to +/- 1). The angle must be in brads.

```
10 MODE7 : CLS
20 FOR I=0 TO 255
30    PLOT I<<2,-USR9(I)+384
40 NEXT I
```