



Mini Memory Relocator

RAMpaging Through Memory

A Program By Martin Kroll, Jr.

218 Kaplan Ave.
Pittsburgh, PA 15227

BASIC

If you purchased your Mini Memory cartridge thinking you would have 4K of RAM in which to write Assembly Language programs, you may find yourself a bit disappointed. In order to write in Assembly Language, you need an *assembler*, a program which takes your assembly statements and converts them into the machine code the computer can understand. The Line-by-Line Assembler that comes with the Mini Memory only lets you assemble one line at a time, until you have written your program. And even this simple assembler requires Mini Memory to run in, so you won't have all the RAM to use for your program. There is, however, a remedy for this limited memory problem, which we present in the form of the *Mini Memory Relocator*.

This program will let you move any program in Mini Memory from one location to another. You may ask "What good will that do, when the amount of memory space is still small?" First, it will let you make use of the space taken by the assembler, once the program has been written in. It will also allow you to write several short programs, or subroutines, move them to predetermined locations, then load another program and move it to its own location, thus making more efficient use of the memory you have.

For example, suppose you have written four subroutines. For simplicity's sake, let's say that each subroutine was assembled to start at location 7D00 and end at 7DFF. You can take the first routine, and move it to 7300-73FF, relocating two other routines at 7400-74FF, and 7500-75FF. These routines can all be linked together, either internally with each other, or by a fifth routine you can write to access each subroutine.

The Program

The program is, of course, written in TI BASIC so that it can be used with the Mini Memory cartridge. The Mini Memory cartridge must be installed before the program can function. Failure to install it will cause the program to halt with an error.

When running the program, the user will have to answer some questions pertaining to the program's relocation. First, the program asks: DO YOU WANT TO DO LIMIT CHECKS FOR MEMORY BORDERS. (Y/N). In other words, the system is asking whether you want it to alert you if the relocator program tries to go outside of a predetermined area of memory. For example, let's say you already have a subroutine which takes up locations 7200-72FF. If you set the low memory limit to 7300, any time the relocator program tries to go below that location, it will stop moving, inform you, and ask if the move outside the limit is valid.

After telling the system that you wish to do limit checks, you must then enter the HIGH and LOW limits. Any time the program being relocated exceeds those limits, you will be notified and asked whether you want to enter new parameters or continue as is. In only one circumstance will you not be allowed to re-enter the parameters—if the relocator is already in the process of making the move when it encounters the limit error. Then your choice will be to either continue as is, or halt the program, and reload Mini Memory with the code to be relocated.

Next, the relocator will ask: FIRST ADDRESS OF MEMORY TO BE RELOCATED? (4 DIGIT HEX ADDRESS):. This will be the very first location in which code for your program occurs. (This includes data as well.) The next ques-

tion asks: LAST ADDRESS OF MEMORY TO BE RELOCATED? (4 DIGIT HEX ADDRESS):. After that is keyed in, you will need to enter the entry address of your program. This is the address that gets put into the REF/DEF table and tells the computer where the starting instruction is located. This location cannot be DATA or your code will not run.

The next question you are asked is RELOCATE MEMORY TO WHICH ADDRESS. You then enter the new address to which your code will be moved. To continue our example, if the first address of memory to be relocated were 7D00, and the address to which it moved were 7200, then all code starting at 7D00 would now start at 7200. Any internal location references (such as instructions which refer to relative addresses) will be modified so that after the move, the code should be ready to run.

There are two more options which you must consider before the move takes place. The first is whether or not you want a hard copy report of the relocation. If you have a printer, you may want to select this option to keep a permanent record while developing your assembly program.

The last option lets you either update the REF/DEF table or leave it as it is. You may be relocating code which you do not want entered in the REF/DEF table. If so, you can answer no to this option. If you choose to update the REF/DEF table, you can either replace an existing entry or add a completely new one. If you want to re-

EXPLANATION OF THE PROGRAM		
<i>Mini Memory Relocator</i>		
<hr/>		
Line Nos.		
100-160	Header and REMs.	1080-1240
170-240	Dimension arrays, and initialize variables.	1250-1680
250-360	Input option for limit checks.	1609-1790
370-610	Input memory addresses.	1800-1900
620-680	Input printer option.	1910-2030
690-940	Input option to update the REF/DEF table.	2040-2300
950-1070	Set up variables for the memory move.	2310-2490
		2500-3000
		3010
		Transfer memory.
		Update the REF/DEF table.
		Convert a hexadecimal value to a decimal value.
		Convert a decimal value to a hexadecimal value.
		Subroutine to input a 4 digit hexadecimal address and check limits.
		Print the final report on the screen.
		Optional print routine for the printer.
		Error messages.
		The end.

MAGIC Software, Inc. Presents

All new - Assembly Language Games
for the TI99/4 with Mini Memory or
expansion RAM. Fast - Fun - Fairly
Priced. Available on disc or cass.

BOMBER BOXER Cat & Mouse 3DMAZE and more....

	cass.	disc
Pricing: 1 game	\$12.00	\$14.00
2 games	\$22.00	\$24.00
3 games	\$31.00	\$32.00
4 games	\$40.00	\$40.00

For descriptions - send lg. SSAE
To ORDER - send check to:

MAGIC Software, Inc.
3148 South 14th Street
Milwaukee, Wisconsin 53215

TIME TUTOR

CHILDREN LEARN
TO TELL TIME
AND HAVE FUN DOING IT



WITH MUSIC
ANIMATION AND COLOR

Children love to watch the mouse
perform to nursery rhyme music
as a reward for correct answers.

FOR THE TI-99/4A
EXTENDED BASIC REQUIRED

SEND \$12.95 FOR CASSETTE

J.K.S. SOFTWARE
212 MAPLE ROAD
WALLINGFORD, PA. 19086

QUALITY SOFTWARE TEX-WRITER

A word processor that gives you
total control. Suitable for compos-
ing and printing personal letters,
resumes, mailing labels, business
records, and books.

Full upper and lower case—even
on the 99/4. Variable margin,
selectable line length, and
automatic page numbering.
Underlining. *Italics*. **Emphasized**. 4
print sizes including Expanded
and Compressed. Full cursor con-
trol for complete text editing
without retyping!

Requires Extended Basic, one
disk drive, and RS232 compati-
ble printer (Epson, T.I. Impact,
etc.)-----\$44.95

SEND FOR FREE CATALOG
OF ADDITIONAL PROGRAMS.

SEND CASH, CHECK, OR MONEY ORDER,
FREE SHIPPING IN THE U.S.
CANADIAN RESIDENTS ADD 20%.
PA RESIDENTS ADD 6% SALES TAX.

SOFT-TEX

TI 99/4(A) SOFTWARE
SOFT-TEX
3 WALNUT LANE
BERWYN, PA 19312

place an old entry, you will need to sup-
ply its name. If the name does not exist,
NOT IN THE TABLE will be displayed. Upon
entering the name of the REF/DEF entry,
the program will be relocated. When the
move is complete, a report will be printed
on the screen. If you earlier chose the
printer option, pressing ENTER will pro-
duce your printed report and then end the
program. Otherwise, you will need to
copy the information from the screen
before halting the program.

Now that you don't have to worry quite
as much about running short on memory,
you will be able to more fully explore the
world of Assembly Language program-
ming. The more ambitious you are as a
programmer, the more uses you will find
for this program (with a few modifications).
The only limit is your imagination, so hap-
py computing—and don't let the program
bug bite.

99'er

```

100 REM *****
110 REM * MEMORY RELOCATOR *
120 REM *****
130 REM BY MARTIN KROLL JR
140 REM 99'ER VERSION 2.7.1MM
150 REM
160 REM
170 REM DEFINE DIVISOR/MULTIPLIE
R FOR NUMBER CONVERSIONS
180 DIM S(4), PNAME$(20), PLOC$(20,2
)
190 LM=28672
200 HM=32767
210 S(1)=4096
220 S(2)=256
230 S(3)=16
240 S(4)=1
250 CALL CLEAR
260 PRINT "DO YOU WANT TO DO LIMIT
": "CHECKS FOR MEMORY BORDERS."
270 INPUT "(Y/N)": CT$
280 PRINT ::
290 IF (CT$<>"Y")*(CT$<>"N") THEN 2
60
300 IF CT$="N" THEN 370
310 PRINT "HIGH MEMORY LIMIT?"
320 GOSUB 1910
330 HM=TN2
340 PRINT "LOW MEMORY LIMIT?"
350 GOSUB 1910
360 LM=TN2
370 PRINT "FIRST ADDRESS OF MEMORY
TO BE RELOCATED?":
380 GOSUB 1910
390 A=TN2
400 FAP$=TEMP$
410 PRINT "LAST ADDRESS OF MEMORY
TO BE RELOCATED?":
420 GOSUB 1910
430 B=TN2
440 IF B<=A+1 THEN 2970
450 LAP$=TEMP$
460 PRINT "FIRST EXECUTION ADDRESS
OF THE PROGRAM; (ENTRY ADDRESS
S)?":
470 GOSUB 1910

```

```

480 C=TN2
490 EAP$=TEMP$
500 IF (C<A)+(C>B) THEN 2990
510 PRINT "RELOCATE MEMORY TO WHIC
H": "ADDRESS?":
520 GOSUB 1910
530 D=TN2
540 IF D>A THEN 570
550 K=0
560 GOTO 580
570 K=1
580 IF CT$="N" THEN 610
590 IF D<LM THEN 2590
600 IF D+(B-A)>HM THEN 2500
610 NPA$=TEMP$
620 PRINT "DO YOU WANT A PRINTED R
EPORT ON THE CHANGES IN MEMORY?"
630 INPUT "(Y/N)": PR$
640 PRINT ::
650 IF (PR$<>"Y")*(PR$<>"N") THEN 6
20
660 IF PR$="N" THEN 690
670 INPUT "PRINTER DEVICE:": DEV$
680 PRINT ::
690 PRINT "WANT REF/DEF TABLE UPDA
TED?"
700 INPUT "(Y/N)": RD$
710 PRINT ::
720 IF (RD$<>"Y")*(RD$<>"N") THEN 6
90
730 IF RD$="N" THEN 950
740 PRINT ::
750 PRINT "DO YOU WANT TO REPLACE
A": "REF/DEF ENTRY?":
760 INPUT "(Y/N)": RP$
770 IF (RP$<>"Y")*(RP$<>"N") THEN 7
50
780 IF RP$="N" THEN 910
790 INPUT "NAME OF OLD REF/DEF TAB
LE ENTRY TO BE REPLACED:": NM$
800 NM$=SEG$(NM$&"", 1, 6)
810 CALL PEEK(28702, N, D)

```

LEARN TO FLY!



REVIEWED IN JAN. 99'er

The Dow-4 Gazelle is a simulation of
a 4-place, single-engine, high perfor-
mance aircraft, which will provide fun
and challenge as you learn to fly. A high
quality program written by a professional
programmer/analyst who is also an
experienced instrument-rated pilot, the
Gazelle is a real-time simulation which
responds rapidly to the controls (within
one second on the average).

On your screen you see the instrument
panel, which has 10 dials with moving
pointers and 11 indicator lights. The
plane is flown with the joystick, while
the keyboard is used to control power,
flaps, fuel, etc.

The manual contains 30 pages of text,
a glossary, and seven full page figures.
It introduces you to the art of flying
and leads you, a step at a time, from
novice to professional. Learn to take-off,
land, navigate, fly instrument approaches,
and more. If you get into trouble, you
can freeze the action in case you need
time to assess your situation. Sound
effects add to realism.

This program pushes the TI-99/4(A) to
its limits. For both 99/4 and 99/4A,
Does not require anything other than a
joystick and cassette recorder.

Send \$30.00 to:
John T. Dow

6360 Caton
Pittsburgh, PA 15217
PA residents add 6%

Continued on p. 68


```

820 RDT=N*256+D
830 IF RDT>32760 THEN 890
840 CALL PEEK(RDT,C1,C2,C3,C4,C5,C
6)
850 RDT$=CHR$(C1)&CHR$(C2)&CHR$(C3
)&CHR$(C4)&CHR$(C5)&CHR$(C6)
860 IF RDT$=NM$ THEN 910
870 RDT=RDT+8
880 GOTO 830
890 PRINT :="NOT IN THE TABLE":
900 GOTO 750
910 INPUT "NAME OF NEW REF/DEF TAB
LE ENTRY:":TITLE$
920 TITLE$=SEG$(TITLE$," ",1,
6)
930 PN=1
940 PNAME$(PN)=TITLE$
950 DIF=A-D
960 C2=C-DIF
970 VA=C2
980 GOSUB 1810
990 PLOC$(1,2)=VAL$
1000 IF K=0 THEN 1050
1010 A1=B
1020 B1=A
1030 STP=-2
1040 GOTO 1080
1050 A1=A
1060 B1=B
1070 STP=2
1080 FOR LOC=A1 TO B1 STEP STP
1090 IF LOC<32768 THEN 1120
1100 LOC1=LOC-65536
1110 GOTO 1130
1120 LOC1=LOC
1130 CALL PEEK(LOC1,X,Y)
1140 IF (X<113)+(X>127) THEN 1220
1150 Z=X*256+Y
1160 V=Z-DIF
1170 IF OT$="N" THEN 1200
1180 IF (V>LM-1)*(V<HM+1) THEN 1200
1190 GOSUB 2660
1200 X=INT(V/256)
1210 Y=V-X*256
1220 CALL LOAD(LOC-DIF,X,Y)
1230 NEXT LOC
1240 IF RD$="N" THEN 1670
1250 REM UPDATE REF/DEF TABLE
1260 CALL PEEK(28700,L,M,N,D)
1270 FFAM=L*256+M
1280 LFAM=N*256+D
1290 IF RP$="Y" THEN 1310
1300 LFAM=LFAM-B
1310 VA=LFAM
1320 GOSUB 1810
1330 LAVAILS=VAL$
1340 PLOC$(1,1)=VAL$
1350 FFAM=B-DIF+2
1360 VA=FFAM-2
1370 GOSUB 1810
1380 FAVAILS=VAL$
1390 L=INT(FFAM/256)
1400 M=FFAM-L*256
1410 N=INT(LFAM/256)
1420 D=LFAM-N*256
1430 CALL LOAD(28700,L,M,N,D)
1440 IF RP$="N" THEN 1470
1450 LOC=RDT
1460 GOTO 1480
1470 LOC=LFAM
1480 FOR SS=1 TO 6
1490 CALL LOAD(LOC,ASC(SEG$(TITLE$,
SS,1)))
1500 LOC=LOC+1
1510 NEXT SS
1520 J=INT(C2/256)
1530 K=C2-J*256
1540 IF RP$="N" THEN 1570
1550 CALL LOAD(RDT+6,J,K)
1560 GOTO 1580
1570 CALL LOAD(LFAM+6,J,K)
1580 PN=PN+1
1590 IF LFAM+((PN-1)*8)>32760 THEN
2040
1600 CALL PEEK(LFAM+((PN-1)*8),C1,C
2,C3,C4,C5,C6,A1,A2)
1610 PNAME$(PN)=CHR$(C1)&CHR$(C2)&C
HR$(C3)&CHR$(C4)&CHR$(C5)&CHR$(
C6)
1620 VA=A1*256+A2
1630 GOSUB 1810
1640 PLOC$(PN,2)=VAL$
1650 VA=LFAM+((PN-1)*8)

```

```

1660 GOSUB 1810
1670 PLOC$(PN,1)=VAL$
1680 GOTO 1580
1690 REM CONVERT TO DECIMAL
1700 TN2=0
1710 FOR P=1 TO 4
1720 TEMP$=SEG$(TEMP$,P,1)
1730 IF ASC(TEMP$)>57 THEN 1780
1740 TN=ASC(TEMP$)-48
1750 TN2=TN2+TN*(P)
1760 NEXT P
1770 RETURN
1780 TN=ASC(TEMP$)-55
1790 GOTO 1750
1800 REM CONVERT TO HEX
1810 VAL$=""
1820 FOR P=1 TO 4
1830 V=INT(VA/S(P))
1840 VA=VA-(V*S(P))
1850 IF V>9 THEN 1880
1860 VAL$=VAL$&STR$(V)
1870 GOTO 1890
1880 VAL$=VAL$&CHR$(V+55)
1890 NEXT P
1900 RETURN
1910 INPUT "(4 DIGIT HEX ADDRESS):"
:TEMP$
1920 IF LEN(TEMP$)>4 THEN 1910
1930 IF LEN(TEMP$)=4 THEN 1950
1940 TEMP$=SEG$("0000"&TEMP$,LEN(TE
MP$)+1,4)
1950 FOR TT=1 TO 4
1960 TT1=ASC(SEG$(TEMP$,TT,1))
1970 IF (TT1<48)+(TT1>70)+((TT1>57)
*(TT1<65)) THEN 1910
1980 NEXT TT
1990 GOSUB 1700
2000 IF CT$="N" THEN 2020
2010 IF (TN2<LM)+(TN2>HM) THEN 2850
2020 PRINT :
2030 RETURN
2040 CALL CLEAR
2050 PRINT TAB(19);"OLD NEW"
2060 PRINT "FIRST ADD.":TAB(19);FAP
$:TAB(25);NPA$
2070 PRINT "LAST ADD.":TAB(19);LAP$
:TAB(25);FAVAIL$
2080 PRINT "ENTRY POINT":TAB(19);EA
P$:TAB(25);PLOC$(1,2)
2090 PRINT "REF/DEF STARTS AT":TAB(
25);LAVAIL$
2100 PRINT :="MEMORY RELOCATED ";(-
1)*DIF;"BYTES"
2110 PRINT :
2120 PRINT "REF/DEF TABLE ENTRIES"
2130 SS1=1
2140 PRINT :="LOCATION NAME E
NTRY"
2150 FOR SS=SS1 TO SS1+4
2160 IF SS=21 THEN 2250
2170 IF PNAME$(SS)="" THEN 2250
2180 PRINT PLOC$(SS,1);TAB(12);PNAM
E$(SS);TAB(22);PLOC$(SS,2)
2190 NEXT SS
2200 PRINT :="PRESS ENTER TO CONTIN
UE"
2210 CALL KEY(0,K,S1)
2220 IF S1=0 THEN 2210
2230 SS1=SS
2240 GOTO 2150
2250 IF PR$="Y" THEN 2270
2260 PRINT :="RECORD ALL DATA, AND"
2270 PRINT "PRESS ENTER TO CONTINUE
"
2280 CALL KEY(0,K,S1)
2290 IF S1=0 THEN 2280
2300 IF PR$="N" THEN 2490
2310 OPEN #1:DEV$
2320 PRINT #1:"MEMORY RELOCATION RE
PORT":
2330 PRINT #1:TAB(25);"OLD";TAB(35)
;"NEW"
2340 PRINT #1:"FIRST ADDRESS":TAB(2
5);FAP$;TAB(35);NPA$
2350 PRINT #1:"LAST ADDRESS":TAB(25
);LAP$;TAB(35);FAVAIL$
2360 PRINT #1:"ENTRY POINT":TAB(25)
;EAP$;TAB(35);PLOC$(1,2)
2370 PRINT #1:"REF/DEF STARTS AT":T
AB(35);LAVAIL$
2380 PRINT #1:="MEMORY RELOCATED";
(-1)*DIF;"BYTES"
2390 PRINT #1:
2400 PRINT #1:"REF/DEF TABLE ENTRI
E S"

```

```

2410 PRINT #1:
2420 PRINT #1:"LOCATION":TAB(12);"N
AME":TAB(25);"REF/DEF ENTRY PO
INT"
2430 FOR SS=1 TO 20
2440 IF PNAME$(SS)="" THEN 2470
2450 PRINT #1:PLOC$(SS,1);TAB(12);P
NAME$(SS);TAB(25);PLOC$(SS,2)
2460 NEXT SS
2470 PRINT #1:="END OF REPORT"
2480 CLOSE #1
2490 STOP
2500 PRINT :="YOU ARE TRYING TO M
OVE ABOVEYOUR LIMIT"
2510 VA=D+(B-A)
2520 GOSUB 1810
2530 PRINT "HIGH LIMIT NEEDS TO BE
":VAL$
2540 PRINT :="DO YOU WISH TO CHANG
E THE PARAMETERS?"
2550 INPUT "(Y/N)":CHP$
2560 PRINT :
2570 IF (CHP$<>"N")*(CHP$<>"Y") THEN
2540
2580 IF CHP$="Y" THEN 250 ELSE 620
2590 PRINT :="YOU ARE TRYING TO M
OVE BELOWYOUR LIMIT":
2600 PRINT "YOUR LOW LIMIT NEEDS TO
BE":TEMP$:
2610 PRINT "DO YOU WISH TO CHANGE T
HE PARAMETERS?"
2620 INPUT "(Y/N)":CHP$
2630 PRINT :
2640 IF (CHP$<>"N")*(CHP$<>"Y") THEN
2610
2650 IF CHP$="Y" THEN 250 ELSE 610
2660 VA=V
2670 GOSUB 1810
2680 TO$=VAL$
2690 VA=Z
2700 GOSUB 1810
2710 FROM$=VAL$
2720 VA=LOC1
2730 GOSUB 1810
2740 PRINT :="YOUR PROGRAM WILL N
EED TO WORK OUTSIDE OF YOUR
LIMITS"
2750 PRINT :="THE INSTRUCTION AT LO
CATION":VAL$;" ORIGINALLY USED
THE":"ADDRESS REFERENCE ";FRO
M$;"."
2760 PRINT :="IT NOW NEEDS TO CHANG
E THAT ADDRESS REFERENCE TO ";
TO$;"."
2770 PRINT "DO YOU WISH TO HALT THI
S":"PROGRAM AND START OVER?"
2780 INPUT "(Y/N)":CHP$
2790 PRINT :
2800 IF (CHP$<>"N")*(CHP$<>"Y") THEN
2770
2810 IF CHP$="Y" THEN 2830
2820 RETURN
2830 PRINT :="YOU MUST RELOAD MEM
ORY WITH YOUR ASSEMBLY PROGRAM
BEFOREUSING THIS PROGRAM AGAI
N":
2840 STOP
2850 PRINT :="THAT ADDRESS IS OUT
SIDE OF YOUR LIMITS":
2860 VA=HM
2870 GOSUB 1810
2880 PRINT "HIGH MEMORY LIMIT=":VAL
$
2890 VA=LM
2900 GOSUB 1810
2910 PRINT "LOW MEMORY LIMIT=":VAL$
2920 PRINT :="YOUR INPUT WAS FOR AD
DRESS":TEMP$:
2930 PRINT "IS THIS A VALID ADDRESS
?"
2940 INPUT "(Y/N)":CHP$
2950 IF (CHP$<>"Y")*(CHP$<>"N") THEN
2920
2960 IF CHP$="Y" THEN 2030 ELSE 260
2970 PRINT :="YOU HAVE ENTERED A L
AST":"ADDRESS LOWER THAN OR EQ
UAL TO THE FIRST ADDRESS":
2980 GOTO 260
2990 PRINT "THE ENTRY ADDRESS MUST
BE":"BETWEEN THE FIRST, AND LA
ST ADDRESSES. TRY AGAIN"
3000 GOTO 260
3010 END

```