

Extended Basic V1.0 for the MC-10
User Manual – 20K ram needed
Adds 27 extra basic commands to the Tandy MC-10
Gary Furr 1986

Known machine language programs that won't work with Extended Basic V1.0

- Little-R October 1985 p47
- @ Edit February 1986 p64
- Rom1.1
- All Graham Pollock's hires programs

INDEX

Loading instructions	P2
Entering Commands	P2
Error messages	P2
Demonstration programs	P11

Commands

Programming aids

AUTO	P3
CGOSUB	P3
CGOTO	P3
EDIT	P4
LRESET	P4
MERGE	P5
M>SAVE	P5
PAUSE	P5
RENUM	P6

Screen aids

CCOL	P6
DUMP	P6
PLOT	P7
SPEED	P7
SPDOFF	P7

Numeric aids

HEX	P7
DEC	P7

Structured programming

PROC	P8
DPROC	P8
EPROC	P8
POP	P9

Error trapping

ERROR	P9
ERROFF	P9
ERRLN	P9

Sound

EFFECT	P10
KEY	P10
BEEP	P10
BPOFF	P10

Loading instructions

To load **Extended Basic** enter:

CLOADM

After loading the following will be displayed on the screen:

EXTENDED BASIC V1.0

FURR 1986

OK

All of the Extended Basic commands are now included in the operating system of the MC-10, and may be used at any time like the normal basic commands.

Extended Basic uses approximately 2.330K bytes.

Entering Commands

All of the Extended Basic commands are entered in the same way as those that are in standard basic, most of the new commands can be incorporated in programs except *DEC* and *HEX*.

When the reset button is pressed, the sign-on-message will be displayed and the current program in memory won't be destroyed.

Note: The RESET button doesn't disable any commands.

Error messages

?PROC NF ERROR IN (line number)

The procedure you have called can't be found

?EPROC ERROR

No procedure has been called for the application to return to

?RG ERROR during a POP command

No GOSUB or PROC command has been used in the program so far

? ERROR during an ERROR(x) command

The line number contained the brackets can't be located

If a line number above 63999 is used in AUTO then an ERROR is displayed

If a number below 1 or above 8 is used in CCOL then a MICROSOFT error is displayed

AUTO

FORMAT: AUTO start line number, increment

PURPOSE: To generate program line numbers at a specified increment

When the AUTO command is entered, the start program line number is displayed, with the cursor following it, ready to type a normal basic line.

Each time you finish typing your line of basic code after pressing <ENTER>, the increment that you specified will be added to the number of the previous line.

To terminate the AUTO command, press <ENTER> twice after the line number is displayed.

Example: To generate line numbers automatically in intervals of 10 beginning at 10.

ENTER: AUTO 10,10

DISPLAYED: 10

ENTER: PRINT " DEMO OF AUTO" <ENTER>

DISPLAY: 20

ENTER: PRINT "FINISHED" <ENTER>

DISPLAY : 30 <ENTER> <ENTER>

CGOSUB

FORMAT: CGOSUB variable

PURPOSE: To GOSUB a line number determined by a variable or mathematical formula.

In normal basic the GOSUB command doesn't allow to GOSUB a line number determined by a variable , CGOSUB does.

Example: NEW <ENTER>

```
10 REM * EXAMPLE OF CGOSUB *
```

```
20 CLS : FORNO=1TO 5
```

```
30 CGOSUB NO*10+70
```

```
40 PAUSE1,; : NEXTNO : END
```

```
80 PRINT "ROUTINE 1" : RETURN
```

```
90 PRINT "ROUTINE 2" : RETURN
```

```
100 PRINT "ROUTINE 3" : RETURN
```

```
110 PRINT "ROUTINE 4" : RETURN
```

```
120 PRINT "ROUTINE 5" : RETURN
```

CGOTO

FORMAT: CGOTO line number

PURPOSE: Works in the same way a CGOSUB except CGOTO is used

CGOTO works the same way as CGOSUB except that CGOTO doesn't use a return.

```
5 REM ** DEMO OF CGOTO**
```

```
10 A=RND(3)
```

```
20 IF A=0 THEN10
```

```
30 IF A=1 THEN A=80
```

```
40 IF A=2 THEN A=90
```

```
50 IF A=3 THEN A=100
```

```
60 CGOTO A
```

```
70 END
```

```
80 PRINT "A = ";A : END
```

```
90 PRINT "A = ";A : END
```

```
100 PRINT "A = ";A : END
```

EDIT By H. Allan Curtis

FORMAT: EDIT line number

PURPOSE: To edit a line of basic code

Because Extended Basic wouldn't allow Little-e to be run, we decided to make Little-e a built in command. Little-e remains all the same features as the original except after you have finished editing a line number use <ENTER> to disable the command and return to basic.

The right arrow key moves the cursor left and the down arrow key moves the cursor down one line, it also moves the cursor up a line only when you are positioned at the bottom of line#4.

Keywords can be used. Eg. Control K will display the SOUND command during editing.

Deletion: L.DEL – Deletes the character under the cursor.

Insertion: SHIFT @ - Inserts one space at a time.

After editing the line make sure the cursor is positioned at the end of the text you require, as anything after the cursor will be eliminated.

LRESET

FORMAT: LRESET line number

PURPOSE: Reset data pointer to a certain line number

Standard basic data is always read one DATA statement per one READ command.

RESET resets all data statements, and LRESET only resets one data statement. Therefore all of the data doesn't need to be read again, just specify a particular line and only that line will be reset, ready to be READ again.

Example:

```
10 REM ** EXAMPLE OF LRESET **
20 CLS : PLOT(2,5)
30 PRINT "INPUT A CATEGORY LISTED BELOW";
35 PRINT
40 PRINT "1) CARS", "2) COMPUTERS"
50 PRINT "3) NAMES", "4) COMMANDS"
60 A$=INKEY$ : IFA$<"0" OR A$>"4" THEN 60
70 IF A$="1" THEN LRESET140
80 IF A$="2" THEN LRESET150
90 IF A$="3" THEN LRESET160
100 IF A$="4" THEN LRESET170
110 FORG=1TO4 : READL$ : PRINTL$ : NEXTG
120 PAUSE5,; : GOTO20
140 DATA HOLDEN,FORD,MAGNA,MINI
150 DATA CO/CO,*MC-10*,T100,T200
160 DATA JIM,KEVIN,GARY,GRAHAM
170 DATA AUTO,RENUM,ERROR,EDIT
```

MERGE

FORMAT: MERGE "name"

PURPOSE: To tag a basic program onto the end of another basic program currently in memory.

If no name is defined the first program found will be used.

Example: Load a basic program into the MC-10, then enter MERGE "program name" <ENTER>.

Press play on the tape, when the OK is display on the screen, enter LIST and displayed will be the first basic program followed by the second basic program.

****CAUTION****

The merged program will follow the program currently in memory. In most cases it will be necessary to use the RENUM command to "fix-up" the line numbers.

M>SAVE

FORMAT: M>SAVE start, end, "name"

PURPOSE: To save machine language on tape.

M>SAVE – saves a machine language program from the start to the end address given. Only decimal numbers can be entered.

Example: Enter the below program and RUN. The <BREAK> to will terminate.

```
4 REM **LETTER WRITER**
5 REM <BREAK> TO END
10 FOR C=24576 TO 24594
20 READX : POKEC,X : NEXTC
30 EXEC 24576
40 DATA 189,251,212,189,248,101,129
50 DATA 3,39,6,189,249,201,126,96
60 DATA3,126,226,113
```

ENTER: M>SAVE 24576,24594 "LETTER"

RESULT: PRESS RECORD THEN ENTER

The above program has now been saved onto tape. To load it back in:
CLOADM "LETTER" and EXEC 24576

PAUSE

FORMAT: PAUSE number of seconds , "message"

PURPOSE: To stop program execution for a given amount of time and display a message.

PAUSE causes the computer to wait the given amount of time specified.

The "message" is optional.

Example 1:

PAUSE 5, "PAUSE COMMAND" <ENTER>

RESULT : PAUSE COMMAND will be displayed on the screen for 5 seconds.

Example 2:

PAUSE5,; <ENTER>

RESULT: The computer will wait for 5 seconds.

Note: PAUSE can be used in the same way as PRINT – TAB(x) and @x, "message"

PAUSE 2,@120,"I" or PAUSE 2, TAB(2), "J"

RENUM

FORMAT: RENUM start line, increment, new start line

PURPOSE: To automatically renumber all program lines after the specified start line number.

RENUM changes every line number after the specified start line number. All subsequent line numbers are then changed to the selected increment. The new start line variable is where the program will start to be renumbered from.

Example: To renumber all lines in a basic program.

```
10 FOR X=1 TO 10
20 PRINT X
30 EFFECT 150,30
40 NEXT X
```

Type: RENUM 10,15,100 <ENTER> LIST <ENTER>

DISPLAY:

```
100 FOR X=1 TO 10
115 PRINT X
130 EFFECT 150,30
145 NEXT X
```

Note: RENUM doesn't renumber lines associated with GOSUB or GOTO.

CCOL

FORMAT: CCOL cursor colour

PURPOSE: To change the colour of the cursor

The colour code to which the CCOL command uses are located on page 118 of your MC-10 operation and language manual.

Numbers below 1 and greater than 8 will result in a Microsoft Error

DUMP

FORMAT: DUMP

PURPOSE: To give a screen dump onto the printer

Load a program which is graphical. Somewhere in the program, insert a new line containing the command DUMP

Example:

```
10 REM ** EXAMPLE OF DUMP**
20 F=134 : H=416 : T=128 : G$=CHR$(F)
30 PRINT@T,G$
40 T=T+1 : IF T=H THEN 60
50 GOTO30
60 PRINT "EXAMPLE OF DUMP"
70 DUMP : CLS
```

When line 70 executed, a screen dump will occur onto your printer.

Ensure the printer is switched on.

PLOT

FORMAT: PLOT(x,y)

PURPOSE: To position the cursor at the coordinates defined

The PLOT command is another text positioning command that is easier to use.

There is no adding or subtracting needed. The value for x is between 0 - 31 and for y the value is between 0 -15.

Page 93 of the MC-10 manual shows a grid with all the graphics locations, which can be used as a reference guide when using the PLOT command.

Example:

```
10 CLS
20 PLOT(5,5)
30 PRINT "EXTENDED"
40 PLOT(13,6)
50 PRINT "BASIC"
60 END
```

SPEED

FORMAT: SPEED delay

PURPOSE: To slow the printing of characters onto the screen

Delay value = 1 – 255. Where 1 is fast and 255 is very slow.

This command can be used in two ways:

1. Slow list
2. Slowing down printing onto the screen during a programs execution.

Note: The SPEED command alters how fast the keyboard will respond to the keys being pressed and being displayed on the screen.

SPEED must be disabled before saving or loading programs and before using your printer.

SPDOFF

FORMAT: SPDOFF

PURPOSE: To disable the SPEED command

HEX

FORMAT: HEX decimal, decimal

PURPOSE: To convert a decimal number into hex number

The maximum decimal number that can be converted is 65535. Multiple conversions can be performed if separated by a comma.

Example:

```
ENTER: HEX 65535,16384 <ENTER>
RESULT: $FFFF, $4000
```

DEC

FORMAT: DEC hex number, hex number

PURPOSE: To convert a hex number to a decimal number

Multiple conversions can be performed if separated by a comma. The result will be an eight or sixteen bit number.

Example:

```
ENTER: DEC FFFF, 009D, 7A00 <ENTER>
RESULT: 65535, 157, 31232
```

Note: Four characters must be specified.

PROC

FORMAT: PROC procedure name

PURPOSE: To transfer the program execution to a specific line of code

The PROC command acts in the same way as the GOSUB command, except that a name is used in the place of a line number.

Everything that follows a PROC on the same line is used as the name of the procedure being called, so therefore the PROC and the procedure name must not be followed by any other programming on the same line.

Note:

- The procedure name is limited to eight characters
- There must be no space between the PROC command and the procedure name

DPROC

FORMAT: DPROC procedure name

PURPOSE: To label a program routine

DPROC enables you to label program routines, and call these routines by name when they are required. All characters on the line which follow the DPROC are taken as the name of the procedure. Therefore, DPROC and the procedure name must not be followed by any other code on the same program line.

Note:

- The procedure name is limited to eight characters
- There must be no space between the PROC command and the procedure name

EPROC

FORMAT: EPROC

PURPOSE: To indicate the end of a procedure

This command acts in the same way as the RETURN command. When the procedure ends, the program returns to the line following that on which the procedure was called.

Example of PROC, DPROC and EPROC

```
05 REM ** DEMO OF PROC, DPROC EPROC **
10 CLS : PLOT(5,5) : PRINT "A' TO STOP";
20 PROC SOUND
30 A$=INKEY$ : IF A$="A" THEN 50
40 GOTO 20
50 PROC OFF
70 END
80 DPROC SOUND
90 POKE 34943,9 : POKE34958,8
100 FOR F =225 TO 30 STEP-2 : EFFECTF,2
110 EFFECTF-30,2 : NEXTF: PROC OFF
120 EPROC
130 DPROC OFF
140 POKE 34943,11 : POKE34958,9 : EPROC
```

POP

FORMAT: POP

PURPOSE: GOSUB without RETURN, PROC without EPROC

The POP command allows the use of the GOSUB, CGOSUB and PROC commands without having to return to where it started from when the routine that was called has been completed.

Example:

```
10 PRINT "EXTENDED"  
20 GOSUB 50  
30 PRINT "V1.0"  
40 END  
50 POP : PRINT "BASIC"  
60 RETURN
```

In theory the above program if the POP command wasn't in line 50, the output would be EXTENDED BASIC V1.0 displayed on the screen under each other, but with the POP command in line 50, the result will be EXTENDED BASIC displayed on the screen.

ERROR

FORMAT: ERROR (line number)

PURPOSE: To trap errors in programs

If an error is found during a program execution, the program will be diverted to the line number specified in the brackets. It can be used to produce your own personalised error messages. It is very important that after finding an error, that the ERROFF command be executed. ERROR must be disabled before LOADING.

ERROFF

FORMAT: ERROFF

PURPOSE: To disable the ERROR command

This command must be executed after displaying your new error message, and before CLOADing any programs.

ERRLN

FORMAT: ERRLN

PURPOSE: Holds the last line number that an error occurred in

This command can be used in direct mode or within a program.

Example of ERROR/ERROFF and ERRLN

Deliberate mistake in line 40

```
10 ERROR(100)  
20 FOR B = 1 TO 10  
30 EFFECTB,4  
40 PRONT "HI"  
50 NEXTB : END  
100 CLS : PRINT "ERROR IN LINE "; : ERRLN  
110 ERROFF : END
```

EFFECT

FORMAT: EFFECT tone, duration

PURPOSE: To create sound effects

The EFFECT command allows you to produce better sound for arcade style games.

Tone – specifies the sound and is represented by a number between 1 – 255.

Duration – specifies the length of time the sound and is also between 1 – 255.

Both tone and duration must be specified.

Example:

```
10 FOR F= 1 TO 225
20 EFFECT F,1 : EFFECT F+30,1
30 NEXT F
```

EFFECT #2 (bonus command)

Enable: POKE 34943,9 : POKE 34958,8

Disable: POKE 34943,11 : POKE 34958,9

Can only be used in conjunction with the EFFECT command

Example:

```
10 CLS
20 FORD=1TO225:EFFECTD,1
30 EFFECTD+30,2 : NEXTD
40 PRINT "PRESS A KEY"
50 A$=INKEY$ : IF A$="" THEN 50
60 REM ENABLE EFFECT #2
70 POKE 34943,9 : POKE 34958,8
80 FOR G = 255 TO 30 STEP -1 : EFFECTG-29,2
90 EFFECTG,1 : NEXTG
100 REM DISABLE EFFECT #2
110 POKE 34943,11 : POKE 34958,9
```

KEY

FORMAT: KEY

PURPOSE: Produces a sound when any key is depressed

Every key on the keyboard has been assigned a note. Each time a key is pushed it produces its very own distinctive note. Except CONTROL and SHIFT keys.

BEEP

FORMAT: BEEP

PURPOSE: Produces a sharper and shorter pitched sound when any key is pressed

BEEP must be disabled before using your printer and before saving or loading programs.

BPOFF

FORMAT: BPOFF

PURPOSE: Disables both the KEY and BEEP commands

Demonstration programs

```
10 REM ** THE SOUND SYSTEM**
20 CLS0
30 PLOT(7,4) : PRINT "SOUND SYSTEM";
40 PLOT(8,6) : PRINT"1) UP (EFFECT)";
50 PLOT(8,8) : PRINT"2) DOWN (EFFECT#2)";
60 PLOT(8,10) : PRINT"3) UFO TAKE-OFF";
70 A$=INKEY$ : IF A$="" THEN 70
80 IF A$="1" THEN PROCUP
90 IF A$="2" THEN PROCDOWN
100 IF A$="3" THEN PROCUFO
110 GOTO 70
120 DPROCUP
130 G=1: H=255 : J=1: GOSUB380:POP:GOTO30
160 DPROCDOWN
170 GOSUB420:G=-1 : H=1 :J=255:GOSUB380
190 GOSUB430 : POP :GOTO30
210 DPROCUFO
220 GOSUB420
230 A=0: FORX=1TO3:FORF=255TO210STEP-1:EFFECTF,2:NEXTF
250 FORG=1TO3:FORC=1TO40:EFFECTC,1:NEXT:NEXT:NEXTX
260 FORF=255TO210STEP-1:EFFECTF,1:EFFECTRND(255),1: NEXT
270 A=A+1 : IFA=2THEN290
280 GOTO260
290 FORF=16TO255STEP19
300 EFFECTF,1:GOSUB420:EFFECTF-10,1
310 GOSUB430:NEXT:B=0
330 EFFECT243,5:B=B+1:IF B=30THEN350
340 GOTO330
350 FOR F=210TO255:EFFECTF,20:NEXT
370 POP:GOTO30
380 FORF=JTOHSTEPG:EFFECTF,1:NEXT
390 RETURN
420 POKE34943,9:POKE34958,8:RETURN
430 POKE34943,11:POKE34958,9:RETURN
```

```
10 REM ** SAVE SCREEN – PRESS A KEY **
20 REM ** LOAD SCREEN – PRESS "1" **
30 CLS0
40 N=RND(63) : M=RND(31) :B=RND(8)
50 SET (N,M,B) :EFFECTN,2:EFFECTM,1
60 A$=INKEY$:IF A$="" THEN 40
80 IF A$="1" THEN 110
90 M>SAVE16384,16896,"SCREEN"
110 CLOADM:EFFECT45,60
120 GOTO120
```