dimension

MICRO CRAFT CORPORATION

January 5, 1984

The Unibasic Reference Manual (part number 680-0200-200) has been revised. Here are the new, revised pages that are to be placed into the manual.

The pages to be replaced are as follows:

Page to be Replaced	New Page
7,8	7,8
9,10	9,10
13,14	13,14
21,22	21,22
25,26	25,26
33,34	33,34
57,58	57,58
61,62	61,62
73,74	73,74
75,76	75,76
85,86	85,86
135,136	135,136

UNIBASIC

REFERENCE MANUAL

Micro Craft Corporation

AS VERSION

680-0200-200A

PRELIMINARY

01/05/84 REVISION

NOTICE

Micro Craft Corporation reserves the right to make improvements in the product described in this manual at any time and without notice.

DISCLAIMER OF ALL WARRANTIES AND LIABILITY

MICRO CRAFT CORPORATION MAKES NO WARRANTIES, EITHER EXPRESS OR IMPLIED, WITH RESPECT TO THIS MANUAL OR WITH RESPECT TO THE SOFTWARE DESCRIBED IN THIS MANUAL, ITS QUALITY, PERFORMANCE, MERCHANTABILITY, OR FITNESS FOR ANY PARTICULAR PURPOSE. MICRO CRAFT CORPORATION SOFTWARE IS SOLD OR LICENSED "AS IS." THE ENTIRE RISK AS TO ITS QUALITY AND PERFORMANCE IS WITH THE BUYER. SHOULD THE PROGRAMS PROVE DEFECTIVE FOLLOWING THEIR PURCHASE, THE BUYER (AND NOT MICRO CRAFT CORPORATION ITS DISTRIBUTOR, OR ITS RETAILER) ASSUMES THE ENTIRE COST OF ALL NECESSARY SERVICING, REPAIR, OR CORRECTION AND ANY INCIDENTAL OR CONSEQUENTIAL DAMAGES. IN NO EVENT WILL MICRO CRAFT CORPORATION BE LIABLE FOR DIRECT, INDIRECT, INCIDENTAL, OR CONSEQUENTIAL DAMAGES RESULTING FROM ANY DEFECT IN THE SOFTWARE, EVEN IF MICRO CRAFT CORPORATION HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES. SOME STATES DO NOT ALLOW THE EXCLUSION OR LIMITATION OF IMPLIED WARRANTIES OR LIABILITY FOR INCIDENTAL OR CONSEQUENTIAL DAMAGES, SO THE ABOVE LIMITATION OR EXCLUSTION MAY NOT APPLY TO YOU.

Micro Craft Corporation
4747 Irving Blvd.
Dallas, Texas 75247
(214)63Ø-2562

GETTING STARTED

The Dimension 68000 system is shipped with a "SYSTEM 1" diskette and a "SYSTEM 2" diskette. Before doing ANYTHING else, make a copy of the "system diskettes that were shipped with your DImension 68000 system. A step by step procedure for making these copies, or "BACKING-UP" these diskettes is included in the "BACKING-UP" APPENDIX.

Always make a back-up of any diskettes received from Micro Craft, Inc.

If you should damage the "SYSTEM" diskette or the "LANGUAGES UTILITIES" diskette, additional diskettes may be purchased from Micro Craft, Inc., for $350.00 plus shipping and handling fees.

RUNNING UNIBASIC

To use the Micro Craft, Inc., UNIBASIC interpreter on the Dimension 68000 system, insert the "SYSTEM 1" diskette into the "A" diskette drive. Then, either type

```
BASIC
```

or

```
BASIC filename
```

where filename = the name of the file that contains the basic program to be run.

FILE NAMING CONVENTIONS

Filenames are a combination of the CP/M-68K and the APPLESOFT (TM) naming conventions. All UNIBASIC filenames consist of three parts:

- the FILENAME
- the FILETYPE
- the DRIVE SPECIFICATION

The FILENAME consists of from one to eight characters. The first character must be alphabetic. All of the rest of the characters may be either alphabetic or numeric.

The FILETYPE consists of a period (.), followed by up to three characters. The characters may be either alphabetic or numeric. CP/M normally reserves certain FILETYPES (such as .BAS for BASIC programs or .$$$ for temporary files). UNIBASIC does not require that it's programs have a specific FILETYPE.

The DRIVE SPECIFICATION consists of a comma (,), followed by a D, followed by either a 1, a 2, a 3, or a 4. The numbers 1, 2, 3, and 4 correspond to the disk drives A:, B:, C:, AND D:. If no DRIVE SPECIFICATION is included, the system will use the system default disk drive.

MODES OF OPERATION

When UNIBASIC is initialized it displays the prompt character ":". The prompt character indicates that UNIBASIC is at the command level; that is, it is ready to accept commands. At this point, UNIBASIC may be used in either of two modes: direct mode or indirect mode.

In direct mode, UNIBASIC statements and commands are not preceded by line numbers. They are executed as they are entered. Results of arithmetic and logical operations may be displayed immediately and stored for later use but the instructions themselves are lost after execution. Direct mode is useful for debugging and for using UNIBASIC as a "calculator" for quick computations that do not require a complete program.

Indirect mode is used for entering programs. Program lines are preceded by line numbers and are stored in memory. The program stored in memory is executed by entering the RUN command.

LINE FORMAT

UNIBASIC program lines have the following format (square brackets indicate optional input):

```
   nnnnn UNIBASIC-STATEMENT[:UNIBASIC-STATEMENT...] <CR>
```

More than one UNIBASIC statement may be placed on a line, but each must be separated from the last by a colon.

A UNIBASIC program line always begins with a line number and ends with a carriage return. A line may contain a maximum of 255 characters.

A line may contain up to 256 characters. When the line displayed requires more characters than a physical CRT line contains, UNIBASIC automatically continues displaying the line on the next physical line of the CRT.

LINE NUMBERS

Every UNIBASIC program line begins with a line number. Line numbers indicate the order in which the program lines are stored in memory. Line numbers are also used as references in branching and editing. line numbers must be in the range of Ø to 63999.

A period (.) may be used in the LIST, and the DELETE commands to refer to the current line.

CHARACTER SET

The UNIBASIC character set is comprised of the alphabetic characters, numeric characters, and special characters.

The alphabetic characters in UNIBASIC are the upper-case letters of the alphabet.

The UNIBASIC numeric characters include the digits Ø through 9.

In addition, the following special characters and terminal keys are recognized by UNIBASIC:

CHARACTER	ACTION
	Blank or Space
=	Equals sign or assignment symbol
+	Plus sign
-	Minus sign
*	Asterisk or multiplication symbol
/	Slash or division symbol
^	Up arrow or exponentiation symbol
(	Left or open parenthesis
)	Right or close parenthesis
%	Percent
#	Number or pound sign
$	Dollar sign
!	Exclamation point or "bang"
[	Left or open bracket
]	Right or close bracket
,	Comma
.	Period
;	Semicolon
:	Colon
&	Ampersand or and sign
'	Single quotation mark (apostrophe)
?	Question mark
<	Less than
>	Greater than
@	At sign
"	Quotation mark
<BS>	Back Space key - deletes the last character typed
<ESC>	Escape key
<BREAK>	Break key
<CR>	Carriage Return keys (marked "Retrn" and "Enter")
<LINE_FEED>	Line feed key (Ctrl-L)

CONTROL CHARACTERS

UNIBASIC supports the following control characters:

CONTROL CHARACTER	ACTION
<break>	Interrupts program execution and returns to UNIBASIC command level when at an INPUT statement.
CTRL-C	Interrupts program execution and returns to CP/M command levelwhen at an INPUT statement.
CTRL-G	Rings the bell at the terminal.
CTRL-H	Backspaces. Deletes the last character typed.
CTRL-I	Tabs to the next tab stop. Tab stops are set every eight columns.
CTRL-L	Line Feed. Moves the cursor down one line.
CTRL-M	Carriage Return. Moves the cursor to the left side of the screen
CTRL-S	Suspends output to the CRT.
CTRL-Q	Resumes output to the CRT after a CTRL-S.

CONSTANTS

Constants are the values UNIBASIC uses during execution. There are two types of constants: string and numeric.

A string constant is a sequence of up to 255 alphanumeric characters enclosed in quotation marks(").

Examples

```
"HELLO"
"$25,000.00"
"Number of Employees"
```

ARRAY VARIABLES

An array is a group or table of values referenced by the same variable name. Each element in an array is referenced by an array variable that is subscripted with an integer or an integer expression. An array variable name has many subscripts as there are dimensions in the array. For example, V(10) would reference a value in a one-dimension array, T(1,4) would reference a value in a two-dimension array, and so on. The maximum number of elements per dimension is 32,767. The maximum number of dimensions is 88.

SPACE REQUIREMENTS

All UNIBASIC variables and arrays have a data header. The data headers are located in the UNIBASIC's data area. The data area is located between the location of the interpreter in memory and the location of the interpreter's stack in memory. The interpreter's stack is located just below the CP/M kernal in memory. The CP/M kernal is located in the top approximately 8100 (hex) of RAM. The spaces that are occupied by the interpreter, by the data area, and by the interpreter's stack are allocated dynamically. The data headers are shown below for each data type:

INTEGER

```
+-+-+-+-+
|POINTER|  a 4 byte pointer to the next data header
+-+-+-+-+-+-+-+-+
| VARIABLE NAME |  an 8 byte ASCII string
+-+-+-+-+-+-+-+-+
|D-T|  the 2 byte long data type value
+-+-+-+-+-+-+
|VAL|UNUSED |  the 2 byte integer value and 4 unused bytes
+-+-+-+-+-+-+
```

REAL

```
+-+-+-+-+
|POINTER|  a 4 byte pointer to the next data header
+-+-+-+-+-+-+-+-+
| VARIABLE NAME |  an 8 byte ASCII string
+-+-+-+-+-+-+-+-+
|D-T|  the 2 byte long data type value
+-+-+-+-+-+-+
| VALUE |   |  the 4 byte real value
+-+-+-+-+-+-+
```

STRING

```
+-+-+-+-+
|POINTER|  a 4 byte pointer to the next data header
+-+-+-+-+-+-+-+-+
| VARIABLE NAME |  an 8 byte ASCII string
+-+-+-+-+-+-+-+-+
|D-T|  the 2 byte long data type value
+-+-+-+-+-+-+
|LEN|EL-PNTR|  2 byte string   and   the string element
+-+-+-+-+-+-+  length pointer        pointer
```

ARRAY

```
+-+-+-+-+
|POINTER|  a 4 byte pointer to the next data header
+-+-+-+-+-+-+-+-+
| VARIABLE NAME |  an 8 byte ASCII string
+-+-+-+-+-+-+-+-+
|D-T|  the 2 byte long data type value
+-+-+-+-+-+-+
|AR-PNTR|   |  a 4 byte pointer to the data
+-+-+-+-+-+-+
```

The string element has the following layout in memory:

STRING ELEMENT

```
+-+-+-+- -+-+-+-+-+
|C| STRING DATA |Ø|  1 byte link   then   the bytes of    then   a null
+-+-+-+- -+-+-+-+-+  count value          string data            byte
```

FUNCTIONAL OPERATORS

A function is used in an expression to call a predetermined operation that is to be preformed on an operand. UNIBASIC has "intrinsic" functions that reside in the system, such as SQR (square root) or SIN (sine). All UNIBASIC intrinsic functions are described in Chapter 3.

UNIBASIC also allows "user-defined" functions that are written by the programmer. See "DEF FN" in a later section.

STRING OPERATORS

Strings may be concatenated by using +.

Example

```
10 A$ = "FILE": B$ = "NAME"
20 PRINT A$+B$
30 PRINT "NEW"+A$+B$
RUN
FILENAME
NEWFILENAME
```

Strings may be compared using the same relational operators that are used with numbers:

```
=    <>    <    >    <=    >=
```

String comparisons are made by taking one character at a time from each string and comparing the ASCII codes. If all the ASCII codes are the same, the strings are equal. If the ASCII codes differ, the lower code number precedes the higher. If during string comparison, the end of one string is reached, the shorter string is said to be smaller. Leading and trailing blanks ARE significant.

Examples

```
"AA"<"AB"
"FILENAME"="FILENAME"
"X&">"X#"
"CL">"CL"
"kg">"KG"
"SMYTH"<"SMYTHE"
B$<"9/12/78" where
B$="8/12/78"
```

Thus, string comparisons can be used to test string values or to alphabetize strings. All string constants used in comparison expressions must be enclosed in quotation (") marks.

HIGH RESOLUTION GRAPHICS

There are two pages of high resolution graphics. The user selects the page desired by issuing either a PAGE#1 command or a PAGE#2 command.

SHAPE TABLE

The shape table begins at address 4000 decimal. The shape table has a default size of 500 bytes. The shape table size can be changed by using the SHSIZE command. The shape table is loaded either from a disk file by using the SHLOAD command or by POKEing the values in starting at address 4000 decimal. The shape table can be saved into a disk file by using the SHSAVE command.

INPUT EDITING

If an incorrect character is entered as a line is being typed, it can be deleted with the <Back Space> (<BS>) key or with CONTROL-H. Both the <BS> key and CONTROL-H have the effect of backspacing over a character and erasing it. Once a character(s) has been deleted, simply continue typing the line as desired.

To delete a line that is in the process of being typed, type CONTROL-U. A carriage return is executed automatically after the line is deleted.

To correct program lines for a program that is currently in memory, simply retype the line using the same line number. UNIBASIC will automatically replace the old line with the new line.

More sophisticated editing capabilities are provided. See the alternate arrow commands in a later section.

To delete the entire program currently residing in memory, enter the NEW command. (See the "NEW" command in a later section.) NEW is usually used to clear memory prior to entering a new program.

ERROR MESSAGES

If an error causes program execution to terminate, an error message is printed. For a complete list of UNIBASIC error codes and error messages, see the APPENDICES.

EDITING - alt arrow

Syntax

alt →
alt ←
alt ↓
alt ↑

Description

These commands do not affect characters moved over by the cursor: the characters remain both on the CRT screen and in memory. By themselves, these commands do not affect the program line being typed.

alt →	moves the cursor one space to the right
alt ←	moves the cursor one space to the left
alt ↓	moves the cursor one space down
alt ↑	moves the cursor one space up

Parameters

None.

Notes

- To change a program line: LIST the line on the CRT screen and use the alt. arrow commands to place the cursor over the first character of the line. Use the right-arrow key to move across the line, stopping at characters you wish to change and entering the desired character. When you are finished changing the line, press RETURN to store or execute the corrected line. If you did not use LIST to display the line, do not copy the prompt character (:).

- The alt arrow commands may be used in the immediate execution mode only.

Error Messages

Examples

Caveat

<BREAK> key

Syntax

break

Description

break interrupts the current process immediately after the statement that is currently being executed.

Parameters

None.

Notes

- break may be entered to interrupt an INPUT or GET but must be the first character entered. The interruption occurs when return is pressed for INPUT and immediately for GET.

- BREAK IN line-number is displayed a program is executing.

- break may be used in the deferred execution mode only.

Error Messages

Examples

Caveat

CATALOG

Syntax

CATALOG[,Dn]

Description

This command causes a list of the contents of the directory of the disk drive specified to be displayed on the screen.

Parameters

n is the number of the disk drive that the directory is to be displayed for. The following is a correlation of disk drive numbers and CP/M-68K drive specifiers:

DRIVE NUMBER	CP/M-68K DRIVE
1	A:
2	B:
3	C:
4	D:

If no disk drive is specified, then the disk drive that was most recently accessed will be used.

Notes

Error Messages

Examples

Caveat

COLOR

Syntax

COLOR = arithexpr

Description

Sets the color for plotting in low resolution graphics mode.

Parameters

The range of values for arithexpr is from Ø through 255.

Color numbers and their associated names are:

Ø	black	4	dark green	8	dark aqua	12	green
1	dark blue	5	grey	9	bright blue	13	aqua
2	red	6	orange	1Ø	grey	14	yellow
3	magenta	7	pink	11	light blue	15	white

COLOR evaluates arithexpr modulo 16 to return a value in the range of Ø to 15.

Notes

- In high-resolution graphics mode COLOR has no meaning.
- See SCRN and PLOT for more information.

Error Messages

Examples

Caveat

When used in TEXT mode with PLOT, COLOR will affect which character the PLOT instruction places in the text window.

HCOLOR

Syntax

```
HCOLOR = arithmetic-expression
```

Description

This command sets the high-resolution GRaphics color to that specified by the value of arithmetic-expression.

Parameters

The range of values for arithmetic-expression is from Ø through 15 if COLOR is ON. And, a range of Ø to 1 if COLOR is OFF.

The colors are as follows:

COLOR = ON	COLOR = OFF
Ø = Black	Ø = Black
1 = Dark Blue	1 = White
2 = Red	
3 = Magenta	
4 = Dark Green	
5 = Grey	
6 = Orange	
7 = Pink	
8 = Dark Aqua	
9 = Bright Blue	
1Ø = Grey	
11 = Light Blue	
12 = Green	
13 = Aqua	
14 = Yellow	
15 = White	

Notes

- In the low-resolution graphics mode, HCOLOR has no meaning.

Error Messages

Examples

Caveat

HGR

Syntax

HGR

Description

Sets the screen for High-resolution GRaphics mode. The resolution depends on the MODE# command.

Displays the bottom N lines of the text window below the graphics.

Clears the screen to black and displays page 1 of memory.

Parameters

None.

Notes

- HCOLOR is not changed.
- Text screen memory is not affected.
- Leaves the text "window" at full screen, but only the bottom N text lines are visible below the graphics. The cursor will still be in the text "window", but may not be visible unless moved to one of the bottom N lines.
- See MODE#, PAGE#, GR, HGR2, TEXT, COLOR, and HCOLOR for more information.

Error Messages

?SYNTAX ERROR
If the reserved word HGR appears as the first three characters of a variable name.

Examples

Caveat

If the reserved word HGR is used as the first characters of a variable name, the HGR may be executed before the

```
?SYNTAX ERROR
```

appears. Executing the statement

```
HGRIP=4
```

sets the high-resolution graphics mode.

HOME

Syntax

HOME

Description

HOME moves the cursor to the upper left screen position within the scrolling window and clears all text within the window.

Parameters

None.

Notes

- HOME may be used in either the immediate or deferred execution mode.

Error Messages

Examples

Caveat

HPLOT

Syntax

```
HPLOT X1, Y1
HPLOT TO X2, Y2
HPLOT X1, Y1 TO X2, Y2 [TO Xm, Ym ... TO Xn, Yn]
```

Description

This command will draw a high-resolution dot or line. If only X1 and Y1 are specified, then a dot will be drawn. If only X2 and Y2 are specified, then a line will be drawn from the last point plotted to the co-ordinates specified. If both the X1, Y1 and X2, Y2 co-ordinates are specified, then a line will be plotted from the X1, Y1 co-ordinates to the X2, Y2 co-ordinates.

Notes

Error Messages

Examples

Caveat

LOAD

Syntax

LOAD filename

Description

This command causes UNIBASIC to attempt to "load" into memory the filename specified from disk as a UNIBASIC program.

Parameters

Filename is the name of a disk file in the form specified in the UNIBASIC USER'S GUIDE.

Notes

Error Messages

Examples

Caveat

MODE#

Syntax MODE# modenumber

Description This command selects various graphics and text screen options, based on the value of modenumber. This statement is executed instead of PEEKing and POKEing. This statement must be immediately followed by either a TEXT command, an HGR command, or by an HGR2 command.

Parameters Modenumber has the following options and values:

Mode# Option

0 Initializes video to 80 columns by 24 lines

MODE#0

1 Reset the ERROR FLAG to OFF

MODE#1

Note: If the ERROR FLAG is ON, then when an attempt is made to plot a point outside of the screen window, an ?OUT OF RANGE ERROR message is given and execution is terminated.

2 Set the ERROR FLAG to ON

MODE#2

3 Set COLOR to OFF

MODE#3

4 Set COLOR to ON

MODE#4

5 Mixed Graphics and Text
for TEXT of 40 columns by 24 lines
or
for GRAPHICS of 320 x 240 pixels

MODE#5

MODE# (cont'd)

6 Mixed Graphics and Text
for TEXT of 40 columns by 48 lines
or
for GRAPHICS of 320 x 480 pixels

MODE#6

7 Mixed Graphics and Text
for TEXT of 80 columns by 24 lines
or
for GRAPHICS of 640 x 240 pixels

MODE#7

8 Mixed Graphics and Text
for TEXT of 80 columns by 48 lines
or
for GRAPHICS of 640 x 480 pixels

MODE#8

9 INTERNAL USE ONLY

1xx Mixed Graphics and Text

the graphics are as chosen on the preselected page with xx lines of text on the preselected MIXED page
where xx = the number of lines of text in the range of from 0 to the maximum number of lines on the MIXED page.

Example

MODE#100
sets 0 lines of text, all graphics

MODE# 106
sets 6 lines of text, rest of screen = graphics

MODE# (cont'd)

Notes	See GR, HGR, HGR2, and TEXT.
Error Messages	
Caveat	

PLOT

Syntax

PLOT x, y

Description

In low resolution graphics mode, this command places a dot on the screen at screen location (x, y).

Parameters

X and y must be arithmetic expressions.

X must be in the range of Ø to 79.

Y must be in the range of Ø to 47.

Notes

The origin (Ø,Ø) is the upper left corner of the screen.

The most recently executed COLOR statement determines the color of the dot.

PLOT has no visible effect when used in HGR2 mode. This is true even if GR precedes PLOT, because the screen is not "looking at" the low resolution graphics page (page one) of memory.

See the GR and the TEXT commands.

Error Messages

?ILLEGAL QUANTITY ERROR
If the arithmetic expression for X is not in the range of Ø to 79, or if the arithmetic expression for Y is not in the range of Ø to 47.

Examples

PLOT Ø,Ø

places a dot in the upper left corner of the screen.

Caveat

Attempting to PLOT to a TEXT window results in a character being placed where the dot would have appeared. (A character occupies the space of 2 low resolution graphics characters stacked vertically.)

POKE

Syntax POKE address, arithexpr

Description This command places the eight bit value of arithexpr in the location address.

Parameters Address is a 32 bit real number.

Arithexpr is an arithmetic expression whose value is in the range of Ø to 255.

Notes

Error Messages

Examples

Caveat

NF#

Syntax

NF#

Description

This function, NF# (No File) returns information on the most recently opened file, as follows:

NF# = Ø	the file already existed.
NF# = 1	the file did not exist and was created.

Parameters

Notes

See the commands for DISK OPERATIONS in a previous section.

Error Messages

Example

```
1Ø IF NF# = 1 GOTO 4Ø
```

or the equivalent

```
1Ø IF NF# GOTO 4Ø
```

If the file was created, then the program branches to line 4Ø.

Caveat

PDL

Syntax

PDL (paddlenumber)

Description

This function returns the current value of the game control (paddle) specified by paddlenumber.

Parameters

The arithmetic-expression paddlenumber must be in the range of Ø to 3.

Notes

See the appendices for information about other game switches.

Error Messages

?ILLEGAL QUANTITY ERROR
If paddlenumber is less than Ø or greater than 3.

Examples

```
1Ø PPOS = PDL(1)
```

Sets the variable PPOS to the value of game control number one.

Caveat