

4051 MAXIPACK Extended FONTS Instructions

4051 EXTENDED FONTS are 10x16 dot matrix fonts. The 4051 BASIC dot matrix text font is 5x7, which is why the EXTENDED FONTS print slower. The EXTENDED FONTS uses the 10-bit X and Y vector DACs - so these fit horizontally and vertically in the same 72 characters per line as the BASIC 5x7 fonts.

The EXTENDED FONTS characters are drawn in a vertical raster (column dots) inner order. As a result, the memory bitmaps are rotated by -90 degrees in a 10x16 format, with 16 being the vertical and 10 being the horizontal axis. The horizontal character pitch is 14 GDUs (Graphic Display Units) and the vertical pitch for each line of characters is 22 GDUs.

EXTENDED FONTS - BASIC CALLS:

NOFONT - 4051 BASIC 5x7 dot matrix text font

FONT00 - 10x16 standard text font

FONT01 - Superscript and Subscript 10x16 font

FONT02 - Greek 10x16 font

FONT03 - Superscript and Subscript Greek 10x16 font

FONT04 - Graphic Symbol 10x16 font

@FONTN - Selectable 10x16 text font in PRINT statement with each "@n" in the text string changing to FONT0n where n=0 to 4. NOTE: you cannot display "@" in this font mode

CALL "FONT00" through "FONT04" can be individually called with no parameters. All subsequent PRINT commands will use that font until another CALL "FONT0n" or CALL "NOFONT" command.

CALL "@FONTN" allows you to switch between the fonts dynamically, within the text in a PRINT statement. The fonts are enumerated 0 through 4, so after that call, adding @0 in a PRINT string selects FONT00 until you type @1 when it switches to FONT01, etc. This means that you have to give up typing "@" in any text, since it is used to select between the various fonts.

CALL "@FONTN" treats the first character printed on the display after the CALL as though an AT SIGN ("@") had already been detected. As a result, a CALL "@FONTN" followed by a "0" will select font 0. This is true for anything writing a number to the display immediately after the CALL "@FONTN". This may be unintended behavior for some. The workaround is to print any character except a 0, 1, 2, 3, or 4, immediately after the CALL, unless you actually want one of those fonts immediately. After this startup sequence has passed, the "@X" sequence, will always select font X, as this is the intended behavior.

Any invalid font number or character used after the AT SIGN ("@") when CALL "@FONTN" is active, is ignored and the system starts looking for the next AT SIGN ("@"). Neither the AT SIGN ("@") nor the immediately following character are displayed.

While the EXTENDED FONTS ROMPACK has support for write-thru (non-store) characters, there is no practical way to use it without using assembly code. FAST GRAPHICS has CALLs that support write-thru characters (but not user defined) and user defined graphics points and vectors.

CALL "NOFONT" or else power cycling the 4051 returns you to the standard 4051 BASIC 5x7 font.

The EXTENDED FONTS ROMPACK overrides the EXTENDED BASIC auto indenting when LISTing a program, since they use the same (and only) character display vector. The BASIC INIT command disables EXTENDED FONTS and restores the EXTENDED BASIC auto indenting.

```

MAXIPACK/MAXIROM Extended FONT ROM TEST

NOFONT
@ABCD EFGHIJKLMNOPQRSTUVWXYZ[\]^_ !"#$%&'()*+,-./0123456789:;<=>?@ABCDEFGH
IJKLMNOPQRSTUVWXYZ[\]^_`abcdefghijklmnopqrstuvwxyz{|}~↓

FONT00
@ABCD EFGHIJKLMNOPQRSTUVWXYZ[\]^_ !"#$%&'()*+,-./0123456789:;<=>?@ABCDEFGH
IJKLMNOPQRSTUVWXYZ[\]^_`abcdefghijklmnopqrstuvwxyz{|}~■

FONT01
@ABCD EFGHIJKLMNOPQRSTUVWXYZ[\]^_ 123456789*+ ,-. / 0123456789*+ ,-. / 0 ABCDEFG
H IJKLMNOPQRSTUVWXYZ ( ) ↑ - 0 ABCDEFGHIJKLMNOPQRSTUVWXYZ ( ) ↑ - _

FONT02
@ABCD EFGHIJKLMNOPQRSTUVWXYZ[\]^_ 123456789*+ ,-. / 0123456789*+ ,-. / 0 ABCDEFG
H IJKLMNOPQRSTUVWXYZ ( ) ↑ - 0 ABCDEFGHIJKLMNOPQRSTUVWXYZ ( ) ↑ - _

FONT03
@ABCD EFGHIJKLMNOPQRSTUVWXYZ[\]^_ 123456789*+ ,-. / 0123456789*+ ,-. / 0 ABCDEFG
H IJKLMNOPQRSTUVWXYZ ( ) ↑ - 0 ABCDEFGHIJKLMNOPQRSTUVWXYZ ( ) ↑ - _

FONT04
@ABCD EFGHIJKLMNOPQRSTUVWXYZ[\]^_ 123456789*+ ,-. / 0123456789*+ ,-. / 0 ABCDEFG
H IJKLMNOPQRSTUVWXYZ ( ) ↑ - 0 ABCDEFGHIJKLMNOPQRSTUVWXYZ ( ) ↑ - _

Multiple FONTS in PRINT statement using CALL "@FONTN":
H2O

E=mc2

32+42=52

√196=14

```

MAXIPACK Extended FONTs

MAXIPACK/MAXIROM Extended FONT ROM TEST

NOFONT

@ABCDEFGHIJKLMNPQRSTUVWXYZ[\]^_!"#\$%&'()*+,-./0123456789:;<=>?@ABCDEFGH
IJKLMNOPQRSTUVWXYZ[\]^_`abcdefghijklmnopqrstuvwxyz{|}~^

FONT00

@ABCDEFGHIJKLMNPQRSTUVWXYZ[\]^_!"#\$%&'()*+,-./0123456789:;<=>?@ABCDEFGH
IJKLMNOPQRSTUVWXYZ[\]^_`abcdefghijklmnopqrstuvwxyz{|}~■

FONT01

ABCDEFGHIJKLMNPQRSTUVWXYZ[\]^_ 123456789*+,-./0123456789*+,-./0 ABCDEFG
HIJKLMNOPQRSTUVWXYZ () ^ _ 0 ABCDEFGHIJKLMNOPQRSTUVWXYZ ^ _

FONT02

ABCD EFGHIJKLMNOPQRSTUVWXYZ[\]^_ 123456789*+,-./0123456789*+,-./0 ABCDEFG
HIJKLMNOPQRSTUVWXYZ[\]^_`abcdefghijklmnopqrstuvwxyz{|}~^

FONT03

ABCDEFGHIJKLMNPQRSTUVWXYZ[\]^_ 123456789*+,-./0123456789*+,-./0 ABCDEFG
HIJKLMNOPQRSTUVWXYZ () ^ _ 0 ABCDEFGHIJKLMNOPQRSTUVWXYZ ^ _

FONT04

ABCDEFGHIJKLMNPQRSTUVWXYZ[\]^_ 123456789*+,-./0123456789*+,-./0 ABCDEFG
HIJKLMNOPQRSTUVWXYZ[\]^_`abcdefghijklmnopqrstuvwxyz{|}~^

Multiple FONTs in PRINT statement using CALL "@FONTN":

H₂O

E=mc²

3²+4²=5²

√196=14

Example 4051 EXTENDED ROM test program using 4051 MAXIPACK:

```

CALL "TWOCOL"
CALL "FONT00"
LIST
100 INIT
110 DIM A$(128)
120 A$=""
130 FOR I=0 TO 127
140 B$=CHR(I)
150 A$=A$&B$
160 NEXT I
170 PRINT "LMAXIPACK/MAXIROM Extended
d FONT ROM TEST"
180 FOR F=1 TO 7
190 CALL "NOFONT"
200 GO TO F OF 210,230,260,290,320,3
50,380
210 PRINT " _NOFONT"
220 GO TO 450
230 PRINT " _FONT00"
240 CALL "F0NT00"
250 GO TO 450
260 PRINT " _FONT01"
270 CALL "F0NT01"
280 GO TO 450
290 PRINT " _FONT02"
300 CALL "F0NT02"
310 GO TO 450
320 PRINT " _FONT03"
330 CALL "F0NT03"
340 GO TO 450
350 PRINT " _FONT04"
360 CALL "F0NT04"
370 GO TO 450
380 PRINT " _Multiple FONTs in PRINT
statement using CALL ""@FONTN"":"
390 CALL "@FONTN"
400 PRINT "0H@12@00 "
410 PRINT "E=mc@1""@0_ "
420 PRINT "@03@1""@0+4@1""@0=5@1""
430 PRINT "@0 ----@2v@0196=14"
440 GO TO 470
450 CALL "PRINTS",A$
460 PRINT " -
470 NEXT F
480 CALL "NOFONT"
490 END

```

CALL "TWOCOL" is a 4051 MAXIROM Enhanced BASIC CALL to print in two columns.

CALL "FONT00" is a 4051 EXTENDED FONTS CALL to change from BASIC 5x7 text characters to EXTENDED FONTS 10x16 text characters.

The left column of the EXTENDED FONTS test program prints all 128 characters of FONT00 thru FONT04 using MAXIROM/MAXIPACK Fast Graphics CALL "PRINTS" in line 440 to print every character in A\$ including printing control characters with underline.

First Multiple FONT example for H subscript 2 O (formula for water molecule)

- Line 390 enables EXTENDED FONTS "@FONTN"
- The first zero in line 400 PRINT statement selects FONT00, H is printed, @1 selects FONT01, 2 prints a subscript 2, @0 selects FONT00 and then the O is printed.

Second example for E=mc squared

- We are already in FONT00 mode so line 410 begins by printing E=mc, then @1 selects FONT01 and the next two quotes print superscript 2. *TIP: Use FONT00 characters in @FONTN PRINT statements. Find the corresponding letter in FONT00 for the particular letter or symbol in FONT01 to FONT04.*
- For this example, the FONT01 superscript 2 is directly below the quote character in FONT00 - but to PRINT a quote character BASIC requires two quotes in the PRINT string.

Third example for 3 squared + 4 squared = 5 squared

- Line 420 @0 selects FONT00, 3 is printed, @1 selects FONT01, "" prints superscript 2, @0 selects FONT00, +4 is printed, @1 selects FONT01, "" prints superscript 2, @0 selects FONT00, =5 is printed, @1 selects FONT01 and "" prints superscript 2.

Fourth example for square root of 196

- Line 430 @0 selects FONT00, space character and three underscores are printed then CTRL-Underscore does CR/LF to line below, @2 selects FONT02, v is the square root character, @0 selects FONT00 and finally 196=14 is printed.