Hi-RES BASIC

Hi-RES BASIC was developed using the Grafyx Solution™ high resolution board from Micro-Labs, Inc.

Hi-RES BASIC

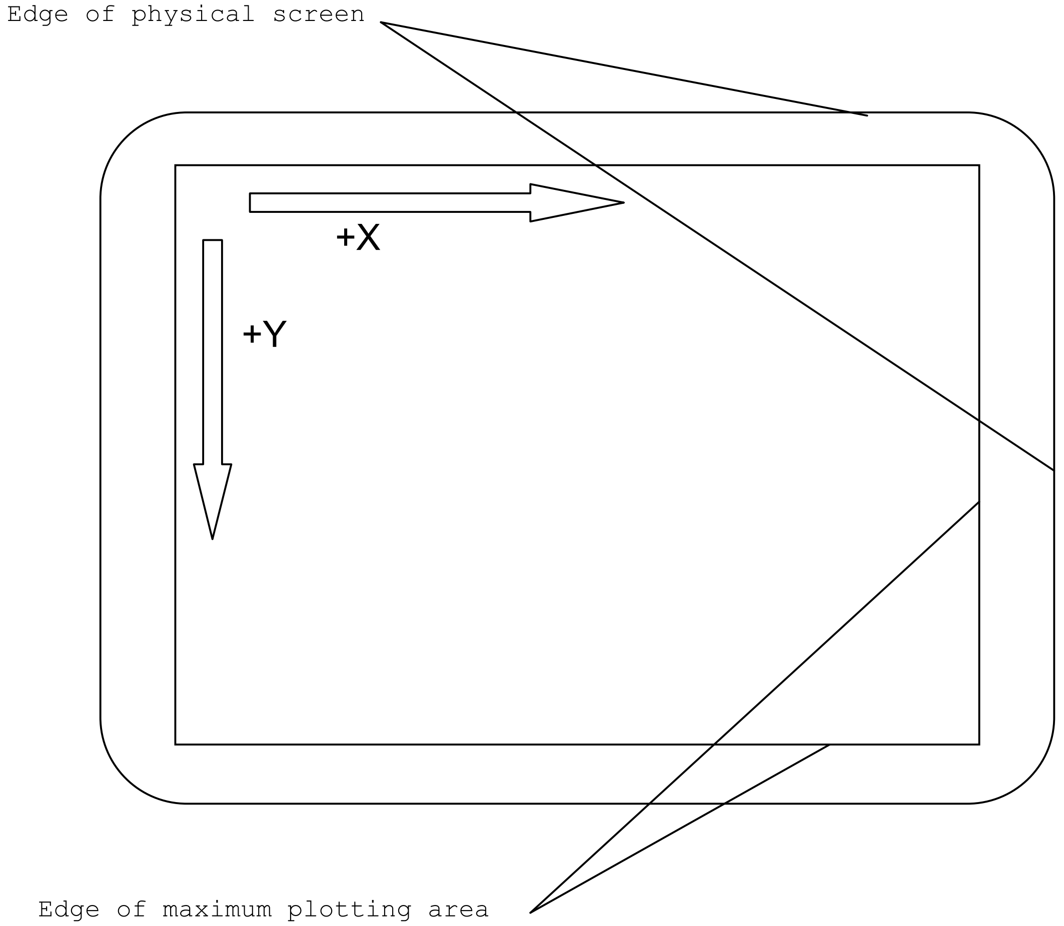
REFERENCE MANUAL

Hi—RES BASIC coordinate system

The high resolution screen has 153,600 dots arranged in 240 rows of 640 dots each, and is separate from the standard screen. The high resolution screen provides twice as much horizontal resolution than it does vertical; therefore, horizontal distances (as viewed) between dots are half the vertical distance (as viewed) between dots.

All points are referenced in the form (HORIZONTAL,VERTICAL) or (X,Y). The input range of the X and/or Y values are -32,768 through 32,767. However, Hi-RES BASIC only processes X values of 0 through 639, and Y values of 0 through 239 when set to the maximum resolution. Refer to the information on the next page for plotting values using different resolutions.

Typical screen:



The plotting area is the area of the physical screen in which Hi-RES BASIC is able to place a point (in which the computer is able to display information). This plotting area is 640 dots wide and 240 dots high when you enter Hi-RES BASIC. You may reduce the plotting area (define plotting area) and/or increase the number of dots per point (set resolution). The 640 dot horizontal locations are further divided into 80 eight-dot wide columns that are used by several Hi-RES BASIC statements: erase, fill, and reverse the high resolution screen.

HR(n)

Enable high resolution screen, set resolution, and maximize plotting area.

HR(exp)

exp = resolution code: an integer expression between 0 and 15

	Video 80x24		Video 64x16			
exp	Resolution	Plotting Values	Resolution	Plotting Values	Point Size[1]	Standard Screen Data Displayed
0	640 x 240	-1<x<640,	512 x 192	-1<x<512,	1x1	Yes
1	640 x 240	-1<y<240	512 x 192[2]	-1<y<192	1x1	No
2	320 x 240	-1<x<320,	256 x 192	-1<x<256,	2x1	Yes
3	320 x 240	-1<y<240	256 x 192[2]	-1<y<192	2x1	No
4	160 x 240	-1<x<160,	128 x 192	-1<x<128,	4x1	Yes
5	160 x 240	-1<y<240	128 x 192[2]	-1<y<192	4x1	No
6	80 x 240	-1<x<80,	64 x 192	-1<x<64,	8x1	Yes
7	80 x 240	-1<y<240	64 x 192[2]	-1<y<192	8x1	No
8	640 x 120	-1<x<640,	512 x 96	-1<x<512,	1x2	Yes
9	640 x 120	-1<y<120	512 x 96[2]	-1<y<96	1x2	No
10	320 x 120	-1<x<320,	256 x 96	-1<x<256,	2x2	Yes
11	320 x 120	-1<y<120	256 x 96[2]	-1<y<96	2x2	No
12	160 x 120	-1<x<160,	128 x 96	-1<x<128,	4x2	Yes
13	160 x 120	-1<y<120	128 x 96[2]	-1<y<96	4x2	No
14	80 x 120	-1<x<80,	64 x 96	-1<x<64,	8x2	Yes
15	80 x 120	-1<y<120	64 x 96[2]	-1<y<96	8x2	No

1. The point size is the number of dots wide by the number of dots high.
2. The plotting area is sized in a 64x16 subset of the 80x24 screen.

If exp is greater than 15, then exp is computed on a modulo 16 basis (values up to 255 do not produce an error). e.g., HR(23) is the same as HR(7).

If exp is evaluated as odd, then the standard screen data is not displayed (you don't see what you key); however, normal processing occurs for commands and/or statements.

EXAMPLE:

```
100 HR(9)
```

This enables the high resolution display for 640 x 120 resolution with standard screen data not displayed. Although the standard screen is disabled, any command you issue is executed as if the standard screen was enabled. A change in resolution does not affect data previously drawn onto the high resolution screen.

HR M

Return the value of the resolution code.

HR M

This statement requires no arguments or parameters. HR M returns the value of exp used in the last encountered HR(exp) statement.

EXAMPLE:

```
10 HR(2)
20 HR CLS
30 HR D(108,80)-(215,159)
40 PRINT HR M
```

The execution of this program segment prints 2.

HR X

Disable the high resolution screen.

HR X

This statement requires no arguments or parameters. HR X disables the high resolution screen and enables standard screen data. The resolution, the defined plotting area, and/or high resolution screen data are not changed as a result of the HR X statement.

HR X does not erase the high resolution screen.

To enable (to see what's there) the high resolution screen, execute a HR(exp) statement.

NOTE: When the screen saver (SSAVER/CMD) restores the Hi-RES screen, the screen saver uses the presentation mode determined by the last encountered HR(exp) statement.

HR CLS

Clear the high resolution screen.

HR CLS

This statement requires no arguments or parameters. HR CLS clears (erases) the high resolution screen without affecting standard screen data. The portion of the high resolution screen erased depends on the defined plotting area.

HR CLS clears the screen vertically to the defined plotting area. HR CLS clears the screen horizontally in eight-dot wide columns (regardless of the resolution). If the defined plotting area starts at the start of an eight-dot column border and ends at the end of an eight-dot column border, then the defined plotting area is totally erased.

The starts of eight-dot columns for 640 x 240 or 640 x 120 resolution are the X locations that when divided by eight leave no remainder. e.g., 0, 8, 16, 24, 32, etc.

The ends of eight-dot columns for 640 x 240 or 640 x 120 resolution are the X locations that when added to one and then divided by eight leave no remainder. e.g., 7, 15, 23, 31, etc.

If the defined plotting area does not end on eight-dot column borders, then HR CLS uses the next lower eight-dot column border. Explanation of example is covered under HR D (next page).

EXAMPLE:

```
10 HR (0)
20 HR D (60,0)-(140,40)
...
80 HR CLS
```

This erases dots from horizontal location 56 (yes, less than the defined plotting area) through horizontal location 135.

RESOLUTION: 320 x 240 or 320 x 120

The start of eight-dot columns are the X locations that when divided by four leave no remainder. e.g., 0, 4, 8, 12, 16, etc.

The end of eight-dot columns are the X locations that when added to one and then divided by four leave no remainder. e.g., 3, 7, 11, 15, 19, etc.

RESOLUTION: 160 x 240 or 160 x 120

The start of eight-dot columns are the X locations that when divided by two leave no remainder. e.g., 0, 2, 4, 6, 8, etc.

The end of eight-dot columns are the X locations that when added to one and then divided by two leave no remainder. e.g., 1, 3, 5, 7, 9, etc.

RESOLUTION: 80 x 240 or 80 x 120

The start and end of eight-dot columns are all X locations.

HR D

Define the plotting area.

HR D [(x1,y1)]-(x2,y2)

x1,y1 =the upper left corner of the plotting area. Defaults to previous x2,y2 values

x2,y2 =the lower right corner of the plotting area

x1 must be greater than -1 and less than the maximum horizontal resolution value
y1 must be greater than -1 and less than the maximum vertical resolution value
x2 must be greater than x1 and less than the maximum horizontal resolution value
y2 must be greater than y1 and less than the maximum vertical resolution value

if the resolution is set to 640 x 240, then:

```
-1 < x1 < 640
-1 < y1 < 240
x1 < x2 < 640
y1 < y2 < 240
```

if the resolution is set to 320 x 120, then:

```
-1 < x1 < 320
-1 < y1 < 120
x1 < x2 < 320
y1 < y2 < 120
```

When a plotting area is defined, points outside this area are not drawn. Depending on the X values used in defining the plotting area, points outside (only to the left) may be erased and/or reversed - refer to the HR CLS usage of eight-dot wide columns.

Any change of resolution using the HR(exp) statement resets the plotting area to the maximum plotting area and sets the resolution (for new points) to the new resolution code as determined by exp.

EXAMPLES:

```
100 HR (0) 'set resolution to 640 x 240
200 HR D (60,0)-(140,40) 'set an 81 x 41 plotting area starting at the top
    and 60 dots from the left of the maximum plotting area
```

EXAMPLE:

```
10 HR (0)
20 HR D (60,0)-(140,40)
...
80 HR CLS
```

This erases dots from horizontal location 56 (yes, less than the defined plotting area) through horizontal location 135.

HR DEF

Return a coordinate of the defined plotting area.

HR DEF(exp)

exp = integer expression between 0 and 3

if exp is evaluated as 0, then the upper left X coordinate value is returned
if exp is evaluated as 1, then the upper left Y coordinate value is returned
if exp is evaluated as 2, then the lower right X coordinate value is returned
if exp is evaluated as 3, then the lower right Y coordinate value is returned

If exp is greater than 3, then exp is computed on a modulo 4 basis (values up to 255 do not produce an error). e.g., HR DEF(31) is the same as HR DEF(3)

EXAMPLES:

```
100 HR C((HR DEF(2)+1)/2,(HR DEF(3)+1)/2),20
```

This code segment plots a circle in the center of the high resolution screen regardless of the resolution.

Use of HR DEF(exp) enables you to specify the use of defined plotting limits without having to hard code values in your programs. And, this is especially useful if you use many different resolutions in the same program.

```
10000 '"HRPATTRN/BAS
11000 CLS:RANDOM
12000 ON STOP GOTO "END"
13000 DEFINT X - Z
14000 HR (0):HR CLS
15000 LABEL "NEW PATTERN"
16000 HR (RND(15) AND 254)
17000 IF RND(150) = 75 THEN HR (0):HR R
18000 LABEL "NEW HORIZONTAL"
19000 Z = (HR DEF (2) + 1) / 2 * (RND(2) - 1)
20000 X1 = (RND((HR DEF (2) + 1) / 2) + Z) AND &X1111111111110000
21000 Z = (HR DEF (2) + 1) / 2 * (RND(2) - 1)
22000 X2 = ((RND((HR DEF (2) + 1) / 2) + Z) AND &X1111111111110000) - 1
23000 IF X1 => X2 THEN "NEW HORIZONTAL"
24000 LABEL "NEW VERTICAL"
25000 Z = (HR DEF (3) + 1) / 2 * (RND(2) - 1)
26000 Y1 = (RND((HR DEF (3) + 1) / 2) + Z) AND &X1111111111111000
27000 Z = (HR DEF (3) + 1) / 2 * (RND(2) - 1))
28000 Y2 = ((RND((HR DEF (3) + 1) / 2) + Z) AND &X1111111111111000) - 1
29000 IF Y1 => Y2 THEN "NEW VERTICAL"
30000 HR D(X1, Y1) - (X2, Y2)
31000 IF RND(2) -1 THEN HR CLS ELSE HR R
32000 GOTO "NEW PATTERN"
33000 LABEL "END"
34000 HR (0):HR X:ON STOP GOTO 0
```

This program produces random rectangular patterns on the high resolution screen.

HR R

Reverse the high resolution screen.

HR R

This statement requires no arguments or parameters. HR R simply reverses the high resolution screen. The portion of the high resolution screen reversed depends on the defined plotting area.

HR R reverses the screen vertically to the defined plotting area. HR R reverses the screen horizontally in eight-dot wide columns (regardless of the resolution). If the defined plotting area starts at the start of an eight-dot column border and ends at the end of an eight-dot column border, then the defined plotting area is totally reversed.

The starts of eight-dot columns for 640 x 240 or 640 x 120 resolution are the X locations that when divided by eight leave no remainder. e.g., 0, 8, 16, 24, 32, etc.

The ends of eight-dot columns for 640 x 240 or 640 x 120 resolution are the X locations that when added to one and then divided by eight leave no remainder. e.g., 7, 15, 23, 31, etc.

If the defined plotting area does not end on eight-dot column borders, then HR R uses the next lower eight-dot column border.

EXAMPLE:

```
10 HR (0)
20 HR D (60,0)-(140,40)
...
80 HR R
```

This reverses dots from horizontal location 56 (yes, less than the defined plotting area) through horizontal location 135.

RESOLUTION: 320 x 240 or 320 x 120

The start of eight-dot columns are the X locations that when divided by four leave no remainder. e.g., 0, 4, 8, 12, 16, etc.

The end of eight-dot columns are the X locations that when added to one and then divided by four leave no remainder. e.g., 3, 7, 11, 15, 19, etc.

RESOLUTION: 160 x 240 or 160 x 120

The start of eight-dot columns are the X locations that when divided by two leave no remainder. e.g., 0, 2, 4, 6, 8, etc.

The end of eight-dot columns are the X locations that when added to one and then divided by two leave no remainder. e.g., 1, 3, 5, 7, 9, etc.

RESOLUTION: 80 x 240 or 80 x 120

The start and end of eight-dot columns are all X locations.

HR P

Draw one point (plot a point).

HR P (x,y)[,c]

x,y = the location where the point is drawn
c = point color: values 0 through 2
a color of 0 resets the point
a color of 1 (default) sets the point
a color of 2 complements the point

EXAMPLES:

```
10 HR (0)
20 HR CLS
30 FOR X = 0 TO 639
40 HR P (X,120)
50 NEXT
```

This draws a series of connected points (a line) from the middle of the left side of the high resolution screen to the middle of the right side of the high resolution screen.

HR POINT

Return the status of a point.

HR POINT (x,y)

x,y = the point location whose status is to be returned

HR POINT returns zero if the point is not set, a one if the point is set, and a negative one if the point is not within the defined plotting area.

EXAMPLE:

```
10 HR(0)
20 HR CLS
30 PRINT HR POINT(-666,4092)
```

The execution of this program segment prints -1.

HR L

Draw a line.

HR L [(x1,y1)]-(x2,y2)[,c|l]

x1,y1 =one end of the line. Defaults to previous x2,y2 values
x2,y2 =the other end of the line
c = line drawing color: values 0 through 2
    a color of 0 resets the points as the line is drawn
    a color of 1 (default) sets the points as the line is drawn
    a color of 2 complements the points as the line is drawn
l = line drawing style: values 3 through 255. The style is the binary bit pattern of the value. i.e., a value of 23 decimal, 00010111 binary, starts with three spaces, sets one point, another spaces, then sets the next three points. This style is repeated until the end point is reached.

The line is drawn from (x1,y1) to (x2,y2).

EXAMPLES:

```
10 HR (0)
...
200 HR L (639,239)-(0,0)
```

This draws a line from the lower right corner of the screen to the upper left corner of the screen.

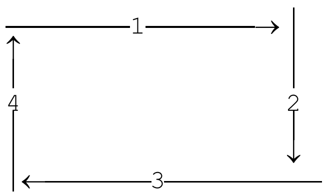
HR B

Draw a rectangle.

HR B [(x1,y1)]-(x2,y2)[,c|l]

x1,y1 =one corner of the rectangle. Defaults to previous x2,y2 values
x2,y2 =another corner of the rectangle that must be diagonal to the first corner
c = line drawing color: values 0 through 2
    a color of 0 resets the points as the rectangle is drawn
    a color of 1 (default) sets the points as the rectangle is drawn
    a color of 2 complements the points as the rectangle is drawn
l = line drawing style: values 3 through 255. The style is the binary bit pattern of the value. i.e., a value of 23 decimal, 00010111 binary, starts with three spaces, sets one point, another spaces, then sets the next three points. This style is repeated until the end point is reached.

Regardless of the arrangement of the diagonal corner values, the rectangle is drawn clockwise from the upper-left corner.



The corners of the rectangle do not overlap.

If x2 = x1 or y2 = y1, then HR B draws a line.
If x2 = x1 and y2 = y1, then HR B draws a point.

EXAMPLE:

```
10 HR(0)
20 HR CLS
30 HR B(400,120)-(100,30)
```

This draws a rectangle with four lines. The first line is drawn from 100,30 to 399,30; the second line is drawn from 400,30 to 400,119; the third line is drawn from 400,120 to 101,120; and the fourth line is drawn from 100,120 to 100,29. And, the x2 and y2 values are 100 and 30 respectively.

HR C

Draw a closed curve or arc functions.

HR C (x,y),r[,c][,s][,e][,a]

x,y = the center of the curve
r = half length of the major axis. For circles this is the radius in horizontal units. r must be greater than zero
c = drawing color: values 0 through 2
a color of 0 resets the points as the curve is drawn
a color of 1 (default) sets the points as the curve is drawn
a color of 2 complements the points as the curve is drawn (this is not allowed if s and/or e is specified.
s = the starting angle in degrees for an arc (the angle is anisotropic if the aspect ratio <> 1). Default is null. If e is specified then s defaults to zero.
e = the ending angle in degrees for an arc (the angle is anisotropic if the aspect ratio <> 1). Default is null. If s is specified then e defaults to zero.
a = aspect ratio. Default is one. The aspect ratio is the ratio of the width of the curve to the height of the curve, as viewed. i.e., an aspect ratio of 1 presents a circle.

Each of the optional parameters must be in specific positions. If you want to default a parameter and specify a parameter whose position is after the defaulted parameter, then you must include the comma to properly position the specified parameter. To specify an aspect ratio, defaulting c, s, and e, use the following format: HR C (x,y),r,,,,a

In the examples for curves and arcs, HR(0) and HR CLS are left out for simplicity.

EXAMPLES:

```
500 HR C(320,120),200
```

This draws a circle with the center at 320,120 and a radius of 200 points (measured in the horizontal direction).

```
100 HR C(320,120),50,,,,2
```

This draws an ellipse with the center at 320,120 and a ½ major axis of 50 points (measured in the horizontal direction). The width is twice the height (as viewed).

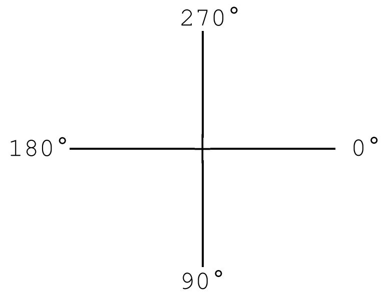
```
200 HR C(320,120),50,,,,.5
```

This draws an ellipse with the center at 320,120 and a ½ minor axis of 50 points (measured in the horizontal direction). The width is half the height (as viewed).

The horizontal point distance from the center of the curve (x,y) is always r regardless of the aspect ratio (a).

Arcs, sector lines, and chords

Angles in the Cartesian coordinate system as well as Hi-RES BASIC increase in the +X, +Y direction. Therefore, angles in Hi-RES BASIC are the inverse of the Cartesian coordinate system because in Hi-RES BASIC the +Y direction is down. In the Cartesian coordinate system 90° would be in the 12 O'clock position. i.e., angles increase in the counterclockwise direction. In Hi-RES BASIC, angles increase in the clockwise direction.




```
600 HR C(320,120),100,,30,180
```
This draws an arc from 30 degrees to 180 degrees.

```
600 HR C(320,120),100,,230,150
```
This draws an arc from 230 degrees to 150 degrees.

If the **s** and/or **e** parameters are evaluated as negative, then a sector line is drawn from the end of the arc to the center of the curve.

```
600 HR C(320,120),100,,-30,180
```
This draws an arc from 30 degrees, to 180 degrees, with a sector line from the 30° point on the curve to the center of the curve.

```
600 HR C(320,120),100,,30,-180
```
This draws an arc from 30 degrees to 180 degrees, with a sector line from the 180° point on the curve to the center of the curve.

```
600 HR C(320,120),100,,-30,-180
```
This draws an arc from 30 degrees, to 180 degrees, with a sector line from the 30° point on the curve to the center of the curve, and a sector line from the 180° point on the curve to the center of the curve.

If the **a** parameter is evaluated as negative and the **s** and **e** parameters are specified, then a chord is drawn between the ends of the arc.

```
600 HR C(320,120),100,,30,180,-1
```
This draws an arc from 30 degrees to 180 degrees, with a chord connecting the 30° point on the curve to the 180° point on the curve.

If the aspect ratio is not one, then the angle is anisotropic.

HR F

Fill an enclosed area.

HR F (x,y),f

x,y = the starting location of the fill pattern
f = fill pattern: 0 through 303

The fill patterns from 0 through 127 are the binary bit pattern with the high bit set, and is placed on every other row. The fill patterns from 128 through 255 are the binary bit pattern, and is placed on every row. The points set correspond to the set bit in the fill pattern. These fill patterns are aligned with the start of the eight-dot wide columns. i.e., if you specify a fill pattern of 32 ([1]0100000 binary), and the left edge of the area being filled has a horizontal location of 23, start of eight dot wide column+7, then the start (fills are left to right) of the fill pattern placed on the high resolution screen is the right most zero in this binary bit pattern. This produces a pattern of 0 10100000 10100000 etc. (the spaces are shown for clarity).

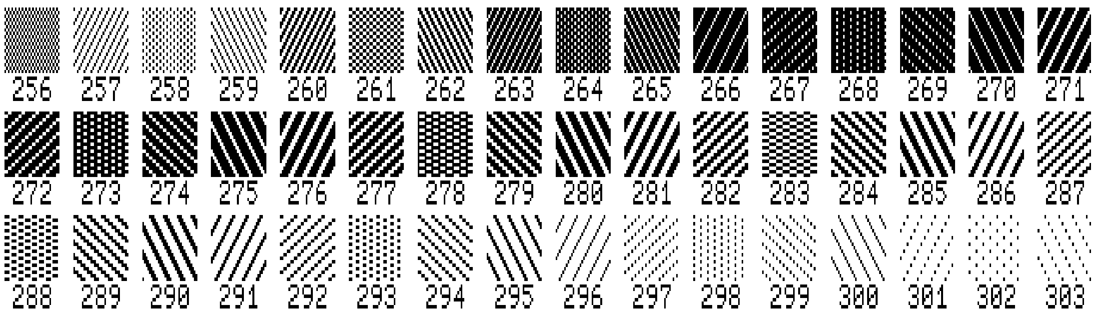
The fill patterns above 255 have horizontal alignment shifts to produce crisscross and diagonal patterns.

EXAMPLE:

```
10 HR (0)
...
80 HR C((HR DEF(2)+1)/2,(HR DEF(3)+1)/2),20 'circle centered on screen
90 HR F((HR DEF(2)+1)/2,(HR DEF(3)+1)/2),42 'fill circle with pattern 42
```

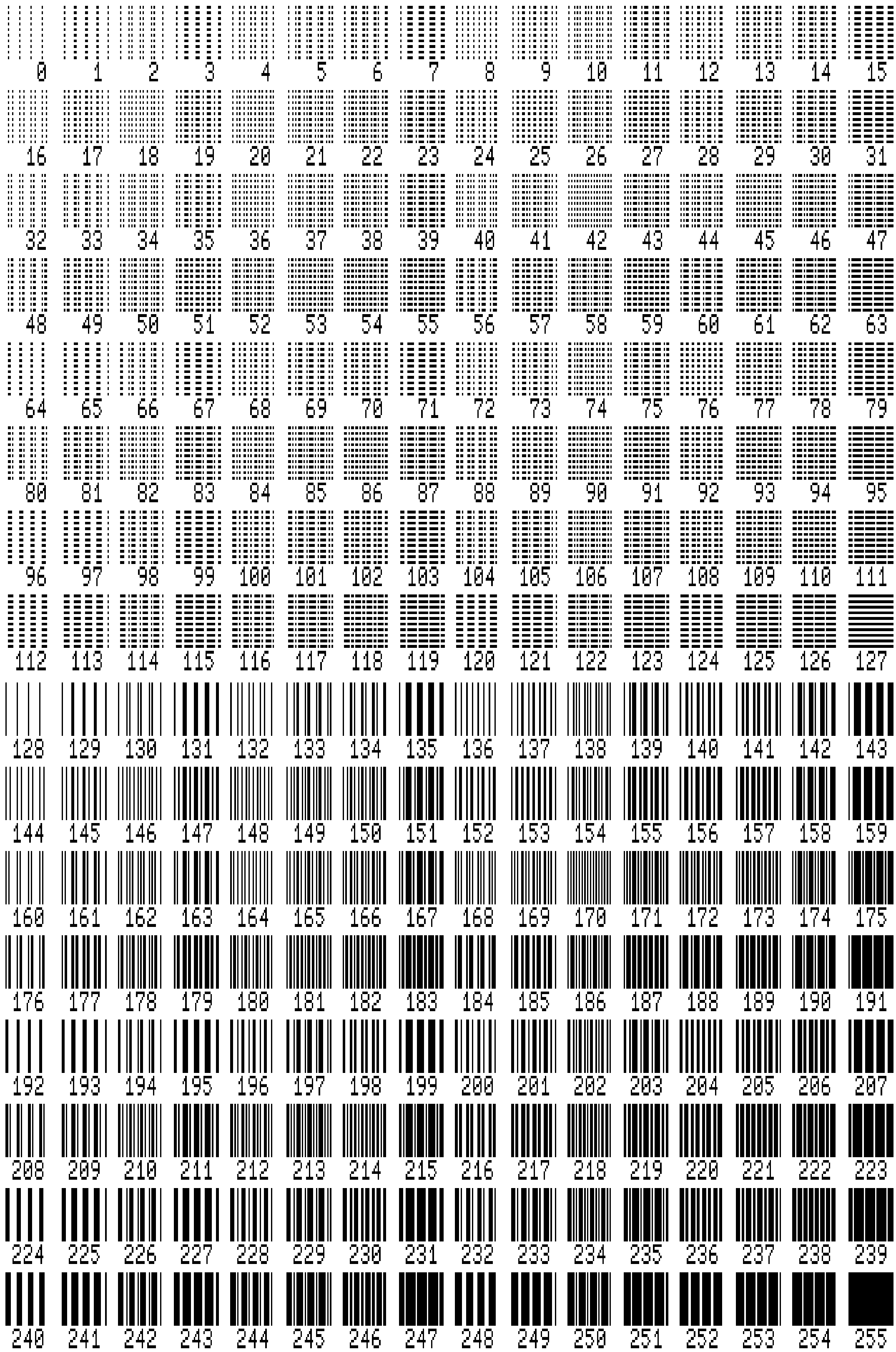
This plots a circle in the center of the high resolution screen then fills the circle with fill pattern 42.

Fill patterns from 256 through 303



HR F temporarily uses free memory to manage multiple fill paths. If the fill pattern is very complex, HR F possibly could cause an "Insufficient MEMORY" error to occur.

Fill patterns from 0 through 255 are on the following page.


0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15
16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31
32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47
48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63
64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79
80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95
96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111
112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127
128 129 130 131 132 133 134 135 136 137 138 139 140 141 142 143
144 145 146 147 148 149 150 151 152 153 154 155 156 157 158 159
160 161 162 163 164 165 166 167 168 169 170 171 172 173 174 175
176 177 178 179 180 181 182 183 184 185 186 187 188 189 190 191
192 193 194 195 196 197 198 199 200 201 202 203 204 205 206 207
208 209 210 211 212 213 214 215 216 217 218 219 220 221 222 223
224 225 226 227 228 229 230 231 232 233 234 235 236 237 238 239
240 241 242 243 244 245 246 247 248 249 250 251 252 253 254 255

HR G

Text, block graphics, and special characters on the high resolution screen.

HR G (x,y),o

x,y = the starting location for the upper left corner of the text, block graphic, or special character (this must be within the defined plotting area).

o = orientation for the text, block graphic, or special character. o must be evaluated in the range of 0 through 3

o	orientation
0	left to right
1	top to bottom
2	right to left (upside down)
3	bottom to top

Text (decimal 32 through 127), block graphics (decimal 128 through 191), and special characters (decimal 192 through 255) are eight points wide. Text and special characters are eight points high whereas blocks graphics are 10 points high if the video is in the 80x24 mode or 12 points high if the video is in the 64x16 mode.

The HR G statement redirects all subsequent printing to the high resolution screen. The high resolution screen video driver is simple and only recognizes two control characters, Carriage Return (decimal 13) and Backspace (decimal 24 [Model 4 MULTIDOS and ESOTERIC use 24 instead of 8]).

The HR G statement requires the starting location to be within the defined plotting area.

Printing that would extend beyond the defined plotting area (in the direction of the orientation) wraps around to the opposite edge of the defined plotting area onto the next line (relative to the orientation). If the starting location to the end of the defined plotting area is not a multiple of eight, then the text, block graphics, or special characters that are placed in the last position are clipped at the end of the defined plotting area.

Printing that would extend below (relative to the orientation) the defined plotting area wraps around to the top (relative to the orientation). If the starting location to the end of the defined plotting area is not a multiple of 10 (80x24 video) or 12 (64x16 video), then the text, block graphics, or special characters are clipped at the bottom (relative to the orientation) of the defined plotting area.

Text, block graphics, and special characters printed to the high resolution screen are XORed with other data on the high resolution screen. Therefore, when the printing of text, block graphics, or special characters onto the high resolution screen wraps, you may see some very peculiar things.

If you use the high resolution screen driver in the command mode, then the high resolution screen is erased when the printing of text, block graphics, or special characters cause a wrap to the top (relative to the orientation).

The high resolution text, block graphics, and special characters are in the overlay GRAF/BOL to maximize the available memory for Hi-RES BASIC. The first time a program encounters a HR G statement, the overlay is loaded.

HR N

Redirect the printing of text, block graphics, and special characters to the standard screen.

HR N

This statement requires no arguments or parameters. Execution of HR N disables the effect of a HR G statement.

NOTE: When the screen saver (SSAVER/CMD) restores the Hi-RES screen, the screen saver uses the presentation mode determined by the last encountered HR(exp) statement.

HR GET

Store high resolution screen data into a graphic array[1].

HR GET [(x1,y1)]-(x2,y2),var

x1,y1 =one corner of the rectangle. Defaults to previous x2,y2 values
x2,y2 =another corner of the rectangle that must be diagonal to the first corner. x2 cannot equal x1, and y2 cannot equal y1.
var = the name of the graphic array: standard variable name

x1, y1, x2, and y2 must be within the defined plotting area.

1. Graphic arrays were developed for Hi-RES BASIC. Graphic arrays are dynamically allocated (dimensioned), do not require parentheses, and the name does not conflict with scalar variable names and/or array variable names.

This statement copies the contents of a logical (not necessarily visible) rectangle into a graphic array, array object. The name of the array uniquely identifies the array object. If you use the same array name in a subsequent HR GET statement, then the array object of the previous HR GET statement is erased. BASIC operations that perform CLEAR or implied CLEAR erase the array object.

EXAMPLES:

```
10 HR (0)
...
200 HR GET (160,50)-(240,130),A
```

This copies the contents of a 81 x 81 rectangle into graphic array A starting at location (160,50).

```
10 HR (0)
...
200 HR GET (160,50)-(239,129),A
```

This copies the contents of a 80 x 80 rectangle into graphic array A starting at location (160,50).

```
10 HR (0) 'set resolution to 640 x 240 and plotting area to 640 x 240
20 R = 20
30 HR C((HR DEF(2)+1)/2,(HR DEF(3)+1)/2),R 'circle centered on screen
40 HR F((HR DEF(2)+1)/2,(HR DEF(3)+1)/2),42 'fill circle with pattern 42
50 HR GET ((HR DEF(2)+1)/2-R,(HR DEF(3)+1)/2-R/2)-((HR DEF(2)+1)/2+R,(HR
   DEF(3)+1)/2+R/2),A
```

This plots a circle in the center of the high resolution screen, fills the circle with fill pattern 42, then copies the circle into a 41 x 21 graphics array named A.

Last example hard coded.

```
10 HR (0)
20
30 HR C(320,120),20
40 HR F(320,120),42
50 HR GET (300,110)-(340,130),A
```

HR PUT

Copy an array object to the high resolution screen.

HR PUT (x,y),var,m

x,y = destination location for the array object (the upper left corner of the logical rectangle copied with a HR GET statement)

var = the name of the graphic array that must have been created with a HR GET statement.

m = mode used to copy the array object to the high resolution screen. m must be evaluated in the range of 0 through 4

The destination (x,y) must be within the defined plotting area. However, if the array object would extend beyond the defined plotting area, then the array object wraps around.

m	copy mode	copy mode description
0	logical NAND	NANDs array object with high resolution screen
1	direct	overlays high resolution screen with array object
2	logical AND	ANDs array object with high resolution screen
3	logical OR	ORs array object with high resolution screen
4	logical XOR	XORs array object with high resolution screen

Exotic logic: non-set points are indicated with 0, and set points are indicated with 1

m	array object point	high resolution screen point before copy	high resolution screen point after copy
0	0	0	1
	1	0	1
	0	1	1
	1	1	0
1	0	0	0
	1	0	1
	0	1	0
	1	1	1
2	0	0	0
	1	0	0
	0	1	0
	1	1	1
3	0	0	0
	1	0	1
	0	1	1
	1	1	1
4	0	0	0
	1	0	1
	0	1	1
	1	1	0

HR SAVE

Copy the high resolution screen to a disk file.

HR SAVE var$
or **HR SAVE** "filespec"

var$ = a string defined as a filespec.
filespec = a valid program file specification.

The entire plotting area of the high resolution screen, disregarding the resolution and/or the defined plotting area, is saved to filespec. The filespec size will be 75 sectors. The resolution and/or the defined plotting area are not changed as a result of the HR SAVE statement.

NOTE: If you have the video set to 80x24 with data on the high resolution screen, switch to the 64x16 video without clearing the 80x24 high resolution screen data, change the high resolution screen data, and execute a HR SAVE statement, then any residue left over from the 80x24 video format is also saved.

HR LOAD

Copy a disk file to the high resolution screen.

HR LOAD var$
or **HR LOAD** "filespec"

var$ = a string defined as a filespec.
filespec = a valid program file specification. The file size must be 19200 bytes long (75 sectors).

The filespec is copied to the high resolution screen overlaying any previous data on the high resolution screen. The data from the filespec disregards the resolution and/or the defined plotting area. However, the resolution and/or the defined plotting area are not changed as a result of the HR LOAD statement.

EXAMPLE:

```
10 HR(2)
20 HR CLS
30 HR D(108,80)-(215,159)
40 HR R
50 HR LOAD "HAT/HRG"
60 HR R
```

Line 10: Sets the resolution to 320 x 240
Line 20: Clears the entire high resolution screen
Line 30: Defines the plotting area to be approximately 1/9 the size of the plotting area in the center of the high resolution screen (108 points by 80 points - 216 dots x 80 dots)
Line 40: Reverses the defined plotting area (whites out)
Line 50: Loads the high resolution file HAT/HRG
Line 60: Reverses the defined plotting area - part of the data loaded with HAT/HRG

The HR LOAD statement does not care what the file name's extension is, and does not have a default extension. You must provide the complete filename.

HR PRINT

Print the high resolution screen using Hewlett-Packard's PCL 5 printer language.

HR PRINT (H,V),o[,d]

H = the starting horizontal location in PCL units (4 digits maximum)
V = the starting vertical location in PCL units (4 digits maximum)
o = orientation: integer expression between 0 and 1: 0 = portrait, 1 = landscape
d = resolution in dots per inch. d defaults to:
100 dpi if the orientation is portrait
75 dpi if the orientation is landscape

Values of H and/or V greater than four digits have the most significant digits clipped. i.e., a value of 31457 is interpreted as 1457.

If o is greater than 1, then o is computed on a modulo 2 basis (values up to 255 do not produce an error). i.e., HR PRINT(-1,-1),74 is the same as HR PRINT(-1,-1),0

Values of d not in the domain of 75, 100, 150, 200, 300, and 600 are mapped to the next lower value. Values between 0 and 75 are mapped to 75. Negative values are mapped to 600.

To make either the H and/or V parameters default to the present PCL "cursor" position, provide a negative argument for the parameter to default. e.g., HR PRINT (156,-1),0 prints the high resolution screen 156 PCL units from the left margin and the present vertical location.

NOTES: HR PRINT statements emulate a FORMS (OFF) without effecting a previous FORMS (C) command.

HR PRINT statements do not eject the page after printing the high resolution screen. This enables you to place multiple high resolution images on the same page and/or mix text and high resolution on the same page. The recommended code to eject a page after the high resolution screen in printed is:
LPRINT CHR$(27);"E";

The image printed is an exact replica of the entire 640 x 240 high resolution screen (presentation is exactly how it appears when executed on a standard TRS-80® microcomputer monitor - the output has a blank line between adjacent vertical data). The presentation disregards the resolution and/or the defined plotting area. HR PRINT does not output standard screen data. Obviously, HR PRINT prints text, block graphics, and special characters created with the HR G statement. The resolution and/or the defined plotting area are not changed as a result of a HR PRINT statement.

If you do not want the presentation to output blank horizontal lines, then use HR LPRINT. This has the effect of halving the vertical size (actually presenting the screen data with an aspect ratio of two). e.g., if you want to print a "perfect" circle, then use the following code segment:

```
HR C(X,Y),R,,,,0.5
HR LPRINT (H,V),o[,d]
```

Hi-RES BASIC

Using BASICH/CMD to interface simple machine language programs to the Hi-RES board.

The locations of specific entry points are available via a double peek. i.e., you have to peek a location to get the start of the table that point to specific routines in BASICH/CMD. The word contents in 409EH for MULTIDOS or 12E4H for ESOTERIC points to HVECS

Address	Points to	
HVECS+00	EX2	address of EX2
HVECS+02	WHY2	address of WHY2
HVECS+04	COLOR	address of color
HVECS+06	HCLS	clear Hi-RES screen: HR CLS
HVECS+08	REV	reverse Hi-RES screen: HR R
HVECS+10	TPLOT	perform point action: HR P(EX2, WHY2)
HVECS+12	NRMTX	set video to standard text: HR N
HVECS+14	SCOLR	set color or line style
HVECS+16	HRESO	return present resolution: HR M
HVECS+18	HPONT	return point status: HR POINT(EX2,WHY2)
HVECS+20	JDEF	set defined plotting area: HR D(EX1,WHY1)-(EX2,WHY2)
HVECS+22	EX1	address of EX1
HVECS+24	WHY1	address of WHY1
HVECS+26	JBOX	Draw a rectangle: HR B(EX1,WHY1)-(EX2,WHY2)
HVECS+28	JLINE	Draw a line: HR L(EX1,WHY1)-(EX2,WHY2)
HVECS+30	JTXT	set Hi-RES text orientation to value in A (accumulator)
HVECS+32	JRES	set resolution to value in Z80 E register
HVECC+34	PRESS	present configuration

EX2 HVECS+0

This is usually used with WHY2 forming a coordinate pair (EX2,WHY2). EX2 and WHY2 are valued with the last pointed acted upon (when BASICH is initialized, these values are zero). To place a value into EX2, your code should be similar to the following:

```
POKE WPEEK(WPEEK(&409E)+0), LSB value
POKE WPEEK(WPEEK(&409E)+0)+1, MSB value
```

To return the present value of EX2, your code should be similar to the following:

```
PRINT WPEEK(WPEEK(WPEEK(&409E)+0))
```

WHY2 HVECS+2

This is usually used with EX2 forming a coordinate pair (EX2,WHY2). EX2 and WHY2 are valued with the last pointed acted upon (when BASICH is initialized, these values are zero). To place a value into WHY2, your code should be similar to the following:

```
POKE WPEEK(WPEEK(&409E)+2), LSB value
POKE WPEEK(WPEEK(&409E)+2)+1, MSB value (usually zero)
```

To return the present value of WHY2, your code should be similar to the following:

```
PRINT WPEEK(WPEEK(WPEEK(&409E)+2))
```

COLOR HVECS+4
Set the color. Values not in the domain of 0, 1, and 2 are processed as a 2.

```
POKE, PEEK(WPEEK(&409E)+4), color
```

To determine the present color: PRINT PEEK(WPEEK(WPEEK(&409E)+4))

HCLS HVECS+6
Clear the Hi-RES screen.

```
CALL WPEEK(WPEEK(&409E)+6)
```

REV HVECS+8
Reverse the Hi-RES screen.

```
CALL WPEEK(WPEEK(&409E)+8)
```

TPLOT HVECS+10
Perform point action, if in range, at present resolution and color (the color can be set with either COLOR or SCOLR).

```
CALL WPEEK(WPEEK(&409E)+10)
```

NRMTX HVECS+12
Set the video to the standard screen driver.

```
CALL WPEEK(WPEEK(&409E)+12)
```

SCOLR HVECS+14
This is used to set the color or line style. This must be used for drawing lines and rectangles with the proper color or line style. To set the color or style, you must define a USR function and execute the USR function. Code segment example that sets the color or line style:

```
DEF USRn = WPEEK(WPEEK(&409E)+14)
X = USRn(color|style)
```

HRESO HVECS+16
Return the present resolution. You must define a USR function and execute the USR function. Code segment example that returns the resolution:

```
DEF USRn = WPEEK(WPEEK(&409E)+16)
PRINT USRn(dummy)
```

HPONT HVECS+18

Return the status of a point. You must specify the location of the point [located at (EX2,WHY2)], define a USR function, and execute the USR function. Code segment example that returns a point status:

```
POKE WPEEK(WPEEK(&409E)+0), LSB X value
POKE WPEEK(WPEEK(&409E)+0)+1, MSB X value
POKE WPEEK(WPEEK(&409E)+2), LSB Y value
POKE WPEEK(WPEEK(&409E)+2)+1, MSB Y value (usually zero)
DEF USRn = WPEEK(WPEEK(&409E)+18)
PRINT USRn(dummy)
```

JDEF HVECS+20

Define the plotting area. You must specify the location of the opposite corners [located at (EX1,WHY1) and (EX2,WHY2)]. Code segment example that defines the plotting area:

```
POKE WPEEK(WPEEK(&409E)+22), LSB X1 value
POKE WPEEK(WPEEK(&409E)+22)+1, MSB X1 value
POKE WPEEK(WPEEK(&409E)+24), LSB Y1 value
POKE WPEEK(WPEEK(&409E)+25)+1, MSB Y1 value (must be zero)
POKE WPEEK(WPEEK(&409E)+0), LSB X2 value
POKE WPEEK(WPEEK(&409E)+0)+1, MSB X2 value
POKE WPEEK(WPEEK(&409E)+2), LSB Y2 value
POKE WPEEK(WPEEK(&409E)+2)+1, MSB Y2 value (must be zero)
CALL WPEEK(WPEEK(&409E)+20)
```

EX1 HVECS+22

This is usually used with WHY1 forming a coordinate pair (EX1,WHY1). EX1 and WHY1 are used when the Hi-RES command requires a pair of coordinates. To place a value into EX1, your code should be similar to the following:

```
POKE WPEEK(WPEEK(&409E)+22), LSB value
POKE WPEEK(WPEEK(&409E)+22)+1, MSB value
```

To return the present value of EX1, your code should be similar to the following:

```
PRINT WPEEK(WPEEK(WPEEK(&409E)+22))
```

WHY1 HVECS+24

This is usually used with EX1 forming a coordinate pair (EX1,WHY1). To place a value into WHY1, your code should be similar to the following:

```
POKE WPEEK(WPEEK(&409E)+24), LSB value
POKE WPEEK(WPEEK(&409E)+24)+1, MSB value (usually zero)
```

To return the present value of WHY1, your code should be similar to the following:

```
PRINT WPEEK(WPEEK(WPEEK(&409E)+24))
```

JBOX HVECS+26

Draw a rectangle. You must specify the location of the opposite corners [located at (EX1,WHY1) and (EX2,WHY2)], define a USR function, and execute the USR function. Code segment example that produces a rectangle:

```
POKE WPEEK(WPEEK(&409E)+22), LSB X1 value
POKE WPEEK(WPEEK(&409E)+22)+1, MSB X1 value
POKE WPEEK(WPEEK(&409E)+24), LSB Y1 value
POKE WPEEK(WPEEK(&409E)+25)+1, MSB Y1 value (usually zero)
POKE WPEEK(WPEEK(&409E)+0), LSB X2 value
POKE WPEEK(WPEEK(&409E)+0)+1, MSB X2 value
POKE WPEEK(WPEEK(&409E)+2), LSB Y2 value
POKE WPEEK(WPEEK(&409E)+2)+1, MSB Y2 value (usually zero)
DEF USRn = WPEEK(WPEEK(&409E)+14)
X = USRn(color|style)
CALL WPEEK(WPEEK(&409E)+26)
```

JLINE HVECS+28

Draw a line. You must specify the location of the line ends [located at (EX1,WHY1) and (EX2,WHY2)], define a USR function, and execute the USR function. Code segment example that produces a line:

```
POKE WPEEK(WPEEK(&409E)+22), LSB X1 value
POKE WPEEK(WPEEK(&409E)+22)+1, MSB X1 value
POKE WPEEK(WPEEK(&409E)+24), LSB Y1 value
POKE WPEEK(WPEEK(&409E)+25)+1, MSB Y1 value (usually zero)
POKE WPEEK(WPEEK(&409E)+0), LSB X2 value
POKE WPEEK(WPEEK(&409E)+0)+1, MSB X2 value
POKE WPEEK(WPEEK(&409E)+2), LSB Y2 value
POKE WPEEK(WPEEK(&409E)+2)+1, MSB Y2 value (usually zero)
DEF USRn = WPEEK(WPEEK(&409E)+14)
X = USRn(color|style)
CALL WPEEK(WPEEK(&409E)+28)
```

PRESS HVECS+34

Returns the last value output to the Hi-RES control port (83H).

To return the present value, your code should be similar to the following:

```
PRINT PEEK(WPEEK(WPEEK(&409E)+34))
```

JTXT and **JRES** HVECS+30 and HVECS+32

These only can be used with machine language routines, because you cannot control what values are placed into specific Z80 registers from BASICH.

The following is an assembly language program that will copy the standard Model 4 MULTIDOS screen onto the Hi-RES screen. You may ORG this program in any location where there is sufficient memory (103 bytes). And, you may place the code into a string because there are no internal calls and/or jumps. One thing that I noticed with Model

4’s that have the gate array board is the characters are one dot position lower than the characters for the original Model 4 and Jeff Vavasour’s Emulator (see line 42).
Put standard screen characters onto Hi-RES screen

```
                00001 ;SCRNHRES
                00002
0084            00003 VRAM    EQU     84H     ;VIDEO and memory configuration port
                00004
0004            00005 VIDWD   EQU     0004H   ;holds present VIDEO width
0033            00006 PUTVO   EQU     0033H   ;display character in A at cursor position
0044            00007 SUBZ    EQU     0044H   ;loads HL with (HL) and (HL)+1: uses A
005A            00008 IMAGE   EQU     005AH   ;holder of memory & video attributes
                00009                         ;usually copy of value output to VRAM
005D            00010 CTHL    EQU     005DH   ;execute where HL is
4416            00011 MXLIN   EQU     4416H   ;returns VIDEO width in A, VIDEO height in D
44FA            00012 FIXRR   EQU     44FAH   ;position cursor to B row and C column
                00013
409E            00014 HVECS   EQU     409EH   ;where HR entry point addresses are found
0002            00015 WHY2    EQU     2       ;offset to pointer to WHY2
000C            00016 NRMTX   EQU     12      ;offset to pointer to NRMTX
001E            00017 JTXT    EQU     30      ;offset to pointer to HTXT
0020            00018 JRES    EQU     32      ;offset to pointer to HRES
                00019
FF80            00020         ORG     0FF80H
FF80  2A9E40    00021 START   LD      HL,(HVECS)      ;get address of table into HL
FF83  7C        00022         LD      A,H             ;get table address MSB into HL
FF84  B7        00023         OR      A               ;is table address MSB = zero?
FF85  C8        00024         RET     Z               ;"Z" if not Hi-RES BASIC
FF86  E5        00025         PUSH    HL              ;save address of HVECS
FF87  112000    00026         LD      DE,JRES         ;offset to pointer to HRES
FF8A  19        00027         ADD     HL,DE           ;HL to location where HRES address is
FF8B  CD4400    00028         CALL    SUBZ            ;get address of HRES into HL
FF8E  1E00      00029         LD      E,0             ;for setting resolution to maximum
FF90  CD5D00    00030         CALL    CTHL            ;set resolution to maximum
FF93  E1        00031         POP     HL              ;retrieve address of HVECS
FF94  E5        00032         PUSH    HL              ;save address of HVECS
FF95  CD4400    00033         CALL    SUBZ            ;get address of EX1 into HL
FF98  110200    00034         LD      DE,WHY2         ;offset to pointer to WHY2 (D=0)
FF9B  72        00035         LD      (HL),D          ;set EX2 LSB to zero
FF9C  23        00036         INC     HL              ;to where EX2 MSB is
FF9D  72        00037         LD      (HL),D          ;set EX2 MSB to zero
FF9E  E1        00038         POP     HL              ;retrieve address of HVECS
FF9F  E5        00039         PUSH    HL              ;save address of HVECS
FFA0  19        00040         ADD     HL,DE           ;HL to location where WHY2 address is
FFA1  CD4400    00041         CALL    SUBZ            ;get address of WHY2 into HL
FFA4  3600      00042         LD      (HL),0          ;WHY2 LSB to start of std. characters
FFA6  23        00043         INC     HL              ;to where WHY2 MSB
FFA7  72        00044         LD      (HL),D          ;set WHY2 MSB to zero
FFA8  E1        00045         POP     HL              ;retrieve address of HVECS
FFA9  111E00    00046         LD      DE,JTXT         ;offset to pointer to HTXT
FFAC  19        00047         ADD     HL,DE           ;HL to location where HTXT address is
FFAD  CD4400    00048         CALL    SUBZ            ;get address of HTXT into HL
FFB0  AF        00049         XOR     A               ;for setting orientation to LR
FFB1  CD5D00    00050         CALL    CTHL            ;set for Hi-RES text
FFB4  CD1644    00051         CALL    MXLIN           ;get screen maximums +1: D = row count
FFB7  5F        00052         LD      E,A             ;hold maximum columns +1 in E
FFB8  010000    00053         LD      BC,0            ;to set cursor to (0,0)
FFBB  3A5A00    00054         LD      A,(IMAGE)       ;get MEMORY/DISPLAY configuration
FFBE  F5        00055         PUSH    AF              ;save MEMORY/DISPLAY configuration
FFBF  D5        00056 loop    PUSH    DE              ;save screen maximums +1
FFC0  2AFB44    00057         LD      HL,(FIXRR+1)    ;get address to position cursor
FFC3  CD5D00    00058         CALL    CTHL            ;position cursor to row B & column C
FFC6  7E        00059         LD      A,(HL)          ;get standard screen character
FFC7  CD3300    00060         CALL    PUTVO           ;put character to Hi-RES screen
FFCA  D1        00061         POP     DE              ;retrieve screen maximums +1
FFCB  0C        00062         INC     C               ;advance column
FFCC  79        00063         LD      A,C             ;get column value
FFCD  93        00064         SUB     E               ;is it above maximum?
FFCE  38EF      00065         JR      C,loop          ;"C" if below maximum
FFD0  4F        00066         LD      C,A             ;set column to zero
FFD1  04        00067         INC     B               ;advance row
```

```
FFD2  78        00068        LD      A,B             ;get row value
FFD3  92        00069        SUB     D               ;is it above maximum?
FFD4  38E9      00070        JR      C,loop          ;"C" if below maximum
FFD6  F1        00071        POP     AF              ;retrieve MEMORY/DISPLAY configuration
FFD7  325A00    00072        LD      (IMAGE),A       ;update MEMORY/DISPLAY holder
FFDA  D384      00073        OUT     (VRAM),A        ;update MEMORY/DISPLAY configuration
FFDC  2A9E40    00074        LD      HL,(HVECS)      ;get addess of table into HL
FFDF  110C00    00075        LD      DE,NRMTX        ;offset to pointer to NRMTX
FFE2  19        00076        ADD     HL,DE           ;HL to location whete NRMTX address is
FFE3  CD4400    00077        CALL    SUBZ            ;get address of NRMTX into HL
FFE6  E9        00078        JP      (HL)            ;set video to standard
                00079
0000            00080        END

No errors.

CTHL  005D   FIXRR 44FA   HVECS 409E   IMAGE 005A   JRES  0020   JTXT  001E   MXLIN 4416   NRMTX 000C
PUTVO 0033   START FF80   SUBZ  0044   VIDWD 0004   VRAM  0084   WHY2  0002   loop  FFBF
```

The above program does not clear the Hi-RES screen, and line 42 should be LD (HL),1 for the gate array board (are they all the same?). Please note the first four instructions determine if you are in BASICH. This is possible because BASIC/CMD values the word 409EH with 0000H and BASICH/CMD values the word 409EH with the address of the HVECS table start.

You normally would use the BASICH Hi-RES statements (the ones beginning with HR) to perform the desired action. e.g., HR B (X1,Y1)-(X2,Y2) to draw a rectangle. The interface to the specific commands enables you to embed Hi-RES commands into short assembly language programs. Or, you may keep the last point plotted for later use.

This interface is used in the following programs:

HRTEST/BAS	uses HVECS+00 (EX2) and HVECS+02 (WHY2) to capture the last point plotted to cap the ends of the 𝕊 in BASIC.
HRPLOT/BAS	uses HVECS+00, HVECS+02, and HVECS+10 in a small assembly language routine to print the coordinate values on the display.
HRSQRSUM/BAS	uses HVECS+00 (EX2), HVECS+02(WHY2), HVECS+06 (HCLS), HVECS+08 (REV), and HVECS+10 (TPLOT) in a small assembly language routine to print the numbers and flash the square being modified. And HVECS+04 (COLOR) is updated in two BASIC statements.
UTIL/BOL	Uses HVECS+12 (NRMTX), to ensure the VIDEO is directed to the device the VIDEO was directed to prior to executing a HR G(x,y),o statement.