

G-Basic Beta-version

MSX Basic extension for the Video9000
and GFX9000

Koen van Hartingsveldt
Albert Beevendorp

Date: 28-12-2011

Contents

Starting G-Basic	3
SHIFT+ESC hot-key	3
G-Basic instructions: syntax and description	4
GBASE	4
GBLOAD	5
GBSAVE	5
GCIRCLE	6
GCLS	7
GCOPY	7
GCOLOR	10
GDRAW	12
GLINE	12
GPAINT	13
GPOINT	14
GPRESET	14
GPSET	14
GPUT SPRITE	15
GSCREEN	15
GSET ADJUST	16
GSET IMAGE	16
GSET PAGE	17
GSET PALETTE	18
GSET SCROLL	18
GSET VIDEO	19
GSPRITES\$	19
GSTOP	21
GVDP	21
GVPEEK	22
GVPOKE	24
The &L hexadecimal notation	25
Screen modes available in G-Basic	26
Charactersitics of palette mode 0 and 1	27
Extended description of the G-Basic screenmodes	28

Starting G-Basic

Make sure you are in MSX Disk-Basic and not in MSX-DOS or MSX-DOS2. Type:

```
BLOAD "GBASIC.BIN",R <return>
```

G-Basic is now installed on your computer. It stays resident, so you can just execute other programs and DOS or DOS2 applications. Note, however, that there exists programs that violate the MSX-standard, and so might overwrite G-Basic partly or completely. Your computer may hang in that case. A simple reset will always restore your computer. DOS1 or Disk Basic 1 applications can also overwrite G-Basic, as the BDOS (Basic Disk Operating System) doesn't provide any memory management and memory reservation.

For this reason it's always possible to uninstall G-Basic from your computer. Uninstall G-Basic at any moment by typing:

```
GSTOP <return>
```

To avoid memory conflicts, using DOS2 is preferred.

Note: By default G-Basic starts with GSCREEN 26,0,0

SHIFT+ESC hot-key

As many GFX9000 programs will not restore the old V9990 settings when quitting, G-Basic provides for a kind of hot-key, that restores the V9990 according to the settings of G-Basic. When leaving a GFX9000 application (and, if applicable, after returning from DOS to Basic), it's wise to press the SHIFT+ESC key combination to be sure that the V9990 settings correspond to the G-Basic settings.

G-Basic performs a GSCREEN ,,0, a GSET PAGE 0,0, a GSET PALETTE 0,0, a GSET SCROLL 0,0, a GCOLOR=NEW and a GSET VIDEO 2,0,0 when pressing SHIFT+ESC. This hot-key can also be used by Video9000 users to make the MSX2/2+ screen visible at once, without having to execute a GSET VIDEO 2,0,0 command.

The hot-key is only active in the G-Basic direct command mode and cannot be used when running a Basic program.

G-Basic instructions: syntax and description

Note: The syntax notations may not be 100% complete, as G-Basic is still in the development-phase. However, the most important forms are given and explained.

GBASE (<n>)=<address>

<n> = 0 to write P1 screenmode sprite pattern generator base address
 1 to write P2 screenmode sprite pattern generator base address
<address> = address on which the sprite pattern generator table should start

This command sets the start address in VRAM of the sprite pattern generator table in the P1 or P2 screen modes. The table start address can only start at the beginning of a 32kB block: &H0000 (top of VRAM map), &H8000 (at 32kB), &L10000 (at 64kB), &L18000 (96kB), etc. The maximum address in the P1 mode is &L38000, and in the P2 mode &L78000.

Because <address> can be a 19-bit value and the &Hxxxx notation can't handle values of more than 16 bits, the &Lxxxxx notation can be used to specify larger values in hexadecimal form. See also page 25.

Example: GBASE (0)=&L18000

Note: This command is not useful at this moment, because the P1 and P2 screen modes are not available yet.

A=GBASE (<n>)

<n> = 0 to read P1 screen mode sprite pattern generator base address
 1 to read P2 screen mode sprite pattern generator base address

This function reads the sprite pattern generator base address (start address). See for more information the GBASE (<n>)=<address> command.

Example: PRINT GBASE (1)

Note: This function is not useful at this moment, because the P1 and P2 screen modes are not available yet.

GBLOAD <filename>,S

<filename> = the name of the file to be loaded

Load a picture from disk.

Example:

```
10 GSCREEN 31
20 GBLOAD "HORSE.PIC",S
```

Note: A picture is only displayed correctly if the correct screenmode is chosen.

Note: In the palette screen modes, the palette data is read automatically. It can be made active by using the GCOLOR=RESTORE command.

Note: The HORSE.PIC file is included. Of course you can also make a digitized picture by yourself, by using the Video9000 digitize program or the GCOPY SCREEN command, and save it in the .PIC format. Then you can load it in G-Basic (using one of the 32768 colour modes).

GBSAVE <filename>,S

GBSAVE <filename>[,<begadr>,<endadr>],S

<filename> = the name of the file to be loaded

<begadr> = begin address of VRAM area to be stored in a file

<endadr> = end address of VRAM area to be stored in a file

Save a picture on disk. If <begadr> and <endadr> are not given, G-Basic automatically saves the contents of the active screen page. If <begadr> and <endadr> are added, G-Basic saves the given VRAM address-space in the file. This address-space is not necessarily equal to one screen page. See page 25 for the &L notation.

Example:

```
10 GSCREEN 17
20 GLINE (0,0)-(511,211),8
30 GBSAVE "REDLINE.G17",S
```

Example:

```
10 GSCREEN 17
20 GSET IMAGE=1024
30 GLINE (0,0)-(1023,211),8
40 GBSAVE "REDLINE.G17",0,&L1A800,S
```

Note: In the first example the palette data is saved as well. In the second example ONLY the address space 0 - &L1A800 is saved.

GCIRCLE [**STEP**] (<X_coor>,<Y_coor>),<radius>[,<colour>
 [, [-]<beg_angle>[, [-]<end_angle>[,<aspect_ratio>
 [, {<log_op> | **TRON** | **CTRON** | **TROFF**}]]]]]

- <X_coor> = X-coordinate of circle center
- <Y_coor> = Y-coordinate of circle center
- <radius> = circle radius, distance from circle center
- <colour> = drawing colour
- <beg_angle> = starting point in radials
- <end_angle> = ending point in radials
- <aspect_ratio> = aspect ratio
- <log_op> = logical operation (see also GCOPY command)
- TRON = transparency on, not changing the original colour value
- CTRON = transparency on and change the original colour into the given value
- TROFF = transparency off, the original colour is visible again

Draw a circle or an ellipse with <X_coor>,<Y_coor> as center with a radius of <radius>. These parameters are the minimum requirements to draw a circle.

The drawing colour can be given by <colour>. The default value is the foreground colour given by the GCOLOR command.

The starting and ending points for the arc can be given by <beg_angle> and <end_angle>. Both values are in radials. Defaults are 0 and 2 * PI to draw a full circle. Providing negative values for these will additionally draw a line from the angle point to the center of the circle or ellipse.

The aspect ratio can be given by <aspect_ratio>. The default value is 0, which will draw a normal circle.

Additionally a logic operation can be given. This can be done with <log_op>. The TRON, CTRON and TROFF options are only valid in the 32768 colour screen-modes. When using the TRON or TROFF command, <colour> is ignored.

Example:

```
10 GSCREEN 25: FOR I=.1 TO 2.9 STEP .2
20 GCIRCLE (128,106),100,31,,,I
30 NEXT I: GCIRCLE (128,106),100,992
```

Note: The visual result of the aspect ratio differs between 50Hz and 60Hz display.

Note: A few bugs in the MSX-Basic CIRCLE command have been removed in GCIRCLE. The program CIRCLE.BAS will show and explain these bugs.

GCLS

Clear screen command. Fills the screen with the backdrop colour. If the backdrop colour is 0, the screen is made transparent automatically, when using a Video9000, except in the YJK and YUV screen modes. The backdrop colour is 0 by default, but can be changed with the GCOLOR command.

```
GCOPY [[STEP] (<X_start>,<Y_start>)]-(<X_end>,<Y_end>)  
        [,<src_page>] TO (<X_dest>,<Y_dest>) [,<dest_page>]  
        [, {<log_op>|TRON} ]]
```

<X_start> = start X-coordinate of the area to be copied
<Y_start> = start Y-coordinate of the area to be copied
<X_end> = end X-coordinate of the area to be copied
<Y_end> = end Y-coordinate of the area to be copied
<X_dest> = destination X-coordinate to which the area must be copied
<Y_dest> = destination Y-coordinate to which the area must be copied
<src_page> = source screenpage
<dest_page> = destination screenpage
<log_op> = logical operation

Copy a rectangular area of page <src_page> with diagonal coordinates (<X_start>, <Y_start>) and (<X_end>,<Y_end>) to coordinate (<X_dest>,<Y_dest>) of page <dest_page>. A logical operation can be applied to the transferred data.

All X- and Y-coordinates are clipped by the imagespace borders rather than by the screenpage borders. If <src_page> or <dest_page> is omitted, the current active page is used (change the active page with the GSET PAGE command).

The following logical operations are available for <log_op>:

PSET = normal draw mode (default, also available as IMP)
OR = perform an OR operation
XOR = perform a XOR operation
PRESET= draw the negated/inverted colour (also available as NOT)
AND = perform an AND operation
EQV = perform a XNOR operation

The logical operations are always applied to the colour of a dot from the area that must be copied (the source colour), and the colour of a dot that would normally have been overwritten (the destination colour).

You can add a T-prefix to <log_op>. In that case if the source dot colour that should be copied turns out to be 0, this dot will not be copied and the old or original dot remains (T = transparent).

Example:

```
10 GSCREEN 31
20 GBLOAD "HORSE.PIC",S
30 GCOPY (0,0)-(99,99) TO (100,100),,TPRESET
```

Example:

```
10 GSCREEN 15: GSET PAGE 1,1: GCLS
20 GLINE (0,0)-(255,211),4
30 GCOPY (100,100)-(199,199),1 TO (100,0),0
40 A$=INPUT$(1): GSET PAGE 0,0
```

Note: In the 32768-colour screenmodes a logical operation, except for TPSET and PRESET operation, performed on a transparent destination area will not change the dot into a non-transparent one; a transparent dot stays transparent.

Note: In the 32768-colour screenmodes the logical operation TRON can be used to copy the transparency states (YS-bits) of the dots, rather than their colours.

```
GCOPY [[STEP] (<X_start>,<Y_start>)]-(<X_end>,<Y_end>)
[,<src_pag>] TO <filename>
GCOPY <filename>[,<dir>] TO (<X_dest>,<Y_dest>)
[,<dest_pag>[,<log_op>]]
```

<filename> = filename of the COPY-file

<dir> = 0 normal
1 horizontal mirrored
2 vertical mirrored
3 horizontal and vertical mirrored

The given area is copied to a file with name <filename>, or a file containing the image data of a certain area is copied to the VRAM at the given coordinates. If a COPY-file is loaded to VRAM, the area can be mirrored before it's placed at the screen, by using the <dir> parameter.

Example:

```
10 GSCREEN 31: GBLOAD "HORSE.PIC",S
20 GCOPY (100,100)-(199,199) TO "PIECE.C31"
30 GCLS: GCOPY "PIECE.C31",3 TO (0,0)
```



```

GCOPY [[STEP] (<X_start>,<Y_start>)]-(<X_end>,<Y_end>)
[,<src_pag>] TO <array>
GCOPY <array>[,<dir>] TO (<X_dest>,<Y_dest>)[,<dest_pag>
[,<log_op>]]

```

<array> = array variable name

Copies a given area into an array variable, or copies image data from an array variable to the screen. The array size must be large enough. The minimum array size can be calculated with:

$$\langle \text{array_size} \rangle \geq (\text{INT} ((\text{bits/dot}) * \langle \text{nx} \rangle * \langle \text{ny} \rangle + 7) / 8) + 4) / \langle \text{var_size} \rangle$$

<nx> = ABS (<X_end> - <X_start>) + 1

<ny> = ABS (<Y_end> - <Y_start>) + 1

<bits/dot> = 2 for GSCREEN 6 and 16
4 for GSCREEN 5, 7, 15, 17, 27 and 28
8 for GSCREEN 8, 10-14, 18-24, 29 and 30
16 for GSCREEN 25, 26 and 31

<var_size> = 2 for integer array
4 for single precision array
8 for double precision array

Use DIM <var_name> (<array_size> - 1) to define the array.

Example:

```

10 GSCREEN 31
20 DEFINT A: DIM A (807)
30 GBLOAD "HORSE.PIC",S
40 GCOPY (20,20)-(45,50) TO A
50 GCOPY A TO (100,100)

```

```

GCOPY <array> TO <filename>
GCOPY <filename> TO <array>
GCOPY <filename> TO <filename>

```

To copy image data from an array variable to a file or vice versa or to copy a file to another file.

Note: The normal COPY commands from MSX-Basic can be used as well. The above three command are just executing the original COPY command.

GCOPY SCREEN [= {**ON** | **OFF** }]

Digitizes a picture. GCOPY SCREEN grabs a picture only once, or twice during interlace. GCOPY SCREEN=ON turns the continuous digitization mode on and GCOPY SCREEN=OFF halts the continuous digitization. GCOPY SCREEN=ON is more or less equivalent with:

```
10 GCOPY SCREEN: GOTO 10
```

but is much faster. The continuous digitization mode is also halted after a normal GCOPY SCREEN command.

Example:

```
10 GSCREEN 26
20 GSET VIDEO 1,0,1
30 GCOPY SCREEN
```

Note: The digitized image is stored in the current visual page (display page) and not in the active page.

Note: This command is only applicable if a Video9000 is present.

GCOLOR [<foreground_colour>[,<backdrop_colour>[,<border_colour>]]]

<foreground_colour> = change the default drawing colour, e.g. used by GLINE
<backdrop_colour> = change the default backdrop colour, e.g. used by GCLS
<border_colour> = change the border colour

In all screenmodes except for the YJK and YUV modes, colour 0 is transparent when having a Video9000.

Example:

```
10 GSCREEN 17
20 GCOLOR 15,0,4
30 GLINE (300,100)-(500,200),,BF
40 GCOLOR 8
50 GLINE (100,0)-(300,100),,BF
60 A$=INPUT$ (1)
70 GCLS: GCOLOR ,3
80 A$=INPUT$ (1)
90 GCLS
```

Note: <border_colour> can never be higher than 63 and is always a palette colour.

GCOLOR= (<pal_col>[,<R-value>[,<G-value>[,<B-value>]])

<pal_col> = the palette colour that has to be changed

<R-value> = the value of the RED colour component (0 - 31, 128)

<G-value> = the value of the GREEN colour component (0 - 31)

<B-value> = the value of the BLUE colour component (0 - 31)

Changes the contents of the colour palette. Each palette number can have its own colour, chosen out of 512 colours in palette mode 0 and out of 32768 colours in palette mode 1. It is also possible just to change one colour component, instead of three.

Example:

```
10 GSCREEN 14
20 GLINE (100,100)-(300,200),10,BF
30 A$=INPUT$(1): GCOLOR=(10,31,0,0)
40 A$=INPUT$(1): GCOLOR=(10,,,31)
```

Note: In palette mode 0 <value> must be 0 - 7.

Note: If <R-value>=128, the specified palette colour is made transparent when having a Video9000. This is only possible in palette mode 1.

A=GCOLOR (<pal_col>,<component>)

<pal_col> = the palette colour that has to be read

<component> = R to read Red-intensity value

G to read Green-intensity value

B to read Blue-intensity value

T to read transparency status (YS-bit)

Reads the active palette data. If <component>=T then A=-1 (True) if palette colour is transparent and A=0 (False) if not.

Example:

```
10 GSCREEN 13
20 PRINT GCOLOR (4,B)
```

GCOLOR=NEW

Initializes all palettes of the active page to the default settings. After a GSCREEN <mode> command the palette is initialized automatically.

GCOLOR=RESTORE

Restores all palettes according to the palette back-up data stored in the active page. Use this command to restore the palette colours after having loaded a palette picture with the GBLOAD command.

GDRAW

Not supported yet.

GLINE [[**STEP**] (<X_start>,<Y_start>)] - (<X_end>,<Y_end>)
[,<colour>[, {**B** | **BF**} [, {<log_op> | **TRON** | **CTRON** |
TROFF}]]]

<X_start> = start X-coordinate

<Y_start> = start Y-coordinate

<X_end> = end X-coordinate

<Y_end> = end Y-coordinate

<colour> = drawing colour

B = draw a box instead of a line

BF = draw a filled box instead of a line or box

<log_op> = logical operation (see also GCOPY command)

TRON = transparency on, not changing the original colour value

CTRON = transparency on and change the original colour into the given value

TROFF = transparency off, the original colour is visible again

Draws a line, box of filled rectangle on the screen. The coordinates are clipped by the imagespace borders rather than by the screen page borders. The TRON, CTRON and TROFF options are only valid in the 32768 colour screenmodes. When using the TRON or TROFF command, <colour> is ignored.

Example:

```
10 GSCREEN 14
20 GLINE (10,10)-(100,100),4
30 GLINE (100,100)-(200,150),8,B
40 GLINE (200,150)-(400,50),2,BF
```

Example:

```
10 GSCREEN 31
20 GBLOAD "HORSE.PIC",S
30 GLINE (100,100)-(200,200),992,BF,TRON
40 A$=INPUT$(1)
50 GLINE (100,100)-(200,200),992,BF,TROFF
```

```

60 A$=INPUT$ (1)
70 GLINE (100,100)-(200,200),992,BF,CTRON
80 A$=INPUT$ (1)
90 GLINE (100,100)-(200,200),,BF,TROFF

```

Note: In the palette screenmodes an area can be made transparent by filling that area with colour 0. The old picture data of that area is lost then.

GPAINT [**STEP**] (<X_coor>,<Y_coor>) [,<fill_colour>
[,<border_colour>[,{<log_op>|**TRON**|**CTRON**|**TROFF**}]
,<option>]]]]

<X_coor> = X-coordinate of starting point
<Y_coor> = Y-coordinate of starting point
<fill_colour> = colour to fill the area with
<border_colour> = edge colour of the area to be filled
<log_op> = logical operation (see also GCOPY command)
<option> = 0 the filling cannot extend beyond the screen page borders
1 the filling can/will extend beyond the screen page borders
TRON = transparency on, not changing the original colour value
CTRON = transparency on and change the original colour into the given value
TROFF = transparency off, the original colour is visible again

To fill a arbitrary shaped area surrounded by a closed border. The point to start the filling is given by <X_coor> and <Y_coor>, the painting colour is given by <fill_colour>. The default <fill_colour> is the value set by the GCOLOR <foreground_colour> command. The edge of the area is specified by its colour with <border_colour>. By default <border_colour> is equal to <fill_colour>.

In the 32768 colour screenmodes **TRON**, **CTRON** and **TROFF** can be used to make the area transparent or non-transparent with or without altering the original area colour(s). For more information, see the GLINE command.

By making <option>=1 the **GPAINT** instruction van be used to fill areas in the entire imagespace, rather than in just a screen page. Areas that are partly in one screenpage and partly in an adjacent screen page can be filled by setting this option at 1. The default value is 0.

Example:

```

10 GSCREEN 19
20 GLINE (100,100)-(300,0),8
30 GLINE (300,0)-(500,100),8
40 GLINE (500,100)-(300,200),8

```

```
50 GLINE (300,200)-(100,100),8
60 GPAINT (300,100),4,8
```

Example:

```
10 GSCREEN 31
20 GBLOAD "HORSE.PIC",S
30 GLINE (100,50)-(200,150),31,B
40 GPAINT (150,100),,31,,TRON
50 A$=INPUT$(1)
60 GPAINT (150,100),,31,,TROFF
```

A=GPOINT (<X_coor>,<Y_coor>)[,T]

<X_coor> =X-coordinate of dot

<Y_coor> =Y-coordinate of dot

Returns the colour value of the specified dot. The coordinates can be specified over the entire imagespace. In the 32768-colour screenmodes the T-option can be added. By doing so, -1 (True) will be returned if the specified dot is transparent and 0 (False) if the dot is non-transparent.

GPRESET

Works the same way as GPSET, The default <colour> is the value set by the GCOLOR <backdrop_colour> command.

**GPSET [STEP] (<X_coor>,<Y_coor>)[,<colour>[,<log_op>|
TRON|CTRON|TROFF]]]**

<X_coor> =X-coordinate of dot

<Y_coor> =Y-coordinate of dot

<colour> =drawing colour

<log_op> =logical operation, see also GCOPY command

TRON =transparency on, not changing the original colour value

CTRON =transparency on and change the original colour into the given value

TROFF =transparency off, the original colour is visible again

Draws one dot on the screen at the specified position and with the given colour value. The default <colour> is the value set by the GCOLOR <foreground_colour> command.

Example:

```
10 GSCREEN 13
20 GPSET (100,100),15
```

GPSET SPRITE <nr>[, [STEP] (<X_coor>,<Y_coor>) [,<colour>]]

<nr> = sprite number (0 or 1)
<X_coor> = X-coordinate of dot
<Y_coor> = Y-coordinate of dot
<colour> = sprite colour (0-3, 255)

Places one of the two hardware cursors on the screen at the specified position. The hardware cursors can only have one colour and the colour value itself is limited to 5 different values, of which colour 0 is transparent. Colour 255 activates a XOR-coloured sprite, where each 1-pixel of the sprite pattern is logically XOR'ed with the colour of the graphic pixel it covers. This way the sprite will always be visible. As soon as the P1 and P2 modes are supported, this instruction will become very powerful, due to the full colour sprites available in these modes.

The coordinates can extend from -32 to the right resp. lower screenpage border.

Example: See GSPRITE\$ command.

Note: The Y-coordinate is internally decreased with 1 to correct it accordingly to the picture Y-coordinate.

GSCREEN <mode>[,<on/off>[,<interlace>[,<freq>]]]

<mode> = 5 - 31, except 9, see table at page 26
<on/off> = 0 image and sprites visible (default value)
 1 sprite invisible (switched off)
 2 sprites and image display switched off
<interlace> = 0 non-interlaced mode
 1 non-interlaced mode, only odd lines visible
 2 interlaced, not doubled Y-res.
 3 interlaced, double Y-res.
<freq> = 0 50Hz frame frequency
 1 60Hz frame frequency

To set the screenmode, to switch the sprites and the display on and off, to switch the interlaced mode on and off, and to chose the frame frequency.

Example:

```
GSCREEN 26,1,3,0
```

Note: In GSCREEN 31, <interlace> can't be 1 or 3 when using a 50Hz frame frequency.

Note: By default G-Basic starts with GSCREEN 26,0,0

GSET ADJUST (<X_shift>,<Y_shift>)

<X_shift> = the displacement of the screen in horizontal direction (-7 to 8)

<Y_shift> = the displacement of the screen in vertical direction (-7 to 8)

Adjusts the display position. By default G-Basic starts with the display position set at (0,0).

GSET IMAGE=<image_width>

<image_width> = the width of the imagespace.

Sets the X-size of the imagespace. The Y-size is automatically set, as the total number of available dots is fixed for every screenmode. The Y-size can be calculated as follows:

<Y-size> = $4194304 / (<image_width> * <bits/dot>)$

<bits/dot> = 2 for GSCREEN 6 and 16

4 for GSCREEN 5, 7, 15, 17, 27 and 28

8 for GSCREEN 8, 10 - 15, 18 - 24, 29 and 30

16 for GSCREEN 25, 26 and 31

<image_width> can only be 256, 512, 1024 or 2048, but not all of these are possible in every screenmode. So, which of these four values can be used depends on the current screen mode.

Example:

```
10 GSCREEN 5
```

```
20 GSET IMAGE=512
```


Now the imagespace is like: 0 256

0	page 0	page 1
256	page 2	page 3
512	page 4	page 5
768	page 6	page 7
1024	page 8	page 9
1280	page 10	page 11
1536	page 12	page 13
1792	page 14	page 15

```
GSET IMAGE=256
```

Now the imagespace is as wide as one screenpage (256 dots), so all 16 pages are just below each other.

Note: After a GSET IMAGE command, a GCLS should be execute to clear the screen.

Note: Usually, changing the imagespace size will deteriorate a picture that was already stored in memory, because that picture was loaded or made during another imagespace size.

GSET PAGE [*<visual_page>* [, *<active_page>*]]

<visual_page> = the screenpage that will be showed

<active_page> = the screenpage on which all grafical commands will act (on which the lines will be drawn, etc)

Selects the visible screenpage which is display and the activescreen page on which all the graphical commands are acting. The number of available pages depends on the actual screen mode. GSCREEN 15, for instance, has 16 pages available.

Example:

```
10 GSCREEN 15
20 GSET PAGE 13,13: GCLS
30 GLINE (0,0)-(255,211),8
40 GSET PAGE ,0
50 GLINE (255,0)-(0,211),4
60 A$=INPUT$(1)
70 GSET PAGE 0
```

GSET PALETTE [<visual palette>[,<active palette>]]

<visual palette> = palette which is visible (colours according to this palette)

<active palette> = palette to which GCOLOR=(c,r,g,b), A=GCOLOR (c,R), etc. commands are applied

Change the visible and the active palette. Not all screenmodes have more than one palette. GSCREEN 15 has 4 palettes of 16 colours, GSCREEN 16 has 16 palettes of 4 colours, and GSCREEN 14 has 1 palette of 64 colours.

Example:

```
10 GSCREEN 15
20 GLINE (0,0)-(255,211),8
30 GSET PALETTE ,1
40 GCOLOR=(8,31,31,31)
50 A$=INPUT$ (1)
60 GSET PALETTE 1
```

Note: Changing palette is only possible in palette mode 1.

Note: GCOLOR=NEW and GCOLOR=RESTORE changes all palettes of, or according to the active page.

GSET SCROLL [<X_coor>[,<Y_coor>]]

<X_coor> = X_coordinate of the dot that will be shifted to the most upper-left corner of the screen

<Y_coor> = Y_coordinate of the dot that will be shifted to the most upper-left corner of the screen

Example:

```
10 GSCREEN 14
20 GLINE (0,0)-(511,211),63
30 A$=INPUT$ (1)
40 GSET SCROLL 256,0
```

Note: You can scroll through the complete imagespace (travelling through all screen pages).

Note: After GSET SCROLL 100,100 the coordinate of the upper-left dot is (100,100) and not (0,0)

GSET VIDEO [`<mode>` [, `<Ym>` [, `<source>`]]]

`<mode>` = 0 normal mode, only V9990 visible, no superimpose and digitization
1 digitize mode, no superimposition
2 superimpose mode
3 only external video signal visible
`<Ym>` = 0 no half-tone, no mixing
1 external signal displayed with half intensity
2 V9990 is displayed with half intensity
3 mixing mode, external video signal and V9990 mixed together
`<source>` = 0 select RGB input
1 select CVBS input
2 select S-Video input

Sets the superimpose, digitize and mixing modes, and selects the external video input to be used. G-Basic starts with `GSET VIDEO 2,0,0`, so that the MSX2/2+/TurboR text screen is visible.

Note: This instruction is only applicable when having a Video9000.

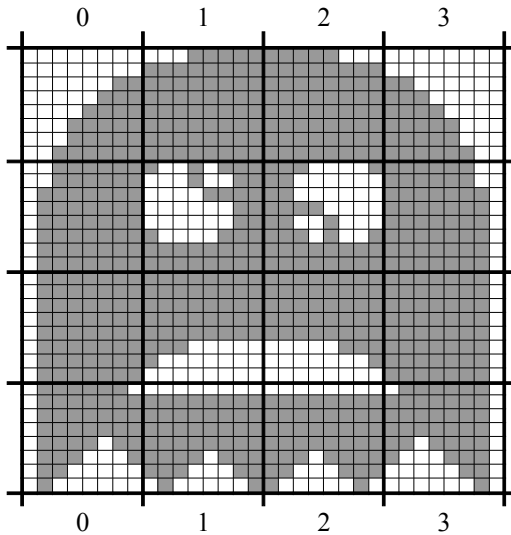
Note: Be sure that the frame frequency is set to 50Hz if PAL video signals are used (CVBS or S-Video input). Otherwise superimposing is not performed properly.

GSPRITE\$ (`<nr>`)=`<string>`

`<nr>` = sprite number (0 or 1)
`<string>` = string of $32 * 4 = 128$ characters defining the sprite pattern

Defines the pattern of the specified sprite according to `<string>`. There are two sprites available (two hardware cursors) each having a size of 32×32 dots.

Example: We want to define the following sprite pattern for sprite 1



Make the first link of $4 \times 8 = 32$ dots (line 0):

1st	8 bits: &B00000000	->	0
2nd	8 bits: &B00011111	->	31
3rd	8 bits: &B11111000	->	248
4th	8 bits: &B00000000	->	0

The the second line of 32 dots (line 1):

1st	8 bits: &B00000000	->	0
2nd	8 bits: &B11111111	->	255
3rd	8 bits: &B11111111	->	255
4th	8 bits: &B00000000	->	0

Likewise make all 32-dot lines (2 - 31) in that order. Finally we have:

```

100 DATA 0,31,248,0,0,255,255,0:' Line 0 & 1
110 DATA 3,255,255,192,7,255,255,224:' Line 2 & 3
120 DATA 15,255,255,240,31,255,255,248:' Line 4 & 5
130 DATA 31,255,255,248,63,255,255,252:' Line 6 & 7
140 DATA 63,255,255,252,63,19,192,252:' Line 8 & 9
150 DATA 127,15,192,254,127,3,240,254:' Line 10 & 11
160 DATA 127,3,200,254,127,255,255,254:' Line 12 & 13
170 DATA 127,255,255,254,127,255,255,254:' Line 14 & 15
180 DATA 127,255,255,254,127,255,255,254:' Line 16 & 17
190 DATA 127,255,255,254,127,255,255,254:' Line 18 & 19
200 DATA 127,255,255,254,127,224,7,254:' Line 20 & 21
210 DATA 127,128,1,254,127,0,0,254:' Line 22 & 23
220 DATA 126,0,0,126,127,255,255,254:' Line 24 & 25

```

```
230 DATA 127,255,255,254,127,255,255,254:' Line 26 & 27
240 DATA 123,255,255,222,113,247,239,142:' Line 28 & 29
250 DATA 96,227,199,6,64,65,130,2:' Line 30 & 31
```

Add the following lines:

```
10 CLEAR 300: DEFINT A-Z: GSCREEN 15,0
20 RESTORE 100: A$=""
30 FOR A=0 TO 127: READ B: A$=A$+CHR$(B): NEXT
40 GSPRITE$(1)=A$: A$=""
50 FOR X=-32 TO 255 STEP 2
60 GWAIT 150:' Wait for display-line 150
70 GPUT SPRITE 1, (X,100),3
80 NEXT
90 GOTO 50
```

Note: The WAIT instruction makes sure that there is only one sprite movement per display refresh so that the sprite movement will be smooth.

GSTOP

To uninstall G-Basic. After this command, none of the G-Basic commands are available anymore. Only reinstalling G-Basic by BLOAD "GBASIC.BIN",R will make the commands available again.

GVDP (<register>)=<value>

<register> = V9990 register number (0 - 28, 32 - 52)

<value> = 8-bit value to write to the specified register

Writes to a V9990 register. The 8-bit value is written to the specified register of the V9990 Video Display Processor. The function of each register can be found in the V9990 application manual.

Example:

```
10 GSCREEN 26: GSET IMAGE=512
20 GSET PAGE ,0: GCOLOR ,992: GCLS
30 GSET PAGE ,1: GCOLOR ,31744: GCLS
40 GVDP (6)=&B10011011:' Image width is 1024 dots now
50 ' GSET IMAGE=1024 would be much more convenient
```

Note: Changing the V9990's registers may cause G-Basic commands or functions to act incorrectly. Therefore, care has to be taken when writing to the registers directly. The GSCREEN command restores most of the registers.

A=GVDP (<register>)

<register> = V9990 register number (6 - 13, 15 - 27, 53, 54)

Reads a V9990 register.. The 8-bit contents of the specified register of the V9990 Video Display Processor is returned. The function of each register can be found in the V9990 application manual.

Example:

```
10 GSCREEN 26
20 A=GVDP (6)
30 PRINT "Bit: 76543210"
40 PRINT "-----"
50 LOCATE 5: PRINT RIGHT$ ("00000000"+BIN$ (A),8)
```

Note: Reading the V9990's registers will never cause G-Basic commands or functions to act incorrectly.

A=GVPEEK (<address>)

<address> = V9990 VRAM absolute address to read from (19-bit value)

Reads a byte from the specified VRAM address. Because <address> can be a 19-bit value and the &Hxxxx notation can't handle values of more than 16 bits, the &Lxxxxx notation can be used to specify larger values in hexadecimal form. See also page 25.

To calculate the VRAM address from the X- and Y-coordinate, the following formula can be used:

$$\text{<address>} = \text{INT} (\text{<bits/dot>} * (\text{<image_width>} * \text{<Y_coor>} + (\text{<X_coor>} / 8))$$

- <bits/dot> = 2 for GSCREEN 6 and 16
- 4 for GSCREEN 5, 7, 15, 17, 27 and 28
- 6 for GSCREEN 8, 10 - 14, 18 - 24, 29 and 30
- 8 for GSCREEN 25, 26 and 31

<image_width> = imagespace width in dots (see GSET IMAGE)

<X_coor> = X-coordinate

<Y_coor> = Y-coordinate

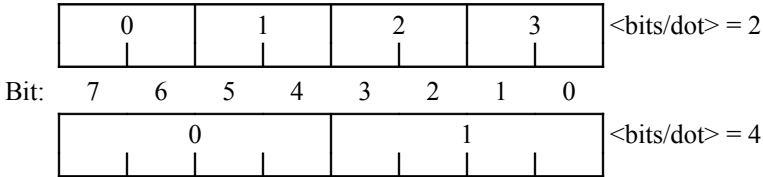
If <bits/dot> = 16, the above calculated address is the address of the lower byte of dot (<X_coor>,<Y_coor>). The high-byte is located at <address>+1.

If <bits/dot> = 4 or 2, the byte at the calculated address contains the colour value of more than one dot. Which group of bits specifies the colour value of the dot at

($\langle X_coord \rangle, \langle Y_coord \rangle$) can be determined with:

$$\langle index \rangle = \langle X_coord \rangle \text{ MOD } (8 / \langle bits/dot \rangle)$$

In the figure below the two or four bits of the dot are indicated by the calculated index number.



Example:

```
10 GSCREEN 18
20 GSET IMAGE=256: GCLS
30 GPSET (0,0),255
40 PRINT GVPEEK (0)
50 GPSET (1,511),123
60 PRINT GVPEEK (&L1FF01)
```

Note: The address of dot (0,0) at page 0 differs from the address of dot (0,0) at page 1. So the GSET PAGE command doesn't change the address. The specified address is an absolute address.

GVPOKE <address>,<value>

<address> = V9990 VRAM address to write to (19-bit value)

<value> = 8-bit value to write to the specified address

Writes a byte to the specified VRAM address. Because <address> can be a 19-bit value and the &Hxxxx notation can't handle values of more than 16-bits, the &Lxxxxx notation can be used to specify larger values in hexadecimal form. See also page 25.

Example:

```
10 GSCREEN 15
20 GSET IMAGE=256: GCLS
30 A=GVPEEK (&H3232) AND &H0F
40 GVPOKE (&H3232),A OR &H80
50 ' GPSET (100,100),8 would be more convenient
```

Note: The address of dot (0,0) at page 0 differs from the address of dot (0,0) at page 1. So the GSET PAGE commands doesn't change the address. The specified address is an absolute address.

The &L hexadecimal notation

Because the V9990 has a 512kB VRAM size, the address to be given in the GVPOKE, GVPEEK, GBASE and GBSAVE instructions are 19-bit values generally. A 16-bit address is usually given in the hexadecimal notation with the &H prefix, like:

```
VPOKE &HFA00, 0
```

However, the &H notation can only be used with 16-bit values. Typing:

```
PRINT &H1FA00
```

will give an overflow error.

For this reason G-Basic provides for a 19-bit equivalent of the &H prefix: the &L prefix (L = long hexadecimal). Thus typing:

```
PRINT &L1FA00
```

will not give an error and returns the value 129536.

Now the complete address space of the V9990, ranging from &L00000 to &L7FFFF, can be used.

Example:

```
10 PRINT &HFFFF
20 PRINT &LFFFF
30 PRINT &L10000
40 PRINT &L10000+&HFFFF
50 PRINT &L7FFFF
```

Note: There is no 19-bit equivalent of the HEX\$ instruction. LEX\$ or LHEX\$ do not exist.

Note: The &L prefix can't be used in combination with the VAL function. VAL("&L"+"7FFFF") will give a syntax error.

Note: The &L prefix can't be used in DATA statements. DATA &L3FFFF will give an error.

Screen modes available in G-Basic

- GSCREEN 5 256×212, 16 palette colours, palette mode 0 (like SCREEN 5)
- 6 512×212, 4 palette colours, palette mode 0 (like SCREEN 6)
- 7 512×212, 16 palette colours, palette mode 0 (like SCREEN 7)
- 8 256×212, 256 fixed colours, palette mode 0 (like SCREEN 8)
- 9 not used
- 10 256×212, 12499 colours BYJK, palette mode 0 (like SCREEN 10)
- 11 256×212, 12499 colours BYJK, palette mode 0 (like SCREEN 11)
- 12 256×212, 19286 colours YJK, palette mode 0 (like SCREEN 12)
- 13 256×212, 64 palette colours
- 14 512×212, 64 palette colours
- 15 256×212, 16 palette colours, palette mode 1, 4 palettes
- 16 512×212, 4 palette colours, palette mode 1, 16 palettes
- 17 512×212, 16 palette colours, palette mode 1, 4 palettes
- 18 256×212, 256 fixed colours, palette mode 1, 1 palette
- 19 512×212, 256 colours fixed (high-resolution GSCREEN 18)
- 20 256×212, 12499 colours BYJK, palette mode 1, 4 palettes
- 21 256×212, 12499 colours BYJK, palette mode 1, 4 palettes
- 22 256×212, YUV (like GSCREEN 12 but then YUV)
- 23 512×212, YJK (high-resolution SCREEN 12)
- 24 512×212, YUV
- 25 256×212, 32768 fixed colours
- 26 512×212, 32768 fixed colours
- 27 384×240/290, overscan, 16 palette colours, 4 palettes
- 28 768×240/290, overscan, 16 palette colours, 4 palettes
- 29 384×240/290, overscan, 64 palette colours, 1 palette
- 30 384×240/290, overscan, 256 fixed colours
- 31 384×240/290, overscan, 32768 fixed colours

Note: In the overscan screenmodes GSCREEN 27-31, the Y-resolution is 240 lines when using a 60Hz frame frequency, and 290 when using 50Hz.

Note: YUV screenmodes are more suitable for digitized pictures than YJK modes.

Characteristics of palette mode 0 and 1

Mode 0:- R,G,B-intensities ranging from 0-7 only (see GCOLOR=(c,r,g,b))

- Only one palette of 16 colours available
- GSET PALETTE command not available
- Palette data is stored in VRAM just below each screen page (somewhere between lines 212 and 255)
- In graphics files (like .GE5, .GE7, .PIC, etc.) the palette data is stored not directly below the actual image data; there is a gap between the image data and the palette data, just like at the MSX2/2+

Mode 1:- R,G,B-intensities ranging from 0-31 (full range)

- More than one palette available (e.g. GSCREEN 15 has 4 palettes)
- GSET PALETTE command is available
- Palette data for ALL screen pages is stored in VRAM completely at the bottom of the VRAM map (imagespace)
- In graphics files the palette data is stored directly after the actual image data

Only GSCREEN 5-12 are palette mode 0, all the others are always palette mode 1. Palette mode 1 is much more sophisticated than palette mode 0. Especially the fact that almost the complete imagespace can be used for image data without corrupting the palette data is a big advantage.

For this reason GSCREEN 5-12 pictures can be converted to palette mode 1 (GSCREEN 15-22) just by changing from GSCREEN 5 to GSCREEN 15, GSCREEN 6 to GSCREEN 16, etc. The image is not erased in those situations.

The palette mode 0 screenmodes are made to be compatible with MSX2/2+ screenmodes and are only needed when 'old' MSX2/2+ pictures are loaded. It's preferred to convert these pictures to palette mode 1, but converted pictures are not MSX2/2+ compatible any more. To make them MSX2/2+ compatible again, convert them back to palette mode 0, by changing the screen mode.

Note: MSX2/2+ pictures that don't use the palette data, that is stored in VRAM, can be loaded directly in a palette mode 1 screenmode.

Extended description of the G-Basic screenmodes

In this section each screenmode is described more extensively. The resolution, number of colours, number of palettes, possible imagespace sizes and default settings are mentioned.

All the imagespace layouts are valid for the non-interlaced situation. If interlacing is used, one non-interlaced page and the page below will form one interlaced page.

Example: In GSCREEN 5, with GSET IMAGE=512, non-interlaced page 0 and 2 will be combined to interlaced page 0, non-interlaced page 1 and 3 will be combined to interlaced page 1, non-interlaced page 4 and 6 will be combined to interlaced page 2, etc.

Screenmode 27 to 31 are the so-called overscan modes, which means that these screenmodes don't have a border around the image. The complete display area can be used for drawing, etc.

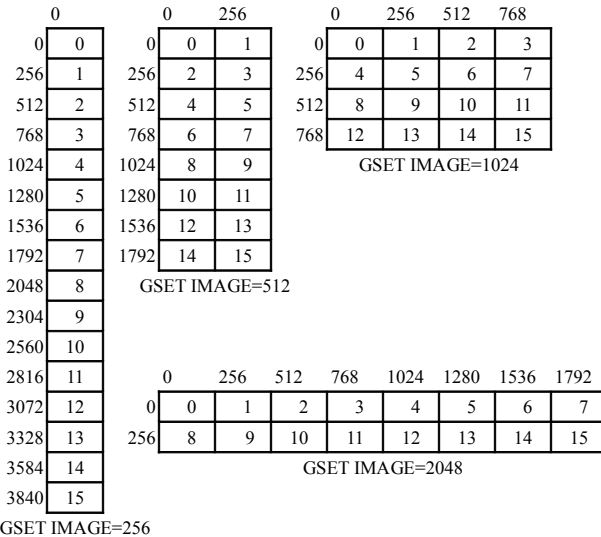
Note: In some screenmodes the largest available imagespace width may become invalid when interlacing is used. If this largest imagespace is used, the upper and lower half of the interlaced screen will be exactly the same.

Note: In the overscan modes a difference in vertical resolution, page size, available number of pages, and available imagespace sizes exists between 50Hz and 60Hz operation.

GSCREEN 5

Resolution: 256 × 212 non-interlaced
 256 × 424 interlaced
 Colours: 16 palette colours out of 512
 Page size: 256 × 256 non-interlaced
 256 × 512 interlaced
 Pages available: 16 pages non-interlaced
 8 pages interlaced
 Palette mode: 0
 Palettes available: --
 Imagespace width: 256, 512, 1024 and 2048 dots
 Number of sprites: 2
 Sprite size: 32 × 32 dots
 Sprite colours: 3 + transparent colour

Imagespace layout:



The default imagespace width after a GSCREEN <mode> execution is 512 dots.

To switch to the same screen mode but then with palette mode 1, just execute the GSCREEN 15 instruction. The image and palette data are automatically converted from palette mode 0 to palette mode 1. None of the pages will be erased.

GSCREEN 6

Resolution: 512 × 212 non-interlaced
 512 × 424 interlaced
 Colours: 4 palette colours out of 512
 Page size: 512 × 256 non-interlaced
 512 × 512 interlaced
 Pages available: 16 pages non-interlaced
 8 pages interlaced
 Palette mode: 0
 Palettes available: --
 Imagespace width: 512, 1024 and 2048 dots
 Number of sprites: 2
 Sprite size: 32 × 32 dots
 Sprite colours: 3 + transparent colour

Imagespace layout:

0	0	0	512	0	1	0	512	1024	1536	0	1	2	3
256	1	256		2	3	256				4	5	6	7
512	2	512		4	5	512				8	9	10	11
768	3	768		6	7	768				12	13	14	15
1024	4	1024		8	9					GSET IMAGE=2048			
1280	5	1280		10	11								
1536	6	1536		12	13								
1792	7	1792		14	15								
2048	8	GSET IMAGE=1024											
2304	9												
2560	10												
2816	11												
3072	12												
3328	13												
3584	14												
3840	15												

GSET IMAGE=512

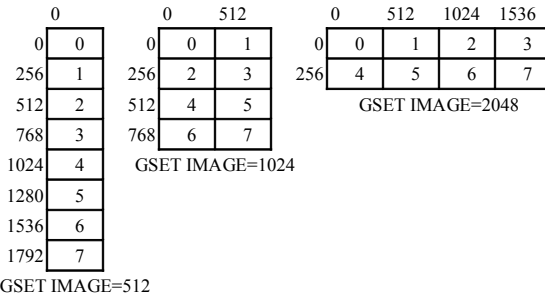
The default imagespace width after a GSCREEN <mode> execution is 1024 dots.

To switch to the same screen mode but then with palette mode 1, just execute the GSCREEN 16 instruction. The image and palette data are automatically converted from palette mode 0 to palette mode 1. None of the pages will be erased.

GSCREEN 7

Resolution: 512 × 212 non-interlaced
512 × 424 interlaced
Colours: 16 palette colours out of 512
Page size: 512 × 256 non-interlaced
512 × 512 interlaced
Pages available: 8 pages non-interlaced
4 pages interlaced
Palette mode: 0
Palettes available: --
Imagespace width: 512, 1024 and 2048 dots
Number of sprites: 2
Sprite size: 32 × 32 dots
Sprite colours: 3 + transparent colour

Imagespace layout:



The default imagespace width after a GSCREEN <mode> execution is 1024 dots.

To switch to the same screen mode but then with palette mode 1, just execute the GSCREEN 17 instruction. The image and palette data are automatically converted from palette mode 0 to palette mode 1. None of the pages will be erased.

GSCREEN 8

Resolution: 256 × 212 non-interlaced
 256 × 424 interlaced
 Colours: 256 fixed colours
 Page size: 256 × 256 non-interlaced
 256 × 512 interlaced
 Pages available: 16 pages non-interlaced
 8 pages interlaced
 Palette mode: 0
 Palettes available: --
 Imagespace width: 256, 512, 1024 and 2048 dots
 Number of sprites: 2
 Sprite size: 32 × 32 dots
 Sprite colours: 3 + transparent colour

Imagespace layout:



The default imagespace width after a GSCREEN <mode> execution is 512 dots.

To switch to the same screen mode but then with palette mode 1, just execute the GSCREEN 18 instruction. The image and palette data are automatically converted from palette mode 0 to palette mode 1. None of the pages will be erased.

Note: In this screen mode, the palette is only used for the border colour.

GSCREEN 10

Resolution: 256 × 212 non-interlaced
 256 × 424 interlaced

Colours: 16 palette colours out of 512 (for drawing and displayed)
 12499 fixed colours, 16-bit YJK (displayed only)

Page size: 256 × 256 non-interlaced
 256 × 512 interlaced

Pages available: 8 pages non-interlaced
 4 pages interlaced

Palette mode: 0

Palettes available: --

Imagespace width: 256, 512, 1024 and 2048 dots

Number of sprites: 2

Sprite size: 32 × 32 dots

Sprite colours: 3 + transparent colour

Imagespace layout:



The default imagespace width after a GSCREEN <mode> execution is 512 dots.

To switch to the same screen mode but then with palette mode 1, just execute the GSCREEN 20 instruction. The image and palette data are automatically converted from palette mode 0 to palette mode 1. None of the pages will be erased.

Note: To draw in 16-bit YJK directly switch to GSCREEN 11. No clear screen will be performed.

GSCREEN 11

Resolution: 256 × 212 non-interlaced
 256 × 424 interlaced

Colours: 12499 fixed colours, 16-bit YJK (for drawing and displayed)
 16 palette colours out of 512 (displayed only)

Page size: 256 × 256 non-interlaced
 256 × 512 interlaced

Pages available: 8 pages non-interlaced
 4 pages interlaced

Palette mode: 0

Palettes available: --

Imagespace width: 256, 512, 1024 and 2048 dots

Number of sprites: 2

Sprite size: 32 × 32 dots

Sprite colours: 3 + transparent colour

Imagespace layout:



The default imagespace width after a GSCREEN <mode> execution is 512 dots.

To switch to the same screen mode but then with palette mode 1, just execute the GSCREEN 21 instruction. The image and palette data are automatically converted from palette mode 0 to palette mode 1. None of the pages will be erased.

Note: To draw with the 16 palette colours switch to GSCREEN 10. No clear screen will be performed.

GSCREEN 12

Resolution: 256 × 212 non-interlaced
 256 × 424 interlaced
 Colours: 19286 fixed colours, 17-bit YJK
 Page size: 256 × 256 non-interlaced
 256 × 512 interlaced
 Pages available: 8 pages non-interlaced
 4 pages interlaced
 Palette mode: 0
 Palettes available: --
 Imagespace width: 256, 512, 1024 and 2048 dots
 Number of sprites: 2
 Sprite size: 32 × 32 dots
 Sprite colours: 3 + transparent colour

Imagespace layout:



The default imagespace width after a GSCREEN <mode> execution is 512 dots.

Note: There is no palette mode 1 equivalent available for this screenmode. GSCREEN 22 is a 17-bit YUV mode.

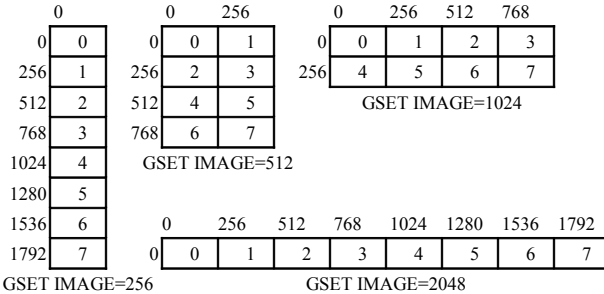
Note: GSCREEN 22 is more suitable for digitized pictures than GSCREEN 12.

Note: In this screenmode the palette is only used for the border colour.

GSCREEN 13

Resolution: 256 × 212 non-interlaced
 256 × 424 interlaced
 Colours: 64 palette colours out of 32768
 Page size: 256 × 256 non-interlaced
 256 × 512 interlaced
 Pages available: 8 pages non-interlaced
 4 pages interlaced
 Palette mode: 1
 Palettes available: 1
 Imagespace width: 256, 512, 1024 and 2048 dots
 Number of sprites: 2
 Sprite size: 32 × 32 dots
 Sprite colours: 3 + transparent colour

Imagespace layout:

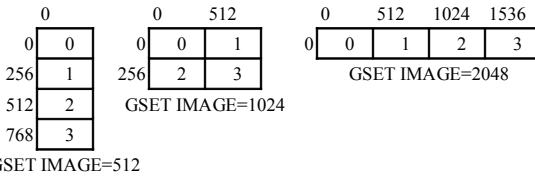


The default imagespace width after a GSCREEN <mode> execution is 512 dots.

GSCREEN 14

Resolution: 512 × 212 non-interlaced
512 × 424 interlaced
Colours: 64 palette colours out of 32768
Page size: 512 × 256 non-interlaced
512 × 512 interlaced
Pages available: 8 pages non-interlaced
4 pages interlaced
Palette mode: 1
Palettes available: 1
Imagespace width: 512, 1024 and 2048 dots
Number of sprites: 2
Sprite size: 32 × 32 dots
Sprite colours: 3 + transparent colour

Imagespace layout:

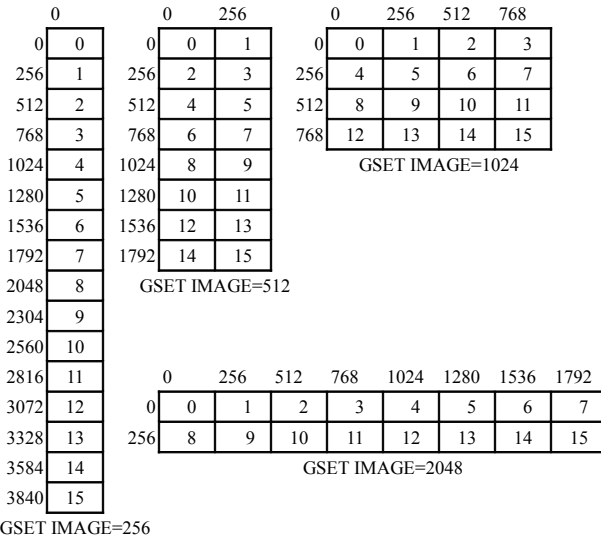


The default imagespace width after a GSCREEN <mode> execution is 1024 dots.

GSCREEN 15

Resolution: 256 × 212 non-interlaced
 256 × 424 interlaced
 Colours: 64 palette colours out of 32768
 Page size: 256 × 256 non-interlaced
 256 × 512 interlaced
 Pages available: 16 pages non-interlaced
 8 pages interlaced
 Palette mode: 1
 Palettes available: 4
 Imagespace width: 256, 512, 1024 and 2048 dots
 Number of sprites: 2
 Sprite size: 32 × 32 dots
 Sprite colours: 3 + transparent colour

Imagespace layout:



The default imagespace width after a GSCREEN <mode> execution is 512 dots.

To switch to the same screen mode but then with palette mode 0, just execute the GSCREEN 5 instruction. The image and palette data are automatically converted from palette mode 1 to palette mode 0. None of the pages will be erased.

GSCREEN 16

Resolution: 512 × 212 non-interlaced
 512 × 424 interlaced
 Colours: 4 palette colours out of 32768
 Page size: 512 × 256 non-interlaced
 512 × 512 interlaced
 Pages available: 16 pages non-interlaced
 8 pages interlaced
 Palette mode: 1
 Palettes available: 4
 Imagespace width: 512, 1024 and 2048 dots
 Number of sprites: 2
 Sprite size: 32 × 32 dots
 Sprite colours: 3 + transparent colour

Imagespace layout:

0	0	0	512	0	1	0	512	1024	1536	0	1	2	3
256	1	256		2	3	256				4	5	6	7
512	2	512		4	5	512				8	9	10	11
768	3	768		6	7	768				12	13	14	15
1024	4	1024		8	9	GSET IMAGE=2048							
1280	5	1280		10	11								
1536	6	1536		12	13								
1792	7	1792		14	15								
2048	8	GSET IMAGE=1024											
2304	9												
2560	10												
2816	11												
3072	12												
3328	13												
3584	14												
3840	15												

GSET IMAGE=512

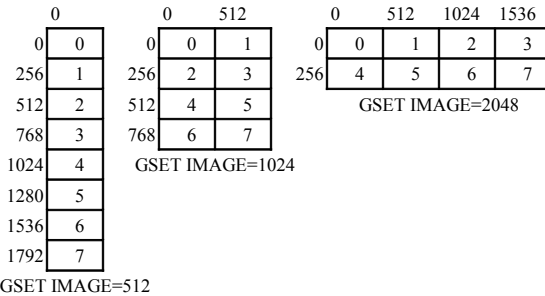
The default imagespace width after a GSCREEN <mode> execution is 1024 dots.

To switch to the same screen mode but then with palette mode 0, just execute the GSCREEN 6 instruction. The image and palette data are automatically converted from palette mode 1 to palette mode 0. None of the pages will be erased.

GSCREEN 17

Resolution: 512 × 212 non-interlaced
512 × 424 interlaced
Colours: 64 palette colours out of 32768
Page size: 512 × 256 non-interlaced
512 × 512 interlaced
Pages available: 8 pages non-interlaced
4 pages interlaced
Palette mode: 1
Palettes available: 4
Imagespace width: 512, 1024 and 2048 dots
Number of sprites: 2
Sprite size: 32 × 32 dots
Sprite colours: 3 + transparent colour

Imagespace layout:



The default imagespace width after a GSCREEN <mode> execution is 1024 dots.

To switch to the same screen mode but then with palette mode 0, just execute the GSCREEN 7 instruction. The image and palette data are automatically converted from palette mode 1 to palette mode 0. None of the pages will be erased.

GSCREEN 18

Resolution: 256 × 212 non-interlaced
 256 × 424 interlaced
 Colours: 256 fixed colours
 Page size: 256 × 256 non-interlaced
 256 × 512 interlaced
 Pages available: 16 pages non-interlaced
 8 pages interlaced
 Palette mode: 1
 Palettes available: 1
 Imagespace width: 256, 512, 1024 and 2048 dots
 Number of sprites: 2
 Sprite size: 32 × 32 dots
 Sprite colours: 3 + transparent colour

Imagespace layout:



The default imagespace width after a GSCREEN <mode> execution is 512 dots.

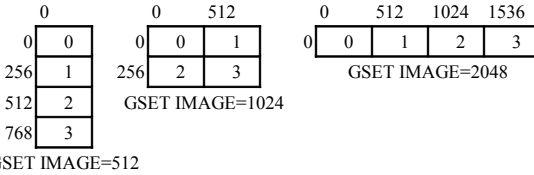
To switch to the same screen mode but then with palette mode 0, just execute the GSCREEN 8 instruction. The image and palette data are automatically converted from palette mode 1 to palette mode 0. None of the pages will be erased.

Note: In this screen mode, the palette is only used for the border colour.

GSCREEN 19

Resolution: 512 × 212 non-interlaced
512 × 424 interlaced
Colours: 256 fixed colours
Page size: 512 × 256 non-interlaced
512 × 512 interlaced
Pages available: 8 pages non-interlaced
4 pages interlaced
Palette mode: 1
Palettes available: 1
Imagespace width: 512, 1024 and 2048 dots
Number of sprites: 2
Sprite size: 32 × 32 dots
Sprite colours: 3 + transparent colour

Imagespace layout:



The default imagespace width after a GSCREEN <mode> execution is 1024 dots.

Note: In this screen mode, the palette is only used for the border colour.

GSCREEN 20

Resolution: 256 × 212 non-interlaced
 256 × 424 interlaced

Colours: 16 palette colours out of 32768 (for drawing and displayed)
 12499 fixed colours, 16-bit YJK (displayed only)

Page size: 256 × 256 non-interlaced
 256 × 512 interlaced

Pages available: 8 pages non-interlaced
 4 pages interlaced

Palette mode: 1

Palettes available: 4

Imagespace width: 256, 512, 1024 and 2048 dots

Number of sprites: 2

Sprite size: 32 × 32 dots

Sprite colours: 3 + transparent colour

Imagespace layout:



The default imagespace width after a GSCREEN <mode> execution is 512 dots.

To switch to the same screen mode but then with palette mode 0, just execute the GSCREEN 10 instruction. The image and palette data are automatically converted from palette mode 1 to palette mode 0. None of the pages will be erased.

Note: To draw in 16-bit YJK directly switch to GSCREEN 21. No clear screen will be performed.

GSCREEN 21

Resolution: 256 × 212 non-interlaced
 256 × 424 interlaced

Colours: 12499 fixed colours, 16-bit YJK (for drawing and displayed)
 16 palette colours out of 32768 (displayed only)

Page size: 256 × 256 non-interlaced
 256 × 512 interlaced

Pages available: 8 pages non-interlaced
 4 pages interlaced

Palette mode: 1

Palettes available: 4

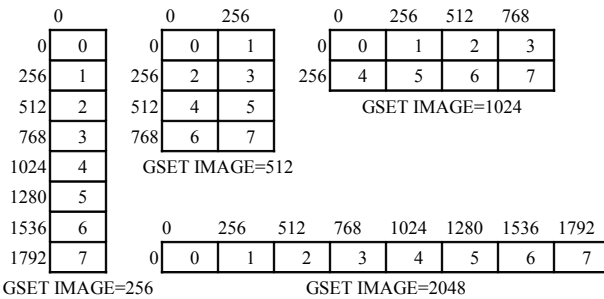
Imagespace width: 256, 512, 1024 and 2048 dots

Number of sprites: 2

Sprite size: 32 × 32 dots

Sprite colours: 3 + transparent colour

Imagespace layout:



The default imagespace width after a GSCREEN <mode> execution is 512 dots.

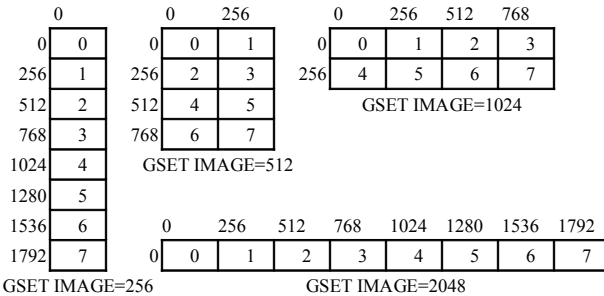
To switch to the same screen mode but then with palette mode 0, just execute the GSCREEN 11 instruction. The image and palette data are automatically converted from palette mode 1 to palette mode 0. None of the pages will be erased.

Note: To draw with the 16 palette colours switch to GSCREEN 20. No clear screen will be performed.

GSCREEN 22

Resolution: 256 × 212 non-interlaced
 256 × 424 interlaced
 Colours: 19286 fixed colours, 17-bit YUV
 Page size: 256 × 256 non-interlaced
 256 × 512 interlaced
 Pages available: 8 pages non-interlaced
 4 pages interlaced
 Palette mode: 1
 Palettes available: 1
 Imagespace width: 256, 512, 1024 and 2048 dots
 Number of sprites: 2
 Sprite size: 32 × 32 dots
 Sprite colours: 3 + transparent colour

Imagespace layout:



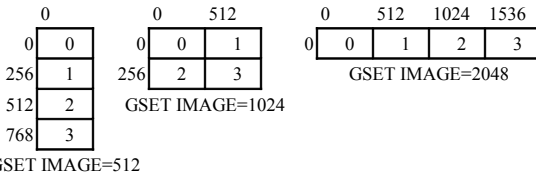
The default imagespace width after a GSCREEN <mode> execution is 512 dots.

Note: In this screenmode the palette is only used for the border colour.

GSCREEN 23

Resolution: 512 × 212 non-interlaced
512 × 424 interlaced
Colours: 19286 fixed colours, 17-bit YJK
Page size: 512 × 256 non-interlaced
512 × 512 interlaced
Pages available: 4 pages non-interlaced
2 pages interlaced
Palette mode: 1
Palettes available: 1
Imagespace width: 512, 1024 and 2048 dots
Number of sprites: 2
Sprite size: 32 × 32 dots
Sprite colours: 3 + transparent colour

Imagespace layout:



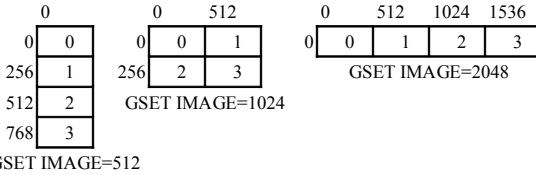
The default imagespace width after a GSCREEN <mode> execution is 1024 dots.

Note: In this screenmode the palette is only used for the border colour.

GSCREEN 24

Resolution: 512 × 212 non-interlaced
512 × 424 interlaced
Colours: 19286 fixed colours, 17-bit YUV
Page size: 512 × 256 non-interlaced
512 × 512 interlaced
Pages available: 4 pages non-interlaced
2 pages interlaced
Palette mode: 1
Palettes available: 1
Imagespace width: 512, 1024 and 2048 dots
Number of sprites: 2
Sprite size: 32 × 32 dots
Sprite colours: 3 + transparent colour

Imagespace layout:



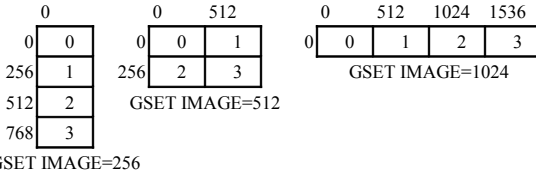
The default imagespace width after a GSCREEN <mode> execution is 1024 dots.

Note: In this screenmode the palette is only used for the border colour.

GSCREEN 25

Resolution: 256 × 212 non-interlaced
256 × 424 interlaced
Colours: 32768 fixed colours
Page size: 256 × 256 non-interlaced
256 × 512 interlaced
Pages available: 4 pages non-interlaced
2 pages interlaced
Palette mode: 1
Palettes available: 1
Imagespace width: 256, 512 and 1024 dots
Number of sprites: 2
Sprite size: 32 × 32 dots
Sprite colours: 3 + transparent colour

Imagespace layout:



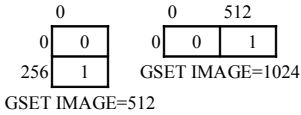
The default imagespace width after a GSCREEN <mode> execution is 512 dots.

Note: In this screenmode the palette is only used for the border colour.

GSCREEN 26

Resolution: 512 × 212 non-interlaced
512 × 424 interlaced
Colours: 32768 fixed colours
Page size: 512 × 256 non-interlaced
512 × 512 interlaced
Pages available: 2 pages non-interlaced
1 pages interlaced
Palette mode: 1
Palettes available: 1
Imagespace width: 512 and 1024 dots
Number of sprites: 2
Sprite size: 32 × 32 dots
Sprite colours: 3 + transparent colour

Imagespace layout:



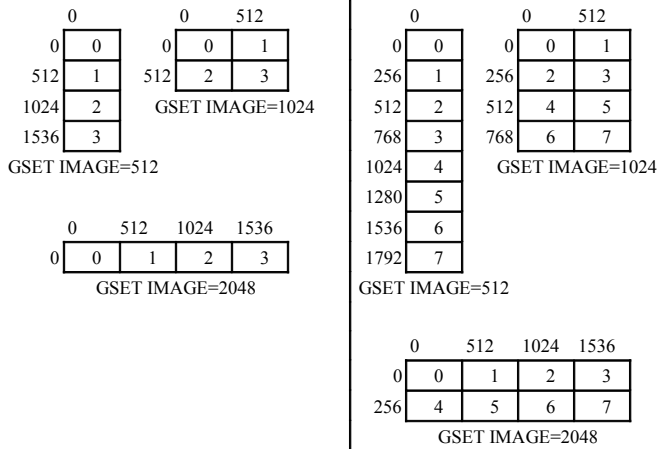
The default imagespace width after a GSCREEN <mode> execution is 512 dots.

Note: In this screenmode the palette is only used for the border colour.

GSCREEN 27

	50Hz	60Hz
Resolution:	384 × 290 non-interlaced 384 × 580 interlaced	384 × 240 non-interlaced 384 × 480 interlaced
Colours:	16 palette colours out of 32768	
Page size:	512 × 512 non-interlaced 512 × 1024 interlaced	512 × 256 non-interlaced 512 × 512 interlaced
Pages available:	4 pages non-interlaced 2 pages interlaced	8 pages non-interlaced 4 pages interlaced
Palette mode:	1	
Palettes available:	4	
Imagespace width:	512, 1024 and 2048 dots	512, 1024 and 2048 dots
Number of sprites:	2	
Sprite size:	32 × 32 dots	
Sprite colours:	3 + transparent colour	

Imagespace layout:

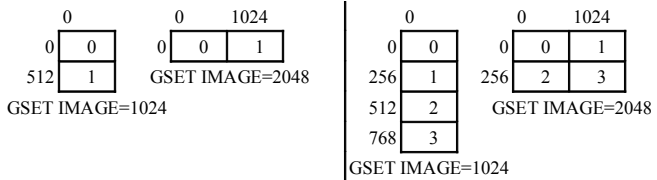


The default imagespace width after a GSCREEN <mode> execution is 1024 dots.

GSCREEN 28

	50Hz	60Hz
Resolution:	768 × 290 non-interlaced 768 × 580 interlaced	768 × 240 non-interlaced 768 × 480 interlaced
Colours:	16 palette colours out of 32768	
Page size:	1024 × 512 non-interlaced 1024 × 1024 interlaced	1024 × 256 non-interlaced 1024 × 512 interlaced
Pages available:	2 pages non-interlaced 1 pages interlaced	4 pages non-interlaced 2 pages interlaced
Palette mode:	1	
Palettes available:	4	
Imagespace width:	1024 and 2048 dots	1024 and 2048 dots
Number of sprites:	2	
Sprite size:	32 × 32 dots	
Sprite colours:	3 + transparent colour	

Imagespace layout:

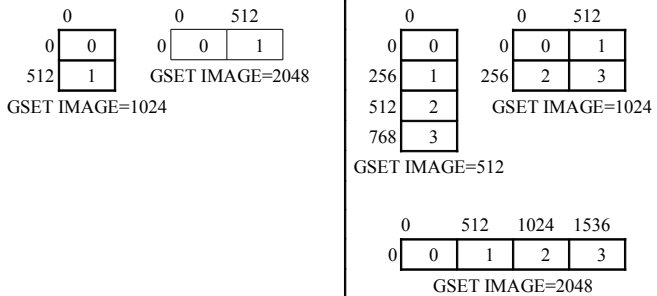


The default imagespace width after a GSCREEN <mode> execution is 2048 dots.

GSCREEN 29

	50Hz	60Hz
Resolution:	384 × 290 non-interlaced 384 × 580 interlaced	384 × 240 non-interlaced 384 × 480 interlaced
Colours:	64 palette colours out of 32768	
Page size:	512 × 512 non-interlaced 512 × 1024 interlaced	512 × 256 non-interlaced 512 × 512 interlaced
Pages available:	2 pages non-interlaced 1 pages interlaced	4 pages non-interlaced 2 pages interlaced
Palette mode:	1	
Palettes available:	1	
Imagespace width:	512 and 1024 dots	512, 1024 and 2048 dots
Number of sprites:	2	
Sprite size:	32 × 32 dots	
Sprite colours:	3 + transparent colour	

Imagespace layout:



The default imagespace width after a GSCREEN <mode> execution is 512 dots.

GSCREEN 30

	50Hz	60Hz
Resolution:	384 × 290 non-interlaced 384 × 580 interlaced	384 × 240 non-interlaced 384 × 480 interlaced
Colours:	256 fixed colours	
Page size:	512 × 512 non-interlaced 512 × 1024 interlaced	512 × 256 non-interlaced 512 × 512 interlaced
Pages available:	2 pages non-interlaced 1 pages interlaced	4 pages non-interlaced 2 pages interlaced
Palette mode:	1	
Palettes available:	4	
Imagespace width:	512 and 1024 dots	512, 1024 and 2048 dots
Number of sprites:	2	
Sprite size:	32 × 32 dots	
Sprite colours:	3 + transparent colour	

Imagespace layout:



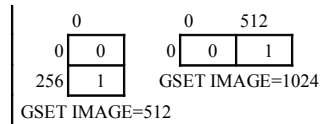
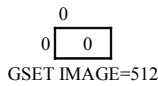
The default imagespace width after a GSCREEN <mode> execution is 512 dots.

Note: In this screenmode the palette is only used for the border colour.

GSCREEN 31

	50Hz	60Hz
Resolution:	384 × 290 non-interlaced 384 × 580 interlaced	384 × 240 non-interlaced 384 × 480 interlaced
Colours:	32768 fixed colours	
Page size:	512 × 512 non-interlaced interlaced not possible	512 × 256 non-interlaced 512 × 512 interlaced
Pages available:	2 pages non-interlaced interlaced not possible	4 pages non-interlaced 2 pages interlaced
Palette mode:	1	
Palettes available:	4	
Imagespace width:	512 dots	512 and 1024 dots
Number of sprites:	2	
Sprite size:	32 × 32 dots	
Sprite colours:	3 + transparent colour	

Imagespace layout:



The default imagespace width after a GSCREEN <mode> execution is 512 dots.

Note: In this screenmode the palette is only used for the border colour.