|  |
| --- |
| agstu |
| VGA IP COMPONENT |
| Software manual for programmers |
|  |
| **Saif Saadaldin** |
| **2021-01-15** |

Innehåll

[1 print\_pix 4](#_Toc61648723)

[1.1 Description 4](#_Toc61648724)

[1.2 Declaration 4](#_Toc61648725)

[1.3 Arguments 4](#_Toc61648726)

[1.4 Return codes 4](#_Toc61648727)

[2 read\_pixel\_ram\_int 5](#_Toc61648728)

[2.1 Description 5](#_Toc61648729)

[2.2 Declaration 5](#_Toc61648730)

[2.3 Arguments 5](#_Toc61648731)

[2.4 Return codes 5](#_Toc61648732)

[3 clear\_screen 6](#_Toc61648733)

[3.1 Description 6](#_Toc61648734)

[3.2 Declaration 6](#_Toc61648735)

[3.3 Arguments 6](#_Toc61648736)

[3.4 Return codes 6](#_Toc61648737)

[4 print\_hline 7](#_Toc61648738)

[4.1 Description 7](#_Toc61648739)

[4.2 Declaration 7](#_Toc61648740)

[4.3 Argument 7](#_Toc61648741)

[4.4 Return codes 7](#_Toc61648742)

[5 print\_vline 8](#_Toc61648743)

[5.1 Description 8](#_Toc61648744)

[5.2 Declaration 8](#_Toc61648745)

[5.3 Argument 8](#_Toc61648746)

[5.4 Return codes 8](#_Toc61648747)

[6 print\_char 9](#_Toc61648748)

[6.1 Description 9](#_Toc61648749)

[6.2 Declaration 9](#_Toc61648750)

[6.3 Argument 9](#_Toc61648751)

[6.4 Return codes 9](#_Toc61648752)

[6.5 More information 9](#_Toc61648753)

[7 print\_circle 10](#_Toc61648754)

[7.1 Description 10](#_Toc61648755)

[7.2 Declaration 10](#_Toc61648756)

[7.3 Argument 10](#_Toc61648757)

[7.4 Return codes 10](#_Toc61648758)

# print\_pix

## Description

Using the HAL IO macro IOWR\_32DIRECT(BASE, OFFSET, DATA) can this function draws one pixel at coordinate (x, y) on VGA monitor. The macros write the DATA to the ram address ((320 \* (y) + x) \* 4).

## Declaration

print\_pix(alt\_u32 x, alt\_u32 y, alt\_u32 color)

## Arguments

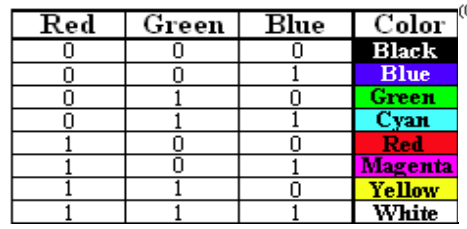
x = 0 to 320  
y = 0 to 240  
color = se figure 1. Use the three bits of the last byte.

Figure 1: color codes

## Return codes

None

# read\_pixel\_ram\_int

## Description

Using the HAL IO macro IORD\_32DIRECT(BASE, OFFSET) can this function reads the data stored in a specific space on the ram.

## Declaration

read\_pixel\_ram\_int(alt\_u32 x, alt\_u32 y)

## Arguments

x = 0 to 320  
y = 0 to 240

## Return codes

Alt\_u32 color data stored at ram

# clear\_screen

## Description

Iterates over all pixels on VGA monitor and draw them with given color

## Declaration

void clear\_screen(alt\_u32 color)

## Arguments

color = se figure 1

## Return codes

None

# print\_hline

## Description

Draws horizontal line with given color, start coordinates (x, y) and line length

## Declaration

void print\_hline(alt\_u32 x, alt\_u32 y, alt\_u32 line\_lenght, alt\_u32 color)

## Argument

x = 0 to 320  
y = 0 to 240  
line\_lenght = 0 to 320  
color = se figure 1

## Return codes

none

# print\_vline

## Description

Draws vertical line with given color, start coordinates (x, y) and line length

## Declaration

void print\_vline(alt\_u32 x, alt\_u32 y, alt\_u32 line\_lenght, alt\_u32 color)

## Argument

x = 0 to 320  
y = 0 to 240  
line\_lenght = 0 to 240  
color = se figure 1

## Return codes

none

# print\_char

## Description

Prints a char that starts at coordinates (x, y) and has a color and background color.

## Declaration

print\_char(alt\_32 x, alt\_32 y, alt\_u8 char\_to\_print, alt\_32 color, alt\_u32 BGcolor)

## Argument

x = 0 to 320  
y = 0 to 240  
char\_to\_print = for example 'A’. Se file “sign\_font.h” for supported chars.   
color = se figure 1  
BGcolor = se figure 1

## Return codes

None

## More information

Some additional information about how a characters are printed on a VGA monitor <https://ece320web.groups.et.byu.net/labs/VGATextGeneration/VGA_Terminal.html>, the link was copied jan/2021.

# print\_circle

## Description

Draws a circle with the specified color, center coordinates (x, y) and radius

## Declaration

void print\_circle(alt\_32 x0, alt\_32 y0, alt\_32 radius, alt\_32 color)

## Argument

x0 = 0 to 320  
y0 = 0 to 240  
radius = 1 to 120   
color = se figure 1

## Return codes

None