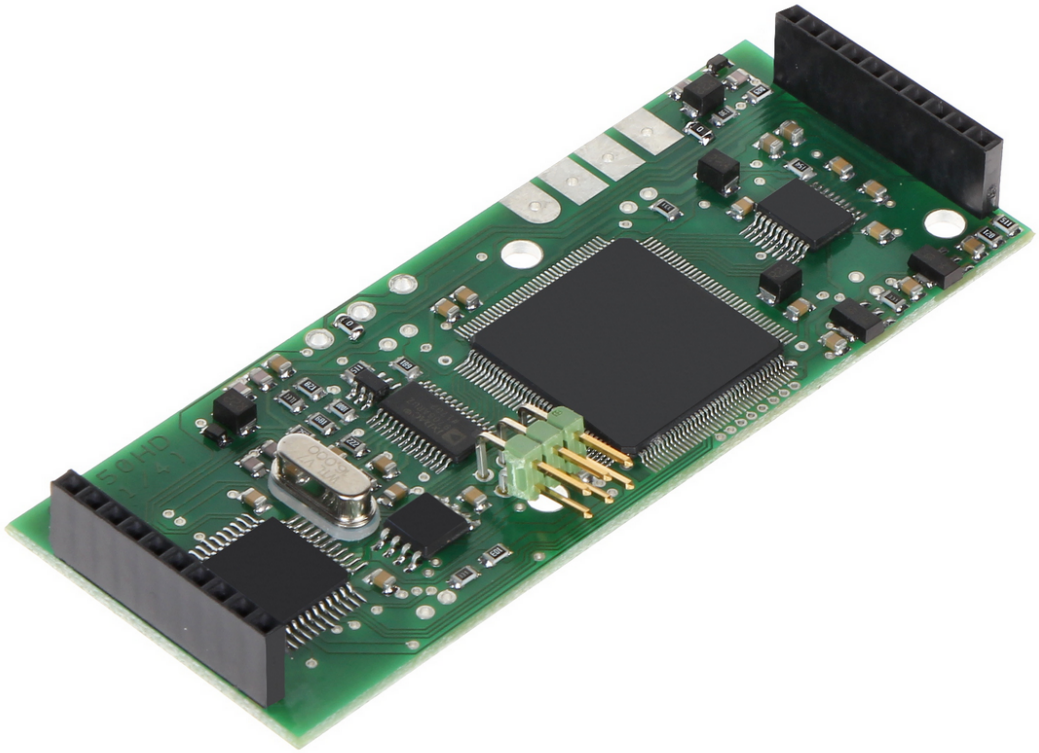
Operating Instructions



www.osd.systems

V.1.34.0 21.12.2018

Table of contents

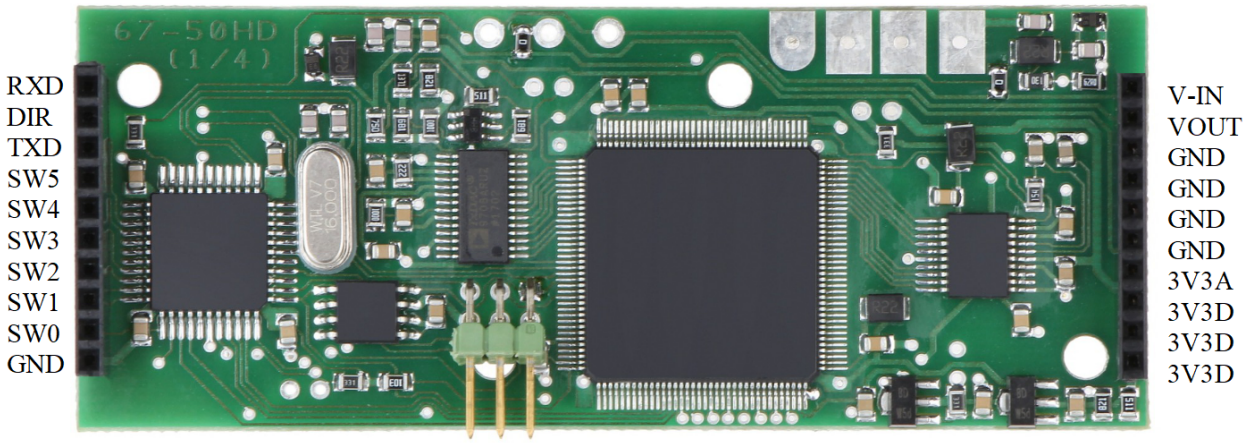
1. Technical data

Standards supported video:	**AHD, HD-CVI HD-TVI** (720p / 1080p) CVBS (D1 / 960H)
Standard Video	PAL / NTSC
Signal Gain Video:	0 dB
Communication / Control:	UART (3.3V)
Transmission speed (B/s)	2400,4800,9600,19200,28800,38400,57600,115200, 234000,468000, 936000, 1872000
Cooperation with other devices:	PORT-22
Power supply:	DC 3.3V / 0.8 A
Weight:	0.02 kg
Dimensions:	91 x 36 x 11 mm

The device is used to impose alphanumeric characters, bitmapped graphics and vector graphics on the camera image analog. Required for the power supply voltage is 3.3V. This is controlled by the port UART (3.3V). The most important features of the:

- Support for Vector Graphics
- Support for Raster Images
- Applying any texts
- The free library for integration with the Arduino
- Compatible with the Arduino / Nucleo (Required module PORT-22)
- The built-in application cross
- Cooperation with weights
- Intuitive menu device
- The ability to change the font and create your own character table graphic
- The ability to update software

2. Description of the pinouts



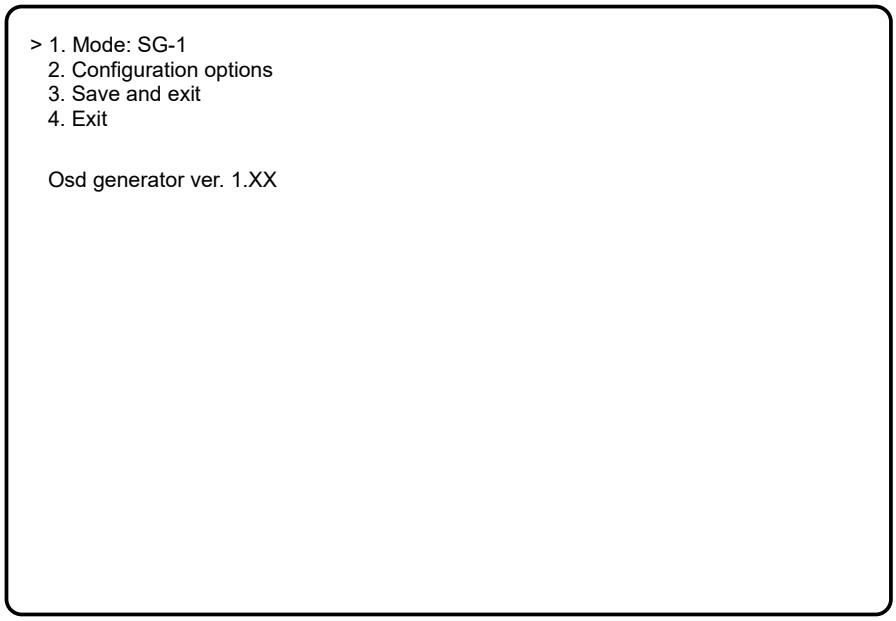
RXD	Line RX UART (3.3V)
DIR	Direction of transmission (3.3V)
TXD	Line TX UART (3.3V)
SW5	Button (bottom)
SW4	Button (top)
SW3	Button (right)
SW2	Button (left)
SW1	Button (SYSTEM)
SW0	Button (Enter) / BOOTLOADER Mode
Vin	Video input
Vout	Video Output
GND	Weight of the unit
3V3A	The power supply of the Analog
3V3D	The power supply of the digital

3. Configuration of the video standard

By default the new device is set in HD-CVI. If the input signal is in another standard video position the correct standard. The following options are available: CVBS, AHD720, AHD1080, HD-CVI720, HD-CVI1080, HD-TVI720, HD-TVI1080. To change the standard use the "SYSTEM". Pressing it or short SW 1 to GND selects the next system. It must be held (short circuit the SW1 to GND) as many times until the screen appear stable subtitles and will this setting compatible with the camera which has been connected. Pressing the "Enter" (SW 0 to GND) opens the menu screen. Navigate between successive menu items using the buttons, change the selected parameter or choosing options you make with "ENTER". Output from the menu screen after you select Exit.

3.1 Menu

To enter the menu use the ENTER button



Mode:

TERMINAL - display frames that are received by the device with regard to mark the new line (NL) and/or (CR),

HEX - display each received the mark in the form of a hexadecimal,

SG-1 - display raster and vector images, strings in any location on the screen,

CT5 - cooperation with a thermometer CT5,

RI-8300 (printer) - printer fiscal Riotech (Graphics mode only raster, without the ability to view the text in the text mode),

8530 cougar - weight supporting standard 8530 cougar e.g.: METTLER TOLEDO 8530,

AXIS - weight and force gauge, Axis

RADWAG - weight Radwag,

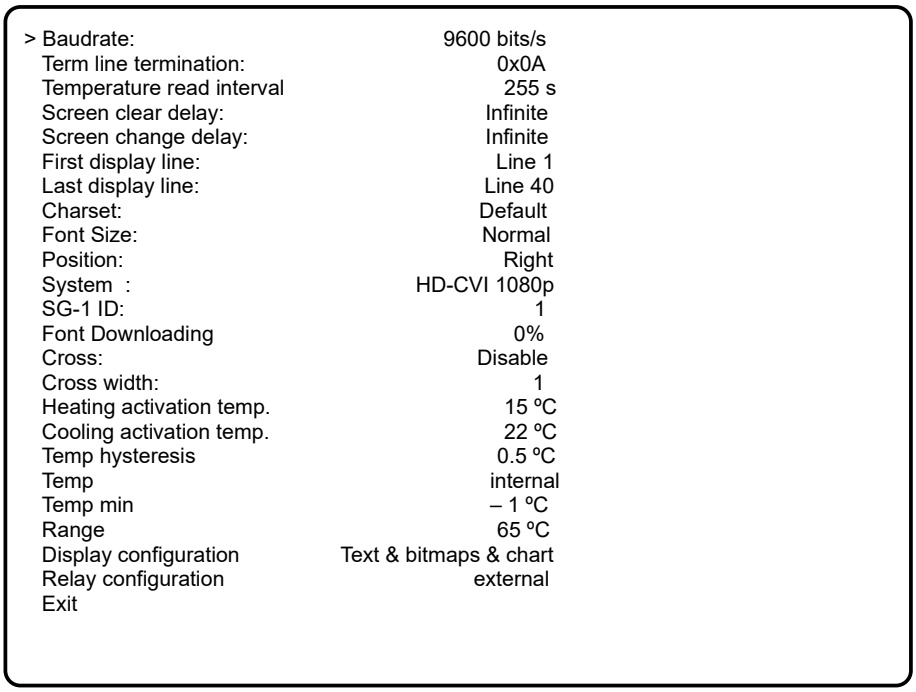
RHEWA Rhewa weight

Configuration options: Entering the detailed configuration options (detailed description - next chapter)

Save and exit - Write configuration and exit the menu,

Exit to exit the OSD menu.

3.2 Configuration Options



In the Configuration menu options the following settings are available:

1. Baudrate - the baudrate of the comunications port, for most devices the default value is 9600 b/s,

2.Term line termination - select a character to the end of the line for the terminal, or no mark the end of the line,

3. Temperature read interval - the frequency of temperature measurements,

4. Screen clear delay - the time after which the screen will be cleaned if no new data appears (1 … 250s or infinite)

5. Screen change delay -(not in SG-1) the time after which the screen will change (0.5s, 1S, 2S, 4s)

6. First display line - Select the first line in which will appear text received by the device option allows you to specify a screen that you will appear data,

7. Last display line - Select the last line in which appears in the text received by the device,

8. Charset - select the code page for letters,

9. Font Size - Select font size. For the CVBS to select only the "normal" and "big". For HD system available "small" option,

10. Position - select the position of the subtitle. In SG-1 moves a fixed distance applying all the objects in the left,

11. System - Choice of standard video.

12. SG-1 ID - The ID number of the device,

13. Font Downloading - the progress of the uploading user character set,

14. Cross - function draws crosshair marker in the the screen

15. Cross width - line thickness markers

16. Heating activation temp. - Relay start temperature for heating devices. Above the set temperature the relay switches off ;

17. Cooling activation temp. - Relay start temperature for cooling devices. Below the set temperature the relay is switched on;

18. Temp - hysteresis hysteresis engagement relay;

19. Temp - choice of configurations for the external or internal (external or internal);

20. Temp min - The lowest value presented in the chart and graphical thermometer;

21. Range - range of temperature values

22. Display configuration - select data presentation:
Text - display of the temperature in the form of text data
Bitmaps - presentation in the form of an analog thermometer
Chart - Drawing Graph
To select there are 4 options: text / text & bitmap / text & bitmaps & chart / text & chart

23. Relay configuration - sensor selection for the thermostat.

24. Exit – exit the menu.

4. Fonts modification

SG-55HD has the ability to change the font and font by the user by creating their own characters. There are two sets of fonts. One used in the menu - this kit fonts cannot be changed. The second set of, automatically loaded is the output from the main menu and can be modified by the user. Edit letters can be done using the free program "Font Editor" available at http://osd.systems .

You can also directly modify the bitmap in any graphics program, for example: Paint.
The image must have a resolution of 170x425 pixels. Color depth of 8 bits. From our side http://osd.systems you can download editable file from the font system (FONTS_16x24_Basic.bmp). Then so created file must be loaded to the device. To upload file .bmp with its own set of fonts used program SG-55 Bootloader.

Font loading:

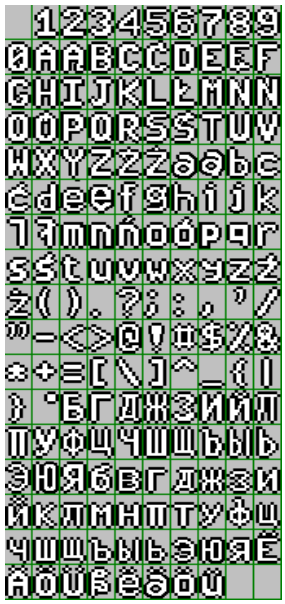
You must connected device to your computer using adapters PORT-22 + USB/RS485. No matter what is currently selected standard or protocol. To load the fonts only needs to enter the configuration menu. In the menu under the heading "Font Downloading" is displayed percentage progress downloaded data.

Instructions for uploading fonts
- Set baudrate in program to the same as in the device
- click on "send fonts" and then select the file with fonts

Note!

- The menu display is always using the original system fonts.
- Uploaded fonts are displayed only after exit from the main menu.
- After updating the software, you must re-write your own character table

Default available characters table (Polish letters, Cyrillic and German diacritical characters):



5. Mode CT5 - temperature measurement

In this mode requires the use of a dedicated the thermometer CT5

Configuration Mode CT5:

16. Heating activation temp. - Relay start temperature for heating devices. Above the set temperature the relay switches off ;

17. Cooling activation temp. - Relay start temperature for cooling devices. Below the set temperature the relay is switched on;

18. Temp - hysteresis hysteresis engagement relay;

19. Temp - choice of configurations for the external or internal (external or internal);

20. Temp min - The lowest value presented in the chart and graphical thermometer;

21. Range - range of temperature values

22. Display configuration - select data presentation:
Text - display of the temperature in the form of text data
Bitmaps - presentation in the form of an analog thermometer
Chart - Drawing Graph
To select there are 4 options: text / text & bitmap / text & bitmaps & chart / text & chart

23. Relay configuration - sensor selection for the thermostat.

6. Mode SG-1 (protocol SG-1)

In this mode you can display raster, vector graphics and strings of text characters anywhere on the screen. The mode allows you to implement your own protocols. The following functions are available:

CLRSCR	0x03 - Cleaning the text on the screen,
TEXT	0x04 - Text display,
LINE	0x05 - drawing any line (diagonal),
CIRCLE	0x06 - drawing circle,
LINEHV	0x07 - drawing horizontal and vertical lines,
WYKRES	0x08 - Drawing graph,
SETWYKRES	0x0A - Define the graph,
SETGRID	0x0B - drawing grid,
UPGRADEBITMAP	0x0C - Display bitmap,
SETBITMAP	0x0D - Settings for bitmap,

The quantity displayed elements is limited. Below are the maximum you can view data elements:
- text (text area 49 x 40 signs)
- The diagonal lines (4)
- Vertical and horizontal lines or filled rectangles (12),
- Wheel or circle (2),
- Net (2),
- graphs (2),
- Bitmaps (2).

Each element is a separate object that can be modified without affecting the other. The grid is composed of multiple line horizontal and vertical. In this case however there is no limit to the number of line. It may be as much as will fit on the screen.

6.1 Clear text on the screen

The frame structure for the cleaning of the screen with the text is as follows:

0xFF, SG-1_ID, 0x03, 0, 0, 0, 0, 0, 0, 0

0xFF - header,
SG -1_ID - The address to which the device is to be cleaned text,
0x03 - Cleaning function screen

6.2 Text display

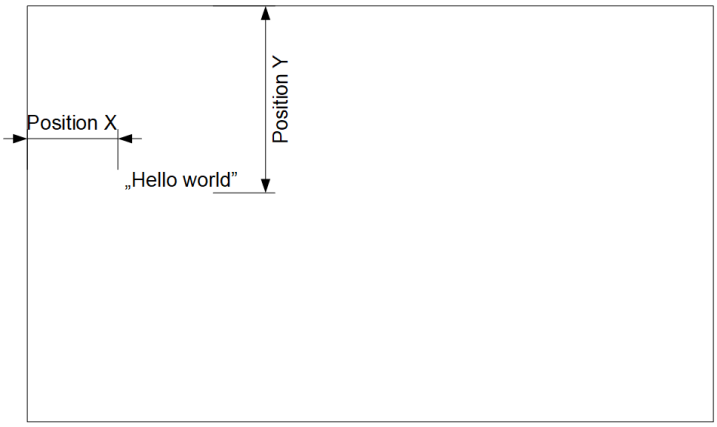
The data frame structure for the display text function is as follows:
0xFF, SG-1_ID, 0x04, POS_X, POS_Y, quantity, 0, 0, 0, 0,<text>,0x03

0xFF – header,
SG -1_ID – address of osd devices,
0x04 – function that displays text
POS_X – X position on the screen (counted in max. 49 characters) ,
POS_Y – Y position on the screen (counted in characters up to 40 for normal and small letters and 22 for large letters),
quantity – number of text characters (number <64),
<tekst> – text in ASCII code
0x03 – end of text character

The interval between the command and the sent text can not be longer than 500ms (auto-timeout),
The appearance of default fonts is shown in the picture:

Example:

0xFF,0x01,0x04,0x05,0x0B,0x0D,0x00,0x00,0x00,0x00,0x22,0x48,0x65,0x6C,0x6C,0x6F,
0x20,0x77,0x6F,0x72,0x6C,0x64,0x22,0x03



0xFF – Header,
0x01 – SG_ID,
0x04 – Text,
0x05 – Position X=5signs,
0x0B – Position Y=11signs,
0x0D – Count of signs,
0x00,0x00,0x00,0x00 - reserved,
0x22,0x48,0x65,0x6C,0x6C,0x6F,0x20,0x77,0x6F,0x72,0x6C,0x64,0x22 – Text,
0x03 – End of string

A sample file text.hex is available for download on the http://osd.systems.

6.3 Line drawing

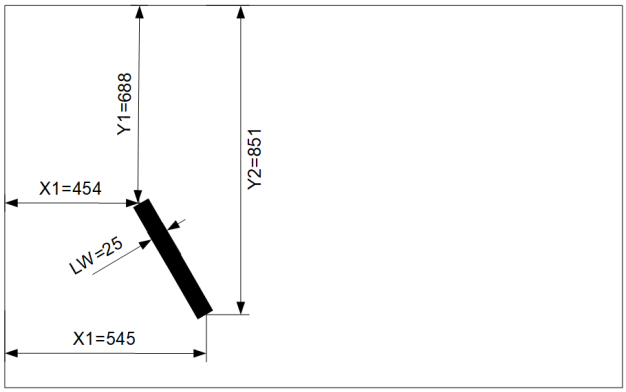
With this function, you can draw lines with a given thickness or a filled rectangle at any angle. The frame structure for the display function of this object is as follows:

0xFF, SG-1_ID, 0x05, 0, LineWidthH, LineWidthL, Y2H, Y2L, X2H, X2L, Y1H, Y1L, X1H, X1L, object, colour, Vsync,0,0.

0xFF – header,
SG -1_ID – address of osd devices,
0x05 – function drawing lines
LineWidthH, LineWidthL – High and Low byte of the line width,
Y2H, Y2L – High and Low byte Y coordinates of the end point line,
X2H, X2L – High and Low byte X coordinates of the end point line,
Y1H, Y1L – High and Low byte Y coordinates of the start point line,
X1H, X1L – High and Low byte X coordinates of the start point line,
object – object number (0 - 3) can be up to 4 objects of this type on the screen.
colour – object colour
Two colors are available: 1 - white, 2 – black, and 0 – transparent (remove line). White objects are always drawn on black.
Vsync – line redrawing only during the vertical sync pulse.

Example:

0xFF,0x01,0x05,0x00,0x00,0x19,0x02,0xB0,0x01,0xC6,0x03,0x53,0x02,0x21,0x00,0x02,
0x01,0x00,0x00



0xFF – header
0x01 – SG-1 ID ,
0x05 – line,
0x00 – reserved,
0x00,0x19 – LW; witdth of the line,
0x02,0xB0 – Y1H, Y1L, Y1=0x02B0=688pixels
0x01,0xC6 – X1H, X1L, X1=0x01C6=454pixels
0x03,0x53 – Y2H, Y2L, Y2=0x0353=851pixels
0x02,0x21 – X2H, X2L, X2=0x0221=545pixels
0x00 – object,
0x02 – colour,
0x01 – Vsync,
0x00,0x00 – reserved.

A sample file line.hex is available for download on the http://osd.systems.

6.4 Circle drawing

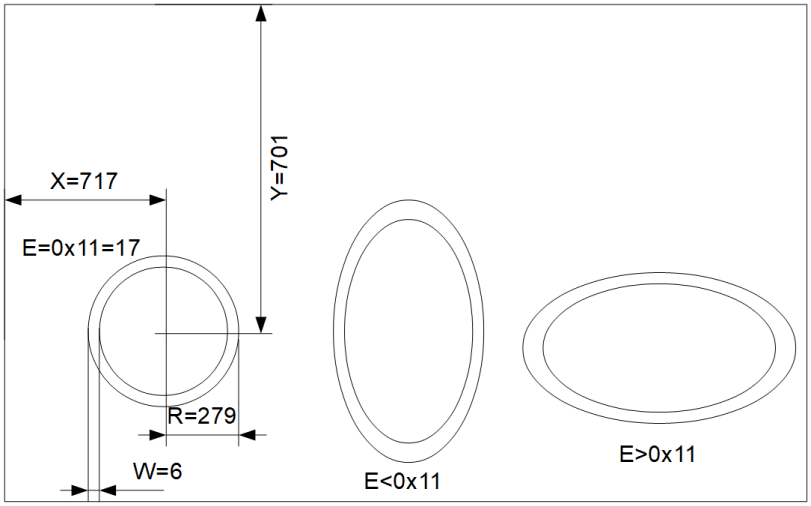
With this function you can draw circles. The frame structure for the display function of this object is as follows:

0xFF, SG-1_ID, 0x06, 0, WidthH, WidthL, YH, YL, XH, XL, radixH, radixL, 0, 17, colour, object, Vsync,0,0,0

0xFF – header,
SG -1_ID – address of osd devices,
0x06 – function drawing a circle
WidhtH, WidhtL – High and Low byte of the width line.
YH, YL – High and Low byte the Y coordinates of center point,
XH, XL – High and Low byte the X coordinates of center point,
radixH, radixL – High and Low byte of the radius
object – object number (8 - 9) can be up to 2 objects of this type on the screen,
colour *– object colour,*
Two colors are available: 1 - white, 2 - black. If we want an object to disappear from the screen, we must draw it in color 0 (transparent). White objects are always drawn on black.
Vsync – line redrawing only during the vertical sync pulse.

Example:

0xFF,0x01,0x06,0x00,0x00,0x06,0x02,0xBD,0x02,0xCD,0x01,0x17,0x00,0x11,0x01,0x09,0x01
0x00,0x00,0x00



0xFF – Header
0x01 – SG_ID,
0x06 – Circle,
0x00 – reserved,
0x00,0x06 – WH, WL, W=0x0006=6pixels
0x02,0xBD – YH, YL, Y=0x02BD=701pixels
0x02,0xCD – XH, XL, X=0x02CD=717pixels
0x01,0x17 – RH, RL, R=0x0117=279pixels
0x00 – reserved,
0x11 – E,
0x01 – colour,
0x09 – object,
0x00 - Vsync
0x00,0x00 – reserved.

Example of the circle.hex is available for download on the http://osd.systems

6.5 Horizontal or vertical line drawing

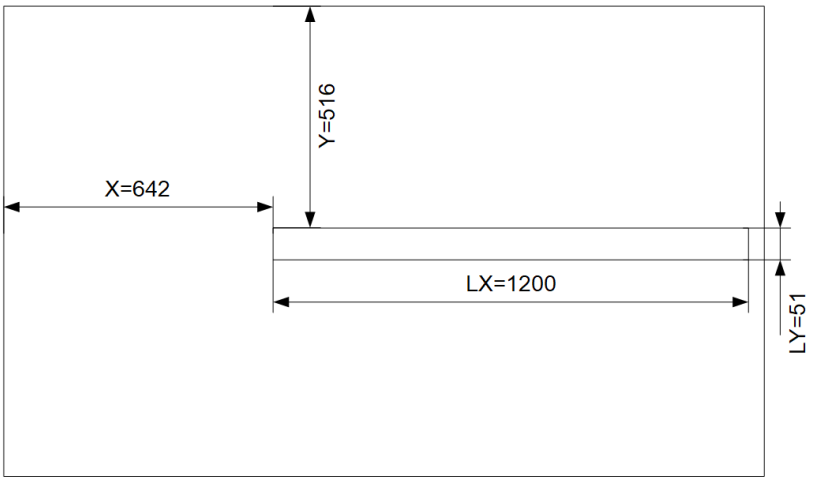
With this function you can draw lines with a given thickness or a filled rectangle but only vertical or horizontal.
The frame structure for the display function of this object is as follows:

0xFF, SG-1_ID, 0x07, 0 , 0, color, LYH, LYL, LYH, LYL, YH, YL, XH, XL, object, Hsync, 0, 0, 0

00xFF - header,
SG -1_ID - The address to which the device is to appear (set on the device,
0x07 - function rysująca HV lines.
Color - color line,
Two colors are available: 1 - white, 2 - black. If we want an object to disappear from the screen, we must draw it in color 0 (transparent). White objects are always drawn on black.
LYH, LYL - older and younger byte height line (completed rectangle),
LXH, LXL - older and younger byte height line (completed rectangle)
YH, YL - older and younger byte Y coordinate the beginning of the line,
XH, XL - older and younger byte X coordinates the beginning of the line,
object – object number (12 - 19) can be up to 8 objects of this type on the screen,
Vsync – line redrawing only during the vertical sync pulse.

Example:

0xFF,0x01,0x07,0x00,0x00,0x01,0x00,0x33,0x04,0xB0,0x02,0x04,0x02,0x82,0x11,0x01,
0x00,0x00,0x00



0xFF – Header
0x01 – SG_ID,
0x07 – LineHV,
0x00 – reserved,
0x00 – reserved,
0x01 – colour,
0x00,0x33 – LYH, LYL, LY=0x0033=51pixels,
0x04,0xB0 – LXH, LXL, LX=0x04B0=1200pixels,
0x02,0x04 – YH, YL, Y=0x0204=516pixels,
0x02,0x82 – XH, XL, X=0x0282=642pixels,
0x11 – object
0x01 – Vsync,
0x00,0x00,0x00 – reserved.

A sample file LineHV.hex is available for download on the http://osd.systems.

6.6 Grid drawing

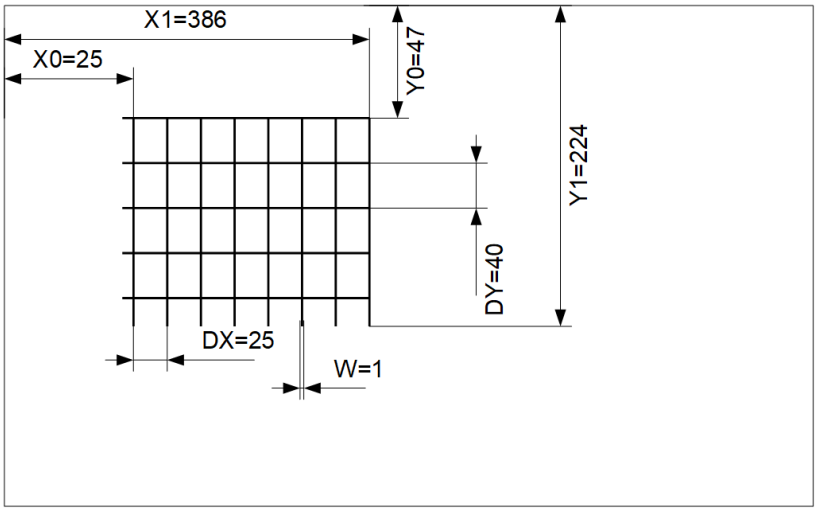
With this function, you can draw a grid of horizontal and vertical lines with adjustable spacing, line thickness and the area to occupy on the screen.The grid consists of many horizontal and vertical lines. However, in this case there is no limit to the number of lines. It can be as much as it can fit on the screen.
The frame structure for the grid display function is as follows:

0xFF, SG-1_ID, 0x0B, 0 , color, LineWidth, DYH, DYL, DXH, DXL, Y1H, Y1L, X1H, X1L, Y0H, Y0L, X1H, X1L, object.

0xFF - header,
SG -1_ID - address of osd devices,
0x0B - function grid drawing
color - color line,
LineWidth - thickness of the grid lines
DYH, DYL – high and low byte distance between the lines in the Y axis,
DXH, DXL – high and low byte byte distance between the lines in the X axis,
Y1H, Y1L – high and low byte byte Y coordinate of the lower right corner of the grid,
X1H, X1L – high and low byte byte X coordinate of the lower right corner of the grid.
Y0H, Y0L – high and low byte byte Y coordinate of the upper left corner of the grid,
X0H, X0L – high and low byte byte X coordinate of the upper left corner of the grid,
object - object number (30-31), the maximum may be 2 objects of this type on the screen.

Example:

0xFF,0x01,0x0B,0x00,0x01,0x01,0x00,0x28,0x00,0x12,0x00,0xE0,0x01,0x82,0x00,0x2F,
0x00,0x19,0x1F



0xFF – Header
0x01 – SG_ID,
0x0B – SetGrid,
0x00,
0x01 – colour , white = 1,
0x01 – W = 1
0x00,0x28 – DYH, DYL, DY=0x0028=40pixels
0x00,0x12 – DXH, DXL, DX=0x0012=25pixels
0x00,0xE0 – Y1H, Y1L, Y1=0x00E0=224pixels
0x01,0x82 – X1H, X1L, X1=0x0182=386pixels
0x00,0x2F – Y0H, Y0L, Y0=0x002F=47pixels,
0x00,0x19 – X0H, X0L, X0=0x0019=25pixels
0x1F – object.

A sample file SetGrid.hex is available for download on the http://osd.systems.

6.7 Plot settings

With this function, we define the area on which we will draw a graph.
The frame structure for the graph settings functions is as follows:

0xFF, SG-1_ID, 0x0A, 0, 0, colour, LY, LX, YH, YL, XXH, XXL, object, 0, 0, 0, 0

0xFF – header
SG-1_ID – address of osd devices,
0x0A – plot settings,
colour – object colour,
LY – scale on the Y axis, the increase in the parameter increases the graph. Maximum value 255
LX – length of the graph expressed in multiples of 16. For example: 10 means 160 pixels.
YH, YL - High and Low byte of the coordinate of the upper edge of the chart.
XXH, XXL - High and Low byte of the parameter specifying where on the X axis is the end of the chart,
XX = X + 16 * LX,
X, Y - coordinates of the upper left corner of the chart,
object – object number (28 - 29) can be up to 2 objects of this type on the screen,

Example:

0xFF,0x01,0x0A,0x00,0x00,0x01,0x14,0x16,0x00,0x55,0x01,0x78,0x1C,0x00,0x00,0x00,0x00,0x00,0x00

0xFF – Header,
0x01 – SG_ID,
0x0A – SetPlot,
0x00, 0x00 – reserved,
0x01 – colour,
0x14 – SY=0x14=20
0x16 – DX=0x16*0x10=22*16=352pixels
0x00,0x55 – YH, YL, Y=0x0055=85pixels
0x01,0x78 – XH, XL, X=0x0178=376pixels
0x1C – object,
0x00,0x00,0x00,0x00,0x00,0x00 – reserved,

A sample file SetPlot.hex is available for download on the http://osd.systems.

6.8 Graph drawing

Using this feature, enter data into the memory to draw the graph.
The frame structure for the graph drawing function is as follows:

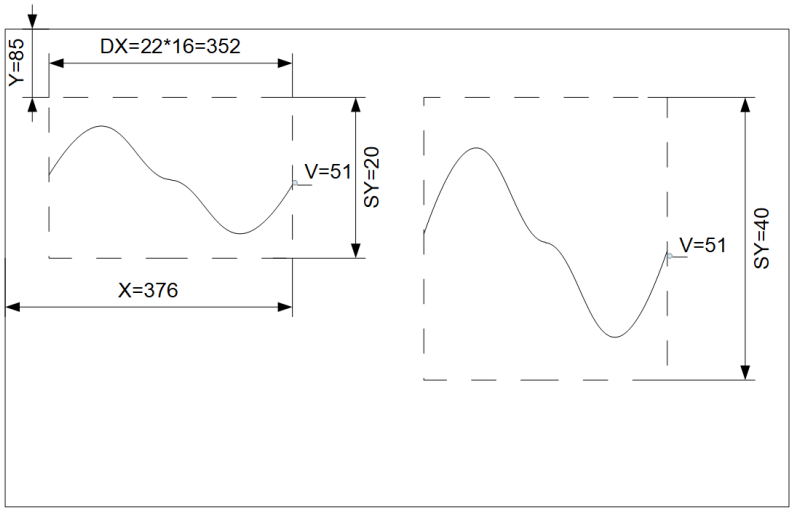
0xFF, SG-1_ID, 0x08, 0, plot, valueH, valueL.

0xFF – header
SG-1_ID – address of osd devices,
0x08 – plot drawing function,
plot – number of plot (0 – 1),
valueH, valueL – High and Low byte of data plot. This value is 8 bits (0..255),

Example:

0xFF,0x01,0x08,0x00,0x00,0x00,0x33

0xFF – Header,
0x01 – SG_ID,
0x08 – UpdatePlot,
0x00 – reserved,
0x00 – number of plot,
0x00 – reserved,
0x33, V=0x33=51.



An example of the file Update.hex (display one value : 51) is available for download on the http://osd.systems.

6.9 Bitmap display

6.9.1 How to prepare the bitmap to display

Each bitmap image can be created in any image program, but in order to facilitate the

We create the bitmap in any image program such as "Paint". The image must have a specific resolution and the number of colors. The width must be multiples of 32. The bitmap file may not be greater than 40960 pixels . The image at the end of the save editing a mirror reflection. The following options are available:

dim	x	y
0	32	1280
1	64	640
2	96	426
3	128	320
4	160	256
5	192	213
6	224	182
7	256	160
8	288	142
9	320	128

dim	x	y
10	352	116
11	384	106
12	416	98
13	448	91
14	480	85
15	512	80
16	544	75
17	576	71
18	608	67
19	640	64

dim	x	y
20	672	60
21	704	58
22	736	55
23	768	53
24	800	51
25	832	49
26	864	47
27	896	45
28	928	44
29	960	42

dim	x	y
30	992	41
31	1024	40
32	1056	38
33	1088	37
34	1120	36
35	1152	35
36	1184	34
37	1216	33
38	1248	32
39	1280	32

6.9.2 Osd Bitmap Editor

For the preparation of the bitmap preferably use a program Osd Bitmap Editor that is available in the tab (Tab Tools > Osd Bitmap editor). The available functions:

Choose bmp - Open file (bitmap selection),

Center DIM - centering frame bitmap,

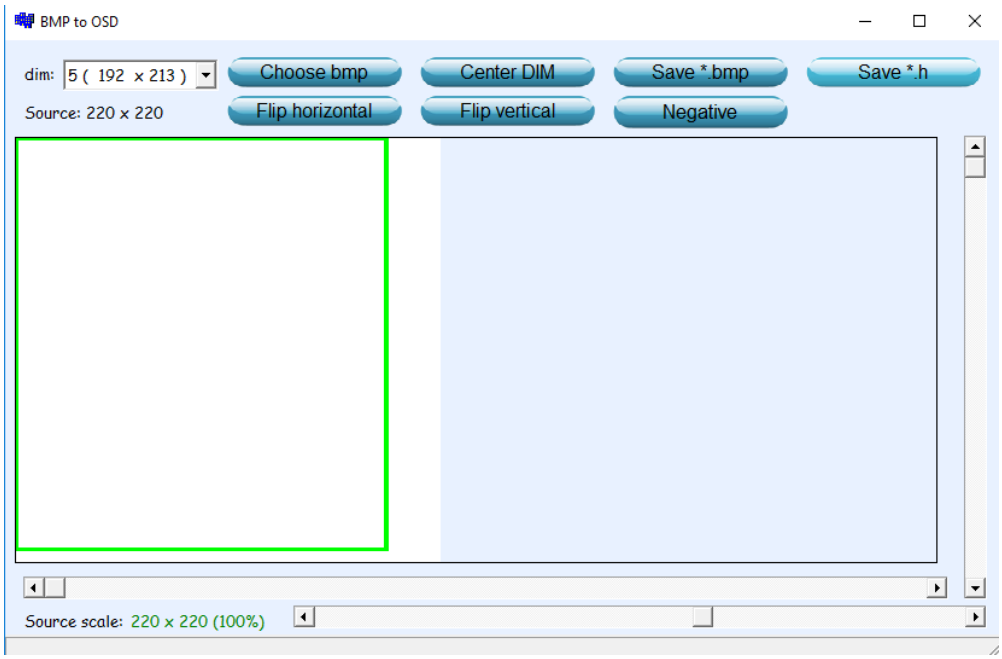
Save *.bmp - Recording bitmap format monochrome,

Save the *.h - saving to the file header.content EEPROM for the processor Atmega328,

Flip Horizontal - rotate the image horizontally,

Flip Vertical - rotate the image vertically,

Negative - bit reversal color.



6.9.3 Smooth drawing graphics

You can enable an additional option to increase the smoothness of the display to draw moving lines and circles. This function is called Vsync and consists in drawing objects into pulses of vertical synchronization of the video signal. After sending a drawing command with the Vsync option enabled, wait 40 ms before sending the next command.

6.9.4 Bitmap settings

The device allows you to display a single-color bitmap. To make this possible, the appropriate conditions must be met. Before sending a bitmap to the device, you first need to define the parameters of this image. Data to send: position of the image on the screen, parameter "dim" describing the bitmap proportions, scale X and Y, color. The frame structure for the text display function is as follows:

0xFF, SG-1_ID, 0x0D, 0, 0, dim_colour, YH, YL, XH, XL, SY, SX, 0, 0, object, 0, 0, 0, 0

0xFF – header
SG-1_ID – address of osd devices,
0x0D – a function that prepares a bitmap for display.
dim_colour = **dim** + **colour***64 ,
dim – value from table,
colour – object colour,
YH, YL – High and Low byte Y coordinates upper left point of bitmap,
XH, XL – High and Low byte X coordinates upper left point of bitmap,
LY – Y axis scale,
LX – X axis scale,
Value 255 - an unscaled object. Decreasing the LX, LY value extends the bitmap in the appropriate axis.
object – object number (38 - 39) can be up to 2 objects of this type on the screen,

Example:

0xFF,0x01,0x0D,0x00,0x00,0x42,0x02,0x67,0x00,0x23,0xFF,0xFF,0x00,0x00,0x26,0x00,
0x00,0x00,0x00

0xFF – Header,
0x01 – SG_ID,
0x0D – SetBitmap,
0x00,0x00 – reserved,
0x42 => *colour* * *0x40* + *dim* = *1* * 0x40 + 2 = 0x42,
0x02,0x67 – YH, YL, Y=0x0267=615pixels,
0x00,0x23 – XH, XL, X=0x0023=35pixels,
0xFF – SY,
0xFF – SX,
0x00,0x00 – reserved,
0x26 – obiect,
0x00,0x00,0x00,0x00 – reserved,

An example of the file UpdateBitmap.hex (display one value : 51) is available for download on the http://osd.systems.

6.9.5 Wyświetlenie bitmapy

With this function we display a bitmap

The frame structure for the bitmap display function is as follows:

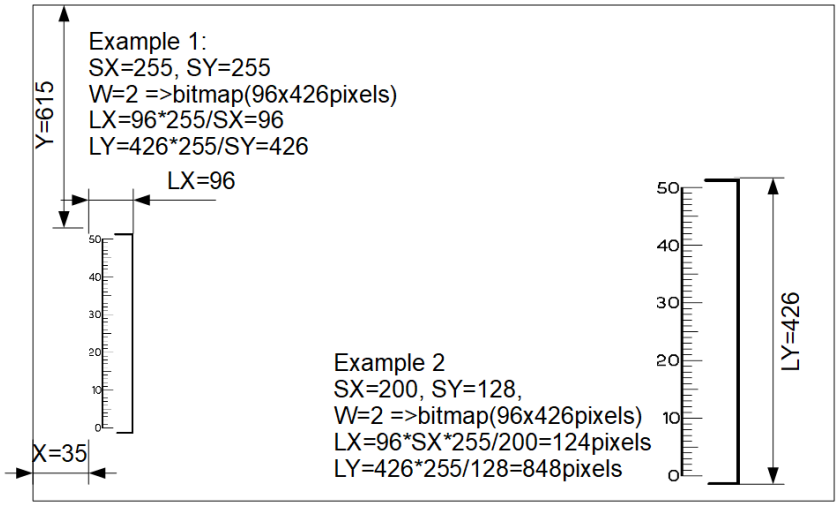
0xFF, SG-1_ID, 0x0C, 0, adresH, adresL, LenghtL, LenghtH, nrBmp, data.

0xFF – nagłówek,
SG -1_ID – adres, na którym urządzeniu ma pojawić się napis (ustawiany w urządzeniu),
0x0C – funkcja rysująca bitmapę w pamięci urządzenia,
adresH, adresL – low and high bytes of the address from which we want to draw a bitmap. If the address = 0, then we draw the whole bitmap,
LenghtL, LenghtH – low and high bytes of the amount of data to send from the address indicated ,
nrBmp – numer bitmapy, którą aktualizujemy (0, 1)
data – zadeklarowana ilość danych informacji o pikselach.

Dostępne są dwa kolory: 1 – biały, 2 – czarny. Jeśli chcemy, żeby jakiś obiekt zniknął z ekranu musimy go narysować w kolorze 0 (przezroczysty). Obiekty białe zawsze są rysowane na czarnych. Nigdy odwrotnie.

Example:

0xFF,0x01,0x0C,0x00,0x00,0x00,0xF8,0x13,0x00,0xFF,0xFF,0xFF,0xFF,0xFF,0xFF,0xFF,
0xFF … 0xFF,0xFF,0x07,0x00,0x00,0x00,0x00.



0xFF – Header,
0x01 – SG_ID,
0x0C – UpdateBitmap,
0x00 – reserved,
0x00,0x00 – address,
0xF8,0x13 – LenghtL, LenghtH, Lenght of file = 0x13F8= 5112 bajts = 5112*8 = 40896 pixels
0x00 – number of bitmap,
0xFF,0xFF,0xFF,0xFF,0xFF,0xFF,0xFF,0xFF … 0xFF,0xFF,0x07,0x00,0x00,0x00,0x00 – data.

7. Integration with the Arduino

The OSD-50HD can be easily integrated with other systems. The easiest way to implement your own solutions is to use the PORT-22 module. The PORT-22 module is compatible with Arduino Nano and ST / Nucleo.

There is also a free library for Arduino that supports all OSD-50HD features. The library can be downloaded from the manufacturer's website: https://osd.systems
The following functions are available:

SendText() - Sending text to the screen,

SendLineHV() - drawing a vertical, horizontal or diagonal line of a given thickness, position, vertical synchronization and color,

UpdateBitmap() - change the bitmap content,

SetBitmap() - Bitmap parameters setting: position, "dim" parameter describing bitmap proportions, X and Y scale, color,
SendGrid() - drawing a grid with a given drawing area, the distance between the X and Y lines, the thickness of the line,
SetPlot() - setting the parameters of the chart: position, size, color,

UpdatePlot() - plot update,
SendCircle() - drawing a circle with a given position, radius and vertical synchronization,

SendLine() - drawing a vertical, horizontal or filled rectangle line with a given position, sizes and vertical synchronization.

7.1 SendText

SendText (char iID, unsigned char iX, unsigned char iY, char* str);

SendText(ID, X, Y, text)

ID – device address,
X, Y – coordinates of start point,
text - ASCII string of characters

example:
SendText(1, 5, 5, "Hello world!");

7.2 SendLineHV

This function draws horizontal or vertical line. Also dwars filled rectangles. 12 pcs object(12..23)

SendLineHV(char iID, unsigned short X, unsigned short Y, unsigned short LX, unsigned short LY, byte colour, byte Vsync, byte object)

SendLineHV(ID, X, Y, lenght_X, lenght_Y, colour, Vsync, object)

ID – device address (default 1)
X,Y - Position of top left corner.
lenght_X, lenght_Y - lenght and width of the object,
colour - 0 - none, 1 - white, 2 - black
Vsync - 1 - Waiting for vertical synchronization impulse before drawing.
Object – number of object (12 … 23)

Example:
SendLineHV(1, 720, 20, 190, 9, 1, 1, 12);

7.3 SendGrid

SendGrid(char iID, unsigned short X1, unsigned short Y1, unsigned short X2, unsigned short Y2, byte LineWidth, unsigned short DistanceX, unsigned short DistanceY, byte colour, byte object)

SendGrid(ID,leftup_X, leftup_Y, rightdown_X, rightdown_Y, line_width, distance_X, distance_Y, colour, object)

ID – device address,
leftup_X, leftup_Y – coordinates of left up grid edge point,
rightdown_X, rightdown_Y – coordinates of right down edge point,
line_width - width of the line,
distance_X – distance between horizontal lines,
distance_Y – distance between vertical lines,
colour – colour: 0 - none, 1 - white, 2 – black,
object – number of object,

Example:
SendGrid(1, 20, 700, 700, 950, 1, 20, 20, 1, 30);

7.4 SetBitmap

SetBitmap(char iID, unsigned short X, unsigned short Y, byte LX, byte LY, byte dim, unsigned short styl, byte colour, byte object)

SetBitmap(ID, X, Y, scale_X, scale_Y, dim, style, colour, object)

ID – device address,
X, Y – coordinates of left up bitmap
scale_X, scale_Y – picture scale (value 255 – picture in real format),
dim – value from resolution table,
style – must be set 0,
colour – 0 - none, 1 - white, 2 – black
object – number of display bitmap

7.5 UpdateBitmap

UpdateBitmap(char iID, int NrBmp, int adrress, uint16_t DataCount, byte *data)

UpdateBitmap(ID, object, address, count, data)

ID – device address,
object – number of display bitmap
address – addres of bitmap memory
count – quantity of display data
data – pointer to bitmap data

7.6 SetPlot

SetPlot(char iID, unsigned short X, unsigned short Y, byte LX, byte LY, byte styl, byte colour, byte object);

SetPlot(ID, X, Y, LX, LY, style, colour, object);

ID – device address,
X, Y – coordinates of left up bitmap
LX – plot lenght. Lenght = LX * 16 pixels
LY – plot height in pixels
style – option not active
colour – 0 - none, 1 - white, 2 – black
object – number of set plot (0 or 1)

7.7 UpdatePlot

UpdatePlot(char iID, byte plot, unsigned short value)

UpdatePlot(ID, object, value)
ID – device address,
object – number of set plot (0 or 1),
value – single value of Y axis.
Po każdym wprowadzeniu wartości następuje przesunięcie wykresu o jeden pomiar w lewo.

7.8 SendCircle

SendCircle(char iID, unsigned short X, unsigned short Y, unsigned short radixX, unsigned short radixY, unsigned short width, byte colour, byte Vsync, byte object)

SendCircle (ID, X, Y, 0, radius, width, colour, Vsync, object)

ID – device address,
X, Y – coordinates of left up bitmap,
radius – lenght of radius,
widthd – width of line,
colour - 0 - none, 1 - white, 2 – black,
Vsync - 1 - Waiting for vertical synchronization impulse before drawing,
object – number of object (8 … 9).

example:
SendCircle(500, 770, 0, 100, 8, 1, 0, 8);

7.9 SendLine

SendLine(char iID, unsigned short X1, unsigned short Y1, unsigned short X2, unsigned short Y2, unsigned short LineWidth, byte colour, byte Vsync, byte object);

SendLine(ID, X1, Y1, X2, Y2, width, colour, Vsync, object)

ID – device address,
X1, Y1 – coordinates start point of line,
X2, Y2 – coordinates end point of line,
width – width of line,
colour – 0 - none, 1 - white, 2 – black,
Vsync - 1 - Waiting for vertical synchronization impulse before drawing,
object – number of set plot (0 or 3).

8. Firmware update

The device has the ability to update the software. Entering the update mode is done by holding the ENTER button while connecting the device's power supply. The current version of the software and a description of the update procedure are available at http://osd.systems