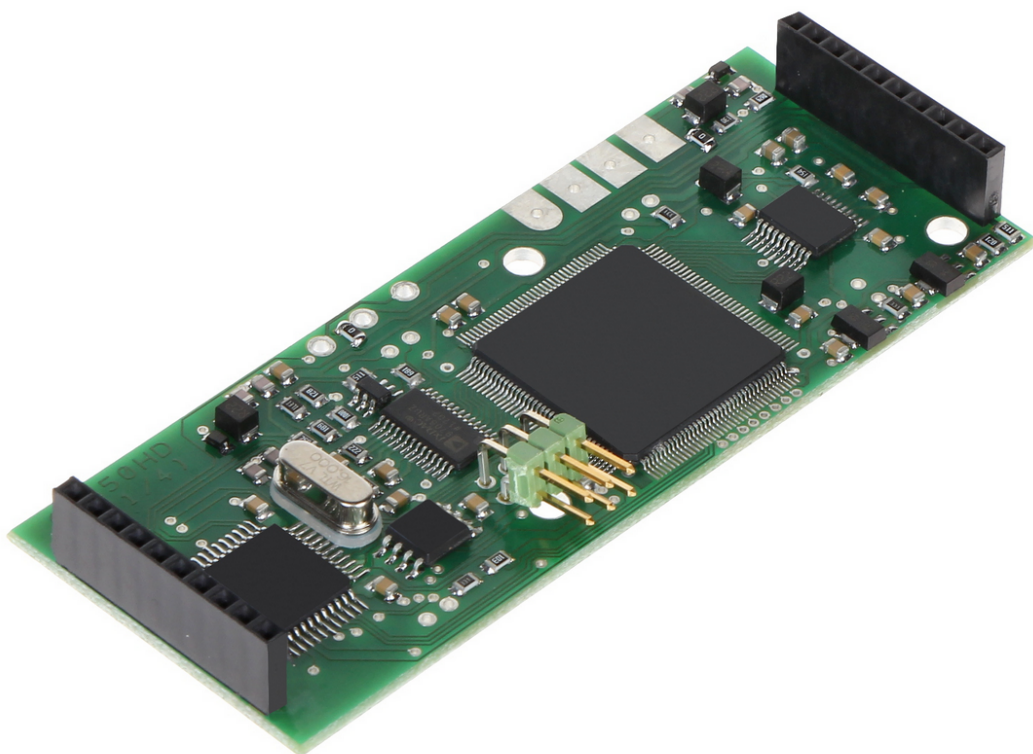


INSTRUKCJA OBSŁUGI



www.osd.systems

v.1.34.0 21.12.2018

Spis treści

1. Dane techniczne.....	3
2. Opis wyprowadzeń.....	4
3. Konfiguracja standardu wideo.....	4
3.1 Menu urządzenia.....	5
3.2 Ustawienia urządzenia - Configuration options.....	6
4. Modyfikacja fontów.....	8
5. Tryb CT5 – pomiar temperatury.....	9
6. Tryb SG-1 (protokół SG-1)	10
6.1 Czyszczenie tekstu na ekranie.....	11
6.2 Wyświetlanie tekstu.....	12
6.3 Rysowanie linii.....	13
6.4 Rysowanie okręgu.....	14
6.5 Rysowanie linii HV.....	15
6.6 Rysowanie siatki.....	16
6.7 Ustawienia wykresu.....	17
6.8 Rysowanie wykresu.....	18
6.9 Wyświetlanie bitmapy	19
6.9.1 Sposób przygotowania bitmapy do wyświetlenia.....	19
6.9.2 Osd Bitmap Editor.....	20
6.9.3 Płynność rysowania poruszającej się grafiki.....	20
6.9.4 Ustawienia bitmapy 0x0D.....	21
6.9.5 Wyświetlenie bitmapy 0x0C.....	22
7. Integracja z Arduino.....	23
7.1 Wyświetlanie tekstu.....	23
7.2 Rysowanie linii H V.....	24
7.3 Rysowanie siatki.....	24
7.4 Definiowanie bitmapy.....	25
7.5 Wyświetlanie bitmapy.....	25
7.6 Ustawienia wykresu.....	25
7.7 Rysowanie wykresu.....	26
7.8 Rysowanie okręgu.....	27
7.9 Rysowanie linii.....	27
8. Aktualizacja oprogramowania.....	27

1. Dane techniczne

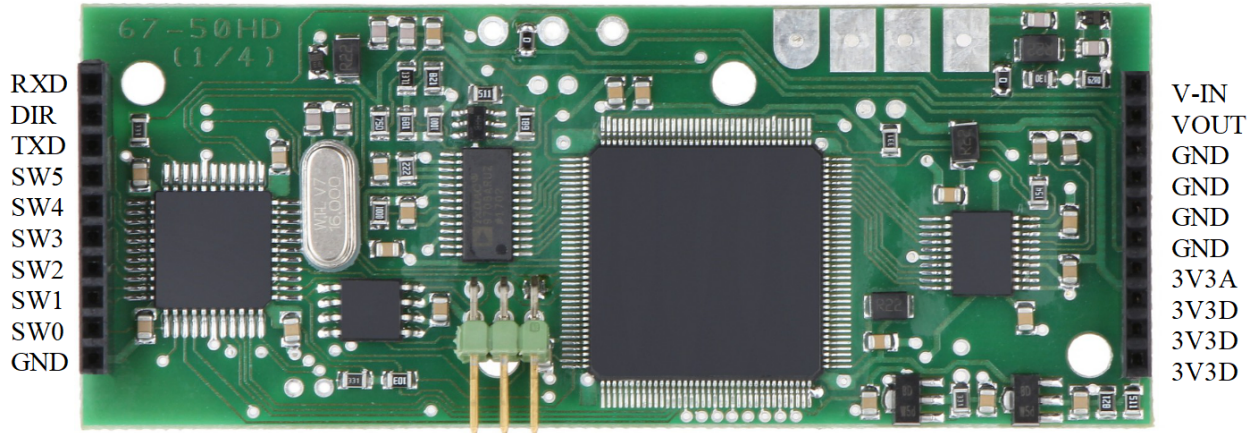
Obsługiwane standardy wideo:	AHD, HD-CVI, HD-TVI (720p / 1080p) CVBS (D1 / 960H)
System wideo	PAL / NTSC
Wzmocnienie sygnału wideo:	0 dB
Komunikacja / sterowanie:	UART (3.3V)
Prędkość transmisji [b/s]	2400,4800,9600,19200,28800,38400,57600,115200, 234000,468000, 936000, 1872000
Współpraca z innymi urządzeniami:	PORT-22
Zasilanie:	DC 3.3V / 0.8 A
Waga:	0.02 kg
Wymiary:	91 x 36 x 11 mm

Urządzenie służy do nakładania znaków alfanumerycznych, grafiki rastrowej oraz grafiki wektorowej na obraz z kamery analogowej. Do zasilania wymagane jest napięcie 3,3V. Sterowanie odbywa się poprzez port UART (3,3V).

Najważniejsze funkcje urządzenia:

- Obsługa grafiki wektorowej
- Obsługa grafiki rastrowej
- Nakładanie dowolnych tekstów
- Darmowa biblioteka do integracji z Arduino
- Kompatybilne z Arduino / Nucleo (wymagany moduł PORT-22)
- Wbudowana funkcja nakładania krzyża
- Współpraca z wagami i siłomierzami
- Intuicyjne menu urządzenia
- Możliwość zmiany czcionki oraz stworzenia własnej tablicy znaków graficznych
- Możliwość aktualizacji oprogramowania

2. Opis wyprowadzeń



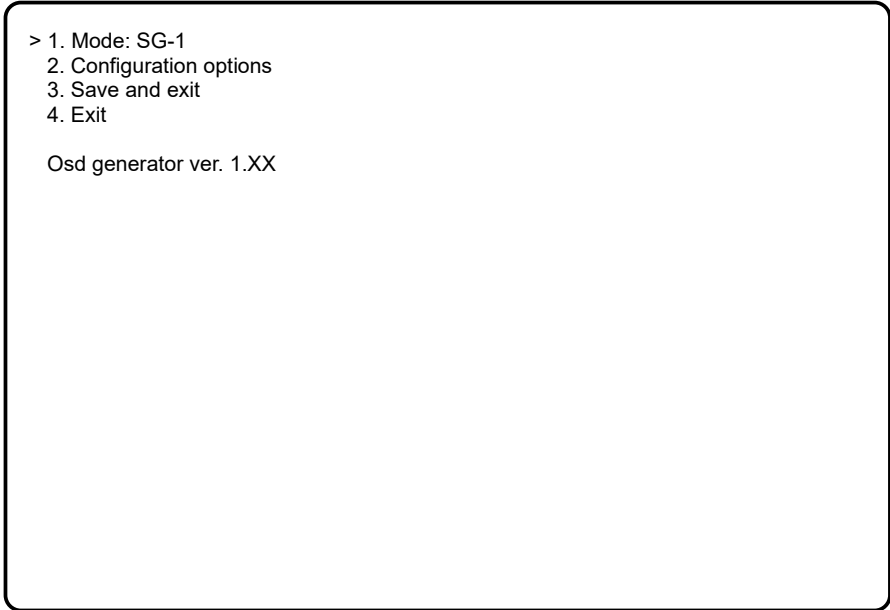
RXD	Linia RX UART (3,3V)
DIR	Kierunek transmisji (3,3V)
TXD	Linia TX UART (3,3V)
SW5	Przycisk (dół)
SW4	Przycisk (górze)
SW3	Przycisk (prawo)
SW2	Przycisk (lewo)
SW1	Przycisk (SYSTEM)
SW0	Przycisk (ENTER) / Tryb BOOTLOADER
Vin	Wejście wideo
Vout	Wyjście wideo
GND	Masa urządzenia
3V3A	Zasilanie części analogowej
3V3D	Zasilanie części cyfrowej

3. Konfiguracja standardu wideo

Domyślnie nowe urządzenie ustawione jest w trybie HD-CVI. Jeśli sygnał wejściowy jest w innym standardzie wideo to należy ustawić poprawny standard. Dostępne są następujące możliwości: CVBS, AHD720, AHD1080, HD-CVI720, HD-CVI1080, HD-TVI720, HD-TVI1080. Do zmiany standardu należy użyć przycisku „SYSTEM”. Wciśnięcie go lub zwarcie SW 1 do GND powoduje wybór kolejnego systemu. Należy go przyciskać (zwierać SW1 do GND) tyle razy aż na ekranie pojawią się stabilne napisy i będzie to ustawienie zgodne z systemem kamery jaka została podłączona. Naciśnięcie przycisku „ENTER” (zwarcie SW 0 do GND) otwiera menu ekranowe urządzenia. Przechodzenie pomiędzy kolejnymi pozycjami menu odbywa się za pomocą przycisków, zmiany wybranego parametru lub wybór opcji użytkownik dokonuje przyciskiem „ENTER”. Wyjście z menu ekranowego następuje po wybraniu pozycji EXIT.

3.1 Menu urządzenia

Aby wejść do menu użyj przycisku ENTER



```
> 1. Mode: SG-1
  2. Configuration options
  3. Save and exit
  4. Exit

Osd generator ver. 1.XX
```

W menu głównym urządzenia dostępne są następujące opcje:

1. Mode (tryb pracy):

TERMINAL – wyświetlanie ciągów znaków odbieranych przez urządzenie z uwzględnieniem znaku nowej linii (NL) i/lub (CR),

HEX – wyświetlanie każdego odebranego znaku w formie szesnastkowej,

SG-1 – wyświetlanie grafiki rastrowej, wektorowej oraz ciągów znaków w dowolnym miejscu ekranu,

CT5 – tryb obsługi termometru CT5,

RI-8300 (printer) – drukarka fiskalna Riotech (tryb grafiki tylko rastrowej, bez możliwości wyświetlania tekstu w trybie tekstowym),

8530 cougar – wagi obsługujące standard 8530 cougar np.: METTLER TOLEDO 8530,

AXIS – wagi i siłomierze Axis,

RADWAG – wagi Radwag,

RHEWA – wagi Rhewa

2. Configuration options: wejście do menu szczegółowych opcji konfiguracji, (szczegółowy opis – następny rozdział)

3. Save and exit – zapis konfiguracji i wyjście z menu,

4. Exit – wyjście z menu ekranowego.

3.2 Ustawienia urządzenia - Configuration options

> Baudrate:	9600 bits/s
Term line termination:	0x0A
Temperature read interval	255 s
Screen clear delay:	infinite
Screen change delay:	infinite
First display line:	line 1
Last display line:	line 40
Charset:	default
Font size:	normal
Position:	right
System:	HD-CVI 1080p
SG-1 ID:	1
Font Downloading	0%
Cross:	disable
Cross width:	1
Temp relay-1 HEAT	15 °C
Temp relay-2 COOL	50 °C
Temp hysteresis	0.5 °C
Temp	internal
Temp min	- 15 °C
Range	65 °C
Display configuration	Text & bitmap & chart
Relay configuration	external
Exit	

W menu Configuration options dostępne są następujące ustawienia:

1. **Baudrate** – prędkość transmisji RS - 485, dla większości urządzeń domyślną wartością jest 9600 b/s,
2. **Term line termination** – wybór znaku końca linii dla trybu TERMINAL, lub brak znaku końca linii,
3. **Temperature read interval** – odstęp czasu pomiędzy odczytami temperatury z modułu CT5 (1...255s),
4. **Screen clear delay** – czas, po którym nastąpi czyszczenie ekranu jeżeli nie pojawiają się żadne nowe dane (1 ... 250s lub nieskończoność)
5. **Screen change delay** –(nie dotyczy trybu SG-1) czas, po którym nastąpi zmiana ekranu (0.5s, 1s, 2s, 4s)
6. **First display line** – wybór pierwszej linii w której będzie się pojawiać tekst odbierany przez urządzenie, opcja pozwala na określenie fragmentu ekranu, w którym będą pojawiać się dane (1...40),
7. **Last display line** – wybór ostatniej linii, w której pojawi się tekst odbierany przez urządzenie (1...40),
8. **Charset** – wybór strony kodowej dla polskich liter, dla domyślnego ustawienia należy pozostawić default (tryb tekstowy).
9. **Font size** – wybór wielkości czcionki. Dla standardu CVBS do wyboru tylko „normal” i „big”. Dla systemów HD dodatkowo „small”,
10. **Position** – wybór położenia napisów (lewo/prawo). W trybie SG-1 przesuwają o stałą odległość nakładanie wszystkich obiektów w prawo,
11. **System** – wybór standardu wideo,
12. **SG-1 ID** – numer ID urządzenia, dla pracy wielu urządzeń,
13. **Font Downloading** – postęp wczytywania zestawu znaków użytkownika,
14. **Cross** – funkcja rysująca markery w postaci krzyża na ekranie,
15. **Cross width** – grubość linii markerów (1...255 pikseli),

16. Heating activation temp. – temperatura załączenia przekaźnika dla urządzeń grzejących. Poniżej tej temperatury zostaje załączone wyjście,

17. Cooling activation temp. – temperatura załączenia przekaźnika dla urządzeń schładzających. Powyżej zdefiniowanej temperatury zostaje załączone wyjście,

18. Temp hysteresis – histereza załączania przekaźników;

19. Temp – wybór konfiguracji dla czujnika zewnętrznego lub wewnętrznego (external lub internal);

20. Temp min – najniższa wartość prezentowana na wykresie i graficznym termometrze;

21. Range – zakres prezentowanych wartości temperatury

22. Display configuration – wybór prezentacji danych:

Text – wyświetlanie wartości temperatury w formie danych tekstowych

bitmap – prezentacja w formie analogowego termometru

chart – rysowanie wykresu

Do wyboru dostępne są 4 opcje: text / text & bitmap / text & bitmap & chart / text & chart

23. Relay configuration – wybór czujnika dla funkcji termostatu,

24. Exit – wyjście z menu

4. Modyfikacja fontów

OSD-50HD posiada możliwość zmiany fontów przez użytkownika poprzez stworzenie własnej tablicy znaków. Istnieją dwa zestawy fontów. Jeden używany w menu - tego zestawu fontów nie można zmienić. Drugi zestaw, automatycznie ładowany jest po wyjściu z menu głównego i może być zmodyfikowany przez użytkownika. Edycji liter można dokonać za pomocą darmowego programu „Font Editor” dostępnego na stronie <http://osd.systems> .

Można także bezpośrednio modyfikować bitmapę w dowolnym programie graficznym, np: Paint. Obraz musi mieć rozdzielczość 170x425 pikseli. Głębokość kolorów 8 bitów. Z naszej strony <http://osd.systems> można pobrać edytowalny plik z fontami systemowymi (FONTS_16x24_Basic.bmp). Następnie tak stworzony plik należy wgrać do urządzenia. Do wgrania pliku .bmp z własnym zestawem fontów służy program SG-55 Bootloader.

Wczytywanie pliku z fontami :

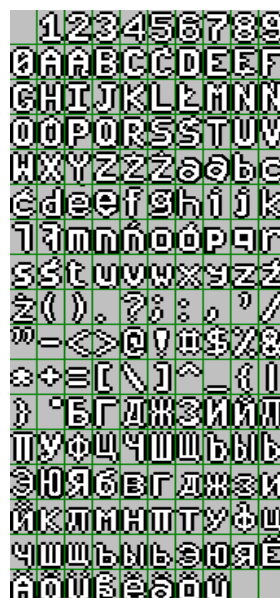
Aby wysłać plik, urządzenie musi być podłączone do komputera za pomocą adapterów PORT-22 + USB/RS485. Nie ma znaczenia jaki aktualnie jest wybrany standard, ani protokół. Żeby wczytać fonty trzeba jedynie wejść do menu konfiguracyjnego. W menu pod pozycją „Font Downloading” wyświetlany jest procentowy postęp ściąganych danych.

Obsługa programu „SG-55 Bootloader”:

- Ustawiamy szybkość transmisji na taką samą jak w OSD-50HD.
- Klikamy opcje „Send Fonts” i wybieramy plik z fontami.

Uwaga!

- Menu urządzenia wyświetlane jest przy użyciu oryginalnych fontów systemowych.
 - Wczytane fonty wyświetlane są dopiero po wyjściu z menu głównego.
 - Po aktualizacji oprogramowania należy ponownie wczytać własną tablicę fontów.
- Domyślnie dostępne fonty: (polskie litery, cyrylica oraz niemieckie znaki diakrytyczne):



5. Tryb CT5 – pomiar temperatury

W tym trybie wymagane jest użycie dedykowanego modułu termometru dwupunktowego CT5

Konfiguracja trybu CT5:

16. Heating activation temp. – temperatura załączenia przełącznika dla urządzeń grzejących. Poniżej tej temperatury zostaje załączone wyjście,

17. Cooling activation temp. – temperatura załączenia przełącznika dla urządzeń schładzających. Powyżej zdefiniowanej temperatury zostaje załączone wyjście,

18. Temp hysteresis – histereza załączania przełączników;

19. Temp – wybór konfiguracji dla czujnika zewnętrznego lub wewnętrznego (external lub internal);

20. Temp min – najniższa wartość prezentowana na wykresie i graficznym termometrze;

21. Range – zakres prezentowanych wartości temperatury

22. Display configuration – wybór prezentacji danych:

Text – wyświetlanie wartości temperatury w formie danych tekstowych

bitmap – prezentacja w formie analogowego termometru

chart – rysowanie wykresu

Do wyboru dostępne są 4 opcje: text / text & bitmap / text & bitmap & chart / text & chart

23. Relay configuration – wybór czujnika dla funkcji termostatu,

6. Tryb SG-1 (protokół SG-1)

W tym trybie można wyświetlać grafikę rastrową, wektorową oraz ciągi znaków tekstowych w dowolnym miejscu ekranu. Tryb umożliwia implementację własnych protokołów. Dostępne są następujące funkcje:

CLRSCR	0x03 – czyszczenie tekstu na ekranie,
TEXT	0x04 – wyświetlanie tekstu,
LINE	0x05 – rysowanie dowolnych linii (ukośne),
CIRCLE	0x06 – rysowanie okręgu,
LINEHV	0x07 – rysowanie linii poziomych i pionowych,
WYKRES	0x08 – rysowanie wykresu,
SETWYKRES	0x0A – definiowanie wykresu,
SETGRID	0x0B – rysowanie siatki,
UPGRADEBITMAP	0x0C – wyświetlanie bitmapy,
SETBITMAP	0x0D – ustawienia dla bitmapy,

Ilość wyświetlonych elementów jest ograniczona. Poniżej podano ile maksymalnie można wyświetlić danych elementów:

- tekst (pole tekstowe 49 x 40 znaków)
- linie ukośne (4)
- linie pionowe, poziome lub wypełnione prostokąty (12),
- koła lub okręgi (2),
- siatki (2),
- wykresy (2),
- bitmapy (2).

Każdy element jest odrębnym obiektem, który można dowolnie modyfikować nie wpływając na pozostałe. Siatka składa się z wielu linii poziomych oraz pionowych. Jednak w tym przypadku nie ma ograniczenia ilości linii. Może być tyle ile się zmieści na ekranie.

6.1 Czyszczenie tekstu na ekranie

Struktura ramki dla funkcji czyszczenia ekranu z tekstu wygląda następująco:

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

0xFF – nagłówek,

SG -1_ID – adres, na którym urządzeniu ma być wyczyszczony tekst,

0x03 – funkcja czyszczenia ekranu

6.2 Wyświetlanie tekstu

Struktura ramki dla funkcji wyświetlania tekstu wygląda następująco:

0xFF, SG-1_ID, 0x04, POS_X, POS_Y, ILOŚĆ, 0, 0, 0, 0, <tekst>, 0x03

0xFF – nagłówek,

SG -1_ID – adres, na którym urządzeniu ma pojawić się napis,

0x04 – funkcja wyświetlająca napis

POS_X – Położenie X na ekranie (liczona w znakach maks 49),

POS_Y – Położenie Y na ekranie (liczona w znakach maks 40 dla liter normalnych i małych i 22 dla liter dużych),

ILOŚĆ – ilość znaków tekstu (jednorazowo można wysłać maksymalnie 64 znaki),

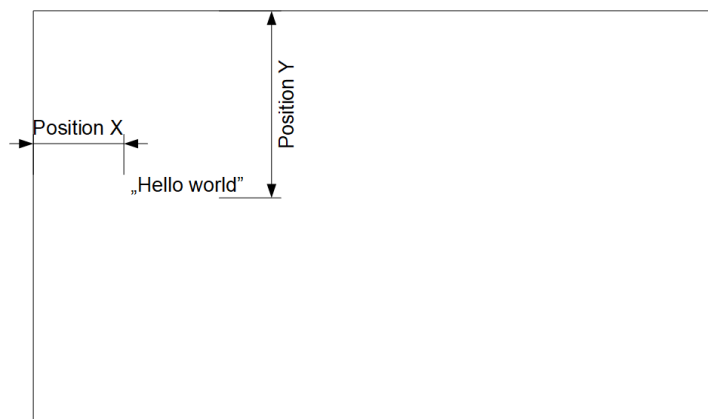
<tekst> – tekst w kodzie ASCII

0x03 – znak końca tekstu

Przerwa pomiędzy komendą a wysyłanym tekstem nie może być dłuższa niż 500ms (auto-timeout),

Przykład:

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 – nagłówek,

0x01 – adres urządzenia,

0x04 – funkcja wyświetlająca tekst,

0x05 – Położenie tekstu na osi X, przesunięcie o X= 5 znaków,

0x0B – Położenie tekstu na osi Y, przesunięcie o Y= 5 znaków,

0x0D – ilość znaków (maks. 64),

0x00,0x00,0x00,0x00 - zarezerwowane,

0x22,0x48,0x65,0x6C,0x6C,0x6F,0x20,0x77,0x6F,0x72,0x6C,0x64,0x22 – tekst „Hello world” w kodzie ASCII,

0x03 – znak końca tekstu.

Przykładowy plik text.hex dostępny jest do pobrania na stronie <http://osd.systems>.

6.3 Rysowanie linii

Za pomocą tej funkcji można narysować linie o zadanej grubości lub wypełniony prostokąt, pod dowolnym kątem. Struktura ramki dla funkcji wyświetlania tego obiektu wygląda następująco:

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

0xFF – nagłówek,

SG -1_ID – adres, na którym urządzeniu ma pojawić się linia,

0x05 – funkcja rysująca linie,

LineWidthH, LineWidthL – starszy i młodszy bajt szerokości linii,

Y1H, Y1L – starszy i młodszy bajt współrzędnych Y punktu początkowego linii,

X1H, X1L – starszy i młodszy bajt współrzędnych X punktu początkowego linii,

Y2H, Y2L – starszy i młodszy bajt współrzędnych Y punktu końcowego linii,

X2H, X2L – starszy i młodszy bajt współrzędnych X punktu końcowego linii,

object – numer obiektu (0 – 3) maksymalnie mogą być 4 obiekty tego typu na ekranie.

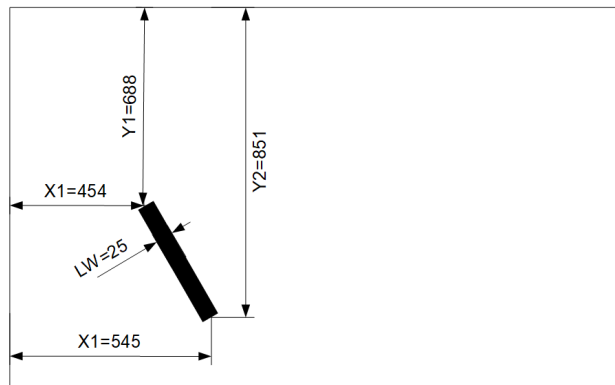
colour – kolor

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.

Vsync – przerysowanie linii tylko w czasie impulsu synchronizacji pionowej.

Przykład:

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



0xFF – nagłówek,

0x01 – adres urządzenia,

0x05 – funkcja rysująca linie,

0x00 – zarezerwowane,

0x00,0x19 – starszy i młodszy bajt szerokości linii = 25 pikseli,

0x02,0xB0 – starszy i młodszy współrzędnej Y1 = 0x02B0 = 688 pikseli,

0x01,0xC6 – starszy i młodszy współrzędnej X1 = 0x01C6 = 454 pikseli,

0x03,0x53 – starszy i młodszy współrzędnej Y2 = 0x0353 = 851 pikseli,

0x02,0x21 – starszy i młodszy współrzędnej X2 = 0x0221 = 545 pikseli,

0x00 – numer obiektu,

0x02 – kolor,

0x01 – rysowanie wraz impulsami synchronizacji,

0x00,0x00 – zarezerwowane.

Przykładowy plik line.hex dostępny jest do pobrania na stronie <http://osd.systems>

6.4 Rysowanie okręgu

Za pomocą tej funkcji można rysować okręgi i koła.

Struktura ramki dla funkcji wyświetlania tego obiektu wygląda następująco:

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

0xFF – nagłówek,

SG -1_ID – adres, na którym urządzeniu ma pojawić się napis (ustawiany w urządzeniu),

0x06 – funkcja rysująca okrąg

WidthH, WidthL – Starszy i młodszy bajt grubości linii tworzącej okrąg.

YH, YL – Współrzędne Y środka okręgu,

XH, XL – Współrzędne X środka okręgu,

radixH, radixL – Starszy i młodszy bajt promienia okręgu,

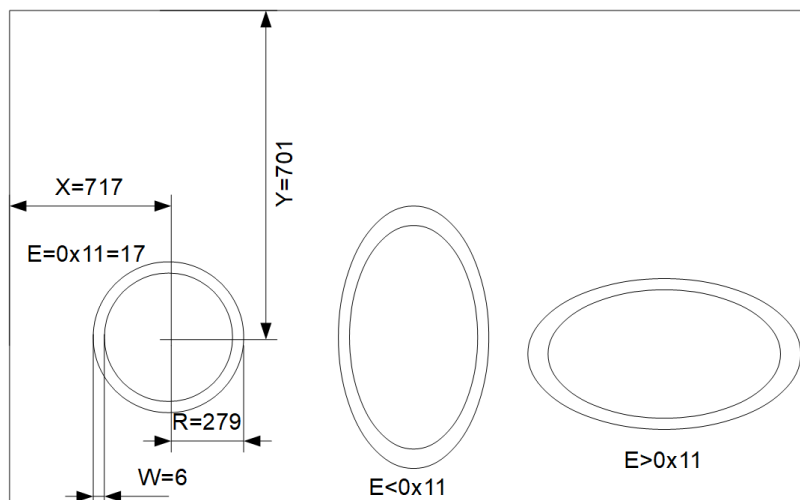
colour – kolor okręgu,

object – numer obiektu (8 – 9), maksymalnie mogą być 2 obiekty tego typu na ekranie.

Vsync – przerysowanie okręgu tylko w czasie impulsu synchronizacji pionowej.

Przykład:

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

Przykładowy plik circle.hex dostępny jest do pobrania na stronie <http://osd.systems>

6.5 Rysowanie linii HV

Za pomocą tej funkcji można narysować linie o zadanej grubości lub wypełniony prostokąt ale tylko pionowe lub poziome.

Struktura ramki dla funkcji wyświetlania tego obiektu wygląda następująco:

0xFF, SG-1_ID, 0x07, 0, 0, colour, LYH, LYL, LXH, LXL, YH, YL, XH, XL, object, Vsync, 0, 0, 0

0xFF – nagłówek,

SG -1_ID – adres, na którym urządzeniu ma pojawić się napis (ustawiany w urządzeniu,

0x07 – funkcja rysująca linie HV.

colour – kolor linii,

LYH, LYL – Starszy i młodszy bajt wysokości linii (wypełnionego prostokąta),

LXH, LXL – Starszy i młodszy bajt wysokości linii (wypełnionego prostokąta)

YH, YL – Starszy i młodszy bajt współrzędnych Y początku linii,

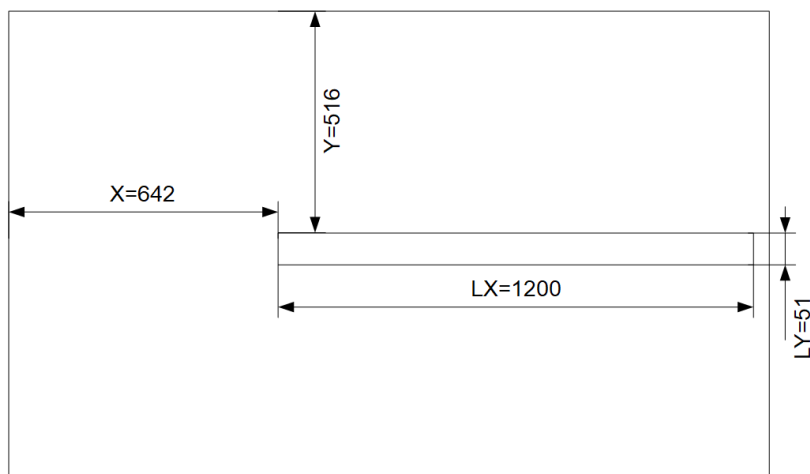
XH, XL – Starszy i młodszy bajt współrzędnych X początku linii,

object – numer obiektu (12-19), maksymalnie może być 8 obiektów tego typu na ekranie.

Vsync – przerysowanie linii tylko w czasie impulsu synchronizacji pionowej.

Przykład:

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.

Przykładowy plik LineHV.hex dostępny jest do pobrania na stronie <http://osd.systems>

6.6 Rysowanie siatki

Za pomocą tej funkcji można narysować siatkę złożoną z linii poziomych i pionowych o regulowanym odstępie, grubości linii i obszarze, który mają zajmować na ekranie.

Struktura ramki dla funkcji wyświetlania siatki wygląda następująco:

0xFF, SG-1_ID, 0x0B, 0 , colour, LineWidth, DYH, DYL, DXH, DXL, Y1H, Y1L, X1H, X1L, Y0H, Y0L, X0H, X0L, object.

0xFF – nagłówek,

SG -1_ID – adres, na którym urządzeniu ma pojawić się napis (ustawiany w urządzeniu,

0x0B – funkcja rysująca siatkę

colour – kolor linii,

LineWidth – grubość linii siatki

DYH, DYL – Starszy i młodszy bajt odległości między liniami w osi Y,

DXH, DXL – Starszy i młodszy bajt odległości między liniami w osi X,

Y1H, Y1L – Starszy i młodszy bajt współrzędnej Y prawego dolnego rogu siatki,

X1H, X1L – Starszy i młodszy bajt współrzędnej X prawego dolnego rogu siatki.

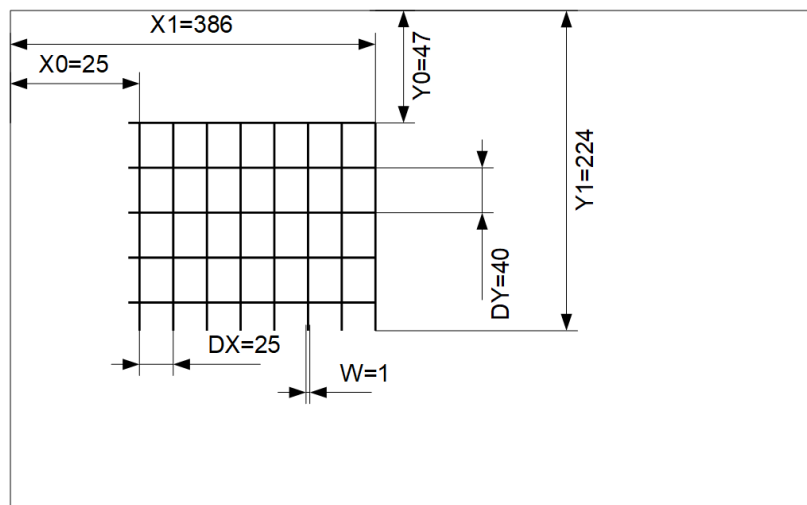
Y0H, Y0L – Starszy i młodszy bajt współrzędnej Y lewego górnego rogu siatki,

X0H, X0L – Starszy i młodszy bajt współrzędnej X lewego górnego rogu siatki,

object – numer obiektu (30-31), maksymalnie mogą być 2 obiekty tego typu na ekranie.

Przykład:

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.

Przykładowy plik SetGrid.hex dostępny jest do pobrania na stronie <http://osd.systems>.

6.7 Ustawienia wykresu

Za pomocą tej funkcji definiujemy obszar, na którym będziemy rysować wykres.
Struktura ramki dla funkcji ustawień wykresu wygląda następująco:

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

0xFF – nagłówek,

SG -1_ID – adres, na którym urządzeniu ma pojawić się wykres (ustawiany w urządzeniu),

0x0A – funkcja ustawiająca parametry wykresu,

colour – kolor wykresu,

LY – skala na osi Y, wzrost parametru powoduje wzrost wykresu. Wartość maksymalna 255.

LX – Długość wykresu wyrażony w wielokrotnościach 16. Np: 10 oznacza 160 pikseli.

YH, YL – Starszy i młodszy bajt współrzędnej górnej krawędzi wykresu.

XXH, XXL – Starszy i młodszy bajt parametru określającego gdzie na osi X jest koniec wykresu,

$XX = X + 16 * LX$,

X, Y – współrzędne lewego górnego rogu wykresu,

object – numer obiektu (28-29), maksymalnie mogą być 2 obiekty tego typu na ekranie.

Przykład:

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,

Przykładowy plik SetPlot.hex dostępny jest do pobrania na stronie <http://osd.systems>

6.8 Rysowanie wykresu

Za pomocą tej funkcji wpisujemy do pamięci urządzenia dane do narysowania wykresu. Struktura ramki dla funkcji rysowania wykresu wygląda następująco:

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

0xFF – nagłówek,

SG -1_ID – adres, na którym urządzeniu ma pojawić się wykres (ustawiany w urządzeniu),

0x08 – funkcja rysująca wykres,

plot – numer wykresu (0 – 1),

valueH, valueL – starszy i młodszy bajt zapisywanej wartości. Wartość jest 8-cio bitowa (0..255),

Przykład:

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

0xFF – Header,

0x01 – SG_ID,

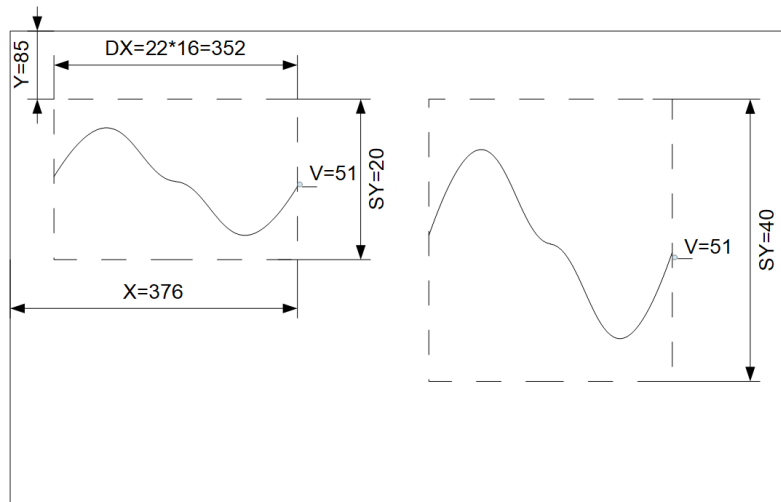
0x08 – UpdatePlot,

0x00 – reserved,

0x00 – number of plot,

0x00 – reserved,

0x33, V=0x33=51.



Przykładowy plik Update.hex (wyświetlenie jednej wartości : 51) dostępny jest do pobrania na stronie <http://osd.systems>.

6.9 Wyświetlanie bitmapy

6.9.1 Sposób przygotowania bitmapy do wyświetlenia

Obraz bitmapy można utworzyć w dowolnym programie graficznym lub w dołączonym do produktu programie „OSD Bitmap Editor”, który jest dostępny w zakładce programu „Font Editor”.

Obraz musi mieć ściśle określoną rozdzielczość i ilość kolorów. Szerokość musi być wielokrotności 32. Bitmapa nie może być większa niż 40960 pikseli. Dostępne są następujące możliwości rozdzielczości:

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

Program Osd Bitmap Editor dostępny jest w zakładce programu Font Editor (Tools > Osd Bitmap editor).

Program jest do pobrania ze strony <http://osd.systems>.

Program umożliwia wykonanie podstawowych funkcji potrzebnych do przygotowania obrazu bitmapy dla urządzenia OSD-50HD. Dostępne funkcje:

Choose bmp – otwieranie pliku (wybór bitmapy),

Center DIM – centrowanie kadru bitmapy,

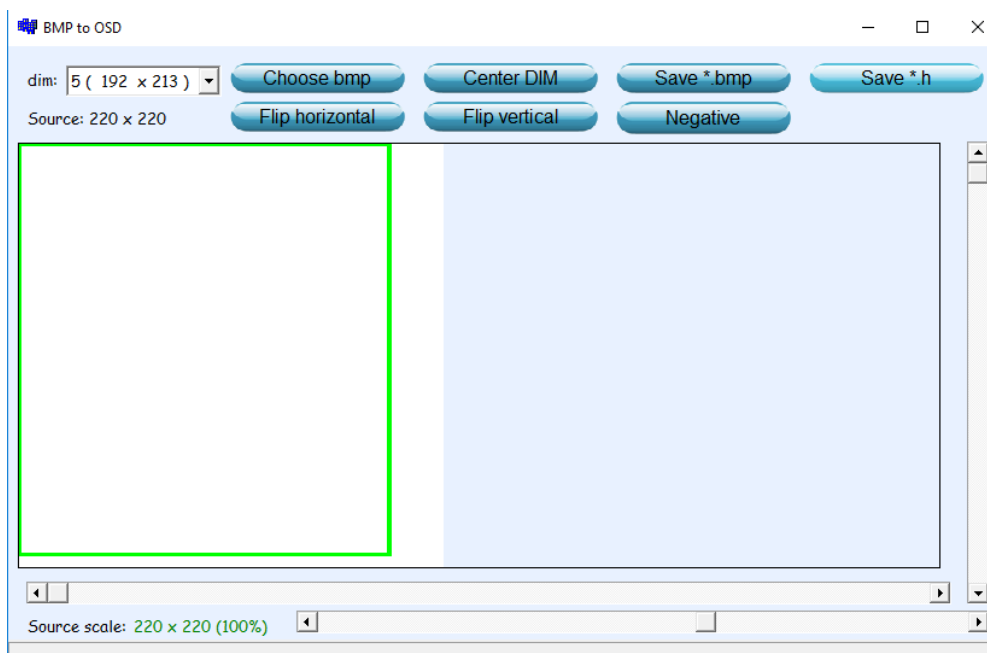
Save *.bmp – zapis bitmapy w formacie monochromatycznym,

Save *.h – zapis do pliku nagłówkowego zawartości pamięci EEPROM dla procesorów Atmega328,

Flip horizontal – odwrócenie obrazu w poziomie,

Flip vertical – odwrócenie obrazu w pionie,

Negative – bitowe odwrócenie koloru.



6.9.3 Płynność rysowania poruszającej się grafiki

Do rysowania poruszających się linii i okręgów można włączyć dodatkową opcję zwiększającą płynność wyświetlania. Funkcja ta nazywa się Vsync i polega na rysowaniu obiektów w takt impulsów synchronizacji pionowej sygnału wideo. Po wysłaniu komendy rysowania z włączoną opcją Vsync należy odczekać czas 40ms przed wysłaniem kolejnej komendy.

6.9.4 Ustawienia bitmapy 0x0D

Urządzenie umożliwia wyświetlenie jednokolorowej bitmapy. Aby to było możliwe muszą być spełnione odpowiednie warunki. Przed wysłaniem bitmapy do urządzenia należy najpierw zdefiniować urządzeniu parametry tego obrazu. Dane jakie należy wysłać: położenie obrazu na ekranie, parametr „dim” opisujący proporcję bitmapy, skala X i Y, kolor. Struktura ramki dla funkcji wyświetlania tekstu wygląda następująco:

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

0xFF – nagłówek,

SG -1_ID – adres, na którym urządzeniu ma pojawić się bitmapa (ustawiany w urządzeniu),

0x0D – funkcja przygotowująca bitmapę do wyświetlenia.

dim_colour = dim + colour*64 ,

gdzie: dim – wartość z tabeli, colour – kolor,

YH, YL – starszy i młodszy bajt współrzędnych Y lewego górnego rogu bitmapy,

XH, XL – starszy i młodszy bajt współrzędnych X lewego górnego rogu bitmapy,

LY – skala w osi Y,

LX – skala w osi X,

Wartość 255 – obiekt nieprzeskalowany. Zmniejszanie wartości LX, LY powoduje rozciągnięcie bitmapy w odpowiedniej osi.

object – numer obiektu. Wartości (38 - 39), maksymalnie mogą być 2 obiekty tego typu na ekranie.

Przykład:

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 – obiekt,

0x00,0x00,0x00,0x00 – reserved,

Przykładowy plik UpdateBitmap.hex dostępny jest do pobrania na stronie <http://osd.systems>.

6.9.5 Wyświetlenie bitmapy 0x0C

Za pomocą tej funkcji wyświetlamy bitmapę

Struktura ramki dla funkcji wyświetlania bitmapy wygląda następująco:

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 – Starszy i młodszy bajt adresu od którego chcemy rysować bitmapę. Jeśli adres=0 to rysujemy całą bitmapę,

LenghtL, LenghtH – młodszy i starszy bajt (zwróć uwagę na kolejność) ilości danych do przesłania od wskazanego adresu,

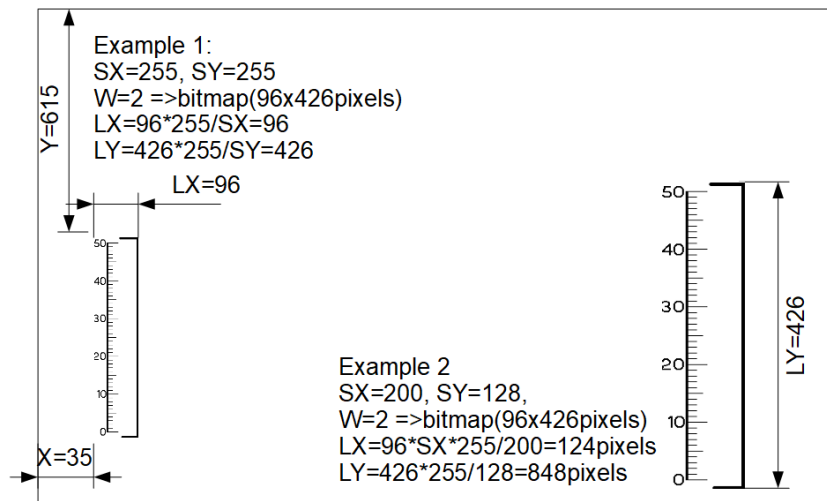
nrBmp – numer bitmapy, którą aktualizujemy (0, 1)

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.

Przykład:

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 = $0x13F8 = 5112 \text{ bajts} = 5112 * 8 = 40896 \text{ pixels}$

0x00 – number of bitmap,

0xFF,0xFF,0xFF,0xFF,0xFF,0xFF,0xFF,0xFF ... 0xFF,0xFF,0x07,0x00,0x00,0x00,0x00 – data.

7. Integracja z Arduino

Urządzenie OSD-50HD dzięki swojej funkcjonalności może być w prosty sposób zintegrowane z innymi systemami. Najprostszym sposobem na implementację własnych rozwiązań jest użycie modułu PORT-22. Moduł PORT-22 kompatybilny jest z Arduino Nano oraz ST/Nucleo.

Dostępna jest również darmowa biblioteka dla Arduino obsługująca wszystkie funkcje OSD-50HD. Bibliotekę można pobrać ze strony producenta: www.osd.systems.

Dostępne są następujące funkcje:

SendText(); - Wysłanie tekstu na ekran .

ClearText(); - Czyszczenie całego tekstu,

SendLineHV(); - rysowanie linii pionowej, poziomej, lub ukośnej o zadanej grubości, położeniu, synchronizacji pionowej i kolorze.

UpdateBitmap(); - zmiana zawartości bitmapy.

SetBitmap(); - Ustawienie parametrów bitmapy: położenie, parametr „dim” opisujący proporcje bitmapy, skala X i Y, kolor .

SendGrid(); - narysowanie siatki o zadanym obszarze rysowania, odstępnie między liniami X i Y, grubości linii, .

SetPlot(); - ustawienie parametrów wykresu: położenie, rozmiar kolor.

UpdatePlot(); - aktualizacja wykresu.

SendCircle(); - rysowanie koła o zadanym położeniu, promieniu i synchronizacji pionowej.

SendLine(); - rysowanie linii pionowej, poziomej lub wypełnionego prostokąta o zadanym położeniu, rozmiarach i synchronizacji pionowej.

7.1 Wyświetlanie tekstu

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

SendText(ID, X, Y, text)

ID – adres urządzenia,

X, Y – współrzędne początku tekstu,

text - tekst w kodzie ASCII.

przykład:

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

7.2 Rysowanie linii H V

Funkcja rysuje poziome lub pionowe linie o zadanej grubości. Za pomocą tej funkcji można także rysować wypełnione prostokąty. Maksymalnie można wyświetlić 12 tego typu obiektów (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 – adres urządzenia,

X,Y – położenie lewego górnego punktu linii,

lenght_X, lenght_Y – długość oraz szerokość linii,

colour – kolor: 0 - brak, 1 - biały, 2 – czarny,

Vsync – jeśli ustawione 1 to rysowanie rozpoczyna się po wykryciu impulsu synchronizacji pionowej,

Object – numer obiektu (12 ... 23),

Przykład:

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

7.3 Rysowanie siatki

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 – adres urządzenia,

leftup_X, leftup_Y – współrzędne lewego górnego punktu siatki,

rightdown_X, rightdown_Y – współrzędne prawego dolnego punktu siatki,

line_width – szerokość linii,

distance_X – odległość pomiędzy liniami poziomymi,

distance_Y – odległość pomiędzy liniami pionowymi,

colour – kolor: 0 - brak, 1 - biały, 2 – czarny,

object – numer siatki

Przykład:

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

7.4 Definiowanie bitmapy

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 – adres urządzenia,

X, Y – współrzędne lewego górnego punktu bitmapy,

scale_X, scale_Y – rozciąganie obrazu (wartość 255 – obraz rzeczywisty, 1 – maksymalne rozciągnięcie),

dim – wartość z tabeli dim,

style – wartość musi być 0,

colour – 0 - przezroczysty, 1 - biały, 2 – czarny,

object – numer wyświetlanej bitmapy

7.5 Wyświetlanie bitmapy

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

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

ID – adres urządzenia,

object – numer wyświetlanej bitmapy,

address – adres początkowy odczytu pamięci bitmapy,

count – ilość wyświetlanych danych,

data – wskaźnik do bitmapy,

7.6 Ustawienia wykresu

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

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

ID – adres urządzenia,

X, Y – współrzędne położenia lewego górnego rogu obszaru wykresu,

LX – długość wykresu. Długość musi być wielokrotnością 16. $LX = x * 16$ pixeli,

LY – wysokość wykresu,

style – opcja nieaktywna,

colour – 0 - przezroczysty, 1 - biały, 2 – czarny,

object – numer wykresu(0 lub 1),

7.7 Rysowanie wykresu

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

UpdatePlot(ID, object, value)

ID – adres urządzenia,

object – numer obiektu(0 lub 1),

value – wstawienie pojedynczej wartości.

Po każdym wprowadzeniu wartości następuje przesunięcie wykresu o jeden pomiar w lewo.

7.8 Rysowanie okręgu

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 – adres urządzenia,

X, Y – współrzędne środka okręgu,

radius – długość promienia,

width – grubość linii,

colour – kolor: 0 - przezroczysty, 1 - biały, 2 – czarny,

Vsync - 1 – Włączenie synchronizacji rysowania wraz z impulsami synchronizacji pionowej w sygnale wideo,

object – numer obiektu (8 ... 9).

Przykład:

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

7.9 Rysowanie linii

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 – adres urządzenia,

X1, Y1 – współrzędne początku linii,

X2, Y2 – współrzędne końca linii,

width – grubość linii,

colour – kolor: 0 - przezroczysty, 1 - biały, 2 – czarny,

Vsync - 1 – Włączenie synchronizacji rysowania wraz z impulsami synchronizacji pionowej w sygnale wideo,

object – numer obiektu (0 ... 3).

8. Aktualizacja oprogramowania

Urządzenie posiada możliwość aktualizacji oprogramowania. Wejście w tryb aktualizacji odbywa się poprzez przytrzymanie przycisku ENTER podczas podłączania zasilania urządzenia. Aktualna wersja oprogramowania i opis procedury aktualizacji dostępne są na stronie <http://osd.systems>