

**Dokumentacja biblioteki**  
**Wyświetlacz 128x64**  
**i2c SSD1306**

Michał Czosnyka  
Przemysław Domagała

December 17, 2024

# Zawartość

<b>1</b>	<b>Wstęp</b>	<b>1</b>
<b>2</b>	<b>Opis funkcji</b>	<b>1</b>
2.1	i2c_init . . . . .	1
2.2	ssd1306_init . . . . .	2
2.3	ssd1306_clear_screen . . . . .	2
2.4	ssd1306_display_text . . . . .	2
2.5	ssd1306_handle_display_text . . . . .	3
2.6	ssd1306_invert . . . . .	3
2.7	ssd1306_contrast . . . . .	4
2.8	ssd1306_yoffset . . . . .	4
<b>3</b>	<b>Opis szczegółowy wybranych elementów</b>	<b>4</b>
3.1	Struktura urządzenia SSD1306_t . . . . .	4
3.2	fonts . . . . .	5

## 1 Wstęp

Plik zawiera dokumentację biblioteki obsługującej użycie wyświetlacza z sterownikiem ssd1306 o rozmiarach 128x64 z komunikacją (i jej inicjalizacją) przez magistralę i2c.

## 2 Opis funkcji

Opis wszystkich funkcji biblioteki, ich działania, parametrów i wartości zwracanych.

### 2.1 i2c\_init

```
void i2c_init(SSD1306_t * dev, int16_t sda, int16_t scl);
```

Funkcja inicjalizująca komunikację przez magistralę i2c i komunikację konkretnie z wyświetlaczem.

- SSD1306\_t \* dev [in] - Struktura przechowująca niezmiennie informacje o wyświetlaczu<sup>3.1</sup>.

- `int16_t sda [in]` - Serial Data Line — linia danych, która służy do przesyłania informacji esp32 i wyświetlaczem.
- `int16_t scl [in]` - Serial Clock Line — linia zegarowa, która synchronizuje przesył danych.

## 2.2 `ssd1306_init`

`void ssd1306_init(SSD1306_t * dev);`

Funkcja inicjalizująca domyślne parametry wyświetlacza. Funkcja kolejno ustawia niezmiennicze pola w strukturze, wyłącza wyświetlacz, ustawia parametry z wartościami domyślnymi i wysyła konfigurację przez magistralę. Parametry są ustawiane jako komendy, których definicje są w pliku z [dokumentacją SSD1306](#).

- `SSD1306_t * dev [in]` - Struktura przechowująca niezmiennicze informacje o wyświetlaczu.

## 2.3 `ssd1306_clear_screen`

`void ssd1306_clear_screen(SSD1306_t * dev, bool invert);`

Funkcja czyści zawartość tekstową ekranu.

- `SSD1306_t * dev [in]` - Struktura przechowująca niezmiennicze informacje o wyświetlaczu.
- `bool invert [in]` - inwersja bitowa, można wyczyścić ekran zachowując inwersję kolorów.

## 2.4 `ssd1306_display_text`

`void ssd1306_display_text(SSD1306_t * dev, int page, char * text, int text_len, bool invert, int seg);`

Funkcja wypisuje tekst na ekranie.

- `SSD1306_t * dev [in]` - Struktura przechowująca niezmiennicze informacje o wyświetlaczu.
- `int page [in]` - numer linii na której powinien być wypisany kod (na ekranie wyznaczone jest 8 linii, od 0 do 7, w przypadku podania wartości z poza przedziału przyjętą wartością jest kraniec przedziału)

- `char * text [in]` - tablica znaków z tekstem do wypisania na ekranie.
- `int text_len [in]` - długość tekstu w tablicy znaków
- `bool invert [in]` - inwersja bitowa, można wyczyścić ekran zachowując inwersję kolorów.
- `int seg [in]` - segment, przesunięcie tekstów o liczbę pikseli `seg` w poziomie.

## 2.5 `ssd1306_handle_display_text`

`void ssd1306_handle_display_text(SSD1306_t * dev, int page, int seg, uint8_t * images, int width);`

Funkcja pomocnicza używana w funkcji `ssd1306_display_text` - nie jest przeznaczona do użytku oddzielnego.

- `SSD1306_t * dev [in]` - Struktura przechowująca niezmiennie informacje o wyświetlaczu.
- `int page [in]` - numer lini na której powinien być wypisany kod (na ekranie wyznaczone jest 8 lini, od 0 do 7, w przypadku podania wartości z poza przedziału przyjętą wartością jest kraniec przedziału)
- `int seg [in]` - segment, przesunięcie tekstów o liczbę pikseli `seg` w poziomie.
- `uint8_t * images [in]` - tablica reprezentująca 8 pikseli tekstu do wypisania - argument przekazany z głównej funkcji wypisywania na ekranie.
- `int width [in]` - szerokość znaku - z funkcji przekazywane jest 8 pikseli.

## 2.6 `ssd1306_invert`

`void ssd1306_invert(uint8_t *buf, size_t blen);`

Funkcja odwracająca kolory na ekranie.

- `uint8_t *buf [in]` - dane na których zostanie wykonana operacja inwersji (przekazując `invert` do `ssd1306_display_text` inwersja dokonuje się na całym wyświetlaczu i jest to zalecany sposób użycia)
- `size_t blen [in]` - długość bufora, na którym zostanie wykonana operacja inwersji bitowej.

## 2.7 **ssd1306\_contrast**

`ssd1306_contrast(SSD1306_t * dev, int contrast);`

Funkcja ustawiająca kontrast wyświetlacza.

- `SSD1306_t * dev [in]` - Struktura przechowująca niezmiennie informacje o wyświetlaczu.
- `int contrast [in]` - ustawienie kontrastu (0-255).

## 2.8 **ssd1306\_yoffset**

`void ssd1306_yoffset(SSD1306_t * dev, int xoffset);`

Funkcja ustawiająca offset w pionie (liczba o ile pikseli w dół od góry wyświetlacza będzie wyświetlany tekst).

- `SSD1306_t * dev [in]` - Struktura przechowująca niezmiennie informacje o wyświetlaczu.
- `int yoffset [in]` - ustawienie offsetu w pikselach (0-63).

# 3 **Opis szczegółowy wybranych elementów**

## 3.1 **Struktura urządzenia SSD1306\_t**

Struktura zawierająca niezmiennie poprzez konfigurację informacje o urządzeniu. Pola w strukturze:

- `int _address` - adres wyświetlacza w i2c (`I2C_ADDRESS`);
- `int _width` - szerokość wyświetlacza w pikselach (64);
- `int _height` - wysokość wyświetlacza w pikselach (128);
- `int _pages` - liczba linii wyznaczonych na wyświetlaczu (8);
- `PAGE_t _page[8]` - tablica lini;
- `i2c_port_t _i2c_num` (`I2C_NUM`);
- `i2c_master_bus_handle_t _i2c_bus_handle` - handle magistrali i2c;
- `i2c_master_dev_handle_t _i2c_dev_handle` - handle urządzenia na magistrali i2c;

## **3.2 fonts**

Czcionka literek wyświetlanych na wyświetlaczu - zawartość w pliku font8x8\_basic.h.  
Jest to tablica: static uint8\_t font8x8\_basic\_tr.