

# Wyświetlacz GM12864-59N VER:2.0 ST7567S (Arduino)

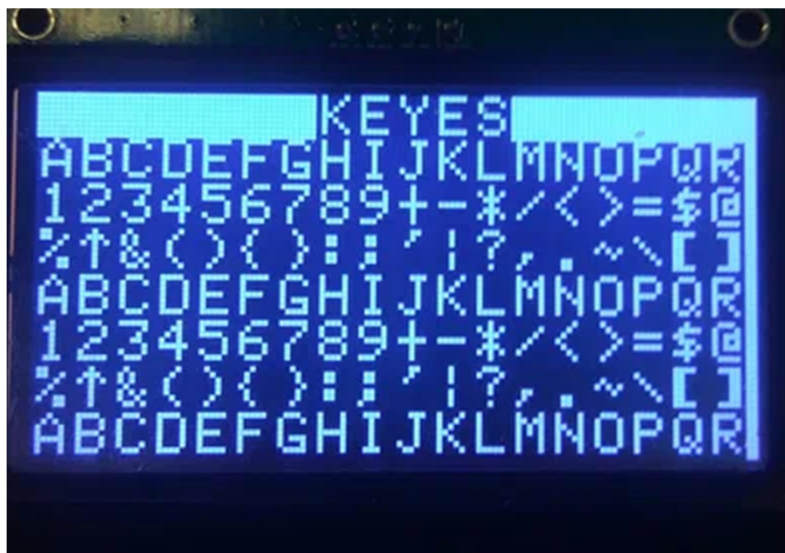
## Pomoc dotycząca Arduino

02.10.2022

Autor: [liman28](#)

Wyświetlacz GM12864-59N VER:2.0 na kontrolerze ST7567S ma połączenie I2C. Ten wyświetlacz ma rozmiar 128x64 pikseli i jest w stanie wyświetlać obrazy graficzne i symbole na ekranie.

Aby wyświetlacz działał poprawnie w środowisku Arduino IDE należy dodać bibliotekę [ST7567S\\_128X64\\_I2C](#).



Biblioteka ma jeden przykład, który pokazuje, jak działa wyświetlacz:

```
/*
*****
*oryginalny kod z
* URL: https://github.com/mworkfun/ST7567A_128X32DOT_LCD
* Zmodyfikuj dla LCD ST7567S 128x64
* URL: https://github.com/luetee/ST7567S_128X64_I2C.git
*****/
#include <Wire.h>
#include<lcd_st7567s.h> // https://github.com/luetee/ST7567S_128X64_I2C

lcd_st7567s LCD;

unieważnienie konfiguracji ( ) {
  Przewód. rozpocząć ( ) ;
  LCD. init ( ) ;
}

pusta pętla ( ) {
  Lcd. testPixel ( 2 ) ; //użyj do testowania ekranu LCD. Parametrem jest prędkość.

  for ( int a= 0 ; a < 64 ; a++ ) {
    Lcd. DisplayPixel ( a, a ) ; //wyświetl jeden piksel. X=0-31, Y=0-127
    opóźnienie ( 5 ) ;
  }
}
```

```

    for ( int a= 0 ; a < 64 ; a++ ) {
    Lcd. ClearPixel ( a,a ) ; //Nie wyświetla piksela. X=0-31, Y=0-127
    opóźnienie ( 5 ) ;
    }

    LCD. jasne ( prawda ) ;
    for ( int a= 0 ; a < 64 ; a++ ) {
    Lcd. ClearPixel ( a,a ) ; //Nie wyświetla piksela. X=0-31, Y=0-127
    opóźnienie ( 5 ) ;
    }
    opóźnienie ( 1000 ) ;

    LCD. jasne ( fałsz ) ;
    for ( int a= 0 ; a < 128 ; a++ ) {
    Lcd. DisplayPixel ( a, 0 ) ; //wyświetl jeden piksel. X=0-31, Y=0-127
    Wyświetlacz. DisplayPixel ( a, 63 ) ; //wyświetl jeden piksel. X=0-31, Y=0-127
    }
    for ( int a= 0 ; a < 64 ; a++ ) {
    Lcd. DisplayPixel ( 0, za ) ; //wyświetl jeden piksel. X=0-31, Y=0-127
    Wyświetlacz. DisplayPixel ( 127 , a ) ; //wyświetl jeden piksel. X=0-31, Y=0-127
    Wyświetlacz. DisplayPixel ( 4 , a ) ;
    }
    Wyświetlacz DrawLine ( 5 , 5 , 125 , 60 , false ) ;
    LCD. DrawLine ( 125 , 5 , 5 , 60 , fałsz ) ;
    LCD. DrawLine ( 53 , 31 , 73 , 31 , false ) ;
    LCD. DrawLine ( 63 , 20 , 63 , 40 , false ) ;

    LCD. draw_circle ( 20 , 30 , 10 , DRAW_ALL, false , false ) ;
    LCD. draw_circle ( 20 , 30 , 15 , DRAW_ALL, false , false ) ;

    LCD. draw_circle ( 105 , 30 , 15 , DRAW_ALL, false , true ) ;

    opóźnienie ( 1000 ) ;

    LCD. jasne ( fałsz ) ;
    LCD. DrawLine ( 53 , 31 , 73 , 31 , false ) ;
    LCD. DrawLine ( 63 , 20 , 63 , 40 , false ) ;
    LCD. draw_circle ( 90 , 30 , 10 , DRAW_ALL, false , false ) ;
    LCD. draw_circle ( 30 , 30 , 10 , DRAW_ALL, false , true ) ;
    opóźnienie ( 1000 ) ;

    LCD. jasne ( prawda ) ;
    LCD. DrawLine ( 53 , 31 , 73 , 31 , prawda ) ;
    LCD. DrawLine ( 63 , 20 , 63 , 40 , true ) ;
    LCD. draw_circle ( 90 , 30 , 10 , DRAW_ALL, true , false ) ;
    LCD. draw_circle ( 30 , 30 , 10 , DRAW_ALL, prawda , prawda ) ;
    opóźnienie ( 1000 ) ;

    /**/
    Wyświetlacz Kursor ( 7 , 0 ) ; //Pozycja wyświetlania znaków. y=0-3, x=0-17
    Wyświetlacz. Wyświetlacz ( "KLUCZE" ) ;
    LCD. Kursor ( 0 , 1 ) ;
    LCD. Wyświetl ( "ABCDEFGHJKLMNOPQR" ) ;
    LCD. Kursor ( 0 , 2 ) ;
    LCD. Wyświetl ( "123456789+*/<>=$@" ) ;
    LCD. Kursor ( 0 , 3 ) ;
    LCD. Wyświetl ( "%^&(){}:; '|? , . ~ \ \ [ ]" ) ;
    LCD. Kursor ( 0 , 4 ) ;
    LCD. Wyświetl ( "ABCDEFGHJKLMNOPQR" ) ;
    LCD. Kursor ( 0 , 5 ) ;
    LCD. Wyświetl ( "123456789+*/<>=$@" ) ;
    LCD. Kursor ( 0 , 6 ) ;
    LCD. Wyświetl ( "%^&(){}:; '|? , . ~ \ \ [ ]" ) ;
    LCD. Kursor ( 0 , 7 ) ;
    LCD. Wyświetl ( "ABCDEFGHJKLMNOPQR" ) ;
    opóźnienie ( 1000 ) ;
    LCD. jasne ( fałsz ) ;
    LCD. Kursor ( 0 , 0 ) ;
    for ( int i= 0 ; i < 100 ; i++ ) {
    opóźnienie ( 100 ) ;
    Ciąg znaków = Ciąg znaków ( i ) ;
    znak cstr [ 16 ] ;
    ul. toCharArray ( cstr, 16 ) ;
    LCD. Wyświetl ( cstr ) ;
    }

    LCD. jasne ( fałsz ) ; //Wszystkie piksele wyłączają się.

    LCD. Wyświetl obraz ( ) ; //Wyświetla dane obrazu dla
    opóźnienia pliku picture.c ( 1000 ) ;

```

