



# TinySine OLED Breakout

## User Manual



## INTRODUCTION

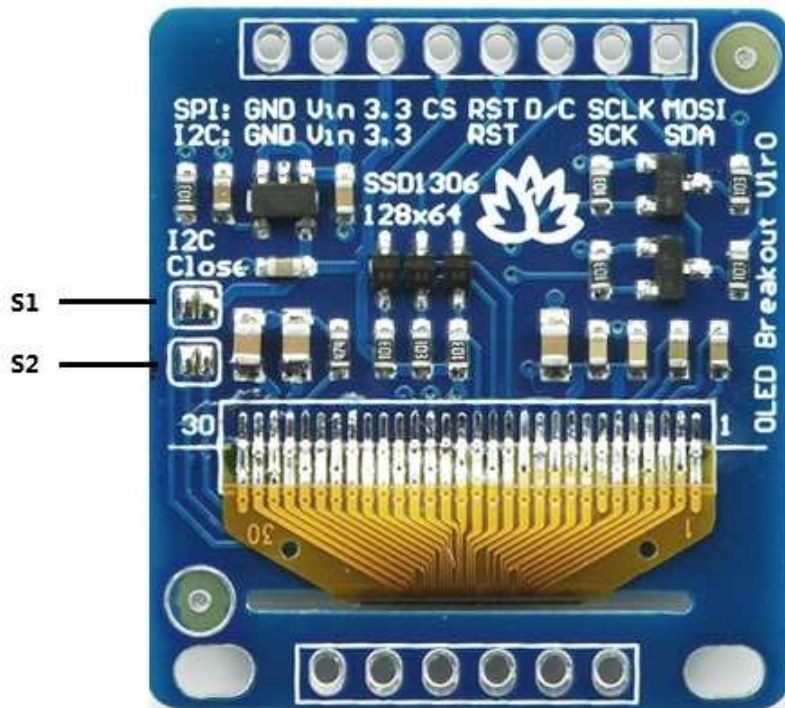
OLED Breakout is very readable due to the high contrast of an OLED display. This display is made of 128x64 individual white OLED pixels, each one is turned on or off by the controller chip. Because the display makes its own light, no backlight is required. This reduces the power required to run the OLED and is why the display has such high contrast; we really like this miniature display for its crispness!

This breakout can be used with either an SPI or I2C interface - selectable by soldering two jumpers on the back. The design is completely 5V-ready, with an onboard regulator and built in boost converter. It's easier than ever to connect directly to your 3V or 5V microcontroller without needing any kind of level shifter!

## SPECIFICATIONS

- PCB: 29mm x 31mm
- Screen: 25mm x 14mm
- Display current draw is completely dependent on your usage: each OLED LED draws current when on so the more pixels you have lit, the more current is used. They tend to draw ~20mA or so in practice but for precise numbers you must measure the current in your usage circuit.

## Overview



### Jumper setting

Function	S1	S2
I2C Mode	Connected	Connected
SPI Mode	Disconnected	Disconnected

# How to use

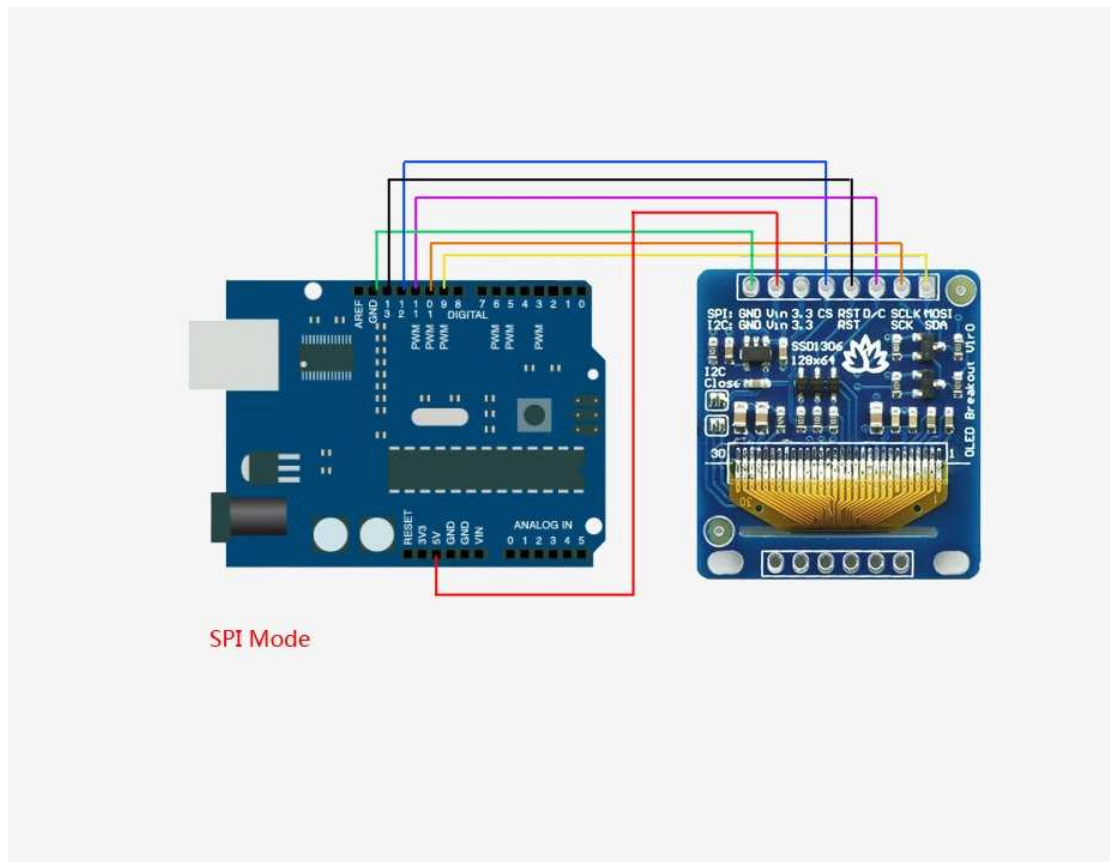
Please download [library](#) from Tinysine website. Unzip the files and copy 'Tinysine\_GFX' and 'Tinysine\_SSD1306' two directory to  
 C:\arduino-1.x.x-windows\arduino-1.0.5\libraries

## SPI Mode

The breakouts are ready for SPI by default, but if you used them for I2C at some point, you'll need to remove the solder jumpers. Use wick or a solder sucker to make sure both are clear!

Finally, connect the pins to your Arduino -

- GND goes to ground
- Vin goes to 5V
- MOSI to digital 9
- SCLK to digital 10
- D/C to digital 11
- RST to digital 13
- CS to digital 12



(Note: If using the display with other SPI devices, D/C, SCLK and MOSI may be shared, but CS must be unique for each device.)

This matches the example code we have written. Once you get this working, you can try another set of pins.

Finally you can run the File→examples→Tinysine\_SSD1306→SSD1306\_128x64\_spi example (Based Arduino 1.01).

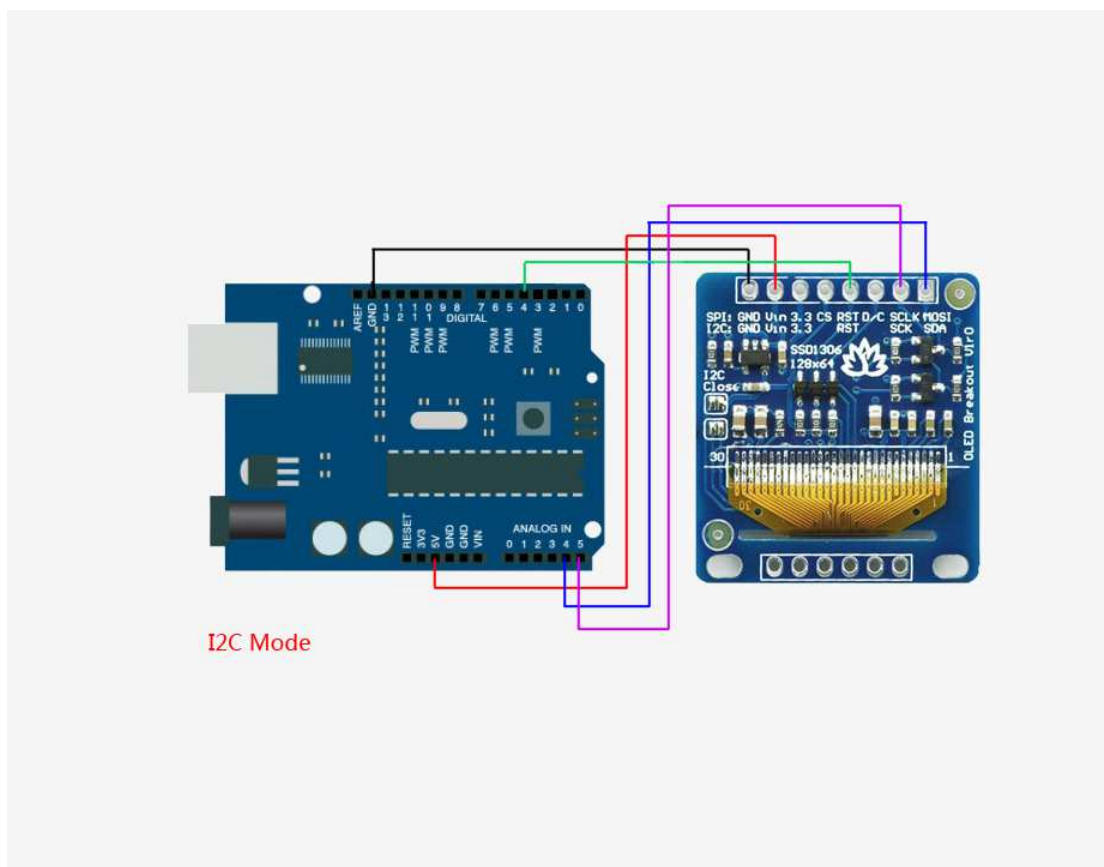
## I2C Mode

The display can be used with any I2C microcontroller. Because the I2C interface is for 'writing' to the display only, you'll still have to buffer the entire 512 byte frame in the microcontroller RAM - you can't read data from the OLED (even though I2C is a bidirectional protocol).

To start, you'll need to solder the two jumpers on the back of the OLED. Both must be soldered 'closed' for I2C to work!

Finally, connect the pins to your Arduino

- GND goes to ground
- Vin goes to 5V
- SDA to I2C SDA (on the Uno, this is A4 on the Mega it is 20 and on the Leonardo digital 2)
- SCK to I2C SCL (on the Uno, this is A5 on the Mega it is 21 and on the Leonardo digital 3)
- RST to digital 4 (you can change this pin in the code, later)



This matches the example code we have written. Once you get this working, you can

try a different Reset pin (you can't change the SCA and SCL pins).

Finally you can run the File→Examples→Tinysine\_SSD1306→SSD1306\_128x64\_i2c  
example(Based Arduino 1.01)