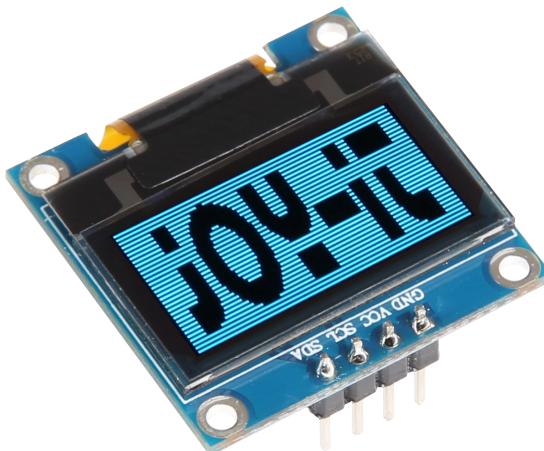


# SBC-OLED01

OLED-Display 128x64 i2c



## 1. ALLGEMEINE INFORMATIONEN

Sehr geehrter Kunde,  
vielen Dank, dass Sie sich für unser Produkt entschieden haben.  
Im Folgenden zeigen wir Ihnen, was bei der Inbetriebnahme und  
der Verwendung zu beachten ist.

Sollten Sie während der Verwendung unerwartet auf Probleme  
stoßen, so können Sie uns selbstverständlich gerne kontaktieren.

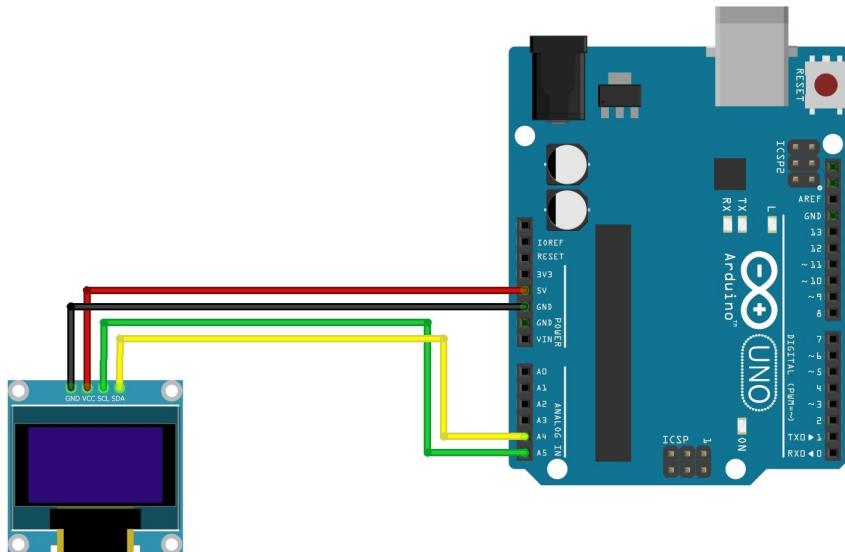
## INDEX

1. Verwendung mit einem Arduino
  - 1.1 Anschließen des Displays
  - 1.2 Installation der Bibliotheken
  - 1.3 Beispiel-Code
2. Verwendung mit einem Raspberry Pi
  - 2.1 Anschließen des Displays
  - 2.2 Einrichtung des Displays
  - 2.3 Installation der Bibliothek
  - 2.4 Kommunikation mit dem Display
  - 2.5 Beispiel-Code
3. Sonstige Informationen
4. Support

## 1. VERWENDUNG MIT EINEM ARDUINO

## Anschießen des Displays

Schließen Sie das Display, wie abgebildet, an die Pins Ihres Arduino UNO an.

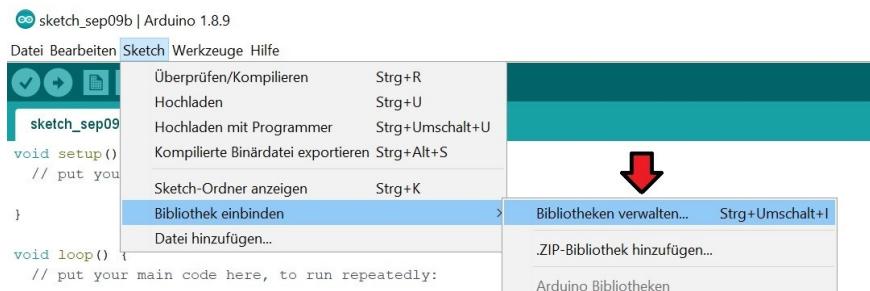


SBC-OLED01	Arduino Uno
GND	GND
VCC	5V oder 3.3V
SCL	A5
SDA	A4

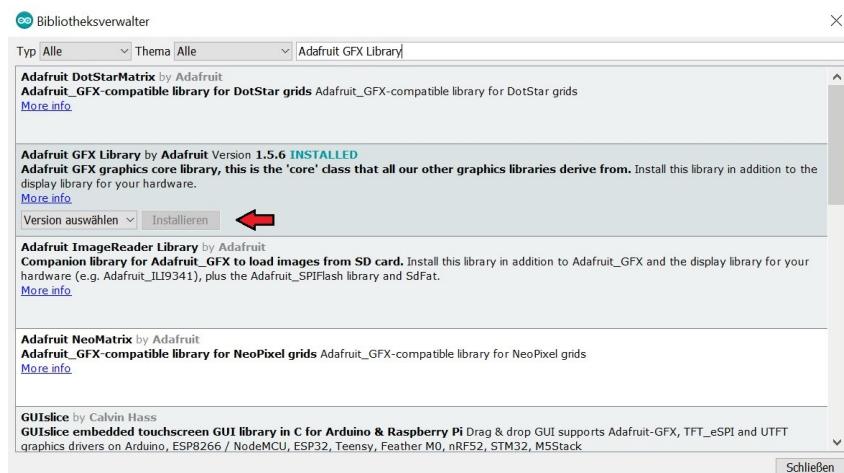
## Installation der Bibliotheken

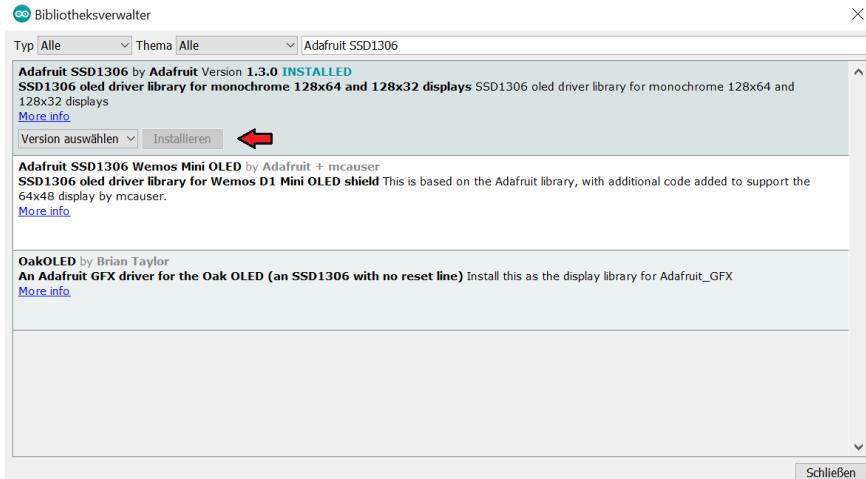
Bevor Sie das Code-Beispiel auf Ihren Arduino übertragen können, müssen Sie zunächst die zusätzlichen Bibliotheken Adafruit GFX Library und Adafruit SSD1306 installieren. Diese Bibliotheken ermöglichen Ihnen eine möglichst einfache und schnelle Verwendung des Displays.

Öffnen Sie hierzu zunächst die Bibliotheksverwaltung in Ihrer Arduino Software.



Suchen Sie nun in dem sich öffnenden Bibliotheksverwalter nach den Bibliotheken **Adafruit GFX Library** und **Adafruit SSD1306** und installieren Sie diese.





Sobald die Bibliotheken installiert sind, ist die Installation des Displays abgeschlossen.

Sie können nun im nächsten Schritt unseren Beispielcode übertragen, um das Display zu testen.

## Beispiel-Code

Im Nachfolgenden können Sie ein Codebeispiel zur Verwendung des Displays entnehmen.

Kopieren Sie diesen Quelltext vollständig auf Ihren Arduino.

Vergewissern Sie sich, dass das korrekte Board und der korrekte Port unter **Werkzeuge** in der Menüleiste ausgewählt ist.

## Beispiel-Code 1 (Text)

```
#include <SPI.h>
#include <Wire.h>
#include <Adafruit_GFX.h>
#include <Adafruit_SSD1306.h>

#define OLED_RESET 4
Adafruit_SSD1306 display(OLED_RESET);

void setup()  {

    display.begin(SSD1306_SWITCHCAPVCC, 0x3C); // 
Adressierung beachten, hier 0x3C!
}

int i;

void loop() {
    display.clearDisplay();
    display.setTextColor(WHITE);
    display.setTextSize(1);
    display.setCursor(23,0);
    display.println("OLED - Display");
    display.setCursor(23,12);
    display.println("www.joy-it.net");
    display.setCursor(36,24);
    display.println("SBC-OLED01");
    display.display();
    delay(8000);
    display.clearDisplay();
    display.invertDisplay(true);
    delay(8000);
    display.invertDisplay(false);
    delay(1000);
}
```

## Beispiel-Code 2 (Grafik)

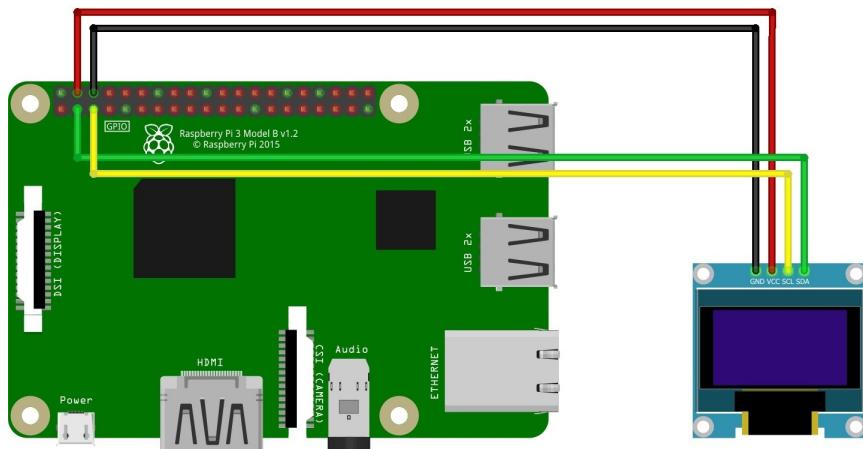
```
void setup()  {
  display.begin(SSD1306_SWITCHCAPVCC, 0x3C);
  display.clearDisplay();
  display.drawBitmap(0, 0, logo, 128, 64, WHITE);
  display.display();
}

void loop() {
```

## 2. VERWENDUNG MIT EINEM RASPBERRY PI

### Anschließen des Displays

Schließen Sie das Display, wie im folgenden Bild, bzw. in der folgenden Tabelle, zu sehen ist, an die Pins Ihres Raspberry Pis an.



SBC-OLED01	Raspberry Pi
GND	Pin 6 (Ground)
VCC	Pin 4 (5V) oder Pin 1 (3.3V)
SCL	Pin 5 (BCM 3 / SCL)
SDA	Pin 3 (BCM 2 / SDA)

## Einrichtung des Displays

Sobald Sie das System gestartet haben, öffnen Sie die Terminal-Konsole und führen Sie folgende Kommandos aus:

```
sudo apt-get update
```

```
sudo apt-get install python-pip python-dev build-essential
```

```
sudo pip install RPi.GPIO
```

```
sudo apt-get install python-pil
```

```
sudo apt-get install python-smbus i2c-tools
```

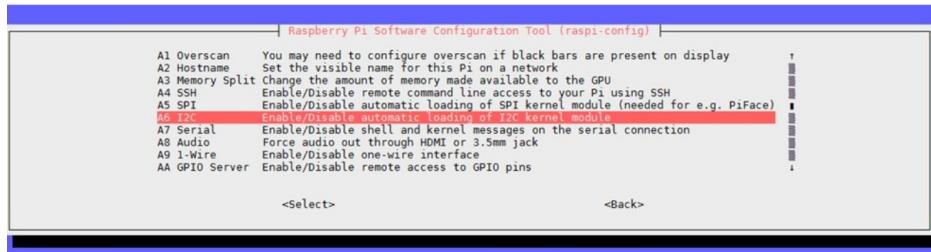
Sollte die I2C-Funktion auf Ihrem Raspberry Pi noch nicht aktiviert sein, so müssen Sie dies nun in den Einstellungen nachholen.

Öffnen Sie hierzu die Einstellungen mit folgendem Befehl:

```
sudo raspi-config
```

In dem sich nun öffnenden Fenster wählen Sie die Option **Interfacing Options**.

Hier wählen und aktivieren Sie die Option **I2C**.



## Installation der Bibliothek

Da Sie nun die I2C-Funktion aktiviert haben, machen wir uns eine Python-Bibliothek mit folgenden Befehlen zunutze:

```
sudo apt-get install git
```

```
git clone https://github.com/adafruit/  
Adafruit_Python_SSD1306.git
```

```
cd Adafruit_Python_SSD1306
```

```
sudo python setup.py install
```

## Kommunikation mit dem Display

Nun beginnen wir mit einer ersten Kommunikation zum Display. Führen Sie das folgende Kommando aus:  
(Bitte beachten Sie, dass das Display hierfür bereits angeschlossen sein muss)

```
sudo i2cdetect -y 1
```

Die Ausgabe zeigt nun das folgende an:

	0	1	2	3	4	5	6	7	8	9	a	b	c	d	e	f
00:	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--
10:	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--
20:	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--
30:	--	--	--	--	--	--	--	--	--	--	--	3c	--	--	--	--
40:	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--
50:	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--
60:	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--
70:	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--

Die eben gestartete Prüfung hat ein I2C-Signal an der Adresse 3C festgestellt. Dies ist die Standard-Adresse des Displays und ist in der von uns verwendeten Python-Bibliothek schon vorausgewählt.

## Achtung:

Sollte die Adresse des I2C-Signals an der Adresse **3D** liegen, müssen wir diese in unserer Python-Bibliothek ändern.

Lautet die Adresse **3C** können Sie mit dem nächsten Punkt **Beispiel-Code** fortfahren.

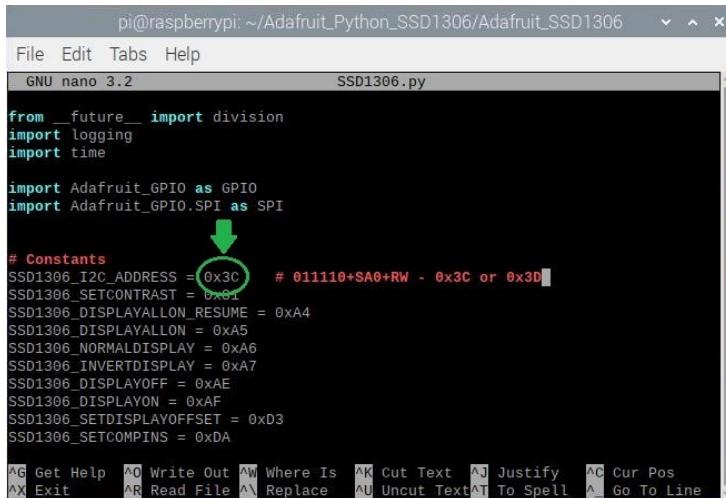
Hierfür müssen wir zunächst in den erforderlichen Unterordner wechseln mit folgendem Kommando:

```
cd Adafruit_SSD1306
```

Öffnen Sie nun die Datei die geändert werden muss mit dem Befehl:

```
sudo nano SSD1306.py
```

Ändern Sie nun im (auf der Abbildung grün) markierten Bereich die Adresse von 3D zu 3C um.



```
pi@raspberrypi: ~/Adafruit_Python_SSD1306/Adafruit_SSD1306
```

```
File Edit Tabs Help
```

```
GNU nano 3.2 SSD1306.py
```

```
from __future__ import division
import logging
import time

import Adafruit_GPIO as GPIO
import Adafruit_GPIO.SPI as SPI

# Constants
SSD1306_I2C_ADDRESS = 0x3C # 011110+SA0+RW - 0x3C or 0x3D
SSD1306_SETCONTRAST = 0x81
SSD1306_DISPLAYALLON_RESUME = 0xA4
SSD1306_DISPLAYALLON = 0xA5
SSD1306_NORMALDISPLAY = 0xA6
SSD1306_INVERTDISPLAY = 0xA7
SSD1306_DISPLAYOFF = 0xAE
SSD1306_DISPLAYON = 0xAF
SSD1306_SETDISPLAYOFFSET = 0xD3
SSD1306_SETCOMPINS = 0xDA
```

```
^G Get Help ^O Write Out ^W Where Is ^K Cut Text ^J Justify ^C Cur Pos
^X Exit ^R Read File ^N Replace ^U Uncut Text ^T To Spell ^L Go To Line
```

Speichern und Verlassen Sie die Datei mit der Tastenkombination:

**STRG + O , ENTER , STRG + X**

Verlassen Sie, bevor Sie mit der Anleitung fortfahren, den Unterordner mit diesem Befehl:

```
cd ..
```

## Beispiel-Code

Im nachfolgenden können Sie ein Beispiel-Code zur Verwendung des Displays entnehmen.

Wir wechseln in das Verzeichnis unserer Bibliothek und erstellen dort eine neue Datei mit folgenden Befehlen:

```
cd examples
```

```
sudo nano text.py
```

Kopieren Sie diesen Code vollständig in Ihre geöffnete Datei:

### **Beispiel-Code 1 (Text)**

```
import time
import Adafruit_GPIO.SPI as SPI
import Adafruit_SSD1306

from PIL import Image, ImageDraw, ImageFont

RST = 24

disp = Adafruit_SSD1306.SSD1306_128_64(rst=RST)

disp.begin()

disp.clear()
disp.display()
```

```
width = disp.width
height = disp.height
image = Image.new('1', (width, height))

draw = ImageDraw.Draw(image)

font = ImageFont.load_default()

x = 40
top = 20

draw.text((x, top),      'SBC-OLED01',
font=font, fill=255)
draw.text((x, top+20), 'Joy-IT', font=font,
fill=255)

disp.image(image)
disp.display()
time.sleep(.1)
```

Speichern und Verlassen Sie nun die Datei mit der Tastenkombination:

**STRG + O , ENTER , STRG + X**

Der Beispiel-Code ist somit fertig und wird mit folgendem Kommando ausgeführt:

```
sudo python text.py
```

### 3. SONSTIGE INFORMATIONEN

Unsere Informations- und Rücknahmepflichten nach dem ElektroGesetz (ElektroG)



**Symbol auf Elektro- und Elektronikgeräten:**

Diese durchgestrichene Mülltonne bedeutet, dass Elektro- und Elektronikgeräte **nicht** in den Hausmüll gehören. Sie müssen die Altgeräte an einer Erfassungsstelle abgeben.

Vor der Abgabe haben Sie Altbatterien und Altakkumulatoren, die nicht vom Altgerät umschlossen sind, von diesem zu trennen.

**Rückgabemöglichkeiten:**

Als Endnutzer können Sie beim Kauf eines neuen Gerätes, Ihr Altgerät (das im Wesentlichen die gleiche Funktion wie das bei uns erworbene neue erfüllt) kostenlos zur Entsorgung abgeben.

Kleingeräte bei denen keine äußere Abmessungen größer als 25 cm sind können unabhängig vom Kauf eines Neugerätes in Haushaltsüblichen Mengen abgeben werden.

**Möglichkeit Rückgabe an unserem Firmenstandort während der Öffnungszeiten:**

Simac GmbH, Pascalstr. 8, D-47506 Neukirchen-Vluyn

**Möglichkeit Rückgabe in Ihrer Nähe:**

Wir senden Ihnen eine Paketmarke zu mit der Sie das Gerät kostenlos an uns zurücksenden können. Hierzu wenden Sie sich bitte per E-Mail an Service@joy-it.net oder per Telefon an uns.

**Informationen zur Verpackung:**

Verpacken Sie Ihr Altgerät bitte transportsicher, sollten Sie kein geeignetes Verpackungsmaterial haben oder kein eigenes Nutzen möchten kontaktieren Sie uns, wir lassen Ihnen dann eine geeignete Verpackung zukommen.

## 4. SUPPORT

Wir sind auch nach dem Kauf für Sie da. Sollten noch Fragen offen bleiben oder Probleme auftauchen stehen wir Ihnen auch per E-Mail, Telefon und Ticket-Supportsystem zur Seite.

E-Mail: [service@joy-it.net](mailto:service@joy-it.net)

Ticket-System: <http://support.joy-it.net>

Telefon: +49 (0)2845 98469 – 66 (10 - 17 Uhr)

Für weitere Informationen besuchen Sie unsere Website:  
[www.joy-it.net](http://www.joy-it.net)

Veröffentlicht: 02.07.2020

[www.joy-it.net](http://www.joy-it.net)

SIMAC Electronics GmbH

Pascalstr. 8 47506 Neukirchen-Vluyn