

## Willkommen!

Und herzlichen Dank für den Kauf unseres AZ-Delivery 1,3" I<sup>2</sup>C OLED Display. Auf den folgenden Seiten gehen wir mit dir gemeinsam die ersten Programmierschritte durch.

**Viel Spaß!**



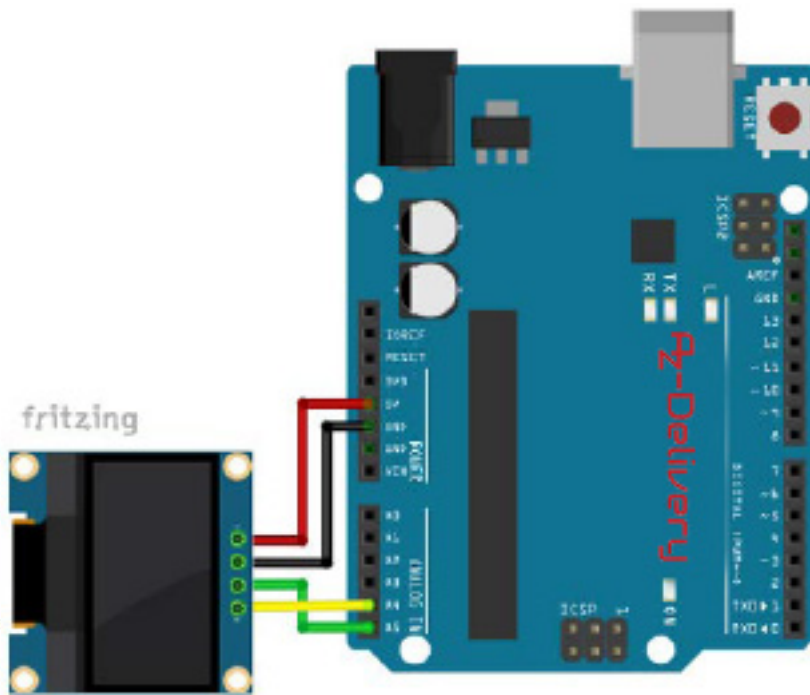
Das Display mit 1,3" hat eine Auflösung von 128 x 64 Pixel und einen starken Kontrast. Zur Ansteuerung steht eine i<sup>2</sup>c Schnittstelle mit dem Standard-Controller SH1106 zur Verfügung.

## Programmieren des OLED-Displays mit der u8g2 Bibliothek

### Verdrahten des Moduls mit einem Atmega328P Board:

**VDD** wird mit **5V** am Board verbunden  
**GND** wird mit **GND** verbunden  
**SCK** wird mit **SCL** verbunden  
**SDA** wird mit **SDA** verbunden

Rote Leitung  
Schwarze Leitung  
Grüne Leitung  
Gelbe Leitung



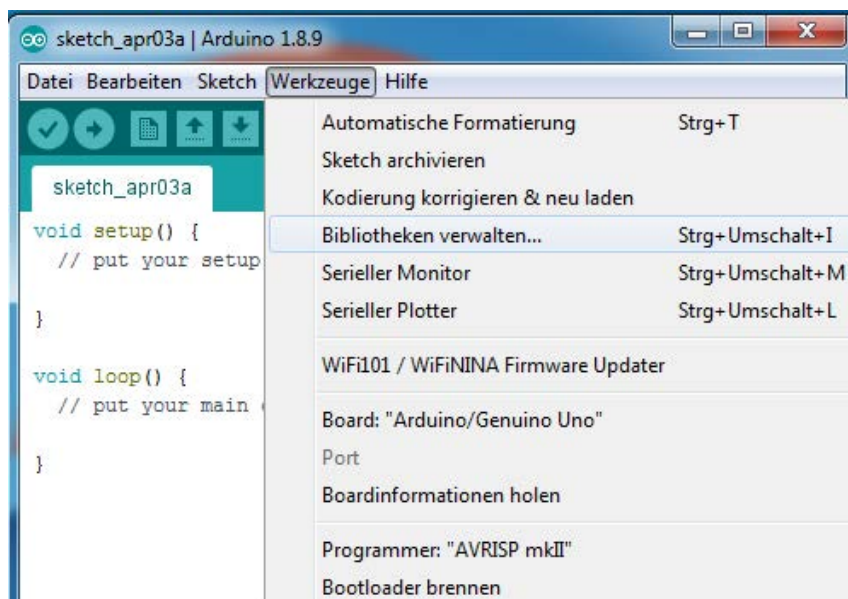
## Vorbereiten der Software:

Die Software sehen wir in diesem Schritt als Installiert an, sollte diese bei dir noch fehlen, so kannst du diese unter <https://www.arduino.cc/en/Main/Software#> herunterladen und auf deinen PC installieren.

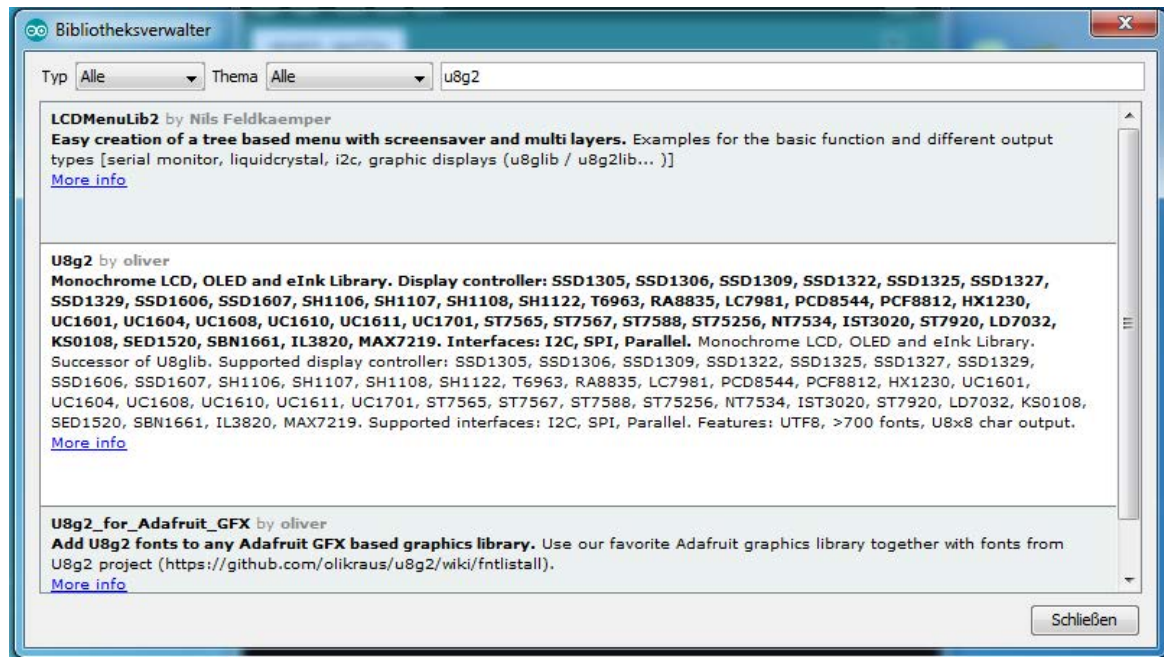
## Ansteuern des OLED-Displays:

Für die Ansteuerung des Displays benötigen wir noch die entsprechenden Bibliotheken (Informationen) in der Software.

Starten wir unter Werkzeuge > Bibliotheken verwalten ...



den Bibliotheksverwalter und suchen dort nach „u8g2“



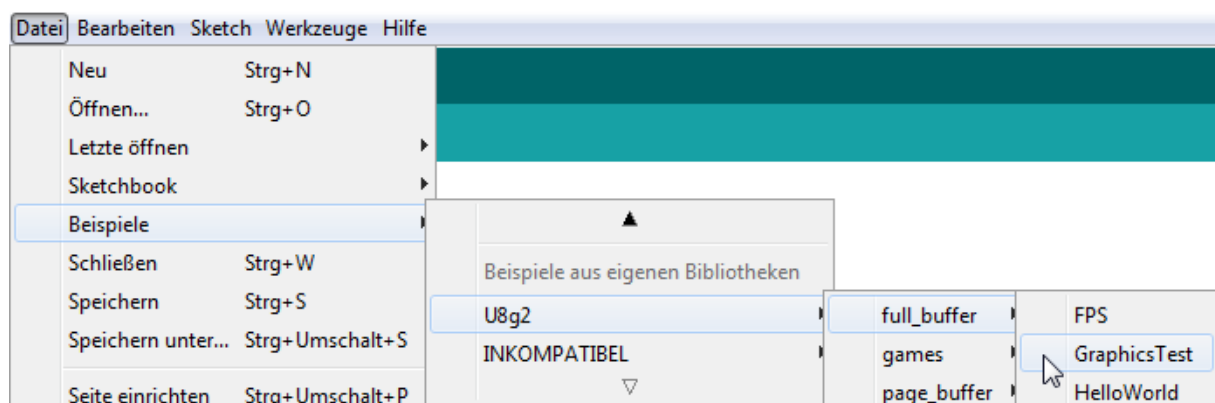
und klicken rechts unten auf Installieren, nachdem das Paket angewählt wurde.

Nach ein paar Sekunden Wartezeit erscheint „INSTALLED“

**U8g2** by **oliver** Version 2.25.10 **INSTALLED**  
**Monochrome LCD, OLED and eInk Library.** Display controller: **SSD1305, SSD1306, SSD1309, SSD1322, SSD1325, SSD1327, SSD1329, SSD1606, SSD1607, SH1106, SH1107, SH1108, SH1122, T6963, RA8835, LC7981, PCD8544, PCF8812, HX1230, UC1601, UC1604, UC1608, UC1610, UC1611, UC1701, ST7565, ST7567, ST7588, ST75256, NT7534, IST3020, ST7920, LD7032, KS0108, SED1520, SBN1661, IL3820, MAX7219.** Interfaces: **I2C, SPI, Parallel.** Monochrome LCD, OLED and eInk Library. Successor of U8glib. Supported display controller: **SSD1305, SSD1306, SSD1309, SSD1322, SSD1325, SSD1327, SSD1329, SSD1606, SSD1607, SH1106, SH1107, SH1108, SH1122, T6963, RA8835, LC7981, PCD8544, PCF8812, HX1230, UC1601, UC1604, UC1608, UC1610, UC1611, UC1701, ST7565, ST7567, ST7588, ST75256, NT7534, IST3020, ST7920, LD7032, KS0108, SED1520, SBN1661, IL3820, MAX7219.** Supported interfaces: **I2C, SPI, Parallel.** Features: **UTF8, >700 fonts, U8x8 char output.**  
[More info](#)

Nun Schließen wir das Fenster und können mit dem Programmieren loslegen.


Wählen wir unter Beispiele > U8g2 > full\_buffer > GraphicsTest aus:



Es wird nun ein langer Code geöffnet, in den ersten Zeilen sind sehr viele Displaytypen eingetragen, diese sind aber mit den „//“ am Zeilenanfang auskommentiert. Für unser Display müssen wir nun diese Zeile suchen und aktivieren, indem wir die // am Anfang der Zeile entfernen:

# AZ-Delivery

```
//U8G2_SH1106_128X64_NONAME_F_HW_I2C u8g2(U8G2_R0, /* reset=*/  
U8X8_PIN_NONE);
```

Nach dem übertragen  zeigt das Display nun Demotexte und Bilder an.

Basierend auf dieser Demonstration können wir auch einen Lauftext programmieren.

Hinweis möchte man einen längeren Lauftext machen, muss in der u8g2.h Datei die 16 Bit Unterstützung aktiviert werden. Die Datei findet ihr in euerem Atmega328P Board

Verzeichnis unter:

Arduino\libraries\U8g2\src\clib\u8g2.h

In der Zeile 72 steht: `//#define U8G2_16BIT`

Dies wird geändert auf: `#define U8G2_16BIT`

Anschließend die Datei speichern und den Code neu Übertragen.

Hier folgt der Code:

```
#include <U8g2lib.h>  
U8G2_SH1106_128X64_NONAME_F_HW_I2C u8g2(U8G2_R0, /* reset=*/  
U8X8_PIN_NONE);  
u8g2_uint_t offset;  
u8g2_uint_t width;  
const char *text = "AZ-Delivery";  
void setup(void) {  
  u8g2.begin();  
  u8g2.setFont(u8g2_font_logisoso32_tf);  
  width = u8g2.getUTF8Width(text);  
  u8g2.setFontMode(0);  
}  
  
void loop(void) {  
  for (int i = 0 ; i < 128 + width*3 ; i++) {  
    u8g2.firstPage();  
    u8g2.setFont(u8g2_font_logisoso32_tf);  
    u8g2.drawUTF8(128 - i, 48, text);  
    u8g2.nextPage();  
  }  
  u8g2.clearBuffer();  
}
```

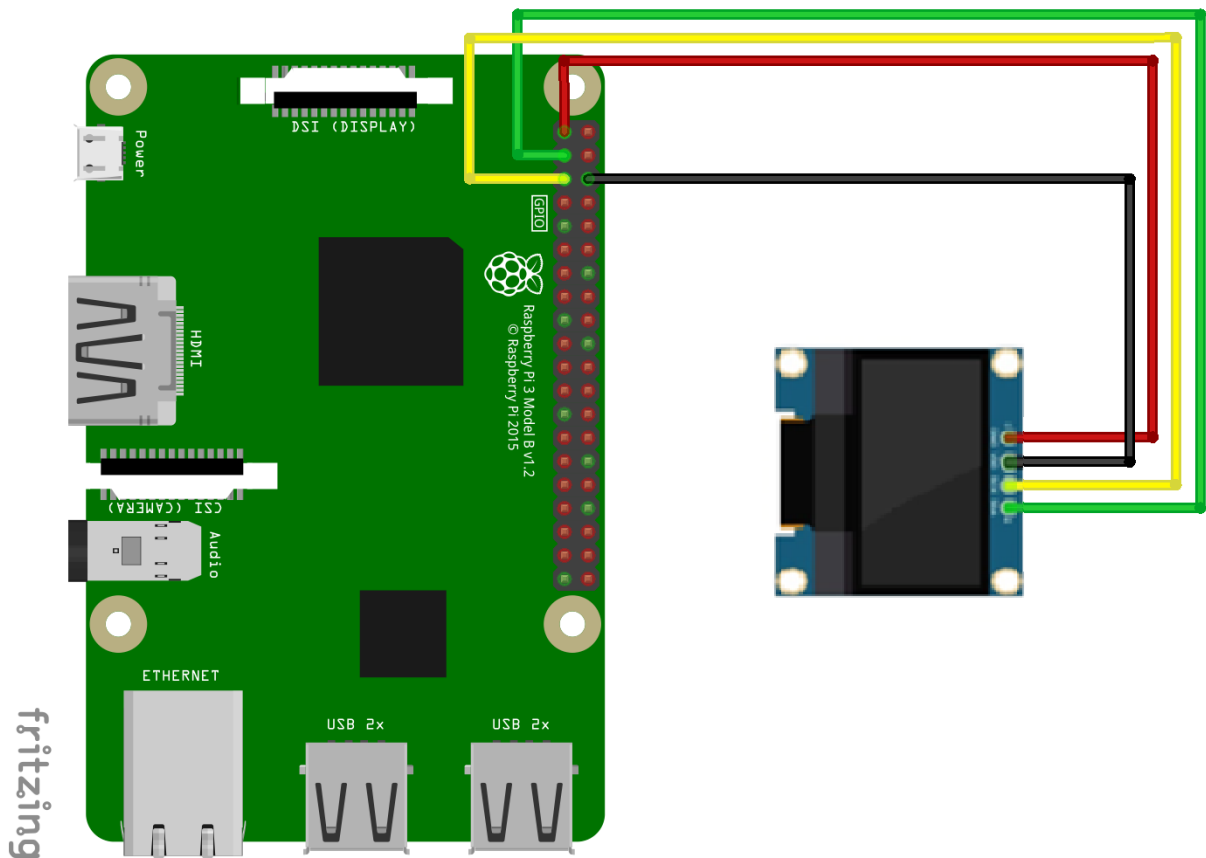
**Du hast es geschafft, du kannst nun in für deine Projekte ein OLED-Display mit der u8g2 Bibliothek verwenden!**

## Programmieren des OLED-Displays mit einem Raspberry Pi

### Verdrahten des Moduls mit einem Raspberry Pi:

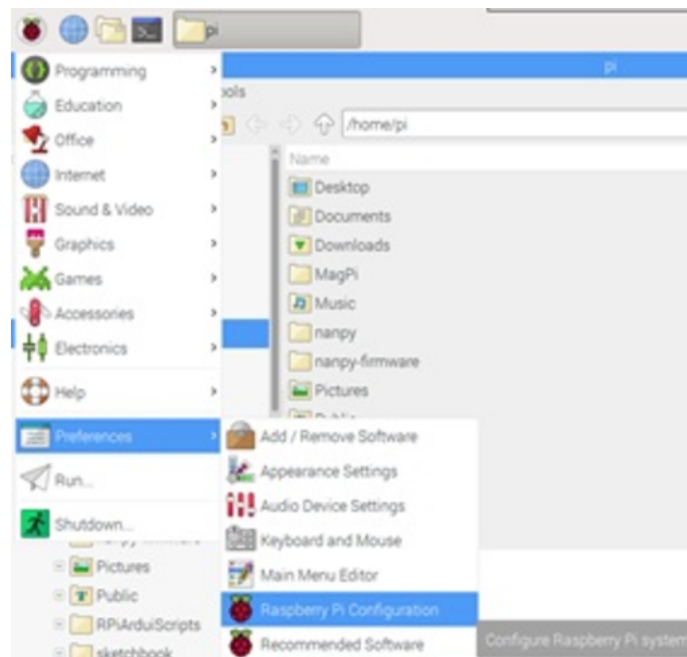
VDD wird mit **5V** am Raspberry Pi verbunden  
GND wird mit **GND** verbunden  
SCK wird mit **SCL** verbunden  
SDA wird mit **SDA** verbunden

Rote Leitung  
Schwarze Leitung  
Grüne Leitung  
Gelbe Leitung

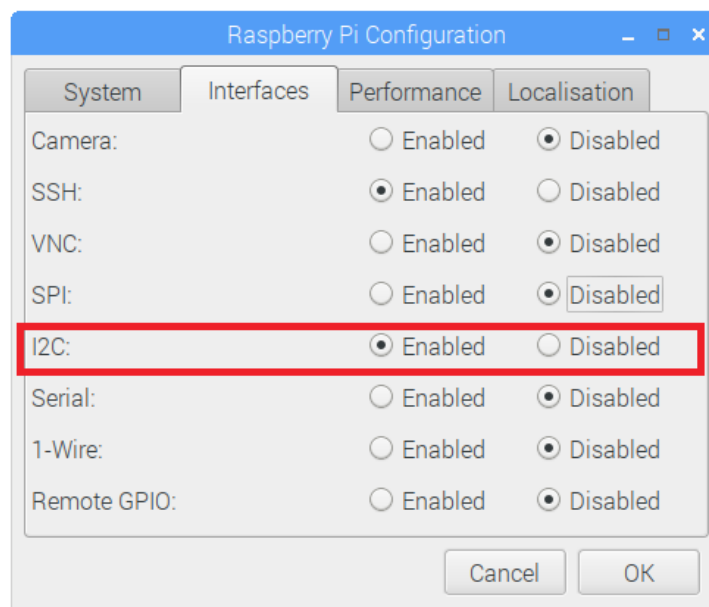


## Aktivierung der I2C-Schnittstelle auf Raspbian

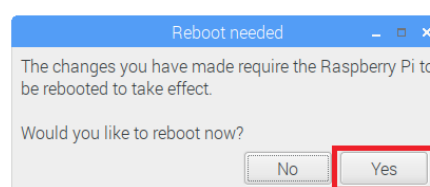
Dafür gehen Sie zu Preferences > Raspberry Pi Configuration wie unten abgebildet:



Als nächstes öffnen Sie den "Interfaces" Tab, finden Sie die "I2C" Taste und und schalten diese an, wie unten abgebildet:



Sie werden aufgefordert, das System neu zu starten. Wir empfehlen Ihnen, dies durch Klicken auf "Ja" zu tun, wie in der Abbildung unten dargestellt:





## Finden der Adresse des OLED-Moduls

Wenn es aktiviert ist, werden wir den Befehl `i2cdetect` verwenden, um das Modul am I2C-Bus zu finden:

**`i2cdetect -y 1`**

Das Ergebnis sollte so aussehen, wie unten abgebildet:

```
pi@rpi3py:~ $ i2cdetect -y 1
    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:  --  --  --  --  --  --  --  --  --  --  --  --  --  --  --
```

Unser Gerät wurde mit der Adresse "0x3c" erkannt. Dies ist die Standardadresse für diese Art von Gerät.



## Python-library

Für die Anzeige von Formen, Text und Bildern werden wir ein Python-library verwenden. Auf dem aktuellen Raspberry Pi OS ist Python3, pip3 und git bereits vorinstalliert, sollte das jedoch nicht der Fall sein können Sie das ganze mit folgenden Befehlen nachinstallieren:

```
sudo apt-get install python3-dev libffi-dev libssl-dev python3-pil libjpeg-dev  
zlib1g-dev libfreetype6-dev liblcms2-dev libopenjp2-7 libtiff5 -y
```

```
sudo apt-get install python3-rpi.gpio python3-pip -y
```

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

Als Bibliothek verwenden wir die "luma.oled" diese kann mit folgendem Befehl installiert werden:

```
sudo -H pip3 install luma.oled
```



## Python-Skript

Im Ordner "pi" erstellen wir jetzt einen Ordner namens "oled" und begeben uns in diesen Ordner

```
sudo mkdir oled  
cd oled
```

luma.oled bietet viele Beispiele und diese können wir uns mit folgendem Befehl herunterladen:

```
sudo git clone https://github.com/rm-hull/luma.examples  
mit:
```

```
cd luma.examples/examples/
```

wechseln wir in den Ordner in dem die Beispiele liegen.

mit:

```
python3 demo.py
```

können wir eines der Beispiele starten. Sollte auf dem Display ein weißes rauschen zu sehen sein, muss der richtige controller übergeben werden. dies kann man mit:

```
python3 demo.py --device [controller]
```

luma.oled nimmt standardmäßig den SSD1306 her, hat man zu Beispiel SH1106 würde das starten des Skriptes wie folgt aussehen:

```
python3 demo.py --device ssh1106
```

## **Du hast es geschafft, du kannst nun in für deine Raspberry Pi Projekte ein OLED-Display verwenden!**

Jetzt sind Sie dran! Entwickeln Sie Ihre eigenen Projekte und Smart- Home Installationen. Wie Sie das bewerkstelligen können, zeigen wir Ihnen unkompliziert und verständlich auf unserem Blog. Dort bieten wir Ihnen Beispielskripte und Tutorials mit interessanten kleinen Projekten an, um schnell in die Welt der Mikroelektronik einzusteigen. Zusätzlich bietet Ihnen auch das Internet unzählige Möglichkeiten, um sich in Sachen Mikroelektronik weiterzubilden.

**Falls Sie noch nach weiteren hochwertigen Produkten für Raspberry Pi suchen, sind Sie bei der AZ-Delivery Vertriebs GmbH goldrichtig. Wir bieten Ihnen zahlreiche Anwendungsbeispiele, ausführliche Installationsanleitungen, Ebooks, Bibliotheken und natürlich die Unterstützung unserer technischen Experten.**

<https://az-delivery.de>

Viel Spaß!

Impressum

<https://az-delivery.de/pages/about-us>