Willkommen!

Und herzlichen Dank für den Kauf unseres AZ-Delivery 1,3“ I²C OLED Display. Auf den folgenden Seiten gehen wir mit dir gemeinsam die ersten Programmierschritte durch.

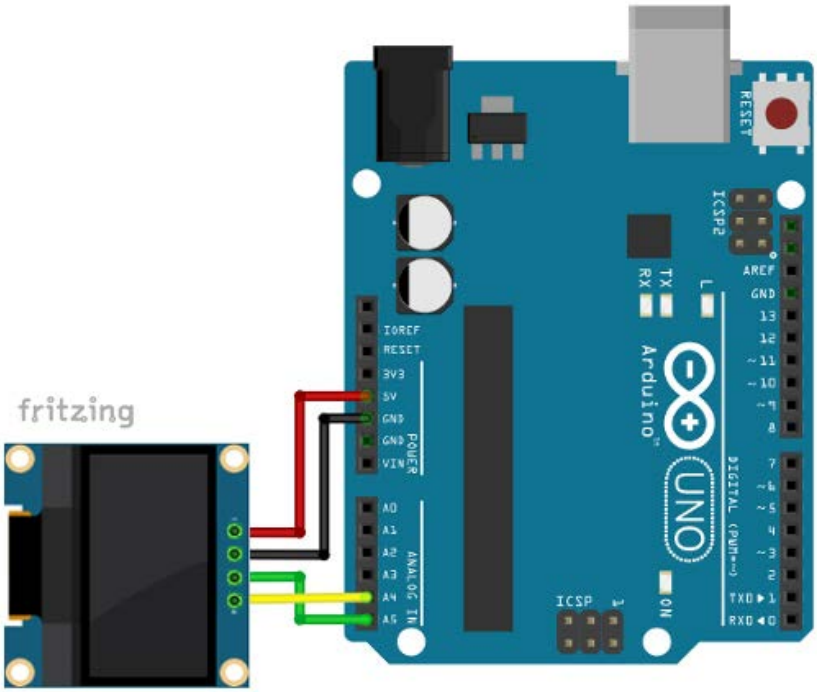
Viel Spaß!

Das Display mit 1,3“ hat eine Auflösung von 128 x 64 Pixel und einen starken Kontrast. Zur Ansteuerung steht eine i²c Schnittstelle mit dem Standard-Controller SH1106 zur Verfügung.

Programmieren des OLED-Displays mit der u8g2 Bibliothek

Verdrahten des Moduls mit einem Arduino Uno:

VDD wird mit **5V** am Arduino verbunden	Rote Leitung
GND wird mit **GND** verbunden	Schwarze Leitung
SCK wird mit **SCL** verbunden	Grüne Leitung
SDA wird mit **SDA** verbunden	Gelbe Leitung



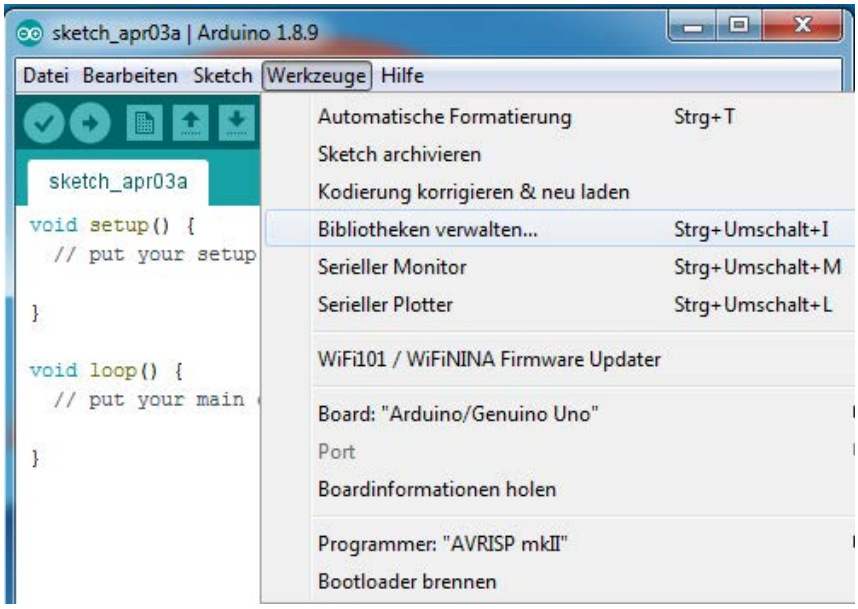
Vorbereiten der Software:

Die Arduino 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.

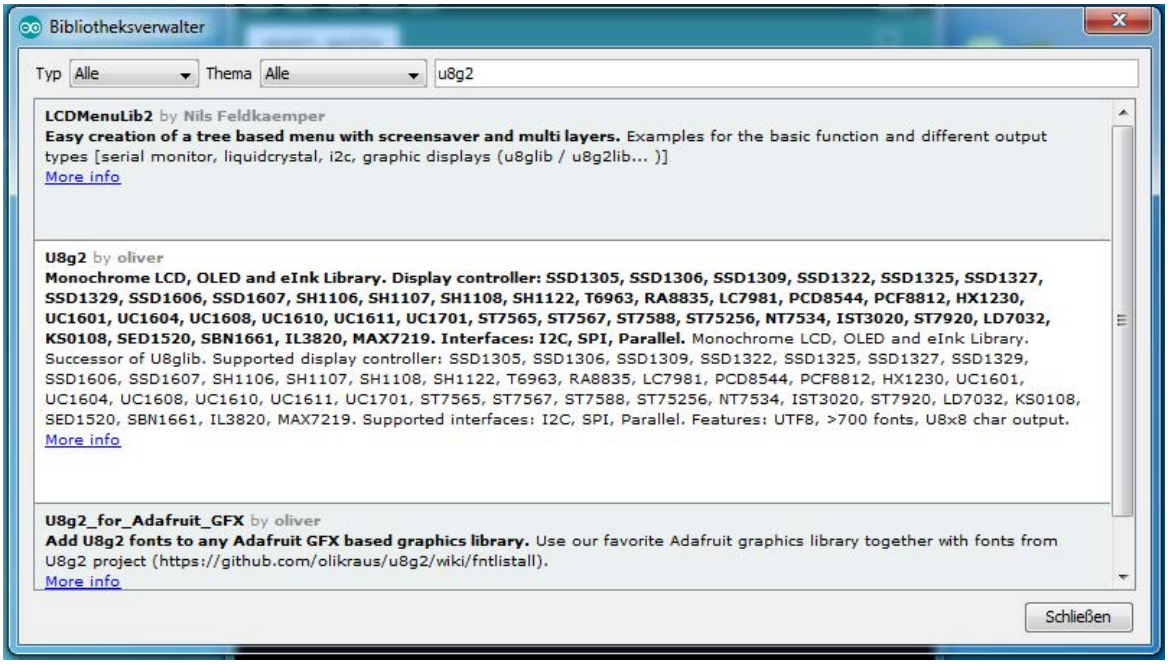
Ansteuern des OLED-Displays:

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

Starten wir unter Werkzeuge > Bibliotheken verwalten …

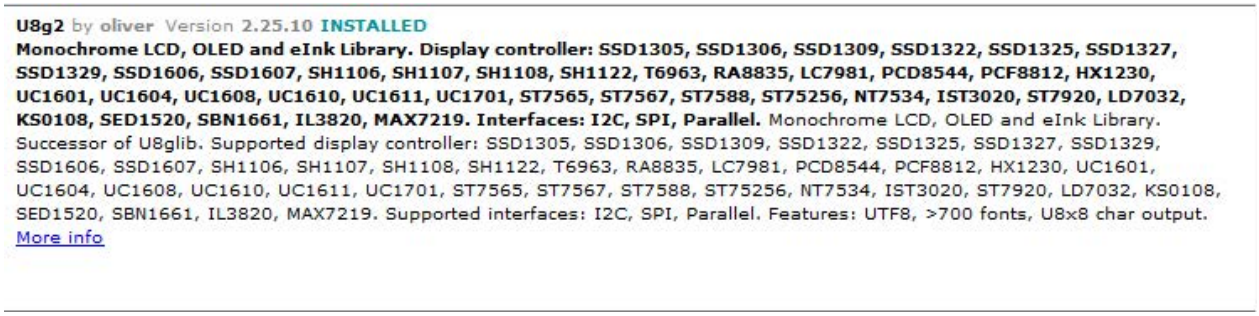


den Bibliotheksverwalter und suchen dort nach „u8g2“



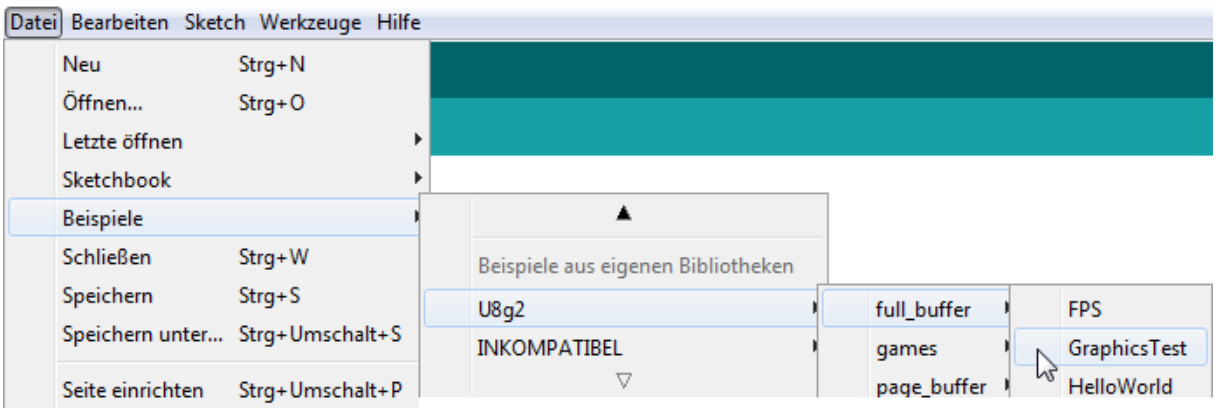
und klicken rechts unten auf Installieren, nachdem das Paket angewählt wurde.

Nach ein paar Sekunden Wartezeit erscheint „INSTALLED“



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:

```
//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 Arduino 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:

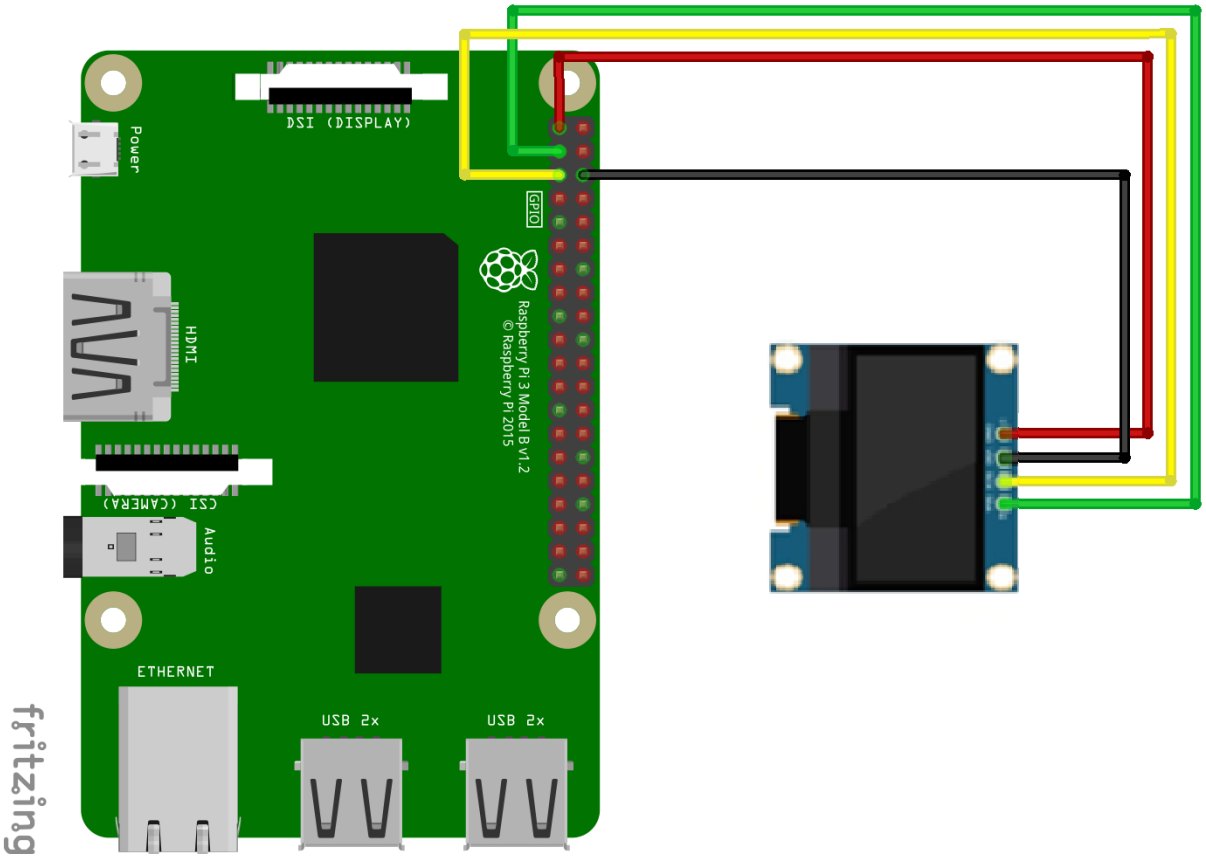
```
#include <U8g2lib.h>
U8G2_SH1106_128X64_NONAME_F_HW_I2C u8g2(U8G2_R0, /* reset=*/
U8X8_PIN_NONE);
u8g2_uint_t offset;
u8g2_uint_t width;
const char *text = "AZ-Delivery";
void setup(void) {
  u8g2.begin();
  u8g2.setFont(u8g2_font_logisoso32_tf);
  width = u8g2.getUTF8Width(text);
  u8g2.setFontMode(0);
}

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 Arduino 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 — Rote Leitung
GND wird mit **GND** verbunden — Schwarze Leitung
SCK wird mit **SCL** verbunden — Grüne Leitung
SDA wird mit **SDA** verbunden — Gelbe Leitung



Vorbereiten der Software:

Der Raspberry sollte entsprechend dem eBook für Raspberry Pi vorbereitet werden und aktualisiert werden.

Anschließend aktivieren die I²C Schnittstelle am Raspberry. Dazu gehen wir in die Konfiguration des Raspberry Pi.

```
sudo raspi-config
```

Nun wählen wir unter „Interfacing Options“ den Punkt „I2C“ und bestätigen mit <Yes>

Anschließend benötigen wir noch ein paar System-Programme:

```
sudo apt-get install -y python-dev python3-dev python-imaging python-smbus i2c-tools git python3-pip python-setuptools build-essential git-core libi2c-dev i2c-tools lm-sensors python-pip
```

Mit dem Befehl `i2cdetect -y 1` bekommen wir folgende Ausgabe, wenn das Display richtig verdrahtet wurde:

```
pi@raspberrypi:~ $ 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: -- -- -- -- -- -- -- --
```

Wenn das Display erfolgreich erkannt wurde, können wir nun die benötigten Bibliotheken herunterladen und installieren:

```
git clone git://github.com/rm-hull/ssd1306.git
```

```
cd ssd1306
```

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

```
cd examples
```

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

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

Wenn das alles nun erledigt wurde können wir verschiedene Demos starten:

Analog Uhr mit Datum	`python clock.py --display sh1106`
Text und Logo (animiert)	`python crawl.py --display sh1106`
3D Animation	`python sprite_animation.py --display sh1106`
gif (animiert) Player	`python animated_gif.py --display sh1106`

Um Systeminfos anzeigen zu lassen benötigen wir noch psutil:

```
sudo pip install psutil
```

```
python sys_info.py --display sh1106
```

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

Ab jetzt heißt es Experimentieren.

Und für mehr Hardware sorgt natürlich dein Online-Shop auf:

https://az-delivery.de

Viel Spaß!
Impressum

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