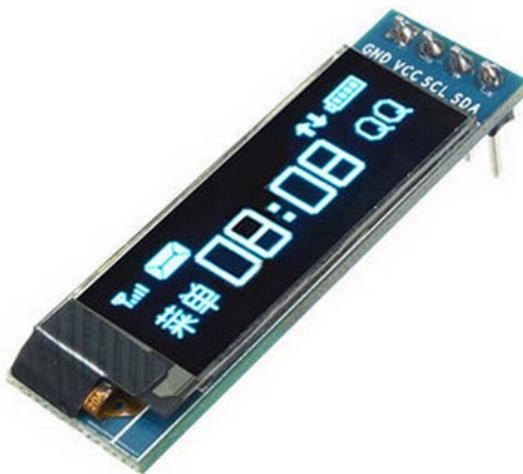


Willkommen!

Und herzlichen Dank für den Kauf unseres AZ-Delivery 0,91“ i²c OLED Display. Auf den folgenden Seiten gehen wir mit dir gemeinsam die ersten Programmierschritte durch.

Viel Spaß!



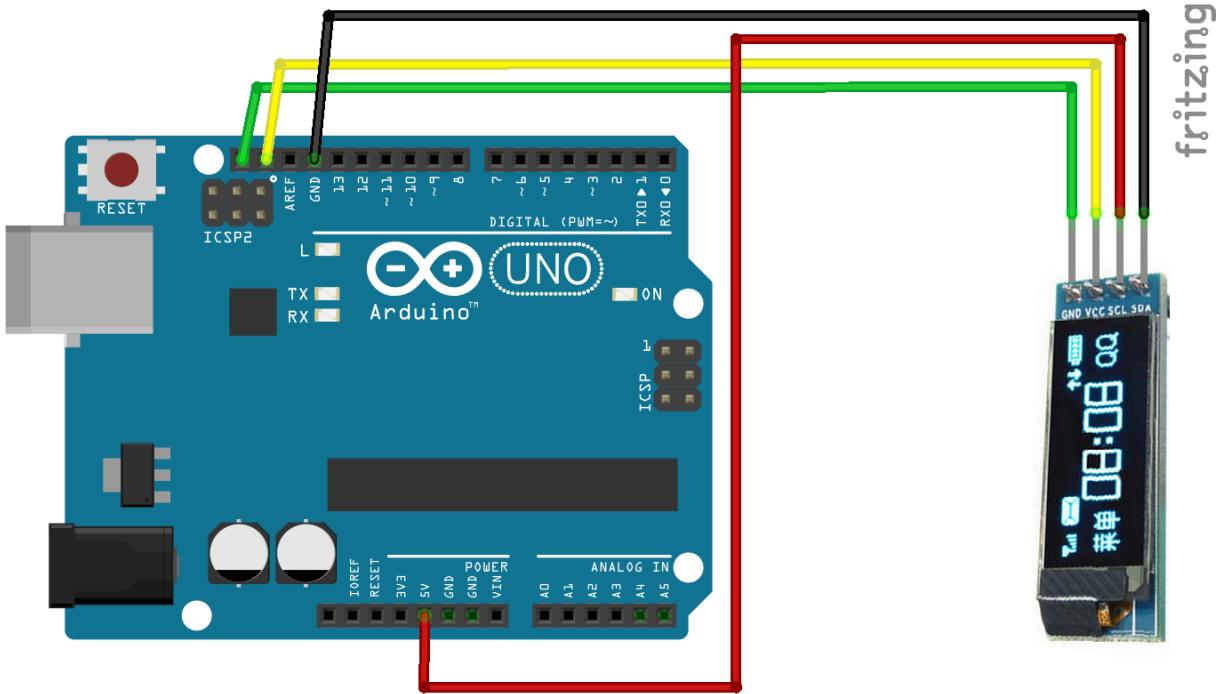
Das Display mit 0,91“ hat eine Auflösung von 128 x 32 Pixel und einen starken Kontrast. Zur Ansteuerung steht eine i²c Schnittstelle mit dem Standard-Controller SSD 1306 zur Verfügung.

Programmieren des OLED-Display

Verdrahten des Moduls mit einem Arduino Uno:

+5V wird mit **5V** am Arduino 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:

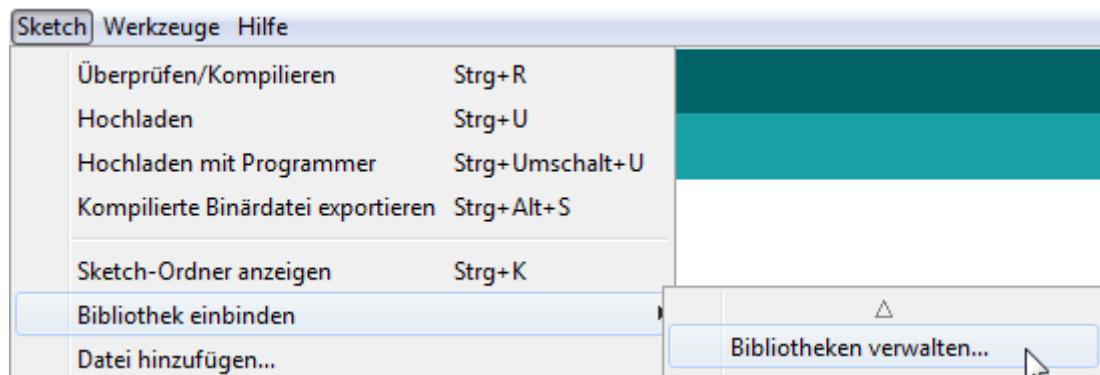
Die Arduino Software sehen wir in diesem Schritt als installiert an, sollte diese bei dir noch fehlen, so kannst du diese unter <https://www.arduino.cc/en/Main/Software#> herunterladen und auf deinen PC installieren.

Der Arduino Code:

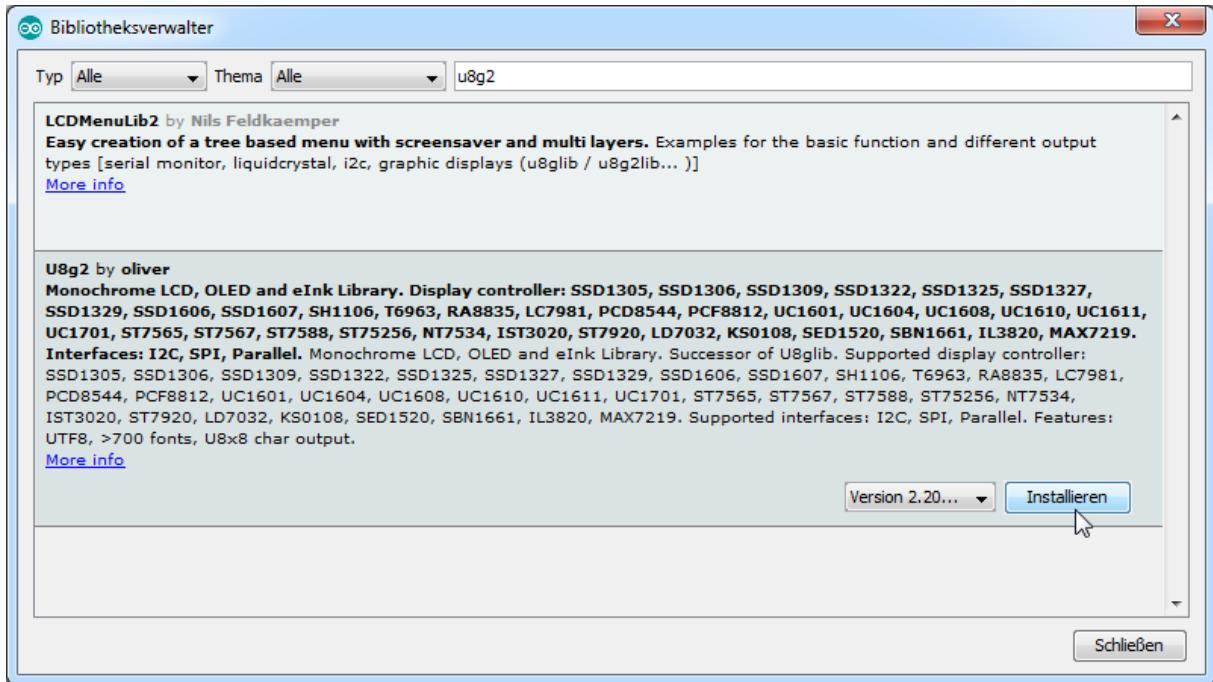
Ansteuern des OLED-Displays:

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

Starten wir unter Sketch > Bibliothek einbinden > Bibliotheken verwalten ...



den Bibliotheksverwalter und suchen dort nach „u8g2“



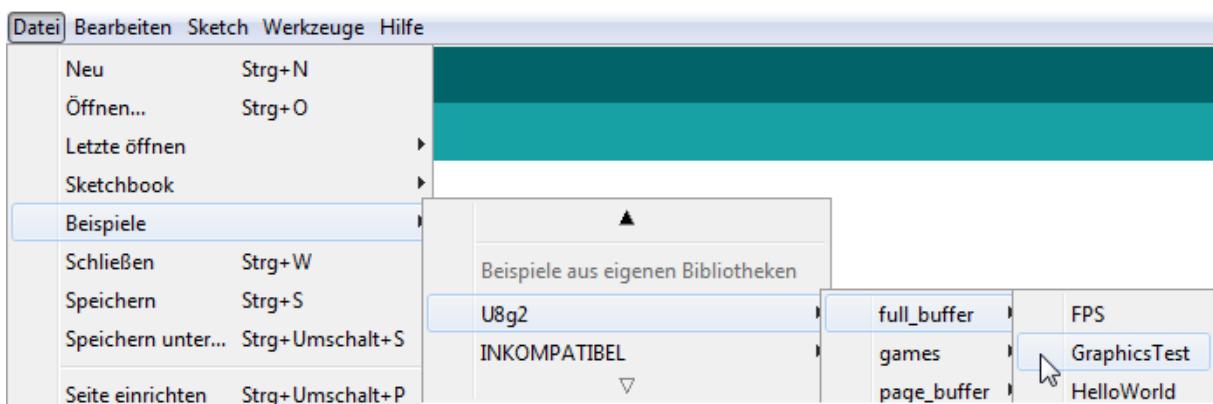
und klicken rechts unten auf Installieren, nachdem das Paket angewählt wurde.

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_SSD1306_128X32_UNIVISION_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\arduino_168079\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_SSD1306_128X32_UNIVISION_F_HW_I2C u8g2(U8G2_R0, /* reset=*/
U8X8_PIN_NONE);
u8g2_uint_t offset;
u8g2_uint_t width;
const char *text = "Florian ";
void setup(void) {
    u8g2.begin();
    u8g2.setFont(u8g2_font_logisoso32_tf);
    width = u8g2.getUTF8Width(text);
    u8g2.setFontMode(0);
}

void loop(void) {
    u8g2_uint_t x;
    u8g2.firstPage();
    do {
        x = offset;
        u8g2.setFont(u8g2_font_logisoso32_tf);
        do {
            u8g2.drawUTF8(x, 32, text);
            x += width;
        } while ( x < u8g2.getDisplayWidth() );
        u8g2.setFont(u8g2_font_logisoso32_tf);
        u8g2.setCursor(0, 64);
        u8g2.print(width);
    } while ( u8g2.nextPage() );
    offset -= 1;
    if ( (u8g2_uint_t)offset < (u8g2_uint_t) - width ){
        offset = 0;
        delay(1000);
        u8g2.clearBuffer();
    }
}
```

**Du hast es geschafft, du kannst nun in für deine Projekte
OLED-Display programmieren und verwirklichen!**

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>