

Willkommen!

Und herzlichen Dank für den Kauf unseres AZ-Delivery 0,96" i²c OLED Display. Auf den folgenden Seiten gehen wir mit dir gemeinsam die ersten Programmierschritte durch.

Viel Spaß!



Das Display mit 0,96" hat eine Auflösung von 128 x 64 Pixel und einen starken Kontrast. Zur Ansteuerung steht eine i²c Schnittstelle mit dem Standard-Controller SSD 1306 zur Verfügung.

Programmieren des OLED-Displays mit der u8g2 Bibliothek

Verdrahten des Moduls mit einem Arduino Uno:

VCC wird mit 5V am Arduino verbunden GND wird mit GND verbunden SCK wird mit A5 verbunden SDA wird mit A4 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 <u>https://www.arduino.cc/en/Main/Software#</u> 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 Sketch > Bibliothek einbinden > Bibliotheken verwalten ...

Skete	h Werkzeuge Hilfe			
	Überprüfen/Kompilieren	Strg+R		
	Hochladen	Strg+U		
	Hochladen mit Programmer	Strg+Umschalt+U		
	Kompilierte Binärdatei exportieren	Strg+Alt+S		
	Sketch-Ordner anzeigen	Strg+K		
	Bibliothek einbinden	1	Δ	
	Datei hinzufügen	Bibliotheken verwalten	A.	

den Bibliotheksverwalter und suchen dort nach "u8g2"

Bibliotheksverwalte	r			
yp Alle 👻	Thema	Alle	•	u8g2
CDMenuLib2 by Ni Easy creation of a ypes [serial monit More info	ils Feldk t ree bas or, liquio	aemper sed menu with s dcrystal, i2c, gra	screen aphic d	saver and multi layers. Examples for the basic function and different output displays (u8glib / u8g2lib)]
onochrome LCD,	OLED ar	nd eInk Library.	. Displ	av controller: SSD1305 SSD1306 SSD1300 SSD1322 SSD1325 SSD1327
IC1701, ST7565, S IC1701, ST7565, S Interfaces: 12C, SF SD1305, SSD1306 CD8544, PCF8812 ST3020, ST7920, ITF8, >700 fonts, Inter info	5, SSD16 T7567, 9 PI, Para 5, SSD13 2, UC160 LD7032, U8x8 ch	507, SH1106, T 517588, ST7525 Ilel. Monochrom 809, SSD1322, S 01, UC1604, UC KS0108, SED1 ar output.	5963, F 6, NT7 ne LCD SSD13 C1608, 520, S	RA8835, LC7981, PCD8544, PCF8812, UC1601, UC1604, UC1608, UC1610, UC1611, 7534, IST3020, ST7920, LD7032, KS0108, SED1520, SBN1661, IL3820, MAX7219. D, OLED and eInk Library. Successor of U8glib. Supported display controller: 125, SSD1327, SSD1329, SSD1606, SSD1607, SH1106, T6963, RA8835, LC7981, UC1610, UC1611, UC1701, ST7565, ST7567, ST7588, ST75256, NT7534, 3BN1661, IL3820, MAX7219. Supported interfaces: I2C, SPI, Parallel. Features:
UC1701, ST7565, S Interfaces: 12C, SF SSD1305, SSD1306 PCD8544, PCF8812 IST3020, ST7920, I JTF8, >700 fonts, I More info	5, SSD10 T7567, S PI, Para 5, SSD13 2, UC160 LD7032, U8x8 ch	507, SH1106, TG 517588, ST7525 Ilel. Monochrom 309, SSD1322, S 309, SSD1322, S 301, UC1604, UC KS0108, SED1 ar output.	5963, F 6, NT7 ne LCD SSD13 :1608, 520, S	RA8835, LC7981, PCD8544, PCF8812, UC1601, UC1604, UC1608, UC1610, UC1611, 7534, IST3020, ST7920, LD7032, KS0108, SED1520, SBN1661, IL3820, MAX7219. D, OLED and eInk Library. Successor of U8glib. Supported display controller: 125, SSD1327, SSD1329, SSD1606, SSD1607, SH1106, T6963, RA8835, LC7981, UC1610, UC1611, UC1701, ST7565, ST7567, ST7588, ST75256, NT7534, IBN1661, IL3820, MAX7219. Supported interfaces: I2C, SPI, Parallel. Features: Version 2.20

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:

Date	ij Bearbeiten Sketo	h Werkzeuge Hilfe:		
	Neu	Strg+N		
	Öffnen	Strg+O		
	Letzte öffnen	1		
	Sketchbook	1		
	Beispiele	1	▲	
	Schließen	Strg+W	Beispiele aus eigenen Bibliotheken	
	Speichern Strg+S	Strg+S	U8g2 full	buffer FPS
	Speichern unter	Strg+Umschalt+S	INKOMPATIBEL gam	es GraphicsTest
	Seite einrichten	Strg+Umschalt+P	⊽ pag	e_buffer I 😽 HelloWorld

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_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\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_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++ ){</pre>
  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-Display Mit der Adafruit Bibliothek

Verdrahten des Moduls mit einem Arduino Uno:

VCC wird mit 5V am Arduino verbunden GND wird mit GND verbunden SCK wird mit A5 verbunden SDA wird mit A4 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 <u>https://www.arduino.cc/en/Main/Software#</u> 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 Sketch > Bibliothek einbinden > Bibliotheken verwalten ...



Sketo	h Werkzeuge Hilfe		
	Überprüfen/Kompilieren	Strg+R	
	Hochladen	Strg+U	
	Hochladen mit Programmer	Strg+Umschalt+U	
	Kompilierte Binärdatei exportieren	Strg+Alt+S	
	Sketch-Ordner anzeigen	Strg+K	
	Bibliothek einbinden	1	Δ
	Datei hinzufügen		Bibliotheken verwalten

den Bibliotheksverwalter und suchen dort nach "Adafruit_SSD1306"

(💿 Bibliotheksverwalter	
	Typ Alle 🔹 Thema Alle	✓ Adafruit_SSD1306
	Adafruit SSD1306 by Adafruit SSD1306 oled driver library for mon and 128x32 displays More info	Installieren

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

Nach ein paar Sekunden Wartezeit erscheint "INSTALLED"



Dies wiederholen wir mit der "Adafruit GFX" Bibliothek:

💿 в	ibliothe	ksverwalte	er															
Тур	Alle	•	Thema	Alle		•	Adafruit (GFX										
A A ac <u>M</u>	dafruit dafruit ddition	GFX Libra GFX grap to the dis	nry by A hics cor play libr	dafruit e library ary for y	this is to our hard	: he ' ¢ ware	core' class	s that a	all our o	ther g	raphics	librarie	es der	rive fror	n. Insta	ill this l	library i	n
														Version	1.3.6 👻	Ins	tallierer	R
Adaf Adaf addit <u>More</u>	ruit GF ruit GF ion to f	X Library X graphic the displa	by Ada s core l y library	ifruit Ver ibrary, t y for you	sion 1.3 his is the hardwa	.6 I e'co ire.	NSTALLEI re' class t	D that all	l our ot	her gra	aphics l	ibraries	5 deri	ive from	n. Insta	ll this l	ibrary i	n

Nun muss man noch die passende Displayauflösung einstellen. Das AZ-Delivery OLED Display hat eine Breite von 128 Pixeln und eine Höhe von 64 Pixeln. Um die Auflösung zu ändern, muss man die SSD_1306.h ändern, welche sich im Order SSD_1306 befindet, der sich im libraries Order befindet, welcher sich im Arduino Ordner befindet.



Nun Schließen wir das Fenster und können mit dem Programmieren loslegen. Wählen wir unter Beispiele > Adafruit SSD1306 > ssd1306_128x64_i2c aus:

💿 sketch_jan19a Arduino 1.8.5	Adafruit GFX Library	•	
Datei Bearbeiten Sketch Werkzeuge Hilfe	Adafruit MCP23017 Arduino Library		
Neu Stra+N	Adafruit NeoPixel		
Öfferer Strey O	Adafruit SSD1306	•	ssd1306_128x32_i2c
Strg+0	Adafruit WS2801 Library		ssd1306_128x32_spi
Letzte offnen	DallasTemperature	1	ssd1306_128x64_i2c
Sketchbook	DS3231		ssd1306 128x64 spi
Beispiele	DS3232RTC	▶ T	
Schließen Strg+W	Elegon TETLCD		
Speichern Strg+S	Keened		
Speichern unter Strg+Umschalt+S	Keypad		
	NeoPixelBus by Makuna		
Seite einrichten Strg+Umschalt+P	OneWire	•	
Drucken Strg+P	rc-switch	•	
Versinstellungen Strav Komma	RTClib	۱.	
Voreinstellungen Strg+Komma	TheThingsNetwork	F .	
Beenden Strg+Q	TouchScreen	۶.	
	U8g2	۶.	
	\bigtriangledown		

Es wird nun ein fertiger Code geöffnet, im Teil setup müssen nur noch die i²c Adresse auf 3C angepasst werden und die Spannungsversorgung aktiviert werden:

```
void setup() {
  Serial.begin(9600);

SSD1306_SWITCHCAPVCC; // = generate display voltage from 3.3V internally
  if(!display.begin(SSD1306_SWITCHCAPVCC, 0x3C)) { // Address 0x3D for 128x64
    Serial.println(F("SSD1306 allocation failed"));
    for(;;); // Don't proceed, loop forever
  }
```

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

Du hast es geschafft, du kannst nun in für deine Arduino Projekte ein OLED-Display mit der Adafruit 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



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.

<mark>sudo raspi-config</mark>

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 i2ctools git python3-pip python-setuptools build-essential git-core libi2c-dev i2ctools lm-sensors python-pip

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

pi@1	casp	bei	rry	pi:•		i20	det	tect	: -7	/ 1						
	0	1	2	3	4	5	6	7	8	9	a	b	С	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

<mark>cd ssd1306</mark>

sudo python setup.py install

<mark>cd examples</mark>

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.pydisplay ssd1306
Text und Logo (animiert)	python crawl.pydisplay ssd1306
3D Animation	<pre>python sprite_animation.pydisplay ssd1306</pre>
gif (animiert) Player	python animated_gif.pydisplay ssd1306

Um Systeminfos anzeigen zu lassen benötigen wir noch psutil:

sudo pip install psutil

python sys_info.py --display ssd1306

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