

— GRAFIKDISPLAY • I2C

0,96" OLED SSD1306

Das Standarddisplay für Mikrocontroller:
128 × 64 weiße Pixel über vier Leitungen.
Erstes Bild in rund 20 Minuten – mit UNO R3
und etwas Löten.



AUFLÖSUNG

128 × 64 Pixel

CONTROLLER

SSD1306

VERSORGUNG

3,3 / 5 V DC

SCHNITTSTELLE

I2C 0x3C

01

Verkabeln

4 Leitungen zum UNO

02

Einrichten

IDE, Treiber, Bibliothek

03

Hochladen

Beispiel-Sketch

04

Anzeigen

Text und Grafik

01 Auf einen Blick

Das 0,96-Zoll-OLED leuchtet selbst – ohne Hintergrundbeleuchtung, mit sattem Schwarz und hohem Kontrast. Über I2C reichen vier Leitungen, weitere Sensoren können am selben Bus hängen.

Technische Daten

Display	OLED, weiß, selbstleuchtend
Auflösung	128 × 64 Pixel
Controller	SSD1306
Schnittstelle	I2C, Adresse 0x3C
Versorgung	3,3 bis 5 V DC
Blickwinkel	über 160°
Maße	ca. 27 × 27 × 4 mm
Anschluss	4 Pads, Stiftleiste liegt bei

Lieferumfang

- 1 × 0,96" OLED-Displaymodul
- 1 × Plastikbox für den Transport



Nur I2C

Dieses Modul hat vier Pins und spricht ausschließlich I2C. Anleitungen für 7-Pin-SPI-Varianten passen nicht.



Stiftleiste einlöten

Die Pins sind nicht vorgelötet. Ohne Lötkolben lässt sich das Display nicht aufs Steckbrett stecken.

Pinbelegung

PIN	FUNKTION	SPANNUNG	HINWEIS
● GND	Masse GND	0 V	Gemeinsame Masse.
● VCC	Versorgung POWER	3,3–5 V DC	Auf der Platine als „VDD“ beschriftet. Am UNO an 5V.
● SCL	I2C Takt I2C	3,3 V Logik	Aufdruck „SCK“. Am UNO an A5.
● SDA	I2C Daten I2C	3,3 V Logik	Am UNO an A4.

Die Beschriftung steht oben auf der Platine: GND · VDD · SCK · SDA. VDD ist VCC, SCK ist SCL.

Was du zusätzlich brauchst

UNO R3 mit USB-C

ATmega328P, CH340G-USB-Wandler
MKR-UNO-R3-013

Breadboard + Jumper-Set

400 Kontakte, 40 Dupont-Kabel
MKR-SET-BB400-JUMPER

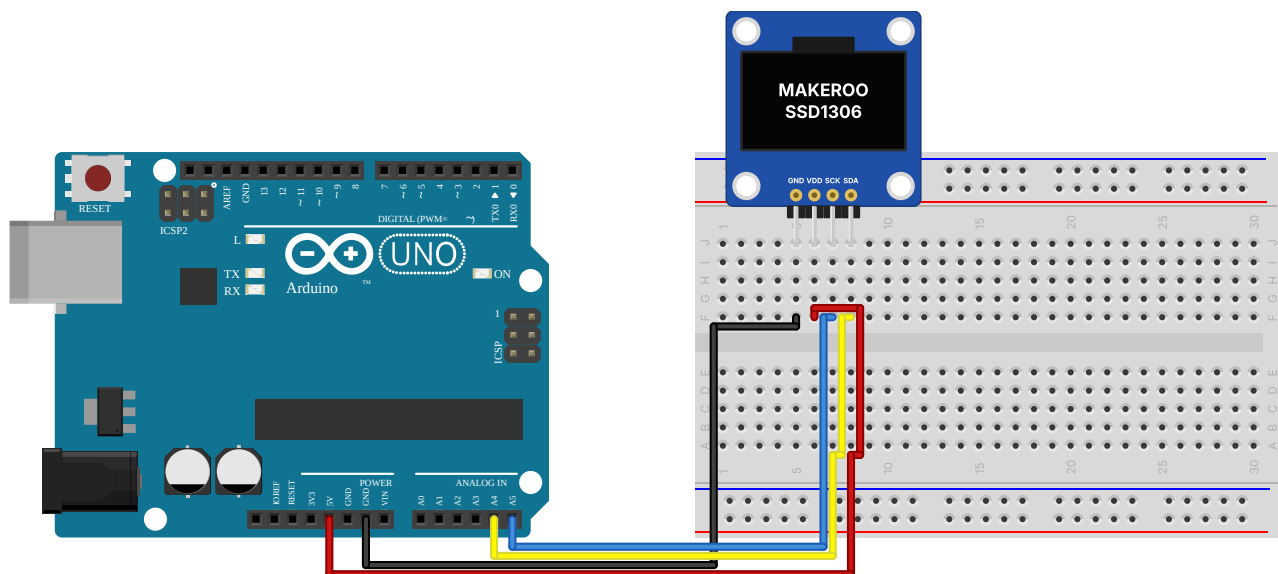
Lötkolben

Stiftleiste einlöten (z. B. TOOLCRAFT LS-30E)
MKR-LS30E-70

02 Verkabeln

Vier Leitungen genügen – zwei für Strom, zwei für I2C.

STECKPLATINEN-ANSICHT



Verbindungen

DISPLAY		UNO R3	LEITUNG
● GND	→	GND	Masse
● VCC	→	5V	Versorgung
● SCL	→	A5	I2C Takt
● SDA	→	A4	I2C Daten

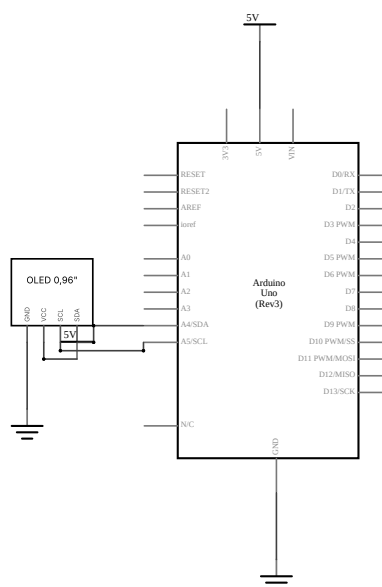
i Beschriftung lesen

Auf der Platine heißen die Pins VDD und SCK – das sind VCC und SCL.

! Erst prüfen, dann einstecken

Vertauschte VCC und GND können das Display zerstören.

SCHALTPLAN



Verdrahtung erstellt mit Fritzing

03 Einrichten & Hochladen

Schritte 1 bis 3 machst du nur ein einziges Mal – danach ist dein UNO R3 für alle weiteren Projekte startklar.

1 Arduino IDE installieren

Lade die aktuelle Arduino IDE 2 von arduino.cc/en/software herunter und installiere sie.

2 Treiber für den CH340G prüfen

Der UNO R3 nutzt den CH340G als USB-Seriell-Wandler. Aktuelle Systeme erkennen ihn meist automatisch. Taucht in Schritt 3 kein Port auf, installiere den Treiber „CH341SER“ des Chipherstellers WCH (wch-ic.com).

3 Board und Port wählen

UNO per USB-C anschließen. Dann **Werkzeuge** → **Board** → **Arduino AVR Boards** → **Arduino Uno** wählen und unter **Werkzeuge** → **Port** den neu aufgetauchten COM-Port (Windows) bzw. `/dev/ttyUSB0` (Linux) / `/dev/cu.wchusbserial...` (macOS).

4 Bibliothek installieren

Bibliotheksverwalter öffnen (`Strg` + `Umschalt` + `I`), nach **Adafruit SSD1306** von Adafruit suchen und installieren. Abhängigkeiten (u. a. *Adafruit GFX Library*, *Adafruit BusIO*) mit **Alle installieren** bestätigen.

5 Sketch hochladen

Neuen Sketch anlegen, den Code von Seite 5 einfügen und mit dem Pfeil-Symbol hochladen (`Strg` + `U`). Unten erscheint „Hochladen abgeschlossen“.

6 Seriellen Monitor öffnen (optional)

`Strg` + `Umschalt` + `M` , unten rechts **9600 Baud**. Dort erscheint „Display bereit.“ – sonst „Display nicht gefunden!“.



i Schnelltest

Nach dem Upload erscheint sofort „MAKEROO“. Die Laufzeit zählt jede Sekunde hoch, der Balken wächst mit.

Nachbildung der Anzeige – berechnet aus dem Sketch auf Seite 5, Pixel für Pixel.

04

Der erste Sketch

Zeigt Titel, Laufzeit und einen wachsenden Balken – und damit die drei Werkzeuge Text, Linien und Rechtecke.

```
MAKER00_OLED096_Start.ino

1 // MAKER00 · 0,96" OLED SSD1306 (I2C) · Erster Start
2 #include <Wire.h>
3 #include <Adafruit_GFX.h>
4 #include <Adafruit_SSD1306.h>
5
6 const byte ADRESSE = 0x3C;           // I2C-Adresse
7 Adafruit_SSD1306 oled(128, 64, &Wire, -1); // 128 x 64, ohne Reset-Pin
8
9 void setup() {
10   Serial.begin(9600);
11   if (!oled.begin(SSD1306_SWITCHCAPVCC, ADRESSE)) {
12     Serial.println(F("Display nicht gefunden!"));
13     while (true) {}
14   }
15   Serial.println(F("Display bereit.));
16 }
17
18 void loop() {
19   unsigned long s = millis() / 1000; // Sekunden seit Start
20   oled.clearDisplay();
21   oled.setTextColor(SSD1306_WHITE);
22   oled.setTextSize(2);                // 12 x 16 Pixel je Zeichen
23   oled.setCursor(22, 0);
24   oled.print(F("MAKER00"));
25   oled.drawLine(0, 18, 127, 18, SSD1306_WHITE);
26   oled.setTextSize(1);                // 6 x 8 Pixel je Zeichen
27   oled.setCursor(0, 26);
28   oled.print(F("Laufzeit: "));
29   oled.print(s);
30   oled.print(F(" s"));
31   oled.drawRect(0, 44, 128, 12, SSD1306_WHITE); // Rahmen
32   oled.fillRect(2, 46, (s % 11) * 124 / 10, 8, SSD1306_WHITE); // Balken
33   oled.display();                      // Puffer anzeigen
34   delay(200);
35 }
```

Zeile 7

Display-Objekt

Auflösung 128 × 64, angeschlossen am Standard-I2C-Bus, ohne eigenen Reset-Pin.

Zeile 11-14

Starten

Startet das Display auf Adresse 0x3C. Antwortet es nicht, hält der Sketch mit einer Meldung an.

Zeile 20

Puffer leeren

Alles wird erst im Arbeitsspeicher gezeichnet ...

Zeile 22-24

Text

Größe 1 = 6 × 8 Pixel je Zeichen,
Größe 2 = doppelt so groß.

Zeile 32

Balken

Wächst jede Sekunde um ein Zehntel und beginnt nach zehn Schritten von vorn.

Zeile 33

Anzeigen

... und erst mit `display()` auf einen Schlag übertragen – das verhindert Flackern.

Geprüft: kompiliert für Arduino Uno mit Adafruit SSD1306 2.5.17 (14.3 kB Flash = 45 %, 523 Byte RAM).

05

Wenn es nicht klappt

SYMPTOM

LÖSUNG

„Display nicht gefunden!“

SDA an A4 und SCL an A5? Nicht vertauscht? Mit dem I2C-Scanner (Beispiel im ESP32-Guide, läuft auch am UNO) prüfen, ob 0x3C oder 0x3D antwortet, und die Adresse im Sketch anpassen.

Display bleibt schwarz, kein Fehler

Lötstellen prüfen, VCC wirklich an 5V? Bei manchen Modulen hilft Adresse 0x3D statt 0x3C.

Sketch hängt nach dem Upload

Zu wenig RAM: Der Bildpuffer belegt 1 kB der 2 kB des UNO. Texte mit F("...") in den Flash legen.

Kein Port in der IDE

Anderes USB-C-Kabel probieren – viele Kabel laden nur. Danach CH340-Treiber installieren (Schritt 2).

06

Wie geht's weiter?

Mit ESP32 & ESPHome

VCC an **3V3**, SDA an **GPIO21**, SCL an **GPIO22**. In ESPHome:

```
i2c:
  sda: GPIO21
  scl: GPIO22
font:
  - file: "gfonts://Roboto"
    id: roboto
    size: 14
display:
  - platform: ssd1306_i2c
    model: "SSD1306 128x64"
    address: 0x3C
    lambda: |-
      it.print(0, 0, id(roboto), "MAKER00");
```

Projektideen

Wetterstation

BMP280 oder DHT22 messen, OLED zeigt

Mini-Menü

Mit Drehgeber oder Tastern bedienen

Status-Anzeige

IP-Adresse und WLAN-Stärke am ESP32

Kleines Spiel

Pong oder Snake auf 128 × 64

Passt dazu

BMP280 Luftdruck & Temperatur

Messwerte zum Anzeigen

[MKR-GY-BMP280-024](#)

DHT22 Temperatur & Feuchte

Klassiker für die

Wetteranzeige

[MKR-DHT22-009](#)

ESP32 DevKit USB-C

WROOM-32, CH340C –

3,3-V-Logik

[MKR-ESP32-005](#)

Drehgeber EC11

Werte einstellen per

Drehknopf

[9505453479775](#)

Sicherheitshinweise. Elektronikmodul für Ausbildungs-, Entwicklungs- und Experimentierzwecke. Nur mit Kleinspannung betreiben, niemals an Netzspannung. Nicht geeignet für Kinder unter 14 Jahren; Nutzung nur unter Aufsicht. Elektroaltgeräte gehören nicht in den Hausmüll – Rückgabe kostenlos bei kommunalen Sammelstellen.

Verantwortlicher Wirtschaftsakteur (GPSR): Alexander Prez · makeroo · Hinterr Bahnhof 4 · 90513 Zirndorf · support@makeroo.de
Fragen? Schreib uns mit einem Foto deiner Verkabelung. Arduino ist eine Marke der Arduino SA und dient nur der Kompatibilitätsangabe.