GROSSES GRAFIKDISPLAY · I2C

1,3" OLED SH1106

Gleiche 128 × 64 Pixel wie das 0,96-Zoll-OLED – auf rund 80 % mehr Fläche. Erstes Bild in rund 15 Minuten, ganz ohne Löten.



AUFLÖSUNG 128 × 64 Pixel	CONTROLLER SH1106
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 Uhrzeiger

01 Auf einen Blick

Das 1,3-Zoll-OLED verteilt 128 × 64 Pixel auf eine deutlich größere Fläche – aus größerer Entfernung ablesbar und ruhiger im Bild. Die Stiftleiste ist bereits vorgelötet.

Technische Daten

Display	**OLED, weiß, selbstleuchtend**
Diagonale	**1,3 Zoll**
Auflösung	**128 × 64 Pixel**
Controller	**SH1106 (nicht SSD1306)**
Schnittstelle	**I2C, Adresse 0x3C**
Versorgung	**3,3 bis 5 V DC**
Anschluss	**4 Pins, vorgelötet**

Lieferumfang

1 × **1,3" I2C-OLED-Displaymodul (weiß, 128 × 64) mit vorgelöteter 4-Pin-Stiftleiste**

! **SH1106, nicht SSD1306**
Mit einem SSD1306-Treiber ist das Bild um zwei Pixel verschoben und am Rand abgeschnitten. Der Sketch nutzt deshalb die Bibliothek Adafruit SH110X.

i **Andere Pin-Reihenfolge**
Hier steht VCC zuerst: VCC · GND · SCL · SDA – beim 0,96-Zoll-Modul ist es GND · VCC.

Pinbelegung

PIN	FUNKTION	SPANNUNG	HINWEIS
VCC	Versorgung POWER	3,3–5 V DC	Am UNO an 5V.
GND	Masse GND	0 V	Gemeinsame Masse.
SCL	I2C Takt I2C	3,3 V Logik	Am UNO an A5.
SDA	I2C Daten I2C	3,3 V Logik	Am UNO an A4.

Reihenfolge laut Aufdruck auf der Platine: VCC · GND · SCL · SDA.

Was du zusätzlich brauchst

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

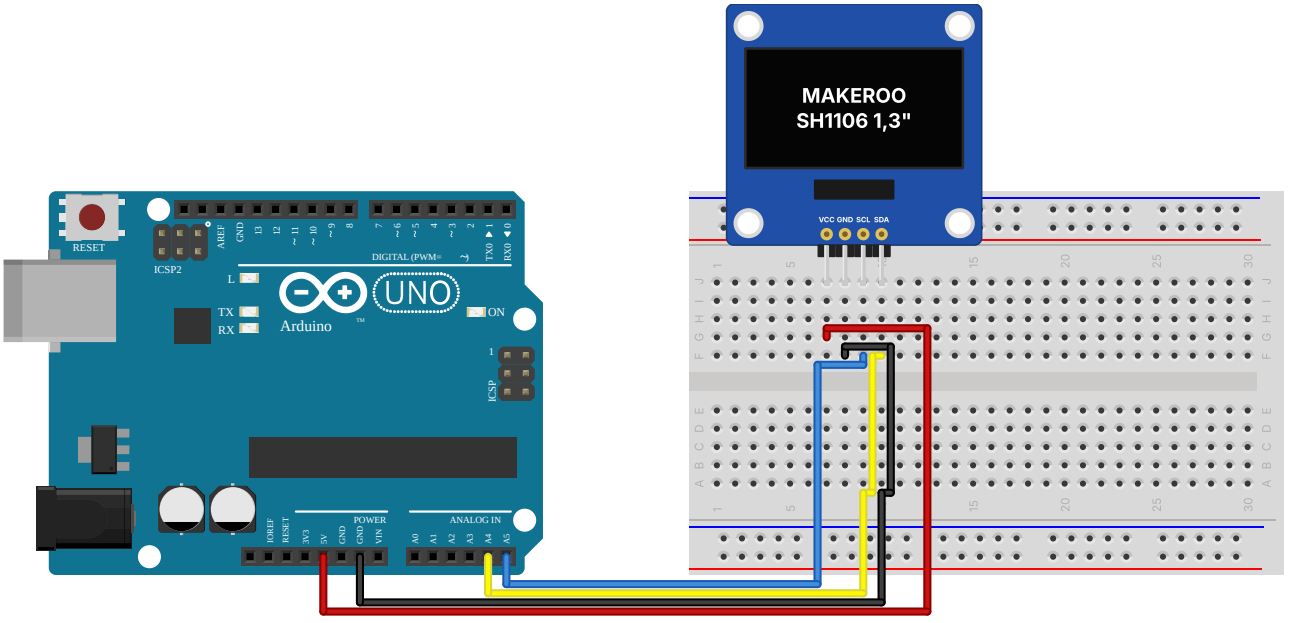
Breadboard + Jumper-Set
400 Kontakte, 40 Dupont-Kabel
MKR-SET-BB400-JUMPER

USB-C-Datenkabel
Kein reines Ladekabel – es muss Daten übertragen

02 Verkabeln

Vier Leitungen genügen. Achtung: VCC liegt bei diesem Modul ganz links.

STECKPLATINEN-ANSICHT



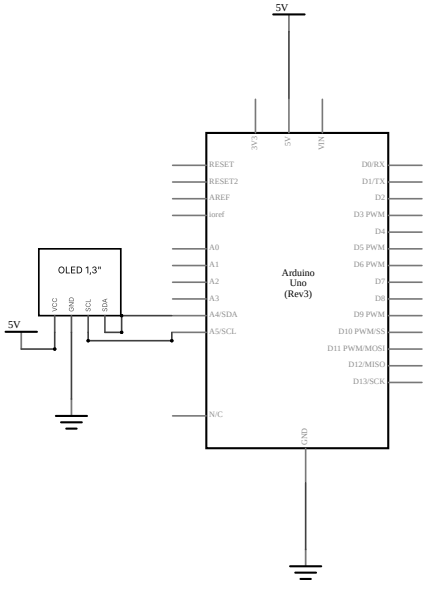
Verbindungen

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

VCC und GND nicht verwechseln
Wer vom 0,96-Zoll-OLED umsteigt, vertauscht hier leicht die beiden Versorgungspins.

Kein Löten
Die Stiftleiste ist vorgelötet – Display direkt ins Steckbrett stecken.

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 SH110X** 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!".

SO SIEHT ES AUF DEM DISPLAY AUS



i Schnelltest

Nach dem Upload erscheint „MAKEROO", darunter zählt die Laufzeit. Rechts dreht sich ein Sekundenzeiger einmal pro Minute.

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

04 Der erste Sketch

Text in zwei Größen und eine kleine Uhr mit Zeiger – Kreise und Linien aus der GFX-Bibliothek.

MAKEROO_OLED13_Start.ino

```
// MAKEROO · 1,3" OLED SH1106 (I2C) · Erster Start
#include <Wire.h>
#include <Adafruit_GFX.h>
#include <Adafruit_SH110X.h>

const byte ADRESSE = 0x3C;                    // I2C-Adresse
Adafruit_SH1106G oled(128, 64, &Wire, -1);    // SH1106, nicht SSD1306!

void setup() {
  Serial.begin(9600);
  if (!oled.begin(ADRESSE, true)) {
    Serial.println(F("Display nicht gefunden!"));
    while (true) {}
  }
  Serial.println(F("Display bereit."));
}

void loop() {
  unsigned long s = millis() / 1000;          // Sekunden seit Start
  oled.clearDisplay();
  oled.setTextColor(SH110X_WHITE);
  oled.setTextSize(2);
  oled.setCursor(0, 0);
  oled.print(F("MAKEROO"));
  oled.setTextSize(1);
  oled.setCursor(0, 26);
  oled.print(F("Laufzeit"));
  oled.setTextSize(2);
  oled.setCursor(0, 40);
  oled.print(s);
  oled.print(F(" s"));
  // kleine Uhr: ein Zeiger, der pro Sekunde 6 Grad weiterlaeuft
  float w = (s % 60) * 6 * PI / 180;
  oled.drawCircle(104, 40, 20, SH110X_WHITE);
  oled.drawLine(104, 40, 104 + 17 * sin(w), 40 - 17 * cos(w), SH110X_WHITE);
  oled.display();                             // Puffer anzeigen
  delay(200);
}
```

Zeile 7
Richtiger Treiber
SH1106G aus der SH110X-Bibliothek – der Schlüssel für ein sauberes Bild.

Zeile 11–14
Starten
Adresse 0x3C; das `true` setzt das Display beim Start zurück.

Zeile 22–24
Titel
Größe 2: 12 × 16 Pixel je Zeichen.

Zeile 33
Winkel
6 Grad pro Sekunde, umgerechnet ins Bogenmaß.

Zeile 35
Zeiger
Sinus und Kosinus berechnen die Zeigerspitze auf dem Kreis.

Zeile 36
Anzeigen
Überträgt den fertigen Puffer – ohne Flackern.

Geprüft: kompiliert für Arduino Uno mit Adafruit SH110X 2.1.15 (14.9 kB Flash = 47 %, 594 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.
Rand abgeschnitten, Pixelmüll rechts	SSD1306-Bibliothek verwendet. Für dieses Display immer den SH1106-Treiber nehmen.
Display wird warm, nichts zu sehen	Sofort trennen: VCC und GND vertauscht?
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: 16
display:
  - platform: ssd1306_i2c
    model: "SH1106 128x64"
    address: 0x3C
    lambda: |-
      it.print(0, 0, id(roboto), "MAKEROO");
```

Projektideen

Smart-Home-Anzeige
Werte aus Home Assistant am ESP32

Uhr mit Zeigern
Uhrzeit per WLAN, analog dargestellt

Messgerät
Spannung oder Strom groß anzeigen

Upgrade
Ersetzt 0,96"-Displays in fertigen Projekten

Passt dazu

BMP280 Luftdruck & Temperatur
Messwerte zum Anzeigen
`MKR-GY-BMP280-024`

ESP32 DevKit USB-C
WROOM-32, CH340C – 3,3-V-Logik
`MKR-ESP32-005`

Drehgeber EC11
Werte einstellen per Drehknopf
`9505453479775`

DHT22 Temperatur & Feuchte
Klassiker für die Wetteranzeige
`MKR-DHT22-009`

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 · Hinterm 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.