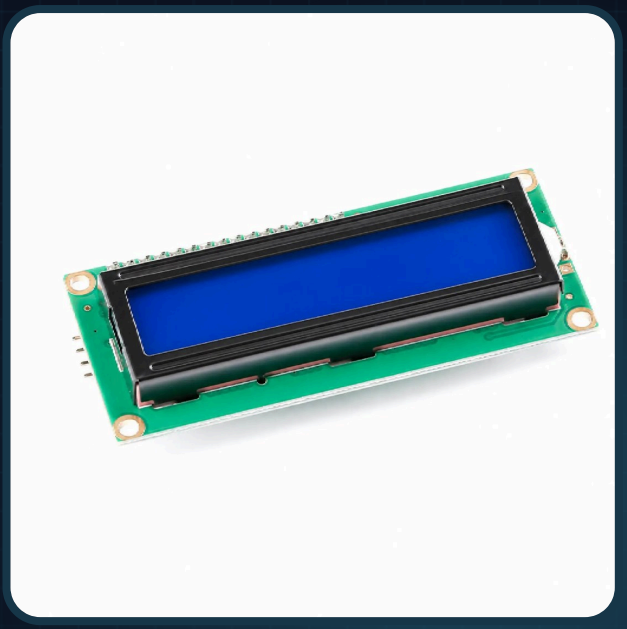
ZEICHENDISPLAY · I2C

LCD1602 16 × 2 Zeichen

Zwei Zeilen à 16 Zeichen, weiß auf blau, mit I2C-Backpack. Erster Text in rund 15 Minuten – mit UNO R3 und vier Kabeln.



ANZEIGE 16 × 2 Zeichen	CONTROLLER HD44780 + PCF8574
VERSORGUNG 5 V DC	SCHNITTSTELLE I2C 0x27

01
Verkabeln
4 Leitungen zum UNO

02
Einrichten
IDE, Treiber, Bibliothek

03
Hochladen
Beispiel-Sketch

04
Anzeigen
Zwei Zeilen Text

01 Auf einen Blick

Das LCD1602 ist das klassische Textdisplay: bei Tageslicht gut ablesbar, sparsam im Code. Der vorgelötete I2C-Backpack spart sechs Datenleitungen – vier Kabel reichen.

Technische Daten

Anzeige	**16 Zeichen × 2 Zeilen, 5 × 8 Punkte**
Farbe	**weiße Schrift auf blauem Grund**
Controller	**HD44780-kompatibel**
Backpack	**PCF8574, vorgelötet**
Schnittstelle	**I2C, Adresse 0x27 (0x20–0x27)**
Versorgung	**5 V DC**
Kontrast	**Poti auf dem Backpack**

Lieferumfang

1 × **LCD1602 (16 × 2, blau) mit vorgelötetem I2C-Backpack**

! **Braucht 5 V**
An 3,3 V bleibt die Schrift zu blass. Am ESP32 VCC an 5 V (VIN) und einen Pegelwandler in SDA/SCL setzen.

i **Nichts zu sehen?**
Fast immer ist der Kontrast falsch eingestellt – am blauen Poti hinten drehen.

Pinbelegung

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

Die vier Pins sitzen am Backpack auf der Rückseite, beschriftet GND · VCC · SDA · SCL.

Was du zusätzlich brauchst

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

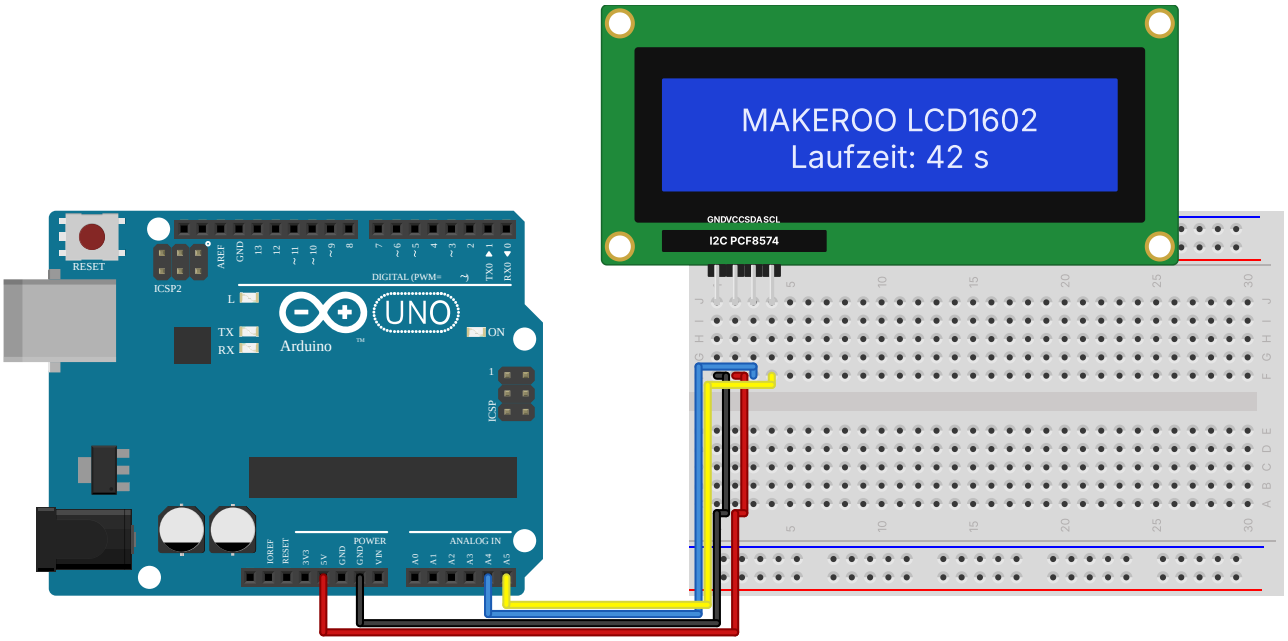
4 Jumper-Kabel female–male
Backpack-Stifte zum UNO
`MKR-JUMPER-MM-10CM`

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

02 Verkabeln

Vier Leitungen vom Backpack zum UNO. Achtung: Hier kommt SDA vor SCL.

STECKPLATINEN-ANSICHT



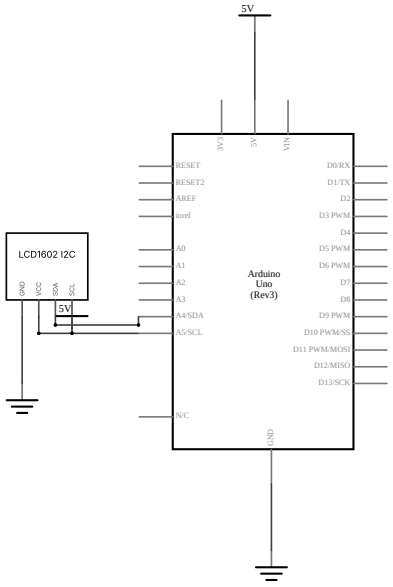
Verbindungen

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

Female-Kabel nutzen
Die Stifte am Backpack zeigen zur Seite – am einfachsten sind Jumper-Kabel mit Buchse auf der Display-Seite.

SDA und SCL
Die Reihenfolge am Backpack ist SDA vor SCL – andersherum als bei vielen OLEDs.

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 **LiquidCrystal I2C** von Frank de Brabander suchen und installieren.

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 „LCD bereit.".

SO SIEHT ES AUF DEM DISPLAY AUS



Nachbildung der Anzeige – Zeichen im 5 × 8-Raster, wie sie das LCD darstellt.

Schnelltest

Nach dem Upload leuchtet die Beleuchtung, oben steht „MAKEROO LCD1602", unten zählt die Laufzeit. Nur Kästchen? Dann am Poti drehen.

04 Der erste Sketch

Zwei Zeilen Text mit Positionsangabe – mehr braucht ein Zeichendisplay nicht.

MAKEROO_LCD1602_Start.ino

```
// MAKEROO · LCD1602 mit I2C-Backpack · Erster Start
#include <Wire.h>
#include <LiquidCrystal_I2C.h>

LiquidCrystal_I2C lcd(0x27, 16, 2);          // Adresse, Spalten, Zeilen

void setup() {
  Serial.begin(9600);
  lcd.init();                                // Display starten
  lcd.backlight();                           // Hintergrundlicht an
  lcd.setCursor(0, 0);                       // Spalte 0, Zeile 0
  lcd.print(F("MAKEROO LCD1602"));
  Serial.println(F("LCD bereit."));
}

void loop() {
  unsigned long s = millis() / 1000;         // Sekunden seit Start
  lcd.setCursor(0, 1);                       // Spalte 0, Zeile 1
  lcd.print(F("Laufzeit: "));
  lcd.print(s);
  lcd.print(F(" s  "));                      // Leerzeichen loeschen Reste
  delay(500);
}
```

Zeile 5
Display-Objekt
Adresse 0x27, 16 Spalten, 2 Zeilen.

Zeile 9–10
Starten
Initialisiert das LCD und schaltet die Hintergrundbeleuchtung ein.

Zeile 11–12
Erste Zeile
Position in Spalte, Zeile – gezählt wird ab 0.

Zeile 18
Zweite Zeile
Hier wird in jedem Durchlauf überschrieben.

Zeile 21
Reste löschen
Leerzeichen überschreiben Ziffern, die beim Kürzerwerden stehen bleiben würden.

Geprüft: kompiliert für Arduino Uno mit LiquidCrystal I2C 1.1.x (4.4 kB Flash = 14 %, 437 Byte RAM).

05 Wenn es nicht klappt

SYMPTOM	LÖSUNG
Nur helle Kästchen in Zeile 1	Kontrast am blauen Poti einstellen – das ist die häufigste Ursache.
Beleuchtung an, aber kein Text	Adresse prüfen: mit I2C-Scanner suchen, manche Backpacks nutzen 0x3F statt 0x27.
Gar nichts leuchtet	VCC an 5V? Steckt der Jumper „LED“ am Backpack?
Text blass am ESP32	VCC muss 5 V bekommen (VIN). 3,3 V reichen für dieses LCD nicht.
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 **VIN (5 V)**, SDA an **GPIO21**, SCL an **GPIO22** – mit Pegelwandler. In ESPHome:

```
i2c:
  sda: GPIO21
  scl: GPIO22
display:
  - platform: lcd_pcf8574
    dimensions: 16x2
    address: 0x27
    lambda: |-
      it.print("MAKEROO LCD1602");
```

Projektideen

Raumklima
Temperatur und Feuchte in zwei Zeilen

Zähler
Besucher oder Stückzahlen zählen

Menü mit Tasten
Einfache Einstellungen ohne Grafik

Uhr
Mit DS3231-Echtzeituhr

Passt dazu

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

DS3231 Echtzeituhr
Uhrzeit auch ohne Strom
`MKR-GY-DS3231-029`

Keypad 4×4
Zahlen und PIN eingeben
`MKR-KEYPAD-4X4`

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

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.