

Ein LCD-Display am Arduino anschließen



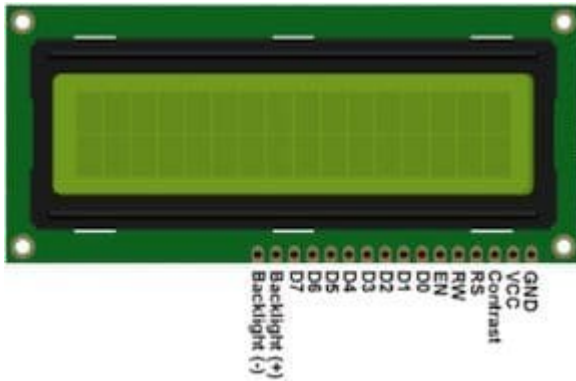
Ein [LCD-Display](#)* an den Arduino anzuschließen ist nicht schwer – auch wenn ein ganzer Haufen Kabel dafür nötig ist. Erfahre in diesem Beitrag, wo diese Kabel hingehören, wie du das Display ansteuerst und wie du Zeichen darauf erscheinen lässt.

Arten von LCD-Displays

Es gibt verschiedene Arten von LCD-Displays, die mit Arduino-Boards verwendet werden können:

1. **Zeichenbasierte LCD-Displays:** Diese Displays zeigen Text und einfache Symbole an und sind in verschiedenen Größen erhältlich, meist mit **16×2 oder 20×4 Zeichen**. Sie verwenden in der Regel den HD44780-Controller oder kompatible Chips.
2. **Grafik-LCD-Displays:** Im Gegensatz zu zeichenbasierten Displays können Grafik-LCDs komplexe Grafiken, Text und Symbole anzeigen. Sie sind in verschiedenen Auflösungen erhältlich, z.B. **128×64, 240×128 oder 320×240 Pixel**.
3. **TFT-LCD-Displays:** Diese Farbdisplays sind eine Weiterentwicklung der grafischen LCD-Displays und bieten eine höhere Auflösung und Farbtiefe. Sie sind in verschiedenen Größen und Auflösungen erhältlich, z.B. 1,8 Zoll mit 160×128 Pixeln oder 3,5 Zoll mit 480×320 Pixeln.

In diesem Tutorial erfährst du, wie du ein zeichenbasiertes Display anschließt. Schauen wir uns zunächst die Anschluss-Pins des Displays an (den vollständigen Plan für den Anschluss findest du ein wenig weiter unten):



Am Display findest du insgesamt 16 Pins. Jeweils ganz außen (GND, Backlight -) legst du die Masse an und eins daneben (VCC, Backlight +) 5 Volt über einen 220Ω Vor-Widerstand. Beides am besten direkt von deinem Arduino Board. Am dritten Pin von links schließt du ein Potentiometer an, mit dem du den Kontrast des Bildschirms einstellen kannst.

An den Pin RS sendest du von deinem Arduino Board entweder Daten oder Befehle. Wenn vom Datenpin des Arduinos kein Strom kommt, kannst du Befehle übertragen wie z.B. „Setze den Cursor an die 1. Stelle der 2. Reihe“ oder „Lösche das Display“. Wenn dein Sketch Strom anlegt, sendest du Daten, sprich die Zeichen, die du ausgeben möchtest.

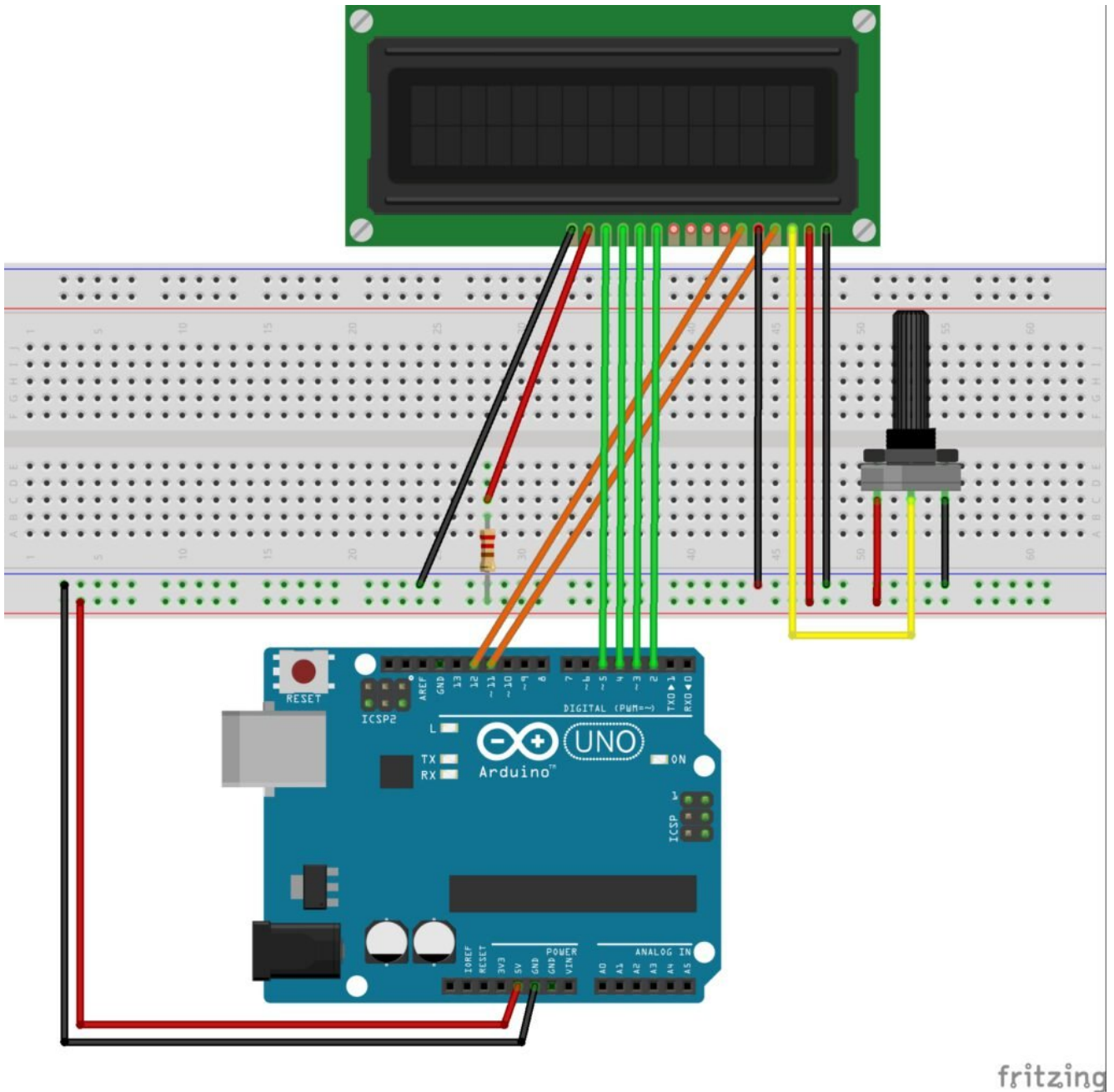
Der Pin RW steht für „read – write“ – ziemlich klar: Wir wollen Daten und Befehle „schreiben“, also senden. Um den Schreibmodus zu aktivieren, schließe diesen Pin an die Masse an.

Die Daten-Pins des LCD-Displays

Jetzt kommen insgesamt 9 Pins, von denen du 5 an deinen Arduino anschließen musst. Neben RW findest du EN wie „Enable“ – dieser Pin ermöglicht das Schreiben auf die 8 Daten-Pins daneben: Wenn hier Strom anliegt, werden Daten angenommen. Apropos 8: Du kannst deinen Bildschirm entweder mit 8 Bit oder 4 Bit betreiben. Ein großes A sieht in 8 Bit z.B. so aus: **01000001**. Diese Bits landen in den Pins D0 bis D7.

Der Betrieb mit 4 Bit hat den Vorteil, dass du nur die Hälfte der Pins ansteuern musst, was weniger Kabelei bedeutet. Das bedeutet dann allerdings auch, dass man immer 2 Zugriffe benötigt, um etwas zu übertragen, denn $2 \times 4 = 8$. Das muss allerdings nicht deine Sorge sein. Tiefer musst du an dieser Stelle jetzt erst mal auch nicht einsteigen. Alle Details zu diesem Thema [findest du bei Wikipedia](#).

So sieht ein angeschlossenes LCD-Display schematisch aus:



Das LCD-Display ansteuern

Jetzt, wo du dein Display angeschlossen hast, prüfe es am besten gleich mal, indem du Strom anlegst. Es erscheinen natürlich noch keine Zeichen darauf, aber es sollte aufleuchten. Mit dem Potentiometer kannst du auch schon den Kontrast einstellen.

Funktioniert? Dann wird es Zeit für einen kleinen Sketch und dem guten alten „**hello, world!**“ auf dem Display. Außerdem lässt du in der zweiten Zeile des Bildschirms noch einen Counter laufen.

Der passende Sketch

Mit dem folgenden einfachen Sketch bringst du deine Message aufs Display und startest den Counter. Die Erklärungen zum Code findest du in den Kommentaren im Sketch.

```
/*
  LCD-Display Test "hello, world!" + counter
  polluxlabs.net
*/

// Die Bibliothek LiquidCrystal einbinden:
#include <LiquidCrystal.h>

// Die Bibliothek initialisieren und festlegen, welcher
// Pin des Displays mit welchem Pin auf dem Board verbunden
ist:
const int rs = 11, en = 12, d4 = 2, d5 = 3, d6 = 4, d7 = 5;
LiquidCrystal lcd(rs, en, d4, d5, d6, d7);

void setup() {
  // Festlegen, wie viele Zeichen und Reihen das Display hat:
  lcd.begin(16, 2);
  // "hello, world!" an das Display senden:
  lcd.print("hello, world!");
}

void loop() {
  // Den Cursor in Spalte 0, Reihe 1 setzen:
  // (Wichtig: Reihe 1 ist die 2. Reihe, da ab 0 gezählt
wird):
  lcd.setCursor(0, 1);
  // Den Counter jede Sekunde eins hoch zählen lassen:
  lcd.print(millis() / 1000);
}
```

Und das war's auch schon. Du hast deinen Arduino mit einem LCD-Display verbunden und den Klassiker aller Messages mit einem Counter auf den Bildschirm gezaubert. Gratulation!

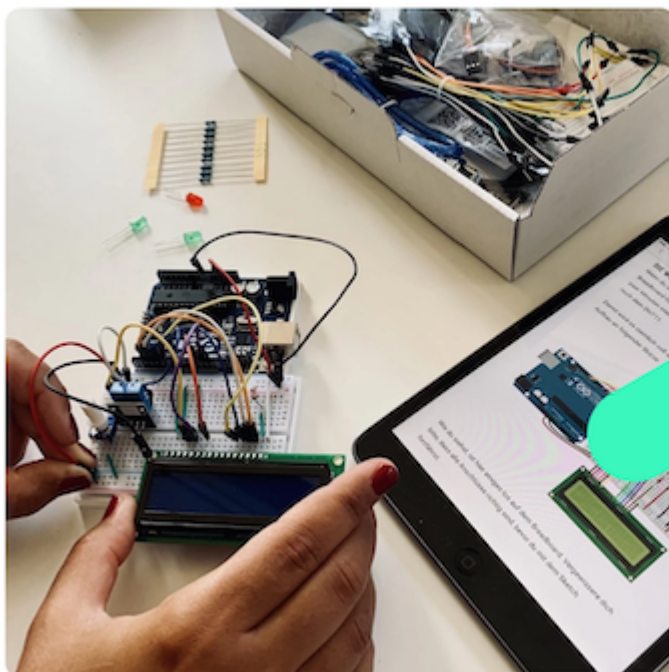
Wie geht es weiter?

Weitere Projekte mit deinem LCD-Display und anderen Sensoren, Motoren und auch Displays findest du im Arduino Online-Kurs für Einsteiger:

Schritt für Schritt zum Maker

Im Arduino Online-Kurs **lernst du C++ und baust spannende Projekte** mit dem Arduino und ESP8266. Alles ist **einsteigerfreundlich erklärt**. Programm-Beispiele kannst du herunterladen und anpassen – ohne Abtippen aus Büchern.

[Mehr erfahren](#)



Hier bei Pollux Labs gibt es noch weitere passende Tutorials: Erfahre, wie du [eigene Zeichen gestalten und auf dem Bildschirm anzeigen](#) kannst. Vielleicht möchtest du auch lieber ein [TFT-Display am Arduino anschließen](#) oder [eigene Bilder auf einem OLED-Display anzeigen](#)?

In unseren [Arduino-Projekten](#) findest du noch viele weitere Ideen.

Das LCD-Display per I²C anschließen

Du sparst dir eine ganze Menge Kabel, wenn du ein Display mit I²C verwendest – Hier benötigst du nur vier Anschlüsse. Auch musst du dann kein extra Potentiometer auf deinem Breadboard verbauen.

I²C (Inter-Integrated Circuit), ausgesprochen als „I-squared-C“ oder „I-two-C“, ist ein serieller Kommunikationsbus, der in den 1980er Jahren von Philips (jetzt NXP Semiconductors) entwickelt wurde. **I²C hat sich seitdem zu einem weit verbreiteten Protokoll entwickelt, das in vielen elektronischen Geräten und Systemen zum Einsatz kommt, um Kommunikation zwischen verschiedenen integrierten Schaltkreisen (ICs) zu ermöglichen.** In diesem Artikel werden wir die Grundlagen von I²C, seine Funktionsweise und einige Anwendungsfälle betrachten.

Funktionsweise von I²C

I²C verwendet nur zwei bidirektionale Leitungen für die Kommunikation: Serial Data Line (SDA) und Serial Clock Line (SCL). SDA ist für die Übertragung von Daten verantwortlich, während SCL die Taktimpulse bereitstellt, die den Datenaustausch synchronisieren. Durch diese einfache Konfiguration können mehrere Geräte über den gleichen Bus

miteinander kommunizieren.

In einem I²C-System gibt es zwei Hauptrollen: Master und Slave. Der Master initiiert die Kommunikation, indem er die Adresse des Slaves sendet, mit dem er kommunizieren möchte, gefolgt von einer Lese- oder Schreiboperation. Ein Slave-Gerät antwortet, indem es Daten an den Master sendet oder von ihm empfängt.

Vorteile von I²C

1. Einfachheit: I²C benötigt nur zwei Leitungen für die Kommunikation, was zu einer einfacheren Verdrahtung und geringerem Platzbedarf auf einer Leiterplatte führt.
2. Mehrere Geräte: I²C ermöglicht die Kommunikation zwischen mehreren Geräten über den gleichen Bus, wodurch die Komplexität des Systems reduziert wird.
3. Flexibilität: I²C unterstützt unterschiedliche Geschwindigkeiten, z. B. Standardmodus (100 kbps), Fast-Modus (400 kbps), High-Speed-Modus (3,4 Mbps) und Ultra-Fast-Modus (5 Mbps).

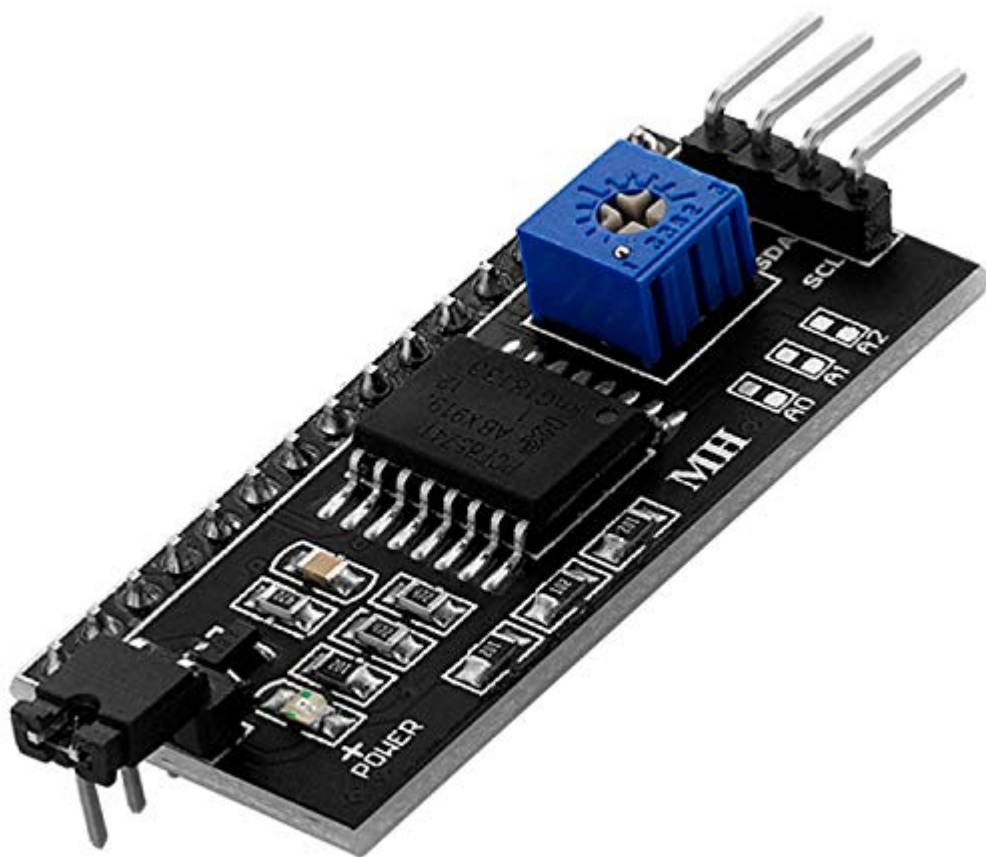
Angebot



[AZDelivery HD44780 16x2 LCD Modul Display Bundle mit I2C Schnittstelle 2x16 Zeichen kompatibel mit Arduino inklusive E-Book! \(mit Grünem Hintergrund und Schwarzen Zeichen\)](#)

6,99 €

Mit einem separaten I²C Adapter kannst du dein „normales“ LCD-Display auch nachträglich aufrüsten. Der Adapter wird auf der Rückseite des Displays montiert bzw. verlötet. Indem du mehrere Adressen verwendest, kannst du auch mehrere LCD-Displays an einem Arduino anschließen – ohne I²C wäre das aufgrund der Anzahl der Kabel unmöglich.



[AZDelivery I2C Serial Adapter Board Modul Schnittstelle für LCD Display 1602 und 2004 kompatibel mit Arduino inklusive E-Book!](#)

4,99 €