**CD-Display am Arduino anschließen**

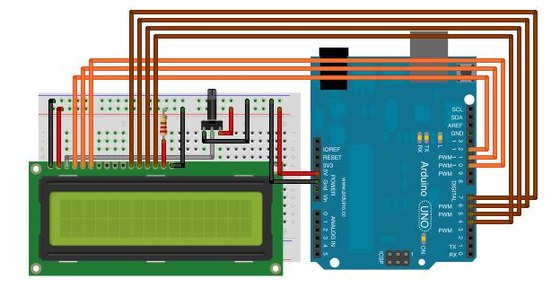
[](https://i2.wp.com/fluuux.de/wp-content/uploads/2012/09/Bildschirmfoto-2012-09-02-um-15.29.00.png)

In diesem kleinen Beitrag zeige ich wie man ein LCD-Display an einem Arduino anschließt. Zusätzlich habe ich zwei kleine Videos aufgenommen die man auf Youtube finden kann.

Ich benutze ein Display mit zwei Zeilen und 16 Zeichen je Spalte.

**Anschluss des Display am Arduino**

1. Vss (GND) -> GND
2. Vdd (+5V) -> +5V
3. V0 (Kontrastspannung (0V bis 5V)) -> Schleifer Mitte (10k Drehpotentiometer)
4. RS (Register Select (Befehle/Daten)) -> Digital 12
5. R/W (Read/Write) -> Digital 11 oder an GND
6. E (Enable) -> Digital 10
7. DB0 (Datenbit 0)
8. DB1 (Datenbit 1)
9. DB2 (Datenbit 2)
10. DB3 (Datenbit 3)
11. DB4 (Datenbit 4) -> Digital 7
12. DB5 (Datenbit 5) -> Digital 6
13. DB6 (Datenbit 6) -> Digital 5
14. DB7 (Datenbit 7) -> Digital 4
15. A (Anode Hintergrundbeleuchtung) -> Mit einem Vorwiderstand an +5V oder an einen zweiten 10k – Drehpotentiometer um so die Displayhelligkeit einstellen zu können
16. K (Kathode Hintergrundbeleuchtung) -> GND
17. Drehpotentiometer rechts -> +5V
18. Drehpotentiometer links -> Ground

[](https://i2.wp.com/fluuux.de/wp-content/uploads/2012/09/16x2-LCD-Display-Am-Arduino.png)

**Arduino Sketch schreiben**

|  |
| --- |
| #include <LiquidCrystal.h>  LiquidCrystal lcd(12, 11, 10, 7, 6, 5, 4); //Hier die Ports angeben an denen das Display angeschlossen wurde    void setup()  {   lcd.begin(16, 2); //Art des Displays   lcd.print("Hallo Welt");  }    void loop()  {   lcd.setCursor(0, 1);   lcd.print(millis()/1000);  } |

Aufgabe: Ein LCD Display soll mit einem Arduino Mikrocontroller angesteuert werden. Danach soll auf dem Display ein vorher festgelegter Text wie auf folgendem Beispielfoto erscheinen.

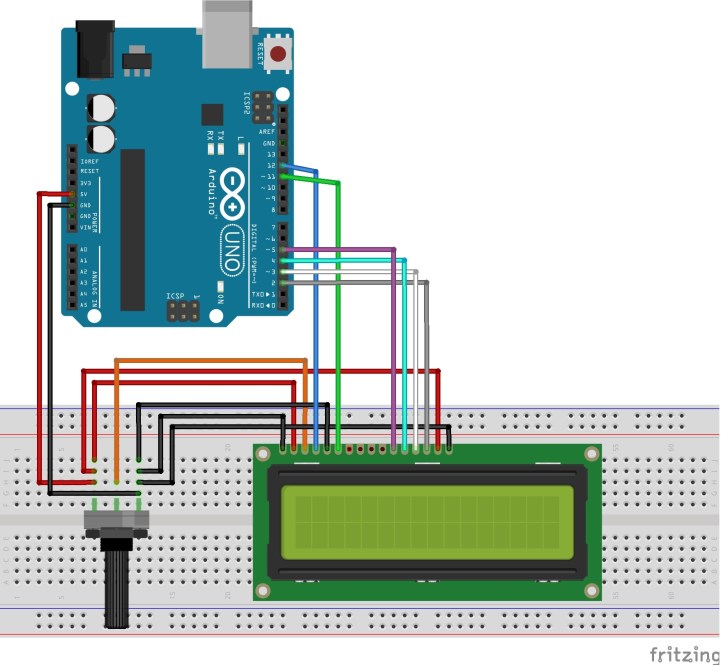


Material: Arduino Mikrocontrollerboard (In diesem Beispiel UNO R3), ein Drehregler (bzw. Potentiometer), 14 Jumperkabel, Breadboard  ([Materialbeschaffung: www.funduinoshop.com](http://www.funduinoshop.com))

Anhand des Materials und der Skizze erkennt man schnell, dass die Verkabelung nicht so leicht ist. Das liegt daran, dass das LCD Display mit sehr vielen Kabeln verbunden werden muss. Eine weitere Schwierigkeit sind fehlende Buchsen mit denen man die Kabel anschließen könnte. Daher bietet es sich an, eine Steckbuchsenleiste aufzulöten oder die Kabel direkt an das LCD anzulöten. Im zweiten Fall bietet es sich an, ein Flachbandkabel zu verwenden (Bspw. von alten IDE-Laufwerken wie Festplatten, CD- oder DVD Laufwerken). Ohne eine gelötete Verbindung ist es nahezu ausgeschlossen, gute Ergebnisse zu erzielen.

Der Drehregler ist dazu da, den Kontrast des LCD einzustellen. Die LED Hintergrundbeleuchtung wird wie in der Skizze dargestellt mit 5V versorgt. Da man in der Skizze die Beschriftung am LCD nicht erkennen würde, wird sie nicht dargestellt. Man muss also die Position der Kabel am LCD Abzählen (Beispiel: Am LCD wird das erste Kabel von rechts mit GND verbunden. Das zweite Kabel von rechts wird mit 5V verbunden usw…). Info: Für schnelle Basteleien benutzen viele Bastler lieber ein LCD-Keypad-Shield oder ein I2C-LCD, da man sich bei den beiden Alternativen nicht mehr um die Verkabelung des LCD kümmern muss. Allerdings sind die beiden genannten Alternativen auch teurer.

Skizze:



Wenn man die Verkabelung erfolgreich hergestellt hat, kann man sich mit der Software befassen. Wie bei vielen anderen Bauteilen, wird auch hier auf eine „Library“ zurückgegriffen. Die Library für das LCD Display ist Bestandteil der Arduino-Software und muss nicht zusätzlich installiert werden.

#include <**LiquidCrystal**.h> //LCD-Bibliothek laden

**LiquidCrystal** lcd(12, 11, 5, 4, 3, 2); //In dieser Zeile wird festgelegt, welche Pins des Mikrocontrollerboards für das LCD verwendet wird (Am besten erstmal nicht verändern).

void setup()

{

lcd.begin(16, 2); //Im Setup wird angegeben, wie viele Zeichen und Zeilen verwendet werden. Hier: 16 Zeichen in 2 Zeilen.

}

void loop()

{

lcd.setCursor(0, 0); //Startposition der Darstellung auf dem LCD festlegen. lcd.setCursor(0,0) bedeutet: Erstes Zeichen in der ersten Zeile.

lcd.print("www.funduino.de"); //Dort soll der Text „www.funduino.de“ erscheinen. Der Befehl lcd.setCursor ist dem Mikrocontrollerboard durch das Aufrufen der Bibliothek bekannt.

lcd.setCursor(0, 1); // lcd.setCursor(0,1) bedeutet: Erstes Zeichen in der zweiten Zeile.

lcd.print("Viel Erfolg"); //Dort soll dann der Text „Viel Erfolg!!!“ auftauchen.

}

Eine Variation: Im Display soll abwechselnd erst oben und dann unten ein Text erscheinen. In diesem Beispiel der Text „Oben“ und „Unten“.

#include <**LiquidCrystal**.h>

**LiquidCrystal** lcd(12, 11, 5, 4, 3, 2);

void setup()

{

lcd.begin(16, 2);

}

void loop()

{

lcd.setCursor(0, 0);

lcd.print("Oben"); //Beginn beim ersten Zeichen in der ersten Zeile mit dem Text „Oben“.

delay (2000); //Zwei Sekunden warten.

lcd.clear(); //Display löschen.

lcd.setCursor(5, 1);

lcd.print("unten"); //Erneuter Beginn beim fünften Zeichen in der zweiten Zeile mit dem Text „Unten“.

delay (2000); //Zwei Sekunden warten.

lcd.clear(); //Display löschen.

}

Ein LCcD eignet sich besonders gut, um Sensorwerte oder andere Ausgaben des Mikrocontrollerboards anzuzeigen. Weitere Hilfe bekommt man z.B. in der Arduino-Software. Unter den Beispiel-Sketches findet man eine Reihe von verschiedenen Beispielen unter dem Menüpunkt „LiquidCrystal“.