

Arbeitsblatt: Liquid Crystal Display (LCD)

Beschreibung

Die Parksensor-Schaltung aus dem Projekt „Parksensor“ soll nun um eine Anzeige der gemessenen Entfernung ergänzt und der Sketch entsprechend angepasst werden. Hierzu wird ein zweizeiliges Display mit 16 Zeichen pro Zeile verwendet. Ein solches Display hat 16 Anschlüsse, von denen aber „nur“ 11 verwendet werden müssen. (Ein Anschluss über I²C ist auch möglich, wird aber hier nicht verwendet.)



LC-Display (LCD = Liquid Crystal Display, Flüssigkristallanzeige)

Ein LC-Display nutzt flüssige Kristalle und polarisiertes Licht, um ein Bildelement (= Pixel) hell oder dunkel zu schalten. Dies wird durch Anlegen einer Spannung am Pixel erreicht. Die Technik wird seit Beginn der 1970er-Jahre verwendet. Das Bild rechts zeigt eine Armbanduhr mit LC-Display von 1978.



Quelle: Wikimedia Commons,
Lizenz: CC-BY-SA 3.0

Über die Anschlüsse DB0 bis DB7 kann gleichzeitig ein Byte, also 8 bit, übertragen werden. Man spricht von paralleler Signalverarbeitung. Im Halbbyte-Modus werden nur vier Eingänge benutzt (DB4 bis DB7). Das reduziert die notwendige Zahl der Kabelverbindungen.

Das folgende Bild zeigt das LCD schematisch mit den 16 Anschlüssen an der oberen Kante.



Foto: R. P. Dröge

Anschlüsse von links nach rechts:

VSS	= Ground/Minuspole (Arduino GND)
VDD	= Stromversorgung 5 V (Arduino 5 V)
VO	= Kontrastregelung (über Potentiometer)
RS	= Register select pin (Arduino Pin 12)
RW	= Read/Write pin (Arduino GND)
E	= Enabled pin (Arduino Pin 11)
DB0 bis DB7	= Dateneingänge

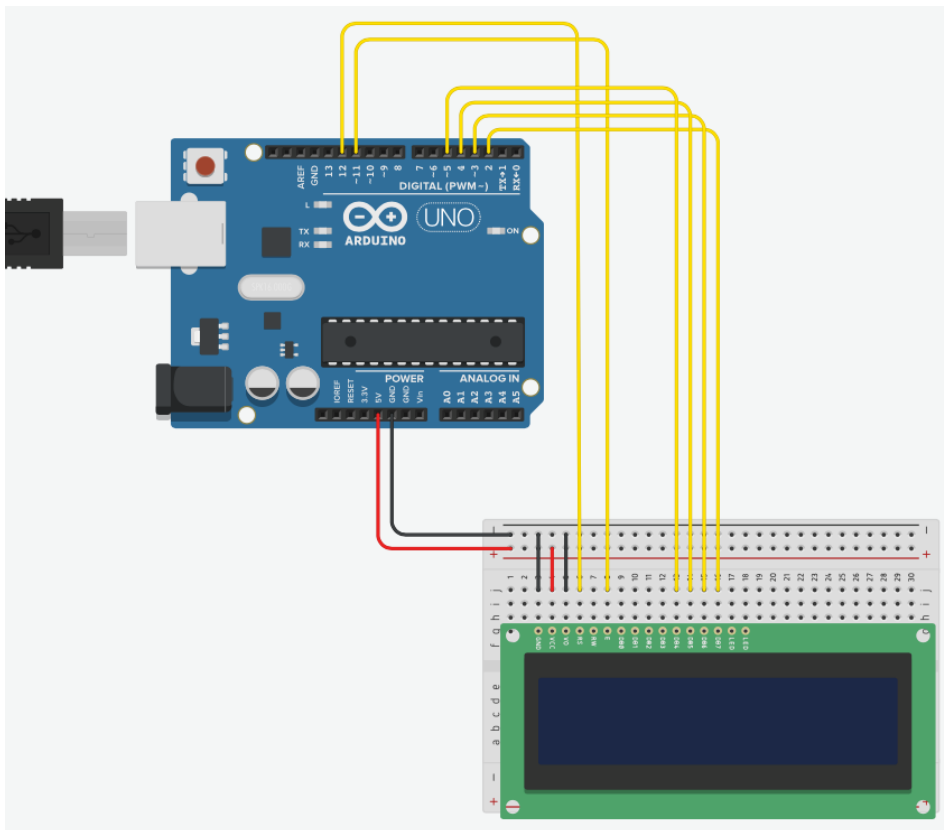
A und K = Spannungsversorgung für die LED-Hintergrundbeleuchtung (erfordert einen Vorwiderstand von 220 Ohm)



Aufgabe 1: Betrieb eines LCDs am Arduino

Als erstes soll das LCD an den Arduino angeschlossen werden, um einen Wert anzuzeigen. Es werden benötigt: Arduino Uno, ein Breadboard, ein LCD und diverse Kabel.

Öffne die entsprechende Vorlage in TinkerCAD.



Quelle: BSZ Bietigheim

Zum Verwenden eines LCD am Arduino müssen zusätzliche Befehle im Programmcode verfügbar gemacht werden. Dies geschieht mithilfe einer **Befehlsbibliothek (= Library)**, die für den jeweiligen Anwendungszweck im Sketch aufgerufen wird. Bibliotheken ersparen dem Programmierer viele Zeilen Programmcode und ermöglichen es gerade Programmier-Neulingen, schnell zu Ergebnissen zu kommen und viele verschiedene Bauteile (Servo-Motoren, Tastatur, Infrarot-Fernbedienung usw.) anzu steuern. Viele Bibliotheken sind in der Arduino IDE bereits vorinstalliert. Weitere lassen sich aus dem Internet laden.

Die Bibliothek für das LCD wird durch den Aufruf

```
#include <LiquidCrystal.h>
```

noch vor dem Setup-Teil des Sketches geladen. Danach muss das LCD aktiviert werden. Dies geschieht durch Zuweisung eines Namens („LCD“) und die Deklaration der mindestens sechs Variablen.



Übertrage den folgenden Programmcode in die TinkerCAD-Vorlage und teste den Sketch.

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

void setup() {
    // LCD als 16 Zeichen x 2 Zeilen festlegen
    lcd.begin(16, 2);
    // Ausgeben einer Nachricht
    lcd.print("Sek. ab Neustart");
}

void loop() {
    // Setzen des Cursors in Spalte 0, Zeile 1
    lcd.setCursor(0, 1);
    // Ausgeben der Zeit in Sekunden seit dem letzten Neustart
    lcd.print(millis()/1000);
}
```



- a. Frage: Welche Bedeutung hat der Aufruf „`lcd.begin(16, 2)`“?
(Hinweis: evtl. Internetrecherche durchführen)

- b. Frage: Welche Bedeutung hat der Aufruf „`lcd.print(„Sek. ab Neustart“)`“?

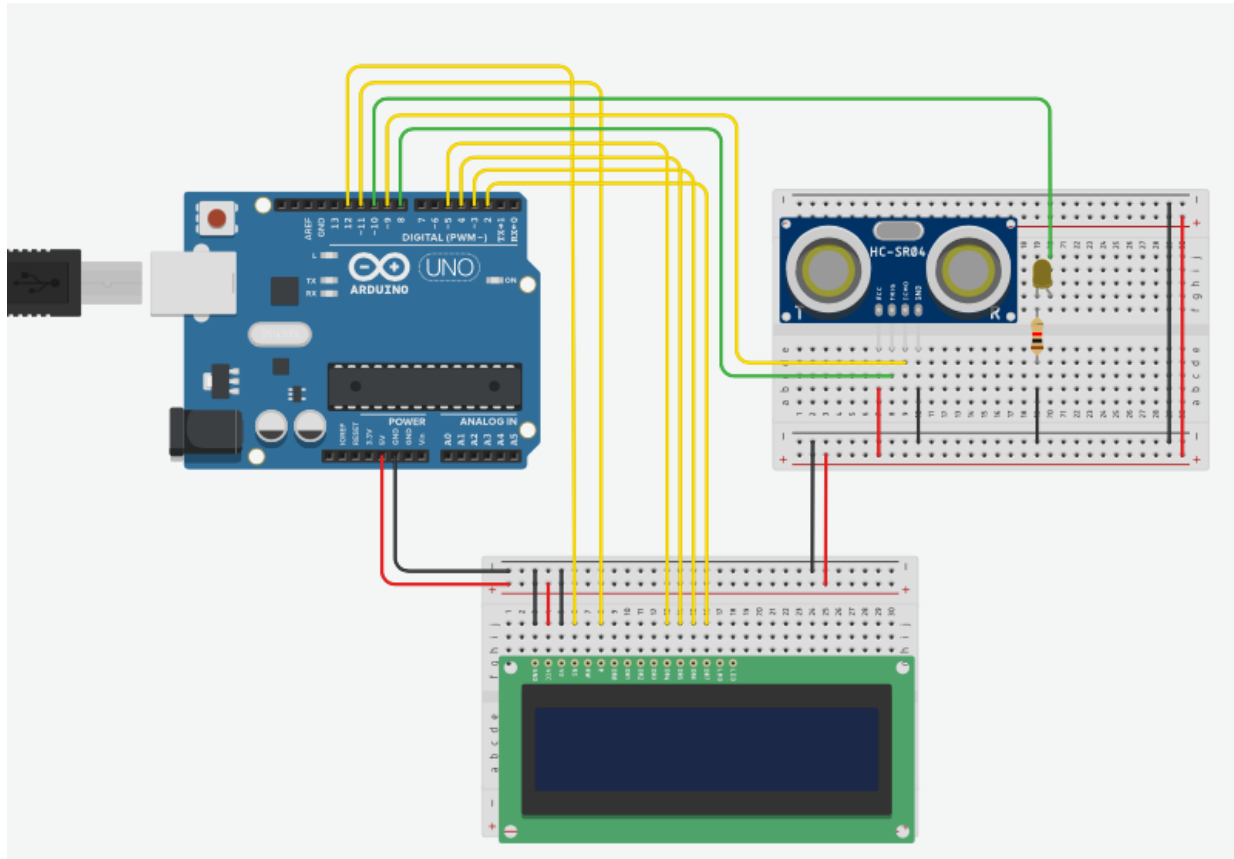
- c. Frage: Welche Bedeutung hat der Aufruf „`lcd.setCursor(0, 1)`“?

- d. Frage: Warum wird bei dem Aufruf „`lcd.print(millis()/1000)`“ kein Text ausgegeben?
Wie unterscheidet sich dieser Aufruf von Frage b.?



Aufgabe 2: Parksensor mit Distanzanzeige

Die Parksensor-Schaltung soll eine Distanzanzeige erhalten. Öffne hierfür die entsprechende Tinker-CAD-Vorlage. Das folgende Bild zeigt das fertige Anschlussbild mit Parksensor und Distanzanzeige.



Quelle: BSZ Bietigheim

Ändere das Programm so ab, dass die gemessene Distanz auf dem LCD-Bildschirm in der folgenden Form angezeigt wird:

Distanz:
XXXXX cm



Für Profis:

Die Bibliothek für das LCD enthält auch eine Funktion namens `createChar()`. Finde heraus, was diese Funktion kann und wie man diese benutzt. Schreibe einen kurzen Sketch, der die Funktion `createChar()` zeigt.

Quellen

Flüssigkristallanzeige: <https://de.wikipedia.org/wiki/Flüssigkristallanzeige>

LC-Display: <https://www.arduino.cc/en/Reference/LiquidCrystalDisplay>

Funktion LiquidDisplay: <https://www.arduino.cc/en/Reference/LiquidCrystalConstructor>

Bibliotheken: <https://www.arduinelibraries.info>