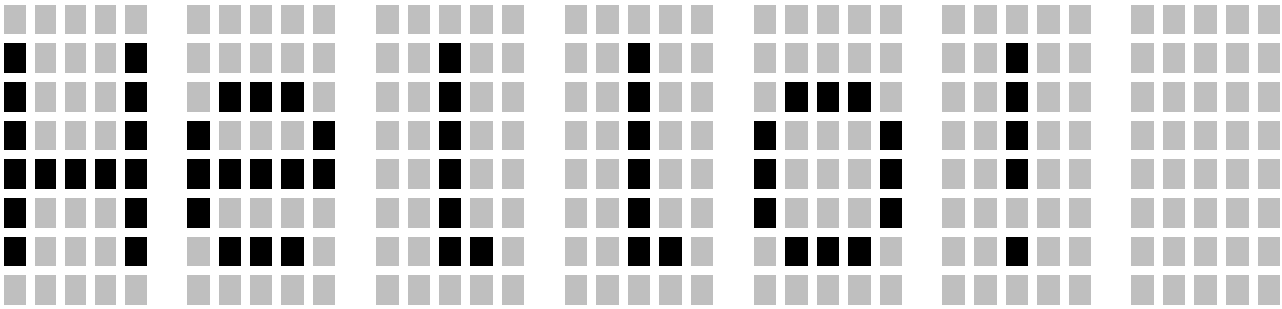


## 4.6. Die LCD-Anzeige

### 4.6.1. Aufbau und Funktion

Die LCD (Liquid Crystal Display) Anzeige vom Typ **1602** besteht aus 2 x 16 Feldern (**Matrizen**) mit jeweils 5 x 8 Punkten (**Dots**). Unten sind 7 solcher Felder gezeigt:

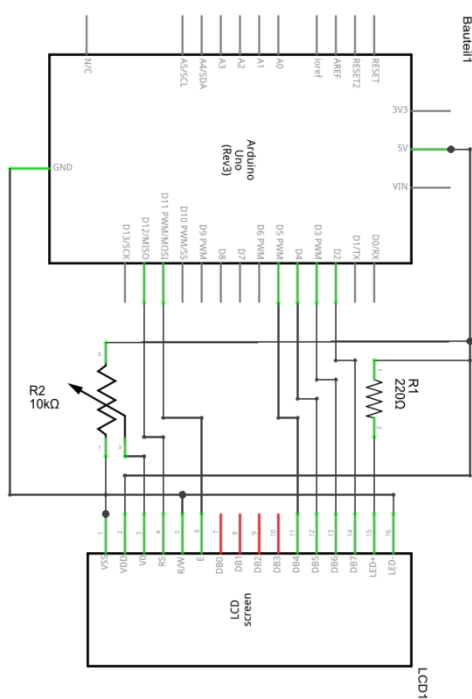


Jeder Punkt enthält eine **Suspension** bzw. Lösung von winzigen **chiralen** („händigen“) Kristallen bzw. Molekülen, die die **Polarisationsebene** des sichtbaren Lichtes ändern. Die Teilchen haben außerdem einen **Dipolcharakter**, so dass sie sich durch das elektrische Feld zweier **Elektroden** gemeinsam ausrichten lassen. Zwei **Polarisationsfilter** vor und hinter der Suspension bzw. Lösung bewirken, dass das Licht der **Hintergrundbeleuchtung** je nach angelegter Elektrodenspannung bzw. Orientierung der Teilchen blockiert oder durchgelassen wird. (Das Licht eines LCD-Displays ist daher polarisiert, so dass die Betrachtung durch eine waagrecht polarisierte Sonnenbrille eventuell schwierig ist...).

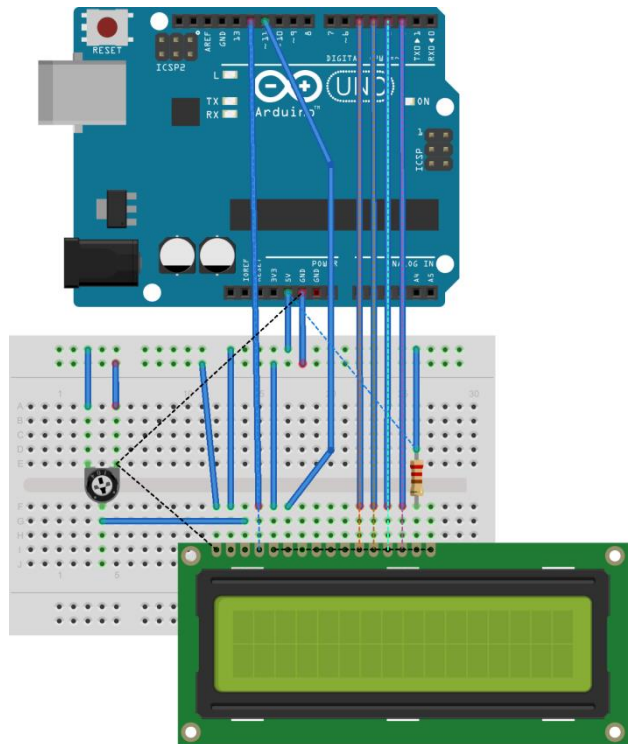
Die Elektrodenspannung und damit der Kontrast werden über ein 10 k $\Omega$ -**Potentiometer** geregelt.

Die Elektroden werden von einem eigenen kleinen Mikroprozessor gesteuert und können mittels des integrierten Treibers (Controllers) **Hitachi HDD 44780** die  $2^8 = 256$  alphanumerische Zeichen des **ASCII (American Standard Code for Information Interchange)**, <http://www.poenitz-net.de/Informatik/4.Mikrocontroller/4.6.S.ASCII.pdf> Zeichensatzes darstellen. Dem Controller werden im **4-bit-Modus** über vier die **Datenleitungen** D4 – D7 jeweils 2 x 4 Bit hintereinander gesendet, um die gewünschte Zeichen anzugeben. Im etwas schnelleren **8-bit-Modus** werden zusätzlich die Datenleitungen D0 – D3 benutzt, um **gleichzeitig** alle 8 Bit zu senden. Dafür werden aber dann auch 8 Ausgänge bzw. Verbindler benötigt, so dass wir uns für den Normalfall mit dem einfacher zu verkabelnden 4-bit-Modus zufrieden geben. Es kommen auch so schon einige Anschlüsse zusammen:

### 4.6.2. Anschluss



fritzing



fritzing

VSS: Kathode für Steuerelektroden → GND

VDD: Anode für Steuerelektroden → 5 V

V0: Kontrastregler für Steuerelektroden → Schleifer des Potentiometers

RS: Register Select: wählt zwischen Datenregister und Befehlsregister des LCD-Mikrocontrollers → D 12

RW: Read/Write: wählt zwischen Lesen und Beschreiben des LCDs. Da wir nur beschreiben wollen: → GND

E: Enable: aktiviert die Übertragung → D 11

D 0 – D 3: Datenleitungen für 8-bit Betrieb: bleiben ungenutzt

D 4 – D 7: Datenleitungen für 4-bit-Betrieb → D 5 – D 2

A: Anode für Hintergrundbeleuchtung → 5 V über Vorwiderstand 220 Ω

K: Kathode für Hintergrundbeleuchtung → GND

### 4.6.3. Sketch

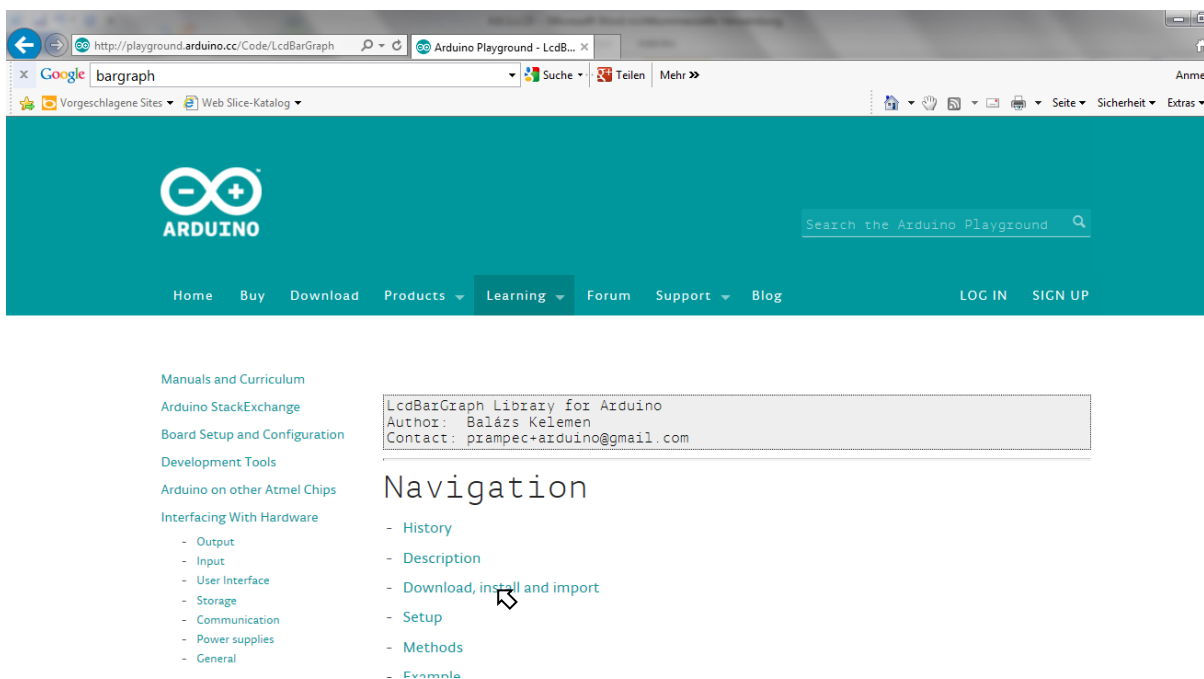
Um diesen Controller ansprechen zu können, benötigt man einen extra Satz Befehle, der zu Beginn des Sketches mit der Anweisung `#include <LiquidCrystal.h>` geladen wird. Diese Befehle sind leicht verständlich und machen den Sketch recht übersichtlich:

```
#include <LiquidCrystal.h>
int RS = 12;           // Variable für Register Select mit Pin 12 initialisieren
int E = 11;           // Variable für Enable mit Pin 11 initialisieren
int D4 = 5;           // Variable für Datenleitung 4 mit Pin 5 initialisieren
int D5 = 4;           // Variable für Datenleitung 5 mit Pin 4 initialisieren
int D6 = 3;           // Variable für Datenleitung 6 mit Pin 3 initialisieren
int D7 = 2;           // Variable für Datenleitung 7 mit Pin 2 initialisieren
LiquidCrystal lcd(RS, E, D4, D5, D6, D7); // setup für Display
void setup()
{
  lcd.begin(16, 2);    // Spaltenzahl 16 und Zeilenzahl 2 initialisieren
  lcd.print("Hello World!"); // Ausgabe 1. Zeile
  lcd.setCursor(0, 1); // Cursor auf Spalte 0 und Zeile 1 setzen
  lcd.print("Arduino"); // Ausgabe 2. Zeile
}
void loop(){}         // diesmal ohne Inhalt, da statische Anzeige
```

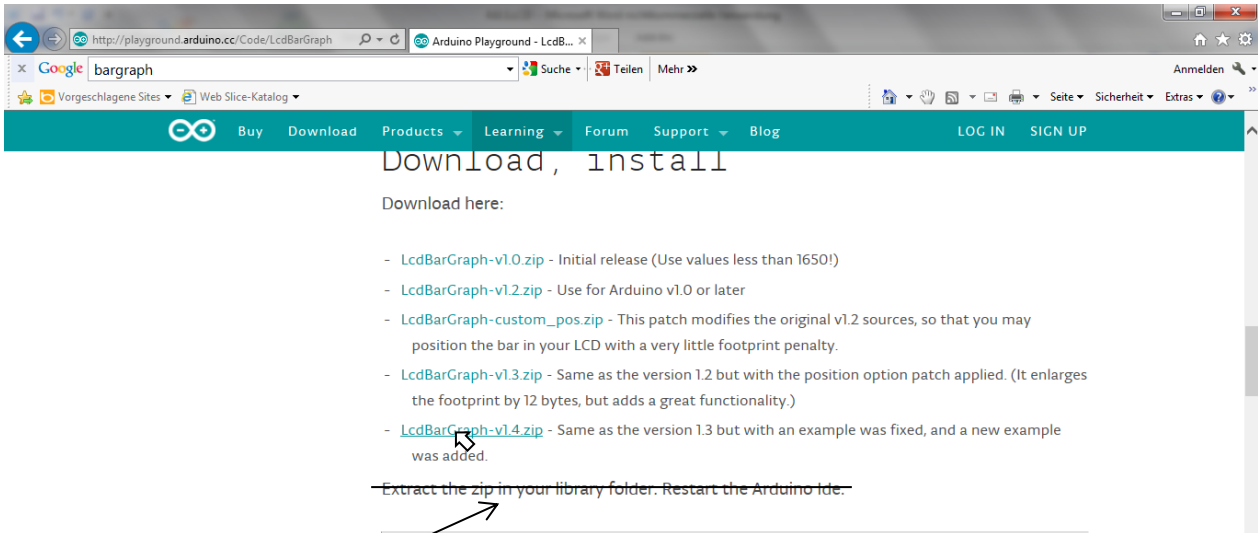
Übungen: Aufgaben zur LCD-Anzeige Nr. 1

### 4.6.4. Balkendiagramme

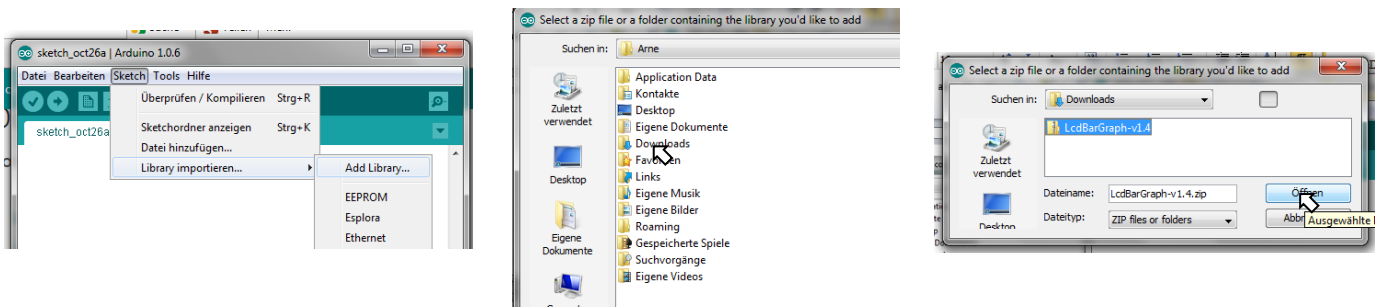
Das Display ist durch seine schon recht hohe Auflösung auch schon für einfache graphische Anzeigen geeignet. Mit Hilfe des Befehls `draw` lässt sich eine Balkenanzeige für die im letzten Abschnitt besprochenen Sensoren realisieren. Die dazu gehörige **library** `LcdBarGraph.h` muss dazu von der unten gezeigten Seite <http://playground.arduino.cc/Code/LcdBarGraph> heruntergeladen und in die IDE integriert werden:



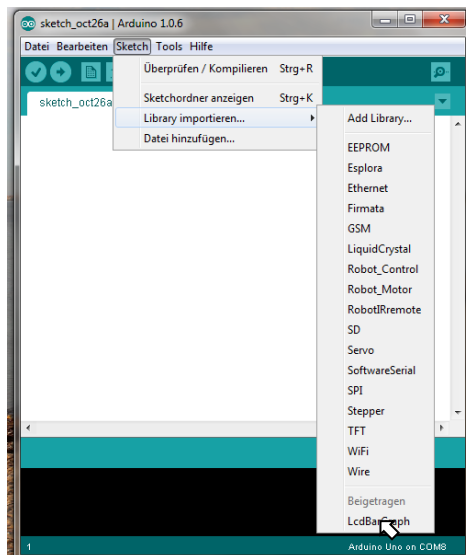
The screenshot shows a web browser displaying the Arduino Playground website. The page title is "LcdBarGraph Library for Arduino". The author is Balázs Kelemen, and the contact email is piampec+arduino@gmail.com. The page has a navigation menu with options: History, Description, Download, install and import, Setup, Methods, and Example. A mouse cursor is pointing at the "Download, install and import" option. The page also features a search bar and a navigation menu with options: Home, Buy, Download, Products, Learning, Forum, Support, Blog, LOG IN, and SIGN UP.



Man folgt **NICHT** der Anweisung unten, sondern überlässt das Entpacken und Installieren der heruntergeladenen zip-Datei der Arduino-IDE. Dazu schließt man den Browser, startet die Arduino-IDE und wählt *Sketch/library importieren.../Add Library...* Im daraufhin gezeigten Benutzerverzeichnis öffnet man den Ordner *Download*, markiert die zip-Datei *LcdBarGraph-v1.4.zip* und klickt auf *Öffnen*:



Anschließend sollte die neue Library ganz unten in der Liste der verfügbaren Libraries angezeigt werden:



#### 4.6.5. Sketch

```

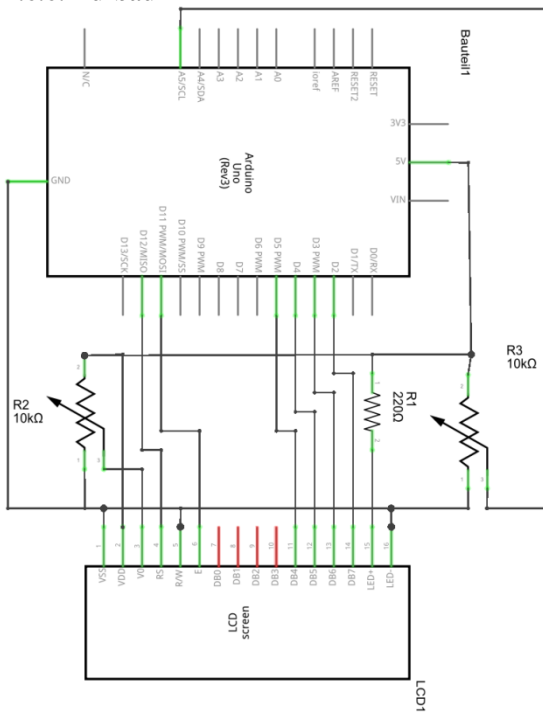
#include <LiquidCrystal.h> // Befehlssatz für LCD laden
#include <LcdBarGraph.h> // Befehlssatz für Balken laden
int sensorPin = 0; // Eingangspin auf A0
LiquidCrystal lcd(12,11,5,4,3,2); // Anschlüsse des LCDs initialisieren
LcdBarGraph lbg(&lcd, 15, 0, 1); // Balkendiagramm initialisieren
// Werte für LCD übernehmen
// Länge: 16 Felder
// Start in Spalte 0 und Zeile 1

void setup() {
  lcd.begin(2, 16); // Spaltenzahl und Zeilenzahl initialisieren
  lcd.print(" 1V 2V 3V 4V 5V"); // Ausgabe 1. Zeile
  lcd.setCursor(0, 1); // Zeilensprung
  delay(100); // Reaktionszeit für Orientierung der Kristalle
}

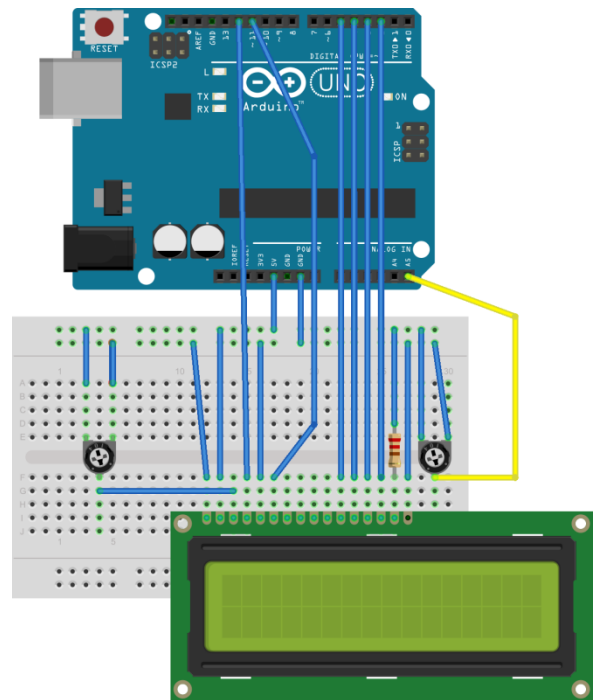
void loop()
{
  lbg.drawValue( analogRead(sensorPin), 1024); // Balken zeichnen
  delay(100); // Reaktionszeit für Orientierung der Kristalle
}

```

#### 4.6.6. Aufbau



fritzing



fritzing

Übungen: Aufgaben zur LCD-Anzeige Nr. 2 und 3