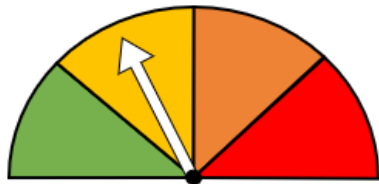




Arduino Uno

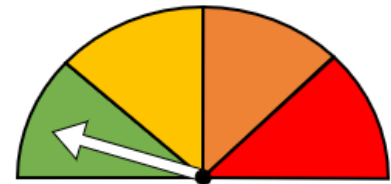
Station 4 - ZUSATZ | 1+1=10



algorithmisches Denken



Programmieraufwand



Komplexität der Schaltung



ZUSATZAUFGABE

Sehr gut – für eine vierstellige Dualzahl funktioniert dein Projekt bereits. Für ein alltagstaugliches Projekt ist es natürlich nicht so toll andauernd den seriellen Monitor nutzen zu müssen, verändere deinen Aufbau daher nun so, dass die Ausgabe über ein LCD-Bildschirm funktioniert.

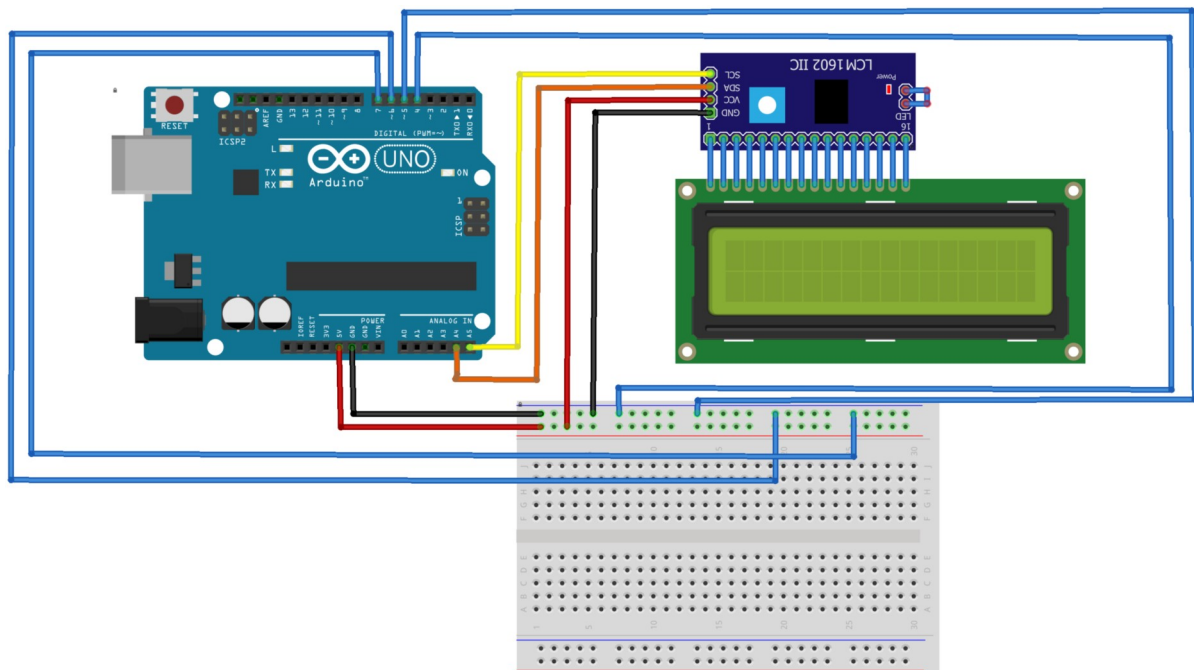
DER LCD-BILDSCHIRM

Den LCD-Bildschirm den wir verwenden siehst du auf der rechten Seite. Dieser ist in der Lage insgesamt 32 Zeichen, verteilt auf 2 Zeilen, darzustellen. Im oberen Bereich siehst du viele silberne Pins. Keine Angst – du musst nicht alle davon anschließen.



Abbildung 1: LCD-Bildschirm

Drehe den Bildschirm einmal um, dann siehst das dort ein zusätzliches Modul verbaut ist. Dieses ermöglicht es uns mit Hilfe einer Programmbibliothek nur vier Kabel anschließen zu müssen. Wie das geht zeigt dir die nächste Grafik.



fritzing



AUFGABE 6 – SCHLIESSE DEN LCD-BILDSCHIRM AN

Schließe den LCD-Bildschirm gemäß der oben stehenden Grafik an.



WAS IST EINE PROGRAMMBIBLIOTHEK?

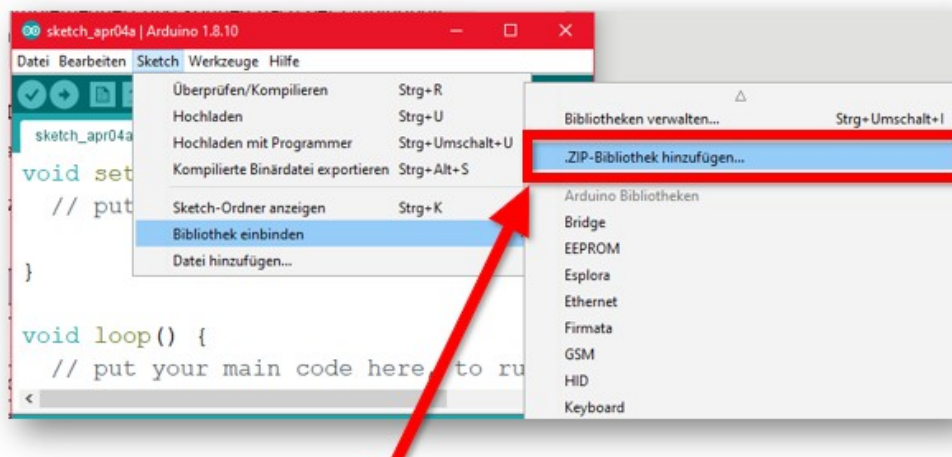
Bibliotheken sind quasi nichts anderes als eine Sammlung von Funktionen. Diese vorgefertigten Funktion können nach der Implementierung einfach genutzt werden und sollen die Arbeit des Programmierenden erleichtern.

PROGRAMMBIBLIOTHEK FÜR DEN LCD-BILDSCHIRM

Wir wollen im weiteren Verlauf die „LiquidCrystal-I2C“ Bibliothek nutzen. Wir schauen uns als erstes einmal an, welche Funktionen diese mitbringt und gehen dann zur Einbindung und Nutzung im Programm über. Nachfolgend sind einige Funktionen dargestellt:

Befehl	Bedeutung
<code>lcd.begin();</code>	Startet die Kommunikationsverbindung mit dem LCD-Bildschirm.
<code>lcd.clear();</code>	Löscht alle Inhalte auf dem LCD-Bildschirm.
<code>lcd.setCursor(0,0);</code>	Legt den Zeiger für den Beginn einer Ausgabe fest. In diesem Fall entspricht (0,0) dem ersten Zeichen in der ersten Zeile. Reihenfolge: (Spalte, Zeile).
<code>lcd.print("Hallo!");</code>	Gibt einen Text oder den Inhalt einer Variable aus.

Jetzt müssen wir uns nur noch anschauen, wie wir die Bibliothek nutzen können. Dazu steht dir zum einen eine .zip-Datei zur Verfügung, welche wir nun in die Entwicklungsumgebung importieren müssen. Du navigierst über „Sketch“ zu „Bibliothek einbinden“ und wählst dann „.ZIP-Bibliothek hinzufügen...“ aus. Nachfolgend ist dies auch nochmals grafisch gezeigt.



```

sketch_apr04a | Arduino 1.8.10
Datei Bearbeiten Sketch Werkzeuge Hilfe
✓ ↻ 📄 ⬆️ ⬇️
sketch_apr04a
void set
// put

void loop() {
// put your main code here, to ru

```

```

sketch_apr04a | Arduino 1.8.10
Datei Bearbeiten Sketch Werkzeuge Hilfe
✓ ↻ 📄 ⬆️ ⬇️
sketch_apr04a $
#include <LiquidCrystal_I2C.h>

void setup() {
// put your setup code here, to run once:

```

Weiterhin wird direkt im Programmcode eine neue Zeile eingefügt. Diese importiert die Bibliothek und macht diese für uns nutzbar. Immer wenn du auf Funktionen einer Bibliothek zurückgreifen willst, muss diese zu Beginn hinzugefügt worden sein.

AUFGABE 7 - BIBLIOTHEK EINBINDEN

Importiere zunächst die Bibliothek, bevor wir uns mit der veränderten Ausgabe näher beschäftigen.

Direkt zu Beginn der setup()-Methode startest du zunächst die Verbindung zum LCD-Bildschirm und löschst alle Zeichen auf dem Bildschirm. Dann wollen wir natürlich nicht nur die umgewandelte Dezimalzahl ausgeben, sondern auch die Eingabe. Dazu nutzen wir eine Variable des Datentyps String. In dieser wollen wir während des Auslesens der Pins speichern, ob gerade eine 1 oder eine 0 erkannt wurde. Die Initialisierung sieht wie folgt aus:

```
String binaerzahl = "";
```

Bei einer Zeichenkette können wir selbstverständlich nichts addieren, deshalb wollen wir Zeichen hinzufügen, man spricht dabei vom **Konkatenieren**. Der Operator dafür ist ebenfalls das Plus-Zeichen. In unserem Programm sieht das dann wie folgt aus:

```
binaerzahl = digitalRead(i) + binaerzahl
```

Nach Beendigung der Berechnung und Belegen der Variablen geben wir in der ersten Zeile auf dem LCD-Bildschirm die Eingabe (Binärzahl) und in der zweiten Zeile die Ausgabe (Dezimalzahl) aus.



AUFGABE 8 – AUSGABE AUF DEM LCD-BILDSCHIRM

Initialisiere die Variable zur Speicherung der Eingabe und füge die Konkatenation innerhalb der Zählschleife ein.

Setze nach der Zählschleife den Zeiger zunächst auf die erste Zeile und gib die Binärzahl aus, im Anschluss springst du zum Beginn der zweiten Zeile und gibst die Dezimalzahl aus.

Damit hast du die Aufgabe erfolgreich bewältigt. Teste dein Programm mit weiteren Binärzahlen deiner Wahl!



Grafik auf dem Deckblatt: Computer, Clker-Free-Vector-Images, Pixabay License, <https://pixabay.com/de/service/license/>, <https://pixabay.com/de/vectors/computer-notebook-laptop-304146/>

Abbildung 1: Struktog., Klaus Ramm / Thiemo Leonhardt, MIT © 2019 Didaktik der Informatik der TU Dresden, <https://gitlab.com/ddi-tu-dresden/cs-school-tools/struktog/-/blob/master/license.md>, <https://dditools.inf.tu-dresden.de/struktog/>

Abbildung 2: output of the sketch on a 16x2 LCD, alias SM, CC BY-SA 3.0, <https://creativecommons.org/licenses/by-sa/3.0/>, <https://www.arduino.cc/en/Tutorial/HelloWorld>

Screenshots: fritzing electronics made by easy und Arduino IDE 1.8.12 (windows)

Alle weiteren Grafiken: Patrick Binkert, EduInf@TUD