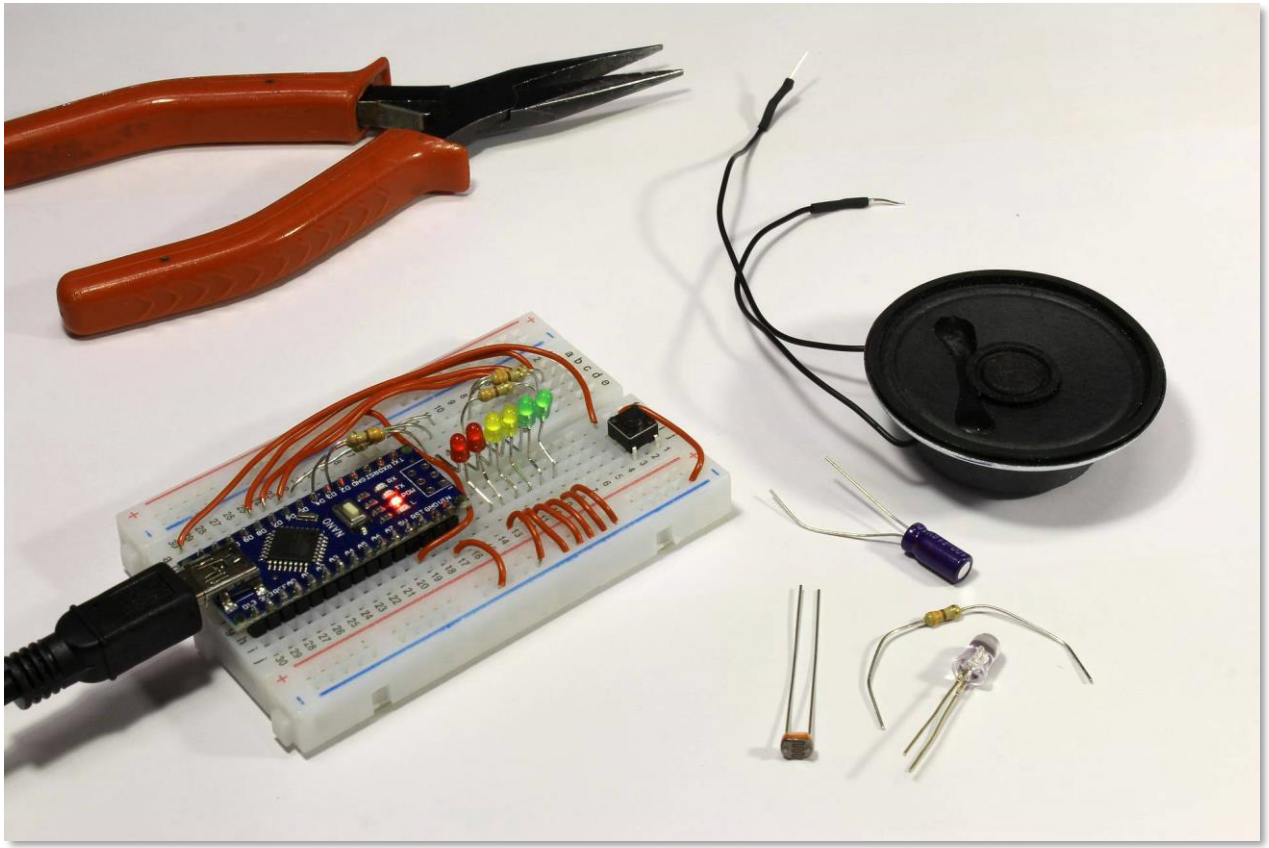
Arduino Kurs – Das LC-Display

Stephan Laage-Witt
FES Lörrach - 2018



FREIE EVANGELISCHE SCHULE
LÖRRACH

Themen

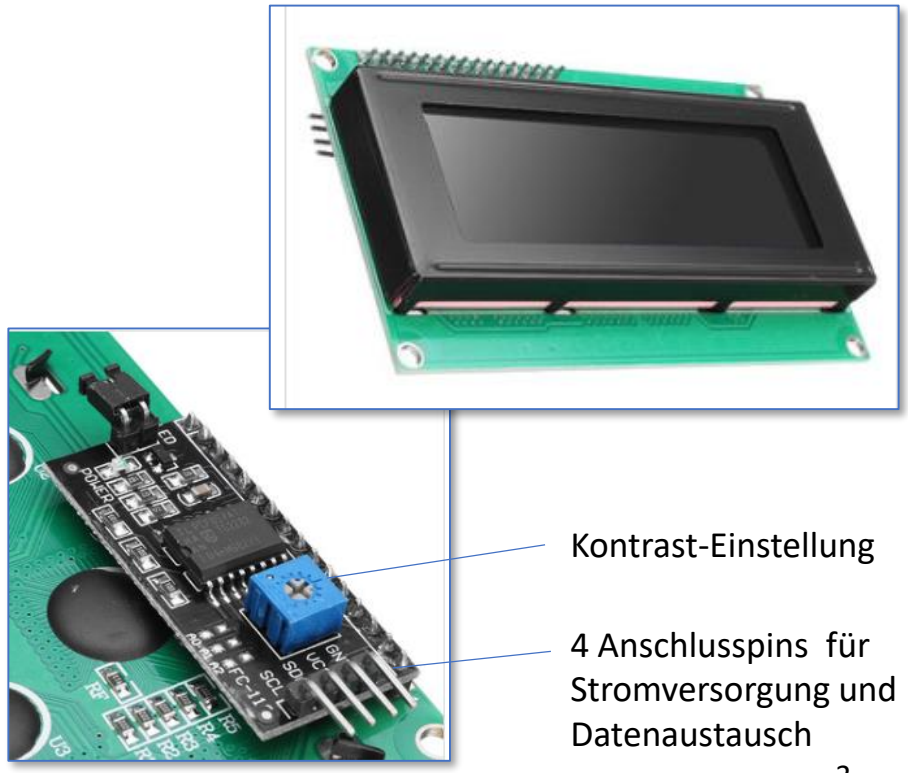
LC-Display zur Anzeige von Text
Serieller Datenbus
Ausgabe von Zeichen, Texten und Zahlen

LC-Display zur Anzeige von Text

Mikrocontroller haben verschiedene Möglichkeiten, mit der Außenwelt kommunizieren. Der einfachste Weg ist in der Regel eine (oder mehrere) Leuchtdiode. Oft werden aber mehr Information benötigt. Eine umfangreiche Kommunikations-Methode ist die Ausgabe von Text. Dazu verwenden wir ein Liquid Crystal Display, LCD.

Wir verwenden ein Display, das 4 Zeilen von jeweils 20 Zeichen darstellen kann, häufig als LCD 20x04 bezeichnet. Das Display hat eine Hintergrundbeleuchtung, die das Ablesen vereinfacht.

Das Display hat 16 Anschlüsse für Stromversorgung, Datenübertragung und Steuerung der Anzeige. Um den Anschluss an den Mikrocontroller zur vereinfachen, befindet sich auf der Rückseite eine kleine Interface-Platine, die die 16 Anschlüsse auf nur 4 Leitungen umsetzt. Außerdem befindet sich dort ein Trimmer, mit dem der Kontrast eingestellt werden kann.



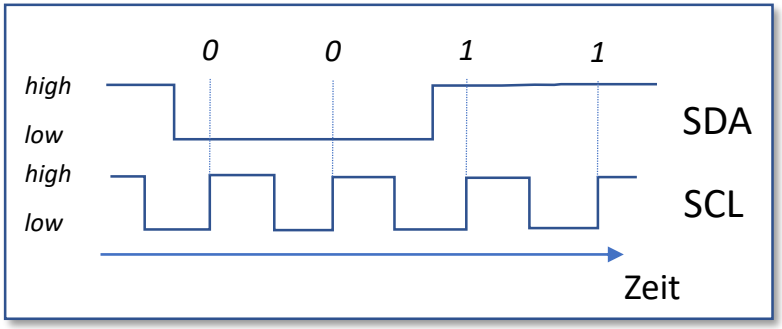
Serieller I2C-Bus

I2C ist ein serieller Bus, der ursprünglich von der Firma *Philips* erarbeitet und dann von vielen Herstellern übernommen wurde. Der Bus dient dazu, verschiedene Chips (ICs) in einer Schaltung miteinander zu verbinden. Alle Teilnehmer des Bus verwenden dieselben Leitungen. Inzwischen gibt es auch in der Arduino-Welt verschiedene I2C-Bausteine, z.B. eine Reihe von Sensoren und Aktoren.

Der I2C-Bus verwendet **2 Signal-Leitungen** und wird deshalb oft auch *Two Wire Interface* (TWI) genannt:

- SCL (Clock) gibt den Takt vor
- SDA (Data) übermittelt die Daten

Die Daten werden seriell, also ein Bit nach dem anderen übertragen. Immer, wenn die Clock-Leitung von *low* auf *high* wechselt, wird der aktuelle Zustand der Datenleitung (*low* oder *high*) als ein Bit übertragen. Nachdem 8 Bits gesendet wurden, ist ein Byte vollständig übertragen.



Der I2C-Bus verwendet eine **Master-Slave-Architektur**. Nur ein Teilnehmer im Bus ist der Master und sendet Meldungen und Anfragen für Daten aus. Alle anderen Teilnehmer hören zu und reagieren, wenn sie angesprochen werden. Dazu hat jeder Teilnehmer eine eindeutige Adresse.

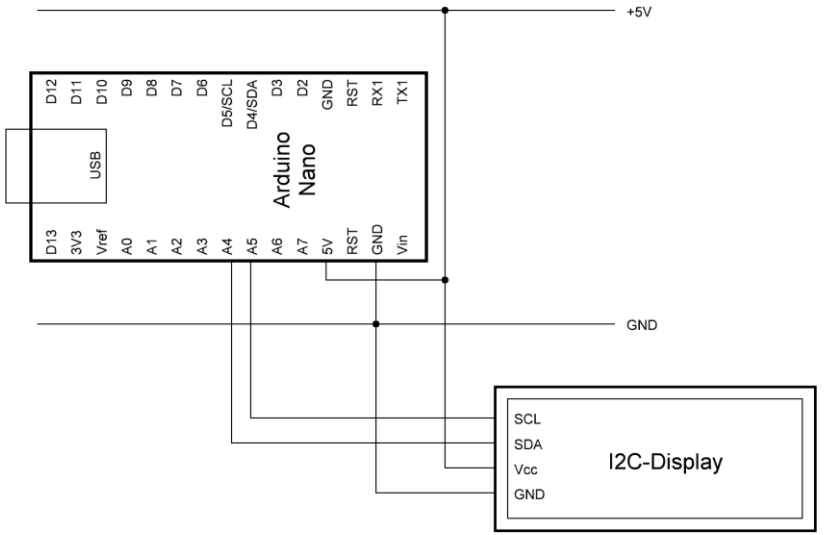
Das Display in Betrieb nehmen

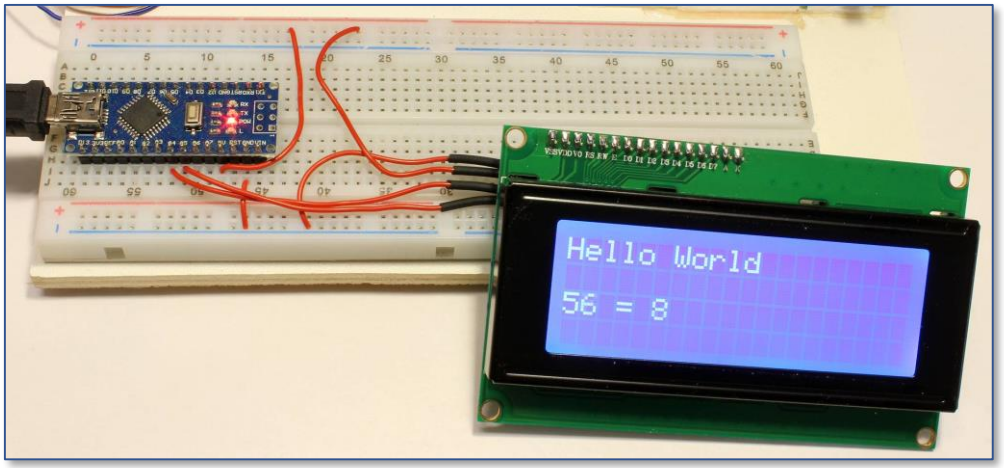
Bei unserem Aufbau ist der Arduino der „Master“. Das Display arbeitet als „Slave“.

Wir benötigen 4 Leitungen vom Display zum Arduino:

- GND geht auf die blaue Masse-Schiene
- Vcc ist die Stromversorgung und geht an die rote +5V-Schiene
- SCL ist die Taktleitung und geht an den Anschluss A5
- SDA ist die Datenleitung und geht an den Anschluss A4

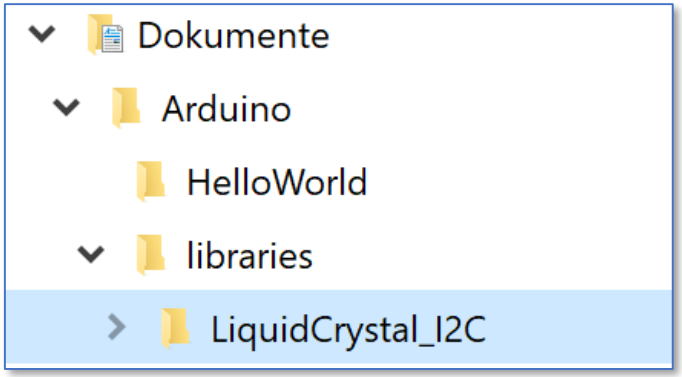
Die Anschlüsse A4 und A5 sind eigentlich analoge Eingänge, die hier aber als I2C-Interface genutzt werden.





Software für das Display

- Für den Betrieb des Displays benötigen wir Bibliotheken (libraries). Das sind vorgefertigte Programme von anderen Programmierern, die wir verwenden können, ohne sie im Detail vollständig verstehen zu müssen.
  - `Wire` ist eine Bibliothek, um das I2C-Interface zu verwenden. Diese Bibliothek ist bei Arduino bereits vorinstalliert. Sie unterstützt die Interaktion mit einer Vielzahl von I2C-Geräten.
  - `LiquidCrystal_I2C` ist eine Bibliothek, die die Funktionen zur Ausgabe auf das Display bereit stellt. Diese Bibliothek ist nicht automatisch vorhanden und muss deshalb in den Ordner Arduino -> libraries kopiert werden.



- Die wichtigsten Funktionen in der Bibliothek `LiquidCrystal_I2C` sind:
  - `lcd.setCursor(spalte, zeile);` für das Setzen des Ausgabe-Cursors
  - `lcd.print();` für die Ausgabe von Zeichen, Texten und Zahlen
  - `lcd.backlight();` schaltet die Hintergrundbeleuchtung ein

Hello World!

HelloWorld

```
#include <Wire.h>
#include <LiquidCrystal_I2C.h>

// Set the LCD address to 0x3f for a 20 chars / 4 line display
LiquidCrystal_I2C lcd(0x3f, 20, 4);

/* setup() ------------------------------
void setup()
{
  // initialize the LCD
  lcd.init();

  // Turn on the backlight and print a message.
  lcd.backlight();
  lcd.setCursor(0, 0);
  lcd.print("Hello World");
}

/* loop() -------------------------------
void loop()
{
  int i;
  char c;

  for (i = 32; i < 255; ++i) {
    lcd.setCursor(0, 2);
    lcd.print(i);
    lcd.print(" = ");
    c = char(i);
    lcd.print(c);
    delay(500);
  };
}
```

Display-Parameter setzen: I2C-Adresse und Display-Größe

Display initialisieren

Beleuchtung einschalten

Cursor setzen

Text ausgeben

Cursor setzen

Zeichen-Code als Zahl anzeigen

Kurzen Text ausgeben

Zeichen ausgeben, das zum entsprechenden Code gehört

Wenn Programmierer es mit einem neuen, für sie unbekannten System zu tun bekommen, hat es sich eingebürgert, zuerst einmal den Text „Hello World!“ auszugeben. Traditionen soll man bewahren! Deshalb schreibt dieses Programm dieselbe Nachricht an die Welt … und macht noch mehr.

In der Setup()-Funktion wird das Display initialisiert und der Text ausgegeben.

In der Loop()-Funktion werden alle Zeichen, die das Display kennt, mit ihrem zugehörigen Zeichen-Code der Reihe nach angezeigt.

ASCII Zeichensatz

Das Display verfügt über einen eingebauten Zeichensatz. Dabei bekommt jedes Zeichen eine Nummer (Code) Schon seit Anbeginn der Computerzeit gibt es eine weltweit gültige Vereinbarung, welcher Code zu welchem Zeichen gehört.

Die Werte 0 bis 31 sind für Sonderaktionen reserviert (z.B. bedeutet 13 „neue Zeile“). Auf dem Display sind diese Sonderaktionen nicht verfügbar. 32 ist das Leerzeichen. Im Bereich 33 bis 127 befinden sich die üblichen Zeichen und Buchstaben, z.B. 65 für ‘A‘.

Von 128 bis 255 gibt es spezielle Zeichen, die je nach Ausgabegerät verschieden sein können.

Datentypen

In der Programmiersprache C gibt es verschiedene Datentypen. Jede Variable und jede Konstante hat einen definierten Datentyp. Es ist wichtig, zu jedem Zeitpunkt zu wissen, welchen Datentyp man gerade verwendet.

Unser Programm benutzt folgende Datentypen:

`int i;` Hier wird eine Variable vom Typ integer definiert. Die Variable kann ganze Zahlen abspeichern, z.B. `i = 1267;`. Die Anweisung `lcd.print(i);` gibt den aktuellen Zahlenwert der Variable aus.

Beim Arduino reicht der zulässige Zahlenbereich von etwa -32000 bis +32000. Andere Computer können andere Wertebereiche verwenden.

`char c;` Damit wird eine Variable definiert, die ein einzelnes Zeichen annehmen kann, also einen beliebigen Wert aus der ASCII-Tabelle, z.B. `c = 'A';` Die Anweisung `c = char(i);` konvertiert die Integer-Zahl `i` in das entsprechende Zeichen aus der ASCII-Tabelle. `lcd.print(c);` gibt das Zeichen auf dem Display aus.

`"Hello World!";` ist ein String, eine Aneinanderreihung von Zeichen. Am Ende des Strings wird automatisch ein Zeichen mit dem Wert 0 eingefügt, um so das Ende zu markieren.

Strings sind ein Beispiel für Arrays, die an anderer Stelle im Detail besprochen werden.