

# Weihnachtstürchen Nr. 4

In den vergangenen drei Tagen haben Sie Grundkenntnisse zum Thema Aufbau einer kleinen Schaltung, Löten und Inbetriebnahme eines MicroControllers bekommen. Was bisher die Weihnachtswichtel nicht herausgegeben haben, war Peripherie für den MicroController. Dies wolle die Weihnachtswichtel heute korrigieren und haben gleich 6 Bauteile hinter Tür vier versteckt.

## Was ist drin hinter Türchen Nummer 4

Hinter Türchen Nummer vier befinden sich:

- 3 x HD44780 1602 LCD Modul Display (grüne Beleuchtung)
- 3 x I2C-Schnittstellenadapter zum anlöten

Zusätzlich benötigen Sie die Bibliothek **LiquidCrystal** und **LiquidCrystal\_I2C**, wie das geht, zeigen wir u.a. [hier](#).

Generell haben Sie drei verschiedene Möglichkeiten, das Display mit einem MicroController zu verbinden, siehe Abbildung 1.

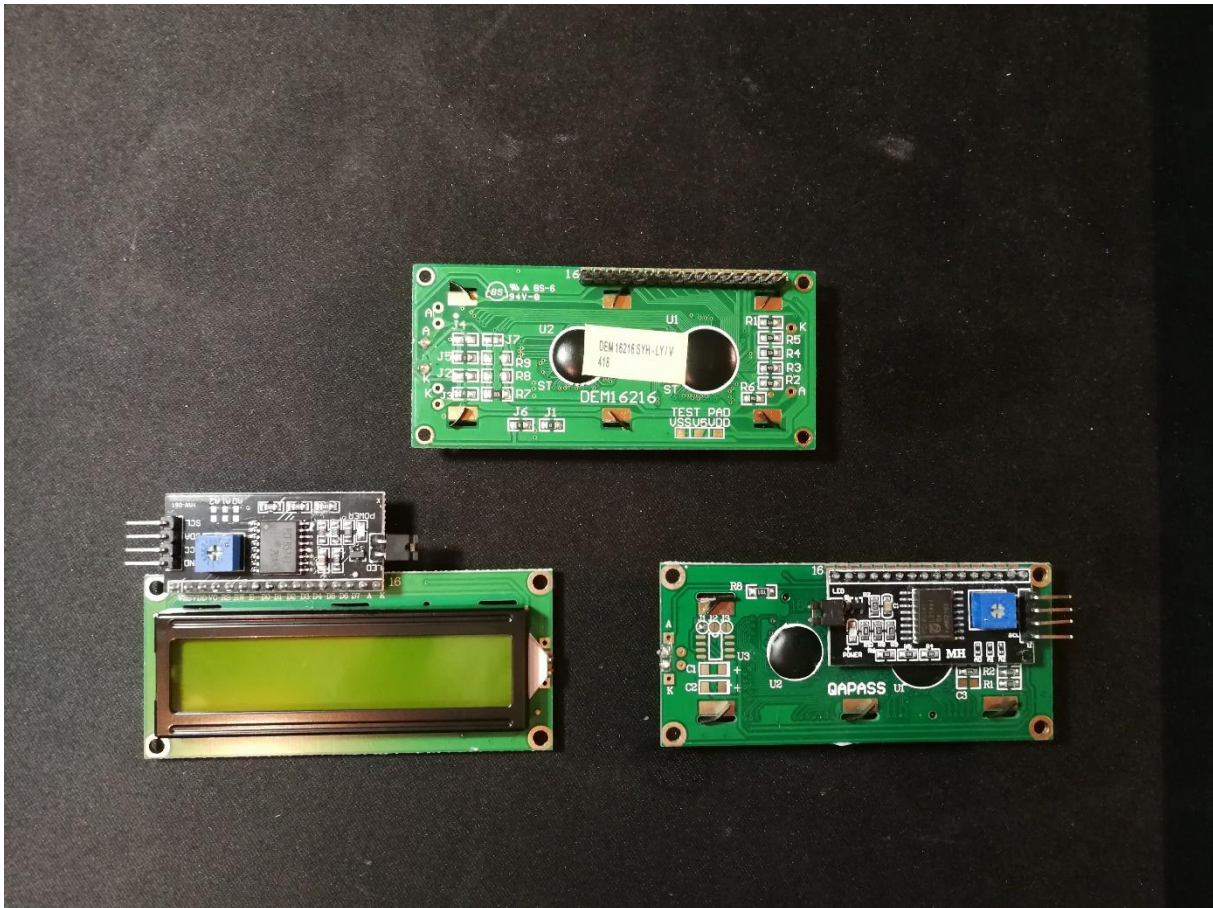


Abbildung 1: HD44780 1602 LCD Modul Displays Anschlussvarianten

Die erste Variante, im Bild oben, ist die Ursprungsvariante mit angelöteter Stiftleiste (nicht im Lieferumfang). Dies bedeutet aber auch, dass auf einem MicroController mindestens 6 Pins belegt werden, gleich mehr dazu. Die andere Variante ist mit dem angelöteten I2C-Schnittstellenadapter, der wahlweise vorne oder hinten angelötet werden kann. Achten Sie darauf, dass die vier Stiftleisten zum Anschluss an den MicroController immer nach außen zeigen, da sonst das HD44780 1602 LCD Modul Display zerstört werden kann.

## Verbindung klassisch

In der klassischen Variante wird das LCD-Display an insgesamt 6 Pins mit dem MicroController verbunden. Abbildung 2 zeigt eine solche Verbindung, wobei das 10kOhm Potentiometer für den Kontrast am Display verantwortlich ist.

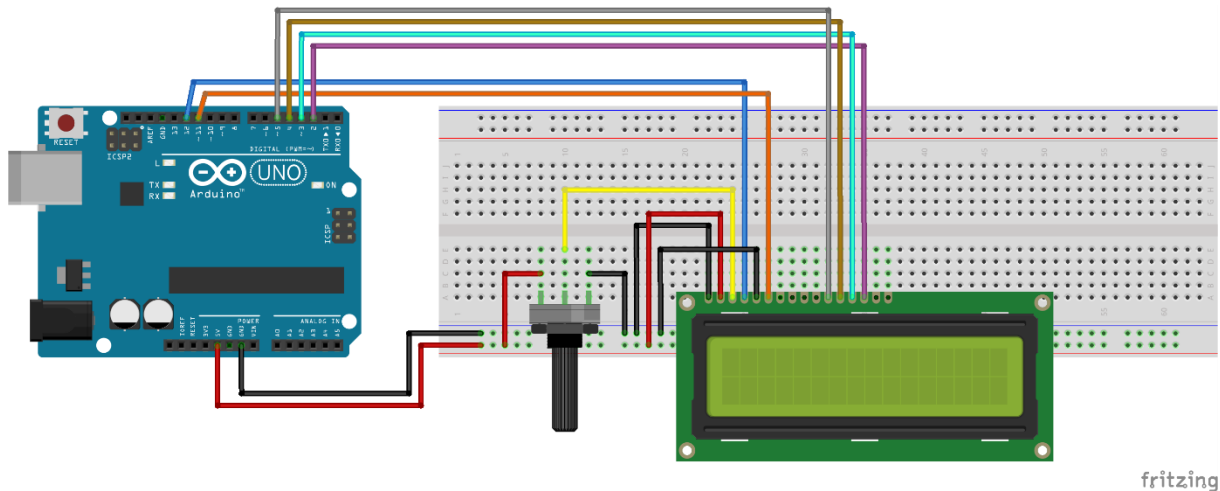


Abbildung 2: LCD-Anschluss paralleles Interface

Der entsprechende Code dazu, siehe Code 1, ist schlank und gibt in zwei Zeilen die Nachricht „LCD-Display without ic2“ aus.

```
//-----  
// LiquidCrystal parallel interface  
// Autor: Joern Weise  
// License: GNU GPI 3.0  
// Created: 24. Nov 2020  
// Update: 24. Nov 2020  
//-----  
#include <LiquidCrystal.h>           //include the LiquidCrystal library  
  
LiquidCrystal lcd(12, 11, 5, 4, 3, 2); //define LCD pins (RS, E, D4, D5, D6, D7)  
  
void setup() {  
  lcd.begin(16, 2); //initializes the LCD and specifies the dimensions  
  lcd.home();  
  lcd.print(" LCD-Display ");  
  lcd.setCursor(0, 1);  
  lcd.print(" without i2c ");  
}  
  
void loop()  
{  
  
}
```

Code 1: Simple Ausgabe

## Verbindung mit I2C-Schnittstellenadapter

Wollen Sie das Display mit I2C betreiben, müssen Sie den I2C-Schnittstellenadapter an das Display löten. Vom Prinzip können Sie, sofern sie über die Adresspads verschiedene Hex-Adressen einstellen, bis zu 8 LCD-Displays betreiben. Tabelle 1 zeigt, wie Sie welchen

Adresspad löten müssen, um die gewünschte HEX-Adresse zu bekommen, siehe Abbildung 3. „X“ steht für verbunden, „0“ für nicht verbunden.

| Pos | A0 | A1 | A2 | Hex-Adresse |
|-----|----|----|----|-------------|
| 1   | X  | X  | X  | 0X20        |
| 2   | 0  | X  | X  | 0X21        |
| 3   | X  | 0  | X  | 0X22        |
| 4   | 0  | 0  | X  | 0X23        |
| 5   | X  | X  | 0  | 0X24        |
| 6   | 0  | X  | 0  | 0X25        |
| 7   | X  | 0  | 0  | 0X26        |
| 8   | 0  | 0  | 0  | 0X27        |

Tabelle 1: Hex-Adresse für Adresspads

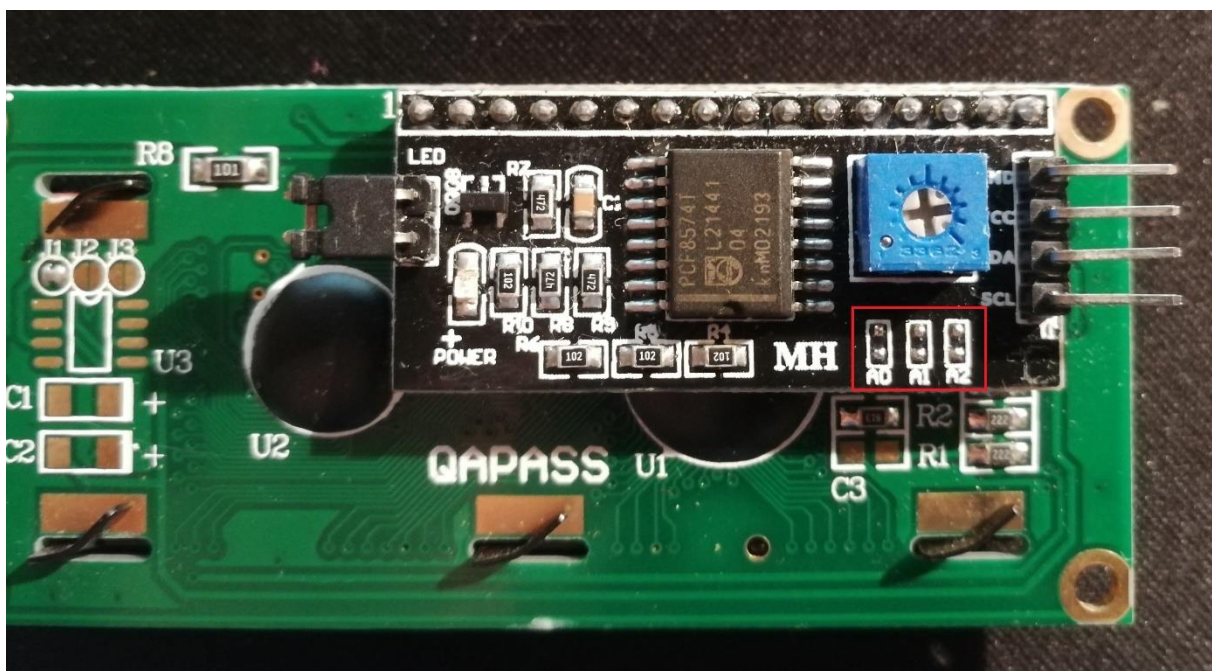


Abbildung 3: Position Adresspads zum Ändern der HEX-Adresse

Der Anschluss an einen z.B. Uno R3 ist recht simpel, da die I2C-Pins des MicroControllers verwendet werden, siehe Abbildung 4.

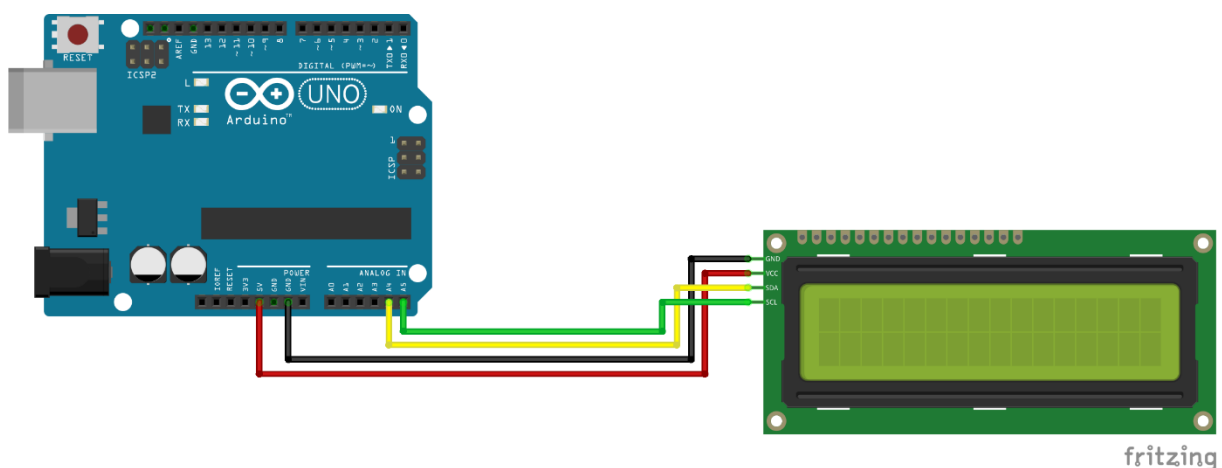


Abbildung 4: Anschluss HD44780 1602 LCD mit I2C an Uno R3

Code 2 gibt auch auf zwei Zeilen eine Nachricht aus, hier ist es der Satz „LCD-Display with i2c“.

```
//-----  
// LiquidCrystal I2C connection  
// Autor: Joern Weise  
// License: GNU GPI 3.0  
// Created: 24. Nov 2020  
// Update: 24. Nov 2020  
//-----  
#include <Wire.h>  
#include <LiquidCrystal_I2C.h>
```

```
LiquidCrystal_I2C lcd(0x27,20,4); // set the LCD address to 0x27 for a 16 chars and 2 line display
```

```
void setup()  
{  
  lcd.init(); // initialize the lcd  
  lcd.backlight(); // backlight on  
  
  lcd.home(); // set cursor to 0,0  
  
  //Write some text  
  lcd.print(" LCD-Display ");  
  lcd.setCursor(0, 1);  
  lcd.print(" with i2c ");  
}
```

```
void loop()  
{  
  
}
```

*Code 2: Simple Ausgabe via i2c*

Natürlich ist auch die Kombination beider Anschlussvarianten möglich, dazu müssen Sie die Schaltung aus Abbildung 2 und Abbildung 4 gleichzeitig aufbauen.

In diesem Sinne....



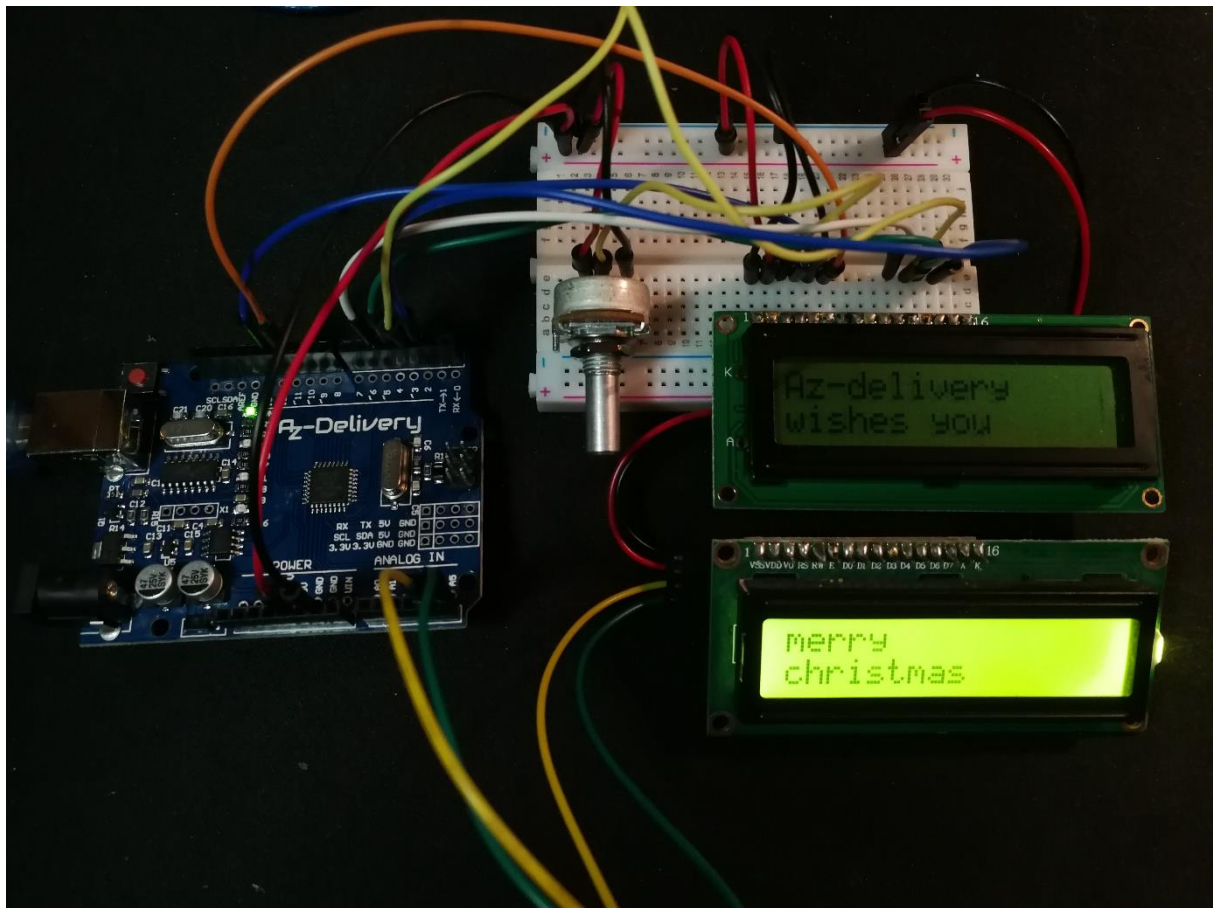


Abbildung 5: We wish a merry xmas

Damit haben Sie Türchen Nummer 4 in Betrieb genommen und noch ein bisschen Hintergrundwissen erhalten. Sie wollen Projektideen, dann hier einige Projekte:

- [Elektronisches Namensschild](#)
- [Binärumrechner mit Arduino und I2C-LCD-Display](#)

Weitere Projekte für AZ-Delivery von mir, finden Sie unter <https://github.com/M3taKn1ght/Blog-Repo>.