

Display 16 Zeichen mit I2C Adapter

Beschreibung

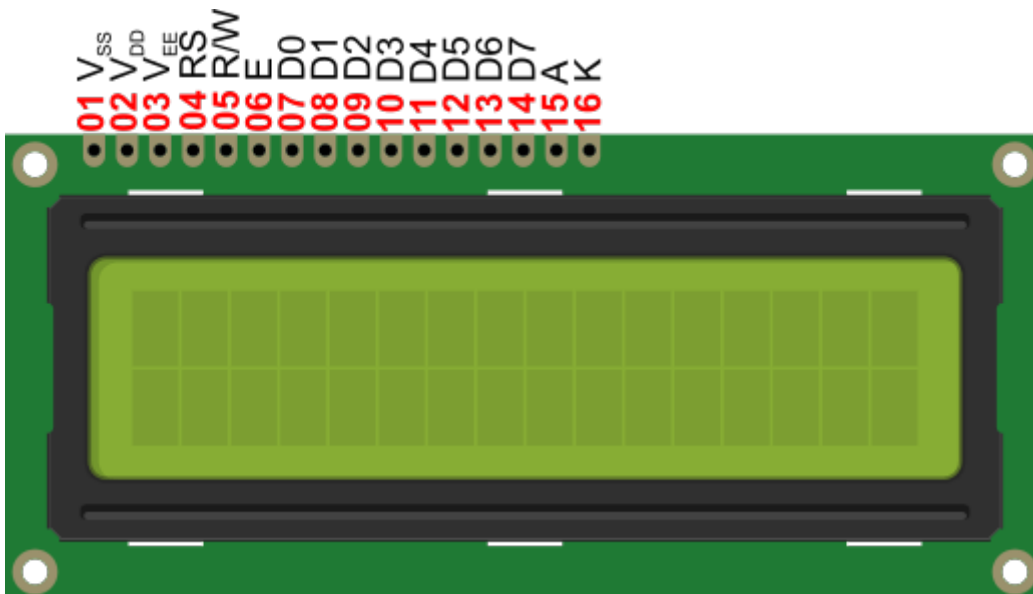
Das I2C Schnittstellen Modul vereinfacht die Ansteuerung der LCD Displays 1602 und 2004 erheblich.

Normal brauchst du zur Ansteuerung der Displays 6 I/O vom Arduino und eine Leitung, um den Kontrast einzustellen.

Dazu kommt noch die 5V Speisung und GND (Insgesamt also 9 Verbindungen).

Mit dem I2C Schnittstellen Modul brauchst du nur 2 Steuerleitungen, 5V Speisung und GND (Insgesamt nur 4 Verbindungen). Das LCD Schnittstellen Modul wird über den I2C Bus anhand der eingestellten Adresse angesprochen. Standardmässig ist das Modul auf die Adresse 0x27 eingestellt. Die Adresse kann durch löten von Brücken bei A0, A1 und A2

Anschlussbelegung Display



Es werden folgende PINS angeschlossen:

- **Pin 1 (V_{SS})** und **Pin 2 (V_{DD})** dienen der Stromversorgung des Displays und der Ansteuerungselektronik. Pin 1 ist dabei auf Masse zu legen, auf Pin 2 sind +5 V

Versorgungsspannung zuzuführen.

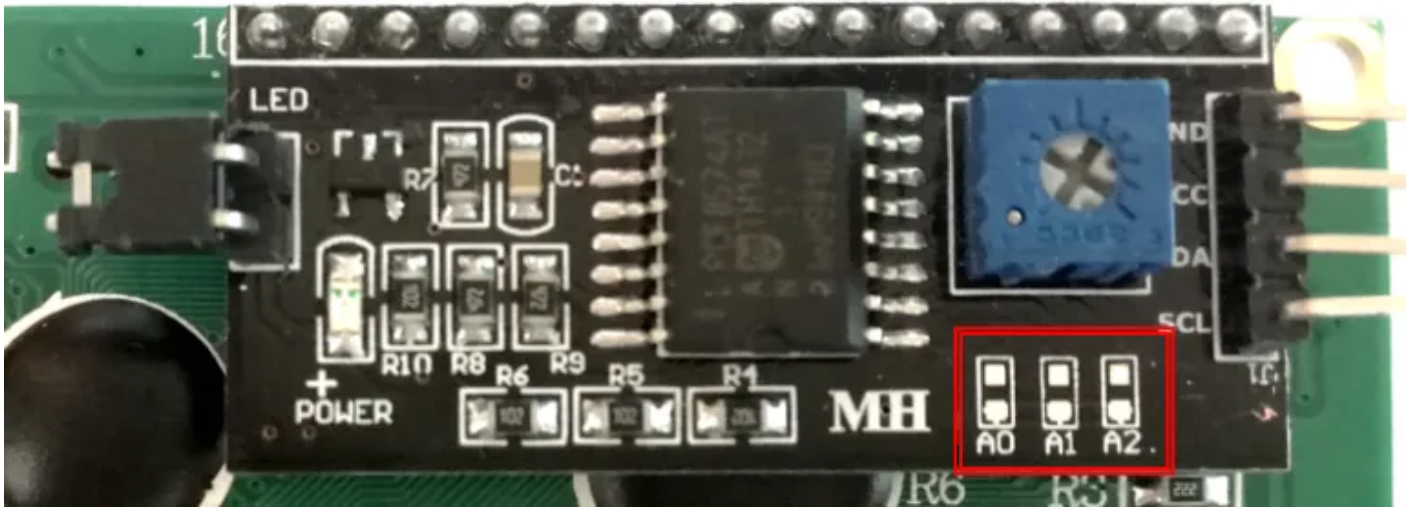
- **Pin 3 (V_{EE})** ist ein analoger Eingang und dient der Kontrastregelung des Displays. Der Wert muss zwischen 0 V und +5 V liegen.
- **Pin 4 (RS)** ist ein digitaler Eingang und bestimmt, ob die zum Display übermittelten Datenbits als Befehl (LOW) oder Zeichendaten (HIGH) interpretiert werden sollen.
- **Pin 5 (R/W)** ist ein digitaler Eingang, der entscheidet, ob Daten auf dem Display geschrieben (LOW) oder vom Display eingelesen (HIGH) werden sollen. Es ist also tatsächlich möglich, den Inhalt des Displays wieder mit dem Arduino einzulesen. In der Praxis ist das aber eigentlich nie erforderlich. Daher legt man diesen Pin einfach dauerhaft auf Masse (LOW).
- **Pin 6 (E)** ist ein digitaler Eingang, der auf HIGH geschaltet werden muss, damit das Display die an den Datenpins anliegenden Bits ausliest.
- **Pin 7 - Pin 14 (D0 - D7)** sind die 8 Bits des bidirektionalen, parallelen Datenbusses. Da man ungern ganze 8 Ports des Arduinos nur für die Datenübertragung zum Display verbrauchen möchte, nutzt man die Fähigkeit der Ansteuerungselektronik, in den 4-Bit-Modus zu schalten. In diesem Fall werden nur die hinteren Pins 11 - 14 (D4 - D7) mit dem Arduino verbunden und die 8 Bit in zwei Schritten (jeweils 4 Bit) nacheinander übertragen. Die Pins 7 - 10 lässt man einfach offen.
- **Pin 15 (A)** und **Pin 16 (K)** existieren nur an LCD mit eingebauter Hintergrundbeleuchtung und dienen der Stromversorgung selbiger. An Pin 15 (Anode) kommt die Versorgungsspannung, Pin 16 (Kathode) wird auf Masse gelegt. Je nach LCD muss hier entweder ein Vorwiderstand für die im LCD verbaute LED vorgeschaltet werden oder aber der entsprechende Widerstand befindet sich bereits im LCD. Wenn man sich unsicher ist und kein Hinweis darauf beim LCD zu finden ist, kann man vorsichtshalber einen 220 Ω -Widerstand einbauen.

Anschluss IC2 Modul an LCD Display

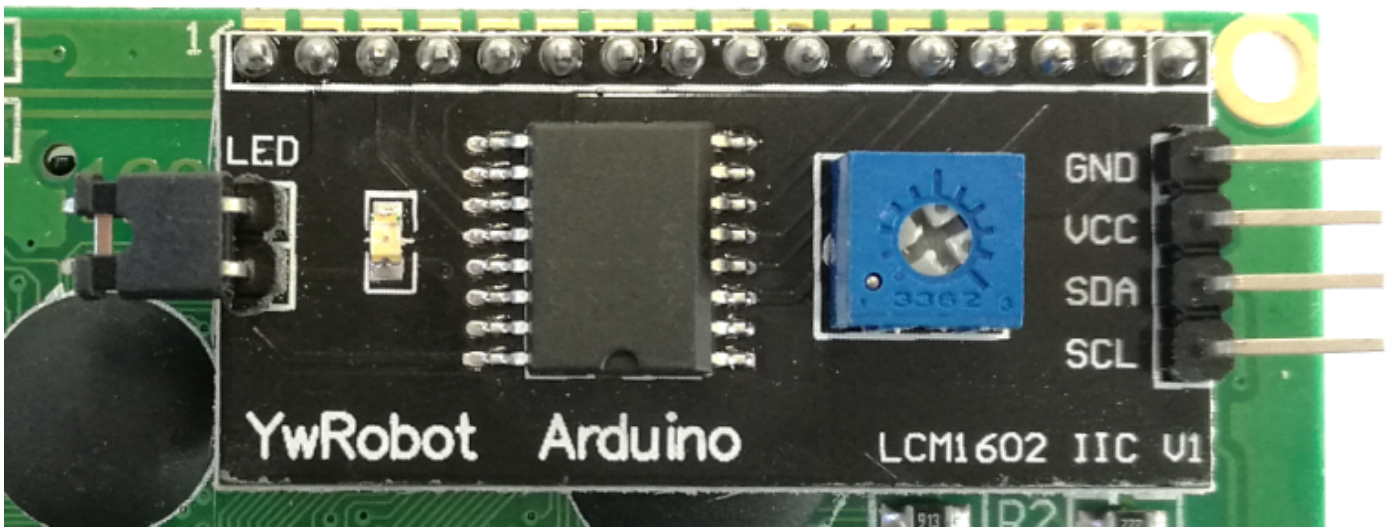
Es gibt zwei Varianten von IC2 LCD Modulen. Einmal mit Lötstellen A0 -A3. Standardmäßig ist hier meist auch 0x27 eingestellt

Über die Lötstellen kann die Adresse eingestellt werden. Bei beiden Varianten ist oben rechts GND für PIN1 vom LCD Display.

Dann die Kontakte einfach 1:1 verbinden.



und einmal ohne, da ist die Adresse fest meist 0x27



I2C Schnittstelle => Arduino (A4 und A5 sind die Seriellen Datenleitungen. GND und 5V dienen der Spannungsversorgung des Moduls

GND => GND

VCC => 5V

SDA => A4

SCL => A5

Ändern der Adresse

Jedes I²C Modul hat eine sogenannte „HEX Adresse“. Über diese Adresse reagiert das I²C-Modul auf die Daten, die vom Arduino auf dem Datenbus an genau diese Adresse gesendet werden. Viele I²C-LCDs haben auch die gleiche HEX-Adresse. Das bedeutet, dass beim Verwenden von zwei Displays beide Displays auf die gesendeten Daten vom Arduino-Board reagieren würden. Man könnte also auf zwei Displays keine unterschiedlichen Daten darstellen.

Die HEX-Adresse kann bei dem Display mit Hilfe der A0, A1 und A2 Lötstellen jedoch verändert werden. Im unveränderten Zustand sind alle drei Lötstellen nicht verbunden. Je nach Kombination, welche der Stellen man mit einer Lötstelle überbrückt, sind also 8 verschiedene Adressen möglich. Abhängig vom Display Typ kann diese Adresse anfangs 0x27 oder 0x3F sein (kann mit dem Adressen „Scanner“ herausgefunden werden, dazu später mehr).

Der Kontrast des LCD Displays kann über das Blaue Potentiometer auf dem Modul eingestellt werden. Die LCD Hintergrundbeleuchtung kann durch entfernen des Jumpers «LED» deaktiviert werden. Für die Ansteuerung gibt es bereits eine fertige Arduino Library, die du im Library Manager unter «LiquidCrystal_I2C» findest.

Tabellen zu HEX Adressen je nach verlöteten Stellen(**I** = verbunden, **:** = nicht verbunden):

A0	A1	A2	HEX Adresse	HEX Adresse
:	:	:	0x27	0x3F
I	:	:	0x26	0x3E
:	I	:	0x25	0x3D
I	I	:	0x24	0x3C
:	:	I	0x23	0x3B
I	:	I	0x22	0x3A
:	I	I	0x21	0x39
I	I	I	0x20	0x38

IC2 Hex Scanner

```
// I2C Scanner
// Written by Nick Gammon
// Date: 20th April 2011
#include <Arduino.h>
#include <Wire.h>
void setup() {
  Serial.begin (9600);
  // Leonardo: wait for serial port to connect
  while (!Serial)
  {
  }
  Serial.println ();
  Serial.println ("I2C scanner. Scanning ...");
  byte count = 0;
```

```
Wire.begin();
for (byte i = 8; i < 120; i++)
{
Wire.beginTransmission (i);
if (Wire.endTransmission () == 0)
{
Serial.print ("Found address: ");
Serial.print (i, DEC);
Serial.print (" (0x");
Serial.print (i, HEX);
Serial.println ("");
count++;
delay (1); // maybe unneeded?
} // end of good response
} // end of for loop
Serial.println ("Done.");
Serial.print ("Found ");
Serial.print (count, DEC);
Serial.println (" device(s).");
} // end of setup
void loop() {}
```

In der Platform.ini folgendes hinzufügen

```
...
; Custom Serial Monitor port
monitor_port = COM4

; Custom Serial Monitor speed (baud rate)
monitor_speed = 9600
...
```

Nun unten in der Leiste auf den Stecker klicken. Dort bekommen wir nun die Serialausgabe. In diesem Fall wissen wir nun das unser Display den Hex Wert 0x27 hat. Siehe Ausgabe. Dank des Hexscanners.

```
TERMINAL  PROBLEME 11  AUSGABE  DEBUGGING-KONSOLE
Task wird im Ordner LCDIC2 ausgeführt: platformio device monitor

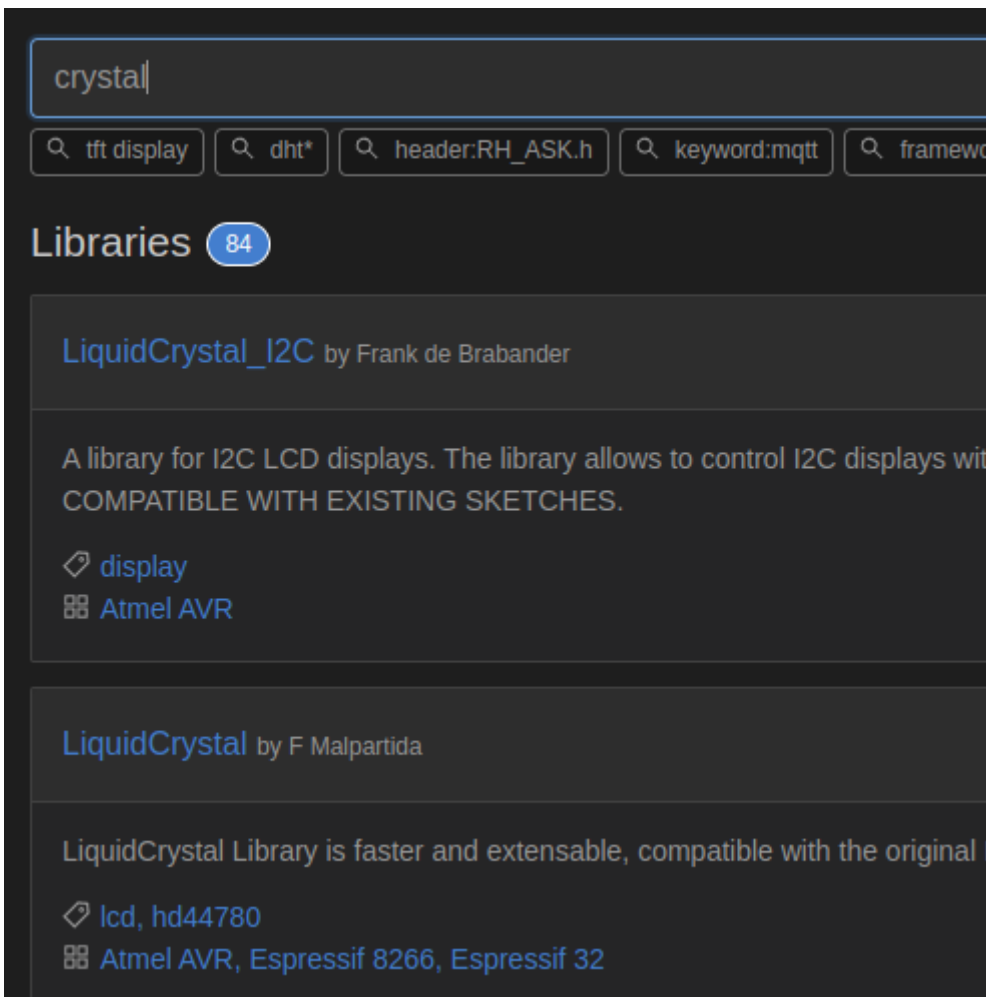
--- Terminal on /dev/ttyACM0 | 9600 8-N-1
--- Available filters and text transformations: colorize, debug, default, direct, hexlify, log2file, nocontrol, printable, send_on_enter, time
--- More details at https://bit.ly/pio-monitor-filters
--- Quit: Ctrl+C | Menu: Ctrl+T | Help: Ctrl+T followed by Ctrl+H

I2C scanner. Scanning ...
Found address: 39 (0x27)
Done.
Found 1 device(s).
```

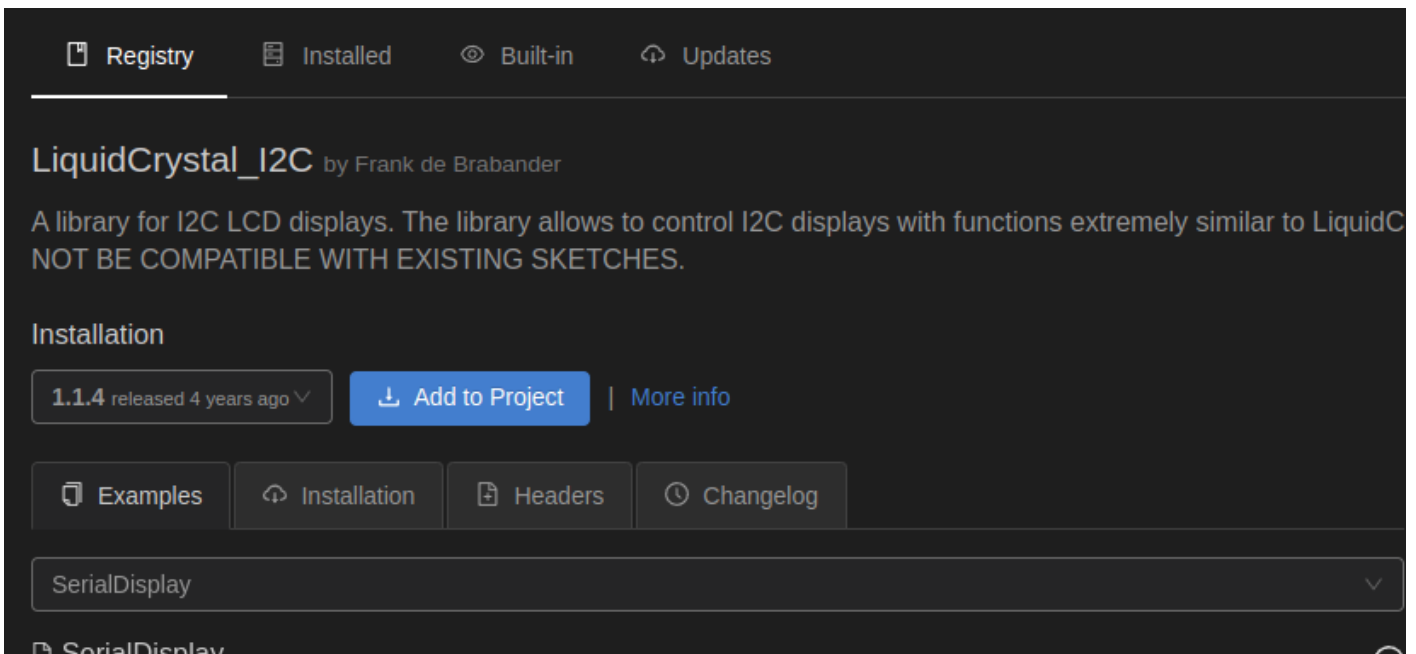
Einbindung 16 Zeichen Display

Bibliotheken installieren

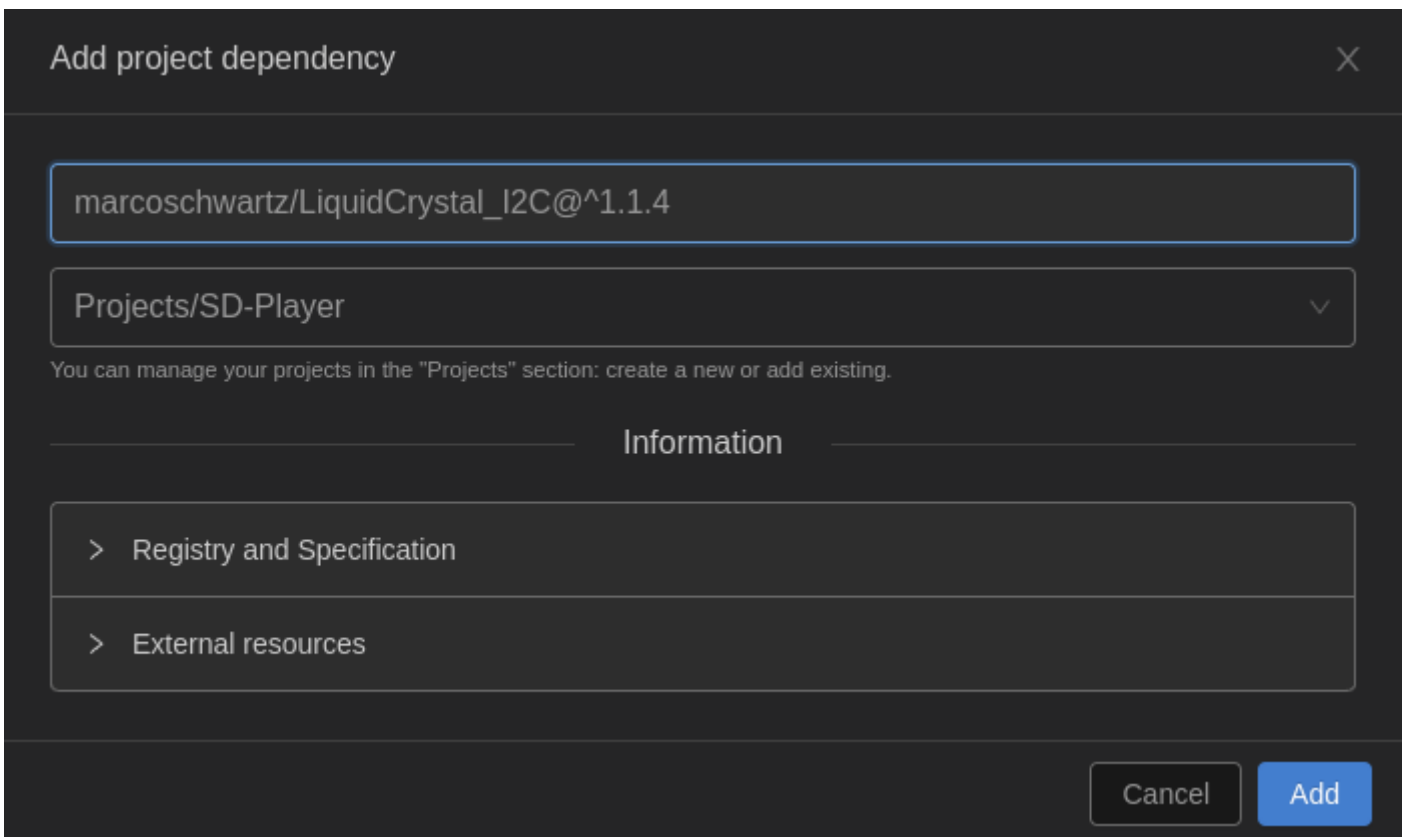
Wir benötigen dazu die LiquidCrystal Library IC2 (Diese im PIO-Home Library herunterladen)
Dazu den Suchbegriff crystal eingeben



Dann auf Add to Project klicken



Und nun das Project auswählen



Code

```
#include <Arduino.h>
#include <Wire.h> // Wire Bibliothek einbinden
#include <LiquidCrystal_I2C.h> // Vorher hinzugefügte LiquidCrystal_I2C Bibliothek einbinden
LiquidCrystal_I2C lcd(0x27, 16, 2); //Hier wird festgelegt um was für einen Display es sich handelt. In diesem Fall
eines mit 16 Zeichen in 2 Zeilen und der HEX-Adresse 0x27. Für ein vierzeiliges I2C-LCD verwendet man den
```

```
Code "LiquidCrystal_I2C lcd(0x27, 20, 4)"
```

```
void setup()
{
  lcd.init(); //Im Setup wird der LCD gestartet
  lcd.backlight(); //Hintergrundbeleuchtung einschalten (lcd.noBacklight(); schaltet die Beleuchtung aus).
}

void loop()
{
  lcd.setCursor(0, 0); //Hier wird die Position des ersten Zeichens festgelegt. In diesem Fall bedeutet (0,0) das erste Zeichen in der ersten Zeile.
  lcd.print("Hacker-Net Telekommuniktion");
  lcd.setCursor(0, 1); // In diesem Fall bedeutet (0,1) das erste Zeichen in der zweiten Zeile.
  lcd.print("Viel Erfolg!");
}
```

Quelle

<https://funduino.de/nr-19-i%C2%B2c-display>

<https://www.bastelgarage.ch/i2c-schnittstelle-pcf8574-fur-lcd-display>

<https://funduino.de/nr-06-zwei-i%C2%B2c-displays-gleichzeitig>

Version #6

Erstellt: 29 Dezember 2022 02:25:33 von Admin

Zuletzt aktualisiert: 29 Dezember 2022 10:36:17 von Admin