

I2C Serial 2.6" LCD Modul

Sehr geehrter Kunde,

vielen Dank, dass Sie sich für unser Produkt entschieden haben.

Im Folgenden haben wir aufgelistet, was bei der Inbetriebnahme zu beachten ist:

Verwendung mit einem Arduino

Schritt 1 – Anschließen des Displays

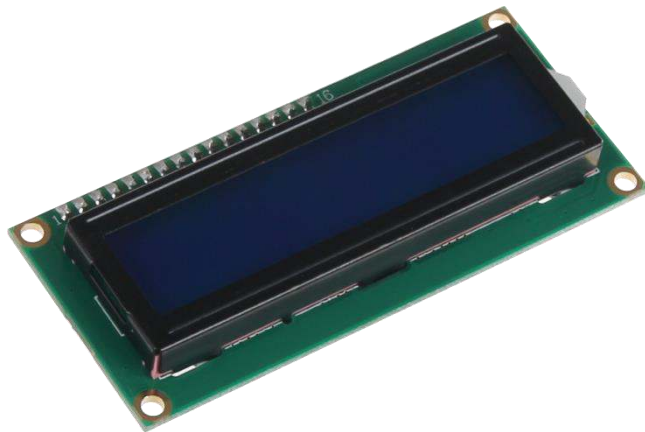


Bild 1: I2C Serial 2.6" LCD-Display

Schließen Sie das Display, wie im folgenden Bild 2, bzw. in folgender Tabelle 1, zu sehen, an die PINS des Arduinos an.

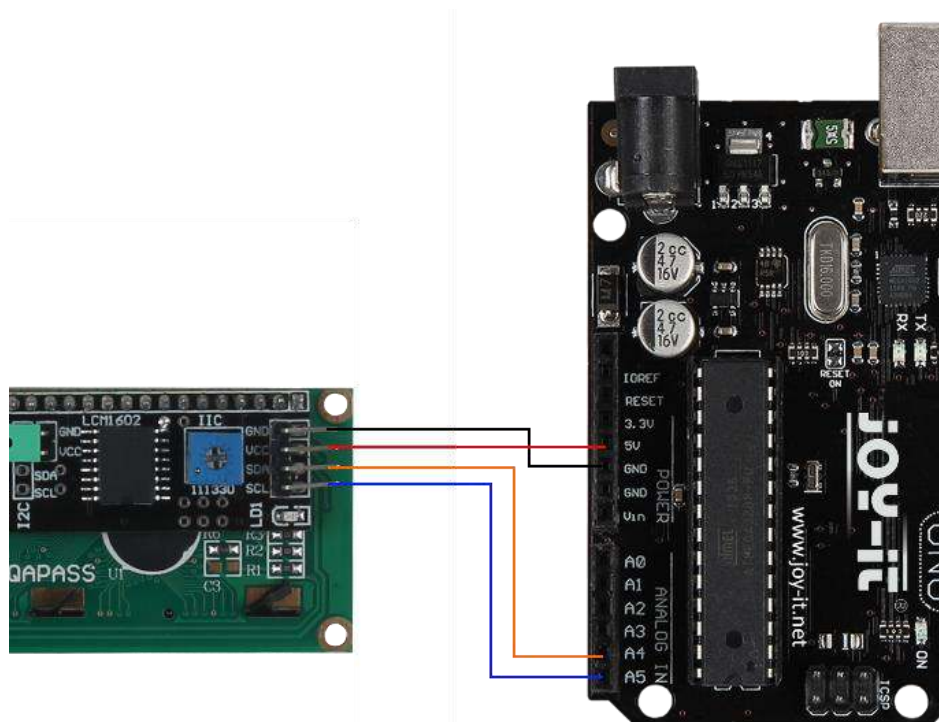


Bild 2: Verkabelung mit Arduino

I2C Serial LCD PIN	Arduino PIN
GND	GND
VCC	+5V
SDA	A4
SCL	A5

Tabelle 1: PIN-Verbindung zwischen Arduino und Serial-LCD

Schritt 2 – Installation der Bibliothek

Bevor Sie den unten befindlichen Quellcode auf Ihren Arduino übertragen, muss zunächst die **LiquidCrystal_I2C** Bibliothek hinzugefügt werden.

Dazu klicken Sie bitte, wie in Bild 3 zu sehen, auf Sketch → Bibliothek einbinden → Bibliotheken verwalten.

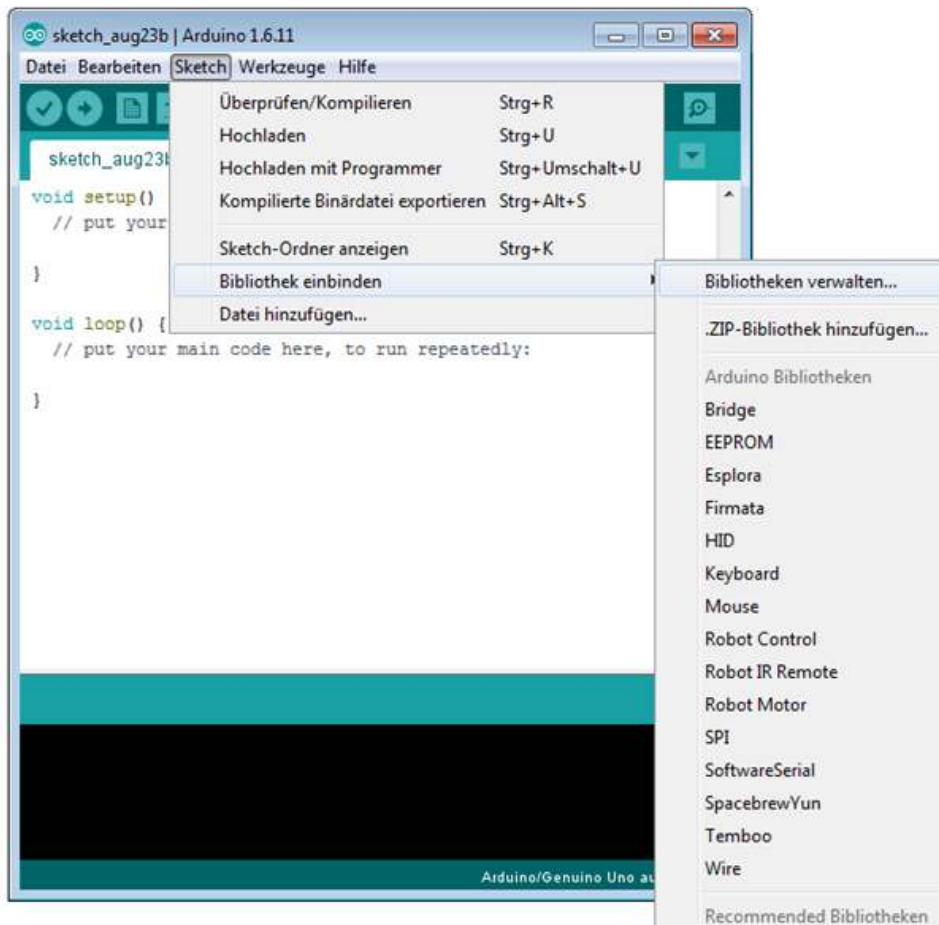


Bild 3: Bibliothekverwaltung im Arduino-Sketch

Im sich darauf öffnenden Bibliotheksverwalter suchen Sie, unter dem Suchbegriff **LiquidCrystal_I2C** die gleichnamige Bibliothek und installieren diese.

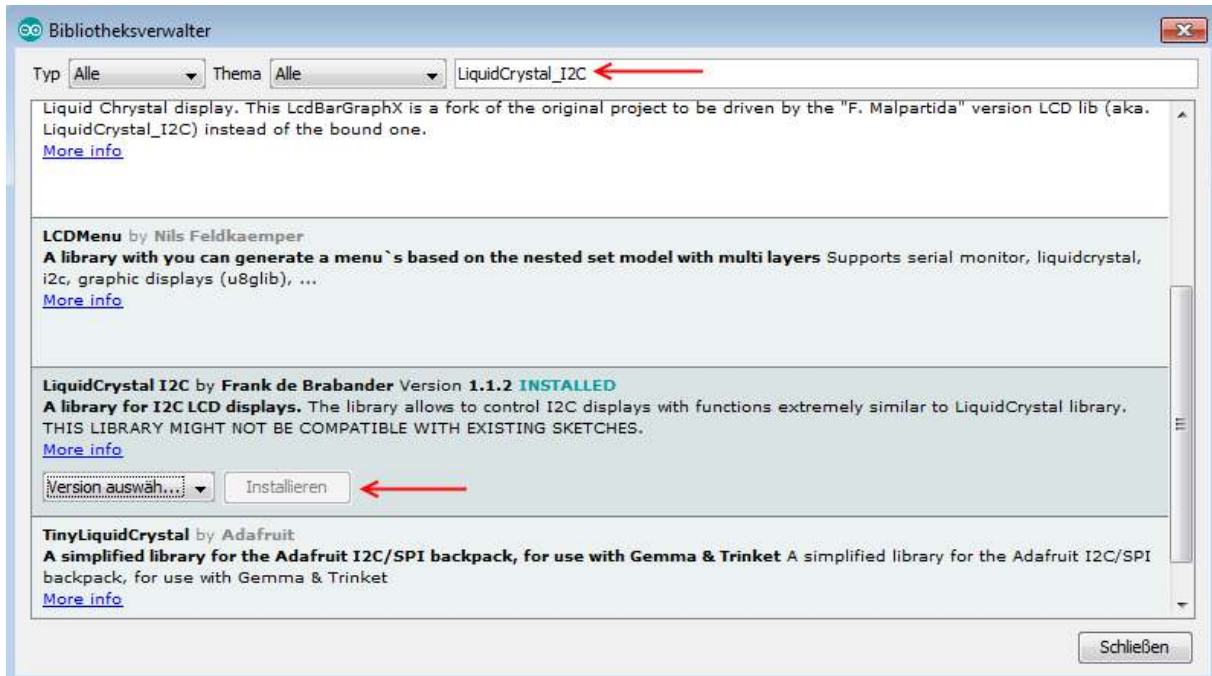


Bild 4: Installation der LiquidCrystal_I2C Bibliothek

Schritt 3 – Installation des Displays

Im Nachfolgenden können Sie ein Codebeispiel zur Verwendung des Displays entnehmen. Kopieren Sie dies vollständig auf Ihren Arduino.

Da es unterschiedliche Hardware-Adressierungen der Displays gibt, muss auch im Code die richtige Adressierung verwendet werden.

Sollte Ihr Display beim ersten Versuch nicht die gewünschte Aktion durchführen, so tauschen Sie bitte die Adressierung im dafür vorgesehenen Bereich aus.

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

//-----Hardware Adressierung-----
//Bei falscher Funktion bitte obere Zeile auskommentieren,
//und untere Zeile freigeben
LiquidCrystal_I2C lcd(0x27,20,4);
//LiquidCrystal_I2C lcd(0x3F,20,4);

void setup()
{
  lcd.init();
}

void loop()
{
  lcd.backlight();

  //Nachricht ausgeben
  lcd.setCursor(0,0);
  lcd.print(" joy-IT");
  lcd.setCursor(0,1);
  lcd.print(" I2C Serial LCD");
}
```

Code 1: Arduino Quellcode

Verwendung mit einem Raspberry Pi

Schritt 1 – Anschließen des Display

Schließen Sie das Display, wie im folgenden Bild 5, bzw. in der folgenden Tabelle 2, zu sehen, an die PINs des Raspberry Pis an.

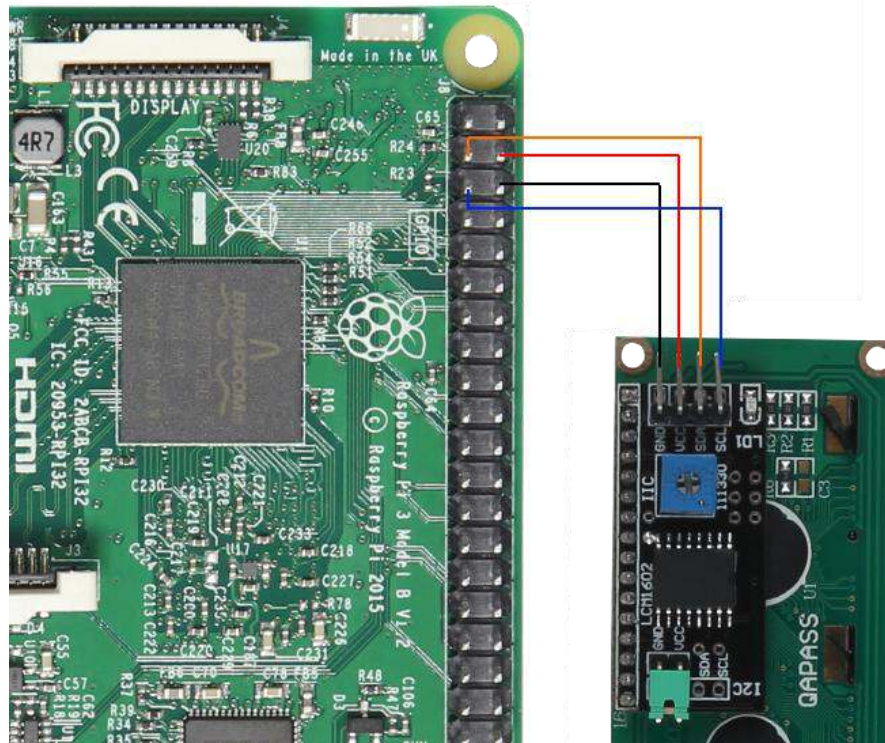


Bild 5: Verkabelung mit Raspberry Pi

Raspberry Pi PIN	I2C LCD PIN
PIN 3 (BCM 2 / SDA)	SDA
PIN 4 (5v Power)	VCC
PIN 5 (BCM 3 / SCL)	SCL
PIN 6 (Ground)	GND

Tabelle 2: PIN-Verbindung zwischen Raspberry und Serial-LCD

Schritt 2 – Installation der Software

Sollten Sie bereits ein aktuelles Raspbian-System auf Ihrem Raspberry verwenden, so können Sie diesen Schritt überspringen und sofort mit Schritt 3 fortfahren.

Installieren Sie auf Ihre SD-Karte mit Hilfe des „Win32 Disk Imager“-Programms das aktuelle Raspbian Image, welches Sie unter dem folgenden [Link](#) zum Download finden.

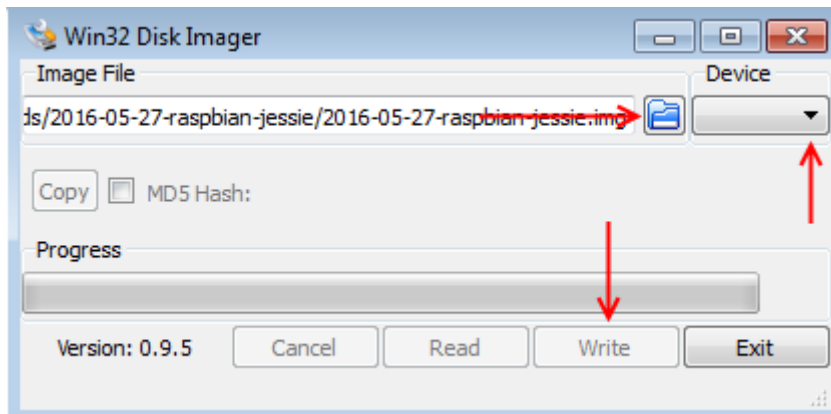


Bild 6: Screenshot des Win32 Disk Imagers

Schritt 3 – Installation der Bibliotheken

Sobald Sie die Installation abgeschlossen und das System gestartet haben, öffnen Sie die Terminal-Konsole und führen Sie folgende Kommandos aus:

```
sudo apt-get install python-pip python-dev build-essential
sudo pip install RPi.GPIO
```

Terminal 1: Installation der GPIO Bibliothek

```
sudo apt-get install python-imaging
```

Terminal 2: Installation der Python Bibliothek

```
sudo apt-get install python-smbus i2c-tools
```

Terminal 3: Installation der I2C-Tools

Sollte I2C nicht bereits freigeschaltet sein, so muss dies nun in den Einstellungen nachgeholt werden.

`sudo raspi-config`

Terminal 4: Raspberry Pi Konfigurationsmenü

In dem sich nun öffnenden Fenster wählen Sie die Option **Advanced Options**.

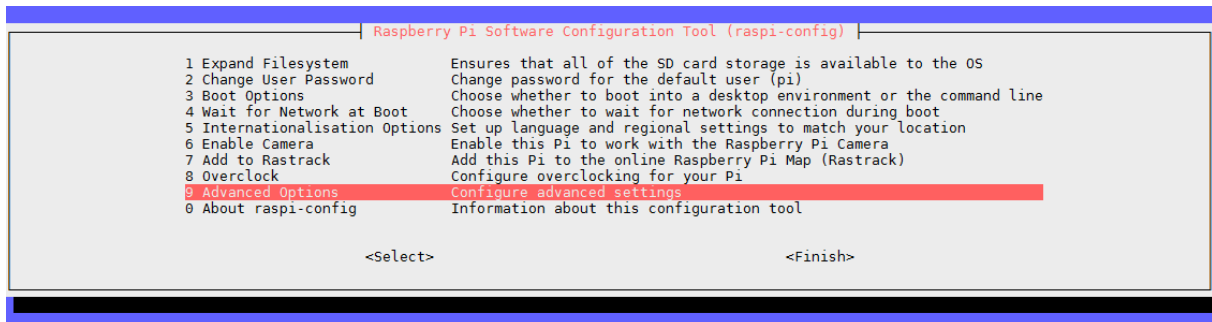


Bild 7: Raspi-Config

Hier wählen und aktivieren Sie die Option **I2C**.

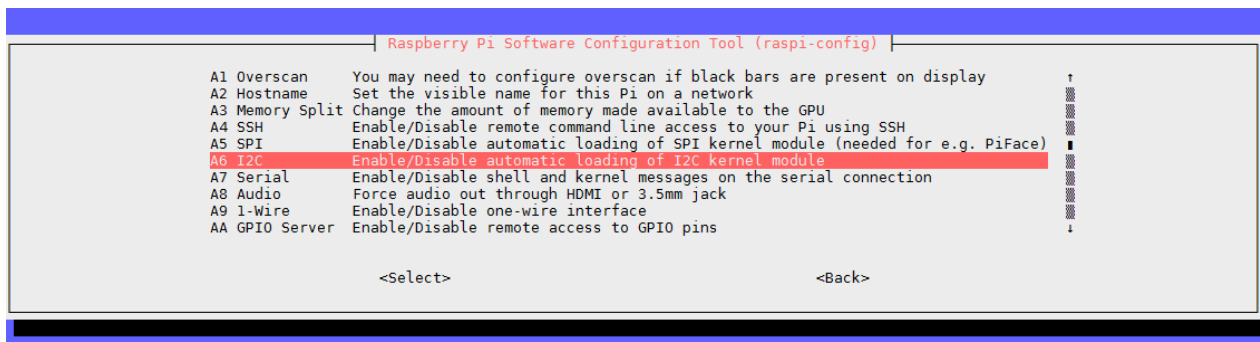


Bild 8: I2C Option

Anschließend müssen noch Einträge zur Modul-Datei hinzugefügt werden.

Öffnen Sie dazu die Modul-Datei:

`sudo nano /etc/modules`

Terminal 5: Bearbeiten der Modul-Datei

Fügen Sie ans Ende dieser Datei folgende Zeilen:

```
i2c-bcm2708
i2c-dev
```

Code 2: Anhängen an die Modul-Datei

Speichern Sie Ihre Änderungen mit **Strg+O** und verlassen Sie den Editor mit **Strg+X**.
Starten Sie Ihren Raspberry Pi nun neu.

```
sudo reboot
```

Terminal 6: Neustart

Um eine Möglichst einfache Anwendung zu ermöglichen, wird im Folgenden auf einen Code von **Tutorials-Raspberrypi** zurückgegriffen.

Dieses können Sie auch [hier](#) abrufen.

Zunächst laden wir den Code herunter, der es uns ermöglicht, mit dem Display zu kommunizieren.

```
mkdir I2C-LCD && cd I2C-LCD  
wget http://tutorials-raspberrypi.de/wp-content/uploads/scripts/hd44780_i2c.zip  
unzip hd44780_i2c.zip
```

Terminal 7: I2C-LCD Code herunterladen

Als nächstes muss das Display, insofern es bereits angeschlossen wurde, angesprochen werden.

```
sudo i2cdetect -y 1
```

Terminal 8: Erkennung des Displays

Die Ausgabe sollte folgendes Anzeigen:

```
    0  1  2  3  4  5  6  7  8  9  a  b  c  d  e  f  
00:  --  --  --  --  --  --  --  --  --  --  --  --  --  --  --  
10:  --  --  --  --  --  --  --  --  --  --  --  --  --  --  --  
20:  --  --  --  --  --  --  --  27  --  --  --  --  --  --  --  
30:  --  --  --  --  --  --  --  --  --  --  --  --  --  --  --  
40:  --  --  --  --  --  --  --  --  --  --  --  --  --  --  --  
50:  --  --  --  --  --  --  --  --  --  --  --  --  --  --  --  
60:  --  --  --  --  --  --  --  --  --  --  --  --  --  --  --  
70:  --  --  --  --  --  --  --  --  --  --  --  --  --  --  --
```

Ausgabe 1: I2C-Prüfung

Sollten Sie hier eine andere Zahl als **27** angezeigt bekommen, so muss dies noch in den Einstellungen angepasst werden.

Passen Sie daher die Zahl in der Zeile **ADDRESS = 0x27** an.

```
sudo nano lcddriver.py
```

Terminal 9: Bearbeiten der Treibereinstellungen

Schritt 4 – Kommunikation mit dem Display

Abschließend können Sie nun eine neue Datei erstellen, um mit dem Display zu kommunizieren.

```
sudo nano LCD.py
```

Terminal 10: Erstellen einer neuen Datei

Geben Sie nun folgenden Code ein.

Die Datei können Sie danach mit **Strg+O** speichern und mit **Strg+X** verlassen.

```
import lcddriver
from time import *

lcd = lcddriver.lcd()
lcd.lcd_clear()

lcd.lcd_display_string(" joy-IT", 1)
lcd.lcd_display_string(" I2C Serial LCD", 2)
```

Code 3: Kommunikation mit dem Display

Den Text innerhalb des **lcd.lcd_display_string** können Sie nach Ihren Vorstellungen anpassen. Haben Sie Ihre Datei gespeichert und verlassen, kann der Text nun auf das Display übertragen werden.

```
sudo python LCD.py
```

Terminal 11: Ausführen des Codes