

MIKROKONTROLLER & I²C BUS

by AS

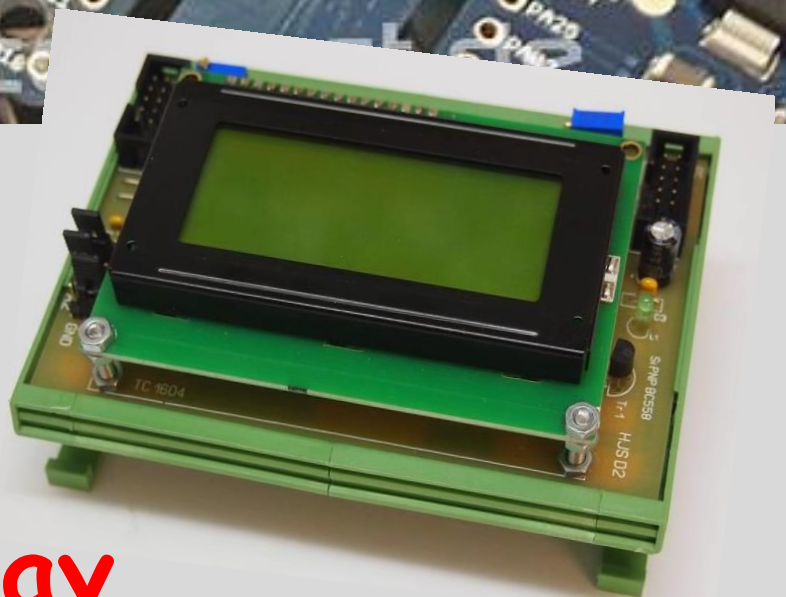
www.makerconnect.de

<https://www.makerconnect.de/resource>

makerconnect

I2C Bus und
LCD Display
Hard- und Software

LCD - Display



Copyright

Sofern nicht anders angegeben, stehen die Inhalte dieser Dokumentation unter einer „Creative Commons - Namensnennung-NichtKommerziell-Weitergabe unter gleichen Bedingungen 3.0 DE Lizenz“



Sicherheitshinweise

Lesen Sie diese Gebrauchsanleitung, bevor Sie diesen Bausatz in Betrieb nehmen und bewahren Sie diese an einem für alle Benutzer jederzeit zugänglichen Platz auf. Bei Schäden, die durch Nichtbeachtung dieser Bedienungsanleitung verursacht werden, erlischt die Gewährleistung/Garantie. Für Folgeschäden übernehmen wir keine Haftung! Bei allen Geräten, die zu ihrem Betrieb eine elektrische Spannung benötigen, müssen die gültigen VDE-Vorschriften beachtet werden. Besonders relevant sind für diesen Bausatz die VDE-Richtlinien VDE 0100, VDE 0550/0551, VDE 0700, VDE 0711 und VDE 0860. Bitte beachten Sie auch nachfolgende Sicherheitshinweise:

- Nehmen Sie diesen Bausatz nur dann in Betrieb, wenn er zuvor berührungssicher in ein Gehäuse eingebaut wurde. Erst danach darf dieser an eine Spannungsversorgung angeschlossen werden.
- Lassen Sie Geräte, die mit einer Versorgungsspannung größer als 24 V- betrieben werden, nur durch eine fachkundige Person anschließen.
- In Schulen, Ausbildungseinrichtungen, Hobby- und Selbsthilfewerkstätten ist das Betreiben dieser Baugruppe durch geschultes Personal verantwortlich zu überwachen.
- In einer Umgebung in der brennbare Gase, Dämpfe oder Stäube vorhanden sind oder vorhanden sein können, darf diese Baugruppe nicht betrieben werden.
- Im Falle einer Reparatur dieser Baugruppe, dürfen nur Original-Ersatzteile verwendet werden! Die Verwendung abweichender Ersatzteile kann zu ernsthaften Sach- und Personenschäden führen. Eine Reparatur des Gerätes darf nur von fachkundigen Personen durchgeführt werden.
- Spannungsführende Teile an dieser Baugruppe dürfen nur dann berührt werden (gilt auch für Werkzeuge, Messinstrumente o.ä.), wenn sichergestellt ist, dass die Baugruppe von der Versorgungsspannung getrennt wurde und elektrische Ladungen, die in den in der Baugruppe befindlichen Bauteilen gespeichert sind, vorher entladen wurden.
- Sind Messungen bei geöffnetem Gehäuse unumgänglich, muss ein Trenntrafo zur Spannungsversorgung verwendet werden
- Spannungsführende Kabel oder Leitungen, mit denen die Baugruppe verbunden ist, müssen immer auf Isolationsfehler oder Bruchstellen kontrolliert werden. Bei einem Fehlers muss das Gerät unverzüglich ausser Betrieb genommen werden, bis die defekte Leitung ausgewechselt worden ist.
- Es ist auf die genaue Einhaltung der genannten Kenndaten der Baugruppe und der in der Baugruppe verwendeten Bauteile zu achten. Gehen diese aus der beiliegenden Beschreibung nicht hervor, so ist eine fachkundige Person hinzuzuziehen

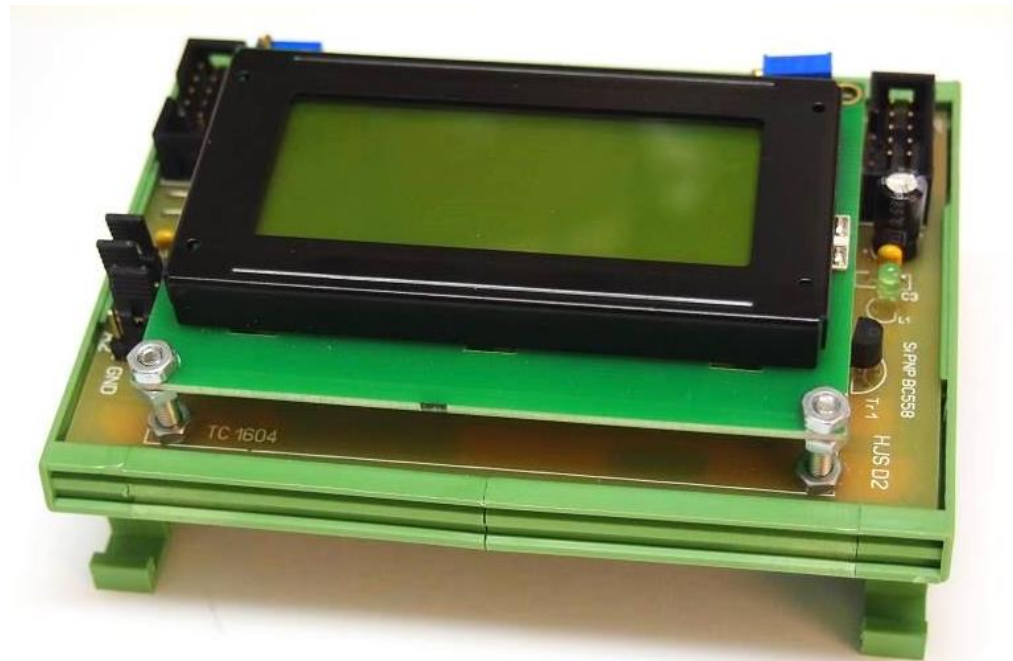
Bestimmungsgemäße Verwendung

- Auf keinen Fall darf 230 V~ Netzspannung angeschlossen werden. Es besteht dann Lebensgefahr!
- Dieser Bausatz ist nur zum Einsatz unter Lern- und Laborbedingungen konzipiert worden. Er ist nicht geeignet, reale Steuerungsaufgaben jeglicher Art zu übernehmen. Ein anderer Einsatz als angegeben ist nicht zulässig!
- Der Bausatz ist nur für den Gebrauch in trockenen und sauberen Räumen bestimmt.
- Wird dieser Bausatz nicht bestimmungsgemäß eingesetzt kann er beschädigt werden, was mit Gefahren, wie z.B. Kurzschluss, Brand, elektrischer Schlag etc. verbunden ist. Der Bausatz darf nicht geändert bzw. umgebaut werden!
- Für alle Personen- und Sachschäden, die aus nicht bestimmungsgemäßer Verwendung entstehen, ist nicht der Hersteller, sondern der Betreiber verantwortlich. Bitte beachten Sie, dass Bedien- und /oder Anschlussfehler außerhalb unseres Einflussbereiches liegen. Verständlicherweise können wir für Schäden, die daraus entstehen, keinerlei Haftung übernehmen.
- Der Autor dieses Tutorials übernimmt keine Haftung für Schäden. Die Nutzung der Hard- und Software erfolgt auf eigenes Risiko.

Display 1

Heute möchte ich euch ein weiteres Modul vorstellen. Es hat den schönen Namen Display 1. Damit haben wir auch schon seine Funktion.

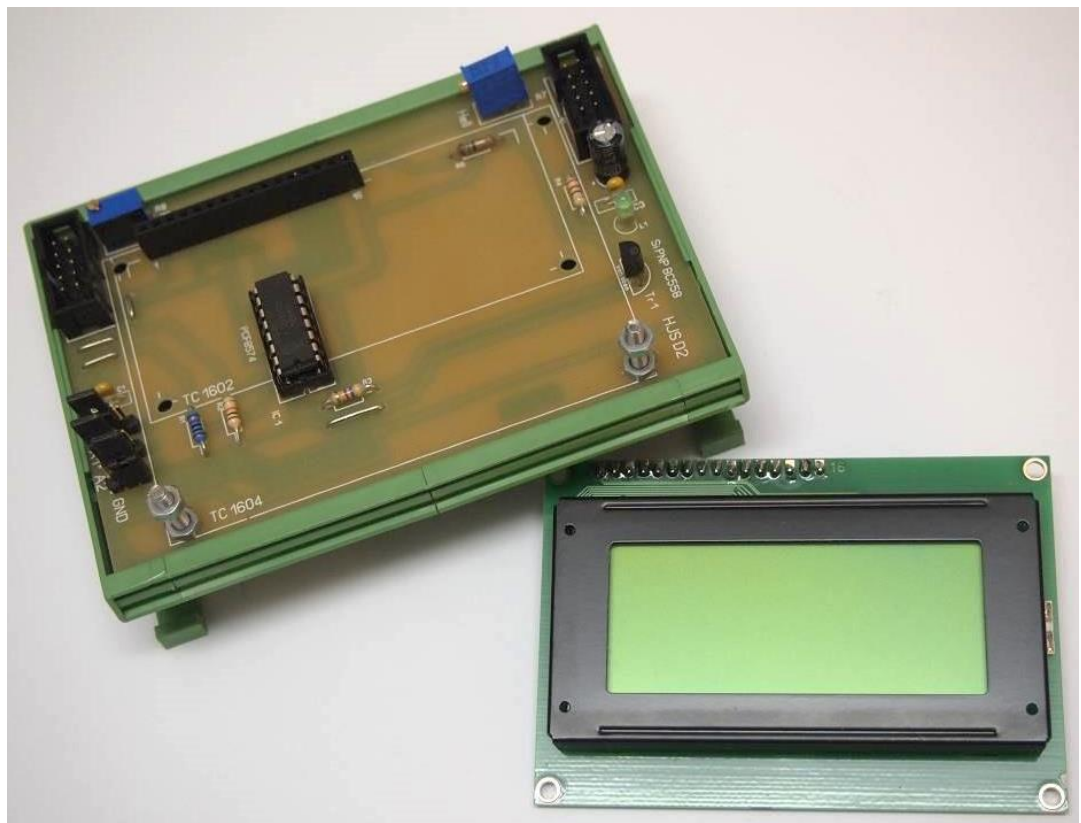
Display 1



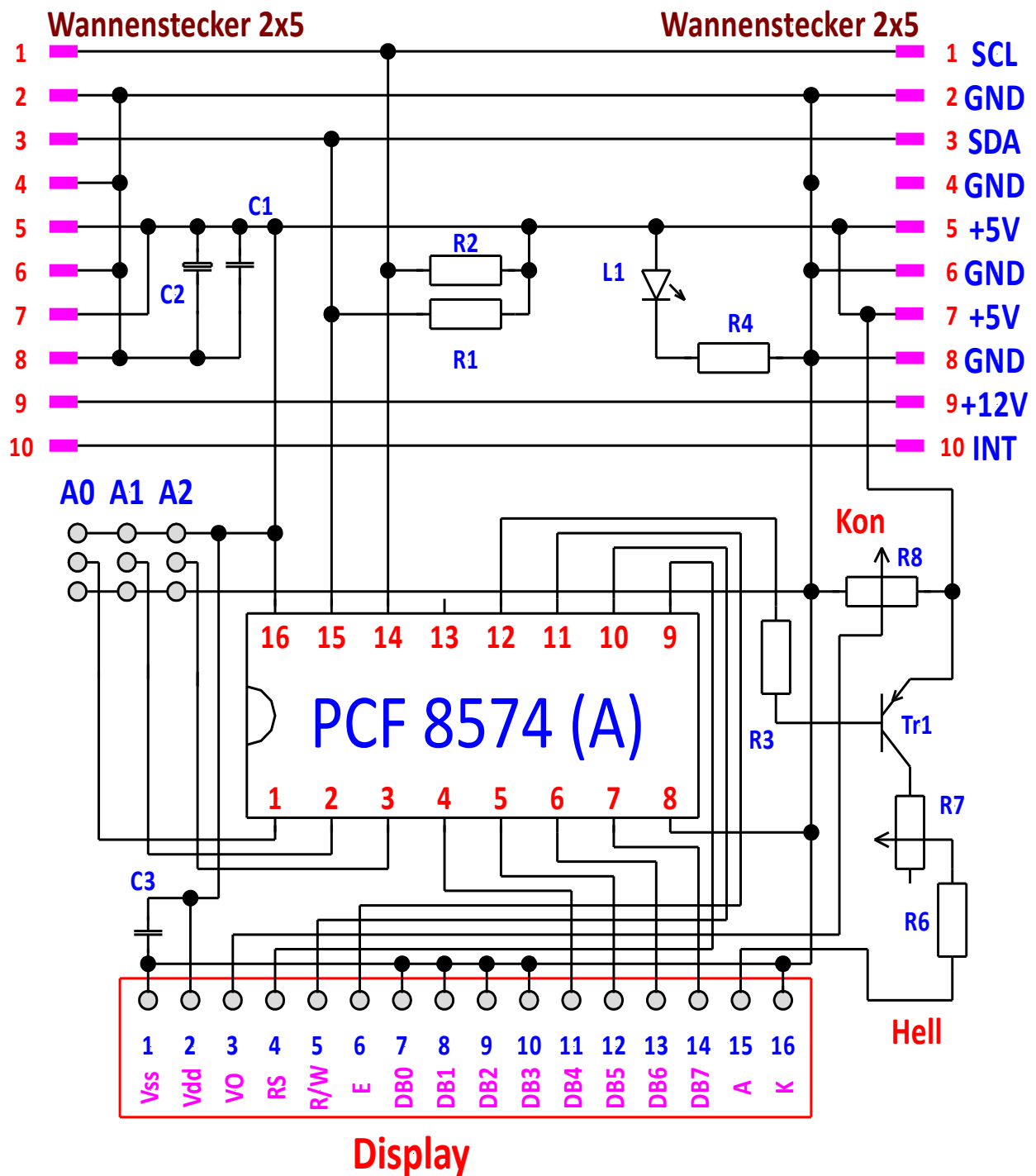
Es ist in meinem System ein Anzeige Modul und wird durch den I²C Bus angesteuert.

Eigentlich sind es zwei Platinen (Module). Es besteht zu einem aus einer Grundplatine mit der notwendigen Elektronik und einem LCD Display (4x16).

Grundplatte und LCD Display



Sehen wir uns als nächste einmal das Schaltbild an:



Schaltung Display 1 (Anschluss im 4 Bit Modus)

Diese Schaltung wird auch von vielen anderen Herstellern verwendet. Ich habe eigentlich nur die Montage verändert / angepasst. Je nach Hersteller oder Typ kann der Regler R7 durch eine Brücke ersetzt werden.

Funktionsbeschreibung:

Im oberen Teil des Schaltbildes ist der Anschluss mit den 2 Wannensteckern zum I²C Bus. Zwischen den beiden Wannensteckern befinden sich die Abblockung der Betriebsspannung und die LED mit Vorwiderstand zur Anzeige der Betriebsspannung. In der Mitte, auf der linken Seite befinden sich die drei Jumper zur Einstellung der Adressen. Daneben befindet sich der PCF8574. Unterhalb des ICs befindet sich unser Display. Auf der rechten Seite befindet sich der Regler R8 und wird zur Einstellung des Kontrastes genutzt. Darunter befindet sich ein Transistor, mit dem die LED (Beleuchtung) eingeschaltet wird. Darunter befindet sich der Regler R7. Mit diesem kann die Helligkeit des Displays eingestellt werden. (Hersteller beachten) Mit den beiden Widerständen R1 und R2 wird der Bus ständig auf Vcc (+5V) gelegt.

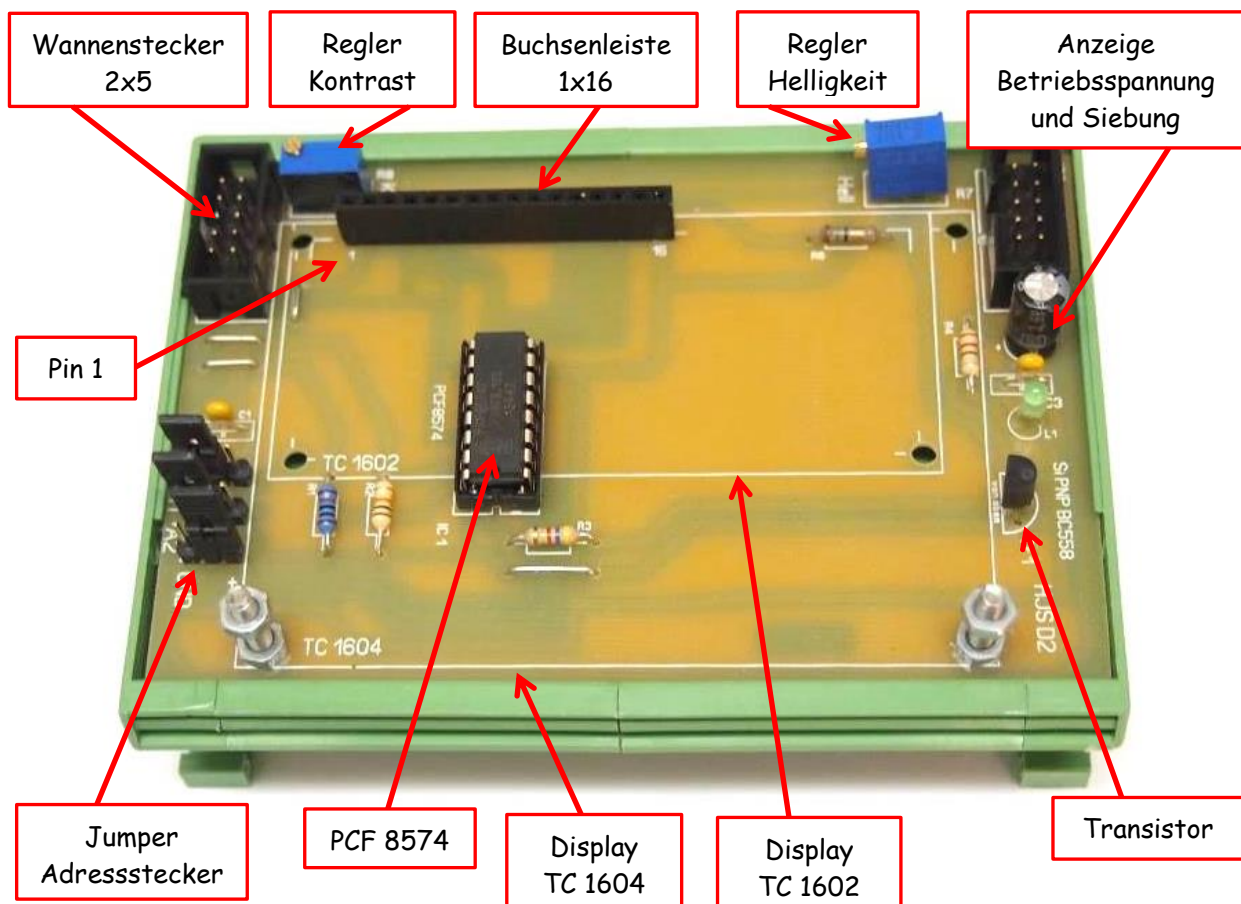
Stückliste:

2 x Wannenstecker 2x5
R1, R2 - Widerstand 10 kOhm
R4 - 220 Ohm
R7 - Regler 100 Ohm
C2 - Elko 100/16
L1 - LED 3 mm grün 20 mA
Steckerleiste 1x16

Platine ca. 72 x 110 mm (D2)
R3 - Widerstand 4,7 kOhm
R6 - Widerstand 22 Ohm
R8 - Regler 10 kOhm
C1, C3 - Kondensator 100nF
Tr1 - Si PNP BC558
Buchsenleiste 1x16

Display - TC 1604 - 4x16 (TC 1602 2x16)
Widerstand R5 habe ich leider vergessen.

Sehen wir uns die Platine einmal genauer an:



Es gibt müssen noch ein paar Sachen zu beachtet.

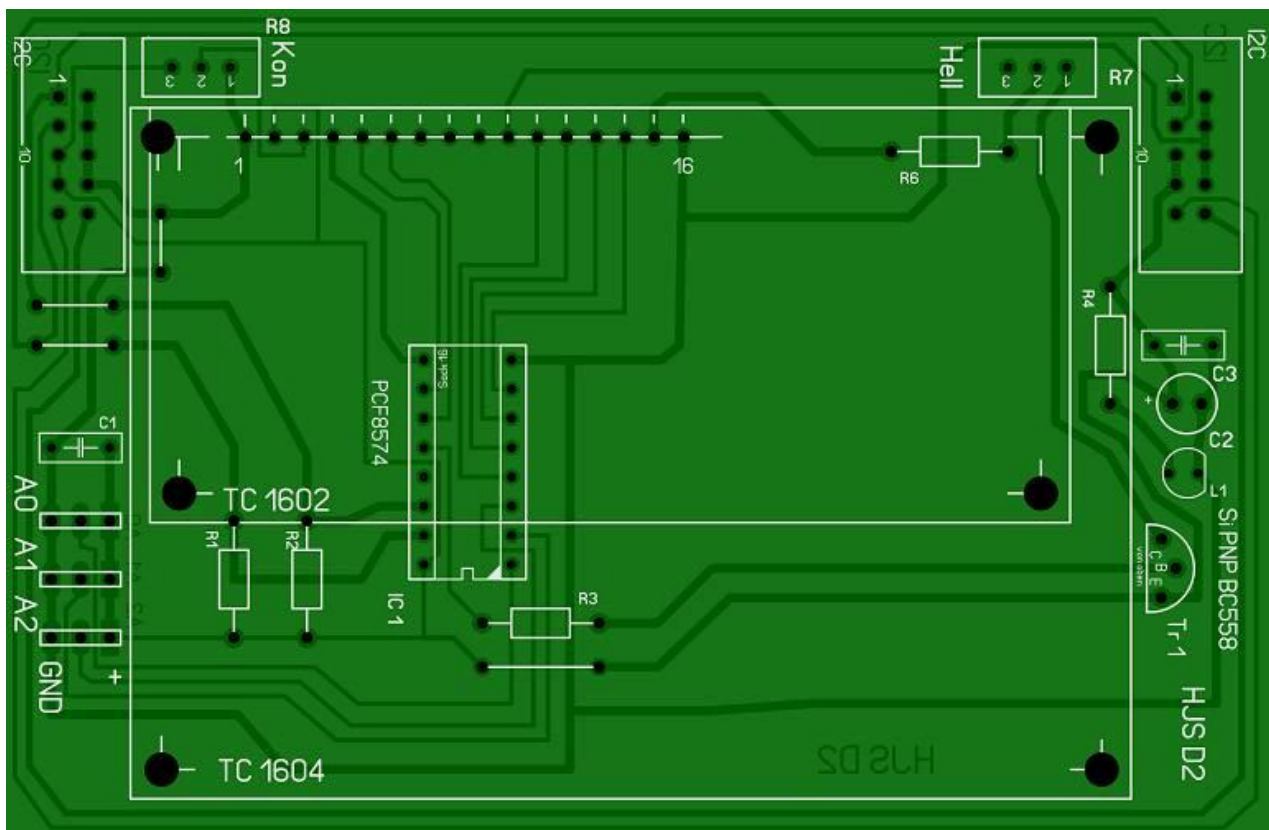
Auf der Platine befinden sich die Angaben für 2 verschiedenen Displays. Da beide Displays die gleichen Anschlüsse haben, können auch wahlweise beide verwendet werden. Das muss allerdings in der Software beachtet werden.

Der Widerstand R6
(Vorwiderstand LED-
Display) kann je nach
verwendetem Display
Typ unterschiedlich
sein. Er kann in einem
bestimmten Bereich
variieren. Eventuell
muss er verändert
werden.

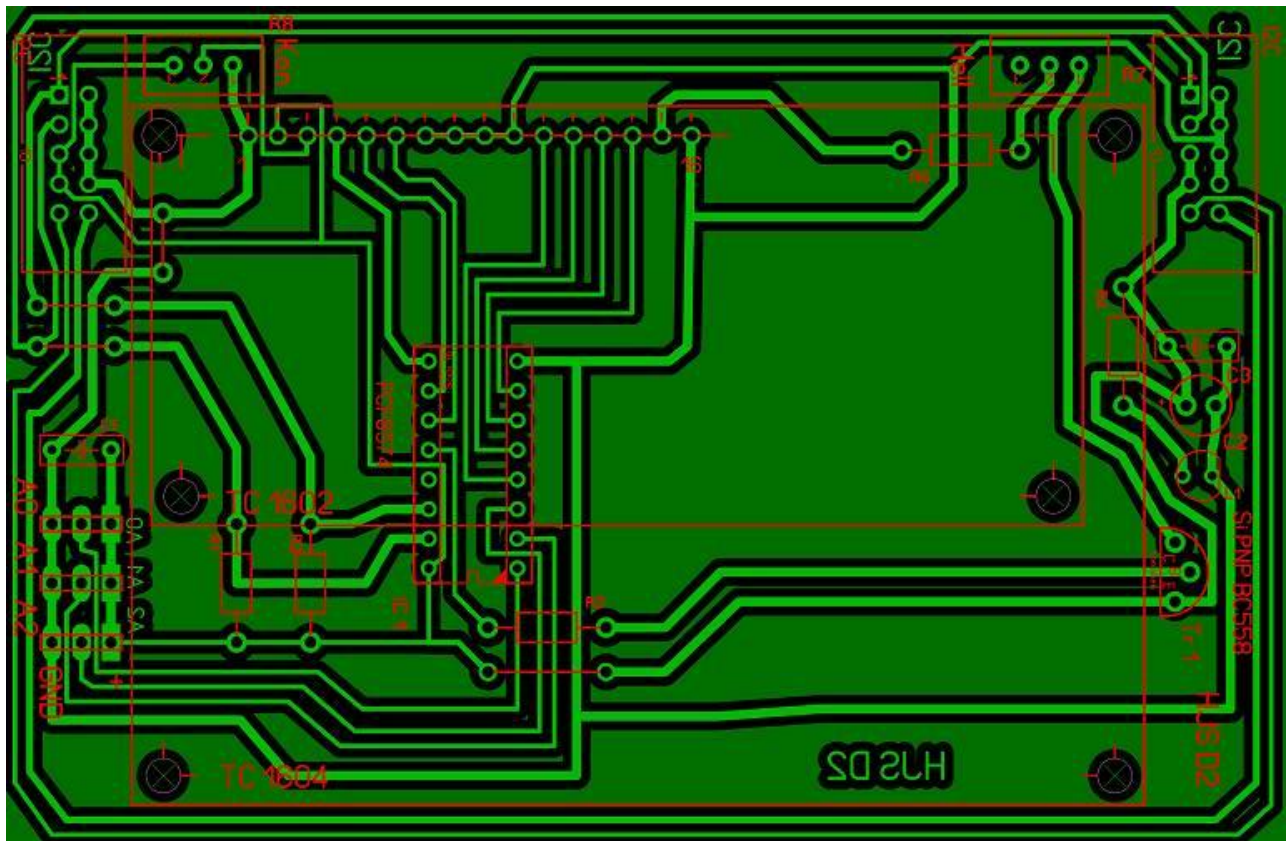
Ansicht Display von hinten



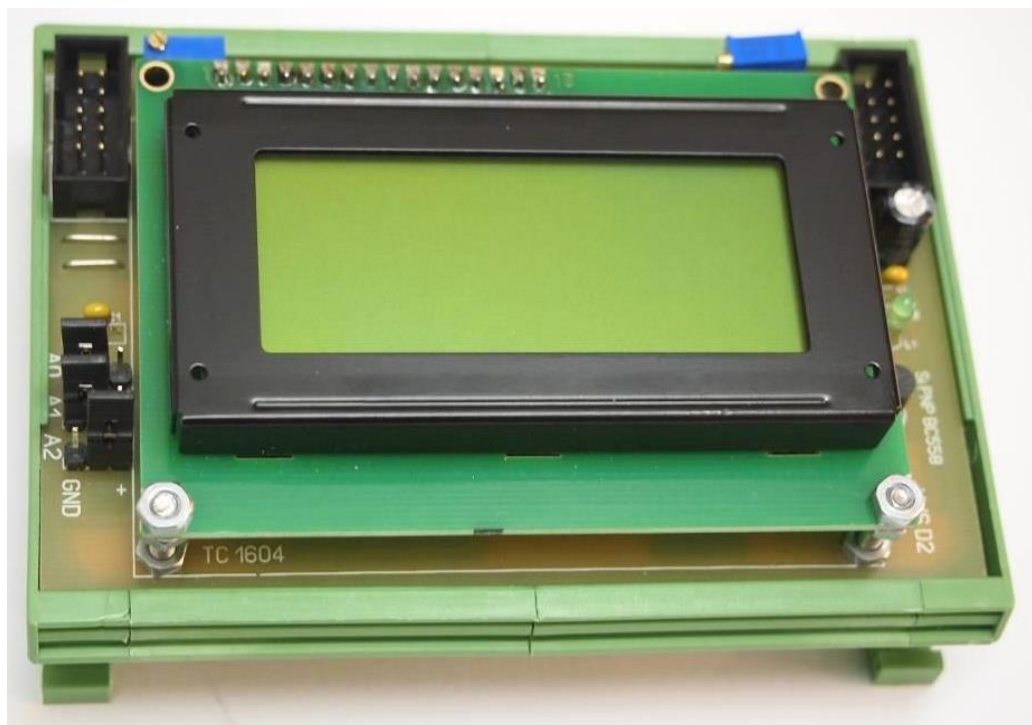
Auf diesem Bild seht ihr das Display von der Rück-seite. Hier habe ich die Stecker-leiste eingelötet. Dadurch kann ich auf der Grundplatine keinen Kurzschluss verursachen.



Platine in der Fotoansicht mit Bestückungsdruck



Platine in der Durchsicht



Wenn man alles richtig montiert hat, könnte das fertige Modul mit Halterung so aussehen. Das Display habe ich mit 2 Schrauben M3 befestigt. Bitte keine Gewalt anwenden. Ich habe wieder alle Bauteile beschriftet und die notwendigen Brücken eingezeichnet. Leider sind einige Brücken notwendig, da ich mit einer einseitigen Platine auskommen wollte.

Die Auswahl der Adressen erfolgt wieder durch die Jumper A0, A1 und A2.

Möglich Adressen für
den PCF 8574 (A)

	A0	A1	A2	PCF 8574		PCF 8574 A	
				Schreiben	Lesen	Schreiben	Lesen
+							
GND				0x40	- 0x41	0x70	- 0x71
+							
GND				0x42	- 0x43	0x72	- 0x73
+							
GND				0x44	- 0x45	0x74	- 0x75
+							
GND				0x46	- 0x47	0x76	- 0x77
+							
GND				0x48	- 0x49	0x78	- 0x79
+							
GND				0x4a	- 0x4b	0x7a	- 0x7b
+							
GND				0x4c	- 0x4d	0x7c	- 0x7d
+							
GND				0x4e	- 0x4f	0x7e	- 0x7f

Der PCF 8574 wird in
zwei verschiedenen
Versionen produziert.
Einmal als PCF 8574
und PCF 8574 A.
Unterschiedliche

Bauarten (DIP oder SOP) nicht berücksichtigt. Durch die Jumper A0, A1 und A2 können unterschiedliche Adressen ausgewählt werden. Dabei stehen die Adressräume von 0x40 bis 0x4e für den PCF 8574 und 0x70 bis 0x7e für den PCF 8574A zur Verfügung. Dadurch können bis zu 16 Schaltkreise an einen Bus angeschlossen werden.

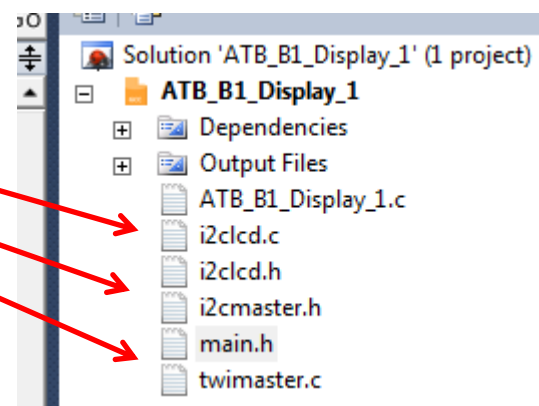
In der Tabelle habe ich die einzelnen Adressen angegeben. Es besteht ein Unterschied zwischen lesen und schreiben, so ist die 0x40 die Schreibadresse und 0x41 die Leseadresse

Wir müssen wieder mehrere Dateien einbinden und einige Inhalte überprüfen.

Es ist `i2clcd.c`, `i2clcd.h`, `i2cmaster.h`, `main.h` und `twimaster.c`.

Einbinden bzw.
ändern/kontrollieren

Wie diese Dateien eingebunden werden,
habe ich bereits in einem anderen Teil
beschrieben.



In der Datei **main.h** speichern wir alle Angaben die für die Ausführung aller Programme notwendig sind.

Inhalt der Datei main.h

```
#define F_CPU 16000000 // CPU clock in Hz
#define adr1_w 0x40 // Anzeige Display
```

(Angabe der Quarzfrequenz - 16MHz und der Adresse 1 - 0x40)

In der Datei **twimaster.c** müssen wir diese Einstellungen kontrollieren bzw. ändern

```
#include <inttypes.h>
#include <compat/twi.h>
#include "i2cmaster.h"
#include "main.h"
```

In der Datei **i2clcd.h** müssen wir unserem Programm mitteilen, welches Display wir verwenden:

```
#define LCD_I2C_DEVICE adr1_w /**< 0x40 Change this to the address of your expander */
#define LCD_LINES 4 /**< Enter the number of lines of your display here */
#define LCD_COLS 16 /**< Enter the number of columns of your display here */
#define LCD_LINE_MODE LCD_2LINE /**< Enter line mode your display here */

#define LCD_LINE1 0x00 /**< This should be 0x00 on all displays */
#define LCD_LINE2 0x40 /**< Change this to the address for line 2 on your display */
#define LCD_LINE3 0x10 /**< Change this to the address for line 3 on your display */
#define LCD_LINE4 0x50 /**< Change this to the address for line 4 on your display */
```

- # define LCD_I2C_Devices - Angabe der Adresse - **adr1_w**
- # define LCD_I2C_Devices - Angabe der möglichen Zeilen - **4**
- # define LCD_I2C_Devices - Angabe der möglichen Zeichen - **16**
- # define LCD_I2C_Devices - Beginn der Speicheradresse Zeile 1 - **0x00**
- # define LCD_I2C_Devices - Beginn der Speicheradresse Zeile 2 - **0x40**
- # define LCD_I2C_Devices - Beginn der Speicheradresse Zeile 3 - **0x10**
- # define LCD_I2C_Devices - Beginn der Speicheradresse Zeile 4 - **0x50**

```
/* LCD Pin Connections */
#define LCD_D4_PIN 0 /**< 4LCD-Pin D4 is connected to P4 on the PCF8574 */
#define LCD_D5_PIN 1 /**< 5LCD-Pin D5 is connected to P5 on the PCF8574 */
#define LCD_D6_PIN 2 /**< 6LCD-Pin D6 is connected to P6 on the PCF8574 */
#define LCD_D7_PIN 3 /**< 7LCD-Pin D7 is connected to P7 on the PCF8574 */
#define LCD_RS_PIN 4 /**< 0LCD-Pin RS is connected to P0 on the PCF8574 */
#define LCD_RW_PIN 5 /**< 1LCD-Pin RW is connected to P1 on the PCF8574 */
#define LCD_E_PIN 6 /**< 2LCD-Pin E is connected to P2 on the PCF8574 */
#define LCD_LIGHT_PIN 7 /**< 3LCD backlight is connected to P3 on the PCF8574, low active */
```

An dieser Stelle teile ich dem Programm mit, an welchem Pin des PCF8574 welcher Pin vom Display angeschlossen ist.

Verwende ich ein anderes Display, muss ich die entsprechenden Angaben ändern.

Die Dateien, die Bestandteil dieses Tut`s sind, könnten die notwendigen Änderungen bereits enthalten. Trotzdem muss alles kontrolliert werden.

In einem anderen Tutorial habe ich genau beschrieben, wie notwendige Programme eingebunden werden. Dazu ist es notwendig diese Dateien in einem extra Ordner, z.B. mit dem Namen „Dateien“, abzuspeichern. Ich werde diese Dateien zusammen mit den Programmen zur Verfügung stellen. Nach dem ich diese Änderungen durchgeführt habe, dürfte es keine Problem geben das Programm zu starten und auszuführen.

Für den ersten Versuch und zum Testen habe ich wieder ein Programm geschrieben.

```
/* ATB_B1_Display_1.c Created: 11.10.2014 16:14:43 Author: AS */
```

```
#include <stdbool.h>
#include <avr/pgmspace.h>
#include "main.h"           // Aufruf „main“
#include "i2clcd.h"
#include "i2cmaster.h"
#include "avr/io.h"
#include <util/delay.h>

int main(void)
{
    i2c_init();              // Starte I2C Bus
    lcd_init();              // Starte I2CLCD
    lcd_light(0);            // 0=Licht an, 1=Licht aus

    // Display Befehle
    lcd_command(LCD_DISPLAYON | LCD_CURSOROFF | LCD_BLINKINGOFF);
    // Display ON/OFF / Cursor ON/OFF / Blinken ON/OFF

    lcd_command(LCD_CLEAR);  // Leere Display
    _delay_ms(2);            // warte 2ms

    lcd_printlc(1,6,"Boxtec"); // Text Zeile 1
    lcd_printlc(2,2,"*****"); // Text Zeile 2
    lcd_printlc(3,2,"Display Modul 1"); // Text Zeile 3
    lcd_printlc(4,2,"(by achim S.)"); // Text Zeile 4
    _delay_ms(5000);         // Warte 5000ms

    while(1)                 // while Schleife
    {                         // Beginn Schleife
        lcd_command(LCD_CLEAR); // Leere Display
        _delay_ms(1000);        // Warte 1s

        lcd_printlc(1,1,"Die 1. Zeile"); // Text Zeile 1
        _delay_ms(2000);        // warte 2s

        lcd_printlc(2,2,"Die 2. Zeile"); // Text Zeile 2
        _delay_ms(2000);        // warte 2s

        lcd_printlc(3,3,"Die 3. Zeile"); // Text Zeile 3
        _delay_ms(2000);        // warte 2s

        lcd_printlc(4,4,"Die 4. Zeile"); // Text Zeile 4
        _delay_ms(5000);        // warte 5s
    }                         // Ende Schleife
}
```

Hinter fast jeder Zeile habe ich Kommentare geschrieben. Dadurch kann man relativ leicht die Funktion nachvollziehen. Einiges möchte ich nochmal kurz erläutern.

```
lcd_light(0); // 0=Licht an, 1=Licht aus
```

Damit kann ich die LED am Display ein- oder ausschalten. Achtung: Logik verdreht. Bei einem anderen Typ kann es anders erfolgen. Die Einstellung der Helligkeit muss mit dem entsprechenden Regler erfolgen

```
// Display Befehle  
lcd_command(LCD_DISPLAYON | LCD_CURSOROFF | LCD_BLINKINGOFF);  
// Display ON/OFF / Cursor ON/OFF / Blinken ON/OFF
```

Hier werden Befehle direkt an das Display gesandt. Funktion habe ich mit angegeben

```
lcd_command(LCD_CLEAR); // Leere Display  
_delay_ms(1000); // Warte 1s
```

Damit wird der Inhalt im Display gelöscht. Nach jedem LCD Befehl braucht es eine kurze Pause, sonst kann die Übertragung nicht korrekt ausgeführt werden.

```
lcd_printlc(4,4,"Die 4. Zeile"); // Text Zeile 4
```

Angabe der Zeilen Nummer und Position des Zeichens. Die erste Ziffer ist die Zeilennummer. Dabei liegt der mögliche Bereich von 1 bis 4. Die zweite Ziffer gibt die Position des ersten Zeichens an. Bei der Angabe 4 erscheint das erste Zeichen an der 4 Stelle. Dabei liegt der mögliche Bereich von 1 bis 16. Alle Angaben wieder in Abhängigkeit vom Typ.

Das fertige Display 1
mit Aufschrift an
Anschlusskabel



Einige Teile des Textes wurden zur besseren Übersicht farblich gestaltet.

Die Nutzung erfolgt auf eigenes Risiko.

Ich wünsche viel Spaß beim Bauen und programmieren

Achim

myroboter@web.de