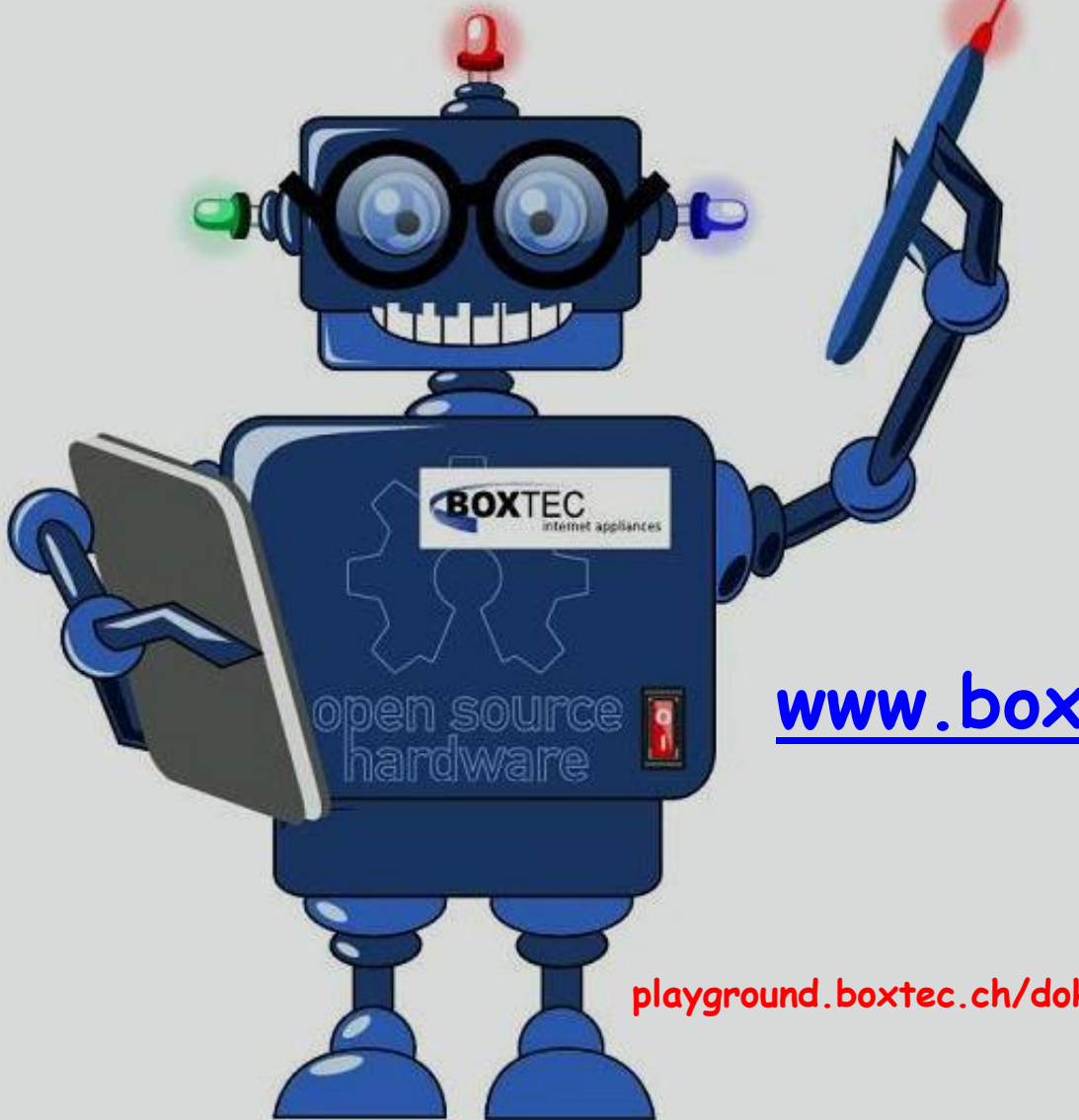


# MIKROKONTROLLER & I<sup>2</sup>C BUS



[www.boxtec.ch](http://www.boxtec.ch)

[playground.boxtec.ch/doku.php/tutorial](http://playground.boxtec.ch/doku.php/tutorial)

I<sup>2</sup>C - Bus und  
Out 2 (16 Bit)



## I<sup>2</sup>C-Bus - Out 2

## 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

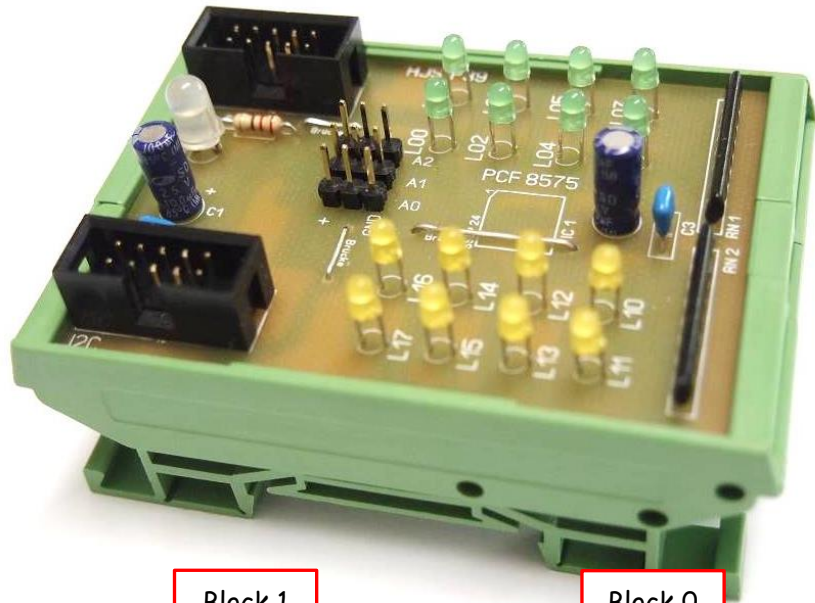
## Bestimmungsgemäße Verwendung

- 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.

## I<sup>2</sup>C Bus - Out 2 (16 Bit)

Im I<sup>2</sup>C - Bus gibt es neben dem PCF 8574 mit seinen 8 Bit auch einen anderen Typ mit 16 Bit. Leider führt er neben seinem bekannten kleinen Bruder eher ein Schattendasein. Es ist der PCF 8575 mit seinen 16 Bit.

Platine mit 16 LED und dem PCF 8575 im System 72



Vom Schaltkreis selber ist eigentlich nicht viel zu sehen.

Da es ein SMD ist, befindet er sich auf der Unterseite der Platine. Ich habe ihn direkt auf die Leiterzüge gelötet. Das ist wahrscheinlich der Hauptgrund für seinen seltenen Einsatz. Der Abstand der einzelnen Pins beträgt ca. 0,6 mm. Das ist sehr klein für eine Handlötung. Habe aber kein Problem damit. Er funktioniert sehr gut. Denn Anblick möchte ich euch lieber ersparen.

Im oberen Bereich befindet sich wieder die Verbindung zum I<sup>2</sup>C - Bus, die Stecker für die Adressen und die LED zur Anzeige der Betriebsspannung. Darunter befinden sich die beiden Blöcke mir jeweils 8 x LED.

An der Unterkante befinden sich 2 Widerstandsnetzwerke mit jeweils 8 x 220 Ohm.

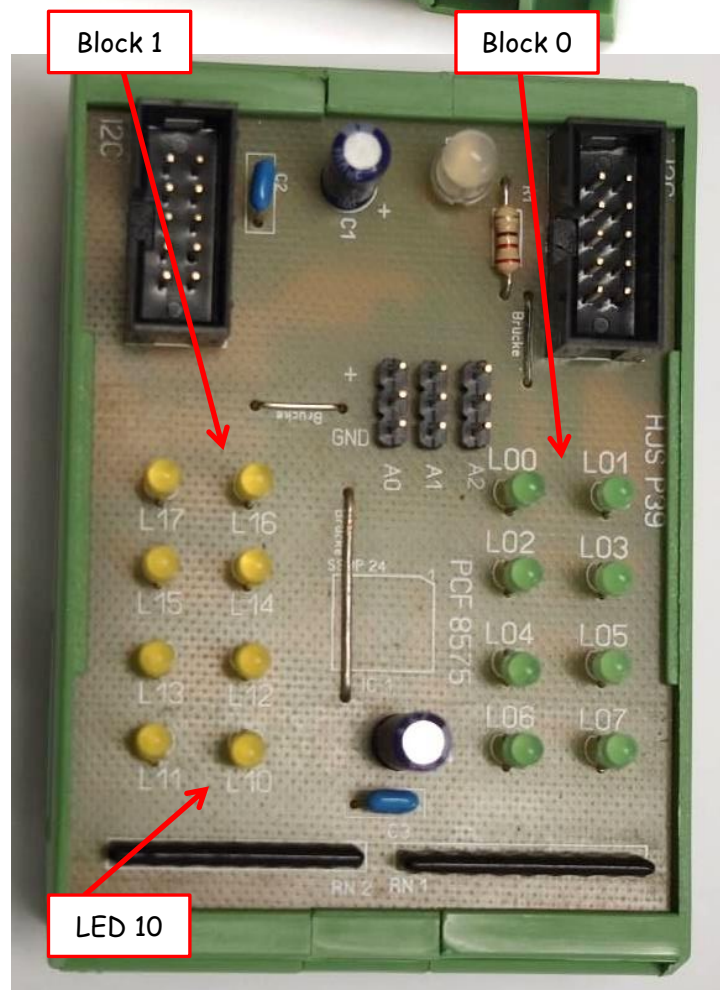
### Achtung!

Die Anordnung der LED ist anders.

Auf der rechten Seite ist der Block 0

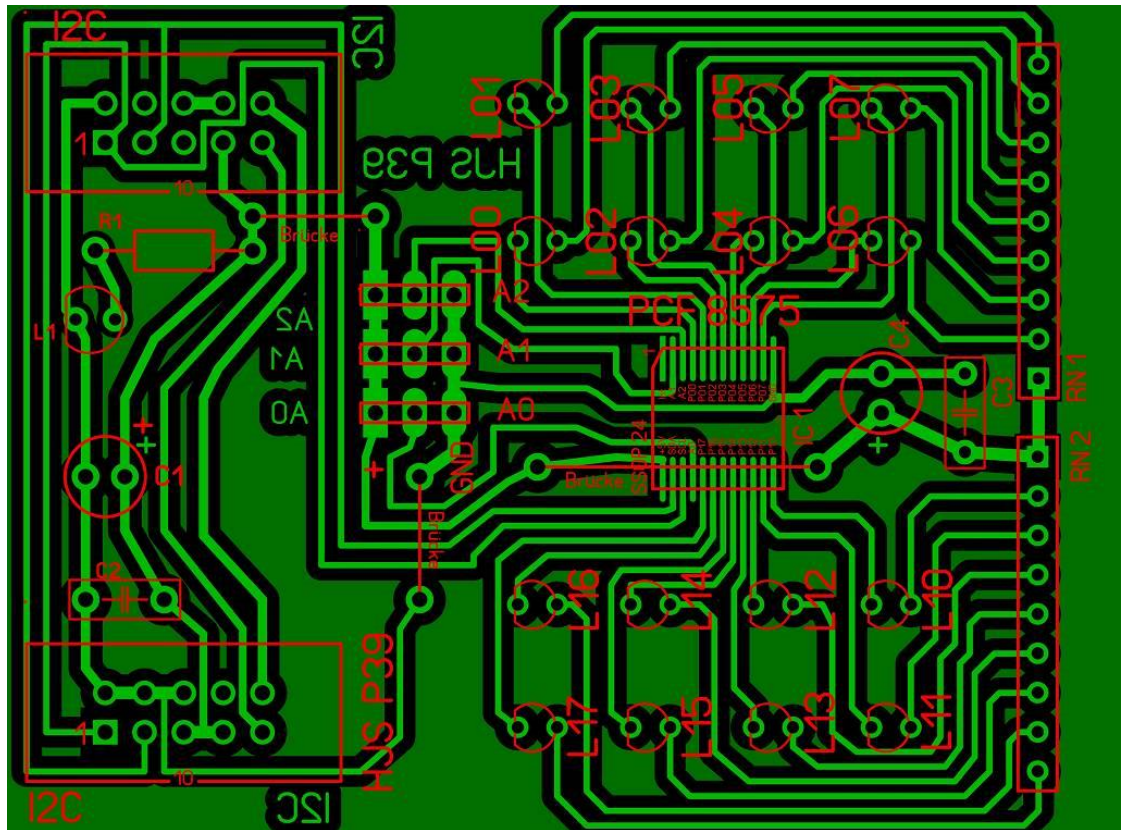
mit den LED 00 bis 07. Auf der linken Seite befindet sich der Block 1 mit den LED 10 bis 17.

**Dieser Block beginnt unten rechts!**

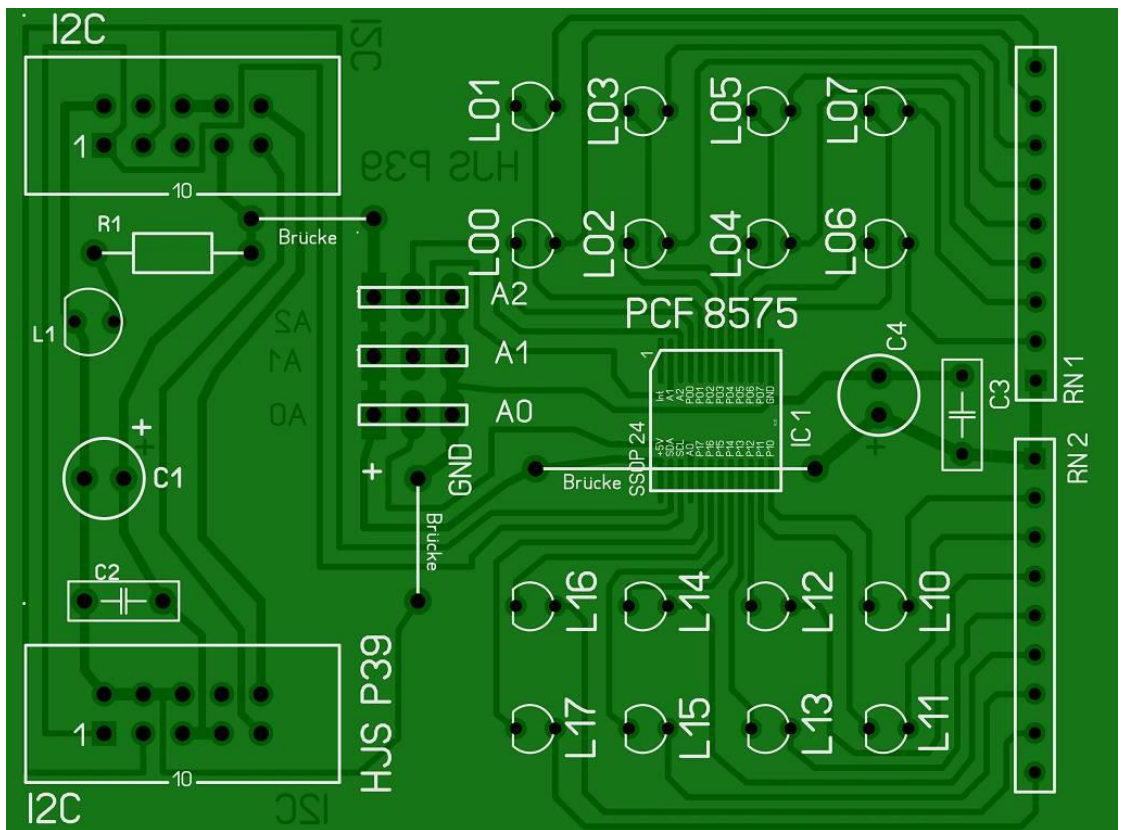


Sehen wir uns die Platine einmal genauer an. Man kann auf der Unterseite die Leiterzüge für den PCF 8575 genau erkennen. Zusätzlich habe ich noch die genaue Belegung dazu geschrieben.

Platine in der Durchsicht



Platine in der Fotoansicht



Die Platinen habe ich wieder mit Sprint Layout 6.0 gezeichnet. Falls jemand Interesse hat, kann er von mir die originalen Dateien bekommen.

Bitte die genaue Montage der beiden RN beachten. Sie sind gegeneinander verdreht montiert, dadurch liegt der Anschluss +5V in der Mitte. Zusätzlich habe ich die beide Kondensatoren in der Nähe des PCF 8575 zur Abblockung eingebaut.

### Stückliste:

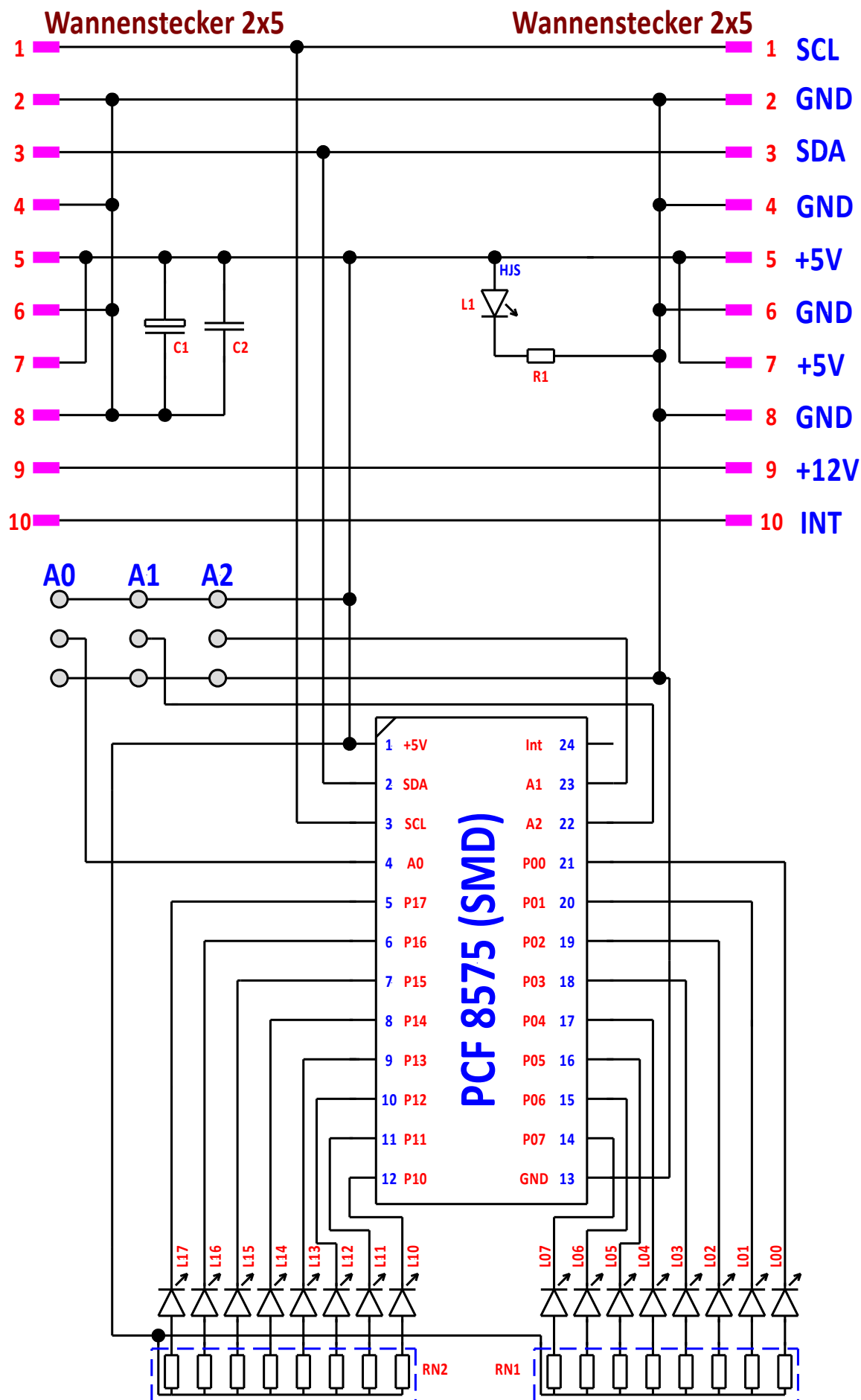
<b>RN1, RN2</b> - Widerstandsnetzwerk 220 Ohm	2 x Wannenstecker 2x5
<b>R1</b> - Widerstand 220 Ohm	3 x Jumper
<b>C2, C3</b> - Kondensator 100 nF	Platine ca. 72 x 53 mm
<b>C1, C4</b> - Elko 100/16	3 x Steckerleiste 3 polg.
<b>L1</b> - LED 5 mm rot (20 mA)	1 x PCF 8575 ( <b>Achtung SMD</b> )
<b>L00 - L07</b> - LED 3 mm grün 20 mA	<b>L10 - L17</b> - LED 3 mm gelb 20 mA

Durch die Jumper A0, A1 und A2 können unterschiedliche Adressen ausgewählt werden. Dabei stehen die Adressräume von 0x32 bis 0x39 zur Verfügung. Dadurch können bis zu 8 Schaltkreise an einen Bus angeschlossen werden. In der Tabelle habe ich die einzelnen Adressen angegeben. Bitte den Unterschied zwischen Dez und Hex beachten.

( Adressen wurden dem Datenblatt des Herstellers entnommen )

			PCF 8575 (SMD)	
			Dez	Hex
+	A0	A1		
+	A1	A2		
GND	A0	A1	32	20
+	A1	A2		
GND	A0	A1	33	21
+	A1	A2		
GND	A0	A1	34	22
+	A1	A2		
GND	A0	A1	35	23
+	A1	A2		
GND	A0	A1	36	24
+	A1	A2		
GND	A0	A1	37	25
+	A1	A2		
GND	A0	A1	38	26
+	A1	A2		
GND	A0	A1	39	27

Auf der nächsten Seite habe ich die Schaltung dargestellt. Im oberen Bereich befindet sich wieder der Busanschluss mit der Abblockung und der Anzeige der Betriebsspannung. Darunter kommen die drei Adressstecker. (A0, A1, A2) Im mittleren Bereich befindet sich der eigentlich PCF 8575. Bitte daran denken, es ist ein SMD IC. Er wird direkt auf die Leiterzüge (von unten) gelötet. Darunter befindet sich die 16 LED mit den beiden Widerstandsnetzwerken.



Natürlich gibt es wieder ein Programm zum testen. Es werden auf jeder Seite 2 LED angesteuert. Beachtet bitte den kompletten Aufbau.

```

/* ATB_Out2_1.c 31.07.2014 19:57:28 Author: AS */

#define F_CPU 16000000L           // definiere auf 16MHz
#include <util/delay.h>           // Einbinden der .h Datei
#include "i2cmaster.h"

unsigned char adr1_w = 0x40;     // Schreibadresse
unsigned char adr1_r = 0x41;     // Leseadresse

unsigned char adr2_w = 0x42;     // Schreibadresse
unsigned char adr2_r = 0x43;     // Leseadresse

unsigned char d;                 // Taster
unsigned char e;                 // LED

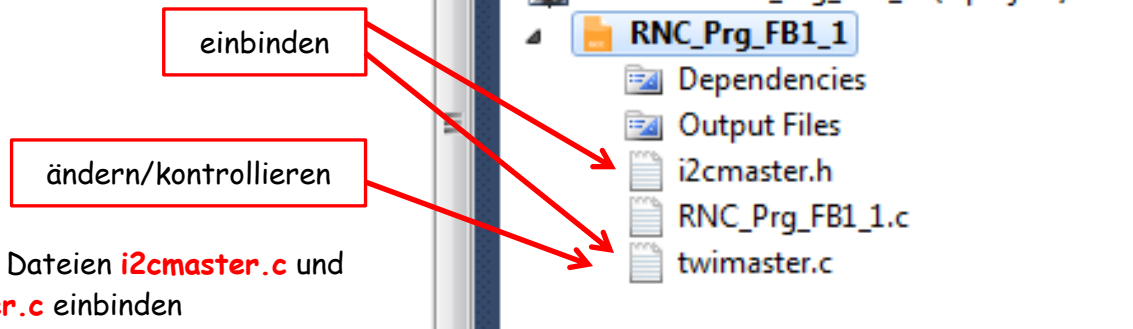
int main(void)
{
    // Hauptprogramm
    i2c_init ();
    i2c_start(adr1_w);           // Schreibbefehl für Device 1
    i2c_write(0xff);            // Alle Pins des PCF auf 0

    while(1)
    {
        // Hauptschleife
        i2c_write(0xff);        // Alle Pins des PCF auf 0
        i2c_start(adr1_r);      // Starte Lesezugriff
        d=i2c_readNak();        // Schreib Leseergebnis in d

        if (~d & 0x01)          // Abfrage T1
        {
            // Wenn T1 gedrückt ist...
            i2c_start(adr2_w);  // Schreibbefehl
            e = 0x7e;           // Angabe LED Port0
            i2c_write(e);       // Schreibe e
            e = 0xbd;           // Angabe LED Port1
            i2c_write(e);       // Schreibe e
            _delay_ms(100);     // 100ms warten
        }
        else
        {
            e = 0xff;           // alle LED aus
            i2c_start(adr2_w);  // Schreibbefehl
            i2c_write(e);       // Schreibe e
        }
    }
    _delay_ms(100);
    i2c_stop();
}

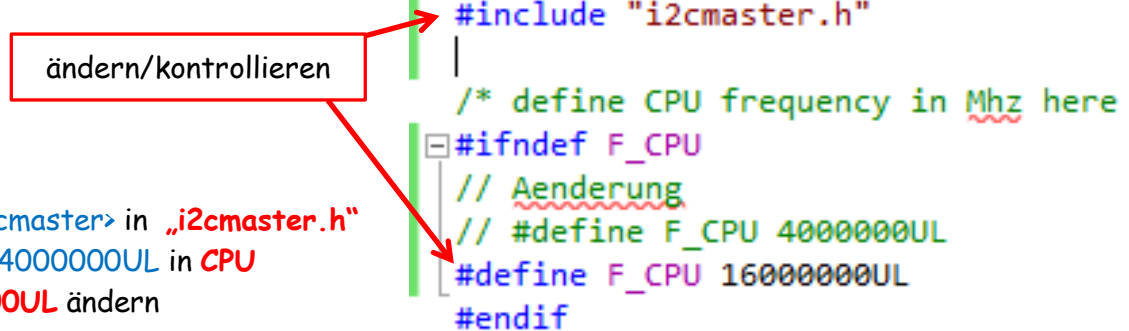
```

Wir müssen wieder zwei Programme einbinden. Es ist `i2cmaster.h` und `twimaster.c`.



Bitte die Dateien `i2cmaster.c` und `twimaster.c` einbinden

In der Datei `twimaster.c` müssen wir die Änderungen vornehmen oder die Einstellungen kontrollieren



Bitte `<i2cmaster>` in `„i2cmaster.h“` und `CPU 4000000UL` in `CPU 16000000UL` ändern

In einem anderen Tutorial habe ich genau beschrieben, wie notwendige Programme eingebunden werden.

Nach dem ich diese Änderungen durchgeführt habe, dürfte es keine Problem geben das Programm zu starten und auszuführen.

Sehen wir uns das Programm einmal genauer an:

```
#define F_CPU 16000000L          // definiere auf 16MHz
#include <util/delay.h>         // Einbinden der .h Datei
#include "i2cmaster.h"
```

**Angabe der Frequenz für Prozessor und einbinden der notwendigen Dateien**

```
unsigned char adr1_w = 0x40;    // Schreibadresse
unsigned char adr1_r = 0x41;    // Leseadresse
unsigned char adr2_w = 0x42;    // Schreibadresse
unsigned char adr2_r = 0x43;    // Leseadresse
```

**Adressen für den I<sup>2</sup>C Bus**

```
unsigned char d;               // Taster
unsigned char e;               // LED
```

**Variablen für Taster und LED**



```
int main(void)
{
    // Hauptprogramm
    i2c_init ();
    i2c_start(adr1_w); // Schreibbefehl für Device 1
    i2c_write(0xff); // Alle Pins des PCF auf 0
```

## Start des Hauptprogrammes und des I<sup>2</sup>C Busses

```
while(1)
{
    // Hauptschleife
    i2c_write(0xff); // Alle Pins des PCF auf 0
    i2c_start(adr1_r); // Starte Lesezugriff
    d=i2c_readNak(); // Schreib Leseergebnis in d
```

## Start der Schleife und Abfrage Taster 1

```
if (~d & 0x01) // Abfrage T1
{
    // Wenn T1 gedrückt ist...
    i2c_start(adr2_w); // Schreibbefehl
    e = 0x7e; // Angabe LED Port0
    i2c_write(e); // Schreibe e
    e = 0xbd; // Angabe LED Port1
    i2c_write(e); // Schreibe e
    _delay_ms(100); // 100ms warten
}
```

## Wenn Taster 1 gedrückt, dann schalte LED ...

```
else
{
    e = 0xff; // alle LED aus
    i2c_start(adr2_w); // Schreibbefehl
    i2c_write(e); // Schreibe e
}
```

## Wenn Taster T1 nicht gedrückt, schalte Led ... aus

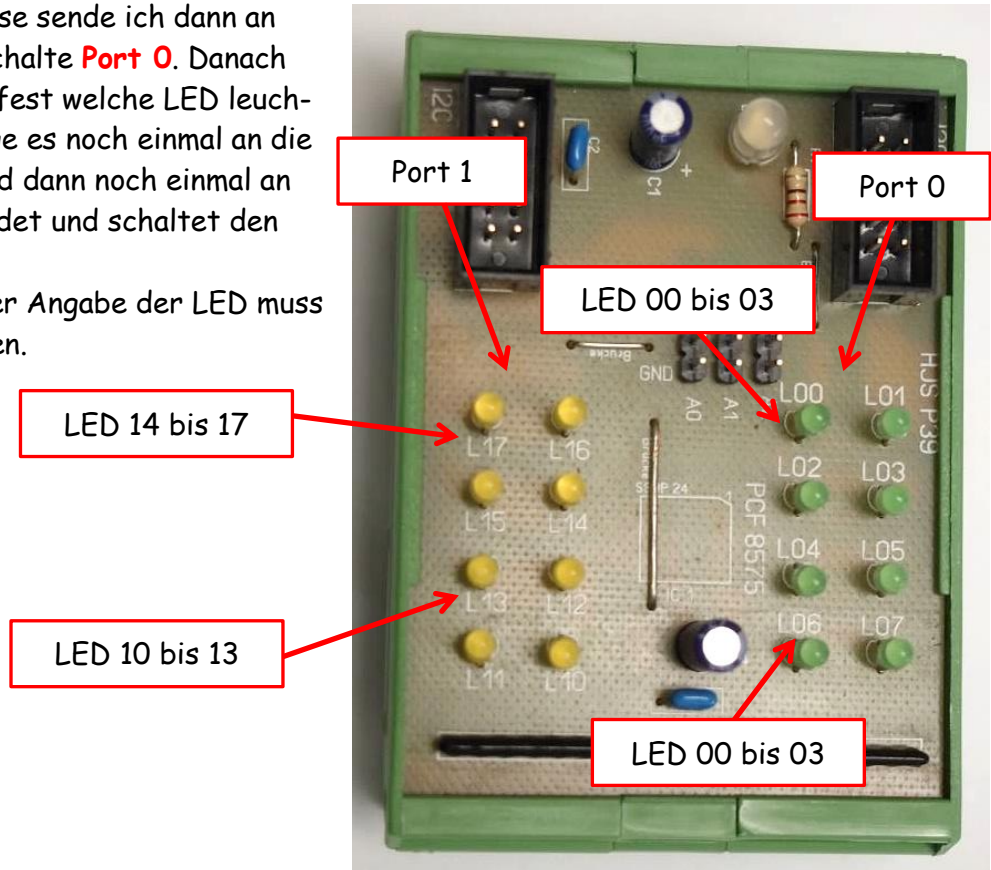
```
}
_delay_ms(100);
i2c_stop();
}
```

Bei diesem IC gibt es eine Besonderheit. Die Angabe der Ports wird nacheinander gesendet.

```
i2c_start(adr2_w); // Schreibbefehl
e = 0x7e; // Angabe LED Port0
i2c_write(e); // Schreibe e
e = 0xbd; // Angabe LED Port1
i2c_write(e); // Schreibe e
_delay_ms(100); // 100ms warten
```

Zuerst wird die **adr2** gesendet, dann lege ich fest welche LED leuchten soll und übertrage das in die Variable **e**. Diese sende ich dann an den **PCF 8575** und schalte **Port 0**. Danach lege ich noch einmal fest welche LED leuchten soll und übertrage es noch einmal an die Variable **e**. Diese wird dann noch einmal an den **PCF 8575** gesendet und schaltet den **Port 1**.

Die Variable **e** mit der Angabe der LED muss 2x übertragen werden.



Noch mal das gleich Stück aus dem Programm:

```
i2c_start(adr2_w);           // Schreibbefehl
e = 0x7e;                   // Angabe LED Port0
i2c_write(e);               // Schreibe e
e = 0xbd;                   // Angabe LED Port1
i2c_write(e);               // Schreibe e
_delay_ms(100);             // 100ms warten
```

In der Variablen **e** wird Zuordnung der LED hinterlegt. Sehen wir uns den Aufbau einmal genauer an:

1. Angabe **e = 0 x 7e;** // Angabe LED Port 0
2. Angabe **e = 0 x bd;** // Angabe LED Port 1

Die Variable **e** besteht neben dem **0x ..** noch aus **.. 7e** bzw. **.. bd**.

Untergliedern wir diese Angabe in ein linkes **Zeichen** und in ein rechtes **Zeichen**.

Jeder dieser beiden **Zeichen** hat die Aufgabe verschiedene LED anzusteuern bzw. dem PCF 8575 mitzuteilen welcher Pin angesteuert werden soll. Dadurch können wir verschieden LED gleichzeitig schalten.

**Rechtes Zeichen** schaltet die Pin 4 - 7 → LED 00 - 03 bzw. Pin 13 - 16 → LED 10 - 13

**Linkes Zeichen** schaltet die Pins 8 - 11 → LED 04 - 07 bzw. Pin 17 - 20 → LED 14 - 17

In der nächsten Tabelle habe ich einmal die entsprechenden Kombinationen zusammengefasst.

Rechtes Zeichen	Pin 4/13	Pin 5/14	Pin 6/15	Pin 7/16
Linkes Zeichen	Pin 8/17	Pin 9/18	Pin 10/19	Pin 11/20
0	X	X	X	X
1	---	X	X	X
2	X	---	X	X
3	---	---	X	X
4	X	X	---	X
5	---	X	---	X
6	X	---	---	X
7	---	---	---	X
8	X	X	X	---
9	---	X	X	---
a	X	---	X	---
b	---	---	X	---
c	X	X	---	---
d	---	X	---	---
e	X	---	---	---
f	---	---	---	---

PIN an: X      PIN aus: ---



Diese BM (Busmodule) habe ich verwendet

**PBM Board 1**

ATmega 1284p



**BM Netzteil 2**

(+5V, +12V, 3A)



## BM In/Out 1

( 4xT + 4xLED )

Adresse 40



## BM Out 2 (16 Bit)

( 16 x LED )

Adresse 42

Zum besseren Verständnis habe ich wieder einige Programme geschrieben:

- ATB\_I2C\_Out2\_1 → T1 und 1 x LED
- ATB\_I2C\_Out2\_2 → T2 und 2 x LED
- ATB\_I2C\_Out2\_3 → T3 und 3 x LED
- ATB\_I2C\_Out2\_4 → T4 und 4 x LED
- ATB\_I2C\_Out2\_5 → T1 bis T4 und 1, 2, 3 und 4 x LED

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](mailto:myroboter@web.de)