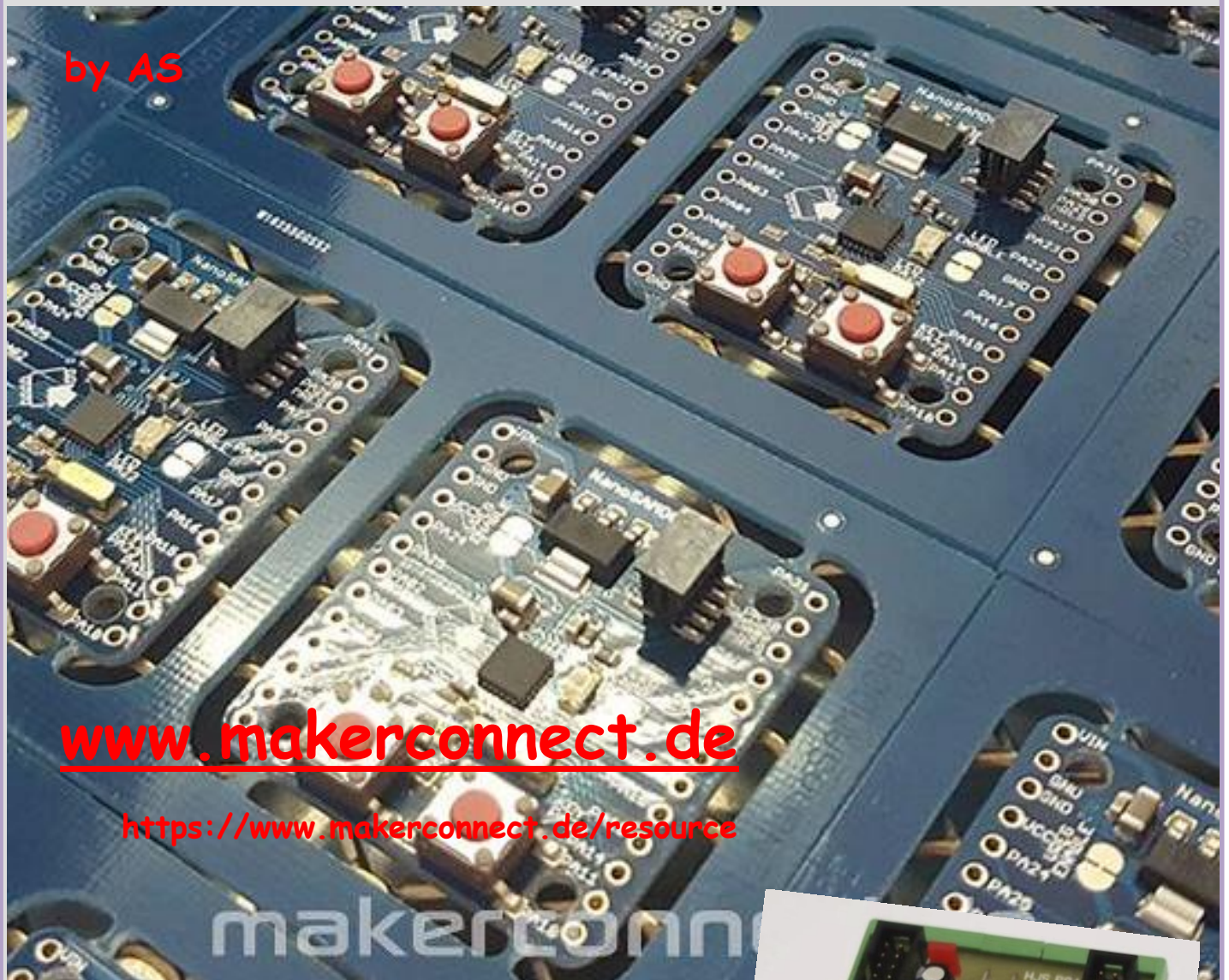


MIKROKONTROLLER & I²C BUS

by AS

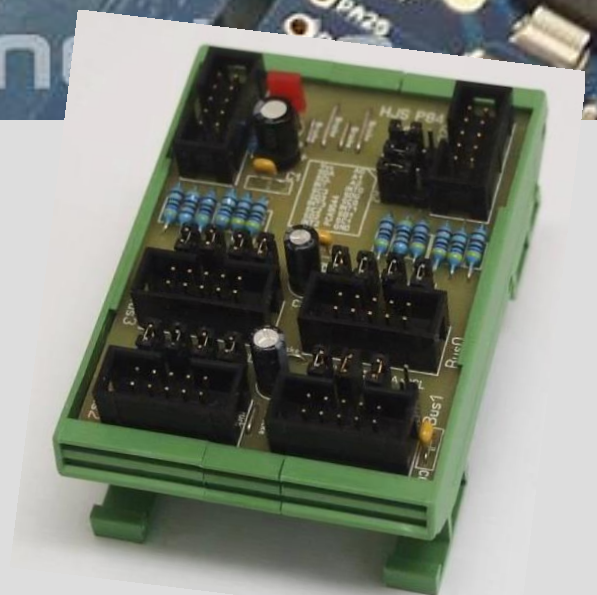


www.makerconnect.de

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

Gleiche Adressen am I²C Bus
Teil 2 - Software

I²C Bus und
der PCA 9544



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 Fehler 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

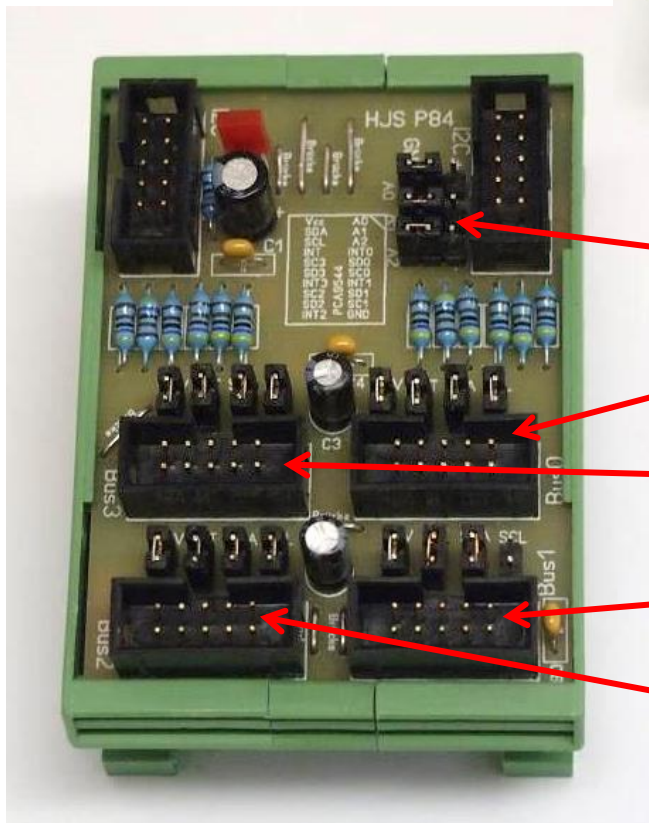
- Auf keinen Fall darf 230 V~ Netzspannung angeschlossen werden. Es besteht dann Lebensgefahr!
- Dieser Bausatz ist nur zum Einsatz unter Lern- und Laborbedingungen konzipiert worden. Er ist nicht geeignet, reale Steuerungsaufgaben jeglicher Art zu übernehmen. Ein anderer Einsatz als angegeben ist nicht 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.

Der I2C Bus und der PCA9544 - Teil 1

Im ersten Teil habe ich die Hardware zu dem PCA9544 vorgestellt. Mit Hilfe dieses Multiplexers können mehrere Slave mit gleichen Adressen an einem I²C Bus betrieben werden.

PCA9544 auf der Platine 84 mit 4 Kanälen für den I²C Bus mit verschiedenen Schaltmöglichkeiten an den Ausgangskanälen und Adresswahl

Sehen wir uns noch mal die Adressstecker und die Steckbuchsen für die verschiedenen Ausgänge an.



Adressen A0, A1, A2

Kanal 0

Kanal 3

Kanal 1

Kanal 2

Auf dem Bild habe ich noch mal die Zuordnung der Adressen und der Steckbuchsen zu den einzelnen Kanälen dargestellt. Mit den DIP Schaltern (Jumper) kann ich zu jedem Kanal Widerstände schalten um die Vcc auf den Bus zu legen.

Welche Einstellungen muss ich den machen um den PCA9544 zu nutzen?

Am besten man sieht in das Datenblatt des Herstellers. In der Figur 9 wird die Einstellung der Adressen für den I²C Bus angegeben.

Es können eingestellt werden:

Von - **0 x 111 0 000 0 = 0 x e0**

Bis - **0 x 111 0 111 0 = 0 x ee**

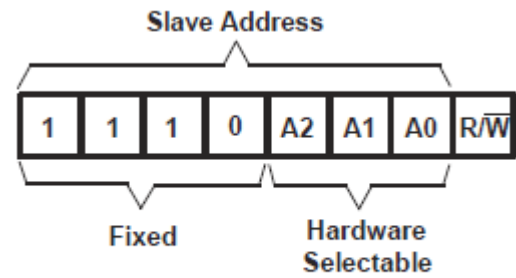


Figure 9. PCA9544A Address

Nach dem Start muss der Busmaster die Adresse des Slaves ausgeben, auf den er zugreift. Die Auswahl der Adresse des PCA9544 ist in Figure 9 dargestellt.

Das letzte Bit der Slave-Adresse definiert den auszuführenden Vorgang. Wenn sie auf eine Logik 1 eingestellt ist, wird ein Lesen ausgewählt, während eine Logik 0 einen Schreibvorgang auswählt.

Table 1. Control Register Write (Channel Selection), Control Register Read (Channel Status)⁽¹⁾

INT3	INT2	INT1	INT0	D3	B2	B1	B0	COMMAND
X	X	X	X	X	0	X	X	No channel selected
X	X	X	X	X	1	0	0	Channel 0 enabled
X	X	X	X	X	1	0	1	Channel 1 enabled
X	X	X	X	X	1	1	0	Channel 2 enabled
X	X	X	X	X	1	1	1	Channel 3 enabled
0	0	0	0	0	0	0	0	No channel selected, power-up default state

(1) Only one channel may be selected at a time.

Ein oder mehrere nachgeschaltete SCn/SDn-Paare oder Kanäle werden durch den Inhalt des Steuerregisters ausgewählt (siehe Tabelle 1). Dieses Register wird geschrieben, nachdem die PCA9544A angesprochen wurde. Die drei LSBs (B2, B1, B0) des Kontrollbytes werden verwendet, um zu bestimmen, welcher Kanal (oder welche Kanäle) ausgewählt werden sollen.

Im Programm kann ich den ausgewählten Kanal so angeben:

```
i2c_write(0x04);           // Sende Daten Channel 0
i2c_write(0x05);           // Sende Daten Channel 1
i2c_write(0x06);           // Sende Daten Channel 2
i2c_write(0x07);           // Sende Daten Channel 3
```

Table 2. Control Register Read (Interrupt)⁽¹⁾

INT3	INT2	INT1	INT0	D3	B2	B1	B0	COMMAND
X	X	X	0	X	X	X	X	No interrupt on channel 0
			1					Interrupt on channel 0
X	X	X	X	X	X	X	X	No interrupt on channel 1
								Interrupt on channel 1
X	X	X	X	X	X	X	X	No interrupt on channel 2
								Interrupt on channel 2
X	X	X	X	X	X	X	X	No interrupt on channel 3
								Interrupt on channel 3

(1) Several interrupts can be active at the same time. For example, INT3 = 0, INT2 = 1, INT1 = 1, INT0 = 0 means that there is no interrupt on channels 0 and 3, and there is interrupt on channels 1 and 2.

Der PCA9544A verfügt über vier Interrupt-Eingänge (einer für jeden Kanal) und einen Open-Drain-Interrupt Ausgang. Wenn ein Interrupt vom PCA9544 erkannt wird, wird der Interrupt Ausgang auf low geschaltet.

Der Kanal muss zur Erkennung des Interrupts nicht aktiv sein. Die Bits 4-7 des Steuerregisters entsprechen den Kanälen 0-3 des PCA9544. Bei Nichtgebrauch müssen die Interrupteingänge mit dem Vcc verbunden werden.

Auf weitere Funktionen und Anwendung werde ich nicht weiter eingehen. Weitere Infos bitte dem Datenblatt des Herstellers entnehmen.

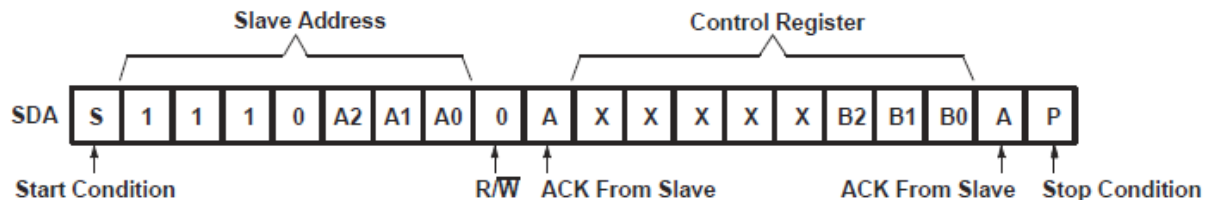


Figure 7. Write Control Register

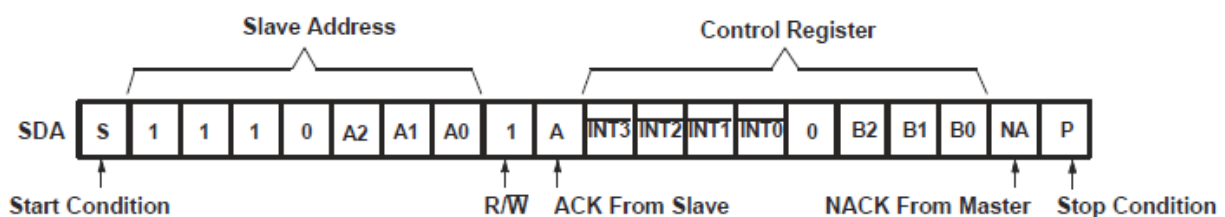


Figure 8. Read Control Register

In den Figure 7 und 8 habe ich noch mal die genaue Belegung der Register für Write und Read angegeben. Die Bilder und andere habe ich dem Datenblatt des Herstellers entnommen.

Kommen wir zu einem Beispiel für die Umsetzung eines Programmes.

Als ersten nehmen wir ein Programm um eine LED auf einem I²C Bus Slave blinken zu lassen:

```
/* ATB I2C PCA9544 Prg 1.c Created: 03.08.2018 21:02:09 Author : hjsee */
// Hardware: Board1, NT2, P11 mit PCF8574
#define F_CPU 16000000L // definiere auf 16MHz
#include <util/delay.h> // Einbinden der .h Datei
#include "i2cmaster.h"
unsigned char adr1_w = 0x48; // Adresse des PCF8574
int main(void)
{
    // Hauptprogramm
    i2c_init ();
    while(1)
    {
        // Hauptschleife
        i2c_start(adr1_w); // Schreibbefehl
        i2c_write(0xff); // Alle Pins des PCF aus
        i2c_stop();
        _delay_ms(1000); // 1000ms warten
        i2c_start(adr1_w); // Schreibbefehl
        i2c_write(0x0f); // Reihe 1 oder 2 Pins des PCF an
        i2c_stop();
        _delay_ms(1000); // 1000ms warten
    }
}
```

Funktion des Programmes:

Der Bus Master schaltet auf dem Slave (PCF 8574 mit Adresse 0x48) abwechselnd LEDs mit einer Frequenz von 2 Hz ein und aus.

Das Programm wollen wir nun über einen PCA9544 betreiben.

```

/* ATB I2C PCA9544 Prg 1.c Created: 03.08.2018 21:02:09 Author : hjsee */
// Hardware: Board1, NT2, P11 mit PCF8574

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

unsigned char adr1_w = 0x48;      // Schreibadresse des PCF
unsigned char adr2 = 0xe0;        // Adresse PCA9544

int main(void)
{
    // Hauptprogramm
    i2c_init ();
    while(1)
    {
        // Hauptschleife
        // Beispiel Chanel 0
        i2c_start(adr2);          // Start i2C mit Adresse Slave PCA9544
        i2c_write(0x04);          // Sende Daten Channel 0
        // i2c_write(0x05);        // Sende Daten Channel 1
        // i2c_write(0x06);        // Sende Daten Channel 2
        // i2c_write(0x07);        // Sende Daten Channel 3
        i2c_stop();

        i2c_start(adr1_w);        // Schreibbefehl PF8574
        i2c_write(0xff);          // Alle Pins des PCF aus
        i2c_stop();

        _delay_ms(1000);          // 1000ms warten

        // Beispiel Chanel 0
        i2c_start(adr2);          // Start i2C mit Adresse Slave PCA9544
        i2c_write(0x04);          // Sende Daten Channel 0
        // i2c_write(0x05);        // Sende Daten Channel 1
        // i2c_write(0x06);        // Sende Daten Channel 2
        // i2c_write(0x07);        // Sende Daten Channel 3
        i2c_stop();

        i2c_start(adr1_w);        // Schreibbefehl PCF8574
        i2c_write(0x0f);          // Reihe 1 oder 2 Pins des PCF an
        i2c_stop();

        _delay_ms(1000);          // 1000ms warten
    }
}

```

Nochmal die wichtigsten Befehle:

```
// Beispiel Channel 0
i2c_start(adr2);           // Start i2C mit Adresse Slave PCA9544
i2c_write(0x04);           // Sende Daten Channel 0
// i2c_write(0x05);        // Sende Daten Channel 1
// i2c_write(0x06);        // Sende Daten Channel 2
// i2c_write(0x07);        // Sende Daten Channel 3
i2c_stop();
```

Erklärung am Channel 0. Es wird die Adresse **adr2** aufgerufen. Dann wird in das Register **0x04** geschrieben. Damit erfolgt die Auswahl des Channels 0.

Es muss vor jeder Übertragung von Daten an einen Slave die Auswahl des Channels erfolgen.

Die Angaben **0x05**, **0x06** und **0x07** sind für die Channels **1**, **2** und **3**.

Die Ansteuerung des PCF8574 erfolgt definitiv nur über den Channel 0.

Ansicht des fertigen Modules



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