

MIKROKONTROLLER & I²C BUS



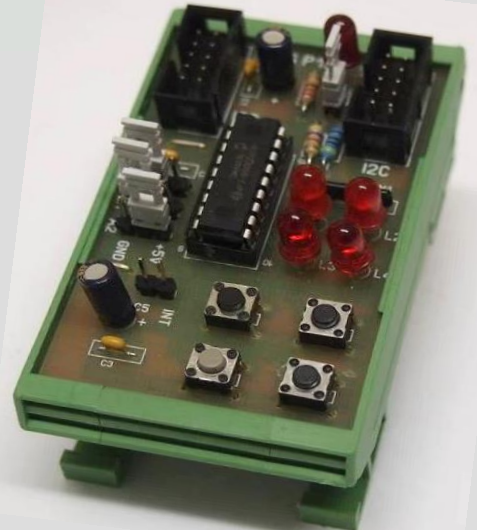
by AS

www.boxtec.ch

playground.boxtec.ch/doku.php/tutorial

I²C Bus und der MCP23008
Teil 2 - Software

I²C Bus und
der MCP 23008



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 Selbsthilfwerkstätten 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.

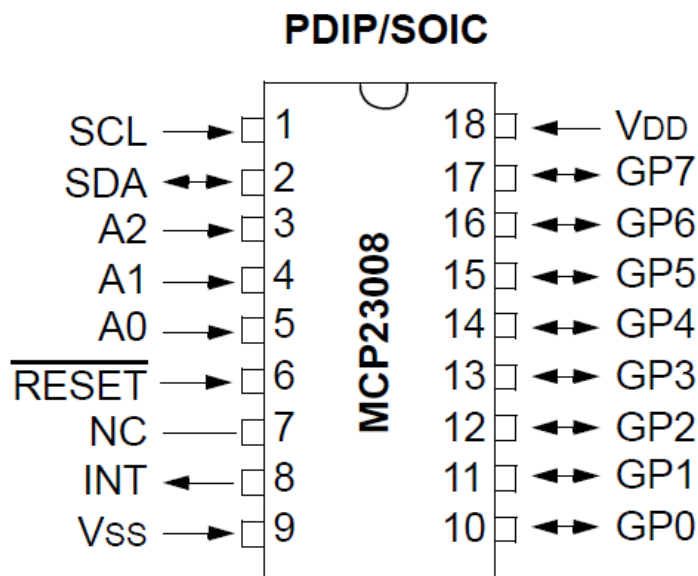
Der I2C Bus und der MCP 23008 - Teil 2

Auch der MCP23008 hat aufgrund seiner Register einige gewöhnungsbedürftige Einstellungen gegenüber seinen Vorgängern.

MCP 23008 auf der Platine 158 mit 2 Anschlüssen für den I²C Bus, 4 LEDs zur Anzeige, 4 Taster zur Eingabe, 3 Stecker zur Auswahl der Adresse, Stecker zur Schaltung der Interrupts, Jumper zum Anlegen der Vcc an den Bus und der Anzeige der Betriebsspannung



Ansicht des ICs von oben mit Pinbelegung



Die einzelnen Pins die wir benutzen:

LED1 - LED4 - GP 4 bis GP 7 - Pin 14 bis 17

T1 - T4 - GP 0 bis GP 3 - Pin 10 bis 13

Bus SCL - Pin 1

Bus SDA - Pin 2

Angabe Adresse A0 / A1 / A2 - PIN 5 / 4 / 3

Betriebsspannung Vcc / GND - Pin 18 / 9

Diese beiden Tabellen habe ich dem Datenblatt des Herstellers entnommen.

Es zeigt die Adresse und das zugehörige Register.

In der Tabelle 1-3 habe ich noch mal die Konfiguration der Steuerung Register dargestellt

Address	Access to:
00h	IODIR
01h	IPOL
02h	GPINTEN
03h	DEFVAL
04h	INTCON
05h	IOCON
06h	GPPU
07h	INTF
08h	INTCAP (Read-only)
09h	GPIO
0Ah	OLAT

TABLE 1-3: CONFIGURATION AND CONTROL REGISTERS

Register Name	Address (hex)	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0	POR/RST value
IODIR	00	IO7	IO6	IO5	IO4	IO3	IO2	IO1	IO0	1111 1111
IPOL	01	IP7	IP6	IP5	IP4	IP3	IP2	IP1	IP0	0000 0000
GPINTEN	02	GPINT7	GPINT6	GPINT5	GPINT4	GPINT3	GPINT2	GPINT1	GPINT0	0000 0000
DEFVAL	03	DEF7	DEF6	DEF5	DEF4	DEF3	DEF2	DEF1	DEF0	0000 0000
INTCON	04	IOC7	IOC6	IOC5	IOC4	IOC3	IOC2	IOC1	IOC0	0000 0000
IOCON	05	—	—	SREAD	DISSLW	HAEN*	ODR	INTPOL	—	--00 000-
GPPU	06	PU7	PU6	PU5	PU4	PU3	PU2	PU1	PU0	0000 0000
INTF	07	INT7	INT6	INT5	INT4	INT3	INT2	INT1	INT0	0000 0000
INTCAP	08	ICP7	ICP6	ICP5	ICP4	ICP3	ICP2	ICP1	ICP0	0000 0000
GPIO	09	GP7	GP6	GP5	GP4	GP3	GP2	GP1	GP0	0000 0000
OLAT	0A	OL7	OL6	OL5	OL4	OL3	OL2	OL1	OL0	0000 0000

* Not used on the MCP23008.

Die genaue Erklärungen habe ich in der Datei **MCP 23008.h** als Kommentar angegeben.

```
#define MCP23008_IODIR 0x00 // Steuert die Richtung der Daten am Port
#define MCP23008_IPOL 0x01 // konfiguriert die Polarität GPIO Port Bits
#define MCP23008_GPINTEN 0x02 // steuert die Interrupt-On-Change
#define MCP23008_DEFVAL 0x03 // Standardvorgabe für Eingangspins
#define MCP23008_INTCON 0x04 // legt fest wie der zugehörige Pin-Wert für die Interrupt-On-
// Change-Funktion verglichen wird
#define MCP23008_IOCON 0x05 // enthält etliche Bits für die Konfiguration
#define MCP23008_GPPU 0x06 // steuert die Pull-UP-Widerstände für Port-Pins
#define MCP23008_INTF 0x07 // spiegelt die Interupt-Bedingungen auf den Port-Pins wieder
#define MCP23008_INTCAP 0x08 // erfassen den GPIO-Port-Wert zum Zeitpunkt des Interrups
#define MCP23008_GPIO 0x09 // geben den Wert der Portleitungen zurück.
#define MCP23008_OLAT 0x0A // erlaubt den Zugriff auf die als Ausgang konfigurierten Pins
// Register A
```

Im oberen Teil habe ich den Inhalt der Datei **MCP23008.h** dargestellt. Ein Teil der Kommentare habe ich entfernt. Wer mehr wissen möchte, bitte die Datei anschauen.

Die Angabe der Frequenz unseres Prozessors und der I²C Bus Adresse erfolgt in der Datei **main.h**. Für die Kommunikation des I²C Busses verwende ich die Dateien von Peter Fleury.

Zum Start des Hauptprogrammes verwende ich diese Dateien:

```
#include <stdbool.h>
#include <avr/pgmspace.h>
#include "main.h"
#include "i2cmaster.h"
#include "avr/io.h"
#include "util/delay.h"
#include "avr/interrupt.h"
#include "stdlib.h"
#include "MCP23008.h"
```

Ob sie alle benötigt werden habe ich nicht getestet.

Im nächsten Teil erfolgt die Einstellung der Register:

```
uint8_t Data;
int main(void)
{
    i2c_init();                // initialize I2C library

    //setzt Port als Eingang/Ausgang mit IODIR
    i2c_start(MCP23008_ADDR);  // Angabe Adresse
    i2c_write(MCP23008_IODIR); // Steuert die Richtung der Daten am Port A
    i2c_write(0x0F);          // Angabe Ausgang 0x00-alle, 0xff-alle aus
    i2c_stop();

    // Eingänge R einschalten
    i2c_start(MCP23008_ADDR);  // Angabe Adresse
    i2c_write(MCP23008_GPPU);  // Taster schalten Masse (active-low)
    i2c_write(0x0F);          // alle Eingänge invertieren
    i2c_stop();

    while(1)
    {
        // Abfrage der Eingänge am MCP
        i2c_start(MCP23008_ADDR); // Angabe Adresse
        i2c_write(MCP23008_GPIO);
        i2c_stop();

        i2c_start(MCP23008_ADDR + 1); // Auslesen Adresse + 1
        Data = i2c_readNak();         // liest Eingänge ein
        i2c_stop();

        // Abfrage Taster 1
        if (Data & 0x01)              // Abfrage DATA, welcher Taster
        {                             // Angabe Taster mit 0x01
            i2c_start(MCP23008_ADDR); // Angabe Adresse
            i2c_write(MCP23008_OLAT); // Schalte Ausgang
            i2c_write(0x40);          // Auswahl der LED
            i2c_stop();
        }
        else
    }
```

```

    {
        i2c_start(MCP23008_ADDR); // Angabe Adresse
        i2c_write(MCP23008_OLAT); // Schalte Ausgang
        i2c_write(0x00); // Auswahl der LED, alle aus
        i2c_stop();
    }
// Abfrage Taster 2
if (Data & 0x02) // Abfrage DATA, welcher Taster
    { // Angabe Taster mit 0x02
        i2c_start(MCP23008_ADDR); // Angabe Adresse
        i2c_write(MCP23008_OLAT); // Schalte Ausgang
        i2c_write(0x80); // Auswahl der LED
        i2c_stop();
    }
// Abfrage Taster 3
if (Data & 0x04) // Abfrage DATA, welcher Taster
    { // Angabe Taster mit 0x04
        i2c_start(MCP23008_ADDR); // Angabe Adresse
        i2c_write(MCP23008_OLAT); // Schalte Ausgang
        i2c_write(0x10); // Auswahl der LED
        i2c_stop();
    }
// Abfrage Taster 4
if (Data & 0x08) // Abfrage DATA, welcher Taster
    { // Angabe Taster mit 0x08
        i2c_start(MCP23008_ADDR); // Angabe Adresse
        i2c_write(MCP23008_OLAT); // Schalte Ausgang
        i2c_write(0x20); // Auswahl der LED
        i2c_stop();
    }
} // while
}

```

Weiter Angaben stehen in den Kommentaren

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

