

MIKROKONTROLLER & I²C BUS



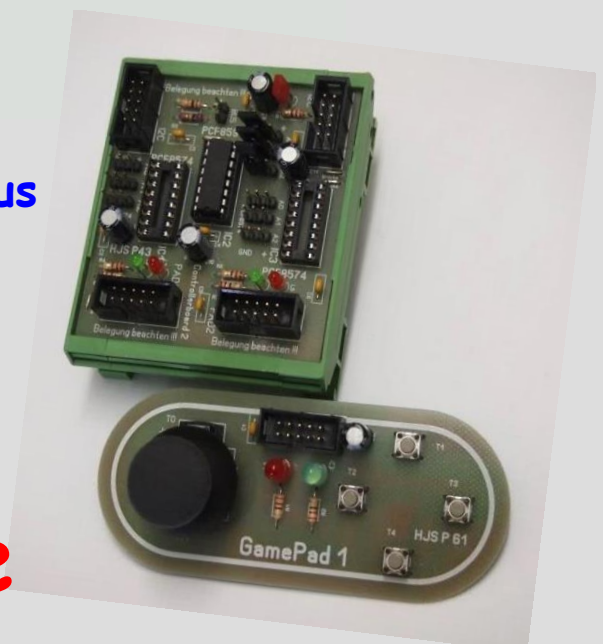
by AS

www.boxtec.ch

playground.boxtec.ch/doku.php/tutorial

Eine Spielkonsole mit dem I²C Bus
Teil 2 - Software 1

Der I²C Bus als Spielkonsole



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.

Eine Spielkonsole mit dem I²C Bus - Software 1

Als erstes möchte ich die Software zur Kontrolle der Joystick vorstellen. Die Aufgabe dieser Software ist es, die Joysticks auszuwerten und anzuzeigen. Dabei soll eine Begrenzung der möglichen Anzeige erfolgen, sowie die Stellungsanzeige und ein Umrechnung der Spannung. Anschliessend soll die Spannung korrekt angezeigt werden. Die Frequenzangabe und die Adresse der Anzeige stehen in der main.h

```
/* ATB Gameboard P43 Author: AS */
#include <stdbool.h>
#include <avr/pgmspace.h>
#include "main.h"
#include <util/delay.h>
#include "i2clcd.h"
#include "i2cmaster.h"
#include "avr/io.h"
#include "avr/interrupt.h"
#include <stdlib.h>

#define PCF8591w 0x90
#define PCF8591r 0x91

int8_t i;
float results[5],adc1 ,adc2 ,adc3 ,adc4;

void startanzeige() // Titelbild
{
    lcd_command(LCD_CLEAR);
    _delay_ms(2);
    lcd_printlc(1,2,"Gameboard 2 - I2C");
    lcd_printlc(2,1,"Modul P43-2xJoystick");
    lcd_printlc(3,2,"AT1284 und OLED");
    lcd_printlc(4,4,"bei achim S.");
    _delay_ms(5000);
}

int main(void)
{
    char Buffer[20];
    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
    startanzeige();
    lcd_command(LCD_CLEAR); // Leere Display
    _delay_ms(2); // warte 2ms
}
```

```
while(1)
{
    i2c_start(PCF8591w);
    i2c_write(0x04);
    i2c_stop();
    i2c_start(PCF8591r);
    i2c_readAck();
    for (i=0; i<4; i++)
    {
        results[i]=i2c_readAck();
    }
    i2c_readNak();
    i2c_stop();
    _delay_ms(2);

    // Anzeige der P und ADC Werte
    itoa( results[0], Buffer, 10 );           // Zeile 1
    lcd_printlc(1,1,"P1 ");
    lcd_printlc(1,4,Buffer);

    itoa( results[1], Buffer, 10 );           // Zeile 2
    lcd_printlc(2,1,"P2 ");
    lcd_printlc(2,4,Buffer);

    itoa( results[2], Buffer, 10 );           // Zeile 3
    lcd_printlc(3,1,"P3 ");
    lcd_printlc(3,4,Buffer);

    itoa( results[3], Buffer, 10 );           // Zeile 4
    lcd_printlc(4,1,"P4 ");
    lcd_printlc(4,4,Buffer);

    lcd_printlc(1,8,"U1 ");
    lcd_printlc(2,8,"U2 ");
    lcd_printlc(3,8,"U3 ");
    lcd_printlc(4,8,"U4 ");

    lcd_printlc(1,16,"V");
    lcd_printlc(2,16,"V");
    lcd_printlc(3,16,"V");
    lcd_printlc(4,16,"V");

    lcd_printlc(1,12,".");                   // Komma 1 Z
    lcd_printlc(2,12,".");                   // Komma 2 Z
    lcd_printlc(3,12,".");                   // Komma 3 Z
    lcd_printlc(4,12,".");                   // Komma 4 Z

    // Zeile 1
    int16_t adc1l, adc1a, adc1r1, adc1r2, adc1b;
    adc1=(results[0]*196);
    if (adc1>50200)
```

```
{
  adc1=50200;
}

adc1=adc1/100;           // 1. Schritt
adc1a=adc1;             // 2. Schritt
adc1r2=adc1a % 10;      // 3. Schritt rechts 2
adc1b=adc1a/10;         // 4. Schritt
adc1r1=adc1b % 10;      // 5. Schritt rechts 1
adc1l=adc1b/10;         // 6. Schritt links

itoa( adc1l, Buffer, 10 ); // Zeile 1 Zahl 1
lcd_printlc(1,11,Buffer);
itoa( adc1r1, Buffer, 10 ); // Zeile 1 Zahl 2
lcd_printlc(1,13,Buffer);
itoa( adc1r2, Buffer, 10 ); // Zeile 1 Zahl 3
lcd_printlc(1,14,Buffer);

// Zeile 2
int16_t adc2l, adc2a, adc2r1, adc2r2, adc2b;
adc2=(results[1]*196);
if (adc2>50200)
{
  adc2=50200;
}

adc2=adc2/100;           // 1. Schritt
adc2a=adc2;             // 2. Schritt
adc2r2=adc2a % 10;      // 3. Schritt rechts 2
adc2b=adc2a/10;         // 4. Schritt
adc2r1=adc2b % 10;      // 5. Schritt rechts 1
adc2l=adc2b/10;         // 6. Schritt links

itoa( adc2l, Buffer, 10 ); // Zeile 2 Zahl 1
lcd_printlc(2,11,Buffer);
itoa( adc2r1, Buffer, 10 ); // Zeile 2 Zahl 2
lcd_printlc(2,13,Buffer);
itoa( adc2r2, Buffer, 10 ); // Zeile 2 Zahl 3
lcd_printlc(2,14,Buffer);

// Zeile 3
int16_t adc3l, adc3a, adc3r1, adc3r2, adc3b;
adc3=(results[2]*196);
if (adc3>50200)
{
  adc3=50200;
}

adc3=adc3/100;           // 1. Schritt
adc3a=adc3;             // 2. Schritt
adc3r2=adc3a % 10;      // 3. Schritt rechts 2
```

```
adc3b=adc3a/10; // 4. Schritt
adc3r1=adc3b % 10; // 5. Schritt rechts 1
adc3l=adc3b/10; // 6. Schritt links

itoa( adc3l, Buffer, 10 ); // Zeile 3 Zahl 1
lcd_printlc(3,11,Buffer);
itoa( adc3r1, Buffer, 10 ); // Zeile 3 Zahl 2
lcd_printlc(3,13,Buffer);
itoa( adc3r2, Buffer, 10 ); // Zeile 3 Zahl 3
lcd_printlc(3,14,Buffer);

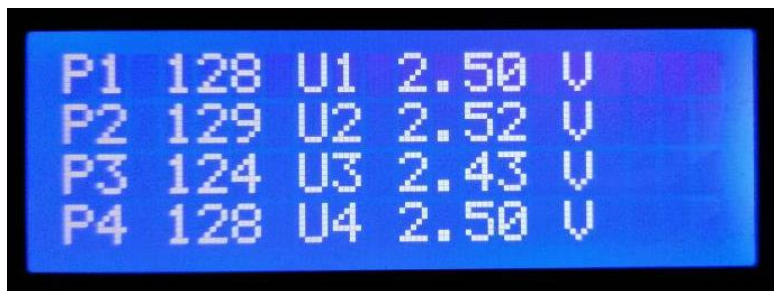
// Zeile 4
int16_t adc4l, adc4a, adc4r1, adc4r2, adc4b;
adc4=(results[3]*196);
if (adc4>50200)
{
    adc4=50200;
}

adc4=adc4/100; // 1. Schritt
adc4a=adc4; // 2. Schritt
adc4r2=adc4a % 10; // 3. Schritt rechts 2
adc4b=adc4a/10; // 4. Schritt
adc4r1=adc4b % 10; // 5. Schritt rechts 1
adc4l=adc4b/10; // 6. Schritt links

itoa( adc4l, Buffer, 10 ); // Zeile 4 Zahl 1
lcd_printlc(4,11,Buffer);
itoa( adc4r1, Buffer, 10 ); // Zeile 4 Zahl 2
lcd_printlc(4,13,Buffer);
itoa( adc4r2, Buffer, 10 ); // Zeile 4 Zahl 3
lcd_printlc(4,14,Buffer);

    _delay_ms(50);
}
}
```

Das ist die gesamte Software für den ersten Teil. Die Bibliotheken für den Bus und LCD habe ich hier nicht angegeben.



Anzeige der Werte auf einem Display. Beide Joysticks befinden sich in Mittelstellung

Falls sie jemand haben möchte kann ich sie gern schicken.

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