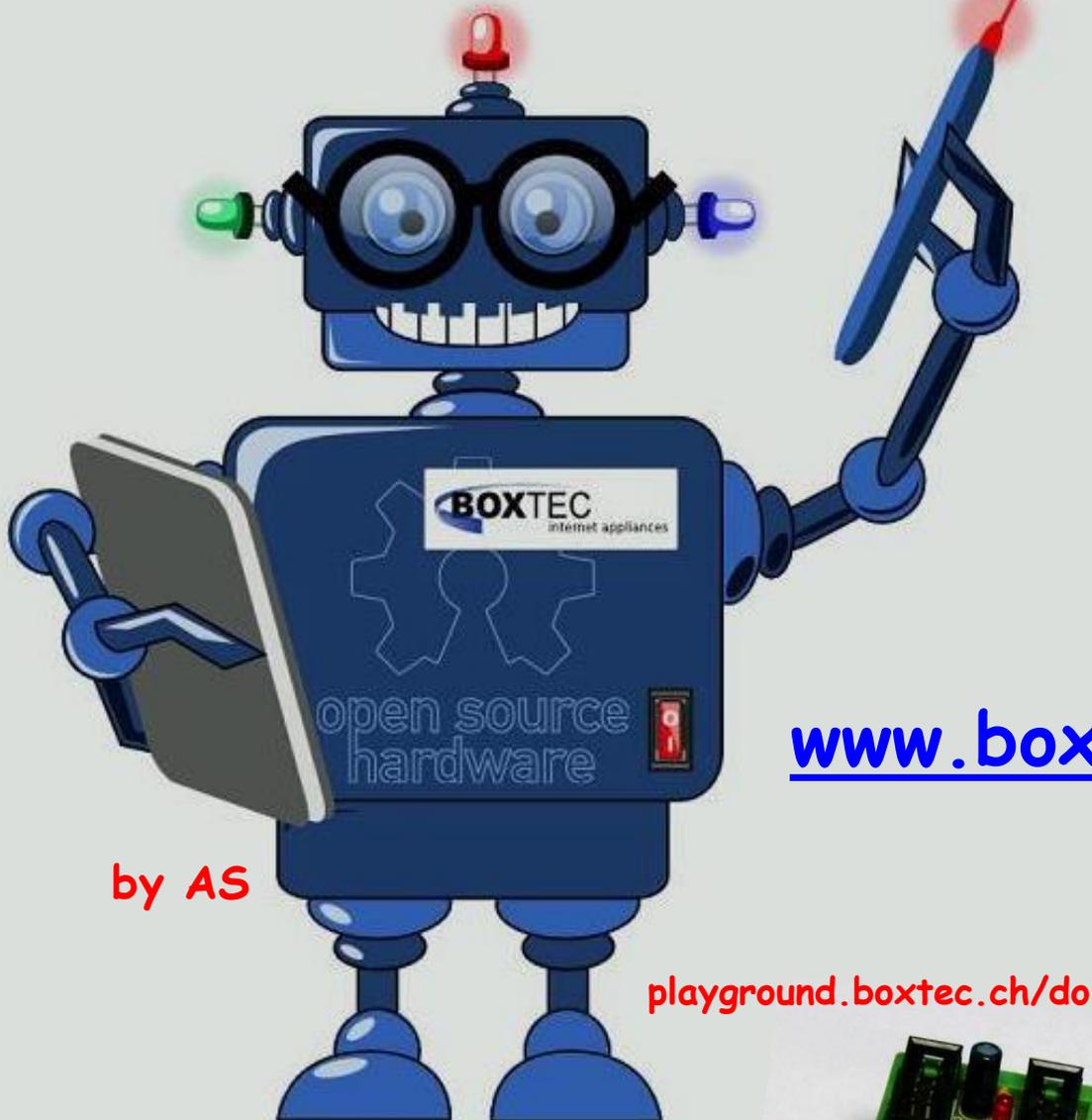


MIKROKONTROLLER & I²C BUS



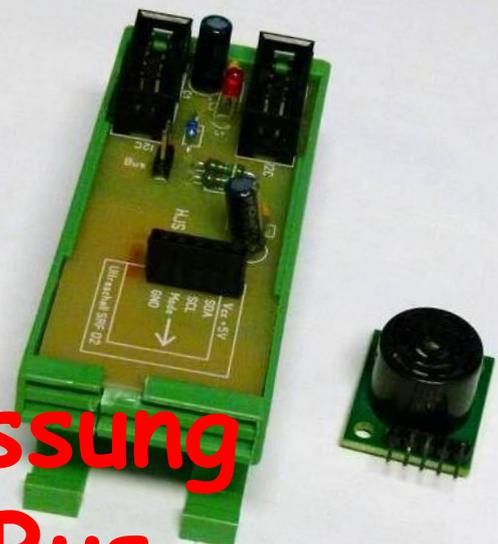
by AS

www.boxtec.ch

playground.boxtec.ch/doku.php/tutorial

Entfernungsmessung
mit dem SRF02
Teil 2 - Software

Entfernungsmessung
mit dem I²C Bus



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.

Entfernungsmessung mit dem SRF02 - Teil 2 - Software

Im ersten Teil habe ich die Hardware zum SRF02 beschrieben. Auf der Platine P152 befindet sich der US-Sensor, die Auswertung erfolgt mit der Platine P 30 und dem Atmega 1284p.

Der SRF 02 verfügt über
6 Register

An Hand dieser Register bzw. Einstellungen möchte ich zeigen wie die Messung erfolgt und die einzelnen Aufgabe kurz vorstellen.

Registernummer	Rückgabe wenn Register gelesen	Funktion wenn Register beschrieben wird
0	Software Firmware Revision	Befehlsregister
1	Ungenutzt (liefert immer den Wert Hex 80)	nicht möglich
2	Entfernungswert (High Byte)	nicht möglich
3	Entfernungswert (Low Byte)	nicht möglich
4	Autotune Minimum (High Byte)	nicht möglich
5	Autotune Minimum (Low Byte)	nicht möglich

Befehlscode zur
Übertragung
an den SRF02

Mit 0x51 starte
ich die Messung
in cm

Befehlscode		Funktion
Decimal	Hex	
80	0x50	Startet Messvorgang (gemessen wird in der Einheit inches)
81	0x51	Startet Messvorgang (gemessen wird in der Einheit Zentimeter)
82	0x52	Startet Messvorgang (gemessen wird in der Einheit Microsekunden)
86	0x56	Synchron Messvorgang (gemessen wird in der Einheit inches)
87	0x57	Synchron Messvorgang (gemessen wird in der Einheit Zentimeter)
88	0x58	Synchron Messvorgang (gemessen wird in der Einheit Microsekunden)
92	0x5C	Erzeugt einen 8 zyklischen 40khz Impuls/Ton
96	0x60	Startet automatische Kalibrierung – Befehl kann ignoriert werden da dies bereits beim einschalten (Power on) automatisch erfolgt
160	0xA0	Erste Sequence zum ändern der I2C-Bus Slave ID
165	0xA5	Dritte Sequence zum ändern der I2C-Bus Slave ID
170	0xAA	Zweite Sequence zum ändern der I2C-Bus Slave ID

Die Software habe ich in verschiedene Aufgaben unterteilt:

- Kontrolle ob der US-Sensor und das Display am Bus erkannt werden
- Abfrage und Anzeige der Version
- Eigentliche Messung der Entfernung in cm und Bildung des Abstandes
- Berechnung des Abstandes und Anzeige des Wertes

Die Adressen für den I²C Bus:

```
#define adr1_w 0x40 // Adresse Display
#define adr_srf02 0xe0 // Adresse US-Modul SRF02
```

Innerhalb der main erfolgt der Aufruf der einzelnen Unterprogramme:

```
i2c_init(); // Starte I2C Bus
lcd_init(); // Starte I2C-LCD
startanzeige(); // Start Bildschirm
testbus(); // Kontrolle der Module am Bus
abfrage_version(); // Starte Abfrage Version
messung_srf02(); // Starte eigentliche Messung
anzeige_werte(); // Starte eigentliche Anzeige des Abstandes
```

Durch die Aufteilung der einzelnen Funktionen kann ich den Code leicht in anderen Programmen verwenden und jedes Teile separat schreiben und testen.

```
#define F_CPU 16000000UL // CPU clock in Hz
#include <stdbool.h>
#include <avr/pgmspace.h>
#include <util/delay.h>
#include "i2clcd.h"
#include "i2cmaster.h"
#include "avr/io.h"
#include "util/delay.h"
#include "stdlib.h"

uint8_t ret1; // Kontrollvariable ob Slave 1 vorhanden
uint8_t ret2; // Kontrollvariable ob Slave 2 vorhanden
uint16_t version;
uint16_t messung = 0;
uint8_t messung_high = 0;
uint8_t messung_low = 0;
#define adr1_w 0x40 // Adresse Display
#define adr_srf02 0xe0 // Adresse US-Modul SRF02
char wert[8];
// Anzeige Startbildschirm
void startanzeige() // Titelbild
{
    lcd_command(LCD_CLEAR); // Leere Display
    _delay_ms(5); // Warte 5ms
    lcd_printlc(1,2,"US-Modul mit SRF02"); // Zeile 1
    lcd_printlc(2,1,"====="); // Zeile 2
    lcd_printlc(3,3," Programm 1 "); // Zeile 3
    lcd_printlc(4,4,"(by achim S.)"); // Zeile 4
}
// Abfrage ob Display und SRF02 vorhanden ist und Anzeige
void testbus()
{
    // Abfrage ob Slave vorhanden ist
    ret1 = i2c_start(adr1_w); // Start i2C mit Adresse 1 Slave
    i2c_write(0x00); // Sende Daten
    i2c_stop(); // I2C Stop
    ret2 = i2c_start(adr_srf02); // Start i2C mit Adresse 2 Slave
    i2c_write(0x00); // Sende Daten
    i2c_stop(); // I2C Stop
    _delay_ms(3);
    lcd_command(LCD_CLEAR); // Leere Display
    _delay_ms(5);
    lcd_printlc(1,3,"Kontrolle des Bus");
    lcd_printlc(2,1,"=====");
    if (ret1==0) // Bus ok - 0, kein Bus 1
```

```
{
  lcd_printlc(3,3,"Display ist ok");           // Ausgabe Text
}
if (ret2==0)                                  // Bus ok - 0, kein Bus 1
{
  lcd_printlc(4,3,"US-Modul ist ok");         // Ausgabe Text
}
else
{
  lcd_printlc(4,3,"US-Modul defekt");         // Ausgabe Text
}
}
// Abfrage und Anzeige Version
void abfrage_version()
{
  i2c_start(adr_srf02);                       // I2C Slaveadresse übergeben
  i2c_write(0);                               // Register 0 wählen
  i2c_start(adr_srf02 + 1);                   // I2C Slaveadresse übergeben
  version=i2c_readNak();                      // Register 0 lesen
  i2c_stop();
  lcd_command(LCD_CLEAR);                     // Leere Display
  _delay_ms(5);
  lcd_printlc(1,3,"Abfrage Version");
  lcd_printlc(2,1,"=====");
  utoa(version,wert,16);                       // Berechnung und Anzeige des Wertes
  lcd_printlc(3,2,"Version:");
  lcd_printlc(3,14,wert);
}
// Eigentliche Messung durchführen
void messung_srf02()
{
  i2c_start(adr_srf02);                       // I2C Slaveadresse übergeben
  i2c_write(0);                               // Register 0 wählen
  i2c_write(0x51);                            // Messung starten in cm
  i2c_stop();
  _delay_ms(65);                              // 65ms Wartezeit bis Messung fertig
  i2c_start(adr_srf02);                       // I2C Slaveadresse übergeben
  i2c_write(2);                               // Register 2 auswählen
  i2c_start(adr_srf02+1);                     // I2C Slaveadresse übergeben
  messung_high = i2c_readAck();                // Register 2 (high) lesen
  messung_low = i2c_readNak();                 // Register 3 (low) lesen und stop
  messung = (( messung_high << 8) + messung_low); // Register 2 + Register 3
  i2c_stop();
}
// Berechne Wert für Abstand und Anzeige des Wertes
void anzeige_werte()                          // Anzeige Wert für Abstand
```

```
{
  lcd_command(LCD_CLEAR);           // Leere Display
  _delay_ms(5);                    // Warte 5ms
  lcd_printlc(1,3,"Anzeige Abstand"); // Zeile 1
  lcd_printlc(2,1,"=====");     // Zeile 2
  lcd_printlc(3,16,"cm");         // Zeile 3
  utoa(messung,wert,10);          // Berechnung und Anzeige des Wertes
  lcd_printlc(3,2,"Abstand:");
  lcd_printlc(3,12,wert);         // Anzeige in Zeile 3
  _delay_ms(5000);
}
// Hauptprogramm mit Aufruf der einzelnen Unterprogramme
int main(void)
{
  i2c_init();                      // Starte I2C Bus
  _delay_ms(2);
  lcd_init();                      // Starte I2CLCD
  _delay_ms(2);
  // Display Befehle
  lcd_command(LCD_DISPLAYON | LCD_CURSOROFF | LCD_BLINKINGOFF);
  _delay_ms(2);
  lcd_light(0);                   // Licht an
  _delay_ms(2);
  startanzeige();
  _delay_ms(3000);               // Warte
  testbus();                      // Kontrolle der Module am Bus
  _delay_ms(2000);
  abfrage_version();
  _delay_ms(3000);
  while(1)
  {
    messung_srf02();
    _delay_ms(10);
    anzeige_werte();
    _delay_ms(30);
  }
  return 0;
}
```

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

Falls jemand das komplette Programm braucht, kann ich es euch gern schicken.