

MIKROKONTROLLER & I²C BUS



by AS

www.boxtec.c

playground.boxtec.ch/doku.php/tutorial

Strommessung mit dem I²C Bus
Teil 3 - Software 2

I²C Bus und
der INA 219



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.

Strommessung mit dem I²C Bus - Teil 3

Im dritten Teil dieses Tuts möchte ich die Strom- und Spannungsmessung vorstellen. Vom Hersteller wird der INA219 mit der folgenden Einstellung geliefert:

Werkseinstellungen

PG Bit: /8 (+/- 320mV)
 ADC Einstellung: 12 Bit
 Mode Einstellung: Shunt und Bus, kontinuierlich

Ansicht Modul mit INA219



Tabelle der Register und eingestellten Werte des Herstellers

Table 2. Summary of Register Set

POINTER ADDRESS HEX	REGISTER NAME	FUNCTION	POWER-ON RESET		TYPE ⁽¹⁾
			BINARY	HEX	
00	Configuration	All-register reset, settings for bus voltage range, PGA Gain, ADC resolution/averaging.	00111001 10011111	399F	R \bar{W}
01	Shunt voltage	Shunt voltage measurement data.	Shunt voltage	—	R
02	Bus voltage	Bus voltage measurement data.	Bus voltage	—	R
03	Power ⁽²⁾	Power measurement data.	00000000 00000000	0000	R
04	Current ⁽²⁾	Contains the value of the current flowing through the shunt resistor.	00000000 00000000	0000	R
05	Calibration	Sets full-scale range and LSB of current and power measurements. Overall system calibration.	00000000 00000000	0000	R \bar{W}

Die Register **Shunt voltage (01) (Spannung über den Shunt)** und **Bus voltage (02) (Busspannung)** können ohne Eingabe in das **Calibration (05) (Kalibrations Register)** ausgelesen werden. Die Register **Power (03)**, **Strom (04)** und **Calibration (05)** sind vom Hersteller mit **0** vorbelegt worden. Werden die Register **Power (03)** und **Current (04)** ausgelesen erfolgt die Ausgabe von **0**.

Im zweiten Teil habe ich bereits beschrieben, wie das Register **01** und **02** ausgelesen werden kann. In diesem Teil möchte die Berechnung für das **Calibration Register (05)** vorstellen und an einem Beispiel erklären.

Nach der Eingabe eines **Calibration Wertes (05)** können die Register **Power(03)** und **Strom (04)** ausgelesen und berechnet werden.

Bitte die angezeigten Ergebnisse mit einem Messgerät vergleichen.

In den nächsten Tabellen habe ich die Einstellungen des Herstellers kopiert. Damit kann jeder die Angaben kontrollieren oder seine eigenen Einstellungen ablesen. Es gelten die grau hinterlegten Angaben.

V _{SHUNT}	Full-scale current sense (input) voltage range	PGA = /1	0	±40
		PGA = /2	0	±80
		PGA = /4	0	±160
		PGA = /8	0	±320
	Bus voltage (input voltage) range ⁽²⁾	BRNG = 1	0	32
		BRNG = 0	0	16

Table 4. PG Bit Settings⁽¹⁾

PG1	PG0	GAIN	Range
0	0	1	±40 mV
0	1	/2	±80 mV
1	0	/4	±160 mV
1	1	/8	±320 mV

Table 5. ADC Settings⁽¹⁾

ADC4	ADC3	ADC2	ADC1	Mode/Samples	Conversion Time
0	X ⁽²⁾	0	0	9 bit	84 µs
0	X ⁽²⁾	0	1	10 bit	148 µs
0	X ⁽²⁾	1	0	11 bit	276 µs
0	X ⁽²⁾	1	1	12 bit	532 µs
1	0	0	0	12 bit	532 µs
1	0	0	1	2	1.06 ms
1	0	1	0	4	2.13 ms
1	0	1	1	8	4.26 ms
1	1	0	0	16	8.51 ms
1	1	0	1	32	17.02 ms

Table 6. Mode Settings⁽¹⁾

MODE3	MODE2	MODE1	MODE
0	0	0	Power-down
0	0	1	Shunt voltage, triggered
0	1	0	Bus voltage, triggered
0	1	1	Shunt and bus, triggered
1	0	0	ADC off (disabled)
1	0	1	Shunt voltage, continuous
1	1	0	Bus voltage, continuous
1	1	1	Shunt and bus, continuous

Programmierung des Kalibrierregisters

Das INA219 Datenblatt enthält einige Beispiele für die Programmierung des Kalibrierregisters. Dieses Beispiel folgt der gleichen Reihenfolge, die im Datenblatt beschrieben ist. In einem Beispiel möchte ich einen Kalibrierwert ausrechnen. Die entsprechenden Werte sind dem Datenblatt des Herstellers zu entnehmen.

- Bestimme die maximale **Messspannung** für dein System und konfiguriere BRNG basierend auf diesen Wert (16V oder 32V)

$$V_{\text{Spannung}} = 32V \quad (\text{BRNG}=1)$$

- Bestimme die maximale Shuntspannung $V_{\text{Shunt max}}$ für das System und konfiguriere die PGA-Verstärkung basierend auf diese Werte (/1, /2, /4, /8) PGA stellt die Verstärkung und Empfindlichkeit ein. Die PGA ist standardmäßig auf /8 (320mV Bereich) eingestellt. Tabelle 4 im DB zeigt die Verstärkung und den Bereich für die verschiedenen Verstärkungseinstellungen.

$$V_{\text{Shunt max}} = 320mV \quad (\text{PGA}=8)$$

- Bestimme den Wert des Shunt-Widerstandes durch ablesen vom Schaltbild R_{Shunt} :

$$R_{Shunt} = 0,1 \Omega$$

- Berechne anhand der in den ersten drei Schritten gewählten Werte den maximalen Strom, der möglicherweise durch den Messwiderstand fließen könnte:

$$I_{max\ möglich} = \frac{V_{Shunt\ max}}{R_{Shunt}} = \frac{320mV}{0,1\Omega} = 3,2\ A$$

- Ermittle anhand der zu erwarteten Systemleistung den tatsächlich zu erwartenden Maximalstrom. Für dieses System nehmen wir an, dass es bis zu 3,2A Strom aufnehmen soll.

$$I_{max\ erwartet} = I_{max\ möglich} = 3,2\ A$$

- Das **Current Register** ist 16 Bit breit, aber da das Gerät bidirektionalen Strom messen kann, wird das MSB verwendet, um das Vorzeichen in der Zweierkomplement-Darstellung anzuzeigen. Daher können die restlichen 15 Bit verwendet werden, um den gesamten zu erwarteten I_{max} zu repräsentieren. Da die ADC-Auflösung 12 Bit beträgt, ist es wünschenswert, die **Current Register**-Auflösung zwischen 12-15 Bit zu wählen, idealerweise so nahe wie möglich an der 15-Bit-Auflösung.
- Berechne den minimalen Strom LSB (15 Bit Auflösung = 32767):

$$LSB_{minimum} = \frac{I_{max\ erwartet}}{32767} = \frac{3,2\ A}{32767} = 97,6\ \mu A$$

- Wähle einen aktuellen LSB-Wert, wobei $LSB_{minimum} < LSB_{erwartet} < LSB_{maximum}$

Aus dem Datenblatt:

"Beachten Sie, dass die Ergebnisse die höchste Auflösung haben, wenn das minimale LSB gewählt wird. Normalerweise wird ein LSB so gewählt, dass es dem minimalen LSB-Wert am nächsten kommt."

Wählen wir $LSB_{erwartet} = 98\ \mu A$

Berechnen den Wert des Kalibrierregisters: (Cal.Reg)

$$Cal.\ Reg = \left(\frac{0,04096}{LSB_{erwartet} * R_{Shunt}} \right)$$

$$= \frac{0,04096}{98\ \mu A * 0,1} = 4222 = 0x107E$$

Der Wert für das **Kalibrierregister (05)** ist 4222. Dabei handelt es sich um eine Dezimalzahl, umgerechnet auf hex ergibt sich **0x107E**. Im Programm tragen wir diesen Wert ein. Es sind weiter Berechnungen möglich, die ich nicht weiter führen möchte. Bei Interesse im Datenblatt des Herstellers nachlesen.

Sehen wir uns das Programm genauer an:

```
/* ATB_I2C_Strommessung_Prg2.c
 * Created: 07.04.2018 19:00:28 Author : Hans-Joachim Seeger */

// Meine Hardware: NT2, Board1 mit ATm 1284p, LCD Display, INA219 Modul

#include <stdbool.h>
#include <avr/pgmspace.h>
#include "main.h" // Angabe Adressen
#include <util/delay.h>
#include "i2clcd.h"
#include "i2cmaster.h"
#include "avr/io.h"
#include "util/delay.h"
#include "avr/interrupt.h"
#include "stdlib.h"

#define calibration 0x107E // Cal Reg 0x107E

uint16_t msb_strom; // Oberes Strom-Byte
uint16_t lsb_strom; // Unteres Strom-Byte
uint16_t strom_wrd; // Ganzes Strom-Wort
uint16_t strom_anz; // Ganzes Strom-Anzeige

uint16_t msb_spannung; // Oberes Spannungs-Byte
uint16_t lsb_spannung; // Unteres Spannungs-Byte
uint16_t spannung_wrd; // Ganzes Spannungs-Wort
uint16_t spannung_anz; // Ganzes Spannungs-Anzeige
uint16_t spannung_anz1; // Ganzes Spannungs-Anzeige
uint16_t spannung_anz2; // Ganzes Spannungs-Anzeige
uint16_t spannung_anz3; // Ganzes Spannungs-Anzeige
uint16_t spannung_anz4; // Ganzes Spannungs-Anzeige
uint16_t spannung_anz5; // Ganzes Spannungs-Anzeige
uint16_t spannung_anz6; // Ganzes Spannungs-Anzeige
uint16_t spannung_anz7; // Ganzes Spannungs-Anzeige

uint16_t msb_power; // Oberes Power-Byte
uint16_t lsb_power; // Unteres Power-Byte
uint16_t power_wrd; // Ganzes Power-Wort
uint16_t power_anz; // Ganzes Power-Anzeige

char Buffer[30]; // Umwandlungs-Variable für LCD Anzeige

void anzeige_1(void) // Anzeige erster Text
{
    lcd_printlc(1,2,"Strommessung"); // Zeile 1
    lcd_printlc(2,3,"Programm 2"); // Zeile 2
    lcd_printlc(3,1,"mit INA219 Modul"); // Zeile 3
    lcd_printlc(4,5,"(by AS)"); // Zeile 4
}

void anzeige_2(void) // Anzeige zweiter Text
{
    lcd_printlc(1,1,"U Shunt"); // Zeile 1
    lcd_printlc(1,15,"mV"); // Zeile 1
}
```

```
lcd_printlc(2,1,"U Bus:");           // Zeile 2
lcd_printlc(2,16,"V");              // Zeile 2
lcd_printlc(2,11,",");              // Zeile 2
lcd_printlc(3,1,"Strom");           // Zeile 3
lcd_printlc(3,15,"mA");             // Zeile 3
lcd_printlc(4,1,"Power");          // Zeile 4
lcd_printlc(4,15,"mW");            // Zeile 4
}

int main(void)
{
    cli();                          // Interrupts deaktiviert
    i2c_init();                      // Starte I2C Bus
    lcd_init();                      // Starte I2CLCD

    // Display Befehle
    lcd_command(LCD_DISPLAYON | LCD_CURSOROFF | LCD_BLINKINGOFF);
    lcd_command(LCD_CLEAR);          // Leere Display
    _delay_ms(2);                   // Warte 2ms
    anzeige_1();
    _delay_ms(3000);
    lcd_command(LCD_CLEAR);          // Leere Display
    _delay_ms(2);                   // Warte 2ms
    anzeige_2();
    _delay_ms(100);                 // Warte 100ms

    // Kalibrierung eingabe
    i2c_start(adr_ina219);           // Angabe Adresse
    i2c_write(0x05);                 // Angabe Calibration
    i2c_write(calibration >> 8);    // MSB
    i2c_write(calibration & 0xFF);  // LSB
    i2c_stop();

    while(1)
    {
        // Shunt voltage 01
        i2c_start(adr_ina219);       // Angabe Adresse
        i2c_write(01);               // Register Strom
        i2c_stop();
        i2c_start(adr_ina219 + 1);   // Auslesen Adresse + 1
        msb_strom = i2c_readAck();    //...speichere oberes Bit
        lsb_strom = i2c_readNak();    //...speichere unteres Bit
        i2c_stop();
        strom_wrd = (msb_strom << 8 | lsb_strom); // Zusammensetzung von o. & u. Byte
        strom_anz = strom_wrd / 8;    //Berechne Strom

        if (strom_anz == 8191)
        {
            strom_anz = 0;
        }

        itoa(strom_anz, Buffer, 10);
        lcd_printlc(1,10," ");
        lcd_printlc(1,10,Buffer);
    }
}
```

```
// Bus voltage 02
i2c_start(adr_ina219);           // Angabe Adresse
i2c_write(0x02);                // Register Spannung
i2c_stop();
i2c_start(adr_ina219 + 1);      // Auslesen Adresse + 1
msb_spannung = i2c_readAck();    //...speichere oberes Bit
lsb_spannung = i2c_readNak();    //...speichere unteres Bit
i2c_stop();
spannung_wrd = (msb_spannung << 8 | lsb_spannung); // Zusammensetzung von o. & u. Byte
spannung_anz = spannung_wrd / 2; // Berechnung Wert Spannung / 2

spannung_anz1 = spannung_anz % 10; // 4. Zahl Berechnung
spannung_anz2 = spannung_anz / 10;
spannung_anz3 = spannung_anz2 % 10; // 3. Zahl Berechnung
spannung_anz4 = spannung_anz2 / 10;
spannung_anz5 = spannung_anz4 % 10; // 2. Zahl Berechnung
spannung_anz6 = spannung_anz4 / 10;
spannung_anz7 = spannung_anz6 % 10; // 1. Zahl Berechnung

itoa(spannung_anz7, Buffer, 10 ); // 1. Zahl Anzeige
lcd_printlc(2,10," ");
lcd_printlc(2,10,Buffer);

itoa(spannung_anz5, Buffer, 10 ); // 2. Zahl Anzeige
lcd_printlc(2,12," ");
lcd_printlc(2,12,Buffer);

itoa(spannung_anz3, Buffer, 10 ); // 3. Zahl Anzeige
lcd_printlc(2,13," ");
lcd_printlc(2,13,Buffer);

itoa(spannung_anz1, Buffer, 10 ); // 4. Zahl Anzeige
lcd_printlc(2,14," ");
lcd_printlc(2,14,Buffer);

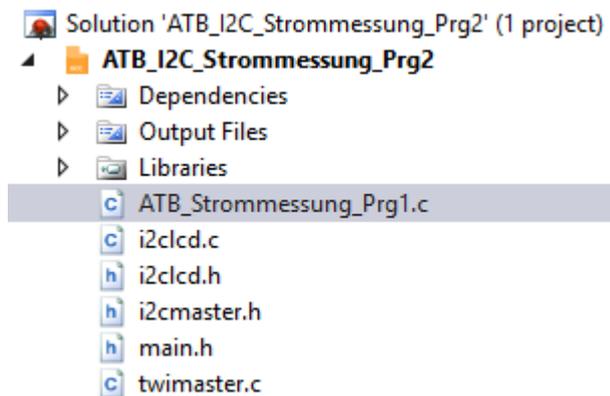
// Current 04
i2c_start(adr_ina219);           // Angabe Adresse
i2c_write(04);                   // Register Strom
i2c_stop();
i2c_start(adr_ina219 + 1);      // Auslesen Adresse + 1
msb_strom = i2c_readAck();       //...speichere oberes Bit
lsb_strom = i2c_readNak();      //...speichere unteres Bit
i2c_stop();
strom_wrd = (msb_strom << 8 | lsb_strom); // Zusammensetzung von o. & u. Byte
strom_anz = strom_wrd / 8;       // Berechnung Wert Strom / 8

if (strom_anz == 8191)           //Begrenzung Anzeige
{
    strom_anz = 0;
}

itoa(strom_anz, Buffer, 10 );
lcd_printlc(3,10," ");
lcd_printlc(3,10,Buffer);
```

```
// Power 03
i2c_start(adr_ina219);           // Angabe Adresse
i2c_write(0x03);                // Register Power
i2c_stop();
i2c_start(adr_ina219 +1);       // Auslesen Adresse + 1
msb_power = i2c_readAck();      //...speichere oberes Bit
lsb_power = i2c_readNak();      //...speichere unteres Bit
i2c_stop();
power_wrd = (msb_power << 8 | lsb_power); // Zusammensetzung von o. & u. Byte
power_anz = power_wrd * 2;      // Wert Spannung / 2
itoa(power_anz, Buffer, 10 );
lcd_printlc(4,10," ");
lcd_printlc(4,10,Buffer);
}
}
```

Diese Dateien müssen eingebunden werden



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