

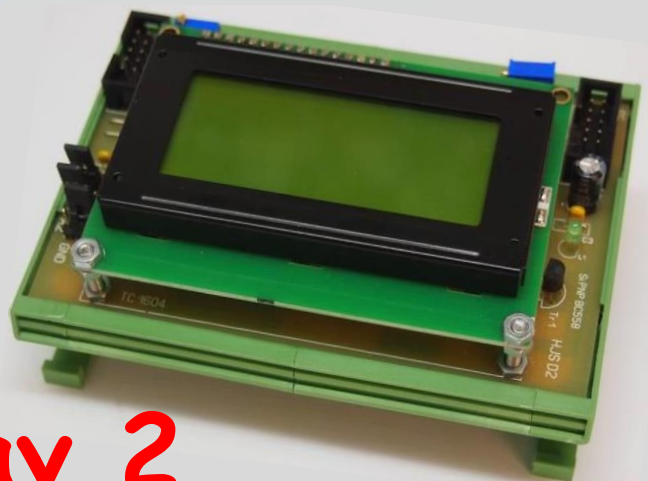
MIKROKONTROLLER & I²C BUS



www.boxtec.ch

playground.boxtec.ch/doku.php/tutorial

Bus - Modul
I²C - Display 2



BM - Display 2

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 Fehlers 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

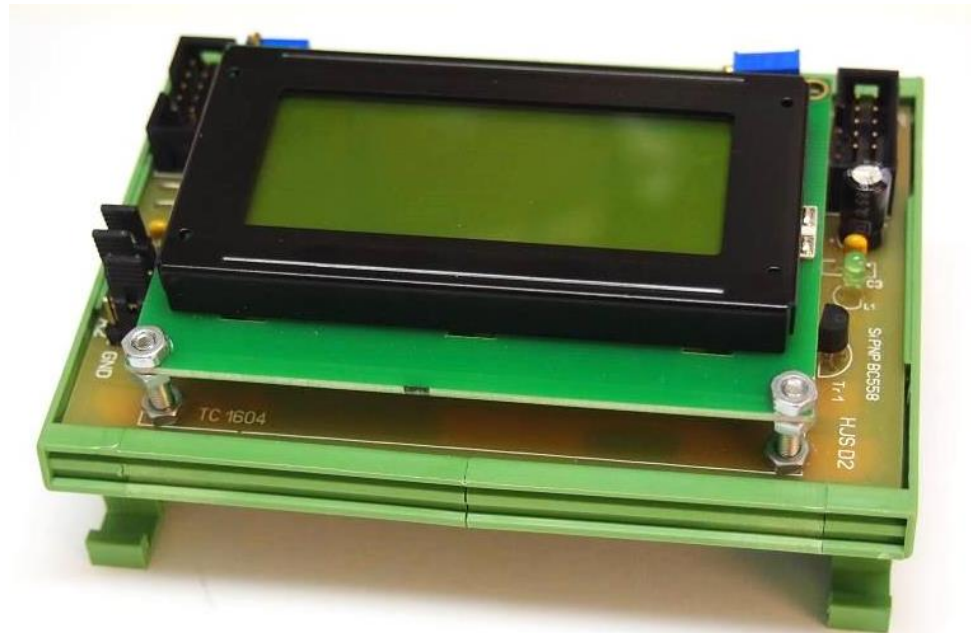
Bestimmungsgemäße Verwendung

- 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.

Display 2 - Sonderzeichen

Es ist relativ leicht einen Text auf dem Display auszugeben. Dazu gibt es Zeichensätze, die vom Hersteller bereits „eingebaut“ sind.

Wie kann ich aber eigene Zeichensätze oder spezielle Sonderzeichen darstellen?
Wie bekomme ich das Eurozeichen auf das Display?

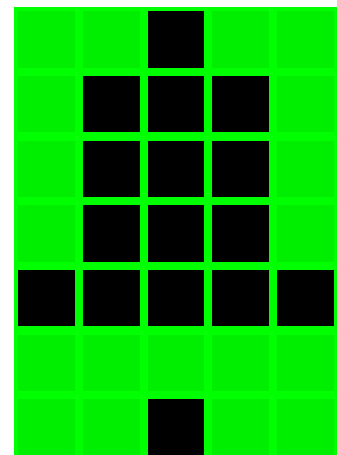


BM Display 1

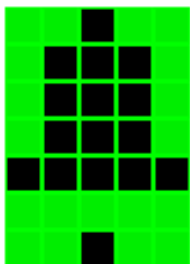
In diesem Teil wollen wir uns einmal ansehen, wie wir eigene Zeichen erstellen und auf unserem Display ausgeben können. Was ist denn alles möglich? Im Grunde fast alles, solange es in einem Rahmen von 5 x 8 (7) Zeichen darstellbar ist.

Auf unserem Display befinden sich 4 Zeilen mit jeweils 16 Zeichen. Jedes Zeichen wird aus einer bestimmten Anzahl von Quadraten gebildet. Wird ein Quadrat angesteuert, erscheint es auf dem Display dunkel und ist für uns sichtbar.

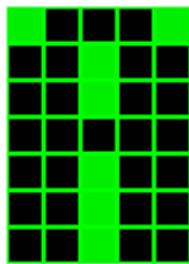
Zeichen bestehend aus 5x7 Quadraten (35 Stück)



Das können z.B. eine



Glocke



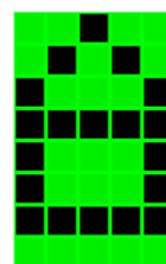
Buchstabe



Herz



Sendeturm



Haus sein.

Sehen wir und dazu einmal den Zeichensatz eines Displays mit dem Controller HD44780 an. Man kann dabei 3 verschiedene Bereiche erkennen. Auf der rechten Seite befindet sich der Bereich mit chinesischen- (?) und Sonderzeichen. In der Mitte (ca.) befinden sich die lateinischen Schriftzeichen. Da vor befindet sich ein freier Bereich. Bei einigen Controllern wird dieser Bereich erst gar nicht angegeben. An der oberen und linken Kante sind die Adressen angegeben.

	Lower 4 Bits	Upper 4 Bits	0000	0001	0010	0011	0100	0101	0110	0111	1000	1001	1010	1011	1100	1101	1110	1111
Adressen	xxxx0000	CG RAM (1)			0	1	P	`	P				-	9	3	0	P	
freier Bereich	xxxx0001	(2)	!	1	A	Q	a	4					o	A	7	4	ä	q
Lateinische Schriftzeichen	xxxx0010	(3)	"	2	B	R	b	r					r	i	u	x	ß	ø
Chinesische (?) Schriftzeichen Sonderzeichen	xxxx0011	(4)	#	3	C	S	c	s					u	u	t	e	ε	ω
	xxxx0100	(5)	\$	4	D	T	d	t					v	i	t	þ	μ	Ω
	xxxx0101	(6)	%	5	E	U	e	u					o	o	1	1	ö	ü
	xxxx0110	(7)	&	6	F	V	f	v					o	o	2	2	ρ	Σ
	xxxx0111	(8)	'	7	G	W	g	w					o	o	3	3	q	π
	xxxx1000	(1)	(8	H	X	h	x					i	o	4	4	r	×
	xxxx1001	(2))	9	I	Y	i	y					o	o	5	5	'	y
	xxxx1010	(3)	*	:	J	Z	j	z					e	o	6	6	j	7
	xxxx1011	(4)	+	;	K	[k	{					o	o	7	7	*	7
	xxxx1100	(5)	,	<	L	¥	l	l					o	o	8	8	φ	π
	xxxx1101	(6)	-	=	M]	m	}					u	u	9	9	ε	÷
	xxxx1110	(7)	.	>	N	^	n	‡					o	o	0	0	ñ	
	xxxx1111	(8)	/	?	O	_	o	+					u	u	1	1	ö	█

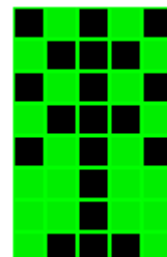
Uns interessieren nur die ersten 8 Adressen, da diese frei sind und von uns genutzt werden können. Es sind also 8 Sonderzeichen darstellbar.

In Abhängigkeit vom Displaytyp sind zwei verschiedene Darstellungen möglich. Es gibt Felder mit 5 x 7 Zeichen und Felder 5 x 8 Zeichen

Feld mit 5 x 7 Zeichen



Feld mit 5 x 8 Zeichen



An einem einfachen Beispiel möchte ich es einmal zeigen.

CGRAM-Adresse	16	8	4	2	1	Byte dezimal	Byte hexadezimal
0x00						4	0x04
0x01						8+2 = 10	0x0a
0x02						16+1 = 17	0x11
0x03						16+8+4+2+1 = 31	0x1f
0x04						16+1 = 17	0x11
0x05						16+1 = 17	0x11
0x06						16+8+4+2+1 = 31	0x1f
0x07						0	0x00

Dazu nutze ich das Sonderzeichen „Haus“. Unser Feld besteht in der waagerechten aus 5 Zeichen und in der senkrechten aus 8 Zeichen. Damit haben wir 40 Zeichen zur Verfügung. In diesem Raster kann ich mein Zeichen erstellen. Ich kann jedes gewählte Feld mit einem Kreuz kennzeichnen oder einfach schwarz hinterlegen. Als Beispiel habe ich einmal die Form eines Hauses eingezeichnet. Oberhalb dieser Felder sind die einzelnen Bit angegeben. Durch eine einfache Addition und Umrechnung in Hex bekomme ich meine Werte. Damit hat das „Haus“ die folgenden Werte:

0x04, 0x0a, 0x11, 0x1f, 0x11, 0x11, 0x1f, 0x00

Diese Daten müssen wir an unser Display übertragen und darstellen. Es sind auch andere Symbole möglich. Ein paar Beispiele:

```

0x00, 0x0A, 0x1B, 0x1B, 0x18, 0x1F, 0x0E, 0x00, // Uhrensymb. 3 Uhr
0x00, 0x0A, 0x1B, 0x1B, 0x03, 0x1F, 0x0E, 0x00, // Uhrensymb. 9 Uhr
0x04, 0x0e, 0x0e, 0x0e, 0x1f, 0x00, 0x04, 0x00 // Glocke
0x01, 0x03, 0x0f, 0x0f, 0x0f, 0x03, 0x01, 0x00 // Lautsprecher
0x00, 0x0f, 0x0e, 0x04, 0x04, 0x0e, 0x1f, 0x00 // Eieruhr senk.
0x00, 0x11, 0x13, 0x1f, 0x1b, 0x11, 0x00, 0x00 // Eieruhr waag.
    
```

Es handelt sich hierbei um Vorschläge von mir, zur Anregung einmal selber zu testen und eigene Zeichen zu erstellen.

Die notwendigen Einstellungen habe ich bereits im ersten Teil beschrieben.

Nochmal für alle. Verändert nicht willkürlich die Einstellungen. Notiert euch die Einstellungen. Aus Erfahrung muss ich sagen, falsche Einstellungen können zu einem kompletten Absturz vom AVR-Studio führen oder hunderte von Fehlermeldungen verursachen. Unter Umständen hilft nur eine Neuinstallationen der gesamten Software.

Sehen wir uns als nächste das Programm an. Ich habe versucht verschiedene Möglichkeiten der Darstellung zu nutzen. Hinter den Programmzeilen befindet sich der entsprechende Kommentar.

Das Programm habe ich in verschiedene Bereiche aufgeteilt. Jeder Bereich soll verschiedene Möglichkeiten der Darstellung demonstrieren. Teilweise sind die Möglichkeiten miteinander kombiniert. Im letzten Bereich habe ich den „Robox“ als Sonderzeichen erschaffen und ein wenig zum Leben erweckt.


```

/* ATB_B1_Display_2.c Created: 18.10.2014 11:38:10 Author: AS */

#include <stdbool.h>           // AT1284p, 16 MHz, I2C Bus
#include <stdint.h>
#include <stdlib.h>
#include <avr/pgmspace.h>
#include "main.h"
#include <util/delay.h>
#include "i2clcd.h"           // Dateien von Falk
#include "i2cmaster.h"

// string_flasch
const char string_flash1[] PROGMEM = "Tut Display 2"; // definiert string_flash1
const char string_flash2[] PROGMEM = "Boxtec";       // definiert string_flash2
const char string_flash3[] PROGMEM = "by achim S.";  // definiert string_flash3

char string1[] = "Hallo Gemeinde";                  // definiert string1 "Hallo Gemeinde"
char string2[] = "Diese Programm..";                // definiert string2 "Dieses Programm"
char string3[] = "testet Anzeige";                  // definiert string3 "testet Anzeige"

uint8_t x;                                          // definiert x

// Sonderzeichen
const char SonderZ1[] PROGMEM = {0x06, 0x09, 0x10, 0x0E, 0x10, 0x09, 0x06, 0x00}; // Euro
const char SonderZ2[] PROGMEM = {0x04, 0x0a, 0x11, 0x1f, 0x11, 0x11, 0x1f, 0x00}; // Haus
const char SonderZ3[] PROGMEM = {0x00, 0x0A, 0x1B, 0x1B, 0x18, 0x1F, 0x0E, 0x00}; // Uhr 3
const char SonderZ4[] PROGMEM = {0x00, 0x0A, 0x1B, 0x1B, 0x03, 0x1F, 0x0E, 0x00}; // Uhr 9
const char SonderZ5[] PROGMEM = {0x15, 0x0e, 0x15, 0x0e, 0x04, 0x04, 0x04, 0x0e}; // Turm
const char SonderZ6[] PROGMEM = {0x0e, 0x0e, 0x04, 0x05, 0x0e, 0x1a, 0x0e, 0x11}; // Ro 1 li
const char SonderZ7[] PROGMEM = {0x0e, 0x0e, 0x04, 0x04, 0x1f, 0x0a, 0x0e, 0x11}; // Ro 1 mi
const char SonderZ8[] PROGMEM = {0x0e, 0x0e, 0x04, 0x14, 0x0e, 0x0b, 0x0e, 0x11}; // Ro 1 re

void Zeichen_def()
{
    // Definiert Sonderzeichen
    lcd_def_char(SonderZ1, 1);
    lcd_def_char(SonderZ2, 2);
    lcd_def_char(SonderZ3, 3);
    lcd_def_char(SonderZ4, 4);
    lcd_def_char(SonderZ5, 5);
    lcd_def_char(SonderZ6, 6);
    lcd_def_char(SonderZ7, 7);
    lcd_def_char(SonderZ8, 8);
};

int main(void)
{
    i2c_init();           // start init i2c
    lcd_init();           // start init lcd

    // Display ein/aus, Cursor ein/aus, Blinken ein/aus
    lcd_command(LCD_DISPLAYON | LCD_CURSOROFF | LCD_BLINKINGOFF);
    lcd_light(0);        // Licht an

    while (1)
    {
        // Bildschirm 1 - Tut, boxtec, achim
    }
}

```

```
// *****
lcd_command(LCD_CLEAR);           // LCD löschen
_delay_ms(2);                     // Pause
lcd_print_P(string_flash2);       // Ausgabe string_flash Zeile 3 Pos 1
lcd_printlc_P(3, 2, string_flash1); // Ausgabe string_flash Zeile 2 Pos 2
lcd_printlc_P(2, 1, PSTR("*****")); // Ausgabe *
lcd_printlc_P(4, 2, string_flash3); // Ausgabe string_flash Zeile 4 Pos 2
_delay_ms(5000);                  // Pause

// Bildschirm 2 - Anzeige Funktion
// *****
lcd_command(LCD_CLEAR);           // LCD löschen
_delay_ms(2);                     // Pause
lcd_print(string1);                // Ausgabe "Hallo Gemeinde"
lcd_nextline();                   // nächste Linie
lcd_printlc_P(2, 1, PSTR("*****")); // Ausgabe *
lcd_gotohc(3,1);
lcd_print(string2);                // Ausgabe Text
lcd_gotohc(4,2);
lcd_print(string3);                // Ausgabe Text
_delay_ms(5000);                  // Pause

Zeichen_def();                    // Definiert Sonderzeichen

// Bildschirm 3 - Anzeige Sonderzeichen
// *****
lcd_command(LCD_CLEAR);           // LCD löschen
_delay_ms(2);                     // Pause
lcd_printlc_P(1, 1, PSTR("Ausgabe")); // Ausgabe Text
lcd_printlc_P(2, 1, PSTR("Sonderzeichen")); // Ausgabe Text
lcd_printlc_P(3, 1, PSTR("*****"));
// Anzeige der Sonderzeichen
lcd_printlc(4,1,("\x1"" \x2"" \x3"" \x4"" \x5"" \x6"" \x7"" \x8"));
_delay_ms(5000);                  // Pause

// Bildschirm 4 - Sonderzeichen bewegen
// *****
lcd_command(LCD_CLEAR);           // LCD löschen
_delay_ms(2);                     // Pause
lcd_printlc_P(1, 1, PSTR("Sonderzeichen")); // Ausgabe Text
lcd_printlc_P(2, 1, PSTR("*****")); // Ausgabe *
// Anzeige der Sonderzeichen
lcd_printlc(4,1,("\x1"" \x2"" \x3"" \x4"" \x5"" \x6"" \x7"" \x8"));
lcd_printlc(3,1,("\x6""\x7""\x8")); // Sonderzeichen 6, 7, 8
_delay_ms(400);                   // Pause
lcd_printlc(3,1,("\x8""\x6""\x7")); // Sonderzeichen 8, 6, 7
_delay_ms(400);                   // Pause
lcd_printlc(3,x,("\x7""\x8""\x6")); // Sonderzeichen 7, 8, 6
_delay_ms(1000);                  // Pause

for (x=1;x<=16;x++)
{
    lcd_printlc(3,x,("\x7""\x8""\x6")); // Sonderzeichen 7, 8, 6
}
```

```
    _delay_ms(400);                // Pause
    lcd_printlc(3,x,("\x8""\x6""\x7")); // Sonderzeichen 8, 6, 7
    _delay_ms(400);                // Pause
    lcd_printlc(3,x,("\x6""\x7""\x8")); // Sonderzeichen 6, 7, 8
    _delay_ms(400);                // Pause
    lcd_printlc(3,x,("\x7""\x8""\x6")); // Sonderzeichen 7, 8, 6
    _delay_ms(400);                // Pause
    lcd_printlc(3,x,("\x8""\x6""\x7")); // Sonderzeichen 8, 6, 7
    _delay_ms(400);                // Pause
    lcd_printlc(3,x,("\x6""\x7""\x8")); // Sonderzeichen 6, 7, 8
    _delay_ms(400);                // Pause
  }
  _delay_ms(2000);                // Pause
}
}
```

In der Anlage zu diesem Tut habe ich auch die notwendigen Dateien gepackt. Diese Dateien sind speziell von Falk angepasst worden. Ohne diese Änderungen ist die Darstellung der Sonderzeichen nicht möglich. **Bitte nicht mit anderen Dateien verwechseln !!!**

Andere Displays mit z.B. 2x16 oder 4x20 Zeichen habe ich nicht getestet.

Der Aufbau und Test erfolgte im System 72 mit den Modulen Board1, Display1 und NT2.

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