

MIKROKONTROLLER & I²C BUS

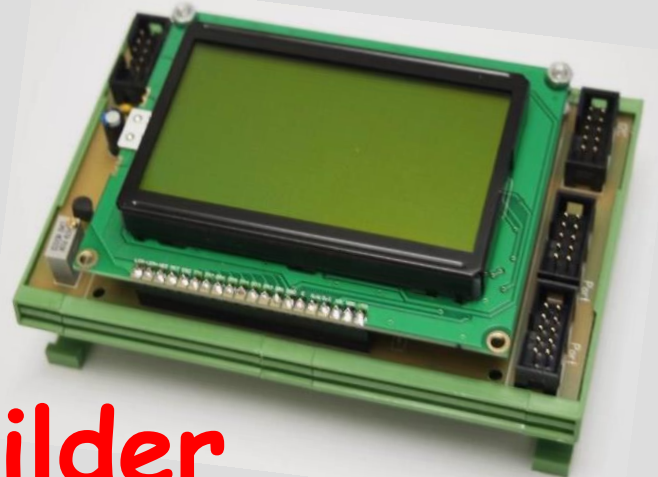


by AS

www.boxtec.ch

playground.boxtec.ch/doku.php/tutorial

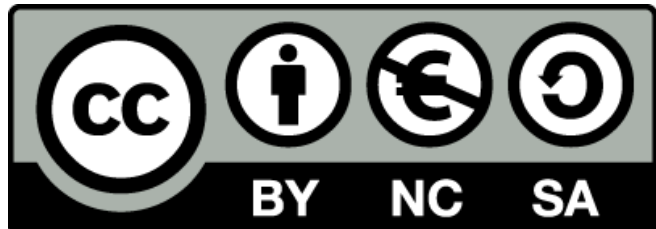
Graphik Display (128x64)
mit dem KS0108, 2 Ports
und 2 x I²C Verbindung



Graphik 3 - Bilder

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.

Graphik 3 - Bilder

In den beiden letzten Teilen haben wir unser Graphikdisplay an unser Board 1 angeschlossen, haben verschiedene Schriften und einfache Zeichen auf dem Display dargestellt.

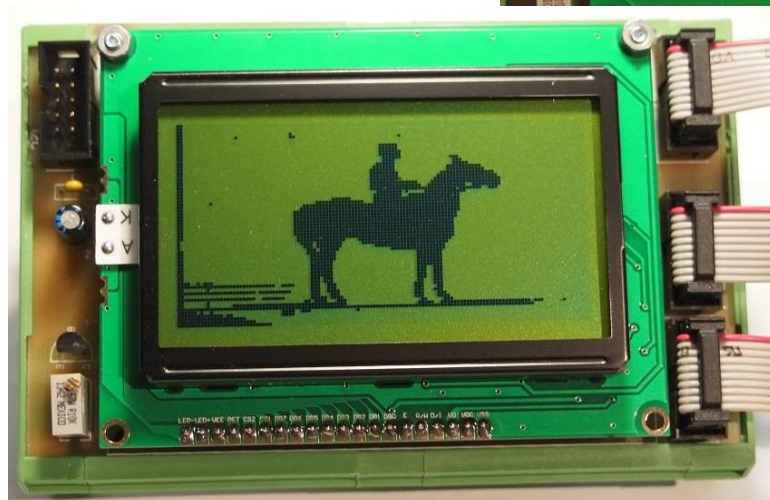
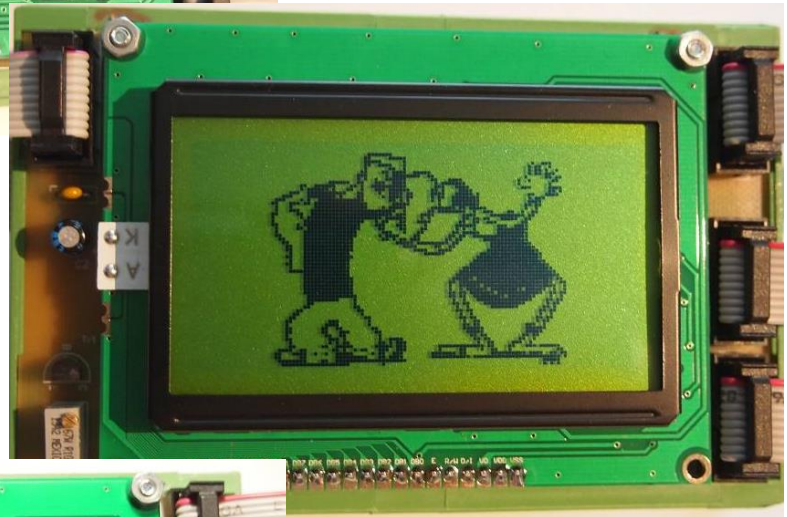
Startanzeige



Verschiedene Schriften

In diesem Teil wollen wir ein Bild auf das Display bringen.

Display mit Bild

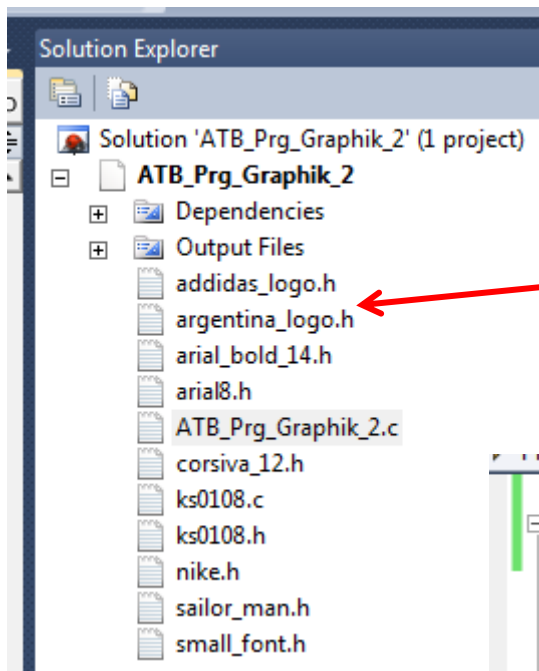


Es ist sogar eine Art „Bewegungsablauf“ darstellbar.

Laufendes Pferd

Fangen wir mit einigen Bildern an, die ich im Netz gefunden habe. Im Beispielprogramm, (weiter unten) habe ich verschiedene Dateien eingebunden. Die einzelnen Dateien müssen wieder in unser Programm eingebunden werden.

Angabe der notwendigen Dateien
in unserem Programm

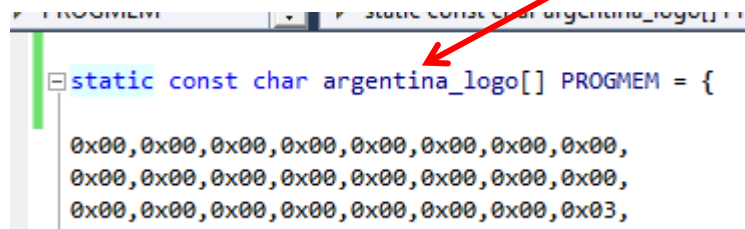


```
#define F_CPU 16000000UL
#include <inttypes.h>
#include <avr/io.h>
#include <avr/pgmspace.h>
#include <util/delay.h>
|
#include "ks0108.h"
#include "arial_bold_14.h"
#include "corsiva_12.h"
#include "arial8.h"
#include "small_font.h"

#include "addidas_logo.h"
#include "argentina_logo.h"
#include "sailor_man.h"
#include "nike.h"
```

Diese Dateien müssen wir den Solution Explorer
laden

Bitte die erste Zeile beachten !



Es steht jedem frei, eigene Bilder, Zeichnungen oder Schriften zu entwerfen und in seinem Programm zu nutzen. Bitte dabei immer an das notwendige Format halten.

```
/* ATB_Prg_Graphik_2.c Created: 06.04.2015 14:37:28 Author: AS */
```

```
#define F_CPU 16000000UL
#include <inttypes.h>
#include <avr/io.h>
#include <avr/pgmspace.h>
#include <util/delay.h>
```

```
#include "ks0108.h"
#include "arial_bold_14.h"
#include "corsiva_12.h"
#include "arial8.h"
#include "small_font.h"
```

```
#include "addidas_logo.h"
#include "argentina_logo.h"
#include "sailor_man.h"
#include "nike.h"
```

```
void LoadBitmap(unsigned char *bitmap, uint8_t size_x, uint8_t size_y)
{
    uint16_t i, j, by;
```

```

for(i=0; i<size_x; i+=8)
for(j=0; j<size_y; j++)
{
    by=pgm_read_byte(bitmap++);
    ks0108GotoXY(j, i);
    ks0108WriteData(by);
}
}

int main(void)
{
    DDRD=0b00001000;           // Angabe Pin am AT1284p
    //PORTD &=~(1<<PD3);      // LED aus
    PORTD |= (1<<PD3);        // LED Licht an
    ks0108Init(0);            // Initialize GLCD
    _delay_ms(5);             // Pause
    // Ausgabe Rand
    ks0108DrawRoundRect(1, 2, 125, 61, 8, BLACK);
    //Ausgabe Linie
    ks0108DrawLine(110, 32, 67, 60, BLACK);
    //Ausgabe Viereck leer
    ks0108DrawRect(12, 37, 10, 20, BLACK);
    //Ausgabe Viereck gefüllt
    ks0108FillRect(40, 50, 10, 10, BLACK);
    // Ausgabe Kreis
    ks0108Circle(110, 50, 10, BLACK);
    // Auswahl Schrift
    ks0108SelectFont(Arial_Bold_14, ks0108ReadFontData, BLACK);
    ks0108GotoXY(14,5);       // Angabe Position
    // Ausgabe Text
    ks0108Puts_P(PSTR("Display - Modul\nmit dem KS0108"));
    _delay_ms(1000);         // Pause
    ks0108SelectFont(small_font, ks0108ReadFontData, BLACK);
    ks0108GotoXY(35,35);     //Angabe Position
    ks0108Puts_P(PSTR("Bilder")); // Ausgabe Text
    _delay_ms(5000);        // Pause
    ks0108ClearScreen();    // Display löschen
    _delay_ms(1000);       // Pause
    // Bild addidas
    LoadBitmap(addidas_logo, 64, 128);
    _delay_ms(5000);
    // Bild argentina
    LoadBitmap(argentina_logo, 64, 128);
    _delay_ms(5000);
    // Bild nike
    LoadBitmap(nike, 64, 128);
    _delay_ms(5000);
    // Bild Sailor
    LoadBitmap(sailor_man, 64, 128);
    _delay_ms(5000);
    while(1);
}

```

Ein paar Erklärungen zum Programm. Einige der Teile bzw. Anweisungen habe ich bereits im letzten Teil erklärt.

```
void LoadBitmap(unsigned char *bitmap, uint8_t size_x, uint8_t size_y)
{
    uint16_t i, j, by;
    for(i=0; i<size_x; i+=8)
    for(j=0; j<size_y; j++)
    {
        by=pgm_read_byte(bitmap++);
        ks0108GotoXY(j, i);
        ks0108WriteData(by);
    }
}
```

Mit diesem Unterprogramm laden wir die eigentliche Datei und bringen sie auf das Display. Dabei wird mit `by=pgm_read_byte(bitmap++)`; der Name unserer ausgewählten Datei angegeben, und mit `ks0108GotoXY(j, i)`; die genaue Position angegeben.

```
LoadBitmap(argentina_logo, 64, 128);
```

Mit dieser Anweisung rufen wir das Unterprogramm (**LoadBitmap**) auf, teilen ihm mit, welche Datei (**argentina_logo**) geladen werden soll und wie viele Zeilen und Spalten (**64, 128**) sie hat.

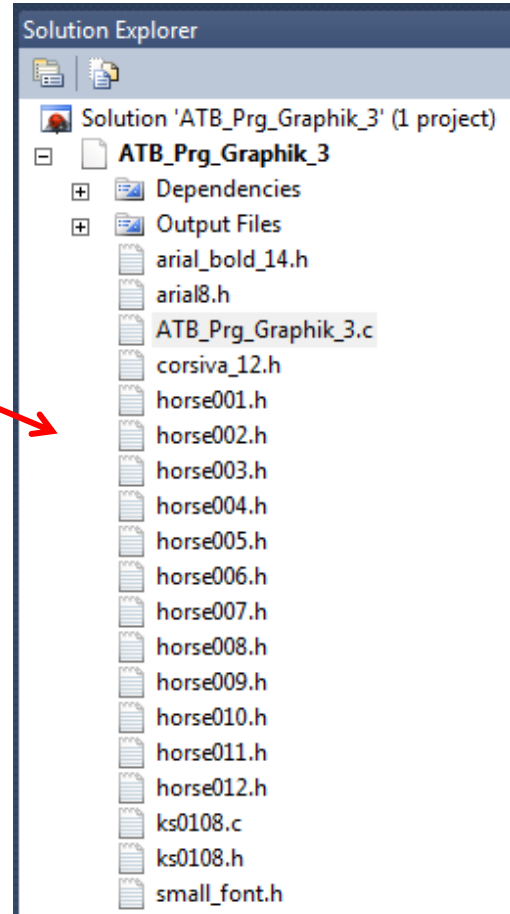
Im Programm **ATB_Prg_Graphik_2.c** verwende ich die folgenden Bilder:

- **addidas_logo**
- **argentina_logo**
- **nike**
- **sailor_man**

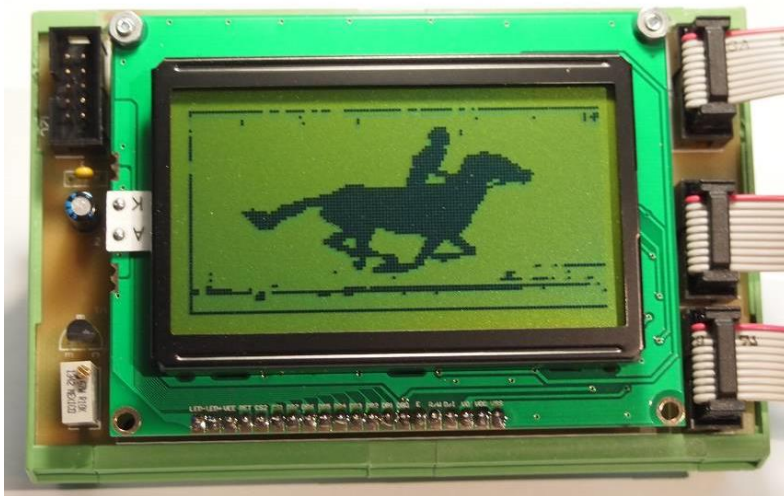
Im Programm **ATB_Prg_Graphik_3.c** müssen die folgenden Bilder eingebunden werden:

- **horse001**
- **horse002**
- **horse003**
- **horse004**
- **horse005**
- **horse006**
- **horse007**
- **horse008**
- **horse009**
- **horse010**
- **horse011**
- **horse012**

Für beide Programme werden auch die verschiedenen Schriftarten benötigt.



So könnten einige „Bewegungen“
aussehen



Der Aufbau der Hardware wurde nicht verändert

Fehlen eigentlich nur noch die entsprechenden bmp-Dateien für eigene Bilder oder Anwendungen.

Leider konnte ich bisher kein lauffähiges oder einfaches Programm zur Erstellung von BMP finden. Vielleicht kennt jemand so was.

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