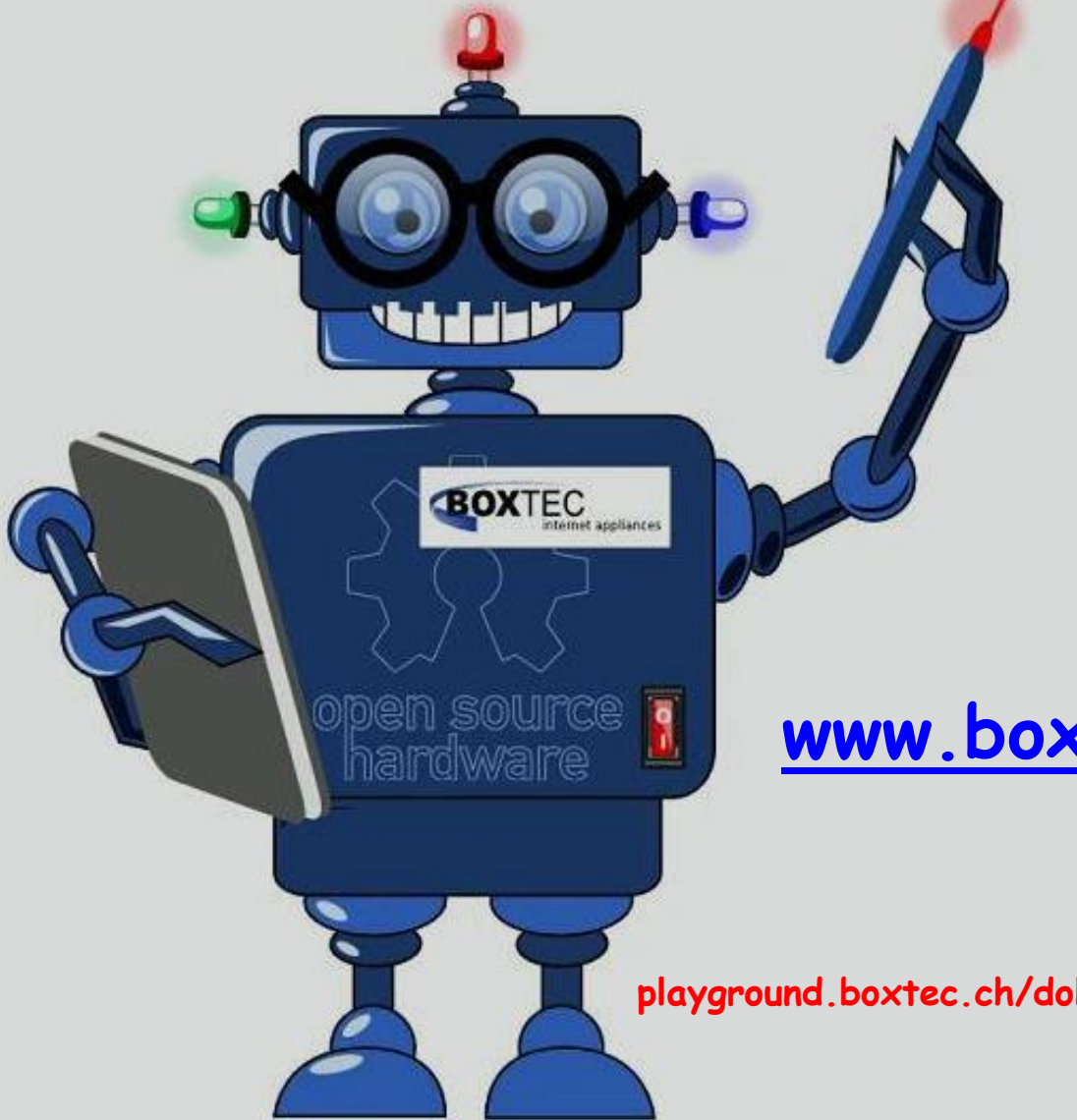


MIKROKONTROLLER & I²C BUS



www.boxtec.ch

playground.boxtec.ch/doku.php/tutorial



+



State Maschine 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.

State Maschine 2 (Sprünge ...)

Im ersten Teil haben wir uns mit der Theorie auseinander gesetzt. Hoffe, das es einige soweit verstanden haben.

Im zweiten Teil machen wir uns es einfacher. Der Aufbau der Hardware ist genau wie im ersten Teil. Die LED habe ich diesmal keiner Farbe zugeordnet.

Wir testen die Sache an einem Programm und wollen dabei mit den ersten Sprüngen anfangen. Sprünge sind dabei untertrieben. Wir lassen Teile des Programmes in Unterprogrammen ausführen. Jedes Unterprogramm arbeitet dabei für sich allein und verursacht keinerlei Verzögerung. Nach jedem Aufruf eines Unterprogrammes kehren wir zum Hauptprogramm zurück und rufen das nächste Unterprogramm auf. Die einzige Verzögerung ist im Hauptprogramm und beträgt 1 ms. Das Programm habe ich bereits in einem anderen Teil verwendet.

```
/* ATB_Stm_2.c Created: 20.08.2014 11:10:28 Author: AS */
```

```
#define F_CPU 16000000UL // Angabe der Quarzfrequenz, wichtig für die Zeit
#include <util/delay.h> // Einbindung Datei Pause
#include <avr/io.h> // Einbindung Datei Ausgänge
```

```
int16_t led1=0;
int16_t led2=0;
```

```
void led_blinken1()
{
    led1++;
    if(led1==1999)
        PORTA &= ~(1<<PA5); // Schaltet Pin
    else
    {
        if(led1==3999)
        {
            PORTA |= (1<<PA5); // Schaltet Pin
            led1=0;
        }
    }
}
```

```
void led_blinken2()
{
    led2++;
    if(led2==599)
        PORTA &= ~(1<<PA6); // Schaltet Pin
    else
    {
        if(led2==1199)
        {
            PORTA |= (1<<PA6); // Schaltet Pin
            led2=0;
        }
    }
}
```

```

    }
}
int main(void)
{
    DDRA=0b01100000;           // Port A auf Ausgang schalten
    while(1)                   // Programmschleife
    {
        led_blinken1();       // Aufruf Unterprogramm 1
        led_blinken2();       // Aufruf Unterprogramm 2
        _delay_ms(1);
    }
}

```

Sehen wir uns das Programm im Einzelnen an: (angefangen von hinten)

```

while(1)                       // Programmschleife
{
    led_blinken1();           // Aufruf Unterprogramm 1
    led_blinken2();           // Aufruf Unterprogramm 2
    _delay_ms(1);
}

```

In unserem Hauptprogramm rufen wir nacheinander unsere Unterprogramme `led_blinken1` und `led_blinken2` auf. Gefolgt von einer Pause von 1ms. Diese ist notwendig, um eine definierte Durchlaufzeit zu haben.

```

int16_t led1=0;
int16_t led2=0;

```

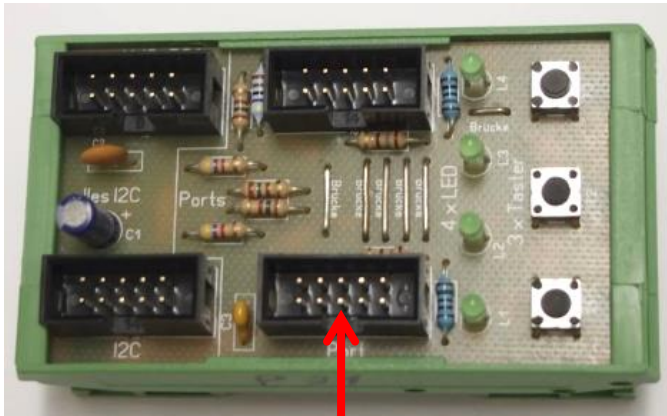
wir definieren zwei Variable und setzen sie auf 0

```

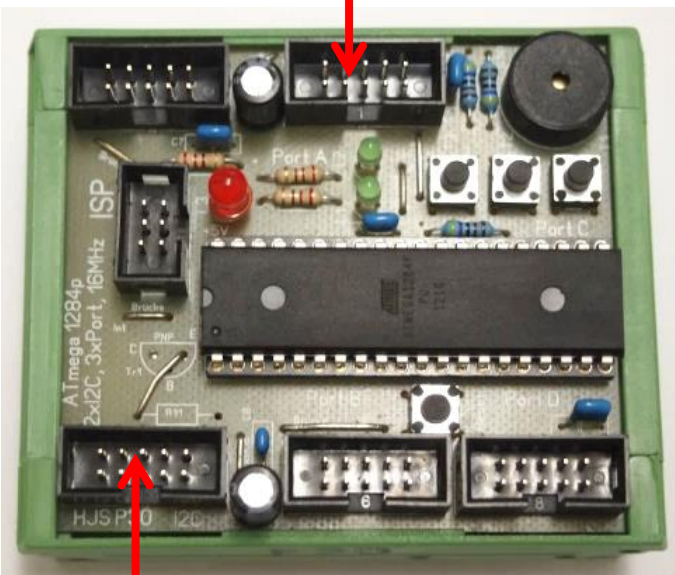
void led_blinken1()
{
    led1++;
    if(led1==1999)
        PORTA &= ~(1<<PA5);           // Schaltet Pin
    else
    {
        if(led1==3999)
        {
            PORTA |= (1<<PA5);         // Schaltet Pin
            led1=0;
        }
    }
}

```

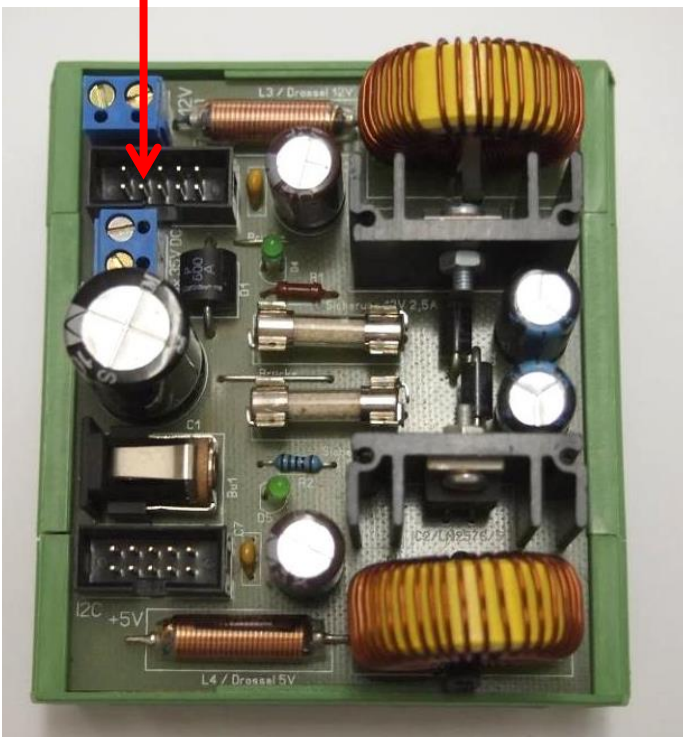
Bei jedem Durchlauf unseres Hauptprogrammes wird das Unterprogramm `void led_blinken1()` aufgerufen. Dabei wird die Variable `led1` jeweils um 1 erhöht. Hat `led1` der Wert von `1999` erreicht, wird `PA5` geschaltet. `led1` wird weiter bei jedem Durchlauf um 1 erhöht und erreicht den Wert von `3999`. Es wird wiederum `PA5` geschaltet und `led1` auf `0` gesetzt.



BPM In / Out 1



BM Board 1



BM NT 2

Die Verbindung der einzelnen Module erfolgt an den roten Linien.

Leider hat unser Programm noch einige Nachteile, z.B.

- Aus einem Unterprogramm mit **void** kann keine Rückgabe eines Wertes erfolgen
- Dadurch kann keine Auswahl stattfinden
- Das Programm ist sehr unflexibel
- Es ist zwar Multitasking fähig, mehr aber nicht

Bleibt die Frage, wie können wir es vernünftig ändern.

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