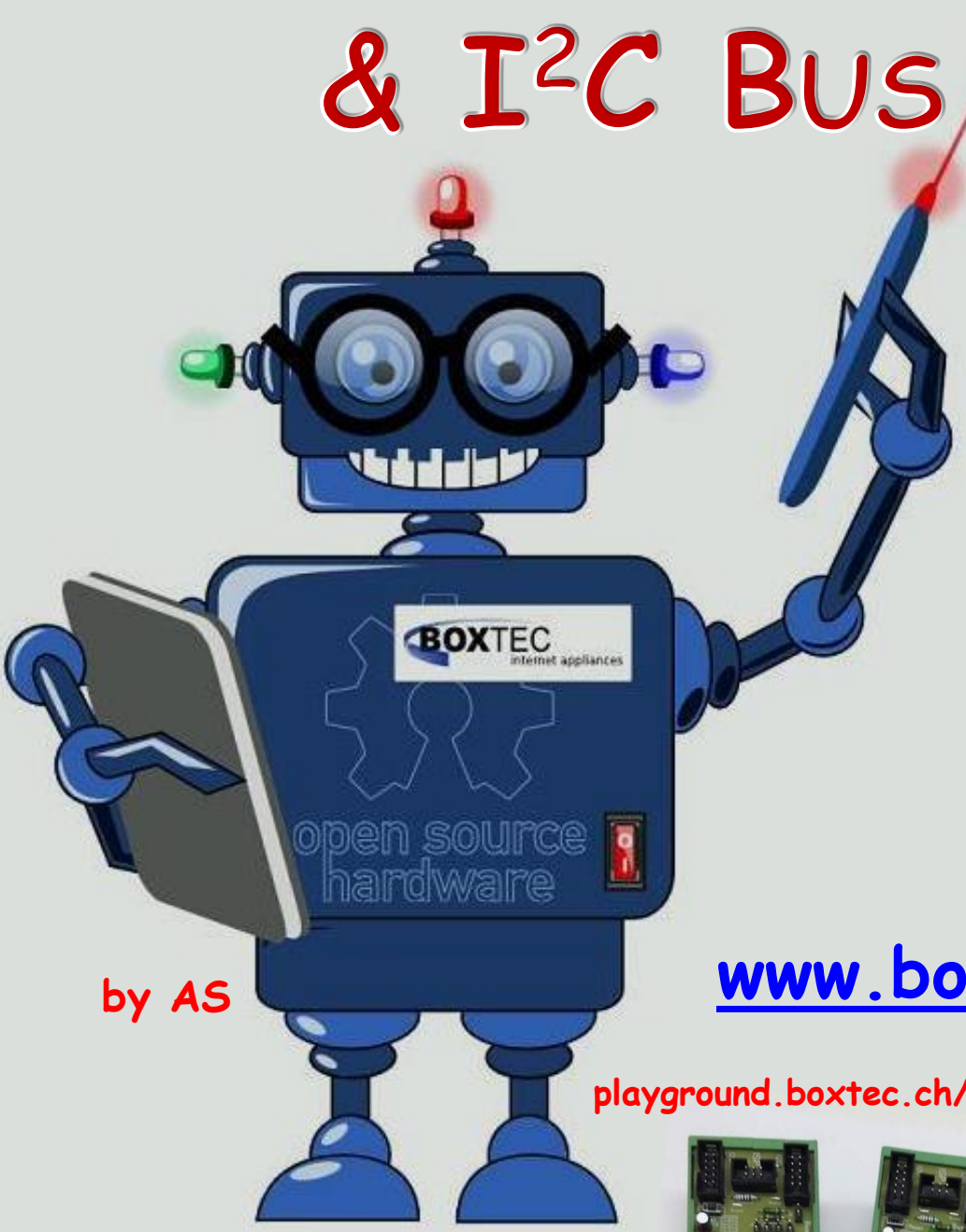
Mikrokontroller & I²C Bus



by AS

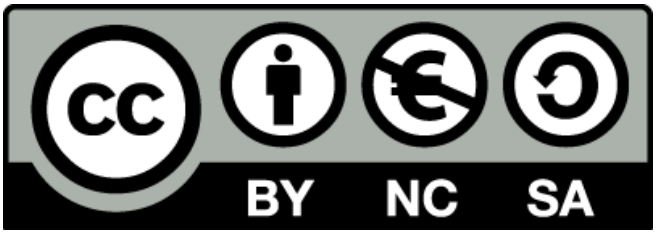
www.boxtec.ch

playground.boxtec.ch/doku.php/tutorial

Attiny 841 – ein IC und
5 verschiedene Module
Teil 2 – Software 1

I²C Bus und der Attiny 841

Copyright



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 Nichtbe-achtung 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:

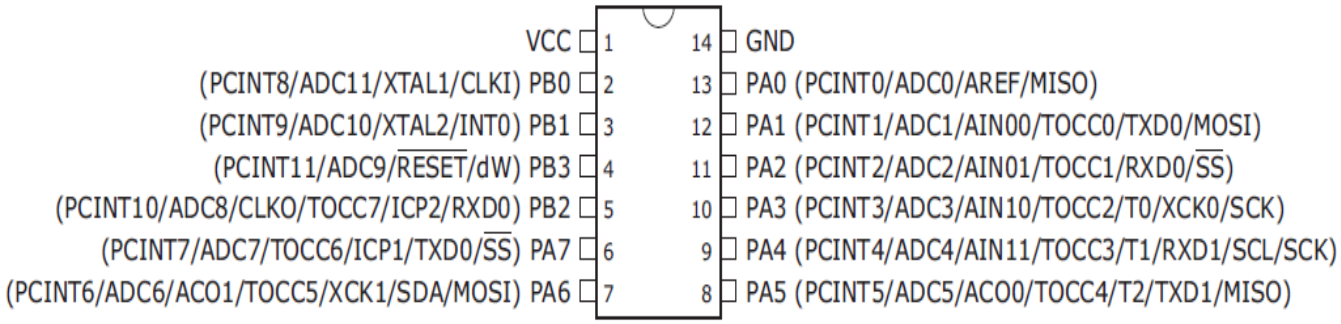
- Nehmen Sie diesen Bausatz nur dann in Betrieb, wenn er zuvor berührungssicher in ein Gehäuse eingebaut wurde. Erst danach darf dieser an eine Spannungsversorgung angeschlossen werden.
- Lassen Sie Geräte, die mit einer Versorgungsspannung größer als 24 V- betrieben werden, nur durch eine fachkundige 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 eine 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 Versor-gungsspannung 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 Spannungsver-sorgung 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 unver-züglich 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 Ge-fahren, 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.

Attiny 841 – 5 Module – Software 1 (In/Out)

Beim ATtiny841 handelt es sich um einen der modernsten AVR-Mikrocontroller. Er wird in einem 14 poligen SMD-Gehäuse geliefert. Dadurch kann er direkt auf die Leiterzüge montiert werden. Bitte SMD bei der Montage unbedingt beachten.



Auf Grund der beschränken Anzahl von Beinen sind viel Pins mehrfach belegt.

Einige Daten aus dem Datenblatt des Herstellers:

Merkmale der Peripherie

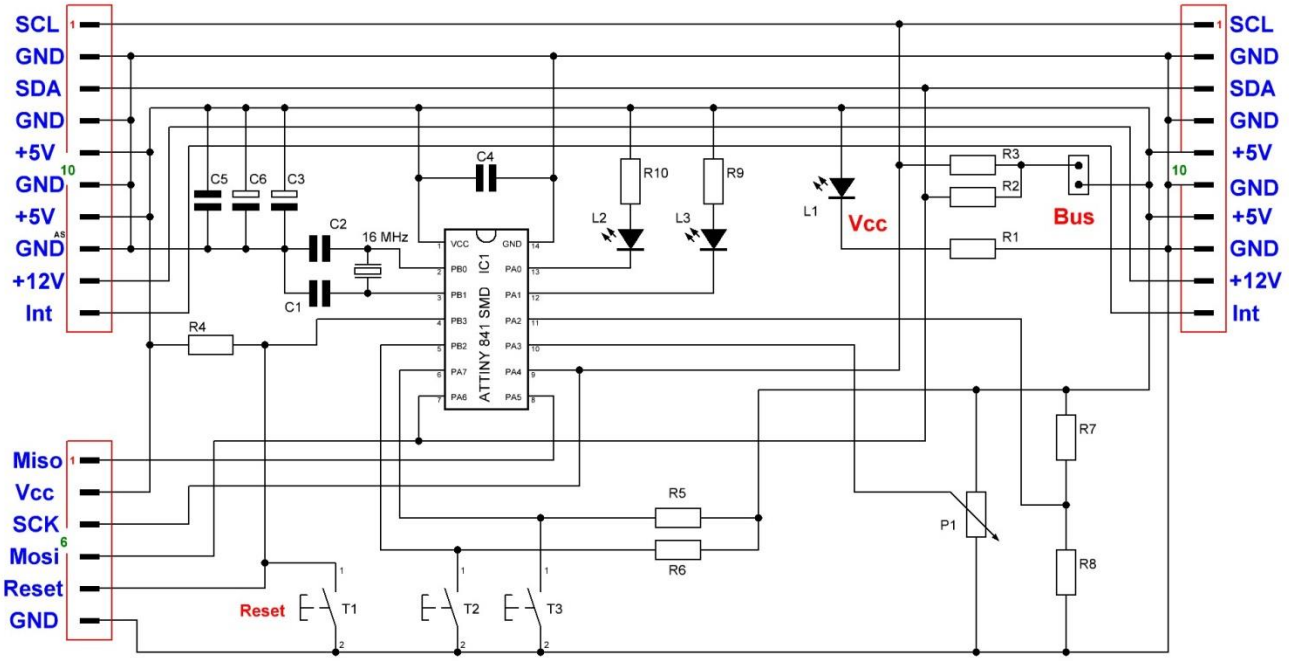
- Ein 8-Bit- und zwei 16-Bit-Timer/Zähler mit je zwei PWM-Kanälen
- Programmierbarer Watchdog-Timer mit extrem niedriger Leistung
- 10-Bit-Analog-Digital-Wandler
- 12 externe und 5 interne, unsymmetrische Eingangskanäle
- 46 differentielle ADC-Kanalpaare mit programmierbarer Verstärkung (1x / 20x / 100x)
- Zwei analoge On-Chip-Komparatoren
- Zwei Full Duplex USARTs mit Startbilderkennung
- Master/Slave SPI Serielle Schnittstelle
- Slave I2C Serielle Schnittstelle

Besondere Merkmale des Mikrocontrollers

- Leerlauf-, ADC-Rauschunterdrückungs-, Standby- und Abschaltmodi mit niedriger Leistung
- Verbesserte Einschalt-Rücksetzschaltung
- Programmierbare Ausbrecherkennungsschaltung mit Netzspannungsabtastung
- Externe und interne Interruptquellen
- Pinwechselunterbrechung an 12 Pins
- Kalibrierter 8MHz Oszillator mit Temperaturkalibrierung Option
- Kalibrierter 32kHz Ultra Low Power Oszillator mit extrem niedriger Leistung
- Hochstrom-Antriebsfähigkeit auf 2 I/O-Pins

Die Angaben aus dem Datenblatt habe ich mittels DeepL übersetzt. Dadurch können einige Angaben nicht korrekt sein.
In diesem Teil werde ich die einzelnen Schaltung und Belegung nochmals darstellen und mit kleinen Programmen in Betrieb nehmen.

An der Platine 173 werde ich die Funktonen der In/Out Pins erläutern. In diesem Teil werde ich nicht auf die AC/ADC bzw. Poti eingehen. Die Werte der Bauteile bitte dem Teil Hardware entnehmen.



Schaltung P173 mit Attiny 841 mit Quarz, 2 Taster, 2 LEDs, Poti und Ansicht Platine

Belegung:
PA0 – LED 2
PA1 – LED 3
PA2 – Vcc/2
PA3 – Regler
PB2 – Taster 2
PA7 – Taster 3

Funktion
Reset Taster
Anzeige Vcc
2 x Taster
2 x LEDs
Einstellung Spannung
U/2 Spannung
Mit Quarz
Vcc für den I^2C Bus schaltbar

Der Attiny 841 verfügt über 14 Pins, die verschieden genutzt werden können. Einige Pins haben einen festen Anschluss mit den folgenden Funktionen:

Versorgung	Quarz	Reset	AC/ADC	Taster	LED	I2C Bus
Pin 1 – Vcc	PB 0	PB 3	PA 2	PA 7	PA 0	PA 4
Pin 14 – GND	PB 1		PA 3	PB 2	PA 1	PA 6

Somit können für die Programmierung der In/Out Pins nur die PA0, PA1, PA7 und PB2 verwendet werden. Einige Pins werden zur Programmierung mit dem ISP verwendet. Weitere Programme werde ich für die Platinen 170 und 177 vorstellen. Auf die Adresswahl am I^2C Bus werde ich nicht eingehen.

Funktion des Programmes:

Mit dem Taster 2 oder 3 werden bei Betätigung die LEDs 2 oder 3 ein- oder ausgeschaltet

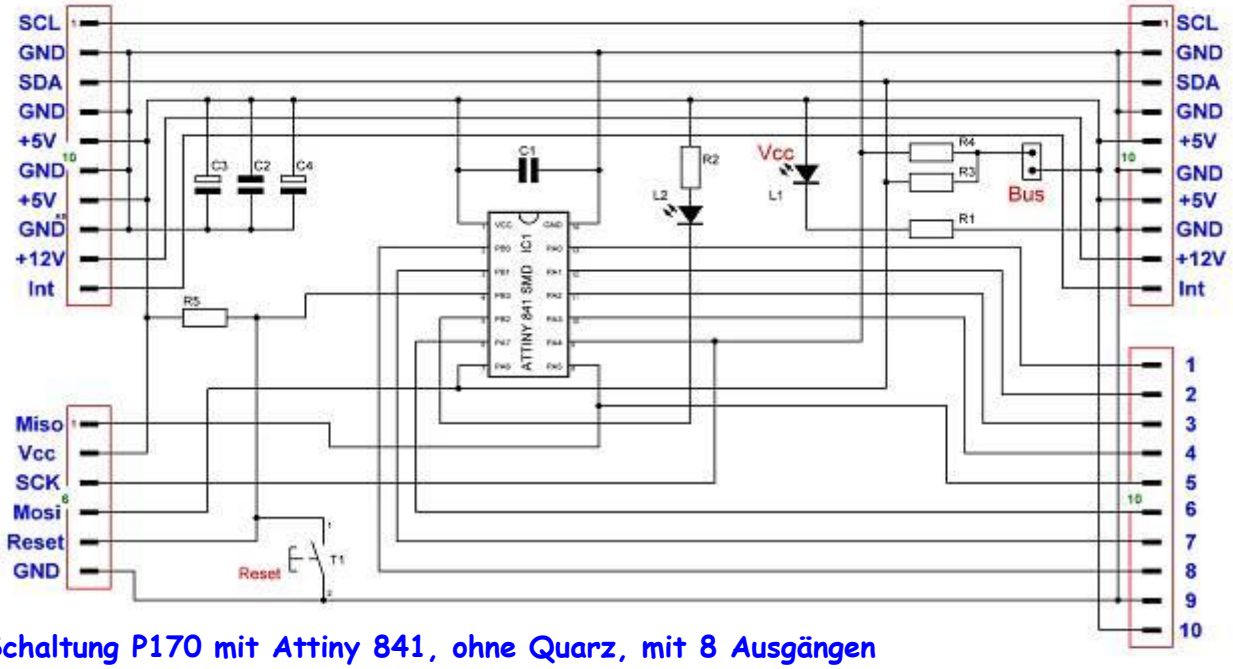
```
/* ATB_Ati_841_Prg_2.c  * Author : hjsee */
// Programm für Attiny 841 mit Quarz mit Platine 173, 2 Taster schalten 2 LEDs
// Platine P173 Attiny 841 mit Quarz

#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

int main(void)
 {
   DDRA=0b00000011;                     // DDRA auf Ein/Ausgang schalten
   DDRB=0b00000000;                     // DDRB auf Eingang schalten
   PORTA=0b00000011;                    // Port A auf aus/ein
   PORTB=0b00000000;                    // Port B auf aus

   while(1)                             // Programmschleife
    {
      if (PINB & (1<<PINB2))            // Taster T2
       {                                // Wenn T2 gedrückt...
         PORTA |=(1<<PINA0);            // LED 2 ein
       }
      else
       {                                // wenn nicht
         PORTA &=~(1<<PINA0);           // LED 2 aus
       }
      if (PINA & (1<<PINA7))            // Taster T3
       {                                // Wenn T3 gedrückt...
         PORTA |=(1<<PINA1);            // LED 3 ein
       }
      else
       {                                // wenn nicht
         PORTA &=~(1<<PINA1);           // LED 3 aus
       }
    }
 }
```

Einstellung der Fuse beim Attiny 841 mit Quarz:

Ex 0xFF
Hi 0xDF
Lo 0x8f

Achtung Ausgangsbeschaltung

Die Ausgänge des Attiny 841 sind unterschiedlich belastbar. Eine Überlastung der Pins kann zu einer sofortigen Zerstörung führen. Alle LEDs und Testboards habe ich daher für einen maximalen Strom von 2 mA ausgelegt und mit entsprechenden Widerständen begrenzt.



Schaltung P170 mit Attiny 841, ohne Quarz, mit 8 Ausgängen

Funktion des Programmes:

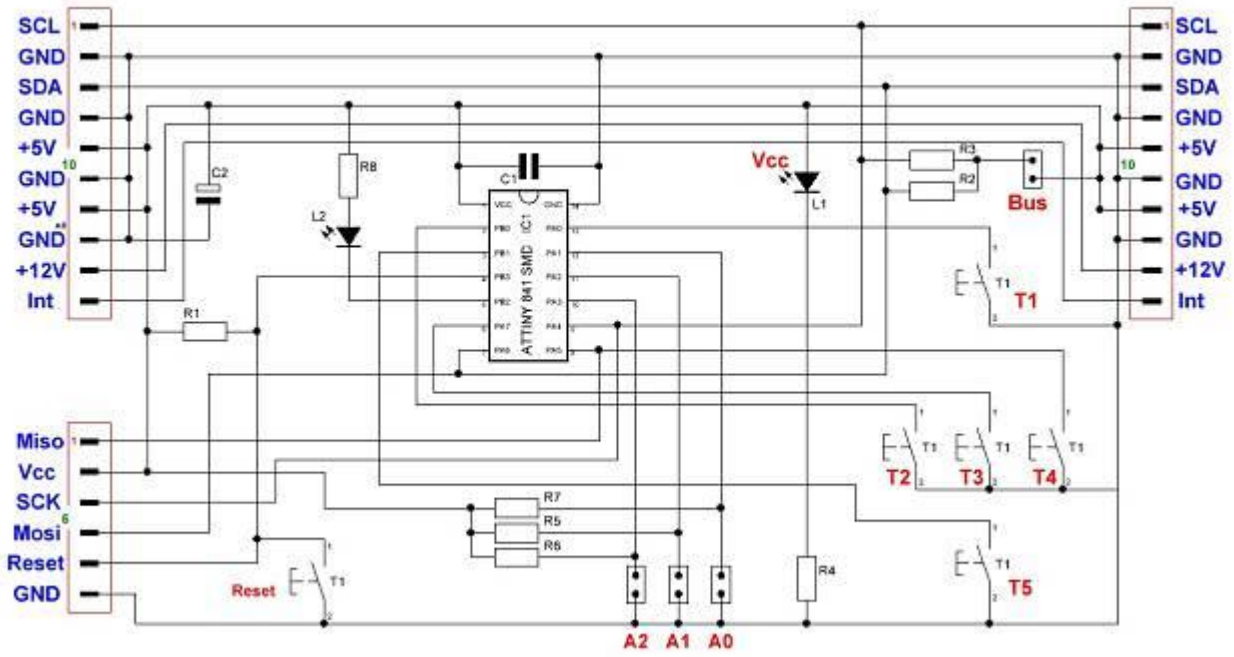
Es werden 8 Ausgänge angesteuert und mit einem Takt von einer Sekunde durchgeschaltet.

```
/* ATB_Ati_841_Prg_1.c * Auhtor hjsee Programm für Attiny 841 ohne Quarz mit Plat. 170 */
#define F_CPU 8000000UL                 // Angabe der Frequenz, wichtig für die Zeit
#include <util/delay.h>                 // Einbindung Datei Pause
#include <avr/io.h>                     // Einbindung Datei Ausgänge
int main(void)
 {
   DDRA=0b10101111;                     // Port A auf Ausgang schalten
   DDRB=0b00000011;                     // Port B auf Ausgang schalten
   PORTA=0b10101111;                    // Port A auf aus
   PORTB=0b00000011;                    // Port B auf aus
   while(1)                             // Programmschleife
    {
      // Pin 1
      PORTA &= ~(1<<PA0);               // Schaltet Pin aus
      _delay_ms(500);                   // Pause 500 ms
      PORTA |= (1<<PA0);                // Schaltet Pin ein
      _delay_ms(500);                   // Pause 500 ms
      // PIN 2
      PORTA &= ~(1<<PA1);               // Schaltet Pin aus
      _delay_ms(500);                   // Pause 500 ms
      PORTA |= (1<<PA1);                // Schaltet Pin ein
      _delay_ms(500);                   // Pause 500 ms
      // PIN 3
      PORTA &= ~(1<<PA2);               // Schaltet Pin aus
      _delay_ms(500);                   // Pause 500 ms
      _delay_ms(500);                   // Pause 500 ms
      PORTA |= (1<<PA2);                // Schaltet Pin ein
```

```
    _delay_ms(500);                 // Pause 500 ms
    // PIN 4
    PORTA &= ~(1<<PA3);             // Schaltet Pin aus
    _delay_ms(500);                 // Pause 500 ms
    PORTA |= (1<<PA3);              // Schaltet Pin ein
    _delay_ms(500);                 // Pause 500 ms
    // PIN 5
    PORTA &= ~(1<<PA5);             // Schaltet Pin aus
    _delay_ms(500);                 // Pause 500 ms
    PORTA |= (1<<PA5);              // Schaltet Pin ein
    _delay_ms(500);                 // Pause 500 ms
    // PIN 6
    PORTA &= ~(1<<PA7);             // Schaltet Pin aus
    _delay_ms(500);                 // Pause 500 ms
    PORTA |= (1<<PA7);              // Schaltet Pin ein
    _delay_ms(500);                 // Pause 500 ms
    // PIN 7
    PORTB &= ~(1<<PB1);             // Schaltet Pin aus
    _delay_ms(500);                 // Pause 500 ms
    PORTB |= (1<<PB1);              // Schaltet Pin ein
    _delay_ms(500);                 // Pause 500 ms
    // PIN 8
    PORTB &= ~(1<<PB0);             // Schaltet Pin aus
    _delay_ms(500);                 // Pause 500 ms
    PORTB |= (1<<PB0);              // Schaltet Pin ein
    _delay_ms(500);                 // Pause 500 ms
  }
}
```

Einstellung der Fuse beim Attiny 841 ohne Quarz:

Ex 0xFF
Hi 0xDF
Lo 0xC2



Schaltung P177 mit Attiny 841, ohne Quarz, mit 5 Tastern

```
/* ATB_Ati_841_Prg_3.c  * Author : hjsee */

// Programm für Attiny 841 ohne Quarz mit Platine 177
// 5 x Taster als Bedienkreuz ohne Widerstand nach Vcc und Kontroll LED
// 3 x Wahl für Busadresse A0, A1, A2

// Platine P177 Attiny 841 ohne Quarz
// PA0 --> T1
// PA1 --> A0
// PA2 --> A1
// PA3 --> A2
// PB0 --> T2
// PA7 --> T3
// PA5 --> T4
// PB1 --> T5
// PB2 --> LED 2

#define F_CPU 8000000UL                 // Angabe der Frequenz, wichtig für die Zeit
#include <util/delay.h>                 // Einbindung Datei Pause
#include <avr/io.h>                     // Einbindung Datei Ausgänge

int main(void)
 {
   DDRA=0b00000000;                     // DDRA schalten
   DDRB=0b00000100;                     // DDRB schalten
   PORTA=0b00000000;                    // Port A ein/aus 0=Eingang
   PORTB=0b00000100;                    // Port B ein/aus 1=Ausgang

   // PUEA=0b00000001;                  // R auf Taster 1 PA0
   // PUEB=0b00000001;                  // R auf Taster 2 PB0
   // PUEA=0b10000000;                  // R auf Taster 3 PA7
   // PUEA=0b00100000;                  // R auf Taster 4 PA5
   PUEB=0b00000010;                     // R auf Taster 5 PB1

   while(1)                             // Programmschleife
    {
      // if (PINA & (1<<PINA0))         // Taster T1 PA0
      // if (PINB & (1<<PINB0))         // Taster T2 PB0
      // if (PINA & (1<<PINA7))         // Taster T3 PA7
      // if (PINA & (1<<PINA5))         // Taster T4 PA5
      if (PINB & (1<<PINB1))            // Taster T5 PB1
       {                                // Wenn T1 gedrückt...
         PORTB |=(1<<PINB2);            // LED 2 ein
       }
      else
       {                                // wenn nicht
         PORTB &=~(1<<PINB2);           // LED 2 aus
       }
   }
```

In der Schaltung habe ich die Taster ohne einen Widerstand nach Vcc eingebaut. Die Widerstände schalte ich durch die Anweisung

```
PUEB=0b00000010;                    // R auf Taster 5 PB1
```

ein und verbinde die Taster nach Vcc. Es muss der korrekte Widerstand durch den Anweisung PUEB ausgewählt werden.

Die Einstellungen für Fuse beim Attiny 841:

Fuse Einstellung ohne Quarz

Ex - 0xFF
Hi - 0XDF
Lo - 0xC2

Fuse Einstellung mit Quarz

Ex - 0xFF
Hi - 0XDF
Lo - 0x8F

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