

# MIKROKONTROLLER & I<sup>2</sup>C BUS



by AS

[www.boxtec.ch](http://www.boxtec.ch)

[playground.boxtec.ch/doku.php/tutorial](http://playground.boxtec.ch/doku.php/tutorial)

SPI mit dem Attiny 841  
Teil 6 - Anzeige Software

SPI und der  
Attiny 841



## 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.

# Attiny 841 und SPI Teil 6 - Anzeige Software

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

Platine P166 mit Attiny 841 und Display DOG M 163 von EA und LED Beleuchtung



Beim Display DOG M 163 gibt es 3 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.

Doch zunächst sehen wir uns den vom Hersteller integrierten Zeichensatz an.

Angabe der Adresse:  
**obere Zeile: b7 - b4**  
**Untere Zeile: b3 - b0**  
 Entweder falsch bezeichnet oder sehr schlecht zu lesen

b7-b4 b3-b0	0000	0001	0010	0011	0100	0101	0110	0111	1000	1001	1010	1011	1100	1101	1110	1111
0000	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐
0001	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐
0010	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐
0011	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐
0100	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐
0101	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐
0110	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐
0111	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐
1000	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐
1001	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐
1010	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐
1011	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐
1100	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐
1101	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐
1110	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐
1111	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐

Es wurden bereits die Buchstaben und Symbole in der Tabelle hinterlegt. Zusätzlich auch Zeichen aus einer anderen Sprache.

Zusätzlich können bis zu 8 Zeichen frei definiert werden.

Wie man Sonderzeichen definieren kann und wie sie auf das Display kommen beschreibe ich in einem anderen Tut.



## Welche Anweisungen sind zur Darstellung nötig?

In der Bibliothek `dog_m163.h` werden die einzelnen Funktionsdeklaration durchgeführt.

- `void init_3lines_DOGM163(void);` // initiiert Display mit 3 Zeilen
- `void write_char_DOGM163 (char instruction);` // Darstellung von ASCII auf dem Display
- `void write_string_DOGM163 (char *buffer);` // Ausgabe von Text in Klammern
- `void set_cursor_DOGM163 (char row, char column);` // setzt Cursor an bestimmten Punkt (Zeile / Spalte)
- `void clear_DOGM163 (void);` // löscht gesamtes Display
- `void set_contrast_DOGM163 (uint8_t contrastvalue);` // setzt die Helligkeit
- `void set_character_DOGM163 (uint8_t characterNumber, uint8_t data0, uint8_t data1, uint8_t data2, uint8_t data3, uint8_t data4, uint8_t data5, uint8_t data6, uint8_t data7);` // Definiert Sonderzeichen
- `void cursorON_blinkOFF_DOGM163 (void);` // Cursor aus nach letzten Buchstaben blinkt mit Strich
- `void cursorON_blinkON_DOGM163 (void);` // Cursor ein nach letzten Buchstaben blinkt mit Strich
- `void cursorOFF_blinkOFF_DOGM163 (void);` // Cursor aus nach letzten Buchstaben blinkt ohne Strich
- `void cursorOFF_blinkON_DOGM163 (void);` // Cursor ein nach letzten Buchstaben blinkt ohne Strich

In der Bibliothek `dog_m163.c` erfolgt die Definition der Funktionen.

Aufruf der einzelnen Funktionen erfolgt mit:

```
spi_init(); // initiiert SPI
init_3lines_DOGM163(); // initiiert Display
write_char_DOGM163 (0x1E); // Ω Zeichen aus Speicher darstellen
write_char_DOGM163 (0); // Sonderzeichen 0 auf Display
write_string_DOGM163("Hello"); // Ausgabe Text „Hello“
set_cursor_DOGM163(0,1); // setzt Cursor an Position Zeile 0, Spalte 1
clear_DOGM163 (); // löscht kompletten Inhalt des Displays
set_contrast_DOGM163 (30); // setzt Kontrast auf 30
set_character_DOGM163 (0, 0x04, 0x0a, 0x11, 0x1f, 0x11, 0x11, 0x1f, 0x00); // Haus
cursorON_blinkON_DOGM163 (); // verschiedene Funktionen für Cursor
cursorON_blinkOFF_DOGM163 ();
cursorOFF_blinkON_DOGM163 ();
cursorOFF_blinkOFF_DOGM163 ();
```

Als Beispiel verwende ich ein Teil des Codes um die Anwendung zu zeigen:

```
clear_DOGM163 (); // löscht Display
_delay_ms(2); // bei clear 2ms Pause
set_cursor_DOGM163(0,1); // setzt Cursor an Position
write_string_DOGM163("Sonderzeichen"); // Ausgabe Text
_delay_ms(1000);
set_cursor_DOGM163(1,0); // setzt Cursor an Position
write_string_DOGM163("Sonderzeichen1"); // Ausgabe Text
```

```
cursorON_blinkON_DOGM163 ();           // Cursor ein nach letzten Buchstaben blinkt mit Strich
_delay_ms (2000);
cursorON_blinkOFF_DOGM163 ();          // Cursor aus nach letzten Buchstaben blinkt mit Strich
_delay_ms (2000);
set_cursor_DOGM163(2,0);                // setzt Cursor an Position
write_string_DOGM163("Sonderzeichen2"); // Ausgabe Text
_delay_ms (2000);
cursorOFF_blinkON_DOGM163 ();          // Cursor ein nach letzten Buchstaben blinkt ohne Strich
_delay_ms (2000);
cursorOFF_blinkOFF_DOGM163 ();         // Cursor aus nach letzten Buchstaben blinkt ohne Strich
```

Man kann auch ohne Probleme die Schrift über das Display laufen lassen.

```
for(i=0; i<10; i++)
{
  set_cursor_DOGM163(2,i);              // setzt Cursor an Position
  write_string_DOGM163(" Hello");       // Ausgabe Text
  _delay_ms(500);
}
```

Die kompletten Programme habe ich als Beispiel als zip hinterlegt. Dort sind alle notwendigen Dateien enthalten.

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](mailto:myroboter@web.de)

<http://elektrotino.blogspot.com/2012/07/ea-dog-m.html>

Ich danke besonders Herrn Tino Kühn für seine Arbeit und dem Programm.