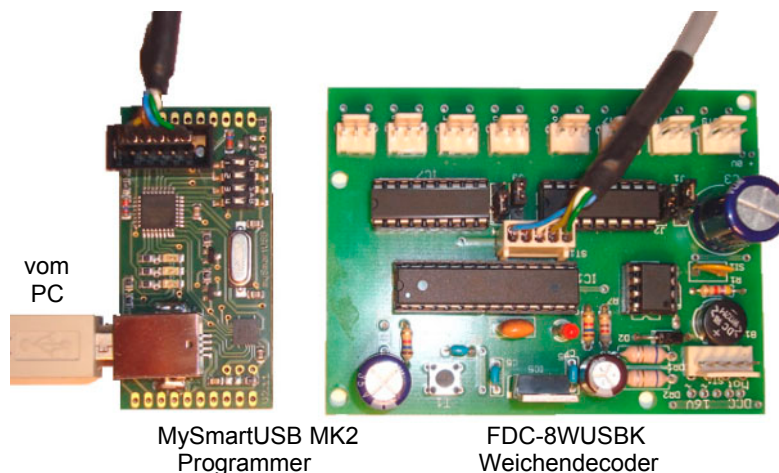


# ATMEL-Controller programmieren mit Programmer mySmartUSB MK2

© D. Ratschmeier



## Inhalt:

1.	<b>Allgemein</b>	<b>1</b>
2.	<b>Schaltereinstellung mySmartUSB KK2</b>	<b>2</b>
3.	<b>Installation myAVR_Prog_Tool</b>	<b>2</b>
4.	<b>ATMEL-Controller brennen</b>	<b>3</b>
5.	<b>Fuses brennen</b>	<b>3</b>
6.	<b>Haftung / Sicherheitshinweise</b>	<b>4</b>

## 1. Allgemein

Auf allen, auf dieser Webseite vorgestellten Funktionsmodulen mit AVR-Controller ist eine **ISP-Schnittstelle** (In System Programming) vorgesehen.

Über diese Schnittstelle kann das Betriebsprogramm direkt vom PC über einen Programmer und über eine 5-polige Datenleitung seriell in den Controller gebrannt werden.

Anfänglich habe ich wie unter **Software/ ISP-Programmierung** beschrieben die Programmierung der ATMEL-Controller einem selbst erstellten Schnittstellenkabel **Ser-ISP-Kab** und mit dem Programm **"PonyProg2000"** durchgeführt.

Seit längerer Zeit programmiere ich die Controller über eine kleine Programmerplatine **mySmartUSB MK2** von **myAVR** über die **USB-Schnittstelle**.

Diese in SMD-Technik ausgeführte Programmer-Leiterplatte kostet zur Zeit 28,00 Euro + Versandkosten.

Nähere Einzelheiten siehe

<http://shop.myavr.de/>

bzw. direkt unter

<http://shop.myavr.de/index.php?404>;<http://www.myavr.de:80/shop/article.php?artDataID=36>

Die Freeware-Programmier-Software **myAVR ProgTool** ist unter

[http://shop.myavr.de/index.php?ws=download\\_file.ws.php&dld=112&filename=software/tool\\_myavr-progtool-131-b2107\\_en\\_de.zip](http://shop.myavr.de/index.php?ws=download_file.ws.php&dld=112&filename=software/tool_myavr-progtool-131-b2107_en_de.zip)

herunter zu laden.

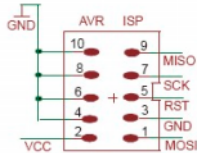
### Vorteile zur bisherigen Programmierung mit PonyProg2000.

- Kein selbst zu erstellendes **Ser-USB-Kabel** mit Leiterplatte im RS232-Anschlussstecker.
- Softwareunterstützung aller gängigen 8-Bit ATMELE-Controller
- Wesentlich schnellerer Programmiervorgang über die USB-Schnittstelle.  
(neue PCs haben nur noch USB-Schnittstellen)
- Programmierung ohne Stromversorgung des Anwendermoduls möglich.  
(Die Stromversorgung erfolgt vom PC über das USB-Kabel)

Es ist nur noch eine kurze 5-polige Datenleitung zwischen der Programmerplatine mySmartUSB MK2 zu der 5-poligen ISP-Stiftleiste der Nachbaumodule zu erstellen.

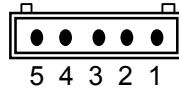
Pinbelegung der 5-poligen Verbindungsleitung vom myAVR MK2 zur 5-poligen Stiftleiste des Funktionsmoduls.  
z.B. FDC-8WUSBK

Belegung 10 polige ISP-Stiftleiste



Draufsicht auf 10 pol. Wannestiftleiste

Belegung 5 polige Buchsenleiste / Stiftleiste zum Funktionsmodul

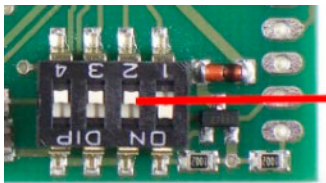


Draufsicht auf Drahteführseite

Signal	MySmart USB MK Pin	Funktionsmodul z.B. FDC-8WUSBK
MISO	9	4
SCK	7	3
RST	5	2
GND	3	1
MOSI	1	5

## 2. Schaltereinstellung mySmartUSB KK2

Nach Anleitung die 4 Schiebeschalter einstellen



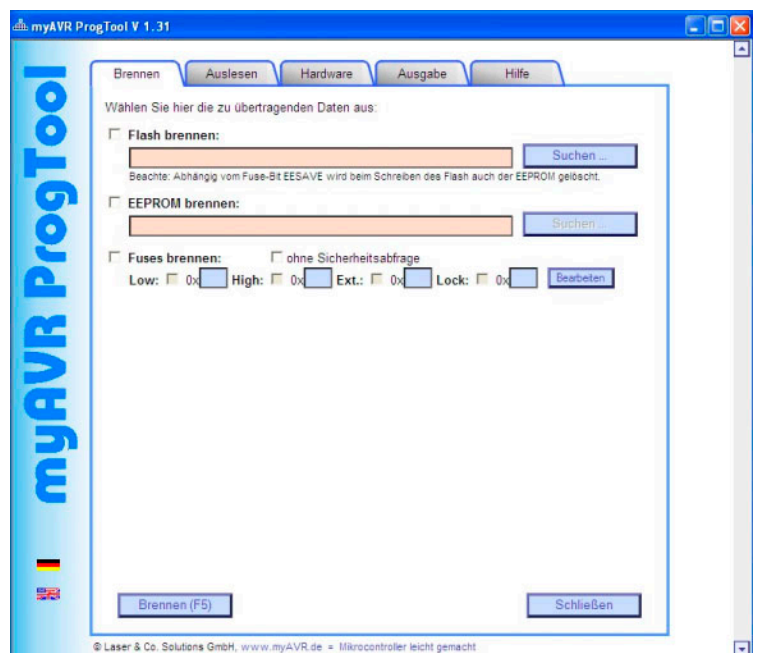
Schiebeschalter wie nebenstehend gezeigt einstellen.  
Bei dieser Einstellung ist keine Hilfsspannung der Controllerplatine notwendig wenn der Strombedarf kleiner 80mA bei 5V ist.  
Die weiteren Einstellungen sind in der technischen Anleitung ausführlich beschrieben.

## 3. Installation myAVR\_Prog\_Tool

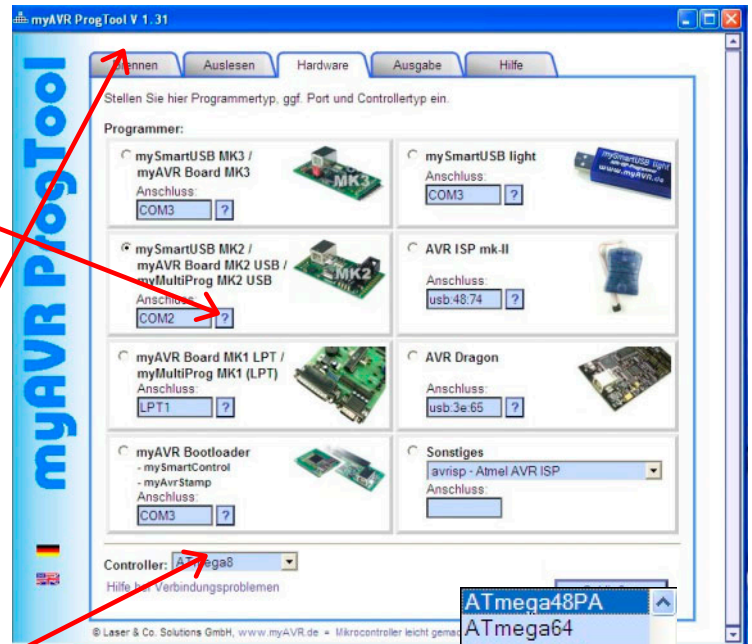
Installation wie in der entsprechenden myAVR Prog Tool Anleitung ausführlich beschrieben.

Nach dem Programmstart sollte nebenstehendes Menü erscheinen.

Im Register Hilfe sind alle wichtigen Dinge ausführlich beschrieben.



Über das Register **Hardware** ist zuerst die Programmierplatine **mySmartUSB MK2** und die serielle Schnittstelle einzustellen. Mit dem nebenstehenden **Button ?** kann der Istzustand und der ausgewählte Controller angezeigt werden.



Nach den grundsätzlichen Einstellungen kann das Controllerprogramm in den angegebenen Controller durch Anklicken von **Brennen** eingeleitet werden.

In der Auswahlliste **Controller** ist der entsprechende Controller des Anwendermoduls auszuwählen.

- ATmega48PA
- ATmega64
- ATmega640
- ATmega644
- ATmega644P
- ATmega645
- ATmega6450
- ATmega649
- ATmega6490
- ATmega64A
- ATmega64C1
- ATmega64M1
- ATmega8
- ATmega8515
- ATmega8535
- ATmega88

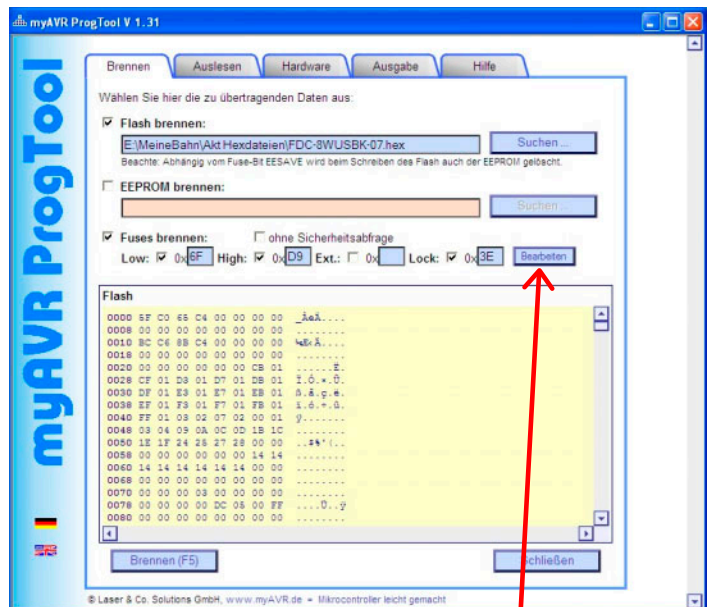
#### 4. ATMEL-Controller brennen

Vor dem Brennen (5) ist die entsprechende HEX-Datei im Dateiverzeichnis zu suchen und auszuwählen.  
 In den meisten Fällen sind abweichend von den Standardeinstellungen der **Fuses-Bits** bestimmte Anpassungen vorzunehmen.  
 Dazu sind entsprechende Häkchen unter **Fuses brennen** zu setzen.

#### 5. Fuses brennen

**Achtung!**  
 In jedem Fall muss die Auswirkung der Änderungen dem Anwender bekannt sein. In manchen Fällen kann der Controller nicht mehr im seriellen Mode programmiert werden.

Zum Ändern der **Fuses-Bits** sind die entsprechende Häkchen zu setzen. Abhängig von der Controllertyp sind dann noch die Häkchen für das Low, High, Ext. und Lockbyte zu setzen. Die einzelnen Funktionen sind in den entsprechenden Datenblättern der Controller zu finden. Mit dem Anklicken von **Bearbeiten** wird nachfolgendes Menü zum Ändern der entsprechenden Register sichtbar. Nach dem Anklicken der Register sind die Fuses-Bits Einstellungen sichtbar und können geändert werden.



Bei Controllern, die mit 5V arbeiten sollte die nebenstehende Einstellung vorgenommen werden.

Bei den Controllern mit L-Bezeichnung d. h. Betriebsspannungen von 3,3 bis 5V ist bei Betriebsspannungen < 5V VCC= 2,7V zu wählen.

Abhängig von der Taktfrequenz und von der Oszilatorschaltung ist die entsprechende Einstellung vorzunehmen.  
Das Beispiel zeigt die Einstellung für das FDC-8WUSBK Modul mit einem 8MHz Resonator.

Mit **Übernehmen** werden die aktuellen Einstellungen dauerhaft übernommen.  
Achtung! Bei Controlleränderungen sind die Einstellungen zu überprüfen und gegebenenfalls zu ändern.

Mit Anklicken von Button **Brennen (5)** wird nun das ausgewählte **ATMEL-Anwenderprogramm** mit den entsprechenden **Fuses-Bits** gebrannt.

Infos über die weiteren Funktionen sind unter dem Menüpunkt Hilfe zu erfahren.

## 6. Haftung / Sicherheitshinweise

Der Autor übernimmt keinerlei Haftung für eventuell entstehende Schäden bei der Anwendung dieser Anleitung und der Nutzung des beschriebenen mySmartUSB MK2 Programmers und der Programmier-Software **myAVR ProgTool**.

Viel Spaß!  
Für Rückfragen stehe ich gerne zur Verfügung

**Autor:**

Dieter Ratschmeier  
Am Rudolphshof 25  
91207 Lauf  
09123 988196

dieter.ratschmeier@Railway-Lauf.de

**Fuse- & Lock-Bits**

Standardwerte einstellen Übernehmen Verlassen

**Achtung** das verändern der Fuse-Bits kann dazu führen, dass der Prozessor nicht mehr programmierbar bzw. überhaupt erreichbar wird.

Hardware Auslesen Jetzt Schreiben

Low Fuse (0x2F)	High Fuse (0xD9)	Lockbits (0x3E)
0 0 1 0 1 1 1 1 1	1 1 0 1 1 0 0 1	1 1 1 1 1 1 0

Low Fuse  High Fuse  Lockbits

Brown-out detection level at VCC=4.0 V  
 Brown-out detection level at VCC=2.7 V

Brown-out detection enabled

- Ext. Crystal/Resonator High Freq.; Start-up time: 258 CK + 4 ms
- Ext. Crystal/Resonator High Freq.; Start-up time: 258 CK + 64 ms
- Ext. Crystal/Resonator High Freq.; Start-up time: 1K CK + 0 ms
- Ext. Crystal/Resonator High Freq.; Start-up time: 1K CK + 4 ms
- Ext. Crystal/Resonator High Freq.; Start-up time: 1K CK + 64 ms
- Ext. Crystal/Resonator High Freq.; Start-up time: 16K CK + 0 ms
- Ext. Crystal/Resonator High Freq.; Start-up time: 16K CK + 4 ms
- Ext. Crystal/Resonator High Freq.; Start-up time: 16K CK + 64 ms

Low Fuse (0x2F)	High Fuse (0xD0)	Lockbits (0x3E)
0 0 1 0 1 1 1 1 1	1 1 0 1 0 0 0 0	1 1 1 1 1 1 0

Low Fuse  High Fuse  Lockbits

Reset Disabled (Enable PC6 as I/O pin)

Watch-dog Timer always on

Serial program downloading (SPI) enabled

Preserve EEPROM memory through the Chip Erase cycle

- Boot Flash section size=128 words Boot start address=0x0F80
- Boot Flash section size=256 words Boot start address=0x0F00
- Boot Flash section size=512 words Boot start address=0x0E00
- Boot Flash section size=1024 words Boot start address=0x0C00; default value

Boot Reset vector Enabled (default address=0x0000)

CKOPT fuse (operation dependent of CKSEL fuses)

**Fuse- & Lock-Bits**

Standardwerte einstellen Übernehmen Verlassen

**Achtung** das verändern der Fuse-Bits kann dazu führen, dass der Prozessor nicht mehr programmierbar bzw. überhaupt erreichbar wird.

Hardware Auslesen Jetzt Schreiben

Low Fuse (0x2F)	High Fuse (0xD9)	Lockbits (0x3F)
0 0 1 0 1 1 1 1 1	1 1 0 1 1 0 0 1	1 1 1 1 1 1 1

Low Fuse  High Fuse  Lockbits

- Mode 1: No memory lock features enabled
- Mode 2: Further programming disabled
- Mode 3: Further programming and verification disabled

- Application Protection Mode 1: No lock on SPM and LPM in Application Section
- Application Protection Mode 2: SPM prohibited in Application Section
- Application Protection Mode 3: LPM and SPM prohibited in Application Section
- Application Protection Mode 4: LPM prohibited in Application Section

- Boot Loader Protection Mode 1: No lock on SPM and LPM in Boot Loader Section
- Boot Loader Protection Mode 2: SPM prohibited in Boot Loader Section
- Boot Loader Protection Mode 3: LPM and SPM prohibited in Boot Loader Section
- Boot Loader Protection Mode 4: LPM prohibited in Boot Loader Section