

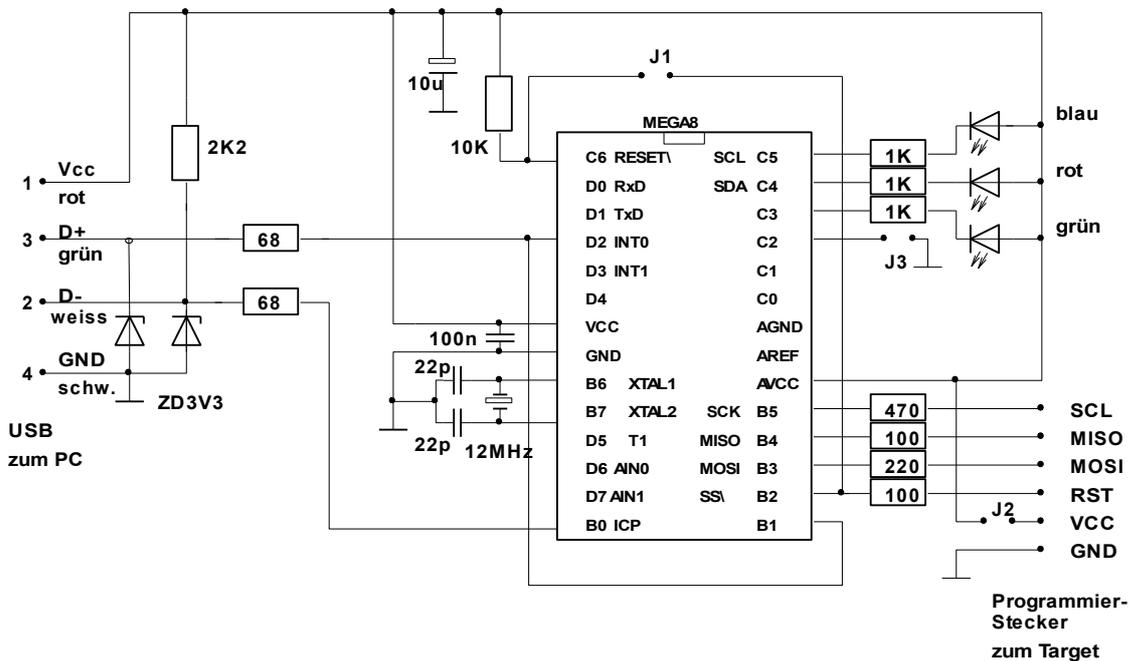
# Selbstbau-USB-Programmiergerät für AVR

Nach Christian Ullrich

Homepage: [www.ullihome.de](http://www.ullihome.de)

Weitere Infos bei [http://weigu.lu/a/pdf/MICEL\\_F6\\_Programmieradapter.pdf](http://weigu.lu/a/pdf/MICEL_F6_Programmieradapter.pdf)

## Schaltung



Keine externe Stromversorgung nötig, Gerät wird vom USB-Bus versorgt.

Im Schaltplan nicht eingetragen und auch nicht unbedingt notwendig (siehe Original-Schaltung):  
Ein Spannungsteiler 22K, 10K zwischen +5V und Masse, dessen Ausgang an C.1 liegt, erlaubt die Messung der Betriebsspannung in AVRStudio.

### Jumper:

Im Normalbetrieb ist kein Jumper gesteckt.

Funktion wenn gesteckt:

J1	Programmierung des internen Mega8 des Programmiergerätes über die SPI-Schnittstelle (wird nur ganz am Anfang 1x benötigt)
J2	Versorgung des Target-Mega8 über das Programmierkabel aus der USB-Schnittstelle.
J3	Erzwingen des Bootloader-Modus wenn der Mega8 des Programmiergerätes eine andere Firmware erhalten soll.

LEDs:

Blau	PC-Verbindung OK Blinkt bei Datentransfer
Rot	Programmivorgang läuft Blinkt wenn Target falsch angeschlossen
Grün	Target richtig angeschlossen (wenn J2 gesetzt)

## Inbetriebnahme

1. Internen Mega8 des Programmiergerätes mit einem Bootloader versehen.  
Dieser kann dann später über das USB-Kabel die eigentliche Firmware laden, ohne dass der Mega8 an ein Programmiergerät angeschlossen werden muss.

Hierzu die Datei BOOTLOADER\_M8.HEX aus avrisp\_bootloader\_m8\_20090401.zip extrahieren und mithilfe eines anderen Programmiergerätes (evtl. ausleihen!) in den Mega8 flashen.

Die Datei ist zu finden bei [www.ullihome.de/index.php/USB AVR ISP Download/de](http://www.ullihome.de/index.php/USB AVR ISP Download/de) unter "Firmware, Aktuelle Firmware, Bootloader für ATMega8 Stand 01.04.2009".

Hierzu kann man das andere Programmiergerät an den SPI-Anschluss hängen und Jumper J1 setzen.

### **Achtung: die Fusebits müssen auf 0xC8 (High) und 0xBF (Low) programmiert werden!**

Dies entspricht BOOTSZ = 00 (Boot size = 1024) , CKOPT = 0, BODLEVEL = 1, BODEN = 1, CKSEL = 1111, SUT = 11

In BASCOM:

Fuses (BF): BODLEVEL=2.7V, BODEN = ENABLED, KLA987=111111

Fuses High (C8): CKOPT=0, Reset vector = boot loader reset

### **Nach dem Programmieren J1 entfernen!**

2. Software AVR Lab Tool von der gleichen Seite herunterladen und installieren.

Schaltung an den USB-Bus anstecken.

Nun müsste unter Windows der Hardware-Assistent aktiv werden.

Diesem den Pfad der USB-Treiberdatei (das ist der Folder „driver“ im Installationsverzeichnis der AVR Lab Tool Software) mitteilen.

Nun kann der PC mit dem Mega8 über USB kommunizieren.

3. „USB AVR Lab Tool“ starten.  
„AVRISPMkII“ auswählen und auf „Programmieren“ , dann „Start“ klicken.

Es sollte kurz die rote LED aufleuchten, dann nur noch blau und grün.

Nun emuliert das Programmiergerät einen AVRISP mkII.

Dieses benötigt als neue Hardware wieder die Angabe des (gleichen) Pfades für den Hardware-Assistenten.

Treiber installieren, fertig.

## **Programmieren unter AVRStudio**

Dies sollte gleich möglich sein, wenn AVR-Studio mit mkII-USB-Driver installiert ist.

Wenn nicht, testen:

-  Connect AVRISP mkII
-  AVR
  - HW Settings: Target voltage muss angezeigt werden mit ca. 5V
  - Read Signature: der Mikrocontroller muss erkannt werden

## **Test von der Kommandozeile**

Bei Problemen kann es interessant sein, den Kommandozeilen-Programmierer STK500.exe zu benutzen, da er Meldungen ausgibt. (Zu finden im AVRTools Directory)

**Achtung: die Befehle unterscheiden Gross- und Kleinschreibung, und es darf keine Leerstellen an falscher Stelle geben, also z.B. „-cUSB“ und nicht „-c usb“ !**

Mit „STK500 -h“ erhält man eine Befehlsübersicht, „STK500 -?“ gibt eine Liste der unterstützten Mikrocontroller aus.

Beispiele:

```
stk500 -cUSB -dATmega8 -s
```

liest die Signatur des Chips aus

```
C:\Program Files\Atmel\AVR Tools\STK500>stk500 -cUSB -dATmega8 -s -y -q -wt -J
```

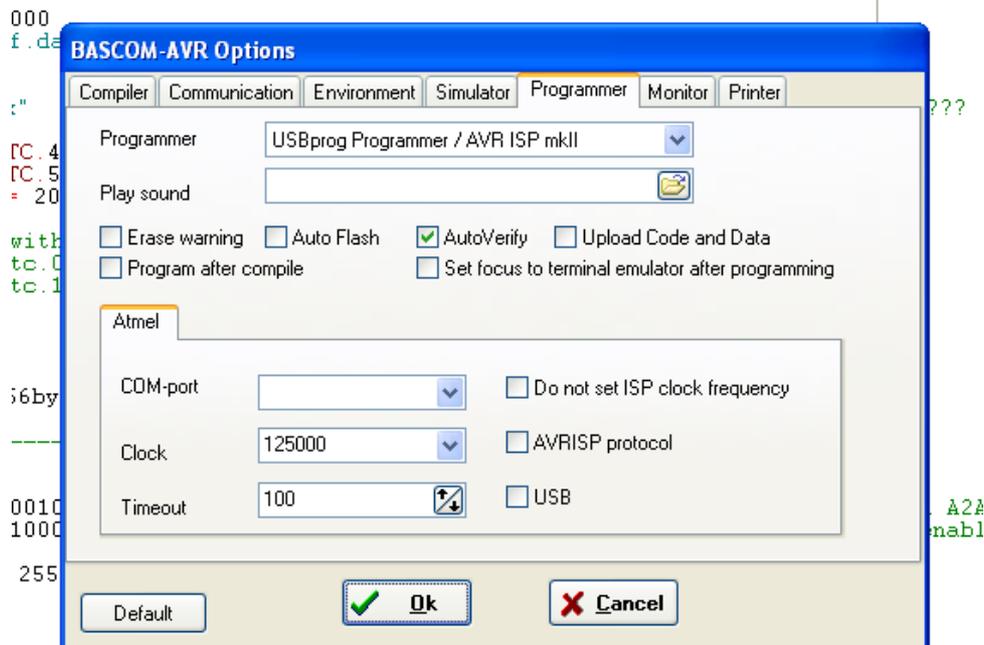
```
STK500 command line programmer, v 2.2 Atmel C
Connected to STK500 V2 on port USB:0000B00021
Getting ISP frequency: 125.0 kHz (0x06)
Getting target voltage VTARGET: 5.1V
Device parameters loaded
Programming mode entered
Signature is 0x1E 0x93 0x07
Reading fuse bits...
Fuse byte 0 read (0xE1)
Fuse byte 1 read (0xD9)
```

Reading lock bits... Lock bits read (0xFF)  
 Programming mode left  
 Connection to STK500 V2 closed

## Programmieren unter BASCOM

Benötigt ein installiertes AVR Studio mit USB-Driver.

1. Wenn noch nicht geschehen, **setublibusb.exe** (zu finden im BASCOM-Installationsverzeichnis) ausführen um **LIBUSB** zu installieren.
2. Als Programmierer „USBprog Programmer / AVR ISP mkII“, früher „Elektor Programmer / AVR ISP mkII“ in den BASCOM-Optionen auswählen.



**Wichtig: Clock = 125000Hz**

Achtung:

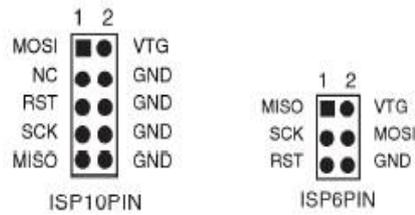
Zum Programmieren ist folgende Minimalbeschaltung nötig:

- Pullup 10K am Reset-Eingang
- Betriebsspannung abgeblockt mit 100nF
- Quarz mit 2 x 22pF  
 (ausser bei Betrieb mit internem Oszillator, wenn Fuse Bits dementsprechend gesetzt sind)

## Fehlersuche

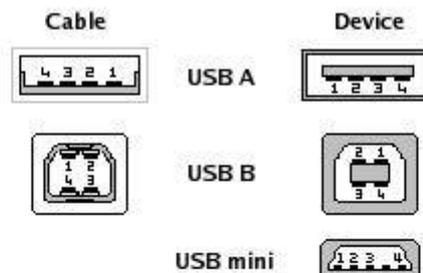
Mit dem USB Device Viewer (Freeware, Microsoft) kann man sehen, welche Geräte an den USB Ports angeschlossen sind)

## Stecker:



LTAM-Connector:

6	SCK
5	MISO
4	MOSI
3	RESET\
2	+5V
1	GND



Pin	Signal	Color	Description
1	VCC	Red	+5V
2	D-	White	Data -
3	D+	Green	Data +
4	GND	Black	Ground