

# USB-AVR-Programmer

(USB-AVR-Mini, AVRISP mk II-Clone)



Gerhard Schmidt, Kastanienallee 20, D-64289 Darmstadt, <http://www.avr-asm-tutorial.net>

## **Inhalt**

**Beschreibung**

**Installation**

**Bedienung**

**Umprogrammieren**

**Copyright und Quellen**

**Anhang**

- **Schaltbild**
- **Bestückungsplan**

## **Beschreibung**

Die Schaltung ist ein universelles Programmiergerät für die In-System-Programmierung von ATMEL-AVRs. Es arbeitet mit einem ATMEL vom Typ ATmega8.

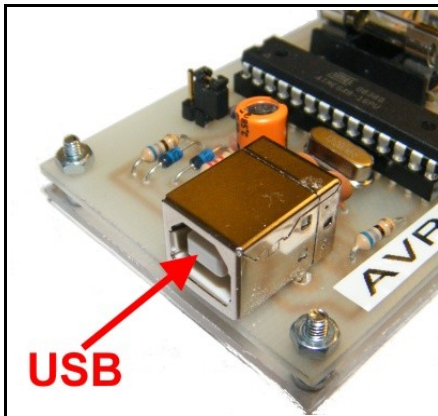
In diesem vorinstalliert ist die Firmware, die ein Programmiergerät von ATMEL mit der Bezeichnung AVR-ISP-mk II emuliert. Zum Programmieren wird es an einen USB-Port des PC angeschlossen und dort als Jungo-Device erkannt. Es ist mit dem AVR Studio 4 und mit der Programmiersoftware avrdude ansteuerbar. Es kann mit weiterer Firmware umprogrammiert werden, sofern diese verfügbar ist.

Als Programmierport ist ein Standard-ISP10-Anschluss mit einem verpolungssicheren Wannensteckeranschluss vorhanden. Die Versorgung der zu programmierenden Zielschaltung mit 5 V Betriebsspannung kann aus dem Rechner über den USB-Anschluss erfolgen. Maximal sind 0,5 A Last zulässig und über eine flinke Feinsicherung gegen Kurzschluss abgesichert. Der USB-Anschluss ist dabei über eine Schottky-Diode gegen Überspannungen aus der Zielschaltung geschützt.

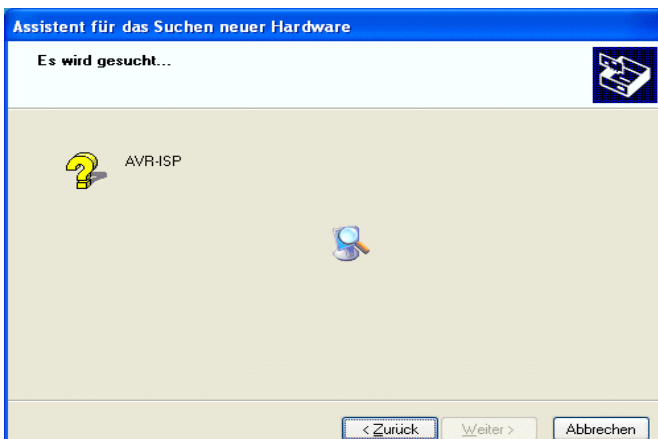
Der ISP10-Anschluss verfügt abweichend von der Standard-Belegung über einen seriellen bidirektionalen Anschluss, der für künftige Debug-Software vorbereitet ist. Der Sende-Anschluss zur Zielschaltung ist strombegrenzt, so dass ohne Hardware-Änderungen auch Standard-ISP10-Hardware angeschlossen werden kann.

## **Installation**

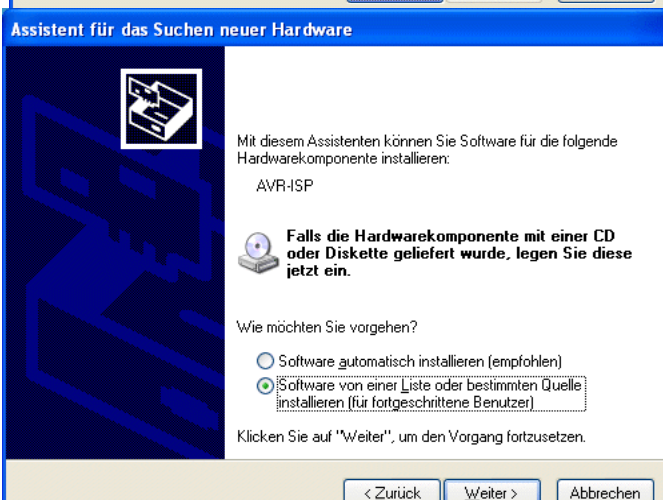
Achtung! Die vorinstallierte Firmware arbeitet nur mit Windows XP, nicht mit Vista!



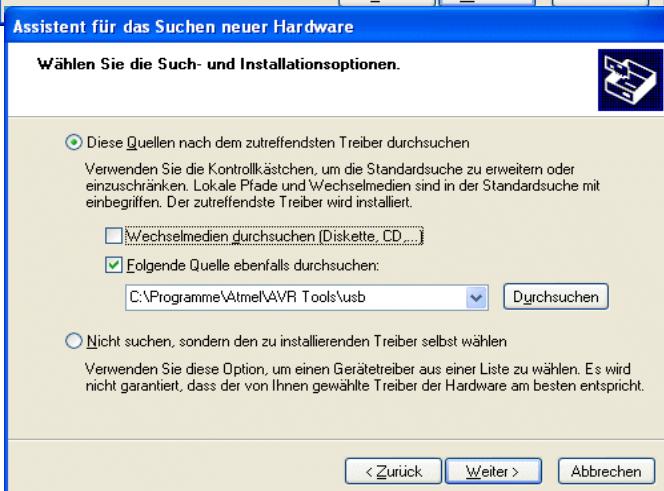
Zur Installation unter Windows XP wird der USB-Anschluss des Programmiergeräts mit einem USB-Anschluss verbunden.



Wenn der PC einen funktionierenden Internet-Zugang hat, wird der Jungo-Device-Treiber automatisch von der Webseite von ATMEL Norway heruntergeladen und installiert.



Wenn kein Internet-Zugang besteht, kann der mit dem ATMEL Studio 4 heruntergeladene USB-Treiber installiert werden. Dazu die automatische Treibersuche von XP deaktivieren.



Stattdessen in das Verzeichnis „C:\Programme\Atmel\AVR Tools\usb“ oder ein anderes Verzeichnis wechseln, unter dem das Studio installiert wurde.

Zum Testen des Programmiergeräts

1. das AVR Studio 4 starten, die Projekt- und Device-Auswahl mit „Cancel“ übergehen und
2. im Menüpunkt „Tools“ „Program AVR“ und „Auto Connect“ auswählen.

Das Studio sollte nun ein Programmiergerät „AVR ISP mk II“ finden und dessen Steuerfenster öffnen (siehe unten). Im Tabulator „Board“ kann nun die ISP-Taktfrequenz z. B. auf 250 kHz verstellt, mit „Write“ an das Programmiergerät übergeben und mit „Read“ wieder eingelesen werden. Beide Werte sollten übereinstimmen.

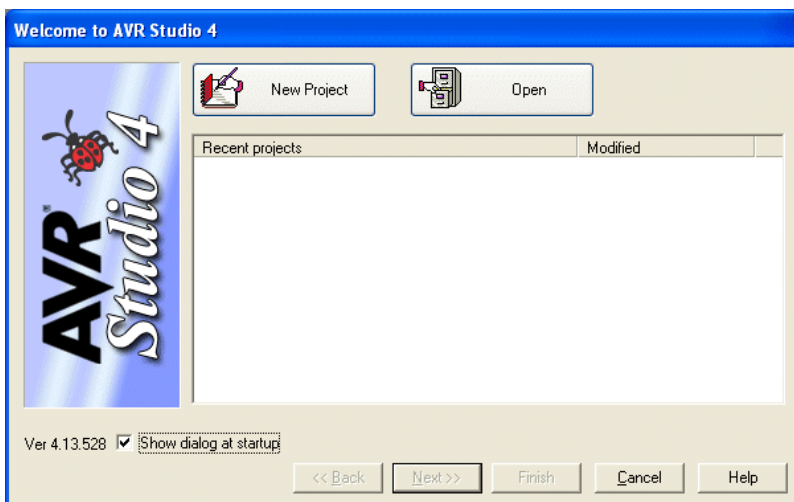
## Bedienung mit dem Studio

Besitzt die zu programmierende Schaltung eine eigene Stromversorgung, kann die Versorgung aus dem USB-Anschluss durch Herausnahme der Feinsicherung abgeschaltet werden. Die Versorgungsspannung der Zielschaltung sollte dann aber nicht wesentlich unter 4,5 V liegen.

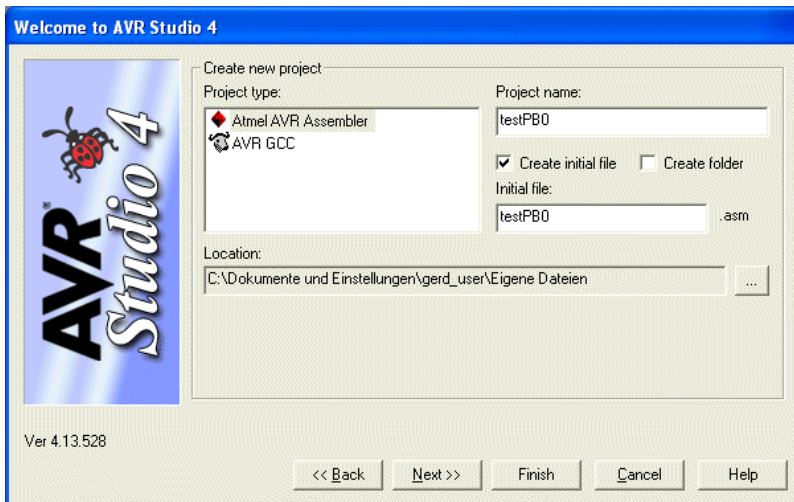


Die Zielschaltung mit einem Flachbandkabel über den ISP10-Steckverbinder mit dem Programmiergerät verbinden. Dabei auf die Nase an der Steckbuchse achten.

Dann die USB-Buchse des Programmiergeräts mit dem PC verbinden. Wenn das USB-Gerät erkannt ist und der Treiber gefunden wurde, sollte nach kurzer Zeit die LED auf dem Programmiergerät rot leuchten.

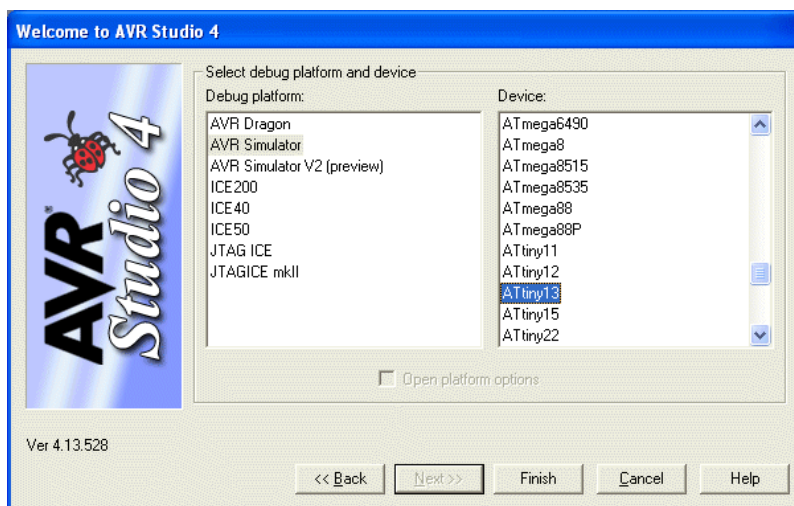


Das ATMEL Studio starten und den Button „New Project“ wählen.



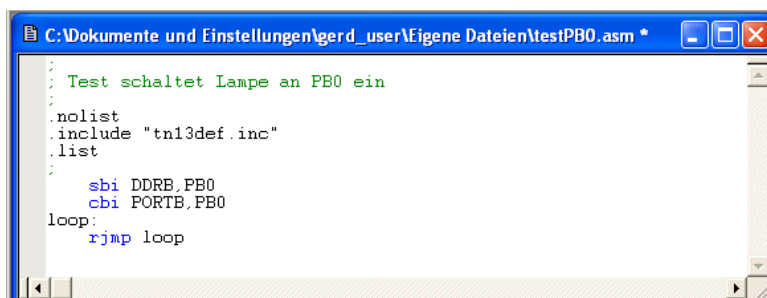
Als Projekttyp „Atmel AVR Assembler“, als Name „testPB0“ und als Location ein bestehendes Verzeichnis wählen.

Den Dialog mit „Next“ beenden.



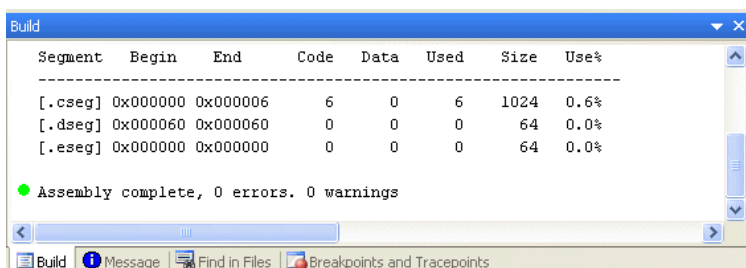
Als „Debug platform“ „AVR Simulator“, als Device „ATtiny13“ wählen.

Den Dialog mit „Finish“ schließen.

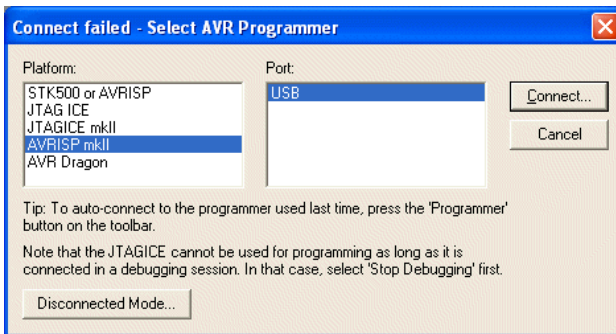


Im Editorfenster des Studio das nebenstehende Programm eintippen.

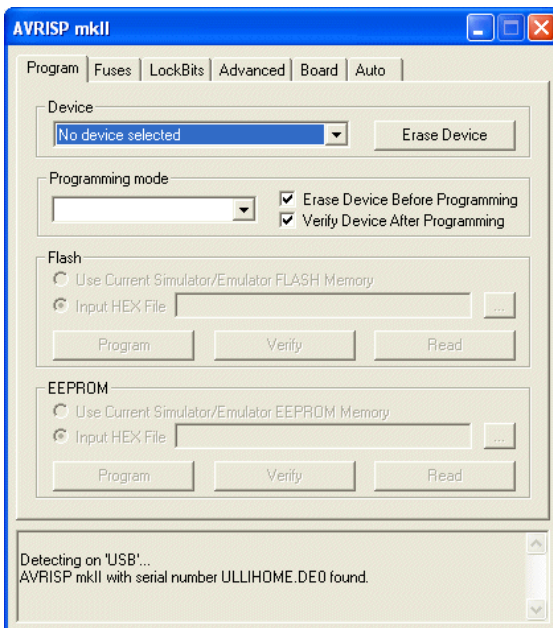
Das Programm mit „Build“ und „Build“ übersetzen.



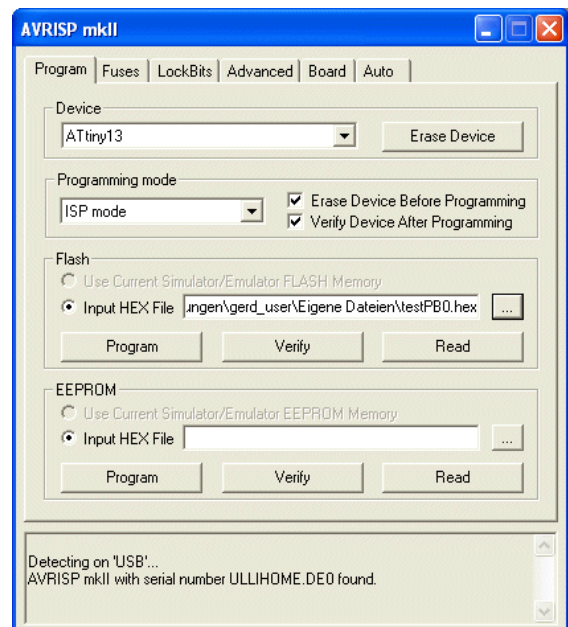
Im Build-Fenster des Studio sollte jetzt die fehlerfreie Assemblierung des Programms angezeigt werden.



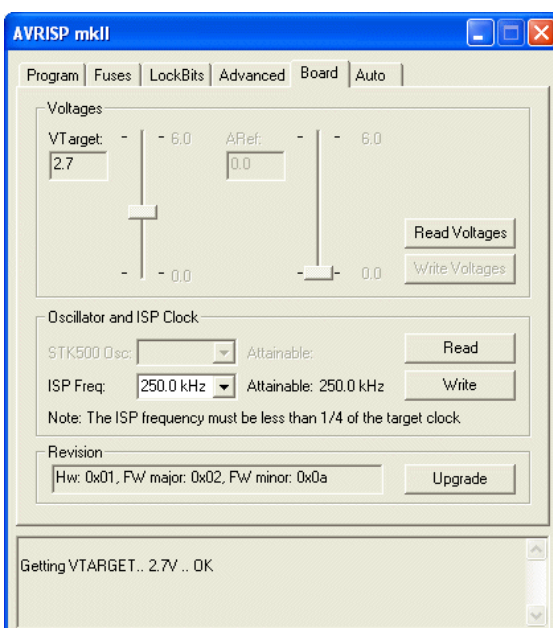
Nun die Verbindung mit den angeschlossenen Geräten mit „Tools“, „Program AVR“ und „Auto Connect“ starten. Wenn keine automatische Erkennung erfolgt („Connect failed“), dann die nebenstehenden Einstellungen auswählen.



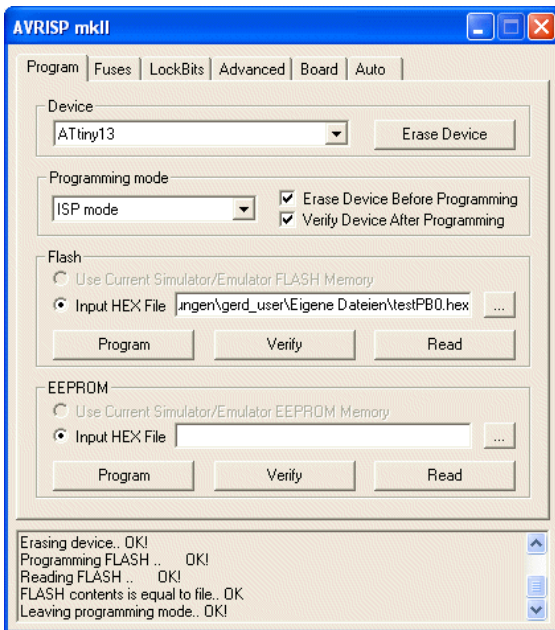
Bei erkanntem AVRISP-Gerät sollte sich dessen Fenster öffnen (links). Hier das Device, den Programming Mode und die Position der Hexadezimaldatei eintragen (rechts).



Nun (wie links gezeigt) die ISP-Frequenz in der Sektion „Oscillator and ISP-Clock“ auf 250 kHz einstellen und mit dem Button „Write“ an das Programmiergerät übermitteln. „Read“ sollte denselben Wert zurückliefern.

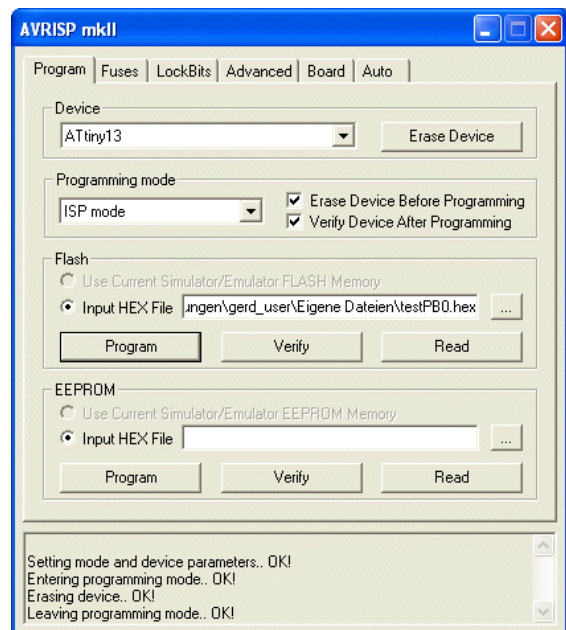


Auslesen und Setzen der Target-Spannung sind im Programmiergerät nicht implementiert. Auslesen liefert keinen korrekten Wert, Setzen bleibt wirkungslos.



Nun im Tabulator „Program“ in der Sektion „Flash“ mit dem „Program“-Button die Hex-Datei in den AVR übertragen.

Die Meldungen im unteren Fensterteil beachten. Nun sollte eine an PB0 angeschlossene LED leuchten.



Mit dem Button „Erase Device“ in der Sektion „Device“ (rechts) sollte die LED wieder erlöschen, da der Inhalt des Flash-Speichers wieder gelöscht wird.

## Umprogrammieren des Programmiergeräts

Zum Umprogrammieren des Programmiergeräts wird ein anderes funktionierendes Programmiergerät benötigt. Die Programmierung erfolgt über den ISP10-Anschluss, der USB-Anschluss darf dabei nicht mit dem PC verbunden sein. Der Programmierjumper muss gesteckt sein. Zur Spannungsversorgung während des Programmierens aus dem zweiten Programmiergerät muss die Schottky-Diode überbrückt werden (z. B. durch Verbinden des Einzelpins neben der LED mit einer Seite der Sicherung).

Zum Umprogrammieren stehen z. Zt. folgende Firmware-Versionen zur Verfügung (siehe Quellen):

- m8\_STK500v2\_1.5.hex: Simuliert ein STK500-Programmierboard,
- m8\_USBasp\_1.1.hex: Simuliert ein USBasp-Programmiergerät,
- m8\_Bitbanging\_1.4.hex: Simuliert einen Bitbanging-Programmieradapter.

In allen Fällen muss die Fuse des ATmega8 auf den externen Quarz (High Speed Xtal) eingestellt sein.

## Copyright und Quellen

Die Original-Hardware dieses Programmiergeräts (AVRlab-Mini) und die verwendete Firmware-Version wurde von Christian Ulrich entwickelt.

<http://www.ullihome.de>

Alle Firmware-Versionen können von dieser Webseite bezogen werden.

Die Hardware-Version, die hier beschrieben ist, das Layout und das Schaltbild wurde von Wolfgang Haubach, Darmstadt, erstellt.

Der Programmieradapter USBasp wurde von Thomas Fischl entwickelt:

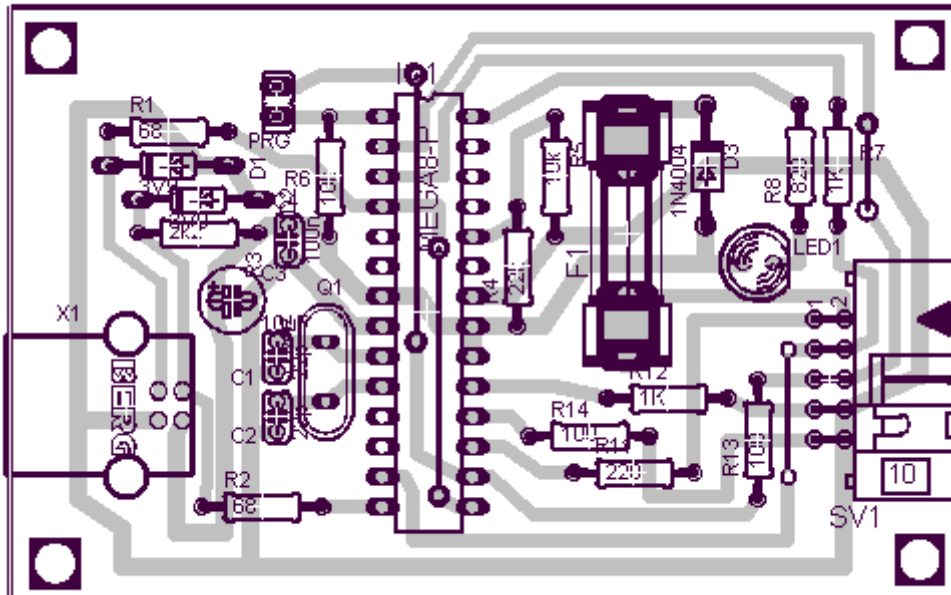
<http://www.fischl.de/usbasp/>

Diese Anleitung wurde von Gerhard Schmidt, Darmstadt, erstellt:

<http://www.avr-asm-tutorial.net>



## Bestückungsplan AVRISP mk II-Clone



### Hinweise:

- Die Sicherung sollte maximal vom Typ 0,5 A Flink sein. Höhere Werte können bei Kurzschlüssen in der angeschlossenen Hardware die USB-Hardware des PC gefährden.
- Anstelle der im Bestückungsplan eingezeichneten Si-Diode 1N4004 ist in den Programmiergeräten eine Schottky-Diode SB140 mit niedrigerem Spannungsabfall verbaut.
- Der Programmierjumper „PRG“ darf im normalen Betrieb nicht gesetzt sein.
- Der Einzelpin neben der LED ist mit 5 V (USB) verbunden. Verbinden mit einem Sicherungsanschluss überbrückt beim Umprogrammieren die Diode.
- In einigen Geräten sind Zenerdioden mit 3,6 V, in anderen mit 3,9 V Zenerspannung verbaut.

# Schaltbild

