

# USBasp unter WinAVR und Atmel Studio

Der Betrieb von Programmern des Typs USBasp (z.B. guloprogramm) ist unter Linux relativ einfach:  
- keine Treiberinstallation notwendig, benötigte Werkzeuge sind im Linux-Standard-Repository.

Natürlich können USBasp-Programmer auch unter dem Betriebssystem Windows genutzt werden. Dazu wird die folgende Software empfohlen:

*avrdude* sehr leistungsfähiges Kommandozeilen-Tool für das Beschreiben und Auslesen von AVR-Mikrocontrollern, einschließlich Fuse-Bytes  
*AVR-Burn-o-Mat* grafische Bedienoberfläche für die wichtigsten Funktionen von avrdude (optional)  
*WinAVR* Komplettpaket zur C-Programm-Entwicklung für Windows, enthält avrdude

Alternativ dazu:

*Atmel Studio* Entwicklungsumgebung des Herstellers (Name vor Version 6: *AVR Studio*)  
*avrdude* als Bindeglied zwischen Atmel Studio und USBasp-Programmer

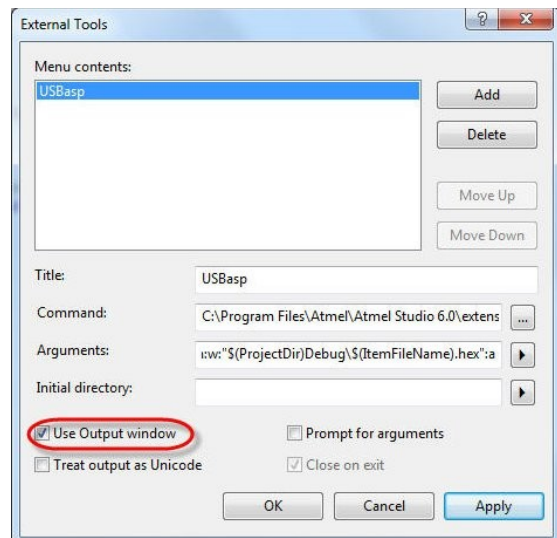
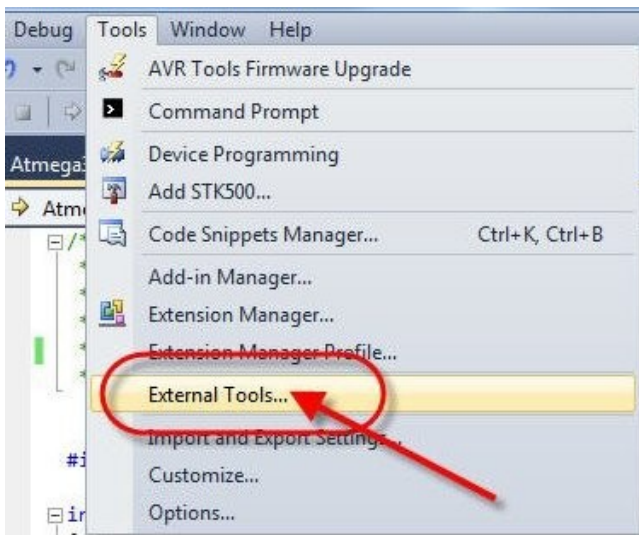
## USBasp unter Atmel Studio

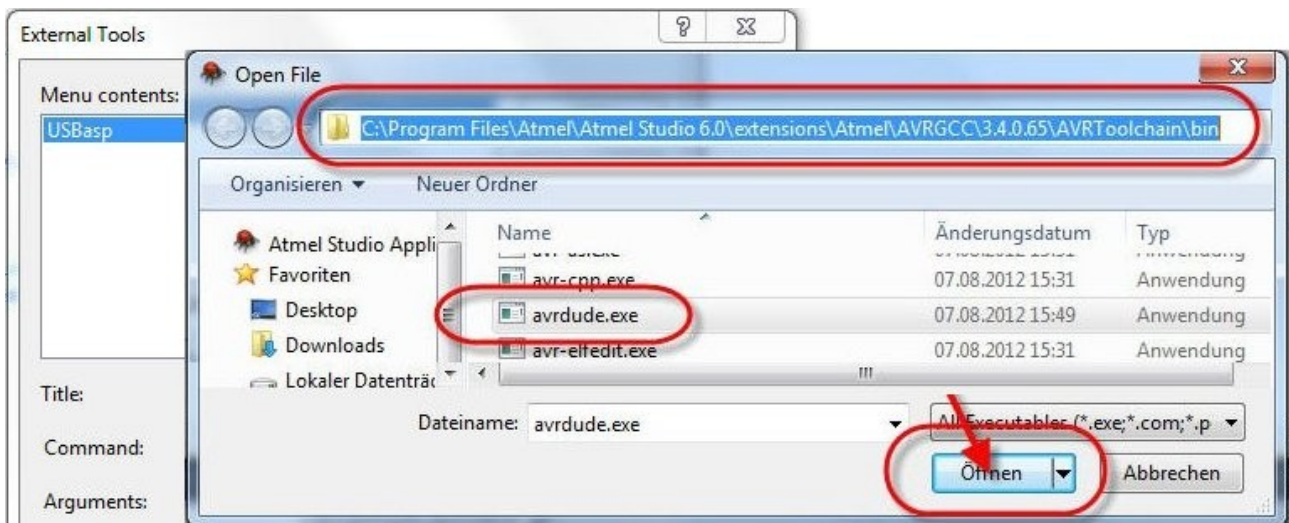
Für den Fall, dass ein USBasp-Prgrammer nicht zusammen mit *WinAVR*, sondern unter *Atmel Studio 6* verwendet werden soll, sind die im Folgenden beschriebenen Schritte notwendig.

Zuerst müssen die Dateien "avrdude.exe", "avrdude.conf" und "libusb0.dll" in den Ordner C:\Program Files\Atmel\Atmel Studio 6.0\extensions\Atmel\AVRGCC\3.4.0.65\AVRToolchain\bin (bzw. C:\Program Files\Atmel Toolchain\AVR8 GCC\Native\3.4.1056\avr8-gnu-toolchain\bin) kopiert werden. Herunterladen dieser drei Dateien z.B. von: <https://guloshop.de/f/atmelstudio>

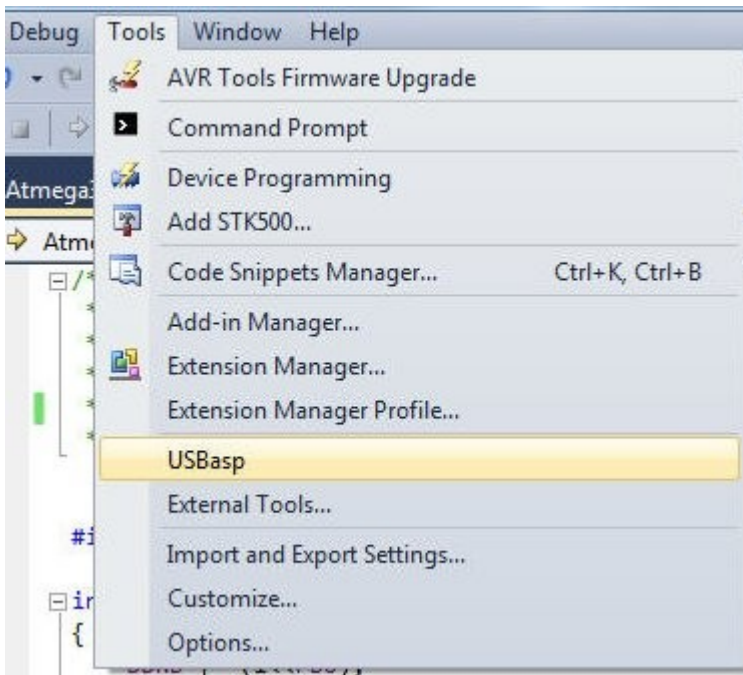
Nun das *Atmel Studio* starten und im Menü "Tools" auf "External Tools" klicken. Mit "Add" ein neues Tool anlegen und die folgenden Zeilen ausfüllen:

Title: USBasp  
 Command: C:\Program Files\Atmel\Atmel Studio 6.0\extensions\Atmel\AVRGCC\3.4.0.65\AVRToolchain\bin\avrdude.exe  
 (dazu rechts neben dem Eingabefeld auf "..." klicken und avrdude.exe im Ordner "AVRToolchain\bin" auswählen – siehe Bild nächste Seite)  
 Arguments: -e -c USBasp -p ATtiny13 -e  
 -U flash:w:"\$(ProjectDir)Debug\\$(ItemFileName).hex":a  
 (den Mikrokontrollertyp "ATtiny13" ggf durch den gewünschten Typ ersetzen)

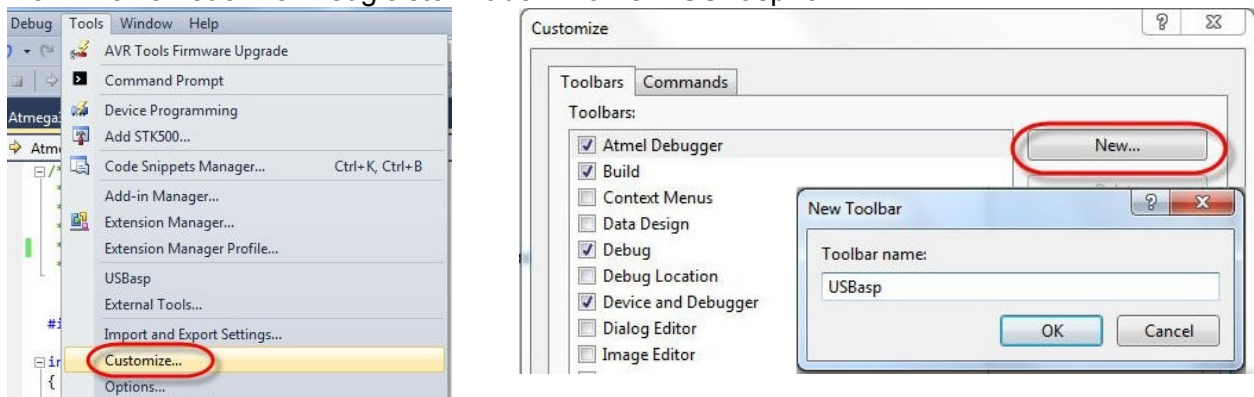




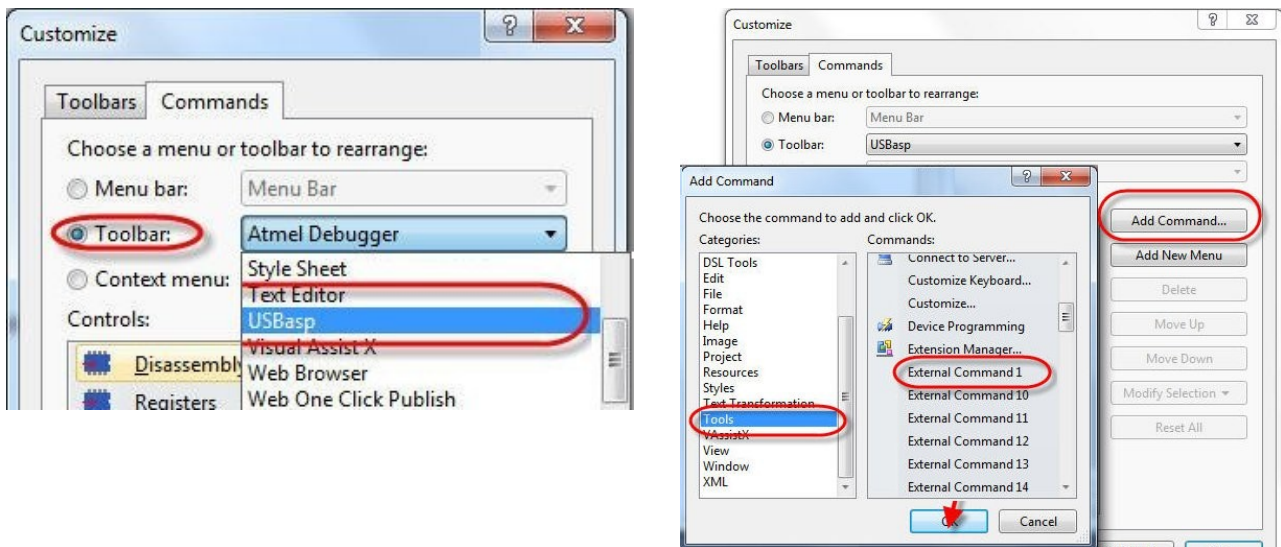
Vor dem Klick auf OK das Häkchen bei "Use Output window" setzen. Ab jetzt taucht der USBasp-Programmer im Tools-Menü auf und kann benutzt werden.



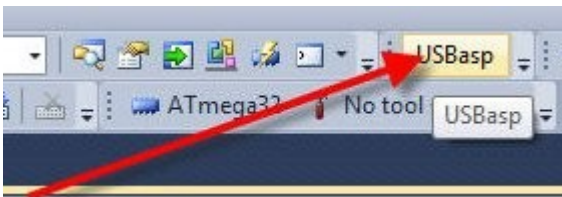
Zur einfacheren Handhabung lässt sich auch ein eigener Button für den Programmer erzeugen. Dazu klickt man im Menü "Tools" auf "Customize..." und legt unter unter dem Reiter "Toolbars" mit "New..." eine neue Werkzeugleiste mit dem Namen "USBasp" an.



Über den Reiter "Commands" wählt man in der Zeile "Toolbar" den "USBasp" aus. Nach einem Klick auf "Add Commands..." erscheint ein neues Fenster. Dort wählt man die Kategorie "Tools", klickt auf "External Command 1" und anschließend auf OK.



Jetzt erscheint in der Toolbar der USBasp-Programmier-Button. Zum Übertragen eines Programms auf den Mikrocontroller braucht er nur noch angeklickt zu werden.



Beim Wechsel des Mikrokontrollertyps bitte daran denken, auch die Typenbezeichnung unter "Tools -> External Tools -> USBasp, Arguments" entsprechend zu ändern.

Diese Anleitung basiert auf dem bei [www.mikrocontroller.net](http://www.mikrocontroller.net) veröffentlichten Konzept von Martin Carreño, welcher sie dankenswerterweise unter freie Lizenzen gestellt hat (GPL, wahlweise V2 oder V3).

Die verwendeten Bildschirmkopien zitieren – zum Teil simplifiziert – die Bedienung der Software *Atmel Studio*. Sämtliche Rechte liegen bei Atmel Corporation, San Jose, California, USA.

Alle Angaben geschahen nach bestem Wissen und Gewissen. Es wird um Verständnis dafür gebeten, dass eine Haftung für Schäden, die durch Verwendung dieser Beschreibung entstehen, nicht übernommen wird.