

# guloprog S mk II

von guloshop.de



## Willkommen bei guloprog S – dem Mikrocontroller-Programmer für Steckplatinen!

Mit dem guloprog S besitzen Sie einen vollwertigen USBasp Programmer ähnlich dem Fischl-Standard (<http://www.fischl.de/usbasp>). Im Gegensatz zu den gängigen Varianten, welche einen ATmega8 als Herzstück verwenden, arbeitet im guloprog S ein neuerer und kleinerer Mikrocontroller-Typ, der ATtiny85. Die Schaltung synchronisiert sich selbst über den USB-Takt des Computers und benötigt kein Schwingquarz als Zeitbasis. Dadurch kommt sie mit sehr wenigen Bauteilen aus und lässt sich im Handumdrehen auf einer Steckplatine aufbauen.

Sie haben die Bauteile für diesen Programmer bei guloshop.de käuflich erworben und unterstützen damit die Weiterentwicklung quelloffener Software. Selbstverständlich dürfen Sie diese Anleitung aber auch dazu verwenden, einen Programmer mit Teilen aus Ihrer Bastelkiste aufzubauen!

Folgende Mikrocontroller lassen sich mit guloprog S programmieren:

ATtiny 13, 13A, 13V, 25, 25V, 45, 45V, 85, 85V  
ATtiny 10, 20, 40 (mit optionalem DIP-Adapter)  
ATtiny 2313, 2313A, 4313  
ATtiny 24, 24A, 44, 44A, 84, 84A  
ATtiny 26, 26L, 261A, 261V, 461A, 461V, 861A, 861V  
ATtiny 43U  
ATtiny 48, 88  
ATtiny 87, 167 (mit optionalem DIP-Adapter)  
ATmega 8, 8L, 48, 48A, 48P, 48PA, 88, 88A, 88P, 88PA  
ATmega 168, 168A, 168P, 168PA, 328, 328P  
ATmega 16, 16A, 32, 32A, 164A, 164P, 164PA, 324A, 324P, 324PA, 644, 644A, 644P, 644PA,  
1284, 1284P, 8535  
ATmega 162, 8515  
ATmega 32U2/DIP, 32U4/DIP

Viele nicht im DIP-Gehäuse erhältliche Atmel-Mikrocontroller werden ebenfalls unterstützt.

Zum vereinfachten Anschließen der zu programmierenden Mikrocontroller empfehlen wir Ihnen das guloboard, das ebenfalls bei guloshop.de erhältlich ist. Das durchgehende Farbschema für die Anschlussleitungen vermeidet Fehler und erlaubt einen raschen Aufbau.

## Lieferumfang

Vor dem Aufbau prüfen Sie bitte, ob alle Komponenten vorhanden sind:

- ein Mikrocontroller ATtiny85 (8 Anschlüsse, bereits programmiert)
- eine USB-Anschlussleitung mit USB-A-Stecker auf der einen und vier Steckerstiften für die Steckplatine auf der anderen Seite
- zwei Zener-Dioden für 3,6 Volt
- zwei Widerstände mit 100 Ohm (Farbringe braun, schwarz, braun, gold)
- ein Widerstand mit 2,2 kOhm (Farbringe rot, rot, rot, gold)
- ein Kondensator mit 100 nF (für den Standard-Aufbau nicht benötigt)
- drei Widerstände mit 470 Ohm (für den Standard-Aufbau nicht benötigt)
- sieben Anschlussdrähte für den zu programmierenden Mikrocontroller in den Farben rot (zwei Stück), schwarz, grün, weiß, blau und gelb

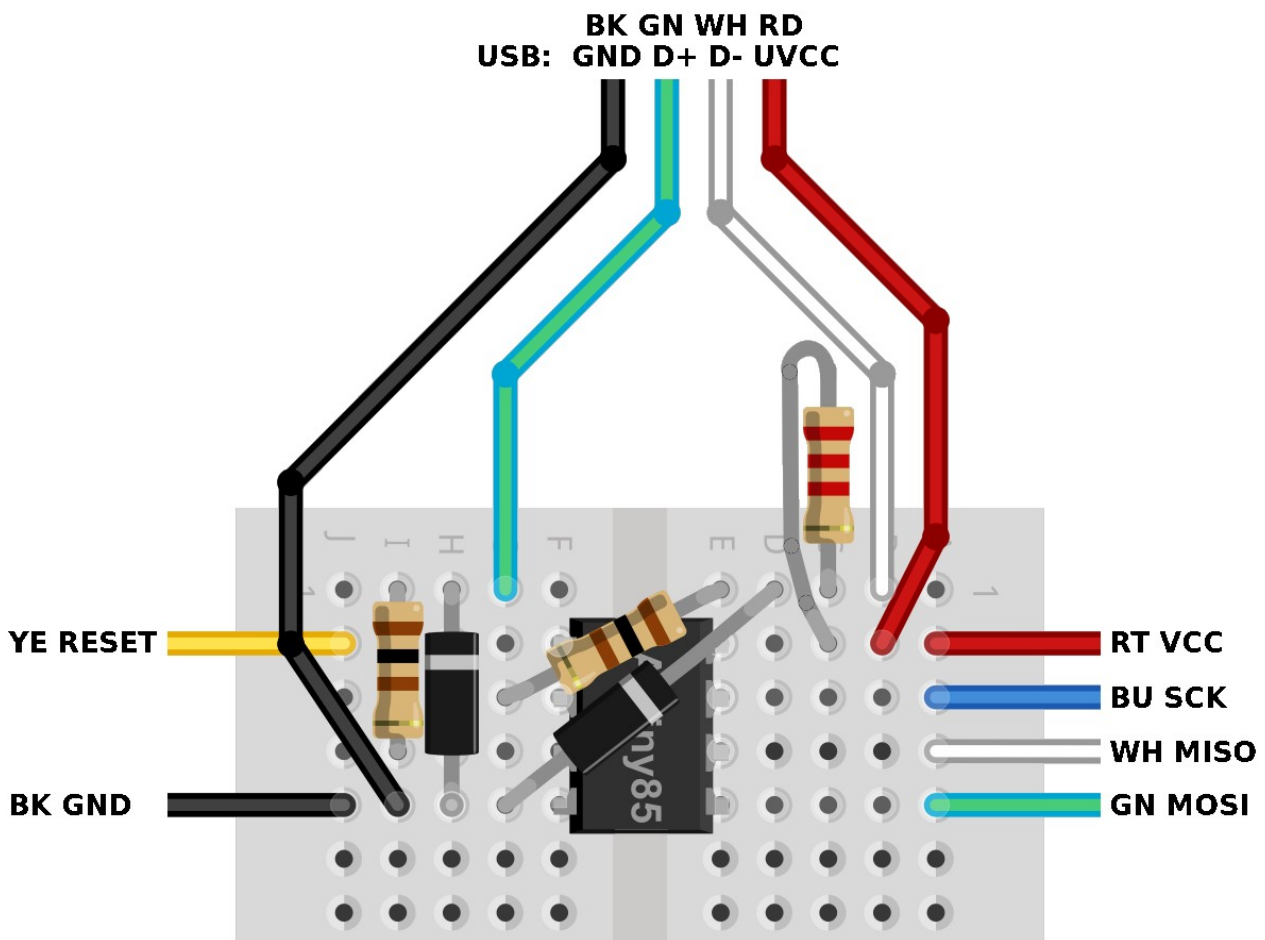
Bei der Grundversion gehört die Steckplatine nicht zum Lieferumfang, weil wir davon ausgehen, dass Sie bereits eine besitzen, wenn Sie sich einen Programmier "für die Steckplatine" beschaffen. Natürlich müssen Sie den Programmier nicht auf einer Steckplatine aufbauen, Sie können die Bauteile auch auf eine Lochrasterplatine löten, wenn Sie einen dauerhaften und stabilen Aufbau bevorzugen. Hierfür würden wir Ihnen dann aber eher unseren Bausatz guloprogramm (ohne S) empfehlen.

Ein Hinweis zu der vorliegenden Version II des guloprogramm S:

Bitte beachten Sie, dass sich alte und neue Version in der Anschlussbelegung des ATtiny85 unterscheiden. Für die ältere Version des guloprogramm S verwenden Sie bitte die zugehörige Aufbauanleitung in der Version 0.6 mit Stand vom 16.10.2013, welche Sie hier herunterladen können: [http://guloshop.de/f/guloprogramm1S\\_doc.pdf](http://guloshop.de/f/guloprogramm1S_doc.pdf)

## Zusammenbau

Setzen Sie die Bauteile so auf die Steckplatine wie auf dem folgenden Schema zu sehen ist. Bitte achten Sie darauf, dass alle Anschlussdrähte weit in die vorgesehenen Löcher gesteckt werden, so dass ein sicherer Kontakt besteht. Wichtig ist natürlich auch, dass die Anschlüsse nicht verwechselt werden und dass sich die Drähte der Bauteile nicht gegenseitig berühren! Aber diese Grundregel kennen Sie sicher, wenn Sie Steckplatinen verwenden. Beim Mikrocontroller und bei den Dioden müssen Sie auf die richtige Polung achten. Anschluss 1 des Mikrocontroller – das ist der in der Nähe des kleinen eingepprägten Kreises – befindet sich links oben. Die Kathoden der Dioden – das ist jeweils das Ende mit dem aufgedruckten Querstrich – befindet sich oben bzw. rechts oben.

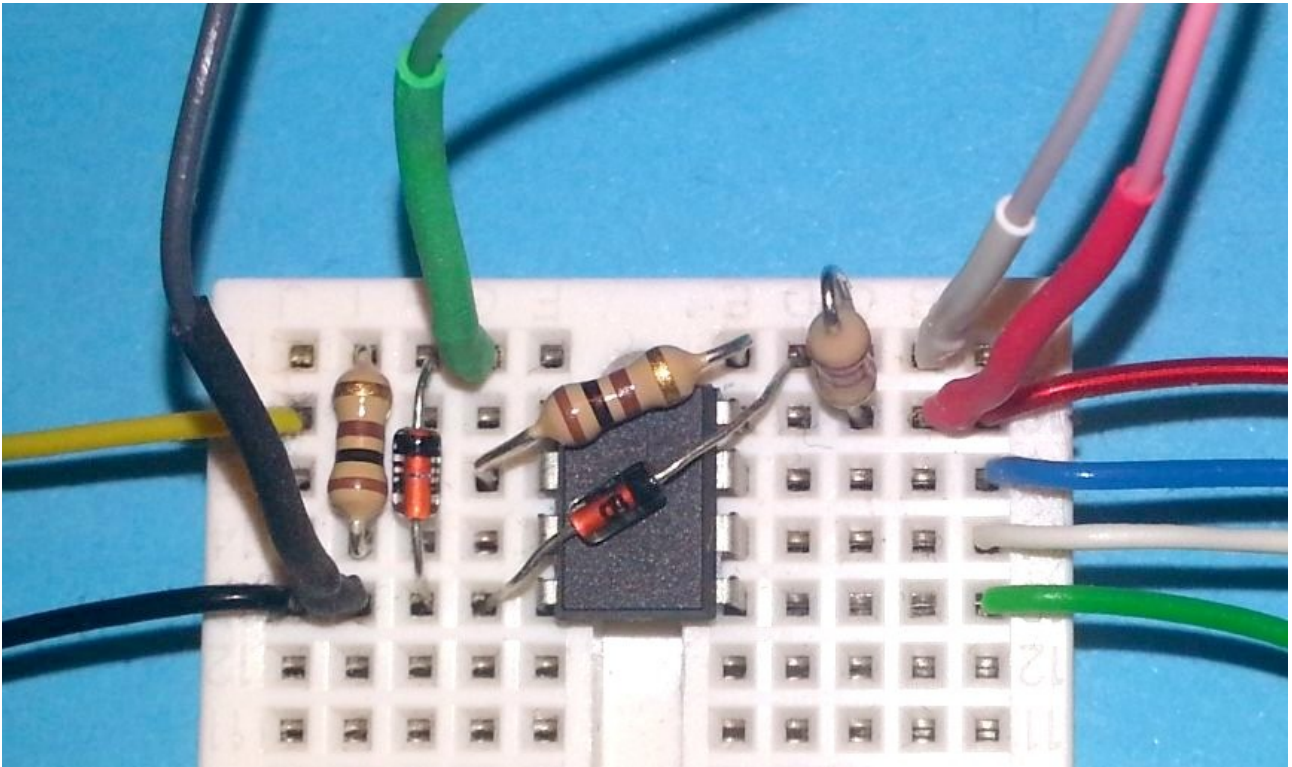


Im Bild sehen Sie die von oben ankommenden Anschlussleitungen des USB-Kabels. An den Seiten sind die Leitungen dargestellt, mit denen der Ziel-Mikrocontroller – das so genannte Target – programmiert wird. Diese Leitungen besitzen das gleiche Farbschema, das sie möglicherweise schon vom guloboard her kennen.

Zur besseren Erkennbarkeit auf Schwarz-Weiß-Ausdrucken – und zur besseren Erkennbarkeit für Menschen mit Farbschwäche – sind die internationalen Abkürzungen für die Farben mit angegeben: RD=rot, BK=schwarz, GN=grün, WH=weiß, BU=blau, YE=gelb.

Falls Sie einen Schaltplan benötigen, Sie finden ihn auf Seite 6.

Das folgende Bild zeigt den Aufbau als Foto:



Vor dem Anschließen an den Computer stellen Sie sicher, dass alle Bauteile korrekt und an der richtigen Stelle aufgesteckt sind und dass alle nichtisolierten Drähte berührungsfrei sind. Der Target-Mikrocontroller muss korrekt angeschlossen sein und darf keine Verbindung zur Ziellanwendung besitzen, er darf also nicht gleichzeitig in einer anderen Schaltung sitzen. Am besten, Sie ziehen ihn aus der Ziellanwendung heraus und setzen ihn alleine auf einen freien Platz der Steckplatine.

Nicht notwendig, aber dann empfohlen, wenn Sie sich bei der Anschlussbelegung des zu programmierenden Mikrocontrollers nicht sicher sind, ist die Verwendung der mitgelieferten 470-Ohm-Widerstände. Diese können Sie an Stelle der Leitungen für MOSI (grün), SCK (blau) und RESET (gelb) verwenden. Dadurch verringern Sie im Fall von Anschlussfehlern das Risiko von Schäden am ATtiny85 des Programmers. Ebenfalls nicht notwendig, aber in Einzelfällen hilfreich ist das Anschließen des Kondensators an VCC und GND: er wird mit seinem einem Anschlussbein mit VCC und mit seinem anderen Anschlussbein mit GND verbunden. In beinahe allen Fällen werden Sie aber den Programmer ohne diesen Kondensator betreiben können.

## Software

Im Internet werden Sie verschiedene Software finden, die zur Benutzung von USBasp-Programmieren geeignet ist. Wir empfehlen Ihnen avrdude. Bei Linux gehört avrdude meist zum Lieferumfang, es ist über das Software-Center aktivierbar, bei Windows leider nicht, hier müssen Sie das Programm selbst herunterladen – zum Beispiel von hier: <http://guloshop.de/f/avr/> Sie benötigen beide Dateien, avrdude.exe und avrdude.conf. Informationen zum Programm finden Sie bei <http://www.mikrocontroller.net/articles/AVRDUDE> und bei <http://www.nongnu.org/avrdude/user-manual/avrdude.html>

Möglicherweise kennen Sie das aber alles schon aus dem Einsteigerpaket gulostart.

Neben avrdude benötigen Sie im Fall von Windows noch einen Gerätetreiber – bei Linux ist dieser schon enthalten. Windows-Benutzer laden sich den Treiber bitte hier herunter:

[http://guloshop.de/f/usbasp\\_driver.zip](http://guloshop.de/f/usbasp_driver.zip) Er wird beim ersten Anschließen des Programmers vom System benötigt und muss dann installiert werden. Eine Anleitung dazu finden Sie hier: [http://guloshop.de/f/usbasp\\_driver.txt](http://guloshop.de/f/usbasp_driver.txt)

## Betrieb

Zum Programmieren eines Mikrocontrollers bauen Sie bitte zuerst den Programmer komplett auf und verbinden den zu programmierenden Mikrocontroller über die sechs farbigen Leitungen. Bei manchen Mikrocontrollern müssen Sie zusätzlich zu VCC noch AVCC anschließen, das ist jedoch im jeweiligen Datenblatt erklärt. Tipp: bei Verwendung des guloboards werden Sie durch das einheitliche Farbschema beim Anschließen der Leitungen unterstützt. Danach verbinden Sie den Programmer über das USB-Kabel mit Ihrem PC. Die Bedienung per avrdude ist über Linux und Windows dieselbe. Bei Linux kann es jedoch sein, dass Sie auf Grund des Berechtigungssystems vor den Befehl jeweils "sudo" schreiben müssen. Im Folgenden finden Sie ein paar Beispiele für den Einstieg.

Datei "beispiel.hex" auf den Mikrocontroller des Typs ATmega328P schreiben:

```
avrdude -c usbasp -B 12 -p m328p -U flash:w:beispiel.hex:i
```

Inhalt des Mikrocontroller mit der Datei "beispiel.hex" vergleichen:

```
avrdude -c usbasp -B 12 -p m328p -U flash:v:beispiel.hex:i
```

Inhalt des Mikrocontroller lesen und in die Datei "beispiel.hex" schreiben:

```
avrdude -c usbasp -B 12 -p m328p -U flash:r:beispiel.hex:i
```

Inhalt des Fuse-Lowbytes anzeigen:

```
avrdude -c usbasp -B 12 -p m328p -U lfuse:r:/dev/stdout:b
```

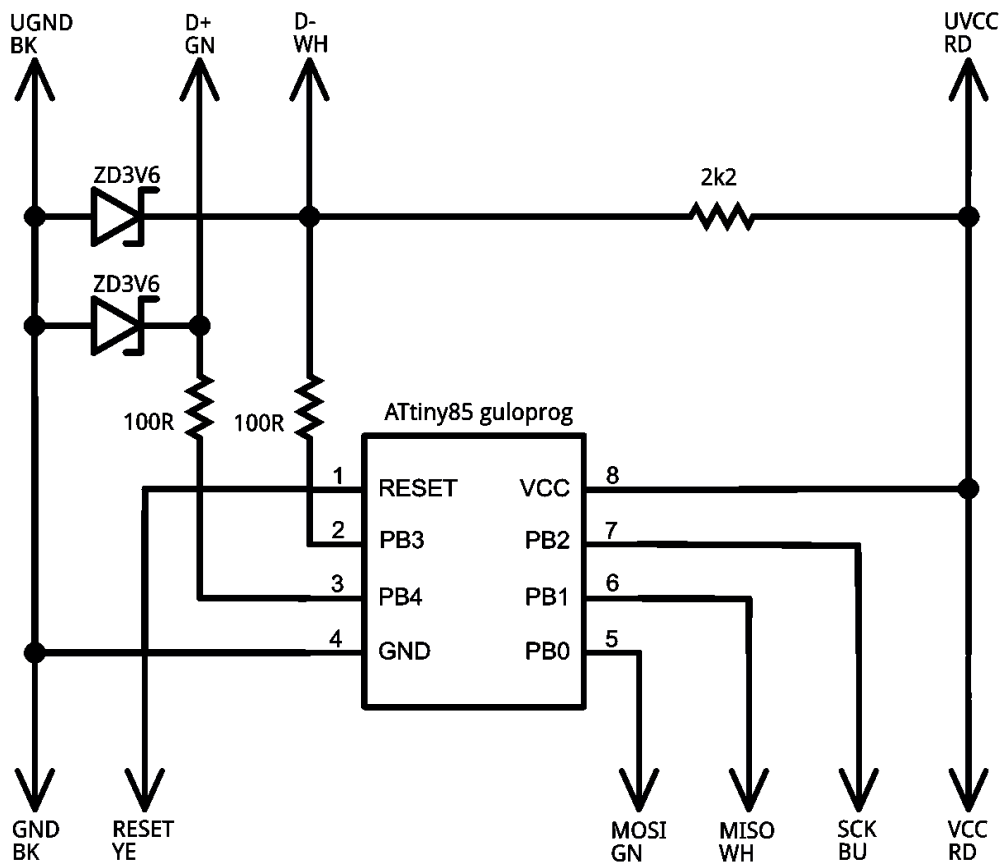
Inhalt des Low-Fusebytes neu schreiben:

```
avrdude -c usbasp -B 12 -p m328p -U lfuse:w:0b01100010:m
```

Am Ende des Programmiervorgangs trennen Sie bitte das USB-Kabel des Programmers von Ihrem Computer, bevor Sie die Schaltung berühren.

## Schaltplan und Nachbau

Wie eingangs schon geschrieben, dürfen Sie den Programmierer auch gerne mit eigenen Bauteilen zusammenbauen. Die Software ist quelloffen und im Wesentlichen nach GPL lizenziert. Genaueres zu den Lizenzen entnehmen Sie bitte den Beschreibungen in den Quelldateien von hier: [http://guloshop.de/f/sources/guloprogramm2\\_src\\_2\\_0.zip](http://guloshop.de/f/sources/guloprogramm2_src_2_0.zip)



Bitte beachten Sie, dass der mitgelieferte ATtiny85 erst dann wieder ausgelesen oder neu beschrieben werden kann, wenn das Fusebit "RSTDISBL" auf 1 gesetzt wurde. Hierzu wird ein HVSP-Fusebit-Programmer benötigt. Das heißt, mit einem normalen Programmierer lässt sich die Software für den Programmierer guloprogramm nur einmal auf den ATtiny85 übertragen, danach kann sie mit einfachen Mitteln weder ausgelesen noch verändert werden. Falls Sie damit keine Erfahrung haben, empfehlen wir Ihnen, sich den bereits programmierten ATtiny85 als "Ersatzteil" bei [guloshop.de](http://guloshop.de) zu beschaffen.

## Weitere Informationen

Wir haben für Sie ein paar Links zusammengestellt, die wir für besonders hilfreich halten:

<http://www.atmel.com/products/microcontrollers/avr/tinyAVR.aspx>

<http://www.atmel.com/products/microcontrollers/avr/megaAVR.aspx>

<http://www.mikrocontroller.net/articles/AVR-Tutorial>

[http://www.rn-wissen.de/index.php/AVR\\_Assembler\\_Einf%C3%BChrung](http://www.rn-wissen.de/index.php/AVR_Assembler_Einf%C3%BChrung)

<http://www.mikrocontroller.net/forum/mikrocontroller-elektronik>

<http://www.roboternetz.de/community/forums/20-AVR-Hardwarethemen>

<http://www.roboternetz.de/community/forums/56-Assembler-Programmierung>

## Danke

Ausdrücklich bedanken wollen wir uns bei

Thomas Fischl (<http://www.fischl.de/usbasp/>)

für die Basisarbeit der Entwicklung eines preiswerten Programmers für die USB-Schnittstelle,

Objective Development Wien (<http://www.obdev.at/products/vusb/index-de.html>)

für die Entwicklung einer Software-USB-Schnittstelle für AVR-Mikrocontroller,

Hannes Jochriem (<http://www.ehajo.de/bausätze>)

für seine ausgezeichneten Bausätze, die uns immer wieder auf gute Ideen bringen,

Fachhochschule Potsdam (<http://www.fh-potsdam.de/> <http://fritzing.org/>)

für die zur Dokumentation von Breadboard-Bestückungen genutzte Software *Fritzing*,  
deren Lizenzbestimmungen die oben abgedruckten schematischen Darstellungen unterliegen,

den Mitgliedern mehrerer Mikrocontroller-Communities  
für Hilfe, Unterstützung und Anregung,

sowie den Entwicklern von avra und avrdude.