

guloprogram

der Programmierer
von guloshop.de



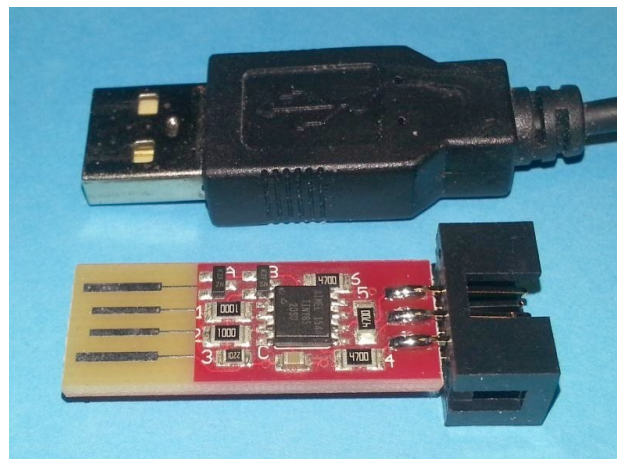
Funktionsumfang

Willkommen bei guloprogram, dem einfachen Programmier-Modul für AVR-Mikrocontroller!

Als Herzstück dieser Neuentwicklung wird ein leistungsstarker Mikrocontroller des Typs ATtiny85 eingesetzt. Dieser besitzt nur 8 Anschlüsse und ermöglicht dadurch die besonders geringen Abmessungen der Programmier-Platine. Einem Einbau in fertige Schaltungen sowie Prototypen steht buchstäblich nichts im Weg.

Mit einem fertig aufgebauten guloprogram lassen sich alle gängigen 8-Bit-AVR programmieren, denn der Programmierer beherrscht beide gängige Programmierverfahren (ISP und TPI):

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 (mit optionalem DIP-Adapter)
 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 weitere nicht im DIP-Gehäuse erhältliche Atmel-Mikrocontroller – beispielsweise der relativ neue ATtiny841 – werden ebenfalls unterstützt.

Zum Anschluss des jeweiligen Mikrocontrollertyps benötigen Sie noch einen entsprechenden Adapter bzw. ein Adapterkabel mit 6-poligem Stecker. Zahlreiche Bauanleitungen dazu finden Sie über die am Ende dieser Anleitung aufgeführten Web-Empfehlungen. Oder Sie entscheiden sich für den Universal-Adapter *guloboard G6*, welchen Sie ebenfalls in unserem Shop finden.

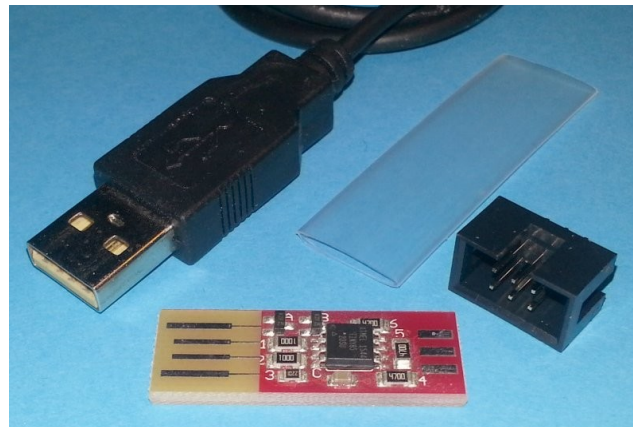
Version 2

Bitte beachten Sie, dass alle nachfolgenden Informationen nur für die Version 2 des Programmers guloprogrammer gültig sind (Programmer mit 6-poligem Anschluss). Sollten Sie guloprogrammer in der Version 1 besitzen (Programmer mit 10-poligem Anschluss), verwenden Sie bitte ausschließlich die Aufbauanleitung für die Version 1.

Lieferumfang

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

- ein USB-Kabel mit Typ-A-Stecker und offenem Ende (oder einer Typ-A-Kupplung)
- eine beidseitig beschichtete Platine, SMD-Bauteile bereits vorgelötet
- ein programmierter SMD-Mikrocontroller ATtiny85 (vorgelötet)
- neun SMD-Kleinteile (vorgelötet)
- ein 6-poliger IDC-Line-Stecker ("Wannenstecker") für Lötmontage
- ein Stück Schumpfschlauch zum Isolieren der fertigen Platine



Werkzeuge

Für den Zusammenbau des guloprogrammers benötigen Sie einen LötKolben, LötZinn und etwas LötErfahrung.

Für das Abisolieren des USB-Kabels verwenden Sie am besten ein kleines Messer oder einen Seitenschneider. Zum Festhalten der Leiter des USB-Kabels während des Anlötens empfiehlt sich eine kleine Zange.

Diese Werkzeuge gehören nicht zum Lieferumfang.

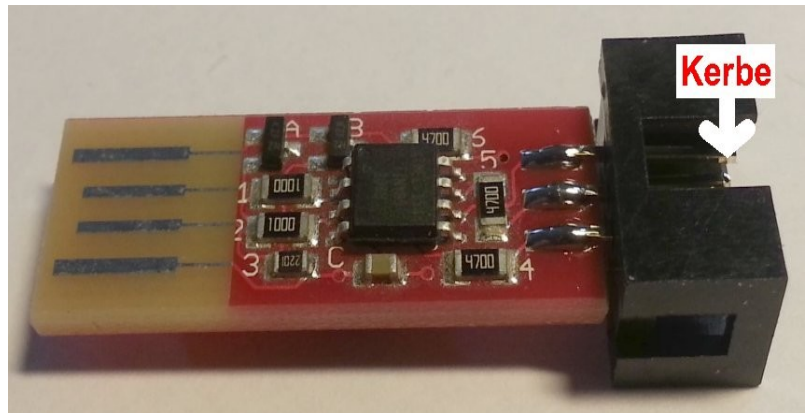


Zusammenbau

Legen Sie die Platine mit den Bauteilen nach oben quer vor sich hin, und zwar so, dass Sie die vier offenen Kontaktzungen nach links zeigen (siehe Abbildung).

Schieben Sie dann den 6-poligen Wannenstecker auf das rechte schmale Ende der Platine. Drei Kontakte des Wannensteckers befinden sich jetzt auf der Platinenoberseite, also auf der Seite, auf der auch die Bauteile sind, die anderen drei Kontakte

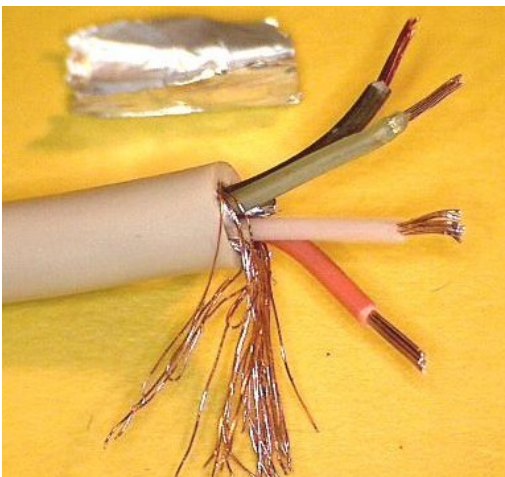
des Steckers befinden sich auf der Platinenunterseite. Achten Sie unbedingt darauf, dass sich die Kerbe des Wannensteckers ebenfalls oben befindet. Der Wannenstecker ist etwas breiter als die Platine. Richten Sie ihn mittig aus, so dass er auf seinen beiden Seiten gleich weit über den Platinenrand herausragt. Die Stifte des Wannensteckers dürfen kein Platinenbauteil berühren.



Löten Sie nun alle 6 Kontakte des Wannensteckers an der Platine fest, 3 auf der Platinenoberseite und 3 auf der Platinenunterseite.

Sie kennen das von besonders kleinen USB-Speichersticks: manche besitzen keinen regulären USB-Stecker, sondern man steckt das flache Plastik-Gehäuse des Sticks direkt in die USB-Buchse des Computers. Diese besondere Art eines USB-Steckers ist mit Vorsicht zu genießen, da sie nicht der Norm entspricht und nicht mit allen Geräten zusammenarbeitet – besonders dann, wenn die USB-Buchsen der Geräte ihrerseits von der Norm abweichen. An USB-Verlängerungskabeln beispielsweise finden nicht genormte Stecker oft erst dann richtig Kontakt, wenn der Blechmantel der jeweiligen Kabel-Buchse mit einer Zange etwas zusammengedrückt wurde.

Die Platine des guloprogramm Programmers kann ebenfalls direkt als ein solcher USB-Stecker verwendet werden. Auf Grund der genannten Abweichung vom USB-Standard geschieht das jedoch stets auf eigene Gefahr. Um eine sichere Funktion zu gewährleisten, muss das beiliegende USB-Kabel eingesetzt werden. Im Folgenden wird beschrieben, wie dieses Kabel an die Platine anzuschließen ist.



Falls dem Programmer kein USB-Kabel mit offenem Ende, sondern ein USB-Verlängerungskabel beiliegt, schneiden Sie bitte das Ende mit der USB-Kupplung mit einem Seitenschneider ab.

Danach entmanteln Sie das offene Kabelende. Dazu ritzen Sie es ca. 1 cm von seinem Ende entfernt vorsichtig rundum ein. Schneiden Sie nicht zu tief, damit die darunter liegenden Adern nicht beschädigt werden. Entfernen Sie dann das 1 cm lange Stück des Mantels – das ist die äußere Isolierung. Falls das Kabel ein Geflecht aus dünnen Kupferdrähten besitzt (siehe Bild), Entflechten Sie diese auf der entmantelten Länge. Eine möglicherweise vorhandene Metallfolie schneiden Sie nahe am Mantel des Kabels ab, sie wird nicht benötigt.

Entfernen Sie nun von den Enden aller 4 freigelegten farbigen Adern jeweils 3 mm der Isolierung und verdrehen Sie jede der Litzenenden etwas.

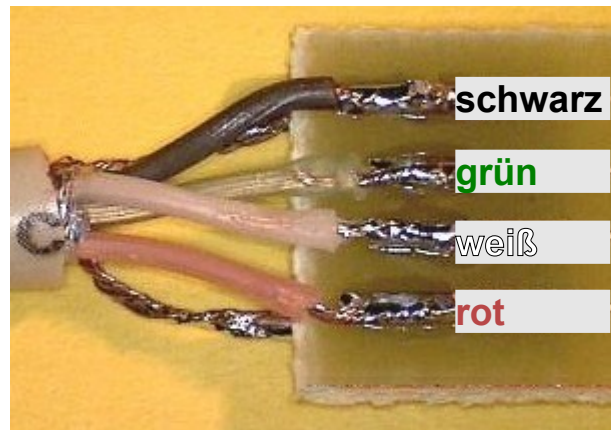
Später soll die Platine mit einem Schrumpfschlauch isoliert werden, damit sie gegen Berührung gesichert ist, wenn sie beispielsweise in ein Metallgehäuse eingebaut wird. Jetzt haben Sie die letzte Möglichkeit dies vorzubereiten: Schieben Sie den mitgelieferten Schrumpfschlauch über das offene Ende des USB-Kabels, am besten bis hin zum USB-Stecker, damit der Schlauch für die gleich folgenden Arbeiten nicht im Weg ist.

Schließen Sie jetzt die einzelnen Leitungen des USB-Kabels an, achten Sie dabei genau auf die Reihenfolge. Wenn Sie von der Bauteilseite aus auf die Platine schauen und die USB-Kontakte dabei nach links zeigen, müssen die Leitungen in dieser Reihenfolge angeschlossen werden (von oben nach unten):

schwarz (GND), grün (D+), weiß (D-), rot (VCC).

Orientieren Sie sich dabei an der Abbildung rechts. Für den Fall, dass das Kabel eine blaue Ader besitzt, gilt eine alternative Farbcodierung:

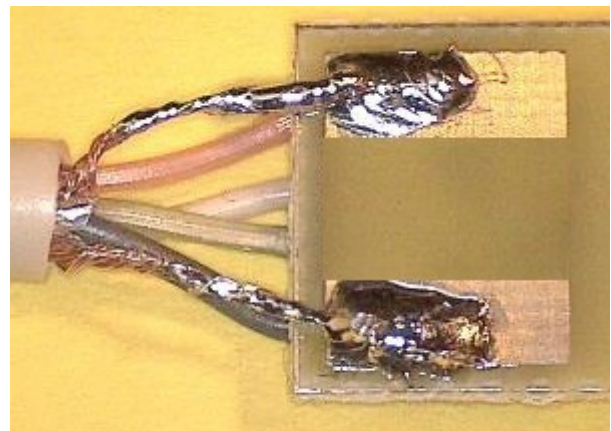
grün (GND), blau (D+), weiß (D-), rot (VCC).



Bei anderen Farbkombinationen oder Zweifeln an der Zuordnung empfiehlt es sich, die Belegung der Adern mit einem Durchgangsmessgerät zu prüfen.

Falls das Kabel mit einem Geflecht abgeschirmt ist, löten Sie dieses Geflecht an die beiden nicht verbundenen Leiterbahnen auf der Unterseite der Platine (siehe Abbildung rechts unten).

Aus elektrischer Sicht wird dieser Anschluss nicht benötigt, Sie verhindern mit dem Festlöten aber, dass das Geflecht andere Kontakte der Platine berührt. Es wird empfohlen, das Geflecht zusätzlich zu verzinnen (siehe Bild).



Zum Abschluss der Arbeiten müssen Sie den Schrumpfschlauch über die Platine schieben. Er lässt sich mit etwas Heißluft fixieren. Am besten warten Sie mit dem Fixieren aber noch, bis der Programmierer seine ersten Tests bestanden hat.

Bitte prüfen Sie zum Schluss noch einmal, ob alles korrekt zusammengebaut wurde.

Insbesondere darf ein nicht isoliertes Geflecht des USB-Kabels keine anderen Anschlüsse oder Bauteile der Platine berühren können. Fehler im Aufbau können im ungünstigsten Fall Beschädigungen an angeschlossenen Geräten zur Folge haben.

Wenn alles in Ordnung ist, wird es Zeit für einen kleinen Test.

Treiber-Installation

Für den Betrieb des Programmers wird ein entsprechender Gerätetreiber benötigt. Sollten Sie Ubuntu-Linux verwenden, können Sie diesen Absatz getrost überspringen, denn dieses Betriebssystem bringt alle üblichen Gerätetreiber von Haus aus mit – so auch den Treiber für den Programmer. Im Fall von Windows ist das nachträgliche Installieren von Treibern üblich, Sie kennen vermutlich diese in Einzelfällen langwierige Prozedur. Machen Sie sich aber keine Sorgen, in aller Regel ist die Installation des richtigen Treibers für den Programmer schnell erledigt. Sie finden alles, was Sie dazu benötigen, in dieser kleinen Anleitung:

http://guloshop.de/f/usbsp_driver.txt

Erster Test

Als Software für die Benutzung des Programmiers empfehlen wir *avrdude*. Als Anwender von Ubuntu-Linux erhalten Sie dieses Programm wie üblich bequem über das Software-Center. Wenn Sie wollen, können Sie es auch per Terminal installieren: `sudo apt-get install avrdude`
Falls Sie Windows als Betriebssystem verwenden, müssen Sie *avrdude* zuerst herunterladen.
Legen Sie dazu ein eigenes Verzeichnis an und speichern Sie diese beiden Dateien dort hinein:

`http://guloshop.de/f/avr/avrdude.exe`

`http://guloshop.de/f/avr/avrdude.conf`

Öffnen Sie dann die Kommandozeile und springen mit dem Kommando `cd` in das neue Verzeichnis.

Schließen Sie nun den Programmierer an die USB-Schnittstelle Ihres Computers an.

Achten Sie darauf, dass der Programmierer zu keiner Zeit Berührung zu leitenden Gegenständen hat. Die Platine des Programmiers ist erst dann isoliert, wenn Sie später den Schrumpfschlauch angebracht haben. Kurzschlüsse können zu Beschädigungen führen.

Als ersten Test geben Sie das folgende (harmlose) Kommando für *avrdude* ein:

```
avrdude -c usbasp -p t85 -B 60 -U lfuse:r:-:b
```

Dieses Kommando liest normalerweise das Fuse-Low-Byte des angeschlossenen Mikrocontrollers und zeigt es am Bildschirm an.

Ein Hinweis für Linux-Anwender: In Abhängigkeit von den eingestellten Berechtigungen des verwendeten Benutzerkontos kann es sein, dass dieses Kommando nur mit vorangestelltem "sudo" vom System akzeptiert wird.

Da bis jetzt lediglich der Programmierer, aber kein Ziel-Mikrocontroller (so genanntes "Target") angeschlossen ist, erhalten Sie – neben anderen Fehlermeldungen – diese Reaktion:

```
"avrdude: error: programm enable: target doesn't answer. 1"
```

Das ist in Ordnung! Es zeigt, dass der Programmierer erkannt wurde und lediglich im Moment kein zu programmierender Mikrocontroller angeschlossen ist.

Für den Fall, dass Sie stattdessen die Meldung

```
"avrdude: error: could not find USB device "USBasp" with vid=0x16c0 pid=0x5dc"
```

erhalten, prüfen Sie bitte alle Verbindungen. Falls Sie die Platine direkt in eine USB-Buchse gesteckt haben, kann es sein, dass beide Steckverbinder nicht zu 100 % kompatibel sind. Es könnte ebenfalls sein, dass Sie die Platine versehentlich verkehrt herum eingesteckt haben.

In seltenen Fällen stören sich USB-Geräte gegenseitig. Das kann dazu führen, dass der Programmierer vom Computer nicht oder nicht zuverlässig erkannt wird. Um einen solchen Fehler auszuschließen, können Sie versuchsweise die USB-Stecker alle anderen angeschlossenen Geräte aus den Buchsen am Computer herausziehen oder auch verschiedene USB-Buchsen für den Programmierer ausprobieren.

Wenn alles funktioniert, entfernen Sie bitte wieder die Verbindung zwischen Programmierer und dem USB-Anschluss ihres Computers. Das sollten Sie aus Sicherheitsgründen auch in Zukunft immer dann tun, wenn Sie den Programmierer gerade nicht verwenden.

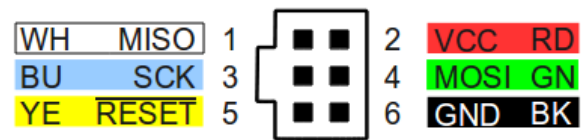
Alternatives Programmierverfahren: TPI

Neben dem üblichen Programmierverfahren ISP beherrscht der Programmierer auch das Verfahren TPI, welches für die Programmierung der kleinsten ATtiny-Mikrocontroller benötigt wird – beispielsweise für den ATtiny10. Welches Programmierverfahren vorgesehen ist, können Sie dem Datenblatt des jeweiligen Mikrocontrollers entnehmen. Meist sieht man es auch schon an der Pinbelegung: Anschlüsse wie TPIDATA und TPICLK deuten auf TPI hin.

Für TPI wird eine Leitung weniger benötigt als für das herkömmliche ISP-Verfahren. Zum Programmieren eines Mikrocontrollers mittels TPI gilt die folgende Anschlussbelegung:

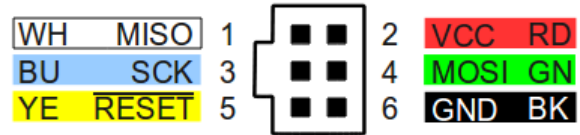
Wie üblich müssen VCC, GND und RESET (Pins 2, 6, 5) des Programmers mit den jeweiligen Pins des Mikrocontrollers verbunden werden.

Der MOSI-Anschluss des Programmers (Pin 4) wird mit TPIDATA des Mikrocontrollers verbunden, der SCK-Anschluss des Programmers (Pin 3) mit TPICLK des Mikrocontrollers. Der MISO-Anschluss wird nicht verbunden.



Alternative Verwendungen

Wenn Sie später hauptsächlich Mikrocontroller programmieren, die sich auf einer Steckplatine befinden, brauchen Sie natürlich den 6-poligen Wannenstecker nicht auflöten. Stattdessen schließen Sie verschiedenfarbige Drähte für die sechs Programmierleitungen an. Als Auswahl für die Farben schlagen wir das guloboard-Schema vor (siehe rechts). Die Anschlüsse mit den Nummern 1, 3 und 5 finden Sie auf der Platinenoberseite, die Anschlüsse 2, 4 und 6 auf der Unterseite.



Natürlich können Sie den Programmer auch in eine größere Schaltung integrieren: kleben Sie die kleine Platine des Programmers auf die Platine Ihrer Schaltung. Achten Sie dabei darauf, dass sich an der betreffenden Stelle Ihrer Platine keine Leiterbahnen befinden.

Eine dritte Möglichkeit besteht darin, den Programmer gar nicht als Programmer zu verwenden, sondern für beliebige andere Zwecke zu nutzen. Beispielsweise kann man damit eine USB-Seriell-Schnittstelle realisieren – oder eine Messdatenerfassung unter Nutzung des im ATtiny85 vorhandenen Analog-Digital-Wandlers. Dazu sind allerdings vertiefte Programmierkenntnisse im AVR-Bereich erforderlich.

Falls Sie vorhaben, guloprogrammer nicht als Programmer sondern alternativ zu nutzen, bestellen Sie ihn bitte "unprogrammiert". Nur dann können Sie ihn mit einfachen externen Programmern neu beschreiben.

Zum Neuprogrammieren des Programmers benötigen Sie einen weiteren Programmer, den Sie 1:1 anschließen:

<u>guloprogrammer</u>	<u>externer Programmer</u>
VCC	VCC
GND	GND
MOSI	MOSI
MISO	MISO
SCK	SCK
RESET	RESET

Stellen Sie dabei sicher, dass die MISO-Leitung einen zusätzlichen Vorwiderstand von mindestens 470 Ohm besitzt. Am besten bauen Sie sich einen entsprechenden Adapter.

Wenn Sie guloprogrammer in seinem normalen Auslieferungszustand, also bereits programmiert erhalten haben, benötigen Sie einen HVSP-Programmer, um den RESET-Anschluss wieder zu aktivieren. Hierfür können Sie zum Beispiel den bei guloshop.de erhältlichen HVSP-Fusebit-Programmer verwenden. Dieser wird wie folgt angeschlossen:

<u>guloprogrammer</u>	<u>externer HVSP</u>	<u>Pin der HVSP-Fassung</u>
VCC	VCC	8
GND	GND	4
MOSI	SDI	5
MISO	SII	6
SCK	SDO	7
USB D-	SCI	2

Stellen Sie dabei sicher, dass die SII-Leitung einen zusätzlichen Vorwiderstand von mindestens 470 Ohm besitzt (beim HVSP Fusebit Programmer von guloshop.de bereits vorhanden).

Betrieb als Signalwandler

Die seit Februar 2013 ausgelieferten Programmer-Module erhalten eine Zusatzfunktion, die es erlaubt, sie nicht nur als Programmer, sondern auch als Signalwandler zu einzusetzen. Dadurch können mit dem Computer per USB-Anschluss elektrische Digital- und Analog-Signale gelesen, gespeichert und verarbeitet werden. In umgekehrter Richtung können digitale Signale ausgegeben und pulsweitenmodulierte Signale erzeugt werden.

Natürlich sind Abtastrate, Spannungsbereich und Genauigkeit des Wandlers begrenzt, der verwendete Mikrocontroller ATtiny85 ersetzt weder ein professionelles Signalaufzeichnungsgerät noch einen professionellen Signalgenerator. Dennoch wird dieser Signalwandler für viele Einsatzfälle ausreichen.

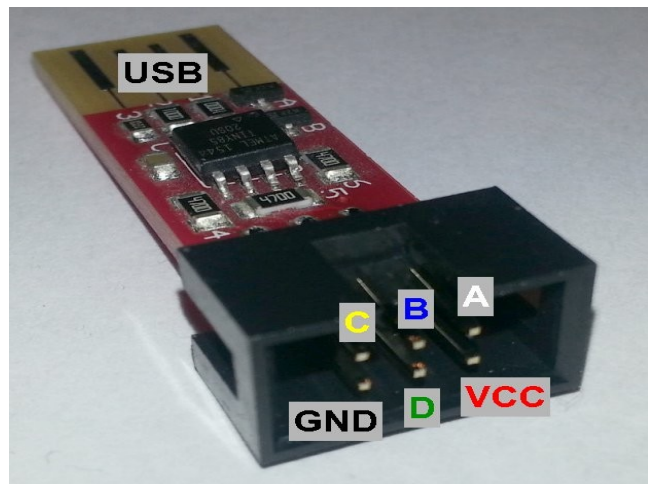
Technische Daten:

- 4 Signalleitungen, die jeweils als Ein- oder Ausgang genutzt werden können
- bis zu 4 digitale TTL-Eingänge (5 Volt)
- bis zu 3 analoge Eingänge (Messbereich 0 bis 1,1 Volt in 1024 Schritten)
- bis zu 4 digitale TTL-Ausgänge (5 Volt)
- bis zu 4 PWM-Ausgänge (161 Hz, 100 Stufen)

Die Signalleitungen im Einzelnen:

- A: Eingang digital/analog, Ausgang 0 Ohm, max. 30 mA
- B: Eingang digital/analog, Ausgang 470 Ohm, max. 10 mA
- C: Eingang digital/analog, Ausgang 470 Ohm, max. 2 mA
- D: Eingang digital, Ausgang 470 Ohm, max. 10 mA

Die Leitungen B, C und D haben wegen ihres integrierten Widerstands den Vorteil, dass dort direkt jeweils eine LED angeschlossen werden kann (bei Leitung C eine Low-Power-LED). Leitung A liefert die maximale Ausgangsleistung und besitzt deswegen keinen Vorwiderstand. Hier ist besonders auf korrekte Beschaltung zu achten. Bitte berücksichtigen Sie auch die genaueren Angaben im Datenblatt des Mikrocontrollers ATtiny85: www.atmel.com/devices/ATTINY85.aspx



Falls eine stabile Spannungsversorgung nicht garantiert werden kann – beispielsweise, weil sehr lange USB-Kabel verwendet werden – empfiehlt es sich, einen zusätzlichen Kondensator auf die Programmer-Platine aufzulöten, um kleine Spannungsschwankungen abzuf puffern. Nehmen Sie dazu einen Elektrolytkondensator mit einer Kapazität von 4,7 oder 10 µF und schließen Sie ihn an die platinenseitigen Stifte VCC (+) und GND (–) des Wannensteckers an. Achten Sie dabei unbedingt auf richtige Polung, Elektrolytkondensatoren dürfen nicht falsch herum angeschlossen werden.

Zur Bedienung der Zusatzfunktionen über die USB-Schnittstelle können Sie eine eigene Software entwickeln oder unser PC-Programm guloprogio herunterladen (Beschreibung siehe nächste Seite).

Software guloprogramm

Mit Hilfe dieser Software haben Sie Zugriff auf die Signalwandler-Funktionen Ihres Programmers.

Wichtig: Stellen Sie sicher, dass kein Mikrocontroller an den Programmer angeschlossen ist, solange Sie die Signalwandlerfunktionen verwenden.

guloprogramm steht für verschiedene Betriebssysteme zum Download zur Verfügung:

www.guloshop.de/f/guloprogramm32 oder www.guloshop.de/f/guloprogramm64 (für Linux 32 oder 64 Bit)

www.guloshop.de/f/guloprogramm_mac64 (für Mac ab 10.7, 64 Bit)

www.guloshop.de/f/guloprogramm.exe (für Windows, Beta-version)

Gestartet wird guloprogramm über die Kommandozeile. Das Programm arbeitet als "Filter", das heißt, es akzeptiert Ausgaben eines per Pipe vorgeschalteten Programms als Eingaben und kann selbst Daten für ein nachgeschaltetes Programm liefern. Wenn Sie guloprogramm ohne vor- oder nachgeschaltete Programme aufrufen, wartet es auf Ihre Eingabe. Mit diesen Anweisungen können Signalausgänge geschaltet werden:

- %=A0** schaltet Ausgang A auf 0 Volt.
- %=A1** schaltet Ausgang A auf 5 Volt.
- %=a50** aktiviert PWM für Ausgang A und wechselt in schneller Folge (ca. 161 Hz) zwischen 5 Volt und 0 Volt. Der 5-Volt-Anteil beträgt dabei 50 %.
- %=a90** tut das Gleiche, jedoch beträgt der 5-Volt-Anteil in diesem Fall 90 %.

An Stelle von A bzw. a können auch andere Leitungen ausgewählt werden: B, C, D bzw. b, c, d.

Um Signale zu empfangen, können Sie die gewünschten Leitungen als Eingang definieren:

- %=A-1** schaltet Leitung A als Eingang.
- %=A-2** schaltet Leitung A als Eingang und aktiviert zusätzlich einen Pull-Up-Widerstand.

Um digitale oder analoge Signale an den Eingangsleitungen abzufragen, müssen diese wie folgt angesprochen werden:

- %A** gibt den digitalen Zustand der Leitung A aus (0 oder 1).
- %a** gibt den analogen Zustand der Leitung A aus (0 bis 1023 für 0 bis ca. 1,1 Volt).
- % a** wie %a, jedoch mit führenden Leerzeichen.
- %0a** wie %a, jedoch mit führenden Nullen.

An Stelle von A bzw. a können auch andere Leitungen ausgewählt werden: B, C, D bzw. b, c. Die Leitung D steht als Eingang nur digital und nicht analog zur Verfügung.

Das Programm guloprogramm akzeptiert weitere Eingaben bzw. Eingabekombinationen:

Eingangsspannung %a und %b

schreibt den angegebenen Text einschließlich der Werte.

%% schreibt ein Prozentzeichen.

%; schreibt ein Tabulatorzeichen.

%... am Zeilenende bewirkt, dass diese Ausgabeanweisung ständig wiederholt wird.

%. beendet das Programm.

guloprogramm kann auch im nicht-interaktiven Modus gestartet werden. Es wartet dann nicht auf eine Eingabe, sondern führt die nach **-c** geschriebene Anweisung sofort aus. Zum Beispiel:

guloprogramm -c %=D1 schaltet Ausgang D sofort auf 5 Volt.

Mit **-h** zeigt das Programm einen kurzen Hilfetext mit weiteren Funktionen: **guloprogramm -h**

Weitere Informationen

Falls Sie noch keine oder sehr wenig Erfahrung mit der Programmierung von Mikrocontrollern haben, empfehlen wir Ihnen das Tutorial unseres Einsteiger-Pakets *gulostart*. Auch im Internet finden Sie zum Teil hervorragende Tutorials. Die beinahe wichtigste Informationsquelle bei der Programmierung von Mikrocontrollern ist das jeweilige Datenblatt:

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

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

Bei Fragen finden Sie Hilfe in einschlägigen Internet-Communities. Besonders empfohlen seien:

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

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>

Falls Sie bereits Experte sind und die Software des Programmers selbst verändern oder weiterentwickeln möchten, finden Sie den Quellcode einschließlich der zugehörigen Lizenzinformationen hier: <http://guloshop.de/f/sources/>

Danke

Ausdrücklich bedanken wollen wir uns bei

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

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/Bausaetze>)

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

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

sowie den Entwicklern von avra und avrdude.