

# GRBL

Einführung in "GRBL" - einer Firmware zur Steuerung von Lasergravierern, CNC-Fräsen, Kameraslidern u.Ä.



HACKERSPACE  
BREMEN E.V.

- Bremen, 15.12.2015

make | hack | modify

# Inhalt

- Was ist GRBL
- Überblick Software & Gcode
- GRBL einrichten und konfigurieren
- Elektronik & Hardware
- Schrittmotortreiberstufen
- Inbetriebnahme „cnc-shield“
- Dinge, die zu beachten sind
- Linkliste
- Feedback

# Was ist GRBL

GRBL ist eine Firmware für den Arduino (Uno/ATMega328)  
Ab 2009 vom Norweger Simen Svale Skogsrud entwickelt

Aktuelle Version : 0.9j (unter GPLv3 Lizenz)

GRBL interpretiert standardisierte Gcode-Anweisungen von der seriellen Schnittstelle und erzeugt daraus entsprechende Steuerimpulse für Schrittmotortreiberstufen.

Es werden komplexe mathematische Berechnungen, z.B. für Kreisbahnen und Beschleunigungsrampen durchgeführt. Ausserdem werden permanent Eingangssignale abgefragt und verarbeitet.

# Überblick Software & Gcode

GRBL kann über die Arduino-IDE angepasst (z.B. für abweichende PIN-Anordnung) und kompiliert werden, es gibt aber auch fertige HEX-Files zur sehr einfachen Übertragung mit dem Standardbootloader vom Arduino.

Konfiguration & Abarbeitungsbefehle werden über die serielle Schnittstelle angewiesen (z.B. über jedes Terminalprogramm, Baudrate dabei beachten).

Zeichnungsdateien im Vektor-/CAD-Format (z.B. svg, dxf, dwg, iges) müssen erst per CAM-Prozessor in „Gcode“ gewandelt werden.

„GCode“ ist eine Maschinen Anweisungssprache im Klartextformat, die ab den 1960ern entwickelt wurde - genormte Anweisungen werden dann zeilenweise von Maschinensteuerungen abgearbeitet. Dieser GCode wird auch im Bereich 3D-Druck benutzt.

Beispiel:

```
G00 X50 Y50 Z10      (fahre im Eilgang auf Koordinate XYZ)
G01 X100 Y100        (im Normalgang auf Koordinate XY, bleibe dabei bei Z10)
G01 Z-10             (senke Z um 10mm)
```

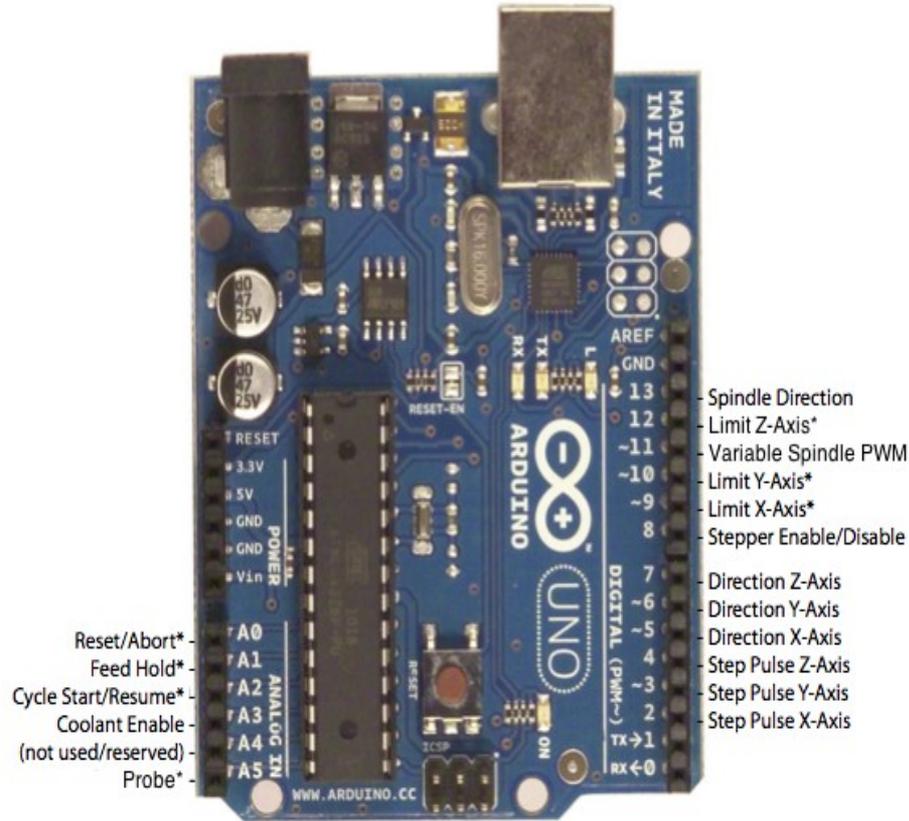
Zur komfortablen Befehlsübertragung benutzt man sogenannte „Gcodesender“, die es in sehr unterschiedlichen Ausführungen unter verschiedenen Betriebssystemen (oder betriebssystemübergreifend) gibt.

# GRBL einrichten und konfigurieren

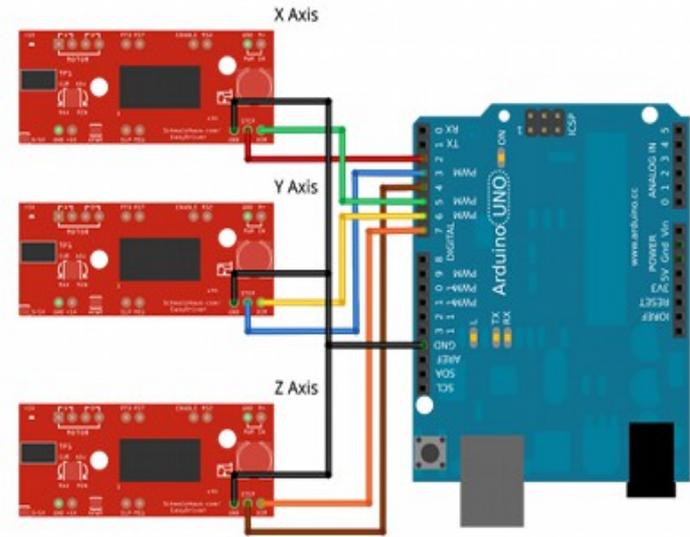
```
Grbl 0.9j ['$' for help]
$$ (view Grbl settings)
$# (view # parameters)
$G (view parser state)
$I (view build info)
$N (view startup blocks)
$x=value (save Grbl setting)
$Nx=line (save startup block)
$C (check gcode mode)
$X (kill alarm lock)
$H (run homing cycle)
~ (cycle start)
! (feed hold)
? (current status)
ctrl-x (reset Grbl)
```

```
Grbl 0.9j ['$' for help]
$0=10 (step pulse, usec)
$1=25 (step idle delay, msec)
$2=0 (step port invert mask:00000000)
$3=0 (dir port invert mask:00000000)
$4=0 (step enable invert, bool)
$5=0 (limit pins invert, bool)
$6=0 (probe pin invert, bool)
$10=3 (status report mask:00000011)
$11=0.010 (junction deviation, mm)
$12=0.002 (arc tolerance, mm)
$13=0 (report inches, bool)
$20=0 (soft limits, bool)
$21=0 (hard limits, bool)
$22=0 (homing cycle, bool)
$23=0 (homing dir invert mask:00000000)
$24=25.000 (homing feed, mm/min)
$25=500.000 (homing seek, mm/min)
$26=250 (homing debounce, msec)
$27=1.000 (homing pull-off, mm)
$100=250.000 (x, step/mm)
$101=250.000 (y, step/mm)
$102=250.000 (z, step/mm)
$110=500.000 (x max rate, mm/min)
$111=500.000 (y max rate, mm/min)
$112=500.000 (z max rate, mm/min)
$120=10.000 (x accel, mm/sec^2)
$121=10.000 (y accel, mm/sec^2)
$122=10.000 (z accel, mm/sec^2)
$130=200.000 (x max travel, mm)
$131=200.000 (y max travel, mm)
$132=200.000 (z max travel, mm)
```

# Elektronik und Hardware



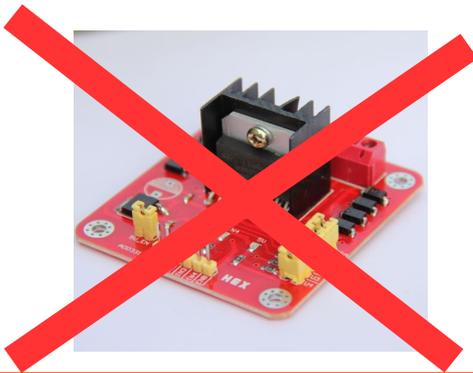
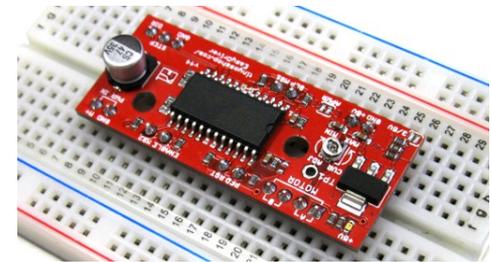
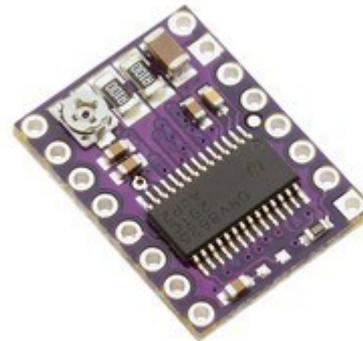
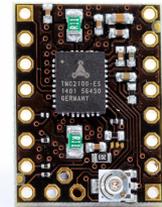
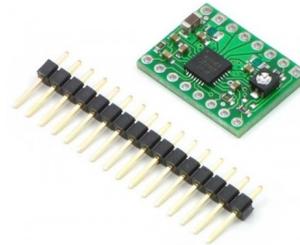
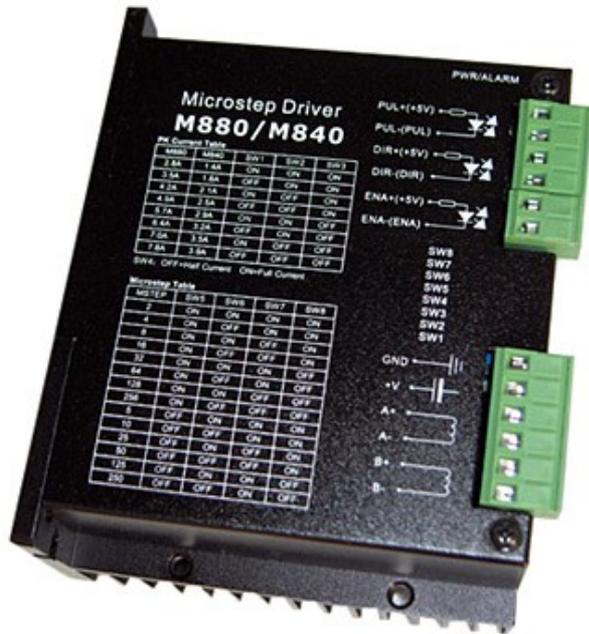
\* - Indicates input pins. Held high with internal pull-up resistors.



Bildquelle : <https://github.com/grbl/grbl/wiki/Connecting-Grbl>



# Schrittmotortreiberstufen





# Dinge, die zu beachten sind

- schaut bei der Inbetriebnahme nochmal Alles gründlich durch
- achtet auf eine ausreichende und einwandfreie Stromversorgung
- bei größeren und teuren Geräten unbedingt eine Notabschaltung einplanen
- achtet auf ausreichende Schutzmaßnahmen (für Mensch & Bauteile)
- Endschalter sind kein Muß, sind aber sinnvoll (z.B. für „homing“)
- Treiberstufen reagieren sehr sensibel auf Rückspannung & Motor „ziehen“
- sichert und dokumentiert Euch die Einstellungen für die Geräte
- ab V0.9 - „Z-Axis limit input on D11 has swapped with spindle enable D12 to support variable spindle PWM output“

# Linkliste

Funktion	Name	Link
Übertragung	Universal GCodesender	<a href="https://github.com/winder/Universal-G-Code-Sender">https://github.com/winder/Universal-G-Code-Sender</a>
CAD	Sketchup	<a href="https://www.sketchup.com/de">https://www.sketchup.com/de</a>
Plugin	Sketchup dxf/stl-export	<a href="http://www.guitar-list.com/download-software/">http://www.guitar-list.com/download-software/</a>
Übertragung	Serialcom CNC	<a href="http://www.serialcominstruments.com/cnc.php">http://www.serialcominstruments.com/cnc.php</a>
CAM	Estlcam	<a href="http://www.estlcam.com/">http://www.estlcam.com/</a>
CAM (Online)	Makercam	<a href="http://www.makercam.com/">http://www.makercam.com/</a>
Lasergraviererprojekt	3DPBurner	<a href="http://3dpburner.blogspot.de/">http://3dpburner.blogspot.de/</a>
Firmware	GRBL	<a href="https://github.com/grbl/grbl/wiki">https://github.com/grbl/grbl/wiki</a>
Hardware / Dokumentation	cncshield	<a href="http://blog.protoneer.co.nz/tag/grbl/">http://blog.protoneer.co.nz/tag/grbl/</a>
Hardware	Arduino Uno	ebay
Hardware	A4988 Treiber	ebay
Hardware	cncshield	ebay
Steuerungssoftware (RPI)	bcNC	<a href="https://github.com/vlachoudis/bcNC/wiki">https://github.com/vlachoudis/bcNC/wiki</a>
Firmwareübertrager	xloader	<a href="http://russemotto.com/xloader/">http://russemotto.com/xloader/</a>
Hardware	SilentStepStick	<a href="http://www.watterott.com/de/SilentStepStick">http://www.watterott.com/de/SilentStepStick</a>

Vielen Dank für Eure  
Aufmerksamkeit!  
Ich hoffe, Ihr hattet  
(auch) Spaß!

Fragen ?