


BL-Ctrl Anleitung

 Diese Seite als **PDF-Dokument**? Einfach auf das Symbol rechts klicken und etwas warten...

siehe auch: FlightCtrlAnleitung

1. Anleitung BL-Ctrl

1. BL-Ctrl

1. Features
2. Funktionsweise
3. Allgemeine Sicherheitshinweise

2. Inbetriebnahme

1. Schritt 1: Anschluss und Kontrolle der Betriebsspannung
2. Schritt 2: Einspielen des Programms
3. Schritt 3: Test der Hardware
4. Schritt 4: Anschluss eines Sollwertgebers

3. Einspielen der Software in den Regler

1. 1. Bootloader flashen
2. Firmware einspielen
2. Einspielen der Software mit PonyProg
3. Einspielen der Software mit Ponyprog und Serieller Konverter aus dem MK-Shop
4. Bedienung von Ponyprog

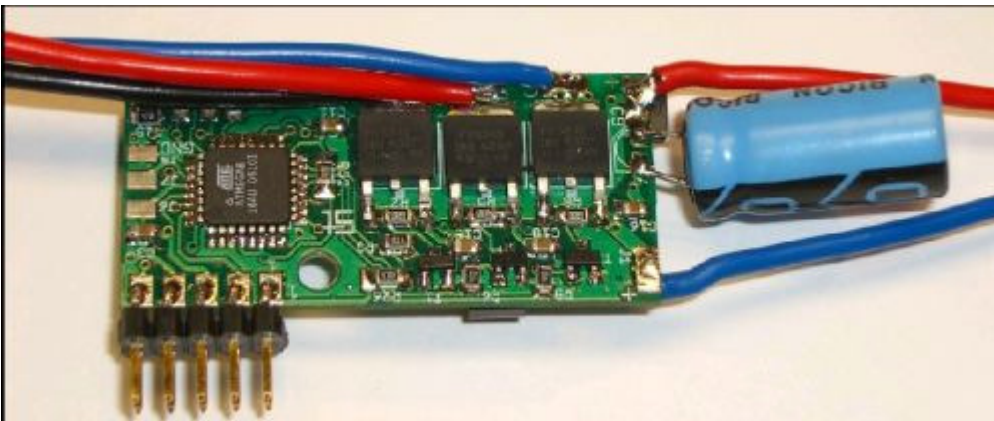
4. Stückliste & Bestückung

5. Alternative MosFets zur Leistungssteigerung

6. Anschluss

7. Schaltplan

8. WICHTIG



Selbstbauregler für Brushless-Motoren BL_CTRL_V1.0/V1.1 Schaltplan, Bestückung und Inbetriebnahme

 Zusatz-Info für den BL-Ctrl V1.1 gibt es hier: [BL-Ctrl_V1_1](#)


Anleitung BL-Ctrl

BL-Ctrl

Die Platine BL_Ctrl ist ein sensorloser Treiber für bürstenlose Gleichstrom-Motoren.

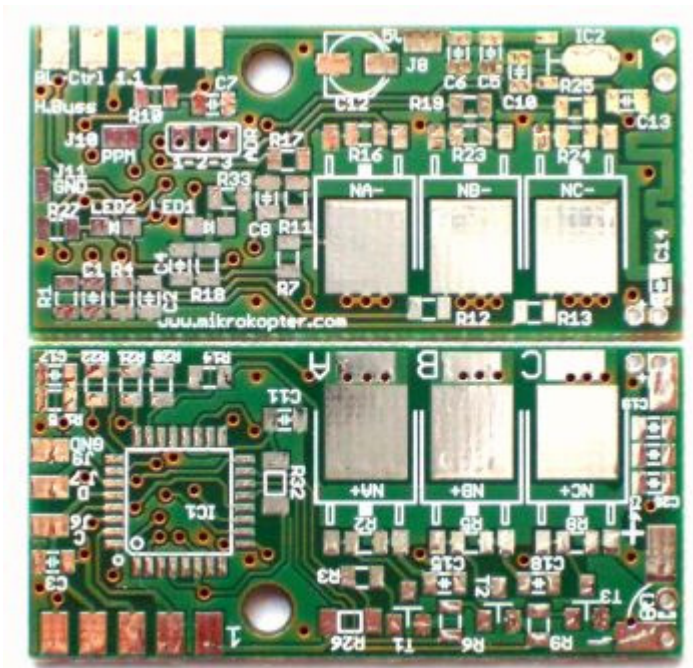
Sie wurde speziell für den Einsatz in Quadrokoptern entwickelt, wo schnelle Sollwertänderungen notwendig sind. Sie kann jedoch auch in anderen Applikation problemlos Verwendung finden.

Features

- Abmessungen 20 * 43mm
- Bestückt mit sechs 30A  MosFets
- Dauerbelastung ca. 5A (bei schlechter Kühlung) (mit alternativen MosFets höhere Leistung möglich)
- Controller: ATMEGA8 von Atmel
- Strommessung und -begrenzung auf der Gleichstromseite
- zwei LEDs (z.B. Okay und Error)
- Batteriespannungsmessung mit Unterspannungserkennung
- Die Software liegt komplett in C vor (Lizenzvereinbarung beachten!)
- diverse Schnittstellen zur Sollwerteingabe
- Ein Empfänger kann von den 5V versorgt werden (max. 50mA)

Als Schnittstellen sind möglich:

- Asynchrone Serielle Schnittstelle (zum Ansteuern oder Debuggen)
- I2C (zur schnellen Sollwertvorgabe)
- PPM-Signal (als Standard-Eingang vom Empfänger)



(Bild: BL-Ctrl V1.1)

Funktionsweise

Die Ansteuerung erfolgt dreiphasig in Impulsgruppen mit überlagerter PWM. Die PWM bestimmt die Höhe der Phasenspannung (genauer: der arithmetische Mittelwert der Spannung)

Die Phasenspannung am Motor (also die PWM) ist eine Stellgröße für die Drehzahl, weil ein Motor durch seine Drehung eine Gegenspannung (wie ein Generator) erzeugt, die der angelegten Spannung entgegenwirkt. Es stellt sich eine Drehzahl ein, bei der die Differenz von angelegter Spannung und Gegenspannung einen Strom fließen lässt, der dem abgenommen Drehmoment entspricht. Es sind immer 2 der sechs MOSFETs in Betrieb, um die Motorwicklungen zu bestromen. Der Zeitpunkt der Kommutierung (heißt: es muss auf die nächste Phase geschaltet werden), wird durch Spannungsmessung (genauer: Vergleich) an der nicht-bestromten Phase bestimmt. Dazu wird der im Atmega8 vorhandene Analogkomparator verwendet. Der Strom wird über den Spannungsabfall an einer definierten Leiterbahn bestimmt.

Das Verfahren ist für eine Überstromabschaltung genau genug.

Allgemeine Sicherheitshinweise

Wir garantieren nicht für fehlerfreies Verhalten des Reglers. Sobald der Regler mit Spannung versorgt ist, kann der am Regler angeschlossene Motor z.B. durch Fehler o.ä. jederzeit unerwartet anlaufen oder sich erhitzen. Trotz sorgfältiger Erstellung und Überprüfung übernehmen wir keinerlei Garantie oder Haftung (direkter oder indirekter Art) für die Fehlerfreiheit der Software, der Hardware oder von Informationen. Sie benutzen den Regler auf eigene Gefahr (dies gilt auch für dazugehörige PC-Programme). Weiterhin übernehmen wir keinerlei Haftung für Folgeschäden an Sachwerten oder Personen, die durch Anwendung des Reglers entstehen. Es liegt in ihrer Verantwortung, einen vollständigen Systemtest durchzuführen. Die Spannungsversorgung darf nicht verpolt werden.

Inbetriebnahme

Schritt 1: Anschluss und Kontrolle der Betriebsspannung

Um sicher zu gehen, dass man keine ungewollten Lötbrücken beim Bestücken in die Versorgung eingebaut hat, prüft man mit einem Durchgangsprüfer die 5V gegen GND (es gibt Löt pads, die mit "5V" und "GND" beschriftet sind).

Man sollte auch gleich die 12V-Versorgung auf Kurzschluss prüfen (direkt zw. den Anschlüssen für die Batteriespannung).

Bei diesen Durchgangsmessungen muss ein Widerstand deutlich über 0 Ohm gemessen werden bzw. der Durchgangsprüfer darf nicht 'piepen'.

Sollte sich doch ein Kurzschluss eingeschlichen haben, sind häufig Lötbrücken an den 100nF-Kondensatoren die Ursache. Bei Verwendung von Lötpaste gilt: weniger ist mehr.

Die Betriebsspannung wird an den Anschlüssen BAT+ und BAT- angelegt. Es wird dringend empfohlen, ein strombegrenztes Netzteil zu verwenden, bis man weiß, dass alles richtig funktioniert. Ein Steckernetzteil mit Gleichspannungsausgang von ca.9-12V tut hier gute Dienste. Nach Anschluss der Betriebsspannung wird die +5V-Versorgung überprüft. Dazu misst man am Anschluss „+5V“ gegen Masse. Die Spannung sollte zwischen 4,8 und 5,2 V liegen.

Schritt 2: Einspielen des Programms

Dies wird weiter unten genau beschrieben.

Schritt 3: Test der Hardware

Wenn der Controller erfolgreich programmiert werden kann, können die Motorleitungen angelötet werden. Die I²C Leitungen dürfen erst angelötet werden, nachdem das endgültige Programm eingespielt wurde, da diese Leitungen Teil der Programmierschnittstelle sind. Wenn mehrere BL-Regler oder die Flight-Ctrl über den I²C Bus verbunden sind, ist eine Programmierung nicht mehr möglich! In den Controller wird das Testprogramm „*Test_Sollwert_Fix_BrushLess-Ctrl_V0_xx.hex*“ eingespielt. Damit läuft der Regler mit einem fixen Sollwert sofort an, wenn er einwandfrei funktioniert. Nach dem Einschalten leuchten die LEDs kurz auf. Sollte der Motor nicht richtig anlaufen, kann das unterschiedliche Gründe haben:

1. Die MOSFET-Brücke (oder deren Ansteuerung) arbeitet nicht richtig.
2. Der Kommutierungszeitpunkt wird nicht richtig bestimmt. In dem Fall sollte man das File: „*Test_Manuell_BrushLess-Ctrl_V0_21.hex*“ einspielen. Damit wird der Motor ohne Kommutierungszeitpunkt-Bestimmung manuell angesteuert.

Läuft der Motor nicht oder unruhig, liegt vermutlich Fehler 1 vor. Wenn der Motor damit richtig anläuft, liegt vermutlich Fehler 2 vor. Die Drehrichtung des Motors dreht man übrigens, indem zwei beliebige Motorphasen getauscht werden. Dies wird auch bei Inbetriebnahme des MikroKopters notwendig, da alle Regler dasselbe Drehfeld aufbauen und somit bei gleichartigem Anschluss aller vier Motoren alle Propeller auch in dieselbe Richtung drehen. Im Zusammenhang mit je zwei Propellern für Recht-/Linkslauf ergibt sich dann folgendes Problem: Zweimal Auftrieb + zweimal Abtrieb = MikroKopter fliegt nicht, sondern fängt höchstens an stark zu schwingen.

In diesem Fall einfach die Drehrichtung beim Ein-/Ausschalten beobachten (Vorne/Hinten: rechtsherum, Links/Rechts: linksherum) und ggf. einfach zwei Anschlußkabel an den betreffenden Motoren vertauschen.

Schritt 4: Anschluss eines Sollwertgebers

Es kann nun z.B. das Programm: „*BrushLess-Ctrl_16kHz_V0_22.hex*“ eingespielt werden. Dies Programm nimmt den PPM-Anschluss des Empfängers als Sollwert. Wenn andere Schnittstellen verwendet werden (z.B. Seriell oder I2C-Bus), muss das natürlich auf den jeweiligen Sollwertgeber abgestimmt werden. Es gibt da keinen Standard.

Einspielen der Software in den Regler

Es gibt verschiedene Möglichkeiten/Programme, um die Software in den Regler zu flashen. Am einfachsten geht es per SerCon und MikroKopterTool. Sollte dies allerdings nicht klappen, so kann man alternativ z.B. auch PonyProg nutzen...


Der Bootloader muss nur **einmalig** per ISP in den AVR programmiert werden. Das MikroKopterTool setzt die Fusebits dabei automatisch korrekt.

Danach können die Updates der Firmware seriell einfach durchgeführt werden.

WICHTIG: Für das Flashen des Bootloaders ist eine echte serielle Schnittstelle erforderlich.

 Auf den SMD-vorbestückten Platinen ist bereits ein Bootloader installiert.

Wenn man keinen Bootloader verwenden will oder kann, so lässt sich die Firmware auch direkt per ISP flashen. (ist aber nicht empfohlen)

Den Bootloader und die BL-Ctrl Software gibt es hier:  <http://mikrocontroller.cco-ev.de/mikrosvn/BL-Ctrl/>

Bootloader flashen

- Die BL-Ctrl muss über das 10-polige kombinierte ISP-/seriell-Kabel (ISP1) an den SerCon angeschlossen sein.
- Die BL-Ctrl muss mit Spannung versorgt werden.
- Es dürfen keine Daten über den I2C-Bus kommen, d.h. entweder ablöten oder während des Programmierens Flight-Ctrl ausschalten/resetten.
- 'Controller flashen & Terminal' auswählen

```

MikroKopter-Tool V1.46 (c) 2007 - I. Busker
Datei: BOOTLOADER für BL-Ctrl
avrdude.exe -c ponyser -P COM1 -p m8 -U flash:w:BootLoader_MEGA8_8MHZ_V0.1.hex
-u -U lfuse:w:0x84:m -U hfuse:w:0xda:m

avrdude.exe: AVR device initialized and ready to accept instructions

Reading | ##### | 100% 0.01s

avrdude.exe: Device signature = 0x1e9307
avrdude.exe: NOTE: FLASH memory has been specified, an erase cycle will be performed
        To disable this feature, specify the -D option.
avrdude.exe: erasing chip
avrdude.exe: reading input file "BootLoader_MEGA8_8MHZ_V0.1.hex"
avrdude.exe: input file BootLoader_MEGA8_8MHZ_V0.1.hex auto detected as Intel Hex
avrdude.exe: writing flash (7978 bytes):

Writing | #####

```

- Jumper des SerCon setzen
- Dann auf 'FLASH' klicken und das Bootloader-Hexfile auswählen.
... jetzt wird der Bootloader programmiert – das dauert etwas ...
- nach erfolgreichem Programmieren meldet sich der Bootloader mit einer Kennung:
Vx.y:MKBL
Vx.y = Version
MKBL für MikroKopter-BootLoader

```

Reading | ##### | 100% 0.00s

avrdude.exe: verifying ...
avrdude.exe: 1 bytes of hfuse verified

avrdude.exe done. Thank you.

V0.1:MKBL

```

Firmware einspielen

- Die BL-Ctrl muss über das 10-polige kombinierte ISP-/seriell-Kabel (ISP1) an den SerCon angeschlossen sein.
- Jumper des SerCon entfernen.
- Die BL-Ctrl muss mit Spannung versorgt werden.

- Die I2C-Bus-Verbindung kann in Betrieb bleiben (noch testen).
- auf 'Update' klicken und aktuelles Hex-File der BL-Ctrl auswählen.



- jetzt wird programmiert - evtl. muss manuell resettet werden (BL-Ctrl einfach aus- und einschalten).
- wenn alles geklappt hat, meldet sich der BL-Regler mit seiner aktuellen Version.
- für BL-Ctrl V1.0: Jeder Motor-Regler benötigt eine eigene Adresse (unterschiedliche Hex-Files).
- für BL-Ctrl V1.1: Jeder Motor-Regler benötigt eine eigene Adresse, die per Lötjumper auf der Platine eingestellt wird (gleiche Hex-Files).
- Motoradressierung: 1=Vorn 2=Hinten 3=Rechts 4=Links

Einspielen der Software mit PonyProg

Das Einspielen der Software geht am einfachsten mit einem sog. ISP-Kabel, das den Controller entweder über die parallele Schnittstelle oder die serielle Schnittstelle des PCs programmiert. Das ISP-Kabel kann man schnell aus einigen Teilen zusammenbauen. Zum Programmieren muss der PC über eine „echte“ serielle oder parallele Schnittstelle verfügen. USB-auf-Seriell/Parallel-Wandler oder ähnliches gehen definitiv nicht! Als Software hat sich die Freeware „PonyProg“ bewährt. Ein Link zu Ponyprog2000 (www.lancos.com) und eine kurze Anleitung zur Bedienung sowie einen fertigen ISP-Adapter findet man auch unter <http://www.Mikrocontroller.com>.

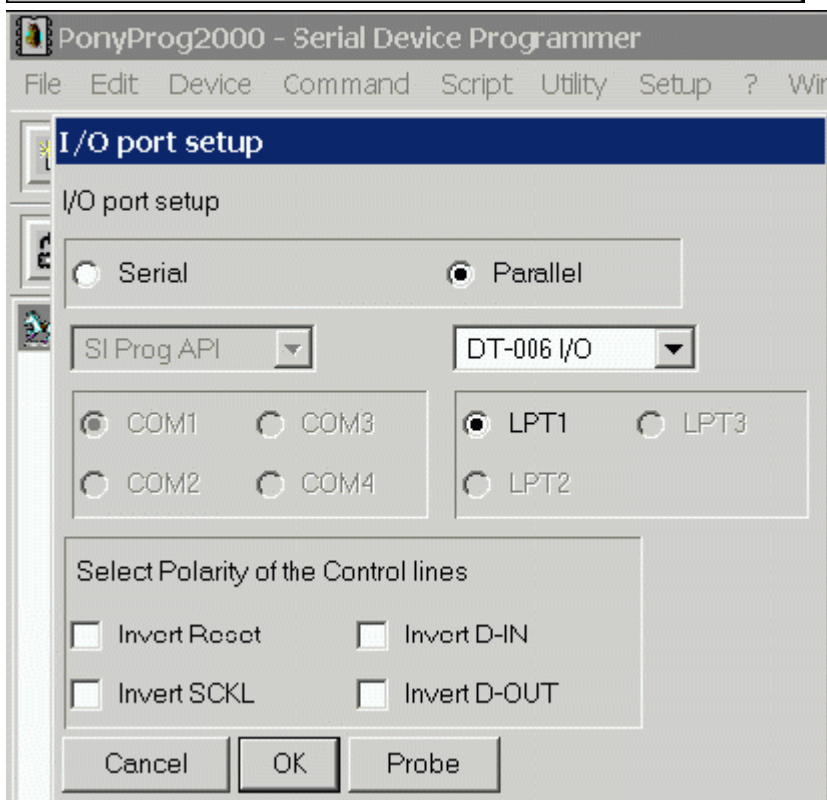
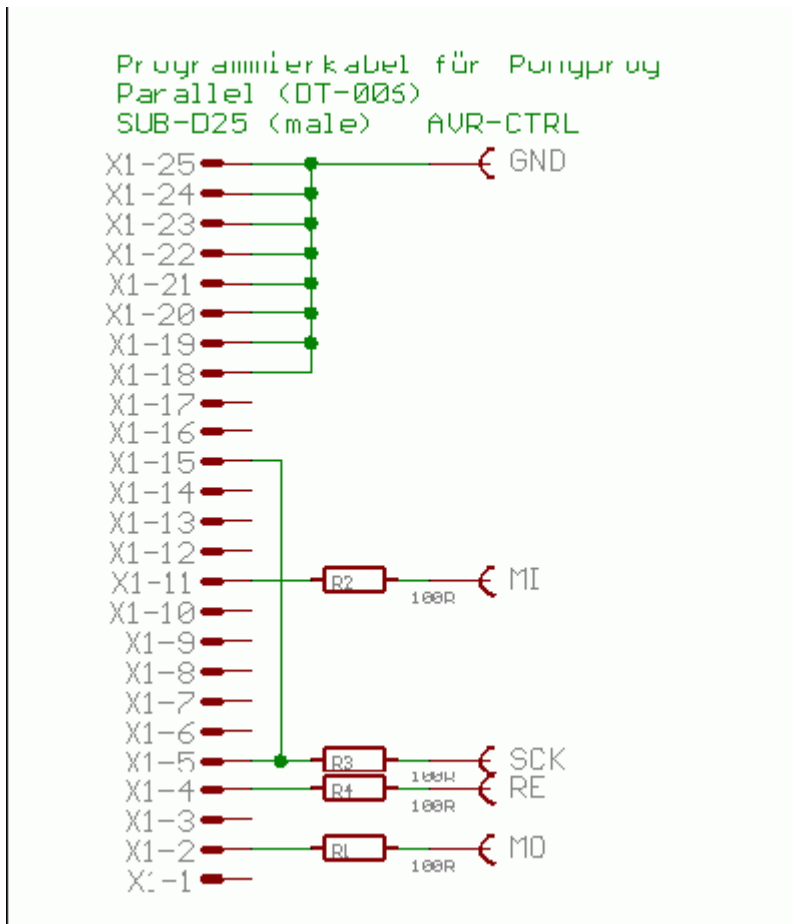


Bild: Schaltplan des parallelen ISP-Adapters und der entsprechenden Einstellung unter Ponyprog

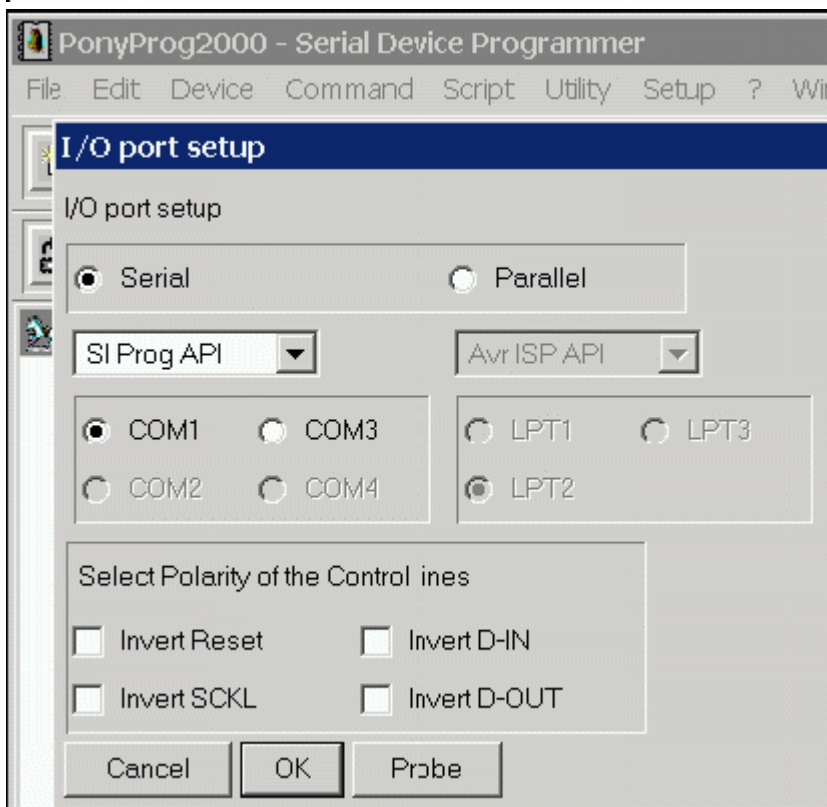
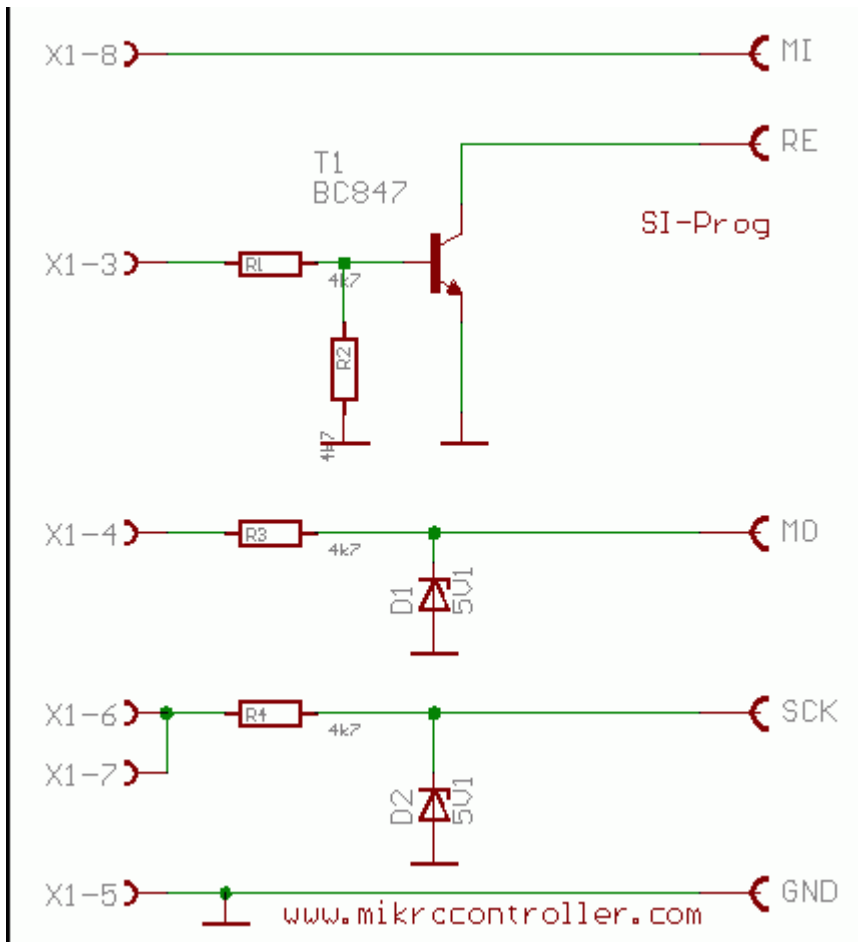
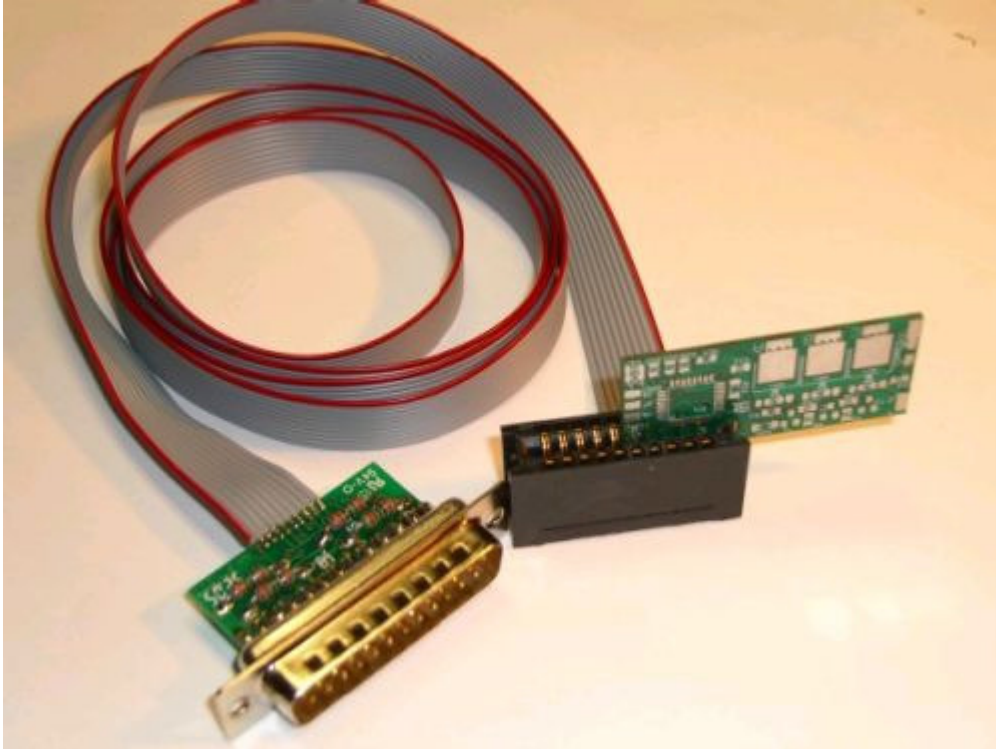


Bild: Schaltplan des seriellen ISP-Adapters und der entsprechenden Einstellung unter Ponyprog

Man kann sich einen einfachen ISP-Adapter aus wenigen Teilen bauen:

Platine ISP-Adapter (von <http://www.mikrocontroller.com>)
4 Widerstände (ca. 100-220 Ohm)
10-poliges Flachbandkabel
EC10 Steckkartenverbinder (Reichelt)

zur Not geht auch EC20 wie auf dem Photo:



Der Stecker wird an der Seite leicht eingesägt, damit man die Platine von der Seite einstecken kann. Danach wird Stecker mit etwas Kleber zusammengesetzt, weil der Bügel nicht mehr rasten kann. Pin1 ist sowohl auf der ISP_Platine, als auch auf der BL_Ctrl und dem Stecker markiert.

Einspielen der Software mit Ponyprog und Serieller Konverter aus dem MK-Shop

Da die obigen Anleitung den meisten Laien auf dem ersten Blick etwas undurchsichtig erscheint, hier eine simple Anleitung für den Serial Konverter (SerCon).

1. Den Serial Konverter vollständig bestellen (Reichelt), aufbauen und mit einer echten seriellen Schnittstelle am PC verbinden.
2. Das 10polige Flachbandkabel konfektionieren. Ordnungshalber liegt das rote Kabel über dem Pfeil auf dem Stecker.
3. Alle BI-Ctrl Platinen ebenfalls mit einer 2x5 poligen Stiftleiste ausrüsten. (siehe Foto oben, Stiftleisten sind nicht in der Reichelt-Bestellliste enthalten)
4. Konverter mit der BI-Ctrl Platine verbinden. (ISP1) Rotes Kabel jeweils auf Pin 1.
5. BI-Ctrl Platine, falls noch nicht geschehen, mit Spannung versorgen. Die grüne LED auf dem Konverter muss nun dauerhaft leuchten.
6. PonyProg entsprechend der unten aufgeführter Anleitung benutzen.

Bedienung von Ponyprog

Nach der Installation von PonyProg muss man es ggf. einmal kalibrieren. Das hat noch nichts mit einer angeschlossenen Hardware zu tun. (Wähle im Menü: Setup - Calibration)

Man sollte den Rechner zum Programmieren/Kalibrieren vorher einmal neu starten und während des Prozesses auch keine weitere Software laufen lassen, bzw. starten. Der Programmiervorgang ist 'zeitkritisch' und 'ungewollte' Unterbrechungen können einen AVR-Prozessor schonmal unbrauchbar machen.

Danach wird der Controller wie folgt programmiert:

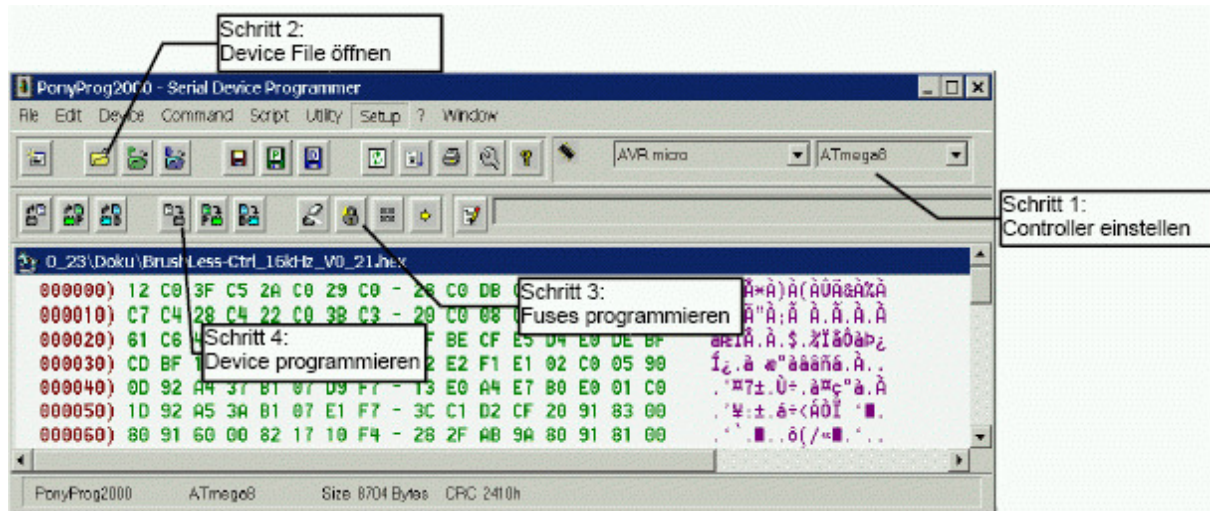


Bild: Schritte zum Programmieren des Controllers:

1. Controller: AVR micro --> ATMEGA8 einstellen
2. Das File: „*BrushLess-Ctrl_16kHz_ADR?_V0_yy.hex*“ (oder ähnlich) laden. Das ? steht hierbei für die Reglerposition. Das yy für die Versionsnummer. In jedem Regler muss also eine andere Firmware geladen werden damit jeder Regler korrekt auf seine Befehlsfolgen reagieren kann.
3. **ab V1.1 wird die Adresse per Lötjumper auf der Platine festgelegt, d.h. in jedem Regler befindet sich die gleiche Firmware.**
4. Fusebits (Konfigurationsbits) des Controllers einstellen (Siehe folgendes Bild) und mit „Write“ schreiben
5. Controller (Device) programmieren

Hinweis

Die Einstellungen hier müssen sorgfältig verglichen werden! Der Regler muss zum Programmieren mit Spannung versorgt sein. Die Fusebits müssen nur einmal programmiert werden. Wenn ein neues Programm eingespielt wird, müssen sie nicht jedes mal neu gesetzt werden.

Configuration and Security bits

☐ 7 ☐ 6 ☐ BootLock12 ☐ BootLock11 ☐ BootLock02 ☐ BootLock01 ☐ Lock2 ☐ Lock1

☐ RSTDISBL ☐ WDTON ☒ SPIEN ☐ CKOPT ☐ EESAVE ☐ BOOTSZ1 ☐ BOOTSZ0 ☐ BOOTRST

☐ BODLEVEL ☒ BODEN ☒ SUT1 ☒ SUT0 ☒ CKSEL3 ☐ CKSEL2 ☒ CKSEL1 ☒ CKSEL0

☒ Checked items means programmed (bit = 0) ☐ UnChecked items means unprogrammed (bit = 1)

Refer to device datasheet, please

Cancel

OK

Clear All

Set All


Write

Read

Bild: Ponyprogeinstellung der Fusebits bei Verwendung des internen 8MHz-Oszillators und der Brown-Out-Detection (Unterspannungs-Watchdog).

Schritt 1: Controller einstellen Schritt 2: Device File öffnen Schritt Schritt 3: Fuses programmieren 4: Device programmieren

Stückliste & Bestückung

 **Tip:** Es empfiehlt sich, beide Seiten der unbestückten Platine zunächst einzuscannen oder zu fotografieren. Das macht es später einfacher, ungewollte Lötbrücken und ähnliche Fehler aufzuspüren.

Die Bauteile sollten in der Reihenfolge eingebaut werden, wie sie in dieser Liste stehen, weil man es dann mit der Bestückung einfacher hat.

Kursiv dargestellte Teile befinden sich auf der Unterseite

Anz	Reichelt-Best.Nr.	Hinweis	Teil	Name
1	ATMEGA8-16TQ	Markierung beachten, Flussmittel verwenden!	Controller	<i>IC1</i>
3	BC817-16SMD		Transistor	<i>T1, T2, T3</i>
1	4k7 SMD 1/4W		Widerstand	<i>R26</i>
1	10R SMD 1/4W		Widerstand	<i>R32</i>
2	4k7 SMD-0805		Widerstand	<i>R6, R9</i>
6	X7R-G0805 100nF		Kondensator	<i>C3, C15, C16, C17, C18, C11</i>
5	4k7 SMD-0805		Widerstand	<i>R3, R20, R21, R22, R15</i>
3	470R SMD-0805 oder 680R SMD-0805		Widerstand	<i>R2, R5, R8</i>

1	18k SMD-0805		Widerstand	<i>R14</i>
1	SMD-LED 0805 GR	Richtung beachten	CHIPLED	LED1
1	SMD-LED 0805 RT	Richtung beachten	CHIPLED	LED2
9	X7R-G0805 100nF		Kondensator	C1, C2, C4, C5, C6, C7, C8, C10, C13 (Version 1.0)
12	X7R-G0805 100nF		Kondensator	C1, C2, C4, C5, C6, C7, C8, C10, C13, C14, C19, C20 (Version 1.1)
7	4k7 SMD-0805		Widerstand	R1, R4, R7, R11, R12, R13, R18
2	1k SMD-0805		Widerstand	R27, R33
4	18k SMD-0805		Widerstand	R10, R16, R23, R24
3	100R SMD-0805		Widerstand	R17, R19, R25
3	IRFR1205	alternativ auch IPD03N03LB, IRLR7843	MOSFET	NA-, NB-, NC-
3	IRFR5305	alternativ auch FDD6637	MOSFET	NA+, NB+, NC+
2	SMD Elko 10/16	Markierung beachten	Elko	C12, C14 (Version 1.0)
1	SMD Elko 10/16	Markierung beachten	Elko	C12 (Version 1.1)
1	RAD330/16	Polung beachten und von der Seite anlöten	Elko	C9
1	TA78L05 oder μ A78L05		Festspannungsregler	IC2

Sonstiges

- 1 Stecker EC10 - Programmierstecker
- 1 D-SUB ST 25 - Sub-D Stecker für Parallelport-Programmierkabel
- 0,5m 10poliges Flachbandkabel
- 1m Anschlussleitungen

Hilfsmittel

- CR44 Zinnhaltiges SMD-Flussmittel

- Entlötlitze AA Entlötlitze 1,5mm
- Lötzinn AG 0,507 Lötzinn 0,5mm
- Bedrahteter Spannungsregler, falls in SMD nicht verfügbar



Alternative MosFets zur Leistungssteigerung

Mit besseren MosFets sind auch dauerhaft 10A und kurzzeitig 20A möglich. Der Änderung erfolgt entsprechend:

IRFR1205 (NA-, NB-, NC-) wird ausgewechselt mit: **IRLR7843** oder **IPD03N03LB** (Infineon)

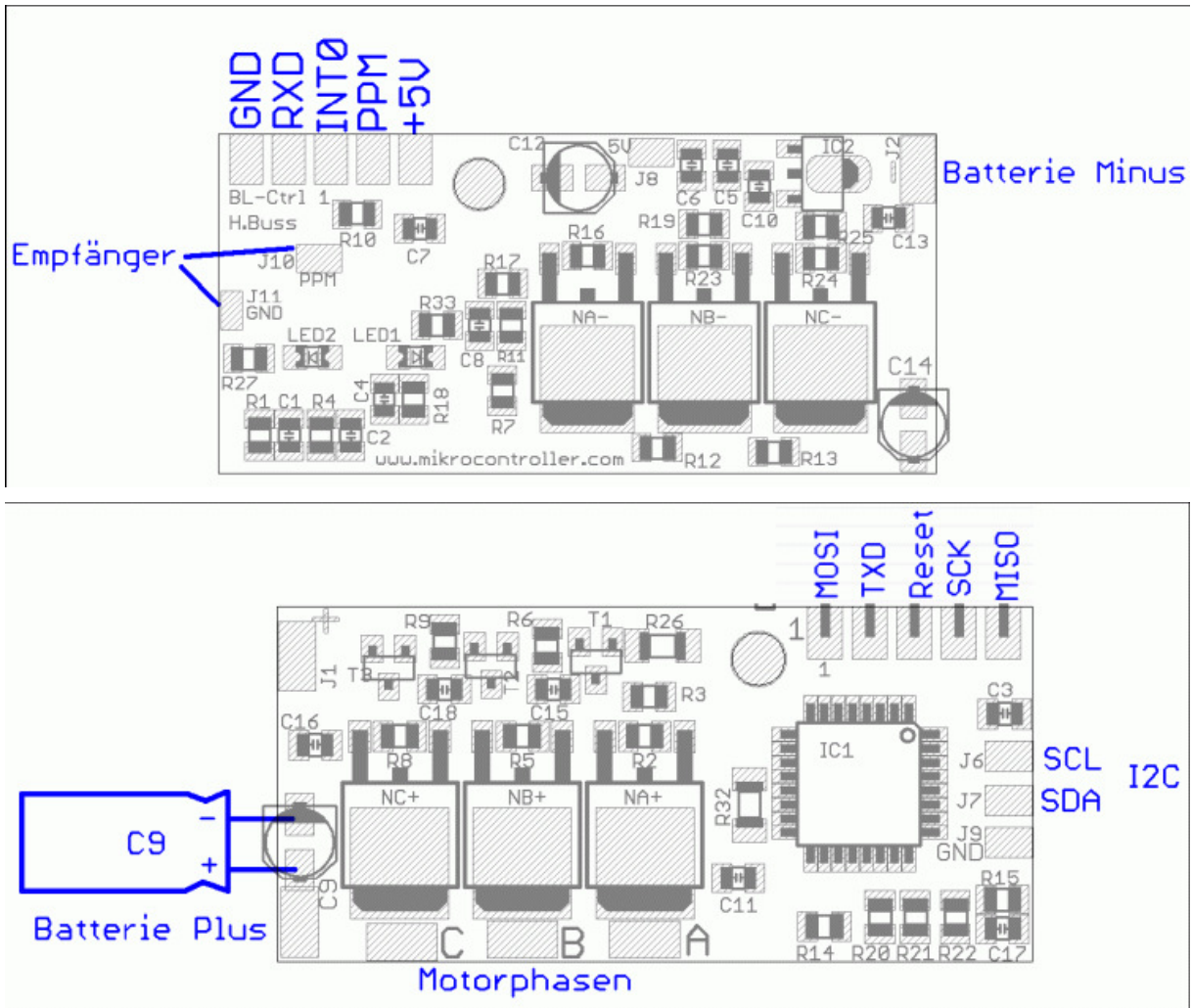
IRFR5305 (NA+, NB+, NC+) wird ausgewechselt mit: **FDD6637** oder **SPD50P03L** (Infineon)

Die beiden Infineon MosFets haben zur Zeit die besten Werte, bei dem **SPD50P03L** müssen allerdings die Beinchen etwas gebogen werden.

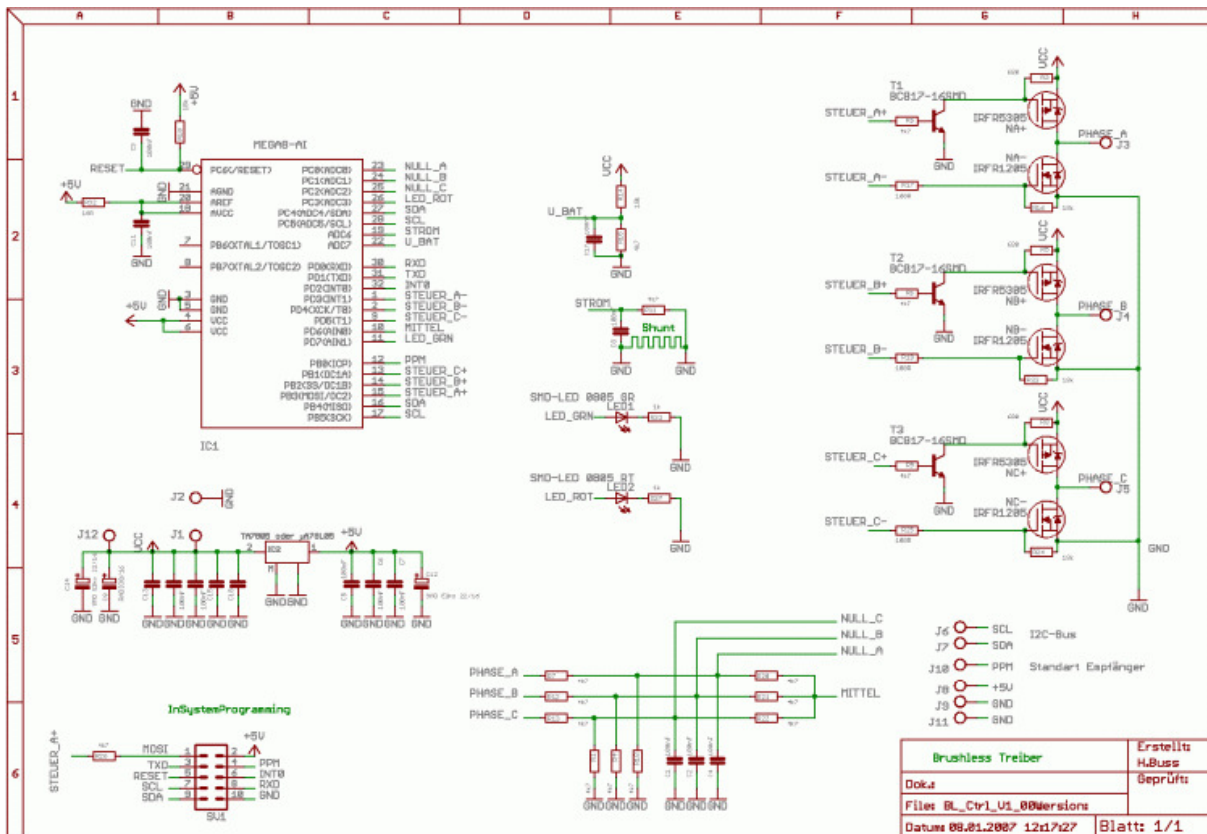
Des weiteren müssen die Regler mit einer neuen Firmware bespielt werden. Diese findest du im Download Bereich bzw. hier als [🌐 Direkt-Link](#). Weitere Informationen findest du im Forum (Suchfunktion!).

Die Mosfets (außer Infineon) können im MK-Shop [🌐](#) bezogen werden und befinden sich auch auf den vorbestückten Reglerplatinen aus dem Shop.

Anschluss



Schaltplan



(Klicken für hohe Auflösung)

WICHTIG

⚠ Die Regler sind vor Feuchtigkeit zu schützen. (Zerstörung der Endstufe !)

Daher **unbedingt** einen Schrumpfschlauch verwenden, um die Regler zu schützen !

KategorieAnleitung

MikroKopter: BL-Ctrl Anleitung (zuletzt geändert am 27.11.2007 durch MoinAdmin1)