

# Anbringen einer Programmierschnittstelle an den TowerPro 25A Brushless Regler

Anbringen einer Programmierschnittstelle an den TowerPro 25A Brushless Regler .....	1
0. Vorwort .....	1
1. Einleitung .....	2
2. TowerPro 17A und 25A Reglertypen, Umbauanleitung und Firmware Dateien .....	5
3. Werkzeuge und Material für den Umbau .....	5
4. Tipps zum Löten von SMD Bauteilen und Prozessor Pins .....	6
5. Der eigentliche Umbau des TowerPro 25A Typ 2 Reglers .....	7
6. Einspielen der neuen Firmware .....	12
7. Testen des umgebauten Reglers .....	13
8. Abschließen des Umbaus .....	13
9. Abschließende Worte .....	15
PS .....	15

## 0. Vorwort

Dies ist eine Übersetzung der Umbauanleitung von Arthur P. von [www.mikrokopter.de](http://www.mikrokopter.de)

Original: MikroKopter: en/TowerPro25A2Regler (zuletzt geändert am 07.11.2007 02:04 durch Arthur P)

Meine Anmerkungen und Ergänzungen sind mit **HJS** gekennzeichnet. Ich hoffe, dass dies dem Einen oder Anderen beim Umbau der preiswerten Chinaregler hilft.



Hier wird nur der Teil des Dokuments verwendet, der sich mit dem Anbringen der Programmierschnittstelle befasst. Der Umbau auf I2C-Ansteuerung wurde weggelassen.

**Alle Veränderungen unterliegen der eigenen Verantwortung. Es sollte jedem klar sein, dass die Garantie damit hinfällig wird. Ich übernehme keine Gewährleistung für die Richtigkeit der hier gemachten Aussagen oder für Fehler in der Software.**

Mein Dank geht an Arthur P für die ausführliche Umbauanleitung die sich sicher auch auf andere Regler übertragen lässt. Und natürlich an Bernhard Konze (Quax) der die Software entwickelt und die nötigen Hardwareänderungen herausgefunden hat.

Hans-Jürgen Schmidt (Spacejet\_Jürgen) im Mai 2008

## 1. Einleitung

Multirotor Plattformen wie der Mikrokopter (<http://www.mikrokopter.de/>), Big Quadro (<http://www.ttronix.de/>), oder UAVP (<http://uavp.org>) benötigen eine hohe Update Rate auf die Gaseinstellung der Motor-Regler. Die meisten angebotenen Brushless Regler sind nicht in der Lage im PPM Betrieb Update Raten von mehr als etwa 150/sec zu verarbeiten. MultiKopter sind aber wesentlich stabiler zu fliegen wenn Update Raten von etwa 300/sec sicher übertragen werden. Dies führte zum Einen zur Entwicklung von Brushless Reglern mit I<sup>2</sup>S (auch TWI oder I2C Interface genannt) und zum Anderen zur Entwicklung alternativer Software für die erhältlichen Regler.

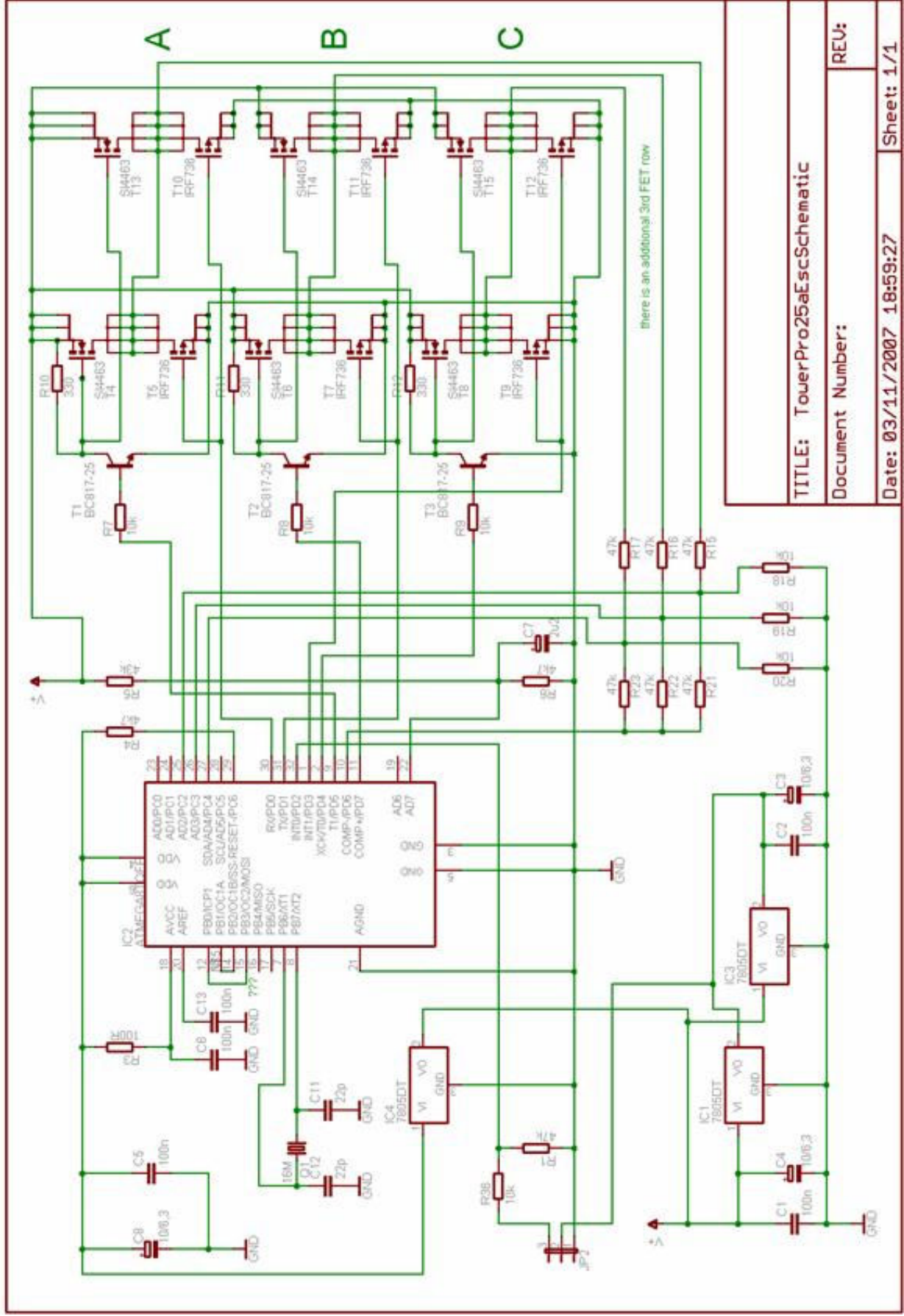
Die bekanntesten Regler mit I2C-Interface sind die Holger Regler (<http://www.mikrokopter.de/>). Diese gibt es in zwei Versionen mit 5A Dauer- und 10A Spitzenstrom, sowie der stärkeren Version mit 10A Dauer und 20A Spitzenstrom. Sie sind recht preiswert aber viele Benutzer finden den Zusammenbau und den Umgang mit den SMD Bauteilen sehr schwierig.

Bis jetzt bietet nur YGE (<http://www.yge.de/>) fertige Regler mit TWI/I2C Interface mit Leistungen von bis zu 30A an (YGE30i). Diese können in einem UAVP System verwendet werden, aber sie benutzen ein anderes Software Interface als der Mikrokopter (YGE verarbeitet Gas-Werte von 0..120, MK sendet Gas-Werte von 0..255). Das macht Änderungen an der MK Software notwendig, wenn diese Regler benutzt werden sollen.

Andere Lösungswege I2C fähige und bezahlbare Regler zu entwickeln wurden von Quax (Bernhard Konze) und einigen anderen beschritten. Sie haben preiswerte Kaufregler so modifiziert, dass sie mit I2C Interface arbeiten oder auch bei PPM-Ansteuerung die hohen Updateraten von 500/sec verarbeiten können. Einer der beliebtesten Umbauten ist der Arkai oder TowerPro 17A Regler. TowerPro bietet außerdem einen 25A und einen 30A Typ an, die beide dem 17A Typ sehr ähnlich ist.

Um bei PPM-Ansteuerung die hohen Updateraten von 500/sec verarbeiten zu können muss der Prozessor des Reglers eine neue Software erhalten. Dafür müssen die zur Programmierung notwendigen Leitungen an die entsprechenden Prozessor Pins und einen für das Programmierinterface geeigneten Stecker gelötet werden.

Hier ist das Schaltbild des 25A Typ 2 (angepasst aus dem Schaltbild des 17A Typs. Es mag einige kleinere Differenzen zu den aktuellen Typen geben wie die Anzahl der Spannungsregler).



TITLE: TowerPro25aEscSchematic

Document Number:

REV:

Date: 03/11/2007 18:59:27

Sheet: 1/1



Hier ist eine bebilderte Anleitung zum Einbau eines Programmierinterface für den TowerPro 25A ESC Typ 2 (mit separater Platine für die Spannungsregler)



Äußerlich ist kaum ein Unterschied zwischen den TowerPro 25A Typ 1 oder Typ 2 Reglern feststellbar. Beachte die Beulen an diesem 25A Typ 2. Beim Typ 1 waren die Beulen nur auf der Seite mit dem Kondensator zu sehen.



## 2. TowerPro 17A und 25A Reglertypen, Umbauanleitung und Firmware Dateien

Bitte beachte, dass sich diese Umbauanleitung auf den neueren Typ 2 Regler (erhältlich seit September 2007) bezieht. Die ältere Typ 1 Version hat ein etwas anderes Layout, ist aber sehr ähnlich. Äußerlich in ihrem gelben Schrumpfschlauch sind sie gleich, es sei denn man erkennt sie an dem unterschiedlichen Beulenmuster der Spannungsregler. *(HJS: Auch der TowerPro 30A hat das gleiche Layout und kann nach dieser Anleitung umgebaut werden.)*

Eine Umbauanleitung für den TowerPro 17A Regler, der ein identisches Layout hat und für den der gleiche Umbau gemacht werden muss gibt es hier: [http://home.versanet.de/~b-konze/low\\_cost/18a\\_regler.htm](http://home.versanet.de/~b-konze/low_cost/18a_regler.htm). Dies ist auch die Stelle wo man die Firmware Dateien zu Umprogrammierung der Regler findet. Die Software kann für beide Typen des 17A und 25A Reglers benutzt werden.

*(HJS: Die Software für PPM-Ansteuerung auf der Seite von Quax kann max. 150Hz Updaterate. Alternative siehe tt-tronix-Forum)*

**Achtung!!! Auf der Seite befinden sich noch andere ZIP Files für andere Reglertypen. Du brauchst die richtige Datei. Die Benutzung der falschen Software kann den Regler oder den Motor beschädigen!!!**

## 3. Werkzeuge und Material für den Umbau

Zur Durchführung des Umbaus dieser Regler brauchst Du folgendes:

- TowerPro 25A Typ 2 ESC (verschiedene Bezeichnungen und Bezugsquellen, mein Favorit: <http://www.unitedhobbies.com/> in Hong Kong).
- Kleine scharfe Schere.
- Dünner isolierter Draht. Fädeldraht ist gut geeignet weil er sich gut löten lässt. Draht mit Plastikisolierung sollte nicht dicker als ein Prozessor Pin sein. *(HJS: geeigneten Kupferlackdraht gibt's bei Reichelt unter Best.-Nr.: CUL 100/0,15 )*
- LötKolben mit feiner Spitze. Es muss nichts besonderes sein, Ich benutze einen einfachen Antex M12 12Watt LötKolben.
- Einen nassen Schwamm im LötKolbenständer.
- • Dünnes Elektronik Lötzinn *(HJS: am Besten 0,5mm nicht bleifrei - das lötet sich besser ☺).*
- Entlötlitze (falls man mehrere Prozessor Pins zusammenlötet).
- Spitze Pinzette (bevorzugt der selbstklemmende Typ).
- Doppelseitiges Klebeband.
- Starke Lupe. Wenn Du eine bekommen und damit arbeiten kannst solltest Du über eine Stereo-Lupe nachdenken.
- Gutes Licht (Eine dieser Hobbylampen mit eingebauter Lupe ist ideal).
- Sauberer Arbeitsplatz.
- Einen Ausdruck der wichtigsten Punkte der Anleitung. Es ist immer einfacher kritische Punkte während der Arbeit auf Papier nachzulesen (oder gleich die wichtigsten Bilder als Unterlage zu benutzen damit man sofort sieht was man tut.).
- Einfaches kleines Volt- Multimeter. Meines ist sehr klein und kostet weniger als 15€. Du brauchst sehr dünne Messspitzen um an den feinen Prozessorbeinchen messen zu können.
- SerCon Serial Konverter (<http://www.mikrokoetter.de/>) oder ein anderes ISP Programmier Interface und einen Computer mit dem der Regler neu programmiert werden kann.

- Software für die Programmierung des ATmega8 Chips, z.B. das Freeware Programm PonyProg (<http://www.lancos.com/prog.html>).
- ZIP File mit der richtigen Firmware und die Einstellungen der ATmega8 Fuse Bits.
- **Geduld !!!** Nicht hetzen. Der Umbau eines Brushless Reglers ist Mikrochirurgie. Zu große Eile und der „Patient“ stirbt. Fehler können Dich mehr kosten als einen Regler. Ich habe mir durch Fehler beim Umbau einen oder zwei Motoren verschmort.

#### 4. Tipps zum Löten von SMD Bauteilen und Prozessor Pins

Die beste Methode um Drähtchen an die Prozessor Pins zu löten, die ich gefunden habe, ist das vorsichtige Verzinnen von Prozessor Pin und dem blanken Ende des Drahtes.

Zunächst reinige die LötKolbenspitze auf dem nassen Schwamm. Dann tippe kurz an das Lötzinn, sodass ein **ganz kleiner** Tropfen Lötzinn an der Spitze bleibt (sollte kaum zu erkennen sein, sonst ist es zuviel). Vorsichtig den richtigen Prozessor Pin mit der LötKolbenspitze berühren und am Beinchen hin und her bewegen um es zu verzinnen. Prüfe alles mit der Lupe sorgfältig auf ungewollte Lötbrücken zu benachbarten Prozessor Pins. Falls nötig greife zur Entlötlitze, sauge alles wieder auf und beginne von vorn.

Jetzt entferne etwa 1-2 mm der Isolierung von beiden Drahtenden und verzinne sie.

*(HJS: Bei dünnem Kupferlackdraht klappt das gut, wenn man das Drahtende mit etwas Flußmittel – Kolophonium- benetzt, etwas Lötzinn an der Lötspitze aufnimmt und dann das Drahtende in das flüssige Lötzinn taucht.)*

Kürze das blanke Ende auf 0,5 bis 1mm und biege es so, dass Du es bequem gegen den Prozessor Pin drücken kannst. Dabei sollte das blanke Drahtende flach am Prozessor Pin anliegen. Falls nötig benutze eine Pinzette um den Draht zu halten. Bei Kunststoffisolierung darf man dann den Draht nicht zu nahe am Ende fassen weil durch das Löten die Isolierung weich wird und man mit der Pinzette ein Loch hineindrücken könnte. (Halte etwa 1cm Abstand.)

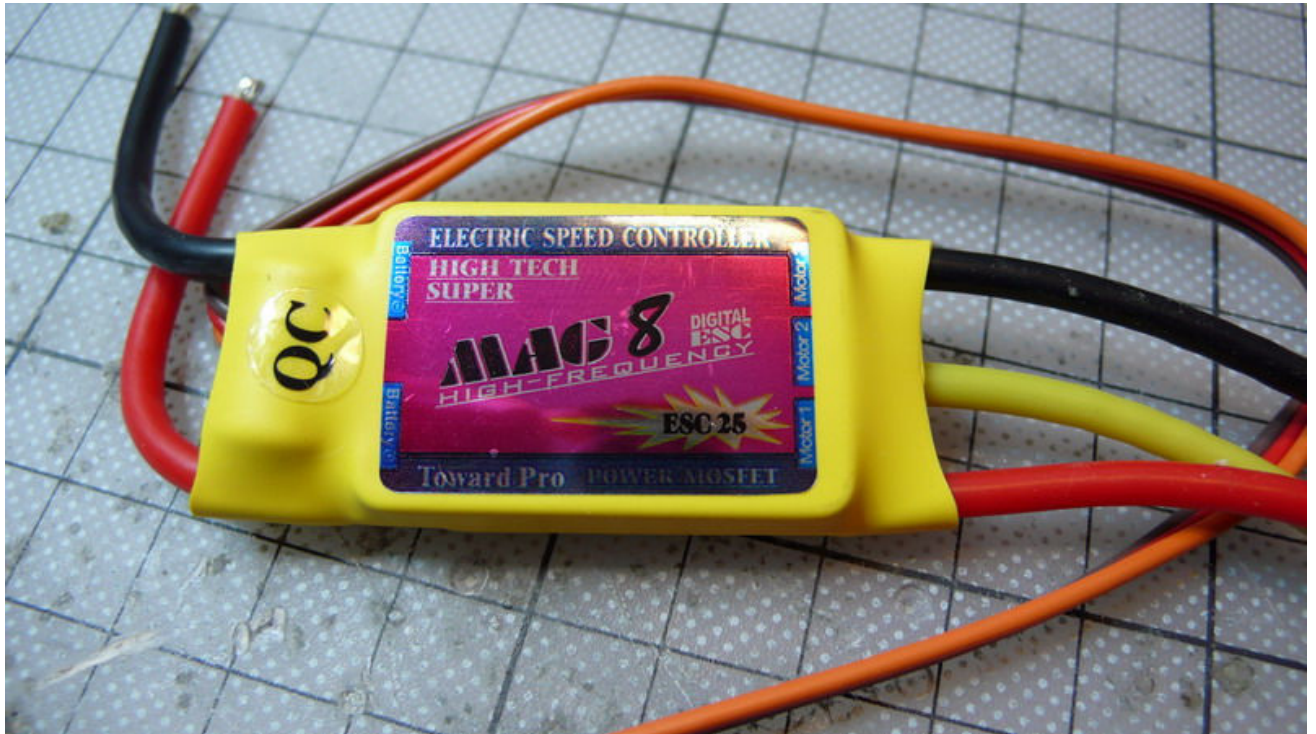
Während Du den Draht gegen das Prozessorbeinchen drückst, berühre ihn kurz mit der LötKolbenspitze. Das sollte das Lot am Draht und Prozessor Pin schmelzen und für eine gute Verbindung sorgen. Warte einen Moment bis das Lot sich abgekühlt hat.

Kontrolliere die Verbindung unter der Lupe. Achte darauf, dass die Lötverbindung über eine gewisse Länge verfügt, damit sie später im Betrieb nicht abreißt und zu schweren Fehlern führt.



## 5. Der eigentliche Umbau des TowerPro 25A Typ 2 Reglers

Zuerst wird der Schrumpfschlauch entfernt. Am Besten man schneidet entlang der Seite oder über die Metallplatte auf der Seite mit dem Aufkleber. So vermeidet man irgendwelche Teile zu beschädigen.

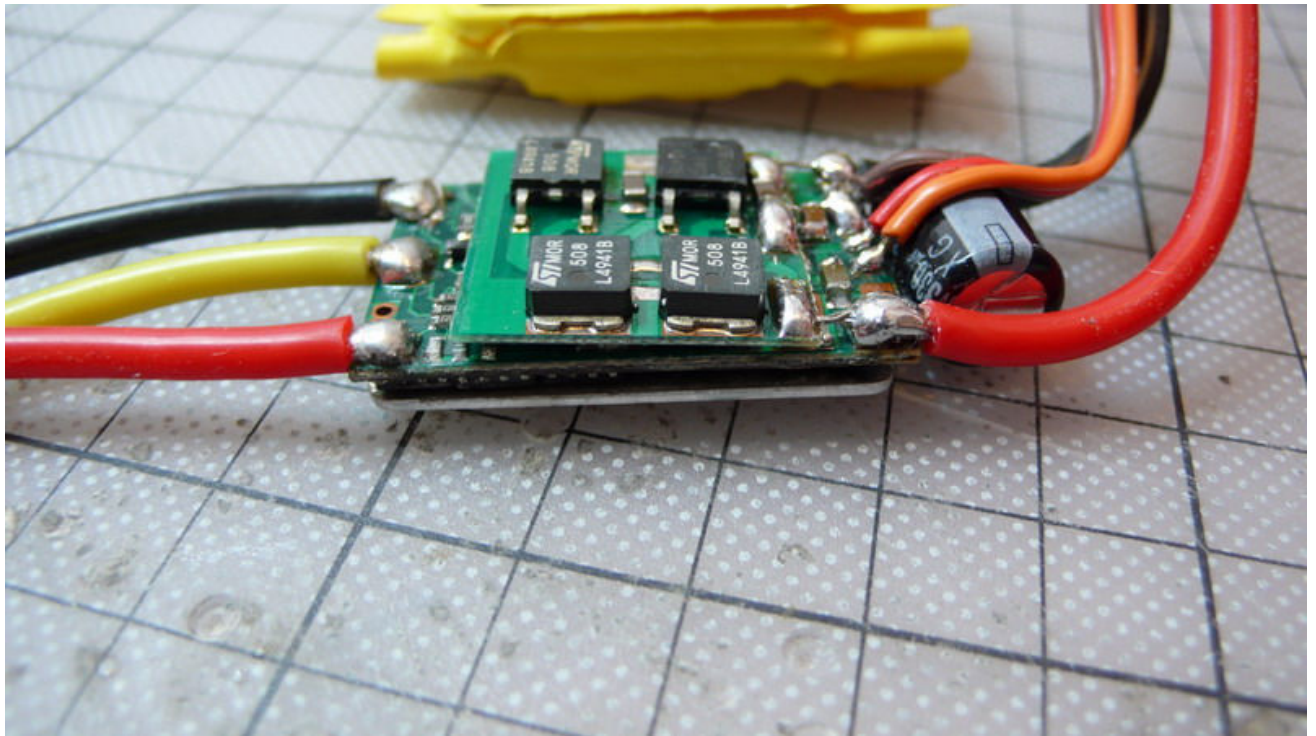


Achte darauf, dass sich - für unseren Zweck - der Aufkleber auf der Rückseite des Reglers, der Seite mit der Metallplatte, befindet.

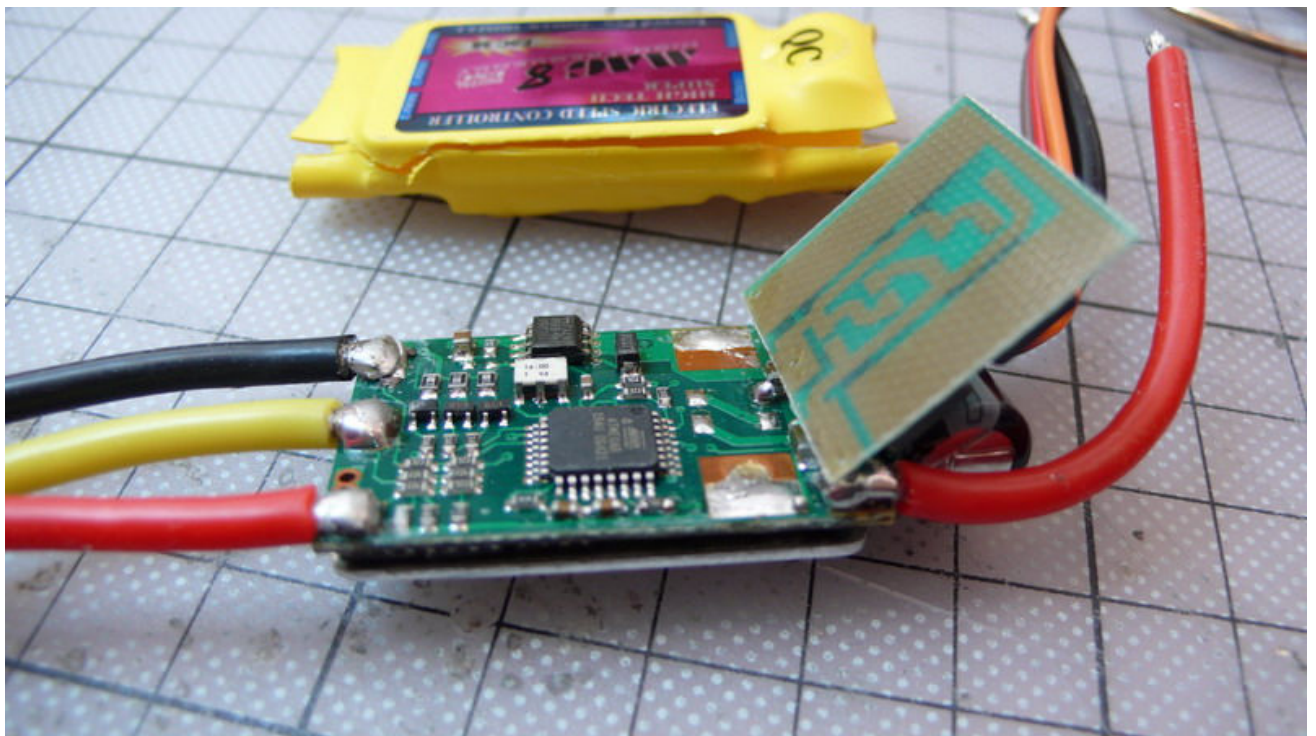


Nach dem Entfernen des Schrumpfschlauchs ist die Oberseite des Typ 2 Reglers klar zu erkennen. Dieser Typ hat eine zweite Leiterplatte mit 4 Spannungsreglern, die mit 3 blanken Drähten mit der Hauptplatine verbunden ist.





Jetzt wird die Spannungsreglerplatine vorsichtig zurück gebogen. Zuviel hin und her biegen sollte vermieden werden, da sonst die Verbindungen zur Hauptplatine brechen können. Versuche die Platine während des Umbaus in der gleichen Position zu belassen.

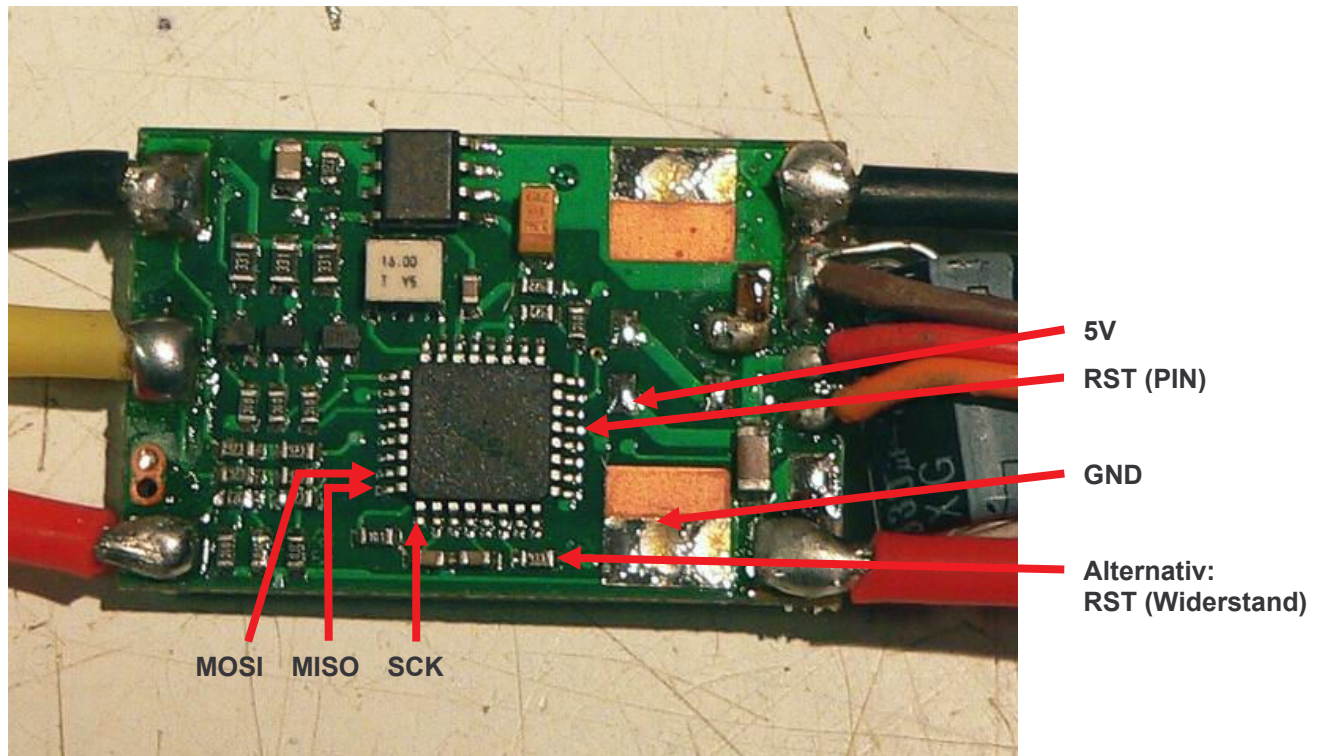


Wenn Du sicher bist, dass du das 5V BEC nicht benötigst, kannst Du auch die Zusatzplatine entfernen. Persönlich finde ich das Verschwendung. Es stört nicht sie am Platz zu lassen und man hat später die Möglichkeit das BEC-System doch zu benutzen. Dann muss man aber darauf achten, wenn das System bereits eine 5V Versorgung hat, eine geeignete Schottky Diode zwischen das BEC und die 5V-Versorgung zu schalten.

*(HJS: Ich habe die Zusatzplatine entfernt und einen der Spannungsregler auf den oberen leeren Platz auf der Hauptplatine gelötet. Dadurch wird der Prozessor versorgt ohne zusätzliche 5V vom Hauptsystem zu benötigen und man hat gut Platz zum Löten. )*



Für die Umprogrammierung des Mikrocontrollers müssen wir das ISP-Interface (In System Programmer) - bestehend aus den MISO, MOSI, SCK, RST, 5V und GND Leitungen - herausführen. Die RST Leitung kann auch am Ende eines Widerstands anstatt direkt am Prozessor Pin abgegriffen werden. ( Bei den verschiedenen Layouts kann der aber an unterschiedlichen Stellen sein. Da wir hier das I2C-Interface nicht brauchen, kann die Leitung auch direkt an den Prozessor Pin gelötet werden.)



Löte die MOSI, MISO und SCK Leitungen an die entsprechenden Pins. Das sind recht einfache Verbindungen. Achte darauf, beim Löten nicht die Isolierung von schon verlegten Leitungen zu berühren. Das kann zu Beschädigungen und später zu Kurzschlüssen führen.

Prüfe diese auch Verbindungen sorgfältig mit der Lupe auf ungewollte Lötbrücken zu benachbarten Prozessor Pins. Falls irgendwelche Zweifel bestehen greife zur Entlötlitze, sauge alles wieder auf und beginne von vorn. Außerdem achte darauf, dass die Lötverbindung über eine gewisse Länge verfügt, damit sie später im Betrieb nicht abreißt.

Jetzt müssen noch die Drähte an die Pads für 5V und GND und an den Widerstand, an dem wir das RST Signal abgreifen wollen (oder ebenfalls direkt an den Prozessor Pin), gelötet werden. Ganz einfach.

Jetzt gibt es 3 Möglichkeiten:

1. Verbinde die für die Neuprogrammierung benötigten Leitungen mit dem Programmieradapter, spiele das neue Programm auf und entferne die Leitungen wieder. Das bedeutet weniger "Spaghetti" auf dem Regler, aber bedeutet auch, dass bei einem Firmware Update ein Teil der Mikrochirurgie noch mal gemacht werden muss.
2. Verbinde die für die Neuprogrammierung benötigten Leitungen mit einem Stück Lochstreifenplatine. Die Programmierleitungen werden immer nur dann angelötet, wenn man ein Update einspielen will. Dies ist meine bevorzugte Lösung. Wenig zusätzliches Gewicht, reduziert die Zugbelastung an den Mikrodrähtchen und man muss bei einem Update die mikrochirurgischen Arbeiten nicht wiederholen.

3. Genau wie 2) aber mit einem dauerhaft angeschlossenen ISP-Steckverbinder. Dies lohnt sich wenn man selbst an der Firmware entwickelt, ansonsten hat man einen etwas klobigen Regler.

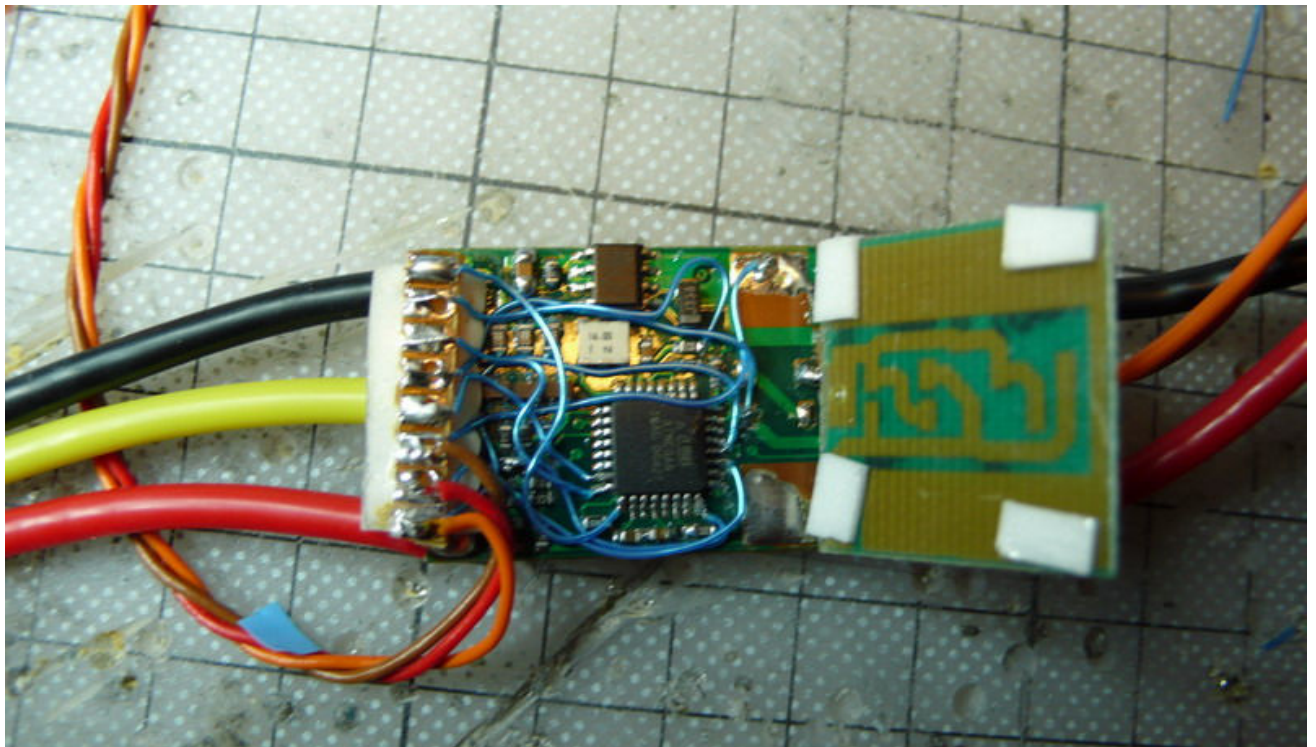
Ich benutze doppelseitiges Schaumstoffklebeband um mein kleines Stück Lochstreifenplatine mit 8 Pads zu befestigen. Da sie über den Motorleitungen platziert ist, muss man erst 2 Lagen zwischen die Leitungen kleben und dann ein langes Stück unter die Lochstreifenplatine. Ich habe das Stück etwas breiter gemacht damit eine gute Isolierung zu den Motorleitungen entsteht und die Platine sicher befestigt ist.

Von oben nach unten sind die Pads: SCK, RST, MISO, 5V, MOSI, GND

Kürze die dünnen Leitungen so, dass sie gut zwischen den anderen Bauteilen hindurchpassen und ohne Zugspannung nach oben zur Anschlussplatine gebracht werden können.

Spaghetti, oder blaue Würmer, was Dir Spaß macht. Aber Hey, Wenn's geht, warum nicht ;=)) Wenn möglich vermeide parallele Verlegung der Leitungen. Es ist besser für die Störsicherheit wenn sich die Leitungen kreuzen. Prüfe auch hier auf ungewollte Lötbrücken.

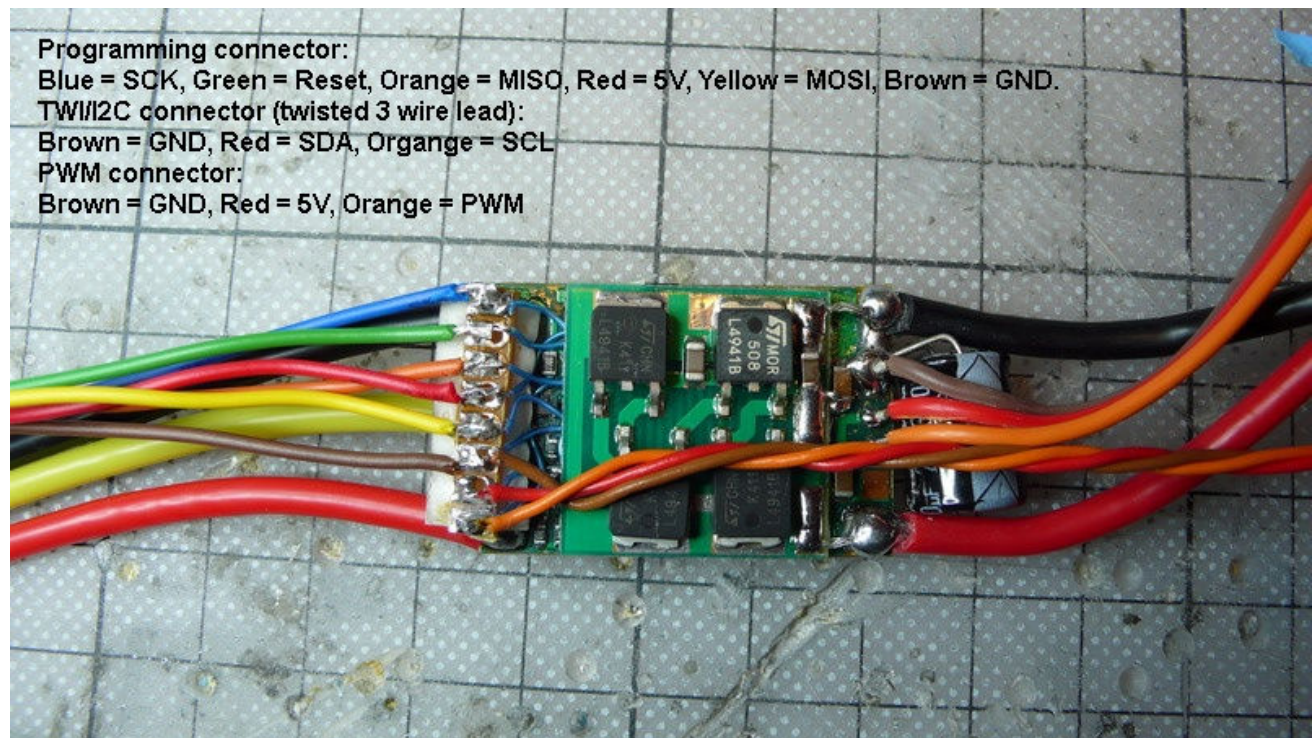
Beachte die kleinen Stücke doppelseitiges Schaumstoffklebeband an der Unterseite der Spannungsreglerplatine. Das soll das Risiko vermindern, dass die Isolierung der Leitungen durchgescheuert wird.



Biege die Spannungsreglerplatine zurück auf die Hauptplatine. Die Kante der Platine darf auf keinen Fall irgendwo die dünnen Drähte berühren.

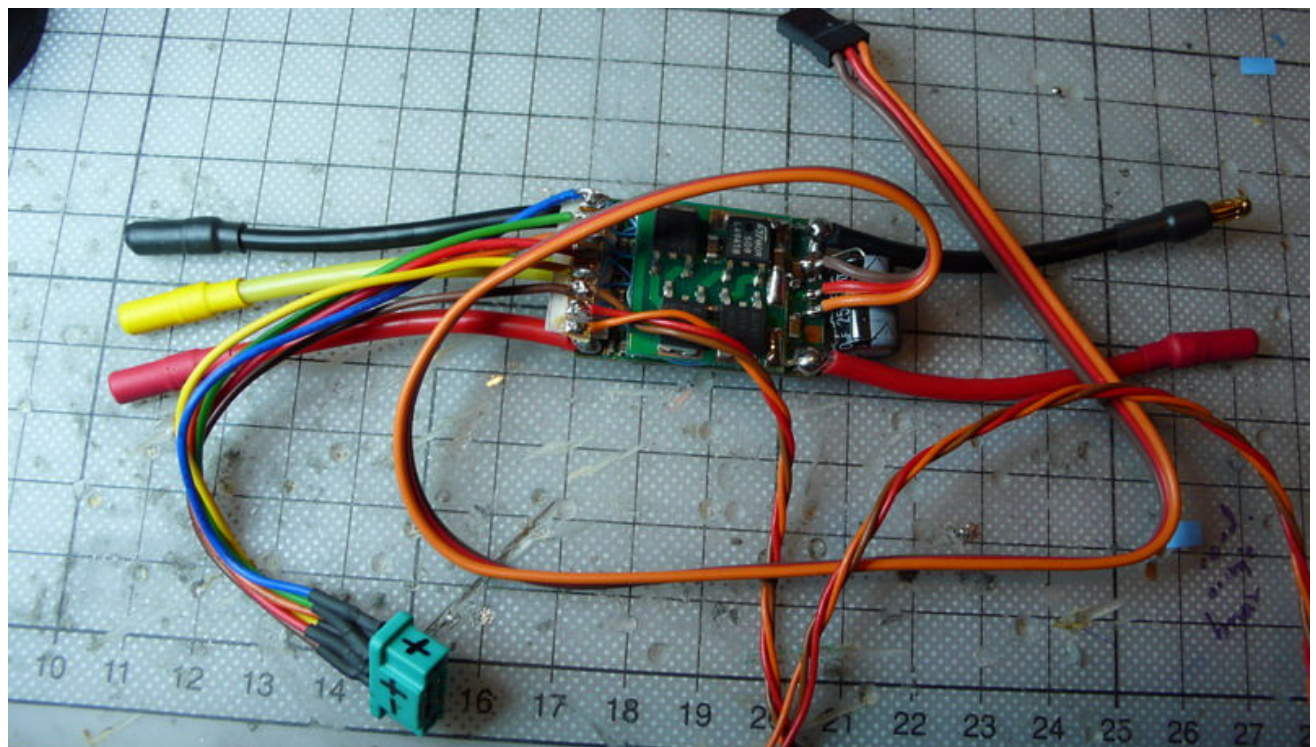
Auf diesem Bild habe ich schon meinen Programmieradapter angeschlossen. Achte auf die Farbcodierung der einzelnen Leitungen im Bild.





Ich finde, dass ein Multiplex Batterie Verbinder der ideale ISP-Verbinder ist: 6 Pins und keine Gefahr einer falsch gesteckten Verbindung. Die Pinbelegung (von hinten gesehen) ist:

	2) RST	4) 5V	
1) SCK			6) GND
	3) MISO	5) MOSI	





Beachte auch, falls Du es nicht schon getan hast, dass jetzt die Buchsen und Stecker an die Motor- und Batterieverbindungen gelötet werden müssen. Isoliere sie mit Schrumpfschlauch. Ich benutze 3mm Goldkontakt Stecker für die Motor- und Batterieverbinding. Die sind gut bis 35A und überschreiten damit die Spezifikation des Reglers. Für stärkere Regler nehme ich dann die 4mm Stecker für die Batterieverbinding. Oops... habe einen Schritt vergessen: Schließe **nicht** alles Mögliche an den Regler an. Benutze zunächst für die Tests ein geregeltes und strombegrenztes Netzteil mit 10-12 Volt und nicht einen teuren LiPo. Außerdem kann eine Art Sicherung (Autobirnenchen), die bei zu hoher Belastung durchbrennt, in der Spannungsversorgung hilfreich sein. Zuerst verbinde nur die Spannungsversorgung und prüfe ob tatsächlich 5Volt zwischen 5V Pad und GND anliegen. Wenn das der Fall ist, kannst Du den Programmieradapter anschließen. Ich benutze den SerCon (Serial Konverter) von der Mikrokopter Site (<http://www.mikrokopter.de/>). Beachte, dass der rote Jumper auf der SerCon Platine gesteckt ist und die grüne LED leuchtet um anzuzeigen, dass die 5V Versorgung über das ISP-Interface erfolgt.

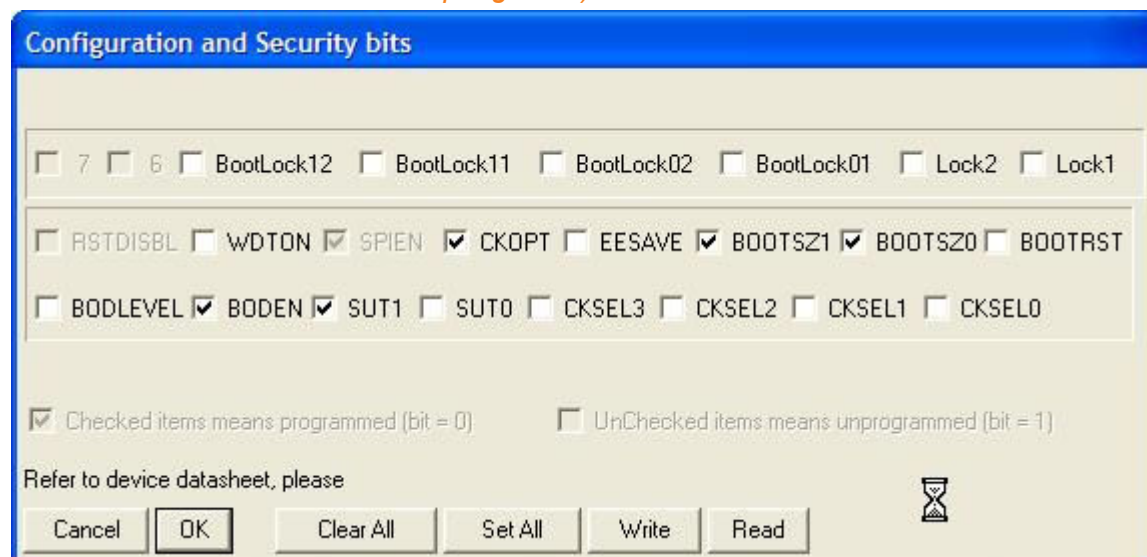
## 6. Einspielen der neuen Firmware

Starte Deine Programmiersoftware z.B. das kostenlose PonyProg Programm (<http://www.lancos.com/prog.html>). Ich gehe hier nicht in die Tiefe mit Erklärungen dazu. Es hat seine eigenen Hilfedateien.

Jetzt nimm das Firmware ZIP File (z.B. [http://home.versanet.de/~b-konze/low\\_cost/17a410\\_i2c\\_r07.zip](http://home.versanet.de/~b-konze/low_cost/17a410_i2c_r07.zip)) und entpacke es falls Du das noch nicht getan hast. Im Hauptverzeichnis findest Du ein GIF Bild, das die Fuse Bits anzeigt.

Setze die Bits wie abgebildet und schreibe sie in den Regler. Wenn das klappt, weißt Du das der Erfolg in Sicht ist ;=))

*(HJS: Ich empfehle erst mal die Fuse Bits auszulesen, um zu prüfen ob der Programmieradapter funktioniert. Dabei kann nichts kaputtgehen.)*



Jetzt lade das benötigte HEX File. Bei PPM-Ansteuerung bekommen alle Regler das gleiche Programm. Schreibe das HEX File in den Regler. Danach folgt eine Verifikation. Durch einen Fehler im PonyProg und/oder im SerCon Interface bekomme ich immer eine Fehlermeldung nach der Verifikation. Es scheint als wäre eine Änderung in einer INI Datei notwendig um das zu korrigieren aber ich bin ein bisschen faul.... Entferne den ISP-Stecker und die Spannungsversorgung vom Regler.

## 7. Testen des umgebauten Reglers

*HJS: Hier verlassen wir ein wenig die Beschreibung von Arthur, da dieser Abschnitt zu Mikrokopter und I2C spezifisch ist.*

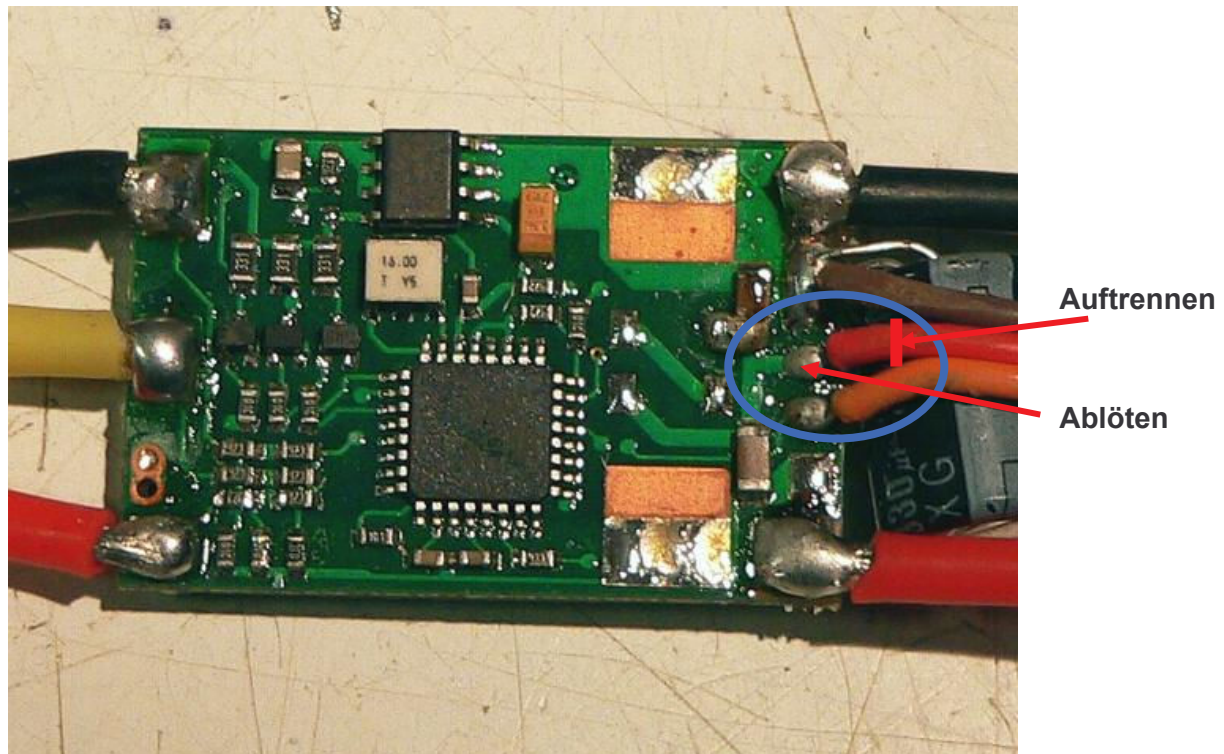
Der erste Funktionstest kann ganz einfach mit einem Servo-Tester oder direkt am Empfänger erfolgen. Stecke das Servokabel des Reglers an den Gaskanal des Empfängers. (Achtung die Stromversorgung des Empfängers erfolgt noch über den Regler) Schließe einen Testmotor an die Motorleitungen und eine strombegrenzte 10-12 Volt Spannungsquelle an die Batterieanschlüsse an.

Schalte den Fernsteuersender ein und stelle den Gashebel nach unten. Wenn jetzt die Stromversorgung eingeschaltet wird, sollte sich der Regler mit 4 aufsteigenden Piepslauten melden. Dann war die Programmierung erfolgreich. Wenn nicht, dann versuche es noch mal mit Setzen der Fuse Bits und Schreiben des Programms. Normalerweise klappt es gleich beim ersten Versuch, aber ich (Arthur) habe auch schon 2 oder selten 3 Versuche gebraucht. *(HJS: Bei mir hat es immer beim ersten Mal geklappt.)*

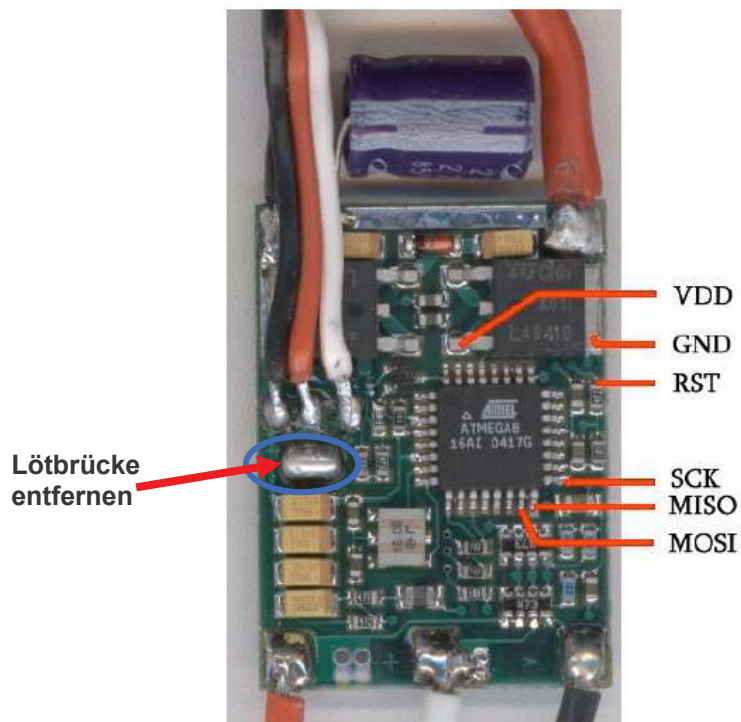
Wenn der Regler glücklich gepiepst hat, gib an der Fernsteuerung langsam Gas. Wenn alles richtig funktioniert, sollte der Motor bei niedrigen Gaswerten sicher anlaufen. Fahre das Gas wieder runter, löse alle Steckverbindungen, lehne Dich zurück und grinse glücklich: Du hast einen Regler erfolgreich umgebaut! ;=))

## 8. Abschließen des Umbaus

Entferne die Leitungen des Programmierinterface wieder von der kleinen Zusatzplatine. Achte darauf, dass die dünnen Drähtchen an der richtigen Stelle bleiben und sich keine Lötbrücken gebildet haben. Das Servokabel bleibt natürlich erhalten. Da die 5V des Reglers auf der QuadroControl zu Störungen führen können, sollte die Leitung unterbrochen werden.



Beim Typ 1 Layout gibt es eine Lötbrücke um das BEC-System auszuschalten



Wenn das geschehen ist kann der Regler wieder verpackt werden. Schneide ein Stück Schrumpfschlauch etwa 2cm länger als der Regler mit Kondensator. Sei vorsichtig beim Erwärmen, damit die Servoleitungen nicht schmelzen. Richtiges Einschrumpfen hält die kleine Anschlussplatine sicher in Position.

### Das Endergebnis:

Um dem Regler eine eigene kleine Note zu geben, habe ich den originalen Sticker von der alten Hülle abgezogen und auf den umgebauten Regler geklebt. Es ist jetzt ein **TowerPro 25A MAG8i ESC Für M1**. Und er hat den Qualitätstest bestanden.





## 9. Abschließende Worte

Natürlich könnt man auch QC als „Quax Conform“ lesen. Ohne die tolle Arbeit von Quax und einigen anderen Quadroter Genies würden wir diese schnelle Entwicklung der Systeme im letzten Jahr nicht erleben.

Natürlich kann niemand der die Firmware entwickelt und die notwendigen Hardwareänderungen herausfindet irgendeine Verantwortung für Deine Fehler oder ein Versagen des Reglers übernehmen. Zudem verlierst Du auf jeden Fall die Garantieansprüche an den Hersteller.

Trotzdem, wenn alles gut geht, wirst Du dich hoffentlich an vielen guten und sicheren Flügen mit diesem Reglern erfreuen.

## PS

Im Mikrokofter Forum hat Neo7530 einen alternativen Namen und einen Aufkleber für diese umgebauten Regler entworfen:



;=))

- Arthur P.