Aufbauanleitung für failsafe_dlux

Autor: Stefan Leidich
Kontakt: fireball@t-online.de
Version des Dokuments: V0.92

Features:

- Realisieren eines Failsafes für den dlux 250A HV Regler
  - Reaktion auf Unterspannung und Signalverlust
  - Senden des Stopp-Kommandos
- Eingangsspannung: 6,3-8,4V
- zwei parallele Reglerausgänge (Y-Kabel)
- Optional: Konvertierung des Eingangssignal von 0% bis +100% auf den Weg von -100% auf +100% für die Verwendung von Pistolensendern.
- https://www.youtube.com/watch?v=K2Zalw2rsIU

Disclaimer:

1. Beschaffung der Platine

Das Design der Platine wird als Eagle-Projekt zur Verfügung gestellt. Nahezu alle Leiterplatinenhersteller könne die *.brd Datei verarbeiten. Um das Board zu betrachten oder zu ändern ist die kostenlose Freeware-Version von eagle erforderlich:

http://www.cadsoft.de/download-eagle/eagle-freeware

Die Platine kann von jedem Hersteller gefertigt werden. Für die Bestellung der bestückten oder unbestückten Platine bei Bilex gibt es in Anlage A eine Musteremail.

2. Eingangskontrolle der Platine

Die Herstellung von Platinen erfolgt mit einer sehr geringen Fehlerquote. Dennoch sollte die Platine einer Sichtkontrolle unterzogen werden. Die Bauelemente sind in der Regel beschriftet. Der Bestückungsplan in Anlage B sollte als Vorlage verwendet werden.

3. Beschaffung des Arduino Nano

Der Arduino Nano sollte in der Version 3 (ATmega328P, 5V, 16MHz) beschafft werden. Andere Versionen wurden nicht getestet. Die Platinen sind zum Teil extrem günstig bei ebay zu bekommen. Den Nano gibt es komplett aufgebaut und mit beigelegter Stiftleiste zu kaufen. Wenn Löten absolut

keine Option ist, sollte der komplett aufgebaut gekauft werden. Dieser kann in das Board mit Buchsenleiste gesteckt werden.

4. Inbetriebnahme des Arduino Nano

Die Inbetriebnahme erfordert ein Mini B-USB Kabel und die Arduino-Software:

https://www.arduino.cc/en/Main/Software

Die Installation der Software ist selbsterklärend. Nach erfolgreicher Installation gilt es folgende Auswahl zu treffen:

Werkzeuge -> Platine -> Arduino Nano

Werkzeuge -> Prozessor -> ATmega328

Es bietet sich an den zur Verfügung gestellten Programmcode (*.ino) in dieses Verzeichnis zu kopieren:

C:\Users\xxxx\Documents\Arduino

Der erste Funktionstest des Arduinos sollte noch außerhalb der Platine erfolgen. Dazu den Arduino an das USB-Kabel anschließen und mit dem Computer verbinden. Unter dem Menüpunkt:

Datei -> Beispiele -> 01.Basics -> Blink

findet man ein sehr einfaches Programm mit dem man die Grundfunktion testen kann. Die Übertragung des Programms in den Arduino erfolgt über:

Datei -> Hochladen

Danach sollte die LED "L" langsam blinken. Wer Interesse hat kann in der Zeile mit delay(1000) die Zahl ändern und so die Frequenz des Blinkens einstellen.

Wenn man den zur Verfügung gestellten Programmcode in oben genanntes Verzeichnis kopiert, finde man das Programm unter

Datei -> Sketchbooks-> xxx

Nach Lesen des Abschnitts 5 und ggf. notwendiger Anpassung der Software kann das Programm hochgeladen werden.

Der Arduino kann nach oder vor dem Programmieren auf die Platine gesteckt oder in diese eingelötet werden. Abbildung 1 zeigt das Layout. Die Orientierung des Arduino ist klar ersichtlich. Der USB Anschluss muss nach rechts zeigen. **Aus fertigungstechnischen Gründen kann die Platine mit einer 16er-Buchsenleiste ausgestattet sein. Der Arduino hat jedoch nur 15 Pins. Es besteht somit die Gefahr, den Arduino falsch einzustecken bzw. einzulöten. Die nicht verwendeten Pins sind mit "not to use" beschriftet. Es ist somit klar, dass der Arduino linksbündig eingesteckt werden muss. Das falsche Einstecken kann zur permanenten Beschädigung führen.**

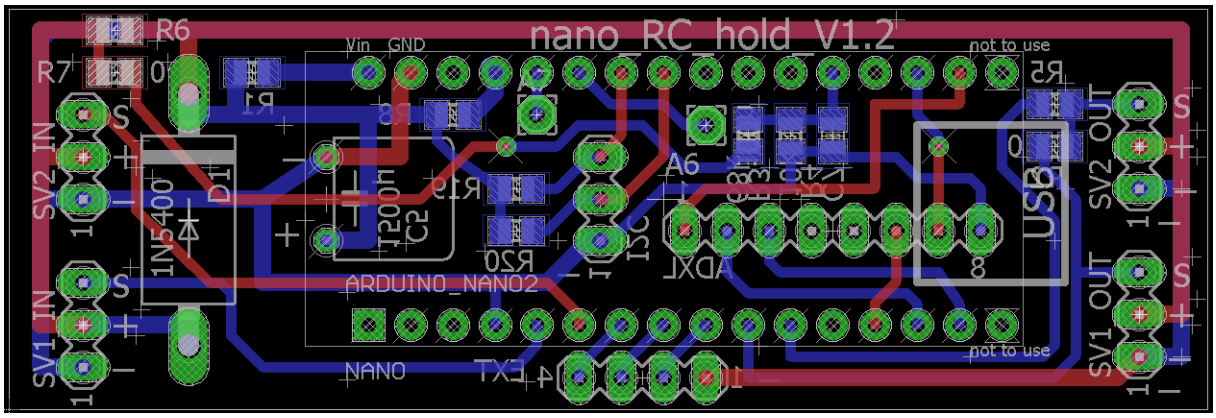

Abb. 1: Platinenlayout

5. Parametrisierung der Software

Die Software ist an verschiedene Anwendungsfälle anpassbar. Dies erfolgt über *define* Kommandos. Wenn die Zeile mit // beginnt, ist das Element nicht aktiv. Die Definitionen DEBUG, DEBUGFULL, DEBUGPPM und DEBUGVOL sollten nur erfahrene Programmierer verwenden. Jeder, der bei Durchsicht des Codes Fragen hat, wofür diese Kommandos gut sind, kann vermutlich auch nicht viel damit anfangen.

```
//#define DEBUG
//#define DEBUGFULL
//#define DEBUGPPM
//#define DEBUGVOL
#define FAILSAFEPPM //activate failsafe for PPM signal loss
#define FAILSAFEVOL //activate failsafe for under-voltage
#define BLINKON // enable blink code
//#define V6INPUT //12k_68k voltage divider for input voltage of 5.5V to 7.2V (initial solution)
#define V8INPUT //10k_68k voltage divider for input voltage of 5.5V to 8.4V (better solution)
//#define PISTOLSTICK // mapping of input signal 0%=-100% +100%=+100%
```

Listing A: Define Sektion

Definition FAILSAFEPPM

Mit dieser Definition wird der Failsafe für den Signalverlust definiert. Der Motor wird gestoppt, wenn das Modul für einen Zeitraum von mindestens 0,1s kein gültiges Signal empfangen. Die Zeitspanne wird durch diese Konstante in µs definiert:

```
const unsigned long delaylimit=100000; //100.000=0.1s failsafe in case of PPM signal loss
```

Die Zeitspanne sollte nicht zu kurz gewählt werden. Es gilt zu beachten, dass er Empfänger nur alle 20ms ein Signal sendet.

Definition FAILSAFEVOL

Mit dieser Definition wird die Überwachung der Eingangsspannung aktiviert. Durch Auskommentierung (also // voranstellen) kann somit die Hauptfunktion des Moduls deaktiviert werden. Sinnvoll kann dies sein, wenn man das Modul nur für die Konvertierung des Gaswegs zur Verwendung einer Pistolensteuerung einsetzt. Die Spannungsschwelle wird mit dieser Konstanten definiert:

```
const int voltagelimit=4500; //4500=4.5V threshold for faisafe
```

Die Schwelle von 4,5V hat sich bisher bewährt. Andere Werte sollten nur mit Vorsicht und mit ausreichend Sachverstand getestet werden.

Definition BLINKON

Die Definition BLINKON aktiviert den Blinkcode zur Anzeige von aufgetretenen Failsafe-Situationen während der Fahrt. Wenn die LED "L" im Ruhezustand bei einer Gasstellung oberhalb der Failsafe-Position (also -95% und höher) ein Mal blinkt, hat das Modul während der Laufzeit für einen bestimmten Zeitraum (siehe FAILSAFEPPM) kein gültiges Signal empfangen. Die Situation sollte nicht auftreten. Sie kann auftreten, wenn der Empfänger im Fall eines Empfangsverlustes kein Signal ausgibt (also kein Empfänger-Failsafe hat).

Wenn die rote LED zwei Mal blinkt, hat das Modul Unterspannung erkannt. Dies sollte man auch durch "stottern" des Motors gemerkt haben. Das Modul hätte dem Regler nämlich in diesem Fall zum Bremsen gezwungen. Eine mögliche Ursache für diesen Fall kann ein zu schwaches BEC sein. Wenn das Servo unter Last das BEC zum Einbrechen bringt, würde das Modul dies erkennen.

Definition V6INPUT, V8INPUT

Die Eingangsspannung wird vom Modul gemessen. Da der Arduino nicht ohne weiteres Spannungen oberhalb seiner eigenen Betriebsspannung messen kann, wird die Spannung durch R3 und R4 geteilt. Die erste Version der Hardware war für 6V Eingangsspannung ausgelegt. Der Spannungsteiler war mit 12k und 68k bestückt. Für den Betrieb bis 8,4V ist die Bestückung mit 10k und 68 erforderlich. Da der Arduino seine Außenbeschaltung nicht kennen kann, muss dies definiert werden. Es darf grundsätzlich nur eine der beiden Definitionen aktiv sein. Es gilt die für die Bestückung richtige zu wählen. (Die SMD-Widerstände sind beschriftet.)

Definition PISTOLSTICK

Mit dieser Definition wird die Konvertierung der Signalwege für Pistolenfernsteuerungen aktiviert. Pistolenfernsteuerung neutralisieren bei 0%. Viele Regler liefern dabei aber 50% Ausgangsleistung. Das Anlernen der Reglerwege ist oft nicht möglich. Die Funktion überträgt das Eingangssignal von 0% bis +100% auf den Weg von -100% bis +100%. Der Bremsbereich ist dann wirkungslos, da das Ausgangssignal -100% nicht unterschreiten kann. Die Definition der Mittenpositon erfolgt über diese Konstante:

```
const int servoneutral=1500; //neutral postion of pistole transmitter
```

Es ist sinnvoll mit der Gastrimmung den Zustand zu erreichen, dass in Neutralposition die rote LED "L" leuchtet. Wenn dies nicht der Fall ist, ist mehr Gasweg erforderlich, um das Anlaufen des Motors zu erreichen. Es gibt somit einen Todbereich.

Failsafe-Position

Die Failsafe-Postion beschreibt das Ausgangssignal im Fall des aktiven Failsafes. Der Wert kann über diese Konstante definiert werden:

```
const int failsafe_pos=1050; //output position in case of failsafe
```

Ein Wert von 1000(µs) entspricht ca. -125%. Test haben gezeigt, dass 1050(µs) besser sind, damit der Wert auch von jeder Fernsteuerung angefahren werden kann (in diesem Fall leuchtet die LED "L").

6. Inbetriebnahme des Gesamtsystems

Nach der Programmierung des Arduinos ist das System einsatzbereit. Das Modul wird linksseitig (SV1_IN) mit dem Empfänger verbunden (siehe Abb.2). Dafür ist ein Doppelbuchsen-Kabel erforderlich (oder Doppelstecker, je nach Verständnis von Stecker und Buchse). Der oder die Regler werden rechtsseitig verbunden (SV1_OUT, SV2_OUT). Für einen ersten Test bietet sich die Verwendung eines Servos an.

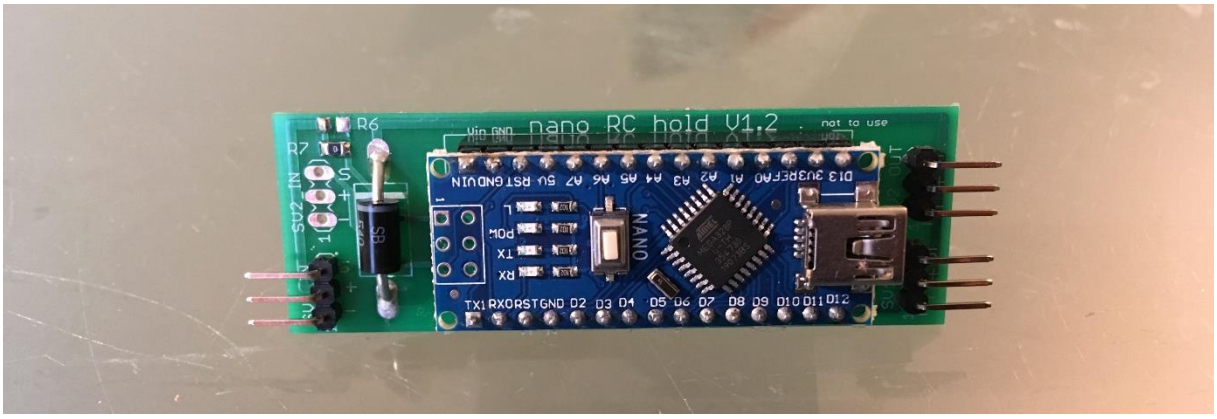

Abb. 2: Aufgebautes Modul

Wie in Abb. 2 ersichtlich sollte der Resettaster auf der Arduino-Platine entfernt werden. Es ist vorstellbar, dass der Taster durch Vibrationen im Betrieb gedrückt wird (Anschlagen an der Rumpfwand). Im Fall eines Resets wird für mehr als 1 Sekunde kein gültiges Signal ausgegeben. Der Motor läuft in diesem Fall weiter. Der Taster kann ohne Nachteile mit dem Lötkolben entfernt werden.

Es ist zum empfehlen die Platine mit WET.PROTECT zu imprägnieren und aufgrund der offen liegenden Leiter einzuschrumpfen. Die Montage im Boot sollte vibrationsgedämpft und nicht in unmittelbarer Nähe zum Empfänger bzw. zur Antenne des Empfängers (mit Telemetrie) platziert werden.

Die ordnungsgemäße Funktion sollte mit Hilfe des Prüfprotokolls aus Anhang C überprüft werden. Jede noch so kleine Abweichung vom erwarteten Verhalten sollte nicht ignoriert werden. Der Einfluss auf die Betriebssicherheit des Gesamtsystems ist nicht in jedem Fall vorhersehbar.

Anlage A: Email für die Bestellung der Platine bei Bilex

Empfänger: info@bilex-lp.com

Betreff: Bestellung Platine nano_hold

Sehr geehrte Damen und Herren,

bitte erstellen Sie mir ein Angebot für die Herstellung und Bestückung von Leiterplatinen. Die Layoutdatei befindet sich im Anhang. Die gewünschte Bestückung* ist ebenfalls im Anhang zu finden.

Anzahl: N
Leiterplatinendicke: 1,55mm (FR4)
Art: 2-Lagig, durchkontaktiert
Kupferdicke: 35µm
Größe: 72mmx24mm
Positionsdruck: Top
Lötstop: grün
Bearbeitung: gesägt
Bestückung nach Plan

Das Bauteil "NANO" sollte mit einer Buchsenleiste bestückt werden**
Das Bauteil "NANO" sollte nicht bestückt werden**

Bestellerinformation:
Vorname Name
Strasse Hausnummer
PLZ Ort
Deutschland

** Wenn die Bestückung verändert wird, sollte dies im Text explizit erwähnt werden, um eventuelle Fehler zu vermeiden.*
*** Treffen einer Auswahl. Die Bestückung mit einer Buchsenleiste erlaubt das Einstecken des Arduino Nano. Das Einlöten ist dann aber nicht mehr möglich. Wenn der Arduino eingelötet wird, sollte die Platine nicht mit der Buchsenleiste bestückt werden.*

Anlage B: Bestückung

Die Excel-Version der Stückliste enthält Shop-Links

Part	Value	Device	Package	Description
A6	NA	WIREPAD2,15/1,0	2,15/1,0	Wire PAD connect wire on PCB
A7	NA	WIREPAD2,15/1,0	2,15/1,0	Wire PAD connect wire on PCB
ADXL	NA	MA08-1	MA08-1	PIN HEADER
C5	1500µF, 16V	C-EU050H075X075	C050H075X075	CAPACITOR, European symbol
C7	NA	C-EUC0805	C0805	CAPACITOR, European symbol
D1	SB140, (SB540)	Diode	DO201-15	DIODE
EXT	NA	MA04-1	MA04-1	PIN HEADER
I2C	NA	MA03-1	MA03-1	PIN HEADER
NANO	ARDUINO Nano eingelötet oder 2x16(15) Buchsenleiste	ARDUINO_NANO2	ARDUINO_NANO2	Arduino Nano2
R1	0Ohm	R-EU_R0805	R0805	RESISTOR, European symbol
R2	0Ohm	R-EU_R0805	R0805	RESISTOR, European symbol
R3	68k	R-EU_R0805	R0805	RESISTOR, European symbol
R4	10k	R-EU_R0805	R0805	RESISTOR, European symbol
R5	NA	R-EU_R0805	R0805	RESISTOR, European symbol
R6	NA	R-EU_R0805	R0805	RESISTOR, European symbol
R7	0Ohm	R-EU_R0805	R0805	RESISTOR, European symbol
R8	NA	R-EU_R0805	R0805	RESISTOR, European symbol
R19	NA	R-EU_R0805	R0805	RESISTOR, European symbol
R20	NA	R-EU_R0805	R0805	RESISTOR, European symbol
SV1_IN	Stecker gewinkelt	MA03-1	MA03-1	PIN HEADER
SV1_OUT	Stecker gewinkelt	MA03-1	MA03-1	PIN HEADER
SV2_IN	NA	MA03-1	MA03-1	PIN HEADER
SV2_OUT	Stecker gewinkelt	MA03-1	MA03-1	PIN HEADER

Anlage C: Prüfprotokoll (beispielhaft ausgefüllt)

Test / Aktivität	Ergebnis	Bearbeiter	Datum
Prüfen der Platinenversion und Bestückung Vergleichen mit Bestückungdplan	Nano_RC_hold _V1.2	Fireball412	22.06.19
Sichtprüfung und Doku (Abb. C1/2)		Fireball412	22.06.19
Aufspielen der aktuellen Software: failsafe_dlux_V_1_0_2.ino #define BLINKON #define FAILSAFEVOL #define FAILSAFEPPM #define V8INPUT		Stromfresser	xx.05.19
Testaufbau nach Abb. C3 aufbauen	ok	Fireball412	22.06.19
Mit Servotester (ohne Servo oder mit HV Servo) an 8,4V für 30 Minuten betreiben	ok	Fireball412	22.06.19
Mit Servotester (an 6,5V) verbinden und Servofunktion an Out 1 und Out 2 prüfen	ok	Fireball412	22.06.19
Prüfen, ob die LED leuchtet, wenn die Failsafe-Position angefahren wird	ok	Fireball412	22.06.19
Mit Servotester (an 6,5V) verbinden und Signalloss (kurzzeitiges Trennen des Signalkabels, siehe Abb. C4) emulieren. Prüfen ob LED danach einmal blinkt.	ok	Fireball412	22.06.19
Mit Servotester (an 6,5V) verbinden und Unterspannung emulieren (durch langsames Spannung reduzieren). Spannungslimit notieren (Sollwert ~4,65V). Spannung auf 6,5V erhöhen und prüfen ob LED zweimal blinkt und die Servofunktion wieder gegeben ist.	4,63V ok	Fireball412	22.06.19
Neustart durch Spannungsunterbrechung	ok	Fireball412	22.06.19
Mit Servotester (an 6,5V) verbinden und 1h laufen lassen. Prüfen, ob keine Failsafe-Situation aufgetreten ist (also LED blinkt nicht).	ok	Fireball412	22.06.19
Prüfen, ob nach 1h Testlauf Servorreaktion noch gegeben ist.	ok	Fireball412	22.06.19
Prüfen, ob nach 1h Testlauf die Unterspannungserkennung noch funktioniert (durch Spannungsreduzierung, siehe oben)	ok	Fireball412	22.06.19
Neustart durch Spannungsunterbrechung	ok	Fireball412	22.06.19
Prüfen auf Servoreaktion und nicht blinkende LED	ok	Fireball412	22.06.19
Funktionskontrolle im Modell (unter anderem bei laufendem Motor den Servostecker aus dem Empfänger ziehen; weitere Tests nach Situation)			

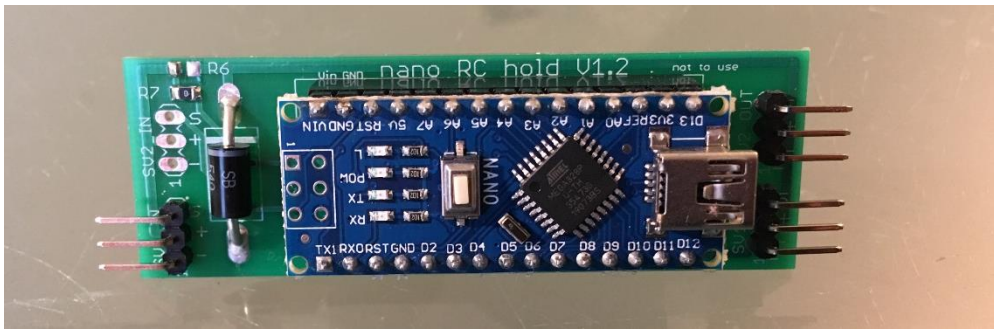

Abb. C1: Platinenoberseite

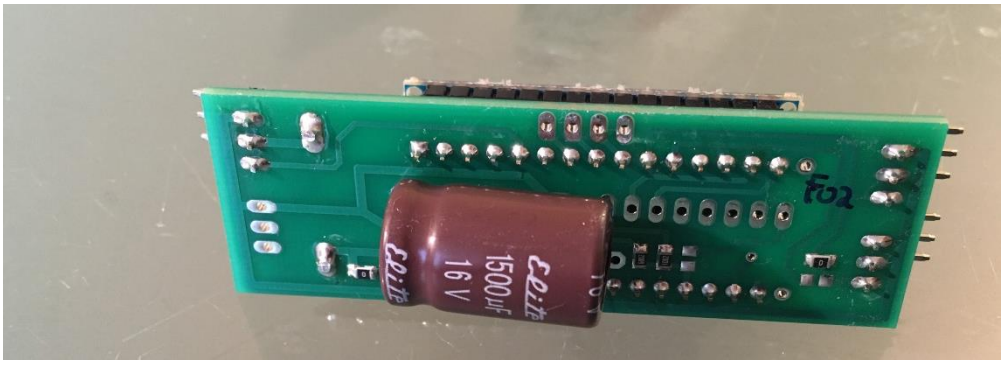

Abb. C2: Platinenunterseite

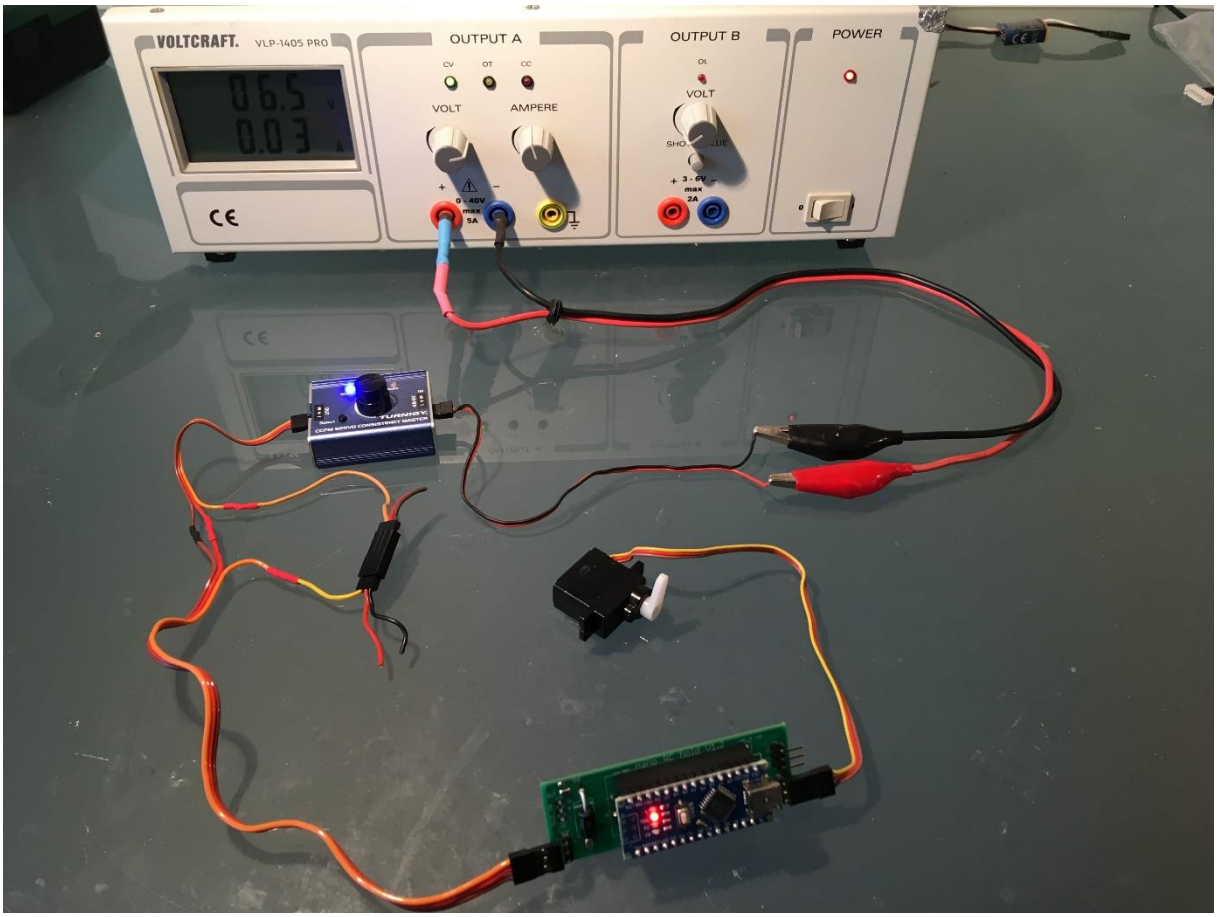

Abb. C3: Testaufbau

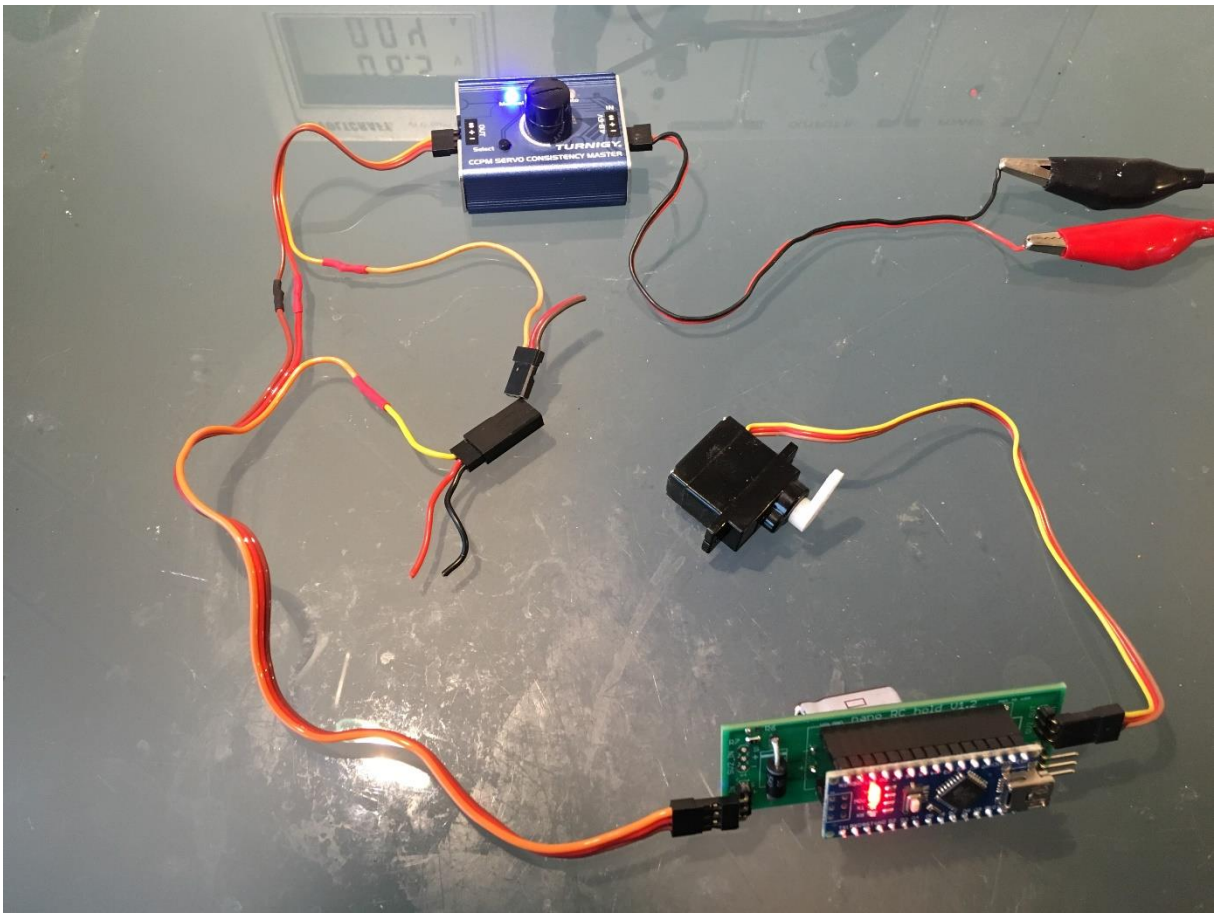

Abb. C4: Testaufbau (aufgetrenntes Signalkabel)