KopterLight-EXT

34

Magomora
MikroKopter.de

Inhaltsverzeichnis

1 Kopter Light mit externer Ansteuerung

by Magomora

Wer Rechtschreibfehler oder sonstige Fehler findet, darf sie gerne behalten. (Oder mir mitteilen, oder selber korrigieren)

Die hier zur Verfügung gestellten Informationen sind so sorgfältig wie möglich erstellt worden.
Für evtl. Fehler in den Informationen (Ich bin auch nur ein Mensch), bzw. Schäden, die durch diese oder fehlerhafte Umsetzung dieser Informationen entstehen, übernehme ich keine Haftung !
Jeder haftet beim Nachbau für sich selber.

Ich versuche aber gerne beim Nachbau zu helfen.

Der Thread dazu, für Diskussionen und weitere Infos: Digitale-RGB-Strips

-
-

Inhaltsverzeichnis

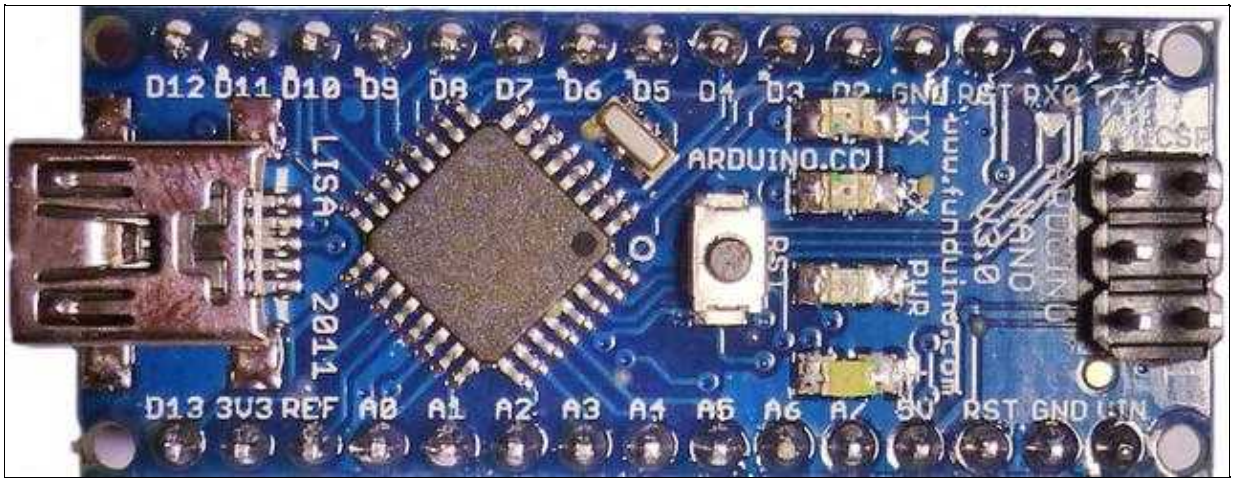
2 Allgemein

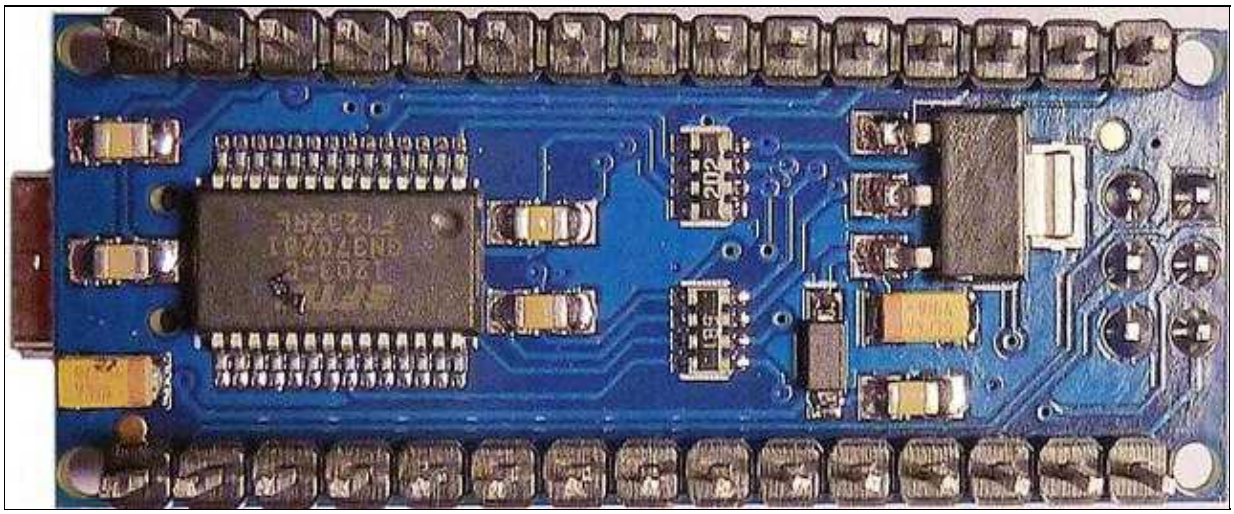
Wer zum ersten mal hier ist, sollte alles in Ruhe durchlesen.

Wer sich schon mit den vorherigen Versionen beschäftigt hat, sollte sich besonders
mit dem Softwareteil noch mal beschäftigen, da es dort diverse Änderungen gegeben hat.

2.1 Arduino Nano

Mit diesem Mikrokontroller werden unsere LED Streifen zum Leben erweckt.



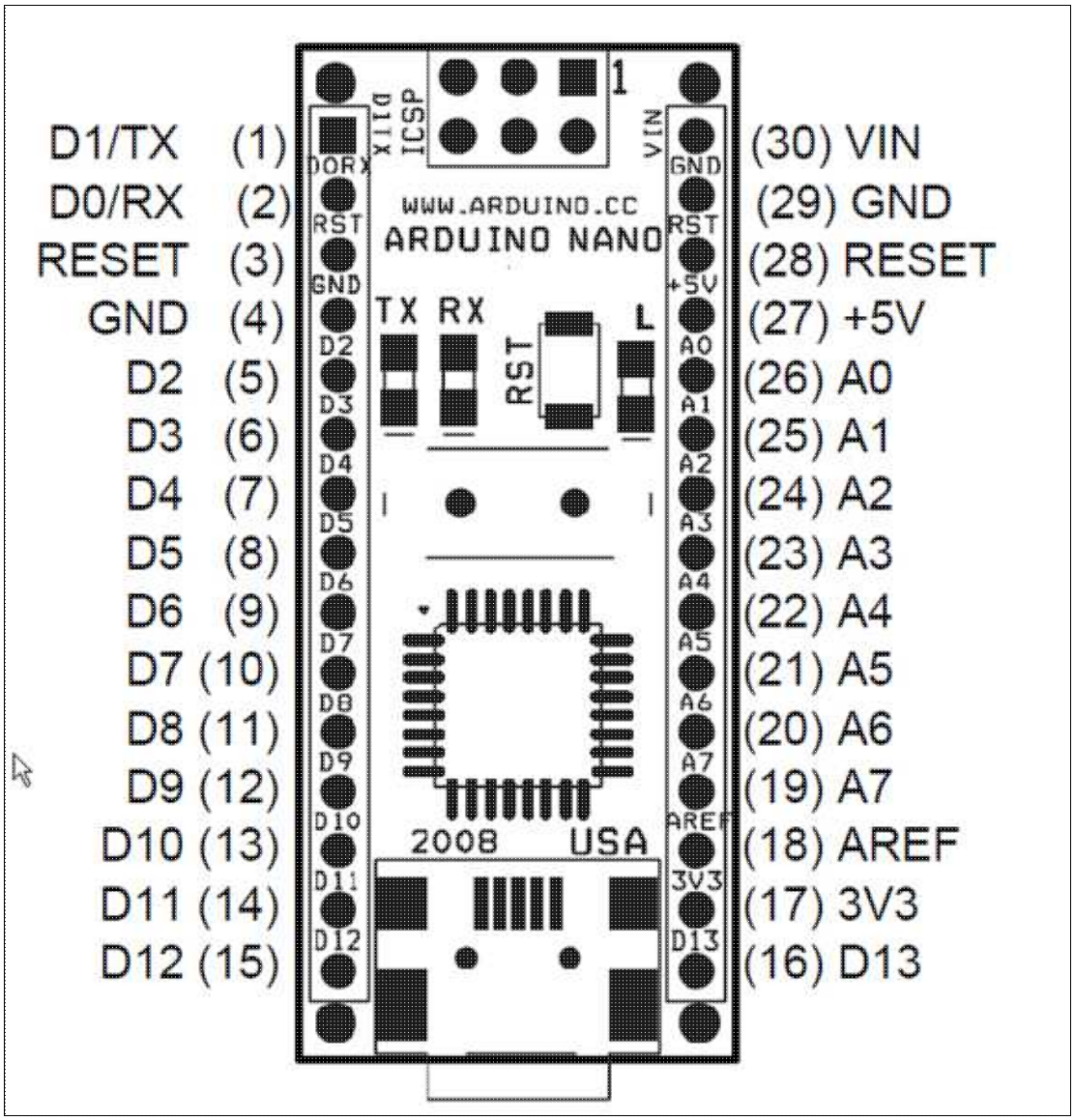


2.1.1 Arduino Spezifikationen:

2.1.1.1 Schaltplan

Arduino Nano V3.0 Schaltplan

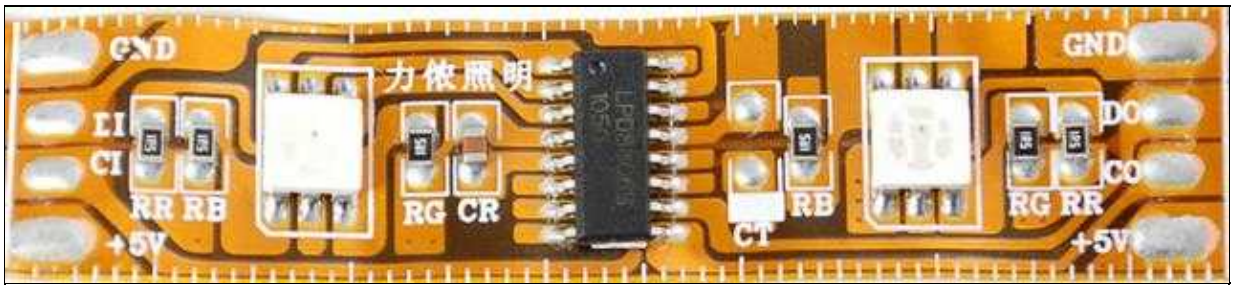
2.1.1.2 Pinlayout



Pin No.	Name	Funktion	Beschreibung
1-2, 5-16	D0-D13	I/O	Digital input/output port 0 to 13
3, 28	RESET	Input	Reset (active low)
4, 29	GND	PWR	Supply ground
17	3V3	Output	+3.3V output (from FTDI)
18	AREF	Input	ADC reference
19-26	A7-A0	Input	Analog input channel 0 to 7
27	+5V	Output oder Input	+5V output (vom on-board Regler oder +5V (input von externer Spannungsversorgung)
30	VIN	PWR	Supply voltage

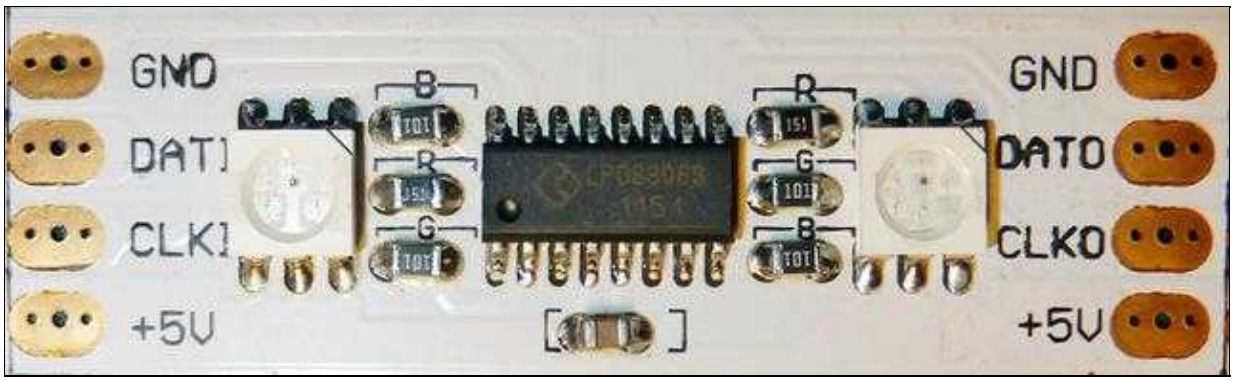
2.2 LPD8806 basierte RGB LED Streifen

Die beiden Streifen unterscheiden sich nicht nur im Design. Ein sehr gravierender Unterschied besteht auch in der Programmierung.
Bei den Streifen aus der China Sammelbestellung muss bei Verwendung eines Adafruit basierten Programcodes Rot und Grün getauscht werden (dieses ist im Kopter Light Code berücksichtigt).



D-RGB-LED Sreifen von Adafruit : Tech specs (newer LPD8806 type)

- 16.5mm (0.65") breit, 4mm (0.16") dick mit Umhüllung, 62.5mm (2.45")Länge pro Segment und 32 LEDs pro Meter
- entfernbare IP65 Wasserschutz-Umhüllung
- Maximum 5V @ 120mA pro Segment (alle LEDs auf voller Helligkeit) Eingangsspannung 5V DC (6V DC nicht überschreiten) - kein Verpolungsschutz
- 2 RGB LEDs mit gemeinsamer Anode pro Segment, individuell programmierbar
- Wellenlänge der LEDs: Rot 630nm / Grün 530nm / Blau 475nm



D-RGB-LED Sreifen aus China Sammelbestellung : Tech specs

- 15mm breit, 4mm dick mit Umhüllung,50mm Länge pro Segment, 40 LEDs pro Meter
- entfernbare IP65 Wasserschutz-Umhüllung
- Maximum 5V @ ca. 80-90mA pro 50mm Stripe-Segment (alle LEDs auf voller Helligkeit), Eingangsspannung 5V DC (5,8V DC nicht überschreiten) - kein Verpolungsschutz
- 2 RGB LEDs mit gemeinsamer Anode pro Segment, individuell programmierbar
- LED Typ SMD5050
- durchschnittliche Lebensdauer ca. 50000 Std.

3 Kopter Light mit externer Ansteuerung

3.1 Was benötige ich

- Ein bischen Zeit, Geduld und den Willen alles sorgfältig zu lesen und mich in den Code einzuarbeiten.
- Spaß am Experimentieren
- Einen Lötkolben
- Lötzinn
- Einen Seitenschneider (eine zusätzliche Abisolier-Zange wäre von Vorteil)
- Kabel und Stecker (je nach Bedarf)
- Digitale RGB LED Streifen (aus der China Sammelbestellung)
- Einen Arduino Nano V3.0
- Einen Switch BEC 3-5A, 5V Out für bis zu 7S (je nach Menge der verbauten LEDs und vorhandenem Akku, für die +5V Spannungsversorgung)

3.2 Aufbau

Zum Testen kann man ein regelbares Netzteil mit einstellbarer Spannung und Strom verwenden.
Um das ganze nachher unter dem Kopter zu betreiben, sollte man sich einen Switch-BEC mit ca. 3-5A bei 5V Out gönnen.
Dies hat den Vorteil, dass man den Arduino gleichzeitig ebenso mit 5V versorgen kann.
Die FC sollte man nicht damit belasten.

Den Streifen immer NUR mit maximal 5V versorgen!

Anderfalls hat man nicht lange was von den Streifen. Der Arduino kann zwar bis ca. 12V vertragen, aber warum unnötig Leistung in Wärme umsetzen.

Beim Anschluss per USB an den PC zum Programmieren, immer die externe Spannungsversorgung und SPI Kabel vom Arduino trennen, wenn keine externe Spannung an Vin anliegt !!

Da sonst die Schaltung über den USB-Port (welcher nur max. 500mA liefern kann) mit Strom / Spannung versorgt wird und dieser dann überlastet wird.
Dann kann man sich evtl. ein neues Mainboard kaufen. Neuere Mainboards haben zwar teilweise einen Schutz eingebaut, aber darauf sollte man sich nicht verlassen.

In den folgenden Schaltungen ist, um die Schaltung nicht unnötig kompliziert zu machen, je Arm nur ein LED-Segment eingezeichnet.
Es können aber beliebig viele Segmente pro Arm verbaut werden. Es muss aber auf jedem Arm die gleiche Anzahl Segmente verbaut werden. (Je nachdem, was Euer Kopter an Strom zur Verfügung hat und tragen kann)
Insgesammt können max. 8 Arme angesteuert werden.

Tip: Die Verbindungen zum Arduino (Spanunnungsversorgung, SPI-Leitungen) sollten steckbar gemacht werden.

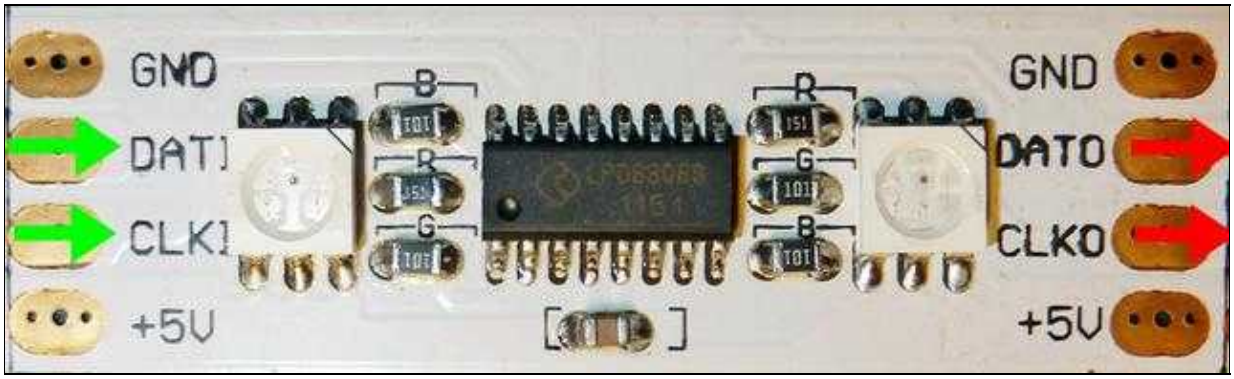
3.2.1 SPI-Anschluss:

Ab der Version mit externer Ansteuerung wird nur noch der Hardware SPI Anschluß unterstüzt !

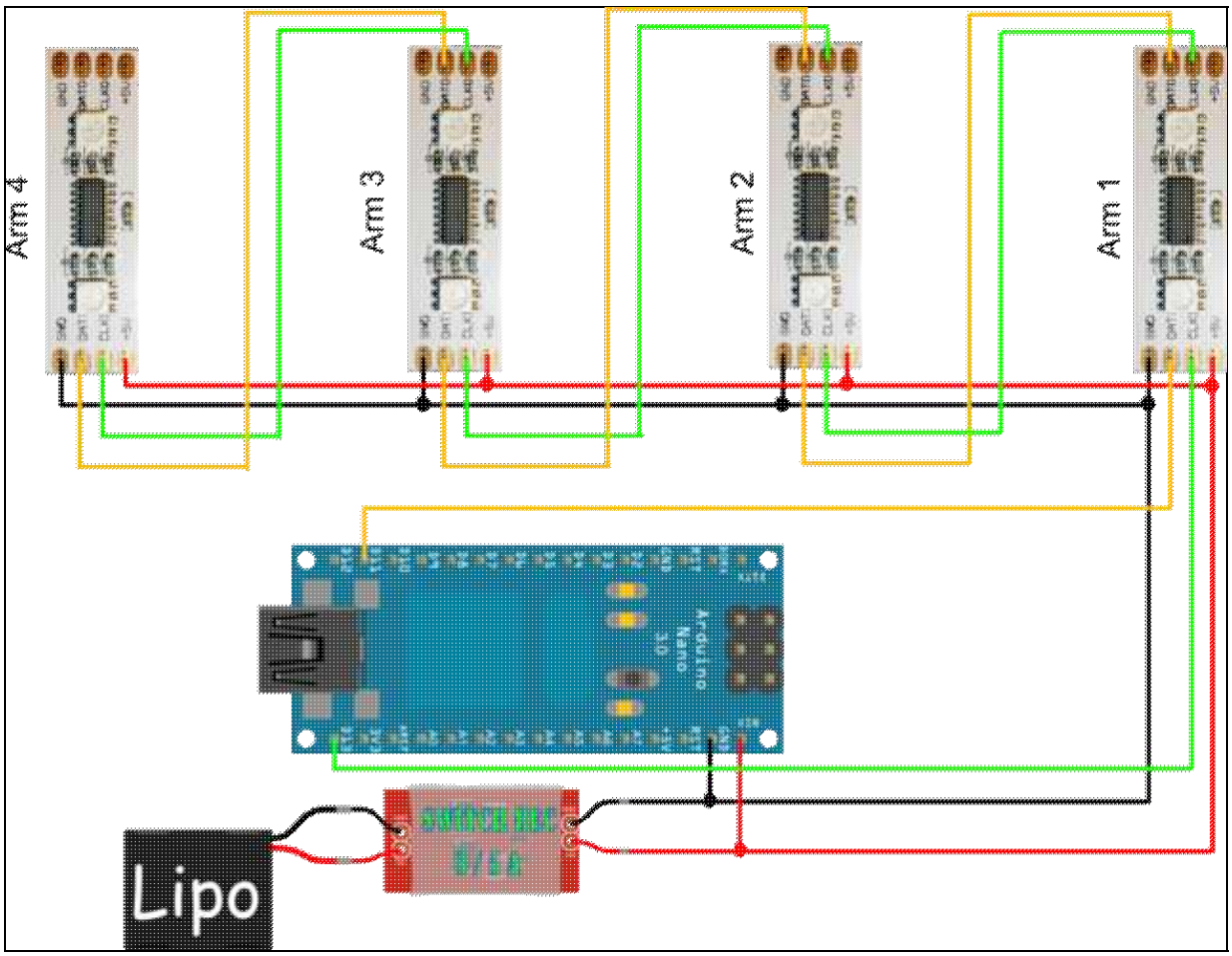
Den Clock- und Data-Ausgang vom Arduino schließt ihr an den CLKI / DATI (Clock In / Data In) von dem ersten Streifen an.

Pin No.	Funktion	Beschreibung	Speed
D11	Data	DATA Out des Arduino für **Hardware** SPI	High
D13	Clock	CLK Out des Arduino für **Hardware** SPI	High

Bitte achtet darauf, die Streifen richtig anzuschließen. Es gibt für Clock und Data jeweils ein IN und OUT



3.2.2 Kopter Light Grundschaltung mit Hardware SPI:



3.3 Software

Die Software ist so ausgelegt, dass man damit 1 bis max. 8 Arme ansteuern kann. Ein Arm fliegt natürlich nicht, aber siehe Tips und Tricks.

3.3.1 Download:

Folgende Software benötigen wir um das Kopter Light zum Laufen zu bringen.
Es kann jeweils die aktuelle Version herunter geladen werden. Bitte auf OS Version achten !

Name	Beschreibung	Link
FTDI VCP Driver	Virtual COM port driver	http://www.ftdichip.com/Drivers/VCP.htm
Arduino IDE	Arduino Programmieroberfläche um Arduino Code (Sketch) zu bearbeiten.	http://arduino.cc/en/Main/Software
Kopter	Sketch zur RGB LED	http://mikrokopter.de/mikrosvn/Projects/Digital_RGB_LED_Stripes/tags/KopterLi

Light - EXT ***Ver 1.0***	Streifen Steuerung. (Modifizierte Adfruit Library, Kopter Light Sketch mit Lichtmuster Sketch)	

3.3.2 SVN:

Wer Abwandlungen oder eigene Licht-Effekte veröffentlichen möchte, kann dies gerne im branches Ordner im SVN tun.
Legt einfach einen neuen Ordner mit eurem Forumsnamen an (im branches Ordner) und speichert euren Code darin.

Den Ordner findet ihr unter: **Subversions Projekte: Projects/Digital_RGB_LED_Stripes**

Branches-Ordner

3.3.3 Installation:

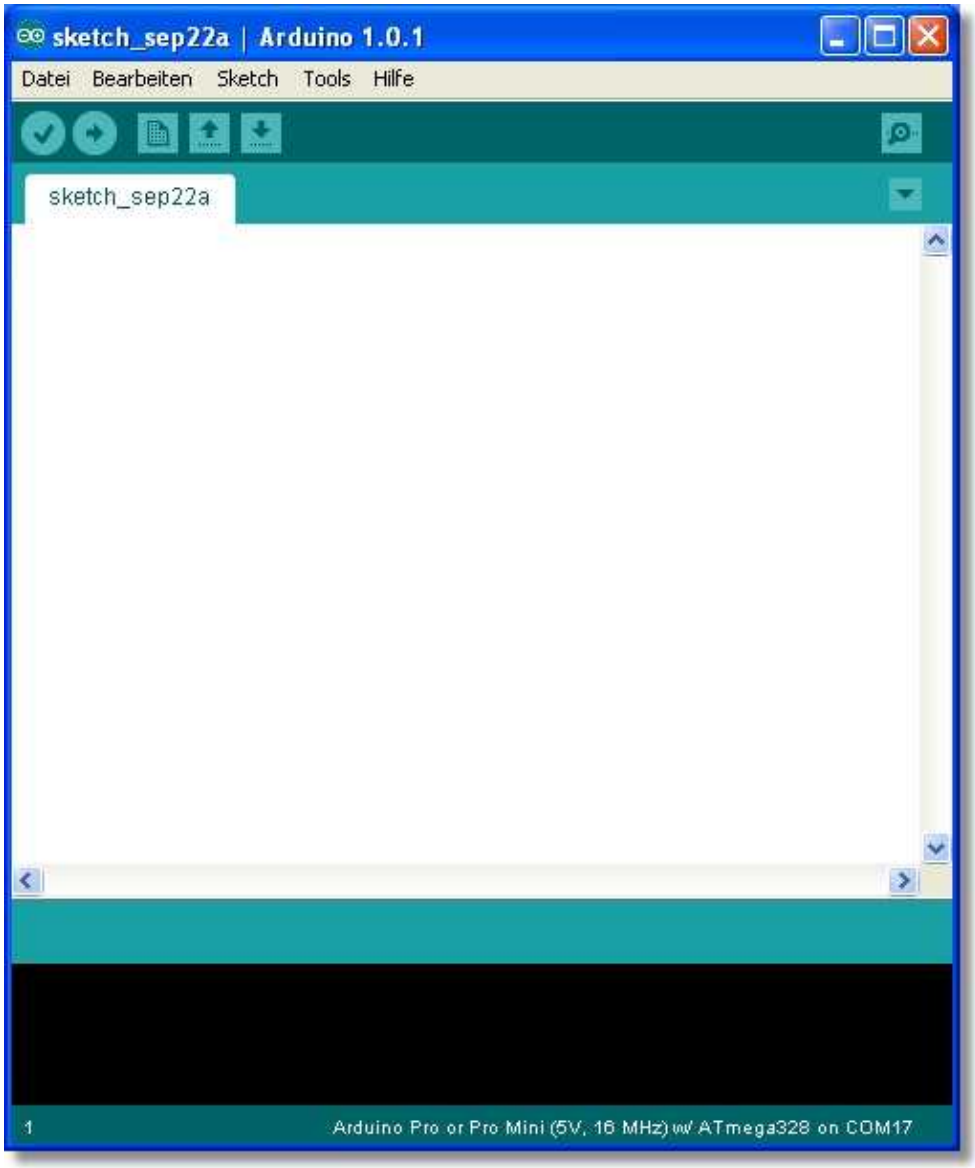
Die folgenden Beschreibungen sind allgemein gehalten (primär für Windows User), da die Beschreibung für alle möglichen OS hier den Rahmen sprengen würde.

FTDI Treiber: (Wer schon mal diese Treiber installiert oder einen Arduino an seinem PC hatte, kann hier überspringen)

- ZIP Datei in ein Verzeichnis deiner Wahl entpacken.
- Arduino an den PC anschliessen.
- Bei der Windows Meldung "Treiber nicht gefunden", über Installationsquelle selber wählen, zu dem Verzeichnis wechseln wohin ihr die ZIP-Datei entpackt habt.
- Ersten Eintrag auswählen und bestätigen.
- Nach Abschluss der Installation / Initialisierung in der Systemsteuerung unter COM-Ports / Serielle-Schnittstellen die zugewiesene Portnummer nachschauen und merken !!
- Danach ziehen wir den Arduino wieder vom PC ab.

Arduino IDE: (Wer schon mal die Arduino IDE installiert benutzt hat, kann hier überspringen)

- ZIP Datei in ein Verzeichnis deiner Wahl entpacken.
- Aus diesem Verzeichnis die Datei "arduino.exe" starten
- Wenn ihr folgendes Fenster seht, ist alles OK und ihr könnt die Arduino IDE wieder schliessen:


sketch_sep22a | Arduino 1.0.1
Datei Bearbeiten Sketch Tools Hilfe
sketch_sep22a
1
Arduino Pro or Pro Mini (5V, 16 MHz) w/ ATmega328 on COM17

-

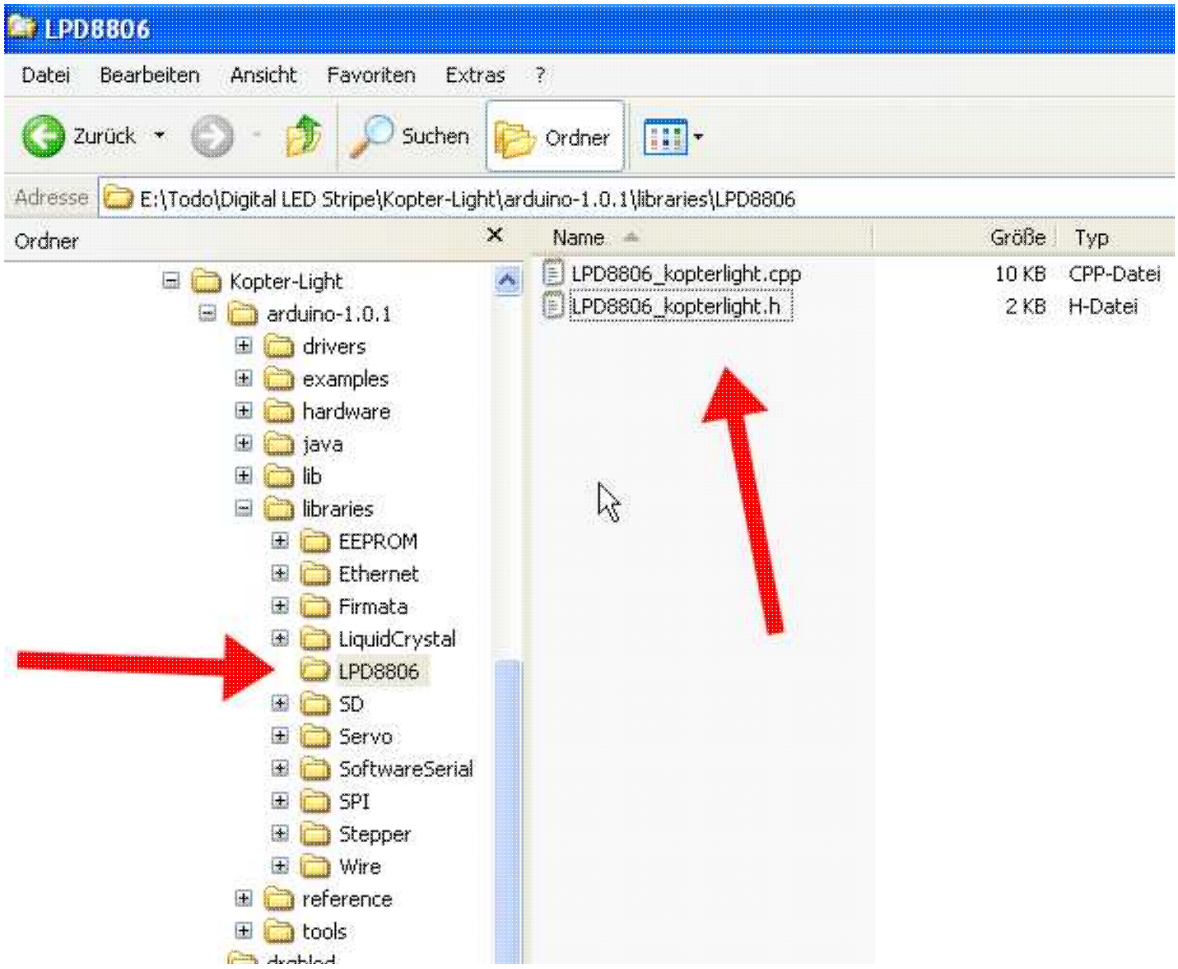
Kopter Light:

(x_xx im Namen der Datei KopterLightEXT_x_xx.ino steht für die Versions Nummer)

- Die herunter geladene gepackte Datei enthält 6 Dateien:

Datei	**Beschreibung**
KopterLightEXT_x_xx.ino	Steuer-Datei: Hier wird das Kopter Light konfiguriert.
Sequences.ino	Hier befinden sich alle Lichteffekte.
Lightshow_Std.ino	Hier erstellt ihr eure Lightshow ohne externe Ansteuerung.
Lightshow_Remote.ino	Hier erstellt ihr eure Lightshow mit externer Ansteuerung.
LPD8806_kopterlight_EXT.cpp LPD8806_kopterlight_EXT.h	In diesen Dateien steckt die eigentliche Intelligenz des Kopter Lights.

- Die Dateien LPD8806_kopterlight_EXT.cpp und LPD8806_kopterlight_EXT.h müsst ihr in das Arduino-libraries-LPD8806 Verzeichnis kopieren.
- Es dürfen nur diese beiden Dateien in diesem Verzeichnis sein !



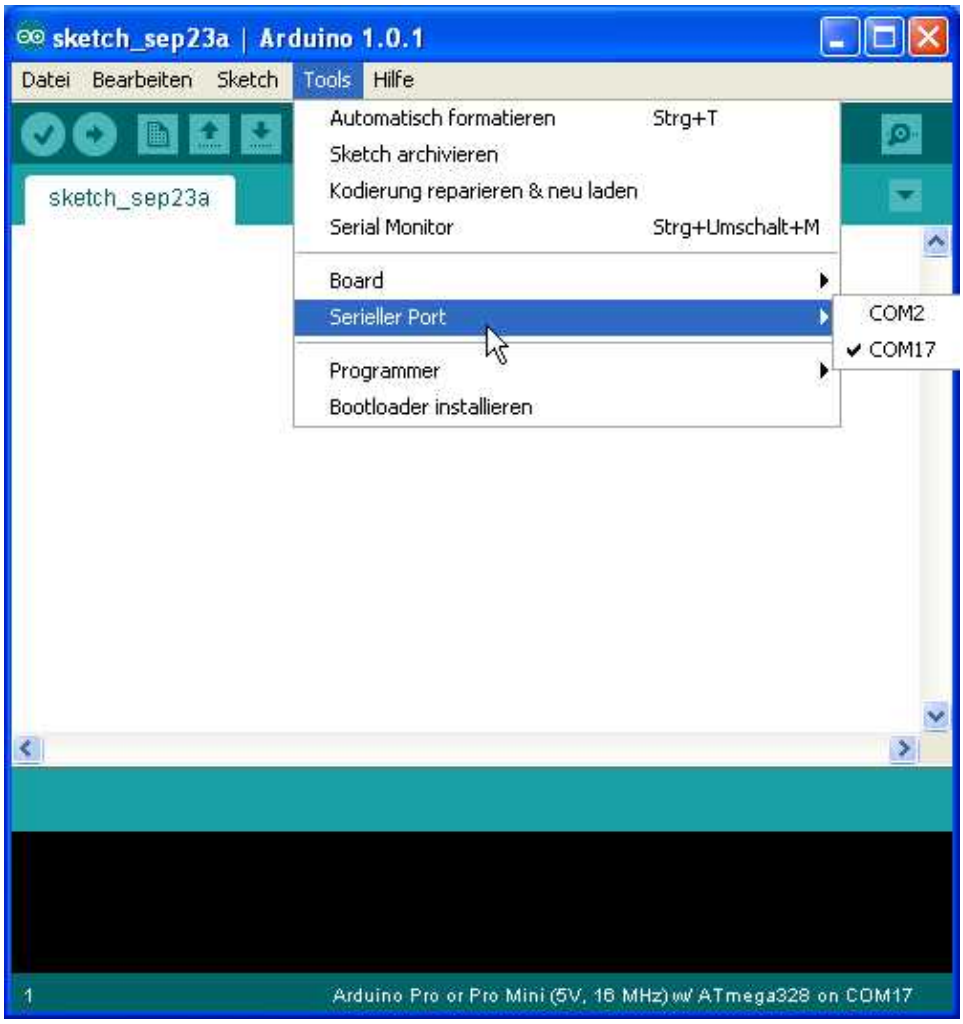
- Sollte das Verzeichnis "LPD8806" noch nicht vorhanden sein, müsst ihr dieses erstellen.

- Die Dateien KopterLightEXT_x_xx.ino, Sequences.ino kopiert ihr in ein Arbeitsverzeichnis.
- Dieses Verzeichnis muss den Namen "KopterLightEXT_x_xx" haben.
- Zusätzlich müsst ihr noch **eine** der beiden Lightshow-Dateien (Lightshow_Std.ino, Lightshow_Remote.ino)mit in dieses Verzeichnis kopieren.
- Das hängt davon ob ihr das Kopter Light mit oder ohne externe Ansteuerung betreiben wollt.
- Für einen ersten Test solltet ihr mit der **Lightshow_Std.ino** anfangen.

Der Vorteil von den beiden Dateien Lightshow_Std.ino und Lightshow_Remote.ino ist, wenn ihr Lightshows später untereinander austauschen
wollt, dann braucht ihr einfach nur eine dieser Dateien tauschen, ohne das ihr das Setup von jemand anderes wieder an euer eigenes anpassen müsst.

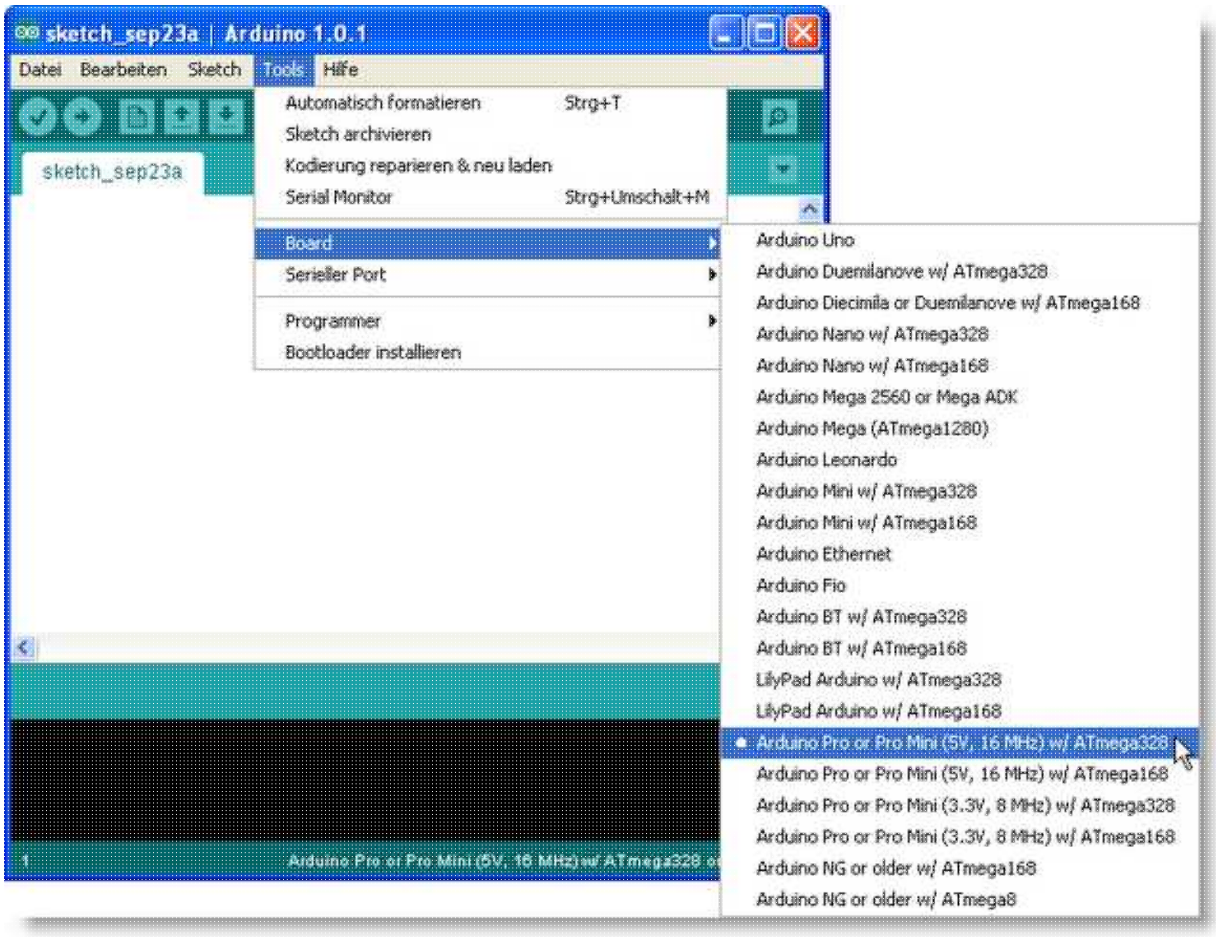
3.3.4 Arduino IDE:

- Als erstes stecken wir nun den Arduino wieder an den PC.
- Nun starten wir die Arduino IDE erneut.
- Zu erst müssen wir nun den Com-Port (den wir uns schon bei der Treiberinstallation gemerkt haben) festlegen:


sketch_sep23a | Arduino 1.0.1
Datei Bearbeiten Sketch Tools Hilfe
sketch_sep23a
Automatisch formatieren Strg+T
Sketch archivieren
Kodierung reparieren & neu laden
Serial Monitor Strg+Umschalt+M
Board
Serieller Port
Programmer
Bootloader installieren
COM2
COM17
1
Arduino Pro or Pro Mini (5V, 16 MHz) w/ ATmega328 on COM17

- Danach legen wir noch fest, welchen Arduino wir am PC angeschlossen haben. Da es in der Auswahliste keinen Nano gibt,

    - ♦ wählen wir den Arduino Pro or Pro Mini (5V, 16MHz) w/ ATmega328. (Der unterschied zum Nano besteht nur darin, das der Nano noch zusätzlich einen USB-Anschluss mit auf dem Board hat.)



3.3.5 Kopter Light Code:

Als erstes öffnen wir nun die Datei KopterLightEXT_x_xx.ino. Dies ist auf zwei Wegen möglich:

- Die Arduino IDE starten und über Datei / Öffnen die Datei laden oder
- einen Doppel-Klick auf die Datei KopterLightEXT_x_xx.ino. Die Arduino IDE wird dann automatisch geladen.
- Die Datei Sequences.ino und die eine von euch gewählte Lightshow-Datei werden bei beiden Möglichkeiten immer automatisch mit geladen.

3.3.6 Setup:

Um das Sketch an das eigene Setup anzupassen, suchen wir nun folgende Bereiche im Sketch (KopterLightEXT_x_xx.ino):

Als erstes definiert Ihr, ob und wie Ihr die externe Ansteuerung haben wollt. (0 = OFF, 1 = Digital getriggert, 2 = PWM.)

```
/*****************************************************************************/
// To be changed according to your setup ¶
/*****************************************************************************/
// Remote control¶
// 0 = OFF ( default )¶
// 1 = Digital input¶
// 2 = PWM input¶
uint8_t RMTCT = 0;¶
```

Hier gebt ihr die Anzahl der Lichteffekte an, die durchschaltbar haben wollt (Lightshow_Remote.ino).
Auf Grund der technischen Beschaffenheit des Drehpotis und der PWM-Signale der Sender, solltet ihr nicht mehr als

- 11 Effekte zum durchschalten erstellen.

- Wenn ihr die Effekte digital durchtriggert, können es auch mehr sein.
  Damit ihr nicht ins leere Schaltet, sollte hier nur die max. Anzahl der von euch gewünschten Effekt angegeben werde.

```
uint8_t NBSWLS = 4; // Number of switchabel light effects. (default: 10)
```

Hier gebt ihr die Anzahl der Arme an, die Euer Kopter hat. (4 = Quad, 6 = Hexa, 8 = Octo.)

```
uint16_t rigger = 6;· // number of rigger (default: 4)
```

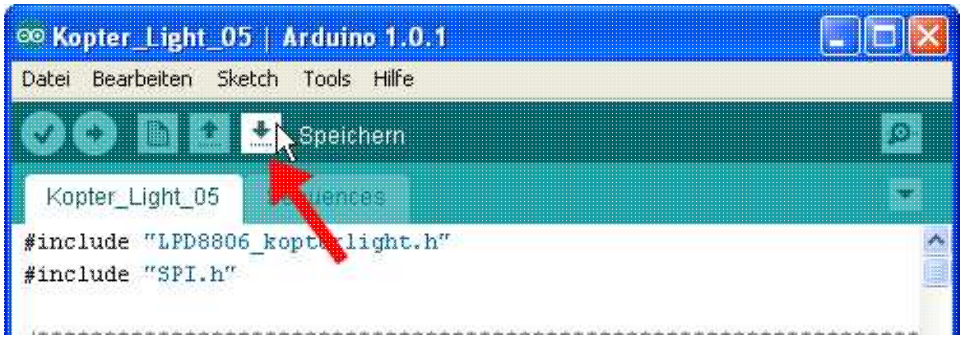
Hier gebt ihr die Anzahl der LEDs pro Arm an.

Bauart bedingt (immer 2 LEDs pro Segment) ergeben sich immer gerade Zahlen.(Bei 10 LEDs sind das 5 Segmente)

```
uint16_t riggerSize = 10;·· // Pixels / LEDs per rigger· (default: 10)
```

Mehr braucht ihr nicht zu konfigurieren.

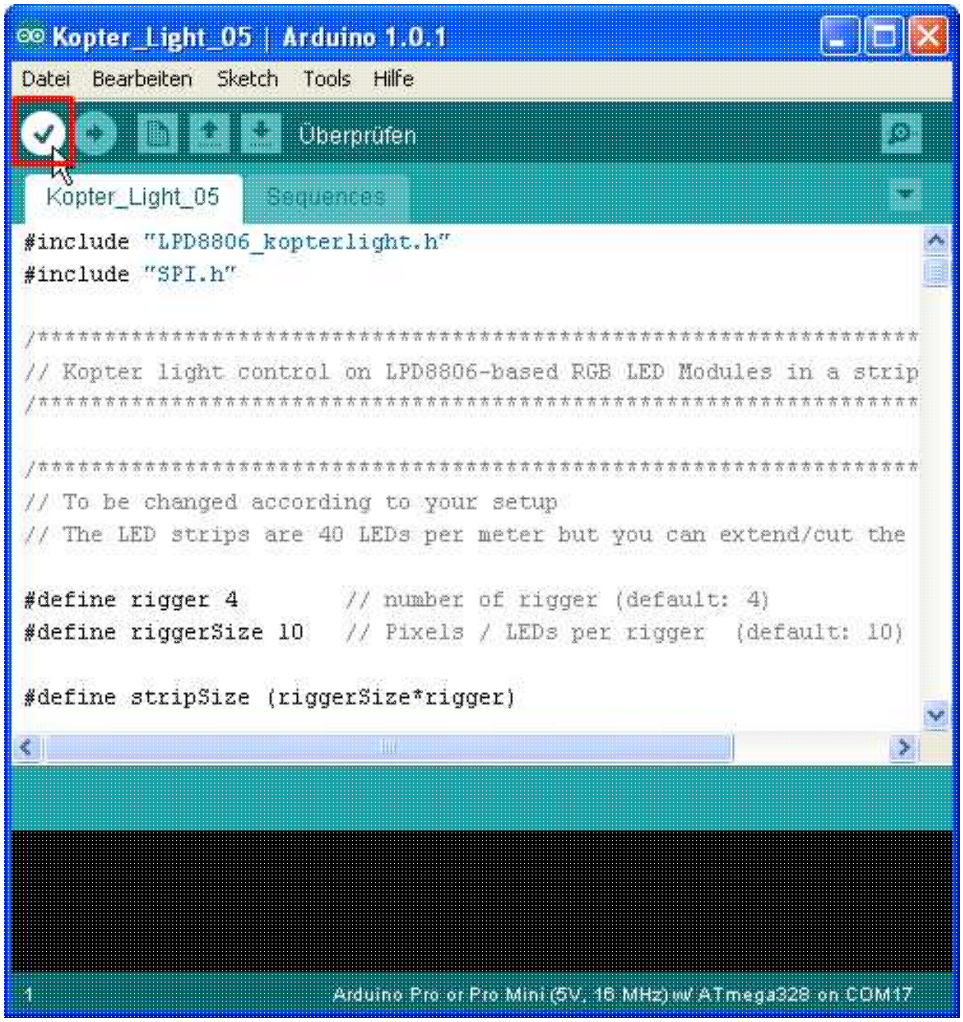
Damit wir das beim nächsten mal nicht erneut wiederholen müssen, speichern wir das Sketch, in dem wir auf den Button "Speichern" klicken:



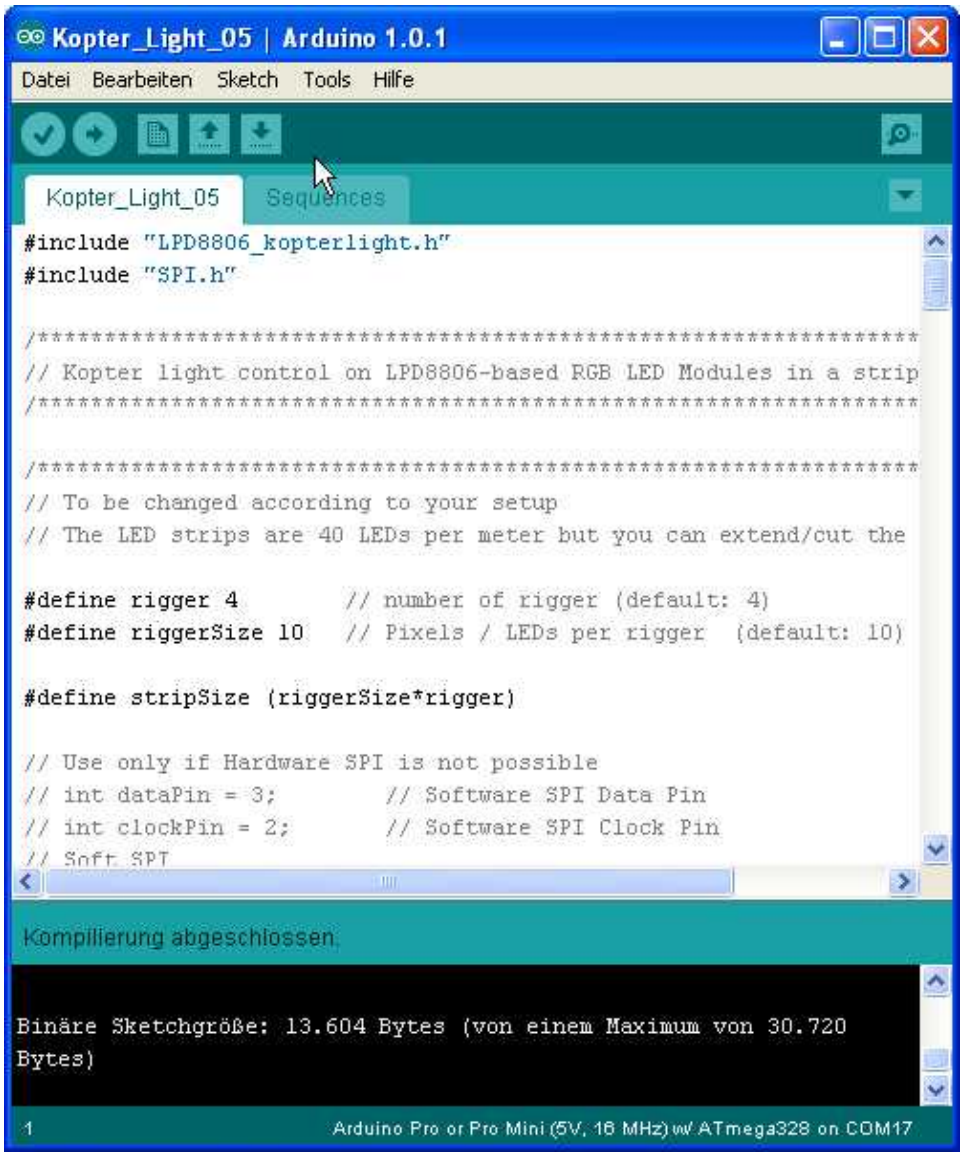
Dies sollte man nach jeder größeren Änderung machen, damit man nicht jedes mal alles neu eingeben muss. Das war es dann schon und wir können das Sketch in den Arduino laden.

3.3.7 Wie bekomme ich das Sketch in den Arduino ?:

Nun machen wir erstmal einen Codecheck, in dem wir auf Button "Überprüfen" klicken.

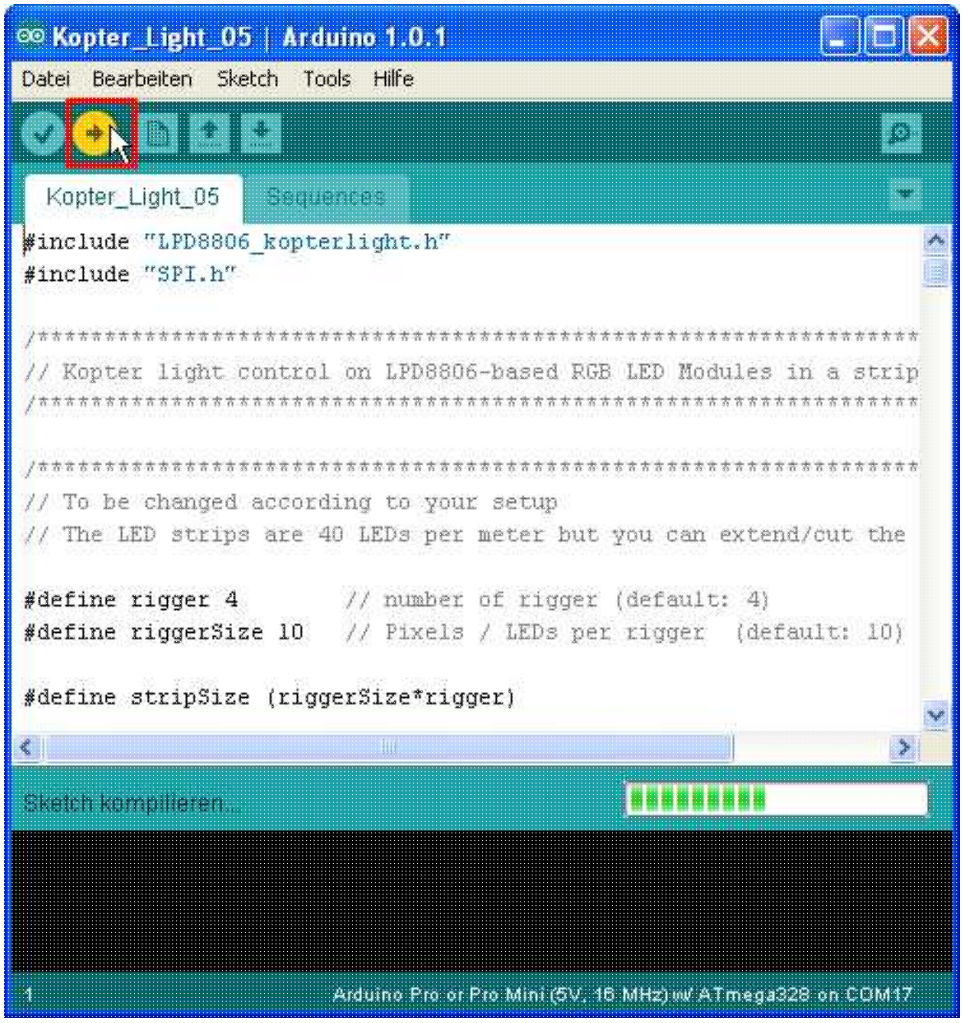

Kopter_Light_05 | Arduino 1.0.1
Datei Bearbeiten Sketch Tools Hilfe
Überprüfen
Kopter_Light_05
Sequences
#include "LPD8806_kopterlight.h"
#include "SPI.h"
/***
// Kopter light control on LPD8806-based RGB LED Modules in a strip
/***
/***
// To be changed according to your setup
// The LED strips are 40 LEDs per meter but you can extend/cut the
#define rigger 4 // number of rigger (default: 4)
#define riggerSize 10 // Pixels / LEDs per rigger (default: 10)
#define stripSize (riggerSize*rigger)
1
Arduino Pro or Pro Mini (5V, 16 MHz) w/ ATmega328 on COM17

Wenn alles OK ist erhaltet ihr folgende Meldung (Kompilierung abgeschlossen):



Der Button "Überprüfen" bietet sich auch immer an, wenn man Änderungen vorgenommen hat, ohne das man den Code in den Arduino laden muss.

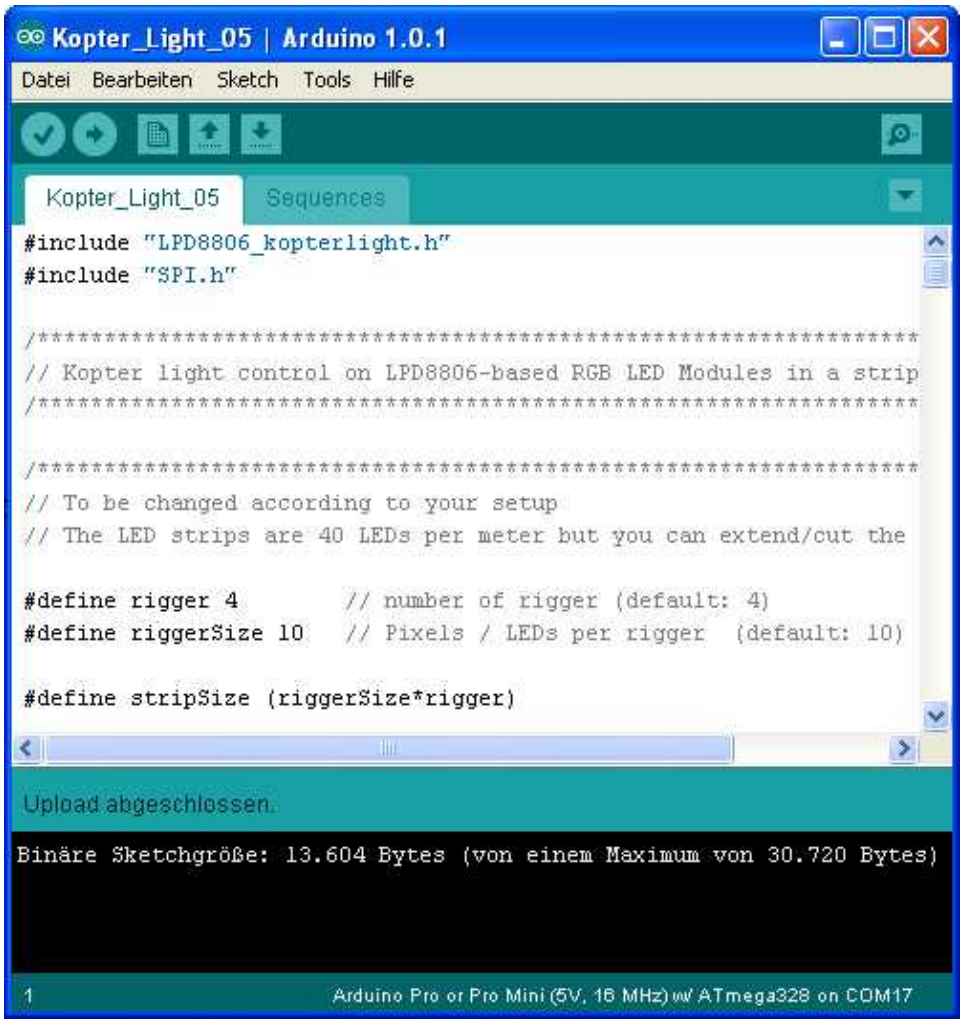
Jetzt sind wir soweit, dass wird den Code in den Arduino laden können.
Dazu klicken wir nun auf den Button "Upload" und der Code wird in den Arduino geladen.



Wenn man nun mal nebenbei auf den Arduino guckt, kann man folgendes Beobachten:

- Erst blinkt eine weisse LED.
- Danach blinken eine rot und eine grüne LED und zeigen eine Kommunikation zwischen Arduino und dem PC an.
- Während die rote und grüne LED blinken, wird nun unser Code in den Arduino geladen.

Wenn dieser Vorgang abgeschlossen ist, erhaltet ihr folgende Meldung "Upload abgeschlossen"



Danach rebootet der Arduino automatisch und unser Code wird nun abgearbeitet. D.h. nun sollten eure LEDs schön blinken.

3.3.8 Eigene Lichteffekte:

Nachdem ihr nun erfolgreich die Hardware aufgebaut habt und das erste Sketch mit den Demo-Lichtmustern zum Laufen gebracht habt, ist es nun Zeit, eigene Sequenzen zu erstellen.

Es sieht schlimmer aus als es ist, also nur Mut und viel Spaß beim Experimentieren.

 Aber folgende Dinge sind noch zu beachten:
1. Das eigentliche Arduino Programm spielt sich grundsätzlich nur in folgenden Bereichen ab:

- D.h., überall wo "**Lightshow Hier**" steht, dürft ihr eure Lightshow platzieren.

Ohne externe Ansteuerung (Lightshow_STD.ino):

```
· void loop() {¶
¶
· Lightshow Hier¶
¶
· }¶
```

Mit externer Ansteuerung (Lightshow_Remote.ino):

```
void loop() {¶
clearstrip ();¶
remoteControl();¶
··· counter2 = counter1;¶
¶
· switch (counter1){¶
··· case 1:¶
··· counter2 = counter1;¶
··· clearstrip ();¶
/************·· Your light effect:·· ************/¶
¶
···· Lightshow Hier¶
¶
/************·· ___________________·· ************/¶
··· break;¶
¶
· }¶
}¶
```

Alles andere drumherum sind nur Subroutinen und Variablendefinitionen.

2. Ausser der Änderungen für euer Setup (KopterLightEXT_x_xx.ino), solltet ihr Änderungen nur in den Hauptprogrammschleifen vornehmen (Lightshow_Std.ino/Lightshow_Remote.ino).

3. Wenn ihr euch mal mit den Parametern für die einzelnen Lichteffekte vertan habt, keine Angst, entweder meckert die Arduino IDE mit euch, es passiert nichts oder die LEDs zeigen nicht das an, was Ihr euch vorgestellt hattet. Schrotten könnt ihr die LEDs damit normalerweise nicht.

4. Die Routinen für die komplizierteren Lichteffekte sind in die Datei "Sequences.ino" ausgelagert, damit es in der Hauptprogrammschleife möglichst übersichtlich bleibt. D.h. wer kann und Lust hat, kann dort auch eigene Lichteffekte und Muster Programmieren.

5. Aus den drei Grundfarben Rot, Grün und Blau (RGB) sowie den Mischfarben aus diesen ergeben sich eure LED-Farben.

6. Einige Lichteffekt sind mit ihrer Parameterdefinition als Beispiel in der Hauptprogrammschleife vorhanden.

7. Zeilen mit "//" am Anfang sind Kommentarzeilen und müssen stehen bleiben.

8. Zeilen ohne "//" am Anfang sind Programmzeilen und müssen so stehen bleiben. (Es sei denn, man weis was man tut.)

Grundsätzlich haben wir folgende Parameter zur Verfügung, um die Lichtmuster zu beeinflussen:
(Farben sind durch das additive RGB-Farbmodell definiert.)

Parameter	**Name**	**Beschreibung**
n	LED No.	Nummer der LED die auf einem Arm gesetzt werden soll.
r	Red	Kann den Wert 0=aus bis max. Wert 127=max. hell annehmen
g	Green	Kann den Wert 0=aus bis max. Wert 127=max. hell annehmen
b	Blue	Kann den Wert 0=aus bis max. Wert 127=max. hell annehmen
dly	Delay	Verzögerung in Millisekunden (Typisch: 10-50)
cyl	Cycles	Anzahl der Wiederholungen eines Lichteffekts
riX	Rigger No.	Kann den Wert 1=ON oder 0= Off annehmen (X=Nummer des Arms (1-8) auf dem der Lichteffekt angezeigt wird.)

In allen Lichteffekte-Befehlen representieren die letzten 8 Parameter (ri1-ri8) immer die 8 Arme des Kopters. Egal, wieviel Arme am Kopter sind, es müssen immer alle 8 Parameter angegeben werden! Eine Ausnahme sind die Befehle
clearStrip und delay. Diese Befehle sind allgemein Gültig.

Die folgen Beispiele zeigen, wie sich die Befehle für die Lichteffekte zusammensetzen.

Bsp. 1:

```
// setLED(n, r, g, b, ri1, ri2, ri3, ri4, ri5, ri6, ri7, ri8)¶
setLED(1, 127, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0);¶
```

Mit diesem Befehl schalten wir LED 1 auf Arm 1 auf Rot.

Bsp. 2:

```
// police(dly,cyl,ri1, ri2, ri3, ri4, ri5, ri6, ri7, ri8);¶
police(110,3,0,1,1,0,0,0,0,0);¶
police(110,3,1,1,1,1,0,0,0,0);¶
```

Mit diesem Befehl lassen wir den Effekt "Police" mit einer Verzögerung von 110 Millsek. zwischen dem Blinken
und einer Wiederhohlung von 3 mal auf den Armen 2 und 3 und danach auf den Armen 1-4 jeweils gleichtzeitig laufen.

Bsp. 3:

```
// colorChase(r,g,b,dly,ri1, ri2, ri3, ri4, ri5, ri6, ri7, ri8)¶
colorChase(127,127,127,80,1,0,0,1,0,0,0,0); // white¶
```

Mit diesem Befehl lassen wir einen weißen Lichtpunkt auf den Armen 1 und 4 gleichzeitig laufen, mit einer Verzögerung vom LED zu
LED von 80 Millsek.

3.3.9 Befehls-Übersicht:

Falls einige Befehle unklar sind, schaut euch die Beispiele in den Lightshow-Dateien an oder fragt im Forum.
Jeder Befehl muss mit einem Semikolon abgeschlossen werden.

3.3.9.1 Steuer-Befehle

Befehl	**Beschreibung**
clearstrip ();	Schaltet alle LEDs aus
setLED(n, r, g, b, ri1, ri2, ri3, ri4, ri5, ri6, ri7, ri8);	Schaltet einzelne LEDs auf einzelnen Armen
strip.show();	Bringt alle mit clearstrip () oder setLED() ausgeführten Befehle zur Anzeige.
delay(dly);	Verzögerung zwischen Lichteffekten: dly = Wert in Millsek.
updateStripConfig (Anzahl Arme, LEDs	Hiermit kann während des Programmablaufs die Anzahl der

pro Arm);	Arme und LEDs pro Arm geändert werden. ⚠ Es muss aber immer die Gesammtanzahl der LEDs gleich bleiben. Bsp.: 6 Arme/je 10 LEDs oder 3 Arme/je 20 LEDs....

3.3.9.2 Licht-Effekte

Befehl	**Beschreibung**
fadein(r, g, b, dly, ri1, ri2, ri3, ri4, ri5, ri6, ri7, ri8);	Blendet eine Farbe auf einem gesammten Arm ein
fadeout(r, g, b, dly, ri1, ri2, ri3, ri4, ri5, ri6, ri7, ri8);	Blendet eine Farbe auf einem gesammten Arm aus
flashLight(dly,cyl,ri1, ri2, ri3, ri4, ri5, ri6, ri7, ri8);	einfarbiges Blinklicht
police(dly,cyl,ri1, ri2, ri3, ri4, ri5, ri6, ri7, ri8);	zweifarbiges Blinklicht (Jeweils eine Farbe auf der linken und die Andere auf der rechten Hälfte des Arms)
scanner(r,g,b,dly,ri1, ri2, ri3, ri4, ri5, ri6, ri7, ri8);	Knight Rider Effekt
circlinglights(dly,cyl,ri1, ri2, ri3, ri4, ri5, ri6, ri7, ri8);	Zwei Farben die auf einem Arm entgegengesetzt laufen
colorChase(r,g,b,dly,ri1, ri2, ri3, ri4, ri5, ri6, ri7, ri8);	Ein Lichtpunkt läuft vorwärts den Arm entlang
colorChaseRev(r,g,b,dly,ri1, ri2, ri3, ri4, ri5, ri6, ri7, ri8);	Wie "colorChase" nur rückwärts
colorWipe(r,g,b,dly,ri1, ri2, ri3, ri4, ri5, ri6, ri7, ri8);	Eine LED nach der anderen wird auf dem Arm eingeschaltet
colorWipeREV(r,g,b,dly,ri1, ri2, ri3, ri4, ri5, ri6, ri7, ri8);	Wie "colorWipe" nur rückwärts
colorFill(r,g,b,ri1, ri2, ri3, ri4, ri5, ri6, ri7, ri8);	Alle LEDs werden auf dem Arm gleichzeitig eingeschaltet
dither(r,g,b,dly,ri1, ri2, ri3, ri4, ri5, ri6, ri7, ri8);	Experimental !
wave(r,g,b,dly, stp, ri1, ri2, ri3, ri4, ri5, ri6, ri7, ri8);	Wellenartiger Lichteffekt
rainbowCycle(dly, cyl, ri1, ri2, ri3, ri4, ri5, ri6, ri7, ri8);	Zirkulierender Regenbogen Effekt
rainbowCycleAll(dly, cyl, ri1, ri2, ri3, ri4, ri5, ri6, ri7, ri8);	Zirkulierender Regenbogen Effekt auf allen LEDs gleichzeitig
spiral(r, g, b, dly);	Spiral Effekt von innen nach aussen
spiralRev(r, g, b, dly);	Spiral Effekt von aussen nach innen
flashingCircle(r, g, b, dly);	Blinkender Kreis von innen nach aussen
flashingCircleRev(r, g, b, dly);	Blinkender Kreis von aussen nach innen

⚠ ToDo: Hier sollen zukünftig noch mehr Lichteffekte hinzugefügt werden !!
Wer Lust hat, seine eigenen Lichteffekte zu programmieren und zu veröffentlichen, ist hiermit herzlichst eingeladen
sich zu beteiligen.

Die Lichteffekte tragt ihr in das Template ein und speichert es dann in einem Ordner mit eurem **Forumsnamen** . Diesen erstellt ihr im **Sequences** Ordner unter **Branches** .
Den Ordner findet ihr unter: **Subversions Projekte: Projects/Digital_RGB_LED_Stripes**

Sequences-Ordner
Template.txt

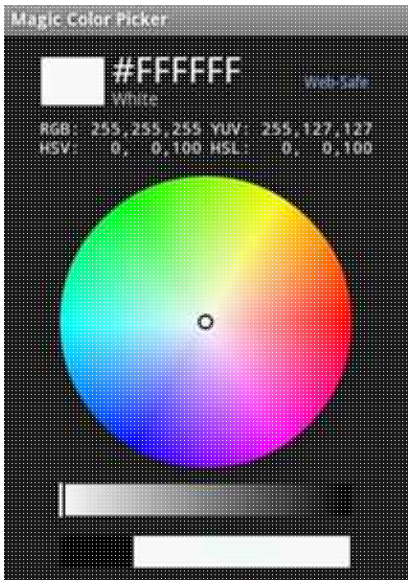
3.4 RGB Farben

Um sich die Farbwahl zu erleichtern und nicht so viel mit den RGB-Farbwerten rumschlagen zu müssen, kann man folgende zwei Tools zur Hilfe nehmen.

⚠ Beachtet aber bitte, da diese Tools nicht speziell für RGB-LED-Streifen geschrieben wurden, sondern für andere Systeme, gehen die Werte bis 255.
Wenn man aber die Werte durch 2 teilt, passt das ganz gut zu den RGB-LED-Streifen.

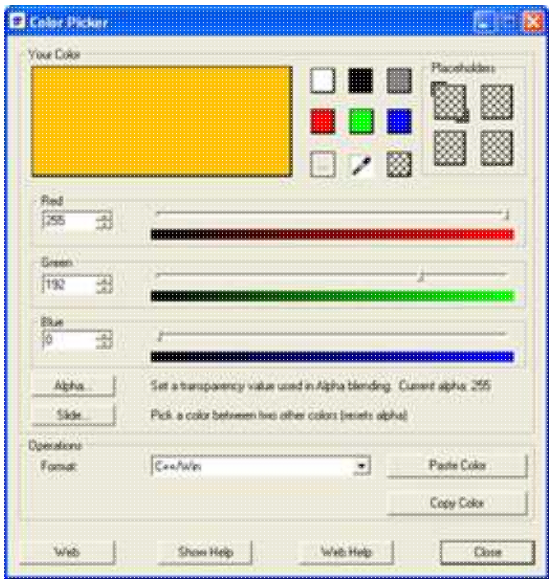
3.4.1 Android:

Magic Color Picker aus dem Google Play Store:



3.4.2 Windows:

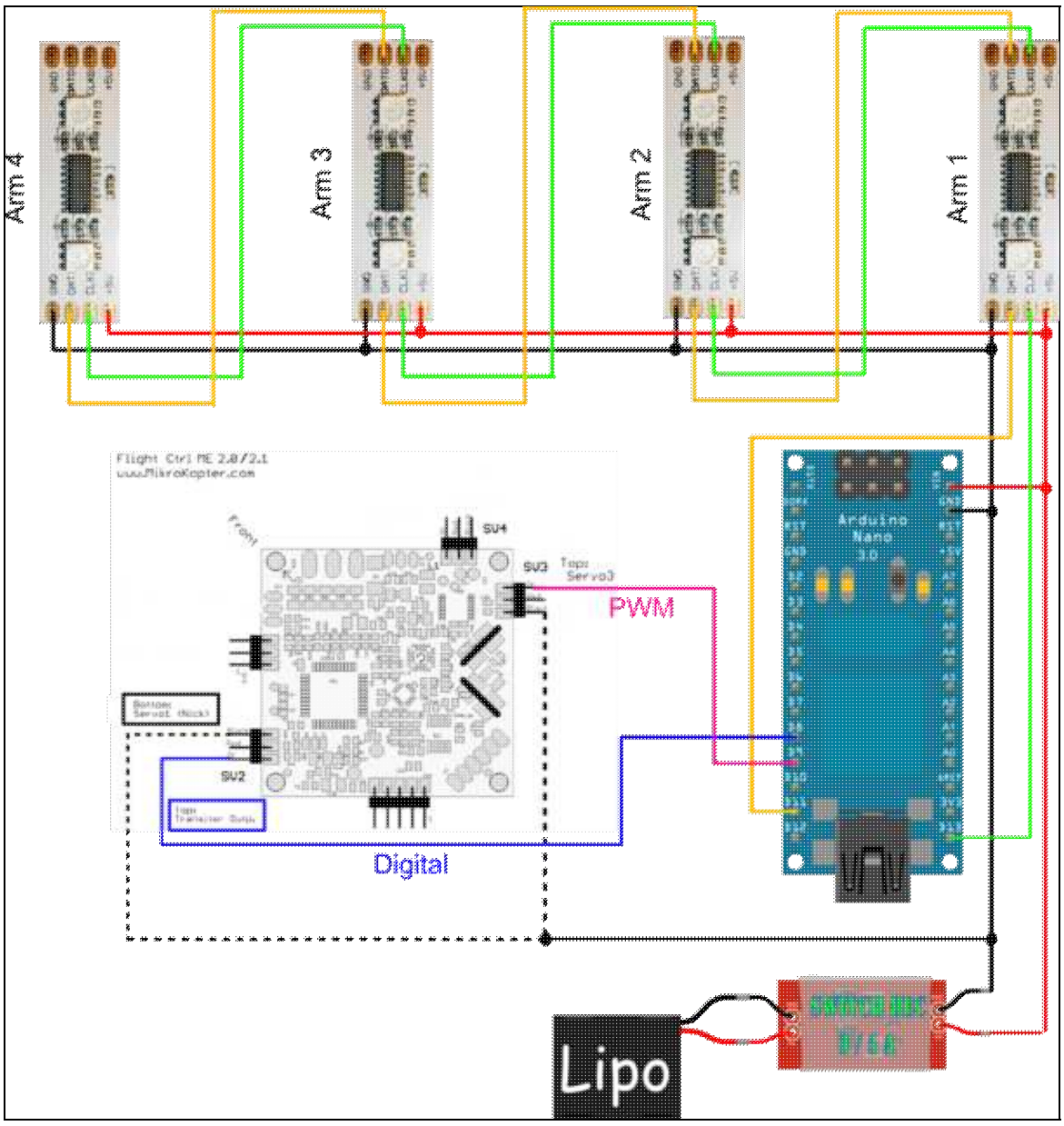
Color Picker (von Granite Tower Software):



Download

3.5 Externe Ansteuerung

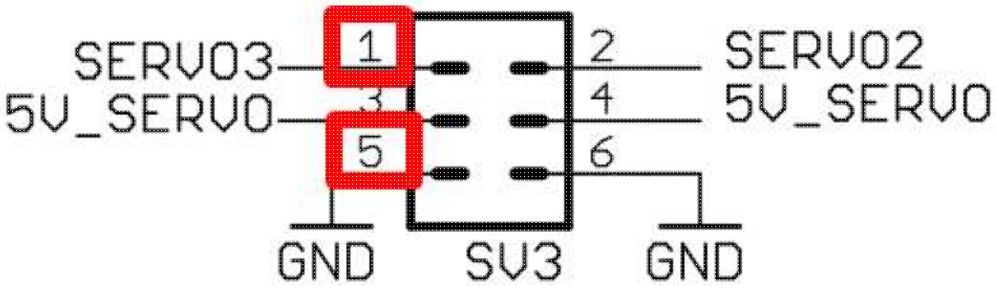
Um die externe Ansteuerung mit PWM oder digitaler Triggerung zu realisieren, muß nun der Arduino mit der FC verbunden werden.
⚠ **WICHTIG: Für die Verbindung dürfen jeweils nur die Signalleitung und Minus (GND) verbunden werden !**
Es darf auf keinen Fall +5V von FC mit dem Arduino verbunden werden, da solange noch kein externe Spannungsversorgung
am Kopter Light angeschlossen ist, diese dann über die FC mit +5V versorgt werden würde und die FC beschädigen kann!


Arm 4
Arm 3
Arm 2
Arm 1
Flight Ctrl ME 2.0/2.1
www.MikroKopter.com
front
SV4
SV3
Servo3
SV2
PWM
Digital
Arduino
Nano
3.0
Lipo

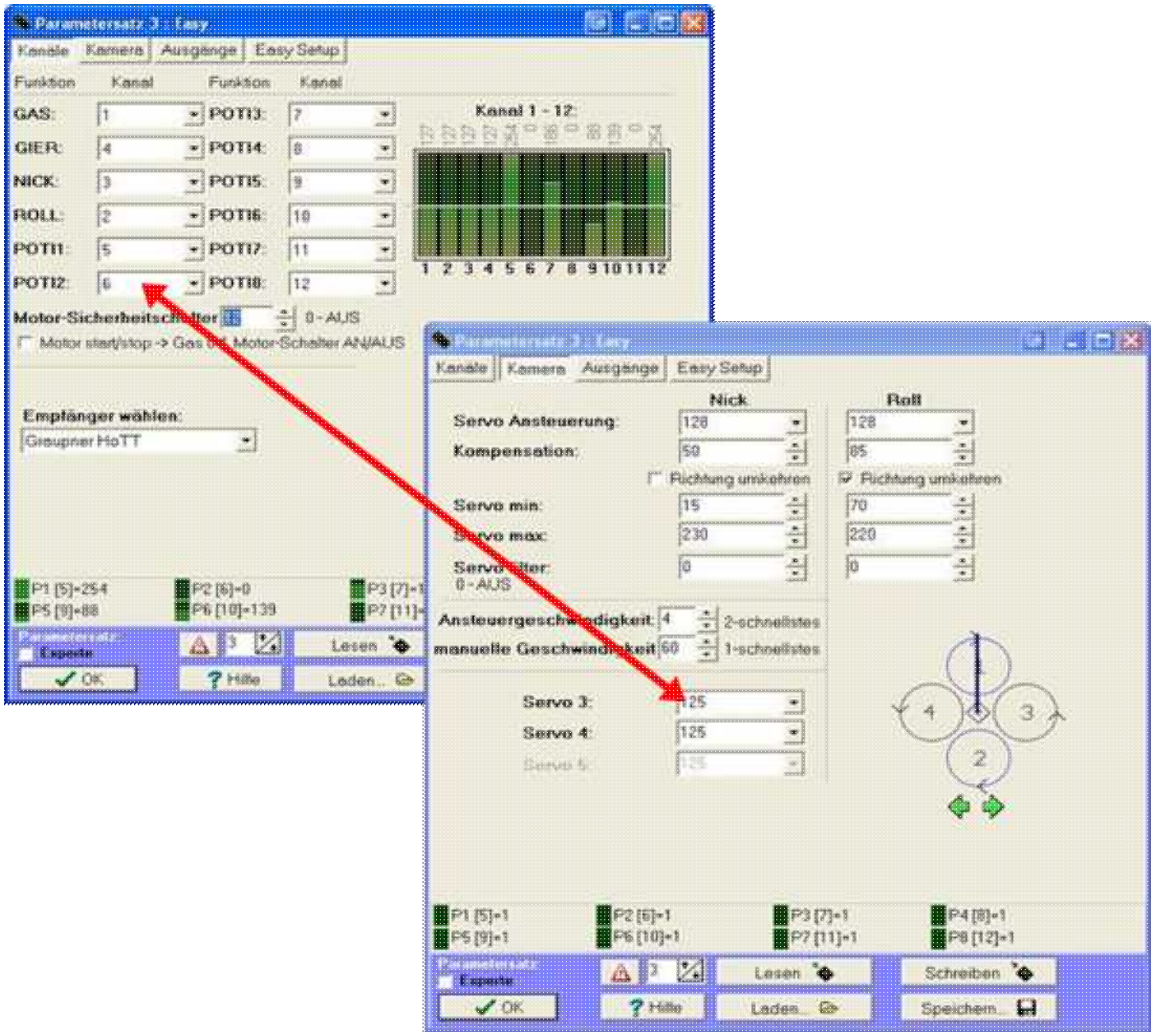
3.5.1 Schalten über PWM (Drehpoti)

Zum Schalten per Drehpoti, verbinden wir:

Arduino Pin	FC Servo 3 Pin
D9	1 (PWM Signal)
GND	5 (GND)



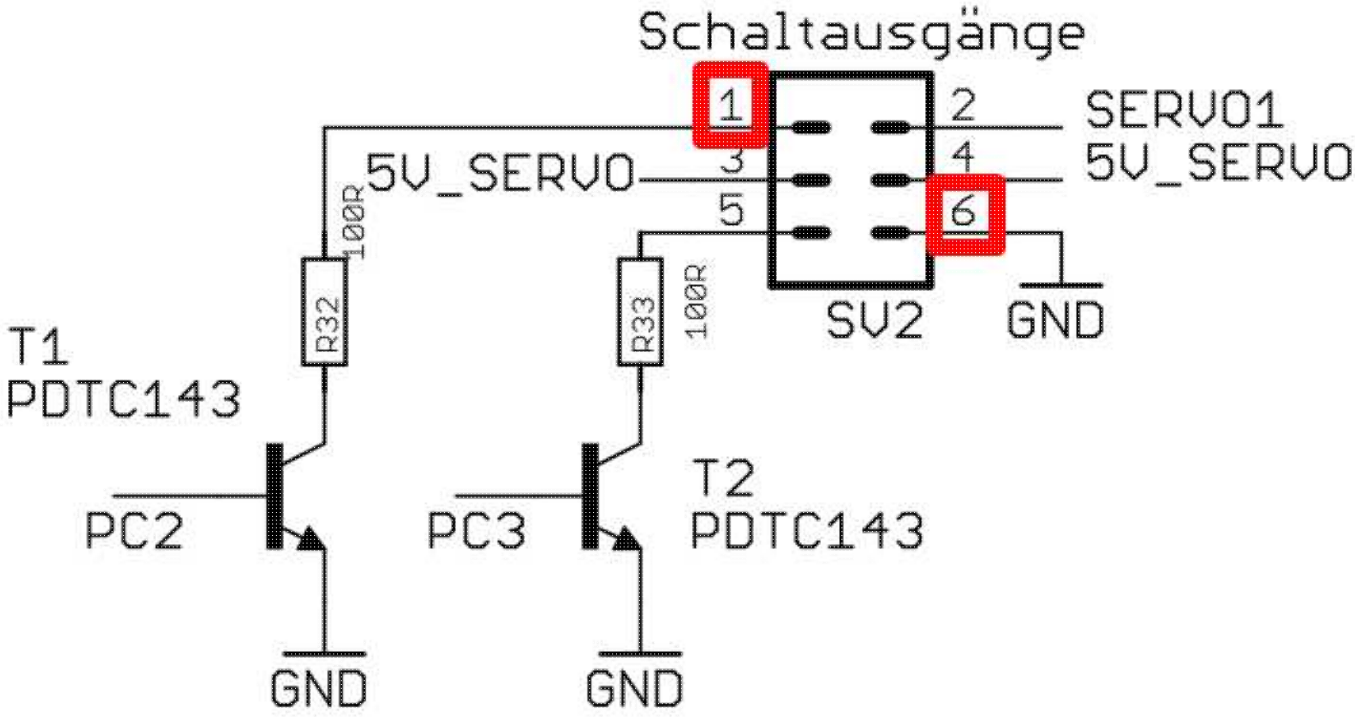
Dann suchen wir uns einen freien Kanal am Sender und weisen diesem ein Drehpoti / Schieberegler zu. Im Kopter Tool legen wir nun den Kanal auf ein Poti und tragen anschliesend das Poti bei Servo 3 ein.



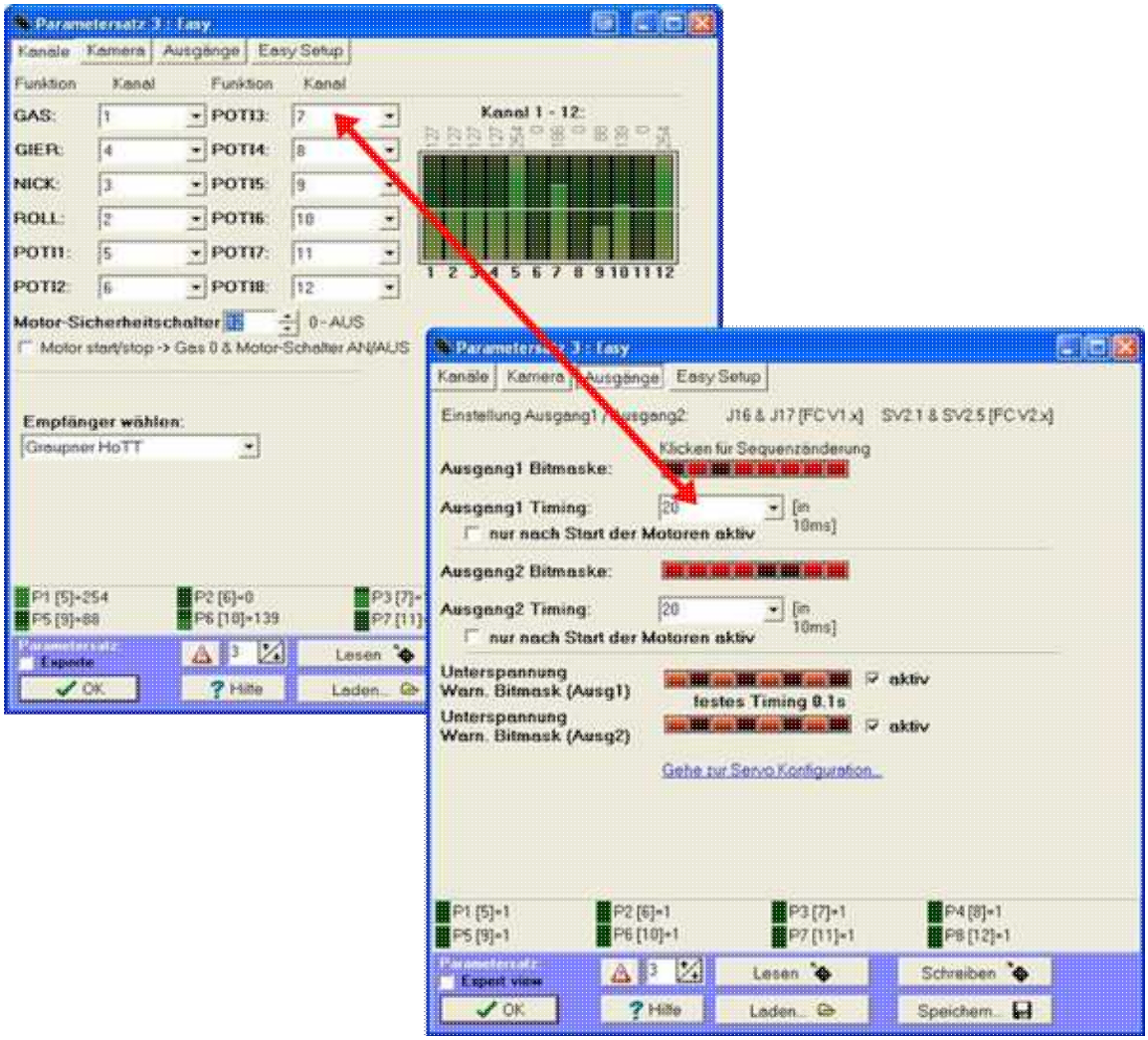
3.5.2 Schalten über Transistor Output J16 (Taster)

Zum Schalten per Taster, verbinden wir:

Arduino Pin	FC Output J16	FC Servo 1 Pin
D8	1 (Trigger Signal)	-
GND	-	6 (GND)



Dann suchen wir uns einen freien Kanal am Sender und weisen diesem einen Taster zu.
Im Kopter Tool legen wir nun den Kanal auf ein Poti und tragen anschliesend das Poti bei Ausgang 1 Timing ein.


Kanäle
Kamera
Ausgänge
Easy Setup
Funktion
Kanal
GAS:
GIER:
NICK:
ROLL:
POTI1:
POTI2:
POTI3:
POTI4:
POTI5:
POTI6:
POTI7:
POTI8:
Kanal 1 - 12:
Motor-Sicherheitschalter
0 - AUS
Motor start/stop -> Gas 0 & Motor-Schalter AN/AUS
Empfänger wählen:
Graupner HoTT
Lesen
Schreiben
Laden...
Speichern...
OK
Hilfe
Einstellung Ausgang1 / Ausgang2:
J16 & J17 [FC V1.x]
SV2.1 & SV2.5 [FC V2.x]
Klicken für Sequenzänderung
Ausgang1 Bitmaske:
Ausgang1 Timing:
nur nach Start der Motoren aktiv
Ausgang2 Bitmaske:
Ausgang2 Timing:
Unterspannung Warn. Bitmask (Ausg1)
Unterspannung Warn. Bitmask (Ausg2)
aktiv
festes Timing 0.1s
Gehe zur Servo Konfiguration...
Expert view

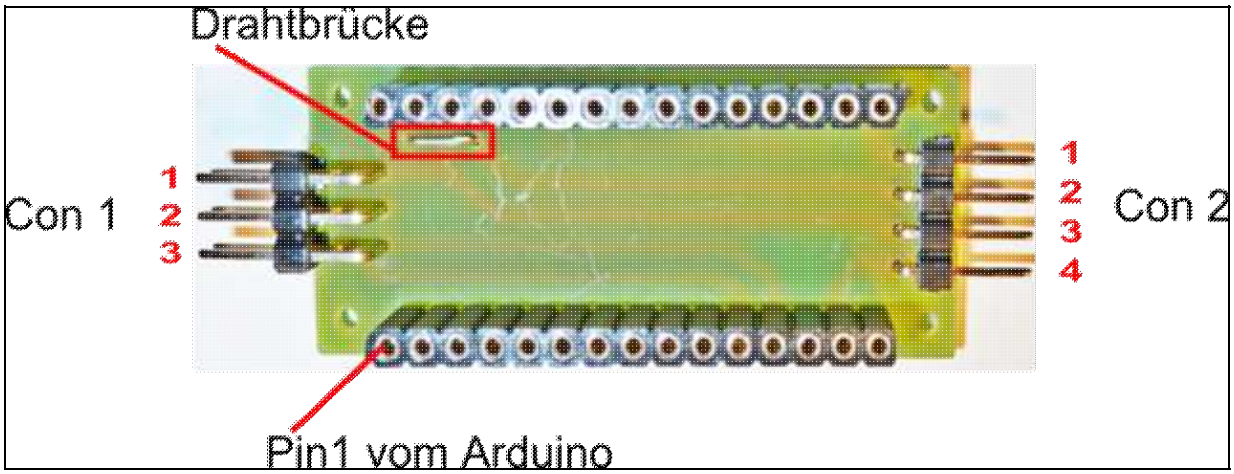
3.5.3 Kopter Light Shield

Nackte Platine:

Bestückung der Platine:
Als erstes löten wir die Drahtbrücke ein.
Die 36-Polige Buchsenleiste teilen wir in 2x 15 Pins und 1x 4 Pins und löten die 2 x 15 Pins in die Platine.
1x 4 Pins dienen als Stecker für **Con2**.
Danach löten wir die Stiftleisten ein.

Und schon sind wir fertig.

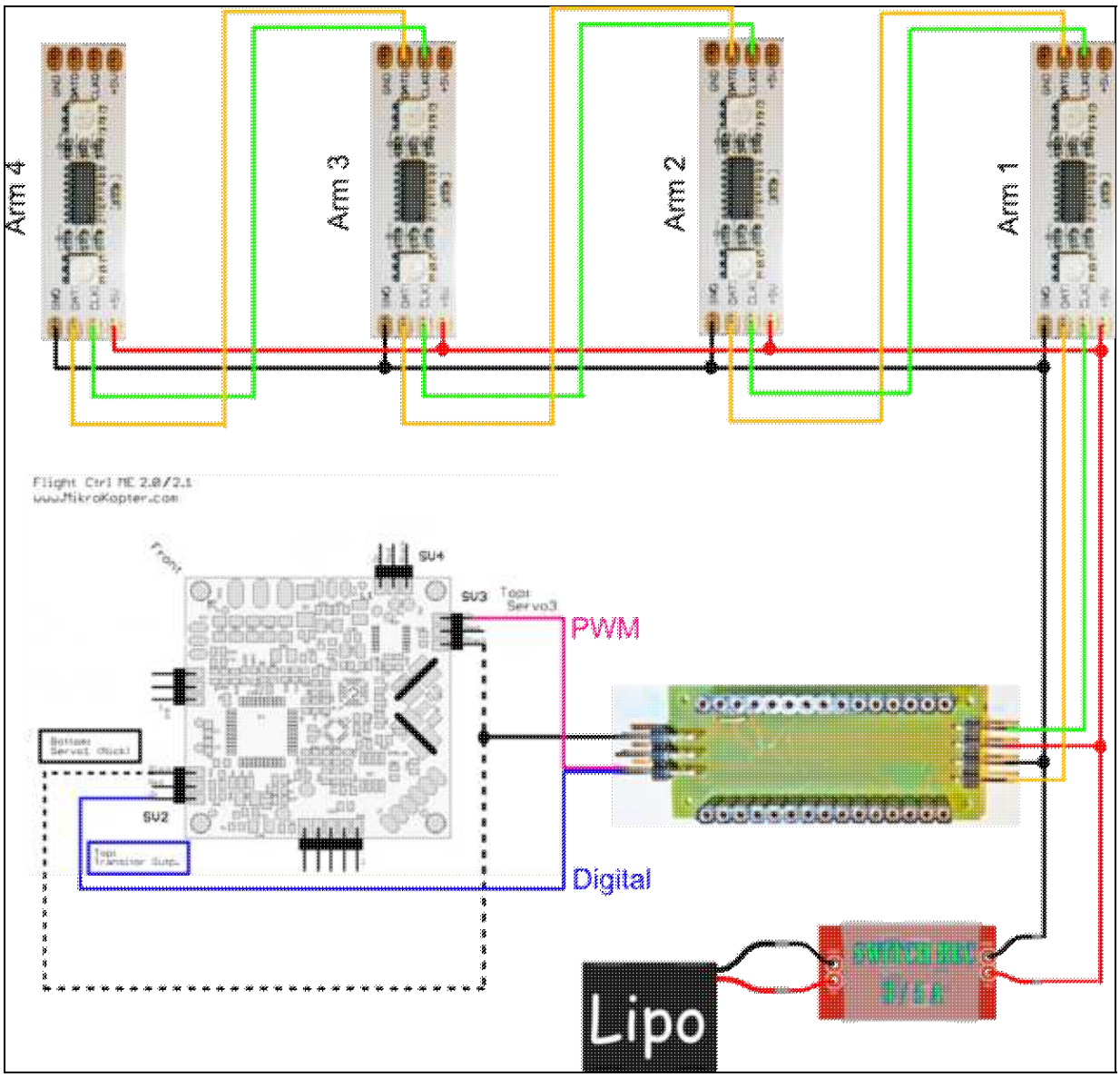
Für **Con1** nehmen normale Servostecker.



Beschaltung:

Con1 (Anschluss zur FC)	**1**	**2**	**3**
Oben	GND	N.C.	PWM
Unten	GND	N.C.	Dig. Triger J16

Con2 (Anschluss zum DRGB-Streifen und Spannungsversorgung)	**1**	**2**	**3**	**4**
	CL (Clock)	+5V	GND	DAT (Data)


Arm 4
Arm 3
Arm 2
Arm 1
Flight Ctrl ME 2.0/2.1
www.MikroKopter.com
Front
SV4
SV3
Top
Servo3
PWM
Bottom
Servo (Nick)
SV2
Top
Transistor Supp.
Digital
Lipo

Und so sieht das ganze fertig aus:

Oberseite (Servostecker steck auf dem PWM-Eingang):

Unterseite (Servostecker steck auf dem PWM-Eingang):

Fertiges Shield mit Arduino:

3.6 Tips und Tricks

- Wenn man die Anzahl der Arme auf 1 setzt und die gesammt Anzahl der LEDs angibt, kann man die Software auch für "einen" langen Streifen nutzen.

3.7 Videos

3.7.1 Erster Testaufbau:

3.7.2 Test mit 1m:

3.7.3 Test mit Einzelarmansteuerung (Kopter Light V1):

3.7.4 Erster Test am Kopter von XKnut (Kopter Light V1):

3.7.5 Test mit Multiarmansteuerung (Kopter Light V2):

3.7.6 Test mit externer Ansteuerung (Kopter Light V2):

3.8 Links

Thread: Digitale-RGB-Strip`s

Netzhautbrenner: Weitere Infos zu LED Streifen

- KategorieProjekte KategorieTools