

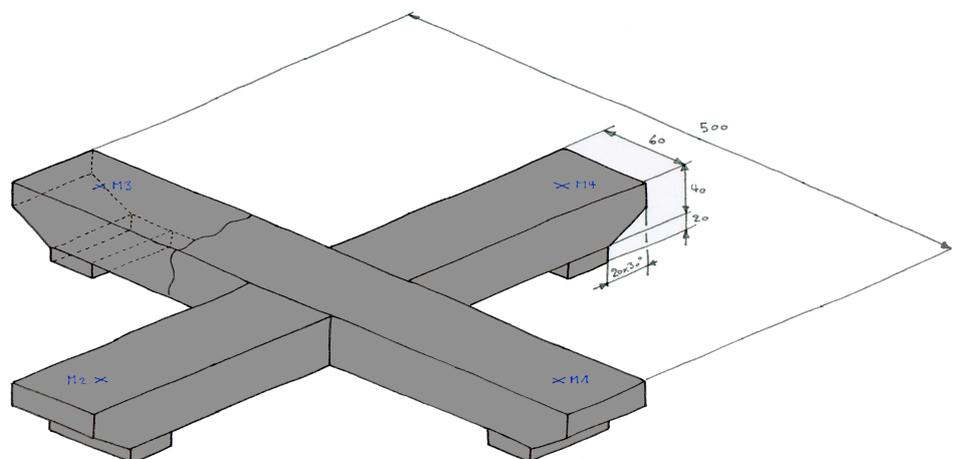


5. Semester | 2010

QUADROCOPTER

FASZINATION FLIEGEN

MECHATRONIC TRINATONAL
PROMOTION PLANCK
MECHATRONIKLABOR



Christan Vögtli und Stebler Michael

FASZINATION FLIEGEN, SUMMARY

Während den letzten Wochen haben wir versucht, einen flugfähigen Quadrocopter zu bauen. Doch was bedeutet das Wort flugfähig, wie soll es interpretiert werden?

In unserem Fall entschieden wir uns, einen Quadrocopter zu bauen, welcher mit einer Fernsteuerung geflogen werden kann. Dabei mussten wir auf einige technische Hilfsmittel, welche mit Sicherheit das Fliegen erleichtern würden, verzichten. Aus Zeittechnischen Gründen war es uns zum Beispiel nicht möglich, eine Stabilisation mittels einem Giro-Sensor zu erzielen.

Wie wir aus etlichen Projekten gelernt haben, ist es nicht nur wichtig über sich selber Kenntnisse zu haben, sondern auch von der „Konkurrenz“ zu lernen. So haben wir uns zu Beginn des Projektes Quadrocopter mit genau dieser Konkurrenz auseinander gesetzt und uns im Gebiet der Quadrocopter Technik somit fachlich Weitergebildet.

Von der Bedeutung des Namens, bis hin zu den Einsatzgebieten haben wir Informationen gesammelt um uns ein Allgemeinwissen, welches nicht technisch begründet ist, zu schaffen.

Nach dem erarbeiten der Grundkenntnisse standen die ersten Gruppenaktivitäten auf dem Plan. Gruppenintern wurden Ziele, sowie der zeitliche Ablauf zu Erreichung der Ziele definiert. Während der Umsetzungsphase wurden in mehreren Meetings unsere Arbeitsstände und das weitere Vorgehen besprochen.

Wie bei jedem Projekt sind auch wir auf einige Probleme gestossen, welche jedoch meist ohne externe Hilfe bewältigt werden konnten und somit von uns als Chance für neue Schritte gesehen werden.

Die erzielten Ergebnisse sind mit den gesetzten Zielen übereinstimmend. Die Anforderungen konnten erfüllt werden und in einigen Punkten haben wir diese sogar übertroffen. So können wir am Ende unseres Projektes von einem erfolgreichen Abschluss reden und mit einem lachenden und weinenden Auge zurückschauen, auf eine angenehme Zeit während einem faszinierenden Projekt.

Basel, 24. November 2010

Christan Vögtli

Michael Stebler

INHALTSVERZEICHNIS

1. AUFGABENSTELLUNG	4
1.1 BESCHRIEB.....	4
1.2 ZIEL.....	4
1.3 VORGEHENSVORSCHLAG	5
2. ZIELSETZUNG.....	6
3. PROJEKTABLAUFPLAN	7
4. KONSTRUKTION	8
4.1 DESIGN	8
4.2 KONSTRUKTION	8
4.3 MATERIALAUSWAHL.....	9
4.4 KLEBSTOFFWAHL	10
4.5 FLUGHILFE.....	10
5. HARDWARE	11
5.1 AT90USB1287	11
5.2 SENDER UND EMPFÄNGER FÜR FERNSTEUERUNG	11
5.3 MOTOREN.....	14
5.4 MOTORENTRIEBER.....	14
5.5 AKKU.....	14
5.6 RGB-BELEUCHTUNG	15
5.7 AKKUSPANNUNGSÜBERWACHUNG.....	15
6. SOFTWARE	17
6.1 PROGRAMMIERUNG	17
6.2 PPM SIGNAL-INTERPRETIERUNG	17
6.3 FUNKTIONSZUWEISUNG	18
6.4 MIXER	18
6.5 SIGNALAUSGABE.....	19
7. PROBLEMSTELLUNGEN	21
7.1 TREIBERINSTALLATION PLATINE	21
7.2 PARALLELE SIGNALVERARBEITUNG UND AUSGABE	21
7.3 PLATZMANGEL	22
7.4 KLEBSTOFF	22
8. REFLEXION	23
9. ABBILDUNGSVERZEICHNIS.....	24
10. ANHANG.....	25



1. AUFGABENSTELLUNG

1.1 BESCHRIEB

Aus Wikipedia:“ Ein Quadrocopter ist ein Luftfahrzeug, das vier in einer Ebene angeordnete, senkrecht nach unten wirkende Rotoren oder Propeller benutzt, um Auftrieb und durch Neigung der Rotorebene auch Vortrieb zu erzeugen. Es gehört zu den Hubschraubern und kann wie diese oder auch wie VTOL-Flugzeuge senkrecht starten und landen. Andere Bezeichnungen sind Quadrotor oder Schwebplattform.“

Ein solches Luftfahrzeug soll gebaut werden. Das Herz ist entweder ein NI Single-Board RIO oder ähnlich und die Programmiersprache Labview oder ein ARM-Board und Programmiersprache AVR-C.

Allg.:

Modell von NI: www.ni.com/info unter Info Code: nsi0108.

Modell von T. Pfeifer: <http://thomaspfeifer.net/quadrocopter-selbstbau-avr.htm>.

1.2 ZIEL

Das Quadcopter von NI soll gemäss dem angegebenen Link nachgebaut werden. Der Artikel dient als Basis für die Entwicklung mit einer Variante bei der Wahl der Rechneinheit, NI-Singleboard oder ARM-Board. Weil der Bau eines flugfähigen Gerätes sehr aufwendig ist und noch keine Vorarbeit besteht wird man sich auf eine Einstiegsphase in zwei Schritten beschränken. Im ersten Schritt wird die Ansteuerung eines einzigen Motors realisiert. Er wird auf eine Platte montiert, die auf einer Seite befestigt ist und die andere Seite frei ist. Der Motor mit Propeller soll die Platte anheben können ähnlich wie bei der Video „Propeller Crash“ auf der thomaspfeifer-Homepage. Im zweiten Schritt ist das Gerüst des Quadcopters fertig mit zwei Motoren und entsprechender Elektronik. Die Einheit ist auf einer Achse auf einem Teststand montiert, so dass sie sich nur um die Rotationsachse drehen kann. Auf der anderen freien Achse sind zwei Motoren befestigt. Die Motoren sind so angesteuert, dass sie den Quadcopter auf einer horizontalen Lage stabilisieren können ähnlich dem Video „2-Motor, single axis test“ auf der thomaspfeifer-Homepage.

Aufgaben von Schritt 1:

- Einarbeitung in Thematik der brushless-Motoren, deren Ansteuerung, Propeller und deren Auftrieb.
- Wahl der Komponenten. Realisierung der Motoransteuerung, um verschiedene Drehgeschwindigkeiten zu erhalten.
- Erstellung des Testaufbaus mit Einstellung der Motordrehzahl vom Bediener

Aufgaben von Schritt 2:

- Einarbeitung in die Thematik Stabilisierung eines Quadcopters. Bestimmung der Sensoren.



QUADROCOPTER

- Test der Sensoren mit entsprechender Software
- Erstellung der Programmstruktur und des Programms zur Stabilisierung.
- Bau eines Teststands und des Quadcopter-Gerüsts. Ergänzung mit Sensoren und Steuerelektronik.ßß
- Inbetriebnahme mit Bedieneroberfläche.

1.3 VORGEHENSVORSCHLAG

1. Projektplanung, Teamaufteilung besprechen, Vorgehen durchlesen und diskutieren. Anschliessend einen Zeitplan aufstellen mit Zielen, welche erreicht werden sollen (Meilensteine).
2. Konzept aufstellen, welche physikalischen Vorgänge sollen gemessen werden (verschiedene Winkel, Geschwindigkeiten, Beschleunigungen, ...). Welche Sensoren brauchen wir dazu? Wie werden diese ausgelesen und welche Speisung brauchen sie (aktiv, passiv)?
3. Bestellliste erstellen und mit Betreuern besprechen.
4. Sensoren und Aktoren „trocken“ testen z.B. mit Oszilloskop, anschliessend mit DAQ auslesen.
5. In die Thematik LabView oder AVR-Software und Schnittstellen einarbeiten. Ein einfaches Programm schreiben, um Sensorsignale einzulesen und Aktoren anzusteuern.
6. Durchführung von Schritt 1.
Durchführung von Schritt 2.



2. ZIELSETZUNG

Am Anfang eines jeden Projektes steht die Zielfindung. Unter Einbezug der vorgegebenen Ziele wurde im Team eine eigene Zieldefinition erstellt. Diese wurden nach den Prioritäten gewichtet und dient als Grundlage des Projektes. Die drei Gruppen Konstruktion, Hardware und Software sollten dabei eine Einteilung geben, welche Aspekte des Projektes für die Erreichung des Zieles notwendig sind.

Zielklasse	Zielformulierung	Priorität
Mechanische Konstruktion		
• Statik	Genügend stabil ausgelegt	Muss
• Gewicht	Das Fluggerät soll leichter 1.2 kg sein	Muss
• Material	Leicht bearbeitbar	Soll
• Design	Ansprechendes Design für einen Quadrocopter	Soll
Hardware		
• Entwicklungsboard	Kennenlernen und beschreiben	Muss
• Fernsteuerung	Einbinden an Board und Testlauf ohne Regelung (2 Motoren)	Muss
• Motoren	Kennenlernen und einbinden	Muss
• Gyro Sensor	Leistungsanalyse	Soll
Software		
• Signalerkennung	Auslesen der Fernsteuerungssignale (PPM-Signal)	Muss
• Signalverarbeitung	Einlesen der Signale + einfache Ausgabe	Muss
• Mixer	Erstellen des Mixerschemas sowie die Umsetzung in der Software, Probelauf mit Regelung (2 Motoren)	Muss
• Mixer	Probelauf mit Regelung (4 Motoren)	Soll
Allgemein		
• Kosten	Kosten ohne vorgegebene Hardware < 100.-	Soll

Mit dem Erreichen der Muss Ziele sollte eine annähernd flugtauglicher Quadrocopter erstellt werden, welcher wenn möglich von uns mittels Hilfsmittel geflogen werden kann. Wir sind uns jedoch bewusst, dass es für das Fliegen des Quadrocopter sehr erfahrene Piloten braucht, was wir leider nicht sind. Dennoch werden wir unser Bestes geben, um der Faszination Fliegen möglichst nahe zu kommen.



3. PROJEKTABLAUFPLAN

Arbeitstag	Morgen	Mittag	Abend	Zusatzaufgaben/ Weiterführende Aufgaben	Visum
1.	Einarbeitung, Problemstellung	Zielsetzung	Aufgabeneinteilung	Dokumentation	CV MS
2.	Design PWM Eingang Fernsteuerung	Werkstoffwahl Ansteuerung von Schrittmotoren	Fertigungsauftrag Ansteuerung von Schrittmotoren	UART Sensoren analysieren Probelauf ohne Mixerregelung mit Fernsteuerung	CV MS
3.	Grundgerüstbau Mixerverbesserungen	Hardware erstellung Software Sicherheit	Hardware erstellung Software Sicherheit	Messungen / Testläufe Messungen / Testläufe	CV MS
4.	Zusammenbau Software Sicherheit	Zusammenbau Testgerät Entwickeln	Probelauf mit Mixerregelung und Fernsteuerung Dokumentation	Dokumentation beenden Präsentationskonzept	CV MS Krank
5.	Präsentation	Abgabe Dokumentation		Vermitteln von Fachwissen	CV MS



4. KONSTRUKTION

Bei der Konstruktion des Quadrocopter haben wir uns persönlich 4 Ziele gesetzt, welche erreicht werden sollten.

4.1 DESIGN

Wie allgemein bekannt ist, ist das Design eines Quadrocopter durch mehrere Einflüsse (4 Propeller usw.) mehrheitlich gegeben. Dennoch haben wir versucht, unserem Quadrocopter eine gewisse Wiedererkennbarkeit zu geben.

Anhand von bestehenden Quadrocopter wurden im Team mehrere Designvarianten diskutiert. Es wurden Argumentationen gesucht und anhand dieser wurde ein Entscheid für ein Design gefällt. Dabei hat mit Sicherheit die Realisierbarkeit den grössten Einfluss auf den Entscheid genommen.



Abb. 1 Diverse Quadrocopter Varianten

Wir haben uns für die Kreuzvariante entschieden. Diese ist mit dem geringsten Aufwand erzielbar und hat eine gute Kraftverteilung.

4.2 KONSTRUKTION

Nach der Designstudie wurde die konkrete Konstruktion in Angriff genommen. Mittels des CAD Programmes Pro Engineer wurden Detailpläne erarbeitet und im Team, sowie mit der mechanischen Werkstatt, besprochen. Aufgrund dessen, dass der Quadrocopter ein Prototyp wird, wurde der Konstruktionsaufwand absichtlich gering gehalten.

In einer ersten Phase wird die Funktionalität als wichtigster Punkt angesehen. Leider wurden die ersten Pläne aufgrund des hohen Arbeitsaufwandes verworfen. Die anschliessenden Skizzen wurden wiederum im Team besprochen und für gut / funktional befunden.

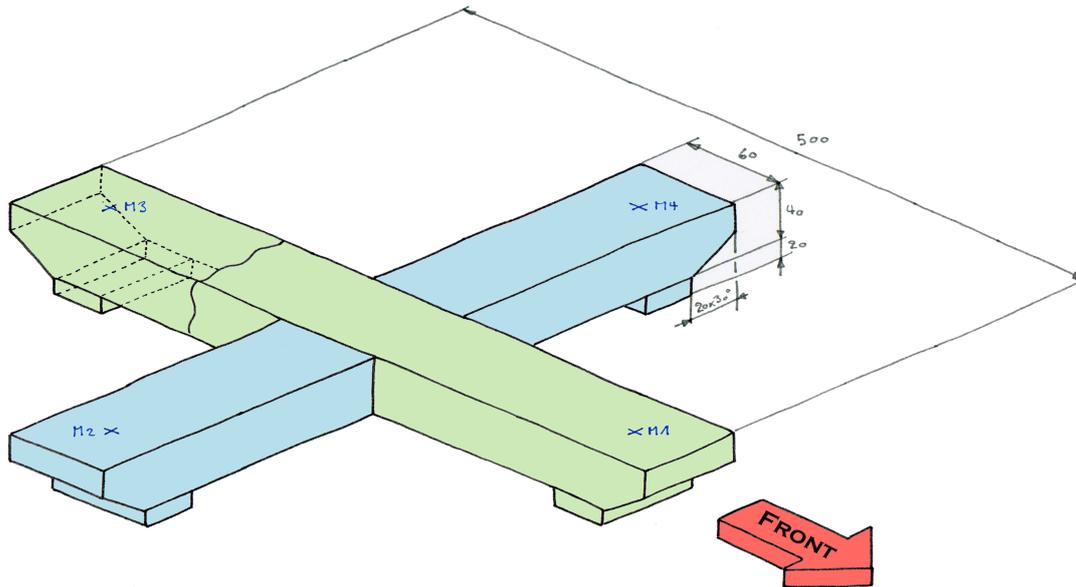


Abb. 2 Handskizze Konstruktion

So wird unser Quadrocopter aus zwei Trägern, welche ein Kreuz bilden und mittels Steckverbindung verbunden sind, realisiert. Durch die einfachen rechteckigen Formen wird versucht, den Bearbeitungsaufwand klein zu halten. Die vollständigen Pläne für die Konstruktion sind im Anhang zu finden.

Zur Grössen Definition musste auf mehrere Punkte geachtet werden. So sind mehrere minimale Achsenabstände zwingend gegeben durch die Grösse der Rotorblätter. Die Dicken der Träger wurden aufgrund von Materialversuchen (Festigkeitsstest) ermittelt. Die Bodenhöhe ist durch die Akkumulatorgrösse vorgegeben.

4.3 MATERIALAUSWAHL

Bei der Werkstoffauswahl musste „nur“ auf zwei Punkte geachtet werden. Die Funktionalität (Gewicht, Festigkeit) muss gewährleistet werden und der Preis des Rohstoffes sollte gering gehalten werden. Zur Auswahl standen nach einer groben Auswahl drei Werkstoffe:

- Balsaholz
- Styrodur
- Polymethylmethacrylat

Aufgrund der Kosten wurde die Variante mit Polymethylmethacrylat nicht realisiert. Zusätzlich wäre das Gewicht wegen der hohen Dichte ($\rho > 1 \text{ kg/dm}^3$) zu gross geworden.

Die Balsaholz, sowie die Styrodur Konstruktionen wurden, nach Belastungstest mit Musterteilen, beide für gut befunden. Um die Materialbeschaffungskosten klein zu halten, wurde die Styrodur-Variante realisiert. Das Material war bereits in der internen Fachhochschulwerkstatt vorhanden und konnte kostenneutral übernommen werden.



4.4 KLEBSTOFFWAHL

Für die Befestigung einzelner Bauteile zu einer Baugruppe wurden verschiedene Klebstoffe verwendet. Alle diese Klebstoffe mussten getestet werden und auf Ihre Verträglichkeit mit den diversen Baustoffen analysiert werden.

Nicht alle Klebstoffe welche getestet wurden, waren schlussendlich auch geeignet für unseren gebrauch. So hat es Klebstoffe (z.B. Zementit), welche die benötigten Festigkeiten auf dem porösen Styrodur nicht erreichten.

Am Ende wurden zwei Klebstoffe angewendet. Einerseits ein Klebstoff welcher die Verbindungsstellen von Styrodur zu Styrodur ausgezeichnet hält und andererseits ein Klebstoff welcher für die Verbindung von Styrodur und anderen Werkstoffen (z.B. Holz, Kunststoff) verwendet wurde.

4.5 FLUGHILFE



Abb. 3 Hängevorrichtung

Um die Flugeigenschaften zu überprüfen ist aufgrund unserer nicht vorhandenen Flugkenntnissen mit Sicherheit eine Flughilfe von Nöten. Diese Flughilfe soll uns in einer ersten Phase als Stabilisierung des Fluggerätes dienen.

Der Aufwand wurde absichtlich sehr gering gehalten. So ist schlussendlich nur eine Befestigung des Quadrocopter an der Decke entstanden, welche ihn seitlich durch die Kabellänge begrenzt.



5. HARDWARE

5.1 AT90USB1287

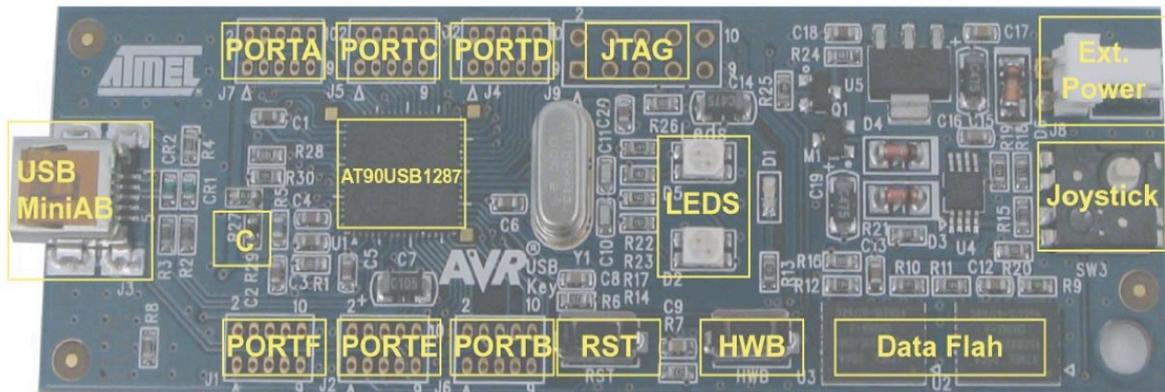


Abb. 4 Entwicklungsboard AT90USB1287

Als Grundlage unseres Projekts dient das AT90USB-Key- Entwicklungsboard, welches sich durch seine geringe Masse auszeichnet und daher geeignet ist. Der darauf verbaute Controller AT90USB1287 ist ein leistungsstarker 8-bit-Mikrocontroller des Chipherstellers ATMEL. Seine Eckdaten, welche für unser Projekt von Bedeutung sein könnten, sind folgende:

- 8MHz Taktrate
- 4x unabhängige 8-bit-, resp. 16-bit-Timer/Counter
- 8x externe Interrupteingänge
- 10x 8-16-bit PWM-Ausgänge
- USB-Bootloader für direktes flashen via USB-Schnittstelle
- USART/UART-Schnittstelle für eventuelle Gyro-Sensoranbindung

Die Leistung dieses Chips ist klar ausreichend für unser Quadrocopter-Projekt, die Kunst wird in der Programmierung liegen, auf welche wir in einem späteren Abschnitt noch zu sprechen kommen.

Alle Zugriffe und Ansteuerungen finden über eine externe Lochrasterplatine statt. Diese wurde von uns konzipiert und zu einem späteren Zeitpunkt fabriziert. Der Schaltplan der Platine ist im Anhang zu finden.

5.2 SENDER UND EMPFÄNGER FÜR FERNSTEUERUNG

Zum Anfang des Projektes wurde uns eine Fernsteuerung, basierend auf einem digitalen 2.4 GHz Funksignal, zur Verfügung gestellt. Diese neuartige Technologie briliert durch exzellente Übertragungsqualität der Sendesignale, sowie einer guten Störungsunempfindlichkeit gegenüber externen Einflüssen. Bereits während den ersten Überlegungen zur Signalinterpretation mit dem Mikrocontroller, stellte sich die Frage, in welcher Abfolge die vier Ausgangssignale des Empfängers, welche z.B. direkt für Servos oder Motorregler verwendet werden können, ausgegeben werden. Erste Signalmessungen führten schnell zum Ergebnis, dass der zur Verfügung gestellte digitale Empfänger alle Pulse der vier Kanäle jeweils zeitgleich ausgibt.

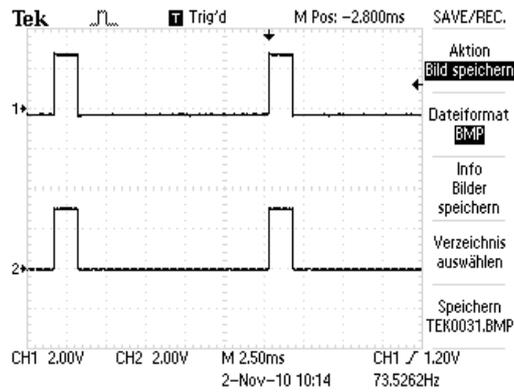


Abb. 5 Signalmessung Eingang

Die Signale, welche einzeln an Interrupteingängen des Controllers je eine Interruptroutine auslösen würden, haben eine äusserst ungünstige Form. Unter Umständen würde es bei deren Interpretation mit einem Mikrokontroller, gerade bei gleich langen Pulslängen aller Kanäle (z.B. Mittelstellungen), zu Interferenzen bei der Interruptabarbeitung kommen. Zusätzlich wird damit auch klar, dass der Empfänger intern nicht mit dem PPM-Signal arbeitet, welches für eine direkte Interpretierung aller Sendekanäle, mit einer Leitung, sehr interessant sein könnte. Aus diesen Gründen entschieden wir uns, unser Projekt mit einer älteren Fernsteuerung zu realisieren und nahmen gleichzeitig deren schlechtere Signalqualität in Kauf.

Um sicher zu gehen dass dieses System die Kanäle nacheinander ausgibt, was gleichzeitig darauf schliessen lässt, dass empfängerintern mit PPM gearbeitet wird, stellten wir ebenfalls Messungen an den Ausgangskanälen an.

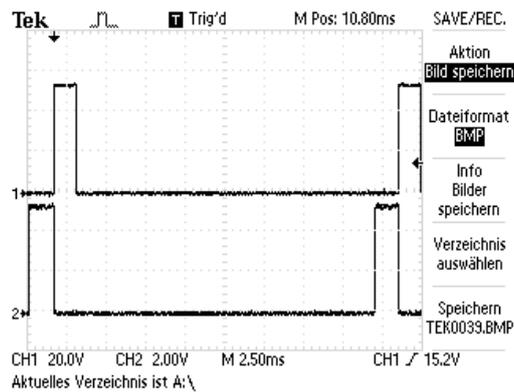


Abb. 6 Signalmessung Ausgang

Wie der obigen Abbildung zu entnehmen ist, bewahrheitete sich unsere Vermutung, der sequenziellen Signalausgabe. Um die Verbindung zwischen Empfänger und Controller zu vereinfachen und die Signalverarbeitung effizienter gestalten zu können, suchten wir auf der Empfängerplatine nach dem PPM-Signal, welches alle Kanalstellungen enthält.

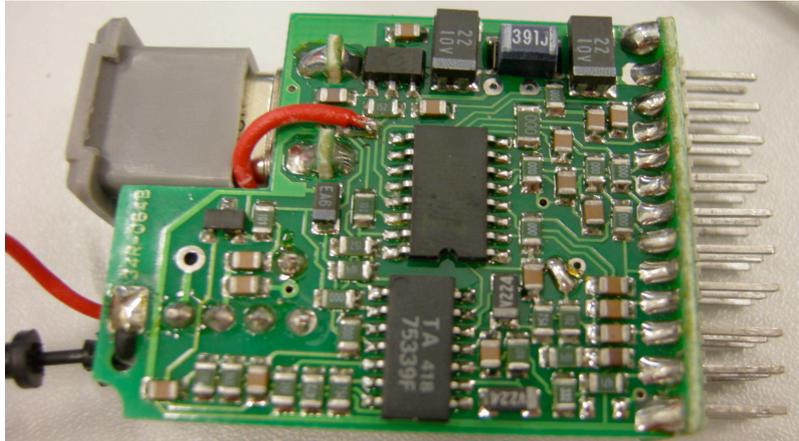


Abb. 7 PPM Signalabgriff an Empfänger (rote Litze)

Mit etwas logischem Schaltungsverständnis stiessen wir bald auf dieses Summensignal, welches wir mittels zusätzlicher Litze (rot, später blau) aus dem Empfänger hinausführten.

So ist es nun möglich, den Empfänger mit nur drei Verbindungen mit dem System zu verbinden: positiver Anschluss, Ground und PPM-Signal.

DAS PPM-SIGNAL

Wie auf den nachfolgenden Oszilloskop-Bildern zu sehen ist, enthält dieses Summensignal bei Sendung von acht Kanälen neun Pulse. Deren Zeitabstände (zwischen 0.8 bis 2.0 ms) bestimmen die Stellungen des Sendekanals.

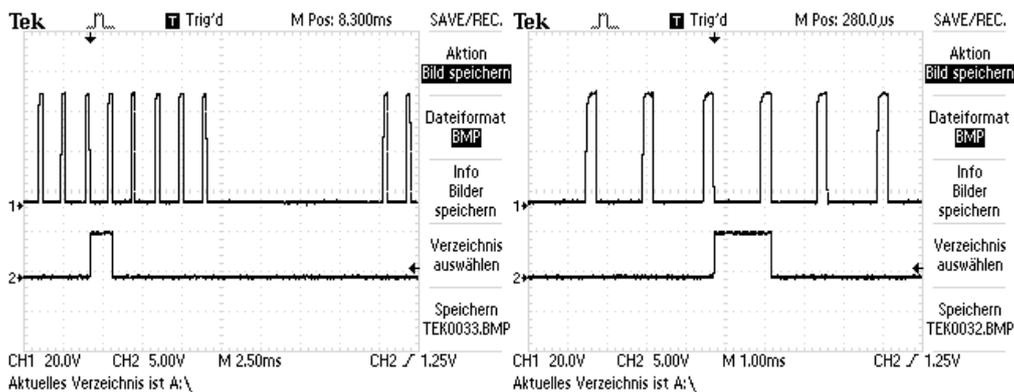


Abb. 8 Summensignal (PPM)

Das Ende, resp. der Anfang, der Signalabfolge wird durch eine Synchronisationsphase von ca. 12 ms gekennzeichnet.



Abb. 9 Empfänger zu MC-14

Zum ausgewählten Empfänger gehört ein Sender des Typs „mc-14“ der Modellbau-Elektronikmarke „Graupner“.



5.3 MOTOREN

Bei den 4 einbaufertigen Motoren, welche je an einzelnen Motorentreiber angeschlossen werden, handelt es sich um kleine Dreiphasen-Gleichstrommotoren. Diese unterliegen grundsätzlich dem Prinzip der dreiphasigen Asynchronmotoren. Das auffällige an den Motoren ist, dass sie „Aussenläufer“ sind. Dabei dreht sich eine Stahlglocke, bestückt mit sogenannten Neodym-Dauermagneten, um den innen sitzenden Stator, welcher mit in Dreieckschaltung geschalteten Stromspulen umwickelt ist.



Abb. 10 Brushless-Motor (Aussenläufer)

Die Motoren sind abwechslungsweise mit links- resp. rechtslaufenden Rotorblätter versehen. Dies soll verhindern, dass sich der Quadrocopter wie ein Helikopter ohne Heckrotor um seine eigene Höhenachse dreht. Werden nun alle vier Motoren gleichmässig angesteuert, hebt sich deren Drehmoment, welches sie auf den Rumpf ausüben, gegenseitig auf.

5.4 MOTORENTRIEBER



Abb. 11 Motorentreiber

Für die Ansteuerung der Motoren stehen uns vier einbaufertige Dreiphasen-Motorensteller zur Verfügung. Diese werden mittels einem, für den Modellbau üblichen, PWM-Signal angesteuert, welches eine Pulslänge von 0.8 bis 2.1 ms bei einer Trägerfrequenz von typischen 50 Hz ($T=20$ ms) besitzt. Um die Reaktionszeit der Regler zu verbessern, kann diese Trägerfrequenz bis auf ca. 70 Hz erhöht werden.

5.5 AKKU

Bei dem Akkumulator handelt es sich um ein dreizelliges Lithium-Polymer (LiPo) Akkupack mit einer Nennspannung von 11.1 Volt. Diese LiPo's zeichnen sich durch eine sehr hohe Energiedichte aus, was vor allem die Eignung für Flugobjekte begründet. In unserem Fall beträgt die Kapazität des Akkupacks 4900 mAh, was einer theoretischen Energie von 54.4Wh ($4.9Ah \times 11.1V$) entspricht.



Abb. 12 LiPo-Akku 4900mAh, 11.1V

5.6 RGB-BELEUCHTUNG

Zur Ortung und Orientierung des Flugobjekts, sowie für die Anzeige verschiedener Betriebszustände (Aufstarten, Akkustand, Nullabgleich) wurde der Quadrocopter mit RGB-Streifen versehen. Diese selbstklebenden LED-Leuchtstreifen haben je einen einzeln ansteuerbaren Rot-, Grün- und Blaustrang, welcher mit bis zu 12V versorgt werden kann. Die Streifen sind aus einem flexiblen Material und lassen sich an vorgesehenen Stellen beliebig trennen und einzeln einspeisen. Die Versorgung erfolgt über die Lochrasterplatine, auf welcher drei kleine MOSFETs als Leistungstreiber zwischen Akkupack und LED-Streifen dienen. Seitens des Mikrocontrollers werden diese am PWM-Ausgang an Port C (PC4, PC5, PC6) angeschlossen.

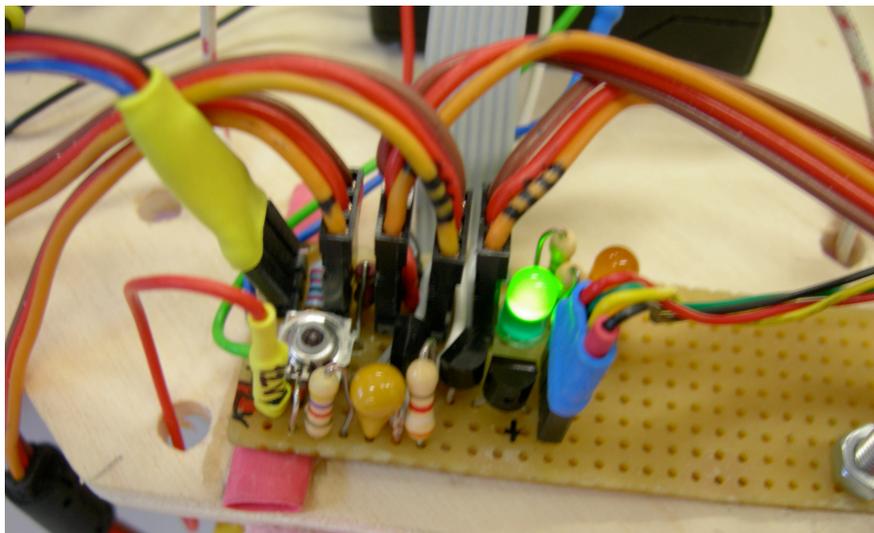


Abb. 13 Lochrasterplatine

5.7 AKKUSPANNUNGSÜBERWACHUNG

Ein weiteres wichtiges Feature ist die Spannungsüberwachung des Lithium-Polymer-Akkumulators. Die Akkuspannung wird mittels eines speziellen Spannungsteilers auf einen Bereich von 0-3.3 V begrenzt, welcher Analog zu den ca. 8-12.6 V des Akkus verläuft. Der Wert des Teilers wird am Port F.3, welcher als Analog-Digitalwandler konfiguriert ist, eingelesen. Der Ladezustand des Akkus soll dem Pilot durch die RGB-Beleuchtung mitgeteilt werden. Die für den Spannungsteiler notwendige Schaltungsberechnung sowie den



Akkuspannungen entsprechende Zahlenwerte im Controller sind in Nachfolgender Tabelle mit den entsprechenden Aktionen aufgeführt.

ADC-Akkuüberwachung an Port F.3

	Max	Nenn	Min	U-Z eff	R1	R2
AkkuSpg:	12.60 V	11.10 V	9.00 V	5.30 V	3900	4700
U nach Z-Diode:	7.30 V	5.80 V	3.70 V	Uref		
U ADC PortF.3	3.31 V	2.63 V	1.68 V	3.35 V		
I-quer	0.000848837	0.000674419	0.000430233	ADC-Auflösung		
ADC-Wert	252.9788268	200.9968761	128.2221451	256	8-Bit	

Signalisierungswerte	Spg. unter:	ADC-Wert	Was Tun:
Zu hohe AkkuSpg	12.66 V	255.0581048	RGB-Random-Blinken
Alle OK	12.60 V	252.9788268	ab und zu grün blinken
bald laden!	10.00 V	162.8767789	D9 Orange + ab und zu orange blinken
Akku laden!	9.50 V	145.549462	RGB Orange Blinken!
Akku leer! - kritisch!	9.20 V	135.1530719	RGB Rot Blinken!
Akku leer! - sehr kritisch!!!	8.90 V	124.7566817	RGB Rot Blinken! Flugunfähig, Alles 0!

Abb. 14 Akkuspannungsüberwachung



6. SOFTWARE

6.1 PROGRAMMIERUNG

Nach einigen Recherchen und Versuchen, den Controller mit dem ISP-Programmer (In System Programmer) JTAG MKII zu programmieren, gelang es uns später, mit der richtigen Software, direkt über USB zu programmieren. Der leistungsstarke AT90USB1287 Chip verfügt über einen sogenannten USB-Bootloader, welcher es ermöglicht direkt und ohne weiteren Hardware-mässigen Programmierhilfsmittel zu programmieren. Der Bootloader liegt in einem vordefinierten Teil des Speichers, welcher von ihm selbst nicht überschrieben werden kann, jedoch auf den restlichen Flashspeicher im Controller zugriff hat. Wird das Board nun durch Halten des „HBW“-Tasters und loslassen des Resettasters gestartet, so verbindet sich der Bootloader via USB mit dem Computer. Mit der AVR-Software „Flip“ des Chipherstellers Atmel lässt sich der Controller nun programmieren. Der Bootloader selbst lässt sich nur durch ein ISP-Programmer löschen oder ändern.

6.2 PPM SIGNAL-INTERPRETIERUNG

Als Zeitbasis der Signalverarbeitung dient der Timer/Counter 0 des Controllers, welcher auf eine Taktfrequenz von 125 kHz und einen Zählerbereich von 256 Schritten konfiguriert ist.

Das bereits im Abschnitt „Hardware“ beschriebene PPM-Signal des Empfängers wird an einem externen Interrupteingang „INT0“ des Mikrocontrollers angeschlossen. Die Interrupterkennung ist auf „steigende Flanke“ eingestellt, so dass bei jedem Flankenanstieg des PPM-Signals die Interruptroutine „EXT_INT0“ ausgelöst wird.

Bei jedem Interrupt des PPM-Signals wird nun der aktuelle Zählerstand des Zählers „TCNT0“ in eine Arrayvariable „kanal[x]“ gespeichert und der Timer wird zurückgesetzt. Anschliessend wird der Kanalpointer „CHin“, eine Variable welche den aktuellen Kanal und gleichzeitig den Speicherplatz des Zählerstandes im Array bestimmt, um eins erhöht. Des Weiteren wird bestätigt dass alle Kanäle gesendet wurden, indem bei der neunten Flanke die Bit-Variable „AlleCH“ auf 1 gesetzt wird.

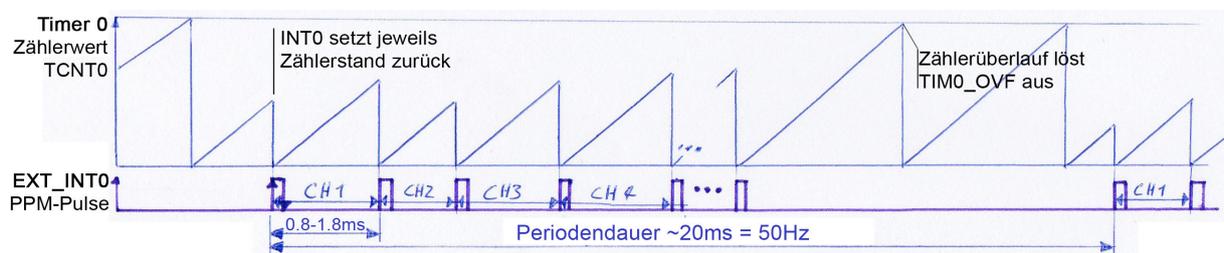


Abb. 15 PPM Signal Interpretation

Während der Synchronisationsphase, welche wie bereits beschrieben ca. 6-14 ms dauert, überläuft der Zählerwert mehrmals. Dabei wird ein „Timer/Counter 0 Overflow Interrupt“ ausgelöst: „TIMO_OVF“. Bei dieser Interruptroutine wird der Kanalpointer auf null zurückgesetzt.



Die gesamte Signalerkennung gibt also ein Array des Typs „Char“ mit den einzelnen Kanalwerten zwischen jeweils ca. 170 bis 255 für die weitere Verarbeitung im Programm zurück.

6.3 FUNKTIONSZUWEISUNG

Im Programmabschnitt der Funktionszuweisung werden, sofern das empfangene Signal OK ist, die Sendekanalstellungen den vier Flugfunktionen (pitch, gier, nick, roll) mit einem bestimmten Offset zugewiesen. Offset-Variablen dienen dazu, die Mittelstellungen der Sendekanäle mit denen des Programmes abzugleichen. Bei einer Neuprogrammierung des Controllers besitzen die Offsetvariablen vorgegebene Werte. Eine weitere extern ansteuerbare Interruptroutine (EXT_INT1) ermöglicht den Nullabgleich automatisch und speichert die Offsetvariablen im EEPROM-Speicher. Somit sind diese Einstellungen fix und auch nach Reset, Ausschalten oder Neuprogrammierung noch im EEPROM gespeichert. Der Abgleich wird durch einen Taster auf der externen Platine (Lochraserplatine) ausgelöst.

Beim Aufstarten des Controllers, sowie beim Neuabgleich der Nullpunkte, werden die Offsetvariablen vom EEPROM zusätzlich in den RAM-Speicher geschrieben um die Zugriffszeiten auf diese Werte, während des regulären Programmablaufs, erheblich zu verkürzen.

6.4 MIXER

Der Mixer ist im eigentlichen Sinne eine mathematische Berechnung der vier Stellgrößen aus den einzelnen Funktionen. Wie im unteren Mix-Diagramm ersichtlich wird, ist jede Ausgangsgröße von mindestens drei Funktionsgrößen abhängig.

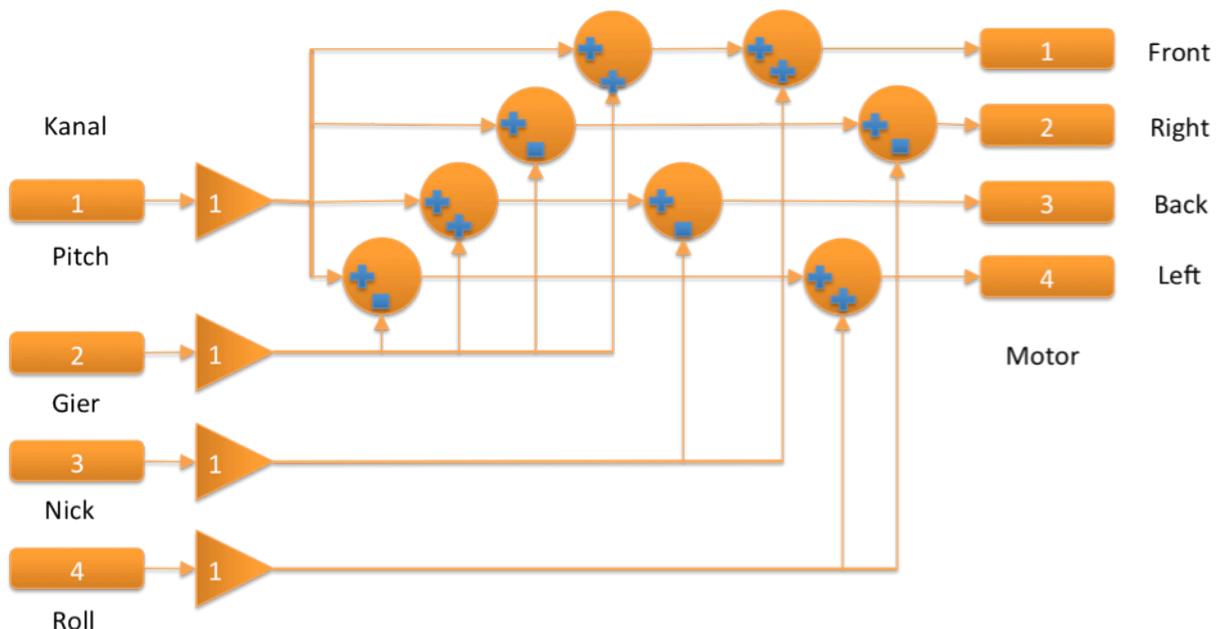


Abb. 16 Mixer Schema

Wie sich die einzelnen Größen zusammensetzen, wird durch die Drehrichtung sowie durch die Position des anzusteuern Motors bezüglich des Quadrocopter-Schwerpunkts bestimmt.

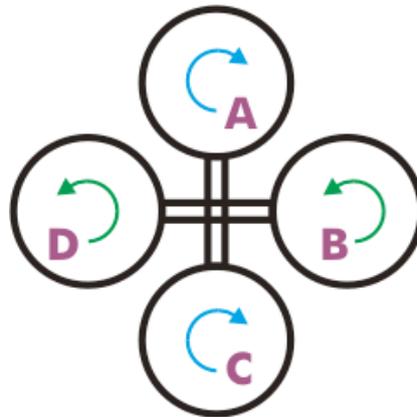


Abb. 17 Drehrichtung Motoren

Wird durch die Fernsteuerung z. B. der Kanal „Pitch“ erhöht, so sollen alle Ausgangskanäle synchron ein höheres Signal erhalten. Wird „Gear“ verändert, so sollen z. B. die Linksläufer schneller und die Rechtsläufer langsamer drehen, oder umgekehrt. Mit dem Steuerknüppel für „Nick“ resp. „Roll“ soll jeweils ein Motor schneller und sein gegenüberliegender langsamer angesteuert werden.

6.5 SIGNALAUSGABE

Die Aufbereitung des Signals für die vier Motorensteller wurde mittels eines weiteren Timers bewerkstelligt, welcher unabhängig vom restlichen Ablauf seine Routinen aufruft.

Wie bei der Signalinterpretation handelt es sich dabei um einen 8-Bit-Zähler (256 Schritte), welcher mit einer Taktfrequenz von 125 kHz hoch zählt.

OVERFLOW-METHODE

Die Signale werden bei dieser Variante immer bei Überlauf resp. bei Aufruf des Overflow-Interrupts, kommutiert. Zur Ausgabe eines Signals wird an einem Ausgang das entsprechende Bit eingeschaltet und der Zählerwert (TCNT2) wird auf 255 abzüglich Signalwert geladen. Sobald der Zähler das nächste Mal überläuft, was zeitlich vom vorgeladenen Zählerwert abhängig ist, wird das zuvor aktivierte Bit wieder ausgeschaltet und der nächste Kanal bearbeitet. Dabei sind theoretische Pulslängen von ~0 bis 2.05 ms möglich.

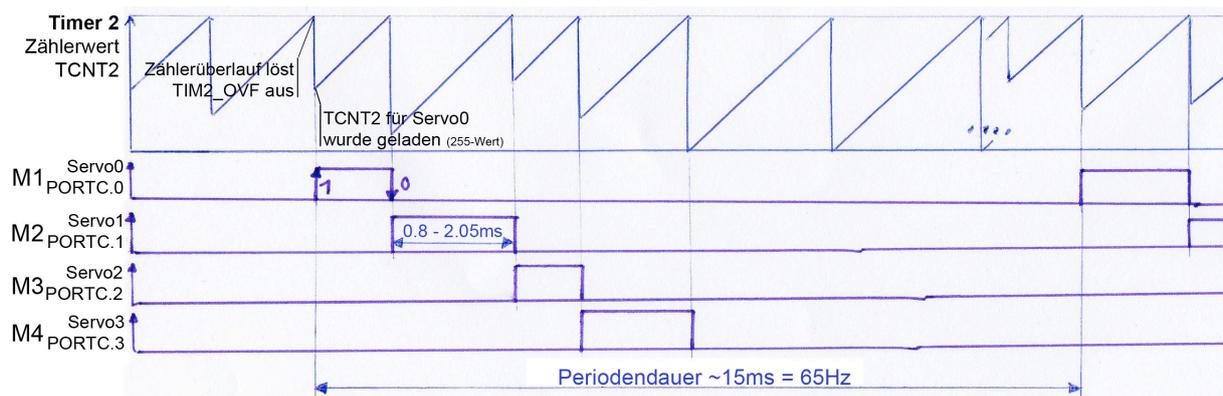


Abb. 18 Signalaufbereitung nach Overflow



Ein Signalwert 125 generiert z. B. ein Signal von genau einer Millisekunde Pulslänge. Der Zählerwert TCNT2 wird dabei auf $255-125=130$ gesetzt, was ihn bis zum nächsten Overflow-Interrupt (Wert 255) 125 Schritte zählen lässt. Die gesamte Periodendauer des Signals wird laufend summiert, damit der Signalanfang per Software rechtzeitig wieder von vorne gestartet wird.

COMPARE-MATCH-METHODE

Eine andere Methode für die Aufbereitung des gleichen Signals ist mit der Funktion „Compare-Match“ für welche der Mikrokontroller ebenfalls eine Interrupt Routine bereitstellt. Dabei wird der Zähler immer von 0 aufwärts zählen. Ein weiterer Registerwert OCR2A hingegen bestimmt, wann das Interrupt ausgelöst werden soll.

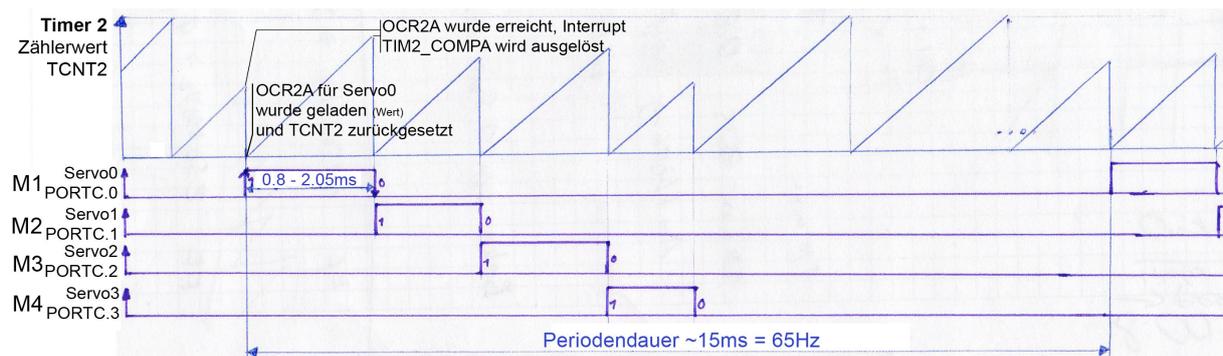


Abb. 19 Signalaufbereitung nach Compare-Match

Um nun mit dieser Methode einen Signalimpuls von 1 ms zu generieren (Taktrate 125 kHz), wird nach Rücksetzen des Zählerwertes $TCNT2=0$ der Vergleichswert OCR2A direkt auf 125 gesetzt. Sobald nun der Zähler 125 erreicht hat, wird die TIM2-COMPA-Interruptroutine ausgelöst und das Signal kann beendet werden.

Ein möglicher Vorteil dieser Methode liegt darin, dass die CPU etwas weniger rechnen muss als bei der zuvor beschriebenen Methode mit dem Vorladen des Zählerwertes auf einen bestimmten Wert.



7. PROBLEMSTELLUNGEN

7.1 TREIBERINSTALLATION PLATINE

Im Gegensatz zu allen anderen Gruppen, waren für unser Projekt (unseren Controller), keine Treiber vorhanden. In einigen mühsamen Arbeitsschritten mussten diese gesucht werden. Dabei erlaubten die Treiber das direkte Ansteuern des Controllers nicht. Durch neues flashen des USB Bootloader auf dem Chip und der Verwendung des Flashprogrammes Flip von ATMEL, konnten wir eine direkte Verbindung herstellen und uns die Arbeit erleichtern. Leider kostete uns diese Installation ca. einen Tag unseres kuren Zeitfensters.

7.2 PARALLELE SIGNALVERARBEITUNG UND AUSGABE

Dadurch, dass mehrere Interrupts gleichzeitig auftreten können (Timer und Externe Pins), kommt es unweigerlich zu Überschneidungen der einzelnen Interruptroutinen. Dabei spielt die jeweilige Priorität eine grosse Rolle. ATMEL-Microcontroller bieten dazu ein „[Multi-Level Interrupt Handling](#)“, welches die Abarbeitung der Routinen gemäss deren Priorität behandelt. Dabei werden Interrupts, welche während dem Ausführen eines tiefer priorisierten Interrupts auftreten, sofort abgearbeitet, während der vorherige Interrupt unterbrochen wird.

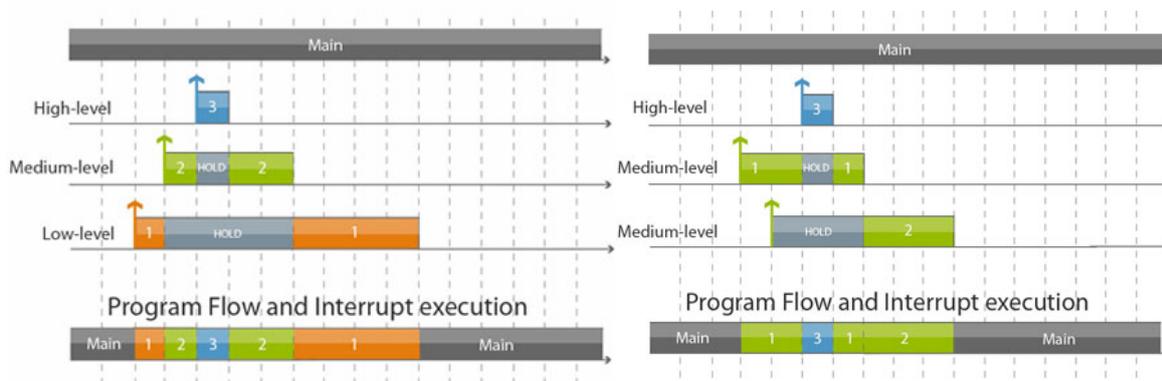


Abb. 20 Interrupt Verarbeitung

Wie der obigen Darstellung zu entnehmen ist, werden gleich priorisierte Interrupts nach dem Prinzip „first come, first served“ abgehandelt.

Allerdings lässt sich gemäss unseren Recherchen die Interruptpriorität mit dem CodeVision AVR C-Compiler nicht einstellen, weshalb wir davon ausgehen, dass alle Interrupts (von Timer und externen Pins) die gleiche Priorität haben.

Werden nun Interruptroutinen nicht zu ihrem erwünschten Zeitpunkt abgearbeitet, führt dies zu Unregelmässigkeiten im Programmablauf. Zählerwerte bei der Signalinterpretation, welche zu spät ausgelesen werden, haben da bereits einen verfälschten Wert, während sich verlängerte Pulsbreiten bei der Signalausgabe auf die Motorensteller mit vergrösserten Stellwerten bemerkbar machen.

Diesem Problem müsste man mit mehreren Controllern, welche autonom arbeiten, oder mit einer durchwegs synchronisierten Ablaufsteuerung, Rechnung tragen.



7.3 PLATZMANGEL

Aufgrund dessen, dass wir unsere Konstruktion möglichst klein halten wollten, ist es zu einem späteren Zeitpunkt schwerer gefallen, alle Komponenten auf dem Fluggerät zu platzieren. Durch Austesten mehrere Varianten konnte schlussendlich eine geeignete gewählt und umgesetzt werden. So sind momentan alle Komponenten untergebracht und miteinander verbunden.

7.4 KLEBSTOFF

Der Klebstoff, welcher unser Kreuz in der Mitte zusammenhalten sollte, erreichte nicht die gewünschte Festigkeit. Deshalb wurde mit einer Deck- und Grundplatte das gesamte System zusammen verschraubt. Dadurch gewinnt das System an Stabilität und wird nur leicht schwerer (ca. 80g). Aufgrund dieser Stabilitätsstütze wurden nachträglich auch die Ziele der Statik erreicht.



8. REFLEXION

In den vergangenen Wochen haben wir uns mit unserem Projekt „Quadrocopter - Faszination Fliegen“ auseinandergesetzt. Während dieser Zeit haben wir uns im Team tiefer mit dem Gebiet der Mechatronik befasst. Wir haben unser Wissen mit Spass und Freude erweitert und dabei etwas auf die Beine gestellt, was uns nachhaltig in Erinnerung bleiben wird.

Nicht immer lief alles wie gewollt (z.B. Treiberinstallation), dennoch versuchten wir, die an uns gestellten Hürden zu überwinden und mit einer lösungsorientierten Strategie die Probleme zu beheben und aus diesen neue Chancen zu kreieren. Mit dieser Strategie ist es uns gelungen, unsere Ziele nach unserer Zieldefinition zu erreichen.

Am Ende können wir von einem erfolgreichen Projekt sprechen, obschon einige Hürden zu hoch waren um sie in dieser geringen Zeit zu überspringen.



9. ABBILDUNGSVERZEICHNIS

ABB. 1 DIVERSE QUADROCOPTER VARIANTEN.....	8
ABB. 2 HANDSKIZZE KONSTRUKTION	9
ABB. 3 HÄNGEVORRICHTUNG	10
ABB. 4 ENTWICKLUNGSBOARD AT90USB1287	11
ABB. 5 SIGNALMESSUNG EINGANG	12
ABB. 6 SIGNALMESSUNG AUSGANG	12
ABB. 7 PPM SIGNALABGRIFF AN EMPFÄNGER (ROTE LITZE).....	13
ABB. 8 SUMMENSIGNAL (PPM)	13
ABB. 9 EMPFÄNGER ZU MC-14.....	13
ABB. 10 BRUSHLESS-MOTOR (AUSSENLÄUFER)	14
ABB. 11 MOTORENTREIBER	14
ABB. 12 LiPo-AKKU 4900MAH, 11.1V	15
ABB. 13 LOCHRASTERPLATINE	15
ABB. 14 AKKUSPANNUNGSÜBERWACHUNG	16
ABB. 15 PPM SIGNAL INTERPRETATION	17
ABB. 16 MIXER SCHEMA.....	18
ABB. 17 DREHRICHTUNG MOTOREN.....	19
ABB. 18 SIGNALAUFBEREITUNG NACH OVERFLOW	19
ABB. 19 SIGNALAUFBEREITUNG NACH COMPARE-MATCH	20
ABB. 20 INTERRUPT VERARBEITUNG	21

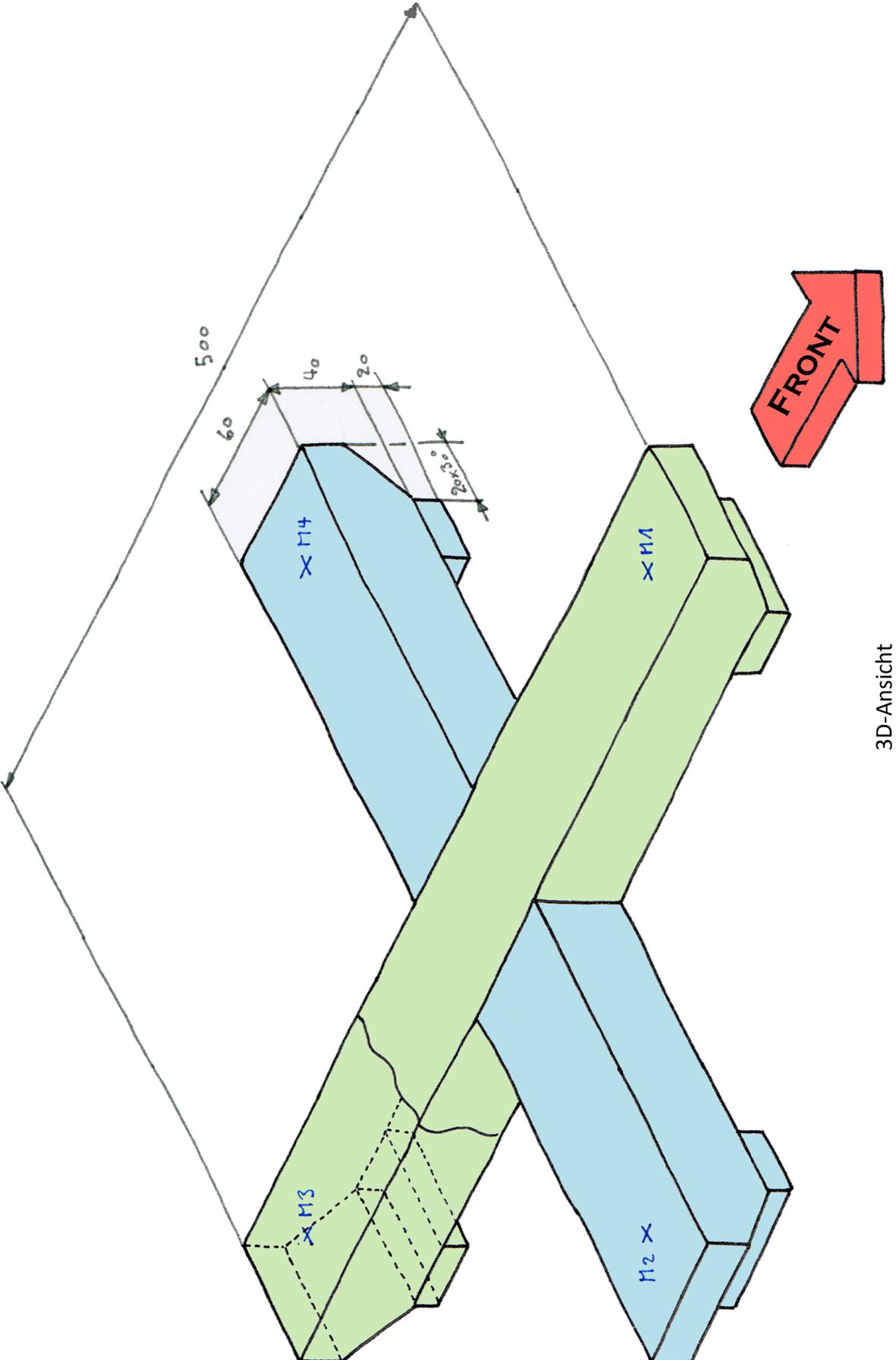


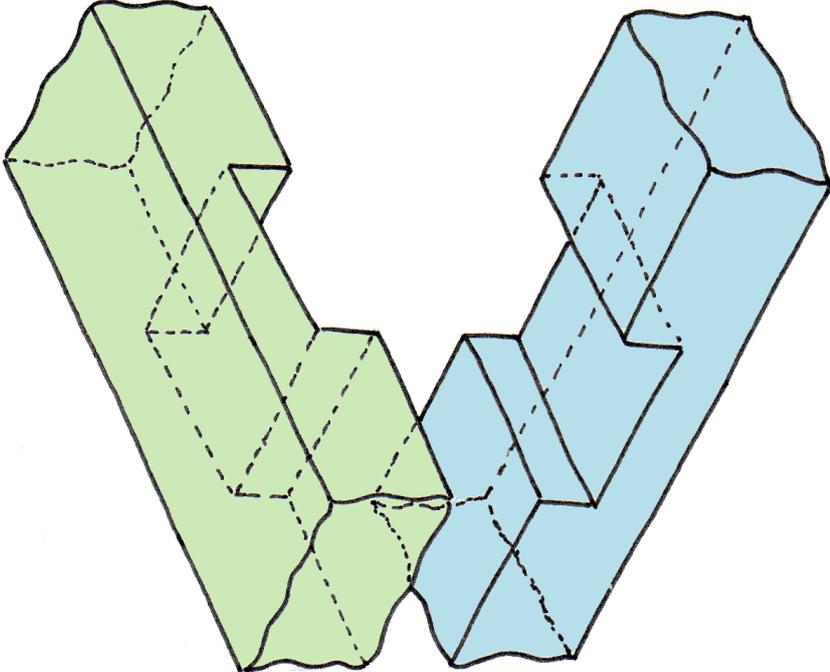
10. ANHANG

KONSTRUKTIONSZEICHNUNGEN	26
LOCHRASTERPLATINE (SCHALTPLAN)	28
SOFTWARE	29

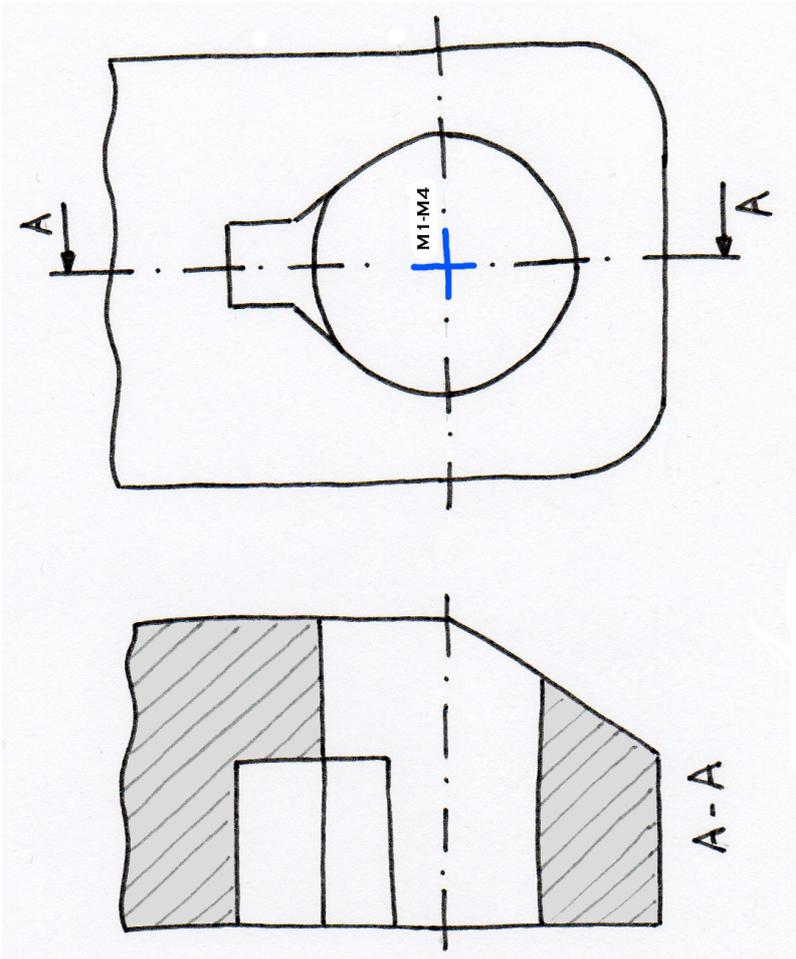


KONSTRUKTIONSZEICHNUNGEN





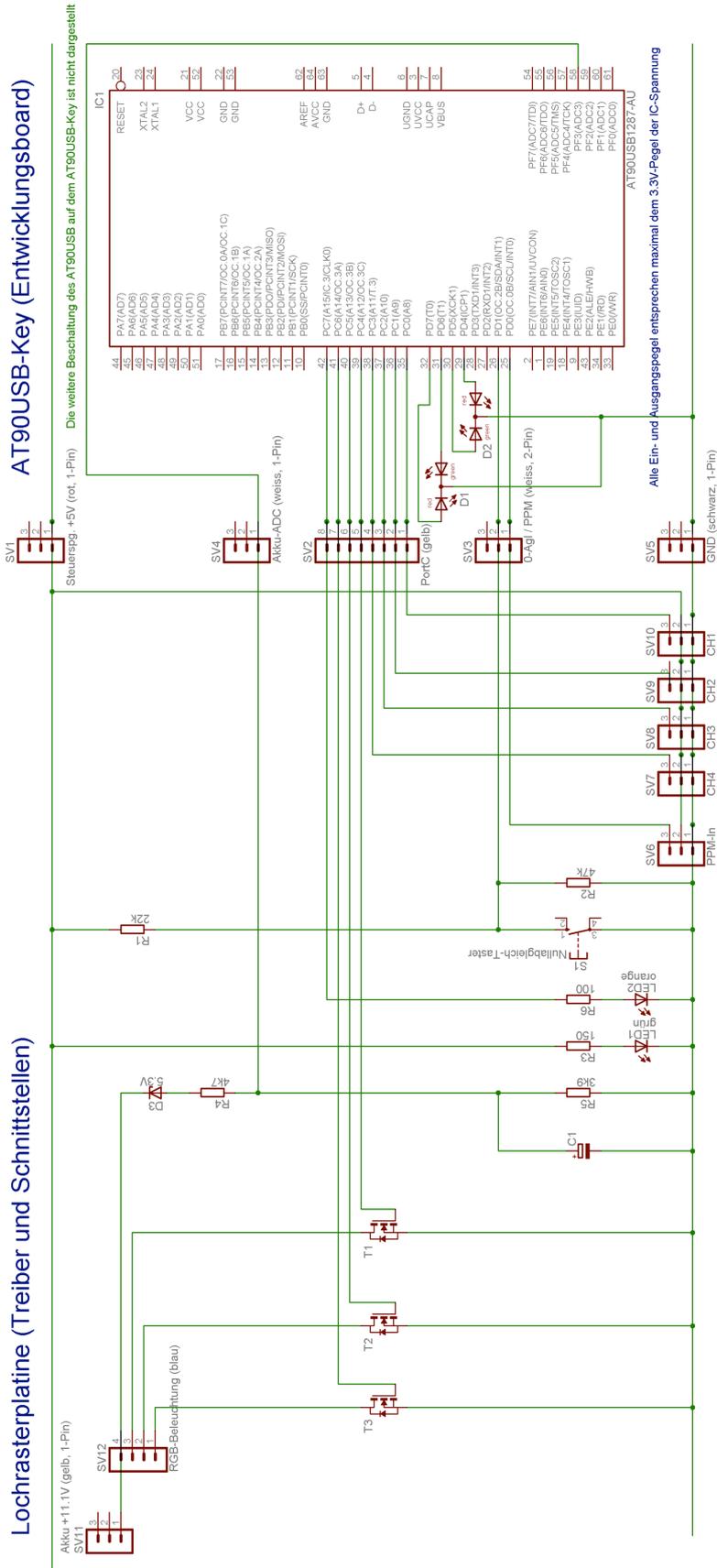
Steckverbindung



Motorenbefestigung



LOCHRASTERPLATINE (SCHALTPLAN)



QUADROCOPTER



SOFTWARE

Aus ökologischen Gründen kann die Originalsoftware nur der beigelegten Daten-CD entnommen werden und ist hier nicht publiziert.