

Die Hubschrauber von morgen Ein Blick in die nahe Zukunft

Seminararbeit von Tobias Schlegel

Thema: Die Hubschrauber von morgen – ein Blick in die nahe Zukunft.
Autor: Tobias Schlegel
Datum: 08.06.08 - 23:01:00

ONLINEVERSION – Dokument steht unter Creative Commons-Lizenz. Bilderrechte bei den Urhebern.

Meta: 06.04.08:20.56.16:29:0.10 LIGHT

0) Inhaltsverzeichnis

0) Inhaltsverzeichnis.....	Seite: 2
1) Einleitung.....	Seite: 3
1.1) Einleitung.....	Seite: 3
2) Historisches.....	Seite: 5
2.1) Erfindung des Hubschraubers.....	Seite: 5
2.2) Probleme, Hürden und deren Lösung.....	Seite: 7
2.3) Heutige Verwendung.....	Seite: 8
2.4) Zukunft.....	Seite: 9
3) Planung.....	Seite: 9
3.1) Hinführung.....	Seite: 9
3.2) Allgemeine Probleme.....	Seite: 10
3.2.a) Allgemeines zum Problem „Autonomer Hubschrauber“.....	Seite: 10
3.2.b) Lageerkennung.....	Seite: 11
3.2.c) Gyroskope und Beschleunigungssensoren.....	Seite: 12
3.2.d) Das Rauschproblem und dessen Lösung.....	Seite: 14
3.2.e) Lageregelung.....	Seite: 15
3.3) Systemdesign.....	Seite: 15
3.3.a) Genereller Überblick.....	Seite: 15
3.3.b) Lageerkennungssystem.....	Seite: 16
3.3.c) Warum viele kleine statt eines großen Prozessors?.....	Seite: 16
3.3.d) Kommunikations-Infrastruktur.....	Seite: 17
3.3.e) Kommunikationsprotokoll - Advanced Serial Communication Protocol.....	Seite: 20
3.3.f) Redundanz.....	Seite: 21
3.3.g) Softwarearchitektur.....	Seite: 22
4) Umsetzung.....	Seite: 23
4.1) Allgemeines.....	Seite: 23
4.2) Gyro-Module.....	Seite: 24
5) Zusammenfassung.....	Seite: 26
6) Danksagung.....	Seite: 27
7) Anhang.....	Seite: 28
7.1) Quellenverzeichnis.....	Seite: 28
7.2) Abbildungsverzeichnis.....	Seite: 29
7.3) Schaltplan und Blockschaltbild des Gyromoduls.....	Seite: 29
7.3.a) Blockschaltbild.....	Seite: 29

1) Einleitung

1.1) Einleitung

Inspiziert von den Vögeln baute Otto Lilienthal¹ seine filigranen Fluggeräte und erfüllte sich somit seinen langjährigen Traum vom Fliegen. Er schaffte damit zwar nur wenige hundert Meter, doch stellten seine Versuche alles bisher da gewesene in den Schatten. Bald darauf entwickelten die Gebrüder Wright² ihren Wright-Flyer und begannen so die Ära der Flächenflugzeuge.

Wir alle kennen diese wunderbare Story wie der Mensch zu seinen Flügeln aus Stahl kam. Doch wer kennt die Erfinder des Hubschraubers?

Nun, der ein oder andere Flugzeugliebhaber wird mir nun entgegenen, er wisse ganz genau, es sei Igor Iwanowitsch Sikorski³ gewesen, der den Hubschrauber erfunden hat. Wieder andere entgegenen, das sei ganz sicher Leonardo da Vinci⁴ gewesen... Doch leider ist dies ein weit verbreiteter Irrglaube.

Leonardo da Vinci⁵ dachte zwar bereits in seiner Zeit über Hubschrauber nach.

Mit seiner „Flugspirale“⁶

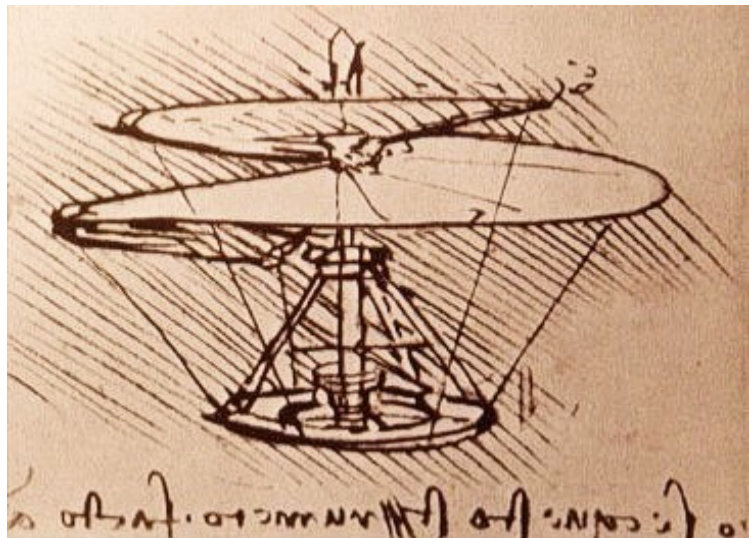


Abbildung 1: Leonardo da Vincis Flugspirale

schuf er auch erstmals Pläne für eine Art frühen Hubschrauber. Doch selbst er war nicht der „wahre“ Erfinder des Hubschraubers, denn die Chinesen

1 Otto Lilienthal: (*1848 in Anklam, Provinz Pommern; † 10. August 1896 in Berlin (durch Absturz)) Deutscher Flugpionier und Ingenieur, flog als erster Mensch mit einem Gleiter.

2 Gebrüder Wright: Orville Wright (* 19. August 1871 in Dayton, Ohio; † 30. Januar 1948) und Wilbur Wright (* 16. April 1867 in Melville, Indiana; † 30. Mai 1912 in Dayton, Ohio) Amerikanische Flugpioniere und Erfinder des Motorflugs.

3 Igor Iwanowitsch Sikorski: (* 25. Mai 1889 in Kiew; † 26. Oktober 1972 in Easton (Connecticut)) Ingenieur und Luftfahrpionier, wurde bekannt durch seine Hubschrauber-Konstruktionen und durch seine (noch heute existierende) Firma, die Sikorsky Aircraft Corporation.

4 Siehe FN No. 5

5 Leonardo da Vinci (* 15. April 1452 in Anchiano bei Vinci; † 2. Mai 1519 auf Schloss Clos Lucé, Amboise) italienisches Universalgenie.

6 Leonardo da Vincis Flugspirale

hatten bereits ca. 500 v. Chr. ein Spielzeug, das auf dem Hubschrauber-Prinzip beruhte, entwickelt. Dieses „chinesischer Kreisel“⁷ genannte Spielzeug bestand aus einem Holzkorpus mit radial angebrachten Federn, das, wenn es mit einer Schnur in Drehung versetzt wurde, abheben und etwas fliegen konnte. Noch heute gibt es diese Art von Spielzeug in vielen Farben und Variationen.

Heutzutage sind Hubschrauber aus dem täglichen Leben nicht mehr wegzudenken. Ob sie nun Geschäftsleute schnell zum Flughafen bringen, eine Einsatzplattform für die Feuerwehr oder die Polizei bilden oder Leute aus Gefahrensituationen retten; Hubschrauber bieten vielseitigste Einsatzmöglichkeiten. Ihr Hauptvorteil ist ihre Fähigkeit zum senkrechten Start und Landung sowie die Möglichkeit zu schweben. Leider haben das die Militärs zuerst erkannt; Hubschrauber wurden zuerst beim US-Militär eingesetzt. Auch heute noch sind beim Militär oft und viele Hubschrauber im Einsatz. Doch im Gegensatz zu den Anfangsjahren, in denen Hubschrauber Militärische „Geheimwaffen“ waren, sind heute mindestens ebenso viele Hubschrauber im zivilen Einsatz. Sogar das Abfliegen und Überprüfen von Gasleitungen wird heute von Hubschraubern aus durchgeführt.

In den letzten Jahren wurde allerdings eine interessante Richtung in der Hubschrauberentwicklung deutlich: Die Hubschrauberentwicklung geht immer mehr in die Richtung UAV⁸. Dies sind kleine Hubschrauber, meist nicht größer als ein Laptop, die teilweise oder vollkommen autonom in Ihrer Umgebung navigieren und fliegen können.

Auch im Hobbybereich ist eindeutig ein Zunehmen solcher Projekte zu verspüren, nicht zuletzt deshalb, weil entsprechende Mikrocontroller und Sensortechnik in den letzten Jahren immer billiger wurde. Leider schlagen viele Projekte auf Hobbybasis durch den doch recht hohen Schwierigkeitsgrad fehl.

Ich habe mich mit der Entstehung und den Einsatzgebieten von

⁷ Über den „chinesischen Kreisel“ ist nicht viel bekannt. Eine kleine Beschreibung gibt [PDF01].

⁸ UAV: Unmanned Aerial Vehicle – sinngemäß: „Unbemanntes, automatisches Fluggerät“

Hubschraubern beschäftigt, theoretisch ein UAV geplant und mit dessen Bau begonnen.

2) Historisches

2.1) Erfindung des Hubschraubers

Die Erfindung des Hubschraubers war kein kontinuierlicher Prozess und deshalb auch immer wieder unterbrochen. Vor allem aber ist zu erwähnen, dass beim Hubschrauber, ähnlich wie beim Flugzeug, viele Leute gleichzeitig an einer Entwicklung gearbeitet haben, sodass z.T. manche Erfindungen 2 mal gemacht wurden.

Obwohl der Hubschrauber im heutigen Sinne im 20. Jahrhundert „entwickelt“ wurde, hatten die Chinesen bereits 500 v. Chr. ein Spielzeug („chinesischer Kreisel“ genannt) entwickelt, das mittels 2er Federn und in schnelle Drehung versetzt, wie ein Hubschrauber fliegen konnte.

Sogar Leonardo Da Vinci hat 1483 mit seiner „Flugspirale“ eine Art Hubschrauber thematisiert. Leider mangelte es damals an entsprechend leichten Baumaterialien und einem entsprechend leistungsstarken Antrieb.

Natürlich haben heutige Hubschrauber nicht mehr viel mit dem chinesischen Kreisel oder Da Vincis Flugspirale gemeinsam. Es waren z.T. kleine Entwicklungen und Veränderungen, die die Hubschrauberentwicklung weiter brachte.

Zunächst einmal muss man zwischen dem Hubschrauber und dem Gyrocopter unterscheiden. Der Hubschrauber verwendet einen aktiv angetriebenen Rotor um Auftrieb zu erzeugen, der Gyrocopter dagegen hat einen passiv (vom Fahrtwind) angetriebenen Rotor, der eine Tragfläche ersetzt. Der Gyrocopter muss außerdem wie ein Flugzeug von einem Propeller nach vorn beschleunigt werden.

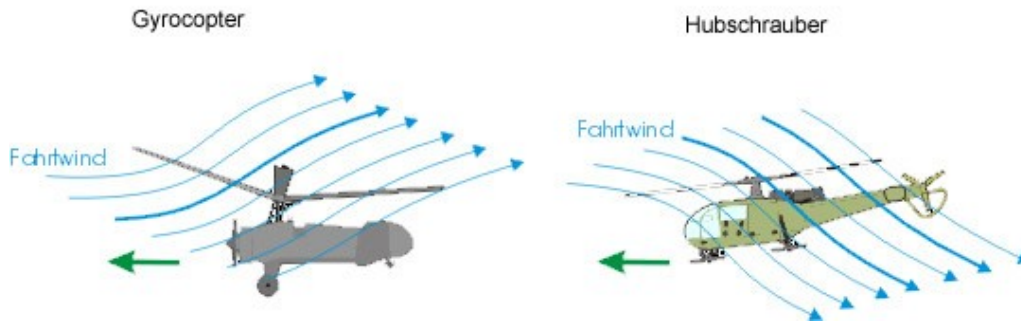


Abbildung 2: Gyrocopter - Helikopter

Der erste, der einen echten Hubschrauber im heutigen Sinne geflogen hat war der Franzose Paul Cornu am 13. November 1907. Sein Flug mit seinem „fliegenden Fahrrad“ dauerte ganze 20 Sekunden und Cornu erreichte eine Flughöhe von 30cm. Leider war sein Fluggerät nicht manövrierfähig, sodass das Projekt bald aufgegeben wurde.

Steuerbar wurde der Hubschrauber erst, nachdem ein gewisser Herr Pescara in Spanien einen Hubschrauber mit Koaxialrotor entwickelte, der über eine funktionierende Steuerung verfügte. Der Hubschrauber wog 850kg und erreichte eine Höchstgeschwindigkeit von 13 km/h. Pescaras Hubschrauber war der erste, der mit einer zyklischen und kollektiven Blattverstellung ausgestattet war, eine Steuerungsmethode, die auch heute noch in jedem Hubschrauber zum Einsatz kommt.

Der erste wirklich einsetzbare Hubschrauber war der Focke-Wulf FW-61. Dies war ein in Hitlerdeutschland entwickelter Hubschrauber.

Das größte Problem bei Hubschraubern ist das vom Rotor erzeugte Drehmoment. Denn es ist physikalisch



Abbildung 3: Hubschrauber mit Koaxial-Rotor der Heliswiss

gesehen egal, ob der Motor den Rotor oder der Rotor den Motor antreibt. Das hängt mit Newton's 3. Axiom zusammen: „Kraft beruht auf Gegenkraft“ (Actio

= Reactio). Ein Hubschrauber muss also dieses Drehmoment ausgleichen, um nicht um die eigene Achse zu rotieren. Dieser Drehmomentausgleich wurde damals mittels 2er identischen, gegenläufigen Rotoren (sog. Koaxialrotor) bewerkstelligt. Dieses Prinzip hat allerdings den Nachteil, dass ein relativ komplexes Getriebe und eine komplexere Aufhängung von Nöten ist. Daher war der nächste Fortschritt in der Hubschrauberentwicklung der einzelne Rotor.

Igor Iwanowitsch Sikorski entwickelte die auch heute noch üblichste Form des Hubschraubers, in Fachkreisen als „Heckrotor-Konfiguration“ bezeichnet. Sikorsky, der 1919 in die USA emigrierte, flog am 13. Mai 1940 mit dem von ihm konstruierten Hubschrauber Sikorsky VS-300 der mit einem Hauptrotor und einem Heckrotor ausgestattet war.

Von diesem Hubschrauber sind es nur kleine Schritte zu heutigen Hubschraubern. Man ging mit der Zeit dazu über, Turbinen an Stelle von normalen Verbrennungsmotoren zu verwenden, da diese ein besseres Leistungs/Masse-Verhältnis haben.

2.2) Probleme, Hürden und deren Lösung

Wie oben schon gesagt, ist die Manövrierbarkeit eines Hubschraubers kein triviales Problem. Herr Pescara löste es, indem er den Anstellwinkel der Rotorblätter allgemein (kollektive Blattverstellung) oder einzeln, abhängig von der Position des Blattes (zyklische Blattverstellung) veränderte.

Die kollektive Blattverstellung ermöglicht es dem Piloten, alle Rotorblätter gleichzeitig nach



Abbildung 4: Taumelscheibe eines Modellhubschraubers (Ikarus)

oben oder unten zu neigen, was den Auftrieb des Rotors maßgeblich beeinflusst (sog. „Pitch“).

Die zyklische Blattverstellung neigt die Rotorblätter je nach Position zum Rubschrauberrumpf. Werden die Blätter am Bug des Hubschraubers z.B. nach unten, am Heck nach oben geneigt, so wird am Heck des Hubschraubers Auftrieb, und am Bug Abtrieb erzeugt, und der Hubschrauber neigt sich nach vorne. Diese Steuerung wurde erst durch die sogenannte Taumelscheibe möglich. Sie Überträgt die Steuerbefehle vom Piloten an den Rotorkopf.

Des Weiteren muss der Heckrotor von einem speziellen Kreiselsystem überwacht werden, da der Hubschrauber sonst durch Windböen z.B. unkontrolliert ausbrechen könnte.

2.3) Heutige Verwendung

Hubschrauber finden heute in vielen Bereichen Anwendung. Sie werden u.a. zur Rettung von Menschenleben, dem Transport von wichtigen Gütern, von der Polizei zur Überwachung oder auch für diverse Sportarten (wie z.B. „Helicopter-Surfing“ oder „Heli-Skiing“) eingesetzt. Der Hubschrauber hat gegenüber dem Flugzeug einige Vorteile, die ihn für diverse Zwecke geradezu prädestinieren. Der größte dieser Vorteile ist seine senkrecht-Start-und-Lande-Fähigkeit, und die damit verbundene extreme Wendigkeit. So können Unfallpatienten aus unzugänglichem Gelände (z.B. Waldlichtung oder Bergabhang) geborgen werden.

Leider wird der Hubschrauber auch zu militärischen Zwecken eingesetzt. Die US-Army war auch der Haupt-Geldgeber von Herrn Sikorsky. Auch die militärischen Einsatzmöglichkeiten kennen (leider) keine Grenzen; Ob als Kampfhubschrauber oder Rettungshelikopter; auch beim Militär kennt man die Vorteile des Hubschraubers.

2.4) Zukunft

Wie bei allen Fluggeräten strebt man auch beim Hubschrauber nach immer mehr Automation. Schon längst sind Hubschrauber mit umfassenden Bordcomputern ausgestattet, die den Piloten beim Steuern unterstützen. Es gibt sogar Autopiloten für Hubschrauber.

Hubschrauber kommen aber auch zunehmend bei Konstrukteuren sogenannter UAVs in Mode. Das sind kleine, autonom operierende Flugobjekte, auch als Drohnen bekannt.

Während früher vor allem kleine Modellflugzeuge zum UAV umgebaut wurden, geht der Trend heute eher zum Hubschrauber. Der Grund ist, dass Hubschrauber gegenüber Flugzeugen den Vorteil haben, auf der Stelle schweben zu können, was z.B. in der Überwachung von z.B. Menschenmassen bei einer Demonstration große Vorteile bietet.

3) Planung

3.1) Hinführung

Die Aufgabenstellung war es, ein UAV zu entwickeln, das völlig autonom operieren kann. Zum Beispiel die Polizei setzt schon lange normale Hubschrauber für Überwachungs- und Verfolgungszwecke ein. Allerdings geht der Trend immer mehr zu kleinen, ferngesteuerten Hubschraubern, nahe Verwandte zu dem, was ich konstruieren will, nämlich einem vollautonomen Hubschrauber (oft auch „Drohne“ oder UAV genannt). Heutzutage sind sog. „Quadrocopter“ für solche Zwecke „in Mode“, also Hubschrauber mit 4 nach unten gerichteten Rotoren. Quadrocopter haben den Vorteil, dass sie die gesamte zum Antrieb verwendete Energie in den Auftrieb des Fluggeräts stecken. Dafür ist die Steuerung eines Quadrocopters recht komplex im Vergleich zu der eines normalen Hubschraubers.

Ich habe mich dazu entschieden, ein System zu entwickeln, das in einen handelsüblichen Modellhubschrauber eingebaut werden kann. Dazu hatte ich

verschiedene Gründe.

- 1) Ein handelsüblicher Hubschrauber fliegt. Es können keine Probleme mit dem Flugwerk auftreten, die nicht schon einmal gelöst wurden
- 2) Ersatzteile sind einfach verfügbar
- 3) Er bietet mehr Platz bei gleicher Leistung.
- 4) Ein handelsüblicher Hubschrauber ist billiger als ein selbstentwickelter Quadrocopter

Nachdem die ungefähren Umrissse des Projekts festgelegt waren, begann ich mit einer konkreten Planung.

3.2) Allgemeine Probleme

3.2.a) Allgemeines zum Problem „Autonomer Hubschrauber“

Stellen wir uns vor, wir haben einen perfekten Hubschrauber in einer perfekten Umgebung und wollen ihn schweben lassen. Wir starten den Rotor und Vergrößern den Anstellwinkel der Rotorblätter um mehr Auftrieb zu erhalten. Was passiert? Richtig; wir heben ab und der Hubschrauber pendelt sich irgendwo schwebend auf einer Höhe X ein. Soweit so gut.

Nun wissen wir aber, dass wir in einer ziemlich chaotischen Umgebung leben, das heißt für unseren kleinen Hubschrauber: Er selbst erzeugt Luftströmungen und -Wirbel und wird von Luftwirbeln (u.a. auch von seinen eigenen) erfasst und „gestört“. Leider ist es also in der realen Welt nicht so einfach, den Hubschrauber abheben zu lassen.

De facto ist ein Hubschrauber nämlich ein chaotisches, instabiles System. Um es bildhaft zu erklären: Man muss aktiv etwas tun, um den Hubschrauber am Fliegen zu erhalten, von selbst würde er abstürzen; ganz im Gegensatz zu normalen Flächenflugzeugen.

Jeder hat schon einmal versucht, eine Stange auf der Handfläche zu balancieren. Das ist garnicht so einfach und bedarf einiges an Konzentration.

Diese Stange auf der Handfläche ist nichts anderes als ein sog. inverses Pendel. Auch dieses Pendel ist instabil und muss aktiv stabilisiert werden.

Man muss also einen Hubschrauber unablässig vor dem Absturz bewahren d.h. seine Lage im dreidimensionalen Raum ständig messen und korrigieren.

Bei einem Hubschrauberpiloten läuft dieses Messen und Korrigieren vermutlich unterbewusst ab, genauso wie wir beim Autofahren ohne viel Nachdenken die Spur halten. Sein Roboterkollege allerdings braucht dazu Sensoren und passende Auswertlogik, um die Daten entsprechend verarbeiten zu können.

Meine Planung basiert auf der Annahme, dass der Schwebезustand der Grundzustand des Hubschraubers ist, und alle anderen Zustände (z.B. Beschleunigung, Geradeausflug, Kurven, Landen etc.) aus diesem Zustand ableitbar oder durch Mischung dieser Zustände entstehen.

Beschleunigen würde man einen Hubschrauber also z.B. indem man die „Normallage“ etwas nach vorn (oder in eine andere Richtung) kippt. So neigt sich der Hubschrauber nach vorn und beschleunigt. Möchte man abbremsen führt man das gleiche Manöver nur in entgegengesetzter Richtung aus.

Kurven entstehen dann durch eine Geradeaus-Komponente, bei gleichzeitigem Beschleunigen in eine andere Richtung sowie dem Abbremsen der Geradeaus-Komponente.

3.2.b) Lageerkennung

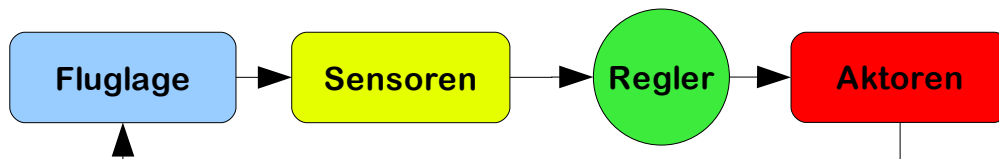
Wie oben bereits beschrieben muss der Hubschrauber seine Lage aktiv regeln. Dazu muss er 2 Bedingungen erfüllen:

- Der Hubschrauber muss wissen, wie er sich relativ zum Boden im Raum befindet
- Der Hubschrauber muss eine etwaige Fehlstellung korrigieren können.

Der Part zwischen „Erkennen“ und „Gegenlenken“ nennt man einen Regelkreis. Eine Fehlstellung tritt auf, wird von den Sensoren registriert, die

Fehlstellung wird an den Regler gemeldet; dieser berechnet den Korrekturwert und schickt ihn an die Aktoren („Beweger“) am Rotor, die entsprechend reagieren und die Fehlstellung korrigieren. Dies ist ein endloser Prozess der quasi in einer Endlosschleife ausgeführt wird.

Zeichnung 1 (s.o.) zeigt einen typischen Regelkreis, den ich bereits auf unsere Gegebenheiten angepasst habe.



Zeichnung 1: Hubschrauber Regelkreis

Für uns heißt dies: Die Sensoren nehmen eine Winkelabweichung zur 0-Lage⁹ auf, diese wird vom System verarbeitet und schließlich mittels der Aktoren am Rotor korrigiert.

Diese Regelung erfolgt natürlich nicht nur in einer (Raum-)Achse, sondern in allen 3 Dimensionen. Da man jede Achse für sich betrachten kann, ist es möglich, jede Achse für sich zu messen und zu regeln, was viel Arbeit spart, da sonst alle Achsen miteinander verrechnet werden müssten.

Das Konzept sieht vor, dass jede Achse für sich von einem eigenen Sensor vermessen wird. Die Regelung erfolgt dann über einen zentralen Prozessor, der nur für die 3 Achsenregler zuständig ist. Interessant an diesem Aspekt ist, dass sich **jede** Neigung des Hubschraubers durch dieses 3-Achsen-Modell beschreiben und kompensieren lässt, denn jede erdenkliche Lage des Hubschraubers lässt sich immer in eine X, eine Y und eine Z-Komponente zerlegen.

3.2.c) Gyroskope und Beschleunigungssensoren

Die Messung der (in 3.1.b erwähnten) Winkelabweichung ist allerdings nicht trivial. Es gibt dafür Sensoren, sogenannte Gyroskope (kurz Gyros). Diese

⁹ Gerade Ausrichtung des Hubschraubers, sodass er theoretisch nicht abdriften dürfte. Stellen Sie es sich wie ein Dreidimensionales Koordinatensystem vor, dessen Ursprung der Schwerpunkt des Hubschraubers ist, und dessen X-, und Y-Achse parallel zum Erdboden verlaufen (dessen Z-Achse durch den Erdmittelpunkt verläuft).

Gyroskope beruhen auf dem Satz der Drehimpulserhaltung der besagt, dass sich ein Drehimpuls in einem abgeschlossenen System weder in Betrag noch Richtung ändert.

Wenn eine Masse um sich selbst rotiert, dann versucht sie, ihre Drehgeschwindigkeit sowie ihre Ausrichtung bei zu behalten. Gyrosensoren messen, wie schnell sie sich um ihre Messachse drehen, also die sog. Drehrate oder Winkelgeschwindigkeit (Einheit: $^{\circ}/s$). Integriert man diese Messwerte über die Zeit, kommt man auf den Absolutwinkel. Nun arbeiten die Gyros, die ich verwendet habe, nicht mit drehenden Massen, sondern mit 2 schwingenden Massen, bei denen der Satz der Drehimpulserhaltung ebenfalls gilt (dieses Prinzip ist auch als „resonator Gyro“ bekannt). Dabei wird eine Masse auf einem Stab zum Schwingen angeregt. Diese Masse schwingt an einer gegenüberliegenden Oberfläche vorbei. Auf der Masse und auf der Oberfläche sind 2 elektrisch leitende Flächen angebracht, deren Flächeninhalte bekannt sind. Mittels dieser Flächen wird der Abstand zwischen schwingender Masse und der Oberfläche kapazitiv gemessen.

Die Gyro-Sensoren haben ein großes Manko. Dadurch, dass diese Massen schwingen, sich also mit einer Frequenz von mehreren Kiloherz hin und her bewegen, kann es passieren, dass sie nicht immer wieder exakt zu ihrem Nullpunkt zurückkehren. Der Nullpunkt dieser Sensoren schwankt – zwar in kleinem Maßstab aber gut messbar. Dieses an sich kleine Problem zieht eine Kette anderer, wesentlich schwerer wiegende Probleme nach sich; denn durch die notwendige Integration, um auf den Absolutwinkel zu kommen, vergrößert sich der Messfehler immer mehr. Laut erster Experimente kann der Messfehler ohne entsprechende stochastische Filter bis zu 300° pro Stunde betragen.

Beschleunigungssensoren

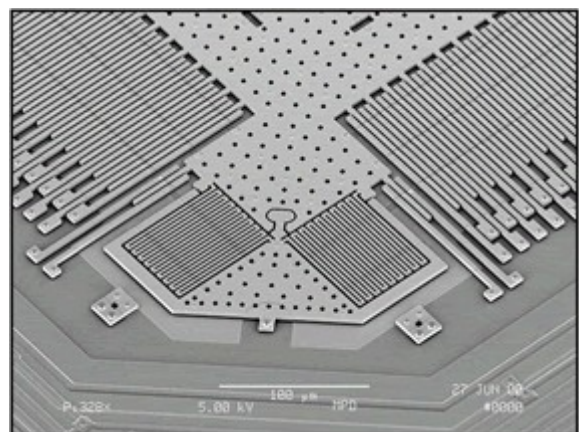


Abbildung 5: MEMS-Beschleunigungssensor: Ecke der trägen Masse (links und rechts: kapazitive Wegsensoren)

funktionieren nach einem sehr ähnlichen Prinzip. Bei diesen Sensoren wird eine Masse beweglich über Federn an der Umgebung aufgehängt. Wird nun der Sensor beschleunigt, beharrt die Masse auf ihrer Position (Massenträgheit). Sprich, sie drückt auf die Federn, und bewegt sich ein kleines Stück. Dieses kleine Stück ist direkt proportional zur Beschleunigung. Diese kleine Bewegung wird wiederum kapazitiv gemessen und – in unserem Fall - analog ausgegeben.

Nun darf man sich diese Sensoren nicht als große, feinmechanische Bauteile vorstellen. Ich beschreibe viel mehr Sensoren, die in der sogenannten MEMS-Technologie gefertigt wurden. Das heißt, dass zur Fertigung der Feinmechanik Techniken zum Einsatz kamen, die normalerweise zur Fertigung von integrierten Schaltkreisen Verwendung finden. Das heißt, die Strukturen sind z.T. nur Mikrometer groß. Der Vorteil ist hier, dass die elektrischen Einrichtungen (kapazitive Wegsensoren, Auswertelektronik etc.) direkt mit auf das selbe Substrat aufgebracht werden können. Dadurch lassen sich die Sensoren und deren feinmechanische Komponenten in einem unglaublich kleinen Maßstab fertigen.

Zu den Problemen, die bei der Verarbeitung der Sensordaten auftreten, siehe auch 3.3.a.

3.2.d) Das Rauschproblem und dessen Lösung

Wie bereits in 3.2.c beschrieben, schwingen die Pendel des Gyros nicht immer exakt in ihre Ausgangsposition zurück. Das Resultat ist ein wandernder Nullpunkt und eine Verunreinigung des Signals mit weißem Rauschen.

Um dieses Problem in den Griff zu bekommen gibt es leider keine perfekte Lösung. Man kann das Problem nur eindämmen.

Dazu gibt es den sog. Kalman-Filter. Dieser Filter arbeitet auf stochastischer Basis. Er minimiert den „Quadratischen Fehler“. Leider ist der Kalman-Filter sehr komplex.

3.2.e) Lageregelung

Um den Eingangswert für den in 3.2.b beschriebenen Regelkreis zu errechnen, ist es wichtig, den Hubschrauber zunächst einmal am Boden zu „eichen“, d.h. die Abweichung (von der 0-Lage) zunächst auf 0° zu stellen. Mit diesen Grundwerten kann man bereits abheben. Kleinere Störungen, wie z.B. kleinere Windböen, lassen sich per Gyro ausregeln (Drehrate minimieren). Um aber eine wirklich exakte 0-Lage zu erreichen, muss der Hubschrauber über dem Startplatz in geringer Höhe in die Schwebelage gehen. Da die Ausrichtung noch nicht perfekt ist, steht der Hubschrauber etwas schräg und wird somit in irgendeine Richtung beschleunigt. Diese Beschleunigung wird aber von den Beschleunigungssensoren erfasst, und durch Veränderung der 0-Lage ausgeglichen. Im weiteren Flug muss der Drift der Gyro-Sensoren weiter kompensiert werden, was allerdings über Messungen an den Beschleunigungssensoren vorgenommen werden kann (gemessene Kurvenlage stimmt nicht mit gemessener Beschleunigung überein).

3.3) Systemdesign

3.3.a) Genereller Überblick

Das Steuerungssystem des Hubschraubers wurde als ein Netzwerk mit Sterntopografie aus mehreren, für ihre Aufgabe spezialisierten Knoten konzipiert. Dabei verfügt jeder Knoten über eine spezielle Aufgabe und dementsprechende Hardware. Alle Knoten sind über ein spezielles Protokoll miteinander verbunden, die Gesamtsteuerung ist das Resultat der Zusammenarbeit mehrerer Prozessoren (Knoten). Dinge wie z.B. die Navigation oder die Messung der Winkelabweichung auf der X-Achse übernehmen spezialisierte Knoten mit GPS und Gyro-Sensor.

Jeder Knoten hat dazu seine eigene Intelligenz, d.h. einen eigenen Mikroprozessor (siehe auch 3.3.b). Die Stromversorgung erfolgt mehrfach redundant durch mehreren Akkus (siehe auch 3.3.f).

3.3.b) Lageerkennungssystem

Der Hubschrauber verfügt über ein ausgeklügeltes Lageerkennungssystem (siehe auch 3.2.b 3.2.c 3.2.e). Es besteht aus 3 Gyromodulen, die jeweils einen Winkel pro Modul und Raumachse relativ zur 0-Lage messen. Jedes Modul arbeitet für sich und selbstständig, allerdings werden Informationen untereinander ausgetauscht und Ergebnisse verglichen. Des Weiteren lassen sich die Beschleunigungs-Rohdaten jedes Moduls von einem anderen Modul abrufen, so kann im Notfall ein Gyro ersetzt werden.

Das Lageerkennungssystem liefert Winkeldaten zur 0-Lage sowie Beschleunigungsdaten.

3.3.c) Warum viele kleine statt eines großen Prozessors?

Sollte man per Bus oder jeder mit einem kleinen Auto fahren? Jedes Vorgehen hat seine Vorteile. Der Bus, groß und geräumig, verbraucht wenig Sprit pro Person, und man benötigt nur einen Fahrer. Die Fahrtkosten sind also für alle besonders günstig. Wenn jetzt aber alle Fahrgäste an einen unterschiedlichen Zielort wollen, und das möglichst schnell, so ist der Bus denkbar ungeschickt, denn wenn er an jeder Bushaltestelle hält, benötigt er sehr lange. Hier sind viele kleine Autos effizienter. Es kommt darauf an, worauf man mehr Wert legt: Spritsparen bzw. die Kosten oder Geschwindigkeit.

In Falle des Hubschraubers war das Problem, möglichst viele unterschiedliche, spezielle Aufgaben möglichst schnell auszuführen. Hätte ich mich für einen großen, schnellen Prozessor entscheiden, müsste dieser einiges an Rechenleistung mitbringen. Leider ist es so, dass sich Prozessoren ab einer gewissen Größe nur noch mit Betriebssystem sinnvoll verwenden lassen. Würde ich allerdings ein Betriebssystem verwenden, würde dieses wiederum auf die Ausführungsgeschwindigkeit drücken. Dazu kommt, dass die meisten Betriebssysteme, die für Embedded-Systeme¹⁰ erhältlich sind (Linux, μ C-Linux, VX-Works etc.), sog. Multitasking-

¹⁰ Ein Embedded System ist ein Computer, der in einer Technischen Applikation eingebunden ist – oft wird auch der Ausdruck „Embedded Device“ verwendet.

Betriebssysteme sind. Das heißt, sie versuchen die Rechenzeit auf der CPU so aufzuteilen, dass mehrere Prozesse scheinbar gleichzeitig laufen. Das hört sich zwar toll an, aber de facto werden die verschiedenen Prozesse nur stückchenweise hintereinander ausgeführt. Dieses „Scheduling“ führt allerdings in den meisten Fällen¹¹ dazu, dass die Prozesse, die gerade laufen, nicht in Echtzeit auf Ereignisse reagieren können, sondern eben nur dann, wenn der CTS¹² gerade Zeit auf der CPU zur Verfügung stellt. Da aber ein autonomer Hubschrauber ein echtzeitbasiertes Computersystem quasi voraussetzt, ist ein zentraler, großer Rechner nicht die beste Wahl. Denn selbst ohne Betriebssystem und CTS müsste die CPU die einzelnen Aufgaben nacheinander ausführen. Man müsste die CPU dann sehr schnell machen...

Ich habe mich daher dazu entschieden, verschiedene Aufgaben verschiedenen, kleineren Prozessoren zuzuteilen, die für ihre jeweilige Aufgabe gerade genug Rechenleistung haben. Das hat nicht nur den Vorteil, dass die Ereignisse in Echtzeit verarbeitet werden können, sondern auch, dass alle verschiedenen Aufgaben (Messung der Winkel, Berechnung der Servostellungen etc. etc.) nicht nur scheinbar gleichzeitig sondern wirklich physikalisch gleichzeitig ablaufen. Man spricht hier auch von „realem Multitasking“. Des Weiteren muss spezielle Hardware, wie z.B. ein Analog-Digital-Wandler¹³ (ADC) nur an Knoten angeschlossen werden, die diese Hardware auch wirklich benötigt. Ein weiterer, nicht zu unterschätzender Vorteil ist, dass die Daten bereits vorverarbeitet beim nächsten Knoten ankommen, sodass dieser z.B. nicht mit dem Filtern von Gyro-Daten von seiner eigentlichen Tätigkeit abgehalten wird.

3.3.d) Kommunikations-Infrastruktur

Wie bereits in 3.3.a angesprochen, basiert das Computersystem des Hubschraubers auf Knoten. Diese Knoten sind alle über ein Netzwerk miteinander vernetzt.

11 Manche Betriebssysteme wie z.B. Linux können mit Realtime-Modulen im Kernel ausgeführt werden.

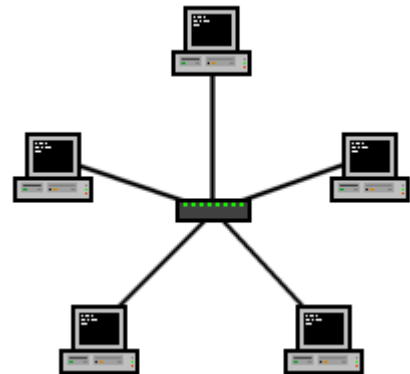
12 CTS: Command Time Scheduler – Programmteil des Betriebssystems, das die Ausführung (-szeit) der einzelnen Prozesse auf der CPU kontrolliert und steuert.

13 Ein Analog-Digital-Wandler (ADC) ist ein elektrotechnischer Baustein, der analoge Spannungen misst und digital in Form einer (binär-) Zahl wieder ausgibt.

Da das Kommunikationsnetzwerk des Hubschraubers ein wichtiges, wenn nicht das wichtigste System ist, muss es selbstverständlich ausfallsicher ausgelegt werden. Da am primären Netzwerk allerdings auch andere Peripherie vorgesehen ist, gibt es noch 2 weitere Sekundär/Reservesysteme. Alle Netzwerke tragen Namen, das primäre Netzwerk MAIN BUS A (MBA), das Reservesystem 1 MAIN BUS B (MBB), das 2. Reservesystem MAIN BUS C (MBC). Der MAIN BUS B (also das erste Reservesystem) arbeitet exakt gleich wie der MAIN BUS A (primäres Netzwerk), allerdings mit dem Unterschied, dass nur lebenswichtige Systeme (Gyro-Module, Servosteuerungen, Regelsystem etc., (siehe dazu auch 3.3.e)) angeschlossen sind. Das dritte Netzwerk (MAIN BUS C) arbeitet mit einem von Grund auf völlig anderen Protokoll. Während MBA und MBB mit einem dem RS232-Protokoll gleichen Protokoll zur Datenübertragung arbeiten, arbeitet MBC mit dem I²C-Protokoll. Während zur Zusammenführung der Knoten bei MBA / MBB spezielle Hardware (ASCP-Switch, siehe weiter unten) notwendig ist, kommt der MBC vollkommen ohne diese Hardware aus. Dies hat den Vorteil, dass der Hardwareteil des Netzwerkes tatsächlich nur aus Kabeln besteht, und somit nichts ausfallen kann.

MAIN BUS A und B sind in einer Stern-Topologie realisiert (siehe Abbildung 6).

Wie in Abbildung 6 dargestellt, sind bei einer Sterntopologie die einzelnen „Rechner“ mit einem Switch (Mitte) oder einem anderen Verbindungselement verbunden, welches den Datenfluss zwischen den einzelnen Rechnern steuert.



Der ASCP-Switch ist, wie in der heutigen *Abbildung 6: Stern-Topologie (WikiMedia Commons)* Ethernet-Netzwerktechnik auch, ein intelligentes

Gerät, das den Kommunikationsfluss gezielt steuert. Zwar wäre diese Flusststeuerung nicht unbedingt notwendig, aber der Vorteil wird deutlich,

wenn man die Informationsverarbeitung der einzelnen Knoten genauer betrachtet. Beim Empfang eines Datenpakets wird ein Interrupt ausgelöst, d.h. der Prozessor unterbricht seine aktuelle Aufgabe, um ein gewisses Ereignis (hier: Empfang eines ASCP-Pakets) zu bearbeiten. Dies ist im Normalfall nicht weiter schlimm, ist aber auch eine gewisse Gefahrenquelle. Ein Interrupt kann den Programmablauf eines Prozessors empfindlich stören, und so eine Kaskade an weiteren Fehlern auslösen. Man kann sich das wie in einer Fabrik vorstellen: Der erste Mitarbeiter stellt das Gehäuse her, der zweite Mitarbeiter baut das Innenleben ein, der Dritte schraubt das Gerät zu usw. Wenn einer der Mitarbeiter mit der Produktion nicht schritthalten kann, kann der gesamte Betrieb zum Erliegen kommen.

Da die Gyros z.B. ein Timing-empfindliches Programm ausführen (Realtime-Auswertung der Gyrodaten) sollten diese Interrupts auf ein Minimum reduziert werden. Ohne intelligente Datenverteilung würde bei jedem gesendeten Datenpaket bei jedem Knoten ein Interrupt ausgelöst, unabhängig davon, an welchen Knoten das Paket adressiert war.

Der ASCP-Switch hat mehrere Ein/Ausgänge (Interface-Ports), an die jeweils ein Knoten angeschlossen werden kann. Der Switch analysiert beim Systemstart das an ihn angeschlossene Netzwerk und erkennt so die angeschlossenen Knoten und merkt sich, welcher Knoten an dem jeweiligen Interface-Port liegt. Mit den so gewonnenen Daten kann der Switch Pakete direkt auf dem Interface-Port senden, der zur Empfängeradresse gehört, ohne das gesamte Netzwerk mithören zu lassen.

Allerdings bietet ein solches Gerät natürlich auch andere interessante Möglichkeiten. Man könnte z.B. einzelne Knoten aus dem Netzwerk ausschließen (Fehlerhafte Hardware, Programmabsturz o.ä.), „Kollisionen“ zwischen 2 Datenpaketen können zentral erkannt und behoben werden und nicht zuletzt lässt sich damit ermitteln, ob ein Knoten noch „lebt“ oder „tot“, sprich nicht ansprechbar, ist.

3.3.e) Kommunikationsprotokoll - Advanced Serial Communication Protocol

Das ASCP (Advanced Serial Communication Protocol) verwendet den EIA-232 Standard (früher: RS-232) um mit anderen Mikrocontrollern Informationen auszutauschen. Die Entscheidung fiel auf den EIA 232-Standard, weil die meisten heute verwendeten Mikrocontroller eine solche Schnittstelle bereits mitbringen. Der Standard sieht einen Pegel von -15V (1) und +15V (0) vor, der aber aufgrund der Bedingungen auf 3,3V-Pegel (TTL mit 3,3V) heruntergefahren wurde (so ist keine aufwändige Elektronik zur Pegelwandlung notwendig).

Der EIA-232 Standard beschreibt den physikalischen Teil des Protokolls („physical layer“), also den Teil, der sich mit dem Datentransfer selbst beschäftigt.

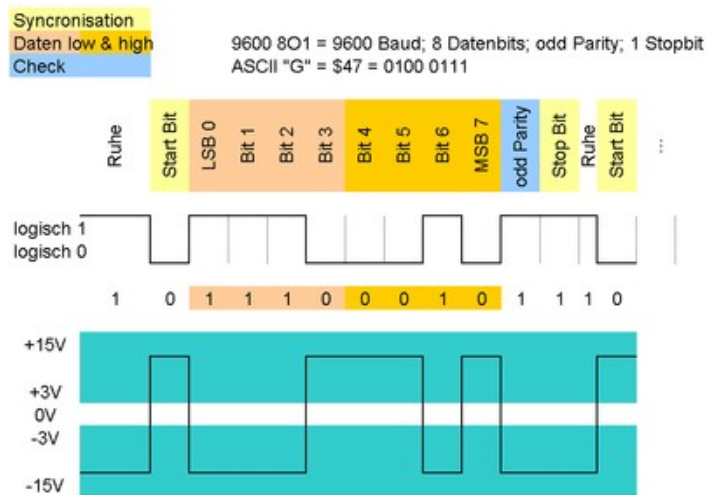


Abbildung 7: EIA 232 - Standard

Das ASCP-Protokoll schreibt vor, wie die einzelnen Bytes mittels des EIA 232 Standards versendet werden müssen:

Startsequenz / Header						*	Daten			*	Stopsequenz		*
Start	Senderadr.	Empfängeradr.	ID	Typ	Länge		DB0	...	DBn		Check	Ende	ACK/FAIL

Tabelle 1: Aufbau eines ASCP-Pakets

Das ASCP basiert auf früheren Erfahrungen, die ich mit der Kommunikation mehrere Mikroprozessoren gemacht habe.

Ein Paket beginnt mit der Startsequenz, einem bekannten Startbyte. Dieses Byte veranlasst den Empfänger auf „Empfang“ zu schalten, um bildlich zu sprechen. Anschließend folgt der Header-Teil, der die Adresse des Sendenden, die Adresse des Empfängers, eine temporär eindeutige Paket-ID, den Datentyp der im Paket enthaltenen Daten und die Länge des Datenteils

enthält. Der Empfänger quittiert den Header mit einem N/ACK-Byte (Acknowledge / Not Acknowledge). Dies dient dazu, dem Sendenden Feedback darüber zu senden, ob – kurz gesagt – die Übertragung zu diesem Zeitpunkt möglich ist. Wird der Header mit ACK (Acknowledge) bestätigt, werden die Datenbytes übertragen. Die Länge der Übertragung kann variieren (Länge-Byte im Header spezifiziert die zu übertragende Datenmenge) und die Übertragung kann vom Empfänger mit einem Stop-Byte abgebrochen werden. Auch der Datenblock wird mit einem N/ACK-Byte quittiert. Es folgt die Stoppssequenz. Diese enthält eine Prüfsumme des Datenteils und des Headers sowie ein Stopbyte. Der Empfänger berechnet nun die Prüfsumme selbst. Stimmen die Prüfsummen überein, antwortet der Empfänger mit ACK, wenn nicht, mit einem FAIL-Byte. Ein FAIL-Byte hat eine nochmalige Übertragung des Pakets zur Folge. Die Paket-ID gibt Referenz über das gesendete Paket. Somit können Pakete, bei denen beim Empfänger Probleme aufgetreten sind, beim Sender noch zugeordnet werden und im Zweifel nochmals übertragen werden, obwohl der Übertragungsvorgang schon beendet wurde.

3.3.f) Redundanz

Ein Systemausfall in einem autonom gesteuerten Hubschrauber kann fatale Folgen haben. Das Szenario wäre ähnlich dem, wenn ein Hubschrauberpilot ohnmächtig wird; einem Absturz. Bei einem autonomen Computersystem ist nochmals mehr Vorsicht geboten, da solch ein komplexes System im Normalfall einige „kritische Punkte“ hat. Es gilt also, das Versagen von Hard- oder Software so gut wie möglich abzufedern und im Notfall so Schlimmeres zu verhindern.

Es gibt bei meinem Hubschrauber mehrere Systeme, ohne die ein Betrieb des Hubschraubers nicht möglich wäre. Dazu gehören z.B. die Gyromodule (Lageerkennungssystem), die Servosteuerung, Lageregelung usw. Diese Systeme sind zusätzlich zum primären Kommunikationssystem über 2 getrennte Kommunikationssysteme (MBA, MBC, siehe 3.3.c) miteinander

verbunden, um einen Ausfall des primären oder sekundären Kommunikationssystems kompensieren zu können. Des Weiteren können diese Systeme über einen eigenen, nur für diese Systeme gedachten Backup-Akku mit Strom versorgt werden, selbst wenn alle anderen Akkus ausgefallen sind. Diese Maßnahmen garantieren die Kommunikation und die unterbrechungsfreie Spannungsversorgung der einzelnen (wichtigen) Komponenten.

Ein System, das mit besonderem Augenmerk auf die Ausfallsicherheit entwickelt wurde, ist das Lageerkennungssystem. Es besteht aus 3 Gyromodulen, die die Beschleunigung und die Drehrate in 3 Raumachsen messen. Fällt ein Gyro-Sensor aus, kann die genaue Raumneigung nicht mehr gemessen werden. Da eine doppelte Auslegung des Lageerkennungssystems den finanziellen Rahmen sprengen würde, habe ich einen anderen Ansatz zu dessen redundanten Auslegung gefunden: Jedes Gyromodul enthält einen 3-Achsen Beschleunigungssensor. Fällt ein Gyro aus, kann mittels der 2 verbleibenden Beschleunigungssensoren und deren bekannten Koordinaten relativ zum Hubschrauber für eine gewisse Zeit ein 3. Gyro ersetzt werden.

3.3.g) Softwarearchitektur

Die Software der einzelnen Knoten ist nach einem semi-ereignisorientierten System aufgebaut. Der Prozessor arbeitet auf Interrupt-Basis, d.h. kommt z.B. ein ASCP-Paket herein oder ein gewisses Zeitintervall ist abgelaufen, wird ein Interrupt ausgelöst. Je nach Art des Ereignisses, wird dieses in einer Ereignis-Warteschlange (low priority), oder (je nach Priorität) für „dringende Fälle“ direkt aus dem Interrupt heraus behandelt (high priority). Tritt kein Ereignis ein, arbeitet der Prozessor an Rechenzeit-intensiven Aufgaben. Als Beispiel möchte ich das Gyro-Modul anführen: Die Sensoren werden 1000-10000 mal pro Sekunde ausgelesen, das wird von einer Art internen Eieruhr (in Folgenden: Timer) gesteuert. Dieser Timer löst einen Interrupt aus, aus welchem (da hohe Priorität) die Daten der Sensoren

ausgelesen werden. Dies geschieht sehr zeitnah, da zwischen Interruptaufruf und -ausführung praktisch keine Zeit vergeht. Der Timer wird neu gestartet, und die gesammelten Daten zur Weiterverarbeitung bereitgelegt. Die Daten wurden ja bereits zeitnah gesammelt, damit ist das zeitkritische Element kein Problem mehr. Der Prozessor kann nun in aller Ruhe zwischen den Timer-Interrupts den komplizierten Kalman-Filter ausführen, der zum Ausgleich des Nullpunktdrifts sowie des Rauschens notwendig ist (siehe auch 3.2.d). So wird die Zeit des Prozessors optimal genutzt.

Diese Architektur hat den Vorteil, dass sie ohne komplexe Betriebssystem-Infrastruktur auskommt, und relativ einfach zu implementieren ist.

4) Umsetzung

4.1) Allgemeines

Wie der Umfang des Planungsteils zeigt, ist der Hubschrauber schon recht genau durchgeplant. Leider kam es zu einigen Verzögerungen, die das Projekt und dessen Realisierung entscheidend verlangsamten:

- Schwierigkeiten bei der ARM-Programmierung
- Verzögerungen beim Reflowlöten der Gyro- und Beschleunigungssensoren

Die ARM-Prozessorarchitektur war mir neu, und verlangte einiges an Einarbeitungszeit. Leider gab es relativ wenig Material, das ausreichend tief in die Materie einführt, sodass ich größtenteils per „try and error“-Verfahren arbeiten musste. Es gab dann noch weitere Probleme mit dem Anschluss an den Computer, über den der ARM programmiert wird. Auch hier war die Dokumentation relativ schlecht, und der Fehler musste per „try and error“ gefunden werden.

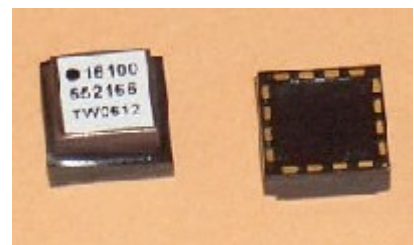


Abbildung 8: ADIS16100 Gyro im LGA-16 Gehäuse

Die Gyro- und Beschleunigungssensoren werden nur in einem speziellen Gehäuse ausgeliefert, das nur in einem speziellen Lötverfahren, dem sog. Reflowlöten auf eine Platine gelötet werden kann. Beim Reflowlöten wird das gesamte zu lötende Teil, also Platine mit Bauteilen in einen Hochtemperatur-Ofen gefahren, und nach einem speziellen Temperaturprofil auf bis zu 280°C erhitzt. Dadurch wird auch löten unter dem Bauteil ermöglicht, was in unserem Fall von Nöten war, denn die Bauteile haben ihre Kontakte auf der Unterseite, wo sie mit LötKolben nicht zu erreichen sind. Daher musste ich die Sensoren von einem Bekannten bei JetCat (eine Firma für Modellstrahltriebwerke) löten lassen. Dies zog sich leider ebenfalls in die Länge, was zu weiteren Verzögerungen führte, da ich die Sensoren nicht ansteuern konnte.

4.2) Gyro-Module

Derzeit stehen die Gyromodule kurz vor der Vollendung, die Prototypen sind bereits im Test. Wie bereits 3.2.c beschreiben, sind die Gyro-Module dazu da, die Drehrate und auftretende Beschleunigungen zu messen und zu verarbeiten.

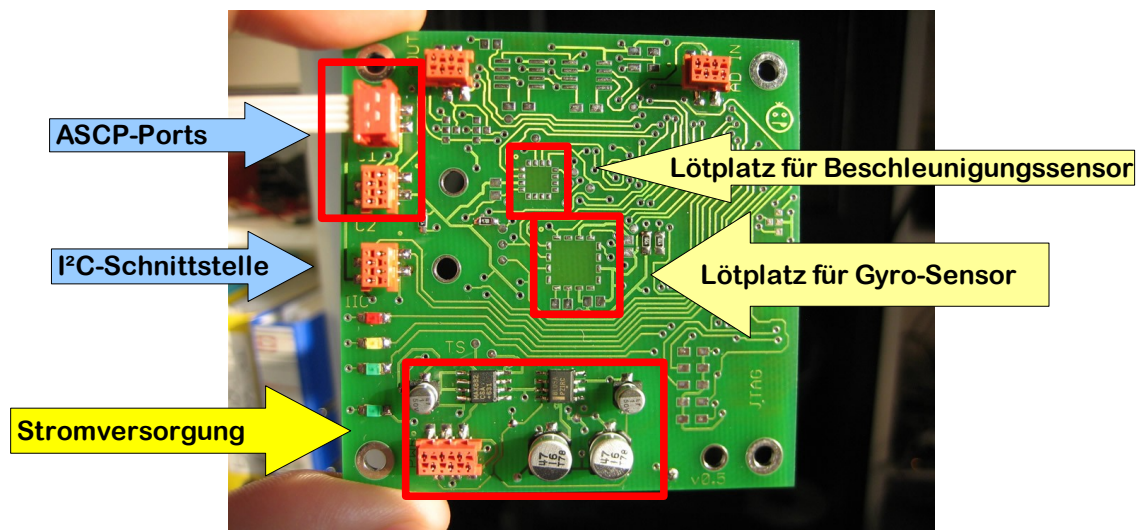


Abbildung 9: Gyro-Modul Top Layer (Prototyp, nicht alles bestückt)

Dazu haben die Module jeweils einen ADIS16100-Gyro, einen MMA7260-Beschleunigungssensor, sowie einen AT91SAM7S256-Prozessor plus zu deren Betrieb notwendige Hardware.

Das Modul beinhaltet alle nötige Hardware, um die Drehrate so exakt wie möglich zu messen. Es muss nur eine Spannung von 7,5V angelegt werden. Das System verfügt über 2 ASCP-Ports sowie einen I²C-Anschluss für die

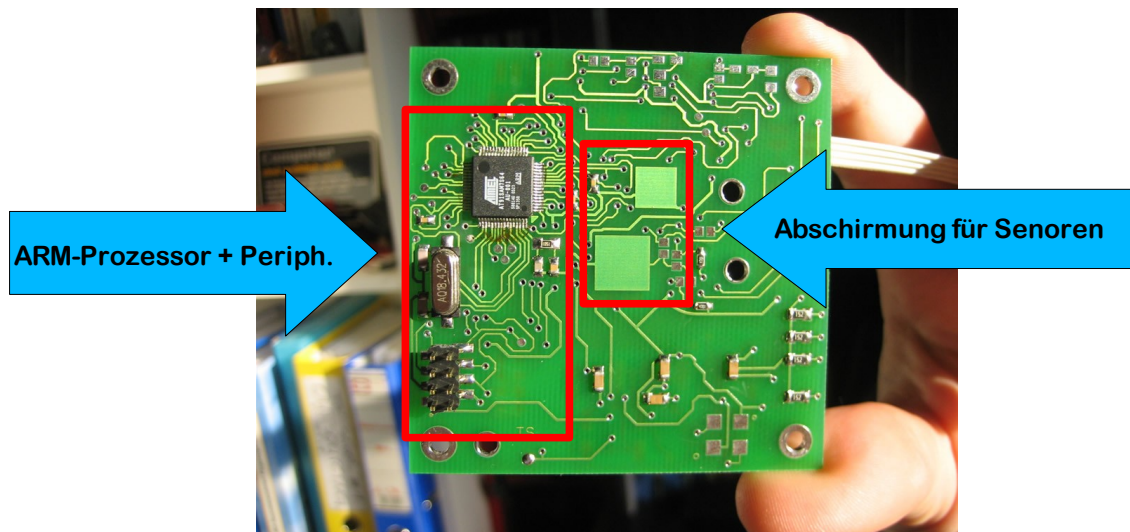


Abbildung 10: Gyro-Modul Bottom Layer (Prototyp nicht alles bestückt)

Verwendung am MAIN BUS C.

Die Verarbeitung der gemessenen Daten erfolgt über einen AT91SAM7S256-Prozessor der Firma Atmel, der über einen ARM7TDMI-Prozessorkern verfügt. Dieser Prozessor zeichnet sich einerseits durch seine 32Bit Busbreite, sowie durch seine RISC¹⁴-Architektur (ARM-Befehlssatz) aus. Der Prozessor kann auch auf einen THUMB-Befehlssatz umgeschaltet werden, der platzsparender (1 Wort / Befehl statt 2 Wort / Befehl wie beim ARM-Befehlssatz) im Speicher, dafür aber langsamer in der Ausführung ist. Der Prozessor wird mit 50MHz getaktet und hat 256kB internen FLASH-Speicher. Der Prozessor ist theoretisch voll Multithreading fähig und verfügt über eine Fülle an „On Chip Hardware“¹⁵, wie zum Beispiel 3 16Bit-Timer oder 2 USART-Tranceiver (ua. EIA-232). Des weiteren verfügt jede Platine noch über 2 hoch

¹⁴ RISC: Reduced Instruction Set Computing: Verzicht auf komplizierte Maschinenbefehle, sodass die Ausführungszeit eines Maschinenbefehls wesentlich höher ist wie bei einem CISC (Complex Instruction Set Computing) -Prozessor.

¹⁵ On Chip Hardware sind Peripheriebausteine, die mit dem Mikrocontroller selbst auf einem Chip (Die) sitzen.

präzise Messverstärker, um im Notfall (Gyroausfall, Messen der Drehrate über 2 Beschleunigungssensoren) Beschleunigungsdaten schon analog effektiv vorzuverarbeiten. Schaltplan und Blockschaltbild siehe Anhang.

5) Zusammenfassung

Schon historisch gesehen sind Hubschrauber ein interessantes Thema. Die Entwicklung vom nicht steuerbaren „fliegenden Fahrrad“ zu unseren heutigen Hubschraubern brauchte seine Zeit. Doch heute sind Hubschrauber aus dem täglichen Leben nicht mehr weg zu denken. Sie retten Leute aus Katastrophengebieten oder fliegen Schwerstverletzte ins Krankenhaus.

Aber auch bei anderen Anwendungen sind Hubschrauber optimal einsetzbar, zum Beispiel bei der Polizei. Dort werden sie zur Überwachung oder zur Verfolgung von Verbrechern sehr effektiv eingesetzt. Doch heute geht der Trend in diesem Sektor eher in Richtung der vollautonomen Hubschrauber, so genannter Drohnen oder UAVs. Dies sind selbstfliegende, entweder ferngesteuerte oder autonom agierende Flugkörper, die völlig ohne Pilot auskommen. Vermehrt kommen auch kleine Hubschrauber zum Einsatz, wie ich in dieser Arbeit einen geplant habe.

Mein Hubschrauber, bzw. mein UAV ist ein völlig autonomer Hubschrauber, der später anhand von GPS-Waypoints frei im Raum navigieren können soll. Aktuell befindet sich der Hubschrauber in der Hardware-Entwicklungsphase. Das heißt, zur Zeit werden die theoretischen Planungen in Elektronik „umgesetzt“. Später folgt noch die Softwarephase, in der der Hubschrauber mit den Programmen für den späteren Flug versorgt wird.

Das Computersystem des Helikopters ist dezentral organisiert, es gibt keinen Master, der den Datenfluss steuert. Das Netzwerk ist 3 fach redundant ausgelegt.

Während der Entwicklung des Konzepts und der Elektronik traten viele Probleme auf, und ich muss sagen, es ist eine schöne Herausforderung, diese Probleme zu lösen.

6) Danksagung

Zunächst möchte ich mich bei meinen Eltern bedanken, die dieses ganze Projekt erst möglich gemacht haben, ganz besonders meinem Vater, der die Idee selbst mitentwickelt hat. Dann wäre da Markus Zipperer vom der Cat-ING GmbH, der meine Platinen reflowgelötet hat. Des weiteren ist hier die Firma Analog Devices zu nennen, die mir für dieses Projekt 4 Gyros gespendet hat.

Vielen Dank!

7) Anhang

7.1) Quellenverzeichnis

- PDF01 : Marc Luthiger : Der Traum vom Fliegen und wie er verwirklicht wurde (Maturaarbeit), Kantonsschule Zug, 4. Nov. 2002, (Punkt 2.2) Seite 6f (real 7)
<http://ma.kanti-zug.ch/maturaprojects/Marc%20Luthiger%20-%20Der%20Traum%20vom%20Fliegen%20und%20wie%20er%20verwirklicht%20wurde.pdf> (im Internet vom 30.5.2008)
- WP01 : Wikipedia: Otto Lilienthal, unbekannter Autor,
http://de.wikipedia.org/wiki/Otto_Lilienthal (im Internet vom 20.5.2008)
- WP02 : Wikipedia: Leonardo da Vinci, unbekannter Autor,
http://de.wikipedia.org/wiki/Da_Vinci (im Internet vom 20.5.2008)
- WP03 : Wikipedia: Flugspirale, unbekannter Autor, <http://de.wikipedia.org/wiki/Flugspirale>
(im Internet vom 20.5.2008)
- WP04 : Wikipedia: Gebrüder Wright, unbekannter Autor, http://de.wikipedia.org/wiki/Gebr%C3%BCder_Wright (im Internet vom 20.5.2008)
- WP05 : Wikipedia: Paul Cornu, unbekannter Autor, http://de.wikipedia.org/wiki/Paul_Cornu
(im Internet vom 22.5.2008)
- WP06 : Wikipedia: Newtonsche Gesetze, unbekannter Autor,
http://de.wikipedia.org/wiki/Newtonsche_Axiome (im Internet vom 12.5.2008)
- WP07: Wikipedia: Drehimpuls, unbekannter Autor, <http://de.wikipedia.org/wiki/Drehimpuls>
(im Internet vom 11.5.2008)
- WP08: Wikipedia: Kreiselkompass, unbekannter Autor,
<http://de.wikipedia.org/wiki/Gyro> (im Internet vom 11.5.2008)
- WP09: Wikipedia: EIA-232, unbekannter Autor,
<http://de.wikipedia.org/wiki/EIA-232> (im Internet vom 1.6.2008)
- WP10: Wikipedia: Topologie (Rechnernetz) Stern-Topologie, unbekannter Autor,
http://de.wikipedia.org/wiki/Topologie_%28Rechnernetz%29#Stern-Topologie (im Internet vom 2.6.2008)
- WP11: Wikipedia: Koaxialrotor, unbekannter Autor,
<http://de.wikipedia.org/wiki/Koaxialrotor> (im Internet vom 7.6.2008)
- PIC01: Gerald Wittling, über den Modellhubschrauber ECO-8 (Bilder)
<http://www.geraldwittling.de/ECO.htm> (im Internet vom 2.6.2008)

PIC02: US-Africa Materials Institute and Princeton University, Princeton, NJ 08544-5263, MEMS, <http://usami.princeton.edu/programs/mems.shtml> (im Internet vom 3.6.2008)

EXT01 : „Helikopter Fliegerei, Hubschrauber Geschichte, Technik und Aerodynamik“, unbekannter Autor und Herausgeber, http://www.hubschrauber.li/geschichte/sowars/mai_ges_beg.htm (im Internet vom 23.5.2008)

7.2) Abbildungsverzeichnis

• Abbildungen, Zeichnungen, Tabellen

Abbildung 1: Leonardo da Vincis Flugspirale.....	3
Abbildung 2: Gyrocopter - Helikopter.....	6
Abbildung 3: Hubschrauber mit Koaxial-Rotor der Heliswiss.....	6
Abbildung 4: Taumelscheibe eines Modellhubschraubers (Ikarus).....	7
Abbildung 5: MEMS-Beschleunigungssensor: Ecke der trägen Masse (links und rechts: kapazitive Wegsensoren).....	13
Abbildung 6: Stern-Topologie (WikiMedia Commons).....	18
Abbildung 7: EIA 232 - Standard.....	20
Abbildung 8: ADIS16100 Gyro im LGA-16 Gehäuse.....	24
Abbildung 9: Gyro-Modul Top Layer (Prototyp, nicht alles bestückt).....	24
Abbildung 10: Gyro-Modul Bottom Layer (Prototyp nicht alles bestückt).....	25
Zeichnung 1: Hubschrauber Regelkreis.....	11
Tabelle 1: Aufbau eines ASCP-Pakets.....	20

7.3) Schaltplan und Blockschaftbild des Gyromoduls

7.3.a) Blockschaftbild

