



HOCHSCHULE DER MEDIEN, STUTTGART

143601

INNOVATIONSPROJEKT

Untersuchungen zur Realisierung eines Flugschreibers für unbemannte Flugobjekte im Modellflug

Autoren:

Lang, Christian
Miederer, Jonas
Kopp, Martin

Matrikelnummern:

32721
32723
32720

Zusammenfassung

Diese Projektdokumentation beschreibt die Erkenntnisse und Erfahrungen bei der Durchführung des Innovationsprojekts zur Untersuchung der Realisierung eines Flugschreibers für unbemannte Flugobjekte im Modellflug.

Aufgrund mangelhafter Qualität sowie Lieferschwierigkeiten bei der Bestellung einer Versuchs-Drohne konnte das Projekt nicht wie geplant durchgeführt werden. Hierdurch sind zum einen die verfügbaren Mittel stark eingeschränkt worden und zum anderen Verzögerungen entstanden, welche schließlich auch zu einem verzögerten Projektabschluss geführt haben. Trotz dieser Einschränkungen, die direkten Einfluss auf den Projektverlauf genommen haben, liefert dieses Dokument Anknüpfungspunkte und Anregungen für zukünftige Projekte und Ideen. Die Vielzahl an Anforderungen gegenüber Hard- und Software eines solchen Produkts bieten unzählige Themengebiete zur weiteren Forschung.

Neben einer Marktanalyse wird vor allem der Frage nach einer zuverlässigen und autarken Energieversorgung nachgegangen und in mehreren Feldtests untersucht. Hierbei liegt der Fokus auf einem Energy-Harvesting-Konzept, welches verschiedene Kopplungen zur Energiegewinnung zwischen Motor und Flugschreiber untersucht. Als weiterer Schwerpunkt wird die Datenhaltung des Flugschreibers untersucht. Hierbei werden bestehende Technologien, wie beispielsweise die Trusted-Computing-Plattform betrachtet, um mithilfe dieser ein zuverlässiges Modell zur Datenhaltung zu konzipieren und anhand verschiedener Sicherheitsziele die zu erfassenden Flugdaten zu kategorisieren. Bei der Positionsbestimmung wird auf die besonderen Eigenschaften des GPS-Standards und auf die Versuche mit verschiedenen GPS-Modulen eingegangen. Mithilfe eines Mikrocontrollers wurde ein Prototyp entwickelt, der die empfangenen GPS-Daten auswertet, signiert und auf ein Speichermedium loggt. Da die Umsetzung jedoch auch hier durch die Einschränkungen der verfügbaren Hardware beeinträchtigt wurde, wurde daher auch der Hauptfokus auf die theoretischen Grundlagen des *Global Positioning Systems* gelegt.

Inhaltsverzeichnis

1 Einleitung	4
2 Marktanalyse	6
2.1 Produkte	6
2.2 Standards	8
3 Anforderungen an den Flugschreiber	10
3.1 Physische Anforderungen	10
3.2 Funktionale Anforderungen	11
3.3 Rechtliche Anforderungen	11
3.3.1 Deutschland	12
3.3.2 Europa	12
4 Externe Abhängigkeiten	14
4.1 Stromversorgung	14
4.1.1 Batterie	14
4.1.2 Stromkreis der Drohne	14
4.1.3 Energy-Harvesting	15
4.1.4 Ausblick	15
4.2 Datenhaltung	15
4.2.1 Offline	16
4.2.2 Online	16
4.2.3 Ausblick	17
5 Datenhaltung	18
5.1 Sicherheitsziele	18
5.2 Technologien	19
5.2.1 Einweg-Funktion	20
5.2.2 Symmetrische Verschlüsselung	20
5.2.3 Asymmetrische Verschlüsselung	20
5.2.4 Digitale Signatur	20
5.2.5 Digitales Zertifikat	21
5.2.6 Trusted-Computing-Plattform	21
5.3 Systemarchitektur des Flugschreibers	23
6 Feldtests	26
6.1 Energy Harvesting	26
6.1.1 Testaufbau	26

6.1.2	Induktive Kopplung	28
6.1.3	Mechanische Kopplung	30
6.1.4	Batterie	32
6.1.5	Ergebnis	34
6.2	GPS	36
6.2.1	Theoretische Grundlagen	36
6.2.2	GPS im Kontext des Flugschreibers	41
7	Schlussbetrachtung	53
7.1	Fazit	53
7.2	Ausblick	54

1 Einleitung

Die Zahl unbemannter Flugobjekte (*v.a. Drohnen*) hat in den letzten Jahren rasant zugenommen. Zum einen ist dies auf die immer marktreiferen Modelle zurückzuführen, zum anderen erleben die Geräte aufgrund der Erschließung des Marktes durch eine Vielzahl neuer Hersteller einen deutlichen Preisverfall. Es wird geschätzt, dass bis 2020 ca. 1,2 Millionen Drohnen bundesweit im Besitz von Privatpersonen sein sollen¹. Dieser Boom birgt jedoch insbesondere bei fahrlässigem Einsatz von Drohnen einige Sicherheitsrisiken, wovon eine Gefährdung von Privatpersonen bis hin zur Luftfahrt ausgehen kann.

Die zunehmenden Risiken werden vor allem deutlich bei Zwischenfällen wie der Beinahe-Kollision einer privaten Drohne mit einer Lufthansa-Maschine im Münchner Luftraum im August 2016. Dabei näherte sich diese während des Landeanflugs des Flugzeugs in 1700 Metern Höhe bis auf wenige Meter gefährlich nahe². Eine Kollision blieb zum Glück aller Beteiligten jedoch aus. Eine Regulierung durch den Gesetzgeber erscheint daher als sinnvoll. Der deutsche Gesetzgeber reagiert auf diese Risiken beispielsweise mit einem Maximalgewicht der Fluggeräte, maximal erlaubter Flughöhe, Flugverbotszonen und der Verpflichtung aller Hobbypiloten zu einer Haftierhaftpflichtversicherung, die für Schäden aufkommt, welche durch den Betrieb von Fluggeräten entstehen. Zusätzlich gelten in den meisten europäischen Ländern strikte Regelungen beim Betrieb von Fluggeräten. Diese können, wie etwa in Schweden, sogar bis zu einem generellen Verbot von Kameradrohnen reichen.

Ähnlich des Modells eines Fahrtenschreibers, bei dem die Lenk- und Ruhezeiten von LKW-Fahrern überprüft werden, gibt es derzeit keine Technologie, welche die Einhaltung der Regeln und Gesetze im privaten Drohnenverkehr dokumentiert. So kann ein Pilot trotz des Verbots seine Drohne beispielsweise in Flugverbotszonen einsetzen und entsprechenden Sanktionen durch den Gesetzgeber entgehen.

Dieses Innovationsprojekt entstand aus der Idee heraus, den Gesetzgeber sowie Versicherungen bei der Unfalluntersuchung zu unterstützen, um gesicherte Informationen über einen Unfallhergang mit Fluggeräten in Erfahrung bringen zu können. Als Vorbild dienten hierfür Flugdatenschreiber aus der Luftfahrt. Diese speichern über einen längeren Zeitraum verschiedene Flugdaten und ermöglichen im Falle eines Absturzes genaue Analysen, um auf die Ursache schließen zu können.

Diese Ausarbeitung beschäftigt sich mit der Fragestellung, welche Anforderungen an einen Flugschreiber für unbemannte Flugobjekte gestellt werden müssen, anhand

¹Marktforschung: Von wegen 400.000 Drohnen in der Luft (Link)

²Drohne kommt Lufthansa-Airbus gefährlich nahe (Link)

derer eine System-Architektur zur sicheren Datenhaltung konzipiert wurde. Zuvor gibt sie einen kurzen Einblick in die aktuelle Marktsituation und stellt Produkte und deren Technologien zur Standortermittlung vor. Es wird die Erfassung von Standortdaten mithilfe des GPS-Standards erläutert und auf die Besonderheiten des Standards eingegangen. Über ein breites Spektrum an Feldversuchen werden hierbei verschiedene GPS-Module getestet und die empfangenen Informationen ausgewertet und weiterverarbeitet. Außerdem wurden verschiedene Konzepte zur Energieversorgung des Flugschreibers erarbeitet und deren Effektivität anhand einer Vielzahl von Feldversuchen bewertet.

Das Ziel des Projekts ist es, die Anforderungen an einen Flugdatenschreiber für Drohnen zu erfassen und Konzepte zur Umsetzung zu erarbeiten. Außerdem sollen einige Technologien anhand von Feldversuchen untersucht werden, um im Anschluss eine Empfehlung bei der Konzeption eines Flugdatenschreibers für Drohnen aussprechen zu können.

2 Marktanalyse

Mit der steigenden Nachfrage nach Drohnen nimmt auch der Bedarf an Hardware zum Tracken dieser Flugobjekte zu. Auf dem Markt finden sich autarke Lösungen in Form von separater Hardware von verschiedenen Herstellern sowie herstellerübergreifende Kommunikations-Standards zur Erfassung von Sensordaten und Steuerung der Drohne.

2.1 Produkte

Die bestehenden Lösungen können eingeteilt werden in Hardware zum *Offline*- oder *Online*-Tracking. Produkte diverser Hersteller wie *Flytrex*, *Trackimo* (siehe Abb. 1) oder *TK102* setzen dabei zum Teil auf die Übertragung der aktuellen Position über das Mobilfunknetz. Dies setzt jedoch sowohl eine SIM-Karte, als auch eine ausreichende Netzabdeckung voraus.



Abbildung 1: GPS-Tracker des Herstellers *Trackimo*

Quelle:

<https://buytrackimo.myshopify.com/products/3g-trackimo-gps-tracker>

Der *Trackimo Universal GPS Tracker* aus Abbildung 1 eignet sich beispielsweise sehr gut zur Echtzeitortung. Der Hersteller bietet zur Standortermittlung einen Internetdienst sowie Smartphone-Apps an. Im Gegensatz zu vielen anderen vergleichbaren Produkten ist in der Hardware bereits eine SIM-Karte integriert, welche ein begrenztes Kontingent an Datenvolumen bereithält, um die erfassten Standortdaten an das Backend des Herstellers zu übertragen. Ein mögliches Datenroaming ist ebenfalls im Datenvolumen enthalten, muss aber jährlich gegen eine Gebühr verlängert werden. Außerdem wird die Position zusätzlich zum GPS-Empfänger mittels Triangulation über das Mobilfunknetz ermittelt. Das Gewicht beträgt 42 Gramm bei einer Größe von $4,0 \times 4,7 \times 1,7$ Zentimeter. Die Stromversorgung erfolgt über einen Akku

2.1

mit 600 mAh bei 2,2 Wh und hält laut Herstellerangaben zwischen 2 bis 7 Tage abhängig der Empfangsqualität.

Im Gegensatz hierzu legt das Produkt *LightBug* den Fokus auf eine autarke Stromversorgung und verwendet kleine Solarzellen (vgl. Abbildung 2), um genügend Strom zu erzeugen, der den GPS-Tracker mit Energie versorgt.



Abbildung 2: LightBug angebracht an einem Schlüsselbund
Quelle: <https://thelightbug.com/>

Der Akku bietet hierfür eine Kapazität von 800 mAh und erlaubt eine periodische Übertragung der Daten mit einem Abstand von 15 Minuten über eine Dauer von bis zu acht Tagen ohne Sonnenlicht. Die Übertragung erfolgt ebenfalls mittels einer integrierten SIM-Karte über das Mobilfunknetz. Die Größe mit $5,3 \times 2,7 \times 1,5$ Zentimeter ähnelt der des *Trackimo Universal GPS Trackers*. Zusätzlich bietet der Empfänger außerdem eine Bluetooth Low Energy-Schnittstelle.

Ein vollkommen anderes Konzept verfolgt der Hersteller *TrackR* mit seinen gleichnamigen Produkten. Das Produktpportfolio bietet einen extrem kompakten GPS-Tracker, welcher mit der Größe von $2,6 \times 2,6 \times 0,6$ Zentimeter in etwa die Größe einer 2-Euro-Münze erreicht (vgl. Abbildung 3).



Abbildung 3: TrackR pixel
Quelle: <https://store.thetrackr.com/pixel/>

Im Gegensatz zu den bereits vorgestellten Produkten setzt *TrackR* auf ein crowd-sourced Netzwerk. Hierbei stellen Nutzer des Systems bei Reichweite eine Verbindung mit dem Tracker über ihr Smartphone her und teilen mithilfe der Sensoren des

Smartphones dem Netzwerk ihre Position mit. Verwendet wird hierfür der Bluetooth Low Energy-Standard. Über diesen ermöglicht der Tracker außerdem eine Distanzangabe im Nahumfeld darüber, wie weit der Tracker vom Smartphone des Besitzers entfernt ist. Es besteht zudem die Möglichkeit den Nutzer beispielsweise zu benachrichtigen, wenn sich der Tracker außerhalb seiner Reichweite befindet. Der Tracker enthält keine weitere Hardware zur Positionsbestimmung und verzichtet gezielt auf GPS-Empfänger und ein SIM-Karten-Modul. Daher genügt als Energieversorgung eine Knopfzelle des Typs CR1620. Der Hersteller wirbt mit einer Laufzeit von bis zu einem Jahr.

2.2 Standards

Um einen Flugschreiber für unbemannte Flugobjekte zu entwickeln, sind insbesondere die genauen Flug- und Positionsdaten von großer Relevanz. Um nicht nur die externen Zustände, wie etwa die Position des Flugobjekts, bestimmen zu können, sondern auch interne Daten wie Ladekapazität, Zustand des Motors usw. abzugreifen, ist eine unmittelbare Datenkommunikation mit dem Fluggerät notwendig.

Einige Hersteller machen bestimmte Softwareschnittstellen ihrer Drohnen für Entwickler zugänglich. Auf diese Weise ist ein einfacher und direkter Datenaustausch mit dem Flugobjekt möglich. Jedoch bestand die Grundidee dieses Projektes darin, den Flugschreiber so universell und allgemeinnutzbar wie möglich zu entwickeln, sodass der Flugschreiber im einfachsten Fall nach einer kurzen und einfachen Montage mit einer Vielzahl von Drohnen genutzt werden kann. Dies erfordert jedoch eine universelle Bereitstellung einer einheitlichen Schnittstelle, die unabhängig vom Hersteller oder Modell des Gerätes genutzt werden kann.

In der Open-Source Gemeinde kursieren verschiedene Lösungen zur herstellerübergreifenden Programmierung und Kommunikation mit Drohnen.

Das *UAVCAN*³⁴ ist einer dieser Lösungen. Es basiert auf dem aus der Automobilindustrie bekannten CAN-Bus und erlaubt so einen effizienten und vielseitigen Austausch kleiner und großer Datenmengen. Das *UAVCAN*-Protokoll wurde explizit für Anwendungen aus der Luft- und Raumfahrt sowie der Robotik entwickelt. Aufgrund der guten und umfangreichen Dokumentation, der in mehreren Programmiersprachen zur Verfügung gestellten Bibliotheken und des ebenfalls frei-verfügbarren GUI-Tools bietet sich eine Nutzung dieses Protokolls an.

Ein weiteres vielversprechendes Protokoll ist das *MAVLink Micro Air Vehicle Com-*

³<http://uavcan.org/>

⁴<https://github.com/uavcan>

*munication Protocol*⁵. Dieses ist im Gegensatz zum *UAVCAN*-Protokoll explizit für *unmanned Micro-Air-Vehicles*, also kleine unbemannte Flugobjekte entwickelt worden. Dazu gehören vor allem Modellflugzeuge oder auch leichtgewichtige Drohnen. Auch hier wird eine ausführliche Dokumentation mitsamt Beispielen zur Verfügung gestellt, was eine Implementierung stark vereinfacht.

Aufbauend auf den sehr hardwarenahen Protokollen gibt es auch verschiedene Plattformen, die weitere softwarebasierte Lösungen anbieten, um die Kommunikation zu vereinfachen.

Dazu gehört beispielsweise das weitverbreitete Open-Source-Projekt *Dronecode*⁶. Hier finden sich unter anderem SDKs und APIs für den Datenaustausch mit unbemannten Luftfahrzeugen, die von einer Community entwickelt und bereitgestellt werden.

Einen ähnlichen Zweck verfolgt auch *Dronekit*⁷. Hier stehen neben den SDKs für unterschiedliche Programmiersprachen auch High-Level-Funktionalitäten wie intelligente Routenplanung oder autonome Flüge zur Verfügung.

Es ist also ersichtlich, dass innerhalb der Open-Source-Community durchaus vielversprechende und auch professionelle Standards, Schnittstellen und Development-Kits entwickelt werden, die sich im Sinne eines möglichst vielseitig-einsetzbaren und herstellerunabhängigen Flugschreibers für den praktischen Einsatz anbieten würden. Dennoch scheitert diese Idee an einer grundsätzlichen Voraussetzung: Die Hersteller der Flugobjekte müssen eine Möglichkeit zur Nutzung offener Standards schaffen. Dies wird jedoch derzeit nur von sehr wenigen Herstellern umgesetzt. Bei den meisten erfolgt die Kommunikation über proprietäre Protokolle, sodass ein Auslesen der internen Daten nicht oder nur mit erheblichem Aufwand möglich ist.

Aus diesem Grund wurde im Rahmen dieses Projekts darauf verzichtet, eine direkte Kommunikation mit der Drohne zu etablieren. Vielmehr wurde der Fokus auf die Stromversorgung, Datenhaltung und Nutzung externer Fluginformationen (GPS) gelegt.

⁵<http://qgroundcontrol.org/mavlink/start>

⁶<https://www.dronecode.org/>

⁷<http://dronekit.io/>

3 Anforderungen an den Flugschreiber

Ein Flugschreiber wird Extremsituationen ausgesetzt. Diese können beispielsweise bedingt durch Wettereinflüsse oder Abstürze auftreten. Während die Hardware des Flugobjekts irreversiblen Schaden erleiden kann, sollte die Funktionsfähigkeit des Flugschreibers, mindestens jedoch der Zugriff auf die gespeicherten Daten, gewährleistet bleiben. Daher wurden die folgenden Anforderungen definiert, welche zu Teilen im Rahmen des Projekts weiter betrachtet werden.

3.1 Physische Anforderungen

Gewicht	Da das maximal erlaubte Gewicht an Zuladung der gängigen Drohnen stark limitiert ist, spielt das zusätzliche Gewicht des Flugschreibers eine entscheidende Rolle
Form	Die Form spielt hinsichtlich der Gewichtsverteilung eine Rolle. Eine möglichst flache Form mit ausgewogener Gewichtsverteilung bietet sich besonders an.
Aerodynamik	Besonders bei hohen Geschwindigkeiten spielt die Aerodynamik der Drohne und somit auch des Flugschreibers eine Rolle.
Größe	In Zusammenspiel mit dem Gewicht ist eine Bauform möglichst geringer Größe erstrebenswert, um die Aerodynamik sowie das Flugverhalten minimal zu beeinflussen.
Vibrationsresistenz	Die Hardware muss resistent gegen Vibrationen unterschiedlicher Art sein. Dabei können gleichmäßig konstante Vibrationen durch die hohe Drehzahl der Rotoren und die auftretenden Luftströme während des Fluges, sowie starke Erschütterungen bei der Landung oder eines Absturzes auftreten.
Temperaturresistenz	Im Idealfall ist der Flugschreiber resistent gegen besonders hohe Temperaturen (im Falle eines Brands) und resistent gegen niedrige Temperaturen (längere Zeit im Schnee oder Flug bei Minusgraden).

3.3.1

Wasserdichtigkeit	Der gesamte Flugschreiber (mindestens aber die Speichereinheit) sollte gegen eindringendes Wasser und Feuchtigkeit bis zu einem Druck von einem Bar (bis 10 Meter Wassertiefe) geschützt sein.
Deformationsresistenz	Einwirkender Druck (z.B. durch ein überfahrendes Auto) darf der Hardware und vor allem der Speichereinheit keinen Schaden zuführen.

3.2 Funktionale Anforderungen

Manipulationssicherheit	Die erfassten Messwerte müssen manipulationssicher gespeichert werden. Hierzu muss ein System eingesetzt werden, welches die erfassten Daten beispielsweise signiert, um Manipulationen erkennbar oder im Idealfall unmöglich zu machen.
Erfassung der Messwerte	Der Flugschreiber ist in der Lage verschiedene Sensoren, mindestens jedoch ein Sensor zur Erfassung des Standorts, zu lesen und deren Werte zu speichern. Die Messwerte der Sensoren müssen ein Mindestmaß an Präzision aufweisen.
Ausfallsicherheit	Das Hardware- und Softwaredesign soll maximal resilient gegen Ausfälle sein. Dies schließt Basisfunktionalitäten wie die Stromversorgung und komplexe Merkmale wie die Ansteuerung beim Auslesen von Sensoren mit ein.

3.3 Rechtliche Anforderungen

Durch die steigenden Absatzzahlen von Drohnen im Endkunden-Segment befinden sich immer mehr unbemannte Flugobjekte in der Luft. Die Gesetzgeber weltweit reagieren hierauf mittels neuer Verordnungen und Gesetze. Im Folgenden wird die Situation in Europa, mit Schwerpunkt in Deutschland, erläutert.

3.3.2

3.3.1 Deutschland

Am 07. April 2017 ist in Deutschland eine Drohnen-Verordnung in Kraft getreten. Diese umfasst zahlreiche Vorschriften und Regelungen beim Umgang und Betrieb unbemannter Flugobjekte, welche vor allem nach dem Gesamtgewicht des Objekts gestaffelt sind. Neben der bereits bestehenden Verpflichtung zum Abschluss einer Drohnen-Haftpflichtversicherung, besteht neuerdings generell eine Kennzeichnungspflicht der Flugobjekte. Diese muss feuerfest sein und den Namen sowie die Adresse des Besitzers enthalten. Ab einem Gesamtgewicht von mehr als zwei Kilogramm wird auf der Seite des Piloten ein Flugkundenachweis vorausgesetzt. Alle Flugobjekte mit einem Gesamtgewicht von über fünf Kilogramm benötigen zusätzlich zu den bereits genannten Verpflichtungen eine Aufstiegserlaubnis durch die verantwortliche Luftfahrtbehörde. Drohnen mit einem Gewicht über 25 Kilogramm sind generell verboten. Die Regelungen zur Kennzeichnungspflicht sowie zum Flugkundenachweis treten zum 01. Oktober 2017 in Kraft. Außerdem wurden Vorschriften zur maximalen Flughöhe (100 Meter), Sichtkontakt (dauerhaft), Nachtflüge (Aufstiegserlaubnis benötigt) und weitere Regelungen definiert, wie beispielsweise ein Flugverbot über Flugplätze, Industrieanlagen und Wohngrundstücke. Eine Regelung zur Erfassung der zurückgelegten Flugstrecke besteht nicht.

3.3.2 Europa

Im Vergleich zu Deutschland liegt der Schwerpunkt in Österreich nicht in der Staffelung der Drohnen nach Gewicht, sondern in der Kategorisierung anhand des zu überfliegenden Gebietes und ob das Flugobjekt in der Lage ist Bildmaterial zu erfassen. In Abhängigkeit der zugeordneten Kategorie von A bis D und des Gewichts finden unterschiedliche Vorschriften und Regelungen Anwendung. Diese reichen von einer Betriebsbewilligung und Versicherungspflicht bis hin zu einem Lärmessbericht und einer Betriebssicherheitsanalyse.

In der Schweiz gilt für Drohnen unter 30 Kilogramm eine Verordnung, welche den Einsatz unbemannter Flugobjekte regelt. Diese schreibt eine Versicherungspflicht ab 500 Gramm vor und erlaubt den Betrieb von Drohnen bis 30 Kilogramm ohne vorherige Genehmigung. Neben der Einhaltung von Sicherheitsabständen zu beispielsweise Flughäfen und Menschenansammlungen sowie der maximalen Flughöhe von 150 Metern, sind die Vorschriften, im direkten Vergleich mit Deutschland und Österreich, weniger restriktiv gehalten.

Innerhalb Europas finden länderspezifisch unterschiedliche Verordnungen und Regelungen anwendung, welche jedoch meist ebenfalls anhand von Gewicht oder Einsatzort kategorisiert werden. Schweden bildet hierbei eine Ausnahme. Dort gelten

3.3.2

besonders strenge Regelungen für Kamera-Drohnen, welche diese prinzipiell flächen-deckend verbieten.

Eine Vorschrift zur Erfassung der zurückgelegten Flugstrecke besteht zum Zeitpunkt des Projekts in keinem europäischen Land. Durch die Anforderungen wird aber deutlich, dass das Gesamtgewicht eine entscheidende Rolle bei der Klassifizierung spielt. Außerdem rückt die Haftbarkeit durch die Kennzeichnungspflicht und Verpflichtung zur Haftpflichtversicherung weiter in den Fokus.

4 Externe Abhangigkeiten

Beim Entwurf eines Flugschreibers mussen verschiedene externe Abhangigkeiten betrachtet werden. Im Folgenden werden die Moglichkeiten zur Stromversorgung erlautert, sowie die Optionen zur Datenubermittlung gepruft.

4.1 Stromversorgung

Zur wichtigsten externen Abhangigkeit zahlt die Stromversorgung des Flugschreibers. Diese kann mithilfe verschiedener Ansatzen erfolgen. Neben einer eigenen Energiequelle in Form einer Batterie wird ebenfalls die Moglichkeit einer autarken Energieversorgung naher betrachtet. Dies umfasst neben den Anschluss des Flugschreibers an den bestehenden Stromkreis der Drohne auerdem das Prinzip des *Energy-Harvestings*.

4.1.1 Batterie

Eine triviale Losung zur Stromversorgung des Flugschreibers kann mithilfe einer Batterie erreicht werden. Abhangig von der benotigten Leistung kann diese dimensioniert werden. Der Vorteil liegt in der Einfachheit der Losung und dem geringen zusatzlichen Hardware-Umfang. Der Nachteil liegt in der eingeschrankten Laufzeit des Flugschreibers, da eine Batterie regelmaig getauscht oder geladen werden muss. Hierdurch wurde eine hohe Unzuverlassigkeit entstehen, da der Ladezustand des Flugschreibers vor dem Flug gepruft und angeschalten sowie nach dem Flug wieder abgeschalten werden musste. Der Ablauf hierbei ist auf Seiten des Benutzers fehleranfellig, wodurch das Risiko maximiert wird, dass einige Fluge nicht aufgezeichnet werden.

4.1.2 Stromkreis der Drohne

Als Alternative kann der Flugschreiber direkt an den Stromkreis der Drohne angeschlossen werden. Dies ermoglicht eine dauerhafte Stromversorgung und eine Erhohung der Zuverlassigkeit, dass der Flugschreiber bei Betrieb der Drohne ebenfalls in Betrieb ist. Ein groer Nachteil besteht in der Notwendigkeit das Gehause sowie den Stromkreis der Drohne zu offnen, um einen weiteren Verbraucher anzuschlieen. Dies birgt ein Risiko bei einer moglichen spateren Garantieabwicklung gegenber des Herstellers, indem dieser die Garantie nicht mehr anerkennt, da sich das Gerat in keinem

4.2.1

unversehrten Zustand mehr befindet. Ebenfalls können durch, nicht-fachgerechtes, Auftrennen des Stromkreises Schäden am Flugobjekt entstehen, was ein gewisses Verständnis über die Grundlagen der verbauten Technologien vorausgesetzt.

4.1.3 Energy-Harvesting

Energy-Harvesting⁸ (deutsch: Energie-Ernte/Sammlung) ist ein beliebtes Verfahren für Sensoren oder Schalter mit geringer Energieaufnahme. So kann beispielsweise beim Auslösen eines Schalters die physikalische Kraft, die auf den Schalter wirkt, genutzt werden, um genug Energie für die nötige Datenübertragung zu erzeugen. Weitere Quellen wie Licht oder Temperaturunterschiede, z.B. für Regelungskomponenten an einer Heizung, bieten ein weites Feld an Möglichkeiten. Bei einer Drohne kommt beispielsweise der herrschende Luftzug sowie die Rotationsbewegungen der Rotoren zur Energierückgewinnung in Frage. Des weiteren können die Vibrationen der Drohne betrachtet werden, um zu klären ob hierdurch Energie erzeugt werden kann. Der Hauptvorteil dieser Energierückgewinnung liegt in der Zuverlässigkeit, indem der Flugschreiber bei Inbetriebnahme der Drohne sicher mit Strom versorgt wird. Nachteile bestehen im ungewissen Hardware-Umfang, dem daraus resultierenden Gewicht und den Einflüssen auf die Aerodynamik.

4.1.4 Ausblick

Im Rahmen des Projekts wurden vorrangig die Möglichkeiten der Energierückgewinnung mittels verschiedener Energy-Harvesting-Ansätze im Vergleich zur konventionellen Methode unter Verwendung einer Batterie untersucht. Aufgrund der Lieferschwierigkeiten der Versuchs-Drohne konnte der Betrieb eines Flugschreibers nicht unter Zuhilfenahme des bestehenden Stromkreises der Drohne getestet werden.

4.2 Datenhaltung

In diesem Abschnitt werden verschiedene Vorgehensweisen zur Datenhaltung erfasster Flugdaten, wie beispielsweise GPS-Informationen, erläutert.

⁸siehe auch: Harvesting-Energiequellen für Elektroniksysteme ([Link](#))

4.2.3

4.2.1 Offline

Unabhängig davon, ob die erfassten Daten an einen Server übertragen werden sollen oder nicht, erfolgt stets eine lokale Datenhaltung, welche temporär oder dauerhaft erfolgen kann. Dies ist notwendig, da selbst bei einer Übertragung der Daten über eine Datenverbindung, beispielsweise über das Mobilfunknetz, keine dauerhaft bestehende Verbindung garantiert werden kann bzw. Datenübertragungsfehler auftreten können. Die lokale und somit offline stattfindende Datenspeicherung bildet daher die Grundlage der Datenhaltung. In Abschnitt 5.1 werden die Sicherheitsziele, die hierfür erreicht werden müssen, definiert und geeignete Technologien vorgestellt. Im Projekt erfolgte die Speicherung der GPS-Informationen auf einer handelsüblichen SD-Karte (vgl. Abschnitt 6.2.2.2).

4.2.2 Online

Im Gegensatz zur Offline-Speicherung der erfassten Flugdaten bietet eine Online-Übertragung der Informationen den Vorteil, dass die Daten selbst bei einer Zerstörung des Flugschreibers bis kurz vor dem Absturz erhalten bleiben. Hierbei kann zwischen zwei Modellen unterschieden werden:

- Übertragung mittels Mobilfunknetz
- Übertragung mittels lokalem Funknetz

Bei der Datenübertragung mittels Mobilfunknetz verfügt das Flugobjekt eine SIM-Karte, welches der Drohne erlaubt, eine Datenverbindung über das Mobilfunknetz aufzubauen. Mithilfe dieser können Daten nahezu in Echtzeit an einen Server übertragen und gespeichert werden. Wie im vorherigen Abschnitt 4.2.1 beschrieben, setzt dies dennoch mindestens eine temporäre Offline-Datenspeicherung voraus, um kurzzeitige Verbindungsabbrüche zu kompensieren.

Bei der Datenübertragung durch eine lokales Funknetz wird eine Datenverbindung zwischen der Steuereinheit beim Piloten und der Drohne aufgebaut. Hierfür kann beispielsweise ebenfalls das Frequenzband verwendet werden, welches zur Steuerung eingesetzt wird und meist ähnlich des WLAN-Standards bei 2,4 GHz liegt. Die Speicherung der Fluginformationen kann in Echtzeit beim Piloten erfolgen. Alternativ könnte statt der Steuereinheit zusätzliche Hardware, wie beispielsweise ein Smartphone, verwendet werden. Diese Variante würde ebenfalls eine temporäre offline Datenspeicherung auf Seiten der Drohne benötigen, um Verbindungsabbrüche zu kompensieren, bietet aber den Vorteil, dass die Datenübertragung auch in Gebieten außerhalb der Mobilfunknetzabdeckung funktionsfähig bleibt.

4.2.3

4.2.3 Ausblick

Im Rahmen des Projekts wurde vorrangig die Systemarchitektur zur Datenhaltung untersucht. Hierzu wurden die zu erreichenden Sicherheitsziele definiert und auf die lokal erfassten Daten angewendet. Verschiedene Technologien wurden untersucht und kombiniert, um diese Ziele zu erreichen. Der Fokus wurde hierbei auf die korrekte, nachverfolgbare und fälschungssichere Datenhaltung gelegt. Die reine Datenübermittlung stellt nach Erreichen dieser Sicherheitsziele aufgrund der bereits etablierten Technologien zur Datenübertragung nur noch eine untergeordnete Rolle dar. Dabei kann beispielsweise auf eine Public-Key-Infrastruktur zwischen Client und Server gesetzt werden.

5 Datenhaltung

Der folgende Abschnitt beschreibt unter Zuhilfenahme vorhandener Technologien und Sicherheitsstandards die Konzeption einer Systemarchitektur zur sicheren Speicherung von Daten.

5.1 Sicherheitsziele

Mithilfe des Flugschreibers werden über den GPS-Empfänger Standortdaten ermittelt und gespeichert. Die Speicherung der Daten erlaubt die Erfüllung verschiedener Sicherheitsziele, welche im Folgenden erläutert werden.

Nach [Sch17] und [IBM17] lassen sich in der Informationssicherheit unter anderem die folgenden Sicherheitsziele definieren:

Tabelle 3: Sicherheitsziele in der Informationssicherheit

Sicherheitsziel	Beschreibung
Integrität (integrity)	Es können keine Informationen unbemerkt verändert werden.
Verfügbarkeit (availability)	Verhinderung von Ausfällen, sodass der Zugriff auf die Informationen gewährleistet ist.
Vertraulichkeit (confidentiality)	Ausschließlich der Empfänger bzw. autorisierte Personen können die Nachricht lesen.
Authentizität (authenticity)	Die Kommunikationspartner können ihre jeweilige Identität dem Gegenüber beweisen.
Nachweisbarkeit (non repudiation)	Die Übermittlung von Informationen bzw. die Durchführung von Handlungen kann nicht abgestritten werden.
Privatsphäre (privacy)	Der Zugriff auf Daten, die in Beziehung zur Identität stehen, werden eingeschränkt.

Die Notwendigkeit der Erfüllung der Sicherheitsziele bei der Datenhaltung durch

5.2.1

den Flugschreiber wurde wie folgt bewertet:

Tabelle 5: Notwendigkeit der Erfüllung der Sicherheitsziele bei der Datenhaltung durch den Flugschreiber

Sicherheitsziel	Benötigt	Optional
Integrität (integrity)	X	
Verfügbarkeit (availability)	X	
Vertraulichkeit (confidentiality)		X
Authentizität (authenticity)	X	
Nachweisbarkeit (non repudiation)	X	
Privatsphäre (privacy)		X

Die Integrität des Flugschreibers und dessen aufgezeichneten Daten ist ein elementarer Bestandteil des Projektziels, da die Informationen unter keinen Umständen unbemerkt verändert werden dürfen. Wird dieses Sicherheitsziel nicht erreicht, verfehlt der Flugschreiber seine Bestimmung. Ebenso muss die Verfügbarkeit des Flugschreibers maximiert werden, um zu verhindern, dass Daten nicht aufgezeichnet oder auf aufgezeichnete Daten nicht mehr zugegriffen werden kann.

Weitere zu erreichende Sicherheitsziele sind die Authentizität und die Nachweisbarkeit. Diese werden benötigt, um sicherzustellen, dass die übermittelten Standortdaten vom autorisierten GPS-Empfänger des Flugschreibers stammen und diese auf das hierfür vorhandene Speichermedium nachweisbar geschrieben wurden.

Hingegen werden die Sicherheitsziele Vertraulichkeit und Privatsphäre als optional eingestuft, da der Verschluss der Daten weder die Funktionalität des Flugschreibers noch die Projektziele beeinflussen.

5.2 Technologien

Die folgenden Technologien tragen zur Erfüllung der im vorigen Abschnitt deklarierten Sicherheitsziele bei und werden in den nachfolgenden Abschnitten kurz erläutert. Sie werden in Teilen in der Konzeption der Systemarchitektur des Flugschreibers verwendet.

5.2.5

5.2.1 Einweg-Funktion

Eine Einweg-Funktion beschreibt eine mathematische Funktion, welche den Funktionswert einer gegebenen Zeichenfolge ohne komplexen Aufwand berechnet, während sich die Umkehrung als zu komplex zur Durchführung in angemessener Zeit gestaltet. In Kombination mit Komprimierungsfunktionen (Hash) bieten Einweg-Funktionen unter anderem die Hauptgrundlage für digitale Signaturen. [Ric02]

5.2.2 Symmetrische Verschlüsselung

Bei der symmetrischen Verschlüsselung erfolgt die Ver- und Entschlüsselung einer gegebenen Zeichenfolge mit dem gleichen Schlüssel. Die Schwäche hierbei liegt in der Notwendigkeit, den gemeinsamen Schlüssel zwischen Sender und Empfänger auszutauschen. Wird dieser Austausch kompromittiert ist die gesamte nachfolgende Verschlüsselung zwecklos.

5.2.3 Asymmetrische Verschlüsselung

Die asymmetrische Verschlüsselung behebt die Schwäche der symmetrischen Verschlüsselung durch Verwendung von Schlüsselpaaren, die aus je einem privaten und einem öffentlichen Schlüssel bestehen. Wird einer der beiden Schlüssel zur Verschlüsselung verwendet, bietet der andere Schlüssel die Möglichkeit zur Entschlüsselung. Es wird daher vorab kein privater Schlüsselaustausch, wie bei einer symmetrischen Verschlüsselung, benötigt.

Der private Schlüssel verbleibt beim entsprechenden Teilnehmer, während der öffentliche Schlüssel der Allgemeinheit zugänglich gemacht wird.

Jeder zukünftige Kommunikationspartner kann mit Kenntnis des öffentlichen Schlüssels des Empfängers die Nachrichten mit dessen öffentlichen Schlüssel verschlüsseln, während ausschließlich der Empfänger die Nachricht unter Verwendung seines privaten Schlüssels entschlüsseln kann.

5.2.4 Digitale Signatur

Digitale Signaturen werden von Objekten über eine Einweg-Funktion erstellt und anschließend mit dem privaten Schlüssel des Signierenden verschlüsselt. Das Objekt wird gemeinsam mit der Signatur versendet. Der Empfänger kann die Signatur

5.2.6

mithilfe des öffentlichen Schlüssels entschlüsseln und den erhaltenen Hash-Wert mit dem Hash-Wert des Objekts vergleichen und so dessen Integrität prüfen.

5.2.5 Digitales Zertifikat

Ein digitales Zertifikat bestimmt mittels kryptografischer Verfahren die Echtheit der Angaben über die Eigenschaften von Personen oder Objekte. Hierbei wird der öffentliche Teil eines asymmetrischen Schlüsselpaares von einer Zertifizierungsstelle mit einer digitalen Signatur versehen, um die Überprüfung der gemachten Angaben durch Dritte zu ermöglichen. Voraussetzung ist das Vertrauen in die Zertifizierungsstellen und in die hieraus entstehende Public-Key-Infrastruktur.

5.2.6 Trusted-Computing-Plattform

Eine *Trusted-Computing-Plattform* besteht aus einem Computer-System (z.B.: PC), welches mit einem *Trusted-Platform-Module* (TPM)-Chip ausgestattet ist und wurde von der *Trusted Computing Group* (TCG) spezifiziert. Der TPM-Chip ermöglicht die Zustandsspeicherung der Hardware, Software und des Betriebssystems. Dies gestattet eine Integritätsprüfung der genannten Komponenten. Änderungen werden erkannt und Reaktionen können daraufhin ausgelöst werden.

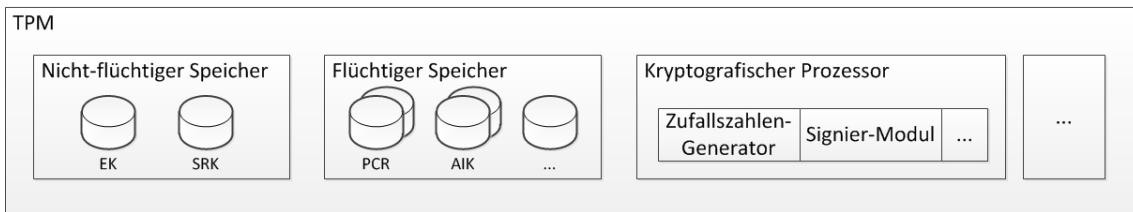


Abbildung 4: Funktionalitäten des TPMs

Abbildung 4 zeigt exemplarisch die für das Projekt relevanten Funktionalitäten des TPMs. Diese lassen sich grob in die drei Bereiche *Nicht-flüchtiger Speicher*, *Flüchtiger Speicher* und *Kryptografischer Prozessor* einteilen.

Im nicht-flüchtigen Speicher ist der *Endorsement Key (EK)* hinterlegt, welcher beispielsweise in der aktuellen TPM-Version 2.0 aus einem asymmetrischen 2048Bit-RSA-Schlüsselpaar besteht. Der EK ist einzigartig und wird erstmals bei der Herstellung des TPM-Chips durch den Hersteller erzeugt. Dieser kann durch den späteren Besitzer jedoch gelöscht und neu erstellt werden. Trotz Neuerstellung lässt sich ausschließlich immer nur der öffentliche Schlüssel des asymmetrischen Schlüsselpaares auslesen und niemals der private Schlüssel.

5.2.6

Neben dem EK befindet sich im nicht-flüchtigen Speicher außerdem der *Storage Root Key (SRK)*. Mithilfe des EK und eines benutzerspezifischen Passworts wird der SRK generiert. Dieser wird verwendet, um weitere vom Benutzer erzeugte Schlüssel zu verschlüsseln, bevor diese in freien Registern des flüchtigen Speichers abgelegt werden. Der SRK bildet die oberste Hierarchiestufe innerhalb der Vertrauenskette des TPM-Chips, ähnlich einer übergeordneten Zertifizierungsstelle innerhalb einer Public-Key-Infrastruktur.

Im flüchtigen Speicher befinden sich neben beliebigen vom Benutzer erzeugten Schlüsseln das *Platform Configuration Register* (PCR). In diesem werden gemessene Konfigurationen als Hash gespeichert, welche wiederum von *Attestation Identity Keys* (AIK) signiert werden. Durch die Verwendung des AIK in Kombination mit den gespeicherten Hash-Werten können Soft- und Hardware-Zustände abgebildet und identifiziert werden.

Dabei werden die Daten im flüchtigen Speicher, wie beispielsweise die AIKs, nicht dauerhaft gespeichert. Stattdessen können die Schlüssel verschlüsselt aus dem Speicher kopiert, bei Bedarf wieder in den TPM geladen, entschlüsselt und verwendet werden. Dies ermöglicht Funktionalitäten wie beispielsweise einen abgesicherten Boot-Vorgang, bei dem das zuvor identifizierte Betriebssystem Zugriff auf einen zuvor erstellten symmetrischen Schlüssel erhält, um den Bootvorgang auf einem verschlüsselten Datenträger fortzuführen, welcher mithilfe des symmetrischen Schlüssels entschlüsselt werden kann.

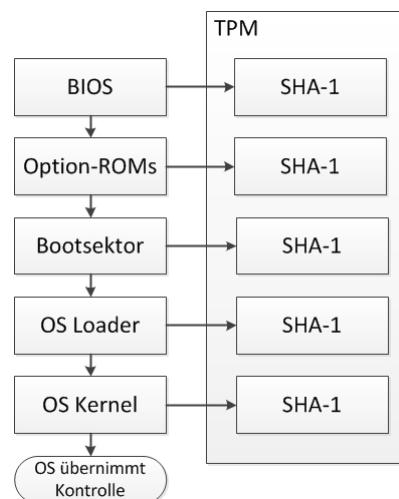


Abbildung 5: Funktionalitäten des TPMs

Quelle: angelehnt an http://ra.ziti.uni-heidelberg.de/pages/student_work/seminar/hws06/Michael_Krauss/praezentation.pdf

Können Komponenten, wie beispielsweise das Betriebssystem oder das BIOS nicht fehlerfrei identifiziert werden, wird die Herausgabe des Schlüssels verweigert, wodurch das Betriebssystem nicht gestartet werden kann (vgl. Abbildung 5).

Das Vorgehen verschlüsselte Daten an eine Plattformkonfiguration zu binden, welche von einem TPM-Chip erfasst wurde, wird als *Data Sealing* bezeichnet. Hierbei ist die Wurzel des Vertrauens der SRK, dessen privater Schlüssel im nicht-flüchtigen Speicher sicher hinterlegt ist. Ausschließlich mithilfe des SRK lässt sich das RSA-Objekt entschlüsseln. Dieses kann wiederum den symmetrischen AES-Schlüssel entschlüsseln, mithilfe dessen die Nutzdaten schließlich wieder in Klartext überführt werden können. Abbildung 6 veranschaulicht die Abhängigkeiten grafisch.

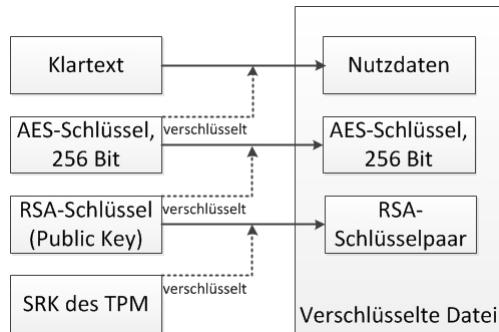


Abbildung 6: Schrittweises Vorgehen beim Data Sealing

Quelle: angelehnt an http://ra.ziti.uni-heidelberg.de/pages/student_work/seminar/hws06/Michael_Krauss/praesentation.pdf

Neben den Speichern existiert der kryptografische Prozessor, welche für die beschriebenen Aufgaben entsprechende hardwaregestützte Funktionen zur Verfügung stellt, wie beispielsweise einen Zufallszahlen- oder Schlüsselgenerator.

5.3 Systemarchitektur des Flugschreibers

Um die in Abschnitt 5.1 definierten und als notwendig bewerteten Sicherheitsziele zu erreichen, wurde im Folgenden eine Systemarchitektur konzipiert, welche mithilfe der Funktionalitäten eines TPM-Chips die Unversehrtheit des Flugschreibers und der aufgezeichneten Daten sicherstellen soll.

Abbildung 7 zeigt die für die Datenerfassung und Datenhaltung relevanten Module des Flugschreibers. Hierbei werden Standortdaten über den GPS-Empfänger erfasst, über ein Geschäftslogik-Modul ausgewertet und an das Speicherkarten-Modul weitergegeben. Dieses legt die Daten auf einer Speicherkarte ab.

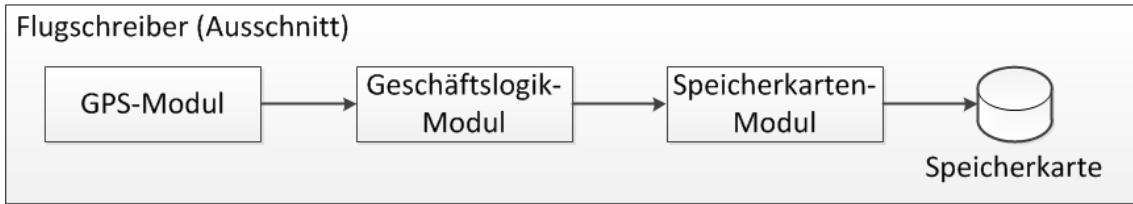


Abbildung 7: Für die für die Datenerfassung und Datenhaltung relevante Flugschreibermodule

Um eine Manipulation des GPS- und Geschäftslogik-Moduls zu verhindern, können diese entsprechend des in Abschnitt 5.2 erläuterten Verfahren zur Identifizierung des Soft- und Hardwarezustandes auf Unversehrtheit überprüft werden. Die erfassten Standortdaten können innerhalb der Geschäftslogik mithilfe des Signier-Moduls des TPM-Chips unterschrieben werden, bevor diese an das Speicherkarten-Modul weitergegeben werden. Da die Daten vor Verlassen des Geschäftslogik-Moduls bereits signiert wurden, muss die Integrität nachfolgender Komponenten, wie beispielsweise des Speicherkarten-Moduls und der Speicherkarte, nicht mehr geprüft werden. Erfolgt hier eine Manipulation der Daten, kann diese durch die Überprüfung der Signatur festgestellt werden.

Da abhängig von der Energiequelle des Flugschreibers die eingespeiste Leistung sehr gering ist, kann eine Architektur, welche nicht auf einem zusätzlichen TPM-Chip beruht, sinnvoll sein. Ähnlich der Hochzeit im Automobilbau, bei der die Karosserie und der Motor zusammengeführt werden, zeigt Abbildung 8 exemplarisch die Möglichkeit einzelne Komponenten direkt miteinander „zu verheiraten“. Hierbei wird den Komponenten, die direkt miteinander kommunizieren, das jeweilige Zertifikat des Kommunikationspartners fest in die Hardware eingetragen. Es entsteht, ähnlich des Prinzips des Web of Trust, ein Netz aus Vertrauen.

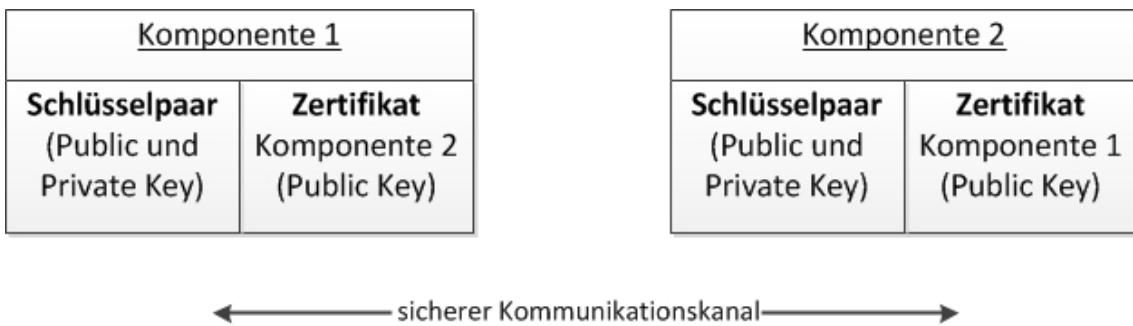


Abbildung 8: Prinzip einer softwareseitigen Heirat von Komponenten

In Abbildung 8 beinhaltet die Komponente 1 das Zertifikat (öffentlicher Schlüssel)

5.3

von Komponente 2 und umgekehrt beinhaltet Komponente 2 das Zertifikat von Komponente 1. Zusätzlich beinhaltet jede Komponente das eigene Schlüsselpaar (öffentlicher und privater Schlüssel) entsprechend des, an andere Komponenten verteilten, Zertifikates.

Bei der Anwendung auf die Systemarchitektur des Flugschreibers würde das GPS-Modul mit dem Geschäftslogik-Modul verheiratet werden. Das Geschäftslogik-Modul könnte außerdem den eigenen privaten Schlüssel verwenden, um die empfangenen Daten zu signieren, bevor diese auf ein externes Speichermedium übertragen werden (vgl. Abbildung 9).

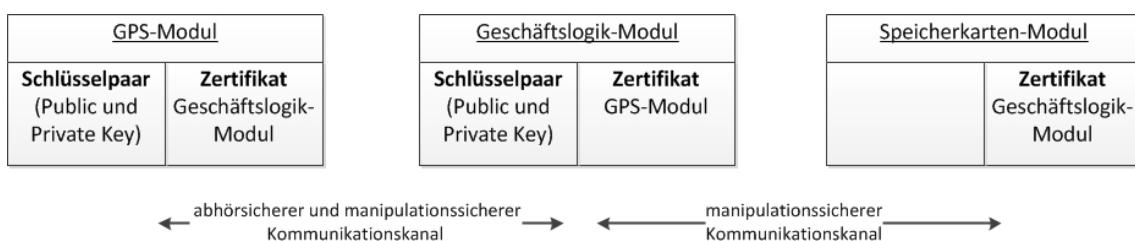


Abbildung 9: Verheiratete Komponenten am Beispiel der Systemarchitektur zur Datenhaltung im Flugschreiber

Sowohl bei der Nutzung eines TPM-Chips, als auch bei der Nutzung verheirateter Komponenten, muss die Hardware logische und physikalische Gegenmaßnahmen implementieren, um diverse Angriffe, wie beispielsweise das Eindringen in das Gehäuse, zu erschweren.⁹

⁹siehe: Hacker liest Kryptoschlüssel aus TPM-Chip aus (Link)

6 Feldtests

Während des Projekts wurden einige Feldtests durchgeführt, welche in diesem Abschnitt näher erläutert werden. Dies beinhaltet Versuche zur Energierückgewinnung sowie Experimente mit verschiedenen GPS-Empfängern am Arduino.

6.1 Energy Harvesting

Zur autarken Energiegewinnung für den Betrieb des Flugschreibers wurden anhand verschiedener Testaufbauten mehrere Energy Harvesting-Ansätze getestet. Im Folgenden wird der Testaufbau sowie die durchgeführten Versuche näher erläutert.

6.1.1 Testaufbau

Da während des Projekts keine funktionsfähige Drohne zur Verfügung stand, wurde zur Durchführung der Versuche kurzfristig eigene Hardware beschafft. Um das Prinzip des Energy Harvestings mittels verschiedener Ansätze testen zu können, wurde der in Abbildung 10 dargestellte exemplarische Minimalaufbau zur Ansteuerung eines Motors durchgeführt, an welchem Messungen vorgenommen werden konnten.

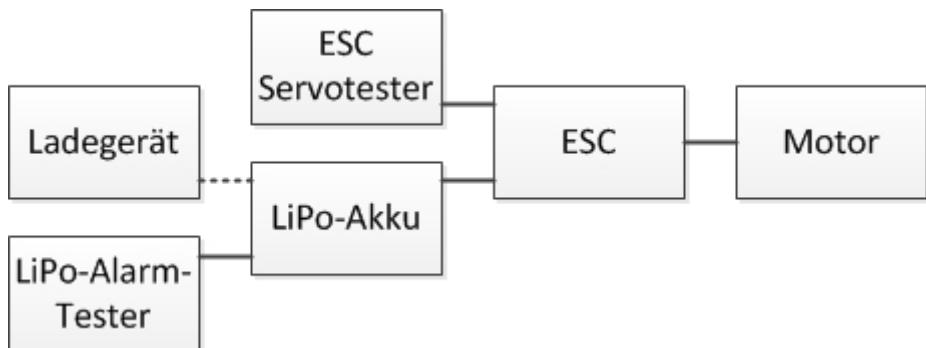


Abbildung 10: Exemplarischer Minimalaufbau zur Ansteuerung eines Motors

Der Motor stellt die Kernkomponente des Aufbaus dar. Hierfür wurde ein Brushless-Motor (A2212) mit einer Leistungsaufnahme von bis zu 180 Watt verwendet. Dieser Typ findet bereits Anwendung auf verschiedenen Mittelklasse-Drohnen und bietet somit eine gute Versuchs-Grundlage. Entsprechend des Prinzips einer Synchronmaschine befindet sich innerhalb von Brushless-Motoren typischerweise eine Drehstromwicklung. Die hier dreisträngige Drehstromwicklung wird durch den Drehzahlregler

6.1.2

(engl: Electronic Speed Control (ESC)) mittels eines um 120° versetzten Rechtecksignals so angesteuert, dass ein drehendes magnetisches Feld erzeugt wird, welches den permanenterregten Rotor in Bewegung setzt.

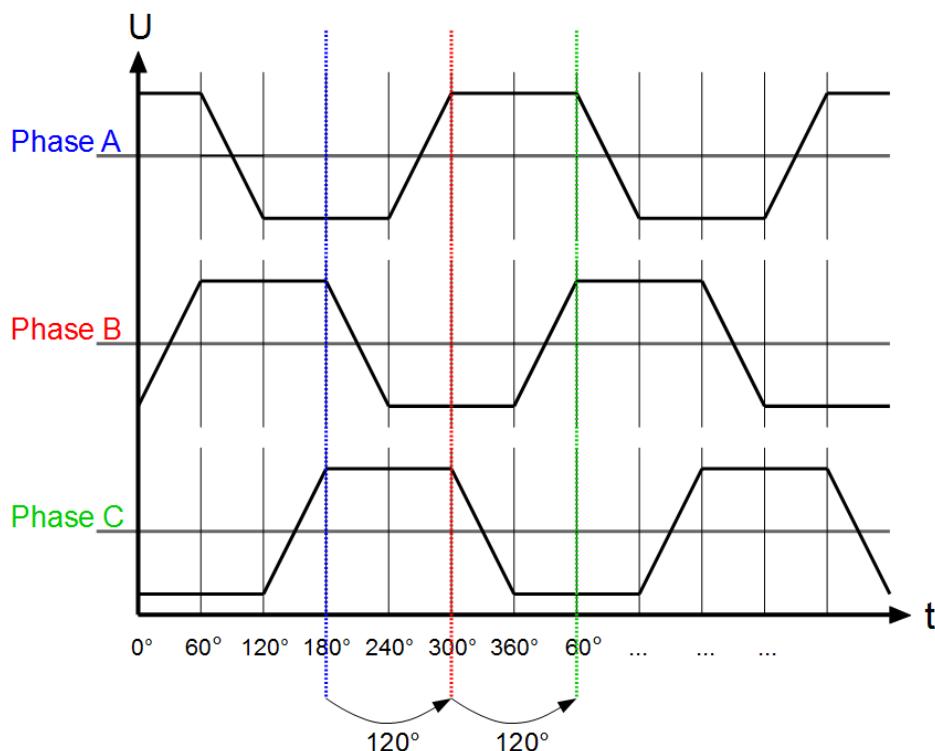


Abbildung 11: Ansteuerung und Verschiebung der verschiedenen Phasen des Elektromotors in Abhängigkeit der Zeit
Quelle: <http://www.breadboarding.de/brushless-motor/>

Abbildung 11 zeigt die Ansteuerung und Verschiebung der verschiedenen Phasen des Elektromotors in Abhängigkeit der Zeit. Die Generierung dieser Signale erfolgt über den sogenannten ESC-Servotester.

Als Energiequelle für den Drehzahlregler, welcher wiederum dessen Servotester mit Strom versorgt, kann wahlweise ein Netzgerät oder ein Akku verwendet werden. Aus Kostengründen wurde für den Testaufbau ein Lithium-Polymer-Akku verwendet. Aufgrund der Verwendung eines Akkus musste außerdem ein Ladegerät und aus Sicherheitsgründen ein sogenannter Alarm-Tester für Lithium-Polymer-Akkus beschafft werden. Letzterer warnt akustisch bei der Verwendung des Akkus, sobald eine Tiefentladung der Zellen stattfindet. Eine Tiefentladung der Zellen ist schädlich für den Akku und kann neben der Zerstörung einzelner Zellen bei Weiterverwendung

6.1.2

zu starken Erhitzungen des Akkus und sogar zum Brand führen.

6.1.2 Induktive Kopplung

Um den Flugschreiber mit Energie zu versorgen wurde unter anderem versucht, die notwendige Energie anhand einer elektromagnetischen induktiven Kopplung zu gewinnen. Durch die Änderungen des Magnetfeldes im Brushless-Motor wurde angenommen, dass eine elektrische Spannung und ein Stromfluss in einem darum liegenden Leiter erzeugt werden kann. Hierzu wurde mittels Kupferdraht ein benachbarter Stromkreis geschaffen, welcher mit mehreren Wicklungen um den Motor gelegt wurde (vgl. Abbildung 12).



Abbildung 12: Am Motor anliegender Kupferdraht

Der Kupferdraht aus Abbildung 12 wurde an eine Schaltung mit einem Brückengleichrichter angelegt (vgl. Abbildung 13), welcher mithilfe eines Elektrolytkondensators aus der eingespeisten pulsierenden Gleichspannung (ähnlich eines Rechtecksignals) eine kontinuierliche Gleichspannung erzeugt.

Der Spannungsregler (IC 7805) stabilisiert hierbei die Eingangsspannung und wirkt als Strombegrenzer auf den dahinterliegenden Stromkreis, an welchen der Verbraucher angeschlossen ist. Dieser besteht aus einer LED mit einem Vorwiderstand von 200Ω und dient als einfaches Ausgabe-Instrument, um zu prüfen ob Strom in den Stromkreis injiziert wird. Die Eingangsspannung hängt hierbei von der Anzahl der Windungen und der Drehzahl des Motors ab.

Während der Durchführung des Tests wurde der injizierte Strom und die Spannung mithilfe eines Multimeters hinter dem Brückengleichrichter gemessen. Die Werte waren jedoch sehr gering, sodass beispielsweise maximal eine Spannung von 0,5 mV bei entladenem Kondensator gemessen werden konnte, welche aufgrund des noch

6.1.3

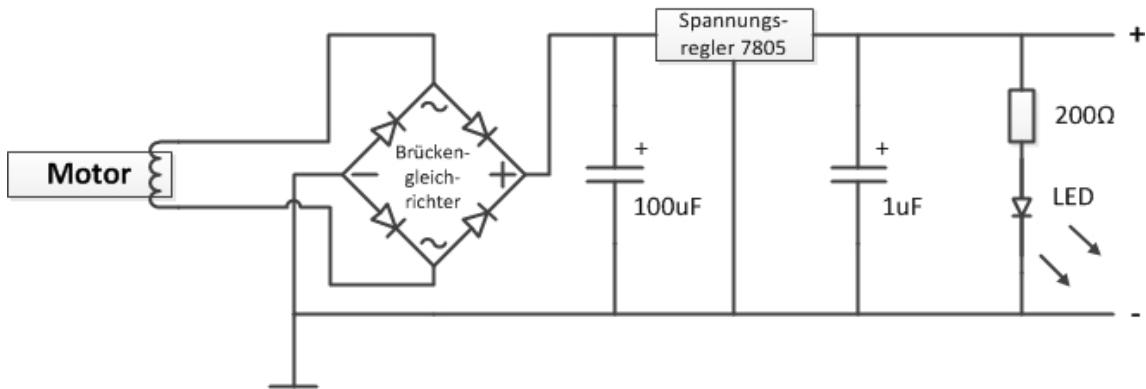


Abbildung 13: Brückengleichrichter mit elektromagnetischer induktiver Kopplung am Motor

geringeren Stroms nicht annähernd ausreichend war, um die angeschlossene LED betreiben zu können. Dies lässt auf ein sehr schwaches Magnetfeld um den Motor schließen.

Bei einem weiteren Test (vgl. Abbildung 14) wurde der Kupferdraht um eine Phase zwischen Drehzahlregler und dem Motor gewickelt. Die hierbei injizierte Leistung war höher als im vorigen Test, jedoch immer noch deutlich unterhalb der benötigten Leistung zum Betrieb der angeschlossenen LED.

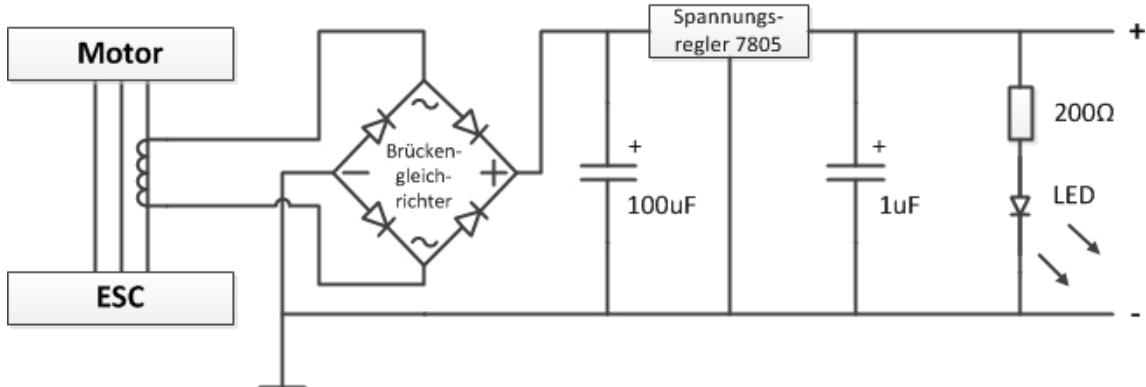


Abbildung 14: Brückengleichrichter mit elektromagnetischer induktiver Kopplung an der Phase

Um die eingespeiste Leistung zu erhöhen und die magnetische Leitfähigkeit zu verbessern, wurde zusätzlich ein Eisenkern eingesetzt. Dieser wurde mit dem Kupferdraht umwickelt und anschließend mehrere Windungen einer Phase (rot) darum gelegt (vgl. Abbildung 15).

6.1.3.2

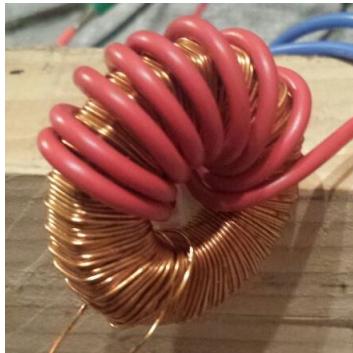


Abbildung 15: Eingesetzter Eisenkern zur Erhöhung der magnetischen Leitfähigkeit

6.1.3 Mechanische Kopplung

Im Gegensatz zur induktiven Kopplung wurden im Folgenden mehrere Ansätze der direkten mechanischen Kopplung zwischen Motor bzw. Rotor und dem Stromkreislauf des Versuchsaufbaus aus dem vorigen Abschnitt durchgeführt.

6.1.3.1 Gleichstrommaschine Durch die Anbringung einer Gleichstrommaschine am Motor kann diese, angetrieben durch die Rotation des Motors, ebenfalls Strom erzeugen. Dieses Prinzip wird beispielsweise bei Fahrrad-Dynamos eingesetzt, um Energie für die Beleuchtung zu gewinnen. In einem Versuch wurde ein solcher handelsüblicher Fahrrad-Dynamo verwendet und mithilfe eines Keilriemen an den Motor angeschlossen (vgl. Abbildung 16). Als Keilriemen wurde ein handelsübliches Gummiband verwendet. Während des Versuchs stellte sich heraus, dass die Positionierung des Dynamos entscheidend ist, um ein Abrutschen des Keilriemens zu verhindern.

Die elektronische Schaltung zur Gleichrichtung konnte, wie in Abschnitt 6.1.2 beschrieben, wiederverwendet werden. Mit der Ausnahme, dass der Kupferdraht keine Spule mehr bildet, sondern die beiden Pole direkt an den Dynamo angeschlossen werden und von dort zur Schaltung des Brückengleichrichters führen. Während des Versuchs konnte eine maximale Spannung von 5,13V bei einem Widerstand von 100Ω erreicht werden. Bei Demontage des Rotors konnten sogar dauerhaft 6,45V gemessen werden. Von einem Realbetrieb mit Rotor ausgehend kann bei einer Spannung von 5,13V mithilfe der Formel $P = \frac{U^2}{R}$ eine Leistung von ca. 263mW bei einer Stromstärke von ca. 51mA berechnet werden. Die Test-LED konnte während des Versuchs zum Leuchten gebracht werden.

6.1.3.2

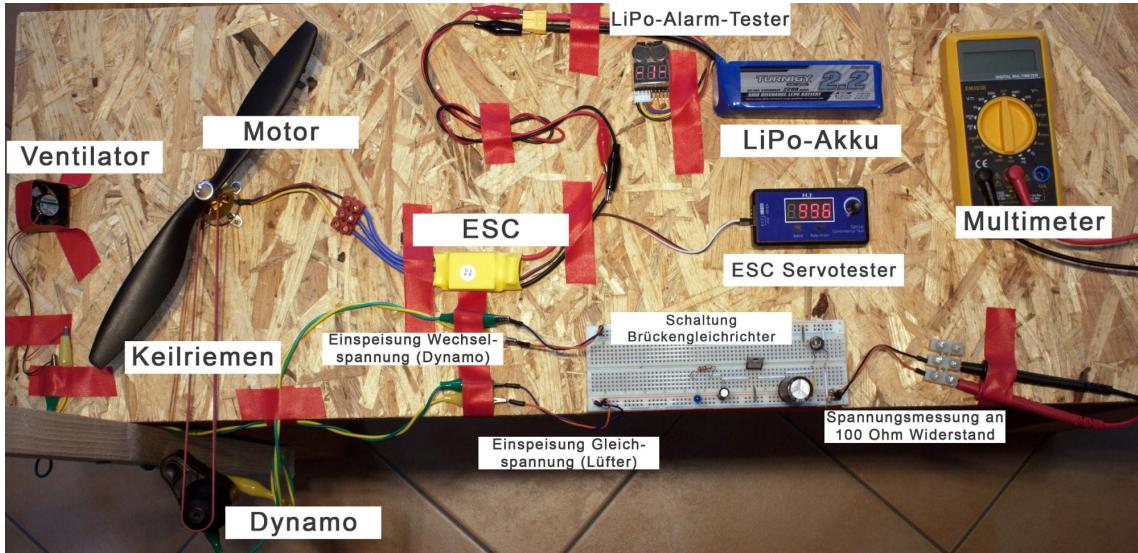


Abbildung 16: Vollständiger Versuchsaufbau

6.1.3.2 Axialventilator Im Gegensatz zur mechanischen Kopplung mit direktem Kontakt zum Motor kann eine Gleichstrommaschine unter Nutzung der erzeugten Luftströme ebenfalls angetrieben werden. Dieses Prinzip wird in der Luftfahrt bereits mittels sogenannter *Ram-Air-Turbinen (RAT)* angewendet, um in einer Notfallsituation aus dem Fahrtwind Energie gewinnen zu können.

Im Rahmen des Projekts wurde ebenfalls ein Versuchsaufbau errichtet, welcher der *Ram-Air-Turbine* sehr ähnlich ist. Hierzu wurde ein *Axialventilator* unter den Rotor des Test-Motors montiert (vgl. Abbildung 16), um durch die erzeugten Luftströmungen den Ventilator anzutreiben und so die Energie mittels des Prinzips einer Gleichstrommaschine zurückzuführen. Verwendet wurde hierfür ein Lüfter des Herstellers *SUNON* mit einem Querschnitt von $30\text{mm} \times 30\text{mm}$ und einer maximalen Leistungsaufnahme von $0,6\text{W}$ bei einer Eingangsspannung von 5V .

Da der Axialventilator eine Gleichspannung liefert, wurden dessen Phasen direkt an die Gleichspannungs-Ausgänge des Brückengleichrichters angeschlossen, um diesen zu übergehen. Während des Versuchsaufbaus stellte sich heraus, dass eine optimale Platzierung des Ventilators relativ aufwendig erprobt werden muss. Dies lag unter anderem am Versuchsaufbau, da dieser für diesen Versuch keinen realitätsnahen Nachbau darstellte, da der Motor direkt auf einer Unterlage montiert wurde, wodurch der Luftstrom nicht ungehindert fließen konnte und es zu starken Verwirbelungen kam.

Während des Versuchs konnte der Ventilator durch den verursachten Luftstrom in Bewegung gesetzt und gehalten werden. Jedoch konnten, trotz mehrfacher Versuche

6.1.4

die Position des Ventilators weiter zu optimieren, nur sehr geringe Spannungsspitzen gemessen werden. Diese betragen maximal $0,12V$ bei einem Widerstand von 100Ω .

6.1.3.3 Vibration Eine weitere Möglichkeit zur Energiegewinnung stellen piezoelektrische Kristalle dar. Diese erzeugen bei Krafteinwirkung, wie beispielsweise durch Vibrationen, Druck oder Erschütterungen, elektrische Spannungen. Diese Technologie wird unter anderem bei Funkschaltern oder Reifendrucksystemen verwendet, ohne dabei auf eine externe Stromversorgung zurückgreifen zu müssen.

Die Firma *MicroGen Systems* hat hierfür bereits 2013 mehrere Produkte auf den Markt gebracht.¹⁰ Das Produkt *BOLT™ Power Cell (M0120B)* wurde hierbei bereits vom Hersteller auf Drohnen getestet und reicht aus, um beispielsweise eine Test-LED aufgrund der Eigenvibrationen der Drohne mit Energie zu versorgen.¹¹ Spitzenmodelle erreichen hierbei bei optimalen Bedingungen mit einer Vibrationsfrequenz von ca. $600Hz$ eine Leistung von bis zu $900uW$.

Da die Leistung selbst im optimalen Fall für den vollständigen Betrieb eines Flugschreibers nicht ausreicht, wurden bei der Betrachtung dieser Technologie keine weiteren Versuche durchgeführt.

6.1.4 Batterie

Als Alternative wurden zwei 9V-Lithium-Batterien getestet. Verwendet wurden hierfür handelsübliche 9V-Blöcke der Serie L522 des Herstellers Energizer. Diese wurden über einen Langzeittest unterschiedlich stark belastet, um Erfahrungswerte über die Leistungsfähigkeit gewinnen zu können.

Im Versuch wurde die ausgehende Spannung beider Blöcke mittels Spannungsregler (*78L05*) auf $5V$ gebracht, was in etwa der Betriebsspannung der verwendeten GPS-Empfänger entspricht. An diesen wurden verschiedene Lüfter als Verbraucher angeschlossen.

Der erste Block wurde mit $50mA$ belastet, was einer Leistungsaufnahme am Verbraucher von $250mW$ entspricht, während der zweite Block mit $180mA$ belastet und der Verbraucher mit $900mW$ gespeist wurde.

Bis zum Stillstand des ersten Lüfter vergingen ca. 14,6 Stunden. Der zweite Lüfter stoppte bereits nach 4,1 Stunden.

¹⁰<https://www.cornestech.co.jp/images/uploads/file/products/pdf/bolt.pdf>

¹¹<https://www.youtube.com/watch?v=bw14GBBcyd4>

6.1.5

Während des Versuchs wurde nach einer Stunde an beiden Batterien Spannungen zwischen 7,7 und 8,3V gemessen. Die Differenz zwischen der Betriebsspannung des Verbrauchers (5V) und der Spannung der Batterie (im Mittel 8V) beträgt 3V. Diese fiel während des Versuchs am Spannungsregler in Form von Wärme ab. Durch den späteren Einsatz eines Abwärtswandler anstatt des eingesetzten Spannungsreglers kann diese Verlustleistung mit einem Wirkungsgrad von bis zu 90% ¹² minimiert werden. Diese lässt sich dann wie folgt berechnen:

$$\begin{array}{lll} \text{Erster Block: } & P = U \cdot I & P = 3V \cdot 50mA = 150mW \\ \hline \text{Zweiter Block: } & P = U \cdot I & P = 3V \cdot 180mA = 540mW \end{array}$$

Aus der verlorenen Leistung kann eine Verlängerung der Betriebszeit wie folgt geschätzt werden:

$$\begin{array}{lll} W = 250mWh \equiv 14,6h & & \\ \text{Erster Block: } & 150mW \cdot 14,6h = 2190mWh & \\ & t = \frac{2190mWh}{250mW} = 8,76h & \\ & 8,76h \cdot 0,9 = 7,9h & \\ & 14,6h + 7,9h = & \mathbf{22,5 \ h} \\ \hline W = 900mWh \equiv 4,1h & & \\ \text{Zweiter Block: } & 540mW \cdot 4,1h = 2214mWh & \\ & t = \frac{2214mWh}{900mW} = 2,46h & \\ & 2,46h \cdot 0,9 = 2,2h & \\ & 4,1h + 2,2h = & \mathbf{6,3 \ h} \end{array}$$

Theoretisch erreicht der Versuchsaufbau der ersten Batterie im Optimalfall eine Betriebszeit von bis zu 22,5 Stunden, während der zweite Versuchsaufbau bis zu 6,3 Stunden erreicht.

¹²Wirkungsgrad und Leistungsdichte erhöhen ([Link](#))

6.1.5

6.1.5 Ergebnis

Sowohl durch die induktive als auch durch die mechanische Kopplung und die daraus entstehende Rückgewinnung von Energie werden einzelne Rotoren der Drohne ausgebremst. Der Drehzahlregler der Drohne reguliert diese Entnahme jedoch, indem der jeweilige Motor mehr Leistung erhält, sodass unabhängig von der Entnahme eine konstante Drehzahl erhalten bleibt.

Die Energierückgewinnung reduziert jedoch die Akku-Laufzeit der Drohne und beeinflusst abhängig von der Größe und der Form die Aerodynamik negativ. Ebenfalls führt das Gewicht der zusätzlich angebrachten Hardware zu einer Reduzierung der Flugzeit, die je nach Leistung der Drohne sehr unterschiedlich ausfallen kann.

Die Ergebnisse der durchgeführten Versuche sind als Richtwerte zu verstehen. Der Aufbau und die verwendete Hardware können optimiert werden, indem beispielsweise ein rutschfester Keilriemen für die externe Gleichstrommaschine oder ein Axialventilator mit für den Versuchszweck angepasster Beschaufelung verwendet wird.

Tabelle 9 fasst die Erkenntnisse der durchgeführten Versuche nochmals zusammen. Hierbei schneidet die induktive Kopplung mit einer Energierückgewinnung von $1,2W$ (eine Phase), als auch bei der Energierückgewinnung in Abhängigkeit des Gewichts mit $\frac{12,2mW}{g}$ am besten unter den autarken Lösungen ab. Da die mechanische Kopplung im Versuch mittels eines Axialventilators nach dem Prinzip einer Ram-Air-Turbine nur eine extrem niedrige Leistung in den Stromkreis injizieren konnte, kann für diese Technologie keine Empfehlung ausgesprochen werden. Ebenso bleibt die mechanische Kopplung mittels Vibration bei einer Leistung von $900\mu W$ weit unter der erforderlichen Leistung zum Betrieb des Flugschreibers.

Zur Versorgung des Flugschreibers wird lt. Tabelle 9 eine Leistung zwischen 60 und $358mW$ für den GPS-Empfänger benötigt. Der verwendete Arduino Uno benötigte bei Messungen zwischen 300 und $400mW$. Aus diesen Daten ergibt sich im ungünstigsten Fall eine benötigte Leistung von bis zu $758mW(358mW + 400mW)$. Dies wird sowohl durch die induktive Kopplung mittels einer Phase als auch durch die Verwendung von mindestens drei Dynamos an verschiedenen Motoren erreicht. Hierbei ist die induktive Kopplung vorzuziehen, da diese eine deutliche höhere Leistung pro Gewicht erreicht, in Schätzungen weniger Einfluss auf die Aerodynamik aufweist und mit $98g$ deutlich leichter ist als die Montage mehrerer Gleichstrommaschinen.

Alternativ kann beispielsweise auch eine 9V-Batterie bei einer Betriebsspannung von $5V$ verwendet werden. Im vorherigen Abschnitt wurde ein Verbraucher mit $900mW$ gespeist. Diese Leistung wäre ausreichend, um den Flugschreiber über einen Zeitraum von etwa sechs bis sieben Stunden bei maximaler Auslastung mit Strom

6.1.5

zu versorgen.

Tabelle 9: Überblick über die Messergebnisse

	Induktive Kopplung	Mechanische Kopplung			Batterie
	Dynamo	Ventilator	Vibration		
Energierückgewinnung					
bei Anbindung ...					
... eines Motors	1,2W	0,26W	144uW	900uW	0W
... vierer Motoren	4,8W	1,04W	576uW		
Gewicht					
(ohne Brückengleichrichter)					
bei Anbindung ...					
... eines Motors	98g	127g	6g	21g	34g
... vierer Motoren	392g	508g	24g		
Leistung pro Gewicht	$\frac{12,2mW}{g}$	$\frac{2,0mW}{g}$	$\frac{24,0uW}{g}$	$\frac{42,9uW}{g}$	$\frac{26,5mW}{g}$ 13
Einfluss auf Aerodynamik	mittel	hoch	mittel	mittel	gering

¹² ausgehend vom Versuchsaufbau mit 900mW

6.2 GPS

Neben der Energieversorgung sind für einen Flugschreiber selbstverständlich auch Flugdaten nötig, die es aufzuzeichnen gilt. Da interne Flugdaten des Flugobjektes wie in Abschnitt 2.2 beschrieben nur schwer abgegriffen werden können, liegt der Fokus hier auf den externen Flugdaten, genauer gesagt auf den Navigationsdaten. Auch hier wurden praktische Versuche durchgeführt, welche die Verwendung von GPS-Modulen zusammen mit einem Mikrocontroller zeigen. Um Datenmanipulationen erkennen zu können, werden die Navigationsdaten vor der Speicherung auf einer SD-Karte signiert.

Da jedoch aufgrund der bestehenden Hardwarelimitation (siehe Abschnitt 6.2.2.2) keine allumfassende Implementierung möglich war, liegt der Fokus auch besonders auf den theoretischen Grundlagen des Software-Prototyps.

6.2.1 Theoretische Grundlagen

Seinen Ursprung findet das *Global Positioning System*, kurz *GPS*, im Jahr 1978, als vom amerikanischen Verteidigungsministerium die ersten Satelliten für das damals sogenannte und auch heute noch offiziell so bezeichnete NAVSTAR GPS (**N**avigation **S**ystem for **T**iming and **R**anging), in Betrieb genommen wurden [17a].

Während das System anfangs lediglich zur militärischen und staatlichen Nutzung vorgesehen war, wurde es im Jahr 1993 offiziell auch für den zivilen Gebrauch geöffnet und bereitgestellt [Alf10]. Zwei Jahre später wurde das GPS-System offiziell als funktionsbereit erklärt und somit in Betrieb genommen [95]. Seitdem findet es in den unterschiedlichsten Bereichen seine Anwendung z.B. in der Satellitennavigation, zum Geofencing, zur exakten Lokalisierung von Objekten, Höhenmessung, Zeitbestimmung uvm.

6.2.1.1 Ziele des GPS Damit diese Dienste zuverlässig und zufriedenstellend funktionieren, wurde das GPS-System mit den folgenden Zielen entwickelt:

- **Global Coverage:** Vor allem in Anbetracht der Einsatzmöglichkeiten im militärischen Bereich war eine globale Abdeckung des Satellitensignals unabdingbar. Nur auf diese Weise konnte eine einfache, schnelle und zuverlässige Lokalisierung auf der gesamten Erdoberfläche stattfinden. Dies war insbesondere auch aufgrund der politischen Lage zwischen den USA und der Sowjetunion während der Entwicklungs- und Umsetzungszeit von Bedeutung, da auch das russische Pendant zum amerikanischen GPS, das GLONASS, im gleichen

6.2.1.2

Zeitraum gestartet wurde. Während bei GLONASS die globale Abdeckung in den Folgejahren nicht mehr gegeben war (für eine exakte Positionsbestimmung sind mindestens 24 Satelliten notwendig, zeitweise waren jedoch lediglich 7 im Einsatz), war das GPS durchgängig global zu nutzen.

- **Wetterrobustheit:** Aufgrund der gewählten Wellenlänge von GPS-Signalen (1575 MHz bzw. 1227 MHz) haben Wetterbedingungen keinen oder nur sehr geringen Einfluss auf die Empfangsqualität des Signals. Weder dichte Bewölkung, noch Regen oder Schnee beeinträchtigt diese in für handelsübliche Empfänger messbare Weise.
- **Hohe Genauigkeit:** Innerhalb des GPS Standard Positioning Service [08] (SPS) Performance Standard verpflichtet sich der Betreiber des Systems, also das amerikanische Verteidigungsministerium respektive die amerikanische Regierung, dazu, das Signal mit einem durchschnittlichen User Range Error (URE) von weniger als 7,8 Meter mit einer Wahrscheinlichkeit von 95% auszusenden (derzeit liegt der URE bei weniger als 0,715 Meter). Dieser Wert bezieht sich jedoch lediglich auf das im Weltraum ausgestrahlte Signal und nicht auf das tatsächlich vom Nutzer empfangene Signal. Dieses ist stark abhängig von mehreren Faktoren, beispielsweise der Zustand der Atmosphäre, Qualität des Empfängers, Konstellation der Satelliten, Signalreflektionen oder -störungen, uvm. [17b] Falls Idealbedingungen vorliegen, ist es sogar möglich bei längerer Messung eine Genauigkeit im Millimeterbereich zu erreichen, aber auch sonst ist das GPS mit wenigen Zentimetern bis Metern sehr exakt.

6.2.1.2 Aufbau des Global Positioning Systems

Grundlegend kann die Funktionsweise vom GPS in drei Bereich unterteilt werden: Das Usersegment, das (Welt-) Raumsegment sowie das Kontroll- bzw. Bodensegment.

Das Raumsegment besteht aus den Satelliten, die in einer Umlaufbahn von etwa 20.200 km um die Erde kreisen. Für eine globale exakte Lokalisierung sind mindestens 24 solcher Satelliten notwendig. Damit sichergestellt werden kann, dass von jeder Position und zu jedem Zeitpunkt mindestens vier Satelliten direkt sichtbar (nicht von der Erdoberfläche verdeckt) sind, sind die Umlaufbahnen jeweils in einem Winkel von 60° zueinander verdreht. Um Ausfälle oder Ungenauigkeiten im System zu vermeiden, kreisen meist mehr Satelliten als benötigt um die Erde, teilweise bis zu 32.

Das Kontrollsegment besteht aus weltweit verteilten Kontrolleinrichtungen, die den laufenden GPS-Betrieb kontrollieren, analysieren und steuern. Sie greifen beispielsweise ein, um die Umlaufbahnen einzelner Satelliten zu berichtigen, die Uhrzeiten zwischen den Satelliten zu synchronisieren oder deren Zustand (nutzbar / nicht

6.2.1.3

nutzbar) zu aktualisieren. Somit kann das für den Nutzer lediglich unidirektional empfangbare System auch durch staatliche Kontrolleinrichtungen zur bidirektionalen Kommunikation genutzt werden.

Um das GPS-Signal nutzen zu können, sind entsprechende Empfänger dafür notwendig, die zum Usersegment gehören. Die Vielfalt an möglichen Empfangsgeräten ist sehr groß, sie reicht von extrem leistungsstarken und teuren militärischen Antennen bis hin zu preisgünstigen Empfangsmodulen mit Abmessungen im Millimeterbereich.

6.2.1.3 Funktionsweise des Global Positioning Systems Grundsätzlich beruht die Funktionsweise des GPS auf der Trilateration, also der Positionsbestimmung aus der Entfernungsmessung zu drei Referenzpunkten. Jeder aktive Satellit sendet kontinuierlich ein Signal mit seinen jeweils aktuellen Positionsdaten sowie der genauen Zeit aus. In jedem der Satelliten ist eine Atomuhr verbaut, da eine exakte Zeitbestimmung für die Positionsbestimmung unabdingbar ist, da aus der Laufzeit die Entfernung zum Satellit ermittelt werden kann. Diese Signale können anschließend von einem GPS-Modul auf der Erde empfangen und ausgewertet werden. Im einfachsten Fall sind für eine Positionsbestimmung (*Fix*) lediglich zwei Satelliten notwendig. Die beiden Entfernungskugeln zu den Satelliten besitzen eine Schnittlinie in Form eines Kreises, auf der sich der Empfänger befindet. Solange angenommen wird, dass sich der Empfänger direkt auf der Erdoberfläche befindet, kann die Erdkugel als dritte Entfernungskugel genutzt werden und die Anzahl an möglichen Positionen auf lediglich 2 Orte auf der Erdoberfläche eingeschränkt werden. Wenn der Nutzer nun über seinen ungefähren Standort Bescheid weiß (oft liegen die zwei Schnittpunkte so weit voneinander entfernt, dass es meist schon ausreicht zu wissen, auf welchem Kontinent oder ob man sich zur See oder auf dem Festland befindet), kann er einen der beiden Standorte ausschließen. Dieses Verfahren hat jedoch mehrere Nachteile. Einerseits ist eine Positionsbestimmung so nur möglich, wenn von einem Standort auf der Erdoberfläche ausgegangen wird (2D-Fix). GPS wäre somit beispielsweise im Flugverkehr nicht nutzbar. Außerdem ist keine eindeutige Positionsermittlung möglich, es hängt immer zusätzlich vom Wissen des Nutzers oder des GPS-Moduls ab, eine der beiden möglichen Positionen zu eliminieren.

Um diese beiden Ungenauigkeiten zu vermeiden, kann ein zusätzlicher dritter Satellit genutzt werden. Auch auf diese Weise ergeben sich wieder zwei Schnittpunkte, wobei ebenfalls wieder einer davon vom Empfängermodul ausgeschlossen werden muss. Dies ist in diesem Fall jedoch einfacher möglich, da sich lediglich einer der beiden Punkte in Erdnähe befindet und der jeweils andere weit entfernt im Weltall, sodass die Wahl des erdnahen Fixpunktes mit extrem hoher Wahrscheinlichkeit korrekt ist. Somit sind prinzipiell drei Satelliten ausreichend, um den dreidimensionalen Fixpunkt (Position auf der Erdoberfläche + Höhe) zu bestimmen.

6.2.1.3

Dennoch sind für eine wirklich exakte Bestimmung wie bereits erwähnt mindestens vier Satelliten notwendig. Der Grund dafür liegt in der ungenauen Zeitbestimmung und der daraus resultierenden ungenauen Entfernungsbestimmung zu den Satelliten. Die Entfernung kann aus der Signallaufzeit und der Lichtgeschwindigkeit berechnet werden. Jedes vom Satelliten ausgesendete Signal beinhaltet die aktuelle Zeit, die von der integrierten Atomuhr gegeben ist. Eine Ermittlung der Laufzeit ist also möglich, indem der Empfänger die Zeit nimmt, zu der das Signal empfangen wird, und davon die Zeit subtrahiert, zu der das Signal gesendet wurde. Dafür wären jedoch in den GPS-Empfangsgeräten ebenfalls Atomuhren nötig, die außerdem zu den Uhren der GPS-Satelliten synchronisiert sind. Dies würde jedoch dazu führen, dass die Empfänger sehr teuer werden und somit nicht für alltägliche Aufgaben genutzt werden können. Daher muss berücksichtigt werden, dass die Quarzuhren in den Empfängern eine gewisse Ungenauigkeit aufweisen. Da allerdings die Uhren der Satelliten untereinander synchronisiert sind, sind somit alle Signallaufzeiten von verschiedenen Satelliten zu einem Empfänger um denselben Betrag verfälscht. Diese Zeitverfälschung kann dann als zusätzliche vierte Variable neben den drei Positionsvariablen in das Gleichungssystem aufgenommen werden, wodurch auch ein vierter Satellit zur Lösung benötigt wird. Ohne die Zeitkorrektur spricht man von Pseudoranging, was einen großen Einfluss auf die Genauigkeit des Signals besitzt. Bereits ein Uhrenfehler von 0.01 Sekunden resultiert in einer Positionsabweichung von etwa 300km.

Neben der genauen Uhrzeit werden von jedem Satelliten auch noch weitere relevante Daten ausgesandt, wie beispielsweise die jeweiligen Bahndaten und -parameter. Diese Daten werden auf eine bestimmte Trägerfrequenz moduliert. Dafür wurde eine Frequenz von $1575,42\text{MHz}$ mit einer Wellenlänge von $19,05\text{cm}$ für die *L1-Frequenz* gewählt. Die *L2-Frequenz* mit $1227,60\text{MHz}$ und einer Wellenlänge von $24,45\text{cm}$ ist lediglich für den militärischen Gebrauch verfügbar, da auf diese Weise noch genauere Positionsbestimmungen verfügbar sind. Dieser Kanal ist verschlüsselt und ist somit für die zivile Nutzung unbrauchbar. Aus diesem Grund wird im Folgenden lediglich die öffentliche L1-Frequenz näher beschrieben.

Die Daten werden durch Phasenmodulation auf das Trägersignal moduliert. Hierbei wird die Phase des Sinussignals bei jeder Änderung des Datensignals um 180° gedreht. Auf diese Weise wird der Frequenzbereich des Trägersignals verbreitert (*spread spectrum*)

Eines der beiden Signale, das auf das Trägersignal moduliert wird, ist der *C/A-Code* (*coarse aquisition*, dt. “grobe Bestimmung”). Dieser Code ist ein pseudozufälliger Code (*PRN, pseudo random noise*), der zwar wie zufälliges Rauschen aussieht, jedoch besitzt jeder Satellit einen eindeutig festgelegten Code. Dieser hat eine Länge von 1023Chips (Chips entsprechen Bits, jedoch ohne Informationswert) und eine

6.2.1.3

Frequenz von $1023MHz$. Der Code wird lediglich zur Demodulation des Signals benötigt und besitzt daher auch keinen Informationsgehalt.

Neben dem C/A-Code werden auf das Trägersignal auch die eigentlichen Navigationsdaten aufmoduliert, dieses Signal besitzt eine Frequenz von $50Hz$. Eine komplette Nachricht besteht aus $25Frames$ mit einer Länge von jeweils $1500Bit$. Die gesamte Nachricht beinhaltet demnach also $37,5kBit$.

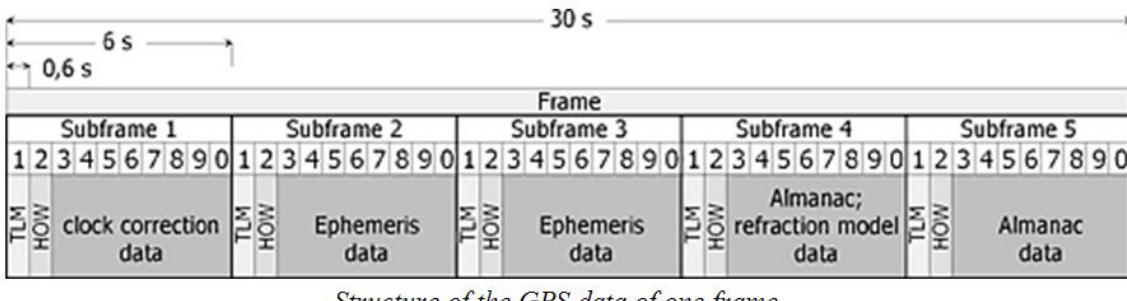


Abbildung 17: Der Aufbau eines GPS-Datenframes

Quelle: <https://gps2008.wordpress.com/gps-signals/relationship-between-the-data/>

Jeder Frame ist untergliedert in 5 Subframes, die jeweils wiederum in 10 Words unterteilt sind (siehe Abb. 17). Das erste Wort eines jeden Subframes stellt das sogenannte *Telemetry Word (TLM)* dar. Dadurch weiß der Empfänger, dass an dieser Stelle ein neuer Subframe beginnt, was vor allem für die Synchronisierung wichtig ist. An zweiter Stelle eines jeden Subframes kommt das *HOW (Hand-over Word)*, das angibt, wie viel Zeit seit dem letzten Sonntag, 0:00 Uhr vergangen ist.

Anschließend folgen im ersten Subframe Werte zur Uhrenkorrektur sowie dem aktuellen Zustand des Satelliten. Im zweiten und dritten Subframe werden die *Ephemeridendaten* des Satelliten übermittelt. Diese stellen eine Vorhersage der aktuellen Satellitenposition in Abhängigkeit von der Zeit dar. Subframe vier und fünf enthalten dagegen die *Almanachinformationen*, also Daten über die Umlaufbahnen und atmosphärische Verzögerungsparameter enthält. Diese besitzen Ähnlichkeiten zu den Ephemeriden, jedoch sind die Alamanachen länger gültig, dafür aber weniger genau.

Aus diesem Grund werden die ersten drei Subframes bei jedem Frame mitübermittelt, sodass die wichtigsten Informationen alle 30 Sekunden empfangen werden.

Dennoch kann es unter Umständen sehr lange dauern, bis die Position genau bestimmt wurde, d.h. ein Fix gefunden wurde. Hierbei spricht man vom sogenannten *TTFF*, der *Time To First Fix*. Wenn das Signal nur kurzzeitig unterbrochen wurde,

6.2.2.1

etwa bei einer Tunneldurchfahrt oder dem Passieren eines Gebäudes, muss keine aufwändige Neuberechnung stattfinden, sondern das Signal kann einfach wieder aufgenommen werden (*Reacquisition*). Dies dauert nicht länger als ein paar Sekunden. Wenn das Gerät etwa 2 - 6 Stunden keine Verbindung zu den Satelliten hatte, es beispielsweise ausgeschaltet war, kann es im *Hot Start* starten. In diesem Fall sind alle Daten wie die Uhrzeit, Almanachen und Ephemeriden noch aktuell, sodass es lediglich ca. 15 Sekunden dauert, bis ein Fix erfolgt. Bei einem *Warm Start* dagegen konnte länger als 2 - 6 Stunden keine Verbindung zu den Satelliten hergestellt werden. Hierbei sind zwar Uhrzeit und Almanachen noch gültig, jedoch sind die Ephemeridendaten (also die genaueren, aber nur kurze Zeit gültigen Bahndaten) veraltet. Bis zu einem Fix vergehen so etwa 45 Sekunden. Wenn das Gerät im *Cold Start* startet, kann dies aus mehreren Gründen erfolgen. Entweder ist der interne Speicher des Gerätes beschädigt, sodass keine aktuellen Informationen zur Uhrzeit, den Alamanachen und Ephemeriden vorliegen, oder aber der Nutzer hat sich seit dem letzten Start des Gerätes etwa 300km vom letzten bekannten Standpunkt entfernt. Auch eine längere Lagerung des Empfängers in ausgeschaltetem Zustand und ohne Batterie kann eine Ursache für den *Cold Start* sein. In diesem Fall muss eine komplette GPS Nachricht gelesen werden, um alle nötigen Informationen zu erhalten. Dies dauert bei einer Übertragungsfrequenz von $50Hz$ und bei $25Frames$ von je $1500Bit$ im schlechtesten Fall $750Sekunden$, also $12,5Minuten$.

Das Signal kann anschließend vom Nutzer empfangen und dekodiert werden. Um die Navigationsdaten sinnvoll auswerten zu können, muss er wissen, von welchem Satelliten das Signal stammt. Da, wie bereits beschrieben, von jedem Satelliten ein eindeutiger C/A-Code auf das Trägersignal moduliert wird, kann der Empfänger, dem die C/A-Codes aller Satelliten bekannt sind, die Signale miteinander vergleichen. Der C/A-Code, für den die Übereinstimmung maximal ist, weist somit auf den sendenden Satelliten hin. Dieses Autokorrelationsverfahren bietet jedoch noch einen weiteren Vorteil: Der Empfänger kann ermitteln, wie weit er die Signale zueinander verschieben muss, um eine maximale Übereinstimmung zu erreichen, womit auch die Laufzeit des Signal sehr exakt bestimmt werden kann. Auf diese Weise kann ein bestimmter Standort sehr genau bestimmt werden.

6.2.2 GPS im Kontext des Flugschreibers

6.2.2.1 Anforderungen Damit ein GPS-Empfänger als wichtiger Bestandteil des Flugschreibers eingesetzt werden kann, müssen mehrere wichtige Nebenbedingungen sowie Anforderungen berücksichtigt werden:

- **Größe & Gewicht:** Um die Flugeigenschaften des Fluggerätes, also z.B. der Drohne, in möglichst geringer Weise zu beeinflussen bzw. beeinträchtigen, soll-

6.2.2.1

te das Trackingmodul so klein und leicht wie möglich sein.

- **Genauigkeit:** Trotz einer geringen Größe sollte die Trackingeinheit dennoch nicht an Genauigkeit in Bezug auf die Geopositionen einbüßen. Um die Daten beispielsweise in Versicherungsfällen als Beweismittel auswerten zu können, muss sichergestellt sein, dass die Positionsdaten der tatsächlichen Flugroute entsprechen und nicht außerhalb eines gewissen Toleranzbereiches liegen.
- **Zuverlässigkeit:** Ähnlich wie die Genauigkeit muss auch die Zuverlässigkeit der Positionsbestimmung gewährleistet werden. Ausfälle widersprechen dem Grundprinzip eines Flugschreibers, da somit eine spätere Auswertung und Analyse des Fluges verhindert wird. Dies ist vor allem bei Unfällen oder anderen Streitfällen von Relevanz.
- **Manipulationssicherheit:** Damit eine Auswertung der Originaldaten sichergestellt werden kann, muss der Flugschreiber eine Möglichkeit zur Vermeidung von Datenmanipulationen integrieren, andernfalls könnten Gesetzeswidrigkeiten verschleiert werden. Hierbei geht es also auch um die Zuverlässigkeit der aufgezeichneten Daten, jedoch bezieht sich die Zuverlässigkeit in diesem Sinne nicht auf das zu empfangende Signal, sondern vielmehr auf die Auswertung der aufgezeichneten Flugroute.
- **Energieverbrauch:** Vgl. Abschnitt 4.1. Der Flugschreiber soll möglichst wenig Energie verbrauchen, um den Instandhaltungsaufwand für den Nutzer möglichst gering zu halten. So soll eine Batterie beispielsweise nicht nach jedem Flugbetrieb gewechselt oder geladen werden, sondern nach Möglichkeit mehrere Monate oder gar Jahre genutzt werden können. Jedoch ist der Energiebedarf auch stark von der Funktionalität sowie der Architektur des Flugschreibers abhängig; so wird bei einer Online-Datenübermittlung und starker Verschlüsselung ungleich mehr Leistung benötigt als wenn die Daten lediglich im Klartext auf ein lokales Speichermedium geschrieben werden.
- **Datenübermittlung / Datenaustausch:** Vgl. Abschnitt 4.2. Für einen Flugschreiber wäre es wünschenswert, wenn die aktuellen Flugdaten direkt an einen zentralen Server übermittelt werden, sodass auf dem Fluggerät keine lokale Speichereinheit benötigt wird. Dies hat zur Folge, dass eine nachträgliche Datenmanipulation deutlich erschwert wird. Jedoch bietet die Persistierung der Flugdaten auf einen lokalen Datenträger den Vorteil eines geringeren Energiebedarfs sowie einer erhöhten Zuverlässigkeit, da die Daten auch gespeichert werden können, falls beispielsweise keine Verbindung zum Server besteht.
- **Speicherplatz:** Falls die Daten auf einem lokalen Speicher abgelegt werden, sollte dies möglichst platzsparend geschehen. Da für eine spätere Auswertung

6.2.2.2

der Flugdaten möglicherweise historische Daten benötigt werden, muss das Speichergerät über genügend Kapazität verfügen, damit ältere Daten erst nach langer Zeit überschrieben werden. Falls keine Daten überschrieben werden dürfen, kann auch das Speichermedium, z.B. eine SD-Karte gewechselt werden, was jedoch mit zusätzlichem Aufwand und Kosten für den Nutzer verbunden ist.

- **Flexibilität / Unabhängigkeit:** Damit der Flugschreiber universell einsetzbar ist, muss dessen Architektur möglichst unabhängig vom Fluggerät gestaltet sein. Dies ermöglicht dem Nutzer auch, den Flugschreiber z.B. nach dem Kauf einer neuen Drohne unverändert weiterzunutzen, womit er nicht auf einen bestimmten Hersteller oder ein spezifisches Produkt angewiesen ist
- **Usability:** Der Flugschreiber soll einfach einzurichten, anzubringen und zu bedienen sein, sodass sich für den Endanwender lediglich ein minimaler Zusaufwand ergibt. Auf diese Weise wird auch die Akzeptanz zur Nutzung eines Flugschreibers erhöht.

Wie aus diesen Anforderungen für den Flugschreiber ersichtlich wird, sind einige der beschriebenen Eigenschaften konträr zueinander. Beispielsweise wird einerseits durch eine Online-Datenübermittlung an einen zentralen Server die Manipulationssicherheit erhöht, außerdem entfällt die Anforderung für ein lokales Speichergerät. Andererseits wird dadurch unter Umständen die Zuverlässigkeit verringert, da hierfür eine aktive Internetverbindung benötigt wird. Dies bedeutet außerdem zusätzlichen Aufwand, zusätzliches Gewicht für ein GSM-Modul sowie zusätzlichen Energiebedarf. Ein weiteres Beispiel für die Diskrepanz einzelner Anforderung ergibt sich aus der Flexibilität sowie der Usability des Flugschreibers, da dieser mit hoher Wahrscheinlichkeit weniger intuitiv zu installieren und betreiben ist, wenn er möglichst generisch gestaltet ist, um die Kompatibilität mit einer großen Zahl an Fluggeräten sicherzustellen.

6.2.2.2 Der Prototyp Um aufzuzeigen, wie ein solcher Flugschreiber umgesetzt werden könnte, wurde ein grundlegender Hard- und Softwareprototyp entwickelt, der die wichtigsten der oben genannten Anforderungen erfüllen soll. Dafür wurden verschiedene Hardwarekomponenten ausprobiert und ausgewertet.

Ausgangsbasis für die Entwicklung stellt ein Arduino Uno dar. Arduino ist eine Kombination aus quelloffener Hard- und Softwareplattform, deren Kernprodukt Einplatinencomputer darstellen, die über analoge sowie digitale Ein- und Ausgaben besitzen. Dieser single-board Mikrocontroller besitzt mehrere vorteilhafte Eigenschaften: Zum einen ist er aufgrund seiner kompakten Bauweise, die nur das nötigste enthält, sehr klein und leicht. Auch im Bezug auf die Programmierung ist der Arduino sehr leicht-

6.2.2.2

gewichtig: Die frei verfügbare Entwicklungsumgebung ermöglicht eine schnelle und unkomplizierte Prototypisierung. Darüber hinaus können die digitalen und analogen In- und Outputs genutzt werden, um damit weitere Komponenten zu verbinden.

In Abb. 18 ist der schematische Aufbau des Prototyps dargestellt. Zentrale Einheit ist der Arduino, der sowohl das GPS- als auch das SD-Karten-Modul steuert. Der Mikrocontroller wird durch eine externe Stromversorgung betrieben.

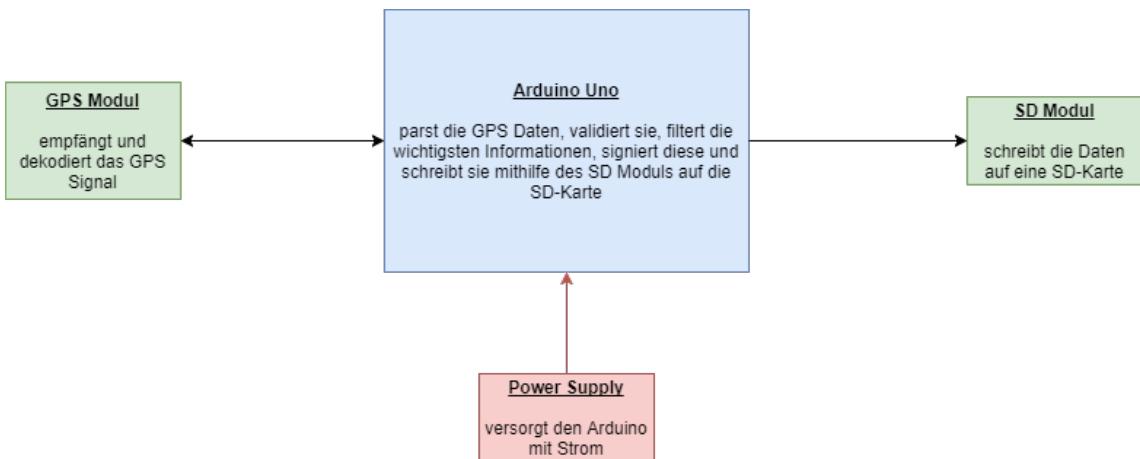


Abbildung 18: Der schematische Aufbau des Software-Prototyps

Zu Beginn des Projektes wurden mehrere Hardwarebauteile angeschafft, um eine einfache und schnelle prototypische Umsetzung zu realisieren. Die wichtigsten Komponenten hierbei waren neben dem Arduino Uno vor allem die GPS-Empfänger (vgl. Abb. 19), um die Flugdaten aufzzeichnen zu können. Aufgrund der teilweise konträren Anforderungen an den Flugschreiber konnte kein GPS-Modul gefunden werden, das alle oder die meisten Bedingungen erfüllt. Aus diesem Grund wurden drei verschiedene GPS-Trackingeinheiten eingesetzt, die jeweils unterschiedliche Stärken besitzen.

Zunächst wurden alle drei GPS-Empfänger getestet um festzustellen, ob überhaupt ein Signal empfangen wird. Bei der Einrichtung und Verbindung mit dem Arduino zeigten sich bereits die ersten Unterschiede zwischen den Modulen: Während sowohl der *USGlobalSat*- (Abb. 19a) als auch der *SainSmart-Receiver* (Abb. 19c) zusätzliche Lötverbindungen benötigen, handelt es sich beim *Adafruit GPS Logger* (Abb. 19b) um ein Arduino Shield. Dies bedeutet, dass es einfach auf den Arduino aufgesteckt werden kann, was den Aufwand deutlich verringert. Auch im Bezug auf die bereitgestellten Anleitungen und Bibliotheken unterscheiden sich die Receiver stark: Für den *SainSmart-Empfänger* wurden vom Hersteller kaum Informationen veröffentlicht, die eine genaue Belegung der PINs beschreibt, auch eine Softwarelibrary oder Beispielimplementierung wurde nicht bereitgestellt. Demzufolge war der

6.2.2.2

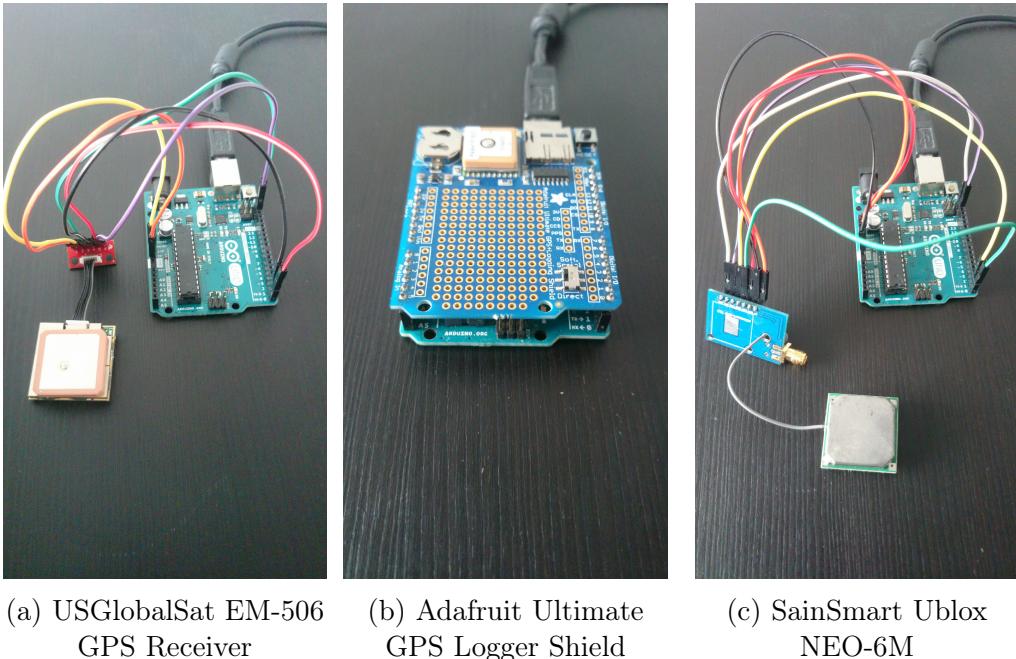


Abbildung 19: Die drei getesteten GPS-Empfänger im Vergleich

Aufwand der Inbetriebnahme vergleichsweise groß. Für das Modul des Herstellers USGlobalSat waren dagegen deutlich mehr Informationen und Hilfestellungen gegeben, auch wenn die Beschreibung an einigen Punkten sehr unklar formuliert ist. Im Gegensatz dazu ist das Logger Shield von Adafruit durchgehend klar verständlich und strukturiert dokumentiert, es werden alle Funktionalitäten des Receivers genau beschrieben und eine dedizierte Softwarelibrary angeboten, wofür auch viele Beispielimplementierungen gegeben sind.

Darüber hinaus bietet das Adafruit-Shield im Vergleich zu den anderen beiden GPS-Modulen noch weitere Vorteile: Zum einen war die Empfangsqualität auch innerhalb von Gebäuden nahe eines Fensters sehr gut. In diesen Fällen konnte das Logger Shield innerhalb kurzer Zeit einen Position-Fix erlangen, während die beiden anderen Empfänger, vor allem der SainSmart-Receiver, hier deutlich mehr Zeit benötigten und teilweise erst nach über 30 Minuten den aktuellen Standort ermitteln konnten. Des Weiteren besitzt das Logger Shield einen SD-Kartenslot, der direkt auf dem Bauteil angebracht ist, sodass für eine Speicherung der Daten kein weiteres Arduino-Modul notwendig ist, was vor allem die Kommunikation zwischen beiden Komponenten erleichtert. Nicht zuletzt ist auch die Möglichkeit, eine zusätzliche Knopfbatterie auf dem Modul anzubringen, vor allem im Hinblick auf die Zuverlässigkeit des Datenloggings von Vorteil. Auf diese Weise können sowohl der GPS-Empfänger als auch die SD-Karte weiterbetrieben werden, auch wenn das Modul von

6.2.2.2

Tabelle 10: Vergleich der drei GPS-Module basierend auf Herstellerangaben

	USGlobalSat EM-506 GPS Rec.	Adafruit Ultimate GPS Logger Shield	SainSmart Ublox NEO-6M
Sensitivität (dBm)	−163	−165	−161
Kanäle	48	66	50
Genauigkeit (m)	2,5	5 – 10	2,5
Hot Start (s)	1	keine Angabe	1
Warm Start (s)	35 (15 mit CGEE ¹⁴)	keine Angabe	27
Cold Start (s)	35 (15 mit CGEE)	keine Angabe	27
Spannung (V)	4,5 – 6,5	3	3 – 5
Leistung (mW)	358	60	Stromstärke nicht bekannt
Abmessungen (mm)	30 × 30 × 10,7	96 × 53 × 6,7	16 × 12,2 × 2,4
Gewicht (g)	16	23	22,7
Integr. Backupbatterie	✗	✓	✓
Integr. SD-Slot	✗	✓	✗

der eigentlichen Stromversorgung getrennt ist. All diese Vorteile führten schließlich zur Entscheidung, für den Prototypen das Adafruit Ultimate GPS Logger Shield zu nutzen.

Zunächst mussten dafür die empfangenen GPS-Signale geparst werden. Für diesen Zweck sind jedoch die oben beschriebenen GPS-Rohdaten ungeeignet. Einerseits enthalten diese selbstverständlich noch nicht die Positionsangabe, da sie lediglich vom Satelliten ausgesendet werden, andererseits enthalten diese auch viele zusätzliche Informationen, die für den Zweck des Flugschreibers nicht relevant sind. Daher werden die Signale vom Empfängermodul ausgewertet und in Positionsdaten umgerechnet. Um eine gewisse Kompatibilität zu ermöglichen und den Datenaustausch von GPS-Informationen zu erleichtern, wurde im Jahr 1983 der sogenannte NMEA 0183-Standard entwickelt. Dieser beschreibt mehrere Formate, in denen Navigations-

¹⁴Client Generated Extended Ephemeris. Technologie, mit der die Satellitenpositionen für mehr als drei Tage vorausgesagt werden können

6.2.2.2

daten dargestellt werden können. Der Standard ist im Bereich der GPS-Navigation sehr weit verbreitet, sodass dieser auch im Rahmen dieses Projektes genutzt wurde. Hierfür sind insbesondere zwei verschiedene NMEA 0183-Formate (“Sätze”) von Relevanz: Der GPGGA-Satz liefert detaillierte Informationen über den aktuellen Fix, dazu gehören beispielsweise die aktuellen 3D-Positions- sowie Genauigkeitsdaten. Ein Beispielsatz ist entsprechend der Definition folgendermaßen aufgebaut:

\$GPGGA,213845,4846.747,N,92.227,E,1,07,0.9,545.4,M,46.9,M,08,AAAA*47

Einzelne Informationen sind jeweils durch ein Komma getrennt, der GPGGA-Satz enthält jeweils 15 Informationsteile. Der erste Informationsteil beschreibt, um welche Art von NMEA-Satz es sich handelt. An zweiter Stelle folgt die genaue Uhrzeit, zu der der Fix erfolgt ist, in diesem Fall um 21:38:45 UTC. Anschließend werden die genauen Positionsdaten in Breiten- und Längengraden angegeben. Wichtig ist hierbei, dass die Angaben nicht in Dezimalschreibweise erfolgen, also etwa 48, 46°, sondern in Minutenbeschreibung, also 48°46.747'. N bzw. E steht hierbei für “nördlich” respektive “östlich”. Die Ziffer im siebenten Informationsteil beschreibt die Qualität des Fixes. Hierfür gibt es mehrere Möglichkeiten, die wichtigsten sind jedoch 0 (kein Fix) und 1 (GPS Fix). Danach folgt die Anzahl der Satelliten, die getrackt werden, d.h. die Satelliten, die zur Berechnung der Position sichtbar waren und genutzt wurden. Der anschließende Wert enthält die HDOP, was für *horizontal dilution of precision* steht. Dieser Wert beschreibt die relative horizontale Genauigkeit der Positionsdaten. Je geringer dieser Wert ist, desto besser ist die Datenqualität. Der zehnte Informationsteil gibt Auskunft über die aktuelle Höhe des Empfängers über dem mittleren Meeresniveau, zusammen mit der entsprechenden Einheit, hier in Meter. Auch der nachfolgende Wert beschreibt die Höhe, hier jedoch in Bezug auf den Geoid. Anschließend folgt die Zeit seit dem letzten Update der Korrekturdaten sowie die ID der Basisstation. Am Ende des GPGGA-Satzes ist jeweils eine Checksumme enthalten, gekennzeichnet durch das Sternchen.

Der zweite wichtige NMEA-Datensatz ist das GPRMC-Format, dessen Bezeichnung für *Recommended Minimum Sentence C* steht, und beispielhaft folgendermaßen aussieht:

\$GPRMC,213845,A,4846.747,N,92.227,E,022.4,084.4,290817,003.1,W*6A

Auch hier sind die Informationseinheiten durch Kommas getrennt und auch beim RMC steht an zweiter Stelle die aktuelle Uhrzeit. Anschließend folgt entweder der Buchstabe *A*, der für *active* steht, wenn also ein Fix erfolgt ist, oder *V* für *void*, falls (noch) keine Position bestimmt werden konnte. Nach den aktuellen Koordinaten wird die aktuelle Geschwindigkeit über Grund in Knoten sowie die derzeitige Bewegungsrichtung in Grad übermittelt. Die beiden Informationseinheiten vor der Checksumme stellen das Datum sowie die Deklination (Missweisung) in Grad dar

6.2.2.2

(hier: $3,1^\circ$ in westlicher Richtung).

Sowohl die GPGGA- als auch die GPRMC-Sätze werden geparsst. Da beide Sätze jeweils Informationen enthalten, die im jeweils anderen Format nicht enthalten sind, werden die Daten beider Sätze in Kombination benötigt. So enthält beispielsweise GPGGA die aktuelle Höhe, was insbesondere für einen Flugschreiber von Bedeutung ist, wohingegen mit GPRMC die Geschwindigkeit und Bewegungsrichtung ermittelt werden kann.

Die Sätze werden in regelmäßigen Intervall vom Arduino über das GPS-Modul abgefragt. Die Update-Rate wurde in diesem Fall auf 1 Hz festgelegt, da ein Datensatz pro Sekunde für gewöhnlich ausreicht um die aktuellen Navigationsdaten auszuwerten und dadurch ein Kompromiss zwischen der Genauigkeit der Daten (zeitliche Auflösung) und der Speichereffizienz eingegangen wird. Das Lesen und Parsen der Daten vom GPS-Modul erfolgt mithilfe eines Interrupts, sodass sichergestellt werden kann, dass jedes Zeichen der GPGGA- und GPRMC-Sätze ordnungsgemäß gespeichert werden kann und nicht durch das nächste Zeichen überschrieben wird. Sobald ein kompletter Datensatz gelesen wurde, also nach der Prüfsumme das Wagenrücklauf- und Zeilenvorschubzeichen erfolgt, wird die Checksumme überprüft. Dafür wird für jedes Zeichen zwischen dem Anfangszeichen "\$" und dem Endzeichen "*" jeweils die XOR-Operation mit dem jeweils nächsten Zeichen ausgeführt. Die hexadezimale Darstellung des Ergebnisses muss dann der Prüfsumme entsprechen. Sollte es zu Fehlern beim Empfang oder Parsen gekommen sein und die Werte nicht übereinstimmen, wird der Datensatz einfach verworfen. Stimmen sie jedoch überein, sind die Daten nun prinzipiell bereit zur Speicherung auf dem lokalen Datenträger. Jedoch wäre eine solche Speicherung nicht manipulationssicher, sodass der Anwender die Daten nach einem Zwischenfall zu seinen Gunsten abändern oder sogar von Beginn an falsche Daten in das System einspeisen kann. Um dies zu verhindern, müssen die Informationen manipulationssicher abgelegt werden. Da die Flugdaten für den Anwender selbst jedoch ohne Bedenken auch gelesen werden dürfen, besteht keine Notwendigkeit, die Datensätze zu verschlüsseln. Somit reicht es aus, die Daten mit einer Signatur zu versehen. Auf diese Weise wird sichergestellt, dass die Navigationsdaten nicht unbemerkt manipuliert werden können.

Dafür wird zunächst eine Hashfunktion auf die einzelnen Datensätze angewandt und die resultierenden Werte anschließend mit dem privaten Schlüssel des Systems verschlüsselt.

Grundlage dafür bildet der *Ed25519*-Algorithmus¹⁵, der als public-key Signatursystem mehrere Vorteile bietet: Er ist sehr schnell, was vor allem für die zeitkritische Aufzeichnung der Flugdaten wichtig ist. Außerdem bietet er eine sehr hohe

¹⁵<https://ed25519.cr.yp.to/>

6.2.2.2

Sicherheit, die mit einer RSA-Verschlüsselung mit einer *3000Bit* - Schlüssellänge vergleichbar ist. Nicht zuletzt ist auch das Hashing kollisionsresilient, d.h. auch eine mögliche Kollision bricht das System nicht. Aus technischer Sicht basiert *Ed25519* auf elliptischen Kurven bzw. genauer auf einer Twisted Edwards Kurve.

Da für den Prototypen kein *Trusted-Platform-Modul* verfügbar war, mithilfe dessen ein Schlüsselpaar für das GPS-Modul bzw. den Arduino bereitgestellt werden kann, wurde einfach zu Beginn jeder Session ein privater und ein dazugehöriger öffentlicher Schlüssel generiert. Diese besitzen jeweils eine Größe von lediglich 255 Bit. Mit der gehaschten Nachricht (also der NMEA-Sätze) sowie dem privaten Schlüssel kann diese schließlich signiert werden. Die Signatur ist unabhängig von der Größe der Originalnachricht lediglich 64 Byte groß, da es sich dabei um eine komprimierte Version einer längeren Signatur handelt. Auf diese Weise können die Nachrichten nun nicht mehr unbemerkt manipuliert werden (natürlich vorausgesetzt, dass der private Schlüssel nicht bekannt ist).

Dies verhindert jedoch nicht, dass vom Nutzer dennoch einfach Daten unbemerkt gelöscht werden können. Hierfür bieten sich verschiedene Möglichkeiten an, um dies zu umgehen: Eine Option ist, anstatt der einzelnen Nachrichten jeweils die komplette Datei zu signieren. Dies hat jedoch den Nachteil, dass eine Datei sehr groß werden kann, was sich auf die Performanz beim Hashing und der Verschlüsselung auswirkt. Zudem kann so nicht verhindert werden, dass einfach die komplette Datei gelöscht wird, ebenso kann die Löschung nicht verhindert werden, sondern lediglich eine Manipulation festgestellt werden. Eine schnellere Alternative dazu bietet die Berechnung einer Checksum: Beispielsweise kann die Anzahl aller bereits gelogten Nachrichten ebenfalls signiert und Datei angehängt werden, was jedoch ebenfalls nicht das Problem, dass Nachrichten gelöscht werden können, löst. Dies kann zum Beispiel dadurch behoben werden, dass für das Speichermedium lediglich eine Schreibberechtigung vergeben wird, wenn die schreibende Komponente in Besitz des Schlüssels ist.

Nachdem die digitale Signatur zu einer Nachricht ermittelt wurde, kann diese zusammen mit der Originalnachricht auf die SD-Karte geschrieben werden. Da es hierfür bereits Standardbibliotheken gibt, konnte dies ohne große Probleme umgesetzt werden.

Somit beinhaltet der Arduino-Prototyp nun die wichtigsten Funktionalitäten für den Flugschreiber. Dennoch stellte sich bei der Implementierung heraus, dass der Code für den Arduino in der beschriebenen Form nicht lauffähig ist. Der Grund dafür ist der begrenzte Flash-Speicher des Arduino Uno. Dort stehen lediglich *32kB* Speicher zur Verfügung, wovon allein der Bootloader *5kB* benötigt. Der Sketch, wie Arduino-Programme genannt werden, wurde bereits auf das nötigste gekürzt und

6.2.2.2

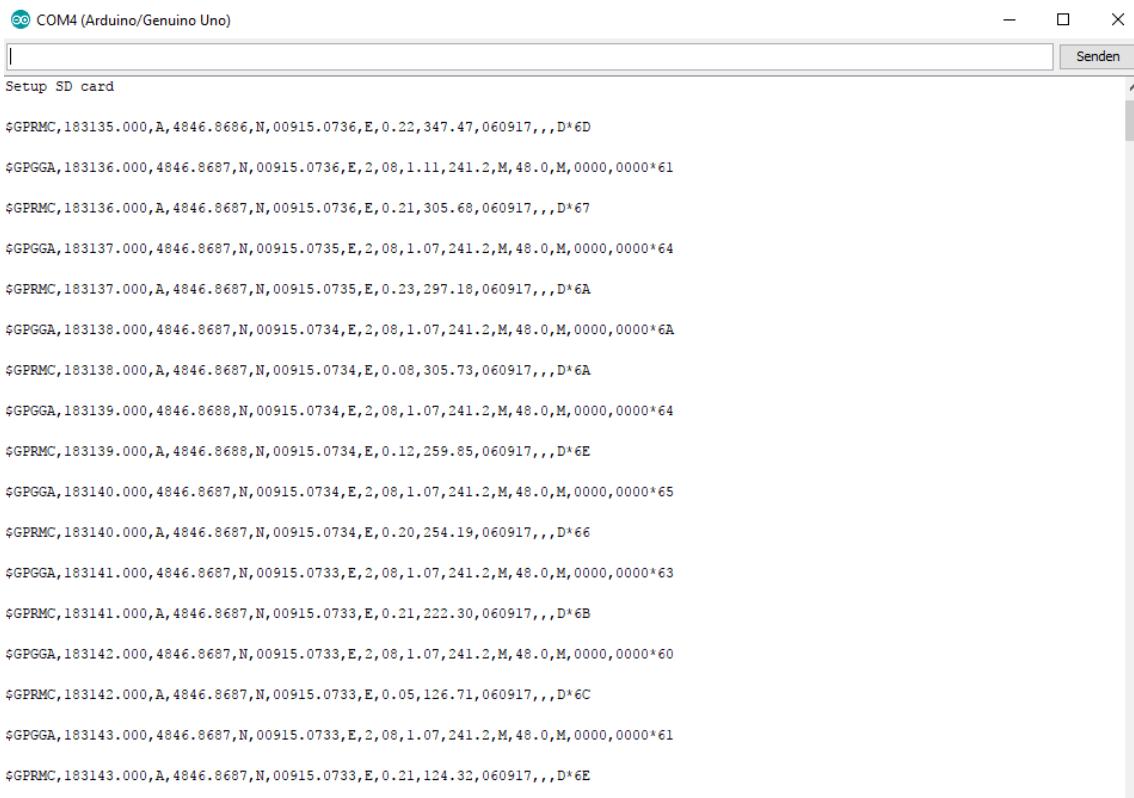
auf das wichtigste beschränkt, jedoch wird der verfügbare Speicherplatz dennoch überschritten. Der Grund dafür liegt in den vergleichsweise großen Bibliotheken, die hier zum Einsatz kommen. Allein die GPS-Bibliothek, die unverzichtbar für das Auslesen, Parsen und Aufbereiten der GPS-Signale ist, belegt 17,3kB. Ähnlich groß sind auch die drei restlichen Bibliotheken *SoftwareSerial* (17kB), *SD* (17,9kB) sowie *Crypto/Ed25519* (24,2kB). Die Maximalgröße wird also um mehr als das Doppelte überschritten, womit auch kleine Optimierungen wie z.B. die Wahl kleinerer Datentypen nicht zum Erfolg führt. Dieses Problem könnte dadurch gelöst werden, dass anstatt des *Arduino Uno* ein *Arduino Mega 2560* eingesetzt wird. Dieser besitzt mit 256kB acht mal soviel Flash-Speicher, wodurch genügend Platz für die zuvor genannten Bibliotheken bereit steht. Jedoch war diese Problematik zu Beginn des Projektes noch nicht bekannt bzw. absehbar, wodurch beim Kauf nicht auf den geringen Speicherplatz geachtet wurde. Dieses Wissen ist vor allem für zukünftige Arbeiten mit Mikrocontrollern hilfreich. In Zeiten von immer schnelleren und kleineren Computern mit immer größeren Speicherkapazitäten werden Probleme im Bezug auf Speicherkapazitäten im Bereich von wenigen Kilobytes nicht mehr bedacht.

Um dennoch die Funktionstüchtigkeit des Softwareprototypen überprüfen zu können, wurde der Sketch in zwei Teile aufgeteilt: Im ersten Sketch ist die GPS-Logging-Funktionalität implementiert, d.h. es werden die GPS-Signale geparsst, verarbeitet und auf die SD-Karte geloggt. Außerdem werden die Daten auch parallel über den seriellen Monitor ausgegeben (vgl. Abb. 20)

Wie in Abbildung 20 ersichtlich ist, werden die NMEA-Sätze (*GPGGA* und *GPRMC*) korrekt ausgegeben. Auch das Logging auf die SD-Karte funktioniert wie erwartet.

Im zweiten Sketch dagegen wurde die Signatur-Funktionalität umgesetzt. Hier werden ein privater und ein öffentlicher Schlüssel generiert und mithilfe derer anschließend die Signatur zu einem NMEA-Satz berechnet. Die Ausgabe ist in Abb. 21 dargestellt.

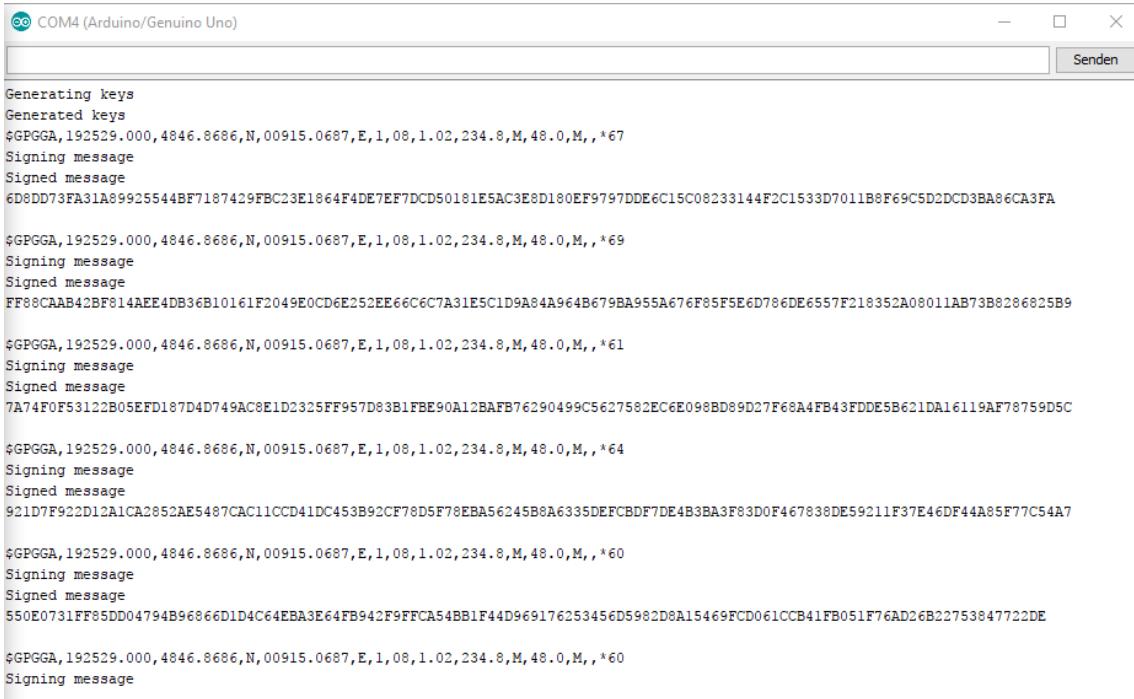
6.2.2.2



The screenshot shows the Arduino Serial Monitor window titled "COM4 (Arduino/Genuino Uno)". The window has a "Senden" button in the top right corner. The text area displays a series of GPS NMEA messages. The messages include:

- \$GPRMC,183135.000,A,4846.8686,N,00915.0736,E,0.22,347.47,060917,,,D*6D
- \$GPGGA,183136.000,4846.8687,N,00915.0736,E,2,08,1.11,241.2,M,48.0,M,0000,0000*61
- \$GPRMC,183136.000,A,4846.8687,N,00915.0736,E,0.21,305.68,060917,,,D*67
- \$GPGGA,183137.000,4846.8687,N,00915.0735,E,2,08,1.07,241.2,M,48.0,M,0000,0000*64
- \$GPRMC,183137.000,A,4846.8687,N,00915.0735,E,0.23,297.18,060917,,,D*6A
- \$GPGGA,183138.000,4846.8687,N,00915.0734,E,2,08,1.07,241.2,M,48.0,M,0000,0000*6A
- \$GPRMC,183138.000,A,4846.8687,N,00915.0734,E,0.08,305.73,060917,,,D*6A
- \$GPGGA,183139.000,4846.8688,N,00915.0734,E,2,08,1.07,241.2,M,48.0,M,0000,0000*64
- \$GPRMC,183139.000,A,4846.8688,N,00915.0734,E,0.12,259.85,060917,,,D*6E
- \$GPGGA,183140.000,4846.8687,N,00915.0734,E,2,08,1.07,241.2,M,48.0,M,0000,0000*65
- \$GPRMC,183140.000,A,4846.8687,N,00915.0734,E,0.20,254.19,060917,,,D*66
- \$GPGGA,183141.000,4846.8687,N,00915.0733,E,2,08,1.07,241.2,M,48.0,M,0000,0000*63
- \$GPRMC,183141.000,A,4846.8687,N,00915.0733,E,0.21,222.30,060917,,,D*6B
- \$GPGGA,183142.000,4846.8687,N,00915.0733,E,2,08,1.07,241.2,M,48.0,M,0000,0000*60
- \$GPRMC,183142.000,A,4846.8687,N,00915.0733,E,0.05,126.71,060917,,,D*6C
- \$GPGGA,183143.000,4846.8687,N,00915.0733,E,2,08,1.07,241.2,M,48.0,M,0000,0000*61
- \$GPRMC,183143.000,A,4846.8687,N,00915.0733,E,0.21,124.32,060917,,,D*6E

Abbildung 20: Die Ausgabe des GPS-Sketches auf dem seriellen Monitor



The screenshot shows the Arduino Serial Monitor window titled "COM4 (Arduino/Genuino Uno)". The window has a "Senden" button in the top right corner. The text area displays a series of messages related to key generation and signing. The messages include:

- Generating keys
- Generated keys
- \$GPGGA,192529.000,4846.8686,N,00915.0687,E,1,08,1.02,234.8,M,48.0,M,,*67
- Signing message
- Signed message
- 6D8DD73FA31A89925544BF7187429FBC23E1864F4DE7EF7DCD50181E5AC3E8D180EF9797DDE6C15C08233144F2C1533D7011B8F69C5D2DCD3BA86CA3FA
- \$GPGGA,192529.000,4846.8686,N,00915.0687,E,1,08,1.02,234.8,M,48.0,M,,*69
- Signing message
- Signed message
- FF88CAAB42BF814AEE4DB36B10161F2049E0CD6E252EE66C6C7A31E5C1D9A84A964B679BA955A676F85F5E6D786DE6557F218352A08011AB73B8286825B9
- \$GPGGA,192529.000,4846.8686,N,00915.0687,E,1,08,1.02,234.8,M,48.0,M,,*61
- Signing message
- Signed message
- 7A14F0F53122B05EFD187D4D749AC8E1D2325FF957D83B1FBE90A12BAFB76290499C5627582EC6E098BD89D27F68A4FB43FDDE5B621DA16119AF78759D5C
- \$GPGGA,192529.000,4846.8686,N,00915.0687,E,1,08,1.02,234.8,M,48.0,M,,*64
- Signing message
- Signed message
- 921DF922D12A1CA2852AE5487CAC11CCD41DC453B92CF78D5F78EBA56245B8A6335DEFBCDF7DE4B3BA3F83D0F467838DE59211F37E46DF44A85F77C54A7
- \$GPGGA,192529.000,4846.8686,N,00915.0687,E,1,08,1.02,234.8,M,48.0,M,,*60
- Signing message
- Signed message
- 550E0731FF85DD04794B96866D1D4C64EBA3E64FB942F9FFCA54BB1F44D969176253456D5982D8A15469FC061CCB41FB051F76AD26B22753847722DE
- \$GPGGA,192529.000,4846.8686,N,00915.0687,E,1,08,1.02,234.8,M,48.0,M,,*60
- Signing message

Abbildung 21: Die Ausgabe des Signatur-Sketches auf dem seriellen Monitor

6.2.2.2

Diese Ausgaben konnten jedoch nicht auf die SD-Karte geloggt werden, da auch hier zu wenig Speicherplatz zur Verfügung steht.

Somit wurde gezeigt, wie der Softwareprototyp implementiert werden kann. Da im Laufe des Projektes mehrere Schwierigkeiten zu bewältigen waren (beispielsweise funktionierten zu Beginn zwei der drei GPS-Empfänger nicht, dazu die geringe Speicherplatzkapazität des Arduino, fehlendes TPM-Modul, . . .), konnte dieser nicht wie ursprünglich geplant umgesetzt werden.

Wenn der Sketch jedoch auf einem Mikrocontroller mit größerem Flash-Speicher betrieben wird, sind so bereits die wichtigsten Anforderungen an einen Flugschreiber erfüllt. Zusammen mit einer induktiven Stromversorgung kann der Mikrocontroller über längere Zeit betrieben werden, das Gesamtgewicht ist eher gering, die Zuverlässigkeit und Genauigkeit des GPS-Moduls ist sehr hoch, die Daten können auf einer lokalen SD-Karte gespeichert werden, was zudem sehr platzsparend ist¹⁶ und auch die Flexibilität ist gegeben. Allein die Sicherheit stellt die Schwachstelle dar, da der private Schlüssel in der jetzigen Form nicht vor dem Nutzer verborgen bleibt bzw. dieser nicht durch eine TPM verwaltet wird. Die Integration eines TPM stellt somit also einen interessanten Ansatz zur Weiterentwicklung des Flugschreiber-Projekts dar.

¹⁶max. 82Bytes pro NMEA-Satz + 64Bytes Signatur; bei etwa einer Nachricht + Signatur pro Sekunde reicht eine standardmäßige 16GB-Speicherkarte somit über 3,7Jahre bei kontinuierlicher Aufzeichnung

7 Schlussbetrachtung

In diesem Kapitel wird die Projektarbeit im Fazit zusammengefasst und Erweiterungsmöglichkeiten für zukünftige Arbeiten im Ausblick genannt.

7.1 Fazit

Das Ziel des Projekts war es, die Anforderungen an einen Flugdatenschreiber für Drohnen zu erfassen und Konzepte zur Umsetzung zu erarbeiten. Außerdem sollten einige Technologien anhand von Feldversuchen untersucht werden.

Wie die Marktanalyse zu Beginn des Projekts zeigt, sind keine Produkte vorhanden, die alle oder die meisten der von uns festgelegten Anforderungen eines Flugschreibers abdecken. Diese Produkte bieten jedoch eine gute technische Orientierungshilfe hinsichtlich der Realisierbarkeit im industriellen Ausmaß.

Die zunehmend eingeschränktere rechtliche Lage für die Nutzung von Drohnen fordert geradezu ein Produkt wie den von uns untersuchten Flugschreiber.

Aus den praktischen Feldtests zur Energieversorgung und sicherem GPS-Logging gehen einige Erkenntnisse hervor. Das Energy Harvesting mit induktiver Kopplung bringt überraschend gute Ergebnisse und könnte neben der mechanischen Kopplung über einen Dynamo für den Betrieb des GPS-Moduls und Arduino Uno ausreichen.

Durch die ausführliche Einarbeitung in die GPS-Technologie sowie deren Protokoll konnten die resultierenden Daten hinsichtlich der Geschwindigkeit und Datenmenge zur sicheren Speicherung, optimal analysiert werden. Der praktische Einsatz verschiedener GPS-Hardwaremodule für den Arduino Uno hat gezeigt, dass hier enorme Unterschiede in der Funktionsweise, Zuverlässigkeit und Dokumentation bestehen.

Von einem solchen potentiellen Produkt würden vor allem drei Parteien profitieren. Der **Gesetzgeber** erlangt mehr Transparenz und Sicherheit für die Kontrolle und Strafverfolgung, welcher den Einsatz eines Flugschreibers als gesetzlich verpflichtend erklären müsste. Die **Versicherungen** sparen Ressourcen für die Gutachten von Einzelfällen, welche dank der aufgezeichneten fälschungssicheren Flugroute leicht nachvollziehbar sind. Daraus resultiert ebenso der Wettbewerbsvorteil durch vergünstigte Versicherungen, welche den Einsatz eines Flugschreibers zur Vertragsgrundlage erklären. Und abschließend der **Pilot**, welcher neben der Auswertung der Flugdaten einen rechtsverbindlichen Nachweis seiner Flugroute vorweisen kann.

Trotz der genannten Herausforderungen mit den beiden bestellten Drohnen können die Ergebnisse des theoretischen Teils sowie die Erfahrungen der Feldtests für

zukünftige Projekte weiter verwendet werden.

7.2 Ausblick

Für auf dieser Arbeit aufbauende Projekte werden abschließend einige wichtige Punkte genannt, die beachtet werden müssen.

Durch die Einschränkung des Speicherplatzes am Arduino Uno sowie der verfügbaren Projektzeit war es leider nicht möglich, ein TPM-Chip im Zusammenspiel mit dem GPS-Modul zu testen. Aus dieser Einschränkung folgt ebenso der Hinweis für zukünftige Test einen Mikrocontroller mit mehr verfügbarem Speicherplatz einzusetzen. Erst dann kann der Gesamtlauf von GPS-Empfang und Signatur der Daten durch ein TPM in Kombination getestet werden.

Für ein marktreifes Produkt müssen ebenfalls folgende Punkte untersucht werden:

- Materialbelastbarkeiten (vgl. Abschnitt 3)
- Flugeinschränkungen der Drohne unter Einsatz eines Energy Harvesting-Moduls
- Aerodynamik von zusätzlich angebrachter Hardware
- Maximal zulässiges Zusatzgewicht bei gängigen Drohnen auf dem Markt
- Datenschutzregelungen bei Übermittlung der Daten
- Optimierung der Positionsbestimmung
 - Was passiert, wenn das GPS-Signal nicht empfangbar oder zu ungenau ist? Darf der Nutzer dennoch fliegen? Falls nicht: Wie kann ein Start verhindert werden? Falls doch: Wer haftet dann?
 - Mittels GPS ist lediglich eine relative und dadurch eher ungenaue Höhenbestimmung möglich (Höhe in Bezug auf das mittlere Meeresniveau). Um die absolute Höhe zum Grund zuverlässig bestimmten zu können, ist daher beispielsweise zusätzlich ein Altimeter notwendig.

Literatur

- [08] *Global Positioning System*. Standard Positioning Service Performance Standard. Version 4. 1. Sep. 2008. URL: <http://www.gps.gov/technical/ps/2008-SPS-performance-standard.pdf> (besucht am 01.09.2017) (siehe S. 37).
- [17a] *Global Positioning System History*. 4. Aug. 2017. URL: https://www.nasa.gov/directorates/heo/scan/communications/policy/GPS_History.html (besucht am 01.09.2017) (siehe S. 36).
- [17b] *GPS Accuracy*. 15. Aug. 2017. URL: <http://www.gps.gov/systems/gps/performance/accuracy/> (besucht am 01.09.2017) (siehe S. 37).
- [95] *GPS FULLY OPERATIONAL STATEMENT OF 1995*. 17. Juli 1995. URL: <https://www.navcen.uscg.gov/?pageName=global> (besucht am 01.09.2017) (siehe S. 36).
- [Alf10] R. Alfred. *DEC. 8, 1993: Location, Location, Location*. 12. Aug. 2010. URL: <https://www.wired.com/2010/12/1208-gps-open-civilians/> (besucht am 01.09.2017) (siehe S. 36).
- [IBM17] IBM. *Security policy and objectives*. 2017. URL: https://www.ibm.com/support/knowledgecenter/en/ssw_i5_54/rzaj4/rzaj40j0securitypolco.htm (besucht am 01.09.2017) (siehe S. 18).
- [Ric02] H. Richter. *Verschlüsselung im Internet*. 6. März 2002. URL: <http://www.runway.ch/images/stories/dienstleistungen/m5%20verschluesselung%20im%20internet.pdf> (besucht am 01.09.2017) (siehe S. 20).
- [Sch17] R. Schmitz. *IT Security, Introduction, Security Objectives*. 15. Aug. 2017. URL: https://www.hdm-stuttgart.de/studenten/stundenplan/pers_stundenplan/stundenplan_bearbeiten/skript_download/5212593/SS17/its2017-01-intro.pdf (besucht am 11.09.2017) (siehe S. 18).