



Entwicklung einer App zur Steuerung und Datenauswertung einer Drohne zur Luftqualitätsmessung

Studienarbeit

an der Dualen Hochschule Baden-Württemberg Stuttgart

von

Julian Riegger

04.06.2018

Bearbeitungszeitraum
Matrikelnummer, Kurs
Ausbildungsfirma
Betreuer

xx Wochen
1577610, STG-TINF15-ITA
Robert Bosch GmbH, Stuttgart
Thilo Ackermann, Rene Lasse

Erklärung

Ich erkläre hiermit ehrenwörtlich:

1. dass ich meine Studienarbeit mit dem Thema *Entwicklung einer App zur Steuerung und Datenauswertung einer Drohne zur Luftqualitätsmessung* ohne fremde Hilfe angefertigt habe;
2. dass ich die Übernahme wörtlicher Zitate aus der Literatur sowie die Verwendung der Gedanken anderer Autoren an den entsprechenden Stellen innerhalb der Arbeit gekennzeichnet habe;
3. dass ich meine Studienarbeit bei keiner anderen Prüfung vorgelegt habe;
4. dass die eingereichte elektronische Fassung exakt mit der eingereichten schriftlichen Fassung übereinstimmt.

Ich bin mir bewusst, dass eine falsche Erklärung rechtliche Folgen haben wird.

Stuttgart, 04.06.2018

Julian Riegger

Abstract

Logfiles beinhalten eine große Menge an Daten, deren Analyse bei der Suche nach Fehlern und der Überwachung einer IT Infrastruktur eine große Hilfe sind. Dabei stellen sich mehrere Herausforderungen. Die Erste ist, dass Logfiles textbasiert sind. Der Nachteil hierbei ist, dass im Vergleich zu einer Datenbank oder einer XML Datei textbasierte Dateien keine klar auslesbare oder durchsuchbare Struktur besitzen. Die Zweite ist, dass Systeme so viele Informationen wie möglich loggen und dadurch die nützlichen bzw. wichtigen Informationen erst herausgefiltert werden müssen.

Für die Analyse von textbasierten Daten eignet sich sehr gut das MapReduce Modell. Außerdem lässt sich das Modell sehr einfach skalieren und auf mehrere Programmläufe verteilen (Master-Worker). Das Apache Hadoop Projekt stellt sowohl für MapReduce, als auch für die Verwaltung von mehreren Programmläufen, ein Basisframework bereit, mit welchem die Entwicklung eines Analyseprogramms durchgeführt werden soll.

Ziel dieser Arbeit ist die Entwicklung einer prototypischen Anwendung zur formatunabhängigen Analyse von Logfiles unter Zuhilfenahme von Apache Hadoop MapReduce. Die Anwendung soll die bisher vorhandenen Monitoring Systeme innerhalb der Infrastruktur ergänzen, wodurch Informationen über den Zustand des Systems schneller erhoben werden können. Des Weiteren sollen aufkommende Fehler besser erkannt werden, um die Reaktionszeit auf diese zu optimieren.

Inhaltsverzeichnis

Abkürzungsverzeichnis	I
Abbildungsverzeichnis	II
Tabellenverzeichnis	III
Listings	IV
1 Einleitung	1
1.1 Problemstellung	1
1.2 Aufgabenstellung	2
2 Theoretische Grundlagen	3
2.1 Luftqualität	3
2.1.1 Indices	3
2.1.2 Schadstoffe	5
2.2 Hardware	6
2.2.1 DJI Phantom 3 Standard	6
2.2.2 iPad 3	9
2.3 iOS-Appentwicklung	10
2.3.1 Model View Controller (MVC)	10
2.3.2 Cocoa	11
2.3.3 Xcode	14
2.3.4 SWIFT	15
2.3.5 DJI-Software Development Kit (SDK)	15
Literatur	i
Anhang	ii

Abkürzungsverzeichnis

API	Application Programming Interface
AQI	Air Quality Index
CAQI	Common Air Quality Index
CO	Kohlenstoffmonoxid
EPA	Environmental Protection Agency
IDE	Integrated Development Environment
GUI	Graphical User Interface
GPS	Global Positioning System
MVC	Model View Controller
NO₂	Stickstoffdioxid
NO_x	Stickoxide
O₃	Ozon
SDK	Software Development Kit
SO₂	Schwefeldioxid
OS	Betriebssystem

Abbildungsverzeichnis

2.1	DJI Phantom 3 Standard	7
2.2	DJI Phantom 3 Standard - Fernbedienung	8
2.3	iPad 3	10
2.4	MVC Design-Pattern Diagramm mvc1	11
2.5	Externe Bibliotheken in Xcode	13
2.6	Xcode-Oberfläche	14

Tabellenverzeichnis

2.1	AQI Übersicht	4
2.2	DJI Phantom 3 Standard - technische Daten	9

Listings

2.1 Podfile Beispiel	12
--------------------------------	----

1 Einleitung

In den letzten Jahren bekam das Thema der Luftqualität immer mehr Aufmerksamkeit und gewinnt immer mehr an Bedeutung in der Tagespolitik sowie in der Industrie. Hier ist vor allem die Automobilindustrie in den Fokus gerückt, da die Verbrennungsmotoren in einer sehr emotional geführten Debatte für einen Großteil der schlechten Luft in Großstädten verantwortlich gemacht werden. Nun trägt nicht nur der Verkehr sondern auch die Industrie mit verschiedenen Fabriken, wie auch andere Faktoren, wie zum Beispiel das heizen mit Holz im Winter, zur Verschlechterung der Luftqualität bei. Es wurden in den letzten Jahren immer mehr Messstationen in großen und kleineren Städten platziert, um die Luftqualität zu überwachen.

Zum Thema Luftqualität stellen sich folgende Fragen, welche in der folgenden Arbeit teilweise beantwortet werden sollen.

- Was ist Luftqualität?
- Kann man die Luftqualität messen?
- Was sind Faktoren für die Luftqualität?
- Was sind für den Menschen gefährliche Faktoren in der Luft?

1.1 Problemstellung

Von den im Kapitel Einleitung genannten Messstationen ist in der Region Stuttgart die Messstation am Neckartor die bekannteste. Diese misst die Luftqualität aber nur an einer Stelle. Hierbei kann man diskutieren, ob dieser Wert überhaupt aussagekräftig ist oder nicht. Es könnte sein, dass die Wahl für den Ort der Messstation missglückt ist und die gemessenen Werte deshalb nicht aussagekräftig sind.

Ein weiterer Aspekt, welcher zu berücksichtigen ist, sind die Auswirkungen des Wetters auf die Luftqualität.

1.2 Aufgabenstellung

Um die genannten Probleme zu umgehen, soll eine Air-Quality-Drone erstellt werden. Hierbei soll eine bereits existierende Drone mit Sensoren ausgestattet werden, welche klassische Werte zur Beurteilung der Luftqualität und zur Beurteilung der Umgebung, wie zum Beispiel die Luftfeuchtigkeit und Temperatur erfassen können. Eine Drone ist agil und kann an verschiedenen Orten und in verschiedenen Luftschichten Messungen durchführen. In dieser Arbeit soll ein Prototyp für eine Drohne zur Messung der Luftqualität erstellt werden. Zu der Drone soll eine App erstellt werden, über die die Drohne bedient werden kann.

2 Theoretische Grundlagen

Für die Erstellung der App, sowie für die Auswahl der Sensoren und Erstellung des Messaufbaus ist verschiedenes Wissen notwendig. Diese theoretischen Grundlagen werden im folgenden erläutert.

2.1 Luftqualität

Die Luftqualität gibt den Gütegrad der Luft an. Dabei handelt es sich um die Bewertung und den Einfluss von Verunreinigung durch Fahrzeugabgase oder Kraftwerke. Bestandteile der ausgestoßenen Luft, wie beispielsweise Stickoxide (**NO_x**) oder Feinstaub, sind für die Luftverschmutzung verantwortlich und werden in Kapitel? näher erläutert. Um die Luftqualität möglichst hoch zu halten und den Menschen wenig zu schaden werden Richtlinien mit Grenzwerten eingeführt sowie fortlaufend verfeinert. Diese Richtlinien geben die Höhe sowie deren Eintrittshäufigkeit pro Zeiteinheit an. Das überschreiten der Grenzwerte hat Sanktionen zur Folge.

2.1.1 Indices

Zur allgemeingültigen Bewertung der Luft gibt es diverse Luftqualitätsindices, die basierend auf den gemessenen Werten von Feinstaub (PM_{2,5} und PM₁₀), bodennahem Ozon (**O₃**), Stickstoffdioxid (**NO₂**) und Schwefeldioxid (**SO₂**), den Gütegrad der lokalen Luft bestimmen. Zur Messung dieser Werte werden derzeit meist stationäre Sensoren, oftmals an viel befahrenen Straßen, eingesetzt.

Es gibt Länder- und Regionenspezifische Indices. Im folgenden werden die Indices für Europa und für die Vereinigten Staaten erläutert.

Air Quality Index (**AQI**)

Der sogenannte Air Quality Index (**AQI**) bildet für die Vereinigten Staaten eine Skala von 0 bis 500 ab. Mit steigendem Wert wird die Luftqualität, bezogen auf den Tag, schlechter

und die Risiken für den Menschen schwerwiegender. Die folgende Tabelle stellt die Zusammenhänge von AQI-Wert und den Folgen sowie deren Bedeutung dar. Laut Environmental Protection Agency (EPA) ist ein Wert von maximal 100 als gesundheitlicher Standard angesetzt.

Air Quality Index Levels of Health Concern	Numerical Value	
Good	0 to 50	
Moderate	51 to 100	Air quality is acceptable; however
Unhealthy for Sensitive Groups	101 to 150	
Unhealthy	151 to 200	Every
Very Unhealthy	201 to 300	
Hazardous	301 to 500	

Tabelle 2.1: AQI Übersicht

Common Air Quality Index (CAQI)

Europäische Länder haben eine eigene, angepasste Skala zur Bewertung der Luftqualität: den Common Air Quality Index (CAQI). Man unterscheidet hierbei kurzfristige (stündlich oder täglich) von langfristigen (jährlich) Luftqualitätsindices. Die stündlich und täglich aktualisierte Luftverschmutzung wird als relatives Maß aus den für Europa bedeutendsten Schadstoffen. Dazu gehören der Feinstaub (PM_{2,5} und PM₁₀), NO₂ und O₃. Bei entsprechend verfügbaren Daten können ebenfalls Kohlenstoffmonoxid (CO) sowie SO₂ einbezogen werden. Zur Berechnung der Index-Klasse unterscheidet man zwischen dem Verkehrs- sowie dem Hintergrundindex. Ersteres soll die Verkehrsbelastung anhand Messsensoren in direkter Nähe von vielbefahrenen Straßen und letzteres die allgemeine Belastung einer Stadt darstellen.

Über den für ein ganzes Jahr hinweg berechneten Index lässt sich der Abstand zu den EU-Grenzwerten darstellen. Der Schwellwert ist dabei 1. Ein höherer Wert des Luftqualitätsindex zeugt von einer Überschreitung eines oder mehrerer Schadstoffwerte. Die Vergleichswerte entsprechen einer Empfehlung durch die Welt Gesundheitsorganisation und dienen dem Schutz der Gesundheit.

Die aktuellen Luftqualitätswerte der Messstationen Europas werden von der EUA und der Europäischen Kommission online unter

2.1.2 Schadstoffe

Im Folgenden soll auf messbare und für die Luftqualität bzw. die menschliche Gesundheit entscheidende Schadstoffe eingegangen werden. Neben der Erläuterung der luftverunreinigenden Teilchen wird auch auf deren Konsequenzen für den Menschen eingegangen.

Feinstaub

Ein für die Gesundheit des Menschen entscheidender Schadstoff ist der Feinstaub oder auch Schwebstaub genannt. Diese kleinen, für das menschliche Auge nicht sichtbaren Teilchen, die nur langsam zu Boden sinken, sind zum größten Teil menschlicher Herkunft. Dazu gehören die Emissionen, die durch Kraftfahrzeuge, Öfen und Heizungen in Innenräumen sowie durch die Metallerzeugung, auftreten. Dabei wird der durch Kraftfahrzeuge entstehende Feinstaub nicht nur aus dem Motor emittiert, sondern auch Bremsen- und Reifenabrieb sowie die Aufwirbeln des Straßenstaubes verunreinigen die Luft. Die eben genannten Faktoren tragen zum primären Feinstaub bei. Der sekundäre Feinstaub hat seine Herkunft in der Landwirtschaft. Hierbei entstehen in der Tierhaltung gasförmige Schadstoffe durch die Ammoniakemissionen. Dabei unterteilt man die Partikel nach ihrer Größe. Alle Teilchen mit einem aerodynamischen Durchmesser kleiner 10 Mikrometer werden als PM₁₀, wobei man diejenigen mit einem Durchmesser von kleiner 2,5 Mikrometer als PM_{2,5} bezeichnet. Eine weitere Aufschlüsselung innerhalb der eben genannten Grenzen ergibt die Begrifflichkeiten Grobfraction, für alle Partikel zwischen 2,5 und 10 Mikrometer, sowie die Feinfraction, für Partikel mit einem Durchmesser kleiner 2,5 Mikrometer. Die aller kleinsten Partikel, aerodynamischer Durchmesser kleiner als 0,1 Mikrometer nennt man ultrafeine Partikel. PM₁₀ wandern in die Nasenhöhlen, PM_{2,5} gelangen über die Atemwege in die Bronchien sowie Lungenbläschen und setzen sich dort ab. Die ultrafeinen Partikel können bis in die Blutgefäße eindringen können. Als Folgen können abhängig von der Partikelgröße Atembeschwerden entstehen sowie die Gefahr eines Herzinfarkts oder einer Lungenerkrankung steigen.

NO_x

Stickstoffoxide sind die Verbindung aus unterschiedlich vielen Stickstoff- und Sauerstoffatomen. Die wichtigsten Vertreter dieser Gruppe sind das Stickstoffmonoxid und das Stickstoffdioxid. Diese Oxide entstehen bei unerwünschten Nebenreaktionen während der Verbrennung von Benzin, Öl, Gas oder Kohle. Sie stellt einen sehr reaktionsfreudigen Stoff dar, wodurch es nicht nur zur Ozonbildung, sondern auch zur Feinstaubbelastung beiträgt. Stickstoffdioxide können vor allem bei Asthmatikern zur Bronchienverengung führen.

SO₂

Das farblose aber stark riechende Schwefeldioxid wird vor allem bei der Verbrennung von Kohle oder Öl erzeugt und liegt im Normalfall als Gas vor. Für den Menschen hat dieser Schadstoff Schleimhaut- und Augenreizungen sowie Atemwegsprobleme zur Folge. Heutzutage ist die Belastung durch SO₂ jedoch nicht mehr kritisch und die Gesundheitsrisiken akut nicht vorhanden. Zudem entstehen aus Schwefeldioxid Sulfatpartikel in der Atmosphäre, die die PM₁₀ Belastung verstärken.

O₃

Bodennahes Ozon gilt als sekundärer Schadstoff, da es erst durch photochemische Prozesse entsteht und nicht direkt emittiert wird. Es ergibt sich vor allem aus Stickstoffoxiden sowie flüchtigen organischen Verbindungen. Diese beiden sogenannten Vorläuferstoffe werden überwiegend vom Menschen erzeugt. Während Stickstoffoxide von Kraftfahrzeugen emittiert werden, entstehen flüchtige organische Stoffe bei der Verwendung von Lösemitteln, wie zum Beispiel in Farben, Klebstoffen, Reinigungsmitteln oder durch die Verbrennung von Kraftstoff. Zu den gesundheitlichen Folgen gehören eine geringere Lungenfunktion sowie Atemwegsbeschwerden. Diese Wirkungen treten vor allem bei körperlicher Belastung sowie bei besonders anfälligen oder vorgeschädigten Personen auf.

2.2 Hardware

Um die Luftqualität in verschiedenen Luftschichten messen zu können ist diverse Hardware notwendig.

Für die Messung werden unterschiedliche Sensoren benötigt, die die wichtigsten Aspekte der Luftqualität, wie zum Beispiel den Feinstaub, messen.

Um die Daten zu verarbeiten und auszulesen ist ein μ -Controller notwendig. In dieser Arbeit wurde ein XDK der Firma Bosch verwendet.

Um agil messen zu können wird eine Drone der Marke DJI eingesetzt. Die Modellbezeichnung lautet Phantom 3 Standard.

2.2.1 DJI Phantom 3 Standard

Der Aufbau der Drone ist in folgender Abbildung zu sehen. Die Drone hat vier Propeller, welche die Antriebskraft leisten. Um Bilder und Videos aufzunehmen ist auf der Unterseite der Drone ein Gimbal mit einer Kamera befestigt. Für eine sichere Landung und einen

guten Stand hat die Drone zwei Beine.



Abbildung 2.1: DJI Phantom 3 Standard

Die Drone lässt sich auf verschiedenen Wegen steuern. Es gibt die Möglichkeit der klassischen manuellen Steuerung über die Fernbedienung oder man lässt die Drone autonom fliegen.

Allgemein ist eine App notwendig, um die Drohne zu steuern und alle Features, welche mit der Drohne kommen, zu nutzen. Hierfür stellt die Firma DJI eine eigene App, die DJI GO App zur Verfügung. Es wird aber auch ein Software Development Kit ([SDK](#)) bereitgestellt, mit dem sich eigene Apps entwickeln lassen. Durch die durch das Software Development Kit bereitgestellten Funktionen lassen sich die Features der DJI GO App replizieren und erweitern.

Das manuelle fliegen lässt sich einfach über die mitgelieferte Fernbedienung realisieren. Die zwei Steuerknüppel dienen hierbei zur Steuerung. Die Funktion der einzelnen Steuerknüppel lässt sich über die DJI eigene App konfigurieren. Hier kann der Nutzer seine Vorlieben einstellen.

Um die App bequem während des Fluges bedienen zu können ist an der Fernbedienung eine Halterung montiert in die man das Endgerät, auf der die App ausgeführt wird, befestigen kann.

Der Aufbau der Fernbedienung ist in folgender Abbildung zu sehen.



Abbildung 2.2: DJI Phantom 3 Standard - Fernbedienung

Die zweite Möglichkeit ist, die Drohne autonom fliegen zu lassen. Hierbei werden sogenannte Waypoints einer Mission hinzugefügt, welche gestartet wird. Die Waypoints beinhalten die Koordinaten mit Längengrad und Breitengrad, sowie der Flughöhe und weiteren Informationen, wie der Fluggeschwindigkeit oder dem Radius mit welchem der Punkt umflogen werden soll. Um den autonomen Flug zu starten wird die Mission gestartet. Während des Fluges erkennt die Drohne über die Kamera Hindernisse und vermeidet eine Kollision mit diesen.

Hierbei spielt das Global Positioning System ([GPS](#)) der Drohne eine wichtige Rolle. Das intelligente System merkt sich den Startpunkt der Drohne. Je nach Einstellung kehrt die Drohne automatisch zum Startpunkt zurück, sollte der Akkustand eine bestimmte Grenze unterschreiten, die Drohne außer Reichweite der Fernbedienung sein oder die return-to-home-Funktion ausgeführt wird.

Die folgende Tabelle gibt eine Übersicht über die wichtigsten, für diese Arbeit relevanten technischen Daten.

Gewicht	1216g
Diagonale Größe	350mm
max. Steiggeschwindigkeit	$5 \frac{m}{s}$
max. Sinkgeschwindigkeit	$3 \frac{m}{s}$
max. Fluggeschwindigkeit	$16 \frac{m}{s}$
max. Flughöhe über NN	6000m
max. Flugzeit	ca. 25 min
Betriebstemperatur	0° bis 40°C
Positionsbestimmung	GPS
	FCC: 1000m
max. Sendereichweite	CE: 500m
	Flughöhe: 120m

Tabelle 2.2: DJI Phantom 3 Standard - technische Daten

2.2.2 iPad 3

Um die in dieser Arbeit zu erstellende App auszuführen und das Produkt zu testen ist ein mobiles Endgerät notwendig.

Bei dem mobilen Endgerät handelt es sich um ein iPad der 3. Generation von der Firma Apple.

Das iPad hat die Abmessungen:

- Höhe: 241,2 mm
- Breite: 185,7 mm
- Tiefe: 9,4 mm
- Gewicht: 652 g

Die Bedienfläche ist ein 9,7"großer Multi-Touch Display. Das iPad ist **WLAN!** fähig und kann eine Bluetooth Verbindung aufbauen.



Abbildung 2.3: iPad 3

2.3 iOS-Appentwicklung

Bei der Appentwicklung für iOS Geräte bietet sich die Apple eigene Programmiersprache Swift an, welche für die in dieser Arbeit erstellten App auch verwendet wurde.

Die Entscheidung für ein für das Projekt sinnvolles Design-Pattern fiel auf das Model View Controller (MVC) Pattern.

Für die Ansteuerung der DJI-Drone ist das DJI-SDK notwendig, sowie für die Einbindung externer Bibliotheken ist Wissen über Cocoa Pods notwendig.

Im folgendem werden die genannten Grundlagen in einzelnen Unterkapiteln kurz beschrieben.

2.3.1 Model View Controller (MVC)

Das MVC-Pattern besteht, wie der Name sagt, aus drei verschiedenen Teilen. Dem Model, dem Controller und der View. Das Model dient ausschließlich zur Speicherung von Daten. Zum Beispiel werden aktuelle Daten der Anwendung, wie zum Beispiel eine Flugroute in einem Model abgespeichert. Die View ist für die Darstellung der Inhalte und Daten zuständig. Ebenso ist die View dafür zuständig die Eingaben eines Nutzers an den entsprechenden Controller weiterzuleiten. Die View beinhaltet auch die gesamte Graphical User Interface (GUI). Der Controller beinhaltet die Anwendungslogik und ist für die Steuerung

der Anwendung verantwortlich.

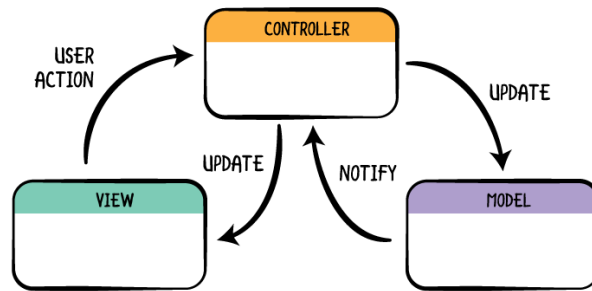


Abbildung 2.4: MVC Design-Pattern Diagramm **mvc1**

Das **MVC** Pattern kann verschieden streng implementiert werden. In dieser Arbeit wird das Pattern in einer leichten Form angewendet. Es wird sich nicht exakt an die Spezifikation aus der Literatur gehalten.

2.3.2 Cocoa

Application Programming Interface (**API**)

Cocoa ist eine Application Programming Interface (**API**) zur Programmierung unter den Betriebssystem (**OS**) MacOS. Für die Apple **OS** von mobilen Endgeräten, welche über Touch-Displays verfügen, wurde die Cocoa **API** zur CocoaTouch **API** erweitert. CocoaTouch beinhaltet Funktionen für die Nutzereingabe über Eingaben durch Gesten. Somit wird für die Entwicklung von iOS Apps, wie in dieser Arbeit, die CocoaTouch **API** verwendet.

Die Entwicklung für Apps mit der Cocoa **API** erfolgt mit den Apple eigenen Developer Tools, Xcode, welches im nächsten Kapitel beschrieben wird, und dem Interface Builder. Die hauptsächlich für die **API** gedachten Programmiersprachen sind Objective-C und die Apple eigene Programmiersprache Swift. Die Programmierung in C und C++ ist generell auch möglich.

Der Aufbau von Cocoa ist im Allgemeinen einfach gehalten. Cocoa besteht aus drei verschiedenen Frameworks.

- *Foundation*: beinhaltet alle relevanten Basisklassen, wie Strings, Arrays, Iterators, et cetera.
- *UIKit*: stellt Klassen zur Entwicklung von Graphical User Interface (**GUI**) zur Verfügung. Zum Beispiel Buttons, Labels, Menüs, usw..
- *Core Data*: dient zur Erstellung von Objektgraphen.

Klassen des Cocoa-Frameworks sind im Quellcode durch die Buchstaben *NS* im Objektnamen zu erkennen.

Pods

CocoaPods ist ein application level dependency manager für Objective-C, Swift und andere Programmiersprachen, welche in Xcode laufen. Pods stellt ein Standardformat zum managen von externen Bibliotheken bereit.

Die Projektabhängigkeiten werden mittels Podfile-Dateien in einem Projekt beschrieben. Im Folgendem ist ein Beispiel zu sehen.

```
1  # platform: iOS, '9.0'
2  use_frameworks!
3  project 'AirQualityDrone.xcodeproj'
4  target 'AirQualityDrone' do
5    pod 'DJI-SDK-iOS' '~> 4.4'
6    pod 'DJI-UILibrary-iOS', '~> 4.4'
7    pod 'CocoaAsyncSocket'
8    pod 'DTMHeatmap'
9  end
```

Listing 2.1: Podfile Beispiel

Um die externen Bibliotheken zu installieren, wird der Befehl *pod install* aufgerufen. Dadurch werden die Quellen der Bibliotheken geladen und das Projekt in Xcode eingerichtet, sodass die Bibliotheken separat gebaut werden. In das Projekt werden die Dateien über eine statische Bibliothek (*libPods.a*).

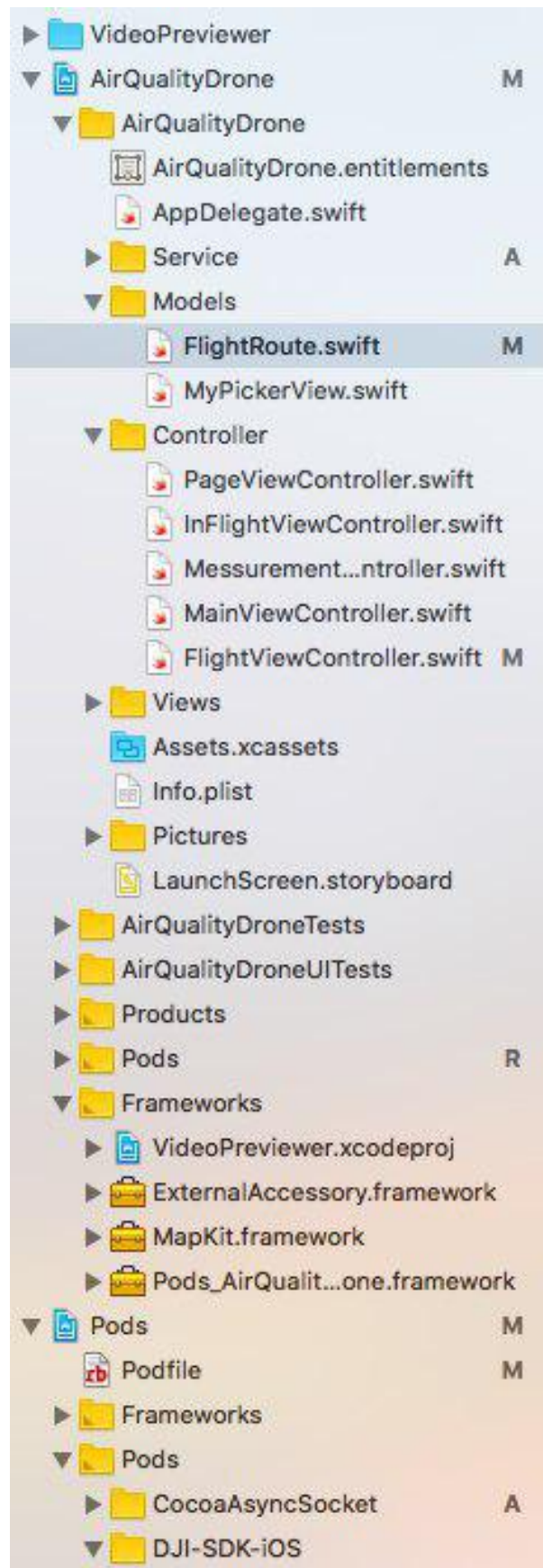


Abbildung 2.5: Externe Bibliotheken in Xcode

2.3.3 Xcode

Xcode ist eine Integrated Development Environment von Apple für das OS macOS. Xcode ist für die Entwicklung von Programmen und Apps für macOS, iOS, tvOS und watchOS gedacht. Die IDE ist Bestandteil der Xcode Tools.

Die Xcode Tools beinhalten:

- Xcode IDE
- Interface Builder
- Instruments
- Xcode Core
- Dashcode
- Quanz Composer
- iPhone Simulator

Die iOS App, welche Bestandteil dieser Arbeit ist, wird ausschließlich mit Xcode entwickelt. Die folgende Abbildung zeigt eine Übersicht der Xcode-Oberfläche.

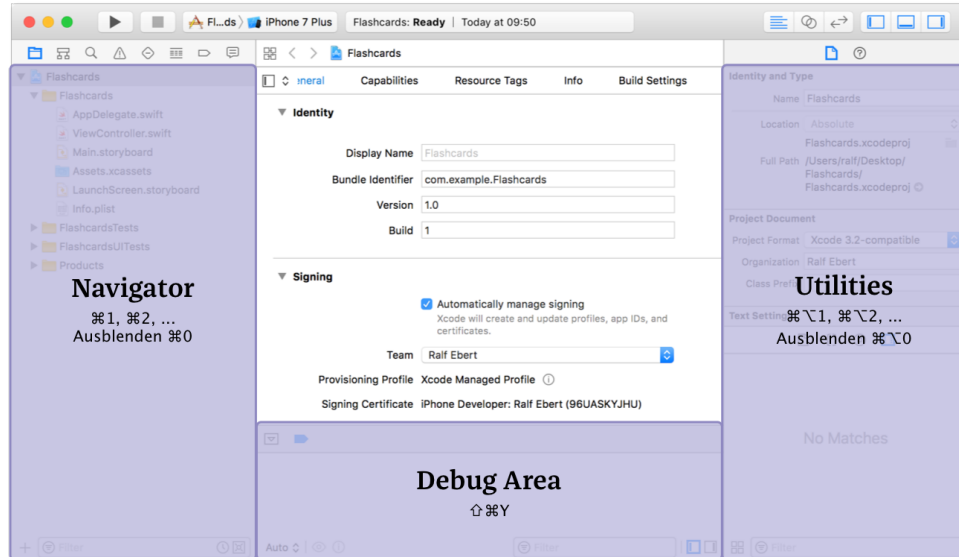


Abbildung 2.6: Xcode-Oberfläche

Zentral ist in Xcode der Editor zu finden. Auf der linken Seite befindet sich der Navigator. Unterhalb des Editors befindet sich die Debug Area und auf der rechten Seite die Utilities. Navigator, Debug Area und Utilities lassen sich über die Buttons in der oberen rechten

Ecke ausblenden und ermöglichen es so das Editor Fenster größer zu machen. Die Projektstruktur ist in Abbildung 2.5 zu sehen.

2.3.4 SWIFT

Swift ist eine Programmiersprache des Technik Konzerns Apple. Sie wurde für die Entwicklung von Apps für iOS, Mac, Apple TV und Apple Watch kreiert. Swift ist kostenlos und Open Source. Die Sprache steht unter der Apache 2.0 Open Source Lizenz. Somit kann eine große Community direkt zum Swift Quellcode beitragen.

Swift vereint unterschiedliche Konzepte verschiedener Programmiersprachen, wie zum Beispiel Objective-C und Python. Dies führt dazu, dass Sie sich durch Paradigmen wie objektorientiert und imperativ beschreiben lässt. Swift greift Mechanismen, wie Klassen, Vererbung, Closures, Typinferenz, generische Typen, etc., auf, welche von anderen Programmiersprachen bereits bekannt sind.

Für die Appentwicklung wurde die Programmiersprache Swift unter der Version 4.1 verwendet.

2.3.5 DJI-Software Development Kit (SDK)

DJI bietet neben dem Verkauf von Drohnen auch noch andere Leistungen an. Hierzu gehören diverse [SDK](#), die die Entwicklung von Apps und Programmen für die Drohnen von DJI erleichtern. [SDK](#) die angeboten werden sind:

- Mobile SDK
- Onboard SDK
- Guidance SDK
- Payload SDK

In dieser Arbeit ist nur das Mobile SDK relevant.

Das DJI Mobile SDK unterstützt die Plattformen iOS 9.0 oder höher, sowie Android 5.0.0 oder höher. Da das mobile Endgerät mit einem iOS [OS](#) läuft, wird in dieser Arbeit die Ausführung des DJI Mobile SDK für iOS verwendet. Hierbei ist die Programmierung in den Sprachen Swift und Objective-C möglich. Wie im Kapitel 2.3.4 erwähnt wird in dieser Arbeit die Programmiersprache Swift 4.1 verwendet.

Das SDK bietet verschiedene Kernfunktionalitäten an. Zu diesen wichtigen und nützlichen Features gehört die Obstacle avoidance, High and low level flight control, Aircraft state through telemetry and sensor data, Live video feed, Pre defined missions, wie Waypoint,

HotPoint oder FollowMe und State information und control of Battery und Remote Controller.

Für die Entwicklung von [GUI](#) stellt DJI eine UXLibrary zur Verfügung, welche graphische Elemente für die wichtigsten Funktionen des mobile [SDK](#) bereitstellt.

Literatur

Anhang

A. Screenshot NameNode Web-Interface

B. DVD Inhalt

C. DVD

A. Screenshot NameNode Web-Interface

Hadoop	Overview	Datanodes	Datanode Volume Failures	Snapshot	Startup Progress	Utilities ▾
--------	----------	-----------	--------------------------	----------	------------------	-------------

Overview 'localhost:9000' (active)

Started:	Fri Jul 10 00:23:31 CEST 2015
Version:	2.7.0, rd4c8d4d4d203c934e8074b31289a28724c0842cf
Compiled:	2015-04-10T18:40Z by jenkins from (detached from d4c8d4d)
Cluster ID:	CID-322169a1-9f18-4284-9cfa-490bd79c1dd4
Block Pool ID:	BP-1249407956-127.0.1.1-1436480592942

Summary

Security is off.

Safemode is off.

1 files and directories, 0 blocks = 1 total filesystem object(s).

Heap Memory used 26.65 MB of 50.49 MB Heap Memory. Max Heap Memory is 966.69 MB.

Non Heap Memory used 30.99 MB of 32.25 MB Committed Non Heap Memory. Max Non Heap Memory is 214 MB.

Configured Capacity:	18.58 GB
DFS Used:	24 KB (0%)
Non DFS Used:	2.85 GB
DFS Remaining:	15.73 GB (84.67%)
Block Pool Used:	24 KB (0%)
DataNodes usages% (Min/Median/Max/stdDev):	0.00% / 0.00% / 0.00% / 0.00%
Live Nodes	1 (Decommissioned: 0)
Dead Nodes	0 (Decommissioned: 0)
Decommissioning Nodes	0
Total Datanode Volume Failures	0 (0 B)
Number of Under-Replicated Blocks	0
Number of Blocks Pending Deletion	0
Block Deletion Start Time	10.7.2015, 00:23:31

NameNode Journal Status

Current transaction ID: 1

Journal Manager	State
FileJournalManager(root=/tmp/hadoop-root/dfs/name)	EditLogFileStream(/tmp/hadoop-root/dfs/name/current/edits_inprogress_0000000000000000001)

NameNode Storage

Storage Directory	Type	State
/tmp/hadoop-root/dfs/name	IMAGE_AND_EDITS	Active

Hadoop, 2014.

C. DVD Inhalt

└ Anwendung/	
– pom.xml	⇒ <i>Maven POM Datei</i>
└ conf/	⇒ <i>*.properties Dateien für Konfiguration</i>
└ src/	⇒ <i>Quellcode Dateien</i>
└ target/	
– Logfileanalyzer-1.0-SNAPSHOT.jar	⇒ <i>Ausführbare JAR-Datei</i>
└ site/apidocs/	⇒ <i>JavaDoc für Browser</i>
└ Literatur/	⇒ <i>PDF Literatur & E-Books</i>
└ Praesentationen/	
– Abschlusspraesentation.pptx	⇒ <i>Präsentation vom 21. August 2015</i>
– Abschlusspraesentation.pdf	
– Kickoffpraesentation.pptx	⇒ <i>Präsentation vom 03. Juni 2015</i>
– Kickoffpraesentation.pdf	
└ Sonstiges/	
– LineareRegression.xlsx	⇒ <i>Berechnung der linearen Regression</i>
└ Latex-Files/	⇒ <i>Editierbare L^AT_EX Dateien der Arbeit</i>
– bibliographie.bib	⇒ <i>Literaturverzeichnis</i>
– dokumentation.pdf	⇒ <i>Bachelorarbeit als PDF</i>
– dokumentation.tex	⇒ <i>Hauptdokument</i>
– einstellungen.tex	⇒ <i>Einstellungen</i>
└ ads/	⇒ <i>Header, Glosar, Abkürzungen, etc.</i>
└ content/	⇒ <i>Kapitel</i>
└ images/	⇒ <i>Bilder</i>
└ lang/	⇒ <i>Sprachdateien für L^AT_EX Template</i>