**DIPLOMARBEIT**

Gesamtprojekt

**FPV-Drohne**

**Elektronik FPV-Drohne**

Marcel Bieder 5AHEL

**Softwareentwicklung FPV-Drohne**

Maximilian Lendl 5AHEL

**CAD-Entwicklung & Datenübertragung**

Ben Heinicke 5AHEL

**Entwicklung einer APP für Smartphone & Videoübertragung**

Sebastian Hinterberger 5AHEL

Betreuer: Dipl.-Ing. Josef Reisinger

Schuljahr 2023/24

Abgabevermerk:

Datum: 02.04.2024 übernommen von:

****

**Höhere Technische Bundeslehranstalt Hollabrunn**

**Höhere Lehranstalt für Elektronik und Technische Informatik**

**EIDESSTATTLICHE ERKLÄRUNG**

**Ich erkläre an Eides statt, dass ich die vorliegende Diplomarbeit selbständig und ohne fremde Hilfe verfasst, andere als die angegebenen Quellen und Hilfsmittel nicht benutzt und die den benutzten Quellen wörtlich und inhaltlich entnommenen Stellen als solche erkenntlich gemacht habe.**

**Marcel Bieder**

**Maximilian Lendl**

**Ben Heinicke**

**Sebastian Hinterberger**

Hollabrunn, am 02.04.2024

**HINWEISE**

Die vorliegende Diplomarbeit wurde in Zusammenarbeit mit der Firma **Dronetech Austria** ausgeführt.

oder

Die vorliegende Diplomarbeit wurde für die Abteilung Elektronik und Technische Informatik der HTL Hollabrunn ausgeführt.

Die in dieser Diplomarbeit entwickelten Prototypen und Software-Produkte dürfen ganz oder auch in Teilen von Privatpersonen oder Firmen nur dann in Verkehr gebracht werden, wenn sie diese selbst geprüft und für den vorgesehenen Verwendungszweck für geeignet befunden haben.

Es wird keinerlei Haftung übernommen für irgendwelche Schäden, die aus der Nutzung der hier entwickelten oder beschriebenen Bestandteile des Projekts resultieren.

Für alle Entwicklungen gilt die GNU General Public License [http://www.gnu.org/licenses/gpl.html] der Free Software Foundation, Boston, USA in der Version 3.

Die Diplomarbeit erfüllt die “Standards für Ingenieur- und Technikerprojekte” entsprechend dem Rundschreiben Nr. 60 aus 1999 des BMBWK (GZ.17.600/101-II/2b/99).

[https://www.bmb.gv.at/ministerium/rs/1999\_60.html]

SCHLÜSSELBEGRIFFE

DANKSAGUNGEN

Reisinger, Wihsböck, Kauer, Stoll, Dronetech Austria (Daniel Stoiber),

**DIPLOMARBEIT**

**DOKUMENTATION**

|  |  |
| --- | --- |
| Namen der  Verfasser/innen |  |
| Jahrgang  Schuljahr |  |
| Thema der Diplomarbeit |  |
| Kooperationspartner |  |

|  |  |
| --- | --- |
| Aufgabenstellung |  |

|  |  |
| --- | --- |
| Realisierung |  |

|  |  |
| --- | --- |
| Ergebnisse |  |

|  |  |
| --- | --- |
| Typische Grafik, Foto etc.  (mit Erläuterung) |  |

|  |  |
| --- | --- |
| Teilnahme an Wettbewerben,  Auszeichnungen |  |

|  |  |
| --- | --- |
| Möglichkeiten der Einsichtnahme in die Arbeit | HTL Hollabrunn  Anton Ehrenfriedstraße 10  2020 Hollabrunn |

|  |  |  |
| --- | --- | --- |
| Approbation  (Datum / Unterschrift) | Prüfer/Prüferin | Direktor/Direktorin  Abteilungsvorstand/Abteilungsvorständin |

**DIPLOMA THESIS**

**Documentation**

|  |  |
| --- | --- |
| Author(s) |  |
| Form  Academic year |  |
| Topic |  |
| Co-operation partners |  |

|  |  |
| --- | --- |
| Assignment of tasks |  |

|  |  |
| --- | --- |
| Realisation |  |

|  |  |
| --- | --- |
| Results |  |

|  |  |
| --- | --- |
| Illustrative graph, photo  (incl. explanation) |  |

|  |  |
| --- | --- |
| Participation in competitions  Awards |  |

|  |  |
| --- | --- |
| Accessibility of  final project thesis | HTL Hollabrunn  Anton Ehrenfriedstraße 10  2020 Hollabrunn |

|  |  |  |
| --- | --- | --- |
| Approval  (Date / Signature) | Examiner/s | Head of Department / College |

DA Antrag und unterschriebene Erklärung aus der Diplomarbeitsdatenbank einfügen

Inhaltsverzeichnis

[1 Einleitung 15](#_Toc158127527)

[1.1 Projektziel 15](#_Toc158127528)

[1.2 Gesamtüberblick 15](#_Toc158127529)

[2 Grundlagen Drohnenflug 16](#_Toc158127530)

[2.1 Inertial Measurement Unit (IMU) 16](#_Toc158127531)

[2.1.1 Gyroskop 16](#_Toc158127532)

[2.1.2 Accelerometer 16](#_Toc158127533)

[2.1.3 Magnetometer 16](#_Toc158127534)

[2.1.4 Barometer 16](#_Toc158127535)

[2.2 Lagewinkel 16](#_Toc158127536)

[2.2.1 Throttle 16](#_Toc158127537)

[2.2.2 Pitch 16](#_Toc158127538)

[2.2.3 Roll 16](#_Toc158127539)

[2.2.4 Yaw 16](#_Toc158127540)

[2.2.5 Komplementärfilter 17](#_Toc158127541)

[2.3 PID-Regler 18](#_Toc158127542)

[2.3.1 P-Glied 19](#_Toc158127543)

[2.3.2 I-Glied 19](#_Toc158127544)

[2.3.3 D-Glied 19](#_Toc158127545)

[2.3.4 PID-Glied 19](#_Toc158127546)

[2.3.5 Einstellen der Regelparameter 19](#_Toc158127547)

[3 Mechanischer Aufbau 20](#_Toc158127548)

[3.1.1 FPV – Drohne Gesamtaufbau 20](#_Toc158127549)

[3.1.2 3D – Modelle 20](#_Toc158127550)

[3.1.2.1 Propellerschutz 20](#_Toc158127551)

[3.1.2.2 Groundstation 20](#_Toc158127552)

[4 Elektronik FPV-Drohne (BIE) 21](#_Toc158127553)

[4.1 Allgemeines 21](#_Toc158127554)

[4.1.1 Grundlegendes Hardwarekonzept 21](#_Toc158127555)

[4.1.2 Anforderungen 21](#_Toc158127556)

[4.1.3 Komponentenauswahl 21](#_Toc158127557)

[4.2 Flight Controller 21](#_Toc158127558)

[4.2.1 Allgemeines 21](#_Toc158127559)

[4.2.2 Spannungsversorgungskonzept 21](#_Toc158127560)

[4.2.3 Mikrocontroller 21](#_Toc158127561)

[4.2.4 Altium PCB Design 21](#_Toc158127562)

[4.2.5 Pinbelegung 21](#_Toc158127563)

[4.3 Electronic Speed Controller (ESC) 21](#_Toc158127564)

[4.3.1 Übersicht 21](#_Toc158127565)

[4.3.2 Aufbau 21](#_Toc158127566)

[4.3.3 Funktionsweise 21](#_Toc158127567)

[4.3.4 Technische Daten 21](#_Toc158127568)

[4.4 Motoren 21](#_Toc158127569)

[4.4.1 Übersicht 21](#_Toc158127570)

[4.4.2 Aufbau 21](#_Toc158127571)

[4.4.3 Funktionsweise 21](#_Toc158127572)

[4.4.4 Technische Daten 21](#_Toc158127573)

[4.5 Receiver Fernsteuerung 21](#_Toc158127574)

[4.5.1 Übersicht 21](#_Toc158127575)

[4.5.2 Funktionsweise 21](#_Toc158127576)

[4.5.3 Technische Daten 21](#_Toc158127577)

[4.6 Video Transmitter (VTx) 21](#_Toc158127578)

[4.6.1 Übersicht 21](#_Toc158127579)

[4.6.2 Funktionsweise 21](#_Toc158127580)

[4.6.3 Technische Daten 21](#_Toc158127581)

[4.7 Live Kamera 21](#_Toc158127582)

[4.7.1 Übersicht 22](#_Toc158127583)

[4.7.2 Funktionsweise 22](#_Toc158127584)

[4.7.3 Technische Daten 22](#_Toc158127585)

[4.8 VR-Brille 22](#_Toc158127586)

[4.8.1 Übersicht 22](#_Toc158127587)

[4.8.2 Funktionsweise 22](#_Toc158127588)

[5 Steuerungssoftware 23](#_Toc158127589)

[5.1 Einführung HAL-Library und STM32CubeMX 23](#_Toc158127590)

[5.2 Softwarearchitektur 23](#_Toc158127591)

[5.3 Fernsteuerung 23](#_Toc158127592)

[5.3.1 Konfiguration 23](#_Toc158127593)

[5.3.2 Unterstützte Protokolle 25](#_Toc158127594)

[5.3.2.1 PPM 25](#_Toc158127595)

[5.3.2.2 S.Bus 25](#_Toc158127596)

[5.3.2.3 I.Bus 26](#_Toc158127597)

[5.3.3 Empfängersoftware 26](#_Toc158127598)

[5.4 Inertial Measurement Unit (IMU) 26](#_Toc158127599)

[5.4.1 I²C Protokoll 26](#_Toc158127600)

[5.4.2 Gyroskop und Accelerometer - MPU9250 27](#_Toc158127601)

[5.4.2.1 Initialisierung 27](#_Toc158127602)

[5.4.2.2 Einlesen der Daten 27](#_Toc158127603)

[5.4.2.3 Komplementärfilter 27](#_Toc158127604)

[5.4.3 Luftdrucksensor - BMP280 27](#_Toc158127605)

[5.4.3.1 Initialisierung 27](#_Toc158127606)

[5.4.3.2 Einlesen der Daten 27](#_Toc158127607)

[5.4.3.3 Berechnung der Höhe 27](#_Toc158127608)

[5.4.4 Magnetometer - AK8963 27](#_Toc158127609)

[5.4.4.1 Initialisierung 27](#_Toc158127610)

[5.4.4.2 Einlesen der Daten 27](#_Toc158127611)

[5.5 Smart Battery Monitor - DS2438 27](#_Toc158127612)

[5.5.1 One-Wire Protokoll 27](#_Toc158127613)

[5.5.2 Spannungsmonitoring 27](#_Toc158127614)

[5.6 Abstandssensor - HCSR04 (Ben) 28](#_Toc158127615)

[5.6.1 Funktionsweise 28](#_Toc158127616)

[5.6.2 Code 28](#_Toc158127617)

[5.7 Motoransteuerung 28](#_Toc158127618)

[5.7.1 Motorstrom auslesen 28](#_Toc158127619)

[5.7.2 DShot Protokoll 28](#_Toc158127620)

[5.8 PID-Regler 28](#_Toc158127621)

[5.8.1 Hover Modus 28](#_Toc158127622)

[5.8.2 Failsafe 28](#_Toc158127623)

[5.9 Debugging 28](#_Toc158127624)

[5.9.1 Terminal Ausgabe 28](#_Toc158127625)

[5.9.2 Status – LEDs 28](#_Toc158127626)

[6 Datenübertragung der Mess- und Videodaten (Ben) 29](#_Toc158127627)

[6.1 Überblick Datenübertragung 29](#_Toc158127628)

[6.2 Kommunikation: Flight Controller und Sender 29](#_Toc158127629)

[*6.2.1 Aufbau Sendermodul (VTx)* 29](#_Toc158127630)

[6.2.2 ASK – Modulation 29](#_Toc158127631)

[6.2.3 Transmit – Programm auf Cortex µC 29](#_Toc158127632)

[6.2.3.1 Gleitkommadarstellung (memcpy) 29](#_Toc158127633)

[6.2.3.2 Code 29](#_Toc158127634)

[*6.2.4* Testen der Übertragung 29](#_Toc158127635)

[6.3 Kommunikation: Sender und Empfänger 29](#_Toc158127636)

[6.3.1 Aufbau Empfängermodul 29](#_Toc158127637)

[6.3.2 Verbindungsaufbau 29](#_Toc158127638)

[6.3.3 Testen der Übertragung 29](#_Toc158127639)

[6.4 Kommunikation: Empfänger und Raspberry Pi 29](#_Toc158127640)

[6.4.1 Empfangsprogramm auf Raspberry Pi 29](#_Toc158127641)

[6.4.1.1 UART auf dem Raspberry Pi 29](#_Toc158127642)

[6.4.1.2 Code 29](#_Toc158127643)

[6.4.1 Senden der Daten auf dem MQTT-Server 29](#_Toc158127644)

[6.5 Testen der Datenübertragungskette 29](#_Toc158127645)

[7 Visualisierungs-App 30](#_Toc158127646)

[7.1 Applikation 30](#_Toc158127647)

[7.1.1 Einführung – Dart / Flutter - Framework 30](#_Toc158127648)

[7.1.1.1 Allgemeines 30](#_Toc158127649)

[7.1.1.2 Pub Dev / Package Installer 30](#_Toc158127650)

[7.1.1.3 Dart Syntax 30](#_Toc158127651)

[7.1.1.4 State Management 30](#_Toc158127652)

[7.1.1.4.1 Stateful vs. Stateless Widgets 30](#_Toc158127653)

[7.1.1.4.2 Provider 30](#_Toc158127654)

[7.1.2 Allgemeines zur App 30](#_Toc158127655)

[7.1.2.1 Usersystem 30](#_Toc158127656)

[7.1.2.2 Datenvisualiserung 30](#_Toc158127657)

[7.1.2.3 3D-Model-Viewer 30](#_Toc158127658)

[7.1.2.4 Livestream-Viewer 30](#_Toc158127659)

[7.1.3 UI-Konzept 30](#_Toc158127660)

[7.1.4 Projektstruktur und -umgebung 30](#_Toc158127661)

[7.1.4.1 Editor – Visual Studio Code 30](#_Toc158127662)

[7.1.4.2 Flutter Installation 30](#_Toc158127663)

[7.1.4.3 Projekterstellung 30](#_Toc158127664)

[7.1.4.4 Projektstruktur 30](#_Toc158127665)

[7.1.4.5 Packages 30](#_Toc158127666)

[7.1.4.5.1 Pubspec.yaml 30](#_Toc158127667)

[7.1.4.5.2 Installieren neuer Packages 30](#_Toc158127668)

[7.1.5 Splash + Willkommensscreen 31](#_Toc158127669)

[7.1.5.1 Native Splash Screen 31](#_Toc158127670)

[7.1.5.2 Willkommensbildschirm 31](#_Toc158127671)

[7.1.5.3 Logo-Design 31](#_Toc158127672)

[7.1.6 Login + Registrierung 31](#_Toc158127673)

[7.1.6.1 E-Mail-Login 31](#_Toc158127674)

[7.1.6.2 Google-Login 31](#_Toc158127675)

[7.1.6.3 Passwort vergessen 31](#_Toc158127676)

[7.1.6.4 Registrierung 31](#_Toc158127677)

[7.1.7 Homepage 31](#_Toc158127678)

[7.1.7.1 Bottom Navigation Bar / GNav-Bar 31](#_Toc158127679)

[7.1.7.2 Serverdatendialog 31](#_Toc158127680)

[7.1.7.3 Flugdatenvisualisierung 31](#_Toc158127681)

[7.1.7.3.1 MQTT-Datenstreams 31](#_Toc158127682)

[7.1.7.3.2 Syncfusion Cartesian Charts 31](#_Toc158127683)

[7.1.7.3.3 Echtzeitdaten + Animierte Diagramme 31](#_Toc158127684)

[7.1.7.3.4 Speicherung der Daten 31](#_Toc158127685)

[7.1.7.4 3D-Model-Viewer 31](#_Toc158127686)

[7.1.7.4.1 Flutter Cube 31](#_Toc158127687)

[7.1.7.4.2 MQTT-Datenstream 31](#_Toc158127688)

[7.1.7.4.3 3D-Dronenmodell 31](#_Toc158127689)

[7.1.7.4.3.1 Optimierungen in Blender 31](#_Toc158127690)

[7.1.7.4.4 Extrabedienungen 31](#_Toc158127691)

[Live-View 31](#_Toc158127692)

[7.1.7.4.5 VLC-Plugin 31](#_Toc158127693)

[7.1.7.4.6 Darstellung eines Netzwerkstreams 31](#_Toc158127694)

[7.1.7.4.7 Video-Overlay 31](#_Toc158127695)

[7.1.8 Sidemenu / Drawer 31](#_Toc158127696)

[7.1.8.1 Userprofil 31](#_Toc158127697)

[7.1.8.1.1 Darstellung der Userdaten 31](#_Toc158127698)

[7.1.8.1.2 Profilbildauswahl 31](#_Toc158127699)

[7.1.8.1.3 Änderung der Userdaten 32](#_Toc158127700)

[7.1.8.2 Credits 32](#_Toc158127701)

[7.1.8.3 Vorherige Flüge 32](#_Toc158127702)

[7.1.8.3.1 Darstellung aller gespeicherten Flüge 32](#_Toc158127703)

[7.1.8.3.2 Sortiermöglichkeiten 32](#_Toc158127704)

[7.1.8.3.3 Einsicht in vorherigen Flug 32](#_Toc158127705)

[7.1.8.4 Einstellungen 32](#_Toc158127706)

[7.1.8.4.1 Dark- / Light-Mode 32](#_Toc158127707)

[7.1.8.4.2 3D-Model Ausrichtung 32](#_Toc158127708)

[7.1.8.5 Logout´ 32](#_Toc158127709)

[7.2 Firebase Backend 32](#_Toc158127710)

[7.2.1 Installation via Firebase CLI 32](#_Toc158127711)

[7.2.1.1.1 Firebase CLI – Setup 32](#_Toc158127712)

[7.2.1.1.2 Firebase für Dart / Flutter aktivieren 32](#_Toc158127713)

[7.2.1.1.3 Flutter App mit Flutterfire konfigurieren 32](#_Toc158127714)

[7.2.2 Einbindung in Flutter 32](#_Toc158127715)

[7.2.2.1.1 Flutterfire Configure 32](#_Toc158127716)

[7.2.2.1.2 Benutzten Firebase Packages hinzufügen 32](#_Toc158127717)

[7.2.3 Nutzen der verschiedenen Datenbanksysteme 32](#_Toc158127718)

[7.2.3.1 Auth 32](#_Toc158127719)

[7.2.3.1.1 E-Mail + Passwort – Login 32](#_Toc158127720)

[7.2.3.1.2 Google Login 32](#_Toc158127721)

[7.2.3.2 Firestore Database 32](#_Toc158127722)

[7.2.3.2.1 Struktur 32](#_Toc158127723)

[7.2.3.2.2 Allgemeine Userdaten 32](#_Toc158127724)

[7.2.3.2.3 Flugdaten 32](#_Toc158127725)

[7.2.3.2.4 Einstellungen 32](#_Toc158127726)

[7.2.3.3 Realtime Database 32](#_Toc158127727)

[7.2.3.3.1 Allgemein 32](#_Toc158127728)

[7.2.3.3.2 Sinn der verschiedenen Flags 33](#_Toc158127729)

[7.2.3.4 Storage 33](#_Toc158127730)

[7.2.3.4.1 Allgemein 33](#_Toc158127731)

[7.2.3.4.2 Referenz zu User in Firestore 33](#_Toc158127732)

[8 Videostreaming 33](#_Toc158127733)

[8.1 Allgemeiner Aufbau 33](#_Toc158127734)

[8.1.1 Anforderungen 33](#_Toc158127735)

[8.1.2 Übertragungskette 33](#_Toc158127736)

[8.2 CADFPX Analog Kamera 33](#_Toc158127737)

[8.2.1 Allgemein 33](#_Toc158127738)

[8.2.2 Produktinformationen 33](#_Toc158127739)

[8.2.3 Verwendungszweck 33](#_Toc158127740)

[8.2.4 Verbindungstest via USB-Camera App 33](#_Toc158127741)

[8.3 USB2.0 VHS Video Grabber 33](#_Toc158127742)

[8.4 RTMP-Server via NGINX aufsetzen 33](#_Toc158127743)

[8.4.1 RTMP Allgemein 33](#_Toc158127744)

[8.4.2 NGINX 33](#_Toc158127745)

[8.4.2.1 Allgemein 33](#_Toc158127746)

[8.4.2.2 Installation 33](#_Toc158127747)

[8.4.2.3 RTMP-Konfiguration 33](#_Toc158127748)

[8.4.2.4 Serverstatus einsehen 33](#_Toc158127749)

[8.5 RTMP-Stream erstellen 33](#_Toc158127750)

[8.5.1 FFMPEG Allgemein 33](#_Toc158127751)

[8.5.2 Aufbau der Konvertierung 33](#_Toc158127752)

[8.5.2.1 Simple Konvertierung 33](#_Toc158127753)

[8.5.2.2 Optimierungen 33](#_Toc158127754)

[8.5.2.2.1 Encodervergleich 33](#_Toc158127755)

[8.5.2.2.2 Optimierungsflags 33](#_Toc158127756)

[8.5.2.2.3 Unoptimiert vs. Optimiert 34](#_Toc158127757)

[9 Videoserver-Port Forwarding 34](#_Toc158127758)

[9.1 Allgemein 34](#_Toc158127759)

[9.2 Routereinstellungen 34](#_Toc158127760)

[9.3 Kontakt mit Routerfirma zur Freischaltung 34](#_Toc158127761)

[9.4 Testen der Ports 34](#_Toc158127762)

[9.4.1 Portchecker.co 34](#_Toc158127763)

[9.5 Testen des Videostreams 34](#_Toc158127764)

[9.5.1 VLC-Player 34](#_Toc158127765)

[9.5.1.1 Devicestream direkt testen 34](#_Toc158127766)

[9.5.1.2 Netzwerkstream im selben Netzwerk 34](#_Toc158127767)

[9.5.1.3 Netzwerkstream via Public IP 34](#_Toc158127768)

[10 Ergebnisse 34](#_Toc158127769)

[11 Anhang 35](#_Toc158127770)

[11.1.1 Einführung CAD – Software (Fusion 360) (Ben) 35](#_Toc158127771)

[11.1.1.1 UI und Projekterstellung 35](#_Toc158127772)

[11.1.1.2 Skizze anfertigen 35](#_Toc158127773)

[11.1.1.3 Körper erstellen 35](#_Toc158127774)

[11.1.1.4 Schrift und Bilder einfügen 35](#_Toc158127775)

[11.1.2 3D – Druck (Ben) 35](#_Toc158127776)

[11.1.2.1 3D – Drucker 35](#_Toc158127777)

[11.1.2.2 Filamente 35](#_Toc158127778)

[11.1.2.3 3D – Drucker Software (Ultimaker Cura) 35](#_Toc158127779)

[11.1.3 Inbetriebnahme Anleitung 35](#_Toc158127780)

[11.1.4 Projektplan 35](#_Toc158127781)

[11.1.5 Projektkosten 35](#_Toc158127782)

[11.1.6 Projekttagebuch 35](#_Toc158127783)

[12 Quellen 36](#_Toc158127784)

[12.1 Gedruckte Medien 36](#_Toc158127785)

[12.2 Online 36](#_Toc158127786)

[13 Verzeichnis der Abbildungen 37](#_Toc158127787)

[14 Begleitprotokoll 38](#_Toc158127788)

[15 Anhang 39](#_Toc158127789)

[15.1 — Projektdokumentation (Kostendarstellung, Besprechungsprotokolle etc.) 39](#_Toc158127790)

[15.2 — Technische Dokumentation (technische Beschreibungen, Berechnungen, 39](#_Toc158127791)

[15.3 Konstruktionszeichnungen, Versuchsberichte, betriebswirtschaftliche Kalkulationen etc.) 39](#_Toc158127792)

[15.4 Schaltungen, Zeichnungssätze, sonstiges 39](#_Toc158127793)

# Einleitung

## Projektziel

## Gesamtüberblick

# Grundlagen Drohnenflug

## Inertial Measurement Unit (IMU)

### Gyroskop

### Accelerometer

### Magnetometer

### Barometer

## Lagewinkel

### Throttle

Throttle beschreibt die Geschwindigkeit aller vier Motoren. Ein anderer Begriff für Throttle ist der Motorschub. Bei der Angabe von zum Beispiel 80% Throttle ist gemeint, dass alle Motoren sich mit 80% Geschwindigkeit drehen. Wenn der Wert hoch ist, fliegt die Drohne nach oben. Bei einem niedrigen Wert fliegt die Drohne nach unten.

Die Motoren und Propellern müssen genug Schub erzeugen, damit das Gesamtgewicht der Drohne angehoben und manövriert werden kann. Die Throttleangabe ist vor allem wichtig bei einem Hover-Modus. Dabei regelt der Quadrokopter die einzelnen Geschwindigkeiten der Motoren, um sich selber aufrecht zu halten.

### Pitch

Bei Pitch handelt es sich um den Neigungswinkel nach vorne und hinten, wenn die Drohne von hinten angeschaut wird. Mit der Angabe vom Pitch kann der Quadrokopter nach vorne und nach hinten gelenkt werden.

Bild Drohne mit Pitch winkel

Vielleicht mehr physikalisch erklären

### Roll

Bei Roll handelt es sich um den Neigungswinkel nach rechts und links, wenn die Drohne von hinten angeschaut wird. Mit der Angabe vom Roll kann der Quadrokopter nach rechts und links gesteuert werden.

Bild Drohne mit Roll winkel

Vielleicht mehr physikalisch erklären

### Yaw

Bei Yaw handelt es sich um die Drehung um die eigene Achse nach links oder rechts. Mit der Angabe vom Yaw kann der Quadrokopter nach links und rechts drehen.

Bild Drohne mit Yaw winkel

Vielleicht mehr physikalisch erklären

### Komplementärfilter

**Accelerometer Winkel**:

Durch die Kombination der terrestrischen Gravitation und Trigonometrie ermöglicht es die Accelerometer-Werte in Neigungswinkel umzuwandeln:

accelerometerPitch/Roll … jeweilig berechneter Winkel in Radiant

accelerometerX/Y/Z … Accelerometer Messwert der jeweiligen Achse in g

Der Yaw-Winkel kann nicht berechnet werden, da die Yaw-Bewegung in der Ebene, um die Z-Achse, stattfindet und daher unabhängig von dem Werten der X- und Y-Achse ist.

Im Gegensatz zum Gyroskop besitzt ein Accelerometer keinen Drift in der Messung und kann daher für längerfristige Messungen verwendet werden.

Für den Filter wird der berechnete Winkel in Grad, statt Radiant, benötigt. Daher müssen die Ergebnisse mit einem Faktor von multipliziert werden.

Problem:

Wenn der Accelerometer bewegt wirkt, kann dieser nicht zwischen den Bewegungsbeschleunigungen und der Erdbeschleunigung unterscheiden und liefert daher verfälschte Werte.

**Gyroskop Winkel**:

Durch die Integration der Winkelbeschleunigungswerte über der Zeit ist es möglich Lagewinkel zu bestimmen:

gyroscopePitch/Roll/Yaw … jeweilig berechneter Winkel in Radiant

gyroscopeX/Y/Z … Gyroskop Messwert der jeweiligen Achse in Grad pro Sekunde

dt … Zeitbereich in Sekunden

Problem:

Durch die Integration werden Messungenauigkeit aufsummiert, die zu einen Wertedrift führen. Daher kann das Gyroskop nur für kurzfristige Messungen verwendet werden.

**Sensorfusion**:

Der Komplementärfilter kann als Kombination von zwei Filter gesehen werden: Ein Hochpassfilter für das Gyroskop und einen Tiefpassfilter für den Accelerometer.

Das Accelerometer liefert eine gute Indikation der Orientierung bei konstanten Bedingungen und das Gyroskop liefert eine gute Indikation bei schnellen Neigungsänderungen.

Winkel … Pitch/Roll-Winkel in Grad

α … Filterkoeffizient

gyroscopeData … Gyroskop Messwerte der jeweiligen Achse in Grad pro Sekunde

dt … Abtastzeit in Sekunden

accelerometerWinkel … berechneter Roll/Pitch nur mit Accelerometerdaten in Grad

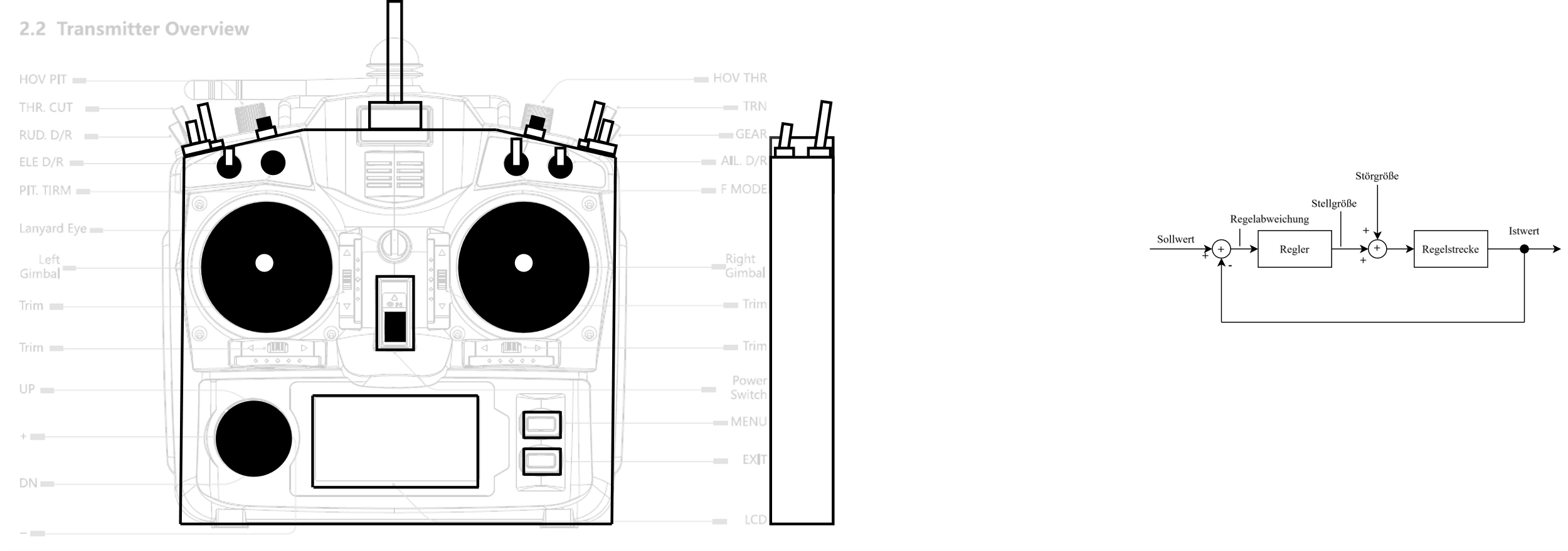
Der Wert für α ist typischerweise 0,98. Das bedeutet, dass die Gyroskop-Messung zu 98% gewichtet wird und die Accelerometer-Messung zu 2%. Daraus folgt:

Dadurch, dass mit dem Accelerometer kein Yaw-Winkel bestimmt werden kann, wird der Winkel nur mit dem Gyroskop berechnet. Dieser Winkel wird über die Zeit immer ungenauer und muss für einen genauen Wert nach einer Zeit zurückgesetzt werden.

## PID-Regler

Im Allgemeinen wird zwischen einer Steuerung und einer Regelung unterschieden. Bei einer Steuerung wird der Sollwert direkt in einen Steuerungsalgorithmus geleitet, der immer nach demselben Schema abläuft. Bei einer Regelung wird der Istwert des Systems wieder zurückgekoppelt und mit dem Sollwert verglichen. Dieser Vergleich ergibt eine Regelabweichung, die das in den Regelalgorithmus geleitet wird. Das Ziel einer Regelung ist, die Regelabweichung zu verschwinden zu bringen.

In einer Drohne wird der PID-Regler verwendet, um die Drohne während des Fluges stabil zu halten. Damit soll die Drohne bei zum Beispiel Windstößen sich wieder selbst richtig ausrichten und den normalen Flug weiterführen.



Sollwert … die gewünschte Einstellung, die von Regler gehalten werden soll

Regelabweichung … Differenz zwischen Soll- und Istwert

Regler …

Stellgröße …

Störgröße …

Regelstrecke …

Istwert …

Es gibt verschiedene Arten für Regler, die für unterschiedliche Anwendungen Vor- und Nachteile liefern. Für den Quadrokopter wurde ein PID-Regler verwendet:

### P-Glied

Blockschaltbild + Formel + Diagramm bei Sprungantwort + Vor/Nachteile

### I-Glied

Blockschaltbild + Formel + Diagramm bei Sprungantwort + Vor/Nachteile

### D-Glied

Blockschaltbild + Formel + Diagramm bei Sprungantwort + Vor/Nachteile

### PID-Glied

Blockschaltbild + Formel + Diagramm bei Sprungantwort + Vor/Nachteile

### Einstellen der Regelparameter

Ablauf der Einstellung der Bestimmung der Parameter

# Mechanischer Aufbau

### FPV – Drohne Gesamtaufbau

### 3D – Modelle

#### Propellerschutz

#### Groundstation

# Elektronik FPV-Drohne (BIE)

## Allgemeines

### Grundlegendes Hardwarekonzept

### Anforderungen

### Komponentenauswahl

## Flight Controller

### Allgemeines

### Spannungsversorgungskonzept

### Mikrocontroller

### Altium PCB Design

### Pinbelegung

## Electronic Speed Controller (ESC)

### Übersicht

### Aufbau

### Funktionsweise

### Technische Daten

## Motoren

### Übersicht

### Aufbau

### Funktionsweise

### Technische Daten

## Receiver Fernsteuerung

### Übersicht

### Funktionsweise

### Technische Daten

## Video Transmitter (VTx)

### Übersicht

### Funktionsweise

### Technische Daten

## Live Kamera

### Übersicht

### Funktionsweise

### Technische Daten

## VR-Brille

### Übersicht

### Funktionsweise

# Steuerungssoftware

## Einführung HAL-Library und STM32CubeMX

Erklärung was ist HAL und STM32CubeMX

Projekterstellung

Setup mit µVision

## Softwarearchitektur

Übersicht Blockschaltbild

+ Erklärung

## Fernsteuerung

Um eine FPV-Drohne steuern zu können, muss mindestens ein 4-Kanal Sender und Empfänger benötigt, um die gewünschten Steuerungsdaten, Throttle, Pitch, Roll und Yaw, zu senden. Mit weiteren Kanälen können zusätzliche Funktionen, wie ON/OFF-Switch, realisiert werden.

Bild Tranmitter und Receiver

Für das Projekt wurde der Turnigy 9X 9Ch Transmitter mit dem TGY-iA6C Receiver ausgewählt, da diese eine große Auswahl an Kanälen, Schalter und digitalen Signalausgang ermöglichen.

Erklärung Layout der Knöpfe + LED an Receiver

Wichtig:

Wenn die Versorgungsspannung der Fernsteuerung weniger als 8,5V beträgt, beginnt ein Buzzer alle 5 Sekunden den Benutzer zu alarmieren. Die aktuelle Spannung kann auf dem Bildschirm überprüft werden.

**Switch Error:**

Beim Starten der Fernsteuerung, kann es vorkommen, dass am Bildschirm „switch-error!“ steht. Dieser Fehler wird angegeben, wenn die Schalter nicht auf ihrer High-Position sind. Das bedeutet beim Starten der Fernsteuerung, müssen alle Schalter nach oben zeigen, beziehungsweise die Schalter, die auf der Oberseite montiert worden sind, müssen nach hinten zeigen.

Bild das das erklärt

### Konfiguration

Damit die ausgewählte Fernsteuerung mit unserem System funktioniert, müssen grundlegende Vorbereitungen vor der Inbetriebnahme getroffen und kontrolliert werden.

**Verbindung mit Empfänger:**

Um die Verbindung zu testen, schalte die Fernsteuerung und den Receiver ein. Die eingebaute Status-LED des Receivers, fangt schnell zum Blinken an. Wenn die LED nach ein paar Sekunden anfangt durchgehend zu leuchten, ist die Fernsteuerung mit dem Receiver verbunden.

Wenn dies nicht der Fall ist, muss die Fernsteuerung mit dem Receiver neugekoppelt werden:

1. Schalte Fernsteuerung und Receiver aus
2. Halte den Bind-Knopf auf der Rückseite der Fernsteuerung, während diese angeschaltet wird
3. Schalte den Receiver ein
4. Warte, bis die Status-LED auf dem Receiver durchgehend leuchtet
5. Lasse den Bind-Knopf los und starte die Fernsteuerung neu, um die Koppelung zu testen

**Einstellungen in Fernsteuerung:**

Damit die Signale richtig von der Flugsoftware interpretiert werden können, müssen folgende Einstellungen in der Fernsteuerung getroffen werden:

Beim längeren gedrückt halten der MENU-Taste wird eine Auswahl zwischen „System Setting“ und „Function Setting“.

|  |  |
| --- | --- |
| **System Setting** | |
| Type Select | ACRO |
| Modeuat | PPM |
| Stick Set | MODEL 1 |
| Output Select | PPM s-BUS |

Die dargestellten Einstellungen sind die empfohlenen Werte. Die Einstellungen Stick Set und Output Select, kann vom Benutzer in der Software festgelegt werden. *siehe Stick Set Änderung* und *Output Select Änderung*

|  |  |  |
| --- | --- | --- |
| **Function Setting** | | |
| AUX-CH  (für Ein-/Ausschalter) | CH5 Gear | |
| PROG. MIX  (für 3-Wege-Schalter) | MIX1 | STATE: ACT  MASTER: GYR  SLAVE: FLP  OFFSET: 000  UPRATE: -100  DNRATE: 100  SW: NOR |
| MIX2 | STATE: ACT  MASTER: GYR  SLAVE: FLP  OFFSET: 000  UPRATE: 000  DNRATE: 000  SW: ID1 |
| MIX3 | STATE: ACT  MASTER: GYR  SLAVE: FLP  OFFSET: 000  UPRATE: 100  DNRATE: -100  SW: ID2 |
| DISPLAY | Kontrolle der Kanal-Ausgänge | |

Die restlichen Einstellungen haben die Standardwerte und müssen nicht verändert werden.

### Unterstützte Protokolle

Der Receiver unterstützt eine parallele digitale und analoge Signalausgabe. Ein Pin gibt immer ein PPM-Signal aus, während der andere Pin entweder ein S.Bus oder I.Bus Signal ausgibt. Die Ausgabe kann in den Fernsteuerungseinstellungen festgelegt werden. *(*[*siehe: 5.3.1 Konfiguration*](#_Konfiguration)*)*

#### PPM

Das PPM-Signal ist das einzige analoge Signale, was der Receiver ausgibt. Die Daten werden mit der Timer-Peripherie eingelesen, indem die Länge des high-Pegels bestimmt werden. Die Daten kommen in 12ms langen Paketen mit einer 8,1ms high-aktiven Ruhezustand. Die Daten befinden sich in den Bereich 0,5ms bis 1,5ms.

Dadurch, dass die digitalen Protokolle um ein Vielfaches schneller ist und nicht sehr fehleranfällig sind, unterstützt die Steuerungssoftware das PPM-Protokoll nicht. Das PPM-Signal wird bei der Initialisierung für einen Verbindungstest mit GPIO-Input verwendet.

#### S.Bus

Das S.Bus-Protokoll ist das empfohlene Protokoll für die Benutzung der Drohne. Es handelt sich um ein digitales Protokoll, was mittels der UART-Peripherie eingelesen werden kann. Das Protokoll ist schneller als das analoge Protokoll PPM und ist weniger fehleranfällig als das I.Bus-Protokoll, da das Signal invertiert ist. Die Datenpackete werden kontinuierlich gesendet, sind ungefähr 3ms lang und besitzen einen ungefähr 4,7ms langen low-aktiven Ruhezustand. Die Daten befinden sich in dem Bereich von 350 bis 1680.

UART-Konfiguration in STM32CubeMX:

Bilder I guess oder Tabelle von Einstellungen

Das S.Bus Protokoll besteht aus 25 Bytes:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Byte[0] | S.Bus header, 0x0F | | | |
| Byte[1…22] | Daten, 16 Kanäle | | | |
| Byte[23] | Bit[0] (0x01)  Kanal 17 | Bit[1] (0x02)  Kanal 18 | Bit[2] (0x04)  Frame Lost Flag | Bit[3] (0x08)  Failsafe Flag |
| Byte[24] | S.Bus footer, 0x00 | | | |

Die Byte[1…22] beinhalten die Daten der einzelnen Kanäle. Jeder Kanal besteht aus 11 Bits, die nacheinander gesendet werden, wobei die ersten Bits des Kanals den niedrigsten Stellwert haben.

Beispiel für eine Dekodierung:

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Byte** | 0 | 1 | 2 | 3 | … | 23 | 24 |
| **Daten**  header  0b001 0011 0001  Kanal 1  0b001 1001 0001  Kanal 2  flags  footer | 0x0F | 0b00110001 | 0b00110001 | 0b00110001 | … | 0b00000000 | 0x00 |

#### I.Bus

Das I.Bus-Protokoll ist das zweite digitale Protokoll, welches der Receiver ausgeben kann. Die Daten können über die UART-Peripherie eingelesen werden. Die Datenpackete werden kontinuierlich gesendet, sind ungefähr 3ms lang und besitzen einen ungefähr 4,7ms langen high-aktiven Ruhezustand. Im Gegensatz zum S.Bus Protokoll kann mit I.Bus keine Signalverlust festgestellt werden, da dieses keine Flags besitzt. Die Daten befinden sich in dem Bereich 1070 bis 1920.

UART-Konfiguration mit STM32CubeMX:

Bilder I guess oder Tabelle von Einstellungen

Das I.Bus Protokoll besteht aus 32 Bytes:

|  |  |
| --- | --- |
| Byte[0] | Protokolllänge, 0x20 |
| Byte[1] | Command Code, 0x40 |
| Byte[2…29] | Daten, 14 Kanäle |
| Byte[30…31] | Checksumme |

Die Byte[2…29] beinhalten die Daten der einzelnen Kanäle. Jeder Kanal besteht aus 2 Bytes, welche in der little-endian-byte-order geschickt werden. Die Checksumme besteht auch aus 2 Byte, welche in der little-endian-byte-order gesendet werden. Die Checksumme berechnet sich aus 0xFFFF weniger der Summe der ersten 30 Bytes.

Beispiel für die Dekodierung:

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Byte** | 0 | 1 | 2 | 3 | … | 30 | 31 |
| **Daten**  Protokolllänge  Command Code  0x05DC  Kanal 1  0x3412  Checksumme | 0x20 | 0x40 | 0xDC | 0x05 | … | 0x12 | 0x34 |

### Empfängersoftware

Programm oder Programm + Programmerklärung

## Inertial Measurement Unit (IMU)

### I²C Protokoll

Das I²C Protokoll besteht aus drei Leitungen: SDA (Datenleitung), SCL (Taktleitung) und GND.

Bevor jeder Datenübertragung muss eine Startbedienung erfüllt werden: während die SCL-Leitung im high-Ruhezustand ist, wechselt die SDA-Leitung auf einen low-Zustand.

Nach der Startbedingung schickt der Master im System (Flight Controller) ein 400kHz Rechtecksignal mit einen Duty Cycle von 50% auf die SCL-Leitung. Auf der SDA-Leitung werden gleichzeitig die Daten gesendet beziehungsweise empfangen und bei einer aktiven high-Flanke der SCL-Leitung interpretiert. Nach jedem Datenbyte schickt der Empfänger der Daten ein Acknowledge-Bit (0), um den Empfang der Daten zu bestätigen. Wenn das Acknowledge-Bit einen high-Pegel hat, kam bei der Übertragung ein Fehler auf.

Am Ende jeder Übertragung wird eine Stoppbedingung gesendet, die eine zeitlich invertierte Startbedingung ist. Während die SCL-Leitung sich in den high-Ruhezustand befindet, wechselt die SDA-Leitung von einen low-Zustand auch in den high-Ruhezustand.

Um zwischen mehreren I²C Geräten zu unterscheiden, hat jedes Gerät eine eigene I²C-Adresse. Die Adresse besteht aus einer 7-Bit-Zahl. Bei Kommunikation mit Sensoren muss ein achtes Bit (R/W Bit) zur Adresse hinzugefügt werden, welches bei Schreibzyklen auf 0 gesetzt werden muss und bei Lesezyklen auf 1 gesetzt werden muss.

|  |  |  |
| --- | --- | --- |
| **Sensor** | **Slave Adressen + R/W Bit** | |
| **Schreiben** | **Lesen** |
| MPU9250 | 0xD0 | 0xD1 |
| BMP280 | 0xEE | 0xEF |
| AK8963 | 0x18 | 0x19 |

Damit mit den Sensoren des IMUs kommuniziert werden kann, müssen bestimmte Lese- und Schreibabläufe eingehalten werden:

Schreibablauf:

Ein Bild, das Text, Screenshot, Reihe, Schrift enthält.

Automatisch generierte Beschreibung

|  |  |
| --- | --- |
| **Signal** | **Beschreibung** |
| S | Startbedingung |
| AD+W | Slave-Adresse + Write Bit |
| ACK | Acknowledge Bit |
| RA | Register Adresse |
| P | Stoppbedingung |

Es können mehrere Byte direkt hintereinander geschickt werden. Dabei können beliebig viele Datenbyte mit Acknowledge-Bit vor der Stoppbedingung gesendet werden. Ein interner Zeiger erhöht die Register Adresse um eine Stelle pro Datenbyte.

Leseablauf:

Ein Bild, das Text, Reihe, Screenshot, Schrift enthält.

Automatisch generierte Beschreibung

|  |  |
| --- | --- |
| **Signal** | **Beschreibung** |
| S | Startbedingung |
| AD+W | Slave-Adresse + Write Bit |
| ACK | Acknowledge Bit |
| RA | Register Adresse |
| AD+R | Slave-Adresse + Read Bit |
| NACK | not Acknowledge Bit |
| P | Stoppbedingung |

Wie beim Schreibablauf können auch mehrere Byte direkt hintereinander eingelesen werden. Dabei wird die Kommunikation mit NACK vom Master im System beendet. Ein interner Zeiger erhöht die Register Adresse um eine Stelle pro Datenbyte.

Einstellungen in CubeMX / im Programm

### Gyroskop und Accelerometer - MPU9250

#### Initialisierung

Init() Programm Aufruf + Programm selber

#### Einlesen der Daten

Programm ReadAccel, ReadGyro

#### Komplementärfilter

Programm GetAngles()

### Luftdrucksensor - BMP280

#### Initialisierung

Programm Init()?

#### Einlesen der Daten

Programm ReadBaro()

#### Berechnung der Höhe

Formel Erklärung

ReadBaro() Altitude Programm

### Magnetometer - AK8963

#### Initialisierung

Init Mag Programm

#### Einlesen der Daten

ReadMag Programm

Fehlererklärung

## Smart Battery Monitor - DS2438

### One-Wire Protokoll

Read / Write Zyklen

Einstellungen in CubeMX

### Spannungsmonitoring

ReadVoltage() Programm

## Abstandssensor - HCSR04 (Ben)

### Funktionsweise

### Code

## Motoransteuerung

### Motorstrom auslesen

ADC Erklärung

ADC Einlesen Erklärung

### DShot Protokoll

Aufbau Protokoll

Geschwindigkeiten

Init() Aufruf + Programm selber

## PID-Regler

### Hover Modus

Schalter auf Fernsteuerung Erkennung im Programm

PID-Algorithmus Programm

### Failsafe

Wann?

Wie?

Programm

## Debugging

### Terminal Ausgabe

Wofür?

Wie?

USB UART Platine

Einstellungen CubeMX

Einstellungen Terminal (VSCode, X-CTU)

### Status – LEDs

Was bedeuten Pattern

Blaue rote LED

# Datenübertragung der Mess- und Videodaten (Ben)

## Überblick Datenübertragung

Blockschaltbild und schaltung

## Kommunikation: Flight Controller und Sender

### *Aufbau Sendermodul (VTx)*

### ASK – Modulation

Überblick und wieso nicht verwendet

### Transmit – Programm auf Cortex µC

#### Gleitkommadarstellung (memcpy)

#### Code

### Testen der Übertragung

+ probleme

## Kommunikation: Sender und Empfänger

### Aufbau Empfängermodul

### Verbindungsaufbau

### Testen der Übertragung

## Kommunikation: Empfänger und Raspberry Pi

### Empfangsprogramm auf Raspberry Pi

#### UART auf dem Raspberry Pi

#### Code

### Senden der Daten auf dem MQTT-Server

## Testen der Datenübertragungskette

# Visualisierungs-App

## Applikation

### Einführung – Dart / Flutter - Framework

#### Allgemeines

#### Pub Dev / Package Installer

#### Dart Syntax

#### State Management

##### Stateful vs. Stateless Widgets

##### Provider

### Allgemeines zur App

#### Usersystem

#### Datenvisualiserung

#### 3D-Model-Viewer

#### Livestream-Viewer

### UI-Konzept

### Projektstruktur und -umgebung

#### Editor – Visual Studio Code

#### Flutter Installation

#### Projekterstellung

#### Projektstruktur

#### Packages

##### Pubspec.yaml

##### Installieren neuer Packages

### Splash + Willkommensscreen

#### Native Splash Screen

#### Willkommensbildschirm

#### Logo-Design

### Login + Registrierung

#### E-Mail-Login

#### Google-Login

#### Passwort vergessen

#### Registrierung

### Homepage

#### Bottom Navigation Bar / GNav-Bar

#### Serverdatendialog

#### Flugdatenvisualisierung

##### MQTT-Datenstreams

##### Syncfusion Cartesian Charts

##### Echtzeitdaten + Animierte Diagramme

##### Speicherung der Daten

#### 3D-Model-Viewer

##### Flutter Cube

##### MQTT-Datenstream

##### 3D-Dronenmodell

###### Optimierungen in Blender

##### Extrabedienungen

#### Live-View

##### VLC-Plugin

##### Darstellung eines Netzwerkstreams

##### Video-Overlay

### Sidemenu / Drawer

#### Userprofil

##### Darstellung der Userdaten

##### Profilbildauswahl

##### Änderung der Userdaten

#### Credits

#### Vorherige Flüge

##### Darstellung aller gespeicherten Flüge

##### Sortiermöglichkeiten

##### Einsicht in vorherigen Flug

#### Einstellungen

##### Dark- / Light-Mode

##### 3D-Model Ausrichtung

#### Logout´

## Firebase Backend

### Installation via Firebase CLI

##### Firebase CLI – Setup

##### Firebase für Dart / Flutter aktivieren

##### Flutter App mit Flutterfire konfigurieren

### Einbindung in Flutter

##### Flutterfire Configure

##### Benutzten Firebase Packages hinzufügen

### Nutzen der verschiedenen Datenbanksysteme

#### Auth

##### E-Mail + Passwort – Login

##### Google Login

#### Firestore Database

##### Struktur

##### Allgemeine Userdaten

##### Flugdaten

##### Einstellungen

#### Realtime Database

##### Allgemein

##### Sinn der verschiedenen Flags

#### Storage

##### Allgemein

##### Referenz zu User in Firestore

# Videostreaming

## Allgemeiner Aufbau

### Anforderungen

### Übertragungskette

## CADFPX Analog Kamera

### Allgemein

### Produktinformationen

### Verwendungszweck

### Verbindungstest via USB-Camera App

## USB2.0 VHS Video Grabber

## RTMP-Server via NGINX aufsetzen

### RTMP Allgemein

### NGINX

#### Allgemein

#### Installation

#### RTMP-Konfiguration

#### Serverstatus einsehen

## RTMP-Stream erstellen

### FFMPEG Allgemein

### Aufbau der Konvertierung

#### Simple Konvertierung

#### Optimierungen

##### Encodervergleich

##### Optimierungsflags

##### Unoptimiert vs. Optimiert

# Videoserver-Port Forwarding

## Allgemein

## Routereinstellungen

## Kontakt mit Routerfirma zur Freischaltung

## Testen der Ports

### Portchecker.co

## Testen des Videostreams

### VLC-Player

#### Devicestream direkt testen

#### Netzwerkstream im selben Netzwerk

#### Netzwerkstream via Public IP

# Ergebnisse

Entweder Gesamt System irgendwelche Bilder

Oder wenn nicht, dann einzelne Teile die funktionieren zeigen / beschreiben

Das Kapitel ist, um die Ergebnisse zu verkaufen, basically alles was geht

Name könnte sich noch ändern

# Anhang

### Einführung CAD – Software (Fusion 360) (Ben)

#### UI und Projekterstellung

#### Skizze anfertigen

#### Körper erstellen

#### Schrift und Bilder einfügen

### 3D – Druck (Ben)

#### 3D – Drucker

#### Filamente

#### 3D – Drucker Software (Ultimaker Cura)

### Inbetriebnahme Anleitung

### Projektplan

### Projektkosten

### Projekttagebuch

# Quellen

## Gedruckte Medien

Für die Quellenangabe bei Fachbüchern ist, wie nachfolgend dargestellt, vorzugehen (löschen):

Vorname Nachname: Titel. Untertitel. - Verlagsort: Verlag, Jahr.

Vorname Nachname: Titel. Untertitel. Auflage - Verlagsort: Verlag, Jahr.

z.B.:

[REI13] Jürgen Reichhardt: Lehrbuch Digitaltechnik - Eine Einführung mit VHDL  
 3. Auflage, München: Oldenbourg Verlag, 2013  
 ISBN: 978-3-486-72765-4

## Online

Für die Quellenangabe von Suchergebenisse in Internet ist, wie nachfolgend dargestellt, vorzugehen (löschen):

Vorname Nachname des Autors: Titel. Online in Internet: URL: www – Adresse, Datum.

(Autor und Titel wenn vorhanden, Online in Internet: www-Adresse, Datum auf jeden Fall )

z.B.:

[APA17] Apache Foundation: The Maven Project

<http://maven.apache.org/>

(Letzter Aufruf: 17.12.2017)

[TOR91] Linus Torvalds: Linux announcement

<https://web.archive.org/web/20100104211620/http://www.linux.org/people/linus_post.html>

(Erstellt: 25.08.1991, Letzter Aufruf: 17.12.2017)

# Verzeichnis der Abbildungen

# Begleitprotokoll

# Anhang

## — Projektdokumentation (Kostendarstellung, Besprechungsprotokolle etc.)

## — Technische Dokumentation (technische Beschreibungen, Berechnungen,

## Konstruktionszeichnungen, Versuchsberichte, betriebswirtschaftliche Kalkulationen etc.)

## Schaltungen, Zeichnungssätze, sonstiges