

---

iiiiii HEAD ===== lllllll f4fefd0014ec69d2fa702f3e7ca7566a10cf35d0  
iiiiii HEAD ===== lllllll f4fefd0014ec69d2fa702f3e7ca7566a10cf35d0

# Erstellung der Hard- und Software zur Steuerung der Segel im Sailwind 4 Projekt

## Projektarbeit

An der Fakultät Elektro- und Informationstechnik

vorgelegt von

**Benedikt Horn und Nico Finkbeiner**

Matrikelnummer: 300889 und 298837

Hochschule Konstanz für Technik, Wirtschaft und Gestaltung

Hochschulbetreuer:

**Prof. Dr. Boris Böck**

**Prof. Dr. Dieter Schwechten**

**Konstanz den 15. Februar 2024**

# Eidesstattliche Erklärung

Ich versichere hiermit, dass ich meine diese Bachelorarbeit mit dem Thema:

## **Erstellung der Hard- und Software zur Steuerung der Segel im Sailwind 4 Projekt**

selbstständig verfasst und keine anderen als die angegebenen Quellen und Hilfsmittel benutzt habe. Ich versichere zudem, dass die eingereichte elektronische Fassung mit der gedruckten Fassung übereinstimmt.

Konstanz, 31.12.2023

---

Nico Finkbeiner

## Kurzfassung

# Abstract

# Inhaltsverzeichnis

Eidesstattliche Erklärung	i
Kurzfassung	ii
Abkürzungsverzeichnis	vi
Abbildungsverzeichnis	vii
Quellcodeverzeichnis	viii
Vorbemerkung	1
<b>1 Einleitung</b>	<b>2</b>
1.1 Ausgangssituation . . . . .	2
1.2 Ziel . . . . .	2
<b>2 Grundlagen</b>	<b>3</b>
<b>3 Hardware</b>	<b>3</b>
iiiiiii HEAD	
3.1 Vorgehen . . . . .	4
3.2 Ziele . . . . .	4
3.3 Analyse der bestehenden Hardware Elemente . . . . .	4
3.3.1 Sensoren . . . . .	5
3.3.2 Aktoren . . . . .	6
3.3.3 Zusammenfassung und Übersicht . . . . .	8
3.4 Auswahl der Platinen Bauteile . . . . .	8
3.4.1 Mikrocontroller . . . . .	9
3.4.2 Relais . . . . .	9
3.4.3 Optokoppler . . . . .	10
3.4.4 RS485 zu UART Konverter . . . . .	10

---

3.4.5	FRAM . . . . .	10
3.4.6	Operations Verstärker . . . . .	11
3.4.7	Spannungsregler . . . . .	11
3.4.8	Platinen Entwurf . . . . .	11
3.4.9	Analoge Eingänge . . . . .	11
3.5	Gehäuse und Anschlüsse . . . . .	11
3.6	Human Machine Interface (HMI) . . . . .	12
3.7	Probleme . . . . .	13
<b>4</b>	<b>Software</b>	<b>13</b>
<b>5</b>	<b>Inbetriebnahme</b>	<b>13</b>
<b>6</b>	<b>Fazit und Ausblick</b>	<b>13</b>
=====		
3.1	Vorgehen . . . . .	3
3.2	Analyse der bestehenden Hardware Elemente . . . . .	4
3.2.1	Sensoren . . . . .	4
3.2.2	Aktoren . . . . .	6
3.2.3	Zusammenfassung und Übersicht . . . . .	7
3.3	Auswahl der Platinen Bauteile . . . . .	8
3.3.1	Mikrocontroller . . . . .	8
3.3.2	Relais . . . . .	9
3.3.3	Optokoppler . . . . .	9
3.3.4	RS485 zu UART Konverter . . . . .	10
3.3.5	FRAM . . . . .	10
3.3.6	Operations Verstärker . . . . .	10
3.3.7	Spannungsregler . . . . .	10
3.3.8	Platinen Entwurf . . . . .	11
3.3.9	Analoge Eingänge . . . . .	11
3.4	Gehäuse und Anschlüsse . . . . .	11
3.5	HMI . . . . .	11

3.6 Probleme . . . . .	12
<b>4 Software</b>	<b>12</b>
<b>5 Inbetriebnahme</b>	<b>14</b>
<b>6 Fazit und Ausblick</b>	<b>14</b>
~~~~~ f4fef0014ec69d2fa702f3e7ca7566a10cf35d0	



**ADC** Analog Digital Converter

**HMI** Human Machine Interface

## Abbildungsverzeichnis

1	Simplifiziertes 3D Modell der Lineraführung . . . . .	3
iiiiiii	HEAD	
2	HMI der Segel Steuerung . . . . .	13
=====		
2	HMI der Segel Steuerung . . . . .	12
3	Software Modulstruktur . . . . .	13
iiiiiii	f4fef0014ec69d2fa702f3e7ca7566a10cf35d0	

## Quellcodeverzeichnis

## Vorbemerkung

In der folgenden Arbeit wird aus Gründen der besseren Lesbarkeit ausschließlich die männliche Form verwendet. Sie bezieht sich auf alle Geschlechter gleichermaßen.

---

# **1 Einleitung**

Erklärung Sailwind 4.0 Projekt

## **1.1 Ausgangssituation**

Erklärung vorherige Gruppe Probleme damit

## **1.2 Ziel**

Ziele erwähnen

## 2 Grundlagen

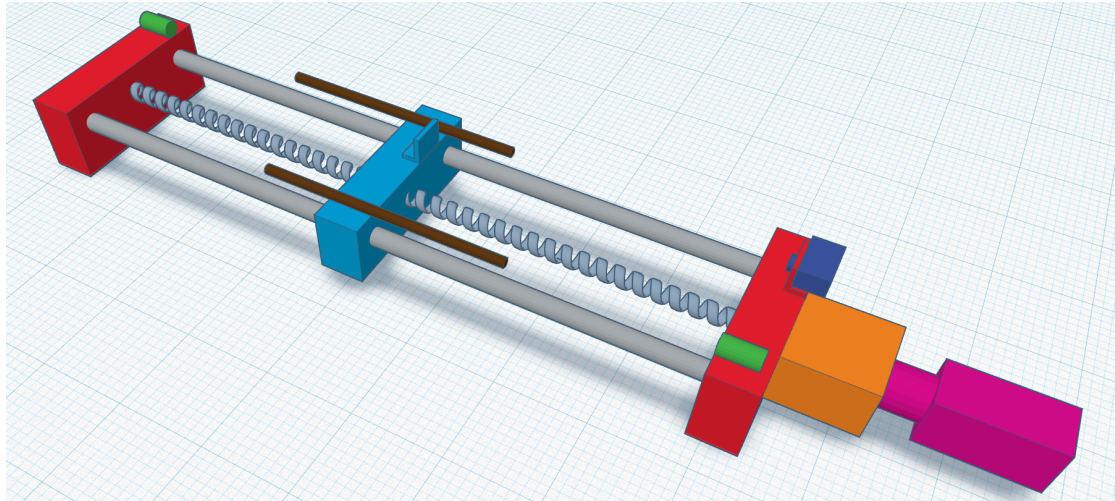


Abbildung 1: Simplifiziertes 3D Modell der Lineraführung

## 3 Hardware

Da sich das Sailwind 4.0 Projekt noch in der Entwicklung befindet, existiert noch kein komplett Zusammengebauter Prototyp in dem alle Teile des Projektes zusammenfinden. Die für diese Arbeit relevante Linearführung ist bereits als eigenstehendes Objekt zusammengebaut und wie in Abbildung 1 zusehen aufgebaut. Dabei wird die rotierende Bewegung eines BLDC Motors (Pink) mit der Hilfe eines Getriebes (Orange) und einer Gewindestange in eine Translatorische Bewegung umgewandelt, mit der der Schlitten (Hellblau) zwischen den beiden Stationären Punkten (Rot) hin und her bewegt werden kann. Die beiden Endschalter (Grün) dienen als Kollisionserkennung und können auch genutzt werden um den Bewegungsraum einzuschränken. Dabei können zwei Metallstangen (Braun) verschoben werden und geben damit vor wie viel Platz zwischen dem Schlitten und dem Stationären Elementen bleibt. Zusätzlich kann über einen Abstandssensor (dunkel Blau) die Position des Schlittens in Relation zum Abstandssensor bestimmt werden.

Die Linearführung soll später dazu genutzt werden um die Segel des Windkraftwerkes zu Trimmen und zu Rollen (siehe ??). Zusätzlich zu dem präsentierten Aufbau soll sich auf dem Schlitten später ein Druckkraftsensor befinden der die Kraft auf die Rotorwelle messen soll. Dieser wurde allerdings zum Zeitpunkt der Arbeit noch nicht geliefert oder auf dem Prototypen angebracht. Getrennt vom Aufbau soll ebenfalls ein Anemometer an der Kuppel angebracht werden, das die Windrichtung und Geschwindigkeit misst.

### 3.1 Vorgehen

### 3.2 Ziele

Die Hardware der Steuerung zu der eben vorgestellten Linearführung soll dabei folgende Aufgaben übernehmen:

- Ansteuerung aller Sensoren und Aktoren
- Stromversorgung aller Komponenten
- Lokale Bedienung der Steuerung
- Ethernetkommunikation mit dem Controllino

Neben diesen Aufgaben soll die ganze Steuerung in einem zumindest Staub- und Spritzwassergeschützten Gehäuse untergebracht werden. Eine Vorgänger Gruppe hatte sich dieser Aufgabe bereits gewidmet, allerdings war das Ergebnis dieser Gruppe leider nicht weiter verwendbar und hatte auch einen zu großen Formfaktor. Diese Probleme sollen in dieser Iteration ebenfalls behoben werden.

### 3.3 Analyse der bestehenden Hardware Elemente

Um eine geeignete Steuerplatine zu entwerfen, bedarf es einer Analyse der anzusteuenden Hardware Komponenten. Diese wurden bereits vorrausgehend festgelegt und wurden übernommen. Diese können in die Kategorien Sensoren und

Aktoren gruppiert werden. Dabei wurden die benötigte Spannung, der Stromverbrauch und die Schnittstellen betrachtet. Durch die Spannung und die Stromaufnahme kann später die benötigte Spurenbreite auf der Platine bestimmt werden. Die Schnittstellen geben an wie mit dem jeweiligen Gerät kommuniziert werden kann und welche davon benötigt werden.

### 3.3.1 Sensoren

Zu den Sensoren zählen die folgenden Komponenten:

- Induktiver Endschalter: IFM IFS204
- Optischer Abstandssensor: IFM OGD580
- Anemometer: MESA WSWD
- Druckkraft-Sensor: Burster 8532

#### Induktiver Endschalter

Zwei der IFM IFS204 Endschalter sind im Design mit eingebaut. Sie funktionieren als PNP-Schließkontakt und geben an einem Ausgang ein 24V high Signal aus, sobald sie ausgelöst werden. Dabei hat jeder Endschalter eine typische Stromaufnahme von 10mA und eine Betriebsspannung von 9-30V.

#### Optischer Abstandssensor

Der Abstandssensor OGD580 funktioniert über einen Laser der vom Gerät ausgehend über eine reflektive Fläche zu diesem zurückgeworfen wird. Der Abstandssensor verfügt über ein Display das den gemessenen Abstand anzeigt und über den das Gerät konfigurierbar ist. Er hat eine typische Stromaufnahme von 10mA und ebenfalls eine Betriebsspannung von 9-30V. Der Abstand wird über einen Digitalen Ausgang, dem sog. IO-Link ausgegeben. Da dieser typischerweise nur in der Automobilbranche zum Einsatz kommt, wurde ein zusätzlicher IO-Link Konverter hinzugefügt.



Der EIO104 konvertiert die digitale IO-Link Schnittstelle zu einer Analogen 4-20mA Schnittstelle mit der einfacher Umgegangen werden kann. Dabei kommt ein zusätzlicher Stromverbrauch von ???mA hinzu.

### **Anemometer**

Das WSWD Anemometer wird genutzt um die Windrichtung als auch die Windgeschwindigkeit zu messen. Dieser basiert ebenfalls auf einer 24V Versorgungsspannung und hat je nach Konfiguration und Ausführung eine Stromaufnahme von 120mA. Er besitzt ebenfalls ein Heizelement um ihn bei sehr niedrigen Temperaturen nutzen zu können. Dieses wird allerdings im Einsatzszenario und im Prototyp nicht benötigt. Die Schnittstellen des Anemometers sind abhängig von dessen Ausführung. In der zum Einsatzkommenden Ausführung, dem WSWD1, kann neben einer Analogen Schnittstelle die Daten auch über eine Digitale RS485/RS422 Schnittstelle abgefragt werden. Die Werte können Analog entweder als ein 0/4-20/24mA Signal oder als ein 0/2-8/10V Signal ausgegeben werden. Die Digitale Schnittstelle kann neben der Datenausgabe auch zur Konfiguration des Gerätes genutzt werden und bietet eine Vielzahl an Protokollen zur Kommunikation an.

### **Druckkraft-Sensor**

Der Burster Druckkraft-Sensor war der einzige Sensor der zum Zeitpunkt der Arbeit noch nicht Bestellt wurde. Dieser wurde aber dennoch mit eingeplant. Der Druckkraft Sensor kann ebenfalls mit den typischen 24V betrieben werden und hat dabei eine Stromaufnahme von ca. 12,5mA. Die Messwerte werden hier über eine Analoge 0-10V Schnittstelle übertragen.

#### **3.3.2 Aktoren**

Zu den Aktoren zählen die folgenden Komponenten:

- Gleichstrommotor: Dunkermotoren BG 45x30 SI
- Externe Relais

### Gleichstrommotor

Der Dunkermotor BG 45x30 SI Gleichstrommotor ist das Kernelement des Aufbaus. Da der Motor eine interne Regelung besitzt hat er eine getrennte Leistungs- und Logikversorgung. Der Motor an sich wird dabei über den Leistungsteil bestromt, während über den Logikteil dieser gesteuert werden kann und Feedback bereitstellt. Die Betriebsspannung der Leistungs- und Logikversorgung sind 24V, wobei der Motor einen maximal zulässigen Dauerstrom von 3,8A ausgesetzt sein darf. Die Logikversorgung hat eine Stromaufnahme von 100mA. Die Kommunikation mit dem Motor findet über vier Digitale- und einen Analogen Eingang statt. Der Motor stellt Feedback zum aktuellen Status über drei Ausgänge bereit. Diese geben die Drehrichtung, aktuelle Störungen und die Drehgeschwindigkeit des Motors an.

Eingang 1 (IN1)	Eingang 0 (IN0)	Funktion
0	0	Motor aus
0	1	Linkslauf
1	0	Rechtslauf
1	1	Stopp mit Haltemoment
Eingang 3 (IN3)	Eingang 2 (IN2)	
0	0	Drehzahlvorgabe Analog
0	1	Stromvorgabe Analog
1	0	Geschwindigkeit 1
1	1	Geschwindigkeit 2

Tabelle 1: Eingänge und Funktionen des BG 45x30 SI

Hier oder später noch Ein und Ausgänge definieren in Tabelle bzw. in Text erwähnen

### Externe Relais

Zum Zeitpunkt der Arbeit gab es noch keinen konkreten Verwendungszweck der

externen relais diese wurden für zusätzliche Funktionalitäten dennoch mit eingepplant und sollten über ein 24V Signal geschaltet werden.

### Controllino

Der Controllino ist der zentrale Microcontroller und verwaltet alle Aktoren im System. Über diesen soll das hier zu entwerfende System, seine Befehle zur Segelausrichtung bekommen. Der Datenaustausch soll hier über eine Ethernet Schnittstelle stattfinden, um die Große Distanz zwischen Kuppel und Basis zu überbrücken.

#### 3.3.3 Zusammenfassung und Übersicht

Damit sind nun alle externen anzusteuenden Komponenten abgedeckt. Die benötigte Stromversorgung sowie alle Schnittstellen sind dabei in Tabelle 2 dargestellt. Dabei kann bereits entnommen werden, das es sehr sinnvoll ist die Platine auf ein 24V System auszulegen und diese auch mit dieser Spannung zu versorgen.

Element	Pinout										
Endschalter 1	24V	GND	OUT								
Endschalter 2	24V	GND	OUT								
Abstandssensor	24V	GND	IO-Link								
IO-Link Konveter	24V	GND	AOUT								
Anemometer	24V	GND	AOUT	AOUT	AGND	RS485+	RS485-				
Druckkraftsensor	24V	GND	AOUT	AGND							
Motor Leistung	24V	GND									
Motor Logik	24V	GND	IN0	IN1	IN2	IN3	AIN	AGND	OUT1	OUT2	OUT3
Externes Relais 1	OUT	GND									
Externes Relais 2	OUT	GND									
Controllino	Ethernet										

Tabelle 2: Übersicht Stromversorgung und Schnittstellen der externen Elemente

### 3.4 Auswahl der Platinen Bauteile

Da nun klar ist welche Anforderungen durch die anzuschließenden Geräte bestehen, kann auf Basis dieser nun weiterverfahren werden. Dabei soll im folgenden ein geeigneter Mikrocontroller, sowie die nötigen Platinenkomponenten ausgewählt wer-

den, um mit den Hardwarekomponenten zu kommunizieren und diese mit Strom zu versorgen.

### 3.4.1 Mikrocontroller

Die Auswahl des Mikrocontrollers wurde auf der Basis folgender Kriterien beschlossen:

- Anzahl Inputs und Outputs
- Architektur
- Softwaresupport
- Entwicklung
- Verfügbarkeit

Dabei gab es die Option den Chip als einzelnes Element direkt auf die selbst erstellte Platine einzubetten oder ein Entwicklungsboard als Basis zu nutzen und eine Erweiterungsplatine dafür zu nutzen. Es wurde sich dabei für ein Entwicklungsboard entschieden um bereits parallel zum Hardwaredesign eine geeignete Software zu entwickeln und diese bereits mit einem Prototypischen Aufbau zu testen (Vielleicht noch erwähnen, probleme bei vorheriger Gruppe). Dies bietet ebenfalls den Vorteil, das bei fatalen Fehlern ein schneller Ersatz angebracht werden kann und auch bereits ein Formfaktor vorgegeben ist.

Durch die große Anzahl an benötigten Inputs und Outputs und den relativ großen Anspruch an Onboard Speicher z.B für den geplanten Webserver, wurde sich für eine 32Bit Architektur entschieden.

### 3.4.2 Relais

Um das Endschalter Ausgangssignal an den Mikrocontroller weiterzugeben, werden zwei 24V Relais genutzt die bei der Aktivierung der Spule ein 3,3V Signal an den

Eingang des Mikrocontrollers weitergeben. Gleichzeitig wird durch das Umschalten des Relais der Eingang 1 oder 2 auf Null gesetzt, so dass dieser direkt ausgeschaltet wird um eine Kollision mit den Enden der Linearführung vorzubeugen.

### 3.4.3 Optokoppler

Um die restlichen 24V Aus- und Eingänge zu steuern oder auszulesen kommen Optokoppler zum Einsatz. Diese isolieren ähnlich zu Relais den 24V und 3,3V Schaltkreis und ermöglichen es durch eine kleinere Spannung eine größere Last zu schalten. Dabei werden insgesamt 9 benötigt um die digitalen Ein- und Ausgänge des Motors und die beiden externen Relais zu schalten. Da es sich beim Ausgang 3? des Motors um ein pulsierendes Signal zur Ermittlung der Drehgeschw. des Motors handelt wurde auf eine schnelle Schaltzeit der Optokoppler geachtet (hier Berechnung einfügen).

### 3.4.4 RS485 zu UART Konverter

Um das Anemometer über die RS485 Schnittstelle zu konfigurieren und Daten abzufragen wird ein zusätzlicher Baustein benötigt. Dieser konvertiert das differenzielle RS485 Signal zu einem nicht differenziellen UART Signal, das der Microcontroller unterstützt.

### 3.4.5 FRAM

Da die mittlere Position des Segels eingestellt werden können soll, wird ein nicht flüchtiger Speicher benötigt in dem die Position gespeichert wird. Dabei ist der FRAM eine der billigsten Speichermöglichkeiten für diesen Zweck, dabei wurde die geringste Speichergröße gewählt, da dieser wie ein EEPROM jederzeit überschrieben werden kann.

### 3.4.6 Operations Verstärker

Um die Drehzahl des Motors vorgeben zu können, wird ein Analoges Signal im Bereich von 0-10V benötigt. Da der Mikrocontroller bei einer Versorgungsspannung von 5V maximal ein Analoges Signal von 5V Ausgeben kann, wird ein Operationsverstärker benötigt. Dieser soll mit der 24V Versorgungsspannung das 0-5V Signal verstärken und auf die 0-10V skalieren.

### 3.4.7 Spannungsregler

Um alle Komponenten zu versorgen Bedarf es insgesamt 3 verschiedenen Spannungspotenzialen: 24V, 5V und 3,3V. Um dies zu erreichen sollen alle externen Elemente direkt mit den 24V versorgt werden. Der Mikrocontroller soll mit den 5V versorgt werden, da er somit auch ohne Debugger Erweiterung genutzt werden kann. Hierfür wird ein Step Down konverter genutzt der die 24V Versorgungsspannung auf 5V herunterbricht. mit diesen 5V kann daraufhin ein Festspannungsregler betrieben werden der die 5V auf eine 3,3V Spannung herunter bringt mit einem sehr geringen ripple.

### 3.4.8 Platinen Entwurf

Hier Schaltplan und Gerendertes Bild einfügen

### 3.4.9 Analoge Eingänge

Um die Analogen Ausgänge des Abstandssensors und des Anemometers auszulesen wird das 4-20mA Stromsignal über einen Widerstand zu einem 0-3V Spannungssignal gewandelt das mit dem Analog Digital Converter (ADC) des Microcontrollers gelesen werden kann. Spannungsteiler Strom zu Spannung Widerstände

## 3.5 Gehäuse und Anschlüsse

Hier Freecad Bild einfügen für mount und Gehäuse Das Gehäuse sollte vorallem im Innenraum genug Platz bieten um Kabel an der Platine zu befestigen, eine

Möglichkeit bieten die Platine darin zu befestigen und genug Platz auf der Oberfläche bieten um das HMI und die unterschiedlichen Kabel Durchführungen zur Außenseite zu leiten. Dabei sollte das Gehäuse allerdings nicht sehr große und trotzdem Platz sparend bleiben

### 3.6 HMI

Das HMI besteht aus einer Reihe von Kippschaltern, Knöpfen und LEDs. Diese sind in Abbildung 2 dargestellt. Dabei soll die gewünschte Neutrale Position der Linearführung durch einen Manuellen Betrieb stattfinden. Dabei kann der Endnutzer über die Trimm und Roll Knöpfe zum gewünschten Mittelpunkt Navigieren und diesen mit dem dritten Knopf speichern. Anschließend kann mittels des Kippschalters in den Automatik Betrieb gewechselt werden in dem das betätigen der Knöpfe ignoriert wird. Eine Reihe von LEDs gibt Feedback über den aktuellen Zustand der Linearführung. Dabei wird konstant der aktuelle Betriebsmodus durch zwei LEDs angegeben. Ebenfalls wird die Position relativ zur festgelegten neutral Position durch zwei gelbe LEDs angegeben. Eine Störung in der Steuerung oder im Motor wird durch eine rote LED angegeben.

Die Knöpfe und Kippschalter sind beide am Gehäuse montiert und sind Spritzwasser geschützt. Im Gegensatz dazu sind die LEDs direkt auf der Platine platziert und werden durch flexible Lichtleiter zur Außenseite des Gehäuses geführt.

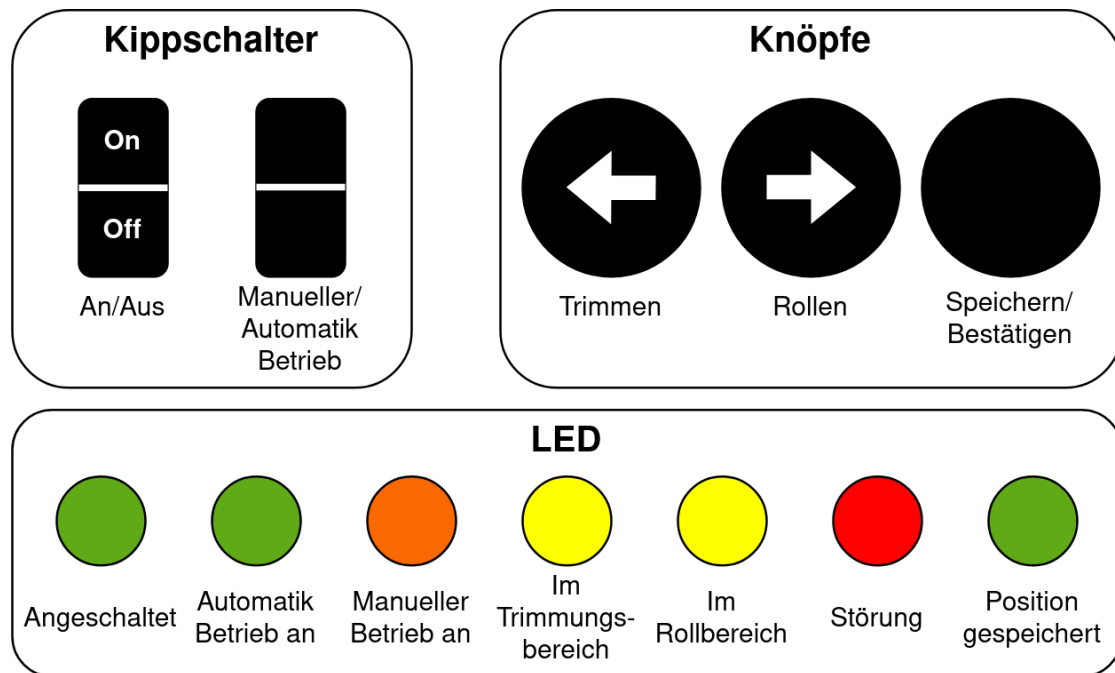


Abbildung 2: HMI der Segel Steuerung

### 3.7 Probleme

Klemmenlöcher Operationsverstärker Widerstände RS485 FRAM MOSI/MISO

## 4 Software

Die Steuerung der Segel soll durch zwei verteilte Systeme erfolgen, die über eine Ethernet-Verbindung miteinander kommunizieren. Die Idee dahinter ist, die Berechnungen für die Ausrichtung der Segel in Abhängigkeit der Windstärke und -Richtung (Controllino) von der hardwarenahen Verarbeitung der Sensoren und Aktuatoren (stm32) logisch zu trennen.

Die vorliegende Arbeit konzentriert sich dabei lediglich auf das System der Sensoren und Aktuatoren, welches mit dem stm32 realisiert wurde. Die Aufgabe dieses Systems ist, zunächst die Linearführung zu kalibrieren, damit die aktuelle Positi-



on korrekt ermittelt werden kann. Außerdem soll über die Buttons am Gehäuse und zusätzlich über einen Webserver eine manuelle Steuerung ermöglicht werden. Während des Automatikbetriebs geschieht ein regelmäßiger Austausch aller relevanter Daten über eine REST basierte Schnittstelle. Dies beinhaltet u.A. die Information über aktuelle Windbedingung, Position der Segel, eventuelle Fehlerzustände (z.B. Motorfehler oder Überstrom) und daraus resultierende Befehle zur Anpassung der Segelstellung. Für die Implementierung wurde die Software in komponentenorientierte Module eingeteilt:

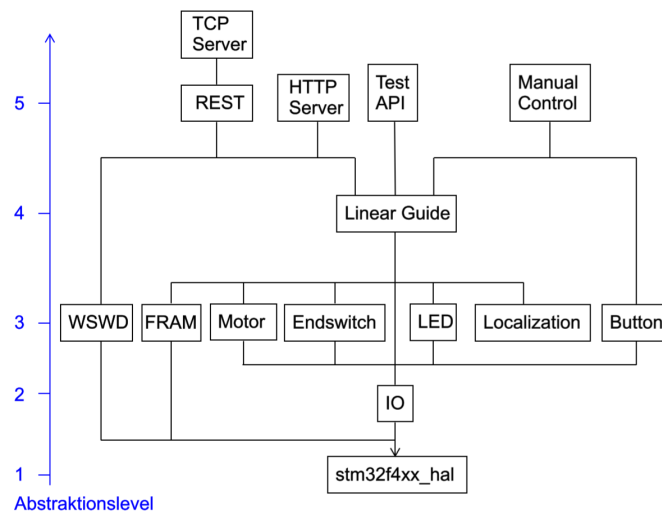


Abbildung 3: Software Modulstruktur

Abbildung 3 zeigt abwärtsgerichtet, wie die einzelnen Module aufeinander zugreifen. Auf unterster Abstraktionsebene laufen jegliche Operationen über die Standardbibliothek des Mikrocontrollers `stm32f4xx_hal`. Darüber liefert das `IO`-Modul ein Set aus Hilfsfunktionen und -Strukturen für den allgemeinen Zugriff auf die GPIO-Pins und zum Auslesen analoger Messwerte der einzelnen Sensoren. Ebene 3 umfasst hauptsächlich Module, welche alle relevanten Funktionalitäten der physischen Teilkomponenten des Systems implementieren, wie z.B. das Anemometer `WSWD` oder der `Motor`. Einige davon greifen dabei auf das `IO`-Modul zu, wobei der allgemeine GPIO-Zugriff von einer spezifischen Funktion eingekapselt wird,

wie z.B. das Einschalten einer LED im Falle des **LED-Moduls**. Eine Ausnahme ist das **Localization**-Modul, welches keine physische Komponente darstellt, sondern einige Hilfsfunktionen zur Kalibrierung und Positionsberechnung bereitstellt. Während die Module bis Ebene 3 überwiegend allgemeingültig entworfen sind, enthält das **Linear Guide**-Modul anwendungsspezifische Funktionen. Als zentrales Element bildet dieses ein High-Level Interface zur Verwendung der Teilkomponenten **FRAM**, **Motor**, **Endswitch**, **LED**, **Localization** und **IO**. Die Module der obersten Abstraktionsebene bilden die direkten Schnittstellen zur Außenwelt. Das **Manual Control**-Modul ermöglicht die manuelle Steuerung der Linearführung über die User-Buttons am Gehäuse und insbesondere die Umsetzung des Kalibrierungsprozesses. Auf der anderen Seite kann das System auch durch ein HTTP Webserver überwacht und gesteuert werden. Zuletzt werden im **REST**-Modul die Anfragen des Controllinos über die **TCP Server**-Verbindung verarbeitet, wie bereits oben erwähnt.

## 5 Inbetriebnahme

## 6 Fazit und Ausblick