Ernst-Abbe-Hochschule Jena
University of Applied Sciences

Fachbereich Elektrotechnik und Informationstechnik

Projektabschlussbericht

Entwicklung und Implementierung der Regelungs- und Steuerungsstrategien

Halbgewachs, Sascha	646980
In der Au, Christoph	641626
Kirchner, Martin	641335
Majid, Muhammad Maaz	641712
Neumeister, Lukas	641334
Zhang, Chen	647914

Jena, März 2021

Inhalt

Inhalt II

Abbildungsverzeichnis IV

Tabellenverzeichnis V

Abkürzungsverzeichnis VI

Symbolverzeichnis VII

1 Auftrag 1

1.1 Ausgangssituation 1

1.2 Projektziele 1

1.3 Projektorganisation 1

2 Evaluierung 3

2.1 Projektverlauf 3

2.2 Zielerreichung 3

3 Ergebnisse 4

3.1 Kommunikation 4

3.1.1 Hardware 4

3.1.1.1 Platinenlayout 6

3.1.1.2 STM32 Entwicklungsboard Nucelo-32 (NUCLEO-G431KB) 7

3.1.1.3 Weitere Bauteile 9

3.1.2 Software 13

3.1.2.1 Anwenderschnittstelle 13

3.1.2.1.1 UART-Schnittstelle 13

3.1.2.1.2 UART1 und UART2 14

3.1.2.2 communication.c und communication.h 14

3.1.2.2.1 Funktionen 15

3.1.2.2.1.1 Funktion InitCom 15

3.1.2.2.1.2 Funktion updateSensorBlock 15

3.1.2.2.1.3 Funktion updateFuehrungsblock 15
3.1.2.2.1.4 Funktion updateLocalData..... 15
3.1.2.2.1.5 Funktion readData 15
3.1.2.2.1.6 Funktion updateStellgrBlock 16
3.1.2.2.1.7 Funktion updateBusData 16
3.1.2.2.1.8 Funktion sendData 16
3.1.2.2.1.9 Funktion CRC8 17
3.1.2.2.2 Codeablauf 17
3.1.2.2.2.1 Daten schreiben 17
3.1.2.2.2.2 Daten lesen..... 18
3.1.2.3 Aktuelle Softwareprobleme 18
3.2 Regelstrategie 19
3.2.1 Was ist das Ziel der Regelstrategie?..... 19
3.2.2 Fahrt von A nach B 20
3.2.3 Methode Heading Point..... 22
3.2.4 Implementation im Quellcode 24
4 Ausblick..... 25
5 Entlastung Projektteam..... 26
6 Literaturverzeichnis 27
Anlagen 28

Abbildungsverzeichnis

Abbildung 1-1: Gruppenunterteilung 2
Abbildung 3-1: Übersicht; 1: Katamaran; 2: Sitz der Platinen [1] 4
Abbildung 3-2: Übersicht; 1: Slave; 2: USER-Unit; 3: Gehäuse ohne Deckel 5
Abbildung 3-3: User-Unit; 1: Platinenlayout; 2: Entwicklungsboard; 3: Weitere Bauteile 5
Abbildung 3-4: Keep-Out-Areas (violett) 6
Abbildung 3-5: Platine (Oberseite) ohne Bestückung; Herstellung bei AISLER; Markierungen: Keep-Out-Areas; Maße der Platine LxBxH: 50,1x70,8x1,6 mm ... 7
Abbildung 3-6: Entwicklungsboard STM32G431KB [2] 7
Abbildung 3-7: Konfiguration der PINs über Software STM32CubeIDE 1.5.0 8
Abbildung 3-8: Modifiziertes Entwicklungsboard; rote Markierungen: Entfernte Lötbrücken SB15 (links) und SB5 (rechts) [2] 9
Abbildung 3-9: Übersicht finale Platine; 1: J3 mit Flachbandkabel; 2: J2 mit Flachbandkabel; 3: J1-Headeranschlussleiste; 4:J2-Headeranschlussleiste; 5: LEDs; 6:NUCLEO-Board 10
Abbildung -3-10: Lichtintensität in Abhängigkeit vom Strom; Beispiel: LED (blau); Bei Lichtintensität = 1000 mcd: ca. Strom I: 8 mA [3] 11
Abbildung 3-11: Strom in Abhängigkeit der Spannung; Beispiel: LED (blau); Strom = 7 mA; Spannung = 2,75 V [3] 11
Abbildung 3-12: Finales Boardlayout 12
Abbildung 3-13: ideale Bahn 19
Abbildung 3-14: Fahrtverlauf ideale und reale Bahn 20
Abbildung 3-15: Fahrt von A nach B - Toleranzgebiete 21
Abbildung 3-16: Fahrt von A nach B - Epsilonschlauch 21
Abbildung 3-17: Verlassen des Epsiolonschlauches 22
Abbildung 3-18: Heading Point-Methode 22
Abbildung 3-19: Fehlerabhängigkeit (Distanz: Boot - ideale Bahn) 23
Abbildung 3-20: Zusammenfassung Heading Point Methode Teil 1 23
Abbildung 3-21: Zusammenfassung Heading Point Methode Teil 2 23

Tabellenverzeichnis

Tabelle 3-1: Verwendete PINs beim Entwicklungsboard STM32G431KB 8
Tabelle 3-2: Übersicht Vorwiderstände der LEDs 12

Abkürzungsverzeichnis

ADC	Analog-zu-Digital-Wandler
ID	Identifikation
IRQ	Interrupt request
CRC	Cyclic redundancy check
CTS	Clear to send
RTS	Request to send
SMD	Surface-mounted-devices
THD	Through-hole-devices
VIA	Vertical interconnect access
WE	Würth Electronic

Symbolverzeichnis

I_D	Durchlassstrom	A
I_V	Lichtintensität	cd
R	Widerstand	Ω
U_D	Durchlassspannung	V

1 Auftrag

1.1 Ausgangssituation

Im Rahmen des Mechatronik Projekts 2020/21 wird ein Wasserfahrzeug entwickelt, welches autonom eine vorgegebene Route fährt. Dabei muss eine Kollision mit bewegten und unbewegten Hindernissen vermieden werden. Der Bau des Wasserfahrzeugs wurde bereits vorher abgeschlossen. Weiterhin sind das Antriebskonzept und das Steuerkonzept auch vor Beginn des Projekts abgeschlossen bzw. nicht im Rahmen dieses Projekts behandelt.

Das Hauptziel dieses Projekts ist die Entwicklung und Implementierung der Regelungs- und Steuerungsstrategien.

1.2 Projektziele

Wie bereits beschrieben, ist die Entwicklung und Implementierung der Regelungs- und Steuerungsstrategien das Hauptziel dieses Projektes. Folgende Aufgaben müssen dabei bearbeitet werden:

- Aufarbeitung der Positionsdaten von GPS und Kompass
- Erfassung der Daten für die Bahnsteuerung
- Regelung des Kurswinkels
- Regelung der Geradeausfahrt
- Anfahren von Zwischen- und Zielpunkten
- Projektdokumentation

1.3 Projektorganisation

Auftraggeber: Herr Prof. Dr.-Ing. Jörg Grabow, Herr Dipl.-Ing. Stefan Franke

Projektleiter: Muhammad Maaz Majid

Projektmitarbeiter: Sascha Halbgewachs, Christoph In der Au, Martin Kirchner, Lukas Neumeister, Chen Zhang

Die Gruppe wird in das „Team Kommunikation" und in das „Team Regelung" unterteilt. Für beide Gruppen wird ein Aufgabenleiter bestimmt, welcher die Arbeitspakete unterteilt und dem Projektleiter ein Status-Update liefert. Das „Team Kommunikation", bestehend aus Lukas und Christoph, bearbeiten das Bussystem und die Hardware und

testen das Bussystem bei Fertigstellung, um die Funktionalität zu gewährleisten. Die Gruppe „Regelung", bestehend aus Chen, Martin, Maaz und Sascha, bearbeiten den Regler und das Streckenmodell, entwickeln eine Regelungsstrategie und führen Simulationen durch.

Nach der Fertigstellung der Gruppenaufgaben wird gemeinsam das Reglermodell in der Programmiersprache C implementiert sowie die Dokumentation verfasst. In der folgenden Abbildung ist die Gruppenunterteilung aufgezeigt.

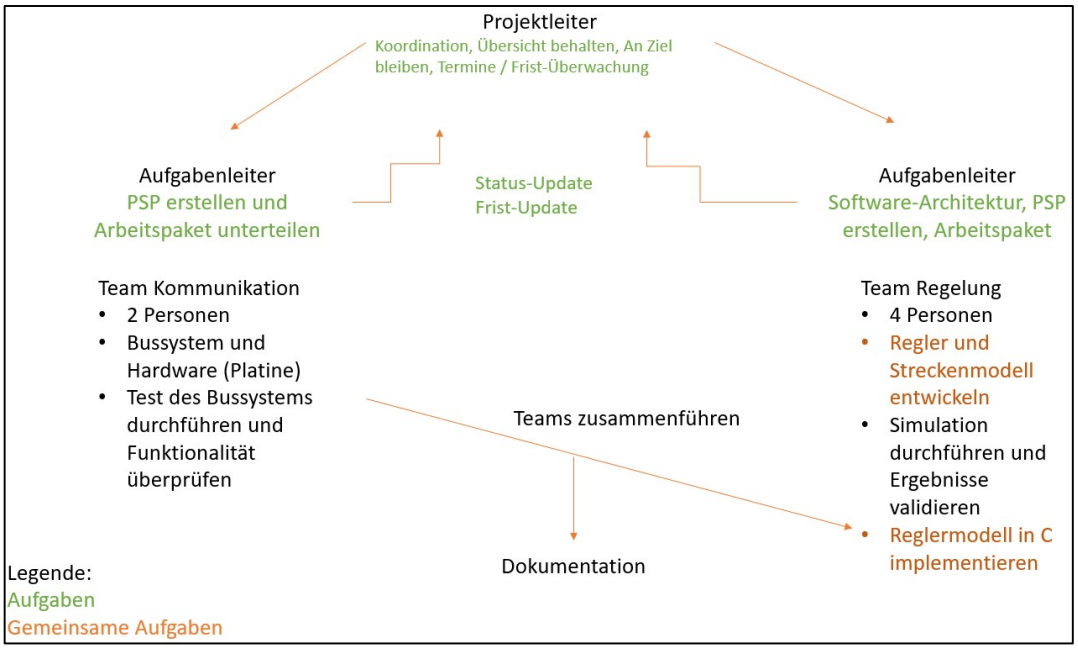


Abbildung 1-1: Gruppenunterteilung

2 Evaluierung

2.1 Projektverlauf

Das Projekt wurde in KW 47 (2020) begonnen und wird in KW 12 (2021) abgeschlossen. Die wesentlichen Aufgaben werden in Teilaufgaben gegliedert und den Gruppen „Team Kommunikation“ und „Team Regelung“ zugeteilt. Beide Teams treffen sich wöchentlich mit dem Projektleiter und informieren die Gruppe über den aktuellen Status der Aufgaben. Die Teilaufgaben bilden den Meilenstein des Projekts. Die Terminplanung bzw. die Aufgabenteilung befindet sich in den Anlagen. Alle Termine werden eingehalten und alle Teilaufgaben werden erfolgreich umgesetzt.

Es gab einige Schwierigkeiten während der Bearbeitung. Aufgrund der Pandemie musste alles (z.B. die Absprache, Meetings) online stattfinden. Dieses Problem wurde aber durch enge Zusammenarbeit und motivierte Teamkollegen kompensiert. Durch die Koordinierung des Projektleiters und der Aufgabenleiter konnte jederzeit identifiziert werden, welcher Mitarbeiter welche Aufgabe momentan bearbeitet und was die nächsten Schritte sind.

2.2 Zielerreichung

Alle Projektziele werden, bis auf die Implementierung des Brownouts und des Watchdogs, erfolgreich innerhalb der entsprechenden Fristen umgesetzt. Die Regelungsstrategie und Kommunikation wurde im Rahmen dieses Projekt konzeptioniert, erfolgreich implementiert und validiert. Neben der Softwareentwicklung wurde auch die Hardware entwickelt. Die Auswahl eines geeigneten Microcontrollers, die Entwicklung und Bestückung der zugehörigen Platine und ausschließlich die Inbetriebnahme der Hardware wurde im Rahmen dieses Projekt erfolgreich umgesetzt.

3 Ergebnisse

In diesem Kapitel werden die Projektergebnisse vorgestellt. Begonnen wird mit der Kommunikation, bei der die entwickelte Platine und die erstellten Programme beschrieben werden. Im zweiten Punkt wird die Regelstrategie dargestellt, welche Probleme während der Bootsfahrt auftreten können und wie diese behoben werden.

3.1 Kommunikation

Dieses Kapitel befasst sich mit den Schnittstellen und der Kommunikation auf dem Katamaran, speziell für die Regelung und Regelstrategie. Auf dem Katamaran befindet sich ein Master, ein Slave und ein Anwender (User-Unit). Der Master kommuniziert mit dem Slave über die Bus-Schnittstelle RS485. Der Slave kommuniziert mit der User-Unit über dem RS232-Busprotokoll. In diesem Kapitel wird nur auf die Kommunikation zwischen Slave und User-Unit eingegangen.

Die Slave-Platine wurde von Laboringenieur Herrn Dipl.-Ing. Stefan Franke entwickelt und bildet die Verbindung und Grundlage zur User-Unit-Platine. Die User-Unit-Platine wurde von den Studenten Christoph In der Au und Lukas Neumeister entwickelt. In Abbildung 3-1 ist der Katamaran mit dem Sitz der Platinen dargestellt.

Zuerst wird ein Einblick in die Entwicklung der Platine (Hardware) und danach in die der Kommunikation (Software) gegeben.

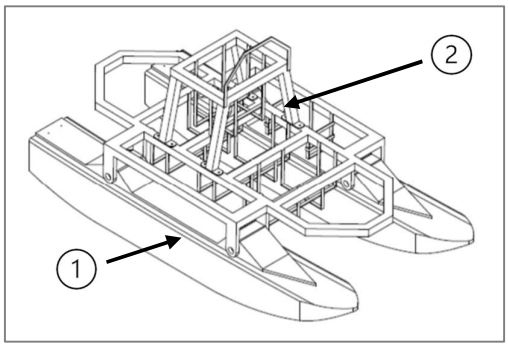


Abbildung 3-1: Übersicht; 1: Katamaran; 2: Sitz der Platinen [1]

3.1.1 Hardware

Die User-Unit zur Kommunikation mit dem Slave ist ein elementarer Bestandteil und bildet die Grundlage der Kommunikation. Dieser Entwicklungsprozess beinhaltet viele Vorüberlegungen, Anforderungen, Berechnungen und praktische Tests. Genau auf diesen Prozess wird in dem Kapitel eingegangen, anschaulich dargestellt und genauer erläutert.

Die Slave-Platine und User-Unit-Platine sind zusammen in einem gemeinsamen Gehäuse untergebracht (siehe Abbildung 3-2). Durch die direkte Kommunikation sind kurze Übertragungswege vom Vorteil (siehe Anlage 10)

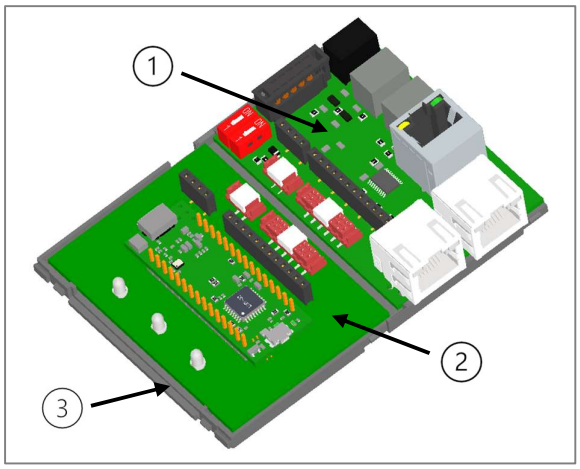


Abbildung 3-2: Übersicht; 1: Slave; 2: USER-Unit; 3: Gehäuse ohne Deckel

Die Slave-Platine wurde von Herrn Dipl.-Ing. Stefan Franke entwickelt, in Betrieb genommen und erfolgreich getestet. Dieser Prozess beinhaltet die Entwicklung der Software und Hardware.

Die User-Unit-Platine wurde in Kooperation der beiden Studenten Christoph In der Au (Software) und Lukas Neumeister (Hardware) entwickelt und umgesetzt. Die Entwicklung der Platine (Abbildung 3-3) gliedert sich in folgende drei Teilaufgaben auf:

- Platinenlayout
- Entwicklungsboard
- Weitere Bauteile

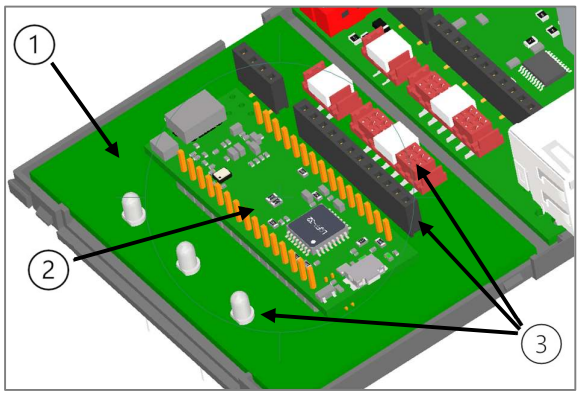


Abbildung 3-3: User-Unit; 1: Platinenlayout; 2: Entwicklungsboard; 3: Weitere Bauteile

In der Anlage 3 ist die Anforderungsliste der Hardware und der Teilaufgaben, dargestellt. In den folgenden Kapiteln werden die Teilaufgaben genauer beschrieben und auf wichtige Schritte der Entwicklung, für einen weiteren erfolgreichen Verlauf des Projekts, eingegangen.

3.1.1.1 Platinenlayout

Das Platinenlayout wird mit EAGLE designed und besteht aus einem Schaltplan und einem Board-Layout. Die Software EAGLE ist ein EDA-Programm der Firma CadSoft. Die Fertigung der Leiterplatte übernimmt der europäische Hersteller AISLER.

Das Platinenlayout beinhaltet einen Schaltplan (Anlage 5) und ein Board-Layout (Anlage 6). Verwendet wird eine 2-Layer-Platine mit einem Bottom- und Top-Layer. Auf dem Board-Layout sind die Keep-Out-Areas zu beachten, zur Befestigung im Gehäuse. In diesen Bereichen dürfen keine Bauteile positioniert werden (siehe Abbildung 3-4). Allgemeine Hinweise zu EAGLE (Anlage 8) und eine EAGLE-Vorlage (Anlage 7) sind in der Anlage dargestellt.

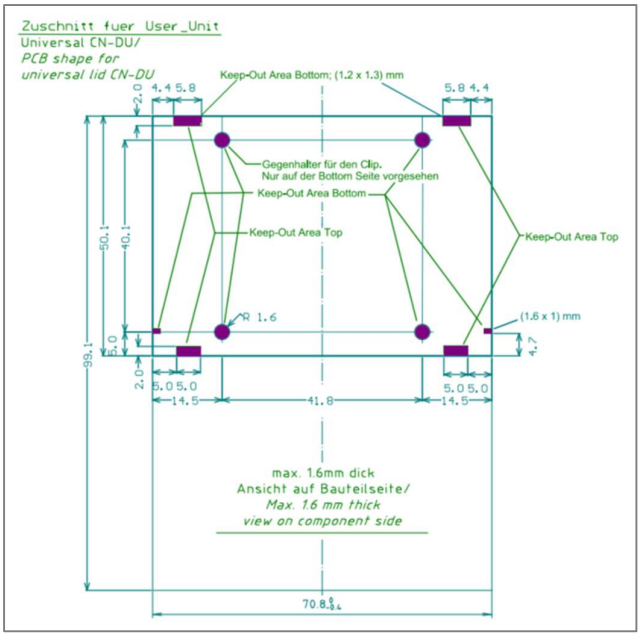


Abbildung 3-4: Keep-Out-Areas (violett) [1]

In den Design Rules sind alle wichtigen Kenngrößen zusammengefasst, um das Platinenlayout erfolgreich zu verwenden. Die Designs werden bestimmt durch die verwendeten SMD-Bauteile (0805), Leitungsbreite, Abstände, VIAs etc.. In Abbildung 3-5 ist die Platine dargestellt.

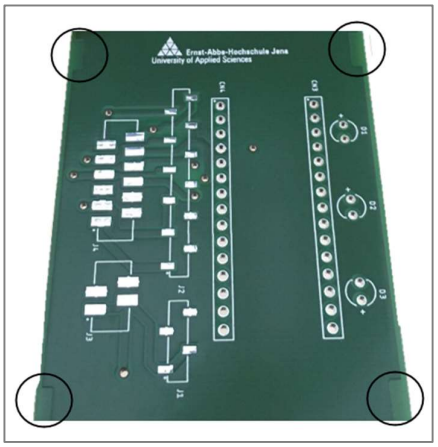


Abbildung 3-5: Platine (Oberseite) ohne Bestückung; Herstellung bei AISLER; Markierungen: Keep-Out-Areas; Maße der Platine LxBxH: 50,1x70,8x1,6 mm

3.1.1.2 STM32 Entwicklungsboard Nucelo-32 (NUCLEO-G431KB)

Anhand einer Vorauswahl und der gestellten Anforderungsliste (Anlage 3) wird das Entwicklungsboard (Abbildung 3-6) vom Hersteller STMicroelectronics aus der Typenreihen NUCLEO verwendet. Die Anforderungsliste an das Entwicklungsboard gibt weiteren Aufschluss zur Entscheidung.

Es folgt eine kurze Übersicht mit wichtigen stichpunktartigen Kenndaten des Boards:

- 170MHz ARM Cortex M4 32 bit CPU
- 128 KB Flash
- Betriebsspannung: 3,3 V DC
- On-board debugger/programmer
- FPU single precision
- Maße (LxBxH): 51 x 19 x 25 mm
- Preis: ~13€ inkl. MwSt.
- Software: STM32CubeIDE 1.5.0

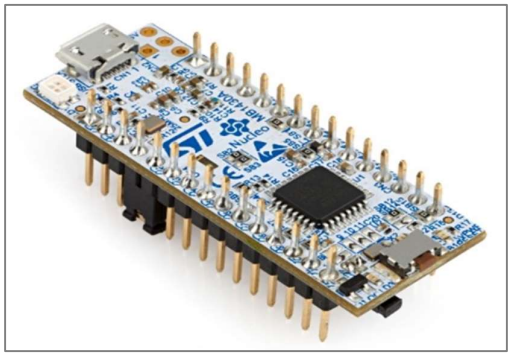


Abbildung 3-6: Entwicklungsboard STM32G431KB [2]

Das Entwicklungsboard besitzt 30 Anschluss-Pins auf jeweils zwei Anschlussleisten (CN3/CN4) verteilt. Die ausgewählten und verwendeten Pins mit Namen, Bezeichnung im Schaltplan, Funktion und Konfiguration in der Software sind in der Tabelle 3-1 dargestellt.

Tabelle 3-1: Verwendete PINs beim Entwicklungsboard STM32G431KB

Pin - NUCLEO	Pin-Name	Bezeichnung Schaltplan	Funktion	Konfiguration-Software
CN3 - 2	GND	GND	Masse	-
CN3 - 6	PA7 / A6	ADC-Input	Detektion Spannungsabfall	GPIO-Input
CN3 - 9	PA4 / A4	GPIO_D1	LED D1 (blau)	GPIO-Output
CN3 - 11	PA1 / A1	GPIO_D2	LED D2 (gelb)	GPIO-Output
CN3 - 12	PA0 / A0	GPIO_D3	LED D3 (grün)	GPIO-Output
CN3 - 14	3V3	3,3 V Input	Spannungsversorgung 3,3 V	VDD
CN4 - 1	PA9 / D1	USART1_TX	Kommunikation	USART1
CN4 - 2	PA10 / D0	USART1_RX		USART1
CN4 - 4	GND	GND	Masse	-
CN4 - 5	PA12 / D2	USART1_RTS	Kommunikation	USART1
CN4 - 6	PB0 / D3	IRQ	Interrupt-Routine	GPIO-Output
CN4 - 13	PA11 / D10	USART1_CTS	Kommunikation	USART1
CN4 - 14	PB5 / D11	Error 1	Fehlermeldungen von Bus	GPIO-Input
CN4 - 15	PB4 / D12	Error 2		GPIO-Input

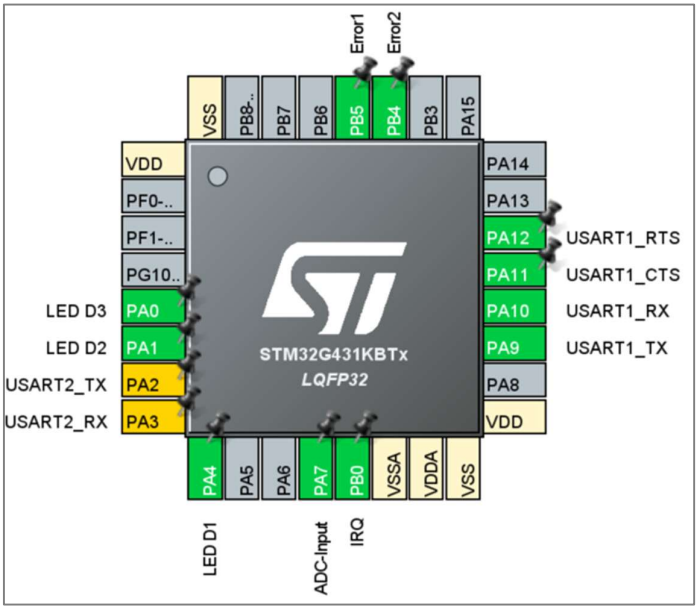


Abbildung 3-7: Konfiguration der PINs über Software STM32CubeIDE 1.5.0

Die Pins werden über die Software STM32CubeIDE (Version 1.5.0) konfiguriert und eingerichtet. Die Software ist ein All-in-One-Entwicklungstool mit fortschrittlicher C / C ++ - Entwicklungsplattform, Peripheriekonfiguration, Codegenerierung, Codekompilierung und Debugger-Funktionen für STM32-Mikrocontroller und Mikroprozessoren. In Abbildung 3-7 ist ein Ausschnitt zur Konfiguration des Microkontrollers dargestellt.

Ein weiterer wichtiger Hinweis zum Entwicklungsboard ist eine durchgeführte Modifizierung des Layouts auf der Oberseite (siehe Abbildung 3-8). Die Modifizierung beinhaltet die Entfernung der beiden Lötbrücken SB5 und SB15. Durch diese Modifizierung kann das Board nur noch mit einer Spannung von 3,3 V betrieben werden. Hintergrund ist, dass die interne Boardspannung bei 3,3 V liegt. Infolgedessen werden zwei Spannungswandler eingespart und eine damit verbundene geringere Stromaufnahme und Wärmeentwicklung gewährleistet.

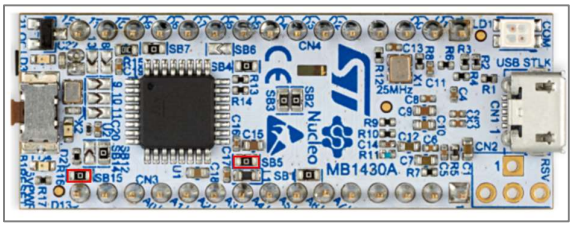


Abbildung 3-8: Modifiziertes Entwicklungsboard; rote Markierungen: Entfernte Lötbrücken SB15 (links) und SB5 (rechts) [2]

3.1.1.3 Weitere Bauteile

Diese Kapitel beinhaltet die Beschreibung aller weiteren Bauteile, welche zur vollständigen Benutzung der User-Unit-Platine beitragen. In der Anlage 6 ist die gesamte Bauteilliste dargestellt.

Auf der Platine befinden sich zwei Header-Anschlussleisten (J1/J2) zum Debuggen/Testen und zwei Anschlussleisten für die Flachbandkabel (J3/J4) zur festen Verdrahtung der User-Unit-Platine mit der Slave-Platine im Katamaran. In den folgenden Stichpunkten werden die Pins der Anschlussleisten beschrieben [6] mit Unterscheidung in Output (O) oder Input (I). In Abbildung 3-9 ist eine finale Übersicht der Platine dargestellt.

- 4-polige Anschlussleiste J1 und J3:
  - Pin 1 und 2: 12 V Spannungsversorgung (O)
  - Pin 3 und 4: GND (O)

- 12-polige Anschlussleiste J2 und J4:
  - Pin 1: 5 V Spannungsversorgung (O)
  - Pin 2: GND (O)
  - Pin 3: 3,3 V Spannungsversorgung (O)
  - Pin 4: GND (O)
  - Pin 5: VCC, Logikpegel für Slave-Platine (I)
  - Pin 6: RX, Daten vom Slave zum Bus (I)
  - Pin 7: TX, Daten vom Bus an Slave (O)
  - Pin 8: RTS, Slave möchte senden (request to send) (O)
  - Pin 9: CTS, warte auf Sendeerlaubnis (clear to send) (I)
  - Pin 10: IRQ, Interrupt Meldung vom Slave (I)
  - Pin 11: Fehlermeldung über Buszustand (O)
  - Pin 12: Fehlermeldung über Buszustand (O)

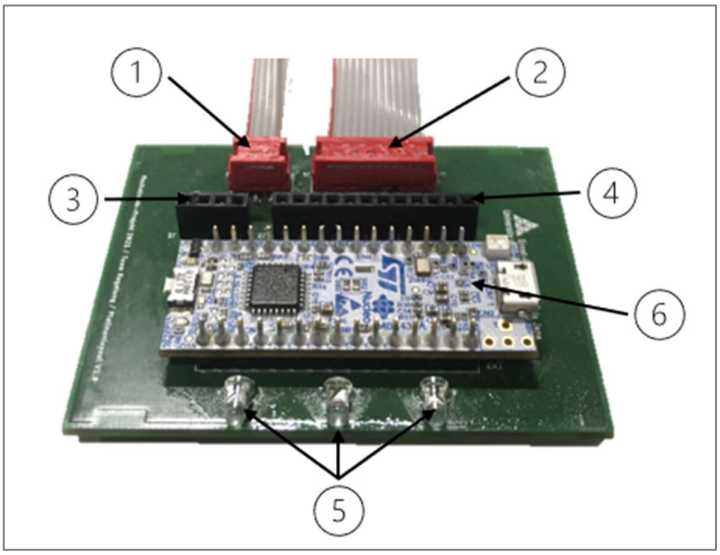


Abbildung 3-9: Übersicht finale Platine; 1: J3 mit 4-polig für Flachbandkabel; 2:J4 mit 12-polig für Flachbandkabel; 3: J1: 4-polig Headeranschlussleiste; 4: J2 12-polig Headeranschlussleiste; 5: LEDs; 6:NUCLEO-Board

Auf der Platine werden drei LEDs mit unterschiedlichen Farben verwendet (blau, gelb und grün). Diese LEDs sind dafür vorgesehen, um Debugg-Prozesse zu vereinfachen und

Zustände/Status des Boards anzuzeigen. Jede LED ist mit einem separaten Pin des NUCLEO-Boards verbunden (siehe Tabelle 3-1). Diese Pins sind als GPIO-Ausgänge (GPIO-Output) mit Pull-Down konfiguriert mittels Software (STM32CubeIDE 1.5.0). Betrieben werden die LEDs mit einem Vorwiderstand bei einer Eingangsspannung von 3,3 V und einer Lichtstärke von 1000 mcd. Der Vorwiderstand wird am Beispiel der LED D1 (blau) erläutert. Die Berechnungsschritte sind für die LED D2 (gelb) und LED D3 (grün) die gleichen, nur die Kennlinien (Lichtintensität, Strom) ändern sich:

1. Festlegung einer Lichtstärke, beispielsweise 1000 mcd
2. Ablesen des Stromes bei der gewählten Lichtintensität (siehe Abbildung -3-10)
3. Ablesen der Spannung bei von Punkt 2 abgelesenen Strom (siehe
4. Berechnung des Vorwiderstandes nach Formel (1)

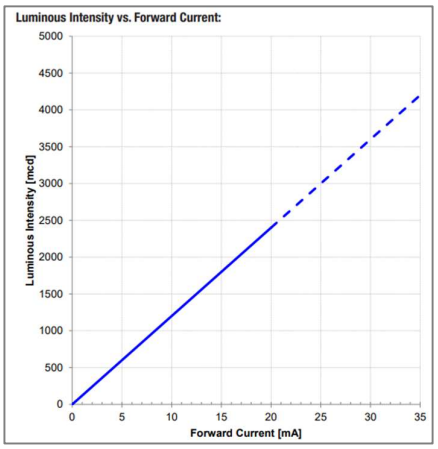


Abbildung -3-10: Lichtintensität in Abhängigkeit vom Strom; Beispiel: LED (blau); Bei Lichtintensität = 1000 mcd: ca. Strom I: 8 mA [3]

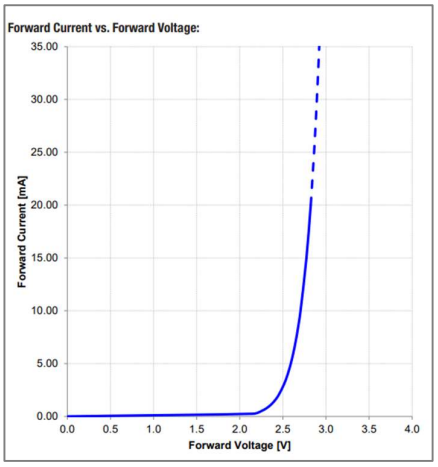


Abbildung 3-11: Strom in Abhängigkeit der Spannung; Beispiel: LED (blau); Strom = 7 mA; Spannung = 2,75 V [3]

<u>Berechnung Vorwiderstand am Beispiel LED D1 (blau):</u>

Eingangsspannung $U_E = 3{,}3\ V$

Durchlassspannung $U_D = 2{,}65\ V$

Durchlassstrom $I_D = 7\ mA$ bei 1000 mcd

Vorwiderstand $$R = \frac{(3{,}3\ V - 2{,}65\ V)}{8 \cdot 10^{-3} A} = 81{,}25\ \Omega \qquad (1)$$

Gewählter und nächst höherer Widerstand der Baureihe 0805:

$$R = 82\ \Omega$$

In Tabelle 3-2 ist eine Übersicht mit den Vorwiderständen der LEDs dargestellt.

Tabelle 3-2: Übersicht Vorwiderstände der LEDs

	LED D1 (blau)	LED D2 (gelb)	LED D3 (grün)
Eingangsspannung U_E in V	3,3		
Durchlassspannung U_D in V	2,65	1,95	2,5
Durchlassstrom I_D in mA	7	9	2
Vorwiderstand R in Ω	82	150	430

Das finale EAGLE-Boardlayout der Platine ist in Abbildung 3-12 dargestellt.

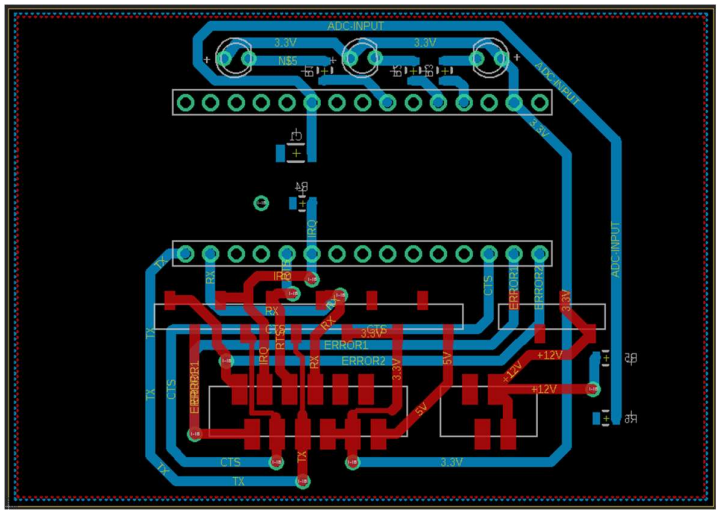


Abbildung 3-12: Finales Boardlayout

Ein Ausfall der Versorgungsspannung wird nicht sichergestellt. Eine Spannungsüberwachung wird stattdessen in die Schaltung integriert. Die Spannungsüberwachung ist über einen Spannungsteiler mit den Widerstand R5 (100 kΩ) und Widerstand R6 (33 kΩ) realisiert und wird über den ADC-Input (CN3/Pin 6) auf das NUCLEO geführt. Die Eingangsspannung (12 V) wird von der Slave-Platine übergeben und liegt an J3 (Pin 1/2) und J1 (Pin 1/2) an der User-Unit-Platine an. Die Voraussetzung ist, dass ein Spannungsabfall der Eingangsspannung um 10 % detektiert werden muss.

3.1.2 Software

Dieser Abschnitt behandelt die Kommunikation zwischen der User-Unit und dem USV-Slave, welcher über das RS232-Bussystem erfolgt. Durch die Verwendung des Nucleo-32-Boards STM32G4 wird das Programm STM32CubeIDE als Entwicklungsumgebung genutzt.

Zuerst wird die Anwenderschnittstelle vorgestellt, welches für die Kommunikation die UART-Schnittstelle darstellt. Anschließend werden die erstellten Funktionen und die Buskommunikation anhand von Programmablaufplänen erläutert.

3.1.2.1 Anwenderschnittstelle

Für das Schreiben und Lesen von Daten auf dem RS232-Bus wird eine Datenschnittstelle und ein Übertragungsprotokoll benötigt. [1] Als Datenschnittstelle wird die UART-Schnittstelle verwendet, da diese in den meisten Mikrocontrollern implementiert ist. [1] Nachfolgend wird die UART-Schnittstelle allgemein erklärt. Anschließend werden die verwendeten UART-Schnittstellen des Nucleo-Boards vorgestellt und deren Aufgaben beschrieben.

3.1.2.1.1 UART-Schnittstelle

Die UART-Schnittstelle ist eine Kommunikationsschnittstelle zwischen zwei Geräten und wird als Teil der RS-232-Standards verstanden. Die UART-Schnittstelle verwendet zum Senden (TX) und Empfangen (RX) von Daten je ein Datensignal. Für die Datenflusskontrolle können zwei weitere Steuerleitungen verwendet werden. Eine Steuerleitung dient zum Empfangen der Sendeerlaubnis (CTS), sodass wieder gesendet werden darf, und eine Steuerleitung dient zum Senden der Empfangserlaubnis (RTR/RTS), dass das Gerät wieder bereit ist, Daten zu empfangen. Die Schnittstelle kann im Vollduplexbetrieb verwendet werden, wodurch beide Geräte gleichzeitig senden und empfangen können. [1]

Die UART-Konfiguration und der Frameaufbau für eine Übertragung sind in dem Dokument „USV_Slave_Anwender_Schnittstelle.pdf“ beschrieben.

Für das Projekt werden beide zur Verfügung stehende Board-UARTs verwendet, welche im folgenden Kapitel vorgestellt werden.

3.1.2.1.2 UART1 und UART2

Die Auswahl und Einstellung der UARTs erfolgt im *Device Configuration Tool* der STM32CubeIDE. Die Pin-Konfiguration wird visuell dargestellt (Abbildung 3-7). Eine Änderung ist auch in der main.c-Datei möglich. Die UART-Konfiguration, wie z.B. Baudrate und Parität, wird vom Dokument „Anwenderschnittstelle USV-Slave“ festgelegt.

Die UART1-Schnittstelle wird zur Kommunikation zwischen dem USV-Slave und der User-Unit verwendet. Die Schnittstelle ist über die entwickelte Platine (siehe Anlage 5) mit dem USV-Slave verbunden. Die Datenflusskontrolle wird als Hardwarehandshake im Programm festgelegt.

Die UART2-Schnittelle ist auf dem Board mit dem „USB Micro-B connector CN1“ verbunden, wodurch eine Kommunikation zwischen dem Board und einem PC-Terminal möglich ist. Über diese Verbindung ist das Flashen und Debuggen eines Programmes auf dem Board möglich.

Nach der Aufgabenbeschreibung der verwendeten UARTs wird die Programmierung der Kommunikation näher erläutert.

3.1.2.2 communication.c und communication.h

Für die Kommunikation werden zwei Dateien zum Abrufen und Schreiben von Daten von und auf dem Bus erstellt. Die Codes sind in den Anlagen hinterlegt.

Die erste Datei communication.h wird als Header-Datei verwendet und beinhaltet alle Konstanten für den Frameaufbau, Variablen für das Datenabbild des Busses, Adresswerte der Busdaten (gelistet im Dokument „Bussystem.pdf“), Statuswerte für die Übertragung und die Funktionsprototypen.

Die zweite Datei communication.c dient zur Ausführung der Funktionen und Verwendung der Variablen und Konstanten, die in der communication.h deklariert sind.

Im folgenden Unterkapitel werden die erstellten Funktionen vorgestellt. Anschließend werden Programmablaufpläne für das Senden und Empfangen von Daten vorgestellt.

3.1.2.2.1 Funktionen

Nachfolgend werden alle Funktionen für die Buskommunikation vorgestellt.

3.1.2.2.1.1 Funktion InitCom

Die Funktion *InitCom* dient zur einmaligen Übergabe des UART1-Handles in das Modul *communication.c*. Dies muss durchgeführt werden, da sonst die UART-Funktionen für das Senden und Empfangen von Daten nicht verwendet werden können.

3.1.2.2.1.2 Funktion updateSensorBlock

Die Funktion *updateSensorBlock* aktualisiert und verarbeitet den gesamten Sensor-Block, der vom Bus gelesen werden kann. Zu den Sensorblock zählen u.a. die GPS-Koordinate (Längen- und Breitengrad) und die GPS-Geschwindigkeitsangabe. Zuerst werden die Adresse des ersten Sensors und die Gesamtzahl der zu lesenden Bytes an die aufgerufene Funktion *readData* übergeben werden. Die Funktion *readData* liest die Daten vom Bus und gibt diese als Fixpunkt-Daten zurück. Die erhaltenen Bytes werden über die externe Funktion *convFixedToF32* in das Float32-Format umgewandelt und anschließend den entsprechenden Sensorgrößen zugeordnet. Neben den Sensorgrößen wird auch ein Zeitstempel gelesen, welcher in Sekunde, Minute und Stunde konvertiert wird.

3.1.2.2.1.3 Funktion updateFuehrungsblock

Die Funktion *updateFuehrungsblock* aktualisiert und verwertet den gesamten Führungsgrößenblock, welcher vom Bus gelesen werden kann. Das Prinzip der Funktion ist simultan der Funktion *updateSensorBlock*. Zu den Führungsgrößen zählen die Längen- und Breitenangaben der Punkte A und B, die Sollgeschwindigkeit zwischen den Punkten A und B und der Epsilonwert für den Epsilonschlauch.

3.1.2.2.1.4 Funktion updateLocalData

Die Funktion *updateLocalData* aktualisiert und verarbeitet alle gelesenen Sensorwerte einzeln. Die Funktion ist aus einer früheren Entwicklungsversion entstanden und dient als Backup.

3.1.2.2.1.5 Funktion readData

Für das Anfordern und Speichern von Daten vom Bus wird die Funktion *readData* verwendet und von den Funktion *updateSensorBlock*, *updateFuehrungsblock* und

updateLocalData aufgerufen. Gemäß dem Frame für das Lesen von Daten wird zuerst ein Sendeframe an den Slave gesendet, um mitzuteilen, dass Daten gelesen werden sollen. [1] Der Sendeframe enthält u.a. die zu lesende Adresse, die Anzahl der Bytes und eine Checksumme, welche durch die Funktion *CRC8* erstellt wird. Sind alle Daten fehlerfrei übertragen, die Slave-ID korrekt und Daten an der angeforderten Adresse, antwortet der Slave mit den gewünschten Daten. Ist dies nicht der Fall, antwortet der Slave mit einem NACK („not acknowledge“) und geht in Busruhe. [1] Sind die Daten korrekt, werden die vom Slave empfangenen Daten in ein Buffer gespeichert und auf Korrektheit überprüft. Ergab die Auswertung der gesendeten Daten und der empfangenen Checksumme, dass der Empfangsframe fehlerfrei vom Slave empfangen wurde, werden die Bytes an die Funktionen zurückgegeben, welche *readData* aufgerufen hat.

3.1.2.2.1.6 Funktion updateStellgrBlock

Die Funktion *updateStellgrBlock* aktualisiert bzw. schreibt die aktuellen Daten aus dem Stellgrößen-Block auf den Bus bzw. zum Slave. Zu den Stellgrößen zählen die Stellwerte für den Schub und den Ruderausschlag sowie das Error Status Byte. Zuerst werden die Daten mittels der externen Funktion *convF32ToFixed* vom Float-Format in das Fixpunktformat umgewandelt. Danach werden die umgewandelten Daten als Datenblock an die Funktion *sendData* übergeben, welche die Daten an den Slave überträgt.

3.1.2.2.1.7 Funktion updateBusData

Für das separate Umwandeln und Schreiben der Daten wird die Funktion *updateBusData* verwendet. Hier werden alle Daten übersetzt und einzeln an die Funktion *sendData* übergeben. Die Funktion *updateBusData* dient als Backup der Funktion *updateStellgrBlock*.

3.1.2.2.1.8 Funktion sendData

Die Funktion *sendData* erstellt einen Sendebuffer gemäß dem Frame für das Schreiben von Daten an den Slave. [1] Der Sendebuffer enthält das Startbyte, die Slave-ID, die Adresse, die Datenbytes, die Checksumme (gebildet durch die Funktion *CRC8*) und das Endbyte. Anschließend wird der Buffer über UART versendet. Werden alle Daten richtig empfangen und dürfen an der gewünschten Adresse die Daten überschrieben werden, antwortet der Slave mit einem ACK („acknowledge“), sonst mit einem NACK („not acknowledge“). [1] Bei einem NACK wird eine Fehlermeldung an die aufrufende Funktion zurückgegeben.

3.1.2.2.1.9 Funktion CRC8

Die Funktion *CRC8* erstellt eine Checksumme aus den relevanten Datenbytes, die empfangen oder gesendet werden. Hierfür wird das Generatorpolynom „0xD5“ verwendet. Die erstellte Checksumme wird an die Funktion, von welcher sie aufgerufen wurde, zurückgegeben. Sie dient unter anderem dazu, die empfangene Checksumme auf Übertragungsfehler zu überprüfen. Ergibt ein Vergleich der empfangenen und der gebildeten Checksumme, dass diese unterschiedlich sind, ist ein Übertragungsfehler der Daten aufgetreten.

3.1.2.2.2 Codeablauf

Zunächst wird der Ablauf des Sendens von Daten auf den Bus bzw. an die Slave-Einheit erläutert, danach folgt der Ablauf des Lesens von Daten auf dem Bus.

3.1.2.2.2.1 Daten schreiben

Das Schreiben bzw. Senden von Daten an den Slave wird durch ein Programmablaufplan in den Anlagen visuell dargestellt.

Über einen Aufruf der Funktion *updateStellgrBlock* werden alle Daten über die externe Funktion *convF32ToFixed* von einem Floatformat in ein Fixpunktformat umgewandelt. Die umgewandelten Dateien werden in das Array *tmp* gespeichert und mitsamt der ersten Adresse des Blocks an die Funktion *sendData* weitergeleitet.

Die Funktion *sendData* erstellt eine Datenbuffer, welches u.a. die umgewandelten Daten und die Checksumme der Daten beinhaltet. Tritt beim Versenden des Datenbuffers ein Fehler auf, wird dieser an die Funktion *updateStellgrBlock* zurückgegeben. Ist das Versenden des Datenbuffers an den Slave erfolgreich verlaufen, wird ein Empfangsbuffer erstellt. In den Empfangsbuffer wird die Antwort vom Slave gespeichert, welche der Slave nach dem Erhalt des Datenbuffers an die User-Unit versendet. Die Antwort vom Slave ist entweder ACK („acknowledge“) oder NACK („not acknowledge“). ACK bedeutet, dass die Nachricht fehlerfrei vom Slave empfangen wurde und die angegebene Adresse überschrieben werden darf. NACK heißt, dass die Nachricht nicht fehlerfrei war, die angegebene Adresse nicht überschrieben werden darf oder beides. Die Antwort vom Slave (ACK, NACK) wird als Status an die Funktion *updateStellgrBlock* zurückgegeben und die Funktion *sendData* ist beendet.

Die von der Funktion *sendData* zurückgegebene Status gibt die Funktion *updateStellgrBlock* wiederum zurück.

3.1.2.2.2.2 Daten lesen

Das Empfangen von Daten vom Slave wird durch ein Programmablaufplan in den Anlagen visuell dargestellt.

Über einen Aufruf der Funktionen *updateSensorBlock* und *updateFuehrungsgrBlock* sollen alle Sensordaten respektive Führungsgrößendaten als Block vom Slave empfangen werden. Dazu wird in jeder Funktion die Funktion *readData* aufgerufen. *readData* wird die erste Adresse des Blocks, die Anzahl der zu empfangenen Bytes und der Zeiger für das Datenbuffer *tmp* mitgegeben, in welcher die Bytes gespeichert werden sollen.

In der Funktion *readData* wird zunächste ein Sendebuffer erstellt. Dies dient dazu, dem Slave mitzuteilen, ab welcher Adresse wie viele Bytes an die User-Unit gesendet werden sollen. Treten beim Senden Fehler oder ein Timeout auf, sendet der Slave ein NACK („not acknowledge“) an die User-Unit zurück und geht in Busruhe. [1] Ist das Senden fehlerfrei und ohne Timeout durchgeführt worden, sendet der Slave die geforderten Daten, welche in ein Empfangsbuffer übertragen werden. Die relevanten Daten aus dem Empfangsbyte werden in das Datenbuffer *pDB* extrahiert. Anschließend wird der Empfangsbuffer ausgewertet und auf Korrektheit überprüft. Wurden die Daten fehlerfrei vom Slave übertragen, wird ein entsprechender Status an die aufrufende Funktion zurückgegeben und der Durchlauf der Funktion *readData* ist beendet.

Die aufrufenden Funktionen *updateSensorBlock* und *updateFuehrungsgrBlock* erhalten die relevanten Daten und den Status von der Funktion *readData* zurück. Ergibt die Abfrage des erhaltenen Status, dass die Kommunikation korrekt abgelaufen ist, werden die Daten vom Fixpunktformat in ein Floatformat konvertiert. Die konvertierten Daten werden mitsamt des Kommunikationsstatus an die Stelle, wo die Funktionen aufgerufen werden, zurückgegeben. Ergab die Statusabfrage, dass während der Kommunikation Fehler aufgetreten sind oder ein NACK vom Slave gesendet wurden, dann wird der entsprechende Status ohne die relevanten Daten zurückgegeben.

Zum Zeitpunkt der Abgabe wurde keine ID für den USV-Slave bereitgestellt. Dieser muss in der Datei *communication.h* ergänzt werden.

3.1.2.3 Aktuelle Softwareprobleme

Bei der Funktionsprüfung der Kommunikation zwischen der User-Unit und dem USV-Slave im Mechatroniklabor der Ernst-Abbe-Hochschule Jena wird beobachtet, dass sich die Kommunikation nach einer kurzen Zeitspanne aufhängt und nur noch mittels Reset

neu gestartet werden kann. Die Ursache wird an einem Flag vermutet, welches nach einer Kommunikation gesetzt und anschließend nicht mehr zurückgesetzt wird. Eine mögliche Behebung ist schon geplant, muss aber noch getestet werden. Der Test für die Behebung des Problems ist auf den 06.04.2021, also nach der Abgabe dieses Berichtes, terminiert.

3.2 Regelstrategie

In diesem Kapitel wird auf die Regelstrategie eingegangen. Hierbei wird zuerst das Ziel der Regelstrategie erläutert und im weiteren Verlauf beispielhaft in schrittweisen Überlegungen erläutert. Es wird auf die Umsetzung in dem daraus erzeugten Code eingegangen und die vorher durchgeführten Simulationen erklärt.

3.2.1 Was ist das Ziel der Regelstrategie?

Für das Steuern des Bootes wird die Route von einer Bahnsteuerung vorgegeben. Hierbei wird ein Punkt B vorgegeben. In Abbildung 3-13 ist eine schematische Darstellung aufgezeigt. Das Boot befindet sich in Punkt A und soll auf der hier gestrichelt dargestellten Route zum Zielpunkt B fahren. Hat das Boot Punkt B erreicht wird ein neuer Punkt B vorgegeben und das Boot fährt zu diesem Punkt. Zusätzlich zum Zielpunkt wird von der Bahnsteuerung die Geschwindigkeit vorgegeben.

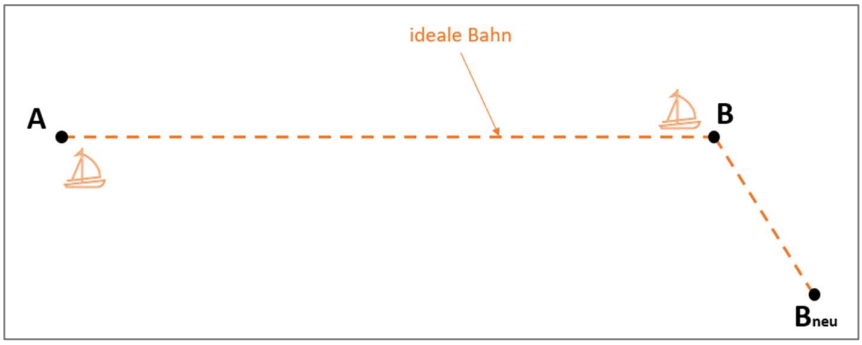


Abbildung 3-13: ideale Bahn

Weicht das Boot aufgrund von Strömungen, Wind oder anderen Einflüssen von der idealen Route ab soll auf diese geregelt werden. Hierfür wird eine Regelungsstrategie benötigt. Ein Beispielhafter Fahrtverlauf ist in Abbildung 3-14 abgebildet. Die gestrichelte Linie visualisiert die ideale Route und mit blau ist die beispielhafte Fahrtroute eingezeichnet. Das Ziel ist es über eine Regelung des Ruders sowie des Motorschubs die vorgegebene ideale Bahn zu verfolgen.

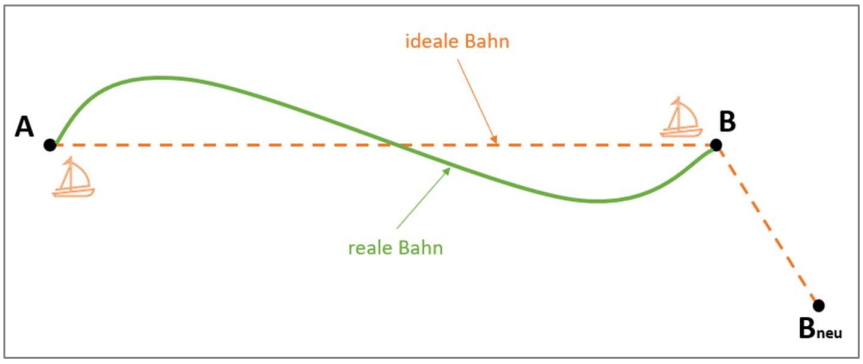


Abbildung 3-14: Fahrtverlauf ideale und reale Bahn

Um dieses Ziel umzusetzen ist auf dem Boot ein GPS angebracht welcher Koordinaten liefert. Zusätzlich zur Bootsposition wird der Winkel mit Hilfe eines Kompasses bestimmt. Die Geschwindigkeit wird ebenfalls mithilfe des GPS ermittelt. Der Endpunkt der Route wird von der Bahnsteuerung übermittelt außerdem wird ein Toleranzband (Epsilon Schlauch) vorgegeben. Für die weiteren Überlegungen wird das Boot als Vektor betrachtet welcher die Eigenschaften Position (GPS) und Winkel (Kompass) besitzt. Mithilfe der Stellgröße Schub wird auf eine vorgegebene Geschwindigkeit geregelt. Mit der Stellgröße Ruder wird die Peilung eingestellt. Die Peilung wird stets aus der aktuellen Bootsposition und dem Zielpunkt ermittelt. Dieser bleibt steht der von der Bahnsteuerung vorgegebene Punkt B. Mit Bahnkorrektur wird in Abhängigkeit zur Abweichung zur idealen Bahn ein Zwischenpunkt (Heading Point) errechnet, welcher dazu dient, die Peilung so einzustellen, dass die Abweichung verringert wird.

3.2.2 Fahrt von A nach B

In Abbildung 3-15 ist die Startposition aufgezeigt. Das Boot befindet sich im Startpunkt A und soll auf der idealen Strecke AB zum Endpunkt B fahren. Um Punkt B ist ein Toleranzgebiet mit Radius r eingezeichnet. Erreicht das Schiff diesen Bereich gilt der Punkt B als angefahren. B’ wird der neue Startpunkt A für den nächsten Endpunkt B. Der Punkt B wird von der Bahnsteuerung aktualisiert. Diese Methode wird angewandt, da der Punkt B nur sehr schwer erreicht werden kann. Die Geschwindigkeit müsste am Ende stark reduziert werden, da man ansonsten beispielsweise am Ziel vorbeifährt und wenden müsste. Da man einen durchgängigen Fahrverlauf erreichen möchte wird daher die Methode des Toleranzgebietes gewählt.

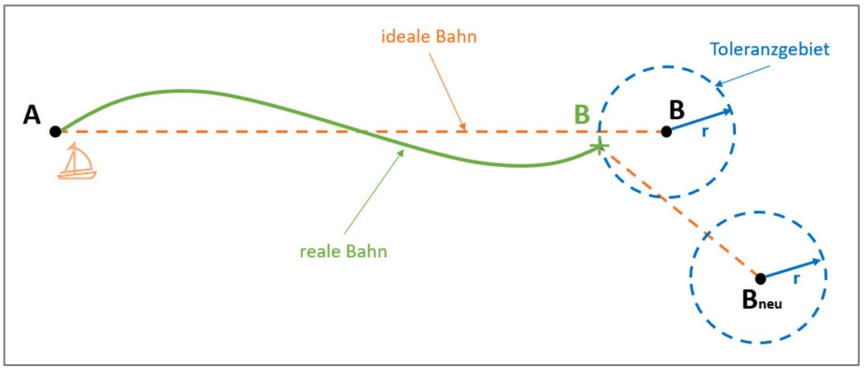


Abbildung 3-15: Fahrt von A nach B - Toleranzgebiete

Zusätzlich zum Toleranzgebiet um den Zielpunkt wird ein Epsilonschlauch eingeführt. Also ein Toleranzgebiet um die ideale Bahn bei dem die ideale Bahn als erreicht betrachtet wird. Umso näher man dem Zielpunkt B kommt, desto kleiner wird dieses Toleranzgebiet, da man diesen Punkt erreichen muss. Mithilfe des Toleranzbandes kann gut zwischen unterschiedlichen Fahrszenarien wie beispielsweise die Fahrt auf hoher See (großer Epsilonschlauch → geringe Genauigkeit) oder der Fahrt in den Hafen (kleiner Epsilonschlauch → große Genauigkeit) variiert werden. In Abbildung 3-16 ist beispielhaft das Toleranzband mir gestrichelten Linien eingezeichnet.

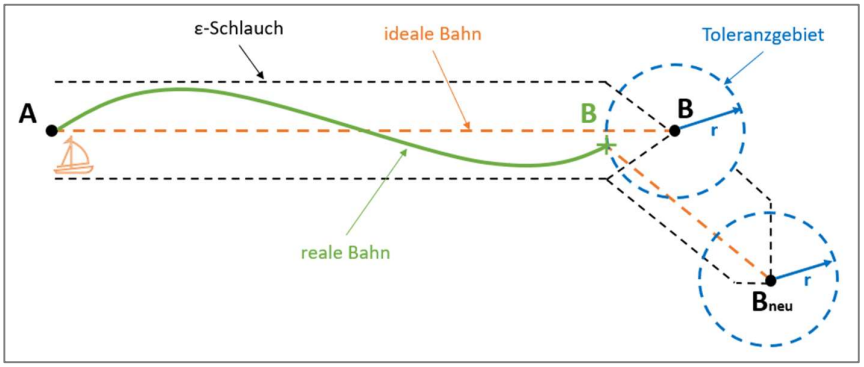


Abbildung 3-16: Fahrt von A nach B - Epsilonschlauch

Es wurde die in Abbildung 3-17 dargestellte Situation überdacht. Was passiert, wenn das Boot sehr weit von der idealen Bahn abgedriftet ist? Wie aus der schematischen Darstellung ersichtlich fährt man annähernd die gesamte Strecke außerhalb des vorgegebenen Toleranzfeldes. Um dies zu verhindern wurde die Methode “Heading Point” eingeführt. Diese wird im weiteren Verlauf erläutert.

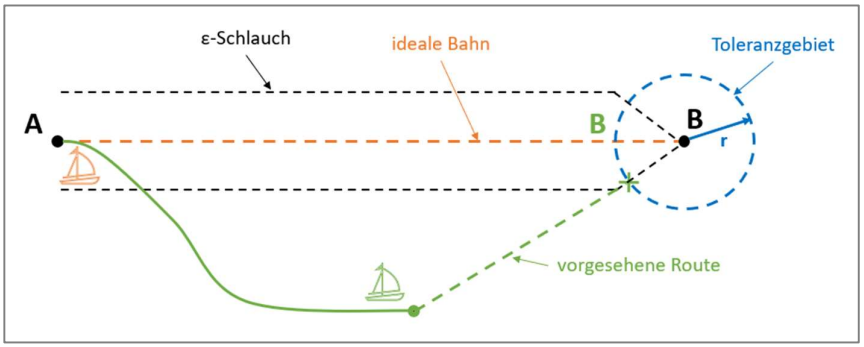


Abbildung 3-17: Verlassen des Epsiolonschlauches

3.2.3 Methode Heading Point

Mithilfe dieser Methode soll vermieden werden, dass das Boot zu viel Strecke außerhalb des Toleranzgebiets zurücklegt. In Abbildung 3-18 ist das Verhalten beispielhaft aufgezeigt. Das Boot ist von seiner idealen Route abgewichen und möchte zu Punkt B fahren. Anstatt direkt zu Punkt B zu fahren, wird ein Zwischenpunkt, der sogenannte Heading Point, als Zielwert bestimmt. Somit legt das Boot weniger Strecke außerhalb des Toleranzgebiets zurück. In Abbildung 3-18 ist der Startpunkt A, Endpunkt B, die ideale Route, die reale Route sowie der Heading Point HP1 und HP2 abgebildet. Zusätzlich ist der Hilfspunkt C eingezeichnet. Ist die Abweichung groß kann der Heading Point nahe an den Punkt auf der idealen Bahn C gesetzt werden (HP1) um schnell in den zugelassenen Bereich zurückzukehren. Beispielhaft ist der etwas weiter entfernte Punkt HP2 eingetragen.

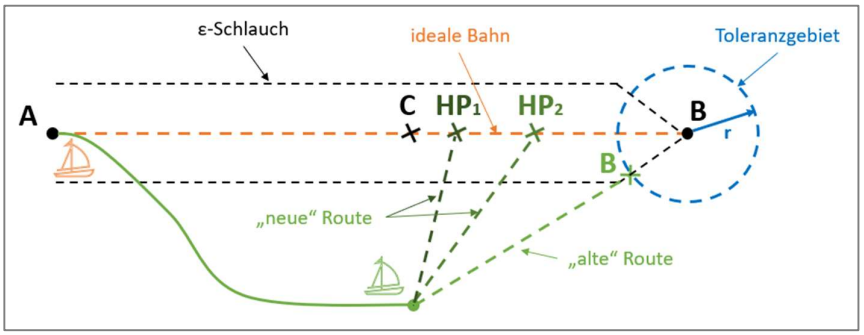


Abbildung 3-18: Heading Point-Methode

Aufgrund des hier sichtbaren Verhaltens wird der Heading Point in Abhängigkeit des Fehlers gesetzt. Ist der Fehler, also die Distanz Boot - ideale Bahn, groß, wird der Heading Point in geringer Distanz vor den Punkt C gesetzt. Ist der Fehler klein, wird der Punkt weiter nach vorn gesetzt. Die Fehlerabhängigkeit ist schematisch in der folgenden Abbildung 3-19 aufgetragen.

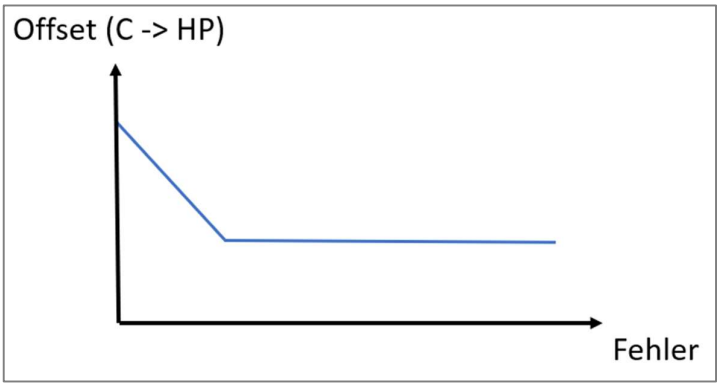


Abbildung 3-19: Fehlerabhängigkeit (Distanz: Boot - ideale Bahn)

In weiteren Arbeiten muss darüber nachgedacht werden die Abhängigkeit des Fehlers in Bezug auf die Geschwindigkeit zu betrachten. Dadurch wird der HP in Abhängigkeit der unterschiedlichen Fahrszenarien betrachtet (Fahrt im Hafen, Fahrt auf hoher See). Außerdem muss überprüft werden wie genau die vom GPS und Kompass gelieferten Ergebnisse sind.

In den folgenden Abbildungen ist die Methode Heading Point schematisch zusammengefasst dargestellt. Das Boot wird hierbei durch einen Pfeil beschrieben. Ein Pfeil entspricht einem Berechnungszyklus.

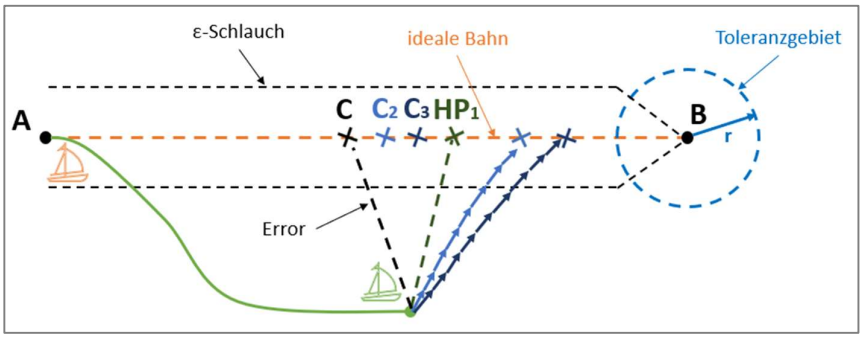


Abbildung 3-20: Zusammenfassung Heading Point Methode Teil 1

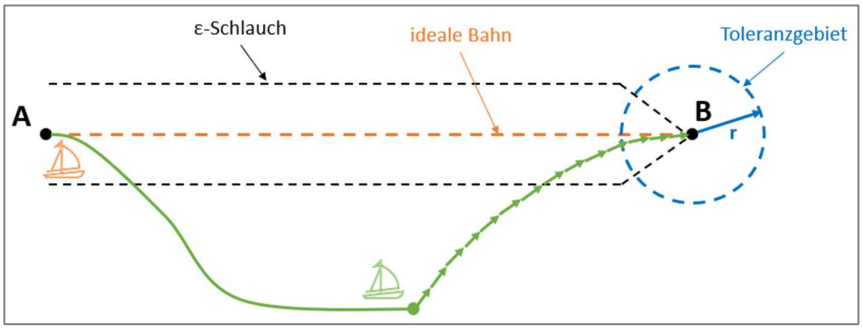


Abbildung 3-21: Zusammenfassung Heading Point Methode Teil 2

Mithilfe der hier beschriebenen Regelstrategie wurde der im weiteren Verlauf beschriebene Programmcode entwickelt.

3.2.4 Implementation im Quellcode

Der PID-Regelalgorithmus ist im Modul „pid_controller“ umgesetzt.

Der Funktion `pid_controll` wird hierfür ein Verweis auf eine zuvor erstellte Instanz einer `controller`-Konfiguration, der Sollwert und der Istwert übergeben. Als Rückgabe erfolgt die ermittelte Stellgröße.

In der `controller`-Konfiguration (struct) befinden sich die Parameter und zwischengespeicherten Werte (Integrations-Summe, vorherige Regelabweichung, …) des jeweiligen Reglers. In `main` sind zwei Instanzen angelegt. Eine Instanz für die Schubregelung und eine für die Ruderregelung.Weiterhin finden sich notwendige Berechnungen im Modul `calculation`. Dazu gehört die Bestimmung:

- des Abstandes zwischen zwei Koordinaten
- der Peilung über zwei Koordinaten
- des Targetpoints aus den Koordinaten des Punkt A, Punkt B und des Bootes

4 Ausblick

Folgende Themen sind aktuell in Bearbeitung und sollten in der nachgereichten Version des Quellcodes fertiggestellt sein:

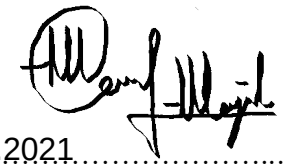
- Implementation Watchdog
- Detektion Brownout (als Reset-Ursache)
- Bugfix für Nichtwiederaufnahme der Kommunikation nach fehlerhafter Übertragung

Die Ermittlung der exakten Regelparametern am finalen bzw. realen Bootmodell ist noch offen.

Folgende Erweiterungen sind denkbar:

- Hinzufügen von Fehlerbehandlung
- Auslösen von Fehlermeldungen
- Offset für Targetpoint in Abhängigkeit der Geschwindigkeit berechnen

5 Entlastung Projektteam



31.03.2021 …….. ……..……

Datum, Unterschrift (Projektleiter) *Unterschrift* (Projektauftraggeber)

6 Literaturverzeichnis

[1] S. F. Jörg Grabow, „Anwenderschnittstelle USV-Slave,“ Jena, 2021.

[2] J. Grabow, „Projektbeschreibung/Themenstellung,“ 04 11 2020. [Online]. Available: https://moodle.stud.eah-jena.de/pluginfile.php/133459/mod_resource/content/3/USV.pdf.

[3] STMicroelectronics, „STMicroelectronics,“ 01 2021. [Online]. Available: https://www.st.com/resource/en/user_manual/dm00493601-stm32g4-nucleo32-board-mb1430-stmicroelectronics.pdf. [Zugriff am 20 03 2021].

[4] W. Electronic, „Würth Electronic,“ 01 2021. [Online]. Available: https://www.we-online.de/katalog/datasheet/151034BS03000.pdf. [Zugriff am 20 03 2021].

Anlagen

Anlagen 28

Anlage 1: Terminplan 29

Anlage 2: Meilensteinplan 30

Anlage 3: Anforderungsliste 31

Anlage 5: Schaltplan (EAGLE) 32

Anlage 6: Boardlayout - Top und Ratsnest (EAGLE) 33

Anlage 7: Keep-Out Area (Top / Bottom) EAGLE-Vorlage 34

Anlage 8: EAGLE Hinweise 35

Anlage 9: Bauteilliste 37

Anlage 10: Slave und User-Unit mit Gehäuse 38

Anlage 11: Buskommunikation - Daten senden 39

Anlage 12: Buskommunikation - Daten lesen 40

Anlage 13: Projektstrukturplan - Kommunikation 41

Anlage 14: Risikomanagement 43

Anlage 15: Sitzungsprotokolle 48

Anlage 1: Terminplan

Balkenplan zur Aufgaben- und Terminplanung

Projekttitel	Unmanned Surface Vehicle
Projektleiter	Muhammad Maaz Majid
Datum	21.11.2020

PSP-Code	Teilaufgabe / Arbeitspaket	Termine		Personaleinsatzplan		2020						2021											
		Start	Ende	Verantwortlicher	Mitarbeiter	47	48	49	50	51	52	1	2	3	4	5	6	7	8	9	10	11	12
1	**Platine Entwicklung und Buskommunikation**	**KW47**	**KW50**	**Team Kommunikation**																			
1.1	Entwicklung der Platine																						
1.2	Rechere und Inbetriebnhame der Buskommunikation																						
2.	**Regel Strategie**	**KW47**	**KW50**	**Team Regelung**																			
2.1	Modell erstellen für Regler und Strecke (Schub (Geschwindigkeit), Ruder (Kurswinkel)) - Vereinfachte Darstellung																						
2.2	In Matlab simulieren und Reglerparameter bestimmen																						
3.	**Programmierung**	**KW49**	**KW4**	**Gemeinsam**																			
3.1	grobe Programmtruktur/ groben Aufbau festlegen			Martin																			
3.2	Berechnung Entfernung (einfach, verbessert)																						
3.3	Berechnung Winkel (einfach, verbessert)																						
3.4	Berechnung Geschwindigkeit (gleitender Mittelwert)																						
3.5	Implementierung Aufgabe 4			Team Kommunikation																			
3.6	Implementierung Buskommunikation																						
3.7	PI-Regler Implementierung																						
4.	**Kursabweichungskorrekturstrategie**	**KW1**	**KW4**	**Gemeinsam**																			
4.1	Kursabweichungskorrekturstrategie festlegen																						
4.2	Kursabweichungskorrekturberechnung																						
5.	**Dokumentation**	**KW7**	**KW9**																				
5.1	Bearbeitung der Dokumentation																						

Bearbeitungshinweise:

1. Teilaufgaben und Arbeitspakete einfügen
2. Optional: Personaleinsatzplan mit Verantwortlichem und Mitarbeitern pro Arbeitspaket
3. Bearbeitungsdauern pro Arbeitspaket eintragen
4. Meilensteine eintragen
5. Soll-Termine eintragen
6. Zum Drucken Fertigstellung und Personaleinsatzplan und Termine ausblenden oder Druckansicht anpassen (bei Seitenansicht oder Druckbereich einstellen)

	Geplante Zeitraum
x	Bearbeitung
!!!!	Überschrittung
	PüfferZeit

Anlage 2: Meilensteinplan

Meilensteinplan

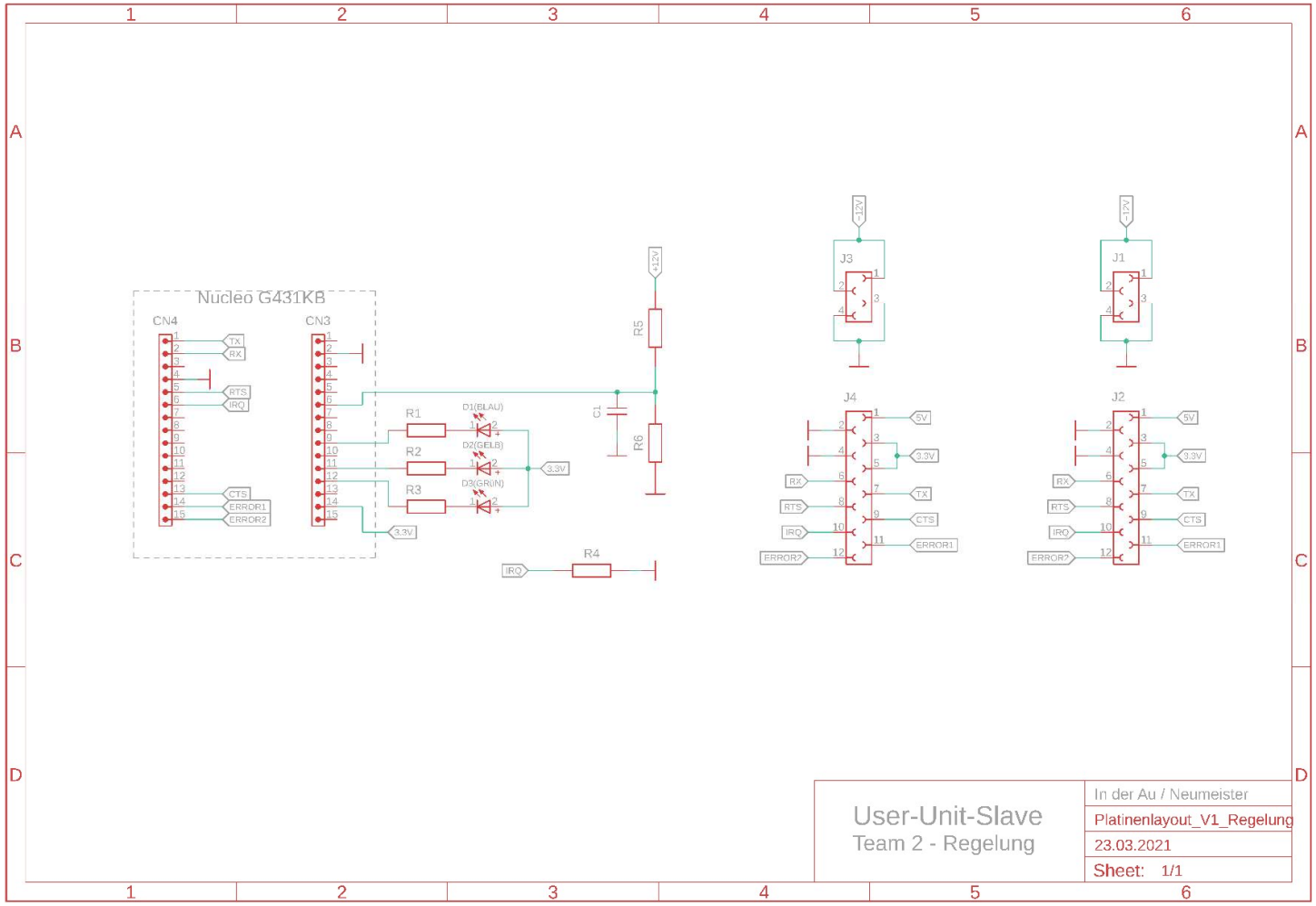
Projekttitel:	Mechatronikprojekt 2020
Projektnummer:	
Projektleiter:	Muhammad Maaz Majid
Aktuelles Datum:	23.03.2021

Nr.	Meilenstein	Termin (Soll)	Termin (Ist)
MS 1	Projektstart → Projektauftrag	04.11.2020	04.11.2020
MS 2	Projektplan	20.11.2020	20.11.2020
MS 3	Reglung- und Steuerstrategie festlegen	25.11.2020	01.12.2020
MS 4	• Erster Entwurf Regler Simulation (mit Statusbericht) und MS-Plan vollständigen (am 16. MS Plan Prof. Grabow präsentieren) • Entwurf der Platine für Gehäuse fertigen und (wenn möglich) bestellen	2.12.2020	25.11.2020 09.12.2020 (Platine)
MS 5	Vorläufige Platine fertigen	17.12.2020	17.12.2020
MS 6	Erster Entwurf des (gesamt) Programms	10.02.2021	30.01.2021
MS 7	Anfang der Testphase für die Software	15.02.2021	1.03.2021
MS 8	Risikoanalyse und Dokumentation vervollständigen	23.03.2021	21.03.2021
MS 9	Projektziele erreicht	23.03.2021	26.03.2021
MS 10	Projektende → Projektorganisation aufgelöst	26.03.2021	26.03.2021

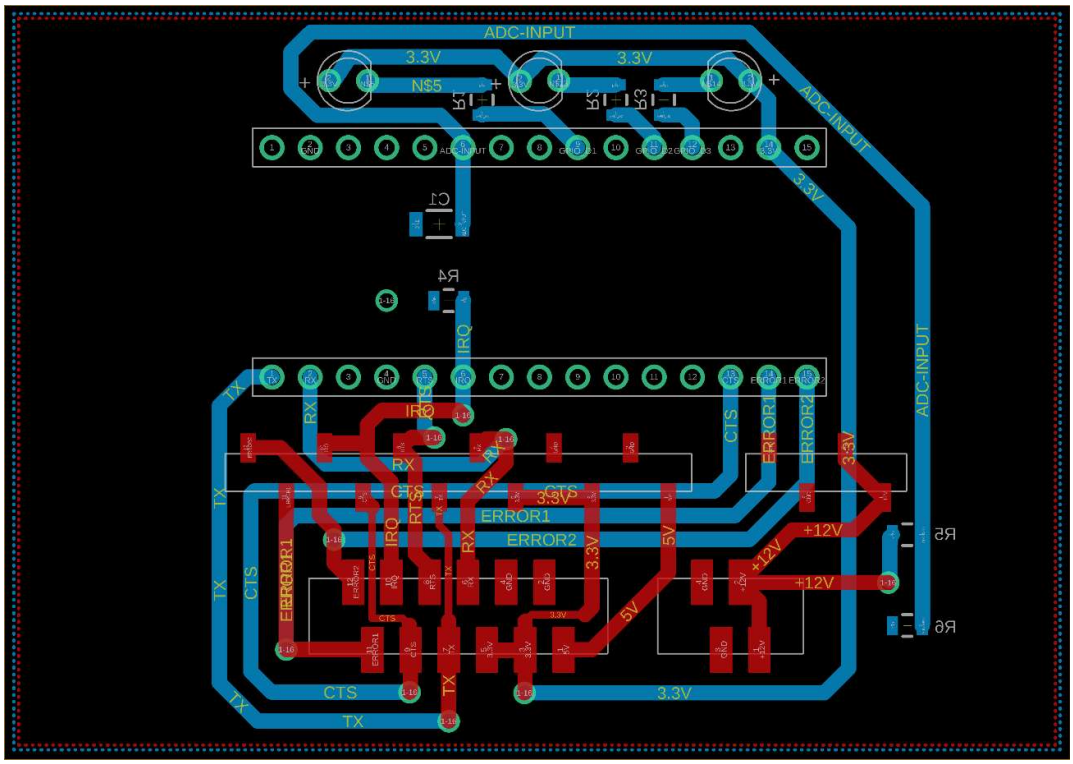
Anlage 3: Anforderungsliste

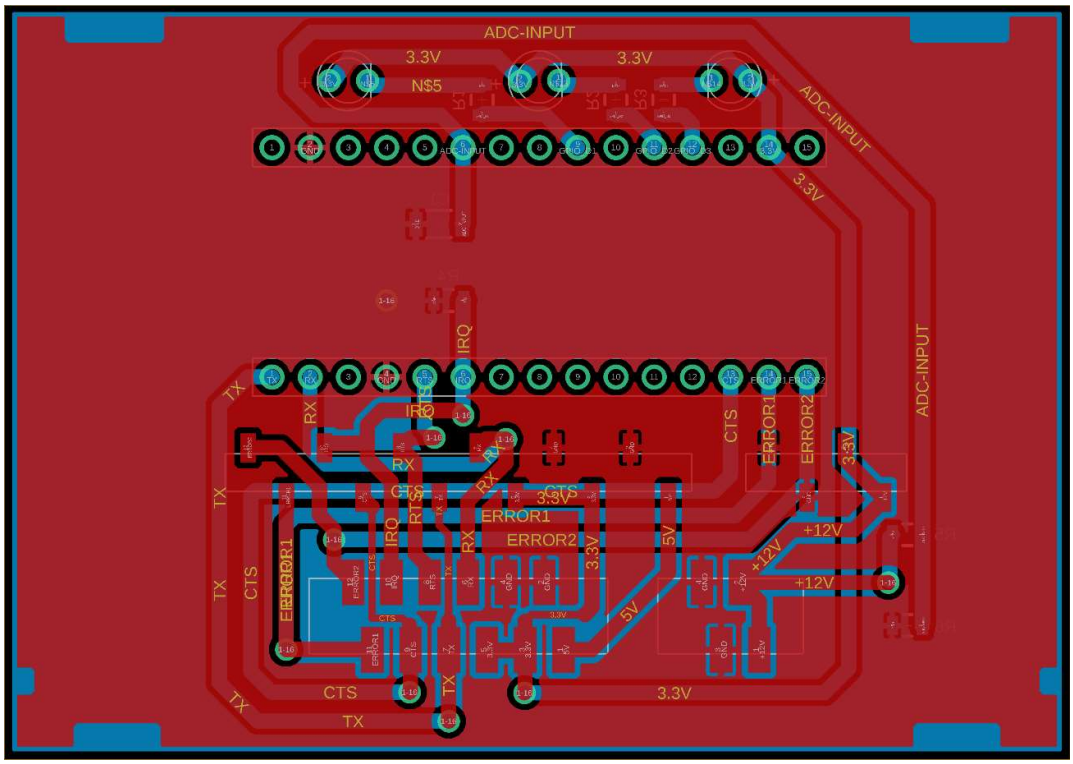
Forderung Wunsch	Anforderung	Kommentar
Platinenlayout		
Forderung	Maße (LxBxH)	70,8mm x 50,1mm x 15 mm
Forderung	Platinenstärke: max. 1,6 mm	Vorlage: Platinenstärke Slave-Platine
Forderung	Design-Rules	SMD (0805); 2,54 mm (Lochraster)
Forderung	Detektion Spannungsabfall	Über Spannungsteiler
Wunsch	2-Layer-Platine	Verwendung VIAs
STM32 Entwicklungsboard Nucleo-32 (NUCLEO-G431KB)		
Forderung	RS232-Schnittstelle	
Forderung	Datentypen	über Bus - Na
Forderung	Versorgungsspannung (12; 5V; 3,3V)	max. 1A
Forderung	RS232-Busspannung: 5V oder 3,3V	
Forderung	Detektion Spannungsabfall	
Forderung	Maximale Maße: (siehe Platine)	
Forderung	Maximale Höhe: 15 mm	Gehäuse
Forderung	On-board debugger/programmer	
Wunsch	FPU single precision	
Wunsch	Sehr gutes Verhältnis: Preis/Leistung	
Wunsch	Hoher Support, Forenbeiträge	
Weitere Bauteile		
Forderung	SMD-Bauteile	Widerstände, Kondensatoren
Forderung	Anschlussockel (4 & 12 Pin)	Kommunikation/Spannungsversorgung mit Slave-Platine
Wunsch	Headerleisten (4 & 12 Pin)	nur für Debugging (Kommunik./Spannungsvers.)
Wunsch	THT-LEDs	Debugging

Anlage 5: Schaltplan (EAGLE)

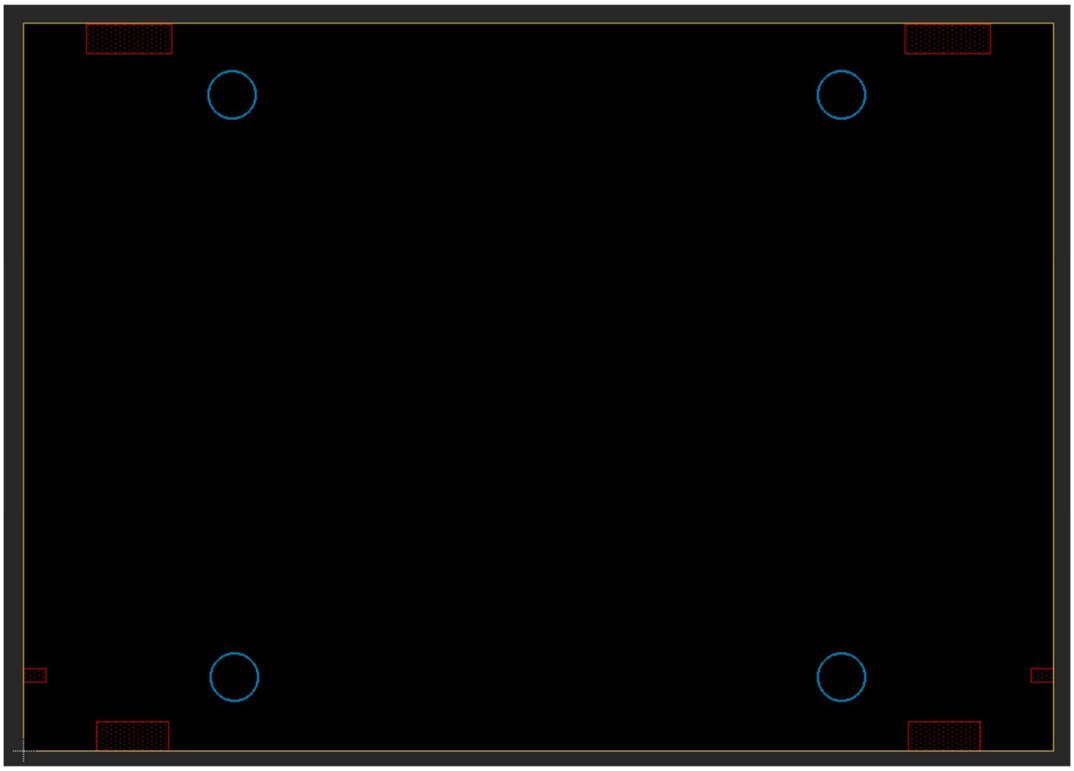


Anlage 6: Boardlayout - Top und Ratsnest (EAGLE)





Anlage 7: Keep-Out Area (Top / Bottom) EAGLE-Vorlage



Anlage 8: EAGLE Hinweise

Hinweise Schaltplan

- für alle Masseverbindungen den Netznamen in „GND“ (Ground) umbenennen

Hinweise Board-Layout

Erste Schritte

- Board-Layout mit dem Layer „Dimension“ auf die Maße der Leiterplatte begrenzen
- Mechanische Komponenten und Randbedingungen, wie z.B. Bohrlöcher, auf das Board integrieren

Design Rule Check (DRC) in EAGLE implementieren

- Beim Leiterplatten-Unternehmen (z.B. LeitOn) die DRC – Bibliothek herunterladen
- Im EAGLE Board-Fenster die Registerkarte „Bearbeiten“ auswählen und die Option „Design-Regeln…“ auswählen
- Im „Design-Regeln“ – Fenster in der Registerkarte „Datei“ die Option „Laden…“ betätigen und die heruntergeladene Bibliothek auswählen
- Anschließend „Übernehmen“ auswählen und mit „Ok“ bestätigen

DRC – Test

- In der linken Symbolleiste „DRC“ auswählen und „Prüfen“ betätigen
  ODER
  in der Registerkarte „Werkzeuge“ die Option „Layout prüfen (DRC)“ auswählen
- Im Fehlerfall erscheint ein neues Fenster
- Wurden keine Fehler gefunden, steht dies in der untersten Zeile

Leiterbahnen

- Keine scharfen Ecken (90°), weil hier die Signale reflektiert werden; es werden Winkel von 45° empfohlen
- Leiterbahndicke der Versorgungsspannung und Masseverbindung: mind. 1 mm
- Leiterbahndicke des Maschinenanschlusses: mind. 0,6 mm
- Leiterbahndicke der Signalleitungen: mind. 0,2 mm

- Bei nur zwei Platinenschichten (Top und Bottom):
    - Leiterbahnlayer für Top: Layer „1 Top“
    - Leiterbahnlayer für Bottom: Layer „2 Bottom“
- Verbindung der Leiterbahnen zwischen Top und Bottom mittels „Via“; die Drill-Größe des Vias sollten der Leiterbahndicke entsprechen

„Keep Out Area“

1. „Polygon“ in das Suchfeld eingeben
2. Spacing- und Isolate-Bereich einstellen (in dieser Arbeit: Spacing: 1 mm, Isolate: 0,3 mm)
3. Als Top-Layer „39 tKeepout“ und als Bottom-Layer „40 bKeepout“ auswählen
4. Äußere Umrandung zeichnen (Anfang und Ende müssen sich verbinden)
5. Es entsteht automatische eine gestrichelte Linie
6. Rechtsklick auf die gestrichelte Linie und „Ratsnest“ auswählen
7. Es erfolgt eine automatische Erstellung der Keepout-Fläche auf der Platine

Masserverbindung auf Board erstellen

1. „Polygon GND“ in das Suchfeld eingeben (WICHTIG: im Schaltplan muss jedes Massenetz in „GND“ benannt sein; GND = Ground)
2. Isolate – und Spacing-Bereich eingeben (siehe Hinweise zu „Keep-Out-Area“)
3. Für eine GND-Schicht auf der Top-Schicht die Layer „1 Top“ und für die GND-Schicht auf der Bottom-Schicht die Layer „2 Bottom“ auswählen
8. Äußere Umrandung zeichnen (Anfang und Ende müssen sich verbinden)
9. Es entsteht automatische eine gestrichelte Linie
10. Rechtsklick auf die gestrichelte Linie und „Ratsnest“ auswählen
11. Es erfolgt eine automatische Erstellung der Masse-Fläche auf der Platine
12. Löschen des Ratsnests: Rechtsklick auf die gestrichelte Linie und „Ripup“ auswählen und anschließend „Tryce, Via, or Polygon“

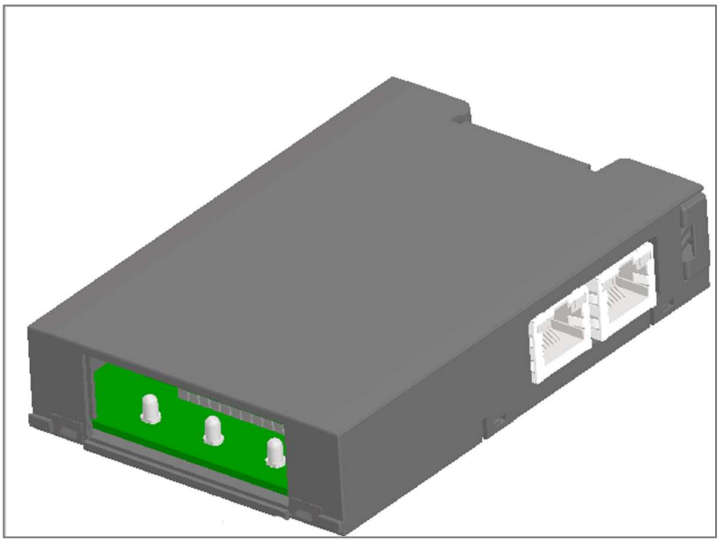
Massepins mit Massefläche verbinden (Hinweise)

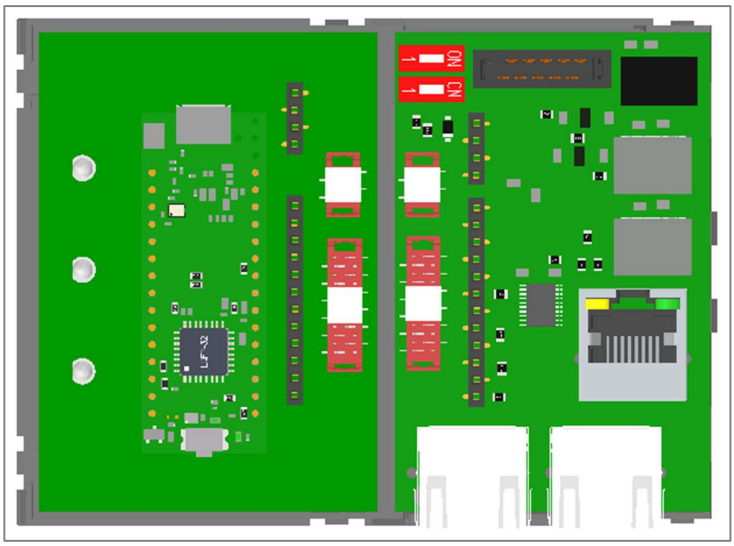
- In regelmäßigen Abständen die Massefläche mittels der Option „Ratsnest“ (siehe Hinweise „Masseverbindung auf Board erstellen) einstellen und wieder löschen, da erst hierdurch unnötige und überflüssige Airways (gelbe Luftlinien) verschwinden

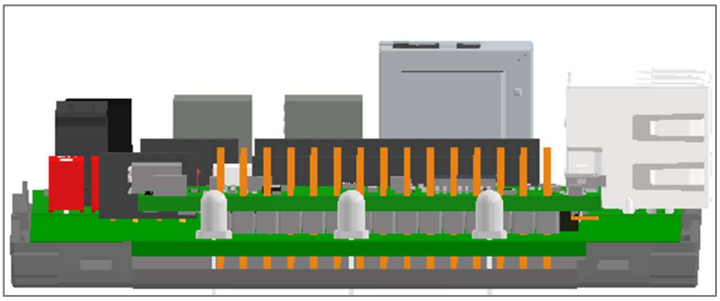
Anlage 9: Bauteilliste

Bauteil		Typ	Wert	Art.Nr.	Hersteller	Stück
-	NUCLEO	STM32G 431	-	3A991A2	STM	1
J1	SMT Socket Header	SMT Female	4-polig	61000411921	WE	1
J2			12-polig	61001211921		1
J3	WR-MM Connector	SMT Female	4-polig	690367290476		1
J4			12-polig	690367291276		1
R1	Widerstand	0805	82 Ω	RND 0805 1 82	RND	1
R2			150 Ω	RND 0805 1 150	RND	1
R3			430 Ω	RND 0805 1 430	RND	1
R4			10 kΩ	RND 0805 1 10K	RND	1
R5			100 kΩ	SMD-0805 100K	YAGEO	1
R6			33 kΩ	SMD-0805 33,0K		1
C1	Kondensator	0805	10 nF	X7R-G0805 10N	YAGEO	1
D1	LED	THT, Round, Watercle ar, 3mm, Round	Blau; 2,8V; 7mA	151034BS03 000	WE	1
D2			Gelb; 2V; 10mA	151034YS03 000		1
D3			Grün; 3,2V; 2mA	151034GS03 000		1
Für J3	WR-MM Connector	Male IDC	4-polig	690157000472	WE	1
Für J4			12-polig	690157001272		1
Für J3	WR-CAB	1.27 mm	4 pins	63910415521	WE	0,1 m
Für J4			12 piins	63911215521		0,1 m

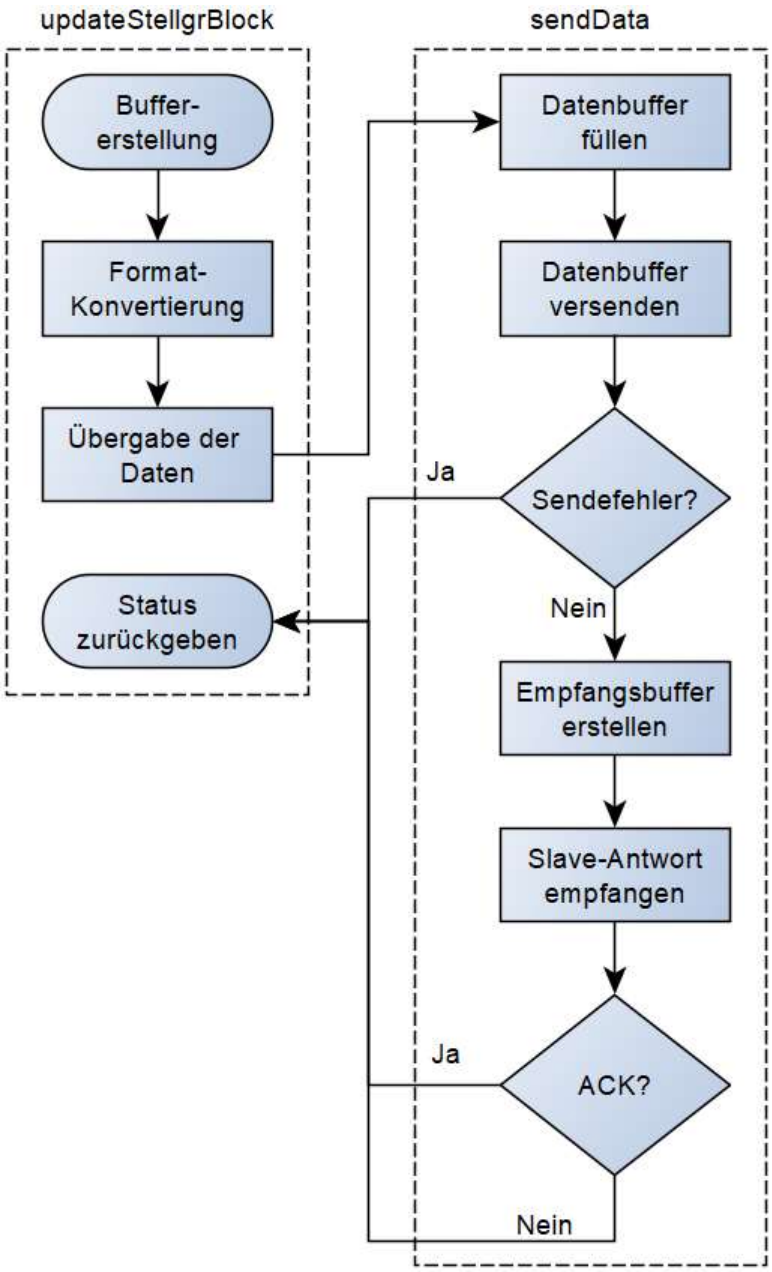
Anlage 10: Slave und User-unit; Gehäuse; Draufsicht; Vorderansicht



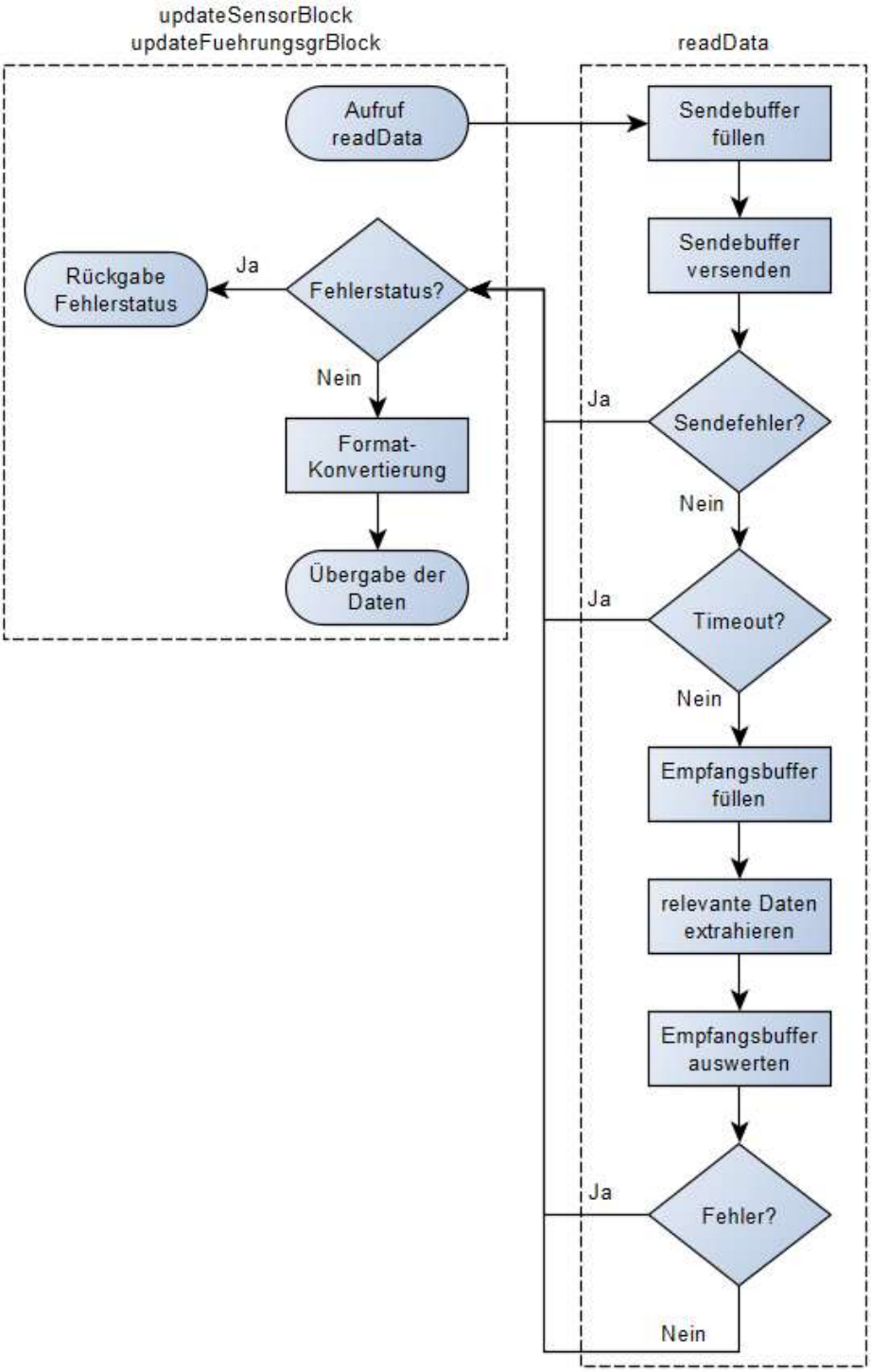




Anlage 11: Buskommunikation - Daten senden



Anlage 12: Buskommunikation - Daten lesen



Anlage 13: Projektstrukturplan - Kommunikation

Projektstrukturplan (PSP)

Projekttitel:	Kommunikation
Projektnummer:	MechatronikProjekt_2021_Team2_Kommunikation
Projektleiter:	LNe
Aktuelles Datum:	23.03.2021

PSP-Code	Teilaufgabe / Arbeitspaket	Verantwortlich	Mitarbeit	Fertigstellung (Soll)	Fertigstellung (Ist)
1.	**Platinenlayout**			**KW50**	**KW11**
1.1	Layout erstellen (Eagle) für Lochrasterplatine	Cla	LNe	KW48	KW48
1.2	Berechnungen (Vorwiderstand, Pufferkondensator)	LNe	Cla	KW49	KW49
1.3	Fertigung (Löten, Zuschnitt) Lochrasterplatine	LNe / CZh	Cla	KW49	KW49
1.4	Beginn Dokumentation	LNe	Cla	KW50 – KW12	KW12
1.5	Fertigstellung CAD-Modell	LNe	Cla	KW50	KW50
1.6	Inbetriebnahme Lochrasterplatine	LNe	Cla	KW50	KW50
1.7	Beginn Entwicklung professionelles Platinenlayout	Cla / LNe	BBe	KW1	KW1
1.8	Erste Tests Regelung mit Lochrasterplatine	MKi	SFr	KW5	KW5
1.9	Fertigstellung professionelles Platinenlayout	Cla / LNe	BBe / JGr	KW6	KW6
2.0	Bestellung bei AISLER auslösen	Cla / LNe	JGr	KW7	KW7
2.1	Löten professionelle Platine & Fertigung Flachbandkabel	Cla / LNe	SFr	KW11	KW11
2.2	Inbetriebnahme professionelle Platine	Cla / LNe	SFr	KW11	KW11
2.3	Tests der Regelung mit professionellen Platine	MKi	SFr	KW13	-
2.	**Bussystem**			**KW50**	KW12
2.1	Recherche	Cla/LNe	SHa/MMa	KW49	KW49
2.2	Dokumentation	Cla	LNe	KW50 – KW12	KW12
2.3	Programmierung	Cla	MKi	KW12	KW8
2.4	Communication.h	Cla	MKi	KW12	KW8
2.5	USART	Cla	LNe	KW2	KW2

2.6	Erste Tests Regelung und Kommunikation mit Lochrasterplatine	MKi	SFr	KW5	KW5
2.7	Tests der Regelung und Kommunikation mit professionellen Platine	MKi	SFr	KW13	-

Namenskürzel:

BBe	-	Bernd Becher
Cla	-	Christoph In der Au
CZh	-	Chen Zhang
JGr	-	Prof. Jörg Grabow
LNe	-	Lukas Neumeister
MKi	-	Martin Kirchner
MMa		Muhamma Maaz Majid
SFr	-	Dipl.-Ing. Stefan Franke
SHa		Sascha Halbgewachs

Anlage 14: Risikomanagement

Risikomanagement

Projektbezeichnung	Unmanned Surface Vehicle	
Projektleiter	Jörg Grabow	
Verantwortlich		
Erstellt am	04.12.2020	
Zuletzt geändert	31.03.2021	
Bearbeitungsstand	i.B.	in Bearbeitung vorgelegt fertig gestellt
Dokumentenablage	Projekte\Boot\00 Doku\01 RM	

Änderungsverzeichnis

Änderung			geänderte Kapitel	Beschreibung	Autor	neuer Zustand
Nr.	Datum	Version				
1	04.12.20	1.00	-	Startversion	Gr.	i.B.
2	06.01.21	1.01	alle	ALARP Struktur übernommen	Gr.	i.B.

in Bearbeitung (i.B.)

Vorlage (Vg.)

fertig gestellt (f.g.)

Risikoanalyse

Nr	Gefährdung	Schaden	Ursache	Häufigkeit	Risiko	Maßnahme	Maßn-Nr.	Häufigk nach	Risiko nach
1	Ausfall des GPS	3	Störung GPS-Baugruppe	3	9	Einsatz eines Backup GPS	M-1-1	2	6
2	Ausfall des Nucleo Board	4	Nucleo hängt sich auf	2	8	Kontrolle mit Watchdog	M-2-2	1	4
3	Falsche/ungenaue/keine Koordinaten des GPS	3	Ungenauigkeit der GPS-Baugruppe, EMV-Störungen	3	9	Validierung der GPS-Daten	M-2-3	2	6
4	Falscher/ungenauer/kein Winkel des Kompasses	3	Ungenauigkeit der Kompass-Baugruppe, EMV-Störungen, zu starker Seegang	3	9	Validierung der Kompass-Daten	M-2-4	2	6
5	Ausfall von Stecker, Platinenbefestigung, etc	3	Herstellungsfehler, schlechte Montage	2	6	Qualitätssicherung nach Montage	M-5-5	1	3
6	Ausfall von Stecker, Platinenbefestigung, etc	3	Verschleiß durch Tests/Montage	2	6	Blickkontrolle vor Inbetriebnahme der Baugruppe	M-5-6	1	3
7	Kurzschluss durch Wellengang	4	Lockerung im Gehäuse	1	4	Sicherungssteg im Gehäuse Einsetzen	M-1-7	1	4
8	Crash	5	Fremdes Schiff durchkreuzt die geplante Strecke	2	10	Ändern der geplanten Bahn, abbremsen des Schiffs, bis das fremde Schiff den Bereich verlassen hat	M-8-8	1	5
9	Fehlberechnung der Regelparameter	2	Softwarebug bspw. Durch andere reale Verhältnisse die im Programm nicht beachtet wurden	3	6	Redundanz der Software bzw. Hardware und Erstellung des Programmcodes durch zwei separate Teams	M-3-9	1	2
10	Fehlberechnung der Regelparameter	2	Gleiche Daten von GPS und Kompass hintereinander	3	6	Timestamp betrachten	M-2-10	1	2

Maßnahmenplan

Maßn.-Nr.	Maßnahme	Umsetzung	Termin	Erfüllungsstand in %				Kontrolle		Resultat	Bemerkung
		Verantwortlich		25	50	75	100	Wer	Wann		
M-1-1	Backup GPS auf Slave BG Positions-Controller	Grabow	30.03.21					Franke		Konstruktionszeichnungen, Dokumentation	
M-2-2	Kontrolle des Nucleo Boards mithilfe eines Watchdog	Majid						Team		Codeerstellung	
M-2-3	Validierung der GPS-Daten	Kleber								Info an Team GPS	
M-2-4	Validierung der Kompass-Daten	Kleber								Info an Team GPS	
M-5-5	Qualitätssicherung nach Montage	Neumeister						Kirchner		Sichtprüfung bei mehreren Funktionstest	
M-5-6	Blickkontrolle vor Inbetriebnahme der Baugruppe	Inbetriebnehmer								Sichtprüfung bei mehreren Funktionstest	
M-1-7	Sicherungssteg im Gehäuse einsetzen	Neumeister						Team		Konstruktionszeichnung	
M-3-8	Szenarien zur Kursänderung	Bahnsteuerung								Hinweis in der Dokumentation	
M-3-9	Doppelte Erstellung des Codes									Hinweis in der Dokumentation	
M-2-10	Betrachten des Timestamps im Code	Kirchner						Team		Codeerstellung	

Legende Maßnahmen-Nr.
M-1 konstruktive Maßnahmen
M-2 Schutzmaßnahmen
M-3 Dokumentation
M-4 Kennzeichnung
M-5 externe Prüfung

Risikomethoden

Schweregrade (qualitativ)		
Begriffe	**Wert in RA**	**Beschreibung**
katastrophal	5	Totalverlust des USV
kritisch	4	Kommunikation mit USV nicht mehr möglich
spürbar	3	Steuerbarkeit USV eingeschränkt
gering	2	Störung von Zusatzbaugruppen und Sensoren
unbedeutend	1	unbedeutende Störungen
Wahrscheinlichkeitsniveaus (halbquantitativ)		
Begriffe	**Wert in RA**	**Beschreibung**
häufig	5	1 pro 10 Benutzungen / 1 i n 10 h
möglich	4	1 pro 100 Benutzungen / 1 in 100 h
selten	3	1 pro 1000 Benutzungen / 1 in 1.000 h
sehr selten	2	1 pro 10.000 Benutzungen / 1 in 10.000 h
unwahrscheinlich	1	1 pro 50.000 Benutzungen / 1 in 50.000 h

Beschreibung der Risikobereiche	
Bereich	**Beschreibung**
roter Bereich:	Diese Risiken sind nicht akzeptabel.
grüner Bereich:	Diese Risiken sind akzeptabel.

Definieren der Risikobereiche (Bewertungskriterien)						
häufig	5	grün	rot	rot	rot	rot
möglich	4	grün	rot	rot	rot	rot
selten	3	grün	grün	rot	rot	rot
sehr selten	2	grün	grün	grün	grün	rot
unwahrscheinlich	1	grün	grün	grün	grün	grün
		1	2	3	4	5
		unbedeutend	gering	spürbar	kritisch	katastrophal

Risikoauswertung

Gesamtrisiko vor Maßnahmenumsetzung

		1 unbedeutend	2 gering	3 spürbar	4 kritisch	5 katastrophal
häufig	5	0	3	6	6	0
möglich	4	0	3	4	12	0
selten	3	0	0	17	14	0
sehr selten	2	0	2	1	4	0
unwahrscheinlich	1	0	0	0	0	0

Gesamtrestrisiko Risikoanalyse

		1	2	3	4	5
häufig	5	0	0	0	0	0
möglich	4	0	0	0	0	0
selten	3	0	3	0	0	0
sehr selten	2	0	0	15	22	0
unwahrscheinlich	1	0	5	13	13	0

Anlage 15: Sitzungsprotokolle

Mechatronik-Projekt
Team Regelung

Protokoll xx.xx.xxxx....1
Protokoll 17.02.2021....2
Protokoll 10.02.2021....4
Protokoll 03.02.2021....6
Protokoll 27.01.2021....8
Protokoll 20.01.2021....10
Protokoll 13.01.2021....12
Protokoll 06.01.2021....15
Protokoll 17.12.2020....18
Protokoll 09.12.2020....21
Protokoll 02.12.2020....24
Protokoll 30.11.2020....27
Protokoll 25.11.2020....29
Protokoll 23.11.2020....34
Protokoll 20.11.2020....36
Protokoll 18.11.2020....39
Protokoll 16.11.2020....42
Protokoll 13.11.2020....46
Protokoll 12.11.2020....48
Protokoll 09.11.2020....51
Protokoll 04.11.2020....52

Protokoll xx.xx.xxxx

Projekttitel:	Entwicklung und Implementierung der Regelungs- und Steuerungsstrategien
Projektnummer:	MechatronikProjekt_2021_Team2_Regelung
Projektleiter:	Muhammad Maaz Majid

Sitzung am:	xx.xx.xxxx, von xx:xx bis xx:xx Uhr
Ort der Sitzung:	xxxx
TeilnehmerInnen:	xxxx
Entschuldigt:	-
Nicht entschuldigt:	-

Inhalte der Sitzung:

1. xxxx
2. xxxx

Ergebnisse:

zu 1. xxxx

xxxx
- xxxx
  - xxxx
    - xxxx

zu 2. xxxx

xxxx
- xxxx
  - xxxx
    - xxxx

Nächste Schritte / Arbeitsaufträge:

Was?	Wer?	Bis wann?
xxxx	xxxx	xx.xx.xxxx
xxxx	xxxx	xx.xx.xxxx

Protokoll 17.02.2021

Projekttitel:	Entwicklung und Implementierung der Regelungs- und Steuerungsstrategien
Projektnummer:	MechatronikProjekt_2021_Team2_Regelung
Projektleiter:	Muhammad Maaz Majid

Sitzung am:	17.02.2021, von 09:30 bis 11:00 Uhr
Ort der Sitzung:	Online über BBB
TeilnehmerInnen:	Prof. Grabow, Dipl.Ing. Stefan Franke und alle M.Eng. Mechatoniker
Entschuldigt:	-
Nicht entschuldigt:	-

Inhalte der Sitzung:

1. Aktueller Stand von Team 2
2. Klärung offener Fragen von Team 2
3. Aktueller Stand von Team 1
4. Klärung offener Fragen von Team 1

Ergebnisse:

zu 1. Aktueller Stand von Team 2

- Inbetriebnahme Versuchsplatine
- Kommunikation in Betrieb genommen
- Konvertierung der Daten funktioniert

ToDo:
- WatchDog
- BrownOut
- Regler-Main-Loop
- Dokumentation

Hinweis von Prof. Grabow:
Auswertung der Sprungfunktion
- Wenn Ruderwinkel = 0 : Boot dreht sich
- Wenn Ruderwinkel ungleich 0 : Boot wird gesteuert
- Weitere Tests abwarten

Antwort von Martin Kirchner:
- Bisher Keine Auswertung der Sprungfunktion
- Wenn Geschwindigkeit = 0 -> können wir uns drehen

zu 2. Klärung offener Fragen von Team 2

Bestellung der Platine bei Aisler?
- Probleme: Erstellung Businesskonto bei Aisler

Kompletter Test der Regelung?
- Möglich ab 22.02.2021

zu 3. Aktueller Stand von Team 1

Test des Algorithmus

zu 4. Klärung offener Fragen von Team 1

Bestellungen?
- Pololu-Platinen sind im Labor (abholbereit)

GPS noch nicht angekommen?
- Nachsendeverfolgung bei DHL beantragt

Zeitraum für Software-Test in vollem Umfang?
- Jederzeit mit vorheriger Terminvergabe möglich

zu 5. Frage von Prof. Grabow:

Weiteres Vorgehen für Meetings
- Einigung: nach Prüfungsphase, wöchentliches Treffen
- Teilnehmer: Teamleiter, Prof. Grabow, Hr. Franke
- Optional, wenn erforderlich: mit allen anderen Teammitgliedern

Nächste Schritte / Arbeitsaufträge:

Was?	Wer?	Bis wann?
Kompletter Test der Regelung von Team 2	Team 2	Ab 22.02.2021

Protokoll 10.02.2021

Projekttitel:	Entwicklung und Implementierung der Regelungs- und Steuerungsstrategien
Projektnummer:	MechatronikProjekt_2021_Team2_Regelung
Projektleiter:	Muhammad Maaz Majid

Sitzung am:	10.02.2021, von 09:30 bis 11:00 Uhr
Ort der Sitzung:	Online über BBB
TeilnehmerInnen:	Prof. Grabow, Dipl.Ing. Stefan Franke und alle M.Eng. Mechatoniker
Entschuldigt:	-
Nicht entschuldigt:	-

Inhalte der Sitzung:

1. Aktueller Stand von Team 1
2. Hardware Update von Team 1
3. Klärung offener Fragen von Team 1
4. Aktueller Stand von Team 2
5. Hardware Update von Team 2
6. Klärung offener Fragen von Team 2

Ergebnisse:

zu 1. Aktueller Stand von Team 1

Virtuelles Modell ist fertig

Software: Codes verheiratet
- Tests können durchgeführt werden

Brown-Out auslesen

zu 2. Hardware Update von Team 1

Nachfrage von Bernd Becher zu RX/TX
- Korrektur von Prof. Grabow
- Anschlüsse sind richtig

Aktuelles Platinenlayout

zu 3. Klärung offener Fragen von Team 1

GPS noch nicht angekommen bei Herrn Kleber
- Sendungsverfolgung von Prof. Grabow

Bestellungen
- Polulus sind bestellt
- Noch nicht angekommen an der EAH

Termin Software-Test am Donnerstag 11.02.
- Ist möglich mit Herrn Franke

Satellitenanzahl mitschicken?
- Anzahl kleiner 4 – Fehler
- Anzahl großer 4 – keinen Fehler

zu 4. Aktueller Stand von Team 1

Softwaretests mit Herrn Franke durchgeführt
- Kleine Anpassungen waren nötig
- Funktionen sind alle gegeben

zu 5. Hardware Update von Team2

Spannungsabfall soll bei 12V über Spannungsteiler detektiert werden
- Keine Überwachung der Spannung bei Slave-Platine

zu 6. Klärung offener Fragen von Team 2

Abgabe Dokumentation?
- 26.03.2021

Präsentation halten?
- 26.03.2021

Nächste Schritte / Arbeitsaufträge:

Was?	Wer?	Bis wann?
Fertigstellung Platinenlayouts	Team 1/ Team 2	12.02.2021

Protokoll 03.02.2021

Projekttitel:	Entwicklung und Implementierung der Regelungs- und Steuerungsstrategien
Projektnummer:	MechatronikProjekt_2021_Team2_Regelung
Projektleiter:	Muhammad Maaz Majid

Sitzung am:	03.02.2021, von 09:30 bis 11:00 Uhr
Ort der Sitzung:	Online über BBB
TeilnehmerInnen:	Prof. Grabow, Dipl.Ing. Stefan Franke und alle M.Eng. Mechatoniker
Entschuldigt:	-
Nicht entschuldigt:	-

Inhalte der Sitzung:

1. Aktueller Stand von Team 2
2. Klärung offener Fragen von Team 2
3. Aktueller Stand von Team 1
4. Klärung offener Fragen von Team 1

Ergebnisse:

zu 1. Aktueller Stand von Team 1

Vorstellung und Auswertung des aufgestellten Terminplans von Maaz Majid
- Bis zum heutigen Tag, haben wir die Meilensteine erreicht

Test des Programms im Labor
- Martin Kirchner testet die Software mit Herrn Franke im Mechatronik-Labor
- erste Tests im Labor

Interne Test der Software
- Bahnkorrekturstrategie getestet

zu 2. Klärung offener Fragen von Team 2

Geschwindigkeit und Epsilon haben dieselbe Adresse wie Schub und Ruderausschlag?
- Prof. Grabow notiert und gibt Rückmeldung
- Herr Franke sollte Fehler korrigieren

zu 3. Aktueller Stand von Team 1

- Keine Veränderungen zur Vorwoche

zu 4. Offene Fragen von Team 1

Stand Feedback Platinen-Layout und Entscheidung Herstellung der Platine?
- Platinenlayout: Herstellung bei Aisler
- 2-Layer-Platine:
  - Brücken durch Leiterzug ersetzen
  - Steckverbinder für Stromversorgung ersetzen durch SMD-Verbinder

Bestellungen? / Hinweis GPS noch nicht angekommen
- GPS noch nicht angekommen
- Bestellungen sind ausgelöst

Format-Vorlage/ Vorgabe Darstellungsform der Algorithmen und Inhalt-Abschlussbericht?
- Vollständige Projektdokumentation
- Studenten sollen nahtlos weiterbearbeiten können
- Programmablaufplan mit State-Machine für Algorithmus

Nächste Schritte / Arbeitsaufträge:

Was?	Wer?	Bis wann?
-	-	-

Protokoll 27.01.2021

Projekttitel:	Entwicklung und Implementierung der Regelungs- und Steuerungsstrategien
Projektnummer:	MechatronikProjekt_2021_Team2_Regelung
Projektleiter:	Muhammad Maaz Majid

Sitzung am:	27.01.2021, von 09:30 bis 11:00 Uhr
Ort der Sitzung:	Online über BBB
TeilnehmerInnen:	Prof. Grabow, Dipl.Ing. Stefan Franke und alle M.Eng. Mechatoniker
Entschuldigt:	-
Nicht entschuldigt:	-

Inhalte der Sitzung:

1. Aktueller Stand von Team 1
2. Hardware-Update von Team 1
3. Offene Fragen von Team 1
4. Aktueller Stand von Team 2
5. Offene Fragen von Team 2

Ergebnisse:

zu 1. Aktueller Stand von Team 1

Präsentation von Michael Kleber
- Anpassung Meilensteine
- Abhängig von Bestellungen/Hardware
- Software fast fertig

ToDo:
- Ausplanung bis Ende des Semester

zu 2. Hardware-Update von Team 1

Präsentation von Michael Kleber und Bernd Becher
- Vorstellung Schaltplan
- Spannungsversorgung von 5V
- Datenleitung über TTL (GPS/Kompass und Slave)
- Level-Shift fällt komplett raus
- Spannungsüberwachung bei 12V
  - Spannungsteiler
  - Komparator-Eingang
  - Vergleich mit interner Referenzspannung
- LEDs
  - 2mA-LEDs gibt es auf dem Markt
- Leiterbreite auf dem Board
  - Mindestdicke: 0.16 inch
  - bei Laborfertigung zum Selbstätzen
- Leitungsdicke bei professionellen und selbstgeätzten Leitungen
  - je dünner, desto kritischer

- Thema: Ätzen
  - wenig Geld ist möglich professionelle Platinen zu bestellen
  - Online-Bestellung
  - EU-Hersteller: Aisler

@ Franke: Hinweise zu LEDs
- im Datenblatt nachschauen
- Kontrolle der LEDs
- Diagramme dazu auswählen
- Als erstes die Lichtintensität wählen

zu 3. Offene Fragen von Team 1

Nach aktuellem Stand Pololu BrownOut nur über Boodloader brennen möglich -> kein Interrupt möglich
- Grenze bei 2,9V
- Interrupt-Routine oder Flag setzen
- Von Grabow: nicht über Bootlader
- Interrupt:
  - Wenn Brown-Out auslöst -> Fehlerbit setzen
  - @Grabow: Prozessor merkt sich Fehlerzustände
    - Power-On-Reset, Brown-Out-Reset, Watch-Dog
    - Nach Reset -> sieht man Zustand, mit welchem er ausgegangen ist
    - Kompiler-Bug, wenn Betriebsspannung dran ist, dann sollte er nicht löschen
    - Reset-Zustand bleibt stehen, abhängig vom Kompiler (Arduino)

Unterstützung beim Ätzen der Platinen?
- Organisatorisch im EAH-Labor möglich
- Fertigung außer Haus?
- Rückmeldung von Grabow und Franke

Termin Software-Test mit Bus?
- Termin jederzeit möglich
- Mit Herrn Franke

Aktueller Stand der Bestellungen?
- Bestellungen werden ausgelöst

Aktueller Stand mit dem neuen NEO?
- Versand per Post

zu 4. Aktueller Stand von Team 2

Hardware-Layout in Bearbeitung

zu 5. Offene Fragen von Team 2

Nächste Schritte / Arbeitsaufträge:

Was?	Wer?	Bis wann?
Rückmeldung Platinenfertigung	Grabow/Franke	03.02.2021

Protokoll 20.01.2021

Projekttitel:	Entwicklung und Implementierung der Regelungs- und Steuerungsstrategien
Projektnummer:	MechatronikProjekt_2021_Team2_Regelung
Projektleiter:	Muhammad Maaz Majid

Sitzung am:	20.01.2021, von 09:30 bis 11:00 Uhr
Ort der Sitzung:	Online über BBB
TeilnehmerInnen:	Prof. Grabow, Dipl.Ing. Stefan Franke und alle M.Eng. Mechatoniker
Entschuldigt:	-
Nicht entschuldigt:	-

Inhalte der Sitzung:

1. Aktueller Stand von Team 1
2. Klärung offener Fragen von Team 1
3. Aktueller Stand von Team 2
4. Update Platinenlayout
5. Klärung offener Fragen

Ergebnisse:

zu 1. Aktueller Stand von Team 1

- keine Änderungen zur vorherigen Woche
- aktuell hoher zeitlicher Aufwand im Studium
    - viele Praktika, Übungen, Aufgaben zu erledigen

zu 2. Klärung offener Fragen von Team 1

Michael Kleber: Abholung der GPS-Module an der EAH?
- Prof. Grabow verschickt das Modul per Post

Aktueller Stand der Bestellung der Pololu-Boards?
- Bestellung wurde noch nicht ausgelöst
- Mitarbeiter dürfen noch keine Bestellungen auslösen

Können Studenten das Pololu-Board selbst organisieren und Quittungen einreichen?
- Prof. Grabow: Das ist nicht möglich

zu 3. Aktueller Stand von Team 1

- keine Änderungen zur vorherigen Woche
- aktuell hoher zeitlicher Aufwand im Studium
    - viele Praktika, Übungen, Aufgaben zu erledigen

zu 4. Update Platinenlayout

Präsentation von Lukas Neumeister (Thema: Spannungsausfall und Spannungsabfall)

Spannungsausfall
- Strommessung im Mechatroniklabor mit Nucleo-Board durchgeführt
- Erkenntnisse:
  - Hohe Stromaufnahme und Wärmeentwicklung bei 12V-Betrieb
    - 2 lineare Spannungswandler in Reihe
  - Lösung: 3,3 V direkt von Slave-Platine benutzen

Spannungsabfall
- ADC-Input verwenden mit Spannungsteiler
  - Schwellwert softwareseitig festlegen
- 12V von Slave-Platine für Spannungsteiler verwenden
  - Hintergrund: 12V-Spannungsversorgung fallen schneller ab, als die 3,3 V

zu 5. Klärung offener Fragen

Keine weiteren Hinweise, Fragen oder Anmerkungen etc.

Nächste Schritte / Arbeitsaufträge:

Was?	Wer?	Bis wann?
-	-	-

Protokoll 13.01.2021

Projekttitel:	Entwicklung und Implementierung der Regelungs- und Steuerungsstrategien
Projektnummer:	MechatronikProjekt_2021_Team2_Regelung
Projektleiter:	Muhammad Maaz Majid

Sitzung am:	13.01.2021, von 09:30 bis 11:00 Uhr
Ort der Sitzung:	Online über BBB
TeilnehmerInnen:	Prof. Grabow, Dipl.Ing. Stefan Franke und alle M.Eng. Mechatoniker
Entschuldigt:	-
Nicht entschuldigt:	-

Inhalte der Sitzung:

1. Aktueller Stand von Team 1 und Team 2
2. Update Platinenlayout von Team 1
3. Offene Fragen von Team 2
4. Offen Fragen von Team 1
5. Allgemeine Hinweise

Ergebnisse:

zu 1. Aktueller Stand von Team 1 und Team 2

Präsentation von Michael Kleber
- Unveränderter Stand vom 06.01.2021

Präsentation von Maaz Majid
- Implementierung der Software
- Vorstellung Ergebnisse der Berechnung Entfernung zu Peilung

Ergebnisse der Berechnungen

Ernst-Abbe-Hochschule Jena
University of Applied Sciences

Ergebnisse der Berechnungen

Berechnung der Entfernung					
Startpunkt	Ziel	Soll [km]	Ist [km]	Fehler [km]	Relativer Fehler
EAH_H3	Brandenburger-Tor	217,07	217,25	0,18	0,0008%
EAH_H3	Paradiesbahnhof	1,23	1,226	-0,004	-0,003%
EAH_H3	Netto Parkplatz	0,25	0,246	-0,004	-0,016%

Berechnung der Peilung (Bearing)			
Richtung	soll [°]	ist [°]	Fehler
NordOst	34,34	34,338	-0,002
NordWest	341,52	341,517	-0,003
SüdWest	235,94	235,935	-0,005
SüdOst	154,25	154,249	-0,001
Süd	180	180	0
Nord	0	0	0

zu 2. Update Platinenlayout

Präsentation von Bernd Becher
- Spannungsdetektion als Referenzspannungsquelle
  - Komperator als Eingang
- 0805 bei Widerständen
- LEDs: Baugröße 0805

Hauptstromversorgung bricht zusammen
- Was ist wenn 12V ausfallen? (Frage von Grabow)
  - Referenzspannung aus AVR
  - Stromversorgung nicht aus 5V, sondern aus 12V
  - großen Elko verwenden
  - Referenzspannung von 5V
  - AVR-Controller hat eigene Referenzspannungsquelle
    - Referenzspannung ist selbst einstellbar
    - Mit dieser Spannung kann man interne AD-Wandler betreiben
    - Stabile Referenzspannung unabhängig von Versorgungsspannung

1-Layer/2-Layer-Platine
- Layout auf einer Seite, sollten wir bevorzugen
- Eine Seite als Lötseite, andere als Bestückungsseite
- Von oben Bestücken und von unten verlöten

zu 3. Offene Fragen von Team 2

Überprüfung der Software im Labor möglich?
- Ja ist möglich ,wenn 1-2 Personen im Labor
- Schnittstelle ist funktionsfähig im Labor
- Tests durchführen mit der Regel
- Buskommunikation-Test?
  - Ja auch möglich
  - Virtueller Com-Port wird eingerichtet
  - Über Internet können wir Com-Schnittstelle ‚Tunneln'

Welche Dokumentation sind abzugeben? Sind alle Dokumente abzugeben (siehe Projektabschlussbericht.docx)?
- Nur welche wir selbst verwenden! (Ist nur ein Vorschlag von Grabow gewesen)

Hinweis für Projektabschlussbericht von Prof. Grabow!
- so aufbauen, dass nachfolgende Studierende einfach einsteigen können
-

Dokumentation bei letzten Meeting gezeigt für Abschätzung des Risikos. Wo sind Unterlagen zu finden?
- Moodle - Unterlagen - Organisation – Risikomanagment
- Dokument ist aktuell

TimeStamp, welche Form/Aufbau hat RTC?
- Rückmeldung von Grabow in Moodle

Anwenderschnittstelle Länge:
- Richtig: 8 Byte

zu 4. Offene Fragen von Team 1

Ab wann können der statische Fehler des Kompass am Boot ermittelt werden?
- Boot ist in Montagehalle und wird verschoben um 4 Wochen
- Wassertest sieht schlecht aus
- Möglichkeit im Labor zu drehen

Lötverbindung Pololu-Board? (Ein-Layer-Platine):
- Ein-Layer-Platine verwenden
- Datei schicken an Grabow

Funktion von Error1/2?
- Inputs
- Kommen vom Master zum Slave
- Funktion noch nicht festgelegt

Notwendigkeit Überwachung?
- Nur die Spannung überwachen die wir verwenden

Bestellungen? (GPS, Polulu)
- Polulu noch nicht bestellt
- NEOs sind abholbar

Interne/Externe Referenspannung. Woher?
- Datenblatt vom Controller
- AREF – PIN Referenzspannung
- Externe Belegung mit Brücken
- Interne Referenzspannung mit C5 zur Glättung
- SMD-Variante -> PIN20
- PB-Variante -> PIN18
- Für Grabows-Software
    - Interne AD-Wandler
    - Gibt Wert aus
    - Über Spannungsteiler
    - Über Software wird Spannung überprüft
- Komperator-Funktion:
    - Schwellwerte festlegen

zu 5. Allgemeine Hinweise von Prof. Grabow

Prof. Grabow ist am Montag, den 18.01.2021 in Jena im Labor
- NEOs können abgeholt werden

Nächste Schritte / Arbeitsaufträge:

Was?	Wer?	Bis wann?
-	-	-

Protokoll 06.01.2021

Projekttitel:	Entwicklung und Implementierung der Regelungs- und Steuerungsstrategien
Projektnummer:	MechatronikProjekt_2021_Team2_Regelung
Projektleiter:	Muhammad Maaz Majid

Sitzung am:	06.01.2021, von 09:30 bis 11:00 Uhr
Ort der Sitzung:	Online über BBB
TeilnehmerInnen:	Prof. Grabow, Dipl.Ing. Stefan Franke und alle M.Eng. Mechatoniker
Entschuldigt:	-
Nicht entschuldigt:	-

Inhalte der Sitzung:

1. Aktueller Stand von Team 1 und Team 2
2. Risikomanagment
3. Fragen über Hardware und Platinenlayout von Team 1 und 2
4. Klärung offener Fragen von Team 1

Ergebnisse:

zu 1. Aktueller Stand von Team 1 und Team 2

Präsentation von Maaz Majid
- Aktueller Stand von Team 2
- Implementierug des Teil-Programms
- Stand der Simulation
    - Parameter von Wasserversuchen? Wann sind Tests geplant?
        - Xcos-Modell lädt Prof. Grabow in Moodle
            - Reale Werte für Reibung und Antriebskräfte verwenden

Präsentation von Michael Kleber
- Aktueller Stand von Team 1
- Präsentation von Michael Kleber

zu 2. Risikomanagment

Präsentation von Michael Kleber
- Hinweis von Grabow:
    - Dokument von Grabow mit Tipps und Hinweisen:

zu 3. Fragen über Hardware und Platinenlayout von Team 1 und 2

Präsentation von Lukas Neumeister:
- 1. Frage: Erstellung einer professionellen Platine mit Entwicklungsboard?
  - Team 2 mit Nucleo-Board: Ja, das ist möglich
  - Team 1 mit Pololu-Board: kein großer Unterschied

- 2. Frage: Spannungsüberwachung
  - Spannungsteiler mit 2 hochohmigen Widerständen
  - Analog-In-Eingänge verwenden

Hinweise von Prof. Grabow und Herrn Franke:
- 1-Layer-Platine verwenden bei Team 2
- Überprüfung des Board-Layouts (https://aisler.net/)

zu 4. Klärung offener Fragen von Team 1

Stand der Bestellung Pololu-Board?
- Kein neuer Lieferstatus

Stand NEO M8P?
- Rücksprache mit Prof. Voß
- Keine freien NEOs verfügbar, alle in Verwendung
- Bestellung ausgelöst von Prof. Grabow für zwei NEOs

Brown Out Testumgebung?
- Möglich im Mechatronik-Labor
- Testschaltung aufbauen
- Tipp von Prof. Grabow:
  - 1. Draht mit 5V-Versorgung
  - 2. Draht mit 1. Draht verdrillen und Spannungs-Impulse geben

Watchdog-Brown-Out zählen? Maßnahmen?
- In Fehler-Prokoll schreiben
- Watchdog-Interrupt-Routine löst aus
- Internes E-Prom wichtige Daten sichern

Was soll bei einem NACK passieren?
- Daten nochmal senden
- Nach nochmaligen NACK? Rücksprache mit Herrn Franke
- Von COM-Port können Störungen eingekoppelt werden

Wie genau soll der Error-Frame gesendet werden? Wann kommt Error-Byte und wie?
- Ganz normaler Datensatz, nur an andere Adresse
- Komplett neuer Frame
- Error-Byte dranhängen
- Wie oft senden? Wenn Fehler auftritt: Muss im Speicher stehen.
- Intern zählen wie oft Error-Byte auftritt

Hinweis/Idee von Prof. Grabow:
- Byte immer senden (Fehler-Bit)
- Keine Fehler bei 0

- Wenn Fehler, dann 1
- Bei Watchdog-Reset ausgelöst? Was soll passieren?
    - Rest aufgetreten -> Bit zurücksetzen
    - Register ist ein Flag-Register
    - Reset-Flag
    - Brown-Out-Flag

Wie ist CRC aufgebaut? (Frage von Inken Kestler)
- Innerhalb des Frames
- Hinten dran gehängt
- Nur Daten werden berechnet
- Siehe Dokumentation

Nächste Schritte / Arbeitsaufträge:

Was?	Wer?	Bis wann?
-	-	-

Protokoll 17.12.2020

Projekttitel:	Entwicklung und Implementierung der Regelungs- und Steuerungsstrategien
Projektnummer:	MechatronikProjekt_2021_Team2_Regelung
Projektleiter:	Muhammad Maaz Majid

Sitzung am:	17.12.2020, von 08:00 bis 09:30 Uhr
Ort der Sitzung:	Online über BBB
TeilnehmerInnen:	Prof. Grabow, Dipl.Ing. Stefan Franke und alle M.Eng. Mechatoniker
Entschuldigt:	-
Nicht entschuldigt:	-

Inhalte der Sitzung:

1. Aktueller Stand von Team 1 und Team 2
2. Vorstellung Platinenlayout von Team 2
3. Auswahl Entwicklungsboard von Team 2
4. Genauigkeit GPS von Team 2
5. Offene Fragen von Team 1
6. Hinweise von Prof. Grabow und Herrn Franke

Ergebnisse:

zu 1. Aktueller Stand von Team 1 und Team 2

Präsentation von Michael Kleber und Muhammad Maaz Majid

zu 2. Vorstellung Platinenlayout von Team 2

Präsentation von Lukas Neumeister
- aktueller Stand des Platinenlayout
- vorläufige Platine mit Lochrasterplatine

zu 3. Auswahl Entwicklungsboard von Team 2

Präsentation von Michael Kleber und Bernd Becher
- Arduino NANO Every
- A-Star Pololu
  - ATMega 328PB
- Wattuino Pro mini PB
  - ATMega 328PB

zu 4. Genauigkeit GPS von Team 2

- Präsentation von Michael Kleber
- Warmlauf von 2-3h
- Gesamtdauer: 60min
- große Unterschiede der Genauigkeit bei wolkigen, sonnigen Wetter und der Position (Fensterbrett, Garten)
- gute Ergebnisse
  - Garten und Fensterbrett
  - 6 m Abweichung unter freien Himmel
  - im Durchschnitt: 8 Satelliten
- Vorschlag:
  - Anderes GPS-Modul verwenden
  - Weitere Tests mit neuen GPS-Modul durchführen
  - Ergebnisse vergleichen
  - Prof. Voss kontaktieren

zu 5. Offene Fragen von Team 1

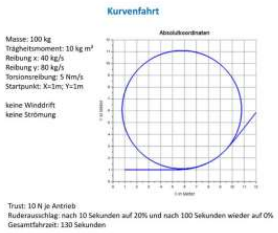
Wie groß ist der Input-Buffer des Masters? Kann der komplette Frame gesendet werden?
- Kompletter Frame muss gesendet werden

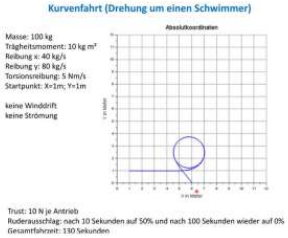
Timestamp ist bei Kompass nicht verfügbar. Kann RTC implementiert werden?
- Timer selbst programmieren
  - Variante: Real-Time-Clock von außen (siehe Schaltung von Grabow)
- Software-Uhr und Synchronisation mit Bus

Kann die Spannungsmessung über eine externe Schaltung realisiert werden?
- Spannungsteiler verwenden
- Bown-Out

Wie soll der Watchdog-Reset erkannt/gezählt werden?
- Je nach Microcontroller wird die Brown-Out-Auslösung nicht gespeichert.
- Spannungsteiler und mit 2x ADC-Boards
- Referenzspannung verwenden
- Spannungsabweichung von 20% nachweisen
- Spannungsquelle puffern
- Controller mit 5V
- Betriebsspannung stützen! Sehr wichtig!

zu 6. Hinweise von Prof. Grabow und Herrn Franke





Wie hängt Schub mit Geschwindigkeit zusammen?

- Parameter entsprechen dem Schiffsmodell
- Steuerung der beiden Motoren über Schub und Ruderausschlag
- Start bei 1/1
- Nach 10s Ruder bei 20% eingeschalgen für 100s
- Nach 100s Ruder wieder auf 0% gestellt
- 50% Ruderausschlage:
- Ein Motor 1 aus, Motor 2 ist an

Überprüfung unserer Hardware/Software (Herr Franke)

- Wir schließen Hardware an Franke's Software-Modell an
- Kontrolle vom Verhalten des Schiffs

Nächste Schritte / Arbeitsaufträge:

Was?	Wer?	Bis wann?
Präsentation Risikomanagment von Team 1	Michael Kleber	06.01.2020

Protokoll 09.12.2020

Projekttitel:	Entwicklung und Implementierung der Regelungs- und Steuerungsstrategien
Projektnummer:	MechatronikProjekt_2021_Team2_Regelung
Projektleiter:	Muhammad Maaz Majid

Sitzung am:	09.12.2020, von 09:30 bis 11:00 Uhr
Ort der Sitzung:	Online über BBB
TeilnehmerInnen:	Prof. Grabow, Dipl.Ing. Stefan Franke und alle M.Eng. Mechatoniker
Entschuldigt:	CDHWs (Parallelveranstaltung)
Nicht entschuldigt:	-

Inhalte der Sitzung:

1. Aktueller Stand
2. Vorstellung Platinenlayout von Team 2
3. Vorstellung Bahnkorrekturstrategie von Team 2
4. Offene Fragen von Team 1
5. Offene Fragen von Team 2
6. Hinweise von Prof. Grabow und Hr. Franke

Ergebnisse:

zu 1. Aktueller Stand

Team 1:
- Präsentation von Michael Kleber
- Code für GPS ist fertig
  - Dezimalgrad-Zahlen kommen raus
- Entwicklung Fixpoint-Programmierung
- Entwicklung CRC

Team 2:
- Schaltplan und Platinenlayout ist fertig
- grobe Struktur des Programms entwickelt (Softwarearchitektur)
- Arbeitspaket für Implementierung

zu 2. Vorstellung Platinenlayout von Team 2

Präsentation von Christoph In der Au
- Hinweise von Prof. Grabow:
  - Kurzschlüsse bei Spannungsversorgung beachten

zu 3. Vorstellung Bahnkorrekturstrategie von Team 2

Präsentation von Maaz Majid

zu 4. Offene Fragen von Team 1

Stundenplan-Kollision von CDHAW-Studenten
- Vorschlag:

- Aktueller Stand wird von Projektleiter übermittelt
- späteres Zuschalten in das Meeting

Genauigkeit der GPS-Daten:
- Tests von Michael Kleber durchgeführt
- Hinweis Grabow:
  - Algorithmus zur Überprüfung der GPS-Daten ist sinnvoll
  - Mitführen von einfachen Überprüfungs-Algorithmus

Höhenangabe auf dem Meer mitsenden? Alternative: Anzahl verfügbarer Satelliten mitsenden?
- Höhenangabe ist nicht notwendig
- Anzahl verfügbarer Satelliten ist sinnvoll mitzusenden
- Probleme: andere Frequenzen können GPS beeinflussen

Basisstation, welche die Route überwacht?
- Bodenstation mit Richtungsantenne ist vorhanden

Frage über FIFO von Inken Kestler:
- FIFO hängt an COM-Port
- Wir schieben alle GPS-Daten in FIFO rein
- Hardware-Interrupt-Routine
  - Liest Byte für Byte raus
- Empfangen:
  - Daten stehen im FIFO sicher drin
  - Durch FIFO gehen Daten nicht verloren
- Bei 100 Byte senden
- Wir lesen kompletten Speicherblock in FIFO einlesen
  - Wenn Block drin ist
  - Dann auslesen des Blocks

zu 5. Offene Fragen von Team 2

Wird die GPS-Geschwindigkeit anhand von GPS-Daten ermittelt oder in Kombination mit einem integrierten Beschleunigungssensor?
- wird über GPS-Daten ermittelt
- kein Beschleunigssensor
- Genaunigekiten -> schickt Michael Kleber zu
- Berechnung Geschwindigkeit über GPS
- Geplant: ein Inertialsysten

Timestamp
- Wird mit aufgenommen
- Bekommen wir mit geliefert im Protokoll

Antrieb – Haswing Protruar 1.0
- Schubkraft ist positiv und negativ anzusetzen
- Kein Datenblatt von Drehzahl und Kraft

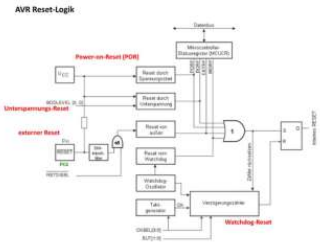
Sign. Fixpoint
- immer 2er-Kompliment

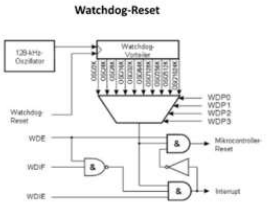
zu 6. Hinweise von Prof. Grabow und Hr. Franke

Über Risikomanagment nachdenken
- siehe Risikimanagmnent.docx und Präsentation von Prof. Grabow
  - Was passiert wenn?
  - Wenn Risiko nicht zu beseitigen?
    - Dann notieren, was passiert

Fehlerbeseitigung (RESET-Logik)
- Variante: externer Reset über RC-Glied (verwenden wir)
  - Prozessor erst gestartet, wenn 5V anliegt

- 2 weitere RESET-Arten integrieren
  - 1. Unterspannungsreset
    - programmieren auf Unterspannungsquelle
    - Wenn unter 3,3 oder 5V fällt -> dann löst Rest aus
  - 2. Watchdog-Reset verwednen
    - Software selbst überwachbar
    - Wenn Software abstürzt
    - Timer überwacht, ob Software hängt





Virtuelles Schiff und virtuelle Ostsee
- Präsentation von Hr. Franke
- Schiffsmodell erstellt mit MATLAB
- Wir senden Ihm Steuerdaten
- Modell antwortet mit GPS-Daten
- Wie reagiert die Steuerung?
- Der Bus ist Simulationssystem
- Wir können Regler simulieren/überprüfen
- Echtzeit ist hinterlegt
- Nach Modell 2 mit Corioliskraft und Reibkräften hinterlegt

Nächste Schritte / Arbeitsaufträge:

Was?	Wer?	Bis wann?

Protokoll 02.12.2020

Projekttitel:	Entwicklung und Implementierung der Regelungs- und Steuerungsstrategien
Projektnummer:	MechatronikProjekt_2021_Team2_Regelung
Projektleiter:	Muhammad Maaz Majid

Sitzung am:	02.12.2020, von 09:30 bis 11:00 Uhr
Ort der Sitzung:	Online über BBB
TeilnehmerInnen:	Prof. Grabow, Dipl.Ing. Stefan Franke und alle M.Eng. Mechatoniker
Entschuldigt:	-
Nicht entschuldigt:	-

Inhalte der Sitzung:

1. Aktueller Stand - Was wurde gemacht? Was sind die nächsten Schritte?
2. Vorstellung Platinenlayout
3. Vorstellung Bahnkorrekturstrategie
4. Klärung offener Fragen
5. Hinweise von Prof. Grabow an Team 2
6. Hinweise von Prof. Grabow an Team 1

Ergebnisse:

zu 1. Aktueller Stand - Was wurde gemacht? Was sind die nächsten Schritte?

Präsentation von Inken Kestler

zu 2. Vorstellung Platinenlayout

Präsentation von Bernd Becher

zu 3. Vorstellung Bahnkorrekturstrategie

Präsentation von Martin Kirchner (Verschoben auf 09.12.2020)

zu 4. Klärung offener Fragen

Wie ist aktueller Lieferstand Entwicklungsboard?
- Hr. Franke meldet sich (1-2 Tage)

Wird Boot getestet bevor es zu Wasser gelassen wird?
- Regelung wird getestet vorher

Angaben von Geschwindigkeit in m/s? Warum?
- SI-Einheiten ausgewählt

Fixpoint-Format:
- z.B. 7.9
- Datenformat sieht wie Integer aus
- Vorder Punkt 2hoch n+1, nach Punkt 2 hoch n-1
- 4.4 Fixpoint addieren
- 2 Bitzahlen addieren -> Überlauf mit Carry-Bit

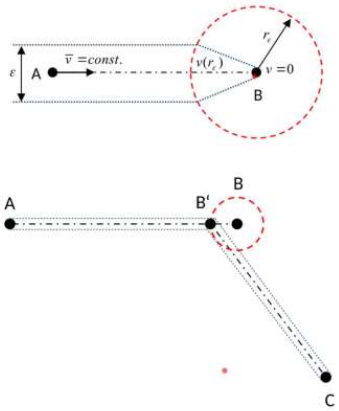
- Multiplikation -> Verdopplung der Bits
  - 2 mal 4.4 = 8.8 Fixpoint
  - Höhere und untere 4 Bits abschneiden
  - Ergebnis = 4.4
- Kompassdaten: 359°
  - Begrenzung einbauen

Ablauf RS232-Kommunikation:
- CTS und RTS-Leitungen
- RTS auf Masse:
- Buffer voll: Leitung auf Masse lassen
- FIFO: nicht voll -> Leitung auf Masse lassen

Prüfungsanmeldung
- für beide Teilmodule im SelfService anmelden

zu 5. Hinweise von Prof. Grabow an Team 2

Stichwort: Bahnsteuerung



- Wenn Punkt B' erreicht, dann Punkt B gelöscht
- Ziel: keine Abbremsung des Bootes

zu 6. Hinweise von Prof. Grabow an Team 1

Kompass-System
- Auf Schiff gibt es magnetische Einflüsse (Motoren, statisches Magnetfeld)
- Ziel: Störeinflüsse kompensieren
  - Korrekturtabelle erstellen

GPS-System
- 5min GPS-Daten geloggt
- Keine Bewegung des GPS-Sensors
- Hohe Streuung der Kompass-Daten
- Ziel für Team 1:
  - Algorithmus entwickeln

- Punktgenauigkeit verbessern

Lat	Lng	Höhe	I
50.665219	11.12046	385 m	
50.665235	11.120454	387 m	
50.665216	11.12045	388 m	
50.665236	11.120419	390 m	
50.665254	11.120419	389 m	
50.665217	11.120446	391 m	
50.665234	11.120438	391 m	
50.665217	11.12045	390 m	
50.665242	11.120461	388 m	
50.665229	11.120459	387 m	
50.665236	11.120447	387 m	
50.665223	11.120454	386 m	
50.665253	11.120437	385 m	
50.665242	11.120464	383 m	
50.665263	11.120424	385 m	
50.66523	11.120461	386 m	
50.665189	11.120469	388 m	
50.665196	11.120485	388 m	
50.665176	11.120511	387 m	
50.665217	11.120485	387 m	

Nächste Schritte / Arbeitsaufträge:

Was?	Wer?	Bis wann?
Präsentation Bahnsteuerung	Martin Kirchner	09.12.2020

Protokoll 30.11.2020

Projekttitel:	Entwicklung und Implementierung der Regelungs- und Steuerungsstrategien
Projektnummer:	MechatronikProjekt_2021_Team2_Regelung
Projektleiter:	Muhammad Maaz Majid

Sitzung am:	30.11.2020, von 15:15 bis 16:15 Uhr
Ort der Sitzung:	Online über Skype
TeilnehmerInnen:	Muhammad Maaz Majid, Chen Zhang, Christoph In der Au, Sascha Halbgewachs, Martin Kirchner, Lukas Neumeister
Entschuldigt:	-
Nicht entschuldigt:	-

Inhalte der Sitzung:

1. Versionsverwaltungsplattform festlegen
2. Aktueller Stand von Team Kommunikation
3. Aktueller Stand von Team Regelung

Ergebnisse:

zu 1. Versionsverwaltungsplattform festlegen

Erfahrungen
- GIT-Hub @Sascha
- SVN @Christoph
- GIT-Lab @Martin
- Bit Bucket @Martin
- CUBE-IDE -> Plugins

zu 2. Aktueller Stand von Team Kommunikation

neues Layout mit USART 2
- mit GPIO's
- Debug-Funktionen
- Verwendung von LEDs

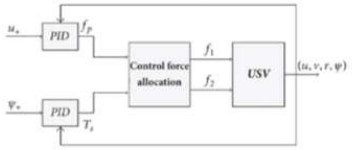
zu 3. Aktueller Stand von Team Regelung

Präsentation von Sascha mit aktuellen Stand
- mehrere Modelle von Grabow für SCILab
- mathematische Dokumente

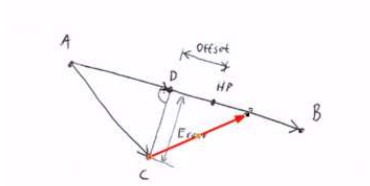
FormForceAlloc Präsentation von Martin
- Variante 2 ist Favorit:
- engere Radien fahren möglich

Grabow:
- Upload von SCILab Unterlagen
- ähnlich Variante 2
- kein Schub und trotzdem lenken
- Schwerpunkt kann als konstant angenommen werden

ToDo
- USV-Block



@ Chen:
- Programmierung:
  - Berechnung der Abstände und WInkel von 2 Koordinaten



- Boot soll Unterpunkte anfahren
- Punkt berechnen, der knapp vor dem Boot ist
- mit steileren Winkel zur originalen Bahn, wenn Entfernung zu groß
- flacherer Winkel zur Bahn, wenn wir uns der idealen Bahn nähern

- Was wollen wir berechnen?
- zuerst Punkt D berechnen
- danach Offset:
  - Abhängig von dem veränderlichen Error
  - wenn bei HP angekommen
    - ständig neu berechnen
- Alternativ: mit abklingender E-Funktion
  - Referenz: Norden von Kurswinkel
  - Referenzlinie von Kompass
  - Wichtig: Funktionen mit denen wir Abstände berechnen können
  - Punkt HP berechnet sich aus Punkt A, B und C
  - Fallunterscheidung: wenn über Punkt B gedriftet
  - wenn Punkt HP über B, dann direkt zu HP fahren

Nächste Schritte / Arbeitsaufträge:

Was?	Wer?	Bis wann?
Offene Fragen klären! (siehe Fragenliste von Maaz)	Maaz Majid	02.12.2020
Präsentation von Martin zu FormForceAlloc	Martin Kirchner	02.12.2020

Protokoll 25.11.2020

Projekttitel:	Entwicklung und Implementierung der Regelungs- und Steuerungsstrategien
Projektnummer:	MechatronikProjekt_2021_Team2_Regelung
Projektleiter:	Muhammad Maaz Majid

Sitzung am:	25.11.2020, von 09:30 bis 11:00 Uhr
Ort der Sitzung:	Online über BBB
TeilnehmerInnen:	Prof. Grabow, Dipl.Ing. Stefan Franke und alle M.Eng. Mechatoniker
Entschuldigt:	-
Nicht entschuldigt:	-

Inhalte der Sitzung:

1. Vorstellen aktueller Stand (Team Sensorik und Team Regelung)
2. Vorstellen Projekt- und Terminplan (Team Regelung)
3. Vorstellen Regelungsstrategie (Team Regelung)
4. Klärung offener Fragen (Team Sensorik und Team Regelung)

Ergebnisse:

zu 1. und 2. Vorstellen aktueller Stand, Projekt- und Terminplan (Team Sensorik und Team Regelung)

Aktueller Stand Team 1
- Präsentation von Maaz Majid

Aktueller Stand Team 2
- Präsentation von Michael Kleber
- Umrechnung und Umwandlung Datentypen macht Team 1 - Sensorik

Hinweis von Grabow:
- GPS-Geschwindigkeit soll übertragen werden von Team 1 – Sensorik
- notwendig für Bahnsteuerung und Regelung

zu 3. Vorstellen Regelungsstrategie (Team Regelung)

Präsentation von Martin Kirchner zur Bahnsteuerung
- Vorstellung einer Demo-Regel-Strategie

- Punkt A, Punkt B mit mittlerer Geschwindigkeit
- Punkt zu Punkt fahren

Muss mittlere Geschwindigkeit eingehalten werden?
- diese Geschwindigkeit wird nicht überschritten
- Begrenzung der Geschwindigkeit

Störgrößen:
- Strömung und Wind
- Abtreiben von der Bahn

Wie reagieren wir, wenn wir Bahn verlassen haben?

- Idealpfad und Epsilonschlauch vorhanden
- Regelung soll dafür sorgen, dass wir innerhalb des Schlauchs bleiben
- Zwischenpunkt festlegen um in Schlauch wieder zu kommen?
    - wir bräuchten viele Zwischenpunkte
    - Zwischenpunkte generieren
    - sinnvoll, den Schlauch nicht parallel zuführen
    - von Punkt B einen Kegel zu Punkt A, damit wir Punkt B genau erreichen
    - Punkt B ziemlich genau (Punktgenau) anfahren!

Wann werden Punkte der Bahnsteuerung aktualisiert?
- keine Aktualisierung
- Geschwindigkeit muss nicht 0 werden
- durch den Punkt B fahren
- nicht aktiv auf 0 fahren

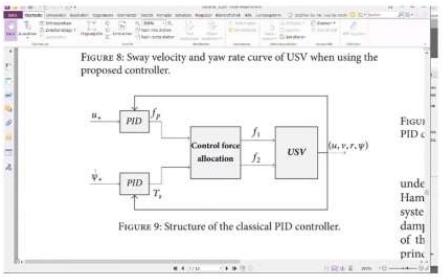
Regelgröße:
- Kurswinkel

Toleranzzone, wenn diese erreicht, dann aktualisiert?
- Abschätzung von Toleranzzone noch nicht möglich zu sagen (ausstehende Tests)

Wie genau ist das GPS?
- noch keine Aussage (ausstehende Tests)

Hinweis:
- Variable-Parameter in Regelung anlegen für Toleranzzone und GPS-Genauigkeit

Frage zu Reglermodell:



- Stellgrößen auf Geradeausfahrt regeln und Dreh-Änderungsrate?
- Stellgröße: Schub und Drehmoment, welches auf Schiff wirkt!
- Wir übergeben Schub und Ruderwinkel

Welche Kräfte wirken bei Motor 1 und Motor 2, wenn Schub und Winkel angegeben werden?
- 2 Stellgrößen: Schub und Winkel
- Schub: -32000 bis +32000
- Ruderwinkel: -1 bis +1 (linear)
    - entspricht Drehzahlverhalten beider Motoren
    - linker Motor läuft volle Kraft vorwärts, rechter Motor volle Kraft rückwärts
    - 0 - beide Motoren gleiche Drehzahl
    - Schub = 0 -> passiert nichts

    - Schub multipliziert zu 1 oder -1
    - voller Schub multiplizieren mit 1
    - schnelle oder langsame
    - Wenn winkel = 0 -> volle geradeAusfahrt

Welche Versuche werden demnächst gemacht?
- Schwimmfähigkeit
- Geschwindigkeiten
- folgende Parameter werden ermittelt::
    - Masse und Massenträgheitsmoment
    - Reibung, wenn Schiff um Schwerpunkt dreht -> Torsion
    - Reibung in Fahrtrichtung
    - Reibung quer zur Fahrtrichtung (Sprungfunktion)
    - Drehbewegung (Federkraftmesser)
    - Schub der Motoren:
    - angelegte PWM und aktuelle Geschwindigkeit
    - Drehzahl und Kraft

zu 4. Klärung offener Fragen (Team Sensorik und Team Regelung)

CAD-Datei von 3D-Druck-Teil für Platinenbefestigung?
- noch keine Befestigung
- Klammer für Befestigung

CAD-Datei von der Slave-Platine?
- ja gibt er frei

Konnektierung zwischen Slave und User-Unit? Leitungen?
- Anfang ganz normalen PIN-Header
- Konnektor J5 und J3
- Anschluss an Flachbandkabel (Finale)
- PIN-Header für Debug-Zwecke
- Kabel selbst konfektionieren

Thermische Vorgänge im Gehäuse zu beachten?
- wie viel Verlustleistung
- untergeordnete Rolle
- Schrauben Platinenbefestigung?
- keine
- Hutschiene mit Boxbefestigung

Vorgabe Platinenlayout?
- Lochrasterplatine oder Professionell?
- besser, professionell herzustellen
- Board von ST draufstecken
- zum Testen, einfach ganz normale selbstgefertigte Platine
- in Zukunft:
    - dann Layout von ST auf Platine übernehmen
    - für ersten Test -> Lochrasterplatine
    - bei Herm Franke löten @nachfragen zum Löten
- Slave mit Gehäuse zum rumspielen? @ Herm Franke nachfragen
- Kommunikation vorbereiten und ausprobieren
- Verbindungen checken
- Spannungsversorgung
    - Unser Board mit 5V

- eigene Spannungsregler integriert
- intern: 3,3 V
- Spannungsregler sind dafür da, um Spannung auszuregeln
- +- 0,25V - Spannungstoleranz von OPV
  - sollte eigentlich passen

Steuerleitungen CTS & RTS verwenden oder nicht (siehe PDF)?
- RTS und CTS verwenden
- auf Schnittstelle von Slave mit drauf
- wenn Buffer leer ist
  - FIFO voll -> eine Leitung auf 0 gezogen (RTS-Leitung)
  - wenn FIFO leer -> dann auf 1 (RTS-Leitung)
  - Ready to recieve -> auf 0 gezogen
    - wenn frei ist, dann auf 1 gelegt
  - RTS - ready to send
  - CTS- clear to send

Anmerkung von Prof. Grabow:
- Wie müssen das optimale Verhalten festlegen!
  - Grabow hat keine Erfahrung

Fragen von Team 2 -Sensorik

Slavespezifische ID für Module?
- müssen wir noch definieren

Wie oft Daten weiterschicken?
- alle 1s Daten schicken
- Bus refreshen alle 200ms

Welche Genauigkeit der Kompass-Daten:
- höchste Genauigkeit

Wass passiert, wenn CRC und ACR nicht ankommt?
- Behandlung?
- im ersten Moment senden
- wenn nochmals Fehler:
  - dann verwerfen und nächstes Paket verwenden
  - Gesamtfehlerbehandlung fehlt noch!
    - wenn Kabelbruch
    - dann mehrmals abfragen, dann Slave aus Bus entfernt
    - Sind neue Module im Bus? Abfragung

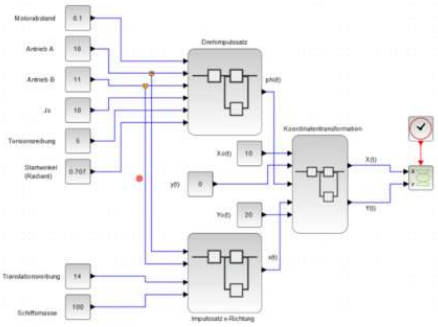
Frameaufbau und Framelänge!
- Framelänge mitsenden
  - n+6 -> Anzahl der Datenbyts eigentlich 7

RS485 und RS232! Muss Frame übereinstimmen?
- wir senden nur mit RS232
- Frameaufbau bleibt einfach so
- wird auf RS485 durchgereicht!

Wie wird die Bewertung/Note von diesem Modul festgelegt?
- Dokumentationen
- Protokolle
- Arbeitspläne
- Abschlussdokumentation

Abgabetermin:
- Projektabschluss:
- Projektdokumentation auch zum Abschluss vorliegen!

Anmerkung zur Regelung von Grabow:
- Ist in Simulation von Herrn Kirchner die Dynamik hinterlegt?
- Schiffsdynamik Modell wird erstellt
    - Vorstellung von Grabow der Schiffsdynamik



Bereitstellung der Blöcke von Grabow
- Antriebe A mit 10N
- Bahnkurve
- Corioliskraft

- Modell wird bereitgestellt
- SCILab-Modell
- Was wird noch eingebaut?
    - Wellen, Wind, Corioliskraft, Strömung
    - für Regelungsstrategie

Nächste Schritte / Arbeitsaufträge:

Was?	Wer?	Bis wann?

Protokoll 23.11.2020

Projekttitel:	Entwicklung und Implementierung der Regelungs- und Steuerungsstrategien
Projektnummer:	MechatronikProjekt_2021_Team2_Regelung
Projektleiter:	Muhammad Maaz Majid

Sitzung am:	23.11.2020, von 15:15 bis 16:15 Uhr
Ort der Sitzung:	Online über Skype
TeilnehmerInnen:	Muhammad Maaz Majid, Chen Zhang, Christoph In der Au, Sascha Halbgewachs, Martin Kirchner, Lukas Neumeister
Entschuldigt:	-
Nicht entschuldigt:	-

Inhalte der Sitzung:

1. Vorstellung Aufgaben- und Terminplanung von Maaz
2. Team Kommunikation Vortrag von Lukas und Christoph
3. Vortrag über Regelungsstrategie von Martin

Ergebnisse:

zu 1. Vorstellung Aufgaben- und Terminplanung von Maaz

Fragen in Maaz .ppt ergänzen
Lochrasterplatine: ja/nein

zu 2. Team Kommunikation Vortrag von Lukas und Christoph

Bussysteme.doc
- Führungsgrößen:
- Bekommen nur Punkt A und Punkt B
- Epsilon-Schlauch
- Registrierung -> das Punkt geändert wurde
- Wann wird B zu A und A zu B?
- Bahnkorrektur bestimmen
- Toleranzschlauch festlegen
- wenn Schlauch verlassen -> Strategie überlegen
- Regler und Parameter bestimmen
- Einstellkriterien
- viel abhängig von Bahnsteuerung -> Was brauchen wir? @Grabow fragen!
    - Wie verhält sich Punkt A und Punkt B?
    - Epsilon-Schlauch variable und/oder fix?
    - Wann werden Punkte aktualisiert? -> Wann sind Punkte erreicht?
        - Schiebt Punkt B sich immer weiter, z.B. für 100m? (dynamisches Verhalten)
    - Maximale Geschwindigkeit = Mittlere Geschwindigkeit zu nutzen? Um Abbremsvorgang zu vermeiden

zu 3. Vortrag über Regelungsstrategie von Martin

- Wir regeln Position:
    - Wir regeln Entfernung die soll auf 0 gehen, berechnen Zwischengrößen
    - Regelgröße ist Entfernung
    - Abbremsverhalten realisieren

- Was macht USV eigentlich?
  - nebenbei! nicht so wichtig!

- MATLAB -> Simulink
  - @ Sascha
  - Sascha hat eine Vorlage für die Simulation eines Schiffes
  - gute Simulation eines Schiffes
  - Regler -> zeitdiskreter PI-Regler

- Strategie der Implementierung
- Winkel ausrechnen zw. Punkt A und B
- GPS-Koordiante
- Bahnkorrektur, das Winkel nicht zu groß wird
- auf berechneten Winkel regeln
- Bussystem.docx
  - Epsilon-Schlauch ändert sich permanent
  - wenn in Hafen, dann kleiner

Nächste Schritte / Arbeitsaufträge:

Was?	Wer?	Bis wann?
Fragen stellen	Maaz Majid	25.11.2020
Programmaufbau (Interruptgesteuert, oder nur Polling?)	Martin Kirchner	noch offen!

Protokoll 20.11.2020

Projekttitel:	Entwicklung und Implementierung der Regelungs- und Steuerungsstrategien
Projektnummer:	MechatronikProjekt_2021_Team2_Regelung
Projektleiter:	Muhammad Maaz Majid

Sitzung am:	20.11.2020, von 09:30 bis 10:30 Uhr
Ort der Sitzung:	Online über Skype
TeilnehmerInnen:	Muhammad Maaz Majid, Chen Zhang, Christoph In der Au, Sascha Halbgewachs, Martin Kirchner, Lukas Neumeister
Entschuldigt:	-
Nicht entschuldigt:	-

Inhalte der Sitzung:

1. Vorstellung Projektplan von Maaz
2. Vorstellung Regelstrategie von Martin
3. Offene Fragen und ToDo

Ergebnisse:

zu 1. Vorstellung Projektplan von Maaz

Vorstellung Zeitplan
Verteilung Arbeitspakete
- Team Bussystem
- Regelungstechnisches Team
- Steuerstrategie -> Zusammen

Meilensteinplan vorstellen von Word-Dokument
- Projektplan vorstellen
- Aufgabenpakete
    - Aufbau, Aufgabenleiter
    - Teams

Projektstatusberichte erstellen

Vorstellung Aufgaben:
- Bussystem verstehen
- RS232-Protokoll und Schnittstelle
    - Serielle Kommunikation, ähnlich UART, nur auf anderen Logik-Pegel
- Hardware von Platine-Design
- Schnittstelle
- TTL interessiert uns nicht
- Aus Board kommen 3,3 V oder 5V raus
- Pegelwandler auf Slave-Modul
- Welche Register? Lesen/Schreiben? Was verwenden?

Software-Architektur festlegen @Martin

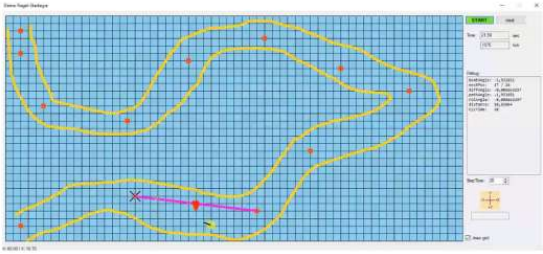
Slave-Platine @Hr. Franke
- Temperaturen?
- Kühlung?
- 3D-Druck-Teil? -> damit es nicht wackelt

@Martin
- Translatorische Bewegung
- Rotatorische Bewegung
- CAD-Modell vom Katamaran?
- Wir wissen wo Motoren sitzen!
- Datenformat umrechnen
- Team 1? weil 9 Personen
- 1 Byte für Minute, 1 Byte für sekunde
- 3 Bytes werden verwendet, siehe Bussystem.word
- Double -> 64
- Float, Single -> 32

zu 2. Vorstellung Regelstrategie von Martin

In der Nähe der Zone, des Zielpunktes
- Verringerter Schub
- Punkt angefahren:
  - Ist nächste Punkt der letzte?
  - Zwischenpunkt anfahren
  - Endpunkt anfahren, wenn es der letzte Punkt ist
  - Schubregelung -> beim Erreichen des Punktes = 0 Schub
- Endpunkt erreicht -> Stopp-Zustand

@Idee-Sascha
- Gerade Route
- Epsilon-Schlauch



- Zwischen 2 Punkten
- Punkte neu berechnen
- Bahnpunktliste auf BUS
  - Interne Berechnung der Punkte selber
- Unterschiedliche Klassen:
  - Anlegen
  - Abfahren
  - Hohes Gewässer
  - Von Bahnsteuerung gegeben
  - Wenn Punkte näher sind, dann langsamer fahren
  - Grenzwerte festlegen
  - Unterschiedliche Moden:
    - Langsame Fahrt im Hafen
    - Schnelle Fahrt auf See

zu 3. Offene Fragen und ToDo

Aufgabenverteilung
- Recherche für alle von Team Regelung
- STM-MicroElectronics
  - Nucleon-Board
  - Entwicklungsumgebung
    - Integrierte Editor ist kacke
  - Konfiguration von Board

Hr. Franke:
- Kabel, Leitungen von Slave zu uns
- Konnektierung zwischen Slave und User-Modul

Nächste Schritte / Arbeitsaufträge:

Was?	Wer?	Bis wann?
Wer macht Umrechnung der Daten?	Team 1/Team 2	noch offen!
Arbeitspakete planen	Maaz Majid	23.11.2020

Protokoll 18.11.2020

Projekttitel:	Entwicklung und Implementierung der Regelungs- und Steuerungsstrategien
Projektnummer:	MechatronikProjekt_2021_Team2_Regelung
Projektleiter:	Muhammad Maaz Majid

Sitzung am:	18.11.2020, von 09:30 bis 11:00 Uhr
Ort der Sitzung:	Online über BBB
TeilnehmerInnen:	Prof. Grabow, Dipl.Ing. Stefan Franke und alle M.Eng. Mechatoniker
Entschuldigt:	-
Nicht entschuldigt:	-

Inhalte der Sitzung:

1. Team 1 leitet Meeting und Vorstellung Präsentation von Michael
2. Vorstellung Präsentation von Maaz
3. Beantwortung Fragen von Team 2

Ergebnisse:

zu 1. Team 1 leitet Meeting und Vorstellung Präsentation von Michael

Hinweise von Hr. Grabow:
- Bussystem-Protokolle werden erweitert/aktualisiert in Moodle (ständig)
- in Dokumente reinschauen
- von unserem Modul in Slave-Platine einspeisen
- RS485 haben wir nichts zutun

Hinweise von Hr. Franke:
- Datenformat-Platinendesign
- Gerber-Format
  - Altium-Designer mit viele Bibliotheken
  - Eagle (bei Grabow), (nicht so viel bei Franke)
    - Eagle -> Studentenversion
- technologische Daten zur Platinenherstellung
- zw. Platinenformat und technologischen Format
- Bohrdurchmesser, Isolationswiderstand
- Schaltplan und Layout

zu 2. Vorstellung Präsentation von Maaz

Hinweise von Hr. Franke und Hr. Grabow:
- Slave-Board mit TTL-Wandler
- CMOS oder TTL, Spannung ausgeben
- PGA, DIP-Gehäuse?
- Schnittstelle zum Slave herstellen
- TTL-Schnittstelle verwenden
- Grundplatine entwerfen -> Steckverbinder mit Entwicklungsboard
- Verbindungsplatine mit Schnittstellen
- passive Platine entwickeln
  - Steckverbinder

zu 3. Beantwortung Fragen von Team 2

Einzelansteuerung von Motoren möglich oder über Motorregelung?
- möglich
- nur Angabe, wieviel gesteuert werden muss

Platine für Team 1 und Team 2 zusammen oder bekommt jeder ein eigenes Gehäuse?
- jedes Team ein Gehäuse
- Auflagen für 3D-Druck werden gedruckt, damit die Platine nicht wackelt
- Zuschnitt User-Unit = Größe Slave-Platine

Masse des Schiffs?
- nur zulässige Masse gegeben: 275,1 kg
- noch offen! Wird ermittelt von Prof.Grabow

Bekommen wir die Daten von der CD von Masterarbeit?
- Ja
- Streckenmodell komplett neu überarbeiten
- Reglermodell, PD-Regler

Welche Werte werden übergeben?
- Schub 100 bis -100?
- Ruder: Von 100 bis -100? Als Winkel angegeben?
- Wertebereich? Datentyp?
- @ E-Techniker-Gruppe beschäftigt sich damit?

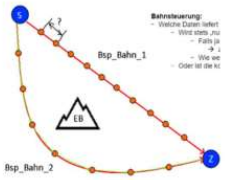
Word-Dokument 'Bussystem' -> Kapitel 'Datenblock'
- wird an alle Slaves übertragen und muss gelesen werden
- Adressbereiche in Tabelle
- Wie werden die Adressen in Bytes verschlüsselt?
- Umrechnung: Breitengrad in Grad, Minute, usw.

Wer macht Umrechnung?
- Abtastrate: 0,1 s bis 1 s

Welche Daten liefert die Bahnsteuerung? (Wird stets „nur" die nächste anzufahrende Pos. geliefert? Falls ja: Wann wird diese Pos. aktualisiert? z.B. wenn Pos. + Toleranz angefahren? Wie weit liegen die Punkte auseinander? Oder ist die komplette Bahn (alle Pos. inkl. Reihenfolge) abrufbar?
- Führungsgröße
  - eine Tabelle mit allen Punkten
  - aktueller Punkt und nächster Punkt
  - 2 Punkte gibt es nur:
  - Wie weit liegen Punkte auseinander?
  - (sehr enge: Punkte) -> Geschwindigkeit drosseln
  - weite Punkte auseinander -> Geschwindigkeit erhöhen
  - Gegensteuern?
  - Epsilon-Schlauch festlegen
  - Epsilon-Schlauch festlegen bei Erprobungen (1m, 2m oder 5m)?
  - dynamische Änderungen möglich (Fischemetze, Bojen, andere Schiffe)

- Korrektur beim Abdriften (z.B. durch Wind) vom Kurs nötig?
  - wenn vom Startpunkt abdriften zu Punkt A

- Fehler auf Bahn
- Unsere Aufgabe: auf dem Kurs zu bleiben
- Regelabweichung bestimmen
- solange Fehler in Toleranzschlauch, können wir ignorieren
- Fehler zu groß -> verlassen Toleranzbereich -> erst in Toleranzbereich zurück
- Möglichkeiten:
- Tangential in Toleranzband oder schräg zu Bahnsteuerung zurück
- Vorschlag: kürzesten Weg in Toleranzband reinfahren
- gelöstes Problem bei Bahnsteuerung in Literatur nachlesen
- sinnvolle Strategien sind schon in Literatur gelöst
- Grabow sendet Vorlesungssatz zu!



Wird „exakt" alle 100ms kommuniziert bzw. die Daten aktualisiert?
- wird noch ermittelt
- aber davon ausgehen, dass die Zeit sehr exakt ist

Nächste Schritte / Arbeitsaufträge:

Was?	Wer?	Bis wann?
Wer macht Umrechnung der Daten?	Team 1/Team 2	noch offen!
Epsilon-Schlauch festlegen!	noch offen!	noch offen!
Platine designen!		
Bestellung losschicken!		
3D-Druck-Teil von Hr. Franke? in CAD-Daten?		
Meeting-Protokolle, Übersicht von Michael?		
Protokolle an Grabow senden!	Maaz Majid	19.11.2020

Protokoll 16.11.2020

Projekttitel:	Entwicklung und Implementierung der Regelungs- und Steuerungsstrategien
Projektnummer:	MechatronikProjekt_2021_Team2_Regelung
Projektleiter:	Muhammad Maaz Majid

Sitzung am:	16.11.2020, von 15:30 bis 16:30 Uhr
Ort der Sitzung:	Online über Skype
TeilnehmerInnen:	Muhammad Maaz Majid, Chen Zhang, Christoph In der Au, Sascha Halbgewachs, Martin Kirchner, Lukas Neumeister
Entschuldigt:	-
Nicht entschuldigt:	-

Inhalte der Sitzung:

1. Vorstellung: Steuer und Regelungsstrategie von Masterarbeit-Chris + Fuzzy-Regelung @Sascha
2. Recherche Entwicklungsboards @Martin-Präsentation
3. Vorstellung Anforderungsliste @Christoph und Lukas
4. Bestellung der Bauteile
5. Zeitplan @Maaz

Ergebnisse:

zu 1. Vorstellung: Steuer und Regelungsstrategie von Masterarbeit-Chris + Fuzzy-Regelung @Sascha

Masse des Schiffs? Haben wir nicht!
- nur zulässige Masse: 275,1 kg

Antriebssystem -> HASWING-Motoren
Motorsteuerung direkt im Motor
es gibt ein Eingangssignal: 0 bis 5V
- über Spannungslevel, herausfinden was er gerade macht
- über Spannungslevel -> Motordrehzahl wird übergeben
- 3 Spannungslevel:
  - 1. Level: 0,5 bis 1,3V
  - 2. Level: 1,3 bis 2,2 V
  - 3. Level: 2,2 bis 4,5 V
- Annahme: linearer Anstieg des Drehzahlverlaufs

PWM-Steuersignale für Fernsteuerung
Motorsteuerung über PWM-Signal
Auf dem Wasser können wir als eine Ebene betrachten! @Frage von Martin
- Strecke, GPS-Position, -> Regelungsstrategie
  - 1. Ausrichten (MA Seite?)
  - dann fahren
  - Winkel mit 360°

SCILAB - Seite 87
- Betrachtung Schiffsmotor
- Was ist im Baustein drinne?
  - Übertragungsfunktion

zu 2. Recherche Entwicklungsboards @Martin-Präsentation

- Entwicklungs-Board @Präsentation von Martin
- Gleitkomma-Format? @Grabow?
- Längengrad-Format?
- Berechnungsmethode:
    - Wie genau?
    - Rechenleistung?
    - Entfernung der Punkte?
- Bahnsteuerung:
    - von Team-Bahnsteuerung?
    - durch Wind abdriften? auf Punkt A
    - dann neue Route
        - zurückkehren auf Kurs realisieren?
- Polygonzug:
    - einzelne Positionen
    - Punkt zu Punkt fahren
    - wenn ein Punkt erreicht, dann wird der nächste bekannt gegeben?!
        - nur Annahme
- Beispielhafter Programmcode
    - reiner PI-Regler
    - kein großer Rechenaufwand
- FPU: schnellere Auswertung (Floating Point Unit)
- Look-Up-Tabellen?
- Zukünftige ToDos:
- TTL-Logik alle Boards
    - +- 5V-Distanz
    - Konverter -> Pegelwandler
- Folie 13:
- github, gitlab -> Dokumentation von Quellcodes, Historie (wer hat was geändert?)
- Ausnahmebehandlung:
    - keine GPS-Daten?
    - Bus fällt aus?
- Festlegung:
- Nucleo-Entwicklungsboard

zu 3. Vorstellung Anforderungsliste @Christoph und Lukas

- Lüftungsschlitze für Kühlung?
- Simulation
- Anforderungsliste ergänzen:
- Auswahlmatrix
- Preis?
- Dokumentation der Plattform (Aufwand)

zu 4. Bestellung der Bauteile

2x Konverter - Maße Platine: 15x20mm? Verbindung mit Steckleitungen
- -5V bis +5V
- TTL zu RS-232

2x Entwicklungsboards Nucleo

zu 5. Zeitplan @Maaz

- Termineinhaltung schwierig
  - viele Randbedingungen
- Plan für die nächsten 2 Wochen bis Mitte Dezember
- Anforderung Strecke und Regelung
- Grobe Steuerstrategie
  - grobe Idee für Anforderung
  - Fuzzy-Regelung @Grabow vorgeschlagen
  - Simulink -> Fuzzy-Regelung
  - PID-Regler? Maximierungsprinzip?
  - Zustände:
    - Ausrichten
    - vom Kurs abweichen
- Korrektur-Strategie
  - Was liefert uns die Bahnsteuerung

- übernächste Woche: grobe Arbeitspaket-Verteilung
  - kleine Teams bilden
  - Bussystem
  - Regelungstechnisches Team
  - Steuerstrategie -> Zusammen

- Vorstellung bei Grabow
  - Teilaufgaben aufklären (Verantwortliche)
  - Kommunikation- Verantwortlicher
    - Code und Modul

- Ausarbeitung in kleinen Teams
- Regelung 2x:
  - Simulation in SciLab
  - in C implementieren

Nächste Schritte / Arbeitsaufträge:

Was?	Wer?	Bis wann?
Grobstrukturierung vorstellen, Meilensteine per PowerPoint Projektstrukturplan, Terminplan, Meilensteinplan	Maaz	18.11.2020
Fragen @Hr. Grabow und Hr. Franke: 1. Welche Werte werden übergeben? Schub 100 bis -100? Ruder: Von 100 bis -100? Als Winkel angegeben? Wertebereich? Datentyp? @ E-Techniker-Gruppe beschäftigt sich damit? 2. Einzelansteuerung von Motoren möglich oder über Motorregelung? 3. Platine für Team 1 und Team 2 zusammen oder bekommt jeder ein eigenes Gehäuse? a. Inventor-Zeichnung @Lukas 4. Masse des Schiffs? a. nur zulässige Masse gegeben: 275,1 kg 5. Bekommen wir die Daten von der CD von Masterarbeit? 6. Weitere Fragen in Martin's Präsentation	Gemeinsam	18.11.2020

Protokoll 13.11.2020

Projekttitel:	Entwicklung und Implementierung der Regelungs- und Steuerungsstrategien
Projektnummer:	MechatronikProjekt_2021_Team2_Regelung
Projektleiter:	Muhammad Maaz Majid

Sitzung am:	13.11.2020, von 14:00 bis 15:00 Uhr
Ort der Sitzung:	Online über Skype
TeilnehmerInnen:	Muhammad Maaz Majid, Chen Zhang, Christoph In der Au, Sascha Halbgewachs, Martin Kirchner, Lukas Neumeister
Entschuldigt:	-
Nicht entschuldigt:	-

Inhalte der Sitzung:

1. Auswertung/Zusammenfassung Protokoll vom 12.11.2020
2. Anforderungsliste
3. Weiteres Vorgehen

Ergebnisse:

zu 1. Auswertung/Zusammenfassung Protokoll vom 12.11.2020

Masterarbeit online von Katamaran

Datei `GPS-Rohdaten` in GoogleDrive von Michael Kleber

Längen und Breitengrad wird angegeben
Navigation ist nicht planar
Krümmung der Erde relevant?

Kompasswinkel und GPS Signale kriegen wir

Katamaran steuern
- Schub und Winkel
- Ähnlich, Roboterwettbewerb

Motoren einzeln ansteuern? Rechts und links?
- Franke sagte: nein! -> nochmals nachfragen!

Nur Lesen und Schreiben auf dem Bus

zu 2. Anforderungsliste

Spannungen 12V, 5V, 3.3V
Größe Platine
RS232-Schnittstelle
Datentypen

zu 3. Weiteres Vorgehen

Siehe nächste Schritte

Nächste Schritte / Arbeitsaufträge:

Was?	Wer?	Bis wann?
Streckenmodell recherchieren	Gemeinsam	16.11.2020
Entwicklungsboards recherchieren und vergleichen! z.B. Arduino MKR Zero, ESP32-Modul Überdimensioniertes Modul bevorzugen!?	Gemeinsam	16.11.2020
Anforderungsliste erstellen	Christoph, Lukas	16.11.2020
Masterarbeit: Steuer und Regelungsstrategie durchlesen + Fuzzy-Regelung -> Vorstellung	Sascha	16.11.2020
Grobstrukturierung vorstellen, Meilensteine per PowerPoint Projektstrukturplan, Terminplan, Meilensteinplan	Maaz	18.11.2020
Fragen @Grabow und Franke: 1. Welche Werte werden übergeben? Schub 100 bis -100? Ruder: Von 100 bis -100? Als Winkel angegeben? Wertebereich? Datentyp? 2. Einzelansteuerung von Motoren möglich oder über Motorregelung?	Gemeinsam	18.11.2020

Protokoll 12.11.2020

Projekttitel:	Entwicklung und Implementierung der Regelungs- und Steuerungsstrategien
Projektnummer:	MechatronikProjekt_2021_Team2_Regelung
Projektleiter:	Muhammad Maaz Majid

Sitzung am:	12.11.2020, von 08:00 bis 08:30 Uhr
Ort der Sitzung:	Online über BBB
TeilnehmerInnen:	Prof. Grabow, Dipl.Ing. Stefan Franke und alle M.Eng. Mechatoniker
Entschuldigt:	-
Nicht entschuldigt:	-

Inhalte der Sitzung:

1. Begrüßung und ergänzende Bemerkungen
2. Vorstellung Teams
3. Fragen und Diskussion
4. Hinweise zur Vorgehensweise von Prof. Grabow

Ergebnisse:

zu 1. Begrüßung und ergänzende Bemerkungen

Zusätzliche Bemerkungen für Team 2 Regelung:
- Slave-Modul mit Regler-Modul
- Antriebsmodul mit Slave-Modul über Bus ist schon vorbereitet
- Steuer und Regelalgorithmen erstellen
- RS232-Schnittstelle
- Programmierung der Steuer und Regelungstrategie
- Steuerdaten über Bus und Motor

zu 2. Vorstellung Teams

Vorstellung Teammitglieder und aktueller Stand

zu 3. Fragen und Diskussion

Datenblätter für Antriebe
- Wird noch zugestellt

Katamaran: Gewicht, Größe, Strömungstechnisches Verhalten? CAD-Daten? Techn. Aufzeichnungen? Ist es überhaupt für uns relevant?
- Ist relevant für uns!
- Masterarbeit erst fertig gestellt
- Mit Größe und Gewicht -> Massenträgheitsmomente berechnen
- Aus CAD-Modell
- Keine Strömungsverhältnisse aktuell vorhanden
- Wasserversuche in 14 Tagen -> Reibwerte werden gemessen -> Strömungsverhältnisse werden ermittelt und uns dann zugestellt

Modul-Plattform: Vorgaben? Empfehlungen? (vielleicht Arduino Nano, Atmega 8, etc.)
- Anforderung: RS232-Schnittstelle von Slave bereit gestellt

- Atmel, ARM-Controller, INTEL-Controller
- Wichtigste ist die RS232-Schnittstelle
- Erhalten Führungsgröße darüber
- Begrenzung:
    - nur 12V, 5V, 3.3 V
    - Größe der Platine
- GPS-Positionen
- Wie sind die aufbereitet? Bekommen wir noch
- Länge und Breitengrad
    - Winkel
- Kompasswinkel bekommen wir
- Über BUS: Koordinate wo wir hinfahren wollen
- Kurswinkel einregeln
    - Bahnkoordinate liegt im Norden, wir sind im Osten
    - Kurswinkel ausrechnen über Koordinate und Position

Wie soll Rückkopplung aussehen? Motorregler?
- Master steuert Motor?
- Unterschiedliche Steuersignale
- Hardware:
    - 2 Motoren
    - Jeder Motor wird von Slave-Schnittstelle Vollgas vorne und nach hinten
    - 2 Analogsignale, die an Motor gesendet werden
    - Über Bus-Schnittstelle nehmen wir 2 Signale auf
    - Welche Signale:
    - Keine Einzelansteuerung möglich
    - Nehmen zwei Signale auf
    - Schubachse und Drehachse = voll und Drehachse = große Kurve
    - Steuerache (links(rechts)
    - Schubachse * Steuerachse =
    - Schub = 0…linke Motor Vollgas
    - Mit Schub = Kurve fahren
    - Signale von Controller und rechnen für Motoren um
    - Schubachse zum stellen
    - Drehachse zum stellen
    - Getrennt möglich
    - Testfahrt
    - Grabow: Ein Motor und ein Ruder
    - Rudereinschlag, Winkel

zu 4. Hinweise zur Vorgehensweise von Prof. Grabow

Hinweise von Prof. Grabow
- Modell notwendig vom Schiff
- Welche Strecke?
- Regelungstechnisches Modell
- Mechanik/Kinematik -> regelungstechnisches Modell aufbauen
- Simulationsmodell erstellen über Software
    - Simulink? MATLAB? SCILAB? LTSpice?
- Einfaches Modell
- P-Regler, nächster Schritt dann PI-Regler
- Reale Dinge -> einbauen
- Exakte Führungsgröße: fester Koordinatenpunkt
- Große Störeinflüsse: Asymmetrie, Wind, Wellen

- Fuzzy-Regelung über Modell .... Epsilon-Schlauch
- Auf Modell übertragen

MicroController beschaffen
- Frühzeitig Bescheid sagen -> Einkauf und Kassenschluss
- Bis nächste Woche

Nächste Schritte / Arbeitsaufträge:

Was?	Wer?	Bis wann?
Vorstellung abwechselnd, Team 1 beginnt als Moderator, versendet Einladung und Orga für Meeting	Team 1	18.11.2020
Grobstrukturierung vorstellen, Meilensteine per PowerPoint	Team 2	18.11.2020
Beschaffung MicroController	Team 2	18.11.2020

Protokoll 09.11.2020

Projekttitel:	Entwicklung und Implementierung der Regelungs- und Steuerungsstrategien
Projektnummer:	MechatronikProjekt_2021_Team2_Regelung
Projektleiter:	Muhammad Maaz Majid

Sitzung am:	09.11.2020, von 15:30 bis 16:30 Uhr
Ort der Sitzung:	Online über Zoom
TeilnehmerInnen:	Muhammad Maaz Majid, Chen Zhang, Christoph In der Au, Sascha Halbgewachs, Martin Kirchner, Lukas Neumeister
Entschuldigt:	-
Nicht entschuldigt:	-

Inhalte der Sitzung:

1. Wahl des Projektleiters
2. Festlegung der Online-Plattform für Austausch
3. Ideen und Diskussion
4. Details und Rückfragen

Ergebnisse:

zu 1. Wahl des Projektleiters

Muhammad Maaz Majid

zu 2. Festlegung der Online-Plattform für Austausch

Google Drive

zu 3. Ideen und Diskussion

6.Semester: Roboterwettbewerb, Platine, Hardware, Software (siehe Google Drive)
Sascha: früheres Projekt mit Arduino, Motorsteuerung, PWM, Rückkopplung, Motorregler
Sascha: Projekt mit autonomen Boot im Fluss
Christoph: viele Umwelteinflüsse und Störgrößen auf dem Meer

zu 4. Details und Rückfragen

Details und Rückfragen
- Datenblätter für Antriebe
- Katamaran: Gewicht, Größe, Strömungstechnisches Verhalten? CAD-Daten? Techn. Aufzeichnungen? Ist es überhaupt für uns relevant?
- Modul-Plattform: Vorgaben? Empfehlungen? (vielleicht Arduino Nano, Atmega 8, etc.)
- Wie soll Rückkopplung aussehen? Motorregler?
- Ein paar Worte zu Slave-Regelung: Seite 18 und 20 bei Aufgabestellung.pdf

Nächste Schritte / Arbeitsaufträge:

Was?	Wer?	Bis wann?
Details und Rückfragen klären mit Prof. Grabow und Herrn Franke	Gemeinsam	12.11.2020

Protokoll 04.11.2020

Projekttitel:	Mechatronik Projekt - Auftaktbesprechung
Projektnummer:	MechatronikProjekt_2021
Projektleiter:	-

Sitzung am:	04.11.2020, von 09:30 bis 11:00 Uhr
Ort der Sitzung:	Online über BBB
TeilnehmerInnen:	Prof. Grabow, Dipl.Ing. Stefan Franke und alle M.Eng. Mechatoniker
Entschuldigt:	-
Nicht entschuldigt:	-

Inhalte der Sitzung:

1. Vorstellung Projekt
2. Aufgaben zur nächsten Besprechung

Ergebnisse:

zu 1. Vorstellung Projekt

@Prof. Grabow
Thema:
- Einweisung in das Projekt
- Erklärung der Aufgaben

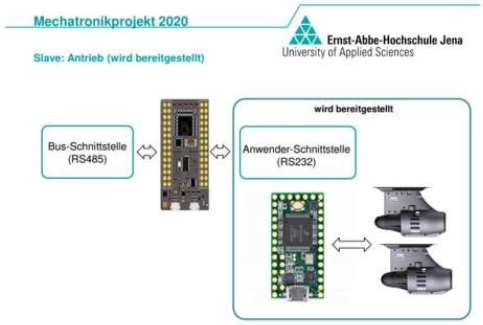
Slave-Module:
- Antrieb des Katamarans
- Aktives Slave-Modul
- Daten senden an Master
- 2 Elektromotoren
- Eigene Stromversorgung
- Aktive Steuerung

Schnittstellen:
- RS485-Protokoll mit RS485-Schnittstelle ist verriegelt
- RS485: Stromversorgung
- RS232 - Kommuniziert mit RS232-Schnitstelle
- Antriebsstrom für Elektromotoren wird separat bereitgestellt

Aufgabe:
- Hardware entwickeln
- Software entwickeln
- Schnittstellen bedienen
- Hardware so aufgebaut werden, das:
    - An Slave-Modul passt
    - Genormte Spannungsschnittstellen
    - Genormte Schnittstellen

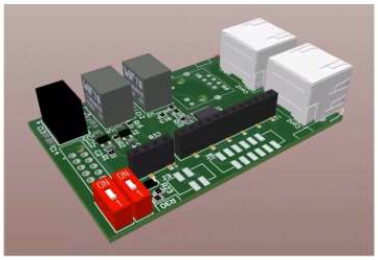
- Positionsdaten aufarbeiten:
    - Positionssensor = Länge und Breite
    - Kompass = ...°-Zahl

- Daten für Bahnsteuerung erfassen
  - Gerade von zwei Punkten
  - Dynamische Änderung von Bahndaten verarbeiten
    - Containerfrachter
    - Keine Durchfahrt
- Regelungsaufgabe
  - Regelung des Kurswinkels
    - Punkt zu Punkt fahren
  - Regelung der Geradeausfahrt
    - Wellen und Wind lassen das Boot abtreiben
- Start, Ziel und Zwischenpunkte anfahren



- Ansteuerung für beide Antriebsmotoren
- Ansteuerung PWM
  - Analoge Ansteuerspannung
  - Volle Fahrt Rückwärts und volle Fahrt Vorwärts

@ Hr. Franke:
- Labor nutzen
- Platinen bestücken
- Software
  - Eagle
  - Altium
- Eagle benutzen
- Anmeldung über Projektleiter
- Interface der Slave-Baugruppe:
  - Versorgung mit 24V mit externer Batterie
  - 3 Versorgungsspannungen: 12V, 5V und 3.3V



- 2x RS485-Bus-Anschluss
  - Senden der Daten an Micro-Controller
- MicroController:
  - Umwandlung RS485-Protokoll in RS232-Protokoll
- RS232-Protkoll
  - Erhalten wir über große Leiste über Flusssteuerung
  - Kurze Leiste: 12V, 5V und 3.3V
  - Max. 1A
- Vorgabe RS232: 3.3 oder 5V
- Jeder Aktor erhält Speicherbereich
  - Sensor darf schreiben
  - Aktor darf lesen

- Gehäuse:
  - 1. Seite mit Slave-Platine
  - 2. Seite können wir verwenden (siehe CAD-Daten)

zu 2. Aufgaben zur nächsten Besprechung

Mechatronik Projekt (Studiengangskürzel) WiSe 20/21 » Foren » Ankündigungen » Projektunterlagen

Projektunterlagen
von Jörg Grabow - Donnerstag, 5. November 2020, 08:06

Die Aufgabenstellungen und Projektunterlagen stehen bereit. Das nächste Meeting findet am 12.011.2020 von 08:00 Uhr bis 09:00 Uhr statt.

Agenda

- Vorstellung der Teams mit den jeweiligen Teamleitern
- Abstimmung zu den nächsten Teilaufgaben
- Diskussion

Diskussion im Forum zeigen

Nächste Schritte / Arbeitsaufträge:

Was?	Wer?	Bis wann?
Vorstellung Teams	Jedes Team	12.11.2020
Abstimmung Teilaufgaben	Jedes Team	12.11.2020
Diskussion	Jedes Team	12.11.2020