Pflichtenheft

Projekt: Spindel mit Linearführung

Version 0.4

Autor des Dokuments	Thorsten Kimmel	Erstellt am	16.02.2025
Dateiname	Pflichtenheft.doc		
Seitenanzahl	30	© 2025 Thorsten Kimmel *DHBW-Stuttgart*	**Vertraulich!**

Historie der Dokumentversionen

Version	Datum	Autor	Änderungsgrund / Bemerkungen
0.1	01.01.2025	Thorsten Kimmel	Ersterstellung
0.2	28.03.2025	Thorsten Kimmel	Anpassung LEDs und UART-RX-IRQ-Wunschkriterium
0.3	14.09.2025	Thorsten Kimmel	Bewertung und Kleinigkeiten
0.4	23.08.2026	Thorsten Kimmel	Anpassung für STM32H7 sowie Anpassung der Bewertung

Inhaltsverzeichnis

Historie der Dokumentversionen 2
Inhaltsverzeichnis 3
1 Einleitung 5
1.1 Allgemeines 5
1.1.1 Zweck und Ziel dieses Dokuments 5
1.1.2 Ausgangssituation 5
1.1.3 Musskriterien 6
1.1.4 Wunschkriterien 6
1.1.5 Bereitstellung von SW-Komponenten durch den Kunden 9
1.1.6 Projektstruktur und Ablagekonzept 9
1.2 Verteiler und Freigabe 10
1.2.1 Verteiler für dieses Lastenheft 10
Konzept und Rahmenbedingungen 11
1.3 Systemanforderungen 11
1.3.1 Mechanische Anforderungen 11
1.3.2 Elektrische Anforderungen 11
1.3.3 Baugruppen 12
1.4 Ressourcen 13
1.4.1 Dokumentation 13
1.4.2 Hardware 13
1.5 Übersicht der Meilensteine 13
2 Beschreibung der Anforderungen 14
2.1 Anforderungen an die Bewegungssteuerung 14
2.2 Anforderungen an die Spindelsteuerung 14
2.3 Anforderungen an die Kommunikation und Steuerung 14
2.4 Sicherheitsanforderungen 14
2.5 Betriebsbedingungen 15
2.6 Testanforderungen 15
2.7 Anforderung – Spindelbetrieb – Start 16
2.7.1 Beschreibung 16
2.7.2 Testhinweise 16
2.8 Anforderung – Spindelbetrieb – Stopp 17
2.8.1 Beschreibung 17
2.8.2 Testhinweise 17
2.9 Anforderung – Spindelbetrieb – Status 18
2.9.1 Beschreibung 18
2.9.2 Testhinweise 18
2.10 Anforderung – Schrittmotor – Reset 19
2.10.1 Beschreibung 19
2.10.2 Testhinweise 19
2.11 Anforderung – Schrittmotor – Referenzfahrt 20
2.11.1 Beschreibung 20
2.11.2 Testhinweise 20
2.12 Anforderung – Schrittmotor – Bewegung 21
2.12.1 Beschreibung 21

2.12.2 Testhinweise 21
2.13 Anforderung – Schrittmotor – Positionsabfrage 22
2.13.1 Beschreibung 22
2.13.2 Testhinweise 22
2.14 Anforderung – Schrittmotor – Statusabfrage 23
2.14.1 Beschreibung 23
2.14.2 Testhinweise 23
2.15 Anforderung – Schrittmotor – Konfigurationsparameter Lesen 24
2.15.1 Beschreibung 24
2.15.2 Testhinweise 24
2.16 Anforderung – Schrittmotor – Konfigurationsparameter Schreiben 25
2.16.1 Beschreibung 25
2.16.2 Testhinweise 25
2.17 Anforderung – DigitalTwin – Windows Simulator 26
2.17.1 Beschreibung 26
2.17.2 Testhinweise 26
2.18 Anforderung – Master/Slave Timer für PWM – 2 Timer für den Stepper 27
2.18.1 Beschreibung 27
2.18.2 Testhinweise 27
2.19 Anforderung – Unit Tests für die Stepper-Implementierung 28
2.19.1 Beschreibung 28
2.19.2 Testhinweise 28
2.20 Anforderung – Lib6474 in asynchronem Betrieb implementieren 29
2.20.1 Beschreibung 29
2.20.2 Testhinweise 29
Bedienungskonzept 30
2.21 Allgemeines Zustandsdiagramm 30
2.22 Schrittmotorzustandsdiagramme in Abhängigkeit der implementierten Anforderungen 30
3 Freigabe / Genehmigung 30

1 Einleitung

Dieser Abschnitt hat die Aufgabe als eine Art Einleitung zu dienen. Es soll ein kurzer Umriss über Ziel und Motivation des Gesamt- und des Teilprojektes dargestellt werden. Beschrieben wird die Hauptaufgabe des Systems. Wichtig ist, den Grund für die Systementwicklung (Probleme oder Geschäftsidee) und damit ihre Ziele herauszuarbeiten.

Das vorliegende Pflichtenheft enthält die gestellten funktionalen sowie nicht-funktionalen Anforderungen an das Produkt. Es dient als Basis für die Ausschreibung und Vertragsgestaltung und bildet somit die Vorgabe für die Angebotserstellung. Kommt es zwischen Auftragnehmer und Auftraggeber zu einem Vertragsabschluss, ist das bestehende Pflichtenheft rechtlich bindend. Alle zuvor zwischen Auftraggeber und Auftragnehmer getroffenen Absprachen verlieren in der Regel durch das Pflichtenheft ihre Gültigkeit – sofern hier nichts Gegenteiliges vermerkt ist. Mit den Anforderungen werden die Rahmenbedingungen für die Entwicklung festgelegt, die vom Auftragnehmer im Pflichtenheft detailliert ausgestaltet werden.

Im Rahmen der Vorlesungseinheit werden in gewissen Abschnitten Kundenvorgaben als Musskriterien (harte Anforderungen) und als Wunschkriterien (Kann-Anforderung oder optionale Anforderungen) formuliert. Zum bestehen des Modulabschnitts müssen die Musskriterien erfüllt sein. Das bestehen bedeutet 50% der erzielbaren Punkte. Dies entspricht einer Note von 4.0.

Die insgesamt erzielbaren Punkte für dieses Projekt liegen bei 130 Punkte. Wobei lediglich 100 Punkte für eine 1,0 erzielt werden müssen. Somit muss nicht jede Optionale Anforderung erfüllt werden!

1.1 Allgemeines

1.1.1 Zweck und Ziel dieses Dokuments

Dieses Pflichtenheft beschreibt ein Produkt bestehend aus einer Spindel sowie einer Linearführung mit Schrittmotor zur genauen Positionierung der Spindel. Die Spindel ist in Drehrichtung und Drehzahl variierbar und dient zur späteren Gravur und Kennzeichnung von beliebigen Werkstücken entlang einer festen Achse.

Das Ziel dieses Projekts ist die Entwicklung und Herstellung einer Software für eine Gravierspindel mit einem Linearantrieb. Der Linearantrieb wird aus einem Schrittmotor bestehen, der die lineare Bewegung steuert, während die Spindel selbst durch einen DC-Motor betrieben wird. Das System wird für die präzise Gravur in Materialien wie Metall, Kunststoff und Holz verwendet.

Die Bedienung des Produkts erfolgt mithilfe einer seriellen Schnittstelle sowie einem definierten Protokoll und Befehlssatz. Dieses Dokument beschreibt die Anforderungen an die einzelnen Features des Produkts sowie dessen Schnittstellen.

1.1.2 Ausgangssituation

Der Kunde ist ein Betreiber einer großen Produktionsanlage verschiedenster Import- und Exportwaren, welche Fertigungstechnisch veredelt werden. Eine Produktionslinie übernimmt die Aufgabe linearer Gravuren unterschiedlicher Tiefen auf Materialien wie Holz, Stein, Stahl, Aluminium und Kunststoff. Für die Erweiterung und den Ausbau seiner Produktionskapazitäten benötigt der Kunde weitere Graviersysteme bestehend aus einer Spindel und einer Linearführung.

1.1.3 Musskriterien

Das Produkt muss mindestens die Kundenseitig bereitgestellten Bibliotheken LibSpindle, LibRTOSConsole und LibL6474 implementieren und den Minimalbefehlssatz folgender API-Befehle unterstützen.

Die genaue Beschreibung und das Verhalten der einzelnen API-Befehle können der Interface-Spezifikation entnommen werden.

Die Anzahl der Punkte pro Feature lassen sich im Kapitel Beschreibung der Anforderungen detailliert entnehmen.

API-Befehl	Unterbefehl	Rückgabewerte	Weitere Anmerkungen
stepper	move <AbsPos>	„Ok" oder „Fail"	Siehe Interface-Spezifikation. In Kombination mit anderen Anforderungen einsetzbar.
stepper	reference	„Ok" oder „Fail"	Referenzfahrt. In Kombination mit anderen Anforderungen einsetzbar.
stepper	reset	„Ok" oder „Fail"	Stepper komplett Zurücksetzen und re-initialisieren.
spindle	start <RPM> stop status	„Ok" oder „FAIL" für Status folgen weitere Rückgabewerte	Siehe Interface-Spezifikation
capability		Bitfolge + „OK" oder „FAIL"	

Der Kunde stellt ferner eine SpecTest und/oder eine Ethernet Anwendung zur Prüfung der Spezifikation und Funktion des Produkts bereit. Die oben genannten Features müssen durch diesen Test fehlerfrei ausgeführt werden können, um eine vollständige Punktebewertung zu erzielen!

1.1.4 Wunschkriterien

Die Erfüllung weiterer Anforderungen mit optionalem Charakter dient zur Verbesserung der Produktbewertung und erhöht die Zufriedenheit des Kunden, da das Produkt vielfältiger und zielsicherer einsetzbar wird. Die Erfüllung optionaler Anforderungen (Wunschkriterien) verbessert auch die erzielbaren Punkte für das Modul. Die folgenden Wunschkriterien können teilweise oder vollständig umgesetzt werden.

Die genaue Beschreibung und das Verhalten der einzelnen API-Befehle können der Interface-Spezifikation entnommen werden.

Die Anzahl der Punkte pro Feature lassen sich im Kapitel Beschreibung der Anforderungen detailliert entnehmen.

API-Befehl	Unterbefehl	Rückgabewerte	Weitere Anmerkungen
stepper	move <RelPos> -r	„Ok" oder „Fail"	Relative Position. In Kombination mit anderen Anforderungen einsetzbar.
stepper	move <AbsPos> -a	„Ok" oder „Fail"	Asynchrone Ausführung. In Kombination mit anderen Anforderungen

			einsetzbar.
stepper	move <AbsPos> -s <Speed_in_mm/min>	„Ok" oder „Fail"	Bewegung mit einer vorgegebenen Geschwindigkeit. In Kombination mit anderen Anforderungen einsetzbar. Die Punkte für dieses Feature fallen großzügiger aus, wenn eine ausreichende Granularität erzielt wird!
stepper	reference -e	„Ok" oder „Fail"	Power Outputs aktiv lassen nach Referenzfahrt. In Kombination mit anderen Anforderungen einsetzbar.
stepper	reference -t <time>	„Ok" oder „Fail"	Referenzfahrt mit Timeout. In Kombination mit anderen Anforderungen einsetzbar.
stepper	reference -s	„Ok" oder „Fail"	Referenzfahrt überspringen. In Kombination mit anderen Anforderungen einsetzbar.
stepper	position	„Ok" oder „Fail" und Positionswert	Gibt die aktuelle Position zurück.
stepper	cancel	„Ok" oder „Fail"	Bricht die aktuelle Bewegung ab. Nur mit Asynchroner Operation sinnvoll!
stepper	status	„Ok" oder „Fail" Weitere Rückgabewerte	Abfrage des aktuellen Zustandes.
stepper	config powerena [-v 0|1]	„Ok" oder „Fail" Rückgabewert bei Lesen	Abschalten oder Anschalten des Power Outputs, ohne „-v" auch Rücklesbar. Ändert die Allgemeine Hochschaltsequenz des Produkts!
stepper	config <parameter> [-v <value>]	„Ok" oder „Fail" Rückgabewert bei Lesen	Lesen oder Schreiben weiterer Parameter. In Kombination mit anderen Anforderungen einsetzbar. Hierunter fallen mehrere Parameter, die jeweils Einzelpunkte pro Parameter erzielen
Einbau der LibL6474 im asynchronen Betriebsmodus	Hochgradig empfohlen für „move -a" und „cancel" Anforderung	Ändert das Designpattern des Codes!	Diese Anforderung muss dem Kunden bei der Abnahme vorgestellt werden, es ist über das Interface nicht zwangsweise erkennbar, ob dieses Feature umgesetzt wurde.

Den PWM Timer mit Gated Enable eines anderen Timers (Master/Slave Timer) realisieren.	Dieses Feature ist für die korrekte Geschwidnigkeitsregelung bei maximaler Geschwindigkeit unbedingt zu empfehlen. Diese Implementierung wird für „move -s" empfohlen	Erfordert die korrekte Konfiguration und Behandlung der Timer Interrupts in korrekter Art und Weise	Diese Anforderung muss dem Kunden bei der Abnahme vorgestellt werden, es ist über das Interface nicht zwangsweise erkennbar, ob dieses Feature umgesetzt wurde.
Einbau des Designs in die vom Kunden bereitgestellte WIN32-Simulationsumgebung (DigitalTwin)	Für HomeOffice-Entwicklung empfohlen, wenn keine HW vorhanden ist. Ferner sind auch einige weitere Debugmöglichkeiten im Simulator gegeben!	Erfordert den Umgang mit Visual Studio 2022 (min Community Edition) sowie einer möglichen Anpassung des HAL-Simulationscodes für den gegebenen Anwendungsfall. Nur unter Windows nutzbar	Diese Anforderung muss dem Kunden bei der Abnahme vorgestellt werden, es ist über das Interface nicht zwangsweise erkennbar, ob dieses Feature umgesetzt wurde.
Eigene Designimplementierung ist in eine sinnvolle Bibliothek gekapselt und kann mit LibCMocka unter Windows Unit getestet werden!	Erhöht die Ausführungssicherheit / Korrektheit des Produkts und wird als Nachweis vom Kunden in Form eines Testprotokolls akzeptiert	Erfordert den Umgang mit Visual Studio 2022 (min Community Edition) sowie einer möglichen Anpassung des Designpatterns sowie den Umgang mit LibCMocka. Nur unter Windows nutzbar	Diese Anforderung muss dem Kunden bei der Abnahme vorgestellt werden, es ist über das Interface nicht zwangsweise erkennbar, ob dieses Feature umgesetzt wurde.
Status LEDs für den Schrittmotor Regler implementieren	--	--	Grüne LED, wenn Betriebsbereit. Blaue/Gelbe LED 1Hz blinken, wenn aktive Treiber Rote LED, wenn Fehler
Die asymmetrische Kommunikationsbibliothek mit openAMP für Netzwerkfunktion ins Design einbinden	Auf dem zweiten Kern (Cortex M4) ist ein Netzwerkstack implementiert, welchen man mit OpenAMP (LibETHProducer) ansprehcen kann	--	Das Programm für den zweiten Kern muss ohnehin geflasht werden!
Testprotokoll für die serielle Schnittstelle und die Funktion	Die Funktionen werden mit dem Kundentool getestet, ein Protokoll wird beigefügt	--	Das Protokoll erhöht den Qualitätsstandard und führt bei erfolgreichen Tests zu einer erhöhten Punktzahl

Der Kunde stellt auch für die Wunschanforderungen eine SpecTest Anwendung zur Prüfung der Spezifikation und Funktion des Produkts bereit. Die oben genannten Features müssen durch diesen Test fehlerfrei ausgeführt werden können, um eine vollständige Punktebewertung zu erzielen!

Für das Erreichen der Bestnote sind 100 Punkte zu erzielen. Dies ist somit eine 100% Bewertung und entspricht der Note 1,0. Jeder weitere Punkt über 100% kann nur gewertet werden, wenn ein anderes implementiertes Feature wegen eines Fehlverhaltens mit 0 Punkten bewertet wird. Mit anderen Worten. Man kann nicht mehr als die maximal mögliche Punktzahl erreichen, allerdings sichern weitere Punkte die Garantie, auch im Falle von Produktfehlern, noch immer 100% der Punkte zu erhalten!

1.1.5 Bereitstellung von SW-Komponenten durch den Kunden

Der Kunde hat bereits bestehende Steuerungseinheiten in seinen Anlagen und fordert aus diesem Grund den Einsatz seiner bereitgestellten Bibliotheken, um sowohl die Produktentwicklungszeit zu verkürzen aber auch um die Schnittstellenspezifikation so exakt wie möglich einzuhalten. Besonders die Nicht-funktionalen Anforderungen wie die Laufzeit möchte der Kunde damit in dem ihm möglichen Maße sicherstellen, indem er auf getestete und integrierte Bibliotheken zurückgreift.

Nachfolgend ist die Liste der Bibliotheken und ihre Abhängigkeiten aufgelistet. Weitere Informationen und der Stand der Bibliothek werden ebenfalls geführt.

Name	**Code-Doku vorhanden**	**Unit-Tests vorhanden**	**Abhängigkeiten**	**Weitere Anmerkungen**
LibL6474	HTML	Ja	Keine	Synchron oder Asynchron nutzbar, Support für RTOS MultiThreading
LibSpindle	HTML	Nein	LibRTOSConsole, FreeRTOS Kernel 11	Strenge Integration von FreeRTOS und LibConsole
LibRTOSConsole	HTML	Nein	FreeRTOS Kernel 11	Strenge Integration von FreeRTOS, WIN32 Simulationssupport
LibEthProducer	HTML	Nein	FreeRTOS Kernel 11 OpenAMP	Producer um mit dem Cortex M4 Netzwerkstack zu kommunizieren

1.1.6 Projektstruktur und Ablagekonzept

Die Ordnerstruktur dieses Projekts ergibt sich wie folgt:

- Stepper: Dieser Ordner enthält das STM32 Projekt samt CubeMX Projekt zur Generierung des HAL-Codes. Die Implementierung des MCU-Projekts findet dort statt
- Doc: Dieser Ordner enthält Dokumentationen, Schaltpläne, Folien und Vorlesungsdokumente sowie Blockdiagramme und Pinlisten. In diesem Ordner sind die wichtigsten Allgemeinen Informationen zu den Bauteilen dem Projekt und einigen Lesungsinhalten enthalten.
- Libs: Dieser Ordner enthält die Bibliotheken für Testing, für die Spindel, die RTOS-Konsole sowie den Stepper Treiber. Innerhalb der Bibliotheksunterverzeichnisse finden sich ferner die zugehörigen Source-Dateien, Header, ggf. Binaries und eine separate Code-Dokumentation zur Bibliothek.
- Simulation: Dieser Ordner enthält ein Win32 Visual Studio MSVC Simulatorprojekt, welches genutzt werden kann, um die eigene Lösung unter einem x64 Windows PC in gewissen Teilen in Betrieb zu nehmen bzw. in groben Zügen zu implementieren, ohne eine HW nutzen zu müssen.
- Spec: Der Spec Ordner enthält dieses Pflichtenheft sowie die API-Interface-Dokumentation, welche für die Konsolenbefehle eingehalten werden muss.
- Test: Dieser Ordner enthält eine Anwendung (SpecTest-App), welche zum Test der Implementierten Features genutzt wird. Diese Anwendung prüft das Verhalten und die

Konsolenbefehle auf ihre Gültigkeit. Sie kann den Simulator als Testobjekt nutzen oder direkt das STM32 Target!

1.2 Verteiler und Freigabe

1.2.1 Verteiler für dieses Lastenheft

Rolle / Rollen	Name	Telefon	E-Mail	Bemerkungen
Projektleiter	Thorsten Kimmel		Thorsten.kimmel@lehre.dhbw-stuttgart.de	

Konzept und Rahmenbedingungen

1.3 Systemanforderungen

1.3.1 Mechanische Anforderungen

1.3.1.1.1 Linearantrieb

- **Typ**: Schrittmotor
- **Baugröße**: NEMA 17
- **Drehmoment**: 0,4 Nm bei Phasenstrom von ~1,3A (Bei kleinen Testmotoren im Labor nicht mehr als 0,6A)
- **Winkelschritte Motor**: 200 = 1,8° im Vollschrittbetrieb, 3200 = 0,1125° im 16er Mikroschrittbetrieb, Die kleinen Testmotoren im Labor haben 400 Schritte im Vollschritt
- **Spindelsteigung**: 4mm
- **Auflösung**: mechanisch 0,01 mm, elektrisch 0,02 mm im Vollschritt; 0,00125mm im 16er Mikroschrittbetrieb
- **Steuerung**: Mikroschritte (16 Schritte)
- **Bauform**: Linearführung mit Gewindetrieb und Schrittmotor

1.3.1.1.2 Spindel

- **Typ**: DC-Motor
- **Drehzahlbereich**: 1600 - 9.000 U/min bei 24V Versorgungsspannung
- **Drehmoment**: 0,3 Nm
- **Baugröße**: 42 mm Durchmesser
- **Lagerung**: Kugellager für hohe Präzision und Lebensdauer
- **Spindelmaterial**: z. B. gehärteter Stahl
- **Werkzeugaufnahme**: ER11 mit wechselbaren Spannzangen

1.3.2 Elektrische Anforderungen

1.3.2.1.1 Stromversorgung

- **Spannung**: 12-28 V DC, Optimaler Betrieb bei 12V oder 24V gewünscht
- **Strombedarf**: bis zu 1,5A für Schrittmotor, 2A für DC-Motor

1.3.2.1.2 Steuerungssystem

- **Steuerung**: Steuerung erfolgt mithilfe einer Firmware auf einem STM32F746/STM32H755 Discovery Board und einer Kundenseitig bereitgestellten Bibliothek für eine serielle Konsole, der Spindelsteuerung sowie der Schrittmotorsteuerung

- **Kommunikation**: RS232 optional ETH
- **Feedback-System**: Schrittmotor: Open-Loop, DC-Motor: Open-Loop

1.3.3 Baugruppen

- **Rahmen**: Aluminiumrahmen mit hoher Steifigkeit zur Minimierung von Vibrationen
- **Linearantriebsystem**: Linearführung und Trapezgewindespindel mit Anti-Backlash Lagerung für präzise Positionierung
- **Kühlung**: Luftkühlung für den DC-Motor, um Überhitzung zu vermeiden
- **Schutz**: IP54 Schutzklasse gegen Staub und Spritzwasser

1.4 Ressourcen

1.4.1 Dokumentation

- **Datenblätter**: werden für den Schrittmotor, das NUCLEO-Entwicklungsboard, das Stepper-Shield und die H-Brücke durch den Kunden bereitgestellt.
- **Schaltpläne**: werden für die H-Brücke sowie das NUCLEO-Entwicklungsboard durch den Kunden bereitgestellt.
- **Blockdiagramme**: für den vom Kunden bereitgestellten Testaufbau wird ein Blockdiagramm mit einigen Anschlussbeschreibungen bereitgestellt.
- **Softwaredokumentation**: Für die Kundenseitig bereitgestellten Bibliotheken werden Softwaredokumentationen in HTML und als PDF bereitgestellt.

1.4.2 Hardware

- **Entwicklungsplattform**: Die Entwicklungsplattform ist ein STM32F746/STM32H755 NUCLEO mit aufgesetzten STM-Stepper Shield.
- **Baugruppe**: Die Baugruppe wird vom Kunden für Detailtests sowie Spezifikationstests aber auch für fortgeschrittene Smoke-Tests bereitgestellt und kann in kurzen Zeiteinheiten für Tests genutzt werden.
- **Smoke-Test-Komponenten**: Für erste Tests ohne Baugruppenaufbau wird ein alternativer Schrittmotor bereitgestellt, welcher 400 Schritte pro Umdrehung (0,9° im Vollschrittbetrieb) auflöst. Dieser Motor darf mit einem maximalen Phasenstrom von 0,6A betrieben werden!

1.5 Übersicht der Meilensteine

Vorbereitungsphase	
Freigabe Pflichtenheft	
Implementierung und Test	
Simulation (Digital Twin) für Erstimplementierung entwickeln	
Implementierung der Spindel	
Implementierung des Steppers (synchron oder asynchron)	
Inbetriebnahme auf Zielhardware und Smoke-Testing	
Unit-Testing	
Spezifikationstest	
Einführung	
Schulung und Übergabe an den Kunden	
Voraussichtlicher Verkaufsstart	

2 Beschreibung der Anforderungen

2.1 Anforderungen an die Bewegungssteuerung

- **Achsenbewegung:** Der Linearantrieb ermöglicht eine präzise Positionierung entlang der X-Achse mit einer Wiederholgenauigkeit von 0,01 mm.

- **Geschwindigkeit:** Maximale Bewegungsgeschwindigkeit des Linearantriebs beträgt 1400 mm/min im 16er Mikroschrittbetrieb.

- **Anzeige des Betriebszustandes mit LEDs:** Der aktuelle betriebszustand soll mithilfe der blauen/gelben, roten und grünen LED angezeigt werden. Die Grüne LED zeigt die Betriebsbereitschaft des Steppers an, die blaue/gelbe LED blinkt mit 1Hz, sobald die Ausgangstreiber aktiv sind und eine Bewegung somit möglich ist. Während einer Bewegung bleibt das Blinkmuster dasselbe. Die rote LED ist im Zustand scsFAULT aktiv. Das heißt, wenn der Stepper einen Fehler meldet.

2.2 Anforderungen an die Spindelsteuerung

- **Drehzahlregelung:** Der DC-Motor der Spindel kann über die Konsole in einem Drehzahlbereich von 1600 - 9.000 U/min bei 24V gestartet und gestoppt werden. Da die Spindelsteuerung Open-Loop realisiert ist, weicht die Drehzahl bei geringerer Spannung ab!

- **Befehl zur Steuerung:**

    - **"spindle start <RPM>"** – Startet die Spindel mit angegebener Drehzahl. Während des Betriebs kann mit demselben Befehl die Drehzahl und Drehrichtung geändert werden.

    - **"spindle stop"** – Stoppt die Spindel.

2.3 Anforderungen an die Kommunikation und Steuerung

- **RS232-Schnittstelle:** Das System empfängt Steuerbefehle über die RS232-Verbindung. Die Konsole muss in der Lage sein, die Befehle im Kapitel Musskriterien sowie die Befehle im Kapitel Wunschkriterien zu verarbeiten.

- **Protokoll:** Das System akzeptiert einfache Textbefehle, die über RS232 übermittelt werden. Das System muss in der Lage sein, Befehle zu empfangen und die entsprechende Aktion auszuführen. Jeder Befehl muss korrekt terminieren und kann optionale Fehlertexte ausgeben. Details zu den Befehlen und dem Protokoll lassen sich aus der Schnittstellenspezifikation entnehmen.

2.4 Sicherheitsanforderungen

- **Not-Aus:** Ein Not-Aus-Schalter muss nicht eingebaut werden, da das System später nicht für Bediener und Monteure zugänglich sein wird und durch eine Risikoabschätzung auf Kundenseite eine minimale Verletzungsgefahr abgeschätzt wird. Die Haftbarkeit des Dienstleisters wird hiermit explizit abgetreten!

- **Überlastschutz:** Überlast- und Überhitzungsschutz für den Schrittmotor sind einzubauen, für den DC-Motor wird keine Überlast- und Überhitzungsschutzeinrichtung notwendig, da lediglich kurze Graviervorgänge stattfinden!

- **Sicherheitsvorkehrungen:** Schutzabdeckungen für rotierende Teile werden Kundenseitig angebaut. Laserlichtschranken werden ebenfalls Kundenseitig angebracht.

2.5 Betriebsbedingungen

- **Betriebstemperatur:** Das System muss bei Temperaturen von 10 °C bis 35 °C zuverlässig arbeiten.
- **Luftfeuchtigkeit:** Der Betrieb muss bei einer relativen Luftfeuchtigkeit von 30 % bis 70 % möglich sein.
- **Lagerung:** Das System muss auch bei extremen Lagerbedingungen -20 °C bis +60 °C stabil bleiben.

2.6 Testanforderungen

- **Funktionalität:** Alle Bewegungsachsen und die Spindel müssen nach der Montage und Inbetriebnahme in einem Testbetrieb fehlerfrei funktionieren. Der Kunde stellt für diesen Test eine Testapplikation zur Verfügung.
- **Präzision:** Die Positioniergenauigkeit des Linearantriebs muss ±0,01 mm betragen.
- **Stabilität:** Die Temperatur und das Drehmoment des DC-Motors müssen unter realen Arbeitsbedingungen innerhalb spezifizierter Grenzen bleiben.

2.7 Anforderung – Spindelbetrieb – Start

Nr. / ID	F10_01	**Nichttechnischer Titel**	Spindelstart		
Punkte		**Verweise**	Interfacespezifikation	**Kriterium**	MUSS

2.7.1 Beschreibung

Die Spindel muss durch die serielle Schnittstelle gestartet werden können. Dies geschieht mit dem Befehl „spindle start" in der Konsole. Der Befehl darf auch erfolgreich abgesetzt werden, wenn die Spindel bereits gestartet ist.

2.7.2 Testhinweise

Diese Anforderung wird durch die Testapplikation des Kunden getestet.

2.8 Anforderung – Spindelbetrieb – Stopp

Nr. / ID	F10_02	**Nichttechnischer Titel**	Spindelstopp		
Punkte		**Verweise**	Interfacespezifikation	**Kriterium**	MUSS

2.8.1 Beschreibung

Die Spindel muss durch die serielle Schnittstelle gestoppt werden können. Dies geschieht mit dem Befehl „spindle stop" in der Konsole. Der Befehl darf auch erfolgreich abgesetzt werden, wenn die Spindel bereits gestoppt ist.

2.8.2 Testhinweise

Diese Anforderung wird durch die Testapplikation des Kunden getestet.

2.9 Anforderung – Spindelbetrieb – Status

Nr. / ID	F10_03	**Nichttechnischer Titel**	Spindelstatus		
Punkte		**Verweise**	Interfacespezifikation	**Kriterium**	MUSS

2.9.1 Beschreibung

Der Status der Spindel muss über den Befehl „spindle status" in der Konsole mithilfe der seriellen Schnittstelle abgefragt werden können. Der Befehl muss immer erfolgreich abgesetzt werden können und darf nie mit einem Fehlercode terminieren. Als Rückgabe muss die Drehzahl sowie der Start/Stopp Zustand ausgegeben werden.

2.9.2 Testhinweise

Diese Anforderung wird durch die Testapplikation des Kunden getestet.

2.10 Anforderung – Schrittmotor – Reset

Nr. / ID	F20_01	**Nichttechnischer Titel**	Reset-Reinitialisierung		
Punkte		**Verweise**	Interfacespezifikation	**Kriterium**	MUSS

2.10.1 Beschreibung

Mithilfe des Reset Befehls kann der Kunde den Zustand des Schrittmotors und somit des Linearantriebs wieder in den Ursprungszustand nach dem Einschalten des Systems zurücksetzen. Der Befehl „stepper reset" führt zum abschalten der Ausgangstreiber, zur Reinitialisierung des Stepper Chips und zu der Notwendigkeit, eine neue Referenzfahrt auszuführen! Der Befehl kann immer abgesetzt werden und muss auch dazu führen, dass eine aktuelle Bewegung abgebrochen wird, sofern diese asynchron ausgeführt wird.

2.10.2 Testhinweise

Diese Anforderung wird durch die Testapplikation des Kunden getestet.

2.11 Anforderung – Schrittmotor – Referenzfahrt

Nr. / ID	F20_02	**Nichttechnischer Titel**	Referenzfahrt/Nullpunktabgleich		
Punkte		**Verweise**	Interfacespezifikation	**Kriterium**	MUSS

2.11.1 Beschreibung

Der Kunde muss in der Lage sein, eine Referenzfahrt anzustoßen. Bei diesem Vorgang muss der Linearmotor den Nullpunkt (Endschalter) anfahren und anschließend die korrekte Position in der internen Positionssteuerung zurücksetzen. Die neue Position wird als Bezugspunkt für die absolute Position des Bewegungsbefehls herangezogen. Die Referenzfahrt wird mit dem Befehl „stepper reference" über die serielle Schnittstelle gestartet. Die Referenzfahrt darf nicht beide Endlagen benutzen um daraus die Schrittanzahl zu bestimmen. Vielmehr darf nur der Referenzsensor benutzt werden und die Lage muss über die Firmware berechnet werden. Wird die Referenzfahrt angestoßen, wenn der Sensor bereits betätigt ist, dann muss zuerst eine kurze Strecke vom Sensor weggefahren werden, bevor der Sensor wieder angefahren wird.

2.11.2 Testhinweise

Diese Anforderung wird durch die Testapplikation des Kunden getestet.

2.12 Anforderung – Schrittmotor – Bewegung

Nr. / ID	F20_03	**Nichttechnischer Titel**	Position anfahren		
Punkte		**Verweise**	Interfacespezifikation	**Kriterium**	MUSS

2.12.1 Beschreibung

Der Kunde muss in der Lage sein, den Linearantrieb auf eine angeforderte Position fahren zu können, sofern eine Referenzfahrt durchgeführt wurde und kein Fehler des Stepper Treibers vorliegt. Die Position muss sich in den Grenzen des Linearsystems liegen. Angeforderte Werte außerhalb des anfahrbaren Bereichs müssen abgelehnt werden. Die Bewegung wird über die serielle Schnittstelle mit dem Befehl „stepper move <absPos>" gestartet.

Befindet sich das System in einem fehlerfreien Zustand, so muss eine grüne Status LED dies signalisieren. Ist das System in einem aktiven Zustand (Bewegungsfähig durch aktive Treiber oder in Bewegung), so muss die blaue/gelbe LED mit 1Hz blinken. Ist das System in einem Fehlerzustand, so muss die rote LED leuchten.

2.12.2 Testhinweise

Diese Anforderung wird durch die Testapplikation des Kunden getestet.

2.13 Anforderung – Schrittmotor – Positionsabfrage

Nr. / ID	F20_04	**Nichttechnischer Titel**	Positionsabfrage		
Punkte		**Verweise**	Interfacespezifikation	**Kriterium**	WUNSCH

2.13.1 Beschreibung

Mit diesem Befehl kann die aktuelle absolute Position des Schrittmotors abgefragt werden. Der Kunde kann somit gegenprüfen, ob seine angeforderte Position erreicht wurde oder kann diesen Befehl für weitere Zustandsinformationen in seiner Applikation nutzen. Der Befehl „stepper position" wird über die serielle Konsole abgesetzt.

2.13.2 Testhinweise

Diese Anforderung wird durch die Testapplikation des Kunden getestet.

2.14 Anforderung – Schrittmotor – Statusabfrage

Nr. / ID	F20_05	**Nichttechnischer Titel**	Statusabfrage		
Punkte		**Verweise**	Interfacespezifikation	**Kriterium**	WUNSCH

2.14.1 Beschreibung

Der Kunde möchte im Falle eines Fehlers den aktuellen Zustand des Zustandsautomaten sowie die Fehlerbits des Steppers einsehen können. Dazu ist der Befehl „stepper status" in der seriellen Konsole erforderlich, der neben diesen beiden Informationen zusätzlich auch angibt, ob eine asynchrone Operation im Hintergrund aktiv ist.

2.14.2 Testhinweise

Diese Anforderung wird durch die Testapplikation des Kunden getestet.

2.15 Anforderung – Schrittmotor – Konfigurationsparameter Lesen

Nr. / ID	F20_06	**Nichttechnischer Titel**	Lesen von Parametern		
Punkte		**Verweise**	Interfacespezifikation	**Kriterium**	WUNSCH

2.15.1 Beschreibung

Siehe API-Interfacespezifikation für genaue Details, Der Kunde kann mit diesem Befehl verschiedene Einstellungen und Randbedingungen im Betrieb des Systems einsehen. Der Lesebefehl erfolgt über die serielle Konsole mithilfe des Befehls „stepper config <Paramname>.

2.15.2 Testhinweise

Diese Anforderung wird durch die Testapplikation des Kunden getestet.

2.16 Anforderung – Schrittmotor – Konfigurationsparameter Schreiben

Nr. / ID	F20_07	**Nichttechnischer Titel**	Schreiben von Parametern		
Punkte		**Verweise**	Interfacespezifikation	**Kriterium**	WUNSCH

2.16.1 Beschreibung

Siehe API-Interfacespezifikation für genaue Details, Der Kunde kann mit diesem Befehl verschiedene Einstellungen und Randbedingungen im Betrieb des Systems umschalten. Der Schreibbefehl erfolgt über die serielle Konsole mithilfe des Befehls „stepper config <Paramname> -v <Value>.

2.16.2 Testhinweise

Diese Anforderung wird durch die Testapplikation des Kunden getestet.

2.17 Anforderung – DigitalTwin – Windows Simulator

Nr. / ID	F20_08	**Nichttechnischer Titel**	Digital Twin FreeRTOS Simulator		
Punkte		**Verweise**	Interfacespezifikation	**Kriterium**	WUNSCH

2.17.1 Beschreibung

Optionale Anforderung für den Qualitätsnachweis gegenüber dem Kunden. Hierzu wird ein Windowssimulator für FreeRTOS genutzt, der das Verhalten der Implementierung nachbildet und möglichst den vollständigen Source-Code der Lösung einbindet, um dies umzusetzen. Der Kunde kann darüber recht einfach API-Tests nachprüfen.

2.17.2 Testhinweise

Diese Anforderung wird durch die Testapplikation des Kunden getestet.

2.18 Anforderung – Master/Slave Timer für PWM – 2 Timer für den Stepper

Nr. / ID	F20_09	**Nichttechnischer Titel**		HW-Implementierungsvariante		
Punkte			**Verweise**	Interfacespezifikation	**Kriterium**	WUNSCH

2.18.1 Beschreibung

Siehe Vorlesungsfolien zu den möglichen Lösungsansätzen der PWM-Generierung. Ist für die asynchrone Implementierung mit freier Geschwindigkeitswahl erforderlich und vermutlich auch die sinnvollste Implementierungsvariante.

2.18.2 Testhinweise

Diese Anforderung wird durch eine Kundendemo und Sourcecode-Vorstellung geprüft.

2.19 Anforderung – Unit Tests für die Stepper-Implementierung

Nr. / ID	F20_10	**Nichttechnischer Titel**		SW-Qualitätsansprüche		
Punkte			**Verweise**	Interfacespezifikation	**Kriterium**	WUNSCH

2.19.1 Beschreibung

Mithilfe der LibCMocka sollen Unit Tests für die Stepper-Controller-Implementierung umgesetzt werden. Der Kunde akzeptiert bereits 4 einfache Schnittstellen Tests, die den Nachweis erbringen, dass mindestens zwei Teilfunktionen korrekt implementiert sind. Die Wahl darf durch den Entwickler erfolgen, Der Unit Test muss ein Testprotokoll erzeugen, welches mindestens beim Aufruf in der Konsole eingesehen werden kann.

2.19.2 Testhinweise

Diese Anforderung wird durch eine Kundendemo vorgeführt

2.20 Anforderung – Lib6474 in asynchronem Betrieb implementieren

Nr. / ID	F20_11	**Nichttechnischer Titel**		SW-Implementierungsvariante		
Punkte			**Verweise**	Interfacespezifikation	**Kriterium**	WUNSCH

2.20.1 Beschreibung

Siehe Code-Dokumentation der LibL6474. Der Betriebsmodus wird über den Konfigurationsheader aktiviert.

2.20.2 Testhinweise

Diese Anforderung wird durch eine Kundendemo und Sourcecode-Vorstellung geprüft.

Bedienungskonzept

2.21 Allgemeines Zustandsdiagramm

Siehe API-Interfacespezifikation

2.22 Schrittmotorzustandsdiagramme in Abhängigkeit der implementierten Anforderungen

Siehe API-Interfacespezifikation

3 Freigabe / Genehmigung

Datum:	
Unterschrift Auftraggeber:	
Unterschrift Projektleiter:	