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

KI-generierte Inhalte können fehlerhaft sein.

**Konzeption und Implementierung eines eigenständigen HMI-Systems auf Basis eines Encoder-Displays und einer selbst entwickelten Elektronikplatine mit STM32-Mikrocontroller zur Überwachung und Steuerung eines Pass-Laser-Engravers**

**Bachelor-Thesis**

zur Erlangung des Grades

**Bachelor of Engineering (B. Eng)**

HOCHSCHULE DARMSTADT

University of Applied Sciences

Fachbereich Elektrotechnik und Informationstechnik

von

**Orian Jordy Jomo Kamgo**

**1116805**

Studiengang: Elektrotechnik und Informationstechnik

Abgabedatum: 09.12.2025

Referent: **Prof. Dr. Carsten Zahout-Heil**

Korreferent: **Prof. Dr. Christian Jakob**

Industriebetreuer: **Sebastian Hinrichs**

Erklärung

Hiermit erkläre ich, dass ich die vorliegende Arbeit selbständig erstellt und keine anderen als die angegebenen Hilfsmittel benutzt habe. Soweit ich auf fremde Materialien, Texte oder Gedankengänge zurückgegriffen habe, enthalten meine Ausführungen vollständige und eindeutige Verweise auf die Urheber und Quellen. Alle weiteren Inhalte der vorgelegten Arbeit stammen von mir im urheberrechtlichen Sinn, soweit keine Verweise und Zitate erfolgen. Mir ist bekannt, dass ein Täuschungsversuch vorliegt, wenn die vorstehende Erklärung sich als unrichtig erweist.

Ort, Datum Unterschrift

Inhaltsverzeichnis

[Abstract 1](#_Toc219806046)

[1. Einleitung 1](#_Toc219806047)

[1.1 Motivation 1](#_Toc219806048)

[1.2 Zielsetzung 2](#_Toc219806049)

[1.4 Anforderungen und Rahmbedingungen 3](#_Toc219806050)

[1.3 Aufbau der Arbeit 5](#_Toc219806051)

[2. Stand der Technik 6](#_Toc219806052)

[2.1 Pass-Laser-Engraver 6](#_Toc219806053)

[2.2 Stand des HMI-Systems 7](#_Toc219806054)

[2.3 Übersicht bestehender HMI-Lösungen 8](#_Toc219806055)

[2.3.1 Displaybasierte Embedded-HMI-Systeme (Mikrocontroller + TFT) 8](#_Toc219806056)

[2.3.2 Industrielle Panel-HMIs (Touchpanels) 10](#_Toc219806057)

[2.3.4 Web-basierte und Remote-HMI-Systeme 12](#_Toc219806058)

[2.4 Zusammenfassung und Auswahl der angewandte 14](#_Toc219806059)

[Technologie 14](#_Toc219806060)

[2.4.1 Methodik 14](#_Toc219806061)

[Ergebnisse der Nutzwertanalyse 16](#_Toc219806062)

[Diskussion 18](#_Toc219806063)

[3. Theoretische Grundlagen 20](#_Toc219806064)

[3.1 TFT-Display 20](#_Toc219806065)

[3.2 Mikrocontroller 22](#_Toc219806066)

[3.2.1 STM32-Mikrocontroller Familie 23](#_Toc219806067)

[3.3 HMI-System 24](#_Toc219806068)

[3.4 CAN-Bus 25](#_Toc219806069)

[3.4.1 Allgemeine Informationen 25](#_Toc219806070)

[3.4.2 Verschiedene Varianten der CAN-Busses 26](#_Toc219806071)

[3.4.3 Aufbau der CAN-Nachrichten 27](#_Toc219806072)

[3.5 SPI 29](#_Toc219806073)

[3.5.1 Allgemein 29](#_Toc219806074)

[3.5.2 Prinzip 29](#_Toc219806075)

[3.5.3 Datenübertragung 31](#_Toc219806076)

[3.6 Interne FLASH \_MEMORY eines STM32-Mikrocontroller 32](#_Toc219806077)

[3.6.1 Allgemein Grundlage nicht-flüchtiger Speicher 32](#_Toc219806078)

[3.6.2 Architektur und Funktionsweise von Flash-Speicher 32](#_Toc219806079)

[3.6.3 Programmiertechnik und Zugriff 33](#_Toc219806080)

[4. Konzept 34](#_Toc219806081)

[4.1 Entwurf der Platine 34](#_Toc219806082)

[4.1.1 Anforderungen 34](#_Toc219806083)

[4.1.2 Hardwaredesign 34](#_Toc219806084)

[4.1.3 Leiterplattenentwurf 36](#_Toc219806085)

[4.2 Softwareentwicklung 43](#_Toc219806086)

[4.2.1 Anforderungen 43](#_Toc219806087)

[4.2.2 Softwaredesign 43](#_Toc219806088)

[45](#_Toc219806089)

[4.2.3 Softwareimplementierung 46](#_Toc219806090)

[5 Validierung 48](#_Toc219806091)

[5.1 Durchführung der Testläufe & Ergebnis 48](#_Toc219806092)

[5.2 Diskussion 52](#_Toc219806093)

[6 Zusammenfassung & Ausblick 56](#_Toc219806094)

[Anhang 56](#_Toc219806095)

[Abbildungsverzeichnis 56](#_Toc219806096)

[Tabellenverzeichnis 56](#_Toc219806097)

[Abkürzungsverzeichnis 56](#_Toc219806098)

[Literaturverzeichnis 57](#_Toc219806099)

# Abstract

# 1. Einleitung

## 1.1 Motivation

Die Qualität der Mensch-Maschine-Schnittstelle (HMI[[1]](#footnote-1)) ist relevant für die Gesamtbewertung einer Maschine[1]. HMI-Systeme ermöglichen die Interaktion zwischen Menschen und Maschinen über Benutzeroberflächen. Sie sind in der Automobilindustrie ein integraler Bestandteil, da nahezu jedes Fahrzeug mit einem Display ausgestattet ist[2]. Fehlerhafte HMI-Systeme können schwerwiegende Konsequenzen haben. Sie können zu Fehlbedienungen und Produktionsausfällen führen und die Sicherheit beeinträchtigen.

Das bestehende System (siehe Kapitel 2.1) verwendet ein konventionelles Anzeige- und Steuerkonzept auf Basis von RGB[[2]](#footnote-2)-LEDs. Diese klassische Form des HMI ist funktional eingeschränkt. Im Rahmen des Projekts wird die Entwicklung eines modernen HMI-Systems auf Basis eines TFT-Displays angestrebt, das bestehende System ablösen soll. Ziel ist die Optimierung der Informationsdarstellung und der Benutzerinteraktion. Das System soll zusätzliche Funktionen bieten, beispielsweise das Auswerfen eines Passes durch Betätigung des Push-Buttons auf dem Display und Fehlermeldungen ausgeben können.

Eine präzise Abstimmung zwischen Hard- und Software ist erforderlich, um eine zuverlässige Funktionalität und störungsfreie Kommunikation zu gewährleisten. Ziel des Projekts ist die Entwicklung eines effizienten, sicheren und benutzerfreundlichen HMI-Systems, das den aktuellen Anforderungen entspricht.

Die Integration des HMI-Systems in der Maschine ist in Abbildung 1 dargestellt. Es handelt sich um einen Pass-Laser-Engraver, der in Kapitel 2.1 beschrieben wird.

Ein Bild, das Text, Haushaltsgerät, Im Haus, Design enthält.

KI-generierte Inhalte können fehlerhaft sein.

Abbildung 1: Laser-Pass-Engaver der Firma „Surys GmbH“ und Encoder-Display der Firma

„ddm hopt+schuler GmBH&Co. KG“

## 1.2 Zielsetzung

Ziel dieser Arbeit ist die Entwicklung und Implementierung eines Human-Machine-Interface (HMI)-Systems, das eine zuverlässige Visualisierung der Systemzustände und Systeminformationen auf einem TFT-Display ermöglicht. Über den im Display integrierten Encoder und Push-Button sollen Steuerbefehle an die Maschine übermittelt werden. Das System zielt auf eine einwandfreie und stabile Kommunikation zwischen der Maschine und dem Display ab.

Im Rahmen der Entwicklung wird eine elektronische Leiterplatte entworfen und realisiert, welche sämtliche notwendigen Komponenten integriert. Die Hardware ist konzipiert, um eine sichere, unterbrechungsfreie und störungsfreie Datenübertragung zwischen Display und Maschine zu gewährleisten. Parallel dazu erfolgt die Entwicklung einer optimierten Softwarearchitektur, welche eine reibungslose Interaktion zwischen Hard- und Softwarekomponenten gewährleistet, und eine effiziente Systemsteuerung ermöglicht.

Zur Erreichung dieser Zielsetzung werden verschiedene Mikrocontroller und serielle Schnittstellen analysiert und hinsichtlich ihrer Eignung bewertet. Der Fokus liegt auf den Mikrocontrollern der STM32-Serie sowie den Kommunikationsschnittstellen SPI, I2C, CAN-BUS und RS485. Auf Grundlage der durchgeführten Analyse wird ein technisches Gesamtkonzept für ein leistungsfähiges HMI-System entwickelt. Im Anschluss an die Installation erfolgt die Implementierung der Software sowie die Inbetriebnahme des Systems. Abschließend erfolgt eine Funktionsprüfung und Integrationsbewertung, um die Einhaltung der definierten Anforderungen und die Systemstabilität zu verifizieren.

## 1.4 Anforderungen und Rahmbedingungen

Bei der Entwicklung des HMI-Systems sind viele technische und funktionale Anforderungen zu berücksichtigen, z. B. die zuverlässige Aktualisierung der Maschinenzustände auf dem Display. Die Zustandsabfrage kann dabei über Polling-Mechanismen oder ereignisgesteuerte Trigger erfolgen.

Die Signalintegrität muss bereits während der Entwurfsphase sichergestellt werden, z. B. durch ein zielgerechtes Leiterplattendesign und eine robuste Firmwarearchitektur, um Störanfälligkeiten im Übertragungsweg zu minimieren.

Die Hardware muss gegen externe Störeinflüsse abgesichert sein, z. B. durch ESD-Schutz und EMI-Unterdrückung. Nur so kann die Funktionalität des HMI-Systems gewährleistet werden.

Eine nachträgliche Aktualisierung der Firmware des Mikrocontrollers ist bei Bedarf vorzusehen, z. B. durch geeignete Programmier- und Debug-Schnittstellen. In der vorliegenden Arbeit werden SWD und USB untersucht.

Für den Datenaustausch zwischen HMI-System und Maschine ist eine serielle Kommunikationsschnittstelle erforderlich. Die Auswahl des geeigneten Kommunikationsbusses wird maßgeblich durch die Schnittstellen der Maschinensteuerung bestimmt, welche die Protokolle SPI, I2C, CAN-Bus und RS-485 unterstützen. Die gewählte Schnittstelle muss eine ausreichende Datenrate, Robustheit sowie Störsicherheit gewährleisten, um eine zuverlässige Übertragung sicherzustellen.

Die mechanische Integration des Systems sieht eine Verschraubung mit dem Display vor, wofür im Leiterplattenlayout Pad-Holes[[3]](#footnote-3) mit einem Innendurchmesser von 3 mm und einem Außendurchmesser von 5 mm berücksichtigt werden müssen. Die Spannungsversorgung des HMI-Systems erfolgt über die Maschine mit einer Versorgungsspannung von 5 V/DC. Für den eingesetzten Mikrocontroller muss eine entsprechende Aufbereitung und Stabilisierung vorgenommen werden. Das HMI-System besteht aus einem TFT-Display und einem Drehencoder mit zusätzlicher Push-Button-Funktion für die Benutzereingabe. Zur Realisierung der erforderlichen Datenverarbeitung und Interaktion wird ein STM32-Mikrocontroller eingesetzt, da diese Mikrocontrollerfamilie im Entwicklungsumfeld bereits etabliert ist und eine hohe Leistungsfähigkeit sowie langfristige Verfügbarkeit bietet. Das Display dient der Visualisierung von Statusinformationen der Maschine sowie Änderungen relevanter Maschinenparameter. Die Firmwareentwicklung erfolgt in der herstellereigenen, kostenfreien Entwicklungsumgebung STM32CubeIDE, für die Erstellung der Leiterplatte wird Altium Designer eingesetzt. Der Entwicklungsumfang umfasst gemäß den vorliegenden Informationen die Konzeption, die Auswahl der Bauteile, die Schaltplanerstellung, die Ausarbeitung des PCB-Layouts, die Erstellung der Fertigungsunterlagen sowie die Übergabe an den Leiterplattenhersteller und Programmierung. Darüber hinaus umfasst der Projektumfang auch die Bauteilbeschaffung sowie die Korrespondenz mit Lieferanten.

Das Projektziel ist die Realisierung eines funktionsfähigen Prototyps mit vollständiger technischer Dokumentation für die Serienentwicklung. Eine Testumgebung gewährleistet die Verifikation des HMI-Systems.

Eine Zusammenfassung der Anforderungen wird in der nachfolgenden Tabelle aufgelistet.

**Funktionale Anforderungen (F):**

**F-A1:** Das HMI-System muss Maschinenzustände erfassen und in Echtzeit auf dem Display darstellen.

**F-A2:** Zustandsübermittlung über Kommunikation oder Polling

Das System empfängt und verarbeitet Maschinenzustände zyklisch oder ereignisgesteuert.

**F-A3:** Datenaustausch zwischen HMI-System und Maschinesteuerung

Das HMI-System muss einen bidirektionalen Datenaustausch mit der Maschinesteuerung ermöglichen.

**F-A4:** Firmware-Update-Funktion

Das HMI-System muss Schnittstellen für Firmware-Aktualisierungen unterstützen.

**F-A5:** Benutzerinteraktion über Encoder und Push-Button

Das HMI-System erfasst Benutzereingaben über einen Drehencoder und verwendet sie zur Navigation und Funktionsauswahl.

**F-A6:** Menübasierte Benutzerführung: Das System muss eine logisch strukturierte, hierarchische Menüführung bereitstellen, die eine intuitive Bedienung ermöglicht.

**Nicht-funktionale Anforderungen (NF):**

**NF-A1:** Das HMI-System muss mit geringer Latenz auf Benutzerinteraktionen und Zustandsänderungen reagieren und bestimmtes Echtzeitverhalten aufweisen.

**NF-A2:** Die Signalintegrität aller Signale muss durch Hardwaredesign und Softwaredesign sichergestellt werden.

**NF-A3:** Schutz gegen ESD- und EMI-Störeinflüsse

Die HMI-Platine muss gegen elektrostatische Entladungen und elektromagnetische Störungen abgesichert sein.

**NF-A4:** Mechanische Integrationsfähigkeit

Die Platine muss über mechanische Befestigung sicher in das Gesamtsystem integriert werden können.

**NF-A5:** Die Software- und Hardwarearchitektur muss so gestaltet sein, dass Erweiterungen, Wartung und Fehlerdiagnose möglich sind.

## 1.3 Aufbau der Arbeit

# 2. Stand der Technik

In diesem Kapitel werden der aktuelle Stand der Technik sowie wissenschaftliche und praxisrelevante Arbeiten im Umfeld von Pass-Laser-Engravern und HMI vorgestellt. Das Ziel besteht darin, wesentliche technologische Entwicklungen und zentrale Trends zu identifizieren, die für die Gestaltung des HMI-Systems im Maschinenkontext von besonderer Relevanz sind. Die Arbeit verfolgt einen methodischen und technologischen Lösungsansatz, der durch eine systematische Analyse bestehender Konzepte und technischer Ansätze fundiert und eingeordnet werden soll.

## 2.1 Pass-Laser-Engraver

Der im Unternehmen IN Groupe / SURYS eingesetzte Pass-Laser-Engraver stellt ein spezialisiertes System zur sicheren Personalisierung von Identitäts- und Reisedokumenten, insbesondere von Pässen, dar. Das System dient der direkten Gravur von personenbezogenen Daten, maschinenlesbaren Informationen sowie sicherheitsrelevanten Grafiken auf die Polycarbonat-Datenseite des Dokuments. Die Lasergravur generiert dauerhafte und fälschungssichere Merkmale, die eine hohe Beständigkeit gegenüber chemischen Einflüssen und Manipulationsversuchen aufweisen, sodass als Endprodukt ein vollständig personalisierter, einsatzbereiter elektronischer Reisepass mit integrierten Sicherheitsmerkmalen entsteht. In der Konsequenz wird ein durchgängiger, hochgradig automatisierter Personalisierungsprozess realisiert. Diese präzise Lasergravurtechnologie leistet einen wesentlichen Beitrag zur Erfüllung der hohen Sicherheits-, Qualitäts- und Effizienzanforderungen, die an moderne Pass- und Identitätsdokumente gestellt werden[3,4]. Eine Abbildung des Geräts ist in Abbildung … dargestellt.

## 2.2 Stand des HMI-Systems

Das vorliegende HMI-System des Pass-Laser-Engravers, welches auf zwei RGB-LEDs basiert, repräsentiert eine elementare Form der Mensch-Maschine-Schnittstelle und findet in kompakten sowie kostenbewussten Maschinen häufig Anwendung. Solches System weist sowohl positive als auch negative Eigenschaften auf, die im Folgenden gegenübergestellt werden.

**Vorteile:**

* **Hohe Robustheit:** LEDs sind unempfindlich gegenüber Vibrationen, Staub und Temperaturschwankungen, was sie besonders für industrielle Umgebungen geeignet macht.
* **Einfache Ansteuerung:** Die Ansteuerung erfolgt über digitale Ausgänge, wodurch kaum Software- oder Hardwareaufwand entsteht.
* **Geringer Platzbedarf:** Das System benötigt nur sehr wenig Einbauraum auf der Frontplatte und der Leiterplatte.
* **Hohe Zuverlässigkeit:** Durch die einfache Struktur ist die Fehleranfälligkeit gering.

**Nachteile:**

* **Sehr eingeschränkte Informationsdarstellung:** Es können ausschließlich Zustände über Farben signalisiert werden; detaillierte Prozessinformationen sind nicht darstellbar.
* **Keine Interaktionsmöglichkeit:** Der Benutzer kann keine Parameter ändern oder gezielt Eingaben vornehmen.
* **Mehrdeutigkeit der Signale:** Farbcodes müssen erlernt werden und können bei schlechter Beleuchtung oder Farbsehschwäche missverstanden werden.
* **Keine Prozessvisualisierung:** Weder Messwerte noch Arbeitsfortschritte oder Fehlermeldungen können textlich oder grafisch angezeigt werden.
* **Begrenzte Erweiterbarkeit:** Zusätzliche Funktionen erfordern meist Hardwareänderungen.
* **Nicht mehr zeitgemäß:** Für moderne, komplexe Maschinen entspricht ein reines LED-HMI nicht mehr den Anforderungen an Bedienkomfort und Rückverfolgbarkeit.

Das LED-basierte HMI-System findet seinen Einsatz bei der Realisierung sehr einfacher Zustandsanzeigen und zeichnet sich durch einen geringen Kosten- und Hardwareaufwand aus. Für moderne Anwendungen, wie den Pass-Laser-Engraver, die hohen Anforderungen an Prozessüberwachung, Parametrierung und Benutzerinteraktion stellt, ist ein solches System jedoch funktional stark limitiert. Daher ist die Analyse verschiedener HMI-Lösungen von essentieller Bedeutung.

## 2.3 Übersicht bestehender HMI-Lösungen

Im Rahmen der vorliegenden Arbeit wurde ein HMI-System entwickelt. Um eine fundierte Einordnung zu gewährleisten, werden die wichtigsten HMI-Lösungsansätze vorgestellt. Diese unterscheiden sich in Bezug auf Komplexität, Leistungsfähigkeit, Kosten und Integrationsaufwand.

### 2.3.1 Displaybasierte Embedded-HMI-Systeme (Mikrocontroller + TFT)

Displaybasierte Embedded-HMI-Systeme sind eine Kategorie von Benutzerschnittstellen. Ihre Funktionsweise basiert auf der Kombination eines Mikrocontrollers mit einem grafischen TFT-Display. Im Unterkapitel 3.1 wird der Begriff TFT-Display ausführlich erklärt. Die genannten Komponenten gestatten die Visualisierung von System- und Prozesszuständen, die Darstellung von Messwerten sowie die Eingabe von Steuerbefehlen über Touchscreen, Encoder oder Tasten.[6]

Ein Bild, das Elektronik, Schaltung enthält.

KI-generierte Inhalte können fehlerhaft sein.

Displaybasierte Embedded HMI-System[7]

Die nachfolgende Tabelle zeigt die Vor und Nachteile des Systems[8] [9] [10]:

|  |  |
| --- | --- |
| **Vorteile** | **Nachteile** |
| Kombination aus Mikrocontroller und TFT-Display in einem kompakten Modul | Eingeschränkte grafische Möglichkeiten bei aufwändigen Animationen oder komplexen Visualisierungen |
| Direkte Maschinenanbindung über serielle Schnittstellen (z. B. SPI, I²C, UART, CAN, RS-485) | Erhöhter Entwicklungsaufwand, da sowohl Hardware als auch Software individuell erstellt werden müssen |
| Echtzeitfähigkeit durch direkten Hardwarezugriff ohne Betriebssystem | Geringere Erweiterbarkeit, da spätere Funktionsänderungen oft Hardware-Anpassungen erfordern |
| Geringer Energieverbrauch im Vergleich zu PC-basierten Lösungen |  |
| Hohe Anpassbarkeit, da sowohl Hardware als auch Firmware projektspezifisch ausgelegt werden können |  |

Ein farbiges TFT-Display ermöglicht eine gut lesbare, grafische Darstellung von Zuständen, Symbolen und Prozesswerten. In Kombination mit Touch, Tasten oder Encoder entsteht eine direkte, intuitive Interaktion ohne Umweg über einen PC. Die grafische Benutzeroberfläche kann an die spezifischen Erfordernisse der Maschine angepasst werden, beispielsweise in Bezug auf das Layout, die Sprache oder die Symbole. Dadurch werden die für den Benutzer relevanten Informationen und Bedienpfade angezeigt, was zu einer Reduzierung von Fehlbedienungen und einer Verkürzung der Einarbeitungszeiten führt. Darüber hinaus bieten TFT-Displays hohe Auflösung, schnelle Reaktionszeit, gute Helligkeit und lange Verfügbarkeit in industriellen Anwendungen. Dies führt zu einer zuverlässigen, modernen und ergonomischen Wahrnehmung der Displays.[11], [12]

### 2.3.2 Industrielle Panel-HMIs (Touchpanels)

Industrielle Panel-HMIs sind als kompakte, vorkonfektionierte Bediengeräte konzipiert. Sie integrieren ein robustes Touchdisplay, eine Recheneinheit sowie standardisierte industrielle Kommunikationsschnittstellen. Sie fungieren als grafische Benutzerschnittstelle, die dazu dient, technische Prozesse in Maschinen und Anlagen zu steuern, zu visualisieren und zu überwachen. Aufgrund ihrer industriellen Auslegung sind sie für den dauerhaften Einsatz in anspruchsvollen Produktionsumgebungen konzipiert, in denen Erschütterungen, Temperaturwechsel, Feuchtigkeit oder elektromagnetische Störeinflüsse auftreten können.[13]

Ein Bild, das Text, Elektronik, Maschine, Im Haus enthält.

KI-generierte Inhalte können fehlerhaft sein.

Industrielle Panel HMI[14]

Die nachfolgende Tabelle stellt die Vor und Nachteile dar:[15], [16]

|  |  |
| --- | --- |
| **Vorteile** | **Nachteile** |
| Hohe Zuverlässigkeit: Durch robuste Bauweise und industrielle Schutzklassen für harte Einsatzumgebungen. | Hohe Kosten: Industrielle Panels sind deutlich teurer als embedded HMI-Lösungen mit Mikrocontroller. |
| Schnelle Inbetriebnahme: Da Hard- und Softwareplattform bereits integriert und standardisiert sind. | Begrenzte Anpassbarkeit: Hardware- und Softwarearchitektur sind oft herstellergebunden. |
| Erweiterte Visualisierungsmöglichkeiten: Grafische Darstellungen, Trends, Parameteransichten, Rezepturen und Fehlermeldungen. | Erhöhter Platzbedarf: Für Frontmontage oder Schaltschrankintegration wird vergleichsweise viel Raum benötigt. |
| Standardisierte Integration: Kompatibilität mit gängigen Automatisierungsprotokollen (z. B. Modbus, OPC UA, Profinet). | Aufwändiger bei einfachen Systemen: Für kleinere Maschinen kann die Leistung eines Industrie-Panels unverhältnismäßig hoch sein. |
| Hoher Bedienkomfort: Touchoberflächen ermöglichen effiziente Navigation und reduzieren Schulungsaufwand. |  |

Visuelle Statusanzeigen, Menüstrukturen und Fehlermeldungen können demnach schneller interpretiert werden als traditionelle LED-Signale oder mechanische Tasten. Die Verwendung großer Displays hat sich in der Praxis als effektives Mittel zur Verbesserung der Situationswahrnehmung erwiesen. Empirische Studien belegen, dass sich dadurch die Fehlerquote bei der Bedienung reduziert und die Prozesssicherheit erhöht.[17]

### 2.3.3 Web-basierte und Remote-HMI-Systeme

Web-basierte und Remote-HMI-Systeme sind eine Art der Bedienlösungen, bei denen die Visualisierung über einen integrierten Webserver erfolgt. Die Nutzung kann mit Standard-Webbrowsern auf beliebigen Endgeräten erfolgen. Die technische Grundlage ist eine Client-Server-Architektur. Die Steuerung stellt die HMI-Seiten als Webanwendung zur Verfügung, bindet Prozessdaten ein und erlaubt den gleichzeitigen Zugriff mehrerer Nutzer. Dies ermöglicht eine Bedienung, Diagnose und Überwachung sowohl lokal als auch remote.[18], [19]

Ein Bild, das Screenshot, Tablet, Quadrat, Text enthält.

KI-generierte Inhalte können fehlerhaft sein.

Web-basierte HMI-System[20]

|  |  |
| --- | --- |
| **Vorteile** | **Nachteile** |
| Hohe Flexibilität und Skalierbarkeit, ideal für Industrie 4.0[[4]](#footnote-4). | Abhängigkeit von Netzwerkverfügbarkeit und Latenzen, was in industriellen Umgebungen kritisch sein kann. |
| Einfache Integration in IT-Netzwerke, inklusive Cloud-Anbindung. | Nicht ideal für sicherheitskritische Prozesse, die lokale, echtzeitfähige Bedienoberflächen benötigen. |
| Geringere Hardwarekosten, da kein lokales Display benötigt wird. | Variierende Bedienqualität, abhängig vom Endgerät des Nutzers. |
| Remote-Wartung und Diagnose reduzieren Servicezeiten und Stillstände. |  |

Web-basierte HMI-Systeme sind besonders flexibel und benutzerfreundlich, da sie über standardisierte Webtechnologien und Netzwerkprotokolle realisiert werden. Sie kommen ohne dedizierte Bedienhardware aus und können unmittelbar als HMI genutzt werden. Anwender profitieren vom orts- und geräteunabhängigen Zugriff, von modernen Benutzeroberflächen und der zentralisierten Wartung.

## 2.4 Zusammenfassung und Auswahl der angewandte

## Technologie

Im Rahmen dieser Arbeit wurde die Nutzwertanalyse als systematisches Entscheidungsinstrument eingesetzt. Dieses Verfahren ermöglicht die quantitative Bewertung mehrerer Kriterien, indem diesen Gewichtungsfaktoren zugeordnet und die jeweiligen Zielerreichungsgrade der betrachteten Alternativen ermittelt werden. Durch die Aggregation der gewichteten Einzelbewertungen wird für jede Alternative ein Gesamtnutzwert berechnet. Ziel der Methode ist es, den relativen Nutzen der untersuchten Lösungsansätze transparent, nachvollziehbar und vergleichbar darzustellen sowie eine fundierte Rangfolge der Alternativen abzuleiten.[22]

### 2.4.1 Methodik

Für die Entwicklung des HMI-Systems müssen alle relevanten technischen und wirtschaftlichen Einflussfaktoren erfasst und bewertet werden, um eine Lösung zu identifizieren, die den Anforderungen der Maschine und den wirtschaftlichen Rahmenbedingungen des Projekts gerecht wird. Für das Projekt wurden Kriterien wie Integrationsfähigkeit in die Maschine, Kosten, intuitive Bedienbarkeit sowie Reaktionszeit des Systems definiert und entsprechend gewichtet. Die Anforderungsmatrix bildet die Grundlage für die Nutzwertanalyse und die Auswahl des HMI-Systems.

In der Nutzwertanalyse beschreibt die Gewichtung die relative Bedeutung eines Kriteriums (in Prozent), während die Bewertung auf einer Skala von 1 (schlecht) bis 5 (sehr gut) angibt, in welchem Maß eine Alternative dieses Kriterium erfüllt.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Kriterium** | **Gewichtung (%)** | **Zielvorgabe** | **Bewertung (1-5)** | **Bemerkung** |
| **Integration** | 30 | vernünftige Einsetzung in der Maschine | 5 | Kompakte Bauform, direkte Integration in bestehende Maschinenarchitektur |
| **Kosten** | 30 |  | 4 | Entwicklungskosten sollen nicht mehr als 4500 € |
| **Intuitive Bedienung** | 20 | Einfache, verständliche Benutzerführung | 4 | Menüstruktur und grafische Darstellung erhöhen Benutzerfreundlichkeit |
| **Reaktionszeit** | 20 | Schnelle Systemreaktion | 5 | Direkte MCU-Ansteuerung ohne übergeordnetes Betriebssystem ermöglicht geringe Latenzen |

**Integration (30%):**

Die Integrationsfähigkeit eines HMI-Systems ist ein wesentliches Bewertungskriterium für die Zuverlässigkeit, Bedienbarkeit und funktionale Einbettung in die Maschinenumgebung. Eine fest integrierte Lösung reduziert mechanische Belastungen, minimiert Störquellen und gewährleistet Betriebssicherheit. Durch Beispielweise eine frontseitige Platzierung erhält das Bedienpersonal einen optimalen Zugriff auf Prozessinformationen und Steuerfunktionen, was die Bediengeschwindigkeit und Reaktionsfähigkeit verbessert. Die vorliegende Lösung ermöglicht eine unkomplizierte Einbindung in die bestehende Systemarchitektur.

**Kosten (30%):**

Die Kosten sind ein wesentliches Kriterium bei der Auswahl eines HMI-Systems. Insbesondere in der Prototypenentwicklung ist eine wirtschaftlich tragfähige Lösung entscheidend. Weitere Faktoren sind die Kosten für Bauteile, Fertigung, Softwarelizenzen sowie Wartung oder Ersatzkomponenten. Eine kosteneffiziente Ausgestaltung des HMI-Systems ist unerlässlich, um die Wirtschaftlichkeit des Gesamtsystems sicherzustellen und eine spätere Integration in Serienprodukte zu ermöglichen.

**Intuitive Bedienung (20%):**

Ein HMI-System sollte eine einfache, verständliche Benutzerführung gewährleisten, damit der Bediener Aktionen schnell und fehlerfrei ausführen kann, ohne umfangreiche Schulungen. Die Menüstruktur und grafische Darstellung spielen dabei eine entscheidende Rolle für die Benutzerfreundlichkeit.

**Reaktionszeit (20%):**

Das Kriterium der Reaktionszeit erfordert eine schnelle Systemantwort bei minimaler Latenz. Die direkte Ansteuerung der Hardware über den Mikrocontroller ohne Betriebssystem erfüllt diese Anforderung und gewährleistet die Echtzeitfähigkeit des HMI-Systems.

Die Anforderungsmatrix ist die Grundlage für die Bewertung und den Vergleich der HMI-Ansätze. Durch Gewichtung und Bewertung der Kriterien wird eine objektive Entscheidungsfindung ermöglicht. Das HMI-Konzept mit den höchsten Punktzahlen wird als geeignetste und technisch vielversprechendste Lösung identifiziert. Dieses Vorgehen berücksichtigt funktionale, wirtschaftliche und ergonomische Aspekte und gewährleistet so eine fundierte Auswahl des HMI-Systems.

### Ergebnisse der Nutzwertanalyse

Die Bewertung der HMI-Ansätze erfolgte anhand einer Bewertungsskala von 1 (ungenügend) bis 5 (sehr gut).

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Kriterium** | **Gewichtung** | **Displaybasierte Embedded-HMI-Systeme** | **Industrielle Panel-HMIs** | **Web-basierte und Remote-HMI-Systeme** |
| **Integration** | 30 % | 5 | 4 | 3 |
| **Kosten** | 30 % | 5 | 2 | 4 |
| **Intuitive Bediebung** | 20% | 4 | 5 | 4 |
| **Reaktionszeit** | 20 % | 5 | 4 | 3 |

**Ein Bild, das Text, Screenshot, Schrift, Diagramm enthält.

KI-generierte Inhalte können fehlerhaft sein.**

Grafische Darstellung der Bewertungsgrößen jeder mögliche Ansätze nach Kriterien

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Technologie/Kriterium** | **Integration** | **Kosten** | **Intuitive Bediebung** | **Rekationszeit** | **Gesamtnutzwert** |
| **Displaybasierte Embedded-HMI-Systeme** | 5 \* 0.3 = 1.5 | 5 \* 0.3 = 1.5 | 4 \* 0.2 = 0.8 | 5 \* 0.2 = 1 | 4.8 |
| **Industrielle Panel-HMIs** | 4 \* 0.3 = 1.2 | 2 \* 0.3 = 0.6 | 5 \* 0.2 = 1 | 4 \* 0.2 = 0.8 | 3.6 |
| **Web-basierte und Remote-HMI-Systeme** | 3 \* 0.3 = 0.9 | 4 \* 0.3 = 1.2 | 4 \* 0.2 = 0.8 | 3 \* 0.2 = 0.6 | 3.5 |

Berechnung des Gesamtnutzwerts jeder Ansatz

### Diskussion

Ein Bild, das Text, Screenshot, Diagramm, Design enthält.

KI-generierte Inhalte können fehlerhaft sein.

Grafische Darstellung des Gesamtnutzwerts jedes Ansatzes

Die Ergebnisse der Nutzwertanalyse aus der Tabelle x ermöglichen eine differenzierte Bewertung der HMI-Systemansätze. Die gewichteten Kriterien Integration, Kosten, intuitive Bedienung und Reaktionszeit zeigen die Stärken und Schwächen der HMI-Lösungen.

**Displaybasierte Embedded-HMI-Systeme** erreichen mit einem Gesamtnutzwert von **4,8** die höchste Bewertung. Ausschlaggebend hierfür sind insbesondere die sehr gute Integrationsfähigkeit in die bestehende Maschinenarchitektur, die schnelle Reaktionszeit sowie die vergleichsweisen geringen Kosten. Diese Lösung bietet ein ausgewogenes Verhältnis zwischen technischer Leistungsfähigkeit und wirtschaftlicher Effizienz und eignet sich daher besonders für den vorgesehenen Einsatz im Passport-Engraver

**Industrielle Panel-HMIs** erzielen einen Gesamtnutzwert von **3,6**. Zwar überzeugen sie durch eine hohe Benutzerfreundlichkeit und industrielle Robustheit, jedoch wirken sich die hohen Anschaffungs- und Integrationskosten sowie der größere Platzbedarf negativ auf die Gesamtbewertung aus.

**Web-basierte und Remote-HMI-Systeme** weisen mit **3,5** den niedrigsten Gesamtnutzwert auf. Trotz ihrer Flexibilität und der Möglichkeit zum ortsunabhängigen Zugriff sind sie aufgrund von Abhängigkeiten von Netzwerkverfügbarkeit, höheren Latenzen und eingeschränkter Echtzeitfähigkeit für den betrachteten Anwendungsfall weniger geeignet.

Zusammenfassend zeigt die Nutzwertanalyse, dass das displaybasierte Embedded-HMI-System sowohl aus technischer als auch aus wirtschaftlicher Sicht die geeignetste Lösung für das vorliegende Projekt darstellt und daher für die weitere Umsetzung ausgewählt wird.

# 3. Theoretische Grundlagen

## 3.1 TFT-Display

Ein **TFT-Display** (Thin-Film-Transistor-Display) ist eine spezielle Form eines LCDs (Liquid Crystal Displays), bei dem jeder einzelne Bildpunkt – also jeder Pixel – durch einen eigenen Dünnschichttransistor angesteuert wird. Kombiniert mit einem Drehencoder und ein Push-Button, bietet er eine intuitive und präzise Benutzerinteraktion. Im Vergleich zu älteren passiven LCD-Technologien ermöglicht das TFT-Prinzip eine wesentlich präzisere Steuerung der Bildpunkte, wodurch eine deutlich bessere Bildqualität, höhere Auflösungen und schnellere Reaktionszeiten erzielt werden.[23]

Ein Bild, das Kreis, Kompass enthält.

KI-generierte Inhalte können fehlerhaft sein.

Display Encoder 582

|  |  |
| --- | --- |
| **Technische Merkmale** | **TFT-Display** |
| Betriebsspannung | 3.3 V |
| Displaygröße | 1.28 Zoll |
| Auflösung | 240 x 240 Pixel |
| Pixelabstand | 0.135 x 0.135 mm |
| Schnittstelle | SPI-Schnittstelle (4-Leiter) |
| Treiber-IC | GC9A01 |
| Aktive Anzeigefläche | Ø 32.4 mm |
| Gewicht | 8.4 g |
| Betriebstemperatur | -20 °C bis +70 °C |

Technische Daten des angewandten Displays

Selbstverständlich weist diese Displaytechnologie sowohl Vorteile als auch Nachteile auf.

**Vorteile:**

* **Klare Darstellung und hohe Auflösung:** TFTs bieten exakte Anzeige von Text, Grafiken oder Messwerten – von QVGA bis Full HD und darüber hinaus. Die feine Pixelstruktur erlaubt auch bei kleinen Zollgrößen eine detailgetreue Darstellung.[23]
* **Geringe Reaktionszeit für dynamische Inhalte:** Moderne Panels bieten Reaktionszeiten von 8–25 ms, was für die meisten industriellen Anwendungen ausreichend ist.[23]
* **Hohe Helligkeit und gute Ablesbarkeit:** Dank leistungsfähiger Hintergrundbeleuchtung (oft 500–1000 cd/m²) sind moderne TFT-Displays auch bei schwierigen Lichtverhältnissen gut lesbar.[23]
* **Robuste Module für industrielle Bedingungen:** Viele TFT-Module sind für hohe Temperaturbereiche, Schockbelastungen und Langzeitverfügbarkeit konzipiert.[23]

**Nachteile:**

* **Energieverbrauch bei hoher Helligkeit:** Für gute Ablesbarkeit unter Sonnenlicht ist eine starke LED-Hinterleuchtung nötig, was den Stromverbrauch und die thermische Belastung erhöht.[23]
* **Eingeschränkter Schwarzwert und Kontrast:** Da TFTs auf LCD-Technologie basieren, ist echtes Schwarz schwer erreichbar – das Display ist nie vollständig „aus“, da die Hintergrundbeleuchtung immer aktiv ist.[23]

## 3.2 Mikrocontroller

Ein Mikrocontroller (MCU) wird in Abbildung … dargestellt. Es handelt sich dabei um ein komplexes System on Chip (SoC), das auf einem einzigen integrierten Schaltkreis (IC) einen Prozessorkern (Control Process Unit, CPU), Speicher (EEPROM, FLASH und RAM), analoge IPs und viele Ein-/Ausgänge (IOs) integriert.[24]. Mikrocontroller weisen eine Reihe von Vorteilen auf, die ihre Anwendung in verschiedenen Bereichen ermöglichen. Dazu zählen ihre kompakte Größe, die niedrigen Kosten, der geringe Stromverbrauch und die hohe Zuverlässigkeit.[25]. MCUs sind für den Einsatz in eingebetteten Anwendungen konzipiert. Im Gegensatz dazu werden sie mit weniger anspruchsvollen Technologien als Mikroprozessoren entwickelt und arbeiten mit niedrigeren Frequenzen.

Ein Bild, das Text, Screenshot, Farbigkeit, Rechteck enthält.

KI-generierte Inhalte können fehlerhaft sein.

Mikrocontroller Schematisch

Die Entwicklung eines solchen Sytems erfolgt typischerweise in mehreren fachlich getrennten Domänen. Zu den relevanten Aspekten zählen in diesem Zusammenhang das Analogdesign (u. a. A/D- und D/A-Wandler, Spannungsregelung und Überwachung), das Digitaldesign (Prozessor-Kern, Bus- und Kommunikationslogik), das I/O-Design (Anbindung des Chip-Kerns an die Gehäusepins inklusive Schutzbeschaltungen) sowie das SoC-Integrationsdesign, in dem die verschiedenen IP-Blöcke zu einem vollständigen System zusammengeführt werden.

Um den vielfältigen Anforderungen des Marktes gerecht zu werden, wurden verschiedene Mikrocontrollerfamilien entwickelt.

### 3.2.1 STM32-Mikrocontroller Familie

In dieser Arbeit wird der Schwerpunkt auf die Mikrocontroller der STM32F-Serie gelegt, die für Mainstream- und Hochleistungsanwendungen konzipiert wurden.

Die Abbildung … veranschaulicht die diversen Mikrocontrollerfamilien von STMicroelectronics sowie deren jeweilige Merkmale. Ein Mikrocontroller stellt einen programmierbaren integrierten Schaltkreis dar und kann in einem breiten Spektrum von Anwendungen eingesetzt werden. Er findet beispielsweise Anwendung in Haushaltsgeräten wie Waschmaschinen, in Mobiltelefonen, in der Medizintechnik, im Automobilbereich, in der Industrieautomation sowie in der Robotik.

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

KI-generierte Inhalte können fehlerhaft sein.

ST-Mikrocontrollers Familie[26]

Diese Familie des Herstellers STMicroelectronics beinhaltet eine Vielzahl von 32-Bit-Mikrocontrollern, welche auf unterschiedlichen ARM-Cortex-M-Kernen basieren (darunter M0, M3 und M4). Sie zeichnet sich durch eine hohe Rechenleistung, große Flash- und RAM-Kapazitäten, umfangreiche Peripherie (z. B. ADC, DAC, SPI, I2C, UART, USB, CAN) sowie einen weiten Taktfrequenzbereich aus. Aufgrund spezifischer Eigenschaften und Merkmale ist sie insbesondere für HMI-Anwendungen geeignet.

## 3.3 HMI-System

Human-Machine-Interfaces (HMIs) sind zentrale technologische Komponenten, die als Schnittstelle zwischen Menschen und Maschine fungieren. Ihr primärer Zweck besteht darin, den Benutzern den Zugriff auf technische Systeme, Produktionsanlagen oder Prozessleittechnik zu ermöglichen, um Zustände zu überwachen und gezielt auf Prozesse einzuwirken.[27].

## 3.4 CAN-Bus

In diesem Abschnitt wird der Controller Area Network (CAN-Bus) als zentrale Kommunikationsschnittstelle des Gesamtsystems vorgestellt. Da über diesen Bus der Datenaustausch zwischen dem Maschinencontroller und dem HMI-System erfolgt, bildet er einen elementaren Bestandteil der Systemarchitektur. Aus diesem Grund erfolgt an dieser Stelle eine ausführlichere Erläuterung des CAN-Busses.

### 3.4.1 Allgemeine Informationen

Der CAN-Bus (Controller Area Network) wurde ab 1983 von der Robert Bosch GmbH entwickelt, um komplexe Verkabelungssysteme im Fahrzeug deutlich zu vereinfachen und eine effiziente Kommunikation zwischen verschiedenen Steuergeräten zu ermöglichen. Die Erstpräsentation des Systems erfolgte 1986 auf dem SAE-Kongress, während der internationale Durchbruch im Jahr 1991 mit der Veröffentlichung der CAN-Spezifikation Version 2.0 erzielt wurde. Die Normung des Protokolls wurde schließlich durch die ISO vorgenommen und ist gegenwärtig in den Normen ISO 11898-1 bis ISO 11898-5 festgelegt[28]. Aufgrund der hohen Störsicherheit, der geringen Kosten und der Echtzeitfähigkeit wird CAN nicht nur in der Automobilindustrie, sondern auch in vielen anderen Branchen (z. B. in Nutzfahrzeugen, mobilen Arbeitsmaschinen, Eisenbahnen, in der Medizintechnik, in der Industrieautomation, in Aufzügen, und als Maschinenbussystem) eingesetzt.[29].

Das Controller Area Network (CAN) verbindet mehrere gleichberechtigte Komponenten (Knoten, Node) über einen 2-Draht Bus plus zusätzlicher Masseleitung miteinander.[29]

Der CAN-Bus ist als bitstromorientierter Linienbus ausgeführt und unterstützt Datenübertragungsraten von bis zu 1 Mbit/s. Die maximal zulässige Buslänge sowie die Länge der Stichleitungen sind dabei direkt von der gewählten Bitrate abhängig. Innerhalb eines CAN-Netzwerks ist ausschließlich eine einheitliche Bitrate zulässig. Im Falle des Betriebes von Geräten mit unterschiedlichen Bitraten ist der Einsatz mehrerer CAN-Segmente erforderlich, die über ein Gateway miteinander verbunden sind.

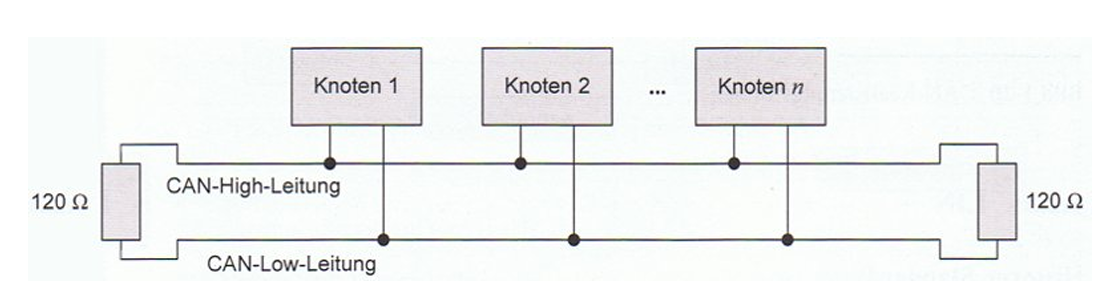
Ein Bild, das Text, Schrift, Reihe, Screenshot enthält.

KI-generierte Inhalte können fehlerhaft sein.

Controller Area Network (CAN)[28]

### 3.4.2 Verschiedene Varianten der CAN-Busses

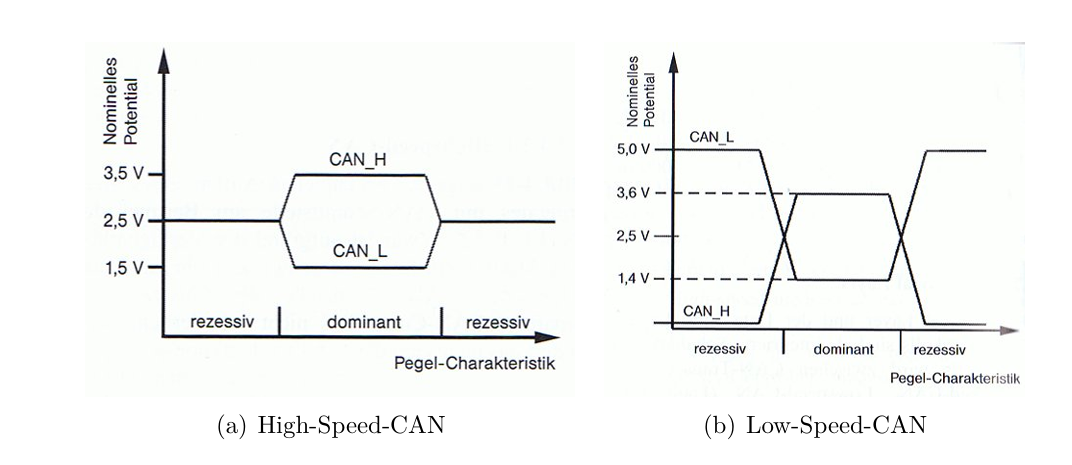
Der CAN-Bus gibt es in zwei Varianten, die sich in ihren physikalischen Eigenschaften und Einsatzgebieten unterscheiden.



High-Speed-CAN-Bus[30, S. 19]

Die erste Variante ist der High-Speed-CAN, der für schnelle Datenübertragungen konzipiert ist. Er wird vor allem im automobilen Umfeld für die Vernetzung von Komponenten des Antriebsstrang eingesetzt und arbeitet mit einer Bitrate von 500 kbit/s. Die Übertragung erfolgt mittels einer verdrillten Zwei-Draht-Leitung, die eine Signalübertragung ermöglicht. An den Enden wird der Bus mit 120 Ω abgeschlossen. In Abbildung 2.2 ist der Aufbau eines High-Speed-CAN-Systems dargestellt.

Die zweite Variante ist der Low-Speed-CAN, der vorwiegend im Bereich der Komfort- und Karosserieelektronik mit geringeren Bitraten von 125 kbit/s eingesetzt wird, wobei 100 kbit/s üblich sind. Eine Bitrate von 100 kbit/s wird zum Beispiel durch eine fest eingestellte Komponente bestimmt. Auch beim Low-Speed-CAN wird eine verdrillte Zwei-Draht-Leitung für die Übertragung genutzt, wobei keine Abschlusswiderstände erforderlich sind. Die Datenübertragung erfolgt ebenfalls über differenzielle Signale. Ein wesentlicher Unterschied besteht darin, dass der Low-Speed-CAN den Ausfall einer der beiden Leitungen tolerieren kann. In diesem Fall wird die verbleibende Leitung gegen Masse ausgewertet, sodass die Kommunikation weiterhin aufrechterhalten wird.



Signalpegel des CAN-Bus[31, S. 206]

Die Verwendung einer verdrillten Zwei-Drahtleitung trägt zur hohen Störfestigkeit des CAN-Busses bei. Die Ausfilterung elektromagnetischer Störungen erfolgt über die Auswertung des Differenzsignals.

Die Datenübertragung auf dem CAN-Bus erfolgt über zwei Signalzustände, die dominanten und rezessiven Pegel. Eine logische Null entspricht dem dominanten und eine logische Eins dem rezessiven Pegel. Im Busystem setzt sich der dominante Pegel stets durch. Die konkreten Signalpegel unterscheiden sich je nach verwendeter CAN-Variante. Beim High-Speed-CAN sind beide Leitungen im rezessiven Zustand bei 2,5 V, das Differenzsignal beträgt 0 V. Im dominanten Zustand liegt der Pegel der CAN-H-Leitung bei 3,5 V, die CAN-L-Leitung wird auf 1,5 V abgesenkt, das Differenzsignal beträgt 2 V. Beim Low-Speed-CAN sind andere Pegel definiert: Im rezessiven Zustand liegt die CAN-H-Leitung bei 0 V, die CAN-L-Leitung bei 5 V. Im dominanten Zustand ändern sich die Pegel auf 3,6 V für CAN-H und 1,4 V für CAN-L, das Differenzsignal beträgt bis zu 5 V.[31]

### 3.4.3 Aufbau der CAN-Nachrichten

Der Austausch von Nachrichten auf dem CAN-Bus basiert auf der CAN-Spezifikation 2.0.[28]

In der Aktuelle Version gibt es zwei verschiedene Formate des Daten-Frames. Die beiden Formate unterscheiden sich in der Länge ihrer Nachrichten-IDs. Das Standard-Format verwendet einen 11 Bit langen Identifier für 2048 Nachrichten. Wegen der großen Akzeptanz von CAN und dem Einsatz in vielen Bereichen musste eine feste Zuordnung von Nachrichten-IDs zu Funktionen geschaffen werden. Das Extended-Format verwendet einen 29 Bit langen Identifier.[28]

Das Standard-Format wird eingesetzt, da es für die Anzahl der Nachrichten ausreichend ist.

Im Folgenden werden die Daten-Frames genauer vorgestellt und kurz auf den Remote eingegangen.

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

KI-generierte Inhalte können fehlerhaft sein.

Der Daten-Frame im Standard-Format aus Abbildung… beginnt mit einem **SOF**-Bit, das den Beginn des Frames kennzeichnet und immer einen dominanten Pegel besitzt. Es folgen die 11 Bit der Nachrichten-**ID** zur eindeutigen Identifizierung des Daten-Frame.

Das nächste Bit ist das **R**emote-**T**ransmission-**R**equest-Bit zur Markierung der Nachricht als Remote-Frame. In normalen Daten-Frames ist dieses Bit dominant, damit sich ein Daten-Frame bei der Arbitrierung immer gegen einen zugehörigen Remote-Frame durchsetzt.

Das Identifier-Extension-Bit zeigt an, ob es sich um eine Nachricht im Standard- oder im Extended-Format handelt. Beim Einsatz des Standard-Formats hat dieses Bit immer einen dominanten Pegel.

Das **r0**-Feld ist ein reserviertes Bit, das für spätere Erweiterungen vorgesehen ist und einen dominanten Pegel besitzt.

Das anschließende **DLC**-Feld (Data-Length-Code) zeigt die Anzahl der Bytes an Daten in der Nachricht an. Dabei sind Werte zwischen 0 Byte und 8 Byte möglich.

Nach dem DLC-Feld folgt das **DATA**-Feld mit den Daten der Nachricht und einer variablen Länge. Darauf folgt das **CRC**-Feld (Cyclic Redundancy Check) mit der Prüfsumme der Nachricht.

Das **ACK**-Feld (Acknowledgement) hat eine besondere Bedeutung. Der Sender legt ein Bit auf den Bus, der Empfänger ein anderes. Wenn der Sender das erkennt, weiß er, dass die Nachricht von mindestens einem Empfänger empfangen wurde. Ist das Signal nicht dominant, sendet der Sender die Nachricht erneut.

Das 7-Bit-Feld "**E**nd-**o**f-**F**rame" zeigt das Ende der Nachricht an und besitzt immer einen rezessiven Pegel.

## 3.5 SPI

### 3.5.1 Allgemein

Das Serial Peripheral Interface (SPI) ist eine von Motorola entwickelte Schnittstelle zur Kommunikation zwischen digitalen Schaltungen. Sie arbeitet nach dem Master-Slave-Prinzip, wobei der Mikrocontroller die Master-Rolle übernimmt und die Peripherie-Bausteine die Slaves sind. SPI ist lizenzfrei und stellt keinen genormten Standard dar, da lediglich die Hardware-Funktionsweise, nicht jedoch ein verbindliches Protokoll spezifiziert ist. Trotz dieser fehlenden Standardisierung hat sich SPI als weit verbreitete Schnittstelle etabliert, insbesondere für schnelle, synchrone Datenübertragungen zwischen Mikrocontrollern und integrierten Schaltungen.

### 3.5.2 Prinzip

Das Serial Peripheral Interface, kurz SPI, ist ein Bussystem, das für den schnellen, synchronen Datenaustausch zwischen einem Mikrocontroller und mehreren integrierten Schaltungen entwickelt wurde.

Es besteht aus separaten Leitungen für Daten und Takt. Für die bidirektionale Datenübertragung werden die Leitungen MOSI (Master Out Slave In) und MISO (Master In Slave Out) verwendet, während die Synchronisation über die vom Master erzeugte Taktleitung SCK erfolgt. Zusätzlich wird für jeden Slave eine Chip-Select-Leitung (CS) eingesetzt, die aktiv-low arbeitet und die gezielte Auswahl des jeweiligen Slave-Bausteins ermöglicht. In einem SPI-System existiert genau ein Master, der den Kommunikationsablauf steuert und festlegt, mit welchem Slave kommuniziert wird.

Ein Bild, das Diagramm, Text, Plan, Reihe enthält.

KI-generierte Inhalte können fehlerhaft sein.

Die Datenübertragung erfolgt voll-duplex, wobei gleichzeitig Daten vom Master zum Slaven und umgekehrt übertragen werden. Da kein einheitliches Protokoll definiert ist, müssen zentrale Kommunikationsparameter wie Bitreihenfolge, Taktpolarität und Taktphase zwischen Master und Slave abgestimmt werden. Durch die Kombination dieser Parameter ergeben sich vier mögliche SPI-Betriebsmodi. Die maximal mögliche Taktrate ist prinzipiell flexibel, wird jedoch durch die elektrischen Eigenschaften und die zulässigen Grenzwerte der beteiligten Bausteine begrenzt.

Für das Serial Peripheral Interface (SPI) gibt es vier Betriebsmodi, die sich aus der Kombination der beiden Parameter Clock Polarity (CPOL) und Clock Phase (CPHA) ergeben.

|  |  |  |
| --- | --- | --- |
| **SPI-modi** | **CPOL** | **CPHA** |
| 0 | 0 | 0 |
| 1 | 0 | 1 |
| 2 | 1 | 0 |
| 3 | 1 | 1 |

Die Datenübernahme erfolgt bei der steigenden Taktflanke, wenn CPOL und CPHA denselben Wert besitzen, andernfalls bei der fallenden Taktflanke. Pro Taktzyklus wird ein Datenbit übertragen, für ein Byte sind acht Taktzyklen erforderlich. Eine Übertragung beginnt mit der Aktivierung der Chip-Select-Leitung (LOW-Pegel) und endet mit deren Deaktivierung (HIGH-Pegel).

Ein Bild, das Diagramm, Reihe, Muster enthält.

KI-generierte Inhalte können fehlerhaft sein.

### 3.5.3 Datenübertragung

Nach Konfiguration der SPI-Schnittstelle erfolgt die Datenübertragung gemäß dem Master-Slave-Prinzip. Zunächst stellt der Master eine Taktfrequenz ein, die die maximal zulässige Übertragungsrate aller angeschlossenen Slave-Geräte nicht überschreitet. Anschließend wird der gewünschte Slave durch Aktivierung der Chip-Select-Leitung (CS = LOW) ausgewählt. Abhängig vom jeweiligen Peripheriegerät kann vor Beginn der Datenübertragung eine definierte Wartezeit erforderlich sein.

Die Datenübertragung erfolgt synchron zum Taktsignal über Schieberegister. In der Regel wird das höchstwertige Bit (MSB) zuerst übertragen, wobei ein Datenwort typischerweise 8 Bit umfasst. Während eines Taktzyklus sendet der Master Daten über die MOSI-Leitung an den Slave, während der Slave gleichzeitig über die MISO-Leitung Daten an den Master zurücksendet. SPI arbeitet im Vollduplex-Betrieb, sodass Senden und Empfangen simultan stattfinden.

Nach Abschluss der Übertragung werden Statusregister oder Interrupts verwendet, um das Ende der Kommunikation zu erkennen. Abschließend wird die CS-Leitung wieder deaktiviert (CS = HIGH), wodurch der Slave vom Bus getrennt wird. Die empfangenen Daten stehen im selben Register zur Verfügung, das auch für den Versand der Daten genutzt wurde.

Ein Bild, das Text, Diagramm, Reihe, parallel enthält.

KI-generierte Inhalte können fehlerhaft sein.

SPI Master/Slave Übertragungsblock Diagramm.

## 3.6 Interne FLASH \_MEMORY eines STM32-Mikrocontroller

### 3.6.1 Allgemein Grundlage nicht-flüchtiger Speicher

Nicht-flüchtige Speicher (Non-Volatile Memory, NVM) sind digitale Speicherbausteine, die Daten dauerhaft speichern können, ohne dass eine konstante Stromversorgung erforderlich ist. Sie sind ideal für die Speicherung von Programmen, Kalibrierungsdaten und Konfigurationsparametern in eingebetteten Systemen wie dem HMI-System[32].

Flash-Speicher ist eine Weiterentwicklung des EEPROM (Electrically Erasable Programmable Read-Only Memory) und wird auch Flash-EEPROM genannt. Im Gegensatz zum klassischen EEPROM löscht Flash-Speicher blockweise, was zu schnelleren Lösch- und Schreibgeschwindigkeiten führt[33].

### 3.6.2 Architektur und Funktionsweise von Flash-Speicher

Flash-Speicher basiert auf einer Feldeffekttransistor-Matrix mit isoliertem Steueranschluss (Floating-Gate-Struktur). Während des Programmierens wird eine elektrische Ladung auf das Floating Gate gebracht, die dort isoliert verbleibt und nur durch einen Löschprozess wieder entfernt werden kann[34].

Es gibt zwei Hauptvarianten von Flash-Speicher[35]:

**NOR-Flash-Speicher**: Parallel verschaltete Speicherzellen ermöglichen direkten Zugriff auf einzelne Bytes mit kurzen Lese- und Schreibzeiten. NOR-Flash wird typischerweise als Programmspeicher in Mikrocontrollern eingesetzt.

**NAND-Flash-Speicher**: In Reihe geschaltete Speicherzellen ermöglichen schnelleres Schreiben und Löschen großer Datenblöcke mit höherer Speicherdichte. NAND-Flash wird häufiger in großkapazitiven Speichern wie USB-Sticks und SSDs verwendet.

### 3.6.3 Programmiertechnik und Zugriff

Im Rahmen dieser Arbeit erfolgt der Zugriff auf den Flash-Speicher des STM32 über die Hardware-Abstraktionsschicht (HAL). Diese fasst die im Programming Manual beschriebenen Programmierabläufe zusammen. Zunächst werden die Flash-Kontrollregister für Schreibzugriffe entsperrt. Anschließend werden – abhängig von der verwendeten Mikrocontroller-Serie – komplette Speicherbereiche gelöscht und danach wortweise neu beschrieben. Lesezugriffe erfordern keine besondere Vorbereitung, da der interne Flash in den linearen Adressraum des STM32 eingebunden ist und der dort abgelegte Programminhalt wie normaler Speicher über Zeigerzugriffe gelesen werden kann. Die verwendeten Routinen für das Schreiben und Lesen von Konfigurationsdaten werden im Kapitel ... anhand von Codeausschnitten vorgestellt.

Das EEPROM garantiert mindestens 100.000 Schreib-/Löschzyklen pro Byte, kann aber deutlich höher liegen (bis zu 1 Million Zyklen).[36]

Der STM32F429 spezifiziert eine Flash-Endurance von typischerweise 10.000 Schreibzyklen pro Seite.[37]

# 4. Konzept

## 4.1 Entwurf der Platine

### 4.1.1 Anforderungen

Bevor die Leiterplatte entworfen wurde, die den zentralen Bestandteil des HMI-Systems bildet, wurden die grundlegenden elektrischen und mechanischen Anforderungen definiert. Diese lassen sich wie folgt zusammenfassen:

* Signalintegrität: Die Leiterplatte ist so auszulegen, dass eine störungsfreie Übertragung der digitalen und analogen Signale gewährleistet ist.
* Mechanische Befestigung: Für die Montage auf der Rückseite des Displays sind mechanische Befestigungsbohrungen zu integrieren.
* Status-LEDs: Zur Anzeige systemrelevanter Zustände ist der Einsatz zweier LEDs vorgesehen:

eine Orange LED zur Darstellung der anliegenden Betriebsspannung und

eine rote LED, die während eines Firmware-Updates blinkt und somit den Aktualisierungsvorgang signalisiert.

* Firmware-Update-Schnittstellen: Für das Aktualisieren der Firmware sollen zwei Möglichkeiten implementiert werden:

ein USB-Anschluss zum Hochladen neuer Firmware und ein zusätzlicher programmierbarer Steckverbinder, über den mithilfe eines ST-Link-V3-Sets ein Firmware-Update durchgeführt werden kann. Diese Redundanz erhöht die Flexibilität bei Service- und Wartungsprozessen.

* Kommunikationsschnittstellen: Für die Kommunikation mit der Maschinensteuerung ist eine CAN-Bus-Schnittstelle vorgesehen, da diese besonders robust ist und Fehlertoleranz bietet. Zusätzlich wird eine SPI-Schnittstelle integriert, die einen schnellen Datenaustausch zwischen Mikrocontroller und TFT-Display ermöglicht.
* Hardware-Reset: Durch das Drücken auf dem Taster soll die Firmware neu gestartet werden.

### 4.1.2 Hardwaredesign

Die Abbildung zeigt das Blockdiagramm des entwickelten Micro-HMI-Systems und veranschaulicht den Aufbau sowie die Schnittstellen zwischen den Systemkomponenten. Zentrales Element des Systems ist der Mikrocontroller, der die Kommunikation mit der Maschinesteuerung sowie die Ansteuerung des Displays und der Peripherie übernimmt.

Ein Bild, das Text, Diagramm, Screenshot, Plan enthält.

KI-generierte Inhalte können fehlerhaft sein.

Die Spannungsversorgung des HMI-Systems erfolgt über die Maschinesteuerung, welche eine Versorgungsspannung von 5 V bereitstellt. Diese Spannung wird auf der HMI-Platine durch einen Spannungsregler auf 3,3 V umgesetzt, da sowohl der Mikrocontroller als auch das TFT-Display mit dieser Betriebsspannung arbeiten. Eine Power-LED(Orange) signalisiert den Versorgungszustand der Platine.

Das TFT-Display ist über eine SPI-Schnittstelle mit dem Mikrocontroller verbunden und wird zusätzlich über digitale I/O-Signale gesteuert. Eine System-LED dient zur Signalisierung des erfolgreichen Programmstarts. Die Technikerinteraktion erfolgt über einen Reset-Taster, der den Firmware-Neustart ermöglicht.

Die Kommunikation zwischen HMI-System und Maschinesteuerung erfolgt über den CAN-Bus, welcher robuste und echtzeitfähige Datenübertragung gewährleistet. Ein externer Quarzoszillator mit einer Frequenz von 24 MHz dient der präzisen Takterzeugung und zuverlässigen zeitlichen Ausführung der Software.

Zwei Programmierschnittstellen ermöglichen die Wartung und den Service: Die JTAG-Schnittstelle in Verbindung mit einem ST-LINK-Debugger dient der Firmwareentwicklung, -programmierung und -debugging. Ein USB-Anschluss in Kombination mit dem BOOT0-Signal ermöglicht die Firmware-Aktualisierung über den integrierten USB-DFU-Bootloader, ohne dass ein externer Programmierer erforderlich ist.

Das dargestellte Hardwaredesign zeichnet sich durch eine klare Struktur, modulare Erweiterbarkeit und hohe Servicefreundlichkeit aus. Es bildet die Grundlage für den zuverlässigen Betrieb des HMI-Systems und ermöglicht eine einfache Integration in die bestehende Maschinenarchitektur sowie zukünftige Anpassungen und Erweiterungen.

### 4.1.3 Leiterplattenentwurf

Im Rahmen der Arbeit wurde Altium Designer zur Entwicklung der Leiterplatte eingesetzt. Die wesentlichen Bauteile, ihre Funktionen sowie ihre Platzierung werden nachfolgend erläutert.

**Auswahl der wichtigen Technologie:**

* Für das HMI-System wird ein Mikrocontroller der STM32F-Serie verwendet, da dieser bereits im Entwicklungsumfeld etabliert ist und eine hohe Rechenleistung sowie Verfügbarkeit bietet. In Kapitel 3.2 werden die verschiedenen Familien der STM32-Serie vorgestellt und grob miteinander verglichen. Der STM32F429VIT6 wurde als geeignet identifiziert. Er arbeitet mit einer Taktfrequenz bis 180 MHz, verfügt über bis zu 2 MB internen Flash-Speicher und bietet mehrere Kommunikationsschnittstellen. Die MCU unterstützt externe Quarzoszillatoren im Bereich von 4 bis 26 MHz, was eine stabile Taktung und präzise Zeitmessung ermöglicht. Diese Eigenschaften prädestinieren den STM32F429 für den Einsatz in einem eigenständigen, grafikorientierten HMI-System mit Echtzeitanforderungen.
* Die Auswahl der Kommunikationsschnittstelle zwischen der Maschinesteuerung und dem HMI-System wurde durch das Unternehmen vorgegeben. Als Schnittstelle wurde der CAN-Bus festgelegt, da dieser eine hohe Störsicherheit und zuverlässige Datenübertragung gewährleistet. Der CAN-Bus ist in der Maschinenarchitektur integriert, was eine einfache Systemintegration und Reduzierung des Verkabelungsaufwands ermöglicht.
* Das TFT-Display wurde vom Auftraggeber (IN Groupe / SURYS) zur Verfügung gestellt und ist für die Integration in die PASS-Laser-Engraver-Maschine vorkonfiguriert. Detaillierte Beschreibungen zu Display-Spezifikationen, Schnittstellen und Kalibrierungsparametern finden sich in Kapitel 3.1.

Ein Bild, das Text, Diagramm, Plan, Zahl enthält.

KI-generierte Inhalte können fehlerhaft sein.

Die Energieversorgung der Platine wird durch den CAN-Bus-Stecker sichergestellt, welcher mit der Maschinensteuerung verbunden ist. Über den ersten Pin werden +5 V bereitgestellt und an die Leiterplatte weitergeleitet. Da der verwendete Mikrocontroller jedoch mit einer Betriebsspannung von 3,3 V arbeitet, muss diese Spannung über einen geeigneten Spannungswandler reduziert werden. Die Abbildung… zeigt die Schaltung der Spannungswandler. Diese Verschaltung des Wandlers wurden gemäß den Vorgaben des jeweiligen Datenblatts ausgelegt. Am Ausgang des Spannungswandlers wurde eine grüne Leuchtdiode installiert, die den Zustand der Versorgungsspannung visuell anzeigt.

Die Abbildung… zeigt den Stromversorgungsbereich des Mikrocontroller. Stromversorgungseingangspins zum Mikrocontroller werden als VDD (sowie VBAT) bezeichnet, womit die digitale Spannungsschiene gemeint ist. Für jeden VDD-Pins wird ein 100-nF-Entkopplungskondensator parallel geschaltet, der in der Nähe der entsprechenden VDD- und VSS- (oder GND-) Pins platziert wird, um Lastsprünge abzufangen und eine stabile Versorgungsspannung sicherzustellen. Der VBAT-Pin, der zur Versorgung der integrierten Echtzeituhr (RTC) vorgesehen ist, wird nicht dauerhaft mit der Versorgungsspannung verbunden, sondern über einen Jumper geführt. Dadurch bleibt die Möglichkeit erhalten, die RTC-Funktion bei Bedarf flexibel zu- oder abzuschalten, da sie im aktuellen Einsatzfall nicht unmittelbar erforderlich ist. Darüber hinaus verfügt der Mikrocontroller über analoge Peripheriegeräte, wie zum Beispiel einen Analog-Digital-Wandler, der ebenfalls Strom benötigt. Diese Versorgungseingang (VDDA und VREF +) werden ebenfalls über ein 100nF-Entkopplungskondensator mit 3.3 V angeschlossen. Der Mikrocontroller verfügt über interne Regler, die extern mit weiteren Entkopplungskondensatoren umgangen werden müssen. Diese Pins werden als VCAP bezeichnet. Der Wert von 2.2 µF pro Pin wurde aus dem entsprechenden Datenblatt entnommen.

Im Hinblick auf STM32-Mikrocontroller existieren prinzipiell zwei Methoden, um den Programcode auf den Mikrocontroller zu übertragen. Die beide Methode werden in der Platine eingefügt.

Die erste Möglichkeit erfolgt über den integrierten Bootloader, der bei STM32-Bausteinen das Laden von Firmware über USB-Schnittstelle ohne den Einsatz eines Debuggers ermöglicht. Die Aktivierung des Bootloaders erfordert den BOOT0-Pin auf der High-Pegel zu legen, und dies hat vor dem Einschalten des Systems zu erfolgen. Für das anschließende Ausführen des Anwenderprogramms ist der Pin wieder auf Low zu setzen, bevor der Mikrocontroller erneut gestartet wird. Um eine flexible Durchführung dieser Umschaltung im Entwicklungsprozess zu gewährleisten, wurde der BOOT0-Pin über einen Switch geführt. Infolgedessen kann der Betriebsmodus des Mikrocontrollers zwischen Bootloader-Modus und normalem Applikationsmodus umgeschaltet werden.

Die Geräte verfügen über ein integriertes USB-Full-Speed-Device. Dies wurde gemäß der USB-2.0-Spezifikation durchgeführt.

Die USB-Stromversorgung (VBUS) wird durch eine Kombination aus zwei Keramikkondensatoren (C3 und C4) und einer Ferritperle (FB1) gefiltert. Die Filterkette dient der Elimination von hochfrequentem Rauschen und Spannungsspitzen aus der USB-Stromleitung, da der VBUS-Pin eine hohe Empfindlichkeit gegenüber Störungen aufweist. Dadurch ist der Mikrocontroller in der Lage, sowohl das Anschließen als auch das Abziehen von USB-Geräten zuverlässig zu erkennen. Darüber hinaus ist er auch in der Lage, Spannungsschwankungen zu detektieren. Die USB-Datenleitungen D+ und D− werden als sogenanntes differentielles Leitungspaar (Differential Pair) geführt. Zunächst wird eine ESD-Schutzschaltung passiert, die typischerweise als TVS-Diode (Transient Voltage Suppression) ausgeführt ist und zum Schutz vor elektrostatischen Entladungen dient. Im Anschluss durchqueren die Datenleitungen einen Common Mode Choke, dessen Funktion darin besteht, Gleichtakt Störungen zu unterdrücken. Der Spannungspegel wird durch serielle Widerstände auf etwa 3,3 V begrenzt, was der Eingangsspannung des Mikrocontrollers entspricht. Das Metallgehäuse der USB-Buchse wird über einen 100-nF-Kondensator mit dem Massepotential verbunden, um Störströme zu filtern und ein stabil Betrieb zu gewährleisten.

Die zweite Methode erfolgt über ein STLINK-V3 und JTAG-DebugSchnittstelle der Mikrocontroller, bei dem es sich um eine fünfdrahtschnittstelle handelt. Die Tabelle xx veranschaulicht die verschiedenen Pins und ihre Funktion.

|  |  |  |
| --- | --- | --- |
| **JTAG pin Name** | **Type** | **Description** |
| JTMS | I | JTAG Test Mode Selection |
| JTCK | I | JTAG Test Clock |
| JTDO | I | JTAG-Test Data Input |
| JTDI | O | JTAG-Test Data Output |
| NJTRST | I | JTAG-Test nReset |

JTAG debug port pins[38].

Das STLINK-V3 ist eine eigenständige, modulare Debugging- und Programmier-Sonde, die für die Verwendung mit STM32-Mikrocontrollern konzipiert wurde. Die Unterstützung von SWIM- und JTAG/SWD-Schnittstellen für die Kommunikation mit jedem STM32-Mikrocontroller auf einer Anwendungsplatine.

Ein Bild, das Elektronik, Elektronisches Bauteil, Elektrisches Bauelement, passives Bauelement enthält.

KI-generierte Inhalte können fehlerhaft sein.

Das JTAG-Protokoll ist ein Debug-/Programmprotokoll, das für STM32-Mikrocontroller verwendet wird.

Die Signale sind 3,3-V-kompatibel und können bis zu 21 MHz leisten. Für einen korrekten Betrieb benötigt der STLINK-V3 einen JTAG-Rücklauf-Takt. Standardmäßig wird dieser Rücklauf-Takt über den geschlossenen Jumper JP1 auf dem STLINK bereitgestellt, kann aber auch extern bereitgestellt werden. Diese Konfiguration ist erforderlich, um hohe JTAG-Frequenzen zu erreichen; in diesem Fall muss JP1 auf dem STLINK geöffnet werden. Die Signale werden über einen Stecker an der Platine direkt an den entsprechenden Pins am Mikrocontroller angeschlossen.

Obwohl die Mehrheit der Mikrocontroller über einen internen Oszillator verfügt, der für eine Vielzahl von Designs ausreichend ist, wird im Projekt einen externen Quarz (oder Oszillator) an den Mikrocontroller anzuschließen.

Insbesondere im Hinblick auf STM32-Geräte existieren zwei Arten von Oszillatoren: die sogenannte External High-Speed (HSE) für allgemeine Timing-Zwecke und die External Low-Speed (LSE) für die Echtzeituhr. Bezüglich der Anforderung wurde sich für ein HSE entschieden.

Ein Bild, das Diagramm, technische Zeichnung, Reihe, Plan enthält.

KI-generierte Inhalte können fehlerhaft sein.

Pierce-Oszillatorschaltung[39]

Das oben dargestellte Bild veranschaulicht die bewährte Methode zur Verbindung eines externen Quarzes mit dem Mikrocontroller. Ein externer Quarz-Oszillator generiert ein präzises, periodisches Taktsignal in Form einer Rechteckschwingung mit Frequenz von 24 MHz. Es ist erforderlich, Belastungskondensatoren (CL) (in diesem Beispiel C6 und C7) hinzuzufügen, deren Kapazität von der im Datenblatt des Quarzes angegebenen Lastkapazität abhängt. Nach Ermittlung der Lastkapazität des Quarzes ist zunächst eine Subtraktion von 3 bis 5 pF Streukapazität erforderlich. Daraufhin ist eine Multiplikation mit zwei notwendig, um den erforderlichen Kapazitätswert zu ermitteln.

CL = 10 pF, CS = 5 pF.

Die Aufgabe dieses Widerstands besteht darin, den Ansteuerungspegel des Quarzes zu begrenzen. Zusammen mit CL2 bildet er einen Tiefpassfilter, der den Oszillator zwingt, bei der Grundfrequenz und nicht bei den Obertönen zu starten (verhindert, dass der Oszillator bei den ungeraden Harmonischen der Grundfrequenz schwingt). Wenn die Verlustleistung im ausgewählten Quarz geringer ist als der vom Quarzhersteller angegebene Ansteuerungspegel, wird der Einsatz von RExt nicht empfohlen, und sein Wert beträgt dann 0 Ω.[39]

Nach Abschluss der Schaltplanerstellung wurde das Leiterplattenlayout (Routing) durchgeführt. Dabei lag der Schwerpunkt auf einer sauberen Versorgung des Mikrocontrollers, einer störungsarmen Führung der Takt‑ und Hochgeschwindigkeitssignale sowie einer übersichtlichen Struktur des Gesamtdesigns.

Die Entkopplungskondensatoren der Mikrocontroller-Versorgung wurden nah am jeweiligen VDD-Pin platziert, um parasitäre Induktivitäten zu minimieren und hochfrequente Störungen lokal kurzschließen zu können. Der Quarzoszillator und seine Lastkondensatoren wurden ebenfalls nah am Mikrocontroller platziert, um die Signalqualität des Takts nicht zu beeinträchtigen.

Aufgrund der funktionalen Komplexität wurde eine vierlagige Leiterplatte realisiert. Eine Lage (Layer 3) wurde als durchgehende Massefläche(GND) ausgeführt, eine weitere( Layer 2) als Versorgungslage für die Hauptspannungen(+3.3V), während die verbleibenden Lagen(Layer 1 und 2) überwiegend der Signalführung dienten. Die nachfolgende Abbildung zeigt den Schichtaufbau der vierlagigen Leiterplatte.

Ein Bild, das Text, Screenshot, Zahl, Spielstandanzeige enthält.

KI-generierte Inhalte können fehlerhaft sein.

Kritische Versorgungsnetze wurden mit erhöhten Leiterbahnbreiten oder großflächigen Kupferflächen geführt, um Spannungsabfälle zu minimieren und ausreichende Stromtragfähigkeit sicherzustellen.

Für die USB-Datenleitungen D+ und D- wurde ein differenzielles Routing mit 90 Ω Impedanz angewendet. Beide Leitungen wurden mit konstanter Leiterbahnbreite, definiertem Abstand und möglichst identischer Länge geführt, um Laufzeitdifferenzen, Reflexionen und Gleichtaktstörungen zu reduzieren.

Weniger empfindliche Baugruppen konnten dagegen mit größerem Abstand ohne nennenswerten Einfluss auf das Systemverhalten platziert werden.

## 4.2 Softwareentwicklung

Aufgrund verlängerter Lieferzeiten bei der Leiterplattenfertigung konnte die Zielhardware zunächst nicht zur Verfügung gestellt werden. Die Entwicklungsplattform mit dem Ersatzboard STM32F756 wurde daher für die Softwareentwicklung verwendet.

In diesem Unterkapitel wird die Entwicklung der Software für das HMI-System erläutert. Die Entwicklung unterteilt sich in die Anforderungen, wo die Systemfunktionen, Schnittstellen, Reaktionszeit sowie Anforderungen an Bedienbarkeit, Zuverlässigkeit und Wartbarkeit festgelegt wird.

### 4.2.1 Anforderungen

Die Software des HMIs für das Pass-Laser-Engraver muss folgende funktionale und nicht funktionale Anforderungen erfüllen.

**Funktionale Anforderungen:**

* Die Software des HMI-Systems muss relevante Maschinenzustände (Betriebszustände, Prozessfortschritte, Warn- und Fehlermeldungen) visualisieren.
* Die Software erfasst und interpretiert Benutzereingaben (Drehencoder und integrierter Push-Button) und generiert die entsprechenden Steuerbefehle.
* Die Software muss Datenpakete von der Maschinesteuerung über CAN-BUS empfangen und verarbeiten, um die Benutzeroberfläche des TFT-Displays über SPI-Schnittstelle entsprechend zu aktualisieren.
* Eine logisch aufgebaute Menüstruktur ist bereitzustellen, die den Zugriff auf unterschiedliche Anzeigeseiten und Funktionen erlaubt.

**Nicht funktionale Anforderungen:**

* Die Software soll eine hohe Reaktionsgeschwindigkeit aufweisen, um Benutzerinteraktionen und Statusänderungen der Maschine ohne Verzögerung zu verarbeiten und anzuzeigen.
* Die Software soll eine übersichtliche, benutzerfreundliche grafische Oberfläche bereitstellen, die eine intuitive Bedienung des HMI-Systems ermöglicht.

### 4.2.2 Softwaredesign

Die Anforderungen des HMI-Systems sind im Use-Case Diagramm (Abbildung …) dargestellt. Es gibt zwei Rollen: Kunde und Techniker.

Ein Bild, das Text, Diagramm, Kreis, Reihe enthält.

KI-generierte Inhalte können fehlerhaft sein.

Use Case Diagramm

Das HMI-System wird nach Benutzerrollen genutzt. Der Kunde nutzt es zur Überwachung und grundlegenden Funktionen. Er zeigt den aktuellen Maschinenzustand(Ready, Warning, Error) an und kann während des Prozesses zusätzliche jobbezogene Informationen anzeigen. Das HMI-System zeigt auch relevante Systeminformationen an, wie Temperatur, Luftfeuchtigkeit, Firmware und Seriennummer. Eine zentrale Funktion ist das gezielte Auswerfen eines Passes.

Der Techniker nutzt das HMI-System für Wartungs- und Servicezwecke. Dazu gehören Firmware-Updates, die über USB-DFU oder JTAG erfolgen. Für ein Update über USB-DFU ist die Aktivierung des Bootloaders nötig.

Die klare Trennung der Benutzerrollen trägt zur Betriebssicherheit des HMI-Systems bei und ermöglicht eine kontrollierte Wartung.

Dieses Diagramm ist die Grundlage für die Implementierung der Menülogik und der Systemfunktionen. Die Menüführung wie in Abbildung … dargestellt, folgt dem Prinzip einer zustandsbasierten Navigation und ist auf die Nutzung eines Drehencoders mit integrierter Push-Button-Funktion ausgelegt.

### Ein Bild, das Text, Diagramm, Plan, technische Zeichnung enthält. KI-generierte Inhalte können fehlerhaft sein.

Menüstruktur

**Hauptmenü und Navigation**

Die Benutzeroberfläche hat ein Hauptmenü mit den drei Menüpunkten Startseite, auf die die Maschinenzustände angezeigt werden und/oder die Jobs bezogene Informationen, Funktionen und Infos. Die Navigation innerhalb des Hauptmenüs erfolgt durch Drehen des Encoders nach links oder rechts, wobei zyklisch zwischen den Menüpunkten gewechselt werden kann. Die Auswahl eines Menüpunktes wird durch Betätigung des Push-Buttons ausgelöst.

Vom Hauptmenü aus gibt es zwei Untermenüs.

**Untermenüs**

Im Untermenü 1 (Funktionen) wird eine Maschinenfunktion bereitgestellt, konkret das Auswerfen eines Passes. Diese Funktion ist ohne weitere Unterebenen implementiert, um eine schnelle und fehlerarme Ausführung zu gewährleisten.

Das Untermenü 2 (Infos) dient der Informationsanzeige. Hier können Systemparameter wie Temperatur und Luftfeuchtigkeit eingesehen werden. Die Navigation zwischen den Informationsseiten erfolgt ebenfalls über den Drehencoder, während der Push-Button zur Rückkehr in das übergeordnete Menü verwendet wird.

### 4.2.3 Softwareimplementierung

**Auswahl der Technologie:**

Die Software für das HMI-System wurde in der integrierten Entwicklungsumgebung STM32CubeIDE unter Verwendung der Programmiersprache C entwickelt. Diese Kombination bietet Unterstützung für die eingesetzten STM32-Mikrocontroller und eine effiziente Implementierung eingebetteter Systeme. STM32CubeIDE stellt umfangreiche Debug- und Analysewerkzeuge zur Verfügung.

Für die nicht-flüchtige Speicherung der von der Maschinensteuerung empfangenen Daten auf dem STM32F429 wurden zwei Speichertechnologien betrachtet: EEPROM und Flash Memory. EEPROM erlaubt einen byteweisen Schreib- und Löschzugriff mit hoher Schreibzyklenfestigkeit( bis zu 1 Million Zyklen), ist jedoch langsamer, teurer und meist nur mit geringer Kapazität(4KBytes) verfügbar. Flash Memory arbeitet blockweise, bietet höhere Speicherdichten, niedrigere Kosten pro Bit und eine bessere Performance bei größeren Datenmengen, weist jedoch eine geringere Anzahl zulässiger Schreibzyklen (typischerweise 10.000) pro Speicherbereich auf. Die interne Flash-Memory(siehe Kapitel 3.6) des STM32F429 wurde als primärer Speicherort ausgewählt, da sie nicht mit sehr hoher Frequenz überschrieben wird, bis zu 2 MB Kapazität bietet, keine zusätzlichen externen Bauteile benötigt und mittels verfügbarer Softwarebibliotheken (HAL-Library) genutzt werden kann.

**Verwendete Bibliothek:**

**GC9A01.h**: Zur Ansteuerung des TFT-Displays wurde die Bibliothek verwendet. Diese Bibliothek implementiert grundlegende Funktionen zur Initialisierung und Steuerung des Displaycontrollers und abstrahiert die zugrunde liegende SPI-Kommunikation. [40]

**FLASH\_SECTOR\_F4.h:** Die Bibliothek speichert Daten im Flash-Speicher des Mikrocontrollers und ermöglicht Zugriff auf diese Daten. Sie dient insbesondere der persistenten Ablage von Konfigurationsparametern und Systemeinstellungen.[41]

**Image.h:** Die Bibliothek enthält die Definition von grafischen Bilddaten für das TFT-Display.

**fonts.h:** Diese Bibliothek stellt Schriftarten in Form von vordefinierten Font-Tabellen bereit und ermöglicht die textbasierte Darstellung von Informationen auf dem TFT-Display.

Der erste Schritt der Software- bzw. Firmwareentwicklung bestand in der grundlegenden Konfiguration des verwendeten Mikrocontrollers STM32F429 mithilfe des Werkzeugs STM32CubeMX[[5]](#footnote-5). Die nachfolgenden Abbildungen veranschaulichen die wesentlichen Einstellungen, die im Rahmen dieser Konfiguration vorgenommen wurden.

Zunächst wurde die Taktkonfiguration (Clock Configuration) definiert. Hierbei wurde die externe Hochgeschwindigkeitsquelle (HSE) aktiviert und auf eine Eingangsfrequenz von 24 MHz eingestellt, entsprechend der verwendeten Quarzoszillatorfrequenz auf der HMI-Platine (siehe Abbildung …).

Ein Bild, das Text, Diagramm, Zahl, Schrift enthält.

KI-generierte Inhalte können fehlerhaft sein.

In den Systemeinstellungen (System Core → RCC) wurde der HSE-Modus auf *Crystal/Ceramic Resonator* gesetzt.

Im nächsten Schritt erfolgte die **Pin- und Schnittstellenkonfiguration (Pinout & Configuration)**. Dabei wurden die für das HMI-System erforderlichen Peripheriemodule aktiviert und konfiguriert. Die **CAN1-Schnittstelle** wurde für die Kommunikation mit der Maschinesteuerung eingerichtet. Für die Ansteuerung des TFT-Displays wurde die **SPI1-Schnittstelle** aktiviert. Zusätzlich wurde die **USB\_OTG\_FS-Schnittstelle** konfiguriert, um Firmware-Updates über den USB-DFU-Bootloader zu ermöglichen.

Da Firmware-Updates ebenfalls über eine Debug-Schnittstelle vorgesehen sind, wurde im Bereich *System Core → SYS* die **JTAG-Debug-Funktion** aktiviert. Dies ermöglicht das Programmieren und Debuggen der Firmware über eine externe ST-LINK-Schnittstelle.

Zur Erfassung der Drehencoder-Signale wurde ein **Timer im Encoder-Modus** verwendet. Hierzu wurde **TIM2** entsprechend konfiguriert, da dieser laut Datenblatt des STM32F429 für Encoder-Anwendungen geeignet ist. Die Nutzung eines Hardware-Timers erlaubt eine zuverlässige und entprellte Auswertung der Drehbewegungen.

Der **Push-Button** des Displays wurde einem geeigneten GPIO-Pin (PB12) zugewiesen. Zusätzlich wurde das entsprechende externe Interrupt **(EXTI Line 15:10)** aktiviert, sodass Tasteneingaben ereignisgesteuert verarbeitet werden können. Auch die System-LEDs wurden als GPIO-Ausgänge konfiguriert.

Besondere Aufmerksamkeit wurde der **korrekten Pinzuordnung** gewidmet. Die in STM32CubeMX definierte Pinbelegung muss zwingend mit dem Schaltplan der HMI-Platine übereinstimmen, da Abweichungen zu Fehlfunktionen der Hardware oder zu nicht reproduzierbaren Laufzeitfehlern führen können.

Nach Auswahl der Konfiguration folgt die Implementierung der Funktionen. Die Tabelle ... zeigt das Klassendiagramm, das die Architektur der Software-Komponenten und deren Beziehungen veranschaulicht.

Ein Bild, das Text, Screenshot, Diagramm, parallel enthält.

KI-generierte Inhalte können fehlerhaft sein.

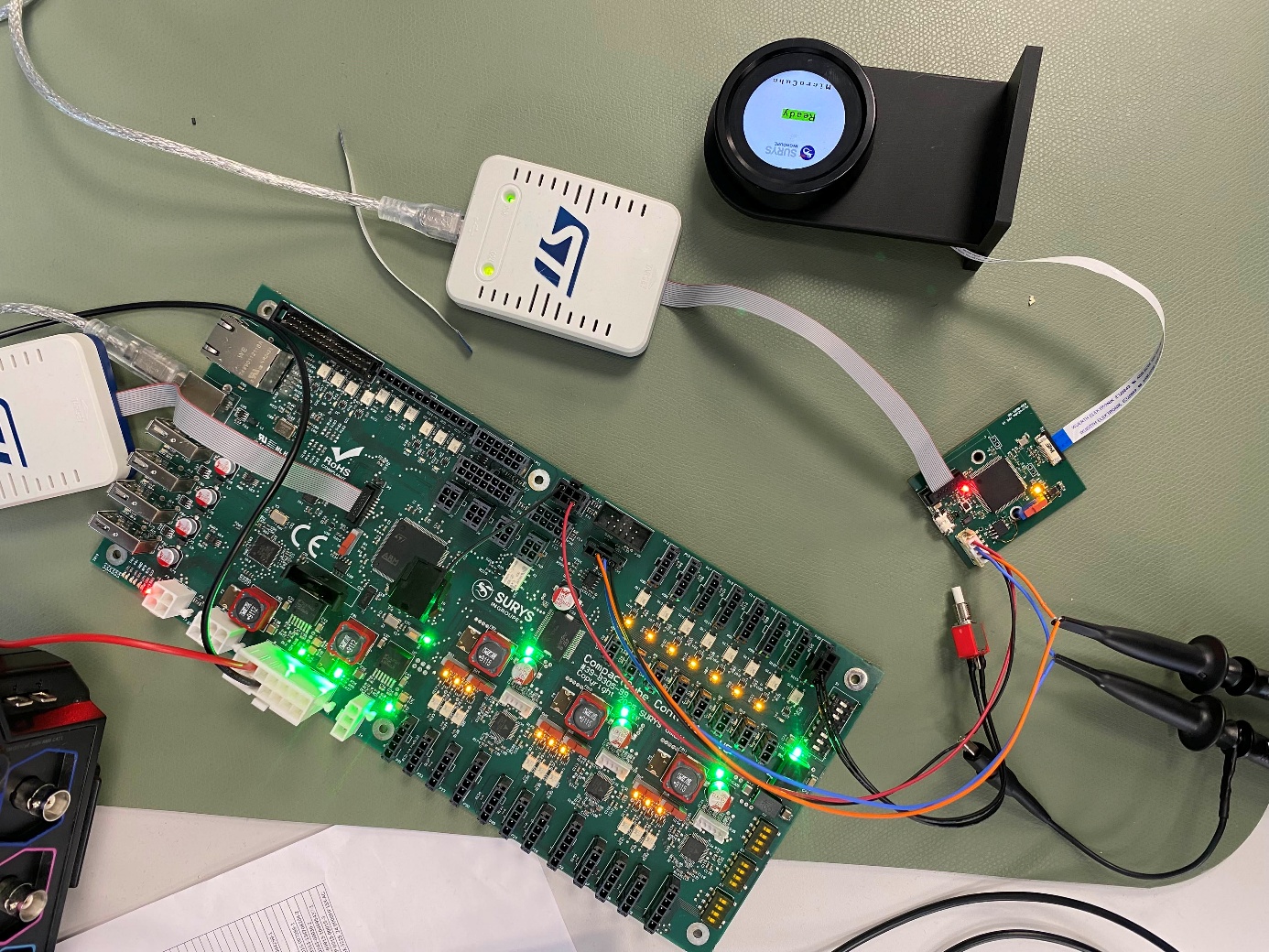
Klassendiagramm

# 5 Validierung

In diesem Kapitel wird die Funktionalität des HMI-Systems anhand praktischer Tests validiert. Ziel ist es, die Erfüllung der Anforderungen und eine zuverlässige sowie stabile Bedienung der Maschine nachzuweisen. Dabei werden insbesondere die Anzeige von Maschinenzuständen, die Benutzerinteraktion und die Kommunikation zwischen HMI-System und Maschinesteuerung überprüft.

## 5.1 Durchführung der Testläufe & Ergebnis

Die Tests wurden an einem Versuchsstand durchgeführt, der aus der HMI-Platine, dem TFT-Display und einer Testumgebung zur Simulation der Maschinesteuerung besteht.



Die Testläufe erfolgten in folgendem Ablauf:

* **Versorgung der HMI-Platine:**

Die Versorgung des HMI-Systems erfolgt über den CAN-Bus mit 5 V, die sie über einen Spannungswandler auf 3,3 V runtersetzen muss, da der Mikrocontroller und das Display damit arbeiten.

Mehrere Indikatoren zeigen die korrekte Funktion der Spannungsversorgung: Die orangefarbene Status-LED leuchtet dauerhaft, die rote System-LED blinkt periodisch.

Dieser Test zeigt, dass die Platine zuverlässig mit Spannung versorgt wird und alle wesentlichen Systemkomponenten korrekt in Betrieb sind. Damit ist eine grundlegende Voraussetzung für den stabilen Betrieb des HMI-Systems erfüllt.

* **Firmware-Update-Test( STLink-V3SET und/oder USB-DFU)**

Im Rahmen der Wartungsfähigkeit wurde die Firmwareaktualisierung überprüft. Das Firmware-Update erfolgte über die Debug-Schnittstelle mithilfe eines ST-LINK-V3SET. Hierzu wurde der Programmieradapter über den Stecker mit der HMI-Platine verbunden und die Firmware auf den Mikrocontroller übertragen.

Als Testprogramm wurde ein Beispielcode verwendet, der das Blinken der System-LED (SYS\_LED) auslöst. Nach dem Aufspielen der Firmware zeigte die System-LED das erwartete Blinkverhalten, womit die Programmierung nachgewiesen werden konnte.

Anschließend wurde derselbe Firmwarestand über den USB-DFU-Bootloader auf das HMI-System geladen. Hierzu wurde das System in den Bootloader-Modus versetzt und das Programmierwerkzeug STM32CubeProgrammer verwendet. Auch bei dieser Update-Methode konnte die Firmware erfolgreich übertragen und ausgeführt werden, was wiederum durch das korrekte Blinkverhalten der SYS\_LED bestätigt wurde.

Durch diese Tests wurde nachgewiesen, dass das HMI-System zuverlässig aktualisiert werden kann. Damit sind die Anforderungen an eine flexible und servicefreundliche Firmwarewartung vollständig erfüllt.

* **Diplay Test (SPI, Encoder & Push\_Button):**

Nachdem der stabile Betrieb des HMI-Systems verifiziert worden war, wurde die Funktion des TFT-Displays überprüft. Im Testlauf wurden Text- und Grafikobjekte dargestellt, um die korrekte Initialisierung des Displays sowie die SPI-basierte Datenübertragung zwischen Mikrocontroller und Displaytreiber zu validieren. Die erfolgreiche Darstellung der Inhalte bestätigte die Funktion der Displaybibliothek und die zuverlässige Übertragung der Anzeigedaten über die SPI-Schnittstelle.

Zur Verifikation der dynamischen Anzeige wurde die Darstellung eines Maschinenzustands getestet. Dazu wurde im Programmcode( siehe Abbildung..) ein definierter Zustand gesetzt und auf dem Display ausgegeben. Die Anzeige des Maschinenzustands "Ready" auf der Benutzeroberfläche ist in Abbildung … dargestellt. Damit konnte nachgewiesen werden, dass Zustandsinformationen korrekt verarbeitet und in der grafischen Oberfläche visualisiert werden.

Ein Bild, das Text, Screenshot, Schrift enthält.

KI-generierte Inhalte können fehlerhaft sein.

Code zum Anzeigen der Maschinenzustand Ready

Ein Bild, das Text, Kreis, Logo, Lautsprecher enthält.

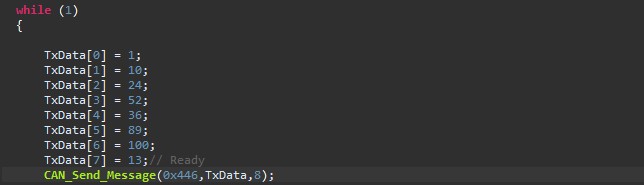
KI-generierte Inhalte können fehlerhaft sein.

Maschinezustand Ready auf dem Display

Im nächsten Schritt wurden die Menüstruktur sowie die Benutzereingabe geprüft. Dabei wurde kontrolliert, ob eine zyklische Navigation zwischen den Menüpunkten möglich ist und ob Auswahl- sowie Rücksprungfunktionen wie vorgesehen durch Tastendruck ausgelöst werden. Die Auswertung der Eingaben und die Zustandsübergänge dienten als Nachweis für die Funktion der Interaktionsschnittstelle. Der für diesen Test verwendete Programmcode ist im **Anhang** der Arbeit dokumentiert.

* **Kommunikation zwischen HMI-System und Maschinesteuerung (CAN-BUS):**

Die Überprüfung erfolgte durch den Versand definierter CAN-Nachrichten von der Maschinesteuerung an das HMI-System. Die Datenpakete wurden ausgewertet.



Das Bild… zeigt einen Ausschnitt aus dem Quellcode der Maschinensimulation, der zur Übertragung von Daten an das HMI-System verwendet wurde. Innerhalb einer Endlosschleife wird ein CAN-Datenfeld mit mehreren Werten befüllt. Anschließend wird das Datenpaket mit einer Länge von acht Bytes über eine definierte CAN-Identifier-Adresse (ID) an das HMI-System gesendet.

Die Auswertung der empfangener CAN-Daten wird in einem einen Codeausschnitt aus der HMI-Software beschrieben. Die Abbildung… zeigt es. Die Verarbeitung erfolgt in der Callback-Funktion HAL\_CAN\_RxFifo0MsgPendingCallback().

Nach dem Empfang der CAN-Nachricht wird anhand des Standard-Identifier (StdId = 0x446) überprüft, ob es sich um eine für das HMI relevante Nachricht handelt. Nur dann werden die Nutzdaten weiterverarbeitet. Die Daten werden in ein geeignetes Speicherformat kopiert und in den Flash-Speicher des Mikrocontrollers geschrieben.

Zur Verifikation der Datenspeicherung werden die geschriebenen Daten ausgelesen und in einen Vergleichspuffer kopiert.

Ein Bild, das Text, Software, Multimedia-Software, Schrift enthält.

KI-generierte Inhalte können fehlerhaft sein.

Die Abbildung zeigt die Debugger-Ansicht während der Laufzeitanalyse des HMI-Systems. Dargestellt wird der Inhalt des Arrays "rx\_back", das die aus dem Flash-Speicher zurückgelesenen Daten enthält. Die einzelnen Elemente entsprechen den zuvor empfangenen Werten. Der Werteabgleich zeigt, dass Daten korrekt ausgelesen wurden. Damit wird die Funktion der Flash-Operationen nachgewiesen. Dieser Test belegt die Persistenzfähigkeit des HMI-Systems und erfüllt eine zentrale Anforderung an Datensicherheit und Systemzuverlässigkeit.

Ein Bild, das Text, Software, Multimedia-Software, Grafiksoftware enthält.

KI-generierte Inhalte können fehlerhaft sein.

## 5.2 Diskussion

Die Testläufe zeigen, dass das HMI-System die Anforderungen erfüllt. Alle Kernfunktionen wurden erfolgreich überprüft. Die Tests zur Spannungsversorgung bestätigten eine stabile Energieversorgung. Die Umwandlung der 5 V auf die erforderliche Betriebsspannung von 3,3 V funktionierte fehlerfrei. Die Status- und System-LEDs dienten als visuelle Indikatoren für den ordnungsgemäßen Betrieb.

Die Firmware-Update-Tests zeigten, dass das HMI-System zuverlässig aktualisiert werden kann. Damit wurde die geforderte Wartungs- und Servicefähigkeit nachgewiesen. Die Ausführung identischer Firmwarestände über beide Updatepfade erhöht die Flexibilität und unterstützt die Wartbarkeit des Systems.

Im Rahmen der Display- und Benutzerschnittstellentests wurde die korrekte Initialisierung des TFT-Displays sowie die stabile Datenübertragung verifiziert. Die fehlerfreie Darstellung von Text- und Grafikobjekten sowie die dynamische Anzeige von Maschinenzuständen belegen die Erfüllung der Anforderungen an die Visualisierung. Ebenso wurde die intuitive Bedienbarkeit bestätigt, da die Menüstruktur erwartungsgemäß reagierte und alle Navigations- und Auswahlfunktionen korrekt ausgeführt wurden.

Die Tests zur CAN-Kommunikation bestätigten eine zuverlässige Datenübertragung zwischen Maschinesteuerung und HMI-System. Die Filterung und Verarbeitung der CAN-Nachrichten funktionierten fehlerfrei. Die erfolgreiche Rücklesung und Verifikation der gespeicherten Daten belegt die Persistenz Fähigkeit des Systems und erfüllt die Anforderungen an Datensicherheit und Systemzuverlässigkeit.

Das entwickelte HMI-System erfüllt alle wesentlichen Anforderungen hinsichtlich Funktionalität, Stabilität, Bedienbarkeit, Kommunikationsfähigkeit und Wartbarkeit. Die Testergebnisse bestätigen die Eignung des Systems für den vorgesehenen Einsatz im Pass-Laser-Engraver. Es bestehen Optimierungspotenziale in der Erweiterung der Menüstruktur und der Implementierung zusätzlicher Diagnosefunktionen.

|  |  |  |  |
| --- | --- | --- | --- |
| **Nr.** | **Anforderungen** | **Testfall / Validierungsmethode** | **Bewertung** |
| **F-A1** | Das HMI-System muss Maschinenzustände erfassen und in Echtzeit auf dem Display darstellen | Display-Test: Setzen definierter Maschinenzustände im Code und visuelle Darstellung (z. B. „Ready“) auf dem TFT-Display. | Erfüllt |
| **F-A2** | Zustandsübermittlung über Kommunikation oder Polling  Das System empfängt und verarbeitet Maschinenzustände zyklisch oder ereignisgesteuert. | CAN-Kommunikationstest: Empfang zyklischer CAN-Nachrichten und Auswertung im Callback. | Erfüllt |
| **F-A3** | Datenaustausch zwischen HMI-System und Maschinesteuerung  Das HMI-System muss einen bidirektionalen Datenaustausch mit der Maschinesteuerung ermöglichen. | CAN-BUS-Test: Versand definierter CAN-Datenpakete, Verarbeitung, Speicherung und Rücklesen aus Flash-Speicher. | Erfüllt |
| **F-A4** | Firmware-Update-Funktion  Das HMI-System muss Schnittstellen für Firmware-Aktualisierungen unterstützen. | Firmware-Update-Test über ST-LINK-V3SET und USB-DFU (STM32CubeProgrammer), Verifikation über LED-Blinkcode. | Erfüllt |
| **F-A5** | Benutzerinteraktion über Encoder und Push-Button  Das HMI-System erfasst Benutzereingaben über einen Drehencoder und verwendet sie zur Navigation und Funktionsauswahl. | Eingabetest: Zyklische Menü-Navigation, Auswahl und Rücksprung mittels Encoder und Push-Button. | Erfüllt |
| **F-A6** | Menübasierte Benutzerführung: Das System muss eine logisch strukturierte, hierarchische Menüführung bereitstellen, die eine intuitive Bedienung ermöglicht. | Menüstruktur-Test: Navigation zwischen Hauptmenü und Untermenüs, Zustandswechsel korrekt ausgeführt. | Erfüllt |
| **NF-A1** | Das HMI-System muss mit geringer Latenz auf Benutzerinteraktionen und Zustandsänderungen reagieren und bestimmtes Echtzeitverhalten aufweisen. | Beobachtung der Reaktionszeit bei Benutzerinteraktionen und Zustandswechseln ohne wahrnehmbare Verzögerung. | Erfüllt |
| **NF-A2** | Die Signalintegrität aller Signale muss durch Hardwaredesign und Softwaredesign sichergestellt werden. | Erfolgreicher Betrieb von SPI-, CAN- und GPIO-Schnittstellen ohne Kommunikationsfehler während aller Tests. | Erfüllt |
| **NF-A3** | Schutz gegen ESD- und EMI-Störeinflüsse  Die HMI-Platine muss gegen elektrostatische Entladungen und elektromagnetische Störungen abgesichert sein. | Indirekte Validierung durch stabilen Betrieb während Tests (keine Fehlfunktionen oder Resets). | Teilweise erfüllt |
| **NF-A4** | Mechanische Integrationsfähigkeit  Die Platine muss über vorgesehene Bohrungen sicher in das Gesamtsystem integriert werden können. | Mechanischer Test: Montage der Platine über vorgesehene Bohrungen am Versuchsstand. | Erfüllt |
| **NF-A5** | Die Software- und Hardwarearchitektur muss so gestaltet sein, dass Erweiterungen, Wartung und Fehlerdiagnose möglich sind. | Erfolgreiche Firmware-Updates über zwei unabhängige Updatepfade; modularer Softwareaufbau. | teilweise erfüllt |
| **F-A7** | Das HMI-System zeigt dem Bediener während eines Laser-Engraverjobs fortlaufend Statusinformationen, Fehler- und Warnmeldungen an. | Nicht getestet | Nicht erfüllt |

# 6 Zusammenfassung & Ausblick

# Anhang

# Abbildungsverzeichnis

# Tabellenverzeichnis

# Abkürzungsverzeichnis

| **Abkürzung** | **Bedeutung** |
| --- | --- |
| EMI | Elektromagnetische Interferenz (Electromagnetic Interference) |
| ESD | Elektrostatische Entladung (Electrostatic Discharge) |
| HAL | Hardware Abstraction Layer |
| JTAG | Joint Test Action Group |
| STLINK | ST-Link Debugger/Programmer |
| FS | Full-Speed (USB) |
| OTG | On-The-Go (USB) |
| USB | Universal Serial Bus |
| VSS | Masse/Ground (Spannungsreferenz) |
| VDDA | Analog-Versorgungsspannung |
| VREF | Referenzspannung |
| GND | Ground/Masse |
| VBAT | Batterie-Versorgungsspannung |
| VDD | Digital-Versorgungsspannung |
| RTC | Echtzeituhr (Real-Time Clock) |
| TFT | Dünnfilmtransistor-Display (Thin-Film Transistor) |
| LCD | Flüssigkristall-Display (Liquid Crystal Display) |
| CAN | Controller Area Network |
| HMI | Mensch-Maschine-Schnittstelle (Human-Machine Interface) |
| MCU | Mikrocontroller-Einheit (Microcontroller Unit) |
| STM32 | 32-Bit ARM Cortex-M Mikrocontroller (STMicroelectronics) |
| PCB | Leiterplatte (Printed Circuit Board) |
| SPI | Serielle Peripherie-Schnittstelle (Serial Peripheral Interface) |
| UART | Universelle asynchrone Sender-Empfänger (Universal Asynchronous Receiver-Transmitter) |
| GPIO | Universelle Ein-/Ausgangs-Pin (General-Purpose Input/Output) |
| LED | Lichtemittierende Diode (Light-Emitting Diode) |
| PWM | Pulsweitenmodulation (Pulse Width Modulation) |
| ADC | Analog-Digital-Wandler (Analog-to-Digital Converter) |
| DAC | Digital-Analog-Wandler (Digital-to-Analog Converter) |

# Literaturverzeichnis

[1] S. Lorenz, „Neue Dimensionen von Mensch-Maschine-Interfaces“, 2020.

[2] M. Peissner und H. Cornellia, *Potenziale der Mensch-technik interaktion für die effiziente und vernetzte Produktion von Morgen*. Stuttgart: Fraunhofer Verlag, 2013. [Online]. Verfügbar unter: https://www.iao.fraunhofer.de/content/dam/iao/images/iao-news/studie\_future\_hmi.pdf

[3] *Printed Circuits Handbook*. McGraw-Hill.

[4] SURYS, „SURYS - Hologram Industries“, https://surys.com. Zugegriffen: 8. Dezember 2025. [Online]. Verfügbar unter: https://surys.com/app-identity-passports/

[5] „Issuance Equipment“, IN Groupe. Zugegriffen: 8. Dezember 2025. [Online]. Verfügbar unter: https://ingroupe.com/product/issuance-equipment/

[6] H. Jabeen, „HMI Technologies: The Ultimate Guide to Human-Machine Interface Innovations“, Wevolver. Zugegriffen: 13. Dezember 2025. [Online]. Verfügbar unter: https://www.wevolver.com/article/human-machine-interface

[7] „STM32 Embedded Display - Capacitive Touch Panel - Optical bonding - 5-inch TFT LCD screen - RVT50HQSNWC00-B“, Riverdi. Zugegriffen: 8. Dezember 2025. [Online]. Verfügbar unter: https://riverdi.com/product/5-inch-lcd-display-capacitive-touch-panel-optical-bonding-uxtouch-stm32u5-rvt50hqsnwc00-b

[8] „Panel PC vs HMI: Difference Explained for Industrial Use“. Zugegriffen: 9. Dezember 2025. [Online]. Verfügbar unter: https://www.valanoipc.com/panel-pc-vs-hmi-difference-explained-for-industrial-use/

[9] „EAO\_WP\_HMI-Systems\_EN.pdf“. Zugegriffen: 9. Dezember 2025. [Online]. Verfügbar unter: https://www.mouser.com/pdfDocs/EAO\_WP\_HMI-Systems\_EN.pdf

[10] M. V. S. Import, „Wozu dienen HMI-Systeme und welche Varianten gibt es?“, KEM. Zugegriffen: 13. Dezember 2025. [Online]. Verfügbar unter: https://kem.industrie.de/steuerungstechnik/wozu-dienen-hmi-systeme-und-welche-varianten-gibt-es/

[11] „Was ist ein TFT-Display und wie setzt man es ein?“ Zugegriffen: 13. Dezember 2025. [Online]. Verfügbar unter: https://lcd-mikroelektronik.de/ratgeber/was-ist-ein-tft-display/

[12] C. Salomon, „Vorteile beim Einsatz intelligenter Displaymodule als HMI“. Zugegriffen: 13. Dezember 2025. [Online]. Verfügbar unter: https://www.all-electronics.de/elektronik-entwicklung/vorteile-beim-einsatz-intelligenter-displaymodule-als-hmi/773599

[13] „Maschinennahes Bedienen und Beobachten“. Zugegriffen: 13. Dezember 2025. [Online]. Verfügbar unter: https://www.siemens.com/de/de/produkte/automatisierung/simatic-hmi/panels.html

[14] „Introduction of new touch panel“, METER MIX. Zugegriffen: 8. Dezember 2025. [Online]. Verfügbar unter: https://www.meter-mix.com/news/new-touch-panel-for-all-metering-and-mixing-machines/

[15] Sunboorugged, „What is the Difference Between HMI and Industrial PC?“, Sunboorugged. Zugegriffen: 13. Dezember 2025. [Online]. Verfügbar unter: https://www.sunboorugged.com/hmi-vs-industrial-pc/

[16] „The Five Most Common HMI Hardware Variations“, Maple Systems. Zugegriffen: 13. Dezember 2025. [Online]. Verfügbar unter: https://maplesystems.com/hmi-design-guide/common-hmi-hardware-variations/

[17] J. Bowden und C. Rusnock, „Evaluation of Human Machine Interface Design Factors on Situation Awareness and Task Performance“, *Proceedings of the Human Factors and Ergonomics Society Annual Meeting*, Bd. 59, S. 1361–1365, Sep. 2015, doi: 10.1177/1541931215591226.

[18] „HMI/SCADA“, Weidmüller. Zugegriffen: 13. Dezember 2025. [Online]. Verfügbar unter: https://www.weidmueller-gti-software.com/de/software/hmi\_scada/index.jsp

[19] „Movicon WebHMI | Emerson DE“. Zugegriffen: 13. Dezember 2025. [Online]. Verfügbar unter: https://www.emerson.com/de-de/automation/control-and-safety-systems/movicon/movicon-webhmi

[20] „HMI Droid“. Zugegriffen: 8. Dezember 2025. [Online]. Verfügbar unter: https://www.idea-teknik.com/hmi\_droid.html

[21] K. H., W. W., und H. J., *Umsetzungsempfehlungen für das Zukunftsprojekt Industrie 4.0*. Springer Vieweg, 2013.

[22] „Nutzwertanalyse“, *Wikipedia*. 11. August 2025. Zugegriffen: 16. Dezember 2025. [Online]. Verfügbar unter: https://de.wikipedia.org/w/index.php?title=Nutzwertanalyse&oldid=258781943

[23] „Was ist ein TFT-Display und wie setzt man es ein?“ Zugegriffen: 1. Oktober 2025. [Online]. Verfügbar unter: https://lcd-mikroelektronik.de/ratgeber/was-ist-ein-tft-display/

[24] „Microcontroller“, *Wikipedia*. 14. November 2025. Zugegriffen: 24. November 2025. [Online]. Verfügbar unter: https://en.wikipedia.org/w/index.php?title=Microcontroller&oldid=1322168609

[25] „Research on the Application and Development Technology of Microcontrollers in Electronic Technology“, *acss*, Bd. 8, Nr. 6, 2024, doi: 10.23977/acss.2024.080611.

[26] „STM32 Microcontrollers (MCUs) - STMicroelectronics“. Zugegriffen: 24. November 2025. [Online]. Verfügbar unter: https://www.st.com/en/microcontrollers-microprocessors/stm32-32-bit-arm-cortex-mcus.html

[27] P. Papcun, E. Kajáti, und J. Koziorek, „Human Machine Interface in Concept of Industry 4.0“, in *2018 World Symposium on Digital Intelligence for Systems and Machines (DISA)*, Aug. 2018, S. 289–296. doi: 10.1109/DISA.2018.8490603.

[28] „CAN bus“, *Wikipedia*. 23. November 2025. Zugegriffen: 28. November 2025. [Online]. Verfügbar unter: https://en.wikipedia.org/w/index.php?title=CAN\_bus&oldid=1323822429

[29] „CAN Bus Grundlagen - ME-Systeme | CAN Bus Grundlagen - ME-Systeme“. Zugegriffen: 30. November 2025. [Online]. Verfügbar unter: https://www.me-systeme.de/de/grundlagen/canbus?utm\_source=chatgpt.com

[30] K. Reif, *Automobilelektronik*, 5. Aufl. 2014.

[31] H. Wallentowitz und K. Reif, *Handbuch Kraftfahrzeugelek tronik*. Vieweg & Sohn Verlag, 2006.

[32] „Unterschiede zwischen SPI EEPROMs & SPI Flash Speicher“. Zugegriffen: 17. Januar 2026. [Online]. Verfügbar unter: https://evision-webshop.de/blog/unterschiede-zwischen-spi-eeproms-spi-flash-speicher

[33] „Flash-Speicher“, *Wikipedia*. 29. Mai 2025. Zugegriffen: 17. Januar 2026. [Online]. Verfügbar unter: https://de.wikipedia.org/w/index.php?title=Flash-Speicher&oldid=256453472

[34] D. Kahng und S. M. Sze, „A Floating Gate and its Application to Memory Devices“, *Bell Syst. Tech. J.*, Bd. 46, S. 1288–1295, 1967.

[35] „Flash-Speicher / Flash-Memory“. Zugegriffen: 17. Januar 2026. [Online]. Verfügbar unter: https://www.elektronik-kompendium.de/sites/com/0312261.htm

[36] „Cycling endurance and data retention of EEPROMs in products of the ST25DV-I2C series, based on the CMOS F8H process - Application note“. Zugegriffen: 20. Januar 2026. [Online]. Verfügbar unter: https://www.st.com/resource/en/application\_note/an5085-cycling-endurance-and-data-retention-of-eeproms-in-products-of-the-st25dvi2c-series-based-on-the-cmos-f8h-process-stmicroelectronics.pdf

[37] „DS\_STM32F42x\_43x.book“. Zugegriffen: 20. Januar 2026. [Online]. Verfügbar unter: https://www.st.com/resource/en/datasheet/stm32f427vg.pdf

[38] „STM32F405/415, STM32F407/417, STM32F427/437 and STM32F429/439 advanced Arm<Sup>®</Sup>-based 32-bit MCUs - Reference manual“. Zugegriffen: 21. Januar 2026. [Online]. Verfügbar unter: https://www.st.com/resource/en/reference\_manual/rm0090-stm32f405415-stm32f407417-stm32f427437-and-stm32f429439-advanced-armbased-32bit-mcus-stmicroelectronics.pdf

[39] „Guidelines for oscillator design on STM8AF/AL/S and STM32 MCUs/MPUs - Application note“. Zugegriffen: 20. Januar 2026. [Online]. Verfügbar unter: https://www.st.com/resource/en/application\_note/cd00221665-oscillator-design-guide-for-stm8afals-stm32-mcus-and-mpus-stmicroelectronics.pdf

[40] ControllersTech, „STM32 SPI LVGL Integration with GC9A01 Round Display“, ControllersTech®. Zugegriffen: 10. Januar 2026. [Online]. Verfügbar unter: https://controllerstech.com/how-to-interface-gc9a01-round-display-with-stm32-using-spi-lvgl-integration/

[41] controllerstech, „STM32/FLASH\_PROGRAM at master · controllerstech/STM32“, GitHub. Zugegriffen: 12. Januar 2026. [Online]. Verfügbar unter: https://github.com/controllerstech/STM32/tree/master/FLASH\_PROGRAM

1. Human Maschine Interface [↑](#footnote-ref-1)
2. Red-Green-Blue [↑](#footnote-ref-2)
3. Ein Pad Hole ist ein Loch in einer Leiterplatte mit Kupferfläche zur mechanischen Fixierung und elektrischen Verbindung von bedrahteten Bauteilen.[3] [↑](#footnote-ref-3)
4. Industrie 4.0 bezeichnet die durchgängige digitale Vernetzung von Maschinen, Produkten und Menschen mittels cyber-physischer Systeme, um industrielle Produktionsprozesse autonom, flexibel und in Echtzeit zu steuern.[21] [↑](#footnote-ref-4)
5. STM32CubeMX ist ein Tool von STMicroelectronics, das die Initialisierung von STM32-Mikrocontrollern vereinfacht. Es ermöglicht die visuelle Konfiguration ohne manuelles Programmieren und beschleunigt die Projektentwicklung. [↑](#footnote-ref-5)