

Inhaltsverzeichnis

1	Einleitung	1
1.1	Motivation.....	1
1.2	Zielsetzung.....	2
1.3	Aufbau der Arbeit	3
2	Das Handbediengerät	4
2.1	Aufbau und Funktion	4
2.2	Aktuelle Lösung.....	5
3	Anforderungsanalyse	7
3.1	Vorgegebene Systemkomponenten	7
3.1.1	Sicherheitsrelevante Komponenten.....	7
3.1.2	Protokoll-Anbindung.....	8
3.2	Funktionale Anforderungen	8
3.3	Nicht-Funktionale Anforderungen	10
3.3.1	Schutzmaßnahmen	10
3.3.2	Erweiterungen.....	11
3.3.3	Implementierung	12
3.4	Resultierende technische Spezifikation	14
4	Hardware Umsetzung	15
4.1	Auswahl der Bauteile.....	15
4.1.1	Mikroprozessor	15
4.1.2	Schnittstellen.....	16
4.1.3	Display	16
4.1.4	Peripherie Bausteine.....	17
4.1.5	Spannungsversorgung	18
4.2	Platinendesign.....	20
4.2.1	Schaltplan	20
4.2.2	Layout.....	23
5	Software Umsetzung	25
5.1	Überblick über Maschinensteuerung	25
5.2	Aufteilung von Daten und Intelligenz	26
5.3	Übertragungsprotokoll	26
5.3.1	Aktueller Stand.....	27
5.3.2	Erweiterung.....	28
5.3.3	Betrachtung der Zykluszeit.....	29
5.4	Handbediengerät.....	32
5.4.1	Allgemeiner Aufbau.....	32
5.4.2	Hauptprogramm	33
5.4.3	Schnittstellen.....	35
5.4.4	Datenhandling.....	37
5.4.5	Display	38
5.5	Maschinensteuerung RMS6	39

5.5.1	Allgemeiner Aufbau.....	39
5.5.2	Handbox Modul.....	40
5.5.3	Handhabung der Schnittstellen	42
5.5.4	Datenverwaltung	43
6	Integration	46
6.1	Inbetriebnahme Hardwarekomponenten.....	46
6.2	Mechanik und Gehäuse	50
6.3	Kosten.....	51
6.4	Ergebnis.....	52
6.5	Test.....	53
6.6	Feedback	53
7	Fazit	55
7.1	Zusammenfassung.....	55
7.2	Bewertung.....	56
7.3	Ausblick auf Serienfertigung.....	56
	Literaturverzeichnis.....	58
	Abkürzungsverzeichnis.....	59
	Versicherung über die Selbstständigkeit.....	60
	Anhang	1
	A. Kommunikationsprotokoll Handbediengerät - RT-PC	1
	B. Schaltpläne RMSHB1.....	12

Abbildungsverzeichnis

Abbildung 2.1 Aktuelles Handbediengerät	5
Abbildung 4.1 Lagenaufbau.....	24
Abbildung 5.1 Datenframe von Handbediengerät an Maschinensteuerung	29
Abbildung 5.2 Datenframe von Maschinensteuerung an Handbediengerät.....	29
Abbildung 5.3 Histogramm der Zykluszeiten am Arbeitsplatzrechner	30
Abbildung 5.4 Histogramm der Zykluszeiten an einer Fräsmaschine.....	31
Abbildung 5.5 Grobe Struktur des Mikrocontroller Codes.....	33
Abbildung 5.6 Ablaufdiagramm Empfangs-Interrupt-Routine.....	36
Abbildung 5.7 Klassendiagramm der für das Handbox Modul elementaren Klassen.....	41
Abbildung 5.8 Zustandsmaschine für Schnittstelle zum Handbediengerät.....	42
Abbildung 5.9 Zustandsmaschine für Datenversand von NT an RT	44
Abbildung 5.10 Zustandsmaschine auf RT der zu visualisierenden Daten.....	45
Abbildung 6.1 RS232 Übertragung jeweils eines Telegramms.....	46
Abbildung 6.2 Signal auf Sendeleitung des Handbediengeräts.....	47
Abbildung 6.3 Signal auf Sendeleitung des Echtzeitrechners.....	47
Abbildung 6.4 Takt- und Datensignal am Ausgang des Controllers.....	48
Abbildung 6.5 Übersprechen auf Datenleitung zwischen Controller und LCD	49
Abbildung 6.6 Prototyp des Handbediengeräts	52

Tabellenverzeichnis

Tabelle 3.1 Harte Anforderungen	14
Tabelle 3.2 Weiche Anforderungen	14
Tabelle 4.1 Platinenspezifikationen	23
Tabelle 5.1 Leitungslänge	26

1 Einleitung

1.1 Motivation

Die Firma Röders Tec entwickelt seit 1991 Hochgeschwindigkeits-CNC-Fräsmaschinen. Hauptaugenmerk liegt neben hoher Dynamik auf höchstmöglicher Bearbeitungsgenauigkeit und Oberflächenqualität. Um die Nutzung der Maschinen so effizient wie möglich zu gestalten, werden viele Teilprozesse der Bearbeitung automatisiert. Dazu gehören unter anderem der Werkzeugwechsel der Spindel und der vollautomatische Wechsel von Werkstücken.

Trotz des steigenden Automationsgrades besteht weiterhin der Bedarf, eine solche Maschine manuell bedienen zu können. Dies ist erforderlich, wenn die Maschine mit einem neuen Rohling bestückt wird und seine genaue Lage für die Maschinensteuerung unbekannt ist. Zur Ermittlung der exakten Lage müssen die Maschinenachsen nahe dem Werkstück positioniert werden, von wo aus die Maschine wieder automatisch einen Antastvorgang durchführen kann.

An dieser Stelle kommt das Handbediengerät zum Einsatz, mit dem sich diese Arbeit befasst. Mit diesem, firmenintern „Handbox“ genannten, Gerät ist es möglich, die Achsen der Maschine einzeln manuell zu verfahren.

Ausgangspunkt ist das aktuelle Produkt, das vor etwa 15 Jahren in Kooperation mit einem Zulieferer entwickelt wurde. Dieses Produkt wird seitdem von diesem externen Zulieferer bezogen, wodurch die Flexibilität bei Weiterentwicklungen und Anpassungen für weitere Anwendungsfälle eingeschränkt ist. Design und Herstellungstechniken sind nicht mehr auf dem Stand der aktuellen Technik.

Vor diesem Hintergrund soll der Prototyp einer erweiterten und modernisierten Variante entwickelt werden.

1.2 Zielsetzung

Ziel dieser Arbeit ist die Entwicklung eines Handbediengeräts als funktionaler Prototyp zur manuellen Steuerung von Hochgeschwindigkeits-Fräsmaschinen.

Den Ausgangspunkt der Entwicklung stellt die Analyse des aktuell eingesetzten Handbediengeräts dar.

Funktionale Erweiterungen sollen den Bedienkomfort erhöhen und die Arbeit an den Maschinen unterstützen. Dazu werden die Möglichkeiten der Darstellung von Informationen direkt auf dem Handbediengerät untersucht.

Als Alternative zur Anbindung an die Maschinensteuerung über eine RS232 Schnittstelle werden moderne PC-Schnittstellen untersucht. Ein Erhalt der bisherigen Schnittstelle und das dort eingesetzte Übertragungsprotokoll bleiben aus Kompatibilitätsgründen erhalten.

Aus der Evaluation des Optimierungspotentials wird das Konzept des Prototypen abgestimmt.

Die Auswahl der zu verbauenden Komponenten wird vor dem Hintergrund des Obsolescence-Management und einer kosteneffektiven Fertigung getroffen.

Über die Inbetriebnahme der Platine und der Integration aller Komponenten in einem Gehäuse hinaus wird die Anbindung an die Maschinensteuerung in die Software implementiert.

Um die erweiterten Funktionen zu testen und die damit geschaffenen Möglichkeiten aufzuzeigen, werden Darstellungsszenarien für die Anwendung im Betrieb der Maschine implementiert.

Der Test des Prototypen unter Einsatzbedingungen an einer Maschine im hausinternen Formenbau soll eventuelle Schwachstellen aufzeigen. Das gewonnene Feedback der Maschinenbediener fließt in die weitere Entwicklung zu einem Serienprodukt ein.

1.3 Aufbau der Arbeit

Zunächst beschäftigt sich Kapitel 2 dieser Arbeit mit der Analyse des bestehenden Handbediengeräts. Dabei werden Konstruktion, Funktion und deren Umsetzung betrachtet. Dieser Teil bildet die Grundlage für das weitere Vorgehen.

Anschließend werden in Kapitel 3 die Anforderungen an den Prototypen in hierarchischer Abfolge analysiert und erörtert. Ergebnis ist ein Lastenheft für die folgende Entwicklung des Prototypen.

Kapitel 4 beschreibt die Umsetzung der Anforderungen in Hardware. Diese beginnt mit der Auswahl der Bauteile vor dem Hintergrund von Kosteneffektivität und Obsolescence-Management. Diese Auswahl bildet die Grundlage für die Erstellung des Schaltplans. Mit der anschließenden Erstellung des Platinenlayouts endet dieses Kapitel.

In Kapitel 5 wird zunächst ein grober Überblick über die Rechnerstruktur innerhalb der Maschinen und die Verbindung zwischen Handbediengerät und Maschinensteuerung geschaffen. Daraufhin werden die jeweiligen Systeme für sich betrachtet und die Implementierung der Kommunikation und des Datenflusses erläutert.

Kapitel 6 beschreibt die Integration der Entwicklungen. Dieser Abschnitt beschreibt die Inbetriebnahme der Platine und Anpassung des Gehäuses zur Aufnahme der einzelnen Komponenten. Den Abschluss der Integration stellen die Tests des Produktes am Arbeitsplatz und unter Einsatzbedingungen dar.

Kapitel 7 fasst die Arbeit zusammen und bewertet das Ergebnis bezogen auf die Ausgangsbedingungen. Die Arbeit endet mit einem Ausblick auf den Weg zu einem Serienprodukt.

2 Das Handbediengerät

2.1 Aufbau und Funktion

Ein Handbediengerät für Werkzeugmaschinen bietet im Allgemeinen die Möglichkeit, die Achsen der Maschine einzeln manuell zu bewegen. Dazu werden meist Taster eingesetzt, mit denen kontinuierlich mit definierbarer Geschwindigkeit verfahren wird. Zusätzlich ist ein Handrad verbaut, das rastend in beide Richtungen ohne Anschlag drehbar ist. Damit können die Achsen um einzelne Schritte mit definierter Schrittweite pro Raststellung bewegt werden. Darüber hinaus gibt es Tasten für spezielle, maschinenspezifische Aktionen.

Essentielles Element jedes Handbediengeräts ist ein Not-Aus Schalter, mit dem der Bediener jederzeit die Maschine ausschalten kann. Ein weiteres sicherheitsrelevantes Bauteil ist der so genannte Zustimmungstaster. Dieser muss durch den Bediener betätigt werden, um eine Maschinenachse zu verfahren, sofern der Maschinenraum geöffnet ist und damit eine Gefahr durch elektrische Spannung oder die Mechanik der Maschine besteht.

Die Standardanwendung ist das manuelle Verfahren der Maschinenachsen während des Einrichtens der Maschine für eine Abarbeitung. Zusätzlich wird das Gerät auch zur Steuerung von Handling-Systemen verwendet. Es handelt sich dabei um Maschinen, die Werkzeuge und Werkstücke aus einem Magazin, ohne Zutun eines Bedieners, für die Fräsmaschinen bereit stellen.

Das Handbediengerät ist über ein Kabel, das für Spannungsversorgung und Datenaustausch genutzt wird, mit dem Schaltschrank der Maschine verbunden. Die Länge des Kabels ist so ausgelegt, dass damit jede Position um die Maschine herum erreichbar ist. Am Bedienterminal der Maschine ist eine Haltevorrichtung angebracht, um das Handbediengerät aufzunehmen.

2.2 Aktuelle Lösung

Die Analyse des bestehenden Systems gibt den Ausgangspunkt für die Entwicklung eines neuen Handbediengeräts. Der Aufbau gliedert sich in Gehäuse, Bedienelemente, die elektrische Verbindung zur Maschine und die Platine. Die Betrachtung der Anbindung an die Maschine innerhalb der Software erfolgt im Rahmen der Anforderungsanalyse.

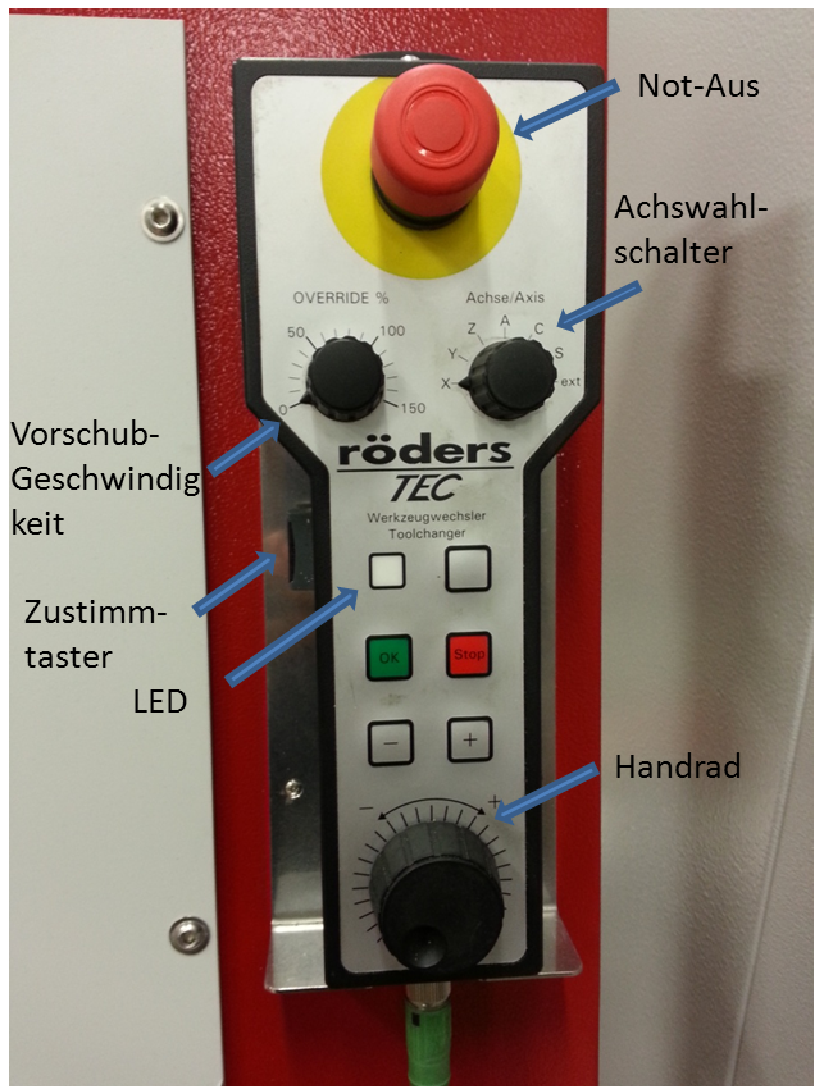


Abbildung 2.1 Aktuelles Handbediengerät

2. Das Handbediengerät

Das Gehäuse besteht aus Kunststoffprofilen, die miteinander verklebt eine Gehäuseschale ergeben. Darin sind Bohrungen für die Montage der Bedienelemente, sowie Aussparungen zum Betätigen von darunter liegenden Tasten. Die Vorderseite ist mit einer bedruckten Folie überklebt, die Beschriftung von Tasten und Skalenringe von Drehschaltern enthält.

Kernstück des Handbediengeräts ist eine Platine mit einem Mikrocontroller und einigen Peripherie-Bausteinen, mit der sämtliche Bedienelemente verbunden sind. Auffällig ist die ausschließliche Verwendung von Bauteilen für Durchsteckmontage.

Auf der Hauptplatine sind neben den elektronischen Bauteilen eine LED und fünf Taster angebracht, die durch die Aussparungen im Gehäuse betätigt werden.

Die elektrische Verbindung zur Maschine stellt ein Kabel dar, das zwei Leitungen für die Spannungsversorgung, zwei Leitungen für die serielle Schnittstelle und sechs weitere Leitungen für die sicherheitsrelevanten Not-Aus- und Zustimmung-Stromkreise beinhaltet. Diese sind jeweils redundant ausgeführt.

Der Controller ist von der Versorgungsspannung, der seriellen Schnittstelle und den Schaltspannungen der Bedienelemente, die über den Hauptstecker angeschlossen sind, galvanisch entkoppelt.

Zu den Bedienelementen gehören ein Not-Aus Schalter, ein Zustimmung-Taster und ein Drehimpulsgeber. Diese Elemente sind über einzelne Kabel auf einen gemeinsamen Stecker geführt, der mit der Platine verbunden ist. Über diesen Hauptstecker ist auch die Verbindung der Platine nach außen realisiert.

Des Weiteren gibt es zwei Drehschalter: zum einen zur Auswahl einer Maschinenachse, zum anderen zur Vorgabe der Verfahrgeschwindigkeit. Diese sind auf einer eigenen Platine montiert und werden über ein Flachbandkabel mit der Hauptplatine verbunden.

3 Anforderungsanalyse

Die grundlegenden funktionalen Anforderungen sind durch das bestehende System vorgegeben. Da sich dieses über viele Jahre bewährt hat, sind nur Erweiterungen zu den bestehenden Funktionen sowie Veränderungen ihrer Implementierung praktikabel.

Die resultierenden Anforderungen und Umsetzungsmöglichkeiten werden im Folgenden erörtert.

3.1 Vorgegebene Systemkomponenten

3.1.1 Sicherheitsrelevante Komponenten

Anforderung Not-Aus

Die bestehende Schaltungstechnik im Schaltschrank der Maschine erfordert einen Not-Aus-Taster mit drei Kanälen.

Zwei Kanäle sind komplett unabhängig voneinander und vom übrigen Handbediengerät. Sie werden direkt mit dem Not-Aus Kreis im Schaltschrank verbunden.

Ein dritter Kanal wird von dem Mikrocontroller ausgewertet, um eine Betätigung des Schalters auch per Software an die Maschinensteuerung übermitteln zu können.

Anforderung Zustimmschalter

Der Zustimmtaster muss, ebenso wie der Not-Aus-Taster, über drei Kanäle verfügen.

Aus Gründen der Sicherheit ist eine dreistufige Ausführung des Tasters erforderlich, da in einer Schrecksituation ein Schalter aus ergonomischer Sicht entweder losgelassen oder mit hoher Kraft betätigt wird. (IDEC, 2013)

Ein dreistufiger Taster trennt den Stromkreis wieder, sobald eine zweite Schaltschwelle mit deutlich höherer Betätigungskraft als die erste überschritten wird.

3.1.2 Protokoll-Anbindung

Die Datenverbindung zwischen dem Echtzeitrechner der Maschine und dem Handbediengerät ist über eine RS232 Schnittstelle realisiert. Es werden zwei Datenleitungen eingesetzt, das Bezugspotential ist durch die Spannungsversorgung gegeben. Prinzipiell wird ein Vollduplex-Betrieb unterstützt, diese Möglichkeit wird aber durch das Betriebssystem MS-DOS des Echtzeitrechners zur Zeit der ersten Entwicklung der Verbindung begrenzt. So ist der Ein- und Ausgangspuffer nur ein Byte groß und der Datenverkehr erfolgt im Halbduplex-Betrieb.

Anforderung Standardprotokoll

Das daraus resultierende Übertragungsprotokoll spezifiziert den Austausch von jeweils einem Byte pro Zyklus. Die Maschinensteuerung fordert mit der Übertragung von einem Steuerbyte die Antwort des Handbediengeräts mit einem Statusbyte an. Durch den Programmablauf der Maschinensteuerung ist eine Zykluszeit von einer Millisekunde vorgegeben.

Aus dem Frameaufbau von einem Startbit, acht Datenbits sowie zwei Stoppbits ergibt sich eine Übertragung von 22 Bits innerhalb einer Millisekunde.

Um die Baudrate so gering wie möglich zu halten und einen gängigen Wert zu verwenden, ist die Baudrate auf 38400 Bit/s festgelegt.

3.2 Funktionale Anforderungen

Anforderung Achswahl

Das aktuelle Handbediengerät unterstützt das manuelle Verfahren von sieben Achsen. Sechs davon sind Maschinenachsen. Über die siebte kann das Magazin des Werkzeugwechslers gesteuert werden.

Zur Umsetzung einer manuellen Steuerung der Verfahrbewegungen bestehen mehrere Möglichkeiten.

Variante 1:

Für jede Maschinenachse stehen zwei Tasten zum Verfahren in positiver und negativer Richtung zur Verfügung. (Nachteil: hoher Platzbedarf auf dem Bedienfeld, geringe Übersichtlichkeit)

Variante 2:

Die Auswahl der zu verfahrenen Achse erfolgt über Tasten, die Darstellung der Auswahl findet auf dem Display oder einem gesonderten Anzeigeelement statt. Zwei weitere

Tasten steuern die Verfahrbewegung. (Vorteil: Bezeichnungen der anwählbaren Achsen sind in Software konfigurierbar)

Variante 3:

Die Auswahl der zu verfahrenen Achse erfolgt über einen Drehschalter und die Steuerung der Verfahrbewegung durch zwei Tasten. (Vorteil: intuitive Bedienbarkeit)

Anforderung Override

Über das Handbediengerät kann die Vorschubgeschwindigkeit der Maschinenachsen, auch Override genannt, eingestellt werden.

In der Maschinensteuerung sind sechzehn Positionen für die Vorschubgeschwindigkeit für einen Stellbereich von 0% bis 150% mit einer Schrittweite von 10% definiert.

Häufigste Einstellung des Overrides sind 0% und 100%. Die Position 0% wird eingestellt sobald die Maschine nicht in Benutzung ist. Dies ist eine Vorsichtsmaßnahme, damit keine ungewollten Verfahrbewegungen ausgeführt werden können. Die Position 100% ist die Standardeinstellung sobald die Abarbeitung eines Fräsprogramms läuft, damit die dort eingestellten Vorschubgeschwindigkeiten, die wichtig für den Fräsprozess sind, wirksam sind.

Variante 1:

Die Einstellung des Override erfolgt über Tasten. Zwei Tasten dienen dem Inkrementieren und Dekrementieren des Wertes. Für schnelle Einstellungen werden zusätzlich zwei Tasten mit den Standardwerten 0% und 100% verwendet.

Variante 2:

Es wird ein Drehschalter mit sechzehn rastenden Positionen verwendet. Falsche Stellungen während des Umschaltens müssen ausgeschlossen werden.

Das Anwählen einer Achse und die Bestimmung der Vorschubgeschwindigkeit über Tasten hat zwar den Vorteil einer kostengünstigen und mechanisch robusten Umsetzung, die keine weiteren Bauteile benötigt und sowohl die Bearbeitung des Gehäuses als auch die Montage einfach gestaltet.

Nach Rücksprache mit einigen Maschinenbedienern und dem zuständigen Abteilungsleiter wird das vorgenannte Konzept verworfen, da bei dieser Umsetzung stets ein Kontrollblick erforderlich ist, um die aktuelle Konfiguration zu prüfen. Für die Umsetzung der Achswahl wird Variante 3 gewählt, für den Override Variante 2.

Anforderung Steuertasten

Für Verfahrbewegungen der Achsen in positiver und negativer Richtung sind zwei Tasten erforderlich.

Weitere Kommandos, die in der Maschinensteuerung implementiert sind und Bedienelemente auf dem Handbediengerät erfordern, sind das Bestätigen und Abbrechen von Aktionen. Dazu ist jeweils eine Taste vorgesehen.

Des Weiteren besteht die Möglichkeit, manuell das Werkzeug in der Frässpindel zu wechseln. Dazu muss zunächst die Werkzeugwechsel-Aktion in einem Dialog der Maschinensteuerung eingeschaltet werden. Dieser Zustand wird über eine LED auf dem Handbediengerät visualisiert. Wird die Aktion dann durch Drücken der entsprechenden Taste ausgelöst, öffnet sich die Spannzange, mit der das Werkzeug gehalten wird.

3.3 Nicht-Funktionale Anforderungen

3.3.1 Schutzmaßnahmen

Die elektromagnetische Verträglichkeit beschreibt den Einfluss von elektrischen und elektromagnetischen Effekten auf ein technisches Gerät und die von ihm verursachten Effekte. Dabei werden leitungsgebundene und feldgebundene Störungen unterschieden. Leitungsgebundene Störungen treten durch galvanische Kopplung von Stromkreisen auf.

Feldgebundene Störungen können durch kapazitive, induktive und Strahlungskopplung auftreten. Kapazitive Kopplungen treten typischerweise bei parallel geführten Leitern mit hochohmigen Abschlussimpedanzen auf, während induktive Kopplungen durch parallel geführte Leiterschleifen mit niederohmigen Abschlussimpedanzen auftreten. Strahlungskopplung entsteht durch den Einfluss elektromagnetischer Felder auf Leiter, die als Antennen wirken. (Franz, 2011)

Eine elektrostatische Entladung, kurz ESD (aus dem englischen für electro static discharge) ist der Ladungsaustausch zwischen zwei Körpern mit unterschiedlichen Spannungspotenzialen. (Wetec, 2009) Hohe Spannungen können in elektrischen Geräten zu Schäden führen. Die Potentialdifferenz entsteht dabei meist durch Reibungselektrizität oder Influenz.

Anforderung Schutzmaßnahmen

Durch den Einsatz von Überspannungsschutzdioden soll das Risiko von Beschädigungen durch elektrostatische Entladungen minimiert werden. Zum Schutz des Controllers werden alle Anschlussleitungen galvanisch getrennt.

Die elektromagnetische Verträglichkeit des Handbediengeräts ist durch entsprechende Gestaltung der Platine und auch des Gehäuses sicherzustellen.

3.3.2 Erweiterungen

Anforderung Schnittstelle

Die serielle RS232 Schnittstelle findet im industriellen Umfeld an vielen Stellen Verwendung. Die Unterstützung von modernen Schnittstellen ist erstrebenswert, um auf lange Sicht kostengünstig und ohne Probleme bei der Beschaffung von Schnittstellenkarten für den PC eine Schnittstelle betreiben zu können.

In Frage kommen hier serielle Bussysteme wie Ethernet, CAN, USB oder der in industriellem Umfeld verbreitete Profibus. (Gerhard Schnell, 2012)

Bei der Untersuchung verschiedener Bussysteme hinsichtlich ihrer Verwendbarkeit als Schnittstelle zwischen Maschine und Handbediengerät hat sich Ethernet als einzig mögliche und wirtschaftliche Alternative zur seriellen Schnittstelle herausgestellt.

Nicht alle Maschinen verfügen über einen CAN-Bus zur Ansteuerung optionaler Geräte wie Werkzeugwechsler. Da ein Handbediengerät an jeder Maschine verbaut wird, bedeutet eine zusätzliche CAN-Steckkarte hohe Kosten.

Die Etablierung eines bislang nicht verwendeten Bus-Systems, wie zum Beispiel Profibus, entfällt aufgrund der hohen Kosten von Adapterkarten für den PC. Eine native USB Schnittstelle zu implementieren erfordert zum einen viel Zeit für die Softwareentwicklung, zum anderen ist dieser Bus nicht für den Einsatz in industriellem Umfeld konzipiert. Maximale Leitungslängen von 5m laut Spezifikation machen ihn für diese Anwendung unbrauchbar.

Für die Verwendung einer Ethernetschnittstelle mit dem Betriebssystem Windows CE 6.0 sind Steckkarten erforderlich, da keine Treiber für die Schnittstellen auf dem verwendeten Board existieren.

Diese Variante wird zunächst favorisiert, da die Kosten für diese Schnittstellenkarten sehr gering sind. Bei einer späteren Betrachtung der Maschinen-PCs ist aufgefallen, dass einige Maschinentypen keinen freien Steckplatz für eine weitere Ethernet Karte haben.

Damit das Handbediengerät mit seinem erweiterten Funktionsumfang an jeder Maschine eingesetzt werden kann, konzentriert sich die Entwicklung auf die Nutzung der bestehenden RS232 Schnittstelle und die Erweiterung des verwendeten Übertragungsprotokolls.

Anforderung Anzeigeelement

Aufgrund der Größe einiger Maschinentypen ist es förderlich, dem Maschinenbediener für manuelle Aktionen wichtige Parameter an seiner aktuellen Arbeitsposition zur Verfügung zu stellen.

Ein Anzeigeelement auf dem Handbediengerät leistet die Visualisierung solcher Informationen. Der Maschinenbediener muss nicht das Bedienterminal im Blick haben und ist daher in seiner Bewegungsfreiheit weniger eingeschränkt, was die Arbeit einfacher und effektiver gestaltet.

Die Kriterien bei der Auswahl des Anzeigeelementes sind Ablesbarkeit, Vielseitigkeit, Kosten und Verfügbarkeit sowie die Ansteuerung. Die Größe des Displays wird durch den vorhandenen Platz auf der Frontseite des Handbediengeräts und den Umfang der Darstellungen bestimmt. Eine Bildschirmdiagonale von 2,8 Zoll hat sich bei Konstruktionen und Skizzen der Komponenten auf der Frontseite des Gehäuses als geeignet herausgestellt.

Konkret soll das Anzeigeelement in der Lage sein, die Positionen der typischerweise fünf vorhandenen Maschinenachsen gleichzeitig und gut ablesbar darzustellen. Auch die optische Hervorhebung einzelner Positionen soll möglich sein.

Anforderung zusätzliche Bedienelemente

Zur Steuerung von Darstellungen infolge der Erweiterung um ein Anzeigeelement können weitere Tasten neben dem Anzeigeelement angeordnet sein. Diese fungieren als Softkeys, ihre Funktion richtet sich nach der aktuellen Darstellung.

3.3.3 Implementierung

Anforderung Gehäuse

Grundanforderung an das Gehäuse des Handbediengeräts ist eine ausreichende Größe, um die oben genannten Komponenten aufzunehmen. Darüber hinaus sind Haptik und Ergonomie wichtige Aspekte. Während des Einrichtens einer Maschine sowie beim Anlernen von Handlingsystemen wird das Handbediengerät teilweise über lange Zeit benötigt. Daher ist eine Unterstützung beim Halten durch geeignetes Design der Grifffläche eine Entlastung des Bedieners.

Da das Handbediengerät trotzdem die meiste Zeit ungenutzt ist, benötigt es einen Platz, an dem es sicher verwahrt, aber trotzdem schnell und einfach zugänglich ist. Dafür ist eine Haltevorrichtung am Bedienterminal neben dem Bildschirm angebracht.

Um eine ortsunabhängige Befestigungsmöglichkeit zu schaffen, kann ein Magnet im Gehäuse untergebracht werden, der das Gerät an jeder magnetisierbaren Fläche fixiert.

Anforderung Platinenentwicklung

Die Platine soll im Hinblick auf eine kosteneffektiven Fertigung entwickelt werden. Daher ist die Verwendung von Bauteilen für Oberflächenmontage der Durchsteckmontage vorzuziehen. Die Auswahl der Bauteile soll zusätzlich vor dem Hintergrund der Zuverlässigkeit und des Obsolescence-Management getroffen werden.

Anforderung Integration

Eine Integration der Bedienelemente auf der Hauptplatine reduziert den Aufwand an Verkabelung, ist aber sehr schwierig umsetzbar, da hierdurch die Einbaulage aller Teile stark eingeschränkt wird. Außerdem muss die Montierbarkeit der Komponenten beachtet werden. Eine Verbindung der Bauteile mit der Hauptplatine über einzelne Leitungen bietet große Flexibilität bei den Einbaupositionen, erhöht aber den Fertigungsaufwand.

Nach Rücksprache mit einem Fertigungsdienstleister wird eine Lösung mit Adapterplatinen, die über vorkonfektionierte Standardkabel mit der Hauptplatine verbunden werden, umgesetzt.

Bei der aktuellen Lösung sind die Tasten zur Bedienung direkt auf der Hauptplatine angebracht, was eine zusätzliche Anbindung überflüssig macht. Dies bringt den Nachteil mit sich, dass die Einbauposition der Platine im Gehäuse unveränderbar ist und zusätzliche Aussparungen im Gehäuse benötigt werden.

Der Einsatz einer Folientastatur bietet den Vorteil einer platzsparenden Integration. Diese implementiert bereits die Dekorfolie, die Position anderer Komponenten wird hierdurch nicht beeinflusst.

3.4 Resultierende technische Spezifikation

Nachfolgend sind die Anforderungen nach der Wichtigkeit ihrer Umsetzung zusammengefasst.

*Anforderungen, die umgesetzt werden **müssen**:*

Not-Aus	dreikanalige Ausführung	
Zustimmschalter	dreikanalige Ausführung	
Standardprotokoll	Unterstützung des alten Übertragungsprotokolls für Kompatibilität zu aktuellem Handbediengerät	
Achswahl	Drehschalter mit sieben Positionen	
Override	Drehschalter mit sechzehn Positionen	
Steuertasten	„+“	zum Verfahren der Achse in positiver Richtung
	„-“	zum Verfahren der Achse in negativer Richtung
	„OK“	zum Bestätigen von Aktionen
	„Stop“	zum Abbrechen von Aktionen
	„WZW“	zum Öffnen der Werkzeugspannzange der Spindel
		Status-LED

Tabelle 3.1 Harte Anforderungen

*Anforderungen, die umgesetzt werden **können**:*

Schutzmaßnahmen	Schutz vor elektrostatischen Entladungen Sicherstellung der elektromagnetischen Verträglichkeit
Schnittstelle	Verwendung einer Ethernet-Schnittstelle als Alternative zu RS232
Anzeigeelement	LCD mit 2,8 Zoll Bildschirmdiagonale
Zusätzliche Bedienelemente	Softkeys neben dem Display
Gehäuse	Bessere Haptik und Ergonomie als aktuelles Handbediengerät Haltemagnet zur ortsunabhängigen Ablage an Maschine
Platine	Kosteneffektive Fertigung Bauteilauswahl entsprechend Obsolescence-Management
Integration	Verbindung der Komponenten und der Hauptplatine über Standardkabel und Adapterplatinen Verwendung einer Folientastatur

Tabelle 3.2 Weiche Anforderungen

4 Hardware Umsetzung

4.1 Auswahl der Bauteile

4.1.1 Mikroprozessor

Kriterien für die Auswahl eines Mikrocontrollers sind die zu unterstützenden Schnittstellen, die benötigte Rechenleistung sowie die Anzahl der benötigten Pins für Ein- und Ausgänge. Gerade im industriellen Umfeld ist eine langfristig gesicherte Versorgung durch den Hersteller erforderlich, um Folgekosten durch Abkündigung des Produktes und damit verbundenen Re-Designs zu minimieren.

Für die Umsetzung einer Ethernet Schnittstelle muss zunächst das OSI-Modell betrachtet werden. Ausschlaggebend für die Hardwareauswahl sind die beiden untersten Schichten, Physical Layer und Data Link Layer.

Die Bitübertragungsschicht, kurz PHY (engl. Physical Layer), stellt die Verbindung zwischen dem digitalen System des Mikrocontrollers und dem modulierten analogen Signal auf der Übertragungsstrecke zur Verfügung.

Die Media Access Control, kurz MAC, als Teil der Sicherungsschicht, umfasst Netzwerkprotokolle und Bauteile, die die Aufteilung eines gemeinsam genutzten Übertragungsmediums regeln. (Plate, 2013)

Zur Implementierung dieser Schichten gibt es Bausteine, die meist über ein Serial Peripheral Interface (SPI) an den Mikrocontroller angebunden werden, was die Auswahl an verwendbaren Mikrocontrollern kaum einschränkt. Jedoch ist die Anzahl benötigter Bauteile größer als bei Verwendung eines Controllers, der diese Elemente auf einem Chip integriert.

Die Firma Microchip Technology Inc. bietet mit der PIC Mikrocontrollerfamilie, die 8, 16 und 32 Bit Controller umfasst, eine große Auswahl an Lösungen für jeden Anwendungsbereich.

Der PIC18F97J60 implementiert als einziger Chip die beiden Schichten MAC und PHY der Ethernet Schnittstelle. Dieser ist in mehreren Ausführungen erhältlich, die sich in der Anzahl der Pins unterscheiden. Damit auf dem Prototypen ausreichend Ein- und Ausgänge des Controllers frei sind, um eventuelle Erweiterungen zu realisieren, wird die Variante mit 100 Pins denen mit 64 und 80 Pins vorgezogen.

Über das In Circuit Serial Programming (ICSP) Interface kann neben der Programmierung des Programmspeichers auch Debugging des Programms auf dem Chip erfolgen.

Ein weiteres Entscheidungskriterium ist das kostenfreie Angebot eines TCP/IP Stacks von Microchip, der zur Entwicklung von Applikationen mit Ethernet-Schnittstelle genutzt werden kann.

4.1.2 Schnittstellen

Bussysteme zum Datenaustausch über die Platine hinaus benötigen Bausteine zur Anpassung von Spannungspegeln und der Bitdarstellung.

Bei der Ethernet-Schnittstelle ist dieser Teil bereits im Controller integriert. Zusätzlich zum Modulieren und Demodulieren der Daten auf dem Physical Layer wird ein Übertrager benötigt, der das modulierte Signal galvanisch vom Netzwerk trennt. Für die Verwendung eines Standard-Netzkabels gibt es entsprechende Buchsen, die diesen Übertrager enthalten. Da der Anschluss des Handbediengeräts über ein Kabel, in dem Daten- und Versorgungsleitungen zusammengefasst sind, erfolgen soll, findet hier ein einzelner Übertrager Verwendung. Für eine einfache Inbetriebnahme und zu Testzwecken wird auf dem Prototypen-Board zusätzlich ein 8P8C Modularstecker verbaut.

Für die RS232-Schnittstelle wird lediglich ein entsprechender Treiberbaustein benötigt, der die Pegel auf den Leitungen denen auf der Platine anpasst.

4.1.3 Display

Die Auswahl eines Displays wird bestimmt durch dessen Größe, Bauweise und das Dateninterface. Wichtigster Faktor ist hier der Preis.

Das eingesetzte Produkt hat eine Bildschirmdiagonale von 2,8" bei einer Auflösung von 320 x 240 Pixel. Durch die Verwendung eines Displays mit integriertem Mikrocontroller wird die Ansteuerung deutlich vereinfacht, da die pixelweise Darstellung durch das Display selbst getätigt wird und lediglich Befehle in Textform zu übertragen sind. Aufgrund der Kosten wird ein solches Display jedoch nicht angewandt. Somit sind sämtliche Zeichensätze durch den verwendeten Mikrocontroller bereit zu stellen.

4.1.4 Peripherie Bausteine

Schutzbeschaltung

Zum Schutz der Bauteile, die unmittelbar über das Anschlusskabel mit dem Schaltschrank verbunden sind, werden Überspannungsschutz-Dioden verbaut. Diese schließen ab einer spezifischen Überschlagspannung die Eingänge kurz und verhindern eine Überlastung der dahinter liegenden Bauteile.

Zur Verbesserung der EMV und zum Schutz des Mikrocontrollers werden sämtliche Signale und Spannungen galvanisch entkoppelt. Hier kommen digitale Isolatoren zum Einsatz, die gegenüber Optokopplern den Vorteil einer geringen Stromaufnahme und hoher möglicher Datenraten haben. Die Stromaufnahme wird relevant, sobald das zu trennende Signal vom Mikrocontroller ausgeht, da dessen Stromtreiberfähigkeit der Ausgänge begrenzt ist. Auch wenn diese Grenzen bei Verwendung von Optokopplern eingehalten werden, stellen sie eine unnötig hohe Belastung dar.

Speicher

Zur Darstellung von Zeichen oder Bildern auf dem Display des Handbediengeräts müssen diese Daten pixelweise bereitgestellt werden. Dies ist zum einen über die Schnittstelle zum PC möglich, zum anderen durch Speicherung der Daten auf dem Handbediengerät und Abruf durch Kommandos vom PC. Da die Textdarstellung auch bei Verwendung der RS232 Schnittstelle möglich sein soll, ist eine unmittelbare Übertragung der Pixeldaten zur Darstellung aufgrund der begrenzten Bandbreite nicht sinnvoll. Die Ablage verschiedener Zeichensätze auf dem Chip-eigenen Speicher ist nicht möglich, da dieser nicht ausreichend Speicherplatz bietet. Daher wird ein externer Flash-Speicher-Baustein implementiert, der über ein SPI Interface an den Mikrocontroller gekoppelt wird.

Drehschalter

Die Auswahl der Drehschalter orientiert sich an den aktuell verwendeten und bewährten Produkten. Der aktuell verwendete Override Drehschalter verfügt über mechanische Kontakte, die einzeln durch den Controller auf ihren Schaltzustand ausgewertet werden. An dieser Stelle wird für den Prototyp eine neuere Variante verwendet, die einen Hallsensor zur Auswertung der Rotationslage verwendet und einen Mikrocontroller integriert. Über diesen wird die Schalterstellung im Grey-Code ausgegeben. Weiterer Vorteil ist der deutlich geringere Platzbedarf.

Verbindungselemente

Um die einzelnen Komponenten mit der Hauptplatine zu verbinden, sind möglichst einheitliche Steckverbinder zu verwendet werden, damit Bauteilvielfalt und damit Kosten und Montageaufwand gering gehalten werden. Da der verwendete Override-Drehschalter in seiner Standardausführung bereits einen zehnpoligen Steckverbinder vom Typ Micro-Match enthält, wird dieser Typ auch für die anderen Komponenten verwendet.

4.1.5 Spannungsversorgung

Die Versorgungsspannung des Handbediengeräts wird durch das Netzteil des RT-PCs bereitgestellt. Hier wird aktuell ein 12 V Ausgang verwendet. Denkbar wäre auch die Verwendung eines 5 V Ausganges. Optional ist auch die Versorgung über die 24 V Schiene im Schaltschrank möglich sein.

Die Spannungsversorgung wird anhand des Eingangsspannungsbereichs und des Ausgangsstroms ausgelegt. Die Qualität der Ausgangsspannung kann durch nachgeschaltete Filter auf die Bedürfnisse angepasst werden.

Aufgrund fehlender Angaben zum Stromverbrauch der Ethernet-Komponenten des Controllers und des Handrades wurden hier zur Kalkulation des Gesamt-Stromverbrauches Annahmen getroffen. Für den Controller wird der maximale Source-Strom angenommen. Bei dem Handrad wurde der maximale Aufnahmestrom vergleichbarer Bauteile herangezogen.

Angenommene Stromverbräuche:

Controller:	200	mA
Display:	70	mA
Handrad:	150	mA
Override:	20	mA
Sonstige ICs:	40	mA
	<hr/>	
Summe:	480	mA

Für den angenommenen Stromverbrauch grenzt die thermische Belastung an die Ableitungsfähigkeit des eingesetzten Gehäusetyps des Schaltreglers. Daher wird für den Prototyp die Spannungsversorgung auf zwei Kreise aufgeteilt, die jeweils über einen Schaltregler verfügen. Liegt der tatsächliche Stromverbrauch der gesamten Schaltung unter 300mA, können die beiden Kreise über eine Brücke auf der Platine verbunden und ein Schaltregler still gelegt werden.

4. Hardware Umsetzung

Die durch die Schaltregler auf 5 V herabgesetzte Eingangsspannung versorgt den Schnittstellentreiber, das Handrad sowie die Isolatoren. Zur galvanischen Entkopplung des Mikrocontrollers und seiner Peripherie-Bauteile stellen zwei DC/DC Converter 3.3 V bzw. 5 V bereit.

4.2 Platinendesign

Zur Erstellung von Schaltplan und Platinenlayout wird das EDA-Programm (Electronic Design Automation) Altium Designer der Firma Altium Limited verwendet. Dieses fasst den gesamten Designprozess elektronischer Komponenten in einer Anwendung zusammen. Dazu gehört nicht nur die Unterstützung beim Platinendesign sondern auch die Softwareentwicklung und Konstruktion des gesamten Systems mit seinen mechanischen Komponenten.

Zu den für diese Arbeit genutzten Möglichkeiten zählen das Anlegen von Bauteilbibliotheken sowie das Erstellen von Schaltplänen und Platinenlayouts.

Das Projekt wird firmenintern "RMSHB1" bezeichnet, daher findet sich dieser Begriff in den Schaltplänen und Dokumentennamen wieder.

4.2.1 Schaltplan

Der Schaltplan dient der grafischen Darstellung der Funktionen von einzelnen Bauteilen und deren elektrischer Verschaltung.

Ein Bauteil wird durch ein Schaltsymbol und die dazugehörige Anschlussfläche, Footprint genannt, definiert. Schaltsymbole sind abstrakte, funktionale Darstellungen. Footprints stellen die eigentlichen Bauteile, beziehungsweise deren Anschlussflächen, dar, auf die die Bauteile im Fertigungsprozess aufgebracht werden.

Die Erstellung eines Schaltplans beginnt mit dem Anlegen der benötigten Bauteile in einer Bauteilbibliothek. Footprints und Schaltsymbole werden getrennt voneinander angelegt und anschließend miteinander verknüpft.

Altium Designer bietet die Möglichkeit, jedem Bauteil weitere beliebige Attribute zuzuordnen. Hier werden Informationen zum Bezug des jeweiligen Bauteils sowie projektspezifische Daten wie Bestückung und RoHS Konformität hinterlegt.

Im gesamten Schaltplan benötigt jedes Bauteil einen eindeutigen Bezeichner, da in der Regel mehrere Bauteile eines Typ Verwendung finden. Typischerweise setzt sich dieser aus einem Buchstaben, der die Gattung des Bauteils spezifiziert, und einer fortlaufenden Nummer zusammen.

Mit steigender Komplexität der Anwendung steigt meist auch die Anzahl der benötigten Bauteile. Um eine übersichtliche Anordnung und Verbindung zu erreichen, wird der Schaltplan auf mehrere Seiten aufgeteilt.

Die Verbindung von Bauteilen im Schaltplan geschieht durch Netze. Wie auch bei den Bauteilen wird jedes Netz durch einen eindeutigen Bezeichner identifiziert. Ein Netz kann an beliebig vielen Stellen angeschlossen sein. Innerhalb einer Schaltplanseite sind alle Netze mit demselben Bezeichner automatisch verbunden.

Zur Definition der Netze bietet Altium Designer mehrere Möglichkeiten, die je nach Anforderung zu präferieren sind.

Zur direkten Verbindung von Bauteilpins innerhalb einer Schaltplanseite kann eine einfache Leitung verwendet werden, deren Bezeichner automatisch gesetzt wird. Zur besseren Übersichtlichkeit werden dabei die einzelnen Leitungen mit ihrem Bezeichner beschriftet.

Soll eine Verbindung von Netzen zwischen mehreren Seiten erfolgen, sind so genannte Ports zu definieren. Diese werden in einem seitenübergreifenden Dokument verbunden. Für die Versorgungsspannungen bietet sich die Definition global gültiger Power Ports an.

Für eine bessere Übersicht können mehrere Verbindungen zwischen zwei Bauteilen zu einem Harness zusammengefügt werden, der durch eine einzelne Leitung dargestellt wird.

Die Unterteilung des Schaltplans in diesem Projekt ist nach Funktionen gegliedert.

Die Seite „Power_Supply.SchDoc“ zeigt den Schaltungsteil, der die unterschiedlichen Spannungen bereitstellt, und die Versorgung der verschiedenen ICs sowie deren Entkopplungskondensatoren. „Communication.SchDoc“ beinhaltet die benötigten Peripherie-Bausteine zur Kommunikation mit dem Mikrocontroller. Wichtigste Komponenten sind hier das RS232 Interface für die Verbindung zur Maschinensteuerung und die Programmierschnittstelle. Für weitere Debugzwecke und zur Datenausgabe ist auf dem Prototyp eine USB Verbindung über einen Protokollwandler-Chip integriert. Da diese im späteren Serienprodukt nicht zum Einsatz kommt, ist die optionale Bestückung im Schaltplan kenntlich gemacht.

Auf der Seite „Periphery.SchDoc“ werden die Pins des Controllers, die keine konkrete, hardwaregebundene Funktion implementieren, verbunden. Die Seite „Interfaces.SchDoc“ umfasst die Anschlussstecker der Platine.

In „RMSHB1.SchDoc“ werden die seitenübergreifenden Verbindungen hergestellt.

Die Beschaltung der einzelnen Bauteile beschränkt sich größtenteils auf Blockkondensatoren an den Spannungsversorgungspins, auf Serienwiderständen in

Signalleitungen, um die Flankensteilheit und Überschwüngen zu mindern und auf Pullup Widerständen, um auf Signalleitungen definierte Pegel zu erzielen.

Die Pins des eingesetzten Controllers verfügen über unterschiedliche Stromtreibfähigkeiten. Pins, die den meisten Strom treiben können, werden für Verbraucher wie LEDs und eventuelle Erweiterungen verwendet. Über Pins mit mittlerer Stromtreibfähigkeit werden Signale innerhalb der Handbox übertragen. Da die Auswertung von Eingangssignalen keine Leistung benötigt, werden hier Pins benutzt, die nur wenig Strom treiben können.

Allgemein werden so wenig unterschiedliche Bauteile wie möglich verwendet, um die Serienfertigung der Platinen mit Bestückungsautomaten kostengünstig zu halten.

4.2.2 Layout

Das Board-Layout stellt die Umsetzung des Schaltplanes dar. Die Positionierung der Bauteile und die Entflechtung erfolgen manuell. Die Berücksichtigung parasitärer Effekte von Bauteilen und Leiterbahnen ist auf diese Weise effektiver umzusetzen, als über die Definition von Regeln, nach denen die Software die Entflechtung automatisch vornimmt. Die Fertigungskosten einer Platine werden maßgeblich von ihren Eigenschaften bestimmt. Größten Einfluss haben Lagenzahl und Abmessungen der Platine. Da diese durch die Anforderungen des Projektes weitestgehend vorgegeben sind, bietet sich hier kaum Einsparpotential. Eigenschaften wie minimale Abstände oder Breiten von Leiterbahnen sowie die Größe von Durchkontaktierungen können, abgesehen von Ausnahmen im Bereich von ICs mit geringem Pinabstand, frei gewählt werden. Eine Aufstellung der Spezifikationen verschiedener Hersteller wird zur Ermittlung der einzuhaltenden Eigenschaften erstellt. Damit ist der Bezug zu den jeweils geringstmöglichen Kosten zu erreichen.

Spezifikation	Hersteller			Verwendeter Wert
	ERKA	EuroCircuits	PCB Pool	
kleinster Leiterbahnabstand [mm]	0,2	0,15	0,15	0,2
kleinste Leiterbahnbreite [mm]	0,2	0,15	0,15	0,2
kleinster Restring [mm]	0,2	Bohrungsdurchmesser < 0,45 -> 0,175 Bohrungsdurchmesser > 0,5 -> 0,2	Bohrungsdurchmesser < 0,49 -> 0,15 Bohrungsdurchmesser > 0,5 -> 0,4	0,175 0,4
kleinster Bohrdurchmesser [mm]	0,3	0,25	0,3	0,3
kleinste Durchkontaktierung [mm]	0,3	0,25	0,3	0,3
min. Linienbreite Bestückungsdruck [mm]	0,2	0,2	0,2	0,2
min. Schrifthöhe Bestückungsdruck [mm]	1	1	k.A.	1
min. kupferfreie Zone von LP-Rand [mm]	k.A.	0,45	0,3	0,45

Tabelle 4.1 Platinenspezifikationen

4. Hardware Umsetzung

Die Platine ist vierlagig konzipiert, um kleinstmöglichen Bauraum bei einseitiger Bestückung zu beanspruchen. Vorteil der einseitigen Bestückung ist eine einfachere und damit günstigere Serienproduktion. Bei dem Vierlagen-Design werden die beiden inneren Lagen zur Spannungsversorgung und deren Rückführung über die Massefläche genutzt. Die äußeren Lagen dienen vorrangig den Signalleitungen. Die hier nicht genutzten Flächen werden für eine bessere EMV zusätzlich als Masseflächen verwendet.

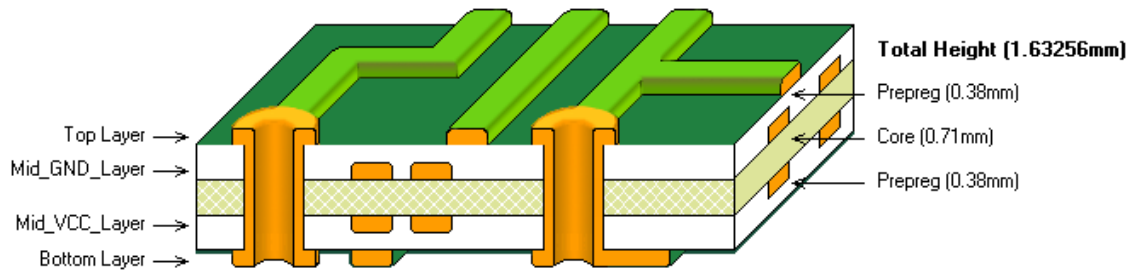


Abbildung 4.1 Lagenaufbau

Die gesamte Platine ist in zwei Bereiche aufgeteilt, um die voneinander isolierten Spannungsversorgungen auch räumlich zu trennen. Die Verbindung zwischen den beiden Bereichen wird durch DC/DC Wandler zur Energieübertragung und digitale Isolatoren zur Signalübertragung hergestellt.

Die Positionierung der einzelnen Bauteile erfolgt entsprechend ihrer Funktion, aufgeteilt auf einen der Bereiche und vor dem Hintergrund der EMV.

Die Versorgungspins jedes IC werden mit Blockkondensatoren versehen, um die bei digitalen Schaltkreisen typischen pulsartigen Stromaufnahmen abzublocken.

5 Software Umsetzung

5.1 Überblick über Maschinensteuerung

Die Maschinensteuerung ist aufgeteilt auf zwei Rechner. Diese Aufteilung ist historisch zu erklären, da nur so mit der vor über 15 Jahren zur Verfügung stehenden Rechenleistung eine hohe Maschinendynamik erreicht werden konnte. Die rechenintensive Bahnplanung sowie die Darstellung der Benutzeroberfläche werden auf dem Bediener-Rechner durchgeführt, der als „NT-PC“ bezeichnet wird, da auf ihm Windows Betriebssysteme der NT-Linie zum Einsatz kommen.

Die berechneten Verfahrbewegungen sowie weitere spezifische Aktionen werden an den „RT-PC“ (RT = Real Time) genannten Echtzeitrechner übertragen.

Die vorberechneten Bahnen werden auf dem Echtzeitrechner unter Berücksichtigung der Positionen und Geschwindigkeiten der Maschinenachsen optimiert, um eine höhere Maschinendynamik zu erreichen. Die daraus resultierenden Verfahrbewegungen werden über ein Sercos-Interface an die Antriebs-Umrichter weitergegeben.

Um den Echtzeitanforderungen gerecht zu werden, kam hier zunächst das Betriebssystem MS-DOS zum Einsatz.

Die Verbindung zwischen den Rechnern wurde zunächst durch ein Small Computer System Interface (SCSI) hergestellt. Im Zuge einer Modernisierung wurde MS-DOS durch Windows CE ersetzt und das SCSI durch eine Ethernet-Verbindung abgelöst.

Der RT-PC ist über Steckkarten direkt mit der Hardware der Maschine verbunden und gibt die vom NT-PC empfangenen Steuerbefehle an diese weiter. Andersherum werden Signale der Maschine ausgewertet und an die Bedienoberfläche auf dem NT-PC gesendet.

Da der RT-PC während des normalen Betriebs der Maschine nicht zugänglich ist, können Einstellungen hier nicht direkt vorgenommen werden. Dateien, die Konfigurationsdaten enthalten, werden daher beim Starten der Maschinensteuerung an den Echtzeitrechner übertragen.

Das Handbediengerät ist an den RT-PC angeschlossen, da die Weitergabe einiger Bedienbefehle an die Maschinensteuerung essentiell und zeitkritisch sind. Die Übertragung der Bedienbefehle an den NT-PC ist zur Steuerung der Benutzeroberfläche sowie zur Rückmeldung und Quittierung erforderlich.

5.2 Aufteilung von Daten und Intelligenz

Die Nutzung des Displays des Handbediengeräts erfordert eine flexible Bereitstellung von Daten zur Visualisierung. Das Handbediengerät stellt unterstützende Funktionen zur Darstellung von Zeichensätzen, Bildern und Graphen bereit. Die Nutzung dieser Funktionen in Abhängigkeit des Maschinenzustandes und des Anwendungsfalls geschieht durch die Maschinensteuerung.

Ziel dieser Aufteilung ist das Handbediengerät als abgeschlossenes System, dessen vielseitige Nutzbarkeit durch Entwicklungen in der Maschinensteuerung erreicht wird.

Die zur Darstellung verwendeten Schriftsätze verschiedener Größen und Bilder werden in Dateien auf dem NT-PC abgelegt. Dort ist eine einfache Veränderbarkeit der Daten durch den Benutzer gewährleistet.

Damit diese zur Darstellung auf dem Handbediengerät zur Verfügung stehen, werden Mechanismen zur Übertragung und Synchronisierung implementiert.

5.3 Übertragungsprotokoll

Das im Folgenden erarbeitete Übertragungsprotokoll wird von der RS232-Schnittstelle und der Ethernet-Schnittstelle gleichermaßen implementiert. Das Protokoll ist vorrangig auf die RS232-Schnittstelle abgestimmt, da diese nicht durch weitere Schichten Redundanz zur Fehlererkennung implementiert.

Um die geforderten Erweiterungen zu unterstützen ist ein Übertragungsprotokoll erforderlich, das den vielseitigen Aufbau von Datentelegrammen ermöglicht.

Um die Kompatibilität mit alten Versionen der Maschinensteuerung zu erhalten, ist auch das bisherige Protokoll zu implementieren.

Der Rahmen, mit dem die Übertragung über die serielle Schnittstelle arbeitet, definiert acht Datenbits, ein Startbit und zwei Stopbits. Das Paritätsbit zum Erkennen einzelner Bitfehler wird nicht verwendet.

Die Baudrate von 38400 Bit/s wird beibehalten, damit unabhängig von der Verwendung des Übertragungsprotokolls eine Verbindung zustande kommen kann. Von Texas Instruments ermittelte Erfahrungswerte von Leitungslänge zu maximaler Baudrate zeigen, dass eine Erhöhung der Baudrate aufgrund der Leitungslänge von bis zu 12m nicht empfehlenswert ist. (Texas Instruments, 2002)

Maximale Baudrate	Maximale Leitungslänge
2400	900 m
4800	300 m
9600	152 m
19200	15 m
57600	5 m
115200	<2 m

Tabelle 5.1 Leitungslänge

5.3.1 Aktueller Stand

Das bisherige Übertragungsprotokoll bietet keine Möglichkeit Fehler zu detektieren, solange der Rahmen in seinem Aufbau nicht verletzt wird.

Damit das Auftreten einzelner Bitfehler bei der Übertragung nicht zu einer direkten Reaktion der Maschine führt, werden sensible Informationen gefiltert. Dadurch wird zusätzlich sichergestellt, dass Prellen eines Tasters nicht zu einer mehrfachen Interpretation eines Tastendrucks führt.

Die Filterung wird dadurch erreicht, dass der Zustand eines Betätigungselementes 50ms unverändert sein muss, bevor er von der Steuerung übernommen wird.

Das bestehende Protokoll umfasst den Austausch von jeweils einem Byte zwischen Maschinensteuerung und Handbediengerät während eines Zyklus, mit einer Zykluszeit von einer Millisekunde.

Die Daten des Handbediengeräts werden dazu auf drei Bytes aufgeteilt. Über ein Steuerbyte fordert die Maschinensteuerung die einzelnen Datenbytes der Reihe nach an. Zusätzlich enthält dieses Steuerbyte ein Statusbit über das die optische Anzeige für den Werkzeugwechselstatus gesteuert wird.

Das erste Byte umfasst vier Bits zur Darstellung der Position des Override-Drehschalters und drei Bits für die Position des Achswahlschalters. Zusätzlich wird das bisher ungenutzte Bit zur Identifikation des neuen Handbediengeräts verwendet.

Das zweite Byte enthält die Zustände der Tasten Start, Stop, Plus, Minus, Werkzeugwechsel, Zustimmungstaster und Not-Aus, wobei zwei Zustimmungstaster unterstützt werden.

Das dritte Byte wird als vorzeichenbehaftete Zahl interpretiert und beinhaltet die Inkremente des Handrades seit dem letzten Lesevorgang dieser Information. Ein Verlust

der Inkremente durch eine fehlerhafte Übertragung ist nicht problematisch, da der Bediener für genaue Arbeiten die Achsposition auf dem Terminal oder dem Bildschirm des Handbediengeräts kontrolliert. Bei groben Verfahrbewegungen wird die Zahl der Raststellungen des Handrades generell nicht mit der tatsächlichen Positionsänderung verglichen.

5.3.2 Erweiterung

Das neue Übertragungsprotokoll muss die Übertragung von Daten zur Visualisierung auf dem Display definieren. Zusätzlich sollen Funktionen implementiert sein, die Fehler bei der Übertragung erkennen und das Interpretieren fehlerhafter Daten verhindern.

Das neue Übertragungsprotokoll definiert für beide Übertragungsrichtungen einen Rahmen fester Länge, damit weiterhin eine konstante Zykluszeit erreicht werden kann. Beide Rahmen beginnen mit einem definierten Startzeichen und enden mit einer Checksumme, die die Richtigkeit der Daten verifiziert. Der weitere Aufbau der Rahmen richtet sich nach den jeweiligen Anforderungen.

Zur Bildung einer Checksumme gibt es mehrere Möglichkeiten. Die einfachste stellt eine Bytesumme dar. Da gerade bei langen Frames über viele Wege die gleiche Summe entstehen kann, ist sie keine verlässliche Quelle zur Fehlererkennung.

Eine andere Möglichkeit ist die zyklische Redundanzprüfung (engl. Cyclic Redundancy Check, kurz CRC). Dazu wird ein Generatorpolynom definiert. Die zu übertragenden Daten werden durch die binäre Darstellung dieses Polynoms dividiert und der Rest der Division als Checksumme an die Daten angehängt.

Da bei dieser Division kein Quotient gebildet werden muss, erfolgt die Berechnung des Rests durch bitweise XOR Verknüpfung der Daten mit dem Generatorpolynom. (mikrocontroller.net)

Daraus resultiert eine Operation des Controllers für jedes Datenbit.

Um die Berechnung der Checksumme effizienter zu gestalten, wird hier eine Tabelle verwendet, mit der die Prüfsummen byteweise ermittelt werden können. (Racz)

Die Verwendung einer Checksumme macht eine nachträgliche Filterung der Daten, zum Ausschluss von Fehlübertragungen überflüssig. Eine Filterung zum Entprellen der Taster ist weiterhin erforderlich. Diese wird durch die Software des Handbediengeräts sichergestellt.

Die Informationen des Handbediengeräts werden in Rahmen, die sieben Bytes umfassen, übertragen. Die ersten drei Datenbytes enthalten die Daten wie sie auch im bisherigen Protokoll definiert sind. Das vierte Byte dient der Übertragung der Zustände zusätzlicher Taster. Mit dem Acknowledge-Byte wird der zuvor von der Maschinensteuerung empfangene Rahmen quittiert. Dieses Byte dient dazu, Inkonsistenzen in der Datenabfolge oder einen vollen Empfangsbuffer zu signalisieren und stellt damit eine Flusskontrolle dar.

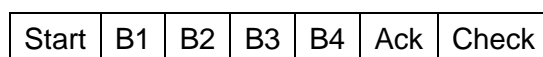


Abbildung 5.1 Datenframe von Handbediengerät an Maschinensteuerung

Kommandos der Maschinensteuerung an das Handbediengerät werden in Rahmen mit einer Länge von dreizehn Bytes übertragen. Ein solches Kommando enthält drei Bytes, die die Verwendung bestimmen und die Datenabfolge steuern, sowie acht Datenbytes.

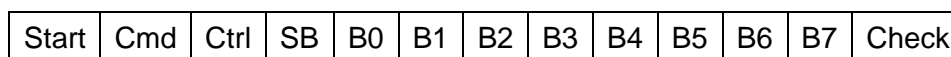


Abbildung 5.2 Datenframe von Maschinensteuerung an Handbediengerät

Der Aufbau der verschiedenen Kommandos und die jeweilige Verwendung der Steuerbytes ist in Anhang A in einem Dokument zusammengefasst.

5.3.3 Betrachtung der Zykluszeit

Bei einer Baudrate von 38400 Bit/s beträgt die Zeit zur Übertragung eines Bits

$$\frac{1s}{38400\text{Bit}} = 26,0416 \frac{\mu s}{\text{Bit}}$$

Aus der Baudrate und der Länge der Rahmen ergeben sich folgende Übertragungszeiten:

$$t_1 = 10 \frac{\text{Bit}}{\text{Byte}} * 7 \text{ Byte} * 26,0416 \frac{\mu s}{\text{Bit}} = 1,8229\text{ms}$$

$$t_2 = 10 \frac{\text{Bit}}{\text{Byte}} * 13 \text{ Byte} * 26,0416 \frac{\mu s}{\text{Bit}} = 3,3854\text{ms}$$

Daraus ergibt sich in Summe die Übertragungszeit:

$$t = t_1 + t_2 = 5,2083\text{ms}$$

Zusätzlich benötigt das Handbediengerät Zeit zur Verarbeitung der eingehenden Kommandos. Diese ist nicht deterministisch, da sie vom Interpretationsaufwand abhängt. Da diese Verarbeitungszeiten im Bereich einiger μs liegen, fallen sie im Vergleich zu den Übertragungszeiten nicht ins Gewicht.

Bei der Schnittstellensteuerung seitens der Maschinensteuerung wird auf eine vorhandene Implementierung zurückgegriffen. Ein Thread kontrolliert und steuert die vom Betriebssystem verwaltete Schnittstelle. Im Millisekundenraster, vorgegeben durch den Aufbau der Maschinensteuerung, kann der Status des Threads abgefragt und gesteuert werden.

Zur Analyse der Zykluszeit werden über eine Protokollierungsfunktion der Maschinensteuerung mehrere Messreihen aufgenommen. Die relativen Häufigkeiten der einzelnen gemessenen Zeiten zeigen, dass diese Implementierung ein nicht-deterministisches Zeitverhalten hat. Darüber hinaus dient sie als Kriterium zur Festlegung eines Grenzwerts der Zykluszeit. Mit diesem Grenzwert wird die ordnungsgemäße Funktion der Schnittstelle überwacht.

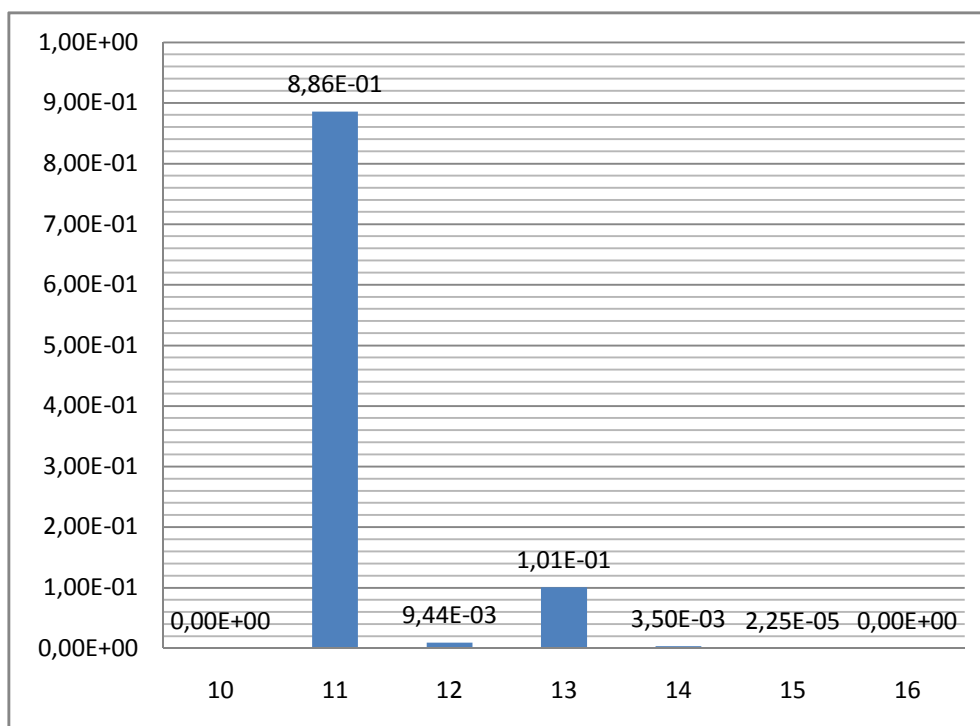


Abbildung 5.3 Histogramm der Zykluszeiten am Arbeitsplatzrechner

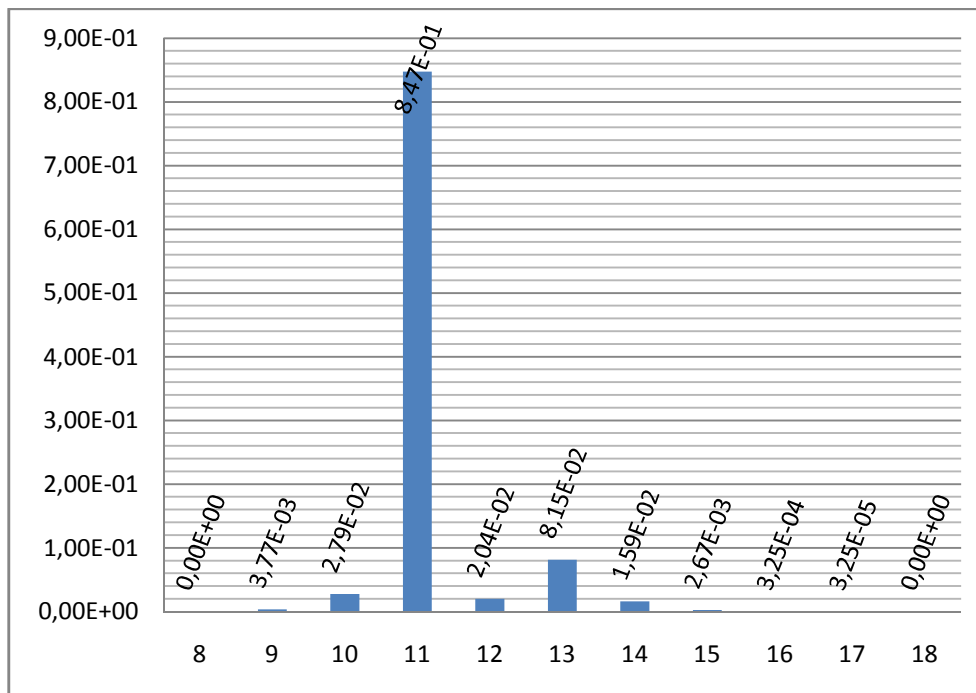


Abbildung 5.4 Histogramm der Zykluszeiten an einer Fräsmaschine

Der Vergleich des Rechnersystems am Arbeitsplatz mit dem einer Fräsmaschine zeigt die gleichen signifikanten Zykluszeiten von 11 ms und 13 ms. Unterschied ist, dass die Zeiten, die an der Maschine gemessen werden, einen breiteren Bereich abdecken als die am Arbeitsplatz. Der Unterschied in der Verteilung der Zykluszeiten ist durch unterschiedliche Rechner-Hardware und andere Belastungen des Rechners in der Maschine zu begründen.

Die Zykluszeit-Obergrenze ist zunächst auf 30 ms eingestellt, kann aber aufgrund dieser Messergebnisse auf 20 ms reduziert werden.

Da beim neuen Übertragungsprotokoll die Zykluszeit deutlich größer ist als beim alten, wird ein möglicher Variablenüberlauf der Inkremente des Handrades betrachtet.

Maximal können 127 Inkremente durch das übertragene Byte dargestellt werden. Der eingesetzte Inkrementalgeber verfügt über 100 Inkremente / Umdrehung. Bei einer Zykluszeit von maximal 17ms ergibt sich die maximal detektierbare Drehzahl von 4500 U/min und liegt damit deutlich über der mechanischen Höchstgrenze des Drehimpulsgebers von 500 U/min.

5.4 Handbediengerät

Auf dem Handbediengerät kommt ein 8-Bit Mikrocontroller der Firma Microchip Inc. zum Einsatz. Der Quellcode ist in der Hochsprache C geschrieben.

Zur Erstellung der Software wird die Entwicklungsumgebung MPLab X der Firma Microchip Technology Inc. verwendet. MPLab X basiert auf der Open Source Entwicklungsumgebung NetBeans von Oracle und steht kostenlos zur Verfügung.

Zusätzlich wird ein Compiler benötigt um den erstellten Quellcode für den Mikrocontroller zu kompilieren. Der verwendete MPLab C18 Compiler der Version 3.46, ebenfalls von der Firma Microchip Technology Inc., ist in seinem Standard-Funktionsumfang kostenfrei erhältlich.

Obwohl die Programmiersprache C nicht zu den objektorientierten Sprachen zählt, ist der Code der Idee der Datenkapselung folgend strukturiert.

5.4.1 Allgemeiner Aufbau

Mikrocontroller bieten zwei Möglichkeiten auf Eingangssignale zu reagieren. Beim Polling wird mittels Schleifen auf Ereignisse gewartet und erst bei deren Auftreten das Programm fortgeführt. Während des Wartens ist der Prozessor vollständig ausgelastet und kann keine weiteren Befehle abarbeiten. Im Gegensatz dazu wird bei interruptbasierter Programmausführung beim Auftreten eines bestimmten Ereignisses der Programmablauf an der aktuellen Stelle unterbrochen und eine definierte Stelle im Programmspeicher angesprungen, an der die zugeordnete Funktion ausgeführt wird. Nach deren Beendigung wird der Programmablauf an der vorherigen Stelle fortgeführt.

Damit der Programmablauf bei gleichzeitigem Auftreten mehrerer Interrupts oder während der Abarbeitung eines zuvor aufgetretenen Interrupts klar definiert bleibt, ist eine Priorisierung und gegenseitige Verriegelung erforderlich. Diese sind durch die Architektur des Controllers gegeben.

Das Programm ist in die drei Bereiche Initialisierung, Hauptschleife und Interrupt-Routinen gegliedert. Zusätzlich läuft im Hintergrund ein Watchdog-Timer. Dieser dient der Überwachung der korrekten Funktion des Programms. Dazu muss in regelmäßigen Abständen der Zähler des Watchdog durch das Programm zurückgesetzt werden. Geschieht dies nicht und der Zähler überschreitet einen definierten Grenzwert, wird der Controller neu gestartet.

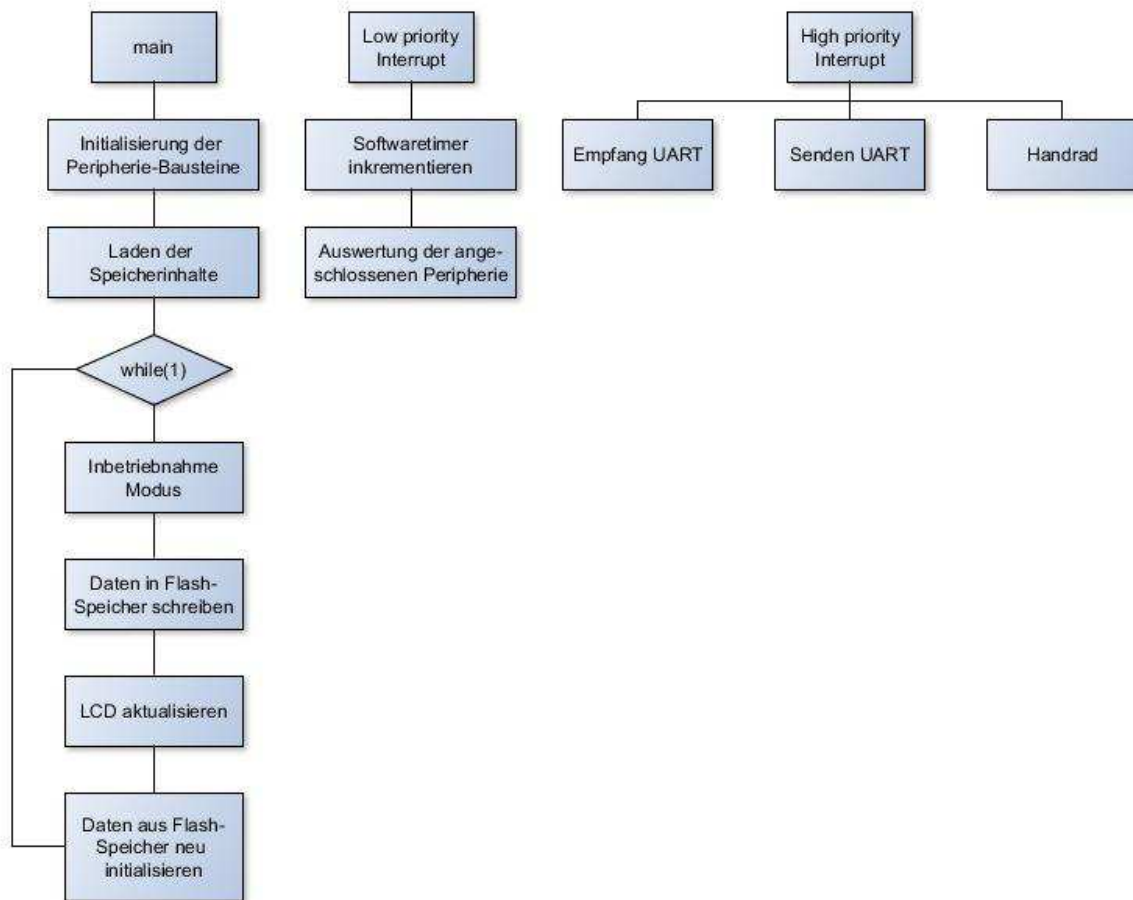


Abbildung 5.5 Grobe Struktur des Mikrocontroller Codes

5.4.2 Hauptprogramm

Initialisierung

Die Initialisierung umfasst das Setzen entsprechender Register des Mikrocontrollers um die Peripherie des Prozessors gemäß den Anforderungen nutzen zu können. Darunter fallen Timer, der UART Schnittstellenbaustein, die Ethernet Schnittstelle sowie die Ein- und Ausgänge für die Peripherie. Die Initialisierung wird nur einmal bei Anlegen der Spannungsversorgung an den Controller ausgeführt.

Abschluss der Initialisierungsphase ist die Prüfung der Konfiguration und der Datenintegrität der Inhalte des externen Speicherbausteins. Diese Prüfung wird bei Veränderungen der Daten auch in der Hauptschleife des Programms durchgeführt.

Hauptscheife

In der Hauptarbeitsschleife werden Funktionen mit niedriger Priorität und hohem Zeitbedarf ausgeführt. Dies betrifft die Ausgabe von Zeichen auf dem Display sowie das Löschen und Beschreiben von Speicherbereichen.

Im Hinblick auf eine spätere Serienfertigung ist ein Inbetriebnahme-Modus implementiert. Dieser bietet eine einfache Möglichkeit, die vollständige und korrekte Funktion des Handbediengeräts zu kontrollieren. Dieser Modus kann durch Betätigen von Not-Aus- und Zustimmungstaster und einer bestimmten Positionierung der Drehschalter bei Anlegen der Versorgungsspannung erreicht werden.

In diesem Modus werden die Zustände aller Tasten und die Positionen der Drehschalter auf dem Display angezeigt. Zur Darstellung von Texten ist dafür ein kleiner Zeichensatz im Code des Handbediengeräts hinterlegt.

Auswertung der Peripherie

Das Einlesen der Signale von Tasten und Schaltern wird durch einen Timer gesteuert, der mit konstanter Frequenz einen Interrupt auslöst. Eine konstante Abtastfrequenz ist für die Filterung der Signale zum Entprellen der Taster erforderlich. Der binäre Zustand jedes einzelnen Tasters wird bei jedem Abtastvorgang aufsummiert. Überschreitet die Summe einen konfigurierbaren Grenzwert, wird der gedrückte Zustand des Tasters angenommen. Das Loslassen des Tasters wird unmittelbar übernommen.

Das Handrad gibt zwei Rechtecksignale mit einem Phasenversatz von 90° aus. Zur Auswertung löst jeweils eine steigende Flanke einen Interrupt aus. Anhand des Pegels des zweiten Signals wird die Drehrichtung ermittelt.

Eine Auswertung über die kontinuierliche Abfrage der entsprechenden Eingänge ist ebenfalls möglich, benötigt aber eine hohe Abtastfrequenz.

Die benötigte Abtastfrequenz ergibt sich aus:

- der maximalen Drehzahl des Handrades: 500 U/min (83 U/sec)
- der Anzahl von Raststellungen pro Umdrehung: 100
- und den Signalfanken pro Raststellung: 4

Damit jede Flanke detektiert wird, muss die Abtastfrequenz mindestens doppelt so hoch sein wie die Signalfrequenz. Daraus ergibt sich eine Abtastfrequenz von ca. 66kHz.

Die für die Berechnung gewählte Drehzahl entspricht dem mechanischen Maximum des Handrades. Die im normalen Betrieb von Hand erreichbare Drehzahl liegt bei nicht einmal 50 U/min. Daher kann auch die Abtastfrequenz auf 6kHz reduziert werden.

Eine kontinuierliche Abfrage lastet den Controller permanent stärker aus, wodurch weniger Rechenkapazität für andere Operationen zur Verfügung steht. Daher wird hier die interruptbasierte Auswertung verwendet.

5.4.3 Schnittstellen

Die RS232-Schnittstelle stellt die Standardverbindung zur Maschinensteuerung dar. Über den im Mikrocontroller integrierten UART Baustein werden aus dem eingehenden seriellen Datenstrom die Daten extrahiert und im Empfangsregister bereitgestellt. Äquivalent dazu werden Daten aus dem Ausgangsregister in einen seriellen Datenstrom umgesetzt.

Ein Wechsel des Übertragungsprotokolls während des Betriebs ist möglich, da bei Empfang jedes Bytes geprüft wird, welchem Protokoll es entspricht und passend darauf reagiert wird. Die Unterscheidung erfolgt anhand des begrenzten Wertebereiches, den das Steuerbyte des alten Übertragungsprotokolls annehmen kann. Da der Wert des Startzeichens des neuen Protokolls außerhalb dieses Wertebereichs liegt, kann der Beginn eines Datenpakets detektiert werden. Durch den festen Aufbau dieser Datenpakete ist auch deren letztes Byte einfach zu ermitteln.

Da innerhalb eines Datenpaketes jeder Wert zulässig ist, kann eine Unterscheidung des verwendeten Protokolls nur zwischen zwei Datenpaketen erfolgen.

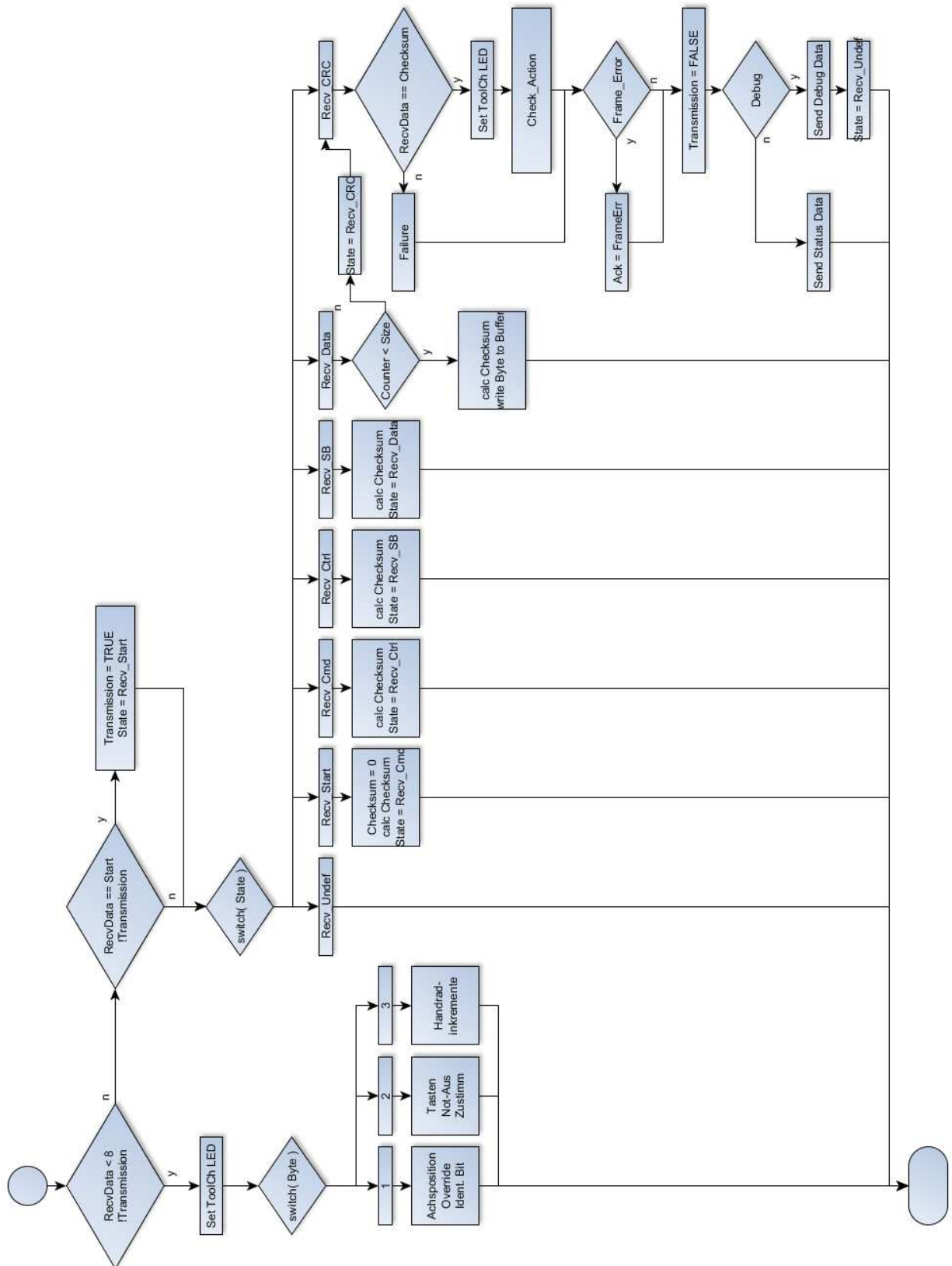


Abbildung 5.6 Ablaufdiagramm Empfangs-Interrupt-Routine

Zur Entwicklung der Ethernet-Schnittstelle zwischen Handbediengerät und Maschinensteuerung wird auf der Seite des Handbediengeräts der von der Firma Microchip Technology Inc. bereitgestellte TCP/IP-Stack verwendet. Dieser umfasst neben den transportorientierten Schichten des OSI-Modell einige Beispiele der Anwendungsschichten. Der Begriff Stack (dt. Stapel) ist durch die Stapelung der einzelnen Schichten aufeinander begründet. Dabei greift jede Schicht auf Funktionen unter ihr liegender Schichten zu. (Microchip Technology Inc., 2008)

Der Stack basiert auf kooperativem Multitasking. Durch den Programmablauf muss sichergestellt sein, dass Funktionen, die ein- und ausgehende Informationen verarbeiten oder Zeiten überwachen, regelmäßig aufgerufen werden.

Für die Verwendung im Handbediengerät wird das Beispiel eines TCP-Servers verwendet. Die Maschinensteuerung stellt einen Client dar und fordert durch die Übertragung eines Datenpakets die Antwort des Handbediengeräts mit ihren Statusdaten an.

Der Test der Kommunikation über die Ethernet-Schnittstelle erfolgt mit einem bestehenden, firmenintern entwickelten Tool, das zyklisch Datenpakete versendet und empfangene Daten anzeigt.

Eine tiefere Integration des TCP-Servers in den Code des Handbediengeräts wird vorerst nicht weiter verfolgt, da die Ethernet-Schnittstelle nicht auf allen Maschinen einsetzbar ist.

5.4.4 Datenhandling

Das Handbediengerät unterstützt die Darstellung von Bildern, den Achspositionen der Maschine, von Bargraphen und freier Texte. Die dafür benötigten Informationen werden von der Maschinensteuerung übertragen.

Die zeitintensive Darstellung auf dem Display macht eine Speicherung eingehender Daten in Buffern erforderlich. Zusätzlich wird dadurch verhindert, dass ein Schreibvorgang auf das Display durch den Empfang von Daten unterbrochen wird und es durch das Überschreiben der Daten zu Inkonsistenz kommt.

Für jeden Darstellungstyp wird ein eigener Buffer zur Speicherung verwendet. Die Buffer sind Arrays von spezifischen Strukturen. Der Zugriff wird durch zwei Variablen gesteuert, die den Index der Lese- bzw. Schreibposition darstellen. Durch den erneuten Beginn am Anfang des Arrays bei Erreichen des Endes wird ein Ringspeicher realisiert. Dieses Prinzip ist der UART-library von Peter Fleury entnommen. (Fleury, 2013)

5.4.5 Display

Das Display integriert einen HX8347D Controller und wird über ein Serial Peripheral Interface angesteuert. Es hat eine Auflösung von 320x240 Pixel bei einer Farbtiefe von maximal 18Bit. Um Speicherplatz zu sparen und die Ansteuerung effizienter zu gestalten, werden nur 16Bit Farbtiefe verwendet. Der Controller verfügt über einen internen Grafikspeicher, in dem der Bildschirminhalt abgelegt wird. Zum Neuzeichnen des Inhalts wird ein rechteckiger Bereich durch die Angabe von zwei Eckpunkten definiert. Gemäß der Größe des definierten Bereichs werden die Pixeldaten direkt aufeinanderfolgend übertragen. Auf diese Weise wird der Adressierungsaufwand stark reduziert.

Aus der Auflösung und der Farbtiefe des Displays ergibt sich eine Datengröße von 153600 Bytes. Das SPI wird mit einer Baudrate von 2,6 MBit/s betrieben. Daraus ergibt sich theoretisch eine Dauer von 0,47 Sekunden um den gesamten Bildschirminhalt zu übertragen. Hinzu kommt die Zeit, die der Mikrocontroller zur Bereitstellung der richtigen Daten und zur Verwaltung des Interfaces benötigt. Eine Messung ergibt eine Dauer von 1,15 Sekunden für ein komplettes Neuzeichnen des Display-Inhalts.

Die Darstellung von Schrift und Zahlen erfordert das Neuzeichnen nur eines Teils des Bildschirms für jedes Zeichen. Damit bei kontinuierlicher Änderung der darzustellenden Zeichen die benötigte Zeit zur Aktualisierung nicht so stark auffällt, wird stetig geprüft, welche Zeichen ein Neuzeichnen benötigen und welche sich nicht verändert haben. Dieses Verfahren findet vorrangig bei der Darstellung der Achspositionen Anwendung.

5.5 Maschinensteuerung RMS6

Die Software zur Steuerung der Maschinen wird firmenintern entwickelt. Die Bezeichnung RMS6 setzt sich aus **R**öders **M**aschinen **S**teuerung und der Anzahl von Antriebsachsen, die von Sercos Bus unterstützt werden, zusammen.

Der Quellcode der Maschinensteuerung ist in der Sprache C++ geschrieben. Als Entwicklungsumgebung wird Visual Studio 2008 des Unternehmens Microsoft Corporation verwendet.

5.5.1 Allgemeiner Aufbau

Die Aufteilung der Maschinensteuerung auf zwei Rechnersysteme erfordert die Entwicklung jeweils einer Anwendung, die die spezifischen Anforderungen des jeweiligen Systems erfüllt.

Entsprechend der Grundidee der Vererbung in der objektorientierten Programmierung werden für das jeweilige Rechnersystem spezialisierte Klassen von einer Datenbasis abgeleitet.

Für den NT-PC sind dies speziell Klassen, die über Dialoge die Verbindung zum Maschinenbediener herstellen, und für den RT-PC die Verbindung zum Schaltschrank der Maschine.

Die gesamte Maschinensteuerung ist grob in Module unterteilt, die jeweils ein Teilsystem der Maschine behandeln.

Die Kommunikation zwischen beiden Rechnersystemen wird durch den Austausch von Fräsersätzen realisiert. Dies sind Datenobjekte fester Größe, deren Inhalt modulspezifisch ist.

Jedes Modul enthält Funktionen, die beim Start und Beenden der Steuerung einmalig ausgeführt werden, Funktionen die kontinuierlich im Millisekundentakt aufgerufen werden und solche, die beim Empfang von Daten des jeweils anderen Rechners ausgeführt werden.

Aufgrund der Komplexität der Maschinen und Integration maschinenspezifischer Eigenschaften gibt es eine umfangreiche Konfigurationsdatei, in der für jedes Modul Optionen freigeschaltet und Parameter konfiguriert werden.

5.5.2 Handbox Modul

Die Grundlage jedes Moduls ist von der Basisklasse `c_Base` abgeleitet, so auch die Klasse `c_HdBox`. Diese überschreibt die Funktionen zum Einlesen des entsprechenden Abschnittes aus der Konfigurationsdatei. Über die Variablen der Konfigurationsdatei können der Verbindungstyp zum Handbediengerät, das Verhalten bei Erkennen eines von der Einstellung verschiedenen Handbediengeräts und die Datenübertragung zum Handbediengerät konfiguriert werden.

Von dieser Klasse werden wiederum die rechner-spezifischen Klassen für den RT-PC und den NT-PC abgeleitet. Für den Datenaustausch zwischen den Rechner-Systemen werden Objekte der Klasse `c_VarHdBox` verwendet.

5.5.3 Handhabung der Schnittstellen

Um die Anpassungen von bestehendem Code gering zu halten, werden für die RS232 Verbindung mit dem erweiterten Protokoll und die Ethernet-Verbindung jeweils eigene Klassen erstellt. Diese sind von der bestehenden Basisklasse `c_HdBoxInterface_Base` abgeleitet, die die Funktionen zur Integration in die gesamte Steuerung deklariert.

Die zu benutzende Verbindung zum Handbediengerät wird durch die Konfigurationsdatei vorgegeben. Eine Zustandsmaschine steuert und überwacht permanent die Verbindung zum Handbediengerät.

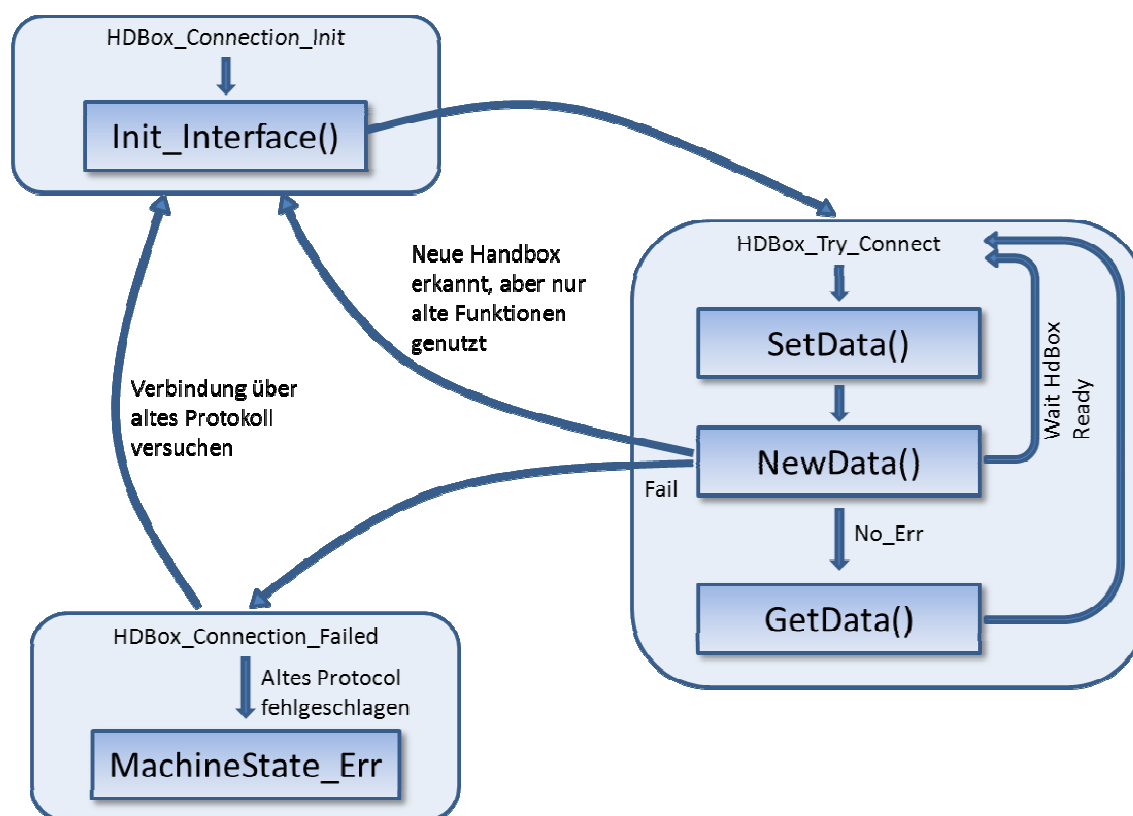


Abbildung 5.8 Zustandsmaschine für Schnittstelle zum Handbediengerät

Der Zustand `HDBox_Connection_Init` ist der Ausgangszustand in dem die Maschine startet. Nach erfolgreicher Initialisierung der konfigurierten Schnittstelle wird im Zustand `HDBox_Try_Connect` der Datenaustausch mit dem Handbediengerät verwaltet. Wird das alte Protokoll verwendet, aber anhand des Identifikationsbit im Datenbyte des Handbediengeräts das neue Produkt erkannt, kann die Schnittstelle erneut initialisiert werden, um den erweiterten Funktionsumfang nutzen zu können. Diese Option wird durch eine Variable in der Konfigurationsdatei freigeschaltet.

Um einen sicheren und unterbrechungsfreien Betrieb zu gewährleisten, wird bei einer Fehlfunktion des erweiterten Protokolls automatisch eine Verbindung über das alte Proto-

koll aufgebaut. Erst bei Versagen des alten Übertragungsprotokolls wird die Maschine in einen Fehlerzustand überführt.

5.5.4 Datenverwaltung

Um Einstellungen, die die Darstellung und das Verhalten des Handbediengeräts beeinflussen, konfigurieren zu können, wird eine separate Datei verwendet. Ebenso werden Zeichensätze, die zur Darstellung von Texten auf dem Display des Handbediengeräts benutzt werden, in Dateien gespeichert. Um die Datenintegrität sicherzustellen, beinhaltet jeder Datensatz Checksummen sowie ein Eigenschaftsfeld um Veränderungen zu detektieren.

Damit die Daten im Speicher des Handbediengeräts auf dem gleichen Stand der Inhalte der Dateien sind, werden beim Start der Maschinensteuerung die Eigenschaftsfelder vom Handbediengerät angefordert und mit den Daten, die aus den Dateien gelesenen werden, verglichen.

Der Vergleich und die Entscheidung, ob Daten übertragen werden, finden auf dem NT-PC statt. Der RT-PC dient bei diesem Vorgang nur als Verbindung zwischen dem Handbediengerät und den auf dem NT-PC verfügbaren Daten.

Die Übertragung von Daten kann zusätzlich durch einen Parameter in der Konfigurationsdatei der Maschinensteuerung beeinflusst werden. Sie kann komplett unterbunden oder die Übertragung bestimmter Datensätze erzwungen werden.

NT-PC

Zur Verwaltung dieser Daten dient die Basisklasse `c_HdBoxDataCtrl`.

Die für die NT-Seite abgeleitete Klasse `c_NHdBoxDataCtrl` implementiert Funktionen zum Einlesen und Herausschreiben der Konfigurationsdatei und zum Einlesen der Schriftsatz-Dateien und Bitmaps.

Eine Zustandsmaschine steuert die Übertragung der Daten an den RT-PC, die an das Handbediengerät weitergeleitet werden. Zur Übertragung werden die Daten in Strukturen angelegt. Diese entsprechen den Telegrammen, die das Handbediengerät interpretieren kann. Sie werden für die Übertragung vom NT-PC an den RT-PC in ein Objekt der Klasse `c_VarHdBox` eingebunden.

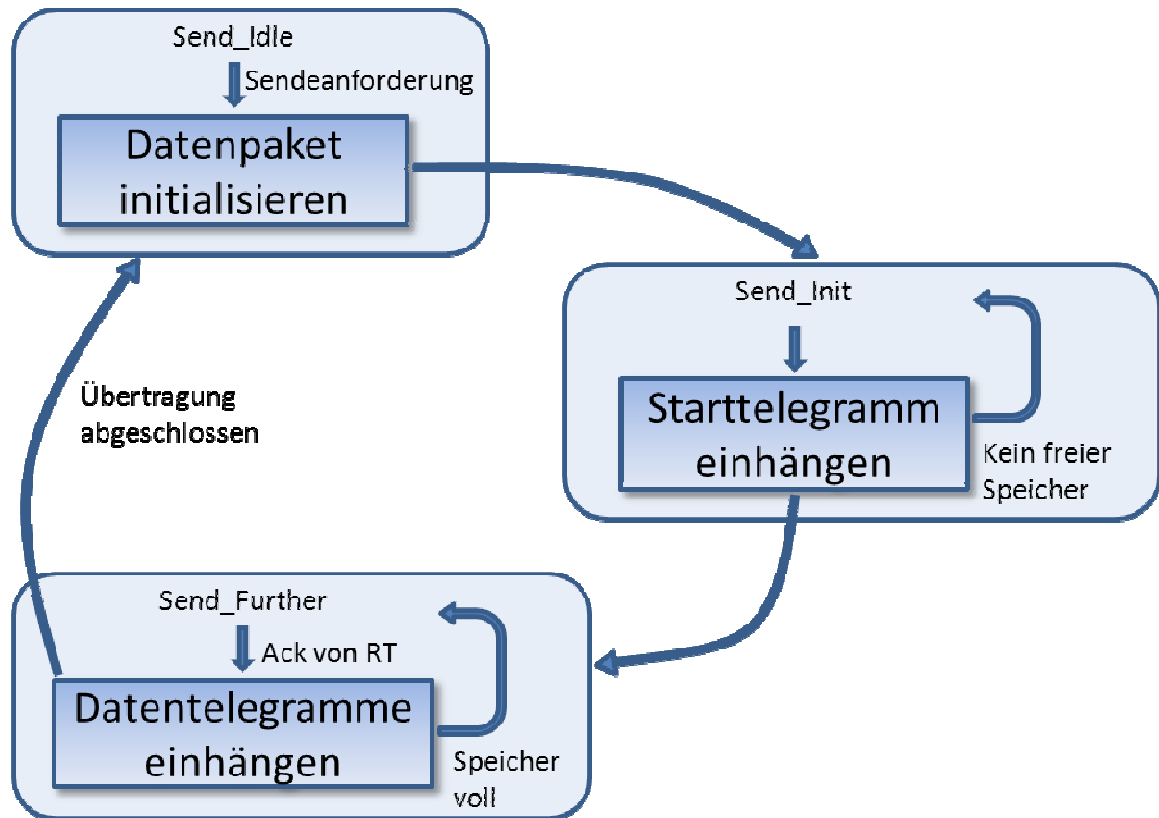


Abbildung 5.9 Zustandsmaschine für Datenversand von NT an RT

RT-PC

Die für die RT-Seite abgeleitete Klasse `c_RHdBoxDataCtrl` stellt eine Funktion zum Empfangen der Datenpakete vom NT-PC bereit. Eine Pufferung der Daten auf dem RT-PC ist erforderlich, da die Übertragung zwischen den beiden Rechnern nicht deterministisch ist.

Über eine Zustandsmaschine wird der Datenfluss zum Handbediengerät gesteuert. Entsprechend dem Maschinenzustand werden Daten zur Visualisierung bereitgestellt, oder Konfigurationsdaten, die vom NT-PC empfangen wurden, aus dem Buffer weitergeleitet.

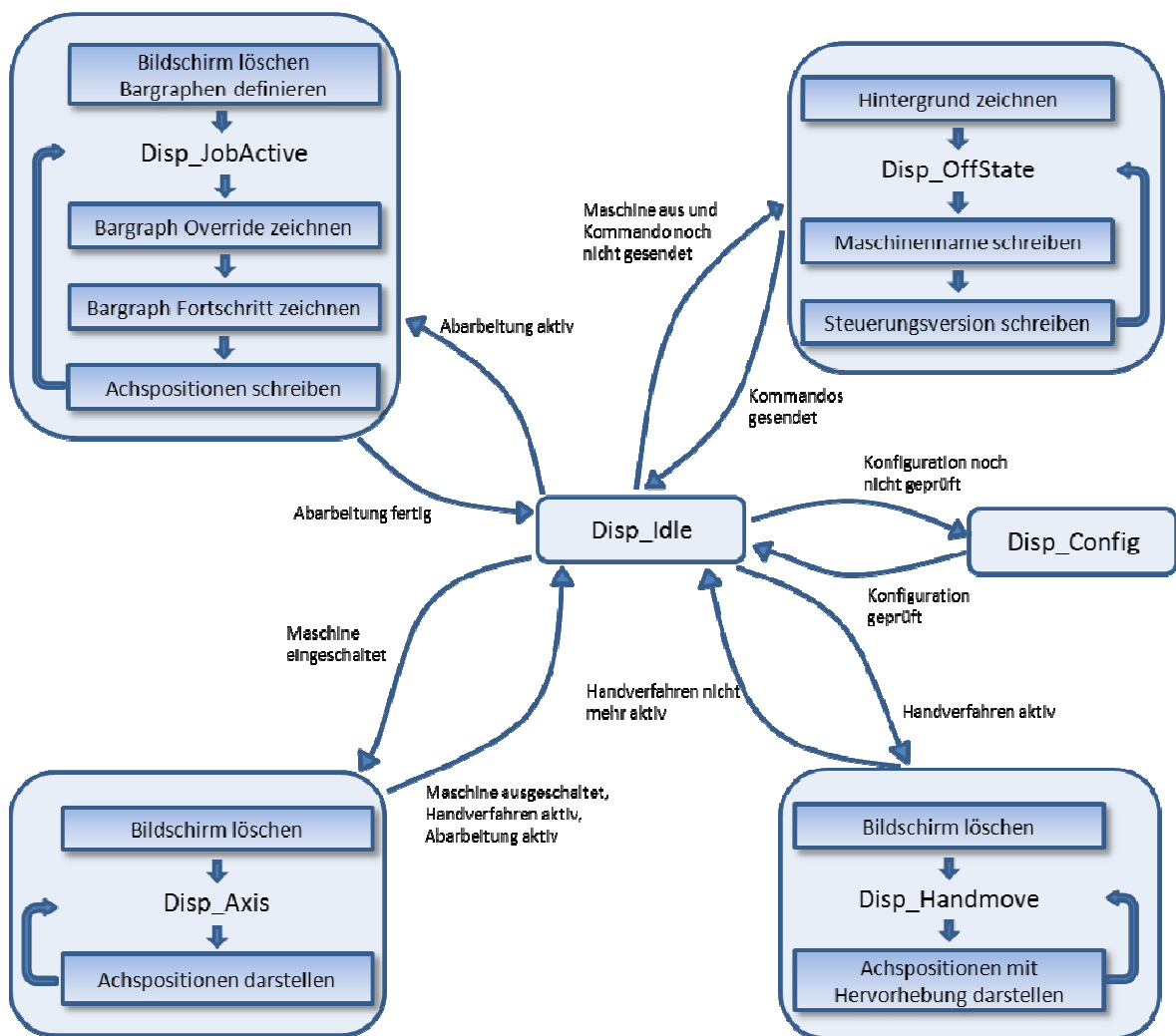


Abbildung 5.10 Zustandsmaschine auf RT der zu visualisierenden Daten

6 Integration

6.1 Inbetriebnahme Hardwarekomponenten

Der Stromverbrauch, der für die Auslegung der Spannungsversorgung mit 480mA angeommen wurde, ist deutlich geringer.

Das gesamte Handbediengerät benötigt im alleinstehenden Betrieb ca. 120mA. Bei aktiver Ethernet-Schnittstelle steigt der Stromverbrauch auf ca. 270mA.

Die Versorgung der kompletten Schaltung erfolgt durch einen der beiden Schaltregler.

Im Rahmen der Inbetriebnahme der Platine werden die Qualität und zeitlichen Verläufe einiger Signale geprüft, die für einen sicheren Betrieb essentiell sind.

Analyse der RS232 Schnittstelle

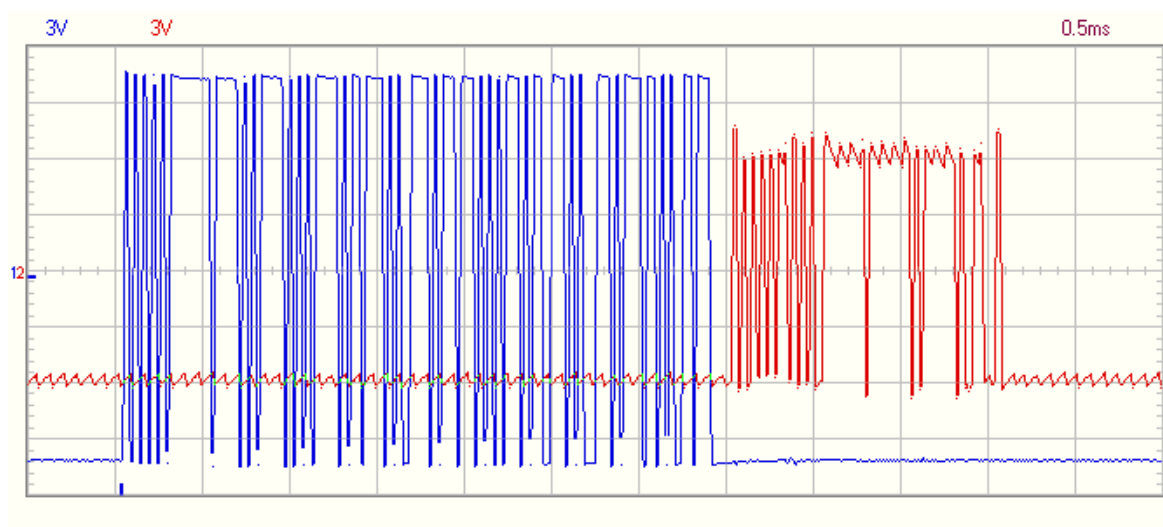


Abbildung 6.1 RS232 Übertragung jeweils eines Telegramms

Die gemessene Zeit zur Übertragung eines Telegramms von Maschinensteuerung an das Handbediengerät beträgt 3,4ms, die mit dem berechneten Wert von 3,38ms übereinstimmt.

Die hier gemessene Übertragungszeit von 1,6ms vom Handbediengerät an die Maschinensteuerung zeigt einen älteren Stand des Übertragungsprotokolls, bei dem der Datenframe nur sechs Bytes umfasst. Der dafür berechnete Wert beträgt 1,56ms.

6. Integration

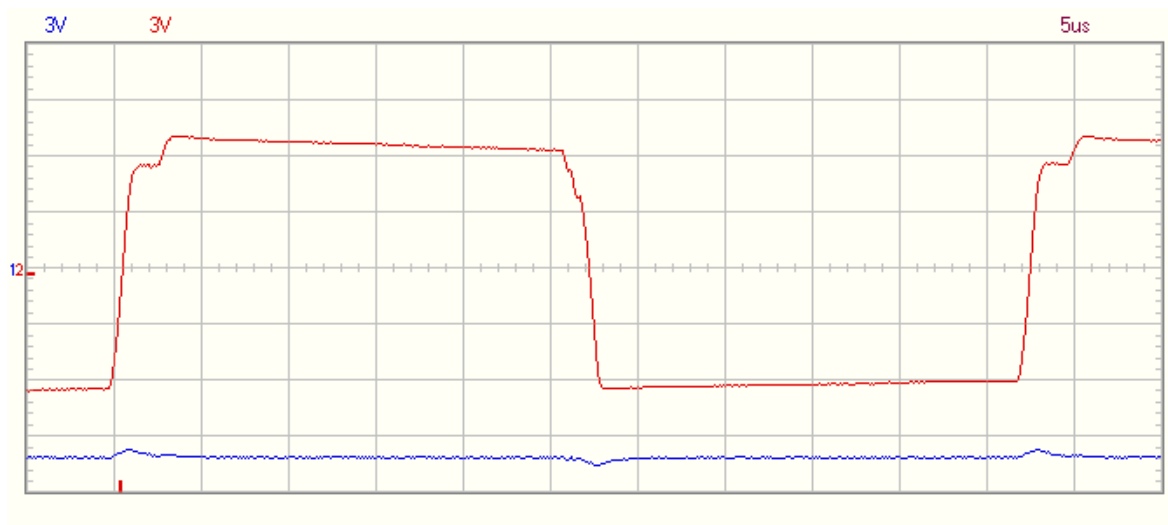


Abbildung 6.2 Signal auf Sendeleitung des Handbediengeräts

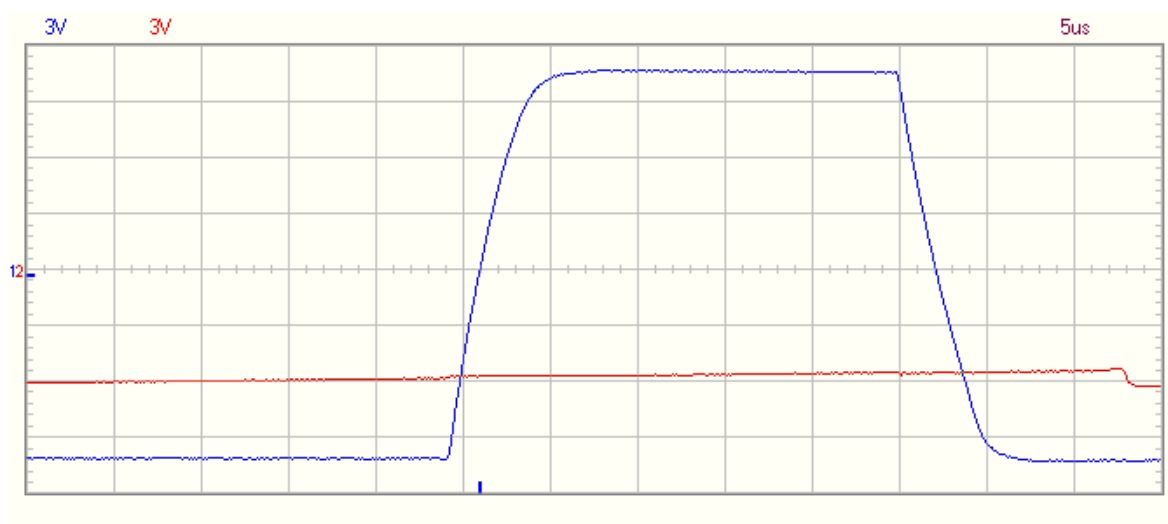


Abbildung 6.3 Signal auf Sendeleitung des Echtzeitrechners

Die Ausgangssignale des Leitungstreibers des Handbediengeräts und der Steckkarte des Rechners unterscheiden sich deutlich in Pegel und Flankensteilheit. Der Signalpegel der PC-Steckkarte liegt bei ca. $\pm 10,5$ V, der des Handbediengeräts bei ca. $\pm 7,2$ V. Zu sehen ist ein Knick in der steigenden Flanke und eine leicht abfallende Spannung gegenüber einem konstanten Pegel bei der PC-Steckkarte. Der Knick erklärt sich mit der Bereitstellung der Spannung über eine Ladungspumpe, die mit der 5 V Versorgungsspannung gespeist wird. Das Einknicken der Spannung deutet auf zu klein dimensionierte Blockkondensatoren hin.

Analyse der SPI Leitungen zum Display

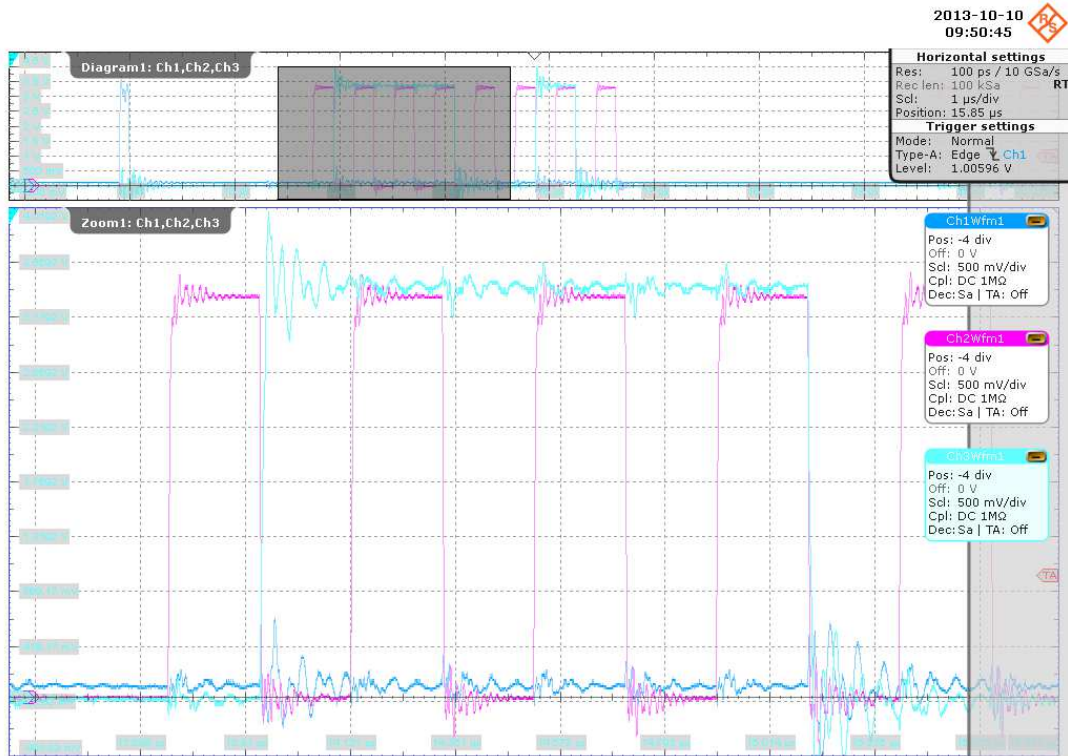


Abbildung 6.4 Takt- und Datensignal am Ausgang des Controllers

Die Abbildung zeigt den Spannungsverlauf von Taktleitung (Ch2) und Signalleitung (Ch3) zwischen Ausgang des Mikrocontrollers und den Serienwiderständen. Bedingt durch die hohe Flankensteilheit treten Überschwinger auf, die bei der Datenleitung besonders ausgeprägt sind.

6. Integration

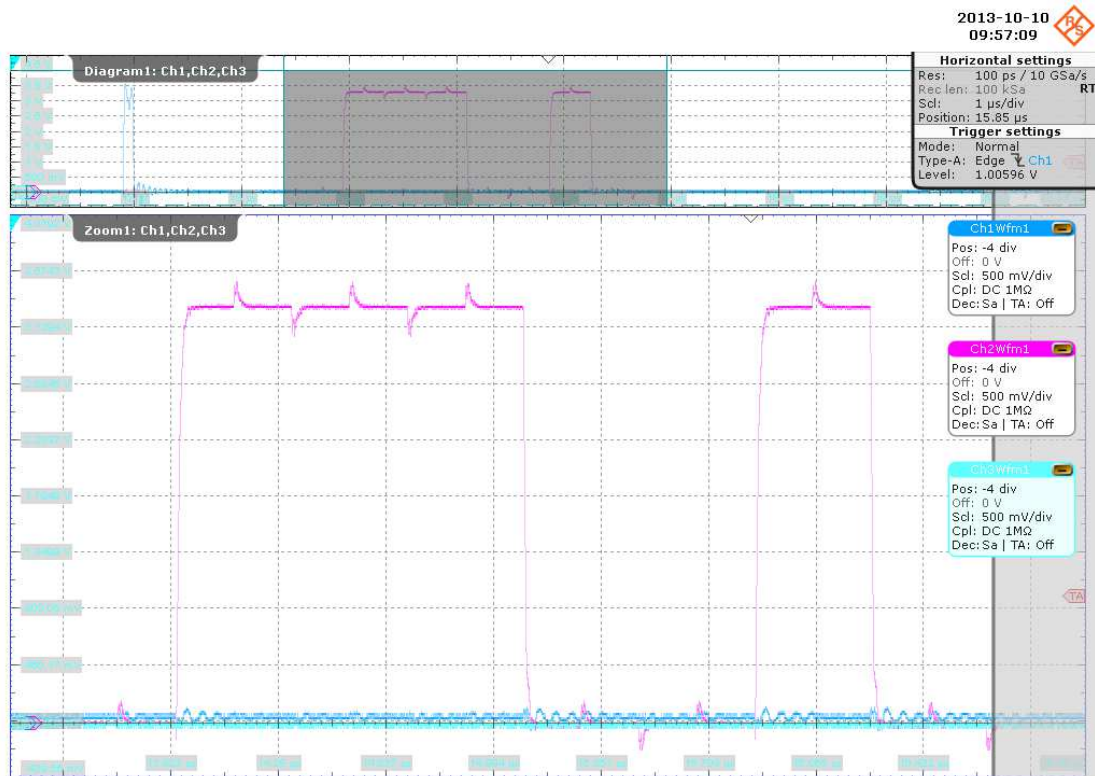


Abbildung 6.5 Übersprechen auf Datenleitung zwischen Controller und LCD

Diese Abbildung zeigt das gleiche Datensignal (Ch2) wie zuvor, hier gemessen zwischen dem Serienwiderstand und dem LCD. Durch den Widerstand wird die Flankensteilheit reduziert, wodurch das Überschwingen fast vollständig eliminiert wird. Es zeigt sich ein weiterer unerwünschter Effekt: das Übersprechen der Taktleitung auf die Datenleitung. Dies ist im vorliegenden Fall stark ausgeprägt, da die Signale über ein 200mm langes Flachbandkabel mit einem Abstand von 0.5mm zueinander laufen.

6.2 Mechanik und Gehäuse

Die Entwicklung eines eigenen, speziell auf die Anforderungen zugeschnittenen Gehäuses ist im Rahmen dieser Arbeit nicht zu leisten. Daher wird hier auf ein bestehendes Produkt einer Firma, die auf die Herstellung von Kunststoffgehäusen spezialisiert ist, zurückgegriffen. Die nötigen Bearbeitungen werden mit herkömmlichen Werkzeugen von Hand durchgeführt. Zum Schutz des LCD ist eine Plexiglasplatte auf der Frontseite aufgebracht.

Die benötigte Stärke des Haltemagneten zu berechnen gestaltet sich sehr aufwendig, da hier dessen Form und Material sowie die Fläche, an der das Handbediengerät gehalten werden soll, und der Abstand der Magnete zur Fläche Einfluss nehmen. Daher wurde hier der pragmatische Weg über empirische Versuche mit dem konkreten Produkt an der Maschine gewählt. Ergebnis ist die Implementierung von drei Magneten, die mit jeweils 60kg Haltekraft angegeben sind. Diese Kraft reduziert sich durch den Abstand der Magnete zum Maschinengehäuse auf ein Maß, dass das Handbediengerät sicher hält und sich ohne unnötig großen Kraftaufwand lösen lässt.

Die Anbindung an die Maschine erfolgt über ein Kabel, das auf einer Klemmleiste auf der Platine aufgelegt ist. Das Kabel wird durch das Gehäuse geführt und durch eine Zugentlastung gesichert. Die Verwendung einer Klemmleiste anstelle eines Steckers bietet eine, für einen Prototypen vorteilhafte, größere Flexibilität der Anschlussmöglichkeiten.

Für den Anschluss des Serienproduktes ist ein Stecker, der direkt auf der Platine aufgebracht ist und mit dem Gehäuse verschraubt wird, vorzuziehen.

Der Zustimmungstaster ist zunächst seitlich am Griffbereich angebracht. Eine Position an der Unterseite des Gehäuses erleichtert die Bedienung, kann aber zur Betätigung des Schalters bei Ablage des Handbediengeräts führen. Mit der Änderung der Befestigung des Handbediengeräts am Maschinenterminal ist eine unbeabsichtigte Betätigung nicht mehr möglich.

Zur Umsetzung der Tasten werden Standard-Folientastaturen verwendet und angepasst. Die Erstellung einer speziell auf diesen Prototypen zugeschnittene Folientastatur ist aus finanziellen Gründen nicht sinnvoll.

Softkeys, über die zusätzliche Steuerungsmöglichkeiten des Bildschirminhaltes realisiert werden können, sind in diesem Prototypen nicht umgesetzt. Der dafür zur Verfügung stehende Platz reicht nicht aus, um Standard-Folientastaturen zu verwenden.

Die Gestaltung der Adapterplatinen für Not-Aus-Taster und Handrad erfordert kleine Anpassungen, damit diese montiert werden können. Bedingt durch die Einbaulage des Not-Aus-Tasters muss eine Ecke der Adapterplatine abgerundet werden.

Die Befestigungsschrauben des Handrades erfordern eine Aussparung in dessen Adapter.

6.3 Kosten

Auch wenn die Kosten des Prototypen deutlich über denen des späteren Serienproduktes liegen, kann eine Einschätzung hinsichtlich zu erreichender Einsparungen getroffen werden.

Die Kosten des Prototypen sind identisch mit den Anschaffungskosten des aktuellen Handbediengeräts. Deutliche Einsparungen können in der Serienfertigung bei Platinen- und Bauteilkosten erzielt werden. Es wird die Annahme getroffen, dass die Fertigungskosten der Platine in Serie bei etwa 20% des Prototypen liegen und die elektronischen Bauteile zu etwa 35% des Einzelpreises bezogen werden können.

Zur Abschätzung des Einsparpotentials bei Komponenten wie Drehschaltern, dem Handrad und der Folientastatur wurden Angebote der jeweiligen Hersteller eingeholt.

Das in Summe resultierende Einsparpotential liegt bei etwa 40% gegenüber dem aktuellen Produkt.

6.4 Ergebnis



Abbildung 6.6 Prototyp des Handbediengeräts

6.5 Test

Der Test des Prototypen erfolgt zunächst am eigenen Arbeitsplatz. Identisch zum Aufbau einer Maschine wird hier ein System aus zwei Rechnern verwendet. Maschinenachsen und andere Peripherie werden simuliert. Mit dieser Testumgebung werden die implementierten Szenarien getestet.

Der Drehschalter zum Einstellen der Vorschubgeschwindigkeit in Kombination mit den Haltemagneten stellt sich problematisch dar, da die starken Magnetfelder die Positionserkennung störend beeinflussen. Der Drehschalter arbeitet mit einem Hallsensor, der die Verdrehung eines Magneten auf der Schalterwelle detektiert.

Um die einwandfreie Funktion des Drehschalters für die Vorschubgeschwindigkeit zu gewährleisten, müssen die Magnete entfernt werden.

Als alternative Befestigungsvariante des Handbediengeräts wird ein Haltewinkel angebracht. Dieser ist kompatibel mit der bestehenden Haltevorrichtung des Maschinenterminals.

Zum Testen des Produktes unter realen Bedingungen wird das Handbediengerät an eine Maschine im hauseigenen Formenbau angeschlossen.

Die durch den Wegfall der mechanischen Kontakte erwartete höhere Lebensdauer des Override-Drehschalters konnte im Betrieb nicht bestätigt werden, da dieser bereits nach zwei Wochen im Einsatz verschleißbedingt ein deutlich höheres Betätigungsmoment benötigte.

6.6 Feedback

Das hier genannte Feedback basiert auf den Aussagen der Maschinenbediener, die mit dem Prototyp des Handbediengeräts während der zweiwöchigen Testphase gearbeitet haben.

Nachteilig an einem im Handbediengerät integrierten Haltemagneten ist das Anhaften eisenhaltiger Späne am Gerät. Gleiches gilt für den Halteriemen, in dessen Oberfläche sich Späne verhaken können.

Im Vergleich zum aktuell eingesetzten Inkrementalgeber des Handrades verfügt das neue Handrad über deutlich mehr Schritte pro Umdrehung. Da die Skalierung der Inkremente zum daraus resultierenden Verfahrensweg nicht geändert wurde, kann eine deutlich höhere Vorschubgeschwindigkeit erreicht werden. Für den Handverfahrbetrieb ist ein Grenzwert

für die Vorschubgeschwindigkeit konfiguriert, der mit diesem Handrad überschritten werden kann, wodurch die Verfahrbewegung gestoppt wird. Dieser Umstand wird von Maschinenbedienern, die bereits lange Zeit mit dem bestehenden Handbediengerät arbeiten, als störend empfunden.

Das für den Prototypen verwendete Gehäuse verbessert die Handhabung des Gerätes nicht, da die Grifffläche zu klein ist und die Frontseite durch ihre Breite eine Bedienung der Tasten mit der haltenden Hand ausschließt. Gerade diese einhändige Bedienbarkeit wird gewünscht.

Insgesamt wird von dem Handbediengerät ein schlichtes und funktionales Design gefordert.

7 Fazit

7.1 Zusammenfassung

Ziel dieser Arbeit ist die Entwicklung eines Handbediengeräts als funktionaler Prototyp zur manuellen Steuerung von Hochgeschwindigkeits-Fräsmaschinen.

Die funktionalen Anforderungen an dieses Gerät, die durch das bestehende Produkt definiert sind, werden hinsichtlich ihres Optimierungspotentials untersucht.

Die Umsetzung dieser Anforderungen und die Anordnung der Komponenten orientiert sich hauptsächlich an dem bestehenden Produkt, da sich dieses etabliert hat und die Maschinenbediener an den Umgang gewöhnt sind.

Der Funktionsumfang ist um ein Display erweitert über das, je nach Maschinenstatus, die Positionen der Maschinenachsen und Prozessabläufe dargestellt werden. Neben dem praktischen Nutzen wird durch die Verwendung eines Farbdisplays eine optische Aufwertung des Handbediengeräts erreicht.

Darüber hinaus besteht die Möglichkeit weitere Tasten anzubringen, über die die Darstellung von Informationen gesteuert werden kann.

Die Datenübertragung zwischen Handbediengerät und Maschinensteuerung erfolgt weiterhin über eine RS232 Schnittstelle. Dazu wird ein erweitertes Kommunikationsprotokoll verwendet.

Die Integration einer Ethernet-Schnittstelle ist zurückgestellt, da bei einigen Maschinentypen kein Platz für die erforderliche Steckkarte auf dem Echtzeitrechner verfügbar ist.

Das Platinendesign ist auf eine kosteneffektive Fertigung ausgelegt und berücksichtigt die Anforderungen für eine automatische Bestückung. Vor allem der verwendete Mikrocontroller wird hinsichtlich des Obsolescence-Managements betrachtet.

Für die Datenübertragung zwischen Handbediengerät und Maschinensteuerung ist ein erweitertes Kommunikationsprotokoll entwickelt, das speziell auf die serielle Schnittstelle angepasst ist. Die Kompatibilität des Prototypen zu dem bestehenden Produkt ist durch automatische Umschaltung des Protokolls vollständig gegeben.

7.2 Bewertung

Eine kosteneffektivere Umsetzung ist im Bereich der Drehschalter möglich. Da dies eine grobe Änderung in der Handhabung bedeutet, werden höhere Kosten zu Gunsten einer unveränderten Bedienung in Kauf genommen.

Ein günstiger und einfach handzuhabender Mikrocontroller mit 8Bit Technologie stößt bei den zum flüssigen Darstellen von Inhalten auf einem Display benötigten Datenraten an seine Grenzen. Die daraus resultierende geringe Aktualisierungsrate fällt beim Zustandswechsel der Maschine und dem damit verbundenen Neuzeichnen der Displays negativ auf.

Das verwendete Gehäuse dient primär der Integration der entwickelten Elektronik, zur Erstellung eines Prototypen. Um eine Verbesserung im Bereich der Bedienbarkeit und Haptik zu erreichen, bedarf es der Entwicklung eines auf die Anforderungen und verwendeten Komponenten zugeschnittenen Gehäuses.

Das Feedback der Maschinenbediener ist mit gewissem Abstand zu betrachten. Die Dauer der Arbeit mit dem bestehenden Produkt hat großen Einfluss auf die Aufgeschlossenheit gegenüber Veränderungen. Dies ist besonders bei dem geänderten Handrad aufgefallen.

Mit den gewonnenen Erfahrungen im Umgang mit dem zur Entwicklung der Platine verwendeten Programm, ist eine bessere Integration der Komponenten und der Platine in ein Gehäuse zu erreichen. Die Unterstützung vom Import von CAD Daten der einzelnen Bauteile ermöglicht die komplette Konstruktion des Gerätes am Computer.

Damit kann der benötigte Bauraum der Komponenten genauer berücksichtigt und die Platinen in ihrer Größe und Anordnung von Steckern genau angepasst werden.

7.3 Ausblick auf Serienfertigung

Zur Erreichung der Serienreife des Produktes ist die Entwicklung eines auf die Anforderungen abgestimmten Gehäuses unumgänglich. Mit verfügbaren Standardgehäusen ist eine ergonomische Integration der benötigten Komponenten kaum möglich.

Um eine optimale Integration der Komponenten in einem Gehäuse zu erzielen, ist ein Re-Design der Platine nötig. Dabei sollte die Kabelverbindung zur Maschine durch einen

7. Fazit

Stecker erfolgen, der direkt auf der Platine aufgebracht ist und mit dem Gehäuse verschraubt wird.

Im Zuge des Re-Designs kann die Spannungsversorgung auf die benötigte Leistung optimiert und die Ansteuerung des Displays verändert werden um eine höhere Aktualisierungsrate zu erreichen.

Damit im Betrieb, nach der Auslieferung des Handbediengeräts, Updates der Software möglich sind, ist die Implementierung eines Bootloaders auf dem Mikrocontroller sinnvoll.

Literaturverzeichnis

Fleury, P. (26. 05 2013). *Peter Fleury Online*. Abgerufen am 19. 06 2013 von <http://homepage.hispeed.ch/peterfleury/avr-software.html>

Franz, J. (2011). *EMV Störungssicherer Aufbau elektronischer Schaltungen*. Vieweg+Teubner.

Gerhard Schnell, B. W. (2012). *Bussysteme in der Automatisierungs- und Prozesstechnik*. Berlin: Vieweg+Teubner.

IDEC. (2013). *IDEC*. Abgerufen am 17. 04 2013 von http://www.idec.de/produkte/-broschueren/sicherheitsprodukte/HE_Zustimmtaster-ge.pdf

Microchip Technology Inc. (21. 08 2008). *Microchip*. Abgerufen am 18. 06 2013 von <http://ww1.microchip.com/downloads/en/AppNotes/00833c.pdf>

mikrocontroller.net. (kein Datum). Abgerufen am 12. 08 2013 von <http://www.mikrocontroller.net/articles/CRC>

Plate, P. J. (08. 07 2013). *Grundlagen Computernetze*. Abgerufen am 16. 07 2013 von <http://www.netzmafia.de/skripten/netze/netz0.html#0.1>

Racz, Z. (kein Datum). *elektronikwissen.net*. Abgerufen am 12. 08 2013 von <http://elektronikwissen.net/kommunikation/14-8-bit-crc-auf-die-schnelle.html>

Texas Instruments. (09 2002). Abgerufen am 13. 08 2013 von <http://www.ti.com/lit/an/slla037a/slla037a.pdf>

Wetec. (Juli 2009). *Wetec Systemlieferant*. Abgerufen am 23. 08 2013 von http://www.wetec.de/file-upload/dl_127_ESD-Leitfaden.pdf

Abkürzungsverzeichnis

CNC	Computerized Numerical Control
OSI-Modell	Open Systems Interconnection Model
MAC	Media Access Control
PHY	Physical Layer
SPI	Serial Peripheral Interface
IC	Integrated Circuit
ICSP	In Circuit Serial Programming
LCD	Liquid Crystal Display
SCSI	Small Computer System Interface
CRC	CyclicRedundancy Check
UART	Universal Asynchronous Receiver Transmitter
EDA	Electronic Design Automation
ESD	Electro Static Discharge
EMV	Elektromagnetische Verträglichkeit

Versicherung über die Selbstständigkeit

Hiermit versichere ich, dass ich die vorliegende Arbeit im Sinne der Prüfungsordnung nach §16(5) APSO-TI-BM ohne fremde Hilfe selbstständig verfasst und nur die angegebenen Hilfsmittel benutzt habe. Wörtlich oder dem Sinn nach aus anderen Werken entnommene Stellen habe ich unter Angabe der Quellen kenntlich gemacht.

Ort, Datum

Unterschrift

Anhang

A. Kommunikationsprotokoll Handbediengerät - RT-PC

Projekt:	Protokoll zur Kommunikation zwischen Handbediengerät RMSHB1 und Maschinensteuerung RMS6
Autor:	Jan Möhlmann
Version:	2.3
Letzte Änderung:	18.06.2013 (Erstellung) 24.07.2013 (komplette Überarbeitung zu V2.0) 13.08.2013 (Anpassungen der Datenordnung in den Telegrammen) 07.10.2013 (Erweiterung des Statustelegramms für weitere Taster)

1 Einleitung

Die bisherige Kommunikation zwischen Handbediengerät und Maschinensteuerung beschränkt sich auf den Austausch eines Bytes über die RS232 Schnittstelle mit einer Zykluszeit von 1ms. Dies war durch den nur 1 Byte großen Empfangspuffer unter Windows DOS und die Zykluszeit von 1ms der Maschinensteuerung vorgegeben.

Voraussetzung für das hier beschriebene Protokoll ist Windows CE, da hier der Empfangspuffer nicht mehr auf 1 Byte beschränkt ist.

2 Allgemein

Das neue Protokoll überträgt die Daten in Frames. Die Übertragung geschieht, bedingt durch den Treiber der seriellen Schnittstelle unter WinCE, streng alternierend.

Die Übertragung erfolgt in folgendem Datenformat:

8 Datenbits
Keine Parität
2 Stopbits

Um die Voraussetzung für eine deterministische Datenübertragung zu bieten, werden Frames mit fester Länge verwendet. Ein Statustelegramm umfasst 7 Bytes und ein Kommandotelegramm 13 Bytes.

Bei o.g. Frameaufbau ergibt sich bei einer Baudrate von 38400 Bit/s eine Übertragungsdauer von 1,8229ms bzw. 3,3854ms.

Die Zykluszeit der Übertragung richtet sich nach der von Treiber und Software benötigten Zeit zum Einlesen des Empfangspuffers und Schreiben des Sendepuffers.(meist 11ms)
Durch die Checksummen können Übertragungsfehler erkannt werden. Daher ist ein Filtern der Statusbytes, die von der Handbox an die Maschine übertragen werden, nicht erforderlich.

(Zum Vergleich: Beim alten Übertragungsprotokoll wird eine Änderung der Schaltzustände der Tasten Ok, Stop, Zustimmung und Not-Aus in der Maschinensteuerung erst nach einer Wartezeit von 50ms, während der sich der Zustand nicht ändern darf, übernommen.)

3 Übertragung

3.1 Altes Protokoll

3.1.1 Steuerbyte (vom PC zur Handbox)

- Bit 7: frei
- Bit 6: frei
- Bit 5: frei
- Bit 4: frei
- Bit 3: frei
- Bit 2: Kontrolllampe für Werkzeugwechsel
- Bit 1: Kodierung des zurückzusendenden Datenwortes
- Bit 0: Kodierung des zurückzusendenden Datenwortes

Bit 1	Bit 0	zurückzusendendes Datenwort
0	0	nicht benutzt
0	1	Datenwort 1
1	0	Datenwort 2
1	1	Datenwort 3

Bit 2	Kontrolllampe
0	ausgeschaltet
1	eingeschaltet

3.1.2 Datenbyte (von der Handbox zum PC)

Datenbyte 1

- Bit 7: frei
- Bit 6: Drehschalter für Achsanwahl
- Bit 5: Drehschalter für Achsanwahl
- Bit 4: Drehschalter für Achsanwahl
- Bit 3: Drehschalter für Override
- Bit 2: Drehschalter für Override
- Bit 1: Drehschalter für Override
- Bit 0: Drehschalter für Override

Bit 6	Bit 5	Bit 4	zurückzusendendes Datenwort
0	0	0	nicht benutzt
0	0	1	Achse X
0	1	0	Achse Y
0	1	1	Achse Z
1	0	0	Achse A
1	0	1	Achse B
1	1	0	Achse S
1	1	1	ext. Achse (Reserve)

Bit 3	Bit 2	Bit 1	Bit 0	zurückzusendendes Datenwort
0	0	0	0	Override 0%
⋮	⋮	⋮	⋮	⋮
1	1	1	1	Override 150%

Datenbyte 2

- Bit 7: Taste 'Stop' gedrückt (1)
- Bit 6: Taste 'Start' gedrückt (1)
- Bit 5: Taste '+' gedrückt (1)
- Bit 4: Taste '-' gedrückt (1)
- Bit 3: Taste 'Werkzeugwechsel' gedrückt (1)
- Bit 2: linke Zustimmungstaste gedrückt (1), falls nicht vorhanden rechte Zustimmungstaste gedrückt
- Bit 1: rechte Zustimmungstaste gedrückt (1)
- Bit 0: Not Aus gedrückt (1)

Datenbyte 3

Anzahl der Inkremente des Handrades seit dem letzten Übertragen des Datenwortes 3 als vorzeichenbehaftete Integer-Zahl im Zahlenbereich +127 (0x7F) ... -128 (0x80).

3.2 Statustelegramm von Handbediengerät an Maschine

Die Übertragung von Handbediengerät zu Maschinensteuerung beschränkt sich in ihrem Inhalt auf vier Bytes. Da diese aber sensible Daten enthalten, muss eine fehlerhafte Übertragung erkannt werden. Dazu wird eine 8bit CRC Checksumme verwendet.

Der Datensatz, der von der Handbox an den PC übertragen wird, ist wie folgt aufgebaut:

- Byte 1: Startzeichen (0x55)
- Byte 2: Datenbyte1
- Byte 3: Datenbyte2
- Byte 4: Datenbyte3
- Byte 5: Datenbyte4
- Byte 6: Acknowledge
 - 1: Kommando richtig übertragen und vollständig
 - 2: Kommando richtig übertragen, weitere Kommandos zur Vervollständigung erwartet
 - 3: Fehler in der CRC Checksumme
 - 4: Datenabfolge inkonsistent
 - 5: Unbekanntes Kommando
 - 6: Overflow (Controller kann Daten momentan nicht abarbeiten)
 - 128: Debugdaten
 - Bit5 (+32): optionales Bit, das einen Frame-Error signalisiert
 - Bit6 (+64): optionales Bit, das Schreibvorgang auf Handbox signalisiert
- Byte 7: CRC8 Checksumme

Start	B1	B2	B3	B4	Ack	Check
0x55	DB1	DB2	DB3	DB4	1...6	CRC8

Der Inhalt von Datenbyte1-3 entspricht den Spezifikationen des alten Protokolls. Datenbyte4 enthält Informationen über die Zustände von zusätzlichen Tasten des Handbediengeräts.

3.3 Debugtelegramm von Handbediengeräten Maschine

Der Debugdatensatz ist im Aufbau identisch mit dem Statusdatensatz. Anstelle der vier Datenbytes werden Debuginformationen übertragen.

Start	B1	B2	B3	B4	Ack	Check
0x55	D1	D2	D3	D4	8	CRC8

Zum Abgleich der Konfigurationsdaten auf der Handbox und der Konfiguration der Maschinensteuerung können die Checksummen der jeweiligen Datensätze angefordert werden.

3.3.1 Status der Handbox-Konfiguration

Mit diesem Telegramm kann die Maschinensteuerung verglichen, ob die Konfiguration der Handbox identisch mit der aus der Konfigurationsdatei ist.

Start	B1	B2	B3	B4	Ack	Check
0x55	Ctrl	DV	Check	ResetSource	128	CRC8

Ctrl: Enum Wert aus Anforderungstelegramm (zur Kontrolle)
 Entspricht Konfigurationsabschnitten aus Tabelle 3.5
 DV: Daten gültig
 0: Daten inkonsistent
 1: Daten konsistent
 Check: berechnete Checksumme über die Konfigurationsstruktur
 ResetSource: Quelle des letzten Resets der Handbox

3.4 Tabelle der Rückgabewerte

Zahlenwert	Beschreibung
1	Kommando richtig übertragen und vollständig
2	Kommando richtig übertragen, weitere Kommandos zur Vervollständigung erwartet
3	Fehler in der CRC Checksumme
4	Datenabfolge inkonsistent
5	Unbekanntes Kommando
6	Overflow (Controller kann Daten momentan nicht abarbeiten)
128	Debugdaten
Bit5 (32)	optionales Bit, das einen Frame-Error signalisiert
Bit6 (64)	optionales Bit, das Schreibvorgang auf Handbox signalisiert

3.5 Kommandotelegramm von Maschine an Handbox

Die Kommandotelegramme enthalten einen Header, der die Verwendung des Datenpaketes definiert.

Der Header ist wie folgt aufgebaut:

- Byte 1: Startzeichen (0x55)
- Byte 2: Verwendung der nachfolgenden Daten
- Bit7: Status der Werkzeugwechsel-Kontrollleuchte
- Bit6-Bit0: enum Wert (siehe Tabelle)
- Byte 3: Steuerbyte, das die Verwendung spezifiziert (für jede Verwendung frei definierbar)
- Byte 4: Spare Byte
- Byte 5: Daten Byte 7
- ...
- Byte 12: Daten Byte 0
- Byte 13: CRC8 Checksumme (Checksumme (unteren 8Bit der Summe aller Bytes des Telegramms))

3.5.1 Telegramm für eine Achsposition

Die Position der Achsen wird zeichenweise übertragen

Start	Cmd	CtrlB	SB	B0	B1	B2	B3	B4	B5	B6	B7	Check
0x55	Pos	Axis	'-'	C1000	C100	C10	C1	C.1	C.01	C.001	C.0001	CRC8

Pos: enum Wert (siehe Tabelle)

Axis: Bit7 bestimmt, ob die Achsposition farblich hervorgehoben werden soll.
Die Farbe wird durch die Konfiguration bestimmt

Bit6 bestimmt, ob die Achsposition in reduzierter Größe dargestellt werden soll.

1 X-Achse

2 Y-Achse

3 Z-Achse

4 A-Achse

5 C-Achse

6 S-Achse

'-': Negatives Vorzeichen, wenn benötigt, sonst 0

C: Dezimalstellen als ASCII Zeichen

3.5.2 Telegramm für freie vmSs Ausgabe

Zum Darstellen von freien Texten auf dem Display sind Ausgabefont, Farbe und Position in X und Y Richtung erforderlich.

Start	Cmd	CtrlB	SB	B0	B1	B2	B3	B4	B5	B6	B7	Check
0x55	EvmSs	1	Font	Color	Color	PosX	PosX	PosY	Bkg	Bkg	X	CRC8

EvmSs: enum Wert (siehe Tabelle)

Font: enum Wert aus Konfiguration

3 klein

4 standard

5 groß

Color: Farbwert (B0 = MSB, B1 = LSB)

PosX: Position in Pixel vom linken Displayrand (B2 = MSB, B3 = LSB; max.320)

PosY: Position in Pixel vom oberen Displayrand (max.240)

Bkg: Farbwert des Hintergrundes(B5 = MSB, B6 = LSB)

Um längere Zeichenketten effizient übertragen zu können, können diese mit einem Folgetelegramm direkt angehängt werden. Dazu muss mit jedem folgenden Telegramm das CtrlB inkrementiert werden. Font und Farbe bleiben bestehen, die Position wird automatisch fortgeführt.

Start	Cmd	CtrlB	SB	B0	B1	B2	B3	B4	B5	B6	B7	Check
0x55	EvmSs	2	X	C1	C2	C3	C4	C5	C6	C7	C8	CRC8

3.5.3 Telegramm für Datenablage im Flash

Um Bilder, Fonts und Konfigurationsdaten im Flash-Speicher der Handbox ablegen zu können, werden diese in Telegrammen mit jeweils 8 Bytes Daten übertragen.

Um ein zufälliges Schreiben in den Speicher zu verhindern, muss ein

Kommandotelegramm voran gehen, in dem Speicheradresse und Gesamtlänge der Daten angegeben sind. Damit die nachfolgenden Telegramme eindeutig sind, wird eine Laufvariable mit jedem Telegramm inkrementiert. Diese wird im Steuerbyte und Sparebyte übertragen (Sparebyte = MSB)

Telegramm zum Starten eines Speicherschreibvorgangs:

Start	Cmd	CtrlB	SB	B0	B1	B2	B3	B4	B5	B6	B7	Check
0x55	Flash	1	0	X	X	X	Len	Len	Len	Len	Addr	CRC8

Flash: enum Wert (siehe Tabelle)

Len: Länge der insgesamt zu übertragenden Daten als UNS4 in Byte (B3 ist MSB und B6 ist LSB)

Addr: Speicherbereich in dem die Daten abgelegt werden sollen (entspricht den Konfigurationsabschnitten aus Tabelle 3.5)

Telegramm zur Datenübertragung und Ablage im Speicher:

Start	Cmd	CtrlB	SB	B0	B1	B2	B3	B4	B5	B6	B7	Check
0x55	Flash	2	0	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0	CRC8

Mit jedem weiteren Telegramm wird die Laufvariable, dargestellt in CtrlB und SB inkrementiert

3.5.4 Telegramm für Bargraphen

Zur Darstellung eines Bargraphen müssen zwei Eckpunkte, die Rahmenfarbe und eine ID zugewiesen werden. Über die ID kann einem Bargraphen ein neuer Wert zugewiesen werden, ohne ihn neu zu definieren.

Telegramm zur Definition eines Bargraphen:

Start	Cmd	CtrlB	SB	B0	B1	B2	B3	B4	B5	B6	B7	Check
0x55	Bar	1	ID	X1	X1	Y1	X2	X2	Y2	Color	Color	CRC8

BarG: enum Wert (siehe Tabelle)

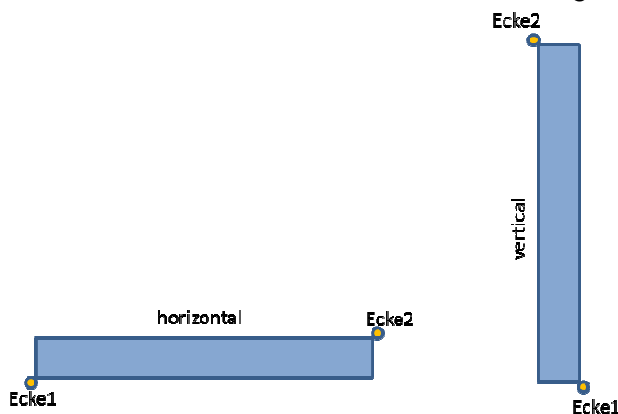
ID: Identifier für diesen Bargraphen (Wertebereich von 0 bis 3)

X1/Y1: Position der Ecke von der aus sich der Graph füllt (B0 = MSB, B1 = LSB)

X2/Y2: Position der Ecke an dem der Graph endet (B3 = MSB, B4 = LSB)

Color: Rahmenfarbe (B6 = MSB, B7 = LSB)

Die Ecken werden so definiert, als würde der Graph horizontal betrachtet. Dabei muss Ecke1 unten links und Ecke2 oben rechts liegen.



Telegramm zum Zuweisen eines Wertes zu einem Bargraphen:

Start	Cmd	CtrlB	SB	B0	B1	B2	B3	B4	B5	B6	B7	Check
0x55	Bar	2	ID	Value	Value	X	X	X	X	Color	Color	CRC8

Value: bestimmt den Füllgrad des Graphen. Wert in Pixel (maximal die Positionsdifferenz der beiden Ecken)

Color: Füllfarbe (der nicht gefüllte Bereich wird mit der Hintergrundfarbe gezeichnet) (B6 = MSB, B7 = LSB)

3.5.5 Telegramm für Bilddarstellung

Zum Darstellen eines Bildes aus dem Speicher der Handbox muss folgendes Telegramm gesendet werden.

Start	Cmd	CtrlB	SB	B0	B1	B2	B3	B4	B5	B6	B7	Check
0x55	Bild	Pic	0	X	X	X	X	X	X	X	X	CRC8

Pic: Nummer des anzuzeigenden Bildes (1 oder 2)

3.5.6 Anfordern des Debugtelegramms

Das Debugtelegramm der Handbox soll nur beim Start oder bei ausgeschalteter Maschine angefordert werden.

Start	Cmd	CtrlB	SB	B0	B1	B2	B3	B4	B5	B6	B7	Check
0x55	Debug	Conf	0	X	X	X	X	X	X	X	X	CRC8

Conf: enum Wert für welche Daten die Checksumme angefordert werden soll
(Konfigurationsabschnitte aus Tabelle 3.5)

3.6 Tabelle der Kommandos

Symbol	Zahlenwert	Beschreibung
Nop	0	Keine Aktion, nur Status antworten
Pos	1	Positionsdaten einer Achse
EvmSs	2	Freie vmSs Ausgabe mit beliebiger Position, Font und Farben
Flash	3	Datenpakete zur Speicherung im Flash
Bild	4	Darstellen von Bild1 aus Speicher
Clear Screen	5	Display löschen mit Standardhintergrundfarbe
Bar	6	Darstellung eines Bargraphen
Debug	16	Senden des Debugtelegramms

3.7 Tabelle der Konfigurationsabschnitte

Zahlenwert	Beschreibung
0	Software Version
1	Sektor 1, Konfigurationsdaten
2	Sektor 2, Font klein
3	Sektor 3, Font mittel
4	Sektor 4, Font groß
5	Sektoren 5,6,7, Bild1
6	Sektoren 8,9,10, Bild2

4 Speicherverwaltung

4.1 Konfigurationsdaten

Die Parameter für eine TCP/IP Verbindung sowie die Standard Ausgabefonts und Farben für Darstellungen auf dem Display werden in einer Konfigurationsstruktur im Flash-Speicher abgelegt. Diese Struktur ist wie folgt aufgebaut:

ENUM Wert: 1

Speichersektor 1

Enum	Speichersektor	Hauptgruppe	Element	Größe [Byte]	Beschreibung		
1	1	TCP/IP					
			IP Adresse	4			
			Subnetz-Maske	4			
			Gateway	4			
			Prim. DNS Server	4			
			MAC Adresse	6			
				Port	2		
				Darstellung			
			Font-Farbe		2	Standard-Schriftfarbe (16Bit)	
			Font-Farbe highlight		2	Schriftfarbe für Hervorhebung(16Bit)	
			Hintergrundfarbe		2	Standard-Hintergrundfarbe(16Bit)	
			Hintergrundfarbe highlight		2	Hintergrundfarbe für Hervorhebung(16Bit)	
			Font	1	Enum Wert für Standard-Font (3, 4, 5)		
			Info	Maschinename	64	Maschinename aus rms6.cfg	
				Steuerungsversion	16	Aus rms6.cfg	
				Hintergrundbild bei Start d. Handbox	1	Bild darstellen anstatt weißem Hintergrund	
			Filterzeiten	Standard-Tasten	1	Zeit in ms	
				Zusätzliche Tasten	1	Zeit in ms	
				Sicherheits-Taster	1	Zeit in ms	

4.2 Fonts

Ein Font enthält den kompletten ASCII Zeichensatz

Die Breite eines Zeichens in Pixel wird im ersten Byte des Sektors gespeichert, die Höhe im zweiten. Im dritten Byte wird die Größe eines Zeichens in Byte abgelegt. Im vierten der Offset der Adresse des Datenbeginns. Im fünften Byte wird die CRC8 Checksumme über die ersten vier Byte gespeichert.

Enum	Speicher-sektor	Hauptgruppe	Element	Größe [Byte]	Beschreibung
2	2	Font	Klein8x12 Pixel	3072	
3	3	Font	Mittel 16x24 Pixel	12288	
4	4	Font	Groß 24x40 Pixel	30720	

4.3 Bilder

Bilder haben eine maximale Größe von 320x240 Pixel bei einer Farbtiefe von 16Bits

Für den Fall, dass ein Bild nicht das komplette Display ausfüllt, muss dessen Größe bekannt sein. Dazu wird die horizontale Größe in Pixel in den ersten zwei Bytes, die vertikale in den zweiten zwei Bytes abgelegt. Im fünften Byte steht der Adressoffset des Bilddatenbeginns. Im sechsten Byte wird die CRC8 Checksumme über die ersten fünf Bytes gespeichert.

Enum	Speicher-sektor	Hauptgruppe	Element	Größe [Byte]	Beschreibung
5	5,6,7	Bild1		153600	
6	8,9,10	Bild2		153600	

Anhang

B. Schaltpläne RMSHB1