

Freie Universität



Berlin

MASTERARBEIT

Entwurf, Umsetzung und Validierung von Konzepten für Anwendungen im Industrie-4.0-Kontext

Autor:

Daniel AKRAP,

Matrikelnummer: 4520369

Betreuer:

Prof. Dr. Raúl ROJAS,

Freie Universität Berlin

M.Eng. Boris BECKMANN-DOBREV,

Fraunhofer IPK

14. Juli 2016

Inhaltsverzeichnis

Selbstständigkeitserklärung	v
Zusammenfassung	vii
Glossar	ix
1 Einführung	1
1.1 Motivation	1
1.2 Aufgabenstellung	1
2 Grundlagen	3
2.1 Industrie 4.0	3
2.1.1 Internet der Dinge	4
2.1.2 Cyber-Physisches-System (CPS)	5
2.2 Computerhaptik	6
2.2.1 Wahrnehmung	6
2.2.2 Haptische Interaktionsgeräte	7
3 Stand der Technik	9
3.1 Automatisierungstechnik	9
3.1.1 Speicherprogrammierbare Steuerung (SPS)	9
3.2 Netzwerktechnik	12
3.2.1 EtherCAT	14
3.2.2 Controller Area Network (CAN)	16
3.2.3 Profibus DP	18
3.3 Mikrocontroller	18
3.3.1 Arduino UNO	19
3.3.2 Arduino DUE	20
3.3.3 Teensy 3.2	20

3.4	Einplatinencomputer	20
3.4.1	Raspberry PI 2	21
3.4.2	BeagleBone Black	21
3.4.3	Texas Instruments Industrial Communications Engine (ICE)	22
4	Konzepte	23
4.1	Haptisches Interaktionsgerät Graspit	23
4.1.1	Aufbau des Graspit-Systems	23
4.1.2	Schematischer Aufbau des Graspit Systems	26
4.1.3	Analyse des bestehenden Systems	27
4.1.4	Entwurf eines neuen Konzepts	27
4.2	Haptikzelle	29
4.2.1	Systemaufbau	29
4.2.2	Zielsetzung	30
4.2.3	Hardware	30
4.2.4	Entwurf eines neuen Konzepts	31
4.3	Bi-Manu-Track	32
4.3.1	Zielsetzung	33
4.3.2	Systemaufbau	33
4.3.3	Entwurf eines neuen Konzepts	36
5	Umsetzung	39
5.1	Graspit-System	39
5.1.1	Hardware	39
5.1.2	Software	40
5.1.3	Steuerung	41
5.2	Bi-Manu-Track	42
5.2.1	Evaluation	43
5.2.2	Entwicklung einer Systemschnittstelle	48
5.2.3	Schaltplan	49
5.2.4	Implementierung	50
6	Auswertung	55
6.1	Graspit-System	55
6.1.1	Performancemessung	55
6.1.2	Validierung	56

6.2	Bi-Manu-Track	57
6.2.1	Validierung	57
7	Fazit	59
	Literatur	61
	Abbildungsverzeichnis	65
	Tabellenverzeichnis	67
	Abkürzungen	69

Selbstständigkeitserklärung

Ich erkläre gegenüber der Freien Universität Berlin, dass ich die vorliegende Arbeit selbstständig und ohne Benutzung anderer als der angegebenen Quellen und Hilfsmittel angefertigt habe.

Die vorliegende Arbeit ist frei von Plagiaten. Alle Ausführungen, die wörtlich oder inhaltlich aus anderen Schriften entnommen sind, habe ich als solche kenntlich gemacht.

Diese Arbeit wurde in gleicher oder ähnlicher Form noch bei keiner anderen Universität als Prüfungsleistung eingereicht und ist auch noch nicht veröffentlicht.

Datum, Ort

Unterschrift

Zusammenfassung

Ein Ziel von Industrie 4.0 ist es, Informationstechnologie und Produktionstechnologie besser miteinander zu verknüpfen. Basierend auf dieser Vorgabe wurde aus informationstechnischer Sicht untersucht, wie bestehende oder neuentwickelte eingebettete Systeme diese Anforderung erfüllen können. Dafür wurden im Rahmen dieser Arbeit am Fraunhofer Institut für Produktionsanlagen und Konstruktionstechnik (Fraunhofer IPK) mehrere Konzepte erarbeitet. Zwei dieser Konzepte wurden umgesetzt. Für das am Fraunhofer IPK entwickelte haptische Interaktionsgerät GRASPiT! wurde unter Verwendung aktueller Automatisierungstechnik eine neue Steuerung entwickelt. Das zweite System ist ein Rehabilitationsgerät zur Verbesserung der motorischen Fähigkeiten von Hand und Arm nach einem Schlaganfall. Das eingebettete System wurde mit einer Netzwerkschnittstelle erweitert, wodurch eine telemedizinische Nutzung ermöglicht wird.

Glossar

Anweisungsliste

AWL ist eine sich an Assembler orientierende Sprache für die Entwicklung speicherorientierter Steuerungen.

Busklemme

Busklemmen sind von der Bauform genormte Klemmen, die auf einer DIN-Hutschiene in einem Schaltschrank montiert werden. Sie können analoge und digitale Eingangs- und Ausgangssignale verarbeiten.

Carrier Sense Multiple Access/Collision Resolution (CSMA/CR)

CSMACR ist eine Methode zur Erkennung und Vermeidung von Kollisionen in Netzwerken.

Common Industrial Protocol

Das Common Industrial Protocol (CIP) ist ein Anwendungsprotokoll für die Automation, das den Übergang der Feldbusse zum Industrial Ethernet und zu IP-Netzen unterstützt. [1].

Complementary metal-oxide-semiconductor

Complementary Metal Oxide Semiconductor (CMOS) ist eine Schalttechnik für Logikbaustein. Eine Betriebsspannung von 3,0V bis 3,6V ist üblich.

Dehnungsmessstreifen

Dehnungsmessstreifen ändern durch Ausdehnung oder Stauchung ihren Messbaren Widerstand.

Endeffektor

Ein Endeffektor ist das letzte Glied einer kinematischen Kette.

Enhanced Parallel Port

EPP ist ein bidirektionale Datenübertragungsmodus für die parallele Schnittstelle im PC.

Field Programmable Gate Array (FPGA)

Ein FPGA ist ein integrierter Schaltkreis. In diesem kann eine programmierte Logik gespeichert werden.

IEC 61131

"IEC 61131 legt die Syntax und Semantik von Programmiersprachen für speicherprogrammierbare Steuerungen fest. Dieser Teil (Teil 3 [Anm. d. Verf.]) der IEC 61131 legt die Syntax und Semantik einer vereinheitlichten Reihe von Programmiersprachen für Speicherprogrammierbare Steuerungen (SPS) fest. Diese Reihe umfasst zwei Textsprachen Anweisungsliste (AWL) und Strukturierter Text (ST) und zwei grafische Sprachen Kontaktplan (KOP) und Funktionsbaustein-Sprache (FBS)." [1].

Jitter

Jitter ist bei der Übertragung von digitalen Signalen ein Störsignal im Takt.

Kernelspace

Kernelspace ist unter Linux der privilegierte Speicherbereich.

Kontaktplan

Kontaktplan ist eine an Stromlaufpläne angelehnte Programmiersprache für speicherprogrammierbare Steuerungen.

Real Time Application Interface

RTAI (Real Time Application Interface) ist eine Erweiterung für Linux zu einem Echtzeitbetriebssystem.

Real-Time Operating System

TI-RTOS ist ein Echtzeitbetriebssystem von Texas Instruments.

Realtime Hardware Abstraction Layer

Ein Kernelmodul für Linux. Das Modul abstrahiert die Hardware vom Linux-Kernel. Anfragen an die Hardware werden an den RTAI Echtzeit-Kernel weitergeleitet. Hardwareinterrupts werden vom RTAI Echtzeit-Kernel an Linux weitergeleitet.

Shield

Shield ist die Bezeichnung für Erweiterungsplatinen der Arduino Familie.

Strukturierter Text

ist eine objektorientierte Sprache zum programmieren von SPS nach International Electrotechnical Commission (IEC) 61131. Die Syntax der Programmiersprache ist an Pascal angelehnt.

Transistor-Transistor-Logik (TTL)

Die Transistor-Transistor-Logik (TTL) ist eine Schaltungstechnik für Logik Bausteine. Die Betriebsspannung liegt bei 5V.

Userspace

Der Userspace ist der Speicherbereich, indem unter Linux die Anwendungen der nicht privilegierten Benutzer ausgeführt wird.

1 Einführung

1.1 Motivation

Die vorliegende Masterarbeit ist am Fraunhofer IPK entstanden. An diesem Institut werden Methoden und Technologien für das Management, die Produktentwicklung, den Produktionsprozess und die Gestaltung industrieller Fabrikbetriebe erarbeitet. Zu den Hauptgeschäftsfeldern Zählen Automatisierungstechnik, Unternehmensmanagement, Medizintechnik, Produktionssysteme und virtuelle Produktentstehung.

Industrie 4.0 steht für den Ansatz einer neuen industriellen Revolution. Durch die Anwendung von aktuellen Kommunikationstechniken sollen Prozesse in der produzierenden Industrie optimiert, modularisiert, durch vorhandene Ressourcen besser genutzt und die Produktionskosten gesenkt werden. Prozesse der Industrie besser zu integrieren, ist schon vor der Prägung des Begriffs Industrie 4.0 ein Schwerpunkt der Forschung am Fraunhofer IPK gewesen.

Eines der Ziele von Industrie 4.0 ist es, Informationstechnologie und Produktionstechnologie besser miteinander zu verknüpfen. Vor diesem Hintergrund wird aus informationstechnischer Sicht untersucht, wie bestehende eingebettete Systeme in diese neue Art von Produktionsablauf integriert werden können.

1.2 Aufgabenstellung

Klassische Anwendungen der Industrie sehen in sich geschlossene Lösungen vor, in denen Maschinen oder Anlagen vorgegebene, sich wiederholende (optimalerweise standardisierte) Routinen mittels Speicherprogrammierbare Steuerung überwachen und ausführen. Eine interaktive Fernsteuerung und Überwachung solcher Anlagen ist nach aktuellem Stand der Technik möglich. Die Industrie-4.0-Anforderungen gehen jedoch darüber hinaus. Zum Zwecke der Planung, Instandhaltung oder Modifikation ist die Online-Kopplung einer realen mit einer virtuellen Anlage erforderlich.

Technische Standards allein reichen jedoch nicht aus. Ein Großteil der Wertschöp-

fung im Maschinen- und Anlagenbau, aber auch in der Automatisierung besteht heute schon aus der Softwareentwicklung. Deren Komplexität nimmt weiter stark zu. Softwaresysteme für Industrie 4.0 müssen ökonomisch tragfähig produziert werden, leistungsfähig und zuverlässig sein und trotz wachsender Komplexität beherrschbar bleiben. [2] Ausgehend von den Anforderungen an Industrie 4.0 ist die Zielsetzung dieser Arbeit innovative Konzepte für Anwendungen im Industrie 4.0 Kontext zu erarbeiten. Dies beinhaltet im Einzelnen:

- Aufnahme von Anwendungsfällen im Industrie-4.0-Kontext,
- Ableitung von technischen Anforderungen, die für die Umsetzung der definierten Anwendungsfällen erforderlich sind,
- fundierte Recherche über den Stand der Technik und Stand der Forschung für aktuelle und in Entwicklung befindlichen Steuerungslösungen,
- Erarbeitung von Konzepten für die Umsetzungsmöglichkeiten der definierten Anwendungsfälle,
- Umsetzung ausgewählter Konzepte und
- Validierung und Performancemessungen der realisierten Lösungen.

2 Grundlagen

2.1 Industrie 4.0

Industrie 4.0 ist heute ein Aktionsfeld von „Digitale Wirtschaft und Gesellschaft“ der Hightech-Strategie der Bundesregierung aus dem Jahre 2014. Der Öffentlichkeit wurde der Begriff erstmals im Jahr 2011 auf der Hannovermesse bekannt gemacht. „Im Mittelpunkt steht eine Vernetzung von autonomen, sich situativ selbst steuern, sich selbst konfigurierenden, wissensbasierten, sensorgestützten und räumlich verteilten Produktionsressourcen (Produktionsmaschinen, Roboter, Förder- und Lagersysteme, Betriebsmittel) inklusive deren Planungs- und Steuerungssysteme“ [3, S. 24].

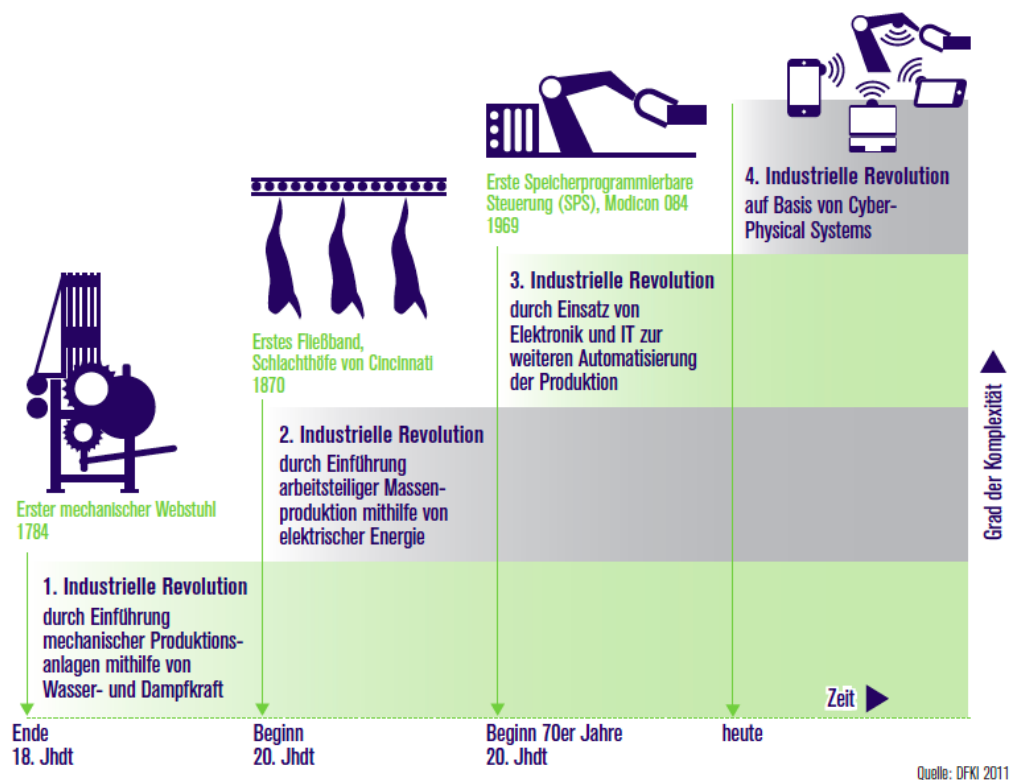


Abbildung 2.1: Industrieentwicklung, Quelle DFKI 2011

Der Begriff Industrie 4.0 soll zum Ausdruck bringen, dass es sich hierbei um die Einleitung der vierten industriellen Revolution handelt. Die erste industrielle Revolution, die Nutzung der Dampfkraft, ging im 18. Jahrhundert von England aus. Als zweite industrielle Revolution wird die Einführung der Fließbandarbeit Ende des 19. Jahrhunderts in Europa und die Massenfertigung zu Beginn der 1920er Jahre in den Vereinigten Staaten von Amerika betrachtet. Die dritte industrielle Revolution, auch als digitale Revolution bekannt, wurde mit der Einführung von Informationstechnologie und Elektronik beginnend in den 1960er Jahre eingeleitet. [4] In Bezug auf die dritte industrielle Revolution gehen die Meinungen stark auseinander. Die Umstrukturierung unserer Wirtschaft hin zu einer mehr ökologischen Form, wird auch als dritte industrielle Revolution bezeichnet, tritt unter dieser Bezeichnung seit Mitte der 2000er Jahre auf. [5][6]

Anknüpfend an die „digitale Revolution“, soll mit Industrie 4.0 die Vernetzung zwischen Menschen, Maschinen und Produkten mit Hilfe aktueller Informationstechnologie erfolgen. Diese Vernetzung von physischen Systemen (SPS, Mikrocontrollern etc.) erfolgt auf der Grundlage der Strukturen des Internet der Dinge (IdD).

2.1.1 Internet der Dinge

Das Internet der Dinge ist die Vernetzung von physischen Objekten in einem digitalen Netz. Dabei wird jedes physische Objekt mit einer virtuellen Identität verknüpft. Bei den physischen Objekten handelt es sich vielfach um Alltagsgegenstände wie Stromzähler, Kleidungsstücke, Uhren und verschiedenste Konsumgüter. Diese physischen Objekte werden meist mit einem eingebetteten System, bestehend aus einem Mikrocontroller, passender Sensorik und einer in der Regel auf Funktechnik basierenden Lösung zur Datenübertragung ausgestattet. Die von den Sensoren gesammelten Informationen (z.B. Puls, Bilder, Geschwindigkeit oder Temperatur) werden von dem Mikrocontroller aufgearbeitet, um zur weiteren Verarbeitung oder Auswertung mit einem Datennetzwerk übertragen zu werden. Weiter kann es sich auch um passive Systeme handeln, die mittels Radio-frequency identifications (RFIDs) oder mit einem Quick Response Code (QR-Code) ausgestattet sind. Entgegengenommen werden die Daten von anderen Maschinen, von denen sie auch weiterverarbeitet werden. Es handelt sich hierbei um eine Maschine-zu-Maschine-Kommunikation, die über gemeinsame Standards und Schnittstellen kommunizieren soll. Letztere sind noch nicht definiert. [7]

Anwendungsfelder

Für das Internet der Dinge ergeben sich Anwendungsfelder in fast allen Bereichen. Eine Auswahl dieser sind:

- Bekleidungsindustrie
- Verkehrstechnik
- Haushaltsgeräte
- Gesundheitstechnik

2.1.2 Cyber-Physisches-System (CPS)

„Cyber-Physische-Systeme sind die technische Lösung zur Verknüpfung der physikalischen und der virtuellen Welt. Als eingebettete Systeme erfassen sie mittels Sensoren unmittelbar physikalische Daten und interagieren auf dieser Grundlage aktiv oder reaktiv mit der physikalischen und der virtuellen Welt. Technisch gesehen sind eingebettete Systeme die Kombination aus Kleinstcomputern und Software, die in andere Objekte (Produkte, Materialien, Maschinen etc.) integriert sind.“ [8, S. 9].

Dies ist eine von vielen Definitionen eines CPS. Der Begriff wurde bis heute nicht klar abgegrenzt und überschneidet sich in anderen Definition deutlich mit denen von Industrie 4.0 wie die Definition der Forschungsagenda CPS: „Cyber-Physical Systems (CPS) sind gekennzeichnet durch eine Verknüpfung von realen (physischen) Objekten und Prozessen mit informationsverarbeitenden (virtuellen) Objekten und Prozessen über offene, teilweise globale und jederzeit miteinander verbundene Informationsnetze.“ [9, S. 2].

Anwendungsfelder

Cyber-Physische-Systeme sind in vielen Anwendungsfeldern präsent. Ein Auswahl dieser:

- medizinische Geräte und Systeme
- industrielle Prozesssteuerungs- und Automationssysteme
- Energieversorgungsmanagementsysteme
- Fahrerassistenzsysteme für Automobile

Cyber-Physisches-System in der Industrie 4.0

Da physische Systeme mit informationsverarbeitenden Elementen verknüpft werden, ist seit den 1970er Jahren gegeben. Der neue Aspekt bei Industrie 4.0 ist, dass diese Systeme über das Internet miteinander verknüpft werden. Die Basis hierfür wurde mit dem IdD geschaffen. Dieser Ansatz ermöglicht eine neue Anwendungsweise eines Cyber-Physischen-Systemes. Vorhandene Cyber-Physische-Systeme können miteinander verbunden werden, um neue Anwendungsszenarien zu bewältigen. Diese Verbindungen sind nicht starr, wodurch flexible und zu einem neuen Anwendungsfall entsprechende Kombinationen von Cyber-Physischen-Systemen möglich werden. Zum Beispiel ist es durch diese Vernetzung möglich, telemedizinische Geräte basierend auf vorhandenen Cyber-Physischen-Systemen der Medizin, zu entwickeln. Weiter können Cyber-Physische-Systeme (CPS) der produzierenden Industrie auf diese Weise flexibel zusammengefügt werden, damit zügig auf individuellen Kundenwünsche eingegangen wird oder um auch eine bessere Verteilung der Produktionslast zu reagieren.

2.2 Computerhaptik

Industrie 4.0 strebt eine Kopplung von virtuellen mit physischen Objekten an. Ein physisches Objekt erhält dafür ein virtuelles Pendant. Die gekoppelten Objekte werden „Digital Twins“ genannt. Mit Sensoren, die an dem physischen Objekt angebracht sind erhält das virtuelle Objekt die gemessenen Daten nach Möglichkeit in Echtzeit. Das können Koordinaten oder auch Informationen über den Zustand des physischen Objekts sein. Die Computerhaptik ist eine Form der Mensch-Computer-Interaktion. Sie ermöglicht es durch die Kopplung eines Eingabegeräts mit einem virtuellen Objekt in einer virtuellen Welt zu agieren. Das Eingabegerät verfügt über eine Krafterückkopplung, wodurch ein intuitives Interagieren im virtuellen Raum möglich wird. Für eine realitätsnahe Interaktion müssen die Eigenschaften der menschlichen Wahrnehmungsphysiologie eingehalten werden.

2.2.1 Wahrnehmung

Die Wahrnehmung spielt eine zentrale Rolle sämtlicher Organismen für das Erleben der eigenen Umwelt. Dieses Erleben beruht auf der Aktivität der zur Verfügung stehenden Sinnesrezeptoren [10]. Somit ist die Wahrnehmung (Perzeption) nichts

anderes als die Registrierung dieses Erlebens in Form von Reizen durch die vorhandenen Rezeptoren. Menschen haben ein breites Spektrum an Sinnesrezeptoren um Reize aufzunehmen und die Information derer durch das Nervensystem an das Gehirn weiterzuleiten. Der Mensch verfügt über 5 bekannte Sinne, unter anderem die Wahrnehmung des menschlichen Tastsinnes, welcher für die vorliegende Arbeit entscheidend ist. Die menschliche Haut beinhaltet nicht nur Rezeptoren für Temperatur und Schmerz, sondern auch für Druck und Vibration. Man unterscheidet zwischen einer passiven und einer haptischen Wahrnehmung. Die passive Wahrnehmung mechanischer Eindrücke wird auch als taktile Wahrnehmung bezeichnet und umfasst u.a. das Empfinden von Schmerz oder Temperatur. Die haptische Wahrnehmung vermittelt zum einen die Wirkung von mechanischen Reizen also auch Informationen über die Beschaffenheit von Gegenständen, wie deren Oberflächenstrukturen.

Für diese unterschiedlichen Wahrnehmungen verfügt der Mensch über spezifische Rezeptoren. Eines der wichtigsten Rezeptortypen des Menschen ist das Pacini-Körperchen, welches vermehrt in den Fingerspitzen vorliegt und Vibrationen wahrnimmt. Die Pacini-Körperchen reagieren auf mechanische Schwingungen zwischen $50 - 900\text{Hz}$, ihre maximale Empfindlichkeit liegt bei ungefähr 300Hz [11, S.684]. Somit können vibrierende Oberflächen bis zu einer maximalen Frequenz von ca. 1000Hz aufgelöst werden. [12, S. 25]

2.2.2 Haptische Interaktionsgeräte

Haptische Interaktionsgeräte dienen dem Anwender im virtuellen Raum, Objekte zu erfühlen, ertasten und auch zu ergreifen. Der Anwender bekommt als Bestätigung der Interaktion eine Kraftrückkoppelung, die über Motoren veranlasst wird. Man unterscheidet innerhalb der haptischen Interaktionsgeräte zwischen kinästhetischen und taktilen Interaktionen.

Die kinästhetische Interaktion umfasst Bewegungen, welche relativ große Kräfte bewirken, so dass Rezeptoren in den Muskulatur bzw. Knochen des Anwenders stimuliert werden. Bei der taktilen Interaktion werden vom Anwender langsame Bewegungen ausgeführt, wodurch Tastrezeptoren innerhalb der Haut stimuliert werden und somit Oberflächenstrukturen wahrgenommen werden. Kraftrückkopplungssysteme werden als passive oder aktive Interaktionsgeräte entwickelt. Bei einem passiven Interaktionsgerät folgt die Bewegung dem Anwender, dabei wird dieser nicht zusätzlich unterstützt. Mit einem aktiven Interaktionsgerät folgt der Anwender der Bewegung des Interaktionsgeräts.

3 Stand der Technik

Die Arbeit befasst sich mit der Umsetzung von Konzepten im Industrie 4.0 Kontext. Dafür soll in diesem Kapitel ein Überblick der aktuell verfügbaren Technologien geschaffen werden, mit denen sich die in dieser Arbeit entwickelten Konzepte realisieren lassen. Dafür werden Technologien aus dem Bereich der Automatisierungs-, Netzwerk- und Mikrocontrollertechnik vorgestellt.

3.1 Automatisierungstechnik

In diesem Abschnitt werden verfügbare Techniken zur Realisierung von Steuerungen und Regelungen vorgestellt.

3.1.1 Speicherprogrammierbare Steuerung (SPS)

„Speicherprogrammierbare Steuerungen (SPS) bilden seit den 80er Jahren das Bindeglied zwischen der Automation und der Computertechnik. Sie stellen eine Bedienerschnittstelle zur Verfügung mit der jeder Benutzer Automatisierungssysteme oder Produktionsmaschinen von unterschiedlichen Perspektiven aus betrachten und programmieren kann. Die Steuerung der entsprechenden Maschinen und Anlagen erfolgt über Aktoren, die von der speicherprogrammierbaren Steuerung über Leistungsstufen angesteuert werden.“ [13]

Vor der SPS wurden Steuerungen als Verbindungsprogrammierte Steuerung realisiert. Die Logik zur Steuerung eines Prozesses wurde mit fest verdrahteten Relais oder Schützen realisiert. Anzutreffen ist diese Art der Steuerung noch in Not-Aus Kreisläufen von Anlagen. Diese müssen unabhängig von der SPS laufen. Eine SPS besteht aus drei wesentlichen Komponente. Einem Betriebssystem, Eingängen und Ausgängen. Das Betriebssystem führt die programmierte Steuerung aus. Das Programm gibt vor, in welcher Reihenfolge ein Eingang gelesen oder ein Ausgang gesetzt wird. Die Steuerung kennt zu jeder Zeit des Programmablaufs die Werte, die an den

Eingängen anliegen. Die Eingänge erhalten durch angeschlossenen Sensoren ihre Daten. An den Ausgängen sind Aktoren angeschlossen. Entwickelt hat sich die SPS von einem geschlossenem System bestehend aus Computer, Betriebssystem und Ein- und Ausgängen hin zu einer modularisierten Lösung. Dabei lassen sich die Module in vier Gruppen einteilen:

1. Eine in Software implementierte SPS, die auf unterschiedlichen Computersystemen als Laufzeitumgebung der programmierten Steuerung dient. Genannt Soft-SPS.
2. Ein Computersystem welches die SPS ausführt.
3. In separate Baugruppen aufgeteilte Ein- und Ausgänge.
4. Ein Feldbus der zur Kommunikation der Einzelnen Komponenten untereinander dient.

Neben der Soft-SPS hat sich noch die Slot-SPS etabliert. Das ist eine Karte die in einem Personal Computer (PC) verbaut wird. Diese Karte hat einen eigenen Mikroprozessor. Der Mikroprozessor führt unabhängig von dem PC eine Steuerung aus. Weitere Baugruppen werden über einen Feldbus an die Slot-SPS angebunden. [14]

Im weiteren Verlauf werden ausgewählte Steuerungslösungen vorgestellt, die für eine Umsetzung der Arbeit nötig sind. Dabei wird auch kurz auf die Komponenten der Firma Beckhoff eingegangen. Dies geschieht vor dem Hintergrund, das Komponenten des Herstellers in Systemen die in dieser Arbeit behandelt werden, im Einsatz sind.

Beckhoff

Die Firma Beckhoff bietet im Bereich der Automatisierungstechnik verschieden Lösungen für eine Prozessautomation an. Diese teilen sich in folgende Felder auf:

Industrie PC (IPC): Die Steuerung wird in Form einer Soft-SPS auf einem Industrie- oder eingebetteten PC ausgeführt. Es werden IPCs und eingebettete PCs angeboten, die sich von der Konfiguration her zu einem PC in der Art des Gehäuses und der Bauform unterscheiden. Dies ist nötig um die Geräte in einem Schaltschrank verbauen zu können. Weiter wird damit garantiert, dass die Geräte vor äußeren Störeinflüssen in einem produzierenden Betrieb geschützt sind.

Input / Output (I/O): Zu dem Feld I/O zählen die *Busklemmen*. Sie verbinden Aktorik -und Sensorik mit der Steuerung. Die *Busklemmen* sind vom verwendeten Bussystem unabhängig. Mittels eines Buskopplers wird die Verbindung zu den *Busklemmen* hergestellt. Buskoppler gibt es für EtherCAT, Lightbus, PROFIBUS, Interbus, CANopen, DeviceNet ControlNet, CC-Link, Modbus. SERCOS und Ethernet.

Motion: Der Bereich Motion bietet verschiedene Servoverstärker für die Ansteuerung von Motoren. Die Servoverstärker haben den Regelkreis für die Motoren bereits integriert. Zur Verfügung stehen positionsbasierte, strombasierte und geschwindigkeitsbasierte Regelkreise.

Software: Für die Entwicklung von SPS - Programmen wird die Entwicklungsumgebung TwinCat bereitgestellt. TwinCat ist eine Erweiterung für Microsoft Visual Studio. Mit TwinCat lassen sich Programme in verschiedenen Programmiersprachen entwickeln. Dazu zählen *Kontaktplan*, *Anweisungsliste* und *Strukturierter Text*. Es wird sich an die *IEC 61131* Norm gehalten. Weiter kann mit TwinCat das Programm auch in C und C++ entwickelt werden. TwinCat erlaubt es das Bussystem Online zu debuggen, sowie an das Bussystem angeschlossenen Komponenten manuell zu manipulieren. TwinCat ist nur für Microsoft Windows erhältlich. Die Software ermöglicht eine Echtzeitsteuerung unter einem Windows Betriebssystem.

[15]

Controller Development System (CoDeSys)

CoDeSys ist eine geräteunabhängige Entwicklungsplattform für das Entwickeln von Steuerungsprogrammen. Mit CoDeSys ist es möglich Steuerungen nach *IEC 61131* Norm zu entwickeln. Weiter gibt es die Möglichkeit in der Programmiersprache C Routinen zu entwickeln und diese in die Steuerung zu integrieren.

CoDeSys ist eine Entwicklungsplattform bestehend aus CoDeSys Engineering, CoDeSys Runtime und weiteren Modulen für Anwendungen nach der *IEC 61131* Norm.

CoDeSys Engineering ist eine integrierte Entwicklungsumgebung für die Programmiersprachen nach *IEC 61131*. Die Entwicklungsumgebung beinhaltet Compiler für eine Reihe von Zielplattformen (C166, TriCore, 80x86, ARM/Cortex,

Power-Architecture, SH, MIPS, BlackFin). Der damit erzeugte Programmcode kann auf eine ausgewählte Zielplattform hochgeladen werden und auch mit Hilfe eines Online-Debuggers im Betrieb auf Fehler getestet werden.

CoDeSys Runtime (auch CoDeSys Controll genannt) wird auf die gewünschte Zielplattform installiert. Die CoDeSys Runtime ist die Soft-SPS, welche den von CoDeSys Engineering generierten Programmcode auf der Zielplattform ausführt. Zu den Zielplattformen zählen ARM/Cortex, x86 und viele mehr.

Real Time Application Interface

Real Time Application Interface (RTAI) ist eine Erweiterung für den Linux-Kernel. Mit RTAI wird die Möglichkeit geschaffen, unter GNU/Linux Prozesse in Echtzeit im *Userspace* und *Kernelspace* auszuführen. Ein Standard Linux-Kernel wird dafür mit RTAI gepatched. Der Patch fügt einen Echtzeit-Kernel zwischen der Hardware und dem Linux-Kernel ein. Weiter erhält der Linux-Kernel ein neues Kernel-Modul Namens *Realtime Hardware Abstraction Layer*. Das Kernel-Modul *Realtime Hardware Abstraction Layer* ist die Schnittstelle zwischen dem Echtzeit-Kernel und dem Linux-Kernel. Hardware Interrupts werden von dem Echtzeit-Kernel abgefangen und an die einzelnen Echtzeitprozesse weitergeleitet. Der Linux-Kernel selbst läuft als ein Echtzeitprozess von RTAI. [16]

3.2 Netzwerktechnik

In diesem Abschnitt werden Netzwerktechniken vorgestellt, die im Bereich der Automatisierungstechnik für das Vernetzen von Baugruppen mit Steuerungsanlagen genutzt werden. Verwendung in der Automatisierungstechnik finden verschieden Bussysteme, die unter dem Oberbegriff Feldbus zusammengefasst werden.

Ein Feldbus verbindet die Aktoren und Sensoren einer Anlage über ein Bussystem mit einer Steuerung. Unter der Verwendung von Kommunikationsprotokollen werden die Nachrichten mehrerer Teilnehmer über den Bus verschickt. Die Nachrichten beinhalten die Mess- und Stellwerte, sowie Statusinformationen und eine Kennung des Busteilnehmers. Feldbusse gibt es in Ausführungen verschiedenster Topologien. Dazu zählen die Baum-, Ring, Stern- und Bustopologie. Eine Mischung verschiedener Topologien ist auch möglich. Feldbusse die in der Industrie Anwendung finden sind nach IEC61158 und IEC61784 genormt. Das unterstützt die Verwendung von

Komponente verschiedener Hersteller in einem gemeinsamen System.

Vor der Einführung der Feldbustechnik wurde für ein Vernetzten der Komponenten die Parallelverdrahtung eingesetzt. Dabei wurde jedes Signal einer Komponente mit einer eigenen Leitung übertragen. Mit dieser Technik wurden sowohl analoge wie auch digitale Signale übermittelt.

Vorteile von Feldbussen gegenüber der Parallelverdrahtung sind ein geringerer Montageaufwand und eine höhere Flexibilität. Weiter bieten Feldbussysteme Möglichkeiten zur Eigendiagnose. Nachteile sind die höheren Reaktionszeiten und in Abhängigkeit der verwendeten Bustopologie kann die Ausfallmöglichkeit mehrerer Komponenten zugleich steigen.

Für verschiedene spezifische Anwendungsfelder ist eine enorme Anzahl an unterschiedlichen Feldbussen entstanden. Beim Entstehen dieser Arbeit existierten weit über 50 verschiedene Feldbus Umsetzungen. Eine Auswahl relevanter Feldbusse für diese Arbeit sind:

- CAN ist ein serielles Bussystem für die Vernetzung von Aktoren, Sensoren und Steuerungen. Er wird bevorzugt für die Vernetzung von Komponenten in Automobilen eingesetzt.
- DeviceNet basiert auf CAN Bus. Es nutzt das *Common Industrial Protocol* für die Datenübermittlung. Im Gegensatz zu CAN wird der Bus vermehrt in der Prozessautomation eingesetzt.
- EtherCAT ist ein auf Ethernet basierendes Protokoll für harte und weiche Echtzeitanforderungen in der Automatisierungstechnik.
- EtherNet/IP ist wie EtherCAT ein auf Ethernet basierendes Protokoll zu Erfüllung von Echtzeitanforderungen in der Automatisierungstechnik.
- Interbus ist ein serieller Feldbus für Sensoren, Aktoren und Steuerungen in der produzierenden Industrie.
- LIN-Bus ist ein serieller Feldbus für die Vernetzung von Aktoren und Sensoren. Der Bus wird in Systemen eingesetzt, die nur eine geringe Bandbreite für die Datenübertragung benötigen.
- Profibus ist ein Feldbus für die Vernetzung von Aktoren, Sensoren und Steuerungen in der produzierenden Industrie.

[17]

Folgend werden einige ausgewählte Feldbussysteme genauer erläutert. Diese zeichnen sich durch Eigenschaften aus, die eine Umsetzung der Aufgabenstellung vorantreiben würden.

3.2.1 EtherCAT

EtherCAT ist ein im IEC - Standard 61158 genormtes Protokoll, welches hauptsächlich Anwendung im Automatisierungsbereich findet. Es zeichnet sich durch geringe Zykluszeiten von unter $100\mu s$ und einem Jitter von weniger als $1\mu s$.

Eine Besonderheit von EtherCAT ist es, dass sich gängige Ethernetkarten mit einer EtherCAT-Firmware flashen lassen. Diese sind danach als EtherCAT-Master einsetzbar. Diese Möglichkeit besteht für die Betriebssysteme GNU/Linux und Windows [18, S. 1].

Funktionsprinzip

EtherCAT basiert auf Fast-EtherNet und benutzt ein Standard Ethernet-Frame zum versenden der Daten. Ein EtherCAT-Netzwerk besteht aus einem Master und n Slaves. In diesem Netzwerk wird ein Frame vom Master an *Slave1* gesendet, von *Slave1* an *Slave2* und dann an *Slaven*. Der Frame enthält für jede angeschlossene Stationen ein Datagram. Ein Datagram enthält das Prozessabbild aus einer angeschlossenen Station und eine eindeutige Identifikation. Das Prozessabbild beinhaltet alle Statusinformation und Prozessdaten einer Station.

Beim Durchlauf des Frames durch eine Station werden die für diese Station relevanten Prozessdaten ausgelesen und mit einem kurzen Versatz das eigene Prozessabbild im Datagram aktualisiert. Jeder Slave hat eine Gerätebeschreibungsdatei. In dieser ist festgelegt um was für eine Slave es sich handelt (z. B. Sensor oder Aktor) und welches Kommunikationsprofil genutzt wird. Dieses muss vor Initialisierung des Busses dem Master mitgeteilt werden [18, S.7].

Topologie

EtherCAT unterstützt Ring- Baum- Stern- und Linientopologien. Zwischen zwei Stationen (*Master – Slave* oder *Slave – Slave*) im EtherCAT Netzwerk kann ein Abstand von bis zu 100m sein. Während des Betriebs können Teilnehmer entfernt oder hinzugefügt werden [18, S. 6].

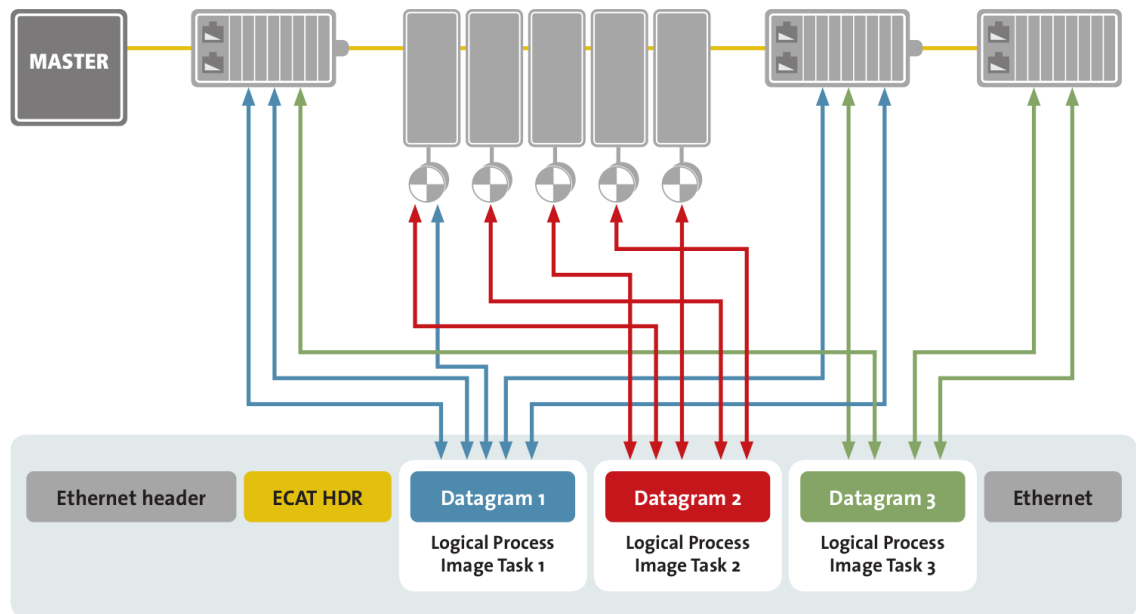


Abbildung 3.1: Prozessdaten im EtherCAT Telegramm [18, S. 7]

Protokolle

EtherCAT unterstützt neben EtherCAT auch die Verwendung anderer Kommunikationsprofile. Damit wird eine Migration von bestehenden Systemen hin zu EtherCAT erleichtert. Um dies zu realisieren stellt EtherCAT noch ein Mailbox genanntes Protokoll zur Verfügung. Die Slaves geben den EtherCAT Master über das Mailbox-Protokoll bekannt, welches Übertragungsprotokoll sie nutzen möchten. Die verfügbaren Protokolle eines Slaves sind in seiner Gerätebeschreibungsdatei hinterlegt. Ein Slave kann mehrere Protokolle unterstützen [18, S. 13]. Folgende Kommunikationsprofile sind implementiert:

- CAN application protocol over EtherCAT (CoE) integriert Baugruppen die für CANopen entwickelt wurden.
- Servo drive profile over EtherCAT (SoE) integriert Baugruppen mit einer Serial Realtime Communication System (SERCOS) - Schnittstelle. Dazu zählen vor allem Servoantriebe.
- Ethernet over EtherCAT (EoE) erlaubt es, Ethernet Pakete aus dem EtherCAT Network an Ethernetteilnehmer zu schicken.
- File over EtherCAT (FoE) erlaubt eine Aktualisierung der Firmware über das Bussystem.

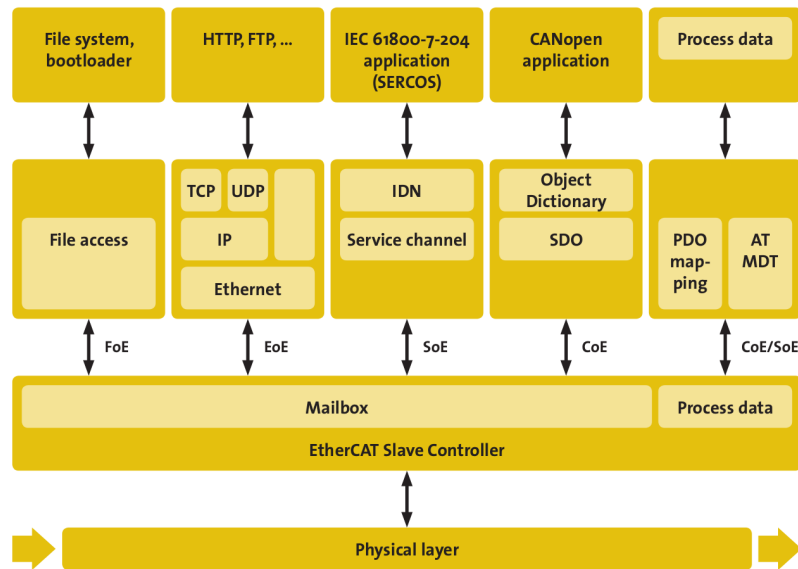


Abbildung 3.2: Struktur der Kommunikationsprofile im EtherCAT Telegramm [18, S. 13]

3.2.2 Controller Area Network (CAN)

Der CAN Bus ist ein Bussystem, welches die Daten seriell über zwei Leitungen überträgt. Das CAN kann eine Buslänge von bis zu 500m haben. Es zeichnet sich vor allem durch seine einfache Verkabelung und weite Verbreitung aus. Die maximale Teilnehmerzahl des Busses liegt bei 128. Das Bussystem wird überwiegend in der Automobilindustrie eingesetzt, dient aber auch als Grundlage für weitere Bussysteme in der Automatisierungstechnik wie CANopen und DeviceNet.

Funktionsprinzip

Der CAN Bus hat vier verschiedene Typen von Frames. Einen Daten-Frame, einen Remote-Frame, einen Error-Frame und einen Overload-Frame. Im Daten-Frame werden die Prozessdaten eines Busteilnehmers übertragen. Der Remote-Frame fordert einen anderen Busteilnehmer auf, einen Daten-Frame zu schicken. Ein Error-Frame teilt allen Teilnehmern mit, dass ein Fehlerfall aufgetreten ist. Der Overload-Frame generiert eine Empfangspause. Ist der Puffer des Empfängers voll, kann damit signalisiert werden das gerade kein Empfang von Daten möglich ist. Ein Frame hat eine eindeutige Identifikation. Diese ist gleichzusetzen mit der Identifikation des Busteilnehmers. Die Prozessdaten, die mit einem Daten-Frame übermittelt werden, können optional eine Objektidentifikation haben. Dadurch kann eine empfangende Station entscheiden, ob die Nachricht für sie relevant ist. Maximal könne 8Byte Nutzdaten

mit dem Daten-Frame übermittelt werden. Jeder Teilnehmer im CAN Bus ist Master und es sind alle gleichberechtigt. Bei Kollisionen von Frames wird *Carrier Sense Multiple Access/Collision Resolution (CSMA/CR)* zur Behebung des Konflikts benutzt.

Topologie

Der CAN Bus kann als Linien- oder Sterntopologie aufgebaut werden. Die Leitungsenden werden mit einem Abschlusswiderstand in der Größe von 120Ω terminiert. Die Leitungslänge ist abhängig von der gewählten Datenübertragungsrate. Je höher die Datenübertragungsrate ist, desto kürzer fällt die Leitungslänge aus (Tab. 3.1).

Leitungslänge	Übertragungsrate
40m	1MBit/s
100m	500kbit/s
500m	125kbit/s

Tabelle 3.1: CAN Bus Leitungslänge und Übertragungsrate

Protokolle

Auf dem Controller Area Network (CAN) basieren weitere Protokolle. Diese sind an spezielle Anforderungen angepasst. Zu den Anpassungen gehören verschieden physische Stecksysteme und Protokollanpassungen an das geforderte Szenario. Eine Auswahl der auf CAN basierten Bussysteme:

EnergyBus ist ein offener Standard für Energiemanagementsysteme. Der Bus wird für eine Kombination aus Energie- und Datenübertragung genutzt.

CANopen ist eine für die Automatisierungstechnik angepasste Version vom CAN. CANOpen führt als Kommunikationskonzept das Client-Server und das Erzeuger-Verbraucher Konzept ein. Weiter werden Kommunikationsobjekte festgelegt, die den Austausch von Prozessdaten regeln.

DeviceNet ist einer weiter für die Automatisierungstechnik angepasste Version. Sie limitiert die Anzahl der Teilnehmer pro Netzwerksegment auf 64 und unterstützt drei wählbare Übertragungsraten. Für die Übertragung von Daten wird das *Common Industrial Protocol* genutzt.

[19]

3.2.3 Profibus DP

Profibus DP (Dezentrale Peripherie) ist ein Bussystem zur Ansteuerung von Sensoren und Aktoren durch eine SPS. Das Bussystem kann nach dem Master-Slave oder Multi-Master Prinzip konzipiert werden. In einem Netzwerksegment sind bis zu 126 Teilnehmer zugelassen. Die Daten werden seriell übertragen bei einer maximalen Buslänge von 1200m. [20]

Funktionsprinzip

Der Master liest die Daten aller Slaves zyklisch ein. Nach der Verarbeitung der Informationen durch den Master werden die Daten zyklisch an alle Slaves und Master geschickt. Profibus DP nutzt das Token Ring verfahren für das Übermitteln von Telegrammen. Die Station die den Token hat ist berechtigt auf den Bus zuzugreifen. Profibus DP unterscheidet zwischen vier Telegrammtypen:

1. Keine Daten
2. Daten variabler Länge von 4 – 249Byte
3. Daten fester Länge sind 8Byte lang
4. Kurzquittierung über den Erhalt eines Telegramms.

Zur Unterscheidung der Telegramme verfügt jeder Typ über eine eigene Kennung.

Topologie

Profibus DB benutzt eine Mischung aus einer Ring- und Baumtopologie.

3.3 Mikrocontroller

Für die Umsetzung der Aufgabenstellung werden Mikrocontroller von Bedeutung sein. In diesem Abschnitt wird der aktuelle Entwicklungsstand der Technik betrachtet.

Mikrocontroller sind Prozessoren, bei denen für den Betrieb notwendige Komponenten wie der Speicher, Timer und die digitalen I/Os auf einem Chip integriert sind. Die Breite des internen Datenbus variiert zwischen 4, 8, 16 oder inzwischen auch 32Bit. Mikrocontroller zeichnen sich auch durch ihre kompakte Bauform und

niedrigen Energiebedarf aus. Durch die Fertigungstechnik von heute, lassen sich inzwischen die Mikrocontroller mit einem *32Bit* großen Datenbus günstig herstellen. Eine Übersicht verbreiteter Mikrocontroller:

- Atmel AVR ist eine auf der Reduced Instruction Set Computer (RISC)-Architektur basierende Mikrocontroller Familie. Die verschiedenen Modelle unterscheiden sich in der Größe des verbauten Speichers (Arbeitsspeicher sowie Flash-Speicher), der I/Os und der zur Verfügung stehenden Netzwerkschnittstellen
- ARM Cortex-M ist eine Serie von Mikrocontrollern die sich durch eine *32Bit* breiten internen Datenbus auszeichnen. Weiter haben die Modelle mehrere Netzwerkschnittstellen integriert wie CAN-Bus, LIN-Bus und Serial Peripheral Interface (SPI).

Programmiert werden Mikrocontroller in der Regel in Assembler oder C. Hersteller wie Atmel AVR bieten Bibliotheken für ihre Modelle an, um das Entwickeln zu erleichtern. Inzwischen lassen sich mit LabView von National Instruments Mikrocontroller auf Basis der ARM-Architektur und der ATmega-Reihe von AVR mit grafischen Werkzeugen wie LabView von National Instruments programmieren. Weiter hat sich mit dem Arduino Projekt ein eigenes Ökosystem entwickelt. Das Projekt bietet von der Hardwareplattform über die eigene Integrated Development Environment (IDE) bis hin zu Erweiterungen der Hardware Lösungen an. Diese sprechen eher Einsteiger und Hobbie-Entwickler an. Weiter bieten sie aber die Möglichkeit an, auch die aus dem Professionellen Umfeld stammenden Entwicklungswerkzeuge wie Atmel AVR Studio oder das Compilerframework GNU Compiler Collection (GCC) zu nutzen.

Für die Umsetzung der Arbeit wurde im Vorfeld eine Vorauswahl an zu verwendende Mikrocontroller getroffen. Diese werden in den Folgenden Abschnitten kurz vorgestellt.

3.3.1 Arduino UNO

Der Arduino UNO ist ein Mikrocontrollerboard mit einem AVR ATmega328 als Mikrocontroller. Das Board verfügt über sechs analoge Eingänge, 14 digitale I/Os und wird mit einer Spannung von 5V betrieben. Ein externes Gerät zum überspielen eines Programms auf den verbauten AVR ATmega328 Mikrocontroller ist nicht nötig. Auf dem Arduino UNO sind die dazu benötigten Teile verbaut. Der Arduino

UNO kann mit sogenannten *Shields* erweitert werden. Das sind Platinen die weitere Hardwarefunktionen bereitstellen. Es gibt *Shields* die verschieden Typen von Netzwerkadapter bereitstellen oder auch die Anzahl der I/Os erhöhen.

3.3.2 Arduino DUE

Der Arduino DUE ist ein Mikrocontrollerboard mit einer Atmel SAM3X8E ARM Cortex-M3 Prozessor. Das Board verfügt über zwölf analoge Eingänge und 54 digitale I/Os und hat eine Betriebsspannung von 3.3V. Weiter verfügt der Arduino DUE über zwei analoge Ausgänge und zwei CAN Bus Schnittstellen. Ein externes Gerät zum überspielen eines Programms Mikrocontroller ist nicht nötig. Der Arduino DUE kann, wie der Arduino UNO, mit *Shields* erweitert werden. Es ist aber darauf zu achten, dass die meisten erhältlichen *Shields* für eine Betriebsspannung von 5V ausgelegt sind. Eine Kompatibilität mit dem Arduino DUE ist nicht garantiert.

3.3.3 Teensy 3.2

Der Teensy 3.2 ist ein Mikrocontrollerboard mit einer MK20DX256VLH7 Cortex-M4 Prozessor. Die Prozessor basiert wie der SAM3X8E ARM Cortex-M3 des Arduino DUE auf einer ARM Architektur. Das Board verfügt über 34 digitale I/Os, 21 analogen Eingänge und einem analogen Ausgang. Eine CAN-Schnittstelle ist ebenfalls verbaut. Die Betriebsspannung liegt bei 3.3V. Im Gegensatz zu dem Arduino DUE sind die digitalen Eingänge 5V tolerant. Dadurch lassen sich Signale von Bauelemente die auf *Transistor-Transistor-Logik (TTL)*-Technik basieren, mit dem Teensy 3.2 auswerten. Programme für den Teensy 3.2 werden, wie die Produkte der Arduino Familien, mit der Arduino IDE entwickelt. Zum überspielen des entwickelten Programmcodes wird ein separates Programm benötigt. Eine Besonderheit des Teensy 3.2 ist seine geringe Baugröße von $35mm * 18mm$.

3.4 Einplatinencomputer

Ein Einplatinencomputer hat die zum Betrieb nötigen Komponenten (Prozessor, Arbeitsspeicher, Schnittstellen u.v.m.) auf einer Leiterplatte verbaut. Dadurch wird eine sehr kompakte Bauform erreicht. Die Abgrenzung zu einem Mikrocontroller besteht darin, dass die Komponenten auf einer Platine verteilt sind und das ein Betriebssystem auf diesen Geräten läuft. Dazu muss erwähnt werden das die Gren-

zen fließend sind. Der Entwicklungstrend von Einplatinencomputer geht in Richtung System-on-a-Chip. Dabei wird versucht einen immer größeren Teil der auf der Platine verbauten Komponenten auf einem einzigen Chip unterzubringen. Eine Auswahl von aktuellen Einplatinencomputer:

- AVM Fritz!Box ist eine Familie von Routern und Telefonanlagen. Die Prozessoren basieren auf der MIPS-Architektur. Das Betriebssystem auf GNU/Linux.
- Raspberry Pi ist eine Serie von Einplatinencomputer basierend auf Prozessoren mit der ARM-Architektur. Eine Besonderheit ist, dass für dieses Systeme die Soft-SPS des Hersteller 3S-Smart Software Solutions, CoDeSys, bereitgestellt wird.
- Cubieboard ist eine Serie von Einplatinencomputer dessen Spezifikation Open-Source sind. Als Besonderheit bieten die Boards einen SATA Anschluss zur Ansteuerung von Festplatten (ausgenommen ist da Cubieboard4, es verfügt dafür über einen USB 3.0 Anschluss).
- Beagle ist eine Familie von Einplatinencomputer dessen Hardware-Design frei verfügbar ist.

Eine Detaillierte Beschreibung einiger ausgewählter Einplatinencomputer, die Relevante Kenndaten zur Bearbeitung der Arbeit haben ist den den folgenden Abschnitten nachzulesen.

3.4.1 Raspberry PI 2

Der Raspberry PI 2 ist ein Einplatinencomputer mit einer ARM Cortex-A7 CPU. Er verfügt über vier Prozessorkerne und 1GB DDR2-SDRAM Arbeitsspeicher. Weiter sind vier Universal Serial Bus (USB) 2.0 Schnittstellen und eine 100Mbit Ethernet-Schnittstelle verbaut. Der Raspberry PI 2 verfügt über 40 General Purpose Input/Outputs (GPIOs). Als Betriebssystem stehen mehrere GNU/Linux Varianten (Debian, ArchLinux, Ubuntu), Windows 10 IOT Core und RISC OS zur Verfügung.

3.4.2 BeagleBone Black

Das BeagleBone Black ist ein Einplatinencomputer mit einer ARM Cortex-A7 CPU und 512MB DDR3 Arbeitsspeicher. Weiter bietet der BeagleBone Black 69 digitale I/Os, eine Ethernet-Schnittstelle, eine CAN-Schnittstelle und eine USB-Schnittstelle.

Als Betriebssystem stehen mehrere GNU/Linux Varianten (Angstrom und Debian), FreeBSD und Android zur Verfügung. Es existieren noch weitere inoffizielle Betriebssysteme für das BeagleBone Black.

Der Einplatinencomputer verfügt zusätzlich zur CPU noch über zwei verbaute Programmable Real-Time Units (PRUs). Diese können bei der Softwareentwicklung dafür genutzt werden um direkten Zugriff auf die Verbauten GPIOs mit einer geringen Zugriffszeit zu bekommen.

3.4.3 Texas Instruments ICE

Das Texas Instruments ICE hat eine ARM Cortex-A8 CPU und 512MB DDR3 Arbeitsspeicher. Es verfügt über zwei Ethernet-Schnittstellen und 96 digitale I/Os. Das Board unterstützt die gängigen Bussysteme. Dazu gehören: PROFIBUS, CANOpen, EtherCAT, PROFINET und SERCOS. Als Betriebssystem kommt *Real-Time Operating System* von Texas Instruments zum Einsatz.

Wie das BeagleBone Black verfügt das Texas Instruments ICE noch über zwei verbaute Programmable Real-Time Units.

4 Konzepte

In diesem Kapitel werden verschiedene Konzepte erarbeitet, wie vorhandene oder neue Steuerungslösungen im Rahmen von Industrie 4.0 umgesetzt werden können. Es werden dabei unterschiedliche Systeme betrachtet und daraufhin Möglichkeiten aufgezeigt, wie diese sich mit dem Fokus auf Industrie 4.0, mit aktuellen Techniken realisieren lassen.

4.1 Haptisches Interaktionsgerät Graspit

Das am Fraunhofer IPK entwickelte System Graspit, ist ein haptisches Interaktionsgerät mit sechs Freiheitsgraden. Graspit ermöglicht Greifinteraktion in einer virtuellen Umgebung. Um eine zwei Wege Mensch-Maschine-Interaktion zu realisieren, darf der Regelkreis nicht unter eine Frequenz von 1000Hz ausgeführt werden (siehe Unterabschnitt 2.2.1). Dieses würde vom Anwender als ein Vibrieren des Systems wahrgenommen werden und eine realitätsnahe Wahrnehmung wäre nicht mehr gegeben. Diese Voraussetzung wird vom Graspit-System erfüllt. Der Regelkreis des Interaktionsgeräts und die Visualisierung sind voneinander entkoppelt. Dadurch kann die Visualisierung mit einer Frequenz von 60Hz betrieben werden, während das Interaktionsgerät eine Regelfrequenz von $\geq 1000\text{Hz}$ hat. [21][22]

4.1.1 Aufbau des Graspit-Systems

Die Eingabehardware besteht aus zwei Phantom Premium 3.0 Geräten, die durch einen *Endeffektor* miteinander gekoppelt sind (Abb. 4.1). Der *Endeffektor* ist variable und kann je nach Anwendungsszenario angepasst werden. Weiter kann das Graspit-System auch mit nur einem Phantom Premium 3.0 betrieben werden. Die Anzahl der Freiheitsgrade halbiert sich dadurch. Die Software zum Koppeln der Eingabehardware mit einem virtuellen Objekt in einer Simulation ist nach dem Client-Server-Modell konzipiert. Der Server wertet die aktuellen Positionsdaten der Eingabehardware aus und sendet diese an den Client. Der Client wiederum ist ein

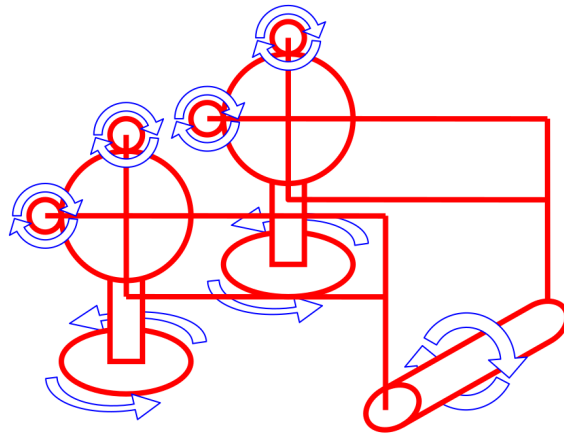


Abbildung 4.1: Das Graspit System[21, S. 11]

Teil der virtuellen Simulation und manipuliert die Position des gekoppelten virtuellen Objektes in Abhängigkeit der übertragenen Positionsdaten. Entstehen in der virtuellen Szene Kollisionen, werden die dabei berechneten Kräfte an den Server gesendet, der diese als Kraftrückkopplung auf dem Eingabegerät wiedergibt.



Abbildung 4.2: Phantom Premium 3.0 (im Vordergrund)

Aufbau der Hardware

Das Graspit-System ist ein PC basiertes haptisches Interaktionsgerät, das aus mehreren Komponenten besteht:

Phantom Premium 3.0 ist das haptische Eingabegerät der Firma Senseble, mit dem der Anwender interagiert. In Abhängigkeit von der Systemkonfiguration ist ein Phantom Premium 3.0 oder zwei, welche dann miteinander gekoppelt sind, an das System angeschlossen. An jedem der drei Gelenken eines Phantoms

sind ein Inkrementalgeber für die Positionserfassung und ein Motor für die Kraftrückkopplung verbaut. (Abb. 4.2)

ADDI-DATA APCI-1710 ist eine digitale Peripheral Component Interconnect (PCI) Multifunktionszählerkarte. Mit dieser Karte werden die verbauten Inkrementalgeber der Phantoms ausgelesen.

ADDI-DATA APCI-3120 ist eine analoge PCI I/O Karte zum Ansteuern der in den Phantoms verbauten Motoren.

Leistungsverstärker. Die sechs Leistungsverstärker verstärken das Steuersignal für jeden Motor. Die Anzahl ist abhängig von den angeschlossenen Phantoms. Die Leistungsverstärker sind in einem separaten Gehäuse untergebracht.

Die beiden I/O-Karten sind in einem PC zur Steuerung des Systems eingebaut. An die Multifunktionszählerkarte sind die Inkrementalgeber angeschlossen. Die Ausgänge der analogen I/O Karte sind mit den Leistungsverstärkern verbunden, dessen Ausgänge an die Motoren der Phantoms angeschlossen sind (Abb. 4.3).

Aufbau der Software

Die Software für das Graspit-System besteht aus drei Komponenten welche in der Programmiersprache C implementiert sind:

Graspit-Treiber. Der Gerätetreiber ist eine Eigenentwicklung des Fraunhofer IPK. Basierend auf den Gerätetreibern für die digitale und der analogen I/O Karte von ADDI-DATA, stellt der Treiber die benötigten Grundfunktion zum auslesen der Inkrementalgeber und zum Ansteuern der Motoren aus dem *Userspace* des Betriebssystems (hier GNU/Linux) zur Verfügung. Die Treiber setzen einen Linux-Kernel in der Version 2.6.x voraus.

Graspit-Server. Der Graspit Server hat mit Hilfe des Treibers Zugriff auf die an das System angeschlossenen Phantom Geräte. Aus den Daten der sechs Inkrementalgeber (bzw. drei bei nur einem angeschlossenen Phantom Gerät) wird die Position des *Endeffektors* bestimmt (direkte Kinematik). Diese Informationen werden über ein Netzwerk übertragen und an den Client gesendet. Das verwendete Übertragungsprotokoll ist das User Datagram Protocol (UDP). Vom Server werden Kräfte empfangen, die der Client aus der laufenden Simulation sendet. Diese werden mit der inversen Kinematik transformiert und durch Ansteuerung der Motoren wiedergegeben.

Graspit-Client. Der Client wird in die Simulationssoftware integriert. In der Simulation wird ein virtuelles Objekt mit den Positionsdaten des *Endeffektors* verknüpft. Kollidieren beim Bewegen des *Endeffektors* zwei virtuelle Objekte, werden von der Simulation die auftretenden Kräfte berechnet. Diese werden an den Graspit Server gesendet.

4.1.2 Schematischer Aufbau des Graspit Systems

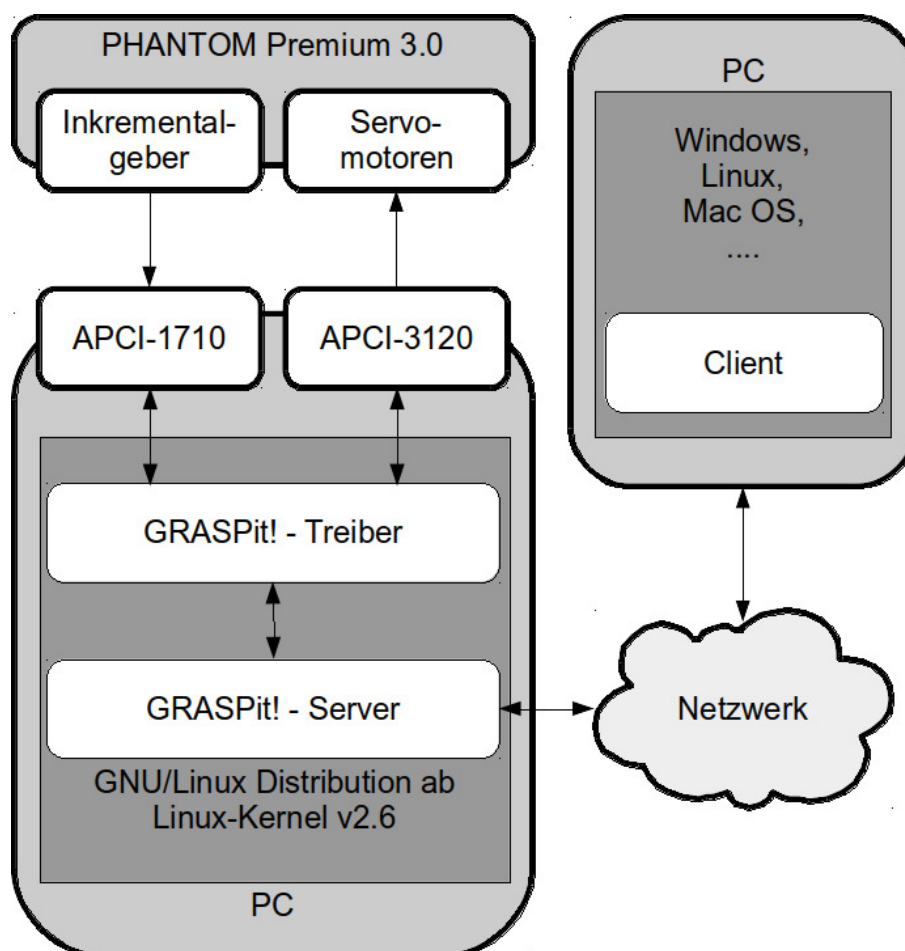


Abbildung 4.3: Das Graspit System - vereinfachte schematische Darstellung

4.1.3 Analyse des bestehenden Systems

Das bestehende Graspit System hat nach heutigem Stand der Technik an vielen Stellen Potential zur Optimierung. Die von der Firma ADDI-DATA verwendeten I/O-Karten nutzen in dem System einen eigens für diesen Anwendungsfall entwickelten Linux Gerätetreiber. Ein Problem darin besteht, dass Linux kein festes Treiber Application Programming Interface (API) und Application Binary Interface (ABI) hat. Linux garantiert eine feste API und ABI nur zwischen *Userspace* Anwendungen und Linux. [23]

Bei jedem Update von Linux muss geprüft werden, ob der entwickelte Treiber noch lauffähig ist. Gegebenenfalls muss dieser angepasst werden. Dadurch werden Updates evtl. aufgeschoben. Bei vernetzten Geräten, die möglicherweise über ein offenes Netzwerk kommunizieren, stellt dies ein erhebliches Sicherheitsrisiko dar.

Weiter ist die Modularität dieses Konzepts nur in Teilen gegeben, wodurch die Ersetzbarkeit einzelner Komponenten oder das Erweitern des System erschwert wird: Die Steuerung, die I/O-Karte sind ein starres, wenig flexibles PC-System. Eine einfache Erweiterung des Systems mit neuen Aktoren oder Sensoren ist mit einem hohem Aufwand verbunden. Der Gerätetreiber für die I/O-Karten müsste angepasst werden, sofern die neuen Sensoren und Aktoren überhaupt diese Karten unterstützt würden. Durch eine Änderung des Treiberschnittstelle muss wiederum der Graspit-Server an die neue Treiberschnittstelle angepasst werden.

Für eine Verbesserung des Systems muss eine Lösung entwickelt werden, welche die Abhängigkeit von dem verwendeten Treiber löst und die Modularisierung des Systems fördert.

4.1.4 Entwurf eines neuen Konzepts

Anforderungen

Bei der Entwicklung eines neuen Konzeptes für das Graspit-System müssen folgende Anforderungen an das Systems berücksichtigt werden:

1. Grundvoraussetzung für ein haptisches Interaktionsgerät ist es, dass eine Mindestfrequenz von $1000Hz$ für den Regelkreislauf der Steuerung eingehalten wird (siehe Unterabschnitt 2.2.1).
2. Die Modularität des Systems sollte erhöht werden, damit das Gerät besser an spezifische Einsatzszenarien angepasst werden kann.

3. Die Wartbarkeit und Pflege des Systems muss vereinfacht werden.

Mögliche Technologien

Um die aufgestellten Anforderungen zu erfüllen, bieten sich ein Technologiewechsel an. Von der zuvor individuellen Lösung hin zu einem neuen System, welches auf aktueller Steuerungstechnik basiert. Das System kann durch den Einsatz von *Busklemmen*, einem Feldbus zur Kommunikation und der Verwendung einer SPS weiter modularisiert werden. Das Bussystem verbindet die *Busklemmen* und die SPS miteinander. Die SPS steuert das einlesen der Inkrementalgeber und die Ausgabe der Steuerströme an die Motoren. Der Client für die Simulationssoftware wird über eine Ethernet fähigen *Busklemme* mit der Steuerung verbunden (Abb. 4.4).

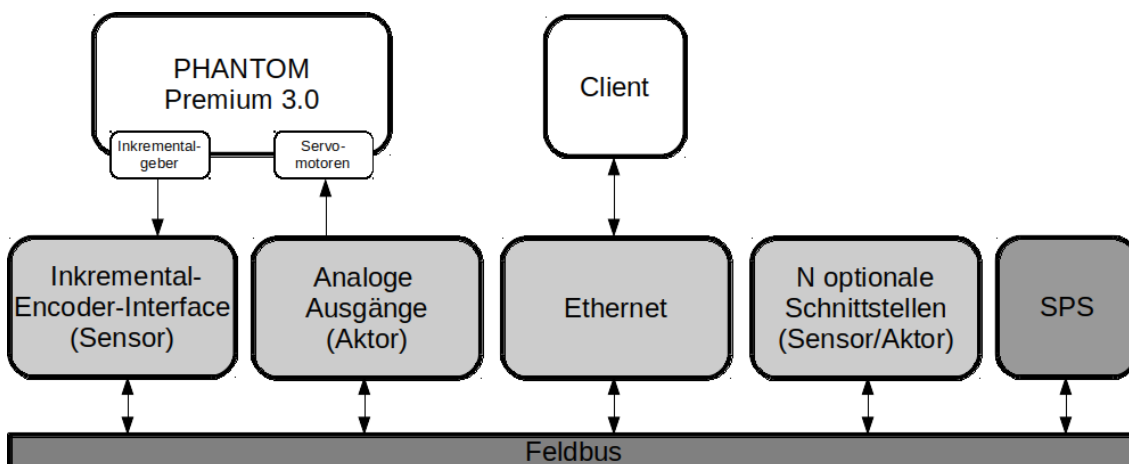


Abbildung 4.4: Schematische Darstellung des neuen Konzeptes für das Graspit-System

Der zu verwendende Feldbus und die SPS müssen die in den Anforderungen bestimmte Regelfrequenz von mindestens 1000Hz einhalten können. Die Gesamtzykluszeit bestehend aus Zeit, die der Feldbus braucht, um die Daten der Busteilnehmer zu lesen oder zu schreiben und der Verarbeitung der SPS darf maximal 1ms ($= 1000\text{Hz}$) betragen. Dabei ist zu beachten, dass die Zykluszeit der SPS abhängig von der Komplexität des ausführenden Programms ist. Tabelle 4.1 verschafft einen Überblick, der im Kapitel Stand der Technik vorgestellten Netzwerktechnologien.

	CANopen	Interbus	Profibus DP	EtherCAT
Max. Teilnehmer pro Bus	127	256	126	65535
Max. Ausdehnung je Segment	100m ¹	100m ²	100m ³	100m
Min. Zykluszeit	1 – 10ms	< 5ms	2ms ⁴	50μs ⁵

Tabelle 4.1: Bussysteme im Vergleich

Aus der Tabelle 4.1 kann entnommen werden, dass für die Umsetzung EtherCAT die aufgestellte Anforderung an die Zykluszeit erfüllt. Die SPS kann als Soft-SPS realisiert werden. Der Vorteil besteht darin, dass die Zykluszeiten für den auszuführenden Programmtask abhängig von der Rechenleistung des verwendeten PCs ist. Beckhoff TwinCat und Codesys Engineering geben als minimale Zykluszeit 50μs an [25]. Eine Aussage über die Komplexität der Steuerung bei der angegebenen minimalen Zykluszeit ist nicht vorhanden. Die angestrebte Minimalfrequenz von 1000Hz kann damit rechnerisch erreicht werden. Sollte die Rechenleistung nicht ausreichen um die Zykluszeiten einzuhalten, kann die Soft-SPS auf leistungsfähigerer Hardware ausgeführt werden.

4.2 Haptikzelle

Die Haptikzelle ist ein sich am Fraunhofer IPK in Entwicklung befindendes haptisches Interaktionsgerät mit drei Freiheitsgraden. Das Gerät soll eine passive und auch aktive Interaktion ermöglichen (siehe Unterabschnitt 2.2.2). Das Gerät ist dafür ausgelegt, eine maximale Kraft von 433N am Endeffektor aufzubringen. Die Haptikzelle hat einen Arbeitsraum von $0.66 * 0.7 * 0.72m^3$ (Abb. 4.5). Mit der Haptikzelle soll es ermöglicht werden das Heben und Verbauen von schweren Komponenten in der Fertigung zu simulieren.

4.2.1 Systemaufbau

Die Haptikzelle hat die geometrische Form eines Würfels. Die zum Anwender liegende Seite ist offen. Von der geöffneten Seite aus kann der Benutzer einen Endeffektor

¹100 m (500 kbit/s), 1000 m (62,5 kbit/s)

²zwischen 2 Geräten max. 400 m

³100 m (12 Mbit/s), 1200 m (9,6 kbit/s)

⁴Mit fünf Busteilnehmern und 5 Byte ad Daten im zyklischen Betrieb DP-V0. [24, S. 63]

⁵Mit analogen Busteilnehmern. Bei rein digitalen Busteilnehmern min. 10μs.

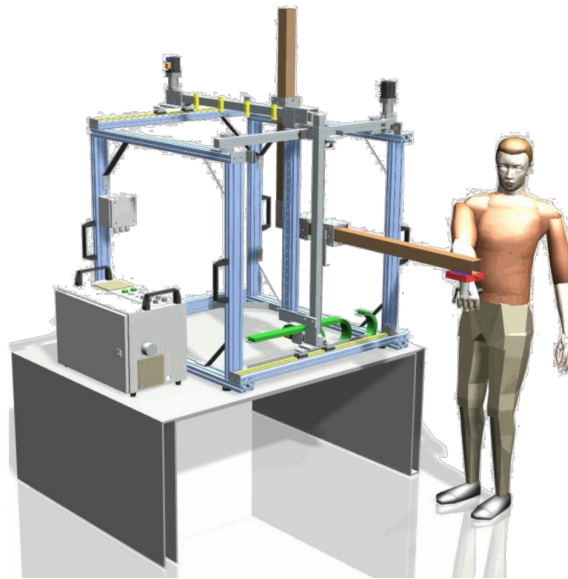


Abbildung 4.5: Haptikzelle Quelle: Faunhofer IPK

bedienen. Der Endeffektor ist ein Hebel, welcher über einen Sensor mit der Haptikzelle verbunden ist. Der Sensor besteht aus drei *Dehnungsmessstreifen*. Jeder *Dehnungsmessstreifen* misst die Bewegung einer Richtung, die auf den Hebel ausgeübt wird (x, y, z). Drei Motoren erlauben das Verfahren des Endeffektors. Zur Positionsbestimmung ist für jede Achse ein Inkrementalgeber verbaut. An den Anschlüssen der Achsen sind Sensoren zur Bestimmung des Endanschlags einer Achse montiert.

4.2.2 Zielsetzung

Für die Haptikzelle ist eine Steuerung zu entwickeln, um die gewünschte passive und aktive Interaktion zu realisieren. Das Gerät soll darüber hinaus mit seinem „Digital Twin“ (siehe Abschnitt 2.2) gekoppelt werden.

4.2.3 Hardware

Die Hardware besteht aus drei Servoverstärkern für den Betrieb der Achsen und drei Busklemmen zum einlesen der Messwerte der *Dehnungsmessstreifen*. Eine Busklemme mit sechs digitalen Eingängen zur Bestimmung der Endanschläge ist ebenfalls verbaut. Die Komponenten sind durch einen EtherCAT-Bus miteinander vernetzt.

4.2.4 Entwurf eines neuen Konzepts

Anforderungen

Für eine mögliche Umsetzung der Zielsetzung müssen folgende Anforderungen eingehalten werden:

1. Die Hardware bei diesem System ist bereits vorgegeben, kann aber durch neue Komponenten, die in das Bussystem integriert werden, erweitert werden.
2. Eine Steuerung für aktiv und passive Interaktion ist zu entwickeln.
3. Die Randbedingung einer Regelfrequenz von $1000Hz$ muss für das System eingehalten werden (siehe Unterabschnitt 2.2.1)
4. Das System soll mit einem „Digital Twin“ gekoppelt werden.
5. Aufgrund der Größe des Systems, muss eine durch den Benutzer veranlasste Bewegung des Endeffektors durch die Motoren unterstützt werden.

Konzept

Um die gestellten Anforderungen zu erfüllen, bietet sich die Verwendung einer Soft-SPS zur Realisierung der Steuerung der vorhandenen Hardwarekomponenten an. Die Tabelle 4.1 zeigt auf, dass der verwendete Feldbus die Bedingungen einer Regelfrequenz von $1000Hz$ erfüllen kann. Die zur Auswahl stehenden Softwaresteuerungen Beckhoff TwinCat und CoDeSys haben eine minimale Prozesszykluszeit von $50\mu s$ an. [25] Für die Kopplung mit einem „Digital Twin“ bietet sich das gleiche Systemschema wie für das Graspit-System an (siehe Abbildung 4.4). Der Unterschied besteht nur in dem verwendeten Eingabegerät.

4.3 Bi-Manu-Track

Der Bi-Manu-Track ist ein von der Firma Reha-Stim vertriebenes Rehabilitationsgerät. Mit dem Bi-Manu-Track können Patienten, die einen Schlaganfall erleiden mussten, drei isolierte Bewegungen der Hand und des Arms zur Steigerungen der Motorik üben (Abb. 4.6):

Pro- und Supination des Unterarms. Die Supination ist die Auswärtsdrehung des Handgelenks durch Rotation des Unterarms. Die Pronation ist die Gegenposition zur Supination.

Flexion und Extension des Handgelenks. Die Flexion des Handgelenks ist die Beugung des Gelenks. Die Extension ist die Streckung des Handgelenks.

Flexion und Extension des Fingergrundgelenke. Die Flexion des Fingergrundgelenke ist die Beugung. Extension ist die Streckung.



Abbildung 4.6: Bi-Manu-Track Pronation und Supination Quelle: <http://www.reha-stim.de/>

Die Übungen erfolgen unter der Aufsicht eines Physiotherapeuten. Dieser wählt für den Patienten eine passende, dem Krankheitsbild entsprechende, Übung aus. Unter Beaufsichtigung führt der Patient sein Übungsprogramm aus. Mit dem Bi-Manu-Track ist eine vielfache Wiederholung von isolierten Übungen möglich. Konsequenterweise angewandt führen Übungen mit dem Bi-Manu-Track zu einer Verbesserung der motorischen Leistung des betroffenen Patienten. [26] [27]

4.3.1 Zielsetzung

Im Rahmen des regionalen Innovationsclusters BeMobil mit dem Förderschwerpunkt Mensch-Technik-Interaktion soll das Gerät um eine Netzwerkschnittstelle erweitert werden. Die Anforderungen sind, dass Daten, die bei einer Übung vom Bi-Manu-Track durch die Sensorik aufgenommen werden extern ausgewertet werden können. Weiter soll es ermöglicht werden, dass zwei Bi-Manu-Tracks miteinander verbunden werden. Ziel ist es dem Patienten zu ermöglichen, sofern die Schwere der Bewegungseinschränkung es zulässt, die Übungen auch von zu Hause aus durchzuführen. Die Übungen werden aus der Ferne weiter von einem Therapeuten begleitet.

BeMobil

Im Zentrum des Vorhabens steht die Unterstützung motorisch eingeschränkter Menschen beim Wiedererlangen der eigenständigen Bewegungsfähigkeit und Alltagsmobilität. Adaptive technische Systeme und neue therapeutische Behandlungsansätze werden gezielt auf die Bedürfnisse des Betroffenen ausgerichtet, um die Bewegungsfähigkeit zu Hause und im Beruf wieder zu ermöglichen. Ziel ist die Entwicklung neuartiger adaptiver Hilfsmittel und Therapieübungssysteme wie intelligente Prothesen, Orthesen (Korsett-Therapie), Reha-Roboter und Virtual-Reality-Systeme, die auf der direkten physischen Interaktion zwischen Mensch und Technik basieren. [28]

4.3.2 Systemaufbau

Der Bi-Manu-Track ist ein eingebettetes System. Für den Therapeuten steht ein Display und eine Tastatur zur Verfügung, um das gewünschte Übungsprogramm auszuwählen. Der Patient hält wie in Abbildung 4.6 die beiden Endeffektoren. Sollte dieser dazu körperlich nicht in der Lage sein, wird er an diesen festgeschnallt. An den beiden Endeffektoren sind jeweils ein Motor angebracht. Im Inneren des Gerätes befindet sich eine Platine der Firma Tigris. Sie wurde speziell für dieses Gerät entwickelt. Die Platine besteht im Kern aus einem *Field Programmable Gate Array (FPGA)*, einem Mikrocontroller und zwei Endstufe, um die beiden Motoren zu betreiben. Weiter sind Sensoren zur Messung der Ströme, der anliegenden Spannungen und der Inkrementalgeber verbaut. [29]

Der Mikrocontroller DS80C320 von Maxim übernimmt die Steuerung des Bi-Manu-Tracks. Dazu zählen die verschiedenen Übungsprogramme für die Patienten,

die Regelung der Motoren sowie die Ansteuerung des FPGAs. Für die Ansteuerung des FPGAs sind unveränderliche Randbedingungen einzuhalten. Diese resultieren aus der Implementierung des FPGAs. Der verbaute FPGA und der Mikrocontroller sind durch einen 8 Bit breiten Adress- und Datenbus miteinander verbunden. Weiter kann der Betriebsmodus des Busses mit weiteren drei Steuerleitungen eingestellt werden. Die Betriebsmodi sind Read, Write und Ale. Read und Write wechseln zwischen Lese- und Schreibzugriff auf den gemeinsamen Bus. Ale signalisiert dem FPGA, dass eine valide Adresse auf dem Bus geschrieben wurde. Der verbaute Mikrocontroller hat folgende technische Eigenschaften:

- 8 I/O Ports. Die 8 I/O Ports können zur Laufzeit vom Betriebsmodus Eingang zum Betriebsmodus Ausgang und vice versa konfiguriert werden. Dies ist nötig, da der Adress- und Datenbus die selben acht Leitungen lesend und schreibend benutzt.
- 3 Ausgänge für die Steuerleitungen Read Write und Ale.
- Ein setzen der Ausgänge erfolgt parallel.
- Für die Ansteuerung der im FPGA verfügbaren Funktionsblöcke Positionserfassung für die Inkrementalgeber, PWM-Signalgenerierung für die Endstufe und das Messen der am Motor anliegenden Kraft benötigt der Mikrocontroller die Zeit von $20\mu s$ ($= 20kHz$).

Der *Field Programmable Gate Array (FPGA)* EPF10K10LC84 von Altera hat mehrere Funktionsblöcke für die Ansteuerung der angeschlossenen Hardwarekomponenten. Diese werden über den gemeinsamen Adress- und Datenbus von Mikrocontroller und FPGA angesprochen. Das Programm des FPGAs wird nach dem Einschalten des Bi-Manu-Tracks aus einem verbauten EEPROM geladen. Folgende Funktionsblöcke stellt der FPGA dem Mikrocontroller bereit:

- PWM-Signalgenerierung für Endstufe 1
- PWM-Signalgenerierung für Endstufe 2
- Positionserfassung für Inkrementalgeber 1
- Positionserfassung für Inkrementalgeber 2
- Zugriff auf den Analog -/ Digitalwandler (auswerten der Motorströme)

- Ausgangsport für Tastaturabfrage
- Ausgangsport für Speicherbankumschaltung

Ansteuern der Funktionsblöcke

Um mit dem verbauten Mikrocontroller einen Funktionsblock anzusteuern, wird die Adresse des anzusteuernenden Funktionsblocks auf den Bus geschrieben und das Signal Write gesetzt. Mit dem Signal Ale wird dem FPGA signalisiert, dass eine valide Adresse auf dem Bus liegt. Der FPGA lädt den an der angeforderten Adresse hinterlegten Wert aus dem Datenregister und schreibt diesen auf den Bus sobald das Signal Read gesetzt wird. Der Mikrocontroller kann den hinterlegten Wert daraufhin vom Bus lesen und weiterverarbeiten.

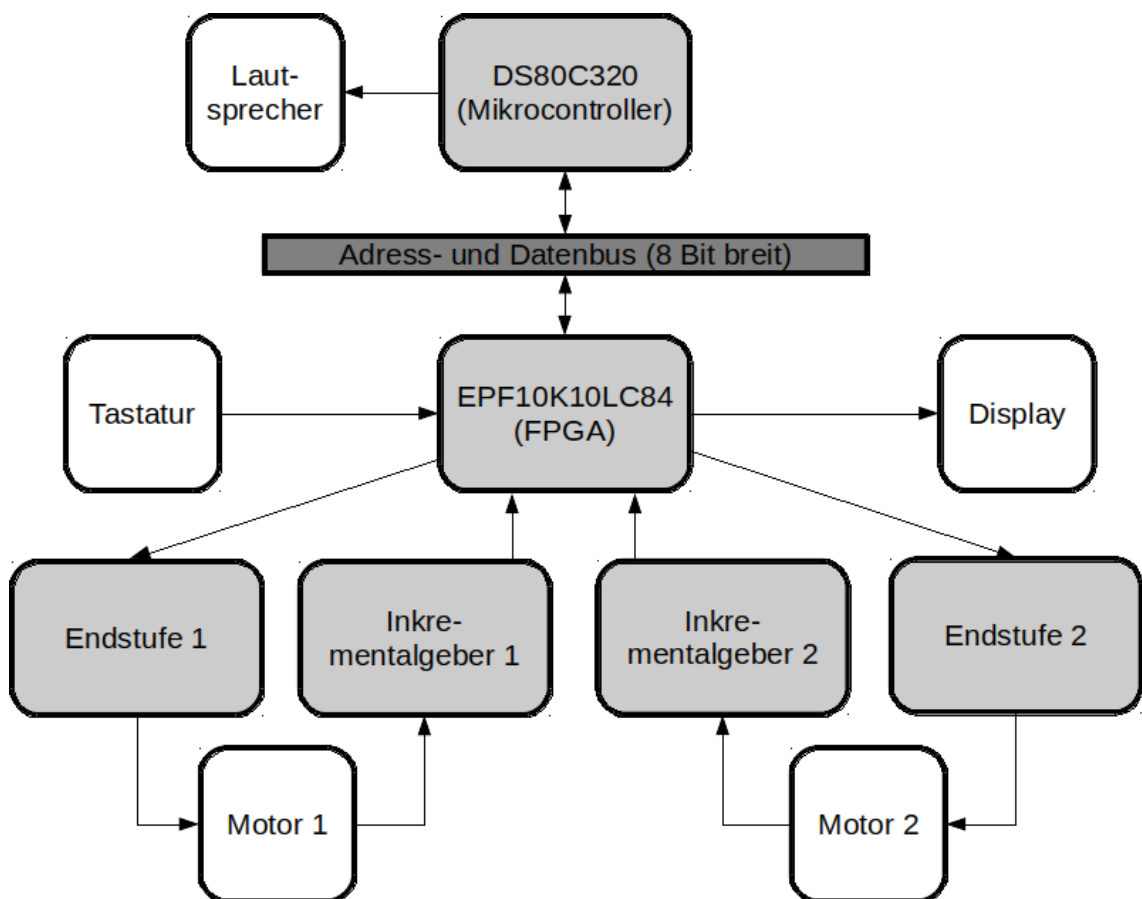


Abbildung 4.7: Schematische Darstellung Systemaufbau Bi-Manu-Track

4.3.3 Entwurf eines neuen Konzepts

Anforderungen

Damit der Bi-Manu-Track mit einer Netzwerkschnittstelle nachgerüstet werden kann, müssen mehrere Punkte beachtet werden:

1. Eine komplette Neugestaltung der Hardware ist in diesem Fall nicht möglich. Für bestehende Systeme soll die Möglichkeit geschaffen werden, die neuen Funktionen nachzurüsten.
2. Der Bi-Manu-Track ist ein Gerät welches für den Betrieb im klinischen Umfeld zugelassen ist. Für diese Zulassung muss das Gerät eine EMV-Prüfung bestehen. Um ein Bestehen mit wenigen Anläufen zu gewährleisten, sollte die Veränderungen zum bestehenden System möglichst gering ausfallen.
3. Die Ansteuerung der Funktionsblöcke (Inkrementalgeber, Motorstrom und PWM-Signalgenerierung) mit einer Zeit von maximal $20\mu s$ muss eingehalten werden. Bei Unterschreitung besteht die Möglichkeit, dass die Regelung der Motoren zu träge wird. Dies kann zu ruckhaften Bewegungen der Endeffektoren führen oder auch zu einem Nachschleppen der Endeffektoren bei Ausübung einer Kraft auf diese in Bewegungsrichtung.
4. Für die Ansteuerung des Adress- und Datenbusses müssen die I/Os der Steuerung im Betrieb von Eingang auf Ausgang und vice versa konfiguriert werden können.
5. Die Daten müssen parallel auf den Adress- und Datenbus geschrieben werden können. Eine kaskadierte Ausgabe von Daten auf den Bus führt zu einem nicht erwünschten Bitmuster und kann zu einem Fehlverhalten des Systems führen.
6. Die Schaltungstechnik der Platine ist *Transistor-Transistor-Logik (TTL)*-Technik und mit einer Betriebsspannung von 5V ausgeführt. Für den Fall der Verwendung eines Mikrocontrollers auf *Complementary metal-oxide-semiconductor*-Technik muss eine Lösung für die unterschiedlichen Signalpegel gefunden werden.

Konzept

Um den aufgestellten Anforderungen gerecht zu werden bieten sich zwei Lösungsansätze an. Es kann ein Netzwerkadapter auf der Systemplatine integriert werden.

Dafür muss das Programm des FPGAs mit einer neuen Funktion zur Ansteuerung angepasst werden. Als zweite Option bietet sich an, den im Bi-Manu-Track verbauten Mikrocontroller zu entfernen und den Sockel als Schnittstelle zu dem System zu benutzen. Die Ansteuerung des FPGAs mit einem Mikrocontroller ist dokumentiert, die Ansteuerung der verbauten Hardware durch den FPGA ist nicht dokumentiert. Durch diesen Umstand wird vom ersten Lösungsansatz Abstand genommen. [29]

Die Verbindung von Mikrocontroller und FPGA erfolgen über elf Leitungen. Acht sind, wie zuvor erwähnt, für das Schreiben und Lesen des Adress- und Datenbusses nötig. Zwei weitere dienen zum Umschalten zwischen Lese- und Schreibzugriff auf den gemeinsamen Bus. Zusätzlich ist die Leitung Ale, ein Strobe Signal für die Übernahme einer validen Adresse vom Bus in den FPGA. Für das Einschalten eines verbauten Lautsprechers ist eine weitere Leitung vorhanden. Der Lautsprecher kennt nur die Zustände eingeschaltet oder ausgeschaltet. Die restlichen Kontakte des Sockels auf der Systemplatine sind nicht belegt.

Anstelle des vorhandenen Mikrocontrollers kommen mehrere Möglichkeiten in Betracht, wie dieser ersetzt werden kann. Diese müssen die Steuerung des Bi-Manu-Tracks übernehmen und das Gerät um eine Netzwerkschnittstelle erweitern. Folgende Konzepte sind realisierbar:

1. Ein Adapter für den Sockel wird konstruiert. Dieser Adapter führt den gemeinsamen Adress- und Datenbus, die drei Steuerleitungen und die Leitungen für den Lautsprecher heraus. Die Leitungen werden auf eine parallele Schnittstelle eines PCs geführt, die im *Enhanced Parallel Port* Modus läuft. Die Steuerung wird von einem PC ausgeführt. Als Betriebssystem kann eine Linux Distribution eingesetzt werden. Damit die Ansprüche für eine Regelung erfüllt werden können, empfiehlt sich GNU/Linux mit der Erweiterung *Real Time Application Interface* zu nutzen. Die Ethernet-Schnittstelle des PC würde zur Kopplung zweier Bi-Manu-Tracks und zum Weitergeben der erfassten Sensorwerte dienen.
2. Eine weitere Option besteht darin, einen Einplatinencomputer oder Mikrocontroller mit genügend GPIOs mit dem Sockel auf der Platine zu verbinden. Dafür bieten sich folgende Geräte an:
 - Der BeagleBone Black hat zwei PRUs, die eine zeitkritische Ansteuerung der I/Os erlaubt. Weiter verfügt er über eine Ethernet-Schnittstelle.
 - Arduino UNO, Arduino DUE und der Teensy 3.2 könnten Aufgrund ihrer

kompakten Bauform mit einem zu entwickelnden Adapter direkt auf den Sockel gesteckt werden. Netzwerkfunktionalität ist bei allen, bis auf den Arduino UNO, vorhanden. Diese kann aber mit einem *Shield* nachgerüstet werden.

- Der Raspberry PI 2 verfügt über eine Ethernet-Schnittstelle und genügend I/Os
- Das Texas Instruments ICE läuft mit einem Echtzeitbetriebssystem und hat wie das BeagleBone Black zwei PRUs. Zudem verfügt das Texas Instruments ICE über eine Implementierung EtherCAT, Ethernet und weiteren Netzwerktechniken.

Für alle Konzepte muss ein Adapter entwickelt werden, um die Steuerung mit dem Bi-Manu-Track zu verbinden. Weiter muss bei den vorgestellten Konzepten zur Realisierung der Zielsetzung evaluiert werden, damit die aufgestellten Rahmenbedingungen eingehalten werden können.

5 Umsetzung

Im letzten Kapitel wurden verschiedenen Konzepte erarbeitet, wie bereits vorhandene Steuerungen oder auch neue Steuerungen mit den Anforderungen an Industrie 4.0 realisiert werden können. Von diesen Konzepten wurden zwei ausgewählt und diese umgesetzt. Ausgewählt für eine Umsetzung wurden das Graspit-System und der Bi-Manu-Track.

5.1 Graspit-System

Das im vorherigen Kapitel entwickelte Konzept für das Graspit-System wird mit ausgewählten Soft- und Hardwarekomponenten umgesetzt, welche in diesem Abschnitt erläutert werden.

5.1.1 Hardware

Für den Aufbau wird ein Raspberry PI 2 als Zielplattform für die Soft-SPS verwendet. Der Raspberry PI 2 verfügt über eine Ethernet Schnittstellen und über einen Universal Serial Bus (USB), an den eine weitere Ethernet-Schnittstelle angeschlossen wird. Die bereits vorhandene Ethernet-Schnittstelle wurde für den Betrieb als EtherCAT-Master konfiguriert. An die EtherCAT-Master-Schnittstelle wird ein EtherCAT Buskoppler angeschlossen. Dieser dient als Verbindung zu den EtherCAT-Busklemmen. Die EtherCAT-Busklemmen bestehen aus einer analogen Ausgangsklemme (Beckhoff EL4038) und einem digitalem Inkremental-Encoder-Interface (Beckhoff EL5101). An die EtherCAT-Busklemmen wird der Leistungsverstärker und an diesen der Phantom Premium 3.0 angeschlossen. Die Ethernet-Schnittstelle dient zur Kommunikation mit dem Client. (Abb. 5.1 ohne Leistungsverstärker).

Raspberry PI 2 wurde ausgewählt, da CoDeSys dieses System als Zielplattform anbietet. Dadurch lässt sich der Raspberry PI 2 als Soft-SPS verwenden.

Beckhoff EK1100. „Der Koppler EK1100 verbindet EtherCAT mit den EtherCAT-Klemmen (ELxxxx). Eine Station besteht aus einem Koppler EK1100, einer beliebigen Anzahl von EtherCAT-Klemmen und einer Busendklemme.“[30] Die Busendklemme terminiert den Feldbus. Weiter hat sie keine Funktion.

Beckhoff EL5101. Die EtherCAT-Klemme EL5101 ist ein Interface zum direkten Anschluss von Inkremental-Enkodern. Für je einen an dem Phantom verbauten Inkrementalgeber wird eine EL5101 benötigt. Es können vier Millionen Inkremente in der Sekunde ausgewertet werden.

Beckhoff EL4038. Die analogen Ausgangsklemme EL4038 erzeugt Signale im Bereich von -10 bis +10V, mit einer Auflösung von 12Bit. Die Wandlungszeit des Signals beträgt ca. 400 μ s.

Maxon ADS 4-Q-DC. Der Maxon ADS 4-Q-DC ist der genutzte Leistungsverstärker. Die Ausgabe der Beckhoff EL4038 wird mit diesem verstärkt.

Spannungsversorgung Die Spannungsversorgung der Komponenten erfordert unterschiedliche Netzgeräte. Für den Leistungsverstärker wird ein 5V Netzteil mit einem max. Ausgangsstrom von 20A verwendet. Die Busklemmen werden mit einem 24V/2A Netzteil, und der Raspberry PI mit einem 5V/2A Netzteil versorgt.

Die Komponenten wurden in einem Schaltschrank verbaut.

Schematischer Aufbau

5.1.2 Software

Die Entwicklung der Steuerung wurde mit dem CoDeSys Development System durchgeführt. Als Programmiersprache wurde *Strukturierter Text* ausgewählt. Beim Programmieren von Steuerungen ist zu beachten, dass der Programmablauf zyklisch ausgeführt wird. Physische Eingänge werden zu Beginn eingelesen, die programmierte Logik des Programms wird ausgeführt und am Ende des Programms werden berechnete Werte an den physischen Ausgängen gesetzt. Daraufhin beginnt der Programmzyklus erneut. Es ist nicht möglich während eines Zyklus einen Ausgang mehrmals mit unterschiedlichen Werten zu setzen. Nur der letzte Wert wird gesetzt, die zuvor gesetzten Werte gehen verloren. Die CoDeSys Runtime wurde auf dem Raspberry PI 2 unter Debian 7 installiert. Debian 7 wurde leicht angepasst, indem

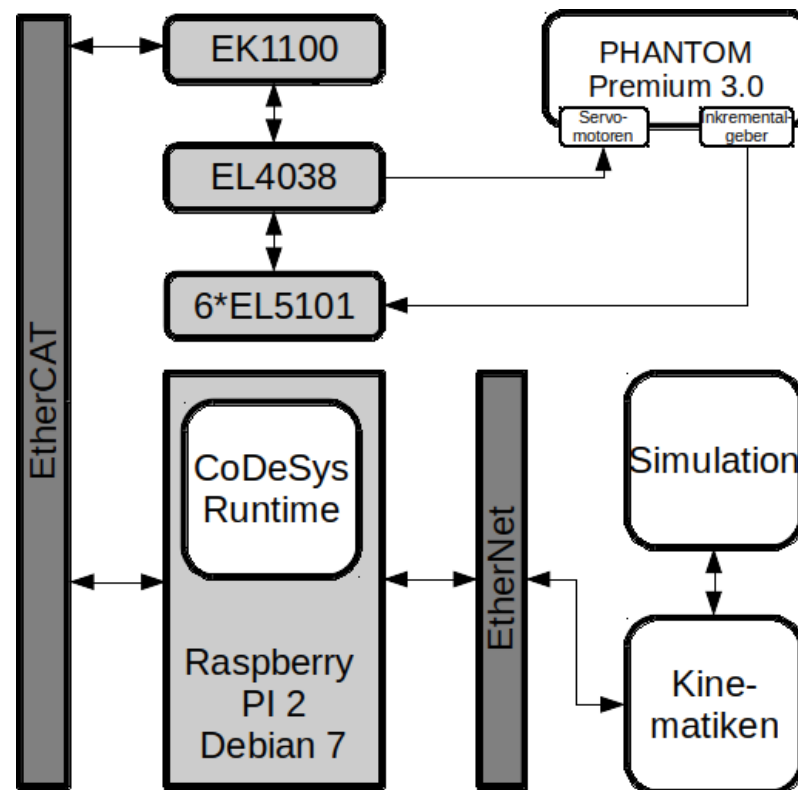


Abbildung 5.1: Das neue Graspit-System - schematische Darstellung

alle nicht benötigten Pakete entfernt wurden. Dazu zählen die grafische Oberfläche bis hin zu vorinstallierten Server.

Die Kinematiken des Graspit-Systems wurden in ein separates Softwaremodul ausgelagert und sind nicht mehr Teil der Steuerung. Sie werden per UDP mit dem System verbunden. Dies entspricht dem im vorherigen Kapitel vorgestellten Graspit-Server, ohne Hardwareansteuerung. Durch die Umgestaltung der Softwarearchitektur wurde die Modularität auch softwareseitig erhöht.

5.1.3 Steuerung

Durch die Modularisierung der Hardware und Auslagerung der Kinematiken ist eine sehr einfache Steuerung entstanden. Zu Beginn des Programmablaufs wird das EtherCAT-E/A-Abbild ausgewertet. Das EtherCAT-E/A Abbild beinhaltet alle Werte, die an den EtherCAT Busklemmen anliegen, gerade ausgegeben werden und alle weiteren Statusinformation die zur Verfügung stehen. Zu den Statusinformation gehören Information die den Zustand des Busses und der Busklemmen beinhalten. Im nächsten Schritt wird die Funktion *init* aufgerufen. Diese initialisiert im ersten

Durchlauf des Programms einen UDP - Socket und die Startwerte der Inkrementalgeber. Ab dem zweiten Durchlauf werden die Werte an den Inkremental-Enkodern aus dem EtherCAT-E/A-Abbild ausgelesen und mit der Funktion *sendUDP* an die Kinematik geschickt. Die empfangenen Kräfte werden daraufhin in das EtherCAT E/A Abbild geschrieben und an den Busklemmen ausgegeben. Die Zykluszeit für den SPS Prozess wurde auf eine $1ms (= 1kHz)$ festgesetzt. Die Einstellung der Zykluszeit wird in CoDeSys Engineering vorgenommen.

Programmablaufplan

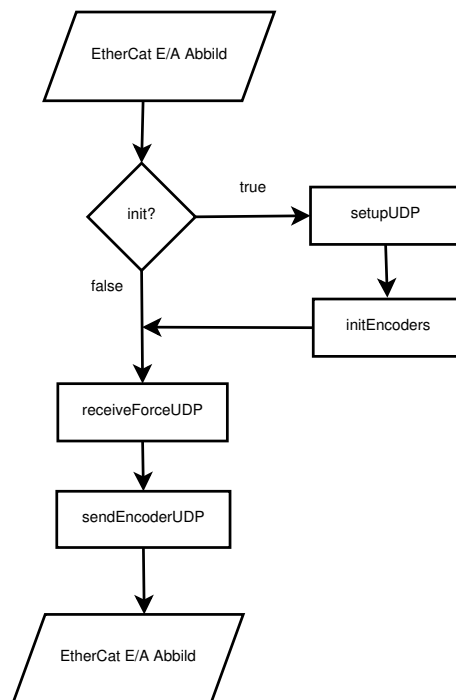


Abbildung 5.2: Programmablaufplan SPS

5.2 Bi-Manu-Track

Im Kapitel Konzepte wurden mehrere Optionen vorgestellt, die es ermöglichen die erforderliche Umrüstung des Bi-Manu-Tracks vorzunehmen. Bevor es an die eigentliche Umsetzung geht, muss evaluiert werden, ob sich die Konzepte umsetzen lassen.

5.2.1 Evaluation

Die zur Auswahl stehenden Plattformen werden daraufhin geprüft ob sie folgende Kriterien erfüllt:

1. Das parallel Setzen von acht Ausgänge zur Ansteuerung des Adress- und Datenbusses.
2. Das Einhalten der Regelfrequenz von mindestens $20kHz$.
3. Die Schaltfrequenz der Ausgänge liegt im hochfrequenten¹ Bereich. Es ist darauf zu achten, dass kein *Jitter* auftritt. *Jitter* führt zu einem verstärkten Symbolübersprechen welches zu einem fehlerhaften Bitmuster bei Schreibzugriffen auf den Adress- und Datenbus führen kann.

Testfeld

Für die Evaluation der verschiedenen Plattformen wurde ein Testfeld aufgebaut. Zwei an die zu untersuchende Plattform angepasstes Testprogramm wurde geschrieben und mit Hilfe eines Oszilloskops wurden die Messungen vorgenommen.

- Das erste Testprogramm konfiguriert zwei I/Os als Ausgänge und schreibt in einer Schleife die Bitmuster 11 gefolgt von 00 (Abb. 5.3). Ein Oszilloskop wurde an zwei Ausgänge der jeweiligen Plattform angeschlossen. Mit dem Oszilloskop wurde überprüft, ob die Ausgänge parallel oder nacheinander gesetzt werden. Das Ergebnis lässt sich anhand der Phasenverschiebung der gemessenen Signale zueinander vom Oszilloskop ablesen (Abb. 5.5).
- Das zweite Testprogramm führt einen simulierten Programmablauf für die Regelung der verbauten Motoren durch. Dabei wird der Lese- und Schreibzugriff auf den gemeinsamen Bus von Mikrocontroller und FPGA nachgebildet. Ziel ist es herauszufinden, ob eine minimale Regelfrequenz von $20kHz$ eingehalten werden kann. Die Ansteuerung der Hardwarekomponenten, die durch den im System verbauten FPGA erreichbar sind, folgen einem einheitlichen Schema. Die Adresse der benötigten Komponente wird auf den Adressbus geschrieben. Dann stellt der FPGA die angeforderten Daten auf dem gemeinsamen Bus bereit und der Mikrocontroller liest die Daten ein.

¹Nach EMV-Normung bei 9kHz beginnend [31]

Das führt zu folgendem Testprogramm:

1. 8 I/Os werden als Ausgang konfiguriert
2. Eine Adresse wird auf den gemeinsamen Bus geschrieben.
3. Die 8 I/Os aus Punkt 1 werden als Eingänge konfiguriert.
4. Daten werden vom Bus gelesen.
5. Operation 1-4 werden sechsmal in Folge ausgeführt, um einen Regelkreis für zwei Motoren zu simulieren. Die Operationen stehen für das Ansteuern folgender Funktionsblöcke des FPGAs:
 - a) Zugriff auf den Analog -/ Digitalwandler (auswerten des Motorstroms)
 - b) Positionserfassung für Inkrementalgeber
 - c) PWM-Signalgenerierung für Endstufe

Die mit dem Testprogramm eingelesenen Daten werden für eine spätere Auswertung gespeichert. Eingelesen wurden die Bitmuster `0xFF`, `0x00`, `0xAA` und `0x55`. Das Bitmuster lag 5s an den I/O an. Mit einem Oszilloskop wurde die Messung der Laufzeit durchgeführt. Dafür wird zu Beginn des Programmdurchlauf ein weiterer Ausgang gesetzt und vor Beendigung zurückgesetzt. (Abb. 5.4).

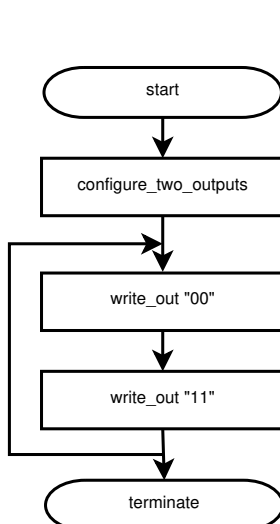


Abbildung 5.3: Programmablaufplan Testprogramm 1

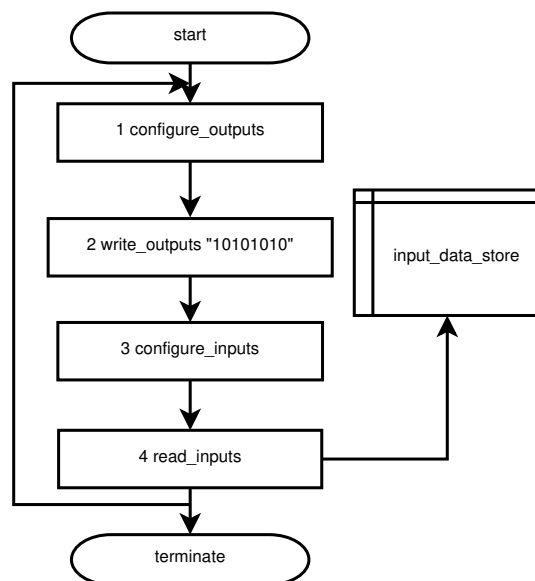


Abbildung 5.4: Programmablaufplan Testprogramm 2

Auswahl und Konfiguration der zu testenden Geräte

1. Für die PC basierte Lösung wurde ein eingebetteter PC ausgewählt. Der verbauter Prozessor ist ein Intel Core i5 und das System verfügt über 2GB Arbeitsspeicher. Aufgrund seiner geringen Größe von 15cm * 20cm lässt er sich in dem Gehäuse des Bi-Manu-Tracks verbauen. Der PC verfügt über zwei mini PCI-Express Schnittstellen. Diese werden für die Nachrüstung eines Parallelports genutzt. Zur Verfügung standen zwei PCI Express Karten mit dem Chipsatz Oxford OXPCIe952 und dem Moschip MCS9901. Das Betriebssystem Debian 7 wurde installiert. Der vorinstallierte Linux-Kernel wurde entfernt und aus den Linux-Quellen ein mit RTAI gepatchter neuer Kernel eingerichtet. Das System erfüllte damit die Anforderungen an einen Echtzeitbetrieb. Die Testprogramme für dieses System wurden als Linux Kernel Module realisiert, welches den Parallelport ansteuert. Unter Verwendung eines Timers wurde die Zykluszeit für die Ansteuerung des Parallelports auf $20kHz$ begrenzt. Ohne Begrenzung des Takts ist eine Messung nicht möglich. Das verwendete Oszilloskop hat einen Messbereich bis $2MHz$.
2. Das Texas Instruments ICE Board nutzt ein eigenes Betriebssystem TI Real-Time Operating System (RTOS). Weiter gibt es eine eigene auf Eclipse basierende Entwicklungsumgebung für das System. Mit dieser Entwicklungsumgebung wurden die Testprogramme implementiert, welches die PRUs für die Ansteuerung der I/Os verwendet.
3. Das BeagleBone Black wurde mit Debian 7 betrieben. Eine Echtzeiterweiterung für diese Plattform in Form von RTAI ist nicht verfügbar. Diese existiert für ARM basierte Prozessoren zu diesem Zeitpunkt nicht. Für die Ansteuerung der I/Os wurden die PRUs verwendet.
4. Die drei Mikrocontroller Arduino UNO, Arduino DUE und Teensy 3.2 teilen sich eine Softwareplattform. Das Testprogramm für diese Plattformen ist identisch.
5. Der Raspberry PI 2 wurde mit Debian 7 betrieben. Eine Echtzeiterweiterung für diese Plattform in Form von RTAI ist nicht verfügbar. Diese existiert für ARM basierte Prozessoren zu diesem Zeitpunkt nicht.

Die Testprogramme wurden auf allen Plattformen in der Programmiersprache C implementiert. Für alle Plattformen besteht die Möglichkeit Ein- und Ausgänge

einzelnen oder mehreren Ein- und Ausgängen gleichzeitig zu setzen. Die letzte Möglichkeit wurde für alle Tests ausgewählt.

Testergebnisse

Nach Durchlaufen der Testprogramme stehen noch zwei mögliche Lösungsansätze zur Auswahl, der Arduino UNO und der Arduino DUE. Beide Systeme erfüllen die gestellten Anforderungen. Die geforderte Regelfrequenz von $20kHz$ wird eingehalten und auch noch weit übertroffen. An den Ausgängen tritt kein Jitter auf und die Ausgänge setzen parallel das Ausgangssignal.

Der PC mit den verbauten Parallelportkarten wies als einziges getestetes System einen Jitter auf. Um Fehler in der Implementierung auszuschließen, wurde zusätzlich noch ein PC System mit nativen Parallelport getestet. Dort trat kein Jitter bei den Messungen auf. Es lässt die Vermutung zu, dass ein Problem mit der Verwendung der PCI-Express Erweiterungskarten existiert. Dies ließ sich Aufgrund fehlender, vergleichbare Systeme nicht überprüfen. Das PC System mit dem nativen Parallelport stand für eine Realisierung der Lösung nicht zur Verfügung.

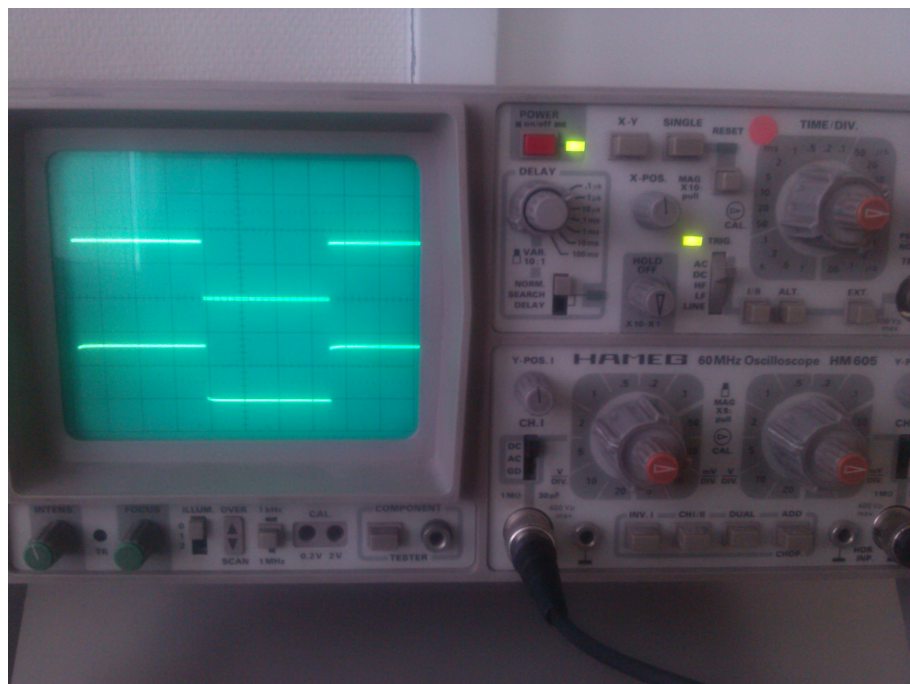


Abbildung 5.5: Messung Testprogramm 1, Texas Instruments ICE

Das BeagleBone Black, der Raspberry PI 2 und der Teensy 3.2 zeigen beim ersten Testprogramm eine Phasenverschiebung der zwei Ausgangssignale von 45° bis 90° . Die Ausgänge bei diesen Plattformen können nicht parallel geschaltet werden. Das Texas Instruments ICE Board erreicht die geforderte Regelfrequenz nicht.

Der zweite Test beinhaltet noch das Einlesen von Daten. Hier zeigt sich das der

BeagleBone Black, der Raspberry PI 2 und der Teensy 3.2 beim Auswerten des Datenspeichers Fehler in den eingelesenen Bitmustern aufweist.

Plattform	Testprogramm 1	Testprogramm 2	Jitter ²
PC mit Oxford OXPCIe952	0°	20kHz	ja
PC mit Moschip MCS9901	0°	20kHz	ja
TI ICE	0°	13kHz	nein
BeagleBone Black	90°	1,2kHz	nein
Arduino UNO	0°	2Mhz	nein
Arduino DUE	0°	444,4kHz	nein
Teensy 3.2	45°	285,71kHz	nein
Raspberry PI 2	90°	5kHz	nein

5.2.2 Entwicklung einer Systemschnittstelle

Aus der Evaluation geht hervor, dass der Arduino UNO unter der Arduino DUE für eine Umsetzung geeignet sind. Der Arduino DUE wurde ausgewählt. Der Vorteil liegt in der bereits vorhandenen CAN-Bus-Schnittstelle. Mit dieser kann eine Kopplung von zwei Bi-Manu-Tracks realisiert werden. Damit die CAN-Bus-Schnittstelle des Arduino DUE genutzt werden kann, wird ein Transceiver im Adapter verbaut. Dieser ist auf dem Board des Arduino DUE nicht integriert. Weiter kann eine externe Auswertung der Sensorwerte erfolgen. Für die Umsetzung muss ein Adapter entwickelt werden, der in den freien Sockel der Platine des Bi-Manu-Tracks gesteckt werden kann.

Eagle

Für die Entwicklung des Adapters wurde das Programm Eagle benutzt. Mit Eagle wurde ein Schaltplan und ein Platinenlayout für eine spätere Fertigung erstellt. Eagle ist ein Programm zur Entwicklung von bestückbaren Leiterplatten. Das Programm ist in zwei Komponenten aufgeteilt:

1. Schaltplan-Editor
2. Layout-Editor

²Bei den Versuchen trat kein messbarer Jitter auf oder der Jitter war so groß, dass das Signal nicht mehr verwertbar war.

Mit dem Schaltplan-Editor wird die zu entwickelnde Schaltung erstellt. Der Schaltplan-Editor enthält alle grundlegende Bauelemente aus der Elektro- und Digitaltechnik. Fehlende Bauelemente können importiert oder selbst erstellt werden. Das Platinenlayout wird aus dem Schaltplan generiert. Im Layout-Editor kann das Platinenlayout für eine spätere Fertigung bearbeitet werden.

5.2.3 Schaltplan

Legende:

- 1 - Verbindung zum Bi-Manu-Track ausgeführt als Stiftleiste
- 2,3,4,6 - Verbindung zum Bi-Manu-Track ausgeführt als Stiftleiste
- 5 - CAN-Bus-Transceiver (Texas Instruments SN65HVD234)
- 7 - Bidirektionaler Pegelwandler für die Verbindung von CMOS und TTL - Technik (TXB0108)
- 8 - 2 Kondensatoren für den Transceiver. $100nF$ und $10\mu F$

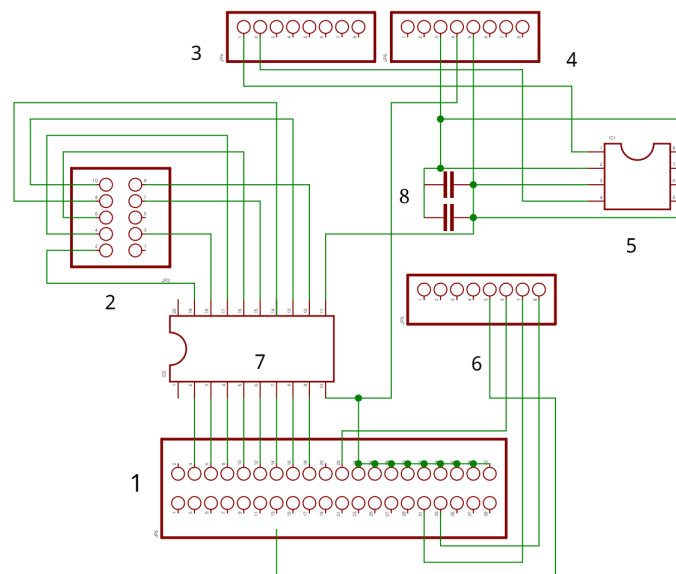


Abbildung 5.6: Bi-Manu-Track-Adapter Schaltplan

Adapter

Nach Vorgabe des Schaltplans wurde eine Europlatine in den Maßen 10cm * 7cm mit Bauteilen bestückt. Die bestückte Platine wurde auf dem Sockel des Bi-Manu-Tracks befestigt und auf dieser der Arduino DUE.

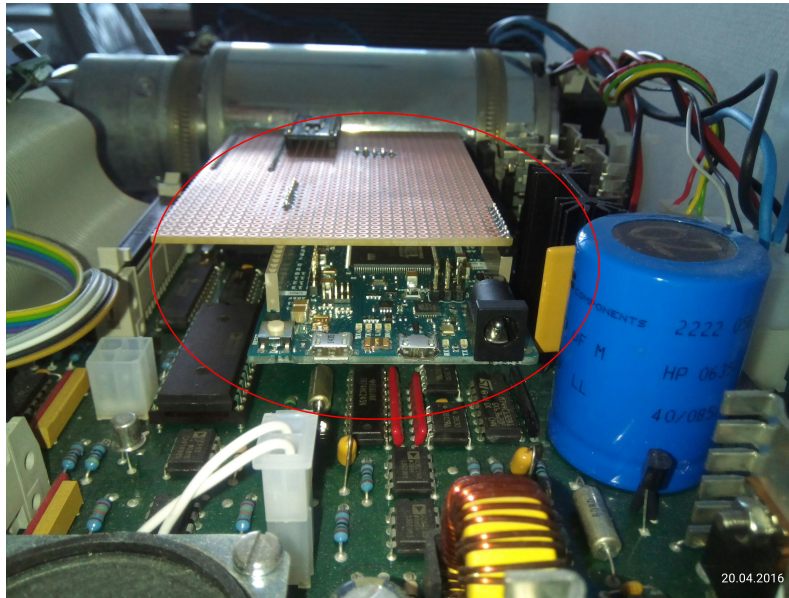


Abbildung 5.7: Bi-Manu-Track mit Adapter und Arduino DUE

5.2.4 Implementierung

Für die Implementierung wurde die Entwicklungsumgebung von PlatformIO genutzt und als Sprache C++ gewählt.

PlatformIO

PlatformIO ist ein Open-Source Entwicklungssystem für Mikrocontroller und eingebettete Systeme. Es wird eine Vielzahl von Frameworks für verschiedene Zielplattformen angeboten. Dazu gehört auch die komplette Arduino Familie unabhängig von dem verwendeten Prozessor. PlatformIO ermöglicht es einem während der Entwicklung die Zielplattform auch herstellerübergreifend zu wechseln. Integriert in die Umgebung ist die Möglichkeit das geschriebenen Programm auf die Zielplattform zu laden und auch zu debuggen. PlatformIO kann mit Eclipse, Atom oder mit jedem beliebigen Editor genutzt werden. In diesem Projekt wurde sich für den Atom Editor entschieden. Für diesen bietet PlatformIO ein Plugin, wodurch eine sehr gute

Integration in den Atom Editor realisiert wird. [32]

Umsetzung der Steuerung

Basierend auf der Systembeschreibung der im Bi-Manu-Track verbauten Platine sind zwei Bibliotheken für die Umsetzung der Steuerung entwickelt worden. Die Bibliothek „io“ realisiert die Zugriffe auf den verbauten FPGA und die Bibliothek „controll“ stellt eine API für die Ansteuerung einzelner Funktionen des Bi-Manu-Tracks bereit.

Das Hauptprogramm bindet die API der Bibliothek „controll“ ein. Damit ist ein Zugriff auf alle verbauten Komponenten des Bi-Manu-Tracks möglich.

Bibliothek „io“

Die Bibliothek bietet direkt Zugriff auf den Adress- und Datenbus. Folgend wird kurz der Ablauf und die Funktionsweise erläutert.

init Mit der Funktion `init` wird der Buszugriff initialisiert.

1. `configure_outputs` wird aufgerufen
2. `configure_data_dir` stellt den Schreibzugriff auf den Bus her.

configure_outputs Diese Funktion konfiguriert vier I/Os des Arduino als Ausgänge, die an den Eingängen `Read`, `Write` und `Ale` des FPGAs angeschlossen sind.

configure_data_dir Mit dieser Funktion wird festgelegt, ob auf den Adress- und Datenbus lesend oder schreibend zugegriffen wird. Mit einem Parameter wird der gewünschte Modus ausgewählt

read_data Die Funktion liest Daten von dem Bus. Als Parameter wird die Zieladresse eines Funktionsblock angegeben.

1. Die Zieladresse wird auf den Bus geschrieben
2. `Ale` setzen, der FPGA übernimmt die Adresse.
3. `configure_data_dir` stellt den Buszugriff auf lesend ein.
4. `Read` wird gesetzt
5. Der Wert, der auf dem Bus liegt ist der Rückgabeparameter
6. `init` wird aufgerufen

write_data Die Funktion `write_data` schreibt Konfigurationsparameter in das Datenregister des FPGAs. Zwei Parameter müssen der Funktion übergeben werden, die Zieladresse und der Wert, der geschrieben werden soll.

1. Die Zieladresse wird auf den Bus geschrieben
2. Ale gesetzt
3. Der Zielwert wird auf den Bus geschrieben
4. Write wird gesetzt
5. init wird aufgerufen

Bibliothek „controll“

Die Funktionen in der Bibliothek nutzen die API der Bibliothek „io“ für eine Lese- oder Schreibzugriff auf den Adress- und Datenbus. Das Parametrieren einzelner Funktionsblöcke und Auslesen erfordert eine Sequenz von Lese- und Schreiboperation. Diese werden anhand des Inkrementalgebers im Detail erläutert. Die weiteren Funktionen arbeiten analog dazu.

encoder_read Die Funktion liefert die aktuellen Werte, die an den Inkrementalgebern liegen zurück. Mit einem Parameter kann zwischen den zwei Inkrementalgebern gewählt werden. Die Rückgabewerte der Inkrementalgebers sind im Gegensatz zu den anderen Werten *16Bit* groß. Das Ergebnis wird von dem FPGA an zwei Stellen des Datenregisters gespeichert. An einer Adresse wird das High Byte und an einer anderen Adresse das Low Byte hinterlegt.

1. `write_data` wird mit der Zieladresse des Inkrementalgebers aufgerufen. Der zweite Parameter konfiguriert den FPGA den aktuellen Zählerstand in den Registern für das Low und High Byte zu hinterlegen.
2. `read_data` wird mit der Zieladresse des Low Bytes aufgerufen
3. `read_data` wird mit der Zieladresse des High Bytes aufgerufen
4. das High Byte wird um acht Stellen nach links geschiftet und mit dem Low Byte verodert. Der 16 Bit große Wert wird von der Funktion zurückgegeben.

encoder_set Mit `encoder_set` wird der Inkrementalgeber mit einem Startwert initialisiert. Mit einem Parameter kann zwischen den zwei Inkrementalgebern gewählt werden.

1. `write_data` wird mit der Zieladresse des Inkrementalgebers aufgerufen. Der zweite Parameter konfiguriert den FPGA Schreibzugriff auf das Zählregister zu erlangen.
2. `write_data` wird mit der Zieladresse des Low Bytes aufgerufen. Der zweite Parameter beinhalten das Low Byte des neuen Zählerwertes.
3. `write_data` wird mit der Zieladresse des High Bytes aufgerufen. Der zweite Parameter beinhalten das High Byte des neuen Zählerwertes.
4. `write_data` wird mit der Zieladresse des Inkrementalgebers aufgerufen. Der zweite Parameter veranlasst den FPGA das High -und Low Byte in das Zählerregister zu kopieren.

motor_enable gibt die Motoren frei. Vor Aufruf dieser Funktion ist ein Verfahren dieser nicht möglich.

1. `write_data` wird mit der Zieladresse der Motoren aufgerufen. Der zweite Parameter konfiguriert den FPGA, die Endstufe freizuschalten.

motor_set_velo Mit dieser Funktion wird die Geschwindigkeit der Motoren eingestellt.

1. `write_data` wird mit der Zieladresse des ausgewählten Motoren-PWM-Registers aufgerufen. Der zweite Parameter gibt die Geschwindigkeit an.

motor_get_current liest den Momentan am Motor anliegenden Strom aus.

1. `read_data` erhält die Zieladresse des Motor-Stromregisters.

motor_get_voltage liefert die Betriebsspannung der Motoren.

1. `read_data` erhält die Zieladresse des Motor-Spannungsregisters.

Steuerung

Auf Basis der entwickelten Bibliotheken für den Zugriff auf die einzelnen Funktionsblöcke des Bi-Manu-Tracks wurde eine Steuerung für das Systems entwickelt. Ein Proportional Integral Regler wurde für die Ansteuerung der Motoren implementiert. Als Regelkreis wurde eine strombasierte Regelung umgesetzt. Mit der strombasierten Regelung wird kontinuierlich der an den Motoren anliegende Strom gemessen. Wird auf die an den Motoren angebrachten Endeffektoren von einer Person eine Kraft aufgebracht, dann wird das an den Motoren anliegende PWM-Signal reduziert bis

der gemessene Strom mit dem vorgegebenen Sollwert übereinstimmt. Weiter werden die gemessenen Werte, der Inkrementalgeber, der Motorspannung und des Motorstroms unter Verwendung der CANbus Schnittstelle an einen weiteren Teilnehmer im CAN-Bus geschickt, der die Messwerte für eine spätere Auswertung speichert.

Für den Anwendungsfall das sich zwei Bi-Manu-Tracks in einem Controller Area Network befinden, ändert sich der Programmablauf der Steuerung. Anstelle der strombasierten Regelung wird eine zweite positionsbasierte Regelung genutzt. Die Eingangsparameter sind die aktuell gemessene Position des Endeffektors und die neue gewünschte Position des Endeffektors. Die gewünschte Zielposition wird aus dem empfangenem Datagramm entnommen. Der positionsbasierte Regler gibt ein PWM-Signal an den Motor aus. Der Endeffektor dreht sich zur neuen gewünschten Position. Erhält der Regler die gewünschte Position als aktuell gemessene Position als Eingangsparameter, wird kein PWM-Signal mehr ausgegeben. Der Endeffektor hat die vom zweiten Bi-Manu-Track empfangene Position erreicht.

6 Auswertung

6.1 Graspit-System

6.1.1 Performancemessung

Ein Teil der Aufgabenstellung beinhaltet den Vergleich der Performance der neuen und alten Umsetzung. Das ursprüngliche System arbeitet mit einer Regelfrequenz von 1000Hz für die haptische Interaktion. Die neue Umsetzung muss das auch leisten. Zur Überprüfung der Regelfrequenz wurde eine Evaluation der neuen Umsetzung durchgeführt. Für die Überprüfung wurde das System wie in Abbildung 5.1 in Betrieb genommen. Für die Simulation wurde eine vorhandene Testanwendung genommen. Die Anwendung stellt dreidimensional gestapelte Quader dar. Das Graspit-System ist mit einem dieser Quader gekoppelt. Mit dem Endeffektor ist es möglich, die anderen Quader zu verschieben. Bei Kollision berechnet die Simulation einen Kraftvektor. Dieser wird durch die beiden Phantoms wiedergegeben (Abb. 6.1).



Abbildung 6.1: Jenga

Die neu implementierte Steuerung wurde für den Performancetest leicht modifiziert. Am Ende eines Prozesszyklus wird ein weiterer Ausgang gesetzt und im nächsten Durchlauf zurückgesetzt. Der Ausgang toggelt. Mit einem Oszilloskop wurde die Frequenz an diesem Ausgang gemessen. Für den Test wurden verschiedenen Zykluszeiten gewählt. Weiter wurde die Auslastung des Prozessors gemessen. Dafür wurde das Kommandozeilenwerkzeug *htop* verwendet. Der Prozess der Soft-SPS läuft auf einem der vier im Raspberry PI 2 vorhandenen Prozessorkerne.

Zykluszeit	Frequenz	CPU Auslastung
<i>0ms</i>	0Hz	2%
<i>1ms</i>	1000Hz	70%
<i>0.5ms</i>	2000Hz	90%
<i>0.33ms</i>	2100Hz	98%
<i>0, 1ms</i>	1900Hz	98%

Tabelle 6.1: Performance Messung Graspit-System

Ergebnis

Die Tabelle 6.1 zeigt, dass die neue Umsetzung des Graspit-Systems die benötigte Regelfrequenz von $1000Hz$ einhält. Allerdings darf nicht ausgelassen werden, dass bei Einhalten der Frequenz die CPU Auslastung bei 70% liegt. Es ist davon auszugehen, dass bei einer komplexeren Steuerung eine Regelfrequenz von $1000Hz$ nicht zu halten ist.

6.1.2 Validierung

Die Anforderung für die Entwicklung einer neuen Steuerung für das Graspit-System wurden in Abschnitt 4.1.4 festgelegt.

1. Grundvoraussetzung für ein haptisches Interaktionsgerät ist es, dass eine Mindestfrequenz von $1000Hz$ für den Regelkreislauf der Steuerung eingehalten wird.
2. Die Modularität des Systems sollte erhöht werden, damit das Gerät besser an spezifische Einsatzszenarien angepasst werden kann.
3. Die Wartbarkeit und Pflege des Systems muss verbessert werden.

Das Einhalten der Regelfrequenz wurde mit einer Performancemessung des Graspit-Systems in Unterabschnitt 6.1.1 bestätigt. Damit erfüllt das umgesetzte System die Anforderungen an die kinästhetische Interaktion. Weiter wurde die geforderte Modularität erhöht. Dies wurde durch die Umsetzung der Anforderung mit modularen Baugruppen aus der Automatisierungstechnik erreicht. Zugleich wurde damit die Wartung und Pflege des Systems verbessert. Einzelne Elemente lassen sich mit geringen Aufwand austauschen oder auch neue hinzufügen. Dies betrifft ebenso die Softwarekomponenten. Steuerung, Kinematik und Anwendungscient sind einzelne Anwendungen, die miteinander vernetzt sind.

6.2 Bi-Manu-Track

6.2.1 Validierung

Für die Erweiterung des Bi-Manu-Tracks wurden in Abschnitt 4.3.3 Anforderungen definiert.

1. Eine Netzwerkschnittstelle muss integriert werden, um die Möglichkeit zu schaffen zwei Geräte miteinander zu koppeln. Weiter soll dadurch die Aufzeichnung der gemessenen Werte ermöglicht werden.
2. Mit der neuentwickelten Lösung muss es möglich sein, bestehende Geräte nachzurüsten.
3. Die Technischen Veränderungen sollen möglichst gering ausfallen.
4. Für die Ansteuerung der Funktionsblöcke des FPGAs (Inkrementalgeber, Motorstrom und PWM-Signalgenerierung) muss eine Regelfrequenz von $20\mu s$ eingehalten werden.
5. Für die Ansteuerung des Adress- und Datenbusses müssen die I/Os der Steuerung im Betrieb von Eingang auf Ausgang und vice versa konfiguriert werden können.
6. Die Daten müssen parallel auf den Adress- und Datenbus geschrieben werden.

Die Anforderungen für eine Erweiterung des Bi-Manu-Tracks, mit eine, Netzwerkschnittstelle, konnten erfüllt werden. Um die sehr speziellen Systemeigenschaften des Bi-Manu-Tracks zu handhaben, wurde nach der Evaluierung potentieller Lösungen,

ein ausführlicher Test dieser durchgeführt. So konnten im Vorfeld Hindernisse ausgeräumt werden. Dies trifft besonders auf die Notwendigkeit zu, hochfrequent auf das Bussystem zuzugreifen und Daten parallel zu schreiben. Die geforderte Netzwerkschnittstelle wurde durch die Integration einer CAN-Schnittstelle realisiert. Eine Kopplung mit einem weiteren Gerät wurde damit ermöglicht, sowie das Aufzeichnen von anfallenden Messwerten. Die Entwickelte Lösung erfüllt auch den geforderten Punkt, bestehende Systeme leicht nachzurüsten. Weiter waren die technischen Veränderungen der Hardware minimal.

7 Fazit

Die ausgewählten Entwickelten Konzepte konnten erfolgreich umgesetzt werden. Durch die in der Konzeption aufgestellten Anforderungen an eine Umsetzung der Systeme, sind die zu Berücksichtigenden Systemeigenschaften erfasst worden. Die im Kapitel Stand der Technik vorgestellten Technologien wurden evaluiert und für die Umsetzung genutzt. Durch diese Verfahrensweise wurden zwei Systeme vor dem Hintergrund Industrie 4.0 realisiert.

Beim Graspit-System wurde die Modularität und der Grad der Vernetzung erhöht, zwei Anforderungen die Industrie 4.0 stellt. Durch eine Messung der Performance wurde bestätigt, dass das neu umgesetzte Konzept die gestellten Anforderungen erfüllt.

Der Bi-Manu-Track wurde um eine CAN-Schnittstelle erweitert. Das wurde durch das Ersetzen des verbauten Mikrocontrollers erreicht. Die Veränderung an der Hardware des Bi-Manu-Tracks fiel durch diese Lösung minimal aus. Durch die Netzwerkschnittstelle ist es möglich geworden, von den verbauten Sensoren erfasste Messwerte auszuwerten. Weiter ist eine Kopplung zweier Geräte möglich. Dadurch lassen sich telemedizinische Anwendungen umsetzen. Die Übungsprogramme für eine Therapie wurden im Rahmen dieser Arbeit nicht implementiert. Weiter ist eine Bedienung des Gerätes durch einen Therapeuten noch nicht gegeben.

Literatur

- [1] ITWissen, Cip-protokoll.
Adresse: <http://www.itwissen.info/definition/lexikon/common-industrial-protocol-CIP-CIP-Protokoll.html> (besucht am 05.02.2016).
- [2] Bundesministerium für Bildung und Forschung,
Zukunftsprojekt industrie 4.0.
Adresse: <https://www.bmbf.de/de/zukunftsprojekt-industrie-4-0-848.html>
(besucht am 20.01.2016).
- [3] H. Kargermann, W. Wahlster und J. Helbig,
Umsetzungsempfehlungen für das Zukunftsprojekt Industrie 4.0, April. 2013,
S. 1–116. Adresse:
https://www.bmbf.de/files/Umsetzungsempfehlungen_Industrie4_0.pdf.
- [4] D. Balkhausen,
Die dritte industrielle Revolution. Wie Mikroelektronik unser Leben verändert.
1978, ISBN: 3430111471.
- [5] Die Dritte industrielle Revolution – Aufbruch in ein ökologisches Jahrhundert.
Bundesministerium für Umwelt, Naturschutz und Reaktorsicherheit, 2008,
S. 148.
- [6] M. Jänicke und K. Jacob, »Die dritte industrielle revolution«,
INTERNATIONALE POLITIK, 2008.
Adresse: http://www.polsoz.fu-berlin.de/polwiss/forschung/systeme/ffu/publikationen/2008/jaenicke_martin_jacob_klaus/Flyer_3__ind__Revol__Otto_Suhr_Institut_IP_10_2008.pdf.
- [7] S. Horvath, »Aktueller begriff internet der dinge«,
Online-Dienste des Deutschen Bundestages, Bd. 12, Nr. 19, S. 1–2, 2007.

- [8] C. Taphorn, D. Wolter, N. Braun, A. Goloverov, S. Ludwig, C. Prasse und E. Fuss, »Erschließen der potenziale der anwendung von industrie 4.0 im mittelstand«, Agiplan, Fraunhofer IML, ZENIT, S. 401, 2015.
- [9] K. Bettenhausen und S. Kowalewski, »Cyber-physical systems : chancen und nutzen aus sicht der automation«, VDI/VDE-Gesellschaft Mess- und Automatisierungstechnik, Nr. April, S. 1–12, 2013.
- [10] K. R. Gegenfurtner, Gehirn und Wahrnehmung. 2003, ISBN: 9783596190218.
- [11] H. Penzlin, Lehrbuch der Tierphysiologie. Spektrum, ISBN: 9783827421142.
- [12] M. Grunwald und L. Beyer, Der bewegte Sinn: Grundlagen und Anwendungen zur haptischen Wahrnehmung. 2001, ISBN: 9783764365165.
- [13] ITWissen, Speicherprogrammierbare steuerung.
Adresse: <http://www.itwissen.info/definition/lexikon/SPC-Speicherprogrammierbare-Steuerung-SPS-storage-programmable-control.html> (besucht am 15.04.2016).
- [14] B. Heinrich, P. Linke und M. Glöckler, Grundlagen Automatisierung, ISBN: 9783658059613.
- [15] Beckhoff, Beckhoff. Adresse: <http://www.beckhoff.de/>.
- [16] L. Dozio und P. Mantegazza, »Linux real time application interface (rtai) in low cost high performance motion control«, Motion Control, 2003.
- [17] N. P. Mahalik, Fieldbus Technology, 9. 2013, Bd. 53, ISBN: 9783662072196.
arXiv: [arXiv:1011.1669v3](https://arxiv.org/abs/1011.1669v3).
- [18] Ethercat Technology Group, »Ethercat - der ethernet-feldbus«,
Adresse: http://www.ethercat.org/pdf/german/ETG_Brochure_DE.pdf.
- [19] Cia, Can lower- and higher-layer protocols.
Adresse: <http://www.can-cia.org/can-knowledge/>.
- [20] Profinet Profibus, Profibus.
Adresse: <http://www.profibus.com/technology/profibus/overview/>.

- [21] U. Völlinger, B. Beckmann-Dobrev, J. Schäfer, J. H. Israel und R. Stark, »Intuitive haptische interfaces für die deformationssimulation«, in 8. Paderborner Workshop Augmented & Virtual Reality in der Produktentstehung, Bd. 252, 2009, S. 189–203, ISBN: 9783939350712.
- [22] F.-L. Krause, J. Israel, J. Neumann und B. Beckmann-Dobrev, »A 6-dof user interface for grasping in vr-based computer aided styling and design«, in Proceedings of the ACM symposium on Virtual reality software and technology, 2005, S. 110–112, ISBN: 1595930981. DOI: [10.1145/1101616.1101638](https://doi.org/10.1145/1101616.1101638). Adresse: <http://portal.acm.org/citation.cfm?doid=1101616.1101638>.
- [23] G. Kroah-Hartman, The linux kernel driver interface. Adresse: https://git.kernel.org/cgit/linux/kernel/git/torvalds/linux.git/tree/Documentation/stable_api_nonsense.txt (besucht am 22.06.2016).
- [24] PROFIBUS Nutzerorganisation, PROFIBUS Planungsrichtlinie, August. 2009, S. 1–232. Adresse: [http://www.profibus.com/index.php?id=1610&tx_pxdprofibus_pi2\[download\]=1023&tx_pxdprofibus_pi2\[file\]=8832&tx_pxdprofibus_pi2\[mode\]=download](http://www.profibus.com/index.php?id=1610&tx_pxdprofibus_pi2[download]=1023&tx_pxdprofibus_pi2[file]=8832&tx_pxdprofibus_pi2[mode]=download).
- [25] Beckhoff New Automation Technology, Echtzeit - einstellungen. Adresse: http://infosys.beckhoff.com/index.php?content=content/1031/TcSystemManager/Basics/TcSysMgr_ConfigRT_Intro2.htm (besucht am 15.06.2016).
- [26] S. Hesse, A. Waldner, J. Mehrholz, C. Tomelleri, M. Pohl und C. Werner, »Combined transcranial direct current stimulation and robot-assisted arm training in subacute stroke patients: an exploratory, randomized multicenter trial.«, Neurorehabilitation and neural repair, Bd. 25, Nr. 9, S. 838–46, 2011, ISSN: 1552-6844. DOI: [10.1177/1545968311413906](https://doi.org/10.1177/1545968311413906). Adresse: <http://www.ncbi.nlm.nih.gov/pubmed/21825004>.
- [27] S. Hesse, C. Werner und A. Bardeleben, »Der schwer betroffene arm ohne distale willkuraktivit - ein sorgenkind der rehabilitation nach schlaganfall?!«, Neurologie und Rehabilitation, Bd. 10, Nr. 3, S. 123–129, 2004, ISSN: 09472177.
- [28] BeMobil, Bemobil: bemobil. Adresse: <http://www.bemobil.net/>.
- [29] H. Westphal und G. Kuhlmann, »Manutrack - systembeschreibung«, Techn. Ber., 2003.

- [30] Beckhoff, Busklemmen analog-output. Adresse: http://www.beckhoff.de/default.asp?bus_terminal/anaout.htm?id=365410973
(besucht am 01.03.2016).
- [31] J. Detlefsen und U. Siart, Grundlagen der Hochfrequenztechnik.
München, 2006, ISBN: 3486578669.
- [32] Platformio.org, What is platformio?
Adresse: <http://docs.platformio.org/en/latest/what-is-platformio.html>
(besucht am 21.05.2016).

Abbildungsverzeichnis

2.1	Industrieentwicklung, Quelle DFKI 2011	3
3.1	Prozessdaten im EtherCAT Telegram [18, S. 7]	15
3.2	Struktur der Kommunikationsprofile im EtherCAT Telegramm [18, S. 13]	16
4.1	Das Graspit System[21, S. 11]	24
4.2	Phantom Premium 3.0 (im Vordergrund)	24
4.3	Das Graspit System - vereinfachte schematische Darstellung	26
4.4	Schematische Darstellung des neuen Konzeptes für das Graspit-System	28
4.5	Haptikzelle Quelle: Faunhofer IPK	30
4.6	Bi-Manu-Track Pronation und Supination Quelle: http://www.reha-stim.de/	32
4.7	Schematische Darstellung Systemaufbau Bi-Manu-Track	35
5.1	Das neue Grapsit-System - schematische Darstellung	41
5.2	Programmablaufplan SPS	42
5.3	Programmablaufplan Testprogramm 1	44
5.4	Programmablaufplan Testprogramm 2	44
5.5	Messung Testprogramm 1, Texas Instruments ICE	47
5.6	Bi-Manu-Track-Adapter Schaltplan	49
5.7	Bi-Manu-Track mit Adapter und Arduino DUE	50
6.1	Jenga	55

Tabellenverzeichnis

3.1	CAN Bus Leitungslänge und Übertragungsrate	17
4.1	Bussysteme im Vergleich	29
6.1	Performance Messung Graspit-System	56

Abkürzungen

ABI	Application Binary Interface
API	Application Programming Interface
CAN	Controller Area Network
CoDeSys	Controller Development System
CoE	CAN application protocol over EtherCAT
CPS	Cyber-Physisches-System
EoE	Ethernet over EtherCAT
FoE	File over EtherCAT
Fraunhofer IPK	Fraunhofer Institut für Produktionsanlagen und Konstruktionstechnik
GCC	GNU Compiler Collection
GPIO	General Purpose Input/Output
ICE	Industrial Communications Engine
IdD	Internet der Dinge
IDE	Integrated Development Environment
IEC	International Electrotechnical Commission
I/O	Input / Output
IPC	Industrie PC
PCI	Peripheral Component Interconnect

PC	Personal Computer
PI	Proportional Integral Regler
PRU	Programmable Real-Time Unit
PWM	Pulsweitenmodulation
QR-Code	Quick Response Code
RFID	Radio-frequency identification
RISC	Reduced Instruction Set Computer
RTAI	Real Time Application Interface
SERCOS	Serial Realtime Communication System
SoE	Servo drive profile over EtherCAT
SPI	Serial Peripheral Interface
SPS	Speicherprogrammierbare Steuerung
UDP	User Datagram Protocol
USB	Universal Serial Bus