

Montanuniversität Leoben

Studienrichtung Montanmaschinenwesen



DIPLOMARBEIT

am

Institut für Automation

zum Thema

„Entwicklung eines Teleoperating – Systems
mit Krafrückkoppelung für ein Kettenfahrzeug“

Teil I

von

cand.ing. Robert Diosi

Mein Dank gilt

**Herrn O. Univ. Prof. Dipl.-Ing. Dr. techn.
Paul O'LEARY**

für die Betreuung und Unterstützung bei all meinen Anliegen, sowie für die Bereitstellung
aller dafür notwendigen Einrichtungen des Institutes.

Weiters möchte ich mich bei

**Herrn Dipl.-Ing.
Ronald Ofner**

und

**Herrn Dipl.-Ing
Gerhard Rath**

für die Unterstützung bei all meinen Anliegen bedanken.

Nicht zuletzt danke ich an dieser Stelle meinen Eltern,
die mir dieses Hochschulstudium ermöglicht haben.

An dieser Stelle erkläre ich Eides Statt, die vorliegende Diplomarbeit selbständig und nur unter der Verwendung der angegebenen Literatur angefertigt zu haben.

Leoben, 26. Februar 1999

(Robert Diosi)

Inhaltsverzeichnis

1	Teleoperatingsysteme	1
1.1	Einleitung.....	1
1.2	Funktionsweise von Teleoperatingsystemen	2
2	Projektbeschreibung	3
2.1	Einleitung.....	3
2.2	Ziel der Diplomarbeit	5
2.3	Versuchsaufbau.....	7
3	Eingabegerät	8
3.1	Einleitung.....	8
3.2	Aufbau des Force Feedback Joysticks	9
3.2.1	Allgemeine Beschreibungen.....	9
3.2.2	Kraftrückführungsmöglichkeiten.....	12
4	Kameraschwenksystem.....	14
4.1	Einleitung.....	14
4.2	Gesamtkonzept	15
4.3	Aufbau des Kameraschwenksystems.....	16
4.4	Schwenkmechanik	17
4.4.1	Schwenkbereiche	17
4.4.2	Hauptlager	18
4.4.3	Erforderliches Antriebsmoment	24
4.4.4	Antrieb der Schwenkachsen	25
4.5	Bereitschaftsanzeigen des Kameraschwenksystems.....	26
4.6	Signal- und Leistungsflüsse zum Kameraschwenksystems.....	27
4.7	Innerer Aufbau des Kameraschwenksystems	28
4.8	Batterieversorgung.....	29
4.9	Servomotor	30
4.10	Pulsweitenmodulation.....	31
4.10.1	SPS – Modul für Pulsweitenmodulation	32

5	Videokamera	33
6	Videofunksystem	35
7	Datenfunkstrecke	36
8	Kettenfahrzeug	37
	4.1 Einleitung.....	37
9	Ultraschallsensor	40
10	Stationäre PC-Einheit	42
11	Abbildungsverfahren	43
	11.1 Abbildung der Motorgeschwindigkeiten zur Joystickposition	43
	11.1.1 Einleitung.....	43
	11.1.2 Eingabegerät	44
	11.1.3 Ansteuerung der Antriebsmotoren.....	45
	11.1.4 Abbildung der Motorgeschwindigkeiten	46
	11.2 Abbildungsverfahren für die Krafrückkoppelung	53
	11.2.1 Einleitung.....	53
	11.2.2 Messung von Abständen zu Hindernissen.....	54
	11.2.3 Mathematische Abbildung.....	55
12	Softwaredokumentation	57
	12.1 Einleitung.....	57
	12.2 Komponenten des Softwarepaketes	58
	12.3 Datenflüsse im Gesamtsystem.....	59
	12.4 Hauptprogramm.....	60
	12.4.1 Bedienungsoberfläche des Hauptprogrammes	61
	12.4.2 Kurzbeschreibung der Komponenten des Hauptprogrammes.....	62
	12.4.3 Funktionsbeschreibung der MDI-Form	63
	12.4.4 Funktionsbeschreibung der MainForm.....	65
	12.4.5 Schnittstellen des ActiveX - Elementes.....	72
	12.5 DDE-Verbindung.....	75
	12.5.1 Einleitung.....	75
	12.5.2 Aufbau der DDE-Verbindung	76

12.6 ActiveX-Komponente.....	77
12.6.1 Einleitung.....	77
12.6.2 Programmierung der ActiveX-Komponente	78
12.7 Echtzeitimplementationen	82
12.7.1 Implementierte Task – Elemente.....	82
13 Resümee	83
13.1 Fazit	83
13.1 Erweiterungs- und Verbesserungsmöglichkeiten	84
14 Literaturliste	85
13.1 Veröffentlichungen.....	85

1 Teleoperatingsysteme

1.1 Einleitung

Teleoperatoren sind ferngesteuerte Manipulatoren, die keine Programmsteuerung besitzen. Diese übernimmt der Mensch, der die notwendigen Entscheidungen trifft und die Bewegungen einleitet. Teleoperatoren sind Kraft-, Leistungs- und Reichweitenverstärker der menschlichen Handhabungseigenschaften. Steht ein entsprechendes Kommunikationssystem zur Verfügung, so kann der Teleoperator in beliebiger Entfernung aufgebaut werden und dort arbeiten. Im industriellen Bereich werden Schwerlast – Manipulatoren dort eingesetzt, wo der Mensch von schwerer physischer Arbeit entlastet werden soll, aber das Steuern der Bewegungsabläufe weiterhin dem Menschen überlassen werden muß. [1]

Teleoperatingsysteme haben in vielen Bereichen der Industrie Einzug gefunden. Derartige Einrichtungen werden besonders dort verwendet, wo der Mensch durch zu große Entfernungen nicht anwesend sein kann, oder die Gesundheit des Bedieners gefährdet wäre. Insbesondere im Bergbau können Teleoperatingsysteme verwendet werden, um den Bediener von Bergwerksmaschinen gesundheitlich zu entlasten, ohne die eigentliche Steuerungsaufgabe durch den Menschen zu beeinflussen.

1.2 Funktionsweise von Teleoperatingsystemen

Bei Teleoperatingsystemen ist die Steuerung eines Manipulators durch einen Bediener unerlässlich, da es sich meist um komplexe Bewegungsabläufe handelt, welche derzeit durch elektronische Steuerungen (Regelungen) nicht, oder nur mit sehr hohem Aufwand, realisiert werden können. Dabei erhält der Bediener die visuelle Darstellung des Handhabungsortes mittels Videobildaufnahme, und kann aufgrund dessen Entscheidungen für den weiteren Bewegungsablauf treffen. Die Bewegungsvorgaben werden durch Eingabegeräte realisiert, welche die Freiheitsgrade des Manipulators abdecken, und intuitiv zu bedienen sind.

Aspekt der Krafrückführung

Der Operator erhält durch eine rein visuelle Darstellung des Handhabungsortes keinen haptischen Eindruck von den am Manipulator wirkenden Kräfte. Dies wirkt sich negativ auf die Steuerung des Manipulators aus. Um dem Bediener einen zusätzlichen Eindruck vom Handhabungsort zu vermitteln, besteht die Möglichkeit Kräfte zum Operator mittels geeigneter Eingabegeräte rückzuführen, um so eine „Regelschleife“ zwischen Operator und Manipulator zu erhalten.(Abb.1.1)

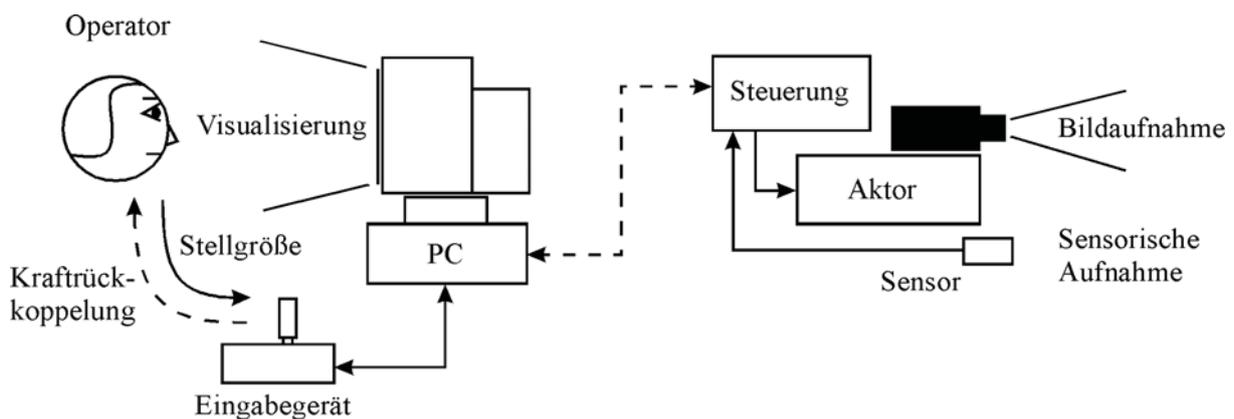


Abb. 1.1 Allgemeiner Aufbau eines Teleoperatingsystems

2 Projektbeschreibung

2.1 Einleitung

Bei der Rohstoffgewinnung wird zunehmend versucht Personen aus Gefahrenbereiche zu entfernen. Insbesondere bei Bergbaumaschinen ist der Bediener großen Verletzungsrisiken und gesundheitlichen Belastungen unterworfen. Mit dem Einsatz eines Teleoperatingsystems ist es möglich, Bergbaumaschinen außerhalb des Gefahrenbereiches zu bedienen.

Das Teleoperatingsystem muß hierbei mit einer geeigneten „Fernbedienung“ und „Fernbildübertragung“ ausgestattet sein.

In Anlehnung an ein Projekt der Firma VA Bergtechnik (Abb.2.1) soll bei dieser Diplomarbeit ein Teleoperatingsystem für ein Kettenfahrzeug mit Krafrückführung entwickelt werden.



Abb. 2.1 Alpine Miner

Durch die Entfernung des Operators zur Bergbaumaschine kommt es zu einer fehlenden haptischen Wahrnehmung. Der Bediener nimmt nicht in gleichem Maße das wahr, was an der Maschine passiert.

In diesem Projekt soll hierfür eine geeignete Lösung für ein Teleoperatingsystem mit Krafrückführung zum Operator entworfen und implementiert werden.

Die Forderung nach teilautonomen Aktionen wird bei diesem Projekt nicht behandelt. Die Hardwarelösung ist bereits so ausgelegt, daß teilautonome Bewegungen implementiert werden können.

2.2 Ziel der Diplomarbeit

Im Rahmen dieser Diplomarbeit soll ein Prototyp für ein Teleoperatingsystem mit haptischer Rückführung entwickelt werden. Der Versuchsaufbau soll als Erprobungsmöglichkeit für Algorithmen zur Prüfung der subjektiven haptischen und visuellen Wahrnehmung dienen. Die Anforderungen und Vorgaben sollen in weiterer Folge als Auflistung dargestellt werden.

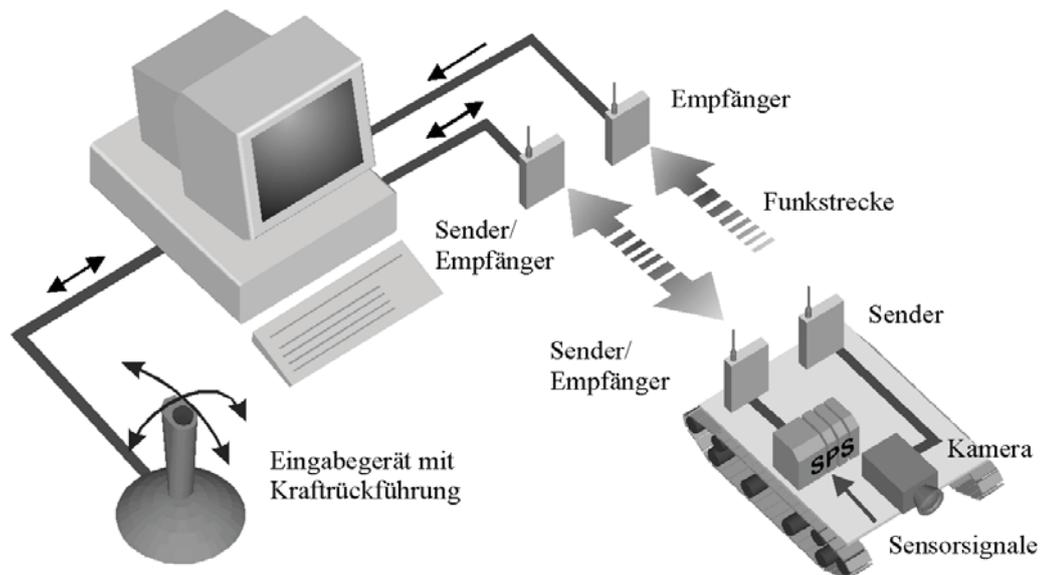


Abb. 2.2 Schematischer Systemaufbau

Das Teleoperatingsystem soll folgende Bestandteile beinhalten :

A. Joystick mit Krafrückführung

Als Eingabegerät soll ein Standard - Joystick mit Krafrückführungskomponenten verwendet werden. Ein geeignetes Abbildungsverfahren für die Krafrückführung muß hierfür gefunden und implementiert werden.

B. Steuer PC

Als stationäre Einheit für die Anbindung und Verknüpfung der einzelnen Hardwarekomponenten soll ein herkömmlicher PC verwendet werden.

C. Funkübertragung

Für die Videobildübertragung und die Übertragung der Daten sollen Funkstrecken getestet werden. Hierfür muß keine Hardware entwickelt werden. Es sollen fertige Komplettlösungen für die Funkstrecken verwendet werden.

D. Bildaufnahme mit der Möglichkeit die Orientierung der Kamera zu beeinflussen

Um den sichtbaren Bereich zu erweitern soll die Videokamera schwenkbar ausgeführt werden. Hierfür soll ein Kameraschwenksystem entwickelt werden, welches vom Ort der Bedienung in die gewünschte Position gedreht werden kann.

E. Fahrzeug und Fahrzeugsteuerung

Es soll ein vom Institut für Automation entwickeltes Kettenfahrzeug verwendet werden, welches mit einer eigenen speicherprogrammierbaren Steuerung (SPS) ausgestattet ist.

F. Sensoren

Für die haptische Rückführung zum Operator sollen Sensoren eingesetzt werden, welche die Kräfte am Fahrzeug messen. Dabei können sensorische Daten als virtuelle Kräfte modelliert werden. Hierzu zählen beispielsweise die Abstände zu Wänden.

G. Software für die Abbildung der Umgebung als Kraft, und zur Bedienung des Systems

Hierbei wird zwischen Echtzeit - Implementation auf einer SPS, und der Software auf dem PC unterschieden.

2.3 Versuchsaufbau

Die in Kapitel 2.2 beschriebenen Elemente des Teleoperatingsystems werden für Laborbedingungen umgesetzt. In Abb.2.3 und Abb.2.4 sind die Hardwarekomponenten des Versuchsaufbaues dargestellt.



Abb. 2.3 Arbeitsstation

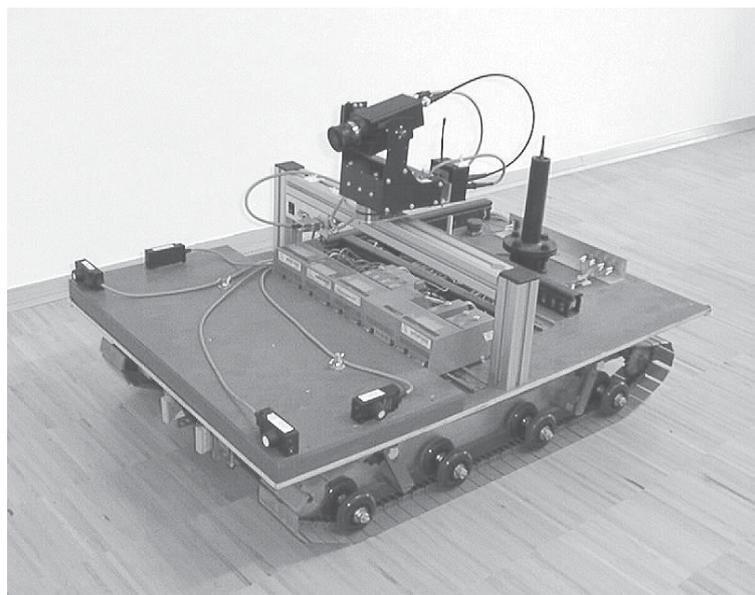


Abb. 2.4 Kettenfahrzeug

3 Eingabegerät

3.1 Einleitung

Für die manuelle Bewegungsvorgabe von automatisierten Fahrzeugen im industriellen Bereich werden meist Joysticks verwendet. Die Bedienung ist intuitiv, und bedarf keiner aufwendigen Einweisungen.

Für das vorliegende Projekt soll ein Joystick mit Krafrückführungsmöglichkeiten verwendet werden, um dem Bediener einen haptischen Eindruck zu übermitteln.

Force Feedback Joysticks (engl. Krafrückführung) für industrielle Anwendungen sind erhältlich. Da diese Joysticks nicht in Serienfertigung hergestellt werden, sind sie in der Anschaffung sehr kostspielig.

Als Eingabegerät für das Teleoperatingsystem soll als kostengünstige Alternative ein Standard – Joystick mit Krafrückführung verwendet werden. [3]

3.2 Aufbau des Force Feedback Joysticks

Für das Teleoperatingsystem wurde ein Joystick von der Firma Microsoft® verwendet. Es handelt sich um einen Analog - Joystick mit Krafrückführungsmöglichkeiten. Die Krafterzeugung erfolgt durch zwei Gleichstrommotoren, welche unabhängige Krafrückwirkungen in X- und Y – Richtung des Joysticks erzeugen können.

3.2.1 Allgemeine Beschreibungen

Fabrikat : **Microsoft Sidewinder Force Feedback Joystick**

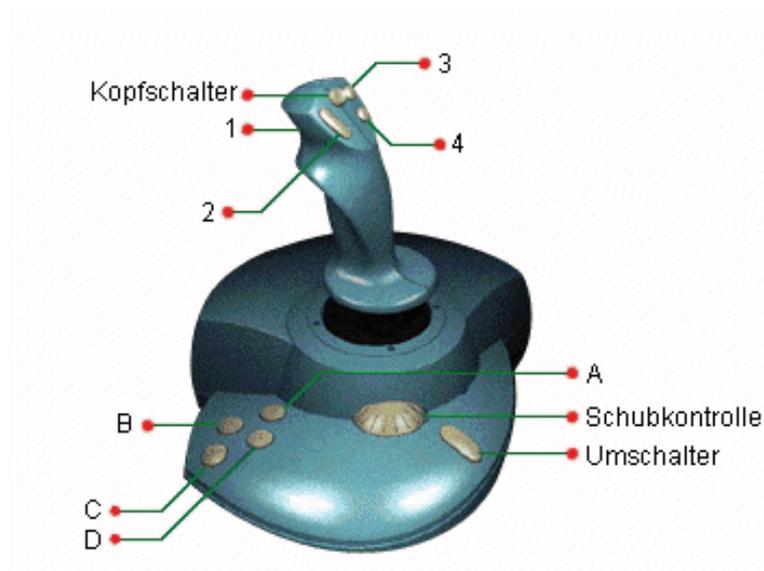


Abb. 3.1 Force Feedback Joystick

Der Joystick verfügt über insgesamt 9 Tasten, welche digitale Zustände annehmen können. Mit der Taste „Umschalter“ ist es möglich doppelte Belegungen der Tasten 1-4 und A-D zu erzielen.

3.2.1.1 Freiheitsgrade

Neben den Bewegungen des Griffes in X/Y – Richtung ist es möglich den Joystick um seine Längsachse zu schwenken (Abb.3.2). Diese zusätzliche Funktion wird für dieses Projekt nicht benützt, könnte aber beispielsweise dazu verwendet werden, das Kettenfahrzeug auf der Stelle zu wenden.

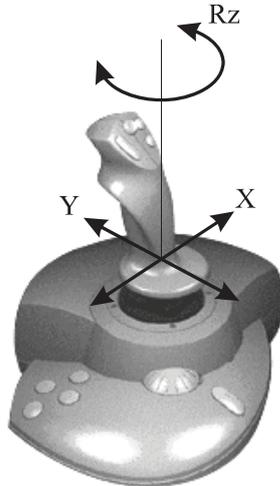


Abb. 3.2 Achsen des Joysticks

3.2.1.2 Joystickkoordinatensystem

Die Position des Joystickgriffes kann als Koordinatensystem wie folgt abgebildet werden :

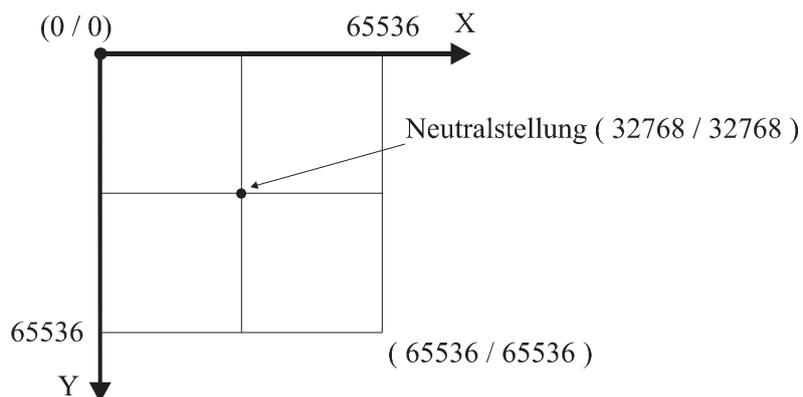


Abb. 3.3 Joystickkoordinatensystem

Der Koordinatenursprung befindet sich nicht in der Neutralstellung des Joysticks.

3.2.1.3 Kopfschalter

Der Kopfschalter läßt sich in zwei Achsrichtungen bewegen, und kann insgesamt 9 Positionen unterscheiden. Dieser digitale Richtungsschalter wird für die Bewegungen des Kameraschwenksystems verwendet.

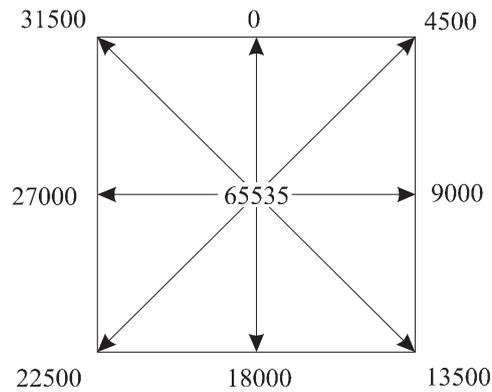


Abb. 3.4 Wertebereich des Kopfschalters

3.2.1.4 Schubregler

Der Schubregler wird für das Teleoperatingsystem verwendet um die maximale Geschwindigkeit des Kettenfahrzeuges zu beeinflussen.

Wertebereich	0 - 65535
--------------	-----------

3.2.1.5 Lichtschranke am Griff des Joysticks

Die Krafrückführung ist nur dann verfügbar, wenn eine Infrarotlichtschranke am Griff unterbrochen wird. Wird der Griff außerhalb der Neutralstellung losgelassen, so bleibt er in dieser Position stehen, da der Joystick über keine Federrückstellung verfügt. Dies muß bei der Steuerung des Kettenfahrzeuges bedacht werden, da sonst unkontrollierte Bewegungen die Folge wären.

3.2.2 Krafrückführungsmöglichkeiten

Der Joystick verfügt über ein großes Spektrum von Krafrückführungsmöglichkeiten und programmierbaren Krafteffekten.

Es wird daher an dieser Stelle nur ein Überblick über die Eigenschaften des Force Feedback Joysticks geboten. [9]

Bezüglich der Joystickposition unterscheidet man grundsätzlich zwischen :

- Positionsabhängige Krafteffekte
- Positionsunabhängige Krafteffekte

3.2.2.1 Positionsabhängige Krafteffekte

Je nach Auslenkung des Joysticks ergeben sich Funktionen für die Krafrückmeldung.

<i>Bezeichnung</i>	<i>Funktion</i>	<i>Bestimmung durch</i>
Feder	$f = K * x$	Federkonstante K, Position x
Dämpfer	$f = B * \dot{x}$	Dämpfungskonstante B, Geschwindigkeit \dot{x}
Reibung	$f = \mu * x$	Reibbeiwert μ , Position x
Beschleunigung	$f = m * \ddot{x}$	Masse m, Beschleunigungsvektor \ddot{x}
Wand	nicht vorhanden	Distanz, Wandneigung (0°,90°,180°,270°)

3.2.2.2 Positionsunabhängige Krafteffekte

Periodische Funktionen

- Sinusfunktion
- Rechteckfunktion
- Dreieckfunktion
- Sägezahnfunktion

Alle aufgeführten Effekte lassen sich in ihrer Dauer, Amplitude und Wirkrichtung verändern.

Nicht-periodische Funktionen

- Konstante Kraft
- Rampenfunktion
- Benutzerdefiniert

ROM / RAW-Force Effekte

- Der Joystick verfügt über vorprogrammierte Effekte im ROM
- Mit RAW-Force Effekten lassen sich die zuvor dargestellten Effekte kombinieren und individuell konfigurieren. Hierfür können Editoren verwendet werden, welche es zulassen Kraftverläufe grafisch zu editieren.

Für das Projekt wird vorläufig nur eine gerichtete linear – proportionale Kraft verwendet, welche aus einer nicht-periodischen Funktion gebildet wird.

4 Kameranahsystem

4.1 Einleitung

Die Position und Orientierung der Bildaufnahme ist bei Teleoperatingssystemen von großer Bedeutung. Bei feststehenden Kameraorientierungen ist das Blickfeld des Operators auf einen zuvor bestimmten, und nicht veränderbaren Bereich eingeschränkt. Dies wirkt sich nachteilig auf den visuellen Gesamteindruck und der Detailtreue aus.

Bei diesem Projekt soll die Möglichkeit bestehen, die Kameraorientierung am Kettenfahrzeug vom Leitstand aus zu verändern, um so ein größeres Blickfeld zu erhalten.

4.2 Gesamtkonzept

Im Rahmen dieses Projekts wurde ein Prototyp für ein Keraschwenksystem entwickelt, mit welchem es möglich ist, Schwenkbewegungen um zwei Achsen auszuführen.

Hauptbestandteile des Keraschwenksystems :

Videokamera

Für die Bildaufnahme wird eine Monochromkamera verwendet. Diese Kamera liefert am Ausgang ein analoges Videosignal nach CCIR – Norm (Kapitel 5), welches auf einem Monitor dargestellt werden kann.

Funkmodul für die Übertragung des Videosignals

Die Weiterleitung des Videosignals zum Leitstand erfolgt durch eine Funkstrecke. Hierfür wird ein Funkmodul verwendet, welches direkt am Keraschwenksystem montiert ist. Die Verbindung von Kamera und Funkmodul kann so auf direktem Wege erfolgen. Die Visualisierung des Videobildes kann entweder durch den Einsatz eines Fernsehbildschirmes, oder durch den Einsatz einer PC-Framegrabber-Karte erfolgen.

Schwenkmechanismus

Für dieses Projekt wurde ein Schwenkmechanismus entwickelt, welcher die Leistungsversorgung der Aktoren beinhaltet, und so zu einem aktiven Bauteil wird.

4.3 Aufbau des Kameranahwensystems

Das Kameranahwensystem weist zwei voneinander unabhangig steuerbare Schwenkachsen auf. Fur die Schwenkbewegungen wurden Servomotoren verwendet, welche reproduzierbare Winkelstellungen anfahren konnen. Die Steuersignale fur die Servomotoren werden dabei von der SPS geliefert. Die Hauptbestandteile des Kameranahwensystems sind an Abb.4.1 zu erkennen.

Hauptbestandteile des Kameranahwensystems

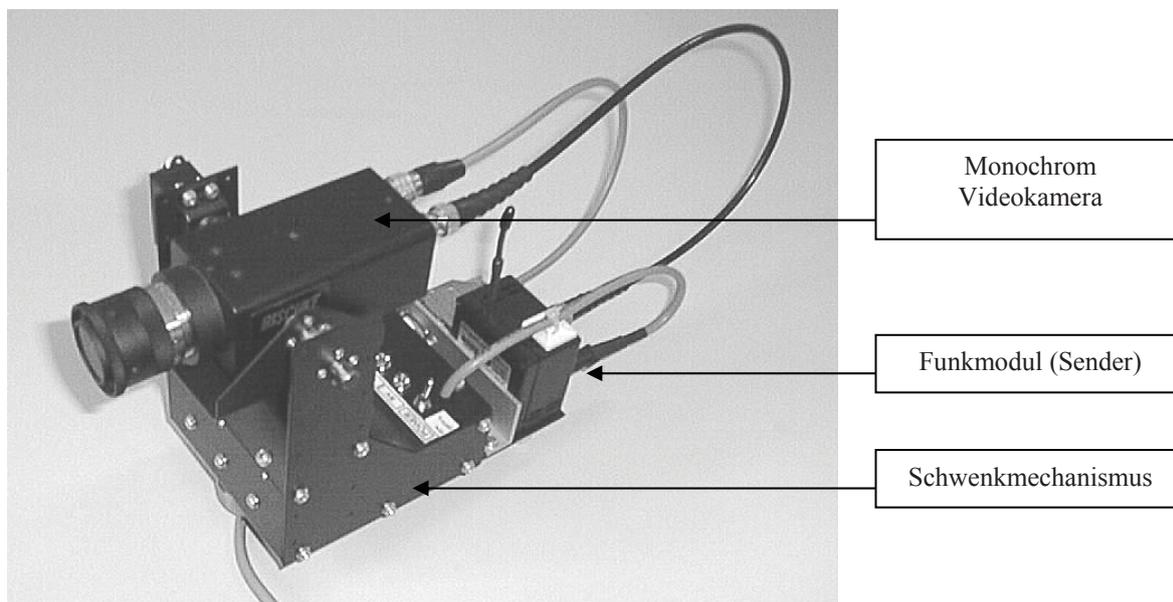


Abb. 4.1 Kameranahwensystem

Durch die Drehbewegungen des Kameranahwensystems ist die Zufuhrung von Versorgungsspannungen, Signalspannungen und Ladespannungen erschwert moglich.

Um Verwindungen von Kabelzufuhrungen zu verhindern, wurde das Funkmodul fur die Videosignalubertragung direkt auf dem Kameranahwensystem montiert.

4.4 Schwenkmechanik

Für die Drehbewegungen der Videokamera wurde ein Prototyp für einen Schwenkmechanismus entwickelt. Die Gehäusekonstruktion wurde dabei als geschraubte, im Laserschnittverfahren hergestellte Stahlblechkonstruktion ausgeführt.

Die mechanische Konstruktion wurde so ausgelegt, daß verschiedene Kameraausführungen montiert werden können.

Die verwendeten Servomotoren können ohne zusätzliche Getriebe oder Montageflansche an die Gehäusekonstruktion montiert werden.

Für die Funkübertragung des Videosignals befindet sich ein zuvor getestetes Funkmodul (Sender) auf dem Schwenkmechanismus.

4.4.1 Schwenkbereiche

In Abb.4.2 kann man die möglichen Schwenkbereiche der Achsen Rx und Rz erkennen.

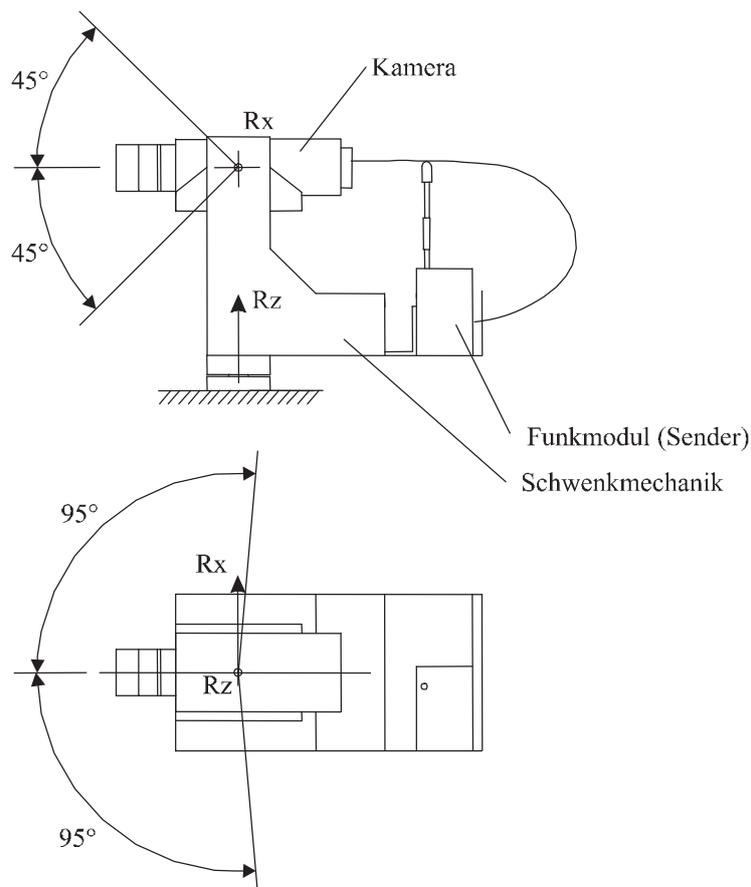


Abb. 4.2 Freiheitsgrade

4.4.2 Hauptlager

Für die Rotationsachse R_z wird ein Rillenkugellager mit Lebensdauerschmierung verwendet. Die Konstruktion wurde so ausgelegt, daß die gesamte Gewichtsbelastung durch das Kameraschwenksystem über dieses Lager übertragen wird.

4.4.2.1 Schwerpunktermittlung

Durch die komplexe Lastverteilung der elektrischen Komponenten, Verschraubungen, usw. ist eine genaue analytische Lösung der Schwerpunktlage nur durch einen hohen mathematischen Aufwand realisierbar.

Die Schwerpunktlage des Kameraschwenksystems wird daher durch einen Auflageversuch ermittelt.

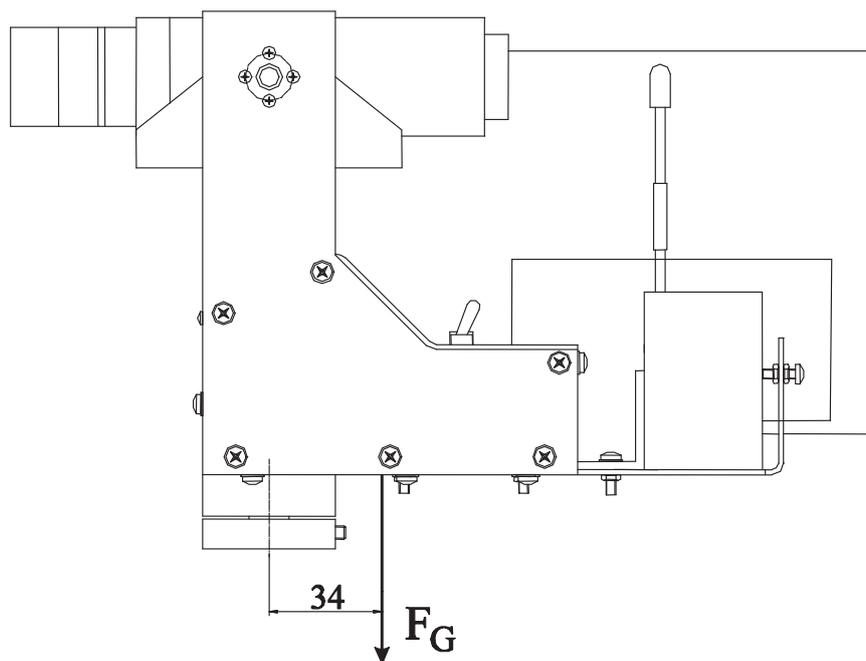


Abb. 4.3 Schwerpunktlage

Die vertikale Schwerpunktlage ist für Belastungsberechnungen nicht von Bedeutung. Gesamtmasse des Kameraschwenksystems $m = 2,24 \text{ kg}$

4.4.2.2 Ermittlung der Lebensdauer

Die gesamte zu übertragende Gewichtskraft des Keraschwenksystems wird durch das Rillenkugellager geleitet (Abb.4.4).

Der Bolzen dient der Drehmomentenübertragung von Servomotor 2 auf den ruhenden Montageflansch, und wird durch vertikale Kraftkomponenten nicht belastet.

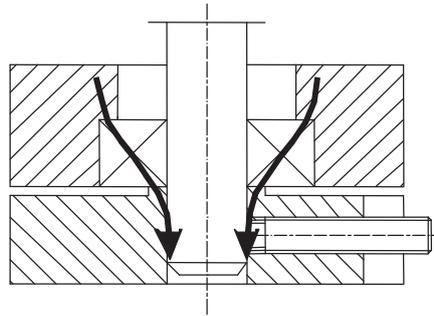


Abb. 4.4 Kraftübertragung

Die Gesamtmasse des Keraschwenksystems ($m = 2,24 \text{ kg}$) muß durch das Rillenkugellager in Form einer axialen Kraft übertragen werden.

Für das Hauptlager wird ein lebensdauergeschmiertes Rillenkugellager verwendet.

Rillenkugellager 608-2Z SKF

Dynamische Tragzahl $C = 3250 \text{ N}$

Statische Tragzahl $C_0 = 1370 \text{ N}$

Wenn ein Lager nicht umläuft, sondern lediglich Schwenkbewegungen um den Winkel $\pm \gamma$ aus der Mittellage heraus ausführt, dann gilt für die nominelle Lebensdauer [2] :

Nominelle Lebensdauer

Aus der Schwenkbewegung läßt sich die äquivalente Lebensdauer wie folgt berechnen.

$$L_{10\text{osz}} = \frac{180}{2 \cdot \gamma} \cdot L_{10} \quad (4.1)$$

γ Auslenkungswinkel um die Neutrallage des Keraschwenksystems [$^\circ$]

L_{10} Nominelle Lebensdauer [10^6 Umdrehungen]

Die Lebensdauer des Rillenkugellagers lässt sich verallgemeinert statisch berechnen.[2]

$$L_{10} = \left(\frac{C_0}{P_0} \right)^p \quad [10^6 \text{ Umdrehungen}] \quad (4.2)$$

- C_0 Statische Tragzahl [N]
 P_0 Äquivalente Lagerbelastung [N]
 p Lebensdauerexponent [-]

Äquivalente Drehzahl

Aus der erwarteten Oszillationsfrequenz n_{osz} [min^{-1}] lässt sich die äquivalente Drehzahl berechnen.

$$n = \frac{2 \cdot \gamma}{180} \cdot n_{\text{osz}} \quad [\text{min}^{-1}] \quad (4.3)$$

- γ Oszillationswinkel aus der Neutrallage [$^\circ$]

Lebensdauer in Betriebsstunden

$$L_{10h} = \frac{10^6}{60 \cdot n} \cdot L_{10} \quad [\text{h}] \quad (4.4)$$

Belastung des Rillenkugellagers

Laut Herstellerangaben sind axiale Belastungen von radialen Rillenkugellagern zulässig. Die Belastung in axialer Richtung sollte im allgemeinen den Wert $0,25 \cdot C_0$ nicht überschreiten.

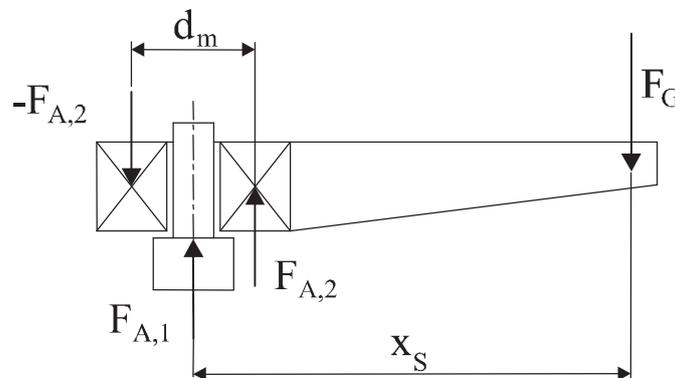


Abb. 4.5 Ersatzsystem für den Kraftangriff

Zusammensetzung der Axialkraft

Die maximale auf das Rillenkugellager übertragene Axialkraft setzt sich aus der Gewichtskraftkomponente F_G und der aus dem Moment resultierenden Axialkraftkomponente zusammen. (Abb.4.5)

$$F_{A,\max} = F_{A,1} + F_{A,2} \quad [\text{N}] \quad (4.5)$$

$$F_{A,1} = F_G \quad [\text{N}] \quad (4.6)$$

Momentengleichgewicht

$$F_G \cdot x_S = F_{A,2} \cdot d_m \quad [\text{N}] \quad (4.7)$$

d_m Mittlerer Lagerdurchmesser [mm]

x_S Schwerpunktabstand laut Abb.4.5 [mm]

Aus (4.7) ergibt sich nach Umformung

$$F_{A,2} = \frac{F_G \cdot x_S}{d_m} \quad [\text{N}] \quad (4.8)$$

Mittlerer Lagerdurchmesser

$$d_m = \frac{D + d}{2} \quad [\text{mm}] \quad (4.9)$$

D Außendurchmesser des Rillenkugellagers [mm]

d Innendurchmesser des Rillenkugellagers [mm]

Aus (4.9) ergibt sich durch einsetzen der Durchmesserwerte

Lageraußendurchmesser $D = 22$ [mm]

Lagerinnendurchmesser $d = 8$ [mm]

$$d_m = \frac{22 + 8}{2} = 15 \quad [\text{mm}] \quad (4.10)$$

Aus (4.7) läßt sich die aus dem Moment resultierende Axialkomponente $F_{A,2}$ berechnen.

Schwerpunktahstand $x_s = 34$ [mm]

Gewichtskraft $F_G = 22$ [N]

$$F_{A,2} = \frac{22 \cdot 34}{15} = 50 \quad [\text{N}] \quad (4.11)$$

Aus (4.5) und (4.6) errechnet sich die maximale Axialbelastung des Rillenkugellagers.

$$F_{A,\max} = F_{A,1} + F_{A,2} = 22 + 50 = 72 \quad [\text{N}] \quad (4.12)$$

Das Hauptlager wird durch eine vernachlässigbare radiale Kräfte belastet. Es ist somit möglich, die äquivalente Lagerbelastung P_0 der maximalen Axialbelastung gleich zu setzen.

$$P_0 = F_{A,\max} = 72 \quad [\text{N}] \quad (4.13)$$

Mit (4.2) erhält man für die nominelle Lebensdauer

Lebensdauerexponent $p = 3$ [-] für Kugellager

Statische Tragzahl $C_0 = 1370$ [N]

Lagerbelastung $P_0 = 72$ [N]

$$L_{10} = \left(\frac{C_0}{P_0} \right)^p = \left(\frac{1370}{72} \right)^3 = 6889 \cdot 10^6 \quad [\text{min}^{-1}] \quad (4.14)$$

Aus der Beziehung (4.3) läßt sich die aus der Schwenkbewegung resultierende äquivalente Drehzahl ermitteln.

Für die Oszillationsfrequenz wird die Annahme $n_{\text{osz}} = 3$ [min^{-1}] getroffen.

Oszillationswinkel $\gamma = 95^\circ$ Maximaler Auslenkungswinkel des Schwenksystems

$$n = \frac{2 \cdot \gamma}{180} \cdot n_{\text{osz}} = \frac{2 \cdot 95}{180} \cdot 3 = 3,2 \quad [\text{min}^{-1}] \quad (4.15)$$

Aus (4.1) und (4.14) erhält man

$$L_{10\text{osz}} = \frac{180}{2 \cdot \gamma} \cdot L_{10} = \frac{180}{2 \cdot 95} \cdot 6889 \cdot 10^6 = 6526 \cdot 10^6 \quad [\text{Oszillationen}] \quad (4.16)$$

Lebensdauererwartung des Rillenkugellagers

Aus (4.3) und (4.4) erhält man durch die Werte aus (4.15) und (4.16)

$$L_{10\text{h}} = \frac{10^6}{60 \cdot n} \cdot L_{10,\text{osz}} = \frac{10^6}{60 \cdot 3,2} \cdot 6526 \cong 3,4 \cdot 10^7 \quad [\text{Betriebsstunden}] \quad (4.17)$$

Erwartungsgemäß befindet sich die Lebensdauererwartung des axial belasteten Rillenkugellagers in sehr hohen Bereichen. Da aus konstruktiven Gründen kein kleineres Kugellager verwendet werden konnte, wird das stark überdimensionierte Hauptlager als geeignet erachtet.

4.4.3 Erforderliches Antriebsmoment

Für die Drehbewegung des Kameranahwensystems werden Servomotoren verwendet. Das erforderliche Antriebsmoment der Servomotoren erhält man aus dem Massenträgheitsmoment des gesamten Kameranahwensystems und der benötigten Winkelbeschleunigung.

Antriebsmoment der Servomotoren

Das Antriebsmoment errechnet sich aus :

$$M_{\text{Antr}} = J \cdot \ddot{\omega} \quad [\text{Nm}] \quad (4.18)$$

$$\begin{array}{ll} J & \text{Massenträgheitsmoment} \quad [\text{kgm}^2] \\ \ddot{\omega} & \text{Winkelbeschleunigung} \quad \left[\frac{\text{rad}}{\text{s}^2} \right] \end{array}$$

Die analytische Ermittlung des Massenträgheitsmoment ist durch die nicht unwesentlichen Beiträge durch Verkabelungen, elektrische Komponenten, usw. nur durch einen sehr hohen mathematischen Aufwand realisierbar.

Eine weitere Möglichkeit das Massenträgheitsmoment des Kameranahwensystems zu ermitteln, kann durch einen Pendelversuch erfolgen. Dabei wird das Schwenksystem auf einem bekannten Punkt fixiert, die Zeit für einen Schwingungszyklus experimentell ermittelt und auf den erforderlichen Rotationspunkt zurückgerechnet.

Winkelbeschleunigung des Kameranahwensystems

Das Kameranahwensystem soll nur sehr geringe Schwenkgeschwindigkeiten ausführen. Durch die mögliche schrittweise Vorgabe der Winkelstellung von Servomotoren sind Winkelbeschleunigungen gezielt programmierbar. Das Massenträgheitsmoment des gesamten Kameranahwensystems war zu Beginn der Entwicklungsarbeiten unbekannt, und somit nicht präventiv festzulegen.

Die eingesetzten Servomotoren wurden einem zuvor getesteten Prototypen entnommen und erwiesen sich beim Einsatz im Kameranahwensystem als geeignet.

Die verwendeten Servomotoren können einen maximalen Schwenkwinkel von 190° - bei einer Auflösung von 1024 Schritten – ausführen.

4.4.4 Antrieb der Schwenkachsen

Durch den mechanischen Aufbau der Servomotoren können die beweglichen Teile des Keraschwenksystems direkt angeflanscht werden.

Um die Biegemomentenbelastung und erhöhte Normalkraftbelastung bei der direkten Montage der Servomotoren zu verhindern, wurden die Achsen mit zusätzlichen Lagerstellen ausgestattet.

Das Gesamtgewicht des Keraschwenksystems der Achse R_z lastet auf einem axial belasteten Rillenkugellager. Die Kraftleitung der Achse R_z erfolgt dabei vom Gehäuse über das Rillenkugellager, wie dies in Abb.4.2 zu erkennen ist.

Die Drehachse R_x wurde einseitig mit einer zusätzlichen Gleitlagerstelle versehen.

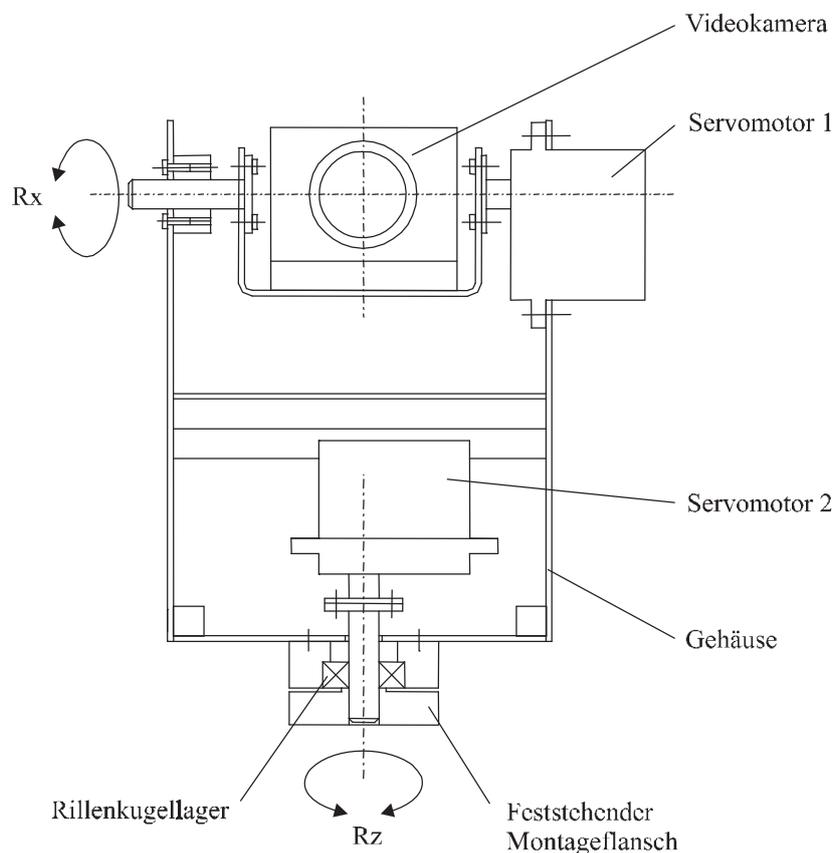


Abb. 4.6 Schwenkachsen mit Servomotoren

4.5 Bereitschaftsanzeigen des Keraschwenksystems

Die Leistungsversorgung der Servomotoren wird durch einen Hauptschalter am Gehäuse der Schwenkmechanik bereitgestellt. Bei abgeschaltetem Hauptschalter ist es möglich die Batterie im Inneren des Gehäuses aufzuladen.

Statusanzeige am Gehäuse :

Bezeichnung	Bedeutung
5V	Leistungsversorgung der Servomotoren
12V	Spannungsversorgung der Kamera und des Sendemoduls für die Videoübertragung
LOAD	Ladevorgang 4,8 V Batterie (Nur im Zustand Off/L des Hauptschalters möglich)

Tabelle 4.1 Statusanzeigen der Leuchtdioden

4.6 Signal- und Leistungsflüsse zum Keraschwenksystems

Die notwendigen Signal- und Leistungsflüsse für die Steuerung und Leistungsversorgung des Keraschwenksystems lassen sich in folgendem Diagramm schematisch darstellen.

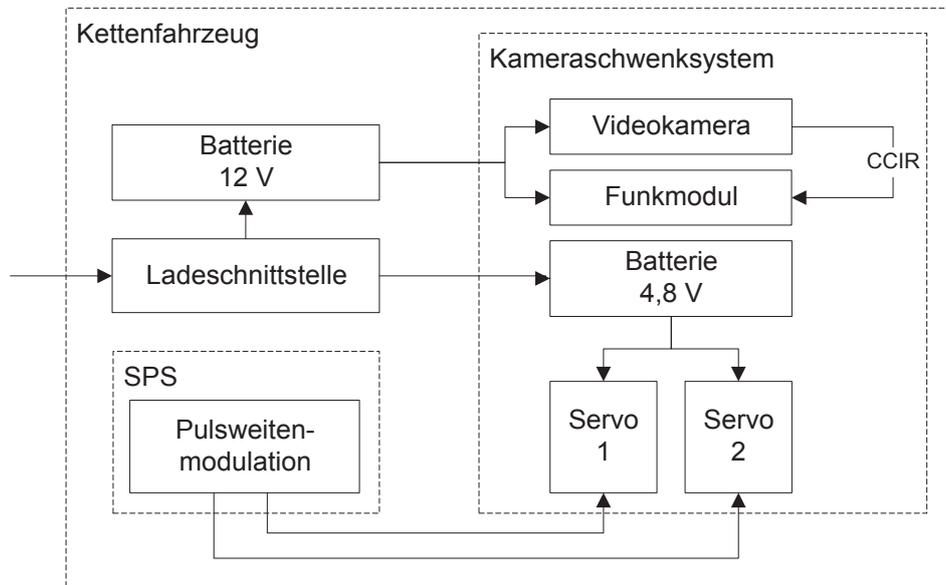


Abb. 4.7 Signalflußdiagramm

Die Versorgung des Keraschwenksystems mit den notwendigen Steuersignalen und der Spannungsversorgung von Kamera und Funkmodul wird durch eine mehradrige Hauptverbindungsleitung bewerkstelligt. Diese Verbindungsleitung wird aus dem Gehäuse so ausgeleitet, daß Verwindungen des Kabels nicht möglich sind.

4.7 Innerer Aufbau des Keraschwenksystems

Im Gehäuse des Keraschwenksystems befinden sich folgende Komponenten :

Servomotoren

Antrieb der beiden Schwenkachsen des Keraschwenksystems

Platine

Für die Signalaufbereitung und Weiterleitung befindet sich im Gehäuse des Keraschwenksystems eine elektrische Schaltung.

Batterieversorgung

Für die Leistungsversorgung der Servomotoren befindet sich ein aufladbarer NiCd – Batterieblock im Inneren des Gehäuses

4.8 Batterieversorgung

Für die Leistungsversorgung der Servomotoren befindet sich im Gehäuse des Keraschwenksystems ein aufladbarer NiCd – Batterieblock.

Nennspannung	[V]	4,8
Akkukapazität	[mAh]	1400
Gewicht	[g]	128
Abmessungen	[mm]	51 x 69 x 17

Ladung

Laut Herstellerangaben soll die Ladung des Akkus mit 140 mA für die Dauer von 14h erfolgen.

4.9 Servomotor

Für die Drehbewegung der Achsen am Keraschwenksystem werden Servomotoren verwendet. Servomotoren sind elektromechanische Stellglieder, welche ein elektrisches Signal in eine mechanische Bewegung umsetzen. Diese Motoren wurden speziell für den Einsatz im Modellbau entwickelt und weisen daher verhältnismäßig geringe Baugrößen und Leistungen auf.

Die wesentlichen Vorteile derartiger Motoren liegen in der direkten Vorgabe absoluter Schwenkwinkel und der einfachen Montage.

Der Stellwinkel des Servomotors wird über ein Potentiometer auf der Servowelle aufgenommen und mit dem Sollwert verglichen. Die hierfür notwendige elektronische Schaltung und das Potentiometer befinden sich im Inneren des Servomotors.

Die Drehbewegung der eingesetzten Servomotoren ist auf einen maximalen Schwenkbereich eingeschränkt. (Siehe Tabelle 4.3)

Kennwerte des verwendeten Servomotors :

Bezeichnung : Futaba SERVO S3003

Max. Schwenkwinkel	[°]	190
Auflösung	[Schritte]	1024
Stellmoment	[Ncm]	32
Nennspannung	[V]	4,8
Betriebsspannung	[V]	4,8 -6
Gewicht	[g]	37
Abmessungen	[mm]	40,4 x 19,8 x 36,0

Tabelle 4.2 Kennwerte des Servomotors

Derartige Servomotoren sind nur in gekapselter Ausführung erhältlich. Durch diesen Aufbau können diese Motoren bei rauen Umgebungseinflüssen für geringe Stellkräfte eingesetzt werden. Intern verfügen diese Motoren über eine Getriebeübersetzung aus Kunststoff oder Stahlzahnäder.

4.10 Pulsweitenmodulation

Die Vorgabe des Stellwinkels des Servomotors erfolgt durch ein Steuersignal in rechteckform. Durch die Veränderung der Pulsweite des Rechtecksignals wird der Schwenkwinkel des Servomotors vorgegeben. Dies erfolgt durch den Einsatz eines Moduls für Pulsweitenmodulationen auf der SPS.

Spezifikation des Steuersignals

Für die Vorgabe des absoluten Schwenkwinkels der Servomotoren ist folgendes Rechteck - Steuersignal erforderlich :

Frequenz	[Hz]	58
Amplitude	[V]	5
Zeitbereich t1	[ms]	1
Zeitbereich t2	[ms]	2

Tabelle 4.3 Spezifikation des Steuersignals

Im angegebenen Zeitbereich von t1 bis t2 (Abb.4.8) ist es möglich 1024 Winkelschritte des Servomotors vorzugeben. Die direkte Verwendung von digitalen Ausgängen der SPS ist nicht möglich, da die geringste Task-Zeitklasse 10 ms beträgt. Das bedeutet es können Wertänderung nur alle 10 ms erfolgen. Es ist daher notwendig ein spezielles Modul für pulswellenmodulierte Signale zu verwenden.

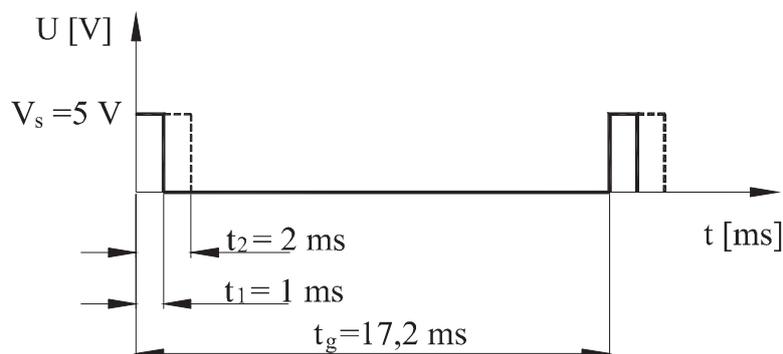


Abb. 4.8 Steuersignal

4.10.1 SPS – Modul für Pulsweitenmodulation

Für die Bereitstellung der erforderlichen pulswertenmodulierten Signale für die Servomotoren wird das Anpassungsmodul DO135 für die SPS 2003 von B&R verwendet.

Mit diesem Modul ist es möglich 4 unabhängige pulswertenmodulierte Signale zu erzeugen. Die Pulsweite des Rechtecksignals kann mit einer Auflösung von 250 ns eingestellt werden. Der erforderliche Zeitbereich von 1ms für die Ansteuerung der Servomotoren kann somit in 4000 Schritte unterteilt werden. Der verwendete Servomotor kann 1024 Winkelschritte auflösen.

Mit dem Modul DO135 ist es möglich die Zeitdauer des High-Signals und die Dauer des Low-Signals unabhängig voneinander vorzugeben.

Vorgabe der Pulsweite

Die Vorgabe der Pulsweite erfolgt pro Kanal durch zwei Variablen. Diese Variablen liegen im LONG-Wertebereich der SPS, und repräsentieren die Anzahl der Ticks in 250 ns Schritten.

High	Low	Schwenkwinkel Servomotor
1000	64966	0°
10000	55966	190°

Tabelle 4.4 Wertebereiche

5 Videokamera

Die Bildaufnahme des Handhabungsortes erfolgt durch eine monochrome CCD – Flächenkamera. Das Bild wird dabei in einzelne Bildpunkte zerlegt, welche durch lichtempfindliche MOS (Metall Oxide Semiconductors) Halbleiterelemente aufgenommen werden. Diese Halbleiterelemente befinden sich auf einem Flächensensor, und bilden ein sog. CCD-Array mit einer genormten Anzahl von Bildpunkten. Die unterschiedlichen Helligkeitswerte der einzelnen Bildpunkte werden durch MOS-Kondensatoren in Spannungen umgewandelt und erzeugen so das Videosignal nach CCIR – Norm.

CCIR¹ – Norm

Durch die Trägheit des menschlichen Auges erscheint dem Betrachter ein stehendes Bild bei einer Bildwiederholungsrate von 50 Bildern pro Sekunde. Dies erfordert einen hohen Aufwand für Verstärker und Übertragungseinrichtungen. Um den Aufwand zu senken werden bei der CCIR – Norm zwei Verfahren unterschieden :

- Vollbildmodus
- Halbbildmodus

Im Vollbildmodus werden 25 Vollbilder pro Sekunde aufgenommen und übertragen. Beim Halbbildmodus wird das Bild in ungerade Zeilen und gerade Zeilen unterteilt (Interlaced Modus). Es können so 50 Halbbilder pro Sekunde aufgenommen werden, welche am Empfangsort zusammengesetzt 25 Vollbilder pro Sekunde ergeben. Es erfolgt so eine Täuschung des menschlichen Auges, und der Betrachter hat den Eindruck eines stehenden Bildes.

Die Zahl der vollständig übertragenen Bilder (Frames) je Sekunde wird als *Bildwechselfrequenz* bezeichnet.

¹ Comité Consultatif International des Radiocommunication

Eingesetzte Videokamera

Die verwendete Videokamera kann für Bildaufnahmen im industriellen Bereich eingesetzt werden. Da die Anschlußmaße derartiger Kameras genormt sind, können auch andere Kameraausführungen auf dem Schwenksystem montiert werden.

Es wurden Versuche für die Bildaufnahme des Handhabungsortes des Kettenfahrzeuges und die Übertragung zum Leitstand des Teleoperatingsystems vorgenommen. Die visuelle Darstellung des Videosignals erfolgt für diese Zwecke auf einem Fernsehbildschirm.

Durch den Einsatz einer PC - Framegrabber Karte ist es möglich, das Videosignal auf dem stationären PC darzustellen, um eine Bildverarbeitung vorzunehmen.

Autonome Bewegungen des Kettenfahrzeuges können durch Bildverarbeitung realisiert werden.

Bei dieser Videokamera können nur Objektive mit Fixbrennweiten verwendet werden.

Kamera	<i>Produktname</i> : Bischke CCD-4000 LL
Auflösung	752 x 582 Bildpunkte
Bildaufnahme	CCIR - Norm
Versorgungsspannung	12 V
Objektiv	<i>Produktname</i> : Computar
Brennweite	6 mm
Lichtempfindlichkeit	1 : 1.2

Tabelle 5.1 Verwendete CCD-Kamera

6 Videofunksystem

Die Funkstrecke für die Videosignalübertragung besteht aus dem mobilen Sendermodul auf dem Keraschwenksystem und einem Empfängermodul auf der stationären Leitstandseite.

Es wurde speziell für den Einsatz im Laborbereich ein Funksystem besorgt und getestet.

Mit dem verwendeten Funksystem können Monochrom- und Farbvideosignale übertragen werden.

Die Sendeleistung beträgt 10mW, und kann in Gebäuden Distanzen bis 30 m überwinden.

Bei Anwendungen außerhalb von Gebäuden können Reichweiten bis 300 m erreicht werden.

Durch die Wellenlänge der elektromagnetischen Wellen können Mauern und andere nichtmetallische Hindernisse überwunden werden.

Die Reichweite der Funkstrecke wird stark von der Beschaffenheit der Wände, und den elektromagnetischen Störeinflüssen beeinträchtigt. Zu den elektromagnetischen Störeinflüssen können beispielsweise Computermonitore gezählt werden.

Die Funkübertragung erfolgt im 2,4 GHz Bereich. Der zur Verfügung stehende Frequenzbereich wurde vom Hersteller in 5 Kanäle unterteilt. Durch Kanalumschaltung besteht die Möglichkeit, Videosignale mehrerer Funksender zu einer Empfangseinheit zu leiten.

Die besten Übertragungsergebnisse werden laut Herstellerangaben in einer Aufstellhöhe der Funkmodule von 1,5 m bis 2 m erreicht.

7 Datenfunkstrecke

Um eine intuitive Bewegungsvorgabe des Kettenfahrzeuges zu gewährleisten, muß eine Mindestgeschwindigkeit für die Datenübertragung zwischen SPS und stationärem PC vorliegen.

Geplant war die Verwendung einer seriellen Datenübertragung in Form einer RS232 Verbindung. Für Testzwecke wurden fertige Funkmodule besorgt, welche als bidirektionaler Leitungsersatz für eine RS232 – Kabelverbindung verwendet werden können.

Derartige Funkmodule sind in ihrer Übertragungsgeschwindigkeit durch die Umschaltung zwischen Sende- und Empfangsmodus sehr eingeschränkt. Das getestete Modell konnte eine maximale Übertragungsgeschwindigkeit von 11 kbaud erreichen. [5]

Versuche ergaben, daß selbst mit Übertragungsraten von 57600 baud keine intuitive Steuerung des Kettenfahrzeuges realisiert werden konnte. Daher wurde die Variante der Funkdatenübertragung über eine RS232 Verbindung verworfen.

Für Untersuchungen am Teleoperatingsystem wurde statt der RS232-Funkverbindung eine RS232-Leitungsverbindung verwendet.

Als alternative Datenübertragung zwischen der mobilen SPS und dem PC wird derzeit vom Institut für Automation eine Funk-Ethernet Verbindung getestet.

Auf dem Kettenfahrzeug wird hierfür ein zusätzlicher Industrie - PC installiert, welcher die notwendigen Daten der SPS aufbereitet, und über die Funk-Ethernet-Verbindung weiterleitet. So lassen sich sehr schnelle Datenübertragungsraten realisieren, welche eine intuitive Bedienung des Teleoperatingsystems gewährleisten.

8 Kettenfahrzeug

8.1 Einleitung

Für den Versuchsaufbau wurde ein vom Institut für Automation entwickeltes Kettenfahrzeug verwendet. Das Fahrzeug wird ausschließlich für Laborversuche verwendet, und mit Komponenten - je nach Anwendungsfall - erweitert. Für das Teleoperatingsystem mußte das Kettenfahrzeug mit einer Videoaufnahme und zusätzlichen Sensoren ausgestattet werden.(Abb.8.1)

Für die Veränderung der Kameraorientierung wurde für dieses Projekt ein Schwenksystem entwickelt, welches zusätzlich auf dem Fahrzeug montiert wurde. (Kapitel 4)

Vom Leitstand des Teleoperatingsystems ist es mittels Joysticksteuerung möglich das Kettenfahrzeug zu bedienen und die Kameraorientierung zu verändern.

Abmessungen des Kettenfahrzeuges

Grundfläche gesamt	850 x 600 mm
Gesamthöhe	500 mm
Kamerahöhe	450 mm

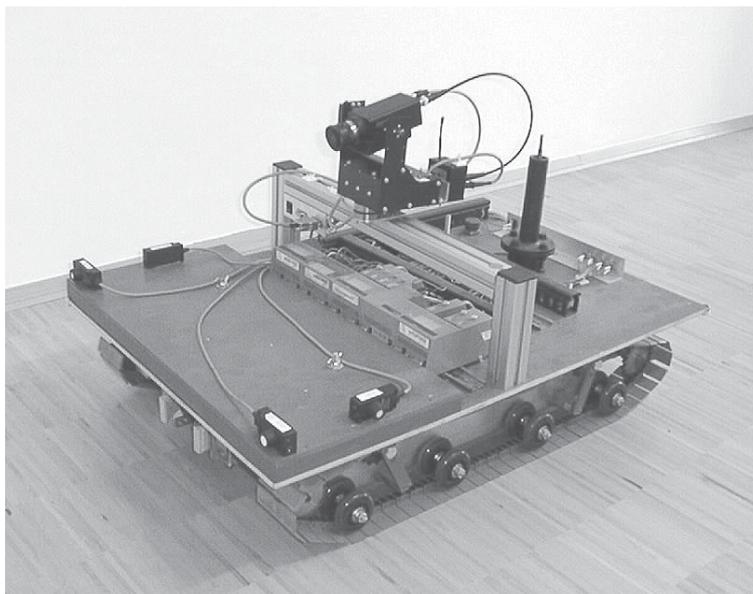


Abb. 8.1 Kettenfahrzeug

Leistungsmerkmale des Kettenfahrzeuges

Das Fahrzeug wird durch zwei bürstenlose Gleichstrom - Getriebemotoren angetrieben, welche durch Bleiakkumulatoren versorgt werden.

Die aufladbaren Akkumulatoren dienen der Spannungsversorgung der Antriebsmotoren und der auf dem Fahrzeug befindlichen SPS.

Für die Versorgung der Videokamera und der Videofunkverbindung wurde ein zusätzlicher 12 V Bleiakkumulator im Inneren des Kettenfahrzeuges eingebaut.

Das Fahrzeug wurde auf eine Maximalgeschwindigkeit von 0,5 m/s ausgelegt. Diese Geschwindigkeit wird bei Versuchen jedoch nicht erreicht, da bereits ab ca. 60% der Maximalgeschwindigkeit keine sinnvolle Bewegungsvorgabe des Kettenfahrzeuges mehr realisiert werden kann.

Betriebselektronik

Die Ansteuerung der Getriebemotoren erfolgt mittels Betriebselektronik mit der Bezeichnung Variotronic VT 2501, welche die Steuersignale von der SPS erhält. Die notwendigen Signale für die Antriebsmotoren sind Kapitel 11 zu entnehmen.

Die Funktion der Betriebselektronik ist der technischen Dokumentation dieses Produktes zu entnehmen.

Leistungsdaten der Antriebsmotoren

Nennspannung	24V
Nennstrom	1,85 A
Untersetzung	16:1
Nennmoment	0,8 Nm
Drehzahlbereich	22 ...215 U/min
Masse	0,54 kg
max. Wellenbelastung radial	60 N
max. Wellenbelastung axial	30 N

Bleiakkumulatoren

Für die Spannungsversorgung der beschriebenen Hardwarekomponenten wurden zwei Bleiakkumulatoren der Firma Sonnenschein, Baureihe Dryfit A500 verwendet.

Abgabespannung	12 V
Maximaler Entladestrom	80 A
Entladestrom für 20 Stunden	325 mA

9 Ultraschallsensor

9.1 Einleitung

Für die Vermessung von Abständen des Kettenfahrzeuges zu Hindernissen wurden für das Teleoperatingsystem Ultraschallsensoren verwendet. Über die Laufzeit der vom Sensor ausgesandten Schallwellen wird die Distanz zu Objekten ermittelt. Der Sensor unterscheidet somit zwischen Sende- und Empfangsmodus. Für das Projekt wurden Ultraschallsensoren verwendet, welche ein analoges Ausgangssignal liefern. Dieses Ausgangssignal ist dem Abstand zu Objekten proportional. Der Meßbereich des verwendeten Sensors kann manuell auf die Einsatzbedingungen justiert werden.

Vorteil von Ultraschallsensoren

Schallwellen werden an harten Oberflächen gut reflektiert. Die zu erfassenden Objekte können fest, flüssig, körnig oder pulverförmig sein. Durchsichtige und andere optisch schwer erfaßbare Objekte werden mit Ultraschallsensoren eindeutig erkannt.

Nachteil von Ultraschallsensoren

Die Ultraschallwellen werden an Objekten rückgespiegelt und vom Sensor im Empfangsmodus detektiert. Ist ein zu erfassendes Objekt geneigt, kann der Fall eintreten, daß die Schallwellen derart reflektiert werden, daß sie vom Sensor nicht erfaßt werden können. Bei kleinen Öffnungswinkeln des Schallkegels kann dieser Fall eintreten, und es kann trotz geringem Abstand zu Hindernissen keine Erkennung erfolgen.

Technische Daten**Fabrikat :** Baumer UNDK 30U9103

Erfassungsbereich	200...700mm
Öffnungswinkel	10°
Schaltfrequenz	230 kHz
Betriebsspannungsbereich	15 – 30 VDC
Ausgangsspannung	0 - 10 V
Auflösung	<0,3 mm
Stromverbrauch	30 mA
Reaktionsszeit	< 100 ms

Für das Teleoperatingsystem wurden 4 Stück dieser Ultraschallsensoren verwendet, um Abstände des Kettenfahrzeuges zu Hindernissen zu detektieren.

Der Einsatz der Ultraschallsensoren für Krafrückführungen im Teleoperatingsystem wird in Kapitel 11 erläutert.

10 Stationäre PC-Einheit

In Kapitel 2 ist die Arbeitsstation des Teleoperatingsystems dargestellt. Für die Programmierarbeiten und den Einsatz für das Teleoperatingsystem wurde ein Standard PC verwendet. Alle Versuche mit Visualisierungen wurden durch einen Fernsehbildschirm durchgeführt, da die Anbindung einfacher, und die Funktionen Bedienung und Kontrolle der Daten klar voneinander zu trennen sind. In diesem Zusammenhang sei darauf verwiesen, daß das Videobild auf der Bedienungsfläche des Teleoperatingsystems sehr einfach durch den Einsatz einer Framegrabber-Karte implementiert werden kann.

Leistungsdaten des stationären PC

Monitor	17"
Prozessortyp	Pentium 200 MHz
Hauptspeicher	48 MB
Festplattenkapazität	5,2 GB
Betriebssystem	Windows95®
Programmierungsumgebung	Visual Basic 5.0® Visual C++ 5.0® B&R PG2000 V2.35

Der Datentransfer zwischen PC und mobiler SPS erfolgt über die serielle Schnittstelle COM 2 mit einer Datenübertragungsrate von 57600 Baud.

Der Anschluß des Force Feedback Joysticks erfolgt über die MIDI-Schnittstelle einer Soundkarte.

11 Abbildungsverfahren

11.1 Abbildung der Motorgeschwindigkeiten zur Joystickposition

11.1.1 Einleitung

Um die Fahrtrichtung und Geschwindigkeit des Kettenfahrzeuges vorzugeben, soll in diesem Fall eines Teleoperating - Systems ein Joystick verwendet werden. Die Steuerung mittels Joystick ist im Gegensatz zu anderen Eingabemöglichkeiten intuitiv und leicht zu erlernen.

Im Vergleich zu Fahrzeugen auf 4 Rädern – als Beispiel sei hier ein Personenkraftwagen erwähnt - ist bei Kettenfahrzeugen nicht die Bewegungsrichtung und getrennt davon die Geschwindigkeit zu wählen. Bei Kettenfahrzeugen wird die Fahrtrichtung durch unterschiedliche Laufgeschwindigkeiten der beiden Ketten vorgegeben. Betrachtet man den Fall die Antriebsketten haben gegenläufige Bewegungsrichtungen so kann das Fahrzeug auf der Stelle gewendet werden – dies ist bei Personenkraftwagen nicht möglich. Die Laufgeschwindigkeiten der beiden Ketten können meist vom Stillstand bis zur maximalen Laufgeschwindigkeit stufenlos gesteigert werden. So können Fahrtgeschwindigkeit und Fahrtrichtung zwar gewählt werden, sie sind jedoch nicht von einander zu trennen.

Der Bezug zwischen den Geschwindigkeiten und Laufrichtungen der Antriebsketten zur Joystickstellung werden in diesem Kapitel auf theoretischer Ebene erläutert. Es soll eine Lösung zur Implementierung in ein Steuerungssystem geboten werden. Dabei wird nicht die Implementierung einer Steuerung auf programmiertechnischer Ebene, sondern die theoretischen Hintergründe und mathematische Ansätze erklärt.

11.1.2 Eingabegerät

Als Eingabegerät für das Teleopeartingsystem soll der in Kapitel 3 beschriebene Force Feedback Joystick der Firma Microsoft verwendet werden. Für weitere Betrachtungen werden die Wertebereiche dieses Joysticks verwendet, um die konkrete Implementation für die Abbildung der Motorgeschwindigkeiten zu erläutern. An dieser Stelle soll darauf verwiesen werden, daß der Einsatz eines herkömmlichen analogen Joysticks die selbe Implementation mit unterschiedlichen Wertebereichen liefern würde.

Joystickkoordinatensystem

Das kartesische Koordinatensystem des implementierten Joysticks wird in Abb.11.1 dargestellt.

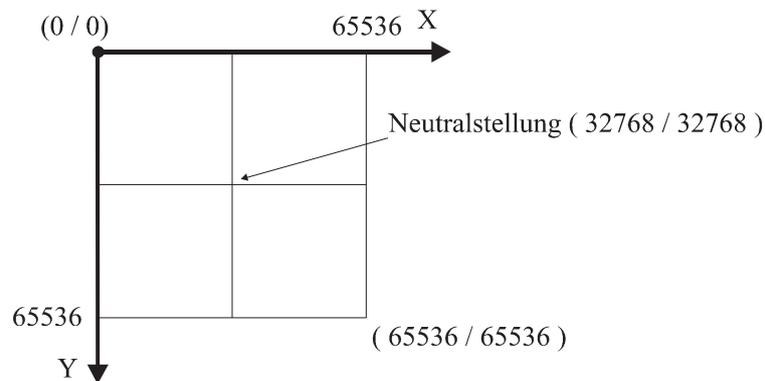


Abb. 11.1 Joystickkoordinatensystem

Der Koordinatenursprung befindet sich in der linken oberen Joystickstellung. Die mechanische Neutralstellung des Joysticks liegt bei den Koordinatenwerten 32768 in X-Richtung und 32768 in Y-Richtung.

11.1.3 Ansteuerung der Antriebsmotoren

In diesem Kapitel sollen die erforderlichen Variablen und deren Wertebereiche beschrieben werden, welche für die Vorgabe der Drehrichtung und Geschwindigkeiten der Antriebsmotoren erforderlich sind.

Die Vorgabe der Geschwindigkeit und Drehrichtung der verwendeten Antriebsmotoren erfolgt durch setzen von drei Ausgangssignalen auf der SPS. Diese Ausgangssignale werden durch Variablen auf der Softwareebene der SPS repräsentiert.

Für die Abbildung der Motorgeschwindigkeiten zu den Joystickkoordinaten sind die genauen Variablenbezeichnungen nicht von Bedeutung. In Tabelle 11.1 werden die notwendigen Ausgangssignale für die Ansteuerung eines Antriebsmotors dargestellt. Darauf aufbauend können die Motorgeschwindigkeiten bezüglich der Joystickposition abgebildet werden.

Bezeichnung	Wertebereich	Ausgangssignal
Geschwindigkeit	0 - 32768	analog
Richtungsbit 1	0 / 1	digital
Richtungsbit 2	0 / 1	digital

Tabelle 11.1 Wertebereich der verwendeten Variablen

Durch die Kombinationsmöglichkeiten von Richtungsbit 1 und Richtungsbit 2 sind verschiedene Eigenschaften der Antriebsmotoren einstellbar. In Tabelle 11.2 werden die Möglichkeiten für die Ansteuerung eines Antriebsmotors betrachtet.

Richtungsbit 1	Richtungsbit 2	Eigenschaft
1	0	Motordrehrichtung 1
0	1	Motordrehrichtung 2
0	0	Motor halt
1	1	Aktive Motorbremse

Tabelle 11.2 Eigenschaften der Richtungsbit

11.1.4 Abbildung der Motorgeschwindigkeiten

Für die Zusammenhänge der Motorgeschwindigkeiten zur Joystickstellung ergeben sich verschiedene Ansatzmöglichkeiten. Durch die Verwendung eines Joysticks als Eingabemedium, muß die Geschwindigkeits- und Drehrichtungsvorgabe für zwei Antriebsmotoren auf das vorhandene Joystickkoordinatensystem abgebildet werden.

Voraussetzungen und Vorgaben

- Das Kettenfahrzeug soll durch den Joystick auf der Stelle gewendet werden können.
- In der mechanischen Neutralstellung des Joysticks soll sich das Kettenfahrzeug in Ruhe befinden.
- Mit größer werdender Auslenkung des Joysticks aus der Neutrallage soll die Fahrgeschwindigkeit in alle Bewegungsrichtungen vergrößert werden können.

11.1.4.1 Intuitive Bedienung

Bei der Auslenkung des Joysticks aus der Neutrallage sollte sich das Kettenfahrzeug wie in Abb.11.2 dargestellt verhalten. Dies würde eine intuitive Bewegungsvorgabe des Kettenfahrzeuges ergeben. Aufgrund der erforderlichen Symmetrie wird nur eine Halbebene des Joystickkoordinatensystems und die Auswirkung auf die Bewegung des Kettenfahrzeuges betrachtet.

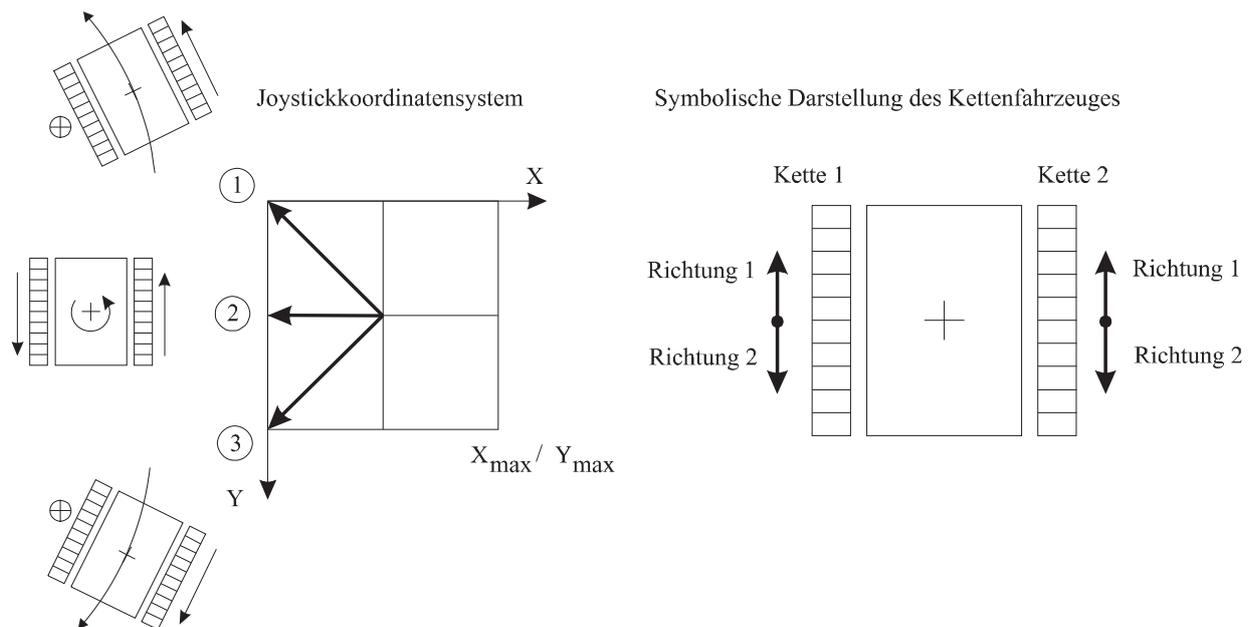


Abb. 11.2 Bewegungsverhalten des Kettenfahrzeuges

Je nach Auslenkung des Joysticks aus der Neutrallage sollen sich proportionale Motorgeschwindigkeiten ergeben. Für das Verhältnis der Joystickkoordinaten zu den Motorgeschwindigkeiten wird der Fall der maximalen Auslenkung des Joysticks entlang der Joystickstellungen 1-3 betrachtet (Abb.11.2). Entlang dieser Joystickpositionen ergeben sich kritische Bedingungen für die Antriebsgeschwindigkeit von Kette 2. (Abb.11.3)

Die Antriebsgeschwindigkeit von Kette 1 wird in diesem Zusammenhang nicht betrachtet, da diese entlang der Joystickposition 1-3 keine kritischen Zustände annimmt.

11.1.4.2 Abbildungsverfahren 1

Entlang der Joystickpositionen 1-3 in Abb.11.3 soll die Maximalgeschwindigkeit von Kette 2 in beide Drehrichtungen vorliegen, um den intuitiven Bewegungsablauf zu erzielen.

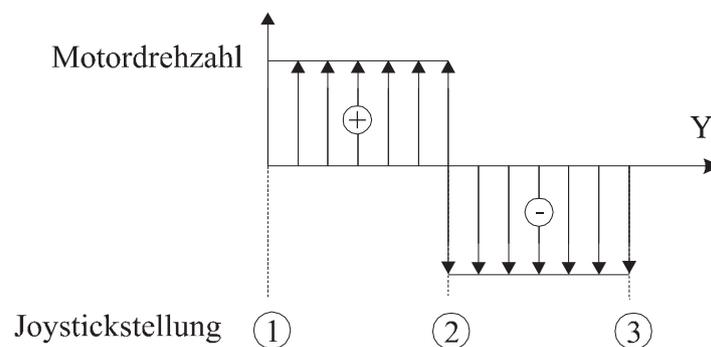


Abb. 11.3 Geschwindigkeitsverlauf von Kette 2

An Abb.11.3 kann man erkennen, daß bei Joystickposition 2 von positiver Maximalgeschwindigkeit auf negative Maximalgeschwindigkeit gewechselt wird. Dies ergibt sehr große Belastungen des Antriebsmotors der Kette 2.

Diese Möglichkeit für die Bewegungsvorgabe des Kettenfahrzeugs wird nicht weiter betrachtet. Versuche ergaben, daß das Kettenfahrzeug mit diesem Abbildungsverfahren nur bedingt gesteuert werden kann.

11.1.4.3 Abbildungsverfahren 2

Mit dem Verzicht auf die Möglichkeit das Kettenfahrzeug auf der Stelle zu wenden, kann folgender Geschwindigkeitsverlauf für Kette 2 entlang der Joystickpositionen 1-3 dargestellt werden.

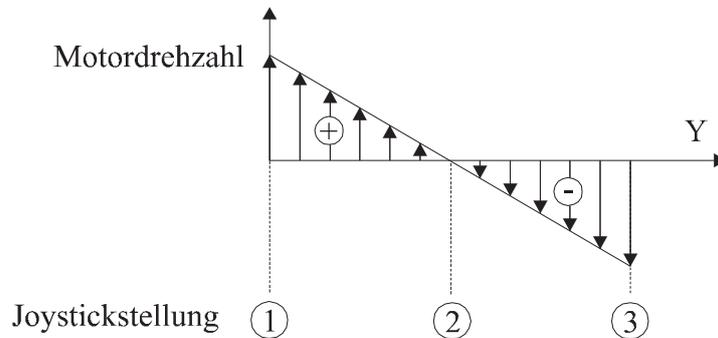


Abb. 11.4 Geschwindigkeitsverlauf von Kette 2

Der Drehrichtungswechsel von Antriebsmotor 2 erfolgt durch einen stetigen Übergang in Joystickstellung 2.

Das Wenden von Kettenfahrzeugen in Bergwerken ist durch die begrenzten Platzverhältnisse von großer Bedeutung. Daher soll dieses Abbildungsverfahren nicht weiter betrachtet werden.

11.1.4.4 Abbildungsverfahren 3

Eine weitere Möglichkeit bietet sich durch den Drehrichtungswechsel entlang der Diagonalen des Joystickkoordinatensystems. Der Verlauf der Motorgeschwindigkeit ist an Abb.11.5 zu erkennen.

Entlang der Joystickpositionen 1-3 erfolgt kein Drehrichtungswechsel für Antriebsmotor 2.

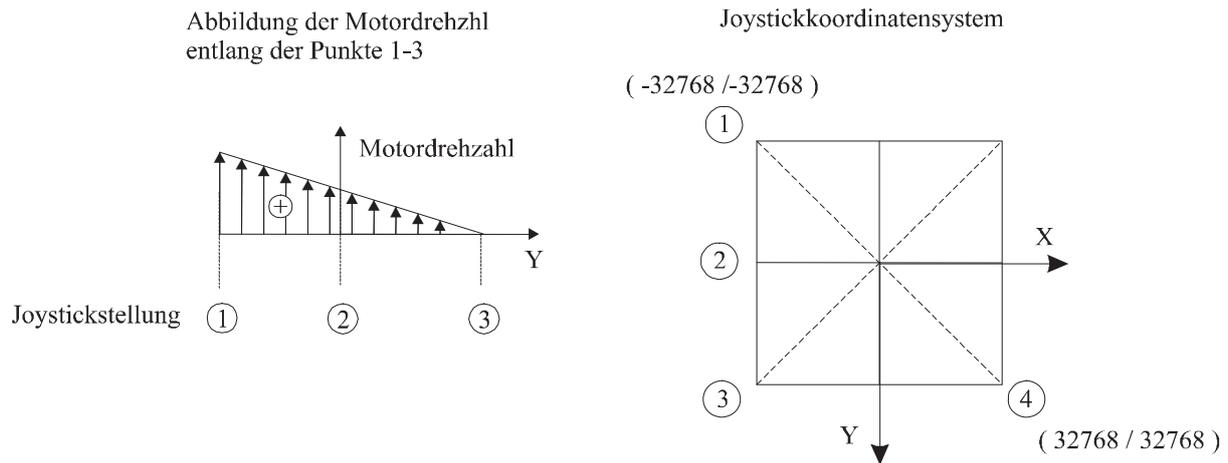


Abb. 11.5 Abbildungsverfahren 3

Der Drehrichtungswechsel für den Antriebsmotor 2 erfolgt entlang der Geraden von Punkt 1 bis Punkt 4. Um dies zu verdeutlichen erfolgt eine dreidimensionale Darstellung der Motorgeschwindigkeit bezüglich der Joystickkoordinaten in Abb.11.6

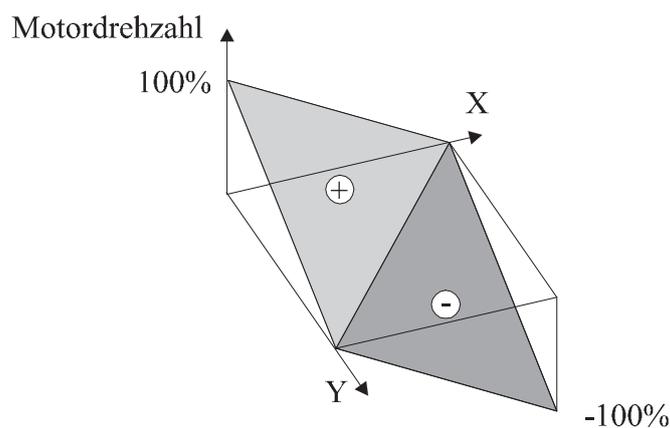


Abb. 11.6 Geschwindigkeitsverlauf von Antriebsmotor 2

Durch die erforderliche Symmetrie für die Geschwindigkeiten der Motoren, erhält man den Verlauf von Motorgeschwindigkeit 1 bezüglich des Joystickkoordinatensystems in Abb.11.7.

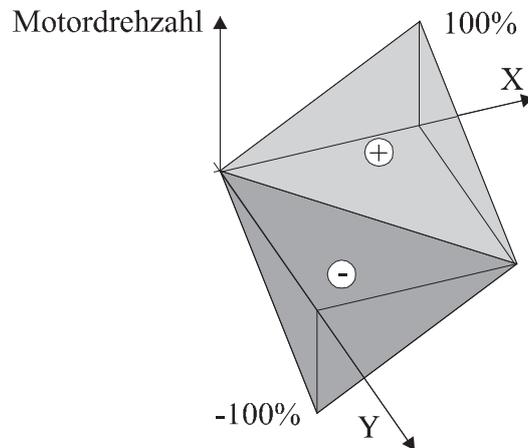


Abb. 11.7 Geschwindigkeitsverlauf von Antriebsmotor 1

Mathematische Darstellung

In Abb.11.5 kann man erkennen, daß die Nullposition in die mechanische Neutrallage des Joysticks verschoben wurde.

Verschiebung in die mechanische Neutrallage des Joysticks:

$$X = JoystickX - 32768 \quad (11.1)$$

$$Y = JoystickY - 32768 \quad (11.2)$$

Die Ebenen für die Motorgeschwindigkeiten errechnen sich aus :

$$Motor\ 1 = \frac{X + Y}{f} + d \quad (11.3)$$

$$Motor\ 2 = \frac{X - Y}{f} \quad (11.4)$$

$f = 2$ Skalierungsfaktor

für den Bezug zwischen maximaler Joystickkoordinate und maximaler
Motorgeschwindigkeit

$d = 32768$ Verschiebung entlang der z-Achse

Aus den Vorzeichen der Motorgeschwindigkeiten ergeben sich die Drehrichtungen der Motoren.

Bedingung	Richtungsbit 1	Richtungsbit 2
$Motor\ 1 > 0$	1	0
$Motor\ 1 \leq 0$	0	1
$Motor\ 2 > 0$	1	0
$Motor\ 2 \leq 0$	0	1

Tabelle 11.3 Richtungsbit der beiden Motoren

Aus (11.3) und (11.4) errechnet sich :

$$Motor1 = |Motor1| \quad (11.5)$$

$$Motor2 = |Motor2| \quad (11.6)$$

Die resultierenden Motorgeschwindigkeiten lassen sich in Abb.11.8 darstellen :

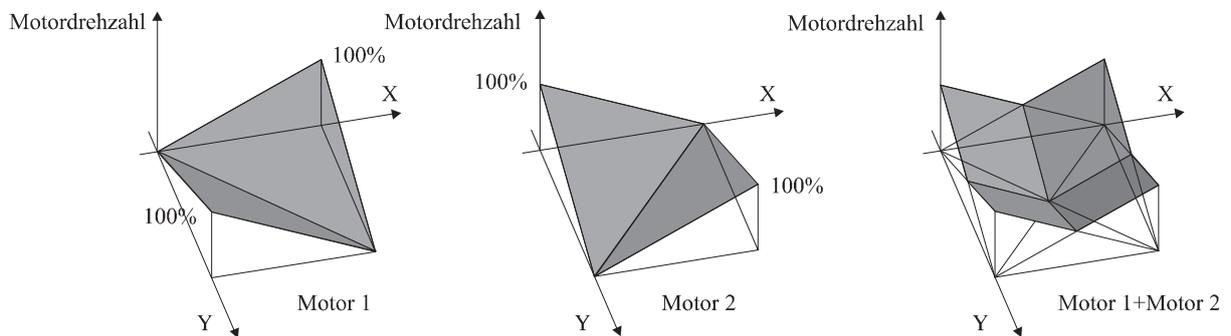


Abb. 11.8 Resultierende Motorgeschwindigkeiten

Resultierende Manövrierungseigenschaften

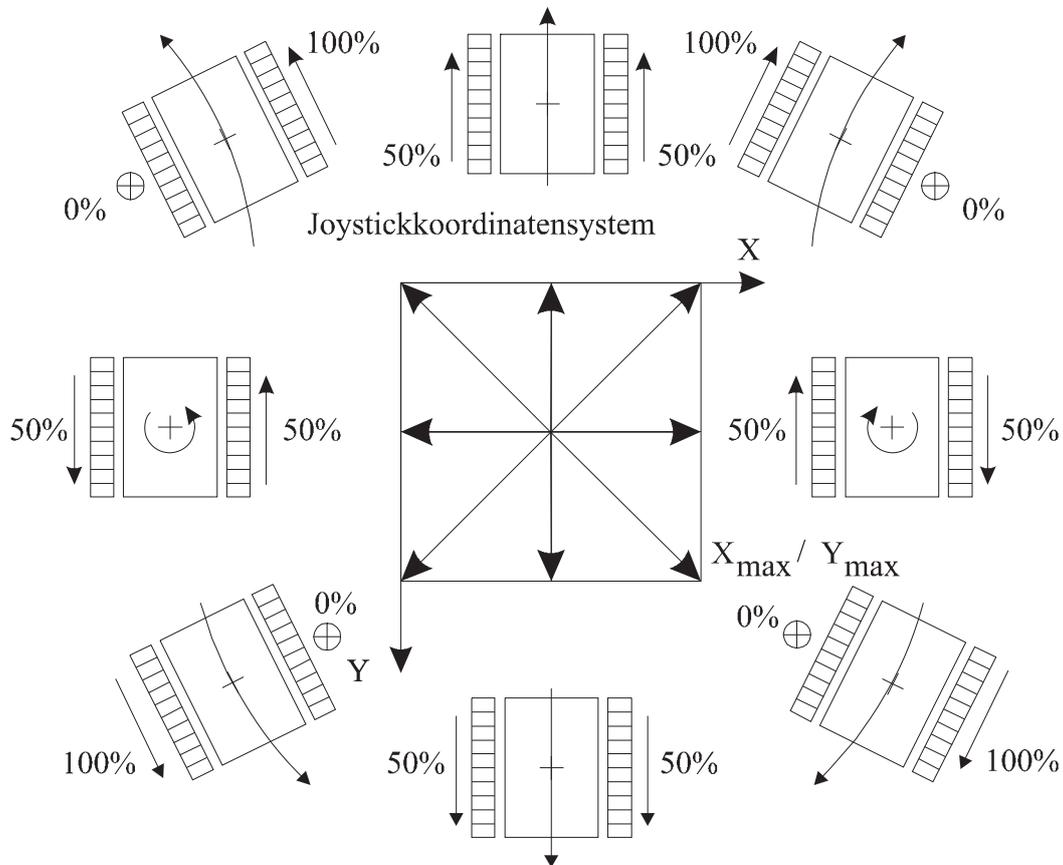


Abb. 11.9 Manövrierungseigenschaften des Kettenfahrzeuges

Wie man an Abb.11.9 erkennen kann, bewegt sich das Kettenfahrzeug in den hinteren Eckpositionen des Joysticks entgegengesetzt der intuitiv erwarteten Richtung.

Aufgrund der Tatsache, daß der Bediener des Teleoperatingsystems keinen Blickkontakt mit dem Kettenfahrzeug hat, wirkt sich dies nicht negativ auf die Steuerung des Fahrzeuges aus.

Die Steuerung des Kettenfahrzeuges mit diesem Abbildungsverfahren wurde als geeignet erachtet, und in das Teleoperatingsystem implementiert.

Die Softwarelösung dieses Abbildungsverfahrens findet sich Teil II der Diplomarbeit.

11.2 Abbildungsverfahren für die Krafrückkoppelung

11.2.1 Einleitung

Für die haptische Wahrnehmung bei der Bedienung des Kettenfahrzeuges ist es notwendig, die wirkenden Kräfte am Kettenfahrzeug zu messen und proportionale Kräfte am Eingabegerät (Joystick) wirken zu lassen.

Für die Krafrückführung in einem Teleoperatingsystem ist der Bewegungswiderstand des Kettenfahrzeuges von Bedeutung. Eine Möglichkeit diesen Widerstand zu ermitteln, kann durch die Drehmomentenmessung an den Antriebsmotoren realisiert werden.

Beim Versuchsaufbau kann aus der Strommessung an den Antriebsmotoren auf die Antriebsmomente – und somit auf tatsächlich wirkende Kräfte - rückgeschlossen werden.

Eine weitere Möglichkeit für die Krafrückführung besteht darin, virtuelle Kräfte zu verwenden. Diese Kräfte werden beispielsweise durch Abstände zu Hindernissen repräsentiert. Im Gegensatz zur Drehmomentenmessung sind bei diesem Verfahren Kollisionen mit Hindernissen ausgeschlossen und somit bevorzugt anzuwenden.

11.2.2 Messung von Abständen zu Hindernissen

Für den Versuchsaufbau wurden Ultraschallsensoren verwendet um Distanzen zu Hindernissen zu messen. Mit den Sensorwerten ist es möglich virtuelle Kräfte zu modellieren und als proportionalen Kraftvektor am Eingabegerät wirken zu lassen.

In Abb. 11.10 ist die Position der Ultraschallsensoren am Kettenfahrzeug dargestellt.

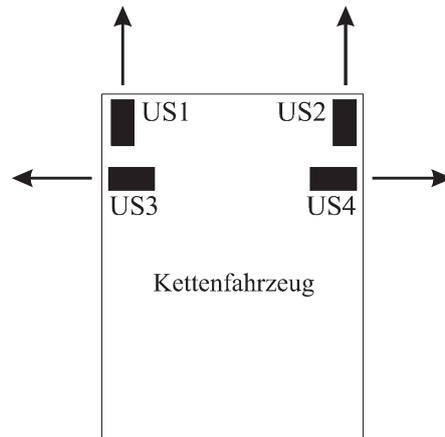


Abb. 11.10 Position der Ultraschallsensoren

Aus den einzelnen Abständen läßt sich eine resultierende linear – proportionale Kraft ermitteln, welche durch die Krafrichtung und Kraftamplitude definiert ist. Der daraus ermittelte Kraftvektor läßt sich am Eingabegerät (Joystick) erzeugen, um so eine Krafrückführung auf das Eingabegerät zu bewirken.

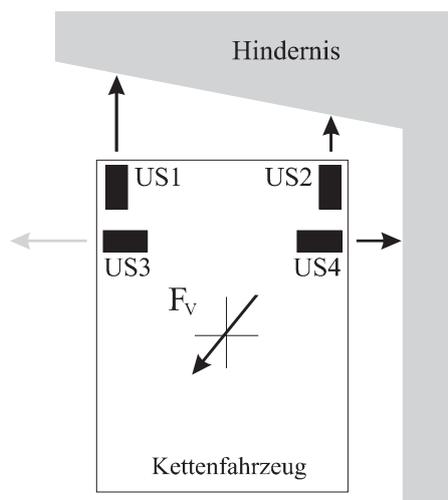


Abb. 11.11 Virtuelle Kraftwirkung bei Hindernissen

11.2.3 Mathematische Abbildung

Aus den Sensorwerten läßt sich eine horizontale und vertikale virtuelle Kraft einführen.

Da geringere Abstände zu Hindernissen kritischere Auswirkungen haben können, wurde für die Vertikalkomponente der kleinere Sensorwert verwendet.

Wertebereiche

Ultraschallsensor	0 - 32767
Kraftrückführungsvektor (Eingabegerät)	0 - 10000

Umrechnung der Sensorwerte auf virtuelle Kräfte

Für weitere Betrachtungen erfolgt die Umrechnung von Sensorwerten auf eine virtuelle Kraft.

$$F_V = 10000 - \frac{\text{Sensorwert}}{32767} \cdot 10000 \quad (11.7)$$

Horizontale Kraftkomponente

$$F_X = F_{V,3} - F_{V,4} \quad (11.8)$$

$F_{V,3}, F_{V,4}$ Virtuelle Kraft von US3 und US4

Vertikale Kraftkomponente

Aus den Ultraschallsensoren US1 und US2 wird die maximale Komponente ermittelt, da diese den kritischeren Abstand darstellt.

$$F_Y = \max(F_{V,1}, F_{V,2}) \quad (11.9)$$

$F_{V,1}, F_{V,2}$ Virtuelle Kraft von US1 und US2

Resultierende Krafrichtung

$$\alpha = \arctan\left(\frac{F_X}{F_Y}\right) \quad (11.10)$$

Aus (11.10) ergeben sich positive und negative Werte für den resultierenden Winkel. Der verwendete Force Feedback Joystick kann nur Krafrichtungen von 0 bis 359° verarbeiten. Über Beziehung 11.11 läßt sich auf geeignete Werte umrechnen.

$$\alpha' = (360 - \alpha) \bmod 360 \quad (11.11)$$

mod Modulo - Funktion von 360°

Resultierender Vektorbetrag

Wenn sich für die Vertikal- und Horizontalkomponente der virtuellen Kraft Maximalwerte ergeben, würde sich bei herkömmlicher Betragsberechnung eine Bereichsüberschreitung ergeben. Aus diesem Grund wird der resultierende Vektorbetrag aus dem größeren Wert von Horizontal- und Vertikalkraft ermittelt.

$$|F_G| = \max(|F_X|, |F_Y|) \quad (11.12)$$

F_X Virtuelle Horizontalkraftkomponente

F_Y Virtuelle Vertikalkraftkomponente

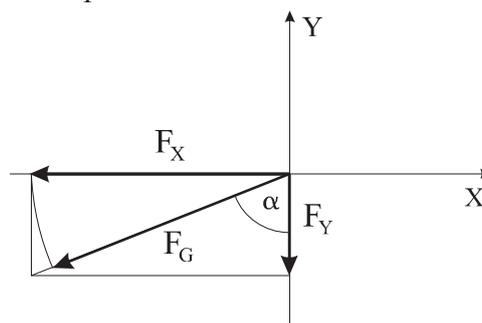


Abb. 11.12 Resultierender Krafrückführungsvektor

Beim erwähnten Versuchsaufbau handelt es sich um die erste Erprobung eines Krafrückführungssystems. Neben dem beschriebenen Abbildungsverfahren der Krafrückführung bestehen weitere Möglichkeiten haptische Rückführungen zu realisieren.

Die Softwarelösung dieses Abbildungsverfahrens ist Teil II der Diplomarbeit zu entnehmen.

12 Softwaredokumentation

12.1 Einleitung

Für das Teleoperatingsystem mußte ein Softwarepaket entwickelt werden, welches alle Hardwarekomponenten zu einem Gesamtsystem verbindet.

Es wird zwischen Programmteilen auf der SPS, und der Programmierung auf dem PC unterschieden. Die Softwarekomponenten auf der mobilen SPS werden in Echtzeit abgearbeitet, und werden dazu verwendet, die Hardwarekomponenten auf dem Kettenfahrzeug zu verwalten.

Implementierte Eigenschaften des Softwarepakets

- Bewegungsvorgabe des Kettenfahrzeuges durch einen Joystick am Leitstand
- Bedienung des Kameranahsystems
- Berechnungen der Kraftrückführung zum Joystick aufgrund der Sensorsignale am Kettenfahrzeug
- Verarbeitung und Darstellung von Ein- Ausgangsvariablen im Gesamtsystem
- Grafische Visualisierung der Motorgeschwindigkeiten

12.2 Komponenten des Softwarepaketes

Das Gesamtsoftwaresystem besteht aus mehreren Komponenten. In diesem Kapitel werden die Funktionen dieser Komponenten beschrieben, und die Zusammenbindung zu einem funktionsfähigen Softwarepaket erläutert.

Hauptprogramm

Alle Komponenten des Softwareprojektes werden durch das Hauptprogramm koordiniert. Das Hauptprogramm auf dem PC wurde in der Programmiersprache Visual Basic® von der Firma Microsoft verfaßt. Es handelt sich um ein selbständig lauffähiges Programm, welches in kompilierter Form vorliegt. Als Betriebssystem auf dem Leitstand-PC wurde Windows95® verwendet.

ActiveX – Komponente

Für die Anbindung des Force Feedback Joysticks mußte eine ActiveX - Komponente in Visual C++ ® verfaßt werden. Mit dieser Softwarekomponente kann der Joystick ausgelesen werden, und die Force Feedback – Eigenschaften des Joysticks genutzt werden.

Die Programmierung des Joysticks erfolgte durch den Einsatz von SDK-Funktionen der Firma Microsoft® (**Software Development Kit**)

DDE – Anbindung (Dynamik Data Exchange)

Die Verbindung von PC und SPS erfolgt durch das DDE-Serverprogramm NET2000 der Firma B&R. Die notwendigen Schreib- und Lesevorgänge für die Kommunikation vom Leitstand-PC mit der SPS werden über eine DDE-Schnittstelle realisiert.

SPS-Programmkomponenten

Die Softwarekomponenten auf der mobilen SPS wurden in der Programmiersprache PG2000 der Firma B&R verfaßt. Es wurden mehrere selbständig lauffähige Programmteile verfaßt, welche die ankommenden Daten vom stationären PC auf die Aktoren weiterleiten, und die Sensordaten für die Verarbeitung an den PC übermitteln.

12.3 Datenflüsse im Gesamtsystem

Für die Verbindung der einzelnen Softwarekomponenten zu einem Gesamtsystem ist es notwendig, die Datenflüsse im Gesamtsystem zu analysieren.

Alle Hardwarekomponenten des Teleoperatingsystems müssen durch Softwareteile auf der PC / SPS – Ebene zu einem Gesamtsystem verbunden werden.

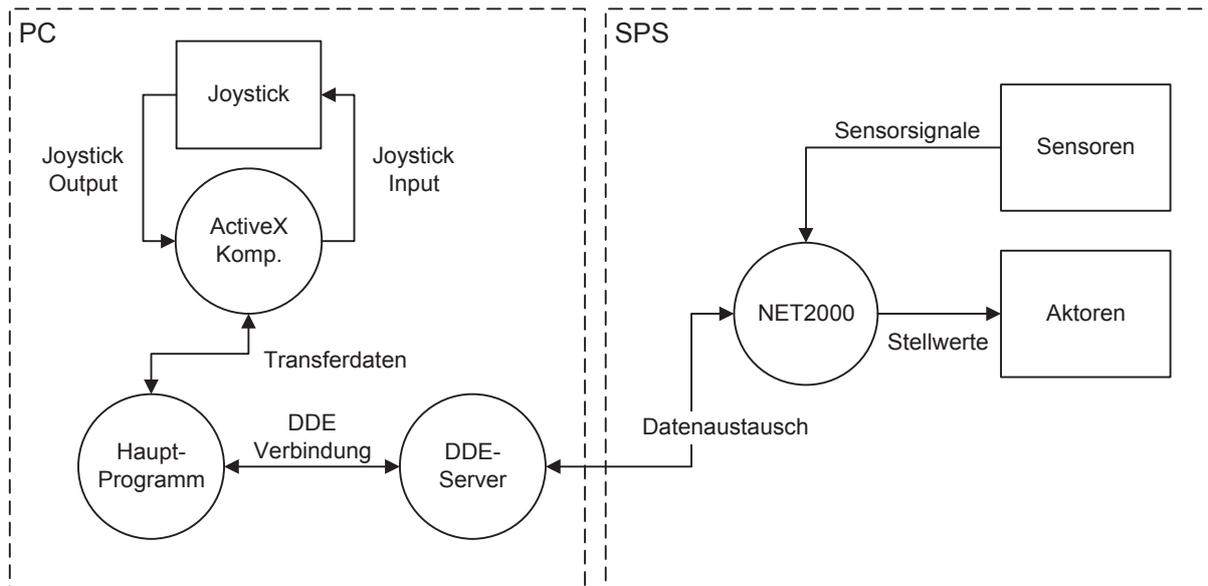


Abb. 12.1 Komponentenverknüpfung

Es wird zwischen PC- und SPS - Programmteilen unterschieden, welche die Prozesse lt. Abb.12.1 beinhalten.

Die Beschreibung der genauen Funktionen der Prozesse und Verbindungen erfolgen im weiteren Verlauf dieses Kapitels.

12.4 Hauptprogramm

Das Hauptprogramm dient der Verbindung aller Softwarekomponenten, und als Schnittstelle zwischen dem Bediener und der Hardware des Teleoperatingsystem.

Auf der grafischen Oberfläche des Hauptprogrammes ist es möglich Daten des Teleoperatingsystems zu visualisieren, und Grundeinstellungen vorzunehmen.

Das Hauptprogramm wurde in der Programmiersprache Visual Basic 5.0® von der Firma Microsoft® verfaßt. Basierend auf der Windows-Technologie von Microsoft wurde eine grafische Bedienungsoberfläche erzeugt.

In Abb.12.1 ist die Implementation des Hauptprogrammes im Gesamtsoftwaresystem dargestellt.

Steuerelemente haben – wie das verfaßte ActiveX Element – bestimmte Eigenschaften welche durch die Programmierung beeinflußt oder abgefragt werden können.

Jedes grafische Fenster, sowie die verwendeten Steuerelement (z.B. Textbox, Label, usw..) besitzen Properties, mit welchen es möglich ist deren Eigenschaften zu beeinflussen.

Die Funktionalität des Hauptprogrammes ergibt sich aus der Kombination der Objekte mit deren Softwarefunktionalität.

12.4.1 Bedienungsoberfläche des Hauptprogrammes

In Abb. 12.2 wird die grafische Oberfläche für die Bedienung des Teleoperatingsystems dargestellt. Die Position der einzelnen Fenster kann verändert werden, um so die Darstellung des Videobildes vom Bedienungsortes als grafisches Fenster auf der Oberfläche zu implementieren.

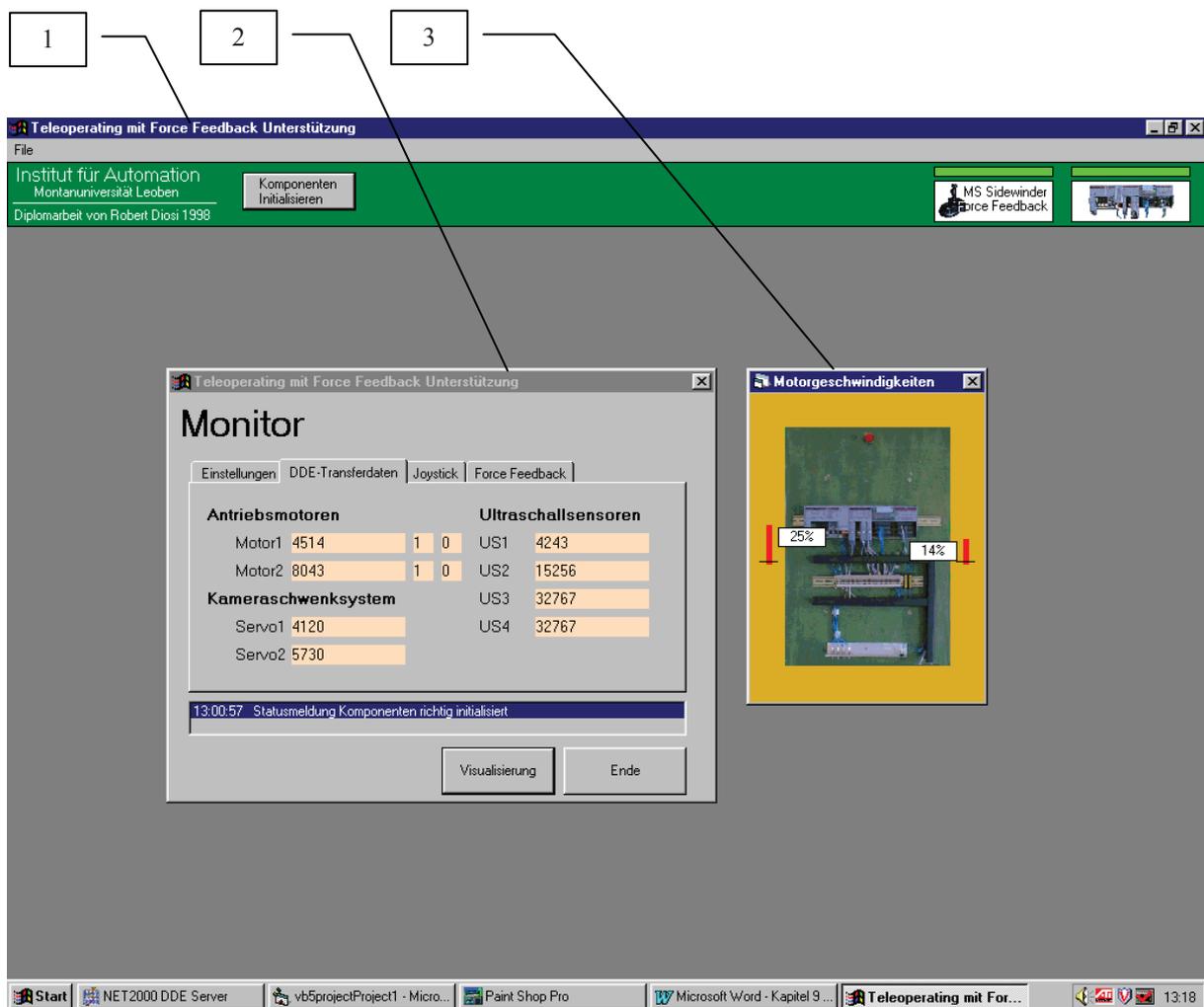


Abb. 12.2 Bedienungsoberfläche des Hauptprogrammes

Komponenten :

- | | | |
|-----|---|--------------|
| (1) | Hauptfenster | (MDIForm) |
| (2) | Grundeinstellungen / Visualisierung von Variablen | (MainForm) |
| (3) | Visualisierung der Motorgeschwindigkeiten | |

12.4.2 Kurzbeschreibung der Komponenten des Hauptprogrammes

Das Hauptprogramm wurde als „Multi - Dokument – Anwendung“ konzipiert (MDI – Anwendung). Die grafische Bedienungsfläche besteht dabei aus einem Hauptfenster und mehreren Dokumentenfenster, den sog. MDI-Child Fenstern. (Abb.12.2)

Die Darstellung und Beschreibung der einzelnen Fenster erfolgt in den folgenden Kapiteln.

1. MDIForm

Das MDI-Form Fenster besteht aus mehreren Dokumentenfenstern, welche die Funktionalität des Hauptprogrammes beinhalten.

2. MainForm (Dokumentenfenster)

In diesem Fenster werden alle Variablen dargestellt. Weiters ist es möglich Grundeinstellungen zu vorzunehmen.

Die Kommunikation mit dem DDE-Server wird über die eingebetteten Objekte realisiert.

3. Motorgeschwindigkeiten (Dokumentenfenster)

Darstellung des Kettenfahrzeuges mit den Kettengeschwindigkeiten in %-Angaben.

Durch die Verwendung eines Visual Basic-Moduls werden globale Funktionen implementiert, welche von jeder Windows-Form aus aufrufbar sind.

Die Eigenschaften der Objekte können über die sog. Properties angesprochen werden.

Der Programmtext des Hauptprogrammes wird größtenteils als Bestandteil der Fenster implementiert. Die Softwarebestandteile sind direkt mit den Objekten (Steuerelementen) verknüpft, und nicht von den Fenstern zu trennen.

Im weiteren Verlauf dieses Kapitels wird die Verwendung und Funktion der Objekte bzw. Steuerelemente der einzelnen Fenster beschrieben.

12.4.3 Funktionsbeschreibung der MDI-Form

Das Hauptprogramm wurde als MDI-Anwendung konzipiert. Mittels untergeordneter Dokumentenfenster wird die Bedienung und Visualisierung am Teleoperatingsystem realisiert. Um das Videobild der Kamera am Kettenfahrzeug als Fenster im Hauptprogramm einzufügen, wurde diese Art der Anwendungsprogrammierung gewählt.

Die weiteren Fenster des Hauptprogrammes sind Bestandteil der MDI-Form.

Im MDI-Fenster (Abb.12.3) wird das System initialisiert und die Statusanzeigen über die Verfügbarkeit der Hardwarekomponenten Joystick und SPS dargestellt. Ist eine der Komponenten nicht verfügbar, wird das Programm beendet.

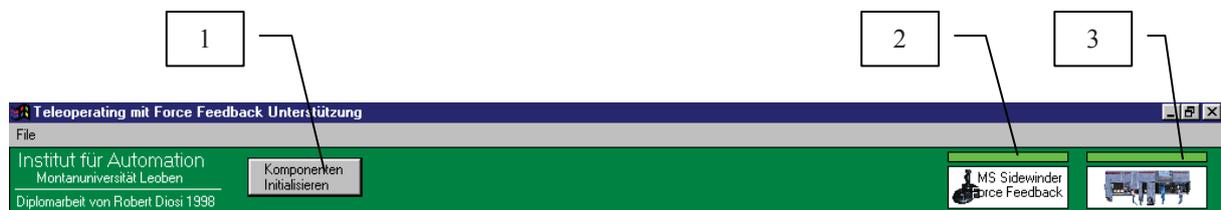


Abb. 12.3 Statuszeile der MDIForm

12.4.3.1 Objekte der MDI-Form

Die Objekte aller Fenster sind von allen Ebenen aufrufbar. Es werden so Kreuzverknüpfungen zwischen den einzelnen Objekten erzeugt.

Command1 {1}	Command Button
Initialisierung des Systems durch Aufruf der MainForm	
Procedure : Click	Programmteil : T1.1

ende {-}	menueintrag / name
Beenden des Hauptprogrammes und deinstallieren des Joysticks	
Procedure : Click	Programmteil : T1.2

Fortsetzung :

Light1 {2}	PictureBox
Anzeige der Verfügbarkeit des Joysticks (Rot / Grün - Leuchte)	
Property : BackColor	Werte : Rot = &HFF& Grün = &HFF00&

Light2 {3}	PictureBox
Anzeige der Verfügbarkeit der SPS (Rot / Grün - Leuchte)	
Property : BackColor	Siehe Light1

Die MDI-Form verfügt über weitere grafische Objekte, welche nur für visuelle Zwecke genutzt werden und keine weitere Softwarefunktionalitäten aufweisen. Auf diese zusätzlichen Elemente wird in diesem Zusammenhang nicht eingegangen, da die Darstellung dieser Objekte in Abb. 12.3 erfolgt.

Initialisierung durch T1.1

1. MainForm Laden
2. MainForm darstellen
3. Aufruf des Basic-Moduls *ForRuntime* (**T 1.1**) um die Joystickkoordinaten und Tasten auszulesen

12.4.4 Funktionsbeschreibung der MainForm

Im MainForm – Fenster wurde die eigentliche Funktionalität des Hauptprogrammes implementiert. Durch die große Anzahl von Funktionsobjekten und die klar zu trennenden Funktionsstrukturen, wurde dieses Programmfenster in mehrere Bereiche unterteilt.

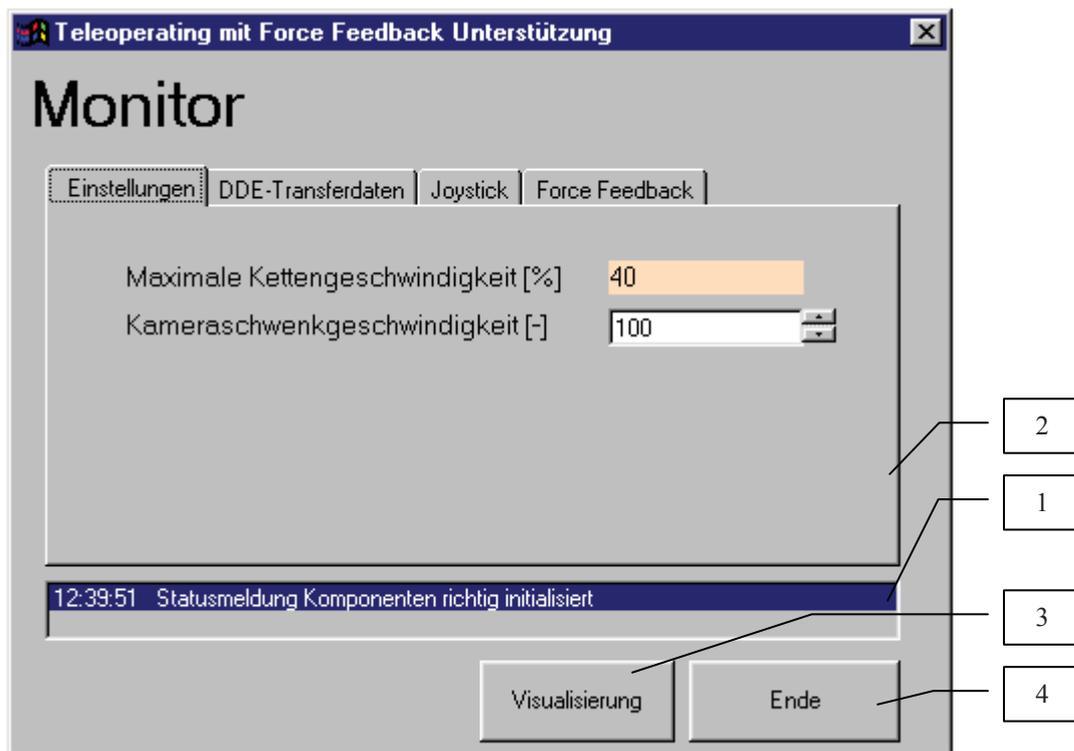


Abb. 12.4 Erscheinungsbild der Main Form

12.4.4.1 Objekte der MainForm

Die verwendeten Objekte verfügen meist über eine große Anzahl von Properties. Es werden nur jene angeführt welche Verwendung finden, da dies sonst den Umfang sprengen würde.

List1 {1}	ListBox
Statusanzeige in Textform	
Properties :	
AddItem	Hinzufügen einer Zeile
Listindex	Ansprechen einer Textzeile

Fortsetzung :

SSTab1 {2}	SSTab
Bereichsunterteilung in Einstellungen, DDE-Transferdaten, Joystick, Force Feedback	
Keine Property verwendet	

Projekt20_ocx1 {-}	Projekt20_ocx
ActiveX – Komponente für die Kommunikation mit dem Force Feedback Joystick	
Siehe Kapitel 12.6	

Cmd_simulation {3}	CommandButton
Öffnen des Fensters „Simulation“ für die Visualisierung der Motorgeschwindigkeiten	
Procedure : Click	Programmteil : T2.1

Cmd_ende {4}	CommandButton
Beenden des Hauptprogramms und Deinstallieren des Joysticks	
Procedure : Click	Programmteil : T2.2

Timer1 {-}	Timer
Zeitabhängige Berechnung des Force Feedback Vektors	
Property : Interval	Wert = 100 ms Programmteil : T2.3

12.4.4.2 Ladevorgang der MainForm

Der Ladevorgang der MainForm wird dann ausgeführt, wenn das Hauptprogramm gestartet wird. Die DDE-Kanäle für die Kommunikation mit der SPS werden geöffnet, und Startwerte für die Variablen gesetzt.

Für den Datentransfer zum DDE-Server wird ein Datenarray mit insgesamt 21 möglichen Kanälen verwendet. Mit einer DDE – Verbindung ist es nur möglich Text- und Bildinformationen zu übertragen.

Programmteil T2.0

Arrayelement	Verwendung	Wertebereich
pv[0]	Motorgeschwindigkeit 1	0-32767
pv[1]	Motorgeschwindigkeit 2	0-32767
pv[2]	Motor 1 / Bit 1	0,1
pv[3]	Motor 1 / Bit 2	0,1
pv[4]	Motor 2 / Bit 1	0,1
pv[5]	Motor 2 / Bit 2	0,1
pv[6]	Servomotor 1	1000 - 10000
pv[7]	Servomotor 2	1000 - 10000
pv[8-11]	Ultraschallsensor 1-4	0 - 32767

*12.4.4.3 Bereichsunterteilung des Tabulatorelement***1. Einstellungen**

In diesem Bereich ist es möglich Grundeinstellungen für die maximale Kettengeschwindigkeit des Fahrzeuges, und die Kameranahgeschwindigkeit vorzunehmen.

Die maximale Kettengeschwindigkeit kann dabei durch den Drehregler am Joystick erfolgen. (Schubkontrolle – siehe Kapitel 3.2.1.3)

2. DDE-Transferdaten

Die Schnittstelle für die Verbindung mit dem DDE-Server NET2000, und für die Visualisierung der Variablen für die Datenkommunikation mit der SPS.

3. Joystick (Lesevorgänge vom Joystick)

Hier werden die gelesenen Joystickwerte (Koordinaten, Drehregler, Hat-Button, usw.) dargestellt.

4. Force Feedback (Schreibvorgänge auf den Joystick)

Der aufgrund der Sensorwerte resultierende Kraftvektor auf den Force Feedback Joystick wird grafisch mit den zugehörigen Werten der Kraftamplitude und Kraftrichtung visualisiert.

12.4.4.4 Bereich Einstellungen

Mit dem Drehregler am Joystick (Schubkontrolle) ist es möglich die maximale Geschwindigkeit des Kettenfahrzeuges zu beeinflussen. (Abb.12.5)

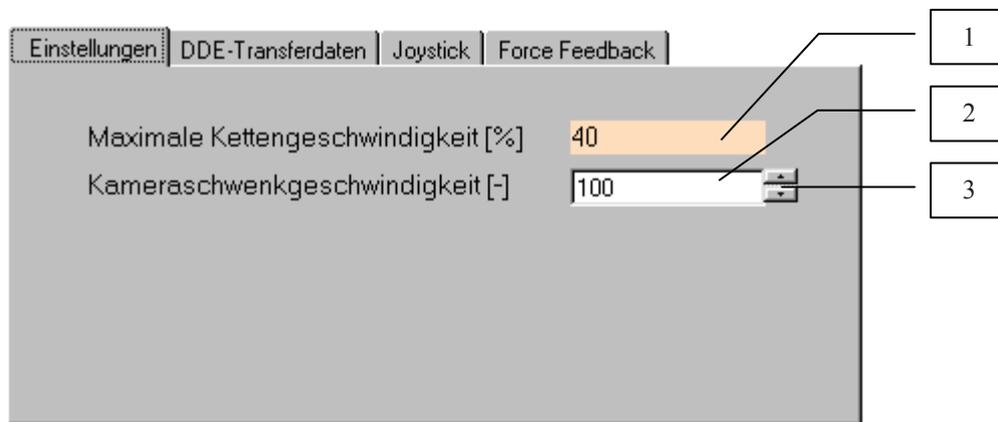


Abb. 12.5 Bereich Einstellungen

Objekte vom Bereich Einstellungen

Label_max_speed {1}	Label
Darstellung der prozentualen Maximalgeschwindigkeit des Kettenfahrzeuges [0-100%] Dieser Wert kann während der Laufzeit durch den Schubregler am Joystick verändert werden.	
Keine Property verwendet	

Text_schwenk_geschwindigkeit {2}	TextBox
Vorgabe der Schrittweite für die Schwenkbewegungen der Servomotoren des Kameraschwenksystems.	
Procedure : Keypress	Wertebereich = 0 -1000 Programmteil : T 2.4

Vscroll1 {3}	Vscroll
Beeinflussung der Schwenkgeschwindigkeit der Servomotoren mittels Mauszeiger	
Procedure : Change	Programmteil : T2.5

12.4.4.5 Bereich DDE-Transferdaten

In diesem Bereich werden die aktuellen Werte aller verwendeten DDE - Kanäle dargestellt.(Abb.12.6)

Für die Visualisierung der Variablen werden ausschließlich Label-Objekte verwendet, welche keine Softwarefunktionalität enthalten.

Beim Ladevorgang der MainForm werden alle Variablen initialisiert (Kapitel 12.4.4.2)

Ändert sich der Inhalt einer Objektvariable, so wird der DDE-Kanal aktualisiert, und bei Programmfehlern eine Procedure für Fehlerbehandlung aufgerufen. In diesem Bereich handelt es sich nur um die Visualisierung von Variablen.

Unterteilung der DDE-Transferdaten :

- Antriebsmotoren
- Ultraschallsensoren
- Kameranahsystem

Einstellungen		DDE-Transferdaten		Joystick		Force Feedback	
Antriebsmotoren				Ultraschallsensoren			
Motor1	2097	0	1	US1	32767		
Motor2	3877	1	0	US2	32767		
Kameranahsystem				US3	32767		
Servo1	4120			US4	32767		
Servo2	5730						

Abb. 12.6 Bereich DDE-Transferdaten

12.4.4.6 Bereich Joystick

Die gelesenen Joystickwerte der ActiveX – Komponente werden visualisiert und die Programmkomponente für die Berechnung der Motorgeschwindigkeiten ausgeführt.

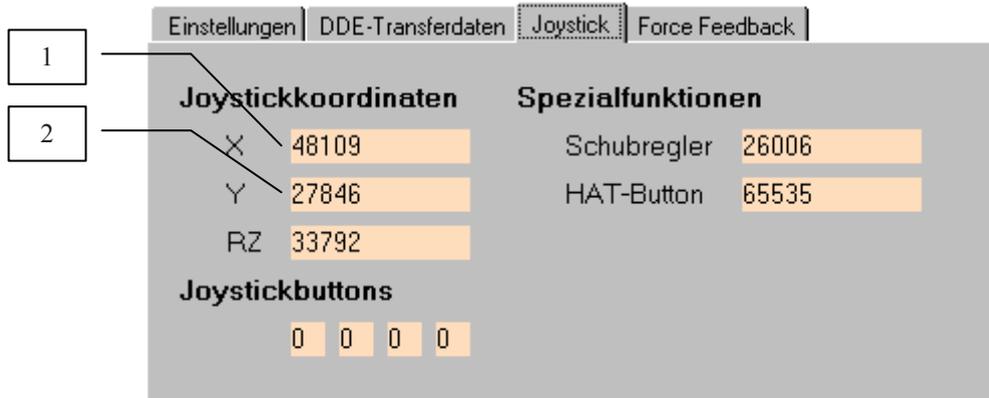


Abb. 12.7 Bereich Joystick

Objekte vom Bereich Joystick

Label_joy_x {1}	Label
Ausgabe der Joystickkoordinate in X – Richtung und Aufruf der Steuerung	
Procedure : Change	Programmteil : T2.7

Label_joy_y {2}	Label
Ausgabe der Joystickkoordinate in Y-Richtung und Aufruf der Steuerung	
Procedure : Change	Programmteil : T2.8

Es befinden sich noch weitere Objekte in dieser Bereichsunterteilung (Abb. 12.7), welche nur der Visualisierung von Variablen dienen.

12.4.4.7 Bereich Force Feedback

Die Rückkoppelung des Kraftvektors auf den Joystick wird grafisch und wertemäßig dargestellt. Die Berechnung der resultierenden Krafrückkoppelung auf das Eingabegerät erfolgt durch das Timer-Element auf der MainForm.

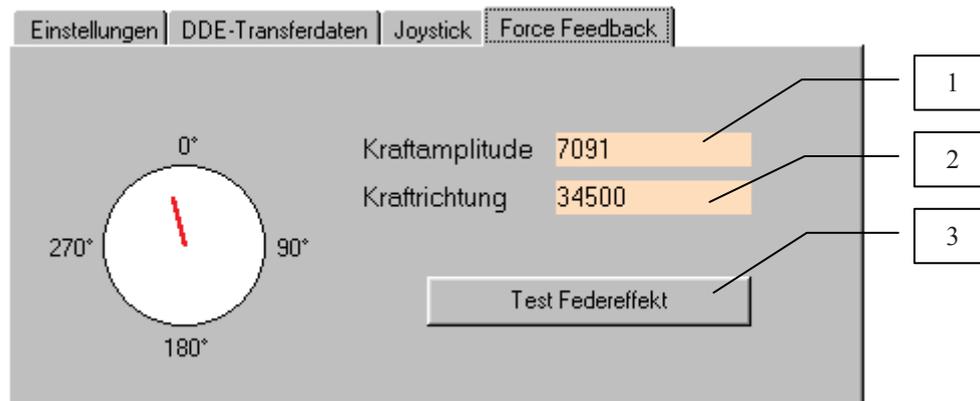


Abb. 12.8 Bereich Force Feedback

Objekte vom Bereich Force Feedback

Label_kraft_amplitude {1}	Label
Ausgabe der Kraftamplitude, und Übergabe der Amplitude an das ActiveX-Element	
Procedure : Change	Wertebereich = 0 - 10000 Programmteil : T2.9

Label_kraft_richtung {2}	Label
Ausgabe der Krafrichtung des Force Feedback Vektors und Übergabe an ActiveX-Element	
Procedure : Change	Wertebereich = 0 – 35599 Programmteil : T2.10

Cmd_test_force {3}	CommandButton
Für Testzwecke kann eine Federkraft mit maximaler Federsteifigkeit am Joystick erzeugt werden	
Procedure : Click	Programmteil : T2.11

12.4.5 Schnittstellen des ActiveX - Elementes

Für die Implementierung des Force Feedback Joysticks im Hauptprogramm werden durch das ActiveX – Element Schnittstellen definiert. Es können so die erforderlichen Werte des Joysticks ausgelesen, und die Eigenschaften der Krafrückkoppelung beeinflusst werden.

Die Beschreibung der Funktionsweise und der Programmierung der ActiveX – Komponente erfolgt in Kapitel 12.6. Im weiteren Verlauf dieses Kapitels sollen die möglichen Aufrufe im Visual Basic Programm erläutert werden.

12.4.5.1 Procedure - Aufrufe

Mit den folgenden Aufrufen können Variablen an die ActiveX – Komponente übergeben werden, um die Force Feedback Eigenschaften des Joysticks zu beeinflussen.

Initialisierung des Joysticks	InitFF()
Akquirierung und Initialisierung des Force Feedback Joysticks	
Federkraft erzeugen	createEffect (lFederkonstante as Long) <i>lFederkonstante = 0-10000</i>
Erzeugen einer bezüglich der Auslenkung des Joysticks variierenden Federkraft	
Konstante Kraft erzeugen	KonstanteKraftErzeugen(lAmplitude As Long, lRichtung As Long) <i>lAmplitude = 0-10000</i> <i>lRichtung = 0-35599</i>
Erzeugen einer konstanten Kraft (auch zur Initialisierung des Force Feedback Vektors)	
Kraftamplitude ändern	KraftAmplitude(lAmplitude As Long) <i>lAmplitude = 0-10000</i>
Die Kraftamplitude der konstanten Kraft kann durch diesen Aufruf geändert werden. Voraussetzung hierfür ist die Initialisierung der konstanten Kraft.	

Fortsetzung :

Kraftrichtung ändern	KraftRichtung(IRichtung As Long) <i>IRichtung = 0 - 35599</i>
Die Kraftrichtung der konstanten Kraft kann durch diesen Aufruf geändert werden. Voraussetzung hierfür ist die Initialisierung der konstanten Kraft.	

Joystickwerte lesen	GetJoy()
Bei diesem Aufruf werden alle Event-Variablen des Joysticks an das Hauptprogramm übermittelt.	

Joystick deaktivieren	shutdown()
Die Force Feedback Eigenschaften werden durch diesen Aufruf beendet	

12.4.5.2 ActiveX - Eventfunktionen

Diese Funktionen übermitteln Variablen von der ActiveX - Komponente an das Hauptprogramm.

Joystickbuttons (1- 4)	SWFFButtons(bB1 As Boolean, bB2 As Boolean, bB3 As Boolean, bB4 As Boolean) <i>bB1-bB4 = 0,1</i>
Die Werte der am Joystickgriff befindlichen Buttons	

HatButton	SWFFHATButton(IVal As Long) <i>IVal = siehe Kapitel 3.2.1.3</i>
Liefert den Wert des HatButtons. (Einsatz für die Bewegungen des Kameranahmensystems)	

Fortsetzung :

Joystickkoordinaten	SWFFKoordinaten(IX As Long, IY As Long, IRz As Long) IX, IY ,IRz = 0 - 65535
Liefert die Werte für die Stellung des Joystickgriffes und die Längsrotation (IRz)	

Schubregler	SWFFSchub(IVal As Long) IVal = 0 - 65535
Liefert den Wert des Schubreglers (Siehe Kapitel 3) (Beeinflussung der Maximalgeschwindigkeit des Kettenfahrzeuges)	

Das ActiveX-Element enthält noch weitere Events/Properties, welche für das Projekt nicht verwendet werden.

12.5 DDE-Verbindung

12.5.1 Einleitung

Mit der DDE-Technologie ist es möglich dynamischen Datenaustausch zwischen zwei Programmen zur Laufzeit zu realisieren. Über sog. DDE-Kanäle können Text- und Bilddaten zwischen Windows-Programmen in beide Richtungen ausgetauscht werden. [6],[7]

In diesem Projekt wird mit Hilfe von DDE-Verbindungen zwischen dem Hauptprogramm und dem DDE-Server von B&R (Abb.12.1) die Kommunikation mit der SPS bewerkstelligt.

Alle Variablen welche zur Kommunikation mit der SPS benötigt werden, müssen über diese Schnittstelle geleitet werden.

DDE-Verbindungen können nur durch Objekte realisiert werden, welche Text- oder Bilddaten enthalten.

Man unterscheidet zwischen manuellem und automatischem Datenaustausch. Im Automatikmodus werden die Daten dann zum Empfänger übertragen, wenn sich Änderungen im Inhalt der Objekte ergeben. Im manuellen Modus kann der Datenaustausch gezielt eingeleitet werden.

12.5.2 Aufbau der DDE-Verbindung

In diesem Kapitel soll die grundsätzliche Vorgangsweise für den Aufbau der DDE – Verbindung zwischen dem Hauptprogramm und dem DDE – Server dargestellt werden.

Mit Hilfe der PV - Variablen werden die Kommunikationskanäle vom PC zur SPS aufgebaut.

Grundsätzlicher Ablauf bei der Herstellung einer DDE-Verbindung

1. Objekt für DDE-Verbindung initialisieren

Objekt.LinkTopic = "Net2DDE|cpu"

Net2DDE = Programmname des zu verbindenden DDE- Server- Programmes.

cpu = Direktzugriff auf Dokument des DDE-Servers.

2. Variable deklarieren

Objekt.Linkitem = "pv[0]"

Beispiel.: Objekt mit Variable pv[0] des DDE – Servers verbinden.

3. Automatikmodus einstellen

Objekt.Linkmode = 1

Die Variable wird bei Wertänderungen automatisch über den DDE - Kanal übertragen und aktualisiert.

12.6 ActiveX-Komponente

12.6.1 Einleitung

ActiveX Objekte sind kompilierte Programmodule, welche eingefügt in Windows-Applikationen als Steuerelemente arbeiten. Die ActiveX – Technologie beruht darauf, daß häufig verwendete Funktionen oder grafische Werkzeuge nur einmal verfaßt werden müssen. Durch eine klar definierte Schnittstelle zwischen der Hauptapplikation und dem ActiveX – Modul ist die Programmiersprache in welchem diese Komponente verfaßt wurde nicht von Bedeutung. Deklarationen von Schnittstellenvariablen in der Hauptapplikation sind für den Datenaustausch nicht notwendig.

Die OLE–Technologie (**O**bjekt **L**inking and **E**mbinding) ist die wichtigste Schlüsseltechnologie der Microsoft – Betriebssysteme und Grundlage der OCX – Steuerelemente, die als In-Process-Server fungieren.

Bei diesem Projekt wurde ein ActiveX – Element in der Programmiersprache Visual C++ erstellt. Wie bereits erwähnt, können ActiveX-Module in verschiedenen Applikationen verwendet werden. Es besteht so die Möglichkeit die Ein- und Ausgänge des verwendeten Force Feedback Joysticks beispielsweise in Excel-Anwendungen, LABview Programmen, usw.. zu verwenden.

12.6.2 Programmierung der ActiveX-Komponente

Das Grundgerüst des ActiveX – Elementes wurde durch den Einsatz der **Microsoft Foundation Classes (MFC)** erzeugt. Die Funktionsweise und der Aufbau dieser Klassen ist einschlägiger Fachliteratur zu entnehmen. [8]

In diesem Kapitel wird die Programmierung der ActiveX – Komponente für die Kommunikation mit dem Force Feedback Joystick beschrieben, ohne auf den genauen Aufbau der MFC – Klassen einzugehen. [9]

Der verwendete Force Feedback Joystick der Firma Microsoft® kann nur durch Aufrufe der DirectX® - Funktionen (Bestandteil von DirectInput®) programmiert werden.

Die Programmierung der ActiveX – Komponente erfolgt in der Programmiersprache Visual C++ 5.0® der Firma Microsoft.

Für dieses Projekt wurden sogenannte Wrapper-Funktionen des DirectX – Systems verwendet. Diese Funktionen enthalten die notwendigen DirectX – Aufrufe, und können durch Variablenübergabe dazu verwendet werden, den Joystick zu programmieren.[9]

Bestandteile des Visual C++ Projektes

- **Projekt20_ocx files**

- projekt20_ocx.cpp

- projekt20_ocx.def

- projekt20_ocx.odl

- projekt20_ocx.rc

- projekt20_ocxCtl.cpp

- projekt20_ocxPgp.cpp

- StdAfx.cpp

- swff_lib.cpp

In Projekt20_ocx.cpp finden sich die wesentlichsten Projektbestandteile, welche für die Wrapper – Funktionsaufrufe zuständig sind.

Das gesamte Visual C++ Softwareprojekt befindet sich auf der beigelegten CD-ROM in Teil 1 der Diplomarbeit

Fortsetzung :

- **Header files**

- ForceFeedbackDemo.h
- projekt20_ocx.h
- projekt20_ocxCtl.h
- projekt20_ocxPgp.h
- Resource.h
- Stdafx.h

- **Resource Files**

- dinput.lib
- projekt20_ocx.ico
- projekt20_ocxCtl.bmp

- **External Dependencies**

Beim Compilieren des Softwareprojektes werden neben der angeführten Programmbausteine zusätzliche externe Files an das Projekt angefügt, und im Bereich „External Dependencies“ verzeichnet.

12.6.2.1 Schnittstellenfunktionen

Die Schnittstellenfunktionen und deren Aufrufe in Visual Basic wurden im Kapitel 12.4.5 beschrieben. Die Realisierung und Programmierung der Methoden und ActiveX - Events erfolgt durch Aufrufe von DirectX® Wrapper – Prozeduren (Alle Wrapper – Funktionen befinden sich in swff_lib.h)

Diese Prozeduren sind Bestandteil der Microsoft® DirectInput SDK® - Paketes, und werden in diesem Zusammenhang nicht beschrieben, da dies den Umfang dieses Kapitels sprengen würde.[9]

Implementierte Funktionsaufrufe in Visual C++

void CProjekt20_ocxCtrl::OnDraw(„Verweis auf Quelltext“)

Diese Funktion wird dann ausgeführt wenn die ActiveX-Komponente in einem Anwenderprogramm eingefügt wird.

void CProjekt20_ocxCtrl::InitFF()

Joystick akquirieren und Statusvariable setzen.
Initialisieren des DirectInput® Device Objektes.

void CProjekt20_ocxCtrl::GetJoy()

Schnittstellenfunktion für die Übermittlung der Joystickvariablen an das aufrufende Programm. Die Variablen werden über ActiveX - Events an das Hauptprogramm übermittelt. Verwendung der DirectInput® - Struktur LPDIJOYSTATE welche die Statusvariablen des Joysticks enthält.

void CProjekt20_ocxCtrl::createEffect(long IFederkonstante)

Erzeugen eines Federeffektes mit folgenden möglichen Angaben :

- Dauer des Effektes
- Federkonstante in X-Richtung
- Mittelpunktabstand in X-Richtung
- Federkonstante in Y-Richtung
- Abhängigkeit eines Joystickknopfes
- Mittelpunktabstand in Y-Richtung

void CProjekt20_ocxCtrl::KonstanteKraftErzeugen(long IAmplitude, long IRichtung)

Befindet sich ein Krafteffekt im RAM des Joysticks, so wird dieser gelöscht und mit den neuen Werten gesetzt.

Erzeugen einer konstanten Kraft mit folgenden möglichen Angaben :

- Dauer des Krafteffektes
- Richtung der Kraft
- Amplitude der Kraft
- Zeitdauer für den Anstieg der Kraftamplitude
- Zeitdauer für das Abklingen der Kraftamplitude

void CProjekt20_ocxCtrl::KraftRichtung(long IRichtung)

Angabe der Krafrichtung eines vorhandenen Effektes

void CProjekt20_ocxCtrl::KraftAmplitude(long IAmplitude)

Angabe der Kraftamplitude eines vorhandenen Effektes

void CProjekt20_ocxCtrl::Shutdown()

Es werden alle Force Feedback Effekte gelöscht, und die Verbindung zum Joystick getrennt.

Alle Wertebereiche sowie Variablendeklarationen sind dem Microsoft® Force Feedback SDK® zu entnehmen.[9]

12.7 Echtzeitimplementationen

Für die Hardwareanbindung des Teleoperatingsystems wurden Echtzeit – Programmelemente auf der SPS implementiert. Es wurde hierfür eine SPS der Firma B&R Modell 2003 verwendet. Alle Programmbausteine wurden in der Programmierumgebung PG2000 der SPS verfaßt.

Die nachfolgenden Betrachtungen und Softwareelemente beziehen sich auf die oben genannte Programmierumgebung dieses Fabrikates. Die Deklarationen der verwendeten Input- und Output - Schnittstellen sind ausschließlich für diese SPS von B&R verfaßt worden.

Implementierte Eigenschaften

- Setzen der Geschwindigkeiten der beiden Antriebsmotoren
- Auslesen der Ultraschallsensoren und Weiterleitung an NET2000 – DDE Server
- Bereitstellung des Pulsweitenmodulierten Signals für das Kameranahsystem

12.7.1 Implementierte Task – Elemente

Durch den einfachen Aufbau der programmierten Task-Elemente werden nur die Kurzbeschreibungen der Softwarekomponenten angeführt. Die Implementierungen werden in Teil 2 der Diplomarbeit dargestellt.

Task BaumerUS

Auslesen der Ultraschallsensoren 1 - 4 und Übergabe auf den zu übertragenden Datenblock. (PV-Variablen)

Task N2_demo

Dieser Programmbaustein kontrolliert die Bewegungen des Kettenfahrzeuges.

Durch Übernahme der PV-Variablen werden die Motorgeschwindigkeiten und Motordrehrichtungen gesetzt.

Task Do135

Kontrolliert die Schwenkbewegungen des Kameranahsystems.

In diesem Task wird das Hardwaremodul DO135 der SPS für die Erzeugung der pulswertenmodulierten Signale verwaltet.

13 Resümee

13.1 Fazit

In dieser Diplomarbeit wurde ein komplettes Teleoperatingsystem mit Krafrückkoppelung entwickelt und erprobt. Dabei wurden Hardwarekomponenten und Softwarekomponenten derart verknüpft, daß ein funktionsfähiges Gesamtsystem erstellt werden konnte.

Für die Erweiterung des Blickfeldes des Bedieners wurde ein Kameranahsystem entwickelt, welches es ermöglicht die Orientierung der Videokamera am Kettenfahrzeug durch den Joystick zu verändern.

Obwohl es nicht möglich war eine geeignete Funkdatenübertragung in das System zu implementieren, konnten doch einige Versuche im Bereich der Krafrückkoppelung, und die Bewegungsvorgabe eines Kettenfahrzeuges durch einen Force Feedback Joystick durchgeführt werden.

Die Abbildung der Motorgeschwindigkeiten zur Joystickposition wurde für dieses Projekt eigens entwickelt, und ist Bestandteil einer Veröffentlichung.[1.1]

Die erstellte ActiveX – Komponente für die Nutzung der Force Feedback Eigenschaften des Joysticks kann in verschiedene Anwendungsprogramme auf Windows95® – Ebene implementiert werden. Auf diesem Wege ist es möglich die Eigenschaften des Joysticks universell in verschiedenen Programmen nutzbar zu machen. (Beispielsweise LabView®)

13.2 Erweiterungs- und Verbesserungsmöglichkeiten

Software

Die erstellte Software stellt ein Grundgerüst für die Zusammenführung und Koordination von Hardwarekomponenten dar. Es konnten alle notwendigen Programmteile in eine Softwarekomponente auf der PC- und SPS – Ebene implementiert werden.

Die Kommunikation zwischen dem stationären PC und der mobilen SPS konnte nur über den Weg des DDE - Servers von B&R (Net2000) realisiert werden. Die DDE- Verbindung stellt dabei keine geeignete Möglichkeit für schnelle Datenübertragungen dar. Es sollte daher versucht werden, schnellere Datenübertragungen durch die direkte Kommunikation des Anwenderprogrammes mit der SPS zu implementieren.

Datenübertragung

Beim vorliegenden Fall eines Teleoperatingsystems sollte die Bedienung des Kettenfahrzeuges in Echtzeit erfolgen. Es konnte aber gezeigt werden, daß die Datenübertragung eine wesentliche Engstelle darstellt, und zu keinem befriedigendem Resultat - bezüglich der Bewegungsvorgabe des Kettenfahrzeuges – führte.

Als Verbesserungsmöglichkeit sollte an dieser Stelle die Möglichkeit erwähnt werden, einen zusätzlichen Industrie-PC am Kettenfahrzeug für die Datenübertragung zu verwenden.

Dabei ist der Engpaß durch eine RS232 – Verbindung ausgeschaltet, und es kann eine Datenübertragung durch eine schnelle Ethernet – Verbindung vom Leitstand-PC zu Industrie-PC erfolgen. Der Industrie-PC kann durch die Anbindung mittels CAN, Profibus oder ähnlicher Bussysteme mit der SPS Daten austauschen.

Versuche auf diesem Bereich der Datenübertragung werden am Institut für Automation erprobt, und könnten gegebenenfalls in das Teleoperatingsystem implementiert werden.

14 Literaturliste

- [1] Dr.-Ing Wolfgang Beitz, Dipl. Ing. Karl-Heinz Küttner, „Taschenbuch für den Maschinenbau“, 18.Aufl., 1994, Springer Verlag Berlin
ISBN 3-540-57650-9

- [2] SKF, „SKF Hauptkatalog“, 1997, SKF Deutschland

- [3] Dimitry M.Gorivensky, Alexander M.Formalsky, Anatoly yu. Schneider
„Force Control of Robotics Systems“, 1997, CRC Press LLC New York
ISBN 0-8493-2671-0

- [4] Leonid I.Slutski, „Remote Manipulation Systems“, 1998, Kluwer Academic
Publishers Netherlands, ISBN 0-7923-4822-2

- [5] Rainer Miserre, „Mobiler Datenfunk“, 1995, Verlag Heinz Heisse GmbH & Co KG
Hannover, ISBN 3-88229-055-2

- [6] Loren D. Eidahl, „Using Visual Basic“, 1997, Que Corporation Indianapolis,
ISBN 0-7897-1412-4

- [7] Andreas Maslo, „Visual Basic 5.0“, 1997, Data Becker GmbH & Co. KG Düsseldorf,
ISBN 3-8158-1322-0

- [8] Dr. Christian Fleischhauer, „Visual C++ 5.0“, 1997, Data Becker GmbH & Co. KG
Düsseldorf, ISBN 3-8158-1261-5

- [9] Microsoft, „Sidewinder Force Feedback SDK“, 1997, Microsoft Corporation USA

14.1 Veröffentlichungen

- [1.1] *Teleoperating System with Force Feedback for Tracked Vehicles*, Ronald Ofner, Paul O'Leary, Robert Diosi, World Mining Congres IAMC'98 13th International Conference on Automation in Mining, High Tatras, Slovakia, 8-11 September 1998

Montanuniversität Leoben

Studienrichtung Montanmaschinenwesen



DIPLOMARBEIT

am

Institut für Automation

zum Thema

„Entwicklung eines Teleoperating – Systems
mit Krafrückkoppelung für ein Kettenfahrzeug“

Teil II: Anhang

von

cand.ing. Robert Diosi

Inhaltsverzeichnis

1	Veröffentlichung	1
2	Softwarequelltext	5
2.1	Visual Basic Quellcode.....	5
2.1.1	MDIForm.....	5
2.1.2	Programmteil MainForm	6
2.1.3	Globale Funktionen und Prozeduren	13
2.2	Visual C++ Quelltext.....	18
2.3	Echtzeitimplementationen	27
2.3.1	Task BaumerUS.....	27
2.3.2	Task N2_Demo.....	29
2.3.3	Task DO135.....	32
3	Elektrische Schaltpläne.....	35
3.1	Verteilerschaltung.....	35
3.2	Interfaceplatine	37
3.3	Spannungsversorgung.....	38
3.4	Beschaltung der Abdeckung	38
4	Konstruktionspläne	39
4.1	Verzeichnis der Konstruktionspläne.....	39

Teleoperating System with Force Feedback for Tracked Vehicles

Ronald Ofner¹, Paul O`Leary², Robert Diosi³

^{1,2,3} Department of Automation, University of Leoben, Peter-Tunnerstr.27, A-8700 Leoben, Austria

Abstract

A problem associated with teleoperating is that the operator does not have the haptic and/or acoustic feedback which he has while operating the machine directly. The aim of the experimental system is to use force and acoustic feedback to give the operator a better “feel” for what he is doing. In addition to the feedback of real forces virtual force fields i.e. mappings of sensor data to force are also made.

1. Introduction

There is an ever increasing demand to remove personnel from hazardous environment in the working place. This is particularly true in mining. The technique of teleoperating is finding increased application in such areas. The aim of this project is to go beyond simple teleoperating by adding force feedback of real forces and by mapping sensor data to forces. It is also desirable to support the user in complex working processes with partial autonomous actions. This leads not only to an increased safety for the miner but also to a higher productivity. An experimental teleoperating system has been developed to control a tracked vehicle. Tracked vehicles like draglines and cutting machines are used due to their special roadabilities in these rough areas. The system consists of a camera to transmit the visual information and a number of ultrasonic sensors which deliver distance data of obstacles to the users force feedback joystick to give the user a haptic impression of the real scene.

2. Principal of operation

The concept of the system is shown in figure 1.

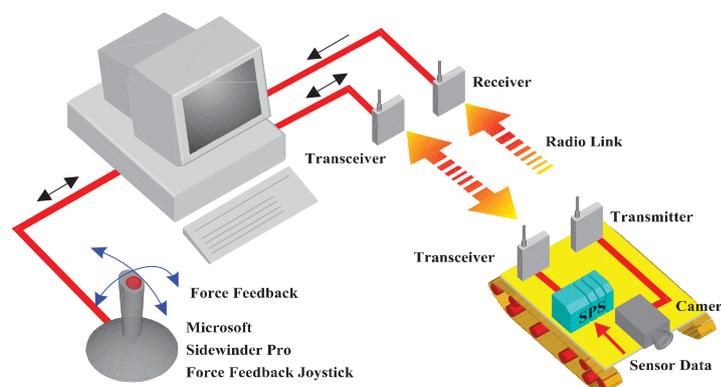


Figure 1. Schematic concept of the system

The system consists of the tracked vehicle with an PLC(Programmable Logic Controller). The PLC is used for all real time control algorithms in motion and navigation. Further, the PLC controls the communication to the operator, for pre-processing of the sensor signals and to control all actuators. This PLC offers a suitable environment of computing the actual data in real time. Only the relevant data are then sent to the operator computer. For the data exchange between the personal computer and the vehicle a bi-directional radio link is installed. A camera which is placed on the vehicle transmits the visual information via radio link to the operator computer. The motions of both vehicle and camera are controlled by an operator joystick.

2.1. Tracked vehicle

In order to test the full functionality of the teleoperating system under laboratory conditions a prototype tracked vehicle has been developed (see figure 2). With this vehicle it was possible to implement navigation algorithms and test the principle characteristics of the system.

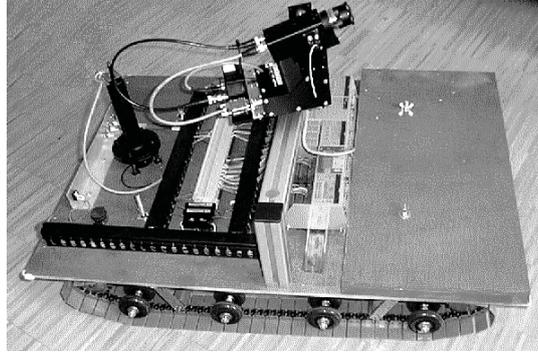


Figure 2. Tracked vehicle

The vehicle is equipped with a standard industry PLC that is programmed in structured text language. The programs on the PLC control the moving of the vehicle as a function of the joystick's position. The information of the joystick is sent via radio link to the receiver unit which is connected to the RS 232 or network interface of the PLC. Further it measures the incoming sensor signals. This sensor data are then pre-processed and the relevant result is sent to the operator terminal via radio link. The PC calculates the data and maps the data to suitable forces. For special navigation procedures, like turning left or right on the place, the user is supported with programs running on the controller. With the assistance of the sensor data from the ultrasonic sensors it is also possible to support further navigation actions such as driving parallel to a wall in a predefined distance or in the middle of a tunnel. By using this sensor data the user gets the information of the distance transformed into a force on the joystick. The user feels the distance when he is driving the vehicle to the wall or an obstacle. The force on the joystick will increase if he drives closer and decrease if the distance increases. This kind of perceptions are far more intuitive than flashing lights and they redirect the users motions away from the hazard.

2.2. Camera system

The camera we use is an intelligent black and white frame camera. An intelligent camera is a camera that has a processor (Digital Signal Processor) on board. With this stand alone solution it is possible to process an image without further framgabber or PC devices. This camera is programmed in a developing program environment. After the download of a program the camera works autonomously. With this camera we can pre-process the image and deliver the computed reduced information to the users monitor. It is also possible to dub information into the users screen. This camera system also supports the navigation of the vehicle for example driving along a marking on the surface of the floor. For further processing of the image it can be still computed in the PC. It also offers flexible I/O's for interfacing to a wide range of peripheral equipment. This camera is mounted on a swivelling device so that it can be moved 180 degrees in horizontal and 90 degrees in vertical direction. The moving of the camera is controlled by the joystick.

2.3. Force feedback joystick

The used joystick is the Microsoft® SideWinder® Force Feedback Pro. This joystick supports the DirectX® 5.0 API (Application Programming Interface). A Force Feedback API is the protocol by which an application communicates with the joystick. It defines a standard set to "call" the force in an application[1]. DirectX provides a consistent interface between hardware and applications, reducing the complexity of installation and configuration and using the hardware to its best advantage. By using the interfaces provided by DirectX, software developers can take advantage of hardware features without being concerned about the implementation details of that hardware. DirectX offers several components that address the performance issues of programming Windows-based applications. One of this components is DirectInput®. It provides support for input devices including the force feedback in- and output devices. The DirectInput object in an application represents the DirectInput subsystem. It is used to enumerate and manage input devices. With this tool it is possible to call the functions of the joystick independently to the monitoring software environment. The used software package only must support the DirectX philosophy. We implemented the joystick in Visual Basic.

2.4. Navigation

The navigation of a tracked vehicle differs from that of a vehicle with wheels. The steering system of a vehicle with wheels is not directly coupled with the motor's velocity. The steering wheel sets the radius of the curve and the motor determines the velocity with which one drives a curve. In our case both motors determine the direction and the velocity of the tracked vehicle's motion. Further, a mapping of the joystick to the track speed is required which can be used both for manual and autonomous control of motion. This requires that the mapping plane is continuous and contains no singularities.

Two planes are defined, one for each motor where the motor speed is a function of the joystick's position. A straight line is the result of cutting these two planes. The diagonal between the right upper and the left lower position of the joystick is an area where the velocity of motor 1 is zero (see figure 3). Moving the joystick to the left upper position the velocity of motor 1 is set to a maximum. This means that the tracked vehicle will drive with 100 percent of its velocity with motor 1 in forward direction. Bringing the joystick in the right lower position will lead to drive with 100 percent with motor 1 backward.

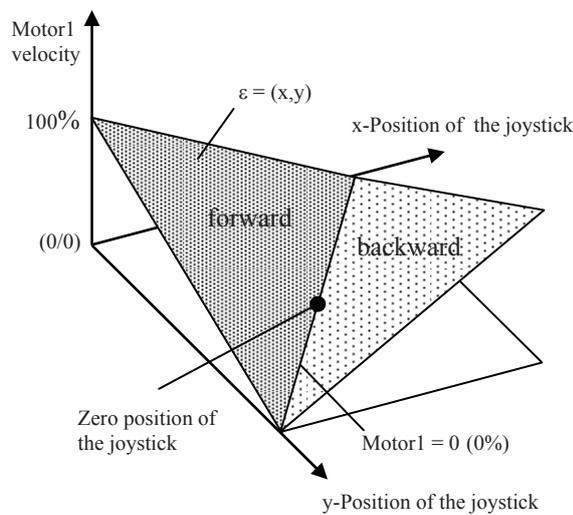


Figure 3. Velocity of motor1 as a function of the joystick's position

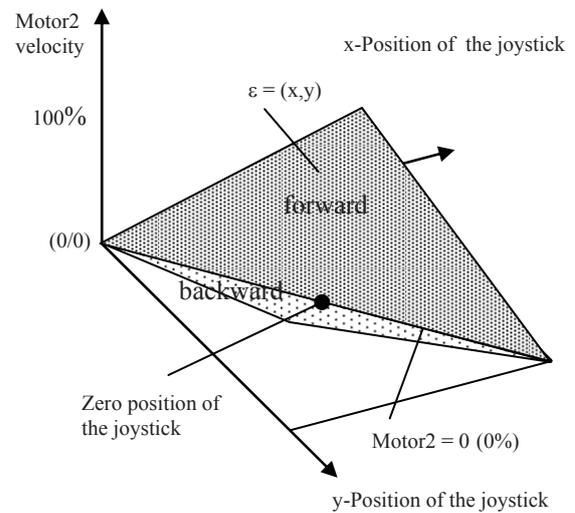


Figure 4. Velocity of motor2 as a function of the joystick's position

A 90 degrees rotation of this plane is used to control motor 2. The resulting mapping is continuous and has no singularities. Further it has the advantage of mapping the joystick to a logical image movement in the teleoperating screen. Note this motion is not a conventional mapping.

By changing of the position from the left upper corner to the left middle position the motion modifies from a left curve into a rotation on place. Carrying on the position to the left lower corner leads to drive a backward right curve for example. Figure 5 shows the driving actions as a function of the determined joystick position for both motors.

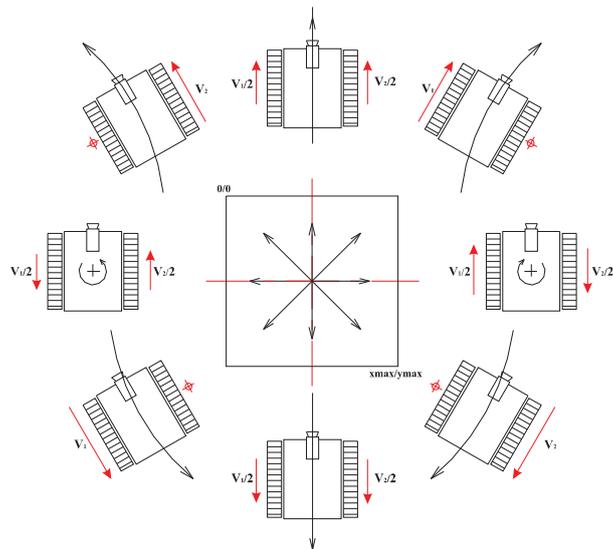


Figure 5. Driving actions as a function of the determined joystick position for both motors

3. Conclusion

This experimental system offers the possibility of testing the individual components of teleoperating under laboratory conditions. It delivers results for further developments not only on hardware but also on implementation of software concepts.

The implementation of the joystick's control via DirectInput makes the system software independent and allows the user to choose the software platform he wants.

The navigation method we use offers an implementation in control algorithms due to its consistent mathematical description and an easy navigation for the operator via joystick.

To complete the navigation system, further sensors like optical gyroscopes and electronic compasses will determine the tasks for investigations in the near future[2].

4. References

- [1] DirectX 5.0 SDK-Manual, Microsoft®
- [2] A. Marek, P. O'Leary, M. Husty, Kinematics and Control Algorithms of Coupled Tracked Vehicles, ISMCR, 8th International Symposium on Measurement and Control in Robotics, June 8-12, 1998, Prague

2 Softwarequelltext

2.1 Visual Basic Quellcode

2.1.1 MDIForm

```

⊘ ++++++
⊘ Θυελλχοδε ζυ ΜΔΙΦορμ
⊘
⊘ Διπλομαρβειτ Τελεοπερατινγ μιτ Φορχε Φεεδβαχκ
⊘ ++++++

```

```

⊘ Κυρζβεσχηρειβυγ :
⊘ Ηαυπτφενστερ δερ ΒεδιενυγσοβερφλTMχηε

```

```

⊘ ζερσιον      : 1.0
⊘ Δατυμ       : 13.07.98
⊘ Αυτορ       : Ροβερτ Διοσι

```

```

⊘ ++++++
⊘ Ινστιτυτ φ|ρ Αυτοματιον
⊘ ΜοντανυνιπερσιτTMτ Λεοβεν
⊘ Πετερ Τυννερ ΣτραUε 27
⊘ 8700 Λεοβεν
⊘ Τελ.:  (+43) 3842 402 9035
⊘ Φαξ.:  (+43) 3842 402 9031
⊘ εμαιλ.: αυτοματιονΞυνιλεοβεν.αχ.ατ
⊘ ++++++

```

```

⊘ Προγραμμεγιινν

```

```

Πριπατε Συβ Χομμανδ1_Χλιχκ()
⊘ Προγραμμτειλ Τ 1.1

```

```

⊘ \φφνεν δερ ΜαινΦορμ
  Λοαδ ΜαινΦορμ
  ΜαινΦορμ.Σηοω

```

```

⊘ ΣτTMνδιγε Αβφραγε δερ θοψστιχκκοορδινατεν
  ΦορΡυντιμε
  Ενδ Συβ

```

```

Πριπατε Συβ ενδε_Χλιχκ()
⊘ Προγραμμτειλ Τ 1.2

```

```

⊘ θοψστιχκ δεινσταλλιερεν
  ΜαινΦορμ.Προφεκτ20_οχξ1.Σηυτδοω
⊘ Ηαυπτπρογραμμι βεενδεν
  Ενδ
  Ενδ Συβ

```

εΠρογραμμαμνδε ΜΔΙΦορμ

2.1.2 Programmteil MainForm

```

ε ++++++
ε Θυελλχοδε ζυ Μαινοφορμ
ε
ε Διπλομαρβειτ Τελεοπερατινγ μιτ Φορχε Φεεδβαχκ
ε ++++++

```

```

ε Κυρζβεσχηρειβυγ :
ε Ιν διεσεμ Φενστερ ιστ εσ μ\γλιχη ζαριαβλεν ζυ πισυαλισιερεν
ε υνδ Γρυνδεινστελλυνγεν πορζυνεημεν
ε Διε εινηγεβεττετεν Οβφεκτε διεενεν δερ Κομμυνικατιον μιτ δεμ
ε ΔΔΕ – Σερπερ Νετ 2000

```

```

ε ζερσιον : 1.0
ε Δατυμ : 13.07.98
ε Αυτορ : Ροβερτ Διοσι

```

```

ε ++++++
ε Ινστιτυτ φ\ρ Αυτοματιον
ε ΜοντανυνιπερσιτTMτ Λεοβεν
ε Πετερ Τυννερ Στρα\ε 27
ε 8700 Λεοβεν
ε Τελ.: (++43) 3842 402 9035
ε Φαξ.: (++43) 3842 402 9031
ε εμαιλ.: αυτοματιον@υνιλεοβεν.αχ.ατ
ε ++++++

```

```

εΠρογραμμεβεινν

```

```

εΑλλε ζαριαβλεν μ\σσειν δεφινιερετ ωερδεν
Οπτιον Εξπλιχιτ

```

```

Πριτωατε Συβ Χμδ_ενδε_Χλιχκ()
εΠρογραμμετειλ T 2.2

```

```

εΚυρζβεσχηρειβυγ :
εΘοψστιχκ δεινσταλλιερεν υνδ Ηαυπτπρογραμμ βεενδεν

```

```

    Προφεκτ20_οχξ1.Σηυτδοων
    Υνλοαδ Μαινοφορμ
εΗαυπτπρογραμμ βεενδεν
    Ενδ
Ενδ Συβ

```

```

Πριτωατε Συβ Χμδ_σιμυλατιον_Χλιχκ()
εΠρογραμμετειλ T 2.1

```

```

εΚυρζβεσχηρειβυγ :
εΑυφρυφ δεσ Φενστερεσ φ\ρ διε ζισυαλισιερυγ δερ Μοτοργεσχηωινδιγκειτεν

```

Λοαδ Σιμυλατιον
 Σιμυλατιον.Σηοω
 Ενδ Συβ

Πριτωατε Συβ Χμδ_τεστ_φορχε_Χλιχκ()
 εΠρογραμμτειλ T2.11

εΚυρζβεσχηρειβυγ :
 εΕρζεευγυγ εινεσ Φεδερεφφεκτεσ μιτ μαξιμαλερ Φεδερκονσταντε

Προφεκτ20_οχξ1.χρεατεΕφφεχτ 10000
 Ενδ Συβ

Πριτωατε Συβ Φορμ_Λοαδ()
 εΠρογραμμτειλ T2.0

εΚυρζβεσχηρειβυγ :
 εΎφφενεν δερ ΔΔΕ-Καν™λε ζυ NET2000 ΔΔΕ-Σερπερ

Ον Ερρορ ΓοΤο ΔΔΕερρορηανδλε

Λαβελ_μοτορ1_σπεεδ.ΛινκΤοπιχ = √NET2ΔΔΕ|χπυ√
 Λαβελ_μοτορ1_σπεεδ.ΛινκΙτεμ = √πω[0]√
 Λαβελ_μοτορ1_σπεεδ.ΛινκΜοδε = 1
 Λαβελ_μοτορ1_σπεεδ.ΛινκΡεθυεστ

Λαβελ_μοτορ1_διρ_βιτ1.ΛινκΤοπιχ = √NET2ΔΔΕ|χπυ√
 Λαβελ_μοτορ1_διρ_βιτ1.ΛινκΙτεμ = √πω[2]√
 Λαβελ_μοτορ1_διρ_βιτ1.ΛινκΜοδε = 1
 Λαβελ_μοτορ1_διρ_βιτ1.ΛινκΡεθυεστ

Λαβελ_μοτορ1_διρ_βιτ2.ΛινκΤοπιχ = √NET2ΔΔΕ|χπυ√
 Λαβελ_μοτορ1_διρ_βιτ2.ΛινκΙτεμ = √πω[3]√
 Λαβελ_μοτορ1_διρ_βιτ2.ΛινκΜοδε = 1
 Λαβελ_μοτορ1_διρ_βιτ2.ΛινκΡεθυεστ

Λαβελ_μοτορ2_σπεεδ.ΛινκΤοπιχ = √NET2ΔΔΕ|χπυ√
 Λαβελ_μοτορ2_σπεεδ.ΛινκΙτεμ = √πω[1]√
 Λαβελ_μοτορ2_σπεεδ.ΛινκΜοδε = 1
 Λαβελ_μοτορ2_σπεεδ.ΛινκΡεθυεστ

Λαβελ_μοτορ2_διρ_βιτ1.ΛινκΤοπιχ = √NET2ΔΔΕ|χπυ√
 Λαβελ_μοτορ2_διρ_βιτ1.ΛινκΙτεμ = √πω[4]√
 Λαβελ_μοτορ2_διρ_βιτ1.ΛινκΜοδε = 1
 Λαβελ_μοτορ2_διρ_βιτ1.ΛινκΡεθυεστ

Λαβελ_μοτορ2_διρ_βιτ2.ΛινκΤοπιχ = √NET2ΔΔΕ|χπυ√
 Λαβελ_μοτορ2_διρ_βιτ2.ΛινκΙτεμ = √πω[5]√
 Λαβελ_μοτορ2_διρ_βιτ2.ΛινκΜοδε = 1
 Λαβελ_μοτορ2_διρ_βιτ2.ΛινκΡεθυεστ

εΚαμερασχηφωενκσψστεμ

Λαβελ_σερπο1.ΛινκΤοπιχ = √NET2ΔΔΕ|χπυ√
 Λαβελ_σερπο1.ΛινκΙτεμ = √πω[6]√

Λαβελ_σερπο1.ΛινκΜοδε = 1
 Λαβελ_σερπο1.ΛινκΡεθυεστ

Λαβελ_σερπο2.ΛινκΤοπιχ = √NET2ΔΔΕ|χπυ√
 Λαβελ_σερπο2.ΛινκΙτεμ = √πω[7]√
 Λαβελ_σερπο2.ΛινκΜοδε = 1
 Λαβελ_σερπο2.ΛινκΡεθυεστ

ΞΚαμερασχηφωενκψστυμ Ενδε

ΞΥλτρασχηαλλσενσορεν ΔΔΕ – Ινιτιαλισιερυγ
 ΞΡεαδ-ονλψ παριαβλεν

Λαβελ_υσ1.ΛινκΤοπιχ = √NET2ΔΔΕ|χπυ√
 Λαβελ_υσ1.ΛινκΙτεμ = √πω[8]√
 Λαβελ_υσ1.ΛινκΜοδε = 1

Λαβελ_υσ2.ΛινκΤοπιχ = √NET2ΔΔΕ|χπυ√
 Λαβελ_υσ2.ΛινκΙτεμ = √πω[9]√
 Λαβελ_υσ2.ΛινκΜοδε = 1

Λαβελ_υσ3.ΛινκΤοπιχ = √NET2ΔΔΕ|χπυ√
 Λαβελ_υσ3.ΛινκΙτεμ = √πω[10]√
 Λαβελ_υσ3.ΛινκΜοδε = 1

Λαβελ_υσ4.ΛινκΤοπιχ = √NET2ΔΔΕ|χπυ√
 Λαβελ_υσ4.ΛινκΙτεμ = √πω[11]√
 Λαβελ_υσ4.ΛινκΜοδε = 1

ΜΔΙΦορμ1.Λιγητ2.ΒαχκΧολορ = &ΗΦΦ00&

Λιστ1.ΑδδΙτεμ (Τιμε & √ Στατυσμελδουγ Κομπονετεν ριχητιγ ινιτιαλισιερτ√)
 Λιστ1.ΛιστΙνδεξ = Λιστ1.ΛιστΧουντ - 1

ΞΛαδεωερτε δερ Μοτορδατεν

Λαβελ_μοτορ1_σπεεδ = 0
 Λαβελ_μοτορ1_σπεεδ.ΛινκΠοκε

Λαβελ_μοτορ2_σπεεδ = 0
 Λαβελ_μοτορ2_σπεεδ.ΛινκΠοκε

Λαβελ_μοτορ1_διρ_βιτ1 = 1
 Λαβελ_μοτορ1_διρ_βιτ1.ΛινκΠοκε

Λαβελ_μοτορ1_διρ_βιτ2 = 0
 Λαβελ_μοτορ1_διρ_βιτ2.ΛινκΠοκε

Λαβελ_μοτορ2_διρ_βιτ1 = 1
 Λαβελ_μοτορ2_διρ_βιτ1.ΛινκΠοκε

Λαβελ_μοτορ2_διρ_βιτ2 = 0
 Λαβελ_μοτορ2_διρ_βιτ2.ΛινκΠοκε

Ον Ερρορ ΓοΤο ΘΟΨερρορηανδλε

```

εΘοψστιχκ ινιτιαλισιερεν
Προφεκτ20_οχξ1.ΙνιτΦΦ
εΚραφτρ|χκκοππελυγγ σταρτεν
Προφεκτ20_οχξ1.ΚοσταντεΚραφτΕρζευγεν 10000, 0

```

```

εΚειν Φεηλερ αυφγετρετεν -> Σετζεν πον Λιγητ1 = γρ|ν
ΜΔΙΦορμ1.Λιγητ1.ΒαχκΧολορ = &ΗΦΦ00&

```

```

εΓρυνδεινστελλυγγ φ|ρ διε Καμερασχηωενκγεσχηωινδιγκειτ
εΔεφινιτιον ιν Στανδαρδσ – Μοδυλε
  Σχηωενκ_Γεσχηωινδιγκειτ = 100
  Τεξτ_σχηωενκ_γεσχηωινδιγκειτ = Σχηωενκ_Γεσχηωινδιγκειτ

```

```

εΠροζεδυρε βεενδεν
Εξιτ Συβ

```

```

εΣπρυνγμαρκε φ|ρ ΔΔΕ – Φεηλερβεηανδλυνγ
ΔΔΕερρορηανδλε: Χαλλ ΔΔΕερρορηανδλε
εΣπρυνγμαρκε φ|ρ Θοψστιχκ Φεηλερβεηανδλυνγ
ΘΟΨερρορηανδλε: Χαλλ ΘΟΨερρορηανδλε

```

Ενδ Συβ

```

Πριπατε Συβ Λαβελ_φοψ_ξ_Χηανγε()
εΠρογραμμειλ Τ2.7

```

```

εΚυρζβεσχηρειβυγγ :
εΘνδερυγγ διεσεσ Ωερτεσ βεωιρκτ διε Νευβερεχηνυγγ δερ Μοτοργεσχηωινδιγκειτεν

```

```

  ΔοΕπεντσ
  προχΣτευερυγγ ζαλ(Λαβελ_φοψ_ξ), ζαλ(Λαβελ_φοψ_ψ)
Ενδ Συβ

```

```

Πριπατε Συβ Λαβελ_φοψ_ψ_Χηανγε()
εΠρογραμμειλ Τ2.8

```

```

εΚυρζβεσχηρειβυγγ :
εΘνδερυγγ διεσεσ Ωερτεσ βεωιρκτ διε Νευβερεχηνυγγ δερ Μοτοργεσχηωινδιγκειτεν

```

```

  ΔοΕπεντσ
  προχΣτευερυγγ ζαλ(Λαβελ_φοψ_ξ), ζαλ(Λαβελ_φοψ_ψ)
Ενδ Συβ

```

```

Πριπατε Συβ Λαβελ_κραφτ_αμπλιτυδε_Χηανγε()
εΠρογραμμειλ 2.9

```

```

εΚυρζβεσχηρειβυγγ :
εΣετζεν δερ Κραφταμπλιτυδε αμ Θοψστιχκ

```

```

ΔοΕπεντσ
  Προφεκτ20_οχξ1.ΚραφτΑμπλιτυδε ΦορχεΑμπλ
Ενδ Συβ

```

Πριτωατε Συβ Λαβελ_κραφτ_ριχητυγγ_Χηανγε()
εΠρογραμμειλ 2.10

εΚυρζβεσχηρειβυγγ :
εΣετζεν δερ Κραφτριχητυγγ αμ θοψστιχκ

ΛοΕπεντσ
Προφεκτ20_οχξ1.ΚραφτΡιχητυγγ ΦορχεΔιρ
Ενδ Συβ

Πριτωατε Συβ Λαβελ_μοτορ1_διρ_βιτ1_Χηανγε()
Ον Ερρορ ΓοΤο ΔΔΕερρορηανδλε
Λαβελ_μοτορ1_διρ_βιτ1.ΛινκΠοκε
Εξιτ Συβ
ΔΔΕερρορηανδλε: Χαλλ ΔΔΕερρορηανδλε
Ενδ Συβ

Πριτωατε Συβ Λαβελ_μοτορ1_διρ_βιτ2_Χηανγε()
Ον Ερρορ ΓοΤο ΔΔΕερρορηανδλε
Λαβελ_μοτορ1_διρ_βιτ2.ΛινκΠοκε
Εξιτ Συβ
ΔΔΕερρορηανδλε: Χαλλ ΔΔΕερρορηανδλε
Ενδ Συβ

Πριτωατε Συβ Λαβελ_μοτορ1_σπεεδ_Χηανγε()
Ον Ερρορ ΓοΤο ΔΔΕερρορηανδλε
Λαβελ_μοτορ1_σπεεδ.ΛινκΠοκε
Εξιτ Συβ
ΔΔΕερρορηανδλε: Χαλλ ΔΔΕερρορηανδλε
Ενδ Συβ

Πριτωατε Συβ Λαβελ_μοτορ2_διρ_βιτ1_Χηανγε()
Ον Ερρορ ΓοΤο ΔΔΕερρορηανδλε
Λαβελ_μοτορ2_διρ_βιτ1.ΛινκΠοκε
Εξιτ Συβ
ΔΔΕερρορηανδλε: Χαλλ ΔΔΕερρορηανδλε
Ενδ Συβ

Πριτωατε Συβ Λαβελ_μοτορ2_διρ_βιτ2_Χηανγε()
Ον Ερρορ ΓοΤο ΔΔΕερρορηανδλε
Λαβελ_μοτορ2_διρ_βιτ2.ΛινκΠοκε
Εξιτ Συβ
ΔΔΕερρορηανδλε: Χαλλ ΔΔΕερρορηανδλε
Ενδ Συβ

Πριτωατε Συβ Λαβελ_μοτορ2_σπεεδ_Χηανγε()
Ον Ερρορ ΓοΤο ΔΔΕερρορηανδλε
Λαβελ_μοτορ2_σπεεδ.ΛινκΠοκε
Εξιτ Συβ
ΔΔΕερρορηανδλε: Χαλλ ΔΔΕερρορηανδλε
Ενδ Συβ

Πριτωατε Συβ Λαβελ_σερπο1_Χηανγε()
Ιφ ζαλ(Λαβελ_σερπο1) < 1000 Τηεν Λαβελ_σερπο1 = 1000
Ιφ ζαλ(Λαβελ_σερπο1) > 10000 Τηεν Λαβελ_σερπο1 = 10000
Λαβελ_σερπο1.ΛινκΠοκε
Ενδ Συβ

Πριτωατε Συβ Λαβελ_σερπο2_Χηανγε()
Ιφ ζαλ(Λαβελ_σερπο2) < 1000 Τηεν Λαβελ_σερπο2 = 1000
Ιφ ζαλ(Λαβελ_σερπο2) > 10000 Τηεν Λαβελ_σερπο2 = 10000
Λαβελ_σερπο2.ΛινκΠοκε
Ενδ Συβ

αΑυφρυφε δεσ ΑχτιπεΞ – Οβφεκτεσ

Πριπατε Συβ Προφεκτ20_οχξ1_ΣΩΦΦΒυττονσ(Βψσαλ βΒ1 Ασ Βοολεαν, Βψσαλ βΒ2 Ασ Βοολεαν, Βψσαλ βΒ3 Ασ Βοολεαν, Βψσαλ βΒ4 Ασ Βοολεαν)

Λαβελ_βυτ1 = ΧΙντ(Αβσ(βΒ1))

Λαβελ_βυτ2 = ΧΙντ(Αβσ(βΒ2))

Λαβελ_βυτ3 = ΧΙντ(Αβσ(βΒ3))

Λαβελ_βυτ4 = ΧΙντ(Αβσ(βΒ4))

Ενδ Συβ

Πριπατε Συβ Προφεκτ20_οχξ1_ΣΩΦΦΗΑΤΒυττον(Βψσαλ λσαλ Ασ Λονγ)

ΔοΕπεντσ

αΑυσγαβε δεσ Ωερτεσ αυφ Λαβελ

Λαβελ_ηατ = λσαλ

αΦ|ηρτ Βερεχηνυνγεν φ|ρ Καμερασχηωενκσψστεμ αυσ

ΚαμεραΣχηωενκ λσαλ

Ενδ Συβ

Πριπατε Συβ Προφεκτ20_οχξ1_ΣΩΦΦΚοορδινατεν(Βψσαλ λΞ Ασ Λονγ, Βψσαλ λΨ Ασ Λονγ, Βψσαλ λΡ ζ Ασ Λονγ)

αΚυρζβεσχηρειβυγ :

αβερ δασ ΑχτιπεΞ – Οβφεκτ ειναλνγενδε θοψστιχκκοορδινατεν

Λαβελ_φοψ_ξ = λΞ

Λαβελ_φοψ_ψ = λΨ

Λαβελ_φοψ_ρζ = λΡζ

Ενδ Συβ

Πριπατε Συβ Προφεκτ20_οχξ1_ΣΩΦΦΣχηυβ(Βψσαλ λσαλ Ασ Λονγ)

Λαβελ_σχηυβ = λσαλ

ΜΑΞ_ΣΠΕΕΔ = ΧΙντ(100 * λσαλ / 65535)

αΥμρεχηνυνγ ιν Προζενταναβαβεν

Λαβελ_μαξ_σπεεδ = ΧΙντ(100 * λσαλ / 65535)

Ενδ Συβ

Πριπατε Συβ Τεξτ_Σχηωενκ_γεσχηωινδιγκειτ_ΚεψΠρεσσ(ΚεψΑσχι Ασ Ιντεγερ)

αΠρογραμμειλ Τ 2.4

αΚυρζβεσχηρειβυγ :

αΤεξτεινγαβε φ|ρ Σχηωενκγεσχηωινδιγκειτ δερ Σερωμοτορεν

αΥμ διε Τεξτεινγαβε ζυ βεενδεν ΡετυρνΤαστε δρ|χκεν = χηρ(13)

Ιφ ΚεψΑσχι = 13 Τηεν

αΒερεχησ|βερπρ|φυγγ δερ Ειναβαβε

Ιφ σαλ(Τεξτ_σχηωενκ_γεσχηωινδιγκειτ) < 0 Τηεν Τεξτ_σχηωενκ_γεσχηωινδιγκειτ = 0

Ιφ σαλ(Τεξτ_σχηωενκ_γεσχηωινδιγκειτ) > 1000 Τηεν Τεξτ_σχηωενκ_γεσχηωινδιγκειτ = 1000

αΒερεχησ|βερπρ|φυγγ Ενδε

Σχηωενκ_Γεσχηωινδιγκειτ = Τεξτ_σχηωενκ_γεσχηωινδιγκειτ

αΦοχυσ αν ζσχρολλ |βεργεβεν

ζΣχρολλ1.ΣετΦοχυσ

ζΣχρολλ1.ζαλυε = σαλ(Τεξτ_σχηωενκ_γεσχηωινδιγκειτ)

Ενδ Ιφ

Ενδ Συβ

Πριτωατε Συβ Τιμερ1_Τιμερ()

 εΠρογραμμειλ T 2.3

εΚυρζβεσχηρειβυγ :

 εΑυφρυφ δερ Ρουτινε φ) ρ διε Βερεχηνυγ δερ Φορχε Φεεδβαχκ Δατεν

Χαλχζεχτ ζαλ(Λαβελ_υσ1), ζαλ(Λαβελ_υσ2), ζαλ(Λαβελ_υσ3), ζαλ(Λαβελ_υσ4)

 Ενδ Συβ

Πριτωατε Συβ ζΣχρολλ1_Χηανγε()

 εΠρογραμμειλ T2.5

εΚυρζβεσχηρειβυγ :

 εΘνδερυγ δερ Σχηωενκγεσχηωινδιγκειτ δερ Σερωμοτορεν

Τεξτ_σχηωενκ_γεσχηωινδιγκειτ = ζΣχρολλ1.ζαλυε

 Σχηωενκ_Γεσχηωινδιγκειτ = ζΣχρολλ1.ζαλυε

 Ενδ Συβ

εΦηλερβεηανδλυγσρουτινεν

Πριτωατε Συβ ΔΔΕερρορηανδλε()

 εΚυρζβεσχηρειβυγ :

 εΦηλερβεηανδλυγσρουτινε βει ΔΔΕ-Φηλερν

Λιστ1.ΑδδΙτεμ (Τιμε & ∇ ΔΔΕ – Ερρορ αυφετρετεν – Νευνιτιαλισιερυγ νοτωενδιγ !∇)

 Λιστ1.ΛιστΙνδεξ = Λιστ1.ΛιστΧουντ – 1

εΦηλερμελδυγ αυσεβεν

 ΜσγΒοξ ∇ΔΔΕ-ζερβινδυγ ζυ NET2000 νιχητ μ)γλιχη ∇ + Χηρ(13) + ∇Προγραμμ ωιρδ αυφγρυνδ εινεσ

 υγ)λτιγεν Ζυστανδεσ αβγεβροχηγ ∇, πβΧριτιχαλ, ∇Ωαρνμελδυγ∇

 εΘοψστιχκ δεινσταλλιερεν

 Προφεκτ20_οχξ1.Σηυτδοων

 εΗαυπππρογραμμ βεενδεν

 Ενδ

Ενδ Συβ

Πριτωατε Συβ ΘΟΨερρορηανδλε()

 εΚυρζβεσχηρειβυγ :

 εΦηλερβεηανδλυγσρουτινε βει Θοψστιχκφεηλερν

Λιστ1.ΑδδΙτεμ (Τιμε & ∇ Θοψστιχκ περυρσαχητε Φηλερ∇)

 Λιστ1.ΛιστΙνδεξ = Λιστ1.ΛιστΧουντ – 1

ΜσγΒοξ ∇Μιχροσοφτ Σιδεωινδερ Θοψστιχκ∇ + Χηρ(13) + ∇περυρσαχητε εινεν Φηλερ∇, πβΧριτιχα

 λ

Ενδ Συβ

εΠρογραμμενδε ΜαινΦορμ

2.1.3 Globale Funktionen und Prozeduren

```

ε ++++++
ε Θυελλχοδε ζυ Στευερυγγεν
ε
ε Διπλομαρβειτ Τελεοπερατιγγ μιτ Φορχε Φεεδβαγκ
ε ++++++

```

```

ε Κυρζβεσχηρειβυγγ :
ε Γλοβαλε Φυνκτιονεν
ε αυφγερυφεν δυρχη Οβφεκτε δερ περωενδετεν Φενστερ

```

```

ε ζερσιον : 1.0
ε Δατυμ : 13.07.98
ε Αυτορ : Ροβερτ Διοσι

```

```

ε ++++++
ε Ινστιτυτ φ|ρ Αυτοματιον
ε ΜοντανυνιπερσιτTM Λεοβεν
ε Πετερ Τυννερ ΣτραUε 27
ε 8700 Λεοβεν
ε Τελ.: (++43) 3842 402 9035
ε Φαξ.: (++43) 3842 402 9031
ε εμαιλ.: αυτοματιονΞυνιλεοβεν.αχ.ατ
ε ++++++

```

```

εΠρογραμμβεγγινν

```

```

εΑλλε ζαριαβλεν μ|σσην δεφινιερετ ωερδεν
Οπτιον Εξπλιχιτ

```

```

Χονστ Πι = 3.14159

```

```

Πυβλιχ ΦορχεΑμπλ Ασ Λογγ
Πυβλιχ ΦορχεΔιρ Ασ Λογγ

```

```

Τυπε ζΕΚΤΟΡ εΤυπενδεκλαρατιον φ|ρ Κραφτπεκτορ αν ΦΦ Θοψστιχη
Ξ Ασ Ιντεγγερ εξ – Κομπονεντε δερ Κραφτ
Ψ Ασ Ιντεγγερ εψ – Κομπονεντε δερ Κραφτ
Ενδ Τυπε

```

```

Πυβλιχ Φ1_2 Ασ ζΕΚΤΟΡ εΡεσυλτιερενδ αυσ δεν Σενσορεν ΥΣ1 + ΥΣ2
Πυβλιχ Φ3 Ασ ζΕΚΤΟΡ εΡεσυλτιερενδ αυσ δεμ Σενσορ ΥΣ3
Πυβλιχ Φ4 Ασ ζΕΚΤΟΡ εΡεσυλτιερενδ αυσ δεμ Σενσορ ΥΣ4

```

```

Πυβλιχ Φ_Νειγγυγγ Ασ Ιντεγγερ

```

```

εΜαξιμαλε Κεττεγγεσχηωινδιγγκειτεν ιν [0–100 %]
Πυβλιχ ΜΑΞ_ΣΠΕΕΔ Ασ Λογγ

```

εΚαμερασχηωενκγεσχηωινδιγκειτ
 εΓρυνδεινστελλυγ Σχηωενκ_Γεσχηωινδιγκειτ = 100
 Πυβλιχ Σχηωενκ_Γεσχηωινδιγκειτ Ασ Ιντεγερ

Πυβλιχ Συβ ΦορΡυντιμε()
 εΠρογραμμειλ Σ 1.1

εΚυρζβεσχηρειβυγ :
 εΔιεσε Ρουτινε ωιρδ ζυρ Λαυφζειτ στTMνδιγ αυφγερυφεν υμ διε
 εΘωψστιχκκοορδινατεν υνδ Θωψστιχκβυττονσ αυσζυλεσεν

ΔοΕπεντσ
 Ωηιλε Err = 0
 ΔοΕπεντσ
 ΜαινΦορμ.Προφεκτ20_οχξ1.Γετθωψ
 Ωενδ
 Ενδ Συβ

Πυβλιχ Συβ προχσιμ()
 εΚυρζβεσχηρειβυγ :
 εΗιερ ωερδεν διε Μοτοργεσχηωινδιγκειτεν πισυαλισιερτ
 εΔαζυ ιστ Φορμ : Σιμυλατιον νοτωενδιγ
 εΔιρεκτηρ Ζυγριφφ αυφ Φορμ Σιμυλατιον

Διμ μ1β1, μ1β2, μ2β1, μ2β2 Ασ Ιντεγερ
 Διμ μ1σπεεδ, μ2σπεεδ Ασ Λονγ

ε←βερναημε δερ Μοτορδατεν αυσ ΜαινΦορμ

Ωιτη ΜαινΦορμ
 μ1β1 = ζαλ(.Λαβελ_μοτορ1_διρ_βιτ1)
 μ1β2 = ζαλ(.Λαβελ_μοτορ1_διρ_βιτ2)
 μ2β1 = ζαλ(.Λαβελ_μοτορ2_διρ_βιτ1)
 μ2β2 = ζαλ(.Λαβελ_μοτορ2_διρ_βιτ2)

 μ1σπεεδ = ζαλ(.Λαβελ_μοτορ1_σπεεδ)
 μ2σπεεδ = ζαλ(.Λαβελ_μοτορ2_σπεεδ)
 Ενδ Ωιτη

Ωιτη Σιμυλατιον

Ιφ μ1β1 = 0 Ανδ μ1β2 = 1 Τηεν .Λινε1.Ψ1 = 2160 + 0.055 * μ1σπεεδ
 Ιφ μ2β1 = 0 Ανδ μ2β2 = 1 Τηεν .Λινε2.Ψ1 = 2160 + 0.055 * μ2σπεεδ

Ιφ μ1β1 = 1 Ανδ μ1β2 = 0 Τηεν .Λινε1.Ψ1 = 2160 - 0.055 * μ1σπεεδ
 Ιφ μ2β1 = 1 Ανδ μ2β2 = 0 Τηεν .Λινε2.Ψ1 = 2160 - 0.055 * μ2σπεεδ

.μοτορρεχητσ.Τοπ = .Λινε1.Ψ1
 .μοτορλινκσ.Τοπ = .Λινε2.Ψ1

.μοτορρεχητσ = ΧΣτρ(ΧΙντ(μ1σπεεδ * 0.0031)) + √%√
 .μοτορλινκσ = ΧΣτρ(ΧΙντ(μ2σπεεδ * 0.0031)) + √%√

Ενδ Ωιτη
Ενδ Συβ

Πυβλιχ Συβ προχΣτευερυνη(Βψζαλ Ξ Ασ Δουβλε, Βψζαλ Ψ Ασ Δουβλε)

εΐν διεσερ Συβ ωερδεν διε Μοτοργεσχηωινδιγκειτεν ερρεχηνετ, ωελχηε αυσ
εδεν θοψστιχκκορδινατεν ρεσυλτιερεν
εΔαζυ ωερδεν Εβενεν αυφγεσπανντ (Σιεηε Βεσχηρειβυνη !)

Διμ Ξ1 Ασ Δουβλε
Διμ Ψ1 Ασ Δουβλε
Διμ ΜΞ Ασ Δουβλε
Διμ Μ1, Μ2, μαξ_σπεεδ_π Ασ Δουβλε

εζερσχηειβυνη δεσ Υρσπρυνησ δερ θοψστιχκκορδινατεν [0..65535]
Ξ1 = Ξ - 32736
Ψ1 = Ψ - 32736

εΕρρεχηνεν δερ Εβενεν (Μοτοργεσχηωινδιγκειτεν)
Μ1 = (Ξ + Ψ) / 2 - 32736
Μ2 = (Ξ - Ψ) / 2

εΒεστιμμυνη δεσ Δρεηριχηρυνησωεχησελσ δερ Αντριεβσμοτορεν
Ιφ Μ2 <= 0 Τηεν
 ΣετΜοτορ2Βιτ 0, 1
 Ελσε
 ΣετΜοτορ2Βιτ 1, 0
Ενδ Ιφ

Ιφ Μ1 <= 0 Τηεν
 ΣετΜοτορ1Βιτ 1, 0
 Ελσε
 ΣετΜοτορ1Βιτ 0, 1
Ενδ Ιφ

εΜΑΞ_ΣΠΕΕΔ ιστ ειν Φακτορ ιμ Βερειχη 0..100%
μαξ_σπεεδ_π = ΜΑΞ_ΣΠΕΕΔ / 100

Ωιτη ΜαινΦορμ
 .Λαβελ_μοτορ1_σπεεδ = ΧΛνη(Αβσ(Μ1 * μαξ_σπεεδ_π))
 .Λαβελ_μοτορ2_σπεεδ = ΧΛνη(Αβσ(Μ2 * μαξ_σπεεδ_π))
Ενδ Ωιτη

εΓραφισχηε Δαρστελλυνη δερ Μοτοργεσχηωινδιγκειτεν
 Χαλλ προχσιμ
Ενδ Συβ

Πυβλιχ Συβ ΚαμεραΣχηωενκ(Βψζαλ ΗΑΤΒυττον Ασ Λονη)
εΚυρζβεσχηρειβυνη :
εΔιε ζαριαβλε ΗατΒυττον αμ θοψστιχκ εργιβτ ωερτε ιν 1/100°
εΔιε Ριχητυνη δεσ ΗΑΤΒυττον βεωιρκτ εινε Θνδερυνη δερ Σερωο - ζαριαβλεν

Ωιτη ΜαινΦορμ

Σελεχτ Χασε ΗΑΤΒυττον

Χασε 0

.Λαβελ_σερπο1 = ζαλ(.Λαβελ_σερπο1) + Σχηωενκ_Γεσχηωινδιγκειτ
Χασε 18000

.Λαβελ_σερπο1 = ζαλ(.Λαβελ_σερπο1) – Σχηωενκ_Γεσχηωινδιγκειτ
Χασε 27000

.Λαβελ_σερπο2 = ζαλ(.Λαβελ_σερπο2) + Σχηωενκ_Γεσχηωινδιγκειτ
Χασε 9000

.Λαβελ_σερπο2 = ζαλ(.Λαβελ_σερπο2) – Σχηωενκ_Γεσχηωινδιγκειτ

Χασε 31500

.Λαβελ_σερπο1 = ζαλ(.Λαβελ_σερπο1) + Σχηωενκ_Γεσχηωινδιγκειτ

.Λαβελ_σερπο2 = ζαλ(.Λαβελ_σερπο2) + Σχηωενκ_Γεσχηωινδιγκειτ

Χασε 4500

.Λαβελ_σερπο1 = ζαλ(.Λαβελ_σερπο1) + Σχηωενκ_Γεσχηωινδιγκειτ

.Λαβελ_σερπο2 = ζαλ(.Λαβελ_σερπο2) – Σχηωενκ_Γεσχηωινδιγκειτ

Χασε 13500

.Λαβελ_σερπο1 = ζαλ(.Λαβελ_σερπο1) – Σχηωενκ_Γεσχηωινδιγκειτ

.Λαβελ_σερπο2 = ζαλ(.Λαβελ_σερπο2) – Σχηωενκ_Γεσχηωινδιγκειτ

Χασε 22500

.Λαβελ_σερπο1 = ζαλ(.Λαβελ_σερπο1) – Σχηωενκ_Γεσχηωινδιγκειτ

.Λαβελ_σερπο2 = ζαλ(.Λαβελ_σερπο2) + Σχηωενκ_Γεσχηωινδιγκειτ

Ενδ Σελεχτ

Ενδ Ωιτη

Ενδ Συβ

Πυβλιχ Φυνηχιον ΣευσΤοθοψ(Σενσ Ασ Δουβλε) Ασ Λονγ

εΚυρζβεσχηρειβυγ :

εΒερεχηνυγ πον θοψστιχκσιγναλεν ιν 0..10000

εΡεσυλτιερενδ αυσ δεν Υλτρασχηαλλσιγναλεν

$$\text{ΣενσΤοθοψ} = \text{ΧΛνγ}(10000 - \text{Σενσ} / 3.2767) + 1$$

Ενδ Φυνηχιον

Πυβλιχ Συβ Χαλχσεχτ(Βψσαλ ΥΣ1 Ασ Λονγ, Βψσαλ ΥΣ2 Ασ Λονγ, Βψσαλ ΥΣ3 Ασ Λονγ, Βψσαλ ΥΣ4 Ασ Λονγ)

εΚυρζβεσχηρειβυγ :

εΡρεχηνυγ δεσ ρεσυλτιερενδεν Φορχε Φεεδβαχκ ζεκτορσ

Διμ ανγλε, ρεσανγλε Ασ Σινγλε

Διμ ΦορχεΑμπλ20, ΦορχεΔιρΡΑΔ Ασ Σινγλε

Διμ Φ1, Φ2, Φ3, Φ4 Ασ Λονγ

Διμ ΦΞ, ΦΨ, ΦΓ Ασ Λονγ

Διμ Ω1, Ω2 Ασ Σινγλε

εΥμερχηνεν δερ Σευστορωερτε αυφ Κραφτπεκτορωερτε

Φ1 = ΣευσΤοθοψ(ΥΣ1)

Φ2 = ΣευσΤοθοψ(ΥΣ2)

Φ3 = ΣευσΤοθοψ(ΥΣ3)

Φ4 = ΣευσΤοθοψ(ΥΣ4)

εΡεσυλτιερενδε ηοριζονταλε Κραφτ

$$\Phi\Xi = \Phi3 - \Phi4$$

εΡεσυλτιερενδε περτικαλε Κραφτ

$$\Phi\Psi = \text{μαξ}(\text{ΧΛνγ}(\Phi1), \text{ΧΛνγ}(\Phi2))$$

εΑνστελλωινκελ δεσ Κραφτπεεκτορσ

$$\Omega1 = (180 / \Pi) * \text{Ατν}(\Phi\Xi / (\Phi\Psi + 1\text{E}-38))$$

$$\Omega2 = (360 - \Omega1) \text{Μοδ } 360$$

εζεεκτορβετραγ αυσ ηοριζονταλερ υνδ περτικαλερ Κραφτ

$$\Phi\Gamma = \text{μαξ}(\text{ΧΛνγ}(\text{Αβσ}(\Phi\Xi)), \text{ΧΛνγ}(\Phi\Psi))$$

εΥμρεχηνυγ δερ Κραφτριχητυνγ

$$\text{ΦορχεΔιρ} = \Omega2 * 100$$

εΣεεζεν δερ Κραφταμπλιτυδε

$$\text{ΦορχεΑμπλ} = \Phi\Gamma$$

Ωιτη ΜαινΦορμ

εΣεεζεν δερ Φορχε – Φεεδβαχκ ζαριαβλεν

εΑυσγαβε δερ Ωερτε

$$\text{.Λαβελ_κραφτ_ριχητυνγ} = \text{ΦορχεΔιρ}$$

εΚραφτριχητυνγ υνδ Κραφταμπλιτυδε

$$\text{.Λαβελ_κραφτ_αμπλιτυδε} = \text{ΦορχεΑμπλ}$$

εΡεινε ζισυαλισιερυνγσαυφγαβεν

$$\text{ΦορχεΑμπλ20} = \text{ΦορχεΑμπλ} / 20$$

ε180° υνδ 1/100%

$$\text{ΦορχεΔιρΡΑΔ} = \text{ΦορχεΔιρ} * \Pi / 18000$$

$$\text{.ζειγερ.}\Psi2 = \text{.ζειγερ.}\Psi1 - (\text{ΦορχεΑμπλ20}) * \text{Χοσ}(\text{ΦορχεΔιρΡΑΔ})$$

$$\text{.ζειγερ.}\Xi2 = \text{.ζειγερ.}\Xi1 + (\text{ΦορχεΑμπλ20}) * \text{Σιν}(\text{ΦορχεΔιρΡΑΔ})$$

Ενδ Ωιτη

Ενδ Συβ

εΣχηνιτσελελενκομμυνικατιονν μιτ ΜαινΦορμ

εδαβει ωερδεν διε Μοτορ–ΡΙχητυνγσβιτσ διρεκτ αυφ ΜαινΦορμ αυφγερυφεν

Πυβλιχ Συβ ΣετΜοτορ1Βιτ(Βψζαλ βιτ1 Ασ Ιντεγερ, Βψζαλ βιτ2 Ασ Ιντεγερ)

εΚυρζβεσχηρειβυγ :

εΚομμυνικατιονσπροζεδυρ μιτ ΜαινΦορμ

εΩιρδ πον δερ Συβ προχΣτευερυνγ αυφγερυφεν

Ωιτη ΜαινΦορμ

$$\text{.Λαβελ_μοτορ1_διρ_βιτ1} = \text{βιτ1}$$

$$\text{.Λαβελ_μοτορ1_διρ_βιτ2} = \text{βιτ2}$$

Ενδ Ωιτη

Ενδ Συβ

Πυβλιχ Συβ ΣετΜοτορ2Βιτ(Βψζαλ βιτ1 Ασ Ιντεγερ, Βψζαλ βιτ2 Ασ Ιντεγερ)

εΚυρζβεσχηρειβυγ :

εΚομμυνικατιονσπροζεδυρ μιτ ΜαινΦορμ

εΩιρδ πον δερ Συβ προχΣτευερυνγ αυφγερυφεν

Ωιτη ΜαινΦορμ

.Λαβελ_μοτορ2_δισ_βιτ1 = βιτ1
.Λαβελ_μοτορ2_δισ_βιτ2 = βιτ2
Ενδ Ωιτη
Ενδ Συβ

εΠρογραμμαμενδε Στευερυνγεν

2.2 Visual C++ Quelltext

```
// ++++++
// Προγραμμα Προφεκτ20_οχξΧτλ.χππ
//
// Διπλομαρβειτ Τελεοπερατινγ μιτ Φορχε Φεεδβαχκ
// ++++++

// Κυρζβεσχηρειβυγ :
// Φορχε Φεεδβαχκ Φυνκτιονεν φ}ρ διε Κομμυνικατιον
// μιτ δεμ Σιδεωινδερ Φορχε Φεεδβαχκ Θουστιχκ
// Δασ Γρυνδγερ}στ δερ ΑχτιπεΞ – Κομπονεντε ωιρδ δυρχη ΜΦΧ Μαναγερ ερστελλτ

// ζερσιον : 1.0
// Δατυμ : 13.07.98
// Αυτορ : Ροβερτ Διοσι

// ++++++
// Ινστιτυτ φ}ρ Αυτοματιον
// Μοντανυνιπερσιτ™τ Λεοβεν
// Πετερ Τυννερ Στρα}ε 27
// 8700 Λεοβεν
// Τελ.: (++43) 3842 402 9035
// Φαξ.: (++43) 3842 402 9031
// εμαιλ.: αυτοματιον}υνιλεοβεν.αχ.ατ
// ++++++

// Προγραμμβεγινν

// Βεγινν δεσ αυτοματισχη ερζευγτεν Γρυνδγερ}στεσ δερ ΑχτιπεΞ–Κομπονεντε

#ινχλυδε }στδαφξ.η}
#ινχλυδε }προφεκτ20_οχξ.η}
#ινχλυδε }Προφεκτ20_οχξΧτλ.η}
#ινχλυδε }Προφεκτ20_οχξΠπγ.η}

#ινχλυδε }σω_φορχε.η}
#ινχλυδε }φορχεφεεδβαχκδεμο.η}

ΛΠΔΙΡΕΧΤΙΝΠΥΤ γ_πΔΙ = ΝΥΛΛ;
ΛΠΔΙΡΕΧΤΙΝΠΥΤΔΕζΙΧΕ2 γ_πΔΙΔεπιχε = ΝΥΛΛ;

#ιφδεφ_ΔΕΒΥΓ
#δεφινε νεω ΔΕΒΥΓ_ΝΕΩ
#υνδεφ ΤΗΙΣ_ΦΙΛΕ
στατιχ χηαρ ΤΗΙΣ_ΦΙΛΕ[] = __ΦΙΛΕ__;
#ενδιφ
```

```
IMPLEMENT_ΔΨΝΧΡΕΑΤΕ(ΧΠροφεκτ20_οχξΧτρλ, ΧΟλεΧοντρολ)
```

```
// Μεσσαγε μαπ
```

```
BEGIN_ΜΕΣΣΑΓΕ_ΜΑΠ(ΧΠροφεκτ20_οχξΧτρλ, ΧΟλεΧοντρολ)
//{{ΑΦΞ_ΜΣΓ_ΜΑΠ(ΧΠροφεκτ20_οχξΧτρλ)
// NOTE – ΧλασσΩιζαρδ ωιλλ αδδ ανδ ρεμοπω μεσσαγε μαπ εντριεσ
// ΔΟ NOT ΕΔΙΤ ωηατ νου σεε ιν τηεσε βλογκσ οφ γενερατεδ χοδε !
//}}ΑΦΞ_ΜΣΓ_ΜΑΠ
ΟΝ_ΜΕΣΣΑΓΕ(ΟΧΜ_ΧΟΜΜΑΝΔ, ΟνΟχμΧομμανδ)
ΟΝ_ΟΛΕςΕΡΒ(ΑΦΞ_ΙΔΣ_ςΕΡΒ_ΠΡΟΠΕΡΤΙΕΣ, ΟνΠροπερτιεσ)
ΕΝΔ_ΜΕΣΣΑΓΕ_ΜΑΠ()
```

```
// Δισπατχη μαπ
```

```
BEGIN_ΔΙΣΠΑΤΧΗ_ΜΑΠ(ΧΠροφεκτ20_οχξΧτρλ, ΧΟλεΧοντρολ)
//{{ΑΦΞ_ΔΙΣΠΑΤΧΗ_ΜΑΠ(ΧΠροφεκτ20_οχξΧτρλ)
ΔΙΣΠ_ΦΥΝΧΤΙΟΝ(ΧΠροφεκτ20_οχξΧτρλ, √ΙνιτΦΦ√, ΙνιτΦΦ, ςΤ_ΕΜΠΤΨ, ςΤΣ_NONE)
ΔΙΣΠ_ΦΥΝΧΤΙΟΝ(ΧΠροφεκτ20_οχξΧτρλ, √ΧρεατεΕφφεχτ√, χρεατεΕφφεχτ, ςΤ_ΕΜΠΤΨ, ςΤΣ_I4)
ΔΙΣΠ_ΦΥΝΧΤΙΟΝ(ΧΠροφεκτ20_οχξΧτρλ, √Γετθου√, Γετθου, ςΤ_ΕΜΠΤΨ, ςΤΣ_NONE)
ΔΙΣΠ_ΦΥΝΧΤΙΟΝ(ΧΠροφεκτ20_οχξΧτρλ, √Σηυτδοων√, Σηυτδοων, ςΤ_ΕΜΠΤΨ, ςΤΣ_NONE)
ΔΙΣΠ_ΦΥΝΧΤΙΟΝ(ΧΠροφεκτ20_οχξΧτρλ, √ΚοσταντεΚραφτΕρζευγεν√, ΚοσταντεΚραφτΕρζευγεν,
ςΤ_ΕΜΠΤΨ, ςΤΣ_I4 ςΤΣ_I4)
ΔΙΣΠ_ΦΥΝΧΤΙΟΝ(ΧΠροφεκτ20_οχξΧτρλ, √ΚραφτΡιχητυνγ√, ΚραφτΡιχητυνγ, ςΤ_ΕΜΠΤΨ, ςΤΣ_I4)
ΔΙΣΠ_ΦΥΝΧΤΙΟΝ(ΧΠροφεκτ20_οχξΧτρλ, √ΚραφτΑμπλιτυδε√, ΚραφτΑμπλιτυδε, ςΤ_ΕΜΠΤΨ, ςΤΣ_I4
)
//}}ΑΦΞ_ΔΙΣΠΑΤΧΗ_ΜΑΠ
ΔΙΣΠ_ΦΥΝΧΤΙΟΝ_ΙΔ(ΧΠροφεκτ20_οχξΧτρλ, √ΑβουτΒοξ√, ΔΙΣΠΙΔ_ΑΒΟΥΤΒΟΞ, ΑβουτΒοξ, ςΤ_ΕΜ
ΠΤΨ, ςΤΣ_NONE)
ΕΝΔ_ΔΙΣΠΑΤΧΗ_ΜΑΠ()
```

```
// Επειντ μαπ
```

```
BEGIN_ΕςΕΝΤ_ΜΑΠ(ΧΠροφεκτ20_οχξΧτρλ, ΧΟλεΧοντρολ)
//{{ΑΦΞ_ΕςΕΝΤ_ΜΑΠ(ΧΠροφεκτ20_οχξΧτρλ)
ΕςΕΝΤ_ΧΥΣΤΟΜ(∇θουΔαταΤεστ√, ΦιρεθουΔαταΤεστ, ςΤΣ_I4 ςΤΣ_I4 ςΤΣ_I4 ςΤΣ_I4 ςΤΣ_I4 ςΤΣ_I
4 ςΤΣ_I4 ςΤΣ_I4 ςΤΣ_I4)
ΕςΕΝΤ_ΧΥΣΤΟΜ(∇Στατυσ√, ΦιρεΣτατυσ, ςΤΣ_BOOLEAN)
ΕςΕΝΤ_ΧΥΣΤΟΜ(∇ΣΩΦΦΚοορδινατεν√, ΦιρεΣΩΦΦΚοορδινατεν, ςΤΣ_I4 ςΤΣ_I4 ςΤΣ_I4)
ΕςΕΝΤ_ΧΥΣΤΟΜ(∇ΣΩΦΦΗΑΤΒυττον√, ΦιρεΣΩΦΦΗΑΤΒυττον, ςΤΣ_I4)
ΕςΕΝΤ_ΧΥΣΤΟΜ(∇ΣΩΦΦΣχηυβ√, ΦιρεΣΩΦΦΣχηυβ, ςΤΣ_I4)
ΕςΕΝΤ_ΧΥΣΤΟΜ(∇ΣΩΦΦΒυττονσ√, ΦιρεΣΩΦΦΒυττονσ, ςΤΣ_BOOLEAN ςΤΣ_BOOLEAN ςΤΣ_BOOLEAN ςΤΣ_B
ΟΟΛ)
//}}ΑΦΞ_ΕςΕΝΤ_ΜΑΠ
ΕΝΔ_ΕςΕΝΤ_ΜΑΠ()
```

```

// Προπερτη παγεσ

BEGIN_ΠΡΟΠΠΑΓΕΙΔΣ(ΧΠροφεκτ20_οχξΧτρλ, 1)
ΠΡΟΠΠΑΓΕΙΔ(ΧΠροφεκτ20_οχξΠροπΠαγε::γυιδ)
ΕΝΔ_ΠΡΟΠΠΑΓΕΙΔΣ(ΧΠροφεκτ20_οχξΧτρλ)

// Ινιτιαλιζε χλασσ φαχτορψ ανδ γυιδ
ΙΜΠΛΕΜΕΝΤ_ΟΛΕΧΡΕΑΤΕ_ΕΞ(ΧΠροφεκτ20_οχξΧτρλ, √ΠΡΟΘΕΚΤ20ΟΧΞ.Προφεκτ20οχξΧτρλ.1√,
    0ξχα09ε426, 0ξ1011, 0ξ11δ2, 0ξα7, 0ξδ, 0, 0ξ60, 0ξ97, 0ξ3ε, 0ξ2β, 0ξ98)

    // Τυπε λιβραρψ ΙΔ ανδ περσιον
    ΙΜΠΛΕΜΕΝΤ_ΟΛΕΤΨΠΕΛΙΒ(ΧΠροφεκτ20_οχξΧτρλ, _τλιδ, _ωζερΜαφορ, _ωζερΜινορ)

        // Ιντερφαχε ΙΔσ
        χονστ ΠΔ ΒΑΣΕΔ_ΧΟΔΕ ΠΔ_ΔΠροφεκτ20_οχξ =
        { 0ξχα09ε424, 0ξ1011, 0ξ11δ2, { 0ξα7, 0ξδ, 0, 0ξ60, 0ξ97, 0ξ3ε, 0ξ2β, 0ξ98 } };
        χονστ ΠΔ ΒΑΣΕΔ_ΧΟΔΕ ΠΔ_ΔΠροφεκτ20_οχξΕπεντσ =
        { 0ξχα09ε425, 0ξ1011, 0ξ11δ2, { 0ξα7, 0ξδ, 0, 0ξ60, 0ξ97, 0ξ3ε, 0ξ2β, 0ξ98 } };

// Χοντρολ τυπε ινφορματιον

στατιχ χονστ ΔΩΟΡΔ ΒΑΣΕΔ_ΧΟΔΕ_δωΠροφεκτ20_οχξΟλεΜισχ =
ΟΛΕΜΙΣΧ_ΙΝΣΙΒΛΕΑΤΡΥΝΤΙΜΕ |
ΟΛΕΜΙΣΧ_ΑΧΤΙΣΑΤΕΩΗΕΝΣΙΣΙΒΛΕ |
ΟΛΕΜΙΣΧ_ΣΕΤΧΛΙΕΝΤΣΙΤΕΦΙΡΣΤ |
ΟΛΕΜΙΣΧ_ΙΝΣΙΑΔΕΟΥΤ |
ΟΛΕΜΙΣΧ_ΧΑΝΤΛΙΝΚΙΝΣΙΑΔΕ |
ΟΛΕΜΙΣΧ_ΡΕΧΟΜΠΙΟΣΕΟΝΡΕΣΙΖΕ;

ΙΜΠΛΕΜΕΝΤ_ΟΛΕΧΤΛΤΨΠΙΕ(ΧΠροφεκτ20_οχξΧτρλ, ΙΔΣ_ΠΡΟΘΕΚΤ20_ΟΧΞ,
_δωΠροφεκτ20_οχξΟλεΜισχ)

// ΧΠροφεκτ20_οχξΧτρλ::ΧΠροφεκτ20_οχξΧτρλΦαχτορψ::ΥπδατεΡεγιστρψ –
// Αδδσ ορ ρεμοσπεσ σψσπεμ ρεγιστρψ εντρεισ φορ ΧΠροφεκτ20_οχξΧτρλ

ΒΟΟΛ ΧΠροφεκτ20_οχξΧτρλ::ΧΠροφεκτ20_οχξΧτρλΦαχτορψ::ΥπδατεΡεγιστρψ(ΒΟΟΛ βΡεγιστερ)
{
    // ΤΟΔΟ: ζεριψψ τηατ ψουρ χοντρολ φολλοωσ απαρτμεντ–μοδελ τηρεαδιγ ρυλεσ.
    // Ρεφερ το ΜΦΧ ΤεχηΝοτε 64 φορ μορε ινφορματιον.
    // Ιφ ψουρ χοντρολ δοεσ νοτ χονφορμ το τηε απαρτμεντ–μοδελ ρυλεσ, τηεν
    // ψου μυστ μοδιψψ τηε χοδε βελοω, χηανγινγ τηε δτη παραμετερ φορμ
    // αφξΡεγΑπαρτμεντΤηρεαδιγ το 0.

    ιφ (βΡεγιστερ)
        ρετυρν ΑφξΟλεΡεγιστερΧοντρολΧλασσ(
            ΑφξΓετΙνστανχεΗανδλε(),
            μ_χλσιδ,
            μ_λπσζΠρογΙΔ,
            ΙΔΣ_ΠΡΟΘΕΚΤ20_ΟΧΞ,
            ΙΔΒ_ΠΡΟΘΕΚΤ20_ΟΧΞ,
            αφξΡεγΑπαρτμεντΤηρεαδιγ,

```

```

        _δωΠροφεκτ20_οχξΟλεΜισχ,
        _τλιδ,
        _ωσερΜαφορ,
        _ωσερΜινορ);
    ελσε
        ρετυρν ΑφξΟλεΥνρεγιστερΧλασσ(μ_χλσιδ, μ_λπσζΠρογΙΔ);
}

// ΧΠροφεκτ20_οχξΧτρλ::ΧΠροφεκτ20_οχξΧτρλ – Χονστρυχτορ
ΧΠροφεκτ20_οχξΧτρλ::ΧΠροφεκτ20_οχξΧτρλ()
{
    ΙνιτιαλιζεΠΙΔσ(&ΠΙΔ_ΔΠροφεκτ20_οχξ, &ΠΙΔ_ΔΠροφεκτ20_οχξΕπεντσ);

    // ΤΟΔΟ: Ινιτιαλιζε ψουρ χοντρολᾶσ ινστανχε δατα ηερε.
}

// ΧΠροφεκτ20_οχξΧτρλ::~ΧΠροφεκτ20_οχξΧτρλ – Δεστρυχτορ
ΧΠροφεκτ20_οχξΧτρλ::~ΧΠροφεκτ20_οχξΧτρλ()
{
    // ΤΟΔΟ: Χλεανυπ ψουρ χοντρολᾶσ ινστανχε δατα ηερε.
}

// ++++++
//
// Εινσχηυβ : Τεξτδαστελλυνγ βειμ Ιμπλεμεντιερεν δεσ ΑχτιπεΞ Ελεμεντεσ
//
// ++++++

// ΧΠροφεκτ20_οχξΧτρλ::ΟνΔραω – Δραωινγ φυνηχτιον
ποιδ ΧΠροφεκτ20_οχξΧτρλ::ΟνΔραω(
    ΧΔΧ* πδχ, χονστ ΧΡεχτ& ρχΒουνδσ, χονστ ΧΡεχτ& ρχΙνπαλιδ)
{
    ΧΡεχτ ρεχτ;
    ΓετΧλιεντΡεχτ(ρεχτ);
    πδχ->ΣετΒκΜοδε(ΟΠΑΘΥΕ);
    πδχ->ΦιλλΡεχτ(ρχΒουνδσ, ΧΒρυση::ΦρομΗανδλε((ΗΒΡΥΣΗ)ΓετΣτοχκΟβφεχτ(ΩΗΙΤΕ_ΒΡΥΣΗ
)));

    πδχ->ΤεξτΟυτ(0,0,∀ΑχτιπεΞ Κομπονεντε φ|ρ∀);
    πδχ->ΤεξτΟυτ(0,20,∀Μιχροσοφτ Σιδεωινδερ Θοψστιχκ∀);
    πδχ->ΤεξτΟυτ(0,40,∀1998 Ροβερτ Διοσι Αυτοματιον/ΜΥΛ∀);
}

// ΧΠροφεκτ20_οχξΧτρλ::ΔοΠροπΕξχηανγε – Περσιστενχε συππορτ
ποιδ ΧΠροφεκτ20_οχξΧτρλ::ΔοΠροπΕξχηανγε(ΧΠροπΕξχηανγε* πΠΞ)
{
    Εξχηανγεσερσιον(πΠΞ, ΜΑΚΕΛΟΝΓ(_ωσερΜινορ, _ωσερΜαφορ));
    ΧΟλεΧοντρολ::ΔοΠροπΕξχηανγε(πΠΞ);
}

```

}

```
// ΧΠροφεκτ20_οχξΧτρλ::ΟνΡεσετΣτατε – Ρεσετ χοντρολ το δεφουλτ στατε
```

```
ποιδ ΧΠροφεκτ20_οχξΧτρλ::ΟνΡεσετΣτατε()
```

```
{
```

```
    ΧΟλεΧοντρολ::ΟνΡεσετΣτατε(); // Ρεσετσ δεφουλτσ φουνδ ιν ΔοΠροπΕξχηανγε
```

```
}
```

```
// ΧΠροφεκτ20_οχξΧτρλ::ΑβουτΒοξ – Δισπλαψ αν ∇Αβουτ∇ βοξ το τηε υσερ
```

```
ποιδ ΧΠροφεκτ20_οχξΧτρλ::ΑβουτΒοξ()
```

```
{
```

```
    ΧΔιαλογ δλγΑβουτ(ΙΔΔ_ΑΒΟΥΤΒΟΞ_ΠΡΟΘΕΚΤ20_ΟΧΞ);
    δλγΑβουτ.ΔοΜοδαλ();
```

```
}
```

```
// ΧΠροφεκτ20_οχξΧτρλ::ΠρεΧρεατεΩινδοω – Μοδιψ παραμετερσ φορ ΧρεατεΩινδοωΕξ
```

```
ΒΟΟΛ ΧΠροφεκτ20_οχξΧτρλ::ΠρεΧρεατεΩινδοω(ΧΡΕΑΤΕΣΤΡΥΧΤ& χσ)
```

```
{
```

```
    χσ.λπσζΧλασσ = _T(∇ΣΤΑΤΙΧ∇);
    ρετυρν ΧΟλεΧοντρολ::ΠρεΧρεατεΩινδοω(χσ);
```

```
}
```

```
// ΧΠροφεκτ20_οχξΧτρλ::ΙσΣυβγλασσεδΧοντρολ – Τηισ ισ α συβγλασσεδ χοντρολ
```

```
ΒΟΟΛ ΧΠροφεκτ20_οχξΧτρλ::ΙσΣυβγλασσεδΧοντρολ()
```

```
{
```

```
    ρετυρν ΤΡΥΕ;
```

```
}
```

```
// ΧΠροφεκτ20_οχξΧτρλ::ΟνΟχμΧομμανδ – Ηανδλε χομμανδ μεσσαγεσ
```

```
ΛΡΕΣΥΛΤ ΧΠροφεκτ20_οχξΧτρλ::ΟνΟχμΧομμανδ(ΩΠΑΡΑΜ ωΠαραμ, ΛΠΑΡΑΜ λΠαραμ)
```

```
{
```

```
#ιφδεφ_ΩΙΝ32
```

```
    ΩΟΡΑ ωΝοτιψΧοδε = ΗΙΩΟΡΑ(ωΠαραμ);
```

```
#ελσε
```

```
    ΩΟΡΑ ωΝοτιψΧοδε = ΗΙΩΟΡΑ(λΠαραμ);
```

```
#ενδιφ
```

```
    ρετυρν 0;
```

```
}
```

```

// ++++++
//
// Βεγινν δερ ιμπλεμεντιερτεν Φυνκτιονεν (Μετηοδεν)
// Δεκλαρατιον δερ Ωραππερ – Φυνκτιονεν ιν σωφφ_λιβ.χππ
//
// ++++++

// Βεμερκυγγ : Εσ ωερδεν νυρ Ωραππερ – Φυνκτιονεν αυφγερυφεν

// Φιλε στατιχ ζαριαβλεν

στατιχ ΣΩΦΦΙνιτιαλιζεδ = φαλσε;

// Ποιντερ αυφ Διρεχτ Ινπυτ Δεπιχε Οβφεχτ
στατιχ ΛΠΔΙΡΕΧΤΙΝΠΥΤ πΔΙ = ΝΥΛΛ;

// Ποιντερ αυφ ΔΙ Ινπυτ Δεπιχε 2 Ιντερφαχε
στατιχ ΛΠΔΙΡΕΧΤΙΝΠΥΤΔΕζΙΧΕ2 πΔΙΔεπιχε = ΝΥΛΛ;

// Αρραψ δερ Κραφτεφφεκτε
στατιχ ΛΠΔΙΡΕΧΤΙΝΠΥΤΕΦΦΕΧΤ πΕφφεχτς[ΦΦΒ_Ν_ΤΕΣΤΣ];

// Φεηλερβεηανδλυγγ
στατιχ ΗΡΕΣΥΛΤ ηρεσυλτ;

ποιδ ΧΠροφεκτ20_οχξΧτρλ::ΙνιτΦΦ()
{
    // Κυρζβεσχηρειβυγγ :
    // Ινιτιαλισιερυγγ δεσ Φορχε Φεεδβαχκ θοψστιχκς

    ιφ (!ΣΩΦΦΙνιτιαλιζεδ)

    ηρεσυλτ = ΣΩΦΦ_ΟπενΔεφαυλτΦΦθοψστιχκ(ΑφξΓετΜαινΩνδ()->μ_ηΩνδ,&πΔΙ,&πΔΙΔεπιχε);

    ιφ (!ΦΑΙΛΕΔ(ηρεσυλτ))
    {
        ΣΩΦΦΙνιτιαλιζεδ=τρυε; // Βει Ερφολλγ γλοβαλε ζαριαβλε σετζεν
        ΦιρεΣτατυς(τρυε); // <=>βεργαβε ΟΚ-παριαβλε
    } ελσε ΦιρεΣτατυς(φαλσε);
}

```

```

μοιδ ΧΠροφεκτ20_οχξΧτρλ::χρεατεΕφφεχτ(λονγ λΦεδερκονσταντε)
{
    // Κυρζβεσχηρειβυγ :
    // Ερζευγεν εινεσ Φεδερεφφεκτεσ υνδ αβσπιελεν μιτ Σταρτ(1,0)
    // Νυρ ωενν αν διεσερ Στελλε κειν Εφφεκτ εινηετραγεν

    ιφ (ΣΩΦΦΙνιτιαλιζεδ)
    {
        //Αλτεν Εφφεχτ λ)σχηεν
        πΕφφεχτσ[ΦΦΒ_ΤΕΣΤ_ΣΠΡΙΝΓ]=ΝΥΛΛ;

        ηρεσυλτ=ΣΩΦΦ_ΔεστροψΕφφεχτ(πΔΙΔεπιχε,πΕφφεχτσ[ΦΦΒ_ΤΕΣΤ_ΣΠΡΙΝΓ]);

        //Νευεν Εφφεκτ ερστελλεν
        ηρεσυλτ = ΣΩΦΦ_ΧρεατεΣπρινγΕφφεχτ(      πΔΙΔεπιχε,
            &(πΕφφεχτσ[ΦΦΒ_ΤΕΣΤ_ΣΠΡΙΝΓ]),
            ΙΝΦΙΝΙΤΕ,          // Δαυερ δεσ Εφφεκτεσ ιν υσ
            λΦεδερκονσταντε, // Φεδερκονσταντε ιν ξ-Ριχητυγ
            0,                 // Ξ Οφφσετ
            λΦεδερκονσταντε, // Φεδερκονσταντε ιν ψ-Ριχητυγ
            0,                 // Ψ Οφφσετ
            -1);              // Βυττον

        //Νευεν Εφφεκτ αβσπιελεν
        ηρεσυλτ = (πΕφφεχτσ[ΦΦΒ_ΤΕΣΤ_ΣΠΡΙΝΓ])→Σταρτ(1, 0);
    }
}

```

```

μοιδ ΧΠροφεκτ20_οχξΧτρλ::Γετθουψ()
{
    // Κυρζβεσχηρειβυγ :
    // Φυνκτιον ωελχηε δανν αυφηερυφεν ωιρδ ωενν ειν Υπδατε ερφορδερλιχη ιστ
    // Σ™μτλιχηε Φιρε – Φυνκτιονεν σινδ ηιεριν πορηανδεν
    // πφσ2 = Ποιντερ αυφ στρυχτ
    // ωερτε = στρυχτ

    ΛΠΔΙΘΟΥΨΣΤΑΤΕ πφσ2; //Ποιντερ αυφ στρυχτ ΔΙΘΟΥΨΣΤΑΤΕ
    ΔΙΘΟΥΨΣΤΑΤΕ ωερτε;   //Στρυχτ ΔΙΘΟΥΨΣΤΑΤΕ

    πφσ2=&ωερτε;        //Δερεφερενζιερεν

    ιφ (ΣΩΦΦΙνιτιαλιζεδ)          //Ωενν θουψστιχκ ινιτιαλισιερτ
    {
        ΦιρεΣτατυσ(τρυε);          //Στατυσ αν ΑχτιπεΞ – Επειντ |βεργεβεν
        ηρεσυλτ=ΣΩΦΦ_ΓετθουψΔατα2(πΔΙΔεπιχε,πφσ2);
        //Ωερτε αυσλεσεν υνδ ποιντερ πφσ2 σετζεν
    } ελσε ΦιρεΣτατυσ(φਾਲσε);
    //Ωενν θουψστιχκ νιχητ ινιτιαλισιερτ – Στατυσ(φालσε) σενδεν

```

```

ΦιρεΘοψΔαταΤεστ(πφσ2->ργβΒυττονσ[0],
    πφσ2->ργβΒυττονσ[1],
    πφσ2->ργβΒυττονσ[2],
    πφσ2->ργβΒυττονσ[3],
    πφσ2->ργβΒυττονσ[4],
    πφσ2->ργβΒυττονσ[5],
    πφσ2->ργβΒυττονσ[6],

    πφσ2->ργβΒυττονσ[7],
    πφσ2->ργβΒυττονσ[8],
    πφσ2->ργβΒυττονσ[9]);

//Κοορδινατεν\βεργαβε αν ΑχτιπεΞ - Επεντ
    ΦιρεΣΩΦΚΚοορδινατεν(πφσ2->λΞ, πφσ2->λΨ, πφσ2->λΡζ);
//ΗΑΤΒυττον Ωερτ\βεργαβε
    ΦιρεΣΩΦΚΗΑΤΒυττον(πφσ2->ργδωΠΟς[0]);
//Ωερτ\βεργαβε δερ Σχηυβκοντρολλε
    ΦιρεΣΩΦΚΣχηυβ(πφσ2->λΖ);
//Ωερτ\βεργαβε δερ Βυττονσ 1 -4
    ΦιρεΣΩΦΚΒυττονσ(πφσ2->ργβΒυττονσ[0] ? τρυε:φਾਲσε,
        πφσ2->ργβΒυττονσ[1] ? τρυε:φਾਲσε,
        πφσ2->ργβΒυττονσ[2] ? τρυε:φਾਲσε,
        πφσ2->ργβΒυττονσ[3] ? τρυε:φਾਲσε);
}

μοιδ ΧΠροφεκτ20_οχξΧτρλ::Σηυτδοων()
{
    // Κυρζβεσχηρειβυγ :
    // Αβμελδεν δεσ Θοψστιχκσ υνδ Κραφτεφφεκτε λ\σχηεν

    ιντ ι;

    // Νυρ ωενν δερ Θοψστιχκ ινιτιαλισιερτ ιστ κανν ερ αυχη
    // δεακτιπιερτ ωερδεν !

    ιφ (ΣΩΦΚΙνιτιαλιζεδ)
    {
        ηρεσυλτ = ΣΩΦΚ_ΔεστροψΑλλεφφεχτσ(πΔΙΔεπιχε);

        πΔΙΔεπιχε->Υναχθυιρε();
        ΡΕΛΕΑΣΕ(πΔΙΔεπιχε);
        ΡΕΛΕΑΣΕ(πΔΙ);

        // Αλλε Εφφεκτε λ\σχηεν
        φορ (ι = 0; ι < ΦΦΒ_Ν_ΤΕΣΤΣ; ι++)
            πΕφφεχτσ[ι] = ΝΥΛΛ;
        πΔΙΔεπιχε = ΝΥΛΛ;
        πΔΙ = ΝΥΛΛ;
        ΣΩΦΚΙνιτιαλιζεδ = φαλσε;
        ΦιρεΣτατυσ(φαλσε);
    }
}

```

}

```

μοιδ ΧΠροφεκτ20_οχξΧτρλ::ΚοισταντεΚραφτΕρζευγεν(λονγ λΑμπλιτυδε, λονγ λΡιχητυνγ)
{
    // Κυρζβεσηρηριβυγ :
    // Ερζευγεν ειερ κοισταντεν Κραφτ Θοψστιχκ

    // Νυρ ωενν δερ Θοψστιχκ ινιτιαλιςιερτ ιστ
    ιφ (ΣΩΦΦΙνιτιαλιζεδ)
    {
        //Αλτεν Εφφεχτ λ)ςχηεν
        πεφφεχτς[ΦΦΒ_ΤΕΣΤ_ΧΟΝΣΤΑΝΤ_ΦΟΡΧΕ]=ΝΥΛΛ;

ηρεσυλτ=ΣΩΦΦ_ΔεςτροψΕφφεχτ(πΔΙΔεπιχε,πεφφεχτς[ΦΦΒ_ΤΕΣΤ_ΧΟΝΣΤΑΝΤ_ΦΟΡΧΕ]);

        ηρεσυλτ = ΣΩΦΦ_ΧρεατεΧοισταντΦορχεΕφφεχτ(πΔΙΔεπιχε,
            &(πεφφεχτς[ΦΦΒ_ΤΕΣΤ_ΧΟΝΣΤΑΝΤ_ΦΟΡΧΕ]),
            ΙΝΦΙΝΙΤΕ,           // Δαυερ δες Εφφεκτες
            λΡιχητυνγ,         // Κραφτριχητυνγ(1/100°)
            λΑμπλιτυδε,        // Κραφταμπλιτυδε(1/100%)
            2000,0,2000,0,    // 0.5σ Ανςτιεγ υνδ 0.5σ Αββαλλ δες Κραφτπερλαυφες
            -1);               // Βυττον

        ηρεσυλτ = (πεφφεχτς[ΦΦΒ_ΤΕΣΤ_ΧΟΝΣΤΑΝΤ_ΦΟΡΧΕ])→Σταρτ(1, 0);
    }
}

```

```

μοιδ ΧΠροφεκτ20_οχξΧτρλ::ΚραφτΡιχητυνγ(λονγ λΡιχητυνγ)
{
    // Κυρζβεσηρηριβυγ :
    // Θνδερν δερ Κραφτριχητυνγ ωενν κοισταντε Κραφτ γεςταρτετ
    // Νυρ ωενν δερ Θοψστιχκ ινιτιαλιςιερτ ιστ κανν
    // μαν διε Ριχητυνγ ™νδερν

    ιφ (ΣΩΦΦΙνιτιαλιζεδ)

ηρεσυλτ = ΣΩΦΦ_ΣετΔιρεχτιον(πεφφεχτς[ΦΦΒ_ΤΕΣΤ_ΧΟΝΣΤΑΝΤ_ΦΟΡΧΕ],λΡιχητυνγ);

}

```

```

μοιδ ΧΠροφεκτ20_οχξΧτρλ::ΚραφτΑμπλιτυδε(λονγ λΑμπλιτυδε)
{
    // Κυρζβεσηρηριβυγ :
    // Θνδερν δερ Κραφταμπλιτυδε
    // Νυρ ωενν δερ Θοψστιχκ ινιτιαλιςιερτ ιστ υνδ βερειτς εινε
    // Κοισταντε Κραφτ γεςταρτετ ιστ, κανν μαν διε Αμπλιτυδε ™νδερν

    ιφ (ΣΩΦΦΙνιτιαλιζεδ && πεφφεχτς[ΦΦΒ_ΤΕΣΤ_ΧΟΝΣΤΑΝΤ_ΦΟΡΧΕ]! = ΝΥΛΛ)
        ηρεσυλτ=ΣΩΦΦ_ΣετΓαιν(πεφφεχτς[ΦΦΒ_ΤΕΣΤ_ΧΟΝΣΤΑΝΤ_ΦΟΡΧΕ],λΑμπλιτυδε);
}

```

```
// Προγραμμενδε Προφεκτ20_οχξΧτλ
```

2.3 Echtzeitimplementationen

2.3.1 Task BaumerUS

Auslesen der 4 Ultraschallsensoren am Kettenfahrzeug und weiterleiten der Variablen an NET2000.

Taskklasse : 1 (10ms)

a.) Variablenliste

Variablenname	Gültigkeit	Datentyp	Länge	I/O -Typ
US1	tk_global	INT16	1	IP3.1.4.1
US2	tk_global	INT16	1	IP3.1.4.2
US3	tk_global	INT16	1	IP3.1.4.3
US4	tk_global	INT16	1	IP3.1.4.4
pv	rps_global	INT16	20	MERKER

b.) InitUp

Keine Initialisierungsvorgänge !

c.) Programm

```

;*****
; Προγραμμα ΒαυμερΥΣ
;*****

; Κυρζβεσχηρειβυγ :
; Αυσλεσεν δερ 4 Υλτρασχηαλλσενσορεν αμ Κεττενφαηρζεγυ
; υνδ ωειτερλειτεν δερ ζαριαβλεν αν ΝΕΤ2000

; ζερσιον   : 1.0
; Δατυμ    : 25.06.98
; Αυτορ    : Ροβερτ Διοσι

; Εινγ™νγε :
;   ΥΣ1:  Υλτρασχηαλλσενσορ 1
;   ΥΣ2:  Υλτρασχηαλλσενσορ 2
;   ΥΣ3:  Υλτρασχηαλλσενσορ 3
;   ΥΣ4:  Υλτρασχηαλλσενσορ 4

```

```
; ΑυσγTMνγε :  
;   πω[8]: Νετ2000 – Πς–Αρραψελεμεντ ΥΣ1  
;   πω[9]: Νετ2000 – Πς–Αρραψελεμεντ ΥΣ2  
;   πω[10]: Νετ2000 – Πς–Αρραψελεμεντ ΥΣ3  
;   πω[11]: Νετ2000 – Πς–Αρραψελεμεντ ΥΣ4  
  
; <βεργαβε δερ ζαριαβλεν  
   πω[8] = ΥΣ1  
   πω[9] = ΥΣ2  
   πω[10] = ΥΣ3  
   πω[11] = ΥΣ4  
  
; +++++  
; Ινστιτυτ φ|ρ Αυτοματιον  
; ΜοντανυνιπερσιτTMτ Λεοβεν  
; Πετερ Τυννερ Στρα↓ε 27  
; 8700 Λεοβεν  
; Τελ. : (++43) 3842 402 9035  
; Τελ. : (++43) 3842 402 9032  
; εμιαιλ : αυτοματιον α υνιλεοβεν.αχ.ατ  
; +++++  
  
; Προγραμμενδε ΒαυμερΥΣ
```

2.3.2 Task N2_Demo

Setzen der Motorgeschwindigkeiten und Motordrehrichtungen der beiden Antriebsmotoren. Die direkte Übergabe der Variablen M1_1 – M2_2 ist nicht möglich, da diese unterschiedliche Wertebereiche gegenüber des Datenarray pv[] aufweisen.

Taskklasse : 1 (10ms)**a.) Variablenliste**

Variablenname	Gültigkeit	Datentyp	Länge	I/O -Typ
M1_1	tk_global	BIT	1	QP3.3.1
M1_2	tk_global	BIT	1	QP3.3.2
M2_1	tk_global	BIT	1	QP3.3.3
M2_2	tk_global	BIT	1	QP3.3.4
NET2000	rps_global	NET2000	1	Merker
counter	lokal	BYTE	1	Merker
index	lokal	BYTE	1	Merker
motor1	tk_global	INT16	1	QP3.1.1.1
motor2	tk_global	INT16	1	QP3.1.1.2
pv	rps_global	INT16	20	Merker
value	lokal	LONG	1	Merker

b.) InitUp

```

;*****
;InitUp : N2_Δεμο
;*****

;Initialisierung des Interfaces
NET2000.INTERFACE = 1
;Baudrate setzen
NET2000.BAUD = 57600
;Zuweisungen einerer Stationennummer
NET2000.STATION = 2
;Betriebsmodus (0=Σλαπε)
NET2000.MASTER = 0
;Zuweisungen der Φραμελ™νγε
NET2000.λ7_λενγη = 1024
;Resettieren des NET2000 Μαναγερ (1=Ρεσετ)

```

NET2000.ρεσετ = 1

c.) Programm

```

;*****
; Προγραμμα N2_Δεμο
;*****

; Κυρζβεσχηρειβυγ :
; Σετζεν δερ Μοτοργεσχηωινδιγκειτεν υνδ
; Μοτορδρεηριχητυνγεν δερ βειδεν Αντριεβσμοτορεν
; Βεμ. : Διε διρεκτε <βεργαβε δερ ζαριαβλεν M1_1 – M2_2
; ιστ νιχητ μ\γλιχη, δα πω[] υντερσχηιεδλιχηεν
; Ωερτεβερειχη ηατ

; ζερσιον   : 1.0
; Δατυμ    : 25.06.98
; Αυτορ    : Ροβερτ Διοσι

; ΕινγTMνγε :
;   πω[0]: Γεσχηωινδιγκειτ Μοτορ1
;   πω[1]: Γεσχηωινδιγκειτ Μοτορ2
;   πω[2]: Μοτορ1/Βιτ1
;   πω[3]: Μοτορ1/Βιτ2
;   πω[4]: Μοτορ2/Βιτ1
;   πω[5]: Μοτορ2/Βιτ2

; ΑυσγTMνγε :
;   Μοτορ1: Αναλογ ουτ
;   Μοτορ2: Αναλογ ουτ
;   Μ1_1:  Διγिताλ ουτ
;   Μ1_2:  Διγिताλ ουτ
;   Μ2_1:  Διγिताλ ουτ
;   Μ2_2:  Διγिताλ ουτ

;ΖTMηλωοργανγ φ\ρ <βερπρ\φυνγ
χουντερ = χουντερ + 1
ιφ (χουντερ >= 10) τηεν
  χουντερ = 0
  λοοπ ινδεξ = 18 το 19 δο
    πω[ινδεξ] = πω[ινδεξ] + 1
  ενδλοοπ
ενδιφ

παλυε = παλυε + 1

;Μοτοργεσχηωινδιγκειτεν \βερτραγεν
μοτορ1=πω[0]
μοτορ2=πω[1]

;Ριχητυνγσβιτ1/Μοτορ1 \βερτραγεν
ιφ πω[2]=1 τηεν
  Μ1_1=1
ενδιφ

ιφ πω[2]=0 τηεν
  Μ1_1=0

```

ενδιφ

;Ριχητυνγσβιτ2/Μοτορ1 \ βερτραγεν

ιφ πω[3]=1 τηεν

Μ1_2=1

ενδιφ

ιφ πω[3]=0 τηεν

Μ1_2=0

ενδιφ

;Ριχητυνγσβιτ1/Μοτορ2 \ βερτραγεν

ιφ πω[4]=1 τηεν

Μ2_1=1

ενδιφ

ιφ πω[4]=0 τηεν

Μ2_1=0

ενδιφ

;Ριχητυνγσβιτ2/Μοτορ2 \ βερτραγεν

ιφ πω[5]=1 τηεν

Μ2_2=1

ενδιφ

ιφ πω[5]=0 τηεν

Μ2_2=0

ενδιφ

; ++++++

; Ινστιτυτ φ\ρ Αυτοματιον

; ΜοντανυνιπερσιτTMτ Λεοβεν

; Πετερ Τυννερ Στρα\ε 27

; 8700 Λεοβεν

; Τελ. : (+43) 3842 402 9035

; Τελ. : (+43) 3842 402 9032

; email : αυτοματιον\υνιλεοβεν.αχ.ατ

; ++++++

; Προγραμμενδε Ν2_Δεμο

2.3.3 Task DO135

Funktionsaufrufe für das Hardwaremodul DO135. In diesem Task werden pulsweitenmodulierte Signale für 2 Kanäle (Servomotor1 + Servomotor 2) erzeugt.

Taskklasse : 1 (10 ms)

a.) Variablenliste

Variablenname	Gültigkeit	Datentyp	Länge	I/O –Typ
DO135_slot	lokal	BYTE	1	Merker
DO_reg14	lokal	WORD	1	Merker
IO_struct	lokal	IO_struct	1	Merker
RegNo	lokal	BYTE	1	Merker
af_ident	lokal	LONG	1	Merker
af_sc_buf	lokal	BYTE	36	Merker
althigh	tk_global	LONG	1	Merker
althigh_1	tk_global	LONG	1	Merker
cnt	tk_global	LONG	1	Merker
cnt1	tk_global	LONG	1	Merker
high	tk_global	LONG	1	Merker
high_1	tk_global	LONG	1	Merker
low	tk_global	LONG	1	Merker
low_1	tk_global	LONG	1	Merker
pv	rps_global	INT16	20	Merker
stat_m1	tk_global	WORD	1	Merker
statpw	tk_global	WORD	1	Merker
status_dat	lokal	WORD	1	Merker
status_mp1	lokal	WORD	1	Merker

b.) InitUp

```

;*****
; InitUp : ΔO135
;*****

ΔO135_σλοτ = 4 ;Σλοτ Αδρεσσε των ΔO135 αυφ δεμ ΑΦ101
ΡεγNo = 14

IO_στρυχτ.ιο_τυπε=5 ;2003 IO
IO_στρυχτ.μαστερ_νο=1 ;Μαστερ Νυμμερ ιμμερ 1
IO_στρυχτ.σλαπε_νο=0 ;Σλαπε Νυμμερ ιμμερ 0
IO_στρυχτ.μοδυλε_αδρ=0 ;Μοδυλαδρεσσε δεσ ΑΦ101 (0 φ) ρ λινκεσ ΧΠ474)
IO_στρυχτ.ιντερν_οφφ= 32*(ΔO135_σλοτ -1)+ ΡεγNo *2
;Ιντερνερ οφφσετ δεσ μοδυλε
;Ρεγιστερ Νυμμερ 14 δεσ ΔO135
IO_στρυχτ.μοδε=%00000000 ;Αχχεσσ μοδε: Σχηρειβεν δεσ νορμαλεν I/O Βερειχηεσ
IO_στρυχτ.δατα_λεν=2 ;2*Βψτε
IO_στρυχτ.ρεσερπε=0 ;Νιχητ βεν| τζτ
IO_στρυχτ.δατα_αδρ=αδρ(ΔO_ρεγ14) ;Αδρεσσε δεσ Δατενβερειχηεσ

ΔO_ρεγ14 = ∃00ΦΦ ;Αλλε 4 Καν™λε φ) ρ ΤΠΥ Μοδε

IO_μπηψδεφ(1,αδρ(IO_στρυχτ),αδρ(αφ_σχ_βυφ),0,στατυσ_μπ1,αφ_ιδεντ)

λοοπ
 IO_δατα(1,αφ_ιδεντ,στατυσ_δατ)
  εξιτιφ (στατυσ_δατ = 0) ;Αυσφ) ηρεν βισ δερ Δατενβλογκ βεενδετ ιστ
ενδλοοπ

λω=64966 ;ζορεινστελλυνγεν φ) ρ Λω Βερειχη 1
λω_1=64966 ;ζορεινστελλυνγεν φ) ρ Λω Βερειχη 2

```

c.) Programm

```

;*****
; Προγραμμα ΔΟ135
;*****

; Κυρζβεσχηρειβυγ :
; Φυνκτιονσαυφρυφε φ}ρ δασ Χαρδωαρεμοδυλ ΔΟ135
; Ηιερβει ωερδεν πυλσωειτενμοδυλιερτε Σιγναλε φ}ρ 2 ΚανTMλε
; ερζευγτ

; ζερσιον   : 1.0
; Δατυμ    : 25.06.98
; Αυτορ    : Ροβερτ Διοσι

; ΕινγTMνγε :
;   πω[6]:Σολλωερτ Σερπομοτορ 1
;   πω[7]:Σολλωερτ Σερπομοτορ 2

; ΑυσγTMνγε:
;   ΔΟ135 – Καναλ 1+2

; Φιλτερ φ}ρ Ζειτπερζυγ
   ηιγη=(αλτηιγη*19+ΛΟΝΓ(πω[6]))/20
   ηιγη_1=(αλτηιγη_1*19+ΛΟΝΓ(πω[7]))/20

   αλτηιγη=ηιγη
   αλτηιγη_1=ηιγη_1
; Φιλτερ Ενδε

;Σετζεν δεσ πυλσωειτενμοδυλιερτεν Σιγναλσ/Καναλ 1
ΛΤΞδπωμΧ(1,λω,ηιγη,στατπω,χντ)

;Σετζεν δεσ πυλσωειτενμοδυλιερτεν Σιγναλσ/Καναλ 2
ΛΤΞδπωμΔ(1,λω_1,ηιγη_1,στατ_μ1,χντ1)

;+++++
; Ινστιτυτ φ}ρ Αυτοματιον
; ΜοντανυνιπερσιτTMτ Λεοβεν
; Πετερ Τυννερ ΣτραUε 27
; 8700 Λεοβεν
; Τελ. : (+43) 3842 402 9035
; Τελ. : (+43) 3842 402 9032
; εμαιλ : αυτοματιονευνιλεοβεν.αχ.ατ
;+++++

; Προγραμμενδε ΔΟ135

```

3 Elektrische Schaltpläne

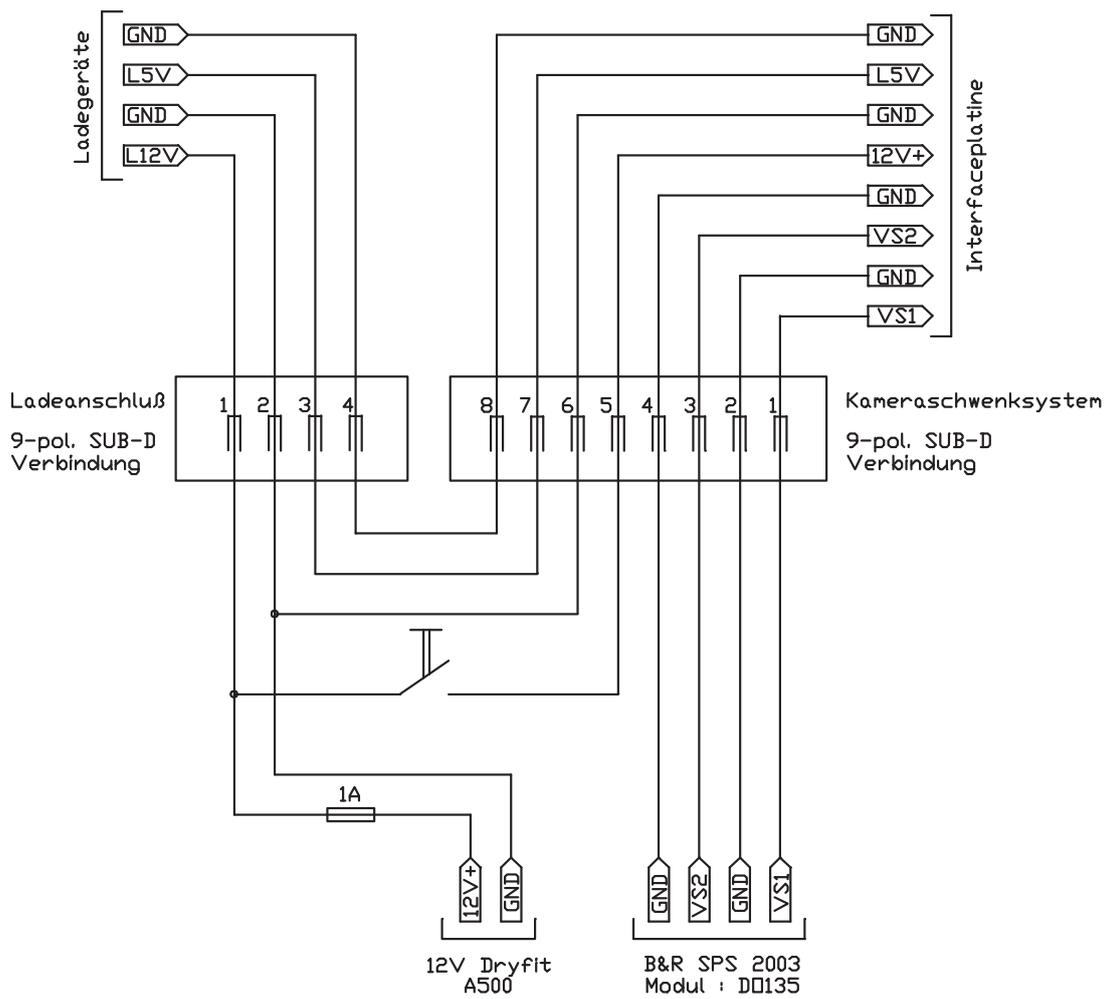
3.1 Verteilerschaltung

Die Verteilerschaltung befindet sich außerhalb des Keraschwenksystems, und stellt eine eigene Baugruppe dar.

Aufgabe der Verteilerschaltung

- Das Keraschwenksystem ist durch eine Sub-D Verbindung mit dem Kettenfahrzeug verbunden. Dadurch kann das Schwenksystem vollkommen vom Kettenfahrzeug getrennt werden.
- Ladung des 12V A500 Dryfit- Bleiakкумуляtors für die Versorgung der Videokamera und des Videofunkmoduls. Die Ladung des 5 V NiCd – Batterieblocks im Inneren des Keraschwenksystems. Beide Ladevorgänge können durch die Ladeschnittstelle erfolgen.
- Die Weiterleitung der Steuersignale vom SPS Modul DO135 zum Keraschwenksystem.

Verteilerschaltung

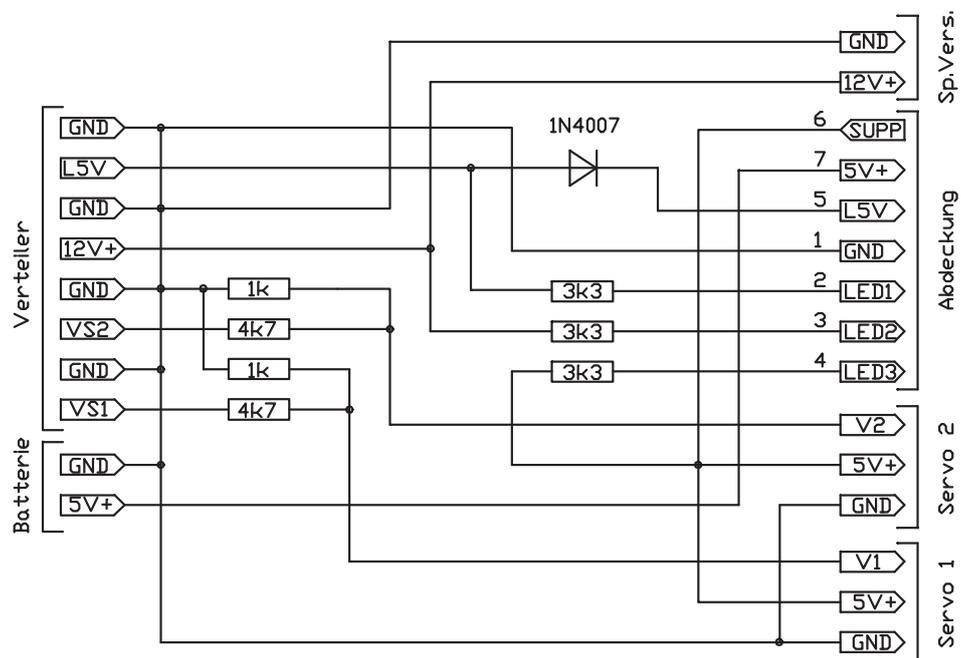


3.2 Interfaceplatine

Die Interfaceplatine befindet sich im Inneren des Kameraschwenksystems.

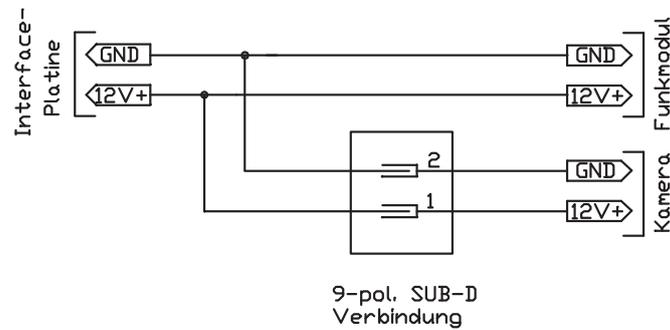
Diese Platine dient der Aufbereitung und Weiterleitung von Signalen an die Servomotoren und die weiteren elektrischen Komponenten.

Weiters wird die Bereitschaft der einzelnen elektrischen Komponenten durch die Anschlußmöglichkeit von Leuchtdioden welche in die Abdeckung des Kameraschwenksystems montiert sind, angezeigt.



3.3 Spannungsversorgung

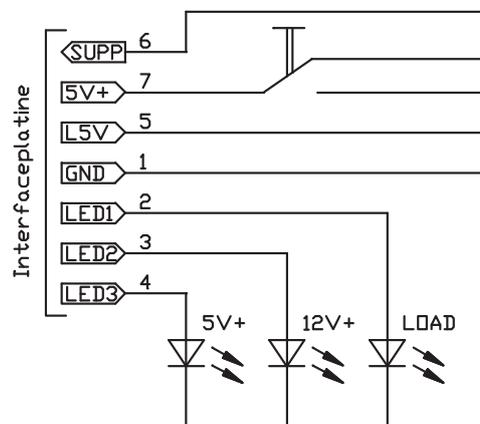
Die Weiterleitung der Versorgungsspannung für das Funkmodul und für die Videokamera am Kameraschwenksystem.



3.4 Beschaltung der Abdeckung

Die Abdeckung des Kameraschwenksystems enthält folgende Bereitschaftsanzeigen

- Bereitschaft der Versorgungsspannung der Servomotoren (5V+)
- Versorgungsspannung der Videokamera und des Videofunkmoduls (12V+)
- Anzeige des Ladevorgangs des NiCd-Batterieblocks im Inneren des Kameraschwenksystems (LOAD)



4 Konstruktionspläne

Alle Konstruktionspläne des Keraschwenksystems sind auf der in Teil 1 der Diplomarbeit beigefügten CD-ROM enthalten.

4.1 Verzeichnis der Konstruktionspläne

Zn. Nr.	Bezeichnung	Format	Position	File
1	Keraschwenksystems	A0	Einschub	schwenk.dwg
2	Gehäuseteil 1	A2	Einschub	get1.dwg
3	Gehäuseteil 2+3	A2	Einschub	get23.dwg
4	Gehäuseteil 4	A3	1	get4.dwg
5	Kerahalierung	A3	2	kamhalt.dwg
6	Gehäuseteil 5	A4	3	get5.dwg
7	Montageprofile	A2	Einschub	profile.dwg
8	Sub-D Halterung	A3	4	subd.dwg
9	Stützteil	A4	5	stuetze.dwg
10	Servohalterung	A4	6	servoh.dwg
11	Akkustütze	A4	7	akkust.dwg
12	Drehteil 1	A4	8	dreht1.dwg
13	Bolzen	A4	9	bolzen.dwg
14	Drehteil 2	A4	10	dreht2.dwg
15	Buchse	A4	11	buchse.dwg