



(19)
Bundesrepublik Deutschland
Deutsches Patent- und Markenamt

(10) **DE 44 11 426 B4** 2004.07.01

(12)

Patentschrift

(21) Aktenzeichen: **P 44 11 426.5**
(22) Anmeldetag: **31.03.1994**
(43) Offenlegungstag: **13.10.1994**
(45) Veröffentlichungstag
der Patenterteilung: **01.07.2004**

(51) Int Cl.⁷: **G05B 19/418**
G06F 9/46
// G06F 15/46

Innerhalb von 3 Monaten nach Veröffentlichung der Erteilung kann Einspruch erhoben werden.

(30) Unionspriorität:
5-79663 06.04.1993 JP

(71) Patentinhaber:
Honda Giken Kogyo K.K., Tokio/Tokyo, JP

(74) Vertreter:
Klunker, Schmitt-Nilson, Hirsch, 80797 München

(72) Erfinder:
Kobayashi, Minoru, Sayama, Saitama, JP;
Nagasawa, Toshiaki, Sayama, Saitama, JP;
Suzuki, Makoto, Sayama, Saitama, JP; Kobayashi,
Shigeo, Sayama, Saitama, JP

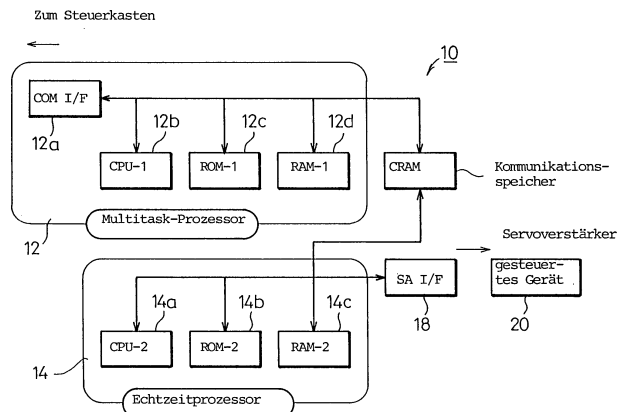
(56) Für die Beurteilung der Patentfähigkeit in Betracht
gezogene Druckschriften:
US 52 29 931 A
US 46 84 862 A
EP 05 90 175 A1
JP 63-0 86 036 A
JP 01-16 540 A
KUSSL, V.: "Programmieren von

Prozeßrechnern",
VDI-Verlag Düsseldorf, S. 72,73;
KALTENBACH/REETZ/WOERRLEIN: "Das große
Computer
Lexikon", Verlag Markt & Technik, 2. Aufl.,
1990, S. 100,204,219,266 u. 267;
SAUERESSIG,F.: "Die moderne CNC-Steuerung",
In: In-
dustrie-Anzeiger 34/1989, S.34-36;
FERRONI,B. PELLONI,N.: "OSI-Architekturen in
CNC-
Steuerungen" In: Technische Rundschau 26/91,
S.62-
68;
SCHELLER,J. SOMMER,E.: "Hierarchisches
Steuerungs-
Konzept für flexible Montagezellen" In: Automatisie-
rungstechnische Praxis atp 31 (1989)4,
S.166-17
3;

(54) Bezeichnung: **Multitask-Steuersystem**

(57) Hauptanspruch: Multitask-Steuersystem zum Steuern des Betriebs mehrerer gesteuerter Geräte (20; 34a, b; 50a, b), insbesondere Schweißroboter, Montageroboter und dergleichen, die über eine gemeinsame Kommunikationsleitung verbunden sind, umfassend:
einen Multitask-Prozessor (12) zur parallelen Steuerung mehrerer Prozesse der jeweiligen gesteuerten Geräte auf Timesharing-Basis mit mehreren vorbestimmten Anwendungstasks (42a, b), die zugehörigen logischen Achsen jeweils eines der mehreren gesteuerten Geräte entsprechen und die von jedem der gesteuerten Geräte durchzuführen sind,
einen Echtzeitprozessor (14) mit Steuerprozessoren (14a1, 14b1; 14a2, 14b2) zum Durchführen einer Betriebs- und einer Positionierungssteuerung auf Echtzeitbasis für jede der logischen Achsen, und
einen Kommunikationsspeicher (16) zum Senden und Empfangen vorbestimmter Befehle und Daten, die für die Betriebssteuerung und die Positionierungssteuerung zwischen dem Multitask-Prozessor (12) und dem Echtzeitprozessor (14) benötigt werden,
wobei der Multitask-Prozessor und der Echtzeitprozessor

als Teilkomponenten einer einzigen Multitasksteuervorrichtung (10) ausgebildet sind, welche baulich von den zu steuernden Geräten getrennt ist, und in der sowohl die...



Beschreibung

[0001] Die Erfindung betrifft ein Multitasking-Steuersystem zum Steuern des Betriebs mehrerer gesteuerter Geräte, insbesondere Schweißroboter, Montageroboter und dergleichen, die über eine gemeinsame Kommunikationsleitung verbunden sind, und die jeweils in der Lage sind, mehrere, in ihnen enthaltene logische Achsen zu positionieren, um Arbeitsabläufe durchzuführen, beispielsweise das Schweißen oder das Zusammenbauen von Werkstücken, wobei eine parallele Steuerung der jeweiligen Geräte durch einen Multitask-Prozessor vorgenommen wird und die Steuerung der Arbeitsabläufe wie zum Beispiel des Schweißens, die naturgemäß eine Echtzeitsteuerung benötigen, von einem Echtzeit-Prozessor vorgenommen wird.

[0002] Verschiedene automatisch arbeitende Anlagen und Maschinen enthalten beispielsweise Schweißroboter, Montageroboter und dergleichen, um in Fertigungsbetrieben, beispielsweise in Kraftfahrzeug-Montagestraßen, verschiedene Arbeiten auszuführen. Im allgemeinen besitzen derartige automatische Anlagen die Fähigkeit, mehrere logische Achsen einer Schweißpistole oder einer zum Schweißen oder zur Montage dienenden Roboterhand zu steuern, wobei die Positionierungs-Steuerung grundsätzlich durch ein rechnergestütztes Servomotor-system erfolgt.

[0003] Entlang eines Montagebandes einer typischen automatischen Fertigungsstraße befinden sich in der Regel mehrere automatische Maschinen der oben beschriebenen Art. Die einzelnen automatischen Maschinen werden entweder individuell oder in Kombination durch eine mit Hilfe eines Mikroprozessors arbeitende Ablaufsteuerung oder mit Hilfe einer rechnergestützten Steuerung gesteuert, um Positionierungs- oder andere Aufgaben durchzuführen.

Stand der Technik

[0004] In den oben beschriebenen automatischen Maschinen eingesetzte Multitask-Steuersysteme sind zum Beispiel in den japanischen offengelegten Patentveröffentlichungen 63-86036 und 1-169540 beschrieben. Der Begriff "Multitask-Steuerung" bezieht sich auf die gleichzeitige Ausführung einer Mehrzahl von Tasks oder Jobs, um auf Time-Sharing-Basis jedesmal dann eine Umschaltung zwischen einzelnen Tasks vorzunehmen, wenn eine Unterbrechung (Interrupt) eingegeben wird. "Echtzeit-Steuerung" bedeutet einen Prozeß, beidem bis zu einer als nächstes folgenden Unterbrechung sämtliche Aktivitäten einer Folge notwendiger Prozeduren einschließlich Eingabe, Berechnung und Ausgabe abgeschlossen werden, wobei diese Prozesse bei jeder Unterbrechungs-Eingabe wiederholt werden.

[0005] Es ist üblich, den in einem Gerät der oben angegebenen Art enthaltenen Achsen (Servomoto-

ren) Targetpositionen oder Zielpositionen zuzuordnen, wobei die gleichzeitige Ankunft der Achsen auf einer Verarbeitung in dem Servomotor-Steuersystem beruht. Wenn also mehrere derartige Geräte vorhanden sind, muß die Verarbeitung für die gleichzeitige Ankunft entsprechend der Anzahl von Geräten wiederholt werden, was zu einer Zunahme des Verarbeitungsaufwands führt. Da eine Änderung der Achsenanordnung durch Parameter erfolgt, ist die Servosteuerungsverarbeitung in hohem Maße komplex, und dementsprechend ist auch das verarbeitende Gerät komplex.

[0006] Wenn mehrere Geräte gesteuert werden müssen, ist der Einsatz einer Multitask-Verarbeitung zum gleichzeitigen Steuern der jeweiligen Geräte zu bevorzugen. Allerdings eignet sich die Multitask-Verarbeitung nicht zur Steuerung von Jobs, die von Natur aus echtzeitorientiert sind. Wenn die Steuerung einer Mehrzahl von Geräten als einzelne Task erfolgt, wird die erforderliche Erstellung eines Rechnerprogramms derart kompliziert, daß sowohl die Zuverlässigkeit als auch die Leistungsfähigkeit des Programms leidet. Außerdem ist ein erheblicher Personalaufwand zum Entwickeln und Einrichten der Programme erforderlich.

[0007] Aus der EP 590 175 A1 (das ist eine europäische Patentanmeldung mit Deutschland als benanntem Vertragsstaat, jedoch nach dem Zeitrang der vorliegenden Anmeldung veröffentlicht) läßt sich ein Prozeßsteuerungssystem entnehmen, bei dem zwei voneinander baulich getrennte Prozessoren vorgesehen sind, von denen der eine Prozessor für die Echtzeitverarbeitung und der andere Prozessor für nicht-zeitgebundene Aufgaben (Tasks) eingesetzt wird. Bei diesem System stehen die beiden als getrennte bauliche Einheiten ausgebildeten Prozessoren über einen internen Bus und einen Speicher in Verbindung, so dass sie Daten austauschen können. Außerdem sind zwei getrennte Teilbusse vorhanden, über die der eine oder der andere Prozessor mit zu steuernden Geräten in Verbindung treten kann. Ein Vorteil eines solchen Systems mit getrennten Prozessoren für Echtzeitaufgaben einerseits und Multitask-Verarbeitung andererseits besteht darin, dass die Programmierung vergleichsweise einfach ist. Denkbar ist außerdem, ohne erheblichen zusätzlichen Programmieraufwand weitere zu steuernde Geräte anzuschließen.

[0008] Aus Kussl, V.: "Programmieren von Prozeßrechnern", VDI-Verlag Düsseldorf, 1975, Seiten 72 und 73, läßt sich entnehmen, dass für den Echtzeit-Betrieb das Multitasking notwendig ist. Dieser Zusammenhang läßt sich so verstehen, dass speziell bei Prozeßrechnern eine Echtzeitverarbeitung einerseits unerlässlich ist, andererseits diese nur durch Multitasking erreicht werden kann.

[0009] Aus Kaltenbach/Reetz/Woerrlein: "Das große Computer-Lexikon", Verlag Markt & Technik, 2. Auflage, 1990, Seiten 100, 204, 219, 266 und 267 lassen sich Einzelheiten über Begriffe die "Echtzeit-

betrieb", "Echtzeitverarbeitung", "Multitasking" und ähnliches entnehmen.

[0010] Aus DE-Z: "Industrie-Anzeiger" 34/1989, S. 34-36 ("Die moderne CNC-Steuerung") läßt sich eine Anlage für eine zentrale NC-Bearbeitung entnehmen, bei der Teile einer Werkzeugmaschine von einem mit einem Host-Rechner gekoppelten Leitreechner gesteuert werden, der auf eine NC-Programmierung und eine PLC-Programmierung (PLC = programmed logic control) zurückgreift.

[0011] Eine spezielle Architektur für CNC-Steuerungen ist beschrieben in "Technische Rundschau" 26/91, S. 62-68. Dabei geht es speziell um die Koppelung mehrerer, über eine Busleitung verbundener CNC-Steuerungen.

[0012] Im Rahmen eines hierarchischen Steuerungssystems besteht die Möglichkeit, in einer bestimmten Ebene der Hierarchie eine modulare oder Zellensteuerung vorzusehen, die weitestgehend anlagenunabhängig arbeitet ("Automatisierungstechnische Praxis atp" 31 (1989) 4, S. 166—173). Das hierarchische System umfasst in der obersten Stufe Entwurf und Planung (CAD, CAP) in einer untergeordneten Stufe Leitreechner für Montage bzw. Fertigung, und in einer darunter angesiedelten Ebene eine Reihe von sogenannten Montagezellen, von denen jede einen eigenen Zellenrechner besitzt, der für Robotersteuerung, Maschinensteuerung, speicherprogrammierte Steuerung und Überwachungsaufgaben eingesetzt wird.

[0013] Aus der US-A-4 684 862 ist eine Steuervorrichtung für eine mehrachsige Werkzeugmaschine bekannt, bei der baulich identische Prozessoren für jeweils eigene Steueraufgaben eingesetzt werden. Jeder einzelne Prozessor steuert einen Komplexen Bewegungsablauf oder mehrere einfache Bewegungsabläufe und empfängt hierzu von einer NC-Einheit Sollpositionsdaten und aus einem Speicher zwischengespeicherte Istpositionsdaten.

Aufgabenstellung

[0014] Es ist Aufgabe der Erfindung, ein Multitask-Steuersystem zum Steuern des Betriebs mehrerer gesteuerter Geräte anzugeben, mit dem eine parallele Steuerung der Prozesse für logische Achsen der jeweiligen Geräte mit Hilfe eines Multitasking-Prozessors auf Timesharing-Basis durchgeführt wird, während die Steuerung von Arbeitsabläufen durch einen Echtzeit-Prozessor vorgenommen wird, so daß der Benutzer des Systems imstande ist, die Software (Rechnerprogramme) in einfacher Weise zu erstellen, ohne daß dabei die Zuverlässigkeit und die Leistungsfähigkeit der Software beeinträchtigt wird.

[0015] Gelöst wird diese Aufgabe durch die in Anspruch 1 angegebene Erfindung.

[0016] Ausgestaltungen der Erfindung sind in den abhängigen Ansprüchen angegeben.

[0017] Das Multitask-Steuersystem besitzt den Echtzeitprozessor mit der Steuerung zum Steuern

des Betriebs, zum Beispiel der Stellung und/oder der Bewegung, der gesteuerten Geräte. Der Echtzeitprozessor kann mithin die gesteuerten Geräte auf Echtzeitbasis steuern, er kann beispielsweise den Arbeitsablauf für die Schweißpistolen oder die Erregung von Servomotoren, welche die Schweißpistolen bewegen, steuern.

[0018] Um eine Stellungs- und/oder Betriebssteuerung in dem Echtzeitprozessor vorzunehmen, besitzt das Multitask-Steuersystem außerdem den Multitask-Prozessor, der die Anwendungstasks enthält, die zu den jeweiligen logischen Achsen und/oder den gesteuerten Geräten gehören. Der Multitask-Prozessor gibt Betriebsbedingungen für die Schweißpistole und bestätigt die Beendigung des Arbeitsablaufs an den Schweißpistolen, oder er gibt Zielpositionsdaten für die Lagesteuerung der Servomotoren und bestätigt den Betrieb der Servomotoren aus deren Istpositionsdaten.

[0019] Der Kommunikationsspeicher befindet sich zwischen dem Echtzeitprozessor und dem Multitask-Prozessor, und er transferiert beispielsweise die Betriebsbedingungen für die Schweißpistolen oder die Zielpositionsdaten und die Istpositionsdaten aus dem Echtzeitprozessor zu dem Multitask-Prozessor, oder aus dem Multitask-Prozessor zu dem Echtzeitprozessor.

[0020] Die Anwendungstasks entsprechen den jeweiligen logischen Achsen, und sie geben die Zielpositionsdaten über den Kommunikationsspeicher an den Echtzeitprozessor, und empfangen die Istpositionsdaten von dem Echtzeitprozessor. Der Echtzeitprozessor berechnet zurückzulegende Strecken von den Istpositionen zu den Zielpositionen auf der Grundlage der Zielpositionsdaten und der Istpositionsdaten, und er gibt auf der Grundlage der Strecken einen Geschwindigkeitsbefehl aus, um den Betrieb der Servomotoren zu steuern.

[0021] Der Multitask-Prozessor besitzt die PLC-(Ablaufsteuerung-)Task zum Übertragen von Information zu und zum Empfangen von Information von der Ablaufsteuerung für eine zentralisierte Steuerung der gesteuerten Geräte. Der Steuerkasten besitzt die Anwendungstasks zur Kommunikation mit den Anwendungstasks des Multitasking-Prozessors, zum Anzeigen von Daten auf einem Anzeigebildschirm und zum Eingeben von Tastaturwerten, wobei die Multitask-Verarbeitung außerdem die Kommunikationstasks zum Kommunizieren mit den Anwendungstasks des Steuerkastens enthält.

[0022] Die Steuerungsvorgänge für die gesteuerten Geräte werden also im Multitaskbetrieb durch den Multitask-Prozessor abgewickelt, und die Steuerungsfunktion, die sofort ausgeführt werden muß, beispielsweise das Steuern eines Schweißvorgangs oder dergleichen, erfolgt durch den Echtzeitprozessor auf Echtzeitbasis. Deshalb kann das Multitask-Steuersystem Steuerungsprozesse handhaben, die aufgrund ihrer Natur echtzeitorientiert sind, und es gestattet dem Benutzer das leichte Erstellen von

Programmen ohne Beeinträchtigung der Zuverlässigkeit und Leistungsfähigkeit, so daß mit vergleichsweise geringem personellen Aufwand entwickelt und revidiert werden können.

Ausführungsbeispiel

[0023] Im folgenden werden Ausführungsbeispiele der Erfindung anhand der Zeichnung näher erläutert. Es zeigen:

[0024] **Fig. 1** ein Blockdiagramm einer Multitask-Steuerungsvorrichtung, in die das erfindungsgemäße Multitask-Steuersystem inkorporiert ist;

[0025] **Fig. 2** ein Blockdiagramm einer logischen Anordnung eines Steuerkastens;

[0026] **Fig. 3** ein Blockdiagramm einer ersten Ausführungsform des erfindungsgemäßen Multitask-Steuersystems;

[0027] **Fig. 4** ein Blockdiagramm einer zweiten Ausführungsform eines erfindungsgemäßen Multitask-Steuersystems;

[0028] **Fig. 5** ein Blockdiagramm eines Schweißrobotersystems;

[0029] **Fig. 6** ein Blockdiagramm einer hierarchischen Steuerungsstruktur für das Schweißrobotersystem;

[0030] **Fig. 7** ein Flußdiagramm einer Verarbeitungsablauffolge einer Kommunikationstask;

[0031] **Fig. 8** ein Flußdiagramm einer Verarbeitungsfolge einer PLC-Task;

[0032] **Fig. 9** ein Flußdiagramm einer Positionierungs-Ablauffolge einer Anwendungstask;

[0033] **Fig. 10** ein Flußdiagramm einer Verarbeitungsfolge in einem Positionierungs-Steuerprozessor;

[0034] **Fig. 11** ein Flußdiagramm eines Verarbeitungsablaufs zur Positionierungssteuerung, die von Hand mittels des Steuerkastens vorgenommen wird;

[0035] **Fig. 12** ein Flußdiagramm eines Positionierungsablaufs in einem Echtzeitprozessor;

[0036] **Fig. 13** ein Flußdiagramm eines Schweißvorgangs einer Anwendungstask; und

[0037] **Fig. 14** ein Flußdiagramm eines Schweißvorgangs in dem Echtzeitprozessor.

Erste Ausführungsform

[0038] **Fig. 1** zeigt als Blockdiagramm eine Multitask-Steuervorrichtung **10**, die das erfindungsgemäße Multitask-Steuersystem beinhaltet. Die Multitask-Steuervorrichtung **10** enthält allgemein einen Multitask-Prozessor **12**, einen Echtzeitprozessor **14** und einen Kommunikationsspeicher **16**.

[0039] Der Multitask-Prozessor **12** enthält eine Kommunikations-Schnittstelle (KOM-I/F) **12a** zur Kommunikation mit einem (später noch beschriebenen) Steuerkasten (TBOX) **22**, einen Mikroprozessor (CPU-1) **12b**, einen Festspeicher (ROM-1) **12c** und einen Schreib-/Lesespeicher (RAM-1) **12d**. Der Echtzeitprozessor **14** enthält einen Mikroprozessor

(CPU-2) **14a**, einen Festspeicher (ROM-2) **14b** und einen Schreib-/Lesespeicher (RAM-2) **14c**.

[0040] Der ROM-1 **12c** speichert verschiedene Tasks, darunter eine Anwendungstask zum Positionieren verschiedener Achsen von gesteuerten Geräten **20**, bei denen es sich zum Beispiel um Schweißroboter oder Montageroboter handelt, und eine Anwendungstask für einen Schweißprozeß. Der ROM-2 **14b** speichert ein Programm zur Positionierungssteuerung in dem Echtzeitprozessor **14** und ein Programm zur Schweißpistolen/Transformator-Steuerung, die für Schweißzwecke dient.

[0041] Der Echtzeitprozessor **14** ist an die gesteuerten Geräte **20** angeschlossen, um die Positionierung und den Arbeitsablauf der gesteuerten Geräte **20** zu steuern. Nach **Fig. 2** ist der Echtzeitprozessor **14** mit einer Servoverstärker-Schnittstelle (SA I/F) **18** verbunden, um den Servoverstärkern zum Aktivieren der in den gesteuerten Geräten **20** befindlichen Servomotoren Steuer-Ausgangssignale zuzuführen und so einen Schweißarm oder eine Roboterhand der gesteuerten Geräte **20** zu positionieren.

[0042] **Fig. 2** zeigt in Form eines Blockdiagramms die logische Ausgestaltung des Steuerkastens **22**, der mit den Anwendertasks der Multitask-Steuervorrichtung **10** kommuniziert und Anzeige- sowie Tastatureingabefunktionen ausübt. Der Steuerkasten **22** setzt sich lediglich zusammen aus einem Multitask-Prozessor **22a**, in welchem als Anwendertasks eine Monitortask **22b**, eine Kommunikations-Task **22c**, eine Tastatur-Task **22d** und eine Flüssigkristallanzeige-(LCD)-Task **22e** registriert sind. Der Multitask-Prozessor **22a** besitzt eine Kommunikations-Bibliothek **22f**, bei der es sich um eine Programmbibliothek handelt, in der Programme zur Ausführung einer Kommunikation von Task zu Task zwischen der Multitask-Steuervorrichtung **10** und dem Steuerkasten **22** über die Kommunikations-Task **22c** speichert. Die Kommunikations-Bibliothek **22f** ist mit der Multitask-Steuervorrichtung **10** über eine Kommunikations-Schnittstelle (Kommunikations-I/F) **22g** verbunden, um Daten und Befehle zu senden und zu empfangen.

[0043] Die LCD-Task **22e** bringt Daten auf dem Anzeigebildschirm einer LCD-Einheit **22i** zur Anzeige, die von der Multitask-Steuervorrichtung **10** übertragen werden. Die LCD-Task **22e** führt eine Fern-Anzeigeschirmsteuerfunktion zum Steuern des LCD-Anzeigeschirms auf der Grundlage spezieller Ablaufdaten innerhalb der von der Multitask-Steuervorrichtung **10** übertragenen Daten aus, und sie führt außerdem eine Fern-Verbindungsfunktion aus, um eine Kommunikationseinheit auszuwählen, an die die Tastatur-Task **22d** Daten übertragen soll.

[0044] Die Tastatur-Task **22d** sendet Daten an die Multitask-Steuervorrichtung **10** auf der Grundlage von Tastenbetätigungen einer Tastatur **22h**. Die von der Tastatur-Task **22d** übertragenen Daten werden zu einer Task gesendet, die von der von der LCD-Task **22e** wahrgenommenen Fern-Verbindungs-

funktion ausgewählt wurde. Die Tastatur-Task **22d** ist im Normalbetrieb und im Zifferneingabe-Betrieb betreibbar. Beim Zifferneingabebetrieb bringt die LCD-Einheit **22i** ein numerisches Eingabefenster auf den Anzeigeschirm zur Darstellung. Der Benutzer stellt zu sendende numerische Daten mit Hilfe eines Zifferntastenfeldes der Tastatur **22h** in dem numerischen Eingabefenster ein. Wenn dann später eine Eingabetaste auf der Tastatur **22h** durch den Benutzer gedrückt wird, ist das numerische Eingabefenster beendet, und die numerischen Daten werden zu der Multitask-Steuervorrichtung **10** übertragen. Ein Umschalten zwischen Normalbetrieb und numerischen Eingabebetrieb wird durch die Ablaufdaten zum Steuern des LCD-Anzeigeschirms gesteuert.

[0045] Ein direkter Datenaustausch zwischen der Tastatur-Task **22d** und der LCD-Task **22e** erfolgt in der Steuerbox **22** praktisch nicht, allerdings wird ein solcher Datenaustausch über die Anwendertask innerhalb der Multitask-Steuervorrichtung **10** ausgeführt.

[0046] **Fig. 3** zeigt in Blockform eine erste Ausführungsform des erfindungsgemäßen Multitask-Steuersystems. Diejenigen Bezugszeichen in **Fig. 3**, die mit denjenigen in **Fig. 1** identisch sind, bezeichnen identische Teile. Das Multitask-Steuersystem ist in **Fig. 3** beispielhaft als System zum Steuern zweier uniaxialer Positioniereinrichtungen, zum Beispiel zum Positionieren von Robotern, dargestellt, was den jeweiligen logischen Achsen in den gesteuerten Geräten **20** entspricht. Die zwei uniaxialen Positioniereinrichtungen der gesteuerten Geräte **20** enthalten zwei Servomotoren **30a** bzw. **30b** und zwei Stellungsdetektoren **32a** und **32b**, die mit den Servomotoren **30a** bzw. **30b** mechanisch gekoppelt sind. Der Servomotor **30a** und der Stellungsdetektor **32a** sind elektrisch mit einem Servoverstärker **34a** verbunden, und der Servomotor **30b** und der Stellungsdetektor **32b** sind elektrisch mit einem Servoverstärker **34b** verbunden.

[0047] Bei Erhalt von Geschwindigkeitsbefehlen steuern die Servoverstärker **34a** und **34b** die Umdrehungsgeschwindigkeiten oder Drehzahlen der Servomotoren **30a** bzw. **30b**, und sie geben festgestellte Positionen oder Stellungen der Motoren aus den Stellungsdetektoren **32a** bzw. **32b** als Istpositionsdaten oder Iststellungsdaten aus. Die Geschwindigkeitsbefehle und die Positionsdaten werden von dem Echtzeitprozessor **14** geliefert bzw. an den Echtzeitprozessor geliefert. Der Echtzeitprozessor **14** besitzt Positionier-Steuerprozessoren **14a1** und **14b1**, die zu den jeweiligen logischen Achsen gehören, d.h. zu den Servoverstärkern **34a** und **34b**, um die Servomotoren **30a**, **30b** zu aktivieren und dadurch die Geräte in Zielpositionen zu bringen und dort zu halten. Die Positionier-Steuerprozessoren **14a1** und **14b1** geben Geschwindigkeitsbefehle an die zugehörigen Servoverstärker **34a** und **34b**, und sie empfangen Istpositionsdaten **40a**, **40b** von den Servomotoren **30a**, **30b** auf der Grundlage der von den Positionsdetektoren **32a**, **32b** erfaßten Positionen.

[0048] Die Positionier-Steuerprozessoren **14a1**, **14b1** beginnen den Betrieb, wenn in dem Kommunikationsspeicher **16** Zielpositionen **36a**, **36b** und Startbefehle **38a**, **38b** eingestellt sind. Die Positionier-Steuerprozessoren **14a1**, **14b1** stellen in dem Kommunikationsspeicher **16** außerdem die ermittelten Positionen von den Stellungsdetektoren **32a**, **32b** als die Istpositionsdaten **40a**, **40b** ein.

[0049] Der Kommunikationsspeicher **16** ist an den Multitask-Prozessor **16** zum Übertragen von Daten zwischen dem Multitask-Prozessor **12** und dem Echtzeitprozessor **14** angeschlossen. Der Multitask-Prozessor **12** besitzt Anwendungstasks **42a** und **42b**, um die Zielpositionen **36a**, **36b** und die Startbefehle **38a**, **38b** an die Positionier-Steuerprozessoren **14a1**, **14b1** des Echtzeitprozessors **14** zu geben und so die Servomotoren **30a** und **30b** zu aktivieren, und um den Betrieb der Servomotoren **30a**, **30b** anhand der Istpositionsdaten **40a**, **40b** zu bestätigen.

[0050] Die Anwendungstasks **42a**, **42b** besitzen eine Login-Funktion zum Zulassen der Steuerungsabläufe aus den jeweiligen Steuerkästen **22**, damit die Servomotoren **30a**, **30b** von den Steuerboxen **22** aus gesteuert werden können. Die Steuerkästen **22** besitzen unabhängigen Kommunikationspuffer **24a1-24a4**, **24b1-24b4** entsprechend den zugehörigen Servomotoren **30a**, **30b**, so daß die zwei Steuerkästen **22** jederzeit unabhängig voneinander betrieben werden können.

[0051] Der Multitask-Prozessor **12** besitzt eine PLC-(Ablaufsteuerung)-Task **44** zum Austauschen von Information für die PLC und eine zentralisierte Steuerung und zur Bereitstellung von Ergebnissen für die Anwendungstasks **42a**, **42b**. Die PLC-Task **44** liest und schreibt Ablaufsteuerungs-Kontaktdaten (gemeinsame Kontaktdaten einschließlich Servo-Ein-Daten, Start-Daten, Handbetriebs-Daten, Funktion-Ein-Daten und Musternummer-Daten, und individuelle Kontaktdaten einschließlich Stop-Daten, Alarm-Daten, Verriegelungs-Antwort-Daten, Startbedingungen-Daten, Verriegelungs-Daten, Watchdog-Daten und Normalsignal-Daten) aus und in PLC-Puffer **44a**, **44b** zu allen Zeiten. Deshalb wird eine zentralisierte Steuerung durch die Ablaufsteuerung möglich.

[0052] Sämtliche Kommunikation zwischen dem Multitask-Prozessor **12** und den Steuerkästen **22** oder der PLC (Ablaufsteuerung) werden über eine Kommunikations-Task **46** und die Kommunikations-Schnittstelle **12a** abgewickelt. Der andere Teilnehmer der Kommunikationsbindung, welcher mit der Kommunikations-Task **46** in Verbindung tritt, wird von jeder Task individuell ausgewählt. Kommunikations-Puffer **24c1**, **24c2** dienen der PLC-Task **44** zur Kommunikation mit der PLC.

[0053] Die Kommunikations-Task **46** bewirkt eine Kommunikation der Art, daß sie Daten zu einer Zielstelle sendet, die von dem Sender bestimmt wird, wobei der Empfänger grundsätzlich keine eingegebenen Daten zurückweist und es an den Anwendungstasks

42, 42b und der PLC-Task **44** des Empfängers liegt, zu entscheiden, ob die eingegebenen Daten gebraucht werden oder nicht. Aus diesem Grund ist es nicht nötig, den anderen Teilnehmer der Kommunikationsverbindung für solche Tasks auszuwählen, die lediglich empfangene Daten handhaben, und man muß den anderen Teilnehmer der Kommunikation nur für solche Tasks auswählen, welche Daten senden.

[0054] Andere Teilnehmer von Kommunikationen sind Tasks, oder Daten werden von einer Task zu einer anderen Task über Zwischen-Task-Kommunikationen übertragen. Somit bedeutet das Auswählen des anderen Teilnehmers eine Kommunikationsverbindung die Angabe einer Task und das Erfordernis, eine Knotennummer und eine Tasknummer auszuwählen. Um eine derartige Funktion auszuführen, sendet die Kommunikations-Task **46** empfangene Daten an den Empfangspuffer der Bestimmungstask, und sie sendet in dein Sendepuffer jeder Task eingestellte Daten zu einer Task eines anderen Knoten als dein anderer Teilnehmer der Kommunikation, der von jeder Task ausgewählt wird.

[0055] Die Kommunikationspuffer **24a1** bis **24a4**, **24b1** bis **24b4**, **24c1**, **24c2** sind Puffer zur Bewerkstellung derartiger Zwischen-Task-Kommunikationen, und sie sind paarweise für Empfangs- und Sendepuffer vorgesehen. Wenn es zu sendende Daten gibt, wählt jede Anwendungstask den anderen Kommunikationsteilnehmer aus und stellt die Daten in einem Sendepuffer ein. Die Kommunikations-Task **46** sendet die Daten zu dem anderen Kommunikationsteilnehmer. Wenn Daten zum Empfangen vorhanden sind, stellt die Kommunikations-Task **46** die an sie selbst gerichteten Daten in einem Empfangspuffer ein, und die entsprechende Anwendungstask **42a** oder **42b** liest die Daten aus dein Empfangspuffer.

[0056] In dein Multitask-Steuersystem bedient der Benutzer unter Verwendung des Steuerkastens **22** jedes über eine Kommunikationsleitung angeschlossene Gerät. Da Kommunikationen als Zwischen-Task-Kommunikationen abgewickelt werden, betätigt der Benutzer eine Anwendungstask für jedes Gerät. Als erstes legt der Benutzer mit Hilfe des Steuerkastens (Steuerbox) fest, welches Gerät und welche Anwendungstask für den Betrieb vorgesehen sind. Dann sendet der Steuerkasten eine Benutzungsstartaufforderung an die festgelegte Anwendungstask.

[0057] Ansprechend auf die Benutzungsstartanforderung seitens des Steuerkastens beginnen die Anwendungstasks **42a**, **42b** mit dem Senden verschiedener vorprogrammierter Anzeigeelemente an den Steuerkasten. Dann empfangen die Anwendungstask **42a**, **42b** von dem Steuerkasten Steuerinformation und können verschiedene vorprogrammierte Anzeigefunktionen über den Betrieb des Steuerkastens aufrufen. Ein Prozeß, bei dem der Steuerkasten hinsichtlich einer gewissen Anwendungstask betriebsbereit gemacht wird, wird als "Lo-

gin" bezeichnet, während ein Prozeß zum Unterbrechen des Betriebs des Steuerkastens und zum In-Die-Lage-Versetzen des Steuerkastens, eine Anwendungstask für den Betrieb auszuwählen, als "Logout" bezeichnet wird.

[0058] Standard-Anwendungstasks, die von dein Steuerkasten **22** nicht gehandhabt werden müssen, erfordern Funktionen zur Bewirkung eines Login und eines Logout, und sie werden durch Menüauswahl unter Verwendung von Funktionstasten gehandhabt. Die Funktionstasten umfassen fünf Funktionstasten F1 bis F5. Ein Verarbeitungsprogramm für den Steuerkasten **22** wird derart generiert, daß, wenn eine Funktionstaste gedrückt wird, eine Status-Variable variiert wird und irgendeine von verschiedenen Funktionen aufgerufen werden kann durch eine Kombination der Status-Variablen und der gedrückten Funktionstaste. Der Erzeuger der Anwendungstask wählt zuerst eine auszuführende Funktion aus, entwirft einen Operationsprozeß, setzt den Operationsprozeß in ein Zustands-Übergangsdiagramm um, ordnet den verschiedenen Zuständen des Zustands-Übergangsdiagramms Nummern zu und verwendet die zugewiesenen Nummern als die Nummern der Statusvariablen, um dadurch eine Anwendungstask zu erzeugen.

Zweite Ausführungsform

[0059] **Fig. 4** zeigt eine zweite Ausführungsform des erfindungsgemäßen Multitask-Steuersystems. In der Figur sind gleiche Teile wie in **Fig. 1** mit entsprechenden Bezugszeichen versehen. Das Multitask-Steuersystem ist in **Fig. 4** beispielhaft für den Zweck dargestellt, zwei Schweißvorrichtungen als die gesteuerten Geräte **20** zu steuern, beispielsweise zwei Schweißroboter. Die gesteuerten Geräte **20** (Schweißvorrichtungen) enthalten Schweißpistolen **50a** und **50b** und zwei Transformatoren **52a** und **52b**, die an zugehörige Wechselrichter **54a** bzw. **54b** angeschlossen sind.

[0060] Die Wechselrichter **54a** und **54b** empfangen PWM-Vorgabesignale und Ausgangsstrom-Rückkopplungssignale (FB). Die PWM-Vorgabesignale werden von dem Echtzeitprozessor **14** geliefert, und die Strom-Rückkopplungssignale werden an den Echtzeitprozessor **14** geliefert. Der Echtzeitprozessor **14** liefert außerdem Ventilsteuersignale zum Steuern der Öffnungs- und der Schließbewegung der Schweißpistolen **50a** und **50b**.

[0061] Der Echtzeitprozessor **14** besitzt Pistolen/Transformator-Steuerprozessoren **14a2**, **14b2**, die zu den jeweiligen Schweißrobotern gehören, d.h. zu den Wechselrichtern **54a**, **54b**, den Transformatoren **52a**, **52b** und den Schweißpistolen **50a**, **50b**, um Werkstücke entsprechend spezifizierter Druckbeaufschlagungs- und Erregungspläne zu schweißen und die Zufuhr eines spezifischen konstanten Stroms zu den Schweißpistolen zu steuern. Wenn die Stromvorgabewerte **56a** und **56b** und die Druckvorgabewerte

58a und **58b** in einem Kommunikationsspeicher **16** eingestellt sind, starten die Pistolen/Transformator-Steuerprozessoren **14a2**, **14b2** mit dem Schweißen von Werkstücken, und sie steuern den Strom, der zum Schweißen der Werkstücke geliefert wird. Wenn das Aktivieren der Schweißpistolen **50a** und **50b** und die Steuerung des Stroms vorbei sind, geben die Pistolen/Transformator-Steuerprozessoren **14a2** und **14b2** Beendigungssignale **60a** und **60b** an den Kommunikationsspeicher **16**.

[0062] Der Kommunikationsspeicher **16** ist an den Multitask-Prozessor **12** angeschlossen, um Daten zwischen dem Multitask-Prozessor **12** und dem Echtzeitprozessor **14** zu übertragen. Der Multitask-Prozessor **12** besitzt Anwendungstasks **42a**, **42b** zur Bereitstellung von Stromvorgabewerten **56a**, **56b**, zum Ausgeben eines Aktivierungsplans und von Druckbeaufschlagungsbefehlen **58a**, **58b** an die Pistolen/Transformator-Steuerprozessoren **14a2**, **14b2**, und zum Bestätigen der Beendigung des Schweißprozesses mit den Beendigungssignalen **60a**, **60b**.

[0063] Die Anwendungstasks **42a**, **42b** besitzen eine Login-Funktion zum Ermöglichen der Ausführung von Steueroperationen seitens eines Steuerkastens **22**, damit über den Steuerkasten **22** Aktivierungs- und Schweißbedingungen eingestellt werden können. Die Anwendungstasks **42a**, **42b** können in Kombination mit der Positioniersteuerung des Multitask-Steuersystems nach der oben beschriebenen ersten Ausführungsform eingesetzt werden.

[0064] Als erstes wird der andere Teilnehmer der Kommunikationsverbindung, der zwischen den Anwendungstask kommunizieren soll, eingestellt. Wenn dann ein Schweißbefehl auf der Seite der Positioniersteuerung ausgeführt wird, wird ein vorbestimmter Positionierprozeß ausgeführt, und anschließend wird eine Schweißbedingungsnummer des Schweißbefehls an eine Anwendungstask der Schweißsteuerseite gesendet, von der Druckbeaufschlagungs- und Erregungspläne entsprechend der Schweißbedingungsnummer gelesen und dem Echtzeitprozessor **14** über den Kommunikationsspeicher **16** zum Schweißen von Werkstücken zugesendet werden. Wenngleich kein Schweißvorgang mit zentralisierter Steuerung vorgenommen wird, werden einige Kontakte einer PLC (Ablaufsteuerung) verwendet. Kommunikationspuffer **24a1-24a4**, **24b1-24b4**, **24c1**, **24c2**, eine Kommunikations-Task **46**, PLC-Puffer **44a**, **44b** und eine PLC-Task **44** haben die gleichen Funktionen, wie sie in **Fig. 3** dargestellt sind.

[0065] Das oben erläuterte und in **Fig. 4** dargestellte Schweißsteuersystem und das Positioniersteuersystem nach **Fig. 3** können in einem Schweißrobotersystem kombiniert werden, wie es unten beschrieben wird.

[0066] Schweißrobotersystem unter Verwendung des Multitask-Steuersystems:

[0067] **Fig. 5** zeigt in Blockdiagrammform ein Schweißrobotersystem, welches das erfindungsge-
mäßige Multitask-Steuersystem beinhaltet. An die Mul-

titask-Steuervorrichtung **14** sind über eine Kommunikationsleitung **4** eine Ablaufsteuerung (PLC) **8**, ein Steuerkasten (TBOX) **22**, ein Positiertreiber (POS) **33** und ein Schweißtreiber (WE) **51** angeschlossen.

[0068] Der Positioniertreiber **43** betätigt Positionierroboter (R/B) **31a**, **31b** zum Positionieren der jeweiligen Schweißpistolen **50a**, **50b** in den jeweils erforderlichen Stellungen, was unter der Steuerung der Positionier-Steuerprozessoren **14a1**, **14b1** (wie **Fig. 3**) geschieht. Wie in **Fig. 3** gezeigt ist, setzen sich die Positionierroboter **31a**, **31b** aus jeweiligen Servomotor-Verstärkern **34a**, **34b** und den Servomotoren **30a**, **30b** zusammen. Der Schweißtreiber **51** aktiviert Transformatoren **52a**, **52b**, damit die Schweißpistolen **50a**, **50b** imstande sind, Werkstücke unter der Steuerung der Pistolen/Transformator-Steuerprozessoren **14a2**, **14b2** zu schweißen (siehe **Fig. 4**).

[0069] **Fig. 6** zeigt einen hierarchischen Steuerablauf des in **Fig. 5** gezeigten Schweißrobotersystems. Wie in **Fig. 6** dargestellt ist, besitzt die Ablaufsteuerung (PLC) **8** interne Kontakte, die in Bereiche gruppiert sind, welche den jeweiligen Geräten entsprechen, d.h. in einen POS-Bereich **8a**, einen Schweiß-Bereich **8b** und einen Bereich für das Gerät **n**, **8c**, auf die durch die jeweiligen Geräte zugegriffen werden kann, d.h. durch den Positioniertreiber (POS) **33**, den Schweißtreiber (SCHWEISS) **51** und ein Gerät **n**. Die Inhalte der internen Kontaktdaten können durch die jeweiligen Geräte definiert werden, und sie können über eine Steuerkonsole **6** eingestellt werden. Signale, die kennzeichnend für Start, Stopp, Musternummer und dergleichen sind, die also die Zustände der internen Kontakte repräsentieren, werden an den Positioniertreiber **33** und den Schweißtreiber **51** gesendet.

[0070] Steuer- und Schweißbefehle von den Steuerkästen **22** werden an den Positioniertreiber **33** und den Schweißtreiber **51** gesendet. Es wird ein sogenannter Playback-Prozeß, welcher Schritt für Schritt einen seitens der Steuerkästen **22** vorgegebenen Schweißprozeß ausführt, gestartet durch einen Steuerungsbefehl, der von dem entsprechenden Steuerkasten **22** an den Positioniertreiber **33** geliefert wird, und einen Schweißbefehl, der von dem entsprechenden Steuerkasten **22** an den Schweißtreiber **51** geliefert wird.

[0071] **Fig. 7** zeigt eine Verarbeitungs-Ablauffolge der Kommunikations-Task **46**. Gemäß **Fig. 7** wird die Kommunikations-Schnittstelle **12a** in einem Schritt S11 initialisiert. Die Kommunikations-Task **46** prüft im Schritt S12 einen Kommunikationszustand (Status), im Schritt S13 empfängt sie Daten und im Schritt S14 sendet sie Daten. Anschließend stellt die Kommunikations-Task **46** fest, ob die durch einen Anwendungstimer eingestellte Zeit verstrichen ist oder nicht. Falls nicht, geht die Steuerung zum Schritt S12 zurück, bei dem die Kommunikations-Task **46** erneut einen Kommunikationszustand prüft. Falls die erwähnte Zeit verstrichen ist, geht die Steuerung zum Schritt

S16, in welchem die Kommunikations-Task **46** den Anwendungs-Timer einstellt. Anschließend führt die Kommunikations-Task **46** eine Anwendung aus, um eine Bestimmungsadresse für die PLC (Ablaufsteuerung) zu erhalten (Schritt S17), woraufhin die Steuerung zum Schritt S12 zurückkehrt.

[0072] **Fig. 8** zeigt einen Verarbeitungsablauf der PLC-Task **44**. Wie oben beschrieben, sind die internen Kontakte der PLC-Task **44** gruppenweise in Bereiche geordnet, welche den jeweiligen Geräten entsprechen, d.h. einen POS-Bereich **8a**, einen Schweiß-Bereich **8b**, und einen Bereich **8c** für ein Gerät *n*, auf die durch die jeweiligen Geräte zugegriffen werden kann, d.h., durch den Positioniertreiber (POS) **33**, den Schweiß-Treiber (SCHWEISS) **51** und ein Gerät *n*. Die Inhalte der internen Kontaktdaten können von den jeweiligen Geräten definiert werden, und sie können über eine Steuerkonsole **6** eingestellt werden (**Fig. 6**).

[0073] Die PLC-Task **44** sendet einen PLC-Schreibbefehl im Schritt S51, stellt im Schritt S22 einen Rückübertragungs-Timer und empfängt im Schritt S23 PLC-Schreiben-Fertig-Daten. Dann stellt die PLC **44** fest, ob ein Empfang stattgefunden hat oder nicht (Schritt S24). Falls es einen Empfang gibt, springt die Steuerung zu einem Schritt S26. Falls kein Empfang gegeben ist, stellt die PLC **44** im Schritt S25 fest, ob die durch den Rückübertragungs-Timer eingestellte Zeit verstrichen ist oder nicht. Falls nicht, geht die Steuerung zum Schritt S27 zurück. Falls die Zeit verstrichen ist, geht die Steuerung zum Schritt S21 zurück.

[0074] Im Schritt S26 sendet die PLC **44** einen PLC-Auslesebefehl. Anschließend stellt die PLC **44** im Schritt S27 den Rückübertragungs-Timer und empfängt im Schritt S28 PLC-Auslese-Fertig-Daten. Anschließend stellt die PLC **44** fest, ob ein Empfang vorhanden ist oder nicht (Schritt S29). Gibt es einen Empfang, so geht die Steuerung zurück zum Schritt S21. Gibt es keinen Empfang, stellt die PLC **44** fest, ob die durch den Rückübertragungs-Timer eingestellte Zeit verstrichen ist oder nicht (Schritt S30). Falls die Zeit noch nicht verstrichen ist, geht die Steuerung zum Schritt S28 zurück. Ist die Zeit jedoch verstrichen, geht die Steuerung zum Schritt S26 zurück.

[0075] **Fig. 9** zeigt eine Positionier-Ablauffolge der Anwendungstasks **42a** und **42b**, wobei die Positionierablauffolge ein Playback-Prozeß ist, welcher Schritt für Schritt einen Schweißvorgang ausführt, der seitens der Steuerkästen **22** vorgegeben wurde.

[0076] Die Anwendungstasks **42a** und **42a** stellen in einem Schritt S31 fest, ob ein Schritt-Ausführungsflag "0" ist oder nicht. Ist das Flag nicht "0" springt die Steuerung zu einem Schritt S38. Ist das Schritt-Ausführungsflag "0", stellen die Anwendungstasks **42a** und **42b** fest, ob der derzeitige Schritt ein Bewegungsbefehl ist oder nicht (Schritt S32). Ist er kein Bewegungsbefehl, führen die Anwendungstasks **42a**, **42b** andere als den Bewegungsbefehl aus (Schritt S33). Ist der derzeitige Schritt ein Bewe-

gungsbefehl, geht die Steuerung zu einem Schritt S34, in welchem die Anwendungstask **42a**, **42b** eine Zielposition für den laufenden Schritt lesen. Die Anwendungstasks **42a**, **42b** steilen dann die Zielposition für den derzeitigen Schritt in Zielpositionen **36a**, **36b** (siehe **Fig. 3**) innerhalb des Kommunikationsspeichers **16** ein (Schritt S36). Die Anwendungstasks **42a**, **42b** schalten im Schritt S36 ein Startsignal für den Kommunikationsspeicher **16** ein und setzen das Schritt-Ausführungsflag im Schritt S37 auf "1".

[0077] Im Schritt S38 bestimmen die Anwendungstasks **42a**, **42b**, ob das Schritt-Ausführungsflag "1" ist oder nicht. Ist das Flag "1", ist die Verarbeitung beendet. Ist das Flag "1", bestimmen die Anwendungstasks **42a**, **42b**, ob der laufende Schritt ein Bewegungsbefehl ist oder nicht (Schritt S39). Liegt kein Bewegungsbefehl vor, führen die Anwendungstasks **42a**, **42b** andere Befehle als den Bewegungsbefehl im Schritt S40 aus.

[0078] Wenn der laufende Schritt ein Bewegungsbefehl ist, geht die Steuerung zu einem Schritt S41, in welchem die Anwendungstasks **42a**, **42b** die Daten über die derzeitige Position, **40a**, **40b** von den Zielpositionen **36a**, **36b** im Kommunikationsspeicher **16** subtrahieren und bestimmen, ob die Differenz in einem erreichbaren Bereich liegt oder nicht. Die derzeitigen Positionsdaten **40a**, **40b** sind von dem Echtzeitprozessor **14** in dem Kommunikationsspeicher **16** eingeschrieben worden.

[0079] Wenn die Differenz in dem erreichbaren Bereich liegt (Schritt S41), setzen die Anwendungstask **42a**, **42b** das Schrittausführungsflag im Schritt S42 auf "0", und die Verarbeitung wird beendet. Liegt die Differenz nicht in der erreichbaren Zone, wird die Verarbeitung beendet.

[0080] **Fig. 10** zeigt eine Verarbeitungsablauffolge der Positioniersteuerprozessoren **14a1**, **14b1** des Echtzeitprozessors **14**. Die Positioniersteuerprozessoren **14a1**, **14b1** lesen die derzeitigen Stellungen der Servomotoren **30a**, **30b** aus den Servoverstärkern **34a**, **34b** (Schritt S51), und sie schreiben dann die derzeitigen Stellungen in die Daten über die derzeitigen Positionen **40a**, **40b** innerhalb des Kommunikationsspeichers **16** ein.

[0081] Anschließend bestimmen die Positioniersteuerprozessoren **14a1**, **14b1**, ob ein Startsignal eingeschaltet ist oder nicht (Schritt S53). Falls nicht eingeschaltet, setzen die Positioniersteuerprozessoren im Schritt S54 den Servoverstärker **34a**, **34b** zuzuführende Geschwindigkeitsbefehle auf "0". Falls eingeschaltet, lesen die Positioniersteuerprozessoren **14a1**, **14b1** im Schritt S52 die Zielpositionen **36a**, **36b** aus dem Kommunikationsspeicher **16**.

[0082] Dann subtrahieren die Positioniersteuerprozessoren **14a1**, **14b1** die derzeitigen Positionen von den Zielpositionen **36a**, **36b**, um Daten zu generieren, die an die Servoverstärker **34a**, **34b** auszugeben sind (Schritt S56). Dann begrenzen die Positioniersteuerprozessoren **14a1**, **14b1** die an die Servoverstärker **34a**, **34b** auszugebenden Daten (Schritt

S57). In einem Schritt 58 schreiben die Positioniersteuerprozessoren **14a1**, **14b1**, die begrenzten Daten als Geschwindigkeitsbefehle ein, die an die Servoverstärker **34a**, **34b** zu liefern sind, woraufhin die Verarbeitung zum Ende gelangt. Auf der Grundlage der Geschwindigkeitsbefehle erregen die Servoverstärker **34a**, **34b** die Servomotoren **30a**, **30b**, und deren derzeitige Stellungen werden von den jeweiligen Positionsdetektoren **32a**, **32b** erfaßt und den Positioniersteuerprozessoren **14a1**, **14b1** gemeldet. Die Daten über die derzeitige Position **40a**, **40b** werden aus dem Echtzeitprozessor **14** über den Kommunikationsspeicher **16** zu dem Multitask-Prozessor **12** übertragen.

[0083] Da von dem Multitask-Prozessor **12** (Seite der Hauptsteuerung) eine Verarbeitung für ein gleichzeitiges Ankommen durchgeführt wird, nicht jedoch von dem Echtzeitprozessor **14** (Regelseite), kann ein beim Hochfahren erzeugter Fehler nicht korrigiert werden, und es entsteht möglicherweise eine Stellungsabweichung. Erfindungsgemäß besitzen sowohl der Echtzeitprozessor **14** als auch der Multitask-Prozessor **12** derzeitige Positionsdaten, und die bis zu den jeweiligen Zielpositionen zurückzulegen den Strecken werden aus den Daten über die derzeitige Position berechnet, um eine derartige Stellungsabweichung zu vermeiden.

[0084] **Fig. 11** zeigt eine Verarbeitungsfolge für eine über den Steuerkasten **22** von Hand vorgenommene Positioniersteuerung. Wenn von dem Steuerkasten **22** über den Positioniertreiber **33** an die Servomotoren **30a**, **30b** Befehle zur Stellungssteuerung gegeben werden, wird festgestellt, ob die Bewegungsrichtung der Servomotoren **30a**, **30b** positiv ist oder nicht (Schritt S61), und ob die Bewegungsrichtung negativ ist oder nicht (Schritt S62). Ist die Bewegungsrichtung positiv, geht die Steuerung zu einem Schritt S67, bei dem die Zielpositionen **36a**, **36b** in dem Kommunikationsspeicher **16** auf einen positiven Maximalwert eingestellt werden. Anschließend wird in einem Schritt S68 ein Startsignal für den Kommunikationsspeicher **16** eingeschaltet.

[0085] Ist die Bewegungsrichtung negativ, geht die Steuerung zu einem Schritt S65, bei dem die Zielpositionen **36a**, **36b** in dem Kommunikationsspeicher **16** auf einen negativen Minimalwert eingestellt werden. Anschließend wird in einem Schritt S66 das Startsignal für den Kommunikationsspeicher **16** eingeschaltet. Wenn die Bewegungsrichtung weder positiv noch negativ ist, wird festgestellt, ob die Servomotoren **30a**, **30b** anzuhalten sind oder nicht (Schritt S63). Sind sie nicht anzuhalten, wird die Verarbeitung beendet. Wenn sie anzuhalten sind, wird das Startsignal für den Kommunikationsspeicher **16** in Schritt S64 ausgeschaltet. Anschließend kommt die Verarbeitung zum Ende.

[0086] **Fig. 12** zeigt einen Positionierablauf des Echtzeitprozessors **14**. Gemäß der Verarbeitung (siehe **Fig. 9**) der Anwendungstasks **42a**, **42b** des Multitask-Prozessors **12** bewirkt der Echtzeitprozes-

sor **14** eine Lagesteuerung des Motors für die erste Achse (Schritt S1), um anschließend eine Lagesteuerung für den Motor der zweiten Achse zu bewirken (Schritt S2) (siehe den Ablauf nach **Fig. 10**).

[0087] **Fig. 13** zeigt einen Schweißvorgang der Anwendungstasks **42a**, **42b**. Die Anwendungstasks **42a**, **42b** stellen im Schritt S71 fest, ob ein Schweißausführungs-Flag "0" ist oder nicht. Ist diesen Flag "0", lesen die Anwendungstasks **42a**, **42b** Druck- und Erregungspläne sowie Strombefehle entsprechend den über den Steuerkasten **22** oder über Zwischengerät-Verbindungen eingegebenen Schweißbedingungen aus dem Speicher aus (Schritt S72). Anschließend schreiben die Anwendungstasks **42a**, **42b** die gelesenen Daten in die Strombefehle **56a**, **56b** und die Druckbefehle **58a**, **58b** im Kommunikationsspeicher **16** ein (Schritt S73), woraufhin die Anwendungstasks **42a**, **42b** das Schweißausführungs-Flag im Schritt S74 auf "1" einstellen.

[0088] Die Pistolen-/Transformator-Steuerprozessoren **14a2**, **14b2** des Echtzeitprozessors **14** überträgt Strombefehle und PWM-Befehle zu den Wechselrichtern **54a**, **54b**, um die Transformatoren **42a**, **42b** und die Schweißpistolen **50a**, **50b** in die Lage zu versetzen, mit spezifischen Drücken gemäß den spezifizierten Erregungsplänen Werkstücke zu schweißen, wobei die Ströme derart gesteuert werden, daß sie bei der Aktivierung der Schweißpistolen **50a**, **50b** mit spezifischen Strömen konstant sind.

[0089] Wenn das Schweißausführungs-Flag nicht "0" ist, bestimmen die Anwendungstasks **42a**, **42b**, ob das Schweißausführungs-Flag "1" (Schritt S75). Ist das Schweißausführungs-Flag "1", bestimmen die Anwendungstasks **42a**, **42b**, ob der Schweißvorgang abgeschlossen ist oder nicht (Schritt S46). Falls abgeschlossen, löschen die Anwendungstasks **42a**, **42b** die Druck- und Erregungspläne und die Strombefehle, und sie setzen die Strombefehle **56a**, **56b** und die Druckbefehle **58a**, **58b** in dem Kommunikationsspeicher **16** zurück (Schritt S77). Anschließend setzen die Anwendungstasks **42a**, **42b** das Schweißausführungs-Flag in einem Schritt S78 auf "2", woraufhin der Verarbeitungsvorgang beendet wird.

[0090] Wenn das Schweißausführungs-Flag im Schritt S75 nicht "1" ist, wird die Verarbeitung beendet.

[0091] **Fig. 14** zeigt einen Schweißvorgang bei dem Echtzeitprozessor **14**. Gemäß der Verarbeitung der Anwendungstasks **42a**, **42b** des Multitask-Prozessors **12** (siehe **Fig. 12**) bewegt der Echtzeitprozessor **14** eine Steuerung des ersten Pistolen-/Transformator-Satzes in einem Schritt S3, um anschließend eine Steuerung des zweiten Pistolen-/Transformator-Satzes in einem Schritt S4 vorzunehmen.

[0092] **Fig. 13**

Schritt S 73: Lesen von Druck und Erregungsplänen und Strombefehlen entsprechend den Schweißbedingungen, die über Steuerbox oder über Zwischenverbindungen zwischen den Geräten angegeben werden

S 73: Einstellen der Druck- und Erregungspläne und Strombefehle im Kommunikationsspeicher
 S 74: Setzen des Schweißausführungs-Flags auf "1"
 S 77: Löschen der Druck- und Erregungspläne und der Strombefehle
 S 78: Setzen des Schweißausführungs-Flags auf "2"

Patentansprüche

1. Multitask-Steuersystem zum Steuern des Betriebs mehrerer gesteuerter Geräte (**20**; **34a, b**; **50a, b**), insbesondere Schweißroboter, Montageroboter und dergleichen, die über eine gemeinsame Kommunikationsleitung verbunden sind, umfassend:
 einen Multitask-Prozessor (**12**) zur parallelen Steuerung mehrerer Prozesse der jeweiligen gesteuerten Geräte auf Timesharing-Basis mit mehreren vorbestimmten Anwendungstasks (**42a, b**), die zugehörigen logischen Achsen jeweils eines der mehreren gesteuerten Geräte entsprechen und die von jedem der gesteuerten Geräte durchzuführen sind,
 einen Echtzeitprozessor (**14**) mit Steuerprozessoren (**14a1, 14b1**; **14a2, 14b2**) zum Durchführen einer Betriebs- und einer Positionierungssteuerung auf Echtzeitbasis für jede der logischen Achsen, und
 einen Kommunikationsspeicher (**16**) zum Senden und Empfangen vorbestimmter Befehle und Daten, die für die Betriebssteuerung und die Positionierungssteuerung zwischen dem Multitask-Prozessor (**12**) und dem Echtzeitprozessor (**14**) benötigt werden,
 wobei der Multitask-Prozessor und der Echtzeitprozessor als Teilkomponenten einer einzigen Multitask-steuervorrichtung (**10**) ausgebildet sind, welche baulich von den zu steuernden Geräten getrennt ist, und in der sowohl die Multitasking- als auch die Echtzeitverarbeitung durchgeführt wird, bevor entsprechende Befehle an die zusteuernenden Geräte gesendet werden,
 wobei Befehle und Daten vom Multitask-Prozessor einschließlich Information bezüglich der zu steuernden Geräte über Kommunikationspuffer (**24a1-24a4, 24b1-24b4, 24c1, 24c2**) gesendet und empfangen werden, um eine Zwischen-Task-Kommunikation zwischen den Tasks mittels einer Kommunikationstask (**46**) durchzuführen, und Befehle zwischen dem Echtzeitprozessor und den mehreren gesteuerten Geräten transferiert werden.

2. Steuersystem nach Anspruch 1, bei dem der Multitask-Prozessor (**12**) außerdem eine PLC-(Ablaufsteuerung-)Task (**44**) zum Übertragen von Information zu einer und zum Empfangen von Information von einer Ablaufsteuerung für die zentralisierte Steuerung der gesteuerten Geräte (**20**) aufweist.

3. Steuersystem nach Anspruch 1 oder 2, gekennzeichnet durch einen Steuerkasten (**22**) mit Anwendungstasks zur Kommunikation mit den Anwendungstasks (**42a, 42b**) des Multitask-Prozessors

(**12**), zum Anzeigen von Daten auf einem Anzeigebildschirm (**22i**) und zum Eingeben von Tastatureingabewerten, wobei die in dem Multitask-Prozessor (**12**) befindliche Kommunikations-Task (**46**) zur Kommunikation mit den Anwendungstasks (**22b-22g**) des Steuerkastens (**22**) dient.

4. Steuersystem nach einem der Ansprüche 1 bis 3, bei dem Zielpositionsdaten über den Kommunikationsspeicher (**16**) an den Echtzeitprozessor (**14**) geliefert und Istpositionsdaten von dem Echtzeitprozessor (**14**) über den Kommunikationsspeicher (**16**) empfangen werden, wobei die Positionier-Steuerprozessoren (**14a1, 14b1**) eine Strecke berechnen, die von einer Istposition zu einer Zielposition zurückzulegen ist, um auf der Grundlage der berechneten Strecke einen Geschwindigkeitsbefehl auszugeben.

5. Steuersystem nach jedem der Ansprüche 1 bis 4, bei dem Druck- und Erregungspläne und ein Strombefehl zum Schweißen des Werkstücks über den Kommunikationsspeicher (**16**) an den Echtzeitprozessor (**14**) geliefert werden, wobei der Echtzeitprozessor (**14**) Mittel zum Ausgeben eines Befehls zum Betreiben einer Schweißpistole (**50a, 50b**) auf der Grundlage der Druck- und Erregungspläne und des Strombefehls aufweist.

Es folgen 14 Blatt Zeichnungen

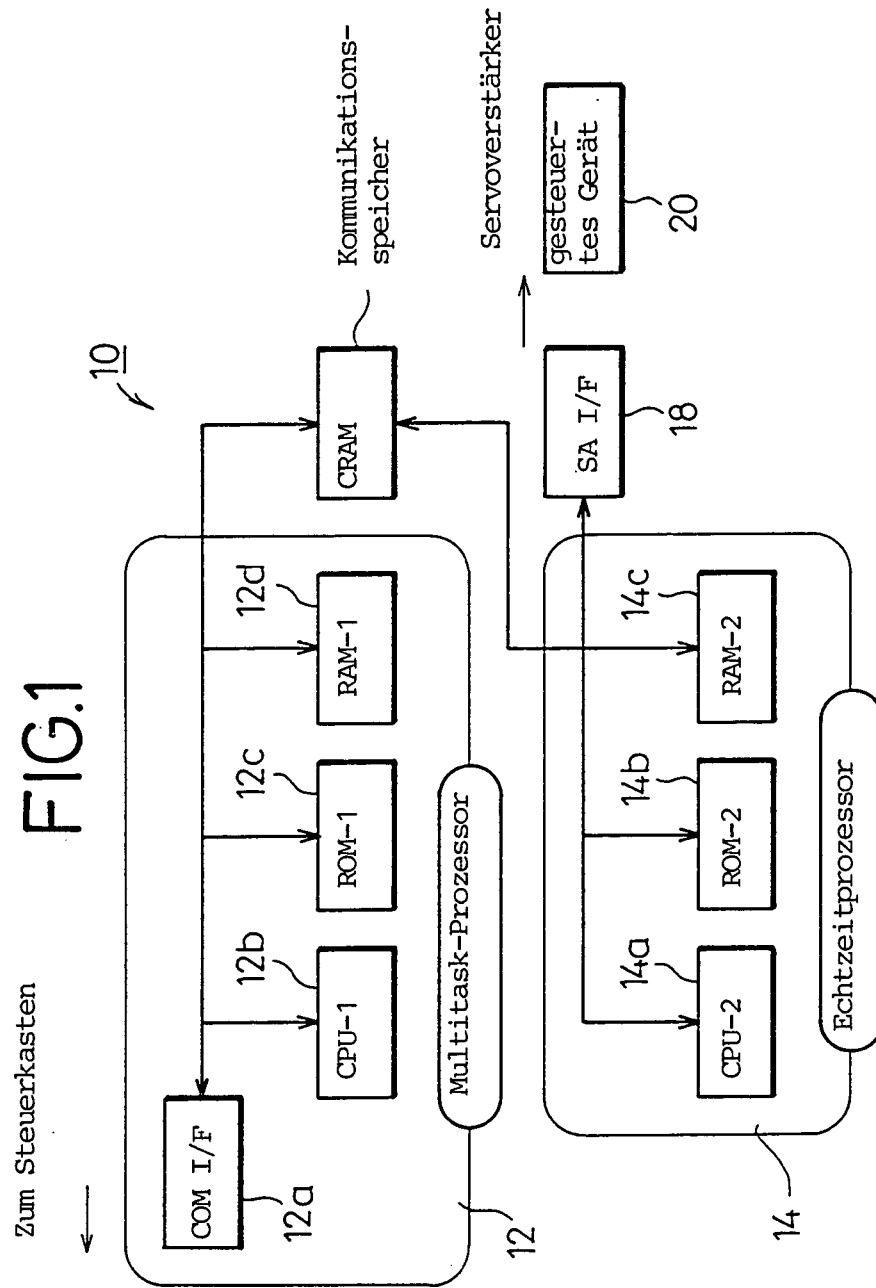
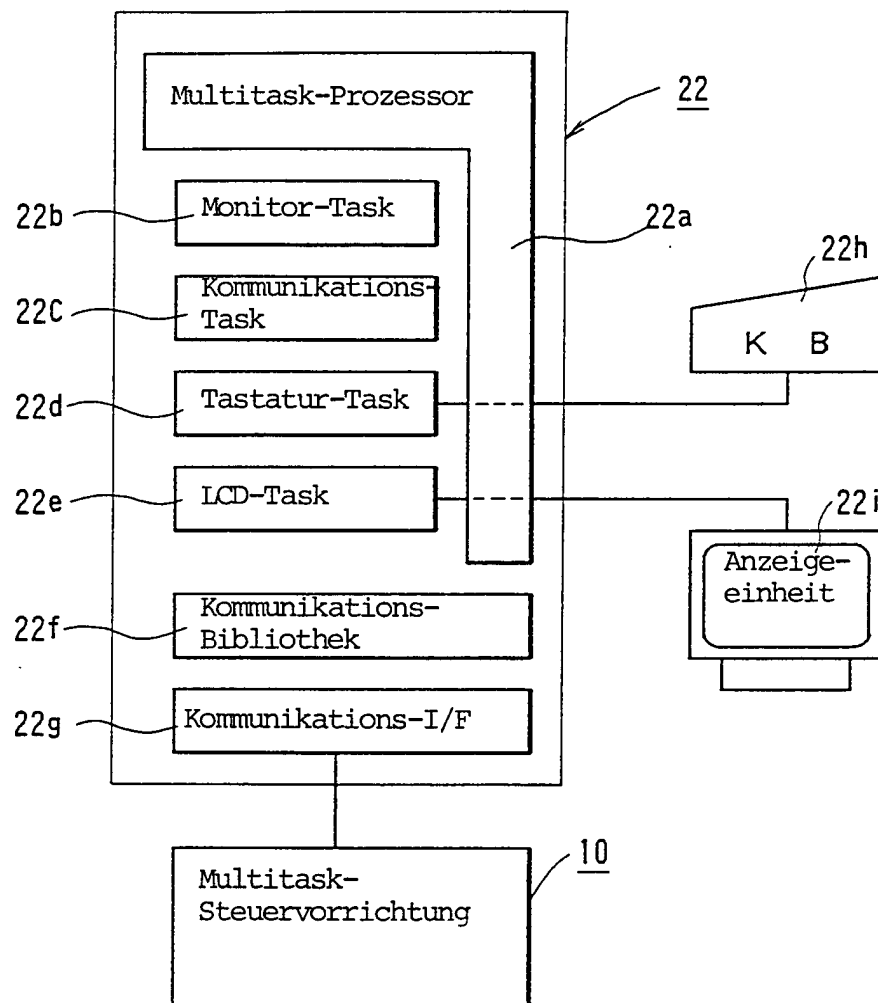
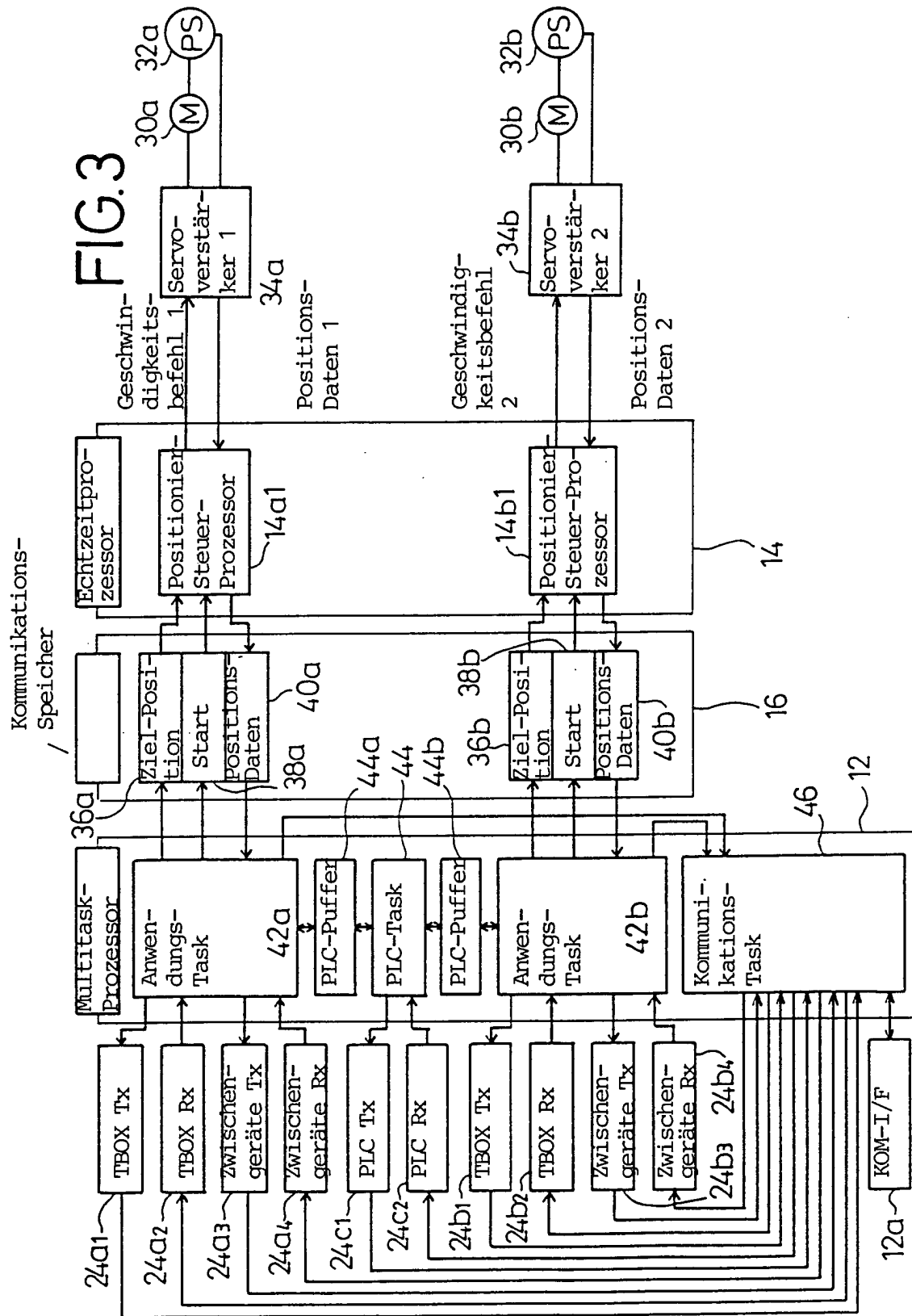


FIG.2





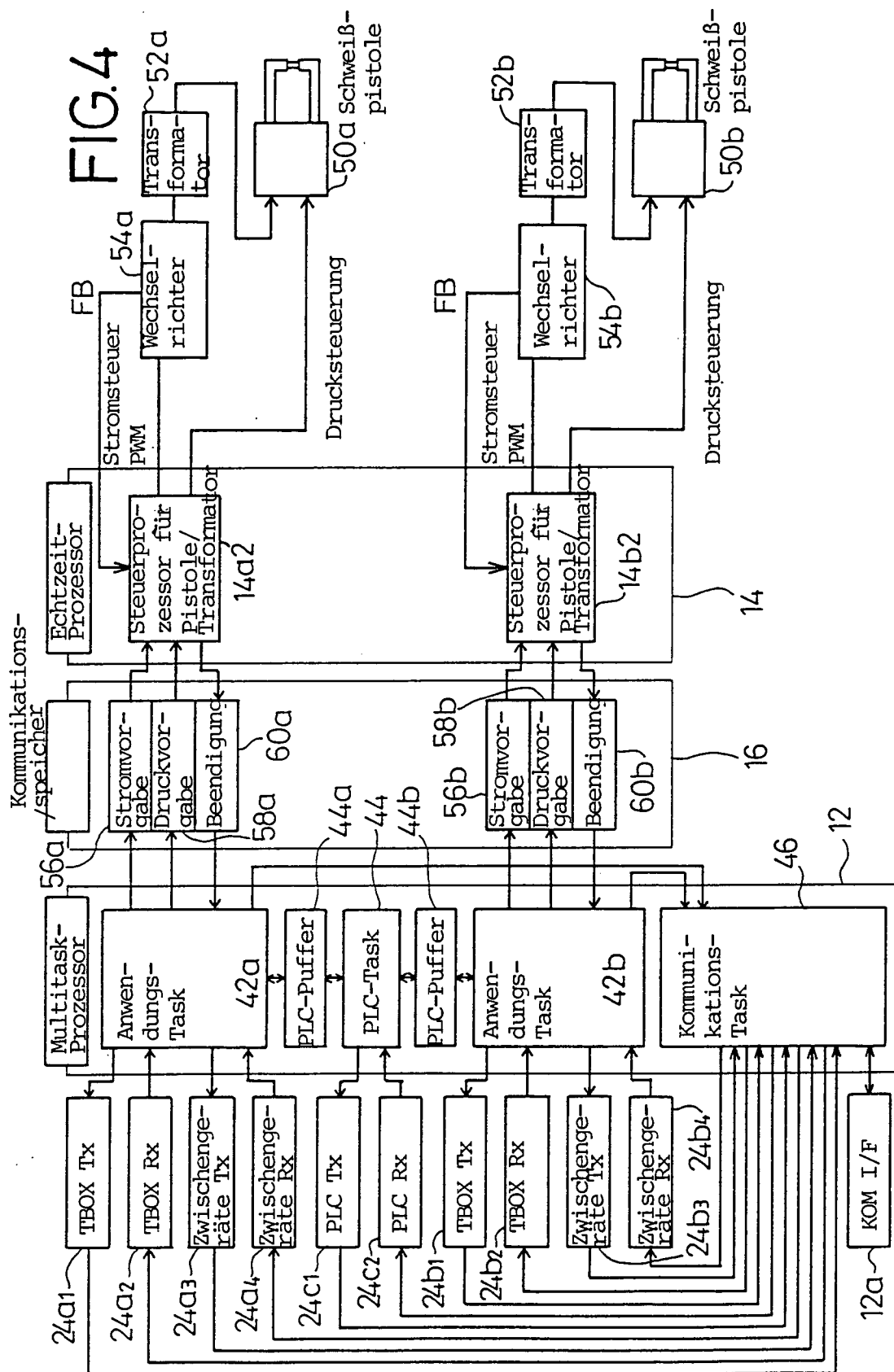


FIG.5

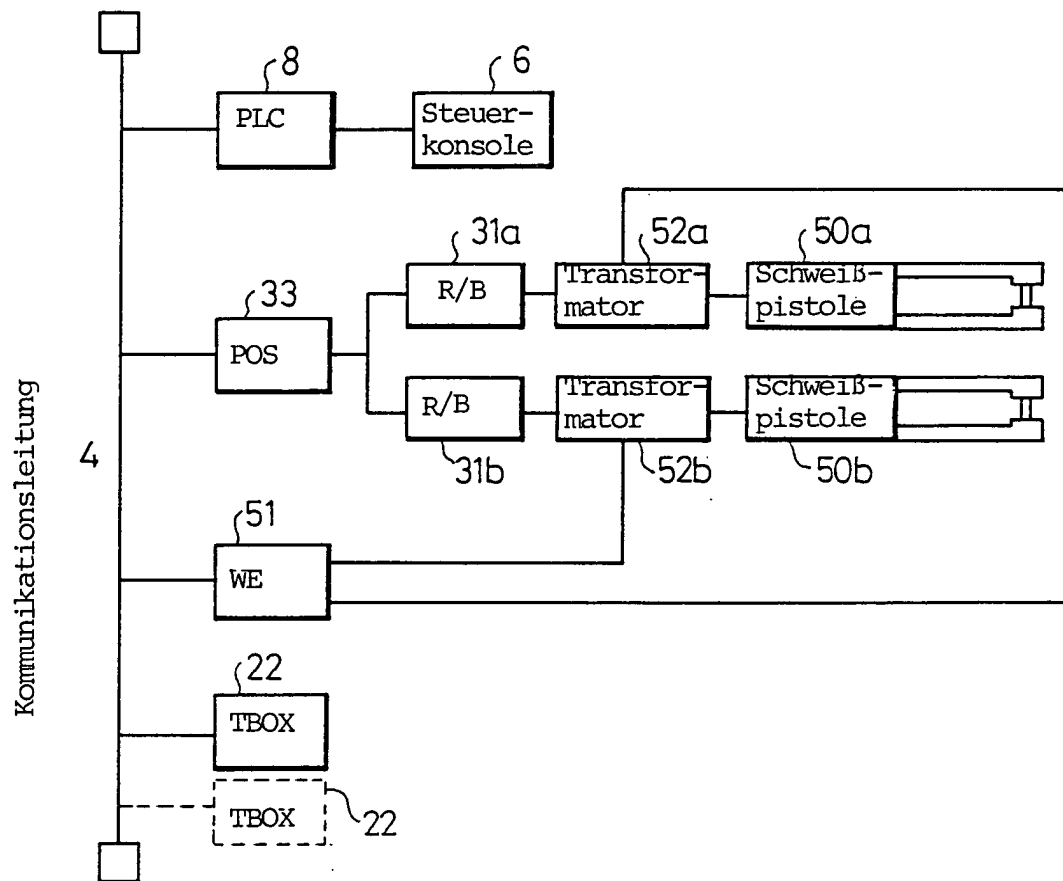


FIG. 6

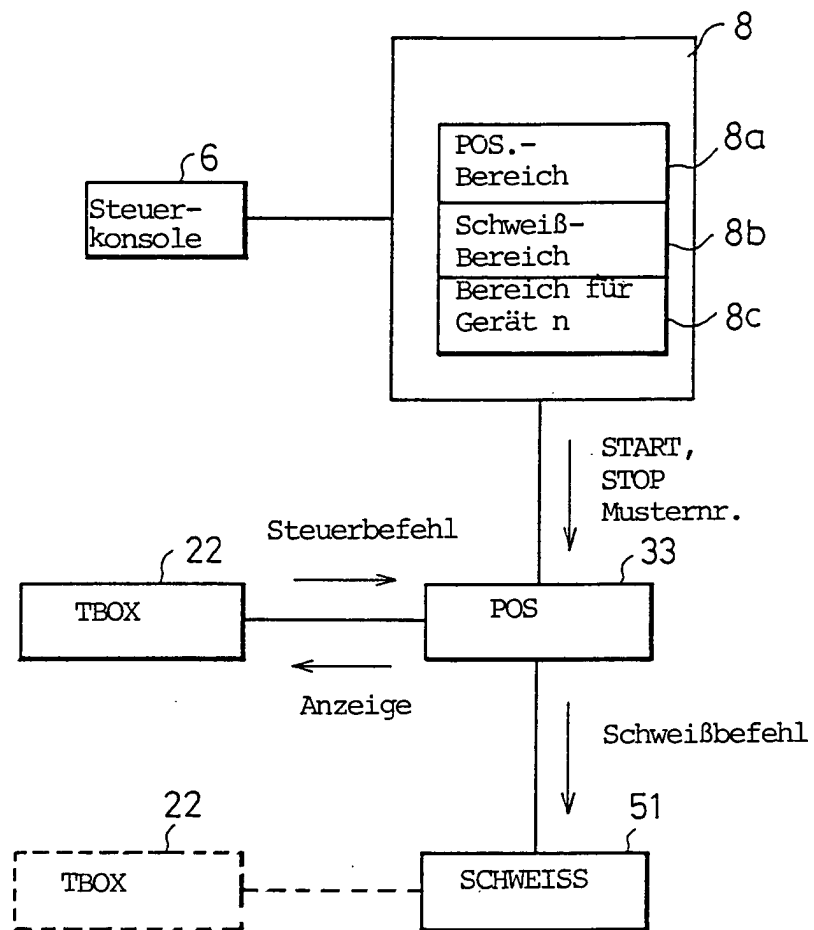


FIG.7

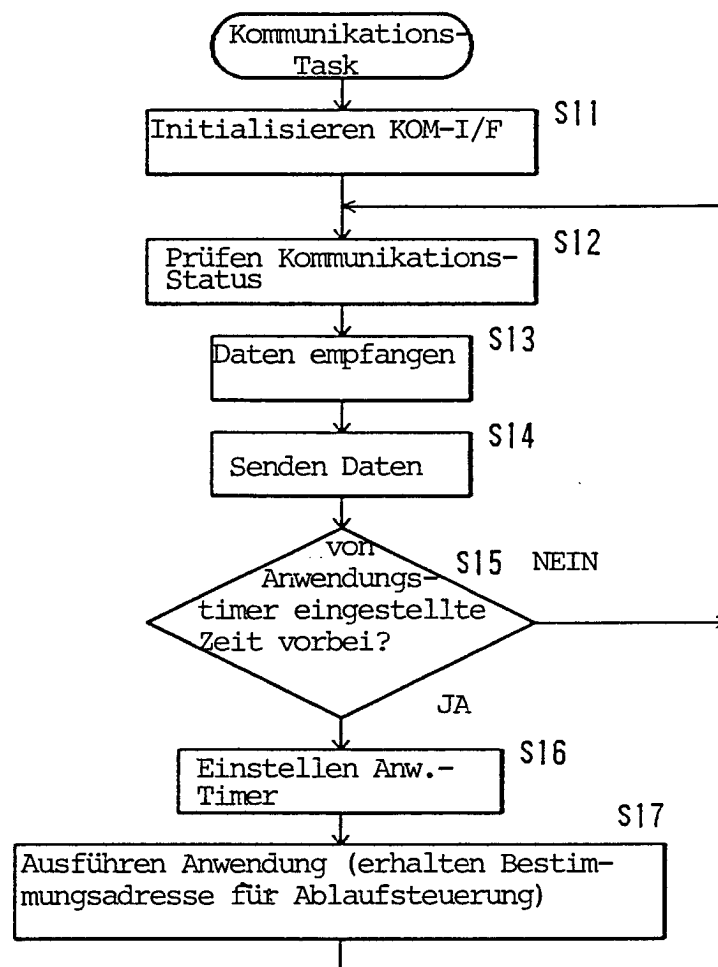


FIG.8

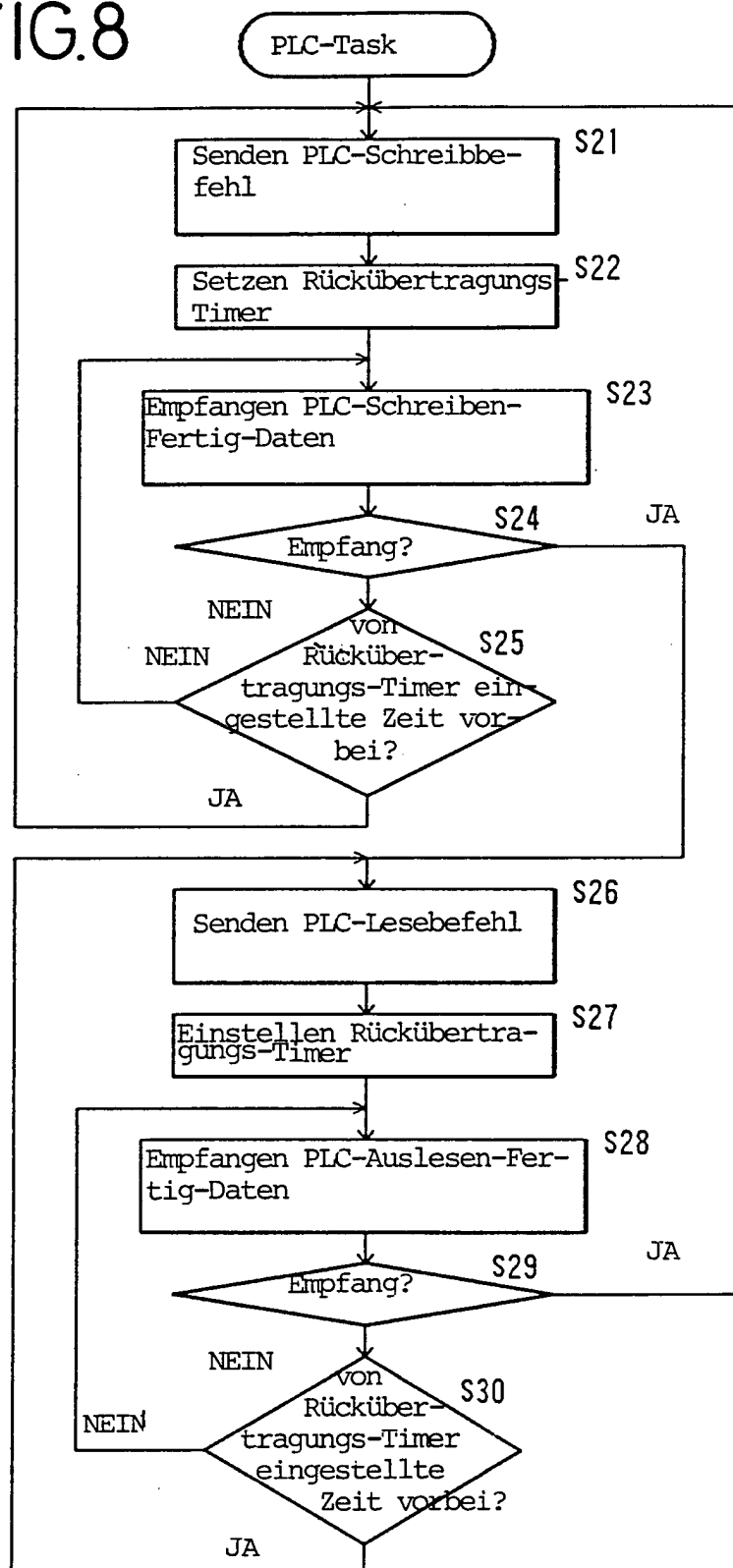


FIG.9

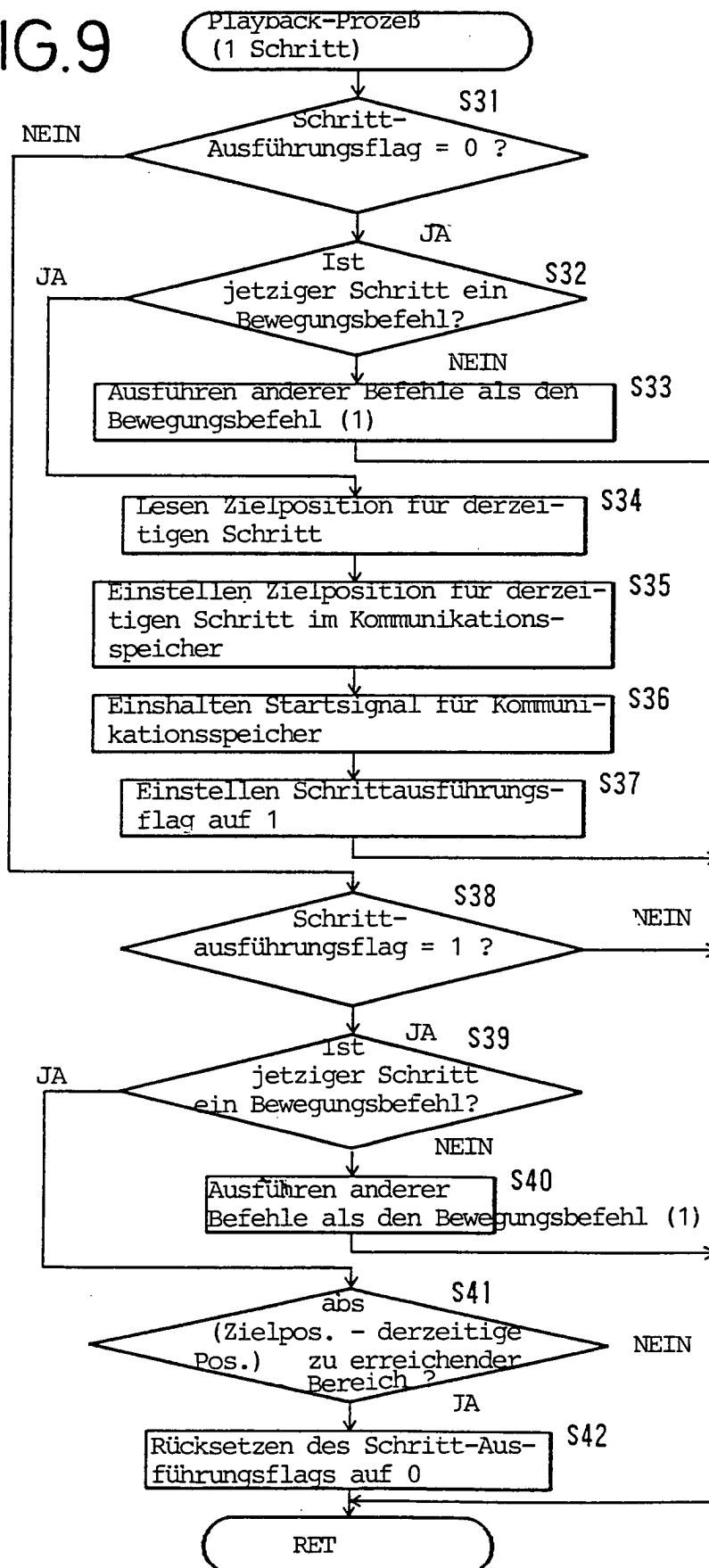


FIG.10

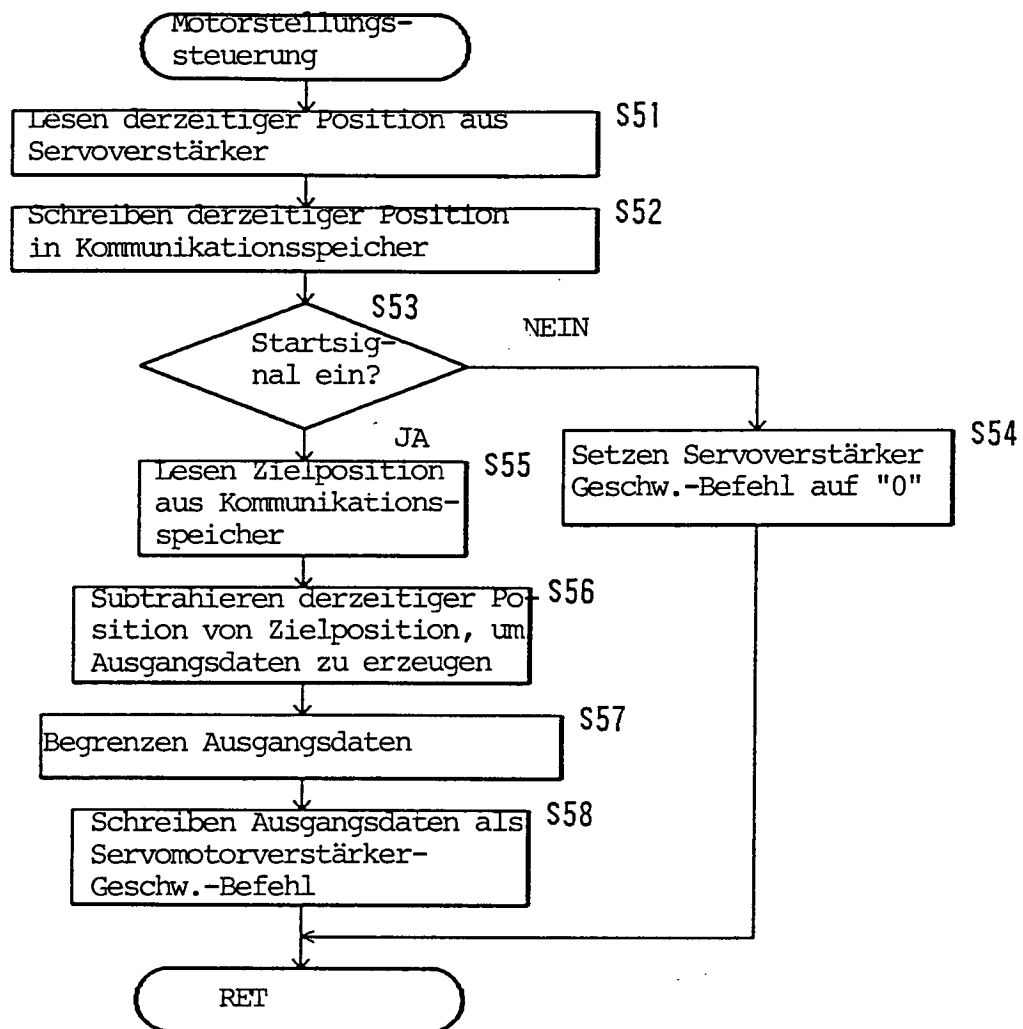


FIG.11

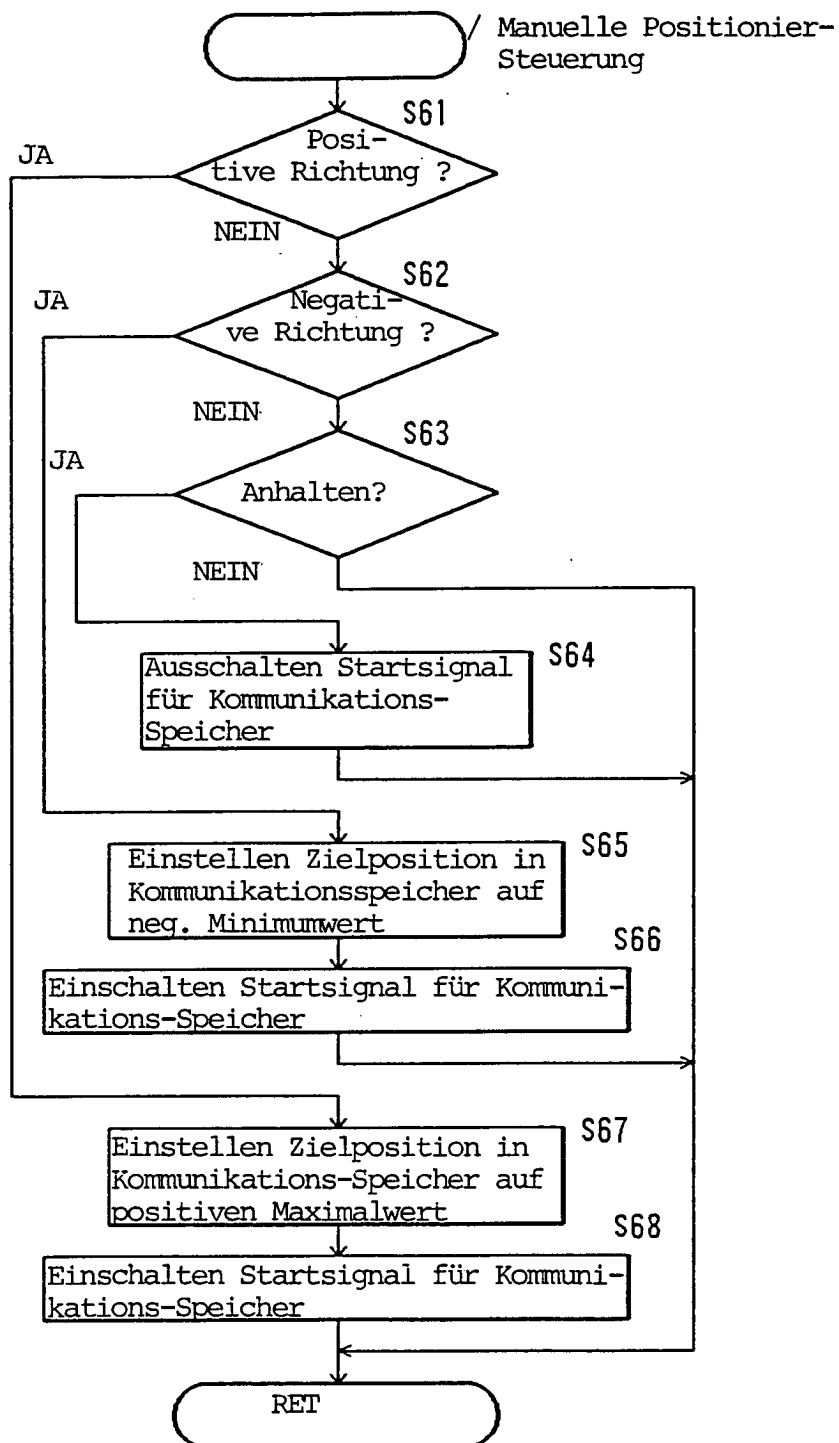


FIG.12

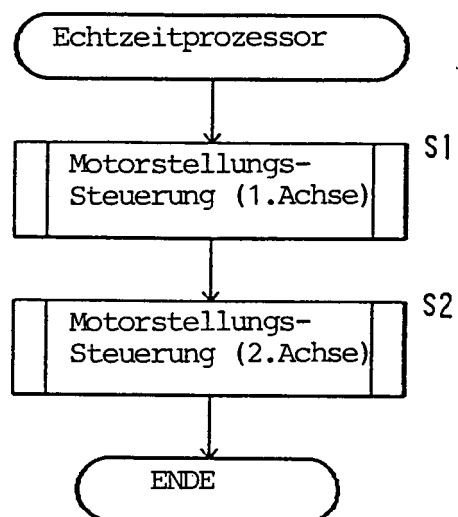


FIG.13

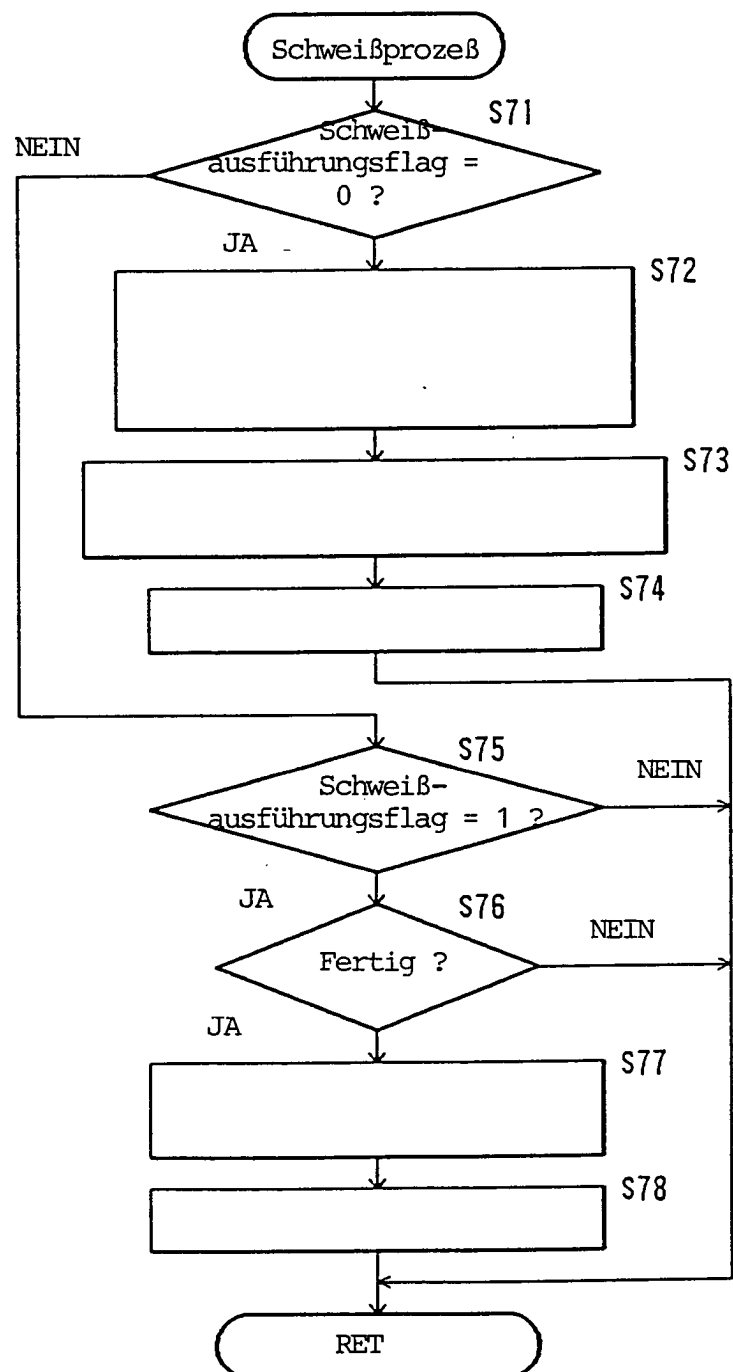


FIG.14

