(19) Deutsches Patent- und Markenamt

(10) **DE 11 2021 004 343 B4** 2025.07.10

(12) # Patentschrift

(21) Deutsches Aktenzeichen: **11 2021 004 343.9**
(86) PCT-Aktenzeichen: **PCT/JP2021/029930**
(87) PCT-Veröffentlichungs-Nr.: **WO 2022/039130**
(86) PCT-Anmeldetag: **16.08.2021**
(87) PCT-Veröffentlichungstag: **24.02.2022**
(43) Veröffentlichungstag der PCT Anmeldung in deutscher Übersetzung: **25.05.2023**
(45) Veröffentlichungstag der Patenterteilung: **10.07.2025**

(51) Int Cl.: ***B25J 9/22*** (2006.01)
B25J 13/00 (2006.01)
B25J 13/06 (2006.01)
B25J 9/16 (2006.01)

Innerhalb von neun Monaten nach Veröffentlichung der Patenterteilung kann nach § 59 Patentgesetz gegen das Patent Einspruch erhoben werden. Der Einspruch ist schriftlich zu erklären und zu begründen. Innerhalb der Einspruchsfrist ist eine Einspruchsgebühr in Höhe von 200 Euro zu entrichten (§ 6 Patentkostengesetz in Verbindung mit der Anlage zu § 2 Abs. 1 Patentkostengesetz).

(30) Unionspriorität:
2020-139400 **20.08.2020** **JP**

(73) Patentinhaber:
FANUC CORPORATION, Oshino-mura, Yamanashi, JP

(74) Vertreter:
Meissner Bolte Patentanwälte Rechtsanwälte Partnerschaft mbB, 80538 München, DE

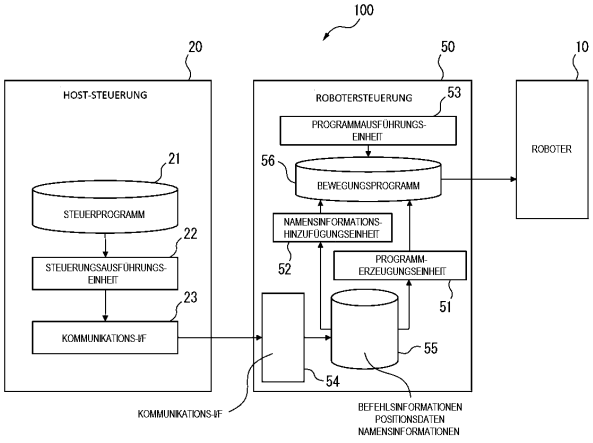
(72) Erfinder:
Katou, Tomoki, Oshino-mura, Yamanashi, JP

(56) Ermittelter Stand der Technik:

DE	**10 2016 012 779**	**B4**
DE	**10 2015 004 932**	**A1**
DE	**10 2015 226 696**	**A1**
DE	**10 2016 002 340**	**A1**
DE	**60 2004 009 264**	**T2**
US	**2018 / 0 276 501**	**A1**
WO	**2019 / 180 916**	**A1**

(54) Bezeichnung: **ROBOTERSTEUERSYSTEM**

(57) Hauptanspruch: Robotersteuersystem (100), das eine Host-Steuerung (20) und eine mit der Host-Steuerung (20) verbundene Robotersteuerung (50) umfasst,
die Host-Steuerung (20) umfassend
eine Steuerungsausführungseinheit (22), die an die Robotersteuerung (50) Befehlsinformationen, die einen Bewegungsbefehl an einen Roboter (10) darstellen, und Positionsdaten, die mit dem Bewegungsbefehl verknüpft sind, wie auch Namensinformationen, die einen Namen der Positionsdaten darstellen, auf Basis eines Steuerprogramms zum Steuern des Roboters (10) überträgt, und
die Robotersteuerung (50) umfassend:
eine Programmerzeugungseinheit (51), die ein Bewegungsprogramm für den Roboter (10) auf Basis der empfangenen Befehlsinformationen und der empfangenen Positionsdaten erzeugt; und
eine Namensinformationshinzufügungseinheit (52), die den Namen, der von den empfangenen Namensinformationen dargestellt wird, zu den Positionsdaten, die in dem Bewegungsprogramm beinhaltet sind, hinzufügt.



Beschreibung

[0001] Die vorliegende Erfindung bezieht sich auf ein Robotersteuersystem, das einen Roboter steuert.

[0002] Ein Industrieroboter wird im Allgemeinen von einer Robotersteuerung gesteuert, um eine Bewegung auszuführen, indem ein Bewegungsprogramm, das unter Verwendung eines Teach Pendant oder eines Programmiergeräts erstellt ist, in die Robotersteuerung geladen wird. In den letzten Jahren wird jedoch auch ein System vorherrschend, das einen Industrieroboter durch eine PLC (programmierbare Logiksteuereinheit, „Programmable Logic Controller“) steuert (siehe z.B. WO 2019/180916 A1). Weitere Systeme sind beispielsweise aus DE 10 2016 012 779 B4, DE 10 2016 002 340 A1, DE 10 2015 226 696 A1, DE 10 2015 004 932 A1, DE 60 2004 009 264 T2 und US 2018 / 0 276 501 A1 bekannt.

[0003] Heutzutage, mit der weitreichenden Verwendung von z.B. der PLCopen (eingetragene Marke) Norm, schreitet auch eine Verbesserung der Entwicklungsumgebung eines PLC-Programms zum Steuern eines Industrieprogramms voran. Ein Robotersteuersystem unter Verwendung einer PLC führt im Allgemeinen ein PLC-Programm derart aus, dass ein Bewegungsbefehl für einen Roboter von der PLCC an eine Robotersteuerung übertragen wird und die Robotersteuerung verwendet, um auf Basis des Bewegungsbefehls, der von der PLC ausgestellt wurde, ein Bewegungsprogramm zu erzeugen und den Roboter zu steuern. In solch einem Robotersteuersystem, angenommen, dass Anlerneinstellungen eines Roboters unter Verwendung eines PLC-Programms und Bewegungsbestätigung in PLC-Programmentwicklung durchgeführt werden, ist es für den Nutzer schwierig, den Zusammenhang zwischen dem PLC-Programm und dem Bewegungsprogramm, die von der Robotersteuerung erzeugt werden, zu verstehen, da das PLC-Programm und das Bewegungsprogramm, die von der Robotersteuerung erzeugt werden, unterschiedliche Programmspezifikationen aufweisen, und daher dauert z.B. eine Prüfung des Auftretens eines Fehlers länger.

[0004] Es ist daher die Aufgabe der vorliegenden Erfindung, ein dahingehend verbessertes Robotersteuersystem anzugeben.

[0005] Diese Aufgabe wird erfindungsgemäß durch ein Robotersteuersystem gemäß Anspruch 1 gelöst.

[0006] Ein Aspekt der vorliegenden Offenbarung stellt ein Robotersteuersystem bereit, das eine Host-Steuerung und eine mit der Host-Steuerung verbundene Robotersteuerung beinhaltet, wobei die Host-Steuerung eine Steuerungsausführungseinheit beinhaltet, die an die Robotersteuerung Befehlsinformationen, die einen Bewegungsbefehl an einen Roboter darstellen, und Positionsdaten, die mit dem Bewegungsbefehl verknüpft sind, wie auch Namensinformationen, die einen Namen der Positionsdaten darstellen, auf Basis eines Steuerprogramms zum Steuern des Roboters überträgt, und die Robotersteuerung eine Programmerzeugungseinheit, die ein Bewegungsprogramm für den Roboter auf Basis der empfangenen Befehlsinformationen und der empfangenen Positionsdaten erzeugt, und eine Namensinformationshinzufügungseinheit, die den Namen, der von den empfangenen Namensinformationen dargestellt wird, zu den Positionsdaten hinzufügt, die in dem Bewegungsprogramm beinhaltet sind, beinhaltet.

[0007] Mit der zuvor erwähnten Konfiguration kann der Nutzer leicht verstehen, wie das Steuerprogramm, das von der Host-Steuerung erzeugt wird, und das Bewegungsprogramm, das von der Robotersteuerung erzeugt wird, miteinander zusammenhängen. Das macht es möglich, eine Fehlerprüfung und ein Debug bei Anlerneinstellung und Programmierung effizient durchzuführen.

[0008] Diese und andere Ziele, Merkmale und Vorteile der vorliegenden Erfindung werden aus der folgenden ausführlichen Beschreibung einer beispielhaften Ausführungsform davon, wie sie in den angehängten Zeichnungen veranschaulicht ist, ersichtlich.

Fig. 1 ist ein Blockdiagramm, das eine Konfiguration eines gesamten Robotersteuersystems gemäß einer Ausführungsform veranschaulicht.

Fig. 2 ist ein Diagramm, das ein beispielhaftes PLC-Programm veranschaulicht, wenn eine PLC als eine Host-Steuerung verwendet wird.

Fig. 3 ist ein Diagramm, das ein Bewegungsprogramm veranschaulicht, das von einer Robotersteuerung in der in **Fig.** 2 veranschaulichten Konfiguration erzeugt wird.

Fig. 4 ist ein Diagramm, das ein beispielhaftes CNC-Programm veranschaulicht, wenn eine CNC als die Host-Steuerung verwendet wird.

Fig. 5 ist ein Diagramm, das ein Bewegungsprogramm veranschaulicht, das von der Robotersteuerung in der in **Fig.** 4 veranschaulichten Konfiguration erzeugt wird.

[0009] Eine Ausführungsform der vorliegenden Offenbarung wird unten in Bezug auf die Zeichnungen beschrieben. In den Zeichnungen, auf die Bezug genommen wird, bezeichnen dieselben oder ähnliche Bezugszeichen dieselben oder ähnliche Komponenten oder funktionale Teile. Um das Verständnis zu erleichtern, verwenden diese Zeichnungen unterschiedliche Skalen, wie sie geeignet sind. Weiter

sind die in den Zeichnungen veranschaulichten Modi bloße Beispiele zum Umsetzen der vorliegenden Erfindung, die nicht auf die in den Zeichnungen veranschaulichten Modi begrenzt ist.

[0010] **Fig.** 1 ist ein Blockdiagramm, das eine Konfiguration eines gesamten Robotersteuersystems 100 gemäß einer Ausführungsform veranschaulicht. Das Robotersteuersystem 100 beinhaltet eine Host-Steuerung 20, eine Robotersteuerung 50 und einen Roboter 10 und ist derart konfiguriert, dass die Robotersteuerung 50, die den Roboter 10 steuert, und die Host-Steuerung 20 kommunizierbar miteinander verbunden sind. Das Robotersteuersystem 100 kann ein Steuerprogramm für den Roboter 10 auf der Host-Steuerung 20 erstellen. Die Host-Steuerung 20 überträgt, an die Robotersteuerung 50, einen Bewegungsbefehl für den Roboter 10 in Übereinstimmung mit dem Steuerprogramm für den Roboter 10, das auf der Host-Steuerung 20 eingerichtet ist. Die Robotersteuerung 50 erzeugt ein Bewegungsprogramm für den Roboter 10 auf Basis des Bewegungsbefehls, der von der Host-Steuerung 20 übertragen wird, und veranlasst den Roboter 10, eine Bewegung auszuführen.

[0011] Die Hoststeuereinheit 20 kann die Konfiguration eines allgemeinen Computers aufweisen, beinhaltend z.B. eine CPU, einen ROM, einen RAM, eine Datenspeichervorrichtung, eine Operationseinheit, eine Anzeigeeinheit, eine I/O-Schnittstelle und eine Netzwerkschnittstelle. Die Host-Steuerung 20 ist z.B. als eine PLC (programmierbare Logiksteuereinheit) oder eine CNC (computerisierte numerische Steuereinheit, „Computerized Numerical Controller) implementiert. Als die Host-Steuerung 20 können nicht nur eine PLC oder eine CNC, sondern auch verschiedene Informationsverarbeitungsvorrichtungen, wie ein PC, verwendet werden. Die Robotersteuerung 50 kann die Konfiguration eines allgemeinen Computers aufweisen, beinhaltend z.B. eine CPU, einen ROM, einen RAM, eine Datenspeichervorrichtung, eine Operationseinheit, eine Anzeigeeinheit, eine I/O-Schnittstelle und eine Netzwerkschnittstelle. Um die Host-Steuerung 20 und die Robotersteuerung 50 miteinander zu verbinden, können ein Feldbus, ein kabelgebundenes oder drahtloses LAN oder verschiedene andere Netzwerke verwendet werden.

[0012] Die Host-Steuerung 20 hält ein Steuerprogramm 21 in der Datenspeichervorrichtung zum Steuern des Roboters 10, das in einer Softwareumgebung eingerichtet ist, die auf der Host-Steuerung 20 läuft, wie in **Fig.** 1 veranschaulicht wird. Die Host-Steuerung 20 beinhaltet weiter eine Steuerausführungseinheit 22 und eine Kommunikationsschnittstelle 23. Die Steuerausführungseinheit 22 interpretiert das Steuerprogramm 21 und überträgt über die Kommunikationsschnittstelle 23 Anweisungsinformationen an die Robotersteuerung 50, die einen Bewegungsbefehl für den Roboter 10 darstellen. Wenn der Befehl in dem Steuerprogramm 21 als ein Bewegungsbefehl implementiert ist, der mit Positionsdaten verknüpft ist, überträgt die Steuerungsausführungseinheit 22 Befehlsinformationen, die einen Bewegungsbefehl für den Roboter 10 darstellen, und Positionsdaten, die mit dem Bewegungsbefehl verknüpft sind, wie auch Namensinformationen, die den Namen der Positionsdaten darstellen, an die Robotersteuerung 50.

[0013] Die Robotersteuerung 50 beinhaltet eine Programmerzeugungseinheit 51, eine Namensinformationshinzufügungseinheit 52, eine Programmausführungseinheit 53 und eine Kommunikationsschnittstelle 54. In der Robotersteuerung 50 erzeugt die Programmerzeugungseinheit 51 einen Bewegungsbefehl an den Roboter 10 auf Basis der Befehlsinformationen, die von der Host-Steuerung 20 übertragen werden, und erstellt den Bewegungsbefehl als ein Bewegungsprogramm 56. In diesem Fall, wenn Positionsdaten, die mit dem Bewegungsbefehl verknüpft sind, und Namensinformationen, die mit den Positionsdaten verknüpft sind, übertragen worden sind, fügt die Namensinformationshinzufügungseinheit 52 den Namen der Positionsdaten zu den Positionsdaten als z.B. einen Kommentartext hinzu. Die Programmausführungseinheit 53 veranlasst den Roboter 10, eine Bewegung auszuführen, indem das Bewegungsprogramm 56 ausgeführt wird, das von der Programmerzeugungseinheit 51 erzeugt wurde.

[0014] Spezifische Konfigurationen für zwei Beispiele, in denen die Host-Steuerung 20 als eine PLC beziehungsweise eine CNC implementiert ist, werden unten beschrieben.

[0015] Ein Konfigurationsbeispiel, wenn die Host-Steuerung 20 als eine PLC 20A implementiert ist, wird unten in Bezug auf **Fig.** 2 und 3 beschrieben. **Fig.** 2 veranschaulicht ein beispielhaftes Steuerprogramm, das auf der PLC 20A (das auf der PLC eingerichtete Steuerprogramm wird hierin nachfolgend auch als ein PLC-Programm bezeichnet) erstellt ist. Das Steuerprogramm auf der PLC kann in Kontaktplansprache oder strukturierter Textsprache erstellt sein. Ein in strukturierter Textsprache erstelltes Steuerprogramm wird hier als ein Beispiel herangezogen. Das in **Fig.** 2 veranschaulichte PLC-Programm beinhaltet eine Variablendefinition 24, einen PLC-Programmkörper (der hierin nachfolgend einfach als ein PLC-Programm 21A bezeichnet wird) und zwei Funktionsblöcke (FBs) 25a und 25b.

[0016] Die Variablendefinition 24 beinhaltet zwei Strukturvariablen HOME.POSITION und PICK.POOSITION, die jeweils aus drei Variablen, X-, Y- und Z-Positionen, gebildet sind. Die Werte dieser Variablen sind jeweils auf HOME.POSITION.X = 0, HOME.POSITION.Y = 0, und HOME.POSITION.Z =

0, und PICK.POSITION.X = 30, PICK.POSITION.Y = 20 und PICK.POSITION.Z = 15 festgelegt.

[0017] Das PLC-Programm 21A beinhaltet eine MoveLinear-Funktion, die einen Befehl beschreibt, den Roboter dazu zu veranlassen, eine lineare Bewegung auszuführen, und eine MoveAxes-Funktion, die einen Befehl beschreibt, den Roboter zu veranlassen, eine angelenkte Bewegung auszuführen. Die von der MoveLinear-Funktion und der MoveAxes-Funktion beschriebenen Bewegungen sind in den Funktionsblöcken (FBs) 25a beziehungsweise 25b definiert. Die MoveLinear-Funktion wird verwendet, um eine Struktur HOME.POSITION von Positionsdaten des Roboters zu einem Argument DestPos einer Zielposition einzugeben. Mit dieser Operation werden X = 0, Y = 0 und Z = 0 als Zielpositionsdaten eines MoveLinear-Befehls (lineare Bewegung) für den Roboter 10 kommandiert. In dem Funktionsblock 25a der MoveLinear-Funktion ist ein numerischer Wert von 1 definiert, um als eine Kommando-ID entsprechend der MoveLinear-Funktion festgelegt zu werden. In diesem Fall interpretiert die Steuerungsausführungseinheit 22 den Funktionsblock 25a, überträgt „Kommando-ID = 1“ an die Robotersteuerung 50 als Befehlsinformationen, die dem Bewegungsbefehl MoveLinear entsprechen, und überträgt weiter die Positionsdaten (DestPos) an die Robotersteuerung 50.

[0018] Die Steuerungsausführungseinheit 22 beinhaltet ‚HOME.POSITION‘ in den zu übertragenden Daten als den Variablennamen der Positionsdaten, die mit dem Bewegungsbefehl MoveLinear verknüpft sind. Mit anderen Worten, in diesem Fall überträgt die Steuerungsausführungseinheit 22 die folgenden Daten an die Robotersteuerung:

.Kommando-ID = 1

.Positionsdaten(X, Y, Z) = (0, 0, 0) ‚HOME.POSITION‘

[0019] Die MoveAxes-Funktion wird verwendet, um eine Struktur PICK.POSITION von Positionsdaten des Roboters zu einem Argument DestPos einer Zielposition einzugeben. Mit dieser Operation werden X = 30, Y = 20 und Z = 15 als Zielpositionsdaten eines MoveAxes-Befehls (lineare Bewegung) für den Roboter 10 kommandiert. In dem Funktionsblock 25b der MoveAxes-Funktion ist ein numerischer Wert von 2 definiert, als eine Kommando-ID entsprechend der MoveAxes-Funktion festgelegt zu werden. In diesem Fall interpretiert die Steuerungsausführungseinheit 22 den Inhalt des Funktionsblocks 25b, überträgt „Kommando-ID = 2“ an die Robotersteuerung 50 als Befehlsinformationen, die dem Bewegungsbefehl MoveAxes entsprechen, und überträgt weiter die Positionsdaten (DestPos) an die Robotersteuerung 50.

[0020] Die Steuerungsausführungseinheit 22 beinhaltet ‚PICK.POSITION‘ in den zu übertragenden Daten als den Variablennamen der Positionsdaten, die mit dem Bewegungsbefehl MoveAxes verknüpft sind. Mit anderen Worten, in diesem Fall überträgt die Steuerungsausführungseinheit 22 die folgenden Daten an die Robotersteuerung 50:

•Kommando-ID = 2

•Positionsdaten (X, Y, Z) = (30, 20, 15) •'PICK.POSITION'

[0021] Die von der PLC 20 übertragenen Daten werden z.B. in einem primären Datenspeicherbereich 55 innerhalb des RAM der Robotersteuerung 50 gespeichert. Die Programmerzeugungseinheit 51 erzeugt ein Bewegungsprogramm für den Roboter 10 auf Basis der von der PLC 20A übertragenen Daten. Als ein Beispiel kann die Robotersteuerung 50 eine Tabelle besitzen, die Kommando-IDs und Bewegungsbefehle miteinander verknüpft, wie in der folgenden Tabelle 1 dargestellt wird. In diesem Fall kann die Programmerzeugungseinheit 51 einen Bewegungsbefehl, entsprechend der Kommando-ID, die von der PLC 20 übertragen wird, erkennen, indem sie in dieser Tabelle nachschlägt. In diesem Fall besitzt die PLC 20A auch eine Tabelle ähnlich Tabelle 1. In der in der folgenden Tabelle 1 dargestellten Tabelle sind eine lineare Bewegung, eine angelenkte Bewegung, eine Pause und ein Bewegungsneustart mit Kommando-IDs = 1, 2, 3 beziehungsweise 4 verknüpft.

[Tabelle 1]

Kommando-ID	Kommandodetails
1	Lineare Bewegung
2	Angelenkte Bewegung
3	Pause
4	Bewegungsneustart

[0022] Es wird angenommen, dass die von der PLC 20A übertragenen Daten eine Kommando-ID = 1 angeben. In diesem Fall erkennt die Programmerzeugungseinheit 51 anhand der Kommando-ID = 1, dass der Bewegungsbefehl eine lineare Bewegung angibt. Die Programmerzeugungseinheit 51 speichert Positionsdaten (0, 0, 0), die gemeinsam mit der Kommando-ID = 1 übertragen werden, in einem Positionsregister P[1], erzeugt einen Befehl ‚L P[1]‘ für eine lineare Bewegung des Roboters 10 und setzt einen Text HOME.POSITION, der den Namen der Positionsdaten (0, 0, 0) darstellt, der gemeinsam mit diesen Positionsdaten an das Positionsregister P [1] übertragen wird, als einen Kommentartext ein. Mit dieser Operation erzeugt die Programmerzeugungseinheit 51

‚L P[1:HOME.POSITION]‘

als eine Anweisung einer linearen Bewegung.

[0023] Die Programmausführungseinheit 53 ist konfiguriert, jedes Mal wenn eine neue Anweisung in einem Bewegungsprogramm 56A als ein Beispiel erzeugt wird, einen Bewegungsbefehl auszuführen, und wenn die zuvor erwähnte ‚L P[1:HOME.POSITION]' in das Bewegungsprogramm 56A eingesetzt wird, bewegt die Programmausführungseinheit 53 einen gesteuerten Abschnitt, der in einem beweglichen Abschnitt des Roboters 10 liegt, linear zu der Zielposition (Home-Position), die in dem Positionsregister P[1] festgelegt ist.

[0024] Wenn eine Kommando-ID = 2 weiter von der PLC 20A übertragen wird, erkennt die Programmerzeugungseinheit 51, anhand der Kommando-ID = 2, dass der Bewegungsbefehl eine angelenkte Bewegung angibt. Die Programmerzeugungseinheit 51 speichert Positionsdaten (30, 20, 15), die gemeinsam mit der Kommando-ID = 2 übertragen werden, in einem Positionsregister P[2], erzeugt einen Befehl ‚J P[2]' für eine angelenkte Bewegung des Roboters 10 und setzt einen Text PICK.POSITION, der den Namen der Positionsdaten (30, 20, 15) darstellt, der gemeinsam mit diesen Positionsdaten an das Positionsregister P[2] übertragen wird, als einen Kommentartext ein. Mit dieser Operation erzeugt die Programmerzeugungseinheit 51

‚J P[2:PICK.POSITION]'

als eine Anweisung einer angelenkten Bewegung.

[0025] Wenn der zuvor erwähnte Befehl ‚J P[2: PICK.POSITION]' in das Bewegungsprogramm 56A eingesetzt wird, bewegt die Programmausführungseinheit 53 den gesteuerten Abschnitt des Roboters 10 zu der Zielposition (Pick-Position), die in dem Positionsregister P[2] festgelegt ist.

[0026] Auf diese Weise wird ein Variablenname, der Positionsdaten in dem PLC-Programm darstellt, in das Bewegungsprogramm, das von der Robotersteuerung 50 erzeugt ist, als ein Kommentartext eingesetzt, der die Positionsdaten darstellt. Mit anderen Worten, Textinformationen, die Positionsdaten in dem PLC-Programm zugewiesen sind, sind als Textinformationen wiedergegeben, die die Positionsdaten in dem Bewegungsprogramm, das von der Robotersteuerung 50 erzeugt ist, darstellen. Deshalb kann der Nutzer leicht verstehen, wie das PLC-Programm und das Bewegungsprogramm miteinander zusammenhängen. Es wird zum Beispiel angenommen, dass die Positionsdateneingabe von der PLC zu der Robotersteuerung einen Fehler aufweist. In diesem Fall kann der Nutzer den Fehler der Eingabedaten bemerken, wenn der Roboter bei einer ungeplanten Position stoppt, wenn aber eine große Anzahl von Befehlen von der PLC zu der Robotersteuerung eingegeben worden sind, ist es schwierig, eine Anweisung zu finden, die den Fehler in dem Bewegungsprogramm aufweist. Selbst in solch einem Fall, wenn ein üblicher Name für den Variablennamen in dem PLC-Programm zu den Positionsdaten in dem Bewegungsprogramm hinzugefügt worden ist, ist es möglich, Positionsdaten, die den Fehler aufweisen, leicht zu suchen und zu finden. Mit anderen Worten, es ist möglich, eine Fehlerprüfung und einen Debug in Anlerneinstellung und Programmierung in dem Robotersteuersystem effizient durchzuführen.

[0027] Ein Konfigurationsbeispiel, wenn die Host-Steuerung 20 als eine CNC (computerisierte numerische Steuereinheit) 20B implementiert ist, wird unten in Bezug auf **Fig.** 4 und 5 beschrieben. **Fig.** 4 veranschaulicht ein beispielhaftes Steuerprogramm, das auf der CNC 20B (das Steuerprogramm, das auf der CNC eingerichtet ist, wird hierin nachfolgend auch als ein CNC-Programm bezeichnet) erstellt ist. Ein CNC-Programm 21B auf Basis von G-Code, das in **Fig.** 4 veranschaulicht ist, beinhaltet den folgenden Inhalt:

G90: Ein absolutes Kommando zum Ausweisen, dass Koordinaten als absolute Werte zu spezifizieren sind.

G01: Ein Kommando, um eine lineare Bewegung auszuführen. Die Bewegung wird bei der folgenden Position und Geschwindigkeit vorgenommen:

Koordinatenposition: X_, Y_, Z_

Lage (Rotationswinkel um X-, Y- und Z-Achsen): A_, B_, C_

Geschwindigkeit: F_

D001: Ein Kommando zum Ausweisen, dass ein Etikett (Zeicheninformationen), das bei Positionsnummer ‚001' in einem spezifischen Bereich auf dem Speicher liegt, an die Robotersteuerung 50 als Namensinformationen für die durch X, Y und Z dargestellte Position zu übertragen ist.

M30: Ein Kommando, das das Ende des Hauptprogramms darstellt.

[0028] Wenn das in **Fig.** 4 veranschaulichte CNC-Programm 21B ausgeführt wird, interpretiert die Steuerungsausführungseinheit 22 das CNC-Programm 21B und versteht, dass der Befehl G01 ein lineares Bewegungskommando ausstellt. Die CNC 20B besitzt die in der zuvor erwähnten Tabelle 1 dargestellte Tabelle und erkennt eine Kommando-ID entsprechend dem Bewegungsbefehl, indem sie in dieser Tabelle nachschlägt. Die Steuerungsausführungseinheit 22 überträgt eine Kommando-ID = 1, die eine lineare Bewegung darstellt, wie auch Positions- und Lagedaten, die in X, Y, Z, A, B und C festgelegt sind, und Geschwindigkeitsdaten. Die Steuerungsausführungseinheit 22 überträgt weiter ein

Etikett, das einer Adresse zugewiesen ist, die mit ‚D001' spezifiziert ist, an die Robotersteuerung 50 als die Namensinformationen der Positionsdaten. Hier wird angenommen, dass ‚HOME.POSITION' bei der Adresse festgelegt ist, die mit ‚D001' als das Etikett spezifiziert ist. In diesem Fall überträgt die Steuerungsausführungseinheit 22 ‚HOME.POSITION' an die Robotersteuerung 50 als die Namensinformationen der Positionsdaten. Die Kommando-ID, die Positionsdaten und das Etikett werden vorübergehend in dem primären Datenspeicherbereich 55 innerhalb der Robotersteuerung 50 gespeichert.

[0029] Die Programmerzeugungseinheit 51 der Robotersteuerung 50 erkennt anhand der Kommando-ID = 1, dass der Bewegungsbefehl eine lineare Bewegung angibt. Die Programmerzeugungseinheit 51 speichert Positions- und Lagedaten (X, Y, Z, A, B, C) und (10,0, 20,0, 30,0, 40,0, 50,0, 60,0), die gemeinsam mit der Kommando-ID = 1 übertragen werden, in einem Positionsregister P[1], erzeugt einen Befehl ‚L P[1]' für eine lineare Bewegung des Roboters 10 als ein Bewegungsprogramm 56B und setzt einen Text HOME.POSITION, der den Namen der Positionsdaten darstellt, der gemeinsam mit den Positions- und Lagedaten an das Positionsregister P[1] übertragen wird, als einen Kommentartext ein. Mit dieser Operation erzeugt die Programmerzeugungseinheit 51

‚L P[1:HOME.POSITION]'

als eine Anweisung einer linearen Bewegung. Die in dem CNC-Programm 21B spezifizierte Geschwindigkeit kann zu der zuvor erwähnten Anweisung als ein numerischer Wert zum Spezifizieren einer Geschwindigkeitsüberschreibung hinzugefügt werden.

[0030] Auf diese Weise wird ein Etikett, das Positionsdaten in dem CNC-Programm 21B darstellt, in das Bewegungsprogramm 56B in der Robotersteuerung 50 als ein Kommentartext eingesetzt, der die Positionsdaten darstellt. Mit anderen Worten, Textinformationen, die Positionsdaten in dem CNC-Programm 21B zugewiesen sind, sind als Textinformationen wiedergegeben, die die Positionsdaten in dem Bewegungsprogramm 56B darstellen, das von der Robotersteuerung 50 erzeugt ist. Deshalb kann der Nutzer leicht verstehen, wie das CNC-Programm und das Bewegungsprogramm miteinander zusammenhängen. Mit anderen Worten, es ist möglich, eine Fehlerprüfung und einen Debug in Anlerneinstellung und Programmierung effizient durchzuführen, so wie in dem zuvor beschriebenen Fall bezüglich des PLC-Programms.

[0031] Da in dem CNC-Programm ein Etikett, das beliebig von dem Nutzer in dem Speicherbereich, der von dem Befehlscode ‚D001' spezifiziert wird, festgelegt wird, als der Name der Positionsdaten in der Robotersteuerung wiedergegeben werden kann, kann die Nutzerfreundlichkeit weiter verbessert werden.

[0032] Wie zuvor beschrieben, kann gemäß dieser Ausführungsform der Nutzer leicht verstehen, wie das Steuerprogramm, das von der Host-Steuerung erzeugt wird, und das Bewegungsprogramm, das von der Robotersteuerung erzeugt wird, miteinander zusammenhängen. Dies macht es möglich, eine Fehlerprüfung und ein Debug bei Anlerneinstellung und Programmierung effizient durchzuführen.

[0033] Obwohl die vorliegende Erfindung zuvor in Bezug auf beispielhafte Ausführungsformen beschrieben wurde, ist es für Fachkundige ersichtlich, dass die vorangehenden und verschiedene andere Änderungen, Auslassungen oder Hinzufügungen an den zuvor beschriebenen Ausführungsformen vorgenommen werden können, ohne von dem Wesen der vorliegenden Erfindung abzuweichen.

[0034] Ein Beispiel, in dem Textinformationen (ein Variablenname oder ein Etikett) selbst von der Host-Steuerung an die Robotersteuerung als die Namensinformationen übertragen werden, wurde in der zuvor beschriebenen Ausführungsform gegeben, aber anstelle solch einer Konfiguration können Identifikationsinformationen (ID), die einen Namen darstellen, als die Namensinformationen übertragen werden. In diesem Fall ist sowohl die Host-Steuerung als auch die Robotersteuerung konfiguriert, eine Tabelle zu besitzen, die die Identifikationsinformationen und die Textinformationen miteinander verknüpft.

[0035] Die Funktion (Steuerungsausführungseinheit) der in **Fig.** 1 veranschaulichten Host-Steuerung kann implementiert werden, indem verschiedene Typen von Software von der CPU der Host-Steuerung ausgeführt werden, oder kann durch eine Konfiguration implementiert werden, die hauptsächlich von Hardware, wie einem ASIC (anwendungsspezifischer integrierter Schaltkreis, „Application Specific Integrated Circuit"), gebildet ist. Die Funktionen (die Programmerzeugungseinheit, die Namensinformationshinzufügungseinheit und die Programmausführungseinheit) der in **Fig.** 1 veranschaulichten Robotersteuerung 50 kann implementiert werden, indem verschiedene Typen von Software von der CPU der Robotersteuerung ausgeführt werden, oder kann durch eine Konfiguration implementiert werden, die hauptsächlich von Hardware, wie einem ASIC, gebildet ist.

BEZUGSZEICHENLISTE

10	Roboter
20	Host-Steuerung

21	Steuerungsprogramm
22	Steuerungsausführungseinheit
23	Kommunikationsschnittstelle
24	Variablendefinition
25a, 25b	Funktionsblock
20A	Programmierbare Logiksteuereinheit
20B	Numerische Steuereinheit
21A	PLC-Programm
21B	CNC-Programm
50	Robotersteuerung
51	Programmerzeugungseinheit
52	Namensinformationshinzufügungseinheit
53	Programmausführungseinheit
54	Kommunikationsschnittstelle
55	Primärer Datenspeicherbereich
56, 56A, 56B	Bewegungsprogramm
100	Robotersteuersystem

Patentansprüche

1. Robotersteuersystem (100), das eine Host-Steuerung (20) und eine mit der Host-Steuerung (20) verbundene Robotersteuerung (50) umfasst,
die Host-Steuerung (20) umfassend
eine Steuerungsausführungseinheit (22), die an die Robotersteuerung (50) Befehlsinformationen, die einen Bewegungsbefehl an einen Roboter (10) darstellen, und Positionsdaten, die mit dem Bewegungsbefehl verknüpft sind, wie auch Namensinformationen, die einen Namen der Positionsdaten darstellen, auf Basis eines Steuerprogramms zum Steuern des Roboters (10) überträgt, und
die Robotersteuerung (50) umfassend:
eine Programmerzeugungseinheit (51), die ein Bewegungsprogramm für den Roboter (10) auf Basis der empfangenen Befehlsinformationen und der empfangenen Positionsdaten erzeugt; und
eine Namensinformationshinzufügungseinheit (52), die den Namen, der von den empfangenen Namensinformationen dargestellt wird, zu den Positionsdaten, die in dem Bewegungsprogramm beinhaltet sind, hinzufügt.

2. Robotersteuersystem (100) nach Anspruch 1, wobei die Steuerausführungseinheit (22) Zeicheninformationen, die den Namen darstellen, als die Namensinformationen überträgt.

3. Robotersteuersystem (100) nach Anspruch 1 oder 2, wobei die Host-Steuerung (20) eine programmierbare Logiksteuereinheit umfasst.

4. Robotersteuersystem (100) nach Anspruch 3, wobei die Steuerungsausführungseinheit (22) einen Variablennamen, der mit den Positionsdaten in dem Steuerprogramm verknüpft ist, als die Namensinformationen überträgt.

5. Robotersteuersystem (100) nach Anspruch 1 oder 2, wobei die Host-Steuerung (20) eine numerische Steuereinheit umfasst.

6. Robotersteuersystem (100) nach Anspruch 5, wobei die Steuerungsausführungseinheit (22) ein Etikett, das mit den Positionsdaten in dem Steuerprogramm verknüpft ist, als die Namensinformationen überträgt.

Es folgen 5 Seiten Zeichnungen

Anhängende Zeichnungen

FIG. 1

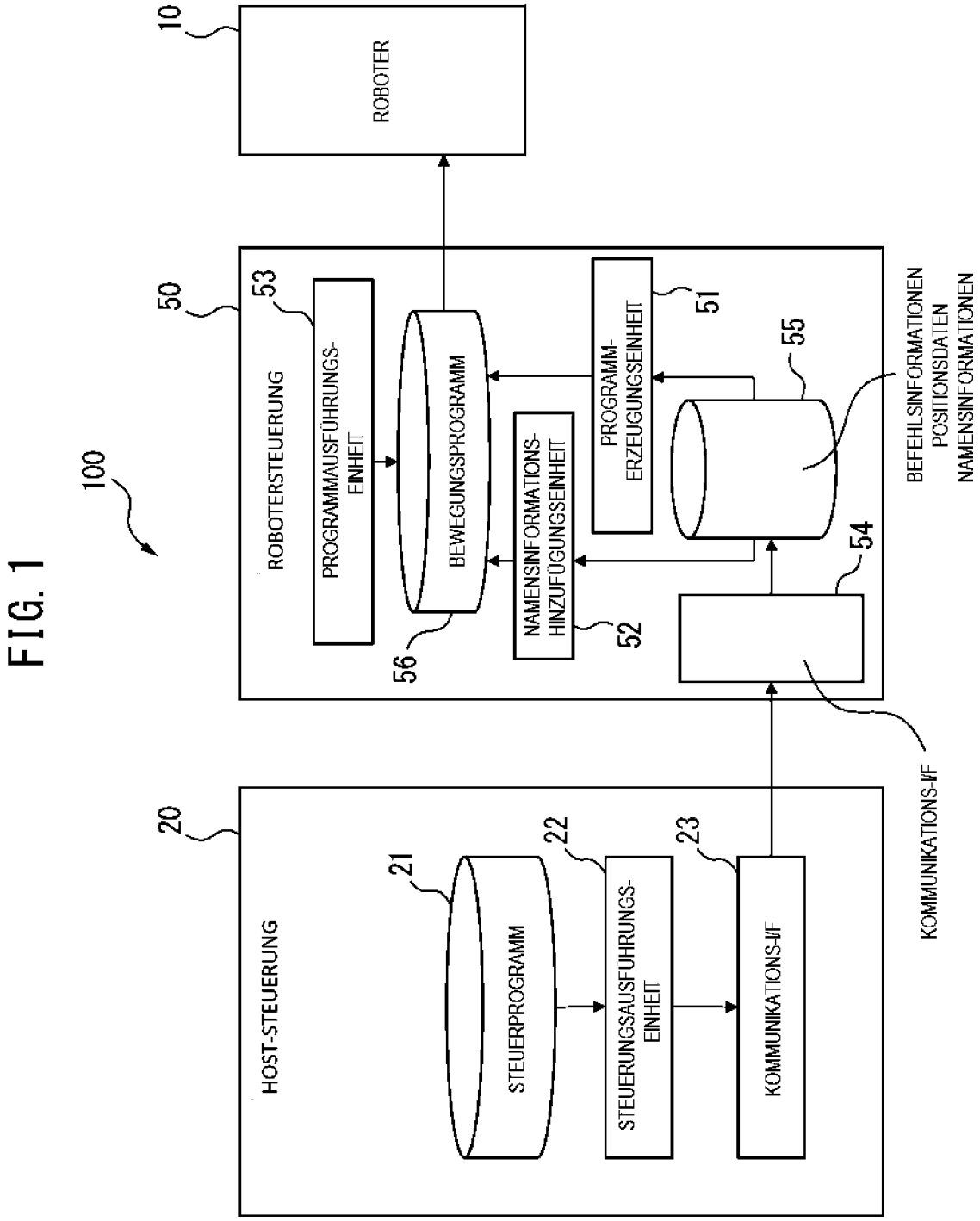


FIG. 2

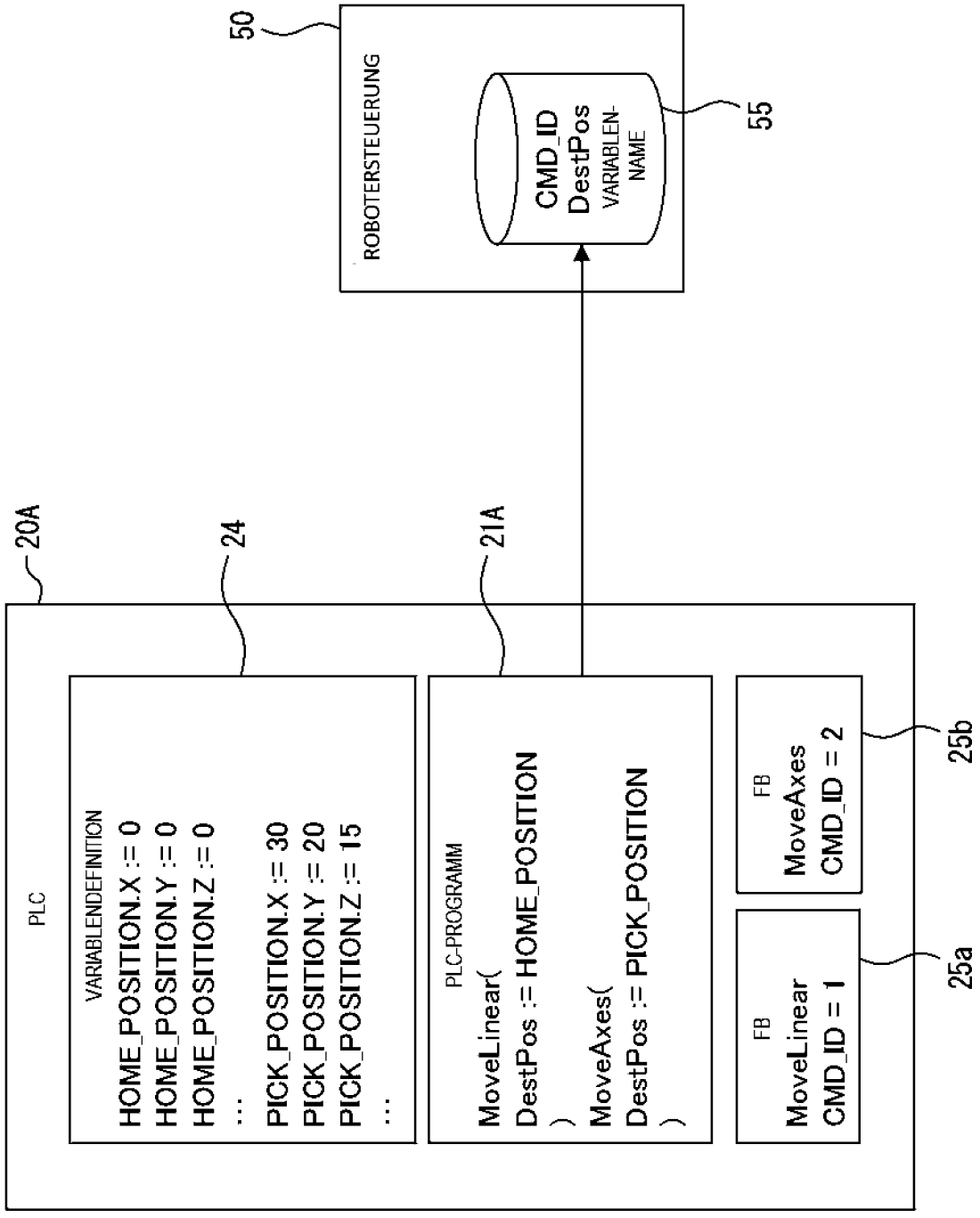

50
ROBOTERSTEUERUNG
CMD_ID
DestPos
VARIABLEN-NAME
55
20A
24
21A
PLC
VARIABLENDEFINITION
HOME_POSITION.X := 0
HOME_POSITION.Y := 0
HOME_POSITION.Z := 0
...
PICK_POSITION.X := 30
PICK_POSITION.Y := 20
PICK_POSITION.Z := 15
...
PLC-PROGRAMM
MoveLinear(
DestPos := HOME_POSITION
)
MoveAxes(
DestPos := PICK_POSITION
)
FB
MoveLinear
CMD_ID = 1
25a
FB
MoveAxes
CMD_ID = 2
25b

FIG. 3

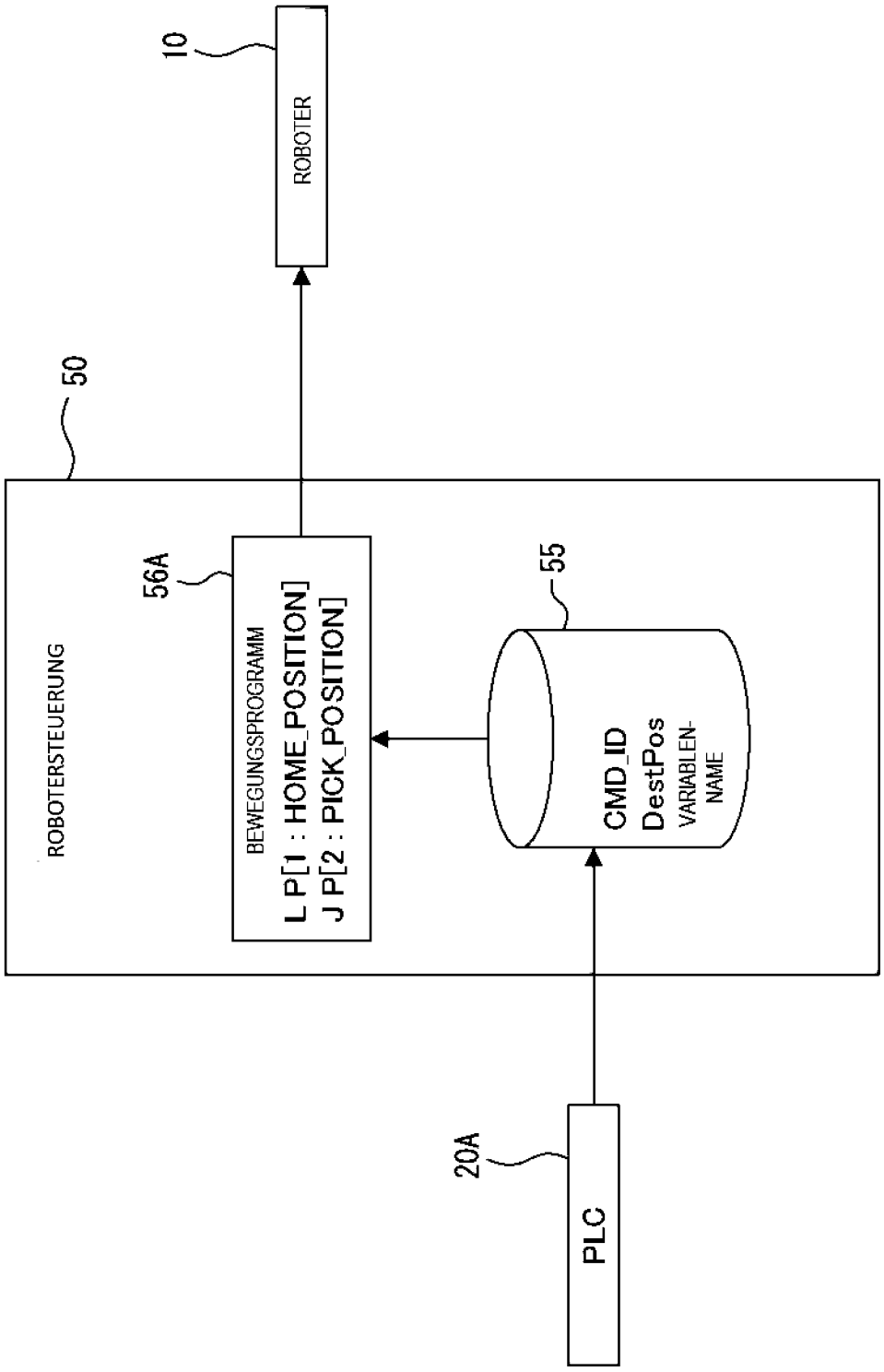

10
ROBOTER
50
ROBOTERSTEUERUNG
56A
BEWEGUNGSPROGRAMM
L P[1 : HOME_POSITION]
J P[2 : PICK_POSITION]
55
CMD_ID
DestPos
VARIABLEN-NAME
20A
PLC

FIG. 4

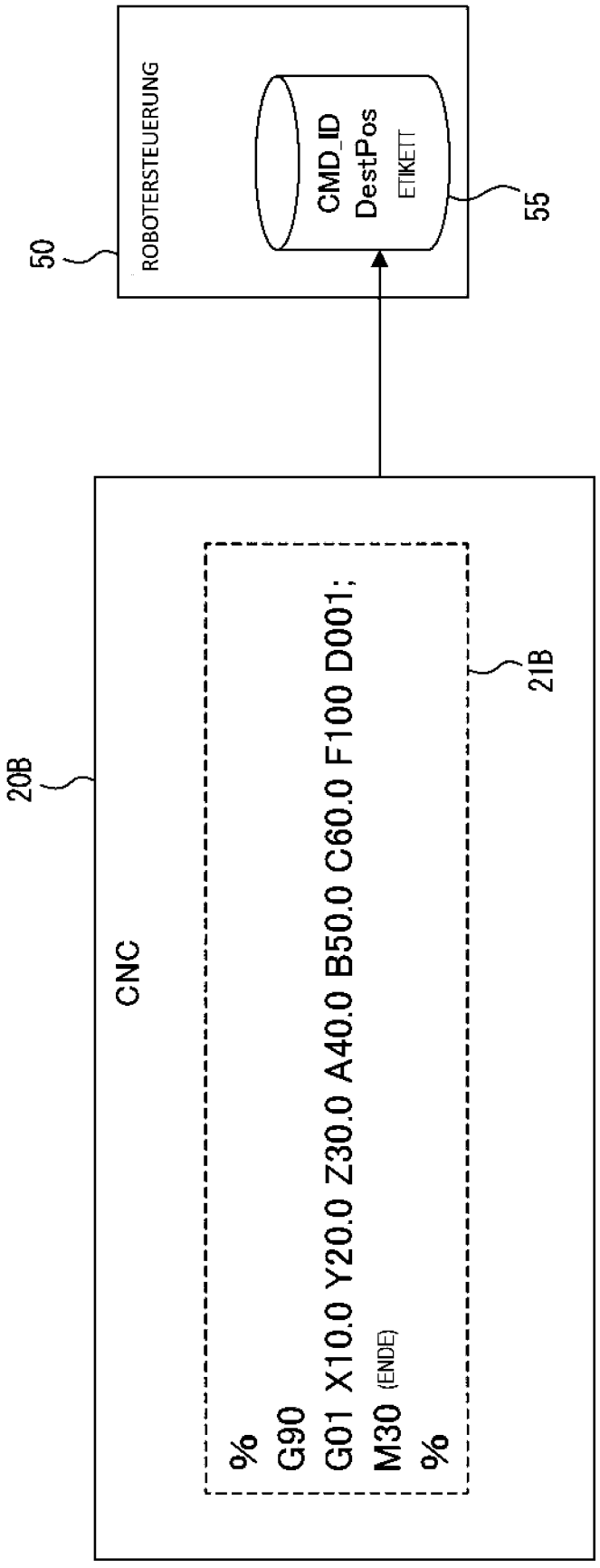
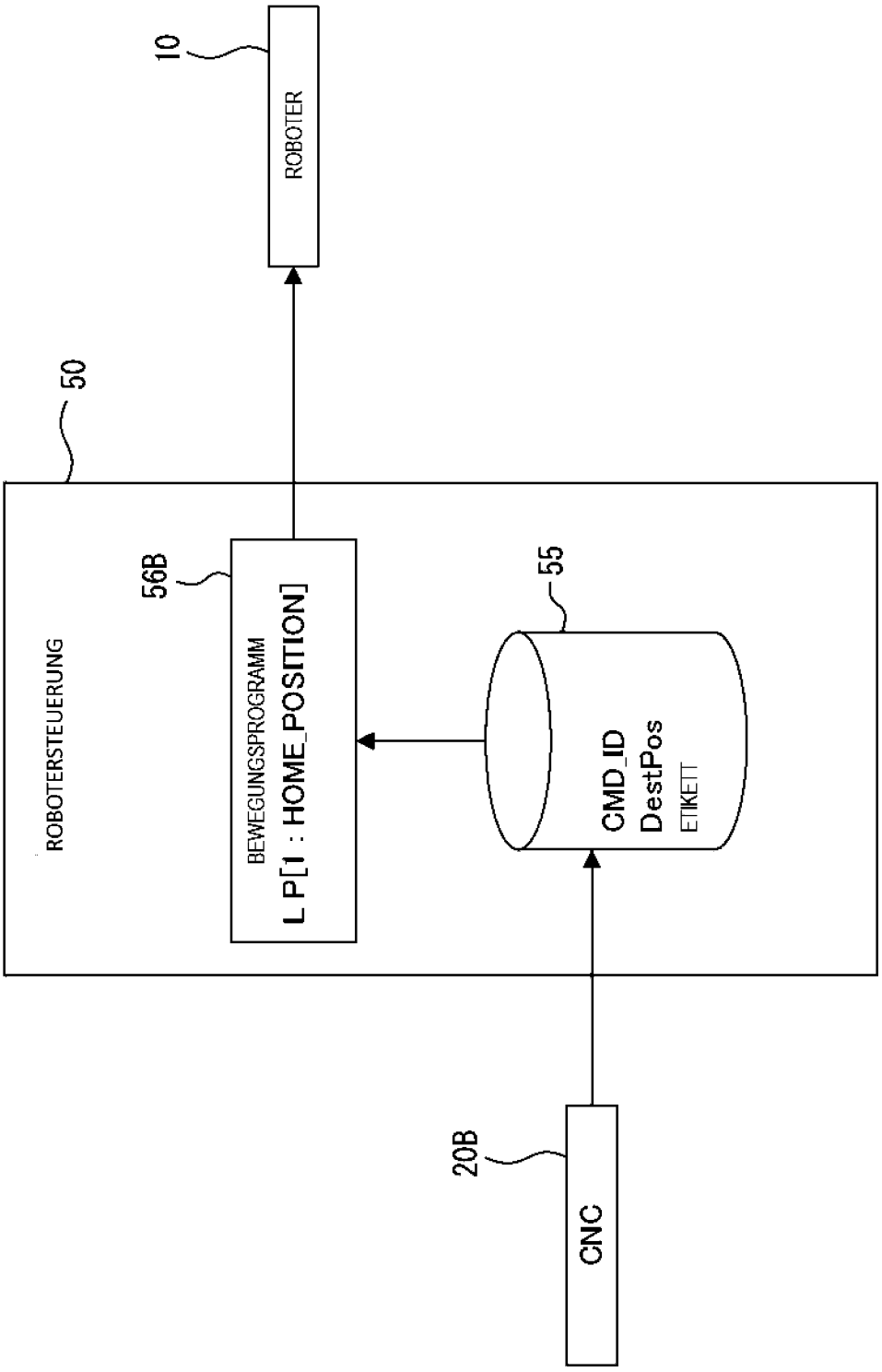

50
ROBOTERSTEUERUNG
CMD_ID
DestPos
ETIKETT
55
20B
CNC
21B
%
G90
G01 X10.0 Y20.0 Z30.0 A40.0 B50.0 C60.0 F100 D001;
M30 (ENDE)
%

FIG. 5



 Das Dokument wurde durch die Firma Luminess hergestellt.