

Beschreibung

Verfahren zum Erzeugen und Abspeichern eines aus Befehlen bestehenden Anwenderprogramms für eine speicherprogrammierbare Steuerung und Betriebsverfahren für eine speicherprogrammierbare Steuerung

Die vorliegende Erfindung betrifft ein Verfahren zum Erzeugen und Abspeichern eines aus Befehlen bestehenden Anwenderprogramms für eine speicherprogrammierbare Steuerung, wobei die speicherprogrammierbare Steuerung bei Abarbeitung des Anwenderprogramms aus Eingangssignalen von einem technischen Prozeß Ausgangssignale für den technischen Prozeß ermittelt, sowie ein hierzu korrespondierendes Betriebsverfahren für eine speicherprogrammierbare Steuerung.

Derartige Verfahren sind allgemein bekannt.

Die Aufgabe der vorliegenden Erfindung besteht darin, neue Verfahren zur Verfügung zu stellen.

Die Aufgabe wird für das Erzeugungs- und Abspeicherverfahren dadurch gelöst, daß die Befehle des Anwenderprogramms jeweils aus einem Funktionscode und dem Funktionscode zugeordneten Funktionsparametern bestehen, daß die Funktionsparameter entweder auf ein Eingangssignal oder auf einen anderen Befehl verweisen und daß das Verfahren folgende Schritte aufweist:

- a) bei der Erzeugung des Anwenderprogramms werden pro Ausgangssignal, beginnend mit dem dem jeweiligen Ausgangssignal zuzuordnenden Befehl, die abzuarbeitenden Befehle abgefragt, wobei folgende Prozedur abgearbeitet wird:
- b) es wird der abzuarbeitende Funktionscode abgefragt und zwischengespeichert;
- c) es wird die Zahl der Funktionsparameter festgestellt;
- d) es werden nacheinander die Funktionsparameter abgefragt;

- e) wenn der Funktionsparameter ein Eingangssignal ist, wird ein Verweis auf das Eingangssignal zwischengespeichert;
- f) wenn der Funktionsparameter ein neuer Funktionscode ist, wird ein Verweis auf einen neuen Befehl zwischengespeichert, wird zu diesem neuen Befehl verzweigt und werden
- 5 der andere Funktionscode zwischengespeichert und die Schritte c) bis f) für den neuen Befehl wiederholt;
- g) die zwischengespeicherten Funktionscodes und die zwischengespeicherten Funktionsparameter werden erst dann in einem
- 10 Remanentspeicher hinterlegt bzw. als gültig gekennzeichnet, wenn alle Befehle zumindest des jeweiligen Ausgangssignals vollständig abgefragt sind.

Auch gemäß dem Betriebsverfahren bestehen die Befehle des Anwenderprogramms jeweils aus einem Funktionscode und dem Funktionscode zugeordneten Funktionsparametern, wobei die Funktionsparameter entweder auf ein Eingangssignal oder auf einen anderen Befehl verweisen. Die erfindungsgemäßen Schritte sind wie folgt:

- 20 a) bei der zyklischen Abarbeitung wird, beginnend mit den den Ausgangssignalen zugeordneten Befehlen, pro Befehl folgende Prozedur abgearbeitet:
- b) es werden der dem Befehl zugeordnete Funktionscode und die diesem Befehl zugeordneten Funktionsparameter gelesen;
- 25 c) pro Funktionsparameter wird überprüft, ob dieser auf einen anderen Befehl verweist;
- d) wenn der Funktionsparameter auf ein Eingangssignal verweist, wird das Eingangssignal zwischengespeichert;
- 30 e) wenn der Funktionsparameter auf einen anderen Befehl verweist, wird zu dem anderen Befehl verzweigt und werden die Schritte b) bis e) mit dem anderen Befehl wiederholt;
- f) wenn alle Funktionsparameter abgearbeitet sind, wird das dem Befehl zugeordnete Ausgangssignal ermittelt.

Weitere Vorteile und Einzelheiten ergeben sich aus der nachfolgenden Beschreibung eines Ausführungsbeispiels. Dabei zeigen:

- 5 FIG 1 den Aufbau eines Programmiergeräts und einer speicherprogrammierbaren Steuerung,
FIG 2 die Benutzerschnittstelle des Programmiergeräts und
FIG 3 ein Programmierbeispiel.

- 10 Gemäß FIG 1 weist das Programmiergerät 1 einen Prozessor 2 auf, der über das Bussystem 3 mit dem Festwertspeicher 4 und dem Schreib-Lese-Speicher 5 sowie den Schnittstellen 6,7 und 8 verbunden ist. Im Festwertspeicher 4 ist das Betriebssystem des Prozessors 2 abgespeichert, der Schreib-Lese-Speicher 5
15 dient dem Programmiergerät 1 als Arbeitsspeicher. Über die Schnittstelle 6 sind mittels der Eingabeeinheit 9, z.B. einer Tastatur, Daten in das Programmiergerät 1 eingebbar. Ebenso sind mittels der Schnittstelle 7 Daten auf der Anzeigeeinheit 10 darstellbar. Mittels der Kommunikationsschnittstelle 8
20 kann das Programmiergerät über die gestrichelt eingezeichnete Kommunikationsverbindung 11 mit einer speicherprogrammierbaren Steuerung 12 kommunizieren.

- Die speicherprogrammierbare Steuerung 12 weist ebenfalls einen
25 Prozessor 13 auf, der ein Anwenderprogramm abarbeitet. Das Anwenderprogramm besteht aus einer Vielzahl von auszuführenden Befehlen und ist im elektrisch lösch- und schreibbaren Festwertspeicher 14 hinterlegt. Gemäß dem abzuarbeitenden Anwenderprogramm greift der Prozessor 13 über den Controller 15
30 und die Prozeßschnittstelle 16 auf einen nicht näher dargestellten technischen Prozeß bzw. eine technische Anlage 17 zu. Er liest von dort Eingangssignale ein, speichert sie im Schreib-Lese-Speicher 18 ab und ermittelt sodann anhand des Anwenderprogramms Ausgangssignale. Die Ausgangssignale werden
35 ebenfalls im Schreib-Lese-Speicher 18 hinterlegt und dann über den Controller 15 und die Prozeßschnittstelle 16 an den

technischen Prozeß 17 ausgegeben. Ferner weist die speicherprogrammierbare Steuerung 12 eine Schnittstelle 19 zum Programmiergerät auf. Der Prozessor 13, die Speicher 14, 18, der Controller 15 sowie die Kommunikationsschnittstelle 19 sind
5 ebenfalls über einen Bus 20 miteinander verbunden.

Programmiergerät 1 und speicherprogrammierbare Steuerung 12 können im Prinzip voneinander getrennte Geräte sein. Vorzugsweise ist das Programmiergerät 1 jedoch, wie in FIG 1 durch
10 die gestrichelte Linie und auch in FIG 2 dargestellt, in die speicherprogrammierbare Steuerung 12 integriert. In beiden Fällen ist es aber möglich, daß das Programmiergerät 1 über die Schnittstellen 8 und 19 sowie die Kommunikationsverbindung 11 auf den Festwertspeicher 14 zugreift. Dabei ist es
15 möglich, daß ein neues Anwenderprogramm in den Festwertspeicher 14 geschrieben wird und daß ein bestehendes Anwenderprogramm gelesen, gelöscht oder geändert wird.

Gemäß FIG 2 weist die speicherprogrammierbare Steuerung 12
20 Schraubanschlüsse 21 bis 24, ein Tastaturfeld 25 sowie ein LCD-Display 26 auf. Über die Schraubanschlüsse 21 ist die speicherprogrammierbare Steuerung 12 mit einer 24 Volt-Stromversorgung verbindbar. Über je zwei der Schraubanschlüsse 22 ist ein Digital-Ausgangssignal an den technischen Prozeß 17
25 ausgebbar. Über je eine der Schraubklemmen 23 ist ein Eingangssignal vom technischen Prozeß 17 einlesbar. Über die Schraubklemmen 24 sind die Aktoren und Sensoren, welche an die Schraubklemmen 22 und 23 angeschlossen sind, mit Strom versorgbar. Die Schraubklemmen 22 bis 24 sind also Bestandteil der Prozeßschnittstelle 16.
30

Das LCD-Display 26 entspricht der Anzeigeeinheit 10 des Programmiergeräts 1. Über das Display 26 sind ferner auch Meldungen der speicherprogrammierbaren Steuerung 12 während des
35 Betriebs anzeigbar. Das Display 26 dient also auch als Beobachtungseinheit. Das Tastaturfeld 25 ermöglicht es, Befehle

direkt in das Programmiergerät 1 und die speicherprogrammierbare Steuerung 12 einzugeben.

Das Anwenderprogramm besteht, wie allgemein üblich aus einzelnen Befehlen, wobei jeder Befehl jeweils aus einem Funktionscode und dem Funktionscode zugeordneten Funktionsparametern besteht. Im Betrieb arbeitet die speicherprogrammierbare Steuerung zyklisch folgende Befehlsfolge ab:

- 10 - Sie liest Eingangssignale vom technischen Prozeß 17 ein.
- Sie ermittelt anhand des Anwenderprogramms Ausgangssignale für den technischen Prozeß.
- Sie gibt die Ausgangssignale an den technischen Prozeß 17 aus.

15

Diese Vorgehensweise ist allgemein bekannt.

Bei der Erzeugung des Anwenderprogramms werden durch die Programmiereinheit 1 nacheinander die abzuarbeitenden Befehle von der Eingabeeinheit 9 abgefragt. Hierzu wird, beginnend mit dem dem ersten Ausgangssignal zuzuordnenden Befehl, abgefragt, welche Funktion abgearbeitet werden soll. Es wird also eine entsprechende Anfrage auf der Anzeigeeinheit 10 ausgegeben und gewartet, bis der Anwender über die Eingabeeinheit 9 den Funktionscode eingegeben hat.

Sowie der Funktionscode eingegeben ist, wird dieser von der Programmiereinheit 1 im Schreib-Lese-Speicher 5 zwischengespeichert. Sodann stellt die Programmiereinheit 1 fest, wie viele Funktionsparameter diesem Befehl zugeordnet sind. Hierbei ist zu unterscheiden zwischen der internen Zuordnung innerhalb des Programmiergeräts 1 und der für den Anwender sichtbaren Zahl der Funktionsparameter. Intern ist die Zahl der Funktionsparameter für alle Funktionscodes derselbe, im vorgegebenen Beispiel gleich vier. Nach außen hingegen wird

stets nur die Zahl der benötigten Funktionsparameter abgefragt. Ein Beispiel soll dies näher erläutern:

Wenn als Funktionscode der Code für die Oderverknüpfung eingegeben wird, werden tatsächlich vier Parameter abgefragt. Diese vier Parameter werden dann später bei der Abarbeitung des Programms auch berücksichtigt. Wenn hingegen z.B. die logische Verneinung als Funktionscode gewählt wird, wird vom Anwender nur ein Parameter, nämlich das zu negierende Eingangssignal, abgefragt. Die anderen drei Parameter werden vom Programmiergerät 1 selbständig aufgefüllt und bei der Abarbeitung des Programms nicht berücksichtigt.

Nach dem Abfragen und Zwischenspeichern des Funktionscodes im Schreib-Lese-Speicher 5 werden durch die Programmiereinheit 1 im Zusammenwirken mit der Eingabeeinheit 9 und der Anzeigeeinheit 10 nacheinander auch die Funktionsparameter abgefragt und zwischengespeichert. Als Funktionsparameter kann entweder ein Eingangssignal oder ein anderer Funktionscode eingegeben werden. Dabei wird bei jeder Eingabe eines Funktionsparameters sofort überprüft, ob dieser ein Funktionscode ist.

Wenn der Funktionsparameter kein Funktionscode, sondern ein Eingangssignal ist, wird dieses zwischengespeichert und mit dem nächsten Funktionsparameter fortgefahren. Unter Eingangssignal sind dabei im vorliegenden Fall nicht nur Eingangssignale vom technischen Prozeß 17 zu verstehen, sondern auch Signale, deren Wert vorbekannt ist. Beispiel für derartige Signale sind das Konstant-Eins- oder das Konstant-Null-Signal sowie ein Taktsignal.

Wenn als Funktionsparameter dagegen ein neuer Funktionscode eingegeben wird, wird als Funktionsparameter nicht der Funktionscode zwischengespeichert, sondern die Adresse eines neuen Befehls. Unmittelbar danach wird zu diesem neuen Befehl verzweigt und dort der unmittelbar zuvor eingegebene Funk-

tionscode zwischengespeichert. Sodann wird erneut die Zahl der Funktionsparameter festgestellt, und es werden die Funktionsparameter abgefragt und zwischengespeichert.

- 5 Auch bei diesen Funktionsparametern wird sofort wieder überprüft, ob sie Eingangssignale oder Funktionscodes sind. Wenn sie neue Funktionscodes sind, wird wiederum erneut sofort die Adresse dieses neuen Befehls als Funktionsparameter abgespeichert und zu diesem neuen Befehl verzweigt. Auch hier wird
10 wieder der soeben eingegebene Funktionscode abgespeichert, festgestellt wieviele Funktionsparameter benötigt werden und die Funktionsparameter abgefragt und zwischengespeichert.

- Dieses verschachtelte Abfragen von Funktionscodes und Funktionsparametern wird solange fortgesetzt, bis alle Funktionsparameter des zuletzt abgefragten und zwischengespeicherten Funktionscodes auf Eingangssignale verweisen. Wenn dann alle Funktionsparameter auf Eingangssignale verweisen, ist der jeweilige Befehl vollständig eingegeben. Es wird dann zurück-
15 gegangen zum vorhergehenden Befehl und bei diesem gegebenenfalls der nächste Funktionsparameter abgefragt und zwischengespeichert, wobei auch hier wieder gegebenenfalls zu weiteren Befehlen verzweigt wird.
20

- 25 Obenstehende Prozedur wird solange fortgesetzt, bis alle zur Ermittlung eines Ausgangssignals erforderlichen Funktionen miteinander verschaltet sind, also alle Funktionsparameter jedes Befehls eingegeben worden sind. Erst wenn dies erfolgt ist, wird der dem nächsten Ausgangssignal zuzuordnende Befehl
30 abgefragt und die Prozedur, soweit erforderlich, wiederholt. Auf diese Art werden nacheinander alle Ausgangssignale programmiert.

- Das Eingeben des Anwenderprogramms kann vom Anwender selbst-
35 verständlich jederzeit abgebrochen werden. In diesem Fall erfolgt jedoch kein Abspeichern des eingegebenen und zwischen-

- gespeicherten Teilprogramms. Beim Ausschalten des Programmiergeräts 1 sind die eingegebenen Daten verloren. Ein Abspeichern der abgefragten und im Schreib-Lese-Speicher 5 zwischengespeicherten Anwenderprogramms im Remanentspeicher 14
- 5 wird erst dann ermöglicht, wenn der Programmiervorgang vollständig beendet ist. Als vollständig beendet gilt der Programmiervorgang bzw. das Erzeugen des Anwenderprogramms dann,
- wenn zu jedem Ausgangssignal ein Befehl eingegeben worden ist, und
- 10 - wenn jeder Funktionsparameter jedes programmierten Funktionscodes belegt ist.

FIG 3 zeigt ein Beispiel eines einfachen Anwenderprogramms. Die durchgezogenen Linien bedeuten dabei die aufgrund der gewünschten Verknüpfungen erforderlichen Verbindungen. Die gestrichelten Linien zeigen die aufgrund des Programmierschemas erforderlichen zusätzlichen Verbindungen, und die strichpunkt-

15 tierten Linien zeigen die vom Programmiergerät 1 selbständig ergänzten Verbindungen. Die Eingabe des Programmierbeispiels gemäß FIG 3 geschieht wie folgt:

20

Zunächst fragt das Programmiergerät über Anzeigeeinheit 10 und Eingabeeinheit 9 ab, welcher Funktionscode für die Ermittlung des Ausgangssignals des ersten Ausgangs A1 benötigt

25 wird. Der Anwender gibt daraufhin über die Eingabeeinheit ein, daß eine UND-Verknüpfung 27 erfolgen soll. Der korrespondierende Funktionscode wird im Schreib-Lese-Speicher 5 des Programmiergeräts 1 zwischengespeichert. Sodann werden nacheinander der erste, der zweite und der dritte Funktionsparameter, d.h. drei der vier Eingangssignale der UND-Verknüpfung

30 27, abgefragt und zwischengespeichert.

Der erste und der zweite Eingangsparameter sind die Eingangssignale E1 und E2. Es wird daher bei der Eingabe dieser beiden Funktionsparameter sofort mit dem nächsten Funktionsparameter fortgefahren. Beim Abfragen des dritten Funktionspa-

35

rameters hingegen wird durch den Anwender eingegeben, daß er eine NICHT-Verknüpfung 28 vorzunehmen wünscht. Als dritter Funktionsparameter wird daher die Adresse des Befehls zwischengespeichert, an dem dann die NICHT-Verknüpfung 28 behandelt wird.

Sodann wird durch das Programmiergerät 1 zu diesem Befehl verzweigt und abgefragt, welches der (einzige) Funktionsparameter der NICHT-Verknüpfung 28 sein soll. Hier wird durch den Anwender das Eingangssignal E3 eingegeben. Das Programmiergerät 1 speichert diesen Funktionsparameter zwischen und ergänzt den Befehl mit drei weiteren Funktionsparametern, welche jeweils auf die logische Eins verweisen. Auch diese drei Funktionsparameter werden im Speicher 5 zwischengespeichert. Hierdurch ist dann die Parametrierung der NICHT-Verknüpfung 28 abgeschlossen. Es wird daher zur weiteren Parametrierung der UND-Verknüpfung 27 zurückgekehrt und der vierte Funktionsparameter abgefragt und zwischengespeichert. Da die UND-Verknüpfung 27 gemäß dem Beispiel nach FIG 3 nur drei echte Eingangsparameter aufweist, wird der vierte Eingang durch entsprechende Eingabe des Anwenders mit der logischen Eins verbunden.

Hiermit ist die Programmierung des Ausgangs A1 abgeschlossen, und es wird mit der Programmierung des Ausgangs A2 fortgefahren.

Zunächst wird abgefragt und nach Eingabe durch den Anwender zwischengespeichert, daß der Funktionscode für die Erzeugung des Ausgangssignals A2 eine ODER-Verknüpfung 29 ist. Sodann werden nacheinander die vier Funktionsparameter der ODER-Verknüpfung 29 abgefragt.

Als ersten Funktionsparameter wird gemäß dem Programmierbeispiel von FIG 3 die bereits programmierte NICHT-Verknüpfung des Eingangs E3 eingegeben. Da diese Verknüpfung bereits fer-

10

tig programmiert ist, ist keine weitere Eingabe bezüglich dieses Funktionsparameters erforderlich und es wird nach der Zwischenspeicherung sofort mit dem zweiten Funktionsparameter fortgefahren.

5

Der zweite und der dritte Funktionsparameter sind die (echten) Eingangssignale E2 und E4, so daß sofort mit dem vierten Funktionsparameter fortgefahren werden kann.

- 10 Gemäß dem Programmierbeispiel von FIG 3 hat auch die ODER-Verknüpfung 29 nur drei Eingangssignale. Der vierte Funktionsparameter verweist daher auf die Konstant-Null.

- Da hiermit auch die Programmierung des Ausgangs A2 abgeschlossen ist, wird mit dem Ausgang A3 fortgefahren.
- 15

- Die Ausgänge A3 und A4 werden gemäß dem gegebenen Programmierbeispiel eigentlich nicht benötigt. Für den Ausgang A3 wird daher die NICHT-Verknüpfung 30 gewählt, welche mit der logischen Eins verknüpft wird. Dadurch ist der Ausgang A3 auf Konstant-Null geschaltet. Gleiches geschieht mit dem Ausgang A4, für welchen die NICHT-Verknüpfung 31 eingegeben wird. Alternativ ist es auch möglich, die Ausgänge A3 und A4 direkt mit der logischen Null zu verbinden.
- 20

25

Das daraus resultierende Programm wird dann im Speicher 14 abgespeichert. Es ist in der nachstehenden Tabelle dargestellt.

- 30 B1: FC (AND)
ADDR (E1)
ADDR (E2)
ADDR (B5)
ADDR ("1")
- 35 B2: FC (OR)
ADDR (B5)

11

```

        ADDR (E2)
        ADDR (E4)
        ADDR ("0")
B3:  FC  (NOT)
5    ADDR ("1")
        ADDR ("1")
        ADDR ("1")
        ADDR ("1")
B4:  FC  (NOT)
10   ADDR ("1")
        ADDR ("1")
        ADDR ("1")
        ADDR ("1")
B5:  FC  (NOT)
15   ADDR (E3)
        ADDR ("1")
        ADDR ("1")
        ADDR ("1")

```

20 Links in Tabelle 1 sind mit B1 bis B5 jeweils der Anfang der einzelnen Befehle dargestellt. Diese Angaben dienen aber nur zur Erläuterung und sind nicht Bestandteil des abgespeicherten Programms. Im eigentlichen Programm steht FC für Funktionscode, ADDR für Adresse.

25

Das Programm als solches besteht aus Blöcken zu je fünf Byte. Das erste Byte stellt jeweils den Funktionscode des Befehls dar, die anderen vier Bytes die vier Funktionsparameter. Die ersten vier Befehle B1 bis B4 definieren stets die Befehle für die vier Ausgangssignale. Ab dem fünften Befehl B5 sind dann Befehle abgespeichert, auf die bei der Berechnung der Ausgangssignale indirekt zurückgegriffen werden muß.

30

Die Abarbeitung des Anwenderprogramms geschieht wie folgt:

35

- Es werden zyklisch die Befehle B1 bis B4 abgearbeitet. Bei der Abarbeitung jedes der Befehle B1 bis B4 wird zunächst der dem Befehl zugeordnete Funktionscode gelesen. Sodann werden nacheinander die diesem Befehl zugeordneten Funktionsparameter
- 5 gelesen. Beim Lesen jedes Funktionsparameters wird sofort überprüft, ob dieser auf einen anderen Befehl verweist. Wenn der Funktionsparameter auf ein Eingangssignal verweist, wird dieses zwischengespeichert.
- 10 Wenn der Funktionsparameter dagegen auf einen anderen Befehl verweist, wird sofort, d.h. vor dem Lesen der anderen Funktionsparameter, zu diesem Befehl verzweigt und der diesen Befehl zugeordnete Funktionscode und die diesem Befehl zugeordneten Funktionsparameter gelesen. Auch hier wird, wenn der
- 15 Funktionscode auf einen neuen Befehl zeigt, sofort wieder dieser Befehl ausgeführt, bevor die weiteren Funktionsparameter abgearbeitet werden. Wenn alle Funktionsparameter eines Befehls abgearbeitet sind, wird das diesem Befehl zugeordnete Ausgangssignal ermittelt, gegebenenfalls zum aufrufenden Befehl
- 20 zurückgesprungen, und das ermittelte Ausgangssignal der weiteren Abarbeitung des Anwenderprogramms zugrunde gelegt.

Auch hier soll das Prinzip anhand des Programms gemäß Tabelle näher erläutert werden.

- 25
- Zunächst wird also der Funktionscode des Befehls B1 gelesen. Sodann werden die Funktionsparameter gelesen, welche auf die Eingänge E1 und E2 verweisen. Beim Lesen des dritten Funktionsparameters wird zum Befehl B5 verzweigt. Dessen Funktionscode und alle vier Funktionsparameter (weil alle auf
- 30 Eingangssignale verweisen) werden gelesen und das dem Befehl B5 zugeordnete Ausgangssignal ermittelt. Mit diesem Ausgangssignal wird dann zum Befehl B1 zurückgekehrt und der vierte Funktionsparameter gelesen. Sodann wird das Ausgangssignal A1
- 35 ermittelt und im Schreib-Lese-Speicher 18 abgespeichert.

Danach wird der Funktionscode des Befehls B2 gelesen und der erste Funktionsparameter des Befehls B2. Sodann wird erneut zum Befehl B5 verzweigt, dessen Funktionscode und dessen vier Funktionsparameter gelesen und das Ausgangssignal des Befehls B5 ermittelt. Sodann wird zum Befehl B2 zurückgekehrt und die übrigen drei Funktionsparameter gelesen. Sodann wird das Ausgangssignal des Befehls B2 ermittelt.

Schließlich werden noch der Funktionscode und die Funktionsparameter des Befehls B3 gelesen und ausgeführt und so das Ausgangssignal A3 ermittelt. Gleiches geschieht mit dem Befehl B4 und dem daraus folgenden Ausgangssignal A4.

Nach der Ermittlung des letzten Ausgangssignals A4 werden die so ermittelten Ausgangssignale A1 bis A4 an den technischen Prozeß 17 ausgegeben und erneut die Eingangssignale E1 bis E4 eingelesen. Sodann wird ein neuer Zyklus bei der Ermittlung der Ausgangssignale A1 bis A4 begonnen.

Alternativ zum beschriebenen Erstellungsverfahren ist es auch möglich, die Befehle, die zu einem Ausgangssignal gehören, sofort im Remanentspeicher 14 abzuspeichern, wenn das jeweilige Ausgangssignal vollständig programmiert ist. Es könnten beispielsweise also die Befehle B1 und B5 sofort abgespeichert werden, sowie das Ausgangssignal A1 vollständig programmiert ist.

Ebenso ist es auch möglich, jeden eingegebenen Funktionscode und jeden eingegebenen Funktionsparameter sofort im Remanentspeicher 14 abzuspeichern. In diesem Fall müßte in einem speziell hierfür vorgesehenen Speicherbereich des Remanentspeichers 14 zu Beginn eines Programmerstellungsvorgangs hinterlegt werden, daß die Programmierung aller Ausgangssignale A1 bis A4 ungültig ist. Beispielsweise könnten vier 1 Bit-Speicherzellen auf logisch-Null gesetzt werden, um anzudeuten, daß die vier Ausgangssignale A1 bis A4 ungültig programmiert

sind. Die vier 1 Bit-Speicherzellen könnten dann auf logisch-Eins gesetzt werden, wenn die Ausgangssignale A1 bis A4 gültig programmiert worden sind. Das Als-gültig-Kennzeichnen kann dabei wahlweise nach der Erstellung des gesamten Anwenderprogramms oder separat für jedes Ausgangssignal A1 bis A4 nach der Abfrage aller Befehle des jeweiligen Ausgangssignals A1 bis A4 erfolgen.

Die Befehle müssen auch nicht in der Reihenfolge gemäß der Tabelle abgespeichert sein. Es ist beispielsweise möglich, alle Befehle, die zur Ermittlung des Ausgangssignals A1 benötigt werden, hier die Befehle B1 und B5, nacheinander im Speicher 14 abzulegen. Die Befehle B2 bis B4 zur Ermittlung der Ausgangssignale A2 bis A4 würden dann hinter diesen Befehlen (hier B1 und B5) abgespeichert werden. Beim Abarbeiten des Anwenderprogramms ist in diesem Fall zu verfolgen, welches die höchste Adresse ist, die zur Ermittlung des Ausgangssignals A1 aufgerufen wird. Der Befehl zur Ermittlung des Ausgangssignals A2 beginnt dann bei der nächsthöheren Adresse. Auf ähnliche Weise, nämlich Zwischenspeichern der höchsten benötigten Adresse, kann dann ermittelt werden, an welcher Stelle die Befehle zur Ermittlung der Ausgangssignale A3 und A4 angeordnet sind. Ob der Fachmann bei der Implementierung der vorliegenden Erfindung der einen oder der anderen Vorgehensweise den Vorzug gibt, liegt jedoch in seinem Belieben.

Patentansprüche

1. Verfahren zum Erzeugen und Abspeichern eines aus Befehlen (B1 bis B5) bestehenden Anwenderprogramms für eine speicher-
5 programmierbare Steuerung,
- wobei die speicherprogrammierbare Steuerung bei Abarbeitung des Anwenderprogramms aus Eingangssignalen (E1 bis E4) von einem technischen Prozeß (17) Ausgangssignale (A1 bis A4) für den technischen Prozeß (17) ermittelt,
10 - wobei die Befehle (B1 bis B5) des Anwenderprogramms jeweils aus einem Funktionscode und dem Funktionscode zugeordneten Funktionsparametern bestehen,
- wobei die Funktionsparameter entweder auf ein Eingangssignal (E1 bis E4) oder auf einen anderen Befehl (B5) verweisen,
15 mit folgenden Schritten:
a) bei der Erzeugung des Anwenderprogramms werden pro Ausgangssignal, beginnend mit dem dem jeweiligen Ausgangssignal (A1 bis A4) zuzuordnenden Befehl (B1 bis B4), die
20 abzuarbeitenden Befehle (B1 bis B5) abgefragt, wobei folgende Prozedur abgearbeitet wird:
b) es wird der abzuarbeitende Funktionscode abgefragt und zwischengespeichert;
c) es wird die Zahl der Funktionsparameter festgestellt;
25 d) es werden nacheinander die Funktionsparameter abgefragt;
e) wenn der Funktionsparameter ein Eingangssignal (E1 bis E4) ist, wird ein Verweis auf das Eingangssignal (E1 bis E4) zwischengespeichert;
f) wenn der Funktionsparameter ein neuer Funktionscode ist,
30 wird ein Verweis auf einen neuen Befehl (B5) zwischengespeichert, wird zu diesem neuen Befehl (B5) verzweigt und werden der andere Funktionscode zwischengespeichert und die Schritte c) bis f) für den neuen Befehl (B5) wiederholt;
35 g) die zwischengespeicherten Funktionscodes und die zwischengespeicherten Funktionsparameter werden erst dann in einem



Remanentspeicher (14) hinterlegt bzw. erst dann als gültig gekennzeichnet, wenn alle Befehle (B1 bis B5) zumindest des jeweiligen Ausgangssignals (A1 bis A4) vollständig abgefragt sind.

5

2. Betriebsverfahren für eine speicherprogrammierbare Steuerung,

- wobei die speicherprogrammierbare Steuerung zyklisch ein aus Befehlen (B1 bis B5) bestehendes Anwenderprogramm abarbeitet, um aus Eingangssignalen (E1 bis E4) von einem technischen Prozeß (17) Ausgangssignale (A1 bis A4) für den technischen Prozeß (17) zu ermitteln,
 - wobei die Befehle (B1 bis B5) des Anwenderprogramms jeweils aus einem Funktionscode und dem Funktionscode zugeordneten Funktionsparametern bestehen,
 - wobei die Funktionsparameter entweder auf ein Eingangssignal (E1 bis E4) oder auf einen anderen Befehl (B5) verweisen,
- mit folgenden Schritten:
- a) bei der zyklischen Abarbeitung wird, beginnend mit den den Ausgangssignalen (A1 bis A4) zugeordneten Befehlen (B1 bis B4), pro Befehl (B1 bis B5) folgende Prozedur abgearbeitet:
 - b) es werden der dem Befehl (B1 bis B5) zugeordnete Funktionscode und die diesem Befehl (B1 bis B5) zugeordneten Funktionsparameter gelesen;
 - c) pro Funktionsparameter wird überprüft, ob dieser auf einen anderen Befehl (B5) verweist;
 - d) wenn der Funktionsparameter auf ein Eingangssignal (E1 bis E4) verweist, wird das Eingangssignal (E1 bis E4) zwischengespeichert;
 - e) wenn der Funktionsparameter auf einen anderen Befehl (B5) verweist, wird zu dem anderen Befehl (B5) verzweigt und werden die Schritte b) bis e) mit dem anderen Befehl (B5) wiederholt;



- f) wenn alle Funktionsparameter abgearbeitet sind, wird das dem Befehl (B1 bis B5) zugeordnete Ausgangssignal (A1 bis A4) ermittelt.
- 5 3. Programmiereinheit zur Durchführung des Verfahren nach Anspruch 1, mit
- Mitteln (2,4,5,6,7,9,10) zum Abfragen und Zwischenspeichern von Funktionscodes abzuarbeitender Befehle;
 - Mitteln (2,4) zum Feststellen der Zahl von Funktionsparametern der abzuarbeitenden Befehle;
 - 10 - Mitteln (2,4,5,6,7,9,10) zum Abfragen und Zwischenspeichern der Funktionsparameter;
 - Mitteln (2,4) zum Überprüfen der Funktionsparameter, ob diese ein neuer Funktionscode sind;
 - 15 - Mitteln (2,4) zum Verzweigen zu einem neuen Befehl, wenn der Funktionsparameter ein neuer Funktionscode ist;
 - Mitteln (2,4) zum Verhindern des Abspeicherns der zwischengespeicherten Funktionscodes und der zwischengespeicherten Funktionsparameter in einem Remanentspeicher (14) bzw. zum
 - 20 Verhindern des Als-gültig-Kennzeichnens, wenn noch nicht alle Funktionsparameter aller abzuarbeitenden Befehle zumindest eines Ausgangssignals (A1 bis A4) zwischengespeichert sind.
- 25 4. Speicherprogrammierbare Steuerung zur Durchführung des Verfahren nach Anspruch 2, mit
- Mitteln (13) zum Lesen von Befehlen zugeordneten Funktionscodes und Funktionsparametern;
 - Mitteln (13) zum Überprüfen der Funktionsparameter, ob diese auf ein Eingangssignal oder auf einen anderen Befehl
 - 30 verweisen;
 - Mitteln (18) zum Zwischenspeichern des Eingangssignals, wenn der Funktionsparameter auf das Eingangssignal verweist;
 - 35 - Mitteln (13) zum Verzweigen zu dem anderen Befehl, wenn der Funktionsparameter auf den anderen Befehl verweist;



18

- Mitteln (13) zum Zurückkehren zu einem aufrufenden Befehl;
- Mitteln (13) zum Ermitteln des dem Befehl zugeordneten Ausgangssignals, wenn alle ihm zugeordneten Funktionsparameter abgearbeitet sind.

5

5. Speicherprogrammierbare Steuerung nach Anspruch 4, die mit einer Programmiereinheit nach Anspruch 3 kombiniert ist.

