

Patentschrift

(51) Int Cl.: **G05B 19/04** (2006.01)

G05B 19/042 (2006.01)

G06F 8/10 (2018.01)

G06F 9/445 (2018.01)

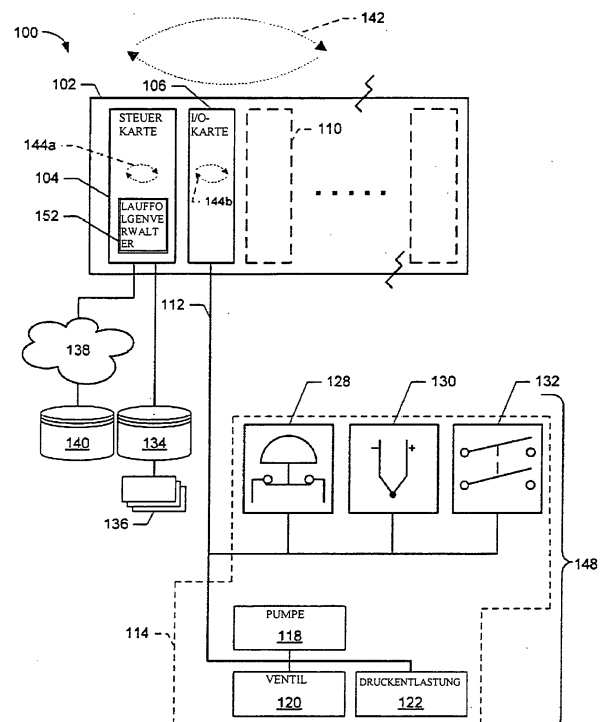
Innerhalb von neun Monaten nach Veröffentlichung der Patenterteilung kann nach § 59 Patentgesetz gegen das Patent Einspruch erhoben werden. Der Einspruch ist schriftlich zu erklären und zu begründen. Innerhalb der Einspruchsfrist ist eine Einspruchsgebühr in Höhe von 200 Euro zu entrichten (§ 6 Patentkostengesetz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

Law, Gary Keith, Georgetown, Tex., US; Hieb, Brandon, Cedar Park, Tex., US; Denison, David R., Austin, Tex., US; Enver, Alper, Austin, Tex., US

DE 41 28 922 A1

DE 41 28 922 A1

(54) Bezeichnung: **Verfahren und Vorrichtung zur Verwaltung von Modullauffolgen in einer Prozesssteuerungsumgebung**



Beschreibung**GEBIET DER OFFENBARUNG**

[0001] Die vorliegende Erfindung betrifft im Allgemeinen Prozesssteuersysteme und insbesondere Verfahren und Vorrichtungen zur Verwaltung von Modullauffolgen in einer Prozesssteuerungsumgebung.

ALLGEMEINER STAND DER TECHNIK

[0002] Prozesssteuerungssysteme wie sie in Chemie-, Erdöl- oder anderen Prozessen verwendet werden, umfassen normalerweise eine oder mehrere zentrale Prozesssteuerungen, die kommunikativ über analoge, digitale oder kombinierte Analog-Digital-Busse mit wenigstens einer Host- oder Bediener-Workstation und mit einem oder mehreren Feldgeräten verbunden sind. Die Feldgeräte, bei denen es sich zum Beispiel um Ventile, Ventilpositionierer, Kessel, Tanks, Schalter und Transmitter (z.B. Temperatur-, Druck- und Durchflusssensoren) handeln kann, führen innerhalb des Prozesses Funktionen wie beispielsweise das Öffnen oder Schließen von Ventilen, Erhöhen/Senken von Temperaturen und/oder Drücken sowie Messen von Prozessparametern aus. Die Prozesssteuerung empfängt Signale, die von den Feldgeräten vorgenommene Prozessmessungen und/oder sonstige Informationen in Bezug auf die Feldgeräte anzeigen, und sie verwendet diese Informationen zum Implementieren einer Steueroutine durch Ausführen eines oder mehrerer Module, die jeweils einen oder mehrere Funktionsblöcke mit Geräteanweisungen enthalten. Diese Geräteanweisungen können ausgeführt werden, um Steuersignale zu erzeugen, die über die Busse oder andere Kommunikationsleitungen an die Feldgeräte gesendet werden, um den Prozessbetrieb zu steuern. Die Informationen von den Feldgeräten und den Steuerungen können für eine oder mehrere Anwendungen verfügbar gemacht werden, die von der Bediener-Workstation ausgeführt werden, um einen Bediener zu befähigen, gewünschte Funktionen in Bezug auf den Prozess auszuführen, wie beispielsweise Anzeigen des aktuellen Prozessstatus, Modifizieren des Prozessbetriebs usw.

[0003] Prozesssteuersystemanwendungen umfassen normalerweise Prozesssteuerroutinen in der Form von Modulen, die so konfiguriert werden können, dass sie verschiedene Funktionen oder Vorgänge in einem Prozesssteuersystem über Funktionsblöcke ausführen. Zum Beispiel kann ein Modul eine Folge von Funktionsblöcken umfassen, um Regelventile, Motoren, Dampfkessel, Heizgeräte und/oder andere Geräte zur Erzeugung eines Produkts (z.B. Erdöl, Kosmetika, Lebensmittel usw.) zu steuern. Die Qualität des hergestellten Produkts kann von der korrekten Funktionsblocksteuerungs-

folge abhängen. Folglich benötigt jedes Produkt möglicherweise ein eindeutiges Modul für jedes gewünschte Prozesssteuerroutinenziel, wie beispielsweise das Halten von Produktspezifikationen innerhalb zulässiger Toleranzen (z.B. Prozentgehalte chemischer Zusammensetzungen, Produktviskosität usw.). Obwohl einige Prozesssteuerroutinenziele die vollständige Ausführung des Moduls (d.h. die Ausführung aller Funktionsblöcke des Moduls) erfordern, benötigen andere Prozesssteuerroutinenziele möglicherweise keine vollständige Ausführung des Moduls (d.h. sie verlangen nicht die Ausführung aller Funktionsblöcke des Moduls). Da ein Teil eines Moduls nicht ausgeführt werden kann, werden dementsprechend häufig zahlreiche zusätzliche Module erstellt, um ein oder mehrere Prozesssteuerroutinenziele für jedes gegebene Produkt zu erfüllen. Aus DE 41 28 922 A1 ist eine elektronische Steuerung für ein Fahrzeug bekannt in der ein computerimplementiertes Verfahren zur Konfiguration einer Modullauffolge implementiert ist, welches umfasst:

- Empfangen eines Moduls, das eine Mehrzahl von Funktionsblöcken enthält,
- Empfangen einer Anzeige eines Teilsatzes der Mehrzahl von Funktionsblöcken,
- Empfangen einer Anzeige einer ersten Ausführungsfolge für den Teilsatz, wobei die erste Ausführungsfolge anders als eine dem Modul zugeordnete zweite Ausführungsfolge ist,
- Zuordnen des Teilsatzes zu einer Lauffolgenkennung und
- Zuordnen der Lauffolgenkennung zu einer Auslöserbedingung.

KURZDARSTELLUNG

[0004] Es werden beispielhafte Vorrichtungen und Verfahren zur Verwaltung von Modullauffolgeroutinen in einem Prozesssteuersystem beschrieben. Ein beispielhaftes Verfahren umfasst ein Empfangen eines Moduls, das eine Mehrzahl von Funktionsblöcken umfasst, Empfangen einer Anzeige eines Teilsatzes der Mehrzahl von Funktionsblöcken und Empfangen einer Anzeige einer ersten Ausführungsfolge für den Teilsatz, wobei die erste Ausführungsfolge anders als eine dem Modul zugeordnete zweite Ausführungsfolge ist. Das beispielhafte Verfahren umfasst außerdem ein Zuordnen des Teilsatzes zu einer Lauffolgenkennung und Zuordnen der Lauffolgenkennung zu einer Auslöserbedingung.

[0005] Gemäß einem anderen Beispiel umfasst eine beispielhafte Vorrichtung einen Lauffolgenverwalter zum Empfangen eines Moduls, das eine Mehrzahl von Funktionsblöcken umfasst, und einen Funktionsblockdefinierer zum Empfangen einer Anzeige eines Teilsatzes von Funktionsblöcken. Die beispielhafte Vorrichtung umfasst außerdem einen Zeitplanver-

walter zum Empfangen einer Anzeige einer ersten Folge Ausführungsfolge für den Teilsatz, wobei die erste Ausführungsfolge für den Teilsatz anders eine dem Modul zugeordnete zweite Folge ist, und einen Auslöserverwalter zum Zuordnen des Teilsatzes einer Lauffolgenkennung und einer Auslöserbedingung.

Besonders bevorzugt wird ein Herstellungsgegenstand, der maschinell zugreifbare Anweisungen speichert, die, wenn ausgeführt, eine Maschine veranlassen zum:

Empfangen eines Moduls, das eine Mehrzahl von Funktionsblöcken enthält; Empfangen einer Anzeige eines Teilsatzes der Mehrzahl von Funktionsblöcken;

Empfangen einer Anzeige einer ersten Ausführungsfolge für den Teilsatz, wobei die erste Ausführungsfolge anders als eine dem Modul zugeordnete zweite Ausführungsfolge ist; Zuordnen des Teilsatzes zu einer Lauffolgenkennung; und

Zuordnen der Lauffolgenkennung zu einer Auslöserbedingung.

Es kann darüber hinaus vorgesehen sein, dass die maschinell zugreifbaren Anweisungen, wenn ausgeführt, die Maschine zum Zuordnen eines der Lauffolgenkennung zugeordneten Maskenwerts zum Teilsatz der Mehrzahl von Funktionsblöcken veranlassen.

Außerdem ist es vorteilhaft, dass die maschinell zugreifbaren Anweisungen, wenn ausgeführt, die Maschine zum Erzeugen eines Lauffolgenzeitplans zum Zuordnen der ersten Ausführungsfolge zu einer Mehrzahl von Feldgeräten veranlassen.

Schließlich kann noch vorgesehen sein, dass die maschinell zugreifbaren Anweisungen, wenn ausgeführt, die Maschine zum Erzwingen von unabhängig von einem dem Modul zugeordneten Zyklus zu übertragenden Daten an einer Gruppe von intelligenten Geräten veranlassen.

Figurenliste

Fig. 1 und **Fig. 2** sind Blockdiagramme von beispielhaften Prozesssteuersystemen, die beim Verwalten von Lauffolgen zu verwenden sind.

Fig. 3 ist ein Blockdiagramm eines beispielhaften Modullauffolgenverwalters, der in **Fig. 1** und **Fig. 2** dargestellt ist.

Fig. 4 ist ein beispielhaftes Modul, das durch die Prozesssteuersysteme von **Fig. 1** und **Fig. 2** ausgeführt werden kann.

Fig. 5 ist ein beispielhaftes lauffolgenfähiges Modul, das durch die Prozesssteuersysteme von **Fig. 1** und **Fig. 2** ausgeführt werden kann.

Fig. 6 und **Fig. 7** zeigen beispielhafte Benutzeroberflächen (UI)-Darstellungen von Modullauffolgenkonfigurationen.

Fig. 8 ist ein Zeitdiagramm eines beispielhaften Modullauffolgenzeitplans.

Fig. 9 und **Fig. 10** sind Flussdiagramme von beispielhaften Verfahren, die zum Implementieren des beispielhaften Modullauffolgenverwalters, des beispielhaften lauffolgenfähigen Moduls, beispielhafter UIs und des beispielhaften Modullauffolgenzeitplans von **Fig. 1** bis **Fig. 8** angewendet werden können.

Fig. 11 ist eine schematische Darstellung einer beispielhaften Prozessorplattform, welche die beispielhaften Prozesse von **Fig. 9** und **Fig. 10** und/oder den beispielhaften Modullauffolgenverwalter von **Fig. 1** bis **Fig. 3** ausführen kann.

AUSFÜHRLICHE BESCHREIBUNG

[0006] Es ist zu erwähnen, dass, obwohl im Folgenden beispielhafte Vorrichtungen und Systeme beschrieben werden, die unter anderen Komponenten auf Hardware ausgeführte Software und/oder Firmware umfassen, diese Systeme lediglich der Veranschaulichung dienen und nicht als einschränkend betrachtet werden sollten. Es ist zum Beispiel vorgesehen, dass beliebige oder alle dieser Hardware-, Software- und Firmwarekomponenten ausschließlich in Hardware, ausschließlich in Software oder in beliebigen Kombinationen von Hardware und Software realisiert sein könnten. Obwohl daher im Folgenden beispielhafte Vorrichtungen und Systeme beschrieben werden, stellen die bereitgestellten Beispiele nicht die einzige Art und Weise zur Implementierung dieser Verfahren und Vorrichtungen dar.

[0007] Prozesssteuersysteme können in beliebigen Maßstäben implementiert werden, wie beispielsweise einem verhältnismäßig kleinen Prozesssteuersystem mit wenigen Eingangs-/Ausgangs (I/O)-Knoten oder einer verhältnismäßig großen Anzahl von Prozesssteuersystemen, die an geografisch völlig verschiedenen Standorten betrieben werden und I/O-Knoten, Pumpensteuerungen/-sensoren, Ventilsteuerungen/-sensoren, Alarmer usw. aufweisen. Jedes Prozesssteuersystem weist normalerweise eine Steuerung auf, wie beispielsweise eine DeltaV™-Steuerung, die von der Fisher-Rosemount Systems, Inc., einem Unternehmen von Emerson Process Management™, vertrieben wird. Die Steuerung kann Anweisungen zum Ausführen eines oder mehrerer Module mit Prozessfunktionsblöcken abrufen und/oder anderweitig empfangen. Module können eine beliebige Anzahl von Prozessfunktionsblöcken zum Steuern eines Chargenbetriebs (z.B. chemische/Lebensmittelverarbeitung), Steuern eines Montagevorgangs (z.B. Materialstanzvorgän-

gen) und/oder Ermöglichen von Tests und Maßprüfung(en) an hergestellten Bauteilen enthalten. Jedes Modul funktioniert als ein Container von Funktionsblöcken, der in der Steuerung auf einer periodischen Basis ausführt. Diese Module ermöglichen es einem Benutzer eines Steuersystems, eine Steuerungsstrategie zu entwickeln, um ein oder mehrere Prozessziele (z.B. Chargenprozesse, Fertigungsprozesse, Testprozesse usw.) zu erfüllen, und die Module können durch das Steuersystem wiederverwendet und/oder nach Bedarf an ein oder mehrere andere Steuersystem(e) verteilt werden. Die Wiederverwendung kann insbesondere für Steuerungssystemfolgen hilfreich sein, die sich in vorhersagbaren Intervallen wiederholen, wie beispielsweise Vorprozessvorgänge (z.B. einen oder mehrere Betriebsvorgänge zur Vorbereitung von Fertigungsmitteln für die Produktion) und Abschaltvorgänge (z.B. einen oder mehrere Betriebsvorgänge zur Vorbereitung von Fertigungsmitteln für den Stillstand außerhalb der Schichtzeiten). In der Tat minimiert die Verwendung von Modulen den Programmieraufwand durch den Benutzer beim Implementieren von Steuerungsstrategien innerhalb des Steuersystems.

[0008] In einem Beispiel umfasst ein Modul eine Kombination von Funktionsblöcken zur Vorbereitung eines Steuersystems und/oder von Aspekten des Steuersystems für den Betrieb, wie beispielsweise einen Kaltstart (Vorprozessvorgang) am Beginn einer Produktionsschicht. In einigen Beispielen umfassen Fertigungs- und/oder Chargenprozesse der Steuerungsumgebung Anfahrfolgen zur Vorbereitung von Einrichtungen für den Betrieb, wie beispielsweise, ohne darauf beschränkt zu sein, das Vorheizen von Öfen (z.B. das Vorbereiten von Öfen für Voralterungstests von Elektronik), Vorheizen von chemischen Mischbehältern, Spülen von Beschickungsrohren mit Reinigungsmittel, Vorfüllen von Pumpen usw. Die Module enthalten entsprechende Funktionsblöcke, die jeweils eine vorgesehene Ablaufreihenfolge durch eine Folge aufweisen. Nach Abschluss einer Modulausführung kann das Modul für eine beliebige Anzahl von Iterationen erneut ausgeführt werden, und/oder es können alternative Module ausgeführt werden, die anderen Steuersystemerfordernissen entsprechen, wie beispielsweise Laufzeit-Chargenfertigungsprozess(en). Für jeden Fall, in dem ein Modul ausführt, führen auch alle dem Modul zugeordneten Funktionsblöcke aus, ungeachtet dessen, ob sie benötigt werden oder nicht.

[0009] Es können alternative Module durch den Benutzer erstellt und/oder anderweitig programmiert werden, um ein oder mehrere alternative Prozesssteuerungsziele zu erfüllen. Alternative Anfahrfolgen können basierend auf Wetterbedingungen, wie beispielsweise, ohne darauf beschränkt zu sein, verhältnismäßig hoher Feuchtigkeit, verhältnismäßig kalten

Temperaturen und/oder verhältnismäßig warmen Temperaturen, erforderlich sein. Zum Beispiel können Betriebseinrichtungen in den Wintermonaten während des Schichtstillstands (z.B. über Nacht) verhältnismäßig niedrige Temperaturen erreichen, wodurch sie einen größeren Zeitaufwand zum Vorheizen benötigen, bevor am nächsten Tag ein oder mehreren Chargenprozesse durchgeführt werden können. In einem anderen Beispiel kann sich während verhältnismäßig feuchter Zeiträume Kondensation auf und/oder in Betriebseinrichtungen (z.B. innerhalb von Chargenprozessrohren) bilden, wodurch sie größere Mengen und/oder höhere Konzentrationen von Reinigungsflüssigkeit für Spülvorgänge benötigen können, bevor ein oder mehrere Chargenprozesse durchgeführt werden können. Um solchen veränderlichen Bedingungen gerecht zu werden, erstellt der Benutzer des Steuersystems normalerweise Module, die für ein gegebenes Szenarium eindeutig sind, und/oder programmiert diese anderweitig. Falls der Unterschied zwischen einem normalen Anfahrvorgang und einem alternativen Anfahrvorgang zum Beispiel ein paar zusätzliche Sekunden Aufheizzeit zum Wegbrennen von übermäßiger Kondensation beträgt, muss der Benutzer normalerweise ein anderes Modul mit einem oder mehreren der alternativen Steuertätigkeit zugeordneten Funktionsblöcken anlegen.

[0010] Im Allgemeinen sind Module in sich abgeschlossene Steuerungsstrategien enthalten, die aus einer beliebigen Anzahl von Funktionsblöcken bestehen. Auf die Funktionsblöcke (FBs) kann nicht unabhängig zur Ausführung getrennt vom Modul zugegriffen werden. Stattdessen können die FBs periodisch und in einer durch das Modul definierten Folge (z.B. FB-1 führt aus, dann FB2, danach FB-3 usw.) ausgeführt werden. Selbst wenn daher die betrieblichen Unterschiede zwischen einem Modul für normales Anfahren und einem für Bedingungen hoher Feuchtigkeit spezifischen Modul minimal sind, werden normalerweise separate Module mit etwas anderen FBs verwendet, um diese minimalen Unterschiede in dem oder den Anfahrvorgängen zu berücksichtigen. Folglich muss ein Benutzer möglicherweise eine verhältnismäßig große Anzahl von separaten, unterschiedlich konfigurierten Modulen entwickeln und/oder anlegen, um dem möglichen Bereich von Steuersystembedingungen (z.B. warme Temperaturen, kalte Temperaturen, feuchte Bedingungen usw.) Rechnung zu tragen.

[0011] Außerdem muss in einigen Fällen jedes Modul, das in einem Prozesssteuersystem (z.B. einer Fabrik, einer Chargenherstellungsanlage, einem Testlabor, einer Prüfstation usw.) ausgeführt werden soll, validiert werden, bevor es zu Verwendung im Steuersystem zugelassen wird. Zum Beispiel schreiben einige Pharmaunternehmen strenge Validierungsverfahren vor, um sicherzustellen, dass

Module eine fehlerfreie Funktionsblocklogik enthalten, und/oder verlangen die Verifizierung, dass alle Funktionsblockeingaben und -ausgaben korrekt konfiguriert sind. Bundes- und Landesgesetze und/oder Gemeindeordnungen können angesichts der Politik in Bezug auf die öffentliche Sicherheit ebenfalls eine Erfüllung von validierten Sicherheitsvorkehrungen für Steuersysteme erforderlich machen (z.B. die von der U.S. Food and Drug Administration angebotenen Richtlinien). Entsprechend kann das Erstellen eines oder mehrerer neuer/zusätzlicher Module für verhältnismäßig unbedeutende Prozesssteuerungsänderungen zu eingehenden und teuren Validierungsverfahren und/oder Tests führen. Nur nach dem erfolgreichen Abschluss solcher Validierungsverfahren können die Module auf ein Steuersystem zur Ausführung verteilt werden, wodurch der Implementierung erhebliche Verzögerungen auferlegt werden. Außerdem schränkt die Unfähigkeit zur bedingten Ausführung (z.B. ereignisbasierte Ausführung, periodische Auslöser usw.) eines oder mehrerer Teile des Moduls die Effizienz der Ausführung ein.

[0012] Im Gegensatz zu bekannten Verwendungen von Modulen und Funktionsblöcken ermöglichen die hierin beschriebenen Verfahren und Vorrichtungen teilweise die Verwendung von Modullauffolgen, die es ermöglichen, einen Teilsatz von Funktionsblöcken innerhalb eines Moduls auszuführen. Wie hierin verwendet, bezieht sich der Begriff „Modullauffolge(n)“ auf einen Satz von Funktionsblöcken innerhalb eines Moduls, die in Reaktion auf eine gegebene Ausführungsanforderung des Moduls auszuführen sind. Mit anderen Worten beschränkt oder benötigt eine Modullauffolge nicht die Ausführung aller FBs innerhalb des Moduls, sondern aktiviert vielmehr nur jene FBs, deren Ausführung zu einem bestimmten Zeitpunkt unter gegebenen Bedingungen bei einem Prozesssteuersystem angebracht oder notwendig ist. Durch die Verwendung von Modullauffolgen kann ein Modul mit einem Teilsatz von Funktionsblöcken in Reaktion auf eine gegebene Bedingung (z.B. ereignisgesteuerte Auslöser, periodische Auslöser usw.) ausführen. Modullauffolgen können in Umgebungen eingesetzt werden, die Ereignisse umfassen, bei welchen das Prozesssteuersystem verhältnismäßig schnell reagieren muss (z.B. Notabschaltung, Reaktion auf Überflutung, Sicherheitsverriegelung usw.). In anderen Prozesssteuerungsumgebungen, die ein zeitplanbasiertes Kommunikationsbusprotokoll (z.B. Fieldbus™ usw.) einsetzen, können Modullauffolgen synchron zum Zeitplan (z.B. synchron zu einem Segment-Makrozyklus) verwendet werden. Außerdem können im Falle einer alternativen oder abweichenden Bedingung (z.B. ein alternatives Ereignis, ein separater Zeitraum in einer Zeitfolge usw.) Modullauffolgen, die einen alternativen oder unterschiedlichen Teilsatz von Funktionsblöcken aktivieren, vom gleichen Modul ausgeführt werden. Die Möglichkeit solch

einer Verwendung der einen oder mehreren alternativen oder verschiedenen Modullauffolgen jedes Moduls in einem Steuersystem kann eine beträchtliche Verringerung der Gesamtanzahl von Modulen bereitstellen, die entwickelt werden müssen und die eine eingehende Validierung erfordern, bevor die Module zur Ausführung durch das Steuersystem zugelassen werden.

[0013] Nunmehr unter Zuwendung zu **Fig. 1** umfasst ein beispielhaftes Prozesssteuersystem 100, das zur Implementierung der hierin beschriebenen beispielhaften Verfahren und Vorrichtungen eingesetzt werden kann, ein Steuerungschassis 102 mit einer Anzahl von Steckplätzen zur Aufnahme von Kartengeräten. Das beispielhafte Steuerungschassis 102 von **Fig. 1** umfasst eine Steuerkarte oder Steuerung 104 und eine Eingangs-/Ausgangs (I/O)-Karte 106, und sie kann je nach Größe des Steuerungschassis 102 einen oder mehrere Reservesteckplätze 110 umfassen. Die I/O-Karte 106 und der entsprechende Datenbus 112 ermöglichen eine Steuerung für industrielle Systeme, wie beispielsweise Produktionsanlagen, Testlabors und/oder Chargenfertigungsbetriebe. Im Allgemeinen können bei den hierin beschriebenen Verfahren und Vorrichtungen Steuerprotokolle und/oder -einrichtungen beliebigen Typs (z.B. intelligente Geräte, standardmäßige Transducergeräte usw.) eingesetzt werden. Obwohl außerdem das in **Fig. 1** veranschaulichte Beispiel in Verbindung mit einer Produktionsumgebung beschrieben wird, können die hierin beschriebenen beispielhaften Verfahren und Vorrichtungen in beliebigen Umgebungen eingesetzt werden, in welchen Prozesssteuersysteme, Datenerfassungssysteme und/oder Test- und Messsysteme (T&M) verwendet werden.

[0014] Der beispielhafte Datenbus 112 ist kommunikativ mit Geräten 114 verbunden, die eine Pumpe 118, ein Ventil 120, ein Entlastungsventil 122, einen Notstopp (N-Stopp)-Schalter 128, ein Thermoelement 130 und einen Verriegelungsschalter 132 z.B. einen Schalter zum Anzeigen eines nominalen Prozessbetriebs oder eines anomalen Betriebs) umfassen. Es kann jegliche Anzahl von Geräten 114 an den Datenbus 112 angeschlossen werden. Die Steuerung 104 kann eine oder mehrere Prozesssteuerungen ausführen, die von einem Werksleiter, einem Prozesssteuerungsdesigner, einem Systemingenieur und/oder beliebigem anderem Personal, das für den Betrieb des Prozesssteuersystems 100 verantwortlich ist, konfiguriert und/oder ausgelegt wurden. Die Steuerung 104 von **Fig. 1** kann zum Beispiel eine DeltaV™-Steuerung sein, die von der Fisher-Rosemount Systems, Inc., einem Unternehmen von Emerson Process Management™, vertrieben wird. Es könnten stattdessen jedoch beliebige andere Steuerungen und/oder Steuerkarten verwendet werden. Obwohl außerdem nur eine Steuerung 104 in

Fig. 1 dargestellt ist, können zusätzliche Steuerungen beliebigen Typs oder beliebiger Typenkombination auf dem beispielhaften Steuerungschassis 102 betrieben und/oder mit dem Steuersystem 100 verbunden werden. Die beispielhafte Steuerung 104 ist kommunikativ mit einer Laufzeitdatenbank 134 verbunden, die ein oder mehrere Module 136 (z.B. als computer- oder prozessorlesbare und -ausführbare Anweisungen) speichert, wobei die Laufzeitdatenbank 134 jedoch auch in der beispielhaften Steuerung 104 enthalten sein kann, ohne darauf beschränkt zu sein.

[0015] In Betrieb arbeitet die beispielhafte Steuerung 104 nach einem Laufzeit-Zeitplan, der ein oder mehrere Module (und entsprechende darin enthaltene FBs) identifiziert und ausführt, die für die gewünschte Steuersystemfunktionalität erforderlich sind. Außerdem ist die beispielhafte Steuerung 104 kommunikativ mit einem Netzwerk 138 (z.B. einem Intranet, dem Internet usw.) verbunden, das ferner mit einer Konfigurationsdatenbank 140 kommunikativ verbunden sein kann. Die beispielhaften Module 136 können auch in der Konfigurationsdatenbank 140 gespeichert werden, so dass sie auf eine beliebige Anzahl von alternativen Steuersystemen verteilt werden können. Falls außerdem die Laufzeitdatenbank 134 beschädigt und/oder anderweitig unfähig wird, der Steuerung 104 Module zur Verfügung zu stellen, können Informationen in Bezug auf die Module 136 und die FBs darin von der Konfigurationsdatenbank 140 abgerufen werden.

[0016] In Betrieb führt das beispielhafte Steuerungschassis 102 einen Chassisbus-Abtastzyklus 142 aus, in welchem jedes Kartengerät in einem belegten Steckplatz des Chassis 102 die Fähigkeit hat, auf einer Rückwandplatine des Chassis 102 zu kommunizieren. Während der Ausführung einer Prozesssteueroutine (z.B. eines Moduls mit einem oder mehreren Funktionsblöcken) kann die Steuerung 104 Informationen (z.B. Befehle, Konfigurationsinformationen, Messdaten, Statusinformationen usw.) mit den Feldgeräten 114 austauschen. Außerdem führt jedes Kartengerät im Steuerungschassis 102 bei einem jeweiligen Kartenabtastzyklus 144a-b aus, wenn sie eine oder mehrere Ausgaben durchführt (z.B. Eingaben misst, Ausgaben aufruft, Daten empfängt, Sollwerte berechnet, Kommunikationsprivilegien von Geräten auf dem oder den Bussen managt, Chargenprozesse, eine Phase des Chargenprozesses oder eine Mehrzahl von Chargenprozessen zur Herstellung eines Produkts organisiert usw.). Die Kartenabtastzyklen 144a-b können asynchron in Bezug aufeinander sein, da zum Beispiel einige Kartengeräte verhältnismäßig schnellere Abtastzyklen benötigen, wie beispielsweise der Abtastzyklus 144b der beispielhaften I/O-Karte 106, um Eingaben von Sicherheitseinrichtungen (z.B. Notstopp 128) zu verarbeiten. Die beispielhaften

Geräte 114 führen gemäß dem Abtastzyklus 144b aus, der eine Zeitdauer für jedes Gerät in einer Gerätegruppe 148 ist, in der es Gelegenheit zum Ausführen und Kommunizieren auf dem Datenbus 112 hat. Die Dauer des Abtastzyklus 144b kann eine Funktion der Anzahl und des Typs von Geräten innerhalb der Abtastgerätegruppe 148 sein, oder sie kann unterbrochen werden, wenn das jeweilige Kommunikationsprotokoll eine Kommunikationsunterbrechung unterstützt.

[0017] Die Feldgeräte 114 arbeiten und/oder werden verarbeitet in einem Intervall synchron zum Kartenabtastzyklus 144b der beispielhaften I/O-Karte 106. Im Falle einer mit der Pumpe 118 verbundenen Notfallbedingung, die zum Beispiel dadurch aufgerufen wird, dass ein Bediener die Notstopp-Taste 128 drückt, ereignen sich folglich ein oder mehrere Befehle zum Abschalten der Pumpe 118 basierend auf dem nächsten ausgeführten Abtastzyklus 144b. In einigen Beispielen weisen Module, die in der Steuerung 104 ausführen, verhältnismäßig langsame Zyklusraten auf, derart dass die Modulausführungszeiten Geräte 114 in verhältnismäßig langsamen Intervallen (z.B. einmal alle 5 Sekunden) aktualisieren. Im Gegensatz dazu verlangen Notfallbedingungen normalerweise verhältnismäßig kurze Reaktionszeiten (z.B. Millisekunden), um die Möglichkeit eines Personen- und/oder Eigentumsschadens zu minimieren. Um mit dem zuvor erwähnten Beispiel fortzufahren, können zur Gewährleistung dessen, dass die beispielhafte Pumpe 118 bald zu laufen aufhört, nachdem vom Bediener die Notstopp-Taste 128 gedrückt wurde, drahtgebundene Relais so konfiguriert werden, dass sie den Netzstrom zur Pumpe 118 elektrisch trennen, statt zur Verarbeitung eines Stoppbefehls an die Pumpe 118 auf die verhältnismäßig langsame Modulausführungsrate angewiesen zu sein. Obwohl drahtgebundene Relais eine reaktionsfähige (z.B. schnelle) Art und Weise zur Trennung der Pumpe 118 ermöglichen, führt die Installation von drahtgebundenen Relais und/oder einer Verkabelung zu zusätzlichen Installationskosten und nimmt in einigen Steuersystemumgebungen möglicherweise zu viel Platz in Anspruch.

[0018] In dem in **Fig. 1** veranschaulichten Beispiel führt ein Modullauffolgenverwalter 152 in der Steuerung 104 aus. In Betrieb überwacht der beispielhafte Modullauffolgenverwalter 152 auf einen oder mehrere Auslöser (z.B. ereignisbasierte Auslöser, periodische Auslöser, sequenzielle Auslöser usw.) und ruft eine dem Auslöser zugeordnete Modullauffolge auf. Im Unterschied zu bekannten Systemen, die ein oder mehrere Module durch Ausführen aller darin enthaltenen FBs gemäß einer vorbestimmten Folge ausführen, ruft die durch den Modullauffolgenverwalter 152 aufgerufene beispielhafte Modullauffolge einen oder mehrere spezifische FBs der

Module zum Ausführen in einer Reihenfolge auf, die auf einem erzeugten Zeitplan basiert. In der Tat bewirkt eine Anforderung zur Ausführung eines Moduls basierend auf einer Modullauffolganweisung, dass das Modul nur ausgewählte FBs von Interesse (d.h. einen Teilsatz der Funktionsblöcke) ausführt, die der oder den Auslöserbedingung(en) zugeordnet sind, und verhindert, dass irgendwelche restlichen FBs des Moduls ausführen. Mit anderen Worten spezifiziert die Modullauffolge eine bedingte Ausführung des Moduls und verbessert die Ausführungseffizienz eines Prozesses durch Begrenzen der FB-Ausführung auf nur jene FBs, die benötigt werden. Außerdem kann die Modullauffolge, statt auf die periodische Beschaffenheit der Module (z.B. eine Ausführung, die nur einmal alle 5 Sekunden stattfindet) angewiesen zu sein, einige FBs dazu vorsehen, periodisch auszuführen, während sie andere FBs dazu vorsieht, basierend auf einem oder mehreren Auslösern (z.B. ereignisbasierte Ausführung) auszuführen. Obwohl das in **Fig. 1** veranschaulichte Beispiel standardmäßige I/O-Feldgeräte 114 beschreibt, sind die hierin beschriebenen Verfahren und Vorrichtungen nicht darauf beschränkt. Zusätzlich oder alternativ können die hierin beschriebenen Verfahren und Vorrichtungen standardmäßige I/O-Feldgeräte, Transducer, intelligente Geräte und/oder beliebige Kombinationen davon einsetzen.

[0019] **Fig. 2** veranschaulicht ein weiteres beispielhaftes Prozesssteuersystem 200 ähnlich dem beispielhaften Prozesssteuersystem 100 von **Fig. 1**. In dem in **Fig. 2** veranschaulichten Beispiel setzt das Prozesssteuersystem 200 ein zeitplanbasiertes Kommunikationsbusprotokoll wie beispielsweise das Foundation Fieldbus™ (im Folgenden Fieldbus™) ein. Segmente innerhalb einer Fieldbus™-Umgebung marshallen die Kommunikation und/oder Ausführung von Steuergeräten gemäß einem Link Active Scheduler (LAS), in dem ein Makrozyklus eine Dauer für die Kommunikation mit jedem Gerät im Segment definiert. Das beispielhafte Prozesssteuersystem 200 umfasst ein Steuerungschassis 202 mit einer Anzahl von Steckplätzen zur Aufnahme von Kartengeräten, welche ferner eine beispielhafte Steuerkarte oder Steuerung 204, eine Fieldbus™-Karte 206, eine I/O-Karte 208 und je nach der Größe des Steuerungschassis 202 eine beliebige Anzahl von Reservesteckplätzen 210 umfasst. Die Fieldbus™-Karte 206 und ein entsprechender Datenbus 212 ermöglichen eine verteilte Steuerung für industrielle Systeme auf eine synchrone Weise, wobei jedes Gerät einen geplanten Zeitpunkt für die Kommunikation über den Bus 212 erhält. Obwohl das in **Fig. 2** veranschaulichte Beispiel Beispiele in Bezug auf das Fieldbus™-Protokoll, wie beispielsweise die Fieldbus™-Karte 206 und die entsprechenden Fieldbus™-Geräte 214, umfasst, sind die hierin beschriebenen Verfahren und Vorrichtungen nicht darauf beschränkt.

[0020] Der beispielhafte Datenbus 212 ist kommunikativ mit den Fieldbus™-Geräten verbunden, die einen Sensor 216, eine Pumpe 218, ein Ventil 220 und ein Entlastungsventil 222 umfassen. Es kann jegliche Anzahl von Fieldbus™-Geräten 214 an den Datenbus 212 angeschlossen werden, der manchmal auch als Segmentbus 248 bezeichnet wird. Jedes Fieldbus™-Gerät 214 ist zum Ausführen von Steuerlogik, wie beispielsweise FBs, die in einem oder mehreren Modulen enthalten sind, imstande. Die Fieldbus™-Geräte 214 werden basierend auf ihren On-Board-Prozessoren, ihrem On-Board-Speicher, ihrer einfachen Konfigurierbarkeit und/oder ihrer Fähigkeit zur Ausführung von FB-Logik (z.B. Alarm(en), Proportional-Integral-Derivative (PID)-Routine(n), Sollwert(en) usw.) manchmal auch als intelligente Geräte bezeichnet. Das beispielhafte Entlastungsventil 222 kann zum Beispiel eine On-Board-PID-Steuerung oder ein On-Board-PID-Modul zum Steuern einer Ventilposition umfassen, um eine Durchflussrate zu regeln, wodurch die Steuerung 204 von der Verantwortung befreit wird, ihre Abtaste des Entlastungsventils 222 zu erhöhen. In Betrieb führt die Steuerung 204 ein Modul aus und gibt Daten über einen oder mehrere verbundene FBs bekannt.

[0021] Ein beispielhafter I/O-Bus 224 ist von der I/O-Karte 208 kommunikativ mit einer beliebigen Anzahl von standardmäßigen I/O-Geräten 226 verbunden. In dem in **Fig. 2** veranschaulichten Beispiel umfassen die standardmäßigen I/O-Geräte 226 Notstoppschalter (N-Stopp) 228, ein Thermoelement 230 und einen Verriegelungsschalter 232 (z.B. einen Schalter zum Anzeigen eines nominalen Prozessbetriebs oder eines anomalen Betriebs). Im Allgemeinen umfassen die standardmäßigen I/O-Geräte 226 keine On-Board-Logik, keinen On-Board-Speicher und/oder keine Fähigkeit zur Ausführung von Logik (z.B. Funktionsblöcken). Die standardmäßigen I/O-Geräte 226 können verhältnismäßig einfache Einrichtungen umfassen, wie beispielsweise, ohne darauf beschränkt zu sein, Transducer, Schalter, Ventile, Anzeigeleuchten, Lichtvorhänge und/oder akustische Alarmer. Die Modullauffolgen erlauben es einem Benutzer jedoch, ereignisbasierte Logik und/oder periodische Ereignisse in industrielle Geräte beliebigen Typs zu integrieren, wie beispielsweise ein intelligentes Gerät (z.B. ein Fieldbus™-Gerät) oder ein standardmäßiges I/O-Gerät. Außerdem sind die hierin beschriebenen Verfahren und Vorrichtungen nicht auf das Fieldbus™-Protokoll und/oder die Fieldbus™-Geräte beschränkt. Stattdessen können Feldgeräte ohne Einschränkung Profibus™-Geräte und/oder HART-kompatible Geräte umfassen, die über den Bus 212 über Profibus™- und/oder HART Protokoll(e) kommunizieren.

[0022] Die Steuerung 204 kann eine oder mehrere Prozesssteuerroutinen ausführen, die vom Werkslei-

ter, dem Prozesssteuerungsingenieur, dem Systemingenieur, dem Konfigurationsingenieur und/oder beliebigem anderem Personal, das für den Betrieb des Prozesssteuersystems 200 verantwortlich ist, konfiguriert und/oder ausgelegt wurden. Ähnlich der Steuerung 100 von **Fig. 1** kann die Steuerung 204 von **Fig. 2** eine DeltaV™-Steuerung sein. Es könnten stattdessen jedoch beliebige andere Steuerungen und/oder Steuerkarten verwendet werden. Obwohl außerdem nur eine Steuerung 204 in **Fig. 2** dargestellt ist, können zusätzliche Steuerungen beliebigen Typs oder beliebiger Typenkombination auf dem beispielhaften Steuerungschassis 202 betrieben und/oder mit dem Steuersystem 200 verbunden werden. Die beispielhafte Steuerung 204 ist kommunikativ mit einer Laufzeitdatenbank 234 verbunden, die ein oder mehrere Module 236 (z.B. als computer- oder prozessorlesbare und -ausführbare Anweisungen) speichert, wobei die Laufzeitdatenbank 234 jedoch auch in der beispielhaften Steuerung 204 enthalten sein kann, ohne darauf beschränkt zu sein. Ähnlich **Fig. 1** arbeitet die beispielhafte Steuerung 204 von **Fig. 2** nach einem Laufzeit-Zeitplan, der ein oder mehrere Module identifiziert, die für die gewünschte Steuersystemfunktionalität erforderlich sind. Die Modullauffolgen sind jedoch mit einem beispielhaften Makrozyklus 246 synchronisiert, der im Folgenden genauer beschrieben wird. Außerdem ist die beispielhafte Steuerung 204 kommunikativ mit einem Netzwerk 238 (z.B. einem Intranet, dem Internet usw.) verbunden, das ferner mit einer Konfigurationsdatenbank 240 kommunikativ verbunden sein kann. Die beispielhaften Module 236 können auch in der Konfigurationsdatenbank 240 gespeichert werden, so dass sie auf eine beliebige Anzahl von alternativen Steuersystemen verteilt werden können. Falls außerdem die Laufzeitdatenbank 234 beschädigt und/oder anderweitig unfähig wird, der Steuerung 204 Module zur Verfügung zu stellen, können Informationen in Bezug auf die Module 236 und die FBs darin von der Konfigurationsdatenbank 240 abgerufen werden.

[0023] In Betrieb führt das beispielhafte Steuerungschassis 202 einen Chassisbus-Abtastzyklus 242 aus, in welchem jedes Kartengerät in einem belegten Steckplatz des Chassis 202 die Fähigkeit hat, auf einer Rückwandplatine des Chassis 202 zu kommunizieren. Während der Ausführung einer Prozesssteuerroutine (z.B. eines Moduls mit einem oder mehreren Funktionsblöcken) kann die Steuerung 204 Informationen (z.B. Befehle, Konfigurationsinformationen, Messdaten, Statusinformationen usw.) mit den Feldgeräten 214 und/oder 226 austauschen. Außerdem führt jedes Kartengerät im Steuerungschassis 202 bei einem jeweiligen Kartenabtastzyklus 244a-c aus, wenn sie eine oder mehrere Ausgaben durchführt (z.B. Eingaben misst, Ausgaben aufruft, Daten empfängt, Sollwerte berechnet, Kommunikationsprivilegien von Geräten auf dem oder

den Bussen marshallt, Chargenprozesse, eine Phase des Chargenprozesses oder eine Mehrzahl von Chargenprozesse zur Herstellung eines Produkts organisiert usw.). Die Kartenabtastzyklen 224a-c können asynchron in Bezug aufeinander sein, da zum Beispiel einige Kartengeräte verhältnismäßig schnellere Abtastzyklen benötigen. Die beispielhaften Fieldbus™-Geräte 214 führen gemäß einem Makrozyklus 246 aus, der eine Zeitdauer für jedes Fieldbus™-Gerät 214 in einem Fieldbus™-Segment 248 ist, in der es Gelegenheit zum Ausführen und Kommunizieren auf dem Datenbus 212 hat. Die Dauer des Makrozyklus 246 ist eine Funktion der Anzahl und des Typs von Geräten innerhalb des Fieldbus™-Segments 248 und wird durch einen Scheduler auf der beispielhaften Fieldbus™-Karte 206 (z.B. dem LAS) gemarshallt. Ein Pool 250 allgemeiner Geräte umfasst Geräte, die im Fieldbus™-Segment 248 arbeiten, sowie die standardmäßigen I/O-Geräte 226.

[0024] Die standardmäßigen I/O-Geräte 226 arbeiten und/oder werden verarbeitet in einem Intervall synchron zum Kartenabtastzyklus 224c der beispielhaften I/O-Karte 208, aber der Kartenabtastzyklus 224c ist asynchron in Bezug auf den beispielhaften Makrozyklus 246. Um Fieldbus™-Geräte (oder ein beliebiges anderes zeitplanbasiertes Kommunikationsprotokoll) und Nicht-Fieldbus™-Geräte in eine oder mehrere Modullauffolgen zu integrieren, werden die beispielhaften Lauffolgen in den Makrozyklus 246 integriert, der allen der ihm zugeordneten Steuergeräte entspricht. Mit anderen Worten ruft die beispielhafte Modullauffolge einen oder mehrere FBs auf, synchron zum Makrozyklus 246 auszuführen.

[0025] In dem in **Fig. 2** veranschaulichten Beispiel führt ein Modullauffolgenverwalter 252 in der Steuerung 204 aus. In Betrieb überwacht der beispielhafte Modullauffolgenverwalter 252 auf einen periodischen und/oder sequenziellen Auslöser und ruft eine dem Auslöser zugeordnete Modullauffolge auf. Wie bereits zuvor angesichts von **Fig. 1** beschrieben, ruft die durch den Modullauffolgenverwalter 252 aufgerufene beispielhafte Modullauffolge im Unterschied zu bekannten Systemen, die ein oder mehrere Module durch Ausführen aller darin enthaltenen FBs gemäß einer vorbestimmten Folge ausführen, einen oder mehrere spezifische FBs der Module, die in einer Reihenfolge ausführen, die auf einem erzeugten Zeitplan basiert. Nur die ausgewählten FBs von Interesse (z.B. ein Teilsatz von Funktionsblöcken), die dem periodischen und/oder sequenziellen Auslöser zugeordnet sind, und sie verhindert, dass irgendwelche restlichen FBs des Moduls ausführen. Außerdem stimmt sich der beispielhafte Lauffolgenverwalter 252 mit dem Makrozyklus 246 ab, der spezifisch für den Typ von zeitplanbasiertem Kommunikationsprotokoll sein kann,

der von dem beispielhaften Prozesssteuersystem 200 verwendet wird (z.B. Fieldbus™).

[0026] In dem in **Fig. 3** veranschaulichten Beispiel umfasst der Modullauffolgenverwalter 152, 252 einen Modullauffolgen-FB-Definierer 302, einen Modullauffolgen-Auslöserverwalter 304 und einen Verwalter 306 erzeugter Zeitpläne, die allesamt kommunikativ mit einer Modullauffolgen-Definitionsdatenbank 308 verbunden sind. Der beispielhafte Modullauffolgen-FB-Definierer 302 und der beispielhafte Modullauffolgen-Auslöserverwalter 304 sind außerdem kommunikativ mit der beispielhaften Laufzeitdatenbank 134, 234 und/oder der beispielhaften Konfigurationsdatenbank 140, 240 von **Fig. 1** und **Fig. 2** verbunden, um Informationen in Bezug auf verfügbare Module, FBs innerhalb eines jeden Moduls und/oder Informationen in Bezug auf die Geräte 148, das Fieldbus™-Segment 248 und/oder den Pool 250 allgemeiner Geräte des beispielhaften Prozesssteuersystems 100, 200 abzurufen. Der beispielhafte Modullauffolgenverwalter 152, 252 umfasst außerdem einen Auslöserüberwacher 310 zum Überwachen des Prozesssteuersystems 100, 200 auf ein oder mehrere Auslöserereignisse (z.B. ereignisbasierte Auslöser, periodische Auslöser usw.), so dass eine entsprechende Modullauffolge zur Ausführung ausgewählt werden kann, wie im Folgenden genauer beschrieben wird. Außerdem umfasst der beispielhafte Lauffolgenverwalter 152, 252 einen Lauffolgenprotokollverwalter 312, einen Lauffolgendiagnoseverwalter 314 und einen Lauffolgenstrukturverwalter 316.

[0027] Vor der Überwachung auf einen oder mehrere Auslöser über den Auslöserüberwacher 310 können ein oder mehrere Benutzer des beispielhaften Prozesssteuersystems 100, 200 eine oder mehrere Modullauffolgen und ihr entsprechendes Verhalten konfigurieren. Der beispielhafte Modullauffolgen-FB-Definierer 302 greift auf ein oder mehrere Module über die beispielhafte Laufzeitdatenbank 134, 234, die beispielhafte Konfigurationsdatenbank 140, 240 und/oder beliebige andere Speicherorte zu, an welchen Module gespeichert sind. Funktionsblöcke, die ein Modul bilden, das durch einen Benutzer ausgewählt wird, werden geparkt und/oder anderweitig identifiziert, einschließlich entsprechender Eigenschaften der FBs. Zum Beispiel umfasst jeder FB einen Benutzungsnamen, ein FB-Symbol zum Darstellen einer oder mehrerer Funktionen des Funktionsblocks, Anschlussparameter zum Identifizieren von Eingängen und/oder Ausgängen, einen Ausführungsreihenfolgenparameter zum Identifizieren, in welcher Reihenfolge der FB innerhalb des Moduls ausführt, und einen Blockabtastratenparameter zum Identifizieren einer entsprechenden FB-Abtastrate. Der beispielhafte Modullauffolgen-FB-Definierer 302 ermöglicht zum Teil eine Fähigkeit, innerhalb des Moduls verfügbare FBs anzuzeigen, um zu prüfen,

ob diese FBs zur Ausführung in Reaktion auf ein Auslöserereignis geeignet sind. Für Prozesssteuersysteme, die kein zeitplanbasiertes Kommunikationsprotokoll einsetzen, wie etwa das beispielhafte Prozesssteuersystem 100 von **Fig. 1**, können Auslöserereignis(se) über den beispielhaften Modullauffolgen-Auslöserverwalter 304 einem oder mehreren FBs zugeordnet und in der Modullauffolgen-Definitionsdatenbank 308 abgelegt werden. Alternativ können für Prozesssteuersysteme, die ein zeitplanbasiertes Kommunikationsprotokoll einsetzen, wie etwa das beispielhafte Prozesssteuersystem 200 von **Fig. 2**, periodische und/oder synchrone Auslöserereignis(se) über den beispielhaften Modullauffolgen-Auslöserverwalter 304 einem oder mehreren FBs so zugeordnet werden, dass sie synchron zu einem Makrozyklus laufen, und in der Modullauffolgen-Definitionsdatenbank 308 abgelegt werden. Um zu ermöglichen, dass FBs in einer gewünschten Ausführungsreihenfolge angeordnet und/oder nicht auf einem oder mehreren synchronen, periodischen und/oder ausgelösten Ereignis(sen) basiert werden, ermöglicht es der beispielhafte Verwalter 306 erzeugter Zeitpläne zum Teil, eine modifizierte FB-Ausführungsreihenfolge (einen Modullauffolgenzeitplan) zu definieren, wie im Folgenden genauer beschrieben wird.

[0028] Während des Betriebs des Prozesssteuersystems 100, 200 überwacht der beispielhafte Auslöserüberwacher 310 auf ein oder mehrere Auslöserereignis(se) (z.B. ereignisbasierte Auslöser, periodische Auslöser, sequenzielle Auslöser usw.). In Reaktion auf das Erkennen eines Auslöserereignisses identifiziert der beispielhafte Auslöserüberwacher 310 eine entsprechende zu aufzurufende Modullauffolge durch Abfragen des beispielhaften Modullauffolgen-Auslöserverwalters 304. Zum Beispiel kann der Auslöser durch den Modullauffolgen-Auslöserverwalter 304 in Form einer Nachschlage-tabelle mit einer entsprechenden Spalte zur Identifizierung der Modullauffolge definiert sein, die in Reaktion auf das Auslöserereignis auszuführen ist. Außerdem wird die vom Auslöserverwalter 304 identifizierte Modullauffolge auch zum Aufrufen des Zeitplanverwalters 306 verwendet, der sich auf den erzeugten Zeitplan bezieht, um die zugeordneten FBs, die auszuführen sind, und ihre Ausführungsreihenfolge zu identifizieren. Falls zusätzlich oder alternativ jeder Funktionsblock Profilinformationen umfasst, welche die Modullauffolge identifizieren, der er zugeordnet ist (z.B. einen Maskenwert), wird der beispielhafte Zeitplanverwalter 306 möglicherweise nicht in jedem Fall benötigt.

[0029] Während des Betriebs des beispielhaften Auslöserüberwachers 310 kann der beispielhafte Lauffolgenprotokollverwalter 312 von einem oder mehreren Eintritt von Lauffolgenaktivität benachrichtigt werden. In einigen Beispielen kann der Lauf-

folgenprotokollverwalter 312 eine Liste von Lauffolgen von Interesse enthalten, die, wenn sie über einen ereignisbasierten Auslöser und/oder einen sequenziellen Auslöser aufgerufen werden, ein Protokolldatum und eine Protokollzeit aufzeichnen. Zusätzlich oder alternativ kann der beispielhafte Auslöserüberwacher 310 den beispielhaften Lauffolgen-diagnoseverwalter 314 von einem oder mehreren Fällen von Lauffolgenaktivität für Diagnosezwecke benachrichtigen. In einigen Beispielen kann der Lauffolgendiagnoseverwalter 314 am Beginn von Prozesssteueraktivität einen Zeitgeber auslösen. Falls eine oder mehrere identifizierte Lauffolgen nicht ausführen, bevor der ausgelöste Zeitgeber einen Schwellenwert erreicht, kann der beispielhafte Lauffolgendiagnoseverwalter 314 eine Benachrichtigungsmeldung an den Benutzer ausgeben. Ohne darauf beschränkt zu sein, kann der beispielhafte Lauffolgendiagnoseverwalter 314 identifizieren, ob eine oder mehrere vorbestimmte Lauffolgen und/oder Reihen von Lauffolgen aufgerufen wurden.

[0030] Wenn der Bediener und/oder anderes Personal, das für den Prozesssteuerbetrieb, die Konfiguration und/oder die Verwaltung verantwortlich ist, Modullauffolgen entwickeln, erweisen sich einige Lauffolgen als besonders nützlich. In einigen Fällen kann eine bestimmte Modullauffolge unter einer beliebigen Anzahl von veränderlichen Bedingungen getestet werden und einen Nutzwert manifestieren, der in anderen Aspekten eines Steuersystems implementiert werden kann, wie beispielsweise den beispielhaften Prozesssteuersystemen 100, 200. Wenn eine Modullauffolge vom Prozesssteuerpersonal als besonders nützlich identifiziert wird, kann der beispielhafte Lauffolgenstrukturverwalter 316 zum Verkapseln einer oder mehrerer Modullauffolgen als eine zusammengesetzte Struktur zur Wiederverwendung in einem oder mehreren zusätzlichen und/oder alternativen Prozesssteuersystemen eingesetzt werden. Die zusammengesetzten Strukturen ermöglichen es dem Personal, Modullauffolgen zu entwerfen und/oder zu implementieren, die über eine oder mehrere Anlagenumgebungen einheitlich sind, und sie können konfigurationsbezogenen Aufwand reduzieren. Das Personal kann neue zusammengesetzte Strukturen zum Beispiel über grafische Benutzeroberflächen entwerfen, die eine Drag-and-Drop-Schnittstelle ermöglichen, und/oder bestehende Modullauffolgen können identifiziert und zur Verteilung in eine zusammengesetzte Struktur umgewandelt werden.

[0031] Nunmehr unter Zuwendung zu **Fig. 4** ist ein beispielhaftes Modul 400 dargestellt, das acht (8) Funktionsblöcke (FBs) enthält. Insbesondere umfasst das Modul 400 einen PERMISSIVE FB 402, einen FORCE_SP FB 404, einen INTERLOCK FB 406, einen OR FB 408, einen DC_CTRL FB 410, einen DEVICE_CTRL FB 412, einen START_STOP

FB 414 und einen MOT_RUNTIME FB 416. Jeder der in **Fig. 4** dargestellten beispielhaften FBs umfasst außerdem einen entsprechenden Ausführungsreihenfolgeparameter 418 a-h zum Identifizieren der Reihenfolge, in welcher der FB innerhalb des Moduls 400 ausführt. Wie bereits erwähnt, ist die Modulausführung bei bekannten Systemen auf eine vordefinierte Ausführungsreihenfolge für alle darin befindlichen FBs beschränkt, was zu Prozessineffizienz führen kann, wenn einer oder mehrere der FBs nicht benötigt werden, um effektiv auf einen oder mehrere Auslöser zu reagieren. Zu Veranschaulichungszwecken nehme man an, dass das beispielhafte Modul 400 in einem gleichbleibenden Zyklus von 10 Sekunden läuft und über Motoraktivität (z.B. einen Motor, der durch den DEVICE_CTRL FB 412 gesteuert wird) eine Ausgabe (z.B. eine Materialausgabe) an einen nachfolgenden Prozess liefert. Man nehme weiter an, dass der nachfolgende Prozess einen Fließwert empfängt und misst, der die Materialausgabe vom Modul 400 anzeigt, und in einem gleichbleibenden Wiederholungszyklus von 100 msec arbeitet. Falls zum Zwecke dieses beispielhaften Szenariums die Materialausgabe vom Modul 400 stoppt, benachrichtigt der nachfolgende Prozess das Modul über den INTERLOCK FB 406 mit einer Abschaltanforderung und verlangt außerdem, dass der Motor (z.B. eine Pumpe), der vom DEVICE_CTRL FB 412 gesteuert wird, innerhalb einer Sekunde abgeschaltet wird, da sonst Prozesseinrichtungen beschädigt werden. Aufgrund der gleich bleibenden Zykluszeit von 10 Sekunden des Moduls 400 kann der Motor jedoch nicht innerhalb einer Sekunde abgeschaltet werden.

[0032] Frühere Versuche, dem zuvor erwähnten beispielhaften Szenarium Rechnung zu tragen, umfassten die Installation von unabhängigen Stromunterbrechungsrelais, die mit dem Motor physisch verdrahtet waren und durch den nachfolgenden Prozess gesteuert wurden, der mit einer verhältnismäßig hohen Abtastrate lief. Alternativ umfassten andere Versuche, dem zuvor erwähnten Szenarium Rechnung zu tragen, ein derartiges Konfigurieren des Moduls, dass es bei einer schnelleren Zyklusrate ausführt (z.B. einer Nyquist-Zyklusrate von einer halben Sekunde, um den Schutz der Prozesseinrichtungen angesichts der Ein-Sekunden-Schadensgrenze zu gewährleisten). Beide Ansätze umfassen eine ziemlich teure und umständliche Implementierung und können in einigen Fällen infolge der jeweiligen Größe des allgemeinen Segments, für das eine Steuerung verantwortlich ist, nicht umgesetzt werden. Zum Beispiel kann in Situationen mit wenigen Steuergeräten (z.B. Pumpen, Sensoren, Ventilen) ein Herabsetzen der Zyklusrate (z.B. des beispielhaften Zyklus 144b von **Fig. 1**) realisierbar sein, aber in Situationen, die wesentlich mehr Steuergeräte einbeziehen, kann es sein, dass ein Herabsetzen der Zyklusrate nicht möglich ist.

[0033] In dem in **Fig. 5** veranschaulichten Beispiel ist das beispielhafte Modul 400 von **Fig. 4** als ein lauffolgenfähiges Modul 500 mit fünf (5) separaten Modullauffolgen (MLF) dargestellt. Konkret umfasst das beispielhafte lauffolgenfähige Modul 500 eine Überwachungs-MLF 502, eine Motorsteuerungs-MLF 504, eine Start/Stopp-MLF 506, eine Wartungs-MLF 508 und eine Verriegelungs-MLF 510. Das beispielhafte lauffolgenfähige Modul 500 umfasst außerdem eine Anzahl von Auslösern T1 bis T7, die, wenn durch den beispielhaften Auslöserüberwacher 310 erkannt, eine entsprechende MLF zur Ausführung aufrufen. Zu Veranschaulichungszwecken identifiziert der Auslöserverwalter 304, falls T2 aufgerufen wird, dann eine entsprechende auszuführende MLF, die in diesem Fall die Überwachungs-MLF 502 ist. Außerdem fragt der Auslöserverwalter 304 den Zeitplanverwalter 306 zum Identifizieren des Funktionsblocks, der eintreten sollte, und seiner Reihenfolge ab. Basierend auf der Identifizierung durch die Überwachungs-MLF 502 identifiziert der Zeitplanverwalter 306 in diesem Beispiel den PERMISSIVE FB 402, den FORCE_SP FB 404, den INTERLOCK FB 406 und den OR FB 408. Falls andererseits T6 aufgerufen wird, wie beispielsweise durch ein Benachrichtigungssignal vom nachfolgenden Prozess, der ein Durchflussratenproblem erkannt hat, bestimmt der Auslöserverwalter 304, dass die Verriegelungs-MLF 510 ausführen sollte.

[0034] Bei Erkennen eines Auslösers im beispielhaften Prozesssteuersystem 100 von **Fig. 1**, das auf einem nichtsynchrone Protokoll basiert, durch den beispielhaften Auslöserüberwacher 310 kann der beispielhafte Modullauffolgenverwalter 152 die Funktionsblöcke zur Ausführung unabhängig von der konfigurierten Ausführungsrate/-periode des Moduls aufrufen, um nötigenfalls ein sofortiges Eingreifen zu gewährleisten. Mit anderen Worten führen zwar die der Verriegelungs-MLF 510 zugeordneten Funktionsblöcke aus, aber es führt kein anderer FB innerhalb des MLF-fähigen Moduls 500 aus. Entsprechend ist die Ausführungseffizienz des lauffolgenfähigen Moduls 500 verbessert, so dass es auf eine Notfallsituation besser anspricht und nicht durch die 5-Sekunden-Ausführungsrate begrenzt ist, die während des Normalbetriebs auftritt. Andererseits kann bei Erkennen eines synchronen Ereignisses/Auslösers im beispielhaften Prozesssteuersystem 200 von **Fig. 2**, das auf einem synchronen Protokoll basiert, durch den beispielhaften Auslöserüberwacher 310 der beispielhafte Modullauffolgenverwalter 252 einen entsprechenden Makrozyklus 246 identifizieren, der durch das synchrone Protokoll (z.B. den Fieldbus™-Makrozyklus für das Segment 248) implementiert ist. Sobald der Makrozyklus 246 identifiziert ist, implementiert der beispielhafte Modullauffolgenverwalter 252 die Laufolge, die mit den Segmentgeräten 248 synchron läuft.

[0035] **Fig. 6** veranschaulicht eine beispielhafte Benutzerschnittstelle (UI) 600 (z.B. eine grafische Benutzeroberfläche, eine Textoberfläche usw.), die zum Konfigurieren einer oder mehrerer Modullauffolgen zur Verwendung mit den hierin beschriebenen Verfahren und Vorrichtungen verwendet werden kann. In dem in **Fig. 6** veranschaulichten Beispiel ist die UI 600 einem ausgewählten Modul (z.B. Modul A) zugeordnet, das über eine Modul-auswählen-Schaltfläche 601 ausgewählt wird und Folgen-Etikettfelder 602 zum Bestimmen einer MLF-Nomenklatur, Folgen-Auswahlkästchen 604 zum Auswählen einer oder mehrerer definierter Modullauffolgen für ihre Löschung, MLF-Aktivieren-Auswahlkästchen 606 zum Aktivieren jeder entsprechenden MLF, wenn geprüft, und Periodisch-Auswahlkästchen 608 zum Bestimmen, welche Modullauffolgen periodisch ausführen. Die beispielhafte UI 600 umfasst auch ein FB-Zuweisungsfeld 610, um die Erstellung einer Zuordnung zwischen einer MLF und einem oder mehreren FBs zu ermöglichen. Die Auswahl einer Browse-Schaltfläche 612 führt zu einer Liste (nicht dargestellt) mit allen FBs, die im Modul (z.B. Modul A) enthalten sind.

[0036] Die beispielhafte UI 600 umfasst auch ein beispielhaftes Maskenwert-Feld 614, um jeder MLF und jedem Funktionsblock einen Maskenwert zuzuordnen zu können. Zum Beispiel kann eine der FB-Eigenschaften den Maskenwert umfassen, so dass, wenn ein lauffolgenfähiges Modul aufgerufen wird, eine einfach binäre UND-Operation identifizieren kann, welche FBs darin zum Ausführen vorgesehen sind. Ohne Einschränkung ist der jedem FB und/oder jeder MLF zugeordnete Maskenwert nicht auf vier Bits beschränkt, sondern kann eine beliebige Länge aufweisen und manuell oder automatisch zugeordnet werden, um eine Doppelung zu verhindern. Falls der Benutzer wünscht, eine oder mehrere konfigurierte Modullauffolgen zu löschen oder eine oder mehrere Modullauffolgen hinzuzufügen, eine entsprechende MLF-Löschen-Schaltfläche 616 bzw. eine MLF-Hinzufügen-Schaltfläche 618.

[0037] **Fig. 7** veranschaulicht eine beispielhafte UI 700, die zum Konfigurieren eines oder mehrerer Auslöser zur Verwendung mit den in **Fig. 6** konfigurierten Modullauffolgen verwendet werden kann. In dem in **Fig. 7** veranschaulichten Beispiel ist die UI 700 einem ausgewählten Modul (z.B. Modul A) zugeordnet und identifiziert, welcher Auslöser überwacht werden sollte, die, wenn wahr, eine oder mehrere Modullauffolgen aufrufen. Die beispielhafte UI umfasst eine Referenzidentifikationsspalte 702, die eine Zuordnung zwischen einem Referenzauslöser und einer zu erstellenden MLF ermöglicht. Referenzen und/oder Referenzauslöser sind Eingaben und/oder Systemereignisse, wie beispielsweise Signalreferenzen für I/O-Daten, externe oder dynamische Referenzen auf andere Module und/oder

interne Referenzen innerhalb des Moduls, ohne darauf beschränkt zu sein. Eine Browse-Spalte 704 umfasst entsprechende Browse-Schaltflächen, die zum Erzeugen einer Liste (nicht dargestellt) von verfügbaren Referenzauslösern innerhalb des Moduls, wie beispielsweise Eingaben in FBs, auszuwählen sind. Eine Bedingungsspalte 706 und eine Wertespalte 708 ermöglichen die Durchführung einer Zuordnung zwischen jedem entsprechenden Referenzauslöser und einem logischen Zustand. Falls der logische Zustand bei einer Prüfung auf Gleichheit wahr ist, dann identifiziert eine Folgenspalte 710, welche MLF aufgerufen werden sollte. Unter anderen Umständen bewirkt jede Änderung des Referenzauslösers, dass eine entsprechende MLF aufgerufen wird, wie durch die beispielhafte Folgenspalte 710 dargestellt.

[0038] Wie bereits erwähnt, werden die FBs, die in Reaktion auf einen Auslöser ausführen, ferner so spezifiziert, dass sie in einer bestimmten Folge basierend auf einem erzeugten Zeitplan ausführen. In dem in **Fig. 8** veranschaulichten Beispiel ist ein erzeugter Zeitplan 800 mit einer Steuerungszeitlinie 802, einer Motor (Pumpen)-Gerätezeitlinie 804, einer Durchflusssensor-Zeitlinie 806 und einer Segmentzeitlinie 808 dargestellt. Die Segmentzeitlinie 808 stellt Daten dar, die während dieses Zeitraums auf einem Bus, wie beispielsweise den Feldgeräten 148, dem Feldbus™ 248 und/oder dem Pool 250 allgemeiner Geräte, übertragen (bekannt gegeben) werden. Die Steuerungszeitlinie 802 stellt Steuerungsaktivität dar und kann sein, wenn die Module und/oder Modullauffolgen ausgeführt werden. Die Motorzeitlinie 804 und die Durchflusssensorzeitlinie 806 stellen zeitliche Aktivität in Bezug auf diese Geräte dar.

[0039] In Betrieb wird der beispielhafte erzeugte Zeitplan 800 so konfiguriert, dass er auf Situationen anspricht, in welchen T6 wahr ist, 810, wie beispielsweise Situationen, in welchen der Durchflusssensor ein Problem identifiziert. In einigen Beispielen kann das Modul A 400 von **Fig. 4** mit einem nichtsynchronisierten Kommunikationsprotokoll bei einer Zyklusrate von einmal alle fünf Sekunden laufen, die eine Dauer oder Reaktionsverzögerung überschreitet, die zu Schäden an Einrichtungen führen könnte. Das beispielhafte lauffolgenfähige Modul 500 von **Fig. 5** spricht jedoch auf den Auslöser T6 an und identifiziert die MLF, die T6 entspricht und in diesem Fall die Motorsteuerungs-MLF 504 ist (814). Der beispielhafte Verwalter 306 erzeugter Zeitpläne, der in einer Steuerung und/oder in einem beliebigen Prozesssteuersystemelement mit Modulen ausführen kann, bestimmt, welche FBs aus dem Modul 500 auszuführen sind und in welcher Reihenfolge. Um mit dem vorstehenden Beispiel fortzufahren, werden der DC_CTRL FB 410 (816) und der DEVICE_CTRL FB 412 (818) aufgerufen, um den Betrieb des Motors

anzuhalten und die Prozesseinrichtungen zu schützen. Obwohl das vorstehende Beispiel eine nichtsynchronisierte Umgebung und einen ereignisbasierten Auslöser veranschaulicht, kann ein beispielhafter erzeugter Zeitplan auch in Verbindung mit synchronisierten Auslösern in einer verteilten Steuerungsumgebung konfiguriert werden, wie beispielsweise jenen, die gemäß einem Segment-Makrozyklus funktionieren.

[0040] Obwohl beispielhafte Prozesssteuersysteme 100, 200 zur Verwaltung von Modullauffolgen dargestellt und beispielhafte Modullauffolgenverwalter 152, 252 in **Fig. 1**, **Fig. 2** und **Fig. 3** veranschaulicht wurden, können eine oder mehrere Schnittstellen, Datenstrukturen, Elemente, Prozesse, Benutzeroberflächen und/oder Geräte, die in **Fig. 1** bis **Fig. 8** dargestellt sind, kombiniert, geteilt, ungeordnet, weggelassen, eliminiert und/oder auf eine beliebige andere Weise implementiert werden. Außerdem können die beispielhafte Laufzeitdatenbank 134, 234, die Konfigurationsdatenbank 140, 240, die Module 136, 236, der Modullauffolgenverwalter 152, 252, der Modullauffolgen-FB-Definierer 302, der Modullauffolgen-Auslöserverwalter 304, der Verwalter 306 erzeugter Zeitpläne, die Modullauffolgen-Definitionsdatenbank 308, der Auslöserüberwacher 310, der Lauffolgenprotokollverwalter 312, der Lauffolgen-diagnoseverwalter 314 und/oder der Lauffolgenstrukturverwalter 316 von **Fig. 1**, **Fig. 2** und **Fig. 3** durch Hardware, Software und/oder Firmware implementiert werden. Demnach können zum Beispiel beliebige der beispielhaften Laufzeitdatenbank 134, 234, der Konfigurationsdatenbank 140, 240, der Module 136, 236, des Modullauffolgenverwalters 152, 252, des Modullauffolgen-FB-Definierers 302, des Modullauffolgen-Auslöserverwalters 304, des Verwalters 306 erzeugter Zeitpläne, der Modullauffolgen-Definitionsdatenbank 308, des Auslöserüberwachers 310, des Lauffolgenprotokollverwalters 312, des Lauffolgendiagnoseverwalters 314 und/oder des Lauffolgenstrukturverwalters 316 durch eine oder mehrere Schaltung(en), anwendungsspezifische integrierte Schaltung(en) (ASIC(s)), programmierbare Logikbaustein(e) (PLD(s)) und/oder anwenderprogrammierbare Logikbaustein(e) (FPLD(s)) usw. implementiert werden. Darüber hinaus kann der Modullauffolgenverwalter 152, 252 Schnittstellen, Datenstrukturen, Elemente, Prozesse und/oder Geräte anstelle der oder zusätzlich zu den in **Fig. 1** bis **Fig. 8** veranschaulichten umfassen, und er kann mehr als eines von beliebigen oder allen der veranschaulichten Schnittstellen, Datenstrukturen, Elemente, Prozesse, Benutzeroberflächen und/oder Geräte umfassen.

[0041] **Fig. 9** und **Fig. 10** veranschaulichen beispielhafte Prozesse, die zur Implementierung des beispielhaften Prozesssteuersystems 100, 200 und des beispielhaften Modullauffolgenverwalters 152,

252 von **Fig. 1** bis **Fig. 8** durchgeführt werden können. 1-8. Die beispielhaften Prozesse von **Fig. 9** und **Fig. 10** können durch einen Prozessor, eine Steuerung und/oder ein beliebiges anderes geeignetes Verarbeitungsgerät durchgeführt werden. Zum Beispiel können die beispielhaften Prozesse von **Fig. 9** und **Fig. 10** in codierten Anweisungen realisiert sein, die auf einem beliebigen dinglichen computerlesbaren Medium gespeichert sind, wie beispielsweise einem Flash-Speicher, einer CD, einer DVD, einer Diskette, einem Festwertspeicher (ROM), einem Direktzugriffsspeicher (RAM), einem programmierbaren ROM (PROM), einem elektronisch programmierbaren ROM (EPROM) und/oder einem elektronisch löschbaren PROM (EEPROM), einer optischen Speicherplatte, einem optischen Speichergerät, einer Magnetspeicherplatte, einem Magnet-speichergerät und/oder beliebigen anderen Medien, die zum Befördern oder Speichern von Programmcode und/oder -anweisungen in der Form von maschinenlesbaren Anweisungen oder Datenstrukturen verwendet werden können und auf die durch einen Prozessor, einen Universal- oder Spezialcomputer oder eine andere Maschine mit einem Prozessor (z.B. die beispielhafte Prozessorplattform P100, die im Folgenden in Verbindung mit **Fig. 11** beschrieben wird) zugegriffen werden kann. Kombinationen der zuvor erwähnten fallen ebenfalls in den Rahmen von computerlesbaren Medien. Maschinenlesbare Anweisungen umfassen zum Beispiel Anweisungen und/oder Daten, die einen Prozessor, einen Universalrechner, einen Spezialrechner oder eine Spezialverarbeitungsmaschine veranlassen, einen oder mehrere bestimmte Prozesse zu implementieren. Alternativ können einige oder alle beispielhaften Prozesse von **Fig. 9** und **Fig. 10** unter Verwendung beliebiger Kombination(en) von ASIC(s), PLD(s), FPLD(s), diskreter Logik, Hardware, Firmware, usw. implementiert werden. Außerdem können ein oder mehrere Vorgänge der beispielhaften Prozesse von **Fig. 9** und **Fig. 10** stattdessen manuell oder als beliebige Kombinationen beliebiger der zuvor genannten Techniken, zum Beispiel beliebigen Kombination(en) von Firmware, Software, diskreter Logik und/oder Hardware implementiert werden. Außerdem können viele andere Verfahren zur Implementierung der beispielhaften Vorgänge von **Fig. 9** und **Fig. 10** eingesetzt werden. Zum Beispiel kann die Reihenfolge der Ausführung der Blöcke geändert werden, und/oder es können ein oder mehrere der beschriebenen Blöcke verändert, entfernt, unterteilt oder kombiniert werden. Außerdem können beliebige oder alle beispielhaften Prozesse von **Fig. 9** und **Fig. 10** der Reihe nach und/oder parallel durchgeführt werden, zum Beispiel durch getrennte Verarbeitungsthreads, diskrete Logik, Prozessoren, Geräte, Schaltungen usw.

[0042] Der beispielhafte Prozess 900 von **Fig. 9** beginnt damit, dass der Modullauffolgen-Funktions-

blockdefinierer 302 eine Modulauswahl empfängt (Block 902), wie beispielsweise eine Modulauswahl von einem Benutzer über die beispielhafte UI 600 von **Fig. 6**. Im Allgemeinen veranschaulicht der beispielhafte Prozess 900 von **Fig. 9** eine Art und Weise, in welcher Modullauffolgen zur Ausführung in den beispielhaften Prozesssteuersystemen 100 und 200 von **Fig. 1** und **Fig. 2** konfiguriert werden können. Das ausgewählte Modul wird durch den beispielhaften MLF-FB-Definierer 302 geparkt und/oder anderweitig überprüft, um alle FBs zu identifizieren, die sich darin befinden können (Block 904). Wie bereits in Verbindung mit **Fig. 3** erwähnt, ist der MLF-FB-Definierer 302 kommunikativ mit der Laufzeitdatenbank 134, 234, der Konfigurationsdatenbank 140, 240 und/oder einer beliebigen anderen Datenquelle verbunden, die ein oder mehrere Module zur Verwendung in den beispielhaften Prozesssteuersystemen 100 und 200 von **Fig. 1** und **Fig. 2** speichern kann. Nach ihrer Extraktion können die FBs innerhalb des ausgewählten Moduls für die UI 600 verfügbar gemacht werden, falls die beispielhafte Browse-Schaltfläche 612 ausgewählt wird. Eine oder mehrere FB-Auswahlen werden durch den beispielhaften MLF-FB-Definierer 302 (z.B. über die UI 600) empfangen und in einer Folge angeordnet, in welcher sie auszuführen sind (Block 906). Zum Beispiel können die ausgewählten FBs in einer Reihenfolge bevorzugter Ausführung im beispielhaften FB-Zuweisungsfeld 610 auf der UI 600 von **Fig. 6** aufgelistet sein. Zusätzlich oder alternativ können eine oder mehrere Eigenschaften der ausgewählten FBs bearbeitet und/oder anderweitig vergrößert werden, um eine Folgen-Reihenfolge anzuzeigen, falls der FB zur Ausführung durch eine MLF aufgerufen wird.

[0043] Der Teilsatz von empfangenen Funktionsblöcken wird einer MLF-Kennung, wie beispielsweise einem beschreibenden Namen für die MLF, zugeordnet (Block 908). Diese MLF-Nomenklatur kann in den beispielhaften Folgen-Etikettfeldern 602 von **Fig. 6** ausgewiesen sein. Wie bereits erwähnt, kann jede MLF einen ihr zugeordneten eindeutigen Maskenwert aufweisen, der beim Vergleich mit Maskenwerten eines oder mehrerer FBs bestimmt, ob der FB der MLF zugeordnet ist oder nicht. Wenn zum Beispiel einer MLF ein binärer Wert von 1010 zugeordnet ist, wird einem oder mehreren FBs, die in Verbindung mit der MLF ausführen, ebenfalls derselbe binäre Wert zugeordnet. Maskenwerte für FBs können als Teil der Parameterwerte gespeichert werden, die jeden FB definieren und die über logische UND-Operationen die Durchführung von Zuordnungen zwischen dem MLF-Maskenwert und einem oder mehreren FB-Maskenwerten erlauben. Außerdem kann die MLF einem Auslöser zugeordnet sein (Block 910), der, wenn aktiviert (wahr), bewirkt, dass die MLF die Folge von ausgewählten FBs ausführt. Auslöser können Fieldbus™-Geräteeingaben (z.B. Pumpen, Ven-

tile, Sensoren usw.), die bei einem Makrozyklus arbeiten, und/oder Nicht-Fieldbus™-Geräteeingaben, die asynchron zu einem Zyklus arbeiten, umfassen, ohne darauf beschränkt zu sein. Nicht-Fieldbus™-Geräte können Sensoren, Transducer, Thermolemente, Steuereingänge und/oder sicherheitsbezogene Eingänge, wie beispielsweise Lichtvorhänge und Notstoppschalter umfassen, ohne darauf beschränkt zu sein. Ohne darauf beschränkt zu sein, können Auslöser auch Eingaben von vorhergehenden periodischen Aktivitäten, Ausgabe(n) von anderen FBs und/oder Ausgaben basierend auf einer vorhergehenden Modulausführung umfassen.

[0044] Der beispielhafte Verwalter 306 erzeugter Zeitpläne erzeugt einen Zeitplan für die definierte MLF (Block 912), wie den beispielhaften erzeugten Zeitplan 800 von **Fig. 8**. Wie in dem in **Fig. 8** veranschaulichten Beispiel dargestellt, ruft die ausgewählte/empfangene Auslösereingabe 810 eine bestimmte Folge von Ereignissen basierend auf den FBs auf, die der aufgerufenen MLF zugeordnet sind. Obwohl der beispielhafte erzeugte Zeitplan 800 von **Fig. 8** als eine Reihe von Zeitlinien und zugehöriger Geräte- und/oder Segmentaktivität veranschaulicht ist, kann der Verwalter 306 erzeugter Zeitpläne den Zeitplan als eine geordnete Tabelle erzeugen und diese in der Lauffolgen-Definitionsdatenbank 308 speichern (Block 914).

[0045] Es kann jegliche Anzahl von Modullauffolgen für ein empfangenes Modul erstellt werden. Falls der Benutzer beschließt, eine andere MLF für das gleiche ausgewählte Modul zu wählen (Block 916), kehrt die Steuerung zu Block 904 zurück. Falls der Benutzer dagegen beschließt, ein alternatives Modul auszuwählen, für das eine oder mehrere Modullauffolgen zu erstellen sind (Block 918), kehrt die Steuerung zu Block 902 zurück.

[0046] Der beispielhafte Prozess 1000 von **Fig. 10** beginnt damit, dass der beispielhafte MLF-Verwalter 152, 252 bestimmt, ob ein aktives Modul ein lauffolgenfähiges Modul ist (Block 1002), wie beispielsweise ein durch den beispielhaften Prozess 900 von **Fig. 9** modifiziertes Lauffolgenmodul. Ist dies nicht der Fall, dann wartet der beispielhafte Prozess 1000 auf einen Fall, bei dem das aktive Modul lauffolgenfähig ist, andernfalls bestimmt der Auslöserüberwacher 310, ob ein oder mehrere Auslöser (z.B. eine Schaltereinabe, eine Schutzschaltereingabe, ein Überflutungssensor, eine periodisch erzeugte Eingabe usw.) wahr sind (Block 1004). Wenn dies der Fall ist, dann fragt der beispielhafte MLF-Auslöserverwalter 304 die beispielhafte MLF-Definitionsdatenbank 308 ab, um zu bestimmen, welche MLF dem erkannten Auslöserereignis zugeordnet ist (Block 1006). Sobald die entsprechende MLF identifiziert ist, wird der der identifizierten MLF zugeordnete Maskenwert extrahiert, und es wird eine logi-

sche UND-Operation durchgeführt, um die entsprechenden FBs zu identifizieren, die bei der identifizierten MLF auszuführen sind (Block 1008).

[0047] Zur Bestimmung der korrekten Betriebsfolge für die identifizierten FBs, die bei der MLF auszuführen sind, fragt der beispielhafte Verwalter 306 erzeugter Zeitpläne die beispielhafte MLF-Definitionsdatenbank 308 ab, um den gespeicherten erzeugten Zeitplan abzurufen (Block 1010). Wie bereits erwähnt, beschreibt der zuvor gespeicherte erzeugte Zeitplan, welche Prozesssteuergeräte an der Ausführung eines oder mehrerer FBs teilnehmen, und spezifiziert die Ausführungsreihenfolge dieser FBs (Block 1012). Wie in **Fig. 8** dargestellt, kann der beispielhafte erzeugte Zeitplan 800 Fälle in synchronen Protokollumgebungen identifizieren, in welchen zu/von einem oder mehreren Geräten (z.B. der Gruppe 214 intelligenter Geräte) und/oder dem Pool 250 allgemeiner Geräte über den Datenbus 212 und/oder den I/O-Bus 224 zu sendende Daten erzwungen werden, und/oder er kann in Verbindung mit nichtsynchrone Protokollumgebungen ablaufen. Der erzeugte Zeitplan 800 veranschaulicht außerdem, dass eine Prozesssteuerfolge in einem Prozesssteuersystem 100 bei mehreren und veränderlichen Zyklusraten für verschiedene Geräte realisiert werden kann. Mit anderen Worten können Geräte, die bei einem Zyklus ausführen, der asynchron zu I/O-Geräten außerhalb des Zyklus ist (z.B. N-Stopp, Verriegelung usw.), dank der zugeordneten MLF-Konfiguration des Moduls oder der Module und/oder dank des Koordinierens einer oder mehrerer Modullaufolge(n) mit einem synchronen Zyklus, wie beispielsweise einem Fieldbus™-Makrozyklus, in einer vorhersagbaren und reaktionsfähigen Weise zusammenarbeiten können. Nach der Ausführung des erzeugten Zeitplans (Block 2012) kehrt die Steuerung zu Block 1002 zurück, um auf einen Fall eines anderen lauffolgenfähigen Moduls zu warten.

[0048] **Fig. 11** ist eine schematische Darstellung einer beispielhaften Prozessorplattform P100, die verwendet und/oder programmiert werden kann, um beliebige oder alle der beispielhaften Laufzeitdatenbank 134, 234, der Konfigurationsdatenbank 140, 240, der Module 136, 236, des Modullauffolgenverwalters 152, 252, des Modullauffolgen-FB-Definierers 302, des Modullauffolgen-Auslöserverwalters 304, des Verwalters 306 erzeugter Zeitpläne, der Modullauffolgen-Definitionsdatenbank 308, der beispielhaften Module 136, 236, des beispielhaften Auslöserüberwachers 310, des Lauffolgenprotokollverwalters 312, des Lauffolgendiagnoseverwalters 314 und/oder des Lauffolgenstrukturverwalters 316 von **Fig. 1**, **Fig. 2** und **Fig. 3** zu implementieren. Zum Beispiel kann die Prozessorplattform P100 durch einen oder mehrere Universalprozessoren, Prozessorkerne, Mikrocontroller usw. implementiert sein.

[0049] Die Prozessorplattform P100 des Beispiels von **Fig. 11** umfasst mindestens einen programmierbaren Universalprozessor P105. Der Prozessor P105 führt codierte Anweisungen P110 und/oder P112 aus, die in einem Hauptspeicher des Prozessors 105 (z.B. in einem RAM P115 und/oder einem ROM P120) vorhanden sind. Der Prozessor P105 kann eine Verarbeitungseinheit beliebigen Typs sein, wie beispielsweise ein Prozessorkern, ein Prozessor und/oder ein Microcontroller. Der Prozessor P105 kann unter anderem die beispielhaften Prozesse von **Fig. 9** und **Fig. 10** zur Implementierung der hierin beschriebenen Verfahren und Vorrichtungen ausführen.

[0050] Der Prozessor P105 steht mit dem Hauptspeicher (einschließlich eines ROMs P120 und/oder RAMs P115) über einen Bus P125 in Verbindung. Der RAM P115 kann durch einen dynamischen Direktzugriffsspeicher (DRAM), einen synchronen dynamischen Direktzugriffsspeicher (SDRAM) und/oder ein RAM-Gerät beliebigen anderen Typs implementiert sein, und der ROM kann durch einen Flash-Speicher und/oder ein Speichergerät beliebigen anderen gewünschten Typs implementiert sein. Der Zugriff auf den Speicher P115 und den Speicher P120 kann von einer Speichersteuerung (nicht dargestellt) gesteuert werden. Der beispielhafte Speicher P115 kann zum Implementieren der beispielhaften Laufzeitdatenbank 134, 234, der beispielhaften Konfigurationsdatenbank 140, 240 und/oder der beispielhaften Modullauffolgen-Definitionsdatenbank 308 von **Fig. 1**, **Fig. 2** und **Fig. 3** verwendet werden.

[0051] Die Prozessorplattform P100 umfasst außerdem eine Schnittstellenschaltung P130. Die Schnittstellenschaltung P130 kann durch einen Schnittstellenstandard beliebigen Typs implementiert sein, wie beispielsweise eine externe Speicherschnittstelle, einen seriellen Port, einen Universal-Eingang/-Ausgang usw. Ein oder mehrere Eingabegeräte P135 und ein oder mehrere Ausgabegeräte P140 sind an die Schnittstellenschaltung P130 angeschlossen.

[0052] Obwohl bestimmte beispielhafte Verfahren, Systeme und Herstellungsgegenstände hierin beschrieben wurden, ist der Deckungsumfang dieses Patents nicht darauf beschränkt. Im Gegenteil deckt dieses Patent alle Verfahren, Systeme und Herstellungsgegenstände, die entweder wörtlich oder gemäß der Äquivalenzlehre zur Gänze in den Rahmen der angehängten Ansprüche fallen.

Patentansprüche

1. Computerimplementiertes Verfahren zur Handhabung einer Modullauffolge, umfassend: Empfangen eines ausgewählten Moduls (136; 400; 500), das eine Mehrzahl von Funktionsblöcken (FB) enthält, wobei das Modul Prozesssteuerroutinen

definiert, um verschiedene Funktionen in einem Prozesssteuersystem (100; 200) auszuführen; Empfangen einer Anzeige einer Auswahl eines Teilsatzes der Mehrzahl von Funktionsblöcken, die auszuführen sind; Empfangen einer Anzeige einer ersten Ausführungsfolge (MLF) für den Teilsatz der Mehrzahl von auszuführenden Funktionsblöcken, wobei die erste Ausführungsfolge anders als eine der Mehrzahl von Funktionsblöcken des Moduls zugeordnete zweite Ausführungsfolge des Teilsatzes der Mehrzahl von Funktionsblöcken mit einer Lauffolgenkennung ist; Zuordnen des Teilsatzes zu einer Lauffolgenkennung; Zuordnen der Lauffolgenkennung zu einer Auslöserbedingung; und Erzeugen eines Lauffolgenzeitplans, um die erste Ausführungsfolge der Mehrzahl von Feldgeräten zuzuordnen, wobei der Lauffolgenzeitplan dazu zwingt, dass zu übertragende Daten unabhängig von einer dem Modul zugeordneten Ausführungsrate auf einem Bus (112; 212) einer Gruppe von intelligenten Geräten (114; 214) übertragen werden.

2. Verfahren nach Anspruch 1, wobei die Lauffolgenkennung einen Maskenwert umfasst, vorzugsweise ferner umfassend ein Zuweisen des Teilsatzes der Mehrzahl von Funktionsblöcken (FB) zu dem der Lauffolgenkennung zugeordneten Maskenwert.

3. Verfahren nach Anspruch 1, wobei die Auslöserbedingung wenigstens eines von einem diskreten Ereignis oder einem periodischen Ereignis umfasst.

4. Verfahren nach Anspruch 1, wobei der Lauffolgenzeitplan die Mehrzahl von Feldgeräten (114; 214) veranlasst, die Ausführungsfolge mit einer Rate auszuführen, die unabhängig von einer dem Modul (136; 400; 500) zugeordneten Ausführungsrate ist.

5. Computerimplementiertes Verfahren zur Ausführung einer Modullauffolge, umfassend: Identifizieren eines Moduls (136; 400; 500), das eine Mehrzahl von Funktionsblöcken (FB) und eine Lauffolgenkennung umfasst; Bestimmen einer der Lauffolgenkennung zugeordneten Auslöserbedingung; Überwachen eines Prozesssteuersystems auf die Auslöserbedingung; Ausführen der der Lauffolgenkennung zugeordneten Modullauffolge (MLF) basierend auf einem Fall der Auslöserbedingung, in dem die Modullauffolge nur einen Teilsatz der Mehrzahl von Funktionsblöcken aus dem Modul aufrufen soll; und Erzeugen eines Lauffolgenzeitplans, um die erste Lauffolge der Mehrzahl von Feldgeräten (114; 214) zuzuordnen, wobei der Lauffolgenzeitplan dazu

zwingt, dass zu übertragende Daten unabhängig von einer dem Modul zugeordneten Ausführungsrate auf einem Bus (112; 212) einer Gruppe von intelligenten Geräten (114; 214) übertragen werden.

6. Verfahren nach Anspruch 5, wobei das Ausführen der Modullauffolge ferner ein Abrufen eines Lauffolgenzeitplans zum Identifizieren einer Folgen-Ausführungsreihenfolge für den Teilsatz von Funktionsblöcken (FB) umfasst.

7. Verfahren nach Anspruch 6, wobei der Lauffolgenzeitplan ferner ein Synchronisieren mit einem Zyklus zum Erzwingen von Daten an einem Gerätepool mit einer Mehrzahl von Prozessgeräten (114; 214) umfasst, wobei es sich bei dem Zyklus vorzugsweise um einen Makrozyklus handelt, der einer Fieldbus™-Umgebung zugeordnet ist.

8. Vorrichtung zum Handhaben einer Modullauffolge, umfassend:
 einen Lauffolgenverwalter (152; 252) zum Empfangen eines ausgewählten Moduls (136; 400; 500), das eine Mehrzahl von Funktionsblöcken (FB) enthält, wobei das Modul Prozesssteuerroutinen definiert, um verschiedene Funktionen in einem Prozesssteuersystem (100; 200) auszuführen;
 einen Funktionsblockdefinierer (302) zum Empfangen einer Anzeige einer Auswahl eines Teilsatzes der Mehrzahl von Funktionsblöcken, die auszuführen sind;
 einen Zeitplanverwalter (306) zum Empfangen einer Anzeige einer ersten Ausführungsfolge (MLF) für den Teilsatz der Mehrzahl von auszuführenden Funktionsblöcken, wobei die erste Ausführungsfolge für den Teilsatz anders als eine der Mehrzahl von Funktionsblöcken des Moduls zugeordnete zweite Ausführungsfolge des Teilsatzes der Mehrzahl von Funktionsblöcken mit einer Lauffolgenkennung ist;
 einen Auslöserverwalter (304) zum Zuordnen des Teilsatzes zu einer Lauffolgenkennung und einer Auslöserbedingung; und Erzeugen eines Lauffolgenzeitplans, um die erste Lauffolge der Mehrzahl von Feldgeräten zuzuordnen, wobei der Lauffolgenzeitplan dazu zwingt, dass zu übertragende Daten unabhängig von einer dem Modul zugeordneten Ausführungsrate auf einem Bus (112; 212) einer Gruppe von intelligenten Geräten (114; 214) übertragen werden.

9. Vorrichtung nach Anspruch 8, ferner umfassend einen Auslöserüberwacher (310) zum Überwachen eines Prozesssteuersystems auf den Fall des Eintretens der Auslöserbedingung.

10. Vorrichtung nach Anspruch 9, wobei der Auslöserverwalter (304) die Modullauffolge (MLF) identifiziert, die basierend auf der Auslöserbedingung auszuführen ist.

11. Vorrichtung nach Anspruch 8, wobei der Zeitplanverwalter (306) den Teilsatz der Mehrzahl von Funktionsblöcken (FB) identifiziert, der basierend auf der identifizierten Modullauffolge (MLF) auszuführen ist.

Es folgen 11 Seiten Zeichnungen

Anhängende Zeichnungen

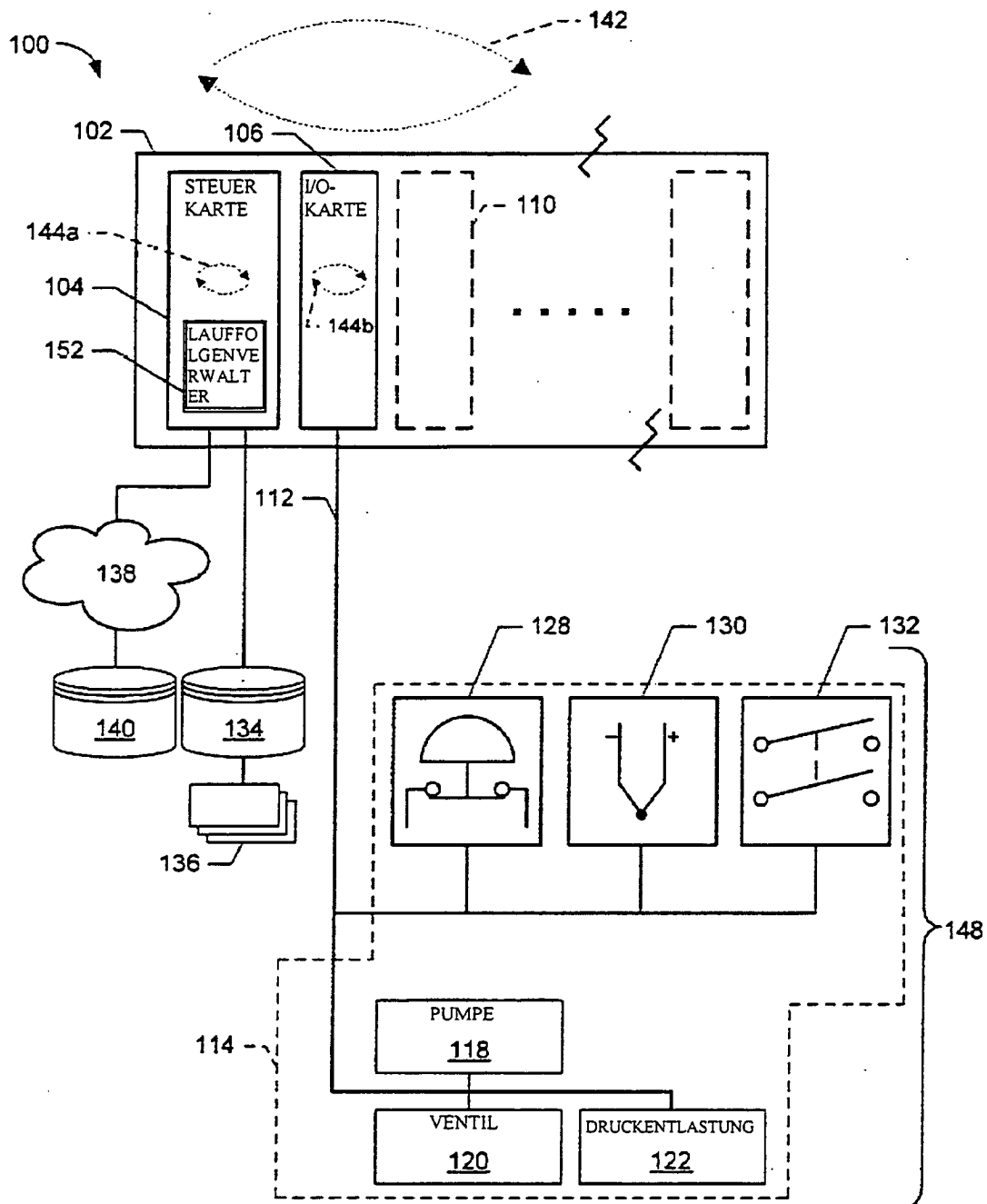


FIG. 1

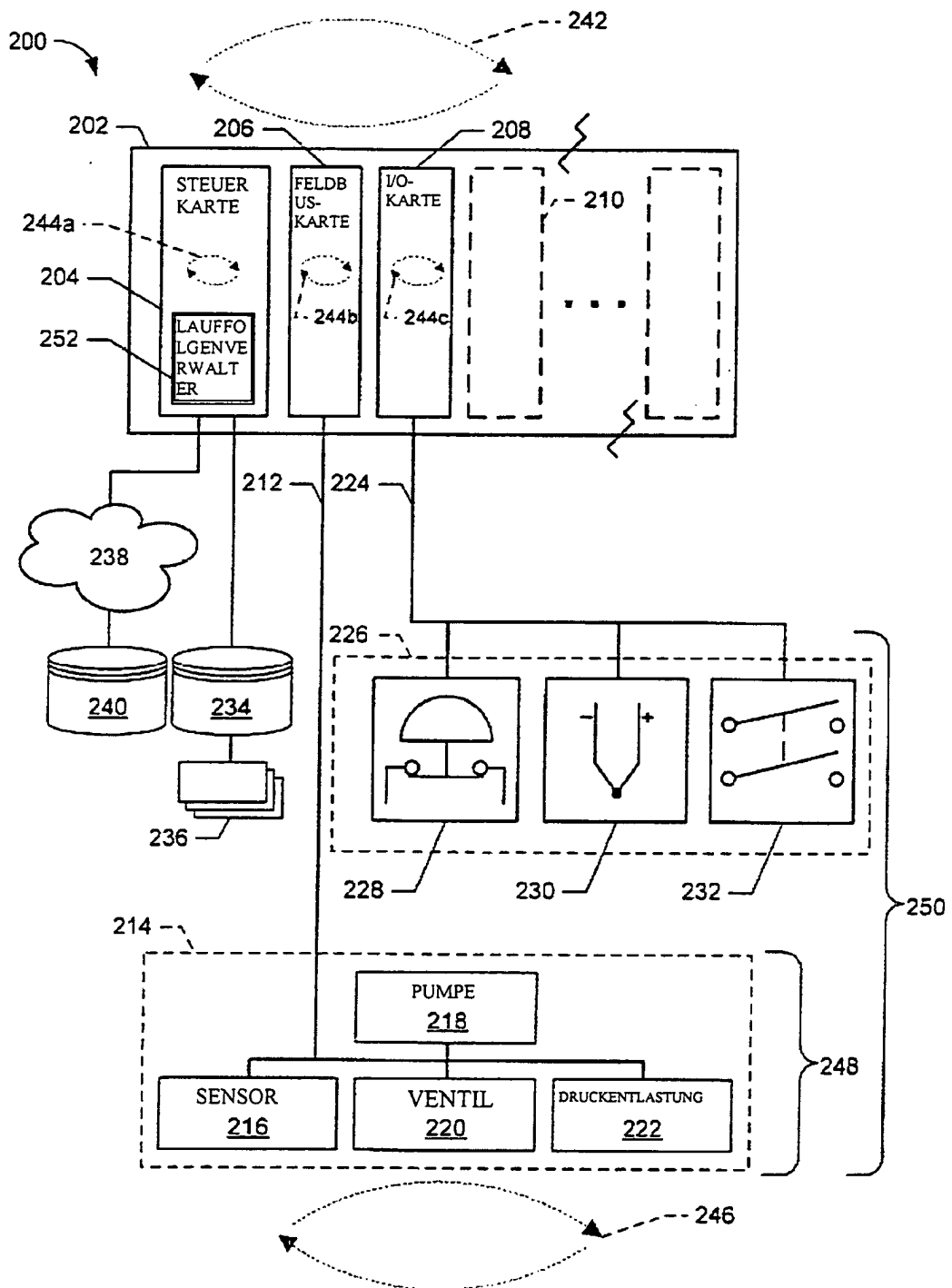


FIG. 2

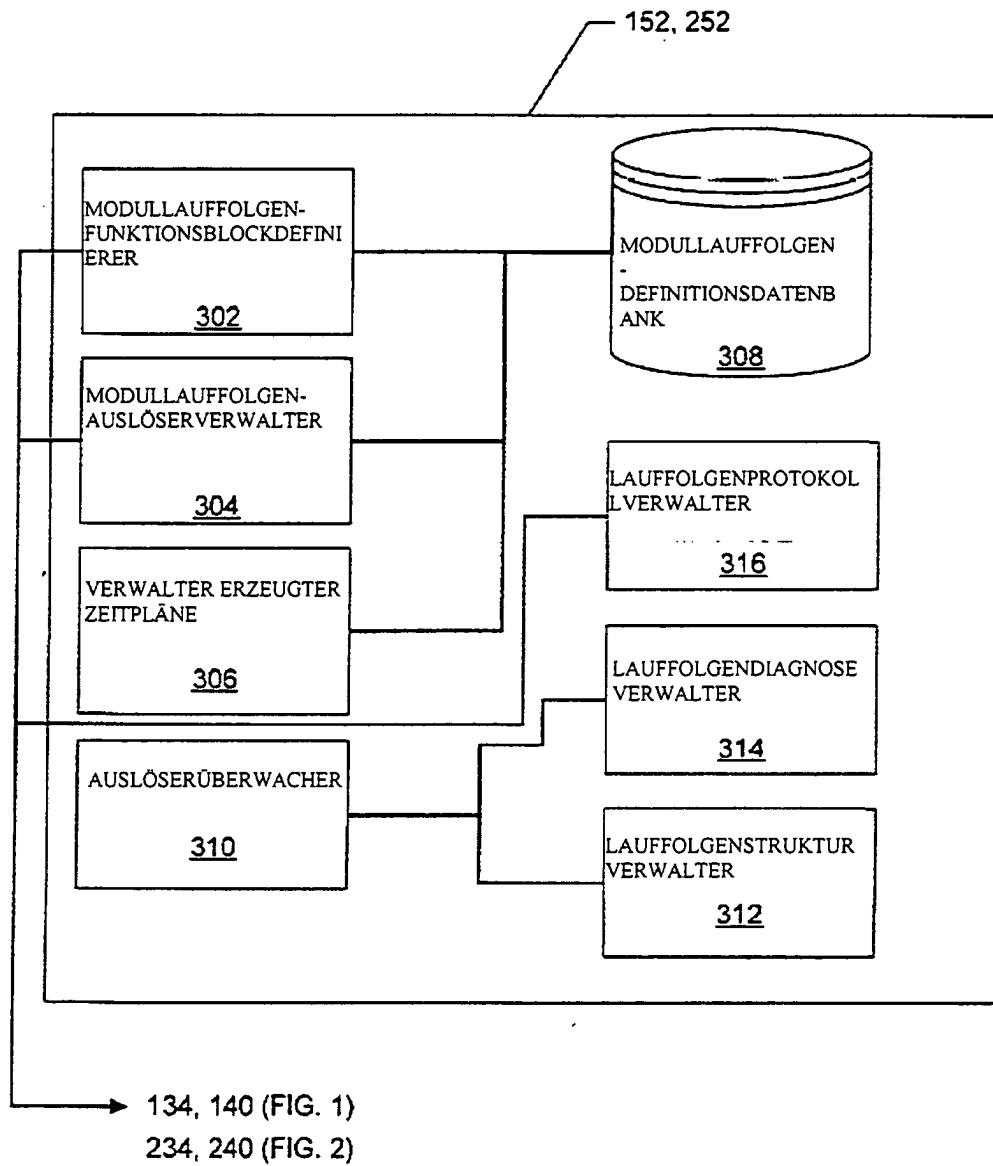


FIG. 3

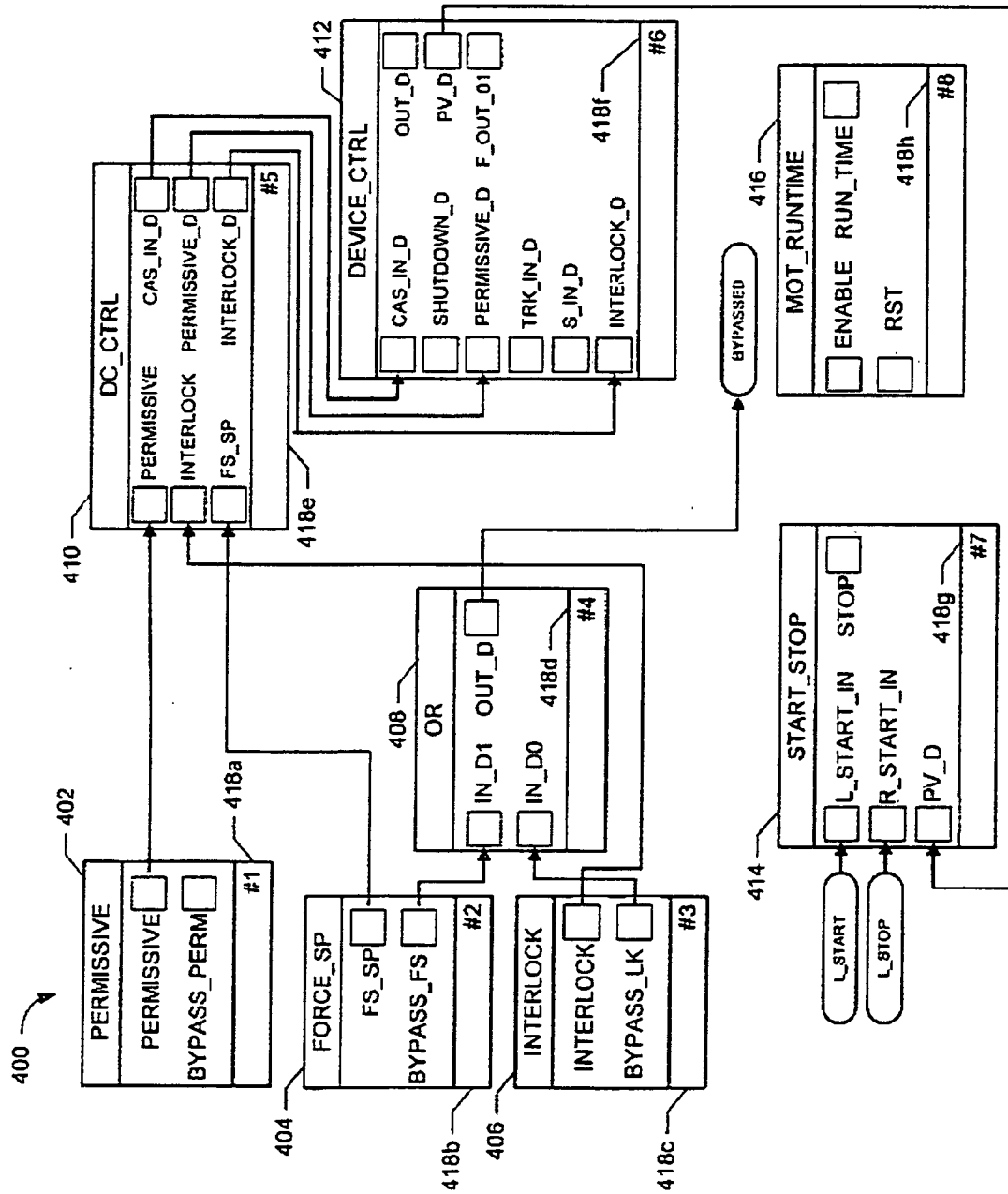


FIG. 4

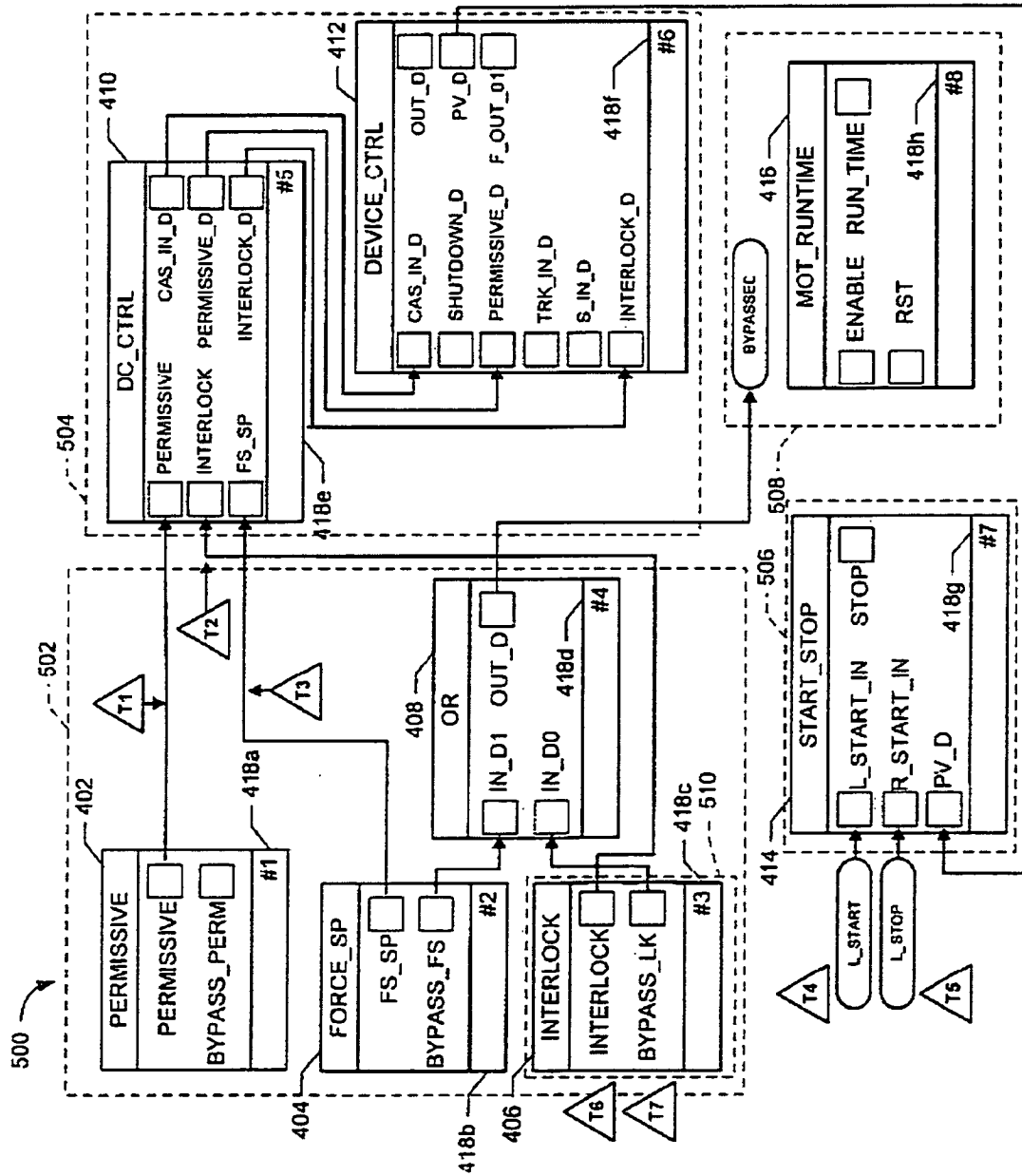


FIG. 5

600

LAUFFOLGENKONFIGURATION FÜR MODUL A

604 ☒ Einstellungen Lauffolge 1 608

606 ☒ Aktiviert ☐ Periodisch

602 Lauffolgenname START_STOP

610 Zugeordnete Funktionsblöcke START_STOP Browse 612

Maskenwert 1101 614

604 ☒ Einstellungen Lauffolge 2 608

606 ☒ Aktiviert ☐ Periodisch

602 Lauffolgenname START_STOP

610 Zugeordnete Funktionsblöcke START_STOP Browse 612

Maskenwert 0111 614

604 ☒ Einstellungen Lauffolge 3 608

606 ☒ Aktiviert ☒ Periodisch

602 Lauffolgenname START_STOP

610 Zugeordnete Funktionsblöcke START_STOP Browse 612

Maskenwert 0100 614

604 ☒ Einstellungen Lauffolge 4 608

606 ☒ Aktiviert ☐ Periodisch

602 Lauffolgenname START_STOP

610 Zugeordnete Funktionsblöcke START_STOP Browse 612

Maskenwert 0001 614

Modullauffolge hinzufügen 618

Modullauffolge löschen 616

MLF auswählen 601

FIG. 6

700

AUSLÖSERKONFIGURATION FÜR MODUL A

Auslöseereinstellungen

Ref.	702	704	706	708	710
L_STOP	Browse	Bedingung: =	Wert: WAHR	Folge: START_STOP	
L_START	Browse	Bedingung: =	Wert: WAHR	Folge: START_STOP	
D_FLOWBAD	Browse	Bedingung: =	Wert: WAHR	Folge: INTERLOCK	
D_LOOPMAN	Browse	Bedingung: =	Wert: WAHR	Folge: INTERLOCK	
PERMISSIVE	Browse	Bedingung: BELIEBIG	Wert: n/a	Folge: MOTOR_CTRL	
INTERLOCK	Browse	Bedingung: BELIEBIG	Wert: n/a	Folge: MOTOR_CTRL	
FORCE_SP	Browse	Bedingung: BELIEBIG	Wert: n/a	Folge: MOTOR_CTRL	
	Browse	Bedingung:	Wert: n/a	Folge:	

FIG. 7

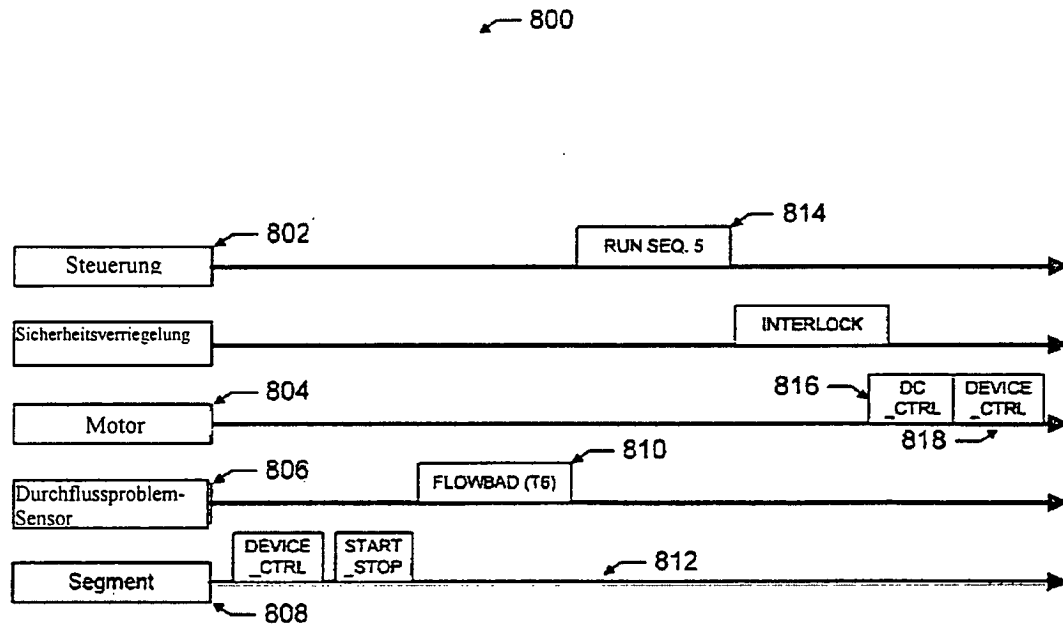


FIG. 8

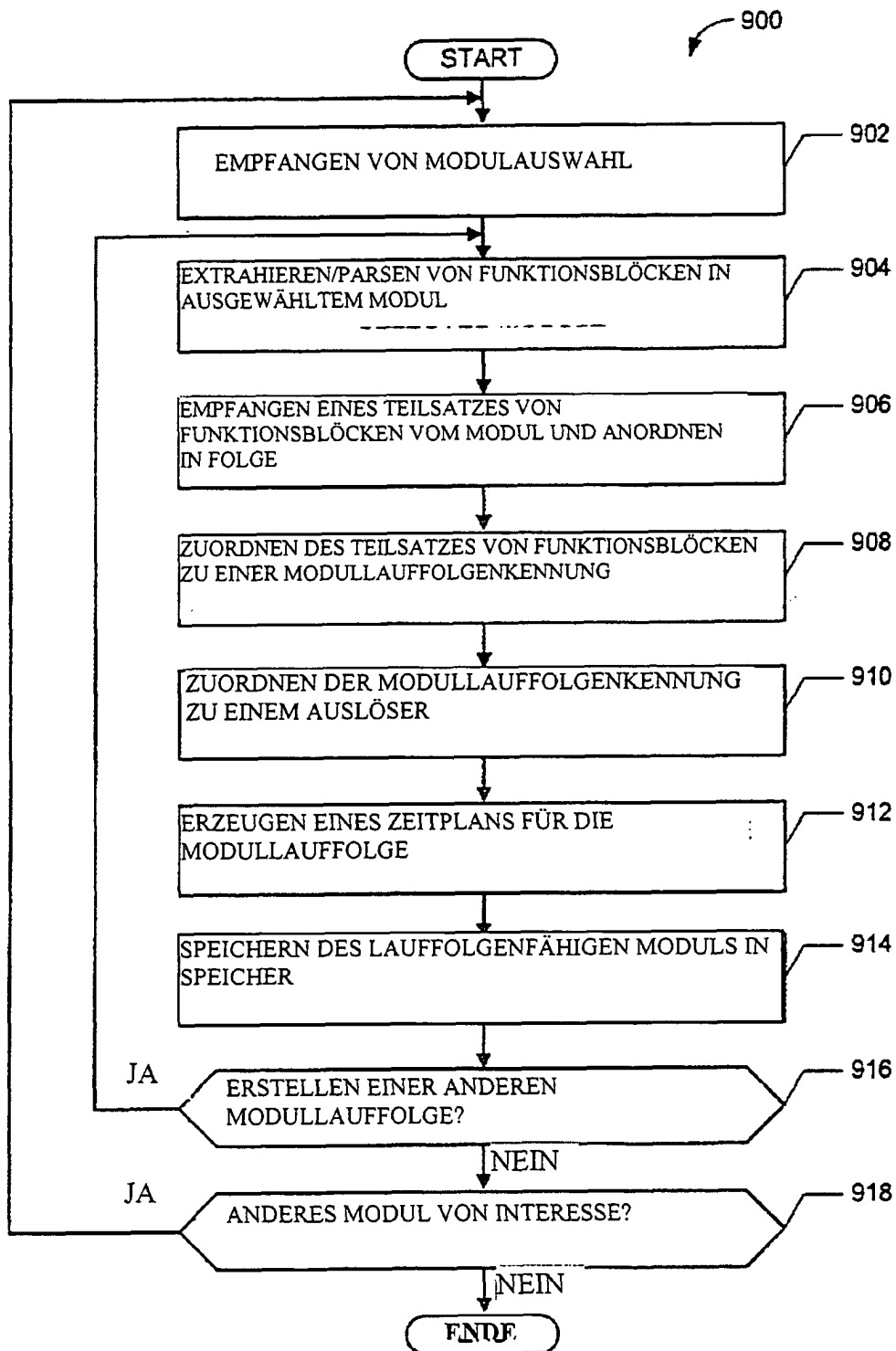


FIG. 9

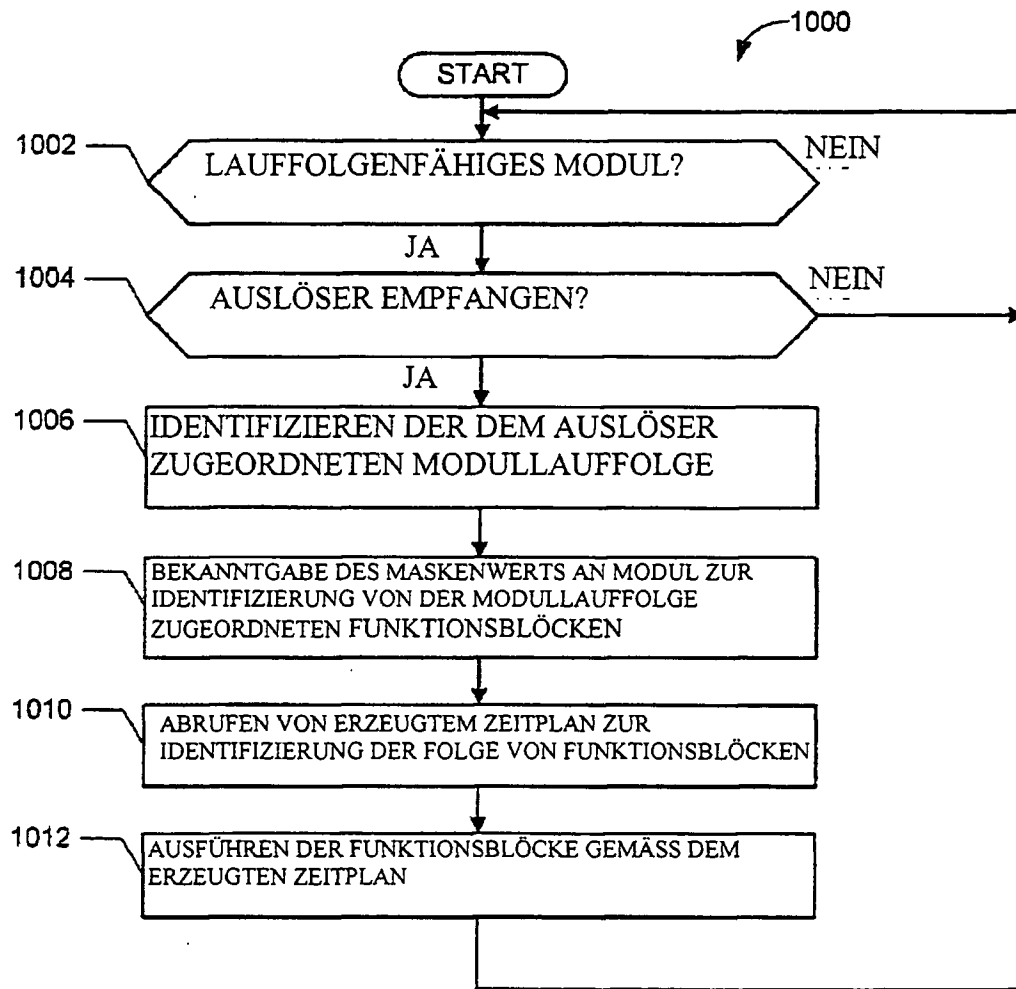


FIG. 10

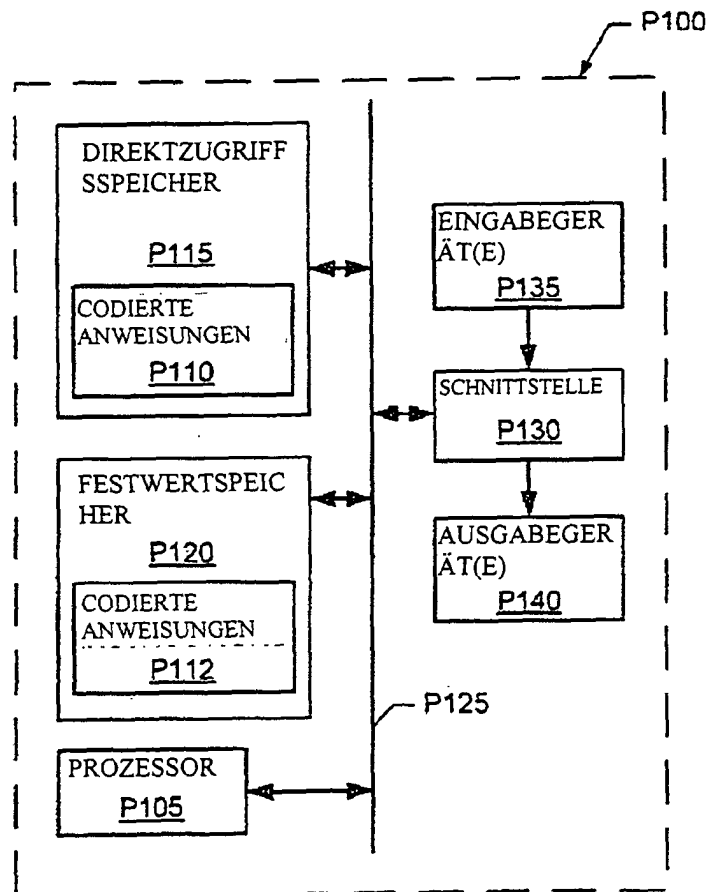


FIG. 11