

(19) 日本国特許庁(JP)

(12) 特 許 公 報(B2)

(11) 特許番号

特許第6571377号  
(P6571377)

(45) 発行日 令和1年9月4日(2019.9.4)

(24) 登録日 令和1年8月16日(2019.8.16)

(51) Int.Cl.		F I			
<b>G05B</b>	<b>19/05</b>	<b>(2006.01)</b>	G05B	19/05	F
<b>G05B</b>	<b>19/404</b>	<b>(2006.01)</b>	G05B	19/05	L
<b>G05D</b>	<b>3/10</b>	<b>(2006.01)</b>	G05B	19/404	L
			G05D	3/10	E

請求項の数 17 (全 23 頁)

(21) 出願番号	特願2015-90769 (P2015-90769)	(73) 特許権者	000129253
(22) 出願日	平成27年4月27日 (2015.4.27)		株式会社キーエンス
(65) 公開番号	特開2016-207088 (P2016-207088A)		大阪府大阪市東淀川区東中島1丁目3番1
(43) 公開日	平成28年12月8日 (2016.12.8)		4号
審査請求日	平成30年2月28日 (2018.2.28)	(74) 代理人	100076428
			弁理士 大塚 康德
		(74) 代理人	100112508
			弁理士 高柳 司郎
		(74) 代理人	100115071
			弁理士 大塚 康弘
		(74) 代理人	100116894
			弁理士 木村 秀二
		(74) 代理人	100131886
			弁理士 坂本 隆志

最終頁に続く

(54) 【発明の名称】 拡張ユニット、プログラマブル・ロジック・コントローラおよびその制御方法

(57) 【特許請求の範囲】

【請求項1】

ユーザプログラムを周期的に実行して外部機器を制御するプログラマブル・ロジック・コントローラであって、

基本ユニットと、

前記基本ユニットと通信バスを介して通信し、前記外部機器に接続された拡張ユニットと

を有し、

前記基本ユニットは、

前記外部機器を制御するための指令値を記憶する第1デバイスメモリと、

前記ユーザプログラムを実行するプログラム実行部と

を有し、

前記拡張ユニットは、

前記第1デバイスメモリに記憶されている指令値と同期した指令値が記憶される第2デバイスメモリと、

前記第2デバイスメモリに記憶された前回の指令値と前記第2デバイスメモリに記憶されている今回の指令値とに基づき前記外部機器の相対的な制御量を演算する指令演算部と、

前記制御量を示す指令信号を生成して前記外部機器に出力する出力部と

を有し、

前記外部機器の位置は前記指令信号にしたがって追従制御され、当該追従制御が開始されたときの前記外部機器の現在位置が前記追従制御の初期位置とされることを特徴とするプログラマブル・ロジック・コントローラ。

【請求項 2】

前記外部機器は前記拡張ユニットによって位置決めされる機器であり、

前記指令信号により伝達される制御量は前記外部機器の現在位置に対する相対的な移動量であることを特徴とする請求項 1 に記載のプログラマブル・ロジック・コントローラ。

【請求項 3】

前記指令値は前記外部機器の絶対的な位置を示していることを特徴とする請求項 1 または 2 に記載のプログラマブル・ロジック・コントローラ。

10

【請求項 4】

前記基本ユニットと通信バスを介して通信し、前記絶対的な位置を指定する指定ユニットをさらに有し、

前記プログラム実行部は前記ユーザプログラムにしたがって前記指定ユニットから取得した前記絶対的な位置を前記指令値として前記第 1 デバイスメモリに記憶させることを特徴とする請求項 3 に記載のプログラマブル・ロジック・コントローラ。

【請求項 5】

前記指定ユニットは、エンコーダから入力される信号をカウントするカウンタユニットであることを特徴とする請求項 4 に記載のプログラマブル・ロジック・コントローラ。

【請求項 6】

20

前記基本ユニットの制御周期と前記指定ユニットの制御周期とを同期させる同期部をさらに有し、

前記プログラム実行部は、前記基本ユニットの制御周期または前記指定ユニットの制御周期に同期して前記第 1 デバイスメモリに記憶されている前記指令値を前記第 2 デバイスメモリにコピーすることを特徴とする請求項 4 または 5 に記載のプログラマブル・ロジック・コントローラ。

【請求項 7】

前記プログラム実行部は、前記ユーザプログラムにしたがって実行する内部演算によって前記指令値を決定することを特徴とする請求項 1 ないし 3 のいずれか 1 項に記載のプログラマブル・ロジック・コントローラ。

30

【請求項 8】

前記プログラム実行部は、前記ユーザプログラムを実行する周期であるスキャンタイムにしたがって前記第 1 デバイスメモリに記憶されている前記指令値を前記第 2 デバイスメモリにコピーすることを特徴とする請求項 1 ないし 7 のいずれか 1 項に記載のプログラマブル・ロジック・コントローラ。

【請求項 9】

前記基本ユニットは、

前記追従制御の開始を示す第 1 リレーデバイスをさらに有し、

前記拡張ユニットは、

前記第 1 リレーデバイスの記憶内容と一致した記憶内容を記憶する第 2 リレーデバイスをさらに有し、

40

前記指令演算部は、前記第 2 リレーデバイスの記憶内容に基づき前記追従制御の開始を認識することを特徴とする請求項 1 ないし 8 のいずれか 1 項に記載のプログラマブル・ロジック・コントローラ。

【請求項 10】

前記出力部は、前記拡張ユニットの制御周期内で前記制御量のすべてを前記外部機器の位置に反映させるよう前記指令信号を生成し、前記拡張ユニットの制御周期と前記制御量とから求められる前記外部機器の移動速度が、前記外部機器について予め定められている上限速度を超える場合は、前記外部機器の移動速度が前記上限速度に一致するように前記制御量を削減することを特徴とする請求項 1 ないし 9 のいずれか 1 項に記載のプログラマ

50

ブル・ロジック・コントローラ。

【請求項 1 1】

前記外部機器の移動を停止させるイベントを検知する検知部をさらに有し、

前記指令演算部は、前記イベントが検知されると、前記外部機器が減速するよう前記制御量を調整することを特徴とする請求項 1 ないし 1 0 のいずれか 1 項に記載のプログラマブル・ロジック・コントローラ。

【請求項 1 2】

基本ユニットと拡張ユニットとを有し、ユーザプログラムを周期的に実行して外部機器を制御するプログラマブル・ロジック・コントローラであって、

前記基本ユニットは、

前記外部機器を制御するための指令値を記憶するデバイスまたはバッファであるメモリと、

前記ユーザプログラムを実行し、当該ユーザプログラムにしたがって前記指令値を更新するプログラム実行部と、

を有し、

前記拡張ユニットは、

前記ユーザプログラムにしたがって前記外部機器についての追従制御が開始されると、前記追従制御が開始されたときの前記外部機器の位置を当該追従制御の初期位置とするとともに、前記ユーザプログラムにしたがって更新される時系列の指令値の変化量に基づき、前記外部機器の位置の制御量を演算する指令演算部と、

前記制御量を示す指令信号を生成して前記外部機器に出力する出力部とを有することを特徴とするプログラマブル・ロジック・コントローラ。

【請求項 1 3】

前記指令演算部は、 $n$  番目に更新された指令値と  $n + 1$  番目に更新された指令値との差分に基づき前記制御量を演算することを特徴とする請求項 1 0 に記載のプログラマブル・ロジック・コントローラ。

【請求項 1 4】

基本ユニットと拡張ユニットとを有し、ユーザプログラムを周期的に実行して外部機器を制御するプログラマブル・ロジック・コントローラであって、

前記基本ユニットは、

前記外部機器を制御するための入力座標系における位置を示す指令値を記憶するデバイスまたはバッファであるメモリと、

前記ユーザプログラムを実行するプログラム実行部と、

を有し、

前記拡張ユニットは、

前記ユーザプログラムにしたがって前記外部機器についての追従制御が開始されると、前記追従制御が開始されたときの前記外部機器の位置を当該追従制御の初期位置とするとともに、前記入力座標系において前記ユーザプログラムにしたがって更新される時系列の指令値の変化量が、前記外部機器の位置を示す出力座標系における当該外部機器の位置の変化量となるように、前記外部機器の位置の制御量を演算する指令演算部と、

前記制御量を示す指令信号を生成して前記外部機器に出力する出力部とを有することを特徴とするプログラマブル・ロジック・コントローラ。

【請求項 1 5】

基本ユニットと、前記基本ユニットと通信バスを介して通信し、外部機器に接続された拡張ユニットとを有し、ユーザプログラムを周期的に実行して外部機器を制御するプログラマブル・ロジック・コントローラの制御方法であって、

前記基本ユニットは、外部機器を制御するための指令値を第 1 デバイスメモリに記憶し、

、

前記拡張ユニットは、前記第 1 デバイスメモリに記憶されている指令値と同期した指令値が記憶される第 2 デバイスメモリに記憶された前回の指令値と前記第 2 デバイスメモリ

10

20

30

40

50

に記憶されている今回の指令値とに基づき前記外部機器の相対的な制御量を演算し、前記制御量を示す指令信号を生成して前記外部機器に出力し、

前記外部機器の位置は前記指令信号にしたがって追従制御され、当該追従制御が開始されたときの前記外部機器の現在位置が前記追従制御の初期位置とされることを特徴とするプログラマブル・ロジック・コントローラの制御方法。

【請求項 16】

ユーザプログラムを周期的に実行して外部機器を制御するプログラマブル・ロジック・コントローラの基本ユニットと通信バスを介して通信し、外部機器に接続された拡張ユニットであって、

前記外部機器を制御するための指令値を記憶する前記基本ユニットの第 1 デバイスメモリに記憶されている指令値と同期した指令値が記憶される第 2 デバイスメモリと、

前記第 2 デバイスメモリに記憶された前回の指令値と前記第 2 デバイスメモリに記憶されている今回の指令値とに基づき前記外部機器の相対的な制御量を演算する指令演算部と、

前記制御量を示す指令信号を生成して前記外部機器に出力する出力部とを有し、

前記外部機器の位置は前記指令信号にしたがって追従制御され、当該追従制御が開始されたときの前記外部機器の現在位置が前記追従制御の初期位置とされることを特徴とする拡張ユニット。

【請求項 17】

ユーザプログラムを周期的に実行して外部機器を制御するプログラマブル・ロジック・コントローラであって、

前記ユーザプログラムを実行するプログラム実行部と、

前記ユーザプログラムの実行結果に基づいて、前記外部機器を制御するための指令値を記憶する第 1 デバイスメモリと、

前記第 1 デバイスメモリに記憶されている指令値と同期した指令値が記憶される第 2 デバイスメモリと、

前記第 2 デバイスメモリに記憶された前回の指令値と前記第 2 デバイスメモリに記憶されている今回の指令値とに基づき前記外部機器の相対的な制御量を演算する指令演算部と、

前記制御量を示す指令信号を生成して前記外部機器に出力する出力部と、を有し、

前記外部機器の位置は前記指令信号にしたがって追従制御され、当該追従制御が開始されたときの前記外部機器の現在位置が前記追従制御の初期位置とされることを特徴とするプログラマブル・ロジック・コントローラ。

【発明の詳細な説明】

【技術分野】

【0001】

本発明は、拡張ユニット、プログラマブル・ロジック・コントローラおよびその制御方法に関する。

【背景技術】

【0002】

特許文献 1 が示すようにプログラマブル・ロジック・コントローラ（PLC）は基本ユニットである CPU ユニットと、拡張ユニットである位置制御ユニットとにより構成され、モータなどの外部機器を制御することがある。外部機器は製品（ワーク）などの物体を所定位置に移動させる位置決め機能を有している。たとえば、操作者によって手動で操作される手動パルサの操作量をカウンタユニットでカウントし、カウント値（目標位置）を位置制御ユニットに転送し、位置制御ユニットは外部機器の現在位置を目標位置へと制御する。

【先行技術文献】

10

20

30

40

50

## 【特許文献】

【0003】

【特許文献1】特開2010-140341号公報

## 【発明の概要】

【発明が解決しようとする課題】

【0004】

このようにCPUユニットは手動パルサなどの入力機器の入力値（CPUユニットが内部で求めた値であってもよい）に応じて外部機器を追従制御することがある。しかし、追従制御においては、PLCがラダープログラムを起動したときに位置制御ユニットの現在位置と目標位置とが大きく乖離している場合に、問題が生じる。つまり、ラダープログラムによる最初の制御周期での外部機器の移動量が極端に大きくなってしまふことがある。たとえば、各制御周期では概ね10レベルずつ外部機器が移動することが想定されている場合に、最初の制御周期だけ100レベルも移動してしまふことがある。このように目標位置を絶対位置として位置制御ユニットに与えてしまふと、問題が生じることがある。そこで、時系列で入力される入力値の差分（相対的な移動量）をラダープログラムにより算出し、相対的な移動量を位置制御ユニットに与えるようにすることが考えられる。しかし、ラダープログラムは各ユーザが作成するものであるため、ユーザの負担が増えてしまふ。

10

【0005】

そこで、本発明は、PLCを用いて追従制御を実行する際のユーザの負担を軽減することを目的とする。

20

【課題を解決するための手段】

【0006】

本発明は、たとえば、ユーザプログラムを周期的に実行して外部機器を制御するプログラマブル・ロジック・コントローラであって、基本ユニットと、前記基本ユニットと通信バスを介して通信し、前記外部機器に接続された拡張ユニットと

を有し、

30

前記基本ユニットは、

前記外部機器を制御するための指令値を記憶する第1デバイスメモリと、

前記ユーザプログラムを実行するプログラム実行部と

を有し、

前記拡張ユニットは、

前記第1デバイスメモリに記憶されている指令値と同期した指令値が記憶される第2デバイスメモリと、

前記第2デバイスメモリに記憶された前回の指令値と前記第2デバイスメモリに記憶されている今回の指令値とに基づき前記外部機器の相対的な制御量を演算する指令演算部と、

40

前記制御量を示す指令信号を生成して前記外部機器に出力する出力部と

を有し、

前記外部機器の位置は前記指令信号にしたがって追従制御され、当該追従制御が開始されたときの前記外部機器の現在位置が前記追従制御の初期位置とされることを特徴とするプログラマブル・ロジック・コントローラが提供される。

【発明の効果】

【0007】

本発明によれば、PLCを用いて追従制御を実行する際のユーザの負担を軽減することが可能となる。

【図面の簡単な説明】

50

## 【 0 0 0 8 】

【図 1】 P L C システムの一例を示す図

【図 2】 ユーザプログラムの一例を示す図

【図 3】 プログラム作成支援装置の一例を示す図

【図 4】 P L C の一例を示す図

【図 5】 スキャンタイムを説明するための図

【図 6】 本実施例の P L C を示す図

【図 7】 本実施例の概念を説明するための図

【図 8】 基本ユニットの機能を示す図

【図 9】 位置決めユニットの機能を示す図

10

【図 10】 カウンタユニットの機能を示す図

【図 11】 基本ユニットの動作を示すフローチャート

【図 12】 追従制御の一例を示すフローチャート

【図 13】 制御量の演算方法を示すフローチャート

【図 14】 追従制御の一例を示す図

【図 15】 一括リフレッシュと同期リフレッシュを説明するための図

【図 16】 同期リフレッシュの一例を示す図

【発明を実施するための形態】

## 【 0 0 0 9 】

以下に本発明の一実施形態を示す。以下で説明される個別の実施形態は、本発明の上位概念、中位概念および下位概念など種々の概念を理解するために役立つであろう。また、本発明の技術的範囲は、特許請求の範囲によって確定されるのであって、以下の個別の実施形態によって限定されるわけではない。

20

## 【 0 0 1 0 】

はじめにプログラマブル・ロジック・コントローラ（ P L C、単にプログラマブルコントローラと呼ばれてもよい）を当業者にとってよりよく理解できるようにするために、一般的な P L C の構成とその動作について説明する。

## 【 0 0 1 1 】

図 1 は、本発明の実施の形態によるプログラマブル・ロジック・コントローラシステムの一構成例を示す概念図である。図 1 に示すように、このシステムは、ラダープログラムなどのユーザプログラムの編集を行うためのプログラム作成支援装置 1 と、工場等に設置される各種制御装置を統括的に制御するための P L C（プログラマブル・ロジック・コントローラ） 2 とを備えている。ユーザプログラムは、ラダー言語やモーションフローなどのグラフィカルプログラミング言語を用いて作成されてもよいし、 C 言語などの高級プログラミング言語を用いて作成されてもよい。以下では、説明の便宜上、ユーザプログラムはラダープログラムとする。 P L C 2 は、 C P U が内蔵された基本ユニット 3 と 1 つないし複数の拡張ユニット 4 を備えている。基本ユニット 3 に対して 1 つないし複数の拡張ユニット 4 が着脱可能となっている。基本ユニット 3 は C P U ユニットと呼ばれることもある。

30

## 【 0 0 1 2 】

基本ユニット 3 には、表示部 5 及び操作部 6 が備えられている。表示部 5 には、基本ユニット 3 に取り付けられている各拡張ユニット 4 の動作状況などを表示することができ、表示部 5 の表示内容は、操作部 6 を操作することにより切り替えることができる。表示部 5 には、通常、 P L C 2 内のデバイスの現在値（デバイス値）や P L C 2 内で生じたエラー情報などが表示される。なお、デバイスとは、デバイス値を格納するために設けられたメモリ上の領域を指す名称であり、デバイスメモリと呼ばれてもよい。デバイス値とは、入力機器からの入力状態、出力機器への出力状態およびユーザプログラム上で設定される内部リレー（補助リレー）、タイマー、カウンタ、データメモリ等の状態を示す情報である。

40

## 【 0 0 1 3 】

50

拡張ユニット4は、PLC2の機能を拡張するために用意されており、基本ユニット3に対して側方から取り付けられる。1つ目の拡張ユニット4は、基本ユニット3に対して側方から直接的に取り付けられる。2つ目以降の拡張ユニット4は、既に取り付けられている拡張ユニット4に対して、側方から直列的に取り付けられる。たとえば、基本ユニット3の右側面と拡張ユニット4の左側面とが連結面になっている。同様に、1つ目の拡張ユニット4の右側面の形状等は基本ユニット3の右側面とほぼ同じであるため、1つ目の拡張ユニット4の右側面に2つ目の拡張ユニット4の左側面が連結される。このような連結方式は、数珠つなぎ方式とかデジチェーン方式と呼ばれてもよい。各連結面にはコネクタが設けられており、通信や電力供給を行うためのバスもコネクタを介して連結される。このようにして、基本ユニット3と複数の拡張ユニット4が直列的に取り付けられると、各拡張ユニット4内に備えられた配線(例:バス)を介して、各拡張ユニット4が基本ユニット3に対して通信可能に接続される。各拡張ユニット4には、その拡張ユニット4の機能に対応する被制御装置16(図4)が接続され、これにより、各被制御装置16が拡張ユニット4を介して基本ユニット3に接続される。被制御装置16には、センサなどの入力装置や、アクチュエータなどの出力装置が含まれる。

10

## 【0014】

プログラム作成支援装置1は、たとえば、携帯可能ないわゆるノートタイプやタブレットタイプのパーソナルコンピュータであって、表示部7及び操作部8を備えている。PLC2を制御するためのユーザプログラムの一例であるラダープログラムは、プログラム作成支援装置1を用いて作成され、その作成されたラダープログラムは、プログラム作成支援装置1内で二モニックコードに変換される。そして、プログラム作成支援装置1を、USB(Universal Serial Bus)などの通信ケーブル9を介してPLC2の基本ユニット3に接続し、二モニックコードに変換されたラダープログラムをプログラム作成支援装置1から基本ユニット3に送ると、そのラダープログラムが基本ユニット3内でマシンコードに変換され、基本ユニット3に備えられたメモリ内に記憶される。なお、ここでは二モニックコードを基本ユニット3に送信するようにしているが、本発明はこれに限られず、例えば二モニックコードを更に中間コードに変換し、中間コードを基本ユニット3に送信するようにしてもよい。

20

## 【0015】

なお、図1では示していないが、プログラム作成支援装置1の操作部8には、プログラム作成支援装置1に接続されたマウスなどのポインティングデバイスが含まれていてもよい。また、プログラム作成支援装置1は、USB以外の他の通信ケーブル9を介して、PLC2の基本ユニット3に対して着脱可能に接続されるような構成であってもよい。

30

## 【0016】

図2は、ラダープログラムの作成時にプログラム作成支援装置1の表示部7に表示されるラダー図17の一例を示す図である。図2に示すように、PLC2を制御するためのラダープログラムは、プログラム作成支援装置1の表示部7にマトリックス状に表示される複数のセル18内に仮想デバイスのシンボル19を適宜配置して、視覚的なリレー回路を表すラダー図17を構築することにより作成される。

## 【0017】

ラダー図17には、たとえば、10列×N行(Nは任意の自然数)のセル18が配置されている。そして、各行のセル18内に、図2に示す左側から右側に向かって、時系列的に仮想デバイスのシンボル19を適宜配置することにより、視覚的なリレー回路を作成することができる。作成されるリレー回路は、1行で表される直列的なリレー回路であってもよいし、複数行に並列的に表されたリレー回路を互いに結合することにより作成された、並列的なリレー回路であってもよい。

40

## 【0018】

図2に示すリレー回路は、入力装置からの入力信号に基づいてオン/オフされる3つの仮想デバイス(以下、「入力デバイス」と呼ぶ。)のシンボル19a, 19b, 19cと、出力装置の動作を制御するためにオン/オフされる仮想デバイス(以下、「出力デバ

50

ス」と呼ぶ。)のシンボル19dとが適宜結合されることにより構成されている。

#### 【0019】

各入力デバイスのシンボル19a, 19b, 19cの上方に表示されている文字(「R0001」、「R0002」及び「R0003」)は、その入力デバイスのデバイス名(アドレス名)21を表している。各入力デバイスのシンボル19a, 19b, 19cの下方に表示されている文字(「フラグ1」、「フラグ2」及び「フラグ3」)は、その入力デバイスに対応付けられたデバイスコメント22を表している。出力デバイスのシンボル19dの上方に表示されている文字(「原点復帰」)は、その出力デバイスの機能を表す文字列からなるラベル23である。

#### 【0020】

図2に示す例では、デバイス名「R0001」及び「R0002」にそれぞれ対応する2つの入力デバイスのシンボル19a, 19bが直列的に結合されることにより、AND回路が構成されている。また、これらの2つの入力デバイスのシンボル19a, 19bからなるAND回路に対して、デバイス名「R0003」に対応する入力デバイスのシンボル19cが並列的に結合されることにより、OR回路が構成されている。すなわち、このリレー回路では、2つのシンボル19a, 19bに対応する入力デバイスがいずれもオンした場合、又は、シンボル19cに対応する入力デバイスがオンした場合にのみ、シンボル19dに対応する出力デバイスがオンされるようになっている。

#### 【0021】

図3は、図1のプログラム作成支援装置1の電気的構成について説明するためのブロック図である。図3に示すように、プログラム作成支援装置1には、CPU24、表示部7、操作部8、記憶装置25及び通信部26が備えられている。表示部7、操作部8、記憶装置25及び通信部26は、それぞれCPU24に対して電気的に接続されている。記憶装置25は、少なくともRAMを含む構成であり、ラダープログラム記憶部25aと、編集ソフト記憶部25bとを備えている。

#### 【0022】

ユーザは、編集ソフト記憶部25bに記憶されている編集ソフトをCPU24に実行させて、操作部8を通じてラダープログラムを編集する。ここで、ラダープログラムの編集には、ラダープログラムの作成及び変更が含まれる。編集ソフトを用いて作成されたラダープログラムは、ラダープログラム記憶部25aに記憶される。また、ユーザは、必要に応じてラダープログラム記憶部25aに記憶されているラダープログラムを読み出し、そのラダープログラムを、編集ソフトを用いて変更することができる。通信部26は、通信ケーブル9を介してプログラム作成支援装置1を基本ユニット3に通信可能に接続するためのものである。

#### 【0023】

図4は、PLC2の電気的構成について説明するためのブロック図である。図4に示すように、基本ユニット3には、CPU10、表示部5、操作部6、記憶装置12及び通信部14が備えられている。表示部5、操作部6、記憶装置12、及び通信部14は、それぞれCPU10に電気的に接続されている。記憶装置12は、RAMやROM、メモ리카ードなどを含んでもよく、ラダープログラムなどを記憶する。記憶装置12には、プログラム作成支援装置1から入力されたラダープログラムやユーザデータが上書きして記憶される。また、記憶装置12には基本ユニット用の制御プログラムも格納されている。図4が示すように基本ユニット3と拡張ユニット4とは拡張バス90の一種であるユニット外部バス90を介して接続されている。なお、ユニット外部バス90に関する通信機能は通信部14の一部として実装されてもよい。

#### 【0024】

図5は、本発明の実施の形態に係るプログラマブルコントローラの基本ユニット3でのスキャンタイムの構成を示す模式図である。図5が示すように1つのスキャンタイムTは、入出力のリフレッシュを行うためのユニット間通信201、プログラム実行202、END処理204により構成されている。ユニット間通信201で、基本ユニット3は、ラ

10

20

30

40

50

ラダープログラムを実行して得られた出力データを基本ユニット3内の記憶装置12から外部機器などに送信するとともに、受信データを含めた入力データを基本ユニット3内の記憶装置12に取り込む。たとえば、基本ユニット3のデバイスに記憶されているデバイス値はリフレッシュによって拡張ユニット4のデバイスに反映される。同様に、拡張ユニット4のデバイスに記憶されているデバイス値はリフレッシュによって基本ユニット3のデバイスに反映される。なお、リフレッシュ以外のタイミングでデバイス値をユニット間で更新する仕組みが採用されてもよい。ただし、基本ユニット3のデバイスは基本ユニット3が随時書き換えており、同様に、拡張ユニット4のデバイスは拡張ユニット4が随時書き換えている。つまり、基本ユニット3のデバイスは基本ユニット3の内部の装置によって随時アクセス可能であり、同様に、拡張ユニット4のデバイスは拡張ユニット4の内部の装置によって随時アクセス可能になっている。基本ユニット3と拡張ユニット4の間では基本的にリフレッシュのタイミングにおいて相互にデバイス値を更新して同期する。プログラム実行202で、基本ユニット3は、更新された入力データを用いてプログラムを実行(演算)する。基本ユニット3はプログラムの実行によりデータを演算処理する。なお、END処理とは、プログラム作成支援装置1や基本ユニット3に接続された表示器(図示せず)等の外部機器とのデータ通信、システムのエラーチェック等の周辺サービスに関する処理全般を意味する。

10

#### 【0025】

このように、プログラム作成支援装置1はユーザの操作に応じたラダープログラムを作成し、作成したラダープログラムをPLC2に転送する。PLC2は、入出力リフレッシュ、ラダープログラムの実行およびEND処理を1サイクル(1スキャン)として、このサイクルを周期的、すなわちサイクリックに繰り返し実行する。これにより、各種入力機器(センサ等)からのタイミング信号に基づいて、各種出力機器(モータ等)を制御する。よって、PLC2は汎用のパーソナルコンピュータ(PC)とは全く異なる動きをする。

20

#### 【0026】

<位置決めユニットを含むPLC>

図6を用いて位置決めユニットを含むPLCの構成例について説明する。図6が示すように基本ユニット3と、拡張ユニット4である位置決めユニット4aおよびカウンタユニット4bは通信バスであるユニット外部バス90を介して接続されている。位置決めユニット4aには外部機器としてモータ45と、モータ45を駆動するドライバ44とが接続されている。モータ45が回転することで、ステージなどワークを搬送する搬送ユニットやワークを保持する保持ユニット、あるいはワーク自体などが位置決めされる。基本ユニット3は、入力データを格納するデバイス31と、出力データを格納するデバイス32と、ユーザプログラムであるラダープログラムを実行するプログラム実行部33とを有している。デバイス31、32はいわゆるデバイスメモリであり、記憶装置12に確保される。デバイス31、32はバッファメモリであってもよいが、ここではデバイスメモリと仮定する。プログラム実行部33はCPU10によって実現される。プログラム実行部33は、たとえば、ラダープログラムにしたがってデバイス31に入力された入力デバイス値に対して演算を実行し、位置決めユニット4aに対する指令値(出力デバイス値)を求め、デバイス32に記憶させる。

30

40

#### 【0027】

位置決めユニット4aは、デバイス41、指令演算部42および出力部43を有している。デバイス31、32がバッファメモリで実現される場合は、デバイス41もバッファメモリで実現される。デバイス41は、基本ユニット3のデバイス32に記憶されている指令値と同期した指令値が記憶されるデバイスメモリである。指令演算部42は、デバイス41に記憶されていた前回の指令値とデバイス41に記憶されている今回の指令値とに基づき外部機器の相対的な制御量を演算する。出力部43は当該制御量を示す指令信号を生成して外部機器に出力する。ドライバ44は指令信号にしたがってモータ45を駆動する。

50

## 【 0 0 2 8 】

カウンタユニット 4 b は入力機器の一例であり、外部機器の絶対的な位置を指定する指定ユニットとして機能する。たとえば、カウンタユニット 4 b は手動パルサなどのエンコーダ 5 3 から信号をカウントするカウント部 5 2 を有し、カウント値をデバイス 5 4 に格納する。デバイス 5 4 に格納されたカウント値は、上述したリフレッシュにより基本ユニット 3 のデバイス 3 1 に格納される。なお、カウンタユニット 4 b はオプションであり、必ずしも本発明に必要ではない。

## 【 0 0 2 9 】

< 外部機器の追従制御の問題点 >

図 7 ( A )、図 7 ( B ) は外部機器の追従制御の問題点を説明するための図である。図 7 ( A ) は手動パルサなどの操作量をカウンタユニットによりカウントした値 ( カウンタ座標 ) を示している。図 7 ( B ) は位置決めユニットの制御量 ( 位置決め座標 ) を示している。図 7 ( A )、図 7 ( B ) が示すように、カウンタユニットの初期位置 ( 例 : 1 1 0 ) と位置決めユニットの初期位置 ( 例 : 1 0 ) は、通常、異なっている。よって、カウンタユニットによりカウントされた操作量をそのまま絶対的な制御量として位置決めユニットに反映させると、最初の制御周期における制御量が極端に大きくなってしまふことがある。これは外部機器を安定して制御する観点からは不都合なことが多い。そこで、本実施例では、追従制御が開始されたときの外部機器の現在位置が追従制御の初期位置とされる場合であっても、最初の制御量が極端に大きくならないように工夫する。

## 【 0 0 3 0 】

図 7 ( C ) はカウンタ座標、追従制御開始フラグ、および、指令信号の関係を示すタイムシーケンスを示している。この例では、追従制御開始フラグが 0 から 1 に変化すると、追従制御が開始される。追従制御が開始されたときのカウンタ座標は前回の指令値であり、1 つ目制御周期でのカウンタ座標は今回の指令値となる。前回の指令値は、位置決めユニット 4 a が備える R A M などに保持される。今回の指令値は、リフレッシュにより、デバイス 5 4 からデバイス 3 1 にコピーされ、デバイス 3 1 からデバイス 3 2 にコピーされ、さらに、デバイス 3 2 からデバイス 4 1 にコピーされ、デバイス 4 1 により保持される。指令演算部 4 2 は、前回の指令値と今回の指令値とに基づき外部機器であるモータ 4 5 の制御量を演算して求める。たとえば、指令演算部 4 2 は、前回の指令値と今回の指令値との差分を制御量として求める。図 7 ( C ) や図 7 ( D ) が示すように、前回の指令値が 1 1 0 であり、今回の指令値が 1 2 0 であれば、差分である 1 0 を示す指令信号が生成される。図 7 ( E ) が示すように、追従制御の初期位置は、追従制御が開始されたときのモータ 4 5 の現在位置 ( モータ 4 5 によって駆動されるステージなどの現在位置 ) が使用される。つまり、前回の指令値と今回の指令値とに基づく制御量は、この初期位置に対する移動量として利用される。

## 【 0 0 3 1 】

図 7 ( B ) と図 7 ( E ) とを比較すると明らかなように、図 7 ( E ) が示す本実施例ではカウンタ座標の初期位置 ( 入力座標系における初期位置 ) と位置決め座標の初期位置 ( 出力座標系における初期位置 ) とが一致していなくても、最初の制御周期における制御量が極端に大きくならないようになる。たとえば、図 7 ( B ) では最初の制御周期における制御量が 1 0 0 であったものが、図 7 ( E ) では 1 0 にまで削減されている。

## 【 0 0 3 2 】

このような前回の指令値と今回の指令値との差分をラダープログラムによって求めることも可能である。しかし、この場合は、各ユーザがラダープログラムに差分演算を実行するための制御コード ( ラダー図 ) を記述しなければならず、ユーザビリティに欠けるだろう。また、この記述をユーザが忘れてしまうと、最初の制御周期における制御量が極端に大きくなってしまふ。一方で本実施例であれば、ユーザが自ら対策コードを記述しなくても、位置決めユニット 4 a の最初の制御周期における制御量が極端に大きくなってしまふことを抑制できる。そのため、ユーザのプログラミングの負担を軽減できるようになる。

## 【 0 0 3 3 】

< 基本ユニットの機能 >

図 8 を用いて基本ユニット 3 の機能について説明する。CPU 10 は不図示の制御プログラムを実行することで、プログラム実行部 33、デバイス制御部 71 および同期部 72 (オプション) として機能する。プログラム実行部 33 は、記憶装置 12 に記憶されているラダープログラム 60 を実行する機能である。プログラム実行部 33 は、演算の高速化のために、他の制御担当する CPU コアとは別の CPU コアにより実現されてもよいし、ASIC (特定用途集積回路) や FPGA (フィールドプログラマブルロジックアレイ) などにより専用の演算ユニットとして実現されてもよい。デバイス制御部 71 は、拡張ユニット 4 が保持しているデバイスの記憶内容と基本ユニット 3 が有しているデバイスの記憶内容とを一致させる機能 (いわゆるリフレッシュ) などを含む。デバイス制御部 71 は、通常、図 5 に示したリフレッシュのタイミングでデバイスの記憶内容を更新する。同期部 72 は拡張ユニット 4 の制御周期と基本ユニット 3 の制御周期とを同期させる機能である。デバイス制御部 71 は、ラダープログラム 60 にしたがって、この制御周期ごとにデバイスを更新してもよい。

10

【 0034 】

記憶装置 12 には出力系のデバイス 32 と入力系のデバイス 31 とが確保されている。出力系のデバイス 32 には、指令値用のデバイス 61、追従制御の開始を拡張ユニット 4 に要求するための開始要求リレー 62 と、その他の情報を記憶するその他のデバイス 63 とが含まれている。なお、入力系のデバイス 31 には、位置制御ユニット 4a が追従制御の開始に成功したことを示すリレーや追従制御を実行中であることを示すリレーが含まれてもよい。なお、リレーとは、一ビットのデータを保持するデバイスの通称である。入力系のデバイス 31 には、カウンタユニット 4b により取得されたカウンタ座標を保持するカウンタ座標デバイス 64 が含まれている。

20

【 0035 】

< 位置決めユニットの機能 >

図 9 を用いて位置決めユニット 4a の機能について説明する。位置決めユニット 4a は、制御プログラム 80 を実行する CPU 46 と、制御プログラム 80 などを記憶する記憶装置 47 と、ユニット外部バス 90 を介して基本ユニット 3 と通信する通信部 50 を有している。なお、CPU 46 の機能は ASIC や FPGA などにより実現されてもよい。記憶装置 47 は、RAM、ROM などを含む。記憶装置 47 には、入力系のデバイス 41、バッファメモリ 48 およびワークメモリ 49 などが確保されている。入力系のデバイス 41 には、指令値用のデバイス 81、基本ユニット 3 からの追従制御の開始を要求するための開始要求リレー 82 と、その他の情報を記憶するその他のデバイス 83 とが含まれている。なお、出力系のデバイスについては図示を省略しているが、これには位置制御ユニット 4a が追従制御の開始に成功したことを示すリレーや追従制御を実行中であることを示すリレーが含まれる。バッファメモリ 48 は、ラダープログラムの実行中に基本ユニット 3 により読み書きされる記憶領域であり、上限速度を記憶する上限速度バッファ 85 と、減速レートを記憶する減速レートバッファ 86 とを含んでいる。ワークメモリ 49 は、CPU 46 が各種の演算をするうえで一時的にデータなどを格納する記憶領域である。この例では、前回の指令値 88 が記憶されているとともに、前回の指令値 88 と指令値用のデバイス 81 に保持されている今回の指令値とから求められた差分 89 が記憶されている。

30

40

【 0036 】

CPU 46 は、制御プログラム 80 を実行することで、デバイス制御部 99、バッファメモリ制御部 91、指令演算部 42、座標管理部 96、出力部 43 として機能する。デバイス制御部 99 は、基本ユニット 3 のデバイス制御部 71 と連携し、拡張ユニット 4 が保持しているデバイスの記憶内容と基本ユニット 3 が有しているデバイスの記憶内容とを一致させる機能である。バッファメモリ制御部 91 は、基本ユニット 3 がラダープログラムにしたがって発行するバッファ書き込みコマンドや読み出しコマンドにしたがって、バッファメモリ 48 にデータを書き込んだり、バッファメモリ 48 からデータを読み出して基本ユニット 3 に転送したりする。

50

## 【 0 0 3 7 】

指令演算部 4 2 は、差分演算部 9 2、制限部 9 3、イベント検知部 9 4 および減速制御部 9 5 を有している。差分演算部 9 2 は、前回の指令値 8 8 と指令値用のデバイス 8 1 に保持されている今回の指令値とから差分 8 9 を演算により求める。制限部 9 3 は、外部機器の移動速度を上限速度以下になるようにクリップ（制限）する。イベント検知部 9 4 は外部機器の移動を停止すべきかどうかを監視する。減速制御部 9 5 は、外部機器を減速させるときの速度の変化を示す制御曲線を演算により求める。座標管理部 9 6 は、外部機器の現在位置（現在座標）を管理する。出力部 4 3 は、モータ 4 5 を制御するためのパルス信号を生成して出力する。通信部 5 0 はユニット外部バス 9 0 を介して基本ユニット 3 と通信する。指令値などのデバイス値はユニット外部バス 9 0 を介して送受信される。

10

## 【 0 0 3 8 】

< カウンタユニットの機能 >

図 1 0 を用いてカウンタユニット 4 b の機能について説明する。カウンタユニット 4 b は他の拡張ユニット 4 と同様で CPU 5 6 と記憶装置 5 7 を有している。なお、CPU 5 6 の機能は ASIC や FPGA などにより実現されてもよい。記憶装置 5 7 は、RAM、ROM などを含む。記憶装置 5 7 は出力系のデバイス 5 1 を有している。出力系のデバイス 5 1 には、カウント部 5 2 のカウント値であるカウンタ座標を保持するカウンタ座標デバイス 7 8 が含まれている。CPU 5 6 は制御プログラム 1 0 0 を実行することで、デバイス制御部 1 0 1 とカウント部 5 2 として機能する。デバイス制御部 1 0 1 は、上述したリフレッシュのタイミングまたは基本ユニット 3 の制御周期と同期したタイミングで、基本ユニット 3 のデバイス制御部 7 1 と連携し、カウンタ座標デバイス 7 8 のカウンタ座標を基本ユニット 3 のカウンタ座標デバイス 6 4 にコピーする。カウント部 5 2 は、エンコーダ 5 3 などの入力機器から入力されるパルスの数をカウントすることで、カウンタ座標（現在座標）を求め、カウンタ座標デバイス 7 8 に書き込む。通信部 5 8 はユニット外部バス 9 0 を介して基本ユニット 3 と通信する。カウンタ座標などのデバイス値はユニット外部バス 9 0 を介して送受信される。

20

## 【 0 0 3 9 】

< 基本ユニットのフローチャート >

図 1 1 を用いて基本ユニット 3 の処理について説明する。この処理は、基本ユニット 3 においてユーザプログラムであるラダープログラム 6 0 の開始が指示されると、CPU 1 0 によって実行される。S 1 で CPU 1 0（デバイス制御部 7 1）は、入力系のデバイスのリフレッシュ（入力リフレッシュ）と出力系のデバイスのリフレッシュ（出力リフレッシュ）を実行する。カウンタユニット 4 b のカウンタ座標と位置決めユニット 4 a への指令値は、一括リフレッシュと同期リフレッシュとのいずれにより更新されてもよいが、ここでは同期リフレッシュにより更新されるものとする。このようにスキャンタイムにおいて一回実行される入出力リフレッシュを一括リフレッシュと呼ぶ。なお、スキャンタイム内で拡張ユニット 4 などの制御周期に同期して複数回実行されるリフレッシュを同期リフレッシュと呼ぶことにする。カウンタ座標や指令値のリフレッシュとしてはこのどちらのタイプのリフレッシュが採用されてもよい。同期リフレッシュは、デバイスメモリだけでなく、バッファメモリに対して適用されてもよいため、カウンタ座標や指令値はバッファメモリに格納されてもよい。

30

40

## 【 0 0 4 0 】

S 2 で CPU 1 0（プログラム実行部 3 3）はラダープログラム 6 0 にしたがって位置決めユニット 4 a に対して追従制御の開始を指示する。たとえば、プログラム実行部 3 3 はデバイス制御部 7 1 に対して開始要求リレー 6 2 が保持しているデバイス値を 0 から 1 に変更するよう指示する。デバイス制御部 7 1 は、この指示に従って開始要求リレー 6 2 が保持しているデバイス値を 1 に変更する。開始要求リレー 6 2 も同期リフレッシュされるものとする。

## 【 0 0 4 1 】

S 3 で CPU 1 0 は位置決めユニット 4 a の制御周期に連動した制御タイミングが到来

50

したかどうかを判定する。CPU 46はタイマーやカウンタを用いて制御周期を管理しており、制御周期に応じて様々な処理を実行する。制御周期が到来すると、CPU 10は、S 4に進む。

【0042】

S 4でCPU 10はデバイス制御部99やバッファメモリ制御部91を介して同期リフレッシュの対象となっているデバイスメモリやバッファメモリの値を更新する。たとえば、CPU 10は、カウンタユニット4bの指令値であるカウンタ座標を基本ユニット3にコピーし、さらにコピーしたカウンタ座標を位置決めユニット4aにコピーする。これにより、カウンタユニット4bのカウンタ座標デバイス78に保持されているカウンタ座標が基本ユニット3のカウンタ座標デバイス64に格納される。また、カウンタ座標デバイス64に格納されたカウンタ座標がそのまま指令値として基本ユニット3の指令値用のデバイス61に格納される。さらに、指令値用のデバイス61に格納された指令値が位置決めユニット4aの指令値用のデバイス81に格納される。

10

【0043】

S 5でCPU 10(プログラム実行部33)はラダープログラム60にしたがって追従制御の終了条件が成立したかどうかを判定する。追従制御の終了条件が成立していなければ、S 3に戻り、ラダープログラム60にしたがった処理を継続して実行する。一方で、追従制御の終了条件が成立していれば、CPU 10はS 6に進む。たとえば、追従制御の終了条件は、実行中リレーが0になったことであってもよい。実行中リレーは、位置決めユニット4aにおいて追従制御が実行中かどうかを示すリレーデバイスであり、1(オン)は追従制御を実行中であることを示し、0(オフ)は追従制御を終了したことを示す。

20

【0044】

S 6でCPU 10(プログラム実行部33)はラダープログラム60にしたがって位置決めユニット4aに対して追従制御の終了を指示する。たとえば、プログラム実行部33はデバイス制御部71に対して開始要求リレー62が保持しているデバイス値を1から0に変更するよう指示する。デバイス制御部71は、この指示に従って開始要求リレー62が保持しているデバイス値を0に変更する。その後、CPU 10はエンド処理を実行してS 1に戻り、次のスキャンタイムに移行する。

【0045】

このように前回の指令値と今回の指令値との差分を求めるなどの処理は位置決めユニット4aで実行されるため、ラダープログラム60には差分を求める処理が不要となる。つまり、本実施例は、ユーザのプログラミングの負担を軽減できる。

30

【0046】

<位置決めユニットのフローチャート>

図12を用いて位置決めユニット4aの処理について説明する。S 11でCPU 46は基本ユニット3によって追従制御を開始するよう要求されているかどうかを判定する。たとえば、CPU 46は開始要求リレー82のデバイス値を取得し、その値が1になっているかどうかを判定する。デバイス値が0から1に変化すると、基本ユニット3によって追従制御を開始するよう要求されたと判定し、CPU 46はS 12に進む。S 12でCPU 46は追従制御の開始時に指令値用のデバイス81に記憶されているデバイス値(指令値)を読み出し、ワークメモリ49に前回の指令値88として保存する。

40

【0047】

S 13でCPU 46は位置決めユニット4aの制御周期に連動した制御タイミングが到来したかどうかを判定する。CPU 46はタイマーやカウンタを用いて制御周期を管理しており、制御周期に応じて様々な処理を実行する。なお、基本ユニット3、位置決めユニット4aおよびカウンタユニット4bの制御周期はユニット外部バス90を通じて同期しているものとする。

【0048】

S 14でCPU 46(指令演算部42)は指令値用のデバイス81に記憶されている今回の指令値を読み出すとともに、ワークメモリ49から前回の指令値88を読み出し、こ

50

れらに基づいて制御量を演算して求める。S 1 5でCPU 4 6（出力部 4 3）は制御量に応じた指令信号を生成して外部機器に出力する。たとえば、出力部 4 3は、制御量を制御周期で除算して移動速度を求め、移動速度に応じた周波数のパルスを生成して出力する。外部機器の移動速度とモータを駆動するためのパルス（指令信号）の周波数は比例している。このように、一つの制御周期の間に制御量のすべてが外部機器の位置に反映される。座標管理部 9 6は、外部機器の現在位置に制御量を加算することで現在位置のデータ（座標）を更新する。

【 0 0 4 9 】

S 1 6でCPU 4 6（指令演算部 4 2）は今回の指令値を前回の指令値 8 8としてワークメモリ 4 9に保存する。S 1 7でCPU 4 6は追従制御の終了条件が満たされたかどうかを判定する。終了条件は、たとえば、外部機器の移動速度が 0 になったことでもよい。終了条件が満たされると、CPU 4 6は実行中リレーを 0 に書き換え、本処理を終了する。終了条件が満たされていないならば、CPU 4 6はS 1 3に戻る。

10

【 0 0 5 0 】

< 制御量の求め方の詳細 >

図 1 3を用いて制御量の決定方法について説明する。ここでは停止イベントが発生したときの減速制御と、移動速度を上限速度に制限する制御（クリップ）とを含めて説明するが、これらはオプションである。

【 0 0 5 1 】

S 2 1でCPU 4 6（イベント検知部 9 4）は外部機器の移動を停止すべき所定のイベント（停止イベント）が生じているかどうかを判定する。たとえば、イベント検知部 9 4は外部機器の位置データや外部機器の位置を検知するセンサの検知結果に基づいて外部機器の位置が所定範囲外になっているかどうかを判定する。外部機器の位置が所定範囲外になると、イベント検知部 9 4は停止イベントが発生したと判定する。なお、イベント検知部 9 4は停止イベントが発生したことを基本ユニット 3 に伝達するために所定のリレーの値を 1 に設定してもよい。停止イベントが発生していないならば、CPU 4 6はS 2 2に進む。

20

【 0 0 5 2 】

S 2 2でCPU 4 6（差分演算部 9 2）は前回の指令値と今回の指令値との差分を求め、ワークメモリ 4 9に記憶する。S 2 3でCPU 4 6（差分演算部 9 2）は、前回の指令値と今回の指令値とから求められた差分と、制御周期とから移動速度を求める。移動速度は、たとえば、差分を制御周期で除算することで容易に求められる。

30

【 0 0 5 3 】

S 2 4でCPU 4 6（制限部 9 3）は、バッファメモリ 4 8から上限速度を読み出し、移動速度が上限速度を超えているかどうかを判定する。移動速度が上限速度を超えていなければ、CPU 4 6はS 2 5をスキップし、S 1 5に進む。一方で、移動速度が上限速度を超えていれば、CPU 4 6はS 2 5に進む。S 2 5でCPU 4 6（制限部 9 3）は移動速度を上限速度にクリップし、制御量を補正する。たとえば、制限部 9 3は移動速度として上限速度を採用し、上限速度に制御周期を乗算することで制御量を求め、求めた制御量を今回の制御量としてワークメモリ 4 9に保存する。

40

【 0 0 5 4 】

S 2 1で停止イベントが検知されると、外部機器を停止させるべく、CPU 4 6はS 2 6に進む。S 2 6でCPU 4 6（減速制御部 9 5）は、ワークメモリ 4 9から前回の移動速度を読み出すとともに、バッファメモリ 4 8から減速レートを読み出し、前回の移動速度と減速レートに基づき今回の移動速度を求める。たとえば、減速制御部 9 5は、前回の移動速度に減速レートを乗算し、今回の移動速度を求める。減速レートは、たとえば、1 . 0 未満の係数である。なお、減速レートは、所定の減速曲線を表す関数に基づき、減速制御部 9 5によって動的に算出されてもよい。S 2 7でCPU 4 6（減速制御部 9 5）は、移動速度が 0 になったかどうかを判定する。移動速度が 0 であれば、CPU 4 6は、追従制御を終了すべく、図 1 2 に示したS 1 7に進む。一方で、移動速度が 0 でなければ、

50

C P U 4 6 は、追従制御を継続するために、図 1 2 に示した S 1 5 に進む。

【 0 0 5 5 】

< 追従制御のタイミングチャートの一例 >

図 1 4 は追従制御のタイミングチャートの一例を示す図である。この例では、時刻  $t_1$  に開始要求リレー 8 2 が 1 (オン) になり、追従制御が開始されている。また、この例では開始要求リレー 8 2 に加え、追従制御が実行中であることを示す実行中リレーと、追従制御を開始したことを報告するための開始報告リレーが用いられている。時刻  $t_n$  から時刻  $t_{n+1}$  までの期間が制御周期を表している。

【 0 0 5 6 】

C P U 4 6 は時刻  $t_1$  で開始要求リレー 8 2 が 1 になったことを認識する ( S 1 1 ) 。  
C P U 4 6 は時刻  $t_2$  で実行中リレーを 1 にセットするとともに開始報告リレーも 1 にセットする。さらに、C P U 4 6 は現在の指令値を前回の指令値として記憶する ( S 1 2 ) 。  
C P U 4 6 は時刻  $t_3$  以降で、前回の指令値と今回の指令値との差分 ( 相対的な制御量 ) を求めて外部機器を追従制御する ( S 1 3 ~ S 1 7 ) 。制御量に応じて外部機器の移動速度と位置が変化する。この例では、時刻  $t_{10}$  で停止イベントが発生している。C P U 4 6 は、停止イベントが発生したため、外部機器の減速制御を実行する ( S 2 1、S 2 6 ~ S 2 8 ) 。減速制御により移動速度が徐々に減少し、位置の変化量も徐々に小さくなっている。時刻  $t_{13}$  では移動速度が 0 となり、外部機器が停止する ( S 2 7 ) 。移動速度が 0 となり、追従制御の終了条件が満たされたため、C P U 4 6 は実行中リレーを 0 に変更し、基本ユニット 3 に追従制御が終了したことを伝達している ( S 1 7 ) 。

【 0 0 5 7 】

< ユニット間同期 >

スキャンタイムは一般に数ミリ秒であるが、位置決めユニット 4 a ではより短い周期で外部機器の位置を制御したいという要請がある。しかし、1つのスキャンタイムで入出力デバイスのリフレッシュ (一括リフレッシュ) は 1 回だけである。そこで、本実施例では上述した同期リフレッシュを採用している。

【 0 0 5 8 】

図 1 5 は一括リフレッシュと同期リフレッシュとの関係を示す図である。図 1 5 が示すように、一括リフレッシュにより基本ユニット 3 と拡張ユニット 4 との間で入出力デバイスのデバイス値が更新される。たとえば、カウンタユニット 4 b のカウンタ座標は 1 つのスキャンタイムで一回だけ更新され、位置決めユニット 4 a への指令値も 1 つのスキャンタイムで一回だけ更新される。しかし、これではより高速な追従制御を実現することができない。そこで、基本ユニット 3 の内部制御周期と拡張ユニット 4 の内部制御周期とを同期させ、この制御周期ごとにデバイスやバッファなどをリフレッシュする同期リフレッシュを本実施例では採用する。なお、ラダープログラムのうち、この制御周期に同期して実行されるプログラムはユニット間同期プログラムと呼ばれる。図 1 5 において同期リフレッシュは、ラダープログラムの実行期間だけでなく、一括リフレッシュの実行期間やエンド処理の実行期間においても実行される。

【 0 0 5 9 】

図 1 6 は同期リフレッシュの一例を示す図である。基本ユニット 3 のデバイス制御部 7 1 は、制御周期ごとに同期リフレッシュを実行する。図 1 6 によれば、制御周期内でまず同期リフレッシュの対象として予め指定されているデバイスであるカウンタユニット 4 b のカウンタ座標  $X_n$  がリフレッシュされる。次に、デバイス制御部 7 1 は、カウンタ座標  $X_n$  をそのまま  $Y_{n+1}$  として指令値用のデバイス 6 1 に格納し、これを位置決めユニット 4 a の指令値用のデバイス 8 1 にコピーする。位置決めユニット 4 a は、指令値用のデバイス 8 1 が保持しているデバイス値を今回の指令値として内部処理 (追従制御) を実行する。以下、同様に制御周期ごとに同期リフレッシュが実行される。

【 0 0 6 0 】

このように同期リフレッシュを採用すると、位置決めユニット 4 a が内部処理を開始する時点では必ず最新の指令値を基本ユニット 3 から取得できるようになる。これにより、

10

20

30

40

50

高速かつきめ細やかな追従制御が実現可能となる。

【 0 0 6 1 】

<まとめ>

以上で説明したように、本実施例は、基本ユニット3と拡張ユニット4とを備え、ユーザプログラムを周期的に実行して外部機器を制御するPLC2に関するものである。基本ユニット3と拡張ユニット4は、通信バス(拡張バス)であるユニット外部バス90を介して通信し、外部機器(被制御装置16)を制御する。

【 0 0 6 2 】

図6を用いて説明したように、基本ユニット3は、外部機器を制御するための指令値を記憶する第1デバイスメモリとしてデバイス32を有している。また、基本ユニット3は、ユーザプログラムを実行し、当該ユーザプログラムにしたがって指令値を更新するプログラム実行部33を有している。拡張ユニット4は、デバイス32に記憶されている指令値と同期した指令値が記憶される第2デバイスメモリであるデバイス41を有している。拡張ユニット4の指令演算部42は、デバイス41に記憶されていた前回の指令値とデバイス41に記憶されている今回の指令値とに基づき外部機器の相対的な制御量を演算する。出力部43は、当該制御量を示す指令信号を生成して、外部機器に出力する。とりわけ、図7(E)を用いて説明したように、本実施例では、外部機器の位置は指令信号にしたがって追従制御され、追従制御が開始されたときの外部機器の現在位置が追従制御の初期位置とされる。また、拡張ユニット4の指令演算部42が相対的な制御量を演算するため、ユーザは、相対的な制御量を求めるためのプログラムをラダープログラム60内に記述する手間を省けるようになる。つまり、PLC2を用いて追従制御を実行する際のユーザの負担が軽減される。なお、位置決めユニット4aは、デバイス41とは別に、外部機器インターフェース用のデバイス41aを有している。このデバイス41aには、外部機器の現在位置が格納され、適宜、基本ユニット3(プログラム実行部33)から参照可能となっている。すなわち、指令演算部42は、デバイス41に記憶されている指令値に基づいて、外部機器の相対的な制御量を演算した後に、追従制御を実現するために制御量を出力部43に渡す一方、制御量に基づいてデバイス41aに格納された外部機器の現在位置を更新する。

【 0 0 6 3 】

図6を用いて説明したように、外部機器は拡張ユニット4によって位置決めされるモータ45などの機器であってもよい。この場合、出力部43が出力する指令信号により伝達される制御量はモータ45の現在位置(回転角度など)またはモータ45によって駆動されるステージなどの現在位置に対する相対的な移動量となる。図7(B)を用いて説明したように、絶対的な移動量をそのまま制御量としてしまうと、追従制御の開始時における移動量が極端に大きくなってしまう。しかし、相対的な移動量を位置決めユニット4aなどの拡張ユニット4が求めるため、追従制御の開始時における移動量が極端に大きくなるようなりになる。

【 0 0 6 4 】

基本ユニット3が拡張ユニット4に送信する指令値は外部機器の絶対的な位置を示してもよい。図7(B)を用いて説明したように、絶対的な位置を指令値とすると、追従制御の開始時における移動量が極端に大きくなってしまふことがある。このようケースであっても本実施例では、相対的な移動量を位置決めユニット4aなどの拡張ユニット4が求めるため、追従制御の開始時における移動量が極端に大きくなるようなりになる。

【 0 0 6 5 】

図6を用いて説明したように、カウンタユニット4bは、基本ユニット3とユニット外部バス90を介して通信し、絶対的な位置を指定する指定ユニットとして機能してもよい。プログラム実行部33はラダープログラム60にしたがってカウンタユニット4bから取得した絶対的な位置を指令値としてデバイス32に記憶させる。このようにカウンタユニット4bのカウンタ座標をそのまま指令値として位置決めユニット4aに渡されることになるため、ユーザは指令値を操作するようなラダープログラムを記述する手間を省ける

10

20

30

40

50

。なお、指令値に対して多少の演算が施されてから位置決めユニット 4 a に渡されてもよい。

【 0 0 6 6 】

このように、指定ユニットは、エンコーダ 5 3 から入力される信号をカウントするカウンタユニット 4 b であってもよいが、エンコーダ 5 3 はたとえば手動パルサなどであってもよい。また、指定ユニットは、アナログ / デジタル変換ユニットなど、入力信号をデジタル値に変換するユニットであってもよい。また、指定ユニットは、基本ユニット 3 であってもよい。基本ユニット 3 は、たとえばラダープログラム 6 0 を実行することで指令値を求めてもよい。つまり、基本ユニット 3 のプログラム実行部 3 3 は、ユーザプログラムにしたがって実行する内部演算によって指令値を決定してもよい。

10

【 0 0 6 7 】

図 8 に示したように、基本ユニット 3 の制御周期と拡張ユニット 4 との制御周期とを同期させる同期部 7 2 がさらに設けられてもよい。図 1 6 を用いて説明したように、プログラム実行部 3 3 は、デバイス制御部 7 1 を通じて、基本ユニット 3 の制御周期または拡張ユニット 4 の制御周期に同期してデバイス 3 2 に記憶されている指令値を位置決めユニット 4 a のデバイス 4 1 にコピーしてもよい。同様に、プログラム実行部 3 3 は、デバイス制御部 7 1 を通じて、基本ユニット 3 の制御周期または拡張ユニット 4 の制御周期に同期してカウンタユニット 4 b のデバイス 5 4 から基本ユニット 3 のデバイス 3 1 にコピーしてもよい。このような同期リフレッシュを採用すれば、位置決めユニット 4 a の内部処理（追従制御）を実行する際には最新の指令値を入手できるようになる。

20

【 0 0 6 8 】

プログラム実行部 3 3 は、ユーザプログラムを実行する周期であるスキャンタイムにしたがってデバイス 3 2 に記憶されている指令値を位置決めユニット 4 a のデバイス 4 1 にコピーしてもよい。高速かつ精緻な制御が不要なケースでは、指令値を一括リフレッシュにより更新してもよい。同期リフレッシュは、一括リフレッシュと比較して実行回数が多いため、基本ユニット 3 や拡張ユニット 4 に対する処理負荷が重くなるだろう。よって、一括リフレッシュを採用すれば処理負荷が軽減されよう。

【 0 0 6 9 】

基本ユニット 3 は、追従制御の開始を示す第 1 リレーデバイスである開始要求リレー 6 2 をさらに有していてもよい。この場合、位置決めユニット 4 a は、開始要求リレー 6 2 の記憶内容と一致した記憶内容を記憶する第 2 リレーデバイスとして開始要求リレー 8 2 を有することになる。指令演算部 4 2 は、開始要求リレー 8 2 の記憶内容に基づき追従制御の開始を認識する。これにより、位置決めユニット 4 a は明示的に追従制御の開始を認識可能となり、追従制御の開始時の指令値を基準値（前回の指令値）として確実に保持できるようになる。

30

【 0 0 7 0 】

指令演算部 4 2 は、拡張ユニット 4 の制御周期内で制御量のすべてを外部機器の位置に反映させるよう出力部 4 3 に指令信号を生成させる。S 2 4、S 2 5 に関して説明したように、拡張ユニット 4 の制御周期と制御量とから求められる外部機器の移動速度が外部機器について予め定められている上限速度を超える場合、指令演算部 4 2 は、外部機器の移動速度が上限速度に一致するように制御量を削減してもよい。これにより、外部機器の移動速度を上限速度以下に制御可能となる。これは、外部機器などの耐久性を高めるために要求されてもよい。

40

【 0 0 7 1 】

S 2 1 に関して説明したように、基本ユニット 3 は、外部機器の移動を停止させるべきイベントを検知するイベント検知部 9 4 を有していてもよい。指令演算部 4 2 は、このような停止イベントが検知されると、外部機器が減速するよう制御量を調整してもよい。これにより外部機器を安全に停止させることが可能となる。

【 0 0 7 2 】

指令演算部 4 2 の動作は次のような概念として理解されてもよい。指令演算部 4 2 は、

50

ユーザプログラムにしたがって外部機器についての追従制御が開始されると、追従制御が開始されたときの外部機器の位置を当該追従制御の初期位置とするとともに、ユーザプログラムにしたがって更新される時系列の指令値の変化量に基づき、外部機器の位置の制御量を演算する。この場合、指令演算部 4 2 は、 $n$  番目に更新された指令値（前回の指令値）と  $n + 1$  番目に更新された指令値（今回の指令値）との差分に基づき制御量を演算することになる。

【 0 0 7 3 】

指令値やカウンタ座標はデバイスメモリではなくバッファメモリによって保持されてもよい。つまり、基本ユニット 3 や拡張ユニット 4 は外部機器を制御するための入力座標系における位置を示す指令値を記憶するデバイスまたはバッファであるメモリを有しているもよい。プログラム実行部 3 3 はユーザプログラムを実行し、当該ユーザプログラムにしたがってメモリに記憶される指令値を更新してもよい。拡張ユニット 4 の指令演算部 4 2 は、入力座標系（カウンタユニット 4 b の座標系）においてユーザプログラムにしたがって更新される時系列の指令値の変化量が、外部機器の位置を示す出力座標系（位置決めユニット 4 a の座標系）における外部機器の位置の変化量となるように、外部機器の位置の制御量を演算してもよい。これにより、ユーザは、相対的な制御量を求めるためのプログラムをラダープログラム 6 0 内に記述する手間を省けるようになる。つまり、PLC 2 を用いて追従制御を実行する際のユーザの負担が軽減される。

【 0 0 7 4 】

上述した実施例では基本ユニット 3、位置決めユニット 4 a およびカウンタユニット 4 b はそれぞれ独立した筐体に收容されているものとして説明したが、これらの 3 つが 1 つの筐体に收容されていてもよい。たとえば、基本ユニット 3 に位置決めユニット 4 a およびカウンタユニット 4 b が設けられていてもよい。また、基本ユニット 3 と位置決めユニット 4 a とが 1 つのユニットを形成し、カウンタユニット 4 b が独立したユニットであってもよい。基本ユニット 3 とカウンタユニット 4 b とが 1 つのユニットを形成し、位置決めユニット 4 a が独立したユニットであってもよい。さらに、位置決めユニット 4 a とカウンタユニット 4 b とが 1 つのユニットを形成し、基本ユニット 3 が独立したユニットであってもよい。いずれの場合であってもユーザプログラムにおいて相対的な制御量を求める必要がなくなるため、ユーザビリティが向上しよう。

上述した実施例では基本ユニット 3 と拡張ユニット 4 とがそれぞれ別の筐体に收容され、バスを介して接続されるものとして説明した。しかし、拡張ユニット 4 の機能は基本ユニット 3 に搭載されてもよい。たとえば、被制御装置 1 6、ドライバ 4 4、モータ 4 5 などの外部機器を直接的に基本ユニット 3 へ接続することが可能となる。この場合に、カウンタユニット 4 b は基本ユニット 3 の外部に設けられてもよいが、カウンタユニット 4 b の機能が基本ユニット 3 に内蔵されてもよい。

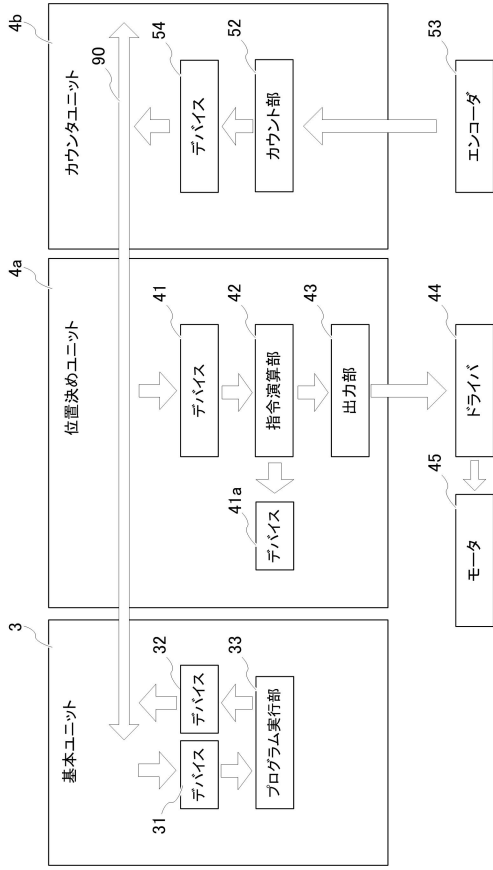
10

20

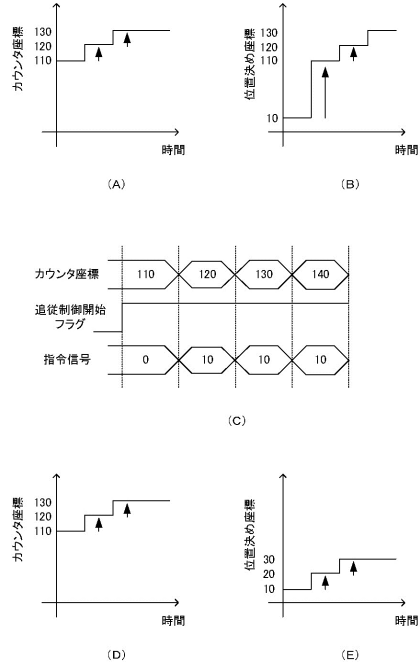
30



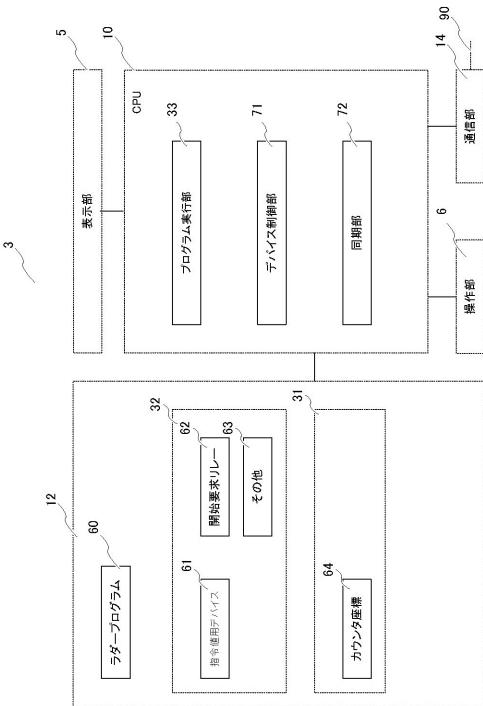
【図 6】



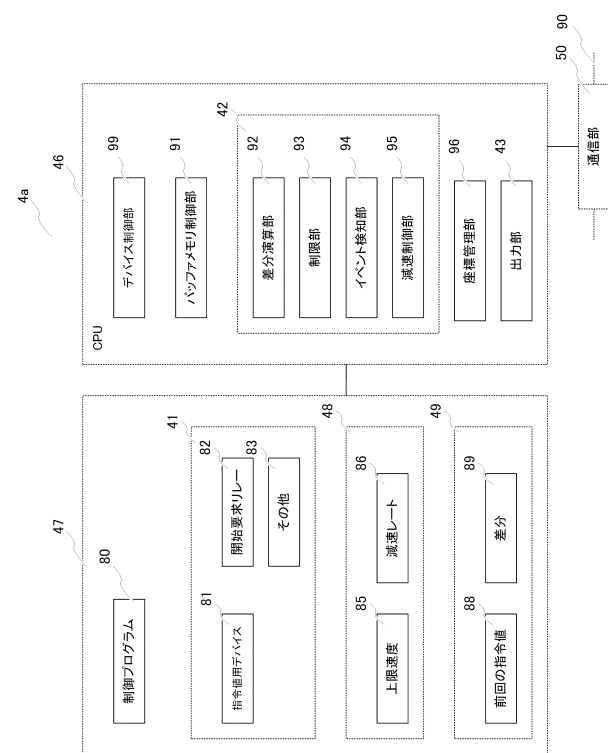
【図 7】



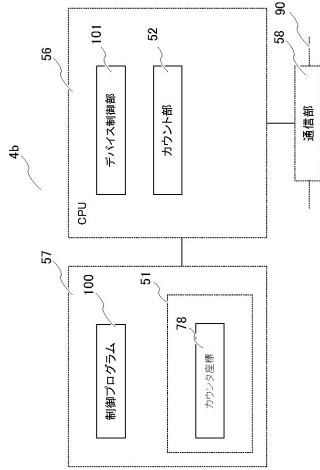
【図 8】



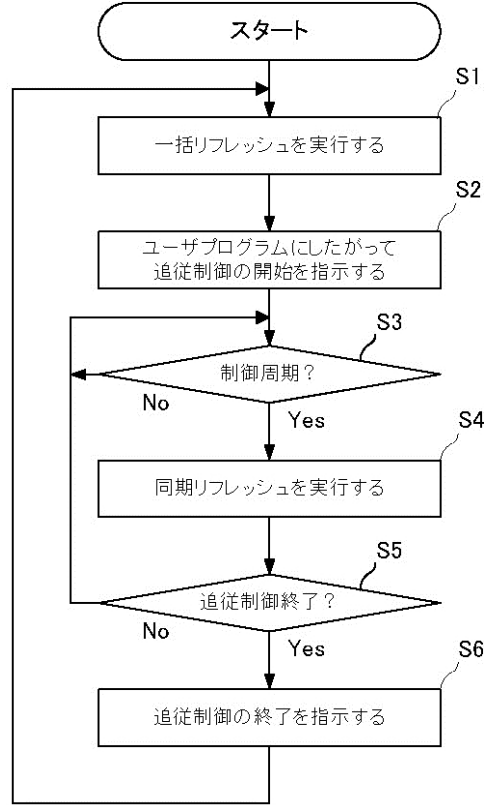
【図 9】



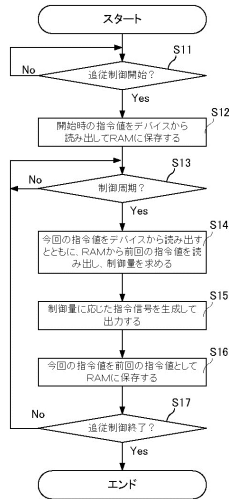
【図10】



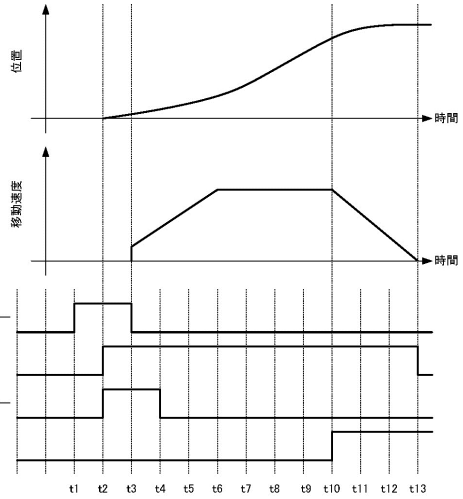
【図11】



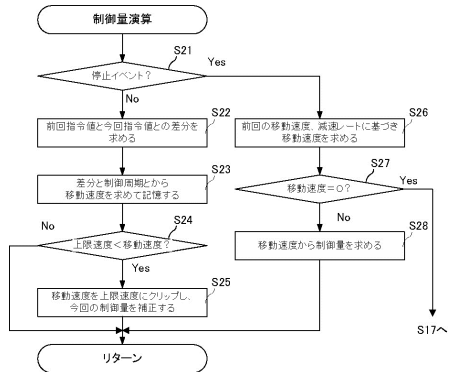
【図12】



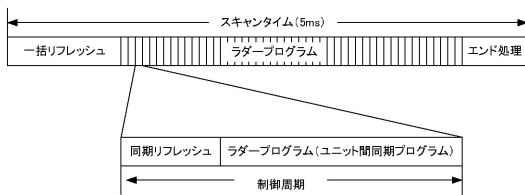
【図14】



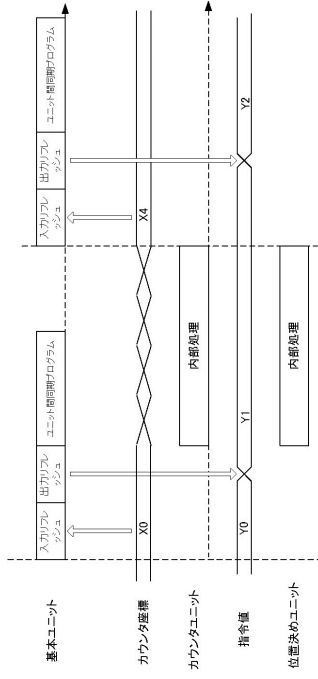
【図13】



【図15】



【 16】



---

フロントページの続き

(72)発明者 丸山 大介  
大阪府大阪市東淀川区東中島1-3-14 株式会社キーエンス内

審査官 永田 和彦

(56)参考文献 特開2010-140341(JP,A)  
特開2000-339009(JP,A)  
特開2014-52670(JP,A)  
特開昭61-69398(JP,A)  
特開2006-338283(JP,A)  
特開平8-211918(JP,A)  
特開2015-35021(JP,A)  
米国特許出願公開第2012/0256566(US,A1)  
特開平9-34522(JP,A)

(58)調査した分野(Int.Cl., DB名)  
G05B 19/00 - 19/46,  
G05D 3/00 - 3/20,  
B23Q 5/00 - 5/58