

(19) 日本国特許庁(JP)

(12) 特許公報(B1)

(11) 特許番号

特許第4894961号  
(P4894961)

(45) 発行日 平成24年3月14日(2012.3.14)

(24) 登録日 平成24年1月6日(2012.1.6)

(51) Int.Cl. F I  
**G05B 19/05 (2006.01)** G05B 19/05 F

請求項の数 14 (全 37 頁)

<p>(21) 出願番号 特願2011-56767(P2011-56767)                  (22) 出願日 平成23年3月15日(2011.3.15)                  審査請求日 平成23年3月25日(2011.3.25)</p>	<p>(73) 特許権者 000002945                  オムロン株式会社                  京都市下京区塩小路通堀川東入南不動堂町                  801番地                  (74) 代理人 110001195                  特許業務法人深見特許事務所                  (72) 発明者 西山 佳秀                  京都府京都市下京区塩小路通堀川東入南不                  動堂町801番地 オムロン株式会社内                  (72) 発明者 ▲濱▼崎 治                  京都府京都市下京区塩小路通堀川東入南不                  動堂町801番地 オムロン株式会社内                  (72) 発明者 江口 重行                  京都府京都市下京区塩小路通堀川東入南不                  動堂町801番地 オムロン株式会社内                  最終頁に続く</p>
---	---

(54) 【発明の名称】 PLCのCPUユニット、PLC用システムプログラムおよびPLC用システムプログラムを格納した記録媒体

(57) 【特許請求の範囲】

【請求項1】

制御対象を制御するPLCのCPUユニットであって、  
 マイクロプロセッサと、  
 記憶手段と、  
 出力データの送信および入力データの受信を行う通信回路と、  
 制御サイクルの周期を設定する手段とを備え、  
 前記PLCのCPUユニットは、前記出力データの送信と、前記入力データの受信と、  
 前記入力データを使用して前記出力データを生成する第1の制御プログラムの実行とを繰り返すこと  
 によって前記制御対象を制御するように構成されており、  
 前記記憶手段は、前記第1の制御プログラムと、前記第1の制御プログラムの実行を制  
 御するスケジューラプログラムとの格納に用いられ、  
 前記マイクロプロセッサは、前記記憶手段に格納された前記スケジューラプログラムお  
 よび前記第1の制御プログラムを実行し、  
 前記通信回路は、一定の制御サイクルの周期で、前記出力データの送信および入力デー  
 タの受信を行い、  
 前記スケジューラプログラムは、  
 前記マイクロプロセッサに、前記第1の制御プログラムの実行が終了した制御サイク  
 ルの次の制御サイクルにおいて、前記通信回路による出力データの送信および入力デー  
 タの受信の後に、前記第1の制御プログラムの実行を開始させる命令と、

10

20

前記マイクロプロセッサに、前記第1の制御プログラムの実行が終了しなかった制御サイクルの次の制御サイクルにおいて、前記第1の制御プログラムの未実行の部分を実行させる命令とを含む、PLCのCPUユニット。

【請求項2】

前記第1の制御プログラムは、実行されるごとにモータドライバに対して出力される指令値を算出するモーション演算プログラムを含み、

前記モーション演算プログラムは、ユーザにおける制御目的に応じて作成されたユーザプログラムにより起動された初回の実行において、指令値の算出を開始するために必要な初期処理を実行することにより後続の実行に要する時間よりも長い実行時間を必要とする、請求項1に記載のPLCのCPUユニット。

10

【請求項3】

前記第1の制御プログラムは、さらに、前記ユーザプログラムを含む、請求項2に記載のPLCのCPUユニット。

【請求項4】

前記記憶手段は、さらに、前記制御サイクルの周期の2以上の整数倍の周期の実行サイクルが設定された第2の制御プログラムの格納に用いられ、

前記スケジューラプログラムは、前記マイクロプロセッサに、前記制御サイクル内の前記第1の制御プログラムの終了後の期間において、前記実行サイクルに従って前記第2の制御プログラムを実行させる命令を含む、請求項1～3のいずれか1項に記載のPLCのCPUユニット。

20

【請求項5】

前記記憶手段は、さらに、前記制御サイクルの周期の2以上の整数倍の周期の実行サイクルが設定された第2の制御プログラムの格納に用いられ、

前記スケジューラプログラムは、前記マイクロプロセッサに、前記制御サイクル内の前記第1の制御プログラムの終了後の期間において、前記実行サイクルに従って前記第2の制御プログラムを実行させる命令を含み、

前記第1の制御プログラムは、前記モーション演算プログラムを含み、

前記第2の制御プログラムは、前記ユーザプログラムを含む、請求項2に記載のPLCのCPUユニット。

【請求項6】

30

前記記憶手段は、さらに、前記第1の制御プログラムの1回の実行が継続する制御サイクル数の限度数を特定する設定内容と、前記第1の制御プログラムの1回の実行が前記限度数の制御サイクル内に終了しない場合に実行されるべき異常処理プログラムとの格納に用いられ、

前記スケジューラプログラムは、前記第1の制御プログラムが1回の実行を継続している制御サイクルの数が前記限度数に達し、かつ、当該制御サイクル内で第1の制御プログラムの実行が終了しなかった場合に、次の制御サイクルにおいて、前記マイクロプロセッサに、第1の制御プログラムの未実行部分を実行させることなく前記異常処理プログラムを実行させる命令を含む、請求項1に記載のPLCのCPUユニット。

【請求項7】

40

前記通信回路は、前記マイクロプロセッサによる制御を受けて前記出力データの送信および入力データの受信を開始するように構成されており、

前記スケジューラプログラムは、さらに、前記通信回路に、一定の制御サイクルの周期で前記通信を開始させる命令を含む、請求項1に記載のPLCのCPUユニット。

【請求項8】

前記スケジューラプログラムは、さらに、制御サイクル内に前記第1の制御プログラムの実行が終了しなかった場合は、次の制御サイクルにおいて、前記通信回路に、前記第1の制御プログラムの未実行の部分に優先して前記通信を開始させるための処理を実行させ、その後、前記マイクロプロセッサに、第1の制御プログラムの未実行の部分を実行させる命令を含む、請求項7に記載のPLCのCPUユニット。

50

## 【請求項 9】

前記制御サイクルの周期で信号を発生するタイマ回路をさらに備え、

前記通信回路は、前記タイマ回路が発生する前記信号をトリガとして前記出力データの送信および入力データの受信を開始するように構成されている、請求項 1 に記載の P L C の C P U ユニット。

## 【請求項 10】

前記マイクロプロセッサは、第 1 および第 2 のコアを含み、

前記第 1 のコアは前記スケジューラプログラムを実行し、前記第 2 のコアは前記第 1 の制御プログラムを実行し、

前記スケジューラプログラムは、前記第 1 の制御プログラムの実行が終了しなかった制御サイクルの次の制御サイクルにおいて、前記第 1 の制御プログラムの実行を中断しないことにより前記第 1 の制御プログラムの未実行の部分を前記マイクロプロセッサに実行させる命令を含む、請求項 1、7、9 のいずれか 1 項に記載の P L C の C P U ユニット。

10

## 【請求項 11】

マイクロプロセッサと、記憶手段と、出力データの送信および入力データの受信を行う通信回路とを備え、前記通信回路による前記出力データの送信および前記入力データの受信を一定の制御サイクルの周期で繰り返すとともに、前記入力データを使用して前記出力データを生成する、前記記憶手段に格納される第 1 の制御プログラムの実行を繰り返すことによって制御対象を制御する P L C の C P U ユニットにおいて、前記記憶手段に格納されて前記マイクロプロセッサによって実行されるための P L C 用システムプログラムであって、

20

前記第 1 の制御プログラムの実行を制御するスケジューラプログラムと、前記制御サイクルの周期を設定するプログラムとを含み、

前記スケジューラプログラムは、

前記マイクロプロセッサに、前記第 1 の制御プログラムの実行が終了した制御サイクルの次の制御サイクルにおいて、前記通信回路による出力データの送信および入力データの受信の後に、前記第 1 の制御プログラムの実行を開始させる命令と、

前記マイクロプロセッサに、前記第 1 の制御プログラムの実行が終了しなかった制御サイクルの次の制御サイクルにおいて、前記第 1 の制御プログラムの未実行の部分を実行させる命令とを含む、P L C 用システムプログラム。

30

## 【請求項 12】

さらに、前記第 1 の制御プログラムの一部または全部であり、実行されるごとにモータドライバに対して出力される指令値を算出するモーション演算プログラムを含み、

前記モーション演算プログラムは、ユーザにおける制御目的に応じて作成されたユーザプログラムにより起動された初回の実行において、指令値の算出を開始するために必要な初期処理を実行することにより後続の実行に要する時間よりも長い実行時間を必要とする、請求項 11 に記載の P L C 用システムプログラム。

## 【請求項 13】

マイクロプロセッサと、記憶手段と、出力データの送信および入力データの受信を行う通信回路とを備え、前記通信回路による前記出力データの送信および前記入力データの受信を一定の制御サイクルの周期で繰り返すとともに、前記入力データを使用して前記出力データを生成する、前記記憶手段に格納される第 1 の制御プログラムの実行を繰り返すことによって制御対象を制御する P L C の C P U ユニットにおいて、前記記憶手段に格納されて前記マイクロプロセッサによって実行されるための P L C 用システムプログラムを格納した記録媒体であって、

40

前記第 1 の制御プログラムの実行を制御するスケジューラプログラムと、前記制御サイクルの周期を設定するプログラムとを含み、

前記スケジューラプログラムは、

前記マイクロプロセッサに、前記第 1 の制御プログラムの実行が終了した制御サイクルの次の制御サイクルにおいて、前記通信回路による出力データの送信および入力データ

50

の受信の後に、前記第1の制御プログラムの実行を開始させる命令と、

前記マイクロプロセッサに、前記第1の制御プログラムの実行が終了しなかった制御サイクルの次の制御サイクルにおいて、前記第1の制御プログラムの未実行の部分を実行させる命令とを含む、PLC用システムプログラムを格納した記録媒体。

【請求項14】

前記PLC用システムプログラムは、さらに、前記第1の制御プログラムの一部または全部であり、モータドライバに対して出力される指令値を算出するモーション演算プログラムを含み、

前記モーション演算プログラムは、ユーザにおける制御目的に応じて作成されたユーザプログラムにより起動された初回の実行において、モーション指令値データの算出を開始するために必要な初期処理を実行することにより後続の実行に要する時間よりも長い実行時間を必要とする、請求項13に記載のPLC用システムプログラムを格納した記録媒体

10

【発明の詳細な説明】

【技術分野】

【0001】

本発明は、機械や設備などの動作を制御するために用いられるPLC(Programmable Logic Controller、あるいはプログラマブルコントローラとも称される)における出力処理、入力処理、演算処理などを含む制御動作のスケジューリングに関する。

【背景技術】

20

【0002】

PLCは、たとえば、ユーザプログラムを実行するマイクロプロセッサを含むCPU(Central Processing Unit)ユニット、外部のスイッチやセンサからの信号入力および外部のリレーやアクチュエータへの信号出力を担当するIO(Input Output)ユニットといった複数のユニットで構成される。それらのユニット間で、制御プログラムの実行サイクルごとに、PLCシステムバスおよび/またはフィールドネットワークを経由してデータの授受をしながら、PLCは制御動作を実行する。

【0003】

たとえば、特許文献1(特開2000-105604号公報)には、PLCにおいて、出力リフレッシュ(出力処理)、入力リフレッシュ(入力処理)、およびシーケンスプログラム実行(演算処理)を一巡実行するスキャンタイム(実行サイクルの周期)を一定化する技術が開示されている。この技術の原理は、一巡実行に要すると想定される時間よりも長い時間であるコンスタントスキャン設定時間を設定しておき、毎回の一巡実行がなされた後に、そのサイクルにおける一巡実行を開始してからの経過時間がコンスタントスキャン設定時間に到達するのを待って次のサイクルの一巡実行を開始するというものである。この方式では、一巡実行中には経過時間とコンスタントスキャン設定時間との対比判断をしないので、何らかの事情により一巡実行がコンスタントスキャン設定時間を越えて継続した場合には、そのサイクルは一巡実行が終了するまで延長されてしまう。したがって、実行サイクルの周期が一定ではなくなる。

30

【先行技術文献】

40

【特許文献】

【0004】

【特許文献1】特開2000-105604号公報

【発明の概要】

【発明が解決しようとする課題】

【0005】

情報技術の分野においては、マイクロプロセッサや通信ネットワークの高速化が進展しつつある。そのため、PLCにおいてもそれらの技術を利用して制御プログラムの実行サイクルの周期が短縮化されつつある。それにより、PLCのCPUユニットと他のユニットとの間で行われる制御データ(入力データおよび出力データ)の通信の周期を高精度に

50

一定化することが有利になりつつある。

【 0 0 0 6 】

PLC内での制御データの通信周期が一定であれば、I/Oユニットにおいて外部の機器に与える出力信号の更新周期を一定化することが容易になる。これにより、外部の機器においても、PLCの制御サイクルの周期に同期した動作をすることが容易になる。

【 0 0 0 7 】

また、制御データの通信周期が一定であれば、フィールドネットワークを介してCPUユニットと通信するリモートI/Oターミナルにおいて、CPUユニットに入力データを送信する予定時刻の直前に外部から入力データを取得するような、精密に時間制御された計画的な動作が可能になる。これにより、CPUユニットは、入手しうる最も新しい入力データをを用いて演算処理を行うことができるようになる。そのため、制御サイクルの周期の短縮化とあいまって、入力データの取得から、その入力データを用いて演算した結果である出力データを送出するまでの応答時間を短縮することができる。

【 0 0 0 8 】

さらに、制御データの通信周期が一定であれば、PLCのいずれかのユニットにおいて、予定された時刻に通信が行われなかったことを検知することによって通信異常の発生を迅速に判定することができる。

【 0 0 0 9 】

CPUユニットがフィールドネットワークを経由してPLCの外部にあるサーボモータドライバのような制御対象機器と直接通信する場合にも、フィールドネットワークによる制御データの通信周期を一定化することができ有利である。PLCの実行サイクルの周期をサーボモータの制御周期として使用できるほどに短縮することが可能になってきたことにより、専用のモーションコントローラを別途設けることなく、PLCのCPUユニットが、サーボモータドライバに対する指令値を実行サイクルごとに生成し出力することによってモータ制御することも可能となってきたが、このような場合には、モータを高精度に制御するために、CPUユニットが一定周期で出力信号を送出できることは重要である。

【 0 0 1 0 】

なお、本発明に係るPLCのCPUユニットの技術的範囲は、もっぱらシーケンスプログラムの実行によって制御を遂行する従来のCPUユニットの機能にとどまらず、モーションコントローラの機能を兼ね備えるなど、その機能が拡張されたものをも含む。

【 0 0 1 1 】

ところで、制御プログラム（シーケンスプログラムおよびモーション演算プログラムなど）の実行所要時間は、制御プログラムを繰り返し実行するたびに異なりうる。この実行時間の分布を見ると、実行回数の大部分において比較的短く、かつ、ばらつきの小さな実行時間となり、一方で、実行回数の少数において比較的長い実行時間となる傾向を示すことが多い。

【 0 0 1 2 】

実行時間がこのような分布となるのは、制御プログラムが特定の条件が成立した場合にだけ実行する処理を含むことがあるためである。そのような条件が成立したサイクルにおいては、他のサイクルにおける処理よりも多くの処理を行うので、実行時間が長くなる。実行時間が長くなる場合の例としては、シーケンスプログラム（ユーザプログラム）の中で用いられるモーションファンクションブロックのようなモーション命令の起動条件が成立したサイクルにおいて、初期処理を多数の軸について行う場合が挙げられる。この初期処理は、モーション命令の起動条件が成立したサイクルを含むそれ以降の各サイクルにおいて、モーション指令値を算出していくための処理を含む。

【 0 0 1 3 】

特許文献1（特開2000-105604号公報）に示されたような実行サイクルの周期の一定化方式では、実行サイクルの周期を、制御プログラムの最大実行時間よりも長くなるように設定しなければ、通信周期を一定にすることができない。

10

20

30

40

50

## 【 0 0 1 4 】

しかしながら、実行サイクルの周期を制御プログラムの最大実行時間よりも長くなるように設定したのでは、大部分のサイクルにおいて、より短い時間で制御プログラムが実行完了しているにもかかわらず、そのような短い実行時間に適合した短い実行サイクルの周期を実現することができない。

## 【 0 0 1 5 】

本発明は、制御プログラムの実行時間が散発的に長くなりうることを考慮して、P L C の C P U ユニットが行う制御データの出力および入力のための通信の周期を、制御プログラムの最大実行時間よりも短い一定時間としうる P L C の C P U ユニットの提供することを目的とする。

10

## 【 課題を解決するための手段 】

## 【 0 0 1 6 】

本発明のある局面によれば、制御対象を制御する P L C の C P U ユニットの提供する。P L C の C P U ユニットは、マイクロプロセッサと、記憶手段と、出力データの送信および入力データの受信を行う通信回路と、制御サイクルの周期を設定する手段とを含む。P L C の C P U ユニットは、出力データの送信と、入力データの受信と、入力データを使用して出力データを生成する第 1 の制御プログラムの実行とを繰り返すことによって制御対象を制御するように構成されている。記憶手段は、第 1 の制御プログラムと、第 1 の制御プログラムの実行を制御するスケジューラプログラムとの格納に用いられる。マイクロプロセッサは、記憶手段に格納されたスケジューラプログラムおよび第 1 の制御プログラム

20

を実行する。通信回路は、制御サイクルごとに、出力データの送信および入力データの受信を行う。スケジューラプログラムは、マイクロプロセッサに、第 1 の制御プログラムの実行が終了した制御サイクルの次の制御サイクルにおいて、通信回路による出力データの送信および入力データの受信の後に、第 1 の制御プログラムの実行を開始させる命令と、マイクロプロセッサに、第 1 の制御プログラムの実行が終了しなかった制御サイクルの次の制御サイクルにおいて、第 1 の制御プログラムの未実行の部分を実行させる命令とを含む。

## 【 0 0 1 7 】

この局面によれば、P L C のユーザは、第 1 の制御プログラムの実行を完了するまでに要する時間が散発的に長くなる状況において、P L C の C P U ユニットが制御データの出力および入力のために制御サイクルと同期して行う通信の周期を、第 1 の制御プログラムの実行を完了するまでに要する時間の最大値よりも短い一定時間に設定することができる。

30

## 【 0 0 1 8 】

好ましくは、第 1 の制御プログラムは、実行されるごとにモータドライバに対して出力される指令値を算出するモーション演算プログラムを含む。モーション演算プログラムは、ユーザにおける制御目的に応じて作成されたユーザプログラムにより起動された初回の実行において、指令値の算出を開始するために必要な初期処理を実行することにより後続の実行に要する時間よりも長い実行時間を必要とする。

## 【 0 0 1 9 】

この局面によれば、モーション演算プログラムが起動された制御サイクル（モーション起動サイクル）より後の制御サイクルにおいては、モーション演算プログラムの実行周期の超過は発生しないが、モーション起動サイクルにおいては実行周期の超過が発生しうる程度に制御サイクルの周期を短く設定することにより、制御サイクルの周期が短く、かつ、モーション動作中における周期超過が発生しにくいモーション制御を実現できる。

40

## 【 0 0 2 0 】

さらに好ましくは、第 1 の制御プログラムは、さらに、ユーザプログラムを含む。好ましくは、記憶手段は、さらに、制御サイクルの周期の 2 以上の整数倍の周期の実行サイクルが設定された第 2 の制御プログラムの格納に用いられる。スケジューラプログラムは、マイクロプロセッサに、制御サイクル内の第 1 の制御プログラムの終了後の期間に

50

において、実行サイクルに従って第2の制御プログラムを実行させる命令を含む。

【0021】

この局面によれば、その実行サイクルの周期が制御サイクルの周期よりも長くなっても影響がないような処理を第2の制御プログラムとして処理することにより、第1の制御プログラムの処理量を削減することができる。それにより、制御サイクルの周期をより短縮できる可能性がある。また、第1の制御プログラムの実行が1回の制御サイクル内に終了しない頻度を小さくできる可能性もある。さらに、第1の制御プログラムの実行が想定を超える制御サイクル数内で終了しない頻度を小さくできる可能性がある。

【0022】

あるいは好ましくは、記憶手段は、さらに、制御サイクルの周期の2以上の整数倍の周期の実行サイクルが設定された第2の制御プログラムの格納に用いられる。スケジューラプログラムは、マイクロプロセッサに、制御サイクル内の第1の制御プログラムの終了後の期間において、実行サイクルに従って第2の制御プログラムを実行させる命令を含む。第1の制御プログラムは、モーション演算プログラムを含み、第2の制御プログラムは、ユーザプログラムを含む。

10

【0023】

好ましくは、記憶手段は、さらに、第1の制御プログラムの1回の実行が継続する制御サイクル数の限度数を特定する設定内容と、第1の制御プログラムの1回の実行が限度数の制御サイクル内に終了しない場合に実行されるべき異常処理プログラムとの格納に用いられる。スケジューラプログラムは、第1の制御プログラムが1回の実行を継続している制御サイクルの数が限度数に達し、かつ、当該制御サイクル内で第1の制御プログラムの実行が終了しなかった場合に、次の制御サイクルにおいて、マイクロプロセッサに、第1の制御プログラムの未実行部分を実行させることなく異常処理プログラムを実行させる命令を含む。

20

【0024】

この局面によれば、想定を超える制御サイクル数まで第1の制御プログラムの実行が継続するといった異常状態が発生した場合には、制御プログラムの実行を停止しつつ、異常処理プログラムを実行することにより、より高い安全を確保することができる。

【0025】

好ましくは、通信回路は、マイクロプロセッサによる制御を受けて出力データの送信および入力データの受信を開始するように構成されている。スケジューラプログラムは、さらに、通信回路に、制御サイクルごとに通信を開始させる命令を含む。

30

【0026】

この局面によれば、マイクロプロセッサが集約的に制御サイクルを管理するので、通信回路に、一定周期で通信するための特別なハードウェアが不要となる。

【0027】

さらに好ましくは、スケジューラプログラムは、さらに、制御サイクル内に第1の制御プログラムの実行が終了しなかった場合は、次の制御サイクルにおいて、通信回路に、第1の制御プログラムの未実行の部分に優先して通信を開始させるための処理を実行させ、その後、マイクロプロセッサに、第1の制御プログラムの未実行の部分を実行させる命令を含む。

40

【0028】

この局面によれば、シングルコアのマイクロプロセッサで通信開始の制御をする場合であって、第1の制御プログラムを複数の制御サイクルにまたがって実行する場合でも、通信の一定周期性を維持することができる。

【0029】

好ましくは、PLCのCPUユニットは、制御サイクルの周期で信号を発生するタイマ回路をさらに含む。通信回路は、タイマ回路が発生する信号をトリガとして出力データの送信および入力データの受信を開始するように構成されている。

【0030】

50

この局面によれば、通信回路における通信開始は、タイマ回路が発生する制御サイクルの周期で発生する信号によってトリガされるので、スケジューラプログラムにおける処理量を低減できる。そのため、第1の制御プログラムに含まれる処理内容のうち未実行の部分を実行する制御サイクルにおいて、スケジューラプログラムは、制御プログラムに含まれる処理内容のうち未実行の部分を実行するための時間をより多く割り当てることができる。その結果、第1の制御プログラムに含まれる処理内容のうち未実行の部分をその制御サイクル内での実行を終了させる可能性を高めることができる。

【0031】

好ましくは、マイクロプロセッサは、第1および第2のコアを含む。第1のコアはスケジューラプログラムを実行し、第2のコアは第1の制御プログラムを実行する。スケジューラプログラムは、第1の制御プログラムの実行が終了しなかった制御サイクルの次の制御サイクルにおいて、第1の制御プログラムの実行を中断しないことにより第1の制御プログラムの未実行の部分をマイクロプロセッサに実行させる命令を含む。

10

【0032】

本発明の別の局面によれば、PLC用システムプログラムを提供する。PLC用システムプログラムは、マイクロプロセッサと、記憶手段と、出力データの送信および入力データの受信を行う通信回路とを含み、通信回路による出力データの送信および入力データの受信を制御サイクルごとに繰り返すとともに、入力データを使用して出力データを生成する、記憶手段に格納される第1の制御プログラムの実行を繰り返すことによって制御対象を制御するPLCのCPUユニットにおいて、記憶手段に格納されてマイクロプロセッサによって実行される。PLC用システムプログラムは、第1の制御プログラムの実行を制御するスケジューラプログラムと、制御サイクルの周期を設定するプログラムとを含む。スケジューラプログラムは、マイクロプロセッサに、第1の制御プログラムの実行が終了した制御サイクルの次の制御サイクルにおいて、通信回路による出力データの送信および入力データの受信の後に、第1の制御プログラムの実行を開始させる命令と、マイクロプロセッサに、第1の制御プログラムの実行が終了しなかった制御サイクルの次の制御サイクルにおいて、第1の制御プログラムの未実行の部分を実行させる命令とを含む。

20

【0033】

好ましくは、PLC用システムプログラムは、実行されるごとにモータドライバに対して出力される指令値を算出するモーション演算プログラムを含む。モーション演算プログラムは、第1の制御プログラムの一部または全部である。モーション演算プログラムは、ユーザにおける制御目的に応じて作成されたユーザプログラムにより起動された初回の実行において、指令値の算出を開始するために必要な初期処理を実行することにより後続の実行に要する時間よりも長い実行時間を必要とする。

30

【0034】

本発明のさらに別の局面によれば、PLC用システムプログラムを格納した記録媒体を提供する。PLC用システムプログラムは、マイクロプロセッサと、記憶手段と、出力データの送信および入力データの受信を行う通信回路とを含み、通信回路による出力データの送信および入力データの受信を制御サイクルごとに繰り返すとともに、入力データを使用して出力データを生成する、記憶手段に格納される第1の制御プログラムの実行を繰り返すことによって制御対象を制御するPLCのCPUユニットにおいて、記憶手段に格納されてマイクロプロセッサによって実行される。PLC用システムプログラムは、第1の制御プログラムの実行を制御するスケジューラプログラムと、制御サイクルの周期を設定するプログラムとを含む。スケジューラプログラムは、マイクロプロセッサに、第1の制御プログラムの実行が終了した制御サイクルの次の制御サイクルにおいて、通信回路による出力データの送信および入力データの受信の後に、第1の制御プログラムの実行を開始させる命令と、マイクロプロセッサに、第1の制御プログラムの実行が終了しなかった制御サイクルの次の制御サイクルにおいて、第1の制御プログラムの未実行の部分を実行させる命令とを含む。

40

【0035】

50

好ましくは、PLC用システムプログラムは、実行されるごとにモータドライバに対して出力される指令値を算出するモーション演算プログラムを含む。モーション演算プログラムは、第1の制御プログラムの一部または全部である。モーション演算プログラムは、ユーザにおける制御目的に応じて作成されたユーザプログラムにより起動された初回の実行において、指令値の算出を開始するために必要な初期処理を実行することにより後続の実行に要する時間よりも長い実行時間を必要とする。

【0036】

なお、本明細書における（日本語の）「命令」は、ある機能を実現するための、プログラムのソースリストに表れる個々の命令に限らず、それら個々の命令や関数等の集合をも意味しており、たとえば英語ではinstructionsのように複数形で表記されるべきものである。

10

【発明の効果】

【0037】

本発明によれば、PLCのユーザは、第1の制御プログラムの実行時間が散発的に長くなる状況において、PLCのCPUユニットが制御データの出力および入力のために制御サイクルと同期して行う通信の周期を、第1の制御プログラムの最大実行時間よりも短い一定時間に設定することができる。

【図面の簡単な説明】

【0038】

【図1】本発明の実施の形態に係るPLCシステムの概略構成を示す模式図である。

20

【図2】本発明の実施の形態に係るCPUユニットのハードウェア構成を示す模式図である。

【図3】本発明の実施の形態に係るCPUユニットで実行されるソフトウェア構成を示す模式図である。

【図4】本発明の実施の形態に係るCPUユニットのメインメモリの領域構成を示す模式図である。

【図5】本発明の実施の形態1に係る全体処理動作を示すシーケンス図である。

【図6】本発明の実施の形態2に係る全体処理動作を示すシーケンス図である。

【図7】本発明の実施の形態3に係る全体処理動作を示すシーケンス図である。

【図8】本発明の実施の形態4に係る全体処理動作を示すシーケンス図である。

30

【図9】本発明の実施の形態5に係る全体処理動作を示すシーケンス図である。

【図10】本発明の実施の形態6に係る全体処理動作を示すシーケンス図である。

【図11】本発明の実施の形態7に係る全体処理動作を示すシーケンス図である。

【図12】本発明の実施の形態に係るCPUユニットに接続して用いられるPLCサポート装置のハードウェア構成を示す模式図である。

【図13】本発明の実施の形態に係るCPUユニットに接続して用いられるPLCサポート装置のソフトウェア構成を示す模式図である。

【発明を実施するための形態】

【0039】

本発明の実施の形態について、図面を参照しながら詳細に説明する。なお、図中の同一または相当部分については、同一符号を付してその説明は繰り返さない。

40

【0040】

< A . システム構成 >

本実施の形態に係るPLCは、機械や設備などの制御対象を制御する。まず、図1を参照して、本実施の形態に係るPLC1のシステム構成について説明する。

【0041】

図1は、本発明の実施の形態に係るPLCシステムの概略構成を示す模式図である。図1を参照して、PLCシステムSYSは、PLC1と、PLC1とフィールドネットワーク2を介して接続されるサーボモータドライバ3およびリモートI/Oターミナル5と、フィールド機器である検出スイッチ6およびリレー7とを含む。また、PLC1には、接続

50

ケーブル10などを介してPLCサポート装置8が接続される。

【0042】

PLC1は、主たる演算処理を実行するCPUユニット13と、1つ以上のIOユニット14と、特殊ユニット15とを含む。これらのユニットは、PLCシステムバス11を介して、データを互いに遣り取りできるように構成される。また、これらのユニットには、電源ユニット12によって適切な電圧の電源が供給される。なお、PLC1として構成される各ユニットは、PLCメーカーが提供するものであるので、PLCシステムバス11は、一般にPLCメーカーごとに独自に開発され、使用されている。これに対して、後述するようにフィールドネットワーク2については、異なるメーカーの製品同士が接続できるように、その規格などが公開されている場合も多い。

10

【0043】

CPUユニット13の詳細については、図2を参照して後述する。

IOユニット14は、一般的な入出力処理に関するユニットであり、オン/オフといった2値化されたデータの入出力を司る。すなわち、IOユニット14は、検出スイッチ6などのセンサが何らかの対象物を検出している状態(オン)および何らかの対象物も検出していない状態(オフ)のいずれであるかという情報を収集する。また、IOユニット14は、リレー7やアクチュエータといった出力先に対して、活性化するための指令(オン)および不活性化するための指令(オフ)のいずれかを出力する。

【0044】

特殊ユニット15は、アナログデータの入出力、温度制御、特定の通信方式による通信といった、IOユニット14ではサポートしない機能を有する。

20

【0045】

フィールドネットワーク2は、CPUユニット13と遣り取りされる各種データを伝送する。フィールドネットワーク2としては、典型的には、各種の産業用イーサネット(登録商標)を用いることができる。産業用イーサネット(登録商標)としては、たとえば、EtherCAT(登録商標)、Profinet IRT、MECHATROLINK(登録商標)-III、Powerlink、SERCOS(登録商標)-III、CIP Motionなどが知られており、これらのうちのいずれを採用してもよい。さらに、産業用イーサネット(登録商標)以外のフィールドネットワークを用いてもよい。たとえば、モーション制御を行わない場合であれば、DeviceNet、Component/IP(登録商標)などを用いてもよい。本実施の形態に係るPLCシステムSYSでは、典型的に、本実施の形態においては、産業用イーサネット(登録商標)であるEtherCAT(登録商標)をフィールドネットワーク2として採用する場合の構成について例示する。

30

【0046】

なお、図1には、PLCシステムバス11およびフィールドネットワーク2の両方を有するPLCシステムSYSを例示するが、一方のみを搭載するシステム構成を採用することもできる。たとえば、フィールドネットワーク2ですべてのユニットを接続してもよい。あるいは、フィールドネットワーク2を使用せずに、サーボモータドライバ3をPLCシステムバス11に直接接続してもよい。さらに、フィールドネットワーク2の通信ユニットをPLCシステムバス11に接続し、CPUユニット13から当該通信ユニット経由で、フィールドネットワーク2に接続された機器との間の通信を行うようにしてもよい。

40

【0047】

なお、PLC1は、CPUユニット13にIOユニット14の機能やサーボモータドライバ3の機能を持たせることにより、IOユニット14やサーボモータドライバ3などを介さずにCPUユニット13が直接制御対象を制御する構成でもよい。

【0048】

サーボモータドライバ3は、フィールドネットワーク2を介してCPUユニット13と接続されるとともに、CPUユニット13からの指令値に従ってサーボモータ4を駆動する。より具体的には、サーボモータドライバ3は、PLC1から一定周期で、位置指令値

50

、速度指令値、トルク指令値といった指令値を受ける。また、サーボモータドライバ3は、サーボモータ4の軸に接続されている位置センサ（ロータリーエンコーダ）やトルクセンサといった検出器から、位置、速度（典型的には、今回位置と前回位置との差から算出される）、トルクといったサーボモータ4の動作に係る実測値を取得する。そして、サーボモータドライバ3は、CPUユニット13からの指令値を目標値に設定し、実測値をフィードバック値として、フィードバック制御を行う。すなわち、サーボモータドライバ3は、実測値が目標値に近づくようにサーボモータ4を駆動するための電流を調整する。なお、サーボモータドライバ3は、サーボモータアンプと称されることもある。

【0049】

また、図1には、サーボモータ4とサーボモータドライバ3とを組み合わせたシステム例を示すが、その他の構成、たとえば、パルスモータとパルスモータドライバとを組み合わせたシステムを採用することもできる。

【0050】

図1に示すPLCシステムSYSのフィールドネットワーク2には、さらに、リモートI/Oターミナル5が接続されている。リモートI/Oターミナル5は、基本的には、I/Oユニット14と同様に、一般的な入出力処理に関する処理を行う。より具体的には、リモートI/Oターミナル5は、フィールドネットワーク2でのデータ伝送に係る処理を行うための通信ケーブル52と、1つ以上のI/Oユニット53とを含む。これらのユニットは、リモートI/Oターミナルバス51を介して、データを互いに遣り取りできるように構成される。

【0051】

なお、PLCサポート装置8については後述する。

< B . CPUユニットのハードウェア構成 >

次に、図2を参照して、CPUユニット13のハードウェア構成について説明する。図2は、本発明の実施の形態に係るCPUユニット13のハードウェア構成を示す模式図である。図2を参照して、CPUユニット13は、マイクロプロセッサ100と、チップセット102と、メインメモリ104と、不揮発性メモリ106と、システムタイマ108と、PLCシステムバスコントローラ120と、フィールドネットワークコントローラ140と、USBコネクタ110とを含む。チップセット102と他のコンポーネントとの間は、各種のバスを介してそれぞれ結合されている。

【0052】

マイクロプロセッサ100およびチップセット102は、典型的には、汎用的なコンピュータアーキテクチャに準じて構成される。すなわち、マイクロプロセッサ100は、チップセット102から内部クロックに従って順次供給される命令コードを解釈して実行する。チップセット102は、接続されている各種コンポーネントとの間で内部的なデータを遣り取りするとともに、マイクロプロセッサ100に必要な命令コードを生成する。さらに、チップセット102は、マイクロプロセッサ100での演算処理の実行の結果得られたデータなどをキャッシュする機能を有する。

【0053】

CPUユニット13は、記憶手段として、メインメモリ104および不揮発性メモリ106を有する。

【0054】

メインメモリ104は、揮発性の記憶領域（RAM）であり、CPUユニット13への電源投入後にマイクロプロセッサ100で実行されるべき各種プログラムを保持する。また、メインメモリ104は、マイクロプロセッサ100による各種プログラムの実行時の作業用メモリとしても使用される。このようなメインメモリ104としては、DRAM（Dynamic Random Access Memory）やSRAM（Static Random Access Memory）といったデバイスが用いられる。

【0055】

一方、不揮発性メモリ106は、リアルタイムOS（Operating System）、PLC1の

10

20

30

40

50

システムプログラム、ユーザプログラム、モーション演算プログラム、システム設定パラメータといったデータを不揮発的に保持する。これらのプログラムやデータは、必要に応じて、マイクロプロセッサ100がアクセスできるようにメインメモリ104にコピーされる。このような不揮発性メモリ106としては、フラッシュメモリのような半導体メモリを用いることができる。あるいは、ハードディスクドライブのような磁気記録媒体や、DVD-RAM (Digital Versatile Disk Random Access Memory) のような光学記録媒体などを用いることもできる。

**【0056】**

システムタイマ108は、一定周期ごとに割り込み信号を発生してマイクロプロセッサ100に提供する。典型的には、ハードウェアの仕様によって、複数の異なる周期でそれぞれ割り込み信号を発生するように構成されるが、OS (Operating System) やBIOS (Basic Input Output System) などによって、任意の周期で割り込み信号を発生するように設定することもできる。このシステムタイマ108が発生する割り込み信号を利用して、後述するような制御サイクルごとの制御動作が実現される。

10

**【0057】**

CPUユニット13は、通信回路として、PLCシステムバスコントローラ120およびフィールドネットワークコントローラ140を有する。これらの通信回路は、出力データの送信および入力データの受信を行う。

**【0058】**

なお、CPUユニット13自体にIOユニット14やサーボモータドライバ3の機能を持たせる場合は、通信回路による出力データの送信および入力データの受信は、それらの機能を担う部分を通信の相手方としてCPUユニット13の内部で行われる送信および受信となる。

20

**【0059】**

PLCシステムバスコントローラ120は、PLCシステムバス11を介したデータの遣り取りを制御する。より具体的には、PLCシステムバスコントローラ120は、DMA (Dynamic Memory Access) 制御回路122と、PLCシステムバス制御回路124と、バッファメモリ126とを含む。なお、PLCシステムバスコントローラ120は、PLCシステムバスコネクタ130を介してPLCシステムバス11と内部的に接続される。

30

**【0060】**

バッファメモリ126は、PLCシステムバス11を介して他のユニットへ出力されるデータ(以下「出力データ」とも称す。)の送信バッファ、および、PLCシステムバス11を介して他のユニットから入力されるデータ(以下「入力データ」とも称す。)の受信バッファとして機能する。なお、マイクロプロセッサ100による演算処理によって作成された出力データは、原始的にはメインメモリ104に格納される。そして、特定のユニットへ転送されるべき出力データは、メインメモリ104から読み出されて、バッファメモリ126に一次的に保持される。また、他のユニットから転送された入力データは、バッファメモリ126に一次的に保持された後、メインメモリ104に移される。

**【0061】**

DMA制御回路122は、メインメモリ104からバッファメモリ126への出力データの転送、および、バッファメモリ126からメインメモリ104への入力データの転送を行う。

40

**【0062】**

PLCシステムバス制御回路124は、PLCシステムバス11に接続される他のユニットとの間で、バッファメモリ126の出力データを送信する処理および入力データを受信してバッファメモリ126に格納する処理を行う。典型的には、PLCシステムバス制御回路124は、PLCシステムバス11における物理層およびデータリンク層の機能を提供する。

**【0063】**

50

フィールドネットワークコントローラ140は、フィールドネットワーク2を介したデータの遣り取りを制御する。すなわち、フィールドネットワークコントローラ140は、用いられるフィールドネットワーク2の規格に従い、出力データの送信および入力データの受信を制御する。上述したように、本実施の形態においてはE t h e r C A T（登録商標）規格に従うフィールドネットワーク2が採用されるので、通常のイーサネット（登録商標）通信を行うためのハードウェアを含む、フィールドネットワークコントローラ140が用いられる。E t h e r C A T（登録商標）規格では、通常のイーサネット（登録商標）規格に従う通信プロトコルを実現する一般的なイーサネット（登録商標）コントローラを利用できる。但し、フィールドネットワーク2として採用される産業用イーサネット（登録商標）の種類によっては、通常の通信プロトコルとは異なる専用仕様の通信プロトコルに対応した特別仕様のイーサネット（登録商標）コントローラが用いられる。また、産業用イーサネット（登録商標）以外のフィールドネットワークを採用した場合には、当該規格に応じた専用のフィールドネットワークコントローラが用いられる。

10

## 【0064】

バッファメモリ146は、フィールドネットワーク2を介して他の装置などへ出力されるデータ（このデータについても、以下「出力データ」と称す。）の送信バッファ、および、フィールドネットワーク2を介して他の装置などから入力されるデータ（このデータについても、以下「入力データ」とも称す。）の受信バッファとして機能する。上述したように、マイクロプロセッサ100による演算処理によって作成された出力データは、原始的にはメインメモリ104に格納される。そして、特定の装置へ転送されるべき出力データは、メインメモリ104から読み出されて、バッファメモリ146に一次的に保持される。また、他の装置から転送された入力データは、バッファメモリ146に一次的に保持された後、メインメモリ104に移される。

20

## 【0065】

DMA制御回路142は、メインメモリ104からバッファメモリ146への出力データの転送、および、バッファメモリ146からメインメモリ104への入力データの転送を行う。

## 【0066】

フィールドネットワーク制御回路144は、フィールドネットワーク2に接続される他の装置との間で、バッファメモリ146の出力データを送信する処理および入力データを受信してバッファメモリ146に格納する処理を行う。典型的には、フィールドネットワーク制御回路144は、フィールドネットワーク2における物理層およびデータリンク層の機能を提供する。

30

## 【0067】

USBコネクタ110は、PLCサポート装置8とCPUユニット13とを接続するためのインターフェイスである。典型的には、PLCサポート装置8から転送される、CPUユニット13のマイクロプロセッサ100で実行可能なプログラムなどは、USBコネクタ110を介してPLC1に取込まれる。

## 【0068】

< C . CPUユニットのソフトウェア構成 >

40

次に、図3を参照して、本実施の形態に係る各種機能を提供するためのソフトウェア群について説明する。これらのソフトウェアに含まれる命令コードは、適切なタイミングで読み出され、CPUユニット13のマイクロプロセッサ100によって実行される。

## 【0069】

図3は、本発明の実施の形態に係るCPUユニット13で実行されるソフトウェア構成を示す模式図である。図3を参照して、CPUユニット13で実行されるソフトウェアとしては、リアルタイムOS200と、システムプログラム210と、ユーザプログラム236との3階層になっている。

## 【0070】

リアルタイムOS200は、CPUユニット13のコンピュータアーキテクチャに応じ

50

て設計されており、マイクロプロセッサ 100 がシステムプログラム 210 およびユーザプログラム 236 を実行するための基本的な実行環境を提供する。このリアルタイム OS は、典型的には、PLC のメーカーあるいは専門のソフトウェア会社などによって提供される。

【0071】

システムプログラム 210 は、PLC 1 としての機能を提供するためのソフトウェア群である。具体的には、システムプログラム 210 は、スケジューラプログラム 212 と、出力処理プログラム 214 と、入力処理プログラム 216 と、シーケンス命令演算プログラム 232 と、モーション演算プログラム 234 と、その他のシステムプログラム 220 とを含む。なお、一般には出力処理プログラム 214 および入力処理プログラム 216 は、連続的（一体として）に実行されるので、これらのプログラムを、IO 処理プログラム 218 と総称する場合もある。

10

【0072】

ユーザプログラム 236 は、ユーザにおける制御目的に応じて作成される。すなわち、PLC システム S Y S を用いて制御する対象のライン（プロセス）などに応じて、任意に設計されるプログラムである。

【0073】

後述するように、ユーザプログラム 236 は、シーケンス命令演算プログラム 232 およびモーション演算プログラム 234 と協働して、ユーザにおける制御目的を実現する。すなわち、ユーザプログラム 236 は、シーケンス命令演算プログラム 232 およびモーション演算プログラム 234 によって提供される命令、関数、機能モジュールなどを利用することで、プログラムされた動作を実現する。そのため、ユーザプログラム 236、シーケンス命令演算プログラム 232、およびモーション演算プログラム 234 を、制御プログラム 230 と総称する場合もある。

20

【0074】

このように、CPU ユニット 13 のマイクロプロセッサ 100 は、記憶手段に格納されたシステムプログラム 210 およびユーザプログラム 236 を実行する。

【0075】

以下、各プログラムについてより詳細に説明する。

ユーザプログラム 236 は、上述したように、ユーザにおける制御目的（たとえば、対象のラインやプロセス）に応じて作成される。ユーザプログラム 236 は、典型的には、CPU ユニット 13 のマイクロプロセッサ 100 で実行可能なオブジェクトプログラム形式になっている。このユーザプログラム 236 は、PLC サポート装置 8 などにおいて、ラダー言語などによって記述されたソースプログラムがコンパイルされることで生成される。そして、生成されたオブジェクトプログラム形式のユーザプログラム 236 は、PLC サポート装置 8 から接続ケーブル 10 を介して CPU ユニット 13 へ転送され、不揮発性メモリ 106 などに格納される。

30

【0076】

スケジューラプログラム 212 は、出力処理プログラム 214、入力処理プログラム 216、および制御プログラム 230 について、各実行サイクルでの処理開始および処理中断後の処理再開を制御する。より具体的には、スケジューラプログラム 212 は、ユーザプログラム 236 およびモーション演算プログラム 234 の実行を制御する。

40

【0077】

本実施の形態に係る CPU ユニット 13 では、モーション演算プログラム 234 に適した一定周期の実行サイクル（制御サイクル）を処理全体の共通サイクルとして採用する。そのため、1 つの制御サイクル内で、すべての処理を完了することは難しいので、実行すべき処理の優先度などに応じて、各制御サイクルにおいて実行を完了すべき処理と、複数の制御サイクルに亘って実行してもよい処理とが区分される。スケジューラプログラム 212 は、これらの区分された処理の実行順序などを管理する。より具体的には、スケジューラプログラム 212 は、各制御サイクル期間内において、より高い優先度が与えられて

50

いるプログラムほど先に実行する。

【 0 0 7 8 】

出力処理プログラム 2 1 4 は、ユーザプログラム 2 3 6 ( 制御プログラム 2 3 0 ) の実行によって生成された出力データを、 P L C システムバスコントローラ 1 2 0 および / またはフィールドネットワークコントローラ 1 4 0 へ転送するのに適した形式に再配置する。 P L C システムバスコントローラ 1 2 0 またはフィールドネットワークコントローラ 1 4 0 が、マイクロプロセッサ 1 0 0 からの、送信を実行するための指示を必要とする場合は、出力処理プログラム 2 1 4 がそのような指示を発行する。

【 0 0 7 9 】

入力処理プログラム 2 1 6 は、 P L C システムバスコントローラ 1 2 0 および / またはフィールドネットワークコントローラ 1 4 0 によって受信された入力データを、制御プログラム 2 3 0 が使用するのに適した形式に再配置する。

【 0 0 8 0 】

シーケンス命令演算プログラム 2 3 2 は、ユーザプログラム 2 3 6 で使用されるある種のシーケンス命令が実行されるときに呼び出されて、その命令の内容を実現するために実行されるプログラムである。

【 0 0 8 1 】

モーション演算プログラム 2 3 4 は、ユーザプログラム 2 3 6 による指示に従って実行され、サーボモータドライバ 3 やパルスモータドライバといったモータドライバに対して出力する指令値を実行されるごとに算出するプログラムである。

【 0 0 8 2 】

その他のシステムプログラム 2 2 0 は、図 3 に個別に示したプログラム以外の、 P L C 1 の各種機能を実現するためのプログラム群をまとめて示したものである。その他のシステムプログラム 2 2 0 は、制御サイクルの周期を設定するプログラム 2 2 2 を含む。

【 0 0 8 3 】

制御サイクルの周期は、制御目的に応じて適宜設定することができる。典型的には、制御サイクルの周期を指定する情報をユーザが P L C サポート装置 8 へ入力する。すると、その入力された情報は、 P L C サポート装置 8 から C P U ユニット 1 3 へ転送される。制御サイクルの周期を設定するプログラム 2 2 2 は、 P L C サポート装置 8 からの情報を不揮発性メモリ 1 0 6 に格納させるとともに、システムタイマ 1 0 8 から指定された制御サイクルの周期で割り込み信号が発生されるように、システムタイマ 1 0 8 を設定する。 C P U ユニット 1 3 への電源投入時に、制御サイクルの周期を設定するプログラム 2 2 2 が実行されることで、制御サイクルの周期を指定する情報が不揮発性メモリ 1 0 6 から読み出され、読み出された情報に従ってシステムタイマ 1 0 8 が設定される。

【 0 0 8 4 】

制御サイクルの周期を指定する情報の形式としては、制御サイクルの周期を示す時間の値や、制御サイクルの周期に関する予め用意された複数の選択肢のうちから 1 つを特定する情報 ( 番号または文字 ) などを採用することができる。

【 0 0 8 5 】

本実施の形態に係る C P U ユニット 1 3 において、制御サイクルの周期を設定する手段としては、制御サイクルの周期を指定する情報を取得するために用いられる P L C サポート装置 8 との通信手段、制御サイクルの周期を設定するプログラム 2 2 2 、ならびに制御サイクルを規定する割り込み信号の周期を任意に設定可能に構成されているシステムタイマ 1 0 8 の構成といった、制御サイクルの周期を任意に設定するために用いられる要素が該当する。制御サイクルの周期を指定する情報を P L C サポート装置 8 から取得する代わりに、制御サイクルの周期を指定するための入力を受け付けるための操作スイッチ等の入力デバイスを C P U ユニット 1 3 自身に設けてもよい。

【 0 0 8 6 】

リアルタイム O S 2 0 0 は、複数のプログラムを時間の経過に従い切り替えて実行するための環境を提供する。本実施の形態に係る P L C 1 においては、 C P U ユニット 1 3 の

10

20

30

40

50

プログラム実行によって生成された出力データを他のユニットまたは他の装置へ出力（送信）するためのイベント（割り込み）として、出力準備割り込み（P）およびフィールドネットワーク送信割り込み（X）が初期設定される。リアルタイムOS 200は、出力準備割り込み（P）またはフィールドネットワーク送信割り込み（X）が発生すると、マイクロプロセッサ100での実行対象を、割り込み発生時点で実行中のプログラムからスケジューラプログラム212に切り替える。なお、リアルタイムOS 200は、スケジューラプログラム212およびスケジューラプログラム212がその実行を制御するプログラムが何ら実行されていない場合に、その他のシステムプログラム210に含まれているプログラムを実行する。このようなプログラムとしては、たとえば、CPUユニット13とPLCサポート装置8との間の接続ケーブル10（USB）などを介した通信処理に関するものが含まれる。

10

【0087】

なお、制御プログラム230およびスケジューラプログラム212は、記憶手段であるメインメモリ104および不揮発性メモリ106に格納される。

【0088】

< D . メインメモリ構成 >

次に、図4を参照して、CPUユニット13のメインメモリ104に構成される記憶領域について説明する。

【0089】

図4は、本発明の実施の形態に係るCPUユニット13のメインメモリ104の領域構成を示す模式図である。図4を参照して、メインメモリ104には、各種のプログラム領域1041と、制御プログラムの作業領域1042と、PLCシステムバス送信バッファ1043と、PLCシステムバス受信バッファ1044と、フィールドネットワーク送信バッファ1045と、フィールドネットワーク受信バッファ1046とが形成される。

20

【0090】

各種のプログラム領域1041には、各種のプログラムを実行するためのコードが一時的に格納される。

【0091】

制御プログラムの作業領域1042には、制御プログラム230の実行により作成された出力データが一時的に格納される。

30

【0092】

PLCシステムバス送信バッファ1043およびPLCシステムバス受信バッファ1044は、PLCシステムバス11を介して送信および受信されるデータをそれぞれ一時的に格納する。同様に、フィールドネットワーク送信バッファ1045およびフィールドネットワーク受信バッファ1046は、フィールドネットワーク2を介して送信および受信されるデータをそれぞれ一時的に格納する。

【0093】

より具体的には、出力処理プログラム214は、PLCシステムバス11を介していずれかの出力データを送信する必要がある場合には、対象の出力データを制御プログラムの作業領域1042からPLCシステムバス送信バッファ1043へコピーする。その際、出力処理プログラム214は、同一のユニットに送信される複数の出力データを一括して送信できるように、PLCシステムバス送信バッファ1043において、同一のユニットに向けられた出力データ群がまとめられるように出力データを再配置する。

40

【0094】

同様に、出力処理プログラム214は、フィールドネットワーク2を介していずれかの出力データを送信する必要がある場合には、対象の出力データを制御プログラムの作業領域1042からフィールドネットワーク送信バッファ1045へコピーする。その際、出力処理プログラム214は、フィールドネットワーク送信バッファ1045において、シリアルフレームとして送信できる形式に出力データを再配置する。

【0095】

50

一方、入力処理プログラム 216 は、PLCシステムバスコントローラ 120 が受信して PLCシステムバス受信バッファ 1044 に格納された入力データ、および/または、フィールドネットワークコントローラ 140 が受信してフィールドネットワーク受信バッファ 1046 に格納された入力データを、制御プログラムの作業領域 1042 へコピーする。その際、入力処理プログラム 216 は、制御プログラムの作業領域 1042 において、制御プログラム 230 が使用するのに適した形式に入力データを再配置する。

【0096】

PLCシステムバスコントローラ 120 の DMA 制御回路 122 は、PLCシステムバス送信バッファ 1043 に格納されている出力データを PLCシステムバスコントローラ 120 のバッファメモリ 146 へ転送するとともに、バッファメモリ 146 に格納されて

10

【0097】

フィールドネットワークコントローラ 140 の DMA 制御回路 142 は、フィールドネットワーク送信バッファ 1045 に格納されている出力データをフィールドネットワークコントローラ 140 のバッファメモリ 146 へ転送するとともに、バッファメモリ 146 に格納されている入力データをフィールドネットワーク受信バッファ 1046 へ転送する。

【0098】

制御プログラムの作業領域 1042、PLCシステムバス送信バッファ 1043、PLCシステムバス受信バッファ 1044、フィールドネットワーク送信バッファ 1045、および、フィールドネットワーク受信バッファ 1046 は、それぞれのアクセスを互いに独立に制御できるように構成されている。そのため、たとえば、以下に示す(1)~(3)といった複数の動作を並列実行することができる。

20

【0099】

(1) マイクロプロセッサ 100 が行う、ユーザプログラム 236 実行に伴う制御プログラムの作業領域 1042 へのアクセス

(2) PLCシステムバスコントローラ 120 の DMA 制御回路 122 が行う、メインメモリ 104 上の PLCシステムバス送信バッファ 1043 および/または PLCシステムバス受信バッファ 1044 と PLCシステムバスコントローラ 120 内のバッファメモリ 126 との間のデータ転送のための、メインメモリ 104 上の PLCシステムバス送信バッファ 1043 および/または PLCシステムバス受信バッファ 1044 へのアクセス

30

(3) フィールドネットワークコントローラ 140 の DMA 制御回路 142 が行う、メインメモリ 104 上のフィールドネットワーク送信バッファ 1045 および/またはフィールドネットワーク受信バッファ 1046 とフィールドネットワークコントローラ 140 内のバッファメモリ 146 との間のデータ転送のための、メインメモリ 104 上のフィールドネットワーク送信バッファ 1045 および/またはフィールドネットワーク受信バッファ 1046 へのアクセス

< E . 全体処理動作 ( 実施の形態 1 ) >

次に、本実施の形態に係る各プログラムの実行順序などについて、以下説明する。なお、本明細書においては、本発明に係る技術思想の具現化例として、実施の形態 1 ~ 7 を例示する。図 5 は、本発明の実施の形態 1 に係る全体処理動作を示すシーケンス図である。

40

【0100】

本実施の形態においては、出力処理プログラム 214、入力処理プログラム 216、制御プログラム 230 ( 第 1 の制御プログラム ) には、それぞれ優先度が与えられている。これらのプログラムに与えられている優先度は、たとえば、高い方から、出力処理プログラム 214、入力処理プログラム 216、制御プログラム 230 の順である。

【0101】

PLC 1 の CPU ユニット 13 は、出力データの送信 ( 出力処理プログラム 214 ) と、入力データの受信 ( 入力処理プログラム 216 ) と、入力データを使用して出力データ

50

を生成する制御プログラム 230 の実行とを繰り返すことによって制御対象を制御する。すなわち、マイクロプロセッサ 100 は、記憶手段に格納されたスケジューラプログラム 212 および制御プログラム 230 を実行し、その実行結果に応じて、通信回路 (PLC システムバスコントローラ 120 および / またはフィールドネットワークコントローラ 140) は、制御サイクルごとに出力データの送信および入力データの受信を行う。

【0102】

スケジューラプログラム 212 は、各制御サイクル内において、優先度の高いプログラム / スレッドから順に、マイクロプロセッサ 100 に実行させる。以下、「マイクロプロセッサ 100 に実行させる」ことを単に「実行する」とも表現する。

【0103】

スケジューラプログラム 212 は、それ自体がオブジェクトインスタンスである場合に限らず、他の各種オブジェクトインスタンス (たとえば、出力処理プログラム 214) に組込まれて動作するプログラム (プロセス) であってもよい。たとえば、他のオブジェクトインスタンスにおいて呼び出されて動作するプログラムであってもよい。

【0104】

図 5 に示される符号 P および X は、それぞれシステムタイマ 108 (図 2) から一定周期ごとに発生される割り込み信号を示す。具体的には、「P」は出力準備割り込みを示し、「X」はフィールドネットワーク送信割り込みを示す。なお、図 5 には、割り込みを示す符号「P」および「X」に、制御サイクルの番号を示す数字を付加して表現する。

【0105】

(e1: 制御サイクル 1 について)

スケジューラプログラム 212 は、出力準備割り込み (P - 1) に応答して、優先度に従って、出力処理プログラム 214 を実行する。

【0106】

出力処理プログラム 214 は、PLC システムバス 11 を介していずれかの出力データを送信する必要がある場合には、対象の出力データを制御プログラムの作業領域 1042 (図 4) から PLC システムバス送信バッファ 1043 へコピーする。また、出力処理プログラム 214 は、フィールドネットワーク 2 を介していずれかの出力データを送信する必要がある場合には、対象の出力データを制御プログラムの作業領域 1042 からフィールドネットワーク送信バッファ 1045 へコピーする。

【0107】

続いて、出力処理プログラム 214 は、PLC システムバス送信バッファ 1043 への出力データのコピーが完了すると、PLC システムバスコントローラ 120 に対して送信実行を指示する (符号 A)。そして、出力処理プログラム 214 は、出力データのコピーがすべて完了すると、いったん実行を停止する。

【0108】

その後、スケジューラプログラム 212 は、フィールドネットワーク送信割り込み (X - 1) に応答して、出力処理プログラム 214 の実行を再開する。

【0109】

出力処理プログラム 214 は、実行を再開すると、フィールドネットワークコントローラ 140 に対して送信実行を指示する (符号 B)。そして、出力処理プログラム 214 は、自身の実行を終了する。

【0110】

スケジューラプログラム 212 は、出力処理プログラム 214 の実行が終了すると、優先度に従って、次に入力処理プログラム 216 を実行する。

【0111】

入力処理プログラム 216 は、PLC システムバスコントローラ 120 からの受信完了通知 (符号 C)、または、フィールドネットワークコントローラ 140 からの受信完了通知 (符号 D) を受信するまで待機する。図 5 では、一例として、PLC システムバスコントローラ 120 からの受信完了通知 (符号 C) が先に発生し、続いて、フィールドネット

10

20

30

40

50

ワークコントローラ 140 からの受信完了通知（符号 D）が発生している例を示すが、この発生順序は逆になる場合もある。

【0112】

入力処理プログラム 216 は、受信完了通知（符号 C または D）が発生すると、いずれの受信完了通知であるかに対応して、PLC システムバス受信バッファ 1044 またはフィールドネットワーク受信バッファ 1046 から制御プログラムの作業領域 1042 に入力データをコピーする。

【0113】

入力処理プログラム 216 は、入力データのコピーが完了すると、実行を終了する。

スケジューラプログラム 212 は、入力処理プログラム 216 の実行が終了すると、優先度に従って、次に制御プログラム 230 を実行する。

【0114】

制御プログラム 230 は、制御プログラムの作業領域 1042 に格納されている入力データを用いて処理を行い、処理結果の出力データを制御プログラムの作業領域 1042 に格納する。

【0115】

制御プログラム 230 の実行が終了してから、次の出力準備割り込み（P-2）があるまでの時間において、マイクロプロセッサ 100 は、リアルタイム OS の制御を利用して、その他のシステムプログラム 220（図 2）を実行する。

【0116】

（e2：PLC システムバスコントローラ 120 の動作について）

PLC システムバスコントローラ 120 の DMA 制御回路 122 は、送信実行指示（符号 A）にตอบสนองして、メインメモリ 104 の PLC システムバス送信バッファ 1043 に格納されている出力データを PLC システムバスコントローラ 120 のバッファメモリ 126 へ転送する。

【0117】

バッファメモリ 126 への出力データの転送が完了すると、PLC システムバス制御回路 124 は、バッファメモリ 126 から各出力データの宛先となっているユニットに対して対象の出力データを転送する。その後、PLC システムバス制御回路 124 は、PLC システムバス 11 上の各ユニットから入力データを PLC システムバスコントローラ 120 のバッファメモリ 126 へ転送する。

【0118】

バッファメモリ 126 への入力データの転送が完了すると、PLC システムバスコントローラ 120 の DMA 制御回路 122 は、バッファメモリ 126 に格納されている入力データをメインメモリ 104 の PLC システムバス受信バッファ 1044 へ転送する。その後、入力データの転送が完了すると、DMA 制御回路 122 は、受信完了通知（符号 C）をマイクロプロセッサ 100 に対して送信する。

【0119】

（e3：フィールドネットワークコントローラ 140 の動作について）

フィールドネットワークコントローラ 140 の DMA 制御回路 142 は、送信実行指示（符号 B）にตอบสนองして、メインメモリ 104 のフィールドネットワーク送信バッファ 1045 に格納されている出力データをフィールドネットワークコントローラ 140 のバッファメモリ 146 へ転送する。

【0120】

バッファメモリ 146 への出力データの転送が完了すると、フィールドネットワーク制御回路 144 は、バッファメモリ 146 から各出力データの宛先となっている装置に対して出力データを転送する。その後、フィールドネットワーク制御回路 144 は、フィールドネットワーク 2 上の各装置から入力データをフィールドネットワークコントローラ 140 のバッファメモリ 146 へ転送する。

【0121】

10

20

30

40

50

バッファメモリ146への入力データの転送が完了すると、フィールドネットワークコントローラ140のDMA制御回路142は、バッファメモリ146に格納されている入力データをメインメモリ104のフィールドネットワーク受信バッファ1046へ転送する。その後、入力データの転送が完了すると、DMA制御回路142は、受信完了通知(符号D)をマイクロプロセッサ100に対して送信する。

【0122】

以上の動作は、フィールドネットワーク2の種類を問わず妥当するが、フィールドネットワーク2が本実施の形態で採用するEthernet(登録商標)である場合は、さらに次のような処理が実行される。

【0123】

すなわち、Ethernet(登録商標)では、単一のフレームに、各ユニット宛の出力データおよび各ユニットからの入力データが含まれる。マスタに設定されるCPUユニット13からフレームが送信されると、送信されたフレームはダイジーチェーン接続されたそれぞれのユニットを経由して順次転送される。なお、CPUユニット13以外のそれぞれのユニットは、スレーブに設定される。フレームが末端のユニットに到達すると、そこで当該フレームは全二重の通信路を折り返して再びスレーブであるそれぞれのユニットを経由してCPUユニット13に戻る。各スレーブにおいては、往路のフレームをFIFO(First In First Out)方式で通過させながら、自スレーブ宛の出力データを選択的に取り込むとともに、自ユニットで発生した入力データをフレームの適切な位置に書き込む。各スレーブは、帰路のフレームについては、出力データの読み出し、および、入力データの書き込みは行わずに通過させる。

【0124】

このように、Ethernet(登録商標)の場合は、各ユニットに対して個別にフレームを送信し、あるいは、各ユニットからフレームを個別に受信するのではなく、単一のフレームを送信および受信することが、フィールドネットワーク制御回路144による、バッファメモリ146から各ユニットへの出力データの転送、および、各ユニットからバッファメモリ146への入力データの転送に相当する。

【0125】

フィールドネットワーク2としてEthernet(登録商標)を用いる場合は、出力データを含むフレームを送信し、そのフレームがスレーブのそれぞれのユニットを経由して戻ってきた場合には、そのフレームに入力データが含まれることになる。そのため、実行順序については、出力処理の後に入力処理を実行することが自然である。但し、使用するフィールドネットワーク2の種類によっては、入力処理の後に出力処理を実行の方が好ましい場合がある。このように、出力処理と入力処理との順序は、フィールドネットワーク2などの仕様に応じて適宜に設計すればよい。

【0126】

(e4:制御サイクル2~4について)

基本的には、制御サイクル2~4における動作は、制御サイクル1における動作と同様である。以下では、主として、制御サイクル1における動作と相違する点について説明する。

【0127】

制御サイクル2では、制御サイクル3の出力準備割り込み(P-3)が発生する時点においても制御プログラム230の実行が終了していない。制御プログラム230の実行周期は、制御サイクルの周期に等しいので、ここで制御プログラム230の実行周期の超過が発生したといえることができる。制御サイクル2で実行されていた制御プログラム230は、この時点で実行が中断され(実行対象がスケジューラプログラム212へ移り)、スケジューラプログラム212は、制御サイクル3の出力処理プログラム214および入力処理プログラム216を実行する。入力処理プログラム216の実行終了後、スケジューラプログラム212は、制御サイクル2において実行を開始した制御プログラム230(制御サイクル3の開始時に実行が中断されている)の実行を再開し、制御プログラム23

10

20

30

40

50

0に含まれる処理内容のうち未実行の部分を実行する。

【0128】

制御プログラム230の未実行の部分は、制御サイクル3の期間中に実行が終了する。スケジューラプログラム212は、制御サイクル3においては、新たに制御プログラムの実行を開始しない。

【0129】

制御サイクル4では、制御サイクル1と同じ通常の動作が実行される。

ところで、制御サイクル3の出力処理時においては、制御サイクル2において実行を開始した制御プログラム230の実行が終了していないので、その演算結果である新たな出力データがまだ完全には得られていないことになる。そのため、制御プログラムの作業領域1042には、制御プログラム230の途中までの実行の結果更新された出力データと未更新の出力データとが混在している可能性がある。そこで、制御サイクル3の出力処理プログラム214は、制御プログラムの作業領域1042からPLCシステムバス送信バッファ1043および/またはフィールドネットワーク送信バッファ1045への出力データのコピーを行わずに、PLCシステムバスコントローラ120に対して送信実行を指示する(符号A)。

【0130】

また、制御サイクル3の出力処理プログラム214は、制御サイクル1および2などと同様に、フィールドネットワーク送信割り込み(X)に応じたスケジューラプログラム212からの実行の再開指示に従って、フィールドネットワークコントローラ140に対して送信実行の指示をする(符号B)。これにより、PLCシステムバス送信バッファ1043および/またはフィールドネットワーク送信バッファ1045に残っている、制御サイクル1において得られた出力データが再度送信される。

【0131】

なお、代替の処理例として、上述したような送信済みの出力データを繰り返し送信するのではなく、出力データの更新ができていないことを示すデータを送信するとともに、その場合の各ユニットにおける動作を各ユニット自身に決定させるように、PLCシステムを設計してもよい。

【0132】

制御サイクル3においても、各ユニットから新たな入力データが得られるが、この入力データを利用して新たに実行開始する制御プログラム230は存在しない。そのため、制御サイクル3の入力処理プログラム216は、PLCシステムバス受信バッファ1044およびフィールドネットワーク受信バッファ1046に格納されている入力データを、制御プログラムの作業領域1042へコピーしない。仮に、このコピー動作を実行すると、制御サイクル3において制御プログラム230の実行が再開されたとき、すなわち単一の制御プログラム230の実行途中で、入力データが更新されることになるので不都合となるためである。

【0133】

制御サイクル3の入力処理プログラム216は、PLCシステムバスコントローラ120からの受信完了通知(符号C)、および、フィールドネットワークコントローラ140からの受信完了通知(D)を確認すると実行を終了する。

【0134】

図5に示す例では、制御サイクル2で実行を開始した制御プログラム230は、制御サイクル2内では終了せず、制御サイクル3において未実行の部分を実行して一連の実行を終了する。なお、制御プログラム230が制御サイクル3内においても一連の実行を終了しなければ、制御サイクル4の開始時点で再び実行を中断し、その後、制御サイクル4において未実行の部分を実行する。この場合の制御サイクル4における処理は、図5に示す制御サイクル3における処理と同様となる。

【0135】

(e5:まとめ)

10

20

30

40

50

上述のように、スケジューラプログラム 212 は、マイクロプロセッサ 100 に、制御プログラム 230 の実行が終了した制御サイクルの次の制御サイクルにおいて、通信回路（PLCシステムバスコントローラ 120 および / またはフィールドネットワークコントローラ 140）による出力データの送信および入力データの受信の後に、制御プログラム 230 の実行を開始させる命令を含む。また、スケジューラプログラム 212 は、マイクロプロセッサ 100 に、制御プログラム 230 の実行が終了しなかった制御サイクルの次の制御サイクルにおいて、制御プログラム 230 の未実行の部分を実行させる命令とを含む。さらに、スケジューラプログラム 212 は、制御サイクル内に制御プログラム 230 の実行が終了しなかった場合は、次の制御サイクルにおいて、通信回路（PLCシステムバスコントローラ 120 および / またはフィールドネットワークコントローラ 140）に、制御プログラム 230 の未実行の部分に優先して通信を開始させるための処理を実行させ、その後、マイクロプロセッサ 100 に、制御プログラム 230 の未実行の部分を実行させる命令を含む。

10

## 【0136】

本実施の形態によれば、制御プログラム 230 の実行時間が散発的に長くなる状況において、PLC 1 の CPU ユニット 13 が制御データの出力および入力のために制御サイクルと同期して行う通信の周期を、制御プログラム 230 の最大実行時間よりも短い一定時間とすることができる。

## 【0137】

PLC 1 による制御データの通信周期が一定であるので、IO ユニット 14 において外部の機器に与える出力信号の更新周期を一定化することが容易になる。これにより、外部の機器においても、PLC 1 の制御サイクルの周期に同期した動作をすることが容易になる。

20

## 【0138】

また、制御データの通信周期が一定であるので、リモート IO ターミナル 5 において、CPU ユニット 13 に入力データを送信する予定時刻の直前に外部から入力データを取得するような、精密に時間制御された計画的な動作が可能になる。これにより、CPU ユニット 13 は、入手しうる最も新しい入力データを用いて演算処理を行うことができるようになる。そのため、制御サイクルの周期の短縮化とあいまって、入力データの取得から、その入力データを用いて演算した結果である出力データを送出するまでの応答時間を短縮

30

## 【0139】

さらに、制御データの通信周期が一定であるので、PLC 1 のいずれかのユニットにおいて、予定された時刻に通信が行われなかったことを検知することによって通信異常の発生を迅速に判定することができる。

## 【0140】

さらに、本実施の形態に従う CPU ユニット 13 は、フィールドネットワーク 2 を経由して PLC 1 の外部にあるサーボモータドライバと直接通信する場合にも、フィールドネットワーク 2 による制御データの通信周期を一定化することができるので、モータを高精度に制御することができる。

40

## 【0141】

さらに、本実施の形態に従う CPU ユニット 13 では、通信回路（PLCシステムバスコントローラ 120 および / またはフィールドネットワークコントローラ 140）は、マイクロプロセッサ 100 による制御を受けて出力データの送信および入力データの受信を開始するように構成される。そして、スケジューラプログラム 212 は、通信回路に、制御サイクルごとに通信を開始させる命令を含む。このように、マイクロプロセッサ 100 が集約的に制御サイクルを管理するので、通信回路に、一定周期で通信するための特別なハードウェアが不要となるという利点もある。

## 【0142】

< F . 全体処理動作（実施の形態 2） >

50

次に、制御目的に応じて作成される制御プログラムに対して異なる優先度を設定できるようにすることも実用的である。実施の形態2においては、より高い優先度を設定された制御プログラムと、より低い優先度を設定された制御プログラムとが存在する場合の処理動作について説明する。

【0143】

図6は、本発明の実施の形態2に係る全体処理動作を示すシーケンス図である。以下では、主として、図5に示すシーケンスと相違する点について説明する。

【0144】

本実施の形態においては、これらのプログラムを対比して説明する便宜上、図5に示される制御プログラム230（第1の制御プログラム）に対してより高い優先度が設定されているとして、このプログラムを「高優先制御プログラム230-1」と称し、より低い優先度が設定されているプログラム（第2の制御プログラム）を「低優先制御プログラム230-2」と称する。すなわち、実施の形態2における高優先制御プログラム230-1は、上述の実施の形態1における制御プログラム230と実質的に同じであり、説明の便宜上、呼び方を変えただけである。

【0145】

低優先制御プログラム230-2は、高優先制御プログラム230-1よりも実行の優先度が低いため、低優先制御プログラム230-2の実行周期は、制御サイクルの周期の2以上の整数倍に設定される。図6の例では、制御サイクル4回分に相当する期間が低優先制御プログラム230-2の実行周期とされる。すなわち、低優先制御プログラム230-2の実行周期は、高優先制御プログラム230-1の周期の4倍である。なお、低優先制御プログラム230-2についても、記憶手段であるメインメモリ104および不揮発性メモリ106に格納される。

【0146】

図6に示すシーケンスにおいて、制御サイクル1～4までが低優先制御プログラム230-2の1つの実行サイクルであり、次の実行サイクルは制御サイクル5～8までとなる。

【0147】

図6を参照して、スケジューラプログラム212は、制御サイクル1において、高優先制御プログラム230-1の実行が終了すると、低優先制御プログラム230-2を実行する。このとき、低優先制御プログラム230-2は、制御サイクル1内では終了せず、送信準備割り込み（P-2）が発生することで実行が中断される。

【0148】

続く制御サイクル2において、高優先制御プログラム230-1の実行が終了しなかったとすると、低優先制御プログラム230-2に含まれる処理内容のうち未実行の部分は、制御サイクル2においては実行されない。

【0149】

続く制御サイクル3において、スケジューラプログラム212は、高優先制御プログラム230-1の実行が終了すると、低優先制御プログラム230-2に含まれる処理内容のうち未実行の部分を実行する。そして、低優先制御プログラム230-2は、制御サイクル3内で終了する。

【0150】

制御サイクル1において実行を開始した低優先制御プログラム230-2の実行周期には、その後の制御サイクル4も含まれているが、図6に示す例では、低優先制御プログラム230-2が制御サイクル3内で既に実行を終了しているため、制御サイクル4内では低優先制御プログラム230-2は実行されない。

【0151】

制御サイクル5では、低優先制御プログラム230-2の新たな実行周期になるため、制御サイクル1の場合と同様に、スケジューラプログラム212は、低優先制御プログラム230-2の実行を開始して最初から処理を実行させる。

10

20

30

40

50

## 【0152】

制御サイクル1で実行を開始した低優先制御プログラム230-2は、制御サイクル1の入力処理プログラム216の実行により得られた入力データを用いて実行される。また、低優先制御プログラム230-2の実行の結果得られた出力データは、新たな低優先制御プログラム230-2の実行サイクルの開始タイミングに相当する制御サイクル5において、出力処理プログラム214の実行により送信される。

## 【0153】

上述のように、スケジューラプログラム212は、マイクロプロセッサ100に、制御サイクル内の高優先制御プログラム230-1の終了後の期間において、実行サイクルに従って低優先制御プログラム230-2を実行させる命令を含む。

10

## 【0154】

図6に示す例では、低優先制御プログラム230-2の出力処理および入力処理を、高優先制御プログラム230-1の出力処理および入力処理とまとめて行う例を示す。しかしながら、低優先制御プログラム230-2の出力および入力のための通信を、高優先制御プログラム230-1の出力および入力のための通信タイミングとは別のタイミングで行うようにしてもよい。そのような場合であっても、高優先制御プログラム230-1の出力および入力のための通信に着目すれば、一定周期で通信が行われていることになる。

## 【0155】

上述のように、実行の優先度が異なる複数の制御プログラム230が並列的に実行される場合、制御サイクルは、最も優先度が高い制御プログラムが周期を超過せずに実行でき

20

ている場合の実行サイクルに対応する。

## 【0156】

上述したように、その実行サイクルの周期が制御サイクルの周期よりも長くなっても影響がないような処理を低優先制御プログラム230-2に含ませて実行させることにより、高優先の制御プログラム230-1に含まれる処理の処理量を削減することができる。それにより、制御サイクルの周期をより短縮できる可能性がある。また、高優先制御プログラム230-1の実行が1回の制御サイクル内に終了しない頻度を小さくできる可能性もある。さらに、高優先制御プログラム230-1の実行が想定を超える制御サイクル数内で終了しない頻度を小さくできる可能性がある。

## 【0157】

< G . 全体処理動作 ( 実施の形態 3 ) >

次に、上述の実施の形態2において説明した、高優先制御プログラムおよび低優先制御プログラムを具体的なプログラムに割り当てた場合の処理について説明する。

30

## 【0158】

図7は、本発明の実施の形態3に係る全体処理動作を示すシーケンス図である。図7に示すシーケンスでは、一例として、実施の形態2に係るシーケンスにおいて示された高優先制御プログラム230-1の内容をモーション演算プログラム234とし、低優先制御プログラム230-2の内容をユーザプログラム236およびその実行の際に必要なシーケンス命令演算プログラム232とした。以下では、主として、図5に示すシーケンスと相違する点について説明する。

40

## 【0159】

モーション演算プログラム234は、ユーザプログラム236に含まれる命令に従って実行され、実行ごとに、モータドライバに対して出力する指令値を算出する。

## 【0160】

図7に示すシーケンスにおいて、制御サイクル1~4までがユーザプログラム236の1つの実行サイクルである。それより1つ前の実行サイクルにおいて、ユーザプログラム236は、モーション演算プログラム234の起動を指示したとする。

## 【0161】

制御サイクル1では、そのユーザプログラム236の指示を受けてモーション演算プログラム234が起動する。モーション演算プログラム234は、起動すると、まずプロフ

50

ファイルデータ生成と称される初期処理を実行する。プロファイルデータは、モーション演算プログラム234の実行ごとに制御対象のそれぞれの軸についての指令値を順次算出するために必要なデータである。このプロファイルデータは、ユーザプログラム236中のモーション命令で指定されたパラメータを用いて算出される。たとえば、モーション命令で指定される、開始位置、開始速度、目標速度、加速度、加加速度(Jerk)などのパラメータを用いて、運動開始時刻からの、加速度増加の終了時間(一定加速度運動の開始時間)、加速度減少の開始時間、一定速度運動の開始時間などが算出される。さらに、モーション演算プログラム234は、その起動時の実行サイクル内において、プロファイルデータ作成に続いて、第1回目の指令値を算出する。

【0162】

多数の軸についてのモーション演算プログラム234を同時に複数起動すると、1回の制御サイクル内ではすべての軸についてのプロファイルデータの生成が完了しないことがある。図7に示すシーケンスにおいては、そのような場合を示しており、制御サイクル1内ではモーション演算プログラム234の実行が完了していない。

【0163】

制御サイクル1は、ユーザプログラム236の1つの実行サイクルに含まれる(4個の)制御サイクルのうち第1番目に相当するが、ユーザプログラム236などを実行するための余剰時間が発生しないので、制御サイクル1においては、ユーザプログラム236は実行されない。

【0164】

本実施の形態においては、モーション演算プログラム234は、ユーザにおける制御目的に応じて作成されたユーザプログラム236により起動された初回の実行において、指令値の算出を開始するために必要な初期処理を実行することにより後続の実行に要する時間よりも長い実行時間を必要とするとする。

【0165】

出力準備割り込み(P-1)に回答して、モーション演算プログラム234は実行を中断し、制御サイクル2において、スケジューラプログラム212は、出力処理プログラム214および入力処理プログラム216を実行する。その後、スケジューラプログラム212は、モーション演算プログラム234に含まれる処理内容のうち未実行の部分を実行する。

【0166】

モーション演算プログラム234が終了すると、スケジューラプログラム212は、ユーザプログラム236を実行する。ユーザプログラム236は、必要に応じてシーケンス命令演算プログラム232を呼び出して実行する。制御サイクル2内ではユーザプログラム236は終了しておらず、出力準備割り込み(P-3)に回答して、ユーザプログラム236は実行を中断する。

【0167】

制御サイクル3以降においては、モーション演算プログラム234は、指令値を算出する処理を行うだけであるので、各制御サイクル内で実行が終了する。

【0168】

図7に示すシーケンスにおいては、ユーザプログラム236が制御サイクル3内でも実行を終了できず、制御サイクル4ではじめて実行を終了する例を示す。

【0169】

なお、図7に示すシーケンスの例では、制御サイクル2において出力データ(モータドライバへの指令値を含む)は更新されないが、制御サイクル2ではまだモータが動作していないので、モータドライバへの指令値が更新されないことによる問題は少ない。図5のシーケンスに示される制御プログラム230がユーザプログラム236の他にモーション演算プログラム234をも含んでいる場合であっても同様のことがいえる。

【0170】

したがって、制御プログラム230がモーション演算プログラム234を含む場合、モ

10

20

30

40

50

ーション演算プログラム234が起動された制御サイクル(モーション起動サイクル)より後の制御サイクルにおいては、モーション演算プログラム234の実行周期の超過は発生しないが、モーション起動サイクルにおいては実行周期の超過が発生しうる程度に制御サイクルの周期を短く設定することにより、制御サイクルの周期が短く、かつ、モーション動作中における周期超過が発生しにくいモーション制御を実現できる。

【0171】

< H . 全体処理動作 ( 実施の形態 4 ) >

次に、上述の実施の形態1において説明したシーケンスにおいて、実行周期の超過を監視および通知/報知するための異常処理プログラムがさらに実装されている場合の処理について説明する。

【0172】

図8は、本発明の実施の形態4に係る全体処理動作を示すシーケンス図である。図8を参照して、異常処理プログラム240は、図3に示すその他のシステムプログラム220に含まれる1つのプログラムとして、不揮発性メモリに格納されているものとする。そして、異常処理プログラム240は、PLC1の電源投入後にメインメモリ104に読み出されて実行可能な状態になっている。以下では、主として、図5に示すシーケンスと相違する点について説明する。

【0173】

上述の実施の形態1に係るシーケンスの一例(図5)においては、たとえば、制御サイクル2で実行開始した制御プログラムが制御サイクル2内で終了せず、未実行部分を実行する制御サイクル3においても実行終了しなければ、制御サイクル4においても実行を継続した。このように制御プログラムの1回の実行を継続する制御サイクル数には限度を設けていなかった。これに対して、本実施の形態においては、単一の制御プログラム230が実行を継続している期間が制御サイクル数の限度数に達し、かつ、その限度数に対応する制御サイクル内で制御プログラム230の実行が終了しなかった場合に、次の制御サイクルにおいて、制御プログラム230に含まれる処理内容のうち未実行の部分を実行するのではなく、異常処理プログラム240を実行する。

【0174】

この限度数は、制御プログラム230の1回の実行が継続する制御サイクル数の限度数、すなわち、異常処理プログラム240を実行する条件を定義するものである。限度数データ(限度数の設定内容)については、PLC1の製造時に不揮発性メモリ106に予め格納しておいてもよいし、ユーザが限度数データを不揮発性メモリ106に格納するようにしてもよい。この限度数データの表現形態としては、限度数の値自体を指定する形式、制御プログラムの未実行部分を実行する制御サイクルの数を定する形式(限度数より1小さい)、限度数に関する予め用意された複数の選択肢のうちから1つを特定する情報(番号または文字)など、PLCが限度数を特定できる形態であればよい。

【0175】

図8に示すシーケンスには、限度数が「2」とされている場合の処理例を示す。制御サイクル2で実行を開始した制御プログラム230については、制御サイクル3が限度数に対応する制御サイクルとなる。制御サイクル2で実行を開始した制御プログラム230が制御サイクル3で実行を終了しなければ、出力準備割り込み(P-4)にตอบสนองして実行されるスケジューラプログラム212は、制御プログラム230の実行サイクル(制御サイクル)の数が限度数に達していること、および、制御プログラム230の実行が終了していないことを検出する。そして、スケジューラプログラム212は、出力処理プログラム214を実行した後、異常処理プログラム240を実行する。なお、異常処理プログラム240を実行した後には、制御プログラム230の実行を行わないようにしてもよい。

【0176】

異常処理プログラム240は、典型的には、PLC1の外部に対して実行周期の超過という異常発生を報知するための処理を行う。たとえば、異常処理プログラム240は、PLC1に搭載された異常報知用のランプ(図示しない)を点灯する、上位ネットワーク(

10

20

30

40

50

図示しない)を介してP L C 1の運用管理者に対して異常発生を示すデータを送信する、などの処理を行う。

【 0 1 7 7 】

制御サイクル5以降においては、スケジューラプログラム212は、出力処理プログラム214だけを実行する。この状態は、電源がリセットされる、あるいは、U S Bを介して接続されたP L Cサポート装置8上でユーザによって適切な異常解除処置が行われる、といった何らかの解除アクションがあるまで継続する。

【 0 1 7 8 】

上述のように、スケジューラプログラム212は、制御プログラム230が1回の実行を継続している制御サイクルの数が限度数に達し、かつ、当該制御サイクル内で制御プログラム230の実行が終了しなかった場合に、次の制御サイクルにおいて、マイクロプロセッサ100に、制御プログラム230の未実行部分を実行させることなく異常処理プログラム240を実行させる命令を含む。

【 0 1 7 9 】

本実施の形態においては、実行周期が限度数の制御サイクルを超過する異常が発生した場合であっても、出力処理プログラム214の実行は継続して、P L Cシステムバス11およびフィールドネットワーク2における一定周期の通信を維持するようにしたが、出力処理プログラム214の実行を中断するといった処理を採用することで、通信回路による一定周期の通信自体も停止させるようにしてもよい。

【 0 1 8 0 】

また、異常処理プログラム240を、システムプログラム210に含まれるプログラムとして実装するのではなく、ユーザプログラム236の中の1つの特別なプログラムとして実装することにより、異常処理の内容をユーザが自由にプログラムできるように構成してもよい。

【 0 1 8 1 】

本実施の形態によれば、想定を超える制御サイクル数まで制御プログラム230の実行が継続するといった異常状態が発生した場合には、制御プログラム230の実行を停止しつつ、異常処理プログラム240を実行することにより、より高い安全を確保することができる。

【 0 1 8 2 】

< I . 全体処理動作 ( 実施の形態 5 ) >

次に、上述の実施の形態1において説明したシーケンスにおいて、制御プログラムが制御サイクル内で終了しなかった場合に、次の制御サイクルではP L Cシステムバスの通信を行わず、フィールドネットワークの通信のみを行うようにした場合の処理について説明する。

【 0 1 8 3 】

図9は、本発明の実施の形態5に係る全体処理動作を示すシーケンス図である。以下では、主として、図5に示すシーケンスと相違する点について説明する。

【 0 1 8 4 】

図9に示すシーケンスには、制御サイクル2で実行を開始した制御プログラム230が、制御サイクル2内でその実行を終了できなかった場合を示す。この場合、出力準備割り込み(P - 3)にตอบสนองして、制御プログラム230は実行を中断する。すると、スケジューラプログラム212は、制御プログラム230の実行が終了しなかったことにตอบสนองして、制御サイクル3ではP L Cシステムバス11の通信処理を行わず、フィールドネットワーク2の通信処理のみを実行する。すなわち、P L Cシステムバスコントローラ120のDMA制御回路122は、送信実行指示(符号A)が与えられないので、メインメモリ104のP L Cシステムバス送信バッファ1043に格納されている出力データをP L Cシステムバスコントローラ120のバッファメモリ126へ転送するという処理をスキップする。

【 0 1 8 5 】

10

20

30

40

50

また、制御サイクル3においては、出力すべき更新された出力データが存在しないので、スケジューラプログラム212は、出力処理プログラム214のうち、PLCシステムバス11を介して送信される出力データを準備する部分の処理を実行しない。一方、スケジューラプログラム212は、出力処理プログラム214のうち、フィールドネットワーク送信割り込み(X-3)にตอบสนองして、フィールドネットワークコントローラ140に対して送信実行を指示する部分の処理については実行する。

フィールドネットワーク2を介して送信される出力データを準備する部分の処理については実行する。すなわち、スケジューラプログラム212は、フィールドネットワーク送信割り込み(X-3)にตอบสนองして、出力処理プログラム214を実行する。

【0186】

図9に示すシーケンスでは、スケジューラプログラム212は、出力準備割り込み(P-3)が発生した後、フィールドネットワーク送信割り込み(X-3)が発生するまでの間は、いずれのプログラムも実行しないが、制御プログラムをこの期間に実行させてもよい。その場合、制御プログラム230は、フィールドネットワーク送信割り込み(X-3)にตอบสนองして、再び実行を中断する。そして、スケジューラプログラム212は、出力処理プログラム214からの送信実行指示(符号B)にตอบสนองして、制御プログラム230の実行を再開する。すなわち、スケジューラプログラムは、制御サイクル内でフィールドネットワーク2の送信指示よりも前の時間においても制御プログラム230に含まれる処理内容のうち未実行部分を実行する。このようにすると、制御プログラム230を制御サイクル3内で終了させることができる可能性が大きくなる。

【0187】

本実施の形態によれば、いずれかの通信経路(PLCシステムバス11またはフィールドネットワーク2)で一定周期の通信が行われていれば、その通信経路との関係で本実施の形態に係る作用効果を維持できる。一般に、フィールドネットワーク2に接続される機器は、公開されたネットワーク通信仕様に基づいて多くのメーカーにより提供される。そのような機器の間で互換性を維持する必要があるので、フィールドネットワーク2の具体的仕様にも依存するが、一般に、PLCシステムバス11よりもフィールドネットワーク2の方が一定の通信周期を維持する必要性が高い場合が多い。

【0188】

<J. 全体処理動作(実施の形態6)>

次に、スケジューラプログラムによるスケジューリング処理の一部をシステムタイマに分担させることで、制御プログラムを本来の実行サイクル内で終了させる可能性を高めた構成について説明する。

【0189】

図10は、本発明の実施の形態6に係る全体処理動作を示すシーケンス図である。以下では、主として、図5に示すシーケンスと相違する点について説明する。図10には、上述のようなスケジューラプログラム212のスケジューリング処理の一部をシステムタイマ108に分担させる形態の一例を示す。具体的には、図10に示す構成は、上述の実施の形態1において説明したシーケンスにおいて、フィールドネットワークコントローラ140がスケジューラプログラム212からの送信実行指示(図5の符号B)を受信して送信動作をするのではなく、システムタイマ108からの送信トリガ信号(Y)を直接受信して送信動作をするものである。

【0190】

図10において、送信トリガ信号(Y)のタイミングは、図5に示すシーケンスにおけるフィールドネットワーク送信割り込み(X)のタイミングと同じに設定されている。すなわち、CPUユニット13は、制御サイクルの周期で信号を発生するタイマ回路をさらに含む。そして、通信回路(フィールドネットワークコントローラ140)は、タイマ回路が発生する信号をトリガとして出力データの送信および入力データの受信を開始する。

【0191】

スケジューラプログラム212は、送信準備割り込み(P)が発生すると、制御プログ

10

20

30

40

50

ラム 230 の実行が終了しているかどうかを判断し、制御プログラム 230 の実行が終了していれば出力処理プログラム 214 を実行する（制御サイクル 1, 2, 4）。これに対して、制御プログラム 230 の実行が終了していなければ、スケジューラプログラム 212 は、制御プログラム 230 に含まれる処理内容のうち未実行の部分を実行する（制御サイクル 3）。

【0192】

なお、説明の便宜上、図 10 には、PLC システムバス 11 および PLC システムバス 11 上のユニットを用いず、フィールドネットワーク 2 だけを用いて、出力データの送信および入力データの受信を実行する例を示しているが、上述の実施の形態 1 と同様に、PLC システムバス 11 を併せて用いることにしてもよい。

10

【0193】

なお、システムタイマ 108 は、フィールドネットワークコントローラ 140 内に実装されていてもよい。

【0194】

本実施の形態によれば、通信回路における通信開始は、システムタイマ 108 が発生する一定周期の割り込み信号によってトリガされるので、スケジューラプログラム 212 における処理量を低減できる。そのため、制御プログラム 230 に含まれる処理内容のうち未実行の部分を実行する制御サイクルにおいて、スケジューラプログラム 212 は、制御プログラムに含まれる処理内容のうち未実行の部分を実行するための時間をより多く割り当てることができる。その結果、制御プログラム 230 に含まれる処理内容のうち未実行の部分とその制御サイクル内での実行を終了させる可能性を高めることができる。

20

【0195】

< K . 全体処理動作（実施の形態 7） >

次に、複数の独立した演算主体を有するマイクロプロセッサを用いる場合の処理について説明する。汎用的なコンピュータアーキテクチャとして、複数のコアを有するマイクロプロセッサが実用化されている。各コアは、互いに独立して演算処理を実行できるので、それぞれのプログラムの実行をこれらのコアで分担することで、制御プログラムを本来の実行サイクル内で終了させる可能性を高めた構成について説明する。

【0196】

図 11 は、本発明の実施の形態 7 に係る全体処理動作を示すシーケンス図である。以下では、主として、図 5 に示すシーケンスと相違する点について説明する。

30

【0197】

図 11 に示すシーケンスを実行するマイクロプロセッサ 100 は、コア 1 およびコア 2 を有している。一例として、コア 1 が、スケジューラプログラム 212、出力処理プログラム 214、および入力処理プログラム 216 を実行し、コア 2 が、制御プログラム 230 を実行するものとする。

【0198】

図 11 に示す例では、制御サイクル 1, 2, 4 において、スケジューラプログラム 212 が、コア 2 に制御プログラム 230 を実行させる。

【0199】

たとえば、制御サイクル 2 において制御プログラム 230 の実行が終了しない場合には、制御サイクル 3 において、コア 2 はそのまま制御プログラム 230 の実行を継続する。すなわち、スケジューラプログラム 212 は、コア 2 が制御プログラム 230 の実行を継続することを妨げないことにより、コア 2 に制御プログラム 230 に含まれる処理内容のうち未実行の部分を実行させる。言い換えれば、スケジューラプログラム 212 は、制御プログラム 230 の実行が終了しなかった制御サイクルの次の制御サイクルにおいて、制御プログラム 230 の実行を中断しないことにより制御プログラム 230 の未実行の部分をマイクロプロセッサ 100 に実行させる命令を含む。この文脈においては、送信準備割り込み（P）があった場合でも制御プログラム 230 の実行をそのまま継続させることについての実行制御を支配しているシステムプログラム 210 の部分（たとえば、送信準備

40

50

割り込み（P）によって起動する割り込み処理プログラム、送信準備割り込み（P）があっても制御プログラム230の実行を継続させる実行制御を行うことをOS310に指示するプログラムなど）を含めて、スケジューラプログラムという。

【0200】

その結果、制御プログラム230を本来の制御サイクル内での実行を終了させる可能性を高めることができる。

【0201】

< L . サポート装置 >

次に、PLC1で実行されるプログラムの作成およびPLC1のメンテナンスなどを行うためのPLCサポート装置8について説明する。

【0202】

図12は、本発明の実施の形態に係るCPUユニットに接続して用いられるPLCサポート装置8のハードウェア構成を示す模式図である。図12を参照して、PLCサポート装置8は、典型的には、汎用のコンピュータで構成される。なお、メンテナンス性の観点からは、可搬性に優れたノート型のパーソナルコンピュータが好ましい。

【0203】

図12を参照して、PLCサポート装置8は、OSを含む各種プログラムを実行するCPU81と、BIOSや各種データを格納するROM（Read Only Memory）82と、CPU81でのプログラムの実行に必要なデータを格納するための作業領域を提供するメモリRAM83と、CPU81で実行されるプログラムなどを不揮発的に格納するハードディスク（HDD）84とを含む。

【0204】

PLCサポート装置8は、さらに、ユーザからの操作を受け取るキーボード85およびマウス86と、情報をユーザに提示するためのモニター87とを含む。さらに、PLCサポート装置8は、PLC1（CPUユニット13）などと通信するための通信インターフェイス（IF）89を含む。

【0205】

後述するように、PLCサポート装置8で実行される各種プログラムは、CD-ROM9に格納されて流通する。このCD-ROM9に格納されたプログラムは、CD-ROM（Compact Disk-Read Only Memory）ドライブ88によって読取られ、ハードディスク（HDD）84などへ格納される。あるいは、上位のホストコンピュータなどからネットワークを通じてプログラムをダウンロードするように構成してもよい。

【0206】

上述したように、PLCサポート装置8は、汎用的なコンピュータを用いて実現されるので、これ以上の詳細な説明は行わない。

【0207】

図13は、本発明の実施の形態に係るCPUユニットに接続して用いられるPLCサポート装置8のソフトウェア構成を示す模式図である。図13を参照して、PLCサポート装置8ではOS310が実行され、PLCサポートプログラム320に含まれる各種のプログラムを実行可能な環境が提供される。

【0208】

PLCサポートプログラム320は、エディタプログラム321と、コンパイラプログラム322と、デバッガプログラム323と、シミュレーション用シーケンス命令演算プログラム324と、シミュレーション用モーション演算プログラム325と、通信プログラム326とを含む。PLCサポートプログラム320に含まれるそれぞれのプログラムは、典型的には、CD-ROM9に格納された状態で流通して、PLCサポート装置8にインストールされる。

【0209】

エディタプログラム321は、ユーザプログラム236を作成するための入力および編集といった機能を提供する。より具体的には、エディタプログラム321は、ユーザがキ

10

20

30

40

50

ーボード 85 やマウス 86 を操作してユーザプログラム 236 のソースプログラム 330 を作成する機能に加えて、作成したソースプログラム 330 の保存機能および編集機能を提供する。また、エディタプログラム 321 は、外部からのソースプログラム 330 の入力を受付ける。

【0210】

コンパイラプログラム 322 は、ソースプログラム 330 をコンパイルして、CPU ユニット 13 のマイクロプロセッサ 100 で実行可能なオブジェクトプログラム形式のユーザプログラム 236 を生成する機能を提供する。また、コンパイラプログラム 322 は、ソースプログラム 330 をコンパイルして、PLC サポート装置 8 の CPU 81 で実行可能なオブジェクトプログラム形式のユーザプログラム 340 を生成する機能を提供する。このユーザプログラム 340 は、PLC サポート装置 8 によって PLC 1 の動作をシミュレート（模擬）するために使用される、シミュレーション用のオブジェクトプログラムである。

10

【0211】

デバッガプログラム 323 は、ユーザプログラムのソースプログラムに対してデバッグを行うための機能を提供する。このデバッグの内容としては、ソースプログラムのうちユーザが指定した範囲を部分的に実行する、ソースプログラムの実行中における変数値の時間的な変化を追跡する、といった動作を含む。

【0212】

デバッガプログラム 323 は、さらに、シミュレーション用のオブジェクトプログラムであるユーザプログラム 340 を実行する機能を提供する。このシミュレーション時には、CPU ユニット 13 のシステムプログラムに含まれるシーケンス命令演算プログラム 232 およびモーション演算プログラム 234 に代えて、PLC サポートプログラム 320 に含まれるシミュレーション用シーケンス命令演算プログラム 324 およびシミュレーション用モーション演算プログラム 325 が用いられる。

20

【0213】

通信プログラム 326 は、PLC 1 の CPU ユニット 13 へユーザプログラム 236 を転送する機能を提供する。

【0214】

一般的には、PLC 1 に実装されるシステムプログラム 210 は、CPU ユニット 13 の製造段階で CPU ユニット 13 の不揮発性メモリ 106 へ格納される。但し、CD-ROM 9 にシステムプログラム 210 を格納しておけば、ユーザは、CD-ROM 9 のシステムプログラム 210 を PLC サポート装置 8 へコピーし、通信プログラム 326 が提供する機能を利用してコピーしたシステムプログラム 210 を CPU ユニット 13 へ転送することもできる。さらに、CD-ROM 9 に、PLC 1 の CPU ユニット 13 で実行されるリアルタイム OS 200 を格納しておけば、リアルタイム OS 200 についてもユーザ操作によって PLC 1 へ再インストールできる。

30

【0215】

今回開示された実施の形態はすべての点で例示であって制限的なものではないと考えられるべきである。本発明の範囲は、上記した説明ではなく、特許請求の範囲によって示され、特許請求の範囲と均等の意味および範囲内でのすべての変更が含まれることが意図される。

40

【符号の説明】

【0216】

1 PLC、2 フィールドネットワーク、3 サーボモータドライバ、4 サーボモータ、5 リモート I/O ターミナル、6 検出スイッチ、7 リレー、8 PLC サポート装置、9 CD-ROM、10 接続ケーブル、11 PLC システムバス、12 電源ユニット、13 CPU ユニット、14、53 I/O ユニット、15 特殊ユニット、51 リモート I/O ターミナルバス、52 通信カプラ、81 CPU、83 RAM、85 キーボード、86 マウス、87 モニタ、88 CD-ROM ドライブ、100

50

マイクロプロセッサ、102 チップセット、104 メインメモリ、106 不揮発性メモリ、108 システムタイマ、110 USBコネクタ、120 PLCシステムバスコントローラ、122 DMA制御回路、124 PLCシステムバス制御回路、126, 146 バッファメモリ、130 PLCシステムバスコネクタ、140 フィールドネットワークコントローラ、142 DMA制御回路、144 フィールドネットワーク制御回路、200 リアルタイムOS、210 システムプログラム、212 スケジューラプログラム、214 出力処理プログラム、216 入力処理プログラム、218 IO処理プログラム、220 その他のシステムプログラム、222 制御サイクルの周期を設定するプログラム、230 制御プログラム、230-1 高優先制御プログラム、230-2 低優先制御プログラム、232 シーケンス命令演算プログラム、234 モーション演算プログラム、236 ユーザプログラム、240 異常処理プログラム、320 PLCサポートプログラム、321 エディタプログラム、322 コンパイラプログラム、323 デバッガプログラム、324 シミュレーション用シーケンス命令演算プログラム、325 シミュレーション用モーション演算プログラム、326 通信プログラム、330 ソースプログラム、340 ユーザプログラム、1041 プログラム領域、1042 制御プログラムの作業領域、1043 PLCシステムバス送信バッファ、1044 PLCシステムバス受信バッファ、1045 フィールドネットワーク送信バッファ、1046 フィールドネットワーク受信バッファ。

10

**【要約】**

**【課題】**制御プログラムの実行時間が散発的に長くなりうることを考慮して、PLCのCPUユニットが行う制御データの出力および入力のための通信の周期を、制御プログラムの最大実行時間よりも短い一定時間としうるPLCのCPUユニットを提供する。

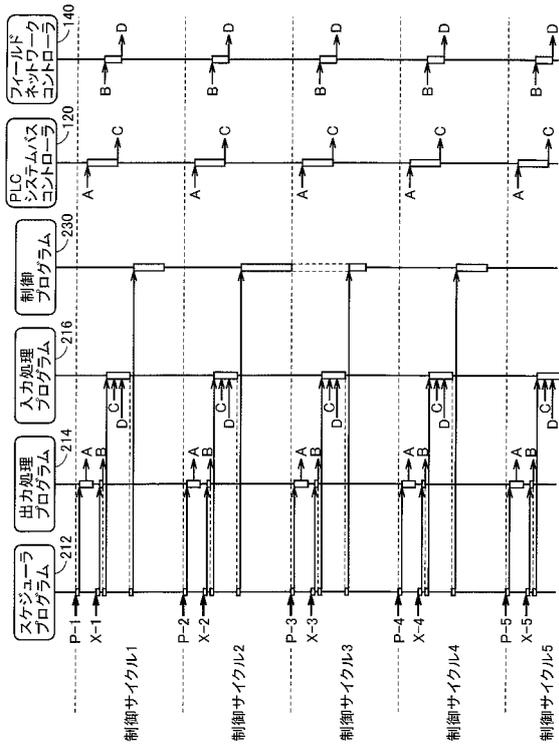
20

**【解決手段】**スケジューラプログラムは、マイクロプロセッサに、第1の制御プログラムの実行が終了した制御サイクルの次の制御サイクルにおいて、通信回路による出力データの送信および入力データの受信の後に、第1の制御プログラムの実行を開始させる命令と、マイクロプロセッサに、第1の制御プログラムの実行が終了しなかった制御サイクルの次の制御サイクルにおいて、第1の制御プログラムの未実行の部分を実行させる命令とを含む。

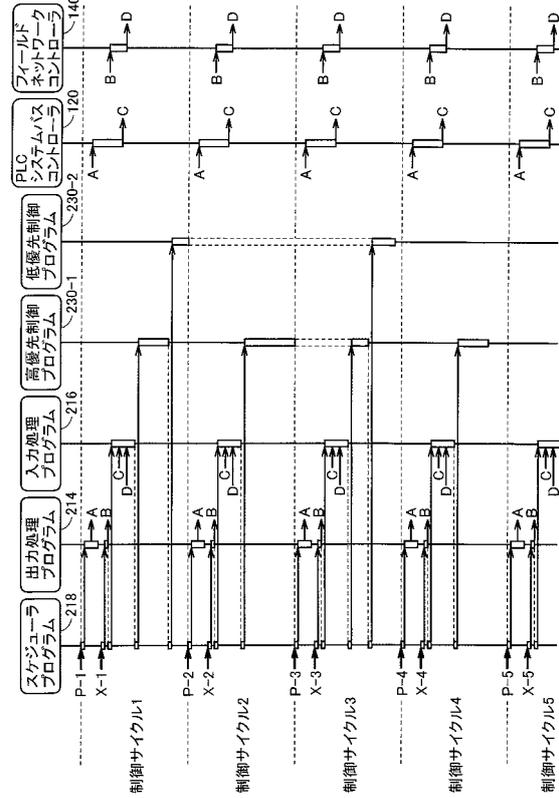
**【選択図】** 図5



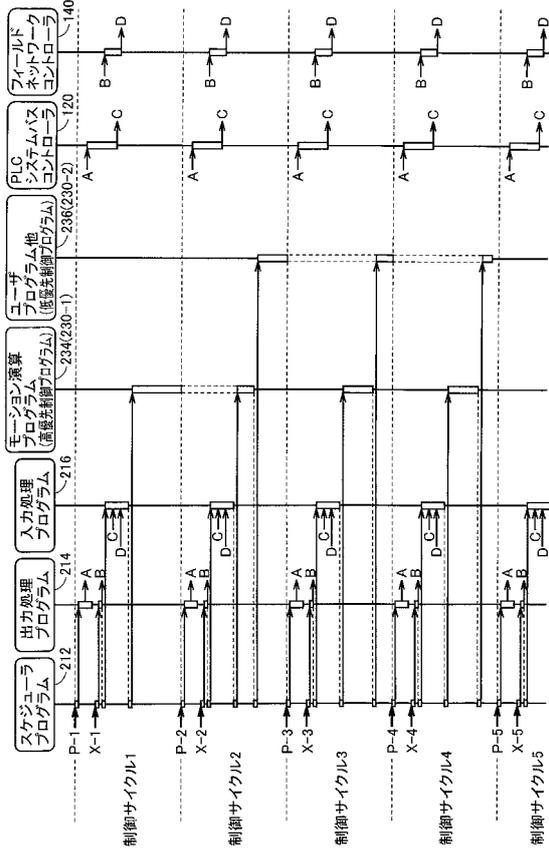
【 図 5 】



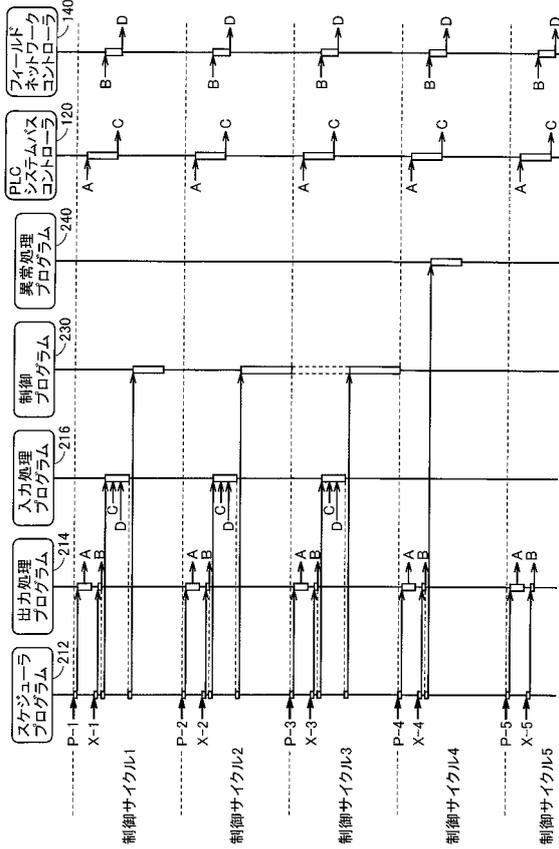
【 図 6 】



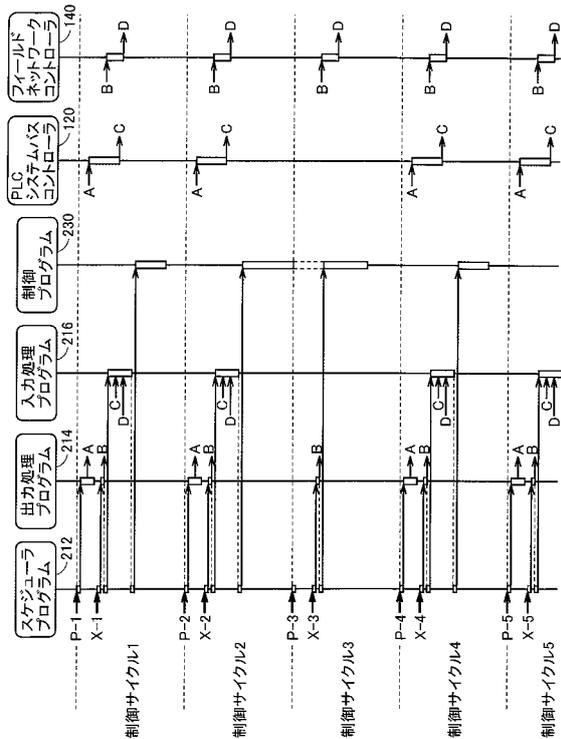
【 図 7 】



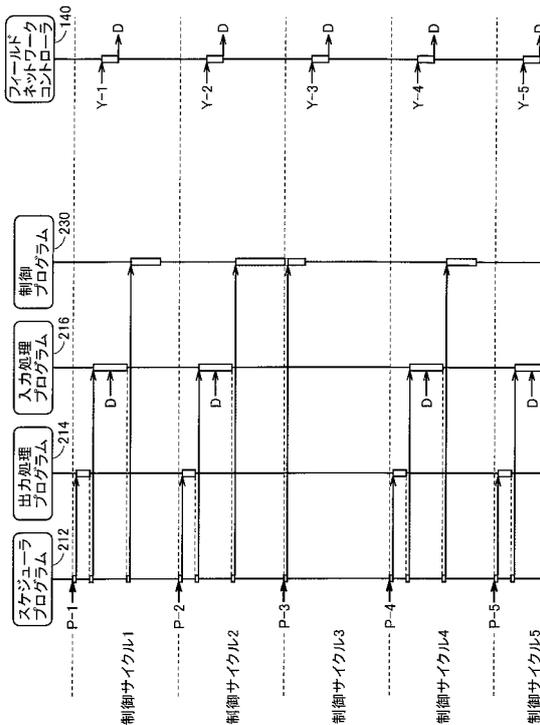
【 図 8 】



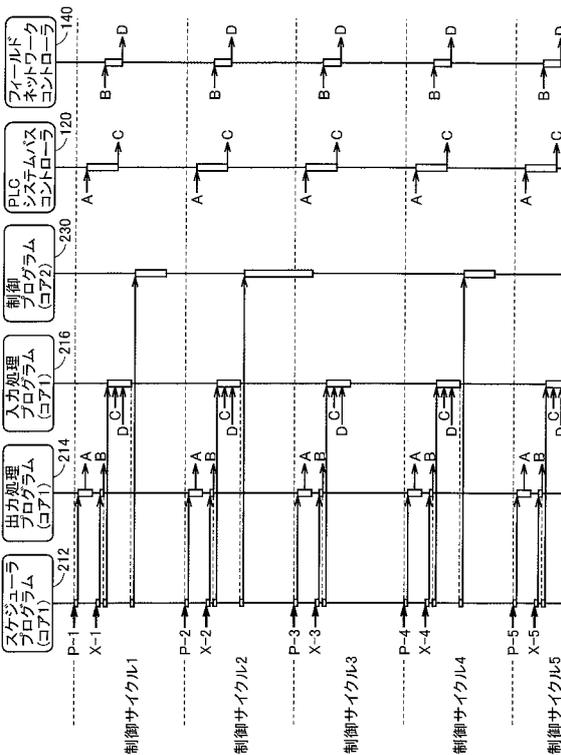
【 図 9 】



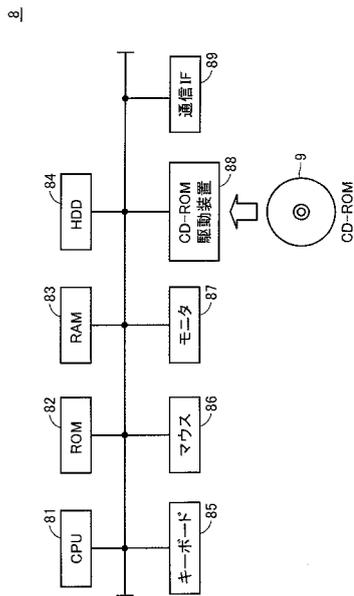
【 図 10 】



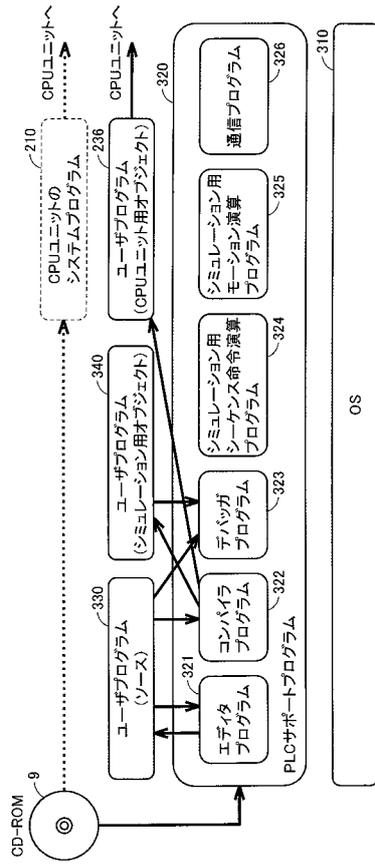
【 図 11 】



【 図 12 】



【図13】



---

フロントページの続き

(72)発明者 田村 嘉英

京都府京都市下京区塩小路通堀川東入南不動堂町801番地 オムロン株式会社内

審査官 稲垣 浩司

(56)参考文献 特開2004-038555(JP,A)

特開昭60-101612(JP,A)

特開2005-267335(JP,A)

特開2003-345406(JP,A)

特開平04-372022(JP,A)

特開2007-140655(JP,A)

特開平11-219211(JP,A)

特開2006-048231(JP,A)

特許第4752983(JP,B1)

特許第4752984(JP,B1)

(58)調査した分野(Int.Cl., DB名)

G05B 19/05