(19)日本国特許庁(JP)　(12)特　許　公　報(B2)　(11)特許番号
特許第5942446号
(P5942446)
(45)発行日　平成28年6月29日(2016.6.29)　(24)登録日　平成28年6月3日(2016.6.3)

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

請求項の数 7　(全 26 頁)

(21)出願番号　特願2012-19785(P2012-19785)
(22)出願日　平成24年2月1日(2012.2.1)
(65)公開番号　特開2013-161106(P2013-161106A)
(43)公開日　平成25年8月19日(2013.8.19)
審査請求日　平成27年1月7日(2015.1.7)

(73)特許権者　000002945
オムロン株式会社
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
(74)代理人　110001195
特許業務法人深見特許事務所
(72)発明者　小野　彰男
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　宇野　憲司
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　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
最終頁に続く

(54)【発明の名称】サポート装置およびサポートプログラム

(57)【特許請求の範囲】
【請求項１】
プロセッサとメモリとネットワークインターフェイスとを含む第１のプログラマブルロジックコントローラで実行される実行可能プログラム、を生成するサポート装置であって、
前記第１のプログラマブルロジックコントローラで扱われるデータ別に変数を定義する情報を受け付ける第１の入力手段と、
前記第１のプログラマブルロジックコントローラで実行される処理を、定義された変数を用いて記述したソースプログラムを受け付ける第２の入力手段と、
前記第１のプログラマブルロジックコントローラで扱われる第１のデータを示す第１の変数と、前記第１のプログラマブルロジックコントローラとネットワーク接続された第２のプログラマブルロジックコントローラで扱われる第２のデータを示す第２の変数とがネットワークを介して対応付けられている場合に、前記第１の変数に関連付けて前記第２のプログラマブルロジックコントローラの種別を特定する情報を受け付ける第３の入力手段と、
前記変数を定義する情報および前記ソースプログラムを用いて前記実行可能プログラムを生成する生成手段とを備え、前記生成手段は、前記第２のプログラマブルロジックコントローラの種別に応じて、前記第１の変数に対応してメモリ上に確保される前記第１のデータのデータ構造を適合化する、サポート装置。
【請求項２】

前記変数を定義する情報は、変数に対応するデータの型を定義する情報を含み、
前記生成手段は、前記第２のプログラマブルロジックコントローラにおける、第１のデータの型に対応するデータ構造に従って、前記第１のデータを前記メモリに格納するための命令を含む実行可能プログラムを生成する、請求項１に記載のサポート装置。
【請求項３】
前記実行可能プログラムは、前記第１のデータに含まれる要素の順序を入れ替えて前記メモリに格納する命令を含む、請求項２に記載のサポート装置。
【請求項４】
前記実行可能プログラムは、前記第１のデータにダミー要素を加えた上で前記メモリに格納する命令を含む、請求項２に記載のサポート装置。
【請求項５】
前記実行可能プログラムは、前記第１のデータからダミー要素を除いた上で前記メモリに格納する命令を含む、請求項２に記載のサポート装置。
【請求項６】
前記実行可能プログラムは、前記メモリに格納された第１のデータに対する出力要求に応答して、前記第１のプログラマブルロジックコントローラにおける前記第１のデータの型に対応するデータ構造に変換した上で、変換後の第１のデータを出力する命令を含む、請求項２に記載のサポート装置。
【請求項７】
プロセッサとメモリとネットワークインターフェイスとを含む第１のプログラマブルロジックコントローラで実行される実行可能プログラムを生成するサポートプログラムであって、前記サポートプログラムは、コンピュータを、
前記第１のプログラマブルロジックコントローラで扱われるデータ別に変数を定義する情報を受け付ける第１の入力手段と、
前記第１のプログラマブルロジックコントローラで実行される処理を、定義された変数を用いて記述したソースプログラムを受け付ける第２の入力手段と、
前記第１のプログラマブルロジックコントローラで扱われる第１のデータを示す第１の変数と、前記第１のプログラマブルロジックコントローラとネットワーク接続された第２のプログラマブルロジックコントローラで扱われる第２のデータを示す第２の変数とがネットワークを介して対応付けられている場合に、前記第１の変数に関連付けて前記第２のプログラマブルロジックコントローラの種別を特定する情報を受け付ける第３の入力手段と、
前記変数を定義する情報および前記ソースプログラムを用いて前記実行可能プログラムを生成する生成手段として機能させ、前記生成手段は、前記第２のプログラマブルロジックコントローラの種別に応じて、前記第１の変数に対応してメモリ上に確保される前記第１のデータのデータ構造を適合化する、サポートプログラム。
【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プログラマブルロジックコントローラ（Programmable Logic Controller：以下「ＰＬＣ」とも称す）の使用および運用を支援するためのサポート装置、およびそのサポート装置を実現するためのサポートプログラムに関する。
【背景技術】
【０００２】
機械や設備などの動作を制御するためのコントローラとしては、典型的には、ＰＬＣが用いられる。このようなＰＬＣは、ユーザが制御対象の機械や設備に応じてプログラムを自由に作成できるので汎用性が高く、広く普及している。従来のＰＬＣでは、入力データ、出力データ、内部計算用データなどは、記憶領域の予め定められた位置に格納されていた。これらのデータをプログラムにおいて利用（参照）する場合には、記憶領域の（絶対

または相対）アドレスを特定する必要があった。
【０００３】
これに対して、プログラムの再利用性や共通処理化（共通モジュール化）などを考慮して、オブジェクト指向のプログラミングが可能となりつつある。具体的には、入力データ、出力データ、内部計算用データなどを変数によって用いて利用（参照）できるようなプログラミング環境が提供されつつある。このようなプログラミング環境によって生成されるプログラムは、従来のプログラムと対比する意味で、「変数プログラム」とも称される。
【０００４】
一方で、ＰＬＣ本体の装置コストを低減するため、汎用的なデバイスが積極的に採用される傾向にある。例えば、特開２００２－２６９０２４号公報（特許文献１）には、汎用パーソナルコンピュータの技術であるプラグアンドプレイ方式が実現できるＰＬＣが開示されている。
【０００５】
このような汎用的なデバイスの採用に伴って、従来の製造メーカ別の独自規格に代えて、複数のメーカがアライアンスを形成し、共通規格を導入するという動きが活発化している。これによって、ユーザから見れば、ＰＬＣの製造メーカに依存することなく、統一的なシステムを構築できるというメリットが得られる。このような共通規格化の典型例としては、ＰＬＣ間やＰＬＣとフィールドデバイスとを接続する通信規格の共通化などが進行しつつある。
【先行技術文献】
【特許文献】
【０００６】
【特許文献１】特開２００２－２６９０２４号公報
【発明の概要】
【発明が解決しようとする課題】
【０００７】
しかしながら、各ＰＬＣのアーキテクチャや扱われるデータ構造については、依然としてメーカの独自性が存在する。すなわち、通信規格が共通化されたとしても、遣り取りされるデータのデータ構造などが一致していなければ、情報を適切に遣り取りすることができず、プログラムが本来的に意図された動作を行えない可能性もある。
【０００８】
一方で、ユーザが相手先のＰＬＣにおけるデータ構造を考慮してプログラムを作成することは容易ではない。
【０００９】
そこで、本発明は、相手先のＰＬＣの種別に依存することなく、より容易にＰＬＣ間でデータを遣り取りすることを支援するサポート装置およびサポートプログラムの提供を目的とする。
【課題を解決するための手段】
【００１０】
本発明のある局面に係るサポート装置は、プロセッサとメモリとネットワークインターフェイスとを含む第１のプログラマブルロジックコントローラで実行される実行可能プログラムを生成する。サポート装置は、第１のプログラマブルロジックコントローラで扱われるデータ別に変数を定義する情報を受け付ける第１の入力手段と、第１のプログラマブルロジックコントローラで実行される処理を、定義された変数を用いて記述したソースプログラムを受け付ける第２の入力手段と、第１のプログラマブルロジックコントローラで扱われる第１のデータを示す第１の変数と、第１のプログラマブルロジックコントローラとネットワーク接続された第２のプログラマブルロジックコントローラで扱われる第２のデータを示す第２の変数とがネットワークを介して対応付けられている場合に、第１の変数に関連付けて第２のプログラマブルロジックコントローラの種別を特定する情報を受け

付ける第３の入力手段と、変数を定義する情報およびソースプログラムを用いて実行可能プログラムを生成する生成手段とを含む。生成手段は、第２のプログラマブルロジックコントローラの種別に応じて、第１の変数に対応してメモリ上に確保される第１のデータのデータ構造を適合化する。

【００１１】

好ましくは、変数を定義する情報は、変数に対応するデータの型を定義する情報を含み、生成手段は、第２のプログラマブルロジックコントローラにおける、第１のデータの型に対応するデータ構造に従って、第１のデータをメモリに格納するための命令を含む実行可能プログラムを生成する。

【００１２】

さらに好ましくは、実行可能プログラムは、第１のデータに含まれる要素の順序を入れ替えてメモリに格納する命令を含む。

【００１３】

あるいはさらに好ましくは、実行可能プログラムは、第１のデータにダミー要素を加えた上でメモリに格納する命令を含む。

【００１４】

あるいはさらに好ましくは、実行可能プログラムは、第１のデータからダミー要素を除いた上でメモリに格納する命令を含む。

【００１５】

あるいはさらに好ましくは、実行可能プログラムは、メモリに格納された第１のデータに対する出力要求に応答して、第１のプログラマブルロジックコントローラにおける第１のデータの型に対応するデータ構造に変換した上で、変換後の第１のデータを出力する命令を含む。

【００１６】

本発明の別の局面に係るサポートプログラムは、プロセッサとメモリとネットワークインターフェイスとを含む第１のプログラマブルロジックコントローラで実行される実行可能プログラムを生成する。サポートプログラムは、コンピュータを、第１のプログラマブルロジックコントローラで扱われるデータ別に変数を定義する情報を受け付ける第１の入力手段と、第１のプログラマブルロジックコントローラで実行される処理を、定義された変数を用いて記述したソースプログラムを受け付ける第２の入力手段と、第１のプログラマブルロジックコントローラで扱われる第１のデータを示す第１の変数と、第１のプログラマブルロジックコントローラとネットワーク接続された第２のプログラマブルロジックコントローラで扱われる第２のデータを示す第２の変数とがネットワークを介して対応付けられている場合に、第１の変数に関連付けて第２のプログラマブルロジックコントローラの種別を特定する情報を受け付ける第３の入力手段と、変数を定義する情報およびソースプログラムを用いて実行可能プログラムを生成する生成手段として機能させる。生成手段は、第２のプログラマブルロジックコントローラの種別に応じて、第１の変数に対応してメモリ上に確保される第１のデータのデータ構造を適合化する。

【発明の効果】

【００１７】

本発明によれば、相手先のＰＬＣの種別に依存することなく、より容易にＰＬＣ間でデータを遣り取りすることを支援できる。

【図面の簡単な説明】

【００１８】

【図１】本発明の実施の形態に係るＰＬＣシステムを示す模式図である。

【図２】本発明の実施の形態に係るＰＬＣの処理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４】本発明の実施の形態に係るＰＬＣの処理ユニットに実装されるソフトウェア構成

を示す模式図である。
【図５】本発明の実施の形態に係るサポート装置に実装されるソフトウェア構成を示す模式図である。
【図６】本発明の実施の形態に係るサポート装置が提供する変数を定義するためのユーザインターフェイスの一例である。
【図７】本発明の実施の形態に係るＰＬＣシステムにおけるネットワークを介したデータの遣り取りを説明するための図である。
【図８】本発明の実施の形態に係るＰＬＣシステムにおけるＰＬＣ間でのデータの遣り取りに必要な対応関係の設定方法を説明するための図である。
【図９】本発明の実施の形態に係るＰＬＣシステムにおけるＰＬＣ間におけるデータ構造の相違の一例を示す図である。
【図１０】本発明の実施の形態に係るサポート装置が生成した実行可能プログラムによって実行されるＰＬＣでの処理を説明するための図である。
【図１１】本発明の実施の形態に係るサポート装置による実行可能プログラムの生成処理を説明するための図である。
【図１２】本発明の実施の形態に係るサポート装置による実行可能プログラムの生成処理の手順を示すフローチャートである。
【発明を実施するための形態】
【００１９】
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。
【００２０】
＜Ａ．システム構成＞
まず、本実施の形態に係るサポート装置が使用および運用を支援するＰＬＣを含むシステムについて説明する。
【００２１】
図１は、本発明の実施の形態に係るＰＬＣシステムＳＹＳを示す模式図である。本実施の形態に係るＰＬＣシステムＳＹＳは、ネットワークＮＷを介してネットワーク接続された複数のＰＬＣ（ＰＬＣ１，ＰＬＣ２，ＰＬＣ３）を含む。これらのＰＬＣの種別は必ずしも同一ではないとする。
【００２２】
本明細書において「ＰＬＣの種別」とは、主として、プログラムによるデータの書込み、読出し、更新などの処理において利用（参照）されるデータ構造が同一であるか否かという観点を示すものである。例えば、２つのＰＬＣの間で、いずれのデータ型であっても、一方のＰＬＣが書込んだデータを他方のＰＬＣが本来のデータの意味で解釈できる場合には、「ＰＬＣの種別が同一」であるとする。これに対して、あるデータ型について、一方のＰＬＣが書込んだデータを他方のＰＬＣが本来のデータの意味で解釈できない場合には、「ＰＬＣの種別が同一ではない（種別が別である）」であるとする。言い換えれば、あるＰＬＣがネットワークＮＷを介して自装置のデータを送信した場合に、別のＰＬＣがその送信されたデータを受信して正しく解釈できるか否かという観点に着目して、「ＰＬＣの種別」が判断される。
【００２３】
以下の説明では、典型例として、ＰＬＣ１，ＰＬＣ２，ＰＬＣ３の間では、種別が異なっているものとする。しかしながら、ＰＬＣ１とＰＬＣ２との間、ＰＬＣ２とＰＬＣ３との間、およびＰＬＣ３とＰＬＣ１との間では、ネットワークＮＷを介してデータを遣り取りする必要があるものとする。本実施の形態に係るサポート装置２００は、このような状況下において、ＰＬＣ間でのデータの遣り取りを適合化する。
【００２４】
各ＰＬＣは、プログラムを実行する主体である処理ユニット１０と、処理ユニット１０などへ電力を供給する電源ユニット１２と、フィールドからの信号を遣り取りするＩＯ（

Input/Output）ユニット１４とを含む。ＩＯユニット１４は、処理ユニット１０とシステムバス１１を介して接続されている。典型的には、ＩＯユニット１４は、フィールド機器である検出センサー６から入力信号を取得し、また処理ユニット１０でのプログラムの実行結果に応じてフィールド機器であるリレー７を駆動する。

【００２５】

サポート装置２００は、ＰＬＣで実行されるプログラム（以下「実行可能プログラム」とも称す）を生成する機能とともに、接続先のＰＬＣの運転状態や各種データの値などをモニタする機能を有している。さらに、サポート装置２００は、ユーザによる実行可能プログラムの生成を支援するため、デバック機能やシミュレーション機能を有していてもよい。

【００２６】

＜Ｂ．ハードウェア構成＞

次に、図１に示すＰＬＣシステムＳＹＳを構成するＰＬＣおよびサポート装置２００のハードウェア構成について説明する。

【００２７】

（ｂ１：ＰＬＣのハードウェア構成）

図２は、本発明の実施の形態に係るＰＬＣの処理ユニット１０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、処理ユニット１０は、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、システムバスコントローラ１２０と、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。

【００２８】

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、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、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、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。

【００２９】

処理ユニット１０は、メモリとして、メインメモリ１０４および不揮発性メモリ１０６を有する。

【００３０】

メインメモリ１０４は、揮発性の記憶領域であり、処理ユニット１０への電源投入後にプロセッサ１００で実行されるべき各種プログラムを格納する。メインメモリ１０４は、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。

【００３１】

不揮発性メモリ１０６は、リアルタイムＯＳ（Operating System）、システムプログラム、実行可能プログラムといった各種プログラム（モジュール）、およびシステム設定パラメータといったデータを不揮発的に格納する。これらのプログラムやデータは、必要に応じて、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。

【００３２】

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。

【００３３】

処理ユニット１０は、通信インターフェイスとして、システムバスコントローラ１２０およびネットワークコントローラ１４０を有する。これらの通信インターフェイスは、出力データの送信および入力データの受信を行う。

【００３４】

システムバスコントローラ１２０は、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、システムバスコントローラ１２０は、ＤＭＡ（Dynamic Memory Access）制御回路１２２と、システムバス制御回路１２４と、バッファメモリ１２６とを含む。システムバスコントローラ１２０は、システムバスコネクタ１３０を介してシステムバス１１と内部的に接続される。

【００３５】

バッファメモリ１２６は、システムバス１１を介してＩＯユニット１４へ出力されるデータの送信バッファ、および、システムバス１１を介してＩＯユニット１４から入力されるデータの受信バッファとして機能する。

【００３６】

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。

【００３７】

システムバス制御回路１２４は、システムバス１１に接続されるＩＯユニット１４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。

【００３８】

ネットワークコントローラ１４０は、ネットワークＮＷを介した他のＰＬＣとの間のデータの遣り取りを制御する。すなわち、ネットワークコントローラ１４０は、用いられるネットワーク規格に従い、出力データの送信および入力データの受信を制御する。一例として、ＯＳＩ参照モデルの物理層およびデータリンク層としてイーサネット（登録商標）を採用し、ＯＳＩ参照モデルのネットワーク層およびトランスポート層としてＴＣＰ／ＩＰまたはＵＤＰ／ＩＰを採用し、ＯＳＩ参照モデルのセッション層およびプレゼンテーション層として産業用共通プロトコル（ＣＩＰ：Common Industrial Protocol）を採用したような構成が採用される。

【００３９】

上述のようなネットワーク規格に限られることなく、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などが知られている。

【００４０】

バッファメモリ１４６は、ネットワークＮＷを介して他のＰＬＣへ出力されるデータの送信バッファ、および、ネットワークＮＷを介して他のＰＬＣから入力されるデータの受信バッファとして機能する。

【００４１】

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。

【００４２】

ネットワーク制御回路１４４は、ネットワークＮＷに接続される他のＰＬＣ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、ネットワーク制御回路１４４は、ネットワークＮＷ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。

【００４３】

ＵＳＢコネクタ１１０は、サポート装置２００と処理ユニット１０とを接続するための通信インターフェイスである。典型的には、サポート装置２００から転送される、処理ユニット１０のプロセッサ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ＰＬＣに取込まれる。

【００４４】

（ｂ２：サポート装置のハードウェア構成）

図３は、本発明の実施の形態に係るサポート装置２００のハードウェア構成を示す模式図である。図３を参照して、サポート装置２００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。

【００４５】

図３を参照して、サポート装置２００は、ＯＳを含む各種プログラムを実行するＣＰＵ２０１と、ＢＩＯＳや各種データを格納するＲＯＭ（Read Only Memory）２０２と、ＣＰＵ２０１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ＲＡＭ２０３と、ＣＰＵ２０１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）２０４とを含む。

【００４６】

サポート装置２００は、さらに、ユーザからの操作を受け付けるキーボード２０５およびマウス２０６と、情報をユーザに提示するためのモニタ２０７とを含む。サポート装置２００は、ＰＬＣ（処理ユニット１０）などと通信するための通信インターフェイス（ＩＦ）２０９を含む。

【００４７】

後述するように、サポート装置２００で実行される各種プログラムは、ＣＤ－ＲＯＭ３００に格納されて流通する。このＣＤ－ＲＯＭ３００に格納されたプログラムは、ＣＤ－ＲＯＭ（Compact Disk-Read Only Memory）ドライブ２０８によって読取られ、ハードディスク（ＨＤＤ）２０４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。

【００４８】

＜Ｃ．ソフトウェア構成＞

（ｃ１：ＰＬＣのソフトウェア構成）

次に、ＰＬＣ（処理ユニット１０）が各種機能を提供するためのソフトウェア構成について説明する。

【００４９】

図４は、本発明の実施の形態に係るＰＬＣの処理ユニット１０に実装されるソフトウェア構成を示す模式図である。図４に示す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出され、処理ユニット１０のプロセッサ１００へ提供されて実行される。

【００５０】

図４を参照して、処理ユニット１０には、リアルタイムＯＳ１９０上にＰＬＣでの処理に必要なプログラムが実装される。

【００５１】

リアルタイムＯＳ１９０は、処理ユニット１０のコンピュータアーキテクチャに応じて設計されており、プロセッサ１００がＰＬＣでの処理に必要なプログラムを実行するための基本的な実行環境を提供する。より具体的には、リアルタイムＯＳ１９０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。リアルタイムＯＳ１９０は、制御サイクル開始の割り込みが発生すると、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１６０に切り替える。

【００５２】

本実施の形態に係る実行可能プログラムとしては、基本的には、ユーザ定義アプリケーション１７０が想定されている。ユーザ定義アプリケーション１７０は、サポート装置２００によって生成され、サポート装置２００から処理ユニット１０へ転送される。なお、ユーザ定義アプリケーション１７０の実行に必要な一部のモジュール（あるいは、ライブラリ）については、処理ユニット１０に予め格納されており、これらを適切なタイミングで呼び出して利用する形態（シンボリックリンク）を採用してもよい。この場合、サポート装置２００は、一部のモジュールを含まない実行可能プログラムを生成することになる。あるいは、ユーザ定義アプリケーション１７０および命令実行モジュール１８０を含めて、実行可能プログラムとしてもよい。

【００５３】

より具体的には、処理ユニット１０には、スケジューラ１６０と、ユーザ定義アプリケーション１７０と、入力処理モジュール１７２と、出力処理モジュール１７４と、通信処理モジュール１７６と、その他のシステムモジュール１７８と、命令実行モジュール１８０と、メモリマネジャー１８４と、データ構造変換モジュール１８６とが実装される。

【００５４】

スケジューラ１６０は、ユーザ定義アプリケーション１７０、入力処理モジュール１７２、出力処理モジュール１７４、および通信処理モジュール１７６について、実行開始タイミングや処理中断を制御することで、各実行サイクルでの処理を保証する。より具体的には、１つの制御サイクル内ですべての処理を完了できないことも多く、この場合には、スケジューラ１６０は、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を区分する。すなわち、スケジューラ１６０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いるプログラムほど先に実行する。

【００５５】

ユーザ定義アプリケーション１７０は、ユーザがその制御目的に応じて作成する。すなわち、ＰＬＣシステムＳＹＳ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。より具体的には、ユーザ定義アプリケーション１７０は、サポート装置２００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。生成された実行可能（オブジェクト）プログラム形式のユーザ定義アプリケーション１７０は、サポート装置２００から接続ケーブル１３を介して処理ユニット１０へ転送され、不揮発性メモリ１０６などに格納される。

【００５６】

ユーザ定義アプリケーション１７０は、シーケンス処理およびモーション演算などの特殊処理を含み得る。ユーザ定義アプリケーション１７０は、命令実行モジュール１８０と協働して、ユーザ定義アプリケーション１７０に含まれる処理を実現する。より具体的には、ユーザ定義アプリケーション１７０は、命令実行モジュール１８０によって提供される命令や関数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。

【００５７】

命令実行モジュール１８０は、ユーザ定義アプリケーション１７０に定義された何らかのシーケンス命令あるいは何らかのファンクション命令が実行されるときに呼び出される。

【００５８】

入力処理モジュール１７２は、システムバスコントローラ１２０によって受信された入力データを、ユーザ定義アプリケーション１７０が使用するのに適した形式に再配置する。出力処理モジュール１７４は、ユーザ定義アプリケーション１７０の実行によって生成された出力データを、システムバスコントローラ１２０へ転送するのに適した形式に再配置する。

【００５９】

通信処理モジュール１７６は、ネットワークコントローラ１４０による他のＰＬＣとの間の通信処理のタイミングを制御する。通信処理モジュール１７６は、さらに、ネットワークコントローラ１４０が他のＰＬＣから受信された入力データをユーザ定義アプリケーション１７０が使用するのに適した形式に再配置するとともに、ユーザ定義アプリケーション１７０の実行によって生成された出力データを、ネットワークコントローラ１４０へ転送するのに適した形式に再配置する。

【００６０】

その他のシステムモジュール１７８は、図４に個別に示したプログラム以外の、ＰＬＣ１の各種機能を実現するための１つまたは複数のモジュールをまとめて示したものである。

【００６１】

メモリマネジャー１８４は、メインメモリ１０４に格納されるデータを管理する。

データ構造変換モジュール１８６は、他のＰＬＣとの間でデータ交換を行うために、データ構造をＰＬＣの種別毎に定義されるデータ構造に適合させる。すなわち、データ構造変換モジュール１８６は、自ＰＬＣが扱うデータのデータ構造を他のＰＬＣが扱うデータ構造に変換する。

【００６２】

上述した、実行可能プログラム（ユーザ定義アプリケーション１７０単体またはユーザ定義アプリケーション１７０および命令実行モジュール１８０）は、記憶手段であるメインメモリ１０４および／または不揮発性メモリ１０６に格納される。

【００６３】

（ｃ２：サポート装置のソフトウェア構成）

次に、サポート装置２００が各種機能を提供するためのソフトウェア構成について説明する。

【００６４】

図５は、本発明の実施の形態に係るサポート装置２００に実装されるソフトウェア構成を示す模式図である。図５に示す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出され、サポート装置２００のＣＰＵ２０１へ提供されて実行される。

【００６５】

図５を参照して、サポート装置２００には、ＯＳ２４０、プログラミングアプリケーション２５０およびネットワーク設定アプリケーション２８０が実装される。サポート装置２００ではＯＳ２４０が実行され、プログラミングアプリケーション２５０およびネットワーク設定アプリケーション２８０を実行可能な環境が提供される。本実施の形態に係るサポート装置２００を実現するためのサポートプログラムは、少なくともプログラミングアプリケーション２５０を含む。

【００６６】

プログラミングアプリケーション２５０は、エディタ２５２と、コンパイラ２５４と、デバッガ２５６と、ＧＵＩ（Graphical User Interface）モジュール２５８と、シミュレータ２６０と、データ格納部２７０とを含む。プログラミングアプリケーション２５０に含まれるそれぞれのモジュールは、典型的には、ＣＤ－ＲＯＭ３００に格納された状態で流通して、サポート装置２００にインストールされる。

【００６７】

エディタ２５２は、実行可能プログラム（ソースプログラム２７４）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２５２は、ユーザがキーボード２０５やマウス２０６を操作してユーザ定義アプリケーション１７０のソースプログラム２７４を作成する機能に加えて、作成したソースプログラム２７４の保存機能および編集機能を提供する。

【００６８】

コンパイラ２５４は、ユーザ定義アプリケーション１７０のソースプログラム２７４をコンパイルして、処理ユニット１０のプロセッサ１００で実行可能（オブジェクト）プロ

グラム形式の実行可能プログラムを生成する機能を提供する。

【００６９】

デバッガ２５６は、実行可能プログラム（ソースプログラム２７４）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２７４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２７４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。

【００７０】

ＧＵＩモジュール２５８は、ユーザが各種データやパラメータなどを入力するためのユーザインターフェイスを提供する機能を有する。

【００７１】

シミュレータ２６０は、サポート装置２００内にＰＬＣ１の処理ユニット１０でのプログラムの実行をシミュレーションする環境を構築する。

【００７２】

データ格納部２７０は、ユーザが作成したソースプログラム２７４やプログラムの実行に必要な変数設定２７２を格納する。

【００７３】

ネットワーク設定アプリケーション２８０は、ＰＬＣ間のデータの遣り取りに係る設定を行うための機能を提供する。ネットワーク設定アプリケーション２８０は、エディタ２８２と、コンフィグレータ２８４と、関連付けモジュール２８６と、データ格納部２９０とを含む。ネットワーク設定アプリケーション２８０に含まれるそれぞれのモジュールは、典型的には、ＣＤ－ＲＯＭ３００に格納された状態で流通して、サポート装置２００にインストールされる。

【００７４】

エディタ２８２は、ＰＬＣ間で遣り取りされるデータ（変数）の設定情報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２８２は、ユーザがキーボード２０５やマウス２０６を操作して、ＰＬＣ間で遣り取りされる変数などを指定する機能に加えて、入力された設定情報の保存機能および編集機能を提供する。また、エディタ２８２は、ネットワーク接続されたＰＬＣからその変数設定などを取得（ダウンロード）することもできる。

【００７５】

コンフィグレータ２８４は、ＰＬＣ間でデータを遣り取りするための設定を対象のＰＬＣへ設定する機能を提供する。このコンフィグレータ２８４によって各ＰＬＣへ設定される情報をコネクション設定２９６とも称す。

【００７６】

関連付けモジュール２８６は、遣り取りされるデータ毎に、対象のＰＬＣ間の関連付けを行う機能を提供する。すなわち、関連付けモジュール２８６は、ＰＬＣの別にコネクション設定２９６を生成する。

【００７７】

データ格納部２９０は、ＰＬＣから取得した変数の情報を示す変数設定２９２，２９４およびコネクション設定２９６を格納する。

【００７８】

＜Ｄ．変数設定＞

上述したように、本実施の形態に係るＰＬＣ（処理ユニット１０）は、プロセッサ１００と、メインメモリ１０４および不揮発性メモリ１０６と、ネットワークインターフェイスであるネットワークコントローラ１４０とを含み、サポート装置２００で生成された実行可能プログラムを実行する。この実行可能プログラムは、変数プログラムであり、サポート装置２００で作成されるソースプログラム２７４がコンパイルすることで生成される。このソースプログラム２７４では、組み合わせられる命令を記述するために、変数を用いて各データが指定される。そのため、サポート装置２００は、ＰＬＣで扱われるデータ別に変数を定義する情報を受け付ける。より具体的には、ユーザは、サポート装置２００

によって提供されるユーザインターフェイスを介して、実行可能プログラムで使用する変数、およびその変数に対応するデータの型を定義する。

【００７９】

図６は、本発明の実施の形態に係るサポート装置２００が提供する変数を定義するためのユーザインターフェイスの一例である。

【００８０】

図６を参照して、ユーザインターフェイスである変数定義画面４００は、変数やデータ型を定義するための領域４１０と、後述する他のＰＬＣとの間でネットワークＮＷを介して遣り取りするための設定を入力するための領域４２０とを含む。図６には、変数やデータ型の定義と他のＰＬＣとの間のデータ供給に係るネットワーク設定とを同一のユーザインターフェイスで提供する例を示すが、これらの入力形態は任意に設計すればよく、領域４１０に係る入力画面と、領域４２０に係る入力画面とを別々に提供するようにしてもよい。

【００８１】

より具体的には、領域４１０は、変数（名称）を入力するためのカラム４１２と、対応する変数のデータの型を入力するためのカラム４１４と、データ型が集合体（配列型変数または構造体変数）である場合に対応する変数（要素／メンバ）が集合体のうちいずれの位置に対応する変数が格納されるかを入力するためのカラム４１６および４１８を含む。すなわち、対応する変数のデータの開始点がバイト（８ビット）とビットとの組み合わせで表現され、バイト数およびビット数がそれぞれカラム４１６および４１８へ入力される。

【００８２】

カラム４１２に入力される変数（名称）は、ユーザが任意に決定することができる。

カラム４１４に入力されるデータ型としては、その使用目的に応じて予め用意された複数種類から選択される。図６には、変数「ＳｔｒＡ」に対応付けて「ＳＴＲＵＣＴ」が指定されており、これは、構造体変数を意味する。また、変数「ｍｂ」および「ｍｉ」に対応付けてそれぞれ「ＢＯＯＬ」および「ＩＮＴ」が指定されており、これらはそれぞれ「ビット」および「符号付き１ワードバイナリー」を意味する。

【００８３】

データ型としては、上述するもの以外に、ＳＩＮＴ（符号付き１バイトバイナリー）、ＩＮＴ（符号付き１ワードバイナリー）、ＤＩＮＴ（符号付き２ワードバイナリー）、ＬＩＮＴ（符号付き４ワードバイナリー）、ＵＳＩＮＴ（符号なし１バイトバイナリー）、ＵＩＮＴ（符号なし１ワードバイナリー）、ＵＤＩＮＴ（符号なし１ワードバイナリー）、ＵＬＩＮＴ（符号なし４ワードバイナリー）、ＲＥＡＬ（浮動小数点２ワード）、ＬＲＥＡＬ（浮動小数点４ワード）、ＢＹＴＥ（１６進１バイト）、ＷＯＲＤ（１６進１ワード）、ＤＷＯＲＤ（１６進２ワード）、ＬＷＯＲＤ（１６進４ワード）などを指定することができる。なお、各ＰＬＣが上述したデータ型のすべてをサポートしている必要はなく、その一部のみをサポートしている場合もある。

【００８４】

上述のように定義された変数を用いて、ユーザは、ソースプログラム２７４を作成することができる。すなわち、サポート装置２００は、ＰＬＣで実行される処理を、定義された変数を用いて記述したソースプログラム２７４を受け付ける。

【００８５】

＜Ｅ．ネットワークを介したデータの遣り取り＞

（ｅ１：概要）

次に、本実施の形態に係るＰＬＣシステムＳＹＳでは、ネットワークＮＷを介して、ＰＬＣ間でデータを遣り取りすることが可能である。このデータを遣り取りする際にも、変数を用いて対象のデータを特定することができる。

【００８６】

上述のサポート装置２００で実行されるネットワーク設定アプリケーション２８０がＰ

ＬＣ間で遣り取りされるデータを示す変数の対応関係を決定し、それぞれのＰＬＣに設定する。すなわち、図５に示すコネクション設定２９６が決定され、それぞれのＰＬＣに対して転送される。

【００８７】

まず、ＰＬＣ間で遣り取りされるデータ（変数値）は、予めネットワーク公開の対象であること（ネットワーク公開属性）が設定される。具体的には、上述の図６において、領域４２０に含まれるカラム４２２には、ネットワーク公開属性として、「出力」、「入力」、「公開のみ」といった値が設定される。これにより、ＰＬＣの内部で使用されているデータ（変数値）がネットワーク接続された他のＰＬＣからアクセス可能になる。なお、このネットワーク公開できる変数（ネットワーク変数）は、実行可能プログラムにおいてグローバル変数として使用されているものに限定してもよい。

【００８８】

この公開先の情報は、後述のＰＬＣの種別を特定する際にも利用される。すなわち、サポート装置２００は、ＰＬＣ１で用いられる変数に関連付けてＰＬＣ２の種別を特定する情報を受け付ける。

【００８９】

また、領域４２０に含まれるカラム４２４には、対応する変数の公開先を特定するための情報が格納される。この公開先を特定するための情報としては、典型的には、ＩＰアドレスやＭＡＣアドレスが用いられる。あるいは、各ＰＬＣが保有する識別情報（機種情報やメーカ情報を含む）であってもよい。

【００９０】

ネットワークＮＷ上に公開された変数（変数の名称）は「タグ」とも称され、この変数を指定してデータを遣り取りする機能は「タグ通信」機能とも称される。さらに、ＰＬＣで実行される実行可能プログラムが扱う変数に代えて、同一の変数値に対して、ネットワークＮＷ上で遣り取りするための「別名」としての変数値を設定することもできる。このような別名を設定する機能は「ネットワークエイリアス」とも称される。このようなネットワークエイリアスを用いることで、ＰＬＣ内部の変数を外部から隠蔽することができる。

【００９１】

図７は、本発明の実施の形態に係るＰＬＣシステムＳＹＳにおけるネットワークを介したデータの遣り取りを説明するための図である。より具体的には、図７（ａ）は、ＰＬＣ内部の変数がそのままネットワーク変数として利用される場合の処理を示し、図７（ｂ）は、ネットワークエイリアスを設定した場合の処理を示す。

【００９２】

図７（ａ）を参照して、ＰＬＣ１においては、内部変数「ＳｔｒＡ」が定義されるとともに、内部変数「ＳｔｒＡ」に対して、ネットワーク公開属性および公開先（この例では、ＰＬＣ２）が設定される。一方、ＰＬＣ２においては、ネットワークＮＷを介して受信するデータに対して内部変数「Ｂ＿ｉｎｐｕｔ」が定義されるとともに、内部変数「ＳｔｒＡ」に対して、ネットワーク公開属性が設定される。また、ＰＬＣ１およびＰＬＣ２には、内部変数「ＳｔｒＡ」と内部変数「Ｂ＿ｉｎｐｕｔ」との対応関係がそれぞれ設定されているものとする。これらの対応関係の設定方法については、後述する。

【００９３】

上述のような変数の定義やネットワーク属性の設定などによって、ＰＬＣ１で順次更新される内部変数「ＳｔｒＡ」に格納される値（データ）は、ＰＬＣ２においては、その内部変数「Ｂ＿ｉｎｐｕｔ」として内部処理に利用することが可能である。

【００９４】

さらに、図７（ｂ）には、ＰＬＣ１の内部変数「ＳｔｒＡ」をネットワーク変数「Ｔａｇ１」としてネットワーク公開する例を示す。このとき、ネットワーク変数を定義するためのネットワーク変数テーブルに、内部変数「ＳｔｒＡ」がネットワーク変数「Ｔａｇ１」に対応することが定義されるとともに、対応するネットワーク公開属性（この例では「

出力」）が設定される。ＰＬＣ２は、ネットワークＮＷ上でネットワーク変数「Ｔａｇ１」を指定することで、ＰＬＣ１の内部変数「ＳｔｒＡ」に格納される値（データ）を内部処理に利用することができる。

【００９５】

（ｅ２：設定）

次に、上述のようなＰＬＣ間でのデータの遣り取りに必要な対応関係（名前解決）の設定について説明する。

【００９６】

図８は、本発明の実施の形態に係るＰＬＣシステムＳＹＳにおけるＰＬＣ間でのデータの遣り取りに必要な対応関係の設定方法を説明するための図である。サポート装置２００で実行されるプログラミングアプリケーション２５０は、それぞれＰＬＣ１およびＰＬＣ２で使用される変数を定義する変数設定１および変数設定２を保持しているものとする。プログラミングアプリケーション２５０は、それぞれのＰＬＣの変数設定２９２，２９４を比較して変数間の対応関係を決定し、その対応関係を反映したコネクション設定２９６を生成する。

【００９７】

説明の便宜上、図８においては、単一のサポート装置２００において２つのプログラミングアプリケーション２５０が実行される例を示すが、ＰＬＣの別にサポート装置２００が用意される場合には、各サポート装置２００においてプログラミングアプリケーション２５０が実行される。この場合には、公知の方法を用いて、それぞれのＰＬＣの変数設定がネットワーク設定アプリケーション２８０へ出力（ｉｍｐｏｒｔ）される。

【００９８】

そして、ネットワーク設定アプリケーション２８０は、生成したコネクション設定２９６に基づいて、ＰＬＣ１およびＰＬＣ２のそれぞれに対して、内部変数とネットワーク変数との対応関係を定義するタグデータリンク設定を出力（ダウンロード）する。

【００９９】

ＰＬＣ１およびＰＬＣ２は、ネットワーク設定アプリケーション２８０から受信したタグデータリンク設定に従ってネットワークを構成することで、他のＰＬＣとの間でデータを遣り取りできる。なお、タグデータリンク設定は、タグ（変数／変数の名称）、タグセット（タグ同士の対応関係）、コネクション設定などを含む。

【０１００】

このように、サポート装置２００は、ＰＬＣ１で扱われるデータを示す変数（タグ）と、ＰＬＣ１とネットワーク接続されたＰＬＣ２で扱われるデータを示す変数（タグ）とをネットワークＮＷを介して対応付けるための対応関係を受け付ける。そして、サポート装置２００は、この対応関係に基づいて、データを遣り取りするそれぞれのＰＬＣについてのネットワークを設定する。

【０１０１】

＜Ｆ．データ構造の相違＞

上述のようなネットワークＮＷを介したデータの遣り取りにおいて、対象のデータを各ＰＬＣの内部で扱うデータのまま他のＰＬＣへ送信されるような形態を考える。すなわち、ＰＬＣ内部のメモリデータそのものが他のＰＬＣへ渡される。この場合、ＰＬＣ間における扱うデータ構造（データフォーマット）の相違によって、データを適切に遣り取りできない場合がある。図９には、このようなデータ構造の相違の一例を示す。

【０１０２】

図９は、本発明の実施の形態に係るＰＬＣシステムＳＹＳにおけるＰＬＣ間におけるデータ構造の相違の一例を示す図である。図９には、集合体のデータ型の例を示すが、これに限られることなく、様々なデータ構造の相違が考えられる。後述するように、本実施の形態に係るサポート装置２００は、このようなデータ構造の相違を適合化した実行可能プログラムを生成する。

【０１０３】

図９（ａ）には、構造体変数が定義されるとともに、ＢＯＯＬ（ビット）である変数ａおよび変数ｂがそのメンバとして定義される例を示す。例えば、ＰＬＣ１では、その構造体変数として２ワード分の領域が確保されるとともに、１ワード目の最下位ビットに変数ａの値が格納され、２ワード目の最下位ビットに変数ｂの値が格納されるものとする。

【０１０４】

これに対して、同様の構造体変数を定義した場合であっても、ＰＬＣ２およびＰＬＣ３内部のメモリデータは異なったフォーマットであるとする。例えば、ＰＬＣ２では、その構造体変数として２ワード分の領域が確保されるとともに、１ワード目の最下位ビットに変数ａの値が格納され、当該最下位ビットの次に位置するビットに変数ｂの値が格納されるものとする。また、ＰＬＣ３では、その構造体変数として１ワード分の領域が確保されるとともに、最下位ビットに変数ａの値が格納され、当該最下位ビットの次に位置するビットに変数ｂの値が格納されるものとする。

【０１０５】

図９（ｂ）には、構造体変数が定義されるとともに、ＩＮＴ（符号付き１ワードバイナリー）である変数ｗ０がそのメンバとして定義される例を示す。例えば、ＰＬＣ１では、その構造体変数として１ワード分の領域が確保されるとともに、その全体に変数ｗ０のデータ値（１ワード）が格納されるものとする。また、ＰＬＣ３においても同様のデータ構造が採用される。

【０１０６】

これに対して、ＰＬＣ２では、その構造体変数として２ワード分の領域が確保されるとともに、１ワード目に変数ｗ０のデータ値（１ワード）が格納され、２ワード目はｎｕｌｌ（あるいは、ダミーデータ）になるものとする。

【０１０７】

図９（ｃ）には、構造体変数が定義されるとともに、ＩＮＴ（符号付き１ワードバイナリー）である変数ｗ０およびＤＩＮＴ（符号付き２ワードバイナリー）である変数ｄ０がそのメンバとして定義される例を示す。例えば、ＰＬＣ１では、その構造体変数として４ワード分の領域が確保されるとともに、１ワード目に変数ｗ０のデータ値（１ワード）が格納され、２ワード目はｎｕｌｌになり、３ワード目および４ワード目に変数ｄ１のデータ値（２ワード）が格納されるものとする。また、ＰＬＣ２においても同様のデータ構造が採用される。

【０１０８】

これに対して、ＰＬＣ３では、その構造体変数として３ワード分の領域が確保されるとともに、１ワード目に変数ｗ０のデータ値（１ワード）が格納され、２ワード目および３ワード目に変数ｄ１のデータ値（２ワード）が格納されるものとする。

【０１０９】

上述したように、ＰＬＣ１，ＰＬＣ２，ＰＬＣ３の間では、一部のデータ型のデータ構造については共通であるが、他のデータ型のデータ構造については相違しているような状況が生じる。

【０１１０】

＜Ｇ．データ構造の適合化処理＞

上述したように、本実施の形態に係るＰＬＣの処理ユニット１０は、タグ名を指定してネットワーク変数の読出し／書込みを行う機能を有する。処理ユニット１０（図４の通信処理モジュール１７６）は、変数にデータ値を書込むときには、変数のデータ型やチェックビット（構造体変数の場合は、ＣＲＣ：Cyclic Redundancy Check）をチェックして、意図しないデータ型へのデータ値の書込みを防止する機能を有する。なお、構造体変数のＣＲＣは、構造体変数のデータ型の情報に基づいて動的に算出される。また、処理ユニット１０（図４の通信処理モジュール１７６）は、受信したメッセージについてのＣＲＣチェックを行う。

【０１１１】

上述したように、ＰＬＣ間でデータが遣り取りされる場合には、ＰＬＣ内部の変数がそ

のままネットワーク変数として送出される。そのため、ＰＬＣ間でデータ構造が異なっていると、上述のようなデータ型やＣＲＣのチェックでエラーが発生する。また、データ構造が異なっているため、ＰＬＣ間で本来的に意図されたデータの遣り取りを行うことができない。

【０１１２】

そこで、本実施の形態に係るサポート装置２００は、上述したようなデータ構造（データのアライメント）の相違を吸収して、ユーザが意識しなくとも、任意のＰＬＣ間でのデータの遣り取りを行うことができる実行可能プログラムを生成する。以下の説明では、ＰＬＣ２およびＰＬＣ３は、固有のデータ構造のまま処理を行うものとし、ＰＬＣ１をＰＬＣ２およびＰＬＣ３に適合させるような形態を考える。但し、いずれのＰＬＣを適合させてもよい。

【０１１３】

図１０は、本発明の実施の形態に係るサポート装置２００が生成した実行可能プログラムによって実行されるＰＬＣ１での処理を説明するための図である。図１０に示す例では、図９（ｃ）に示すようなデータ構造の変数を、ＰＬＣ１とＰＬＣ２との間、および、ＰＬＣ１とＰＬＣ３との間で遣り取りする場合の処理例を示す。図９（ｃ）に示すように、ＰＬＣ１とＰＬＣ２との間ではデータ構造は一致しているが、ＰＬＣ１とＰＬＣ３との間ではデータ構造は相違しているとする。

【０１１４】

（１：ＰＬＣ１とＰＬＣ２との間）

まず、ＰＬＣ１とＰＬＣ２との間で遣り取りされる変数（構造体変数）について説明する。この場合、データ構造の相違を考慮する必要がないので、通常の処理が実行される。

【０１１５】

より具体的には、ＰＬＣ１における実行可能プログラムの実行によって、対象の変数値が更新されると（処理Ｐ１１）、処理ユニット１０のプロセッサ１００は、当該更新された変数値が格納されるメモリ（典型的には、メインメモリ１０４）上のアドレスを特定する（処理Ｐ１２Ａ）。そして、プロセッサ１００は、整合性チェック（処理Ｐ１３Ａ）を行った後、当該更新値を指定されたメモリ上のアドレスに書込む（処理Ｐ１４Ａ）。

【０１１６】

また、ＰＬＣ１における実行可能プログラムの実行によって、いずれかの変数値が要求されると、処理ユニット１０のプロセッサ１００は、メモリに格納されているデータ（変数値）を読出し（処理Ｐ２１Ａ）、読出したデータのＣＲＣなどに基づいてエラーチェック（処理Ｐ２２Ａ）を行った上で、当該変数値を応答する（処理Ｐ２３）。

【０１１７】

また、ＰＬＣ２からネットワーク変数のデータ（変数値）を受信すると（処理Ｐ３１）、処理ユニット１０のプロセッサ１００は、当該受信された変数値が格納されるべきメモリ（典型的には、メインメモリ１０４）上のアドレスを特定する（処理Ｐ３２Ａ）。そして、プロセッサ１００は、整合性チェック（処理Ｐ３３Ａ）を行った後、当該更新値を指定されたメモリ上のアドレスに書込む（処理Ｐ３４Ａ）。

【０１１８】

また、ＰＬＣ２からネットワーク変数のデータ（変数値）が要求されると、処理ユニット１０のプロセッサ１００は、メモリに格納されているデータ（変数値）を読出し（処理Ｐ４１Ａ）、読出したデータにＣＲＣなどにチェックビットを付加（処理Ｐ４２Ａ）する。そして、プロセッサ１００は、要求されたデータ（変数値）の送信先を特定（処理Ｐ４３）した上で、当該変数値を送信する（処理Ｐ４４）。

【０１１９】

（２：ＰＬＣ１とＰＬＣ３との間）

これに対して、データ構造の異なるＰＬＣ３との間でデータを遣り取りする場合には、以下のような手順となる。

【０１２０】

すなわち、ＰＬＣ１における実行可能プログラムの実行によって、対象の変数値が更新されると（処理Ｐ１１）、処理ユニット１０のプロセッサ１００は、当該更新された変数値が格納されるメモリ（典型的には、メインメモリ１０４）上のアドレスを特定する（処理Ｐ１２Ｂ）。この際、メモリ上にはＰＬＣ３で扱われるデータ構造に対応してデータが格納されるので、アドレスの特定には、ＰＬＣ１で扱われるデータ構造に応じた算出方法ではなく、ＰＬＣ３で扱われるデータ構造に応じた算出方法が採用される。そして、プロセッサ１００は、整合性チェックを行う（処理Ｐ１３Ｂ）。この整合性チェックについても、ＰＬＣ３で扱われるデータ構造に応じた算出方法が採用される。さらに、プロセッサ１００は、当該更新値を指定されたメモリ上のアドレスに書込む（処理Ｐ１４Ｂ）。この際、プロセッサ１００は、ＰＬＣ３で扱われるデータ構造に応じたデータ構造でメモリにデータを書込む。すなわち、データ構造が変換される。このように、ＰＬＣ１で実行される実行可能プログラムは、ＰＬＣ１の対象のデータ型に対応する他の他のＰＬＣにおけるデータ構造に従って、対象のデータ（変数値）をメモリに格納するための命令を含む。

【０１２１】

また、ＰＬＣ１における実行可能プログラムの実行によって、いずれかの変数値が要求されると、処理ユニット１０のプロセッサ１００は、メモリに格納されているデータ（変数値）を読出し（処理Ｐ２１Ｂ）、読出したデータのＣＲＣなどに基づいてエラーチェックを行う（処理Ｐ２２Ｂ）。データ（変数値）の読出しにおいては、ＰＬＣ１で扱われるデータ構造への変換が行われる。また、このエラーチェックについても、ＰＬＣ３で扱われるデータ構造に応じた算出方法が採用される。そして、プロセッサ１００は、当該変数値を応答する（処理Ｐ２３）。

【０１２２】

また、ＰＬＣ３からネットワーク変数のデータ（変数値）を受信すると（処理Ｐ３１）、処理ユニット１０のプロセッサ１００は、当該受信された変数値が格納されるべきメモリ（典型的には、メインメモリ１０４）上のアドレスを特定する（処理Ｐ３２Ｂ）。この際、アドレスの特定には、ＰＬＣ１で扱われるデータ構造に応じた算出方法ではなく、ＰＬＣ３で扱われるデータ構造に応じた算出方法が採用される。そして、プロセッサ１００は、整合性チェックを行う（処理Ｐ３３Ｂ）。この整合性チェックについても、ＰＬＣ３で扱われるデータ構造に応じた算出方法が採用される。さらに、プロセッサ１００は、当該更新値を指定されたメモリ上のアドレスに書込む（処理Ｐ３４Ｂ）。

【０１２３】

また、ＰＬＣ３からネットワーク変数のデータ（変数値）が要求されると、処理ユニット１０のプロセッサ１００は、メモリに格納されているデータ（変数値）を読出し（処理Ｐ４１Ｂ）、読出したデータにＣＲＣなどにチェックビットを付加（処理Ｐ４２Ｂ）する。データ（変数値）の読出しおよびエラーチェックについても、ＰＬＣ３で扱われるデータ構造に応じた算出方法が採用される。そして、プロセッサ１００は、要求されたデータ（変数値）の送信先を特定（処理Ｐ４３）した上で、当該変数値を送信する（処理Ｐ４４）。

【０１２４】

上述したように、ＰＬＣ１で実行される実行可能プログラムは、ＰＬＣ１の対象のデータ型に対応する他の他のＰＬＣにおけるデータ構造に従って、対象のデータ（変数値）をメモリに格納するための命令を含む。このようなデータ構造に従って対象のデータ（変数値）をメモリに格納するための命令としては、以下のような命令がある。

【０１２５】

（ａ）ＰＬＣ１で扱われるデータ（変数値）に含まれる要素（メンバ）の順序を入れ替えてメモリに格納する命令（図９（ａ）に示すデータ構造などに適用される）

（ｂ）ＰＬＣ１で扱われるデータ（変数値）にダミー要素を加えた上でメモリに格納する命令（図９（ｂ）や図９（ｃ）に示すデータ構造などに適用される）

（ｃ）ＰＬＣ１で扱われるデータ（変数値）からダミー要素を除いた上でメモリに格納する命令（図９（ｂ）や図９（ｃ）に示すデータ構造などに適用される）

なお、ＰＬＣ１に外部装置（パーソナルコンピュータや表示装置）などが接続される場合には、基本的には、ＰＬＣ１で扱われるデータ構造のデータ（変数値）が出力されることが好ましい。そのため、ＰＬＣ１で扱われるデータ構造とは異なるデータ構造で格納されているデータ（変数値）については、外部装置へ出力する際に、本来のデータ構造で出力されることが好ましい。図１０には、このような場合の処理についても図示する。

【０１２６】

すなわち、外部装置からネットワーク変数のデータ（変数値）を受信すると（処理Ｐ３１）、処理ユニット１０のプロセッサ１００は、当該受信された変数値が格納されるべきメモリ（典型的には、メインメモリ１０４）上のアドレスを特定する（処理Ｐ３２Ｃ）。この際、アドレスの特定には、現在メモリに格納されているデータ構造に応じた算出方法が採用される。そして、プロセッサ１００は、整合性チェックを行う（処理Ｐ３３Ｃ）。この整合性チェックについても、現在メモリに格納されているデータ構造に応じた算出方法が採用される。さらに、プロセッサ１００は、当該更新値を指定されたメモリ上のアドレスに書込む（処理Ｐ３４Ｃ）。さらに、プロセッサ１００は、当該受信された変数値を指定されたメモリ上のアドレスに書込む（処理Ｐ３４Ｃ）。この際、プロセッサ１００は、現在メモリに格納されているデータ構造に応じたデータ構造でメモリにデータを書込む。すなわち、データ構造が変換される。

【０１２７】

また、外部装置からネットワーク変数のデータ（変数値）が要求されると、処理ユニット１０のプロセッサ１００は、メモリに格納されているデータ（変数値）を読出し（処理Ｐ４１Ｃ）、読出したデータにＣＲＣなどにチェックビットを付加（処理Ｐ４２Ｃ）する。データ（変数値）の読出しにおいては、ＰＬＣ１で扱われるデータ構造への変換が行われる。また、このエラーチェックについても、現在メモリに格納されているデータ構造に応じた算出方法が採用される。そして、プロセッサ１００は、要求されたデータ（変数値）の送信先を特定（処理Ｐ４３）した上で、当該変数値を送信する（処理Ｐ４４）。

【０１２８】

上述したように、ＰＬＣ１で実行される実行可能プログラムは、ＰＬＣ１のメモリに格納されたデータに対する出力要求に応答して、ＰＬＣ１における指定されたデータの型に対応するデータ構造に変換した上で、変換後のデータを出力する命令を含む。

【０１２９】

＜Ｈ．実行可能プログラムの生成処理＞

次に、ＰＬＣ１の処理ユニット１０（プロセッサ１００）で実行される実行可能プログラムの生成処理について説明する。上述したように、サポート装置２００が実行可能プログラムを生成する。

【０１３０】

（ｈ１：機能構成）

図１１は、本発明の実施の形態に係るサポート装置２００による実行可能プログラムの生成処理を説明するための図である。図１１に示す機能ブロックは、主としてサポート装置２００のＣＰＵ２０１がプログラミングアプリケーション２５０を実行することによって実現される。なお、プログラミングアプリケーション２５０を構成するコンパイラ２５４は、その機能モジュールとして、パーサ２５４２と、オブジェクトファイル生成部２５４４と、データ構造特定部２５４６と、リンカ２５４８とを含む。

【０１３１】

図１１を参照して、コンパイラ２５４のパーサ２５４２は、ソースプログラム２７４と、内部変数定義２７２１およびネットワーク公開属性２７２２を含む変数設定２７２とを受け付け、構文解析を実行する。ソースプログラム２７４は、ＰＬＣ１で実行される処理を定義された（内部）変数を用いて記述したものである。内部変数定義２７２１は、ＰＬＣ１で扱われるデータ別に（内部）変数を定義する情報である。ネットワーク公開属性２７２２は、内部変数に関連付けて相手先のＰＬＣの種別を特定する情報（ＩＰアドレスなど）を含む。

【０１３２】
パーサ２５４２は、構文の解析結果をオブジェクトファイル生成部２５４４へ出力する。オブジェクトファイル生成部２５４４は、実行可能プログラムを生成するための中間的なネイティブコードであるオブジェクトファイルを出力する。上述したように、ＰＬＣ１は、ＰＬＣ間で遣り取りされる変数について、相手先のＰＬＣで扱うデータ構造に応じたデータ構造でデータ（変数値）を維持する。そのため、オブジェクトファイル生成部２５４４は、各変数に係るオブジェクトファイルを生成する際に、ネットワーク変数であるか否かの情報、および遣り取りされるＰＬＣにおけるデータ構造の情報に基づいて、対応するデータ構造を適合化する。
【０１３３】
より具体的には、オブジェクトファイル生成部２５４４は、パーサ２５４２からの解析結果に基づいて、内部変数定義２７２１で定義された変数のうちネットワーク公開属性が設定されているものを抽出する。そして、オブジェクトファイル生成部２５４４は、ネットワーク公開属性が設定されていない内部変数については、通常のデータ構造となるようにオブジェクトファイルを生成する。すなわち、上述の図１０においてＰＬＣ２との間で遣り取りされる変数についての処理と同様の処理が実行されるようにオブジェクトファイルが生成される。
【０１３４】
これに対して、ネットワーク公開属性が設定されている内部変数について、オブジェクトファイル生成部２５４４は、データ構造特定部２５４６からデータ構造情報を取得する。そして、オブジェクトファイル生成部２５４４は、ネットワーク公開属性が設定されている内部変数については、適切なデータ構造となるようにオブジェクトファイルを生成する。すなわち、上述の図１０においてＰＬＣ３との間で遣り取りされる変数についての処理と同様の処理が実行されるようにオブジェクトファイルが生成される。
【０１３５】
データ構造特定部２５４６は、ネットワーク公開属性が設定されている内部変数のデータ型に基づいて、当該内部変数に設定すべきデータ構造の情報を特定し、その値（データ構造情報）をオブジェクトファイル生成部２５４４へ返す。
【０１３６】
より具体的には、データ構造特定部２５４６は、ネットワーク公開属性２７２２を参照して、対象の内部変数と関連付けられているＰＬＣの情報を取得する。典型的には、ＰＬＣのＩＰアドレスである。
【０１３７】
そして、データ構造特定部２５４６は、取得したＩＰアドレスに基づいて、ネットワーク接続情報２７６を参照することで、対応するＰＬＣの種別情報を取得する。ネットワーク接続情報２７６は、典型的には、ＰＬＣシステムＳＹＳのネットワークＮＷに接続されているそれぞれのＰＬＣの機種情報と割り当てられているＩＰアドレスとの対応関係を記述する。なお、ネットワーク接続情報２７６については、ＰＬＣシステムＳＹＳの管理ユーザが設定してもよいし、公知の方法を用いて動的に生成してもよい。
【０１３８】
さらに、データ構造特定部２５４６は、ＰＬＣの種別毎に予め規定されたデータ構造情報２７８を参照して、対象の種別に対応するデータ構造情報を取得し、指定されたデータ型に対応するデータ構造をオブジェクトファイル生成部２５４４へ返す。データ構造情報２７８は、ＰＬＣ種別毎に、各データ型と対応するデータ構造とを定義する。
【０１３９】
オブジェクトファイル生成部２５４４がすべての内部変数についてのオブジェクトファイルを生成し、さらに必要なオブジェクトファイルを生成すると、リンカ２５４８がこれらのオブジェクトファイルをまとめて、実行可能プログラムを生成する。
【０１４０】
上述のように、サポート装置２００で実行されるプログラミングアプリケーション２５

０は、内部変数を定義する情報（内部変数定義２７２１）およびソースプログラム２７４を用いて実行可能プログラムを生成する。このとき、生成手段であるプログラミングアプリケーション２５０は、相手先のＰＬＣの種別に応じて、ＰＬＣ１の変数に対応してメモリ上に確保されるデータ（変数値）のデータ構造を適合化する。

【０１４１】

（ｈ２：処理手順）

次に、サポート装置２００が実行可能プログラムを生成する処理手順について説明する。

【０１４２】

図１２は、本発明の実施の形態に係るサポート装置２００による実行可能プログラムの生成処理の手順を示すフローチャートである。図１２に示す処理手順は、サポート装置２００のＣＰＵ２０１がプログラミングアプリケーション２５０を実行することによって実現される。

【０１４３】

図１２を参照して、サポート装置２００のＣＰＵ２０１は、ユーザが作成した変数設定２７２およびソースプログラム２７４を受け付ける（ステップＳ１００）。ＣＰＵ２０１は、変数設定２７２を参照して、受け付けたソースプログラム２７４についての構文解析を行い、使用されている内部変数を抽出する（ステップＳ１０２）。

【０１４４】

続いて、ＣＰＵ２０１は、ステップＳ１０２において抽出した内部変数のうち最初の内部変数を処理対象に選択する（ステップＳ１０４）。そして、ＣＰＵ２０１は、処理対象に選択した内部変数に対してネットワーク公開属性が設定されているか否かを判断する（ステップＳ１０６）。

【０１４５】

ネットワーク公開属性が設定されていない場合（ステップＳ１０６においてＮＯの場合）には、ＣＰＵ２０１は、処理対象の内部変数に対して通常のデータ構造で読出し／書込みされるようにオブジェクトファイルを生成する（ステップＳ１０８）。そして、処理は、ステップＳ１１６へ進む。

【０１４６】

これに対して、ネットワーク公開属性が設定されている場合（ステップＳ１０６においてＹＥＳの場合）には、ＣＰＵ２０１は、処理対象に選択した内部変数のネットワーク公開属性に基づいて、接続先のＰＬＣの種別情報を取得する（ステップＳ１１０）。続いて、ＣＰＵ２０１は、取得した種別情報に対応するデータ構造情報を取得し、さらに、取得したデータ構造情報における処理対象の内部変数のデータ型に対応するデータ構造を特定する（ステップＳ１１２）。ＣＰＵ２０１は、処理対象の内部変数に対して特定したデータ構造で読出し／書込みされるようにオブジェクトファイルを生成する（ステップＳ１１４）。そして、処理は、ステップＳ１１６へ進む。

【０１４７】

ＣＰＵ２０１は、ステップＳ１０２において抽出した内部変数のすべてについての処理が完了したか否かを判断する（ステップＳ１１６）。ステップＳ１０２において抽出した内部変数のすべてについての処理は完了していない場合（ステップＳ１１６においてＮＯの場合）には、ＣＰＵ２０１は、ステップＳ１０２において抽出した内部変数のうち未処理のいずれかの内部変数を処理対象に選択し（ステップＳ１１８）、ステップＳ１０６以下の処理を繰り返す。

【０１４８】

これに対して、ステップＳ１０２において抽出した内部変数のすべてについての処理が完了している場合（ステップＳ１１６においてＹＥＳの場合）には、ＣＰＵ２０１は、生成されたオブジェクトファイルを連結して実行可能プログラムを生成する（ステップＳ１２０）。そして、処理は終了する。

【０１４９】

＜Ｉ．変形例＞

上述の実施の形態においては、ネットワーク変数を遣り取りする相手先のＰＬＣの種別をＩＰアドレスなどによって特定する方法について例示したが、これに限られず、任意の方法を採用することができる。

【０１５０】

例えば、相手先のＰＬＣの機種などが既知である場合には、ユーザがネットワーク公開属性を設定する際に、そのＰＬＣの種別を直接的に設定してもよい。このような構成を採用した場合には、図１１に示すネットワーク接続情報２７６が不要になる。

【０１５１】

さらに、相手先のＰＬＣの機種における対応するデータ構造が既知である場合には、ユーザがネットワーク公開属性を設定する際に、そのＰＬＣにおけるデータ構造を直接的に設定してもよい。このような構成を採用した場合には、図１１に示すデータ構造情報２７８が不要になる。

【０１５２】

また、上述の実施の形態においては、メインメモリ１０４に格納されるそれぞれの変数のデータ構造を適合化する構成について例示したが、ネットワークコントローラ１４０のバッファメモリ１４６に格納されるそれぞれの変数のデータ構造を適合化してもよい。この場合には、メインメモリ１０４に格納される変数は、ＰＬＣ１の本来のデータ構造で格納されており、他のＰＬＣから受信されてバッファメモリ１４６に一次的に格納されたデータが対応する変数値としてメインメモリ１０４に転送される際にデータ構造の変換処理（適合化処理）が行われてもよい。また、他のＰＬＣへ送信されるためにメインメモリ１０４からバッファメモリ１４６へデータが転送される際にデータ構造の変換処理（適合化処理）が行われてもよい。

【０１５３】

＜Ｊ．利点＞

本実施の形態に係るサポート装置２００は、ＰＬＣ間で変数を指定してデータが遣り取りされる場合において、相手先のＰＬＣが扱うデータ構造が一致してないくとも、ネットワークを介してデータをより容易にデータを遣り取りできる。

【０１５４】

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。

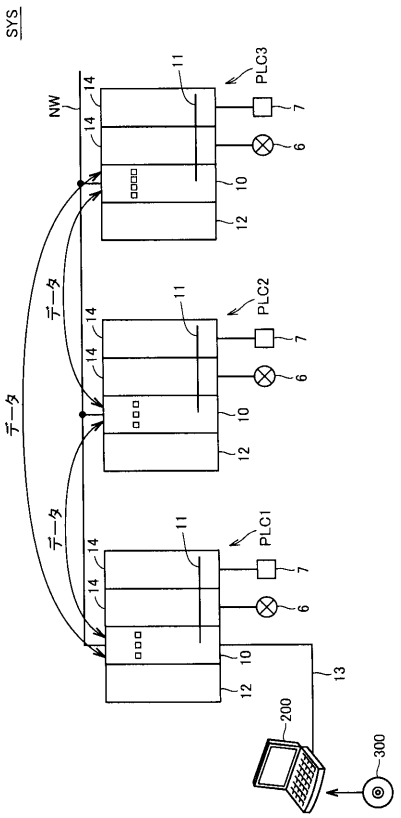
【符号の説明】

【０１５５】

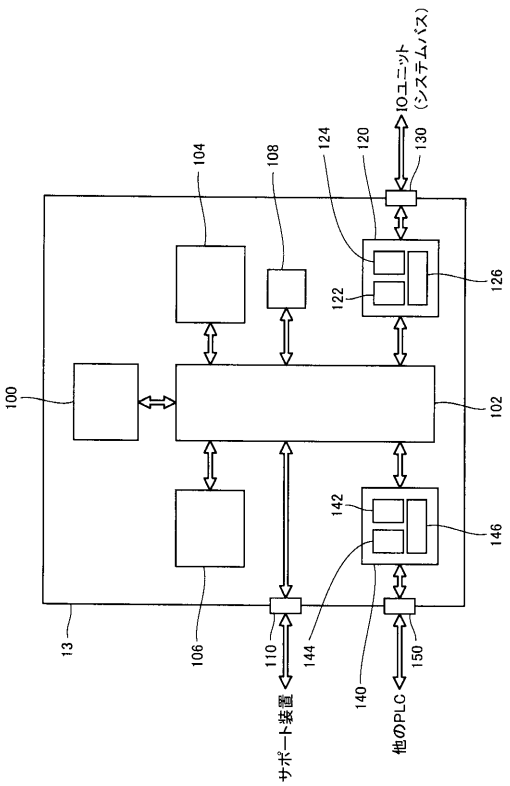
６　検出センサー、７　リレー、１０　処理ユニット、１１　システムバス、１２　電源ユニット、１３　接続ケーブル、１４　ユニット、１００　プロセッサ、１０２　チップセット、１０４　メインメモリ、１０６　不揮発性メモリ、１０８　システムタイマ、１１０　ＵＳＢコネクタ、１２０　システムバスコントローラ、１２２　ＤＭＡ制御回路、１２４　システムバス制御回路、１２６，１４６　バッファメモリ、１３０　システムバスコネクタ、１４０　ネットワークコントローラ、１４４　ネットワーク制御回路、１６０　スケジューラ、１７０　ユーザ定義アプリケーション、１７２　入力処理モジュール、１７４　出力処理モジュール、１７６　通信処理モジュール、１７８　システムモジュール、１８０　命令実行モジュール、１８４　メモリマネジャー、１８６　データ構造変換モジュール、１９０　リアルタイムＯＳ、２００　サポート装置、２０１　ＣＰＵ、２０２　ＲＯＭ、２０３　ＲＡＭ、２０４　ハードディスク（ＨＤＤ）、２０５　キーボード、２０６　マウス、２０７　モニタ、２０８　ＣＤ－ＲＯＭドライブ、２４０　ＯＳ、２５０　プログラミングアプリケーション、２５２，２８２　エディタ、２５４　コンパイラ、２５６　デバッガ、２５８　ＧＵＩモジュール、２６０　シミュレータ、２７０，２９０　データ格納部、２７４　ソースプログラム、２７６　ネットワーク接続情報、

２７８　データ構造情報、２８０　ネットワーク設定アプリケーション、２８４　コンフィグレータ、２９６　コネクション設定、３００　ＣＤ－ＲＯＭ、２５４２　パーサ、２５４４　オブジェクトファイル生成部、２５４６　データ構造特定部、２５４８　リンカ、２７２１　内部変数定義、２７２２　ネットワーク公開属性、ＮＷ　ネットワーク、ＳＹＳ　システム。

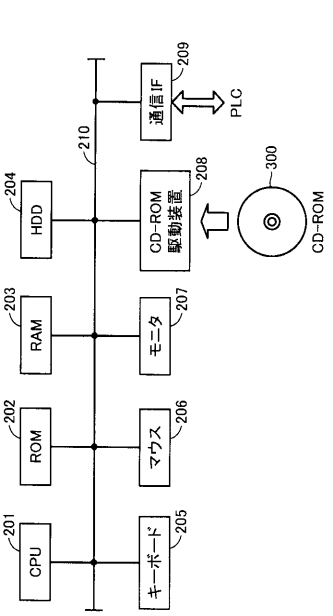
【図１】



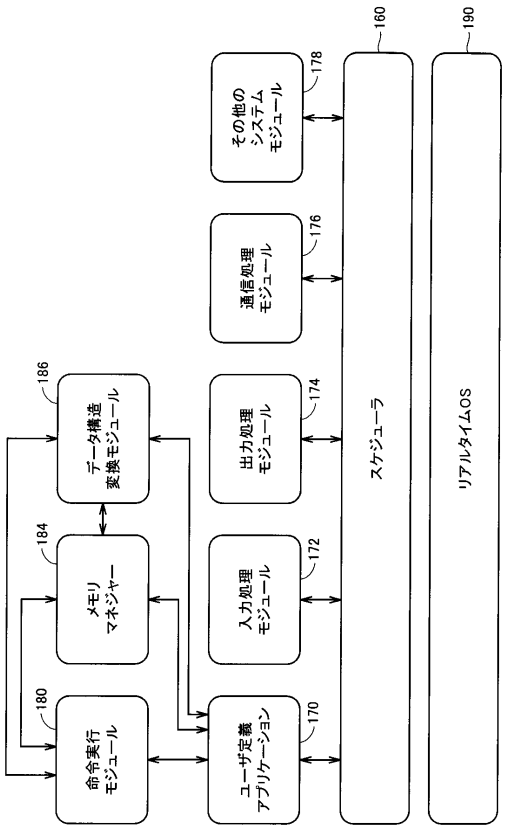
【図２】



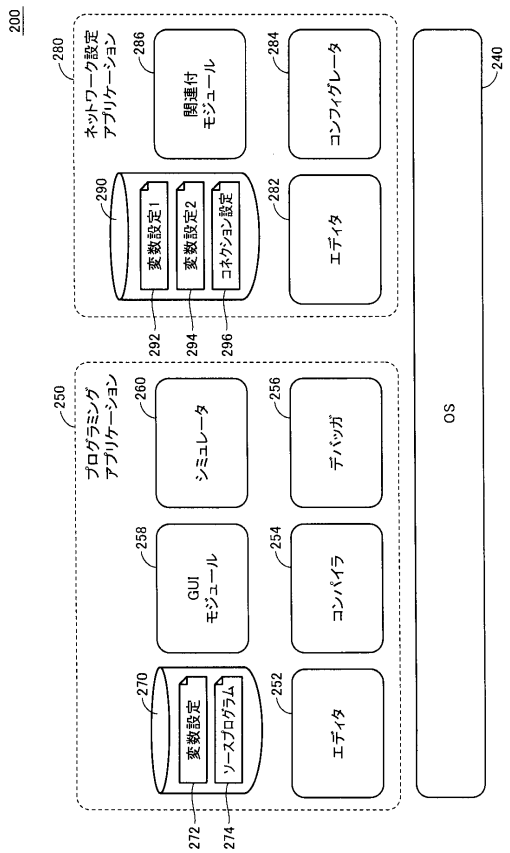
【図３】


200
CPU
201
ROM
202
RAM
203
HDD
204
キーボード
205
マウス
206
モニタ
207
CD-ROM 駆動装置
208
通信IF
209
PLC
210
CD-ROM
300

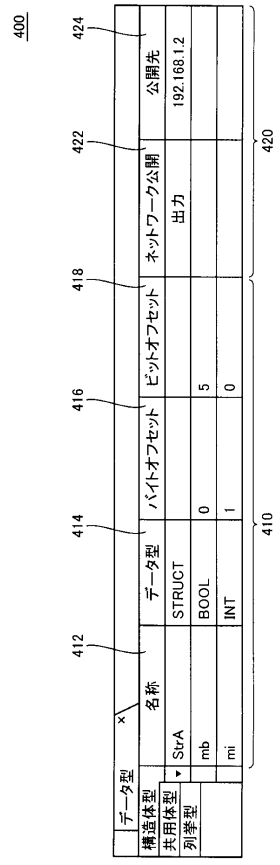
【図４】


10
命令実行 モジュール
180
データ構造 変換モジュール
186
メモリ マネジャー
184
ユーザ定義 アプリケーション
170
入力処理 モジュール
172
出力処理 モジュール
174
通信処理 モジュール
176
その他の システム モジュール
178
スケジューラ
160
リアルタイムOS
190

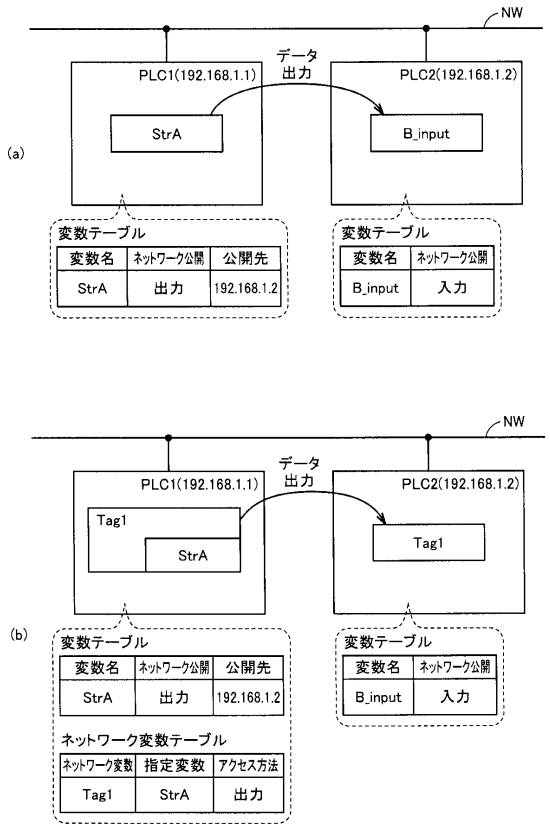
【図５】


200
ネットワーク設定 アプリケーション
280
関連付 モジュール
286
コンフィグレータ
284
変数設定1
変数設定2
コネクション設定
290
エディタ
282
292
294
296
プログラミング アプリケーション
250
シミュレータ
260
デバッガ
256
GUI モジュール
258
コンパイラ
254
変数設定
ソースプログラム
270
エディタ
252
272
274
OS
240

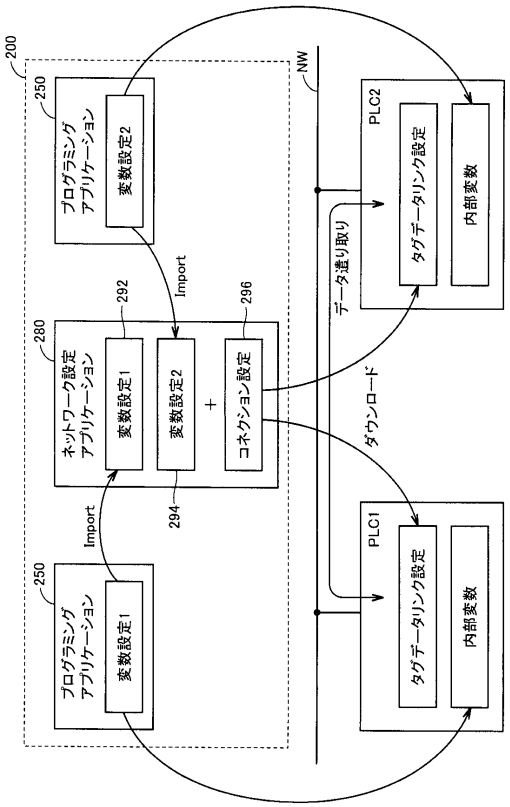
【図６】


400
データ型
構造体型
共用体型
列挙型
名称
データ型
バイトオフセット
ビットオフセット
ネットワーク公開
出力
公開先
StrA
STRUCT
192.168.1.2
mb
BOOL
0
5
mi
INT
1
0
412
414
416
418
420
422
424
410

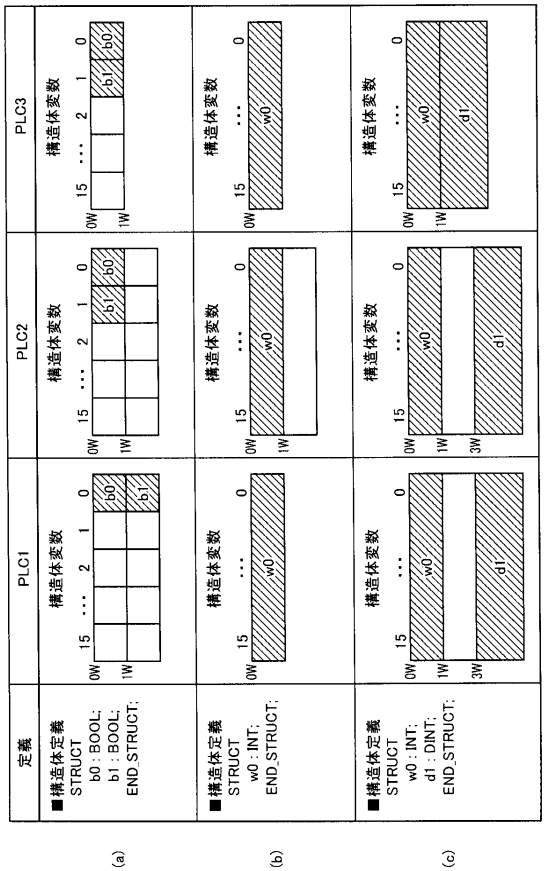
【図７】


NW
データ出力
PLC1(192.168.1.1)
PLC2(192.168.1.2)
StrA
B_input
(a)
変数テーブル
変数名 | ネットワーク公開 | 公開先
StrA | 出力 | 192.168.1.2
変数テーブル
変数名 | ネットワーク公開
B_input | 入力
NW
データ出力
PLC1(192.168.1.1)
PLC2(192.168.1.2)
Tag1
StrA
Tag1
(b)
変数テーブル
変数名 | ネットワーク公開 | 公開先
StrA | 出力 | 192.168.1.2
変数テーブル
変数名 | ネットワーク公開
B_input | 入力
ネットワーク変数テーブル
ネットワーク変数 | 指定変数 | アクセス方法
Tag1 | StrA | 出力

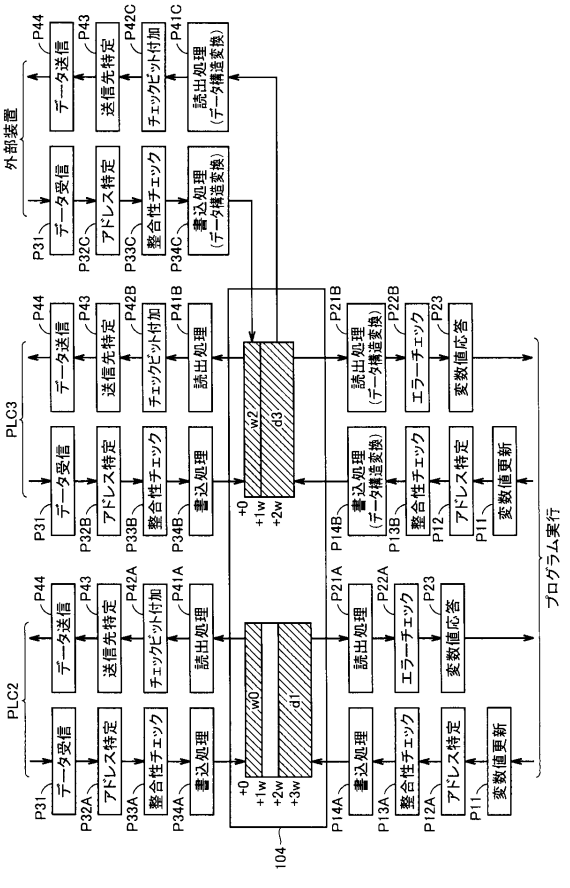
【図８】


200
NW
250
プログラミングアプリケーション
変数設定2
Import
280
ネットワーク設定アプリケーション
変数設定1
292
変数設定2
+
コネクション設定
296
294
Import
250
プログラミングアプリケーション
変数設定1
データ遣り取り
ダウンロード
PLC2
タグデータリンク設定
内部変数
PLC1
タグデータリンク設定
内部変数

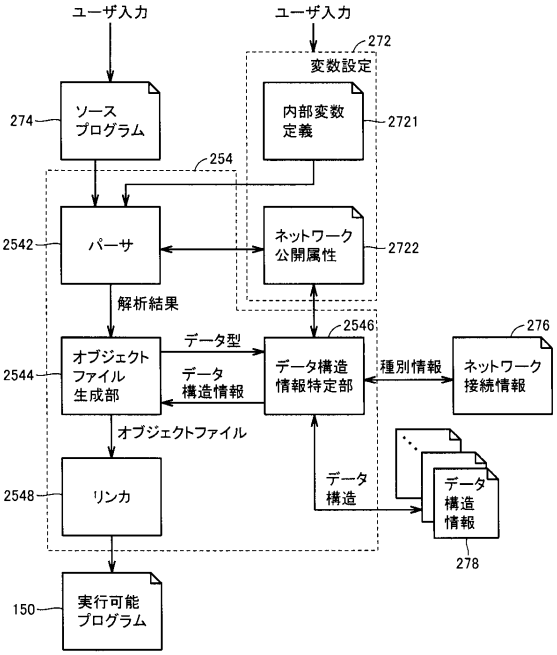
【図９】


定義
PLC1
PLC2
PLC3
■構造体定義
STRUCT
b0 : BOOL;
b1 : BOOL;
END_STRUCT;
構造体変数
■構造体定義
STRUCT
w0 : INT;
END_STRUCT;
構造体変数
■構造体定義
STRUCT
w0 : INT;
d1 : DINT;
END_STRUCT;
構造体変数
(a)
(b)
(c)

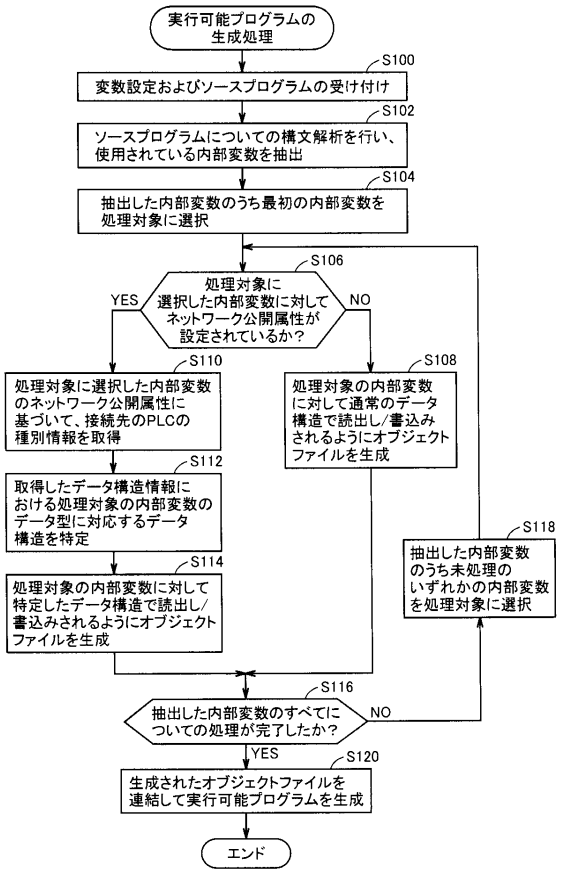
【図１０】


外部装置
PLC3
PLC2
データ受信
アドレス特定
整合性チェック
書込処理
読出処理
送信先特定
チェックビット付加
データ送信
書込処理(データ構造変換)
読出処理(データ構造変換)
エラーチェック
変数値応答
変数値更新
プログラム実行
104

【図１１】


ユーザ入力
ユーザ入力
272
変数設定
274
ソース
プログラム
内部変数
定義
2721
254
2542
パーサ
ネットワーク
公開属性
2722
解析結果
2546
276
データ型
2544
オブジェクト
ファイル
生成部
データ
構造情報
データ構造
情報特定部
種別情報
ネットワーク
接続情報
オブジェクトファイル
データ
構造
情報
2548
リンカ
278
150
実行可能
プログラム

【図１２】


実行可能プログラムの
生成処理
S100
変数設定およびソースプログラムの受け付け
S102
ソースプログラムについての構文解析を行い、
使用されている内部変数を抽出
S104
抽出した内部変数のうち最初の内部変数を
処理対象に選択
S106
処理対象に
選択した内部変数に対して
ネットワーク公開属性が
設定されているか？
YES
NO
S110
処理対象に選択した内部変数
のネットワーク公開属性に
基づいて、接続先のPLCの
種別情報を取得
S108
処理対象の内部変数
に対して通常のデータ
構造で読出し/書込み
されるようにオブジェクト
ファイルを生成
S112
取得したデータ構造情報に
おける処理対象の内部変数の
データ型に対応するデータ
構造を特定
S114
処理対象の内部変数に対して
特定したデータ構造で読出し/
書込みされるようにオブジェクト
ファイルを生成
S118
抽出した内部変数
のうち未処理の
いずれかの内部変数
を処理対象に選択
S116
抽出した内部変数のすべてに
ついての処理が完了したか？
NO
YES
S120
生成されたオブジェクトファイルを
連結して実行可能プログラムを生成
エンド

フロントページの続き

(72)発明者　三浦　悟
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　牧　初

(56)参考文献　特開２０００－１３８７２５（ＪＰ，Ａ）
特開平４－３２９０５（ＪＰ，Ａ）

(58)調査した分野(Int.Cl.，ＤＢ名)
Ｇ０５Ｂ　１９／０４－１９／０５