

(19) 日本国特許庁(JP)

(12) 公開特許公報(A)

(11) 特許出願公開番号

特開2005-293312
(P2005-293312A)

(43) 公開日 平成17年10月20日(2005.10.20)

(51) Int. Cl.⁷
G05B 19/05

F I
G O 5 B 19/05

テーマコード (参考)
5 H 2 2 O

審査請求 未請求 請求項の数 8 O L (全 12 頁)

(21) 出願番号 特願2004-108357 (P2004-108357)
(22) 出願日 平成16年3月31日 (2004.3.31)

(特許庁注：以下のものは登録商標)

1. E T H E R N E T

(71) 出願人 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 100092598
弁理士 松井 伸一
(72) 発明者 本杉 匡史
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 出来 仁太郎
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
Fターム(参考) 5H220 AA01 AA04 AA08 AA09 BB07
CC07 CX05 HH01 HH03 HH08
JJ12 JJ26 JJ34 JJ53 KK03

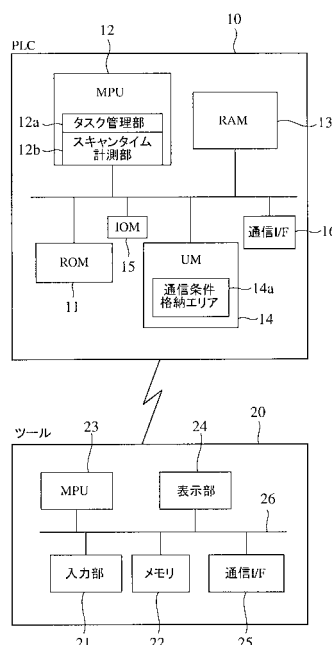
(54) 【発明の名称】 コントローラおよびツール

(57) 【要約】

【課題】 コントローラにおける通信処理の通信性能をユーザが任意に設定できるようにしたコントローラを提供すること

【解決手段】 ユーザプログラムの命令実行と、通信処理を含む1サイクルの処理を繰り返し実行するPLC(CPUユニット10)は、通信条件記憶エリア14aに、通信処理の通信性能を規定する通信条件(パケット処理時間等)を記憶し、MPU12は、その通信条件記憶エリアに格納された通信条件に従い、通信処理を実行する。ユーザがツールを用いて通信条件を更新することにより、そのときの使用状況にあった通信処理と、スキャンタイムのバランスを取ることができる。

【選択図】 図4



【特許請求の範囲】**【請求項 1】**

ユーザプログラムの命令実行と、通信処理を含む 1 サイクルの処理を繰り返し実行するコントローラであって、

前記通信処理の通信性能を規定する通信条件を記憶する通信条件記憶手段と、

その通信条件記憶手段に格納された通信性能に従い、前記通信処理を実行する手段とを備え、

前記通信条件記憶手段に格納された前記通信性能は、外部からの指令に基づき設定可能に構成したことを特徴とするコントローラ。

【請求項 2】

前記通信性能は、1 サイクル当たりのパケットの処理時間であることを特徴とする請求項 1 に記載のコントローラ。

【請求項 3】

前記通信性能は、1 サイクル当たりのパケットの処理量であることを特徴とする請求項 1 に記載のコントローラ。

【請求項 4】

前記通信性能は、前記 1 サイクルを行う際に要するスキャンタイムに対する割合であることを特徴とする請求項 1 に記載のコントローラ。

【請求項 5】

前記 1 サイクルを行う際に要するスキャンタイムを測定する測定手段を備え、

その測定手段により求めたスキャンタイムの実測値を外部へ出力可能としたことを特徴とする請求項 1 から 4 のいずれか 1 項に記載のコントローラ。

【請求項 6】

ユーザプログラムの命令実行と、通信処理を含む 1 サイクルの処理を繰り返し実行するコントローラ用のツールであって、

ユーザからの入力に伴い前記通信処理の通信性能を取得する取得手段と、

その取得した通信性能を前記コントローラに設定する設定手段を備えたことを特徴とするツール。

【請求項 7】

ユーザからの通信性能の入力は、低・高などの相対的な情報とし、

その相対的な情報を具体的な数値情報に変換して前記コントローラに設定する機能を備えたことを特徴する請求項 6 に記載のツール。

【請求項 8】

前記設定手段が前記コントローラに対して設定する通信性能に基づき前記コントローラが動作した場合のスキャンタイムを取得する手段と、

その取得したスキャンタイムを出力手段に出力する機能を備えたことを特徴とする請求項 6 または 7 に記載のツール。

【発明の詳細な説明】**【技術分野】****【0001】**

この発明は、ユーザプログラムの命令実行と、通信処理を含む 1 サイクルの処理を繰り返し実行するコントローラと、そのコントローラ用のツールに関するものである。

【背景技術】**【0002】**

生産工場（製造現場）に設置されるファクトリーオートメーション（FA）の制御装置として、プログラマブルコントローラ（PLC）が用いられている。この PLC には、直接あるいはネットワークを介してスイッチ、センサ等の入力装置や、モータ、アクチュエータ等の出力装置が接続される。PLC は、入力装置から取得した入力データに基づき予め登録されたユーザプログラム記述言語（例えばラダー言語）で組まれたユーザプログラムを演算実行し、得られた出力データに基づき出力装置の動作を制御する。

10

20

30

40

50

【0003】

また、PLCは、ネットワークを介して他のPLCや、上位のコンピュータ（監視装置、管理装置）や、プログラマブル表示器等と接続され、それらの外部の装置と通信処理を行う機能を有する。なお、通信処理は、上記の外部の装置に限らず、入力装置や出力装置との間でも行われることがある。

【0004】

上記のユーザプログラムの演算実行と、通信処理の関係を示すと、図1のようになる。すなわち、PLCは、「ラダー命令実行処理（ユーザプログラムの演算実行を行う）」「通信処理」「その他の処理」を1サイクルとし、サイクリックに繰り返し実行する。そして、1サイクルを実行するのに要する時間をスキャンタイムと称する。そして、通信処理は、1サイクルにつき、1パケットを送るようにしている。これにより、スキャンタイムの変動を抑えるようにしている。

10

【発明の開示】

【発明が解決しようとする課題】

【0005】

しかしながら、上述した従来の1サイクル当たり1パケットについて通信処理をするものでは、以下に示す問題を有する。すなわち、従来一般的に用いられていた伝送速度の遅いシリアル通信ではさほど問題はないが、近年のPLCでは、イーサネット（登録商標）などの伝送速度の速いネットワークにも接続可能となったため、一度に多数の通信（パケット）を受信し、それに対するレスポンスを返すというように通信処理をすべき対象が集中して発生することがある。

20

【0006】

このような場合でも、1サイクル当たり1パケットしか通信処理できないため、ほぼ同時期にN個分の通信処理すべきパケットが存在した場合には、最大でNサイクル分だけ遅れて処理されることになり、迅速な通信処理ができなくなる。さらに、例えば図2に示すように、スキャンタイムが長くなると（図1に示すものに比べて）、上記の問題はより顕著に表れる。

【0007】

一方、係る問題を解決するため、1秒当たりのパケット処理量（PPS：Packet Per sec）を定義するものもある。この場合、通信処理対象のパケットが存在する場合には、定義された間隔で順次通信処理をする。従って、図3に示すように、1サイクル中にラダー命令の実行を中断して複数回通信処理が行なわれることがある。

30

【0008】

1サイクル中に発生する通信処理の回数は、そのときに溜まっている通信処理対象のパケットの数により変動し、0回の場合もあれば、複数回、多数回の場合もある。そして、スキャンタイムは、通信処理の発生回数により変化するため、一義的に定義できず、スキャンタイム n のある程度の時間に保ちたいというユーザには適用できない。

【0009】

さらに、上記の1サイクル当たりの通信処理を1パケットで固定する方式（図1，図2参照）と、1秒当たりのパケット処理量を固定する方式（図3参照）のいずれの場合も、システム設定であり、ユーザが簡単に変更することができなかった。

40

この発明は、コントローラにおける通信処理の通信性能をユーザが任意に設定できるようにしたコントローラおよびツールを提供することを目的とする。

【課題を解決するための手段】

【0010】

この発明のコントローラは、ユーザプログラムの命令実行と、通信処理を含む1サイクルの処理を繰り返し実行するコントローラであって、前記通信処理の通信性能を規定する通信条件を記憶する通信条件記憶手段と、その通信条件記憶手段に格納された通信性能に従い、前記通信処理を実行する手段とを備え、前記通信条件記憶手段に格納された前記通信性能は、外部からの指令に基づき設定可能に構成した。

50

【0011】

ここで、コントローラとは、プログラマブルコントローラその他の制御装置はもちろんのこと、CPUユニットのような構成要素も含む。

【0012】

前記通信性能は、1サイクル当たりに通信処理できる性能であり、この通信性能が高いほど、各パケット等の通信処理対象を遅滞なく伝送することができるようになるが、1サイクルに要するスキャンタイムが長くなり、通信性能が低いほどスキャンタイムを短くすることができるが通信処理対象が集中した場合には、実際に伝送するまでに大きな遅れが生じる。本発明では、外部からの指令によりコントローラが記憶保持する通信性能を設定できるため、上記通信性能とスキャンタイムのバランスを考慮し、各ユーザがそれぞれ適切な通信性能を設定することにより、自己の使用条件に応じた適切な動作状態となる。

10

【0013】

そして、通信性能の規定・設定は各種行えるが、例えば、1サイクル当たりのパケットの処理時間としたり、1サイクル当たりのパケットの処理量としたり、1サイクルを行う際に要するスキャンタイムに対する割合などにより規定できる。

【0014】

前記1サイクルを行う際に要するスキャンタイムを測定する測定手段を備え、その測定手段により求めたスキャンタイムの実測値を外部へ出力可能とするとよい。この外部への出力は、スキャンタイムを測定したことなどを契機とし、自発的に外部（例えばツール）に送信したり、外部からのコマンド（指令）に対するレスポンスとして実測値を出力するようにしてもよく、また、上記とは逆に外部からアクセス可能なメモリ領域に格納しておき、外部からアクセスして取得できるようにしてもよい。スキャンタイムの実測値がわかると、設定した通信性能が適切であったか否かの評価が行え、仮に不適切であった場合には、通信性能を更新する際の参考となる。

20

【0015】

本発明に係るツールは、ユーザプログラムの命令実行と、通信処理を含む1サイクルの処理を繰り返し実行するコントローラ用のツールであって、ユーザからの入力に伴い前記通信処理の通信性能を取得する取得手段と、その取得した通信性能を前記コントローラに設定する設定手段を備えて構成した。

【0016】

取得手段は、実施の形態では、例えば図7や図8などの設定画面を表示したり、入力を受け取る機能であり、S11からS15などを実行する機能部分に対応する。また、設定手段は、実施の形態では、S16や、図10を実行する機能部分に対応する。

30

【0017】

ユーザからの通信性能の入力は、低・高などの相対的な情報とし、その相対的な情報を具体的な数値情報に変換して前記コントローラに設定する機能を備えるとよい。このようにすると、ユーザ感覚的に通信性能を設定することができる。

【0018】

前記設定手段が前記コントローラに対して設定する通信性能に基づき前記コントローラが動作した場合のスキャンタイムを取得する手段と、その取得したスキャンタイムを出力手段に出力する機能を備えるとよい。取得する手段は、設定したコントローラを実際に動作させ、その場合の実測値のスキャンタイムを取得するようにしてもよい、理論値などにより演算して推定してもよい。

40

【発明の効果】

【0019】

この発明では、コントローラにおける通信処理の通信性能をユーザが任意に設定できるようになる。

【発明を実施するための最良の形態】

【0020】

図4は、本発明の好適な一実施の形態を示している。PLCを構成するCPUユニット

50

10と、ツール20とがネットワークを介して接続されている。図示の例では、CPUユニット10とツール20を直接接続しているが、PLCが接続される他のネットワークを介して接続することもできる。つまり、PLCは、このCPUユニット10の他に、電源ユニットや、IOユニットや、通信ユニットなど、各種の機能を実行するユニットを適宜連結して構成される。係る場合に、通信ユニットが接続されるネットワークにツール20を接続し、ツール20は、通信ユニット経由でCPUユニットにアクセスするようにしてもよい。

【0021】

また、CPUユニット10の通信相手は、本実施の形態ではツール20を例に挙げて説明しているが、他のPLCのCPUユニットに対する通信等も適用できる。これにより、分散制御等が行える。このように複数のPLCが存在する場合、ツール20は、ネットワークにつながる複数のPLCのCPUユニットそれぞれに対して、通信処理の設定ができるようになっている。

10

【0022】

なお、PLCは、上記のように複数のユニットを連結するのに限ることはなく、必要な全ての機能を1つの筐体内に実装した一体型のタイプもあり、係るタイプにも本発明は適用できる。

【0023】

CPUユニット10は、PLCの構成要素の一つである。そして、CPUユニット10の本来のサイクリック処理動作をするためのシステムプログラムは、システムROM11に格納されている。そして、MPU12は、ワークメモリとしてのシステムRAM13を適宜使用しながら、システムROM11のシステムプログラムに従った所定の処理を実行する。また、PLCの運転時において、ユーザがプログラミングツールを用いて予め作成したユーザプログラムは、ユーザメモリ14に格納されている。さらに、ユーザプログラム実行の論理演算に用いるI/Oデータやパラメータは、IOメモリ15に格納されている。そしてCPUユニット10は、ユーザプログラム実行処理の前のINリフレッシュ処理においてINデータを、PLCシステムバスを介して接続されたI/Oユニット(図示省略)から取り込む。そしてCPUユニット10における論理演算処理ではそのINデータに基づいて論理演算し、OUTリフレッシュ処理にてその論理演算結果をOUTデータとしてI/Oユニットに対して出力する。

20

30

【0024】

CPUユニット10におけるユーザプログラム実行処理は、MPU12で行われる。MPU12は、ユーザプログラム実行処理に入ると、ユーザメモリ14に格納されたユーザプログラムを順次呼び出して実行する。

【0025】

通信インタフェース16は、ツール20との間でデータの送受を行うものである。本実施の形態における通信インタフェース16は、例えばイーサネット(登録商標)などの、高速なネットワークに対応している。この通信インタフェース16を介してツール20は、CPUユニット10のI/Oデータをモニタしたり、ユーザメモリ14に格納されているユーザプログラムをアップロードしたり、編集したユーザプログラムをユーザメモリ14に対してダウンロードしたり、各種の設定を行ったりする。このツール20との間の通信処理は、例えば、CPUユニット10におけるサイクリック処理の周辺処理にて行う。

40

【0026】

ここで、本実施の形態では、ユーザプログラムは、ラダー言語により構築されており、上記の論理演算(論理演算処理)は、ラダー命令実行と等価である。そして、MPU12は、図5に示すように、従来と同様にラダー命令実行と、通信処理と、その他の処理(周辺処理)を1サイクルとして、サイクリックに繰り返し実行する。本実施の形態における通信処理は、CPUユニット10に内蔵した通信インタフェース16を介して行なう。また、その他の処理は、PLCとパソコンツールとの間で、イーサネット(登録商標)経由

50

でデータをやり取りする処理があり、例えば、上位通信プロトコル処理や、データアクセス処理（型式読み出し、状態読み出し、変数モニタなど）等がある。具体的には、ツールがPLCのメモリアreaを読み出すものである。読み出すデータとしては、スキャンタイム、エラー回数、実行回数、実行合計時間などがある。なお、制御に使うI/Oデータの通信は、各PLCにコントロール通信ユニットを備え、それでコントローラリンク通信をすることで行われる。

【0027】

ラダー命令実行と、通信処理と、その他の処理は、タスク管理部12aがその実行を管理する。すなわち、ラダー命令実行を行う機能と、通信処理を行う機能と、その他の処理を行う機能は、MPU12内に実装され、それぞれタスクとして実行するようになってい 10
る。そこで、タスク管理部12aは、まずラダー命令実行を行うタスクを稼働させ、ユーザメモリ14に格納されたユーザプログラムを先頭から最終行まで命令を逐次実行する。最終行まで実行したならば、次に、通信処理を行うタスクを稼働させる。所望の通信処理が実行されたならば、その他の処理を行うタスクを稼働させる。なお、その他の処理は、上限値（スキャンタイムに対する割合）が予め設定されており、その上限値に達すると処理が終了する。係る処理（実行タスクの切り替え）を逐次行うことにより、図5に示すような1サイクルの処理を繰り返し行うことになる。

【0028】

なお、図には示していないが、ラダー命令を実行するのに伴い、INリフレッシュやOUTリフレッシュ処理を行なう。一例としては、ラダー実行をした後で、INリフレッシュ・OUTリフレッシュを行い、その後通信処理に移行するようにできる。よって、ラダー命令実行の中にINリフレッシュ・OUTリフレッシュ処理が含まれているととらえても良い。 20

【0029】

ここで、本発明では、通信処理の実行条件（通信条件）をユーザの設定により変更できるようにしている。具体的には、本実施の形態では、1スキャンタイム値のデータの処理時間（ t_1 [μ sec]）を任意に設定できるようにしている。具体的な t_1 の値は、ツール20を用いて書き込み可能なユーザメモリ14中の所定記憶エリアである通信条件格納エリア14aに登録する。そして、例えばタスク管理部12aが、その通信条件格納エリア14aにアクセスして、登録されている処理時間 t_1 を取得し、通信処理の実行タ 30
スクを稼働後、 t_1 [μ sec]経過したならば通信処理を実行するタスクを停止するとともに、その他の処理を実行するタスクを稼働するように制御する。また、通信処理を実行するタスク自体が処理時間 t_1 を取得し、設定された処理時間だけ通信処理を実行したらその回の処理を終了するようにしてもよい。なお、通信条件格納エリア14aは、本実施の形態では、ユーザメモリ14にその領域を確保したが、本発明はこれに限ることはなく、ようはユーザがアクセス可能な記憶エリアであればよい。

【0030】

上述したごとく、処理時間 t_1 を格納する通信条件格納エリア14aは、ユーザがアクセス可能なユーザメモリ14に領域を確保しているため、ツール20を用いてユーザが自由に設定できる。従って、例えばスキャンタイム n_1 が長くなってもよいが、通信が遅れるのを避けたいユーザは、処理時間 t_1 を長く設定し、逆に、通信は送られてもよいが、スキャンタイム n_1 を短くしたいユーザは、 t_1 を短く設定することにより、各ユーザの仕様にあった通信条件で動作するPLCを簡単に構築することができる。 40

【0031】

また、通信条件としては、上述したように、パケット処理時間を可変とするものに限ることはなく、例えば図6に示すように、通信処理対象のパケット数 p_1 を変更するようにしてもよい。つまり、1スキャンタイムあたりのパケット処理量を変更するもので、従来のように $p_1 = 1$ [パケット] としてスキャンタイム n_1 を早くすることを優先したり、逆に p_1 を複数に設定し、通信処理を優先するようにできる。さらに通信処理を優先する場合でも、数値を適宜に設定することによりその程度を所望の値に設定することにより、 50

ユーザが望む仕様に設定することができる。

【0032】

さらにまた、時間 t_1 やパケット数 p_1 のように絶対値として設定するのではなく、スキャンタイム n_1 に対する割合（例えば 10%）のように相対的な値により通信条件を設定することもできる。

【0033】

次に、上記の通信条件をユーザが容易に設定するためのツール 20 について説明する。図 4 に示したように、ツール 20 は、入力部 21 と、メモリ 22 と、MPU 23 と、表示部 24 ならびに通信インタフェース 25 が内部バス 26 を介して接続された構成をとる。入力部 21 は、各種の条件を入力するキーボード、ポインティングデバイス等から構成される。また、表示部 24 は、ディスプレイからなり、本発明との関係でいうと、そのディスプレイの表示画面に条件の入力画面を表示し、ユーザに条件の入力を促す機能を有する。

10

【0034】

通信インタフェース 25 は、PLC (CPU ユニット 10) と通信をするためのもので、CPU ユニット 10 に直結する場合には、例えば RS 232C 対応のインタフェースとなり、PLC の通信ユニットなどに接続する場合には、その通信ユニットが接続されているネットワーク（イーサネット（登録商標）等）に対応したインタフェースとなる。

【0035】

メモリ 22 は、図では 1 つのブロックとして示しているが、実際には、システムプログラムや、表示画面のテンプレートや、設定条件のリストなどを格納する不揮発性メモリと、MPU 23 の演算実行中にワークメモリとして使用したりする揮発性メモリがある。

20

【0036】

MPU 23 は、例えば、図 7、図 8 に示すような、通信条件の設定画面を表示部 24 に出力表示し、プルダウンメニューに基づき、パケット処理時間を設定する。このとき、具体的な絶対値で指定する方法もあるが、ユーザが具体的な絶対値を指定することは困難である。そこで、「高」「中」「低」などの抽象的な情報を入力することにより、ツール側で具体的な時間や、パケット数などに変更し、その変換した値を PLC にダウンロードするようにした。そして、具体的には、図 9 以降に示すフローチャートを実行するようになる。

30

【0037】

まず、PLC のシステム設定モードにし (S11)、通信設定を選択する (S12)。すなわち、まず、S11 の処理では、ユーザが表示部 24 に表示されたメニュー画面からシステム設定モードを選択したことを認識すると、図 7 に示すような各種の設定を行なうための設定画面を表示する（実際には、フロントページでもある図 7 における「一般の設定画面」を初期画面として表示する）。次いで、ユーザは、この設定画面中の任意の項目を選択し、各種のパラメータ等を設定するが、本発明との関係で言うと、通信条件の設定画面は、「Ethernet」の画面にあるため、ユーザは「Ethernet」の欄をクリックすることとなる。そこで、S12 の処理では、係る「Ethernet」の欄をクリックされたことを検知して、該当する画面を切り替え表示する（図 7 参照）。

40

【0038】

次いで、通信性能（ここでは、処理時間）の入力画面を表示する (S13)。つまり、図 7 に示す画面において、左欄側のグループ中の「高度な設定」がクリックされたことを検知すると、右欄側に図示するパラメータの入力画面を表示する。この入力画面における「通信帯域量」の欄が、通信処理時間を指定する欄である。本例では、プルダウンメニュー方式で「低・中・高」の 3 種類から選択するようにしている。ここで、「低」は 400 μ sec, 「中」は 1.6 msec, 「高」は 3.2 msec 2 対応づけている。もちろん、具体的な数値を入力するようにしても良いが、具体的な数値としてどれくらいが標準なのかをユーザが任期視していないと、適切な数値を入力できなくなるため、「低・中・高」というように感覚的に理解しやすいようにした。

50

【 0 0 3 9 】

ユーザは、プルダウンメニューを利用して、3つのうちのいずれかの通信速度を選択するため、MPU23は、どの速度を選択されたかを取得する(S14)。そして、選択された速度を表示する。つまり、図7に示す状態から、ユーザが通信帯域量(パケット処理時間)を「高」にすると、図8に示すように、通信帯域量の欄を「高」と表示する。実際には、メニューリストの中で選択されたものを表示する。

【 0 0 4 0 】

この状態で、設定画面の下側に用意された「OK」ボタン或いは「適用」ボタンがクリックされたことを検知すると、そのとき選択されている通信帯域量が通信設定内容として確定される(S15)。本実施の形態では、パケット処理時間を変更・設定することを前提としているため、いきなり通信帯域量の入力を行なうようにしたが、例えば、パケット数や、割合など、通信条件を複数種類用意し、その中から任意のものをユーザに選択させるようにした場合には、係る種類の選択画面も用意する。なお、その場合においても、通信帯域量は、「低・中・高」のように、複数の段階から選択させ、それぞれにパケット数や割合を対応づけておくようにすると良い。なおまた、パケット数は、現状が1パケット固定であることから、数値入力を許容しても良い。

10

【 0 0 4 1 】

そして、上述の処理により確定された通信設定の内容(処理時間)をPLC(CPUユニット)へ送り、設定する(S16)。この設定処理は、ツール20とPLC(CPUユニット10)との間で、図10, 図11に示すフローチャートを実行することで行なう。

20

【 0 0 4 2 】

すなわち、まずツール側では、図10に示すように、PLCのモードをモニタ(S21)し、プログラムモードか否かを判断する(S22)。そして、プログラムモードでない(RUNモード)場合には、プログラムモードに切り替える指令を送信する(S23)。これにより、S24の処理を実行する際には、PLC側はプログラムモードに切替が完了しているため、通信設定内容をPLCへ送信する(S24)。このとき送る通信設定内容(通信条件)は、ユーザが指定した「低・中・高」のいずれかに対応する具体的な数値(予め用意した対応テーブルを参照して取得)を送ることになる。

【 0 0 4 3 】

図15に示すように、PLC(CPUユニット20)側では、ツール20側で少なくともS23の処理を行なうことにより、プログラムモードに切り替わる(S31)。その後、ツール20から通信設定の変更指令を受けるのを待つ(S32)。そして、通信設定を受信したならば、受信した内容に設定を更新する(S33)。これにより、ユーザメモリ14の通信条件格納エリア14aに、取得した通信設定の内容、つまり、具体的なパケット処理時間を格納する。

30

【 0 0 4 4 】

CPUユニット10のMPU12は、設定の更新を行なった後、電源リセットをかけて再起動し(S34)、RUNモードに切り替える(S35)。そして、更新されたパケット処理時間に基づき、ラダー命令実行, 通信処理, その他の処理といった一連の処理を実行する(S35)。このとき、スキャンタイム計測部12bが、実際にかかったスキャンタイムn1を実測する(S36)。そして、実測して得られたスキャンタイムn1をツール20に送る。以後、通常のサイクリックな処理を行なう(S37)。

40

【 0 0 4 5 】

なお、スキャンタイム計測部12bは、通常のサイクリックな処理の実行時においても、継続してスキャンタイムn1を計測し、記憶保持する。また、上記のS36で求めた実測値n1は、CPUユニット20側から自発的に送信しても良いが、ツール20からの要求に応じてレスポンスとしてツール20側に渡すようにしても良い。

【 0 0 4 6 】

一方、ツール20は、S24の通信設定内容をPLCに送ったのち、PLC(CPUユニット20)側からスキャンタイムn1が送られてくるのを待つ(S25)。そして、ス

50

キャンタイム n 1 を受信したならば、そのスキャンタイム (サイクルタイム) n 1 を表示部 2 4 に表示し (S 2 6)、ユーザからの許可を待つ。

【 0 0 4 7 】

ユーザは、表示部に表示されたスキャンタイムを見て、許可できるか否かを判断し、入力部 2 1 を操作してその判断結果を入力する。そこで、M P U 2 3 は、ユーザからの許可の指示を受けると (S 2 7 の分岐判断で Y e s)、設定を終了する (S 2 8)。また、不許可の指示を受けると (S 2 7 の分岐判断で N o)、再設定を行なう (S 2 9)。つまり、図 9 のフローチャートを実行する。

【 0 0 4 8 】

なお、上記した実施の形態では、図 1 0 に示すフローチャートのように、S 2 4 の処理を行ない一旦ツール 1 0 から通信設定の内容を P L C に送った後、S 2 5 から S 2 9 の処理ステップのように、P L C からのサイクルタイムの受信を待って再設定の可否を促す機能を設けたが、本発明はこれに限ることはない。すなわち、例えば、S 2 4 処理を実行することでツール側の処理は一旦終了するようにし、人間が P L C の動作状態 (応答精度) をチェックし、必要に応じて再設定を行うこともできる。この場合の再設定は、例えば図 9 に示すフローチャートなどを実行することにより対応できる。

10

【 0 0 4 9 】

なお、上記した実施の形態では、スキャンタイム n 1 を実測し、その結果をツール 2 0 の表示部 2 3 に表示して、その長短 (適否) をユーザに判断させ、スキャンタイム n 1 と通信処理の調整を行なうようにしたが、本発明では、スキャンタイム n 1 を必ずしも表示する必要はない。また、表示する場合でも、実測するのではなく、ユーザプログラムの命令数などに基つきツールが計算によりスキャンタイムの推定値を求め、それを表示するようにしてもよい。

20

【 0 0 5 0 】

さらに、通信処理の設定として、割合を用いた場合には、例えば、ユーザプログラムの命令数などに基ついてラダー命令実行にかかる時間を推測し、その他の処理にかかる時間を固定値とし、通信処理時間を変数として設定した割合になる処理時間を算出し、それを通信設定内容として通知することにより実現できる。また、一旦通信処理時間を 0 にセットして、1 サイクルの処理を実行し、そのとき実測したスキャンタイム n 1 に基つき所望の割合になり通信処理時間を求め、それを通信設定内容として通知するようにしても良い等、各種の対応がとれる。

30

【 図面の簡単な説明 】

【 0 0 5 1 】

【 図 1 】 従来例を示す図である。

【 図 2 】 従来例を示す図である。

【 図 3 】 従来例を示す図である。

【 図 4 】 本発明の一実施の形態を示す図である。

【 図 5 】 本実施の形態の動作を説明する図である。

【 図 6 】 本実施の形態の動作を説明する図である。

【 図 7 】 条件設定画面の一例を示す図である。

40

【 図 8 】 条件設定画面の一例を示す図である。

【 図 9 】 ツールの機能を示すフローチャートである。

【 図 1 0 】 ツールの機能を示すフローチャートである。

【 図 1 1 】 P L C (C P U ユニット) の機能を示すフローチャートである。

【 符号の説明 】

【 0 0 5 2 】

1 0 C P U ユニット

1 1 システム R O M

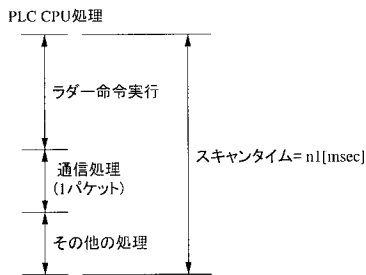
1 2 M P U

1 2 a タスク管理部

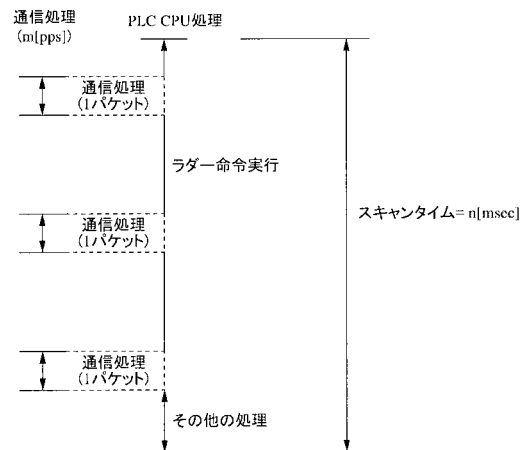
50

- 1 2 b スキャンタイム計測部
- 1 3 システム R A M
- 1 4 ユーザメモリ
- 1 4 a 通信条件格納エリア
- 1 5 I Oメモリ
- 1 6 通信インタフェース
- 2 0 ツール
- 2 1 入力部
- 2 2 メモリ
- 2 3 M P U
- 2 4 表示部
- 2 5 通信インタフェース
- 2 6 バス

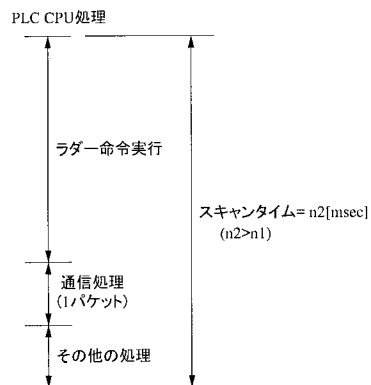
【 図 1 】



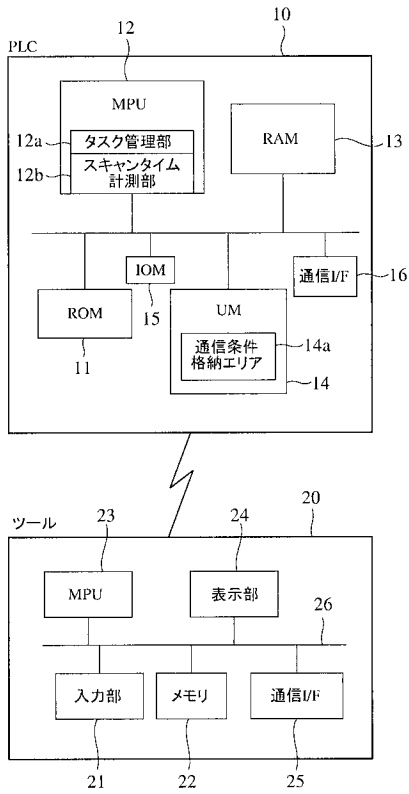
【 図 3 】



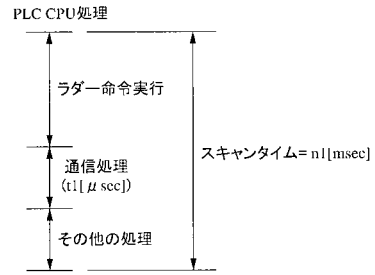
【 図 2 】



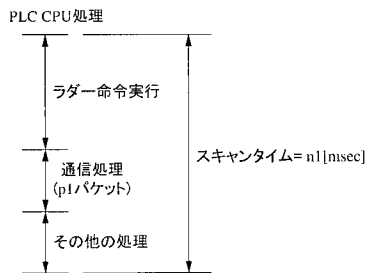
【 図 4 】



【 図 5 】



【 図 6 】



【 図 7 】

パラメータ名	設定
Ethernetを使用する	使用する
通信帯域制御を使用する	使用する
通信帯域量	低

初期設定値: 低

ヘルプ: 通信帯域量を設定します。

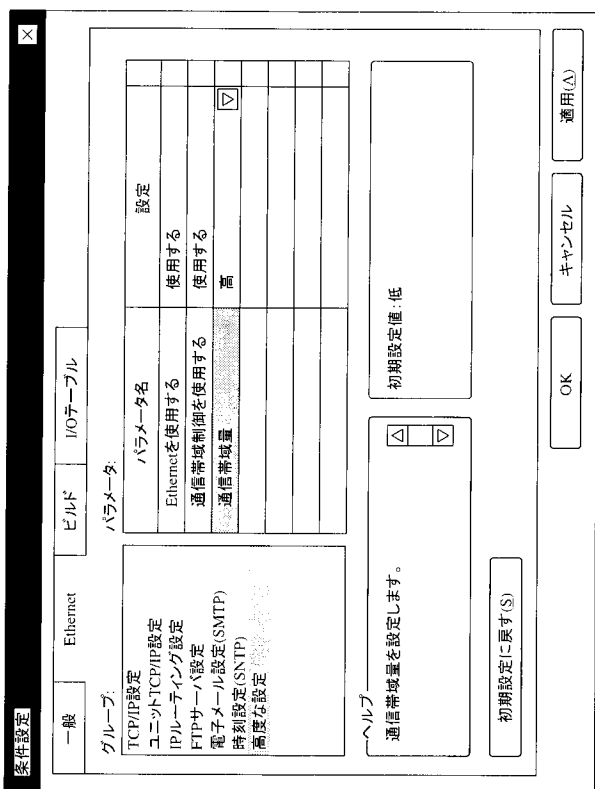
初期設定に戻す(S)

適用(A)

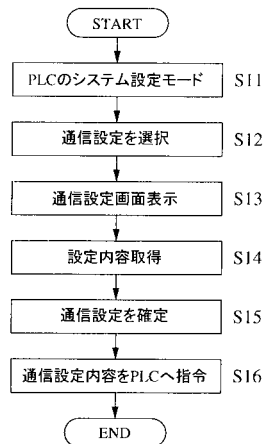
キャンセル

OK

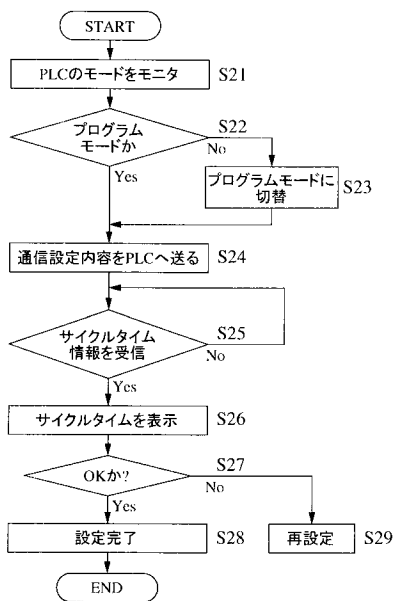
【 図 8 】



【 図 9 】



【 図 10 】



【 図 11 】

