

(19)日本国特許庁(JP)

(12)特許公報(B2)

(11)特許番号  
特許第7608885号  
(P7608885)

(45)発行日 令和7年1月7日(2025.1.7)

(24)登録日 令和6年12月23日(2024.12.23)

(51)国際特許分類 F I  
G 0 5 B 19/05 (2006.01) G 0 5 B 19/05 F

請求項の数 12 (全29頁)

(21)出願番号	特願2021-39408(P2021-39408)	(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
(22)出願日	令和3年3月11日(2021.3.11)	(74)代理人	110001195 弁理士法人深見特許事務所
(65)公開番号	特開2022-139149(P2022-139149 A)	(72)発明者	桐淵 岳 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
(43)公開日	令和4年9月26日(2022.9.26)	(72)発明者	大野 悌 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
審査請求日	令和6年1月16日(2024.1.16)	(72)発明者	山脇 聖 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】 制御システムおよび制御方法

(57)【特許請求の範囲】

【請求項1】

制御システムであって、

第1の制御ユニット用の第1のプロジェクトファイルと、第2の制御ユニット用の第2のプロジェクトファイルと、前記第1および第2のプロジェクトファイルに基づいて生成される統合プロジェクトファイルとを格納する記憶部と、

各プロジェクトファイルに基づいて処理を実行する制御部とを備え、

前記第1のプロジェクトファイルは、前記第1の制御ユニットが、前記第1の制御ユニットに接続されたフィールドネットワーク上の複数のノードの各々と通信するための複数のノード識別子の各々を含み、

前記第2のプロジェクトファイルは、前記第2の制御ユニットが、前記第2の制御ユニットに接続されたフィールドネットワーク上の複数のノードの各々と通信するための複数のノード識別子の各々を含み、

前記統合プロジェクトファイルは、前記第1のプロジェクトファイルおよび前記第2のプロジェクトファイルに含まれる全てのノードの各々に一意に割り当て直した複数のノード識別子の各々を含み、

前記制御部は、前記全てのノードの各々に一意に割り当て直した複数のノード識別子の各々を用いて、前記制御システムに接続されたフィールドネットワーク上の複数のノードの各々と通信可能に構成される、制御システム。

【請求項2】

前記記憶部は、第1の仮想制御ユニットと、第2の仮想制御ユニットと、仮想統合制御ユニットとをさらに格納し、

前記制御部は、

前記第1の仮想制御ユニットに、前記第1のプロジェクトファイルをインストールし、  
前記第2の仮想制御ユニットに、前記第2のプロジェクトファイルをインストールし、  
前記仮想統合制御ユニットに、前記統合プロジェクトファイルをインストールし、

前記仮想統合制御ユニットにおいて、前記制御システムに接続されたフィールドネットワーク上のいずれかのノードから受信した通信データを前記第1の仮想制御ユニットまたは前記第2の仮想制御ユニットのいずれかに転送する、請求項1に記載の制御システム。

【請求項3】

前記記憶部は、前記第1のプロジェクトファイルに含まれる第1のネットワーク情報ファイルおよび前記第2のプロジェクトファイルに含まれる第2のネットワーク情報ファイルと、前記統合プロジェクトファイルに含まれる統合ネットワーク情報ファイルとの変換情報をさらに格納し、

前記第1のネットワーク情報ファイル、前記第2のネットワーク情報ファイルおよび前記統合ネットワーク情報ファイルの各々は、各プロジェクト内で使用されるノード識別子を含み、

前記制御部は、前記変換情報を用いて、前記通信データを前記第1の仮想制御ユニットまたは前記第2の仮想制御ユニットのいずれかに転送するように構成される、請求項2に記載の制御システム。

【請求項4】

前記制御部は、前記第1のネットワーク情報ファイルで定義されている第1の制御周期と、前記第2のネットワーク情報ファイルで定義されている第2の制御周期との公約数である制御周期に基づいて、前記ノードの各々と通信する、請求項3に記載の制御システム。

【請求項5】

複数のプロジェクトファイルを統合する装置をさらに含み、

前記装置は、前記第1のプロジェクトファイルと、前記第2のプロジェクトファイルとに基づいて、前記統合プロジェクトファイルを生成し、

前記制御部は、前記装置から、前記第1のプロジェクトファイルと、前記第2のプロジェクトファイルと、前記統合プロジェクトファイルとを取得する、請求項2～4のいずれかに記載の制御システム。

【請求項6】

前記装置は、前記制御システムの制御周期内における、前記第1の仮想制御ユニットの処理時間の占有率と、前記第2の仮想制御ユニットの処理時間の占有率との合計が100%を超えることに基づいて、エラーを出力する、請求項5に記載の制御システム。

【請求項7】

前記制御部は、前記第1のプロジェクトファイルと、前記第2のプロジェクトファイルとに基づいて、前記統合プロジェクトファイルを生成する、請求項1に記載の制御システム。

【請求項8】

制御ユニットを含む制御システムの制御方法であって、

第1の制御ユニット用の第1のプロジェクトファイルと、第2の制御ユニット用の第2のプロジェクトファイルとに基づいて生成される統合プロジェクトファイルを参照するステップを含み、

前記第1のプロジェクトファイルは、前記第1の制御ユニットが、前記第1の制御ユニットに接続されたフィールドネットワーク上の複数のノードの各々と通信するための複数のノード識別子の各々を含み、

前記第2のプロジェクトファイルは、前記第2の制御ユニットが、前記第2の制御ユニットに接続されたフィールドネットワーク上の複数のノードの各々と通信するための複数のノード識別子の各々を含み、

10

20

30

40

50

前記統合プロジェクトファイルは、前記第 1 のプロジェクトファイルおよび前記第 2 のプロジェクトファイルに含まれる全てのノードの各々に一意に割り当て直した複数のノード識別子の各々を含み、

前記全てのノードの各々に一意に割り当て直した複数のノード識別子の各々を用いて、前記制御システムに接続されたフィールドネットワーク上の複数のノードの各々と通信するステップをさらに含む、制御方法。

【請求項 9】

第 1 の仮想制御ユニットに、前記第 1 のプロジェクトファイルをインストールするステップと、

第 2 の仮想制御ユニットに、前記第 2 のプロジェクトファイルをインストールするステップと、

仮想統合制御ユニットに、前記統合プロジェクトファイルをインストールするステップと、

前記仮想統合制御ユニットにおいて、前記制御システムに接続されたフィールドネットワーク上のいずれかのノードから受信した通信データを前記第 1 の仮想制御ユニットまたは前記第 2 の仮想制御ユニットのいずれかに転送するステップとをさらに含む、請求項 8 に記載の制御方法。

【請求項 10】

前記第 1 のプロジェクトファイルに含まれる第 1 のネットワーク情報ファイルおよび前記第 2 のプロジェクトファイルに含まれる第 2 のネットワーク情報ファイルの各々に含まれるノード識別子と、前記統合プロジェクトファイルに含まれる統合ネットワーク情報ファイルとの変換情報にアクセスするステップをさらに含む、

前記第 1 のネットワーク情報ファイル、前記第 2 のネットワーク情報ファイルおよび前記統合ネットワーク情報ファイルの各々は、各プロジェクト内で使用されるノード識別子を含み、

前記変換情報を用いて、前記通信データを前記第 1 の仮想制御ユニットまたは前記第 2 の仮想制御ユニットのいずれかに転送するステップをさらに含む、請求項 9 に記載の制御方法。

【請求項 11】

前記第 1 のネットワーク情報ファイルで定義されている第 1 の制御周期と、前記第 2 のネットワーク情報ファイルで定義されている第 2 の制御周期との公約数である制御周期に基づいて、前記ノードの各々と通信するステップをさらに含む、請求項 10 に記載の制御方法。

【請求項 12】

前記制御システムの制御周期内における、前記第 1 の仮想制御ユニットの処理時間の占有率と、前記第 2 の仮想制御ユニットの処理時間の占有率との合計が 100%を超えることに基づいて、エラーを出力するステップをさらに含む、請求項 11 に記載の制御方法。

【発明の詳細な説明】

【技術分野】

【0001】

本開示は、制御システムに関し、より特定的には、複数の制御ユニットのプロジェクトファイルの統合に関する。

【背景技術】

【0002】

様々な生産現場において、PLC（プログラマブルコントローラ）等の制御装置を用いたFA（Factory Automation）技術が広く普及している。生産現場では、例えば、複数の制御装置が設置され、各制御装置は、複数のロボットまたは工作機器等を制御し得る。

【0003】

制御装置に関して、例えば、特開 2019-036043 号公報（特許文献 1）は、「実行毎に全体がスキャンされる第 1 のプログラムと、逐次実行される第 2 のプログラムと

10

20

30

40

50

を格納する記憶部と、予め定められた制御周期毎に第1のプログラムを実行して第1の指令値を演算する実行処理部と、第2のプログラムの少なくとも一部を解釈して中間コードを生成するインタプリタと、インタプリタが事前に生成した中間コードに従って制御周期毎に第2の指令値を演算する指令値演算部と、実行処理部により演算された第1の指令値および指令値演算部により演算された第2の指令値を制御周期毎に出力する出力部とを含む」制御装置を開示している（〔要約〕参照）。

【先行技術文献】

【特許文献】

【0004】

【文献】特開2019-036043号公報

10

【発明の概要】

【発明が解決しようとする課題】

【0005】

特許文献1に開示された技術によると、複数の制御ユニット用のプロジェクトファイルを容易に統合することができない。したがって、複数の制御ユニット用のプロジェクトファイルを容易に統合するための技術が必要とされている。

【0006】

本開示は、上記のような背景に鑑みてなされたものであって、ある局面における目的は、複数の制御ユニット用のプロジェクトファイルを容易に統合するための技術を提供することにある。

20

【課題を解決するための手段】

【0007】

本開示の一例に従えば、制御システムが提供される。制御システムは、第1の制御ユニット用の第1のプロジェクトファイルと、第2の制御ユニット用の第2のプロジェクトファイルと、第1および第2のプロジェクトファイルに基づいて生成される統合プロジェクトファイルとを格納する記憶部と、各プロジェクトファイルに基づいて処理を実行する制御部とを備える。第1および第2のプロジェクトファイルの各々は、各プロジェクトファイルごとに独立して複数のノードの各々に割り当てられた複数のノード識別子を含む。統合プロジェクトファイルは、全てのノードの各々に一意に割り当て直したノード識別子を含む。制御部は、全てのノードの各々に一意に割り当て直したノード識別子を用いて、通信するノードのノード識別子を特定する。

30

【0008】

この開示によれば、制御システムは、第1および第2のプロジェクトファイルに含まれる全てのノードの各々に一意なノード識別子を設定することで、各ノードと通信可能になる。

【0009】

上記の開示において、記憶部は、第1の仮想制御ユニットと、第2の仮想制御ユニットと、仮想統合制御ユニットとをさらに格納する。制御部は、第1の仮想制御ユニットに、第1のプロジェクトファイルをインストールし、第2の仮想制御ユニットに、第2のプロジェクトファイルをインストールし、仮想統合制御ユニットに、統合プロジェクトファイルをインストールし、仮想統合制御ユニットおよびノードの各々間の通信データを第1の仮想制御ユニットまたは第2の仮想制御ユニットのいずれかに転送する。

40

【0010】

この開示によれば、制御システムは、各制御ユニットを仮想制御ユニットとして動作させることにより統合し、また、各制御ユニットおよび各ノード間の通信を転送し得る。

【0011】

上記の開示において、記憶部は、第1のプロジェクトファイルに含まれる第1のネットワーク情報ファイルおよび第2のプロジェクトファイルに含まれる第2のネットワーク情報ファイルと、統合プロジェクトファイルに含まれる統合ネットワーク情報ファイルとの変換情報をさらに格納する。第1のネットワーク情報ファイル、第2のネットワーク情報

50

ファイルおよび統合ネットワーク情報ファイルの各々は、各プロジェクト内で使用されるノード識別子を含む。制御部は、変換情報を用いて、仮想統合制御ユニットおよびノードの各々間の通信データを第1の仮想制御ユニットまたは第2の仮想制御ユニットのいずれかに転送する。

【0012】

この開示によれば、制御システムは、変換情報を用いて、各仮想統合制御ユニットが管理するノード識別子と、実ネットワーク上の各ノードに割り当てられたノード識別子とを相互に変換し得る。

【0013】

上記の開示において、制御部は、第1のネットワーク情報ファイルで定義されている第1の制御周期と、第2のネットワーク情報ファイルで定義されている第2の制御周期との公約数である制御周期に基づいて、ノードの各々と通信する。

10

【0014】

この開示によれば、制御システムは、統合する各プロジェクトファイル内で使用される制御周期の公約数である制御周期で動作することにより、各仮想制御ユニットの制御周期（通信周期）で各ノードと通信し得る。

【0015】

上記の開示において、制御システムは、複数のプロジェクトファイルを統合する装置をさらに含む。装置は、第1のプロジェクトファイルと、第2のプロジェクトファイルとに基づいて、統合プロジェクトファイルを生成する。制御部は、装置から、第1のプロジェクトファイルと、第2のプロジェクトファイルと、統合プロジェクトファイルとを取得する。

20

【0016】

この開示によれば、制御システムは、装置によって生成された統合プロジェクトファイルを取得することができる。

【0017】

上記の開示において、装置は、制御システムの制御周期内における、第1の仮想制御ユニットの処理時間の占有率と、第2の仮想制御ユニットの処理時間の占有率との合計が100%を超えることに基づいて、エラーを出力する。

【0018】

この開示によれば、制御システムは、制御システムの制御周期内における各仮想制御ユニットの処理時間の占有率の合計が100%を超える場合、エラーを出力することにより、ユーザにプロジェクトが統合できないことを知らせることができる。

30

【0019】

上記の開示において、制御部は、第1のプロジェクトファイルと、第2のプロジェクトファイルとに基づいて、統合プロジェクトファイルを生成する。

【0020】

この開示によれば、制御システムは、制御部により統合プロジェクトファイルとを生成することができる。

【0021】

本開示の一例に従えば、制御ユニットを含む制御システムの制御方法が提供される。制御方法は、第1の制御ユニット用の第1のプロジェクトファイルと、第2の制御ユニット用の第2のプロジェクトファイルとに基づいて生成される統合プロジェクトファイルを参照するステップを含む。第1および第2のプロジェクトファイルの各々は、各プロジェクトファイルごとに独立して複数のノードの各々に割り当てられた複数のノード識別子を含む。統合プロジェクトファイルは、全てのノードの各々に一意に割り当て直したノード識別子を含む。全てのノードの各々に一意に割り当て直したノード識別子を用いて、通信するノードのノード識別子を特定するステップをさらに含む。

40

【0022】

この開示によれば、第1および第2のプロジェクトファイルに含まれる全てのノードの

50

各々に一意なノード識別子を設定することで、各ノードと通信可能になる。

【 0 0 2 3 】

上記の開示において、制御方法は、第 1 の仮想制御ユニットに、第 1 のプロジェクトファイルをインストールするステップと、第 2 の仮想制御ユニットに、第 2 のプロジェクトファイルをインストールするステップと、仮想統合制御ユニットに、統合プロジェクトファイルをインストールするステップと、仮想統合制御ユニットおよびノードの各々間の通信データを第 1 の仮想制御ユニットまたは第 2 の仮想制御ユニットのいずれかに転送するステップとをさらに含む。

【 0 0 2 4 】

この開示によれば、各制御ユニットを仮想制御ユニットとして動作させることにより統合し、また、各制御ユニットおよび各ノード間の通信を転送し得る。

10

【 0 0 2 5 】

上記の開示において、制御方法は、第 1 のプロジェクトファイルに含まれる第 1 のネットワーク情報ファイルおよび第 2 のプロジェクトファイルに含まれる第 2 のネットワーク情報ファイルと、統合プロジェクトファイルに含まれる統合ネットワーク情報ファイルとの変換情報にアクセスするステップをさらに含む。第 1 のネットワーク情報ファイル、第 2 のネットワーク情報ファイルおよび統合ネットワーク情報ファイルの各々は、各プロジェクト内で使用されるノード識別子を含む。制御方法は、変換情報を用いて、仮想統合制御ユニットおよびノードの各々間の通信データを第 1 の仮想制御ユニットまたは第 2 の仮想制御ユニットのいずれかに転送するステップをさらに含む。

20

【 0 0 2 6 】

この開示によれば、変換情報を用いて、各仮想統合制御ユニットが管理するノード識別子と、実ネットワーク上の各ノードに割り当てられたノード識別子とを相互に変換し得る。

【 0 0 2 7 】

上記の開示において、制御方法は、第 1 のネットワーク情報ファイルで定義されている第 1 の制御周期と、第 2 のネットワーク情報ファイルで定義されている第 2 の制御周期との公約数である制御周期に基づいて、ノードの各々と通信するステップをさらに含む。

【 0 0 2 8 】

この開示によれば、統合する各プロジェクトファイル内で使用される制御周期の公約数である制御周期で動作することにより、各仮想制御ユニットの制御周期（通信周期）で各ノードと通信し得る。

30

【 0 0 2 9 】

上記の開示において、制御方法は、制御システムの制御周期内における、第 1 の仮想制御ユニットの処理時間の占有率と、第 2 の仮想制御ユニットの処理時間の占有率との合計が 1 0 0 % を超えることに基づいて、エラーを出力するステップをさらに含む。

【 0 0 3 0 】

この開示によれば、制御システムの制御周期内における各仮想制御ユニットの処理時間の占有率の合計が 1 0 0 % を超える場合、エラーを出力することにより、ユーザにプロジェクトが統合できないことを知らせることができる。

【 発明の効果 】

40

【 0 0 3 1 】

ある実施の形態に従うと、複数の制御ユニット用のプロジェクトファイルを容易に統合することが可能である。

【 0 0 3 2 】

この開示内容の上記および他の目的、特徴、局面および利点は、添付の図面と関連して理解される本開示に関する次の詳細な説明から明らかとなるであろう。

【 図面の簡単な説明 】

【 0 0 3 3 】

【 図 1 】ある実施の形態が適用される制御ユニットの一例を示す図である。

【 図 2 】ある実施の形態に従う制御システム 1 を備えるネットワークシステム 1 0 の全体

50

構成を模式的に示す図である。

【図 3】ある実施の形態に従う制御システム 1 の構成例を示す外觀図である。

【図 4】ある実施の形態に従う制御システム 1 を構成する制御ユニット 20 のハードウェア構成例を示す模式図である。

【図 5】ある実施の形態に従う制御システム 1 に接続され得るサポート装置 6 のハードウェア構成例を示す模式図である。

【図 6】ある実施の形態に従う制御ユニット 20 の動作例を示す模式図である。

【図 7】制御ユニット 20 が転送するフレームの一例を示す模式図である。

【図 8】ネットワーク情報ファイルの統合の一例を示す模式図である。

【図 9】統合ネットワーク情報ファイル 645 の一例を示す模式図である。

10

【図 10】変換表 650 の一例を示す模式図である。

【図 11】統合制御ユニット 20D の制御周期の一例を示す模式図である。

【図 12】ネットワーク情報ファイル 615 に含まれる制御周期の一例を示す模式図である。

【図 13】複数のプロジェクトファイルの統合処理の手順の一例を示すフローチャートである。

【図 14】統合プロジェクトファイルをインストールされた制御ユニット 20（統合制御ユニット）の処理の手順の一例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】

【0034】

20

以下、図面を参照しつつ、本開示に係る技術思想の実施の形態について説明する。以下の説明では、同一の部品には同一の符号を付してある。それらの名称および機能も同じである。したがって、それらについての詳細な説明は繰り返さない。

【0035】

< A . 適用例 >

まず、図 1 を参照して、本実施の形態に従う技術が適用される場面について説明する。

【0036】

図 1 は、本実施の形態が適用される制御ユニットの一例を示す図である。図 1 を参照して、第 1 の制御ユニット 20A の第 1 のプロジェクトファイルと、第 2 の制御ユニット 20B の第 2 のプロジェクトファイルと、第 3 の制御ユニット 20C の第 3 のプロジェクトファイルを統合して統合制御ユニット 20D にインストールする手順を例に、本実施の形態に従う技術の概要について説明する。統合制御ユニット 20D は、第 1 の制御ユニット 20A、第 2 の制御ユニット 20B および第 3 の制御ユニット 20C と同一のハードウェア構成である。これ以降、第 1 の制御ユニット 20A、第 2 の制御ユニット 20B、第 3 の制御ユニット 20C および統合制御ユニット 20D を総称する場合は、制御ユニット 20 と呼ぶ。なお、制御ユニット 20 は、1 または複数の装置、または、システムとして構成されてもよい。

30

【0037】

なお、図 1 に示す場面は一例であり、本実施の形態に従う技術は、2 台以上の任意の数の制御ユニット 20 のプロジェクトファイルを統合することができる。プロジェクトファイルは、少なくとも、制御ユニット 20 が実行する制御プログラムと、フィールドネットワーク 7 上の構成を定義するネットワーク情報ファイルとを含む。

40

【0038】

PLC 等の制御ユニット 20 は、フィールドネットワーク 7（図 2 参照）を介して、複数のロボットまたは工作機等の制御対象 8 を制御する。ある局面において、フィールドネットワーク 7 は、Ethernet（登録商標）により実現されてもよい。この場合、ネットワーク情報ファイルは、一例として、ENI（Ethernet Network Information）ファイルであってもよい。なお、ネットワーク情報ファイルは、独自のフォーマットのファイルであってもよい。

【0039】

50

制御ユニット 20 は、これらの制御対象 8 をノードとして管理する。各ノードには、一意なノード識別子（ノード ID (Identifier)）が割り当てられる。第 1 の制御ユニット 20 A、第 2 の制御ユニット 20 B および第 3 の制御ユニット 20 C は、いずれも 1 ~ 10 のノード識別子を割り当てられたノードを管理する。

#### 【0040】

各制御ユニット 20 は、各ノード識別子を管理するためのネットワーク情報ファイルを二次記憶装置 33（図 4 参照）に格納する。各制御ユニット 20 は、当該ネットワーク情報ファイルと、制御プログラムとに基づいて、各ノードと通信する。ネットワーク情報ファイルおよび制御プログラムは、プロジェクトという単位で管理され得る。プロジェクトに含まれる各種データをまとめてプロジェクトファイルとも呼ぶ。

10

#### 【0041】

図 1 に示す例では、第 1 の制御ユニット 20 A は第 1 のプロジェクトファイルを格納し、第 2 の制御ユニット 20 B は第 2 のプロジェクトファイルを格納し、第 3 の制御ユニット 20 C は第 3 のプロジェクトファイルを格納する。言い換えれば、第 1 の制御ユニット 20 A は第 1 のプロジェクトによって動作し、第 2 の制御ユニット 20 B は第 2 のプロジェクトによって動作し、第 3 の制御ユニット 20 C は第 3 のプロジェクトによって動作する。

#### 【0042】

第 1 のプロジェクトファイルは、少なくとも、第 1 の制御ユニット 20 A が実行する第 1 の制御プログラムのオブジェクトファイルと、第 1 の制御ユニット 20 A が制御する 1 または複数のノード識別子を含む第 1 のネットワーク情報ファイルとを含む。同様に、第 2 のプロジェクトファイルは、少なくとも、第 2 の制御ユニット 20 B が実行する第 2 の制御プログラムのオブジェクトファイルと、第 2 の制御ユニット 20 B が制御する 1 または複数のノード識別子を含む第 2 のネットワーク情報ファイルとを含む。また、第 3 のプロジェクトファイルは、少なくとも、第 3 の制御ユニット 20 C が実行する第 3 の制御プログラムのオブジェクトファイルと、第 3 の制御ユニット 20 C が制御する 1 または複数のノード識別子を含む第 3 のネットワーク情報ファイルとを含む。

20

#### 【0043】

1 台の制御ユニット 20 が複数のノードと通信する場合、各ノード識別子はユニークでなければならない。同じフィールドネットワーク 7 上に同一のノード識別子を割り当てられた複数のノードが存在する場合、制御ユニット 20 は、これらのノードと正常に通信することができない。例えば、統合制御ユニット 20 D が、第 1 の制御ユニット 20 A、第 2 の制御ユニット 20 B および第 3 の制御ユニット 20 C のプロジェクトファイルをそのまま読み込んで使用した場合、ノード識別子「1 ~ 10」が重複するため正常に各ノードと通信できない。

30

#### 【0044】

そのため、本実施の形態に従う技術は、複数の制御ユニット 20 の各プロジェクトファイルを統合するときに、全てのノードに一意となるノード識別子を設定し直す。こうすることで、ユーザは、例えば、複数の制御ユニット 20 を用いて、分業で制御プログラムを作成し、運用時にはこれらの複数の制御プログラムを統合して運用することができる。

40

#### 【0045】

より具体的には、本実施の形態に従う技術は、複数の制御ユニットの各プロジェクトファイル（第 1 ~ 第 3 のプロジェクトファイル）を統合制御ユニット 20 D 内の複数の仮想制御ユニット 610（図 6 参照）の各々にインストールする。また、本実施の形態に従う技術は、第 1 ~ 第 3 のプロジェクトファイルから、全てのノードに一意となるノード識別子を設定し直した統合プロジェクトファイルを生成し、当該統合プロジェクトファイルを仮想統合制御ユニット 640（図 6 参照）にインストールする。言い換えれば、統合制御ユニット 20 D 内で、仮想的な第 1 の制御ユニット 20 A、仮想的な第 2 の制御ユニット 20 B および仮想的な第 3 の制御ユニット 20 C と、仮想統合制御ユニット 640 とが動作する。統合プロジェクトファイルは、全てのノードに対して一意に割り当て直した識別

50

子を含む。統合制御ユニット20Dは、当該全てのノードに対して一意に割り当て直した識別子により、通信するノードのノード識別子を特定し得る。

【0046】

ある局面において、統合制御ユニット20Dは、インストールされた各プロジェクトファイル(第1~第3のプロジェクトファイル)から統合プロジェクトファイルを生成してもよい。他の局面において、サポート装置6(図2参照)等を含む外部の情報処理装置が、各プロジェクトファイル(第1~第3のプロジェクトファイル)から統合プロジェクトファイルを生成して、各プロジェクトファイルおよび統合プロジェクトファイルを統合制御ユニット20Dにインストールしてもよい。

【0047】

統合制御ユニット20D内の仮想的な第1の制御ユニット20A、第2の制御ユニット20Bおよび第3の制御ユニット20Cは、ノード識別子「1~10」が重複する第1~第3のプロジェクトファイルを使用するため、直接ノードと通信することはできない。そこで、当該仮想統合制御ユニット640は、ノード識別子の変換処理を実行しながら、ノード(制御対象8)のいずれかから受信した通信データを複数の仮想制御ユニット610のいずれかに転送する。また、当該仮想統合制御ユニット640は、ノード識別子の変換処理を実行しながら、複数の仮想制御ユニット610のいずれかから受信した通信データをノードのいずれかに転送する。図1を例に説明すると、仮想統合制御ユニット640は、ノード101と通信したデータを仮想制御ユニット610に転送する場合、ノード識別子「101」(統合プロジェクト内でのノード識別子)をノード識別子「1」(第2のプロジェクト内でのノード識別子)に変換する。

【0048】

上記のように、本実施の形態に従う技術は、統合制御ユニット20D内で、複数の仮想制御ユニット610と、仮想統合制御ユニット640とを動作させ、仮想統合制御ユニット640にノード識別子の変換処理を実行させる。こうすることで、ユーザは、複数のプロジェクトファイルに変更を加えることなく、これらの複数のプロジェクトファイルを統合して使用し得る。

【0049】

さらに、本実施の形態に従う技術は、統合前の各プロジェクトファイル(第1~第3のプロジェクトファイル)自体には変更を加えない。そのため、ユーザは、並列的に複数のプロジェクトファイルを作成して、当該複数のプロジェクトファイルを容易に統合することができる。また、統合後の制御プログラムに変更がある場合でも、ユーザは統合前のプロジェクトファイルの各々を個別に編集して再度統合処理を実行するだけでよく、プロジェクトファイルの保守性も向上する。

【0050】

< B . システム構成 >

次に、図2を参照して、本実施の形態に従うネットワークシステム10の構成例について説明する。これ以降の説明における制御システム1のようなシステムは、1または複数の装置から構成されてもよい。また、当該システムは、装置の一部または他の装置と連携するものであってもよい。

【0051】

図2は、本実施の形態に従う制御システム1を備えるネットワークシステム10の全体構成を模式的に示す図である。上述した制御ユニット20は、制御システム1の一部として実現され得る。

【0052】

ネットワークシステム10は、構成として、制御システム1、サーバ装置2、表示装置3およびゲートウェイ(GW: Gateway)4を備える。これらの構成は、ネットワーク5を介して、相互に接続され得る。また、ネットワーク5は、ゲートウェイ4を介して、外部ネットワークであるインターネットに接続されている。ある局面において、ネットワーク5は、一般的なネットワークプロトコルであるイーサネット(登録商標)またはE t h

10

20

30

40

50

erNet / IP (登録商標) により実現されてもよい。

【0053】

制御システム1は、フィールドネットワーク7を介して、フィールドの設備および装置、ならびに、それらに配置されている各種デバイス(センサまたはアクチュエータ等)を含む制御対象8に接続されている。フィールドネットワーク7は、データの到達時間が保証される、定周期通信を行うバスまたはネットワークを採用することが好ましい。

【0054】

サポート装置6は、ユーザが制御システム1を運用するのを支援する支援ツールを提供する。また、サポート装置6は、制御システム1にプログラムをインストールする機能を備えていてもよい。ある局面において、サポート装置6は、パーソナルコンピュータ、タブレット、スマートフォン、またはその他の任意の情報処理装置であってもよい。

10

【0055】

一例として、サポート装置6は、USB(Universal Serial Bus)により、着脱可能に制御システム1に接続される。このUSB通信には、通信のセキュリティを確保するために、ユーザ認証を行なうための通信プロトコルが採用され得る。他の例として、サポート装置6は、ネットワーク5を介して制御システム1と通信してもよい。

【0056】

サーバ装置2は、一例として、データベースシステム、製造実行システム(MES: Manufacturing Execution System)等である。製造実行システムは、制御対象の製造装置または設備からの情報を取得して、生産全体を監視および管理し、例えば、オーダー情報、品質情報、出荷情報その他の生産に関する情報等を扱うこともできる。また、他の例として、サーバ装置2は、情報系サービス(制御対象から各種情報を取得して、マクロ的またはミクロ的な分析等を行う処理)を提供する装置であってもよい。

20

【0057】

表示装置3は、ユーザからの操作を受けて、制御システム1に対してユーザ操作に応じたコマンド等を出力するとともに、制御システム1での演算結果等をグラフィカルに表示する。ある局面において、表示装置3は、液晶ディスプレイまたは有機EL(Electro-Luminescence)ディスプレイ等の任意の出力装置を備えていてもよい。また、表示装置3は、タッチパネルまたはスイッチ等の任意の入力装置を備えていてもよい。

【0058】

ゲートウェイ4は、ネットワーク5と外部ネットワーク(インターネット)との間のプロトコル変換と、ファイアウォールとしての処理とを実行する。

30

【0059】

< C . ハードウェア構成 >

次に、図3～図5を参照して、本実施の形態に従う主な装置またはシステム(制御システム1、制御ユニット20、および、サポート装置6)のハードウェア構成について説明する。

【0060】

( a . 制御システム1の外観 )

図3は、本実施の形態に従う制御システム1の構成例を示す外観図である。制御システム1は、制御ユニット20、セキュリティユニット21、セーフティユニット22、1または複数の機能ユニット23、および電源ユニット24を含む。ある局面において、制御システム1は、制御ユニット20、セキュリティユニット21、セーフティユニット22、機能ユニット23、および電源ユニット24の各々を1または複数台含んでいてもよい。

40

【0061】

制御ユニット20とセキュリティユニット21とは、PCI Expressのバス等を介して相互に接続される。また、制御ユニット20、セーフティユニット22、1または複数の機能ユニット23、および電源ユニット24は、内部バスを介して相互に接続されている。

【0062】

50

制御ユニット 20 は、制御プログラムを実行することで、制御対象を制御する。制御プログラムは、制御対象である設備および装置、ならびに、それらに配置されている各種デバイス（センサまたはアクチュエータ等）との間で信号を遣り取りする I/O リフレッシュ、制御演算処理等のプログラムを含む。具体的には、I/O リフレッシュは、制御ユニット 20 において算出される指令値を制御対象へ出力、あるいは、制御対象からの入力値を収集する。制御演算処理は、例えば、I/O リフレッシュにより収集した入力値に基づいた指令値または制御量を算出する。このような機能を備える制御プログラムは、制御対象の要求仕様に従ってユーザまたは開発会社が作成するプログラムを含む「ユーザプログラム」の一例でもある。

#### 【0063】

セキュリティユニット 21 は、制御システム 1 の、より特定的には制御ユニット 20 のセキュリティを設定する。このセキュリティの設定（ACL テーブル）は、制御プログラムの意図しない複製、すなわち不正な複製を防止するための設定を含む。

#### 【0064】

セーフティユニット 22 は、制御ユニット 20 とは独立して、制御対象に関するセーフティ機能を実現するための制御演算を実行する。機能ユニット 23 は、制御システム 1 による様々な制御対象に対する制御を実現するための各種機能を提供する。

#### 【0065】

機能ユニット 23 は、典型的には、I/O ユニット、セーフティ I/O ユニット、通信ユニット、モーションコントローラユニット、温度調整ユニット、パルスカウンタユニット等を包含し得る。I/O ユニットとしては、例えば、デジタル入力（DI）ユニット、デジタル出力（DO）ユニット、アナログ出力（AI）ユニット、アナログ出力（AO）ユニット、パルスキャッチ入力ユニット、および、複数の種類を混合させた複合ユニット等が挙げられる。セーフティ I/O ユニットは、セーフティ制御に係る I/O 処理を担当する。電源ユニット 24 は、制御システム 1 を構成する各ユニットに対して、所定電圧の電源を供給する。

#### 【0066】

（b. 制御ユニット 20 のハードウェア構成）

次に、本実施の形態に従う制御システム 1 が含む制御ユニット 20 のハードウェア構成例について説明する。

#### 【0067】

図 4 は、本実施の形態に従う制御システム 1 を構成する制御ユニット 20 のハードウェア構成例を示す模式図である。図 4 を参照して、制御ユニット 20 は、主たるコンポーネントとして、CPU（Central Processing Unit）または GPU（Graphical Processing Unit）等のプロセッサ 31、チップセット 32、二次記憶装置 33、主記憶装置 34、通信コントローラ 35、インジケータ 36、スイッチインターフェイス 37、内部バスコントローラ 39、ネットワークコントローラ 40, 41, 42、メモ리카ードインターフェイス 43、および USB コントローラ 44 を含む。

#### 【0068】

プロセッサ 31 は、二次記憶装置 33 に格納された各種プログラムを読み出して、当該各種プログラムを主記憶装置 34 に展開して実行することで、制御演算およびサービス処理を含む各種の処理を実現する。チップセット 32 は、プロセッサ 31 と各コンポーネントとの間のデータの遣り取りを仲介することで、制御ユニット 20 全体としての処理を実現する。

#### 【0069】

主記憶装置 34 は、DRAM（Dynamic Random Access Memory）または SRAM（Static Random Access Memory）等の揮発性記憶装置を備える。これら揮発性記憶装置の少なくとも一部は、復号済み制御プログラム 51 を格納するための揮発性記憶領域 50 を構成する。ある局面において、復号済み制御プログラム 51 は、例えば、複数の仮想制御ユニット用の制御プログラムと、仮想統合制御ユニット用の制御プログラムと

10

20

30

40

50

を含み得る。また、復号済み制御プログラム 5 1 は、複数の仮想制御ユニット用のネットワーク情報ファイルと、仮想統合制御ユニット用のネットワーク情報ファイルとを参照し得る。

#### 【 0 0 7 0 】

二次記憶装置 3 3 は、典型的には、例えば、HDD (Hard Disk Drive) または SSD (Solid State Drive)、ROM (Read Only Memory)、EPROM (Erasable Programmable Read Only Memory)、EEPROM (Electrically Erasable Programmable Read-Only Memory) 等の不揮発性記憶装置を備える。これら不揮発性記憶装置の少なくとも一部は、暗号化済み制御プログラム 4 9 を格納するための不揮発性記憶領域 4 8 を構成する。

10

#### 【 0 0 7 1 】

二次記憶装置 3 3 は、さらに、OS (Operating System) を含むシステムプログラム 4 5、サービスプログラム 4 6、および、統合プログラム 4 0 1 等を格納する。システムプログラム 4 5 は、復号済み制御プログラム 5 1 およびサービスプログラム 4 6 等のユーザプログラムが動作するためのプログラム実行環境を提供する。

#### 【 0 0 7 2 】

統合プログラム 4 0 1 は、複数のプロジェクトファイルを統合する。より具体的には、統合プログラム 4 0 1 は、複数のプロジェクトファイルから、統合プロジェクトファイルを生成する。統合プロジェクトファイルは、仮想統合制御ユニット 6 4 0 上で動作する統合制御プログラムと、統合ネットワーク情報ファイル 6 4 5 (図 6 参照) と、ノード識別子を変換するための変換表 6 5 0 (図 6 参照) とを含む。

20

#### 【 0 0 7 3 】

変換表 6 5 0 は、統合前の各プロジェクト内 (またはプロジェクトファイル内) のネットワーク情報ファイルに含まれるノード識別子と、統合プロジェクト内 (または統合プロジェクトファイル内) のネットワーク情報ファイルに含まれるノード識別子との変換処理に使用される。図 1 を例に説明すると、制御ユニット 2 0 は、例えば、ノード識別子「1 0 1」(統合プロジェクト内で使用されるノード識別子) と、ノード識別子「1」(第 2 のプロジェクト内で使用されるノード識別子) との変換処理に、変換表 6 5 0 を使用する。言い換えれば、変換表 6 5 0 は、第 1 のネットワーク情報ファイル 6 1 5 A、第 2 のネットワーク情報ファイル 6 1 5 B および第 3 のネットワーク情報ファイル 6 1 5 C と、統合プロジェクトファイルに含まれる統合ネットワーク情報ファイル 6 4 5 との変換情報である。ある局面において、サポート装置 6 が統合ツール 5 5 2 (図 5 参照) を含む場合、制御ユニット 2 0 は統合プログラム 4 0 1 を含まなくてもよい。

30

#### 【 0 0 7 4 】

通信コントローラ 3 5 は、バス 5 2 を介して、セキュリティユニット 2 1 とデータを送受信する。通信コントローラ 3 5 は、例えば、PCI Express 等のバスに対応した通信チップにより実現され得る。

#### 【 0 0 7 5 】

インジケータ 3 6 は、制御ユニット 2 0 の動作状態等を通知するものであり、ユニット表面に配置された 1 または複数の LED (Light Emitting Diode) 等で構成される。スイッチインターフェイス 3 7 は、一例として、ディップスイッチ 3 8 に接続されており、当該ディップスイッチ 3 8 の ON または OFF の信号をプロセッサ 3 1 に出力する。

40

#### 【 0 0 7 6 】

内部バスコントローラ 3 9 は、制御システム 1 を構成するセーフティユニット 2 2 と、1 または複数の機能ユニット 2 3 との間で、内部バスを介してデータを送受信する。この内部バスには、メーカー固有の通信プロトコルを用いてもよいし、いずれかの産業用ネットワークプロトコルと同一あるいは準拠した通信プロトコルを用いてもよい。

#### 【 0 0 7 7 】

ネットワークコントローラ 4 0, 4 1, 4 2 の各々は、ネットワークを介した任意のデバイスとの間のデータの遣り取りを担当する。ネットワークコントローラ 4 0, 4 1, 4

50

2は、EtherCAT、Ethernet/IP（登録商標）、DeviceNet（登録商標）、CompoNet（登録商標）等の産業用ネットワークプロトコルを採用してもよい。

【0078】

メモリカードインターフェイス43は、SDカード等のメモリカード53を着脱可能に構成されており、メモリカード53に対してユーザプログラムまたは各種設定等のデータを書込み、あるいは、メモリカード53から当該ユーザプログラムまたは各種設定等のデータを読み出すことが可能になっている。USBコントローラ44は、USB接続を介して、サポート装置6を含む任意の情報処理装置とデータを送受信し得る。ある局面において、制御ユニット20は、メモリカードインターフェイス43以外にも、任意の記憶媒体を接続可能な外部機器インターフェイスを備えていてもよい。

10

【0079】

図4は、プロセッサ31がプログラムを実行することで必要な機能が提供される構成例を示しているが、これらの提供される機能の一部または全部は、専用のハードウェア回路（例えば、ASIC（Application Specific Integrated Circuit）またはFPGA（Field-Programmable Gate Array）等）を用いて実装されてもよい。あるいは、制御ユニット20の主要部を、汎用的なアーキテクチャに従うハードウェア（例えば、汎用パソコンをベースとした産業用パソコン）を用いて実現してもよい。この場合には、仮想化技術を用いて、用途の異なる複数のOSを並列的に実行させるとともに、各OS上で必要なアプリケーションを実行させるようにしてもよい。

20

【0080】

（c．サポート装置6のハードウェア構成）

次に、本実施の形態に従う制御システム1に接続され得るサポート装置6のハードウェア構成例について説明する。

【0081】

図5は、本実施の形態に従う制御システム1に接続され得るサポート装置6のハードウェア構成例を示す模式図である。サポート装置6は、一例として、汎用的なアーキテクチャに従う装置（パーソナルコンコンピュータまたはタブレット等）を用いて実現され得る。

【0082】

サポート装置6は、CPUまたはGPU等のプロセッサ91、主記憶装置92、入力部93、出力部94、二次記憶装置95、光学ドライブ96、および通信インターフェイス97を含む。これらのコンポーネントは、プロセッサバス98を介して接続されている。主記憶装置92および二次記憶装置95は、それぞれ、制御ユニット20の主記憶装置34および二次記憶装置33と同様に構成することができるので、それらの説明を繰返さない。

30

【0083】

プロセッサ91は、二次記憶装置95に格納されたプログラム（一例として、OS554およびサポートプログラム553）を読み出して、主記憶装置92に展開して実行することで、各種処理を実現する。

【0084】

二次記憶装置95は、基本的な機能を実現するためのOS554に加えて、サポート装置6としての機能を提供するためのサポートプログラム553を格納する。サポート装置6（実質的にはプロセッサ91）は、サポートプログラム553を実行することで、サポート装置6が提供する各種サポートツールの機能を実現する。当該サポートツールは、サポート装置6におけるプログラムの開発環境を提供する。

40

【0085】

また、二次記憶装置95は、サポートツールを用いて作成された制御プログラム551と、統合ツール552とを格納する。制御プログラム551は、制御ユニット20で実行されるプログラムのソースコードであってもよい。また、制御プログラム551は、制御ユニット20で実行されるプログラムの実行ファイル（オブジェクトファイル）を含んで

50

いてもよい。

【 0 0 8 6 】

統合ツール 5 5 2 は、複数のプロジェクトファイルを統合する。より具体的には、統合ツール 5 5 2 は、複数のプロジェクトファイルから、統合プロジェクトファイルを生成する。統合プロジェクトファイルは、仮想統合制御ユニット 6 4 0 上で動作する統合制御プログラムと、統合ネットワーク情報ファイル 6 4 5 と、ノード識別子を変換するための変換表 6 5 0 とを含む。ある局面において制御ユニット 2 0 が統合プログラム 4 0 1 を含む場合、サポート装置 6 は統合ツール 5 5 2 ( 図 5 参照 ) を含まなくてもよい。

【 0 0 8 7 】

また、ある局面において、二次記憶装置 9 5 は、暗号化済み制御プログラムを格納してもよい。さらに、二次記憶装置 9 5 は、制御プログラム 5 5 1 の暗号化のための鍵、および暗号化処理を実行する暗号化処理プログラムを格納してもよい。また、二次記憶装置 9 5 は、暗号化処理プログラムよりも簡易で高速な暗号化処理を実行する簡易暗号化処理プログラムを格納してもよい。プロセッサ 9 1 は、簡易暗号化処理プログラムを実行することで、簡易暗号化済み制御プログラムを生成し得る。

10

【 0 0 8 8 】

入力部 9 3 は、キーボードまたはマウス等で構成され、ユーザの操作を受け付ける。ある局面において、入力部 9 3 は、さらに、カメラ、マイク、およびその他の任意のデバイスを含んでもよい。他の局面において、これらのデバイスは、通信インターフェイス 9 7 を介してサポート装置 6 に接続されてもよい。出力部 9 4 は、ディスプレイ、各種インジケータ、プリンタ等で構成され、プロセッサ 9 1 からの処理結果等を出力する。

20

【 0 0 8 9 】

サポート装置 6 は、光学ドライブ 9 6 を有する。光学ドライブ 9 6 は、記録媒体 5 5 0 ( 例えば、DVD ( Digital Versatile Disc ) 等の光学記録媒体 ) から、その中に格納されたプログラムを読み取り、当該プログラムを二次記憶装置 9 5 等にインストールする。

【 0 0 9 0 】

通信インターフェイス 9 7 は、USB またはイーサネット等の任意の通信媒体を介して、制御システム 1 が備える制御ユニット 2 0 またはセキュリティユニット 2 1 と、データを送受信し得る。

【 0 0 9 1 】

サポート装置 6 で実行されるサポートプログラム 5 5 3 等は、コンピュータ読取可能な記録媒体 5 5 0 を介してインストールされてもよいが、ネットワーク上のサーバ装置等からダウンロードする形でインストールされてもよい。また、本実施の形態に従うサポート装置 6 が提供する機能は、OS 5 5 4 が提供するモジュールの一部を利用する形で実現され得る。

30

【 0 0 9 2 】

図 5 は、プロセッサ 9 1 がプログラムを実行することで、サポート装置 6 として必要な機能が提供される構成例を示しているが、これらの提供される機能の一部または全部は、専用のハードウェア回路 ( 例えば、ASIC または FPGA 等 ) を用いて実装されてもよい。また、本実施の形態では、制御システム 1 の稼動中に、サポート装置 6 が、制御システム 1 から取り外されていてもよい。

40

【 0 0 9 3 】

< D . 制御ユニット 2 0 の動作 >

次に、図 6 および図 7 を参照して、本実施の形態に従う制御ユニット 2 0 の動作について説明する。

【 0 0 9 4 】

図 6 は、本実施の形態に従う制御ユニット 2 0 の動作例を示す模式図である。図 6 に示す例において、制御ユニット 2 0 は、図 1 を参照して説明した統合制御ユニット 2 0 D として動作する。すなわち、制御ユニット 2 0 は、3 つのプロジェクトファイル ( 第 1 の制御ユニット 2 0 A の第 1 のプロジェクトファイル、第 2 の制御ユニット 2 0 B の第 2 のプ

50

プロジェクトファイルおよび第3の制御ユニット20Cの第3のプロジェクトファイル)を統合して使用するものとする。また、各ノード識別子も図1に示すノード識別子と同じであるとする。なお、図6に示す構成は一例であり、制御ユニット20は、2台以上の任意の台数の制御ユニットのプロジェクトファイルを統合し得る。

【0095】

(a. 制御ユニット20の機能構成)

制御ユニット20は、主な機能構成として、第1の仮想制御ユニット610Aと、第2の仮想制御ユニット610Bと、第3の仮想制御ユニット610Cと(総称する場合は「仮想制御ユニット610」と呼ぶ)、仮想ネットワークドライバ620, 630と、仮想統合制御ユニット640と、ネットワークドライバ660と、ハードウェア670とを含む。ある局面において、仮想制御ユニット610と、仮想ネットワークドライバ620, 630と、仮想統合制御ユニット640と、ネットワークドライバ660とは、プログラムにより実現されてもよい。他の局面において、複数の仮想制御ユニット610と、仮想ネットワークドライバ620, 630と、仮想統合制御ユニット640と、ネットワークドライバ660とは、ハードウェアとして実現されてもよい。

10

【0096】

制御ユニット20は、読み込んだプロジェクトファイルの数だけ仮想制御ユニット610を含む。図6に示す例では、制御ユニット20は、3つのプロジェクトファイルを読み込んでいるため、3台の仮想制御ユニット610を含む。ある局面において、プロセッサ31は、仮想制御ユニットを実現するプログラムを実行してもよい。

20

【0097】

ハードウェア670は、一例として、図4に示すハードウェアである。ネットワークドライバ660は、システムプログラム45等がネットワークコントローラ40, 41, 42等を使用するときを使用するドライバである。

【0098】

各仮想制御ユニット610は、制御ユニット20と同じ動作をするプログラムである。仮想制御ユニット610は、制御ユニット20と同様に、インストールされたプロジェクトファイル(制御プログラム、およびENIファイル等のネットワーク情報ファイル)に基づいて動作する。例えば、第1の制御ユニット20Aに対応する仮想制御ユニット610には、第1の制御ユニット20A用の第1のプロジェクトファイルがインストールされる。これ以降、一例として、第1の仮想制御ユニット610Aは、第1の制御ユニット20Aに対応し、第2の仮想制御ユニット610Bは、第2の制御ユニット20Bに対応し、第3の仮想制御ユニット610Cは、第3の制御ユニット20Cに対応するものとして説明する。

30

【0099】

各仮想制御ユニット610は、プロジェクトファイルをインストールされた制御ユニット20と同じ動作をする。例えば、第1のプロジェクトファイルをインストールされた第1の仮想制御ユニット610Aは、第1のプロジェクトファイルをインストールされた第1の制御ユニット20Aと同じ動作をする。第1の仮想制御ユニット610A、第2の仮想制御ユニット610Bおよび第3の仮想制御ユニット610Cの各々は、第1~第3のプロジェクトファイルの各々をインストールされる。

40

【0100】

仮想統合制御ユニット640は、各仮想制御ユニット610と、各ノードとの通信を仲介する。また、仮想統合制御ユニット640は、仮想ネットワークドライバ620, 630を介して、各仮想制御ユニット610と通信する。また、仮想統合制御ユニット640は、ネットワークドライバ660およびハードウェア670を介して、フィールドネットワーク7上の各ノード(制御対象8)と通信する。仮想統合制御ユニット640は、統合プロジェクトファイルをインストールされる。

【0101】

(b. ノード識別子の管理)

50

次に、図 6 に示す構成において、ノード識別子がどのように管理されるのかについて説明する。第 1 の仮想制御ユニット 6 1 0 A、第 2 の仮想制御ユニット 6 1 0 B および第 3 の仮想制御ユニット 6 1 0 C は、独立して動作し得る。より具体的には、例えば、第 1 の仮想制御ユニット 6 1 0 A は、第 1 のプロジェクトファイルに基づいて、各ノードと通信する。このとき、第 1 の仮想制御ユニット 6 1 0 A の動作は、第 2 の仮想制御ユニット 6 1 0 B および第 3 の仮想制御ユニット 6 1 0 C の動作から完全に独立していてもよいし、同期していてもよい。

#### 【 0 1 0 2 】

各仮想制御ユニット 6 1 0 は、ノードへの命令を仮想統合制御ユニット 6 4 0 に送信する。また、各仮想制御ユニット 6 1 0 は、仮想統合制御ユニット 6 4 0 から、ノードからのデータを受信し得る。仮想統合制御ユニット 6 4 0 は、受信したノードへの命令に含まれるノード識別子を各仮想制御ユニット 6 1 0 において使用されるノード識別子から、仮想統合制御ユニット 6 4 0 において使用されるノード識別子に変換し、変換後のノードへの命令をいずれかのノードに送信する。逆に、仮想統合制御ユニット 6 4 0 は、受信したノードからのデータに含まれるノード識別子を仮想統合制御ユニット 6 4 0 において使用されるノード識別子から、各仮想制御ユニット 6 1 0 において使用されるノード識別子に変換し、変換後のノードからのデータをいずれかの仮想制御ユニット 6 1 0 に送信する。

10

#### 【 0 1 0 3 】

例えば、第 1 の仮想制御ユニット 6 1 0 A は、第 1 のネットワーク情報ファイル 6 1 5 A ( E N I ファイル等 ) に基づいて、通信するノードに対してノード識別子「 1 ~ 1 0 」を割り当てている。同様に、第 2 の仮想制御ユニット 6 1 0 B は、第 2 のネットワーク情報ファイル 6 1 5 B ( E N I ファイル等 ) に基づいて、通信するノードに対してノード識別子「 1 ~ 1 0 」を割り当てている。また、第 3 の仮想制御ユニット 6 1 0 C は、第 3 のネットワーク情報ファイル 6 1 5 C ( E N I ファイル等 ) に基づいて、通信するノードに対してノード識別子「 1 ~ 1 0 」を割り当てている。なお、第 1 のネットワーク情報ファイル 6 1 5 A、第 2 のネットワーク情報ファイル 6 1 5 B および第 3 のネットワーク情報ファイル 6 1 5 C を総称する場合は、ネットワーク情報ファイル 6 1 5 と呼ぶ。

20

#### 【 0 1 0 4 】

これに対して、仮想統合制御ユニット 6 4 0 は、第 1 の仮想制御ユニット 6 1 0 A が通信するノードにノード識別子「 2 0 1 ~ 2 1 0 」を割り当て、第 2 の仮想制御ユニット 6 1 0 B が通信するノードにノード識別子「 1 0 1 ~ 1 1 0 」を割り当て、第 3 の仮想制御ユニット 6 1 0 C が通信するノードにノード識別子「 1 ~ 1 0 」を割り当てている。統合ネットワーク情報ファイル 6 4 5 は、これらの新しく割り当てられたノード識別子 ( ノード識別子「 1 ~ 1 0 」、 「 1 0 1 ~ 1 1 0 」および「 2 0 1 ~ 2 1 0 」 ) を含む。

30

#### 【 0 1 0 5 】

変換表 6 5 0 は、各ネットワーク情報ファイル 6 1 5 に含まれるノード識別子と、統合ネットワーク情報ファイル 6 4 5 に含まれるノード識別子との対応関係を管理する。例えば、変換表 6 5 0 は、第 1 のネットワーク情報ファイル 6 1 5 A に含まれるノード識別子「 1 ~ 1 0 」と、統合ネットワーク情報ファイル 6 4 5 に含まれるノード識別子「 2 0 1 ~ 2 1 0 」とを対応付けて管理する。同様に、変換表 6 5 0 は、第 2 のネットワーク情報ファイル 6 1 5 B に含まれるノード識別子「 1 ~ 1 0 」と、統合ネットワーク情報ファイル 6 4 5 に含まれるノード識別子「 1 0 1 ~ 1 1 0 」とを対応付けて管理する。また、変換表 6 5 0 は、第 3 のネットワーク情報ファイル 6 1 5 C に含まれるノード識別子「 1 ~ 1 0 」と、統合ネットワーク情報ファイル 6 4 5 に含まれるノード識別子「 1 ~ 1 0 」とを対応付けて管理する。

40

#### 【 0 1 0 6 】

上記のように、第 1 の仮想制御ユニット 6 1 0 A、第 2 の仮想制御ユニット 6 1 0 B および第 3 の仮想制御ユニット 6 1 0 C の各々は、第 1 のネットワーク情報ファイル 6 1 5 A、第 2 のネットワーク情報ファイル 6 1 5 B および第 3 のネットワーク情報ファイル 6 1 5 C のみを管理すればよい。仮想統合制御ユニット 6 4 0 は、統合ネットワーク情報フ

50

ファイル 6 4 5 と、変換表 6 5 0 とを管理する。仮想統合制御ユニット 6 4 0 は、変換表 6 5 0 に基づいて、統合ネットワーク情報ファイル 6 4 5 内のノード識別子と、各ネットワーク情報ファイル 6 1 5 内のノード識別子とを相互に変換し得る。こうすることで、仮想統合制御ユニット 6 4 0 は、各仮想制御ユニット 6 1 0 が使用するノード識別子が重複していたとしても、各仮想制御ユニット 6 1 0 および各ノード間の通信を相互に転送し得る。

【 0 1 0 7 】

( c . 通信の流れの一例 )

次に、第 2 の仮想制御ユニット 6 1 0 B およびノード 2 0 1 ( ノード識別子「 2 0 1 」のノード)間の通信を例に制御ユニット 2 0 の動作を説明する。

【 0 1 0 8 】

第 2 の仮想制御ユニット 6 1 0 B がノード 2 0 1 に命令を送信する場合、第 1 のステップにおいて、第 2 の仮想制御ユニット 6 1 0 B は、第 2 のネットワーク情報ファイル 6 1 5 B を参照して、仮想ネットワークドライバ 6 2 0 , 6 3 0 を介して、ノード識別子「 1 」を含む命令を仮想統合制御ユニット 6 4 0 に送信する。

【 0 1 0 9 】

第 2 のステップにおいて、仮想統合制御ユニット 6 4 0 は、変換表 6 5 0 を参照して、第 2 のネットワーク情報ファイル 6 1 5 B 内のノード識別子「 1 」に対応する統合ネットワーク情報ファイル 6 4 5 内のノード識別子「 2 0 1 」を取得する。第 3 のステップにおいて、仮想統合制御ユニット 6 4 0 は、命令に含まれるノード識別子「 1 」をノード識別子「 2 0 1 」に変更する。

【 0 1 1 0 】

第 4 のステップにおいて、仮想統合制御ユニット 6 4 0 は、ネットワークドライバ 6 6 0 およびハードウェア 6 7 0 を介して、ノード識別子を変更された命令をフィールドネットワーク 7 に出力する。当該ノード識別子を変更された命令は、ノード 2 0 1 によって受信される。

【 0 1 1 1 】

ノード 2 0 1 が、第 2 の仮想制御ユニット 6 1 0 B にデータを送信する場合、第 1 のステップにおいて、ノード 2 0 1 は、ノード識別子「 2 0 1 」を含むデータをフィールドネットワーク 7 に出力する。そして、仮想統合制御ユニット 6 4 0 は、ネットワークドライバ 6 6 0 およびハードウェア 6 7 0 を介して、当該データを取得する。

【 0 1 1 2 】

第 2 のステップにおいて、仮想統合制御ユニット 6 4 0 は、変換表 6 5 0 を参照して、統合ネットワーク情報ファイル 6 4 5 内のノード識別子「 2 0 1 」に対応する第 2 のネットワーク情報ファイル 6 1 5 B 内のノード識別子「 1 」を取得する。第 3 のステップにおいて、仮想統合制御ユニット 6 4 0 は、命令に含まれるノード識別子「 2 0 1 」をノード識別子「 1 」に変更する。

【 0 1 1 3 】

第 4 のステップにおいて、仮想統合制御ユニット 6 4 0 は、仮想ネットワークドライバ 6 2 0 , 6 3 0 を介して、ノード識別子を変更されたデータを第 2 の仮想制御ユニット 6 1 0 B に送信する。

【 0 1 1 4 】

図 7 は、制御ユニット 2 0 が転送するフレームの一例を示す模式図である。制御ユニット 2 0 からノードに送信される命令およびノードから制御ユニット 2 0 に送信されるデータは、図 7 に示すフレームの形式で実現される。

【 0 1 1 5 】

フレーム 7 0 0 は、各仮想制御ユニット 6 1 0 および仮想統合制御ユニット 6 4 0 の間で送受信される。フレーム 7 1 0 は、仮想統合制御ユニット 6 4 0 および各ノードの間に送受信される。フレーム 7 0 0 , 7 1 0 は、ヘッダ ( Header )、データ ( Data ) および CRC ( Cyclic Redundancy Check ) を含む。さらに、ヘッダは、フレームタイプ ( Frame Type ) およびノード識別子 ( Node ID ) を含む。

10

20

30

40

50

## 【 0 1 1 6 】

第2の仮想制御ユニット610Bがノード201に命令を送信する場合、第2の仮想制御ユニット610Bは、ノード識別子701「1」を含むフレーム700を仮想統合制御ユニット640に送信する。次に、仮想統合制御ユニット640は、変換表650を参照して、ノード識別子701「1」をノード識別子711「201」に変更することにより、フレーム700をフレーム710に変換する。そして、仮想統合制御ユニット640は、フレーム710をノード201に送信する。言い換えれば、仮想統合制御ユニット640は、フレーム700のノード識別子701を書き換えて、当該ノード識別子701を書き換えられたフレーム700をノード201に転送している。

## 【 0 1 1 7 】

ノード201が第2の仮想制御ユニット610Bにデータを送信する場合、ノード201は、ノード識別子711「201」を含むフレーム710を仮想統合制御ユニット640に送信する。次に、仮想統合制御ユニット640は、変換表650を参照して、ノード識別子711「201」をノード識別子701「1」に変更することにより、フレーム710をフレーム700に変換する。そして、仮想統合制御ユニット640は、フレーム700を第2の仮想制御ユニット610Bに送信する。言い換えれば、仮想統合制御ユニット640は、フレーム710のノード識別子711を書き換えて、当該ノード識別子711を書き換えられたフレーム710を第2の仮想制御ユニット610Bに転送している。

## 【 0 1 1 8 】

< E .プロジェクトファイルの統合手順 >

次に、図8～図12を参照して、プロジェクトファイルの統合の手順について説明する。プロジェクトファイルの統合は、少なくとも、各プロジェクトファイルに含まれるネットワーク情報ファイル615から、統合ネットワーク情報ファイル645、変換表650および仮想統合制御ユニット640用の制御プログラムを生成することを含む。

## 【 0 1 1 9 】

図8は、ネットワーク情報ファイルの統合の一例を示す模式図である。制御ユニット20またはサポート装置6は、複数のプロジェクト内のネットワーク情報ファイル615（ENIファイル等）に基づいて、統合ネットワーク情報ファイル645（統合ENIファイル等）を生成する。

## 【 0 1 2 0 】

より具体的には、制御ユニット20が統合ネットワーク情報ファイル645を生成する場合、まず、制御ユニット20は、複数のプロジェクトファイルをいずれかのインターフェイス（通信コントローラ35、USBコントローラ44またはメモリカードインターフェイス43等）から取得して、当該複数のプロジェクトファイルを二次記憶装置33に保存する。

## 【 0 1 2 1 】

次に、制御ユニット20は、統合プログラム401を用いて、複数のプロジェクト内のネットワーク情報ファイル615に基づいて、統合ネットワーク情報ファイル645を生成する。同時に、制御ユニット20は、統合プログラム401を用いて、複数のプロジェクト内のネットワーク情報ファイル615に基づいて、変換表650および統合制御プログラムを生成する。最後に、制御ユニット20は、生成した統合ネットワーク情報ファイル645、変換表650および統合制御プログラムを統合プロジェクトファイルに含めて二次記憶装置33に保存する。なお、統合制御プログラムは、主にフレームの転送処理およびノード識別子の変換処理を含む。

## 【 0 1 2 2 】

サポート装置6が統合ネットワーク情報ファイル645を生成する場合、サポート装置6は、統合ツール552を用いて、制御ユニット20と同様の手順で統合プロジェクトファイル（統合ネットワーク情報ファイル645、変換表650および制御プログラムを含む）を生成する。次に、サポート装置6は、複数のプロジェクトファイルおよび統合プロジェクトファイルを制御ユニット20に送信する。

10

20

30

40

50

## 【 0 1 2 3 】

図 9 は、統合ネットワーク情報ファイル 6 4 5 の一例を示す模式図である。統合ネットワーク情報ファイルは、全てのノードに一意に割り当てられたノード識別子 9 1 0 を含む。統合プログラム 4 0 1 または統合ツール 5 5 2 は、複数のネットワーク情報ファイルに含まれる複数のノード識別子を取得する。そして、統合プログラム 4 0 1 または統合ツール 5 5 2 は、取得した複数のノード識別子に重複がある場合、図 9 に示される例のように、全てのノードに一意に識別可能なノード識別子 9 1 0 を割り当て直すことで、統合ネットワーク情報ファイル 6 4 5 を生成する。

## 【 0 1 2 4 】

図 1 0 は、変換表 6 5 0 の一例を示す模式図である。変換表 6 5 0 は、装置識別子 1 0 1 0 と、第 1 のノード識別子 1 0 2 0 と、第 2 のノード識別子 1 0 3 0 とを含む。装置識別子 1 0 1 0 は、仮想制御ユニット 6 1 0 を一意に特定する。第 1 のノード識別子 1 0 2 0 は、仮想制御ユニット 6 1 0 が使用するノード識別子である。第 2 のノード識別子 1 0 3 0 は、仮想統合制御ユニット 6 4 0 が使用するノード識別子である。

10

## 【 0 1 2 5 】

第 2 の仮想制御ユニット 6 1 0 B およびノード 2 0 1 (ノード識別子「 2 0 1 」のノード)間の通信を例に説明すると、装置識別子 1 0 1 0 「 2 」は、第 2 の仮想制御ユニット 6 1 0 B を特定する。第 1 のノード識別子 1 0 2 0 は、第 2 の仮想制御ユニット 6 1 0 B が使用するノード識別子「 1 」であり、第 2 のノード識別子 1 0 3 0 は、仮想統合制御ユニット 6 4 0 が使用するノード識別子「 2 0 1 」である。仮想統合制御ユニット 6 4 0 は、これらの情報を参照することで、各仮想制御ユニット 6 1 0 および各ノード間の通信の転送処理を実行する。

20

## 【 0 1 2 6 】

図 1 1 は、統合制御ユニット 2 0 D の制御周期の一例を示す模式図である。通常、各制御ユニット 2 0 は、プロジェクトファイルで定められた制御周期に基づいて各ノードと通信する。図 1 1 に示す例において、第 1 の制御ユニット 2 0 A は、制御周期「 1 . 5 m s 」(ミリ秒)でグループ A 内の各ノードと通信する。また、第 2 の制御ユニット 2 0 B は、制御周期「 1 . 5 m s 」でグループ B 内の各ノードと通信する。また、第 3 の制御ユニット 2 0 C は、制御周期「 2 . 0 m s 」でグループ C 内の各ノードと通信する。

## 【 0 1 2 7 】

統合制御ユニット 2 0 D は、第 1 ~ 第 3 のプロジェクトファイルを統合して使用する場合、制御周期「 1 . 5 m s 」でグループ A 内の各ノードと通信し、制御周期「 1 . 5 m s 」でグループ B 内の各ノードと通信し、制御周期「 2 . 0 m s 」でグループ C 内の各ノードと通信する必要がある。

30

## 【 0 1 2 8 】

そこで、統合制御ユニット 2 0 D は、第 1 ~ 第 3 のプロジェクト内で定められた各制御周期の公約数となる制御周期で処理を実行する。第 1 ~ 第 3 のプロジェクト内で定められた各制御周期は、「 1 . 5 m s 」、「 1 . 5 m s 」および「 2 . 0 m s 」である。これらの周期の公約数は「 0 . 1 m s 」、「 0 . 2 5 m s 」、「 0 . 5 m s 」等である。ある局面において、統合制御ユニット 2 0 D は、第 1 ~ 第 3 のプロジェクト内で定められた各制御周期の最大公約数となる制御周期で処理を実行してもよい。

40

## 【 0 1 2 9 】

統合制御ユニット 2 0 D は、第 1 ~ 第 3 のプロジェクト内で定められた各制御周期の公約数(例えば、「 0 . 1 m s 」)で動作することにより、制御周期「 1 . 5 m s 」でグループ A 内の各ノードと通信し、制御周期「 1 . 5 m s 」でグループ B 内の各ノードと通信し、制御周期「 2 . 0 m s 」でグループ C 内の各ノードと通信することができる。

## 【 0 1 3 0 】

図 1 2 は、ネットワーク情報ファイル 6 1 5 に含まれる制御周期の一例を示す模式図である。各プロジェクト内のネットワーク情報ファイル 6 1 5 (ENIファイル等)は、ノード識別子以外にも制御周期 1 2 1 0 を項目として含む。統合プログラム 4 0 1 または統

50

合ツール 5 5 2 は、各プロジェクト内のネットワーク情報ファイルに含まれる制御周期 1 2 1 0 を取得する。ある局面において、統合プログラム 4 0 1 または統合ツール 5 5 2 は、各プロジェクト内の制御プログラムまたは他の任意のファイルから制御周期を取得してもよい。

#### 【 0 1 3 1 】

統合プログラム 4 0 1 または統合ツール 5 5 2 は、取得した複数の制御周期 1 2 1 0 から、当該複数の制御周期 1 2 1 0 の公約数となる制御周期を算出し、算出した制御周期を統合ネットワーク情報ファイル 6 4 5 に含める。

#### 【 0 1 3 2 】

ある局面において、統合プログラム 4 0 1 または統合ツール 5 5 2 は、各プロジェクト内のネットワーク情報ファイル 6 1 5 に基づいて、各プロジェクトの処理時間の占有率を算出してもよい。処理時間の占有率とは、仮想統合制御ユニット 6 4 0 の制御周期内における処理時間の占有率である。例えば、仮想統合制御ユニット 6 4 0 の制御周期が「1.0 ms」の場合に、第 1 の仮想制御ユニット 6 1 0 A の処理時間が「0.2 ms」のとき、第 1 の仮想制御ユニット 6 1 0 A の処理時間の占有率は「20% (0.2 ms / 1.0 ms \* 100 = 20)」になる。

#### 【 0 1 3 3 】

また、図 1 1 の例において、仮想統合制御ユニット 6 4 0 の制御周期内における第 1 の仮想制御ユニット 6 1 0 A の処理時間の占有率、第 2 の仮想制御ユニット 6 1 0 B の処理時間の占有率、および第 3 の仮想制御ユニット 6 1 0 C の処理時間の占有率の合計が、仮想統合制御ユニット 6 4 0 の制御周期を超える場合（処理時間の占有率の合計が 1 (100%) を超える場合）、仮想統合制御ユニット 6 4 0 (統合制御ユニット 2 0 D) は、第 1 ~ 第 3 のプロジェクトファイルの処理を実行できないことになる。この場合、統合プログラム 4 0 1 または統合ツール 5 5 2 は、各プロジェクトファイルを統合できないため、いずれかのインターフェイスを介して、エラーを出力し得る。

#### 【 0 1 3 4 】

< F . フローチャート >

次に、図 1 3 および図 1 4 を参照して、複数のプロジェクトファイルの統合処理と、統合後のプロジェクトファイルをインストールされた制御ユニット 2 0 の動作手順について説明する。

#### 【 0 1 3 5 】

図 1 3 は、複数のプロジェクトファイルの統合処理の手順の一例を示すフローチャートである。図 1 3 を参照して、サポート装置 6 (統合ツール 5 5 2) が、複数のプロジェクトファイルを統合する手順について説明する。なお、制御ユニット 2 0 (統合プログラム 4 0 1) も、以下に示す手順と同様の手順で複数のプロジェクトファイルを統合することができる。

#### 【 0 1 3 6 】

ある局面において、プロセッサ 9 1 は、図 1 3 の処理を行うためのプログラム (統合ツール 5 5 2) を二次記憶装置 9 5 から主記憶装置 9 2 に読み込んで、当該プログラムを実行してもよい。他の局面において、当該処理の一部または全部は、当該処理を実行するように構成された回路素子の組み合わせとしても実現され得る。

#### 【 0 1 3 7 】

ステップ S 1 3 1 0 において、プロセッサ 9 1 は、各制御ユニット 2 0 のプロジェクトファイルをビルドして、オブジェクトファイル (制御プログラムのオブジェクトファイル)、およびネットワーク情報ファイル 6 1 5 を生成する。

#### 【 0 1 3 8 】

ステップ S 1 3 2 0 において、プロセッサ 9 1 は、各制御ユニット 2 0 のオブジェクトファイルおよびネットワーク情報ファイル 6 1 5 を統合ツール 5 5 2 にインポートする。ある局面において、プロセッサ 9 1 は、各制御ユニット 2 0 からオブジェクトファイルおよびネットワーク情報ファイルして、当該取得したオブジェクトファイルおよびネットワ

10

20

30

40

50

ーク情報ファイル 6 1 5 を統合ツール 5 5 2 にインポートしてもよい。

【 0 1 3 9 】

ステップ S 1 3 3 0 において、プロセッサ 9 1 は、統合制御ユニット 2 0 D の仮想ネットワークドライバに仮想制御ユニット 6 1 0 を割り当て、さらに、各仮想制御ユニット 6 1 0 に装置識別子 1 0 1 0 を割り当てる。

【 0 1 4 0 】

ステップ S 1 3 4 0 において、プロセッサ 9 1 は、統合ツール 5 5 2 により、統合後のファイルをビルドして、統合ネットワーク情報ファイル 6 4 5 と、変換表 6 5 0 と、統合制御プログラムとを生成する。統合制御プログラムは、仮想統合制御ユニット 6 4 0 が実行する制御プログラムである。プロセッサ 9 1 は、統合ネットワーク情報ファイル 6 4 5 、変換表 6 5 0 、および統合制御プログラムを含む統合プロジェクトファイルと、統合前の各プロジェクトファイルとを統合制御ユニットに送信またはインストールする。

10

【 0 1 4 1 】

図 1 4 は、統合プロジェクトファイルをインストールされた制御ユニット 2 0 ( 統合制御ユニット ) の処理の手順の一例を示すフローチャートである。ある局面において、プロセッサ 3 1 は、図 1 4 の処理を行うためのプログラム ( 統合プログラム 4 0 1 ) を二次記憶装置 3 3 から主記憶装置 3 4 に読み込んで、当該プログラムを実行してもよい。他の局面において、当該処理の一部または全部は、当該処理を実行するように構成された回路素子の組み合わせとしても実現され得る。

【 0 1 4 2 】

ステップ S 1 4 1 0 において、仮想制御ユニット 6 1 0 の各々は、起動後に、スレーブ ( ノード ) との通信確立待ち状態になる。

20

【 0 1 4 3 】

ステップ S 1 4 2 0 において、仮想統合制御ユニット 6 4 0 は、ハードウェア ( 物理層 ) 6 7 0 の起動後に、統合ネットワーク情報ファイルに記載されている全スレーブを確認する。

【 0 1 4 4 】

ステップ S 1 4 3 0 において、仮想統合制御ユニット 6 4 0 は、統合後の全スレーブに、統合後のノード番号 ( 図 9 におけるノード識別子 9 1 0 に相当 ) を割り当てることで、全スレーブと通信可能状態になる。ある局面において、ステップ S 1 4 1 0 ~ S 1 4 3 0 までの処理は、並列的に実行されてもよいし、順番が入れ替わってもよい。

30

【 0 1 4 5 】

ステップ S 1 4 4 0 において、仮想統合制御ユニット 6 4 0 および仮想制御ユニット 6 1 0 の各々は、ステップ S 1 4 5 0 以降の処理を繰り返し実行する。

【 0 1 4 6 】

ステップ S 1 4 5 0 において、仮想制御ユニット 6 1 0 の各々は、スレーブとの通信可能状態を確認し、仮想ネットワークドライバ 6 2 0 , 6 3 0 を介して、各スレーブとの通信を開始する。ある局面において、仮想制御ユニット 6 1 0 の各々は、各スレーブに対して生存確認用のフレームを送信しその返信を受信することで、スレーブとの通信可能状態を確認してもよい。

40

【 0 1 4 7 】

ステップ S 1 4 6 0 において、仮想統合制御ユニット 6 4 0 は、仮想ネットワークドライバ 6 2 0 , 6 3 0 にセットされた ( 仮想制御ユニット 6 1 0 により送信された ) フレーム 7 0 0 内のノード識別子 7 0 1 を、実ネットワーク ( フィールドネットワーク 7 ) のノード識別子 7 1 1 に変更し、フレーム 7 1 0 の送受信を行う。

【 0 1 4 8 】

ステップ S 1 4 7 0 において、仮想統合制御ユニット 6 4 0 は、変換表 6 5 0 を用いて、実ネットワークから受信したフレーム 7 1 0 内のノード識別子 7 1 1 を、仮想制御ユニット 6 1 0 内のノード識別子 7 0 1 に変更する。そして、仮想統合制御ユニット 6 4 0 は、フレーム 7 0 0 を仮想ネットワークドライバ 6 2 0 , 6 3 0 にセットする ( フレーム 7

50

00を仮想制御ユニット610のいずれかに送信する)。

【0149】

ステップS1480において、仮想制御ユニット610の各々は、仮想ネットワークドライバ620, 630からのレスポンスを受信し、当該レスポンスを元に仮想制御ユニット610内の処理を実行する。

【0150】

ステップS1490において、仮想統合制御ユニット640および仮想制御ユニット610の各々は、停止要求等の入力を検知した場合は処理を終了する。そうでない場合、仮想統合制御ユニット640および仮想制御ユニット610の各々は、ステップS1450以降の処理を繰り返し実行する。ある局面において、ステップS1450～S1480までの処理は、仮想制御ユニット610および仮想統合制御ユニット640の各々により、並列的に実行されてもよいし、順番が入れ替わってもよい。

10

【0151】

以上説明した通り、本実施の形態に従う技術は、統合制御ユニット内で、複数の仮想制御ユニット610と、仮想統合制御ユニット640とを動作させ、仮想統合制御ユニット640にノード識別子の変換処理を実行させる。こうすることで、ユーザは、複数のプロジェクトファイルに変更を加えることなく、これらの複数のプロジェクトファイルを統合して使用し得る。

【0152】

さらに、本実施の形態に従う技術は、統合前の各プロジェクトファイル自体には変更を加えない。そのため、ユーザは、並列的に複数のプロジェクトファイルを作成して、当該複数のプロジェクトファイルを容易に統合することができる。また、統合後の制御プログラムに変更がある場合でも、ユーザは統合前のプロジェクトファイルの各々を個別に編集して再度統合処理を実行するだけでよく、プロジェクトファイルの保守性も向上する。

20

【0153】

< G . 付記 >

以上のように、本実施の形態は以下のような開示を含む。

(構成1)

第1の制御ユニット用(20A)の第1のプロジェクトファイルと、第2の制御ユニット(20B)用の第2のプロジェクトファイルと、上記第1および第2のプロジェクトファイルに基づいて生成される統合プロジェクトファイルとを格納する記憶部と、

30

各プロジェクトファイルに基づいて処理を実行する制御部(31)とを備え、

上記第1および第2のプロジェクトファイルの各々は、各プロジェクトファイルごとに独立して複数のノードの各々に割り当てられた複数のノード識別子を含み、

上記統合プロジェクトファイルは、全てのノードの各々に一意に割り当て直したノード識別子を含み、

上記制御部(31)は、上記全てのノードの各々に一意に割り当て直したノード識別子を用いて、通信するノードのノード識別子を特定する、制御システム(1, 20)。

(構成2)

上記記憶部は、第1の仮想制御ユニット(610)と、第2の仮想制御ユニット(610)と、仮想統合制御ユニット(640)とをさらに格納し、

40

上記制御部(31)は、

上記第1の仮想制御ユニット(610)に、上記第1のプロジェクトファイルをインストールし、

上記第2の仮想制御ユニット(610)に、上記第2のプロジェクトファイルをインストールし、

上記仮想統合制御ユニット(640)に、上記統合プロジェクトファイルをインストールし、

上記仮想統合制御ユニット(640)および上記ノードの各々間の通信データを上記第1の仮想制御ユニット(610)または上記第2の仮想制御ユニット(610)のいずれ

50

れかに転送する、構成 1 の制御システム ( 1 , 2 0 )。

( 構成 3 )

上記記憶部 ( 3 3 ) は、上記第 1 のプロジェクトファイルに含まれる第 1 のネットワーク情報ファイルおよび上記第 2 のプロジェクトファイルに含まれる第 2 のネットワーク情報ファイルと、上記統合プロジェクトファイルに含まれる統合ネットワーク情報ファイルとの変換情報をさらに格納し、

上記第 1 のネットワーク情報ファイル、上記第 2 のネットワーク情報ファイルおよび上記統合ネットワーク情報ファイルの各々は、各プロジェクト内で使用されるノード識別子を含み、

上記制御部 ( 3 1 ) は、上記変換情報を用いて、上記仮想統合制御ユニット ( 6 4 0 ) および上記ノードの各々間の通信データを上記第 1 の仮想制御ユニット ( 6 1 0 ) または上記第 2 の仮想制御ユニット ( 6 1 0 ) のいずれかに転送する、構成 2 の制御システム ( 1 , 2 0 )。

10

( 構成 4 )

上記制御部 ( 3 1 ) は、上記第 1 のネットワーク情報ファイルで定義されている第 1 の制御周期と、上記第 2 のネットワーク情報ファイルで定義されている第 2 の制御周期との公約数である制御周期に基づいて、上記ノードの各々と通信する、構成 3 の制御システム ( 1 , 2 0 )。

( 構成 5 )

複数のプロジェクトファイルを統合する装置 ( 6 ) をさらに含み、

20

上記装置 ( 6 ) は、上記第 1 のプロジェクトファイルと、上記第 2 のプロジェクトファイルとに基づいて、上記統合プロジェクトファイルを生成し、

上記制御部 ( 3 1 ) は、上記装置 ( 6 ) から、上記第 1 のプロジェクトファイルと、上記第 2 のプロジェクトファイルと、上記統合プロジェクトファイルとを取得する、構成 2 ~ 4 のいずれかの制御システム ( 1 , 2 0 )。

( 構成 6 )

上記装置 ( 6 ) は、上記制御システム ( 1 , 2 0 ) の制御周期内における、上記第 1 の仮想制御ユニット ( 6 1 0 ) の処理時間の占有率と、上記第 2 の仮想制御ユニット ( 6 1 0 ) の処理時間の占有率との合計が 1 0 0 % を超えることに基づいて、エラーを出力する、構成 5 の制御システム ( 1 , 2 0 )。

30

( 構成 7 )

上記制御部 ( 3 1 ) は、上記第 1 のプロジェクトファイルと、上記第 2 のプロジェクトファイルとに基づいて、上記統合プロジェクトファイルを生成する、構成 1 の制御システム ( 1 , 2 0 )。

( 構成 8 )

制御ユニット ( 2 0 ) を含む制御システム ( 1 , 2 0 ) の制御方法であって、

第 1 の制御ユニット ( 2 0 ) 用の第 1 のプロジェクトファイルと、第 2 の制御ユニット ( 2 0 ) 用の第 2 のプロジェクトファイルとに基づいて生成される統合プロジェクトファイルを参照するステップを含み、

上記第 1 および第 2 のプロジェクトファイルの各々は、各プロジェクトファイルごとに独立して複数のノードの各々に割り当てられた複数のノード識別子を含み、

40

上記統合プロジェクトファイルは、全てのノードの各々に一意に割り当て直したノード識別子を含み、

上記全てのノードの各々に一意に割り当て直したノード識別子を用いて、通信するノードのノード識別子を特定するステップをさらに含む、制御方法。

( 構成 9 )

第 1 の仮想制御ユニット ( 6 1 0 ) に、上記第 1 のプロジェクトファイルをインストールするステップと、

第 2 の仮想制御ユニット ( 6 1 0 ) に、上記第 2 のプロジェクトファイルをインストールするステップと、

50

仮想統合制御ユニット（ 6 4 0 ）に、上記統合プロジェクトファイルをインストールするステップと、

上記仮想統合制御ユニット（ 6 4 0 ）および上記ノードの各々間の通信データを上記第 1 の仮想制御ユニット（ 6 1 0 ）または上記第 2 の仮想制御ユニット（ 6 1 0 ）のいずれかに転送するステップとをさらに含む、構成 8 の制御方法。

（構成 1 0）

上記第 1 のプロジェクトファイルに含まれる第 1 のネットワーク情報ファイルおよび上記第 2 のプロジェクトファイルに含まれる第 2 のネットワーク情報ファイルと、上記統合プロジェクトファイルに含まれる統合ネットワーク情報ファイルとの変換情報にアクセスするステップをさらに含む、

10

上記第 1 のネットワーク情報ファイル、上記第 2 のネットワーク情報ファイルおよび上記統合ネットワーク情報ファイルの各々は、各プロジェクト内で使用されるノード識別子を含み、

上記変換情報を用いて、上記仮想統合制御ユニット（ 6 4 0 ）および上記ノードの各々間の通信データを上記第 1 の仮想制御ユニット（ 6 1 0 ）または上記第 2 の仮想制御ユニット（ 6 1 0 ）のいずれかに転送するステップをさらに含む、構成 9 の制御方法。

（構成 1 1）

上記第 1 のネットワーク情報ファイルで定義されている第 1 の制御周期と、上記第 2 のネットワーク情報ファイルで定義されている第 2 の制御周期との公約数である制御周期に基づいて、上記ノードの各々と通信するステップをさらに含む、構成 1 0 の制御方法。

20

（構成 1 2）

上記制御システム（ 1 , 2 0 ）の制御周期内における、上記第 1 の仮想制御ユニット（ 6 1 0 ）の処理時間の占有率と、上記第 2 の仮想制御ユニット（ 6 1 0 ）の処理時間の占有率との合計が 1 0 0 % を超えることに基づいて、エラーを出力するステップをさらに含む、構成 1 1 の制御方法。

【 0 1 5 4 】

今回開示された実施の形態は全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。本開示の範囲は上記した説明ではなくて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及び範囲内で全ての変更が含まれることが意図される。また、実施の形態および各変形例において説明された開示内容は、可能な限り、単独でも、組合わせても、実施することが意図される。

30

【符号の説明】

【 0 1 5 5 】

1 制御システム、2 サーバ装置、3 表示装置、4 ゲートウェイ、5 ネットワーク、6 サポート装置、7 フィールドネットワーク、8 制御対象、10 ネットワークシステム、20 制御ユニット、20A 第1の制御ユニット、20B 第2の制御ユニット、20C 第3の制御ユニット、20D 統合制御ユニット、21 セキュリティユニット、22 セーフティユニット、23 機能ユニット、24 電源ユニット、31, 91 プロセッサ、32 チップセット、33, 95 二次記憶装置、34, 92 主記憶装置、35 通信コントローラ、36 インジケータ、37 スイッチインターフェイス、38

40

ディップスイッチ、39 内部バスコントローラ、40, 41, 42 ネットワークコントローラ、43 メモリカードインターフェイス、44 USBコントローラ、45 システムプログラム、46 サービスプログラム、48 揮発性記憶領域、49 暗号化済み制御プログラム、50 揮発性記憶領域、51 復号済み制御プログラム、52 バス、53 メモリカード、93 入力部、94 出力部、96 光学ドライブ、97 通信インターフェイス、98 プロセッサバス、101, 201 ノード、401 統合プログラム、550 記録媒体、551 制御プログラム、552 統合ツール、553 サポートプログラム、610 仮想制御ユニット、610A 第1の仮想制御ユニット、610B 第2の仮想制御ユニット、610C 第3の仮想制御ユニット、615 ネットワーク情報ファイル、615A 第1のネットワーク情報ファイル、615B 第2のネットワーク情報ファイ

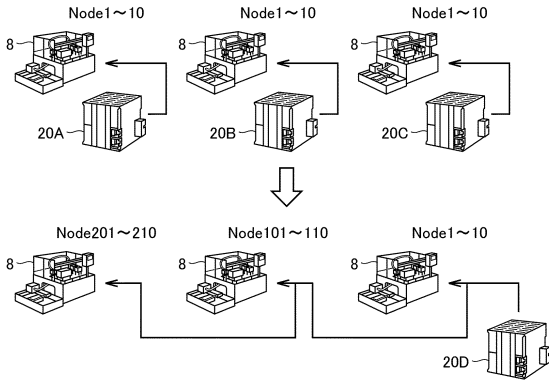
50

ル、615C 第3のネットワーク情報ファイル、620, 630 仮想ネットワークドライバ、640 仮想統合制御ユニット、645 統合ネットワーク情報ファイル、650 変換表、660 ネットワークドライバ、670 ハードウェア、700, 710 フレーム、701, 711, 910 ノード識別子、1010 装置識別子、1020 第1のノード識別子、1030 第2のノード識別子、1210 制御周期。

【図面】

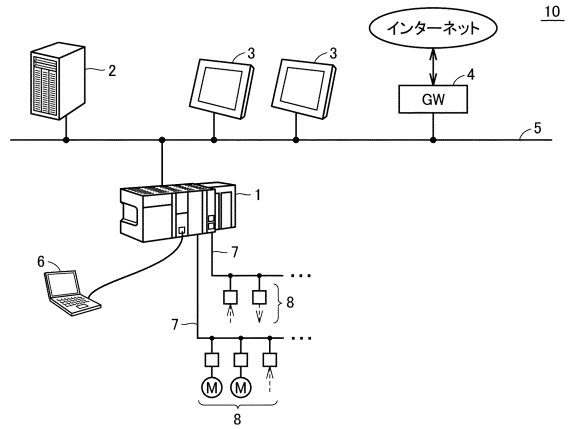
【図1】

図1



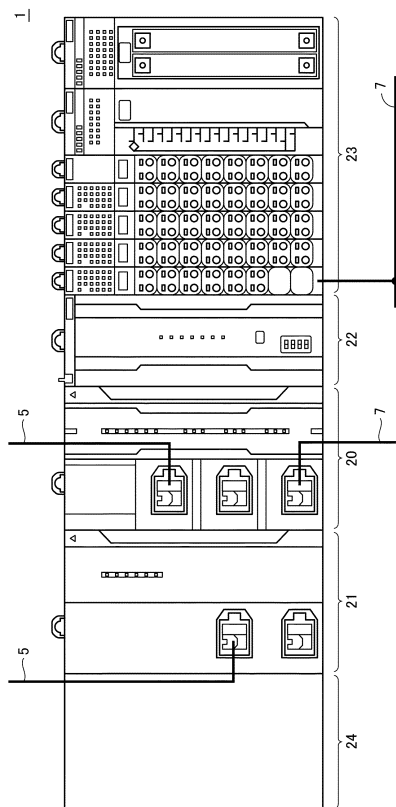
【図2】

図2



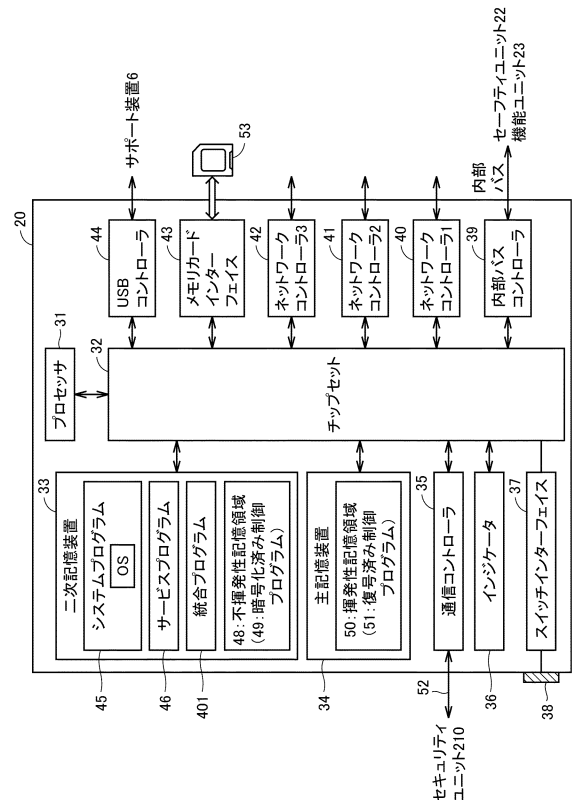
【図3】

図3



【図4】

図4



10

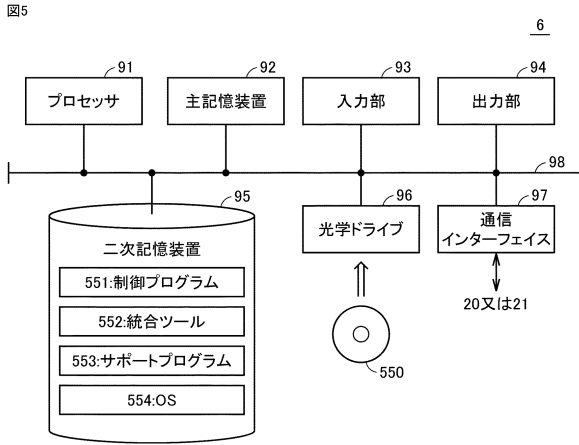
20

30

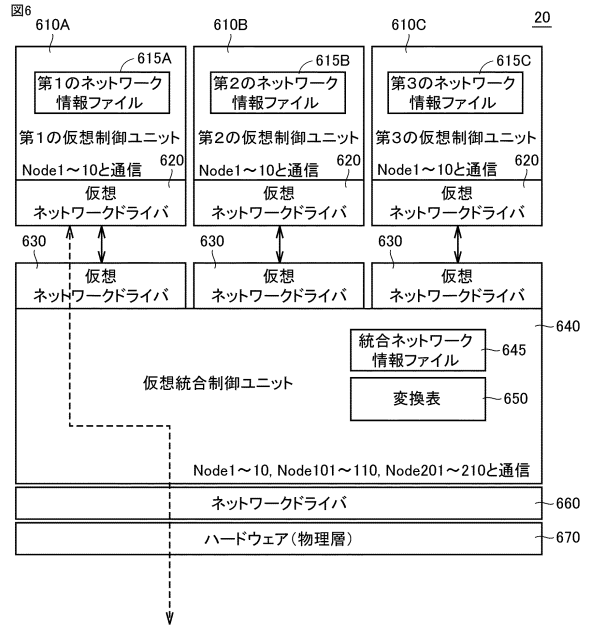
40

50

【図5】



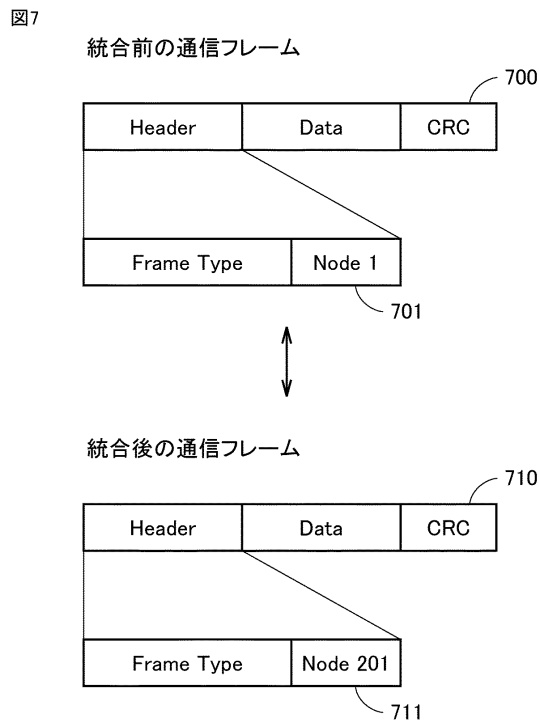
【図6】



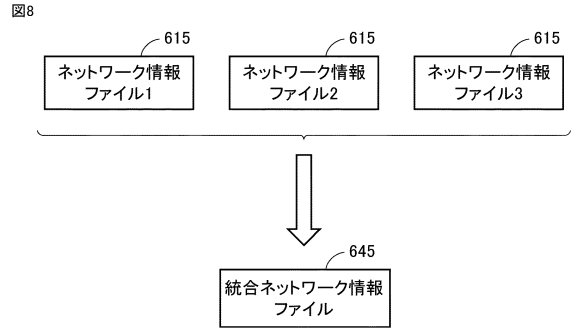
10

20

【図7】



【図8】



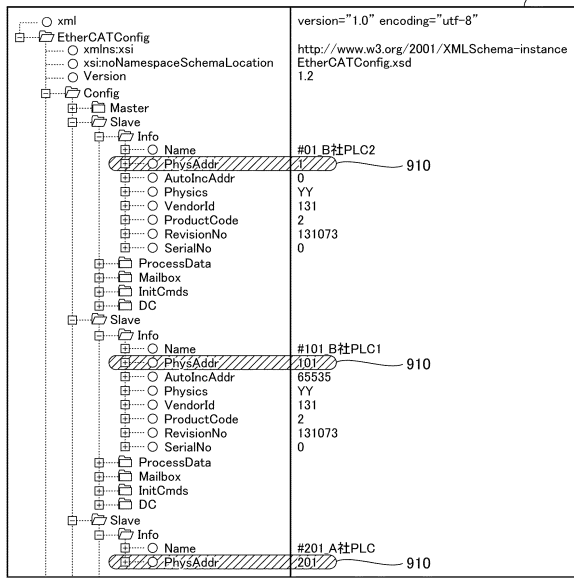
30

40

50

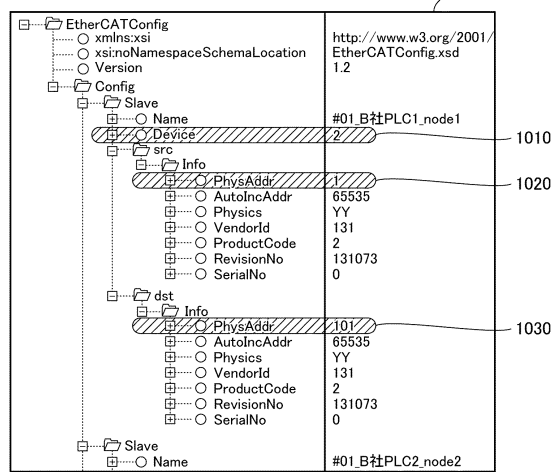
【 図 9 】

図9



【 図 10 】

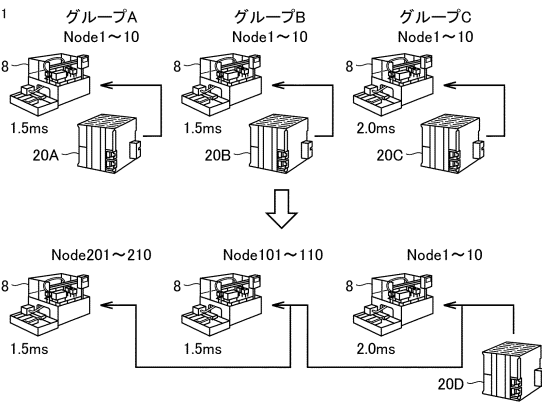
図10



10

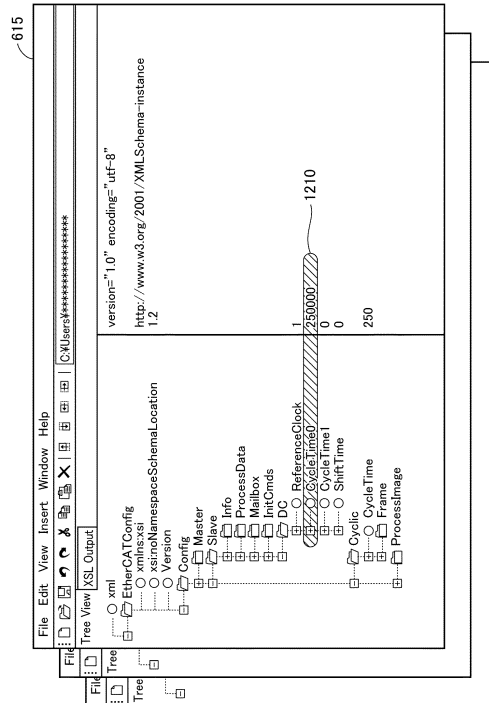
【 図 11 】

図11



【 図 12 】

図12



各制御ユニットが必要とする周期の公約数を算出し、  
統合後のサイクルタイムを公約数に設定

20

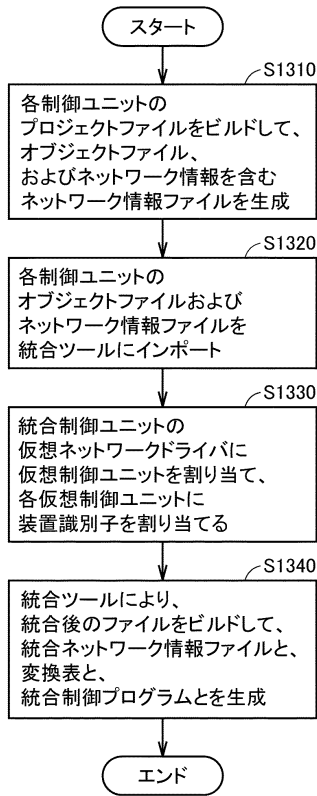
30

40

50

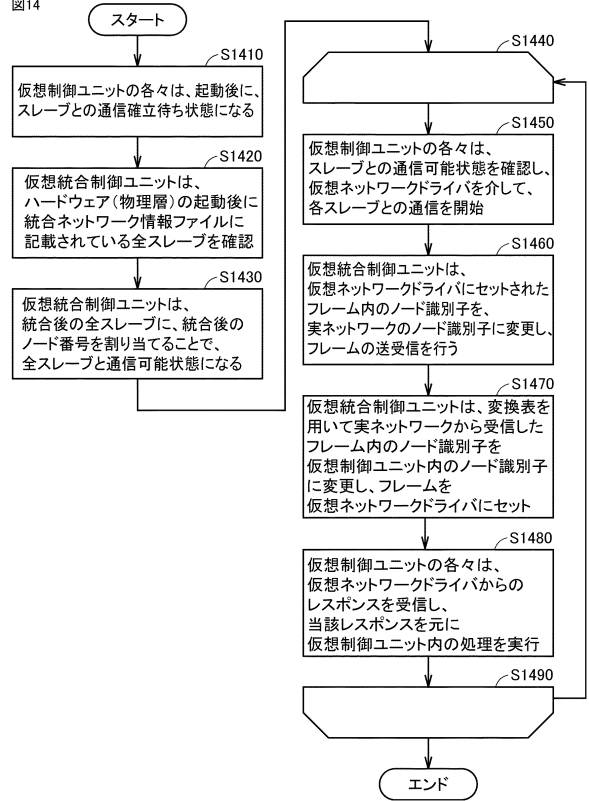
【 図 1 3 】

図13



【 図 1 4 】

図14



10

20

30

40

50

## フロントページの続き

審査官 岩 崎 優

- (56)参考文献 特開2000-341357(JP,A)  
特開2014-010811(JP,A)  
特開2014-206830(JP,A)  
特開平09-024403(JP,A)  
特開2019-159631(JP,A)  
特開2008-159027(JP,A)
- (58)調査した分野 (Int.Cl., DB名)  
G05B 19/04 - 19/05