

(19) 日本国特許庁(JP)

(12) 公表特許公報(A)

(11) 特許出願公表番号

特表2020-523689

(P2020-523689A)

(43) 公表日 令和2年8月6日(2020.8.6)

(51) Int.Cl. F I テーマコード (参考)
G 0 5 B 19/05 (2006.01) G O 5 B 19/05 A 5 H 2 2 O

審査請求 未請求 予備審査請求 有 (全 27 頁)

(21) 出願番号	特願2019-568070 (P2019-568070)	(71) 出願人	518416126 ブルックス・オートメーション・(ジャー マニー)・ゲーエムベーハー Brooks Automation (Germany) GmbH ドイツ連邦共和国、78315 ラードル フツェル、イム・ビーゼングルント 17, 78315 Radolfzell, Germany
(86) (22) 出願日	平成30年6月6日 (2018.6.6)	(74) 代理人	100098394 弁理士 山川 茂樹
(85) 翻訳文提出日	令和1年12月18日 (2019.12.18)	(74) 代理人	100064621 弁理士 山川 政樹
(86) 国際出願番号	PCT/EP2018/064936		
(87) 国際公開番号	W02018/224568		
(87) 国際公開日	平成30年12月13日 (2018.12.13)		
(31) 優先権主張番号	17174804.9		
(32) 優先日	平成29年6月7日 (2017.6.7)		
(33) 優先権主張国・地域又は機関	欧州特許庁 (EP)		

最終頁に続く

(54) 【発明の名称】 PCプログラムを用いてPLCを制御する方法

(57) 【要約】

本発明は、PCプログラム(220)を用いてPLCを制御する方法に関し、PLCプログラムのソースコードは、データセクション(100)を含み、データセクション(100)は、PLCプログラムの実行時にPLCによって公開されるサービスを示すデータを含み、データセクション(100)に含まれるデータおよびPCプログラムテンプレート(200)を用いて、PCプログラムのソースコード(210)を自動生成し、PCプログラムの自動生成されたソースコード(210)から生成されたPCプログラム(220)を用いて、PLCを制御する。

【選択図】図3

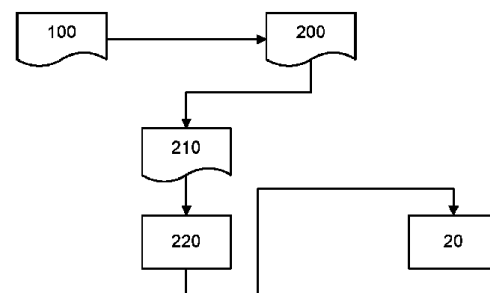


Fig. 3

【特許請求の範囲】**【請求項 1】**

PCプログラム(220)を用いてPLC(30)を制御する方法であって、
PLCプログラムのソースコード(310)は、データセクション(100)を含み、
前記データセクション(100)は、前記PLCプログラムの実行時に前記PLC(30)によって公開されるサービスを示すデータを含み、
前記データセクション(100)に含まれる前記データおよびPCプログラムテンプレート(200)を用いて、前記PCプログラムのソースコード(210)を自動生成し、
前記PCプログラムの前記自動生成されたソースコード(210)から生成された前記PCプログラム(220)を用いて、前記PLC(30)を制御する、方法。

10

【請求項 2】

前記PCプログラムテンプレート(200)は、テキストテンプレートである、請求項1に記載の方法。

【請求項 3】

前記データセクション(100)は、前記PLCプログラムの前記ソースコード(310)のヘッダである、請求項1または2に記載の方法。

【請求項 4】

前記PLCプログラムの前記ソースコード(310)は、高水準プログラミング言語である、請求項1～3のいずれか一項に記載の方法。

【請求項 5】

前記PCプログラム(220)は、PC(20)上で実行されると、グラフィカルユーザインタフェースを表示するように構成される、請求項1～4のいずれか一項に記載の方法。

20

【請求項 6】

前記PCプログラム(220)を用いて前記PLC(30)を制御することは、前記PLCに値を書き込むこと、および/または前記PLCから値を読み取ることを含む、請求項1～5のいずれか一項に記載の方法。

【請求項 7】

前記PCプログラム(220)を用いて前記PLC(30)を制御することは、前記PLCから情報を受信すること、および/または前記PLCにコマンドを送信することを含む、請求項1～6のいずれか一項に記載の方法。

30

【請求項 8】

前記PLC(30)によって公開される前記サービスは、前記PLCにデータを書き込むための少なくとも1つのサービス、および/または前記PLCからデータを読み取るための少なくとも1つのサービスを含む、請求項1～7のいずれか一項に記載の方法。

【請求項 9】

前記サービスは、前記PLC(30)上で動作するサーバを用いて、前記PLC(30)によって公開される、請求項1～8のいずれか一項に記載の方法。

【請求項 10】

前記PLC(30)は、前記PLCプログラムを実行している、請求項1～9のいずれか一項に記載の方法。

40

【請求項 11】

PLC(30)を制御するためのPCプログラム(220)を自動的に作成する方法であって、

PLCプログラムのソースコード(310)のデータセクション(100)に含まれるデータおよびPCプログラムテンプレート(200)を用いて、前記PCプログラムのソースコード(210)を自動生成し、

前記データセクションは、前記PLCプログラムの実行時に前記PLC(30)によって公開されるサービスを示すデータを含む、方法。

【請求項 12】

50

請求項 1 ~ 11 のいずれか一項に記載の方法を実行するように、特にプログラムの、
適応させたコンピュータ。

【請求項 13】

プログラムコードを有するコンピュータプログラムであって、コンピュータ上で実行されることで、請求項 1 ~ 11 のいずれか一項に記載の方法を前記コンピュータに実行させるための、コンピュータプログラム。

【請求項 14】

請求項 13 に記載のコンピュータプログラムを格納している、マシン可読記憶媒体。

【発明の詳細な説明】

【技術分野】

10

【0001】

本発明は、PC プログラムを用いて PLC を制御する方法、ならびに本方法を実行するためのコンピュータおよびコンピュータプログラムに関するものである。

【背景技術】

【0002】

PLC (プログラマブルロジックコントローラ) は、プロセスプラントまたは機械類を制御するための装置であり、デジタル的にプログラムされる。PLC は、通常、入力と、出力と、ファームウェア (オペレーティングシステム) と、アプリケーションプログラムをダウンロードするためのインタフェースと、を有する。アプリケーションプログラムは、入力に応じて、いかに出力が設定されるのかを定義する。ファームウェアによって、アプリケーションプログラムの入力および出力が利用可能となる。

20

【0003】

リアルタイムシステムとしての PLC は、デジタルセンサまたはアナログセンサと接続しており、プロセスプラントまたは機械類の空気圧シリンダもしくは油圧シリンダまたは電気モータのようなアクチュエータを制御する。特に半導体製造装置アプリケーションである多くの機械制御アプリケーションとして、典型的には、PLC および工場内ネットワーク (例えば、ファブまたはファウンドリとしても知られる半導体製造工場のホストシステム) に接続されたパーソナルコンピュータ (PC) アプリケーションがあり、それは、これらのセンサ値を PLC から読み取って、これらの取得したリアルタイムデータについて高度な管理を実行し (例えば、これらのデータを状態変数としてファブホストに送信し)、個々のアクションを実行するためのコマンドを PLC に送信する。

30

【0004】

従来の PC アプリケーションにおける問題は、例えば、新たなセンサの追加または新たなアクチュエータの追加など、PLC アプリケーション側に進展があるたびに、PC アプリケーション側でも、これらの新たに追加された要素にアクセスするための基本機能を実装するために、例えば以下のような、かなりの量の修正作業があることである。

- ・ PC がアクセスできる新たに追加されたセンサ / アクチュエータに関する PLC インタフェースのドキュメンテーションを作成 / 修正する。

- ・ 新たに追加された要素のための、PLC アプリケーションと PC アプリケーションとの間の追加の通信リンクを追加する。

40

- ・ 新たに追加された要素を表現する PC プログラムにおいて内部変数 / 手続きを作成する。

- ・ 新たに追加されたセンサを表示するため、および新たに追加されたアクチュエータを制御するために PLC にコマンドを送信する可能性を提供するために、PC プログラムのグラフィカルユーザインタフェースを作成 / 修正する。

【0005】

例えば OPC サーバを用いて、読み取りおよび / または書き込みのために、PLC の内部変数を PC に公開することができる。ただし、その場合にも、新たに提供される変数を用いるために、PC プログラムを手動で適応させなければならない。

【発明の概要】

50

【発明が解決しようとする課題】**【0006】**

よって、PCプログラムをPLCアプリケーション側の進展に適応させることに関わる必要な作業を軽減することが望ましい。

【課題を解決するための手段】**【0007】**

本発明によれば、独立請求項の特徴により、PCプログラムを用いてPLCを制御するための方法、コンピュータプログラム、およびコンピュータについて提案する。効果的なさらなる改良は、従属請求項の主題および以下の説明の主題をなしている。

【0008】

本発明により、PLCプログラムのソースコードのデータセクションおよびPCプログラムテンプレートから読み出されたデータに基づいて、PLCと通信および/またはPLCを制御するために、好ましくはグラフィカルユーザインタフェース（GUI）を有する、PCプログラムを自動的に生成するための方法およびその実装を導入する。すなわち、そのデータセクションは、PCプログラムとPLCプログラムの両方に共通のソースコード部分となる。従って、PCプログラムをPLCアプリケーション側の進展に手動で適応させることによって導入されるエラーを防ぐことができる。PLCプログラム内では、データセクションは、PLCプログラムの実行時にPLCによって公開されるサービスをそれぞれ指定または宣言するデータを含み、それに対応するサービス公開手続きによって用いることができる。PCプログラム内では、これらのデータを、実際に公開されているサービス

10

20

【0009】

本発明は、PLCプログラム側で進展または変更があって、これらの変更をPCでも実装する必要がある場合に、PCプログラム側での開発時間を節約することを目的とする。PLCプログラム側、すなわちPLCプログラムのソースコードを適応させる必要は依然としてあるものの、PCプログラムが何らかのさらなる開発を必要とすることはない。PLCプログラムのソースコードが定義または変更された場合に、新たなPCプログラムソースコードを作成するために、変更されたデータセクションをPCプログラムテンプレートに結合する必要があるのみである。これにより、PC側での開発時間が大幅に節約されるとともに、ヒューマンエラーが回避され、さらに、自動的な機能整合性チェックが行われる。

30

【0010】

本発明は、効果的には、「監視制御とデータ取得」（SCADA: Supervisory Control And Data Acquisition）の枠内で用いることができる。SCADAは、高度なプロセス監視管理のために、PC、ネットワーク化されたデータ通信、およびグラフィカルユーザインタフェースを用いる制御システムアーキテクチャであるが、プロセスプラントまたは機械類にインタフェースするために、PLCおよび分散PIDコントローラのような他の周辺装置を使用する。監視、およびコントローラ設定点変更のようなプロセスコマンドの発行を可能とするオペレータインタフェースは、SCADA監視コンピュータシステムを介して処理される。一方、リアルタイム制御ロジックまたはコントローラ計算は、フィールドセンサおよびアクチュエータに接続するネットワーク化されたモジュールによって実行される。

40

【0011】

好ましい実施形態によれば、PLCプログラムのソースコードは、高水準プログラミング言語であり、特にC系言語またはIEC 61131-3で定義される言語である。IEC 61131-3は、プログラマブルロジックコントローラ用のオープン国際標準規格IEC 61131の（10部のうちの）第3部である。IEC 61131の第3部は、PLC内の制御プログラムの基本的なソフトウェアアーキテクチャおよびプログラミング

50

グ言語を扱う。それは、２種類の図式プログラミング言語規格および２種類のテキストプログラミング言語規格、すなわち、ラダーダイアグラム（ＬＤ）図式言語；ファンクションブロックダイアグラム（ＦＢＤ）図式言語；ストラクチャードテキスト（ＳＴ）テキスト言語；インストラクションリスト（ＩＬ）テキスト言語、を定義している。

【００１２】

ＰＬＣプログラムの作成は、ＰＣ上でＰＬＣプログラミングツールキットを用いて行うと非常に簡便である。ＰＬＣプログラミングツールキットは、ＰＣプログラムのプログラミング用のプログラミングツールキットとは異なる可能性があるが、本発明の利点は、ＰＬＣプログラミングとＰＣプログラミングとの間の言語の壁をいずれも乗り越えるために共通のファイルを共有することである。ＰＬＣプログラムでは、このファイルは、ソースコードの一部であり、ＰＣプログラムでは、このファイルは、プログラムを自動的に生成するための入力である。本発明により、ＰＣプログラムとＰＬＣプログラムの両方に関連したソースコードを一度のみ提供することによって、コード生成が改善および簡略化される。本発明により、この場合は本発明の意味においてデータセクションである少なくとも１つのモジュールが一度のみ提供されて、それが二度、すなわち両方のプログラムを作成するときに使用されるような、ＰＣプログラムとＰＬＣプログラムの両方のソースコードのモジュール化について提案する。

10

【００１３】

効果的には、そのデータセクションを、別個のファイルの形で提供することができる。これにより、そのデータセクションを、非常に簡単に、ＰＬＣプログラムソースコードおよびＰＣプログラムソースコードに関連付けて利用する。そのデータセクションは、好ましくは、ＰＬＣプログラムのソースコードのヘッダであり、より好ましくは、別個のヘッダファイルの形で提供される。

20

【００１４】

好ましくは、ＰＣプログラムテンプレートは、テキストテンプレートである。従って、同じくテキストである、ソースコードからのデータセクションに関連付けて、ＰＣプログラムのソースコードを非常に簡単に自動生成することができる。特に、Microsoft社が提供するいわゆる「テキストテンプレート変換ツールキット」（Ｔ４）のような、いずれかの周知のコード生成ツールを効果的に使用することができる。開発者は、様々なパラメータを用いてテキストファイルの作成を自動化するためのアプリケーションフレームワークまたはツールフレームワークの一部として、Ｔ４を使用することができる。これらのテキストファイルは、最終的に、コード（例えば、（Visual）Ｃ＃のようなＣ系、もしくはVisual BasicのようなＢＡＳＩＣ系）、XML、HTML、またはXAMLのような、任意のテキスト形式で提供することができる。

30

【００１５】

ＰＣプログラムがＰＬＣと十分に通信可能とするためには、ＰＣプログラムを用いてＰＬＣを制御することは、好ましくは、ＰＬＣへの値の書き込みおよび／またはＰＬＣからの値の読み取りを含む。典型的なＰＬＣでは、ランタイム中に、最終的にＰＬＣの動作を定義する特定の値を有する多くの異なる変数が、メモリに保存される。従って、そのような値を読み取ること、および書き込むことは、ＰＬＣを制御するための基本的な方法である。さらに効果的には、ＰＣプログラムを用いてＰＬＣを制御することは、ＰＬＣから情報を受信すること、および／またはＰＬＣにコマンドを送信すること、を含む。

40

【００１６】

ＰＣプログラムによりＰＬＣを制御することを可能とするためには、ＰＬＣによって公開されるサービスは、好ましくは、ＰＬＣにデータを書き込むための少なくとも１つのサービス、および／またはＰＬＣからデータを読み取るための少なくとも１つのサービス、を含む。これらのサービスにより、効果的に、ＰＣプログラムが、ＰＬＣに値を書き込むこと、および／またはＰＬＣから値を読み取ること、ひいてはＰＬＣを制御することが可能となる。

【００１７】

50

本発明の好ましい実施形態によれば、それらのサービスは、例えば、（特にOPCUAによる）OPC、PVI、ProfiNetなどのサーバである、PLC上で動作するサーバを用いて、PLCによって公開される。これらは、特に、サーバから値（特に測定値）を読み取るため、および/またはサーバすなわちPLCに値（特に設定値）を書き込むための、様々なサービスをクライアントに公開するための、十分に試行されたソリューションである。さらに、公開されたサービスに接続するために、例えば、B & RからのいわゆるPVI（Process Visualization Interface）など、サードパーティのプロシージャおよびクライアントを効果的に使用することができる。PVIは、PC上で動作可能であって、データを受信するためにPLCに接続するソフトウェアである。PCプログラムは、PLCに接続するために、PVIを補助ツールとして用いることができる。

10

【0018】

プログラムコードを含む、本発明によるコンピュータプログラムは、このコンピュータプログラムがコンピュータ上で実行されることで、本発明による方法のうちの1つのすべてのステップをコンピュータに実行させる。そのコンピュータプログラムを提供するための適切な非一時的マシン可読媒体は、特に、フロッピーディスク、ハードディスク、フラッシュメモリ、EEPROM、CD-ROM、およびDVDなどである。コンピュータネットワーク（インターネット、イントラネットなど）上のプログラムをダウンロードすることが効果的である。

20

【0019】

本発明のさらなる効果および実施形態は、説明および添付の図面から明らかになるであろう。

【0020】

前述の特徴および以下でさらに記載する特徴は、個々に示す組み合わせだけではなく、さらなる組み合わせまたは単独でも、本発明の範囲から逸脱することなく、用いることができることに留意すべきである。

【図面の簡単な説明】

【0021】

【図1】図1は、プロセスチェーンと、いくつかのセンサ、およびPLCによって制御されるアクチュエータと、を備える例示的な機械を示している。

30

【図2】図2は、本発明の好ましい実施形態によるデータセクションを概略的に示している。

【図3】図3は、本発明の好ましい実施形態による、ヘッダファイルおよびPCプログラムテンプレートを用いてPCプログラムを作成するプロセスを概略的に示している。

【図4】図4は、本発明の好ましい実施形態による、ヘッダファイルおよびPLCメインプログラムファイルを用いてPLCプログラムを作成するプロセスを概略的に示している。

【発明を実施するための形態】

【0022】

図1は、製造プロセスで使用する基板キャリア（例えば、半導体ウェハキャリア、フラットパネルディスプレイキャリア、レチクルキャリア、および電子装置製造用の他のキャリア）のための自動洗浄・検査システムの一部であり得る機械10を概略的に示している。機械10は、ドア12と、この場合はドアを開閉するための空気圧シリンダとして構成されたアクチュエータ13と、を有するプロセスチェーン11を備える。ドア12が開いているのか閉じているのかを判断するために、アクチュエータの位置を検出するための2つのセンサ14が設けられている。

40

【0023】

プロセスチェーン11内には、チェーンを所望の温度まで加熱するためのヒータ15と、電気モータ17によって所望の速度で回転させる回転板と、が設けられている。

【0024】

50

センサ 14 およびアクチュエータ 13、15、17 は、プロセスを制御するための PLC 30 に接続されている。そして PLC 30 は、PLC を制御するための PC 20 に接続されており、例えば、PC は、PLC にコマンドを送信するとともに、PLC からデータを取得する。

【0025】

図 2 は、様々なサービスを示すための内部ブロックを有する、PLC プログラムのデータセクション 100 を概略的に示している。好ましくは、このデータセクションは、PLC プログラムのヘッダである。従って、そのデータを、信頼性の高い直接的な方法で読み出すことが可能である。

【0026】

前述のように、PLC によって公開されるサービスは、PLC にデータを書き込むための少なくとも 1 つのサービス、および / または PLC からデータを読み取るための少なくとも 1 つのサービス、を含むことができる。従って、実際には、このデータセクションは、PLC にデータを書き込むためのサービス (「CONTROL」) としての 1 つのセクションまたはブロック 110 と、PLC からデータを読み取るためのサービス (「INFO」) としての 1 つのブロック 120 と、を含む。

【0027】

同様に、ブロック 110 は、コマンドとパラメータにそれぞれ関連した 2 つのサブブロック 111、112 に分割することができる。例えば、サブブロック 111 では、装置のアクション (例えば、ドアの開閉、プロセスの開始 / 停止) を開始させるための、PLC への可能なコマンドすなわち PC からの到来データを示すことができ、サブブロック 112 では、機械に関連したパラメータを設定するための、可能な書き込み可能パラメータすなわち PC からの到来データを示すことができる。

【0028】

基本的に、2 つのタイプのパラメータがあり、装置関連パラメータの場合、特定の装置構成部品がそのコマンドに関連しており (例えば、「Rotate Table」コマンドの場合の所望速度)、プロセス関連パラメータの場合、何らかのプロセスの制約がそのコマンドに関連している (例えば、「Run Process」コマンドの場合の所望温度)。

【0029】

ブロック 120 は、プロセス結果と装置状態にそれぞれ関連した 2 つのサブブロック 121、122 に分割することができる。例えば、サブブロック 121 では、プロセスの結果 (例えば、「Run Process」コマンドの場合の測定された平均 / 最低 / 最高温度、および「Desired Temperature」パラメータによる結果の評価) を報告するための、可能な読み取り可能プロセス結果すなわち PC に送信される発信データを示すことができ、サブブロック 122 では、装置状態 (例えば、ドアの開閉状態) および (何らかの障害が生じている場合の) 装置障害状態を報告するための、可能な読み取り可能装置状態データすなわち PC に送信される発信データを示すことができる。

【0030】

図 1 に示す例では、

- ・ PC は、チェンバの OPEN または CLOSE (2 つのコマンド) を PLC に要求できる。
- ・ PC は、ドアが OPEN または CLOSED (2 つの装置状態) のどちらかである状態を通知される。
- ・ PC は、所望の温度および電気モータ速度 (2 つのコマンドパラメータ) でプロセスを開始させること (1 つのコマンド) を PLC に要求できる。
- ・ プロセスが終了すると、PC は、プロセス中の最低 / 最高 / 平均温度 (1 つのプロセス結果) を通知される。

【0031】

本例に関連した好ましいヘッダを、付録に示している。このヘッダファイルは、所望の

10

20

30

40

50

装置機能に応じて、手動で設計されている。これは、P C プログラムに対して、インタフェースを提供するとともに、制御対象の P L C 装置のすべての能力を完全に規定している。

【 0 0 3 2 】

一方で、このヘッダファイルは、図 3 に示すように P C プログラムを作成するために、使用される。他方で、このヘッダファイルは、図 4 に示すように P L C プログラムを作成するためにも、使用される。

【 0 0 3 3 】

次に図 3 を参照して、P C プログラムソースコード 2 1 0 を生成するために、P C プログラムテンプレート 2 0 0 と共にヘッダファイル 1 0 0 を使用する。P L C 3 0 を制御するために P C 2 0 上で実行可能な P C プログラム 2 2 0 を作成するために、P C プログラムソースコード 2 1 0 をコンパイルすることができる。好ましくは、P C プログラム 2 2 0 は、P C 2 0 上で実行されると、グラフィカルユーザインタフェースを表示するように構成される。

10

【 0 0 3 4 】

次に図 4 を参照して、P L C プログラムソースコード 3 1 0 を生成するために、対応する P L C プログラムソースコード 3 0 0 と共にヘッダファイル 1 0 0 を同じく使用する。P L C プログラム 3 2 0 を作成するために、P L C プログラムソースコード 3 1 0 をコンパイルすることができ、P L C プログラム 3 2 0 は、P L C 3 0 に転送されて、実行されることで、プロセスを制御すると同時に、ヘッダファイル 1 0 0 で宣言されているサービスを、P C プログラムにより接続されるように公開する。

20

【 0 0 3 5 】

P C プログラムを用いて、ユーザは、

- ・チェンバの O P E N または C L O S E を P L C に要求できる。
- ・ドアが O P E N または C L O S E D のどちらかである状態を通知される。
- ・所望の温度および電気モータ速度でプロセスを開始させることを P L C に要求できる。
- ・プロセスが終了した後に、プロセス中の最低 / 最高 / 平均温度を通知される。

【 0 0 3 6 】

なお、上記説明は、開示の実施形態の態様の例示的なものにすぎないことは理解されるべきである。当業者であれば、開示の実施形態の態様から逸脱することなく、種々の代替案および変形例を考案することができる。よって、開示の実施形態の態様は、添付の請求項の範囲内にあるすべてのかかる代替案、変更、および変形を包含するものである。さらに、様々に異なる特徴が、互いに異なる従属請求項または独立請求項に記載されているということだけで、それらの特徴の組み合わせを効果的に用いることができないことを示すものではなく、かかる組み合わせは、本発明の態様の範囲から逸脱していない。

30

【 0 0 3 7 】

例示的なヘッダファイル：

```
typedef struct
```

```
{
```

```
CONTROL_INTERFACE tControl; // control of slave (PLC) from master (PC)
```

40

```
INFO_INTERFACE tInfo; // info back from slave (PLC) to master (PC)
```

```
} PLC_INTERFACE;
```

```
typedef struct
```

```
{
```

```
EQUIPMENT_STATUS tStatus;
```

```
PROCESS_RESULT tResult;
```

```
} INFO_INTERFACE;
```

```
typedef struct
```

50


```

{
  BOOL bProcessChamberClosed; // cylinder position sensor (Digital Input)
  BOOL bProcessChamberOpen; // cylinder position sensor (Digital Input)
  INT16 iMotorPosition; // motor position (Analog Input)
} EQUIPMENT_STATUS;

typedef struct
{
  REAL rMinTemperatureDuringProcess; // min temperature during process
  REAL rMaxTemperatureDuringProcess; // max temp during process
  REAL rAverageTemperatureDuringProcess; // average temp during process
} PROCESS_RESULT;

typedef struct
{
  COMMAND tCommand;
  PARAMETERS tParameters;
} CONTROL_INTERFACE;

typedef struct
{
  BOOL bOpenProcessChamber; // command to open the process chamber
  BOOL bCloseProcessChamber; // command to close the process chamber
  BOOL bRunProcess; // command to start a process
} COMMAND;

typedef struct
{
  REAL rTargetProcessTemperature; // target temperature during process
  REAL rTargetMotorSpeed; // target motor speed during process
} PARAMETERS
【 0 0 3 8 】
  P C プログラムテンプレート 2 0 0 によって生成された P C プログラムソースコード 2
  1 0 の例示的な部分 :
public class PLC_INTERFACE : PlcMember
{
  public override void SetValue(string memberName, dynamic value)
  {
    switch(memberName)
    {
      case "tControl":
        tControl = InternalSetValue(memberName, value, tControl);
        break;
      case "tInfo":
        tInfo = InternalSetValue(memberName, value, tInfo);
        break;
    }
  }
}
public override dynamic GetValue(string memberName)
{

```

```

switch(memberName)
{
    case "tControl":
        return tControl;
    case "tInfo":
        return tInfo;
}
return null;
}
public override T GetValue<T>(string memberName) 10
{
    switch(memberName)
    {
        case "tControl":
            return (T)(object)tControl;
        case "tInfo":
            return (T)(object)tInfo;
    }
    return default(T);
} 20
public CONTROL_INTERFACE tControl = new CONTROL_INTERFACE(); // control of slave (PLC) from master (PC)
public INFO_INTERFACE tInfo = new INFO_INTERFACE(); // info back from slave (PLC) to master (PC)
}

public class INFO_INTERFACE : PlcMember
{
    public override void SetValue(string memberName, dynamic value) 30
    {
        switch(memberName)
        {
            case "tStatus":
                tStatus = InternalSetValue(memberName, value, tStatus);
                break;
            case "tResult":
                tResult = InternalSetValue(memberName, value, tResult);
                break;
        }
    }
} 40
public override dynamic GetValue(string memberName)
{
    switch(memberName)
    {
        case "tStatus":
            return tStatus;
        case "tResult":
            return tResult;
    }
    return null;
} 50

```

```

}
public override T GetValue<T>(string memberName)
{
    switch(memberName)
    {
        case "tStatus":
            return (T)(object)tStatus;
        case "tResult":
            return (T)(object)tResult;
    }
    return default(T);
}
public EQUIPMENT_STATUS tStatus = new EQUIPMENT_STATUS();
public PROCESS_RESULT tResult = new PROCESS_RESULT();
}

```

10

EQUIPMENT__STATUSおよびPROCESS__RESULTは、同様のスタイルを有するが、ここでは示していない。

```

public class CONTROL_INTERFACE : PlcMember
{
    public override void SetValue(string memberName, dynamic value)
    {
        switch(memberName)
        {
            case "tCommand":
                tCommand = InternalSetValue(memberName, value, tCommand);
                break;
            case "tParameters":
                tParameters = InternalSetValue(memberName, value, tParameters);
                break;
        }
    }
    public override dynamic GetValue(string memberName)
    {
        switch(memberName)
        {
            case "tCommand":
                return tCommand;
            case "tParameters":
                return tParameters;
        }
        return null;
    }
    public override T GetValue<T>(string memberName)
    {
        switch(memberName)
        {
            case "tCommand":
                return (T)(object)tCommand;

```

20

30

40

50

```
case "tParameters":  
    return (T)(object)tParameters;  
}  
return default(T);  
}  
public COMMAND tCommand = new COMMAND();  
public PARAMETERS tParameters = new PARAMETERS();  
}
```

C O M M A N DおよびP A R A M E T E R Sは、同様のスタイルを有するが、ここでは示
していない。

10

【 図 1 】

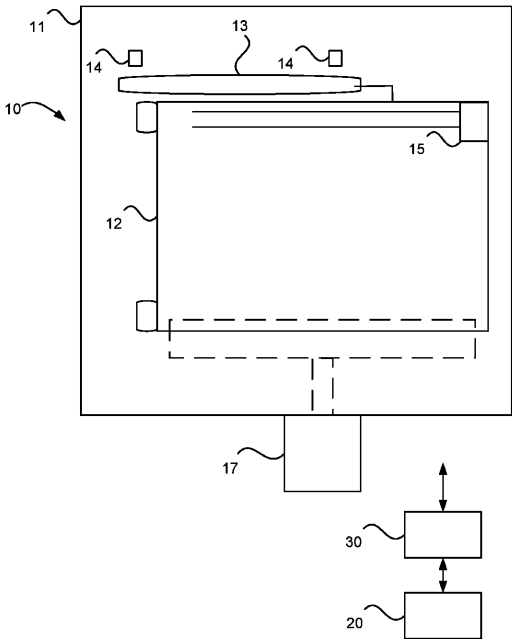


Fig. 1

【 図 2 】

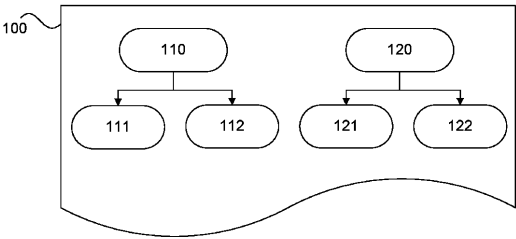


Fig. 2

【 図 3 】

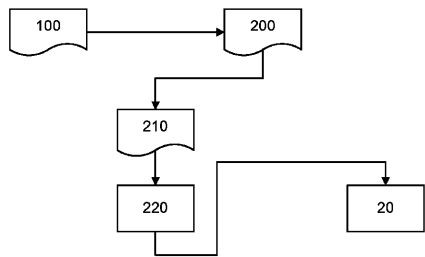


Fig. 3

【 図 4 】

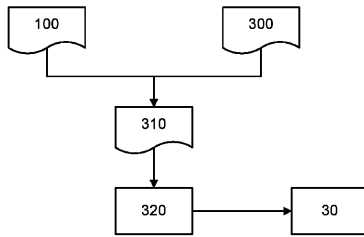


Fig. 4

【 手続補正書 】

【 提出日 】 令和1年7月2日 (2019.7.2)

【 手続補正 1 】

【 補正対象書類名 】 明細書

【 補正対象項目名 】 全文

【 補正方法 】 変更

【 補正の内容 】

【 発明の詳細な説明 】

【 技術分野 】

【 0001 】

本発明は、P C プログラムを用いて P L C を制御する方法、ならびに本方法を実行するためのコンピュータおよびコンピュータプログラムに関するものである。

【 背景技術 】

【 0002 】

P L C (プログラマブルロジックコントローラ) は、プロセスプラントまたは機械類を制御するための装置であり、デジタル的にプログラムされる。P L C は、通常、入力と、出力と、ファームウェア (オペレーティングシステム) と、アプリケーションプログラムをダウンロードするためのインタフェースと、を有する。アプリケーションプログラムは、入力に応じて、いかに出力が設定されるのかを定義する。ファームウェアによって、アプリケーションプログラムの入力および出力が利用可能となる。

【 0003 】

リアルタイムシステムとしての P L C は、デジタルセンサまたはアナログセンサと接続しており、プロセスプラントまたは機械類の空気圧シリンダもしくは油圧シリンダまたは電気モータのようなアクチュエータを制御する。特に半導体製造装置アプリケーションである多くの機械制御アプリケーションとして、典型的には、P L C および工場内ネットワ

ーク（例えば、ファブまたはファウンドリとしても知られる半導体製造工場のホストシステム）に接続されたパーソナルコンピュータ（PC）アプリケーションがあり、それは、これらのセンサ値をPLCから読み取って、これらの取得したリアルタイムデータについて高度な管理を実行し（例えば、これらのデータを状態変数としてファブホストに送信し）、個々のアクションを実行するためのコマンドをPLCに送信する。

【0004】

特許文献1は、PLCとして実施することが可能な制御環境であって、生産プロセスなどの装置の動作を制御する制御環境について開示している。産業オートメーションデータを表示するためのレンダリング環境が提供される。この制御環境は、装置の動作を制御するマシン実行可能制御コードを含むことができる。レンダリング環境の端末は、コンピュータ実行可能コード命令を実行するためのプロセッサ（群）を含む。

【0005】

特許文献2は、プログラマブルロジックコントローラのプログラミングおよび構成設定のための方法について開示しており、この方法は、IEC 61131言語のモジュールを含む構成ツールをプログラミングツール内に配置することと、少なくとも1つの追加モジュールを構成ツールに組み込むことと、Cコードエディタを用いて高水準言語でプログラマブルロジックコントローラのプログラミングおよび構成設定を行うことと、を含み、その少なくとも1つの追加モジュールは、IEC 61131コードと高水準言語コードとの間のインタフェースの定義、および高水準言語コードの構成、を提供するように構成される。

【0006】

従来のPCアプリケーションにおける問題は、例えば、新たなセンサの追加または新たなアクチュエータの追加など、PLCアプリケーション側に進展があるたびに、PCアプリケーション側でも、これらの新たに追加された要素にアクセスするための基本機能を実装するために、例えば以下のような、かなりの量の修正作業があることである。

- ・PCがアクセスできる新たに追加されたセンサ/アクチュエータに関するPLCインタフェースのドキュメンテーションを作成/修正する。
- ・新たに追加された要素のための、PLCアプリケーションとPCアプリケーションとの間の追加の通信リンクを追加する。
- ・新たに追加された要素を表現するPCプログラムにおいて内部変数/手続きを作成する。
- ・新たに追加されたセンサを表示するため、および新たに追加されたアクチュエータを制御するためにPLCにコマンドを送信する可能性を提供するために、PCプログラムのグラフィカルユーザインタフェースを作成/修正する。

【0007】

例えばOPCサーバを用いて、読み取りおよび/または書き込みのために、PLCの内部変数をPCに公開することができる。ただし、その場合にも、新たに提供される変数を用いるために、PCプログラムを手動で適応させなければならない。

【先行技術文献】

【特許文献】

【0008】

【特許文献1】米国特許出願公開第2014/0343696号明細書

【特許文献2】米国特許出願公開第2014/0123104号明細書

【発明の概要】

【発明が解決しようとする課題】

【0009】

よって、PCプログラムをPLCアプリケーション側の進展に適応させることに関わる必要な作業を軽減することが望ましい。

【課題を解決するための手段】

【0010】

本発明によれば、独立請求項の特徴により、ＰＣプログラムを用いてＰＬＣを制御するための方法、コンピュータプログラム、およびコンピュータについて提案する。効果的なさらなる改良は、従属請求項の主題および以下の説明の主題をなしている。

【００１１】

本発明により、ＰＬＣプログラムのソースコードのデータセクションおよびＰＣプログラムテンプレートから読み出されたデータに基づいて、ＰＬＣと通信および／またはＰＬＣを制御するために、好ましくはグラフィカルユーザインタフェース（ＧＵＩ）を有する、ＰＣプログラムを自動的に生成するための方法およびその実装を導入する。すなわち、そのデータセクションは、ＰＣプログラムとＰＬＣプログラムの両方に共通のソースコード部分となる。従って、ＰＣプログラムをＰＬＣアプリケーション側の進展に手動で適応させることによって導入されるエラーを防ぐことができる。ＰＬＣプログラム内では、データセクションは、ＰＬＣプログラムの実行時にＰＬＣによって公開されるサービスをそれぞれ指定または宣言するデータを含み、それに対応するサービス公開手続きによって用いることができる。ＰＣプログラム内では、これらのデータを、実際に公開されているサービスを把握するために用いることができるとともに、対応するサービス接続手続きによって用いることができ、さらに、好ましくは、それぞれの公開サービスのための視覚要素（入力または出力）を作成するために、ＧＵＩ作成手続きによって用いることができる。

【００１２】

本発明は、ＰＬＣプログラム側で進展または変更があつて、これらの変更をＰＣでも実装する必要がある場合に、ＰＣプログラム側での開発時間を節約することを目的とする。ＰＬＣプログラム側、すなわちＰＬＣプログラムのソースコードを適応させる必要は依然としてあるものの、ＰＣプログラムが何らかのさらなる開発を必要とすることはない。ＰＬＣプログラムのソースコードが定義または変更された場合に、新たなＰＣプログラムソースコードを作成するために、変更されたデータセクションをＰＣプログラムテンプレートに結合する必要があるのみである。これにより、ＰＣ側での開発時間が大幅に節約されるとともに、ヒューマンエラーが回避され、さらに、自動的な機能整合性チェックが行われる。

【００１３】

本発明は、効果的には、「監視制御とデータ取得」（ＳＣＡＤＡ：Ｓｕｐｅｒｖｉｓｏｒｙ Ｃｏｎｔｒｏｌ Ａｎｄ Ｄａｔａ Ａｃｑｕｉｓｉｔｉｏｎ）の枠内で用いることができる。ＳＣＡＤＡは、高度なプロセス監視管理のために、ＰＣ、ネットワーク化されたデータ通信、およびグラフィカルユーザインタフェースを用いる制御システムアーキテクチャであるが、プロセスプラントまたは機械類にインタフェースするために、ＰＬＣおよび分散ＰＩＤコントローラのような他の周辺装置を使用する。監視、およびコントローラ設定変更のようなプロセスコマンドの発行を可能とするオペレータインタフェースは、ＳＣＡＤＡ監視コンピュータシステムを介して処理される。一方、リアルタイム制御ロジックまたはコントローラ計算は、フィールドセンサおよびアクチュエータに接続するネットワーク化されたモジュールによって実行される。

【００１４】

好ましい実施形態によれば、ＰＬＣプログラムのソースコードは、高水準プログラミング言語であり、特にＣ系言語またはＩＥＣ ６１１３１－３で定義される言語である。ＩＥＣ ６１１３１－３は、プログラマブルロジックコントローラ用のオープン国際標準規格ＩＥＣ ６１１３１の（１０部のうちの）第３部である。ＩＥＣ ６１１３１の第３部は、ＰＬＣ内の制御プログラムの基本的なソフトウェアアーキテクチャおよびプログラミング言語を扱う。それは、２種類の図式プログラミング言語規格および２種類のテキストプログラミング言語規格、すなわち、ラダーダイアグラム（ＬＤ）図式言語；ファンクションブロックダイアグラム（ＦＢＤ）図式言語；ストラクチャードテキスト（ＳＴ）テキスト言語；インストラクションリスト（ＩＬ）テキスト言語、を定義している。

【００１５】

ＰＬＣプログラムの作成は、ＰＣ上でＰＬＣプログラミングツールキットを用いて行う

と非常に簡便である。P L Cプログラミングツールキットは、P Cプログラムのプログラミング用のプログラミングツールキットとは異なる可能性があるが、本発明の利点は、P L CプログラミングとP Cプログラミングとの間の言語の壁をいずれも乗り越えるために共通のファイルを共有することである。P L Cプログラムでは、このファイルは、ソースコードの一部であり、P Cプログラムでは、このファイルは、プログラムを自動的に生成するための入力である。本発明により、P CプログラムとP L Cプログラムの両方に関連したソースコードを一度のみ提供することによって、コード生成が改善および簡略化される。本発明により、この場合は本発明の意味においてデータセクションである少なくとも1つのモジュールが一度のみ提供されて、それが二度、すなわち両方のプログラムを作成するときに使用されるような、P CプログラムとP L Cプログラムの両方のソースコードのモジュール化について提案する。

【0016】

効果的には、そのデータセクションを、別個のファイルの形で提供することができる。これにより、そのデータセクションを、非常に簡単に、P L CプログラムソースコードおよびP Cプログラムソースコードに関連付けて利用する。そのデータセクションは、好ましくは、P L Cプログラムのソースコードのヘッダであり、より好ましくは、別個のヘッダファイルの形で提供される。

【0017】

好ましくは、P Cプログラムテンプレートは、テキストテンプレートである。従って、同じくテキストである、ソースコードからのデータセクションに関連付けて、P Cプログラムのソースコードを非常に簡単に自動生成することができる。特に、Microsoft社が提供するいわゆる「テキストテンプレート変換ツールキット」(T4)のような、いずれかの周知のコード生成ツールを効果的に使用することができる。開発者は、様々なパラメータを用いてテキストファイルの作成を自動化するためのアプリケーションフレームワークまたはツールフレームワークの一部として、T4を使用することができる。これらのテキストファイルは、最終的に、コード(例えば、(Visual)C#のようなC系、もしくはVisual BasicのようなBASIC系)、XML、HTML、またはXAMLのような、任意のテキスト形式で提供することができる。

【0018】

P CプログラムがP L Cと十分に通信可能とするためには、P Cプログラムを用いてP L Cを制御することは、好ましくは、P L Cへの値の書き込みおよび/またはP L Cからの値の読み取りを含む。典型的なP L Cでは、ランタイム中に、最終的にP L Cの動作を定義する特定の値を有する多くの異なる変数が、メモリに保存される。従って、そのような値を読み取ること、および書き込むことは、P L Cを制御するための基本的な方法である。さらに効果的には、P Cプログラムを用いてP L Cを制御することは、P L Cから情報を受信すること、および/またはP L Cにコマンドを送信すること、を含む。

【0019】

P CプログラムによりP L Cを制御することを可能とするためには、P L Cによって公開されるサービスは、好ましくは、P L Cにデータを書き込むための少なくとも1つのサービス、および/またはP L Cからデータを読み取るための少なくとも1つのサービス、を含む。これらのサービスにより、効果的に、P Cプログラムが、P L Cに値を書き込むこと、および/またはP L Cから値を読み取ること、ひいてはP L Cを制御することが可能となる。

【0020】

本発明の好ましい実施形態によれば、それらのサービスは、例えば、(特にOPC UAによる)OPC、PVI、Prof i Netなどのサーバである、P L C上で動作するサーバを用いて、P L Cによって公開される。これらは、特に、サーバから値(特に測定値)を読み取るため、および/またはサーバすなわちP L Cに値(特に設定値)を書き込むための、様々なサービスをクライアントに公開するための、十分に試行されたソリューションである。さらに、公開されたサービスに接続するために、例えば、B & Rからのい

わゆるPVI(Process Visualization Interface)など、サードパーティのプロシージャおよびクライアントを効果的に使用することができる。PVIは、PC上で動作可能であって、データを受信するためにPLCに接続するソフトウェアである。PCプログラムは、PLCに接続するために、PVIを補助ツールとして用いることができる。

【0021】

プログラムコードを含む、本発明によるコンピュータプログラムは、このコンピュータプログラムがコンピュータ上で実行されることで、本発明による方法のうちの1つのすべてのステップをコンピュータに実行させる。そのコンピュータプログラムを提供するための適切な非一時的マシン可読媒体は、特に、フロッピーディスク、ハードディスク、フラッシュメモリ、EEPROM、CD-ROM、およびDVDなどである。コンピュータネットワーク(インターネット、イントラネットなど)上のプログラムをダウンロードすることが効果的である。

【0022】

本発明のさらなる効果および実施形態は、説明および添付の図面から明らかになるであろう。

【0023】

前述の特徴および以下でさらに記載する特徴は、個々に示す組み合わせだけではなく、さらなる組み合わせまたは単独でも、本発明の範囲から逸脱することなく、用いることができることに留意すべきである。

【図面の簡単な説明】

【0024】

【図1】図1は、プロセスチェーンバと、いくつかのセンサ、およびPLCによって制御されるアクチュエータと、を備える例示的な機械を示している。

【図2】図2は、本発明の好ましい実施形態によるデータセクションを概略的に示している。

【図3】図3は、本発明の好ましい実施形態による、ヘッダファイルおよびPCプログラムテンプレートを用いてPCプログラムを作成するプロセスを概略的に示している。

【図4】図4は、本発明の好ましい実施形態による、ヘッダファイルおよびPLCメインプログラムファイルを用いてPLCプログラムを作成するプロセスを概略的に示している。

【発明を実施するための形態】

【0025】

図1は、製造プロセスで使用する基板キャリア(例えば、半導体ウェハキャリア、フラットパネルディスプレイキャリア、レチクルキャリア、および電子装置製造用の他のキャリア)のための自動洗浄・検査システムの一部であり得る機械10を概略的に示している。機械10は、ドア12と、この場合はドアを開閉するための空気圧シリンダとして構成されたアクチュエータ13と、を有するプロセスチェーンバ11を備える。ドア12が開いているのか閉じているのかを判断するために、アクチュエータの位置を検出するための2つのセンサ14が設けられている。

【0026】

プロセスチェーンバ11内には、チェーンバを所望の温度まで加熱するためのヒータ15と、電気モータ17によって所望の速度で回転させる回転板と、が設けられている。

【0027】

センサ14およびアクチュエータ13、15、17は、プロセスを制御するためのPLC30に接続されている。そしてPLC30は、PLCを制御するためのPC20に接続されており、例えば、PCは、PLCにコマンドを送信するとともに、PLCからデータを取得する。

【0028】

図2は、様々なサービスを示すための内部ブロックを有する、PLCプログラムのデー

タセクション 100 を概略的に示している。好ましくは、このデータセクションは、P L C プログラムのヘッダである。従って、そのデータを、信頼性の高い直接的な方法で読み出すことが可能である。

【0029】

前述のように、P L C によって公開されるサービスは、P L C にデータを書き込むための少なくとも 1 つのサービス、および / または P L C からデータを読み取るための少なくとも 1 つのサービス、を含むことができる。従って、実際には、このデータセクションは、P L C にデータを書き込むためのサービス (「C O N T R O L」) としての 1 つのセクションまたはブロック 110 と、P L C からデータを読み取るためのサービス (「I N F O」) としての 1 つのブロック 120 と、を含む。

【0030】

同様に、ブロック 110 は、コマンドとパラメータにそれぞれ関連した 2 つのサブブロック 111、112 に分割することができる。例えば、サブブロック 111 では、装置のアクション (例えば、ドアの開閉、プロセスの開始 / 停止) を開始させるための、P L C への可能なコマンドすなわち P C からの到来データを示すことができ、サブブロック 112 では、機械に関連したパラメータを設定するための、可能な書き込み可能パラメータすなわち P C からの到来データを示すことができる。

【0031】

基本的に、2 つのタイプのパラメータがあり、装置関連パラメータの場合、特定の装置構成部品がそのコマンドに関連しており (例えば、「R o t a t e T a b l e」コマンドの場合の所望速度)、プロセス関連パラメータの場合、何らかのプロセスの制約がそのコマンドに関連している (例えば、「R u n P r o c e s s」コマンドの場合の所望温度)。

【0032】

ブロック 120 は、プロセス結果と装置状態にそれぞれ関連した 2 つのサブブロック 121、122 に分割することができる。例えば、サブブロック 121 では、プロセスの結果 (例えば、「R u n P r o c e s s」コマンドの場合の測定された平均 / 最低 / 最高温度、および「D e s i r e d T e m p e r a t u r e」パラメータによる結果の評価) を報告するための、可能な読み取り可能プロセス結果すなわち P C に送信される発信データを示すことができ、サブブロック 122 では、装置状態 (例えば、ドアの開閉状態) および (何らかの障害が生じている場合の) 装置障害状態を報告するための、可能な読み取り可能装置状態データすなわち P C に送信される発信データを示すことができる。

【0033】

図 1 に示す例では、

- ・ P C は、チェンバの O P E N または C L O S E (2 つのコマンド) を P L C に要求できる。
- ・ P C は、ドアが O P E N または C L O S E D (2 つの装置状態) のどちらかである状態を通知される。
- ・ P C は、所望の温度および電気モータ速度 (2 つのコマンドパラメータ) でプロセスを開始させること (1 つのコマンド) を P L C に要求できる。
- ・ プロセスが終了すると、P C は、プロセス中の最低 / 最高 / 平均温度 (1 つのプロセス結果) を通知される。

【0034】

本例に関連した好ましいヘッダを、付録に示している。このヘッダファイルは、所望の装置機能に応じて、手動で設計されている。これは、P C プログラムに対して、インタフェースを提供するとともに、制御対象の P L C 装置のすべての能力を完全に規定している。

【0035】

一方で、このヘッダファイルは、図 3 に示すように P C プログラムを作成するために、使用される。他方で、このヘッダファイルは、図 4 に示すように P L C プログラムを作成

するためにも、使用される。

【 0 0 3 6 】

次に図 3 を参照して、P C プログラムソースコード 2 1 0 を生成するために、P C プログラムテンプレート 2 0 0 と共にヘッダファイル 1 0 0 を使用する。P L C 3 0 を制御するために P C 2 0 上で実行可能な P C プログラム 2 2 0 を作成するために、P C プログラムソースコード 2 1 0 をコンパイルすることができる。好ましくは、P C プログラム 2 2 0 は、P C 2 0 上で実行されると、グラフィカルユーザインタフェースを表示するように構成される。

【 0 0 3 7 】

次に図 4 を参照して、P L C プログラムソースコード 3 1 0 を生成するために、対応する P L C プログラムソースコード 3 0 0 と共にヘッダファイル 1 0 0 を同じく使用する。P L C プログラム 3 2 0 を作成するために、P L C プログラムソースコード 3 1 0 をコンパイルすることができ、P L C プログラム 3 2 0 は、P L C 3 0 に転送されて、実行されることで、プロセスを制御すると同時に、ヘッダファイル 1 0 0 で宣言されているサービスを、P C プログラムにより接続されるように公開する。

【 0 0 3 8 】

P C プログラムを用いて、ユーザは、

- ・チェンバの O P E N または C L O S E を P L C に要求できる。
- ・ドアが O P E N または C L O S E D のどちらかである状態を通知される。
- ・所望の温度および電気モータ速度でプロセスを開始させることを P L C に要求できる。
- ・プロセスが終了した後に、プロセス中の最低 / 最高 / 平均温度を通知される。

【 0 0 3 9 】

なお、上記説明は、開示の実施形態の態様の例示的なものにすぎないことは理解されるべきである。当業者であれば、開示の実施形態の態様から逸脱することなく、種々の代替案および変形例を考案することができる。よって、開示の実施形態の態様は、添付の請求項の範囲内にあるすべてのかかる代替案、変更、および変形を包含するものである。さらに、様々に異なる特徴が、互いに異なる従属請求項または独立請求項に記載されているということだけで、それらの特徴の組み合わせを効果的に用いることができないことを示すものではなく、かかる組み合わせは、本発明の態様の範囲から逸脱していない。

【 0 0 4 0 】

例示的なヘッダファイル：

```
typedef struct
{
    CONTROL_INTERFACE tControl; // control of slave (PLC) from master (PC)
    INFO_INTERFACE tInfo; // info back from slave (PLC) to master (PC)
} PLC_INTERFACE;

typedef struct
{
    EQUIPMENT_STATUS tStatus;
    PROCESS_RESULT tResult;
} INFO_INTERFACE;

typedef struct
{
    BOOL bProcessChamberClosed; // cylinder position sensor (Digital Input)
    BOOL bProcessChamberOpen; // cylinder position sensor (Digital Input)
    INT16 iMotorPosition; // motor position (Analog Input)
} EQUIPMENT_STATUS;
```

```
typedef struct
{
    REAL rMinTemperatureDuringProcess; // min temperature during process
    REAL rMaxTemperatureDuringProcess; // max temp during process
    REAL rAverageTemperatureDuringProcess; // average temp during process
} PROCESS_RESULT;
```

```
typedef struct
{
    COMMAND tCommand;
    PARAMETERS tParameters;
} CONTROL_INTERFACE;
```

```
typedef struct
{
    BOOL bOpenProcessChamber; // command to open the process chamber
    BOOL bCloseProcessChamber; // command to close the process chamber
    BOOL bRunProcess; // command to start a process
} COMMAND;
```

```
typedef struct
{
    REAL rTargetProcessTemperature; // target temperature during process
    REAL rTargetMotorSpeed; // target motor speed during process
} PARAMETERS
```

【 0 0 4 1 】

PC プログラムテンプレート 2 0 0 によって生成された PC プログラムソースコード 2 1 0 の例示的な部分：

```
public class PLC_INTERFACE : PlcMember
{
    public override void SetValue(string memberName, dynamic value)
    {
        switch(memberName)
        {
            case "tControl":
                tControl = InternalSetValue(memberName, value, tControl);
                break;
            case "tInfo":
                tInfo = InternalSetValue(memberName, value, tInfo);
                break;
        }
    }
    public override dynamic GetValue(string memberName)
    {
        switch(memberName)
        {
            case "tControl":
                return tControl;
            case "tInfo":
                return tInfo;
        }
    }
}
```

```
    }
    return null;
}
public override T GetValue<T>(string memberName)
{
    switch(memberName)
    {
        case "tControl":
            return (T)(object)tControl;
        case "tInfo":
            return (T)(object)tInfo;
    }
    return default(T);
}
public CONTROL_INTERFACE tControl = new CONTROL_INTERFACE(); // control of slave (PLC) from master (PC)
public INFO_INTERFACE tInfo = new INFO_INTERFACE(); // info back from slave (PLC) to master (PC)
}

public class INFO_INTERFACE : PlcMember
{
    public override void SetValue(string memberName, dynamic value)
    {
        switch(memberName)
        {
            case "tStatus":
                tStatus = InternalSetValue(memberName, value, tStatus);
                break;
            case "tResult":
                tResult = InternalSetValue(memberName, value, tResult);
                break;
        }
    }
}
public override dynamic GetValue(string memberName)
{
    switch(memberName)
    {
        case "tStatus":
            return tStatus;
        case "tResult":
            return tResult;
    }
    return null;
}
public override T GetValue<T>(string memberName)
{
    switch(memberName)
    {
        case "tStatus":
```

```

        return (T)(object)tStatus;
    case "tResult":
        return (T)(object)tResult;
    }
    return default(T);
}
public EQUIPMENT_STATUS tStatus = new EQUIPMENT_STATUS();
public PROCESS_RESULT tResult = new PROCESS_RESULT();
}

```

EQUIPMENT__STATUSおよびPROCESS__RESULTは、同様のスタイルを有するが、ここでは示していない。

```

public class CONTROL_INTERFACE : PlcMember
{
    public override void SetValue(string memberName, dynamic value)
    {
        switch(memberName)
        {
            case "tCommand":
                tCommand = InternalSetValue(memberName, value, tCommand);
                break;
            case "tParameters":
                tParameters = InternalSetValue(memberName, value, tParameters);
                break;
        }
    }
    public override dynamic GetValue(string memberName)
    {
        switch(memberName)
        {
            case "tCommand":
                return tCommand;
            case "tParameters":
                return tParameters;
        }
        return null;
    }
    public override T GetValue<T>(string memberName)
    {
        switch(memberName)
        {
            case "tCommand":
                return (T)(object)tCommand;
            case "tParameters":
                return (T)(object)tParameters;
        }
        return default(T);
    }
    public COMMAND tCommand = new COMMAND();
}

```

```
public PARAMETERS tParameters = new PARAMETERS();  
}
```

COMMANDおよびPARAMETERSは、同様のスタイルを有するが、ここでは示していない。

【手続補正2】

【補正対象書類名】特許請求の範囲

【補正対象項目名】全文

【補正方法】変更

【補正の内容】

【特許請求の範囲】

【請求項1】

PCプログラム(220)を用いてPLC(30)を制御する方法であって、

PLCプログラムのソースコード(310)は、データセクション(100)を含み、前記データセクション(100)は、前記PLCプログラムの実行時に前記PLC(30)によって公開されるサービスを示すデータを含む、方法において、

前記データセクション(100)に含まれる前記データおよびPCプログラムテンプレート(200)を用いて、前記PCプログラムのソースコード(210)を自動生成することと、

前記PCプログラムの前記自動生成されたソースコード(210)から生成された前記PCプログラム(220)を用いて、前記PLC(30)を制御することと、を特徴とする方法。

【請求項2】

前記PCプログラムテンプレート(200)は、テキストテンプレートである、請求項1に記載の方法。

【請求項3】

前記データセクション(100)は、前記PLCプログラムの前記ソースコード(310)のヘッダである、請求項1または2に記載の方法。

【請求項4】

前記PLCプログラムの前記ソースコード(310)は、高水準プログラミング言語である、請求項1～3のいずれか一項に記載の方法。

【請求項5】

前記PCプログラム(220)は、PC(20)上で実行されると、グラフィカルユーザインタフェースを表示するように構成される、請求項1～4のいずれか一項に記載の方法。

【請求項6】

前記PLC(30)によって公開される前記サービスは、前記PLCにデータを書き込むための少なくとも1つのサービス、および/または前記PLCからデータを読み取るための少なくとも1つのサービスを含む、請求項1～5のいずれか一項に記載の方法。

【請求項7】

前記PCプログラム(220)を用いて前記PLC(30)を制御することは、前記PLCから情報を受信すること、および/または前記PLCにコマンドを送信することを含む、請求項6に記載の方法。

【請求項8】

前記PCプログラム(220)を用いて前記PLC(30)を制御することは、前記PLCに値を書き込むこと、および/または前記PLCから値を読み取ることを含む、請求項6または7に記載の方法。

【請求項9】

前記サービスは、前記PLC(30)上で動作するサーバを用いて、前記PLC(30)によって公開される、請求項1～8のいずれか一項に記載の方法。

【請求項 10】

前記 P L C (3 0) は、前記 P L C プログラムを実行している、請求項 1 ~ 9 のいずれか一項に記載の方法。

【請求項 11】

P L C (3 0) を制御するための P C プログラム (2 2 0) を自動的に作成する方法であって、

P L C プログラムのソースコード (3 1 0) のデータセクション (1 0 0) に含まれるデータおよび P C プログラムテンプレート (2 0 0) を用いて、前記 P C プログラムのソースコード (2 1 0) を自動生成することと、

前記データセクションは、前記 P L C プログラムの実行時に前記 P L C (3 0) によって公開されるサービスを示すデータを含むことと、を特徴とする方法。

【請求項 12】

請求項 1 ~ 11 のいずれか一項に記載の方法を実行するように、特にプログラムの、適応させたコンピュータ。

【請求項 13】

プログラムコードを有するコンピュータプログラムであって、コンピュータ上で実行されることで、請求項 1 ~ 11 のいずれか一項に記載の方法を前記コンピュータに実行させるための、コンピュータプログラム。

【請求項 14】

請求項 13 に記載のコンピュータプログラムを格納している、マシン可読記憶媒体。

【国際調査報告】

INTERNATIONAL SEARCH REPORT

International application No

PCT/EP2018/064936

A. CLASSIFICATION OF SUBJECT MATTER

INV. G05B19/042 G06F9/44 G06F8/38
ADD. G05B19/05

According to International Patent Classification (IPC) or to both national classification and IPC

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)

G05B G06F

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched

Electronic data base consulted during the international search (name of data base and, where practicable, search terms used)

EPO-Internal, WPI Data

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X	US 2014/343696 A1 (PLACHE KENNETH SCOTT [US] ET AL) 20 November 2014 (2014-11-20)	1,2,5-14
Y	paragraph [0035] - paragraph [0039]; figure 2 paragraph [0047] - paragraph [0049]; figure 3 paragraph [0053]; figure 4A paragraph [0030] - paragraph [0031] paragraph [0066] - paragraph [0067]; figure 7	3,4
Y	----- US 2014/123104 A1 (FROHBERGER ANKE [DE] ET AL) 1 May 2014 (2014-05-01) paragraph [0063] - paragraph [0079] paragraph [0020] -----	3,4

☐ Further documents are listed in the continuation of Box C.☒ See patent family annex.

* Special categories of cited documents :

"A" document defining the general state of the art which is not considered to be of particular relevance

"E" earlier application or patent but published on or after the international filing date

"L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)

"O" document referring to an oral disclosure, use, exhibition or other means

"P" document published prior to the international filing date but later than the priority date claimed

"T" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention

"X" document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone

"Y" document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art

"&" document member of the same patent family

Date of the actual completion of the international search

15 August 2018

Date of mailing of the international search report

24/08/2018

Name and mailing address of the ISA/

European Patent Office, P.B. 5818 Patentlaan 2
NL - 2280 HV Rijswijk
Tel. (+31-70) 340-2040,
Fax: (+31-70) 340-3016

Authorized officer

Bernard, Eddy

INTERNATIONAL SEARCH REPORT

Information on patent family members

International application No

PCT/EP2018/064936

Patent document cited in search report	Publication date	Patent family member(s)	Publication date
US 2014343696	A1	20-11-2014	CN 103149849 A 12-06-2013
			CN 106383456 A 08-02-2017
			EP 2541354 A2 02-01-2013
			US 2013006395 A1 03-01-2013
			US 2014343696 A1 20-11-2014
			US 2017329299 A1 16-11-2017

US 2014123104	A1	01-05-2014	CN 103814333 A 21-05-2014
			EP 2729855 A1 14-05-2014
			US 2014123104 A1 01-05-2014
			WO 2013004389 A1 10-01-2013

フロントページの続き

(81)指定国・地域 AP(BW, GH, GM, KE, LR, LS, MW, MZ, NA, RW, SD, SL, ST, SZ, TZ, UG, ZM, ZW), EA(AM, AZ, BY, KG, KZ, RU, TJ, TM), EP(AL, AT, BE, BG, CH, CY, CZ, DE, DK, EE, ES, FI, FR, GB, GR, HR, HU, IE, IS, IT, LT, LU, LV, MC, MK, MT, NL, NO, PL, PT, RO, RS, SE, SI, SK, SM, TR), OA(BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, KM, ML, MR, NE, SN, TD, TG), AE, AG, AL, AM, AO, AT, AU, AZ, BA, BB, BG, BH, BN, BR, BW, BY, BZ, CA, CH, CL, CN, CO, CR, CU, CZ, DE, DJ, DK, DM, DO, DZ, EC, EE, EG, ES, FI, GB, GD, GE, GH, GM, GT, HN, HR, HU, ID, IL, IN, IR, IS, JO, JP, KE, KG, KH, KN, KP, KR, KW, KZ, LA, LC, LK, LR, LS, LU, LY, MA, MD, ME, MG, MK, MN, MW, MX, MY, MZ, NA, NG, NI, NO, NZ, OM, PA, PE, PG, PH, PL, PT, QA, RO, RS, RU, RW, SA, SC, SD, SE, SG, SK, SL, SM, ST, SV, SY, TH, TJ, TM, TN, TR, TT

(72)発明者 リー , ファン

ドイツ連邦共和国・ 7 8 2 6 7 ・アーハ・ツアー ヴェベライ・ 2

(72)発明者 ギチャルダス , エルヴェ

ドイツ連邦共和国・ 7 8 2 3 9 ・リーラジンゲン - ヴォル布林ゲン・アルベルト - テン - ブリンク - シュトラーセ・ 8

F ターム(参考) 5H220 AA04 BB12 CC06 CX02 DD01 JJ12 JJ24